

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**GENERACIÓN DE ENTORNOS EDUCATIVOS PARA EL ESTUDIO
DE LOS PROCESOS INVOLUCRADOS EN LAS TECNOLOGÍAS 4G
Y 5G MEDIANTE EL USO DE HERRAMIENTAS OPEN SOURCE**

**Tesis para obtener el título profesional de Ingeniero de las
Telecomunicaciones**

AUTORES:

Agustin Junior Vizcarra Lizarbe

Ronny Eduardo Pastor Kolmakov

ASESOR:

César Augusto Santiváñez Guarniz

Lima, Diciembre, 2024

Informe de Similitud

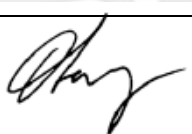
Yo, **César Augusto SANTIVÁÑEZ GUARNIZ**, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada

“GENERACIÓN DE ENTORNOS EDUCATIVOS PARA EL ESTUDIO DE LOS PROCESOS INVOLUCRADOS EN LAS TECNOLOGÍAS 4G Y 5G MEDIANTE EL USO DE HERRAMIENTAS OPEN SOURCE”

de los autores **Agustín Junior Vizcarra Lizarbe y Ronny Eduardo Pastor Kolmakov**, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de **14%**. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 17/12/2024.
- He revisado con detalle dicho reporte y la Tesis y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: San Miguel, 18 de Diciembre de 2024

Apellidos y nombres del asesor: SANTIVÁÑEZ GUARNIZ, César Augusto	
DNI: 09339312	Firma 
ORCID: 0000-0002-0050-3631	

RESUMEN

Hoy en día existe una problemática debido a la adopción tardía de 5G en el Perú, la cual radica en 3 factores: la brecha en la cobertura 4G, la ausencia de un marco regulatorio basado en estándares internacionales y el déficit de capital humano debidamente capacitado para operar, desplegar y hacer uso de las nuevas características de esta tecnología de forma productiva. Ante este panorama, se busca a través de la generación de entornos educativos virtualizados mediante la adopción de herramientas *open source*, desarrollar los conocimientos tanto prácticos como teóricos sobre las tecnologías 4G y 5G, al igual que las tecnologías emergentes tal como es Open RAN haciendo uso del Near RT-RIC, SBA, CUPS y *network slicing*. Para cumplir con este propósito, se diseñó e implementó un sistema basado en la integración entre un gestor de infraestructura abierto como Openstack y un microorquestador, el cual mediante APIs era capaz de desplegar distintas topologías según el escenario de interés. Asimismo, se generaron imágenes de máquinas virtuales acondicionadas con los componentes de *core* y de RAN y herramientas que permiten la trazabilidad de forma que facilite el estudio para cada tipo de escenario propuesto. Adicionalmente, la solución provee acceso a los elementos de red y sus interfaces externas de forma que se puedan realizar las pruebas de funcionamiento respectivas. Para demostrar el funcionamiento de los escenarios se realizaron diversas capturas de paquetes, tanto a nivel de *core* como RAN, y, haciendo uso de la herramienta *Wireshark*, se validó mediante una comparación con los estándares propuestos por la 3GPP y la O-RAN Alliance. Finalmente, con la validación obtenida se pudo evidenciar que los escenarios generados cumplen con los requerimientos educativos pues contribuía con la formación académica de los profesionales que necesita el país y la industria móvil; a su vez, esto permite la apertura de nuevas líneas de investigación en el desarrollo de las tecnologías móviles y surge como una alternativa económica ante las soluciones COTS de alto precio.

DEDICATORIA

A mis padres, Ronny Pastor y Elena Kolmakova, por ser las personas que me acompañaron en todo el proceso universitario y fueron parte en mi crecimiento como persona

También a mi Hermana, Galia Pastor, mi compañera que siempre estuvo apoyándome y motivando en los momentos más complicados.

Ronny Pastor

Dedico el presente trabajo a mis padres, profesores y compañeros de la especialidad de Ingeniería de las Telecomunicaciones

Agustin Vizcarra



AGRADECIMIENTOS

En el transcurso de mi etapa universitaria tuve la oportunidad de rodearme con personas que me ayudaron en este proceso.

Muchas gracias al profesor César Santivañez que durante este complicado proceso siempre nos motivó ante todos los aspectos como académicos y personales.

Muchas gracias al profesor Jose Rodriguez que nos ayudó a entender más sobre las tecnologías móviles y brindar todo su apoyo durante esta etapa.

Muchas gracias al profesor David Chávez por todo el apoyo metodológico para el desarrollo de la tesis.

Muchas gracias a mi compañero Agustin Vizcarra por brindar el soporte necesario en los momentos complicados de esta etapa Muchas gracias a todas las personas que se me apoyaron en esta etapa, tanto Lucía Delgado que me brindó tanto apoyo moral y emocional en esta etapa como mi grupo de amigos que en los momentos difíciles no se olvidan de tratar de sacarte una sonrisa y apoyarme.

Pero especialmente gracias a mi familia que me fomentaron y motivaron a crecer profesionalmente y personalmente y que brindaron su apoyo incondicional en toda esta etapa.

Ronny Pastor

Agradezco a mi asesor Cesar Santivañez y al profesor Jose Rodriguez por la exigencia y buena disposición con el presente trabajo, porque esto refleja todo el esfuerzo conjunto realizado en los últimos meses en pro de mejorar la calidad educativa de nuestra especialidad y el compromiso que tenemos con la calidad en la enseñanza de nuestra casa de estudios.

Agradezco al grupo GIRA y a todos sus integrantes por la buena disposición y el apoyo incondicional en el desarrollo de la presente tesis, así como al equipo de NOKIA USA por siempre estar dispuestos a apoyarnos. Agradezco de sobremanera a mis padres por haberme formado en valores ya que este trabajo es fruto de la disciplina y perseverancia que me han inculcado y por lo cual yo me encuentro sumamente agradecido.

Agradezco a mi compañero Ronny por el soporte brindado durante todos estos meses y por su buena disposición en la realización del presente trabajo.

Agradezco a todos los profesores de la especialidad de Ingeniería de Telecomunicaciones, puesto que, gracias a su compromiso en la enseñanza, los aportes al sector educativo y a la producción de profesionales van incrementando y junto con ello el desarrollo tecnológico en pro de tener un Perú mejor.

Agradezco a todos mis amigos que estuvieron en las buenas y malas dentro de mi proceso formativo y agradezco todas las experiencias compartidas con ellos.

Agustin Vizcarra

ÍNDICE GENERAL

RESUMEN	<i>i</i>
DEDICATORIA	<i>ii</i>
AGRADECIMIENTOS	<i>iii</i>
ÍNDICE GENERAL	<i>v</i>
ÍNDICE DE TABLAS	<i>ix</i>
ÍNDICE DE FIGURAS	<i>x</i>
GLOSARIO	<i>xv</i>
INTRODUCCIÓN	<i>1</i>
Capítulo 1. Entornos educativos para el estudio de redes 4G y 5G en el Perú debido a la situación actual	<i>5</i>
1.1 Definición del problema y justificación	<i>5</i>
1.1.1 Situación de las redes móviles.....	<i>5</i>
1.1.2 Situación de las redes móviles en América Latina y el Caribe.....	<i>6</i>
1.1.3 Situación de las redes móviles en el Perú.....	<i>8</i>
1.1.4 Situación de las tecnologías móviles enfocados en el aspecto académico en el Perú.....	<i>12</i>
1.2 Definición de requerimientos	<i>12</i>
1.3 Objetivos	<i>13</i>
1.3.1 Objetivo principal.....	<i>13</i>
1.3.2 Objetivos específicos.....	<i>14</i>
Capítulo 2. Análisis de las herramientas y paradigmas útiles para el diseño e implementación de entornos educativos 4G y 5G.	<i>16</i>
2.1 Nuevos paradigmas en la industria 4G y 5G	<i>16</i>

2.1.1	Arquitecturas desagregadas	16
2.1.2	Arquitectura basada en servicios	19
2.1.3	MANO	21
2.1.4	CNF	24
2.1.5	CNA (<i>Cloud-Native Architecture</i>) 5G/4G	25
2.2	Herramientas útiles para entornos educativos 4G y 5G	27
2.2.1	Análisis comparativo de herramientas RAN y Core	27
2.1.2	Análisis comparativo de herramientas de orquestación	37
Capítulo 3. Diseño y selección de herramientas para el despliegue de entornos educativos para las tecnologías 4G y 5G		47
3.1	Selección de paradigma	47
3.2	Diseño de arquitectura de la solución y herramientas de despliegue	51
3.2.1	Gestor de Infraestructura Virtual (VIM): Openstack	51
3.2.2	Servidor Backend	55
3.3	Escenarios	56
3.3.1	<i>Attachment and service request</i>	57
3.3.2	<i>Network slicing</i>	58
3.3.3	Interfaz E2 implementación de RIC	58
3.4	Herramientas de simulación y emulación	59
3.4.1	Red Core	59
3.4.2	Red Acceso	59
3.4.2	Resumen de las herramientas a utilizar	60
3.5	Arquitectura de los entornos educativos	60
3.5.1	Arquitectura para pruebas de <i>Attachment and service request</i>	60
3.5.2	Arquitectura para pruebas de <i>Network Slicing</i>	63
3.5.3	Arquitectura para el estudio de la <i>Interfaz E2</i> y el <i>RAN Intelligent Controller (RIC)</i>	64
3.6	Limitaciones de diseño	65

Capítulo 4. Análisis e Implementación de entornos educativos 4G y 5G basados en herramientas Open Source		67
4.1	Análisis de la solución.....	67
4.1.1	Aspecto educativo	67
4.1.2	Aspecto económico	67
4.1.3	Aspecto ambiental.....	70
4.1.3	Aspecto legal.....	70
4.2	Implementación de los escenarios diseñados.....	71
4.2.1	<i>flavors</i>	71
4.2.2	Imágenes	73
4.2.3	Redes.....	74
4.2.4	Topologías	74
4.2.5	Acceso	77
Capítulo 5. Análisis de resultados a nivel de implementación de los entornos 4G y 5G.....		78
5.1	<i>Attachment y PDN Connectivity Request - 4G</i>	78
5.1.1	Análisis y comparación de paquetes	80
5.1.2	eNodeB - UE.....	82
5.1.3	Prueba de funcionamiento.....	83
5.2	<i>Attachment y PDU Session Establishment - 5G</i>	86
5.2.4	gNodeB - UE	96
5.2.5	Prueba de funcionamiento	97
5.3	DN Corporativo.....	100
5.3.1	Core	100
5.3.2	gNodeB.....	103
5.3.3	UE.....	104
5.4	<i>Near RT-RIC</i>	108
5.4.1	<i>FlexRIC callflow</i>	109

5.4.2 Prueba de funcionamiento	110
5.5 Network Slicing	111
5.5.2 Prueba de funcionamiento	118
5.6 Resultados de aprendizaje.....	120
CONCLUSIONES.....	122
RECOMENDACIONES Y OBSERVACIONES.....	123
BIBLIOGRAFÍA.....	124
ANEXOS.....	130



ÍNDICE DE TABLAS

TABLA I . COMPARATIVA ENTRE REDES COMERCIALES 4G Y 5G.....	5
TABLA II. BRECHA DE ESTACIONES BASE POR DEPARTAMENTO PARA EL AÑO 2025.....	9
TABLA III. CRITERIOS DE USABILIDAD DE HERRAMIENTAS CORE PARA ENTORNOS EDUCATIVOS ..	32
TABLA IV. COMPARATIVA DE HERRAMIENTAS DE CORE A NIVEL DE PROCEDURES.....	33
TABLA V. CRITERIOS DE USABILIDAD DE HERRAMIENTAS RAN PARA ENTORNOS EDUCATIVOS	36
TABLA VI. COMPARATIVA DE HERRAMIENTAS DE RAN A NIVEL DE PROTOCOLOS.....	37
TABLA VII. REQUERIMIENTOS DE HARDWARE DEL CLUSTER DE OPENSTACK.....	53
TABLA VIII. DISTRIBUCIÓN DE HERRAMIENTAS POR ESCENARIO	60
TABLA IX. LISTADO DE PRECIOS DE LAS SOLUCIONES DE AMARISOFT.....	68
TABLA X. RECURSOS NECESARIOS PARA LA GENERACIÓN DE TESTBED POR ALUMNO	69
TABLA XI. ESTIMACIÓN DE PRECIOS SEGÚN PROVEEDORES DE NUBE PÚBLICA	69
TABLA XII. REQUERIMIENTOS DE HARDWARE POR NETWORK ELEMENT SEGÚN CASO.....	72
TABLA XIII. LISTADO DE ESCENARIOS EDUCATIVOS DE INTERÉS.....	76
TABLA XIV. LISTADO DE INTERFACES PARA EL ESCENARIO DE 4G.....	78
TABLA XV. LISTADO DE INTERFACES PARA EL ESCENARIO DE 5G	86
TABLA XVI. LISTADO DE INTERFACES PARA EL ESCENARIO DE DN CORPORATIVO	100
TABLA XVII. LISTADO DE INTERFACES PARA EL ESCENARIO DE O-RAN NEAR RT-RIC	108
TABLA XVIII. LISTADO DE INTERFACES PARA EL ESCENARIO DE NETWORK SLICING.....	111

ÍNDICE DE FIGURAS

FIGURA 1. TENDENCIA DE LAS TECNOLOGÍAS MÓVILES PARA EL AÑO 2025 A NIVEL DE LATINOAMÉRICA.....	6
FIGURA 2. ADOPCIÓN DE 5G EN 2025	7
FIGURA 3. USO DE TECNOLOGÍAS MÓVILES PARA EL AÑO 2021 Y PROYECCIÓN PARA EL AÑO 2025 EN PERÚ	8
FIGURA 4. RED CORE DESAGREGADA	17
FIGURA 5. SEPARACIÓN DEL PLANO DE USUARIO Y EL PLANO DE CONTROL A NIVEL DE DU Y CU.....	18
FIGURA 6. COMPONENTES DE UNA ARQUITECTURA OPEN RAN.....	18
FIGURA 7. ARQUITECTURA DE NFV-MANO	22
FIGURA 8. EVOLUCIÓN DE LAS FUNCIONES DE RED MÓVIL	25
FIGURA 9. ARQUITECTURA BASADA EN LA NUBE DE UN CORE MÓVIL	26
FIGURA 10. ASPECTOS FUNDAMENTALES DE UNA ARQUITECTURA CLOUD NATIVE.....	27
FIGURA 11. COMPONENTES DE OPEN5GS.....	28
FIGURA 12. COMPONENTES DE OAI-CORE	29
FIGURA 13. COMPONENTES DE MAGMA	30
FIGURA 14. COMPONENTES DE SRS-EPC.....	31
FIGURA 15. ARQUITECTURA DE OSM.....	38
FIGURA 16. ARQUITECTURA DE ONAP.....	40
FIGURA 17. ESTRUCTURA DE KUBERNETES	41
FIGURA 18. MATRIZ DE COMPARACIÓN ENTRE ONAP Y OSM	44
FIGURA 19. COMPARACIÓN DE LOS TIEMPOS DE DESPLIEGUE EN FUNCIÓN DE LAS INSTANCIAS DESPLEGADAS	45
FIGURA 20. COMPARACIÓN DEL CONSUMO DE CPU EN FUNCIÓN DE LA CANTIDAD DE INSTANCIAS	45
FIGURA 21. COMPARACIÓN DEL CONSUMO DE RAM EN FUNCIÓN DE LA CANTIDAD DE INSTANCIAS	46
FIGURA 22. PARADIGMAS DE IMPLEMENTACIÓN.....	47
FIGURA 23. COMPARATIVA DE LOS PARADIGMAS SEGÚN ENFOQUE EDUCATIVO	50
FIGURA 24. TOPOLOGÍA ESTABLECIDA PARA EL GESTOR DE INFRAESTRUCTURA	53
FIGURA 25. ESQUEMA DE NETWORKING RELACIONADO A UN ESQUEMA NAT PAT.....	54
FIGURA 26. FLUJO LÓGICO DE INTEGRACIÓN DEL SERVIDOR OPENSTACK CON EL SERVIDOR REST.	56
FIGURA 27. ARQUITECTURA DEL ESCENARIO PROPUESTO 5G.....	61
FIGURA 28. ARQUITECTURA DEL ESCENARIO PROPUESTO 4G.....	61

FIGURA 29. ARQUITECTURA DEL ESCENARIO DN CORPORATIVO.....	62
FIGURA 30. ARQUITECTURA DEL ESCENARIO NETWORK SLICING.....	63
FIGURA 31. ARQUITECTURA DEL ESCENARIO 5G CON NEARRT-RIC.....	65
FIGURA 32. IMPLEMENTACIÓN DE ESCENARIO CON INTERFAZ AÉREA USANDO USRPs.....	66
FIGURA 33. ARQUITECTURA DE AMARISOFT CALLBOX MINI.....	68
FIGURA 34. ARQUITECTURA DE AMARISOFT CALLBOX CLASSIC	68
FIGURA 35. ARQUITECTURA DE AMARISOFT UE SIMBOX.....	68
FIGURA 36. LISTA DE IMÁGENES CONFIGURADAS EN EL CLUSTER DE OPENSTACK	73
FIGURA 37. PARÁMETROS DE ENVÍO AL SERVICIO REST	74
FIGURA 38. RESPUESTA DEL SERVICIO REST	75
FIGURA 39. LISTADO DE INSTANCIAS CREADAS	75
FIGURA 40. LISTADO DE REDES CREADAS	76
FIGURA 41. TOPOLOGÍA TIPO 1	76
FIGURA 42. ASIGNACIÓN DE PUERTOS PARA EL ACCESO REMOTO.....	77
FIGURA 43. ACCESO REMOTO	77
FIGURA 44. CALLFLOW DE ATTACHMENT Y PDN CONNECTIVITY REQUEST PARA 4G PARTE 1	79
FIGURA 45. CALLFLOW DE ATTACHMENT Y PDN CONNECTIVITY REQUEST PARA 4G PARTE 2	80
FIGURA 46. CALLFLOW EN EL CORE 4G SUBSECCIÓN B-1	80
FIGURA 47. CALLFLOW EN EL CORE 4G SUBSECCIÓN C-1	81
FIGURA 48. CALLFLOW ENTRE ENB Y EL CORE 4G SUBSECCIONES B-2 Y B-3	81
FIGURA 49. CALLFLOW EN EL CORE 4G SUBSECCIÓN C-2	81
FIGURA 50. CALLFLOW EN EL CORE 4G SUBSECCIÓN C-3	81
FIGURA 51. CALLFLOW ENTRE ENB Y EL CORE 4G SUBSECCIONES B-4	82
FIGURA 52. CALLFLOW EN EL CORE 4G SUBSECCIÓN C-4	82
FIGURA 53. CAPTURAS OBTENIDAS A NIVEL MAC ENTRE EL UE Y EL ENB.....	82
FIGURA 54. COMANDO DE INICIALIZACIÓN DE EPC	83
FIGURA 55. COMANDO DE VERIFICACIÓN DE FUNCIONAMIENTO	83
FIGURA 56. COMANDO DE INICIALIZACIÓN DE ENB.....	83
FIGURA 57. COMANDO DE VERIFICACIÓN DE FUNCIONAMIENTO	84
FIGURA 58. COMANDO DE INICIALIZACIÓN DE UE	84

FIGURA 59. COMANDO DE VERIFICACIÓN DE FUNCIONAMIENTO - 1	84
FIGURA 60. COMANDO DE VERIFICACIÓN DE FUNCIONAMIENTO - 2	85
FIGURA 61. PING A SERVIDOR DE GOOGLE.COM.....	85
FIGURA 62. VALIDACIÓN DEL PROTOCOLO GTP	85
FIGURA 63. CALLFLOW DE ATTACHMENT 5G.....	87
FIGURA 64. CALLFLOW DE SERVICE REQUEST 5G	87
FIGURA 65. CALLFLOW DE PDU ESTABLISHMENT 5G	88
FIGURA 66. DECODIFICACIÓN DE HTTP2 EN WIRESHARK	88
FIGURA 67. MODELO DE COMUNICACIÓN INDIRECTA MEDIANTE EL USO DE SCP.....	89
FIGURA 68. PROCESO DE AUTENTICACIÓN PARTE 1.....	90
FIGURA 69. PROCESO DE AUTENTICACIÓN PARTE 2.....	90
FIGURA 70. PROCESO DE SUSCRIPCIÓN Y OBTENCIÓN DE DATA PARTE 1	91
FIGURA 71. PROCESO DE SUSCRIPCIÓN Y OBTENCIÓN DE DATA PARTE 2	91
FIGURA 72. PROCESO DE SUSCRIPCIÓN Y OBTENCIÓN DE DATA PARTE 3	91
FIGURA 73. APLICACIÓN DE POLÍTICAS.....	92
FIGURA 74. CREACIÓN DEL SM CONTEXT	92
FIGURA 75. SELECCIÓN DEL UPF Y ESTABLECIMIENTO DE LA SESIÓN	93
FIGURA 76. REVISIÓN DE PARÁMETROS DE SESIÓN	93
FIGURA 77. ASIGNACIÓN DE POLÍTICAS	93
FIGURA 78. SOLICITUD PARA PDU SESSION ESTABLISHMENT.....	94
FIGURA 79. SOLICITUD PARA PDU SESSION ESTABLISHMENT - INTERFAZ DEL UPF	94
FIGURA 80. SOLICITUD DE MENSAJE N1 N2.....	94
FIGURA 81. ACTUALIZACIÓN DE LA SESIÓN DE DATOS.....	94
FIGURA 82. REGISTRACIÓN.....	95
FIGURA 83. CONFIGURACIÓN DE IP Y ASIGNACIÓN.....	95
FIGURA 84. CONFIGURACIÓN DE DN	96
FIGURA 85. CONFIGURACIÓN DE IP Y ASIGNACIÓN.....	96
FIGURA 86. COMANDO DE INICIALIZACIÓN DE CORE 5G	97
FIGURA 87. OUTPUT DEL COMANDO DE INICIALIZACIÓN.....	97
FIGURA 88. COMANDO DE INICIALIZACIÓN DE UPF	97

FIGURA 89. OUTPUT DEL COMANDO DE INICIALIZACIÓN DE UPF.....	97
FIGURA 90. COMANDO DE INICIALIZACIÓN DE GNB.....	98
FIGURA 91. COMANDO DE INICIALIZACIÓN DE UE	98
FIGURA 92. VERIFICACIÓN DE CREACIÓN DE INTERFAZ TUN_SRSUE	99
FIGURA 93. VERIFICACIÓN DE CONECTIVIDAD A TRAVÉS DEL COMANDO PING	99
FIGURA 94. VERIFICACIÓN DEL PROTOCOLO GTP COMO ENCAPSULADOR DEL PAQUETE ICMP	99
FIGURA 95. PROCEDIMIENTOS RELACIONADOS AL UE 1 A NIVEL DE CORE.....	101
FIGURA 96. DIAGRAMA DE CALL FLOW DE PROCESOS DE CORE PARA EL UE 1.....	101
FIGURA 97. PROCEDIMIENTOS RELACIONADOS AL UE 2 A NIVEL DE CORE.....	101
FIGURA 98. DIAGRAMA DE CALL FLOW DE PROCESOS DE CORE PARA EL UE 2.....	102
FIGURA 99. ESTABLECIMIENTO DE PDU SESSION EN EL UPF.....	102
FIGURA 100. DN PARA EL UE 1	103
FIGURA 101. DN PARA EL UE 2	103
FIGURA 102. CONEXIÓN EXITOSA DEL UE 1	104
FIGURA 103. CONEXIÓN EXITOSA DEL UE 2	104
FIGURA 104. LOGS DE CONEXIÓN DEL UE 1.....	104
FIGURA 105. ASIGNACIÓN DE IP DEL UE 1	105
FIGURA 106. LOGS DE CONEXIÓN DEL UE 2.....	105
FIGURA 107. ASIGNACIÓN DE IP DEL UE 2	105
FIGURA 108. PRUEBA DE CONECTIVIDAD DEL UE 1	106
FIGURA 109. FLUJO DE TRÁFICO POR MEDIO DEL PROTOCOLO GTP-U PARA EL UE 1	106
FIGURA 110. PRUEBA DE CONECTIVIDAD DEL UE 2	106
FIGURA 111. FLUJO DE TRÁFICO POR MEDIO DEL PROTOCOLO GTP-U PARA EL UE 2	107
FIGURA 112. PRUEBA DE CONECTIVIDAD DE FORMA SIMULTÁNEA	107
FIGURA 113. CALLFLOW DE LA IMPLEMENTACIÓN DE UN RIC A TRAVÉS DE LA INTERFAZ E2	108
FIGURA 114. PAQUETES DE ESTABLECIMIENTO DE CONEXIÓN Y SUSCRIPCIÓN DEL RIC Y XAPP	109
FIGURA 115. CALLFLOW RELACIONADO AL ESCENARIO.....	109
FIGURA 116. COMANDO DE INICIALIZACIÓN DEL FLEXRIC	110
FIGURA 117. COMANDO DE INICIALIZACIÓN DEL GNB.....	110
FIGURA 118. COMANDO DE INICIALIZACIÓN DEL XAPP	111

FIGURA 119. OUTPUT DEL XAPP	111
FIGURA 120. NETWORK SLICING CALL FLOW	112
FIGURA 121. SLICE INFORMATION REQUEST UE 1	113
FIGURA 122. SLICE INFORMATION REQUEST UE 2	113
FIGURA 123. SLICE INFORMATION RESPONSE UE 1	114
FIGURA 124. SLICE INFORMATION RESPONSE UE 2	114
FIGURA 125. NETWORK SLICES SOPORTADOS POR EL AMF	114
FIGURA 126. NETWORK SLICING CALL FLOW CON RECOLOCACIÓN DE AMF	115
FIGURA 127. ASIGNACIÓN DE SM CONTEXT DEL UE 1	115
FIGURA 128. ASIGNACIÓN DE SM CONTEXT DEL UE 2	116
FIGURA 129. PARÁMETROS PARA ESTABLECIMIENTO DE SESIÓN EN EL UE 1	116
FIGURA 130. PARÁMETROS PARA ESTABLECIMIENTO DE SESIÓN EN EL UE 2	116
FIGURA 131. CALLFLOW DE SESSION ESTABLISHMENT PARA NETWORK SLICING	117
FIGURA 132. PAQUETE DE SESSION ESTABLISHMENT ENTRE EL SMF1 Y EL UPF1	117
FIGURA 133. PAYLOAD RELACIONADO AL SESSION ESTABLISHMENT PARA EL SMF1	117
FIGURA 134. PAQUETE DE SESSION ESTABLISHMENT ENTRE EL SMF2 Y EL UPF2	118
FIGURA 135. PAYLOAD RELACIONADO AL SESSION ESTABLISHMENT PARA EL SMF2	118
FIGURA 136. PARÁMETROS DE CONFIGURACIÓN DEL GNB	118
FIGURA 137. CÓDIGO DE EJECUCIÓN DEL GNB	119
FIGURA 138. PARÁMETROS DE CONFIGURACIÓN DEL UE	119
FIGURA 139. CÓDIGO DE EJECUCIÓN DEL UE CON EL SLICE DIFFERENTIATION 000002	119
FIGURA 140. PERCEPCIÓN DE LOS ESTUDIANTES RESPECTO AL LABORATORIO	120
FIGURA 141. RENDIMIENTO ACADÉMICO DE LOS ESTUDIANTES	120
FIGURA 142. DIFICULTADES ENCONTRADAS POR LOS ESTUDIANTES	121
FIGURA 143. SUGERENCIAS BRINDADAS POR LO ESTUDIANTES COMO APLICACIÓN	121

GLOSARIO

Término	Descripción
3GPP	3rd Generation Partnership Project
4G	4th Generation Mobile Communication
5G	5th Generation Mobile Communication
5GC	5th Generation Core
AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
AMF	Access and Mobility Function
API	Application Programming Interface
APN	Access Point Name
AUSF	Authentication Server Function
CNA	Cloud Native Application
CNF	Containerized Network Function
COTS	Commercial Off-The-Shelf
CUPS	Control and User Plane Separation
CU	Centralized Unit
DHCP	Dynamic Host Configuration Protocol
DCAE	Data Collection, Analytics, Events
DN	Data Network
DU	Distributed Unit
EBC	Estaciones Base Celulares
EM	Element Management
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
eNB	Evolved Node B

eMBB	Enhanced Mobile Broadband
GTP	GPRS Tunneling Protocol
GTPC	GPRS Tunneling Protocol - Controlplane
GTPU	GPRS Tunneling Protocol - Userplane
gNB	Next Generation Node B
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HSS	Home Subscriber Server
IETF	Internet Engineering Task Force
IA	Inteligencia Artificial
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ITU	International Telecommunication Meaning
LTE	Long Term Evolution
MAC	Media Access Control
MEC	Mobile Edge Core
ML	Machine Learning
MME	Mobility and Management Entity
MTC	Ministerio de Transporte y Telecomunicaciones
mMTC	Massive Machine Type Communication
NAT	Network Address Translation
NGC	Next Generation Core
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NGAP	Next Generation Application Protocol

NRF	Network Repository Function
NS	Network Service
NSSF	Network Slice Selection Function
OSS/BSS	Operation Support System/Business Support System
OSM	Open Source MANO
PAT	Port Address Translation
PCF	Policy Control Function
PCRF	Policy Control and Charging Rules Function
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDU	Protocol Data Unit
PNAF	Plan Nacional de Atribución de Frecuencias
PTFS	Plan Técnico Fundamental de Señalización
QUIC	Quic UDP Internet Connection
RAN	Radio Access Network
RIC	RAN Intelligent Controller
RLC	Radio Link Control
RO	Resource Orchestrator
RT	Real Time
S1AP	S1 Application Protocol
SBA	Service Based Architecture
SBI	Service Based Interface
SCP	Service Communication Proxy
SDN	Software Defined Networks
SGW	Serving Gateway
SGWC	Serving Gateway-Controlplane
SGWU	Serving Gateway-Userplane

SIM	Subscriber Identity Module
SM	Service Management
SMF	Session Management Function
SNR	Signal to Noise Ratio
SO	Service Orchestrator
SS7	Signaling System 7
SSH	Secure Shell
TLS	Transport Layer Security
UDM	Unified Data Management
UDR	Unified Data Repository
UI	User Interface
UPF	User Plane Function
URLLC	Ultra Reliable Low Latency Communication
USRP	Universal Software Radio Peripheral
VIM	Virtual Infrastructure Manager
VLR	Visitor Location Register
VNFM	Virtual Network Function Management
VNF	Virtual Network Function
VNFCA	VNF Configuration and Abstraction
VoLTE	Voice Over LTE
VoNR	Voice Over New Radio
xApps	O-RAN Compliant Applications
YAML	Yet Another Markup Language

INTRODUCCIÓN

Nuestro país ha adoptado la tecnología 5G de forma tardía en comparación con otros países de la región. Esta problemática reside en diversos factores, entre ellos: (i) la falta de infraestructura necesaria a nivel de transporte, acceso y *core* (lo que dificulta el cierre de la brecha de cobertura 4G primero, y 5G después); (ii) la falta de un marco regulatorio debidamente estructurado (y acorde con los estándares propuestos por referentes de la industria como el 3GPP) que facilite a los operadores implementar redes 5G, (iii) el déficit de capital humano que cuente con los conocimientos técnicos (tanto teóricos como prácticos) que permitan diseñar, desplegar y operar la tecnología 5G de manera efectiva.

Hoy en día los desarrollos innovadores en redes móviles han introducido nuevas características tales como la capacidad de integrar tecnologías API REST como parte de la arquitectura de una red móvil (SBA), reducir la latencia en el tráfico de los usuarios gracias a la separación del plano de control y el plano usuario (CUPS), exponer interfaces comunes y abiertas para la gestión y monitoreo de la red de forma óptima y escalable (Open RAN), sacar provecho a la infraestructura *baremetal* volviéndola más flexible de acuerdo a las necesidades de la red mediante la orquestación de funciones de red, entre otros; con el fin de tener una reducción en costos de capital (*Capex*), tener una mejor gestión de los componentes de una red móvil y con ello la reducción de costos de operación (*Opex*) y brindar una mejor experiencia al usuario. Es por ello que existe una necesidad, respecto al último punto del párrafo anterior (déficit de capital humano) y ante el panorama antes expuesto, de contar con un personal debidamente capacitado en los conocimientos tanto prácticos como teóricos de forma que se pueda sacar el máximo provecho a estas nuevas características.

Es por ello que la finalidad de este trabajo se centra en abordar la solución que haga frente a carencia educativas por medio de la generación de entornos educativos virtualizados. Para ello, primero, es necesario definir los casos de estudio más relevantes tanto para tecnologías 4G y

5G que permita el estudio de los procedimientos operativos base a nivel de RAN como de *core*, así como el estudio de tecnologías emergentes como el caso de Open RAN y sus aplicaciones como el *Near Real Time RAN Intelligent Controller* (Near RT RIC) o las nuevas características introducidas por 5G como *network slicing*, arquitectura basada en servicios (SBA) o la separación del plano de control y plano de usuario (CUPS por sus siglas en inglés).

Una vez definido los casos de estudio de forma conceptual, la siguiente etapa es la selección de herramientas tanto de *core* y RAN que cumplan con los estándares definidos por 3GPP y O-RAN Alliance de forma que se garantice alta fidelidad en los resultados obtenidos en cada uno de los casos de estudio, pero también brinden flexibilidad para asegurar el aprovechamiento de la infraestructura virtual en la implementación, mediante distintos perfiles de carga según objeto de estudio (p.e pasar de un *core* desagregado idóneo para el estudio de la señalización interna del *core* a un *lightweight core* monolítico que permita el funcionamiento de la RAN para el estudio de la integración con un *near RT RIC*).

Por su parte, en la implementación, es necesario, primero, el acondicionamiento de los elementos de red (virtualizados) configurados de acuerdo al experimento a realizar y exponiendo las interfaces de interés para el estudio de los procesos, preparando las imágenes de máquinas virtuales portables los cuales se desplegarán en la infraestructura de cómputo compartida usando gestores de infraestructura disponibles, y segundo, la gestión de acceso remoto de los usuarios (p.ej., estudiantes) de forma que les brinde la visibilidad necesaria para el estudio de los procesos relacionados según el caso de estudio.

Finalmente, para demostrar que la implementación y despliegue han cumplido con su propósito, la validación de estos escenarios resulta crucial de forma que se evidencie el logro de los objetivos educacionales de los escenarios y, así, contribuir con la producción de profesionales debidamente capacitados que puedan hacer frente a la problemática antes expuesta y que esta

propuesta se muestre como solución alternativa y económica frente a productos (*testbeds* académicos dedicados) *Commercial Off-The-Shelf* (COTS) de alto precio como Amarisoft y sus productos mostrados posteriormente en las figuras 33, 34 y 35.

Siguiendo el enfoque de la metodología *design thinking* para el desarrollo de esta solución, se desarrollaron 5 capítulos de los cuales, en el primero se tenga un estudio profundo de la problemática tomando como referencia reportes globales, regionales y nacionales: exponiendo así los 3 aspectos claves que influyen en la adopción de 5G en el Perú. Así, teniendo esto en consideración, se definieron los objetivos que deberá cumplir la solución.

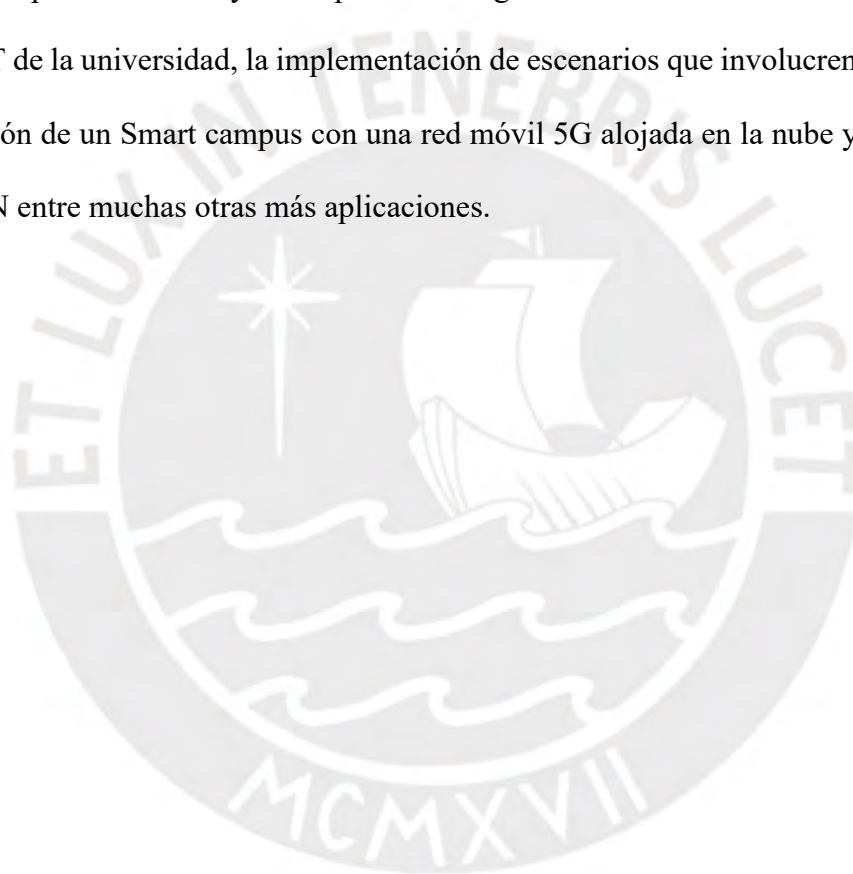
En el segundo capítulo, se presenta el estado del arte, donde se muestran aspectos básicos de una red 4G y 5G, tanto a nivel de componentes, arquitectura y esquemas de funcionamiento.

Por otra parte, en este capítulo se menciona paradigmas útiles y nuevas tecnologías de forma que se exprese la importancia que tienen estos en la industria y su contribución. Además, se expondrán las soluciones de *core* y de RAN que puedan estar presentes en la generación de escenarios, así como también esquemas de orquestación que se podrían tomar como referencia.

En el tercer capítulo, se muestra el diseño y selección de herramientas útiles para la generación de entornos educativos virtualizados. Asimismo, en este capítulo se realiza un análisis sobre qué paradigma resulta más conveniente utilizar para la implementación de los entornos. También, se presentan los escenarios seleccionados para la implementación y, finalmente, se discuten las limitaciones del diseño propuesto dentro de esta solución.

En el cuarto capítulo, se presenta un análisis en el que se consideran las contribuciones desde un aspecto educativo, económico, ambiental y legal de la solución. Además, se exponen las consideraciones que se tuvieron sobre implementación en Openstack, así como el acondicionamiento de las imágenes de máquinas virtuales junto con toda las herramientas necesarias y el acceso remoto provisto.

Por último, en el quinto capítulo, se realizaron pruebas de funcionamiento de los escenarios seleccionados y validaciones con respecto a las normas propuestas por O-RAN Alliance y 3GPP. Finalmente, con esta solución se logró alcanzar los objetivos planteados inicialmente de forma que esta sea capaz de generar profesionales capaces de contribuir con el despliegue y operación de las redes 5G en el Perú. Asimismo, al finalizar este trabajo, se permite la apertura de líneas de investigación orientadas a redes móviles, de manera que se continúe la línea de trabajo dejada por esta tesis ya sea para la integración de esta solución al orquestador VNRT/HAST de la universidad, la implementación de escenarios que involucren a un IMS o la implementación de un Smart campus con una red móvil 5G alojada en la nube y haciendo uso de Open RAN entre muchas otras más aplicaciones.



Capítulo 1. Entornos educativos para el estudio de redes 4G y 5G en el Perú debido a la situación actual

En el siguiente capítulo se describe la problemática, se define los requerimientos y objetivos, las causas de la investigación y desarrollo del trabajo presentado y se formula la propuesta para el diseño e implementación para entornos educativos de comunicaciones móviles.

1.1 Definición del problema y justificación

1.1.1 Situación de las redes móviles

Como se evidencia a partir de [1], a nivel mundial existe una tendencia, debido a la existencia de políticas, que fomentan la implementación a gran escala de las redes 4G, asimismo, se observa también la iniciativa de ciertos países, líderes en proyectos de despliegue tecnológico, de fomentar y desplegar redes 5G. Como ejemplo de estos países se tiene a China, el cual según [2] tenía alrededor de 1 millón de estaciones bases dentro de su área geográfica. Se puede observar claramente este comportamiento a partir de la siguiente tabla.

TABLA I . COMPARATIVA ENTRE REDES COMERCIALES 4G Y 5G

Regiones	LTE	5G	(%) 5G/LTE
África	162	19	12
Asia	140	61	44
Europa	165	113	68
Latinoamérica	129	28	22
Medio Oriente	51	23	45
Oceanía	39	9	23
EE.UU y Canadá	17	14	82
Total Global	703	267	38

Nota: Comparativa entre la cantidad de redes comerciales LTE y 5G y la fracción que estas últimas representan con respecto

a LTE (15-06-23) [1]

Por otro lado, como se evidencia en [1] a partir de [2], es que el despliegue de 4G estaría llegando a una meseta que irá decayendo debido a los requerimientos del mercado que obligan a la industria optar por el despliegue de tecnologías más competentes que suplan las necesidades de los usuarios. Adicionalmente, en la siguiente tabla se mostrará la tendencia descrita por [2] de conexiones móviles a partir de 2021 con proyecciones a 2025.

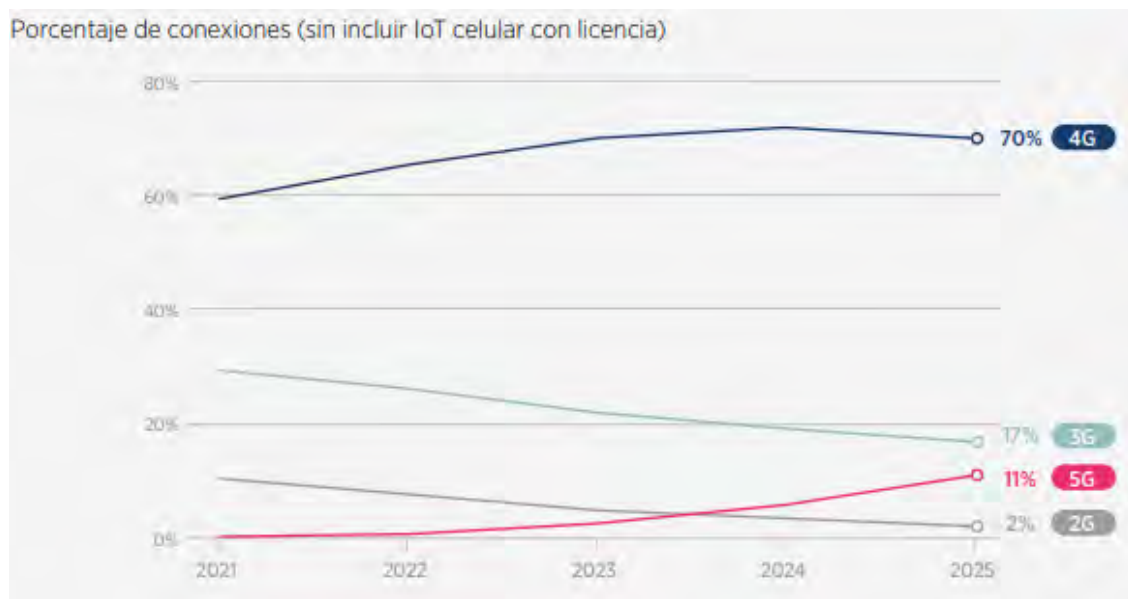


Figura 1. Tendencia de las tecnologías móviles para el año 2025 a nivel de Latinoamérica [3]

Por otro lado, [2] considera que los desafíos que implican la renovación de las actuales tecnologías desplegadas a un entorno enfocado en la red 5G residen en el entendimiento del contexto que se rige en el mercado de las comunicaciones móviles; asimismo, define la necesidad de brindar un valor cuantificable de los servicios que pueden ser provistos por las redes 5G que sirven como ente impulsador de nuevos escenarios de negocio para ciertas industrias.

1.1.2 Situación de las redes móviles en América Latina y el Caribe

Como primer punto a tratar dentro de la problemática descrita anteriormente, podemos realizar un análisis de la situación de nuestra región con respecto a las tecnologías móviles. Como principal aspecto dentro de este análisis se tiene que aún existen un alto grado de dependencia

con la tecnología de 4G, ya que aún no se consolida a nivel regional su uso, lo que dificulta la posibilidad de migración hacia 5G tal como se mencionó en la sección anterior; es por ello, no se fomentan políticas que permitan la adopción de la tecnología 5G en comparación de otros países como el caso de China o Corea del Sur.

Como se describe en [1] y se aprecia como consecuencia descrita por [3] que para el año 2025, proyección realizada en 2022, la adopción de la tecnología 5G tan solo representaría el 11% de la totalidad de conexiones existentes en la región en comparación con las diferentes regiones del mundo, de los que algunos de ellos se estima que superarían el 49% de la totalidad de conexiones móviles dentro de sus regiones, estableciendo así un promedio del 25% representado en la siguiente gráfica.

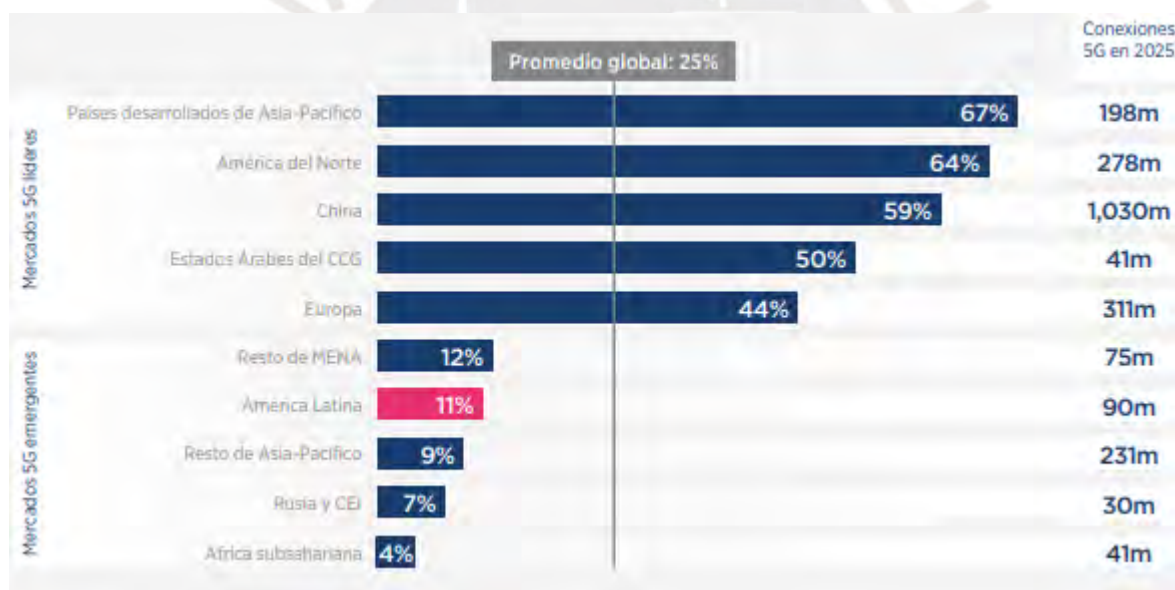


Figura 2. Adopción de 5G en 2025 [3]

En base a lo observado, nuestra región se encuentra por debajo del promedio global estimado debido a que gran cantidad de países están rezagados en cuanto a la cantidad de despliegues 5G, con respecto al despliegue masivo de redes 4G, debido a que dichos países se encuentran aunando sus esfuerzos en suplir la brecha estructural, normativa y de conocimiento necesario para el acceso de 4G por parte de los usuarios que se irá detallando posteriormente.

1.1.3 Situación de las redes móviles en el Perú

Basado en la investigación realizada en [1], se puede establecer a partir de la siguiente figura la situación con respecto a las tecnologías móviles en el Perú.

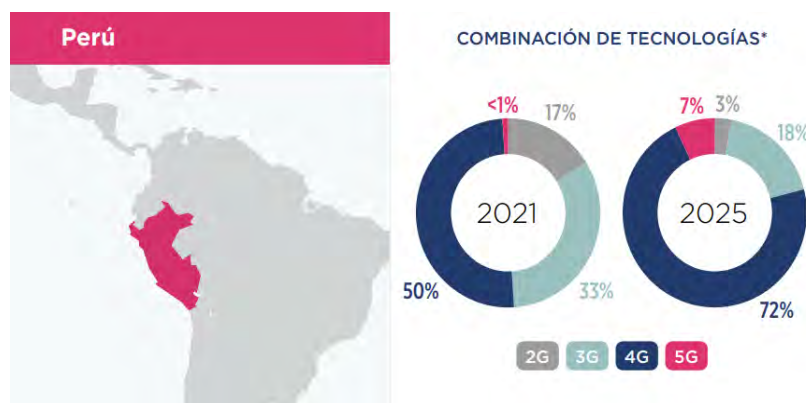


Figura 3. Uso de tecnologías móviles para el año 2021 y proyección para el año 2025 en Perú [2]

A partir de la figura 3, se aprecia, que para el 2025 el Perú se encontrará por debajo del promedio de la región en cuanto al porcentaje de conexiones basadas en la tecnología 5G; sin embargo, se evidencia también que habrá un gran crecimiento de conexiones 4G dentro del territorio nacional, lo que permitirá impulsar la captación de nuevas tecnologías, tal como es el caso de 5G, debido a la aproximación a la meseta establecida en la figura 1 lo que refleja el inicio de la transición entre una tecnología móvil y otra.

Sin embargo, [1] destaca la existencia de ciertos aspectos limitantes, tanto a nivel de la infraestructura de telecomunicaciones requerida para los despliegues, así como aspectos normativos y limitaciones debido al acceso por parte de los usuarios de las tecnologías 4G y 5G.

1.1.3.1 Limitaciones de infraestructura

A partir de [1], se puede evidenciar principalmente dos limitantes importantes ante el despliegue de redes móviles.

En primer lugar, la infraestructura necesaria a nivel de las redes de transporte y acceso en cuanto fibra óptica dentro del país tiene que permitir cumplir con los requerimientos técnicos de las tecnologías 4G y 5G. Sin embargo, en los últimos años las operadoras de servicios de telecomunicaciones y empresas de despliegue de redes ópticas están trabajando para un despliegue de fibra óptica a nivel departamental, pero dentro de estos intentos se suscitaron los siguientes inconvenientes expuestos en [1]:

- Las empresas de electricidad no aceptaron la compartición de su infraestructura y como consecuencia hubo retrasos en el despliegue
- Se presentaron inconvenientes en la adquisición de terrenos que también repercuten en retrasos
- No hubo una buena organización en cuanto a los proyectos tecnológicos previamente desplegados lo que generó la duplicidad de funciones tecnológicas, debido a esquemas de soluciones monolíticos y no *cloudificadas*. Como consecuencia hubo un sobre gasto innecesario debido a la duplicidad en funciones en base a los requerimientos técnicos contemplados.

Por otra parte, otra limitante infraestructural reside en la escasez de estaciones bases para brindar cobertura dentro del territorio nacional. Este dimensionamiento de estaciones bases faltantes serán presentadas en la siguiente tabla provistas por [1]:

TABLA II. BRECHA DE ESTACIONES BASE POR DEPARTAMENTO PARA EL AÑO 2025

Región	EBC existentes para el 2019-IV	EBC estimadas al 2025	% de incremento requerido	Brecha para el año 2025
Amazonas	350	575	64 %	225
Ancash	949	2 136	125 %	1 187
Apurímac	442	623	41 %	181
Arequipa	1 347	4 367	224 %	3 020
Ayacucho	656	981	50 %	325

Cajamarca	975	1 313	35 %	338
Cusco	1 073	2 313	116 %	1 240
Huancavelica	378	531	40 %	153
Huánuco	600	886	48 %	286
Ica	643	2 810	337 %	2 167
Junín	946	2 076	119 %	1 130
La Libertad	1 241	3 955	219 %	2 714
Lambayeque	696	2 574	270 %	1 878
Lima	9 196	25 671	179 %	16 475
Loreto	477	975	104 %	498
Madre de Dios	169	455	169 %	286
Moquegua	208	534	157 %	326
Pasco	231	314	36 %	83
Piura	1 095	2 775	153 %	1 680
Puno	867	1 609	86 %	742
San Martín	722	1 120	55 %	398
Tacna	355	739	108 %	384
Tumbes	191	520	172 %	329
Ucayali	269	919	242 %	650
Total	24 076	60 771	152 %	36 695

Nota: Información obtenida en [1]

1.1.3.2 Limitaciones normativas

Basado en lo descrito por [1], en el Perú, se tiene que los servicios habilitados por la tecnología 5G deberán respetar el PTFS normado por el MTC, en donde este último se rige principalmente por la arquitectura de señalización SS7, el cual fue propuesto por la ITU en 1981 y que permite la interconexión, el cual es usado por los operadores en sus redes de telecomunicaciones; sin embargo, la limitante reside que esta arquitectura *legacy* no se adecúa a los requerimientos para los sistemas móviles 4G y 5G, lo que representa un inconveniente al momento de promover los nuevos servicios brindados por esta última. Asimismo, [5] y [1] convienen que el estado, por

medio del MTC, debería considerar otros tipos de estándares que permitan la adopción de nuevos servicios basados en nuevas tecnologías que se basan en los estándares referentes para estas tecnologías como los propuestos por ETSI, IETF y la 3GPP, ya que son estas las que las normas y requerimientos para el funcionamiento de las nuevas tecnologías como 5G y nuevos servicios habilitados por ésta como es el caso de VoNR. Por otra parte, en [6] se detalla que la licitación de espectro para el despliegue 5G aún sigue presentando un desafío dado que los 679.16 MHz en las 3 bandas (alta, media y baja) que se tiene no resultan suficientes para los nuevos servicios habilitados por esta tecnología (teniendo en cuenta que el ancho de una portadora 5G es de hasta 100 MHz), esto debido a una falta de planificación adecuada del espectro en donde se pueda lograr los objetivos de expansión gubernamentales dentro del proceso de licitación lo que facilita los despliegues con fines de aumentar la cobertura o realizar *refarming* del espectro con el objetivo de reutilizar el espectro asignado a tecnologías que están quedando en desuso tal como es el caso del 2G según menciona [6].

1.1.3.3 Limitaciones sociales

Como se evidencia a partir del análisis de [1] y [5], existen diferentes limitantes con respecto al uso de este tipo de tecnología por parte de los usuarios finales. Dentro de las cuales, se tiene un elevado precio a nivel de los terminales para el acceso servicios de telecomunicaciones y principalmente la carencia de habilidades técnicas y de conocimiento por parte de los usuarios en las empresas en donde se puedan extraer de mejor manera las potencialidades de este tipo de tecnologías y esto principalmente debido a la falta de formación y capacitación a nivel educativo lo que repercute en una falta de personal calificado capaz de gestionar, desarrollar e innovar en estas nuevas tecnologías.

1.1.4 Situación de las tecnologías móviles enfocados en el aspecto académico en el Perú

Partiendo de lo antes mencionado, resulta de suma importancia la producción y capacitación de personal debidamente competente dentro del funcionamiento de las tecnologías 4G y 5G en el que se debe considerar, también, los nuevos paradigmas que se usan en la industria, por lo que instituciones educativas superiores como la Pontificia Universidad Católica del Perú debe estar en la capacidad de poder formar profesionales competentes en estas tecnologías y que, por medio de estos, pueda contribuir a cerrar la brecha de conectividad 4G y 5G expuesta anteriormente que se tiene en el país. Por lo que este enfoque deberá estar presente en los cursos de comunicaciones móviles, puesto que es allí donde se adquieren los conocimientos relacionados a estas tecnologías; pero que, sin embargo, se observa una ausencia de infraestructura que permita a los estudiantes realizar prácticas vivenciales lo cual permite la adquisición de conocimientos empíricos que se complementen con el contenido teórico brindado en el curso. Como consecuencia de esto, resulta necesario la generación de escenarios virtuales de pruebas que permitan contribuir, de forma concreta, con este fin; asimismo, permita a las instituciones educativas superiores, tal como es el caso de la PUCP, contribuir en la formación de profesionales altamente competentes en el campo de las comunicaciones móviles y así poder contribuir con el desarrollo del país y su población.

1.2 Definición de requerimientos

Debido a lo establecido en la sección anterior, el presente trabajo deberá tener como finalidad la generación de entornos educativos virtuales que aborden los principales casos de uso de las tecnologías 4G y 5G. Asimismo, en el aspecto educativo, deberá demostrar que el presente trabajo permita la captación y aprendizaje de nuevos paradigmas tal como es el de las arquitecturas desagregadas, el *core* SBA y el uso de 4G y 5G basado en tecnologías de virtualización que permitan a los estudiantes adquirir el conocimiento necesario para desarrollar

soluciones que agilicen los despliegues de redes 4G y 5G en el país, así como la capacidad de generar innovación en ese sentido. A nivel de implementación, se deberá considerar la capacidad de integración con la infraestructura compartida disponible (p.ej., la nube privada de la sección de Ingeniería de las Telecomunicaciones de la PUCP, basada en OpenStack) y la capacidad de poder aprovechar los recursos de forma flexible manejando distintos perfiles de operación, tomando en cuenta los escenarios de estudio y garantizando el reuso de infraestructura y la escalabilidad, por medio de orquestadores de forma que habiliten la automatización en la generación de escenarios. Por otra parte, desde el punto de vista económico, en el contexto peruano la mayoría de las instituciones educativas carecen de los recursos económicos suficientes para poder realizar la adquisición de soluciones COTS dedicadas, como es el caso de las soluciones de Amarisoft, las cuales permiten lograr los objetivos de aprendizaje relacionados a estas tecnologías; es por ello que, desde ese punto de vista, la generación deberá ser económicamente rentable en comparación con las soluciones COTS.

1.3 Objetivos

1.3.1 Objetivo principal

El objetivo de este trabajo es la generación de entornos de simulación y emulación con un enfoque académico para las tecnologías 4G y 5G en conjunto con un micro-orquestador que permita la automatización en la producción de escenarios. Se hará uso extensivo de herramientas *open source* tanto de *core* como de acceso de radio (Radio Access Network, o RAN) que permitan (i) La configuración de parámetros de red *core* y RAN de forma interactiva así como también el entendimiento de los elementos que componen cada una de estas soluciones desde la visualización de los procesos para su estudio así como la disponibilidad de código para su optimización en un futuro (ii) Permite la reducción del costo en la implementación de cada

uno de los escenarios sin afectar/perder la fidelidad en los resultados obtenidos esto según las políticas de *compliance* que cada una de estas herramientas abiertas tienen con respecto a los estándares (*Technical Specification*) o *releases* definidos por 3GPP y O-RAN Alliance, esta última es una organización internacional que trabaja para la estandarización y desarrollo de arquitecturas de red de acceso abierto y que impulsa la innovación y la interoperabilidad entre distintos proveedores para las redes móviles actuales. La generación de estos entornos conlleva (i) la selección de escenarios de red suficientemente diversos para lograr el estudio de los paradigmas disruptivos clave de 5G, como es el caso de Open RAN, CUPS (separación de las funciones de red *core* 5G para plano de control y plano de usuario) y *Network Slicing* (la coexistencia de distintos modos de operación soportados en una misma red física mediante la creación de redes virtuales independientes y personalizadas según el tipo de servicio), (ii) el preparado de las imágenes (de máquinas virtuales) que implementan las funciones de red requeridas en cada escenario, tomando en consideración el “*trade off*” entre visibilidad y consumo de recursos adecuado al objetivo educativo de cada escenario, y (iii) el despliegue (por parte del micro-orquestador desarrollado) de estos entornos educativos dentro del HAST/VNRT, el cual es la nube privada de la sección de ingeniería de las telecomunicaciones de la PUCP.

1.3.2 Objetivos específicos

Para completar el objetivo principal, se definen los siguientes objetivos:

- Definición de los entornos de prueba según los casos de uso más relevantes que permitan la obtención de los conocimientos necesarios para tecnologías 4G y 5G
- Selección de las herramientas de *core* y RAN para la creación de los entornos educativos
- Integración de las herramientas de *core* y RAN con el micro orquestador mediante el uso de APIs.

- Acondicionamiento de los componentes de los entornos educativos con herramientas que permitan a los estudiantes el análisis en el funcionamiento de estas.
- Validación del correcto funcionamiento de los escenarios implementados a través de planes de prueba.



Capítulo 2. Análisis de las herramientas y paradigmas útiles para el diseño e implementación de entornos educativos 4G y 5G

2.1 Nuevos paradigmas en la industria 4G y 5G

Es importante destacar los nuevos paradigmas en la industria de las tecnologías móviles, dado que habilitan nuevos escenarios y posibilidades en cuanto al despliegue y la integración con otro tipo de tecnologías que optimicen su funcionamiento, por lo que dentro del alcance del presente trabajo se desarrollará las ventajas que introducen estos paradigmas y la importancia que tienen estos en cuanto a la generación de entornos educativos.

2.1.1 Arquitecturas desagregadas

Las arquitecturas desagregadas, tanto para el *core* de una red móvil como para la RAN, toman un rol importante dado que gracias a la desagregación de sus componentes pueden tener prestaciones como la reducción de la latencia, integración con otro tipo de servicios, personalización de la arquitectura de la red, de acuerdo con las necesidades que se tengan tal como se mostrará a continuación.

2.1.1.2 Arquitectura desagregada de la red core

La arquitectura desagregada a nivel de las funciones del *core*, tal como detalla [7] y [8], presentan las siguientes ventajas más resaltantes:

1. CUPS: Hace referencia a la separación entre el plano de control y el plano de usuario el cual habilita casos en los que se pueda tener los UPFs más cercanos al *edge* de forma que se reduzca la latencia a nivel del tráfico de aplicaciones con un enfoque MEC (*Mobile Edge Core*). Esta desagregación se puede apreciar en la siguiente imagen.

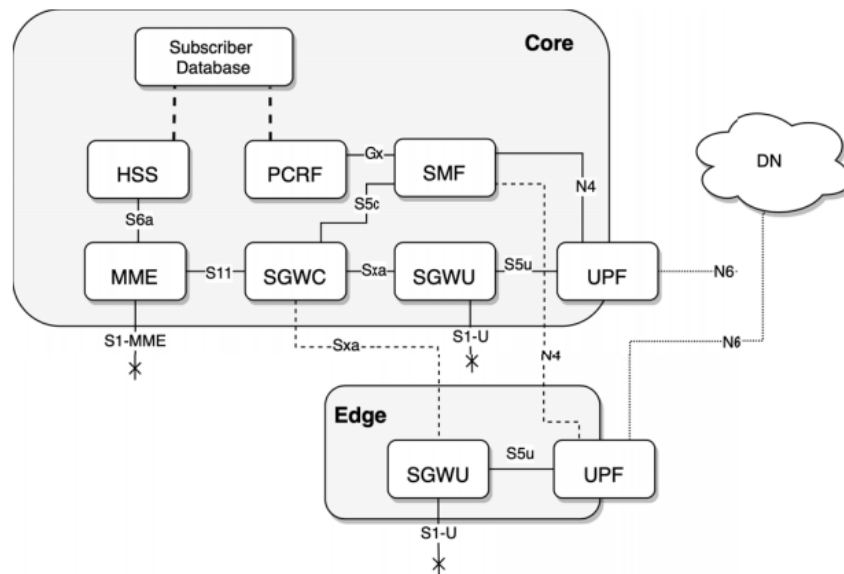


Figura 4. Red core desagregada [7]

2. Funciones principales personalizadas: Dentro de esto se tiene la posibilidad de colocar funciones de *core* ideadas para requerimientos específicos que permitan escalar a nivel de performance el funcionamiento de la red tal como detalla [8]; como consecuencia se tiene la capacidad de escalar en función de los recursos económicos y de infraestructura que el operador tenga.
3. Interoperabilidad entre funciones del *core*: Dado que se puede tener distintos proveedores de funciones *core*, cabe destacar que la desagregación de la red *core* se rige por estándares comunes por la 3GPP lo que garantiza la interoperabilidad de estos.

2.1.1.3 Arquitectura desagregada de RAN

La desagregación de la red de acceso por otra parte se fundamenta principalmente en la división funcional del gNodeB a nivel de CU y DU, especificada en la 3GPP tal como muestra [9].

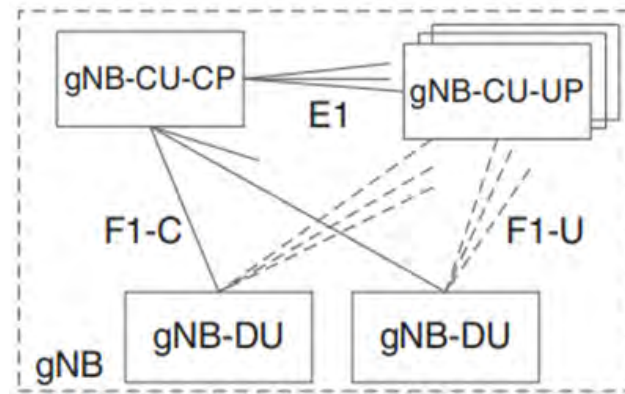


Figura 5. Separación del plano de usuario y el plano de control a nivel de DU y CU [9]

Adicionalmente a ello, la O-RAN Alliance propuso la división funcional a nivel del DU para introducir el RU que permitiría el procesamiento de capas inferiores, considerando principalmente a nivel de RF tal como detalla [10] y, además, se puede visualizar en la siguiente figura:

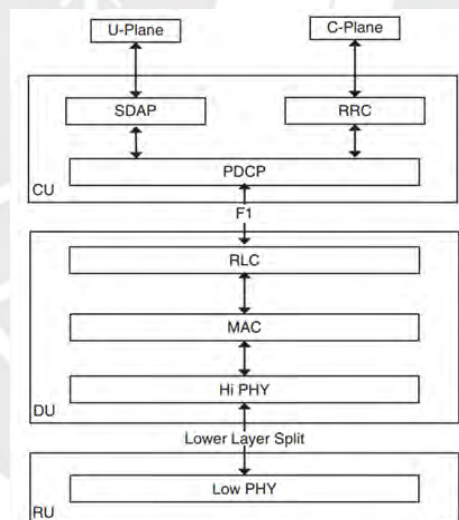


Figura 6. Componentes de una arquitectura Open RAN [21]

Finalmente, con estas divisiones, se habilitan las siguientes prestaciones a nivel de la RAN indicadas por [11] y [12]:

- Aprovechamiento de los recursos provistos por equipos *baremetal*: Existe la capacidad de virtualización de DU y CU que permitan la abstracción de sus funcionalidades a nivel de software y que permita el uso de equipos *baremetal* como host de estas funciones, lo que permite tener un segmento de la red virtualizada

- Esquema *multivendor*: Dentro de un marco común de protocolos definidos por la 3GPP y O-RAN Alliance se garantiza la interoperabilidad de cada uno de estos componentes lo que habilita que estos sean de distintos *vendors* evitando así el *vendor lock-in*, el cual genera una relación de dependencia con un solo proveedor, por lo que genera un entorno más competitivo y además la capacidad de tener una RAN altamente personalizable.
- Integración con servicios de ML e IA: Open RAN expone interfaces E2 que permite el control y manejo de los recursos de radio, QoS, *slices* presentes en la pila de protocolos de la red de acceso por parte de un RIC (RAN *Intelligent Controller*) que integra xApps las cuales son componentes de software que integran servicios de ML e IA.

2.1.2 Arquitectura basada en servicios

La arquitectura de una red basada en servicios (SBA) de una red móvil, tanto a nivel de la red core como a nivel de RAN, se fundamenta bajo los siguientes conceptos y herramientas según detalla [13] y [14]:

- NFV: La virtualización de funciones de red cumple un rol crucial dado que gracias a la abstracción de la funcionalidad de los componentes monolíticos tanto de *core* como de RAN se puede tener la administración de una red basada en funciones virtuales que trabajan en un equipo *baremetal* multipropósito lo que permite tener una mayor flexibilidad, eficiencia y adaptación en cuanto a la orquestación y manejo de los componentes o funciones de red en función de la aplicación, demanda o requerimiento.
- SDN: Se basa en la separación a nivel de plano de control y plano de usuario, lo que habilita la creación de funciones de red y *core* independientes según el plano en la que estas se encuentren (ya sea de control o de usuario).
- HTTP: Es el protocolo que permite la comunicación entre los componentes de red y asume el rol de interfaces basada en servicios (SBI). Para esto, HTTP utiliza la

arquitectura REST APIs para la comunicación entre estos componentes y también para su monitoreo y gestión. Por otra parte, a nivel de seguridad la comunicación entre dichos componentes trabaja con TLS y, además, dado que HTTP es un protocolo que tiene un alto nivel de desarrollo debido a su amplio uso, permite la articulación con nuevas tecnologías tal como es el caso de QUIC (*Quick UDP Internet Connection*) lo que permite una mayor eficiencia y rendimiento en cuanto a la transferencia de datos.

Por otra parte, representa a nivel operacional las siguientes ventajas a nivel de construcción, despliegue y gestión de redes móviles según detalla [13] y [14]

1. Flexibilidad: Permite la adaptación de la red en función de los aplicaciones y servicios, debido a que, en función de estas, los componentes que conforman la red pueden ser desplegados, actualizados o reemplazados.
2. Escalabilidad: Permite el escalamiento de cada componente de forma independiente según la demanda, lo que representa un mejor rendimiento y capacidad de respuesta de la red tanto de *core* como de RAN.
3. Reutilización: Los componentes pueden ser reutilizados en diferentes contextos y aplicaciones. Asimismo, un equipo *baremetal* al eliminarse un componente puede reutilizar sus recursos computacionales para el despliegue de otros componentes según se requiera.
4. Interoperabilidad: Al tener protocolos de comunicación para las interfaces basadas en servicios (SBIs) los componentes de la red permiten la interoperabilidad entre sí lo que facilita la integración para los despliegues y la colaboración entre diferentes proveedores.
5. Mantenibilidad: Debido a la separación de los componentes de plano de usuario y plano de control, resulta más fácil el *troubleshooting* dado que se tiene una mayor

independencia entre dichos componentes lo que facilita la corrección de errores y la mejora continua de la red.

2.1.3 MANO

MANO (o NFV-MANO) es un esquema propuesto por [15] el cual define un *framework* que permite la orquestación y gestión de la vida útil de las VNFs bajo el uso de recursos provistos por una NFVI que se traduce como la producción, mantenimiento, actualización y orquestación de funciones de red más complejas virtualizadas (comúnmente llamados NS por sus siglas en inglés, Network Service) *hosteadas* en equipos *baremetal* que se traduce en los siguientes objetivos de MANO:

- Permitir a los proveedores de servicio tener un despliegue de NSs mucho más rápido y sencillo.
- Permitir a los proveedores de servicio el abaratamiento de costos de red debido a que la abstracción que se brinda al tener NS basados en VNFs permite hacer un uso compartido de infraestructura *baremetal*.
- Permitir a los proveedores de servicio tener la capacidad de escalar sus redes basado en la demanda que se tenga mediante el *upgrade* en cuanto a los recursos provistos a un NS o a una VNF en específico por parte de la NFVI.
- Permitir a los proveedores de servicio tener un esquema de despliegue confiable en el que se tenga NSs con tolerancia a fallos o de alta disponibilidad que permita tener esquemas redundantes y de automatización de aprovisionamiento, despliegue y mantenimiento.

Para entender lo que es MANO, es importante tener en cuenta la definición y requerimientos en base a las funcionalidades provistas por MANO en [15]. Por lo que para tener de mejor manera tengamos como referencia el diagrama de arquitectura propuesto por [15] para MANO en la figura 7 tal como se muestra a continuación:

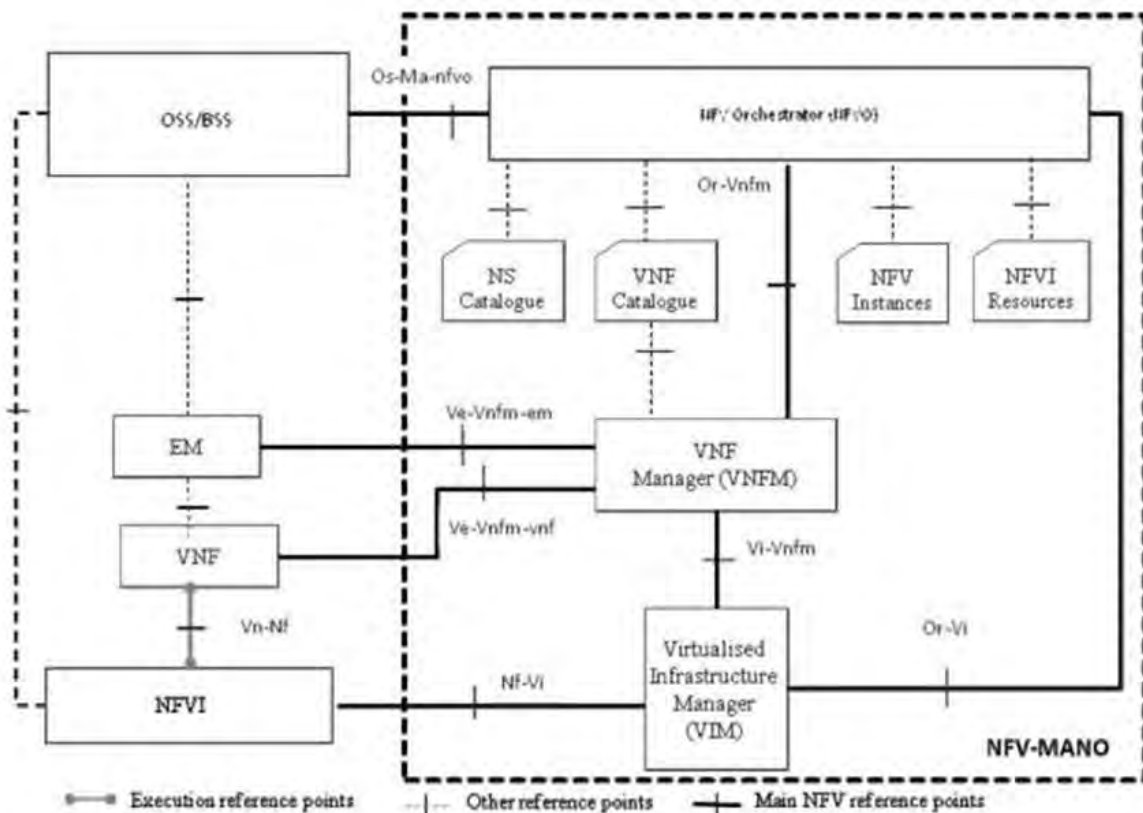


Figura 7. Arquitectura de NFV-MANO [15]

En donde según [15], NFV-MANO se compone por tres principales bloques los cuales son

1. **NFVO (Network Function Virtualization Orchestrator):** Es el componente encargado principalmente de la orquestación de distintos NS basada en la coordinación de grupos de VNFs mediante la creación, configuración (así como cambios dinámicos) tanto de las instancias como de las conexiones requeridas de las mismas de manera conjunta. En otras palabras, el NFVO es el componente encargado que dentro de todos los niveles del ciclo de vida de las VNFs éstas trabajen de manera conjunta para lograr así el funcionamiento de los NS.
2. **VNFM (Virtual Network Function Manager):** Es el encargado de gestionar a nivel de VNF el ciclo de vida de la VNF lo que comprende el aprovisionamiento de recursos, despliegue, configuraciones, monitoreo, escalamiento y mantenimiento. Por lo que se puede

entender, en otras palabras, que el VNFM es el componente encargado de cerciorarse que a nivel de VNF estas trabajan como deberían dentro de todas las etapas del ciclo de vida de esta.

3. VIM (*Virtual Infrastructure Manager*): Es el encargado de controlar y gestionar los recursos de la NFVI que comprenden cómputo, almacenamiento y red. Lo que implica que este componente lleve un inventario de los recursos que se tienen en la NFVI y los que se puedan brindar en función de lo que el VNFM o el NFVO requieran.

Por otra parte, [15] también define que NFV-MANO tenga su información distribuida en los siguientes repositorios:

1. NS Catalogue: Es un repositorio de los NS implementados hasta el momento que, principalmente, contiene información tal como la descripción del NS (la funcionalidad del NS, sus requerimientos y sus dependencias), descripción acerca de las VNFs que componen al NS. Ello permite el respaldo de la creación y la gestión de plantillas para la replicación de NSs.
2. VNF Catalogue: Es un repositorio de las VNFs implementadas hasta el momento que al igual que el NS Catalogue permite la creación de paquetes VNF que contiene imágenes de software, archivos de manifiesto entre otras configuraciones necesarias para el despliegue de VNF.
3. NFV Instances Repository: Es un repositorio que cumple principalmente la función de un registro tanto de NS como de NFV que muestra el estado en el que se encuentran durante su ciclo de vida.
4. NFVI Resources Repository: Es un repositorio que muestra la información acerca de los recursos NFVI disponibles, reservados y asignados provistos por el VIM ya sea desde un dominio de infraestructura como ya sea solo de una.

Por último, [15] define los bloques que comparten puntos de referencia externos con MANO:

1. EM (Element Management): Es el componente responsable de la gestión de la detección y reporte de fallas de las VNFs, configuración, recolección de la data usada por esta, así como la elaboración de sus reportes de recursos usados, monitorea el nivel de performance y el nivel de seguridad de esta.
2. Operations Support System/Business Support System (OSS/BSS): Se entienden como los componentes que permiten recibir información de los EM propios de los proveedores de servicio que permitan realizar un esquema de administración de la red a nivel de atención al cliente, facturación, aprovisionamiento y monitoreo de la red a nivel de los usuarios de la empresa.
3. NFVI (Network Function Virtualization Infrastructure): Se entiende como el componente que abarca todo el hardware a nivel de recursos tales como cómputo, almacenamiento, networking e hipervisores que son provistos en el despliegue y upgrade de VNFs.

2.1.4 CNF

El CNF, según [16], hace referencia a la función de red basada en la nube o comúnmente a la función de red basada en contenedores y es que tal como se pudo apreciar en la sección anterior la virtualización juega un rol importante en lo que es el desarrollo actual en cuanto a las tecnologías que habilitan el despliegue y gestión de una red móvil. Sin embargo, debido al aumento en cuanto a la capacidad que deba tener la red, así como la generación de nuevos servicios y a la necesidad de permitir el desarrollo continuo, las VNF's presentan serias dificultades para poder cumplir con los siguientes desafíos:

- Tienen un grado limitado de escalabilidad en función de la demanda.
- Presentan una sobrecarga de hardware debido a que estas requieren de un hipervisor.
- Al tener máquinas virtuales el despliegue se vuelve más lento.

Ante esta problemática, según menciona [16], las CNF resultan ideales para solucionar estos aspectos, puesto que al usar contenedores garantiza lo siguiente:

- Empaqueta todos los recursos necesarios a nivel de archivos para las aplicaciones en específico lo que lo hace altamente portable y fácil de desplegar.
- Comparte el kernel del sistema operativo de forma aislada lo que permite ser ligero, aislado de otros componentes y seguro para hacer *troubleshooting*
- Permite la replicación y escalamiento en función de la demanda

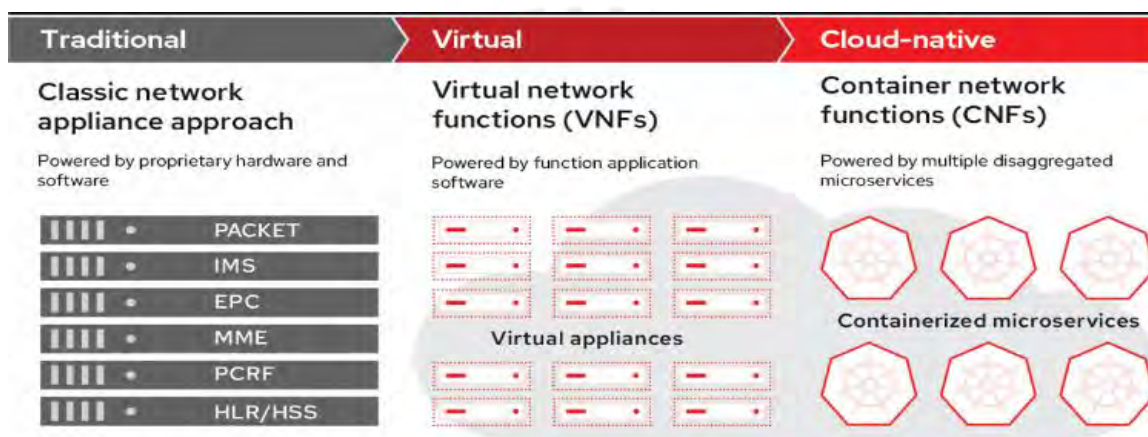


Figura 8. Evolución de las funciones de red móvil [15]

2.1.5 CNA (Cloud-Native Architecture) 5G/4G

Una arquitectura basada en la nube tanto para 4G como para 5G según detalla [17] se basa en un enfoque de construcción y ejecución de aplicaciones en el que se tiene como referencia el aprovechamiento del modelo de computación en la nube. Para esto [17] detalla los siguientes puntos clave:

1. Contenedores: Las aplicaciones o CNF se empaquetan en contenedores, los cuales al ser aislados garantizan la ejecución de forma independiente entre otros procesos lo que facilita el *troubleshooting*, el aprovechamiento de los recursos del sistema por medio de la compartición del kernel, la escalabilidad debido a la generación múltiples instancias y el despliegue rápido.

2. Microservicios: Las aplicaciones se dividen en componentes más pequeños y autónomos lo que permite un desarrollo y despliegue modular y ágil. Por otra parte, permite CI/CD (*Continuous Integration/Continuous Deployment*) y una tolerancia a fallos mayor debido a la independencia en cuanto a la estructura de microservicios que se pueda tener.
3. Orquestación y automatización: Se utiliza un orquestador que permita gestionar y escalar de forma automática los contenedores junto con los microservicios a nivel de instancias y recursos computacionales con un enfoque basado en MANO.
4. DevOps: Permite la integración de profesionales de TI que permitan bajo un mismo enfoque la automatización en cuanto al despliegue y aprovisionamiento de servicios. Por otra parte, permite a los desarrolladores la integración de nuevas tecnologías basadas en software que permita la gestión, control y automatización en el proceso de despliegue y uso.

Todo lo anteriormente mencionado se puede apreciar en la siguiente imagen en el que se articula todos estos puntos y muestran la siguiente arquitectura nativa en la nube.



Figura 9. Arquitectura basada en la nube de un core móvil [15]

Como consecuencia de esto, según detalla [17], se tiene una transformación *cloud-native* en la industria puesto que permite a las organizaciones aprovechar al máximo el potencial de la nube

(ya sea pública o privada), mejorando así las funcionalidades a nivel de desarrollo de software, la eficiencia operativa y la reducción de costos. Por otra parte, permite a los proveedores de servicios de comunicaciones brindar servicios con una mejor calidad y ofrecer nuevos servicios de una manera más ágil y dinámica.

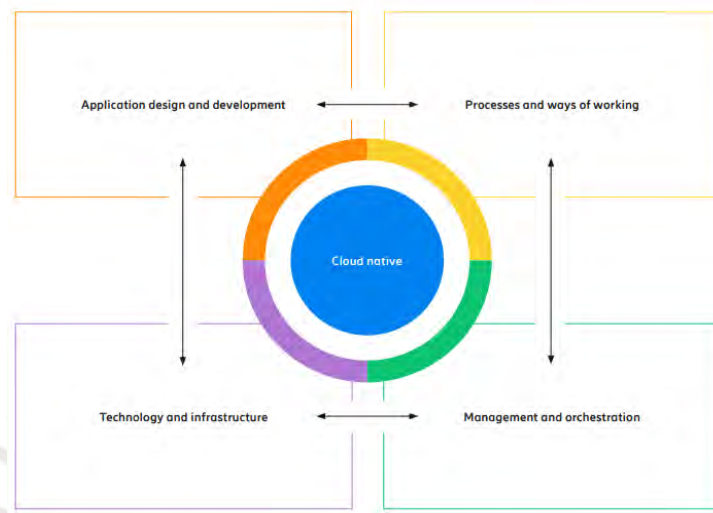


Figura 10. Aspectos fundamentales de una arquitectura *cloud native* [17]

2.2 Herramientas útiles para entornos educativos 4G y 5G

Dentro de las herramientas útiles para el entorno educativo se tiene herramientas *open source* tanto de *core* como RAN que cumplan dicha labor. Es por ello que en esta sección se tomará como referencia el análisis de cada una de estas herramientas para analizar las ventajas y desventajas de las mismas a nivel de performance en operación. Por otra parte, se realizará el análisis de componentes de orquestación que permitan la elaboración de entornos educativos a nivel de ventajas y desventajas.

2.2.1 Análisis comparativo de herramientas RAN y Core

Para esto tomaremos el análisis realizado por [1] para dichas herramientas sin entrar a profundidad en el funcionamiento, pero sin perder de vista los aspectos generales de las mismas.

2.2.1.1 Open 5GS

Según la definición brindada por [19], Open5GS es un *core 4G/5G open source* que adopta un esquema SBA a nivel de las funciones de *core*, se encuentra definido en lenguaje C y presenta la siguiente estructura:

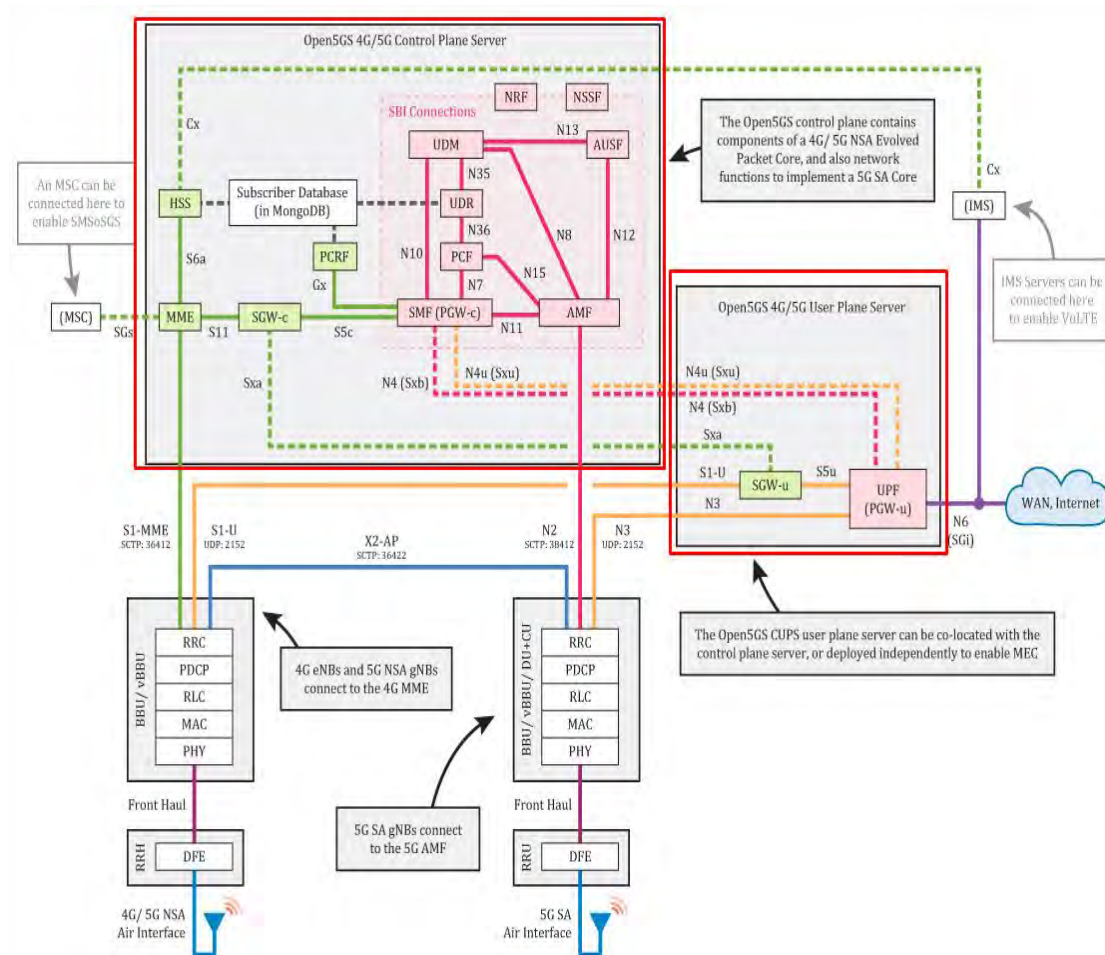


Figura 11. Componentes de Open5GS [19]

Dentro de sus características más relevantes del mismo se tiene que es 3GPP *compliance* con el *release 17* que define los lineamientos de *core*. Por otra parte, permite la integración con componentes como es el caso de IMS que permite servicios como VoLTE. Además, permite el *handover* lo que permite la gestión de movilidad.

2.2.1.2 OAI Core

Asimismo, según [20], se tiene la alternativa de Eurocom que implementa de forma separada, a diferencia de Open5GS, dos proyectos que son el OAI-EPC para 4G y el OAI-5GC para 5G tal como se muestra a continuación:

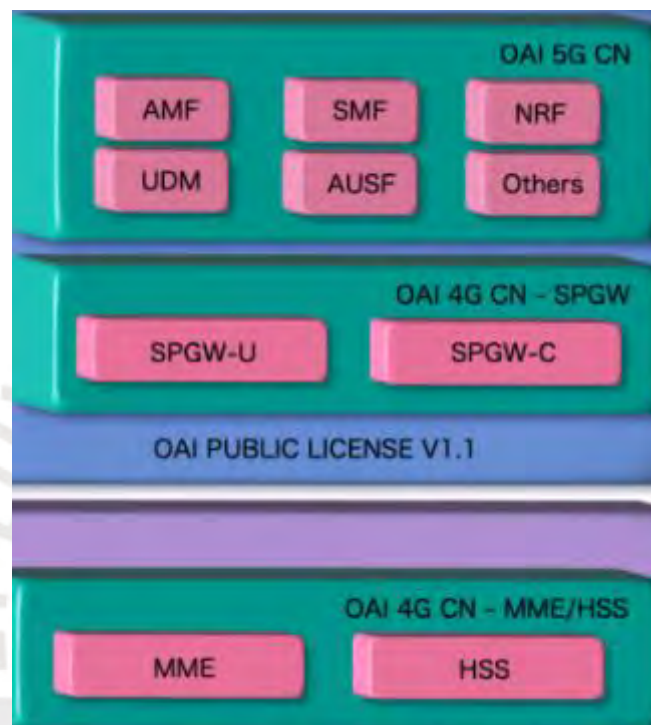


Figura 12. Componentes de OAI-Core [20]

Dentro de sus características más importantes, para el caso del EPC, se tiene soporte de todos los procedimientos de registración, autenticación, service request e implementación de seguridad a nivel de NAS con el algoritmo AES. Por su parte, el 5GC o NGC, al igual que en el caso del EPC, establece todos los procedimientos antes mencionados, así como la comunicación de los componentes vía HTTP con un enfoque SBA.

2.2.1.3 Magma

Por su parte, la Linux Foundation con la colaboración de Facebook, propone Magma como una alternativa de *core* EPC con una propuesta bastante particular tal como se logra apreciar en la arquitectura en la siguiente imagen

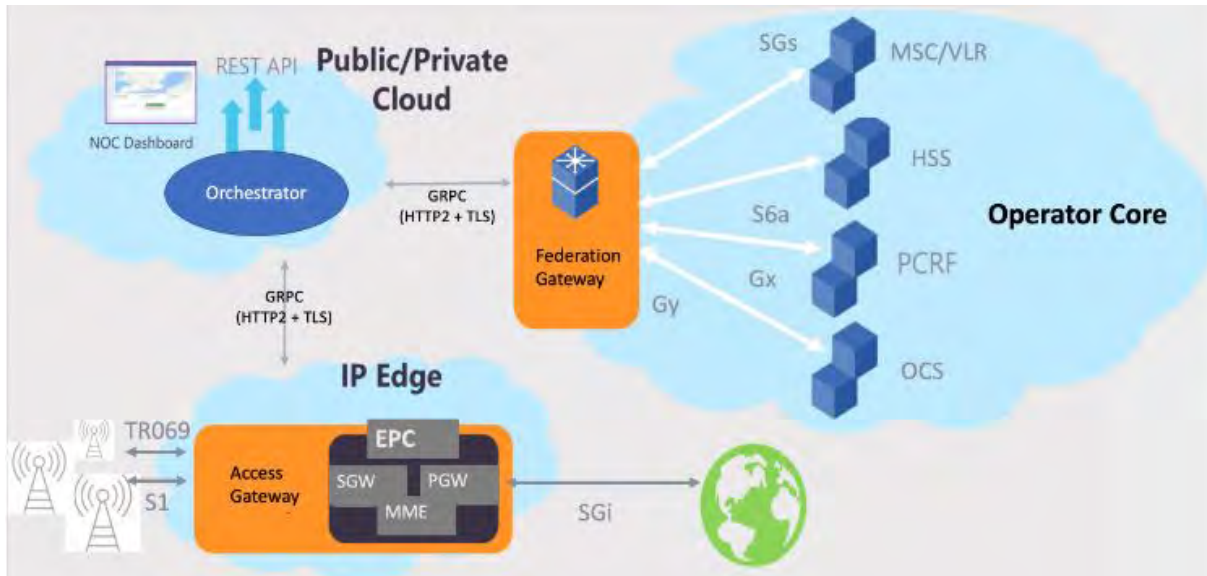


Figura 13. Componentes de Magma [21]

En donde según [21] se tiene 3 componentes fundamentales en su arquitectura los cuales son el Access Gateway, el cual es el componente encargado de implementar las unidades básicas del EPC más cercana al *edge* que permita reducir la latencia y habilitar despliegues distribuidos por IP EDGE. Por otro lado, se tiene el orquestador que permite la gestión y control de la red entre los componentes del *federation gateway*. Por último, se tiene al *federation gateway*, el cual se encarga de la integración de componentes como el HSS, PCRF y VLR entre otros con los IP EDGE. También, es importante notar que no presenta una arquitectura normada por la 3GPP, pero que sin embargo respeta algunas interfaces como las que se encuentran presentes en el *federation gateway* o la salida a la red.

2.2.1.4 srsEPC

Por último, del lado de SRS se tiene su opción para EPC, según [22] se define como un *core* EPC ligero con una arquitectura de los componentes básicos de un *core* 4G:

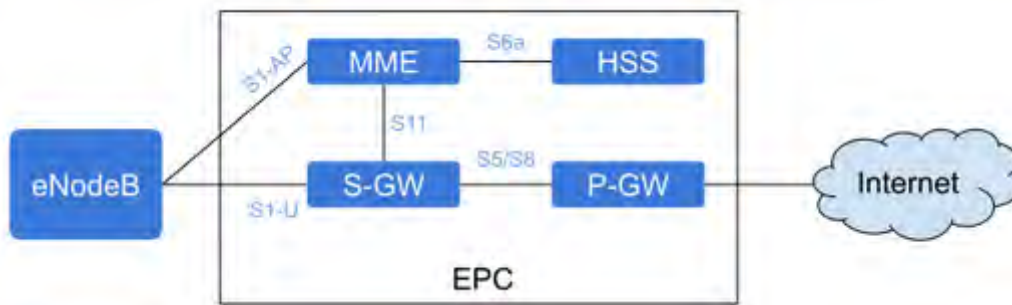


Figura 14. Componentes de srsEPC[22]

Donde entre sus características más resaltantes se tiene el soporte de NB-IoT, características que le permiten la autenticación, registro y *service request*.

2.2.1.5 Análisis comparativo de las soluciones de *core*

Tomando como referencia estas 4 herramientas a nivel de *core*, a continuación, basado en el análisis de los factores que permiten el análisis de una alternativa con respecto a otra basado en la comparativa provista por [1] en donde se realizó basado en los siguientes factores:

1. *3GPP Compliance*: Las soluciones de red *core* deben ser capaces de operar de acuerdo con *releases* definidas por la 3GPP tanto para 4G como para 5G.
2. Soporte 4G/5G: Permite que una solución brinde servicios de 4G como 5G de forma simultánea.
3. Automatización y despliegue: Permite la integración con herramientas de producción a gran escala.
4. Independencia en el despliegue: Permite el despliegue de los componentes de forma independiente entre sí, garantizando así distintas configuraciones a nivel de red *core*.
5. Trazabilidad: Permite la observación de los procesos internos dentro de la red *core*.

Así, con estos criterios idóneos para la implementación de entornos educativos se tiene la siguiente tabla provista por [1]:

TABLA III. CRITERIOS DE USABILIDAD DE HERRAMIENTAS CORE PARA ENTORNOS EDUCATIVOS

Criterio	Alternativas			
	Open5GS	OAI Core	Magma	srsEPC
3GPP <i>Compliance</i>	✓	✓	X	✓
4G/5G	✓	X	X	X
Automatización y despliegue a gran escala	✓	✓	✓	X
Independencia en el despliegue	✓	X	✓	X
Trazabilidad	✓	✓	✓	✓

En donde como se muestra, Open5GS cumple con todos estos criterios de usabilidad. Por otra parte, OAI Core, se encuentra relegado por poco debido a que para el EPC depende de componentes externos como es el caso del MME de Magma, según detalla en [1]. Por esta razón, no presenta independencia a nivel de despliegue por lo que, debido a la ausencia de este componente externo a la solución el EPC, no podrá funcionar. Por su parte, Magma es una solución con un enfoque *enterprise* e innovador ideal para despliegues industriales, sin embargo, bajo un enfoque educativo que permita asociar los conceptos definidos por la 3GPP su arquitectura no termina de completar dicho criterio. Así también con los componentes de NGC o 5GC, puesto que aún se encuentran en una fase de desarrollo hasta la presente fecha por lo que el 5G no estaría habilitado. Por último, al ser una solución ligera la de srsLTE, presenta la ausencia de componentes de un 5GC. Asimismo, no presenta independencia de los componentes debido a que la solución es de tipo monolítica y, finalmente, no tiene registros que permitan la integración con herramientas de despliegue o gestión como en los otros casos.

De forma complementaria a este análisis, se presenta la tabla con los *procedures* soportados por estas herramientas provistas por [1]:

TABLA IV. COMPARATIVA DE HERRAMIENTAS DE CORE A NIVEL DE PROCEDURES

Procedure	Alternativas			
	Open5GS	OAI Core	Magma	srsEPC
Service Request	✓	✓	✓	✓
Service Release	✓	✓	✓	✓
Registration	✓	✓	✓	✓
De-Registration	✓	✓	✓	✓
EPS Bearer Establishment	✓	✓	✓	✓
EPS Bearer Release	✓	✓	✓	✓
PDU Session Establishment	✓	✓	X	X
PDU Session Release	✓	✓	X	X
Network Slice Selection	✓	✓	X	X
Handover S1	✓	✓	✓	X
Handover X2	✓	✓	✓	✓
Handover Xn	✓	✓	X	X
Handover N2	✓	✓	X	X
VoLTE	✓	X	✓	X
VoNR	✓	X	X	X
Roaming	X	X	✓	X

2.2.1.6 srsRAN

Primero, a nivel de soluciones de RAN, se tiene la propuesta de SRS la cual bajo dos proyectos define los componentes de la red de acceso para 4G como para 5G. Sin embargo, para esta última tecnología adopta un esquema que permita la implementación de O-RU en el cual se basa Open RAN. Tal como detalla [23], dentro de sus características más relevantes, cumple con los *releases* 10 hasta el 15 de la 3GPP. Por otra parte, soporta el acceso vía FDD y TDD. Asimismo, en caso de no poseer un interfaz de radio que permita probar el funcionamiento del canal inalámbrico, también permite la articulación con un simulador de canal basado en ZeroMQ. Por último, permite modulaciones de hasta 256-QAM, MIMO y soporte de movilidad (*handover*) habilitado, exclusivamente para el caso de 4G.

2.2.1.7 OAI RAN

Por su parte, OAI de Eurecom, propone un esquema bastante similar al de srsRAN en cuanto al *feature set* tanto para 4G y 5G según muestra [24]. Sin embargo, difiere en cuanto a la gestión de movilidad puesto que implementa el *handover* de forma parcial y permite el *inter-handover* entre eNodeB y gNodeB. También, es importante mencionar que en este enfoque se tiene un RAN clásica propuesta por 3GPP más no por la O-RAN Alliance.

2.2.1.8 UERANSIM

Asimismo, en [25] se detalla un simulador de RAN bastante ligero el cual, en el que toma ventaja del *functional split* propuesto para 3GPP para la división del gNodeB por medio de una unidad funcional como es el CU. En el cual, se habilitan los protocolos de NAS y NGAP, así como todos los procedimientos involucrados a nivel de registración, autenticación y establecimiento de la sesión PDU. Sin embargo, las capas MAC, RLC, PDCP no se encuentran implementadas en UERANSIM; así como, la interfaz de radio que en este caso se encuentra simulada sobre UDP.

2.2.1.9 ns-O-RAN

Finalmente, en el caso de ns-O-RAN, según se detalla en [26], es un simulador de Open RAN implementado en ns-3 el cual tiene como principal objetivo la generación de información, tales como la cantidad de usuarios conectados a la red, SNR, eficiencia espectral, *throughput*, la cual es expuesta por la interfaz E2 y que sirve como input para la integración con un near-RT RIC gracias al protocolo E2AP. Sin embargo, solo tiene como fin ese único propósito por lo que integraciones que muestren el funcionamiento a nivel de core no se encuentra disponibles; así como, la posibilidad de tener una trazabilidad definida de los paquetes que se generan dentro de la red de acceso simulada y por último la observación de *procedures* tales como la registración, *service request*, *service release* entre otros.

2.2.1.10 Análisis comparativo de las soluciones de RAN

De forma similar, al caso anterior para las herramientas de RAN, se tomará como referencia el análisis brindado a estas herramientas basado en los siguientes criterios descritos para su aplicabilidad para entornos educativos por [1]:

- 3GPP compliance: Las herramientas deben ceñirse a los lineamientos expuestos por la 3GPP a nivel de red de acceso.
- O-RAN compliance: Las herramientas deben cumplir con la arquitectura propuesta por O-RAN Alliance para lograr una arquitectura desagregada a nivel de red de acceso.
- Solución 4G/5G: Las herramientas deben tener componentes para operar ya sea para 4G como para 5G.
- Integración con equipos físicos: Las herramientas deben poseer la capacidad de articular con equipos de radio para futuras implementaciones en una interfaz de radio real.
- Trazabilidad: Permite a los estudiantes la posibilidad de observar los procesos involucrados en el funcionamiento de la red de acceso.

Bajo estos criterios, [1] provee la siguiente tabla en el que muestra si es que dichas herramientas cumplen o no con cada uno de estos:

TABLA V. CRITERIOS DE USABILIDAD DE HERRAMIENTAS RAN PARA ENTORNOS EDUCATIVOS

Criterio	Alternativas			
	UERANSIM	OAI RAN	srsRAN	ns-o-RAN
Tipo de herramienta	Simulador	Emulador	Emulador	Simulador
3GPP Compliance	✓	✓	✓	✓
O-RAN Compliance	X	X	✓	✓
Soluciones 4G/5G	X	✓	✓	X
Integración con equipos físicos	X	✓	✓	X
Trazabilidad	✓	✓	✓	X

Según la tabla anterior, se puede apreciar que la mayoría de los criterios los cumple srsRAN. Sin embargo, en el caso de OAI, se evidencia su carencia en la adopción de los parámetros definidos por la O-RAN Alliance en la estructura de sus componentes, hasta la fecha. Por su parte, UERANSIM, resulta una herramienta ligera y modular; sin embargo, al simular los componentes de capas inferiores de la red de acceso no se llega a cumplir en totalidad los lineamientos propuestos por la 3GPP, O-RAN Alliance e integración con equipos físicos. Por último, para el caso de ns-O-RAN, resulta una herramienta ideal para desarrollo de aplicaciones que permitan la integración con servicios de ML e IA en el RIC; pero que, sin embargo, para propósitos que permitan demostrar el funcionamiento de una red de acceso se queda bastante relegado con respecto a la integración de un *core*, equipos físicos o trazabilidad en cuanto a la comunicación entre los distintos componentes de la red de acceso y los UE. Adicionalmente,

se tiene la siguiente tabla provista por [1] en el que se muestran la pila de protocolos soportados por cada una de dichas herramientas que respaldan el análisis hecho anteriormente.

TABLA VI. COMPARATIVA DE HERRAMIENTAS DE RAN A NIVEL DE PROTOCOLOS

Protocolo	Alternativas			
	UERANSIM	OAI RAN	srsRAN	ns-o-RAN
RRC	✓	✓	✓	X
PDCP	X	✓	✓	X
RLC	X	✓	✓	X
MAC	X	✓	✓	X
HIGH PHY	X	✓	✓	X

2.1.2 Análisis comparativo de herramientas de orquestación

En esta sección de forma similar a la anterior, se realizará una breve descripción de las herramientas que permitan la orquestación de componentes para la generación de entornos educativos basados en los paradigmas antes mencionados, así como la capacidad de articulación que tengan junto con los componentes de *core* y RAN. A continuación, se presentarán dichos orquestadores:

2.1.2.1 OSM

Según [27] Open Source MANO, es una herramienta de orquestación y gestión de VNFs propuesto por ETSI, para el despliegue automatizado de redes móviles basadas en funciones virtuales que tiene como lineamientos el *framework* definido previamente con respecto a MANO. Asimismo, OSM toma provecho de las VNFs y NS para poder realizar el despliegue de los componentes de red a nivel de software haciendo uso de una arquitectura agnóstica al

proveedor. Por otra parte, OSM permite a los operadores de telecomunicaciones orquestar y administrar de forma eficiente los servicios y funciones de su red de forma centralizada. A continuación, la arquitectura de OSM se presenta en la siguiente imagen:

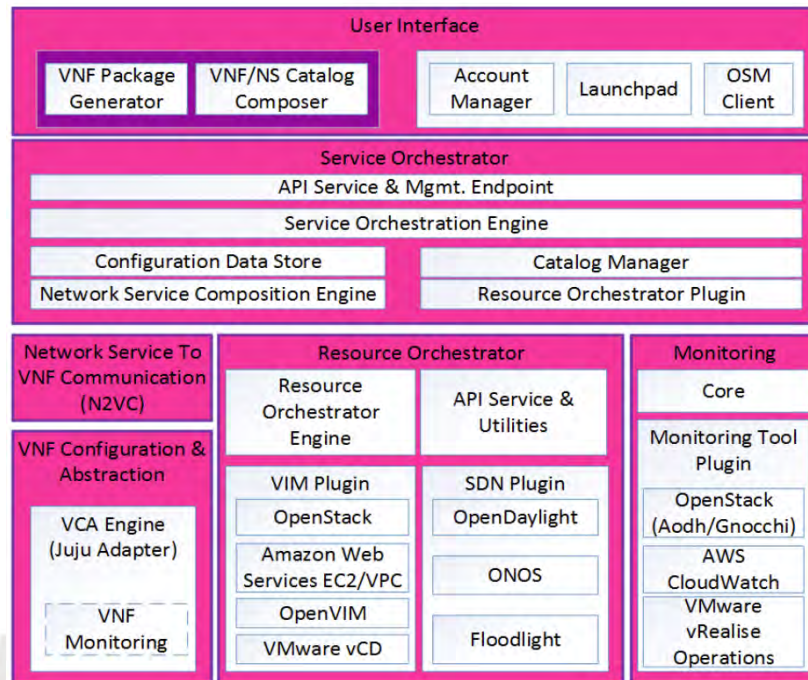


Figura 15. Arquitectura de OSM [27]

En donde como se muestra en la figura se tienen los siguientes componentes que cumplen las siguientes funciones según [27]:

- *User Interface* (UI): Es la interfaz de usuario que permite al usuario la interacción con el sistema para realizar operaciones de gestión y configuración de servicios.
- *Service Orchestrator* (SO): Es el componente principal de OSM y se encarga de la orquestación y gestión de los servicios de red. Asimismo, gestiona el ciclo de vida de las VNFs y se encarga de la coordinación de las diferentes etapas de orquestación, lo que garantiza que los servicios se desplieguen y operen de forma correcta entre sí.
- *Resource Orchestrator* (RO): Es el componente encargado del aprovisionamiento de recursos de infraestructura para la ejecución de las VNFs. Es importante tener en cuenta que esta función de aprovisionamiento se hace mediante la coordinación con el VIM

- VNF Manager (VNFM): Es responsable de la gestión de las VNF en donde se involucra aspectos como la configuración, supervisión, escalado y despliegue.
- Virtual Infrastructure Manager (VIM): Es el componente encargado de la gestión de la infraestructura virtualizada, en donde permite la asignación de recursos de infraestructura como máquinas virtuales, redes virtuales y almacenamiento.
- VNF Configuration and Abstraction (VNFC): Es un componente que permite la comunicación entre las VNF y los componentes de administración lo que permite la inserción de configuraciones, acciones y notificaciones.
- Módulo de comunicación de servicios de red a VNF (N2VC): Permite la comunicación entre el SO y la capa de configuración y abstracción (VNFC).

2.1.2.2 ONAP

Según [28], de forma similar a OSM, la fundación Linux propuso su propia solución para la orquestación de VNFs la cual permite, al igual que OSM, la automatización y orquestación de funciones de red física y virtuales llamada ONAP (Plataforma de Automatización de Red Abierta). Sin embargo, a diferencia de OSM, ONAP se basa tanto en los *frameworks* de MANO propuesto por ETSI; así como ECOMP (Enhanced Control, Orchestration, Management and Policy) propuesto por AT&T. Es por esta razón que ONAP presenta una arquitectura parcialmente distinta a OSM y un poco más compleja tal como [28] muestra a continuación:

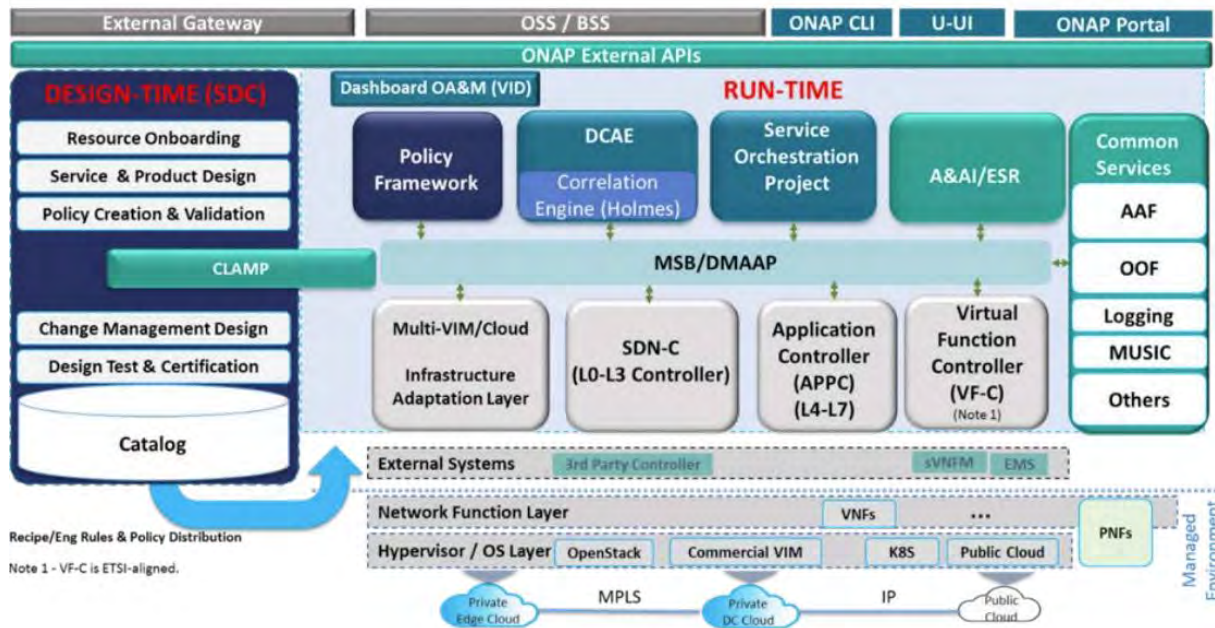


Figura 16. Arquitectura de ONAP [28]

Dentro de los componentes, según [28], más relevantes de la arquitectura de ONAP se detallarán los siguientes a continuación:

- *Service Orchestrator* (SO): Es responsable de la orquestación y automatización, modificación y eliminación de servicios de red, aplicaciones e infraestructura.
- Controladores: Estos son los controladores de servicio que gestionan aspectos tales como la configuración, la red, aplicación de políticas y el estado de las VNF.
- Analítica, Eventos y Recolección de Datos (DCAE): Recopila datos de rendimiento, uso y configuración. Asimismo, facilita en temas tales como el *troubleshooting* ya que permite tener una trazabilidad de los componentes que presentan algún tipo de falla.
- Inventario Activo y Disponible (A&AI): Proporciona información acerca del uso de recursos del sistema en función de los servicios y la relación que tienen entre sí.
- ONAP Portal: Ofrece una interfaz de usuario que permite el acceso a funciones, análisis, control y administración de las NFV, así como el estado de la red.

2.1.2.3 Kubernetes

Según [29], Kubernetes es un sistema de orquestación de contenedores de código abierto que se utiliza para automatizar la implementación, el escalado y gestión de aplicaciones en contenedores, así como de los clusters (conjunto de nodos). Su arquitectura se basa en un modelo maestro-nodo tal como se muestra en la siguiente imagen:

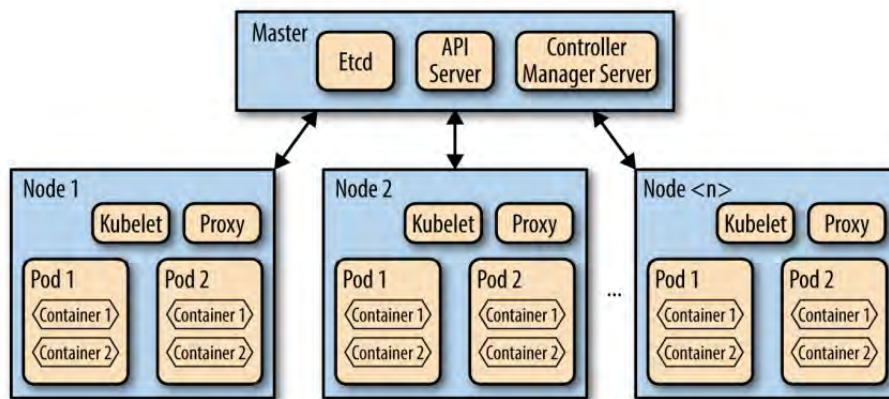


Figura 17. Estructura de Kubernetes [29]

Donde los nodos en este caso, se puede hacer referencia a máquinas virtuales o a servidores físicos. Por otra parte, los componentes de la arquitectura de Kubernetes son los siguientes:

- A nivel del *master*
 - API server: Expone una API de Kubernetes y actúa como punto de entrada para operaciones dentro de algún cluster en específico.
 - Etcd: Almacena el estado del clúster y la configuración de los componentes que lo conforman.
 - Scheduler: Se encarga de asignar las tareas de los contenedores a los nodos disponibles en el clúster.
 - *Controller Manager*: Gestiona los diferentes controladores que supervisan y mantiene el estado óptimo del clúster.
- A nivel del nodo

- Kubelet: Es el agente que se ejecuta en cada nodo y se encarga de gestionar los contenedores en el nodo, reportar su estado y definir cuotas de consumo de recursos según la capacidad del nodo
- Kube-Proxy: Es responsable de manejar las reglas de red y enrutamiento dentro del clúster a nivel de cada nodo.
- Container Runtime: Es el entorno de ejecución que permite correr aplicaciones o servicios empaquetadas por medio de contenedores (Docker).

2.1.2.4 Openstack

Según [30] definimos Openstack como controlador de recursos informáticos, de almacenamiento y de redes dentro de un espacio computacional, clúster de nodos de cómputo, todos administrados y provisionados a través de un REST API con un mecanismo de autenticación simple.

2.1.2.4.1 Servicios

Considerando [30] se procederá a describir los servicios para el despliegue minimalista de la versión victoria.

2.1.2.4.1.1 Servicio de identidad - keystone

El servicio Keystone proporciona un único punto de integración para la gestión de autenticación, autorización dentro del sistema.

Asimismo, los demás servicios de Openstack aprovechan el servicio de identidad para garantizar que los usuarios sean quienes dicen ser y descubrir dónde se encuentran otros servicios dentro de la implementación.

2.1.2.4.1.2 Servicio de imágenes - glance

El servicio de imágenes permite a los usuarios descubrir, registrar y recuperar imágenes de máquinas virtuales. Asimismo, ofrece un *endpoint* que permite consultar los metadatos de las

imágenes de máquinas virtuales y recuperar una imagen real. Puede almacenar imágenes de máquinas virtuales.

2.1.2.4.1.3 Servicio de colocación - placement

Servicio utilizado para la selección del *compute node* como host de la máquina virtual que se desea implementar.

2.1.2.4.1.4 Servicio de cómputo - nova

Servicio para alojar y administrar sistemas de computación en la nube. Este servicio hace uso de los distintos servicios ya abarcados. *Keystone* para la autenticación, *Placement* para el seguimiento y selección del inventario de recursos, *Glance* para la gestión de imágenes de disco y servidor.

2.1.2.4.1.5 Servicio de red - neutron

Neutron administra todas las facetas de red para la infraestructura de red virtual y los aspectos de la capa de acceso de la infraestructura de red física en su entorno Openstack.

Es decir, permite crear y conectar dispositivos de red, interfaces, administrados por otros servicios de Openstack a las redes. Interactúa principalmente con Openstack compute para proporcionar redes y conectividad para sus instancias.

2.1.2.5 Análisis

Tomando como referencia la siguiente tabla provista por [31] en el que se hace un análisis comparativo entre OSM y ONAP

Evaluation Criteria	OSM-4	Remarks	ONAP-B	Remarks
Resource footprint	Low	See Table III	High	Needs high spec rack-mounted servers (See Table III)
Bare metal server Installation support	!	Easier for research & development	X	Not possible to run ONAP in a single PC
Kubernetes Installation support	X	Not supported yet	!	ONAP-B can have multiple instances with different name-spaces
Performance Monitoring	!	Open for 3rd party monitoring services	!	DCAE module of ONAP is responsible for this with much richer APIs for developers of Data Analytics Applications
Multi-VIM Support	!	OpenStack,VMware, AWS	X	Only OpenStack
CLI Support	!	Powerful CLI, seems to follow OpenStack CLI paradigm	!	Not user friendly
LCM Support	X	Not available	X	Not available
Learning Curve	Easy	Very good wiki & active community support through Slack	Hard	Not well documented, pretty difficult to get clear information.
Multi-User Support	!	Can be improved	!	With different roles for Designer, tester, governor and operator
Multi-Site Support	!	Not tested	!	Not tested

Figura 18. Matriz de comparación entre ONAP y OSM [31]

En donde se aprecia que en el caso de OSM, a diferencia de ONAP, no presenta integración con Kubernetes, comparativa realizada en el *release 3* de OSM, lo que limita su capacidad de despliegue a nivel de VNF. Por otra parte, en el caso de ONAP, una desventaja reside en su alto consumo de recursos para el despliegue de redes lo que representa una limitante a nivel de infraestructura. Adicionalmente a ello, ONAP presenta un esquema de despliegue de CNF *non-standalone* esto se refiere a que se necesita de un VIM para poder desplegar un clúster, en primera instancia y, posteriormente, nodos para así poder desplegar CNF. Por otra parte, se tiene registro de soporte a nivel de integración de OSM con Kubernetes para el *release 4*, pero que sin embargo presenta los siguientes inconvenientes según detalla [31]:

- Existe la necesidad de trabajar con un VIM, por lo que el despliegue *stand-alone* no queda habilitado para esta configuración.
- No permite la selección sobre qué clúster desplegar las CNFs lo que representa en ciertos casos la creación de clústeres y el desperdicio de los recursos de infraestructura para el despliegue de CNFs

Por último, la necesidad en la que reside optar por el uso de CNF (o KNF haciendo referencia a Kubernetes *Based Network Function*) reside en el siguiente análisis a nivel de KPIs realizado por [32] que muestra los siguientes indicadores:

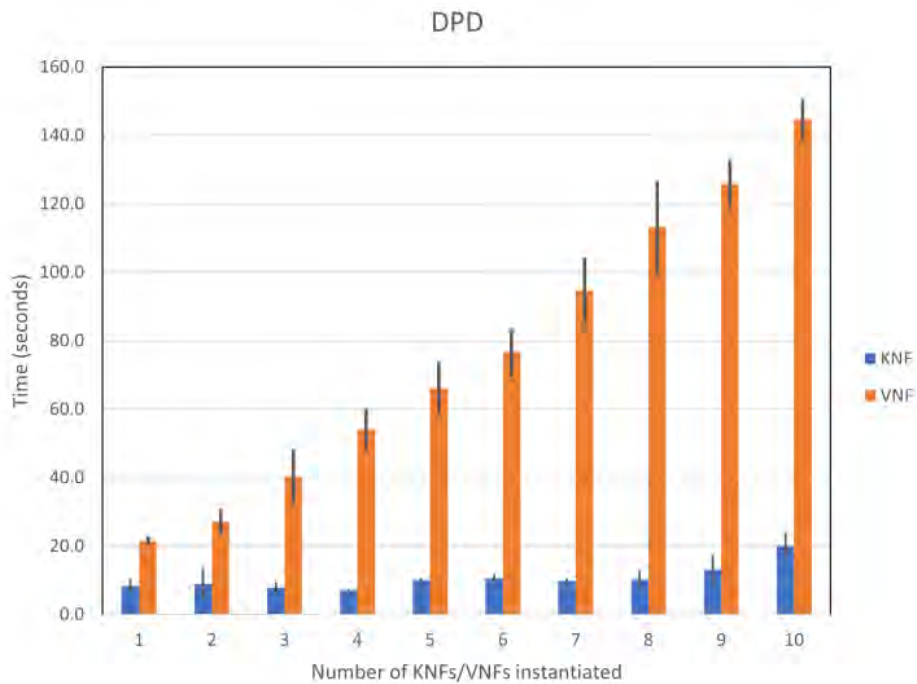


Figura 19. Comparación de los tiempos de despliegue en función de las instancias desplegadas [32]

En donde se muestra que el incremento a nivel del tiempo de despliegue de las KNF o CNF resulta mucho menor en función de la cantidad de instancias que tenga el sistema. Por otra parte, si consideramos el uso de CPU se tiene lo siguiente:

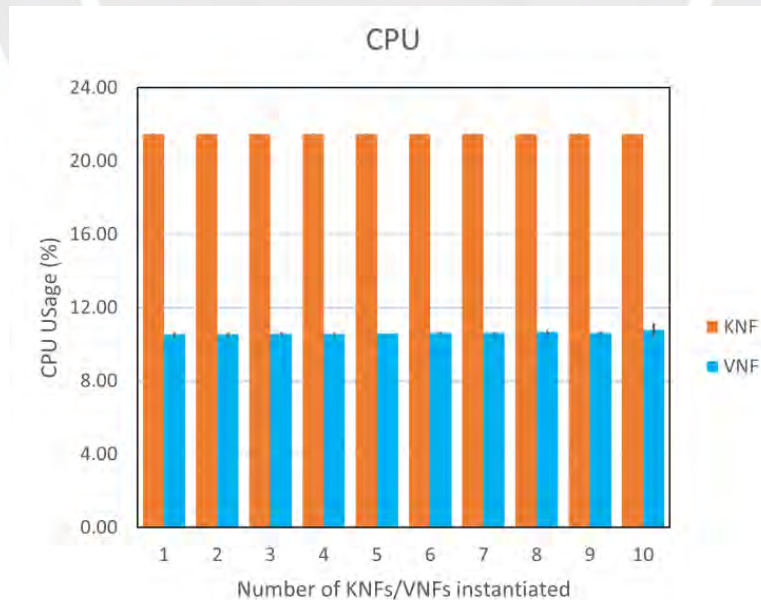


Figura 20. Comparación del consumo de CPU en función de la cantidad de instancias [32]

De forma análoga, se tiene una comparación a nivel de consumo de la RAM para ambos casos en función de las instancias

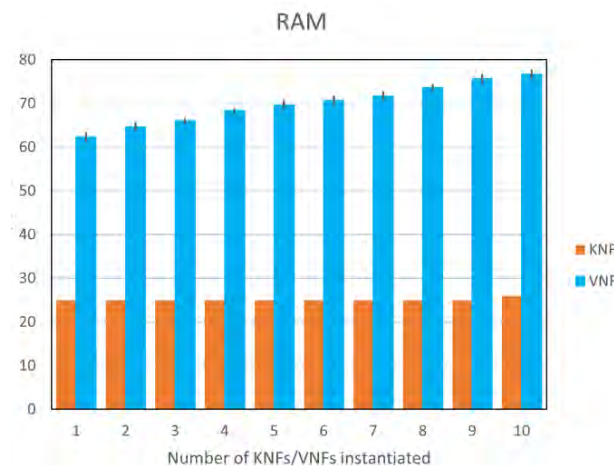


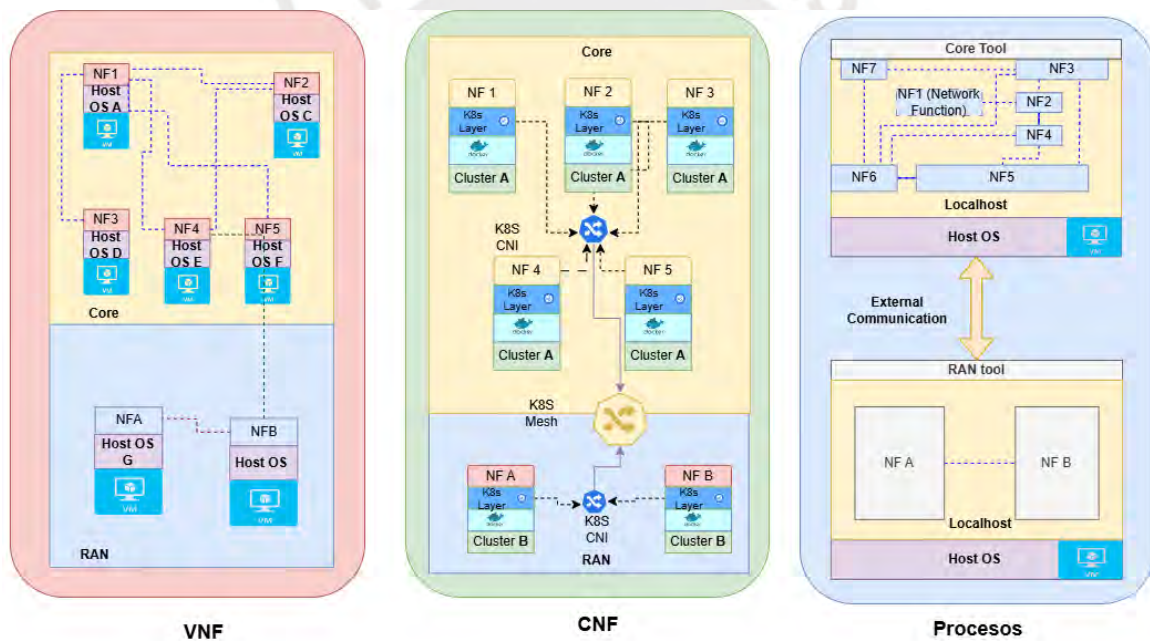
Figura 21. Comparación del consumo de RAM en función de la cantidad de instancias [32]

Por último, en [32] se analizó la cantidad de instancias que se podía tener haciendo uso de una configuración OSM+Openstack para el caso de VNF mientras que OSM+Kubernetes+*dummy* VIM (Openstack) para el caso de KNF o CNF y se tuvo que para el primer caso se tenía un máximo de instancias igual a 10 mientras que en el caso de la segunda configuración se tuvo un máximo de 116 instancias de la misma instancia que se usó para la primera configuración.

Capítulo 3. Diseño y selección de herramientas para el despliegue de entornos educativos para las tecnologías 4G y 5G

3.1 Selección de paradigma

En el análisis de la selección de paradigmas para la implementación de los entornos educativos, según lo expuesto en el capítulo anterior, existen varias configuraciones/paradigmas que permiten el despliegue de funciones de red *core* y de RAN ya sea en infraestructura virtual (VMs) o directamente en infraestructura *baremetal* (servidores multipropósitos). A continuación, se muestra la figura 22 en donde se detalla los tipos de configuraciones que pueden existir para el despliegue de las herramientas de RAN y *core*.



	VNF	CNF		Procesos
# VMs	7	Cluster A	Cluster B	2
		3	2	

Figura 22. Paradigmas de implementación

En donde cada paradigma cuenta con ventajas y limitaciones como se detallan a continuación:

- VNF: El esquema de implementación de VNFs por máquinas virtuales se mostró como una alternativa robusta en la implementación, puesto que al contar con un hipervisor, se contaba con un aislamiento necesario para garantizar la operabilidad entre los *testbeds* de cada estudiante, por otro parte, la configuración de factores de over-provisioning que maneja Openstack permite al usuario tener configuraciones de alta performance en donde se aseguren los recursos al 100 %, mientras que aplicaciones *best-effort* para escenarios en donde no se requieran recursos asegurados. Asimismo, a diferencia de la implementación por CNFs el uso de máquinas virtuales facilita la gestión de acceso remoto, dado que por ejemplo Openstack presente un pool de direcciones IPs que pueden ser expuesta de forma pública para el acceso de los estudiantes, así también se tiene Keystone (un servicio de Openstack) permite el autenticación y acceso a los recursos basados en roles lo que permite un esquema *multi-tenant*. Sin embargo, en contraposición con el caso de CNFs, este tipo de implementación demanda demasiados recursos, puesto que como se muestra en la figura 22 cada VNF cuenta con un sistema operativo aislado y con una VM exclusiva, esto obviamente limita en gran forma la escalabilidad que se pueda tener sobre un escenario en donde la capacidad en infraestructura (recurso de HW) sea limitada lo que en ciertos casos podría comprometer la experiencia del estudiante y, adicionalmente, añade un grado de complejidad mayor en el desarrollo de la experiencia debido a que el estudiante deberá ingresar tantas veces sea como cantidad de VNFs tenga disponible, lo que dificultaría la visibilidad y desarrollo de la experiencia del estudiante.
- CNF: Por su parte, en el caso de la implementación por CNF se tiene un consumo de recursos que es altamente escalable (*Scale-In/Scale-Out*) en función de los recursos que se disponga. Así también cuentan con características que permitan el distribución e integración continua (CI/CD) tal es el caso como Rancher [33], que mediante el uso de Helm charts [34], permite la integración y despliegue de pods desde archivos YAML en repositorios remotos lo que agiliza el ciclo de producción en el ciclo de vida de aplicaciones basadas en software lo que resulta ventajoso si es que se tiene una nube ya sea pública como Azure, AWS o para el caso *on-premise*. Sin embargo, para el propósito que se busca no resulta conveniente, puesto que, si bien un contenedor resulta mucho más ligero que una VM en cuanto uso de recursos de HW, el *overlay* de infraestructura (tanto Kubernetes y la capa subyacente sobre la que se encuentra soportada) tienen un impacto en cuanto al aprovechamiento de recursos disponibles que se desee lograr. Por

otra parte, desde una perspectiva educativa, el uso de contenedores no resultaría del todo conveniente, debido a que no se tiene un aislamiento de los procesos que operan dentro de los contenedores, debido a que no cuentan con un sistema operativo y/o kernel que garantice la independencia de recursos para un estudiante y para otro, dado que para este caso se tendría todos los escenarios desplegados sobre una infraestructura común (VNRT/HAST). Por último, la gestión y la accesibilidad resultan crucial para que los estudiantes puedan acceder y tener la visibilidad necesaria para poder lograr los objetivos de aprendizaje esperado por cada caso de estudio, dado este requerimiento se tuvo en cuenta que Kubernetes posee la capacidad de la gestión de acceso por medio de *Ingress Controllers* y *Load Balancers* para el acceso remoto hacia los Pods/Contenedores lo que resultaba complejo, dado que no se tenía una capa de autenticación ni autorización que permitiera acceder al *testbed* asignado a un estudiante, por lo que el acceso remoto no hubiera sido intuitivo y sencillo. Para lograr ello, se analizó gestionar el acceso por medio de un *frontend* que permitiera la autenticación y gestión mediante TTYD (*Terminal Through Your Display*) basado en *websockets* para el acceso por consola de los componentes (contenedores) que conformarán la topología por medio de una interfaz web. Por lo que esto requería de una mayor complejidad y mucho más tiempo en la cadena de producción de la solución.

- **Procesos:** En este caso, la implementación de las funciones de red por medio de procesos cuenta con las mismas ventajas que se tendría en el caso de la implementación por VNFs pero con el valor adicional de residir todas estas funciones en el mismo host en donde la comunicación se pueda hacer de manera local dentro del host y para comunicarse con otras NFs pueda hacerlo por medio de comunicación externa entre los hosts. Con esta configuración se puede aprovechar de mejor manera los recursos que por ejemplo pudieran haber sido utilizados por el kernel y el sistema operativo de cada VM en el caso de implementar los *testbeds* mediante VNFs y garantizando que se tenga un uso adecuado por parte de los entornos con un propósito educativo de alta fidelidad. Por otra parte, la implementación por procesos permita una flexibilidad considerablemente alta debido a que permite, dependiendo del caso de estudio que se tenga, implementar más o menos procesos dentro de un mismo host lo que permitiría a posteriori contar con distintas distribuciones de VMs según el caso de estudio que se desee estudiar. Sin embargo, este tipo de implementación no es escalable debido a que la flexibilidad con la que se disponen las VMs está ya preconcebida según el propósito de estudio que se

tiene, lo que no las hace viable al momento de escalar en cuanto a la carga que se pueda tener como visiblemente se puede hacer con CNFs y VNFs en menor medida.

Todos estos detalles anteriormente mencionados se pueden evidenciar en la siguiente figura 23

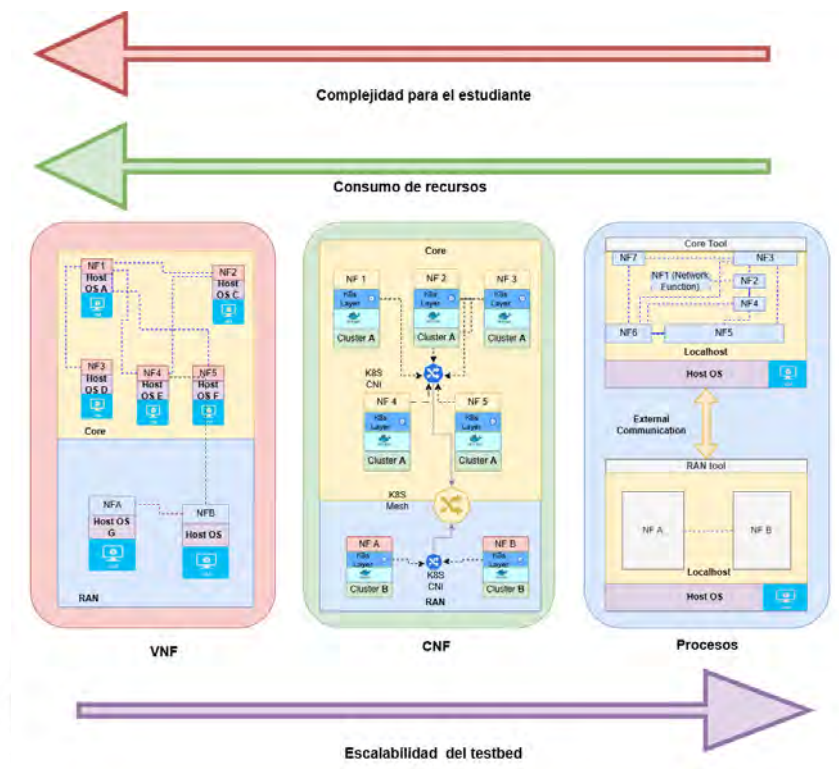


Figura 23. Comparativa de los paradigmas según enfoque educativo

Según el análisis discutido anteriormente, las alternativas de CNFs y VNFs resulta sumamente ventajosas para escenarios de producción (p.e una red móvil comercial) en donde los recursos netamente estén destinados a la carga que se tenga en función del tráfico y donde la escalabilidad juegue un rol crucial en la disposición de servicio que se deba tener para los usuarios. Sin embargo, para el caso que se desea abordar, la implementación por medio de procesos fue la elegida pues satisface el requerimiento de brindar la visibilidad necesaria al estudiante, así como también mostrarse como una alternativa ligera y flexible en cuanto a la infraestructura disponible, de forma que se pueda aprovisionar de estos *testbeds* a varios alumnos sin perder de vista la fidelidad en los resultados.

3.2 Diseño de arquitectura de la solución y herramientas de despliegue

Partiendo de lo establecido en secciones anteriores, se buscaría trabajar con servicios que aborden despliegues de NFs por lo que se tendrían dos posibles candidatos: OSM y Openstack. Por una parte, con respecto a OSM, que para el caso de estudio de la presente tesis, la curva de aprendizaje es bastante alta y adicionalmente a ello el despliegue de NFs, para el caso de estudio, no resulta idóneo puesto que se enfoca en un entorno en donde se tengan varios VIMs ideal para el despliegue en producción como el caso de un operador móvil en donde busque la virtualización de su infraestructura física por medio de la repartición de esta bajo virtualización mediante el uso de distintos VIMs ya sea en una nube privada, pública o híbrida. Por su parte, Openstack que para OSM sería un VIM, permite la creación de entornos educativos dado que permite por medio de sus APIs que permitan la adaptación de imágenes que contengan las NFs, así como la creación de topologías personalizadas de acuerdo con el requerimiento, lo que permite hacer uso de orquestadores diseñados según necesidad a diferencia de OSM. Es por estas razones y, principalmente, la alta flexibilidad que brinda Openstack con respecto a OSM que se toma la decisión de basar nuestro diseño sobre esta herramienta.

3.2.1 Gestor de Infraestructura Virtual (VIM): Openstack

El clúster de Openstack será desplegado dentro de la nube privada VNRT/HAST de la Pontificia Universidad Católica del Perú en la versión Victoria, la cual posee gran cantidad de documentación influyente para la implementación de nuestro entorno de pruebas, asimismo, es la versión en la cual se posee conocimiento tanto de uso como configuración. Asimismo, es importante realizar la observación relacionada a que el clúster se desplegará siguiendo una topología de máquinas virtuales, como ambiente de desarrollo, para que, posteriormente en un futuro, esta solución pueda ser integrada con el VNRT/HAST y que pueda ejecutarse en *baremetal* como ambiente de producción.

3.2.1.1 Servicios

Los servicios desplegados para el desarrollo de este trabajo son los siguientes:

- Keystone - *Identity service*
- Glance - *image service*
- Placement - *placement service*
- Nova - *compute service*
- Neutrón - *Networking service*
- Horizon - *Dashboard*

En este apartado, no se considera el servicio de HEAT porque al ser un servicio de orquestación de infraestructura por medio de *templates*, si bien permite la personalización de los componentes de una topología, necesita realizar el proceso de instalación de toda la paquetería cuando se realiza el despliegue de un recurso. Además, los servicios antes listados permiten realizar dicha tarea a través del uso de imágenes con todos los requisitos (librerías y paquetería) debidamente sin necesidad de añadir una nueva capa de complejidad junto con una curva de aprendizaje involucrada que pueda ser contraproducente para el desarrollo de la solución.

3.2.1.2 Entorno de prueba

Para el desarrollo de la solución y las pruebas iniciales mostradas en el capítulo 5 (que son previas a la puesta en producción en el VNRT y su uso en los laboratorios de los cursos de pregrado) se desplegó un clúster OpenStack *lightweight* sobre VMs (OpenStack *over* OpenStack) en el VNRT de la sección. La topología de este clúster se muestra en la Figura 24 y la configuración de hardware de cada elemento se muestra en la tabla VII.

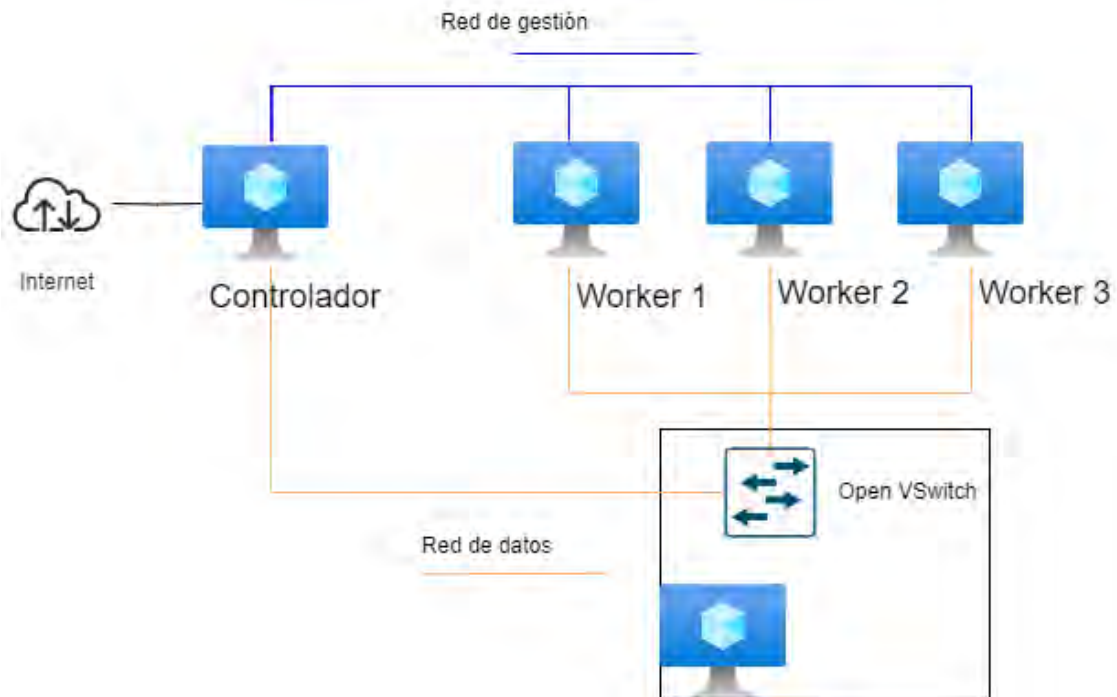


Figura 24. Topología establecida para el gestor de infraestructura

TABLA VII. REQUERIMIENTOS DE HARDWARE DEL CLUSTER DE OPENSTACK

	Controlador	Nodo de cómputo
Sistema operativo	Ubuntu 20.04.6 LTS	Ubuntu 20.04.6 LTS
CPU		
Cantidad de núcleos	4	12
Frecuencia	2 GHz	2 GHz
Tamaño	64 bits	64 bits
Memoria RAM		
Capacidad	8 GiB	16 GiB
Disco duro		
Capacidad	20 GiB	20 GiB
Volumen de almacenamiento		
Capacidad	80 GiB	100 GiB
Red		
Velocidad de transmisión	1000 Mbits	1000 Mbits

Nota: Esta tabla muestra los requerimientos sobre un esquema Openstack sobre Openstack (VNRT/HAST)

Para la gestión de red de este clúster de prueba se usaron Linux bridges en cada *worker* para la red *provider* creada como red *flat* y esta fue utilizada para obtener acceso a la red externa (internet), por otro lado, para las conexiones entre los elementos de red virtualizados se utilizan vlans para distinguir las diferentes redes creadas.

Asimismo, debido a que (para este clúster de prueba embebido en el VNRT) no se tiene la posibilidad de utilizar un pool de IPs flotantes por parte del DHCP de la red de PUCP se realizará el direccionamiento a través de NAT PAT para poder liberar puertos de los NE (*network element*) a la red externa a partir de la interfaz del controlador. Para esta característica se utilizará la herramienta de *iptables* para realizar el *forwarding* de paquetes en el enfoque establecido, NAT PAT.

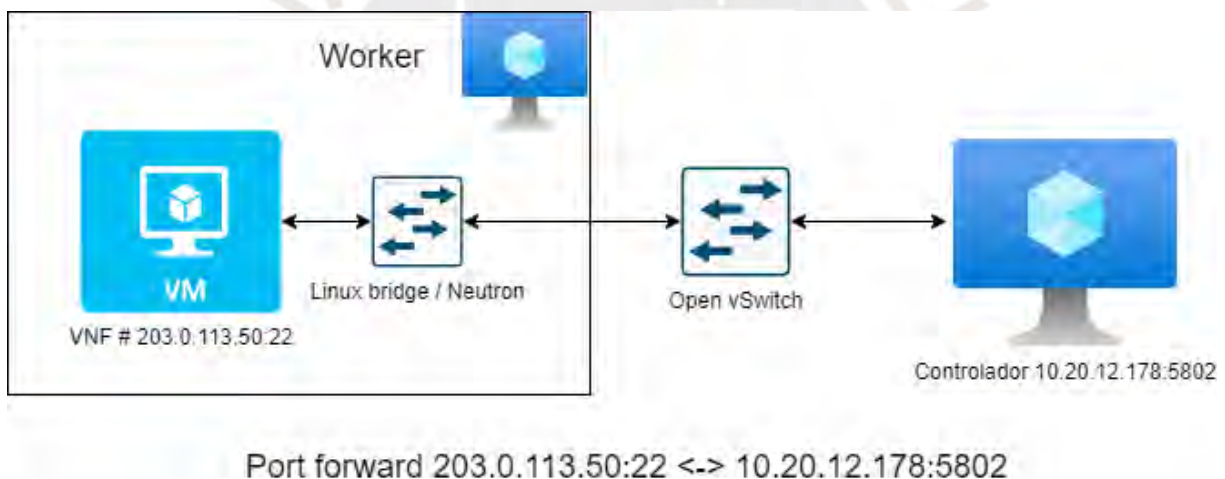


Figura 25. Esquema de *networking* relacionado a un esquema NAT PAT

En la figura anterior, por ejemplo, se observa cómo se libera el puerto 22, puerto por defecto de protocolo SSH, hacia la red externa de la universidad a través de la interfaz externa del controlador, con dirección IP 10.20.12.178, mediante el puerto 5802. Esta configuración permite liberar puertos para diferentes motivos, principalmente para las conexiones remotas.

3.2.2 Servidor Backend

Con respecto al servidor backend, este se encuentra desarrollado en Python y he hizo uso del *framework* FastAPI para la creación de los métodos API REST (GET, POST, PUT, DELETE) necesarios para la creación, listado y eliminación de *testbeds* así como también de Uvicorn para instanciar el servicio y la configuración de los puertos de escucha, así como también la dirección IP que expone el servicio para la recepción de peticiones. Asimismo, el servidor se comunica por medios de sus SDKs a los *endpoints* de cada servicio que conforma Openstack (Nova, Neutron, Glance) para la administración de la infraestructura virtual. En cuanto al proceso de la solicitud para la creación de un *testbed*, por ejemplo, se recibe como parámetros el identificador de usuario (para nuestro caso el código de estudiante) y el identificador del *testbed* que se planea desplegar, luego se valida si es que se encuentra en la base de datos de usuario (se hace uso del SDK de Keystone), haciendo uso del mismo SDK se crea un proyecto y se asigna al usuario, posteriormente, se consulta el SDK de Glance para asignar las imágenes correspondiente al escenario de estudio de las VMs precargadas en Openstack y luego se hace uso del SDK de Nova para realizar el despliegue de la VMs y el de Neutron para configurar la red de acceso y la red de tráfico (esto incluye también la asignación de IPs) del *testbed* todo ello coordinado directamente en el método de creación que expone el servidor *backend*. Para el caso de que el proceso de creación haya sido exitoso se debería tener parámetros de acceso, como la dirección de acceso y puerto por el que el usuario pueda ingresar a las máquinas virtuales. Finalmente, se consideró inicialmente que este reciba peticiones haciendo uso del protocolo HTTPs haciendo uso de un certificado TLS/SSL *self-signed*.

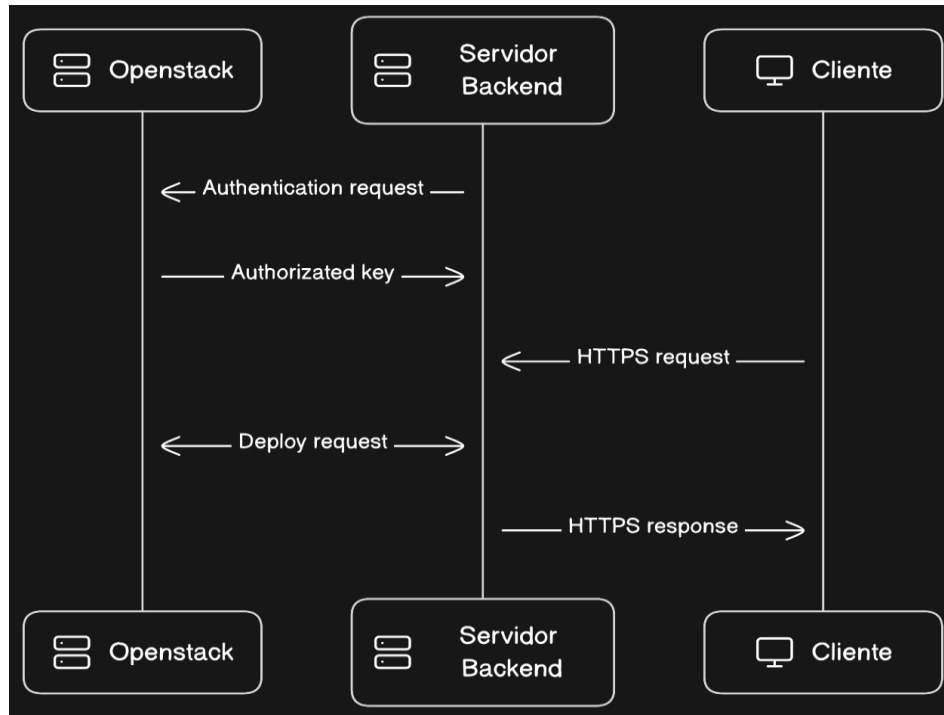


Figura 26. Flujo lógico de integración del servidor Openstack con el servidor REST.

3.3 Escenarios

En la definición de escenarios, se debe considerar el valor educativo que tiene para los estudiantes el aprender conceptos básicos de operación, casos de aplicación y nuevos paradigmas asociados a redes móviles. En esa línea, entender cómo un dispositivo móvil puede transmitir información y conectarse a una red 4G y 5G resulta crucial. Por otra parte, entender cómo múltiples dispositivos pueden transmitir datos a distintas redes resulta idóneo para entender cómo operan las redes comerciales. Por último, mostrar a los estudiantes que la introducción de nuevas tecnologías permite la adopción de arquitecturas flexibles que permitan la coexistencia de distintas redes en una sola, tal como lo hace *network slicing* o como el caso de Open RAN, que permiten una gestión inteligente de la red de acceso por medio de la interfaz E2 que comunica a xApps. A continuación, se listan los escenarios:

3.3.1 Attachment and service request

Estos procesos plantean el caso en el cual uno o más UEs, se registran ante el *core* correspondiente en donde se ven involucrados, de manera inmediata, casi todos sus componentes de la red *core*, dependiendo los siguientes sub-escenarios posibles, podemos mencionar principalmente al AMF, SMF y el UPF, que brindan la capacidad al UE de poder realizar el proceso de registraci3n ante la red y brindar una sesi3n para poder realizar conexiones ante un *Data network*. An3logamente, para el caso de 4G se ven involucrados diferentes componentes como el MME relacionado al registro y manejo de sesiones para la interconexi3n del UE con un PDN en particular, as3 como el SGW y HSS. Por ello, podemos definir que estos procesos son el primer acercamiento al funcionamiento de una red m3vil. Basado en esta idea, a continuaci3n, los sub-escenarios de inter3s para esta tesis.

3.3.1.1 Attachment and service request a nivel de core y a nivel de red de acceso

Como lo mencionamos anteriormente, siendo el proceso principal para las redes m3viles se plantea el caso, tanto para 4G y 5G, donde un UE realiza el proceso de registro y requerimiento de servicio con la finalidad de observar la pila de protocolos y procesos relacionados a estos procesos tanto a nivel de RAN como a nivel de *core*. Con lo que la finalidad de este proceso es afianzar los conocimientos b3sicos adquiridos durante el proceso de desarrollo profesional,

3.3.1.2 Multiples APNs - Corporative DN

Para este caso, considerando el funcionamiento base visto en el escenario anterior se plantea lo siguiente: Una empresa la cual brinda a sus colaboradores dispositivos m3viles como herramientas de trabajo desea asegurar que los dispositivos no solamente tengan acceso a la red de datos habitual, sino, adem3s, a una red privada propia de ellos. Para esto, cada UE deber3 conectarse a m3s de un APN, asimismo, se realizar3 un despliegue mayor de UEs para simular el estado de una compa1a. Esto con la finalidad de estudiar el comportamiento de la red *core*

y observar el comportamiento secuencial de los procedimientos relacionados al escenario y validar el comportamiento propuesto.

3.3.2 *Network slicing*

En las redes 5G de la actualidad, se suelen tener distintos modos de operación, de los cuales, los más conocidos por la ITU-T son eMBB, mMTC y URLLC que pueden coexistir dentro de una misma red móvil física gracias a la división de ésta en subredes lógicas que permitan a la red operar en estos modos gracias al *network slicing*. Por ello, la 3GPP planteó el uso del S-NSSAI dentro de los procedimientos que conforman el *core* 5G, cuya finalidad es brindar y seleccionar diferentes recursos de red a los dispositivos conectados mediante la segmentación de red. Para estudiar dicha funcionalidad, se plantea el escenario en el cual un UE se conecte a dos diferentes APNs, simulando dos servicios que requieran diferentes capacidades de conectividad y como consecuencia dos *network slices* respectivamente.

3.3.3 Interfaz E2 implementación de RIC

La adopción de Open RAN permite optimizar el funcionamiento de la red de acceso (RAN) gracias al uso de algoritmos especialmente diseñados o de herramientas de inteligencia artificial, eligiendo, por ejemplo, los mejores pesos y retardos a usar para el beamforming hacia un usuario móvil. Esta oportunidad, como expresa la *O-RAN alliance*, se da gracias al enfoque de las arquitecturas desagregadas a nivel de RAN, que, a partir de aplicaciones externas, conocidas como xApps, permite tanto realizar el monitoreo como la toma de decisiones a partir de la información recopilada mediante la exposición de la interfaz.

Para ello, el proyecto srsRAN implementó la interfaz E2 dentro del gNB de forma que habilite la capacidad de obtener información de los componentes de la RAN, tanto CU como DU, los cuales a posteriori son utilizados por las aplicaciones xApps por medio de un RIC, como se

mencionó anteriormente. Es por ello por lo que la finalidad de este escenario tiene como objetivo el estudio de la información brindada por la interfaz E2 del gNB en una RAN desagregada y cómo este se integra a un RIC y brinda un control y monitoreo de la RAN a nivel de software.

3.4 Herramientas de simulación y emulación

3.4.1 Red Core

Considerando el análisis revisado en tablas V y VI, se seleccionó para los entornos educativos el *core* 4G y 5G de Open5GS, que si bien está fuertemente enfocado a entornos 5G también cumple con los procedimientos técnicos establecidos por la 3GPP para 4G. Cabe destacar que la configuración de este emulador posee la característica de permitir la desagregación del plano de control y el plano de usuario. Dicha configuración será visualizada en los diseños de los entornos educativos por proceso.

3.4.2 Red Acceso

Para el siguiente trabajo se seleccionó, dentro de las herramientas establecidas en el capítulo 2, los siguientes simuladores y emuladores.

En primer lugar, EURANSIM debido a la capacidad de despliegue masivo que permitirá a futuro realizar pruebas de concurrencia y capacidad de los elementos que conforman la red *core*. Sin embargo, este simulador solamente tiene la capacidad de simular los procesos básicos de la red 5G

En segundo lugar, srsRAN, en sus dos presentaciones como emulador de red 4G y 5G, seleccionado debido a que permite la emulación tanto de la capa física, transmisión electromagnética, como de los procesos dentro de la red móvil. Asimismo, permite la integración con equipos de radio para realizar pruebas bajo un entorno más real en un futuro.

3.4.2 Resumen de las herramientas a utilizar

Finalmente, la siguiente tabla mostrará las herramientas seleccionadas para cada escenario.

TABLA VIII. DISTRIBUCIÓN DE HERRAMIENTAS POR ESCENARIO

CORE	Open5GS-EPC	Open5GS-5GC		Open5GS-NS-5GC	Open5GS-Monolithic
RAN	srsRAN-LTE	srsRAN-NR	UERANSIM	UERANSIM-NS	srsRAN-E2
UE					
	Attachment and service request 4G	Attachment and service request 5G	Múltiples APNs, Corporative DN	Network slicing	Interfaz E2 implementación de RIC

3.5 Arquitectura de los entornos educativos

A continuación, se mostrarán las arquitecturas relacionadas a los escenarios que, anteriormente, fueron nombrados y explicados, asimismo, se identificarán las herramientas que se usarán para cada uno de ellos.

3.5.1 Arquitectura para pruebas de *Attachment and service request*

En primer lugar, en consideración del escenario descrito en el inciso 3.3.1 estableceremos una topología de interconexión lineal, ya que el esquema de interconexiones se basa en una comunicación *p2p*. Posteriormente, dependiendo la generación en la cual estemos trabajando se implementarán los siguientes escenarios.

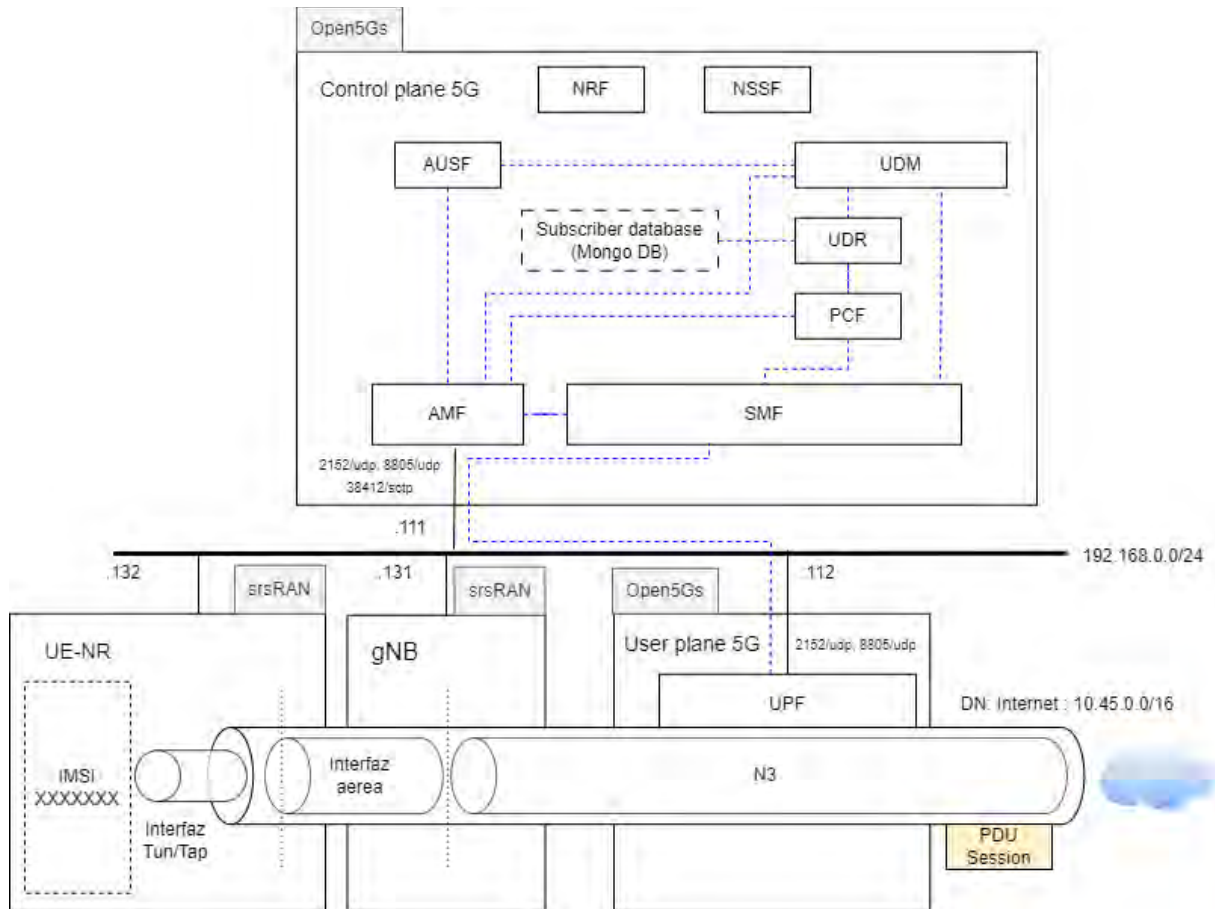


Figura 27. Arquitectura del escenario propuesto 5G

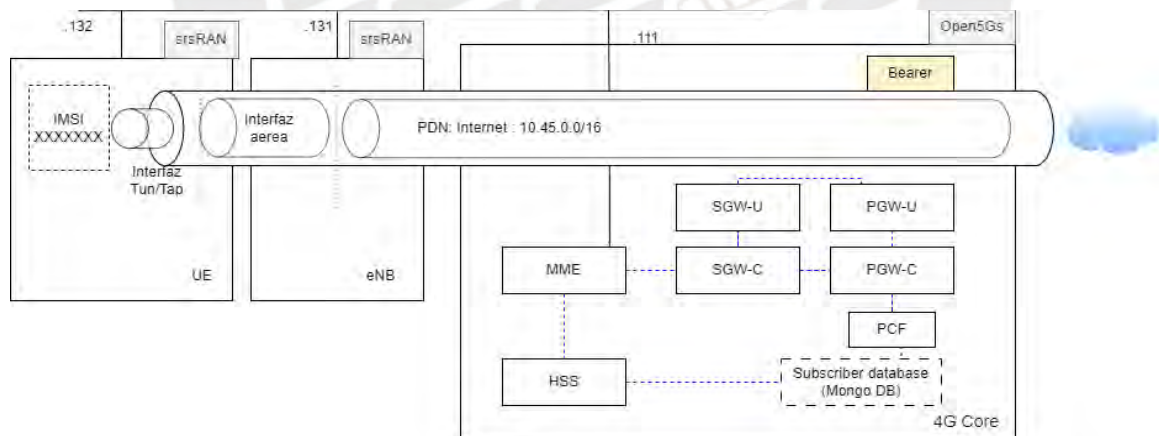


Figura 28. Arquitectura del escenario propuesto 4G

En estos escenarios, verificaremos los procesos mencionados a través de la inspección de paquetes *pcap* en los diferentes elementos que conforman estos escenarios. Cabe destacar que la emulación de la interfaz aérea se realiza a través del protocolo HTTPS no podemos, a simple vista, obtener el *payload* necesario para ver el comportamiento de estos dispositivos. Para ello,

se utilizarán los archivos obtenidos mediante la propia herramienta srsRAN, en el mismo formato *pcap*, para la identificación de los paquetes relacionados desde la conexión entre UE y RAN. Por otro lado, haciendo referencia al siguiente escenario, a continuación, se mostrará la arquitectura propuesta.

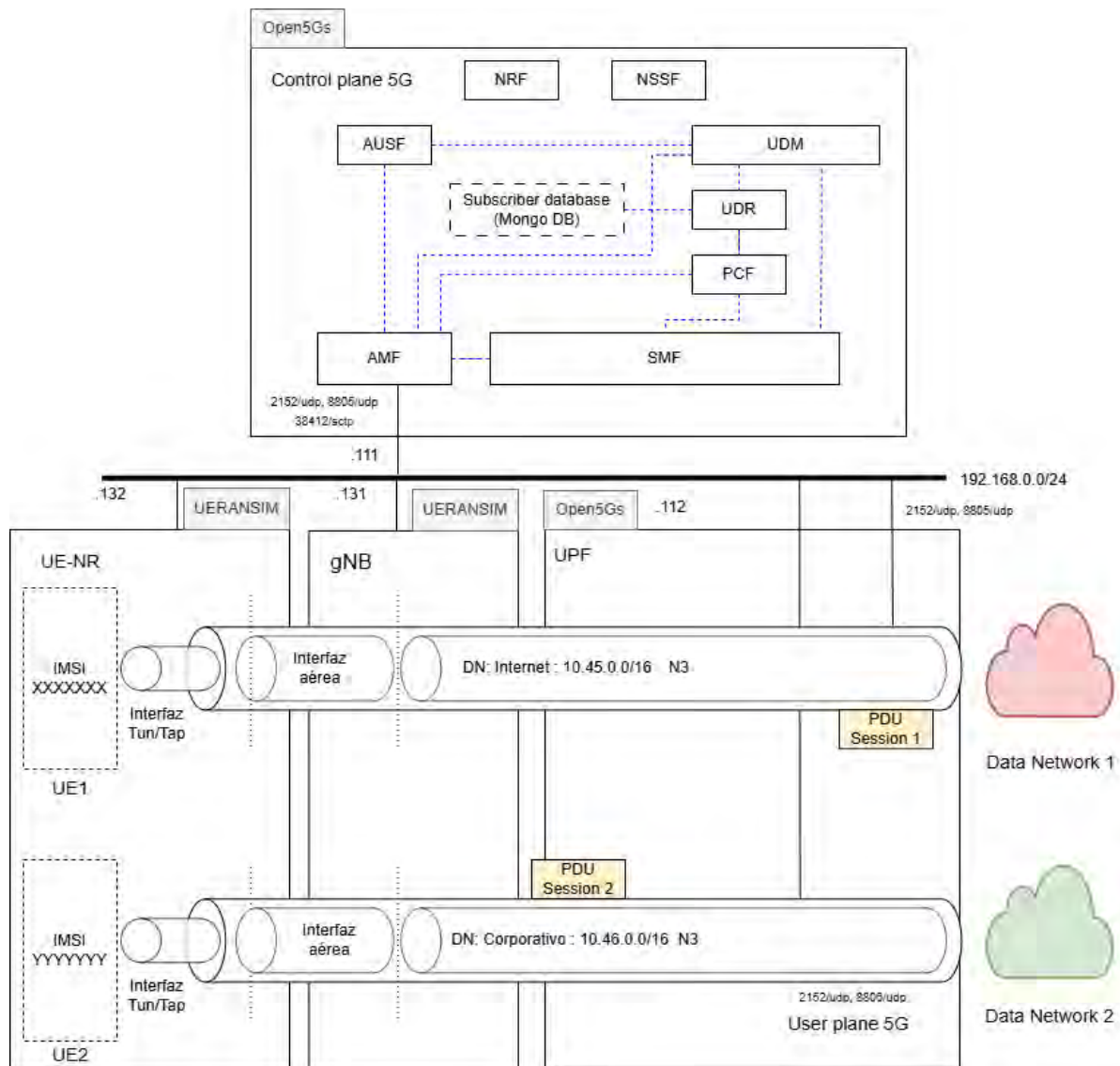


Figura 29. Arquitectura del escenario DN corporativo

Como se hace referencia en el inciso anterior se busca evaluar el caso de uso de una red móvil corporativa en los cuales los dispositivos móviles, UEs, estarán conectados a múltiples APNs.

3.5.2 Arquitectura para pruebas de *Network Slicing*

En segundo lugar, verificaremos el proceso de segmentación de red, en este escenario se observará la simulación de un acondicionamiento de la red de acceso mediante las interfaces implementadas por el simulador, UERANSIM. A continuación, se observará la figura que describe la arquitectura propuesta para el análisis de este escenario.

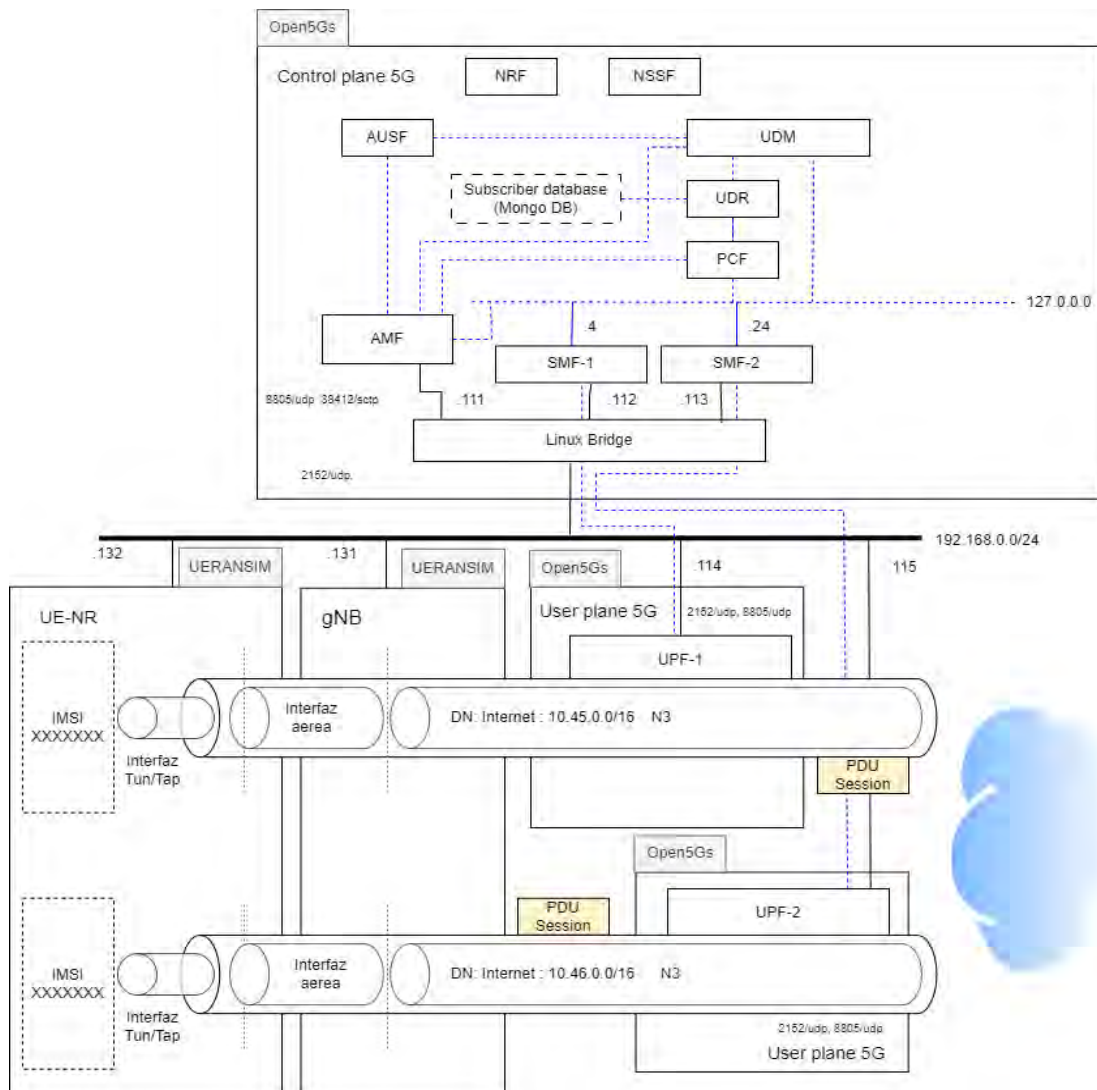


Figura 30. Arquitectura del escenario *network slicing* [37]

En la figura anterior, se aprecia una topología con dos *network slices*, se consideró este número justamente para poder ilustrar en una topología básica cómo se realiza la asignación de *slices* en una misma red. Para este propósito, se tuvo que considerar 2 pares de SMF-UPF correspondientes a cada *network slice* asociado. Para poder soportar los SMF en una misma

VM de forma concurrente, se implementó una imagen de *core* 5G Open5GS que incluye un Linux bridge al que se conectan los SMFs de cada *slice*. Cada SMF tiene una interfaz con su propia dirección IP para la comunicación por la interfaz N4. Cada UPF se desplegó en una VM distinta con una interfaz conectada a la red de comunicación entre VMs (192.168.0.0/24) de forma que permita comunicarse con su SMF. Así, los UEs podrán conectarse a la red con según el S-NSSAI configurado en cada uno de estos según el *slice* soportado/configurado.

3.5.3 Arquitectura para el estudio de la *Interfaz E2* y el *RAN Intelligent Controller (RIC)*

Finalmente, la arquitectura usada para el estudio de OpenRAN es presentada en la Figura 32. Se utiliza, como en los casos anteriores, una topología de red lineal a la que se agrega una VM que implementa el *Near Real Time RAN Intelligent Controller* (NearRT-RIC) de OpenRAN. La VM que implementa el gNodeB (gNB) de srsRAN ha sido compilada con el *flag* “E2-interface-support” a fin de implementar la interfaz E2 que conecta el DU/CU con el NearRT-RIC. Para reducir la carga de procesamiento, y dado que las experiencias a desarrollar en este escenario no requieren la visualización de los paquetes en las interfaces en las funciones del *core* sino que se centran en observar el funcionamiento de la RAN, la VM que implementa el *core* 5G será desplegada con la imagen *Open5GS Monolithic* resultando en un *lightweight core*.

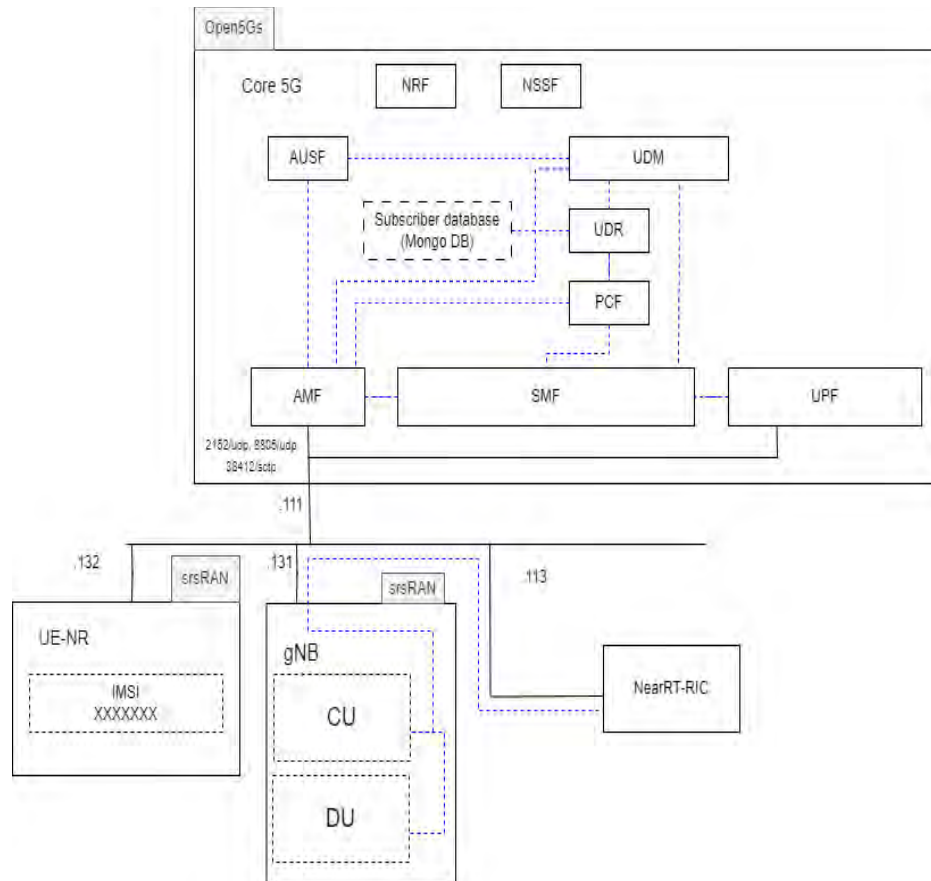


Figura 31. Arquitectura del escenario 5G con NearRT-RIC

3.6 Limitaciones de diseño

En el diseño expuesto anteriormente, es importante mencionar que existen las siguientes limitaciones:

- Ausencia de equipos de radio: Según [22] y [23] srsRAN, tanto para 4G como 5G cuentan con soporte para equipos físicos llamados USRP (Universal Software Radio Peripheral) que permiten incorporar la interfaz de radio a srsRAN. Al no contar con equipos de radio físicos no se pueden probar aplicaciones como VoLTE para 4G o VoNR para 5G ni su integración con un IMS.
- No contar con dispositivos con soporte PTP en la red móvil ya sea *switches* o *routers* con soporte para este protocolo o en su defecto sincronizados por GPS resulta en afectación en los servicios (voz y datos) reflejados en degradación debido al aumento

de la latencia, problemas en la movilidad (*handoffs*), reducción en la eficiencia de enrutamiento, reducción de ancho banda (*throughput*) debido a la ausencia en la sincronización entre los equipos.

- Ausencia de tarjetas SIM para desarrollo: Soluciones como las que plantea Sysmocom en [38] que apertura escenarios para pruebas de *SIM cards* personalizables para dispositivos móviles del mercado o dispositivos IoT no se encuentran disponibles actualmente, ello limita en parte poder tener una pequeña red móvil con interfaz aérea e integración con un IMS, así como aplicaciones de una red móvil para IoT.
- Ausencia de instrumentos aisladores: en caso se tuvieran las tres limitantes anteriormente mencionadas cubiertas, aún existiría esta cuarta limitación. Y es que en el Perú no existen bandas de frecuencia asignadas a la educación y/o investigación. Las bandas disponibles están ya licenciadas a operadores móviles inscritos en el Plan Nacional de Atribución de Frecuencias (PNAF), lo que implica que si se quiere trabajar con dispositivos móviles comunes se deben usar herramientas que permitan aislar el entorno de pruebas físico (p.ej., jaulas de Faraday) para que sus transmisiones no se propaguen en el medio compartido y regulado. Sin estas herramientas, sólo se podrán realizar pruebas como la siguiente en bandas no licenciadas como las de 2.4 o 5 GHz (donde no se puede evitar interferencias que distorsionen los resultados):



Figura 32. Implementación de escenario con interfaz aérea usando USRPs [22]

Capítulo 4. Análisis e Implementación de entornos educativos 4G y 5G basados en herramientas Open Source

4.1 Análisis de la solución

En cuanto a las consideraciones de diseño que se tuvo en cuenta para la generación de entornos educativos se presenta a continuación el siguiente análisis:

4.1.1 Aspecto educativo

La correcta elaboración de escenarios de acuerdo con casos de uso de sumo interés para redes 4G y 5G tiene como consecuencia los siguientes aportes:

- Suplir la necesidad de los estudiantes de poder realizar pruebas experimentales y análisis de los procesos.
- Suplir la carencia de infraestructura móvil con enfoque educativo.
- Aportar en la generación de profesionales mucho más competentes y crear líneas de investigación en redes móviles.
- Complementar de forma vivencial los conceptos teóricos vistos en los cursos de redes móviles.

Por otra parte, es importante mencionar que, para este inciso, en particular, se mostrará pruebas y validaciones en el siguiente capítulo que verificar lo mencionado.

4.1.2 Aspecto económico

Hoy en día, existen soluciones COTS o *ready-to-use* que permiten el despliegue y análisis de redes 4G y 5G, por ejemplo, el caso de Amarisoft, el cual según los registros obtenidos en [39], [40] y [41] presentan de forma general la siguiente arquitectura:



Figura 33. Arquitectura de AMARISOFT callbox mini



Figura 34. Arquitectura de AMARISOFT callbox Classic



Figura 35. Arquitectura de AMARISOFT UE Simbox

Asimismo, dichas soluciones, según consultas a los proveedores, suelen tener los siguientes precios:

TABLA IX. LISTADO DE PRECIOS DE LAS SOLUCIONES DE AMARISOFT

Componente	Precio (USD \$)
Amarisoft callbox Mini	24 999,00
Amarisoft callbox Classic	32 999,00
Amarisoft UE simbox serie E	38 999,00

Nota: Los precios son referenciales al año 2023

Sin embargo, para este aspecto, es importante saber que según las fuentes [39]-[41] plantean solo escenarios generales de una sola red para varios usuarios. Mientras que el enfoque que se busca en el presente trabajo es brindar diversos escenarios para cada estudiante, los cuales sean detallados a mayor profundidad en secciones anteriores, basados en el uso de máquinas virtuales, por lo que para ello sea idóneo definir un *baseline* correspondiente a la cantidad de recursos computacionales que pueda ser definido para un *testbed* por estudiante de forma que basado en esa cantidad de recursos se puedan idear distintos escenarios para su uso y que, al finalizar la experiencia académica, estos recursos puedan ser liberados de forma que se tenga un uso flexible sobre los recursos que se disponen. Es así que a continuación se elaborará la siguiente tabla en donde se tenga listados los recursos computacionales

TABLA X. RECURSOS NECESARIOS PARA LA GENERACIÓN DE TESTBED POR ALUMNO

Recursos	Cantidad
CPU	16 Cores
Memoria RAM	16 GB
Disco	80 GB

Con esto en cuenta, se realizó la estimación de un costo aproximado considerando un *testbed* basado en 4 máquinas virtuales en donde se tenga sus recursos compartidos equitativamente para distintos proveedores de nube pública y estos son los siguientes resultados:

TABLA XI. ESTIMACIÓN DE PRECIOS SEGÚN PROVEEDORES DE NUBE PÚBLICA

Proveedor de Nube	Tiempo de uso	Precio por Estudiante (USD \$)	Cantidad de Estudiantes	Precio Total (USD \$)
Google	1 semana y media	160	24	3 840
Azure	1 semana y media	200	24	4 800

AWS	1 semana y media	183	24	4 392
Oracle	1 semana y media	159	24	3 816

Nota: Precios referenciales al año 2023

Con esta primera aproximación, se tiene en cuenta el valor de la solución, según distintos proveedores de nube, basado en la infraestructura virtual necesaria para la implementación de los *testbeds*. Sin embargo, es importante tener en cuenta que hasta la fecha no se tiene alguna herramienta existente que permita la generación de *testbeds* de acuerdo con escenarios específicos por lo que el valor económico quedaría supeditado a poder suplir las necesidades en la adquisición de conocimientos para los estudiantes dentro de la facultad con respecto a redes móviles.

4.1.3 Aspecto ambiental

El gran aporte de usar una infraestructura virtual en la generación de *testbeds* permite el *refarming* de recursos computacionales permitiendo así que, por medio de un conjunto de servidores, ya sea de una nube pública o privada, cuando un *testbed* ya no se encuentre en uso simplemente se elimine y así libere los recursos computacionales con el fin de generar otros escenarios de acuerdo con otros temas de estudio. Por otra parte, la virtualización permite que no se dependa más de infraestructura física puesto que la capacidad de poder abstraer las funcionalidades de distintos sistemas operativos sobre un mismo *host* físico habilita la consolidación de los recursos computacionales de ese *host* y junto con ello la reducción del consumo de energía al no depender de la adquisición de más infraestructura.

4.1.3 Aspecto legal

Se considera que esta novedosa propuesta para la generación de *testbeds* debe acogerse a la licencia GNU-GPL, de modo que todos los componentes involucrados puedan derivarse

libremente y permanecer abiertos para fomentar la innovación y el desarrollo en beneficio de la sociedad.

4.2 Implementación de los escenarios diseñados.

Con lo expresado anteriormente, al trabajar bajo un paradigma de VNFs se implementó y configuró un entorno de Openstack, en el que se especifican sus características, tipología y configuraciones de red en el capítulo 3, donde esta plataforma será la base para la implementación de los entornos educativos. A continuación, se establecerán las diferentes características que poseen los entornos educativos, como las características computacionales y configuraciones de redes.

4.2.1 *flavors*

Dentro de las herramientas establecidas anteriormente, se cumplen diferentes funciones como la emulación de un dispositivo concreto, como por ejemplo para el caso de srsRAN se tiene una emulación en la interfaz aérea, MAC y capas superiores. Por otro lado, se tiene herramientas que realizan una simulación ligera de procesos a nivel de capas inferiores, pero que mantienen consistencia en las superiores, como es el caso de UERANSIM. Asimismo, se tiene el caso de Open5GS que trabaja con varios componentes de *core* de forma simultánea. También, se tiene el caso de FlexRIC que permite correr xApps así como agentes que permiten recolectar información de la RAN. Es por todo esto que se debe estimar las características computacionales requeridas para cada elemento que constituyen los escenarios, por lo que para lograr dicho objetivo se tendrá un dimensionamiento de acuerdo con los escenarios, ya que cada uno de estos demanda características computacionales propias de acuerdo con la aplicación de estudio.

TABLA XII. REQUERIMIENTOS DE HARDWARE POR NETWORK ELEMENT SEGÚN CASO

Caso de estudio	Componente	Cantidad de núcleos	Cantidad de RAM (GB)	Cantidad de disco (GB)
5G NR	srsRAN-UE	4	4	10
	srsRAN-NR	4	4	10
	Open5GS-5GC	4	4	10
	Open5GS-UPF	2	2	10
4G LTE	srsRAN-UE	4	4	10
	srsRAN-LTE	4	4	10
	Open5GS-EPC	4	4	10
DN corporativo	UERANSIM-UE	2	2	10
	UERANSIM-GNB	2	2	10
	Open5GS-5GC	4	4	10
	Open5GS-UPF-Corp	4	4	10
nearRT-RIC	srsRAN-UE	4	4	10
	srsRAN-E2	4	4	10
	FlexRIC	2	2	10
	Open5GS-Monolithic	4	4	10
Network Slicing	UERANSIM-NS-UE	2	2	10
	UERANSIM-NS-GNB	2	2	10
	Open5GS-NS-5GC	4	4	10
	Open5GS-NS-UPF1	2	2	10
	Open5GS-NS-UPF2	2	2	10

4.2.2 Imágenes

Con respecto a la creación de imagen se realizó, de forma iterativa, la creación y configuración de cada imagen que se necesitará para realizar los entornos preestablecidos. Esto se realizó con la finalidad de que cada VNF utilice una imagen con la herramienta instalada y configurada en relación con el proceso que se busca trabajar.

La secuencia de creación de las imágenes fue la siguiente. En primer lugar, se utilizó una imagen de Ubuntu Server 20.04.6 LTS, versión estable y compatible con las herramientas a utilizar. Posteriormente, se realizó la actualización e instalación de repositorios y paquetes requeridos para la compilación y ejecución de las herramientas. En tercer lugar, se procedió a descargar el código fuente de cada herramienta para posteriormente compilarlo y configurarlo. En último lugar, mediante Openstack, se realizó el *snapshot* correspondiente a la máquina virtual configurada.

Finalmente, mediante la herramienta de terminal para acceder a las imágenes disponibles dentro un clúster de Openstack, se obtiene las siguientes imágenes:

```
ubuntu@controller:~$ openstack image list
```

ID	Name	Status
28448826-e060-4fcd-9266-0e4cff73aa50	FlexRIC	active
55d270d8-ff97-4bb3-a51d-4111d61e797f	Open5GS-5GC	active
8a9f3227-f8ae-415b-9ee5-4892dc5000ac	Open5GS-EPC	active
e7ab7173-bfb6-4e9b-ad60-537465472305	Open5GS-Monolithic	active
289b7939-de2b-472d-9a03-367dc6d68add	Open5GS-NS-5GC	active
83df1b27-6eaa-4380-bea4-0f5edfdb7a06	Open5GS-NS-UPF1	active
3f406713-f6c8-4e3d-bcae-db0617d20544	Open5GS-NS-UPF2	active
22ca5e0a-1d6a-4512-a900-32b47e921a94	Open5GS-UPF	active
826143dc-9197-4cf5-b533-6f66e5979a46	Open5GS-UPF-Corp	active
f1b095cc-4281-418a-b30d-924a21b32b4b	SRSRAN-E2	active
d7e4a200-b380-47e5-81fd-0eb036374743	SRSRAN-LTE	active
12bff54b-36b4-418c-80fa-05689b7f72a9	SRSRAN-NR	active
498eb6a2-8221-4c68-8ea9-aa6bcac29716	UERANSIM-GNB	active
07fdc884-af6f-487a-a344-ffd613a0d2e7	UERANSIM-NS-GNB	active
01d2d7fb-87cd-43a5-acc2-793431c8727d	UERANSIM-NS-UE	active
7e3364a6-bada-497c-8731-b3998df435e2	UERANSIM-UE	active
b6b93c6e-ecc9-4f67-9ead-b1efb80e728d	ubuntu_base	active

Figura 36. Lista de imágenes configuradas en el cluster de Openstack

En la figura anterior se muestran las diferentes imágenes configuradas que permitirán la generación de entornos educativos mediante su integración en una misma topología de acuerdo con los casos antes mencionados en el capítulo anterior.

4.2.3 Redes

La gestión de redes que se posee dentro del entorno de trabajo, Openstack, hace referencia a redes diferenciadas mediante VLANs. Esto nos permite que dentro de una infraestructura compartida se posea diferentes redes, las cuales sean excluyentes entre sí para mantener la integridad de los sistemas desplegados.

Asimismo, la gestión de interfaces de red, tanto en cantidad como en direccionamiento IP, es realizada por el orquestador minimalista. Este es un componente de software que implementa flujos de trabajo predefinidos. Su finalidad es desplegar los escenarios establecidos, en respuesta a una solicitud HTTPS enviada al servidor REST.

4.2.4 Topologías

A continuación, se mostrará la elaboración de las topologías en base a los componentes descritos anteriormente y a los casos de uso definidos en un principio. Para esto, se realizará una solicitud HTTPS en donde se tenga el siguiente contenido:

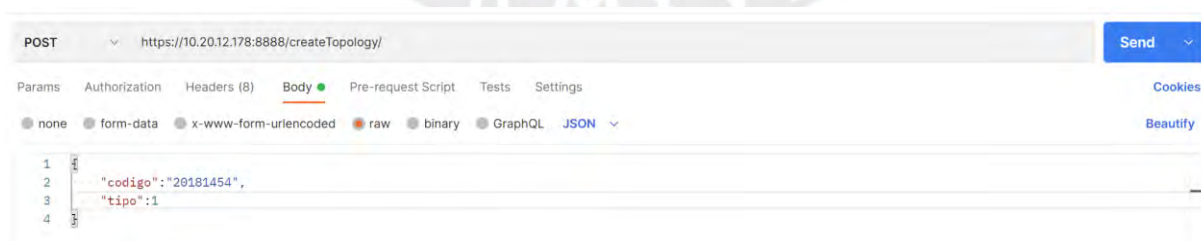


Figura 37. Parámetros de envío al servicio REST

Donde se tiene como parámetros de entrada el código de estudiante y el tipo de topología que se está solicitando y, por consiguiente, se tenga la siguiente respuesta:



Figura 38. Respuesta del servicio REST

Aquí, se tiene como respuesta la dirección IP de acceso al cual el usuario deberá conectarse de forma externa. Asimismo, tiene la interfaz de red con la que hablará entre los distintos componentes y por otra parte, la red de *management* que permite el acceso y la salida de internet de la máquina virtual. Adicionalmente, todo lo descrito anteriormente, se puede visualizar en la interfaz gráfica de Openstack en donde se pueda apreciar los recursos creados, en base a las imágenes y los *flavors* seleccionados, así como también las redes que se han creado para dicha topología a través de la siguiente imagen:

Instancias

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
Open5GS-UPF-20181454	open5gs_upf	49d83d3b-ba94-407e-b5f2-f6b3481b6bc9 192.168.0.112 provider 203.0.113.209	ubuntu	-	Activo	nova	Ninguno	Corriendo	10 minutos	Crear instantánea
Open5GS-5GC-20181454	open5gs_5gc	49d83d3b-ba94-407e-b5f2-f6b3481b6bc9 192.168.0.111 provider 203.0.113.177	ubuntu	-	Activo	nova	Ninguno	Corriendo	12 minutos	Crear instantánea
UERANSIM-GNB-20181454	ueransim	provider 203.0.113.191 49d83d3b-ba94-407e-b5f2-f6b3481b6bc9 192.168.0.131	2/2/10	-	Activo	nova	Ninguno	Corriendo	12 minutos	Crear instantánea
UERANSIM-UE-20181454	ueransim	49d83d3b-ba94-407e-b5f2-f6b3481b6bc9 192.168.0.132 provider 203.0.113.196	2/2/10	-	Activo	nova	Ninguno	Corriendo	13 minutos	Crear instantánea

Figura 39. Listado de instancias creadas

Por otra parte, las redes que se tienen para la gestión de esta topología son las siguientes:

Redes

Nombre = Filtrar [+ Crear red](#) [Borrar redes](#)

Mostrando 2 elementos

<input type="checkbox"/>	Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
<input type="checkbox"/>	49d83d3b-ba94-407e-b5f2-f6b3481b6bc9	e45858f4-173e-45f6-a694-46b22ab9d8db 192.168.0.0/24	no	no	Activo	ARRIBA	nova	Editar red
<input type="checkbox"/>	provider	providerS 203.0.113.0/24	Sí	Sí	Activo	ARRIBA	nova	Editar red

Figura 40. Listado de redes creadas

Y finalmente, la topología creada en el visor de Openstack:

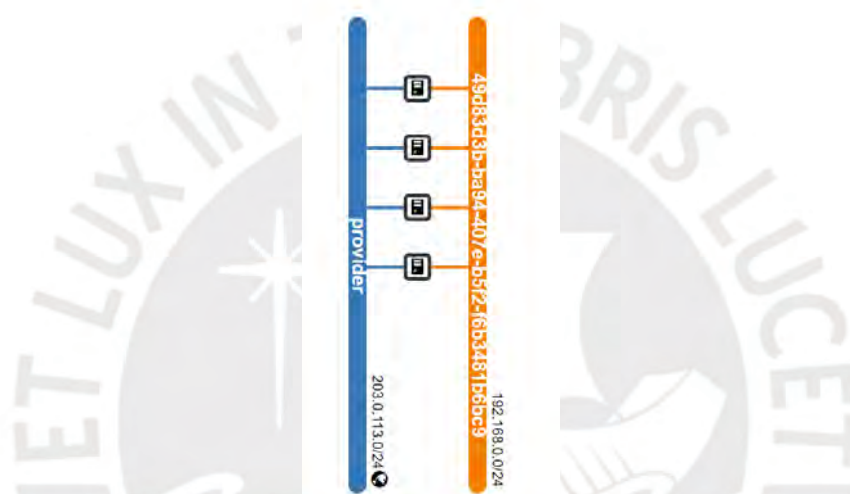


Figura 41. Topología tipo 1

Cabe comentar que el mapeo entre los tipos de topología y los escenarios antes mencionados se pueden apreciar en la siguiente tabla:

TABLA XIII. LISTADO DE ESCENARIOS EDUCATIVOS DE INTERÉS

Tipo	Escenario
1	DN corporativo
2	LTE PDN Connectivity Request & Attachment
3	RIC
4	NR PDU Session Establishment & Attachment
5	Network Slicing

4.2.5 Acceso

Por último, para el acceso a los componentes de la topología antes mencionados se realizará por medio de SSH, para lograr esto, en el momento de la creación de dicha topología, a cada componente se le realizó una asignación de puerto para su acceso por medio de iptables en el controlador, tal como se había mencionado en el capítulo anterior y como se refleja en la siguiente figura:

```
DNAT      tcp -- anywhere          anywhere          tcp dpt:5802 to:203.0.113.196:22
DNAT      tcp -- anywhere          anywhere          tcp dpt:5803 to:203.0.113.191:22
DNAT      tcp -- anywhere          anywhere          tcp dpt:5804 to:203.0.113.177:22
DNAT      tcp -- anywhere          anywhere          tcp dpt:5805 to:203.0.113.209:22
```

Figura 42. Asignación de puertos para el acceso remoto

```
PS C:\Users\Agust> ssh ubuntu@10.20.12.178 -p 5804
ubuntu@10.20.12.178's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-147-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 4.0

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

113 updates can be applied immediately.
63 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Thu Nov  2 21:08:08 2023
ubuntu@56c:~$ |
```

Figura 43. Acceso remoto

Capítulo 5. Análisis de resultados a nivel de implementación de los entornos 4G y 5G

En este capítulo se expondrán los resultados obtenidos de acuerdo con los escenarios propuestos en las secciones anteriores y en donde se realizará un análisis y una comparación con los estándares que se tienen en la industria que demuestren la viabilidad de la solución.

5.1 *Attachment y PDN Connectivity Request - 4G*

Es importante tener en cuenta que en este escenario se tenga las siguientes consideraciones al momento de visualizar los resultados, por lo que para este fin se tiene la siguiente tabla con todos los componentes involucrados en el escenario y la función que cumple:

TABLA XIV. LISTADO DE INTERFACES PARA EL ESCENARIO DE 4G

VM	Componentes de Core 4G	Interfaces	Función
EPC	MME	127.0.0.2:5868	Diameter
		192.168.0.111:3641 2	S1AP
		127.0.0.2:2123	GTPC
	PCRF	127.0.0.9:5868	Diameter
	SGWC	127.0.0.3:2123	GTPC
		127.0.0.3:8805	PFCP
	SGWU	192.168.0.111:2152	GTPU
		127.0.0.6:8805	PFCP
	PGW-C	127.0.0.4:7777	SBI
		127.0.0.4:8805	PFCP
		127.0.0.4:2123	GTPC
		127.0.0.4:2152	GTPU
	PGW-U	127.0.0.7:8805	PFCP
		127.0.0.7:2152	GTPU
HSS	127.0.0.8:5868	Diameter	
VM	Componentes de RAN	Interfaces	Función

eNB	eNB	192.168.0.121	TX/RX
UE	UE	192.168.0.122	TX/RX

Asimismo, es de interés determinar el *callflow* a identificar, a partir del [42] y [43], para verificar la característica especificada anteriormente del escenario en cuestión. A continuación, se puede visualizar en la siguiente figura el *callflow* relacionado al *attachment* y *PDN Connectivity Request* para redes 4G.

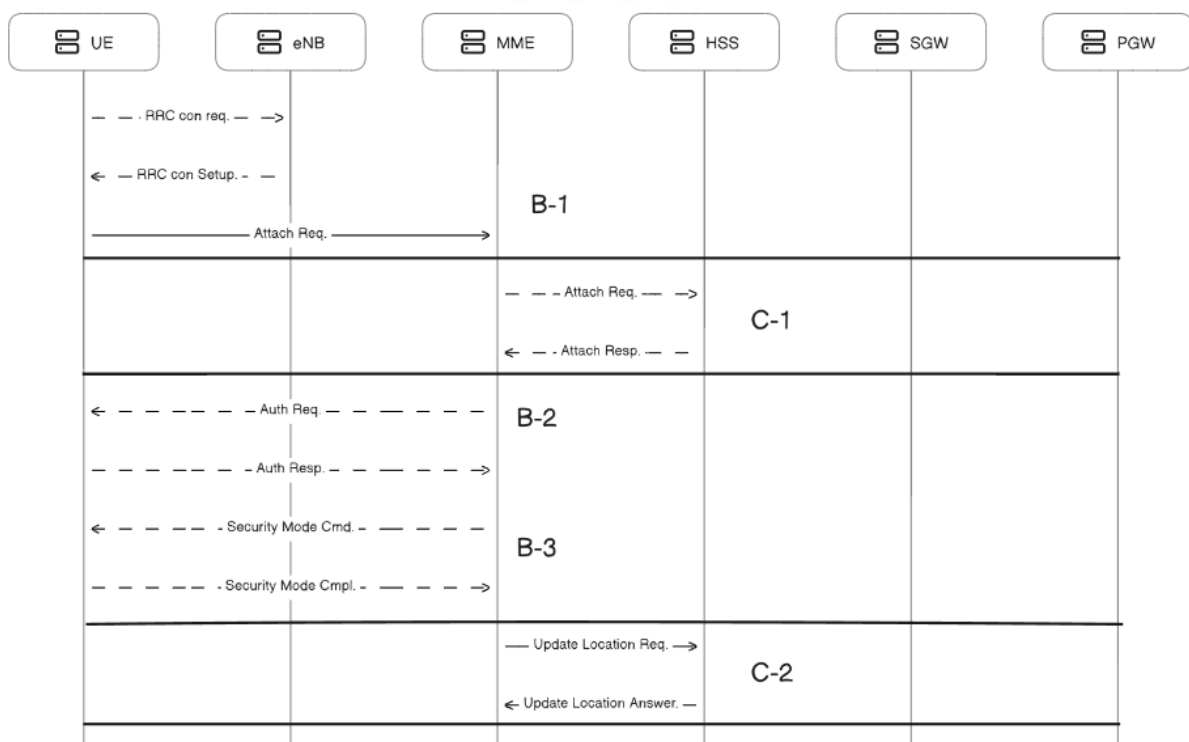


Figura 44. Callflow de attachment y PDN connectivity request para 4G parte 1

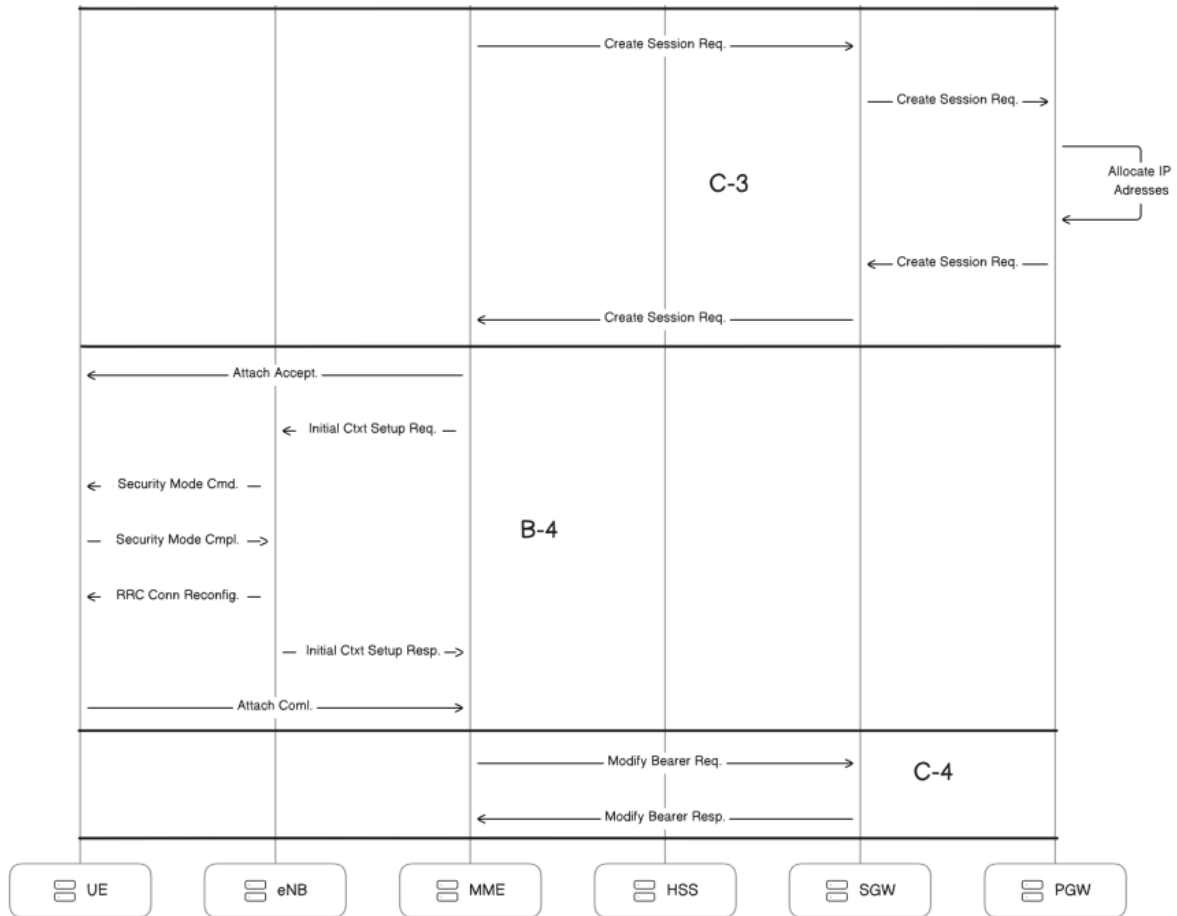


Figura 45. Callflow de attachment y PDN connectivity request para 4G parte 2

5.1.1 Análisis y comparación de paquetes

En la siguiente sección, se analizarán los paquetes obtenidos con el *sniffer* TCPDUMP de las diferentes interfaces (internas y externas) mostradas en la tabla XIV para poder analizar estos paquetes para verificar la secuencialidad de los subprocessos relacionados. Inicialmente, Como se evidencia en la figura 43, se observa el subprocesso de *Attachment request*, B-1, el cual es ejecutado por el UE que es reenviado por el eNB.

Intervalo	eNB	MME/SGW-U	Comentario
46.361404	56710	36412	S1AP/NAS-EPS: InitialUEMessage, Attach request, PDN connecti...

Figura 46. Callflow en el core 4G subsección B-1

Posteriormente, en la subsección C-1, se observó el proceso de obtención de datos relacionados con la autenticación del UE, en el cual se involucra el MME y HSS.

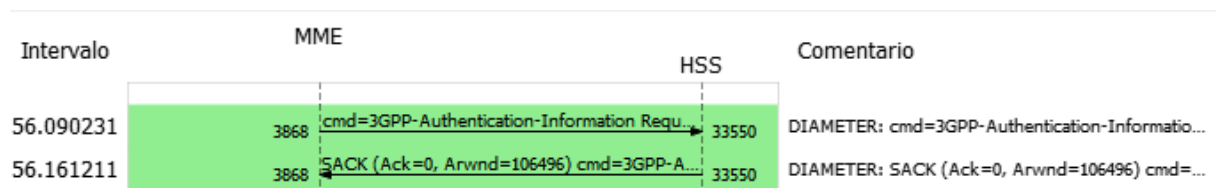


Figura 47. Callflow en el core 4G subsección C-1

Cabe destacar que el proceso C-1 genera los procesos establecidos como B-2 y B-3 relacionados a la autenticación del UE con el *core*.

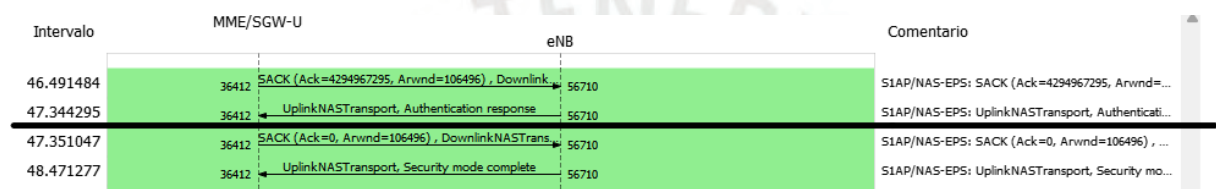


Figura 48. Callflow entre eNB y el core 4G subsecciones B-2 y B-3

A continuación, se ejecuta el proceso de location update entre el MME con el HSS identificado como C-2

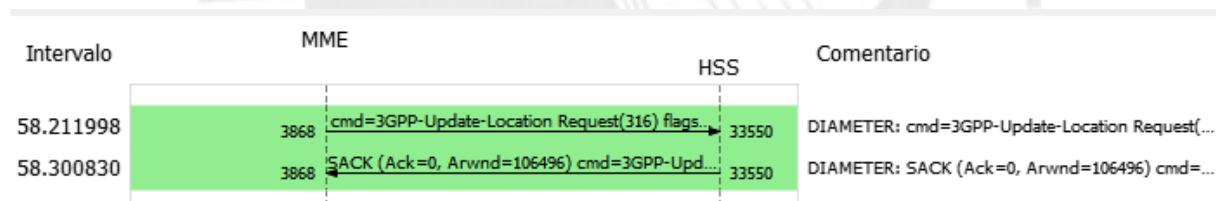


Figura 49. Callflow en el core 4G subsección C-2

Por otro lado, Posteriormente al proceso C-2 se realiza el proceso de creación de sesión identificado como C-3

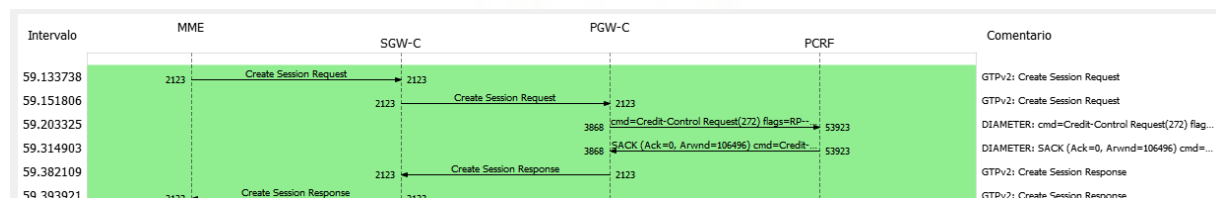


Figura 50. Callflow en el core 4G subsección C-3

A partir de todos los procesos anteriores se ejecutará un conjunto de subprocesos para completar el proceso de *attachment* identificado como B-4.

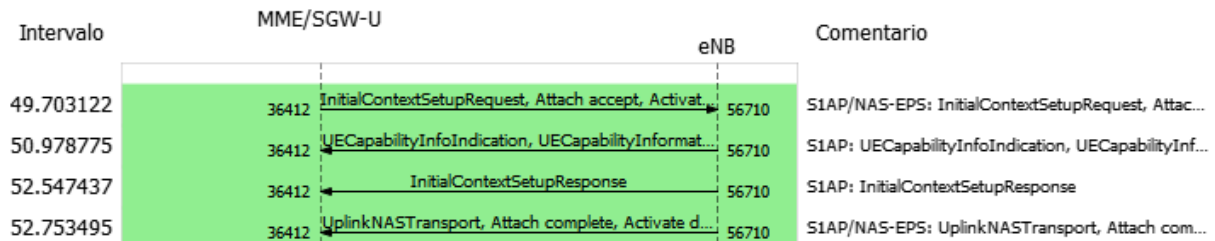


Figura 51. Callflow entre eNB y el core 4G subsecciones B-4

Finalmente, el proceso C-4 concluye el proceso de *attachment* del UE y con el establecimiento del bearer para la transmisión de datos.

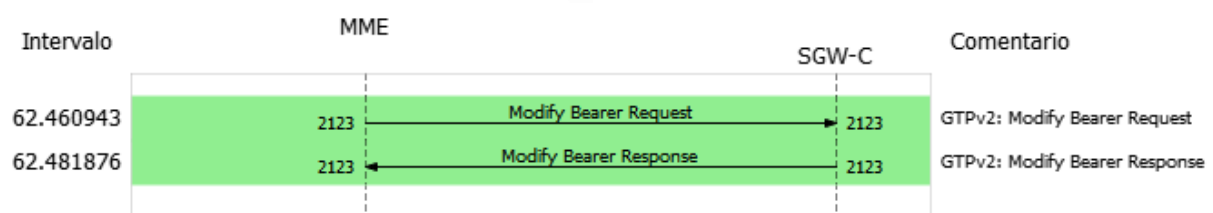


Figura 52. Callflow en el core 4G subsección C-4

5.1.2 eNodeB - UE

En esta sección se mostrará los paquetes obtenidos a nivel MAC y física.

1 0.000000	MAC-LTE	42 RAR (RA-RNTI=2, SFN=34, SF=7) (RAPID=17: TA=0, UL-Grant=52236, Temp C-RNTI=70)
2 0.367310	LTE RRC UL	41 RRCConnectionRequest
3 0.581142	LTE RRC DL	66 RRCConnectionSetup
4 1.686339	LTE RRC UL	192 RRCConnectionSetupComplete, Attach request, PDN connectivity request
5 1.731177	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=1
6 1.890446	LTE RRC DL	52 DLInformationTransfer, Identity request
7 2.786250	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=1 , ULInformationTransfer, Identity response
8 2.786359	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=2
9 3.174072	LTE RRC DL	91 DLInformationTransfer, Authentication request
10 3.832503	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=2 , ULInformationTransfer, Authentication response
11 3.873696	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=3
12 3.914443	LTE RRC DL	75 DLInformationTransfer, Security mode command
13 4.791650	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=3 , ULInformationTransfer, Security mode complete
14 4.850788	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=4
15 5.149570	LTE RRC DL	60 DLInformationTransfer, ESM information request
16 6.222723	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=4 , ULInformationTransfer, ESM information response
17 6.245713	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=5
18 6.798848	LTE RRC DL	52 SecurityModeCommand
19 7.428681	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=5 , SecurityModeComplete
20 7.481036	LTE RRC DL	56 [DL] [AM] SRB:1 [CONTROL] ACK_SN=6 , UECapabilityEnquiry
21 8.053698	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=6 , UECapabilityInformation
22 8.103851	RLC-LTE	41 [DL] [AM] SRB:1 [CONTROL] ACK_SN=7
23 8.240808	LTE RRC DL	147 RRCConnectionReconfiguration, Attach accept, Activate default EPS bearer context request
24 9.493354	LTE RRC UL	551 [UL] [AM] SRB:1 [CONTROL] ACK_SN=7 , ULInformationTransfer, Attach complete, Activate default EPS bearer context accept
25 9.503023	RLC-LTE	52 [DL] [AM] SRB:1 [CONTROL] ACK_SN=8 [DL] [AM] SRB:2 [CONTROL] ACK_SN=1
26 9.809439	LTE RRC DL	91 DLInformationTransfer, EMM information
27 10.651704	RLC-LTE	551 [UL] [AM] SRB:2 [CONTROL] ACK_SN=1

Figura 53. Capturas obtenidas a nivel mac entre el UE y el eNB

En la figura anterior se evidencia todos los paquetes que se tiene en la figura 43 y 44, donde los primeros se establece la conexión RRC entre el UE y eNB. Posteriormente, se realizan los procesos de *attachment* y *PDN connectivity request*.

5.1.3 Prueba de funcionamiento

En la siguiente sección se validará secuencialmente el correcto funcionamiento del escenario.

5.1.3.1 Open5GS

Dentro de la instancia con *hostname* Open5GSCore se ejecuta el siguiente comando mostrado en la figura, cabe destacar que se requerirá que el usuario que ejecute dicho comando tenga permisos de super usuario.

```
ubuntu@Open5GSCore:~$ ./runEPC.sh
```

Figura 54. Comando de inicialización de EPC

Para validar el funcionamiento se debe verificar que los siguientes procesos están ejecutándose correctamente.

```
ubuntu@Open5GSCore:~$ ps -aux | grep open5gs
open5gs 2519 1.3 0.3 2385564 14260 ? Ssl 06:37 0:04 /usr/bin/open5gs-mmed -c /etc/open5gs/mme.yaml
open5gs 2527 0.4 0.5 263152 19864 ? Ssl 06:37 0:01 /usr/bin/open5gs-sgwc -c /etc/open5gs/sgwc.yaml
open5gs 2539 1.9 1.1 2662376 43608 ? Ssl 06:37 0:06 /usr/bin/open5gs-smfd -c /etc/open5gs/smf.yaml
open5gs 2572 0.3 0.4 251860 18644 ? Ssl 06:37 0:01 /usr/bin/open5gs-sgwud -c /etc/open5gs/sgwu.yaml
open5gs 2583 0.3 0.6 276284 24948 ? Ssl 06:37 0:01 /usr/bin/open5gs-upfd -c /etc/open5gs/upf.yaml
open5gs 2592 1.2 0.4 2408500 16932 ? Ssl 06:37 0:04 /usr/bin/open5gs-hssd -c /etc/open5gs/hss.yaml
open5gs 2603 1.2 0.4 2407012 16532 ? Ssl 06:37 0:04 /usr/bin/open5gs-pcrfd -c /etc/open5gs/pcrf.yaml
open5gs 2646 0.6 0.4 131968 15948 ? Ssl 06:37 0:02 /usr/bin/open5gs-nrfd -c /etc/open5gs/nrf.yaml
open5gs 2658 1.1 0.4 144732 17468 ? Ssl 06:37 0:03 /usr/bin/open5gs-scpd -c /etc/open5gs/scp.yaml
open5gs 2697 0.2 0.3 133672 15584 ? Ssl 06:37 0:00 /usr/bin/open5gs-ausfd -c /etc/open5gs/ausf.yaml
open5gs 2730 0.2 0.3 140208 14908 ? Ssl 06:37 0:00 /usr/bin/open5gs-udmd -c /etc/open5gs/udm.yaml
open5gs 2744 0.3 0.5 183320 19976 ? Ssl 06:37 0:01 /usr/bin/open5gs-pcfd -c /etc/open5gs/pcf.yaml
open5gs 2753 0.2 0.3 131948 14652 ? Ssl 06:37 0:00 /usr/bin/open5gs-nssf -c /etc/open5gs/nssf.yaml
open5gs 2758 0.2 0.3 139428 14200 ? Ssl 06:37 0:00 /usr/bin/open5gs-bsfd -c /etc/open5gs/bsf.yaml
open5gs 2763 0.3 0.4 167308 18520 ? Ssl 06:37 0:01 /usr/bin/open5gs-udr -c /etc/open5gs/udr.yaml
```

Figura 55. Comando de verificación de funcionamiento

5.1.3.2 srsRAN eNB

Dentro de la instancia con *hostname* srsRAN se ejecuta el siguiente comando mostrado en la figura, es importante mencionar que dicha la ejecución de este comando se hace mediante el modo superusuario.

```
ubuntu@SRSRAN:~/srsRAN_4G/build/srsenb$ sudo ./src/srsenb enb.conf
```

Figura 56. Comando de inicialización de eNB

Se valida el funcionamiento esperando el siguiente output.

```

ubuntu@SRSRAN:~/srsRAN_4G/build/srsenb$ sudo ./src/srsenb enb.conf
[sudo] password for ubuntu:
--- Software Radio Systems LTE eNodeB ---

Reading configuration file enb.conf...

Built in Release mode using commit fa56836b1 on branch master.

Opening 1 channels in RF device=zmq with args=fail_on_disconnect=true,tx_port=tcp://192.168.0.121:2000,rx_port=tcp://192.168.0.122:2001,id=enb,base_srate=23.04e6
Supported RF device list: zmq file
CHx base_srate=23.04e6
CHx id=enb
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://192.168.0.122:2001
CH0 tx_port=tcp://192.168.0.121:2000
CH0 fail_on_disconnect=true

==== eNodeB started ====
Type <t> to view trace
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Setting frequency: DL=2680.0 Mhz, UL=2560.0 MHz for cc_idx=0 nof_prb=50

```

Figura 57. Comando de verificación de funcionamiento

5.1.3.3 srsRAN UE

Para el caso de inicializar la emulación del UE tendremos que, dentro de la máquina virtual correspondiente, ejecutar el siguiente comando, en la figura se observa el *path* en el cual debe ser ejecutado.

```

ubuntu@SRSRAN:~/srsRAN_4G/build/srsue$ sudo ./src/srsue ue.conf

```

Figura 58. Comando de inicialización de UE

A continuación, para determinar el correcto funcionamiento se hará una serie de pasos. En primer lugar, se debe comparar el output obtenido con el siguiente.

```

ubuntu@SRSRAN:~/srsRAN_4G/build/srsue$ sudo ./src/srsue ue.conf
Reading configuration file ue.conf...

Built in Release mode using commit fa56836b1 on branch master.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://192.168.0.122:2001,rx_port=tcp://192.168.0.121:2000,id=ue,base_srate=23.04e6
Supported RF device list: zmq file
CHx base_srate=23.04e6
CHx id=ue
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
CH0 rx_port=tcp://192.168.0.121:2000
CH0 tx_port=tcp://192.168.0.122:2001
Waiting PHY to initialize ... done!
Attaching UE...
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimation)

Found Cell: Mode=FDD, PCI=1, PRB=50, Ports=1, CP=Normal, CFO=-0.2 KHz
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Current sample rate is 11.52 MHz with a base rate of 23.04 MHz (x2 decimation)
Found PLMN: Id=00101, TAC=1
Random Access Transmission: seq=51, tti=341, ra-rnti=0x2
Random Access Complete: c-rnti=0x46, ta=0
RRC Connected
Network attach successful. IP: 10.45.0.2
nTpJ 20/11/2023 6:55:8 T2:0

```

Figura 59. Comando de verificación de funcionamiento - 1

Posteriormente, utilizando otra sesión SSH observar si se creó una nueva interfaz nombrada como “tun_srsue”. Para esto se apoyó con el comando “ip a”.

```
ubuntu@SRSRAN:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:b6:54:e3 brd ff:ff:ff:ff:ff:ff
    inet 203.0.113.166/24 brd 203.0.113.255 scope global dynamic ens3
        valid_lft 77293sec preferred_lft 77293sec
    inet6 fe80::f816:3eff:feb6:54e3/64 scope link
        valid_lft forever preferred_lft forever
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:40:b3:49 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.122/24 brd 192.168.0.255 scope global dynamic ens4
        valid_lft 77835sec preferred_lft 77835sec
    inet6 fe80::f816:3eff:fe40:b349/64 scope link
        valid_lft forever preferred_lft forever
6: tun_srsue: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.45.0.2/24 scope global tun_srsue
        valid_lft forever preferred_lft forever
```

Figura 60. Comando de verificación de funcionamiento - 2

Finalmente, podemos comprobar la conectividad de UE con un servidor externo mediante el comando **ping**. Asimismo, se valida el funcionamiento del protocolo de encapsulamiento (GTP) mediante el encapsulamiento del tráfico usando el *bearer* de datos.

```
ubuntu@SRSRAN:~$ ping google.com -I tun_srsue -c 5
PING google.com (64.233.186.139) from 10.45.0.2 tun_srsue: 56(84) bytes of data.
64 bytes from cb-in-f139.1e100.net (64.233.186.139): icmp_seq=1 ttl=50 time=724 ms
64 bytes from cb-in-f139.1e100.net (64.233.186.139): icmp_seq=2 ttl=50 time=782 ms
64 bytes from cb-in-f139.1e100.net (64.233.186.139): icmp_seq=3 ttl=50 time=747 ms
64 bytes from cb-in-f139.1e100.net (64.233.186.139): icmp_seq=4 ttl=50 time=852 ms
64 bytes from cb-in-f139.1e100.net (64.233.186.139): icmp_seq=5 ttl=50 time=387 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 386.617/698.216/851.607/161.660 ms
```

Figura 61. ping a servidor de google.com

48	68.182578	10.45.0.2	64.233.190.100	GTP <ICMP>	134 Echo (ping) request	id=0x0001, seq=1/256, ttl=64 (reply in 49)
49	68.252401	64.233.190...	10.45.0.2	GTP <ICMP>	134 Echo (ping) reply	id=0x0001, seq=1/256, ttl=50 (request in 48)
50	68.852628	10.45.0.2	64.233.190.100	GTP <ICMP>	134 Echo (ping) request	id=0x0001, seq=2/512, ttl=64 (reply in 51)
51	68.897626	64.233.190...	10.45.0.2	GTP <ICMP>	134 Echo (ping) reply	id=0x0001, seq=2/512, ttl=50 (request in 50)
52	69.979162	10.45.0.2	64.233.190.100	GTP <ICMP>	134 Echo (ping) request	id=0x0001, seq=3/768, ttl=64 (reply in 53)
53	70.024504	64.233.190...	10.45.0.2	GTP <ICMP>	134 Echo (ping) reply	id=0x0001, seq=3/768, ttl=50 (request in 52)

Figura 62. Validación del protocolo GTP

5.2 Attachment y PDU Session Establishment - 5G

De forma similar al caso anterior, a continuación, se muestra en la siguiente tabla en la que se podrá apreciar los componentes involucrados en este escenario:

TABLA XV. LISTADO DE INTERFACES PARA EL ESCENARIO DE 5G

VM	Componentes de Core 5G	Interfaces	Función
5GC (Open5GS)	AMF	127.0.0.5:7777	SBI
		192.168.0.111:38412	NGAP
	AUSF	127.0.0.11:7777	SBI
	UDM	127.0.0.12:7777	SBI
	UDR	127.0.0.15:7777	SBI
	NRF	127.0.0.10:7777	SBI
	PCF	127.0.0.13:7777	SBI
	SMF	127.0.0.4:7777	SBI
		192.168.0.111:8805	PFCP
		127.0.0.4:2123	GTPC
192.168.0.111:2152		GTPU	
UPF (Open5GS)	UPF	192.168.0.112:8805	PFCP
		192.168.0.112:2152	GTPU
VM	Componentes de RAN	Interfaces	Función
gNB (srsRAN)	gNB	192.168.0.121	TX/RX
UE (srsRAN)	UE	192.168.0.122	TX/RX

Asimismo, según [44]-[46] para este escenario se hará estudio de los procedimientos fundamentales de una red 5G descritos en las siguientes imágenes en donde se muestra sus respectivos *call flows*

5G Network Registration Call Flow

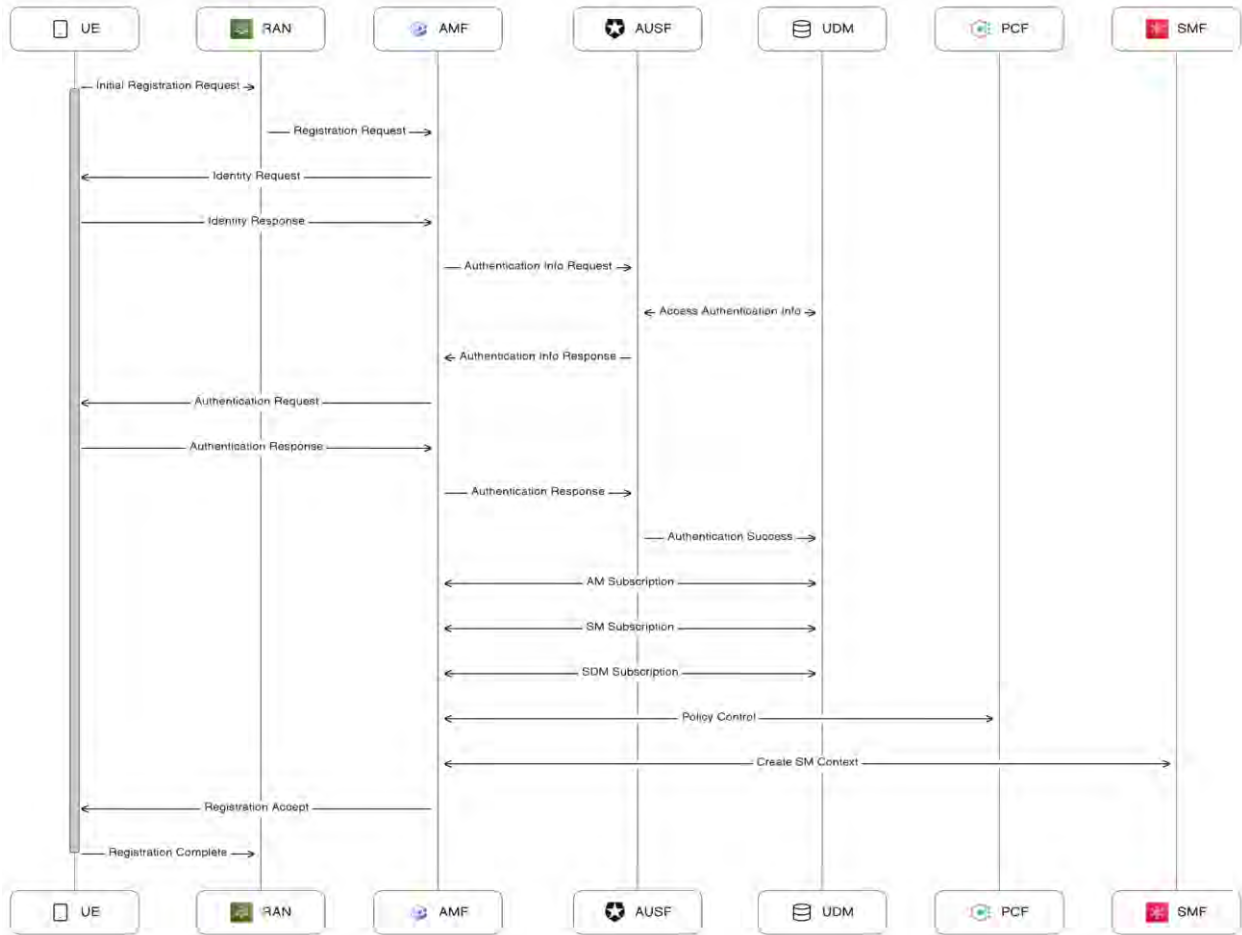


Figura 63. Callflow de attachment 5G

5G Network Service Request Flow

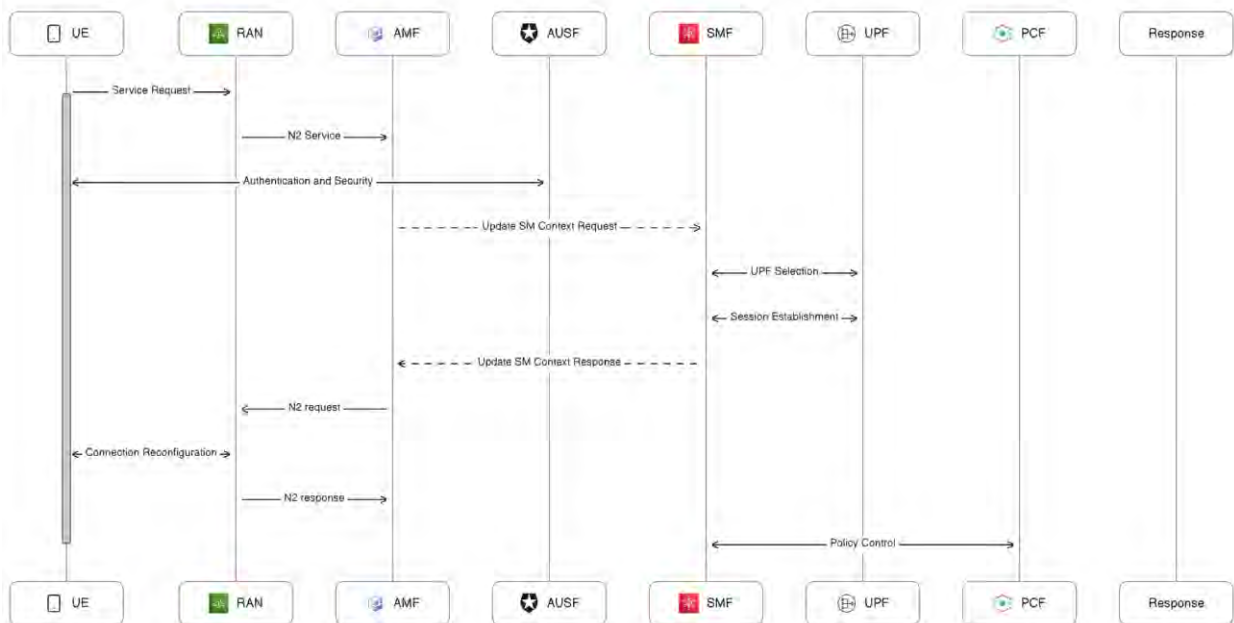


Figura 64. Callflow de service request 5G

5G PDU Establishment Flow

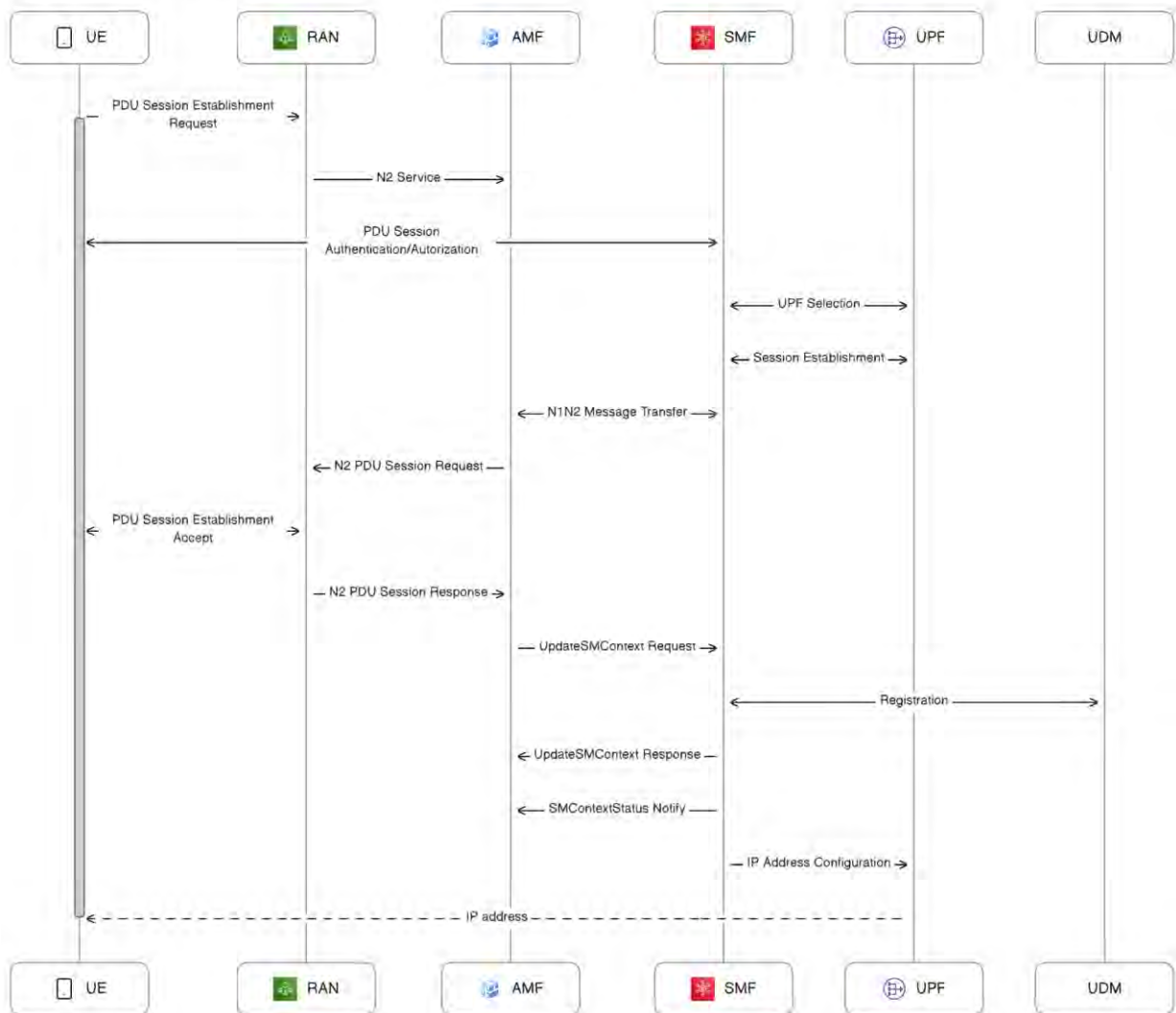


Figura 65. Callflow de PDU Establishment 5G

5.2.1 Core

A diferencia, del caso de 4G en 5G las comunicaciones se realizan por medio del protocolo HTTP2; sin embargo, la herramienta de Wireshark en un inicio no permite la identificación de dichos paquetes (dado que los reconocen como si fueran paquetes TCP) esto se puede evidenciar en la siguiente figura en donde se ilustra este inconveniente:

34	3.586058	:::1	:::1	TCP	86	34120 → 7777 [ACK] Seq=704 Ack=83 Win=512 Len=0 TSval=1065223394 TSecr=1065223394
35	3.591827	127.0.1.10	127.0.0.1	TCP	101	7777 → 40650 [PSH, ACK] Seq=1 Ack=235 Win=512 Len=35 TSval=1099967940 TSecr=34696
36	3.592212	127.0.0.1	127.0.1.10	TCP	66	40650 → 7777 [ACK] Seq=235 Ack=36 Win=512 Len=0 TSval=346968237 TSecr=1099967940
37	4.137621	127.0.0.1	127.0.1.10	TCPCL	162	Contact Header, Contact Header
38	4.139511	127.0.0.1	127.0.1.10	TCP	204	40534 → 7777 [PSH, ACK] Seq=97 Ack=1 Win=512 Len=138 TSval=346968785 TSecr=109995
39	4.140965	127.0.1.10	127.0.0.1	TCP	66	7777 → 40534 [ACK] Seq=1 Ack=235 Win=512 Len=0 TSval=1099968489 TSecr=346968783

Figura 66. Decodificación de HTTP2 en Wireshark

Por lo que, para resolver dicho inconveniente se usó la herramienta [5G Trace Visualizer](#) de Deutsche Telekom de forma que se pudiera apreciar los *call flows* decodificados de forma correcta. Asimismo, antes de realizar el análisis es importante aclarar que en [19] dentro de sus características de funcionamiento de Open5GS existe el uso de un *service communication proxy* (SCP) para comunicación indirecta a través de las SBIs de cada componente del core de forma que se tiene un esquema de direccionamiento tal como se muestra en la siguiente figura:

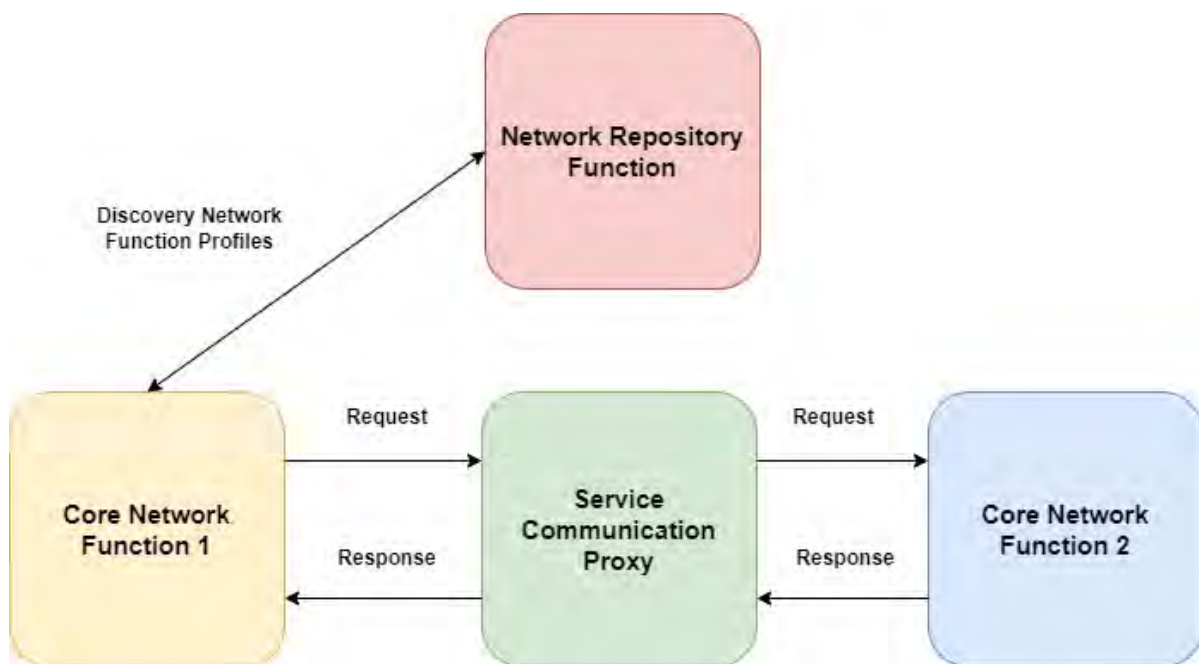


Figura 67. Modelo de comunicación indirecta mediante el uso de SCP

Así, con las siguientes consideraciones, y teniendo en cuenta que la dirección IP de la SBI del SCP es **127.0.1.10 (localhost)** se puede apreciar en un inicio para el caso del *call flow* de registración lo siguiente:

5.2.1.1 Registración

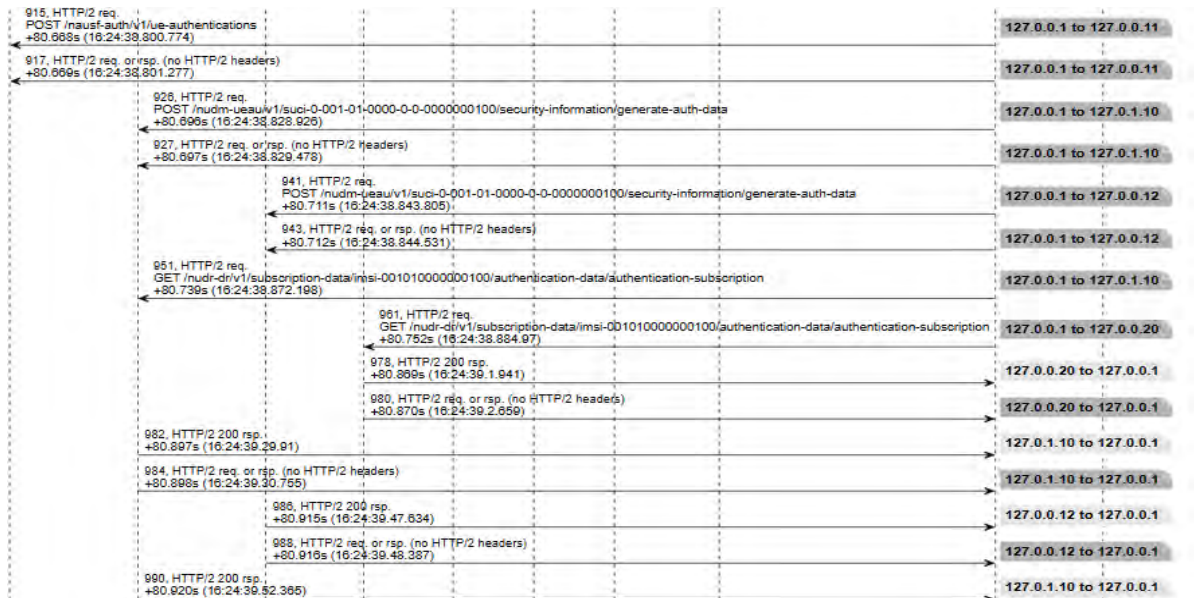


Figura 68. Proceso de autenticación parte 1

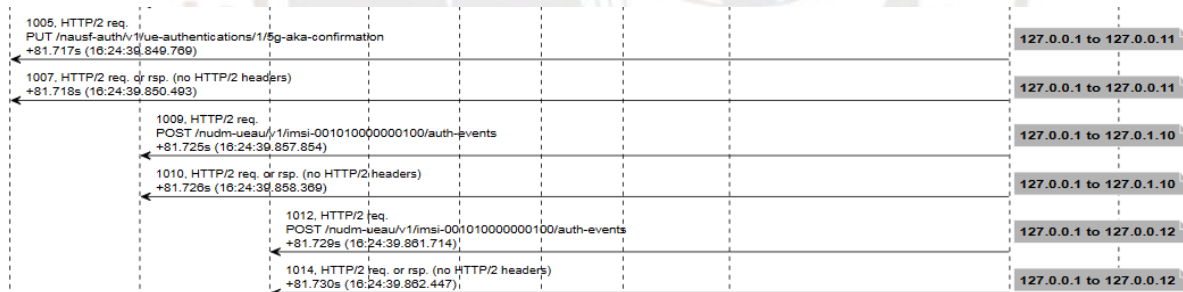


Figura 69. Proceso de autenticación parte 2

Una vez terminado el proceso de autenticación del UE, la siguiente etapa de la registración consiste en la suscripción y obtención de la data del usuario en el UDM; así como en la base de datos (UDR):

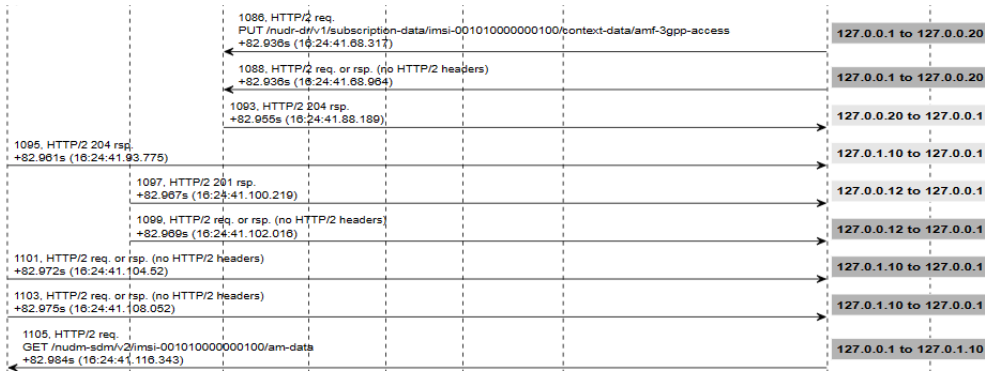


Figura 70. Proceso de suscripción y obtención de data parte 1

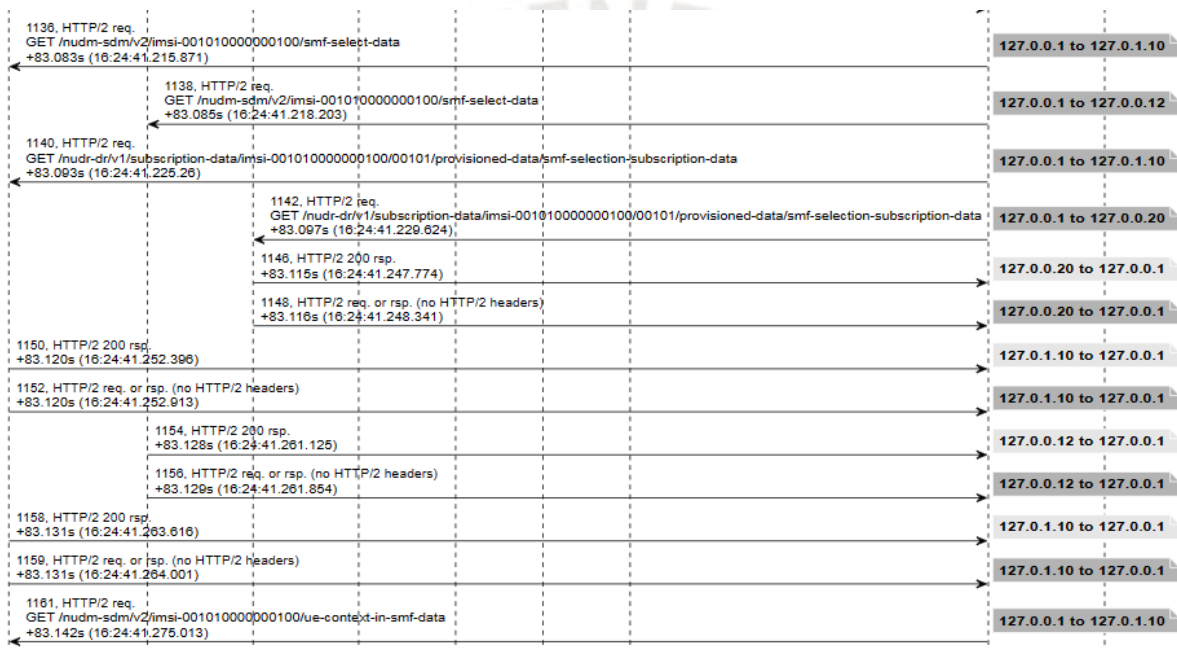


Figura 71. Proceso de suscripción y obtención de data parte 2

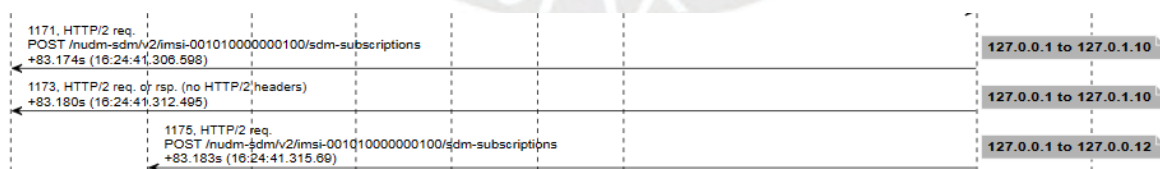


Figura 72. Proceso de suscripción y obtención de data parte 3

Teniendo ya la etapa de obtención y suscripción de data (SDM y UECM) completada, lo siguiente que se realiza es la actualización de las políticas de control:



Figura 73. Aplicación de políticas

Por su parte, teniendo la actualización de políticas actualizadas en el PCF. La siguiente etapa por realizar es el establecimiento de una sesión que permita al usuario solicitudes de PDU *Session Establishment* posteriores y esto se evidencia en la siguiente imagen:

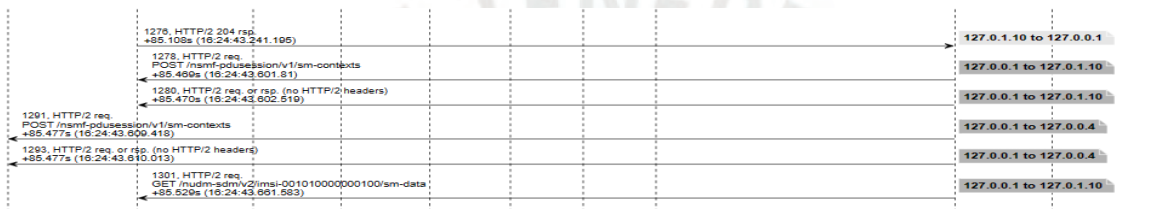


Figura 74. Creación del SM Context

Por último, el mensaje de registración, dado que se envía a través de la interfaz que se expone hacia el gNodeB se mostrará en secciones posteriores.

5.2.1.2 Service Request

Por su parte, para la selección de UPF se tiene la siguiente captura en donde se realiza la comunicación por medio de la interfaz externa que comunica al *core* de control plane con el UPF:

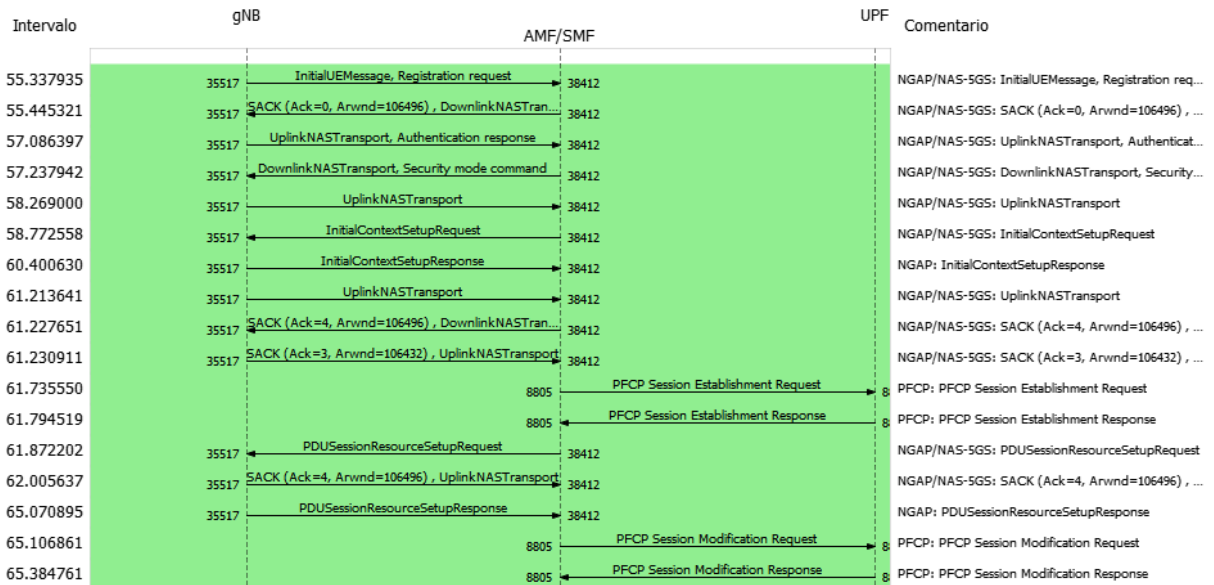


Figura 75. Selección del UPF y establecimiento de la sesión

Dado que se realizó previamente la registración y, adicionalmente, como la sesión del UE se encuentra **activa**, esto es se tiene al UE en estado *RM-registered* y *CM-Connected*, debido al proceso anterior de *attachment*, se realizan las consultas correspondientes a la información de sesión relacionado al SMF por lo que el paso en donde este se cambie y pase el estado del UE *IDLE (RRC-inactive)* a Activo (*RRC-connected*) no es necesario tal como se muestra en la siguiente imagen:

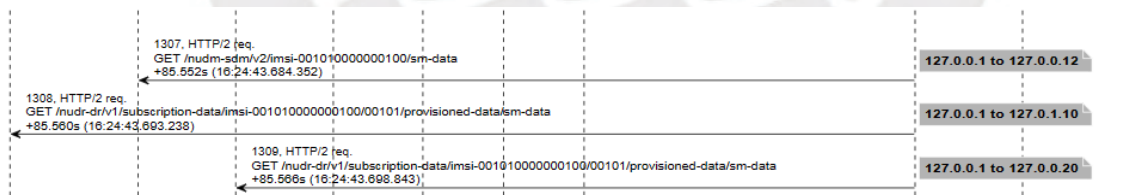


Figura 76. Revisión de parámetros de sesión

Asimismo, se puede apreciar el establecimiento de políticas:



Figura 77. Asignación de políticas

5.2.1.3 PDU Session establishment

A continuación, se muestra el mensaje de solicitud para establecimiento de sesión provista por el UE:

1279	86.286187	127.0.1.10	127.0.0.1	TCP	66 7777 → 40636 [ACK] Seq=3211 Ack=4996 Win=511 Len=0 TSval=1100050635 TSecr=347050931
1280	86.286672	127.0.0.1	127.0.1.10	HTTP2/JSON/NAS-5GS	1186 DATA[87], JavaScript Object Notation (application/json), PDU session establishment request
1281	86.286773	127.0.1.10	127.0.0.1	TCP	66 7777 → 40636 [ACK] Seq=3211 Ack=6116 Win=503 Len=0 TSval=1100050635 TSecr=347050932

Figura 78. Solicitud para PDU session establishment

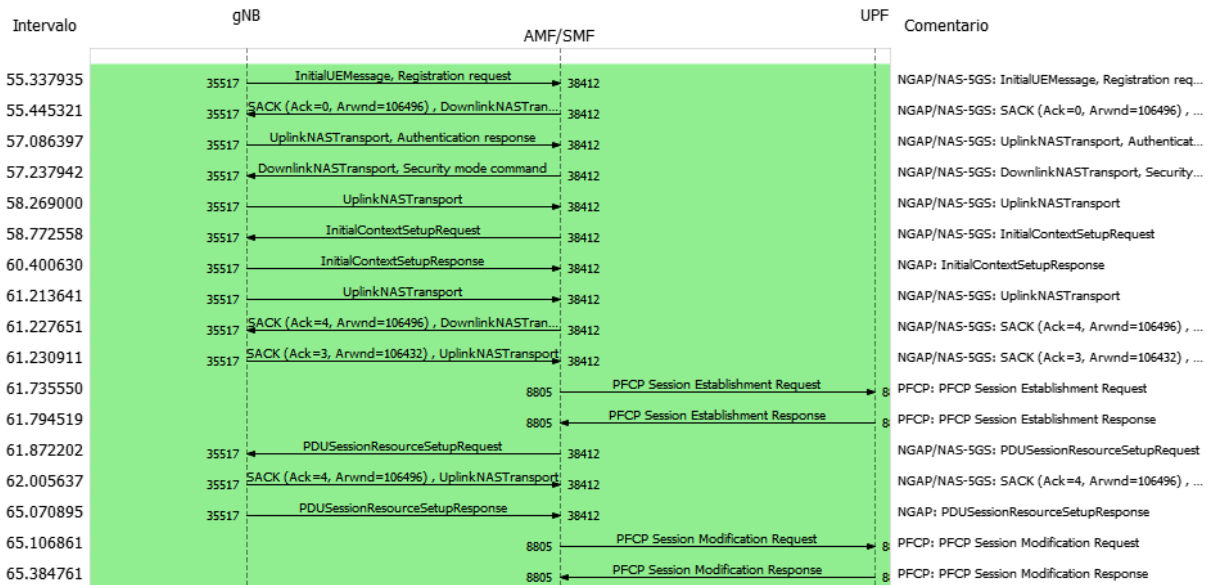


Figura 79. Solicitud para PDU session establishment - Interfaz del UPF

Por otra parte, la solicitud de mensaje N1N2 se muestra a continuación:

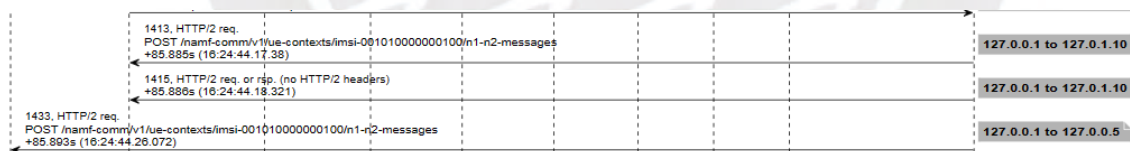


Figura 80. Solicitud de mensaje N1 N2

Por otra parte, se realiza la actualiza los parámetros de sesión para así añadir una sesión de datos tal como se muestra a continuación:

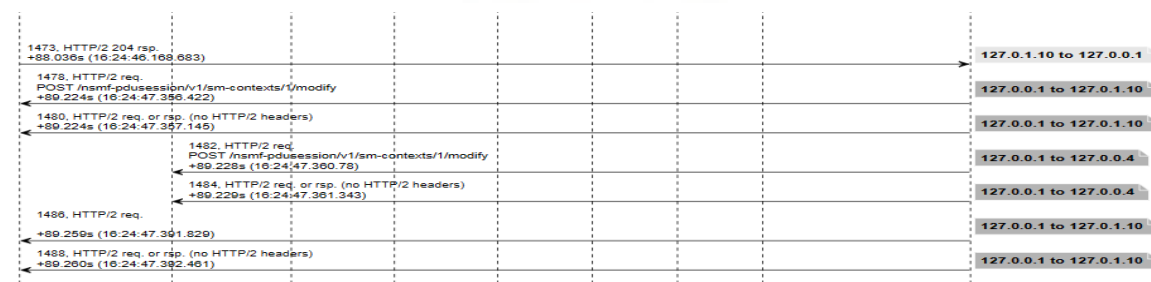


Figura 81. Actualización de la sesión de datos

Así también como el caso de la registración:

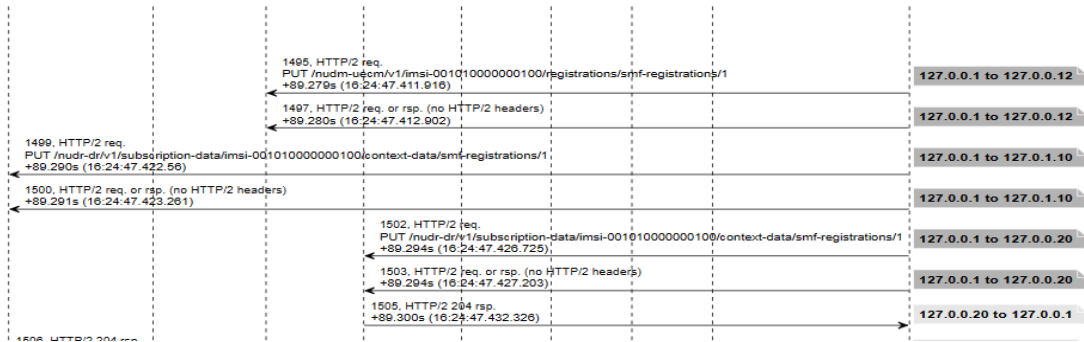


Figura 82. Registración

Finalmente, se realiza la configuración de IP en el UPF y la asignación de esta al UE por medio de la modificación de la sesión haciendo uso del protocolo PFCP.

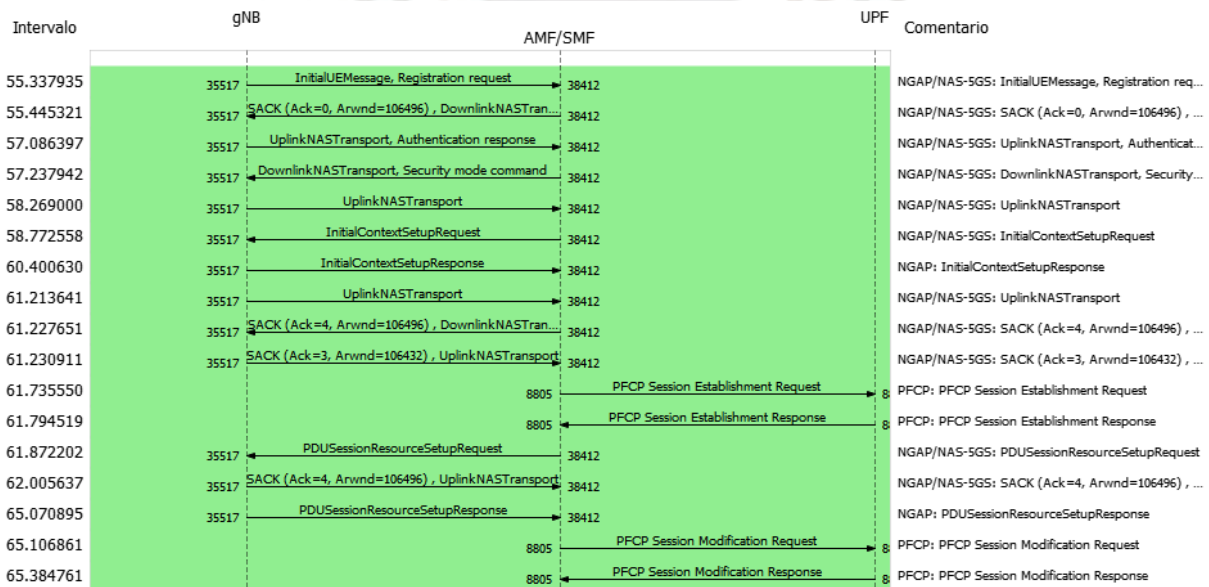


Figura 83. Configuración de IP y asignación

Como se evidencia en los *call flows* observados en la figura 73, SMF establece la sesión en UPF correspondiente para posteriormente situar los recursos de la PDU *session* hacia el UE.

```

33 59.522320 AM.. UPF PFCP          112 PFCP Session Modification Request
.... 0 = EDRT (Eliminate Duplicate Packets for Redundant Transmission): False
Update Forwarding Parameters : [Grouped IE]
  IE Type: Update Forwarding Parameters (11)
  IE Length: 32
  Destination Interface : Access
    IE Type: Destination Interface (42)
    IE Length: 1
    0000 .... = Spare: 0
    .... 0000 = Interface: Access (0)
  Network Instance : internet
    IE Type: Network Instance (22)
    IE Length: 9
    Network Instance: internet
  Outer Header Creation :
    IE Type: Outer Header Creation (84)
    IE Length: 10
    Outer Header Creation Description: GTP-U/UDP/IPv4 (256)
    TEID: 0x00000001
    IPv4 Address: gNB (192.168.0.121)
[Response In: 34]
    
```

Figura 84. Configuración de DN

5.2.4 gNodeB - UE

En esta sección observaremos los paquetes entre el UE y gNB donde se identifican los diferentes procesos anteriormente relacionados, como, por ejemplo, el proceso de *Registration request*, el cual inicializa el proceso de *attachment* y *PDU Session Establishment*.

No.	Time	Source	Destin	Protocol	Length	Info
1	0.000..	NR	RRC	42 RRC Setup Request (Short BSR LCG ID=0 BS=1) (Padding 1 bytes)		
2	0.378..	NR	RRC	356 RRC Setup (Padding 2 bytes)		
3	2.290..	NR	RRC/NAS-SGS	543 RRC Setup Complete, Registration request [33-bytes] (Short BSR LCG ID=0 BS=0) (Padding 472 bytes)		
4	2.340..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=1 (Padding 63 bytes)			
5	2.753..	NR	RRC/NAS-SGS	103 DL Information Transfer, Authentication request [51-bytes] (Padding 16 bytes)		
6	3.860..	NR	RRC/NAS-SGS	559 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=1 , UL Information Transfer, Authentication response [30-bytes] (Short BSR LCG ID=0 BS=5) (Padding 48..)		
7	3.873..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=2 (Padding 63 bytes)			
8	4.090..	NR	RRC/NAS-SGS	103 DL Information Transfer, Security mode command [28-bytes] (Padding 39 bytes)		
9	4.392..	MAC-NR	107 UL-SCH (Short BSR LCG ID=0 BS=0) (Padding 73 bytes)			
10	4.987..	NR	RRC/NAS-SGS	575 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=2 , UL Information Transfer [65-bytes] (Short BSR LCG ID=0 BS=7) (Padding 467 bytes)		
11	4.994..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=3 (Padding 63 bytes)			
12	5.163..	MAC-NR	107 UL-SCH (Short BSR LCG ID=0 BS=0) (Padding 73 bytes)			
13	5.621..	NR	RRC	103 Security Mode Command [9-bytes] (Padding 58 bytes)		
14	6.113..	NR	RRC	575 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=3 , Security Mode Complete [8-bytes] (Short BSR LCG ID=0 BS=2) (Padding 524 bytes)		
15	6.139..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=4 UEId=1 [DL] [AM] SRB:1 [DATA] SN=3 [53-bytes.. (Padding 6 bytes)			
16	6.166..	NR	RRC/NAS-SGS	100 DL Information Transfer (Padding 60 bytes)		
17	6.277..	MAC-NR	107 UL-SCH (Short BSR LCG ID=0 BS=0) (Padding 73 bytes)			
18	6.530..	NR	RRC/NAS-SGS	575 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=4 , UL Information Transfer [19-bytes] , UL Information Transfer [44-bytes] (Short BSR LCG ID=0..)		
19	6.535..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=6 (Padding 63 bytes)			
20	6.569..	NR	RRC/NAS-SGS	100 DL Information Transfer [49-bytes] (Padding 15 bytes)		
21	6.735..	MAC-NR	107 UL-SCH (Short BSR LCG ID=0 BS=0) (Padding 73 bytes)			
22	7.180..	NR	RRC	103 UE Capability Enquiry [14-bytes] (Padding 53 bytes)		
23	7.369..	NR	RRC/NAS-SGS	575 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=5 , UL Information Transfer [19-bytes] (Short BSR LCG ID=0 BS=4) (Padding 513 bytes)		
24	7.385..	RLC-NR	100 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=7 (Padding 63 bytes)			
25	7.594..	NR	RRC	107 UEId=1 [UL] [AM] SRB:1 [CONTROL] ACK_SN=6 , UE Capability Information [36-bytes] (Short BSR LCG ID=0 BS=6) (Padding 28 bytes)		
26	7.648..	RLC-NR	103 UEId=1 [DL] [AM] SRB:1 [CONTROL] ACK_SN=8 (Padding 66 bytes)			
27	7.971..	NR	RRC/NAS-SGS	180 RRC Reconfiguration [119-bytes] (Padding 25 bytes)		

Figura 85. Configuración de IP y asignación

5.2.5 Prueba de funcionamiento

En esta sección estableceremos tanto los pasos a seguir para ejecutar los procesos que emulan los diferentes elementos de una red 5G y, asimismo, verificar el correcto funcionamiento de estos.

5.2.5.1 Open5gs

A continuación, se mostrará el comando para ejecutar los procesos del *core 5G*.

```
ubuntu@5GC:~$ ./run5GC.sh
```

Figura 86. Comando de inicialización de core 5G

Para la verificación debemos observar el siguiente *output* con referencia al comando anterior.

```
open5gs 409454 32.0 1.1 2662460 43368 ? Ssl 17:29 0:03 /usr/bin/open5gs-smfd -c /etc/open5gs/smf.yaml
open5gs 409470 5.7 0.4 153564 17960 ? Ssl 17:29 0:00 /usr/bin/open5gs-amfd -c /etc/open5gs/amf.yaml
open5gs 409484 22.8 0.4 2408500 16848 ? Ssl 17:29 0:02 /usr/bin/open5gs-hssd -c /etc/open5gs/hss.yaml
open5gs 409503 23.8 0.4 2407012 16536 ? Ssl 17:29 0:01 /usr/bin/open5gs-pcrfd -c /etc/open5gs/pcrf.yaml
open5gs 409542 11.4 0.4 131968 15736 ? Ssl 17:29 0:00 /usr/bin/open5gs-nrfd -c /etc/open5gs/nrf.yaml
open5gs 409582 19.0 0.4 144732 16412 ? Ssl 17:29 0:01 /usr/bin/open5gs-scpd -c /etc/open5gs/scp.yaml
open5gs 409630 9.4 0.4 133672 15732 ? Ssl 17:29 0:00 /usr/bin/open5gs-ausfd -c /etc/open5gs/ausf.yaml
open5gs 409638 12.7 0.3 140208 15044 ? Ssl 17:29 0:00 /usr/bin/open5gs-udmd -c /etc/open5gs/udm.yaml
open5gs 409643 21.0 0.5 183320 19944 ? Ssl 17:29 0:00 /usr/bin/open5gs-pcfd -c /etc/open5gs/pcf.yaml
open5gs 409649 16.3 0.3 131948 14724 ? Ssl 17:29 0:00 /usr/bin/open5gs-nssf -c /etc/open5gs/nssf.yaml
open5gs 409656 21.0 0.3 139428 14960 ? Ssl 17:29 0:00 /usr/bin/open5gs-bsfd -c /etc/open5gs/bsf.yaml
open5gs 409662 66.0 0.4 167308 18884 ? Ssl 17:29 0:00 /usr/bin/open5gs-udrd -c /etc/open5gs/udr.yaml
```

Figura 87. Output del comando de inicialización

5.2.5.2 UPF

A continuación, se mostrará el comando para ejecutar el proceso relacionado al UPF.

```
ubuntu@UPF:~$ ./runUPF.sh
```

Figura 88. Comando de inicialización de UPF

Para la verificación debemos observar el siguiente *output* con referencia al comando anterior.

```
open5gs 19543 0.0 0.3 77320 12768 ? Rs 17:30 0:00 /usr/bin/open5gs-upfd -c /etc/open5gs/upf.yaml
```

Figura 89. Output del comando de inicialización de UPF

5.2.5.2 SRSRAN - gNB

Para ejecutar el proceso de gNB debemos ejecutar el comando observado en la siguiente figura.

```

ubuntu@SRSRAN-NR:~/srsRAN_Project/build/apps/gnb$ sudo ./gnb -c gnb.yaml
[sudo] password for ubuntu:
Lower PHY in executor blocking mode.

--== srsRAN gNB (commit 374200dee) ==--

Connecting to AMF on 192.168.0.111:38412
Available radio types: zmq.
Cell pci=1, bw=10 MHz, dl_arfcn=368500 (n3), dl_freq=1842.5 MHz, dl_ssb_arfcn=368410, ul_freq=1747.5 MHz

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

```

Figura 90. Comando de inicialización de gNB

5.2.5.2 SRSRAN - UE

En el caso de la ejecución del UE, se ejecutará el siguiente comando visualizado a continuación:

```

ubuntu@SRSRAN:~/srsRAN_4G/build/srsue$ sudo ./src/srsue ue_5G.conf
Reading configuration file ue_5G.conf...

Built in Release mode using commit fa56836b1 on branch master.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://192.168.0.122:2001,rx_port=tcp://192.168.0.121:2000,base_srate=11.52e6
Supported RF device list: zmq file
CHx base_srate=11.52e6
Current sample rate is 1.92 MHz with a base rate of 11.52 MHz (x6 decimation)
CH0 rx_port=tcp://192.168.0.121:2000
CH0 tx_port=tcp://192.168.0.122:2001
Current sample rate is 11.52 MHz with a base rate of 11.52 MHz (x1 decimation)
Current sample rate is 11.52 MHz with a base rate of 11.52 MHz (x1 decimation)
Waiting PHY to initialize ... done!
Attaching UE...
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0x39, tti=334
Random Access Complete. c-rnti=0x4601, ta=0
RRC Connected
PDU Session Establishment successful. IP: 10.45.0.2
RRC NR reconfiguration successful.

```

Figura 91. Comando de inicialización de UE

Para el caso de verificación de funcionamiento lo dividiremos en tres secciones. La primera, observada en la figura anterior con el *output*. El segundo se evaluará la creación de la interfaz utilizada por emulador y posteriormente ejecutar el comando *ping* a un servidor externo, servidor de google. Finalmente, se verificará que el protocolo GTP encapsula los paquetes ICMP utilizados en el comando *ping*.

```

ubuntu@SRSHAN:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:f4:68:e9 brd ff:ff:ff:ff:ff:ff
    inet 203.0.113.79/24 brd 203.0.113.255 scope global dynamic ens3
        valid_lft 84979sec preferred_lft 84979sec
    inet6 fe80::f816:3eff:fe4:68e9/64 scope link
        valid_lft forever preferred_lft forever
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:16:3e:89:0f:9e brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.122/24 brd 192.168.0.255 scope global dynamic ens4
        valid_lft 84986sec preferred_lft 84986sec
    inet6 fe80::f816:3eff:fe89:f9e/64 scope link
        valid_lft forever preferred_lft forever
4: tun_srsue: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.45.0.2/24 scope global tun_srsue
        valid_lft forever preferred_lft forever
    
```

Figura 92. Verificación de creación de interfaz tun_srsue

```

ubuntu@SRSRAN:~$ ping google.com -I tun_srsue -c 10
PING google.com (64.233.186.100) from 10.45.0.2 tun_srsue: 56(84) bytes of data.
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=1 ttl=50 time=1138 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=2 ttl=50 time=747 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=3 ttl=50 time=538 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=4 ttl=50 time=1371 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=5 ttl=50 time=370 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=6 ttl=50 time=1065 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=7 ttl=50 time=460 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=8 ttl=50 time=742 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=9 ttl=50 time=2205 ms
64 bytes from cb-in-f100.1e100.net (64.233.186.100): icmp_seq=10 ttl=50 time=1226 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9083ms
rtt min/avg/max/mdev = 370.461/986.286/2204.606/518.917 ms, pipe 2
    
```

Figura 93. Verificación de conectividad a través del comando ping

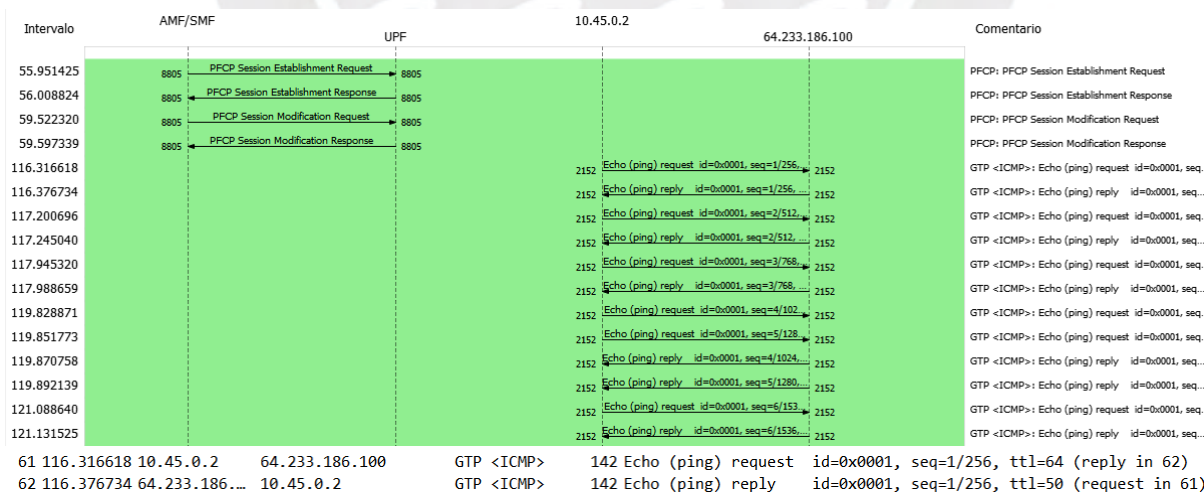


Figura 94. Verificación del protocolo GTP como encapsulador del paquete ICMP

5.3 DN Corporativo

Por su parte, este escenario tendrá a continuación los siguientes componentes junto con sus respectivas funciones como se muestra en la siguiente tabla:

TABLA XVI. LISTADO DE INTERFACES PARA EL ESCENARIO DE DN CORPORATIVO

VM	Componentes de Core 5G	Interfaces	Función
5GC (Open5GS)	AMF	127.0.0.5:7777	SBI
		192.168.0.111:38412	NGAP
	AUSF	127.0.0.11:7777	SBI
	UDM	127.0.0.12:7777	SBI
	UDR	127.0.0.15:7777	SBI
	NRF	127.0.0.10:7777	SBI
	PCF	127.0.0.13:7777	SBI
	SMF	127.0.0.4:7777	SBI
		192.168.0.111:8805	PFCP
		127.0.0.4:2123	GTPC
192.168.0.111:2152		GTPU	
UPF (Open5GS)	UPF	192.168.0.112:8805	PFCP
		192.168.0.112:2152	GTPU
VM	Componentes de RAN	Interfaces	Función
gNB (UERANSIM)	gNB	192.168.0.131	TX/RX
UE (UERANSIM)	UE	192.168.0.132	TX/RX

Para este escenario, es importante considerar los procedimientos antes vistos en la sección correspondiente a los *call flows* base para 5G. Con esto en cuenta se hará el análisis con respecto al establecimiento de la sesión de acuerdo con el DN asignado por UE.

5.3.1 Core

A nivel de *core* se tiene la comunicación entre el UPF y el SMF, así como la conexión entre el *core* y el UE en donde se puede evidenciar lo siguiente:

70	86.614496	UE	Core/SMF	NGAP/NAS-SGS	138	InitialUEMessage, Registration request
71	86.820307	Core/SMF	UE	SCTP	62	SACK (Ack=0, Arwnd=106496) Registración UE 1
72	86.983768	Core/SMF	UE	NGAP/NAS-SGS	130	DownlinkNASTransport, Authentication request
73	87.051332	UE	Core/SMF	NGAP/NAS-SGS	146	SACK (Ack=0, Arwnd=106496), UplinkNASTransport, Authentication response
74	87.214069	Core/SMF	UE	NGAP/NAS-SGS	126	SACK (Ack=1, Arwnd=106496), DownlinkNASTransport, Security mode command
75	87.270545	UE	Core/SMF	NGAP/NAS-SGS	186	SACK (Ack=1, Arwnd=106496), UplinkNASTransport
76	87.475241	Core/SMF	UE	SCTP	62	SACK (Ack=2, Arwnd=106496)
77	87.840146	Core/SMF	UE	NGAP/NAS-SGS	214	InitialContextSetupRequest
78	87.860745	UE	Core/SMF	NGAP	98	SACK (Ack=2, Arwnd=106496), InitialContextSetupResponse
79	88.063373	Core/SMF	UE	SCTP	62	SACK (Ack=3, Arwnd=106496)
80	88.068516	UE	Core/SMF	NGAP/NAS-SGS	238	UplinkNASTransport, UplinkNASTransport
81	88.088132	Core/SMF	UE	NGAP/NAS-SGS	142	SACK (Ack=5, Arwnd=106496), DownlinkNASTransport
82	88.214212	Core/SMF	UPF	PFCP	58	PFCP Heartbeat Request Attachment Complete UE 1
83	88.226072	UPF	Core/SMF	PFCP	58	PFCP Heartbeat Response
84	88.293255	UE	Core/SMF	SCTP	62	SACK (Ack=3, Arwnd=106496)
85	88.643320	Core/SMF	UPF	PFCP	667	PFCP Session Establishment Request
86	88.687343	UPF	Core/SMF	PFCP	158	PFCP Session Establishment Response
87	88.803635	Core/SMF	UE	NGAP/NAS-SGS	242	PDUSessionResourceSetupRequest
88	89.011465	UE	Core/SMF	SCTP	62	SACK (Ack=4, Arwnd=106496) PDU Session Establishment UE 1
89	89.218728	UE	Core/SMF	NGAP	102	PDUSessionResourceSetupResponse
90	89.263506	Core/SMF	UPF	PFCP	112	PFCP Session Modification Request
91	89.283410	UPF	Core/SMF	PFCP	63	PFCP Session Modification Response

Figura 95. Procedimientos relacionados al UE 1 a nivel de core

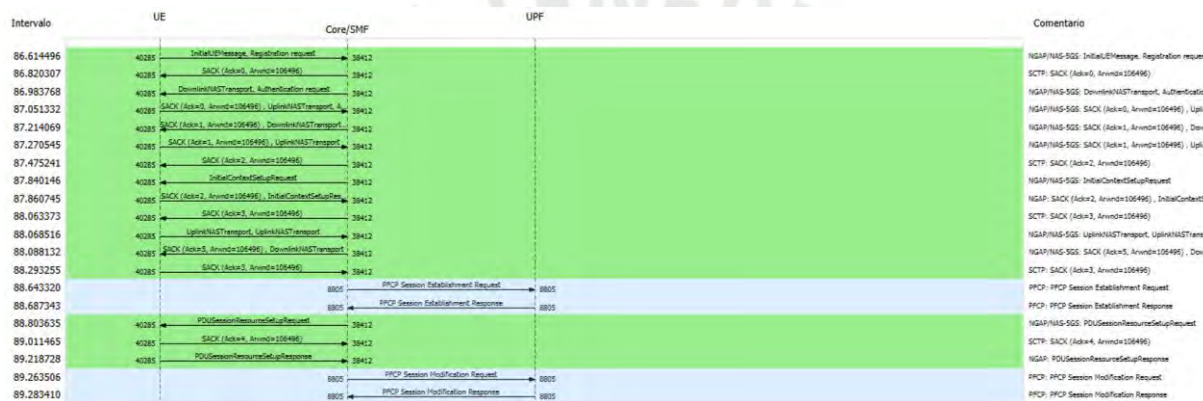


Figura 96. Diagrama de callflow de procesos de core para el UE 1

De forma similar se puede observar este comportamiento para el caso de UE 2 en donde se tiene un DN distinto

295	328.873399	UE	Core/SMF	NGAP/NAS-SGS	138	InitialUEMessage, Registration request
296	329.053705	Core/SMF	UE	NGAP/NAS-SGS	146	SACK (Ack=7, Arwnd=106496), DownlinkNASTransport, Authentication request
297	329.121948	UE	Core/SMF	NGAP/NAS-SGS	146	SACK (Ack=5, Arwnd=106496), UplinkNASTransport, Authentication response
298	329.226031	Core/SMF	UE	NGAP/NAS-SGS	126	SACK (Ack=8, Arwnd=106496), DownlinkNASTransport, Security mode command
299	329.283236	UE	Core/SMF	NGAP/NAS-SGS	186	SACK (Ack=6, Arwnd=106496), UplinkNASTransport
300	329.491225	Core/SMF	UE	SCTP	62	SACK (Ack=9, Arwnd=106496) Registración UE 2
301	329.629704	Core/SMF	UE	NGAP/NAS-SGS	214	InitialContextSetupRequest
302	329.646110	UE	Core/SMF	NGAP	98	SACK (Ack=7, Arwnd=106496), InitialContextSetupResponse
303	329.852054	Core/SMF	UE	SCTP	62	SACK (Ack=10, Arwnd=106496)
304	329.856278	UE	Core/SMF	NGAP/NAS-SGS	234	UplinkNASTransport, UplinkNASTransport
305	329.871439	Core/SMF	UE	NGAP/NAS-SGS	142	SACK (Ack=12, Arwnd=106496), DownlinkNASTransport Attachment Complete UE 2
306	330.076433	UE	Core/SMF	SCTP	62	SACK (Ack=8, Arwnd=106496)
307	330.191248	Core/SMF	UPF	PFCP	639	PFCP Session Establishment Request
308	330.210132	UPF	Core/SMF	PFCP	158	PFCP Session Establishment Response
309	330.245735	Core/SMF	UE	NGAP/NAS-SGS	238	PDUSessionResourceSetupRequest
310	330.453119	UE	Core/SMF	SCTP	62	SACK (Ack=9, Arwnd=106496) PDU Session Establishment UE 2
311	330.673816	UE	Core/SMF	NGAP	102	PDUSessionResourceSetupResponse
312	330.700196	Core/SMF	UPF	PFCP	108	PFCP Session Modification Request
313	330.710062	UPF	Core/SMF	PFCP	63	PFCP Session Modification Response

Figura 97. Procedimientos relacionados al UE 2 a nivel de core

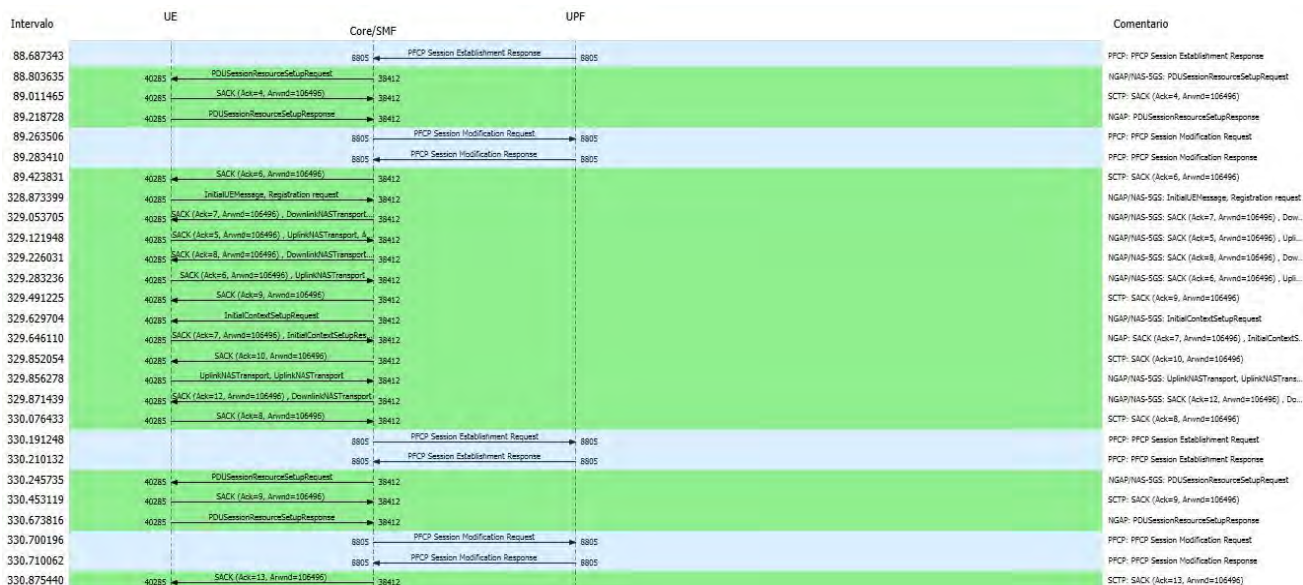


Figura 98. Diagrama de callflow de procesos de core para el UE 2

Por su parte a nivel del UPF se tiene los siguientes resultados, en donde se puede evidenciar el establecimiento de PDU session para ambos UE

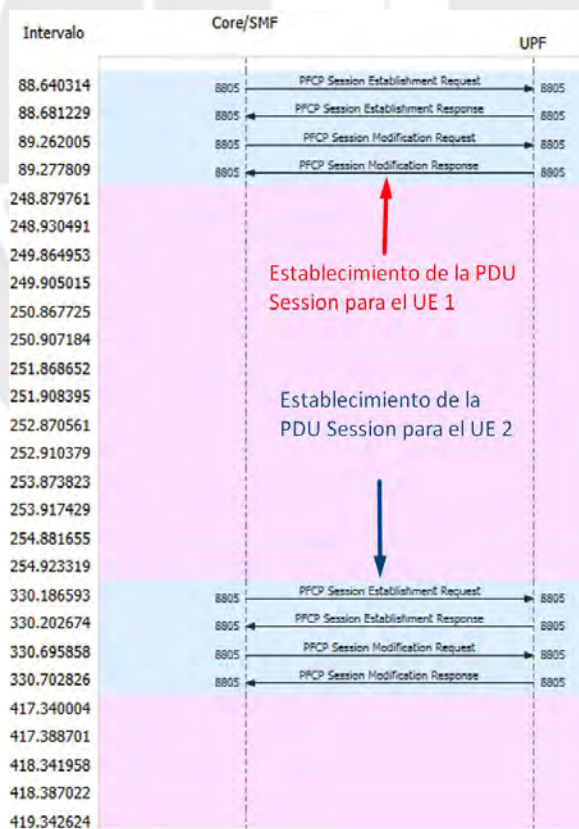


Figura 99. Establecimiento de PDU session en el UPF

Asimismo, dentro de los paquetes en los cuales se tiene la solicitud para el establecimiento se puede evidenciar los campos requeridos para la conexión a la DN respectiva de acuerdo con cada UE dentro del parámetro APN/DNN tal como se aprecia en la siguiente figura:

```

> Node ID : IPv4 address: 192.168.0.111
> F-SEID : SEID: 0x0000000000002a1, IPv4 192.168.0.111
> Create PDR : [Grouped IE]: PDR ID: 1
> Create PDR : [Grouped IE]: PDR ID: 2
> Create PDR : [Grouped IE]: PDR ID: 3
> Create PDR : [Grouped IE]: PDR ID: 4
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 1
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 2
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 3
> Create URR : [Grouped IE]: URR ID: Dynamic by CP 1
> Create QER : [Grouped IE]: QER ID: Dynamic by CP 1
> Create BAR : [Grouped IE]: BAR ID: 1
> PDN Type : IPv4
> User ID :
> APN/DNN : internet

```

DN especificada por el UE 1

Figura 100. DN para el UE 1

```

> Node ID : IPv4 address: 192.168.0.111
> F-SEID : SEID: 0x0000000000002cd, IPv4 192.168.0.111
> Create PDR : [Grouped IE]: PDR ID: 1
> Create PDR : [Grouped IE]: PDR ID: 2
> Create PDR : [Grouped IE]: PDR ID: 3
> Create PDR : [Grouped IE]: PDR ID: 4
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 1
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 2
> Create FAR : [Grouped IE]: FAR ID: Dynamic by CP 3
> Create URR : [Grouped IE]: URR ID: Dynamic by CP 1
> Create QER : [Grouped IE]: QER ID: Dynamic by CP 1
> Create BAR : [Grouped IE]: BAR ID: 1
> PDN Type : IPv4
> User ID :
> APN/DNN : coop

```

DN especificada para el UE 2

Figura 101. DN para el UE 2

5.3.2 gNodeB

Por su parte en el gNodeB pudo conectarse con ambos UEs de forma simultánea:

```

ubuntu@UERANSIM-gNB: ~/l X
[2023-11-22 01:03:49.150] [sctp] [info] SCTP connection established (192
.168.0.111:38412)
[2023-11-22 01:03:49.165] [sctp] [debug] SCTP association setup ascId[3]
[2023-11-22 01:03:49.169] [ngap] [debug] Sending NG Setup Request
[2023-11-22 01:03:49.262] [ngap] [debug] NG Setup Response received
[2023-11-22 01:03:49.273] [ngap] [info] NG Setup procedure is successful
[2023-11-22 01:04:27.775] [rrc] [debug] UE[1] new signal detected
[2023-11-22 01:04:28.035] [rrc] [info] RRC Setup for UE[1]
[2023-11-22 01:04:28.059] [ngap] [debug] Initial NAS message received fr
om UE[1]
[2023-11-22 01:04:29.312] [ngap] [debug] Initial Context Setup Request r
eceived
[2023-11-22 01:04:30.684] [ngap] [info] PDU session resource(s) setup fo
r UE[1] count[1]

```

Figura 102. Conexión exitosa del UE 1

```

ubuntu@UERANSIM-gNB: ~/l X
[2023-11-22 01:04:28.059] [ngap] [debug] Initial NAS message received fr
om UE[1]
[2023-11-22 01:04:29.312] [ngap] [debug] Initial Context Setup Request r
eceived
[2023-11-22 01:04:30.684] [ngap] [info] PDU session resource(s) setup fo
r UE[1] count[1]
[2023-11-22 01:08:27.702] [rrc] [debug] UE[2] new signal detected
[2023-11-22 01:08:30.300] [rrc] [info] RRC Setup for UE[2]
[2023-11-22 01:08:30.321] [ngap] [debug] Initial NAS message received fr
om UE[2]
[2023-11-22 01:08:31.094] [ngap] [debug] Initial Context Setup Request r
eceived
[2023-11-22 01:08:32.130] [ngap] [info] PDU session resource(s) setup fo
r UE[2] count[1]

```

Figura 103. Conexión exitosa del UE 2

5.3.3 UE

Asimismo, en el caso de ambos UEs se tiene la recepción de las direcciones IPs correspondientes a sus respectivas DNs tal como se muestra a continuación:

```

[2023-11-22 01:04:29.365] [nas] [info] Initial Registration is successfu
l
[2023-11-22 01:04:29.371] [nas] [debug] Sending PDU Session Establishmen
t Request
[2023-11-22 01:04:29.398] [nas] [debug] UAC access attempt is allowed fo
r identity[0], category[MO_sig]
[2023-11-22 01:04:29.562] [nas] [debug] Configuration Update Command rec
eived
[2023-11-22 01:04:30.689] [nas] [debug] PDU Session Establishment Accept
received
[2023-11-22 01:04:30.698] [nas] [info] PDU Session establishment is succ
essful PSI[1]
[2023-11-22 01:04:31.645] [app] [info] Connection setup for PDU session[
1] is successful, TUN interface[uesimtun0, 10.45.0.2] is up.

```

Figura 104. Logs de conexión del UE 1

```

4: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu
1400 qdisc fq_codel state UNKNOWN group default qlen 500
  link/none
  inet 10.45.0.2/32 scope global uesimtun0
    valid_lft forever preferred_lft forever
  inet6 fe80::7836:6657:b9dc:4c93/64 scope link stable-privac
y
    valid_lft forever preferred_lft forever
ubuntu@UERANSIM-UE:~$ |

```

Figura 105. Asignación de IP del UE 1

```

[2023-11-22 01:08:31.152] [nas] [info] UE switches to state [MM
-REGISTERED/NORMAL-SERVICE]
[2023-11-22 01:08:31.157] [nas] [debug] Sending Registration Co
mplete
[2023-11-22 01:08:31.160] [nas] [info] Initial Registration is
successful
[2023-11-22 01:08:31.166] [nas] [debug] Sending PDU Session Est
ablishment Request
[2023-11-22 01:08:31.192] [nas] [debug] UAC access attempt is a
llowed for identity[0], category[MO_sig]
[2023-11-22 01:08:31.351] [nas] [debug] Configuration Update Co
mmand received
[2023-11-22 01:08:32.155] [nas] [debug] PDU Session Establishme
nt Accept received
[2023-11-22 01:08:32.168] [nas] [info] PDU Session establishmen
t is successful PSI[1]
[2023-11-22 01:08:32.950] [app] [info] Connection setup for PDU
session[1] is successful, TUN interface[uesimtun1, 10.46.0.2]
is up.

```

Figura 106. Logs de conexión del UE 2

```

5: uesimtun1: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu
1400 qdisc fq_codel state UNKNOWN group default qlen 500
  link/none
  inet 10.46.0.2/32 scope global uesimtun1
    valid_lft forever preferred_lft forever
  inet6 fe80::ef0f:4cf1:7e27:bc61/64 scope link stable-privac
y
    valid_lft forever preferred_lft forever

```

Figura 107. Asignación de IP del UE 2

Por otro lado, se tiene la prueba de conectividad de ambos UE tanto por separado como de forma simultánea:

```
ubuntu@UERANSIM-UE:~$ ping google.com -I uesimtun0 -n
PING google.com (172.217.192.113) from 10.45.0.2 uesimtun0: 56(84) bytes of data.
64 bytes from 172.217.192.113: icmp_seq=1 ttl=50 time=123 ms
64 bytes from 172.217.192.113: icmp_seq=2 ttl=50 time=72.0 ms
64 bytes from 172.217.192.113: icmp_seq=3 ttl=50 time=69.0 ms
64 bytes from 172.217.192.113: icmp_seq=4 ttl=50 time=71.0 ms
64 bytes from 172.217.192.113: icmp_seq=5 ttl=50 time=66.0 ms
64 bytes from 172.217.192.113: icmp_seq=6 ttl=50 time=71.9 ms
64 bytes from 172.217.192.113: icmp_seq=7 ttl=50 time=76.5 ms
```

Figura 108. Prueba de conectividad del UE 1

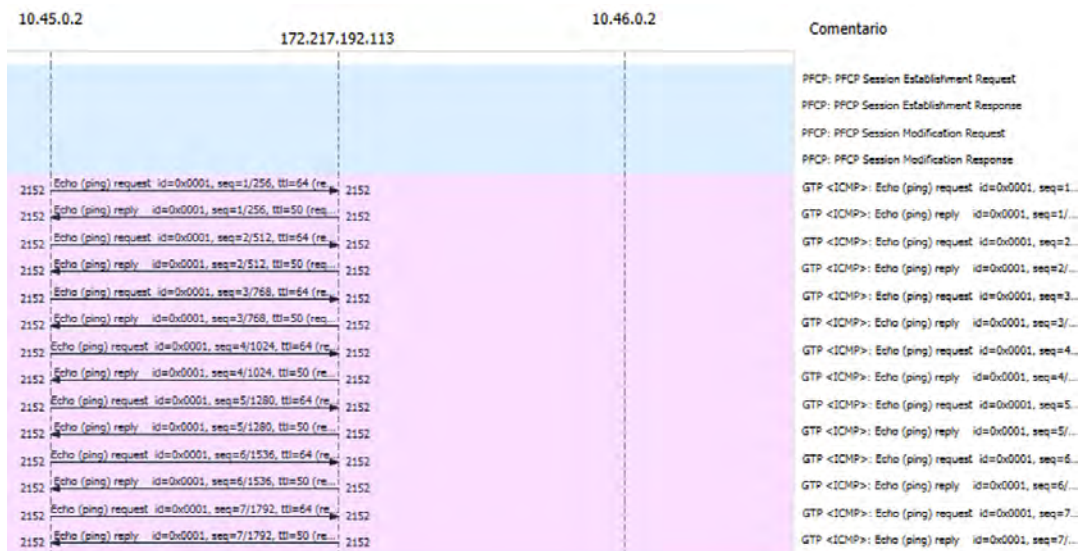


Figura 109. Flujo de tráfico por medio del protocolo GTP-U para el UE 1

```
ubuntu@UERANSIM-UE:~$ ping google.com -I uesimtun1 -n
PING google.com (64.233.190.102) from 10.46.0.2 uesimtun1: 56(84) bytes of data.
64 bytes from 64.233.190.102: icmp_seq=1 ttl=50 time=86.1 ms
64 bytes from 64.233.190.102: icmp_seq=2 ttl=50 time=74.1 ms
64 bytes from 64.233.190.102: icmp_seq=3 ttl=50 time=71.7 ms
64 bytes from 64.233.190.102: icmp_seq=4 ttl=50 time=69.0 ms
64 bytes from 64.233.190.102: icmp_seq=5 ttl=50 time=71.1 ms
64 bytes from 64.233.190.102: icmp_seq=6 ttl=50 time=66.7 ms
64 bytes from 64.233.190.102: icmp_seq=7 ttl=50 time=73.6 ms
64 bytes from 64.233.190.102: icmp_seq=8 ttl=50 time=66.2 ms
64 bytes from 64.233.190.102: icmp_seq=9 ttl=50 time=70.8 ms
```

Figura 110. Prueba de conectividad del UE 2



Figura 111. Flujo de tráfico por medio del protocolo GTP-U para el UE 2

```

ubuntu@UERANSIM-UE: ~
valid_lft forever preferred_lft forever
inet6 fe80::7836:6657:b9dc:4c93/64 scope link stable-privac
y
valid_lft forever preferred_lft forever
ubuntu@UERANSIM-UE:~$ ping google.com -I uesimtun0 -n
PING google.com (64.233.190.101) from 10.45.0.2 uesimtun0: 56(8
4) bytes of data.
64 bytes from 64.233.190.101: icmp_seq=1 ttl=99 time=59.9 ms
64 bytes from 64.233.190.101: icmp_seq=2 ttl=99 time=61.1 ms
64 bytes from 64.233.190.101: icmp_seq=3 ttl=99 time=77.6 ms
64 bytes from 64.233.190.101: icmp_seq=4 ttl=99 time=61.0 ms
64 bytes from 64.233.190.101: icmp_seq=5 ttl=99 time=63.5 ms
64 bytes from 64.233.190.101: icmp_seq=6 ttl=99 time=64.1 ms
64 bytes from 64.233.190.101: icmp_seq=7 ttl=99 time=61.9 ms
64 bytes from 64.233.190.101: icmp_seq=8 ttl=99 time=60.1 ms

64 bytes from 64.233.190.102: icmp_seq=9 ttl=50 time=70.8 ms
^C
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8016ms
rtt min/avg/max/mdev = 66.222/72.134/86.054/5.548 ms
ubuntu@UERANSIM-UE:~$ ping google.com -I uesimtun1 -n
PING google.com (64.233.190.102) from 10.46.0.2 uesimtun1: 56(8
4) bytes of data.
64 bytes from 64.233.190.102: icmp_seq=1 ttl=50 time=78.2 ms
64 bytes from 64.233.190.102: icmp_seq=2 ttl=50 time=68.6 ms
64 bytes from 64.233.190.102: icmp_seq=3 ttl=50 time=63.8 ms
64 bytes from 64.233.190.102: icmp_seq=4 ttl=50 time=66.7 ms
64 bytes from 64.233.190.102: icmp_seq=5 ttl=50 time=65.1 ms
64 bytes from 64.233.190.102: icmp_seq=6 ttl=50 time=63.7 ms
64 bytes from 64.233.190.102: icmp_seq=7 ttl=50 time=61.6 ms

```

Figura 112. Prueba de conectividad de forma simultánea

Con esto se evidencia la capacidad de Open5GS para soportar tráfico de forma concurrente de distintos DN's así como la capacidad de UERANSIM para poder escenificar dichos procesos.

5.4 Near RT-RIC

Ahora para este escenario, se hará énfasis a nivel de la comunicación entre la RAN y el *near* RT-RIC por lo que los componentes involucrados se muestran a continuación:

TABLA XVII. LISTADO DE INTERFACES PARA EL ESCENARIO DE O-RAN NEAR RT-RIC

VM	Componentes de RAN	Interfaces	Función
RIC & xApp (FlexRIC)	Near-RT-RIC	192.168.0.123:36421	E2AP
O-RAN gNB (srsRAN)	gNB	192.168.0.121	TX/RX
	gNB-DU-E2	192.168.0.121:48923	E2AP
UE (srsRAN)	UE	192.168.0.122	TX/RX

Asimismo, según [47]-[49] para este escenario se hará estudio de los procedimientos fundamentales para el correcto funcionamiento de una implementación de RIC mediante la interfaz E2; para ello, es necesario comprobar el siguiente *callflow* relacionado para este escenario.

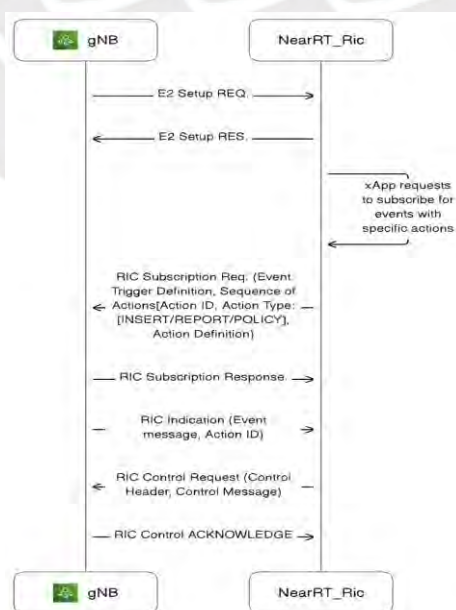


Figura 113. Callflow de la implementación de un RIC a través de la interfaz E2

Sin embargo, en este escenario se utiliza una xAPP de monitoreo lo cual modifica el flujo acotando hasta la sección de RIC *Subscription Response*.

5.4.1 FlexRIC callflow

Este flujo de paquetes es obtenido por la propia herramienta de srsRAN, por lo cual, las direcciones IP de destino y origen no se encuentran establecidas. Sin embargo, debido a *payload* y la propia información del paquete podemos inferir el origen y destino de cada uno.

En la siguiente figura se muestra el *callflow* relacionado a una xAPP de monitoreo dentro del sistema habilitada por la interfaz E2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000			E2AP	1168	E2setupRequest
2	0.030112			E2AP	60	E2setupResponse
3	75.140460			E2AP	82	RICsubscriptionRequest
4	75.172828			E2AP	33	RICsubscriptionResponse
5	85.222918			E2AP	22	RICsubscriptionDeleteRequest
6	85.247526			E2AP	91	RICindication
7	85.254772			E2AP	22	RICsubscriptionDeleteResponse

Figura 114. Paquetes de establecimiento de conexión y suscripción del RIC y xAPP

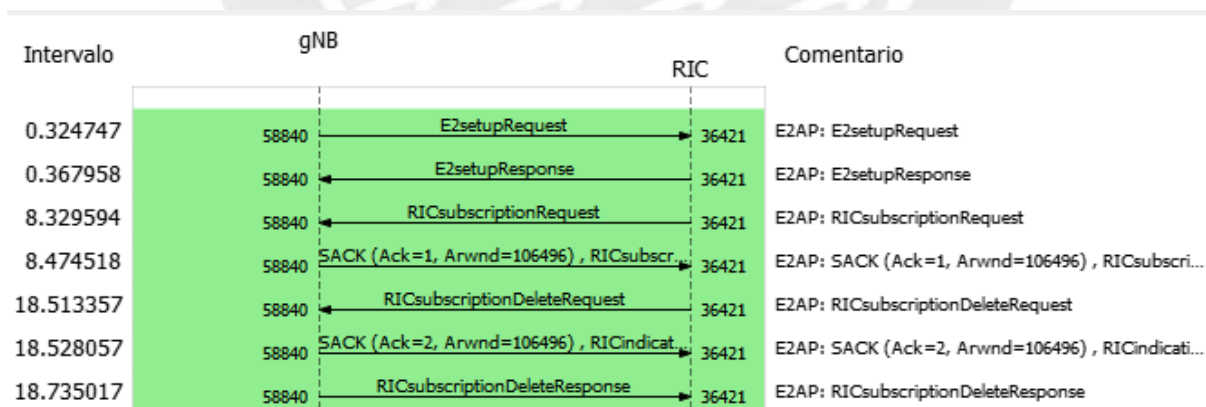


Figura 115. Callflow relacionado al escenario.

En la anterior figura se muestra el proceso de conexión y suscripción de una aplicación al sistema, flujo especificado en la figura 69.

5.4.2 Prueba de funcionamiento

Para este escenario la puesta en marcha es relativamente parecida a escenarios anteriores, en específico al escenario de *attachment and service request* de 5G, por lo tanto, los pasos de levantamiento de *Core* son omitidos en esta sección. Cabe destacar que el *core* se encuentra agrupado y no posee la característica de CUPS vigente en los escenarios anteriores.

En primer lugar, debemos habilitar el FlexRic, debido a que cabe resaltar que es un requerimiento tenerlo activo para que los gNBs se conecten a este.

```
ubuntu@flexric:~/flexric$ sudo ./build/examples/ric/nearRT-RIC -c flexric.conf
[sudo] password for ubuntu:
Setting the config -c file to flexric.conf
Setting path -p for the shared libraries to /usr/local/lib/flexric/
[NEAR-RIC]: nearRT-RIC IP Address = 192.168.0.123, PORT = 36421
[NEAR-RIC]: Initializing
[NEAR-RIC]: Loading SM ID = 144 with def = PDCP_STATS_V0
[NEAR-RIC]: Loading SM ID = 148 with def = GTP_STATS_V0
[NEAR-RIC]: Loading SM ID = 143 with def = RLC_STATS_V0
[NEAR-RIC]: Loading SM ID = 146 with def = TC_STATS_V0
[NEAR-RIC]: Loading SM ID = 142 with def = MAC_STATS_V0
[NEAR-RIC]: Loading SM ID = 147 with def = ORAN-E2SM-KPM
[NEAR-RIC]: Loading SM ID = 145 with def = SLICE_STATS_V0
[iApp]: Initializing ...
[iApp]: nearRT-RIC IP Address = 192.168.0.123, PORT = 36422
fd created with 6
|
```

Figura 116. Comando de inicialización del flexric

Posteriormente, se inicializa el gNB con el siguiente comando donde se evidencia a partir del *output* que se conecta con el flexric.

```
ubuntu@SRSRAN-E2:~/srsRAN_Project/build/apps/gnb$ sudo ./gnb -c gnb.yaml
Lower PHY in executor blocking mode.

--== srsRAN gNB (commit 374200dee) ==--

Connecting to AMF on 192.168.0.111:38412
Available radio types: zmq.
Connecting to NearRT-RIC on 192.168.0.123:36421
Cell pci=1, bw=10 MHz, dl_arfcn=368500 (n3), dl_freq=1842.5 MHz, dl_ssb_arfcn=368410, ul_freq=1747.5 MHz

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

Figura 117. Comando de inicialización del gNB

Finalmente, se ejecutará una xAPP para poder realizar un monitoreo del sistema.

```
ubuntu@flexric:~/flexric$ sudo ./build/examples/xApp/c/monitor/xapp_kpm_moni -c flexric.conf
```

Figura 118. Comando de inicialización del xAPP

```
[xApp]: E42 SETUP-REQUEST sent
adding event fd = 8 ev-> 4
[xApp]: E42 SETUP-RESPONSE received
[xApp]: xApp ID = 7
Registered E2 Nodes = 1
Pending event size before remove = 1
Connected E2 nodes = 1
Registered node 0 ran func id = 147
Generated of req_id = 1
[xApp]: RIC SUBSCRIPTION REQUEST sent
adding event fd = 8 ev-> 5
[xApp]: SUBSCRIPTION RESPONSE received
Pending event size before remove = 1
[xApp]: Successfully SUBSCRIBED to ran function = 147
Remove handle number = 1
E42 RIC_SUBSCRIPTION_DELETE_REQUEST sdr->ric_id.ran_func_id 147 sdr->ric_id.ric_req_id 1
[xApp]: E42 SUBSCRIPTION-DELETE sent
adding event fd = 8 ev-> 7
Received RIC Indication:
---Metric: DRB.RlcSduTransmittedVolumeDL: Value: 0
[xApp]: E42 SUBSCRIPTION DELETE RESPONSE received
Pending event size before remove = 1
[xApp]: Successfully received SUBSCRIPTION-DELETE-RESPONSE
Closing the agent socket: Socket operation on non-socket
[xApp]: Successfully stopped
Test xApp run SUCCESSFULLY
```

Figura 119. Output del xAPP

5.5 Network Slicing

Finalmente, para el último escenario se hará mucho más énfasis en el *core* puesto que en este se realiza la originación de la asignación de los *network slices* por medio de la comunicación de los componentes de *core*. Así, se tendrán los siguientes componentes con sus respectivas funciones en la siguiente tabla a continuación:

TABLA XVIII. LISTADO DE INTERFACES PARA EL ESCENARIO DE NETWORK SLICING

VM	Componentes de Core 5G	Interfaces	Función
5GC (Open5GS)	AMF	127.0.0.5:7777	SBI
		192.168.0.111:38412	NGAP
	AUSF	127.0.0.11:7777	SBI
	UDM	127.0.0.12:7777	SBI
	UDR	127.0.0.15:7777	SBI
	NRF	127.0.0.10:7777	SBI
	PCF	127.0.0.13:7777	SBI
	SMF 1	127.0.0.4:7777	SBI

		192.168.0.112:8805	PFCP
		127.0.0.4:2123	GTPC
		192.168.0.112:2152	GTPU
	SMF 2	127.0.0.24:7777	SBI
		192.168.0.113:8805	PFCP
		127.0.0.24:2123	GTPC
		192.168.0.113:2152	GTPU
Slice 1 UPF (Open5GS)	UPF 1	192.168.0.114:8805	PFCP
		192.168.0.114:2152	GTPU
Slice 2 UPF (Open5GS)	UPF 2	192.168.0.115:8805	PFCP
		192.168.0.115:2152	GTPU

Adicionalmente, dicho procedimiento también se encuentra contemplado en [44]-[46] dentro del proceso de registración en donde se tiene la siguiente secuencia lógica que se aprecia en el siguiente diagrama:

5G Network Slicing Call Flow

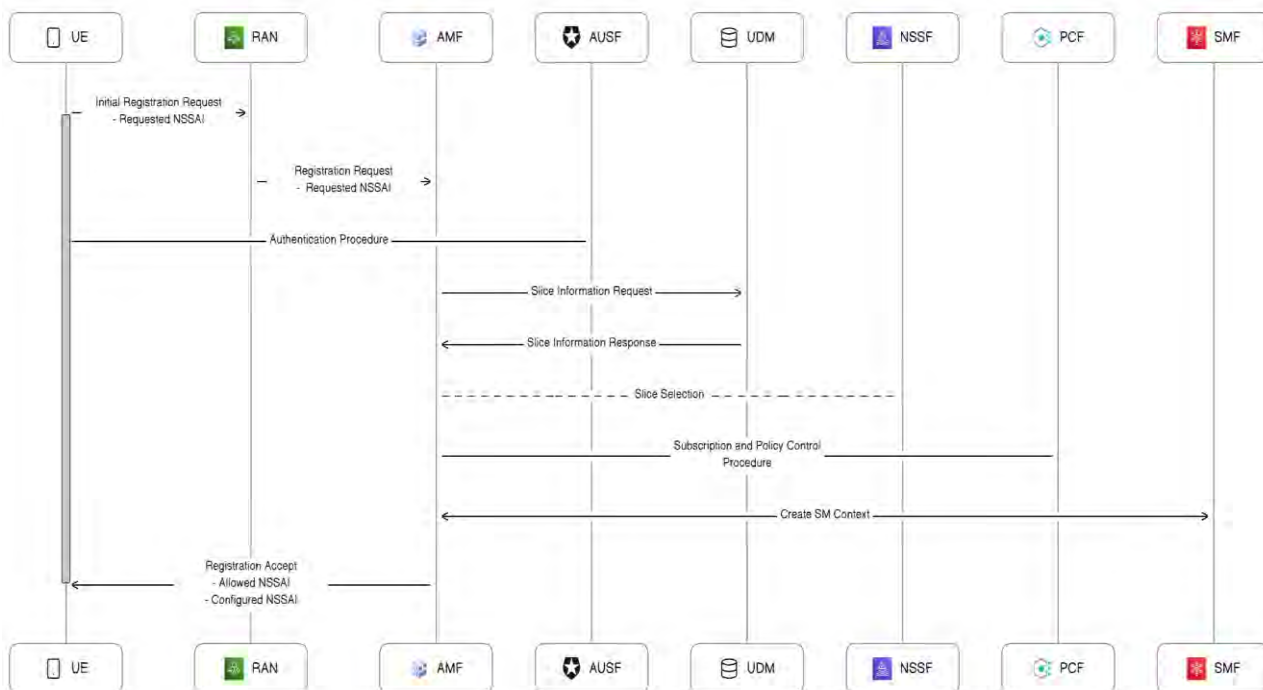


Figura 120. Network Slicing Call Flow

5.5.1 Core

Con dichas consideraciones tomadas en cuenta, se realizaron capturas para poder apreciar la comunicación entre los componentes de *core* apreciados en la tabla XVIII en donde se pueda evidenciar el procedimiento visto en la figura anterior, para esto se usó la herramienta *5G Trace Analyzer* para analizar el contenido de dichas solicitudes hacia el UDM:

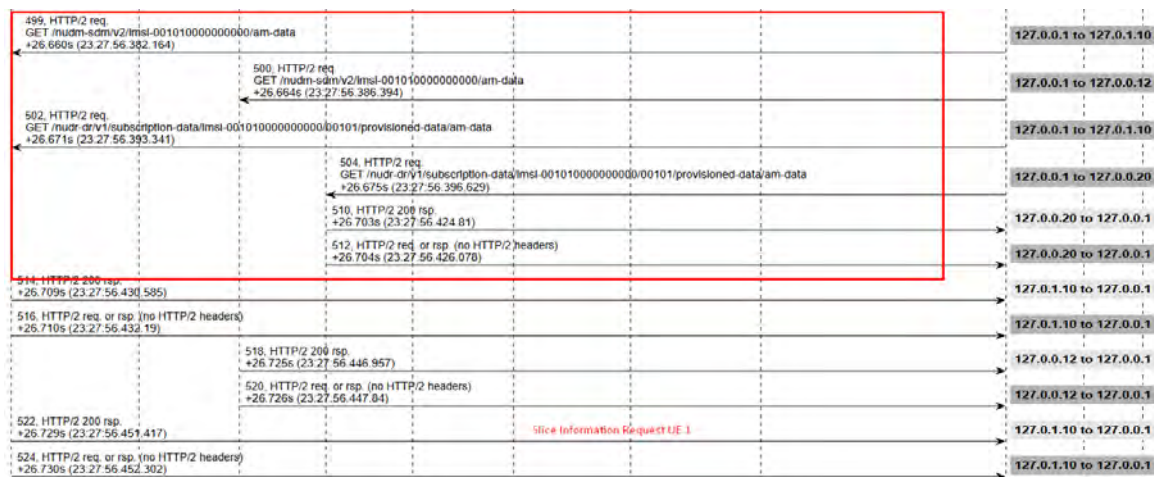


Figura 121. Slice Information Request UE 1

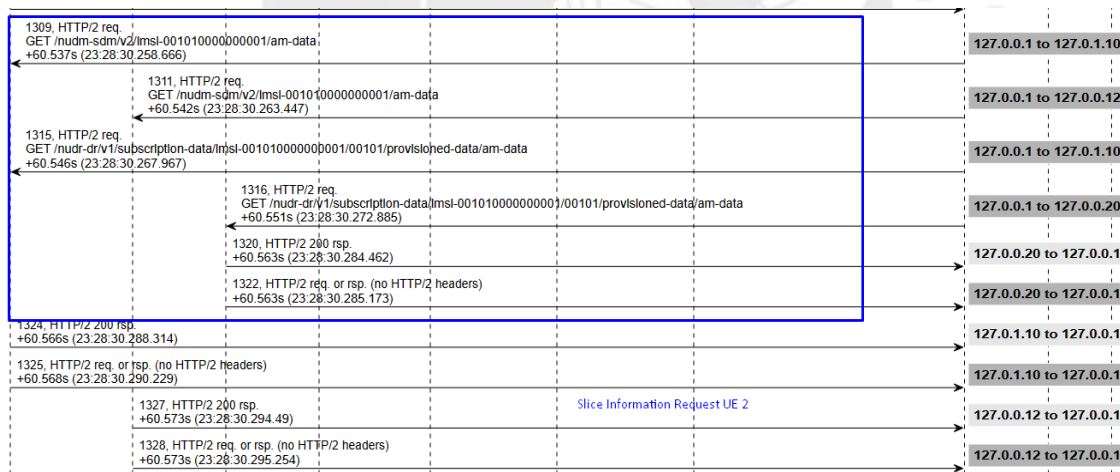


Figura 122. Slice Information Request UE 2

Basado en estas solicitudes se tienen las siguientes respuestas en donde se evidencia que los *network slices* habilitados en la suscripción de cada UE respectivamente, las cuales se puede apreciar en las siguientes figuras:

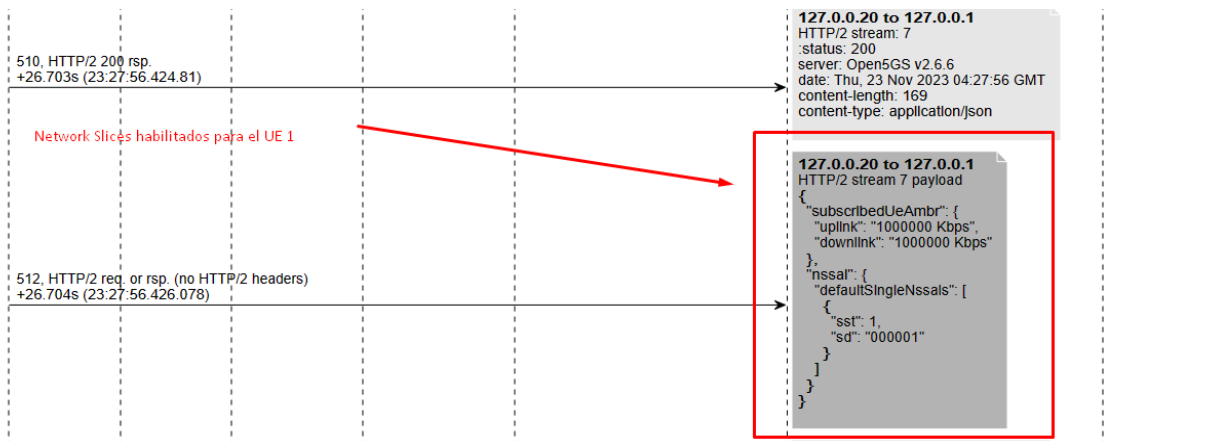


Figura 123. Slice Information Response UE 1

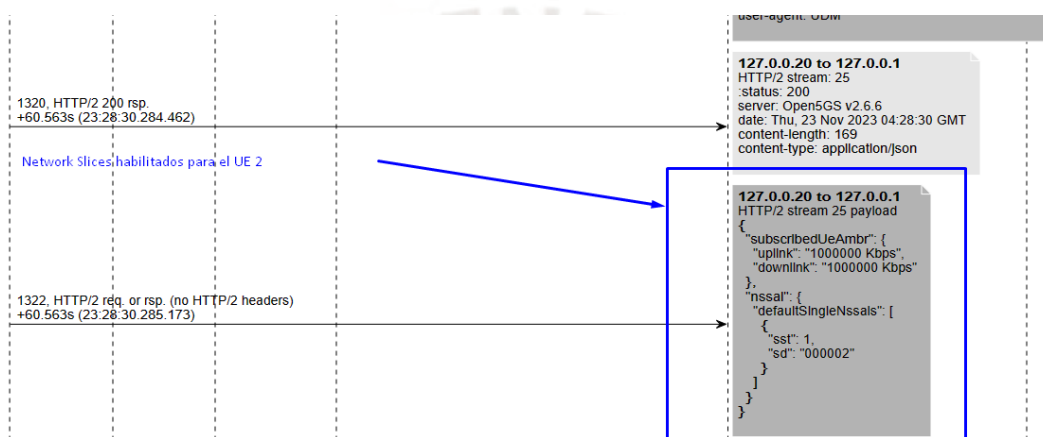


Figura 124. Slice Information Response UE 2

Ahora, para este escenario planteado el AMF no se comunica con el NSSF debido a que este tiene el soporte para los *network slices* habilitados para cada UE; esto se puede evidenciar en el siguiente archivo de configuración correspondiente al AMF.

```

tai:
  - plmn_id:
      mcc: 001
      mnc: 01
      tac: 1
  plmn_support:
    - plmn_id:
        mcc: 001
        mnc: 01
      s_nssai:
        - sst: 1
          sd: 000001
        - sst: 1
          sd: 000002
  security:
    integrity_order : [ NIA2, NIA1, NIA0 ]
    ciphering_order : [ NEA0, NEA1, NEA2 ]
  network_name:
    full: Open5GS
  amf_name: open5gs-amf0
    
```

Network slices soportados por el AMF

Figura 125. Network slices soportados por el AMF

Sin embargo, para tener una idea sobre en qué escenario sería posible la comunicación entre el AMF y el NSSF se tiene el siguiente gráfico:

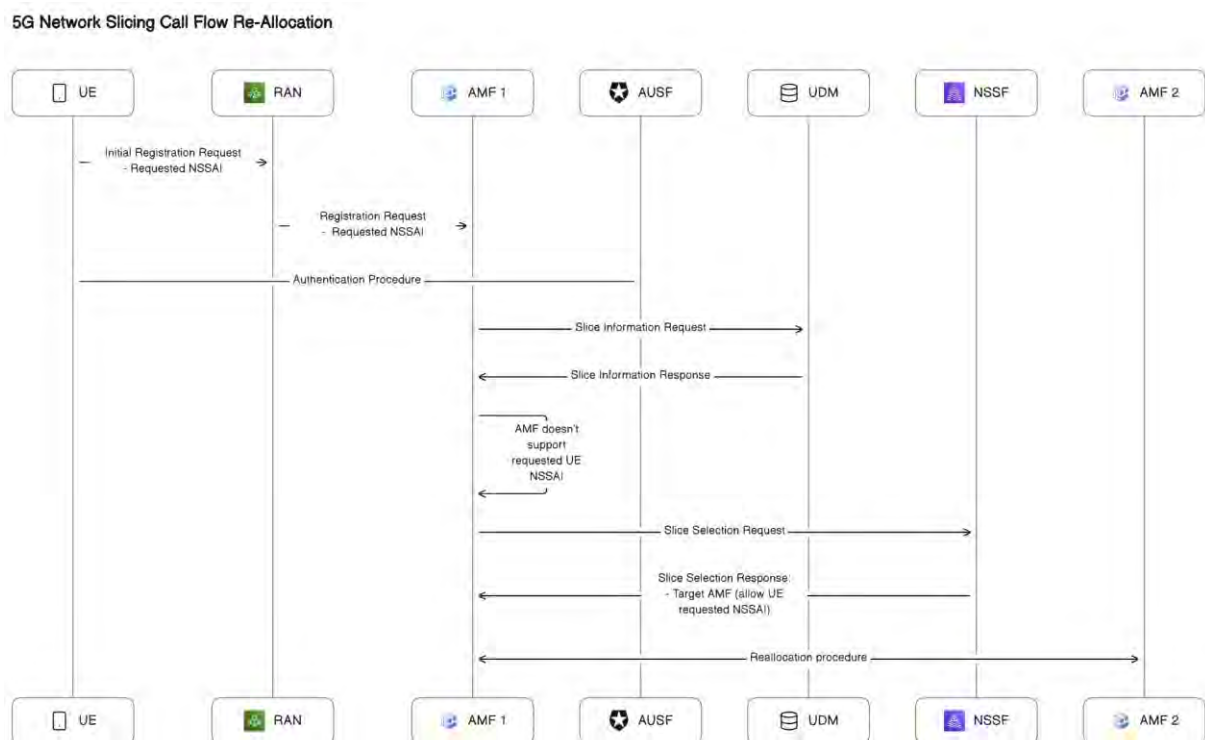


Figura 126. Network Slicing Call Flow con recolocación de AMF

Con esto en consideración, el siguiente paso a verificar es la asignación del SMF que soporta el NSSAI solicitado por el UE.

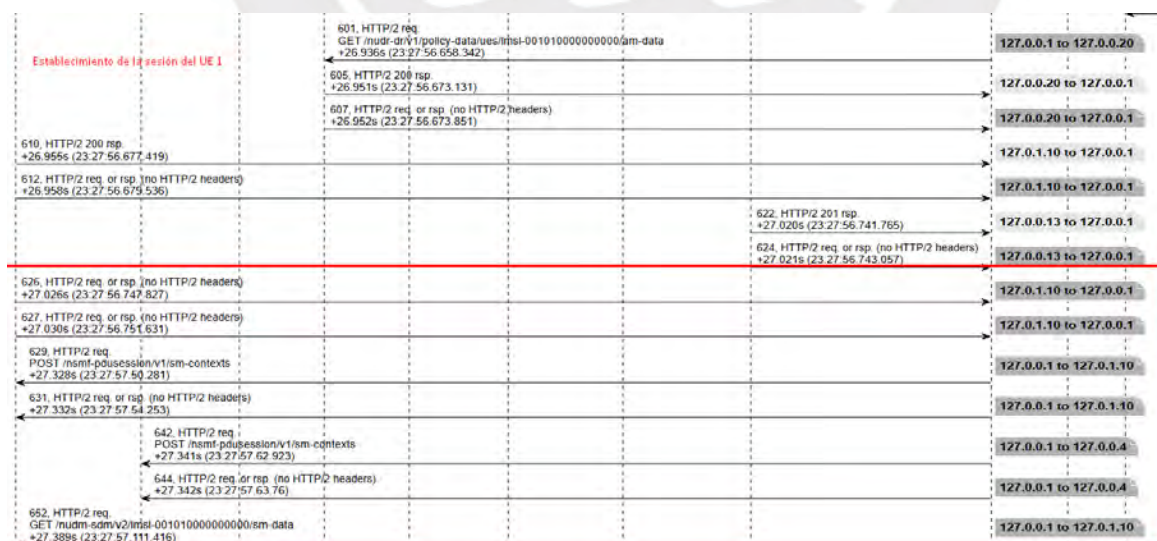


Figura 127. Asignación de SM context del UE 1



Figura 128. Asignación de SM context del UE 2

Dentro de estas solicitudes se tienen los siguientes parámetros para el establecimiento del SM Context en el SMF respectivo.

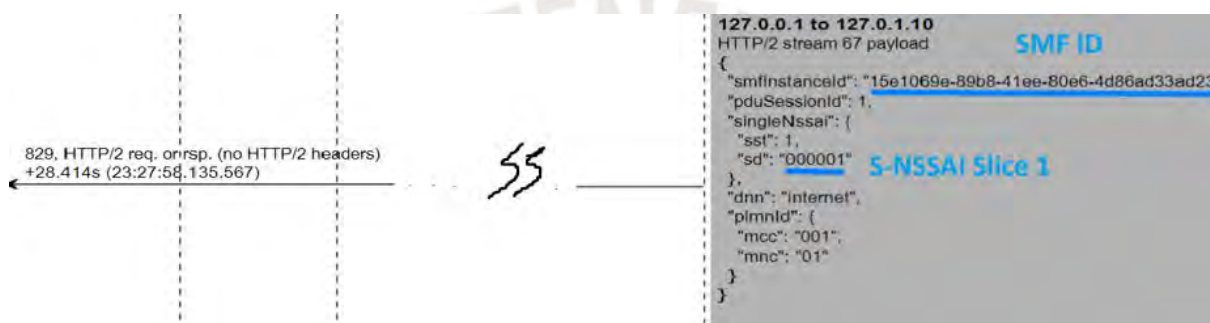


Figura 129. Parámetros para establecimiento de sesión en el UE 1

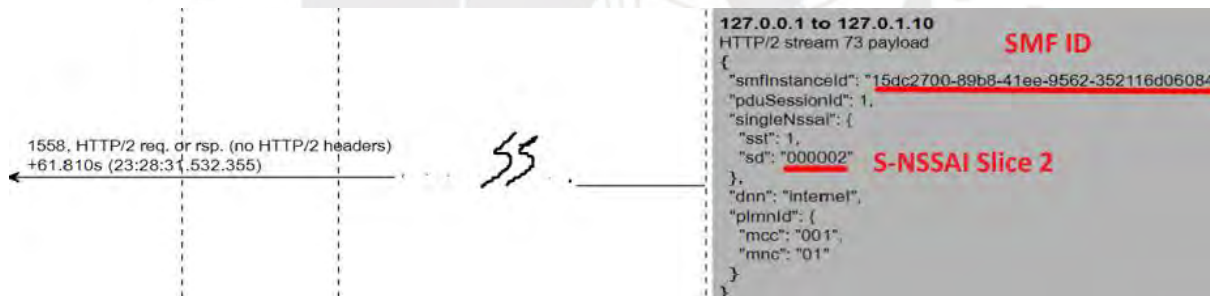


Figura 130. Parámetros para establecimiento de sesión en el UE 2

Por otro lado, se identificarán los *callflows* relacionados a este proceso desde la interfaz expuesta, conexión entre componentes externos con los componentes del *core*. En la siguiente figura se muestra como cada SMF relacionado a un *slice differentiation* establece una sesión de datos con su UPF respectivo, considerando que para el experimento se utilizó dos UE con diferentes *slices* especificados.

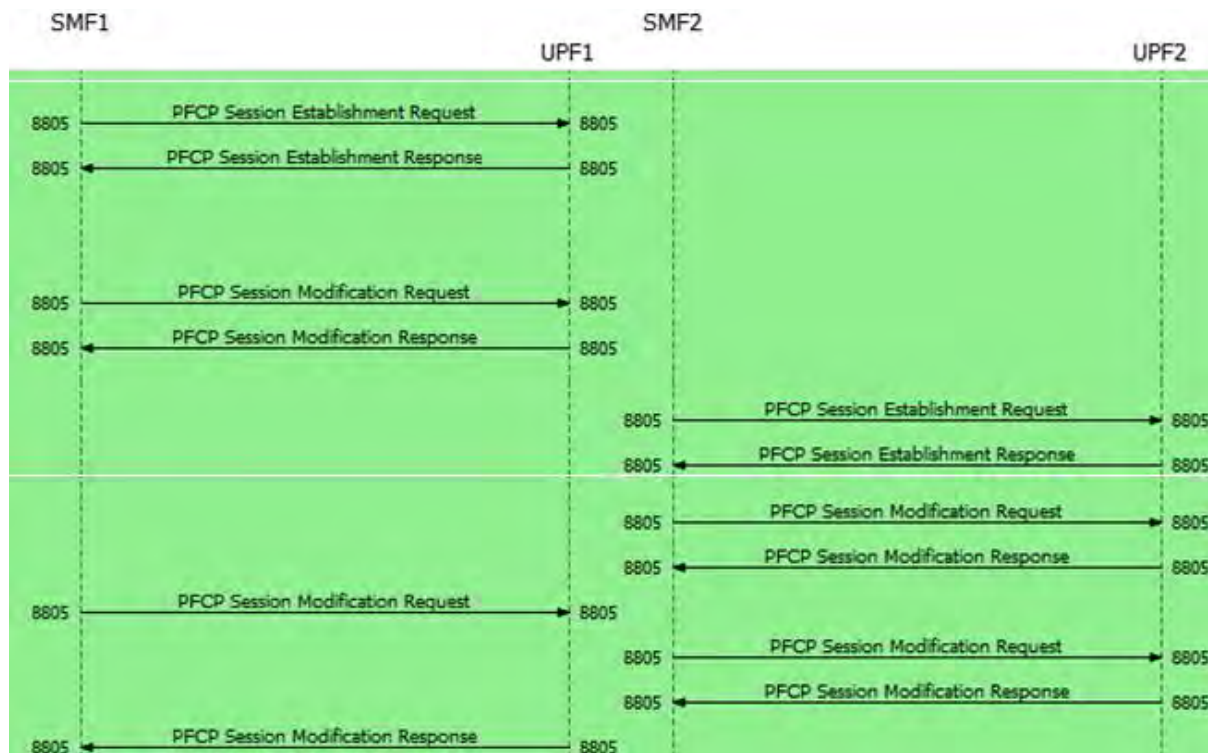


Figura 131. Callflow de session establishment para network slicing

Para concluir el análisis de establecimiento de sesión diferenciado se verifica el *payload* relacionado con las sesiones mostradas previamente en el *callflow*. A continuación, se analizará el *payload* relacionado con el SMF1, relacionado para el sd: 010000.

55 21.229998 SMF1 UPF1 PFCP 667 PFCP Session Establishment Request

Figura 132. Paquete de session establishment entre el SMF1 y el UPF1

Se observa cómo se establece el DN relacionado al UE y el *slice* correspondiente.

```

> PDN Type : IPv4
> User ID :
v APN/DNN : internet
  IE Type: APN/DNN (159)
  IE Length: 9
  APN/DNN: internet
v S-NSSAI :
  IE Type: S-NSSAI (257)
  IE Length: 4
  SST: 01
  SD: 010000

```

[\[Response In: 56\]](#)

Figura 133. Payload relacionado al session establishment para el SMF1

Análogamente para el caso del SMF2 donde observamos cómo identificar la sesión con el *slice differentiation* y su respectivo DN

```
119 54.790918 SMF2          UPF2          PFCP          667 PFCP Session Establishment Request
```

Figura 134. Paquete de *session establishment* entre el SMF2 y el UPF2

```

  v APN/DNN : internet
    IE Type: APN/DNN (159)
    IE Length: 9
    APN/DNN: internet
  v S-NSSAI :
    IE Type: S-NSSAI (257)
    IE Length: 4
    SST: 01
    SD: 020000

```

[\[Response In: 120\]](#)

Figura 135. Payload relacionado al *session establishment* para el SMF2

5.5.2 Prueba de funcionamiento

En esta sección se establecerán las pruebas de funcionamiento de este escenario, cabe destacar que los procesos a nivel de *core* y UPF, puesta en marcha y verificación de estos, se dejan de lado ya que son similares a los escenarios anteriormente descritos. Adicionalmente, se hará hincapié en los parámetros de configuración de los UE y gNB.

```

GNU nano 4.8          ../config/gnb.yaml
mcc: '001'           # Mobile Country Code value
mnc: '01'            # Mobile Network Code value (2 or 3 digits)

nci: '0x000000010'  # NR Cell Identity (36-bit)
idlength: 32       # NR gNB ID length in bits [22..32]
tac: 1              # Tracking Area Code

linkIp: 192.168.0.131 # gNB's local IP address for Radio Link Simulation (Usually same with local IP)
ngapIp: 192.168.0.131 # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 192.168.0.131 # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
- address: 192.168.0.111
  port: 38412
# List of supported S-NSSAIs by this gNB
slices:
- sst: 1
  sd: 0x000001
- sst: 1
  sd: 0x000002
# Indicates whether or not SCTP stream number errors should be ignored.
ignoreStreamIds: true

```

Figura 136. Parámetros de configuración del gNB

```

ubuntu@UERANSIM-gNB:~/UERANSIM/build$ sudo ./nr-gnb -c ../config/gnb.yaml
UERANSIM v3.2.6
[2023-11-24 04:56:24.289] [sctp] [info] Trying to establish SCTP connection... (192.168.0.111:38412)
[2023-11-24 04:56:24.730] [sctp] [info] SCTP connection established (192.168.0.111:38412)
[2023-11-24 04:56:24.744] [sctp] [debug] SCTP association setup ascId[7]
[2023-11-24 04:56:24.748] [ngap] [debug] Sending NG Setup Request
[2023-11-24 04:56:24.819] [ngap] [debug] NG Setup Response received
[2023-11-24 04:56:24.829] [ngap] [info] NG Setup procedure is successful

```

Figura 137. Código de ejecución del gNB

```

# Initial PDU sessions to be established
sessions:
- type: 'IPv4'
  apn: 'internet'
  slice:
    sst: 1
    sd: 0x000002
# Configured NSSAI for this UE by HPLMN
configured-nssai:
- sst: 1
  sd: 0x000002
# Default Configured NSSAI for this UE
default-nssai:
- sst: 1
  sd: 0x000002
# Supported integrity algorithms by this UE
integrity:
  IA1: true
  IA2: true
  IA3: true

```

Figura 138. Parámetros de configuración del UE

```

ubuntu@UERANSIM-UE:~/UERANSIM/build$ sudo ./nr-ue -c ../config/ue2.yaml
[sudo] password for ubuntu:
UERANSIM v3.2.6
[2023-11-24 04:57:29.026] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2023-11-24 04:57:29.088] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2023-11-24 04:57:29.108] [nas] [info] Selected plmn[001/01]
[2023-11-24 04:57:29.119] [rrc] [info] Selected cell plmn[001/01] tac[1] category[SUITABLE]
[2023-11-24 04:57:29.123] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2023-11-24 04:57:29.125] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2023-11-24 04:57:29.126] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2023-11-24 04:57:29.139] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-11-24 04:57:29.142] [nas] [debug] Sending Initial Registration
[2023-11-24 04:57:29.209] [rrc] [debug] Sending RRC Setup Request
[2023-11-24 04:57:29.212] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2023-11-24 04:57:29.258] [rrc] [info] RRC connection established
[2023-11-24 04:57:29.259] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2023-11-24 04:57:29.261] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-11-24 04:57:29.707] [nas] [debug] Authentication Request received
[2023-11-24 04:57:29.917] [nas] [debug] Security Mode Command received
[2023-11-24 04:57:29.926] [nas] [debug] Selected integrity[2] ciphering[0]
[2023-11-24 04:57:30.478] [nas] [debug] Registration accept received
[2023-11-24 04:57:30.486] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2023-11-24 04:57:30.490] [nas] [debug] Sending Registration Complete
[2023-11-24 04:57:30.493] [nas] [info] Initial Registration is successful
[2023-11-24 04:57:30.498] [nas] [debug] Sending PDU Session Establishment Request
[2023-11-24 04:57:30.527] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-11-24 04:57:30.690] [nas] [debug] Configuration Update Command received
[2023-11-24 04:57:31.731] [nas] [debug] PDU Session Establishment Accept received
[2023-11-24 04:57:31.738] [nas] [info] PDU Session establishment is successful PSI[1]
[2023-11-24 04:57:32.794] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.46.0.2] is up.

```

Figura 139. Código de ejecución del UE con el slice differentiation 000002

5.6 Resultados de aprendizaje

En esta sección se discutirán los resultados de aprendizaje obtenidos al implementar un programa piloto en el uso de laboratorios virtualizados para el curso de Ingeniería de Redes Móviles (TEL 341). Se tomó como referencia los resultados provistos del *feedback* de los estudiantes y se tuvo lo siguiente:



Figura 140. Percepción de los estudiantes respecto al laboratorio

Asimismo, los estudiantes obtuvieron el siguiente rendimiento académico en la actividad de laboratorio:

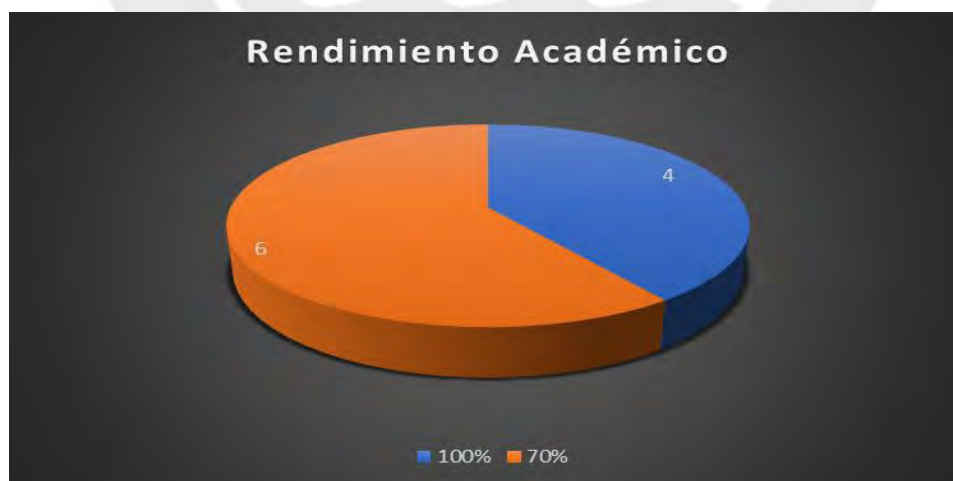


Figura 141. Rendimiento académico de los estudiantes

Asimismo, los estudiantes lograron terminar de forma satisfactoria la experiencia de laboratorio; sin embargo, tuvieron los siguientes inconvenientes en el desarrollo de este:



Figura 142. Dificultades encontradas por los estudiantes

Finalmente, los estudiantes consideraron otros casos de aplicación de entornos virtualizados tales como:



Figura 143. Sugerencias brindadas por los estudiantes como aplicación

CONCLUSIONES

- La evidencia presentada en el capítulo 5 demuestra que la implementación cumple con los requerimientos a nivel técnico según las normas propuestas por la 3GPP y la O-RAN Alliance de forma que permita el correcto análisis y estudio de los procedimientos base de redes 4G como 5G en los escenarios seleccionados.
- Según lo establecido en el capítulo 3 y 4, se logró el correcto acondicionamiento e integración de los componentes involucrados en la generación de escenarios de forma que los estudiantes tengan las herramientas y entornos necesarios para que permitan realizar una trazabilidad y control sobre los componentes tanto de *core* como de RAN.
- Asimismo, el uso de un VIM como OpenStack facilita el uso de infraestructura *baremetal* y la vuelve flexible gracias a la integración de un microorquestador que permita, por medio de APIs, el despliegue automatizado de los escenarios descritos. Esta solución puede integrarse con orquestadores como es el caso de HAST/VNRT para ser brindada a estudiantes de la universidad con fines de investigación y académicos.
- Por las razones mencionadas previamente se concluye que la solución reduce la necesidad de adquisición de dispositivos COTS, los cuales requieren una gran inversión y, a su vez, limitan la cantidad de escenarios de estudio posibles.
- En síntesis, la solución presenta cualidades que permitan contribuir en la formación académica y en investigación de los estudiantes de la Pontificia Universidad Católica del Perú en cuanto a tecnologías 4G, 5G y en tecnologías emergentes como es el caso de Open RAN que permitan producir profesionales capacitados necesarios para cerrar la brecha de infraestructura 4G que se tiene en el país y con ello agilizar el despliegue y manejo de 5G.

RECOMENDACIONES Y OBSERVACIONES

- En caso la tesis sea aplicada en la universidad, se recomienda que esta trabaje en conjunto con el orquestador HAST/VNRT de forma que permita la gestión de escenarios, haciendo uso de la base de datos de este último posee, por alumno y que, en conjunto, brinden la capacidad de generar escenarios altamente personalizados.
- En segundo lugar, con fines de que estas herramientas puedan trabajar en conjunto con unidades de radio y terminales comerciales, se sugiere que pueda haber negociaciones entre representantes de la universidad y los operadores, los cuales son dueños del espectro licitado por el MTC, de forma que, sólo para fines académicos, pueda haber una compartición de las bandas de frecuencia de manera que no se infrinja en una multa y/o sanción por parte del estado hacia la universidad.
- En tercer lugar, con referencia al punto anterior, se recomienda que se realice una adquisición de dispositivos de radio de forma que se pueda conectar a los componentes de RAN y *core* virtualizados alojados en la nube privada de la universidad. Asimismo, la adquisición de tarjetas SIM especiales para el desarrollo de forma que estas permitan a los terminales conectarse a la red móvil gracias a parámetros de suscripción configurados dentro de estas.
- Finalmente, se recomienda que se continúe con la línea de trabajo base dejada en esta tesis de manera que se puedan, con estas herramientas, lograr la implementación de escenarios en los que se puedan apreciar servicios como son VoLTE y/o VoNR, los cuales impliquen la integración con un IMS y de equipos físicos; o, por el contrario, la implementación de un *smart campus* haciendo uso de la tecnología celular 5G virtualizada y Open RAN que permita a la especialidad tener un proyecto multidisciplinario que permita involucrar a todas las ramas de la carrera.

BIBLIOGRAFÍA

- [1] A. Vizcarra and R. Pastor, “Análisis de herramientas Open Source para la implementación de entornos educativos de redes 4G y 5G” Aug, 2023. [Online]. Available: <https://wiki.pucp.edu.pe/GIRA/AnalisisHerramientasOpenSource4G5G>
- [2] O. de León, “Redes 5G en América Latina: desarrollo y potencialidades” Nov. 2022, Accessed: Jun. 08, 2023. [Online]. Available: <https://repositorio.cepal.org/handle/11362/48485>
- [3] Gsma, “América Latina 2022 La Economía Móvil en Contenidos,” 2022, Accessed: Jun. 08, 2023. [Online]. Available: www.gsmaintelligence.com
- [4] J. More Sánchez and M. Gavilano Aspillaga, “Estimación del número de estaciones base celular para atender la demanda de servicios móviles en el Perú al año 2025, Documento de Trabajo N° 50,” *Repositorio Institucional OSIPTEL*, 2020, Accessed: Jun. 09, 2023. [Online]. Available: <https://repositorio.osiptel.gob.pe/xmlui/handle/20.500.12630/746>
- [5] B. Pablo, M. Silva, A. Luis, and A. P. Zevallos, “Impactos & recomendaciones para el despliegue de las redes 5G en el mercado peruano,” Apr. 2021, Accessed: Jun. 09, 2023. [Online]. Available: <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/18883>
- [6] “Asignación de espectro para 5G: un proceso impostergable | eBIZ Noticias.” <https://noticias.ebiz.pe/asignacion-de-espectro-para-5g-un-proceso-impostergable/> (accessed Aug. 10, 2023).
- [7] P. Valente, D. Raposo, P. Rito and S. Sargento, "Disaggregated Mobile Core for Edge City Services," 2023 IEEE 24th International Symposium on a World of Wireless,

Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 2023, pp. 157-166, doi: 10.1109/WoWMoM57956.2023.00030.

[8] “Enterprise 5G – Disaggregating the core.” <https://blogs.oracle.com/oracle-communications/post/enterprise-5g-disaggregating-the-core> (accessed Sep. 8, 2023).

[9] S. Sirotkin, *5G radio access network architecture: The dark side of 5G*. 2020. doi: 10.1002/9781119550921.

[10] O-RAN Alliance, “O-RAN Fronthaul Working Group Control, User and Synchronization Plane Specification,” 2019.

[11] Parallel Wireless, “Everything you need to know about Open RAN,” 2020, Accessed: Jun. 13, 2023. [Online]. Available: <https://www.parallelwireless.com/wp-content/uploads/Parallel-Wireless-e-Book-Everything-You-Need-to-Know-about-Open-RAN.pdf>

[12] O. T. T. and V. Deutsche Telekom, “Open RAN Technical Priorities Executive Summary,” 2021, Accessed: Jun. 12, 2023. [Online]. Available: <https://www.telefonica.com/en/wp-c>

[13] K. Du, L. Wang, Z. Zhu, Y. Yan and X. Wen, "Converged Service-based Architecture for Next-Generation Mobile Communication Networks," 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, United Kingdom, 2023, pp. 1-6, doi: 10.1109/WCNC55385.2023.10118793.

[14] Min Wang, “The GSMA’s Future Networks: 5G Implementation Guidelines,” 2020. Accessed: Sep. 11, 2023. [Online]. Available: <https://www.gsma.com/futurenetworks/wp-content/uploads/2020/02/5G-SA-Option-2-ImplementationGuideline-v1.2.pdf>

[15] ETSI, “GS NFV-MAN 001 - V1.1.1 - Network Functions Virtualization (NFV); Management and Orchestration,” Valbonne, France, Dec. 2014. Accessed: Aug. 18,

2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[16] RedHat, “VNF and CNF, what’s the difference?,” What is network virtualization?, Jul. 28,2022.Available:<https://www.redhat.com/en/topics/cloud-native-apps/vnf-and-cnf-whats-the-difference?pfe-n7107zk3f=resources&pfe-5khp6ux79=resources&pfe-skjcin17h=related-articles>

[17] Samsung Electronics Co., “Cloud Native 5G Core Samsung 5G Core Vol. 2,” Korea, 2020. Accessed: Sep. 10, 2023. [Online]. Available: <https://images.samsung.com/is/content/samsung/p5/global/business/networks/insights/white-paper/cloud-native-5g-core/Samsung-5G-Core-Vol-2-Cloud-Native-5G-Core.pdf>

[18] Ericsson, “The cloud-native transformation,” Sweden, 2022. Accessed: Sep. 10, 2023. [Online]. Available: <https://www.ericsson.com/en/cloud-native>

[19] “Features | Open5GS.” <https://open5gs.org/open5gs/features/> (accessed Aug. 29, 2023).

[20] “Overview – OpenAirInterface.” <https://openairinterface.org/overview/> (accessed Aug. 14, 2023).

[21] “Magma – Linux Foundation Project.” <https://magmacore.org/> (accessed Aug. 14, 2023).

[22] “Introduction — srsRAN 4G 23.04 documentation.” https://docs.srsran.com/projects/4g/en/latest/usermanuals/source/srsepc/source/1_epc_intro.html#features (accessed Aug. 29, 2023).

[23] “srsRAN Project Documentation — srsRAN Project documentation.” <https://docs.srsran.com/projects/project/en/latest/index.html> (accessed Aug. 29, 2023).

- [24] “doc/FEATURE_SET.md · develop · oai / openairinterface5G · GitLab.”
https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/FEATURE_SET.md (accessed Aug. 29, 2023).
- [25] “UERANSIM doc.” <https://github.com/aligungr/UERANSIM/wiki> (accessed Jun. 15, 2023).
- [26] A. Lacava *et al.*, “Programmable and Customized Intelligence for Traffic Steering in 5G Networks Using Open RAN Architectures,” Sep. 2022, doi: 10.1109/TMC.2023.3266642.
- [27] ETSI OSM Community, “OSM RELEASE THREE A TECHNICAL OVERVIEW,” 2017. [Online]. Available: www.etsi.org
- [28] ONAP Community, “Open Network Automation Platform (ONAP) Architecture Overview White Paper,” 2018. Accessed: Sep. 13, 2023. [Online]. Available: https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf
- [29] D. K. Rensin, *Kubernetes - Scheduling the Future at Cloud Scale*. 2015. [Online]. Available: https://cloud.redhat.com/hubfs/pdfs/Kubernetes_OpenShift.pdf
- [30] *OpenStack Docs: 2023.2*. (n.d.). Retrieved October 22, 2023, from <https://docs.openstack.org/2023.2/>
- [31] G. M. Yilma, Z. F. Yousaf, V. Sciancalepore, and X. Costa-Perez, “Benchmarking open source NFV MANO systems: OSM and ONAP,” *Comput Commun*, vol. 161, 2020, doi: 10.1016/j.comcom.2020.07.013.
- [32] A. Pino Martínez, M. S. Siddiqui, A. Xavier, and H. Serra, “Validation and Extension of Kubernetes-based Network Functions (KNFs) in OSM for Cloud Native (CN) applications in 5G and beyond” Barcelona, Jan. 2021.

- [33] “What is Rancher? | Kubernetes Management Simplified | Rancher.” Accessed: Oct. 24, 2023. [Online]. Available: <https://www.rancher.com/why-rancher>
- [34] “Helm | Charts.” Accessed: Oct. 25, 2023. [Online]. Available: <https://helm.sh/docs/topics/charts/>
- [35] “Universal TUN/TAP device driver — The Linux Kernel documentation.” Accessed: Oct. 25, 2023. [Online]. Available: <https://docs.kernel.org/networking/tuntap.html>
- [36] “Operating System Containers vs. Application Containers - RisingStack Engineering.” Accessed: Oct. 25, 2023. [Online]. Available: <https://blog.risingstack.com/operating-system-containers-vs-application-containers/>
- [37] Shigeru Ishida, “Open5GS 5GC & UERANSIM UE / RAN Sample Configuration - Select UPF based on S-NSSAI.” Accessed: Nov. 11, 2023. [Online]. Available: https://github.com/s5uishida/open5gs_5gc_ueransim_snsai_upf_sample_config
- [38] Sysmocom, “SysmoUSIM - User Manual.” Accessed: Oct. 26, 2023. [Online]. Available: <https://sysmocom.de/manuals/sysmousim-manual.pdf>
- [39] Amarisoft, “AMARI Callbox Classic.” Accessed: Oct. 25, 2023. [Online]. Available: <https://www.amarisoft.com/app/uploads/2021/10/AMARI-Callbox-Classic.pdf>
- [40] Amarisoft, “AMARI LTE Callbox - mini.” Accessed: Oct. 25, 2023. [Online]. Available: <https://www.amarisoft.com/app/uploads/2021/10/AMARI-Callbox-Mini.pdf>
- [41] Amarisoft, “AMARI UE Simbox Serie E,” 2022. Accessed: Oct. 25, 2023. [Online]. Available: <https://www.amarisoft.com/app/uploads/2022/03/AMARI-UE-Simbox-E-Series.pdf>
- [42] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," TS 23.401 version 15.12.0

- Release 15, Jun. 2022. [Online]. Available:
https://www.3gpp.org/ftp/Specs/archive/23_series/23.401/23401-f20.zip
- [43] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)", TS 36.413 version 15.19.0 Release 15, Jan. 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.413/36413-f90.zip.
- [44] 3GPP, "Non-Access-Stratum (NAS) protocol for 5G System (5GS)," TS 24.501 version 15.3.0, Sep. 2018. [Online]. Available:
https://www.3gpp.org/ftp/Specs/archive/24_series/24.501/24501-f30.zip.
- [45] 3GPP, "Procedures for the 5G System," TS 23.502 version 15.3.0, Sep. 2018. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.502/23502-f30.zip.
- [46] 3GPP, "NG-RAN; NG Application Protocol (NGAP)," TS 38.413 version 15.4.0, Sep. 2019. [Online]. Available:
https://www.3gpp.org/ftp/Specs/archive/38_series/38.413/38413-f40.zip.
- [47] O-RAN Alliance, "Near-Real-time RAN Intelligent Controller and E2 Interface: E2 Application Protocol (E2AP)" version 04.00, Oct.2023 [Online]. Available:
<https://orandownloadswb.azurewebsites.net/specifications>
- [48] O-RAN Alliance, "Near-Real-time RAN Intelligent Controller and E2 Interface: E2 Service Model" version 04.00, Oct.2023 [Online]. Available:
<https://orandownloadswb.azurewebsites.net/specifications>
- [49] O-RAN Alliance, "Near-Real-time RAN Intelligent Controller and E2 Interface: E2 Service Model RAN Control" version 04.00, Oct.2023 [Online]. Available:
<https://orandownloadswb.azurewebsites.net/specifications>

ANEXOS

Repositorio de Docker Hub	https://hub.docker.com/repository/docker/agustinvl/gira/general
Repositorio de configuraciones	https://github.com/AgustinVizcarra/Gira_4G_5G_Tools
Repositorio del servicio REST API + Openstack	https://github.com/AgustinVizcarra/MobileSystemTopologyGenerator
Capturas Escenario 1	https://drive.google.com/drive/folders/1WikINKqvZ9e5iTWevqonr5nny3KzyWG3?usp=drive_link
Capturas Escenario 2	https://drive.google.com/drive/folders/1HI4EV5Cf2_NE6en58k6of4WTvUqh8kPJ?usp=sharing
Capturas Escenario 3	https://drive.google.com/drive/folders/1gA-AqHMPdh_DGaRra6h5vEfXZ3x5xEIz?usp=sharing
Capturas Escenario 4	https://drive.google.com/drive/folders/1wAmlhYKiVgCGRQebh6oo7BcvIrg9t3kv?usp=sharing
Capturas Escenario 5	https://drive.google.com/drive/folders/1jf9wByT2mP-Kdr7mqrT--QvaWnUev_IG?usp=sharing