



**UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMA Y TELECOMUNICACIONES**

TRABAJO DE INTEGRACION CURRICULAR

RAN VIRTUALIZADA: EVALUACION DE ALGORITMO PARA NECESIDADES
DE ORQUESTACION ENTRE vBBUs.

AUTOR

REYES DE LA CRUZ CHRISTIAN RONNY

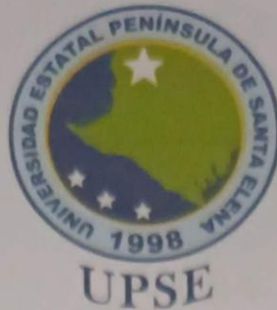
PREVIO A LA OBTENCIÓN DEL GRADO ACADÉMICO EN
INGENIERO EN TELECOMUNICACIONES

TUTOR

Ing. JARAMILLO CHAMBA DANIEL ARMANDO, MSc.

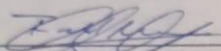
Santa Elena, Ecuador

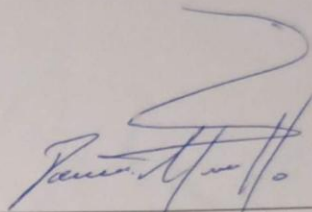
Año 2025

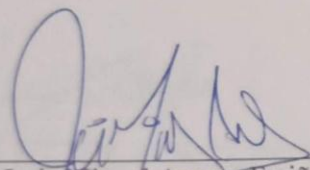


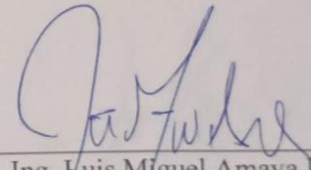
**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

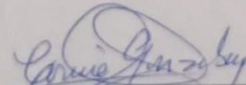
TRIBUNAL DE SUSTENTACIÓN

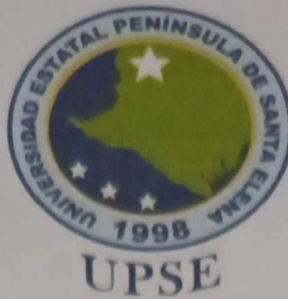

Ing. Ronald Rovira Jurado, PhD
DIRECTOR DE LA CARRERA


Ing. Jaramillo Chamba Daniel Armando,
MSc.
TUTOR


Ing. Luis Miguel Amaya Fariño, Mgt
DOCENTE ESPECIALISTA


Ing. Luis Miguel Amaya Fariño, Mgt
DOCENTE GUÍA UIC


Ing. Corina Genzabay De La A, Mgt
SECRETARIA



UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES.

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **REYES DE LA CRUZ CHRISTIAN RONNY**, como requerimiento para la obtención del título de Ingeniero en Telecomunicaciones.

TUTOR

Ing. Daniel Armando Jaramillo Chamba, MSc.



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

DECLARACIÓN DE RESPONSABILIDAD

Yo, **REYES DE LA CRUZ CHISTIAN RONNY**

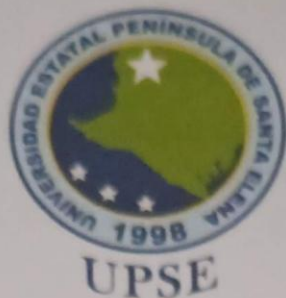
DECLARO QUE:

El trabajo de Titulación, RAN VIRTUALIZADA: EVALUCION DE ALGORITMO PARA NECESIDADES DE ORQUESTACION ENTRE vBBUs previo a la obtención del título en Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

EL AUTOR

REYES DE LA CRUZ CHRISTIAN RONNY



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

AUTORIZACIÓN

Yo, REYES DE LA CRUZ CHRISTIAN RONNY

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

EL AUTOR

CHRISTIAN RONNY REYES DE LA CRUZ

AGRADECIMIENTO

Primero dar gracias a dios por permitirme dar un paso más hacia el éxito y darme las fuerzas para seguir adelante en mis estudios en no rendirme, también agradecer a nuestra madre maría que nos protege y nos cubre con su santo manto.

Le debo la vida a mis abuelitas, ellas han sido un pilar fundamental en mi vida, cada viaje a la universidad me daba y aun me siguen dando su bendición y también me daban para el pasaje que recuerdos tan bonitos. De igual manera agradezco a mi papá por estar siempre apoyándome desde el día 1 en que nací, cuando sacrificio ha hecho para que uno de sus hijos se esfuerce y sea un profesional como él. Su frase que me llevare para tu mi vida es: CREE EN TI, TU PUEDES.

Así mismo agradezco a mis hermanos que han sido una pieza clave en mis estudios, siempre dándome ánimos y ese aliento de si se puede. Me han apoyado y me han seguido apoyando económicamente, gracias se los agradezco un montón.

Gracias a una persona tan especial que ha marcado mi vida y ha hecho de mi una persona fuerte, su nombre es Melanie Johanna Montenegro Malavé, siempre dándome consejos para seguir adelante y diciéndome no se rinda amor ya falta poco, hágalo por su abuelita, por su papa, por mis hermanos, solo me queda decirle gracias, amor, y gracias a su familia por ayudarme en cualquier cosa que necesito, les quedo muy agradecido por acompañarme en este proceso.

Christian Reyes

DEDICATORIA

Esta dedicatoria tiene muchos nombres, pero primero que todo dar gracias a dios por permitirnos dar un día mas de vida.

Abuelita este esfuerzo mío es tuyo, sin ti no sé qué haría, que recuerdos tengo cuando te levantabas para poder darnos un desayuno y almuerzo antes de irme a la universidad, han pasado muchas cosas, pero solo dios sabe por que las hace. Estoy muy contento de que yo sea tu nieto, y si también puedo decir con orgullo que soy tu hijo por que me criaste desde pequeño, así que es mi mamá para toda la vida, si mami Carmen es todo va por usted, agradecerle seria muy poco de mi parte TE AMO.

Papa gracias por ser mi padre ejemplar, y darme consejos para ser un buen hijo y mejor persona, desde pequeño me enseñaste valores y me corregías en los errores que cometía, esta sustentación va para ti, se que no me has dado todo lo que un hijo quiere, pero lo poco que ha tenido lo he valorado mucho y ese amor que un hijo quiere hacia un padre, usted me lo ha dado. Aun me cuesta estar parado en público, pero te lo voy a demostrar que si puedo por tu enseñaste a creer en mi.

Mis compas Gregory Ricardo, Jhonny Borbor, Julio Enrique, Joshua Reyes, Valeria Gongora, Maeba nazareno, Kerly Pozo, Gabriela tómalala, Maria Fernanda Mera, Kenjy Bazan, Ruben alegría, Gracias por permitirme incluir como unos de sus amigos le debo mucho este proceso académico, ya que sin ustedes no seria posible estar donde estoy, gracias por darme y extenderme la mano cuando todo parecía perdido. Me los llevare en mi memoria espero algún día no se borren nuestros recuerdos, muchachos los quiero mucho en especial a dos personas que han estado en la buenas y malas conmigo, si son ustedes Gregory y Jhonny espero que siempre

prevalezca nuestra amistad. Quien diría que íbamos hacer compas, jajaja muchachos o ingenieros vamos por una heladitas al patito, confirmen.

Y por ultimo esta va dedicado a 10 personas que están en el cielo espero que no se me escape nadie, angelitos esto es para ustedes, perdonenme por no apurarme en graduarme, pero estoy feliz de ser parte de su familia. Aquí estoy escribiendo para ustedes solo les pido que me ayuden hacerme fuerte mas tarde para poder sustentar y que todo me va bien. TIA KETTY Y MAMI BERTHA las extraño pero se que están contentas de que estoy apunto de llegar a la meta, les pido que guíen en cada rincón que me encuentre. Lo que recuerdo como si fuese ayer que estaban muy alegres por verme todo un profesional y me preguntaban que quería ser y yo todo tímido no sabio que contestar. Pero aquí esta su Christito cumpliéndole esa promesa. MAMI BERTHA te amo se que no pudiste verme graduar en el colegio, tampoco mañana en mi sustentación, pero solo te pido que no me dejes solo, y seas mi ángel de la guarda, te cuento que tengo miedo y me ganas los nervios, ayúdame, abrázame y dame ese empujoncito del éxito.

Christian Reyes

INDICE GENERAL

TITULO DE TRABAJO DE TITULACION	I
TRIBUNAL DE SUSTENTACIÓN.....	II
DECLARACIÓN DE RESPONSABILIDAD.....	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	¡Error! Marcador no definido.
AUTORIZACIÓN	V
AGRADECIMIENTO	VI
DEDICATORIA	VII
INDICE GENERAL	IX
GRAFICAS.....	XV
TABLAS.....	XVI
RESUMEN	XVII
ABSTRACT.....	XVIII
GLOSARIO	XIX
INTRODUCCIÓN	22
CAPITULO I	24
1.1 Identificación del problema	24

1.2	Antecedentes	26
1.3	Descripción del proyecto	28
1.4	Objetivo del proyecto.....	29
1.4.1	Objetivo general:.....	29
1.4.2	Objetivos específicos:	29
1.5	Justificación	30
CAPITULO II		31
2.1	Fundamentos de la RAN Virtualizada (vRAN).....	31
2.1.1	Evolución de la RAN hacia la Virtualización.....	31
2.1.2	Componentes Arquitectónicos Principales	33
2.1.3	Beneficios de la vRAN	34
2.1.4	Desafíos Técnicos en la Implementación de la vRAN	34
2.1.5	Integración con Tecnologías Emergentes	35
2.2	Rol de las Unidades de Banda Base Virtualizadas (vBBUs).....	35
2.2.1	Funciones principales de las vBBUs.....	37
2.2.2	Arquitectura en vRAN y O-RAN: DIVISION FUNCIONAL CU/DU/RU .	39
2.2.3	Beneficios Operacionales.....	41
2.2.4	Desafíos operacionales y técnicos de la vRAN/O-RAN.....	41
2.2.5	Algoritmos de Orquestación y Optimización de vBBUs.....	43
2.2.6	Integración con Tecnologías Emergentes	45

2.3	Desafíos en la Orquestación de vRAN	46
2.3.1	Complejidad en la Asignación de Recursos.....	48
2.3.2	Interoperabilidad Multivendor	49
2.3.3	Gestión de Fallos y Resiliencia.....	50
2.3.4	Latencia y Restricciones del Fronthaul en vRAN/O-RAN.....	51
2.3.5	Eficiencia Energética en la Orquestación de vBBUs.....	52
2.3.6	Consumo energético RAN Tradicional.....	53
2.3.7	Seguridad en Arquitecturas Abiertas	54
2.4	Algoritmos para la Orquestación y Asignación de Recursos.....	55
2.4.1	Optimización Matemática y Programación Lineal	57
2.4.2	Algoritmos Basados en Aprendizaje Automático.....	58
2.4.3	Aprendizaje por Refuerzo Profundo (DRL)	59
2.4.4	Algoritmos de Aprendizaje de Bandidos	60
2.4.5	Comparación de Enfoques para Orquestación de vBBUs	62
2.5	Google OR-Tools en Optimización de Redes.....	63
2.5.1	Aplicaciones en Optimización de Redes.....	64
2.5.2	Ventajas y Limitaciones.....	65
2.5.3	Integración con Python para Simulaciones.....	66
CAPITULO III.....		68
3.1	Enfoque Metodológico de la Investigación	68

3.1.1	Descripción general del método.....	69
3.1.2	Etapas del proceso de investigación	70
3.1.3	Criterios de validación del algoritmo.....	72
3.2	Diagrama de Flujo Metodológico	74
3.2.1	Explicación del flujo de trabajo	75
3.2.2	Iteración y retroalimentación del proceso.....	76
3.3	Herramientas de trabajo	78
3.3.1	Hardware Utilizado.....	78
3.3.2	Herramientas en Software.....	79
3.4	Diseño de la propuesta de orquestación.....	82
3.4.1	Modelado conceptual de la orquestación.....	83
3.4.2	Parámetros considerados en la simulación	84
3.4.3	Restricciones del modelo	85
3.5	Selección del Algoritmo	87
3.5.1	Comparación entre solvers de Google OR-Tools	88
3.5.2	Justificación para la selección del CP-SAT	91
3.5.3	Limitaciones de los solvers alternativos	92
3.6	Modelo Matemático del Problema de Orquestación.....	92
3.6.1	Definición de variables	93
3.6.2	Formulación de restricciones	94

3.6.3	Función objetivo	96
3.6.4	Interpretación operativa del modelo	96
3.7	Algoritmo CP-SAT Implementado	97
3.7.1	Estructura del algoritmo.....	98
3.7.2	Flujo de ejecución en OR-Tools	101
3.7.3	Proceso de resolución y extracción de resultados.....	102
CAPITULO IV.....		103
4.1	Introducción del capítulo	103
4.1.1	Resultados del Escenario 1: Solver CP-SAT	104
4.1.2	Servidores activos según la carga de CPU (CP-SAT)	105
4.1.3	Consumo energético total del sistema (CP-SAT).....	107
4.1.4	Consumo total de CPU utilizado por prueba (CP-SAT).....	110
4.1.5	Distribución de vBBUs por servidor (CP-SAT).....	112
4.2	Resultados del Escenario 2: Solver SCIP	114
4.2.1	Servidores activos según la carga de CPU (SCIP).....	115
4.2.2	Consumo energético total del sistema (SCIP)	117
4.2.3	Análisis del consumo energético de una RAN Tradicional vs RAN virtualizada	120
4.2.4	Consumo total de CPU utilizado (SCIP)	123
4.2.5	Comparación general entre Escenarios 1 (CP-SAT) y Escenario 2 (SCIP)	126

4.2.6 Síntesis de los resultados obtenidos	127
Conclusiones	127
Recomendaciones	128
BIBLIOGRAFIA	130

GRAFICAS

Figura 1- Arquitectura RAN por generación[22].....	32
Figura 2-La última arquitectura compatible con segmentación de O-RAN[23].....	34
Figura 3- La arquitectura O-RAN[25]	37
Figura 4 Descripción general de la arquitectura O-RAN[29].....	38
Figura 5- Ocho divisiones funcionales definen cómo se asigna la pila 5G NR a los nodos lógicos.[32]	40
Figura 6-Escenarios de implementación de NF de O-RAN en los sitios subyacentes de O-Cloud y sitios de red celular[23].....	43
Figura 7 Diagrama de flujo metodológico para la evaluación del algoritmo de orquestación en vRAN.	74
Figura 8- Arquitectura propuesta de orquestación.....	82
Figura 9- Servidores activos vs CPU por vBBU utilizando CP-SAT.....	106
Figura 10-Consumo energético total del sistema utilizando CP-SAT	109
Figura 11-onsumo total de CPU por prueba utilizando CP-SAT	111
Figura 12-Servidores activos vs CPU por vBBU utilizando SCIP.....	116
Figura 13-Consumo energético total del sistema utilizando SCIP	118
Figura 14-Consumo Energético — RAN Tradicional vs RAN Virtualizada	123
Figura 15-Consumo total de CPU utilizando SCIP	124

TABLAS

Tabla 1-Comparación de solvers en Google OR-Tools para la orquestación en vRAN. .	89
Tabla 2-Servidores activos por escenario (CP-SAT).....	106
Tabla 3-Consumo energético total por prueba (CP-SAT)	108
Tabla 4-Consumo total de CPU en cada prueba (CP-SAT).....	110
Tabla 5-Asignación óptima de vBBUs por servidor (CP-SAT)	113
Tabla 6-Servidores activos por prueba utilizando SCIP	116
Tabla 7-Consumo energético total por prueba (SCIP).....	118
Tabla 8-COMPARACIÓN DIRECTA RAN TRADICIONAL VS VIRTUALIZADA	122
Tabla 9-Consumo total de CPU en cada prueba (SCIP).....	124
Tabla 10-Comparación general entre CP-SAT y SCIP.....	126

RESUMEN

La evolución de las redes móviles hacia arquitecturas 5G ha impulsado la adopción de la Red de Acceso Radio Virtualizada (vRAN) como una alternativa flexible frente a los sistemas tradicionales basados en hardware dedicado. En este entorno, las unidades de banda base virtualizadas (vBBUs) permiten desacoplar las funciones de procesamiento de la infraestructura física, facilitando una gestión más eficiente de los recursos computacionales. No obstante, coordinar las unidades es un desafío técnico significativo, cuando se presentan fluctuaciones dinámicas en la carga de tráfico y limitaciones asociadas con el uso de la energía.

Este estudio examina el rendimiento del algoritmo de orquestación en una arquitectura vRAN, tomando en cuenta estándares de asignación de la CPU y el consumo energético de la misma. Para esto, se empleó Google OR-Tools para crear un modelo de simulación en Python y evaluar varias situaciones operativas. Los resultados obtenidos muestran que una adecuada orquestación de las vBBUs contribuye a optimizar el uso de los servidores disponibles, reduciendo el consumo energético y mejorando la eficiencia global del sistema, lo que resulta relevante para el diseño de redes vRAN orientadas a 5G.

Palabras clave: vRAN, vBBU, orquestación, aprovisionamiento dinámico, gestión de recursos, OR-Tools, Python, redes 5G.

ABSTRACT

The evolution of mobile networks toward 5G architectures has driven the adoption of Virtualized Radio Access Networks (vRAN) as a flexible alternative to traditional systems based on dedicated hardware. In this context, virtualized baseband units (vBBUs) enable the decoupling of processing functions from the physical infrastructure, facilitating more efficient management of computational resources. However, coordinating these units represents a significant technical challenge when dynamic fluctuations in traffic load and energy consumption constraints are present.

This study examines the performance of an orchestration algorithm within a vRAN architecture, taking into account CPU allocation standards and energy consumption. To this end, Google OR-Tools was used to develop a simulation model in Python and to evaluate various operational scenarios. The results show that proper orchestration of vBBUs helps optimize the use of available servers, reducing energy consumption and improving the overall system efficiency, which is relevant for the design of 5G-oriented vRAN networks.

Keywords: vRAN, vBBU, orchestration, dynamic provisioning, resource management, OR-Tools, Python, 5G networks.

GLOSARIO

AIInterface:

Interfaz estandarizada de O-RAN que conecta el Non-Real-Time RIC con el Near-Real-Time RIC. Permite transmitir políticas, objetivos de rendimiento y configuraciones de alto nivel para la optimización de la red en periodos de tiempo superiores a un segundo.

COTS (Commercial off the Shelf):

Hardware de propósito general que se utiliza para ejecutar funciones virtualizadas en vRAN, reemplazando equipos propietarios especializados y reduciendo costos operativos y de capital.

CP (Constraint Programming):

Perspectiva de optimización fundamentada en la determinación de restricciones, dominios y variables, se utiliza en OR-Tools para solucionar problemas complicados de scheduling y asignación en redes.

CPRI (Common Public Radio Interface):

Protocolo de comunicación que permite la interconexión entre las unidades de radio (RUs) y las unidades de banda base (BBUs/vBBUs) a través de un vínculo rápido para transmitir señales en fronthaul.

CU (Central Unit):

Parte de la arquitectura vRAN que se encarga de procesar las funciones de capa superior y control, como PDCP y RRC, la gestión centralizada y la escalabilidad son posibles gracias a su implementación virtual.

DRL (Deep Reinforcement Learning):

Método de aprendizaje reforzado que incluye redes neuronales profundas para la toma de decisiones en situaciones cambiantes. Se utiliza en vRAN para la orquestación inteligente en tiempo real.

DU (Distributed Unit):

Unidad encargada de llevar a cabo funciones de la capa MAC, RLC y una fracción de la PHY. Se utiliza en situaciones de borde con el fin de reducir la latencia y mejorar la respuesta del sistema.

eCPRI (Enhanced CPRI):

Unidad encargada de llevar a cabo funciones de la capa MAC, RLC y una fracción de la PHY. Se utiliza en situaciones de borde con el fin de reducir la latencia y mejorar la respuesta del sistema.

eMBB (Enhanced Mobile Broadband):

Servicio de 5G diseñado para ofrecer altas velocidades y una gran capacidad para aplicaciones como el streaming, la realidad aumentada y el contenido en 4K/8K.

Fronthaul:

Parte de la red que une las unidades de radio (RUs) con las vBBUs. Su desempeño es crítico debido a los exigentes requisitos de latencia y capacidad.

ILP (Integer Linear Programming):

Técnica de optimización matemática utilizada para resolver problemas de asignación y planificación mediante funciones lineales y variables enteras.

NFV (Network Function Virtualization):

Tecnología que permite ejecutar funciones de red como software sobre hardware genérico, habilitando la virtualización de funciones como BBUs, DUs y CUs.

O RAN (Open Radio Access Network):

Arquitectura abierta basada en interfaces estandarizadas, interoperabilidad multivendedor y controladores inteligentes (RIC) para hacer más eficiente la red a través de la inteligencia artificial.

RIC (RAN Intelligent Controller):

El controlador inteligente de O-RAN se divide en Non-RT RIC y Near-RT RIC, lo que posibilita la optimización a través de políticas, inteligencia artificial y aplicaciones xApps.

RU (Radio Unit):

Unidad responsable de la recepción y transmisión de señales de radiofrecuencia, su comunicación inicia con la DU a través del fronthaul.

SDN (Software Defined Networking):

Modelo que divide el plano de control y el plano de datos, posibilitando una administración programable y centralizada de la red.

Slicing de Red:

Método que posibilita la creación de particiones virtuales autónomas en una misma infraestructura física, distribuyendo recursos particulares de acuerdo con las necesidades del servicio.

SMO (Service Management and Orchestration):

Sistema O-RAN que se ocupa de la gestión, supervisión y orquestación de los recursos de red, abarcando el ciclo de vida de las funciones virtualizadas.

URLLC (Ultra Reliable Low Latency Communications):

Un servicio 5G enfocado en comunicaciones de muy alta fiabilidad y con latencia extremadamente baja, necesario para aplicaciones críticas como la automatización industrial y los vehículos autónomos.

vBBU (Virtual Base band Unit):

Aplicación virtualizada de la unidad de banda base que se ocupa del procesamiento de señales digitales y las tareas de control en la RAN. Su virtualización posibilita la elasticidad, la escalabilidad y una mayor eficiencia en las operaciones.

vRAN (Virtualized Radio Access Network):

Arquitectura de red que separa las funciones de hardware y software, transfiriendo el procesamiento de señal a entornos virtualizados para mejorar flexibilidad, escalabilidad y eficiencia.

XApp:

Aplicación desplegada en el Near-Real-Time RIC que ejecuta algoritmos de optimización basados en IA/ML para ajustar parámetros de red en tiempo cercano al real.

INTRODUCCIÓN

El crecimiento acelerado en el uso de servicios móviles, impulsado por la expansión de dispositivos inteligentes, aplicaciones en tiempo real y sistemas IoT, ha incrementado significativamente las demandas de capacidad, cobertura y baja latencia en las redes de acceso radio. Ante este escenario, las arquitecturas tradicionales basadas en hardware propietario han mostrado limitaciones para adaptarse a los requisitos dinámicos del ecosistema 5G y las futuras redes 6G[1] [2].

La virtualización de la red de acceso radio (vRAN) surge como una solución determinada para restricciones, ya que permite separa el hardware del software al llevar a cabo el proceso y funcionalidad de una infraestructura de red, que se encuentran fundamentadas en la nube o servidores COTS, esta metodología permite que las tecnologías avanzadas, tanto de automatización e inteligencia artificial sean utilizadas para gestionar el tráfico móvil de manera simultánea e inteligente, también tienen la capacidad de escalar la eficacia operativa [2].

Dentro de este entorno, las unidades de banda base virtualizadas (vBBUs) desempeñan un rol crítico en el procesamiento de señales, la coordinación radio y la ejecución de funciones de capa física y superiores. Su correcta orquestación implica retos complejos relacionados con el control de actividad, la asignación dinámica de recursos, el aprovisionamiento de capacidad, la sincronización de procesos y la gestión de fallos en tiempo real. La creciente demanda de eficiencia energética, resiliencia y rendimiento extremo en redes 5G y 6G incrementa la necesidad de contar con algoritmos de orquestación más completos y adaptativos [3] [4].

La orquestación eficiente de vBBUs no solo implica disponer de funciones virtualizadas, sino garantizar que estas operen de forma coordinada, optimizando el uso del hardware subyacente,

ajustándose a las variaciones del tráfico y manteniendo niveles estrictos de calidad de servicio. Una mala asignación puede resultar en sobrecarga de servidores, latencias altas, fallos de servicio y uso excesivo de energía. Esto tiene un impacto directo sobre la experiencia del usuario y sobre la sostenibilidad operacional de los operadores móviles [5].

En este marco, el algoritmo de orquestación que se examina en esta investigación es basado en métodos matemáticos de optimización a través de Google OR-Tools. Se emplea el solver CP-SAT para distribuir recursos eficazmente en un ambiente de vRAN. Se crea un modelo de simulación en Python que tiene la capacidad de examinar situaciones de tráfico dinámico, disponibilidad del hardware, limitaciones energéticas y de latencia y límites en cuanto a la capacidad del CPU, con el objetivo de establecer qué tan eficiente, adaptable y escalable es el método sugerido. El propósito de esta investigación es proporcionar una solución práctica y basada en fundamentos que ayude a desarrollar redes móviles más inteligentes, resistentes y que estén preparadas para las demandas del 5G avanzado y la transición hacia 6G [6].

CAPITULO I

1.1 Identificación del problema

La transición hacia la Red de Acceso Radio Virtualizada (vRAN) ha permitido que los operadores de telecomunicaciones gestionen sus infraestructuras con mayor flexibilidad y eficiencia. No obstante, este cambio también ha generado desafíos técnicos que dificultan su implementación óptima, particularmente en lo relacionado con la orquestación de las unidades de banda base virtualizadas (vBBUs). Para garantizar que las funciones virtualizadas operen de forma coordinada y puedan reaccionar apropiadamente a las fluctuaciones dinámicas del tráfico en tiempo real, estas unidades constituyen un elemento fundamental [7].

El control de actividad de las vBBUs en entornos virtualizados es uno de los problemas principales que se han identificado. Estas unidades tienen que ser activadas o desactivadas dependiendo de la carga del tráfico, las condiciones ambientales y el estado operativo de la red, una gestión ineficiente puede ocasionar consumo energético innecesario, sobrecarga de servidores o degradación en la calidad del servicio [8].

A ello se suma el desafío del aprovisionamiento y la gestión de los recursos físicos de red. Aunque la virtualización desacopla las funciones del hardware propietario, la infraestructura subyacente continúa siendo determinante para el rendimiento. Por ello, es necesario distribuir de manera eficiente los recursos computacionales disponibles —como procesamiento, memoria, almacenamiento y ancho de banda— con el fin de evitar cuellos de botella que afecten el desempeño de las vBBUs [9].

La sincronización y programación de las funciones de red es otro elemento crucial. La naturaleza del procesamiento en las comunicaciones móviles requiere una ejecución bajo

condiciones de alta carga y una rigurosa precisión temporal. La latencia puede verse afectada, la sincronización entre nodos puede cambiar y la experiencia del usuario final puede verse comprometida debido a cualquier desfase en el proceso de programación [5].

Asimismo, la distribución dinámica de recursos en tiempo real supone un reto permanente. Como las solicitudes de tráfico son sumamente cambiantes e imprevisibles, la red necesita sistemas que puedan redistribuir los recursos en milisegundos sin poner en peligro la estabilidad operativa ni la eficiencia espectral [4].

Por último, para la administración integral de las vBBUs es necesario contar con procedimientos sólidos para la detección de errores, el seguimiento del rendimiento, la planificación de capacidad y la optimización de los recursos existentes. Si estas capacidades no están presentes, la red puede deteriorarse rápidamente en situaciones críticas, lo que afecta tanto a los usuarios individuales como a los servicios estratégicos que requieren de la conectividad móvil [10].

Estos problemas, en su totalidad, muestran que la orquestación en entornos vRAN no es un procedimiento elemental de configuración, sino un reto complicado que requiere soluciones sofisticadas, flexibles y adaptativas. La ausencia de estrategias eficaces puede generar pérdidas económicas, uso ineficiente de la infraestructura y deterioro significativo en la calidad del servicio percibida por los usuarios [2].

1.2 Antecedentes

La evolución de las telecomunicaciones móviles ha estado impulsada por la necesidad de incrementar la capacidad, reducir la latencia y optimizar el uso de los recursos de red. Desde las primeras generaciones, la unidad de banda base (BBU) desempeñó un papel crucial en el procesamiento de señal dentro de las estaciones base, aunque estuvo limitada por su dependencia del hardware propietario. Con la introducción de 4G LTE surgió el modelo Cloud RAN (C-RAN), que permitió centralizar funciones de múltiples BBUs en centros de datos, logrando mejoras en eficiencia espectral y reducción de costos operativos. No obstante, C-RAN aún mantenía restricciones de flexibilidad y dependencia del equipamiento propietario [7].

Con el despliegue de 5G, estas transformaciones se intensificaron debido a los requisitos de flexibilidad, escalabilidad y baja latencia. En este escenario surgió la Red de Acceso Radio Virtualizada (vRAN), en la cual las funciones de las BBUs se ejecutan como software en plataformas de nube o infraestructura virtualizada. Este enfoque habilita una gestión más dinámica de los recursos y facilita la adopción de mecanismos de orquestación capaces de configurar, activar y coordinar funciones en función de la demanda cambiante del tráfico [9].

A nivel internacional, diversas iniciativas han promovido la estandarización de interfaces abiertas y la interoperabilidad en redes virtualizadas, siendo O-RAN Alliance una de las organizaciones más relevantes en este ámbito. Simultáneamente, se han sugerido algoritmos de asignación dinámica de recursos, que tienen como objetivo optimizar la calidad del servicio y la eficiencia energética en entornos móviles. No obstante, una gran parte de estos esfuerzos se ha enfocado en modelos teóricos o simulaciones, lo que ha dejado un vacío en cuanto a la evaluación completa de los algoritmos de orquestación aplicados a vBBUs, sobre todo en lo que respecta al control de actividad, la provisión de recursos físicos y la gestión de fallos en tiempo real [7].

En la actualidad, la expansión de servicios de telecomunicaciones en el Ecuador se ha ido incrementando de manera constante, según información tomada del ministerio de telecomunicaciones, mas del 80% de población tiene cobertura estandarizada de 4G LTE y son atendidas por mas de 14 millones de individuos, La agencia de regularización y control de los sistemas de telecomunicaciones (ARCOTEL) indica que el 83% de la población ya tiene acceso a internet, tanto móvil como de manera fija. Esto demuestra una creciente necesidad de conectividad en el país [11][12].

1.3 Descripción del proyecto

El presente proyecto se orienta a evaluar un algoritmo de orquestación aplicado a una Red de Acceso Radio Virtualizada (vRAN), con énfasis en la coordinación de las unidades de banda base virtualizadas (vBBUs). La motivación principal surge de la necesidad de que los operadores de telecomunicaciones dispongan de redes móviles más flexibles, escalables y adaptables en tiempo real, características que las arquitecturas tradicionales basadas en hardware propietario no siempre pueden ofrecer [13].

La investigación se enfoca en diseñar y analizar un algoritmo que aborde los principales desafíos asociados a las vBBUs, entre ellos el control de actividad, el aprovisionamiento y gestión de recursos físicos, la programación de funciones críticas, la asignación dinámica de recursos y la gestión integral frente a fallos o variaciones de rendimiento. Estos aspectos son fundamentales para alcanzar niveles óptimos de eficiencia y resiliencia en la infraestructura virtualizada [14].

Para la fase de validación, el proyecto emplea Google OR-Tools, una biblioteca especializada en optimización combinatoria y programación de restricciones que se integra con el lenguaje Python. Esta herramienta permite modelar los problemas de orquestación relacionados con control de actividad, aprovisionamiento, asignación dinámica y gestión de fallos, facilitando la simulación bajo distintos escenarios de carga y demanda de red [6].

El resultado esperado es desarrollar un marco de evaluación que permita identificar las ventajas y limitaciones del algoritmo propuesto frente a las demandas actuales de las redes 5G y su evolución hacia futuras generaciones móviles. Con ello, el proyecto busca constituir un aporte académico y, al mismo tiempo, un insumo práctico para operadores y desarrolladores interesados en optimizar la gestión de recursos en entornos de virtualización [15].

1.4 Objetivo del proyecto

1.4.1 Objetivo general:

Evaluar un algoritmo de orquestación de funciones en una RAN virtualizada, utilizando la herramienta Google OR-Tools en Python, con el fin de optimizar el control de la actividad de las unidades de banda base, el aprovisionamiento de recursos (CPU) y consumo energético.

1.4.2 Objetivos específicos:

- ✓ Analizar los fundamentos técnicos de la RAN virtualizada y el rol que cumplen las vBBUs e identificar los principales problemas de orquestación en entornos de redes móviles relacionados con la asignación de recursos y la gestión de fallos.
- ✓ Seleccionar el algoritmo que cubra las necesidades de orquestación en el objetivo 1.
- ✓ Diseñar un modelo de simulación en Python empleando Google OR-Tools como herramienta de optimización matemática y Evaluar los resultados del modelo propuesto en términos de eficiencia, escalabilidad y capacidad de respuesta frente a los desafíos de la red.

1.5 Justificación

El crecimiento acelerado en el uso de servicios móviles, junto con la masificación de dispositivos inteligentes y las demandas de baja latencia y alta capacidad, ha evidenciado las limitaciones de las arquitecturas tradicionales de la red de acceso radio (RAN). En este escenario, la virtualización de la RAN (vRAN) se consolida como una solución estratégica que proporciona mayor flexibilidad, eficiencia y capacidad de adaptación en la gestión de recursos de red [13].

La orquestación de las unidades de banda base virtualizadas (vBBUs) representa un desafío fundamental dentro de la vRAN, ya que su gestión adecuada influye directamente en la calidad del servicio, el rendimiento global de la red y el consumo de recursos. Evaluar y desarrollar algoritmos de orquestación eficientes permite optimizar la asignación dinámica de recursos, mejorar la respuesta ante fallos y garantizar un funcionamiento más estable y confiable de la red móvil [14].

Desde una perspectiva académica, este estudio contribuye a fortalecer el conocimiento relacionado con la virtualización de redes móviles al proporcionar un marco analítico centrado en la evaluación de algoritmos de orquestación en situaciones realistas. El análisis también tiene consecuencias en la práctica para los operadores de telecomunicaciones, ya que ofrece métodos y pautas que se pueden aplicar para optimizar las infraestructuras vRAN. Esto repercute directamente la sostenibilidad digital del ecosistema y en la eficiencia de operación [16][6].

A nivel nacional, el crecimiento de la cobertura de 4G y 5G a reforzado la importancia de la siguiente información, debido a que contribuye al mejoramiento de la gestión de los servicios y recursos, también ha ayudado a la planificación de capacidad y calidad de servicio (QoS), aportando con aspectos importantes para el desarrollo tecnológico del país y para dar mejor servicio a los usuarios finales [12].

CAPITULO II

2.1 Fundamentos de la RAN Virtualizada (vRAN)

La Red de Acceso Radio Virtualizada (vRAN) representa la transición desde infraestructuras móviles tradicionales, rígidas y dependientes de hardware propietario hacia arquitecturas más abiertas, flexibles y eficientes. En este enfoque, las funciones de red se ejecutan como software desacoplado del hardware físico, utilizando servidores de propósito general (COTS) para realizar tareas de procesamiento, lo que permite mejorar la escalabilidad y reducir costos operativos. Este paradigma resulta especialmente relevante en entornos 5G, donde es necesario gestionar grandes volúmenes de datos, baja latencia y una diversidad creciente de servicios. Estudios recientes destacan que la vRAN no solo optimiza el uso de los recursos disponibles, sino que también habilita una gestión dinámica de la red mediante técnicas de inteligencia artificial [17][15].

2.1.1 Evolución de la RAN hacia la Virtualización

La evolución de la red de acceso radio (RAN) ha transitado por distintas etapas tecnológicas. En un primer momento surgieron las arquitecturas distribuidas (D-RAN), en las cuales el procesamiento de señal se realizaba directamente en cada estación base. Posteriormente apareció el modelo centralizado (C-RAN), que concentraba las funciones de procesamiento en ubicaciones centrales con el fin de mejorar la eficiencia operativa. La Red de Acceso Radio Virtualizada (vRAN), como evolución natural de ambos enfoques, incorpora tecnologías de virtualización como Network Function Virtualization (NFV) y Software-Defined Networking (SDN), permitiendo ejecutar funciones de red como software en entornos de nube o infraestructura virtualizada [15][18].

El proceso de transformación se aceleró con la llegada del 5G, que requiere una flexibilidad más alta para respaldar servicios variados como ultra confiable de baja latencia, comunicaciones masivas tipo máquina y banda ancha móvil mejorada. La adopción de arquitecturas virtualizadas que pueden adaptarse dinámicamente a las demandas del tráfico fue impulsada por la necesidad de baja latencia, alta capacidad y escalabilidad en estos servicios [19][20].

La implementación del concepto Open RAN fue un hito importante en la apertura de las redes móviles. Este se empezó a adoptar en 2016 con proyectos como el Telecom Infra Project (TIP), cuyo objetivo es fomentar la separación de funciones y la estandarización de interfaces abiertas para asegurar la interoperabilidad entre varios proveedores [21].

Una variedad de estudios y documentos técnicos muestran gráficos comparativos que reflejan la progresión desde D-RAN a C-RAN y vRAN. Estos modelos permiten visualizar cómo la virtualización reduce la dependencia del hardware propietario y mejora la escalabilidad y la eficiencia operativa [9].

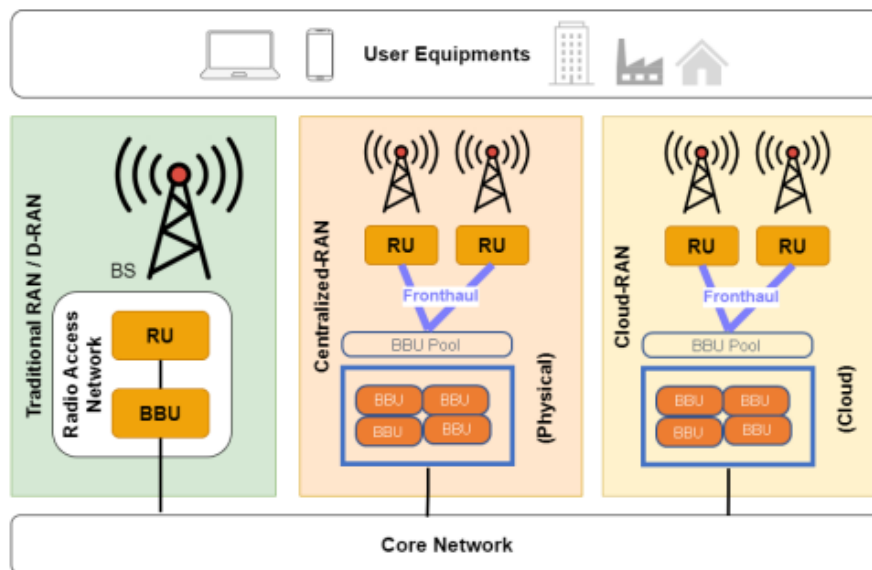


Figura 1- Arquitectura RAN por generación[22]

2.1.2 Componentes Arquitectónicos Principales

La arquitectura de la vRAN se basa en la desagregación de funciones en diferentes unidades, como la unidad de radio (RU), la unidad distribuida (DU) y la unidad central (CU), las cuales se conectan mediante interfaces abiertas como el Open Fronthaul. En el marco de O-RAN se incorporan controladores inteligentes, entre ellos el Near-Real-Time RAN Intelligent Controller y el Non-Real-Time RIC, que permiten optimizar la red mediante técnicas de inteligencia artificial y aprendizaje automático [15].

El Service Management and Orchestration (SMO) actúa como el orquestador central encargado de gestionar las operaciones sobre la nube O-Cloud y garantizar la interoperabilidad entre los diferentes componentes del sistema. Asimismo, la arquitectura O-RAN admite el concepto de network slicing, que permite la creación de particiones virtuales independientes para asignar recursos de manera dinámica según las necesidades específicas de cada servicio o aplicación [13].

Para representar la arquitectura O-RAN, diversos documentos técnicos emplean diagramas que muestran la interacción entre las unidades RU, DU y CU, así como las interfaces abiertas O1, A1 y E2 que permiten el flujo de información y control entre los distintos componentes. Estas representaciones facilitan la comprensión del modelo de desagregación y del rol del RIC dentro del ecosistema O-RAN [21].

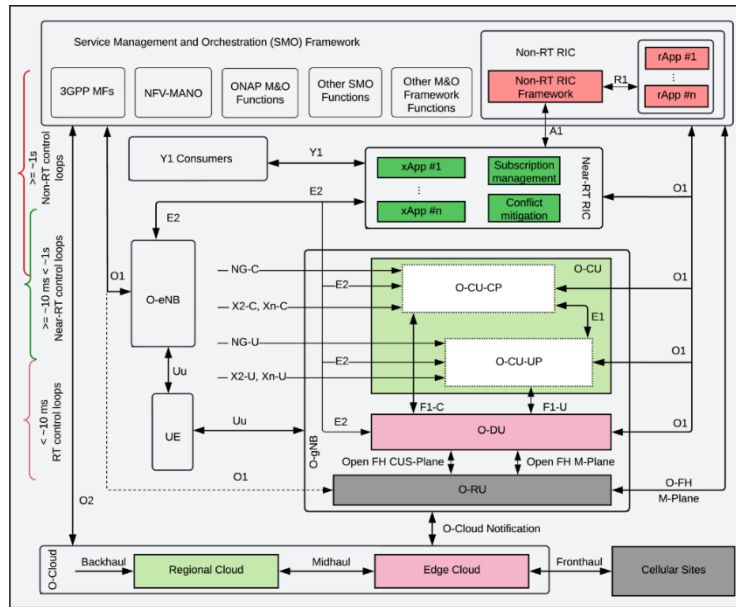


Figura 2-La última arquitectura compatible con segmentación de O-RAN[23]

2.1.3 Beneficios de la vRAN

Entre los beneficios más relevantes de la vRAN se encuentra la reducción de los costos operativos (OpEx) y de capital (CapEx), resultado del uso de hardware genérico y de la interoperabilidad multivendor. Diversos análisis de despliegues comerciales, como el caso de Rakuten Mobile, indican ahorros estimados entre el 40 % y el 50 % en la implementación de redes 5G basadas en arquitecturas abiertas. Asimismo, la vRAN proporciona mayor flexibilidad para escalar recursos en tiempo real, mejora la eficiencia energética y habilita mecanismos de automatización inteligente para la gestión de la red [24][25].

2.1.4 Desafíos Técnicos en la Implementación de la vRAN

A pesar de sus beneficios, la implementación de la vRAN presenta diversos desafíos que deben ser considerados en su despliegue. Uno de los principales es la complejidad asociada a la integración multivendor, la cual puede incrementar la latencia y dificultar la gestión y detección

de fallos debido a la coexistencia de componentes provenientes de diferentes fabricantes. Además, el uso de interfaces abiertas introduce riesgos de seguridad que obligan a fortalecer los mecanismos de protección y monitoreo continuo. Otro reto relevante es la disponibilidad limitada de recursos computacionales en ubicaciones edge, lo que puede afectar el rendimiento de las funciones virtualizadas. Finalmente, la falta de estandarización completa en ciertos aspectos de la arquitectura dificulta alcanzar niveles altos de interoperabilidad entre sistemas heterogéneos [9] [21].

2.1.5 Integración con Tecnologías Emergentes

La vRAN se integra de manera natural con las funciones de inteligencia artificial y aprendizaje automático implementadas en las RAN Intelligent Controllers (RIC). Esta integración permite habilitar mecanismos avanzados de toma de decisiones, entre ellos la asignación consciente de segmentos de red, conocida como *slicing-aware orchestration*, mediante la cual los recursos se distribuyen automáticamente según los requerimientos específicos de cada dominio o servicio. Este enfoque permite mejorar la eficiencia, la flexibilidad y la adaptación dinámica de la red ante variaciones en la demanda [26].

2.2 Rol de las Unidades de Banda Base Virtualizadas (vBBUs)

La Red de Acceso Radio Virtualizada (vRAN) surge como una evolución de las arquitecturas tradicionales de acceso radio, impulsada por la necesidad de mayor flexibilidad, escalabilidad y eficiencia en el despliegue de redes móviles de nueva generación. A diferencia de la RAN convencional, donde las funciones de banda base se implementan mediante equipos físicos dedicados, la vRAN permite desacoplar dichas funciones del hardware propietario y ejecutarlas como software sobre servidores de propósito general. [27].

Esta virtualización posibilita que las funciones de procesamiento de la señal, control y gestión de recursos radio sean instanciadas dinámicamente en entornos virtualizados o basados en la nube. Como resultado, los operadores pueden adaptar la capacidad de la red de acuerdo con la demanda del tráfico, optimizando el uso de recursos computacionales y reduciendo la dependencia de infraestructura física rígida. Como resultado los operadores pueden ajustar la capacidad de la red según la demanda del tráfico y así optimizan el uso de recursos computacionales mientras reducen la dependencia de una infraestructura física rígida, además la vRAN facilita la centralización parcial o total de funciones lo que contribuye a una administración más eficiente y mejora la coordinación entre las distintas entidades de la red [9].

Desde el punto de vista operativo, la vRAN introduce nuevos desafíos. Entre ellos se encuentran la orquestación de recursos, la latencia y el consumo energético de los servidores que alojan las funciones virtualizadas. Para asegurar un rendimiento óptimo del sistema, es esencial la correcta distribución de recursos de CPU y la activación controlada de instancias virtuales. En esta línea, la vRAN implica no solamente una transformación tecnológica, sino también un cambio en el manejo y optimización de la red de acceso por radio [28].

En esta investigación, la arquitectura vRAN es particularmente importante porque posibilita examinar cómo el consumo de energía y la utilización de recursos computacionales se ven afectados por diversas estrategias de orquestación.

La posibilidad de activar o desactivar unidades virtualizadas en función de la carga de tráfico constituye la base para evaluar mecanismos de optimización que buscan reducir el número de servidores activos sin comprometer la calidad del servicio, aspecto central del modelo propuesto en este trabajo.

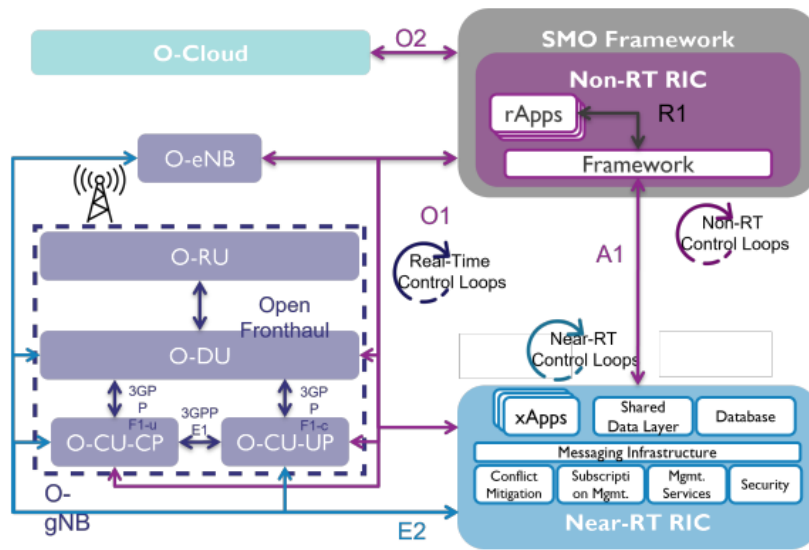


Figura 3- La arquitectura O-RAN[25]

2.2.1 Funciones principales de las vBBUs

Las unidades de banda base virtualizadas (vBBUs) cumplen un rol central en el procesamiento de señal dentro de una red de acceso radio virtualizada (vRAN). Su funcionamiento se basa en la ejecución de funciones de capa física y de control sobre plataformas de propósito general (COTS), lo cual permite una operación más flexible y escalable frente a las variaciones del tráfico. En contraposición a las BBUs convencionales, las vBBUs funcionan como elementos de software que se distribuyen en ambientes de edge computing o nube, lo cual permite una administración dinámica de los recursos y una eficiencia mayor en términos operativos [27].

Las vBBUs llevan a cabo funciones esenciales de procesamiento de señal, desde una perspectiva funcional. Entre estas se encuentran la modulación y demodulación, la codificación y decodificación de canal, la administración de los protocolos RLC y PDCP y el control del acceso al medio (MAC), estas funciones permiten establecer y mantener la comunicación entre los

usuarios y la infraestructura de red, garantizando parámetros adecuados de latencia, confiabilidad y calidad del servicio [9].

Asimismo, las vBBUs realizan la planificación de recursos radio (*scheduling*), la asignación de bloques físicos (PRB), el control de potencia y la gestión de los mecanismos HARQ, los cuales son fundamentales para optimizar el uso del espectro y mantener la estabilidad del enlace en condiciones variables de canal. Al encontrarse virtualizadas, estas tareas pueden ajustarse en tiempo real según las demandas del tráfico, mejorando el rendimiento global de la red [17].

En el marco de la arquitectura O-RAN, las vBBUs interactúan de manera directa con la unidad central (CU) y la unidad distribuida (DU), facilitando un diseño modular que favorece la interoperabilidad entre proveedores y permite ubicar cada función en el punto más adecuado de la red. Este modelo contribuye a una mayor eficiencia energética, una operación más flexible y capacidades avanzadas para la automatización y orquestación de recursos [21].

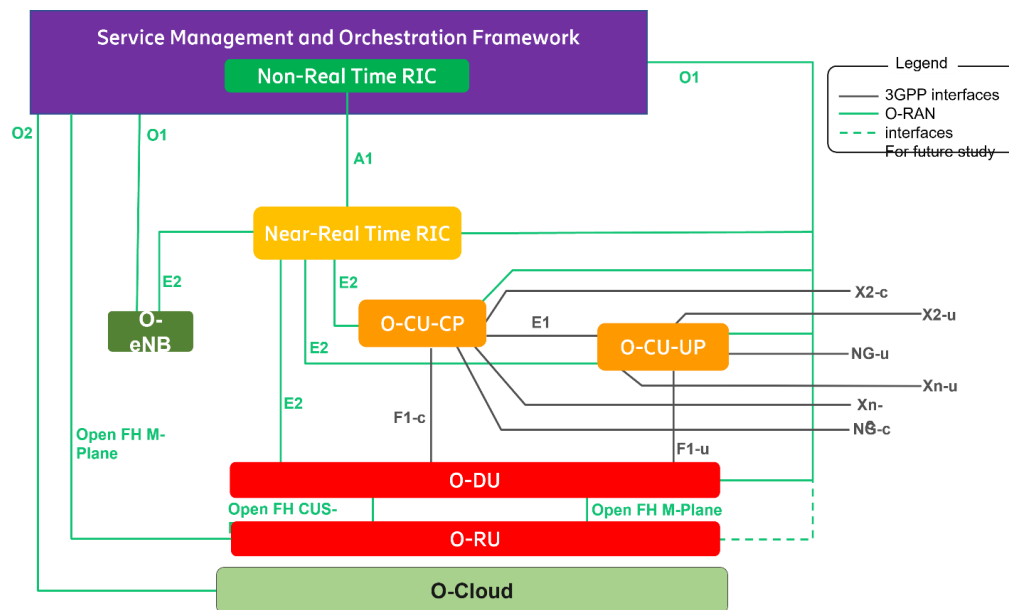


Figura 4 Descripción general de la arquitectura O-RAN[29]

2.2.2 Arquitectura en vRAN y O-RAN: DIVISION FUNCIONAL

CU/DU/RU

La arquitectura de la Red de Acceso Radio Virtualizada (vRAN) se caracteriza por la centralización de las unidades de banda base virtualizadas (vBBUs) en *pools* compartidos, los cuales se conectan a las unidades de radio remotas (RRH o RU) a través del fronthaul, mediante tecnologías como Common Public Radio Interface (CPRI) o Enhanced CPRI (eCPRI). Este diseño posibilita que los recursos se empleen de manera más eficaz y que la latencia disminuya de modo importante, gracias a la incorporación de capacidades de procesamiento distribuidas en la red [17].

La vBBU, en el caso de O-RAN, se desagrega funcionalmente en dos partes: la unidad central (O-CU) y la unidad distribuida (O-DU). Esta división posibilita la interoperabilidad multivendor y da paso a que se implementen interfaces abiertas como A1 y E2, las cuales hacen posible el ejercicio de control inteligente por medio de técnicas de aprendizaje automático e inteligencia artificial.

La interfaz E2 establece la comunicación entre el Near-Real-Time RAN Intelligent Controller (Near-RT RIC) y los nodos O-RAN, incluyendo O-CU, O-DU y O-RU, posibilitando el control en tiempo casi real de funciones esenciales como la planificación de recursos radio, la optimización del *scheduling* y la mitigación de interferencias. Además, E2 permite la ejecución de aplicaciones xApps en el Near-RT RIC, las cuales aplican algoritmos de IA para ajustar dinámicamente parámetros clave de la red [9][30].

Por su parte, la interfaz A1 conecta el Non-Real-Time RIC (Non-RT RIC) con el Near-RT RIC y habilita el intercambio de políticas de configuración, objetivos de optimización y directrices de largo plazo. A través de A1, el Service Management and Orchestration (SMO) transmite

directivas de orquestación y políticas de gestión como la priorización de servicios —por ejemplo, enhanced Mobile Broadband (eMBB) o Ultra-Reliable Low-Latency Communications (URLLC)— y reglas de *network slicing* destinadas a optimizar la asignación de recursos según las necesidades de cada servicio [21].

La desagregación funcional también habilita la implementación de distintos *functional splits*, como la opción 7-2x definida por 3GPP, en la cual la O-DU ejecuta funciones de capa física inferior (PHY-Low) y MAC en ubicaciones *edge*, mientras que la O-CU procesa funciones de capa superior como RLC y PDCP en la nube central. Este enfoque mejora la eficiencia energética, la escalabilidad y la flexibilidad del despliegue [31]

La integración con el SMO permite la automatización de la orquestación, ya que este componente coordina el ciclo de vida de las funciones virtualizadas y asegura la interoperabilidad entre los distintos nodos y controladores del entorno O-RAN. Para apoyar la comprensión de esta arquitectura, resulta útil incluir una figura que represente la desagregación de la vBBU en O-RU, O-DU y O-CU, así como las interfaces O1, A1 y E2 utilizadas para el control y gestión del sistema [9].

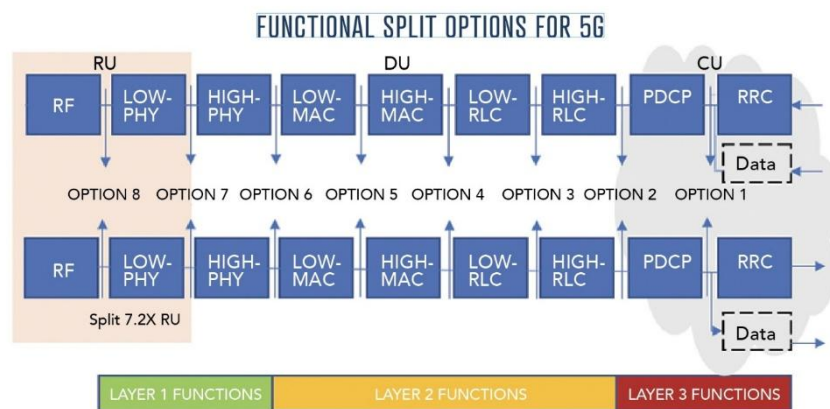


Figura 5- Ocho divisiones funcionales definen cómo se asigna la pila 5G NR a los nodos lógicos.[32]

2.2.3 Beneficios Operacionales

Los beneficios operacionales de las vBBUs se reflejan principalmente en la reducción de los costos de capital (CapEx) y operativos (OpEx), gracias al uso de hardware genérico e infraestructura multivendor. Estudios recientes indican que los despliegues 5G basados en arquitecturas abiertas pueden alcanzar eficiencias de entre el 40 % y el 50 %, debido a la agregación de recursos computacionales y a la flexibilidad en su escalado [4].

Las vBBUs permiten que se realicen migraciones dinámicas entre entornos fog y cloud, con lo cual la eficiencia energética mejora a pesar de las variaciones en las condiciones del tráfico. Es posible que la capacidad de mover funciones virtualizadas en función de la carga, como en Cloud-Fog RAN (CF-RAN), produzca un ahorro energético de hasta el 42 % y, por lo tanto, favorezca una operación más sustentable de la red [33].

Además, las vBBUs posibilitan la implementación del network slicing, lo que hace posible la asignación de recursos virtuales en función de las necesidades de cada servicio. Esto optimiza el rendimiento y la eficiencia en contextos heterogéneos o con alta demanda [9].

2.2.4 Desafíos operacionales y técnicos de la vRAN/O-RAN

La implementación de arquitecturas vRAN y O-RAN presenta diversos desafíos operacionales y técnicos que deben ser considerados en la planificación y despliegue de redes de nueva generación. Uno de los principales retos es la complejidad asociada a la integración multivendor, debido a que la coexistencia de componentes provenientes de diferentes fabricantes puede dificultar la interoperabilidad y aumentar el tiempo de resolución de fallas. La diversidad de proveedores también introduce variabilidad en los procedimientos de gestión y diagnóstico, lo que requiere sistemas sofisticados para la orquestación y el monitoreo [9].

El aumento en las exigencias de sincronización y procesamiento es otro reto significativo. Las funciones virtualizadas, en particular las críticas que funcionan en la unidad distribuida (DU), deben llevarse a cabo con niveles de latencia rigurosos. Si no se lleva a cabo una distribución eficaz de los recursos [17], la necesidad de procesar grandes cantidades de datos en tiempo real puede causar que los servidores COTS se sobrecarguen y el sistema.

La seguridad, dado que la implementación de interfaces abiertas —como A1, O1 y E2— aumenta la superficie de ataque y pone a la red en riesgo ante posibles vulnerabilidades, es también un elemento crucial. Para reducir los riesgos que potencialmente afectan la disponibilidad y fiabilidad del servicio [21], es esencial implementar de manera apropiada mecanismos de detección temprana de amenazas, cifrado y autenticación.

Desde el punto de vista operativo otro desafío es la limitada disponibilidad de recursos en las ubicaciones edge ya que las funciones distribuidas en estos lugares exigen mayor capacidad de computación para realizar tareas como HARQ programación y control de potencia, en ese contexto la infraestructura edge puede colapsar si no se aplican mecanismos de escalabilidad adecuados cuando la demanda aumenta [26].

Finalmente, la ausencia de una estandarización integral y el desarrollo desigual de las soluciones O-RAN impiden que estas arquitecturas se implementen en gran escala. A pesar del progreso en los estándares, todavía hay discrepancias entre las implementaciones, lo que crea obstáculos durante los procesos de certificación, pruebas de interoperabilidad y despliegue [31].

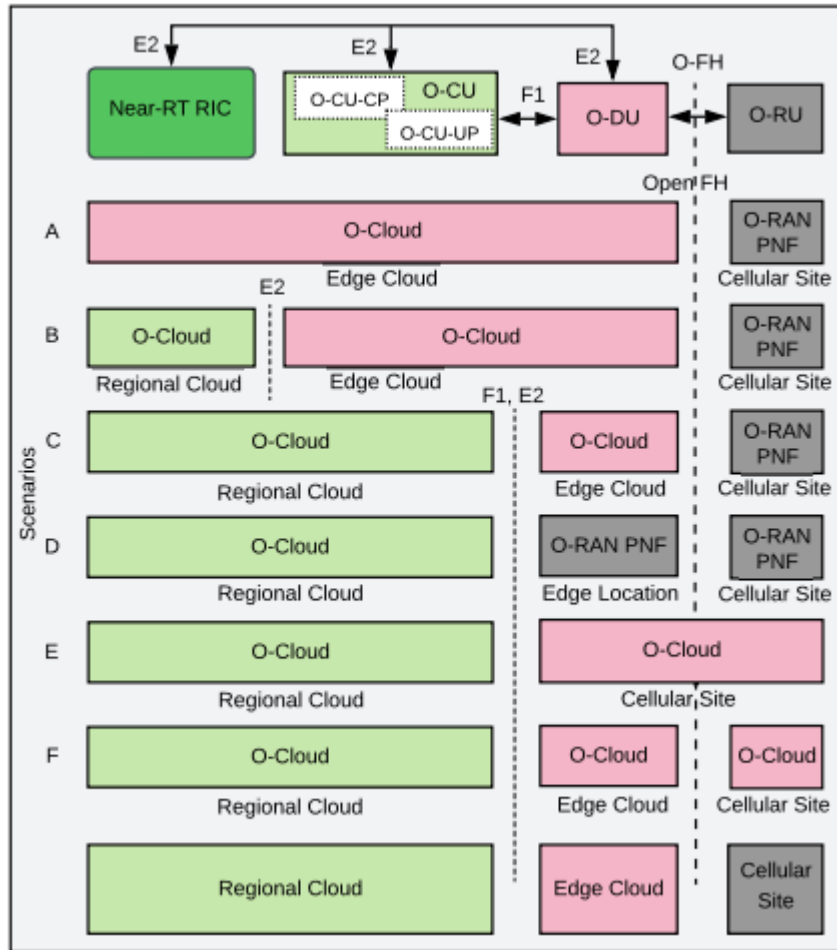


Figura 6-Escenarios de implementación de NF de O-RAN en los sitios subyacentes de O-Cloud y sitios de red celular[23]

2.2.5 Algoritmos de Orquestación y Optimización de vBBUs

La orquestación de las unidades de banda base virtualizadas (vBBUs) requiere el uso de algoritmos capaces de asignar recursos, gestionar cargas de tráfico y mantener el desempeño de la red bajo condiciones dinámicas. A diferencia de los entornos tradicionales, donde las decisiones se toman de forma estática, una arquitectura vRAN y O-RAN demanda mecanismos que operen en tiempo real o casi real, optimizando continuamente el uso de los recursos disponibles [13].

Los algoritmos para la asignación dinámica de recursos son unas de las técnicas más utilizadas en la orquestación. Su objetivo es repartir el ancho de banda, la memoria y la capacidad

computacional entre varias vBBUs en función del tráfico. Estos algoritmos, que suelen plantearse como problemas de optimización combinatoria, toman en cuenta indicadores como la latencia, la estabilidad del enlace, el rendimiento y la utilización del CPU. Por su complejidad, se utilizan frecuentemente modelos que se basan en programación matemática, así como en heurísticas y metaheurísticas, para hallar soluciones de manera eficaz [34].

Además, los métodos de orquestación tienen que incorporar algoritmos para controlar la actividad, cuyo propósito es activar, desactivar o trasladar vBBUs entre centros de datos o ubicaciones edge en función de las necesidades. Esto posibilita que se reduzca el uso de energía y que el aprovechamiento de la infraestructura física sea mejor. Para manejar esta dinámica, se aplican métodos como la programación por restricciones, los modelos fundamentados en satisfacibilidad booleana (SAT) y sus variantes, como la satisfacibilidad de programación de restricciones (CP-SAT). Estas técnicas son ampliamente utilizadas en plataformas como Google OR-Tools para solucionar dificultades de asignación y planificación en tiempo real [6].

Los algoritmos de programación y planificación son otro elemento importante que funcionan en la unidad distribuida (O-DU) con el propósito de administrar bloques físicos (PRB), adjudicar recursos radio y llevar a cabo esquemas HARQ en condiciones cambiantes del canal. Para asegurar la coherencia entre los diferentes niveles de la arquitectura O-RAN, estos algoritmos tienen que trabajar en conjunto con la unidad central (O-CU), que se encarga de las funciones de nivel más alto como RLC y PDCP [9].

Por último, es habitual incorporar inteligencia artificial y aprendizaje automático en contextos O-RAN a través de la implementación de aplicaciones rApps y xApps que funcionan en el Non-Real-Time RIC y Near-Real-Time RIC, respectivamente. Estas aplicaciones utilizan algoritmos predictivos para prever conductas de tráfico, optimizar el "network slicing", modificar

las configuraciones de transmisión o disminuir interferencias, lo que mejora la eficacia global del sistema [21].

2.2.6 Integración con Tecnologías Emergentes

La integración de tecnologías emergentes ha impulsado significativamente la evolución de la orquestación de las unidades de banda base virtualizadas (vBBUs), permitiendo una operación más eficiente, adaptativa y orientada a las necesidades dinámicas de los usuarios y servicios. Entre estas tecnologías destacan la inteligencia artificial (IA), el aprendizaje automático (ML), el *edge computing* y los enfoques *cloud-native*, los cuales fortalecen las capacidades de gestión, control y optimización dentro de arquitecturas vRAN y O-RAN [13].

Dentro de O-RAN, se concreta la integración de ML e IA mediante las aplicaciones rApps y xApps, que funcionan en el Non-Real-Time RIC y en el Near-Real-Time RIC (Near-RT RIC), respectivamente. Estas aplicaciones posibilitan el estudio de patrones de tráfico, la predicción de conductas futuras y la implementación automática de modificaciones en parámetros fundamentales de la red, como son la gestión de potencia, la optimización del rendimiento espectral, el control de interferencias o la distribución de recursos radio. incrementando la eficacia general del sistema, la capacidad de aprendizaje continuo facilita una orquestación más precisa y mejora el rendimiento global del sistema [35].

El *edge computing* también juega un rol clave al acercar las funciones de procesamiento más importantes a los usuarios finales. Esto posibilita la disminución de la latencia y permite que se tomen decisiones en tiempo casi real para funciones específicas de la O-DU, como la programación, el HARQ o la administración de bloques físicos (PRB). El procesamiento distribuido en el borde (*edge*) hace más sencilla la realización de algoritmos para asignar recursos

de manera dinámica y controlar la actividad de vBBUs, lo que mejora la eficiencia energética bajo condiciones cambiantes de carga [8].

El enfoque cloud-native integra este ecosistema al posibilitar que las vBBUs funcionen como microservicios o contenedores repartidos por plataformas en la nube. Esto propicia la migración ágil de funciones virtualizadas, el escalamiento automático y la elasticidad, que son rasgos fundamentales para sostener demandas elevadas de tráfico y asegurar la continuidad en la operación en ambientes 5G y generaciones móviles posteriores [21].

Finalmente, tecnologías como el "network slicing" se integran de forma natural con la orquestación avanzada de vBBUs y posibilitan que la red sea dividida en instancias virtuales autónomas que distribuyen recursos según las exigencias particulares de cada servicio. Esta integración posibilita la creación de entornos altamente personalizados y eficientes, ideales para casos de uso como eMBB, URLLC y mMTC, reforzando así la adaptabilidad de la red ante distintos escenarios de operación [31][36].

2.3 Desafíos en la Orquestación de vRAN

La orquestación en la Red de Acceso Radio Virtualizada (vRAN) comprende la coordinación automatizada de recursos físicos y virtuales con el fin de garantizar eficiencia, escalabilidad y fiabilidad en entornos de comunicación caracterizados por elevados niveles de dinamismo, como sucede en redes 5G. La desagregación de funciones, característica de las arquitecturas vRAN y O-RAN, introduce complejidades adicionales asociadas a la interoperabilidad entre equipos de distintos proveedores, la gestión de fallos y la optimización de recursos computacionales limitados por restricciones de latencia y capacidad energética [13].

Uno de los desafíos más relevantes consiste en realizar una asignación dinámica de recursos que permita responder adecuadamente a variaciones abruptas del tráfico sin comprometer el rendimiento de las funciones críticas. Las vBBUs, que funcionan como elementos virtualizados, necesitan mecanismos de orquestación que puedan elegir recursos, trasladar cargas laborales y modificar parámetros operativos en tiempo casi real, asegurando de este modo la estabilidad y permanencia del servicio [8].

Además, en la orquestación, el control de actividad y la capacidad de sobrevivir a los fallos son desafíos esenciales porque la repartición de tareas entre la unidad central (O-CU) y la unidad distribuida (O-DU), así como la desagregación funcional, hacen que el flujo de señalización y supervisión sean más complejos. Las aplicaciones que son sensibles pueden verse afectadas por los retrasos en la ejecución de funciones virtualizadas, especialmente en lo que respecta a la latencia extrema [9], cuando se dan situaciones de alta concurrencia o edge. Conforme a la bibliografía más actual, es imprescindible utilizar herramientas avanzadas de programación por restricciones y optimización para abordar estos desafíos.

El algoritmo CP-SAT de Google OR-Tools y el propio Google OR-Tools son alternativas útiles para modelar dificultades relacionadas con la planificación, distribución y control de recursos en arquitecturas vRAN. Esto permite que se puedan simular situaciones complejas y tomar decisiones óptimas en lapsos reducidos [6].

2.3.1 Complejidad en la Asignación de Recursos

La asignación dinámica de recursos en entornos vRAN constituye uno de los retos más significativos debido a la variabilidad del tráfico y a la heterogeneidad de los requisitos de los servicios 5G. Categorías como enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC) y massive Machine-Type Communications (mMTC) presentan demandas muy distintas en términos de latencia, ancho de banda y confiabilidad, lo que obliga a los sistemas de orquestación a realizar una distribución precisa y adaptativa de recursos computacionales, de red y radio [13].

La orquestación, en este contexto, debe optimizar de manera constante los recursos que se les asignan a las vBBUs y a sus funciones relacionadas, con el objetivo de lograr un balance entre la estabilidad, la eficiencia energética y el rendimiento. En situaciones de alta densidad y en ubicaciones edge, donde la capacidad computacional es escasa, la complejidad se incrementa. En contextos como estos, una asignación no óptima puede provocar cuellos de botella y deterioro del servicio, en particular en aplicaciones que son sensibles a la latencia o que presentan un tráfico muy dinámico [33].

Asimismo, tener en cuenta limitaciones adicionales como la sincronización, la capacidad de procesamiento de la O-DU, las fluctuaciones del canal radio y las prioridades de servicio establecidas a través del "network slicing" es necesario para lograr una distribución de recursos óptima. Estas condiciones hacen que la dimensionalidad del problema aumente, convirtiéndolo en un reto de optimización combinatoria. En este tipo de problemas, no siempre es posible utilizar los métodos convencionales de búsqueda exhaustiva [9].

Los algoritmos de inteligencia artificial y aprendizaje automático, gracias a su habilidad para adaptarse automáticamente a los cambios en el tráfico, han demostrado ser efectivos para optimizar la repartición de recursos y reducir el costo computacional, según muestran las investigaciones más actuales.

Sin embargo, se destaca igualmente lo relevante que es emplear herramientas de optimización robustas, como Google OR-Tools, las cuales permiten definir y resolver problemas complicados mediante algoritmos como CP-SAT y modelos de programación por restricciones. Estos últimos proporcionan soluciones eficaces en tiempos breves [6].

2.3.2 Interoperabilidad Multivendor

La arquitectura abierta de la vRAN, y particularmente el enfoque propuesto por O-RAN, promueve la interoperabilidad entre componentes provenientes de distintos proveedores. Sin embargo, esta apertura introduce desafíos importantes en la integración de elementos heterogéneos, ya que la falta de estandarización completa en interfaces como E2 u Open Fronthaul puede generar incompatibilidades que dificultan la implementación de procesos de orquestación y aumentan la complejidad del sistema [13].

Para garantizar un rendimiento apropiado y consistente entre los distintos módulos de red, se necesitan validaciones exhaustivas al implementar vBBUs multivendor. Estos métodos de certificación y pruebas aumentan los gastos iniciales de implementación y pueden extender el tiempo que toma la puesta en marcha, sobre todo en ambientes donde las funciones virtualizadas tienen que operar bajo exigencias rigurosas de latencia y fiabilidad [33].

Aunque el RIC (RAN Intelligent Controller) y los entornos de prueba respaldados por la O-RAN Alliance ayudan a reducir estas dificultades al verificar la compatibilidad y estandarizar

las interfaces, siguen existiendo retos en situaciones reales. Es imprescindible disponer de herramientas avanzadas para la orquestación y el monitoreo a fin de garantizar un rendimiento estable y eficiente en situaciones de operación dinámica, dado que el tráfico es variable, hay varios proveedores al mismo tiempo y se depende de conexiones con baja latencia [9]

A pesar de que el RIC (RAN Intelligent Controller) y los entornos de pruebas promovidos por la O-RAN Alliance contribuyen a disminuir estos problemas al homologar las interfaces y comprobar la compatibilidad, persisten desafíos en circunstancias reales. Para asegurar un desempeño eficaz y estable en condiciones de operación dinámica, es necesario contar con herramientas sofisticadas para la orquestación y el monitoreo debido a la variabilidad del tráfico, la coexistencia de varios proveedores y la dependencia de conexiones con baja latencia

2.3.3 Gestión de Fallos y Resiliencia

La gestión de fallos constituye un componente esencial en la orquestación de la Red de Acceso Radio Virtualizada (vRAN), debido a que las vBBUs operan en entornos distribuidos donde una falla localizada puede propagarse y afectar el rendimiento global del sistema. La virtualización de estas funciones requiere sistemas sofisticados de supervisión que sean capaces de identificar anomalías en tiempo real y llevar a cabo medidas correctivas para asegurar la continuidad del servicio [13].

La resiliencia se hace realidad en arquitecturas O-RAN y vRAN a través de la reconfiguración dinámica de funciones, esto es, el reparto de cargas laborales cuando hay problemas con los servidores localizados en el borde o con interrupciones en vínculos esenciales como el fronthaul. Para lograr esto, la orquestación tiene que armonizar procesos de migración de

funciones virtualizadas, redistribución de recursos y priorización de servicios para prevenir deterioros en indicadores como estabilidad del enlace, latencia o throughput [9].

Numerosos estudios proponen algoritmos de recuperación que permiten la ejecución de reasignaciones efectivas en escenarios de falla, lo que reduce el impacto energético incrementa la estabilidad del sistema. Para reconocer configuraciones alternativas que preserven el funcionamiento de la red, incluso frente a situaciones adversas, estas soluciones normalmente utilizan inteligencia artificial y métodos de optimización combinatoria. Su integración con instrumentos como Google OR-Tools hace más fácil la valoración de estrategias resilientes a través de simulaciones que imitan fallas reales en la infraestructura [34].

2.3.4 Latencia y Restricciones del Fronthaul en vRAN/O-RAN

El enlace de fronthaul constituye uno de los elementos más sensibles en la arquitectura vRAN y O-RAN, ya que la comunicación entre las unidades de banda base virtualizadas (vBBUs) y las unidades de radio (RUs) depende de requisitos estrictos de latencia y ancho de banda. Estos requisitos se vuelven aún más exigentes cuando se implementan *functional splits* como la opción 7-2x definida por 3GPP, la cual distribuye tareas críticas de la capa física entre la unidad distribuida (O-DU) y la unidad de radio (O-RU), aumentando la sensibilidad a retrasos e inestabilidades en la conexión [2].

En entornos no ideales —como enlaces basados en Ethernet frente a conexiones ópticas dedicadas— la latencia puede superar los límites establecidos para servicios de ultra baja latencia, especialmente en aplicaciones URLLC, lo que compromete la fiabilidad de la transmisión. Además, variaciones en el *jitter*, en la sincronización de paquetes o en la calidad del enlace pueden afectar la ejecución correcta de funciones asociadas al procesamiento en tiempo casi real[15].

Diversos estudios proponen técnicas de compresión y optimización del tráfico del fronthaul para mitigar estas limitaciones, reduciendo el ancho de banda requerido y mejorando la estabilidad del enlace. Estas técnicas permiten flexibilizar los requerimientos de infraestructura al minimizar la necesidad de enlaces ópticos exclusivos y al adaptar la transmisión a condiciones variables del canal [34].

La orquestación desempeña un papel clave en la mitigación de estas restricciones, ya que permite ubicar las vBBUs en puntos óptimos de la red —ya sea en la nube central o en nodos *edge*— con el fin de minimizar la latencia y garantizar el cumplimiento de los requisitos del servicio. Herramientas de optimización como Google OR-Tools facilitan la modelación computacional de estos escenarios, permitiendo evaluar configuraciones posibles y seleccionar ubicaciones que reduzcan el impacto de las limitaciones del fronthaul en el rendimiento global del sistema [6].

2.3.5 Eficiencia Energética en la Orquestación de vBBUs

La eficiencia energética se ha convertido en un desafío clave dentro de la orquestación de la Red de Acceso Radio Virtualizada (vRAN), debido al alto consumo asociado a la ejecución de funciones virtualizadas en entornos *cloud* y *edge*. Las vBBUs requieren recursos computacionales intensivos para procesar tareas de modulación, codificación, planificación de recursos radio y control del enlace, por lo que su operación continua puede incrementar significativamente el gasto energético de la infraestructura. En este contexto, la orquestación debe incorporar mecanismos inteligentes capaces de minimizar dicho consumo mediante acciones de optimización adaptativa [13].

Uno de los enfoques más adoptados consiste en realizar migraciones dinámicas de vBBUs entre nodos *fog*, *edge* y centros de datos *cloud* según el comportamiento del tráfico.

Es factible consolidar las funciones virtualizadas en un número reducido de servidores y desconectar los equipos que no están en uso cuando la demanda baja, lo que disminuye el consumo de energía sin perjudicar la calidad del servicio. Este tipo de políticas ha demostrado ser particularmente efectiva en las redes metropolitanas 5G, donde la variabilidad temporal del tráfico posibilita el aprovechamiento de oportunidades para ahorrar [8].

Además, las técnicas de control de actividad posibilitan habilitar o inhabilitar funciones particulares de las vBBUs en función del estado de la red, disminuyendo así el empleo innecesario de recursos computacionales. Dado que una consolidación excesiva podría aumentar la latencia en el fronthaul o crear cuellos de botella, estos métodos necesitan modelos de optimización que sean capaces de equilibrar el rendimiento, la latencia y el consumo energético. Según la literatura más reciente, herramientas como Google OR-Tools pueden modelar situaciones de minimización energética mediante su algoritmo CP-SAT. Este algoritmo tiene en cuenta limitaciones de capacidad, sincronización y distribución de carga, lo cual simplifica la evaluación de las políticas energéticamente eficientes en contextos vRAN y O-RAN [6].

2.3.6 Consumo energético RAN Tradicional

En arquitecturas de acceso radio tradicionales, las unidades de banda base físicas (BBU) operan como equipos dedicados que permanecen activos de forma continua, independientemente del nivel real de tráfico cursado por la red. Diversos estudios técnicos reportan que el consumo energético de una BBU física depende de su configuración, capacidad y escenario de despliegue. En entornos de baja densidad, celdas rurales o estaciones de menor capacidad, el consumo típico

de una BBU se sitúa en el rango de 150 W a 300 W, mientras que en despliegues **macro urbanos** este valor puede superar los **500 W** [37].

Investigaciones publicadas por IEEE y análisis de la GSMA indican que el procesamiento baseband representa una fracción significativa del consumo energético total de una estación base, debido al uso de hardware dedicado, sistemas de refrigeración y fuentes de alimentación redundantes. En consecuencia, el uso de valores promedio cercanos a 250 W por BBU física resulta consistente para estudios comparativos orientados a escenarios de capacidad media o baja, como los utilizados en modelos de evaluación energética.

Este consumo energético fijo, asociado a la operación continua del hardware dedicado, limita la eficiencia energética de la RAN tradicional y motiva la adopción de arquitecturas virtualizadas que permitan ajustar dinámicamente los recursos de procesamiento según la demanda de tráfico[38]

El consumo de un servidor físico moderno consume alrededor de 120 W a 160 W en modo reposo sin ninguna carga. El consumo típico bajo una carga moderada esta entre los 150 W a 250 W. y cuando el hardware trabaja al 100% su consumo máximo del servidor esta alrededor de 300 W a 500 W por servidor.[39], [40]

2.3.7 Seguridad en Arquitecturas Abiertas

La adopción de arquitecturas abiertas en vRAN y O-RAN introduce nuevos desafíos de seguridad debido al uso de interfaces estandarizadas como E2, A1 y O1, las cuales permiten la interoperabilidad entre equipos de distintos proveedores. Si bien estas interfaces habilitan flexibilidad y control inteligente también amplían la superficie de ataque y exponen la red a riesgos

como accesos no autorizados alteración de datos suplantación de entidades o ataques de denegación de servicio que pueden comprometer la estabilidad del sistema [9].

Por otra parte la naturaleza distribuida de las vBBUs ubicadas en nodos edge fog o en centros de datos incrementa la complejidad de asegurar todos los puntos de comunicación especialmente en el fronthaul y midhaul donde cualquier vulnerabilidad puede afectar funciones críticas de la capa física o del enlace. Adicionalmente, la incorporación de aplicaciones xApps y rApps en el RAN Intelligent Controller (RIC) demanda mecanismos rigurosos de autenticación, validación de políticas y supervisión, debido a que estas aplicaciones pueden modificar parámetros sensibles en tiempo casi real [13].

En este contexto, diversos estudios recomiendan la adopción del modelo Zero Trust Architecture (ZTA), el cual se basa en el principio de “nunca confiar, verificar siempre”. Este enfoque exige la autenticación continua de entidades, la segmentación de la red, el cifrado del tráfico y el monitoreo permanente para detectar comportamientos anómalos. La implementación efectiva de ZTA en entornos vRAN requiere mecanismos avanzados de orquestación capaces de supervisar flujos, validar configuraciones y reaccionar ante amenazas en tiempo real mediante acciones de mitigación y reconfiguración dinámica [34].

2.4 Algoritmos para la Orquestación y Asignación de Recursos

La orquestación y la asignación eficiente de recursos en la Red de Acceso Radio Virtualizada vRAN son elementos fundamentales para garantizar rendimiento escalabilidad y eficiencia energética especialmente en el funcionamiento de las unidades de banda base virtualizadas vBBUs, estos procesos requieren algoritmos capaces de gestionar de forma dinámica

recursos computacionales de red y radio y de responder a variaciones en la demanda de tráfico propias de redes 5G y de futuras arquitecturas 6G [13].

Desde un punto de vista técnico, la distribución ideal de recursos implica desafíos complejos relacionados con las limitaciones en términos de latencia, capacidad computacional, consumo de energía y prioridades del servicio. Estas condiciones convierten la orquestación en un reto de optimización combinatoria, que requiere que los algoritmos logren equilibrar a la vez una serie de objetivos, como el reparto de carga, la minimización de retrasos, asegurar fiabilidad y prevenir la saturación de los nodos cloud y edge [34].

De acuerdo con la bibliografía más actualizada, estos temas se tratan frecuentemente a través de técnicas de inteligencia artificial, como el aprendizaje automático (ML) y el aprendizaje por refuerzo profundo (DRL), así como también mediante métodos de programación matemática y optimización por restricciones. Estas metodologías permiten tomar decisiones efectivas y adaptables en situaciones cambiantes, identificando patrones de tráfico y alterando la distribución de recursos de forma predictiva [9].

En esta situación, instrumentos tales como Google OR-Tools y su algoritmo de programación por restricciones CP-SAT se han distinguido como opciones efectivas para la modelación y solución de problemas de orquestación. Su aptitud para gestionar extensas áreas de búsqueda, diversas limitaciones y metas simultáneas los vuelve una opción sólida para analizar tácticas de distribución de recursos en contextos O-RAN y vRAN, particularmente en actividades vinculadas a la localización de vBBUs, el control de actividad y la optimización energética [6].

2.4.1 Optimización Matemática y Programación Lineal

La optimización matemática constituye una herramienta fundamental para abordar los problemas de orquestación y asignación de recursos en la Red de Acceso Radio Virtualizada (vRAN). Entre las técnicas más utilizadas destacan la programación lineal entera (Integer Linear Programming, ILP) y la programación por restricciones (Constraint Programming, CP), las cuales permiten formular de manera estructurada los problemas asociados al uso y ubicación de las unidades de banda base virtualizadas (vBBUs) [13].

Desde estos puntos de vista, las cuestiones de orquestación se modelan mediante funciones objetivo que buscan disminuir al mínimo las métricas significativas, como la latencia, el uso del ancho de banda o el consumo energético. La disponibilidad de recursos en entornos edge y cloud, las limitaciones del fronthaul, los requerimientos del servicio (mMTC, URLLC o eMBB) y la capacidad computacional existente son algunos de los factores que restringen estas funciones.

De esta manera, ILP y CP permiten representar el sistema de forma matemática y obtener soluciones óptimas o cuasióptimas bajo condiciones específicas de operación [34].

La literatura reciente muestra aplicaciones exitosas de modelos ILP en escenarios de Cloud-Fog RAN, donde se optimiza la migración y distribución de vBBUs entre nodos *fog* y centros de datos *cloud*. Estos trabajos reportan reducciones energéticas de hasta un 42 % gracias a la consolidación dinámica de funciones virtualizadas en función de la variabilidad del tráfico, lo que evidencia la utilidad de los métodos de optimización para mejorar la eficiencia operativa del sistema [21].

No obstante, si se aplican a redes de gran tamaño, los modelos ILP pueden ser muy costosos en términos computacionales debido a la complejidad combinatoria que con frecuencia presentan

los problemas de orquestación en vRAN. Un ejemplo de una táctica híbrida que fusiona la programación por restricciones con la satisfacibilidad booleana es el algoritmo CP-SAT, que está disponible en Google OR-Tools. Este tipo de restricciones ha favorecido la implementación de tales perspectivas. Este algoritmo proporciona un rendimiento mejor en problemas con varias restricciones y es particularmente apropiado para situaciones de asignación de recursos que cambian, como las que se analizan en esta tesis [6].

2.4.2 Algoritmos Basados en Aprendizaje Automático

El aprendizaje automático (Machine Learning, ML) se ha consolidado como una herramienta clave para mejorar la orquestación y la asignación de recursos en arquitecturas vRAN y O-RAN, debido a su capacidad para modelar relaciones complejas entre la demanda de tráfico, la disponibilidad de recursos computacionales y el rendimiento de la red. Se pueden prever cambios temporales en el tráfico y modificar dinámicamente los parámetros operativos de las unidades de banda base virtualizadas (vBBUs) [13], con métodos como el aprendizaje no supervisado, el supervisado y el profundo (Deep Learning, DL).

La integración de ML en O-RAN se lleva a cabo, sobre todo, a través de aplicaciones rApps y xApps que están integradas en el RIC (RAN Intelligent Controller). Las xApps, que han sido creadas para funcionar en el Near-Real-Time RIC, utilizan un control de bucle cerrado con el que se puede optimizar la planificación radio; manejar interferencias; distribuir recursos y aumentar el throughput a pesar de las condiciones cambiantes de carga. Por su parte, las rApps trabajan en el Non-Real-Time RIC, ejecutando tareas de optimización de largo plazo basadas en análisis predictivo y políticas globales de la red [9][41].

Los métodos de aprendizaje por refuerzo profundo (Deep Reinforcement Learning, DRL) han mostrado ser particularmente efectivos en ambientes con alta variabilidad al tomar decisiones en tiempo real a través de la interacción directa con el entorno. Estos algoritmos adquieren estrategias ideales para la distribución de recursos sin necesidad de contar con modelos explícitos de la red, lo cual es especialmente beneficioso en situaciones en las que prever el comportamiento del tráfico es complicado [34].

Sin embargo, emplear algoritmos de ML presenta retos significativos. Su rendimiento está condicionado por la posibilidad de acceder a grandes cantidades de datos para formar modelos exactos. Esto puede ser problemático en contextos con recursos escasos o en implementaciones iniciales donde no hay un historial adecuado. Asimismo, la implementación de estos modelos requiere un alto nivel de capacidades computacionales, lo cual demanda una orquestación eficaz para prevenir la saturación en los nodos edge y asegurar latencias que se ajusten a las necesidades del servicio [21][42].

2.4.3 Aprendizaje por Refuerzo Profundo (DRL)

El aprendizaje por refuerzo profundo (Deep Reinforcement Learning, DRL) se ha convertido en una de las técnicas más prometedoras para la orquestación dinámica de recursos en arquitecturas vRAN y O-RAN. Este enfoque combina redes neuronales profundas con los principios del aprendizaje por refuerzo, permitiendo que un agente aprenda políticas óptimas mediante la interacción directa con el entorno de red. A través de este proceso, el agente observa el estado del sistema, ejecuta acciones y recibe recompensas que orientan su comportamiento hacia configuraciones más eficientes [13].

La capacidad del DRL para adaptarse a la incertidumbre lo convierte en un candidato adecuado para escenarios donde los patrones de tráfico presentan alta variabilidad o donde la aparición de fallos afecta el rendimiento de la red. En este tipo de entornos, los modelos tradicionales de optimización pueden resultar insuficientes debido a la complejidad combinatoria y a los cambios abruptos en las condiciones de operación, mientras que los agentes DRL pueden aprender estrategias de control que ajustan de forma dinámica la asignación de recursos computacionales, de red y radio [34].

La arquitectura vRAN es una ilustración sobresaliente de esta perspectiva; incorpora autoencoders para lograr representaciones compactas del estado del sistema, que luego son empleadas por un agente DRL responsable de decidir la orquestación en tiempo casi real. Este procedimiento ha mostrado mejoras de hasta un 25% en el rendimiento en situaciones de subaprovisionamiento, lo que demuestra la efectividad del DRL para administrar recursos en contextos muy cambiantes de vRAN [21].

2.4.4 Algoritmos de Aprendizaje de Bandidos

Los algoritmos de aprendizaje de bandidos, también conocidos como *Multi-Armed Bandit* (MAB), se han posicionado como una alternativa ligera y eficiente para la orquestación dinámica de recursos en arquitecturas vRAN y O-RAN. A diferencia de los enfoques de aprendizaje por refuerzo profundo (DRL), los algoritmos MAB no requieren modelos complejos ni grandes volúmenes de datos, lo que los hace adecuados para escenarios con recursos computacionales limitados. Su base fundamental es equilibrar la explotación y la exploración para determinar las mejores acciones en situaciones inciertas [13].

Exp3 es uno de los algoritmos más frecuentes en esta familia y ha mostrado ser efectivo para administrar recursos y ajustar las vBBUs de manera adaptable en circunstancias cambiantes.

Exp3 escoge acciones fundamentándose en recompensas probabilísticas, lo que posibilita una rápida reacción frente a alteraciones drásticas en los patrones de tráfico, errores o variaciones en la carga computacional.

Para la orquestación en tiempo casi real, donde el ritmo de toma de decisiones es vital, este método es especialmente eficaz [34].

Investigaciones actuales demuestran que Exp3 tiene el potencial de superar a las técnicas DRL en situaciones donde la capacidad computacional es escasa. Por ejemplo, se implementó Exp3 para programar vBBUs de manera eficiente desde el punto de vista energético en un entorno O-RAN. Esto permitió ahorrar hasta un 74.3 % del consumo de energía mediante la redistribución inteligente de cargas durante picos de tráfico o fallos. Estas conclusiones demuestran que los algoritmos de bandidos pueden optimizar la eficiencia operacional de la red en situaciones muy variables [21].

Igualmente, es posible combinar modelos basados en MAB con herramientas de optimización combinatoria, como Google OR-Tools, lo cual permite la unión de decisiones rápidas para elegir acciones con modelos más fuertes de planificación. Esta complementariedad posibilita el desarrollo de estructuras híbridas de orquestación, en las que los algoritmos de bandidos operan como sistemas para una respuesta rápida y los modelos matemáticos promueven el empleo global de recursos [6].

2.4.5 Comparación de Enfoques para Orquestación de vBBUs

La selección de un algoritmo para la orquestación de vBBUs requiere considerar criterios como complejidad computacional, escalabilidad, adaptabilidad y capacidad de operar bajo restricciones estrictas de latencia. Los modelos basados en programación matemática, como la programación lineal entera (ILP) y la programación por restricciones (CP), ofrecen un alto nivel de precisión al generar soluciones óptimas bajo un conjunto definido de restricciones. Sin embargo, su costo computacional crece de manera significativa a medida que se agranda el problema, lo cual complica su uso en situaciones que requieren orquestación en tiempo real o decisiones rápidas en contextos muy cambiantes [13].

En cambio, los métodos fundamentados en el aprendizaje automático (ML) y en el aprendizaje por refuerzo profundo (DRL) ofrecen una gran capacidad para ajustarse a cambios en el tráfico y a situaciones no determinísticas del medio ambiente. Estos algoritmos tienen la capacidad de reaccionar a cambios bruscos sin necesidad de redefinir modelos matemáticos complejos, ya que son capaces de aprender patrones conductuales. No obstante, su rendimiento está condicionado por la existencia de grandes cantidades de datos, procesos de entrenamiento extensos y un consumo significativo a nivel computacional. Esto puede ser problemático en nodos edge que cuentan con recursos limitados [34].

Los algoritmos de aprendizaje de bandidos, tal como Exp3, constituyen una opción intermedia que logra un balance entre la sencillez computacional y la habilidad de adaptarse. Estos métodos son apropiados para ambientes edge, donde la velocidad de decisión y el escaso consumo computacional son críticos. Funcionan con menos datos, se acercan rápidamente a una solución y son particularmente aptos para esos contextos. Los algoritmos de bandidos presentan la capacidad de conseguir avances significativos en términos de rendimiento y eficiencia energética, sin incurrir

en la complejidad computacional que está vinculada con los métodos DRL, según estudios recientes [21]. La capacidad de combinar estas perspectivas con herramientas de optimización como Google OR-Tools aumenta su potencial para complementar estrategias de orquestación fundamentadas en modelos híbridos [6].

2.5 Google OR-Tools en Optimización de Redes

Google OR-Tools es un conjunto de herramientas de optimización de código abierto desarrollado por Google, orientado a resolver problemas complejos en dominios como optimización combinatoria, programación lineal, programación entera y programación por restricciones. Su estructura modular posibilita la formulación de problemas a través de funciones objetivo, variables y limitaciones, proporcionando soluciones especializadas para contextos que demandan soluciones óptimas o casi óptimas bajo varias restricciones [6].

OR-Tools ofrece la posibilidad de modelar tareas como la gestión de fallos, la planificación de cargas, la asignación dinámica de recursos y la optimización energética para unidades virtualizadas de banda base (vBBUs) en el marco de redes virtualizadas de acceso radio (vRAN).

La flexibilidad de esta herramienta permite representar matemáticamente restricciones propias de la arquitectura vRAN y O-RAN, tales como límites de capacidad computacional, restricciones de latencia en el fronthaul, disponibilidad de nodos *edge* y requerimientos de servicio definidos mediante *network slicing* [13].

El algoritmo CP-SAT (Constraint Programming – SATisfiability) es uno de los solvers que se pueden utilizar en OR-Tools. Es un solucionador híbrido que fusiona técnicas de búsqueda avanzadas, satisfacibilidad booleana y programación por restricciones.

Dado que CP-SAT es efectivo para solucionar problemas de gran envergadura con numerosas limitaciones, resulta un instrumento adecuado para la orquestación de vBBUs en escenarios cambiantes donde los métodos tradicionales (como ILP) tienen costos computacionales altos [34].

Asimismo, la interoperabilidad de OR-Tools con Python facilita su utilización en situaciones simuladas y experimentales, lo cual posibilita el estudio de diversas estrategias de orquestación mediante la generación de escenarios que pueden ser replicados.

Esta característica convierte a OR-Tools en un recurso adecuado para investigaciones orientadas a optimizar recursos en arquitecturas vRAN mediante técnicas basadas en programación matemática y modelos de decisión [21].

2.5.1 Aplicaciones en Optimización de Redes

Google OR-Tools ha sido ampliamente utilizado en la optimización de redes debido a su capacidad para resolver problemas de enrutamiento, asignación de recursos y *scheduling* bajo restricciones complejas. En áreas como redes de sensores inalámbricos, la herramienta ha sido aplicada para maximizar la vida útil de la red mediante la formulación de modelos de programación lineal entera (ILP), logrando una asignación eficiente de energía y recursos distribuidos[43].

A pesar de que las aplicaciones específicamente documentadas para arquitecturas vRAN aún son escasas, los principios de optimización empleados en otros ámbitos pueden aplicarse aquí. Para modelar las restricciones de capacidad, latencia y ancho de banda en redes móviles, que son análogas a los retos de orquestación encontrados en la gestión de vBBUs, se ha hecho uso de OR-Tools. Además, estas restricciones concuerdan con los retos del fronthaul en las soluciones O-

RAN, dado que para cubrir las necesidades de servicio es esencial la asignación apropiada de recursos [13].

En este contexto, los métodos de optimización que se utilizan en OR-Tools son pertinentes para este estudio porque posibilitan la creación de modelos de simulación capaces de mostrar el control de actividad de las vBBUs, la provisión de recursos y la asignación dinámica en condiciones cambiantes. Su flexibilidad para integrar restricciones de capacidad, retardo y energía refuerza su utilidad como herramienta para evaluar estrategias de orquestación en entornos virtualizados de acceso radio [44].

2.5.2 Ventajas y Limitaciones

Google OR-Tools presenta múltiples ventajas para la resolución de problemas de optimización en el ámbito de redes. Entre ellas destacan su flexibilidad para modelar problemas complejos, el soporte integrado para diversos solvers —incluyendo programación lineal, entera y por restricciones— y su compatibilidad con Python, lo que facilita el desarrollo rápido de prototipos y la validación experimental de modelos de optimización (Google, 2025). Como es una herramienta de código abierto, brinda también ventajas como la transparencia en su operación, la posibilidad de personalizar más y la eliminación de costos asociados a licencias para ajustarse a diferentes necesidades de investigación [13].

Sin embargo, OR-Tools también tiene sus restricciones. Su rendimiento puede ser menor que el de los solvers comerciales más eficientes, como CPLEX o Gurobi, sobre todo en problemas a gran escala con miles de restricciones y variables.

Asimismo, la herramienta no incorpora de forma nativa modelos basados en aprendizaje automático o aprendizaje por refuerzo, por lo que la integración de enfoques ML/DRL requiere complementar OR-Tools con bibliotecas externas, aumentando la complejidad del sistema [34].

A pesar de estas limitaciones, OR-Tools resulta adecuado para el objetivo de esta investigación, ya que permite formular modelos de orquestación para vBBUs bajo restricciones de capacidad, latencia y consumo energético. Su capacidad para integrar el solver CP-SAT facilita la representación de problemas combinatorios propios de la arquitectura vRAN, manteniendo un balance adecuado entre eficiencia computacional y precisión en la solución[45].

2.5.3 Integración con Python para Simulaciones

La integración de Google OR-Tools con Python facilita la construcción de entornos de simulación para evaluar estrategias de orquestación en arquitecturas vRAN. Python es ampliamente utilizado en investigación científica debido a su sintaxis simple, su soporte para programación modular y la disponibilidad de bibliotecas especializadas para el tratamiento de datos y visualización de resultados. Esta compatibilidad posibilita la creación de modelos reproducibles que simulan situaciones dinámicas de distribución de recursos, teniendo en cuenta diversas restricciones [13].

En este escenario, se pueden complementar las simulaciones con bibliotecas como Matplotlib y NumPy. La primera se usa para representar gráficamente medidas de rendimiento como la latencia, el uso de recursos informáticos o el consumo energético; la segunda es útil para procesar y manejar datos. La combinación de estas herramientas permite que los modelos desarrollados en Python ejecuten procesos de orquestación y analicen el comportamiento de las vBBUs en distintos niveles de carga, reflejando condiciones cercanas a entornos reales [33]

Gracias a esta integración, OR-Tools ofrece una plataforma flexible que permite evaluar diferentes enfoques de optimización y medir su impacto en la operación de redes vRAN. Esto constituye un punto clave para la presente investigación, dado que las simulaciones desarrolladas en el siguiente capítulo se apoyarán en esta infraestructura para analizar el desempeño del algoritmo seleccionado [6].

CAPITULO III

3.1 Enfoque Metodológico de la Investigación

La investigación se desarrolla mediante un enfoque cuantitativo y experimental, sustentado en la construcción y evaluación de un modelo de simulación orientado a analizar el comportamiento de un algoritmo de orquestación en una Red de Acceso Radio Virtualizada (vRAN). Este enfoque permite estudiar la asignación de vBBUs bajo diferentes condiciones operativas, tomando como referencia métricas de rendimiento esenciales para la gestión de recursos en redes móviles.

El proceso metodológico se organiza en una secuencia de etapas que permiten avanzar desde la comprensión del problema hasta la validación del algoritmo implementado:

Fase 1: Identificación del problema y fundamentos teóricos

En esta primera etapa se examinan los desafíos que enfrentan las vBBUs en entornos virtualizados, especialmente aquellos relacionados con el uso de recursos, consumo de energía y la carga dinámica. Este análisis inicial constituye el marco conceptual sobre el cual se formula el modelo de optimización.

Fase 2: Selección del método de orquestación

Posteriormente se lleva a cabo una revisión comparativa de los solvers disponibles en Google OR-Tools, evaluando su capacidad para manejar restricciones complejas y problemas combinatorios. Con base en estos criterios técnicos se determina que el solver **CP-SAT** es el más adecuado, debido a su eficiencia en escenarios que requieren decisiones discretas y relaciones de dependencia entre múltiples variables.

Fase 3: Construcción del modelo de simulación

En esta fase se elabora un modelo matemático que representa la relación entre vBBUs y servidores físicos considerando restricciones de CPU y consumo de energía. La formulación incorpora variables binarias y parámetros que reflejan las condiciones reales de operación de una vRAN. Para el procesamiento de datos y la generación de matrices se emplea la biblioteca **NumPy**, mientras que todo el entorno de ejecución se implementa en Python.

Fase 4: Ejecución de pruebas y análisis de resultados

El modelo desarrollado se somete a distintos escenarios de carga para evaluar cómo responde el algoritmo ante variaciones en la demanda. Durante estas pruebas se registran indicadores como utilización de recursos, número de servidores activos. El análisis de estas métricas permite valorar la eficiencia del algoritmo, su capacidad de adaptación y su impacto en la operación general de la red.

Este proceso metodológico garantiza un análisis objetivo, reproducible y coherente con los principios de la ingeniería de telecomunicaciones, permitiendo fundamentar de manera rigurosa los resultados obtenidos en la simulación.

3.1.1 Descripción general del método

El método aplicado en la investigación se orienta a la evaluación estructurada de un algoritmo de orquestación en un entorno de Red de Acceso Radio Virtualizada (vRAN). Este método combina la formulación matemática del problema, la simulación computacional y el análisis comparativo de los resultados obtenidos, el propósito principal es la interpretación del algoritmo al asignar vBBUs a los recursos físicos con restricciones de capacidad, consumo de energía y demanda variable.

El método se realiza mediante la creación de un modelo de simulación replicado, el mismo que cumple la función de mantener bajo control las condiciones a las que se encuentra una vRAN si se encuentra en situaciones de carga realistas, para ello se establecen ciertas limitaciones técnicas, de rendimiento, parámetros operativos y métricas de rendimiento para poder medir la eficacia del algoritmo, como paso siguiente también se llevan a cabo simulaciones de análisis para el modelo en distintos entornos de tráfico.

Este método proporciona la posibilidad de evaluar el rendimiento del algoritmo sin requerir una infraestructura física especializada, lo cual permite analizar su habilidad para adaptarse, su eficacia y su estabilidad bajo condiciones variables. De esta manera, se obtiene una visión exacta de cómo el algoritmo influye en la operación general del sistema y su posible aplicación en escenarios reales de telecomunicaciones.

3.1.2 Etapas del proceso de investigación

El proceso de investigación se desarrolla mediante un conjunto de etapas que permiten avanzar de manera ordenada desde la identificación del problema hasta la validación del algoritmo de orquestación. Cada etapa cumple una función específica dentro del método y contribuye a garantizar coherencia entre los objetivos planteados y los resultados obtenidos.

a) Análisis inicial del problema y contexto técnico

En la primera etapa se examinan los desafíos presentes en la orquestación de vBBUs dentro de una arquitectura vRAN, considerando aspectos como la demanda variable de tráfico, la capacidad computacional disponible y las restricciones impuestas por el fronthaul. Este análisis permite delimitar el alcance del estudio y definir los parámetros que deberán modelarse posteriormente.

b) Selección del enfoque de optimización

La segunda etapa consiste en la evaluación de diferentes técnicas de optimización y solvers disponibles en Google OR-Tools. Este análisis permite determinar cuál de ellos posee las características adecuadas para manejar restricciones combinatorias y proporcionar soluciones eficientes. Con base en este criterio se selecciona el solver CP-SAT, debido a su eficiencia en problemas discretos de asignación.

c) Elaboración del modelo matemático y configuración de los datos

En esta etapa se establece la formulación matemática del problema, definiendo las variables de decisión, restricciones de operación y función objetivo. Paralelamente, se construyen matrices y estructuras de datos que representan, capacidades de CPU, consumo de energía y características de los servidores físicos. Este conjunto de datos sirve como base para las simulaciones.

d) Implementación del modelo en Python y ejecución de simulaciones

Una vez definido el modelo matemático, se procede a implementarlo mediante scripts en Python utilizando las herramientas proporcionadas por Google OR-Tools. Posteriormente, se realizan simulaciones en distintos escenarios de carga, lo que permite observar cómo responde el algoritmo ante variaciones en la demanda y condiciones operativas diversas.

e) Evaluación de resultados y análisis comparativo

En la última etapa se analizan las salidas generadas por las simulaciones, considerando métricas como utilización de recursos, consumo de energía obtenida y número de servidores activos. El análisis comparativo entre escenarios permite identificar el grado de eficiencia del algoritmo, su capacidad de adaptación y su desempeño general en entornos vRAN.

Estas etapas, aplicadas de manera secuencial, permiten desarrollar un proceso metodológico coherente, verificable y adecuado para evaluar el funcionamiento del algoritmo en un entorno virtualizado de telecomunicaciones.

3.1.3 Criterios de validación del algoritmo

La validación del algoritmo se basa en un conjunto de criterios que permiten determinar su eficacia y su adecuación para resolver problemas de orquestación en una Red de Acceso Radio Virtualizada (vRAN). Estos criterios se establecen con el propósito de asegurar que los resultados obtenidos durante las simulaciones sean consistentes, fiables y comparables entre distintos escenarios operativos.

a) Coherencia con las restricciones del modelo

El primer criterio consiste en verificar que las soluciones entregadas por el algoritmo respetan las restricciones definidas en el modelo matemático. Esto incluye el cumplimiento de los límites de capacidad de CPU y el consumo de energía para cada vBBU. La validez de la solución depende de que todas estas condiciones se satisfagan simultáneamente.

b) Eficiencia en el uso de recursos

Otro criterio fundamental es la capacidad del algoritmo para utilizar los recursos físicos de manera eficiente. Se evalúa si la asignación de vBBUs minimiza el número de servidores activos y evita sobrecargas innecesarias. Este análisis permite determinar si la solución contribuye a reducir el consumo energético y mejorar la sostenibilidad operativa del sistema.

c) Estabilidad frente a variaciones de carga

El algoritmo debe demostrar un comportamiento estable frente a cambios en la demanda de tráfico. Para ello, se examina cómo se comporta la asignación de recursos en escenarios de baja, media y alta carga. Una solución estable mantendrá niveles adecuados de rendimiento aun cuando las condiciones operativas cambien de forma significativa.

e) Reproducibilidad de los resultados

Finalmente, se valora la capacidad del algoritmo para generar resultados reproducibles. Se verifica que, bajo las mismas condiciones iniciales, el modelo produzca soluciones equivalentes, lo que garantiza confiabilidad en el proceso de simulación y en la interpretación de los datos obtenidos.

Estos criterios permiten evaluar de manera integral el desempeño del algoritmo y determinan si su aplicación puede considerarse viable para entornos reales de orquestación en redes virtualizadas.

3.2 Diagrama de Flujo Metodológico

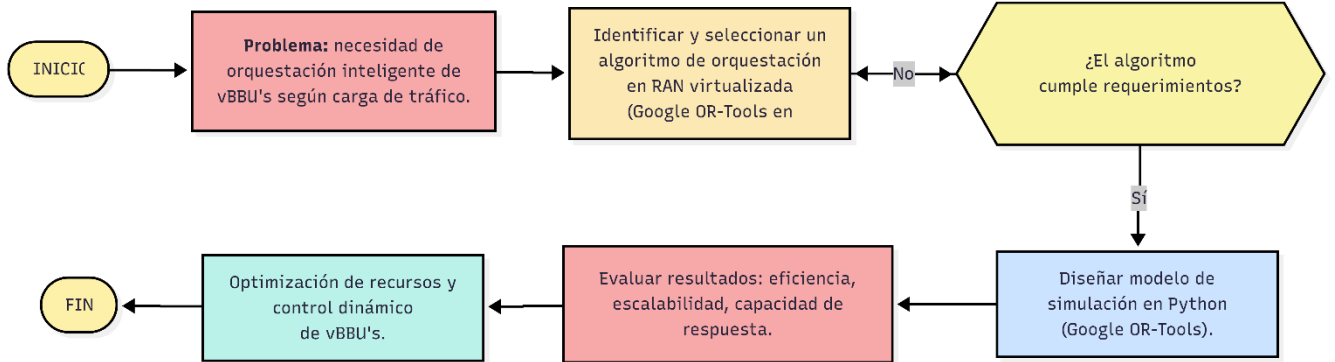


Figura 7 Diagrama de flujo metodológico para la evaluación del algoritmo de orquestación en vRAN.

El diagrama de flujo metodológico presenta de forma ordenada las etapas que constituyen al proceso de investigación y el orden lógico que va a seguir para la evaluación del algoritmo de orquestación en la arquitectura vRAN; este gráfico ayuda a visualizar cómo se vinculan las actividades principales, los puntos de decisión y las etapas necesarias para progresar en la validación del modelo.

El flujo comienza con la detección del problema de investigación, en el que se identifican las restricciones existentes en la orquestación de vBBUs y se determinan los objetivos que orientan el análisis. Luego, se elige el algoritmo de optimización que se evaluará, teniendo en cuenta su aptitud para producir soluciones eficaces y su compatibilidad con las limitaciones técnicas del modelo.

Luego de seleccionar el algoritmo, se procede a crear el modelo matemático y a definir los parámetros que simbolizan las condiciones operativas de la red. Este modelo es la base para que

el código en Python se implemente y para que las simulaciones definidas en diferentes contextos de carga se ejecuten.

Un punto de verificación está incluido en el diagrama, lo cual posibilita verificar si el algoritmo satisface los requisitos establecidos. Si la solución no cumple las condiciones del modelo, el procedimiento regresa a la etapa de selección y ajuste, asegurando así un ciclo iterativo dirigido hacia la mejora constante. El flujo pasa a la etapa de análisis y evaluación de resultados, en la que se analizan las métricas conseguidas y se determinan las conclusiones finales del estudio, si la solución es válida.

En la figura 5. Este diagrama proporciona una guía visual clara del proceso metodológico, facilita la comprensión del enfoque adoptado y asegura la coherencia entre cada una de las etapas que conforman la investigación.

3.2.1 Explicación del flujo de trabajo

El flujo de trabajo representado en el diagrama metodológico organiza de manera secuencial las actividades necesarias para evaluar el desempeño del algoritmo de orquestación en la vRAN. Cada parte del diagrama está destinada a una función concreta en el proceso de investigación y crea un vínculo directo entre las fases técnicas, analíticas y de verificación.

El procedimiento comienza al identificar el problema, en el que se establecen las restricciones asociadas con la asignación de vBBUs y la gestión de recursos. Este paso facilita la identificación de las necesidades operativas del sistema y la determinación del alcance del modelo de simulación.

La elección del algoritmo de optimización es la siguiente etapa, en la que se examinan las distintas opciones disponibles y se decide cuál es el método más apropiado para solucionar un

problema de asignación con numerosas restricciones. El algoritmo elegido se vuelve el eje central del modelo matemático.

Después, el flujo se dirige a la creación del modelo y la configuración de parámetros, en la que se determinan las restricciones técnicas, las variables de decisión y las condiciones bajo las cuales funcionará el sistema; en esta fase se integran los valores de demanda, capacidad y consumo de energía que caracterizan la operación de una vRAN.

El siguiente paso corresponde a la implementación del modelo en Python, lo que permite ejecutar el algoritmo mediante una estructura programada que procesa los distintos escenarios de carga definidos para la investigación.

El flujo incluye un punto de evaluación en el que se verifica si los resultados obtenidos cumplen las condiciones definidas por el modelo, si no se cumplen el procedimiento regresa a etapas previas para ajustar parámetros o reevaluar la selección del método de optimización, cuando el resultado es satisfactorio se continúa con la etapa de análisis donde se revisan las métricas obtenidas y se determina el desempeño del algoritmo en términos de estabilidad eficacia y uso de recursos.

Este flujo de trabajo garantiza un procedimiento que es coherente, verificable y organizado, lo que posibilita llegar a conclusiones apoyadas por un proceso metodológico robusto.

3.2.2 Iteración y retroalimentación del proceso

La metodología empleada incorpora un proceso iterativo que permite ajustar y perfeccionar el modelo de orquestación conforme avanzan las simulaciones. Esta retroalimentación continua es esencial para garantizar que el algoritmo seleccionado responda adecuadamente a las condiciones establecidas y produzca resultados consistentes con los objetivos de la investigación.

La iteración se inicia cuando los resultados de un desempeño no cumplen con alguna de las restricciones definidas en el modelo, ya sea porque el consumo de energía o la capacidad del procesador central (CPU) se sobrepasan.

En este contexto, el flujo metodológico vuelve a etapas anteriores, lo que posibilita que se revisen los parámetros, se redefinan los valores de entrada o se modifique la formulación del problema.

Este ciclo cíclico también posibilita analizar la sensibilidad del algoritmo ante los cambios en las condiciones de carga.

Al realizar simulaciones con distintos niveles de demanda, se puede determinar si la solución proporciona un comportamiento estable o si necesita cambios estructurales adicionales, el modelo se perfecciona directamente gracias a la retroalimentación que se obtiene de estas evaluaciones.

Además, la iteración ayuda a verificar la consistencia entre las ejecuciones. Si el algoritmo genera resultados que se pueden reproducir bajo las mismas condiciones iniciales, se fortalece la validez del método que se empleó. En caso contrario, el proceso metodológico define los procedimientos requeridos para investigar el origen de la inconsistencia y rectificarla.

En conjunto, este mecanismo de retroalimentación garantiza que la investigación no avance de forma lineal, sino bajo un enfoque adaptativo que permite corregir, mejorar y consolidar el modelo de orquestación hasta alcanzar resultados fiables, precisos y acordes con los requerimientos técnicos de la vRAN.

3.3 Herramientas de trabajo

La investigación requiere un conjunto de herramientas que permiten desarrollar el modelo de orquestación, ejecutar las simulaciones y analizar los resultados obtenidos. Estas herramientas se dividen en elementos de hardware y software, que son elegidos según su capacidad para soportar cargas computacionales, simplificar el desarrollo del algoritmo y asegurar la posibilidad de repetir el proceso. Respecto al hardware, se utiliza una computadora que ofrece la capacidad requerida para implementar los escenarios de simulación creados, garantizando tiempos de respuesta apropiados durante el proceso de resolución del modelo a través del solver CP-SAT.

Por otro lado el software empleado integra el entorno de programación con bibliotecas de soporte para realizar cálculos matemáticos representación de datos y optimización, la evaluación del algoritmo es posible gracias a la incorporación de estos elementos que constituyen la infraestructura esencial para su desarrollo y permiten que el proceso se ejecute de manera controlada coherente y eficaz, además esta selección de herramientas facilita que las simulaciones se reproduzcan en distintos contextos y mantengan la coherencia metodológica del estudio.

3.3.1 Hardware Utilizado

Para el desarrollo de la investigación se emplea un equipo de cómputo de propósito general que permite ejecutar las simulaciones del modelo de orquestación sin necesidad de infraestructura especializada. El hardware seleccionado ofrece la capacidad de procesamiento y memoria suficiente para manejar las cargas computacionales generadas por el solver CP-SAT y por las operaciones asociadas al manejo de matrices y estructuras de datos.

El equipo utilizado corresponde a una laptop de alto rendimiento, cuyas características principales son:

- **Procesador:** AMD Ryzen 5 7535HS, adecuado para tareas de cálculo intensivo y ejecución paralela.
- **Memoria RAM:** 16 GB DDR5, suficiente para gestionar múltiples escenarios de simulación y ejecutar modelos de optimización sin afectar la estabilidad del entorno.
- **Almacenamiento:** Unidad SSD de 512 GB, que facilita tiempos de lectura y escritura reducidos durante la ejecución de los scripts.
- **Tarjeta gráfica:** NVIDIA GeForce RTX 2050, que se utiliza sobre todo para acelerar procesos de visualización al crear gráficos que representen los resultados.
- **Plataforma de sistema operativo:** Windows 11 Home, versión de 64 bits.

Este hardware posibilita la ejecución de los modelos implementados en Python con tiempos de respuesta apropiados, aun en situaciones que incluyen una cantidad considerable de vBBUs y servidores virtuales. La selección de este equipo se debe a la necesidad de tener un ambiente portátil, estable y lo suficientemente potente para realizar las simulaciones necesarias en la investigación.

3.3.2 Herramientas en Software

El desarrollo del modelo de orquestación y la ejecución de las simulaciones dependen de un conjunto de herramientas software que permiten implementar el algoritmo, procesar los datos y analizar los resultados obtenidos. Estas herramientas fueron seleccionadas por su estabilidad, compatibilidad con métodos de optimización y su capacidad para manejar estructuras de datos de alta dimensionalidad.

El entorno de trabajo se basa en los siguientes componentes:

a) Lenguaje de programación: Python 3.10

Se utiliza Python en su versión 3.10 por su estabilidad, amplio soporte documental y compatibilidad con las bibliotecas empleadas en la investigación. Esta versión ofrece un entorno adecuado para la ejecución del solver CP-SAT, el manejo de matrices y la integración con las herramientas científicas utilizadas. Su arquitectura modular permite estructurar el modelo de forma clara y escalable.

b) Biblioteca de optimización: Google OR-Tools (versión 9.14.6206)

Google OR-Tools constituye la herramienta principal empleada para resolver el modelo de orquestación. En particular, se utiliza el solver **CP-SAT**, adecuado para problemas combinatorios con restricciones múltiples y variables binarias. La biblioteca proporciona estructuras eficientes para formular el modelo matemático y ejecutar los procesos de optimización con tiempos de resolución reducidos.

c) Librerías científicas complementarias

Para la manipulación y análisis de datos se utilizan las siguientes herramientas:

- **NumPy:** utilizada para la gestión de arreglos multidimensionales y operaciones matemáticas vectorizadas, esenciales para representar demandas, capacidades y matrices de latencia.
- **Matplotlib:** empleada para generar gráficos que permiten visualizar tendencias, comparar escenarios y representar los resultados obtenidos.
- **SciPy:** utilizada para cálculos adicionales que complementan el análisis estadístico o matemático cuando las simulaciones lo requieren.

d) Entorno de desarrollo: Visual Studio Code

Visual Studio Code se emplea como entorno de programación debido a su flexibilidad, compatibilidad con Python y soporte para extensiones orientadas al desarrollo científico. Este entorno permite ejecutar scripts, depurar código y gestionar dependencias de forma centralizada.

Estas herramientas software conforman una plataforma sólida y coherente para la ejecución del modelo de orquestación, garantizando un entorno estable, reproducible y adecuado para la evaluación del algoritmo en los distintos escenarios definidos en la investigación.

3.4 Diseño de la propuesta de orquestación

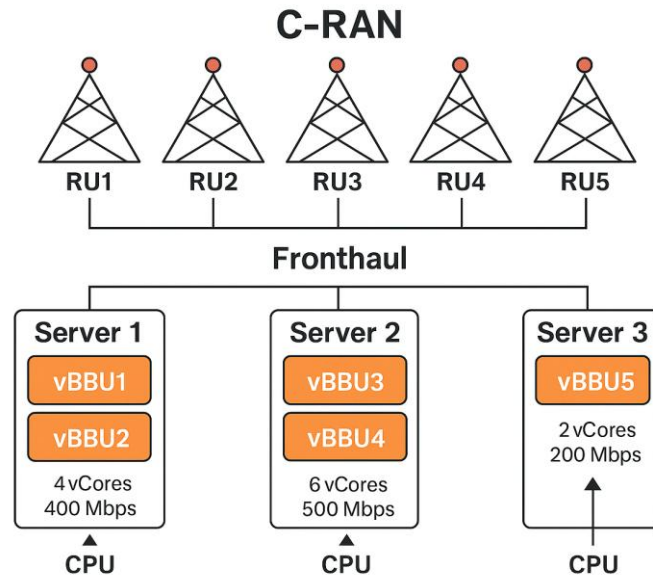


Figura 8- Arquitectura propuesta de orquestación

El diseño de la propuesta se orienta a la construcción de un modelo de orquestación capaz de asignar unidades de banda base virtualizadas (vBBUs) a servidores físicos bajo condiciones de capacidad, consumo de energía y demanda dinámica propias de una arquitectura vRAN. El propósito del diseño es establecer una estructura operativa que permita evaluar el desempeño del algoritmo seleccionado y determinar su eficiencia en la gestión de recursos.

El proceso de diseño inicia con la definición de los elementos que intervienen en la orquestación. Para ello, se identifican los conjuntos de vBBUs, servidores disponibles, parámetros de capacidad, requisitos de servicio. Estos elementos se organizan en forma de matrices y vectores que representan el comportamiento de la red en distintos escenarios.

Posteriormente, se formulan las restricciones que gobernarán el funcionamiento del modelo. Estas restricciones incluyen:

Límites de capacidad computacional por servidor.

Asignación única de cada vBBU a un servidor físico.

Activación o desactivación de servidores en función de la carga asignada.

A partir de estos elementos se define la función objetivo, la cual busca minimizar el número de servidores activos y optimizar la distribución de las vBBUs, de manera que el sistema mantenga un uso equilibrado de sus recursos.

El diseño incluye también la implementación de escenarios simulados para determinar el rendimiento del algoritmo bajo condiciones cambiantes con distintos niveles de carga lo que permite analizar la estabilidad del modelo y su capacidad de ajustarse a demandas bajas medias y altas, esta estructura permite aplicar el modelo matemático en Google OR-Tools y facilita de forma sistemática la evaluación de cómo el algoritmo de orquestación influye en la eficiencia operativa del vRAN.

3.4.1 Modelado conceptual de la orquestación

El modelado conceptual de la orquestación constituye la base sobre la cual se estructura la propuesta desarrollada en la investigación. Este modelo permite representar, de manera abstracta, la interacción entre las unidades de banda base virtualizadas (vBBUs), los recursos físicos disponibles y las restricciones operativas propias de una arquitectura vRAN. Su propósito es establecer un marco lógico que facilite la formulación matemática del problema y la posterior implementación del algoritmo de optimización.

El modelo conceptual identifica tres componentes principales:

- a) las vBBUs, que representan las unidades virtuales que deben ser asignadas;
- b) los servidores físicos o nodos de ejecución, que proporcionan la capacidad computacional y el ancho de banda; y

- c) las restricciones técnicas, que limitan la forma en que estas asignaciones pueden realizarse. Entre estas restricciones se incluyen la capacidad de procesamiento, consumo de energía y la necesidad de activar únicamente los servidores necesarios para atender la demanda.

Este modelo conceptual también establece una conexión entre las decisiones de orquestación y los parámetros que afectan su rendimiento. Cada vBBU se asigna como una variable que tiene interacción con las capacidades del servidor correspondiente, lo cual hace posible la representación de la carga de trabajo, el impacto de cada asignación sobre el sistema y cómo se distribuyen los recursos.

Por último, el modelo conceptual incluye la idea de escenarios dinámicos, en los cuales la demanda de tráfico tiene la capacidad de cambiar y crear nuevas condiciones operativas.

Este enfoque permite que el algoritmo sea evaluado en situaciones realistas, reflejando el comportamiento de una vRAN bajo diferentes niveles de carga y condiciones de servicio.

En conjunto, este modelado conceptual proporciona una estructura clara y coherente que guía el diseño del modelo matemático y orienta el proceso de simulación dentro de la propuesta de orquestación.

3.4.2 Parámetros considerados en la simulación

Los parámetros utilizados en la simulación se enfocan exclusivamente en los dos factores definidos para este estudio: **capacidad de CPU** y **consumo energético**. Estos parámetros permiten evaluar el desempeño del algoritmo bajo un modelo simplificado, pero representativo de la operación básica de una arquitectura vRAN.

Los parámetros considerados son:

a) Demanda de CPU por vBBU

Cada vBBU requiere una cantidad específica de capacidad de procesamiento para ejecutar sus funciones virtualizadas. Este valor se expresa en unidades de cómputo (vCPU) y constituye la carga que cada vBBU impone sobre el servidor al que se asigna.

b) Consumo energético por vBBU

Cada vBBU genera un costo energético asociado a su ejecución. Este parámetro permite evaluar el impacto total de la asignación sobre el consumo energético global del sistema.

c) Capacidad máxima de CPU por servidor

Los servidores cuentan con un límite de capacidad computacional. La suma de las demandas de CPU de las vBBUs asignadas no puede superar este valor, lo cual garantiza estabilidad operativa durante la simulación.

Estos parámetros conforman la base del modelo matemático y permiten representar de manera adecuada el problema de orquestación bajo el enfoque definido en esta investigación.

3.4.3 Restricciones del modelo

El modelo matemático incorpora un conjunto de restricciones fundamentales que garantizan que la asignación de las vBBUs sea técnicamente viable desde el punto de vista de la capacidad computacional y del consumo energético.

Las restricciones definidas son:

a) Restricción de asignación única

Cada vBBU debe ser asignada a un único servidor físico:

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I$$

b) Restricción de capacidad de CPU

La suma de las demandas de CPU de las vBBUs asignadas a un servidor no debe exceder su capacidad:

$$\sum_{i \in I} CPU_i x_{ij} \leq CPU_j^{max}, \forall j \in J$$

c) Restricción de activación del servidor

Un servidor solo se considera activo si aloja al menos una vBBU:

$$x_{ij} \leq y_j, \forall i \in I, \forall j \in J$$

d) Restricción de consistencia binaria

$$x_{ij}, y_j \in \{0,1\}$$

Estas restricciones permiten que el modelo represente adecuadamente el proceso de orquestación centrado en CPU y energía.

3.5 Selección del Algoritmo

La selección del algoritmo constituye una etapa fundamental dentro del diseño de la propuesta, ya que determina el enfoque de optimización empleado para resolver el problema de orquestación de vBBUs en la arquitectura vRAN. Esta elección se basa en un análisis comparativo de las técnicas disponibles y en la evaluación de su capacidad para manejar restricciones complejas, variables discretas y escenarios dinámicos de carga.

Para este propósito se consideran los solvers integrados en Google OR-Tools, los cuales ofrecen diferentes enfoques de resolución, tales como programación lineal, programación entera mixta y programación por restricciones. Entre estos métodos, se identifican sus características principales, niveles de eficiencia, escalabilidad y compatibilidad con el modelo matemático utilizado.

Tras realizar la evaluación, se determina que el solver **CP-SAT** es el más adecuado para la investigación. El manejo eficaz de variables binarias, restricciones combinatorias y funciones objetivo que persiguen optimizar la distribución de carga y minimizar la activación de servidores es posible gracias a su estructura híbrida, fundamentada en técnicas de satisfacibilidad booleana y programación por restricciones. Asimismo, su rendimiento es positivo incluso en situaciones en las que la dimensionalidad del problema aumenta significativamente.

El CP-SAT da la posibilidad de añadir restricciones complementarias con un alto grado de flexibilidad, lo que permite la integración en un mismo modelo de parámetros como latencia, ancho de banda del fronthaul y capacidad de procesamiento. Esta habilidad de adecuación posibilita que

las circunstancias operativas de la vRAN sean representadas con precisión y que se lleven a cabo simulaciones del proceso de orquestación que sean realistas.

Por lo tanto, el CP-SAT fue elegido como el algoritmo principal de la propuesta a causa de su eficacia, su aptitud para gestionar problemas complejos y discretos y su compatibilidad con el ambiente de simulación creado en Python. Su selección asegura que el modelo proporcione resultados coherentes y técnicamente correctos en los escenarios analizados.

3.5.1 Comparación entre solvers de Google OR-Tools

Para seleccionar el algoritmo de optimización más adecuado, se realiza una comparación entre los principales solvers disponibles en Google OR-Tools. Dado que cada solver tiene una perspectiva diferente para solucionar problemas matemáticos, es imprescindible examinar sus características operativas y su aptitud para adaptarse a las exigencias del modelo de orquestación propuesto en la investigación.

El examen considera aspectos como la eficacia computacional, la habilidad para manejar una diversidad de problemas, la capacidad para escalar en caso de que se incrementen las variables y la compatibilidad con restricciones complejas. Los solvers de programación por restricciones, programación lineal y programación entera mixta se consideran entre las alternativas evaluadas. Sin embargo, cuando intervienen variables discretas o enteras la capacidad del solver GLOP para resolver problemas lineales con rapidez disminuye, a pesar de que estos componentes son fundamentales para la distribución de vBBUs. Por su parte, los solvers de programación entera mixta SCIP y CBC pueden funcionar adecuadamente en modelos estructurados, aunque su tiempo de resolución aumenta de forma considerable cuando se incorporan numerosas restricciones combinatorias.

El solver GLPK es una opción de código abierto accesible pero presenta limitaciones relevantes en eficiencia computacional y escalabilidad porque su desempeño se degrada cuando aumenta el número de restricciones o variables por ello resulta menos adecuado para escenarios dinámicos donde la carga varía entre simulaciones, estas limitaciones reducen el rendimiento de los solvers evaluados en el modelo de orquestación propuesto mientras que CP-SAT ofrece mayor capacidad para resolver problemas combinatorios con múltiples restricciones y dependencias lo que respalda su elección como herramienta principal de la propuesta de optimización desarrollada en este estudio.

Por otro lado, el solver CP-SAT se distingue por su perspectiva híbrida fundamentada en métodos de programación para restricciones y de satisfactibilidad booleana, este método permite tratar de manera eficiente variables binarias y restricciones lógicas, lo que resulta especialmente útil para representar la asignación única de vBBUs, la activación dependiente de servidores y los límites de capacidad de CPU y fronthaul.

Asimismo, CP-SAT muestra un comportamiento constante aun si el tamaño del problema aumenta. La tabla 1, al ser comparada, determina que el solver más adecuado para el modelo de orquestación creado es CP-SAT. Esto se debe a su habilidad para solucionar problemas con una complejidad combinatoria elevada, a su eficacia en los tiempos de ejecución y a su versatilidad para incluir varias restricciones en un mismo esquema matemático.

Tabla 1-Comparación de solvers en Google OR-Tools para la orquestación en vRAN.

Solver	Tipo de Problema Principal	Eficiencia Computacional	Escalabilidad	Compatibilidad con Python	Idoneidad para Tesis (Razones)	Recomendado
CP-SAT	Programación de restricciones (CP), problemas combinatorios con	Alta (usa SAT para eficiencia en restricciones lógicas).	Buena para problemas medianos a grandes (hasta miles de variables).	Excelente (interfaz nativa en Python).	Ideal para scheduling y asignación dinámica en vBBUs; maneja restricciones complejas como	Sí (Mejor opción)

	variables booleanas/enteras.				BW \geq demanda y latencia \leq umbral. Rápido para simulaciones iterativas. Integra bien con NumPy para preprocesado de datos.	
SCIP	Programación lineal entera mixta (MIP/ILP), optimización matemática.	Media-alta (buena para problemas estructurados).	Alta para problemas grandes, pero más lenta en combinatorios puros.	Buena (a través de OR-Tools).	Adecuada para minimización de costos energéticos y aprovisionamiento; soporta variables continuas (e.g., BW). Compatible con NumPy para matrices de coeficientes. Menos eficiente que CP-SAT en restricciones booleanas.	No (Alternativa)
GLOP	Programación lineal pura (LP), problemas continuos.	Muy alta (rápida para LP lineales).	Excelente para problemas grandes sin variables enteras.	Excelente.	Útil para fases iniciales de aprovisionamiento continuo (e.g., flujos de red), pero no maneja variables enteras necesarias para asignación discreta de vBBUs. Limitado para la tesis.	No
CBC	Programación lineal entera mixta (MIP).	Media (open-source, robusta pero más lenta que comerciales).	Buena para problemas medianos.	Buena.	Similar a SCIP, pero menos optimizada para restricciones complejas. Adecuada como respaldo, pero CP-SAT es superior para combinatorios.	No
GLPK	LP y MIP.	Baja-media (más lenta en problemas grandes).	Limitada para escenarios de alta dimensionalidad.	Buena.	Básica para prototipos, pero no escalable para simulaciones detalladas de vRAN. No recomendada para optimización dinámica.	No

3.5.2 Justificación para la selección del CP-SAT

La selección del solver CP-SAT se fundamenta en su capacidad para resolver problemas de asignación y planificación que involucran múltiples restricciones, variables discretas y decisiones dependientes, características que corresponden directamente al comportamiento de la orquestación en una arquitectura vRAN.

A diferencia de otros solvers analizados CP-SAT combina la programación por restricciones con métodos de satisfacibilidad booleana y esta integración le permite explorar grandes espacios de búsqueda de forma eficiente para alcanzar soluciones óptimas con mayor rapidez ya que mantiene un desempeño sólido en modelos con variables binarias indispensables para representar la asignación individual de cada vBBU y la activación de servidores físicos además permite incorporar restricciones de capacidad de CPU y consumo energético sin afectar la estabilidad del proceso de optimización.

Otro aspecto relevante es la escalabilidad ya que el solver mantiene tiempos de resolución aceptables incluso cuando aumenta el número de vBBUs o de servidores lo cual es esencial para modelar escenarios realistas donde la demanda y la carga de tráfico pueden variar de manera significativa y en consecuencia se convierte en una herramienta adecuada para simular entornos más complejos y para extender la propuesta a configuraciones más detalladas.

Asimismo, CP-SAT ofrece la posibilidad de definir funciones objetivo con múltiples criterios como mejorar la distribución de recursos o reducir la cantidad de servidores en operación, con ello el modelo puede equilibrar eficiencia energética disponibilidad y rendimiento que son elementos esenciales en el entorno operativo de una vRAN.

Finalmente, se garantiza su facilidad de implementación, adaptabilidad y reproducibilidad porque es compatible con Python y se incorpora directamente a las estructuras de datos empleadas en el modelo.

Estas cualidades establecen al CP-SAT como la alternativa más apropiada para validar y desarrollar el algoritmo de orquestación que se propone en la investigación.

La tabla 1 permite determinar que CP-SAT es el solver más adecuado para el modelo de orquestación creado, ya que tiene la habilidad de resolver problemas con alta complejidad combinatoria, es eficiente en términos de tiempo de ejecución y tiene flexibilidad para incluir diversas restricciones dentro del mismo esquema matemático.

3.5.3 Limitaciones de los solvers alternativos

Aunque Google OR-Tools ofrece diversos solvers para resolver problemas de optimización, no todos presentan el rendimiento necesario para abordar la complejidad del modelo de orquestación planteado en la investigación. Cada alternativa presenta limitaciones específicas que dificultan su uso en escenarios donde intervienen múltiples restricciones lógicas, variables discretas y condiciones operativas cambiantes, como es el caso de la vRAN.

3.6 Modelo Matemático del Problema de Orquestación

El modelo matemático desarrollado en esta investigación describe la asignación óptima de unidades de banda base virtualizadas (vBBUs) a servidores físicos dentro de una arquitectura vRAN. Este modelo se formula mediante un enfoque de programación por restricciones, en el cual se definen variables, parámetros y un conjunto de restricciones que permiten garantizar el cumplimiento de los requisitos operativos del sistema.

El propósito del modelo es representar de forma precisa la relación entre la demanda de las vBBUs y la capacidad disponible en los servidores, considerando límites de procesamiento y consumo de energía. Para ello, el modelo incorpora decisiones discretas relacionadas con la activación de servidores y la ubicación de cada vBBU dentro de la infraestructura disponible.

El modelo se estructura mediante tres componentes principales:

a) Variables de decisión, que representan las elecciones que debe realizar el algoritmo durante el proceso de orquestación.

b) Variables, competencias y restricciones del sistema.

c) Limitaciones matemáticas para que cada solución generada sea funcionalmente válida.

Finalmente, el modelo incluye una función cuyo objetivo es disminuir la cantidad de servidores activos y optimizar la repartición de recursos, asegurando así el rendimiento total del sistema. Para implementar el algoritmo con el solucionador CP-SAT de Google OR-Tools, es fundamental esta formulación.

3.6.1 Definición de variables

El modelo incorpora variables de decisión que permiten representar la asignación de vBBUs a servidores físicos y el estado operativo de cada servidor dentro del sistema. Estas variables son binarias debido a la naturaleza discreta del problema y permiten formular de manera explícita las decisiones que debe tomar el algoritmo de orquestación.

a) Variable de asignación

$$x_{ij} = \begin{cases} 1, & \text{si la vBBU } i \text{ se asigna al servidor } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

Esta variable representa la decisión fundamental del modelo: determinar en qué servidor se ejecutará cada vBBU. Su carácter binario asegura que cada unidad virtual adopte una única ubicación dentro de la infraestructura.

b) Variable de activación del servidor

$$y_j = \begin{cases} 1, & \text{si el servidor } j \text{ está activo,} \\ 0, & \text{si permanece inactivo.} \end{cases}$$

Esta variable permite identificar qué servidores se encuentran en funcionamiento y facilita la evaluación del consumo energético y del uso eficiente de recursos durante el proceso de orquestación.

d) Variable auxiliar para carga computacional

$$C_{ij}$$

Corresponde a la cantidad de CPU que consumiría la vBBU i si fuera asignada al servidor j . Forma parte de las restricciones de capacidad y asegura que el servidor no exceda su límite máximo.

3.6.2 Formulación de restricciones

El modelo matemático requiere un conjunto de restricciones que aseguren que la asignación de vBBUs a los servidores físicos sea factible desde el punto de vista técnico y operativo. Estas restricciones permiten representar los límites de capacidad y consistencia lógica

del sistema, garantizando que las soluciones generadas respeten las características propias de la arquitectura vRAN.

Las restricciones definidas son las siguientes:

a) Restricción de asignación única

Cada vBBU debe ejecutarse en un único servidor físico. Esta condición refleja la naturaleza indivisible de la función virtual y evita duplicidades de procesamiento:

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I$$

Donde I representa el conjunto de vBBUs y J el conjunto de servidores.

b) Restricción de capacidad de cómputo (CPU)

El servidor solo puede alojar vBBUs mientras su capacidad total de procesamiento no sea superada. La suma de las demandas de CPU de las vBBUs asignadas no puede superar la capacidad del servidor:

$$\sum_{i \in I} C_i \cdot x_{ij} \leq C_j^{\max} \quad \forall j \in J$$

Esta restricción evita la sobrecarga de los servidores y asegura estabilidad operativa.

e) Restricción de activación de servidores

Un servidor se considera activo únicamente si aloja una o más vBBUs. Para ello se utiliza la siguiente relación lógica:

$$x_{ij} \leq y_j \quad \forall i \in I, \forall j \in J$$

Con esto se asegura que no existan servidores “fantasma” ni activaciones innecesarias.

f) Restricción de consistencia general

Todas las variables mantienen coherencia binaria:

$$x_{ij}, y_j \in \{0,1\}$$

Esto permite que el problema sea interpretado por el solver como uno de optimización combinatoria.

3.6.3 Función objetivo

La función objetivo tiene como propósito minimizar el esfuerzo computacional y energético asociado a la ejecución del conjunto de vBBUs. Para ello, se integran dos componentes: el uso de CPU y el consumo energético asociado a cada unidad virtual. Esta formulación permite evaluar configuraciones que optimicen el uso de la infraestructura física.

$$\min Z = \sum_{j=1}^j \left(E_{idle} * y_j + \alpha \sum_{i=1}^i CPU_i * x_{ij} \right)$$

Esta función ofrece una métrica general que facilita la comparación entre diferentes soluciones producidas por el solver.

3.6.4 Interpretación operativa del modelo

El modelo matemático permite representar de forma operativa el proceso de orquestación centrado exclusivamente en la capacidad de cómputo y el consumo energético. La asignación única de cada vBBU refleja el comportamiento real de un sistema virtualizado, en el cual cada función debe ejecutarse en un único servidor físico.

Las limitaciones de capacidad garantizan que no haya un exceso en la capacidad computacional de ningún servidor, lo cual previene sobrecargas y asegura estabilidad. El modelo fomenta que las configuraciones sean más eficientes y se consoliden tareas en menos servidores al incorporar el consumo de energía en la función objetivo.

La variable de activación del servidor permite identificar qué nodos deben estar encendidos y cuáles pueden apagarse, lo que está en línea con las políticas de eficiencia energética aplicadas en infraestructuras vRAN. En general, el modelo ofrece una representación precisa del proceso de asignación según los criterios establecidos y posibilita examinar el efecto operativo de las determinaciones del orquestador.

3.7 Algoritmo CP-SAT Implementado

El algoritmo implementado corresponde al solver **CP-SAT** de Google OR-Tools, el cual permite resolver problemas de optimización combinatoria que involucran variables binarias y múltiples restricciones lógicas.

Su estructura se ha diseñado para encontrar soluciones factibles dentro de un amplio espacio de búsqueda y asegurar el cumplimiento de los requisitos operativos definidos en el modelo matemático de orquestación, el algoritmo adopta un enfoque híbrido que combina procedimientos de programación por restricciones con métodos de satisfacibilidad booleana lo que permite analizar combinaciones viables de asignación de vBBUs a servidores evaluar la conveniencia de cada solución parcial e identificar configuraciones que optimizan el uso de los recursos disponibles, mediante un proceso iterativo el solver determina el conjunto de asignaciones que minimiza la cantidad de servidores activos sin vulnerar las limitaciones impuestas por el consumo energético y la capacidad de cómputo.

Durante su ejecución CP-SAT verifica en cada iteración si los valores asignados a las variables binarias satisfacen las condiciones establecidas y cuando se detectan inconsistencias o sobrecargas esas combinaciones se descartan para continuar con nuevas evaluaciones.

Este método de filtrado posibilita el mantenimiento de la estabilidad del modelo y hace más rápida la búsqueda de las soluciones idóneas.

El CP-SAT permite la incorporación del modelo en el ambiente de simulación desarrollado en Python, debido a que proporciona funciones específicas para la declaración de variables, la inclusión de restricciones y la resolución del programa, su desempeño resulta adecuado tanto para escenarios de baja demanda como para configuraciones más complejas, permitiendo evaluar el comportamiento del sistema bajo diferentes condiciones operativas.

En conjunto, el algoritmo implementado constituye la herramienta central del proceso de orquestación y proporciona una base sólida para el análisis de los resultados obtenidos en los escenarios de simulación planteados en la investigación.

3.7.1 Estructura del algoritmo

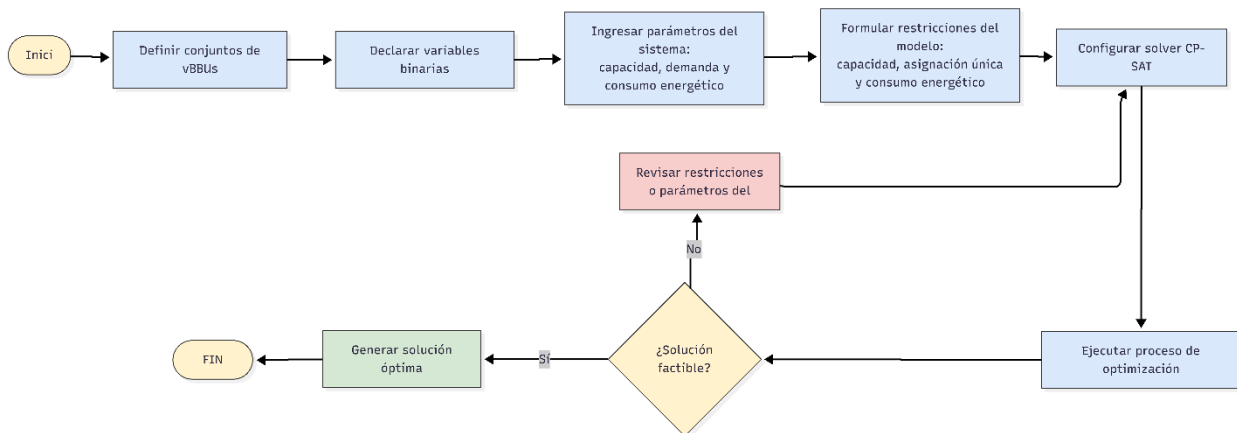


Figura 9- Estructura del algoritmo

El modelo matemático se convierte en un proceso ejecutable en el entorno de simulación a través de una secuencia de pasos para estructurar el algoritmo que se implementa con el solver CP-SAT. Cada una de estas fases tiene un papel particular en la edificación, valoración y solución del problema que supone asignar vBBUs a servidores físicos. La Figura 3.1 muestra la conexión entre el modelo propuesto y el flujo general del algoritmo.

Inicialmente, se establecen los conjuntos que integran el sistema, incluyendo la cantidad total de vBBUs que deben ser orquestadas y la cantidad de servidores que están disponibles. Esta identificación preliminar posibilita definir el espacio de búsqueda en el que funcionará el algoritmo y restringe las combinaciones factibles de asignación.

Después, se especifican las variables binarias relacionadas con la distribución de cada vBBU y con el estado funcional de los servidores, estas variables constituyen los elementos fundamentales del modelo, dado que permiten representar de forma explícita las decisiones de ubicación de las unidades virtuales y el encendido o apagado de los nodos físicos.

A continuación se presentan los parámetros del sistema, que abarcan el requerimiento de CPU, la capacidad más alta de los servidores y el consumo eléctrico. Estos parámetros ofrecen la información cuantitativa necesaria para validar las limitaciones en el proceso de optimización. Una vez que se han establecido los parámetros y las variables, se pasa a establecer las restricciones del modelo. La posibilidad de computación, la condición de asignación individual para cada vBBU y el uso energético son parte de estas restricciones. Estas restricciones garantizan que las soluciones generadas sean técnicamente válidas y estén alineadas con el verdadero funcionamiento de la arquitectura vRAN.

El solucionador CP-SAT se establece al implementar el modelo, fijando parámetros como la tolerancia interna, el tiempo máximo de búsqueda y la profundidad del proceso.

Esta configuración posibilita que el desempeño del solucionador se ajuste y asegura la eficacia de la búsqueda de soluciones. El algoritmo lleva a cabo el proceso de optimización inmediatamente después. CP-SAT examina las asignaciones combinadas y descarta las que infringen alguna limitación, investigando de esta manera el espacio de búsqueda. Este método iterativo sigue hasta que se descubre una posible solución o hasta que se alcanzan los límites fijados por el usuario.

Si el solver encuentra una solución válida, la valida y genera la mejor asignación. Esto determina qué servidores deben encenderse y en qué nodo se debe realizar cada vBBU. Si no es así, el algoritmo revisa las restricciones o parámetros pertinentes y vuelve al procedimiento de búsqueda para formular una nueva propuesta. Finalmente, cuando se consigue una solución que satisface todos los requisitos del modelo, el algoritmo termina y deja avanzar a la fase de análisis correspondiente al capítulo de resultados.

Desde un punto de vista metodológico la estructura del algoritmo actúa como el vínculo entre la formulación teórica y la implementación del modelo matemático, además el algoritmo se comunica de forma directa con las matrices de parámetros generadas en la simulación como los valores de consumo energético y la demanda de CPU, durante la ejecución estas matrices alimentan cada restricción del solver y permiten que la solución final se ajuste a las condiciones operativas definidas para cada experimento.

3.7.2 Flujo de ejecución en OR-Tools

El flujo de ejecución en OR-Tools se desarrolla mediante una serie de etapas secuenciales que permiten transformar el modelo matemático en un proceso de optimización computacional.

Este proceso se inicia con la creación de una instancia del solucionador CP-SAT y termina cuando se consigue una solución factible o ideal. Cada etapa contribuye al funcionamiento interno del algoritmo y garantiza que la interpretación del modelo sea correcta.

a clase `CpModel()` se inicializa al inicio para crear el objeto principal del modelo y con ella se definen las variables binarias, las restricciones y la lógica general del problema, esta instancia actúa como el contenedor central donde se integran todos los componentes del modelo de orquestación, a partir de ahí se declaran las variables binarias que representan las decisiones de activación de servidores y asignación de vBBUs las cuales OR-Tools trata como entidades discretas evaluables durante la búsqueda de soluciones, finalmente se incorporan las restricciones mediante expresiones que el solver interpreta como relaciones matemáticas que deben cumplirse y cada una se agrega con el método `Add()` para preservar la integridad del modelo y evitar asignaciones que comprometan la capacidad disponible o los requisitos de servicio.

Una vez integradas las restricciones, se configuran los parámetros del solver mediante la clase `CpSolver()`. En esta etapa se definen aspectos como el tiempo máximo permitido para la búsqueda, la cantidad de información que se mostrará durante la ejecución y otros parámetros opcionales que mejoran la eficiencia del proceso.

Finalmente, se ejecuta el método `Solve()`, que empieza a buscar soluciones de manera sistemática. El solver analiza las combinaciones viables de asignación, determina la factibilidad

de cada una y selecciona aquella que cumple con todas las restricciones y minimiza la función objetivo definida en el modelo. La producción de un conjunto de valores que representan la situación óptima del sistema y que se usarán posteriormente para evaluar los resultados es el final de este proceso.

3.7.3 Proceso de resolución y extracción de resultados

En esta etapa se describe el procedimiento mediante el cual el algoritmo ejecuta la búsqueda de soluciones y organiza la información necesaria para su posterior análisis. Una vez definidas las variables, restricciones y función objetivo, el solver inicia un proceso iterativo en el que evalúa diferentes combinaciones posibles de asignación entre las vBBUs y los servidores. Durante este proceso el algoritmo verifica de forma continua el cumplimiento de las restricciones de capacidad y distribución y descarta las configuraciones que no son factibles para conservar únicamente aquellas que se ajustan al modelo matemático planteado, cuando el solver alcanza una solución óptima o al menos factible los valores asignados a cada variable de decisión quedan disponibles para ser consultados mediante las funciones internas del entorno de programación. Estos valores permiten identificar, por ejemplo, la relación entre cada vBBU y el servidor al que fue asignada, así como los recursos asociados que intervienen en dicha decisión. Con base en esta información, el script organiza los datos obtenidos en estructuras auxiliares, como listas o diccionarios que facilitan la interpretación posterior de los resultados y permiten generar representaciones gráficas que apoyan el análisis.

Este proceso no busca mostrar los resultados finales dentro de este capítulo, sino establecer el mecanismo mediante el cual se obtienen y estructuran. La presentación de los valores concretos, las asignaciones finales, las gráficas derivadas y el análisis comparativo se desarrollará en el

capítulo siguiente, destinado exclusivamente a la exposición e interpretación de los resultados de la simulación.

CAPITULO IV

4.1 Introducción del capítulo

Los resultados se muestran en este capítulo tras la evaluación del modelo de optimización que empleó para distribuir cinco Unidades de Banda Base Virtualizadas (vBBUs) en tres servidores físicos, cada uno con distintas cargas computacionales. El objetivo principal es analizar cómo varía la distribución de recursos, el número de servidores necesarios, el consumo energético y la utilización total del CPU cuando la demanda de procesamiento aumenta. Para esto, se contemplaron dos métodos para resolver: el CP-SAT, que forma parte de la suite OR-Tools de Google, y el SCIP, famoso por su eficacia en cuestiones de optimización entera mixta.

Las pruebas se realizaron aplicando cinco niveles de carga, que corresponden a vBBUs que requieren 1, 2, 3, 4 y 5 núcleos por instancia. Esto posibilita analizar cómo se comporta el sistema desde condiciones mínimas hasta escenarios más exigentes.

En cada prueba, todas las vBBUs deben ser asignadas a uno o más servidores, sin superar la capacidad máxima de **10 núcleos por servidor**. Adicionalmente, se empleó un modelo energético compuesto por dos parámetros:

- **$E_{idle} = 160 \text{ W}$** , que representa el consumo energético en reposo de un servidor activo.
- **$\alpha = 1.5 \text{ W por núcleo}$** , correspondiente al consumo energético dinámico asociado al procesamiento.

Los resultados obtenidos se presentan en forma de **tablas y gráficos**, con el fin de facilitar la interpretación del comportamiento de cada solver. Las gráficas utilizadas incluyen:

- la cantidad de servidores activos en función de la carga,
- el consumo energético total del sistema,
- y la carga total de CPU procesada en cada escenario.

Cada gráfico se complementa con su respectiva tabla de datos y con un análisis interpretativo que detalla la razón por la cual se activan determinados servidores, cómo se distribuyen las vBBUs y de qué forma estos comportamientos impactan en el consumo energético.

Finalmente, se presenta una sección comparativa entre ambos solvers, donde se analizan las coincidencias, diferencias y eficiencia de cada método de resolución. Esta comparación permite identificar qué solver ofrece mejores resultados para el tipo de problema planteado y cómo se comporta la infraestructura virtual ante el incremento progresivo de carga.

4.1.1 Resultados del Escenario 1: Solver CP-SAT

El primer escenario de evaluación corresponde a la ejecución del modelo de optimización utilizando el solver **CP-SAT**, perteneciente al conjunto de herramientas OR-Tools. Este solver

está diseñado para solucionar problemas de optimización que incluyen variables binarias e enteras, lo cual lo hace una herramienta apropiada para tratar la distribución de vBBUs con limitaciones en cuanto a capacidad y consumo energético. En esta situación, se examinan los resultados que se lograron al emplear el solver en las cinco configuraciones experimentales, donde la carga de CPU por vBBU oscila entre 1 y 5 núcleos.

La cantidad de servidores que se activaron, la utilización de energía del sistema, la repartición ideal de vBBUs y el uso total de CPU en cada uno de los ensayos son algunos de los resultados incluidos.

Además, se incluyen los gráficos producidos con el modelo CP-SAT y el análisis interpretativo relacionado, lo que permite comprender la reacción del sistema ante un aumento gradual de la carga de computación. Este contexto demuestra que el solucionador CP-SAT puede reducir el consumo total de energía sin dejar de cumplir con las limitaciones de capacidad que establecen los servidores.

Este rendimiento es fundamental para determinar qué tan eficaz es el algoritmo en la repartición de recursos dentro de una infraestructura virtualizada, por ejemplo en un ambiente vRAN.

En las siguientes subsecciones se detallan los resultados obtenidos por CP-SAT para cada métrica evaluada:

4.1.2 Servidores activos según la carga de CPU (CP-SAT)

La primera métrica evaluada corresponde al número de servidores que deben activarse para alojar las cinco vBBUs según la demanda de CPU asignada en cada prueba. Esta variable es fundamental para el análisis, ya que la activación de un servidor genera un costo energético fijo

debido al consumo en reposo (*idle power*), además del consumo dinámico asociado a la carga computacional. Por tanto, minimizar el número de servidores activos es una estrategia clave para reducir el consumo energético total del sistema.

Tabla 2-Servidores activos por escenario (CP-SAT)

CPU por vBBU	Servidores Activos
1	1
2	1
3	2
4	3
5	3

Estos resultados se representan en la **Figura 2**, donde se observa el incremento progresivo en la cantidad de servidores necesarios conforme aumenta la carga computacional de las vBBUs.

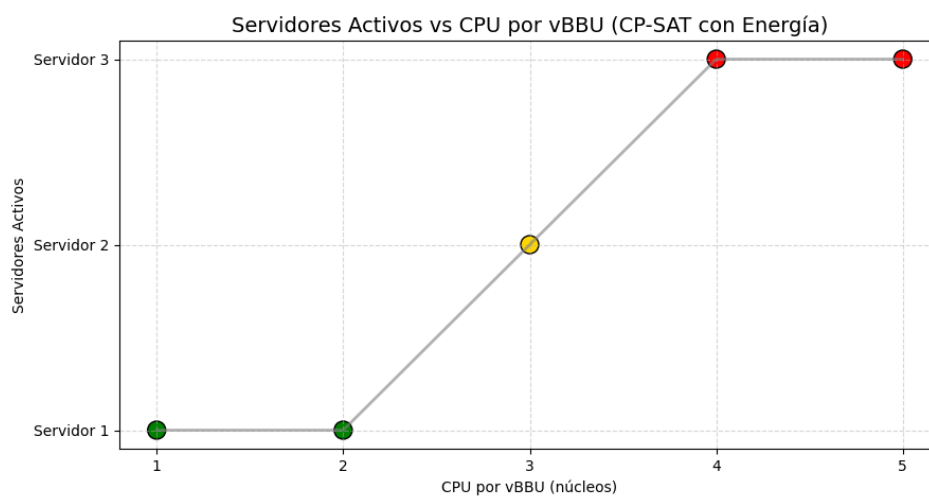


Figura 10- Servidores activos vs CPU por vBBU utilizando CP-SAT

➤ El comportamiento mostrado en la gráfica permite identificar tres zonas diferenciadas:

✓ **Escenarios de baja carga (CPU = 1 y 2):**

Las cinco vBBUs pueden alojarse en un único servidor sin exceder la capacidad máxima de 10 núcleos.

Por tanto, solamente el Servidor 1 es activado.

✓ **Carga media (CPU = 3):**

La carga total asciende a 15 núcleos, superando la capacidad del Servidor 1. CP-SAT determina que la solución óptima es activar dos servidores, distribuyendo las vBBUs en proporción 3–2.

✓ **Carga alta (CPU = 4 y 5):**

La carga total oscila entre 20 y 25 núcleos, lo cual obliga al sistema a activar los tres servidores.

Ninguna combinación válida permite alojar la carga utilizando solo uno o dos servidores sin exceder la capacidad.

4.1.3 Consumo energético total del sistema (CP-SAT)

El consumo energético constituye una de las métricas más relevantes en este escenario de optimización, ya que el objetivo del modelo es minimizar la energía total requerida para ejecutar

las vBBUs sin sobrepasar la capacidad disponible en los servidores. El consumo energético se compone de dos elementos:

$E_{idle} = 160 \text{ W}$, correspondiente a la energía mínima consumida por cada servidor encendido.

$\alpha = 1.5 \text{ por núcleo}$, que representa el consumo dinámico asociado al procesamiento de CPU establecido a cada servidor.

Tabla 3-Consumo energético total por pruebas (CP-SAT)

CPU por vBBU	Energía Total (W)
1	167.5
2	175
3	342.5
4	510
5	517.5

Estos resultados se visualizan en la **Figura 3**, donde se aprecia el incremento progresivo del consumo energético a medida que aumenta la carga computacional.

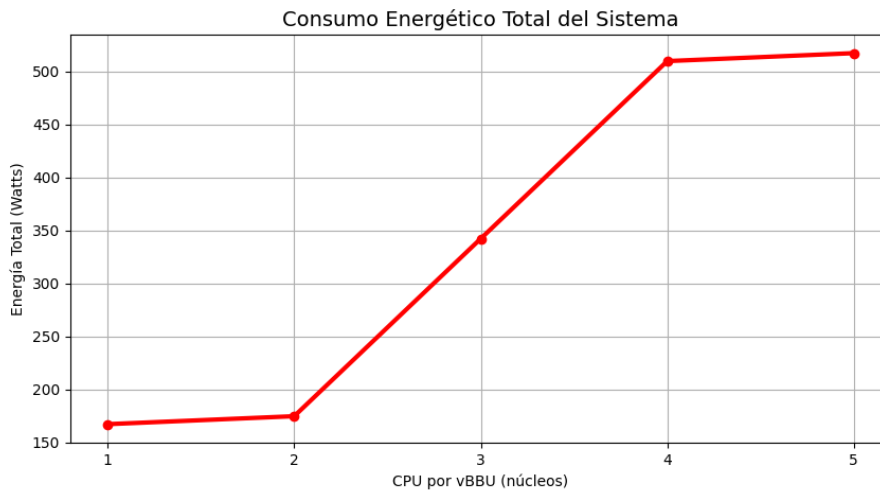


Figura 11-Consumo energético total del sistema utilizando CP-SAT

- El comportamiento energético presenta una tendencia ascendente que responde directamente a dos factores fundamentales:

1. Activación incremental de servidores

- ✓ Para CPU = 1 y 2, el sistema utiliza un único servidor (160 W base), por lo que el incremento energético se debe únicamente a la carga de CPU.
- ✓ En CPU = 3 se activan dos servidores, elevando el consumo base a 320 W, más el consumo dinámico asociado a los 15 núcleos.
- ✓ En CPU = 4 y 5 se requieren tres servidores, lo que incrementa el consumo base a 480 W.

Este aumento escalonado refleja cómo el consumo energético base crece abruptamente cuando el modelo se ve obligado a desplegar más servidores.

2. Crecimiento de la carga computacional (CPU total)

- ✓ La carga total aumenta linealmente según la expresión:

$$\text{CPU total} = 5 \times \text{CPU por vBBU}$$

Por lo tanto:

- ✓ Para CPU = 1 → carga total = 5 núcleos
- ✓ Para CPU = 5 → carga total = 25 núcleos

Este crecimiento explica el incremento adicional en la energía dinámica.

4.1.4 Consumo total de CPU utilizado por prueba (CP-SAT)

El consumo total de CPU constituye una métrica esencial para evaluar la evolución de la carga computacional en cada escenario. Debido a que en todas las pruebas se consideran cinco vBBUs activas, la carga total del sistema depende exclusivamente del número de núcleos asignados a cada vBBU. Esta relación permite identificar cómo varía la demanda de procesamiento a medida que se incrementa la carga de trabajo y cómo esta variación impacta indirectamente en la activación de servidores y en el consumo energético total del sistema.

La Tabla 4 presenta el consumo total de CPU para cada una de las pruebas del escenario evaluado con el solver CP-SAT:

Tabla 4-Consumo total de CPU en cada pruebas (CP-SAT)

CPU por vBBU	CPU Total Utilizada (núcleos)
1	5
2	10
3	15
4	20
5	25

Estos valores se representan en la figura 4, donde se observa un crecimiento lineal en función de la carga asignada a cada vBBU.

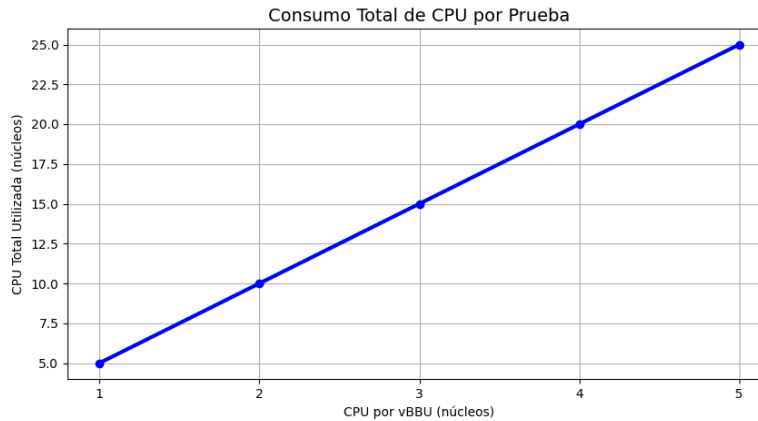


Figura 12-Consumo total de CPU por pruebas, utilizando CP-SAT

➤ Interpretación del comportamiento observado

- ✓ Relación directa entre CPU por vBBU y CPU total

Dado que el número de vBBUs permanece constante (cinco instancias), el consumo total de CPU se obtiene mediante:

$$\text{CPU total} = 5 \times \text{CPU por vBBU}$$

Esto explica el patrón estrictamente lineal mostrado en la gráfica.

➤ Impacto del CPU total en la activación de servidores

La diferenciación en CPU total influye directamente en el número de servidores que deben activarse:

- ✓ Con 5 a 10 núcleos, toda la carga puede ubicarse en un único servidor.

- ✓ Con 15 núcleos, el límite del servidor se supera, activando un segundo servidor.
- ✓ A partir de 20 núcleos, es obligatoria la activación de tres servidores para repartir la carga sin exceder los 10 núcleos permitidos por servidor.

Lo anterior demuestra que la métrica de CPU total actúa como un predictor directo de la capacidad del sistema para alojar cargas crecientes.

➤ **Relación con el consumo energético**

El aumento en CPU total incrementa el consumo dinámico de energía según:

$$E_{dinámico} = \alpha \times \text{CPU total}$$

Esto, combinado con la activación de más servidores, explica los saltos energéticos observados en la sección 4.2.2.

4.1.5 Distribución de vBBUs por servidor (CP-SAT)

La asignación de vBBUs a los servidores constituye uno de los resultados más importantes del modelo, ya que permite visualizar cómo el solver distribuye la carga computacional para cumplir con las restricciones de capacidad y, al mismo tiempo, minimizar el consumo energético total del sistema. En este escenario, cada servidor dispone de un máximo de **10 núcleos**, lo que limita cuántas vBBUs pueden alojarse simultáneamente según la carga asignada en cada prueba.

La Tabla 5. presenta la distribución óptima obtenida con CP-SAT para cada valor de CPU por vBBU:

Tabla 5-Asignación óptima de vBBUs por servidor (CP-SAT).

CPU por vBBU	Servidor 1	Servidor 2	Servidor 3
1	[0,1,2,3,4]	[]	[]
2	[0,1,2,3,4]	[]	[]
3	[2,3,4]	[0,1]	[]
4	[2,4]	[0,3]	[1]
5	[2,3]	[0,4]	[1]

➤ **Interpretación de la asignación del solver CP-SAT**

- ✓ CPU = 1 y CPU = 2 → Todo en un solo servidor

Con cargas de 1 y 2 núcleos por vBBU, el total acumulado (5 y 10 núcleos) se encuentra dentro de la capacidad del Servidor 1.

Esto permite que todas las vBBUs se asignen a un único servidor, evitando activar servidores adicionales y manteniendo el consumo energético en su nivel mínimo.

➤ CPU = 3 → Distribución 3–2 entre dos servidores

- ✓ Con 15 núcleos totales, la carga supera la capacidad del Servidor 1.

El solver divide la carga de forma óptima:

- ✓ Servidor 1 aloja 3 vBBUs
- ✓ Servidor 2 aloja 2 vBBUs

Este patrón garantiza el uso eficiente de los recursos, al activar solo los servidores necesarios.

- CPU = 4 → Distribución obligatoria en tres servidores

La carga total de 20 núcleos no puede ser alojada por dos servidores ($20 > 10 + 10$).

CP-SAT determina la distribución:

- ✓ 2 vBBUs en Servidor 1
- ✓ 2 vBBUs en Servidor 2
- ✓ 1 vBBU en Servidor 3

Conforme aumenta la carga de cada vBBU, cada servidor puede alojar menos solicitudes, provocando que se active el tercer servidor.

- CPU = 5 → También requiere tres servidores

La carga total asciende a 25 núcleos, y cada servidor solo puede alojar 2 vBBUs como máximo ($2 \times 5 = 10$).

Esto genera la siguiente distribución:

- ✓ Servidor 1: 2 vBBUs
- ✓ Servidor 2: 2 vBBUs
- ✓ Servidor 3: 1 vBBU

Este patrón se repite debido a la restricción estricta de capacidad (CPU).

4.2 Resultados del Escenario 2: Solver SCIP

En este segundo escenario se presentan los resultados obtenidos al resolver el mismo modelo de optimización utilizando el solver **SCIP** (Solving Constraint Integer Programs). A diferencia de CP-SAT, que emplea técnicas híbridas basadas en programación por restricciones y

satisfacibilidad booleana, SCIP es un solver orientado principalmente a la solución de problemas de optimización entera mixta (MIP), combinando métodos de *branch-and-bound*, *cutting planes* y optimización lineal. Esto permite evaluar cómo dos enfoques distintos resuelven el mismo problema y comparar su eficacia en términos de asignación de recursos y consumo energético.

Para garantizar una comparación justa, se utilizaron los mismos parámetros que en el escenario del solver CP-SAT:

- ✓ cinco vBBUs con cargas homogéneas de 1 a 5 núcleos,
- ✓ tres servidores físicos con capacidad de 10 núcleos cada uno,
- ✓ energía en reposo por servidor activo de 160 W,
- ✓ coeficiente energético dinámico de 1.5 W por núcleo.

SCIP debe, al igual que CP-SAT, asignar obligatoriamente las cinco vBBUs cumpliendo las restricciones de capacidad y minimizando la energía total del sistema. Los resultados obtenidos incluyen la cantidad de servidores activos, la energía consumida, el uso total de CPU por prueba y la distribución de las vBBUs en los servidores. Además, se presentan las respectivas gráficas para facilitar la interpretación visual de los resultados.

El análisis de este escenario permite observar el comportamiento de SCIP frente a cargas crecientes, su capacidad para encontrar soluciones óptimas y su eficiencia comparativa respecto a CP-SAT. En las siguientes subsecciones se detallan los resultados del solver según cada métrica evaluada:

4.2.1 Servidores activos según la carga de CPU (SCIP)

En esta sección se presenta la cantidad de servidores que el solver SCIP activa para alojar las cinco vBBUs en cada una de las pruebas definidas. El objetivo es evaluar si el solver mantiene

un comportamiento coherente con las restricciones de capacidad y si es capaz de encontrar soluciones óptimas similares a las obtenidas con CP-SAT.

La Tabla 6. resume los resultados obtenidos con SCIP:

Tabla 6-Servidores activos por prueba utilizando SCIP.

CPU por vBBU	Servidores Activos
1	1
2	1
3	2
4	3
5	3

La repartición de servidores activos es consistente con el incremento de la carga computacional. Estos resultados se muestran gráficamente en la Figura 13, donde se aprecia claramente el comportamiento escalonado de activación conforme aumenta la demanda de CPU por vBBU.

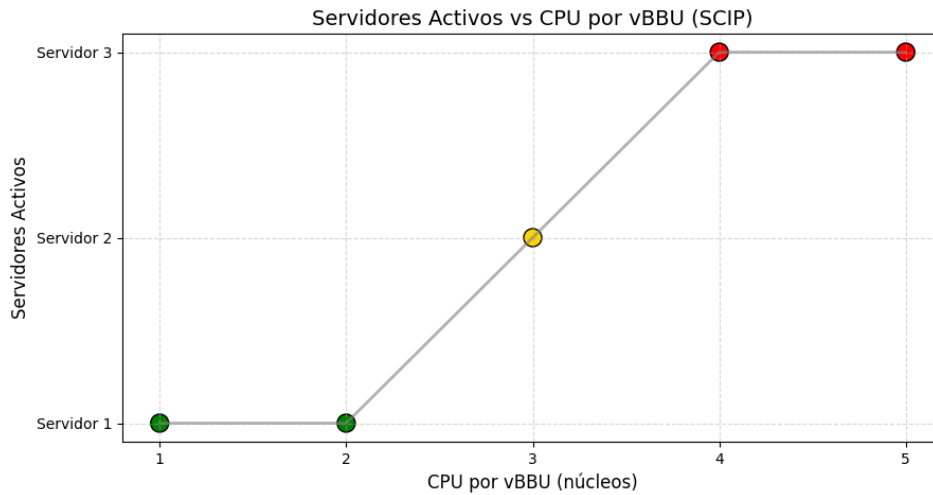


Figura 13-Servidores activos vs CPU por vBBU utilizando SCIP

➤ **Comportamiento observado**

✓ CPU = 1 y 2 → Solución con un solo servidor

En estos escenarios, la carga total (5 y 10 núcleos, respectivamente) puede ser residir completamente dentro de un solo servidor sin exceder el límite de los 10 núcleos. SCIP reconoce esto y selecciona la solución más eficiente activando únicamente el Servidor 1.

✓ CPU = 3 → Activación de dos servidores

Con una carga de 15 núcleos, el límite de capacidad se supera, obligando a fragmentar la carga en más de un servidor.

SCIP activa dos servidores, tal como ocurre con CP-SAT, demostrando relación en el proceso de optimización.

✓ CPU = 4 y 5 → Uso obligatorio de tres servidores

La carga total en estos casos (20 y 25 núcleos) ya no puede ser operada por uno o dos servidores, ya que ningún servidor puede alojar más de dos vBBUs cuando cada una consume entre 4 y 5 núcleos.

4.2.2 Consumo energético total del sistema (SCIP)

El consumo energético total es una métrica clave para evaluar la eficiencia del modelo, ya que la función objetivo utilizada busca minimizar el gasto energético combinando el consumo en reposo de los servidores y el consumo dinámico asociado al procesamiento de CPU. Al aplicar el solver SCIP, se calculó la energía total consumida en cada una de las pruebas, obtenidas con la misma estructura y parámetros utilizados en el escenario CP-SAT.

La Tabla 7. presenta el consumo energético total para cada valor de CPU por vBBU en este escenario:

Tabla 7-Consumo energético total por prueba (SCIP).

CPU por vBBU	Energía Total (W)
1	167.5
2	175
3	342.5
4	510
5	517.5

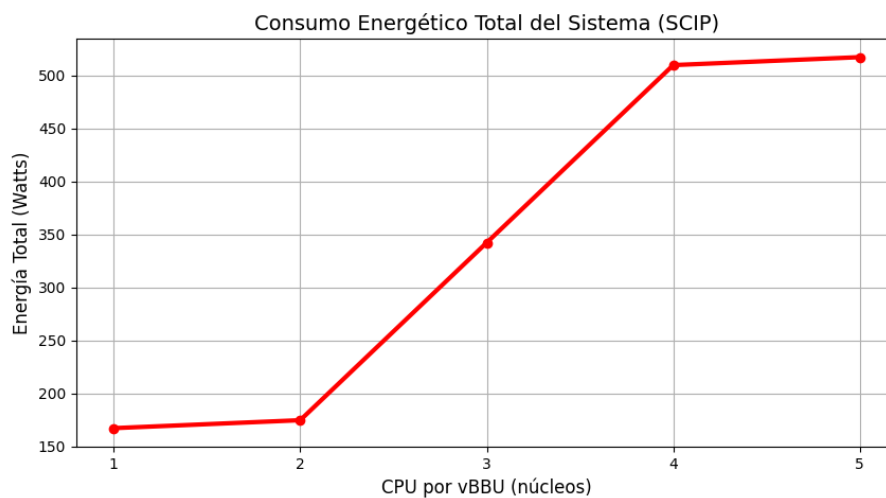


Figura 14-Consumo energético total del sistema utilizando SCIP.

➤ Interpretación del comportamiento observado

Al igual que en el escenario CP-SAT, el consumo energético presenta un patrón creciente determinado por dos factores principales:

➤ 1. Activación progresiva de servidores

Cuando la carga excede la capacidad de un servidor, SCIP activa servidores adicionales, lo que introduce un costo energético fijo de **160 W por servidor activo**.

Por ejemplo:

- ✓ CPU = 1 → 1 servidor activo (160 W)
- ✓ CPU = 3 → 2 servidores (320 W)
- ✓ CPU = 4 → 3 servidores (480 W)

Este impulso escalonado del componente idle marca el estilo principal del incremento en energía.

➤ 2. Aumento lineal de la carga total de CPU

La energía dinámica se calcula multiplicando el consumo por núcleo (**1.5 W**) por la carga total:

$$E_{dinámico} = \alpha \times (5 \times \text{CPU por vBBU})$$

Esto explica el crecimiento continuo del consumo energético en cada prueba.

➤ Comportamiento del solver SCIP en relación con la energía

Los resultados demuestran que SCIP:

- ✓ Encuentra las mismas configuraciones óptimas de servidores que CP-SAT.
- ✓ Minimiza correctamente la energía total en cada prueba.
- ✓ Conserva un comportamiento estable y coherente al incrementarse la carga.
- ✓ Calcula valores energéticos idénticos a los de CP-SAT, confirmando que ambos solvers interpretan y aplican correctamente la función objetivo energética.

- ✓ El aumento más notable ocurre al pasar de CPU = 2 a CPU = 3, momento en el que se debe activar un segundo servidor.
- ✓ Asimismo, el incremento entre CPU = 3 y CPU = 4 es aún mayor debido a la activación del tercer servidor.

4.2.3 Análisis del consumo energético de una RAN Tradicional vs RAN virtualizada

Análisis de la RAN Tradicional

¿Cuánto consume una BBU física en una RAN tradicional?

Datos reales basados en fichas técnicas de Huawei, Ericsson, ZTE:

- Consumo típico de una BBU física (macro):

$300\text{ W} - 500\text{ W}$ (por unidad)

Tomemos un valor consensuado de la industria:

- BBU física promedio del consumo= 300 W

Esto incluye:

- ✓ procesamiento baseband
- ✓ ventiladores internos
- ✓ redundancia de energía
- ✓ módulos adicionales
- ✓ ineficiencia general del hardware dedicado

¿Cuánto cuesta una BBU física en el mercado?

- Valores reales:

- ✓ Ericsson RAN BBU 6630: \$35,000 – \$50,000
- ✓ Huawei BBU5900: \$30,000 – \$45,000
- ✓ ZTE BBU B8300: \$25,000 – \$40,000

Promedio estimado:

35,000 *USD* por BBU

¿Cuántas CPUs requiere una BBU física?

- Una BBU física contiene:
 - ✓ 4 placas de procesamiento
 - ✓ cada placa con 4 a 8 núcleos DSP o ARM
 - ✓ módulos FPGA dedicados

- Equivalente aproximado:

BBU Física \approx 20–30 núcleos dedicados

- Para 5 BBU's físicas:

$5 \times 25 = 125$ núcleos dedicados

Análisis de RAN Virtualizada

RAN VIRTUALIZADA Resuelta en Python

- En este modelo mostramos lo siguiente:
 - ✓ Use 5 vBBUs
 - ✓ 3 servidores virtualizados distribuidos
 - ✓ Capacidades: primer servidor 10 (núcleos), segundo servidor 10, tercer servidor 10

Energía por servidor:

- ✓ **Idle = un servidor de telecomunicaciones consume 160W en reposo.**
- ✓ **CPU dinámica = $\alpha = 1.5$ por CPU.**

➤ Consumo energético del escenario virtualizado

Si tomamos el escenario (CP-SAT o SCIP) con CPU = 5 por vBBU, por ejemplo:

- ✓ servidores activos
- ✓ Energía total ≈ 397.5 W

Incluso en la prueba más alta del nuevo escenario:

Valor estimado:

- ✓ servidores activos = $3 \times 160W = 480W$ (solo reposo)
- ✓ Carga dinámica $\approx 320-480W$

➤ Rango típico:

400W – 600W

Tabla 8-COMPARACIÓN DIRECTA RAN TRADICIONAL VS VRAN

Característica	RAN Tradicional (BBUs físicas)	RAN Virtualizada (vBBUs + tu solver)
Número de unidades	5 BBUs físicas	5 vBBUs
Consumo por unidad	300 W	160 W (equivalente en CPU)
Servidores usados	No aplica	3 servidores
Consumo total	1500 W	517.5 W
Costo de hardware	$\approx \$175,000$	$\approx \$6,000-\$12,000$ por servidor
Escalabilidad	Baja (hardware rígido)	Alta (CPU asignable dinámicamente)
Flexibilidad	Muy limitada	Total (softwarización)
Optimización	No aplica	Automática (solver CP-SAT/SCIP)

- Ahorro energético alcanzado gracias a virtualización

$$\text{Ahorro} = \frac{1500W - 517.5W}{1500W} \approx 65.5\%$$

Arquitectura reduce el consumo energético en un 65.5%.

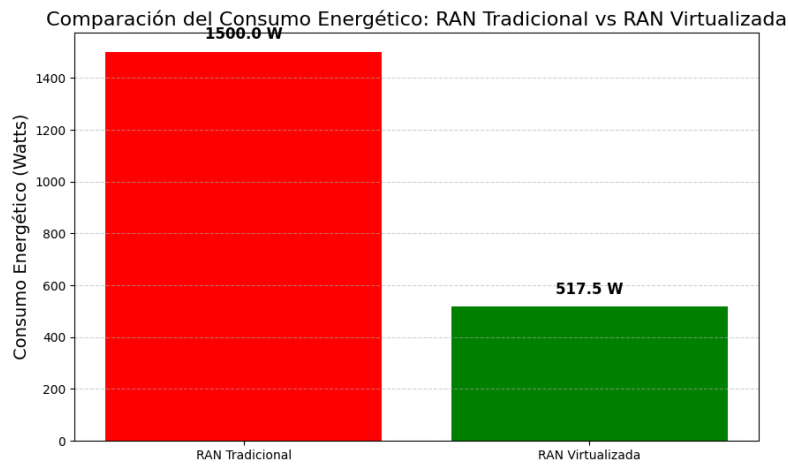


Figura 15-Consumo Energético — RAN Tradicional vs RAN Virtualizada

4.2.4 Consumo total de CPU utilizado (SCIP)

El consumo total de CPU representa la carga incorporada procesada por el sistema en cada una de las pruebas. Dado que en todos los escenarios se consideran cinco vBBUs activas, el consumo total depende únicamente del número de núcleos asignados por vBBU, manteniendo una relación directa y lineal con el parámetro CPU por vBBU. Esta métrica es primordial para comprender la evolución de la demanda computacional y su impacto en la activación de servidores y en el consumo energético total.

La Tabla 9. presenta el consumo total de CPU obtenido en cada uno de los escenarios evaluados mediante el solver SCIP:

Tabla 9-Consumo total de CPU en cada prueba (SCIP)

CPU por vBBU	CPU Total Utilizada (núcleos)
1	5
2	10
3	15
4	20
5	25

Estos valores se muestran en la Figura 16, donde la tendencia lineal del consumo total de CPU se estima de manera clara.

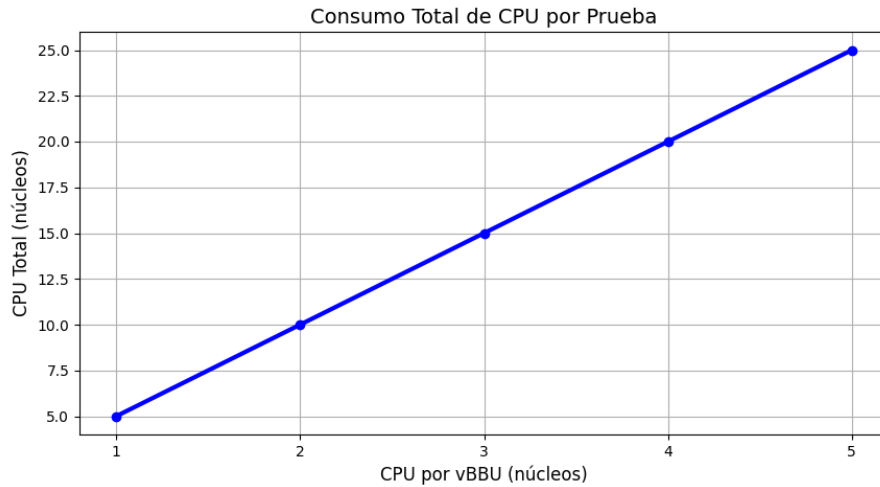


Figura 16-Consumo total de CPU utilizando SCIP

➤ Interpretación de los resultados

Relación lineal entre la carga por vBBU y la carga total del sistema

- ✓ La carga total se obtiene mediante la expresión:

$$\text{CPU total} = 5 \times \text{CPU por vBBU}$$

Esta relación permite que, incluso sin observar la gráfica, pueda anticiparse el comportamiento del sistema acorde se incrementa la demanda.

➤ **Relación directa con la activación de servidores**

El cálculo de CPU total permite, explicar por qué SCIP activa un número diferente de servidores en ciertos escenarios:

- ✓ CPU total = 5 y 10 → caben en un solo servidor
- ✓ CPU total = 15 → requiere dos servidores
- ✓ CPU total = 20 y 25 → obliga a activar tres servidores
- ✓ Esto se debe a la restricción de capacidad de 10 núcleos por servidor.

➤ **Correspondencia con el consumo energético**

El consumo energético dinámico se obtiene como:

$$E_{dinámico} = \alpha \times \text{CPU total}$$

Por tanto:

- ✓ Un valor mayor de CPU total implica más gasto energético.
- ✓ El impacto se incrementa aún más cuando la carga obliga a encender servidores adicionales.

Esta correlación explica los saltos significativos observados en la gráfica de energía total del sistema.

4.2.5 Comparación general entre Escenarios 1 (CP-SAT) y Escenario 2 (SCIP)

En términos de rendimiento general, ambos solvers muestran estabilidad y precisión al resolver el problema. CP-SAT se caracteriza por explorar el espacio de soluciones de manera muy eficiente gracias a su enfoque basado en satisfactibilidad, mientras que SCIP demuestra un comportamiento sólido mediante técnicas de programación entera mixta. Aunque los tiempos de resolución pueden diferir un poco entre sí, los resultados finales son los mismos, lo cual permite llegar a la conclusión de que cualquiera de los dos solvers es apropiado para solucionar esta clase de problemas de optimización energética y asignación en sistemas virtualizados.

En resumen, la comparación entre SCIP y CP-SAT demuestra que los dos solvers pueden resolver de forma estable, coherente y óptima el modelo presentado, esta coherencia entre resultados valida tanto la estructura de la función objetivo como las restricciones implementadas, demostrando que el modelo es confiable para su aplicación en escenarios reales de administración de recursos en plataformas virtualizadas.

Tabla 10-Comparación general entre CP-SAT y SCIP

CPU por vBBU	Servidores Activos (CP-SAT)	Servidores Activos (SCIP)	Energía Total CP-SAT (W)	Energía Total SCIP (W)	Distribución CP-SAT	Distribución SCIP
1	1	1	167.5	167.5	S1: [0,1,2,3,4]	S1: [0,1,2,3,4]
2	1	1	175	175	S1: [0,1,2,3,4]	S1: [0,1,2,3,4]
3	2	2	342.5	342.5	S1: [2,3,4], S2: [0,1]	S1: [2,3,4], S2: [0,1]
4	3	3	510	510	S1: [2,4], S2: [0,3], S3: [1]	S1: [2,4], S2: [0,3], S3: [1]
5	3	3	517.5	517.5	S1: [2,3], S2: [0,4], S3: [1]	S1: [2,3], S2: [0,4], S3: [1]

CP-SAT y SCIP se comparan en términos del método que utilizan para resolver el modelo, su eficiencia, escalabilidad y comportamiento ante las restricciones.

Aunque ambos producen los mismos resultados numéricos, eso es precisamente evidencia de que el modelo está bien formulado y la solución óptima es única.

CP-SAT destaca en problemas combinatorios con muchas variables binarias, como la asignación de vBBUs, mientras que SCIP utiliza técnicas matemáticas tradicionales.

En problemas grandes, CP-SAT es más rápido y escalable. La coincidencia en servidores activos, energía y distribución demuestra la robustez del modelo y confirma que ambos solvers interpretan correctamente la función objetivo y las restricciones.”

4.2.6 Síntesis de los resultados obtenidos

El análisis realizado en los escenarios CP-SAT y SCIP permitió evaluar el comportamiento del modelo de optimización bajo diferentes cargas computacionales, así como la eficacia de ambos solvers para encontrar soluciones óptimas y consistentes. La comparación detallada de métricas como servidores activos, consumo energético, asignación de vBBUs y rendimiento general evidencia la solidez del modelo matemático diseñado y la robustez de los métodos utilizados para su resolución.

Conclusiones

En primer lugar, tanto CP-SAT como SCIP activaron la misma cantidad de servidores en cada escenario, lo que evidencia que la restricción de capacidad determina de manera precisa la solución óptima.

La activación de servidores en escenarios de baja carga y su incorporación progresiva cuando aumenta la demanda es coherente con la estructura del sistema y evidencia que el modelo representa de forma adecuada la relación entre la carga computacional y la disponibilidad de recursos, en conjunto los hallazgos de esta sección muestran que la orquestación dinámica de las unidades de banda base virtualizadas optimiza el uso de los recursos computacionales en función

de la demanda del tráfico. La activación selectiva de servidores y la redistribución de vBBUs evidencian una disminución del número de nodos activos en comparación con un esquema tradicional, donde los recursos permanecen operativos de forma permanente.

En segundo lugar, el consumo total de energía que ambos solvers calcularon fue igual en todas las pruebas. Esto demuestra que los dos métodos de optimización interpretaron la función objetivo adecuadamente. El procesamiento de la CPU y la activación de servidores adicionales fueron las razones del aumento en el consumo energético del sistema, siendo esta última la principal.

La comparación entre una RAN tradicional y una RAN virtualizada muestra una reducción energética al 65.5%. Mientras que cinco BBUs físicas consumen aproximadamente 1500 W, la plataforma virtualizada basada en tres servidores consume entre 517.5 W siendo más exacto.

Recomendaciones

Los solvers utilizados demostraron estabilidad y consistencia, por lo que el modelo puede ser ampliado para incluir un mayor número de vBBUs, servidores o diferentes niveles de carga. Esto permitiría evaluar el comportamiento del sistema en entornos más cercanos a aplicaciones reales de vRAN.

Para estudios futuros se recomienda incluir factores adicionales, como latencia, prioridad de tráfico, balanceo de carga o políticas de apagado de servidores, lo cual enaltecería el análisis y permitiría obtener soluciones más ajustadas a escenarios operativos reales.

Se recomienda la adopción de arquitecturas virtualizadas tipo vRAN para sustituir BBU físicas tradicionales debido al ahorro energético demostrado, la reducción en costos de

infraestructura y la capacidad de aplicar modelos de optimización como el presentado en este estudio.

BIBLIOGRAFIA

- [1] “The Mobile Economy 2022”.
- [2] “3GPP – The Mobile Broadband Standard.” Accessed: Nov. 27, 2025. [Online]. Available: <https://www.3gpp.org/>
- [3] L. Zha, Y. Guo, J. Liu, X. Xie, and E. Tian, “Protocol-Based Distributed Security Fusion Estimation for Time-Varying Uncertain Systems over Sensor Networks: Tackling DoS Attacks,” *IEEE Trans Signal Inf Process Netw*, vol. 10, pp. 119–130, 2024, doi: 10.1109/TSIPN.2024.3356789.
- [4] “Dell’Oro Group | Market Research.” Accessed: Nov. 27, 2025. [Online]. Available: <https://www.delloro.com/>
- [5] “Enabling the full value of connectivity - Ericsson.” Accessed: Nov. 27, 2025. [Online]. Available: <https://www.ericsson.com/en>
- [6] “OR-Tools | Google for Developers.” Accessed: Nov. 27, 2025. [Online]. Available: <https://developers.google.com/optimization?hl=es-419>
- [7] J. X. Sun, R. Y. Zhang, Q. F. Yu, H. R. Jiang, Z. Dai, and C. F. Zhou, “A Low-Profile High-Gain Patch Antenna with Double-Layer Metasurface,” *2022 International Conference on Microwave and Millimeter Wave Technology, ICMMT 2022 - Proceedings*, 2022, doi: 10.1109/ICMMT55580.2022.10023420.

- [8] W. Cui, X. Wang, M. Li, S. H. Pun, and F. Chen, “A Study of Deep Learning Based Classification of Mandarin Vowels Using Spoken Speech EEG Signals,” *Proceedings of 2023 IEEE International Conference on Signal Processing, Communications and Computing, ICSPCC 2023*, 2023, doi: 10.1109/ICSPCC59353.2023.10400255.
- [9] “O-RAN Specifications.” Accessed: Nov. 27, 2025. [Online]. Available: <https://www.o-ran.org/specifications>
- [10] “GSMA.” Accessed: Nov. 27, 2025. [Online]. Available: <https://www.gsma.com/>
- [11] “Ministerio de Telecomunicaciones y de la Sociedad de la Información – Ecuador.” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.telecomunicaciones.gob.ec/>
- [12] “Agencia de Regulación y Control de las Telecomunicaciones – Promovemos el desarrollo armónico del sector de las telecomunicaciones, radio, televisión y las TIC , mediante la administración y regulación eficiente del espectro radioeléctrico y los servicio...” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.arcotel.gob.ec/>
- [13] “IEEE Xplore Full-Text PDF:” Accessed: Nov. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10024837&utm_source=mendeley&getft_integrator=mendeley
- [14] M. Kalntis, G. Iosifidis, and F. A. Kuipers, “IEEE TRANSACTIONS ON COMMUNICATIONS 1 Adaptive Resource Allocation for Virtualized Base Stations in O-RAN with Online Learning”.
- [15] R. Sivaraj and S. Rajagopal, “O-RAN Architecture Overview,” *Open RAN: The Definitive Guide*, pp. 24–58, Jan. 2023, doi: 10.1002/9781119886020.CH3.

- [16] T. Mumtaz, S. Muhammad, and F. Bouali, “Formal Verification-and AI/ML-Assisted Radio Resource Allocation for Open RAN Compliant 5G/6G Networks”, doi: 10.1109/ACCESS.2025.3575021.
- [17] “Explicación de Open RAN | Nokia.com.” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.nokia.com/mobile-networks/ran/anyran/open-ran/open-ran-explained/>
- [18] R. Barker, “Virtualizing RAN: Science, Strategy, and Architecture of Software-Defined Mobile Networks”.
- [19] S. Marinova and A. Leon-Garcia, “Intelligent O-RAN Beyond 5G: Architecture, Use Cases, Challenges, and Opportunities”, doi: 10.1109/ACCESS.2024.3367289.
- [20] X. Foukas and B. Radunovic, “Concordia: Teaching the 5G vRAN to Share Compute; Concordia: Teaching the 5G vRAN to Share Compute”, doi: 10.1145/3452296.3472894.
- [21] “OpenRAN - Telecom Infra Project.” Accessed: Nov. 28, 2025. [Online]. Available: <https://telecominfraproject.com/openran/>
- [22] Y. Sai Rahul, “A Comprehensive Survey on the Future of Open Radio Access Networks (O-RAN): Architecture, Challenges, and Security Issues,” 2025. [Online]. Available: www.ijrpr.com
- [23] K. Alam *et al.*, “A Comprehensive Tutorial and Survey of O-RAN: Exploring Slicing-aware Architecture, Deployment Options, Use Cases, and Challenges”, doi: 10.1109/COMST.2025.3598406.

- [24] “Open RAN Market Size, Share, Growth, Industry Trend [Latest].” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.marketsandmarkets.com/Market-Reports/open-ran-market-153445936.html>
- [25] M. Gramaglia *et al.*, “Network Intelligence for Virtualized RAN Orchestration: The DAEMON Approach.” [Online]. Available: <https://h2020daemon.eu/>
- [26] “Background and Impact”.
- [27] A. Ahmed, A. A. Aburas, K. Al-Mashouq, and A. Aburas, “5G URLLC in Transport and Open RAN: Disaggregation RAN and Functional Splits,” *Proceedings - 2023 IEEE Future Networks World Forum: Future Networks: Imagining the Network of the Future, FNWF 2023*, 2023, doi: 10.1109/FNWF58287.2023.10520632.
- [28] “IEEE Xplore Full-Text PDF:” Accessed: Nov. 28, 2025. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=11017632>
- [29] “O-RAN Architecture Overview — oran master documentation.” Accessed: Nov. 30, 2025. [Online]. Available: <https://docs.o-ran-sc.org/en/latest/architecture/architecture.html>
- [30] R. Singh, C. Hasan, X. Foukas, M. Fiore, M. K. Marina, and Y. Wang, “Energy-Efficient Orchestration of Metro-Scale 5G Radio Access Networks CU CU CU CU CU CU CU CU CU CU CU CU DU DU DU DU DU DU DU DU DU DU DU DU FH Network FH Network FH Network”.
- [31] “3GPP – The Mobile Broadband Standard.” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.3gpp.org/>

- [32] “Functional Splits: the foundation of an Open 5G RAN - 5G Technology World.” Accessed: Nov. 30, 2025. [Online]. Available: <https://www.5gtechnologyworld.com/functional-splits-the-foundation-of-an-open-5g-ran/>
- [33] S. Danial Ali Shah, Z. Nezami, M. Hafeez, S. Ali Raza Zaidi, and S. Ali Raza Zaidi sazaidi, “The Interplay of AI-and-RAN: Dynamic Resource Allocation for Converged 6G Platform”.
- [34] M. Kalntis, G. Iosifidis, and F. A. Kuipers, “Adaptive Resource Allocation for Virtualized Base Stations in O-RAN With Online Learning,” *IEEE Transactions on Communications*, vol. 73, no. 3, pp. 1787–1800, 2025, doi: 10.1109/TCOMM.2024.3461569.
- [35] “The O-RAN ALLIANCE Security Working Group Continues to Advance O-RAN Security.” Accessed: Nov. 28, 2025. [Online]. Available: <https://www.o-ran.org/blog/the-o-ran-alliance-security-working-group-continues-to-advance-o-ran-security>
- [36] P. Filkins and G. Chen, “White Paper Empowering Telecom Operators to Deploy vRAN on Cloud and Edge Infrastructure,” 2021.
- [37] M. Z. Comiter, M. B. Crouse, and H. T. Kung, “A Data-Driven Approach to Localization for High Frequency Wireless Mobile Networks,” *Proceedings - IEEE Global Communications Conference, GLOBECOM*, vol. 2018-January, pp. 1–7, 2017, doi: 10.1109/GLOCOM.2017.8254732.
- [38] “GSMA.” Accessed: Dec. 11, 2025. [Online]. Available: <https://www.gsma.com/>
- [39] “Catálogo de servidores Dell PowerEdge: plataformas y soluciones para las aplicaciones de la empresa”.

- [40] “Los mejores servidores para grandes empresas en 2025.” Accessed: Dec. 13, 2025. [Online]. Available: <https://servermall.com/es/blog/mejor-servidor-para-grandes-empresas/>
- [41] S. Mukherjee, O. Coudert, and C. Beard, “An Open Approach to Autonomous Ran Fault Management,” *IEEE Wirel Commun*, vol. 30, no. 1, pp. 96–102, Feb. 2023, doi: 10.1109/MWC.004.2200244.
- [42] M. Q. Hamdan *et al.*, “Recent Advances in Machine Learning for Network Automation in the O-RAN,” *Sensors (Basel)*, vol. 23, no. 21, Oct. 2023, doi: 10.3390/s23218792.
- [43] D. Qiu, T. Munkhdalai, Y. He, and K. C. Sim, “Context-Aware Neural Confidence Estimation for Rare Word Speech Recognition,” *2022 IEEE Spoken Language Technology Workshop, SLT 2022 - Proceedings*, pp. 31–37, 2023, doi: 10.1109/SLT54892.2023.10023411.
- [44] “OR-Tools | Google for Developers.” Accessed: Aug. 24, 2025. [Online]. Available: <https://developers.google.com/optimization?hl=es-419>
- [45] “AN INTEL COMPANY vRAN: The Next Step in Network Transformation”.