

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo o titular de los Derechos de Autor.

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



---

**Maestría en Ciencias  
en Electrónica y Telecomunicaciones**

---

**Reconocimiento de patrones en métricas de tráfico para el  
cumplimiento de los SLA (Service Level Agreement) en entorno  
de redes B5G (Beyond 5G)**

Tesis  
para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:

**Joan David González Franco**

Ensenada, Baja California, México  
2023

Tesis defendida por  
**Joan David González Franco**

y aprobada por el siguiente Comité

---

**Dr. Jorge Enrique Preciado Velasco**  
Codirector de tesis

---

**Dr. José Eleno Lozano Rizk**  
Codirector de tesis

Miembros del Comité

**Dr. Raúl Rivera Rodríguez**

**Dr. Miguel Ángel Alonso Arévalo**

**Dr. Jorge Torres Rodríguez**



---

**Dra. María del Carmen Maya Sánchez**  
Coordinadora del Posgrado en Electrónica y Telecomunicaciones

---

**Dra. Ana Denise Re Araujo**  
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Joan David González Franco** como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones.

**Reconocimiento de patrones en métricas de tráfico para el cumplimiento de los SLA (Service Level Agreement) en entorno de redes B5G (Beyond 5G)**

Resumen aprobado por:

---

**Dr. Jorge Enrique Preciado Velasco**  
Codirector de tesis

---

**Dr. José Eleno Lozano Rizk**  
Codirector de tesis

Este trabajo de investigación tiene como objetivo principal desarrollar un sistema de gestión de calidad de servicio (QoS), para el cumplimiento de los Acuerdos al Nivel de Servicios (SLA), que se basa en la tecnología de redes definidas por software (SDN). Con esto en mente, se realizó un análisis exhaustivo de los requisitos y desafíos actuales en las redes de nueva generación, se consideró la creciente demanda de servicios y los estándares de calidad establecidos en el IMT-2020. En el marco de la investigación, y lograr el objetivo planteado, se propuso validar el sistema en dos fases, como son: implementar un clasificador de servicios que utiliza técnicas de aprendizaje de máquina (ML), el cual es capaz de identificar y seleccionar los parámetros de QoS adecuados para cada servicio demandado; crear una topología de red que cumple con la Arquitectura Basada en Servicios (SBA) de 5G en el entorno de simulación *Mininet*, donde se utiliza el controlador SDN *OpenDayLight* (ODL), para evaluar el rendimiento del sistema propuesto. Para cumplir con la primera fase, se implementó el clasificador en una API REST para la interacción con el sistema y se realizaron pruebas de funcionamiento. El clasificador de servicios logró una precisión del 97% en la identificación de los servicios demandados y se proporcionaron trayectorias que cumplen con los parámetros de QoS establecidos. Para la fase dos, mediante simulaciones, se encontraron múltiples trayectorias que cumplen con los requerimientos de QoS para cada solicitud de servicio, la cuales son posibles debido a la redundancia de la red. Además, se estableció un criterio de selección de trayectoria, que se basa en la ruta de menor latencia de extremo a extremo (E2E). Las simulaciones realizadas en *Mininet* evidenciaron el rendimiento del sistema en un entorno controlado y realista. Como principales conclusiones de esta investigación se destacan: la importancia de la gestión de QoS para el cumplimiento de los SLA en las redes de nueva generación, la efectividad del enfoque basado en SDN y ML para mejorar la experiencia del usuario, y la viabilidad de este estudio en entornos de simulación como *Mininet*.

**Palabras clave:** API REST, clasificador de servicios, ML, *Mininet*, QoS, SDN, SLA

Abstract of the thesis presented **by Joan David González Franco** as a partial requirement to obtain the Master of science degree in Electronic and Telecommunication.

**Recognition of patterns in traffic metrics for compliance with SLA (Service Level Agreement) in the B5G (Beyond 5G) network environment**

Abstract approved by:

---

**Dr. Jorge Enrique Preciado Velasco**  
Codirector de tesis

---

**Dr. José Eleno Lozano Rizk**  
Codirector de tesis

The main objective of this research work is to develop a quality of service (QoS) management system for compliance with Service Level Agreements (SLA) based on software-defined network (SDN) technology. With this in mind, we comprehensively analyzed the current requirements and challenges in new-generation networks, considering the growing demand for services and the quality standards established in IMT-2020. Within the framework of the investigation, and to achieve the stated objective, we proposed to validate the system in two phases, such as: implement a service classifier that uses machine learning (ML) techniques, which is capable of identifying and selecting the parameters adequate QoS for each requested service; create a 5G Service Based Architecture (SBA) compliant network topology in the Mininet simulation environment, using the SDN OpenDayLight (ODL) controller to evaluate the performance of the proposed system. In the first phase, we implemented a REST API service classifier to test the performance using the interaction with the system. The service classifier achieved an accuracy of 97% in identifying the demanded services and paths that comply with the established QoS parameters provided. For phase two, through simulations, multiple paths that meet the QoS requirements were found for each service request, which is possible due to network redundancy. In addition, we established a path selection criterion based on the lowest latency end-to-end (E2E) path. We conducted simulations using Mininet to evidence the system's performance in a controlled and realistic environment. The main conclusions of this research stand out: the importance of QoS management for compliance with SLAs in new generation networks, the effectiveness of the approach based on SDN and ML to improve the user experience, and the feasibility of the study in simulation environments such as Mininet.

**Keywords: API REST, service classifier, ML, Mininet, QoS, SDN, SLA**

**Dedicatoria**

*A mi familia.*

*Especialmente a mi  
muchachito, mi príncipe*

*Aimar Alain.*

*Te amo mucho.*

## Agradecimientos

A mis padres, que están lejos físicamente pero siempre en mi corazón. Mami, la verdad quería que estuvieras aquí presente, pero no pudo ser. Sin embargo, siempre me has dado todo tu apoyo y amor incondicional para seguir adelante, todo el tiempo pendiente en mis estudios, depositando tu confianza en mí, tu dedicación y sacrificio no tienen techo. Eres mi pilar para salir adelante, mi fuente de inspiración constante. A mi papá, por su especial dedicación conmigo, por su admiración y por estar siempre orgulloso de mí.

A mi esposa, Itzel, apenas un año de casados y hemos experimentado muchas emociones y alegrías. Tu amor, apoyo incondicional y paciencia me han ayudado mucho en este viaje desde el momento que te conocí durante el estudio de esta maestría. Hemos sido bendecidos con la llegada de nuestro hermoso bebé, Aimar, quien hizo de nosotros una familia con mucho amor. Te amo más de lo que pueda expresar en estas líneas.

A mis directores de tesis, por todo el apoyo recibido durante estos años de trabajo. Por siempre darme una mano cuando lo necesitaba, por confiar en mí. A José, por siempre estar dispuesto a brindar su ayuda sin poner peros y con una sonrisa cuando se requirió. La generosidad y disposición incondicional de su parte fueron de gran valor durante este proceso.

A Jorge en especial, mi más sincero agradecimiento, por seguir demostrando que la amistad no tiene fronteras de edad, por ayudarme en todo lo posible de forma incondicional en este proceso. Además, por fortalecer nuestro vínculo de amistad día tras día. Tu presencia y aliento constante me enseñan a mejorar cada vez más. Gracias por ser un amigo especial y hasta un padre la mayoría de las veces.

A mis sinodales, por siempre atender mis solicitudes, por aportar valiosas ideas de contribución, por darme la oportunidad de colaborar con ello, y por apoyarme cuando nació mi hijo.

A los profesores del CICESE y a la institución, por todos los conocimientos transmitidos durante estos dos años, así como por su disponibilidad y ayuda prestada en todo momento.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), por brindarme el apoyo económico para realizar mis estudios de maestría. No. CVU: 1163550.

## Tabla de contenido

	Página
Resumen en español.....	ii
Resumen en inglés.....	iii
Dedicatoria .....	iv
Agradecimientos.....	v
Lista de figuras.....	ix
Lista de tablas .....	xi
Glosario de términos .....	xii
<b>Capítulo 1. Introducción .....</b>	<b>1</b>
1.1 Planteamiento del problema .....	4
1.2 Antecedentes y justificación .....	4
1.3 Hipótesis.....	6
1.4 Objetivos .....	6
1.4.1 Objetivo general.....	6
1.4.2 Objetivos específicos.....	6
1.5 Estructura de la tesis .....	7
<b>Capítulo 2. Marco teórico .....</b>	<b>8</b>
2.1 Acuerdos a Nivel de Servicio (SLA) .....	8
2.1.1 Ciclo de vida de los SLA .....	10
2.2 Interfaz de Programación de Aplicaciones (API).....	12
2.2.1 API RESTful .....	13
2.3 Evolución a una Arquitectura Basada en Servicios (SBA).....	16
2.3.1 Redes Definidas por Software (SDN).....	17



2.3.1.1	Arquitectura de SDN.....	17
2.3.1.2	OpenFlow .....	20
2.3.2	Virtualización de Funciones de Red (NFV) .....	21
2.3.2.1	Arquitectura NFV.....	21
2.3.3	Network Slicing y SBA.....	23
2.4	Aprendizaje de Máquina (ML).....	28
2.4.1	Aprendizaje Supervisado (SL).....	32
2.4.2	Aprendizaje no Supervisado.....	34
2.4.3	ML en gestión de redes de nueva generación (5G/B5G) .....	36
2.5	Conclusiones.....	37
<b>Capítulo 3. Propuesta de un sistema para asignar recursos de red .....</b>		<b>38</b>
3.1	Esquema general de la propuesta.....	38
3.2	Solicitud de un servicio 5G/B5G .....	40
3.3	Clasificador de servicios 5G/B5G.....	42
3.4	Controlador SDN .....	43
3.5	Asignación de recursos.....	44
3.6	Recomendaciones para implementar el sistema de asignación de recursos de red .....	45
3.7	Conclusiones.....	51
<b>Capítulo 4. Validación del sistema propuesto .....</b>		<b>52</b>
4.1	Escenario de validación .....	52
4.2	Topología de red propuesta .....	56
4.3	Simulaciones del sistema propuesto.....	58
4.3.1	API del Clasificador de Servicios 5G/B5G .....	59
4.3.2	Escenario de topología de red en <i>Mininet</i> .....	62
4.4	Resultados obtenidos.....	66

4.5	Conclusiones.....	68
<b>Capítulo 5. Conclusiones y recomendaciones.....</b>		<b>70</b>
5.1	Contribuciones al conocimiento .....	71
5.2	Limitaciones de la investigación.....	72
5.3	Trabajo futuro .....	73
<b>Literatura citada.....</b>		<b>75</b>
<b>Anexos.....</b>		<b>81</b>

## Lista de figuras

Figura	Página
1. Ejemplos de servicios 5G y sus casos de uso genéricos (Vision & M.2083-0, 2015).....	2
2. Distribución de servicios 5G para KPIs de latencia y <i>jitter</i> (Preciado-Velasco, 2021). ....	4
3. Ciclo de vida de los SLA (Kapassa et al., 2019). ....	11
4. Funcionamiento de una API [Elaboración propia]. ....	13
5. Funcionamiento de una API REST [Elaboración propia].....	13
6. Arquitectura de la red móvil de 4G (5gAmericas, 2016). ....	16
7. Arquitectura de SDN (Millán Tejedor, 2014). ....	18
8. Arquitectura de la NFV (Juniper Network, 2023).....	22
9. Arquitectura de NS en 5G (Guan et al., 2018).....	24
10. Ilustración de NS en una topología de red (Wang et al., 2017). ....	25
11. SBA de 5G definida por 3GPP (3GPP, 2019). ....	26
12. Esquema general de ML (Borcoci, 2019).....	29
13. Fases de trabajo de un modelo de ML (Borcoci, 2019).....	30
14. Esquema general de un modelo de SL (Miguel, 2018).....	32
15. Esquema general de un modelo de aprendizaje no supervisado (Miguel, 2018).....	34
16. Esquema general propuesto [Elaboración propia]. ....	39
17. Esquema general del Clasificador de servicios 5G/B5G (Preciado-Velasco et al., 2021). ....	42
18. Topología de red propuesta [Elaboración propia]. ....	56
19. Entorno visual en POSTMAN. ....	60
20. Ejemplo de clasificación de un servicio Vo5G. ....	61
21. Comandos de confirmación de tipo 200 OK de las solicitudes a la API. ....	61
22. Visualización de un método GET para ver los servicios clasificados. ....	62
23. Topología de red simulada en <i>Mininet</i> [Elaboración propia].....	65
24. Respuesta de la API REST que muestra el servicio recién clasificado. ....	82

25. Respuesta de la API REST que muestra el servicio que más veces se ha clasificado. .... 83

## Lista de tablas

Tabla	Página
1. Comparativa de controladores SDN [Elaboración propia]. .....	46
2. Comparativa de plataformas para simular SDN [Elaboración propia]. .....	47
3. Comparativa de los principales proveedores de servicios en la nube. ....	49
4. Nombre de los hosts .....	63
5. Características de QoS de cada enlace de los switches al servicio o UE de la topología [Elaboración propia]. .....	64
6. Características de QoS de cada enlace de los switches de la topología [Elaboración propia].	64
7. Resultados de las simulaciones en el entorno de <i>Mininet</i> [Elaboración propia]. .....	67
8. Especificaciones de cada <i>endpoint</i> de la API REST [Elaboración propia]. .....	81
9. Solicitud de UE a servicios deUHD video <i>streaming</i> , videovigilancia, experiencias inmersivas [Elaboración propia]. .....	84
10. Solicitud de UE a servicios de ITS o Vehículos conectados [Elaboración propia]. ....	85
11. Solicitud de UE a servicios de <i>eHealth</i> [Elaboración propia]. .....	86
12. Solicitud de UE a servicios de automatización industrial, redes inteligentes [Elaboración propia]. .....	87
13. Solicitud de UE a servicios de Vo5G [Elaboración propia]. .....	88

## Glosario de términos

### #

3GPP	<i>Third Generation Partnership</i>	Proyecto de Asociación de 3ra Generación
4G	<i>4th Generation</i>	4ta Generación
5G	<i>5th Generation</i>	5ta Generación

### A

AI	<i>Artificial Intelligence</i>	Inteligencia Artificial
ANN	<i>Artificial Neural Networks</i>	Redes Neuronales Artificiales
API	<i>Application Programming Interface</i>	Interfaz de Programación de Aplicaciones
AR	<i>Augmented Reality</i>	Realidad Aumentada

### B

BS	<i>Base Station</i>	Estación Base
----	---------------------	---------------

### C

CN	<i>Core Network</i>	Red de núcleo
----	---------------------	---------------

### D

DL	<i>Down Link</i>	Enlace Descendente
DT	<i>Decision Tree</i>	Árbol de Decisión

### E

E2E	<i>End to End</i>	Extremo a Extremo
eMBB	<i>Enhanced Mobile Broadband</i>	Banda Ancha Móvil Mejorada
ETSI	<i>European Telecommunications Standards Institute</i>	Instituto Europeo de Normas de Telecomunicaciones

### G

Gbps	<i>Gigabit per second</i>	Gigabit por segundo
------	---------------------------	---------------------

### I

IDE	<i>Integrated Development Environment</i>	Entorno de Desarrollo Integrado
IEEE	<i>Institute of Electrical and Electronics Engineers</i>	Instituto de Ingeniería Eléctrica y Electrónica
IMT-2020	<i>International Mobile Telecommunications 5th Generation</i>	Telecomunicaciones Móviles Internacionales de 5ta Generación
IoT	<i>Internet of Things</i>	Internet de las Cosas
ITS	<i>Intelligent Transport Systems</i>	Sistemas de Transporte Inteligente
ITU	<i>International Telecommunications Union</i>	Unión Internacional de Telecomunicaciones

<b>J</b>		
<i>Jitter</i>		Fluctuación del retardo (Variabilidad del tiempo de ejecución de los paquetes)
JSON	<i>JavaScript Object Notation</i>	Notación de Objetos de <i>JavaScript</i>
<b>K</b>		
KNN	<i>k-Nearest Neighbors</i>	k-Vecinos más Cercanos
KPI	<i>Key Performance Indicators</i>	Indicadores Clave de Desempeño
KQI	<i>Key Quality Indicators</i>	Indicadores Clave de Calidad
<b>L</b>		
LTE	<i>Long-Term Evolution</i>	Evolución a largo plazo
<b>M</b>		
ML	<i>Machine Learning</i>	Aprendizaje automático o aprendizaje de máquinas
mMTC	<i>Massive Machine Type Communication</i>	Comunicación Masiva Tipo Máquina
MLPC	<i>Multilayer Perceptron Classifier</i>	Clasificador Perceptrón Multicapa
ms	<i>milisecond</i>	milisegundos
<b>N</b>		
NGMN	<i>Next Generation Mobile Networks</i>	Redes Móviles de Próxima Generación
NF	<i>Network Function</i>	Función de Red
NFV	<i>Network Function Virtualization</i>	Virtualización de Funciones de Red
NFVO	<i>NFV orchestrator</i>	Orquestador de NFV
NG-RAN	<i>New Generation RAN</i>	RAN de próxima generación
NS	<i>Network Slice</i>	Segmento de red (red virtual)
<b>O</b>		
ONF	<i>Open Networking Foundation</i>	Fundación de Redes Abiertas
OSS	<i>Operations Support Systems</i>	Sistema de Soporte de Operaciones
<b>Q</b>		
QoS	<i>Quality of Service</i>	Calidad de Servicio
QoE	<i>Quality of Experience</i>	Calidad de Experiencia
<b>R</b>		
RAN	<i>Radio Access Network</i>	Red de Acceso de Radio
REST	<i>Representational State Transfer</i>	Transferencia de Estado Representacional
<b>S</b>		
SBA	<i>Service Based Architecture</i>	Arquitectura Basada en Servicios
SDN	<i>Software Defined Networks</i>	Redes Definidas por Software
SLA	<i>Service Level Agreement</i>	Acuerdo de Nivel de Servicio
SLO	<i>Service Level Objectives</i>	Objetivo de Nivel de Servicio

SVM	<i>Support vector machines</i>	Máquinas de vectores de soporte
<b>T</b>		
TPU	<i>Tensor Processing Unit</i>	Unidad de procesamiento tensorial
<b>U</b>		
UE	<i>User Equipment</i>	Equipo de Usuario
UHD	<i>Ultra High Definition</i>	Ultra Alta Definición
UL	<i>Up Link</i>	Enlace Ascendente
urLLC	<i>Ultra-reliable and Low Latency Communication</i>	Comunicaciones UltraConfiables y de Baja Latencia
<b>V</b>		
VM	<i>Virtual Machine</i>	Máquina Virtual
VNF	<i>Virtual Network Functions</i>	Funciones de Red Virtualizadas
Vo5G	<i>Voice over 5G</i>	Voz sobre 5G
VR	<i>Virtual Reality</i>	Realidad Virtual



## Capítulo 1. Introducción

---

Las comunicaciones móviles han cambiado profundamente la vida cotidiana, y el deseo de las personas por una comunicación móvil de mayor rendimiento nunca termina. Las redes móviles de nueva generación 5G/B5G (por las siglas de los términos en inglés, *5th Generation / Beyond 5G*) permiten a los usuarios nuevos servicios incluyendo: Realidad Aumentada, Realidad Virtual (AR, VR, por las siglas de los términos en inglés, *Augmented Reality* y *Virtual Reality*; respectivamente), Internet de las Cosas (IoT, por las siglas del término en inglés, *Internet of Things*), vehículos autónomos y muchos más. Todo esto gracias a que proporciona avances significativos en cuanto a los servicios y la gestión de red comparado con las infraestructuras móviles tradicionales anteriores (Barona López et al., 2017).

La Unión Internacional de Telecomunicaciones (ITU, por las siglas del término en inglés, *International Telecommunications Union*), llegó a un consenso de que los sistemas inalámbricos 5G/B5G soportan tres casos de uso genéricos (Vision & M.2083-0, 2015):

- Banda Ancha Móvil Mejorada (eMBB, por las siglas del término en inglés, *Enhanced Mobile Broadband*): admite conexiones estables con velocidades de datos pico muy altas.
- Ultra-baja latencia y de alta confiabilidad (urLLC, por las siglas del término en inglés, *Ultra-reliable and Low Latency Communication*): donde un conjunto limitado de terminales soporta transmisiones de paquetes de cargas útiles pequeñas que requieren de muy baja latencia y con alta confiabilidad.
- Comunicación masiva tipo máquina (mMTC, por las siglas del término en inglés, *massive Machine Type Communication*): el cual admite un masivo número de dispositivos conectados, que solo están activos esporádicamente y que envían una pequeña carga de datos.

Servicios 5G típicos, como, por ejemplo, AR y VR, los videos de Ultra-Alta Definición (UHD, por las siglas del término en inglés, *Ultra High Definition*) y el hogar inteligente, por mencionar algunos. En la figura 1 se observan algunos servicios 5G y los respectivos casos de uso genéricos con los que se relacionan. Un servicio puede tener requerimientos diversos en términos de Indicadores Claves de Rendimiento y Calidad (KPI y KQI, por las siglas de los términos en inglés, *Key Performance Indicators* y *Key Quality Indicators*; respectivamente), porque puede poseer características de dos casos de uso genéricos como, por ejemplo,

AR que se encuentra en un término medio entre eMBB y urLLC (ver figura 1). Sin embargo, otros servicios pertenecen a un caso de uso en específico, por ejemplo, ciudad inteligente pertenece a mMTC.

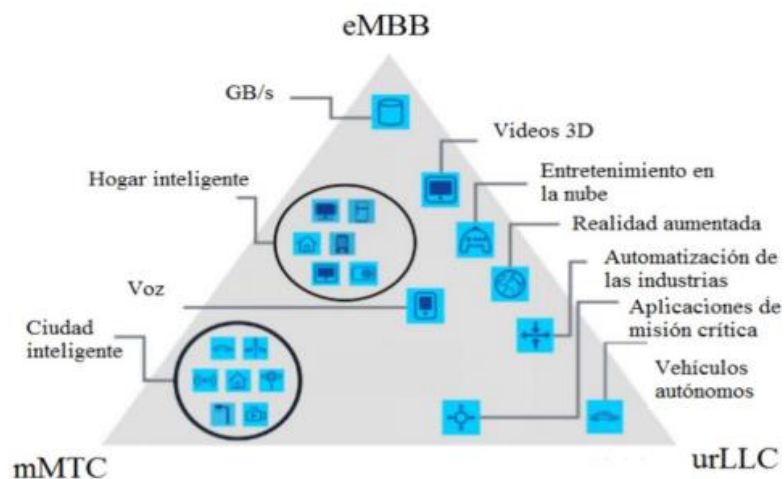


Figura 1 Ejemplos de servicios 5G y sus casos de uso genéricos (Vision & M.2083-0, 2015).

Las redes de nueva generación tienen una gran cantidad de parámetros involucradas en su funcionamiento, así como una variación en los patrones de comportamiento de la propia red, lo que limita a identificar la actividad de la red (5gAmericas, 2016; Vision & M.2083-0, 2015). Esto hace que el modelo tradicional de gestión sea insuficiente, ya que la correlación entre los extensos datos de Calidad de Servicio (QoS, por las siglas del término en inglés, *Quality of Service*) como latencia, *jitter*, pérdida de paquetes, ancho de banda, entre otros; no resulta trivial de hacer sin la ayuda computacional (5gAmericas, 2016; Mullins & Barros, 2017).

La gestión de redes de nueva generación no es un esfuerzo fácil, los retos son aún mayores a los que se ha enfrentado la gestión de las redes inalámbricas actuales (Group, 2015). La complejidad, flexibilidad y dinamismo de las redes 5G requiere que la red en sí sea gestionada de un modo automatizado. Por lo que se requiere se apoye en técnicas de Inteligencia Artificial, Aprendizaje de Máquina, Redes Definidas por Software y Virtualización de Funciones de Red (AI, ML, SDN y NFV; por las siglas de los términos en inglés, *Artificial Intelligence*, *Machine Learning*, *Software Defined Network*, *Network Function Virtualization*; respectivamente) que permitan hacer la gestión del desempeño de los recursos de red (Group, 2015; Popovski et al., 2018; Telecommunication & Maarten, 2015).

La capacidad de conocer la relación entre los patrones de servicios-red es cada vez más crítica, ya que se plantea manejar latencias cada vez más bajas, en el orden de ms y velocidades de transferencia de datos muy altas (Gbps). Estos requerimientos hacen aún más exigente analizar los servicios y los tráficos a través de la red para poder tomar acciones que lleven a un mejor aprovisionamiento de los recursos de red por parte del proveedor de servicios (M. Jiang et al., 2017; Popovski et al., 2018).

Un operador de red necesita poder clasificar los servicios que los usuarios soliciten. No se trata solo de identificar el servicio, se debe conocer, además, los KPI y KQI de éste. Con ello, estar en posibilidad de satisfacer las necesidades del usuario, mejorar la mejor QoS ofrecida y así, inferir en una mejor Calidad de Experiencia (QoE, por las siglas del término en inglés, *Quality of Experience*) del usuario en cuestión (5gAmericas, 2016; Unión Internacional de Telecomunicaciones ITU, 2018). La QoE está directamente relacionada con la QoS, pero para los proveedores de servicios constituye un reto tener el conjunto adecuado de herramientas y procesos para asignar la QoS en el nivel de red a la QoE en los niveles de usuario (Telecommunication & Maarten, 2015).

En la figura 2, se muestra un mapa de dispersión de un conjunto de servicios repartidos en un plano en función de dos variables importantes en términos de *performance* de la red: latencia y el *jitter*. Se observa que cuando los requerimientos de latencia no son tan exigentes (mayor de 60 ms) es relativamente fácil clasificar o identificar de qué servicio se trata (Vo5G, Vigilancia en video, ITS). Sin embargo, cuando los requerimientos se hacen más exigentes (latencias menores de 60 ms y *jitter* menos de 15 ms) se dificulta clasificar el servicio, ya que se encuentran agrupados por las características tan similares que presentan. Estos requerimientos hacen aún más exigente analizar las características de los servicios y los tráficos generados a través de la red para poder tomar acciones que lleven a un mejor aprovisionamiento de los recursos de red por parte del proveedor de servicios.

Debido a la heterogeneidad y requerimientos que tienen los servicios 5G/6G, para un mejor soporte de la infraestructura de la red y una mejor QoS/QoE (Unión Internacional de Telecomunicaciones ITU, 2018), el operador requiere de técnicas de AI/ML para detectar e identificar los patrones en la red relacionados con los servicios. Además, necesita tener un mecanismo de gestión de SDN que sea capaz de redireccionar el tráfico y, en caso de ser necesario cambiar las rutas. Con este conocimiento se hace posible el establecimiento de Segmentos de Red (NS, acrónimo por las siglas del término en inglés, *Network Slices*) que cumplan con los Acuerdos a Nivel de Servicio (SLA, por las siglas del término en inglés, *Service Level Agreement*) y la automatización en todos los niveles (Barona López et al., 2017; HUAWEI TECHNOLOGIES, 2018; Mullins & Barros, 2017).

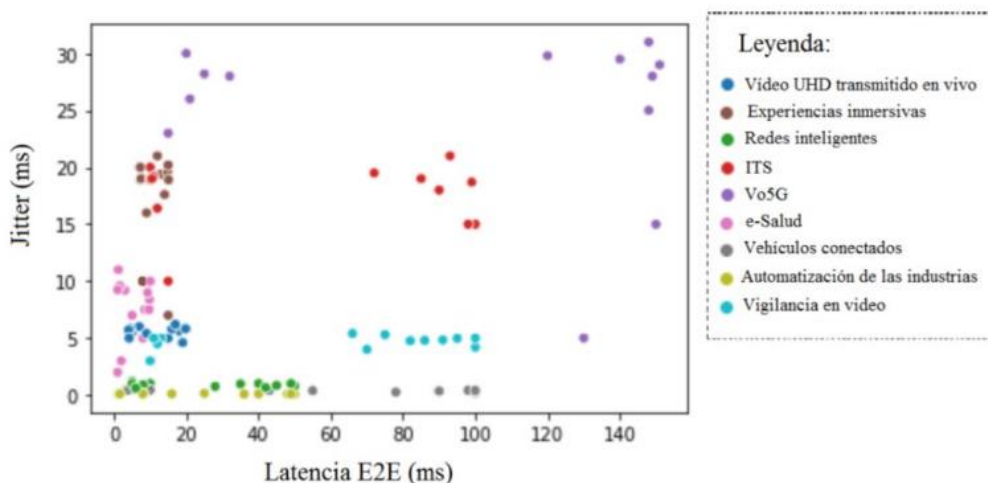


Figura 2 Distribución de servicios 5G para KPIs de latencia y *jitter* (Preciado-Velasco, 2021).

## 1.1 Planteamiento del problema

En la actualidad, las redes de nueva generación han evolucionado para proporcionar una gran variedad de servicios con diferentes requerimientos de QoS. Para garantizar que estos servicios se entreguen de manera efectiva, se establecen SLA que cumplan las condiciones y garantías asociadas a cada servicio. Sin embargo, cumplir con estos contratos sigue siendo un gran desafío para la gestión de recursos de red, especialmente en entornos de alta demanda. La sobreasignación o subasignación de recursos de red puede llevar a la violación de los contratos de SLA, lo que afecta negativamente la experiencia de los usuarios y reduce la eficiencia de la red.

En este sentido, el problema a resolver en esta investigación es:

- ✓ La falta de mecanismos de gestión automatizada de redes de nueva generación que intervengan en la violación de los contratos de los SLA.

## 1.2 Antecedentes y justificación

La clasificación de servicios en redes de nueva generación es un tema de gran importancia en la actualidad, ya que permite optimizar el rendimiento y la calidad de servicio de las redes de comunicación. En la literatura existen diversos trabajos que abordan este problema desde diferentes perspectivas, incluyendo técnicas de clasificación basadas en algoritmos de ML, redes neuronales, heurísticas, entre otras. A

continuación, se presentan algunos de los trabajos más relevantes en este campo, con el objetivo de establecer un estado del arte y contextualizar el enfoque propuesto en esta tesis.

En (Ibarra-Lancheros et al., 2018) los autores evaluaron parámetros de QoS con el empleo de SDN y un enfoque de reglas para determinar cómo formar los NS. Para ello se creó una topología básica en *Mininet*, con un controlador SDN *Floodlight* para tomar las decisiones respecto a la asignación de los recursos de red.

En (Chen et al., 2020) se elaboraron una serie de experimentos basados en la asignación de recursos con el empleo de programas computacionales de libre distribución (*open software*) como: *OpenStack* (como plataforma en la nube), *Tacker* para administrar las Funciones de Red Virtualizadas (VNF, por las siglas del término en inglés, *Virtual Network Function*) requeridas y el conmutador *OpenFlow*. Se presentan resultados que demuestran que la segmentación es factible y el ancho de banda se puede asignar usando colas. Como resultado se garantiza la QoS para cada segmento.

Los autores en (Vale Saraiva et al., 2021) utilizan una SDN vehicular para implementar un enfoque de segmentos 5G, abordan el problema de cómo proporcionar una infraestructura móvil que pueda cumplir dinámicamente con los requisitos de comunicación de diferentes aplicaciones de Sistemas de Transporte Inteligente (ITS, por las siglas del término en inglés, *Intelligent Transport Systems*) en redes vehiculares. Asimismo, presentan una arquitectura con los algoritmos de diseño para tráfico heterogéneo en un entorno vehicular dinámico.

En (Ramos García, 2021) el autor diseña e implementa dos arquitecturas de redes neuronales artificiales: una red recurrente con arquitectura codificador-decodificador, y una red neuronal mixta compuesta por la unión de una red convolucional y una red recurrente. Obtiene como resultado la predicción del *throughput* de la red 5G para una aplicación del segmento eMBB, considerando los indicadores de desempeño y la creación de un sistema de predicción de la calidad del servicio que garantice el cumplimiento de los SLA.

Los autores de (Y.-J. Wu et al., 2022) proponen un marco de SDN con conmutadores de procesador de paquetes (P4) para generar segmentos de red de mMTC y urLLC. Para ello, crean una topología de red básica en *Mininet*, donde asignan características de QoS a los enlaces y luego a través de la identificación del tráfico generan los NS para decidir la ruta de envío de paquetes.

Las distintas variantes que se emplean en los trabajos anteriores hacen una valiosa contribución al campo de la gestión de redes, proponen diversas técnicas y enfoques que permiten mejorar el rendimiento y la QoS. Sin embargo, la mayoría de estos trabajos no contemplan la asignación de recursos de red para prevenir violaciones de los SLA. Por lo tanto, en esta tesis se propone un enfoque que aborda este problema mediante la asignación de recursos de red de manera dinámica y adaptativa, con el objetivo de ayudar a mejorar la QoS, haciendo más precisa la determinación y cumplimiento de los SLA de los clientes.

### 1.3 Hipótesis

En el proceso de asignar recursos de red a través de instancias de NS, es posible emplear algoritmos de ML para clasificar servicios, así como técnicas de SDN para mejorar la QoS brindada por el operador, así como el cumplimiento y no violación de los SLA.

### 1.4 Objetivos

#### 1.4.1 Objetivo general

Mejorar la asignación de recursos de red con parámetros de QoS mediante la creación de NS que cumplan con los SLA inherentes.

#### 1.4.2 Objetivos específicos

- ✓ Utilizar técnicas de AI/ML para el reconocimiento de parámetros de servicios y aplicaciones.
- ✓ Asignar los *networks slices* del servicio acorde a los parámetros del SLA.
- ✓ Diseñar una cama de pruebas (*testbed*) con los escenarios (casos de uso) elegidos.

## 1.5 Estructura de la tesis

Este trabajo de maestría presenta la siguiente estructura: introducción, cinco capítulos, recomendaciones, referencias bibliográficas, glosario de términos y anexos. En los párrafos siguientes, se detalla la estructura de este trabajo y cómo se encuentran organizados los capítulos restantes:

En el **Capítulo 2** se resume el estado del arte sobre los SLA, su gestión y la segmentación de las redes 5G/B5G. Posteriormente, se expone SDN, NFV y la evolución de la arquitectura de red móvil, de 4G a la Arquitectura Basada en Servicios (SBA, por las siglas del término en inglés, *Service Based Architecture*) de 5G. Más adelante, se explica el papel que desempeña la Interfaz de Programación de Aplicaciones (API, por las siglas del término en inglés, *Applications Programming Interface*) en 5G. Para finalizar, se analiza la aplicación de algoritmos de ML en la gestión de redes, enfocados a la mejora de la QoS y a garantizar el cumplimiento de los SLA.

En el **Capítulo 3** se propone un esquema de conexión entre una API que clasifica servicios 5G/B5G, y con ello asignar los parámetros de QoS que corresponda al servicio y permitan formar los NS, con apoyo de aprendizaje supervisado; y la API de un controlador SDN. Se detallan cada uno de los bloques de este esquema, con especial énfasis a las APIs, del clasificador y del controlador; por su importancia en la propuesta. Posteriormente se plantean un conjunto de recomendaciones para implementar el sistema que se propone. En el **Capítulo 4** se define el escenario para la validación del sistema propuesto, que contribuye a mejorar la QoS y la no violación de los SLA. Se emplea el software *Mininet* para la simulación como método de validación, que permite emular topologías de red y modificar los enlaces a través de un controlador de SDN.

En el **Capítulo 5** se concluye este trabajo de tesis, se presentan las contribuciones al conocimiento y las limitaciones. Además, se plantean las recomendaciones para el trabajo futuro asociado a la investigación. Para complementar y ampliar la información expuesta en este trabajo, se elaboraron dos anexos que contienen información relevante sobre el código generado para poder simular el sistema propuesto, así como algunas salidas de las corridas realizadas. Estos anexos se citan en el desarrollo de este trabajo, para una mejor explicación y comprensión de éste.

En este trabajo de tesis, las figuras se dejan en formato original, por un lado, cuidar su entendimiento y aplicación (cuidar los derechos de autor) y, por otro lado, evitar así una posible pérdida de información en la traducción.

## Capítulo 2. Marco teórico

---

Los Acuerdos de Nivel de Servicio (SLA) son un elemento fundamental en la gestión de redes de nueva generación. En la arquitectura de 5G, la tecnología de NS permite crear redes virtuales que se diseñan a las necesidades específicas de los usuarios y aplicaciones, lo que ha llevado a un aumento en la complejidad de la gestión de los SLA. En este contexto, la SBA de 5G ofrece un enfoque flexible y modular para la construcción y gestión de redes de nueva generación, y las Interfaces de Programación de Aplicaciones (API) son una parte esencial de esta arquitectura. Además, el uso de técnicas de ML puede ayudar a mejorar la gestión de los SLA y la optimización de los recursos de red.

El presente capítulo tiene como objetivo presentar el marco teórico necesario para el desarrollo de la propuesta de esta tesis. Para ello, se abordarán diversos temas relacionados con las redes de nueva generación, desde la gestión de los SLA hasta el uso de técnicas de ML para mejorar la gestión de recursos de red. En primer lugar, se presentará una breve introducción sobre qué son los SLA, su importancia y su gestión en entornos de redes de nueva generación. Luego, se explicará la evolución de la red móvil 4G hasta la Arquitectura Basada en Servicios (SBA) de 5G, con el objetivo de comprender las diferencias y ventajas de la última. Posteriormente, se aborda el concepto y función de API RESTful en la arquitectura de 5G. Para finalizar, se brinda un panorama de ML y sus principales técnicas de aprendizaje, así como la aplicación de técnicas de ML en la gestión de la red 5G.

### 2.1 Acuerdos a Nivel de Servicio (SLA)

Un SLA es un acuerdo o contrato escrito entre un proveedor de servicio y su cliente con la esencia de fijar el nivel acordado para la calidad que debe cumplir dicho servicio. El SLA es una herramienta que ayuda a ambas partes (proveedor y cliente) a llegar a un consenso en términos del nivel de QoS, en aspectos tales como tiempo de respuesta, disponibilidad horaria, documentación disponible, personal asignado al servicio, etc. (Forum & Group, 2004).

Los SLA identifican y definen las necesidades del cliente a la vez que controla sus expectativas de servicio con relación a la capacidad del proveedor. Proporcionan un marco de entendimiento, simplifican asuntos complicados, reducen las áreas de conflicto y favorecen el diálogo ante la disputa. También constituyen un punto de referencia para el proceso de mejora continua, ya que el poder medir adecuadamente los



niveles de servicio es el primer paso para mejorarlos y de esa forma aumentar los índices de desempeño y calidad, KPI, KQI, etc. (Kapassa et al., 2018).

Entre los aspectos del sistema 5G se encuentran los procedimientos en torno a la gestión y la aplicación de SLA. En el contexto de los sistemas móviles 5G, los operadores de red proporcionan un segmento de red (*network slice, NS*) de extremo a extremo como servicio, que tiene asociado un SLA según los requisitos del cliente (ETSI, 2018; Papageorgiou et al., 2020).

Los requerimientos para establecer SLA en 5G están en función de los servicios provistos y de la infraestructura del proveedor. Por lo que es imperativo considerar indicadores claves de desempeño (KPI) así como los de calidad (KQI) entre otros (Yousaf, 2016), ya que la QoS, base para el establecimiento del SLA, está en función tanto de las aplicaciones como de la red, así como de factores subjetivos como es la experiencia de usuario (QoE) (Kapassa et al., 2018).

5G ha impactado en los ecosistemas asociados a la red móvil que conocemos. Las garantías necesarias para la QoS se pueden maximizar con las capacidades de las Funciones de Red Virtualizadas (VNF, por las siglas del término en inglés, *Virtual Network Functions*) y los servicios de red. Esto implica la utilización de SLA para garantizar que los servicios de red se proporcionen de manera eficiente y controlada (Kapassa et al., 2018; Sama et al., 2016).

Sin embargo, la complejidad de la tarea de determinar las políticas de provisión de recursos en entornos multimodales, así como las características y propiedades de varias VNF y servicios, da como resultado SLA personalizados que no consideran todos los aspectos del entorno 5G (Kapassa et al., 2018; Sama et al., 2016). De lo anterior surge la necesidad de un marco de gestión de SLA para mapear los requisitos de los usuarios (*high-level requirements*) a los recursos necesarios de la red (*low-level requirements*), para mejorar la capacidad del proveedor de servicios y cumplir así con los correspondientes compromisos de SLA.

La heterogeneidad de la red 5G es una de las principales razones por las que la implementación de SLA se vuelve más compleja. Los diferentes tipos de servicios que se ofrecen como: VR, aplicaciones multimedia y el IoT; tienen características y requerimientos disímiles de QoS. Esto implica que los proveedores de servicios de red deben ser capaces de personalizar los SLA en función de las necesidades específicas de cada servicio y cliente. Los proveedores de servicios deben poder garantizar la QoS en tiempo real,

adaptándose a los cambios en la demanda de los clientes y en las condiciones de la red (Kapassa et al., 2018).

Otro aspecto que aumenta la complejidad de la implementación de SLA en 5G es la necesidad de monitorear y medir continuamente el rendimiento de la red y de los servicios ofrecidos. Esto implica el uso de herramientas de monitoreo y medición que permitan a los proveedores de servicios de red recopilar y analizar los datos en tiempo real. Deben estar preparados para identificar y solucionar rápidamente los problemas de la red y minimizar el impacto en los servicios para cumplir con los compromisos de SLA acordados.

La implementación de un enfoque de ML para la detección de anomalías en la red y la predicción de fallas también puede ser útil para aumentar la eficiencia y reducir los tiempos de respuesta en la solución de problemas de la red y, en la garantía de la QoS. En resumen, la implementación de SLA en el contexto de 5G es crucial para garantizar la QoS y la QoE en una red cada vez más heterogénea y dinámica, aunque implica una complejidad adicional que debe ser abordada adecuadamente.

### 2.1.1 Ciclo de vida de los SLA

En el ámbito de las telecomunicaciones, los SLA son esenciales para garantizar la QoS/QoE suministrados. Para ello, es necesario establecer un ciclo de vida de los SLA que abarque desde su diseño y negociación hasta su monitoreo, cumplimiento y terminación. El ciclo de vida de los SLA proporciona un marco para la gestión y mejora continua de la calidad del servicio prestado y permite a los proveedores de servicios mantener un enfoque orientado al cliente.

En este sentido, es fundamental considerar que el ciclo de vida de los SLA debe estar diseñado de manera flexible y adaptable a los constantes cambios que se presentan en la red de telecomunicaciones. En la era del 5G, donde la complejidad de la red y la variedad de servicios ofrecidos aumenta, es importante contar con un ciclo de vida de los SLA que sea escalable y permita una gestión eficiente de los mismos.

El ciclo de vida del SLA en el dominio 5G es una parte importante de la provisión de servicios, el cual está totalmente alineado con los principios de 5G y se ejecuta en paralelo con el ciclo de vida del servicio de red (Kapassa et al., 2019; Van Rossem et al., 2017). Se administra por las plataformas de servicios 5G que

lo acompañan y es un proceso dinámico que comprende cuatro etapas clave (Kapassa et al., 2018, 2019): (ver figura 3).



**Figura 3 Ciclo de vida de los SLA** (Kapassa et al., 2019).

1. **Arquitectura:** comienza con la selección de un servicio de red y la definición de requisitos por parte del desarrollador. Normalmente, el operador es el responsable de examinarlos, tener en cuenta necesidades comerciales importantes e implementar plantillas de SLA, como una oferta inicial a los clientes de un servicio.
2. **Compromiso:** la selección de diferentes servicios de red es el resultado de aspectos comerciales, que son la base para diferentes restricciones de QoS, que también se pueden definir como requisitos del acuerdo. La preferencia de un operador o proveedor de servicios de red depende del servicio de red deseado, sus características y restricciones presupuestarias. Las expectativas de QoS, hace que los usuarios negocien con sus operadores o proveedores de servicios los niveles de QoS. Después de un proceso de negociación exitoso, se crea un SLA para describir los parámetros de QoS acordados.
3. **Operación:** comprende el despliegue real del servicio, el establecimiento de canales de comunicación y actividades operativas adicionales. Además, en esta fase se monitorea datos en tiempo real, con el propósito de evitar o gestionar infracciones inesperadas de los SLA.
4. **Terminación:** se ocupa del final de la relación entre el operador o proveedor de servicios y el cliente, incluido el fin de la relación jurídica. Esta última fase incluye la evaluación de alternativas, compromisos de liquidación y terminación, exportación de datos, atención al cliente y diligencia,

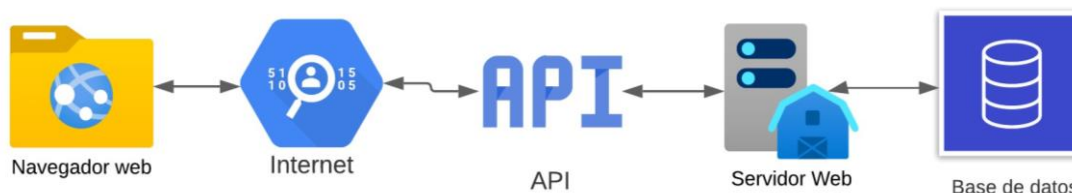
y supresión de datos. Todo lo anterior debe considerarse si el servicio de red fue rescindido, o producto de la violación del SLA.

## 2.2 Interfaz de Programación de Aplicaciones (API)

Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. En el contexto de las API, la palabra aplicación se refiere a cualquier software con una función distinta. La interfaz puede considerarse como un contrato de servicio entre dos aplicaciones. Este contrato define cómo se comunican entre sí mediante solicitudes y respuestas. La arquitectura de las API suele explicarse en términos de cliente y servidor. La aplicación que envía la solicitud se llama cliente, y la que envía la respuesta se llama servidor. En los últimos años, las API se han convertido en una herramienta fundamental en el desarrollo de software y en la integración de sistemas, permitiendo la creación de aplicaciones de manera rápida y eficiente (Amazon Web Services (AWS), 2023; Lozano-Rizk et al., 2022)

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de conocer los detalles técnicos de su implementación. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API otorgan flexibilidad, simplifican el diseño, administran el uso de las aplicaciones y ofrecen oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales). Como se mencionó anteriormente, en ocasiones las API se pueden considerar como contratos, con documentación que representa un acuerdo entre las partes; si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte (Demircioğlu & Kalipsiz, 2022; Lozano-Rizk et al., 2022).

Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos. Las API públicas aportan un valor comercial único porque simplifican y amplían sus conexiones con los *partners* y, además, pueden rentabilizar sus datos (RedHat, 2023b). El principio de funcionamiento de una API se expresa comúnmente a través de la comunicación de solicitud-respuesta entre un cliente y un servidor. El cliente es cualquier aplicación frontal con la que interactúa un usuario. El servidor está a cargo de la lógica de *back-end* y las operaciones de la base de datos. En este escenario, una API funciona como una capa intermedia entre el cliente y el servidor, lo que permite enviar solicitudes y respuestas de datos. Este procedimiento se describe en la figura 4.

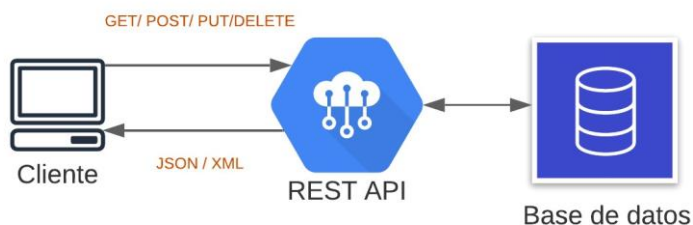


**Figura 4 Funcionamiento de una API [Elaboración propia].**

### 2.2.1 API RESTful

La Transferencia de Estado Representacional (REST, por las siglas del término en inglés, *Representational State Transfer*) es la arquitectura utilizada para diseñar servicios empleado en diferentes plataformas y entornos para admitir la interoperabilidad y la WWW (*World Wide Web*) (L. Li et al., 2016). REST no es un protocolo ni un estándar, sino más bien un conjunto de límites de arquitectura. Los desarrolladores de las API pueden implementarlo de distintas maneras (Baniş et al., 2021). A través de los años, se ha convertido en una forma estandarizada ampliamente seguida de publicar servicios a través de Internet (Neumann et al., 2021). Las API REST son una parte importante del diseño de los microservicios (Pahl & Jamshidi, 2016) y a través de los años, han ganado una gran popularidad debido a su enfoque en la arquitectura de software orientada a servicios.

Las API REST, o RESTful, ampliamente conocidas como API web, consisten en puntos finales que se ajustan a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. Estas API generalmente son accesibles a través de Protocolo de Transferencia de Hipertexto (HTTP, por las siglas del término en inglés, *Hypertext Transfer Protocol*) para la comunicación entre sistemas y se basan en una arquitectura cliente-servidor sin estado. Esto permite una mayor eficiencia y escalabilidad en la comunicación entre sistemas al incluir verbos estándar definidos como GET (para leer), POST (crear), PUT (editar) y DELETE (borrar) (Pahl & Jamshidi, 2016) (ver figura 5).



**Figura 5 Funcionamiento de una API REST [Elaboración propia].**

Uno de los desafíos fue definir un formato estándar de mensajería (es decir, solicitud y respuesta). Inicialmente, se utilizó texto informal para describir las API REST (Pautasso et al., 2008). Los documentos de tipo Notación de Objeto de JavaScript (JSON, por las siglas del término en inglés, *JavaScript Object Notation*) evolucionaron más tarde como un estándar. El formato es texto puro y fácilmente identificable y procesable por máquinas a través de redes y plataformas. Sin embargo, aún persiste el desafío de encontrar una forma estandarizada para la descripción de los servicios REST.

Además de su enfoque en la arquitectura orientada a servicios, una de las principales ventajas de las API RESTful es su capacidad para admitir diferentes formatos de datos. Como se mencionó, el formato JSON se utiliza ampliamente en las API RESTful debido a su facilidad de procesamiento y su compatibilidad con múltiples lenguajes de programación. Además, las API RESTful también pueden admitir otros formatos de datos como XML y CSV (Demircioğlu & Kalipsiz, 2022).

Otro aspecto importante en la arquitectura de las API RESTful es la seguridad. La autenticación y la autorización son fundamentales para proteger los recursos y los datos que se comunican a través de estas. Las API RESTful utilizan diferentes técnicas de seguridad, como el uso de fichas (*tokens*) de acceso, la autenticación de dos factores y la encriptación de datos para garantizar la seguridad en la comunicación entre sistemas (Baniş et al., 2021).

También es necesario tener en cuenta otros aspectos. Los encabezados y los parámetros también son importantes en los métodos de una solicitud HTTP de la API de RESTful, ya que contienen información de identificación importante con respecto a los metadatos, la autorización, el Identificador Uniforme de Recursos (URI, por las siglas del término en inglés, *Uniform Resource Identifier*), el almacenamiento en caché, las *cookies* y otros elementos de la solicitud. Hay encabezados de solicitud y de respuesta, pero cada uno tiene sus propios códigos de estado e información de conexión HTTP.

Para que una API se considere de RESTful, debe cumplir los siguientes criterios (RedHat, 2023b):

- Arquitectura cliente-servidor compuesta de clientes, servidores y recursos, con la gestión de solicitudes a través de HTTP.
- Comunicación entre el cliente y el servidor sin estado, lo cual implica que no se almacena la información del cliente entre las solicitudes de GET y que cada una de ellas es independiente y está desconectada del resto.

- Datos que pueden almacenarse en caché y optimizan las interacciones entre el cliente y el servidor.
- Una interfaz uniforme entre los elementos, para que la información se transfiera de forma estandarizada. Para ello deben cumplirse las siguientes condiciones:
  - Los recursos solicitados deben ser identificables e independientes de las representaciones enviadas al cliente.
  - El cliente debe poder manipular los recursos a través de la representación que recibe, ya que esta contiene suficiente información para permitirlo.
  - Los mensajes autodescriptivos que se envíen al cliente deben contener la información necesaria para describir cómo debe procesarla.
  - Debe contener hipertexto o hipermedios, lo cual significa que cuando el cliente acceda a algún recurso, debe poder utilizar hipervínculos para buscar las demás acciones que se encuentren disponibles en ese momento.
- Un sistema en capas que organiza en jerarquías invisibles para el cliente cada uno de los servidores (los encargados de la seguridad, del equilibrio de carga, etc.) que participan en la recuperación de la información solicitada.
- Código disponible según se solicite (opcional), es decir, la capacidad para enviar códigos ejecutables del servidor al cliente cuando se requiera, lo cual amplía las funciones del cliente.

Si bien la API de REST debe cumplir todos estos parámetros, resulta más fácil de usar que un protocolo definido previamente, como el Protocolo Simple de Acceso a Objetos (SOAP, por las siglas del término en inglés, *Simple Object Access Protocol*), el cual tiene requisitos específicos, como la mensajería XML y la seguridad y el cumplimiento integrados de las operaciones, que lo hacen más lento y pesado. Por el contrario, REST es un conjunto de pautas que pueden implementarse según sea necesario. Por esta razón, las API de REST son más rápidas y ligeras, cuentan con mayor capacidad de ajuste y, por ende, resultan ideales para el IoT y el desarrollo de aplicaciones para dispositivos móviles. Para resumir, las API RESTful son herramientas fundamentales en el desarrollo de software y la integración de sistemas, permiten la

creación de aplicaciones de manera rápida y eficiente. Su enfoque en la arquitectura orientada a servicios, la capacidad para admitir diferentes formatos de datos y la seguridad en la comunicación entre sistemas las hacen una opción popular para la creación de aplicaciones y servicios en línea.

### 2.3 Evolución a una Arquitectura Basada en Servicios (SBA)

Los servicios previstos para 5G tienen demandas diferentes, lo que hace extremadamente complejo crear una sola red capaz de soportarlos a todos, y en caso de ser posible el costo sería muy elevado (Sama et al., 2016). Los distintos servicios que se demandan por una amplia gama de dispositivos (teléfonos inteligentes, dispositivos portátiles y sensores, entre otros) imponen requisitos sin precedentes a la arquitectura de red móvil actual de 4G, que utiliza el concepto *“one size fit all”* (Sama et al., 2016).

Es decir, todos los servicios son tratados de igual forma por la Red de Acceso al Radio (RAN, por las siglas del término en inglés, *Radio Access Network*) (ver Figura 6), en este sentido, se puede inferir que la arquitectura de la red móvil 4G hace un aprovechamiento ineficiente de sus recursos cuando los requisitos para diferentes servicios se tratan de forma similar; ya que la red no sabe diferenciar las necesidades específicas de cada uno de ellos (X. Li et al., 2017).

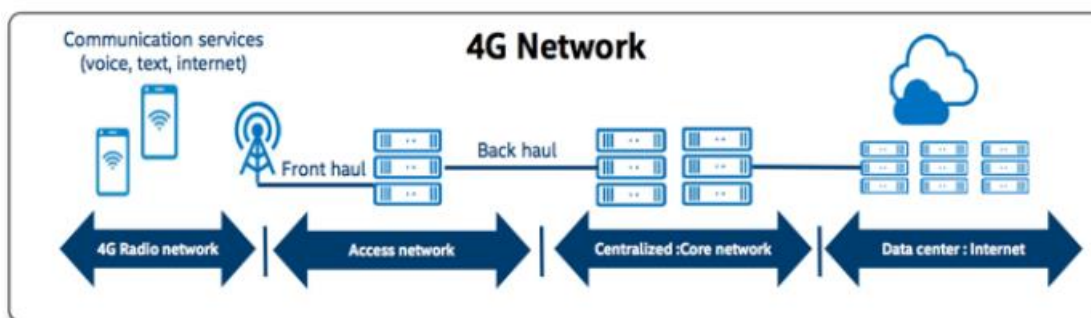


Figura 6 Arquitectura de la red móvil de 4G (5gAmericas, 2016).

La tecnología 4G utiliza un ancho de banda y un espectro limitado, lo que significa que su capacidad para transmitir grandes cantidades de datos de manera eficiente es limitada. Además, el aumento constante de la demanda de servicios de comunicaciones móviles implica una mayor carga en la red y una disminución en la calidad de la señal (5gAmericas, 2016).



En 5G la Calidad de Experiencia (QoE, por las siglas del término en inglés, *Quality of Experience*) de los usuarios, juega un papel fundamental. Por ese motivo, 5G se auxilia de tecnologías como SDN, NFV, ML y NS con el objetivo de perfeccionar la QoS, y también que la QoE sea lo mejor posible. Estas tecnologías permiten, además de mejorar la escalabilidad, flexibilidad y dinamismo de la red, la optimización de servicios en particular, y reducir así riesgos y costos (Schmelz & Nok, 2017).

### 2.3.1 Redes Definidas por Software (SDN)

Las Redes Definidas por Software (SDN) constituyen una arquitectura de red emergente donde el plano de control (software) está desacoplado del plano de datos (hardware), y centraliza lógicamente la inteligencia de la red en el controlador. El controlador centralizado permite al operador de red dirigir el tráfico de datos durante el tiempo de ejecución, y otorga la automatización de tareas como la de configuración y la gestión de políticas (Sama et al., 2016). Como resultado, las empresas y las compañías ganan la automatización y el control de la red, lo que les concede construir redes altamente escalables y flexibles que se adaptan fácilmente a las cambiantes necesidades del negocio, a los usuarios finales y al mercado. Las SDN ofrecen una operación centralizada que incluye la gestión de la topología de red, el seguimiento del estado de sus recursos, y otros controles avanzados (Ballester Macías, 2014).

La automatización de la gestión y provisión de la red, a través de SDN, es la siguiente fase en la virtualización de la infraestructura de tecnologías de la información y las telecomunicaciones (Millán Tejedor, 2014). De este modo, SDN crea una red inteligente mucho más abierta, flexible, escalable y reprogramable. La red se transforma en una herramienta de negocio más eficiente y permite a través de API, que las aplicaciones soliciten y modifiquen dinámicamente los servicios proporcionados por la infraestructura de red, sin necesidad de intervención humana, pudiendo la red informar de su nuevo estado a las aplicaciones. Es así como SDN permite maximizar la fiabilidad, rendimiento, escalabilidad y calidad, que requiere cada tipo de servicio o usuario actual o futuro.

#### 2.3.1.1 Arquitectura de SDN

La figura 7 ilustra la arquitectura de SDN especificada por la Fundación de Redes Abiertas (ONF, por las siglas del término en inglés, *Open Networking Foundation*). Se muestra a alto nivel que está compuesta

por tres capas o planos: aplicación, control e infraestructura; y por las interfaces: *Southbound* API y *Northbound* API.

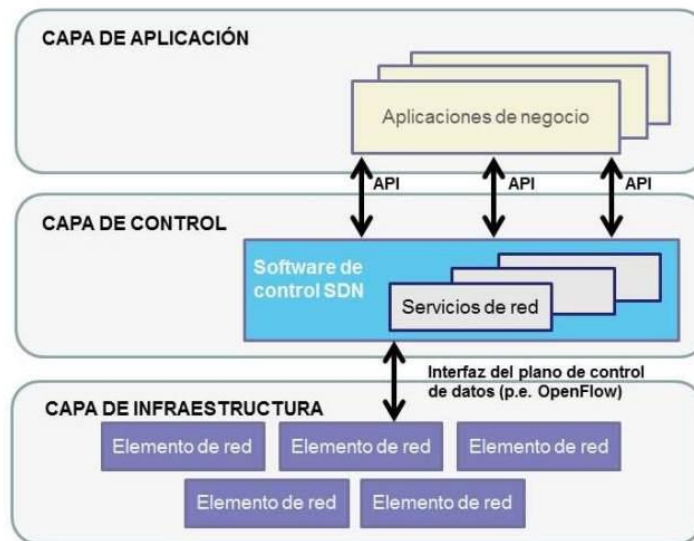


Figura 7 Arquitectura de SDN (Millán Tejedor, 2014).

A continuación, se describen los elementos que conforman la arquitectura de SDN: (Acosta Rams et al., 2017; Millán Tejedor, 2014)

- *Southbound* API: Estas interfaces conectan al controlador SDN con los dispositivos de reenvío, siendo fundamental para lograr la interacción entre los planos inferiores de la arquitectura. Esta API permite al controlador SDN programar dinámicamente a los dispositivos de red.
- *Northbound* API: Estas interfaces permiten al controlador interactuar con las aplicaciones y servicios ubicados en el Plano de Aplicaciones, dándole la posibilidad a éstas de obtener información sobre la infraestructura y controlar las políticas y el comportamiento de la red de forma granular. Esta interfaz comúnmente abstrae el bajo nivel de instrucciones que implementan las interfaces *Southbound* para programar los dispositivos de reenvío, constituyendo una potente herramienta para el desarrollo de aplicaciones.
- Plano de control: Aquí se ubica el controlador SDN, “el cerebro de la red”, quien se encarga de realizar todas las funciones complejas de la misma. Este plano posee interfaces para la comunicación e interacción con el plano de infraestructura y con el plano de aplicaciones,

denominadas *Southbound API* y *Northbound API*, respectivamente. Varios dispositivos del plano de infraestructura pueden ser conectados a un único controlador, quien, con una visión global, toma las decisiones de forma centralizada, en lugar de varios dispositivos tradicionales que sólo poseen un conocimiento limitado de la red. En la actualidad existen diseños que implican dos controladores SDN, un *máster* y un esclavo, para tener redundancia en la red. La función de control está centralizada y permite a los desarrolladores de aplicaciones utilizar capacidades de la red, pero abstrayéndolos de su topología o funciones. El controlador SDN es una entidad que tiene control exclusivo sobre un conjunto abstracto de recursos de plano de control, es decir, es la entidad que controla y configura los nodos de red para dirigir correctamente los flujos de tráfico.

- Plano de infraestructura: A diferencia de los dispositivos tradicionales, la infraestructura subyacente de las SDN está formada por elementos simples de reenvío sin mecanismos que les permitan tomar decisiones de forma autónoma. Estos dispositivos se caracterizan por realizar un conjunto de operaciones elementales, tales como en el reenvío de paquetes, la monitorización de información y recolección de estadísticas. Reciben instrucciones del controlador SDN que determinan su comportamiento, el cual está directamente vinculado a lo establecido por las aplicaciones. La infraestructura está constituida por los nodos de red que realizan la conmutación y encaminamiento de paquetes. Proporciona un acceso abierto programable a través de la *Southbound API*.
- Plano de aplicación: incluye las funciones de red, herramientas de orquestación y aplicaciones de negocio que determinan el comportamiento y características de la red. Este plano recibe la información proveniente de los dispositivos de reenvío a través del controlador SDN, obteniendo una visión abstracta y global del estado de la red. Permite a los servicios y aplicaciones simplificar y automatizar las tareas de configuración, provisionar y gestionar servicios nuevos en la red.

Los controladores actuales ofrecen una gran variedad de *Northbound API*: API RESTful, interfaces de programación multinivel, entre otras. Existen controladores como: *OpenDayLight* (ODL), RYU y ONOS, que proponen y definen su propia *Northbound API*, cada una con definiciones específicas en correspondencia con los intereses de los fabricantes y comunidades de desarrollo. La inexistencia de una *Northbound API* estándar obliga a las aplicaciones a preocuparse por los detalles de bajo nivel de la interfaz *Southbound*. De ahí que hayan surgido lenguajes para la programación de SDN como una alternativa para facilitar el desarrollo de aplicaciones, y lograr así todo el potencial que las SDN prometen. Aunque se destacan varias propuestas para describir esta comunicación, *OpenFlow*, soportada por una gran gama de controladores

y otros dispositivos de red se ha convertido en el estándar por excelencia para las SDN (Acosta Rams et al., 2017).

### 2.3.1.2 OpenFlow

Una de las formas más populares de implementar SDN es el protocolo *OpenFlow*, que es soportado por la ONF. *OpenFlow* es un protocolo de bajo nivel para implementar un control de los nodos de red, para ser compatibles con el controlador SDN requiere de nuevos desarrollos en estos elementos. *OpenFlow* es un estándar abierto que busca la interoperabilidad entre distintos fabricantes.

Para identificar el tráfico de la red *Openflow* utiliza el concepto de flujos basado en la coincidencia de reglas predefinidas que puedan ser programadas de forma estática o dinámica por el controlador SDN, así como, definir cómo debe fluir el tráfico a través de los dispositivos de red (Acosta Rams et al., 2017). De esta forma, el controlador central establece una conexión segura con los *switches* y, utilizando el protocolo *OpenFlow*, les transfiere las configuraciones basadas en tablas de flujo; sin ser necesario utilizar protocolos de enrutamiento (Ballester Macías, 2014). Al ser *OpenFlow* una interfaz abierta, es posible programar desde el controlador una gran variedad de dispositivos de reenvío de múltiples proveedores, algo muy difícil en las arquitecturas de redes tradicionales, debido a diversidad de fabricantes, interfaces cerradas, y a la naturaleza distribuida del plano de control.

Típicamente, cuando un paquete de datos llega a un *switch*, se comprueba el destino del paquete y se reenvía de acuerdo con las reglas predefinidas sobre las que los operadores de red carecen de control. Todos los paquetes dirigidos al mismo nodo se encaminan por una ruta única y se tratan del mismo modo. En una red que ejecuta *OpenFlow*, los administradores pueden añadir, quitar y gestionar reglas (Ballester Macías, 2014). Esto es muy importante cuando se habla de gestión de red y priorización de tráfico, ya que se le puede dar prioridad al tráfico de determinados servicios 5G sobre otros. En resumen, el protocolo *OpenFlow* indica al tráfico cómo fluir; para lo que se toman en consideración varios parámetros como el patrón de uso de la red, las aplicaciones de negocio y los requerimientos específicos de cada una de ellas. Las reglas son definidas para cada flujo, con lo que se obtiene un control sumamente granular que permite a la red tener una gran capacidad de respuesta a los cambios que requieran las aplicaciones o los usuarios (Ballester Macías, 2014). Actualmente se utiliza la versión del protocolo OpenFlow 1.4, publicada en 2013 por grandes empresas en el sector de las telecomunicaciones como son: Google, Microsoft, IBM, Juniper, Oracle, CISCO, etc.

### 2.3.2 Virtualización de Funciones de Red (NFV)

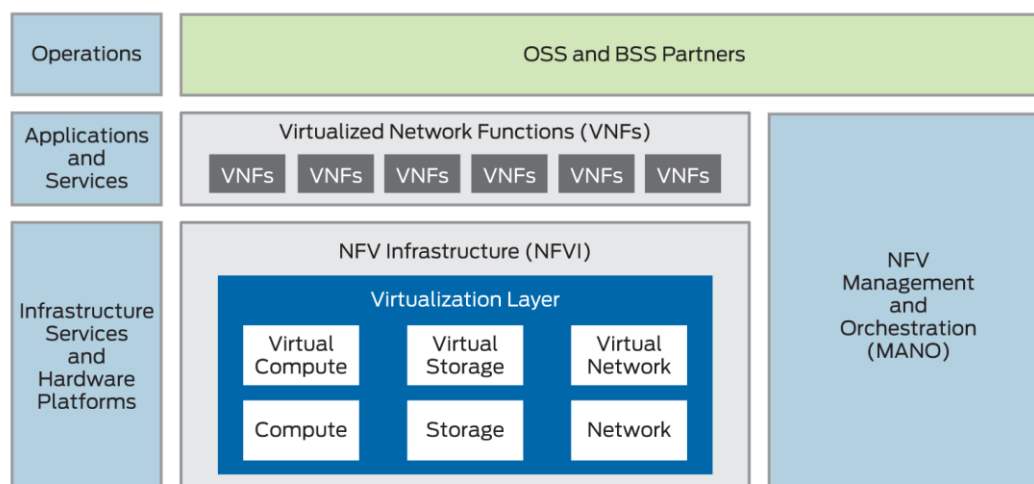
La Virtualización de las Funciones de Red (NFV) se utiliza para virtualizar los servicios de red, como los enrutadores, cortafuegos (*firewalls*) y los equilibradores de carga, que tradicionalmente se ejecutaban en el hardware propietario. Estos servicios se empaquetan como Máquinas Virtuales (VM, por las siglas del término en inglés, *Virtual Machines*) en el hardware básico, lo cual permite que los proveedores de servicios ejecuten sus redes en servidores estándar, en lugar de hacerlo en sistemas propietarios. Es uno de los principales elementos de la nube para las telecomunicaciones y está redefiniendo este sector (RedHat, 2023). No solo promete una reducción de costos, sino que también permite la rápida implementación de nuevos servicios y escalar flexiblemente las funciones de red según sea necesario.

Al aprovechar la tecnología de virtualización, con la NFV no se necesita tener un hardware exclusivo para cada función de la red, ya que se desacopla la función de red del hardware; lo que permite implementar dichas funciones de red en servidores, conmutadores y almacenamiento estándar de alta capacidad (Sama et al., 2016). Además, mejora la capacidad de adaptación y la agilidad, pues permite que los proveedores, según se solicite, ofrezcan nuevos servicios y aplicaciones de red nuevos, sin necesidad de contar con recursos de hardware adicionales (Sama et al., 2016). El deseo de automatizar la orquestación y gestión de redes, el almacenamiento y los recursos computacionales es un impulsor clave para el desarrollo de NFV y SDN. Supóngase un escenario donde hay un servidor físico con aproximadamente 15 VM o cientos de contenedores. En este concepto, si se necesitan operaciones manuales se hace realmente difícil poder modificar su escala. Con la automatización, puede poner a punto o eliminar rápidamente Funciones de Red Virtualizadas (VNF, por las siglas del término en inglés, *Virtual Network Functions*) como VM, contenedores, enrutadores, *firewalls* y otros, para modificar la escala de sus funciones de red forma elástica y satisfacer la demanda dinámica (Juniper Network, 2023).

#### 2.3.2.1 Arquitectura NFV

La arquitectura de una solución NFV típica se conforma de varios componentes, que se pueden ver en la figura 8 (Juniper Network, 2023), se incluyen las VNF, un Administrador de Infraestructura Virtualizada (NFVI) y un orquestador y administrador de NFV (MANO). En conjunto los Sistemas de Soporte de Operación y Negocio (OSS y BSS, por las siglas de los términos en inglés, *Operations Support Systems* y *Business Support Systems*; respectivamente) permiten a los proveedores de servicios de telecomunicaciones gestionar eficazmente tanto la operación de la red como los procesos comerciales, lo

que les permite ofrecer servicios de alta calidad y una experiencia del cliente de forma satisfactoria. La arquitectura modular de NFV es lo que permite a los operadores de telecomunicaciones automatizar todos los niveles. Los componentes de la arquitectura se explican a continuación: (Juniper Network, 2023)



**Figura 8 Arquitectura de la NFV** (Juniper Network, 2023)

- Bloque de construcción de infraestructura NFV (NFVI): proporciona la capa de virtualización (hipervisores o sistemas de gestión de contenedores como *Kubernetes*) y los componentes físicos de almacenamiento, redes y computación que alojan las VNF. La NFVI se administra a través del gestor de infraestructura, que controla la asignación de recursos para las VNF.
- VNF: son aplicaciones basadas en software que brindan uno o más servicios de red. Usan la infraestructura virtualizada que le proporciona la NFVI para conectarse a la red y brindar servicios de red escalables y programables. Los gestores de VNF admiten el ciclo de vida de instancias VNF y la gestión de un software VNF.
- Gestión y orquestación (MANO): proporciona la gestión y orquestación generales de las VNF en la arquitectura NFV. MANO crea una instancia de los servicios de red a través de la automatización, el aprovisionamiento y la coordinación de flujos de trabajo a los gestores y VNF que crean instancias de la VNF y superponen cadenas de servicio de red. MANO conecta la arquitectura de NFV con el OSS/BSS existente.

- OSS: se utiliza para describir los sistemas que se utilizan para la gestión de las operaciones de la red de telecomunicaciones. Los OSS se utilizan para monitorear, administrar y optimizar el rendimiento de la red, así como para proporcionar servicios de soporte para proveer servicios a los clientes. Entre los sistemas que se engloban dentro de los OSS se encuentran los sistemas de gestión de red, sistemas de inventario, sistemas de planificación y diseño de la red, y sistemas de gestión de servicios.
- BSS: se utiliza para describir los sistemas que se emplean para la gestión de los procesos comerciales de los proveedores de servicios de telecomunicaciones. Los BSS se aplican para gestionar los procesos de negocio relacionados con la gestión de clientes, como la facturación, el manejo de pedidos, la gestión de tarifas, la gestión de campañas y la gestión de relaciones con los clientes. Entre los sistemas que se engloban dentro de los BSS se encuentran los sistemas de facturación, sistemas de gestión de pedidos, sistemas de gestión de tarifas, y sistemas de gestión de relaciones con los clientes.

### 2.3.3 Network Slicing y SBA

*Network Slicing* es un concepto que se ha propuesto para abordar los requisitos de servicios diversificados (W. Wu et al., 2022). La idea básica de la segmentación de red es crear varias redes virtuales (es decir, segmentos) sobre una infraestructura de red física común para proporcionar control y abstracción flexibles, centralizados y programables para las redes, así como eliminar el estrecho acoplamiento entre las funciones de red y unidades de hardware particulares (W. Wu et al., 2022).

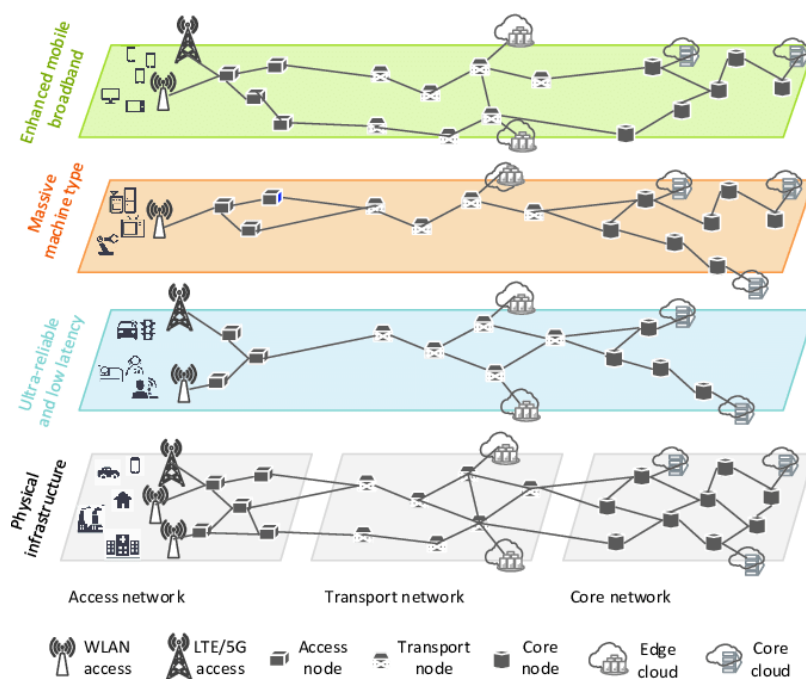
Específicamente, un segmento de red es una red autónoma con sus propios recursos virtuales, topología, flujo de tráfico y reglas de aprovisionamiento que brindan al inquilino del segmento la capacidad de operar su propia red física dedicada. Estas redes lógicas se crean y administran al observar las demandas de los usuarios finales y los administradores y luego se proporcionan a diferentes servicios para cumplir con los diferentes requisitos de comunicación de los usuarios. Por lo tanto, la segmentación de redes se considera uno de los habilitadores claves de los sistemas 5G, ya que permite que las redes al ser dinámicas, ágiles y escalables respondan rápidamente a los requisitos comerciales cambiantes (W. Wu et al., 2022).

La figura 9 muestra la arquitectura de NS en 5G definida para los tres casos de uso, previamente definidos por la ITU, y además se agrega un segmento de infraestructura física según (Guan et al., 2018). Se aprecia

como varios Equipos de Usuario (UE, por las siglas del término en inglés, *User Equipment*) solicitan servicios, las solicitudes se transportan por el segmento de red (*network slice*) que le corresponde, con la premisa de que los segmentos de red están diseñados en función de los requisitos específicos de cada caso de uso.

Sin embargo, en la actualidad se habla de un segmento (*slice*), para cada servicio en específico, más allá de un segmento por caso de uso (*use case*). Lo que se trata de un diseño muy *ad-hoc* al servicio solicitado por parte del usuario. Esta es otra premisa que refuerza la idea de tener un buen clasificador de servicios que identifique lo mejor posible el servicio solicitado por el usuario para asignarle un NS, acorde a los parámetros de QoS que necesite para cumplir con el SLA.

La implementación de NS en 5G es de Extremo a Extremo (E2E, por las siglas del término en inglés, *End-to-End*), desde el núcleo hasta la RAN y un segmento se compone de una colección de funciones de red personalizadas lógicas que respaldan los requisitos del servicio de comunicación de un caso de uso particular (Sama et al., 2016). Según (5gAmericas, 2016), el NS habilita los elementos y las funciones de red para ser configuradas de una manera fácil en partes lógicas sobre una infraestructura de red física, con el objetivo de obtener múltiples redes virtuales optimizadas y garantizar la QoS requerida. Precisamente los proveedores de servicios de redes 5G manejan la idea de implementar segmentos múltiples aislados para una variedad de servicios en una infraestructura común.



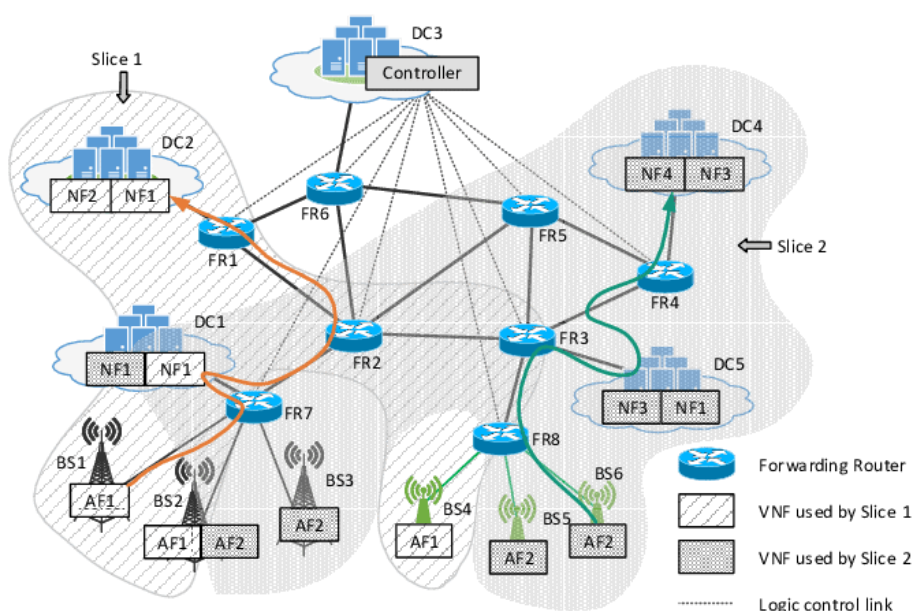
**Figura 9 Arquitectura de NS en 5G (Guan et al., 2018).**



En detalle, en el núcleo de red (*core network*), SDN y NFV virtualizan los elementos y funciones de la red en cada segmento de ésta, para cumplir con sus propios requisitos de infraestructura. La red física del operador se divide en múltiples redes virtuales de E2E, donde cada segmento está lógicamente aislado y tiene un tratamiento dedicado en términos de rendimiento o funcionalidad (X. Li et al., 2017).

En la figura 10 se puede ver un ejemplo de una topología de red que utiliza un controlador SDN y algunas VNF (Wang et al., 2017). Se aprecia como el controlador SDN conoce toda la topología y ya sabe la ubicación de los routers y algunas VNF, por lo que puede separar los NS. Hay que destacar que en este trabajo de tesis se emplea este método de razonamiento.

En otras palabras, NS puede ser visto como parte del paradigma de “*Network as a Service*”, introduciendo además una nueva SBA para la red móvil 5G. La arquitectura 5G se define como basada en servicios y la interacción entre las funciones de red dentro del plano de control y de usuario permiten que otras funciones de red autorizadas tengan acceso a sus servicios (3GPP, 2019).



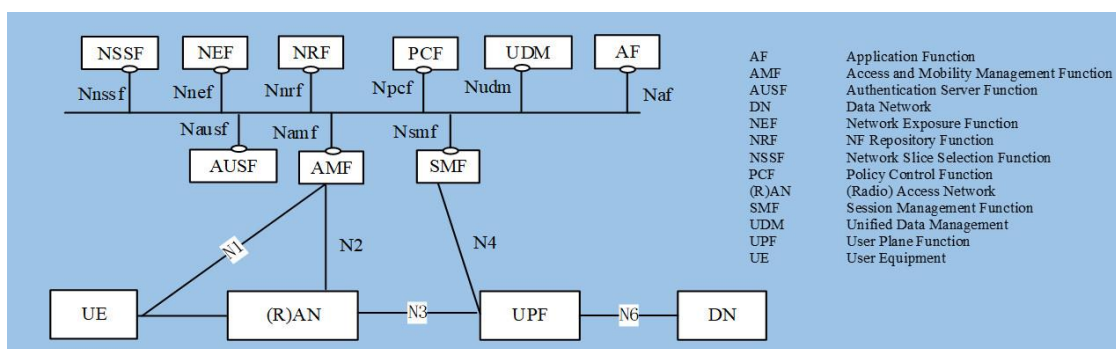
**Figura 10 Ilustración de NS en una topología de red (Wang et al., 2017).**

Es importante destacar que la implementación de SLA en el contexto de 5G implica una complejidad adicional debido a la naturaleza heterogénea y dinámica de la red. Para abordar esta problemática, se ha propuesto la utilización de *network slicing*, que consiste en la creación de segmentos de red virtualizados, capaces de adaptarse a las necesidades específicas de los usuarios y aplicaciones, y que pueden ser

gestionados de manera independiente. De esta forma, se pueden establecer SLA personalizados para cada *network slice* (con un servicio específico), lo que permite una gestión más eficiente de los recursos y una mejor adaptación a los requisitos de los clientes.

La asignación de parámetros de QoS en los *network slices* es un proceso clave para garantizar el cumplimiento de los compromisos de SLA. En este sentido, la utilización de técnicas de ML es de gran ayuda para automatizar y optimizar este proceso. A través del análisis de datos de tráfico y comportamiento de los usuarios, se establecen modelos predictivos que permitan la asignación dinámica de recursos y la adaptación de los parámetros de QoS en tiempo real. Esto no solo mejora la eficiencia de la gestión de los *network slices*, sino que también permite una mejor experiencia de usuario y un aumento en la satisfacción del cliente.

La figura 11 refleja la SBA de la red 5G, definida por el Proyecto de Asociación de 3ra Generación (3GPP, por las siglas del término en inglés, *Third Generation Partnership Project*) en (3GPP, 2019). La SBA de 5G se compone por un conjunto de Funciones de Red (NF, por las siglas del término en inglés, *Network Function*) interconectadas, y encargadas de administrar el plano de usuario y de control. SBA se basa en funcionalidades de entidades de red que se convierten en servicios expuestos y ofrecidos a otras entidades de red. Estas funciones de red exponen su funcionalidad a través de Interfaces Basadas en Servicios (SBI, por las siglas del término en inglés, *Service Based Interfaces*) a través de un bus de mensajes SBI que implementa API RESTful sobre HTTP/2 (3GPP, 2019).



**Figura 11 SBA de 5G definida por 3GPP (3GPP, 2019).**

Esencialmente, todas las NF pueden comunicarse entre sí mediante una solicitud/respuesta o interacciones de suscripción/notificación entre los consumidores y productores de servicios de NF. Desde el punto de vista de la seguridad, dicha comunicación requiere la protección de la confidencialidad y la

integridad de los mensajes intercambiados, así como un fuerte mecanismo de autenticación y autorización (3GPP, 2019).

A continuación, se mencionan la funcionalidad principal de las NF: (3GPP, 2019)

- AMF: Gestiona el registro de UE, la gestión de movilidad de UE. Por ejemplo, rastrea la ubicación del UE y proporciona eventos de movilidad a otras NF.
- AUSF: Servidor que atiende las funciones de Autenticación.
- AF: Funciones generales o específicas para las aplicaciones.
- PCF: Gestión de políticas para suscriptores y para segmentos. Esto incluye la selección de rutas de tráfico, reenvío de tráfico, políticas de itinerancia, tarifas y políticas de QoS.
- NEF: Proporciona un puente hacia las verticales para acceso a los datos y funcionalidades del núcleo 5G. Expone datos a terceros, asigna identificadores y convierte protocolos
- NRF: Admite el descubrimiento de servicios, la autorización del consumidor de NF y mantiene los perfiles de NF, los servicios y la lista de instancias de NF. Actúa como el servidor de autorización OAuth 2.0.
- NSSF: Ayuda en la selección de instancias de segmento de red adecuadas para los usuarios y en la asignación de las funciones de gestión de acceso (AMF) necesarias.
- SMF: Realiza el establecimiento, modificación y liberación de sesiones. La asignación de la dirección IP y la gestión de la UE hace que la recopilación de datos de carga, la dirección del tráfico y el enrutamiento en la UPF.
- UDM: Admite el acceso al almacenamiento de datos. Por ejemplo, para la gestión de suscripción, autorización de acceso y servicio, almacenamiento y gestión de identificación de usuario, autenticación de usuario.
- UPF: Proporciona el punto de sesión de la PDU externa de interconexión a las redes de datos. Realiza inspección de paquetes y aplicación de políticas y reglas de QoS para el plano del usuario.

La ventaja principal de esta arquitectura basada en servicios es la interconexión entre todas sus NF mediante API RESTful. Esto permite que las NF puedan intercambiar datos mediante solicitudes HTTP (GET, POST, PUT, DELETE), lo que admite, además, que APIs de terceros puedan interactuar también en el bus de SBI (siempre que cumpla con los protocolos de seguridad).

## 2.4 Aprendizaje de Máquina (ML)

ML es un subcampo de las ciencias de la computación y una rama de la AI cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender por su cuenta (Ruiz, 2018). De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos. Por lo tanto, es un proceso de inducción del conocimiento y puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos (C. Jiang et al., 2016).

Explícitamente, una máquina aprende la ejecución de una tarea particular  $T$ , con el objetivo de mantener una métrica de rendimiento específica  $P$ , basada en una experiencia particular  $E$ , donde el sistema tiene como objetivo mejorar de manera confiable su rendimiento  $P$  mientras ejecuta la tarea  $T$ , explotando su experiencia  $E$ . En dependencia de cómo se especifique  $T$ ,  $P$  y  $E$ , el aprendizaje se fundamenta en: minería de datos, descubrimiento autónomo o actualización de bases de datos, etc. (C. Jiang et al., 2016).

ML utiliza el poder de *Big Data* y recursos informáticos para procesar patrones en datos históricos, y luego utilizarlos como funciones de predicción al analizar datos futuros (Wei-Lun, 2011). Se basa típicamente en usar algoritmos matemáticos o métodos estadísticos para examinar un conjunto de datos existente y usar los resultados para realizar predicciones (Morocho-Cayamcela et al., 2019).

Un concepto clave para el entendimiento de ML y de este trabajo, es el de conjunto de datos (*dataset*). Un conjunto de datos contiene ejemplos de entrenamiento (filas) y características (columnas), y de forma general, existen dos tipos de conjuntos de datos: etiquetados y no etiquetados. El conjunto de datos etiquetado se representa por  $X = \{x^n \in R^d\}_{n=1}^N$ ,  $Y = \{y^n \in R\}_{n=1}^N$  y el no etiquetado se representa solo por  $X = \{x^n \in R^d\}_{n=1}^N$ . Donde  $X$  denota el conjunto de características que contiene  $N$  muestras. Cada muestra es un vector  $d$ -dimensional y se denomina vector de características, mientras que cada dimensión de un vector se denomina atributo, variable o elemento.  $Y$  representa el conjunto de etiquetas, y registra qué etiqueta se corresponde con su vector de características (Wei-Lun, 2011).

El *dataset* se debe dividir en: datos de entrenamiento y prueba. El conjunto de entrenamiento se usa para entrenar un algoritmo de ML y el conjunto de prueba, para asegurarse de que las predicciones del modelo predictivo generado por el algoritmo sean correctas. Al aplicar ML, el objetivo es estimar qué tan bien se generaliza un algoritmo de aprendizaje más allá de las muestras del conjunto de entrenamiento (Wei-Lun, 2011).

Incorporar ML para ayudar a resolver un problema conlleva un conjunto de desafíos. En la actualidad resulta muy difícil conseguir un *dataset* que contenga parámetros de KPIs, KQIs y rendimientos de los SLA. Por este motivo, este trabajo emplea un *dataset* desarrollado por los autores en (Preciado-Velasco et al., 2021), que contiene un conjunto de 9 servicios básicos 5G/B5G y los principales KPI/KQI, con el cumplimiento a través de los estándares de telecomunicaciones. En el capítulo 3 se describe este *dataset*.

La figura 12 muestra un diagrama en bloques del esquema general de un problema de ML, con un ejemplo de *networking* específicamente (Borcoci, 2019). Como primer paso es la formulación del problema, es decir, qué tipo de problema vas a resolver: regresión, clasificación, agrupamiento, etc. El paso dos, es la colección de los datos y luego seguido por un análisis o preprocesamiento del conjunto. Posteriormente se procede a la construcción del modelo (paso 4), con un entrenamiento fuera de línea (*offline*). Como paso cinco, se tiene la validación del modelo, donde se aplican técnicas de validación cruzada (*cross-validation*) por ejemplo; si el modelo no cumple con las expectativas se deben repetir los pasos (ver figura 12). Si el modelo cumple con las métricas propuestas, se prosigue a la implementación del modelo, que en la actualidad implica la implementación del modelo en una API RESTful que atienda solicitudes de distintos usuarios. Este esquema representa una guía excelente para la solución e implementación de un problema ML.

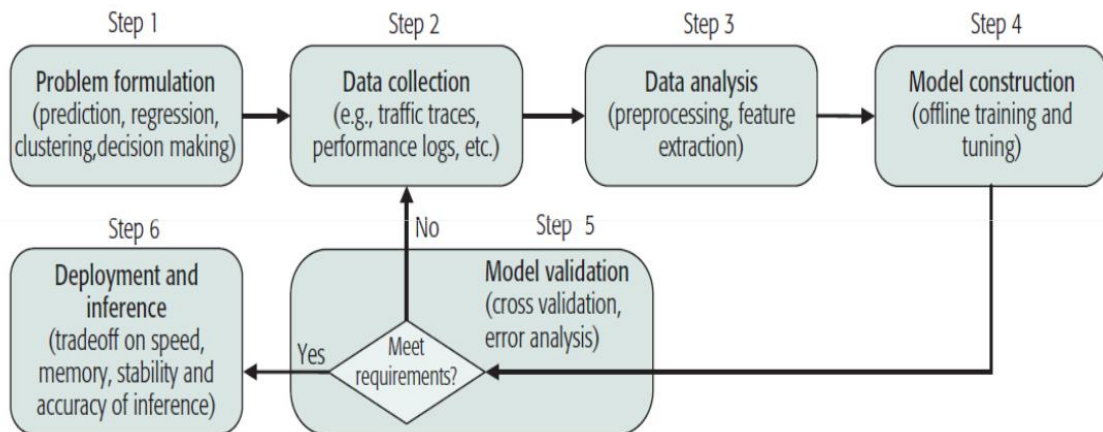


Figura 12 Esquema general de ML (Borcoci, 2019).

El esquema general de todo problema de ML presenta dos fases de trabajo: la fase de aprendizaje y la fase de inducción, ambas se pueden observar en la figura 13. La primera etapa, la de aprendizaje, es donde a través del descubrimiento de patrones en los datos de entrada (entrenamiento), los cuales se usan para describir los datos y posteriormente resumirlos en un modelo mediante un algoritmo de ML. En la fase dos, se induce la predicción al usar el modelo (de la fase de aprendizaje) (Borcoci, 2019). Se prueban otros datos nunca vistos (los datos de entrenamiento deben ser diferentes a los de prueba); los nuevos datos se transforman en un vector de características, que se introducen en el modelo y da como resultado una predicción. El objetivo del modelo de entrenamiento es minimizar el error entre las predicciones y los valores reales.

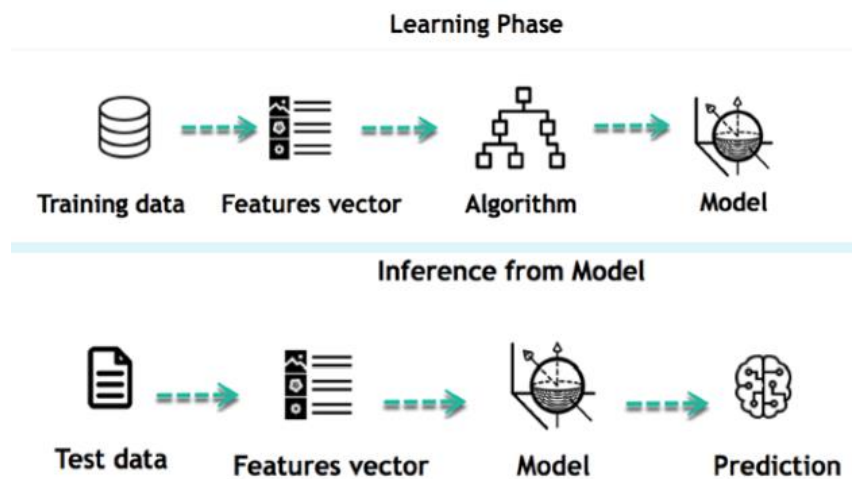


Figura 13 Fases de trabajo de un modelo de ML (Borcoci, 2019).

Como se mencionó, al crear un modelo de ML, el objetivo es seleccionar el que realice las mejores predicciones. Sin embargo, si se selecciona el modelo que presenta el “mejor” rendimiento predictivo en los datos de entrenamiento, es posible que este pueda tener un sobreajuste (*overfitting*) o subajuste (*underfitting*), e indica que el algoritmo de ML no se entrenó bien. Este sobreajuste puede ocurrir cuando un modelo tiene patrones memorizados que aparecen en las fuentes de datos de entrenamiento y evaluación, pero falla al generalizar los patrones de los datos. Para evitar que ocurra, se pueden hacer dos cosas: aumentar el volumen de la base de datos o reservar datos adicionales al dividir el conjunto de datos en tres partes (entrenamiento, validación y prueba) y con esto validar el rendimiento del algoritmo de ML (Liyanapathirana, 2018).

Asimismo, para evaluar los resultados del análisis estadístico y validar el algoritmo en un problema de ML, se usa la técnica de validación cruzada (*cross-validation*). El objetivo de la validación cruzada consiste en

estimar el nivel de ajuste de un modelo a un cierto conjunto de datos de prueba independientes de las utilizadas para entrenar el algoritmo (Xia et al., 2022).

La validación cruzada permite garantizar que los datos de la partición entre entrenamiento y prueba sean independientes (Xia et al., 2022). No se trata de utilizar un nuevo conjunto de datos de validación, sino consiste en emplear una técnica de validación cruzada. Cada técnica de validación cruzada tiene sus propias ventajas y desventajas, y la elección de la ella dependerá del problema específico que se esté resolviendo. A continuación, se listan algunas técnicas: (López Briega, 2016)

- *K*-iteraciones (*K-Folds*): donde el conjunto de datos se divide en *K* subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto (*K*-1) como datos de entrenamiento. El proceso de validación cruzada se repite durante *K* iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Cada iteración produce un modelo de predicción, y por lo tanto una estimación del rendimiento de la generalización, por ejemplo, una estimación de la precisión. Una vez finalizada la validación cruzada, todos los ejemplos se han utilizado sólo una vez para evaluar, pero (*K*-1) veces para entrenar.
- Retención: divide los datos en dos grupos, uno para entrenamiento y otro para validación. El modelo se entrena en el conjunto de entrenamiento y se evalúa en el conjunto de validación.
- Dejadedez: divide los datos en varios grupos y entrena el modelo en cada grupo, dejando uno de ellos fuera. El modelo se evalúa en el grupo que se dejó fuera. Esto se repite hasta que cada grupo haya sido utilizado una vez como conjunto de validación. Ésta se puede ver que tiene un coste computacional alto.
- Estratificada: es similar a la validación cruzada *K-Folds*, pero se asegura de que la distribución de las clases en los datos sea la misma en cada uno de los pliegues.
- Monte Carlo: selecciona aleatoriamente un subconjunto de los datos para entrenamiento y otro subconjunto para validación. Este proceso se repite muchas veces y se promedian los resultados.
- Por bloques: divide los datos en bloques y realiza varias iteraciones en las que se utiliza un bloque diferente como conjunto de validación y los demás bloques se utilizan para entrenamiento.

### 2.4.1 Aprendizaje Supervisado (SL)

El Aprendizaje Supervisado (SL, por las siglas del término en inglés, *Supervised Learning*) es una técnica de ML que toma datos de entrenamiento, organizados en un vector de características de entrada ( $x$ ) y un valor de salida deseado ( $y$ ) para desarrollar un modelo predictivo, al inferir una función  $f(x)$ , devolviendo la salida predicha  $y$  (Moysen & Giupponi, 2017). Las variables ( $x$ ) y ( $y$ ) se les conoce como variable independiente y dependiente respectivamente. Esta técnica utiliza conjuntos de datos de tipo etiquetados y es extremadamente útil cuando la función a abordar requiere estimación o clasificación de variables. La figura 14 muestra un modelo genérico de SL.

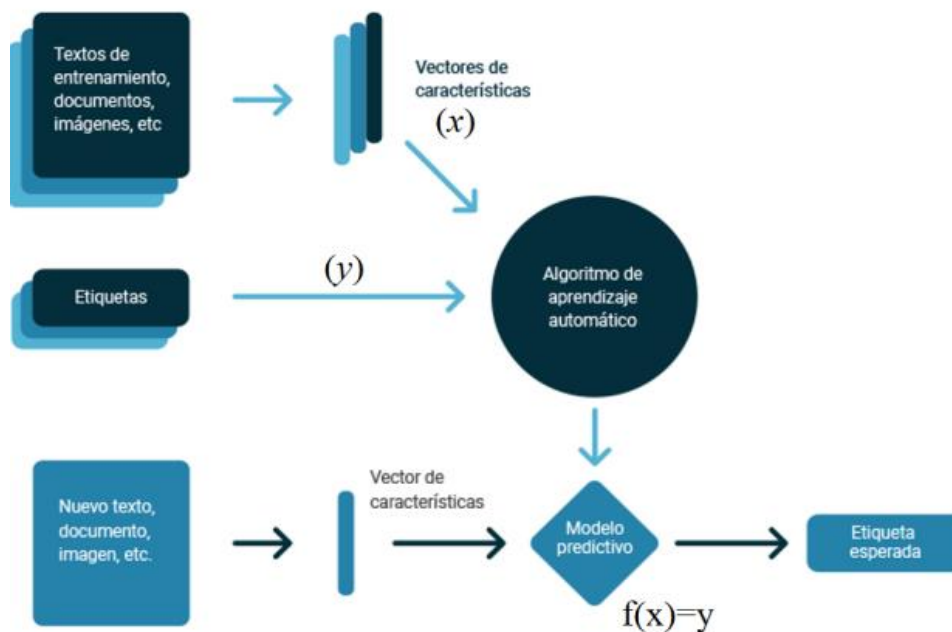


Figura 14 Esquema general de un modelo de SL (Miguel, 2018).

La técnica de SL tiene dos aplicaciones principales: regresión (para estimación) y clasificación. Por un lado, la clasificación se aplica cuando el valor de salida  $y$  (en ocasiones llamadas etiquetas o categorías) que se intenta predecir es discreto (Asiri, 2020). Por otro lado, se aplica la regresión cuando  $y$  es un valor continuo real. Existen una gran cantidad de algoritmos de SL que se pueden utilizar para la clasificación y regresión. A continuación, se presentan algunos de estos algoritmos: (Moysen & Giupponi, 2017)

- **Árbol de Decisión (DT, por las siglas del término en inglés, *Decision Tree*):** es un modelo de diagrama de flujo en el que la capa nodo interno representa una prueba de un atributo. Cada nodo



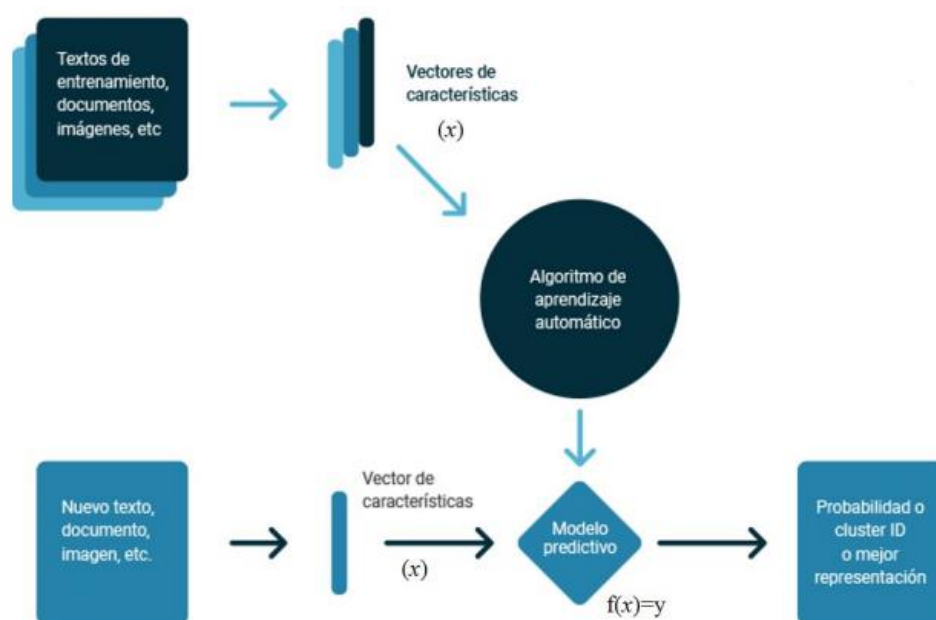
hoja representa una respuesta, y la rama representa el resultado de la prueba. No requieren ningún conocimiento previo de los datos, son robustos (es decir, no sufren del problema de la dimensionalidad, ya que se centran en los atributos más destacados) y funcionan bien para un conjunto de datos ruidosos.

- Bosque Aleatorio (*Random Forest*): es una combinación de DT predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de los árboles. Cada árbol es construido con la aplicación de la siguiente metodología: sea  $N$  el número de casos de prueba,  $M$  es el número de variables para la clasificación de etiquetas. Sea  $m$  el número de variables de entrada a ser usado para determinar la decisión en un nodo dado ( $m \ll M$ ). Para cada nodo del árbol, elegir aleatoriamente  $m$  variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las  $m$  variables. Para la predicción un nuevo caso es empujado hacia abajo por el árbol. Luego se le asigna la etiqueta del nodo terminal donde termina. Este proceso es iterado por todos los árboles en el ensamblado, y la etiqueta que obtenga la mayor cantidad de incidencias es reportada como la predicción (Bagnato, 2017).
- Máquinas de Soporte Vectorial (SVM, por las siglas del término en inglés, *Support Vector Machine*): están inspirados en la teoría del aprendizaje estadístico, que es una herramienta poderosa para estimar funciones multidimensionales. Dadas  $m$  muestras de entrenamiento  $((x_1, y_1), \dots, (x_m, y_m))$ , el objetivo es aprender los parámetros de una función que mejor se adapte a los datos (muestras de hiperplano). Tiene varios tipos de *kernel*, en dependencia de la distribución del conjunto de datos.
- K-Vecinos más Cercanos (KNN, por las siglas del término en inglés, *K-Nearest Neighbors*): es un método no lineal donde la entrada consiste en las  $k$  muestras de entrenamiento más cercanas en el espacio de entrada. El resultado previsto es el promedio de los valores de sus  $k$  vecinos más cercanos.
- Perceptrón Multi-Capa Clasificador (MLPC, por las siglas del término en inglés, *Multi-Layer Perceptron Classifier*): dado un *dataset* puede aprender un aproximador de función no lineal para clasificación o regresión. Se entrena usando un algoritmo de retropropagación (*backpropagation*). Está compuesto por múltiples neuronas artificiales, organizadas en capas. Cada neurona de una capa está conectada con todas las neuronas de la capa siguiente, formando una red densamente

conectada. Las capas ocultas, que se encuentran entre la capa de entrada y la capa de salida, permiten que el MLPC aprenda representaciones intermedias más complejas y no lineales de los datos de entrada (Scikit-learn developers, 2023).

## 2.4.2 Aprendizaje no Supervisado

El aprendizaje no supervisado (*unsupervised learning*) es una técnica de ML que recibe datos de entrada sin etiquetar, con el objetivo de encontrar un patrón en ellos. En comparación con el SL, el aprendizaje no supervisado requiere solo el conjunto de características, no el conjunto de etiquetas. Por tanto, el tipo de conjunto de datos es no etiquetado. En este caso, se deja que el algoritmo aprenda por sí mismo, sin conocer la respuesta correcta al problema que se quiere resolver. El objetivo es construir una representación de entradas que puedan usarse para predecir a partir de entradas futuras sin dar al algoritmo la respuesta correcta, ya que no se tienen las etiquetas esperadas (como en el caso de SL) (Moysen & Giupponi, 2017). En la figura 15 se puede observar un modelo genérico de aprendizaje no supervisado y comprender mejor lo recién explicado.



**Figura 15** Esquema general de un modelo de aprendizaje no supervisado (Miguel, 2018).

Este tipo de aprendizaje puede ser extremadamente útil cuando se requiere identificar comportamientos anómalos, reconocer patrones o reducir la dimensionalidad de los datos. Las familias más importantes de

aprendizaje no supervisado son: agrupación (*clustering*) y reducción de la dimensionalidad (C. Jiang et al., 2016).

Del método de reducción de dimensiones, el algoritmo más empleado es el Análisis de Componentes Principales (PCA, por las siglas del término en inglés, *Principal Component Analysis*). Éste y los algoritmos más usados para técnicas de agrupación se listan a continuación: (Shlens, 2014; Xia et al., 2022)

- PCA: permite extraer las características más relevantes de los datos, reducir su dimensionalidad y representarlos en un espacio de menor dimensión. Para esto se basa en la distribución de la varianza en cada característica de los datos. PCA dice que los datos que menos varianza contengan son los que menos información contienen. Es realmente útil para conjuntos de datos de gran dimensión y se requiere hacer un análisis para representación de datos y descorrelacionar las variables. Es muy común utilizar PCA en un *dataset* y luego aplicar algoritmos de SL o no supervisado. Sin embargo, hay que destacar que PCA puede mejorar el rendimiento de estos algoritmos, pero también puede ocultar información importante y afectar el desempeño de estos.
- K-Medias (*K-Means*): son extremadamente fáciles de implementar y muy eficientes desde el punto de vista computacional. Estas son las principales razones que explican por qué son tan populares. Tiene como objetivo encontrar y agrupar en clases los puntos de datos que tienen una alta similitud entre ellos. En los términos del algoritmo, esta similitud se entiende como lo opuesto de la distancia entre puntos de datos. Cuanto más cerca estén los puntos de datos, más similares y tendrán más probabilidades de pertenecer al mismo clúster.
- Agrupamiento Jerárquico: es una alternativa a los algoritmos de agrupación basados en prototipos. La principal ventaja de la agrupación jerárquica es que no necesita especificar el número de agrupaciones, el algoritmo lo encuentra por sí mismo. Son muy sensibles a los valores atípicos y, en su presencia, el rendimiento del modelo disminuye significativamente.
- Agrupación Espacial Basada en Densidad de Aplicaciones con Ruido (DBSCAN, por las siglas del término en inglés, *Density-Based Spatial Clustering of Applications with Noise*): es útil para identificar correctamente el ruido en los datos. No es necesario especificar el número de grupos. Existe una gran flexibilidad en las formas y tamaños que pueden adoptar los grupos, es muy útil para identificar y tratar con datos de ruido, y valores atípicos.

### 2.4.3 ML en gestión de redes de nueva generación (5G/B5G)

El avance en la tecnología de redes 5G ha creado una necesidad crítica de gestionar y optimizar de manera eficiente los recursos de la red para garantizar una experiencia de usuario satisfactoria. Con el aumento en la complejidad de la red y la demanda de alta QoS, la gestión de redes se ha vuelto cada vez más desafiante. En este contexto, el uso de técnicas de ML para la gestión de redes 5G ha surgido como una solución prometedora. ML ofrece una forma de automatizar y optimizar el proceso de gestión de redes mediante el análisis de datos y la identificación de patrones y tendencias. En este epígrafe se discuten las técnicas de ML utilizadas para la gestión de redes 5G, incluyendo la clasificación de tráfico, la predicción de demanda y la optimización de recursos de red.

ML implica beneficiarse de cómo una computadora puede “aprender” de la experiencia para mejorar su habilidad de: pensar, planear, decidir y actuar, es decir, habilitar a las computadoras para que aprendan por sí solas. ML ha evolucionado hasta el punto de que actualmente permite que las redes inalámbricas aprendan y extraigan conocimiento al interactuar con los datos. El interés preliminar y las discusiones sobre la viabilidad de la evolución de los estándares 5G con la ayuda de protocolos de ML captan la atención y la imaginación de ingenieros e investigadores de todo el mundo (Klaine et al., 2017).

ML es una de las tecnologías claves a utilizar en el monitoreo de la red 5G y facilitar su autoadaptación, ya que permite el ajuste del comportamiento en respuesta a la percepción del entorno y de los datos que ya han sido procesados. Tales funciones no pueden basarse por completo en reglas predefinidas hechas a mano, ya que es demasiado complejo predecir las condiciones de trabajo futuras causadas por cambios en el entorno, necesitándose mecanismos automatizados (Mullins & Barros, 2017).

Con la creciente complejidad de 5G el proceso de autoconfiguración es un desafío en la actualidad para los investigadores, debido a la cantidad de parámetros a considerar. Para realizar la autoconfiguración, se aplican varias técnicas de ML con el objetivo de automatizar las funciones de los equipos de red (Klaine et al., 2017). La autoconfiguración de la red emplea tecnologías como SDN, NFV y ML, que ayudan a la gestión y el análisis lo que permite resolver problemas de la red y mejorar la QoS/QoE (Barona López et al., 2017).

El desafío de 5G/B5G es asegurar el rendimiento de la red, mejorar la QoE de los usuarios y garantizar el cumplimiento de los SLA; para los casos de uso genéricos definidos por la ITU: eMBB, mMTC y urLLC..

En la actualidad la mayoría de los estudios de redes de nueva generación se centran en la detección de violaciones de los SLA. Esto está directamente relacionado con la afectación de la QoS del proveedor y QoE percibida por los usuarios. En la mayoría de los casos de estudio, los autores utilizan herramientas de ML para aprender a detectar anomalías respecto a una métrica, como el *throughput* para predecir violaciones de los SLA (Ramos García, 2021).

## 2.5 Conclusiones

En este capítulo se presentaron los Acuerdos a Nivel de Servicio (SLA), la evolución de la arquitectura de redes móviles de nueva generación, los conceptos fundamentales de API, así como la aplicación de técnicas de ML a la gestión de redes. Se puede concluir que:

- Los SLA son un tema importante para analizar la gestión de las redes, ya que tienen en cuenta los parámetros de QoS del proveedor para mantener una buena QoE para los usuarios.
- Las API RESTful son una herramienta fundamental en el desarrollo de software y la integración de sistemas, permitiendo la creación de aplicaciones de manera rápida y eficiente.
- La arquitectura de 4G no satisface las condiciones ni el enfoque basado en servicios y aplicaciones de las redes móviles de nueva generación. Por este motivo, se emigra a una SBA donde 5G se apoya en tecnologías emergentes como SDN, NFV y NS.
- ML es una herramienta clave para mejorar muchos aspectos de la gestión de redes, permite que las redes aprendan constantemente y mejoren la QoS, así como se enfoquen en garantizar el cumplimiento de los SLA.

## Capítulo 3. Propuesta de un sistema para asignar recursos de red

---

Las API, el clasificador de servicios 5G/B5G, la asignación de parámetros de QoS y el controlador SDN son componentes fundamentales en la gestión de las redes de nueva generación. En particular, la integración de una API de clasificación de servicios 5G/B5G con una API de un controlador SDN puede mejorar significativamente la gestión automatizada de las redes. Al utilizar un algoritmo de SL, la API de clasificación de servicios permite conocer y asignar parámetros de QoS a los servicios y formar redes virtuales de manera eficiente. Mientras tanto, la API del controlador SDN gestiona y monitorea la red. Con la integración de ambas API, se busca garantizar el cumplimiento de los SLA y así proporcionar una mejor experiencia de usuario.

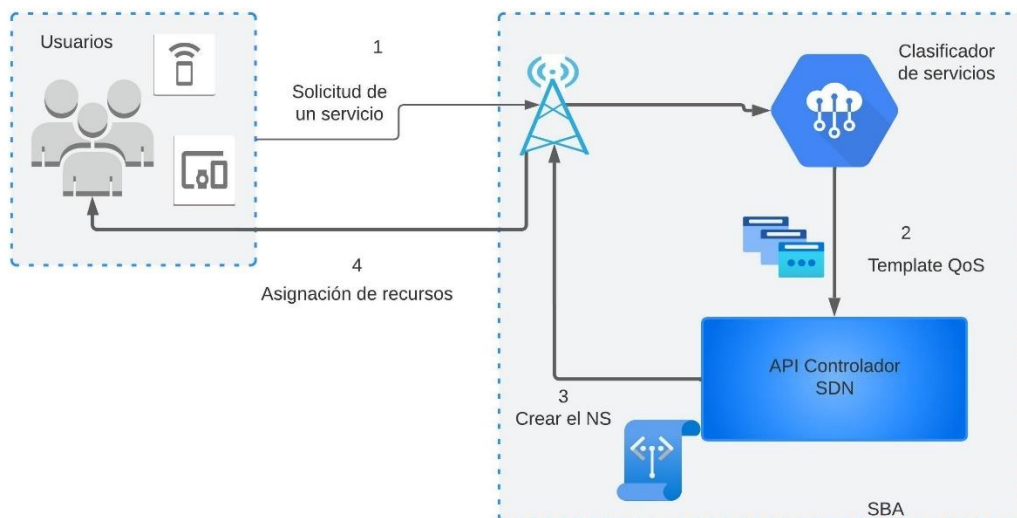
En este capítulo, se propone un esquema de conexión entre dos API: la API del clasificador y la API del controlador SDN. Se detallan cada uno de los bloques del esquema, con especial énfasis en las API, debido a su importancia en la propuesta. Asimismo, se explora en detalle el diseño y la implementación de este esquema de conexión, analizando los desafíos y las oportunidades que presenta para mejorar la gestión de las redes de nueva generación.

### 3.1 Esquema general de la propuesta

En la figura 16 se muestra el esquema general del sistema propuesto para asignar recursos a través de los NS dada una topología de red 5G/B5G. En dicha figura se reflejan los UE, que pueden ser usuarios, sensores, tabletas, cualquier dispositivo que pueda acceder a la red; una Estación Base (BS, por las siglas del término en inglés, *Base Station*); el clasificador de servicios, el cual está ubicado en la nube para que de una cobertura amplia y se disminuya la latencia; y el controlador SDN.

Además, en la figura 16 se representa el momento en que un UE solicita un servicio (1) al operador de red o proveedor de servicios. Dicha solicitud llega a la BS, a través de la RAN y pasa a la API del clasificador de servicios. Una vez clasificado el servicio solicitado por parte del UE, la API del clasificador, manda un formato (*template*) que contiene los parámetros de QoS (latencia, *jitter*, pérdida de paquetes y ancho de banda) necesarios para que el servicio funcione sin inconveniente, según los estándares (2). El controlador de SDN recibe este *template*, crea el NS (3) en la topología de red que ya conoce, y establece la ruta con el objetivo de que el NS asignado cumpla con los SLA correspondientes con la finalidad de evitar violaciones

de estos. Posteriormente, se asignan los recursos de red (4) necesarios para el cumplimiento del servicio. Todo el caso de uso descrito anteriormente cae dentro de la SBA de 5G, y por ello se simulan algunas de las VNF que infieren en todo el proceso explicado.



**Figura 16 Esquema general propuesto [Elaboración propia].**

Supongamos que un usuario de una red 5G/B5G solicita un servicio de *streaming* de video en alta definición a través de su dispositivo móvil. La solicitud de servicio llega a la BS y pasa a la RAN, para dirigirse al clasificador de servicios. El clasificador de servicios identifica el servicio solicitado como "*streaming* de video en alta definición" y envía los parámetros de QoS necesarios para que el servicio funcione correctamente, como el retardo, ancho de banda y la pérdida de paquetes, al controlador de SDN. El controlador de SDN recibe los parámetros de QoS y utiliza la información de la topología de red que ya conoce para crear un NS que cumpla con los requisitos de QoS y establecer la ruta adecuada para el tráfico de video (Pérez Hernández, 2021). Posteriormente, se asignan los recursos de red necesarios, como la asignación de ancho de banda y la priorización de tráfico, para garantizar el cumplimiento del servicio. Finalmente, el usuario puede hacer uso del servicio de *streaming* de video en alta definición de acuerdo con el nivel de SLA previamente contratado con su proveedor de servicio, esto como resultado del proceso de asignación de recursos de red en una red 5G/B5G.

En las redes de nueva generación, es importante destacar, que los servicios de red y los NFV, por lo general, están implementados en la nube (*cloud computing*). El sistema propuesto, también considera el uso de la

nube para el empleo de la API para la interconexión. Se decide usar API porque se ha demostrado que es una herramienta capaz de interactuar y servir a los pedidos de las demás herramientas que trabajan en la infraestructura del operador de red (Schmelz & Nok, 2017). La API del sistema que se propone se crea a través de una estructura conceptual y tecnológica de soporte definido (*web framework*), como *Flask*, *Django*, entre otros. Como alternativa se puede utilizar algún proveedor de servicios de la nube como *Amazon Web Service (AWS)*, *Microsoft Azure* o *Google Cloud*. (AWS, 2023; Google, 2023; Microsoft, 2023)

Esta última opción es bastante recomendable cuando se requiere implementar un sistema para asignar recursos de manera global, por la cantidad de servidores en distintas partes del mundo que ofrecen estas compañías. Además, apoyarse en proveedores de servicios en la nube implica una disminución considerable de la latencia.

En este sentido, la figura 16 refleja la complejidad del sistema propuesto para la asignación de recursos en una red 5G/B5G, ya que se requiere de una serie de pasos para asegurar que el servicio solicitado por el usuario final se cumpla satisfactoriamente. Como se mencionó previamente, el clasificador de servicios juega un papel fundamental en la asignación de recursos, ya que es el encargado de identificar el tipo de servicio solicitado por el UE y enviar los parámetros de QoS correspondientes al controlador SDN. Esto implica que el clasificador debe ser capaz de clasificar eficientemente los diferentes tipos de servicios que pueden ser solicitados por los usuarios. Esto último, es un desafío adicional en el contexto de una red 5G/B5G, ya que presenta una mayor heterogeneidad de servicios.

Por otro lado, la asignación de recursos en una red 5G/B5G también implica una mayor complejidad debido a la necesidad de considerar las distintas tecnologías y protocolos que se utilizan en la red. Por ejemplo, en una red 5G/B5G se utilizan tecnologías de NFV y SDN, lo que implica que la asignación de recursos debe considerar la configuración de VNF y la creación de NS para cada uno de los servicios solicitados por los usuarios. En este sentido, la figura 16 también refleja la importancia de la SBA de 5G, ya que es en este marco donde se pueden simular las diferentes VNF y NS que intervienen en el proceso de asignación de recursos en una red 5G/B5G.

### **3.2 Solicitud de un servicio 5G/B5G**

En las redes de nueva generación, el proceso de solicitud de un servicio y la asignación de recursos de red involucra la interacción de varias entidades, desde el UE hasta las distintas VNF de la SBA de la red 5G.



Cuando el UE solicita o demanda un servicio, esta solicitud se envía a la BS correspondiente a través de la RAN, tal como se ilustra en la figura 11 del capítulo 2. A partir de ahí, la solicitud viaja a través del plano de usuario de la SBA, pasando por las distintas VNF que forman parte de la infraestructura de la red.

Es crucial tener en cuenta las VNF que tienen un mayor impacto en todo el proceso de solicitud de un servicio y asignación de recursos de red para asegurar el cumplimiento de los SLA. Como se explicó en el capítulo 2 de este trabajo, las distintas VNF de la SBA están conectadas mediante las SBI y se comunican entre sí a través de API para intercambiar información relevante al servicio solicitado.

En este sentido, al diseñar y optimizar el esquema propuesto para la asignación de recursos en una red 5G/B5G, se debe prestar especial atención a las VNF que desempeñan un papel fundamental en el flujo de información y en la toma de decisiones para cumplir con los SLA establecidos. Esto implica considerar las características y requisitos específicos de cada VNF, así como la interacción entre ellas, con el objetivo de garantizar una asignación eficiente y efectiva de los recursos de red para satisfacer las demandas de los servicios solicitados por los usuarios.

Es importante destacar que la solicitud de servicios por parte del UE está determinada por los parámetros clave de desempeño (KPI) y los indicadores clave de calidad (KQI) asociados a dichos servicios. Los KPI representan métricas objetivas que miden el rendimiento de los servicios (latencia, *jitter*, la pérdida de paquetes y el ancho de banda). Por otro lado, los KQI son parámetros de calidad que evalúan la experiencia del usuario, considerando aspectos subjetivos como la satisfacción, la fiabilidad y la disponibilidad de red.

Aunque algunos autores no les otorgan la debida importancia, los KQI son fundamentales para mantener una QoS adecuada y cumplir con los SLA establecidos. Estos parámetros reflejan la percepción y satisfacción del usuario final, lo que incide directamente en la experiencia de uso de los servicios. Ignorar los KQI podría llevar a una subestimación de la calidad de la red y, en última instancia, a incumplir con los SLA (Preciado-Velasco, 2021).

Por lo tanto, en el contexto de la asignación de recursos en redes 5G/B5G, es esencial considerar tanto los KPI como los KQI para asegurar una QoS adecuada. La combinación de estos parámetros proporciona una visión completa y holística de la calidad del servicio, permitiendo a los proveedores de servicios ajustar y optimizar sus recursos de red de manera eficiente para cumplir con los SLA y garantizar una QoE satisfactoria para los usuarios.

### 3.3 Clasificador de servicios 5G/B5G

El clasificador de servicios desempeña un papel fundamental en la gestión y asignación eficiente de recursos en las redes de nueva generación, como la red 5G/B5G. Se trata de una componente esencial que se encuentra ubicada en la nube y tiene como objetivo principal clasificar los diferentes tipos de servicios solicitados por los usuarios finales.

Para esta fase del esquema, se emplea el clasificador de servicios 5G/B5G creado en (Preciado-Velasco et al., 2021) a través de una API. En la figura 17 se muestra el esquema general que utiliza este clasificador. Este clasificador pasó por una fase de entrenamiento con un *dataset* de 165x13, validación y prueba; donde se comprobó a través de métricas de rendimiento que el algoritmo de RF obtuvo los mejores resultados. (Preciado-Velasco et al., 2021).

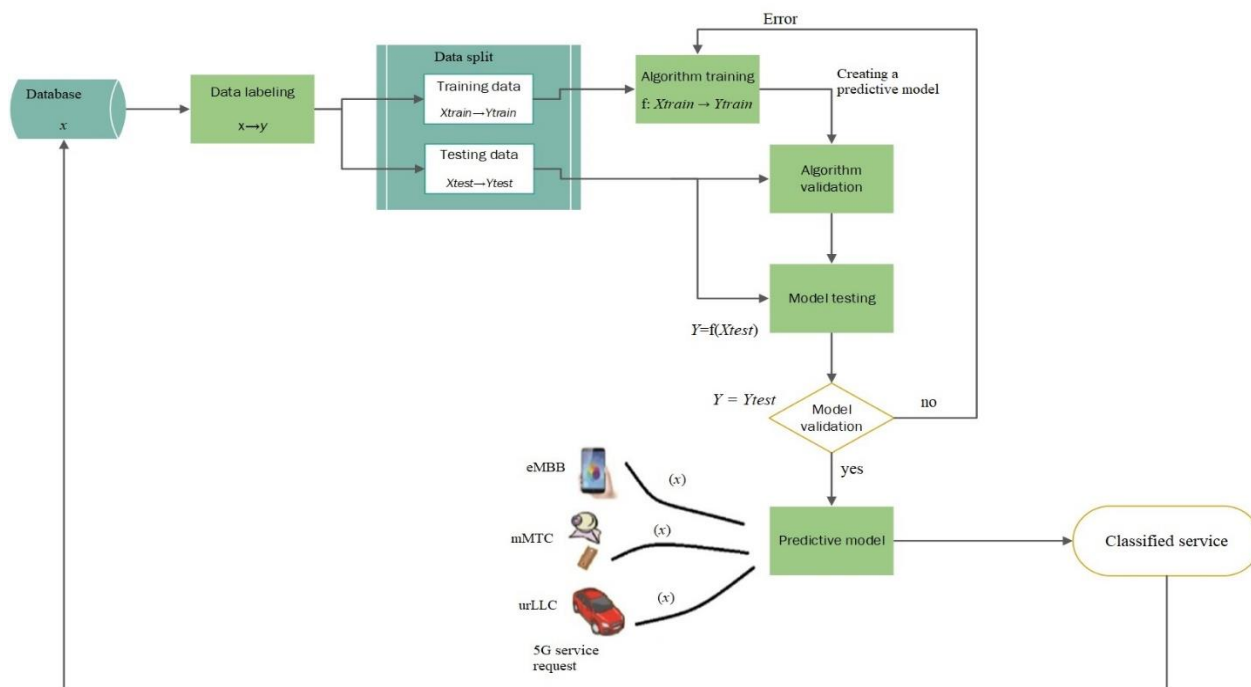


Figura 17 Esquema general del Clasificador de servicios 5G/B5G (Preciado-Velasco et al., 2021).

El clasificador de servicios actúa como un intermediario entre el usuario y la red, analizando la solicitud de servicio realizada por el usuario y determinando su naturaleza y requisitos específicos. Esta clasificación se basa en una serie de parámetros y características, como el tipo de aplicación, los patrones de tráfico, la sensibilidad a la latencia, entre otros.

Una vez que el clasificador de servicios ha identificado y clasificado el servicio solicitado, se encarga de comunicar al controlador de SDN los parámetros de calidad de servicio (QoS) necesarios para su correcta provisión. Estos parámetros incluyen aspectos como la latencia, el ancho de banda, la pérdida de paquetes y el *jitter*, los cuales son fundamentales para garantizar un rendimiento óptimo y cumplir con los SLA establecidos.

En resumen, el clasificador de servicios desempeña un rol crítico en la asignación eficiente de recursos en las redes 5G/B5G. Su capacidad para identificar y clasificar los servicios solicitados por los usuarios, junto con la comunicación de los parámetros de QoS correspondientes al controlador de SDN, permite garantizar una experiencia de uso satisfactoria, optimizando la utilización de los recursos de red y cumpliendo con los SLA establecidos.

### **3.4 Controlador SDN**

El controlador de SDN es una pieza clave en la gestión y control de las redes de nueva generación. Se trata de una entidad centralizada que se encarga de controlar y supervisar la configuración y el funcionamiento de la red de manera programática y basada en políticas. El controlador de SDN actúa como el cerebro de la red, puede tomar decisiones sobre la asignación de recursos y el enrutamiento de tráfico. Además, puede permitir que otras aplicaciones o servicios también puedan generar dichas decisiones y así configurar las rutas de tráfico a través de él. Esto con la finalidad de garantizar un rendimiento óptimo y cumplir con los requisitos de QoS establecidos por los servicios y aplicaciones.

Este componente utiliza información proporcionada por el clasificador de servicios, que identifica y clasifica los diferentes tipos de servicios solicitados por los usuarios, con la finalidad de establecer las rutas más adecuadas y asignar los recursos necesarios en función de los parámetros de QoS requeridos. Asimismo, el controlador de SDN se encarga de monitorear el estado de la red y cuenta con mecanismos que permiten realizar ajustes de manera dinámica y en tiempo real para optimizar el rendimiento de acuerdo con los SLA.

Gracias a la flexibilidad y la capacidad de programación del controlador de SDN, se logra una gestión más eficiente y ágil de la red, lo que permite una rápida adaptación a los cambios en la demanda de servicios y optimiza la utilización de los recursos disponibles. Además, su enfoque centralizado simplifica la

administración y reduce la complejidad operativa de la red, lo cual brinda un mayor control y mejora la experiencia del usuario final.

En conclusión, el controlador de SDN juega un papel fundamental en la gestión y control de las redes 5G/B5G al proporcionar una visión global de la red y tomar decisiones inteligentes para garantizar un rendimiento óptimo y cumplir con los requisitos de QoS. Su capacidad para coordinar y controlar de manera programática los recursos de red brinda flexibilidad, escalabilidad y eficiencia, contribuyendo al éxito de las aplicaciones y servicios en entornos de nueva generación.

### **3.5 Asignación de recursos**

La asignación de recursos de red desempeña un papel fundamental en el funcionamiento eficiente y efectivo de las redes de nueva generación, como la red 5G/B5G. Se refiere al proceso de asignar y distribuir adecuadamente los recursos de red, como ancho de banda, capacidad de procesamiento y almacenamiento, entre los diferentes servicios, aplicaciones y usuarios, con el objetivo de garantizar un rendimiento óptimo y cumplir con los requisitos de QoS establecidos, y así cumplir con los SLA establecidos.

En un entorno de red dinámico y heterogéneo, la asignación de recursos se vuelve especialmente desafiante debido a la diversidad de servicios y aplicaciones que comparten una infraestructura común. Cada servicio puede tener requisitos de QoS específicos, como baja latencia, alta velocidad de transferencia, baja pérdida de paquetes y capacidad garantizada, lo que requiere una asignación cuidadosa y equitativa de los recursos de red.

La asignación de recursos de red se basa en una combinación de técnicas y algoritmos que tienen en cuenta diversos factores, como la demanda de servicios, la disponibilidad de recursos, las políticas de priorización y los acuerdos de nivel de servicio (SLA). Los controladores de SDN y las tecnologías de NFV desempeñan un papel clave en este proceso, al permitir una gestión flexible y programática de los recursos de red.

Además, es importante mencionar que la asignación de recursos de red no solo se centra en la eficiencia de la utilización de los recursos, sino también en garantizar una distribución justa y equitativa de los mismos. Esto implica considerar factores como la priorización de servicios críticos, el equilibrio de carga,

la optimización de la utilización de recursos y la capacidad de adaptación a las fluctuaciones en la demanda.

Uno de los casos más comunes en la red, es que un usuario conectado, no haya solicitado servicios que demanden grandes recursos de red, por ejemplo, un gran ancho de banda. Sin embargo, el proveedor le tiene asignado un ancho de banda mayor al que realmente necesita para ese momento. Esto puede dar el caso de que un usuario que realmente necesite un mayor ancho de banda, el proveedor no pueda asignar, o si los asigna, puede provocar una saturación de la red.

En resumen, la asignación de recursos de red es un proceso clave para lograr un rendimiento óptimo y garantizar una calidad de servicio adecuada en las redes de nueva generación. Mediante la utilización de técnicas avanzadas de gestión y control, se busca optimizar la utilización de los recursos, equilibrar la carga de trabajo y cumplir con los requisitos de QoS establecidos, brindando una experiencia satisfactoria a los usuarios finales y asegurando el éxito de las aplicaciones y servicios en entornos de red avanzados.

### **3.6 Recomendaciones para implementar el sistema de asignación de recursos de red**

En este trabajo se ha propuesto un sistema para asignar recursos que cumpla con los SLA, mediante NS. En el epígrafe 3.1 se mostró un esquema general (ver figura 16), para el cual se deben seleccionar: el clasificador de servicios, el controlador SDN, y el ambiente de simulación.

Como se mencionó, el clasificador de servicios a utilizar en este trabajo es el modelo predictivo creado en (Preciado-Velasco et al., 2021). Hay que destacar que, para este problema, se puede emplear otro tipo de clasificador, basado en tráfico, por ejemplo. Existen una gran variedad de controladores SDN, es por ello por lo que, la Tabla 1 muestra una comparación entre algunos controladores, en cuanto a características principales.

Como se puede ver en la Tabla 1, todos estos controladores son de código abierto y de uso gratuito. También todos tienen buena documentación y apoyo comunitario. Sin embargo, difieren en términos de lenguaje de programación, escalabilidad, características y seguridad.

**Tabla 1 Comparativa de controladores SDN [Elaboración propia].**

Controlador SDN	ONOS	ODL	RYU	Floodlight
Lenguaje de programación	JAVA	JAVA	Python	JAVA
Código abierto	Sí	Sí	Sí	Sí
Escalabilidad	Alta	Alta	Media	Media
Seguridad	Buena	Buena	Básica	Básica
Documentación	Buena	Buena	Poca	Poca
Soporte de comunidad	Bueno	Bueno	Poco	Poco
Características	Extensas	Extensas	Básicas	Básicas
Costo	Gratis	Gratis	Gratis	Gratis

- ONOS es un controlador basado en Java que está diseñado para redes grandes. Tiene una amplia gama de características, que incluyen monitoreo de red, control de tráfico y seguridad. ONOS también está bien documentado y tiene una gran comunidad de usuarios. Sin embargo, su implementación y administración pueden resultar costosas, por los recursos de hardware que demanda (ONOS, 2023).
- ODL es otro controlador basado en Java que está diseñado para redes grandes. Es similar a ONOS en cuanto a sus características y escalabilidad. Sin embargo, ODL no está tan bien documentado como ONOS y no tiene una comunidad de usuarios tan grande. Esto se debe a que muchas empresas empezaron a modificar el código base de ODL para usarlo en sus propios dispositivos (OpenDayLight, 2023).
- RYU es un controlador basado en Python que está diseñado para redes de tamaño mediano. Tiene un conjunto de funciones más pequeño que ONOS u ODL, pero es más fácil de implementar y administrar. RYU también está bien documentado y tiene una comunidad de usuarios pequeña pero activa (RYU, 2023).
- *Floodlight* es un controlador basado en Java diseñado para redes pequeñas. Es el más simple de los controladores enumerados aquí y tiene la menor cantidad de funciones. Sin embargo, también es el más fácil de implementar y administrar. *Floodlight* también está bien documentado y tiene una gran comunidad de usuarios (FloodLight, 2023).

En última instancia, el mejor controlador SDN dependerá de las necesidades y requisitos específicos. Si se tiene una red grande y necesita un controlador con una amplia gama de funciones, ONOS u ODL pueden ser una buena opción. Si tiene una red de tamaño mediano y está buscando un controlador fácil de implementar y administrar, entonces RYU puede ser una buena opción. Si se tiene una red pequeña y se busca un controlador simple y asequible, entonces *Floodlight* puede ser una buena opción.

Una vez definido el controlador, el próximo paso es definir la plataforma donde se realizarán las simulaciones para poner en práctica el sistema propuesto. En la Tabla 2 se muestra una comparación de algunas plataformas usadas para las simulaciones de SDN.

**Tabla 2 Comparativa de plataformas para simular SDN [Elaboración propia].**

Plataforma	Soporte <i>OpenFlow</i>	Características	Precio
<i>Mininet</i>	Sí	De código abierto, rápido y fácil de usar.	Gratis
GNS3	Sí	Interfaz gráfica de usuario, compatible con una amplia gama de dispositivos de red.	Gratis
NS-3	Sí	Conjunto completo de protocolos y modelos, se puede utilizar para simular una amplia gama de redes.	Gratis
EstiNet	Sí	Comercial, admite una amplia gama de controladores SDN y aplicaciones de red.	De pago
OpenNet	Sí	Código abierto, admite una amplia gama de controladores SDN y aplicaciones de red.	Gratis

Esta es solo una pequeña selección de las muchas plataformas de simulación SDN disponibles. Algunas otras plataformas populares incluyen OMNeT++ y OPNET. A continuación, se muestran algunos detalles adicionales sobre cada plataforma:

- *Mininet*: es un emulador de red de código abierto que crea una red virtual utilizando un solo kernel de Linux. Es fácil de usar y rápido, por lo que es una buena opción para experimentos rápidos y creación de prototipos (Mininet, 2023).
- GNS3: es un simulador gráfico de red que le permite diseñar topologías de red complejas y probarlas en un entorno virtual. Es compatible con una amplia gama de dispositivos de red, por lo

que es una buena opción para simulaciones más complejas (GNS3, 2023).

- NS-3: es un simulador de red de eventos discretos que proporciona un conjunto completo de protocolos y modelos para simular varios tipos de redes. Es una poderosa herramienta para simular una amplia gama de escenarios y topologías de red (NS-3, 2023).
- EstiNet: es un simulador SDN comercial que admite varios controladores SDN y aplicaciones de red. Es una buena opción para las organizaciones que necesitan una plataforma de simulación más potente y rica en funciones (EstiNet, 2023).
- OpenNet: es un simulador SDN de código abierto que admite una amplia gama de controladores SDN y aplicaciones de red. Es una buena opción para las organizaciones que buscan una plataforma de simulación gratuita y de código abierto (OpenNet, 2016).

En última instancia, la mejor plataforma de simulación SDN dependerá de las necesidades y requisitos específicos para los que se estén trabajando.

En los últimos años ha ocurrido una migración de las aplicaciones a la nube (*cloud computing*). En este trabajo se propone un sistema que, por ende, se piensa que se encuentre distribuido por todo el país; por lo que se necesita utilizar estas técnicas para un desarrollo de forma satisfactoria.

En la actualidad existen tres gigantes encargados de la computación en la nube: *Amazon Web Services (AWS)*, *Microsoft Azure* y *Google Cloud*. Existen muchas empresas que se dedican a ofrecer servicios de la nube. Sin embargo, en la Tabla 3 se muestra una comparativa de algunas de las más famosas (tomado de los sitios oficiales de *AWS*, *Microsoft Azure* y *Google Cloud*) (AWS, 2023; Google, 2023; Microsoft, 2023)

Como se puede observar en la Tabla 3, las plataformas en la nube tienen sus propias fortalezas y debilidades. La mejor plataforma depende de las necesidades y requisitos específicos de la aplicación en cuestión.

Para implementar de manera exitosa el sistema propuesto de asignación de recursos en redes 5G/B5G, es fundamental tener en cuenta una serie de recomendaciones y buenas prácticas.



**Tabla 3 Comparativa de los principales proveedores de servicios en la nube.**

Característica	Amazon Web Services (AWS)	Microsoft Azure	Google Cloud
Madurez de la empresa	AWS ha existido por más tiempo y tiene el ecosistema más maduro. con más del 30% de cuota de mercado	Azure es una plataforma más nueva, pero ha estado creciendo rápidamente en los últimos años con más del 15% de cuota de mercado.	Google Cloud es la plataforma más nueva de las tres, pero está creciendo rápidamente con una cuota de mercado de alrededor del 8%.
Servicios ofrecidos	Ofrece la más amplia gama de características y servicios. Tiene más de 200 servicios, incluidos cómputo, almacenamiento, redes, bases de datos, ML y AI.	Ofrece una amplia gama de características y servicios, pero no tantos como AWS. Tiene más de 100 servicios, incluidos computación, almacenamiento, redes, bases de datos, análisis, ML y AI.	Ofrece una gama más limitada de características y servicios que AWS o Azure, pero constantemente agrega nuevas características. Tiene más de 60 servicios, incluidos cómputo, almacenamiento, redes, bases de datos, análisis, ML y AI.
Comunidad	AWS tiene la mayor comunidad de usuarios y desarrolladores. Hay más de 1 millón de profesionales certificados por AWS.	Tiene una gran comunidad de usuarios y desarrolladores, pero no es tan grande como la comunidad de AWS. Hay más de 500.000 profesionales certificados de Azure.	Tiene una pequeña comunidad de usuarios y desarrolladores, pero está creciendo rápidamente. Hay más de 100.000 profesionales certificados de Google Cloud.
Precio	Generalmente es la plataforma en la nube más cara. Sin embargo, ofrece descuentos para compromisos a largo plazo.	Es generalmente la plataforma en la nube de precio medio. Ofrece descuentos para compromisos a largo plazo y, a veces, ofrece pruebas gratuitas.	Es generalmente la plataforma en la nube menos costosa. Ofrece descuentos para compromisos a largo plazo y, a veces, ofrece pruebas gratuitas.
Rendimiento	AWS tiene el mejor rendimiento de las tres plataformas en la nube.	Tiene un buen rendimiento, pero no es tan bueno como el rendimiento de AWS.	Tiene un rendimiento justo. Tiene una red global de centros de datos, pero no ofrece tantos servicios optimizados para el rendimiento como AWS o Azure.
Seguridad	Tienen una seguridad sólida.		
Soporte	Ofrecen soporte 24 horas, por vía llamada, chat o email		Ofrece soporte 24 horas, por llamada o email. No ofrece soporte a través de chat.

Estas consideraciones ayudarán a garantizar una implementación eficiente, robusta y acorde a los requisitos de QoS y SLA. A continuación, se presentan algunas recomendaciones clave para tener en cuenta durante el proceso de implementación:

1. Definir una arquitectura adecuada: Es necesario diseñar una arquitectura que se ajuste a las necesidades y objetivos específicos de la red 5G/B5G. Esto implica seleccionar los componentes apropiados, como el clasificador de servicios y el controlador de SDN, y establecer una estructura que permita una gestión eficiente de los recursos de red.
2. Considerar la escalabilidad: Dado el crecimiento exponencial de dispositivos y servicios en las redes 5G/B5G, es fundamental garantizar la escalabilidad del sistema propuesto. Esto implica diseñar una infraestructura capaz de soportar un aumento en la demanda de recursos y servicios sin comprometer el rendimiento y la calidad de servicio.
3. Realizar pruebas exhaustivas: Antes de implementar el sistema en un entorno de producción, se recomienda realizar pruebas exhaustivas para validar su funcionamiento y desempeño. Esto incluye pruebas de carga, pruebas de estrés y simulaciones de escenarios reales para evaluar su capacidad de respuesta y su capacidad para cumplir con los SLA establecidos.
4. Establecer políticas y reglas claras: Es importante definir políticas y reglas claras para la asignación de recursos, el enrutamiento de tráfico y la gestión de QoS. Estas políticas deben estar alineadas con los objetivos de la red y los requisitos de los servicios, y deben ser flexibles para adaptarse a cambios y nuevas demandas.
5. Implementar mecanismos de monitoreo y gestión: Para asegurar un rendimiento óptimo y garantizar el cumplimiento de los SLA, es esencial contar con mecanismos de monitoreo y gestión continuos. Esto permitirá detectar posibles problemas o violaciones de los SLA y tomar medidas correctivas de manera oportuna.
6. Mantener actualización: Dado que las redes 5G/B5G son tecnologías en constante evolución, es importante mantener actualizado el sistema con los avances y las tendencias del sector. Esto permitirá adaptar el sistema propuesto a las nuevas tecnologías y desafíos futuros.

El seguir estas recomendaciones, maximizará la eficiencia, la confiabilidad y el cumplimiento de los SLA en la implementación del sistema propuesto para la asignación de recursos en redes 5G/B5G.

### 3.7 Conclusiones

En este capítulo se presentó el esquema general de una propuesta para asignar recursos de red mediante NS y cumplir con los SLA. Además, se describió cada parte del esquema con sus principales funciones a desarrollar, así como algunas recomendaciones para la implementación del sistema, por lo que se puede concluir que:

- Se presentó un sistema para la asignación de recursos en redes 5G/B5G, que utiliza un clasificador de servicios y un controlador de SDN para garantizar una adecuada gestión de los recursos y cumplir con los requisitos de calidad de servicio.
- La implementación de este sistema ofrece beneficios significativos, como una mayor eficiencia en la asignación de recursos, una gestión centralizada y una rápida adaptación a los cambios en la demanda de servicios, contribuyendo a una mejor experiencia del usuario final.
- La integración de APIs y el uso de tecnologías de *cloud computing* en la implementación del sistema proporcionan flexibilidad y escalabilidad, permitiendo una interconexión eficiente entre los diferentes componentes y una mejor utilización de los recursos disponibles.
- La asignación de recursos en redes 5G/B5G implica considerar tanto los parámetros de calidad de servicio (QoS) como los parámetros de calidad en la red (KQI), asegurando una adecuada calidad en los servicios y evitando violaciones de los acuerdos de nivel de servicio (SLA).
- Para lograr una implementación exitosa del sistema propuesto, se recomienda seguir buenas prácticas, como definir una arquitectura adecuada, realizar pruebas exhaustivas, establecer políticas claras y mantenerse actualizado con las últimas tendencias y avances en el campo de las redes 5G/B5G.

## Capítulo 4. Validación del sistema propuesto

---

En el Capítulo 4, se presenta el escenario de validación de la propuesta del sistema de la API para clasificar servicios y asignar los NS en redes 5G/B5G. El objetivo principal de la validación de la propuesta es evaluar la eficacia y capacidad de mejora de la QoS así como el cumplimiento de los acuerdos de nivel de servicio (SLA) creado mediante el uso de esta solución.

Para llevar a cabo esta validación, se emplea la simulación como método de estudio. La simulación permite recrear un entorno controlado y reproducible que simula las condiciones de una red 5G/B5G. En este caso, se utiliza el software *Mininet*, herramienta ampliamente reconocida en el ámbito de las SDN, que permite emular topologías de red y modificar los enlaces de manera programática a través de un controlador de SDN.

La elección de la simulación como método de validación proporciona varias ventajas. En primer lugar, permite realizar experimentos en un entorno controlado, lo que facilita la medición y la evaluación de diferentes métricas de rendimiento. Además, al utilizar *Mininet* y un controlador de SDN, se puede tener un control preciso sobre la configuración y el comportamiento de la red, lo que permite realizar pruebas exhaustivas y evaluar el impacto de diferentes escenarios en la calidad de servicio y el cumplimiento de los SLA.

En resumen, en este capítulo se presenta el escenario de validación de la propuesta del sistema de la API para clasificar servicios y asignar los NS en redes 5G/B5G, utilizando la simulación como método de estudio y el software *Mininet* como herramienta para emular topologías de red y modificar los enlaces. Este enfoque de validación permitirá evaluar la eficacia y la capacidad de mejora de la QoS y el cumplimiento de los SLA proporcionados por esta solución.

### 4.1 Escenario de validación

Para llevar a cabo la simulación del sistema de asignación de NS teniendo en cuenta los recursos de red disponibles, se elabora un entorno virtual. Es necesario elegir el Entorno de Desarrollo Integrado (IDE, por las siglas del término en inglés, *Integrated Development Environment*) adecuado para llevar las tareas de programación, la plataforma donde se desarrolla el entorno virtual de la simulación, el software para

probar la API REST, el lenguaje de programación, el controlador SDN que se va a encargar de direccionar el tráfico y el *framework* para desarrollar la API.

Se utiliza el IDE de *Microsoft Visual Studio Code* (VSC) para llevar las tareas relacionadas a la programación. Como se mencionó anteriormente, uno de los principales objetivos del sistema propuesto es implementarlo en la nube.

Para el desarrollo de este trabajo, se emplea *Google Cloud*, y se creó una VM de tipo *e2-medium* para llevar a cabo las simulaciones necesarias para validar el sistema. Una de las principales ventajas por las que se decide usar *Google Cloud* es por el enfoque en la AI y ML, por el ofrecimiento de servicios y herramientas avanzadas para el desarrollo y despliegue de modelos de ML, como *Google Cloud Machine Learning Engine*. Otras ventajas respecto a los otros proveedores de nube pueden ser: (clarcat, 2023)

- Precios accesibles y mejores que el de sus competidores: Se paga solamente el tiempo que se utiliza, y ofrecen descuentos para planes de trabajo de largo plazo.
- Rendimiento mejorado: Es prioritario tener una gran capacidad y, las máquinas de Google pueden proveer la capacidad demandada.
- Seguridad: La seguridad de *Google* funciona de extremo a extremo, con una codificación bastante compleja que mantendrá los datos a salvo. Esta plataforma brinda un 100% de fiabilidad y disponibilidad, sin ningún periodo inactivo programado, usando sus redes privadas, por lo que los ciberataques son prácticamente imposibles.

En capítulos anteriores se mencionó el desarrollo de una API REST, la cual se programa en el *framework* de desarrollo web *Flask* (Flask, 2023) y para hacer las pruebas pertinentes al funcionamiento de la API se elige *Postman* (Postman, 2023). Las principales características para la elección de *Flask* que se tienen en cuenta son:

1. Simplicidad y minimalismo: *Flask* es conocido por su enfoque minimalista y su facilidad de uso. Tiene una curva de aprendizaje suave, lo que lo hace ideal tanto para principiantes como para desarrolladores experimentados.

2. Flexibilidad y modularidad: proporciona una base sólida pero flexible para construir aplicaciones web. Se puede elegir las bibliotecas y extensiones que se necesiten según los requerimientos, lo que permite crear aplicaciones altamente personalizadas y optimizadas.
3. Amplia comunidad y ecosistema: tiene una comunidad activa y vibrante de desarrolladores que contribuyen con bibliotecas, extensiones y recursos útiles. Hay una amplia gama de extensiones disponibles que facilitan tareas comunes, como autenticación de usuarios, manipulación de formularios y acceso a bases de datos. Además, *Flask* está bien documentado y cuenta con una gran cantidad de tutoriales y ejemplos en línea, lo que facilita el aprendizaje y la resolución de problemas.

*Mininet* es un popular emulador de red que se utiliza para una variedad de trabajos de simulación. Es rápido, flexible y fácil de usar, por lo que es una excelente opción para una variedad de proyectos. A continuación, se muestran una serie de razones por las cuales se elige *Mininet*:

- Velocidad: Es muy rápido, lo que la hace ideal para la creación rápida de prototipos y pruebas. Puede crear una red virtual en segundos y puede cambiar fácilmente la topología y la configuración de la red según sea necesario.
- Flexibilidad: Es muy flexible y puede usarla para simular una amplia variedad de redes. Puede crear topologías personalizadas, ejecutar *software* real en los nodos de su red e incluso integrar *Mininet* con hardware físico.
- Facilidad de uso: Puede crear una red virtual con unos pocos comandos simples y puede usar la CLI de *Mininet* para controlar su red. Además, se pueden asignar pesos a los enlaces que contengan características de QoS como son la latencia, ancho de banda, *jitter* y pérdida de paquetes.
- Rentabilidad: Es una herramienta gratuita y de código abierto, lo que la convierte en una opción rentable para el trabajo de emulación.
- Portabilidad: Se puede ejecutar en una variedad de plataformas, incluyendo Linux, macOS y Windows. Esto lo convierte en una herramienta portátil que se puede usar en una variedad de dispositivos.

- Soporte de la comunidad: Tiene una comunidad grande y activa de usuarios y desarrolladores. Esto significa que hay una gran cantidad de recursos disponibles para ayudarlo a comenzar con Mininet y para solucionar cualquier problema que pueda encontrar.

En cuanto al controlador SDN utilizado en un escenario de Mininet, *OpenDaylight* (ODL) es un controlador SDN de código abierto que se emplea por una variedad de organizaciones, incluidos operadores, empresas e instituciones de investigación. Resulta sencillo crear una topología de red usando *Python* y las librerías de *Mininet* en el VSC, así como conectar el controlador ODL a dicha topología. Es una opción popular para los controladores SDN porque es:

- Código abierto: *OpenDaylight* es un proyecto gratuito y de código abierto, lo que significa que está disponible para que cualquiera lo use, modifique y redistribuya. Esto lo convierte en una opción rentable para las organizaciones que buscan implementar un controlador SDN.
- Modular: Es una plataforma modular, lo que significa que se puede personalizar para satisfacer las necesidades específicas de diferentes organizaciones.
- Escalable: Significa que se puede utilizar para gestionar redes de cualquier tamaño. Esto lo convierte en una buena opción para organizaciones que tienen redes grandes o complejas.
- Extensible: Se puede ampliar con nuevas características y funcionalidades. Esto lo convierte en una opción a prueba de futuro para las organizaciones que desean mantenerse a la vanguardia.
- Seguridad: Admite una variedad de funciones de seguridad, incluyendo autenticación, autorización y cifrado. Ayuda a proteger su red del acceso no autorizado y las violaciones de datos.
- Rendimiento: Es de alto rendimiento, puede manejar incluso las redes más exigentes con facilidad.
- Manejabilidad: Es fácil de administrar, puede utilizar la GUI o CLI de ODL para configurar su red y solucionar cualquier problema.

## 4.2 Topología de red propuesta

En este epígrafe se presenta la topología de red propuesta para llevar a cabo la validación del sistema de la asignación de NS en redes 5G/B5G. La topología de red es una representación gráfica de la estructura y la conectividad de los elementos que la conforman, y desempeña un papel fundamental en la simulación y evaluación de la propuesta. La topología diseñada se basa en la consideración de diferentes elementos clave, como las estaciones base, el clasificador de servicios, la SBA de 5G, el controlador de SDN y el usuario final. A través de esta topología, se busca simular un entorno realista que permita analizar el desempeño del sistema propuesto y evaluar su capacidad para mejorar la QoS y evitar violaciones de los SLA.

La topología de red propuesta, representada en la figura 18, se basa en el diseño planteado en el capítulo 3 de esta tesis, donde se describe el proceso de solicitud y asignación de recursos de red en 5G/B5G (ver figura 16). En esta topología, se considera la interacción entre diferentes componentes clave, como el UE, la BS, el clasificador de servicios, el controlador de SDN y las VNF de la SBA.

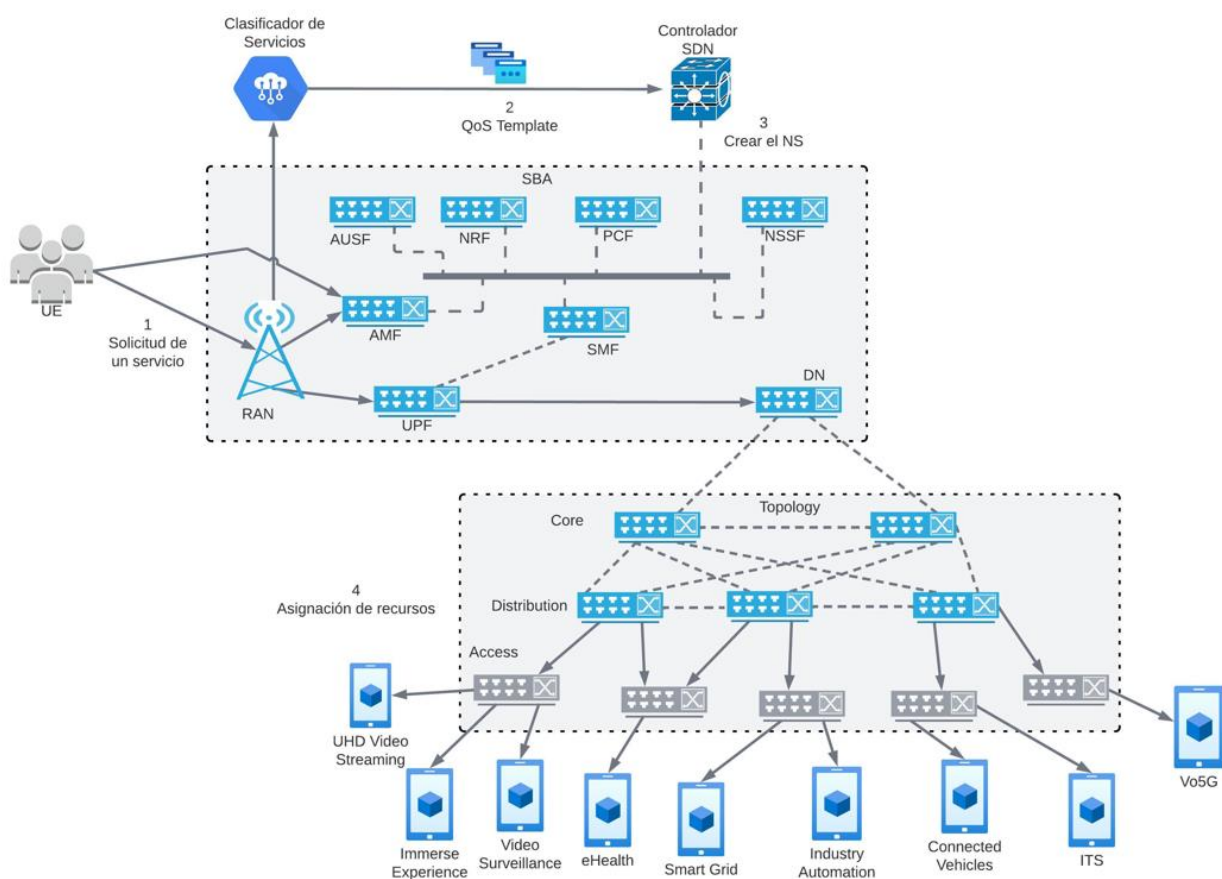


Figura 18 Topología de red propuesta [Elaboración propia].



La validación de un sistema tan completo como el propuesto en la figura 16 plantea diversos desafíos debido a su naturaleza compleja y que aún no se encuentran en su total desarrollo, o incluso el precio de licencias de algunos softwares de simulación es demasiado alto. Con el objetivo de realizar un análisis exhaustivo, se utiliza *Mininet*, que permite diseñar y simular la topología propuesta (figura 18). En esta topología, se pueden identificar nueve servicios básicos que pueden ser solicitados por el UE.

Una característica destacada de esta topología es que, en lugar de asignar los recursos directamente al UE, se realiza la asignación a través de la red hacia el servicio solicitado. Esto crea un entorno virtual para el Network Slice (NS) creado, permite así simular las condiciones necesarias para cumplir con los acuerdos de nivel de servicio (SLA) establecidos. Esta configuración virtualizada de la topología de red proporciona una plataforma idónea para realizar pruebas y validaciones exhaustivas. Al emplear *Mininet*, se logra emular de manera precisa los enlaces y los dispositivos de red, permitiendo evaluar el rendimiento del sistema propuesto en un entorno controlado y reproducible.

Mediante el análisis de esta topología y la simulación de los diferentes servicios, se busca evaluar el comportamiento del sistema de clasificación de servicios y asignación de recursos de red en términos de calidad de servicio (QoS) y cumplimiento de los SLA. Esto permite obtener conclusiones relevantes sobre la efectividad y la eficiencia del sistema en la gestión de los recursos de red y la mejora de la QoS en el contexto de las redes 5G/B5G.

El flujo de datos comienza con el UE que solicita un servicio a la red, el cual se transmite a través de la RAN hasta llegar a la SBA de 5G. En este punto, la información del servicio se dirige a las VNF de la SBA que requieren procesarla, mientras que los KPI/KQI se envían al clasificador de servicios para su evaluación. El clasificador de servicios desempeña un papel crucial al identificar y clasificar el tipo de servicio solicitado por el UE.

Una vez que el servicio ha sido clasificado, el clasificador de servicios envía al controlador de SDN los parámetros de QoS (*Template QoS*) necesarios para garantizar un funcionamiento óptimo del servicio, de acuerdo con los estándares establecidos. El controlador de SDN, a su vez, transmite esta información a las VNF de la SBA, específicamente, a la NSSF; que se encarga de crear el NS con los parámetros de QoS adecuados para el servicio solicitado. Esta información se envía nuevamente al controlador de SDN, que se encarga de la asignación de recursos de red.

El controlador de SDN desempeña un papel fundamental en la topología de red propuesta al actualizar los enlaces de la topología existente, establecer y verificar las posibles rutas, controlar el flujo de datos y realizar otras funciones descritas anteriormente. A través de esta gestión centralizada y programática, el controlador de SDN asegura una asignación eficiente y dinámica de los recursos de red, optimizando el rendimiento y garantizando el cumplimiento de los requisitos de QoS establecidos para evitar violaciones de los SLA.

En resumen, la topología de red propuesta refleja el flujo de información y control en el sistema de clasificación de servicios y asignación de NS en una red 5G/B5G. La interacción entre el clasificador de servicios y el controlador de SDN, junto con las VNF de la SBA, permite la adecuada clasificación de servicios, la creación de NS con parámetros de QoS apropiados y la asignación eficiente de recursos de red. Esta topología se utilizará en la simulación para evaluar el desempeño y la efectividad del sistema propuesto.

### **4.3 Simulaciones del sistema propuesto**

En el entorno propuesto, se llevaron a cabo las simulaciones en dos etapas distintas, cada una con un objetivo específico. La primera etapa se centró en la programación de la API del clasificador de servicios previamente desarrollado. Durante esta etapa, se estableció la comunicación y la interacción entre los componentes del sistema, permitiendo la clasificación adecuada de los servicios solicitados.

Posteriormente, en la segunda etapa, se procedió a configurar la topología de red en *Mininet* para realizar los experimentos necesarios. En este paso, se replicó fielmente la estructura de la red propuesta, incluyendo los enlaces, dispositivos y conexiones entre ellos. Esto permitió simular escenarios realistas y evaluar el desempeño del sistema en un entorno controlado.

Dividir las simulaciones en estas dos etapas permitió un enfoque sistemático y ordenado para validar el funcionamiento del sistema propuesto. La programación de la API del clasificador de servicios estableció las bases para la correcta clasificación de los servicios, mientras que la configuración de la topología de red en *Mininet* proporcionó el entorno necesario para realizar los experimentos y obtener resultados significativos. Es por ello que, para un mejor entendimiento del proceso, a continuación, se dividen en dos secciones: la primera dedicada a la API del clasificador y la segunda al escenario en *Mininet*.

### 4.3.1 API del Clasificador de Servicios 5G/B5G

El clasificador de servicios 5G/B5G, desarrollado por (Preciado-Velasco et al., 2021), desempeña un papel fundamental en la clasificación de nuevos servicios. Sin embargo, para su utilización eficiente, se ha implementado una función adicional que permite exportar el clasificador como un archivo, con la función *load* de la librería *joblib*. De esta manera, puede ser importado como una librería de código y ejecutado de manera conveniente cada vez que sea necesario.

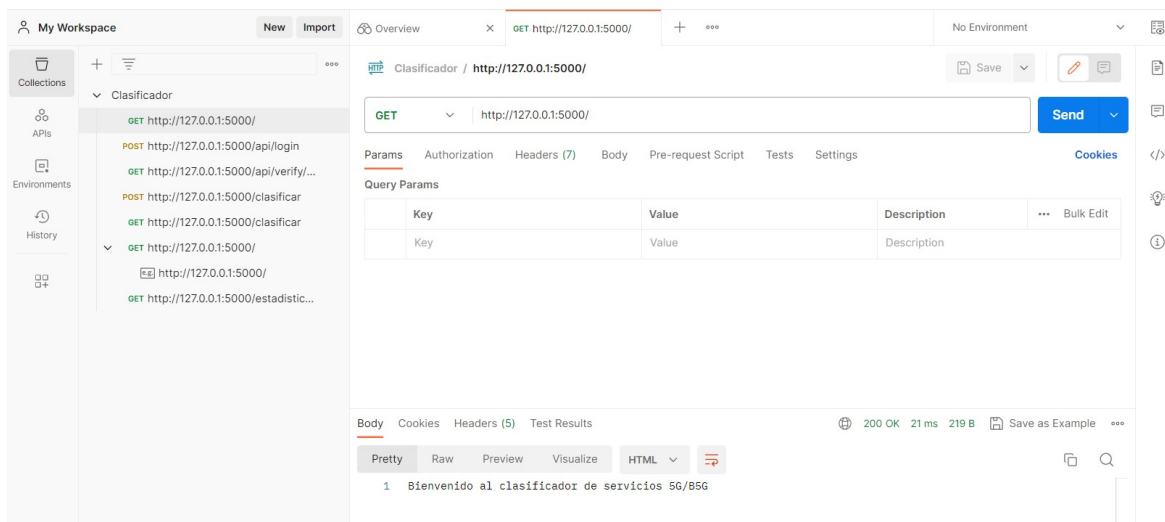
Una vez importado el clasificador, y teniendo conocimiento del servicio solicitado por el UE, se requiere programar una función específica que se encargue de seleccionar el *template* correspondiente con los parámetros de QoS adecuados. Estos parámetros determinan los valores óptimos de ancho de banda, latencia, *jitter* y pérdida de paquetes que deben ser asignados a los enlaces de la red, asegurando así el funcionamiento óptimo de los servicios y evitando cualquier violación de los acuerdos de nivel de servicio (SLA). Esta función desempeña un papel crucial en la asignación de recursos de red de manera eficiente y precisa, asegurando que los servicios se ejecuten sin inconvenientes y cumpliendo con los estándares de calidad establecidos.

En esta etapa del trabajo, se cuenta con un clasificador de servicios completamente funcional, capaz de identificar con aproximadamente un 97% de precisión qué servicio se solicita por el UE, así como de seleccionar el *template* adecuado que contiene los parámetros de QoS específicos para dicho servicio. Ahora es el momento de implementar esta solución en una API REST, tal como se detalló en el capítulo 3 de este trabajo. Para llevar a cabo esta tarea, se decide utilizar el *framework Flask* debido a su simplicidad y agilidad en el desarrollo.

Por el tamaño del código de desarrollo de la API para su referencia, el código está disponible bajo solicitud de quien lo requiera. Para verificar y comprobar el correcto funcionamiento de todo el proceso de la API REST, se utiliza *POSTMAN*. Mediante el uso de solicitudes HTTP, como POST, GET y DELETE, se realizaron pruebas exhaustivas para asegurar que el clasificador devuelva los datos solicitados de manera precisa y consistente. En este contexto, se incluyen imágenes de las solicitudes realizadas en el trabajo, proporcionando una visión clara y visual del proceso.

En la figura 19 se muestra el entorno visual en el *POSTMAN*, una vez que la API se encuentre en funcionamiento. Se observa una solicitud de tipo GET a la dirección <http://127.0.0.1:5000/> y devuelve el

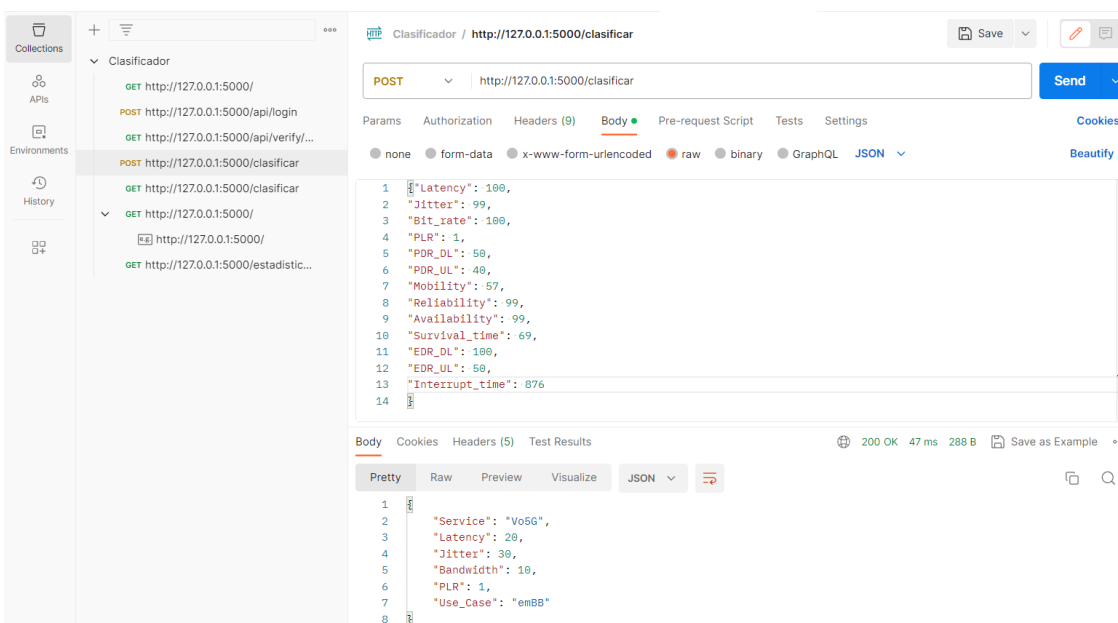
mensaje “Bienvenido al clasificador de servicios 5G/B5G”. Este mensaje es solo para demostrar que la API se encuentra trabajando.



**Figura 19 Entorno visual en POSTMAN.**

Para enviar solicitudes de servicio a la API del clasificador, se empleó un método POST, a la dirección `http://127.0.0.1:5000/clasificar`. En este proceso, el usuario debe introducir los KPI/KQI y enviar la solicitud. La figura 20 ilustra visualmente este flujo, mostrando un ejemplo en concreto. Como resultado, el clasificador determina que se trata de un servicio de Vo5G y genera el correspondiente *template* de QoS con los parámetros necesarios para evitar violaciones de los SLA en los enlaces de la red. Este proceso garantiza una correcta clasificación de los servicios y permite la asignación adecuada de los recursos de red.

Un aspecto clave en esta etapa es la creación de una base de datos destinada a recopilar toda la información relevante de las clasificaciones realizadas para los diferentes servicios. Para esta finalidad, se utiliza *SQLite*, que es un manejador de bases de datos relacionales, debido a su facilidad de uso y configuración. Además, se desarrolló una función adicional para determinar cuál es el servicio más frecuentemente clasificado. Esta información resulta invaluable, ya que se considera para tomar decisiones futuras relacionadas con la gestión del tráfico en la red.



**Figura 20 Ejemplo de clasificación de un servicio Vo5G.**

También se implementó una función que devuelve un archivo JSON con todos los servicios clasificados hasta el momento, incluyendo sus respectivos parámetros de KPI/KQI. Esto proporciona una visión completa y detallada de las clasificaciones realizadas, lo que permite un análisis exhaustivo y facilita la toma de decisiones informadas en el contexto de la gestión de la red.

Para verificar que las solicitudes realizadas a la API llegan al destino, se comprueba mediante confirmaciones de paquetes de tipo 200 OK. En la figura 21 se agregan los comandos que demuestran que la API devuelve códigos de confirmación de tipo 200 OK a las solicitudes HTTP que se hicieron a las direcciones antes mencionadas solicitando la información.

```

127.0.0.1 - - [20/Jun/2023 22:34:32] "POST /clasificar HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 22:44:52] "GET /clasificar HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 22:48:50] "GET /services HTTP/1.1" 200 -

```

**Figura 21 Comandos de confirmación de tipo 200 OK de las solicitudes a la API.**

La figura 22 muestra la respuesta a una solicitud de tipo GET a la dirección `http://127.0.0.1:5000/services`. Este comando hace que la API devuelva un JSON que contenga toda la información referida a los servicios clasificados hasta el momento. Existen más solicitudes a la API las cuales se muestran en el Anexo A.

The screenshot shows a REST client interface with the following details:

- URL: `http://127.0.0.1:5000/services`
- Method: `GET`
- Response Status: `200 OK`
- Response Time: `19 ms`
- Response Size: `7.35 KB`
- Response Body (JSON):
 

```

{
  "idtlog": 1,
  "Latency": 85.0,
  "Jitter": 8.0,
  "Bit_rate": 17.0,
  "PLR": 1e-05,
  "PDL_DL": 15.0,
  "PDR_UL": 10.0,
  "Mobility": 40.0,
  "Reliability": 99.99,
  "Availability": 99.9999,
  "Survival_time": 10.0,
  "EDR_DL": 17.0,
  "EDR_UL": 15.0,
  "Interrupt_time": 10.0,
  "Service": " Smart_Grid"
}

```

Figura 22 Visualización de un método GET para ver los servicios clasificados.

#### 4.3.2 Escenario de topología de red en *Mininet*

En la VM alojada en la plataforma de *Google Cloud*, se procedió a instalar *Mininet* y el controlador ODL de SDN. Una de las principales ventajas de esta configuración es la posibilidad de gestionar la VM desde cualquier estación de trabajo, lo cual brinda flexibilidad y facilidad de acceso. Posteriormente, se cargaron los archivos de la topología en la VM utilizando el protocolo SSH desde la terminal, asegurando así la correcta configuración de la simulación.

El código en *Python* utilizado para crear la topología de red definida (ver Figura 19) está disponible bajo solicitud de quien lo requiera. La elección de la biblioteca *Mininet* se basó en sus capacidades para generar topologías de red y enlaces con características específicas de QoS, aspectos esenciales para los objetivos de esta tesis. En el código se crearon los *hosts*, identificados con la letra "h", correspondientes al UE y a los servicios que serán solicitados (ver Tabla 4).

Con la finalidad de interconectar los enlaces, siguiendo la distribución de QoS detallada en la Tabla 5 se configuraron los *switches* (identificados con la letra "s"). Es importante recordar que todos los valores asignados se basan en los estándares de telecomunicaciones establecidos por la ITU, ETSI y 3GPP. La implementación de esta topología garantiza un entorno de simulación controlado y realista, lo cual es

crucial para la realización exitosa de los experimentos y la evaluación del desempeño del sistema propuesto.

**Tabla 4 Nombre de los hosts**

<b>ID de los <i>hosts</i></b>	<b>Servicio</b>
h1	<i>UHD Video Streaming</i>
h2	<i>Immerse experience</i>
h3	<i>Video surveillance</i>
h4	<i>eHealth</i>
h5	<i>Smart grid</i>
h6	<i>Industry automation</i>
h7	<i>Connected vehicles</i>
h8	ITS
h9	Vo5G
h10	UE

Cabe destacar que la elección de los estándares de telecomunicaciones de la ITU, ETSI y 3GPP asegura que los parámetros de QoS utilizados en la simulación sean representativos de un entorno real de redes de comunicaciones. Esto contribuye a obtener resultados más precisos y confiables al evaluar el desempeño del sistema propuesto y su capacidad para cumplir con los requisitos de calidad y los acuerdos de nivel de servicio (SLA). Asimismo, la configuración de la topología de red de acuerdo con estos estándares permite una comparación más efectiva entre los resultados de la simulación y las expectativas de rendimiento en un entorno de despliegue real.

En la Tabla 6, se detallan las características de QoS asignadas a los enlaces de los *switches* en la topología de red creada. Estas características incluyen parámetros como el ancho de banda, la latencia, el *jitter* y la pérdida de paquetes, los cuales son fundamentales para garantizar un rendimiento óptimo de los servicios y cumplir con los requisitos de calidad. Asimismo, para establecer la conexión entre el controlador SDN ODL y la topología, se utiliza la función *RemoteController*, que permite asignar un controlador de SDN externo al archivo del código.

Se configura la dirección IP del controlador y el puerto de comunicación, siendo en este caso la dirección 127.0.0.1 y el puerto 6633 respectivamente. Esta configuración permite que el controlador pueda supervisar y gestionar de manera centralizada los enlaces y flujos de datos en la topología, facilitando un control dinámico y eficiente de la red.

**Tabla 5 Características de QoS de cada enlace de los switches al servicio o UE de la topología [Elaboración propia].**

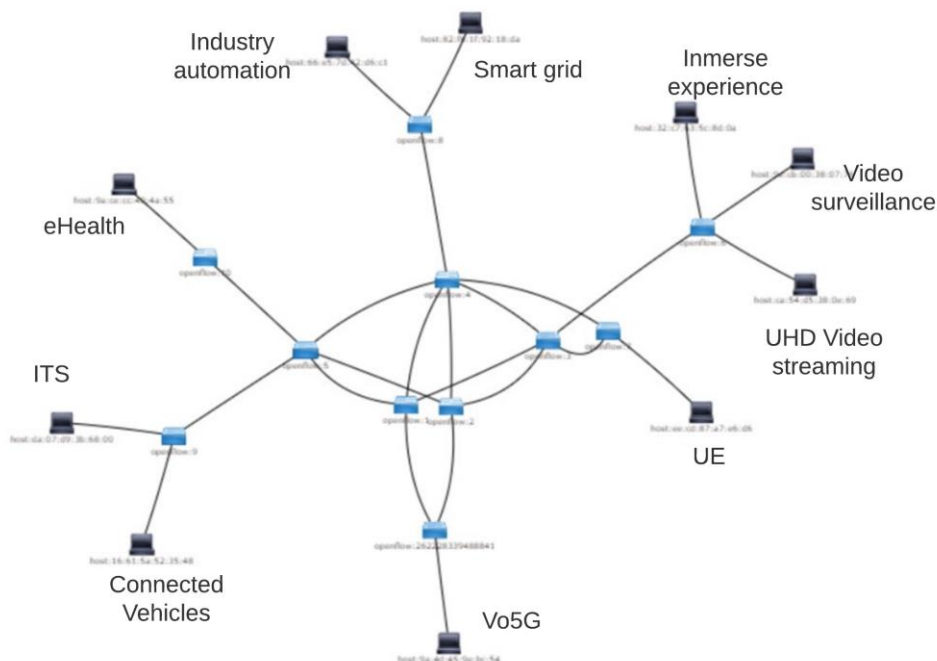
<b>Enlace de conexión (fuente, destino)</b>	<b>BW (Mbps)</b>	<b>Latencia (ms)</b>	<b>Jitter (ms)</b>	<b>Pérdida de paquetes (%)</b>
h1, s6	10	3	6	0.1
h2, s6	50	3	10	0.1
h3, s6	10	3	5	0.1
h4, s7	16	0.5	1	0.00000001
h5, s8	1	0.5	1	0.0001
h6, s8	1	0.4	0.1	0.00000001
h7, s9	10	1	0.5	0.001
h8, s9	0.5	1	2	0.1
h9, s10	10	2	3	0.1
h10, s0	100	0.4	1	0

**Tabla 6 Características de QoS de cada enlace de los switches de la topología [Elaboración propia].**

<b>Enlace de conexión (fuente, destino)</b>	<b>BW (Mbps)</b>	<b>Latencia (ms)</b>	<b>Jitter (ms)</b>	<b>Pérdida de paquetes (%)</b>
s0, s1	100	0.3	2	0
s0, s2	50	0.3	6	0
s1, s3	90	0.5	10	0.1
s1, s4	100	0.5	3	0.01
s1, s5	50	0.4	5	0.1
s2, s3	90	0.4	1	0.1
s2, s4	100	0.5	5	0
s2, s5	50	0.5	2	0
s3, s4	50	1	3	0.1
s4, s5	50	0.6	7	0
s3, s6	50	0.7	6	0.1
s3, s7	16	1	8	0.00000001
s4, s7	16	0.8	9	0.00000001
s4, s8	10	0.6	0.1	0.0001
s5, s9	10	0.9	2	0.1
s5, s10	10	1	1	0.1

Una vez configurados todos los enlaces, la gráfica de la topología en el controlador queda como se muestra:





**Figura 23 Topología de red simulada en *Mininet* [Elaboración propia].**

En la figura 23, se observa que algunos servicios están agrupados en un mismo *switch* de destino. Por ejemplo, los servicios de Redes Inteligentes (*Smart grid*) y Automatización Industrial (*Industry Automation*) comparten similitudes en cuanto a parámetros de rendimiento (KPI) y calidad (KQI). Lo mismo ocurre con los servicios de ITS (Sistemas de Transporte Inteligentes) y Vehículos Conectados (*Connected Vehicles*), como se puede apreciar en la figura 23. Por otro lado, existen servicios que se encuentran aislados de otros en un mismo enlace, como es el caso de Vo5G y *eHealth*. Estas diferencias se deben a las particularidades propias de cada servicio, que los distinguen claramente del resto.

El enfoque de agrupar servicios similares en un mismo *switch* y aislar aquellos que presentan características distintivas tiene como objetivo mejorar la precisión de las simulaciones. De lo contrario, sería necesario diseñar una topología de red enorme para cada servicio individual, lo cual se alejaría de los objetivos principales de esta tesis. La agrupación estratégica de servicios permite reducir la complejidad de la topología y simplificar el proceso de simulación, sin comprometer la precisión en la evaluación del rendimiento del sistema propuesto.

## 4.4 Resultados obtenidos

Para llevar a cabo las simulaciones de las solicitudes de servicio, primeramente, se verifica qué servicio ha sido clasificado por el sistema. A partir de esta información, se realizó un *script* de código que examina todas las posibles rutas desde el UE hasta el *host* del servicio solicitado. Dado que los enlaces cuentan con redundancia, existen múltiples caminos disponibles en caso de que algunos enlaces pierdan conectividad, lo que garantiza así la resiliencia de la red.

Entre todas las rutas posibles, el *script* evalúa la adecuación de cada enlace a los parámetros del *template* de QoS del servicio y las configura a través del controlador SDN. De esta manera, se obtienen las rutas que cumplen con los requisitos establecidos en el *template*. Estas rutas representan los *Network Slices* o NS que pueden ser asignados al UE para satisfacer la solicitud de servicio. En algunos casos, puede haber dos o más rutas o NS que cumplan con estos parámetros, es decir, con el *template* de QoS establecido. Para elegir un camino acorde, el diseño se basa en elegir el NS que tenga el menor retardo de extremo a extremo. Esta decisión obedece a la importancia que tiene la latencia en redes de nueva generación, como se ha explicado en este trabajo de tesis.

En este epígrafe, se describen en detalle las nueve posibles solicitudes de servicios que puede realizar el UE. Además, se presentan tablas que muestran todas las trayectorias disponibles para alcanzar el nodo final, los distintos NS que pueden ser seleccionados y, finalmente, el NS con la menor latencia. Estos datos son fundamentales para comprender la efectividad del sistema propuesto y su capacidad para optimizar la entrega de servicios en función de los requisitos de QoS establecidos, así como para mejorar la QoE del UE y contribuir a la no violación de los SLA.

A continuación, se presenta la Tabla 7, la cual proporciona detalles completos sobre las simulaciones llevadas a cabo en el entorno de *Mininet*. Esta tabla recopila información relevante, como las trayectorias encontradas, las trayectorias que cumplen con los parámetros de QoS establecidos y, por último, destaca la trayectoria (NS) con la menor latencia encontrada. La tabla permite una visión global de los resultados obtenidos, brindando una perspectiva clara de las opciones disponibles para enrutar y asignar recursos en función de los requisitos de QoS, con la consideración de no violar los SLA.

En el Anexo B se incluyen todas las salidas generadas del *script* durante las simulaciones. Estas salidas muestran detalladamente todas las trayectorias identificadas, resaltando aquellas que cumplen con los parámetros de QoS establecidos. Asimismo, se presenta la trayectoria de menor latencia encontrada, que

es un factor determinante en redes de nueva generación, tal como se ha explicado en el desarrollo de esta tesis. Estas salidas proporcionan un análisis detallado y permiten una comprensión más completa de los resultados obtenidos en las simulaciones realizadas en el entorno de *Mininet*.

**Tabla 7 Resultados de las simulaciones en el entorno de *Mininet* [Elaboración propia].**

Solicitud de servicio	Cantidad de trayectorias encontradas	Template de QoS del servicio (BW, latencia, jitter y pérdida de paquetes)	NS que cumplen con el template de QoS	Salto en el NS de menor latencia
UE – UHD video <i>streaming</i> , videovigilancia, experiencias inmersivas	21	[50, 15, 1, 30]	14	['1', '3', '4', '7'] con 1.4 ms.
UE – ITS, vehículos conectados	21	[10, 5, 1, 10]	2	['1', '2', '6', '10'] con 1.6 ms.
UE - <i>eHealth</i>	29	[10, 3, 0.1, 15]	1	['1', '2', '5', '8'] con 1.6 ms.
UE - Automatización industrial, redes inteligentes	17	[10, 10, 1, 20]	10	['1', '2', '5', '9'] con 1.4 ms.
UE – Vo5G	21	[10, 30, 1, 30]	18	['1', '2', '6', '11'] con 1.7 ms.

realizar las simulaciones y debido a la redundancia de la red, la Tabla 7 enfatiza que se encontraron más de 15 trayectorias posibles para cada solicitud de servicio. Sin embargo, es importante destacar que, en general, al menos un trayecto cumple con los parámetros de QoS definidos en los estándares. Este paso es crucial, ya que identifica claramente qué trayectorias cumplen con los requisitos establecidos y cuáles requieren ajustes por parte del proveedor de servicios, como, por ejemplo, la modificación del ancho de banda del enlace para garantizar el cumplimiento de los SLA.

En la última columna de la Tabla 7, se muestra la trayectoria que cumple con el *template* de QoS y, además, es el que presenta la menor latencia. Teniendo en cuenta que los estándares de comunicación de redes de nueva generación, como 5G/B5G, establecen que las latencias E2E deben rondar alrededor de 1 ms. Es notable que, aunque todas las solicitudes de servicio cuentan con trayectorias de 4 saltos, la latencia E2E no supera los 2 ms; por lo que estos resultados se consideran satisfactorios.

Los hallazgos obtenidos demuestran que el sistema propuesto clasifica el tráfico de acuerdo con el tipo de aplicación y genera su SLA, además, es capaz de encontrar trayectorias factibles y cumplir con los objetivos de baja latencia E2E establecidos por los estándares. Esto es fundamental para ofrecer una experiencia de usuario mejorada en servicios de próxima generación, así como, mejorar la QoS ofrecida por los proveedores de servicios y la no violación de los SLA establecidos entre cliente-proveedor.

## 4.5 Conclusiones

En este capítulo, se presentó y validó la propuesta de un sistema de clasificación de servicios y asignación de recursos en redes de nueva generación, basado en la utilización de una API, un controlador SDN y la simulación de topologías de red. A lo largo de este capítulo, ha permitido desarrollar e implementar una solución integral para la clasificación y asignación de servicios en redes 5G/B5G, demostrando su viabilidad, efectividad en la optimización de recursos de red y cumplimiento de los requisitos de calidad de servicio. Se ha realizado un análisis exhaustivo de los resultados obtenidos a través de las simulaciones, y se han extraído conclusiones relevantes que contribuyen a una mejor comprensión y apreciación de la eficacia y eficiencia del sistema propuesto. Se puede concluir que:

- El sistema propuesto de clasificación de servicios y asignación de recursos, basado en una API y un controlador SDN, ha demostrado ser efectivo para garantizar el cumplimiento de los requisitos de QoS establecidos por los servicios y aplicaciones en las redes de nueva generación.
- La integración del clasificador en una API REST utilizando el *framework Flask* ha permitido una comunicación eficiente entre el UE y el controlador SDN, facilitando la solicitud de servicios y la asignación de recursos de red de manera automatizada.
- La validación de las trayectorias de red para cada solicitud de servicio ha demostrado que existen múltiples rutas posibles debido a la redundancia de la red. Sin embargo, se ha identificado al menos una trayectoria que cumple con los parámetros de QoS establecidos, lo cual garantiza el cumplimiento de los SLA y la entrega de servicios de calidad.
- La selección de la trayectoria de menor latencia como criterio principal ha demostrado ser efectiva para garantizar una baja latencia en las comunicaciones de los servicios. Esto es especialmente relevante en las redes de nueva generación, donde la latencia juega un papel crucial.

- Los resultados obtenidos en las simulaciones muestran que las trayectorias encontradas tienen un número de saltos adecuado y una latencia dentro de los límites establecidos por los estándares de comunicación de redes de nueva generación. Estos resultados respaldan la eficiencia y rendimiento del sistema propuesto.
  
- La capacidad de adaptación del sistema ante cambios en la demanda de servicios se ha comprobado, ya que el controlador SDN permite realizar ajustes dinámicos en tiempo real en la asignación de recursos, evitando las congestiones de red.

## Capítulo 5. Conclusiones y recomendaciones

---

Este trabajo de tesis, se retoman los principales aspectos teóricos acerca de redes de nueva generación, SDN, VNF, NS y los SLA que sustentan las bases para atender los objetivos planteados. De igual forma, se abordaron temas referidos a la gestión de redes 5G/B5G y la aplicación de herramientas de ML en esta rama. Asimismo, en esta tesis se enfrentó el desafío de la gestión eficiente de servicios en redes de nueva generación, centrándose específicamente en la clasificación y asignación de servicios en entornos 5G/B5G. Se demostró la importancia de contar con soluciones automatizadas y flexibles para satisfacer las demandas de los usuarios finales y garantizar el cumplimiento de los acuerdos de nivel de servicio (SLA).

Como resultado de la tesis, se propuso un sistema que integra un clasificador de servicios 5G/B5G en un escenario virtual de una topología en *Mininet*, con el objetivo de asignar los NS que cumplan con los SLA del proveedor de servicios. Se demostró que permite una asignación más precisa de los recursos de red y una mejor optimización de la QoS. Además, se utilizó una integración de la tecnología de redes definidas por software (SDN), lo que facilita la gestión dinámica de la red. Esta integración permite una asignación eficiente de recursos y una rápida adaptación a los cambios en las demandas de los servicios. Con relación a la implementación del clasificador de servicios en la API REST, el análisis de los resultados obtenidos arroja las siguientes conclusiones:

El clasificador de servicios implementado demostró ser preciso en la identificación de servicios en entornos 5G/B5G. Mediante el uso de técnicas de ML, el clasificador de Bosque Aleatorio logra una alta tasa de acierto, del 97% aproximadamente; lo que permite una asignación más precisa de recursos y una mejor optimización de la QoS.

La integración del clasificador de servicios en una API REST facilita su uso y accesibilidad. Al proporcionar una interfaz de programación simple y estandarizada, la API permite a los usuarios enviar solicitudes de clasificación de servicios de manera conveniente y obtener respuestas rápidas y precisas. Esto mejora la eficiencia en la gestión de servicios y facilita la integración del clasificador en entornos de redes existentes.

El clasificador de servicios demostró ser escalable y adaptable a diferentes tipos de servicios y requisitos de calidad de servicio (QoS). Mediante la configuración de *templates* de QoS para cada servicio, el clasificador se adapta a las necesidades específicas de cada servicio y garantiza el cumplimiento de los acuerdos de nivel de servicio (SLA). Además, su arquitectura modular permite agregar y modificar fácilmente nuevos servicios y ajustar los parámetros de clasificación según sea necesario.

Con respecto a la utilización del controlador SDN ODL para controlar el tráfico en la topología creada en *Mininet*, luego de las distintas simulaciones, se puede concluir que:

La integración de la tecnología de redes definidas por software (SDN) facilitó la gestión centralizada y dinámica de la red, permite una asignación eficiente de recursos y una rápida adaptación a los cambios en las demandas de los servicios. Mediante el uso del controlador SDN, se habilita una administración simplificada de la topología y la capacidad de ajustar dinámicamente los parámetros de QoS para adaptarse a las necesidades cambiantes de los servicios.

La simulación de la topología de red en *Mininet* brindó un entorno de prueba controlado y realista para evaluar el rendimiento del sistema propuesto. Los resultados obtenidos demostraron la viabilidad y efectividad de la solución implementada en términos de cumplimiento de los SLA y entrega de servicios de calidad.

La selección de la trayectoria de menor latencia como criterio principal para la asignación de servicios demostró ser efectiva para garantizar una baja latencia en las comunicaciones. Esto es especialmente relevante en las aplicaciones de redes de nueva generación que requieren tiempos de respuesta rápidos y confiables.

## **5.1 Contribuciones al conocimiento**

La contribución al conocimiento de esta tesis radica en la implementación exitosa de un clasificador de servicios basado en algoritmos de ML y, su integración con un controlador SDN en un entorno de simulación utilizando *Mininet*. Esta integración permite una gestión eficiente y flexible de la red, así como la optimización de la asignación de recursos y la mejora de la calidad de servicio para los servicios de nueva generación. Además, se obtuvieron resultados prometedores en términos de precisión de clasificación y selección de rutas óptimas basadas en parámetros de QoS. Esta investigación aporta conocimientos valiosos y aplicables para el diseño y la gestión de redes de comunicación avanzadas, contribuyendo al avance de la tecnología y la mejora de la experiencia del usuario en entornos de comunicación modernos. Los aportes de este trabajo son:

La implementación de un clasificador de servicios basado en algoritmos de ML, capaz de identificar y clasificar diferentes tipos de servicios en redes de nueva generación. Este clasificador mejora la eficiencia en la gestión de recursos, la QoS para los usuarios finales, así como la no violación de los SLA.

La integración exitosa del clasificador de servicios con un controlador SDN, lo que permite una gestión más flexible y eficiente de la red. Esta integración facilita la asignación óptima de recursos y la adaptación dinámica de la red a las necesidades cambiantes de los servicios.

La implementación de un entorno de simulación utilizando *Mininet*, que permite realizar experimentos y evaluar el rendimiento del sistema propuesto de manera controlada y realista. Esta simulación proporciona información valiosa sobre la eficacia del enfoque propuesto y ayuda a identificar áreas de mejora y optimización.

La obtención de resultados prometedores en términos de precisión de clasificación y selección de rutas óptimas basadas en parámetros de QoS. Estos resultados muestran la viabilidad y efectividad del enfoque propuesto en la mejora de la experiencia del usuario y la optimización de los recursos de red, con el objetivo de cumplir con los SLA establecidos.

La contribución al conocimiento en el campo de las redes de comunicación avanzadas, proporcionando ideas y enfoques novedosos para el diseño y la gestión de redes de nueva generación. Estos aportes ayudan a avanzar en el desarrollo de tecnologías de comunicación más eficientes y mejor adaptadas a las necesidades actuales y futuras de los usuarios.

## **5.2 Limitaciones de la investigación**

En esta sección se describen algunas limitaciones de la investigación desarrollada en este trabajo:

Generalización de los resultados: Los resultados obtenidos en esta investigación se basan en un escenario de simulación específico y en un conjunto particular de servicios y parámetros de QoS. Es importante reconocer que estos resultados pueden variar en diferentes entornos y con diversas configuraciones de red. Se requiere un análisis adicional y una validación experimental en entornos reales para evaluar la aplicabilidad y eficacia del enfoque propuesto en situaciones más dispares.



Complejidad de la implementación: Si bien se demostró la viabilidad de la implementación del clasificador de servicios y su integración con un controlador SDN en el entorno de simulación, la implementación y despliegue en un entorno de producción real pueden presentar desafíos adicionales. La escalabilidad, la interoperabilidad con otros sistemas y la gestión de la infraestructura física pueden requerir un análisis y una planificación más detallados.

Dependencia de la calidad de los datos de entrada: El rendimiento del clasificador de servicios y la toma de decisiones basada en los parámetros de calidad de servicio dependen en gran medida de la precisión y confiabilidad de los datos de entrada. Cualquier inexactitud o variabilidad en la medición de los parámetros de QoS puede afectar la precisión de las clasificaciones y las decisiones tomadas. Se requiere una gestión cuidadosa de los datos de entrada y posibles técnicas de filtrado o corrección de datos para mejorar la confiabilidad de los resultados.

Evolución tecnológica: La investigación se basa en los estándares y tecnologías disponibles hasta la fecha de corte del conocimiento. Sin embargo, el campo de las redes de nueva generación está en constante evolución y pueden surgir nuevos estándares, protocolos o tecnologías que podrían tener un impacto significativo en la efectividad y relevancia del enfoque propuesto. Se requiere un seguimiento continuo de los avances en el campo y una adaptación del sistema propuesto para mantener su relevancia y competitividad en un entorno en constante cambio.

### **5.3 Trabajo futuro**

Luego de analizar los resultados obtenidos en las simulaciones, así como las limitaciones que presenta esta investigación, se propone lo siguiente como trabajo futuro:

Mejorar la precisión del clasificador: Esto puede incluir el uso de algoritmos de ML más sofisticados, la incorporación de características adicionales de los servicios y la exploración de técnicas de procesamiento de señales para un análisis más detallado de los flujos de tráfico.

Evaluar la solución planteada en un entorno de producción real: Se pueden realizar pruebas piloto en colaboración con proveedores de servicios de telecomunicaciones u otros actores relevantes en el campo.

**Agregar nuevos servicios y estándares:** El enfoque propuesto se centra en un conjunto específico de servicios y parámetros de calidad de servicio. Para ampliar su aplicabilidad, se puede investigar la incorporación de nuevos servicios y la adaptación a diferentes estándares y protocolos de comunicación. Esto permitirá su uso en un espectro más amplio de aplicaciones y contextos.

**Incorporar aspectos de seguridad y privacidad:** Dado que las redes definidas por software y los controladores SDN introducen nuevos desafíos en términos de seguridad y privacidad de la información, se recomienda realizar investigaciones adicionales en este ámbito. Esto puede incluir el desarrollo de mecanismos de autenticación y protección de datos, así como el análisis de posibles vulnerabilidades y contramedidas para garantizar la seguridad de la red y la privacidad de los usuarios.

**Monitorizar los SLA:** Una vez comprobado que se cumplen los SLA, se requiere una monitorización de los enlaces, con el objetivo de cuidar una posible sobreasignación o sub-asignación de recursos de red.

## Literatura citada

---

- 3GPP. (2019). *3GPP TS 23.501 System Architecture for the 5G System: Vol. 16.0.2* (Número Release 16, p. 308). <http://www.3gpp.org>
- 5gAmericas. (2016). Network Slicing for 5G Networks & Services. En *5G Americas White Paper* (Vol. 1, p. 34). [http://www.5gamericas.org/files/3214/7975/0104/5G\\_Americas\\_Network\\_Slicing\\_11.21\\_Final.pdf](http://www.5gamericas.org/files/3214/7975/0104/5G_Americas_Network_Slicing_11.21_Final.pdf)
- Acosta Rams, J., Anías Calderón, C., & Frómeta Fonseca, D. (2017). Lenguajes de alto nivel de abstracción para el desarrollo de aplicaciones SDN. *Telemática*, 16(2), 1–11. <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/258>
- Amazon Web Services (AWS). (2023). *¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS*. <https://aws.amazon.com/es/what-is/api/>
- Asiri, S. (2020). *Machine Learning Classifiers, What is classification?* <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623>
- AWS. (2023). *Amazon Web Services (AWS)*. <https://aws.amazon.com/es/>
- Bagnato, J. I. (2017). *Random Forest, el poder del Ensemble, Aprende Machine Learning*. <https://www.aprendemachinelarning.com/random-forest-el-poder-del-ensemble/>
- Ballester Macías, S. (2014). *Gestión de las SDN*. Universidad Tecnológica de La Habana (CUJAE).
- Baniş, O., Florea, D., Gyalai, R., & Curiac, D.-I. (2021). Automated Specification-Based Testing of REST APIs. *Sensors*, 21(16). <https://doi.org/10.3390/s21165375>
- Barona López, L. I., Maestre Vidal, J., & García Villalba, L. J. (2017). An approach to data analysis in 5G networks. *Entropy*, 19(2), 74. <https://doi.org/10.3390/e19020074>
- Borcoci, E. (2019). Machine Learning Techniques in Advanced Network and Services Management Network and Services Management. *SoftNet 2019*, 102.
- Chen, S., Lee, C.-N., & Lee, M.-F. (2020). Realization of 5G Network Slicing Using Open Source Softwares. *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1549–1556. <https://ieeexplore.ieee.org/document/9306404>
- clarcat. (2023). *Comparativa: AWS vs. Microsoft Azure vs. Google Cloud Platform*. <https://www.clarcat.com/comparativa-aws-vs-microsoft-azure-vs-google-cloud-platform/>
- Huawei Technologies. (2018). *Vo5G Technical White Paper*. <https://www-file.huawei.com/-/media/corporate/pdf/white%20paper/2018/vo5g-technical-white-paper-en-v2.pdf?la=en>
- Demircioğlu, E. D., & Kalipsiz, O. (2022). API Message-Driven Regression Testing Framework. *Electronics*, 11(17). <https://doi.org/10.3390/electronics11172671>

- do Vale Saraiva, T., Campos, C. A. V., Fontes, R. dos R., Rothenberg, C. E., Sorour, S., & Valaee, S. (2021). An Application-Driven Framework for Intelligent Transportation Systems Using 5G Network Slicing. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 5247–5260. <https://doi.org/10.1109/TITS.2021.3086064>
- EstiNet. (2023). *EstiNet Simulator*. [https://www.gordonsmart.com/ns/?page\\_id=24751](https://www.gordonsmart.com/ns/?page_id=24751)
- ETSI, 3GPP. (2018). *TS 22.261 5G; Service requirements for next generation new services and markets (Release 15)*. *TS 22.261(V15.5.0)*, 0–52. <http://www.etsi.org/standards-search>
- Flask. (2023). *Flask Documentation*. <https://flask.palletsprojects.com/en/2.3.x/>
- FloodLight. (2023). *FloodLight Controller*. <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview>
- Forum, T., & Group, O. (2004). SLA Management Handbook: Enterprise Perspective. En *TM Forum, October* (Vol. 4, Número October). TMForum. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:SLA+Management+Handbook+-+Volume+4:+Enterprise+Perspective#0>
- GNS3. (2023). *GNS3 Home Page*. <https://www.gns3.com/>
- Google. (2023). *Google Cloud*. [https://cloud.google.com/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=latam-MX-all-es-dr-BKWS-all-all-trial-e-dr-1605194-LUAC0010195&utm\\_content=text-ad-none-any-DEV\\_c-CRE\\_512379899411-ADGP\\_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Ttxt%20~%20GCP\\_General-KWID\\_43700062784667416-kwd-301173107424&utm\\_term=KW\\_google%20cloud-ST\\_Google%20Cloud&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu6vA4Yi8d84hmDmUp3BYNc3U92XU1-w846TwETcY3gp0bE0OcBTmgRoC90kQAvD\\_BwE&gclidsrc=aw.ds&hl=es](https://cloud.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-MX-all-es-dr-BKWS-all-all-trial-e-dr-1605194-LUAC0010195&utm_content=text-ad-none-any-DEV_c-CRE_512379899411-ADGP_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Ttxt%20~%20GCP_General-KWID_43700062784667416-kwd-301173107424&utm_term=KW_google%20cloud-ST_Google%20Cloud&gclid=CjwKCAjwzJmlBhBBEiwAEJyLu6vA4Yi8d84hmDmUp3BYNc3U92XU1-w846TwETcY3gp0bE0OcBTmgRoC90kQAvD_BwE&gclidsrc=aw.ds&hl=es)
- Group, I.-2020 (5G) P. (2015). *5G Vision and Requirement*. <https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf>
- Guan, W., Wen, X., Wang, L., Lu, Z., & Shen, Y. (2018). A Service-Oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory. *IEEE Access*, 6, 19691–19701. <https://doi.org/10.1109/ACCESS.2018.2822398>
- Ibarra-Lancheros, K., Puerto-Leguizamón, G., & Suárez, C. (2018). Quality of service evaluation based on network slicing for software-defined 5G systems. *Tecnológicas*, 21, 27–41. <https://doi.org/10.22430/22565337.1066>
- Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K. C., & Hanzo, L. (2016). Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24(2), 98–105. <https://doi.org/10.1109/MWC.2016.1500356WC>
- Jiang, M., Condoluci, M., & Mahmoodi, T. (2017). Network slicing in 5G: An auction-based model. *IEEE International Conference on Communications*. <https://doi.org/10.1109/ICC.2017.7996490>

- Juniper Network. (2023). *¿Qué es la virtualización de funciones de red (NFV)? | Juniper Networks*. <https://www.juniper.net/mx/es/research-topics/what-is-network-functions-virtualization-nfv.html>
- Kapassa, E., Touloupou, M., & Kyriazis, Di. (2018). SLAs in 5G: A complete framework facilitating VNF- and NS-tailored SLAs management. *Proceedings - 32nd IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2018, 2018-Janua*, 469–474. <https://doi.org/10.1109/WAINA.2018.00130>
- Kapassa, E., Touloupou, M., Stavrianos, P., Xylouris, G., Kyriazis, D., Kapassa, E., Touloupou, M., Stavrianos, P., Xylouris, G., & Kyriazis, D. (2019). Managing and optimizing quality of service in 5G environments across the complete SLA lifecycle. *Advances in Science, Technology and Engineering Systems*, 4(1), 329–342. <https://doi.org/10.25046/aj040132>
- Klaine, P. V, Imran, M. A., Onireti, O., & Souza, R. D. (2017). A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks. *IEEE Communications Surveys Tutorials*, 19(4), 2392–2431. <https://doi.org/10.1109/COMST.2017.2727878>
- Li, L., Chou, W., Zhou, W., & Luo, M. (2016). Design Patterns and Extensibility of REST API for Networking Applications. *IEEE Transactions on Network and Service Management*, 13(1), 154–167. <https://doi.org/10.1109/TNSM.2016.2516946>
- Li, X., Samaka, M., Chan, H. A., Bhamare, D., Gupta, L., Guo, C., & Jain, R. (2017). Network Slicing for 5G: Challenges and Opportunities. *IEEE Internet Computing*, 21(5), 20–27. <https://doi.org/10.1109/MIC.2017.3481355>
- Liyanapathirana, L. (2018). *Classification Model Evaluation*. <https://heartbeat.fritz.ai/classification-model-evaluation-90d743883106>
- López Briega, R. (2016). *Machine Learning con Python - Sobreajuste*. <https://relopezbriega.github.io/blog/2016/05/29/machine-learning-con-python-sobreajuste/>
- Lozano-Rizk, J. E., Gonzalez-Trejo, J. E., Rivera-Rodriguez, R., Tchernykh, A., Villarreal-Reyes, S., & Galaviz-Mosqueda, A. (2022). Application-Aware Flow Forwarding Service for SDN-Based Data Centers. *Electronics*, 11(23). <https://doi.org/10.3390/electronics11233882>
- Microsoft. (2023). *Microsoft AZURE*. [https://azure.microsoft.com/es-mx/free/search/?ef\\_id=\\_k\\_CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM\\_Np55RS5dSn6ulCdM3qxoCRxAQAvD\\_BwE\\_k\\_&OCID=AIDcmmxotgtm93\\_SEM\\_\\_k\\_CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM\\_Np55RS5dSn6ulCdM3qxoCRxAQAvD\\_BwE\\_k\\_&gclid=CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM\\_Np55RS5dSn6ulCdM3qxoCRxAQAvD\\_BwE](https://azure.microsoft.com/es-mx/free/search/?ef_id=_k_CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM_Np55RS5dSn6ulCdM3qxoCRxAQAvD_BwE_k_&OCID=AIDcmmxotgtm93_SEM__k_CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM_Np55RS5dSn6ulCdM3qxoCRxAQAvD_BwE_k_&gclid=CjwKCAjwzJmIBhBBEiwAEJyLu5vqEYamoK63UcR8GZI6209aCTrvYfa6RM_Np55RS5dSn6ulCdM3qxoCRxAQAvD_BwE)
- Miguel, L. H. (2018). Impacto del Machine Learning en el despliegue de redes 5G. En U. T. Nacional (Ed.), *10th ITU Academic Conference ITU KALEIDOSCOPE* (pp. 1–15). <https://kaleidoscope.frsf.utn.edu.ar/utn-frsf>
- Millán Tejedor, R. J. (2014). SDN: el futuro de las redes inteligentes. *GM2 Publicaciones Técnicas, S.L*, 179. [www.ramonmillan.com](http://www.ramonmillan.com)

- Mininet. (2023). *Mininet Overview*. <http://mininet.org/overview/>
- Morocho-Cayamcela, M. E., Lee, H., & Lim, W. (2019). Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. *IEEE Access*, 7, 137184–137206. <https://doi.org/10.1109/ACCESS.2019.2942390>
- Moysen, J., & Giupponi, L. (2017). *From 4G to 5G: Self-organized Network Management meets Machine Learning*. 1–23. <http://arxiv.org/abs/1707.09300>
- Mullins, R., & Barros, M. (2017). *Cognitive Network Management for 5G The path towards the development and deployment of cognitive networking Whitepaper-draft for consultation By 5GPPP Working Group on Network Management and QoS*. [https://5g-ppp.eu/wp-content/uploads/2016/11/NetworkManagement\\_WhitePaper\\_1.0.pdf](https://5g-ppp.eu/wp-content/uploads/2016/11/NetworkManagement_WhitePaper_1.0.pdf)
- Neumann, A., Laranjeiro, N., & Bernardino, J. (2021). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, 14(4), 957–970. <https://doi.org/10.1109/TSC.2018.2847344>
- NS-3. (2023). *NS-3 Home Page*. <https://www.nsnam.org/>
- ONOS. (2023). *Wiki*. <https://wiki.onosproject.org/>
- OpenDayLight. (2023). *OpenDayLight Home Page*. <https://www.opendaylight.org/>
- OpenNet. (2016). *OpenNet Tutorial*. <https://github.com/dlinknctu/OpenNet/blob/master/doc/TUTORIAL.md>
- Pahl, C., & Jamshidi, P. (2016). Microservices: A systematic mapping study. *CLOSER 2016 - Proceedings of the 6th International Conference on Cloud Computing and Services Science*, 1, 137–146. <https://doi.org/10.5220/0005785501370146>
- Papageorgiou, A., Fernandez-Fernandez, A., Ochoa-Aday, L., Pelaez, M. S., & Siddiqui, M. S. (2020). *SLA Management Procedures in 5G Slicing-based Systems*. 7–11. <https://doi.org/10.1109/eucnc48522.2020.9200904>
- Pautasso, C., Zimmermann, O., & Leymann, F. (2008). Restful Web Services vs. “Big” Web Services: Making the Right Architectural Decision. *Proceedings of the 17th International Conference on World Wide Web*, 805–814. <https://doi.org/10.1145/1367497.1367606>
- Pérez Hernández, R. (2021). *Monitoring and Orchestration of Network Slices for 5G Networks*. <https://e-archivo.uc3m.es/handle/10016/32956>
- Popovski, P., Trillingsgaard, K. F., Simeone, O., & Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. *IEEE Access*, 6, 55765–55779. <https://doi.org/10.1109/ACCESS.2018.2872781>
- Postman. (2023). *POSTMAN Documentation*. <https://www.postman.com/>
- Preciado-Velasco, J. E. (2021). *SISTEMA CLASIFICADOR DE SERVICIOS 5G/B5G SOPORTADO POR TÉCNICAS DE MACHINE LEARNING* [Universidad Autónoma de Baja California, Facultad de Ingeniería, Arquitectura y Diseño]. <https://repositorioinstitucional.uabc.mx/handle/20.500.12930/9108>

- Preciado-Velasco, J. E., Gonzalez-Franco, J. D., Anias-Calderon, C. E., Nieto-Hipolito, J. I., & Rivera-Rodriguez, R. (2021). 5G/B5G Service Classification Using Supervised Learning. *Applied Sciences*, 11(11), 4942. <https://doi.org/10.3390/app11114942>
- Ramos García, R. (2021). *Predicción del comportamiento de SLA en redes 5G utilizando inteligencia artificial* [Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE)]. <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/3656>
- RedHat. (2023a). *¿Qué es la NFV?* <https://www.redhat.com/es/topics/virtualization/what-is-nfv>
- RedHat. (2023b). *¿Qué es una API y cómo funciona?* <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Ruiz, J. Z. (2018). *Comparativa y Análisis De Algoritmos de Aprendizaje Automático para la Predicción del Tipo Predominante de Cubierta Arbórea* (p. 133).
- RYU. (2023). *RYU Controller*. <https://ryu-sdn.org/>
- Sama, M. R., Beker, S., Kiess, W., & Thakolsri, S. (2016). Service-based slice selection function for 5G. *2016 IEEE Global Communications Conference (GLOBECOM)*. <https://doi.org/10.1109/GLOCOM.2016.7842265>
- Schmelz, L. C., & Nok, C. M. (2017). *5G Mobile Network Architecture for diverse services, use cases, and applications in 5G and beyond* (Número 761445). [https://5g-monarch.eu/wp-content/uploads/2017/10/5G-MoNArch\\_761445\\_D6.1\\_Documentation\\_of\\_Requirements\\_and\\_KPIs\\_and\\_Definition\\_of\\_Suitable\\_Evaluation\\_Criteria\\_v1.0.pdf](https://5g-monarch.eu/wp-content/uploads/2017/10/5G-MoNArch_761445_D6.1_Documentation_of_Requirements_and_KPIs_and_Definition_of_Suitable_Evaluation_Criteria_v1.0.pdf)
- Scikit-learn developers. (2023). *Neural network models (supervised)*. Scikit-learn 1.3 documentation. [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)
- Shlens, J. (2014). A Tutorial on Principal Component Analysis. *Educational*, 51.
- Telecommunication, B., & Maarten, St. (2015). *Quality parameters and standards for Telecommunications and Broadcasting networks ( Study of Specifications and Recommendations ) Executive summary* (p. 16). [https://www.sxmregulator.sx/dash/files/Telecommunications/Policies/10t98467537144\\_\\_UW9TIFN0dWR5IDlwMTU=b\\_64.pdf](https://www.sxmregulator.sx/dash/files/Telecommunications/Policies/10t98467537144__UW9TIFN0dWR5IDlwMTU=b_64.pdf)
- Unión Internacional de Telecomunicaciones ITU. (2018). *Sentando las bases para la 5G: Oportunidades y desafíos*. [https://www.itu.int/dms\\_pub/itu-d/opb/pref/D-PREF-BB.5G\\_01-2018-PDF-S.pdf](https://www.itu.int/dms_pub/itu-d/opb/pref/D-PREF-BB.5G_01-2018-PDF-S.pdf)
- Van Rossem, S., Cai, X., Cerrato, I., Danielsson, P., Nemeth, F., Pechenot, B., Pelle, I., Risso, F., Sharma, S., Skoldstrom, P., & John, W. (2017). NFV service dynamicity with a DevOps approach: Insights from a use-case realization. *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 674–679. <https://doi.org/10.23919/INM.2017.7987357>
- Vision, I. M. T., & M.2083-0, I.-R. (2015). *IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond: Vol. M Series* (p. 21). [https://www.itu.int/dms\\_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-!!!PDF-E.pdf](https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-!!!PDF-E.pdf)

- Wang, G., Feng, G., Tan, W., Qin, S., Wen, R., & Sun, S. (2017, mayo). *Resource Allocation for Network Slices in 5G with Network Resource Pricing*. <https://doi.org/10.1109/GLOCOM.2017.8254074>
- Wei-Lun, C. (2011). Machine learning tutorial. En *Aec-Apc* (p. 56). National Taiwan University. <http://disp.ee.ntu.edu.tw/~pujols/Machine>
- Wu, W., Zhou, C., Li, M., Wu, H., Zhou, H., Zhang, N., Shen, X. S., & Zhuang, W. (2022). AI-Native Network Slicing for 6G Networks. *IEEE Wireless Communications*, 29(1), 96–103. <https://doi.org/10.1109/MWC.001.2100338>
- Wu, Y.-J., Hwang, W.-S., Shen, C.-Y., & Chen, Y.-Y. (2022). Network Slicing for mMTC and URLLC Using Software-Defined Networking with P4 Switches. *Electronics*, 11(14). <https://doi.org/10.3390/electronics11142111>
- Xia, Z., Chen, Y., & Xu, C. (2022). Multiview PCA: A Methodology of Feature Extraction and Dimension Reduction for High-Order Data. *IEEE Transactions on Cybernetics*, 52(10), 11068–11080. <https://doi.org/10.1109/TCYB.2021.3106485>
- Yousaf, Z. (2016). *Deliverable D5.1 Definition of connectivity and QoE / QoS management mechanisms – intermediate report* (pp. 1–189). [http://www.it.uc3m.es/wnl/5gnorma/pdf/5g\\_norma\\_d5-1.pdf](http://www.it.uc3m.es/wnl/5gnorma/pdf/5g_norma_d5-1.pdf)



## Anexos

---

### Anexo A: Algunas solicitudes a la API REST del clasificador de servicios

En este anexo se muestra ejemplos concretos de solicitudes realizadas al clasificador de servicios 5G/B5G que se implementó en la API REST. Además, se presentan diversas capturas de pantalla del software *Postman* que ilustran el proceso de envío de solicitudes y los resultados obtenidos. Estas imágenes permiten visualizar de manera práctica cómo interactuar con la API REST y cómo se clasifican los servicios según los parámetros de QoS correspondientes. A través de estas fotografías, se brinda una representación visual que complementa la descripción detallada de cada solicitud y su respuesta, como se explicó en este trabajo de tesis. Además, se muestra una tabla descriptiva a modo de documentación de los *endpoints* más importantes de la API REST.

**Tabla 8 Especificaciones de cada *endpoint* de la API REST [Elaboración propia].**

<i>Endpoints and Methods</i>	Entrada	Salida	Acción
POST http://127.0.0.1:5000/clasificar	Requerida: idapp. La solicitud de entrada se describe en la Lista 1.	200 OK. La salida se muestra en la Lista 2.	Crear
GET http://127.0.0.1:5000/clasificar	-	200 OK. La salida se muestra en la Lista 3 y en la Figura 25.	Leer
GET http://127.0.0.1:5000/services	-	200 OK. La salida se muestra en la Figura 22.	Leer
GET http://127.0.0.1:5000/estadisticas	-	200 OK. La salida se muestra en la Figura 26.	Leer
DELETE http://127.0.0.1:5000/services	-	Borra registros duplicados de servicios.	Borrar

**Lista 1.** Solicitud *<body>* del objeto JSON para el *endpoint* clasificar.

```
{
  "Latency": 0.5,
  "Jitter": 1,
  "Bit_rate": 5,
  "PLR": 0.0000001,
  "PDR_DL": 20,
```

```

"PDR_UL": 20,
"Mobility": 0,
"Reliability": 99.999,
"Availability": 99.99999,
"Survival_time": 8,
"EDR_DL": 83,
"EDR_UL": 10,
"Interrupt_time": 2
}

```

**Lista 2.** Objeto JSON de respuesta al método POST en el *endpoint* clasificar.

```

{
  "Service": "Industry automation",
  "Latency": 1,
  "Jitter": 0.1,
  "Bandwidth": 1,
  "PLR": 1e-07,
  "Use_Case": "mMTC"
}

```

**Lista 3.** Objeto JSON de respuesta al método GET en el *endpoint* clasificar.

```

{
  "Service": " Industry_Automation"
}

```

The screenshot shows a REST client interface with a list of requests on the left and a detailed view of a GET request on the right. The request is to `http://127.0.0.1:5000/clasificar`. The response body is displayed in JSON format:

```

{
  "Service": " Industry_Automation"
}

```

The interface also shows a table for Query Params and a status bar indicating a 200 OK response with 11 ms latency and 207 B body size.

**Figura 24** Respuesta de la API REST que muestra el servicio recién clasificado.

The screenshot displays a REST client interface with the following components:

- Request:** Method: GET, URL: http://127.0.0.1:5000/estadisticas
- Response:** Status: 200 OK, Time: 18 ms, Size: 212 B
- Response Body (JSON):**

```
1 {
2   "Most_frequent_service": " Smart_Grid"
3 }
```

The interface also shows a list of requests on the left and various tabs for request details (Params, Headers, Body, etc.) and response details (Body, Cookies, Headers, Test Results).

**Figura 25** Respuesta de la API REST que muestra el servicio que más veces se ha clasificado.

## Anexo B: Trayectorias encontradas para cada solicitud de servicio

En este anexo se presenta un registro detallado de las diferentes trayectorias identificadas durante las simulaciones realizadas en el entorno de *Mininet*. Además, se muestran las rutas encontradas para cada solicitud de servicio específica, considerando la topología de red definida previamente (ver Figura 24). Las trayectorias se presentan en forma de tablas, donde se especifican los nodos y enlaces que componen cada ruta y se señala en **negrita**, el NS de menor latencia. Este anexo brinda una visión completa de las opciones de conectividad disponibles en la red simulada, lo que permite analizar las distintas alternativas para enrutar el tráfico y garantizar el cumplimiento de los parámetros de QoS establecidos, así como los SLA.

**Tabla 9 Solicitud de UE a servicios deUHD video *streaming*, videovigilancia, experiencias inmersivas [Elaboración propia].**

Solicitud (fuente - destino)	(ID trayectoria, nodos recorridos E2E)	Trayectorias E2E con parámetros QoS (IDTrayectoria, [BW, retardo, pérdida de paquetes, <i>jitter</i> ])	Trayectorias que cumplen con el retardo, del <i>template</i> de QoS (IDTrayectoria: retardo)
UE - UHD video <i>streaming</i> , videovigilancia, experiencias inmersivas	0 [1 2 6 3 4 7]	0 [50 2.3 0.3004 12]	0: 2.3
	1 [1 2 6 3 5 4 7]	1 [50 3.404 0.3004 19]	1: 3.404
	2 [1 2 6 3 5 8 4 7]	2 [16 4.2 0.2002 33]	3: 3.0
	3 [1 2 6 5 4 7]	3 [50 3.0 0.3004 19]	5: 2.9
	4 [1 2 6 5 8 4 7]	4 [16 3.8 0.2002 33]	6: 1.5
	5 [1 2 6 5 3 4 7]	5 [50 2.9 0.3004 22]	7: 2.5
	6 [1 2 4 7]	6 [50 1.5 0.2 18]	8: 3.0
	7 [1 2 5 4 7]	7 [50 2.5 0.2102 14]	10: 2.4
	8 [1 2 5 6 3 4 7]	8 [50 3.0 0.2102 21]	11: 2.4
	9 [1 2 5 8 4 7]	9 [16 3.3 0.1102 28]	12: 3.4004
	10 [1 2 5 3 4 7]	10 [50 2.4 0.2102 17]	14: 3.1
	11 [1 3 6 2 4 7]	11 [50 2.4 0.3004 29]	<b>17: 1.4</b>
	12 [1 3 6 2 5 4 7]	12 [50 3.4004 0.31 25]	18: 2.5
	13 [1 3 6 2 5 8 4 7]	13 [16 4.2 0.2102 39]	21: 2.5
	14 [1 3 6 5 4 7]	14 [50 3.1 0.2 24]	
	15 [1 3 6 5 8 4 7]	15 [16 3.904 0.102 38]	
	16 [1 3 6 5 2 4 7]	16 [50 3.1 0.21 34]	
	<b>17 [1 3 4 7]</b>	<b>17 [50 1.4 0.2 13]</b>	
	18 [1 3 5 4 7]	18 [50 2.5 0.2 20]	
	19 [1 3 5 6 2 4 7]	19 [50 3.0 0.3004 39]	
	20 [1 3 5 8 4 7]	20 [16 3.3 0.1 34]	
21 [1 3 5 2 4 7]	21 [50 2.5 0.21 30]		

**Tabla 10 Solicitud de UE a servicios de ITS o Vehículos conectados [Elaboración propia].**

Solicitud (fuente - destino)	(ID trayectoria, nodos recorridos E2E)	Trayectorias E2E con parámetros QoS (IDTrayectoria, [BW, retardo, pérdida de paquetes, <i>jitter</i> ])	Trayectorias que cumplen con el retardo, del <i>template</i> de QoS (IDTrayectoria: retardo)
UE – ITS, Vehículos conectados	<b>0 [1 2 6 10]</b> 1 [1 2 4 5 6 10] 2 [1 2 4 5 3 6 10] 3 [1 2 4 8 5 6 10] 4 [1 2 4 8 5 3 6 10] 5 [1 2 4 3 6 10] 6 [1 2 4 3 5 6 10] 7 [1 2 5 4 3 6 10] 8 [1 2 5 6 10] 9 [1 2 5 8 4 3 6 10] 10 [1 2 5 3 6 10] 11 [1 3 6 10] 12 [1 3 4 5 6 10] 13 [1 3 4 5 2 6 10] 14 [1 3 4 8 5 6 10] 15 [1 3 4 8 5 2 6 10] 16 [1 3 4 2 6 10] 17 [1 3 4 2 5 6 10] 18 [1 3 5 4 2 6 10] 19 [1 3 5 6 10] 20 [1 3 5 8 4 2 6 10] 21 [1 3 5 2 6 10]	<b>0 [10 1.6 0.2 5]</b> 1 [10 3.3 0.3004 24] 2 [10 3.7 0.3004 24] 3 [10 4.1 0.2002 38] 4 [10 4.5 0.2002 38] 5 [10 2.6 0.3004 17] 6 [10 3.2 0.3004 27] 7 [10 3.6 0.3105 13] 8 [10 2.3 0.11 14] 9 [10 4.4 0.21 27] 10 [10 2.7 0.11 14] 11 [10 1.7 0.1 10] 12 [10 3.2 0.3004 19] 13 [10 3.5 0.4103 16] 14 [10 4 0.2 33] 15 [10 4.3 0.31 30] 16 [10 2.5 0.4 20] 17 [10 3.2 0.31005 29] 18 [10 3.6 0.4 27] 19 [10 2.3 0.1 20] 20 [10 4.4 0.3 41] 21 [10 2.6 0.21 17]	<b>0: 1.6</b> <b>11: 1.7</b>

Tabla 11 Solicitud de UE a servicios de *eHealth* [Elaboración propia].

Solicitud (fuente - destino)	(ID trayectoria, nodos recorridos E2E)	Trayectorias E2E con parámetros QoS (IDTrayectoria, [BW, retardo, pérdida de paquetes, <i>jitter</i> ])	Trayectorias que cumplen con el retardo, del <i>template</i> de QoS (IDTrayectoria: retardo)
UE – <i>eHealth</i>	0 [1 2 6 3 4 5 8] 1 [1 2 6 3 4 8] 2 [1 2 6 3 5 4 8] 3 [1 2 6 3 5 8] 4 [1 2 6 5 4 8] 5 [1 2 6 5 8] 6 [1 2 6 5 3 4 8] 7 [1 2 4 5 8] 8 [1 2 4 8] 9 [1 2 4 3 6 5 8] 10 [1 2 4 3 5 8] 11 [1 2 5 4 8] 12 [1 2 5 6 3 4 8] <b>13 [1 2 5 8]</b> 14 [1 2 5 3 4 8] 15 [1 3 6 2 4 5 8] 16 [1 3 6 2 4 8] 17 [1 3 6 2 5 4 8] 18 [1 3 6 2 5 8] 19 [1 3 6 5 4 8] 20 [1 3 6 5 8] 21 [1 3 6 5 2 4 8] 22 [1 3 4 5 8] 23 [1 3 4 8] 24 [1 3 4 2 6 5 8] 25 [1 3 4 2 5 8] 26 [1 3 5 4 8] 27 [1 3 5 6 2 4 8] 28 [1 3 5 8] 29 [1 3 5 2 4 8]	0 [16 3.4004 0.3 18] 1 [16 2.6 0.2 14] 2 [16 3.7 0.2 21] 3 [16 2.5 0.1 19] 4 [16 3.3 0.2 21] 5 [16 2.1 0.1 19] 6 [16 3.2 0.2 24] 7 [16 2.6 0.2 24] 8 [16 1.8 0.1 20] 9 [16 3.105 0.2 31] 10 [16 2.5 0.2 27] 11 [16 2.8 0.11 16] 12 [16 3.3 0.11 23] <b>13 [16 1.6 0.01 14]</b> 14 [16 2.7 0.11 19] 15 [16 3.5 0.3 35] 16 [16 2.7 0.2 31] 17 [16 3.7 0.21001 27] 18 [16 2.5 0.11 25] 19 [16 3.4 0.1 26] 20 [16 2.2 0.01 24] 21 [16 3.4 0.11 36] 22 [16 2.5 0.2 19] 23 [16 1.7 0.1 15] 24 [16 3.0 0.3 34] 25 [16 2.5 0.21 29] 26 [16 2.8 0.1 22] 27 [16 3.3 0.2 41] 28 [16 1.6 0.01 20] 29 [16 2.8 0.11 32]	<b>13: 1.6</b>

**Tabla 12 Solicitud de UE a servicios de automatización industrial, redes inteligentes [Elaboración propia].**

Solicitud (fuente - destino)	(ID trayectoria, nodos recorridos E2E)	Trayectorias E2E con parámetros QoS (IDTrayectoria, [BW, retardo, pérdida de paquetes, <i>jitter</i> ])	Trayectorias que cumplen con el retardo, del <i>template</i> de QoS (IDTrayectoria: retardo)
UE – automatización industrial, redes inteligentes	0 [1 2 6 3 4 5 9]	0 [10 3.2 0.3003 9.1]	0: 3.2
	1 [1 2 6 3 4 8 5 9]	1 [10 4.0 0.202 23.1]	2: 2.3
	2 [1 2 6 3 5 9]	2 [10 2.3 0.1 10.1]	3: 1.9
	3 [1 2 6 5 9]	3 [10 1.9 0.1 10.1]	4: 2.4
	4 [1 2 4 5 9]	4 [10 2.4 0.2 15.1]	7: 2.3
	5 [1 2 4 8 5 9]	5 [10 3.2 0.1 29.1]	<b>8: 1.4</b>
	6 [1 2 4 3 6 5 9]	6 [10 2.9 0.2 22.1]	11: 2.3
	7 [1 2 4 3 5 9]	7 [10 2.3 0.2 18.1]	12: 2.0
	<b>8 [1 2 5 9]</b>	<b>8 [10 1.4 0.01 5.1]</b>	13: 2.3
	9 [1 3 6 2 4 5 9]	9 [10 3.3 0.3 26.1]	17: 1.4
	10 [1 3 6 2 4 8 5 9]	10 [10 4.1 0.2 40.1]	
	11 [1 3 6 2 5 9]	11 [10 2.3 0.11 16.1]	
	12 [1 3 6 5 9]	12 [10 2.0 0.0001 15.1]	
	13 [1 3 4 5 9]	13 [10 2.3 0.2001 10.1]	
	14 [1 3 4 8 5 9]	14 [10 3.1 0.1 24.1]	
	15 [1 3 4 2 6 5 9]	15 [10 2.8 0.3 25.1]	
	16 [1 3 4 2 5 9]	16 [10 2.3 0.21 20.1]	
17 [1 3 5 9]	17 [10 1.4 0.0001 11.1]		

Tabla 13 Solicitud de UE a servicios de Vo5G [Elaboración propia].

Solicitud (fuente - destino)	(ID trayectoria, nodos recorridos E2E)	Trayectorias E2E con parámetros QoS (IDTrayectoria, [BW, retardo, pérdida de paquetes, <i>jitter</i> ])	Trayectorias que cumplen con el retardo, del <i>template</i> de QoS (IDTrayectoria: retardo)
UE – Vo5G	<b>0 [1 2 6 11]</b>	<b>0 [10 1.7 0.2 4]</b>	<b>0: 1.7</b>
	1 [1 2 4 5 6 11]	1 [10 3.4 0.3004 23]	1: 3.4
	2 [1 2 4 5 3 6 11]	2 [10 3.8 0.3004 23]	2: 3.8
	3 [1 2 4 8 5 6 11]	3 [10 4.2 0.2002 37]	5: 2.7
	4 [1 2 4 8 5 3 6 11]	4 [10 4.6 0.2002 37]	6: 3.3
	5 [1 2 4 3 6 11]	5 [10 2.7 0.3004 16]	7: 3.7
	6 [1 2 4 3 5 6 11]	6 [10 3.3 0.3004 26]	8: 2.4
	7 [1 2 5 4 3 6 11]	7 [10 3.7 0.3105 12]	9: 4.5
	8 [1 2 5 6 11]	8 [10 2.4 0.11 13]	10: 2.8
	9 [1 2 5 8 4 3 6 11]	9 [10 4.5 0.21 26]	11: 1.8
	10 [1 2 5 3 6 11]	10 [10 2.8 0.11 13]	12: 3.3
	11 [1 3 6 11]	11 [10 1.8 0.1 9]	13: 3.6
	12 [1 3 4 5 6 11]	12 [10 3.3 0.3004 18]	15: 4.4
	13 [1 3 4 5 2 6 11]	13 [10 3.6 0.41 15]	16: 2.6
	14 [1 3 4 8 5 6 11]	14 [10 4.1 0.2002 32]	17: 3.3
	15 [1 3 4 8 5 2 6 11]	15 [10 4.4 0.31 29]	18: 3.7
	16 [1 3 4 2 6 11]	16 [10 2.6 0.4 19]	19: 2.4
	17 [1 3 4 2 5 6 11]	17 [10 3.3 0.3105 28]	21: 2.7
	18 [1 3 5 4 2 6 11]	18 [10 3.7 0.4 26]	
	19 [1 3 5 6 11]	19 [10 2.4 0.1 19]	
	20 [1 3 5 8 4 2 6 11]	20 [10 4.5 0.3 40]	
	21 [1 3 5 2 6 11]	21 [10 2.7 0.21 16]	