

# UCUENCA

## Universidad de Cuenca

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

### **Análisis de Algoritmos de Aprendizaje Máquina aplicados a la asignación de recursos en la red de acceso para redes 5G con segmentación**

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

#### **Autores:**

Sthefany Paola Carabajo Avila

Michelle Giovanna Chimbo Vega

#### **Director:**

Darwin Fabián Astudillo Salinas

ORCID:  0000-0001-7644-0270

#### **Co-Director:**

Luis Patricio Tello Oquendo

ORCID:  0000-0002-5274-666X

**Cuenca, Ecuador**

2023-09-26

## Resumen

El presente proyecto de titulación se enfoca en el estudio y mejora del método de acceso aleatorio en redes 5G, aplicando dos enfoques para la administración dinámica de preámbulos. En este contexto, los usuarios solicitan recursos de enlace ascendente mediante la transmisión de preámbulos, los cuales se caracterizan por ser firmas ortogonales. Sin embargo, un problema que se presenta es la posible congestión y colapso de la red cuando múltiples firmas idénticas se transmiten en un corto período de tiempo. El primer método consiste en el uso del aprendizaje supervisado mediante la implementación de una red neuronal que determina una salida ante un conjunto de datos que contienen información sobre los preámbulos que han beneficiado previamente al sistema durante pruebas realizadas. La red neuronal procesa esta información y genera salidas que indican qué preámbulos son más adecuados para evitar colapsos y mejorar la probabilidad de acceso exitoso a los recursos de enlace ascendente. El segundo método propuesto implica el uso del aprendizaje reforzado, que implementa una política para la administración de preámbulos basada en *Q-learning*. En este enfoque, se emplea una función denominada  $Q$  que evalúa las acciones tomadas en estados específicos del sistema. El sistema se retroalimenta continuamente, reforzando aquellas elecciones que benefician al rendimiento del sistema y castigando aquellas que lo perjudican. De esta manera, el sistema de acceso aleatorio puede aprender a asignar preámbulos de manera más inteligente y efectiva, adaptándose a las condiciones de la red y optimizando la utilización de los recursos disponibles.

*Palabras clave:* 5ta Generación (5G), segmentos de red, redes neuronales artificiales, aprendizaje máquina, Q-Learning



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

**Repositorio Institucional:** <https://dspace.ucuenca.edu.ec>

## Abstract

This degree project focuses on the study and improvement of the random access method in 5G networks, applying two approaches for the dynamic management of preambles. In this context, users request uplink resources by transmitting preambles, which are characterized by being orthogonal signatures. However, one problem that arises is the possible congestion and network collapse when multiple identical signatures are transmitted in a short period of time. The first method consists of the use of supervised learning by training a neural network using a set of data that contains information about the preambles that have previously benefited the system during tests carried out. The neural network processes this information and generates outputs that indicate which preambles are best suited to avoid crashes and improve the probability of successful access to uplink resources. The second proposed method involves the use of reinforcement learning, which implements a policy for preamble management based on *Q-learning*. In this approach, a function called Q is used that evaluates actions taken in specific states of the system. The system continually feeds itself back, reinforcing those choices that benefit the system's performance and punishing those that harm it. In this way, the random access system can learn to allocate preambles more intelligently and effectively, adapting to network conditions and optimizing the utilization of available resources.

*Keywords:* 5G Network, Slicing Network, Neural Networks, Machine Learning, Q-Learning



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

**Institutional Repository:** <https://dspace.ucuenca.edu.ec>

## Índice general

<b>1. Introducción</b>	<b>19</b>
1.1. Justificación . . . . .	21
1.2. Alcance . . . . .	22
1.3. Objetivos . . . . .	23
1.3.1. Objetivo general . . . . .	23
1.3.2. Objetivos específicos . . . . .	23
1.4. Estructura del documento . . . . .	24
<b>2. Marco teórico</b>	<b>26</b>
2.1. Tecnología 5G . . . . .	26
2.1.1. Características . . . . .	26
2.1.2. Arquitectura de red . . . . .	27
2.1.3. Aplicaciones . . . . .	29
2.1.4. Control de acceso al medio . . . . .	30
2.1.4.1. Procedimiento de acceso aleatorio . . . . .	31
2.1.4.2. Procedimiento de acceso aleatorio de 4 Mensajes ba- sado en contención . . . . .	34
2.1.4.3. Procedimiento de retroceso . . . . .	39
2.1.5. Técnica de segmentación de red . . . . .	39
2.2. Aprendizaje automático . . . . .	42
2.2.1. Aprendizaje supervisado . . . . .	43
2.2.1.1. Técnicas de aprendizaje supervisado . . . . .	43
2.2.1.2. Modelos de aprendizaje supervisado . . . . .	44
2.2.1.3. Redes neuronales y su aplicación en el aprendizaje su- pervisado . . . . .	45
2.2.2. Aprendizaje no supervisado . . . . .	47
2.2.3. Aprendizaje semi-supervisado . . . . .	47
2.2.4. Aprendizaje reforzado . . . . .	48
2.2.5. Aprendizaje profundo . . . . .	50
<b>3. Trabajos relacionados</b>	<b>51</b>

3.1. Antecedentes a la segmentación . . . . .	51
3.2. Segmentación de red en función de requisitos . . . . .	52
3.3. Uso de aprendizaje máquina aplicado en 5G . . . . .	54
3.3.1. Avances tecnológicos con aprendizaje supervisado . . . . .	54
3.3.2. Avances tecnológicos con aprendizaje reforzado . . . . .	54
3.4. Políticas de asignación de preámbulos en redes segmentadas . . . . .	56
3.5. Conclusiones . . . . .	58
<b>4. Diseño e Implementación</b>	<b>59</b>
4.1. Caso de estudio . . . . .	59
4.2. Metodología . . . . .	60
4.3. Capacidad de trabajo de la celda . . . . .	61
4.4. Modelos de distribución de arribos . . . . .	62
4.4.1. Tráfico mMTC . . . . .	62
4.4.2. Tráfico H2H . . . . .	63
4.4.3. Tráfico emBB . . . . .	63
4.5. Configuración del RACH . . . . .	64
4.6. Enfoque de Redes neuronales . . . . .	65
4.6.1. Recopilación de datos . . . . .	66
4.6.1.1. Comunicación mMTC . . . . .	67
4.6.1.2. Comunicación emBB . . . . .	68
4.6.1.3. Comunicación H2H . . . . .	68
4.6.2. Procesamiento de datos . . . . .	69
4.6.3. Diseño de la red neuronal artificial . . . . .	74
4.7. Enfoque de aprendizaje reforzado . . . . .	78
4.7.1. Modelo del sistema . . . . .	79
4.7.1.1. Acción . . . . .	79
4.7.1.2. Estado . . . . .	79
4.7.1.3. Recompensa . . . . .	84
4.7.2. Implementación del algoritmo Q-Learning . . . . .	87
4.7.2.1. Inicialización . . . . .	89
4.7.2.2. Selección de preámbulos para cada segmento . . . . .	89
4.7.2.3. Obtención de resultados . . . . .	93

4.7.2.4. Actualización de la tabla $Q(s, a)$ . . . . .	93
<b>5. Análisis y resultados</b>	<b>94</b>
5.1. Indicadores de rendimiento . . . . .	94
5.2. Análisis de arquitecturas para la red neuronal artificial . . . . .	95
5.3. Análisis de hiperparámetros óptimos para <i>Q-learning</i> . . . . .	102
5.3.1. Factor de aprendizaje $\alpha$ . . . . .	103
5.3.2. Factor de descuento $\gamma$ . . . . .	104
5.3.3. Tiempo de entrenamiento . . . . .	106
5.4. Análisis comparativo entre algoritmos . . . . .	108
5.4.1. Análisis en el escenario de máximo esfuerzo . . . . .	108
5.4.2. Probabilidad de acceso en escenario múltiples . . . . .	116
<b>6. Conclusiones y recomendaciones</b>	<b>122</b>
6.1. Conclusiones . . . . .	122
6.2. Recomendaciones . . . . .	124
6.3. Trabajos Futuros . . . . .	124
<b>Referencias</b>	<b>126</b>
<b>Apendices</b>	<b>136</b>
<b>A. Configuración de Acceso Aleatorio para formatos de preámbulo 0-3</b>	<b>137</b>
<b>B. Recompensas y penalizaciones asociadas a cada estado del sistema</b>	<b>140</b>

## Índice de figuras

2.1. Descripción general de la arquitectura UE-NG-RAN-5GC [1]. . . . .	28
2.2. Arquitectura NSA [1]. . . . .	28
2.3. Arquitectura SA [1]. . . . .	29
2.4. Entorno IoT en 5G [2]. . . . .	29
2.5. Exigencias de retardo y ancho de banda de aplicaciones IoT [3]. . . . .	30
2.6. Comparación de canales PRACH y UL [4]. . . . .	32
2.7. Ejemplo de prach-ConfigIndex [4]. . . . .	33
2.8. Variables que intervienen en el procedimiento de acceso aleatorio [5]. . . . .	35
2.9. Colisión en la transmisión de Msg1 [5]. . . . .	36
2.10. Colisión en la transmisión de Msg3 [5]. . . . .	38
2.11. Grupos de segmentación [6]. . . . .	41
2.12. Servicios y segmentos de 5G [7]. . . . .	42
2.13. Modelo de una red neuronal artificial [8] . . . . .	46
2.14. Diagrama de aprendizaje reforzado. . . . .	49
4.1. Periodo de decisión $T_{SIB2}$ . . . . .	60
4.2. Diagrama de Flujo para el diseño de una RNA . . . . .	66
4.3. Intensidad de peticiones de acceso para un total de 5000, 10000, 30000 UEsM. . . . .	67
4.4. Intensidad de peticiones de acceso para un total de 500, 1000, 1500 UEsB. . . . .	68
4.5. Intensidad de peticiones de acceso para un tráfico H2H. . . . .	69
4.6. Distribución temporal de arribos UEsM (primeras transmisiones de preám- bulo), transmisiones de preámbulo totales, colisiones y accesos exito- sos por RAO, sin control de acceso implementado [9]. . . . .	70
4.7. Gráfica comparativa entre las peticiones de acceso y el acceso exitoso para una comunicación mMTC con un valor total de 30000 UEsM. . . . .	73
4.8. Gráfica comparativa entre las peticiones de acceso y el acceso exitoso para una comunicación eMBB con un valor total de 1500 UEsB. . . . .	73
4.9. Diagrama de Flujo para el diseño de un <i>Q-learning</i> . . . . .	78
4.10. Definición de estado para <i>Q-learning</i> . . . . .	81

4.11. Probabilidad de acceso exitoso de UEs en función de la asignación de preámbulos. . . . .	82
4.12. Diagrama de flujo de la implementación de <i>Q-learning</i> . . . . .	88
5.1. Resultados del escenario de mínimo esfuerzo para las diferentes arquitecturas RNA. . . . .	98
5.2. Resultados del escenario de máximo esfuerzo para las diferentes arquitecturas de la red neuronal. . . . .	99
5.3. Gráficas comparativas entre la aplicación de la RNA y la función determinista para el segmento mMTC en un escenario de máximo esfuerzo .	101
5.4. Gráficas comparativas entre la aplicación de la RNA y la función determinista para el segmento emBB en un escenario de máximo esfuerzo .	102
5.5. Congestión y acceso exitoso para <i>Q-learning</i> a medida que $\alpha$ varía. . .	104
5.6. Congestión y acceso exitoso para <i>Q-learning</i> a medida que $\gamma$ varía. . .	105
5.7. Congestión y acceso exitoso para <i>Q-learning</i> a medida que el tiempo de entrenamiento varía. . . . .	107
5.8. Gráfica comparativa entre el algoritmo de <i>Q-learning</i> vs la asignación fija de preámbulos para una comunicación mMTC en un escenario de máximo esfuerzo . . . . .	110
5.9. Gráfica comparativa entre la RNA vs la asignación fija de preámbulos para una comunicación mMTC en un escenario de máximo esfuerzo . .	110
5.10. Gráfica comparativa entre la RNA vs el algoritmo de <i>Q-learning</i> para una comunicación mMTC en un escenario de máximo esfuerzo . . . . .	111
5.11. Gráfica comparativa entre la asignación fija de preámbulos vs el algoritmo de <i>Q-learning</i> para una comunicación emBB en un escenario de máximo esfuerzo . . . . .	113
5.12. Gráfica comparativa entre la asignación fija de preámbulos vs la RNA para una comunicación emBB en un escenario de máximo esfuerzo . .	113
5.13. Gráfica comparativa entre la RNA vs el algoritmo de <i>Q-learning</i> de preámbulos para una comunicación emBB en un escenario de máximo esfuerzo . . . . .	114
5.14. Resultados del escenario de máximo esfuerzo para las diferentes experimentaciones. . . . .	115

5.15. Probabilidad de acceso exitoso vs número de UEsM esperados. . . . . 118

5.16. Probabilidad de acceso exitoso vs número de UEsB esperados. . . . . 119

5.17. Probabilidad de acceso exitoso vs número de UEs esperados. . . . . 120

## Índice de tablas

4.1. Modelos de tráfico para mMTC. . . . .	62
4.2. Configuración del RACH. . . . .	65
4.3. Valores objetivos de entrenamiento de la RNA. . . . .	77
4.4. Definición del parámetro $CV_{NU_{Es}}$ en función de $mNU_{Es}$ . . . . .	83
4.5. Composición de estados en Q-Learning con parámetros y valores posibles. . . . .	84
4.6. Ejemplos de combinaciones de estados y recompensas del algoritmo <i>Q-learning</i> . . . . .	87
5.1. Comparación de parámetros de entrenamiento entre las diferentes arquitecturas de la red neuronal. . . . .	96
5.2. Arribos esperados para los escenarios de entrenamiento de la RNA. . .	97
5.3. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de mínimo esfuerzo. . . . .	99
5.4. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de máximo esfuerzo. . . . .	100
5.5. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de máximo esfuerzo . . . . .	102
5.6. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando $\alpha$ varía. . . . .	104
5.7. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando $\gamma$ varía. . . . .	106
5.8. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando el tiempo de entrenamiento varía. . . . .	108
5.9. Valor promedio de los preámbulos transmitidos y la probabilidad de acceso exitoso obtenida con diferentes métodos de asignación de preámbulos. . . . .	116
5.10. Parámetros de Simulación. . . . .	116
5.11. Intervalos de confianza del 95% para mMTC obtenidos con diferentes métodos de asignación de preámbulos. . . . .	118

5.12. Intervalos de confianza del 95 % para emBB obtenidos con diferentes métodos de asignación de preámbulos. . . . .	119
5.13. Intervalos de confianza del 95 % obtenidos con diferentes métodos de asignación de preámbulos. . . . .	120
5.14. Porcentaje de mejora con respecto a la asignación fija de preámbulos .	121
A.1. Configuración de acceso aleatorio para formatos de preámbulo 0-3 (Tabla tomada de [10]) . . . . .	137

### Dedicatoria

Deseo dedicar este trabajo a mi querida madre, Patricia Vega, por su inquebrantable apoyo en cada paso de esta travesía. Tu confianza en mí y tu apoyo constante han sido un faro guía en mi camino. Siempre he encontrado en ti la seguridad y el espacio para compartir tanto mis alegrías como mis desafíos.

A mi padre, Geovanny Chimbo, dedico este logro como un testimonio de tu dedicación y sacrificio. Tu dedicación y el tiempo invertido en llevarme a la universidad para asistir a mis clases han sido esenciales para mi desarrollo académico y personal.

A mi hermano, David Chimbo, quien ha sido un faro de profesionalismo y dedicación en mi vida. A lo largo de mi trayectoria universitaria, sus consejos sabios y su apoyo constante han sido invaluable.

A Randy, quien me acompañó durante innumerables noches de estudio y siempre estuvo a mi lado, brindándome su inquebrantable compañía y cariño. A pesar de que ya no está físicamente conmigo, su espíritu y su amor perduran en mi corazón. Siempre serás parte de mi vida y de este logro.

A todos mis familiares y amigos, quiero expresar mi sincero agradecimiento por su inquebrantable compañía. Cada palabra de aliento, cada gesto de apoyo y cada momento compartido han sido una fuente inagotable de inspiración y motivación a lo largo de este camino.

**Michelle Chimbo**

### Dedicatoria

Con cariño y profundo amor, deseo dedicar esta tesis a mis padres, quienes han sido los pilares inquebrantables en mi vida. A mi mamá, Zoila Ávila, le agradezco por su arduo esfuerzo en brindarnos educación y por ser mi guía en cada paso que he dado. Tus palabras de aliento y tu amor incondicional me han dado la fuerza para alcanzar mis metas. A mi papá, Luis Carabajo, le dedico mi trabajo con gratitud por su comprensión, apoyo y amor constante. Eres mi ejemplo de perseverancia y valentía, y estoy eternamente agradecido por todo lo que has hecho por mí.

También quiero dedicar esta tesis a mis queridos hermanos, Jorge y Cristian. Sus palabras de aliento y apoyo incondicional han sido mi motor en momentos difíciles. Agradezco su presencia constante y su cariño sincero.

Y no puedo olvidar mencionar a mis fieles compañeros de batalla, Erick y Panchito, quienes han estado a mi lado durante la elaboración de esta tesis. Su presencia y afecto han sido una fuente constante de ánimo, paciencia y alegría. Su compañía ha hecho que cada desafío valga la pena. Cada página de este trabajo es de ustedes también.

Con amor y gratitud,  
**Sthefany Carabajo**

## Agradecimientos

Deseamos expresar nuestro más sincero agradecimiento a nuestro director, Ing. Fabián Astudillo, por su invaluable apoyo y dirección durante el desarrollo de nuestra tesis. Su orientación experta y su dedicación han sido fundamentales para la realización de este trabajo. También queremos agradecer a nuestro co-director, PhD. Luis Patricio Tello, por su generosa contribución del código de MATLAB para simular la red 5G, el cual ha sido crucial para el éxito de esta investigación. También extendemos un agradecimiento especial a nuestro director de Carrera, Ing. Alcides Araujo, por su inquebrantable apoyo, y por su compromiso en nuestra formación académica y profesional.

Este logro no habría sido posible sin el respaldo constante de nuestros amigos, familiares y compañeros, quienes han estado a nuestro lado en cada paso del camino. Sus consejos, orientación y aliento han sido un factor crucial para nuestro crecimiento y logros. Cada conversación, cada idea compartida y cada momento de colaboración han dejado una huella imborrable en este proyecto y en nuestra formación personal y profesional.

Queremos dar un enorme agradecimiento a nuestros amigos Machetes. Ustedes han sido parte esencial de este viaje y siempre han estado ahí con su apoyo y cariño. Esperamos que sigan formando parte de nuestras vidas, porque su amistad nos llena de alegría y nos motiva a seguir adelante. Gracias por creer en nosotros y por ser increíbles amigos.

**LOS AUTORES**

**Abreviaciones y acrónimos**

**1G** 1ra Generación. 19

**2G** 2da Generación. 19

**3G** 3ra Generación. 19

**3GPP** 3rd Generation Partnership Project. 62, 67

**4G** 4ta Generación. 19, 27–30

**4G LTE** 4th Generation Long-Term Evolution. 19, 28, 31

**5G** 5ta Generación. 2, 19–31, 39–42, 48, 51, 52, 55, 56, 58–61, 79, 122, 123

**5GC** 5G Core. 27

**BDA** Big Data Analytics. 54

**BI** Indicador de Retroceso. 36, 39

**C-RAN** Cloud RAN. 56

**COQRA** Correlated *Q-learning* based interslice RB Allocation. 56

**CU** Centralized Unit. 56

**DL** Downlink. 56

**DTC** Decision Tree Classifier. 44

**DU** Distributed Unit. 56

**emBB** Enhanced Mobile Broadband. 19, 22, 40, 52, 53, 56, 57, 61, 63, 65, 68–72, 74, 90, 91, 93, 99, 101, 103, 108, 111, 112, 114, 119, 122, 123

**gNodeB** Next Generation Node B. 20, 30, 32–38, 50, 57, 60

**GSM** Global System for Mobile Communications. 19

**H-NOMA** Heterogeneous Non-Orthogonal Multiple Acces. 53

**H2H** Human to Human. 19, 22, 61, 63, 65, 68, 69, 72, 90, 91, 93, 103, 109, 122, 123

**HARQ** Hybrid Automatic Repeat Request. 37, 38

**IA** Inteligencia Artificial. 55

**IoT** Internet of Things. 27, 30, 41, 52

**IP** Protocolo de Internet. 19

**KPI** Indicadores Clave de Rendimiento. 23, 25, 78, 94

**M2M** Machine-to-Machine. 19, 57

**MARL** Multi-Agent Reinforcement Learning. 56

**MEC** Multi-Access Edge Computing. 52

**MIB** Main Information Block. 34

**MIoT** Massive IoT. 52

**ML** Machine Learning. 20–22, 24, 42, 43, 47, 49, 51, 55, 58, 60, 94, 117, 118, 121–123

**mMTC** Massive Machine-Type Communications. 19, 22, 40, 41, 52, 53, 57, 61, 62, 65, 67, 69–72, 74, 90, 91, 93, 98–100, 103, 108, 109, 111, 114, 115, 118, 122, 123

**mmWave** ondas milimétricas. 27

**NB** Naïve Bayes. 44

**NFV** Virtualización de Funciones de Red. 51, 52

**NG-RAN** New generation RAN. 27

**NRflex** New Radio Flexibility. 53

**NSA** Non-Stand Alone. 28

**PDCCH** Physical Downlink Control Channel. 37

**PDSCH** Physical Downlink Shared Channel. 36

- PRACH** Physical Random Access Channel. 30–35, 59, 116
- PUCCH** Physical Uplink Control Channel. 31
- PUSCH** uplink shared Physical channel. 36, 37
- QoS** Calidad de Servicio. 51, 53, 56, 57
- RACH** Canal de Acceso Aleatorio. 22, 24, 39, 57, 64, 94
- RAN** Red de Acceso de Radio. 19, 55–57
- RAO** Oportunidad de Acceso Aleatorio. 20, 22, 31–34, 39, 60, 62–64, 66, 68, 70, 72, 79–82, 86, 89, 93, 117
- RAP** Random Access Procedure. 20, 22, 24, 26, 31, 34, 59, 61, 64, 68, 94
- RAR** Respuesta de Acceso Aleatorio. 36, 37
- RL** Reinforcement Learning. 48, 49, 55
- RNA** Red Neuronal Artificial. 23, 24, 44, 45, 60, 65, 66, 74, 76, 77, 94, 96, 97, 100, 101, 109–114, 116, 118–124
- RRU** Radio Remote Unit. 56
- SA** Stand Alone. 28
- SDN** Redes Definidas por Software. 51, 52, 55
- SFC** Service Function Chaining. 52
- SI** System Information. 34
- SIB** System Information Block. 34, 60
- SIB2** System Information Block Type 2. 34, 35, 37, 57, 60, 67, 68, 71, 79, 80, 89, 93, 95, 97, 98, 101, 103, 109
- SoDeMa** Software-Defined Multiple-Access. 53
- TA** Alineación Temporal. 36, 37
- TSQL** Two-Stage *Q-learning*. 56

**UE** Equipo de Usuario. 20, 22, 27, 30, 31, 33–40, 42, 53, 60–64, 79–83, 85, 86, 90, 91, 93, 98–100, 103, 117–123

**UEsB** Users Equipments for emBB. 68, 70, 112, 116, 117

**UEsH** Users Equipments for H2H. 68, 69, 116

**UEsM** Users Equipments for mMTC. 67, 70, 111, 116, 117

**UIT** Union Internacional de Telecomunicaciones. 26, 40

**UL** Uplink. 30–32, 34, 37, 56, 57, 65, 69–72, 98

**URLLC** Ultra Reliable Low Latency Communications. 19, 40, 41, 52, 53, 56, 57

**VANET** Vehicular Ad-Hoc Network. 52

**WRAR** Window Random Access Response. 37

## Capítulo 1 Introducción

La telefonía móvil ha experimentado una continua evolución a lo largo de las últimas décadas. La presentación del primer teléfono móvil en 1973, de la mano de la empresa Motorola, generó el comienzo de la tecnología inalámbrica de 1ra Generación (1G). La tecnología 1G es analógica, insegura y con una pésima comunicación de voz. En 1980 se implementa la tecnología de comunicaciones móviles globales 2da Generación (2G) o también conocida como Global System for Mobile Communications (GSM); esta tecnología es digital, lo que facilita el envío de voz y datos digitales [11]. A finales de los 90 se incorporó la tecnología de 3ra Generación (3G). La tecnología 3G es un híbrido entre la conmutación de paquetes y de circuitos que incrementó las tasas de transmisión de datos y habilitó aplicaciones multimedia. En el año 2008 se implementó la tecnología 4ta Generación (4G) basada totalmente en conmutación de paquetes. 4G proporciona alta velocidad, calidad, capacidad, seguridad y servicios de bajo coste para voz, datos, multimedia e Internet a través del Protocolo de Internet (IP) [12]. En el contexto actual, la red 5G se destaca por su velocidad superior de transmisión de datos en comparación con la red de cuarta generación 4th Generation Long-Term Evolution (4G LTE), su capacidad para manejar un mayor número de dispositivos y la posibilidad de comunicaciones masivas de máquina a máquina o también conocidas por sus siglas en inglés Machine-to-Machine (M2M).

En este contexto, la demanda en la red 5G es considerablemente mayor que en 4G (mayor flujo de usuarios y aplicaciones), lo que resulta en la sobreexplotación constante de los recursos de radio [13]. La gestión eficiente de estos recursos, el procesamiento de volúmenes de tráfico sustanciales, la garantía de seguridad y la adaptación a cargas variables en el tiempo se presentan como nuevos desafíos. Para abordar esta problemática, la división de recursos y funciones de radio es fundamental para personalizar el abastecimiento de servicios para la Red de Acceso de Radio (RAN).

En 5G se ofrecen los servicios de Enhanced Mobile Broadband (emBB), Massive Machine-Type Communications (mMTC), Human to Human (H2H) y Ultra Reliable Low Latency Communications (URLLC). La comunicación emBB requiere de un ancho de banda muy alto. Los servicios mMTC se relacionan con una conectividad ultra densa. La comunicación URLLC requiere una calidad de servicio con seguridad de comuni-

cación y latencia baja. Todas las características mencionadas no pueden construirse bajo una red homogénea, es por ello que, 5G está diseñada de tal forma que se puedan generar construcciones de segmentos de red [14]. La segmentación gestiona la red de tal forma que los cortes puedan llegar a ser homogéneos, heterogéneos o dinámicos. Un segmento se compone de una colección de funciones y configuraciones de red que se combinan para formar modelos específicos. La técnica de segmentación abarca varios dominios de aplicación como de las redes de acceso por radio, las redes centrales, y las redes de transporte [15].

El Random Access Procedure (RAP) implica la transmisión de códigos de acceso (preámbulos), por parte de los usuarios para solicitar recursos de enlace ascendente. Sin embargo, 5G tiene un número limitado de preámbulos disponibles, los cuales a su vez deben ser asignados de manera adecuada y equitativa entre los diferentes segmentos de la red. Con el aumento del número de Equipo de Usuarios (UEs) se incrementa la probabilidad de que ocurran problemas de colisión de preámbulos durante el RAP. En cada Oportunidad de Acceso Aleatorio (RAO) las peticiones de acceso consisten en una transacción de mensajes. El UE envía un preámbulo a Next Generation Node B (gNodeB) para solicitar acceso. Si varios UEs intentan conectarse en el mismo RAO y utilizan el mismo preámbulo, se producirán colisiones que generarán interferencias y retrasos en el proceso de acceso. Las colisiones afectan negativamente a la eficiencia de la red y la calidad del servicio para los usuarios. Este problema se agrava en entornos con alta densidad de dispositivos, donde el número de peticiones de acceso que arriban por RAO puede ser muy elevado.

En este contexto, se vuelve imperativo encontrar soluciones que puedan prevenir o atenuar los problemas derivados de las colisiones. A medida que la demanda de acceso continúa en aumento, se genera un volumen significativo de datos que requiere procesamiento en tiempo real. No obstante, abordar esta cantidad de información utilizando técnicas convencionales, como modelos estadísticos, puede ser insuficiente para abordar las exigencias de la red. Para enfrentar este desafío, emerge un potencial considerable en los algoritmos de Machine Learning (ML), en particular en el aprendizaje supervisado y reforzado. Estas técnicas tienen la capacidad de mejorar la probabilidad de acceso exitoso, lo que a su vez permite optimizar la asignación de

recursos de radio y reducir al mínimo las posibles colisiones de preámbulos.

### 1.1. Justificación

La automatización de funciones de segmentación de red para manejar grandes volúmenes de datos en intervalos cortos de tiempo resulta un desafío considerable para los operadores humanos [15]. La necesidad de simplificar las operaciones de red requiere nuevos enfoques. ML ha revolucionado la industria al proporcionar soluciones automáticas, simples y adaptables a diversos entornos. En este sentido, los modelos de ML abarcan categorías como el aprendizaje supervisado, no supervisado y por refuerzo. En el marco de este documento, se centra en los enfoques de aprendizaje supervisado y por refuerzo.

El aprendizaje supervisado se fundamenta en algoritmos que asignan salidas a entradas según un conjunto de datos de entrenamiento que ya ha sido previamente analizado. Estos algoritmos han demostrado su eficacia en el procesamiento de grandes volúmenes de datos y han alcanzado éxito en diversas aplicaciones. En particular, en el ámbito de interés, el diseño y la gestión de redes, donde se maneja una gran cantidad de información valiosa, el aprendizaje supervisado ha emergido como un área de investigación relevante [16].

El aprendizaje reforzado, por su parte, es un enfoque del ML que capacita a un agente para tomar decisiones en un entorno específico, basándose en un sistema de recompensas y penalizaciones. Este tipo de algoritmo ha demostrado ser ampliamente utilizado y exitoso en las redes celulares, ya que posee la capacidad de adaptarse a entornos altamente cambiantes y tomar decisiones en tiempo real. Esta adaptabilidad y escalabilidad hacen del aprendizaje reforzado una técnica especialmente adecuada para enfrentar los desafíos dinámicos de las redes de comunicación.

En este proyecto de titulación, se desarrollará un enfoque de aprendizaje automático para la asignación dinámica de recursos de radio en una red 5G segmentada, mediante la aplicación de algoritmos basados en ML. La principal motivación detrás de esta investigación es abordar los desafíos que surgen en entornos de alta demanda de acceso a la red, donde la congestión y las colisiones de preámbulos son frecuentes.

La propuesta busca aumentar la probabilidad de acceso exitoso al Canal de Acceso Aleatorio (RACH), mejorando así la calidad del servicio y la eficiencia de la red. La ventaja fundamental de esta investigación radica en la capacidad de adaptarse dinámicamente a condiciones cambiantes de la red, lo que permite un uso óptimo de los recursos de radio y una reducción significativa de las colisiones de preámbulos. Además, al utilizar algoritmos de aprendizaje máquina, se espera que el sistema pueda aprender y adaptarse a patrones de tráfico cambiantes en tiempo real, lo que lo hace altamente eficaz en un entorno dinámico y altamente congestionado. En última instancia, esta propuesta contribuirá a la mejora sustancial de la eficiencia y la calidad del servicio en redes 5G, lo que es esencial en un contexto de comunicaciones móviles cada vez más exigente y diverso.

## 1.2. Alcance

En el marco de este proyecto de titulación, se ha centrado en la segmentación de la red como elemento fundamental. Esta segmentación implica la asignación de subconjuntos de preámbulos a los tres tipos de servicios de comunicación clave en 5G: mMTC, H2H y emBB, los cuales han sido definidos y delimitados en segmentos de red específicos. El propósito principal de esta investigación es desarrollar un enfoque que mejore la probabilidad de acceso exitoso de los UEs a los recursos físicos del canal de comunicación en este entorno segmentado. Esta estrategia de segmentación se convierte en un pilar esencial para la mejora del rendimiento del RAP y, por ende, de la calidad y eficiencia de la red.

En el enfoque aplicado, se dará prioridad a la optimización individual de cada segmento de red. Esta estrategia busca evitar comprometer el rendimiento global del sistema, evitando situaciones en las que la búsqueda de un alto rendimiento en un segmento afecte negativamente a otros. Para alcanzar este objetivo, se propone la implementación de modelos de ML que gestionen de forma dinámica los subgrupos de preámbulos en la red. Esta dinámica permitirá maximizar el rendimiento en los accesos a la red durante cada RAO.

Para llevar a cabo la implementación propuesta, se ha seleccionado el software ma-

temático MATLAB debido a su amplia gama de herramientas y funciones que facilitan la programación en el contexto del canal de acceso.

Entre los resultados que se esperan alcanzar, nuestro objetivo principal es identificar cuál de las propuestas planteadas logra los mejores resultados en términos de rendimiento. Este rendimiento se evaluará a través de un conjunto de Indicadores Clave de Rendimiento (KPI), que incluyen la probabilidad de acceso exitoso, el número de colisiones registradas, la precisión en la decodificación de preámbulos, la capacidad máxima de arribos, entre otros, tal como se detallan en la Sección 5.1. Además de medir estos KPI, también buscamos obtener una comprensión más profunda de las colisiones que ocurren en el proceso de acceso aleatorio. Esta comprensión nos permitirá desarrollar políticas de aprendizaje específicas con el propósito de aumentar de manera efectiva y eficiente la probabilidad de acceso al canal de comunicación.

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

Diseñar un esquema de asignación dinámica de recursos radio para optimizar la probabilidad de acceso exitoso al canal de acceso aleatorio en una red 5G con segmentación de red usando algoritmos basados en aprendizaje máquina.

#### **1.3.2. Objetivos específicos**

El presente trabajo tiene los siguientes objetivos específicos:

- Implementar algoritmos de aprendizaje máquina con modelos de aprendizaje reforzado y supervisado para la asignación de recursos de radio dentro de la red 5G.
- Diseñar políticas de decisión capaces optimizar los recursos de acceso al medio dentro de los algoritmos de aprendizaje reforzado.
- Comparar los resultados de los modelos de aprendizaje aplicados, que incluyen una Red Neuronal Artificial (RNA) como enfoque de aprendizaje supervisado y

un algoritmo de *Q-Learning* como estrategia de aprendizaje reforzado.

- Analizar las posibilidades de generalización de los algoritmos implementados al aplicarlos en comunicaciones mMTC y emBB.

#### 1.4. Estructura del documento

El documento se organiza de la siguiente manera:

- **Capítulo 2:** Presenta el marco teórico necesario para la implementación del esquema de asignación de recursos de radio. En este capítulo se explican las características y el funcionamiento de la red 5G, con especial énfasis en el RAP. Además, se introducen los conceptos y la formulación necesaria dentro de las técnicas de ML, que se utilizarán para solucionar la colisión de preámbulos.
- **Capítulo 3:** Introduce de manera organizada y descriptiva las principales contribuciones de diversos autores para abordar la problemática planteada en el Capítulo 1. Se presenta una breve descripción de la estructura y conceptos involucrados en el desarrollo, y resultados de las diferentes contribuciones.
- **Capítulo 4:** Proporciona una detallada descripción del entorno de trabajo. Se aborda en profundidad el RACH, incluyendo su capacidad de aceptación de usuarios y los recursos disponibles para la transmisión de preámbulos. Asimismo, se presenta el análisis de modelos de tráfico que describen la probabilidad de peticiones de acceso por cada segmento a aplicar en el RAP. También presenta la descripción de la metodología empleada para la implementación de la RNA y la política de aprendizaje *Q-Learning*. Se exponen las ideas fundamentales que guían la asignación dinámica de preámbulos para cada algoritmo de aprendizaje máquina propuesto. También se describen las herramientas utilizadas durante la implementación y la puesta en marcha de los algoritmos en el entorno de redes 5G.
- **Capítulo 5:** En este capítulo, se ha llevado a cabo una exhaustiva exploración de los resultados obtenidos a través de simulaciones y evaluaciones en diferentes escenarios. Se han analizado los indicadores de rendimiento o también

conocidos como KPI de las propuestas en situaciones de alta demanda y sobrecarga, así como su capacidad para abordar desafíos de bloqueo de acceso y congestión. Los resultados proporcionarán una comprensión más profunda de la eficacia y el rendimiento de la administración dinámica de preámbulos en redes 5G.

- **Capítulo 6:** Presenta las conclusiones derivadas de la investigación y el análisis realizados en el transcurso de este estudio. A lo largo de este capítulo, se resaltarán los hallazgos más significativos y se discutirán inferencias respecto a los KPI. Además, se presentarán recomendaciones prácticas basadas en las experiencias y los resultados obtenidos en este estudio, con el objetivo de orientar futuras investigaciones y aplicaciones en el ámbito de la optimización de recursos en 5G.

## Capítulo 2 Marco teórico

En este capítulo se presenta la base teórica necesaria para proporcionar el conocimiento previo requerido en este proyecto de titulación. Se empieza con la Sección 2.1 que ofrece una visión general de la red 5G, sus características principales y sus diversas aplicaciones. Además en la Sección 2.1.4 se aborda de manera específica el RAP destacando su importancia en el contexto del tema de estudio.

Posteriormente, en la Sección 2.1.5 se detalla información sobre la técnica de segmentación presentando información relevante sobre su aplicación y beneficios. También se aborda los grupos de segmentación estándar recomendados por la Union Internacional de Telecomunicaciones (UIT). Por último, en la Sección 2.2 se introduce el concepto de aprendizaje automático; en esta sección se abarca diversos modelos de aprendizaje como son: aprendizaje supervisado, no supervisado, semi-supervisado, reforzado y profundo.

### 2.1. Tecnología 5G

La tecnología inalámbrica 5G representa la última actualización de los estándares internacionales que rigen el funcionamiento de los teléfonos móviles [17]. Esta evolución tecnológica ha sido impulsada por diversos factores, como la masiva cantidad de usuarios móviles, el flujo cada vez mayor de datos, la demanda de baja latencia y los requerimientos de alta movilidad, entre otros [18]. Se prevé que 5G permitirá alcanzar velocidades de hasta 20 Gbps para comunicaciones inalámbricas de banda ancha, ofreciendo también una latencia de 1 ms o incluso menos para aplicaciones en tiempo real [14, 17].

#### 2.1.1. Características

La tecnología 5G se caracteriza por ocho atributos distintivos, según [17]:

- Busca lograr una latencia extremadamente baja de tan solo 1 ms.
- 100 % de cobertura.
- Una densidad de datos de 10 Tb/s/km<sup>2</sup>

- Los dispositivos Internet of Things (IoT) tendrán una vida útil de 10 años gracias a su bajo consumo de potencia.
- Velocidades que alcancen hasta los 20 Gbps.
- Reducción del consumo energético en la red hasta un 90 % en comparación con 4G.
- Aumento de 10 a 100 veces más dispositivos conectados en comparación con tecnologías móviles anteriores.
- 99.99 % de disponibilidad de la red.

Aunque no hay una especificación exacta de lo que abarcará 5G en [7, 17–19] mencionan que en comparación con la red 4G, las nuevas redes deben abarcar lo siguiente:

- Mayor capacidad de manejo de tráfico.
- 10 veces la eficiencia espectral y la eficiencia energética.
- 25 veces el rendimiento celular promedio.
- 10-100 veces mayor tasa de datos de usuario y mayor número de dispositivos de conexión.
- Espectro de ondas milimétricas (mmWave) 10-100 veces más barato (en comparación con el espectro por debajo de 3GHz).
- 10-100 veces celdas pequeñas más baratas que las macroceldas.
- Soporte de movilidad de 350 km/h a 500 km/h.

### 2.1.2. Arquitectura de red

Un sistema 5G se compone de los mismos elementos fundamentales que sus predecesores: un UE compuesto por un dispositivo móvil que contiene un módulo de identidad de suscriptor universal, la New generation RAN (NG-RAN) y la red central 5G Core (5GC). La Figura 2.1 ilustra esta arquitectura de red [1].

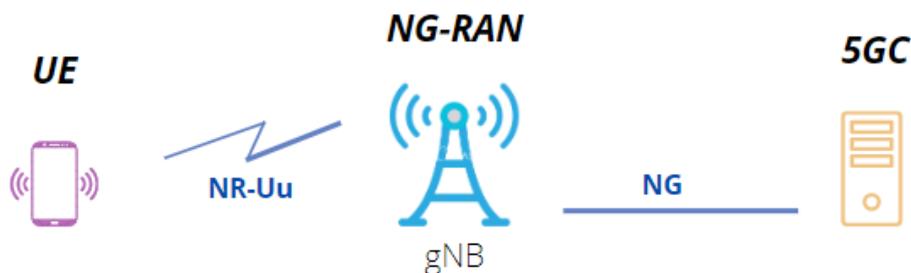


Figura 2.1: Descripción general de la arquitectura UE-NG-RAN-5GC [1].

Existen dos modelos de arquitecturas para 5G: arquitectura Non-Stand Alone (NSA) y arquitectura Stand Alone (SA). En el enfoque NSA, la red de acceso de radio 5G se despliega junto con la infraestructura preexistente de 4G LTE. Esta implementación gradual permite a los operadores móviles realizar una transición suave de las redes 4G hacia 5G, como se representa en la Figura 2.2. Por otro lado, en la arquitectura SA, se implementa una red 5G totalmente autónoma e independiente, como se muestra en la Figura 2.3 [1, 20].

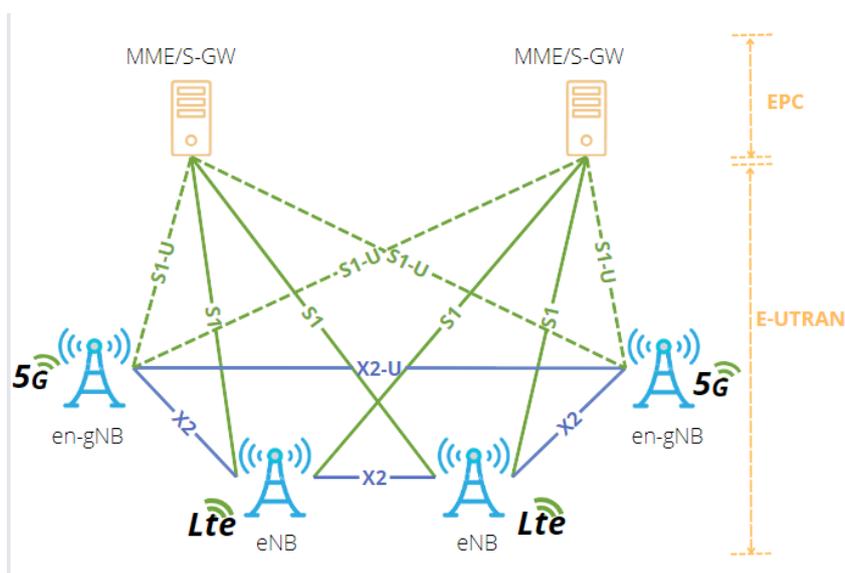


Figura 2.2: Arquitectura NSA [1].

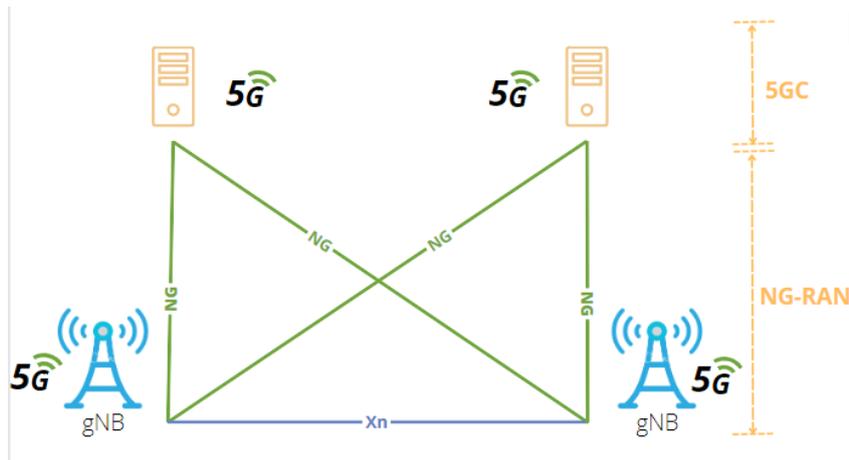


Figura 2.3: Arquitectura SA [1].

### 2.1.3. Aplicaciones

La tecnología 5G se extiende más allá de los teléfonos inteligentes, abarcando una amplia gama de dispositivos en diversos campos, como domótica, comunicación vehicular, robótica, realidad virtual y más. Esta diversificación de aplicaciones se observa en la Figura 2.4. De acuerdo con [19], las redes 4G actuales no son capaces de gestionar la inmensa cantidad de datos que se espera transmitir en el futuro, ya que se pronostica que para el año 2025 habrá más de 100 mil millones de dispositivos conectados, incluyendo sensores, vehículos, electrodomésticos, robots y otros tipos de dispositivos.



Figura 2.4: Entorno IoT en 5G [2].

Si bien muchas aplicaciones son compatibles con las redes 4G, algunas requieren las capacidades avanzadas de 5G, como la alta velocidad y baja latencia en la transmisión de datos [3]. La Figura 2.5 muestra las tasas de velocidad y latencia necesarias para algunas de las aplicaciones IoT más comunes. Por ejemplo, aplicaciones como la conducción autónoma requieren baja latencia, mientras que otras aplicaciones, como las redes de sensores para monitoreo, pueden operar con tasas de velocidad más modestas.

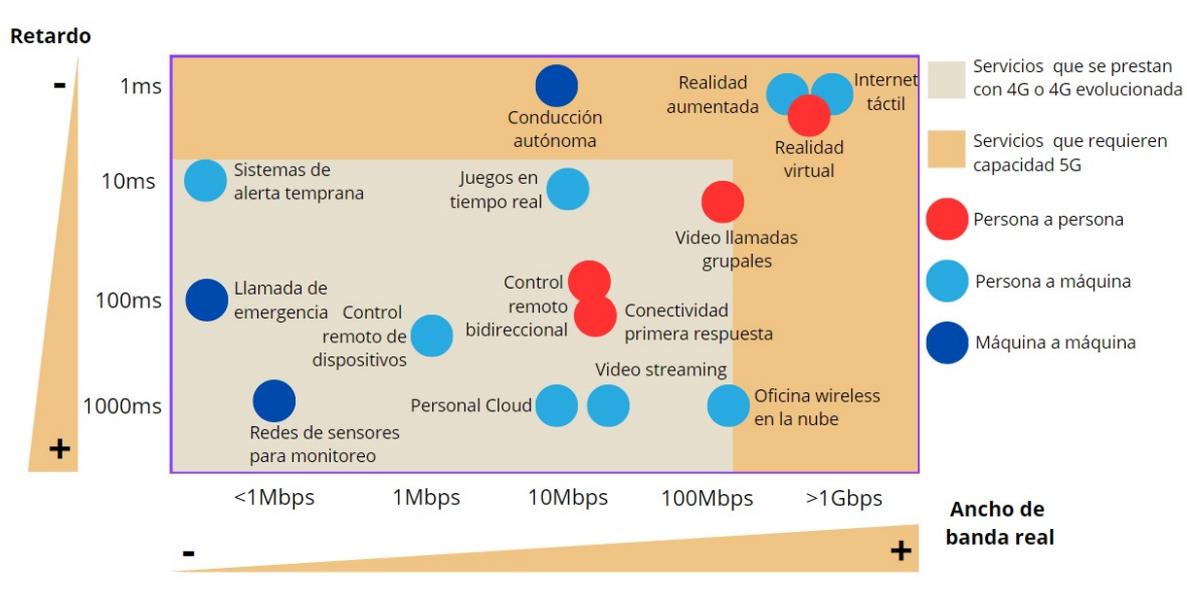


Figura 2.5: Exigencias de retardo y ancho de banda de aplicaciones IoT [3].

### 2.1.4. Control de acceso al medio

El proceso de acceso al Physical Random Access Channel (PRACH) es un paso esencial y preliminar para que los UE se conecten a un gNodeB en la red 5G [21]. A través de intercambios de mensajes, los UEs obtienen recursos de Uplink (UL) dedicados para su uso en futuras comunicaciones [22]. Dado que 5G tiene la capacidad de acomodar una enorme cantidad de dispositivos, surge un desafío cuando múltiples dispositivos intentan acceder a los recursos de la red en un intervalo de tiempo pequeño. En tales situaciones, existe la posibilidad de que varios dispositivos elijan el mismo preámbulo de acceso, lo que puede resultar en colisiones al utilizar los mismos recursos de UL [23].

#### 2.1.4.1. Procedimiento de acceso aleatorio

El RAP en una red 5G presenta similitudes con el sistema implementado en 4G LTE [24]. Se distinguen dos tipos de RAP: acceso de 4 pasos y acceso de 2 pasos. Ambos tipos admiten dos modos de operación: acceso aleatorio sin contención (*Contention Free Random Access*) y acceso aleatorio basado en contención (*Contention based Random Access*).

El modo sin contención se emplea en situaciones críticas como el traspaso (*handover*), la recepción de datos de descarga o el posicionamiento. El modo basado en contención, por otro lado, constituye el método estándar de acceso a la red. Los usuarios móviles lo utilizan para cambiar el estado de control de los recursos de radio de inactivo a conectado, recuperarse de una falla en el enlace de radio, sincronizar el enlace ascendente o enviar solicitudes de programación [25].

El RAP requiere la participación de dos canales de UL. El PRACH interactúa con los mensajes de acceso aleatorio transmitidos por los UEs, utilizados para la transmisión de preámbulos. De manera paralela, el canal de control físico de enlace ascendente o también conocido por sus siglas en inglés Physical Uplink Control Channel (PUCCH) permite la transmisión de señales de control y gestión de red. Los intentos de acceso aleatorio se efectúan empleando recursos predefinidos en términos de tiempo o frecuencia, denominados oportunidades de acceso aleatorio (RAO) [25, 26].

El propósito fundamental del PRACH es indicar una solicitud de conexión para cada UE que necesite acceder a la red celular. Para ello, el PRACH ocupa seis bloques de recursos consecutivos de transmisión en la banda UL ( $6 \times 180$  kHz) en el dominio de frecuencia, equivalente a un ancho de banda total de 1,08 MHz. En el proceso de RAP de 4 pasos, es necesario que la transmisión del preámbulo se realice a través del canal PRACH. El preámbulo se compone de tres partes: el prefijo cíclico, la secuencia y el tiempo de guarda, como se muestra en la Figura 2.6 [1, 27].

De acuerdo con los autores en [4], la separación entre subportadoras en el canal PRACH es de 1,25 kHz, mientras que en un canal UL es de 15 kHz. Por lo tanto, el número total de subportadoras en el canal PRACH y en el canal UL es de 864 y 72, respectivamente. Se reservan 13 y 12 subportadoras en los extremos superior e infe-

rior del PRACH para subportadoras de protección. Como resultado, únicamente 839 subportadoras se utilizan para la transmisión de los preámbulos [28]. La Figura 2.6 brinda una comparativa visual entre los canales PRACH y UL, permitiendo apreciar las diferencias en la separación entre subportadoras y la estructura general de cada canal.

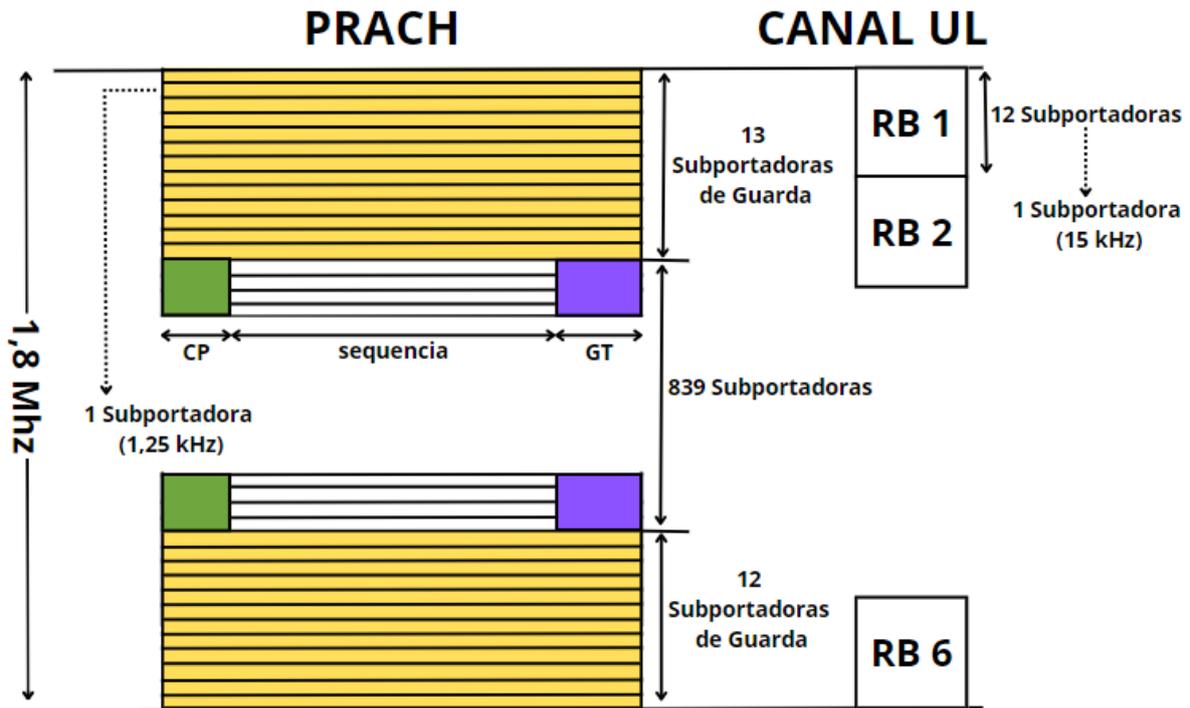


Figura 2.6: Comparación de canales PRACH y UL [4].

En el dominio del tiempo, la periodicidad de los RAO se establece mediante el parámetro `prach-ConfigIndex`, el cual es suministrado por el gNodeB [25]. Como se detalla en la Tabla A.1, este parámetro brinda la siguiente información: (i) El formato del preámbulo, el cual cuenta con cuatro posibles formatos numerados del 0 al 3; (ii) Las tramas del sistema utilizadas para transmitir el preámbulo, las cuales pueden ser enviadas en una trama de sistema con número par o en cualquier otra trama de sistema; y, (iii) La subtrama utilizada para transmitir el preámbulo, también conocida como RAO [4].

En total, existen 64 configuraciones posibles para el canal PRACH [25]. La Figura 2.7 ilustra ejemplos de las subtramas que pueden ser utilizadas para transmitir un preámbulo para seis valores posibles de `prach-ConfigIndex`. Por ejemplo, si el valor

de prach-ConfigIndex transmitido por la estación base es 12, se usará el formato de preámbulo 0 que indica la transmisión en cualquier trama del sistema y se puede usar cualquier subtrama numerada 0, 2, 4, 6 y 8 para transmitir el preámbulo [4]. De esta manera, la periodicidad de cada RAO va desde un mínimo de 1 RAO cada dos tramas hasta un máximo de 1 RAO cada subtrama, es decir, de 1 RAO cada 20 ms a 1 RAO cada 1 ms [25].

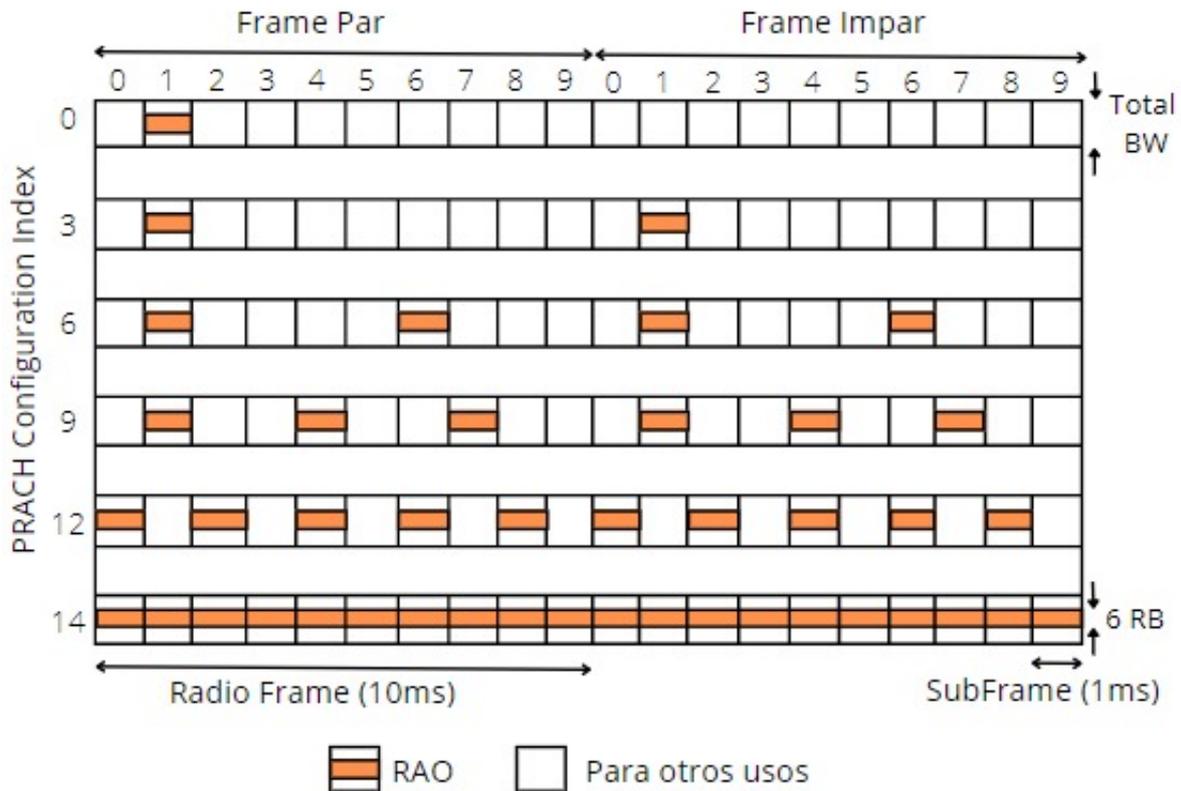


Figura 2.7: Ejemplo de prach-ConfigIndex [4].

Para llevar a cabo el acceso inicial a la red, el canal PRACH utiliza preámbulos que permiten a los dispositivos UE conectarse en la fase inicial. Cada celda pone a disposición 64 preámbulos ortogonales (*signatures*). En el modo sin contención, se evitan las colisiones mediante la asignación coordinada de preámbulos, donde el gNodeB asigna preámbulos específicos a los UE durante intervalos de tiempo específicos. Por otro lado, en el modo basado en contención, los UE eligen preámbulos de manera aleatoria, lo que incrementa el riesgo de colisiones, ya que varios UE en la misma celda podrían seleccionar la misma firma de preámbulo en la misma oportunidad de

acceso aleatorio (RAO).

Debido al número limitado de preámbulos disponibles y a medida que aumenta el número UEs en la red, la congestión de UEs en el RAP aumentará significativamente. Esto resulta en la necesidad de una resolución efectiva de conflictos para evitar colisiones y mejorar la eficiencia del acceso aleatorio [4, 25].

#### **2.1.4.2. Procedimiento de acceso aleatorio de 4 Mensajes basado en contención**

El proceso se inicia con la transmisión de señales de sincronización por parte de gNodeB hacia los dispositivos UEs. Los UEs sincronizan el tiempo y la frecuencia con el gNodeB, permitiéndoles detectar la estructura de trama y la identidad física de la celda. Tras esta sincronización, los UE escuchan el canal de difusión para obtener la información del UL [29].

La información del sistema (System Information (SI)) se subdivide en dos bloques: el bloque de información principal (Main Information Block (MIB)) y el bloque de información del sistema (System Information Block (SIB)). Existen 13 mensajes SIB diferentes, pero para realizar el RAP, los UEs necesitan recibir y procesar la transmisión del mensaje System Information Block Type 2 (SIB2) del gNodeB.

El mensaje SIB2 contiene parámetros esenciales del RAP, como el índice de secuencia raíz física utilizado para generar el preámbulo en el Msg1 del RAP de 4 pasos (rootSequenceIndex), la configuración del canal PRACH (prach-ConfigIndex), la frecuencia en la que un UE transmite el preámbulo (prach-FreqOffset), el número máximo de retransmisiones permitidas para el preámbulo en el RAP (preambleTransMax), el número de preámbulos reservados para el RAP (numberOfRA-Preambles) y el tiempo de espera necesario para que un UE espere después de transmitir el Msg3 (mac-ContentionResolutionTimer) para recopilar el Msg4 enviado por la estación base [4]. Con esta información adquirida, los UE pueden proceder con el protocolo de enlace de cuatro mensajes, como se ilustra en la Figura 2.8.

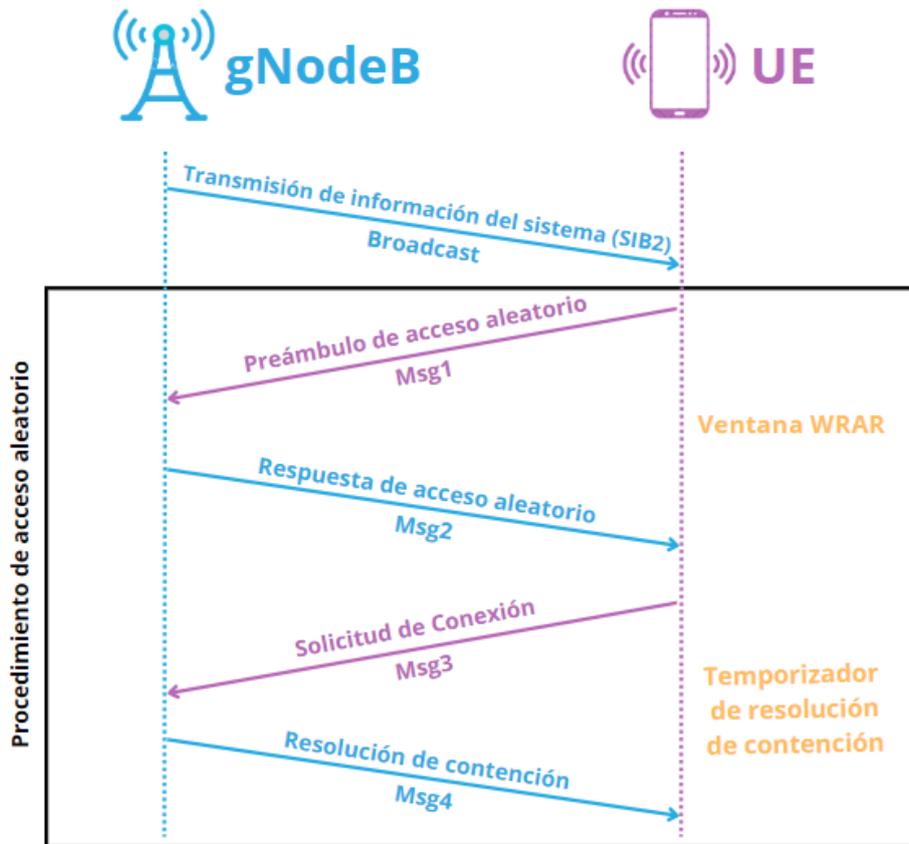


Figura 2.8: Variables que intervienen en el procedimiento de acceso aleatorio [5].

**Preámbulo de acceso aleatorio (Msg1)** El proceso de establecimiento de conexión se inicia cuando un dispositivo UE envía una secuencia conocida como preámbulo de acceso aleatorio (Msg1) a la estación base a través del canal PRACH. Utilizando la fórmula *Zadoff-Chu* (ver [9, 30]), el UE genera 64 preámbulos ortogonales completos basados en el número de subportadoras en el PRACH (839) y el número de secuencia raíz física obtenido del mensaje SIB2. Sin embargo, la estación base reserva 10 preámbulos para los UE que realizan un acceso aleatorio no basado en contención, por lo que los UE que realizan un acceso aleatorio basado en contención deben seleccionar un preámbulo al azar de los 54 preámbulos restantes y utilizarlo como su Msg1 [4, 26].

El gNodeB es capaz de decodificar correctamente un preámbulo transmitido por un único UE si la potencia de la señal es suficiente, permitiendo estimar el tiempo de transmisión del terminal. Sin embargo, si dos o más UEs transmiten el mismo preámbulo, se pueden presentar dos escenarios: en el primero, el gNo-

deB no puede decodificar el preámbulo transmitido y se produce una colisión en la transmisión de  $Msg1$  (consultar la Figura 2.9). En el segundo escenario, los preámbulos transmitidos son decodificados correctamente por el gNodeB. Esto ocurre porque la potencia recibida de uno de los preámbulos puede ser considerablemente mayor que la de los demás, lo que provoca que las señales parezcan fusionarse en una única transmisión que atraviesa diversos caminos de desvanecimiento. Además, si el UE está muy alejado del gNodeB y su potencia de transmisión es insuficiente, la transmisión del preámbulo podría fallar [25].

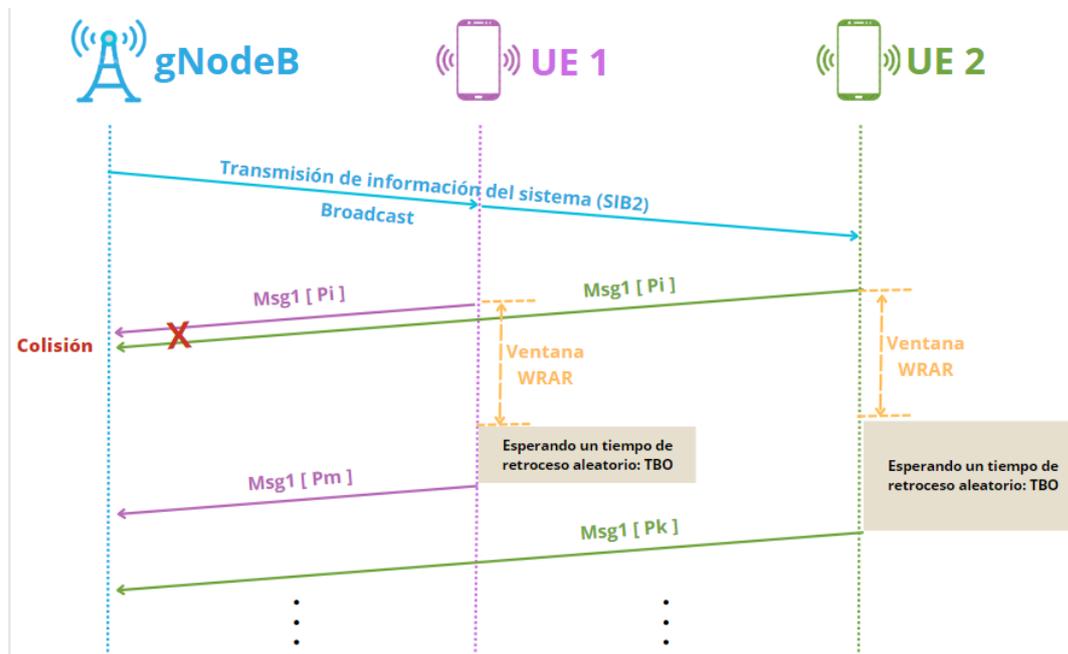


Figura 2.9: Colisión en la transmisión de  $Msg1$  [5].

**Respuesta de acceso aleatorio ( $Msg2$ )** Después de que el gNodeB ha decodificado exitosamente un preámbulo, asigna un identificador único a cada uno de estos preámbulos. Luego, a través del canal físico compartido de descarga o conocido por sus siglas en inglés Physical Downlink Shared Channel (PDSCH), envía una Respuesta de Acceso Aleatorio (RAR) al UE. El RAR, o también llamado  $Msg2$  contiene varios datos, incluyendo información sobre la identificación del preámbulo detectado, la Alineación Temporal (TA), las concesiones de enlace ascendente (recursos reservados del canal físico compartido de enlace ascendente uplink shared Physical channel (PUSCH)) para la transmisión de  $Msg3$ , el Indicador de Retroceso (BI) y la asignación de un identificador temporal [25].

Una vez completada la transmisión del preámbulo (lo cual lleva aproximadamente dos subtramas para que el gNodeB procese los preámbulos recibidos), el UE espera durante una ventana de tiempo (Window Random Access Response (WRAR)) para recibir una concesión de enlace ascendente (UL) del gNodeB. Si el RAR no se recibe dentro de esta ventana de tiempo configurada WRAR, implica que el preámbulo no fue decodificado correctamente. En tal caso, el UE retrocede al Msg1, retransmite el preámbulo con una potencia más alta y aumenta el número de intentos en la transmisión del preámbulo [26].

En cada subtrama, puede haber un máximo de un mensaje RAR, pero este puede contener múltiples concesiones UL, cada una de ellas asociada a un preámbulo que ha sido decodificado exitosamente. Dado que los recursos del canal de control de descarga físico (Physical Downlink Control Channel (PDCCH)) son limitados, se asume que el número máximo de concesiones de UL por RAR es constante, ya que el Msg2 (y Msg4) poseen la máxima prioridad dentro del PDCCH. La longitud de la ventana WRAR (en subtramas) se comunica desde el gNodeB al UE mediante el mensaje SIB2. Como resultado, existe un número máximo de concesiones de UL que se pueden transmitir dentro de la ventana WRAR. Solo aquellos UE que reciben una concesión de UL pueden transmitir el mensaje Msg3. Si el gNodeB no logra decodificar los preámbulos transmitidos por varios UE, estos UE no recibirán concesiones de UL y serán considerados como intentos fallidos en el proceso [25]. Ni la estación base ni los UE son conscientes de las colisiones hasta la transmisión del Msg3 [22].

**Solicitud de conexión (Msg3)** Luego de recibir el RAR correspondiente, el UE ajusta su tiempo de transmisión de UL de acuerdo con el TA proporcionada. A continuación, utiliza los recursos reservados del canal PUSCH para transmitir un mensaje programado denominado Msg3 hacia el gNodeB. Con el objetivo de proteger la transmisión del mensaje, se implementa la solicitud de repetición automática híbrida (Hybrid Automatic Repeat Request (HARQ)). El número de intentos de transmisión es limitado y la probabilidad de retransmisión para Msg3 se establece en 0,1. Los UEs que enviaron preámbulos idénticos se les otorgarán los mismos recursos para la transmisión de Msg3 [22]. Es esencial recordar

que si el gNodeB decodifica correctamente los preámbulos transmitidos por varios UEs, estos UEs transmitirán sus mensajes Msg3 en los mismos recursos físicos, lo que dará lugar a una colisión. En consecuencia, el gNodeB no podrá decodificar los mensajes transmitidos (Ver Figura 2.10).

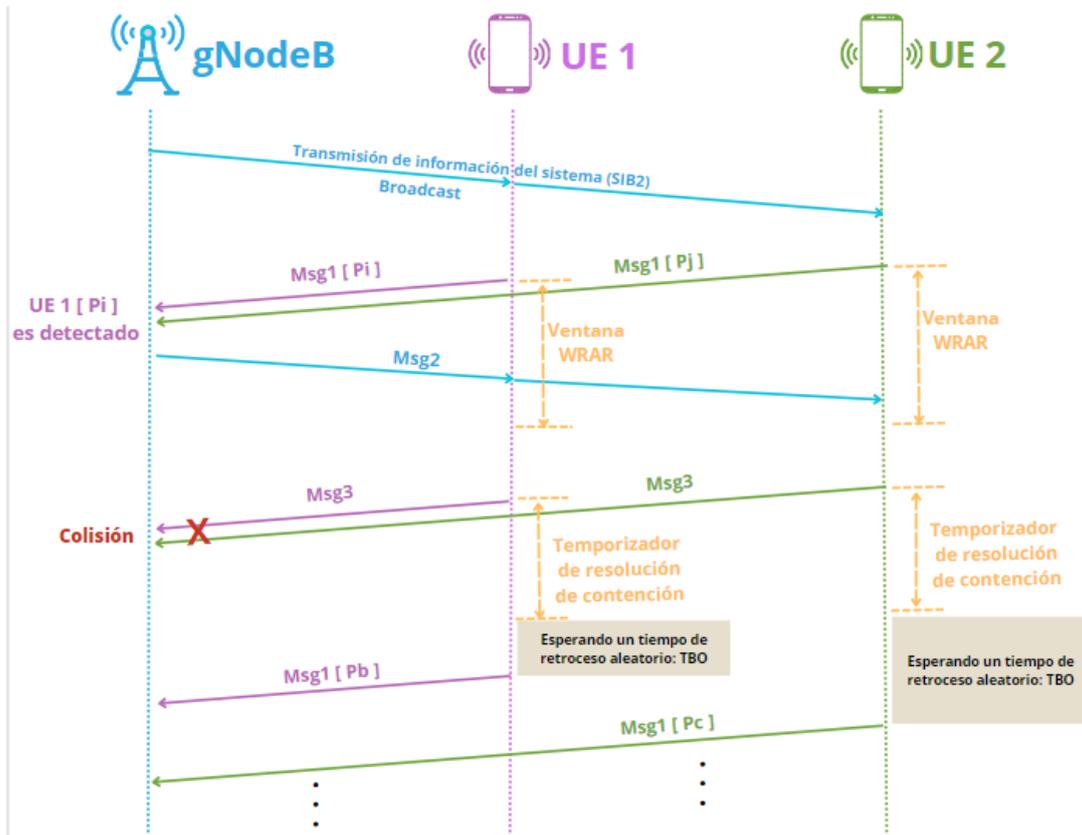


Figura 2.10: Colisión en la transmisión de Msg3 [5].

**Resolución de contención (Msg4)** El gNodeB responde al Msg3 con el Msg4, utilizando un proceso HARQ para transmitirlo a los UE. Similar a la dinámica de Msg3, el número de intentos de transmisión para Msg4 está limitado y se establece una probabilidad de retransmisión del 0,1. Si un UE no recibe el Msg4 antes de que expire el temporizador de resolución de contienda, el UE considera que la resolución de contención ha fallado y programa un nuevo intento de acceso en consecuencia.

En caso de que ocurra un fallo de acceso en cualquiera de los pasos anteriores debido a una potencia de transmisión insuficiente, una colisión o la expiración del temporizador de resolución de contienda, los UEs afectados aumentan su potencia de transmisión y transmiten un nuevo preámbulo seleccionado al azar

en un nuevo intervalo de RAO. Este proceso implementa un esquema de retroceso uniforme utilizando el indicador de retroceso (BI). Cada UE realiza un seguimiento de sus transmisiones de preámbulo y, cuando un UE ha transmitido un número predefinido de preámbulos sin éxito, el UE determina que la red no está disponible. En este caso, el UE informa de un problema de acceso aleatorio a las capas superiores y concluye el procedimiento de acceso aleatorio [25].

#### 2.1.4.3. Procedimiento de retroceso

En caso de que el intento de acceso aleatorio de un UE falle, independientemente de la causa, el UE debe reiniciar el procedimiento de acceso aleatorio. Para lograr esto, el UE debe realizar primero un proceso de retroceso o también llamado *backoff*. Durante este proceso, el UE espera un intervalo de tiempo aleatorio,  $T_{BO}$  [ms], antes de realizar un nuevo intento de transmisión de preámbulo. El tiempo de retroceso se calcula mediante la Ecuación (2.1).

$$T_{BO} = U(0, BI) \quad (2.1)$$

Donde  $U(\cdot)$  representa distribución uniforme, y BI puede oscilar entre 0 y 960 ms. El valor de BI se envía en el mensaje (Msg2) que es recibido por todos los UEs que hayan enviado un preámbulo RACH en el RAO anterior. Por lo tanto, cada UE que no haya recibido un Msg2 debido a un intento fallido, recibirá el valor de BI [25].

#### 2.1.5. Técnica de segmentación de red

La red 5G presenta una serie de desafíos que van desde la gestión eficiente de los recursos hasta la seguridad de la información y el procesamiento de datos masivos [31]. La técnica de segmentación se ha desarrollado para satisfacer la necesidad de solventar los desafíos mediante la segmentación de los recursos en función de las diferentes capacidades que exigen los UE.

Un segmento de red se refiere a una partición lógica dentro de una red que ofrece características y capacidades de red específicas. Cada segmento puede ser diseñado

para atender las solicitudes de usuarios de un tipo particular, brindando una calidad de servicio adaptada a las necesidades del segmento [32]. Esta capacidad de crear redes personalizadas hace que la segmentación de red sea una solución flexible para diversos escenarios del mercado, en términos de funcionalidades, rendimiento y asignación de recursos.

Cada segmento puede proporcionar todas las funcionalidades de una red completa, incluyendo acceso a través de la red de radio y funciones centrales de la red. Por lo tanto, la segmentación de red brinda a los operadores la posibilidad de adaptar su infraestructura a diferentes aplicaciones y clientes. En una única red, es posible incorporar uno o varios segmentos de red [16].

Además de su capacidad para adaptar la red a diferentes usos, la segmentación de red también aporta valor a segmentos verticales, proveedores de aplicaciones y otros actores que carecen de infraestructura de red propia. Esta técnica les brinda acceso a recursos de radio, redes y servicios en la nube, lo que permite la operación de una red ajustada y la diferenciación efectiva de servicios [32].

Siguiendo las recomendaciones de la UIT, se ha establecido un consenso en torno a tres escenarios de aplicación genéricos impulsados por las tecnologías clave de los sistemas inalámbricos 5G: emBB, mMTC y URLLC [7]. La Figura 2.11 representa estos tres grupos de segmentación de manera visual, donde cada grupo ocupa un sector con características específicas para satisfacer los requisitos de los diferentes tipos de UEs.

A continuación, se describen los tres grupos de segmentación:

- **Comunicación emBB:** El objetivo de emBB es atender la creciente necesidad de la población por un estilo de vida cada vez más digital y se concentra en servicios que requieren un ancho de banda elevado, como la reproducción de videos en alta definición, la experiencia de realidad virtual y la realidad aumentada [7]. El escenario de uso de emBB incluye el acceso a banda ancha en áreas densas, acceso a banda ancha en cualquier lugar, mayor movilidad del usuario y servicios similares a la transmisión para servicios que necesitan gran cantidad de datos [4, 33].

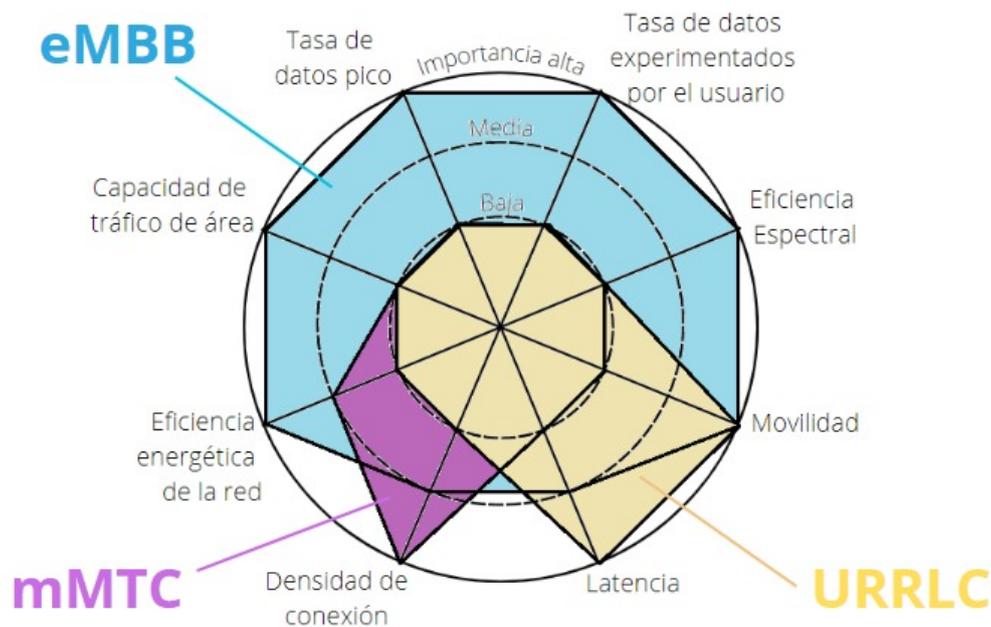


Figura 2.11: Grupos de segmentación [6].

- **Comunicación URLLC:** Este segmento tiene como finalidad abordar las demandas de la industria digital, centrándose en servicios altamente sensibles a la latencia, como la conducción automatizada y asistida, así como la gestión remota [7]. Se compone de comunicaciones en tiempo real extremas, comunicaciones vitales y comunicaciones ultra confiables, todas diseñadas para aplicaciones que requieren una respuesta inmediata y confiable, especialmente en situaciones críticas [32, 33].
- **Comunicación mMTC:** El objetivo de mMTC es atender servicios que requieren una alta densidad de conexión, como las ciudades inteligentes y la agricultura inteligente [7]. Este tipo de comunicación representa a los dispositivos IoT y se caracteriza por la conectividad masiva de una gran cantidad de dispositivos, desempeñando un papel fundamental en el despliegue de soluciones para IoT [34].

La Figura 2.12 ilustra cómo diferentes servicios de 5G se inclinan hacia sus respectivos grupos de segmentación.

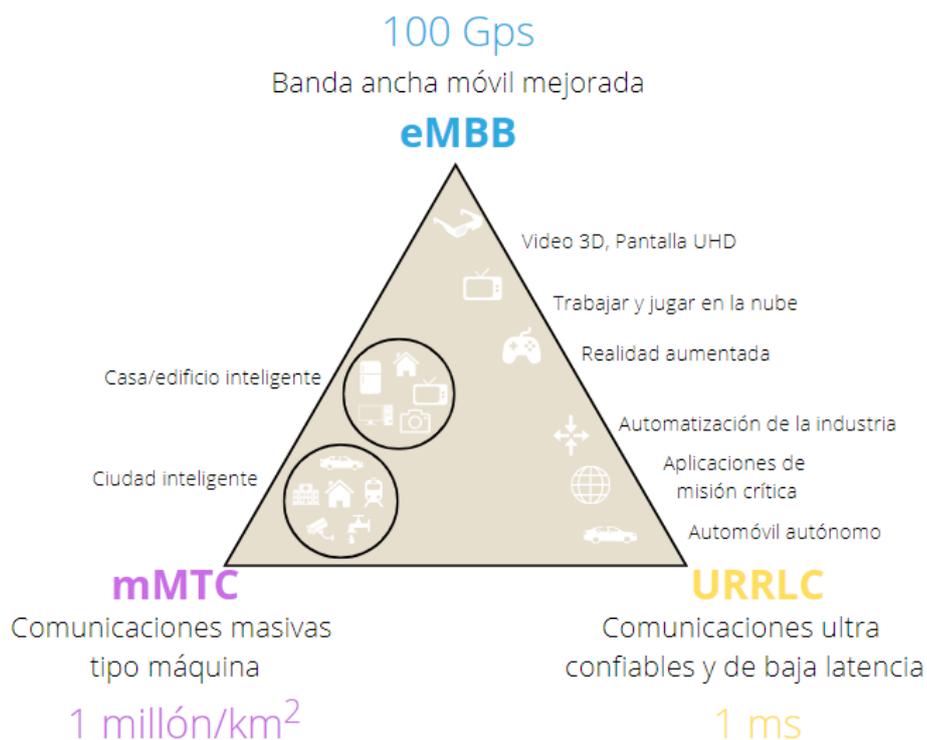


Figura 2.12: Servicios y segmentos de 5G [7].

## 2.2. Aprendizaje automático

La capacidad de una red 5G ha aumentado masivamente. Esta expansión ha llevado a un aumento exponencial en la cantidad de UEs que solicitan acceso, lo que ha superado la capacidad de análisis que un humano podría realizar de manera eficiente [35]. En este contexto, las técnicas de ML juegan un papel fundamental al permitir la automatización de las estrategias de asignación de recursos. Estas técnicas aprovechan su capacidad para detectar patrones, realizar predicciones y prescribir soluciones proactivas [14].

ML es una tecnología informática que brinda a los sistemas informáticos la capacidad de aprender con datos y mejorar el rendimiento y la precisión de la toma de decisiones en una tarea específica, sin estar programados explícitamente [16]. El campo de ML se ocupa de cómo construir algoritmos que mejoren de forma autónoma con la experiencia. Desde una perspectiva informática, la experiencia se materializa en la información producida por los procesos de almacenamiento de datos [35].

Existen varias técnicas de ML, según el tipo de información (información estructurada o no estructurada) y el paradigma de aprendizaje utilizado. La elección de la técnica a aplicar dependerá, entre otras cosas, del objetivo del modelo a construir, así como del tipo de información disponible [35]. Algunas de las técnicas más populares son: aprendizaje supervisado, no supervisado, semi-supervisado, reforzado y profundo.

### 2.2.1. Aprendizaje supervisado

El objetivo del aprendizaje supervisado es entrenar una aplicación que vincula un conjunto de variables  $x$  (llamadas características o atributos), y una variable de salida  $y$ , utilizando un conjunto de datos conocido como conjunto de entrenamiento. Este conjunto de entrenamiento contiene pares de datos  $(x_i, y_i)$ , donde  $i$  varía de 1 a  $N$ , y  $N$  representa el tamaño de la muestra. Si la variable de salida  $y$  es continua, es un problema de regresión, mientras que si  $y$  es discreta o categórica, es un problema de clasificación [35].

El aprendizaje supervisado se centra en construir un modelo que, al recibir nuevas instancias de  $x$ , pueda predecir o estimar de manera precisa los valores correspondientes de  $y$ . Esto implica que el modelo generalice patrones y relaciones identificados en el conjunto de entrenamiento para hacer predicciones en datos no vistos previamente [35].

#### 2.2.1.1. Técnicas de aprendizaje supervisado

Existen varias técnicas dentro del aprendizaje supervisado, cada una diseñada para abordar diferentes tipos de problemas. Algunas de las técnicas más comunes son:

- **Regresión:** En este enfoque, se proporcionan ejemplos concretos. Se establece una relación entre las características de entrada  $x$  y la variable de salida  $y$ . La máquina aprende a encontrar una relación empírica entre los valores de  $x$  y  $y$  a partir de los datos de entrenamiento. Una vez que se entrena lo suficiente, se le pide a la máquina que haga predicciones para nuevas entradas [36].
  - **Clasificación Binaria:** Dos posibles salidas. Comúnmente se usa 0 y 1.

- **Clasificación Multiclase:** Este tipo de clasificación refleja un modelo más complejo, hay varias salidas posibles.
- **Clasificación:** En este modelo, el objetivo es asignar objetos a categorías o grupos predefinidos en función de sus características. Por ejemplo, si se considera un conjunto de 100 estudiantes. Mediante la medición de la altura de cada estudiante, podríamos clasificarlos en tres grupos según su tamaño: pequeño, mediano y grande. Al observar cómo se agrupan los estudiantes en función de su altura, la máquina aprende patrones y características que definen cada grupo. Una vez que la máquina ha adquirido este conocimiento durante el proceso de entrenamiento, puede aplicarlo para clasificar de manera precisa a nuevos estudiantes según su altura en los grupos correspondientes. Esto significa que la máquina podrá asignar adecuadamente etiquetas de clase a atributos desconocidos en el futuro, utilizando los patrones aprendidos previamente [36].

#### 2.2.1.2. Modelos de aprendizaje supervisado

Dentro del aprendizaje supervisado, existen varios modelos que se utilizan para realizar tareas de clasificación y regresión. Algunos de estos modelos incluyen:

- **k vecinos más cercanos:** Clasifica los datos en función de los vecinos. El valor de recorrer espacios entre vecinos puede ser calculado utilizando métricas de distancia como euclidiana, chebyshev, etc [37].
- **Naïve Bayes (NB):** Asigna una distribución de probabilidad a cada clase de respuesta en el conjunto de entrenamiento. Se obtiene dos probabilidades; la probabilidad de una clase de respuesta pico falsa dada su posición y la probabilidad de una clase de respuesta pico original dada su posición [37].
- **Decision Tree Classifier (DTC):** Tiene una gran capacidad para dividir un problema complejo en pequeñas ramificaciones simples y fácil de interpretar [38].
- **Redes neuronales:** Una RNA es un modelo matemático y computacional que se inspira en el funcionamiento del sistema nervioso biológico, especialmente en la estructura y comportamiento de las neuronas. Esta compuesto por un gran número de elementos organizados en diferentes niveles jerárquicos [39]. Cada

neurona artificial toma entradas, realiza operaciones matemáticas y produce una salida. Para un análisis más exhaustivo de este modelo, se encuentra disponible información detallada en la Sección 2.2.1.3.

### 2.2.1.3. Redes neuronales y su aplicación en el aprendizaje supervisado

Una RNA consiste en unidades de procesamiento conectadas de manera densa, llamadas neuronas, por tener un comportamiento similar al de las neuronas biológicas, de forma analógica, una RNA busca aprender experiencia a través de la experiencia de la misma forma que el cerebro [38, 39]. Los componentes principales del funcionamiento de una RNA son:

- **Unidades de procesamiento (Neuronas):** Cada neurona recibe entradas ponderadas, realiza cálculos matemáticos y produce una salida. Las neuronas también pueden tener una función de activación que determina si la salida se activa o no.
- **Función de activación:** La función activación calcula el estado de actividad de una neurona; transformando la entrada en un valor de activación (normalmente binaria). Las funciones de activación más comunes se detallan a continuación:

- *Función Identidad*

$$y = x \quad (2.2)$$

- *Función Escalón:*

$$y = \text{sign}(x) \quad (2.3)$$

- *Función Lineal:*

$$f(x) = \begin{cases} -1 & x \leq -1/a \\ a \cdot x & -1/a < x < 1/a \\ 1 & x \geq 1/a \end{cases} \quad (2.4)$$

- *Función sigmoidea:*

$$f(x) = \frac{1}{1 + e^{-gx}} \quad (2.5)$$

- *Función Tangente hiperbólica*

$$f(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}} \quad (2.6)$$

- **Valor de activación:** Su rango va de 0 a 1 o de -1 a 1 debido a que una neurona puede encontrarse totalmente activa o inactiva.
- **Función de salida:** Es la operación matemática que se aplica en la capa final de la red para producir el resultado o la predicción final [39].
- **Capas:** Las neuronas se estructuran en múltiples capas, de tal manera que las neuronas en una capa están interconectadas con las neuronas en la capa subsiguiente, a través de las cuales pueden transmitir información como se observa en la Figura 2.13.
  - *Capa de entrada:* Recibe información procedente del entorno.
  - *Capas ocultas:* Realiza los cálculos y procesamientos en la red.
  - *Capa de salida:* Proporciona los resultados finales de las operaciones de la red hacia el exterior.

r

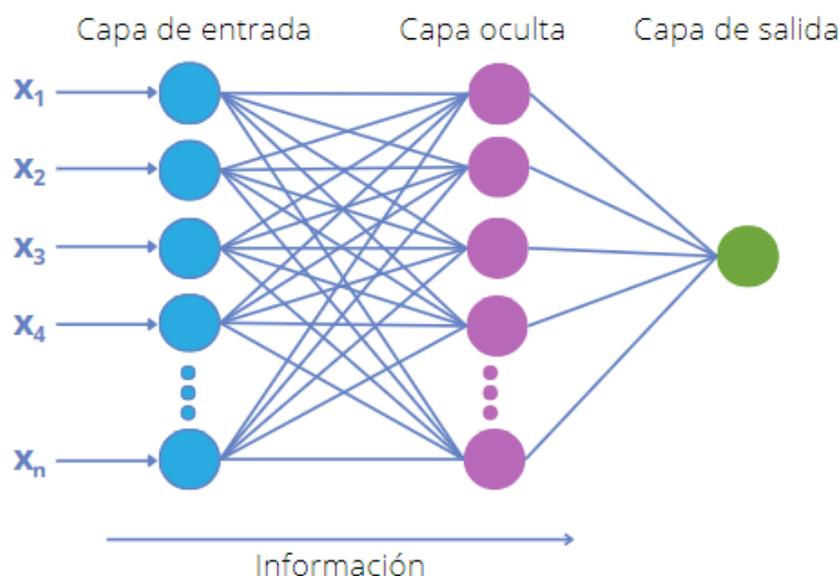


Figura 2.13: Modelo de una red neuronal artificial [8]

### 2.2.2. Aprendizaje no supervisado

A diferencia del aprendizaje supervisado, no se tiene una muestra de información inicial que funcione de base de entrenamiento o aprendizaje. El objetivo de este tipo de modelo es encontrar patrones o relaciones en los datos. Por ello, también se conoce como *knowledge discovery* (proceso de identificar patrones de datos que sean válidos, nuevos, potencialmente útiles y comprensibles) [35]. El aprendizaje no supervisado parece mucho más difícil porque el objetivo es que la computadora aprenda a hacer algo que no le decimos cómo hacer.

### 2.2.3. Aprendizaje semi-supervisado

Con el avance del estudio de las técnicas de aprendizaje supervisado, se comenzó a investigar un tercer tipo, conocido como aprendizaje semi-supervisado. Donde las observaciones son en forma de pares de entrada-salida pero los valores de salida no se conocen en una gran cantidad de observaciones. Buscan mejorar el modelado de conceptos en el aprendizaje supervisado [16]. En el aprendizaje semi-supervisado, se combinan los datos etiquetados con aquellos que no lo están, lo que permite a los algoritmos de ML mejorar su rendimiento al procesar los datos sin etiquetar. Esto se logra al utilizar los datos etiquetados para guiar al algoritmo en la tarea de procesamiento de los datos sin etiquetar.

El aprendizaje semi-supervisado se ubica entre el aprendizaje supervisado y el aprendizaje no supervisado, proporcionando características de ambos tipos de aprendizaje. En este tipo de técnica se trabaja con conjuntos de datos que proporciona una información adicional sobre la variable objetivo, como es el caso del aprendizaje supervisado, pero no necesariamente para todas las instancias.

La importancia del estudio de este tipo de ML, se debe a que generalmente es fácil obtener una base de datos donde no todas las instancias tienen etiqueta. En realidad, es poco frecuente encontrar una base de datos que cuente con todas las muestras etiquetadas, debido a que el proceso de etiquetado puede resultar costoso o presentar desafíos técnicos. A pesar de que se podría optar por no tomar en consideración en el estudio a los ejemplos no etiquetados, estos pueden aportar información útil para

mejorar el algoritmo de aprendizaje de máquina a emplear [40].

#### 2.2.4. Aprendizaje reforzado

El aprendizaje por refuerzo (Reinforcement Learning (RL)) ha demostrado ser una herramienta viable cuando se abordan problemas de toma de decisiones dinámicas en tiempo real, algo que es fundamental para las redes 5G variantes en el tiempo [41]. El aprendizaje por refuerzo busca maximizar la recompensa y minimizar el riesgo mediante la utilización de las observaciones recopiladas durante la interacción con el entorno, lo que significa que modifica o adquiere nuevos comportamientos y habilidades de forma incremental y sin requerir un conocimiento o control completo del entorno [31].

El aprendizaje de refuerzo se compone de los siguientes componentes principales: agente, estado, acción, recompensa y entorno. El agente realiza una acción en un estado particular del entorno y luego observa la respuesta correspondiente del mismo. El agente puede recibir una recompensa positiva o una penalización, dependiendo de si tomó buenas o malas medidas. A través de la exploración de diferentes pares de estados y acciones por medio de una búsqueda de prueba y error, el agente establece una política óptima para tomar decisiones futuras. El objetivo final del aprendizaje de refuerzo es que el agente tome una serie de acciones que maximicen su recompensa a largo plazo, basándose en la experiencia pasada, es decir, el refuerzo recibido [42].

Considere entrenar a un perro para que traiga una pelota. Lanzamos la pelota a cierta distancia y le pedimos al perro que nos la devuelva. Cada vez que el perro lo hace bien, lo premiamos. Lentamente, el perro aprende que hacer bien el trabajo le otorga una recompensa, y luego el perro comienza a hacer bien el trabajo cada vez en el futuro. Exactamente, este concepto se aplica en el tipo de aprendizaje reforzado. Todo el proceso de retroalimentación se puede representar en el siguiente diagrama de la Figura 2.14.

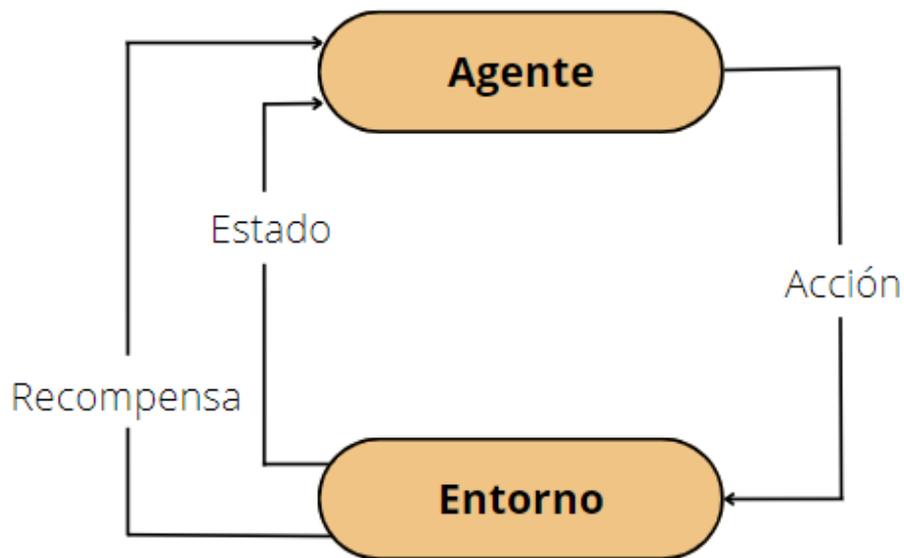


Figura 2.14: Diagrama de aprendizaje reforzado.

Esta técnica de ML se diferencia del aprendizaje supervisado en que no es necesario proporcionar los pares de entrada/salida etiquetados. Sino que la atención se centra en encontrar el equilibrio entre la exploración de nuevas soluciones y la explotación de las soluciones aprendidas [36]. RL se presenta como una técnica prometedora para abordar diversos problemas de optimización en los sistemas de comunicaciones móviles que se caracterizan por la variabilidad temporal y la aleatoriedad del servicio y la disponibilidad de recursos, así como de los parámetros y estados del sistema, como se menciona en [42].

A menudo no se dispone del modelo del sistema. Sin embargo, esto no impide que se derive una política óptima mediante el uso de *Q-learning* [43]. Este algoritmo es uno de los más conocidos entre los algoritmos de RL sin modelo y permite calcular una política óptima que maximiza la recompensa a largo plazo. El algoritmo utiliza una función de recompensa,  $Q$ , para asignar una recompensa acumulada esperada (valor  $Q$ ) a cada par estado-acción, permitiendo que el agente estime y determine las acciones óptimas en respuesta a diferentes estados del sistema. Al registrar todas las acciones que maximizan el valor  $Q$ , el agente obtiene una lista de pares estado-acción óptimos, que se define como la política óptima [42].

De esta forma, se emplea el algoritmo *Q-learning* para aprender la política que deter-

mina qué acción (es decir, la asignación de recursos) tomar en un estado dado (que viene determinado por los recursos disponibles y las solicitudes) para el gNodeB. El gNodeB utiliza *Q-learning* para calcular la función  $Q: S \times A \rightarrow R$  (también conocida como la Tabla  $Q$ ) para evaluar la calidad de la acción  $A$  que produce la recompensa  $R$  en el estado  $S$ . En cada periodo de decisión, el gNodeB selecciona una acción  $a$ , observa una recompensa  $r$  y realiza la transición del estado actual  $s$  a un nuevo estado  $s'$  (esta transición depende del estado actual  $s$  y de la acción  $a$ ), y luego actualiza  $Q$ . *Q-learning* realiza la actualización de la iteración del valor para  $Q$  en función de la Ecuación 2.7.

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot \left( r_t + \gamma \cdot \max_a (s', a) - Q(s, a) \right) \quad (2.7)$$

Donde  $\alpha$  representa la tasa de aprendizaje que puede variar entre 0 y 1 ( $0 < \alpha < 1$ ), siendo 0 la ausencia de actualización y 1 un aprendizaje rápido, mientras que  $\gamma$  es el factor de descuento que puede variar entre 0 y 1 ( $0 < \gamma < 1$ ), siendo 0 la consideración únicamente de las recompensas inmediatas y 1 la lucha por obtener una recompensa alta a largo plazo. El factor de descuento es utilizado para controlar la importancia que se le da a las recompensas futuras en comparación con las recompensas inmediatas [44–46].

### 2.2.5. Aprendizaje profundo

El aprendizaje profundo es un modelo basado en redes neuronales artificiales, más específicamente redes neuronales convolucionales. Hay varias arquitecturas utilizadas en el aprendizaje profundo, como las redes neuronales profundas, las redes de creencias profundas, las redes neuronales recurrentes y las redes neuronales convolucionales.

### Capítulo 3 Trabajos relacionados

La creación de segmentos de red tiene como objetivo facilitar la asignación eficiente de recursos en función de las demandas específicas de cada segmento. Esta estrategia permite que cada segmento ofrezca servicios con requisitos de red particulares, heterogéneos y dinámicos, los cuales pueden escalarse tanto hacia arriba como hacia abajo utilizando una infraestructura física compartida [47, 48]. La técnica de segmentación ha ganado solidez gracias a una serie de investigaciones cuyo propósito es enriquecer y abordar avances que fortalezcan la optimización de recursos y el cumplimiento de los parámetros de Calidad de Servicio (QoS).

Este capítulo explora trabajos relacionados con la técnica de segmentación y su automatización en el contexto de las redes 5G. En la Sección 3.1, se presenta una descripción de trabajos relacionados con la Virtualización de Funciones de Red (NFV) y las Redes Definidas por Software (SDN), dos pilares fundamentales para la implementación de una segmentación automatizada en las redes. Posteriormente, la Sección 3.2 aborda cómo se utiliza la técnica de segmentación en función de los requisitos específicos que una red puede implementar. Adicionalmente, en la Sección 3.3 se analiza la implementación de técnicas de ML, centrándose en la orquestación de recursos. En particular, se exploran en detalle los modelos de aprendizaje supervisado y reforzado, y cómo estos pueden contribuir a la eficiencia de la segmentación. Finalmente, en la Sección 3.4 se aborda diferentes políticas aplicadas a la asignación de preámbulos en redes segmentadas.

#### 3.1. Antecedentes a la segmentación

El diseño de una red 5G se enfoca en la capacidad de construir segmentos de red en función de la demanda y con recursos suficientes mediante el uso de tecnologías habilitantes como NFV y SDN [14, 49]. Aunque las nociones de SDN y NFV comparten muchos elementos comunes, son tecnologías distintas que pueden trabajar en conjunto [50].

SDN crea un plano de control virtualizado que puede aplicar decisiones de gestión inteligentes para las funciones de la red, cerrando la brecha entre el aprovisionamiento

to de servicios y la gestión de la red [51]. Al separar el plano de datos y el plano de control, se habilita la personalización de segmentos de red. Por otro lado, NFV cobra importancia en el plano de datos al proporcionar mecanismos de reenvío de datos y funciones de red adaptados al servicio, con la capacidad de habilitar funciones adicionales mediante inteligencia artificial para lograr mayor flexibilidad y aprovisionamiento [51]. Ambos enfoques brindan funciones de gestión de recursos y control programable que resultan esenciales para implementar segmentos de red dinámicos en 5G [47]. La combinación de SDN y NFV en una arquitectura permite simplificar el despliegue de funciones de red y aplicaciones, además de reducir los costos de gestión [52].

La computación perimetral de acceso o también conocida como Multi-Access Edge Computing (MEC) se beneficia de la incorporación de tecnologías como SDN, NFV, el encadenamiento de funciones de servicio (Service Function Chaining (SFC)) y la segmentación de red. Este conjunto crea un ecosistema capaz de ofrecer acceso en tiempo real, baja latencia, gran ancho de banda e incorporación de IoT en la 5G [53]. La combinación de MEC y segmentación abre nuevos horizontes para aplicaciones críticas de IoT, que se denominan Massive IoT (MIoT) [53]. Los requisitos clave de MIoT incluyen el análisis en el borde de la red, la reducción de los costos de comunicación y la escalabilidad de la infraestructura. En el estudio [54], se emplearon estas tecnologías para administrar los recursos de un servidor de acceso múltiple incorporado en redes vehiculares (Vehicular Ad-Hoc Network (VANET)). Los resultados demostraron una reducción de la latencia, un procesamiento más rápido de las solicitudes y un fortalecimiento de las capacidades de computación en la nube.

### **3.2. Segmentación de red en función de requisitos**

El constante avance tecnológico ha impulsado la búsqueda de soluciones que mejoren y profundicen las capacidades de la técnica de segmentación, teniendo en cuenta factores como velocidad, latencia y conectividad [55]. Esta evolución considera los desafíos impredecibles de los canales inalámbricos y la heterogeneidad inherente a las redes [56]. En un escenario en el que se atiende a conjuntos de dispositivos que requieren diferentes niveles de servicio, como eMBB, mMTC y URLLC, es crucial estudiar cómo se asignan los recursos de manera heterogénea. Por ejemplo, para

abordar esta heterogeneidad, se ha propuesto un enfoque llamado Heterogeneous Non-Orthogonal Multiple Acces (H-NOMA) [57]. Este método permite una utilización más eficiente de los recursos al evitar la inactividad de canales de frecuencia debido a dispositivos URLLC o mMTC, lo que resulta en una mayor eficiencia espectral al permitir que los canales inactivos se utilicen conjuntamente para emBB.

El enfoque *New Radio Flexibility (NRflex)* [58] logra asignar dinámicamente porciones de ancho de banda a cada segmento de red, con el objetivo de cumplir con los parámetros QoS requeridos. Esto permite que los UEs se beneficien de aplicaciones de multiservicio. Los resultados de NRflex, obtenidos a través de simulaciones en *Matlab*, demuestran una reducción en el uso del ancho de banda en comparación con el algoritmo de programación proporcional equitativo (*Fair proportional scheduling algorithm*) para segmentos URLLC. Además, se reduce la cantidad de bloques físicos de recursos en comparación con el algoritmo propuesto por los autores en [59] para segmentos emBB.

Los grupos de segmentación en una red pueden variar según los requisitos que se deseen implementar. Por ejemplo, en [47], se considera una red con dos clases de servicios: una para usuarios de alta prioridad, con baja latencia y altos ingresos; y otra para servicios de baja prioridad. En [56], se implementa un proveedor de servicios que ofrece segmentación de red para emBB, sistemas de asistencia avanzados al conductor (*Advanced Driver Assistance System*) y un segmento de mejor esfuerzo que recibe un ancho de banda garantizado, permitiendo un aprovisionamiento eficiente.

Otra estrategia para mejorar la disponibilidad de acceso al medio es considerar tres enfoques: priorizar niveles de velocidad sin depender de la latencia y la conectividad, priorizar niveles de latencia sin depender de la velocidad y priorizar niveles de conectividad sin depender de la velocidad y la latencia [55]. Bajo este concepto, el algoritmo Software-Defined Multiple-Access (SoDeMa) [55] mejora el tiempo de respuesta a las solicitudes, respaldando así la disponibilidad y el acceso a los servicios de acuerdo con las preferencias de los usuarios. En comparación con un escenario sin segmentación, este algoritmo logra una mejora del 85 % en el tiempo de respuesta.

### 3.3. Uso de aprendizaje máquina aplicado en 5G

#### 3.3.1. Avances tecnológicos con aprendizaje supervisado

El aprendizaje automático brinda a los sistemas informáticos la capacidad de aprender con datos y mejorar el rendimiento y la precisión de la toma de decisiones en una tarea específica, sin estar programados específicamente [16]. En el contexto de las redes 5G, las funciones de segmentación pueden ser automatizadas computacionalmente [60]. Por ejemplo, en [61], se presenta un modelo para gestionar las solicitudes de los usuarios mediante un proceso de decisión *Semi-Markov* que busca maximizar los ingresos del proveedor de infraestructura. La política de toma de decisiones se basa en analizar la red cuando está cerca de un estado de congestión, permitiendo al proveedor decidir si debe admitir solicitudes inmediatamente o esperar a oportunidades más rentables en el futuro.

En otra línea de investigación, [62] aprovecha la técnica Big Data Analytics (BDA) para extraer información valiosa de datos históricos y recopilar requisitos de la red. Esta información ayuda a evaluar si la aceptación de un nuevo segmento podría generar problemas en segmentos ya provisionados en la infraestructura o si podría afectar negativamente su rendimiento. Además, [24] también emplea el BDA para implementar tres algoritmos de aprendizaje supervisado: el clasificador de árbol de decisión, el clasificador Bayesiano ingenuo y el algoritmo de los **K** vecinos más cercanos. Como resultado, el algoritmo de árboles de decisión muestra un menor tiempo de entrenamiento, mientras que el clasificador Bayesiano presenta una mayor velocidad en las predicciones.

#### 3.3.2. Avances tecnológicos con aprendizaje reforzado

En términos generales, la técnica de aprendizaje reforzado tiene como objetivo mejorar el conocimiento de un agente encargado de tomar decisiones. Este agente aprende a través de recompensas o penalizaciones obtenidas tras interactuar con su entorno. Esto le permite ajustar sus comportamientos y habilidades de manera incremental, sin requerir un conocimiento completo o control absoluto del entorno [31].

En el contexto de la orquestación de recursos en la RAN de 5G, se han propuesto estrategias de admisión de segmentos que emplean el aprendizaje reforzado. Por ejemplo, en [56], se presenta una política de admisión que busca aprender qué servicios tienen el potencial de generar mayores ingresos con una penalización baja en términos de uso de recursos de red. Por otro lado, en [34], se utiliza una política de optimización que tiene como objetivo minimizar la pérdida de paquetes, reducir el retardo de paquetes y maximizar la tasa de bits garantizada.

En algunos casos, la política puede adaptarse según el segmento de red con el que se esté trabajando. En esta línea, los autores de [56] proponen un esquema de segmentación de acceso radio impulsado por inquilinos. En este enfoque, un inquilino especifica los requisitos de red y el proveedor configura los parámetros del segmento según la demanda. En este trabajo se emplea un algoritmo de aprendizaje de refuerzo profundo llamado *Deep Deterministic Policy Gradient*.

En el ámbito de mejorar el acceso radio en 5G, [63] presenta un método para asignar recursos de radio en una red inteligente utilizando RL y SDN. Un controlador SDN gestiona la red de acceso y aplica un algoritmo RL para mejorar la calidad de experiencia de los usuarios y la eficiencia del canal. Esto se realiza considerando la prioridad de cada segmento y su tasa de datos, lo que permite asignar la potencia de transmisión en función de la prioridad. De manera similar, [64] explora la aplicación de Inteligencia Artificial (IA) en la segmentación de la red en 5G y presenta un método RL basado en el enfoque actor-crítico (AC) para la asignación de recursos en la RAN y en la red central, demostrando los beneficios de la IA.

En muchos problemas prácticos de RL, el entorno no es completamente conocido, lo que ha llevado al desarrollo de algoritmos de RL sin modelo. Estos algoritmos permiten que el agente tome decisiones mediante prueba y error, basándose en estados del entorno y recompensas por acciones, en lugar de tener un conocimiento exhaustivo del entorno. Uno de los algoritmos de aprendizaje reforzado más destacados es *Q-learning*, el cual se encuentra entre los más conocidos para calcular una política óptima que maximiza las recompensas a largo plazo [42]. En el contexto de la RAN, *Q-learning* se ha utilizado como una herramienta eficiente de ML para la toma de decisiones en entornos de red desconocidos [46].

La aplicabilidad de *Q-learning* en la RAN es evidente en diversas investigaciones. Por ejemplo, en [65], se propone un método de aprendizaje por refuerzo de múltiples agentes (Multi-Agent Reinforcement Learning (MARL)) denominado Correlated *Q-learning* based interslice RB Allocation (COQRA), basado en *Q-learning*, para la asignación de bloques de recursos entre segmentos URLLC y emBB. El objetivo principal es minimizar la latencia de paquetes. En un contexto similar, [21] presenta un enfoque de *Q-learning* de dos etapas, conocido como Two-Stage *Q-learning* (TSQL), para la asignación de recursos en redes de acceso de radio en la nube 5G (Cloud RAN (C-RAN)). En C-RAN, que consta de capas Centralized Unit (CU)/Distributed Unit (DU) y Radio Remote Unit (RRU), se emplea *Q-learning* de múltiples agentes para abordar la asignación de recursos en ambas capas y reducir la complejidad de la tabla de valor  $Q$ .

La asignación dinámica de recursos para segmentos en redes de acceso radio 5G también se ha abordado mediante soluciones basadas en *Q-learning* [66]. Estas soluciones buscan maximizar la eficiencia de la red al asignar de manera óptima las solicitudes de segmentación dentro de un período específico, teniendo en cuenta limitaciones computacionales y de comunicación. Los resultados indican mejoras significativas en la utilidad de la red 5G en comparación con enfoques miopes, aleatorios y basados en el orden de llegada. En conjunto, estos ejemplos demuestran la versatilidad y eficacia del *Q-learning* en el contexto de la RAN para optimizar la asignación de recursos y mejorar el rendimiento de la red.

### 3.4. Políticas de asignación de preámbulos en redes segmentadas

El mayor desafío de segmentación en 5G es encontrar la manera de crear múltiples redes virtuales personalizables y asegurarse de que cada segmento de red tenga los requisitos adecuados para su aplicación o servicio específico sin afectar negativamente a otros segmentos de red. A través de análisis y mediciones de tráfico en la infraestructura con diferentes configuraciones de segmentación, se ha observado en [67] que la eficiencia del tráfico en el UL tiende a degradarse más significativamente que en Downlink (DL) al intentar cumplir con los mismos requisitos de QoS. Además, según [68], servicios en la red 5G, como los juegos móviles y la transmisión de video,

a menudo tienen requisitos de QoS limitados en el canal UL, lo que introduce restricciones en el RACH. El principal desafío se debe a que el acceso aleatorio en una red 5G tiene un número limitado de códigos de acceso aleatorio (preámbulos) disponibles. Para minimizar la interferencia entre dispositivos y maximizar la utilización eficiente del espectro de radio, se requieren configuraciones específicas en la RAN para asignar los preámbulos de manera óptima.

Para abordar este desafío, los autores de [69] presentan un algoritmo adaptativo llamado *Multipreamble RA*. Este algoritmo determina dinámicamente: (i) el índice de inicio del subconjunto de preámbulos que utilizará una clase de servicio (segmento) y (ii) el tamaño del subconjunto de preámbulos. Se logra una alta probabilidad de éxito en el acceso aleatorio y se posibilita el aislamiento de diferentes clases de servicios de red, introduciendo la noción de subconjuntos de preámbulos dedicados y compartidos asignados a tipos específicos de servicios [70].

Además, en [71], se introduce un algoritmo denominado *Priority-based load-adaptive preamble separation*, que establece la asignación de preámbulos en base a las prioridades de clase (eMBB, URLLC y mMTC) y la intensidad de llegadas. La jerarquía de prioridades se determina según las demandas de latencia de cada categoría, destacando la máxima prioridad para los dispositivos eMBB debido a su necesidad de baja latencia. Los dispositivos URLLC ostentan la segunda prioridad, mientras que los dispositivos mMTC, con requerimientos de latencia más flexibles, tienen la prioridad más baja.

Otro enfoque para la asignación de preámbulos se expone en [72], donde se implementa un esquema de separación dinámica de recursos en el canal RACH basado en las condiciones de la red. Los dispositivos M2M monitorean el mensaje SIB2, enviado por el gNodeB, para determinar el rango de preámbulos disponibles. Explorando la creación de políticas de prioridad alineadas con los requerimientos de los segmentos, [73] emplea técnicas de aprendizaje profundo por refuerzo. Un agente relaciona la demanda y establece recompensas conforme a la asignación de preámbulos, resultando en tasas de colisión reducidas y, por ende, un desempeño mejorado en términos de acceso aleatorio.

### 3.5. Conclusiones

El análisis del estado del arte pone en evidencia la relevancia de la segmentación de redes como respuesta a la diversidad de servicios y aplicaciones en 5G. Se han explorado diversas combinaciones para la formación de segmentos, evidenciando que esta segmentación se configura de acuerdo a los servicios a implementar. La generación de segmentos personalizables no solo brinda una gestión de recursos más efectiva, sino que también allana el camino para considerar su aplicación en una amplia variedad de áreas.

En el marco de este proyecto de titulación, se persigue ampliar los beneficios de la segmentación en la red de acceso. Se ha reconocido que el reto más significativo en el proceso radica en la equitativa asignación de recursos a cada segmento. En respuesta, se han presentado diversas técnicas de asignación de preámbulos. La formulación de políticas para la asignación de preámbulos se ha identificado como un ámbito de investigación crucial para optimizar la eficiencia y la probabilidad de acceso en la red.

Los desafíos inherentes, como la limitación de preámbulos disponibles y las variadas prioridades de servicios, requieren el desarrollo de algoritmos adaptables y dinámicos que puedan amoldarse a las condiciones cambiantes de la red. Los enfoques explorados en el estado del arte, tales como el *Multipreamble RA*, la *Priority-based load-adaptive preamble separation* y las estrategias basadas en aprendizaje reforzado, demuestran la viabilidad de mejorar la asignación de preámbulos en función de la demanda de servicios y las particularidades de la red. Estos enfoques abren la puerta a una asignación más eficiente y efectiva de recursos mediante la aplicación de ML.

## Capítulo 4 Diseño e Implementación

Este capítulo abarca el diseño y la implementación de las propuestas para mejorar la probabilidad de acceso en segmentos de red. El capítulo se estructura en varias secciones que desglosan el proceso de diseño e implementación de las soluciones propuestas. Comenzando con la Sección 4.1 donde se establece el entorno donde se llevarán a cabo las pruebas y evaluaciones de las propuestas. La Sección 4.2 explora cómo se ha logrado introducir dinamismo en el proceso de asignación. A continuación, en la sección 4.3, se describe la capacidad de trabajo de la celda, centrándonos en los procedimientos de acceso aleatorio (RAP) en el contexto de la red 5G, con un enfoque particular en el proceso de detección de preámbulos y su probabilidad de éxito en condiciones de canal inalámbrico variable. Posteriormente, en la Sección 4.4, se desglosa el comportamiento del tráfico en cada segmento, lo que establece el contexto en el que los usuarios acceden a la red tras una interrupción de señalización. La Sección 4.5 presenta los parámetros esenciales para la configuración del PRACH. Finalmente, en las Secciones 4.6 y 4.7 se profundiza en el desarrollo y la implementación de los algoritmos que conforman la base de las soluciones propuestas.

### 4.1. Caso de estudio

Esta investigación se enfoca en el análisis detallado del RAP a nivel de una celda dentro de una red de comunicaciones 5G. La elección de una celda específica permite un examen minucioso de los efectos de las colisiones en las solicitudes de acceso, al mismo tiempo que proporciona la oportunidad de manipular y ajustar variables relacionadas con la asignación de preámbulos.

El escenario de estudio se desarrolla en un contexto en el cual una celda experimenta una interrupción en la señalización, lo que lleva a la necesidad de que un grupo de usuarios restablezca la conexión con la red. La celda tiene la capacidad de identificar la cantidad de usuarios interesados en restablecer la señalización. Con el objetivo de comprender en profundidad este proceso, se presenta una caracterización exhaustiva de la capacidad operativa de la celda y los diferentes segmentos de usuarios atendidos por el proveedor de servicios (ver Sección 4.3).

## 4.2. Metodología

Con el objetivo de realizar la asignación dinámica de preámbulos de acceso para cada servicio ofrecido por 5G, se ha propuesto la utilización de dos tipos diferentes de algoritmos de ML: aprendizaje por refuerzo y aprendizaje supervisado. En el primer caso se implementa una RNA que trabaja bajo el conocimiento de una base de datos obtenida mediante el análisis de tráfico detallado en la Sección 4.4. En el segundo caso se ha implementado un algoritmo basado en *Q-learning* clásico (ver Sección 2.7), donde la toma de decisiones se lleva a cabo partir de una política de recompensas asociadas a cada estado del sistema.

En 5G, los mensajes SIB2 se envían periódicamente desde gNodeB hacia los UEs que se encuentren en el área de cobertura. En estos mensajes, se transmite información relacionada con la configuración y los parámetros de la estación base así como los parámetros de red, la capacidad de portadoras, las configuraciones de control de acceso al medio, etc. Además, los mensajes SIB también son utilizados para indicar actualizaciones en la configuración durante la operación de la red. En este trabajo se aprovecha esta característica para realizar una asignación dinámica de preámbulos para cada segmento de red. Según la referencia [74], se establece que el intervalo de envío de un nuevo mensaje SIB2 a todos los UEs es de 80 ms ( $T_{SIB2}$ ). En consecuencia, se puede considerar que  $T_{SIB2}$  representa la duración de un periodo de decisión en nuestro sistema, la cual está compuesta por uno o más RAOs. En el contexto de este trabajo, se generan RAOs cada 5 milisegundos, por lo tanto, un  $T_{SIB2}$  abarca 16 RAOs. Esto implica que durante ese lapso de tiempo se lleva a cabo una única decisión relacionada con la asignación de preámbulos (Ver Figura 4.1).

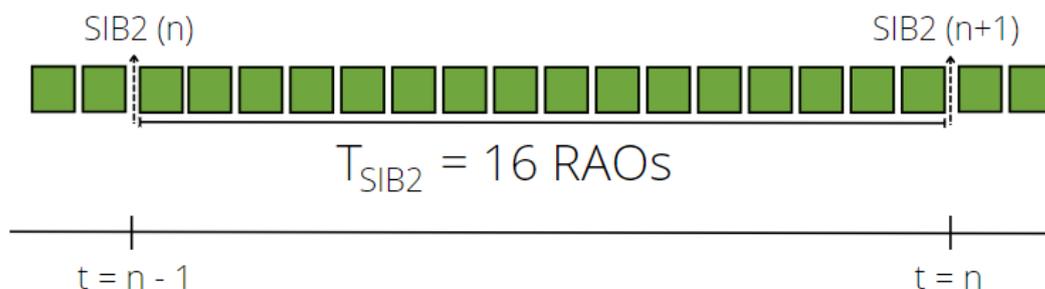


Figura 4.1: Periodo de decisión  $T_{SIB2}$ .

### 4.3. Capacidad de trabajo de la celda

Como se definió en la Sección 2.1.4.1, cuando un UE desea tener recursos de comunicación se ejecuta un RAP. En 5G se admite dos tipos de RAP: acceso aleatorio de 4 pasos y acceso aleatorio de 2 pasos. Ambos tipos de procedimiento pueden ser basados en contención o sin contención [20]. El RAP a utilizar está basado en contención de 4 pasos.

En la primera etapa del proceso, el UE envía un mensaje inicial conocido como preámbulo. En esta fase inicial, pueden ocurrir tres resultados posibles: en primer lugar, la colisión de preámbulos; en segundo lugar, la decodificación errónea del preámbulo; y en tercer lugar, la decodificación correcta del preámbulo. Además, dado que se está enviando un mensaje dentro de un canal inalámbrico y dinámico, no se puede asegurar que este mensaje llegue al decodificador y sea decodificado de forma correcta. La razón principal detrás de una decodificación incorrecta se da por motivo de efectos del canal de radio, como, por ejemplo, desvanecimiento, interferencia entre celdas, pérdida por trayectorias multicamino, etc.

De acuerdo con [75], en ausencia de colisión, la probabilidad de detección del preámbulo está definida por la Ecuación (4.1).

$$P_d = 1 - \frac{1}{e^k} \quad (4.1)$$

Donde  $k$  indica la  $k$ -ésima transmisión del preámbulo.

Como se ha mencionado previamente, se busca implementar técnicas de segmentación en la red de acceso; bajo esta idea, se define en [75] que la separación de recursos se puede realizar dividiendo los preámbulos en grupos en función de los segmentos a implementar. Para la implementación a realizar se plantea que los segmentos a utilizar sean H2H, mMTC y emBB.

#### 4.4. Modelos de distribución de arribos

En el contexto del escenario propuesto, lograr una comprensión y modelado precisos de los patrones de llegada de usuarios a la red resulta de vital importancia. Esto se debe a que el diseño de políticas para la asignación dinámica de preámbulos está intrínsecamente vinculado a los patrones de arribo previstos en la red. En esta sección, se aborda detalladamente los modelos de distribución de arribos para cada segmento de red.

##### 4.4.1. Tráfico mMTC

En [75] se presentan dos modelos de tráfico para mMTC. El primer modelo de tráfico hace referencia a un escenario realista en el que los UE acceden a la red de manera uniforme durante un periodo de tiempo. El segundo, en cambio, representa un escenario extremo en el que una gran cantidad de dispositivos mMTC acceden a la red de manera altamente sincronizada. La 3rd Generation Partnership Project (3GPP) en [75] recomienda el uso de los modelos de tráfico descritos en la Tabla 4.1.

Tabla 4.1: Modelos de tráfico para mMTC.

Características	Modelo de Tráfico 1	Modelo de tráfico 2
Número de dispositivos mMTC	1000, 3000, 5000, 10000, 30000	1000, 2000, 5000, 10000, 30000
Distribución de arribos	Distribución uniforme sobre $T$	Distribución Beta sobre $T$

Se debe tomar en cuenta que se asume que el número de dispositivos  $N$ , detallados en la Tabla 4.1, envían sus peticiones de acceso en un tiempo entre  $[0, T]$ ; esta consideración es de importancia al momento de desarrollar la simulación. La intensidad del acceso aleatorio por RAO se describe mediante la distribución  $p(t)$  y el número total de UEs (Ver ecuación (4.2)).

$$Intensidad_{Acceso}(n) = \left[ N \int_{RAO_n}^{RAO_{n+1}} p(t) dt \right] \quad (4.2)$$

Para el presente estudio se enfocará en el segundo modelo de tráfico  $p(t)$  con la Ecuación (4.3) que define la distribución Beta.

$$p(t) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}Beta(\alpha, \beta)} \quad \alpha > 0, \beta > 0 \quad (4.3)$$

Donde  $p(t)$  representa la cantidad de usuarios esperados por RAO,  $\alpha$  y  $\beta$  son valores positivos que influyen en la forma de la distribución. El parámetro  $\alpha$  controla cómo se configura la distribución en su extremo izquierdo, mientras que el parámetro  $\beta$  afecta la forma de la distribución en su extremo derecho. En términos generales, aumentar el valor de  $\alpha$  ocasiona un desplazamiento hacia la derecha en la distribución, mientras que incrementar el valor de  $\beta$  produce un desplazamiento hacia la izquierda [76]. Para este estudio en particular, se consideran los valores  $\alpha = 3$  y  $\beta = 4$  como lo recomienda [75].

#### 4.4.2. Tráfico H2H

En [5] se define que los intentos de acceso de los UE H2H se distribuyen de forma uniforme en el tiempo. En [9] se ha definido que durante un periodo menor a 10 minutos, el tráfico H2H se considera constante con una tasa promedio de 1 arribo por RAO. Este análisis se basó en datos de red recopilados por el operador italiano "Telecom Italia", específicamente en una celda ubicada en el centro de Milán, alrededor de las 4:00 p. m. Por lo tanto, la función de probabilidad  $p(t)$  se define en la Ecuación (4.4) y es usada para evaluar la intensidad de peticiones de acceso por RAO, tal como se describe en la Ecuación (4.2). En este contexto, el valor de  $N$  representa la cantidad de oportunidades de acceso disponibles, como se detalla en [9].

$$p(t) = \begin{cases} \frac{1}{b-a} & t \in (a, b) \\ 0 & t \notin (a, b) \end{cases} \quad (4.4)$$

#### 4.4.3. Tráfico emBB

Para el modelo de tráfico emBB se toma en cuenta que debido a los requerimientos de este segmento se generan patrones de actualizaciones. Los autores de [77] mencionan 3 patrones de generación de actualizaciones que son:

- *Exponencial*: para este tipo de actualizaciones, el tiempo entre actualizaciones sigue la distribución exponencial, por medio de un proceso de Poisson de tasa  $U$  (tráfico generado por el número de actualizaciones por segundo que cada usuario carga a la red).
- *Uniforme*: el tiempo en el que se envía una actualización es uniforme, validando que en cada intervalo exista solo una actualización. El número de actualizaciones es caracterizado por una distribución de Poisson en función de  $T$ .

Para el presente caso de estudio se usará el modelo de tráfico uniforme en función al estudio realizado en [77].

#### 4.5. Configuración del RACH

En la Sección 2.1.4.1 se definió el RAP y las variables que se implican en el proceso; estas variables fueron incluidas en la especificación [27] en donde se define que el controlador de recursos de radio debe configurar los siguientes parámetros para el RAP.

- `prach-ConfigurationIndex`: Indica el conjunto disponible de ocasiones PRACH para la transmisión del preámbulo de acceso aleatorio para `Msg1`.
- `preambleTransMax`: Indica el número máximo de transmisión de preámbulo de acceso aleatorio.
- `Ra-ResponseWindowSize`: Tiempo de espera de respuesta para los usuarios colisionados.

Para llevar a cabo esta investigación, se ha considerado trabajar bajo la configuración RACH típica especificada en [75], donde se detalla que el `prach-ConfigIndex` es de 6 (ver Apéndice A); por lo que consecuentemente la periodicidad del RAO es de 5 ms (ver Figura 2.7) y la longitud de la subtrama es de 1 ms. Se dispone de un total de 54 preámbulos que serán distribuidos de manera dinámica para cada segmento. El parámetro `preambleTransMax` se ha establecido en 10, lo que significa que cada UE tiene permitido realizar un máximo de 10 intentos de transmisión de preámbulo durante el proceso de acceso aleatorio en la red.

En cambio, para la transmisión del Msg3 se utilizan recursos reservados que ascienden a 15. Esta asignación se debe al hecho de que el tamaño de la ventana de acceso aleatorio (RA-ResponseWindowSize) se ha configurado con una duración de 5 subtramas, y el número máximo de concesiones de UL por ventana se ha establecido en 3. A partir de [75], se define que la forma ideal de trabajo para las concesiones UL (recursos reservados del enlace ascendente) es de 10 para el segmento de mMTC, 4 para emBB y 1 para H2H. En la Tabla 4.2 se presentan algunos parámetros adicionales utilizados durante el desarrollo de este estudio.

Tabla 4.2: Configuración del RACH.

Parámetro	Configuración
Índice de configuración PRACH	prach-ConfigIndex= 6
Periodicidad de RAOs	5ms
Longitud de subtrama	1ms
Preámbulos disponibles	54
Número máximo de transmisiones de preámbulo	10
Tamaño de la ventana RAR	$W_{RAR} = 5$ subtramas
Número máximo de concesiones de enlace ascendente por subtrama	$N_{RAR} = 3$
Número máximo de concesiones de enlace ascendente por ventana RAR	$N_{UL} = W_{RAR} \times N_{RAR} = 15$
Probabilidad de detección de preámbulo para la k-ésima transmisión de preámbulo	$P_d = 1 - \frac{1}{e^k}$
Indicador de retroceso	BI=20ms
Probabilidad de retransmisión para Msg3 y Msg4	0,1
Número máximo de transmisiones de Msg3 y Msg4	5
Retraso en el procesamiento del preámbulo	2 subtramas
Retraso en el procesamiento de la concesión del enlace ascendente	5 subtramas
Retraso en el procesamiento de la solicitud de conexión	4 subtramas
Tiempo de ida y vuelta del Msg3	8 subtramas
Tiempo de ida y vuelta del Msg4	5 subtramas

#### 4.6. Enfoque de Redes neuronales

El diseño e implementación de una RNA para la administración dinámica de preámbulos involucra una serie de etapas interconectadas, como se ilustra en la Figura 4.2. Inicia con la recolección de datos a través de simulaciones que modelan escenarios de tráfico para segmentos de comunicación como mMTC, emBB y H2H (consulte la Sección 4.6.1). Tras esta fase, se procede al procesamiento de los datos, con el objetivo principal de construir una base de datos aplicable al escenario propuesto (detallado en la Sección 4.6.2). Esta base de datos servirá de entrenamiento a la RNA.

El proceso continúa con el diseño de la RNA, descrito en la Sección 4.6.3, donde se aborda la arquitectura de la RNA en la herramienta MATLAB. Finalmente, se realiza la evaluación y validación de la arquitectura de la RNA, como parte de la Sección 5.2. En esta fase, diversas arquitecturas son sometidas a pruebas para determinar la que

ofrece el mejor rendimiento en términos de entrenamiento y resultados.

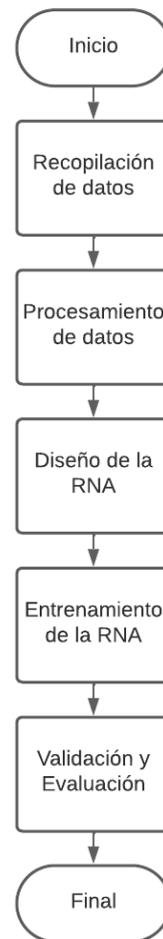


Figura 4.2: Diagrama de Flujo para el diseño de una RNA

#### 4.6.1. Recopilación de datos

El proceso de recopilación de datos es la base para la construcción y entrenamiento del modelo de RNA de aprendizaje supervisado. Para establecer la base de datos, se consideraron los modelos de distribución de tráfico previamente definidos en la Sección 4.4. Estos modelos de distribución constituyen funciones de probabilidad que incorporan variables críticas como el tiempo de evaluación  $T$  y la cantidad anticipada de usuarios que ingresarán durante dicho período. En este contexto, se llevaron a cabo simulaciones para estimar los valores esperados de la cantidad de arribos por cada RAO en un intervalo de tiempo  $T$ . En las Secciones 4.6.1.1, 4.6.1.2 y 4.6.1.3 se detalla el proceso realizado para cada segmento.

#### 4.6.1.1. Comunicación mMTC

Para evaluar el tráfico mMTC, la 3GPP ha propuesto la utilización de una función de distribución Beta con parámetros  $\alpha = 3$  y  $\beta = 4$ , lo que permite reflejar con precisión el comportamiento esperado de este tipo de tráfico en el escenario propuesto. Se plantea trabajar con un valor total Users Equipments for mMTC (UEsM) de 1000, 3000, 5000, 10000 y 30000. Estos valores corresponden a la cantidad total de peticiones realizadas por los dispositivos del grupo mMTC descrito en la Tabla 4.1.

Para evaluar el tráfico se realizan simulaciones que emulan la llegada de dispositivos. Cada simulación se repite un total de 100 veces para capturar diversas instancias del comportamiento del tráfico durante el período  $T$ . A partir de estos resultados, se calculan los valores promedio de los datos generados en cada repetición. Esta metodología brinda una comprensión más amplia del comportamiento promedio del tráfico mMTC. En la Figura 4.3, se presentan los UEsM por cada  $n$ -ésimo SIB2 transmitido para un total anticipado de 5000, 10000 y 30000 usuarios respectivamente. Estas representaciones gráficas permiten visualizar de manera clara el comportamiento de los arribos de tráfico mMTC tal como lo define la Ecuación (4.2), donde los límites de integración representan un periodo SIB2.

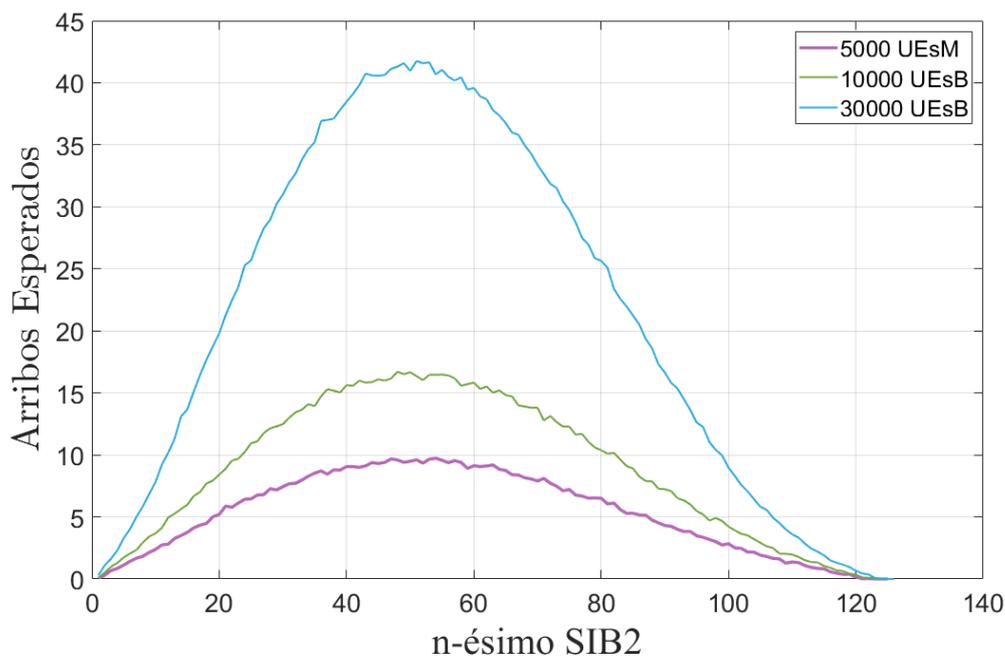


Figura 4.3: Intensidad de peticiones de acceso para un total de 5000, 10000, 30000 UEsM.

#### 4.6.1.2. Comunicación emBB

El modelo de tráfico emBB se fundamenta en la consideración del número de actualizaciones generadas por cada Users Equipments for emBB (UEsB). Se espera que cada usuario que busque acceder a la red genere en promedio 10 actualizaciones. La distribución de estas actualizaciones se describe mediante una función estocástica de Poisson, con una tasa media  $\lambda = 10$  como lo recomienda [77] en su estudio realizado.

El modelo de tráfico emBB exhibe características uniformes y se somete a simulaciones con valores UEsB de 500, 1000 y 1500. Cada simulación se repite un total de 100 veces para luego promediarlas. El resultado se ilustra en la Figura 4.4. Estos valores reflejan la cantidad anticipada de usuarios que accederán a la red y participarán en RAP.

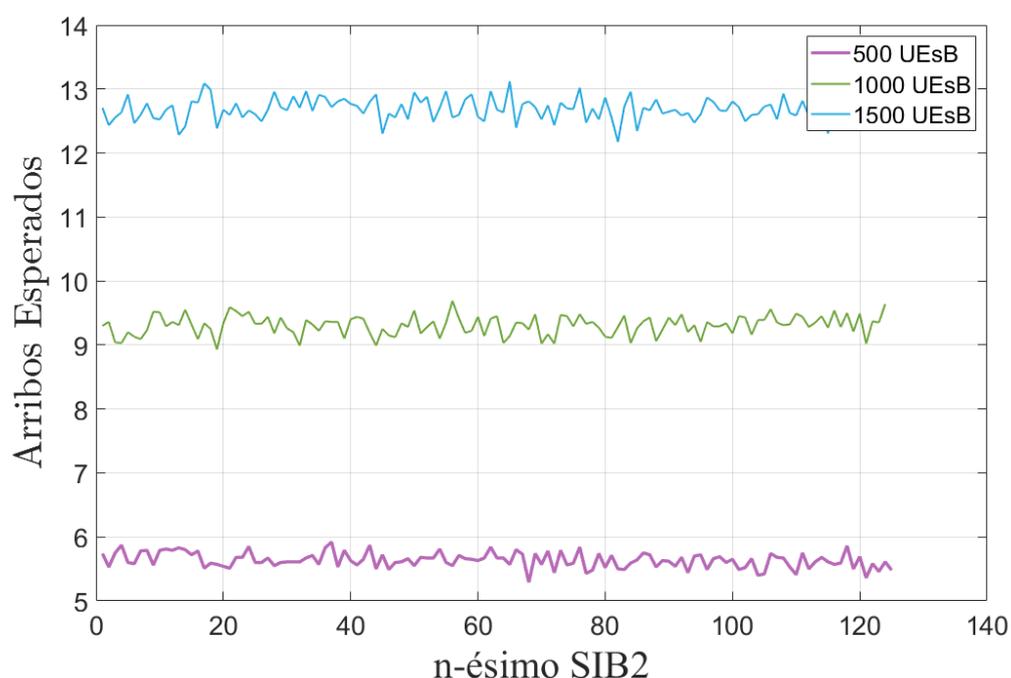


Figura 4.4: Intensidad de peticiones de acceso para un total de 500, 1000, 1500 UEsB.

#### 4.6.1.3. Comunicación H2H

El tráfico H2H se caracteriza por una distribución uniforme, en la que se espera que llegue un arribo por cada RAO. Al igual que en los casos anteriores, se llevó a cabo la simulación de arribos Users Equipments for H2H (UEsH) vs SIB2  $n$ -ésimo. Los

resultados de un tráfico con 5000 UEsH se presentan en la Figura 4.5.

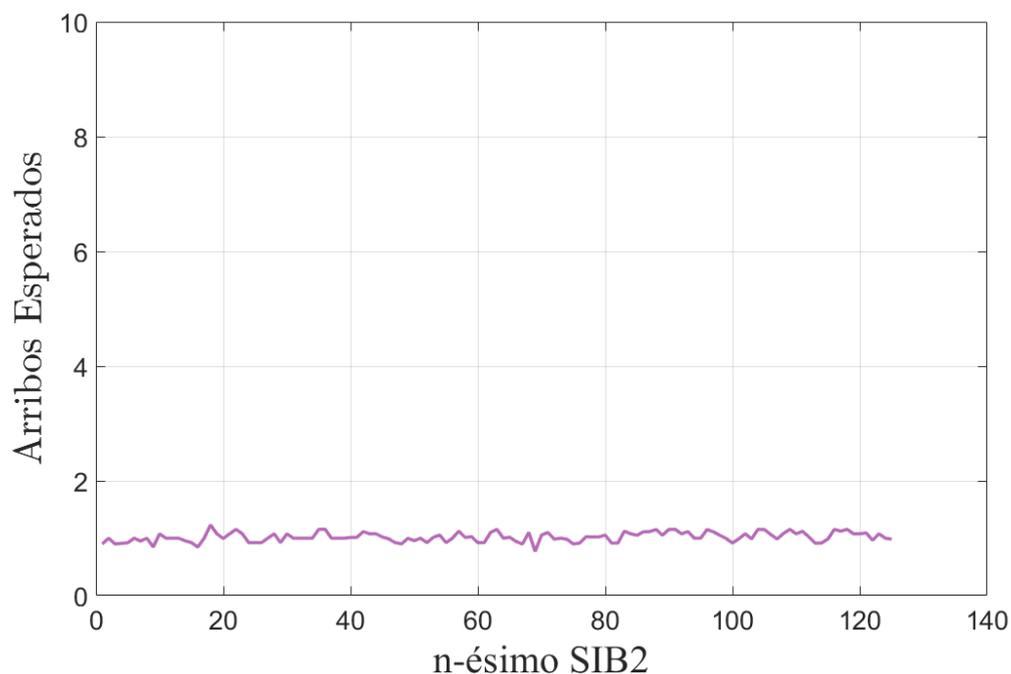


Figura 4.5: Intensidad de peticiones de acceso para un tráfico H2H.

#### 4.6.2. Procesamiento de datos

Una vez obtenidos los datos de las simulaciones anteriores, es esencial someterlos a un procesamiento para construir una base de datos que pueda ser adecuadamente utilizada en la red neuronal propuesta. En esta fase, se extraen características clave de los datos que permitirán una aplicación efectiva del modelo de aprendizaje supervisado en la administración dinámica de preámbulos.

Se destacan las siguientes características de interés:

- La cantidad anticipada de llegadas está en relación directa con la gráfica de intensidad de peticiones de acceso. En términos simples, a medida que aumenta el tráfico esperado, la amplitud de la intensidad de tráfico también se amplía. Este principio es válido tanto para mMTC como para emBB.
- La diferencia entre el número de concesiones UL entre mMTC ( $10 N_{UL}$ ), emBB ( $4 N_{UL}$ ) y H2H ( $1 N_{UL}$ ) sugiere la necesidad de enfocarse en los segmentos de comunicación mMTC y emBB.

- Dado que mMTC posee 10  $N_{UL}$  por cada RAO en una simulación compuesta por 200 RAOs se puede garantizar que en el caso de que ningún preámbulo haya colapsado se deberán acceder exitosamente un máximo de 20000 UEsM.
- Dado que emBB posee 4  $N_{UL}$  por cada RAO en una simulación compuesta por 200 RAOs se garantiza que en el caso de que ningún preámbulo haya colapsado se deberán acceder exitosamente un máximo de 8000 UEsB.

Las retransmisiones generadas por usuarios que colapsan, no decodifican correctamente su preámbulo o carecen de concesiones UL pueden ser significativas y desencadenar colisiones en arribos futuros. Este fenómeno ha sido estudiado por [9] en el contexto del tráfico mMTC. La Figura 4.6 ilustra cómo las colisiones tienen un efecto de bloqueo en los accesos exitosos. Las retransmisiones masivas ocurren cuando la cantidad de arribos supera el número de concesiones disponibles. Este mismo efecto de bloqueo en el acceso se manifiesta en el segmento emBB cuando las retransmisiones se vuelven masivas.

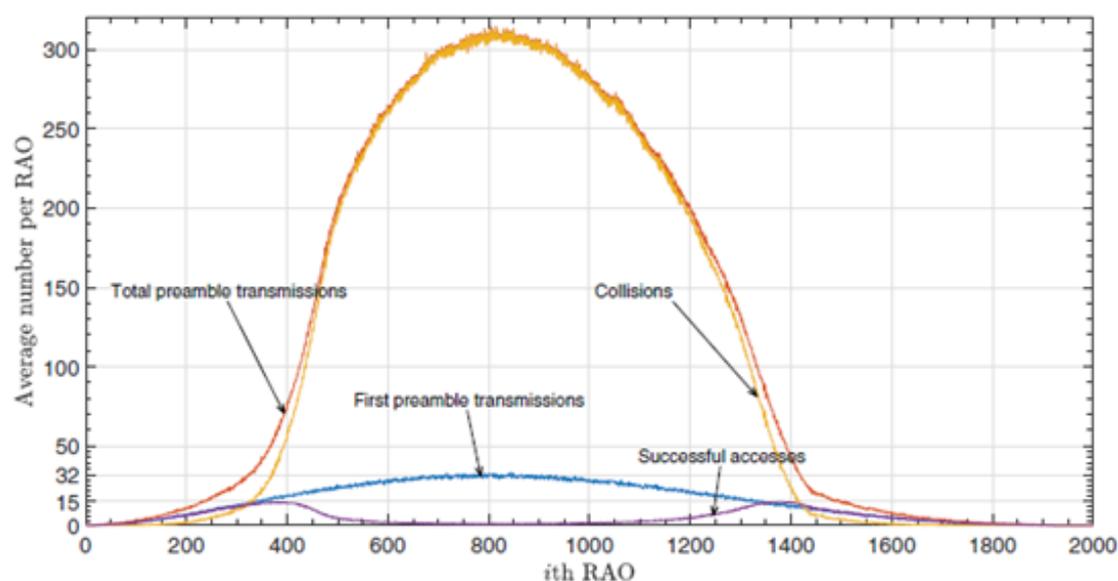


Figura 4.6: Distribución temporal de arribos UEsM (primeras transmisiones de preámbulo), transmisiones de preámbulo totales, colisiones y accesos exitosos por RAO, sin control de acceso implementado [9].

Considerando el efecto de bloqueo causado por las colisiones, se propone la creación

de una política de datos para maximizar la eficiencia del acceso exitoso. En este sentido, la base de datos de la red neuronal debe contener información crítica, como el valor esperado de accesos para emBB, el valor esperado de accesos para mMTC y el  $n$ -ésimo mensaje SIB2 en el que se encuentra la simulación. A partir de estos datos, se pueden derivar las siguientes variables:

- $UEsM$ : Total esperado de usuarios para la comunicación mMTC.
- $UEsB$ : Total esperado de usuarios para la comunicación emBB.
- $UEsM_n$ : Número de arribos esperados para mMTC en el  $n$ -ésimo mensaje SIB2.
- $UEsB_n$ : Número de arribos esperados para emBB en el  $n$ -ésimo mensaje SIB2.
- $N_{ULM}$ : ULs disponibles para mMTC.
- $N_{ULB}$ : ULs disponibles para emBB.
- $w_p(n)$ : Coeficiente de prioridad. Se define en la Ecuación (4.5). Aquí, un valor de 1 indica que las peticiones de acceso superan la cantidad de recursos UL disponibles y requieren el uso del 100% de los preámbulos disponibles. Por otro lado, un valor de 0.7 señala que si las peticiones de acceso no exceden la cantidad de recursos UL disponibles, se puede asignar directamente el 30% de los preámbulos a otro segmento.

$$w_p(n) = \begin{cases} 0,7 & UEsM \leq N_{ULM} \\ 1 & UEsM > N_{ULM} \end{cases} \quad (4.5)$$

- $max_P$ : Número total de preámbulos a agrupar.
- $w_{rao}(n)$ : Representa la carga generada al avanzar con el trayecto de la simulación. Se define en la Ecuación (4.6.2). EL valor 125 representa la cantidad total de mensajes SIB2 que se dan en el tiempo de simulación.

$$w_{RAO}(n) = \frac{n}{125} \quad (4.6)$$

Con estas variables definidas, se procede a formular la política de decisión para la

base de datos que determinará el preámbulo ideal para la comunicación mMTC. Es esencial tener en cuenta que la asignación de preámbulos a cada segmento no puede basarse únicamente en la cantidad esperada de peticiones de acceso por RAO. Si se sigue esta estrategia, se producirá un desperdicio de recursos debido a la disponibilidad de recursos UL. Por esta razón, la Ecuación (4.7) cumple la función de reservar una porción de preámbulos, con el propósito de asignarlos al segmento que requiera recursos adicionales, tomando en cuenta la disponibilidad de recursos UL que no han sido utilizados. La variable  $R_{disponible}$  representa la cantidad de preámbulos disponibles sin considerar aquellos que se reservan.

$$R_{disponible} = (maxP \cdot w_{p_n} + w_{RAO_n}) \quad (4.7)$$

A continuación, los preámbulos disponibles ( $R_{disponible}$ ) pueden dividirse de manera equitativa utilizando una regla de 3. En la Ecuación 4.8 denota la salida para el caso de la comunicación mMTC.

$$Y_M(n) = \frac{R_{disponible} \cdot UEsM_n}{UEsM_n + UEsB_n} \quad (4.8)$$

Para la comunicación H2H, dada su baja demanda de recursos, se asume directamente que los preámbulos asignados serán siempre  $Y_H = 1$ . Por contraste, para la comunicación emBB, la cantidad de preámbulos asignados estará determinada por la diferencia entre los preámbulos disponibles y los ya ocupados por los otros segmentos (ver Ecuación (4.9)).

$$Y_B(i) = maxP - Y_M(i) - 1 \quad (4.9)$$

Las variables  $Y_M$ ,  $Y_B$  y  $Y_H$  representan las salidas deseadas ante un escenario donde las retransmisiones no existen. En las Figuras 4.7 y 4.8 se puede observar el resultado de aplicar la política diseñada anteriormente. El resultado maximiza el acceso exitoso utilizando la mayor parte del tiempo todos los recursos UL disponibles.

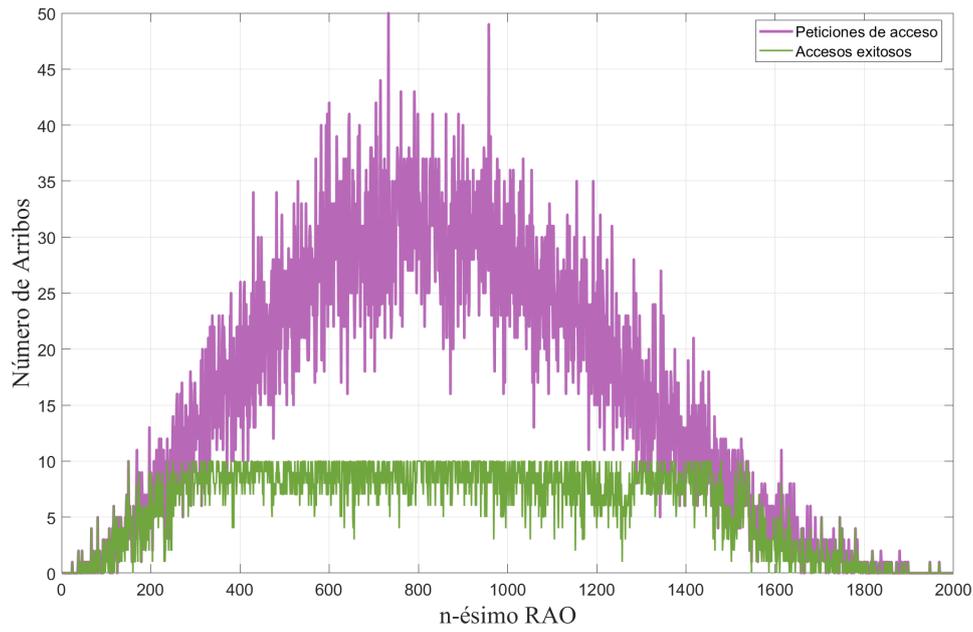


Figura 4.7: Gráfica comparativa entre las peticiones de acceso y el acceso exitoso para una comunicación mMTC con un valor total de 30000 UEsM.

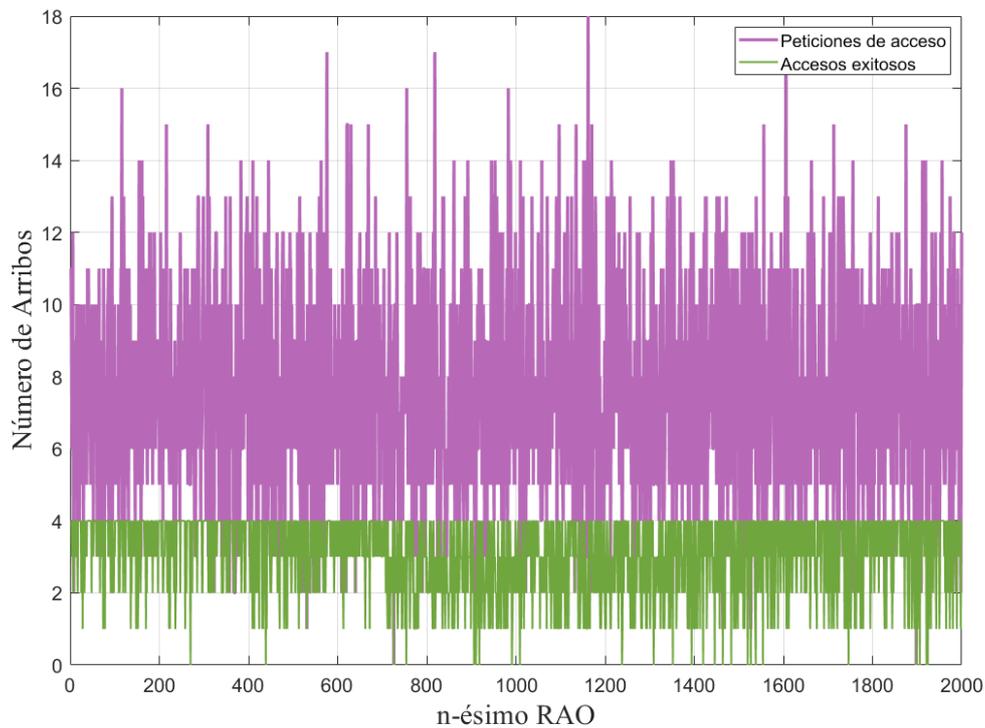


Figura 4.8: Gráfica comparativa entre las peticiones de acceso y el acceso exitoso para una comunicación eMBB con un valor total de 1500 UEsB.

En el escenario descrito en la Sección 4.1, es importante tener en cuenta que las retransmisiones pueden llegar a alcanzar hasta un máximo de 10 peticiones. Estas retransmisiones se consideran una fuente de ruido que puede afectar los datos en la base de datos y además provocan un aumento de peticiones que se acumula con el paso del tiempo debido al aumento de colisiones (ver Figura 4.6). La presencia de ruido en los datos de entrada tiene el potencial de introducir errores y variaciones en las salidas de la RNA lo que podría mermar su capacidad para realizar predicciones precisas. Por esta razón, el siguiente paso consiste en llevar a cabo una evaluación detallada y realizar ajustes manuales en la base de datos. Este proceso tiene como objetivo principal adaptar las variables de entrada y salida de manera óptima al escenario de prueba, considerando la influencia del ruido introducido por las retransmisiones.

#### 4.6.3. Diseño de la red neuronal artificial

El diseño de una RNA es un paso esencial en la construcción de modelos predictivos y clasificatorios eficientes. El propósito de la RNA a diseñar es procesar conjuntos de datos de entrada y generar predicciones correspondientes como salidas. Basándonos en la base de datos generada en la sección anterior, se establece que la red neuronal debe constar de 3 variables de entrada y 2 de salida.

Las variables de entrada son:

- $UEsM$ : Total esperado de usuarios para la comunicación mMTC.
- $UEsB$ : Total esperado de usuarios para la comunicación emBB.
- $n$ : Momento actual de la simulación.

Las variables de salida son:

- $Y_M$ : Preámbulos asignados al segmento mMTC.
- $Y_B$ : Preámbulos asignados al segmento emBB.

La arquitectura de una RNA se construye mediante la disposición de capas, incluyendo  $m$  capas ocultas y una capa de salida. Cada capa oculta incorpora un número

específico de neuronas que realizan cálculos intermedios para transformar y procesar los datos de entrada en características relevantes para la predicción.

En la caja de herramientas de redes neuronales de MATLAB, se utiliza la función `feedforward` para crear una red neuronal de múltiples capas tipo *feedforward*. Esto se logra definiendo el número de capas ocultas a utilizar, así como el número de neuronas mediante un vector. Además, se especifica la función de activación correspondiente para cada capa y el tipo de entrenamiento a emplear. En el Extracto de código 4.1 se presenta un ejemplo de cómo se crea la estructura básica de la red neuronal.

Extracto de código 4.1: Estructura básica para la creación de una RNA en MATLAB

```
net=feedforwardnet([n1 n2 ... nm]);
net.layers{1}.transferFcn='funcionActivacion'
.
.
.
net.layers{m+1}.transferFcn='funcionActivacion'
net.trainFcn='tipoEntrenamiento'
```

En donde  $n_m$  indica el número de neuronas de la capa  $m$  que tendrá la red, y el comando `net.layers.transferFcn = 'funcionActivacion'`; permite indicar la función de activación de la capa oculta. En este caso, se utilizan funciones de activación como `tansig` y `logsig` que introducen no linealidades en la red, lo que le permite modelar relaciones complejas en los datos de entrada y salida tal como lo recomienda [8, 78].

La cantidad de neuronas en una capa oculta puede tener un impacto significativo en el rendimiento y la capacidad de la red para aprender y generalizar a partir de datos. Usar una cantidad pequeña de neuronas implica dificultades para capturar patrones complejos de datos, provocando que la red no sea flexible en cuanto a presentar variables de entrada y de salida. Utilizar un número excesivo de neuronas conlleva a un fenómeno denominado sobreajuste, en el cual la red neuronal tiende a memorizar los datos de entrenamiento en lugar de adaptarse a variaciones y patrones más amplios [79]. Se considera que la cantidad de neuronas usada para cada capa vaya en la escala de 2 a 12 neuronas bajo el criterio de proporcionar flexibilidad y explorar dife-

rentes configuraciones para encontrar la que mejor se adapte al problema. Comenzar con una cantidad pequeña de neuronas (2) puede ayudar a la red a capturar patrones simples y luego aumentar gradualmente la cantidad de neuronas para evaluar su capacidad para manejar patrones más complejos.

Los autores de [8, 38] definen que la última capa oculta de una red neuronal se genere dependiendo del problema de optimización. Si es un problema de regresión lo ideal es usar 1 neurona. Si el problema es de clasificación se busca usar un número de neuronas equivalente al número de grupos de clasificación. Al momento de realizar el entrenamiento, la capa de salida se configura con dos neuronas, cada una de las cuales representa una de las variables de salida ( $Y_M$  y  $Y_B$ ).

Para determinar el número adecuado de capas ocultas a emplear, es esencial considerar el riesgo de sobreajuste que surge cuando se utilizan un exceso de capas [79, 80]. Con el propósito de prevenir este problema, se realizarán experimentos utilizando 2, 3 y 6 capas ocultas.

Si consideramos que tenemos  $n$  capas ocultas, podemos dividir las en dos grupos: las primeras  $n/2$  capas se denominan “capas ocultas iniciales”, mientras que las últimas  $n/2$  capas se denominan “capas ocultas finales”. En las capas ocultas iniciales se optará por la función de activación `logsig`, debido a su versatilidad y aplicabilidad en diversos problemas. Para las capas ocultas finales, se empleará la función de activación `tansig`, con el propósito de introducir elementos no lineales en la red [8, 81]. Para la capa de salida, se seleccionará la función lineal `purelin`, ya que es idónea en problemas de regresión, donde se busca predecir valores continuos [82]. Además, en nuestra red neuronal artificial (RNA) se utilizará el algoritmo de entrenamiento `traingdx` [8].

Por último, es fundamental tener en cuenta los parámetros de entrenamiento para la configuración de una RNA. En el Extracto de código 4.2 se presenta un ejemplo de configuración, donde los valores `a`, `b` y `c` son utilizados de forma ficticia para representar los parámetros de entrenamiento que pueden ser ajustados según las necesidades específicas de la red neuronal.

Extracto de código 4.2: Configuración de parámetros de entrenamiento en MATLAB

```
net.trainParam.epochs = a;
net.trainParam.goal = b;
net.trainParam.time = c;
```

Estas variables desempeñan los siguientes roles:

- `net.trainParam.epochs`: Indica el número de épocas que la red neuronal iterará durante el proceso de entrenamiento.
- `net.trainParam.goal`: Representa el valor mínimo deseado para el error final. Se conoce también como rendimiento u objetivo de rendimiento.
- `net.trainParam.time`: Establece el tiempo máximo permitido para el proceso de entrenamiento.
- `net.trainParam.min_grad`: Define el umbral mínimo para el gradiente de la función de error con respecto a los pesos y sesgos de la red neuronal. Cuando el gradiente es menor que este umbral, se detiene el entrenamiento, indicando que se ha logrado una convergencia suficiente o que los cambios en los parámetros son pequeños. [83]

En la Tabla 4.3 se puede observar los valores escogidos para el entrenamiento de la RNA. Se ha seleccionado un número considerable de épocas para permitir que el entrenamiento de la RNA realice un número adecuado de iteraciones, lo que facilita el ajuste de los pesos y la reducción del error en las predicciones [84]. De manera similar, los valores de rendimiento y gradiente actúan como indicadores de convergencia durante el entrenamiento. Un valor tan bajo como  $1,00E - 05$  sugiere que el proceso de entrenamiento ha llegado a un punto de estabilidad en el cual incrementar las iteraciones no brindará mejoras significativas al modelo [81].

Tabla 4.3: Valores objetivos de entrenamiento de la RNA.

Parámetro	Valor objetivo
Época	10000
Tiempo de entrenamiento	-
Rendimiento	1,00E-05
Gradiente	1,00E-05

#### 4.7. Enfoque de aprendizaje reforzado

El proceso de diseño e implementación del algoritmo *Q-learning* se divide en etapas fundamentales, cada una contribuyendo de manera esencial a la creación de un agente capaz de tomar decisiones óptimas en un entorno determinado, como se ilustra en la Figura 4.9. En la Sección 4.7.1, se establece la base sólida al modelar el sistema en cuestión, definiendo las acciones disponibles para el agente, los parámetros que conforman el estado del sistema y estrategias para asignar recompensas. Luego, en la Sección 4.7.2, se lleva a cabo la implementación del algoritmo *Q-learning*, lo que implica inicializar la matriz  $Q$ , definir métricas que caracterizan el estado, seleccionar preámbulos para asignar a cada segmento, obtener resultados y actualizar la tabla  $Q(s, a)$ . Finalmente, la evaluación y validación de los hiperparámetros de *Q-learning* se ejecuta como parte integral de la Sección 5.3. En este punto, se realizan un conjunto de pruebas utilizando una variedad de valores para el factor de aprendizaje, el factor de descuento y el tiempo de entrenamiento, con el propósito de identificar los valores que maximizan el rendimiento del sistema en función de los KPI.



Figura 4.9: Diagrama de Flujo para el diseño de un *Q-learning*

### 4.7.1. Modelo del sistema

El modelo de sistema en un algoritmo de *Q-learning* se refiere a la representación de un problema en el que un agente está interactuando y aprendiendo con el entorno a través de la toma de decisiones en determinados estados específicos. La acción tomada por el agente recibe una retroalimentación en base a recompensas, las cuales pueden ser positivas o negativas, en dependencia de su utilidad en la resolución del problema.

#### 4.7.1.1. Acción

En el contexto de 5G, considerando un sistema con 54 preámbulos disponibles y garantizando que al menos se asigne un preámbulo a cada segmento, se pueden definir 52 posibles acciones dentro del esquema del sistema para asignar un preámbulo a un segmento determinado,  $a \in \{1, 2, 3, 4, 5, \dots, 51, 52\}$ .

La acción tomada en cada transmisión del mensaje SIB2 se llevará a cabo en el primer RAO de cada periodo de decisión y solo afectará a los 16 RAOs que lo componen. Con el fin de lograr esto, es necesario que cada estado, denotado por  $s$ , represente la situación actual de todos los RAOs que conforman dicho periodo (ver Figura 4.10).

#### 4.7.1.2. Estado

En el contexto de este trabajo, el estado para un periodo determinado se define como  $s = (mN_{UEs}, CVN_{UEs}, \Delta N_{UEs}, N_{PA})$ . A continuación se detallan los componentes de este estado:

- $mN_{UEs}$ : Se calcula el número de UEs que solicitan acceso por primera vez ( $N_{UEs}$ ) entre todos los RAOs de un SIB2 y se selecciona el valor máximo como se muestra en la Ecuación (4.10). Emplear el valor máximo de solicitudes de acceso como métrica para la asignación de preámbulos asegura la disponibilidad de recursos suficientes para manejar situaciones de alta demanda.

$$mN_{UEs} = \text{máx} \{N_{UEs_1}, N_{UEs_2}, N_{UEs_3}, \dots, N_{UEs_k}\} \quad k = 1, 2, 3, \dots, T_{SIB2} \quad (4.10)$$

- $CVN_{UEs}$ : Se trata del coeficiente de variación de los UEs que solicitan acceso por primera vez de cada RAO de un SIB2 en relación con el valor  $mN_{UEs}$ . Se define por la Ecuación (4.11). La variabilidad de las solicitudes de acceso entre las RAOs con respecto al valor máximo ( $mN_{UEs}$ ) constituye un indicador valioso para ajustar la asignación de preámbulos. Una alta variabilidad sugiere la posibilidad de reducir preámbulos debido a que los valores de solicitudes de acceso en las RAOs individuales tienden a ser significativamente menores que el valor máximo, mientras que una baja variabilidad puede indicar la necesidad de asignar más preámbulos debido que los valores de solicitudes de acceso en las RAOs son similares y cercanos al valor máximo.

$$CVN_{UEs} = \frac{\left[ \frac{1}{T_{SIB2}-1} \left( \sum_{k=1}^{T_{SIB2}} |N_{UEs_k} - mN_{UEs}|^2 \right) \right]^{\frac{1}{2}}}{mN_{UEs}} \quad (4.11)$$

- $\Delta N_{UEs}$ : Se describe en la Ecuación (4.12) y es la diferencia entre el número máximo de UEs que solicitan acceso por primera vez en el periodo actual y el periodo anterior.

$$\Delta N_{UEs} = mN_{UEs}(n) - mN_{UEs}(n-1) \quad (4.12)$$

- $N_{PA}$ : Representa el número de preámbulos asignados en el estado anterior. Esta variable es utilizada dentro del resumen de un estado debido a que influye en el número de UEs totales que solicitan acceso.

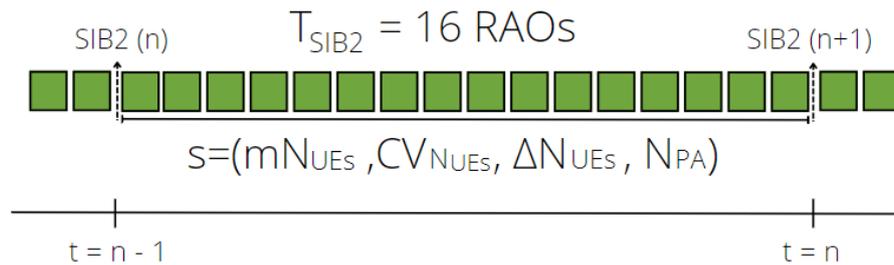


Figura 4.10: Definición de estado para *Q-learning*

Es relevante tener en consideración que en un RAO, el número máximo de preámbulos decodificados correctamente es de 54. Sin embargo, debido a problemas de conectividad y la posibilidad de que distintas UEs elijan el mismo preámbulo, la cantidad real es considerablemente menor.

En la Figura 4.11 se presentan gráficas que representan la probabilidad de acceso en función del número de UEs que solicitan acceso a la red. Para la generación de esta gráfica se realizaron simulaciones aleatorias considerando diferentes asignaciones de preámbulos (6, 12, 17, 28, 39, 45 y 52 preámbulos) para diferentes cantidades de UEs involucrados en el proceso de acceso, sin tener en cuenta la degradación del canal.

En cada simulación, cada UE seleccionó aleatoriamente un número de preámbulos de entre las asignaciones disponibles. Se contabilizó si ocurrieron colisiones, es decir, si varios UEs eligieron el mismo preámbulo al mismo tiempo en todas las simulaciones. Con los datos de colisiones obtenidos de múltiples simulaciones, se calculó la probabilidad de acceso para cada configuración de asignación de preámbulos y número de UEs. En la Figura 4.11 se puede observar que cuando el número de UEs es inferior a 30, existe una alta probabilidad (mayor a 0,7) de evitar la selección de los mismos preámbulos durante el mismo RAO, en el caso de asignar 52 preámbulos. Sin embargo, a medida que aumenta el número de UEs, el número de preámbulos que evitan colisionar disminuirá considerablemente. Es importante tener en cuenta que esta reducción se acentúa aún más cuando se considera la degradación del canal.

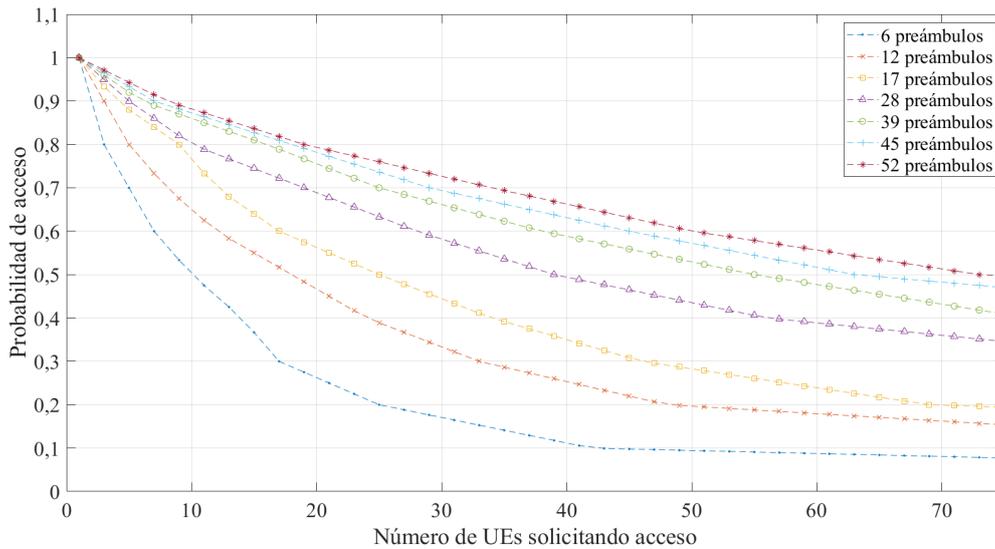


Figura 4.11: Probabilidad de acceso exitoso de UEs en función de la asignación de preámbulos.

De esta manera, mediante la simulación se concluye que incluso al asignar el máximo número posible de preámbulos por segmento (52 preámbulos), es poco probable tener una alta probabilidad de evitar colisiones de preámbulos cuando hay más de 30 UEs solicitando acceso en el mismo RAO. Por lo tanto, en nuestro sistema, los estados en los que el número de UEs es mayor a 30 se agrupan en la categoría de  $N_{UEs} = 30$  y se consideran el peor escenario posible, ya que en estos casos de alto tráfico no se puede garantizar un bajo número de colisiones. Para reducir el número de estados en el modelo de *Q-learning*, se tomaron las 30 posibilidades y se dividieron en 9 grupos en función de la cantidad de UEs presentes en cada grupo. El primer grupo representa una situación con un nivel bajo de tráfico (0 UEs por RAO), mientras que el grupo 9 corresponde a la situación con la mayor cantidad de tráfico (26 a 30 UEs por RAO). Esta estrategia permite reducir el número de estados al agrupar conjuntos de UEs similares en rangos específicos. Al hacerlo, el modelo de *Q-learning* se simplifica y se vuelve más manejable.

El parámetro  $CV_{N_{UEs}}$  comprende una alta cantidad de valores de varianza entre 0 y 1 debido a la posibilidad de tener un número extremadamente grande de valores decimales en ese rango. Para discretizar los valores, se clasifican en dos categorías: Bajo=1 y Alto=2, donde la asignación específica varía según  $mN_{UEs}$  (ver Tabla 4.4).

Se considera una varianza alta a todos los valores mayores a 0,4 cuando el tráfico es inferior a 15 UEs. Sin embargo, cuando se tiene entre 15 y 30 UEs, se considera una varianza alta si excede 0,2. De esta manera, se le otorga una mayor importancia a la varianza en casos de tráfico alto.

Tabla 4.4: Definición del parámetro  $CV_{NU_{Es}}$  en función de  $mNU_{Es}$ .

	$mNU_{Es} \leq 15$	$15 < mNU_{Es} \leq 30$
$CV_{NU_{Es}} = Alto$	$CV_{NU_{Es}} > 0,4$	$CV_{NU_{Es}} > 0,2$
$CV_{NU_{Es}} = Bajo$	$CV_{NU_{Es}} \leq 0,4$	$CV_{NU_{Es}} \leq 0,2$

En cuanto a  $\Delta N_{UEs}$ , también se discretiza los resultados para que solo tenga tres valores posibles. Es decir, cuando el tráfico observado aumenta,  $\Delta N_{UEs} = 1$ ; cuando disminuye,  $\Delta N_{UEs} = 2$ ; y cuando se mantiene constante,  $\Delta N_{UEs} = 3$ . Asimismo, para el parámetro  $N_{PA}$  se estableció una división en nueve grupos según la cantidad de preámbulos asignados. Cada grupo está definido de la siguiente manera:

- Grupo 1: De 1 a 3 preámbulos.
- Grupo 2: De 4 a 8 preámbulos.
- Grupo 3: De 9 a 14 preámbulos.
- Grupo 4: De 15 a 18 preámbulos.
- Grupo 5: De 19 a 24 preámbulos.
- Grupo 6: De 25 a 30 preámbulos.
- Grupo 7: De 31 a 38 preámbulos.
- Grupo 8: De 39 a 49 preámbulos.
- Grupo 9: De 50 a 52 preámbulos.

Basándonos en lo anterior, se tiene un total de 486 estados posibles, donde los diferentes valores que cada componente de un estado puede tomar, se muestran en la Tabla 4.5. En caso de no haber realizado la subdivisión de los parámetros  $mNU_{Es}$  y  $N_{PA}$  en 9 grupos, se habría generado un total de 9360 estados. Este incremento habría resultado en una tabla Q de dimensiones 9360x52, es decir, 19 veces más extensa. Como se puede apreciar, cuando se incrementa el número de estados en el

entorno, se hace necesario agregar una fila adicional en la tabla  $Q$  para cada uno de estos estados. Este aumento en la cantidad de estados no solo implica una mayor demanda de memoria para almacenar la tabla  $Q$ , sino que también puede prolongar el tiempo necesario para aprender los valores  $Q$  óptimos. Esto se debe a que el agente debe explorar y actualizar los valores de calidad para un mayor número de estados y acciones, lo que puede resultar en un proceso de entrenamiento más prolongado. La discretización de  $CV_{NUEs}$  y  $\Delta N_{UEs}$  nos permite reducir el espacio de estados y acciones a un conjunto finito y manejable de opciones, así como también hace que los resultados del algoritmo sean más interpretables y significativos para los usuarios o desarrolladores.

Tabla 4.5: Composición de estados en Q-Learning con parámetros y valores posibles.

Parámetro	Conjunto de valores posibles
$mN_{UEs}$	Grupo 1, Grupo 2, Grupo 3, Grupo 4, Grupo 5, Grupo 6, Grupo 7, Grupo 8 y Grupo 9
$CV_{NUEs}$	Alto y Bajo
$\Delta N_{UEs}$	Aumenta, Disminuye y Mantiene
$N_{PA}$	Grupo 1, Grupo 2, Grupo 3, Grupo 4, Grupo 5, Grupo 6, Grupo 7, Grupo 8 y Grupo 9

#### 4.7.1.3. Recompensa

En cuanto a la recompensa, esta se define como una función de las variables  $mN_{UEs}$ ,  $CV_{NUEs}$ ,  $\Delta N_{UEs}$  y  $N_{PA}$ . Los valores de recompensa o penalización se encuentran en un rango que va desde -100 hasta 100, con incrementos de tamaño de 10 para facilitar su evaluación.

Para establecer los criterios de recompensa y penalización, se implementó un procedimiento manual que involucra la evaluación de cada posible estado en función de las cuatro variables que lo componen. Se procedió a la identificación de patrones en función de esta evaluación, lo que condujo a la creación de siete reglas específicas para la asignación de recompensas y penalizaciones. Con base en estas reglas, se evaluaron todos los estados posibles, un total de 486, y se les asignó la recompensa correspondiente. Estos pares de estado-recompensa se almacenaron en matrices a las cuales el algoritmo recurre para llevar a cabo la actualización de la tabla  $Q$ . A continuación, se enumeran las mencionadas reglas y se detalla la concepción de cada una de ellas.

1. La primera regla penaliza fuertemente la asignación mínima de preámbulos (de 1 a 3 preámbulos) en los peores casos, es decir, cuando el tráfico de usuarios es cercano a 30. Esta penalización se justifica porque en situaciones de alta congestión, asignar muy pocos preámbulos resulta en una alta probabilidad de colisiones durante el acceso al canal de comunicación.
2. La segunda regla impone una penalización cercana a -100 cuando se asigna el máximo de preámbulos (de 50 a 52 preámbulos) en condiciones de mínimo tráfico de usuarios. En esta situación, el agente aprende que asignar más preámbulos de los necesarios cuando la carga de usuarios es baja no es eficiente y conduce a un desperdicio de recursos, en términos de preámbulos no utilizados que podrían haber sido asignados a otros servicios que los necesitaban.
3. La tercera regla asigna una recompensa negativa (penalización) moderada cuando se dan justo el número de preámbulos iguales a  $mN_{UEs}$ . La asignación de preámbulos idénticos al número de UEs conduce a una alta probabilidad de que dos o más UEs seleccionen el mismo preámbulo, lo que resulta en una colisión durante el proceso de acceso al canal de comunicación.
4. La cuarta regla otorga una recompensa positiva a las asignaciones que proporcionen aproximadamente el doble de preámbulos en comparación con el número de UEs en el sistema. Esta regla se respalda con el análisis de la gráfica presentada en la Figura 4.11, donde se observa la relación entre el número de UEs solicitando acceso y la cantidad de preámbulos asignados. Se evidencia que se puede igualar o superar el umbral de 0,7 en la probabilidad de acceso cuando se cumple esta regla. Por ejemplo, cuando hay 20 UEs y se asignan 39 preámbulos, se logra alcanzar una probabilidad de acceso de 0,75. Es importante recalcar que en el contexto de este estudio, se estableció que una probabilidad de acceso igual o superior al 0,7 se considera un umbral aceptable.
5. La quinta regla establece una recompensa positiva de pequeña magnitud cuando se asigna la cantidad máxima de preámbulos en situaciones de alto tráfico, es decir, cuando se registran más de 20 usuarios en un mismo segmento. La justificación de esta regla se basa en el hecho de que, si bien la asignación de los 52 preámbulos garantiza una probabilidad de acceso superior a 0,7 en ese

segmento, esto conlleva problemas de ineficiencia y desequilibrio en la asignación de preámbulos a los otros dos segmentos que en esta situación disponen únicamente de 1 preámbulo cada uno para su utilización.

6. La sexta regla asigna una recompensa menor cuando el estado del sistema indica que el número de UEs está aumentando ( $\Delta N_{UEs} = 1$ ) en comparación con situaciones en las que se mantiene ( $\Delta N_{UEs} = 3$ ) o disminuye ( $\Delta N_{UEs} = 2$ ) el número de usuarios. Esta regla se fundamenta en la necesidad de incrementar la asignación de preámbulos para acomodar la creciente demanda de usuarios.
7. La séptima regla asocia una recompensa menor cuando el estado del sistema refleja una alta variabilidad ( $CVN_{UEs} = 2$ ) en comparación con situaciones de baja variabilidad ( $CVN_{UEs} = 1$ ). Esta regla se basa en la premisa de que una varianza alta indica una distribución desigual de UEs en el período de 16 RAOs, con un número de usuarios predominantemente mucho menor que el valor de referencia  $mN_{UEs}$ . En tales casos, se permite la reducción de la asignación de preámbulos. Por otro lado, cuando la varianza es baja, se sugiere una distribución más uniforme de UEs, generalmente cercana al valor máximo  $mN_{UEs}$ , por lo que se asigna una recompensa más alta que deriva en una mayor asignación de preámbulos.

La Tabla 4.6 presenta algunos ejemplos de combinaciones de estados y recompensas del sistema. Se muestran solo unos pocos ejemplos debido a la gran cantidad de posibles combinaciones. Sin embargo, estos ejemplos nos dan una idea de la asignación de las recompensas y penalizaciones según las reglas descritas anteriormente.

graphicx

Tabla 4.6: Ejemplos de combinaciones de estados y recompensas del algoritmo *Q-learning*.

Estado				Recompensa
<b>Regla 1</b>				
$26 \leq mN_{UEs} \leq 30$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 1$	$N_{PA} \leq 3$	-100
$26 \leq mN_{UEs} \leq 30$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 3$	$N_{PA} \leq 3$	-100
<b>Regla 2</b>				
$mN_{UEs} \leq 1$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 1$	$50 \leq N_{PA} \leq 52$	-80
$mN_{UEs} \leq 1$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 1$	$50 \leq N_{PA} \leq 52$	-90
<b>Regla 3</b>				
$15 \leq mN_{UEs} \leq 19$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 1$	$15 \leq N_{PA} \leq 18$	-50
$4 \leq mN_{UEs} \leq 7$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 3$	$4 \leq N_{PA} \leq 8$	-30
<b>Regla 4</b>				
$8 \leq mN_{UEs} \leq 11$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 3$	$19 \leq N_{PA} \leq 24$	60
$8 \leq mN_{UEs} \leq 11$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 3$	$25 \leq N_{PA} \leq 30$	30
<b>Regla 5</b>				
$20 \leq mN_{UEs} \leq 25$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 2$	$50 \leq N_{PA} \leq 52$	10
$20 \leq mN_{UEs} \leq 25$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 1$	$50 \leq N_{PA} \leq 52$	20
<b>Regla 6</b>				
$4 \leq mN_{UEs} \leq 7$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 2$	$9 \leq N_{PA} \leq 14$	-10
$4 \leq mN_{UEs} \leq 7$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 1$	$9 \leq N_{PA} \leq 14$	-30
<b>Regla 7</b>				
$12 \leq mN_{UEs} \leq 14$	$CV_{N_{UEs}} = 1$	$\Delta N_{UEs} = 2$	$31 \leq N_{PA} \leq 38$	20
$12 \leq mN_{UEs} \leq 14$	$CV_{N_{UEs}} = 2$	$\Delta N_{UEs} = 2$	$31 \leq N_{PA} \leq 38$	10

#### 4.7.2. Implementación del algoritmo Q-Learning

La implementación del algoritmo *Q-learning* es esencial para comprender cómo este enfoque de aprendizaje reforzado opera en la toma de decisiones en un entorno dado. Para lograr una comprensión más visual y coherente de su funcionamiento, se utiliza un diagrama de flujo que destaca los pasos clave involucrados en el proceso. A través de este diagrama, se desglosan las etapas desde la inicialización de la tabla *Q* hasta la iteración a lo largo del entorno, la actualización de los valores *Q* y la mejora gradual de las decisiones tomadas. La Figura 4.12 representa gráficamente cómo el algoritmo *Q-learning* interactúa con el entorno y cómo los valores *Q* se ajustan en función de las recompensas obtenidas.

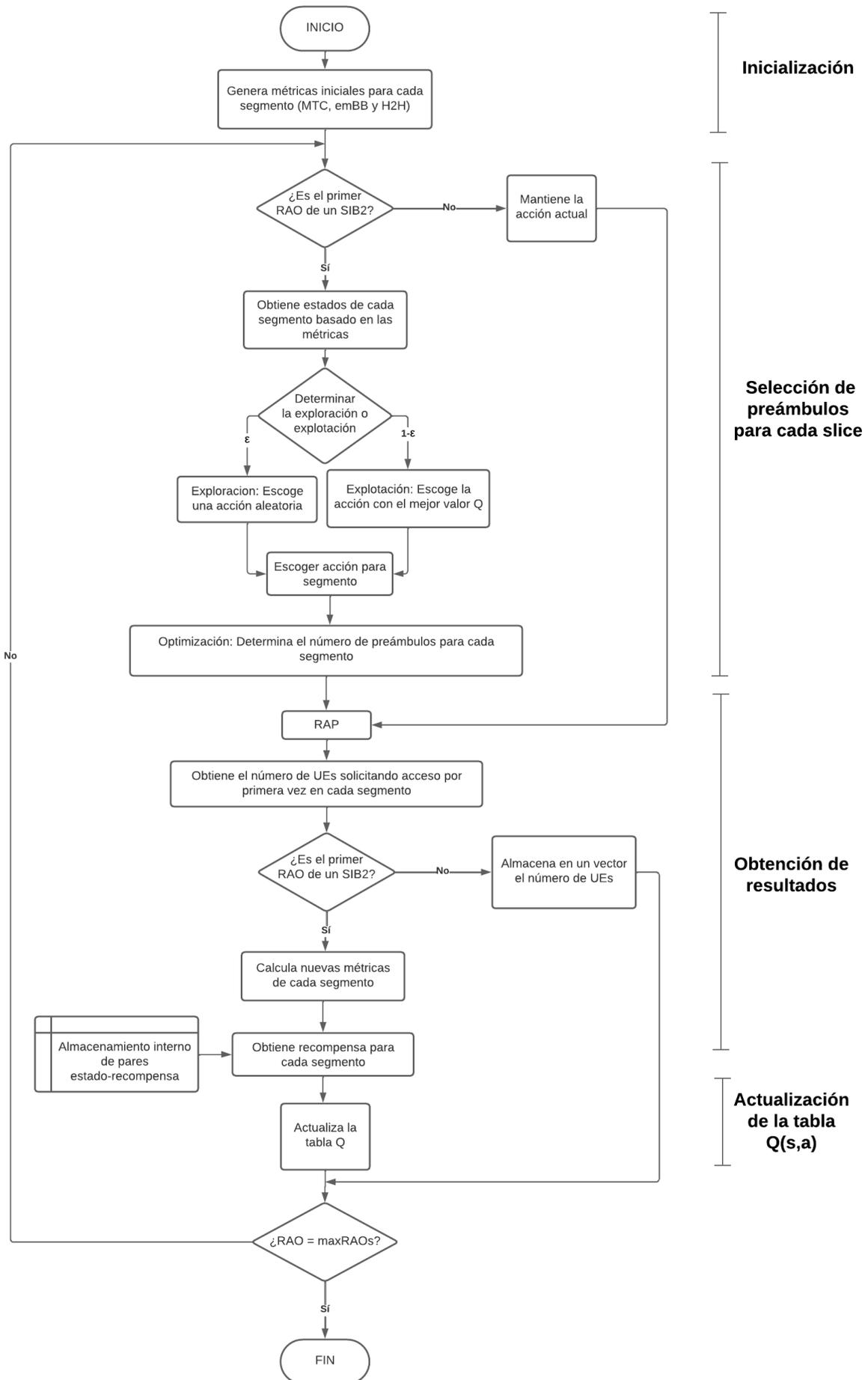


Figura 4.12: Diagrama de flujo de la implementación de Q-learning.

#### 4.7.2.1. Inicialización

Como primer paso, se crea una tabla  $Q(s, a)$  de tamaño 486 estados  $\times$  52 acciones y se llena sus atributos de 0. Según [85] al iniciar con ceros, el algoritmo no tiene prejuicios hacia ninguna acción en particular (exploración inicial neutra). Esto permite que el agente explore inicialmente todas las acciones de manera equitativa, lo que resulta beneficioso en la búsqueda de una política óptima.

Además de la inicialización de la tabla  $Q$ , se genera métricas iniciales para un estado y una acción en los tres segmentos del sistema. Estas métricas proporcionan una base para el proceso de aprendizaje y sirven como punto de partida para que el algoritmo *Q-learning* comience a iterar y mejorar sus estimaciones.

En esta etapa, también se da la configuración del valor inicial de  $\epsilon$  a 1, que representa la probabilidad de realizar una acción aleatoria en lugar de seguir una política aprendida. El valor de  $\epsilon$  se reducirá gradualmente después del entrenamiento, disminuyendo en incrementos de 0.01. Esta reducción gradual permitirá al algoritmo explorar diferentes acciones en las primeras etapas y luego explotar el conocimiento aprendido a medida que avanza el proceso.

Asimismo, se establece el factor de descuento  $\gamma$  y la tasa de aprendizaje  $\alpha$ . El factor de descuento determina el valor de las recompensas futuras en comparación con las recompensas inmediatas. Un  $\gamma$  más alto indica una mayor importancia de las recompensas futuras, mientras que un  $\gamma$  más bajo prioriza las recompensas inmediatas. Por otro lado, la tasa de aprendizaje influye en la rapidez con la que el algoritmo actualiza sus estimaciones en función de las recompensas recibidas. Una tasa de aprendizaje alta significa que las actualizaciones son más rápidas, pero también pueden generar una convergencia más errática, mientras que una tasa de aprendizaje baja hace que las actualizaciones sean más estables pero más lentas.

#### 4.7.2.2. Selección de preámbulos para cada segmento

Si se está dentro de los 16 RAOs de un mismo SIB2, se mantendrá la acción actual. Sin embargo, si es el primer RAO de un nuevo SIB2, se procederá a calcular el nuevo estado considerando el entorno y, posteriormente, se determinará la acción

correspondiente utilizando el método epsilon-greedy. Epsilon-greedy es una estrategia comúnmente utilizada para equilibrar la exploración y la explotación en algoritmos de aprendizaje por refuerzo, como *Q-Learning* [86]. El parámetro  $\epsilon$  se establece en el rango de  $[0, 1]$  y representa la probabilidad de que el agente escoja una acción desconocida de forma aleatoria (exploración) al tomar decisiones. La probabilidad de que el agente escoja la mejor acción conocida (explotación) se denota como  $1 - \epsilon$ , en este caso se elige la acción con el valor más alto en la tabla  $Q$ . En cada periodo de decisión, se genera un número aleatorio entre 0 y 1. Si el número aleatorio es menor o igual a  $\epsilon$ , el agente elige una acción al azar entre las 52 acciones posibles. De otra manera, si el número aleatorio es mayor que  $\epsilon$ , el agente selecciona la acción con el valor  $Q$  más alto para el estado actual.

La idea detrás de colocar el valor máximo a  $\epsilon$  ( $\epsilon = 1$ ) al comienzo del aprendizaje es aumentar la exploración de tal forma que el agente pueda descubrir una amplia variedad de estados y acciones posibles en el entorno. Esta alta probabilidad de exploración permite probar acciones al azar y obtener información acerca del comportamiento del sistema en estas situaciones. Esto es importante debido a que en esta etapa la tabla  $Q$  se encuentra vacía y con valores inciertos. Conforme se realiza más actualizaciones de la tabla  $Q$ , el valor de  $\epsilon$  va disminuyendo para aprovechar la información acumulada durante la exploración y tomar decisiones de forma óptima.

Después de definir la acción sobre el número de preámbulos a asignar a cada segmento, es esencial contar con una función de optimización que considere los tres valores en conjunto. El objetivo de esta función será maximizar la probabilidad de acceso a la red, asegurando una asignación adecuada de preámbulos para los tres segmentos. La función de optimización fue realizada mediante la herramienta de optimización `fmincon` de MATLAB. A continuación se detalla los parámetros involucrados en el procedimiento de la función de optimización.

- Datos de entrada: Los parámetros de entrada que se utilizan para construir la función objetivo en el proceso de optimización son el número de preámbulos asignados para los servicios mMTC, emBB y H2H, el número de UEs que solicitan acceso y el número de accesos exitosos para cada tipo de servicio.
- Definición de la función objetivo: La función objetivo de este problema de op-

timización debe relacionar los valores de preámbulos asignados ( $N_{preambulos}$ ), el número de UEs solicitando acceso ( $N_{UEs}$ ) y el número de accesos exitosos ( $N_{accesos\ exitosos}$ ). La fórmula que describe la probabilidad de acceso exitoso debido a la asignación de preámbulos se describe en la Ecuación 4.13

$$probabilidad\_exito = \frac{N_{accesos\ exitosos}}{N_{UEs} \cdot N_{preambulos}} \quad (4.13)$$

El objetivo del proceso de optimización es maximizar la suma de la probabilidad de éxito de los diferentes tipos de servicios (mMTC, emBB y H2H). En el Extracto de código 4.3 se presenta la definición de la función objetivo para el problema de optimización.

Extracto de código 4.3: Función objetivo para el proceso de optimización en MATLAB

```
f_objetivo=@(x)-(N_accesos_exitosos_mMTC/N_UEsM*x(1) +
                N_accesos_exitosos_emBB/N_UEsB*x(2) +
                N_accesos_exitosos_H2H/N_UEsH*x(3) )
```

- Restricciones lineales: Se define una restricción lineal que asegura que la suma de preámbulos asignados no supere un valor dado (54 preámbulos):  $x(1)+x(2)+x(3)=54$ . El Extracto de código 4.4 se muestra la configuración de la restricción lineal implementada.

Extracto de código 4.4: Restricción lineal para el proceso de optimización en MATLAB

```
Aeq = [1, 1, 1];    % Coeficientes restricción lineal
beq = 54;           % Igualdad de restricción lineal
```

- Límites de las variables de asignación de preámbulos. Los límites inferiores (lb) son los valores mínimos permitidos para cada servicio y corresponden al número de preámbulos asignados por *Q-learning*. Los límites superiores (ub) se establecen en 52 para cada servicio (para garantizar la asignación mínima de un preámbulo a un servicio individual no se puede asignar más de 52 preámbulos). **Nota:** En caso de la suma de lb supere a 54, se reducen los valores

de lb proporcionalmente de acuerdo con prioridades asignadas a cada tipo de servicio. Además, se asegura que no haya asignaciones de preámbulos menores o iguales a cero. A continuación, se ha proporcionado el extracto de código 4.5 que establece los límites de las variables de asignación de preámbulos de acuerdo con lo especificado anteriormente.

Extracto de código 4.5: Límites de las variables de asignación de preámbulos para el proceso de optimización en MATLAB

```
N_preambulos=[N_preambulosbulos_mMTC ,
               N_preambulosbulos_emBB ,
               N_preambulosbulos_H2H];

suma_asignacion = sum(N_preambulos);

if suma_asignacion > 54
    exceso = suma_asignacion - 54;

    % Define las prioridades (mMTC > emBB > H2H)
    prioridad_mMTC = 3; % Prioridad mas alta para mMTC
    prioridad_emBB = 2; % Prioridad intermedia para emBB
    prioridad_H2H = 1; % Prioridad mas baja para H2H
    suma_prioridades=prioridad_MTC+prioridad_emBB+
                    prioridad_H2H;

    N_preambulos(1) = N_preambulos(1)-(prioridad_mMTC*
        N_preambulos(1)/suma_prioridades) * exceso;
    N_preambulos(2) = N_preambulos(2)-(prioridad_emBB*
        N_preambulos(2)/suma_prioridades) * exceso;
    N_preambulos(3) = N_preambulos(3)-(prioridad_H2H*
        N_preambulos(3)/suma_prioridades) * exceso;
end

N_preambulos(N_preambulos <= 0) = 1;
```

```
% Limites de las variables de asignacion de preambulos  
lb = [N_preambulos(1), N_preambulos(2), N_preambulos(3)];  
ub = [52, 52, 52];
```

#### 4.7.2.3. Obtención de resultados

Una vez que se ha establecido el número de preámbulos para cada tipo de tráfico (mMTC, emBB y H2H), se procede a la asignación de recursos mediante el proceso de acceso aleatorio. Durante este procedimiento, se registran los resultados de la cantidad de UEs que han solicitado acceso por primera vez en cada RAO. Si el RAO está contenido dentro de un SIB2, los resultados se almacenan; de lo contrario, se calculan nuevas métricas para el estado actual de cada segmento.

#### 4.7.2.4. Actualización de la tabla $Q(s, a)$

Con base en estas métricas obtenidas anteriormente, el algoritmo determina la recompensa correspondiente utilizando la información almacenada internamente de los pares de estado-recompensa, lo que habilita el cálculo de los valores de estado-acción para la tabla  $Q$ .

## Capítulo 5 Análisis y resultados

En este capítulo se analizan los resultados obtenidos de los algoritmos ML implementados. Cada sección de este capítulo aborda un aspecto específico de la investigación. En la Sección 5.1 se identifican los KPI a utilizar para evaluar el desempeño de la red con los algoritmos planteados. Se explorarán métricas clave como la probabilidad de acceso, número de colisiones, número de transmisiones de preámbulos, número de preámbulos codificados exitosamente, número de accesos exitosos y promedio de retransmisiones. A continuación en la Sección 5.2 se examina diferentes arquitecturas para el diseño de la RNA, variando valores de capas ocultas, neuronas en capas ocultas y funciones de activación. Posteriormente, en la Sección 5.3 se realizan pruebas con diferentes valores para los hiperparámetros: factor de aprendizaje, factor de descuento y tiempo de entrenamiento; con el objetivo de encontrar valores que maximicen la asignación de preámbulos de *Q-learning*. Finalmente, en la Sección 5.4 se realiza un análisis comparativo entre RNA y *Q-learning*, evaluando su eficiencia y adaptabilidad a diferentes escenarios. Se analizan escenarios de máxima demanda así como también en escenarios múltiples.

### 5.1. Indicadores de rendimiento

Los KPI son medidas cuantitativas que se utilizan para evaluar el rendimiento del RACH durante la simulación de la administración dinámica de preámbulos. Estos indicadores permitirán una evaluación cuantitativa del funcionamiento de los algoritmos propuestos. A continuación se describe los indicadores a utilizar:

- Probabilidad de acceso exitoso  $P_s$ :** Se define como la probabilidad de completar con éxito el RAP dentro del número máximo de retransmisiones definidas por el canal [75]. La probabilidad de acceso exitoso se define en la Ecuación 5.1.

$$P_s = \frac{\text{ArribosExitosos}}{\text{ArribosTotales}} \quad (5.1)$$

- Estadísticas del número de colisiones:** Definida como el número de colisiones existentes a lo largo de un periodo  $T$ .

- **Estadísticas del número de transmisiones de Msg1:** Definida como el de número de transmisiones existentes a lo largo de un periodo  $T$ .
- **Estadísticas del número de accesos exitosos:** Definida como el número de accesos exitosos existentes a lo largo de un periodo  $T$ .
- **Estadísticas del número de preámbulos decodificados correctamente:** Definida como el número de preámbulos sin colisionar que fueron decodificados correctamente a lo largo de un periodo  $T$ .
- **Promedio de retransmisiones:** Hace referencia al valor promedio de las retransmisiones dentro de un periodo de envío de mensajes SIB2.

## 5.2. Análisis de arquitecturas para la red neuronal artificial

Para determinar la arquitectura óptima de la red neuronal que mejor se adapte a los requisitos de este estudio, se llevó a cabo un análisis comparativo de 4 arquitecturas con diferentes configuraciones. Cada arquitectura consiste en un número específico de capas ocultas y neuronas en cada capa, junto con funciones de activación particulares. Las siguientes arquitecturas de redes neuronales fueron consideradas:

- **Arquitectura 1:**
  - Capas Ocultas: 3.
  - Neuronas en Capas Ocultas: 4, 6, 1.
  - Funciones de activación: Logsig, Logsig, Tansig, Purelin.
- **Arquitectura 2:**
  - Capas Ocultas: 2.
  - Neuronas en Capas Ocultas: 4, 1.
  - Funciones de activación: Logsig, Tansig, Purelin.
- **Arquitectura 3:**
  - Capas Ocultas: 3.

- Neuronas en Capas Ocultas: 12, 5, 1.
  - Funciones de activación: Logsig, Logsig, Tansig, Purelin.
- **Arquitectura 4:**
- Capas Ocultas: 6.
  - Neuronas en Capas Ocultas: 4, 1, 5, 2, 4, 1.
  - Funciones de activación: Logsig, Logsig, Tansig, Tansig, Tansig, Tansig, Purelin.

Estas arquitecturas fueron elegidas con el propósito de explorar diversas configuraciones de capas y neuronas, así como diferentes combinaciones de funciones de activación (ver Sección 4.6.3). Después de realizar el entrenamiento de la red se obtuvieron valores de época, tiempo de entrenamiento, rendimiento y gradiente. Estos parámetros se encuentran resumidos en la Tabla 5.1. Los valores iniciales indican el valor inicial de la variable o parámetro antes de iniciar el proceso de entrenamiento. Este valor se establece de forma predeterminada por la RNA. Los valores finales representan el valor final de la misma variable o parámetro después de completar el entrenamiento de la red neuronal.

Tabla 5.1: Comparación de parámetros de entrenamiento entre las diferentes arquitecturas de la red neuronal.

Variable	Arquitectura 1		Arquitectura 2		Arquitectura 3		Arquitectura 4	
	Valor inicial	Valor Fin						
Época	0	10000	0	10000	0	10000	0	10000
Tiempo de simulación	-	0:00:14	-	0:00:11	-	0:00:14	-	0:00:17
Rendimiento	219	2,64	385	3,74	447	2,02	644	98,8
Gradiente	137	6,56	538	0,621	602	0,537	1,21E+03	4,05

Los resultados en la Tabla 5.1 representan los resultados del entrenamiento de la red neuronal y se puede inferir lo siguiente:

- **Época:** Este valor indica cuántas épocas se ejecutó el entrenamiento antes de detenerse. Las arquitecturas 1, 2, 3 y 4 se entrenaron durante 10,000 épocas cada una.
- **Rendimiento:** Esta métrica representa el error de entrenamiento final. Cuanto menor sea el valor, mejor será el rendimiento de la red. La Arquitectura 3 tiene el

valor más bajo de 2.02, lo que sugiere que tiene el menor error de entrenamiento entre todas las arquitecturas.

- **Gradiente:** Indica a cómo cambian los pesos de la red durante el entrenamiento. Un gradiente más bajo es deseable, ya que indica que el entrenamiento se ha estabilizado. La Arquitectura 3 tiene el valor más bajo de 0.537, lo que sugiere que ha alcanzado una convergencia más rápida y estable.

Es esencial verificar que las arquitecturas diseñadas sean capaces de funcionar en línea con el escenario de asignación dinámica propuesto. Para ello, se someten las redes entrenadas a dos escenarios de prueba diferentes. El primer escenario representa un escenario de máximo esfuerzo, en el que cada segmento recibe un gran número de usuarios que excede las capacidades del canal de acceso. En este caso, se garantiza que un porcentaje de usuarios no podrá acceder exitosamente. El segundo escenario corresponde a un escenario de mínimo esfuerzo, en el que todos los usuarios tienen una probabilidad de acceso igual a 1, ya que no sobrepasa la capacidad del canal. Estos escenarios permiten evaluar la robustez y eficacia de las arquitecturas diseñadas en 2 situaciones extremas y opuestas, así como su capacidad para manejar la asignación dinámica de preámbulos bajo diferentes cargas de tráfico. Los valores de arribos para cada escenario de experimentación de la RNA se resumen en la Tabla 5.2.

Tabla 5.2: Arribos esperados para los escenarios de entrenamiento de la RNA.

<b>Escenario de Experimentación</b>	<b>UEsM</b>	<b>UEsB</b>	<b>UEsH</b>
Máximo esfuerzo	30000	1500	33000
Mínimo esfuerzo	5000	250	5000

Para evaluar el rendimiento de cada arquitectura, se tomarán en cuenta los indicadores de rendimiento que se detallaron en la Sección 5.1. En la Figura 5.1 se presentan los resultados obtenidos para el escenario de mínimo esfuerzo de las siguientes gráficas comparativas: (a) Número de transmisiones totales hacia la celda; (b) Número de colisiones totales que tuvo la celda durante el periodo de simulación; (c) Cantidad de preámbulos decodificados exitosamente; (d) Arribos exitosos que obtuvo la celda. Todas estas gráficas se presentan en función del n-ésimo SIB2. En primera instancia, se puede observar que todas las arquitecturas presentan la misma tendencia de

comportamiento, lo que sugiere que en situaciones de baja explotación del sistema, cualquiera de las arquitecturas podría ser funcional. Estos resultados resaltan la capacidad general de las arquitecturas para manejar escenarios de baja demanda, donde el sistema no se encuentra sometido a una carga significativa y las asignaciones de preámbulos se realizan con mayor eficacia.

Considerando que las gráficas de la Figura 5.1 muestran los indicadores de rendimiento en una comunicación mMTC, se observa una clara tendencia de campana. Esta tendencia se debe al modelo de distribución Beta que se describió en detalle en la Sección 4.4. Es relevante notar que esta tendencia provoca un pico de amplitud en el rango que abarca desde el quincuagésimo al sexagésimo SIB2.

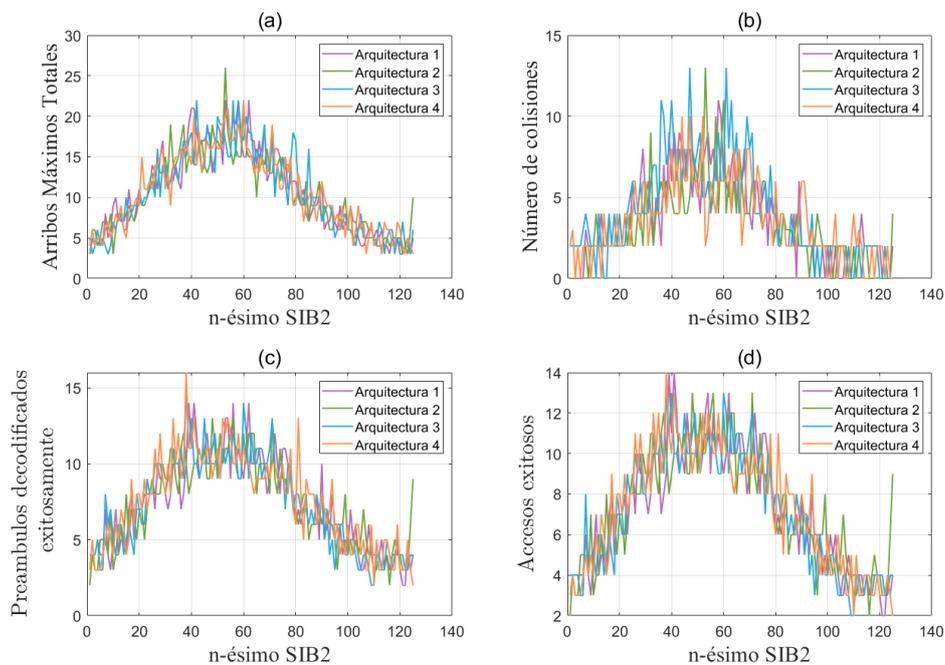


Figura 5.1: Resultados del escenario de mínimo esfuerzo para las diferentes arquitecturas RNA.

Además, en la Tabla 5.3 se presenta los valores promedios de la cantidad de preámbulos transmitidos para que un UE pueda tener un acceso exitoso y la probabilidad de acceso que se obtuvieron en la simulación. Como resultado, se destaca que las 4 arquitecturas tienen una tendencia similar, no obstante, la tercera arquitectura obtuvo un valor menor de transmisiones necesarias para obtener una concesión de UL. La probabilidad de acceso se calcula mediante la Ecuación (5.1).

Tabla 5.3: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de mínimo esfuerzo.

	Preambulos Tx				Probabilidad de Acceso exitoso			
	Arquitectura 1	Arquitectura 2	Arquitectura 3	Arquitectura 4	Arquitectura 1	Arquitectura 2	Arquitectura 3	Arquitectura 4
mMTC	1,6672	1,5109	1,6210	1,6384	1	1	1	1
emBB	1,5030	1,5330	1,6674	1,5386	1	1	1	1
H2H	1,4516	1,6406	1,5344	1,4512	1	1	1	1
TOTAL	1,6125	1,6033	1,4571	1,6031	1	1	1	1

Por otro lado, los resultados de la simulación del escenario de máximo esfuerzo muestran que las cuatro arquitecturas de red neuronal presentaron un comportamiento similar (ver Figura 5.2). Ante un escenario en el que existen problemas de bloqueo de acceso debido a las retransmisiones masivas, las cuatro arquitecturas operaron de manera coherente.

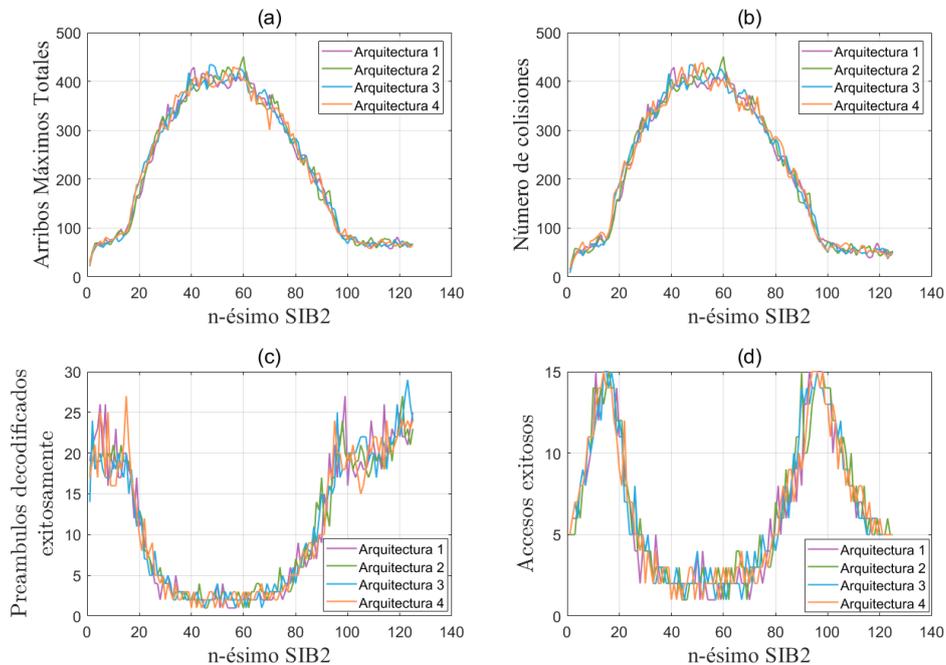


Figura 5.2: Resultados del escenario de máximo esfuerzo para las diferentes arquitecturas de la red neuronal.

De igual forma que en el escenario anterior se detalla en la Tabla 5.4 los valores de transmisiones necesarias para que un UE pueda tener un acceso exitoso y la probabilidad de acceso. Los resultados presentados indican una alta transmisión de mensajes y una probabilidad de acceso muy baja para las comunicaciones mMTC y emBB. Una alta transmisión de mensajes involucra a su vez que el número de colisiones también es alto.

Tabla 5.4: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de máximo esfuerzo.

	Preambulos Tx				Probabilidad de Acceso exitoso			
	Arquitectura 1	Arquitectura 2	Arquitectura 3	Arquitectura 4	Arquitectura 1	Arquitectura 2	Arquitectura 3	Arquitectura 4
<b>mMTC</b>	1,7583	3,5957	3,4984	3,7340	0,1133	0,1240	0,1285	0,1191
<b>emBB</b>	5,2722	5,1462	5,1436	5,1552	0,2900	0,2679	0,2740	0,2682
<b>H2H</b>	3,6218	2,1387	2,1859	2,0141	1	0,9983	0,9964	0,9982
<b>TOTAL</b>	4,3760	4,2403	4,2068	4,3158	0,1816	0,1824	0,1870	0,1791

En función de las métricas de entrenamiento, la Arquitectura 3 es una mejor opción, ya que tiene el menor error de entrenamiento y el gradiente más bajo. Además, esta arquitectura mostró un mejor desempeño al ser sometida a la prueba del escenario de máximo esfuerzo. Estos factores en conjunto indican que la Arquitectura 3 es la más adecuada para el propósito de la asignación dinámica de preámbulos, dadas sus características de rendimiento y adaptabilidad.

Para destacar la aplicación de la Arquitectura 3 en este estudio, se presentan las Figuras 5.3 y 5.4, que comparan la implementación de la Arquitectura de RNA seleccionada con el uso de las Ecuaciones (4.8) y (4.9), que representan funciones deterministas. Estas comparaciones se llevan a cabo en el escenario de máximo esfuerzo detallado en la tabla 5.2. Es importante recordar que las Ecuaciones (4.8) y (4.9) fueron diseñadas bajo la premisa de un escenario ideal, en el cual las transmisiones resultantes de colisiones o decodificaciones erróneas no generan nuevos intentos de transmisión, es decir, un UE dispone de una única oportunidad para acceder al servicio. En contraste, el conjunto de datos utilizado para entrenar la RNA considera un número máximo de intentos de acceso, que asciende a 10.

En el segmento mMTC, se puede observar en las Figuras 5.3(a) y 5.3(b) una clara diferencia entre la función determinista y la RNA. La función determinista muestra un promedio más alto de arribos, lo que se traduce en un mayor número de colisiones. Por otro lado, la RNA logra reducir tanto la cantidad de arribos como las colisiones, lo que resulta en una mayor cantidad de preámbulos decodificados y, por lo tanto, un aumento en el número de accesos, como se aprecia en las Figuras 5.3(c) y 5.3(d).

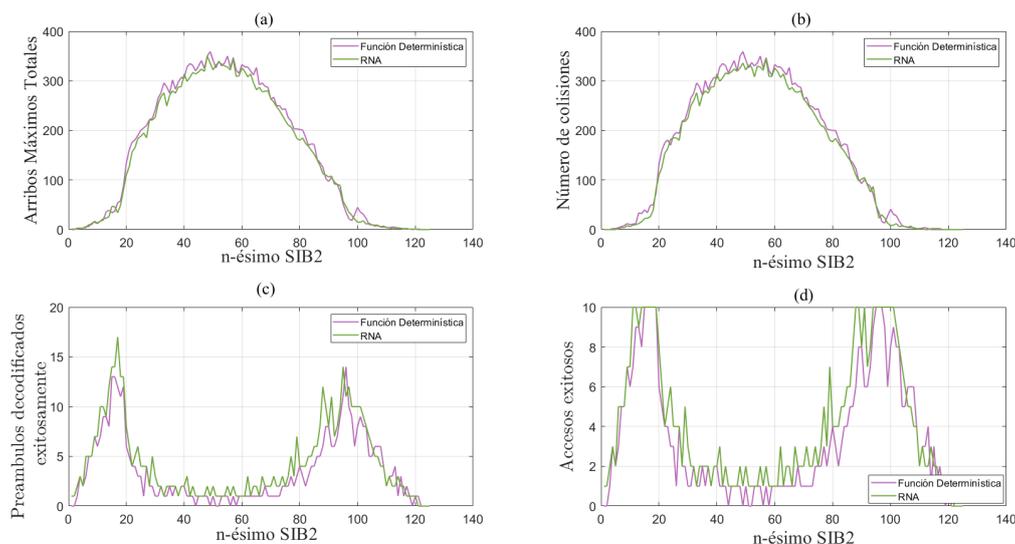


Figura 5.3: Gráficas comparativas entre la aplicación de la RNA y la función determinista para el segmento mMTC en un escenario de máximo esfuerzo

En el segmento emBB, las Figuras 5.4(a) y 5.4(b) ilustran cómo la implementación de la RNA reduce significativamente tanto el número de colisiones como la cantidad máxima de arribos por SIB2. Estos resultados sugieren una correlación clara: a medida que aumentan los arribos por SIB2, también aumenta la probabilidad de colisiones. Por el contrario, cuando la cantidad de arribos es menor, se incrementa la posibilidad de decodificar correctamente un preámbulo sin experimentar colisiones.

La Tabla 5.5 presenta una comparación de los valores obtenidos para la cantidad de preámbulos transmitidos y la probabilidad de acceso entre la RNA y la función determinista. En promedio, los resultados muestran que la RNA supera a la función determinista, logrando una reducción en la cantidad de transmisiones. Estos hallazgos respaldan la eficacia de la RNA en términos de optimización de recursos y mejora en la probabilidad de acceso.

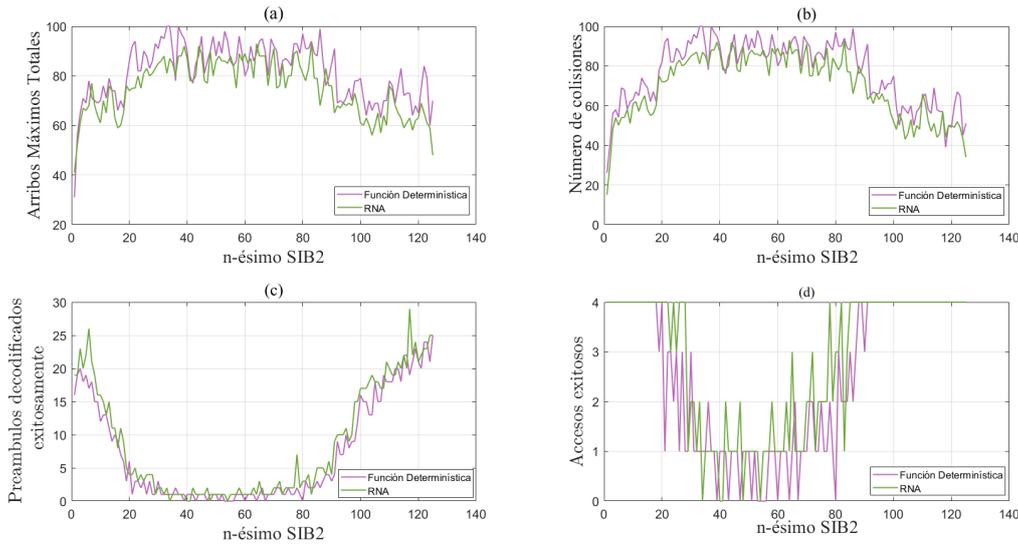


Figura 5.4: Gráficas comparativas entre la aplicación de la RNA y la función determinista para el segmento emBB en un escenario de máximo esfuerzo

Tabla 5.5: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida en el escenario de máximo esfuerzo

	Preámbulos Transmitidos		Probabilidad de acceso exitoso	
	Función determinista	RNA	Función determinista	RNA
<b>mMTC</b>	4,05	3,4984	0,1267	0,1285
<b>emBB</b>	5,1937	5,1436	0,2582	0,2740
<b>H2H</b>	3,5493	2,1859	0,672	0,9964
<b>TOTAL</b>	4,4536	4,2068	0,1760	0,1870

### 5.3. Análisis de hiperparámetros óptimos para Q-learning

La selección adecuada de hiperparámetros es un aspecto fundamental para garantizar el éxito en la implementación del algoritmo *Q-learning*. Estos hiperparámetros desempeñan un papel crucial al influir en la forma en que el agente aprende y ajusta su estrategia de toma de decisiones a lo largo del tiempo. Entre los hiperparámetros clave que se evaluarán, se encuentran el factor de aprendizaje  $\alpha$ , el factor de descuento  $\gamma$  y el tiempo de entrenamiento. En este análisis, se explora detalladamente la influencia de estos hiperparámetros y cómo sus valores óptimos pueden variar según la naturaleza del problema y los objetivos del aprendizaje.

### 5.3.1. Factor de aprendizaje $\alpha$

La Figura 5.5 presenta los resultados del desempeño del sistema con el algoritmo *Q-learning*, empleando cuatro valores de coeficiente  $\alpha$  (0,25, 0,2, 0,15 y 0,1), en el escenario de mínimo esfuerzo descrito en la Tabla 5.2. Este escenario implica la operación de 5000 UEs en el segmento mMTC, 250 UEs en el segmento emBB y 5000 UE en el segmento H2H. La elección de estos valores de  $\alpha$  se basa en la naturaleza de los problemas de asignación de recursos de radio que se desenvuelven en entornos dinámicos y cambiantes. Un  $\alpha$  pequeño fomenta la evaluación de una variedad de asignaciones a lo largo del tiempo antes de decidir cuál es la mejor, evitando así tomar decisiones impulsivas.

En la Gráfica 5.5 se observa que a medida que  $\alpha$  aumenta, surge un deterioro en el rendimiento del sistema. En particular, cuando  $\alpha$  excede 0,20, se evidencia un incremento en las colisiones de preámbulos (Figura 5.5(b)). Para los casos de  $\alpha = 0,20$  y  $\alpha = 0,25$ , el máximo número de colisiones de preámbulos llega a ser aproximadamente 40, mientras que para los valores de  $\alpha$  igual a 0,15 y 0,10, este número no supera las 15 colisiones, es decir, es inferior a la mitad. Este incremento en las colisiones provoca un aumento en los intentos de acceso subsiguientes, tanto dentro del mismo SIB2 como en los SIB2 siguientes, como se evidencia en la Figura 5.5(a). Asimismo, en las Figuras 5.5(c) y 5.5(d) se aprecian descensos notables en la tasa de preámbulos decodificados con éxito y en los accesos exitosos respectivamente.

En relación con la Tabla 5.6, se presentan promedios de transmisiones de preámbulos y probabilidad de acceso exitoso. Se puede notar que al elevar  $\alpha$  a 0,25, se advierte un aumento en las transmisiones de preámbulos, con un incremento de 0,2 en comparación con el caso de  $\alpha = 0,1$ . De manera similar, un aumento en  $\alpha$  no resulta efectivo en la probabilidad de acceso, ya que esta también disminuye de 1 a 0,9956. En este contexto, se ha tomado la decisión de equilibrar la selección y optar por un valor de  $\alpha$  igual a 0,15. Esta elección pretende mantener una cantidad reducida de transmisiones de preámbulos, con una cifra de 1,7183, a la vez que se preserva una alta probabilidad de acceso, con un valor de 0,9999.

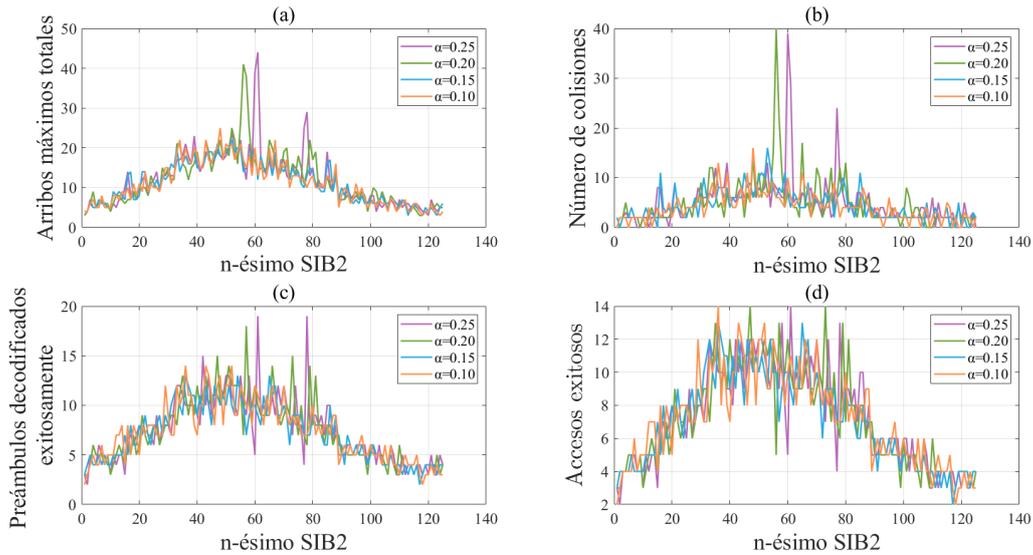


Figura 5.5: Congestión y acceso exitoso para *Q-learning* a medida que  $\alpha$  varía.

Tabla 5.6: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando  $\alpha$  varía.

	Preámbulos Transmitidos				Probabilidad de Acceso exitoso			
	$\alpha = 0,1$	$\alpha = 0,15$	$\alpha = 0,2$	$\alpha = 0,25$	$\alpha = 0,1$	$\alpha = 0,15$	$\alpha = 0,2$	$\alpha = 0,25$
<b>mMTC</b>	1,7132	1,7335	1,7887	1,9473	1	0,9998	0,9998	0,9998
<b>EMBB</b>	1,6773	1,6819	1,7231	1,7361	1	1	0,9996	0,9871
<b>H2H</b>	1,5376	1,4516	1,5484	1,6452	1	1	1	1
<b>TOTAL</b>	1,6993	1,7183	1,7761	1,8745	1	0,9999	0,9998	0,9956

### 5.3.2. Factor de descuento $\gamma$

Al igual que con el parámetro  $\alpha$ , se procede a generar una gráfica de rendimiento del sistema utilizando el algoritmo *Q-learning*, con un escenario mínimo esfuerzo. En esta ocasión, se experimenta con distintos valores de  $\gamma$ , el factor de descuento (0,3, 0,5, 0,7 y 0,9). Los resultados mostrados en la Figura 5.6 destacan que los peores desempeños se dan tanto en valores muy altos (aproximados a 1, como 0,9) como en valores muy bajos (cerca de 0, como 0,3) de  $\gamma$ . En estas configuraciones, se observa un aumento de colisiones, reflejado en picos considerables, así como una disminución en el éxito de decodificación de preámbulos y en los accesos con éxito.

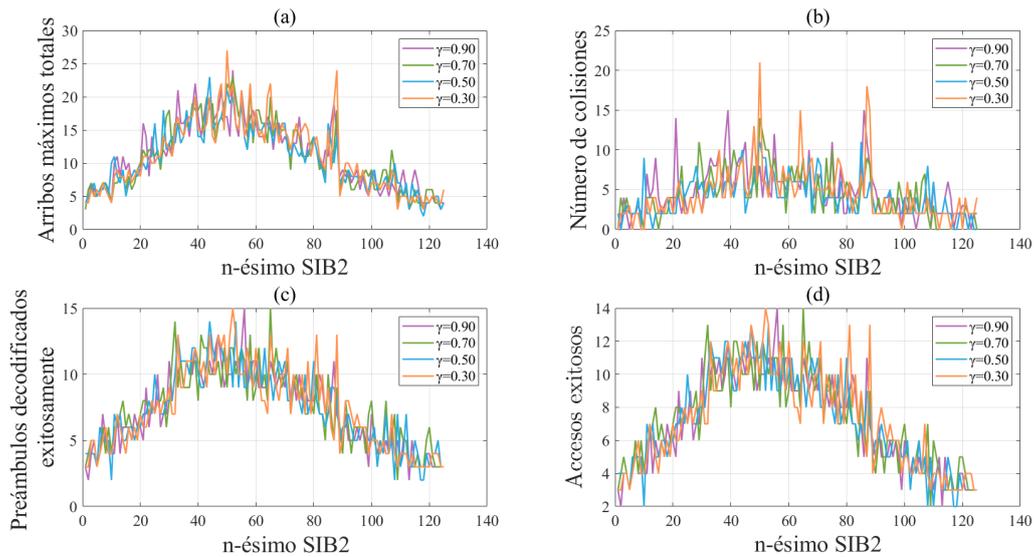


Figura 5.6: Congestión y acceso exitoso para *Q-learning* a medida que  $\gamma$  varía.

Los resultados mostrados en la Tabla 5.7 corroboran lo mencionado anteriormente. Se constata que los valores de  $\gamma$  de 0,5 y 0,7 logran mejores resultados en términos de transmisiones de preámbulos con 1,70038 y 1,6852, en comparación con los valores de 1,8222 y 1,767 obtenidos con  $\gamma$  de 0,3 y 0,9, respectivamente. Asimismo, en la probabilidad de acceso, se observan disminuciones de 0,9985 y 0,9987 en comparación con los valores de 0,9993 y 0,9999 obtenidos con  $\gamma$  de 0,5 y 0,7, respectivamente.

Considerando estos resultados, se opta por trabajar con un valor de  $\gamma$  igual a 0,7, debido a una ligera mejora en el rendimiento en comparación con  $\gamma$  de 0,5. La experiencia de obtener peores resultados con valores de  $\gamma$  muy bajos o muy altos subraya la necesidad de encontrar un equilibrio en el algoritmo *Q-learning* entre la importancia de las recompensas a corto plazo y a largo plazo. En este caso, se favorecerá un enfoque ligeramente mayor en las recompensas a largo plazo, aunque sin descuidar las recompensas inmediatas, las cuales proporcionan información sobre las tendencias de tráfico y asignación de preámbulos en los casos correspondientes.

Tabla 5.7: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando  $\gamma$  varía.

	Preámbulos Transmitidos				Probabilidad de Acceso exitoso			
	$\gamma = 0,3$	$\gamma = 0,5$	$\gamma = 0,7$	$\gamma = 0,9$	$\gamma = 0,3$	$\gamma = 0,5$	$\gamma = 0,7$	$\gamma = 0,9$
<b>mMTC</b>	1,8879	1,7112	1,6910	1,7846	0,9984	0,9990	0,9998	0,9990
<b>emBB</b>	1,7013	1,6951	1,6793	1,7384	0,9988	1	1	0,9980
<b>H2H</b>	1,5591	1,5376	1,5269	1,6344	1	1	1	1
<b>TOTAL</b>	1,8222	1,7038	1,6852	1,7676	0,9985	0,9993	0,9999	0,9987

### 5.3.3. Tiempo de entrenamiento

En esta sección, partimos del mismo escenario utilizado en las pruebas para determinar los valores de  $\alpha$  y  $\gamma$  que, experimentalmente, han demostrado ofrecer los mejores resultados. Basándonos en los resultados obtenidos, donde se identificaron valores efectivos de  $\alpha = 0,15$  y  $\gamma = 0,9$ , se centra en evaluar el rendimiento del sistema en relación con diferentes intervalos de tiempo de entrenamiento (100s, 200s, 300s y 400s). El propósito central de esta evaluación es identificar el punto óptimo para dejar de utilizar el valor de  $\epsilon$  en su máximo ( $\epsilon = 1$ ) para la exploración, y comenzar gradualmente a reducirlo. Este período se lo ha denominado *entrenamiento* y refleja la experiencia necesaria para lograr resultados satisfactorios.

Al analizar la Figura 5.7, se observa que al emplear tiempos de entrenamiento de 100s y 200s, los niveles de colisiones alcanzan picos elevados cercanos a 30. Esto conlleva a un bajo rendimiento en cuanto a la decodificación precisa de preámbulos y a la consecución de accesos exitosos, manifestándose en valles notables visualizados en las Figuras 5.7(c) y 5.7(d). Sin embargo, al extender los tiempos de entrenamiento a 300s y 400s, se observa un mejor control de los valores de colisión. En su mayoría, estos valores son inferiores a 20 en el caso de 300s y a 15 en el caso de 400s (ver Figura 5.7(b)). Esta observación sugiere que aumentar el tiempo de entrenamiento contribuye a reducir significativamente los episodios de colisión y a mejorar el rendimiento general del sistema.

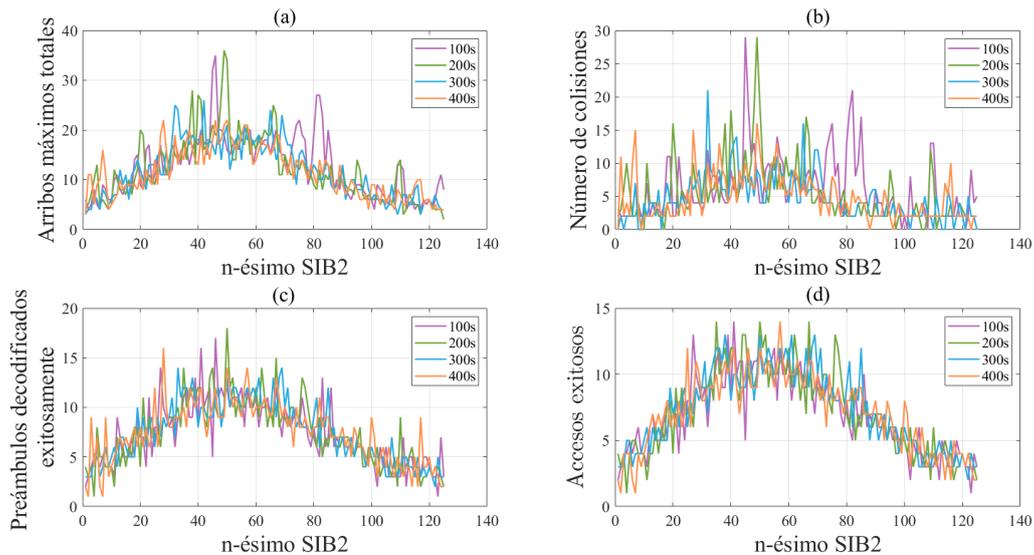


Figura 5.7: Congestión y acceso exitoso para *Q-learning* a medida que el tiempo de entrenamiento varía.

En la Tabla 5.8, se observa que el número de transmisiones de preámbulos presenta una disminución conforme se aumenta el tiempo de entrenamiento. Por ejemplo, con un tiempo de simulación de 100s, el número más alto de transmisiones es de 2,0504, mientras que con un tiempo de entrenamiento de 400s, esta cifra disminuye a 1,7783, lo que significa una reducción de 0,2721 transmisiones, en un contexto donde se emplea una cantidad moderada de dispositivos de usuario por segmento. En relación con la probabilidad de acceso, se tiene una tendencia creciente a medida que aumenta el tiempo de entrenamiento. Este aumento se debe a que la tabla *Q* se llena con más valores provenientes de la exploración y, por lo tanto, contiene menos valores inciertos.

En aras de optimizar la probabilidad de acceso y disminuir la retransmisión de preámbulos, se ha decidido trabajar con un tiempo de entrenamiento de 400s, ya que proporciona la probabilidad de acceso global del sistema más alta con el 0,9988 en el escenario presentado, se debe tener en cuenta que aumentar significativamente este valor no se considera recomendable. Se ha determinado que un tiempo de entrenamiento adecuado es de 400 segundos, ya que representa un equilibrio entre el tiempo de entrenamiento y el rendimiento del algoritmo. Incrementar aún más el tiempo de entrenamiento no proporcionaría mejoras sustanciales en el rendimiento y resultaría

en un uso ineficiente de los recursos computacionales.

Tabla 5.8: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso obtenida cuando el tiempo de entrenamiento varía.

	Preámbulos Transmitidos				Probabilidad de Acceso exitoso			
	100s	200s	300s	400s	100s	200s	300s	400s
<b>mMTC</b>	2,0104	1,9505	1,9353	1,8370	0,9986	0,9966	0,9980	1
<b>emBB</b>	2,1470	1,7225	1,7061	1,6724	0,9860	0,9937	0,9984	0,9963
<b>H2H</b>	1,6947	1,6220	1,5730	1,4713	1	1	1	1
<b>TOTAL</b>	2,0504	1,8737	1,8560	1,7783	0,9945	0,9957	0,9982	0,9988

#### 5.4. Análisis comparativo entre algoritmos

En esta sección final, se realizará un análisis comparativo detallado entre un escenario donde los preámbulos asignados son fijos y los diferentes algoritmos propuestos en el estudio: el modelo de aprendizaje supervisado y el modelo de aprendizaje reforzado. Se compararán sus resultados, ventajas y desventajas, destacando las áreas en las que cada enfoque demuestra ser más efectivo en términos de optimización de recursos, reducción de colisiones y mejora en el acceso a la red.

##### 5.4.1. Análisis en el escenario de máximo esfuerzo

Este escenario se caracteriza por la presencia de una carga de solicitudes considerablemente alta en la red, lo que pone a prueba la eficiencia y la capacidad de adaptación de la red neuronal bajo condiciones de alta demanda. Para evaluar cómo las soluciones planteadas abordan esta situación desafiante, se ha seleccionado el escenario de máximo esfuerzo, cuyos detalles se detallan en la Tabla 5.2. Este escenario se escoge debido a su capacidad para generar bloqueos de acceso como resultado de retransmisiones y colisiones masivas.

En este caso de estudio, el modelo de asignación fija de preámbulos usa la siguiente distribución:

- Preámbulos disponibles para mMTC: 43.
- Preámbulos disponibles para emBB: 10.

- Preámbulos disponibles para H2H: 1.

En las Figuras 5.8, 5.9 y 5.10, se presentan los resultados de la simulación de máxima demanda en el contexto de la comunicación mMTC. Para una mejor claridad y comparación de los eventos observados, los gráficos se han separado en figuras distintas. A partir del decimosexto mensaje SIB2, se evidencia un estancamiento en los resultados. Este fenómeno se debe a las transmisiones generadas por colisiones y decodificaciones fallidas que ocurrieron previamente. Este bloqueo se manifiesta en el momento en el que la cantidad de arribos supera la capacidad del canal.

En la Figura 5.8, se presenta la comparación entre la asignación fija de preámbulos y la aplicación del algoritmo de *Q-learning*. En las Figuras 5.8(a) y 5.8(b), se puede observar que tanto los arribos máximos como las colisiones muestran una tendencia similar, lo que sugiere que, en promedio, las colisiones son similares para ambas propuestas. Sin embargo, en las Figuras 5.8(c) y 5.8(d), se deduce que, en promedio, la asignación fija de preámbulos logra una decodificación exitosa de un mayor número de preámbulos y, en consecuencia, un mayor número de accesos exitosos. Al calcular el porcentaje de mejora en la decodificación correcta de preámbulos, se encuentra que el método de asignación fija supera al algoritmo de *Q-learning* en aproximadamente un 50%. Del mismo modo, el porcentaje de mejora en los accesos exitosos es del 42% para la asignación fija en comparación con el algoritmo de *Q-learning*.

Por otro lado, la Figura 5.9 proporciona una comparación entre los resultados obtenidos al aplicar la RNA frente a una asignación fija de preámbulos. Las Figuras 5.9(a) y 5.9(b), que representan el número total de arribos y las colisiones observadas, muestran una tendencia similar. Esto se debe a que las solicitudes de acceso que experimentan colisiones tienden a retransmitir el Msg1. Por otro lado, las Figuras 5.9(c) y 5.9(d) sugieren que, en promedio, la asignación fija de preámbulos obtiene mejores resultados. Al calcular el promedio de mejora, se destaca que la asignación fija de preámbulos logra una mejora del 28,35% en comparación con la RNA en términos de preámbulos decodificados correctamente. En cuanto a los accesos exitosos, la asignación fija supera a la RNA en un 45%.

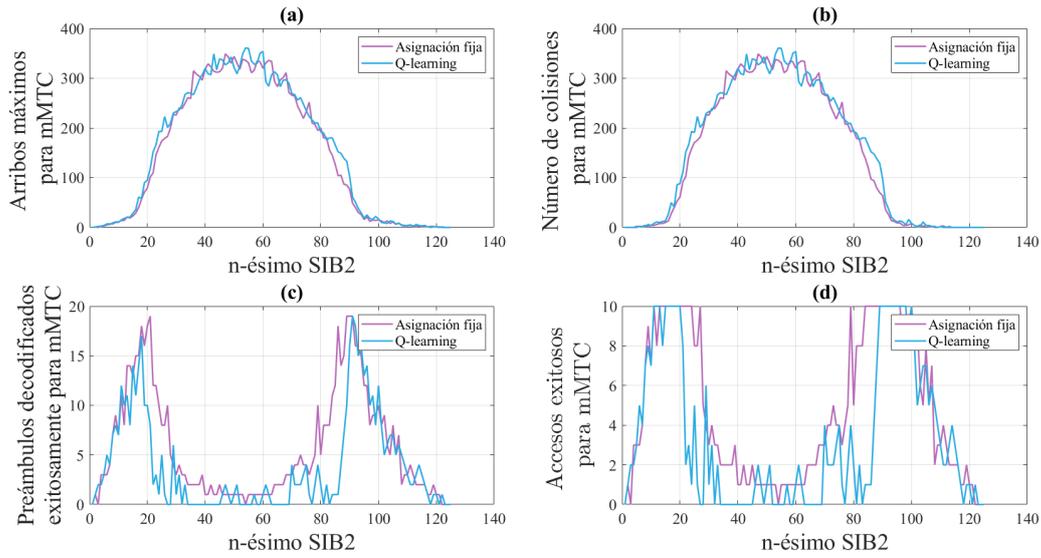


Figura 5.8: Gráfica comparativa entre el algoritmo de *Q-learning* vs la asignación fija de preámbulos para una comunicación mMTC en un escenario de máximo esfuerzo

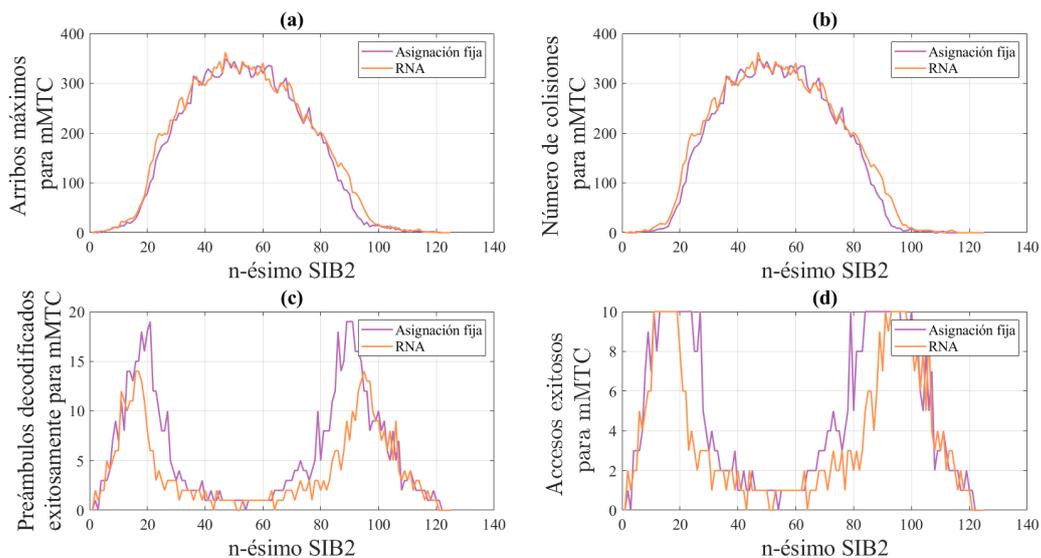


Figura 5.9: Gráfica comparativa entre la RNA vs la asignación fija de preámbulos para una comunicación mMTC en un escenario de máximo esfuerzo

La Figura 5.10, por su parte, presenta una comparación entre los resultados obtenidos al aplicar la RNA y el algoritmo de *Q-learning*. Las Figuras 5.10(a) y 5.10(b), que representan el número total de arribos y las colisiones, muestran una tendencia similar debido a que las solicitudes de acceso que colisionan tienden a retransmitir el Msg1. Por otro lado, las gráficas de las Figuras 5.10(c) y 5.10(d) indican que, en

promedio, la RNA logra mejores resultados tanto en la decodificación como en los accesos exitosos. Sin embargo, el algoritmo de *Q-learning* exhibe picos más altos en la decodificación y los accesos exitosos. Al calcular el promedio de mejora, se destaca que la RNA presenta una ligera mejora del 2,83 % en comparación con el algoritmo de *Q-learning* en términos de preámbulos decodificados correctamente. En cuanto a los accesos exitosos, la RNA supera al algoritmo de *Q-learning* en un 9,74 %.

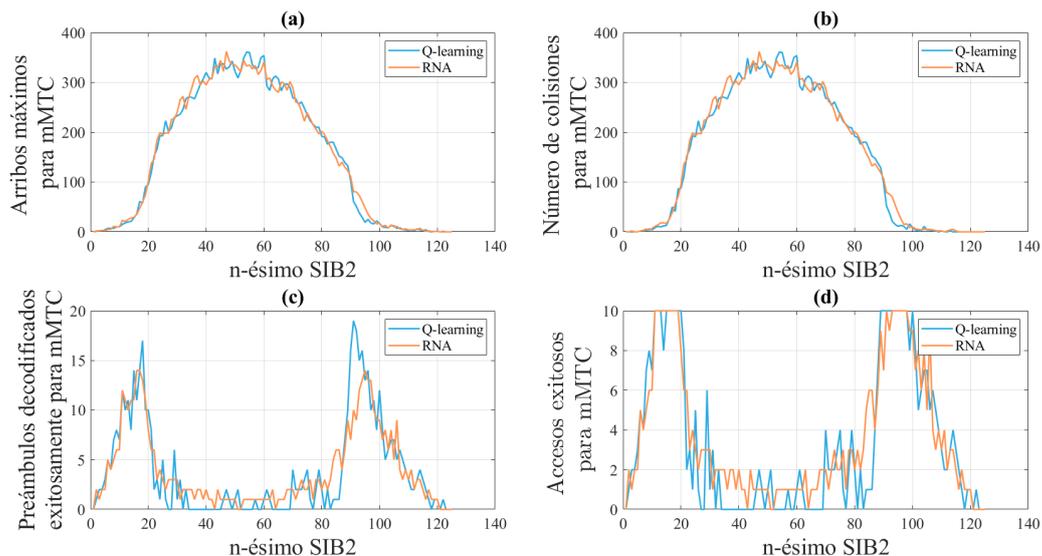


Figura 5.10: Gráfica comparativa entre la RNA vs el algoritmo de *Q-learning* para una comunicación mMTC en un escenario de máximo esfuerzo

En las Figuras 5.8(a), 5.9(a) y 5.10(a), se observa que la tendencia de las gráficas guarda similitud con los resultados presentados en la Figura 2.9 de [9], lo que refuerza la consistencia de los hallazgos obtenidos. La principal diferencia entre estas gráficas radica en el número de UEsM utilizados.

En cuanto a los resultados para la comunicación mMTC, se destaca que la asignación fija de preámbulos logra un mayor número de accesos exitosos. Sin embargo, es importante considerar que para la asignación fija de preámbulos, el 80 % de los preámbulos disponibles se asignan al segmento mMTC, mientras que el segmento emBB recibe solo el 18 % de los preámbulos disponibles. Esto provoca que el segmento emBB sufra de una falta de preámbulos disponibles, lo que resulta en un mayor número de colisiones y un menor número de arribos exitosos para ese segmento, especialmente cuando el número de arribos esperados supera la capacidad del canal.

A continuación, las Figuras 5.11, 5.12 y 5.13 presentan los resultados para el segmento emBB. Las Figuras 5.11(a) y 5.11(b) representan los arribos máximos totales y las colisiones existentes. Se observa claramente que el algoritmo de *Q-learning* reduce significativamente el número de transmisiones existentes y, en consecuencia, el número de colisiones. Además, las Figuras 5.11(c) y 5.11(d) indican que se logra una mayor cantidad de preámbulos decodificados correctamente y un mayor número de accesos exitosos utilizando el algoritmo de *Q-learning*. En la mayor parte de la simulación, la asignación fija de preámbulos no logra superar un valor de 2 preámbulos decodificados correctamente. Al calcular el promedio de mejora, se destaca que el algoritmo de *Q-learning* logra una mejora significativa, representando 10 veces más preámbulos decodificados correctamente que la asignación fija de preámbulos. En cuanto a los accesos exitosos, el algoritmo de *Q-learning* logra una mejora cercana a 5 veces más que la asignación fija de preámbulos.

En la Figura 5.11(b) y en la Figura 5.11(c), se puede apreciar el funcionamiento de la política de castigos y recompensas establecida. A medida que aumenta el número de colisiones, se reduce la disponibilidad de preámbulos. Este enfoque tiene como objetivo mitigar el efecto de las retransmisiones masivas y favorecer una distribución más equitativa de recursos. El algoritmo de *Q-learning* logra un equilibrio adecuado entre la asignación de preámbulos y las colisiones, promoviendo un acceso más eficiente para los UEsB.

Las Figuras 5.12(a) y 5.12(b) presentan los arribos máximos totales y las colisiones existentes. A simple vista, se observa que la RNA reduce notablemente el número de transmisiones existentes y, en consecuencia, la cantidad de colisiones. Del mismo modo, las Figuras 5.12(c) y 5.12(d) indican que la RNA logra una mayor cantidad de preámbulos decodificados correctamente y un mayor número de accesos exitosos en comparación con la asignación fija de preámbulos. Al calcular el promedio de mejora, se destaca que la RNA obtiene una mejora significativa, representando 13 veces más preámbulos decodificados correctamente que la asignación fija de preámbulos. En cuanto a los accesos exitosos, el algoritmo de *Q-learning* logra una mejora de 5 veces más que la asignación fija de preámbulos.

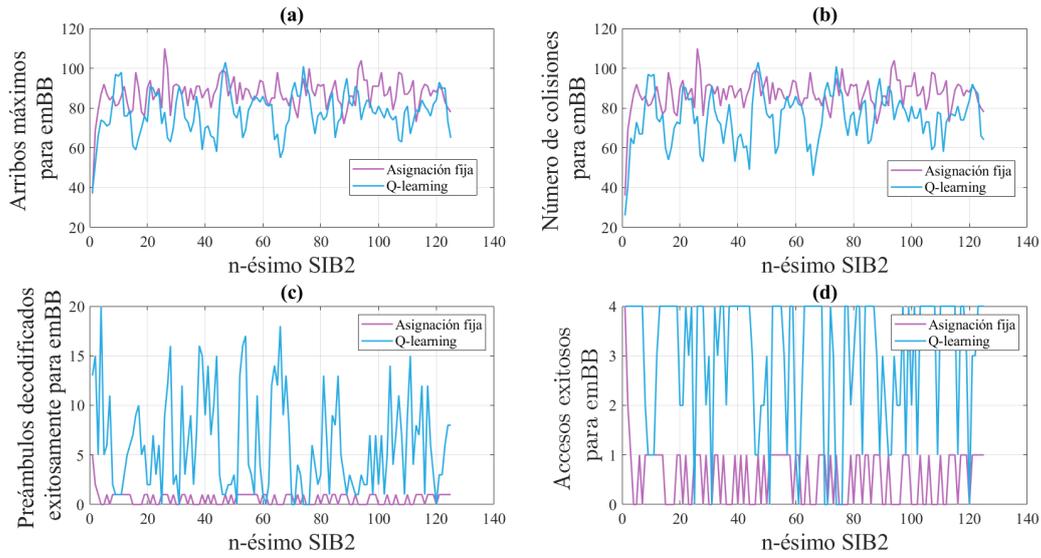


Figura 5.11: Gráfica comparativa entre la asignación fija de preámbulos vs el algoritmo de *Q-learning* para una comunicación emBB en un escenario de máximo esfuerzo

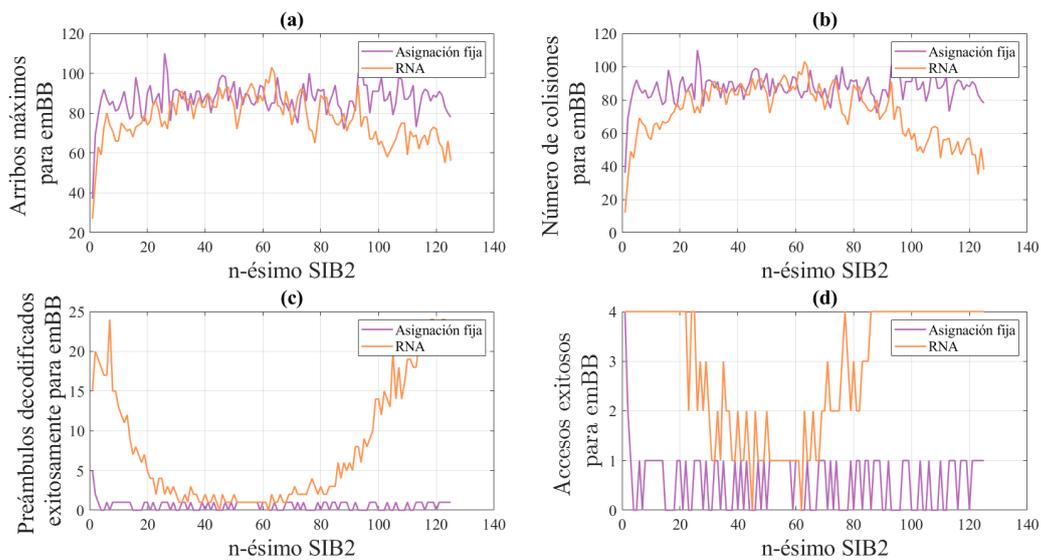


Figura 5.12: Gráfica comparativa entre la asignación fija de preámbulos vs la RNA para una comunicación emBB en un escenario de máximo esfuerzo

En las Figuras 5.13(a) y 5.13(b), se presentan los arribos máximos totales y las colisiones existentes. Se puede observar que la RNA provoca una curvatura en la gráfica, la cual es el resultado de la acumulación de arribos. La Figura 5.13(c) indica que la RNA logra una mayor decodificación de preámbulos en comparación con el algoritmo

de *Q-learning*, mientras que la Figura 5.13(d) muestra un mayor número de accesos exitosos por parte del algoritmo de *Q-learning*. Al calcular el promedio de mejora, se destaca que la RNA logra una mejora del 28,17 % en comparación con los preámbulos decodificados correctamente por el algoritmo de *Q-learning*. En cuanto a los accesos exitosos, el algoritmo de *Q-learning* obtiene un porcentaje de mejora del 3,6 % en comparación con la RNA. Como inferencia final, se destaca que mientras la RNA sufre un bloqueo de acceso provocado por las transmisiones masivas, el algoritmo de *Q-learning* logra mitigar ese efecto.

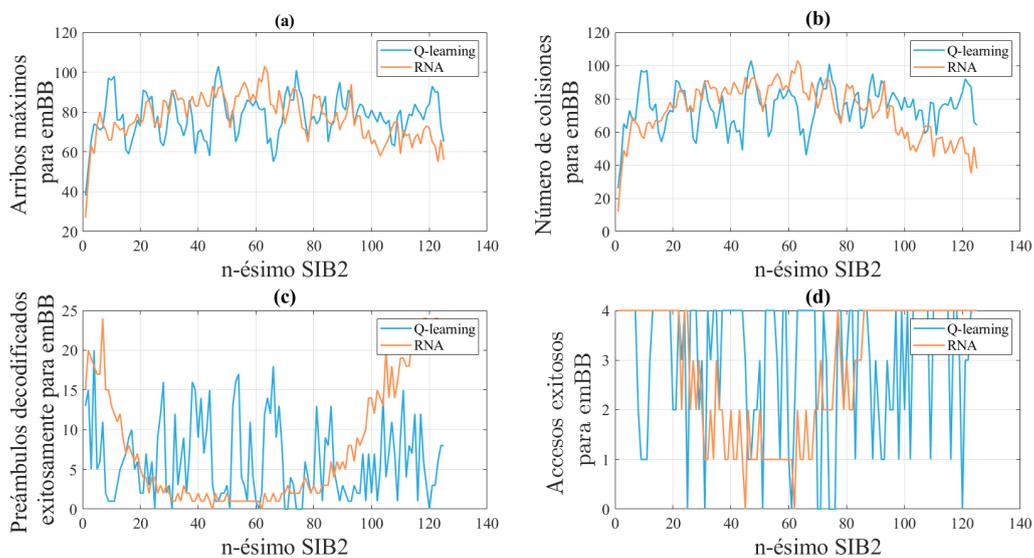


Figura 5.13: Gráfica comparativa entre la RNA vs el algoritmo de *Q-learning* de preámbulos para una comunicación emBB en un escenario de máximo esfuerzo

En las Figuras 5.11, 5.12 y 5.13, se puede notar que el método de asignación fija de preámbulos muestra muy pocos valores decodificados correctamente. Este fenómeno se debe a que en el segmento mMTC, se han asignado una cantidad significativa de preámbulos, lo que reduce considerablemente las oportunidades de acceso para los usuarios del segmento emBB.

En la Figura 5.14, se presentan los resultados a nivel global de la celda, considerando la suma de los indicadores de rendimiento de cada segmento. El análisis se centra particularmente en las Figuras 5.14(c) y 5.14(d), ya que proporcionan información crucial sobre el rendimiento global de la red. En la Figura 5.14(c), se destaca que la simulación con el algoritmo de *Q-learning* ha logrado mitigar el fenómeno de

bloqueo de acceso en ciertos momentos de la simulación, mostrando una variación entre un aumento y una disminución en el número de preámbulos decodificados correctamente. Este efecto se debe a la política de recompensas y penalizaciones del algoritmo. Como resultado, esta simulación alcanza un promedio superior de usuarios exitosos en comparación con otras estrategias. Por otro lado, en las Figuras 5.14(a) y 5.14(b) presentan un efecto similar de colisiones y arribos para los tres modelos de estudio, la tendencia de la curva de campana viene se caracteriza por la presencia de la comunicación mMTC.

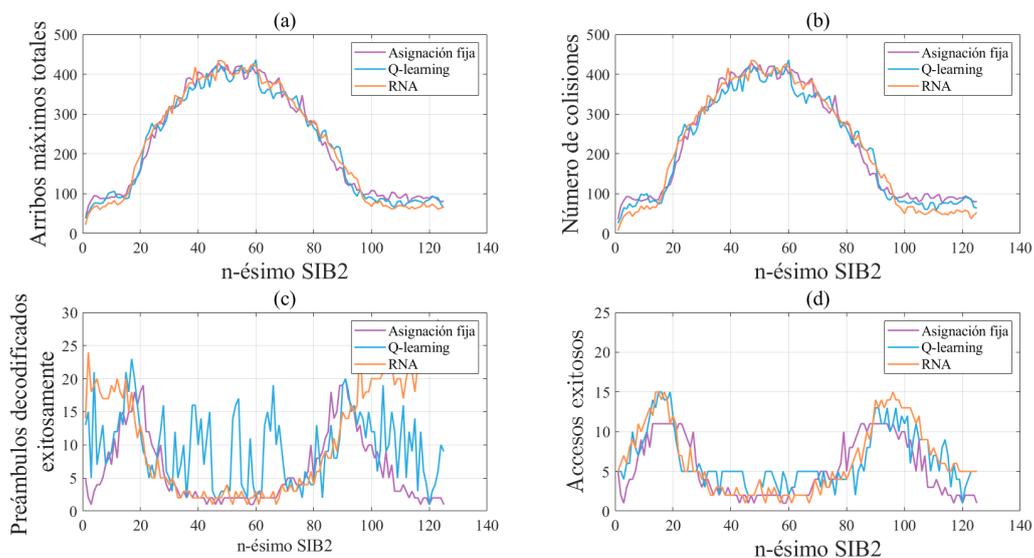


Figura 5.14: Resultados del escenario de máximo esfuerzo para las diferentes experimentaciones.

Finalmente, la Tabla 5.9 presenta el promedio de accesos exitosos y la cantidad promedio de transmisiones necesarias para adquirir recursos de radio. De acuerdo con estos resultados, se concluye que, en un escenario de explotación intensiva como el abordado en este análisis, el algoritmo *Q-learning* demuestra ser la estrategia más eficaz, seguido de cerca por la red neuronal.

Tabla 5.9: Valor promedio de los preámbulos transmitidos y la probabilidad de acceso exitoso obtenida con diferentes métodos de asignación de preámbulos.

	Preámbulos Transmitidos			Probabilidad de Acceso exitoso		
	Preámbulos fijos	Q-learning	RNA	Preámbulos fijos	Q-learning	RNA
<b>mMTC</b>	3,6892	3,3592	3,7340	0,1920	0,1473	0,1191
<b>emBB</b>	4,0893	5,3199	5,1552	0,0076	0,2395	0,2682
<b>H2H</b>	3,1772	1,9848	2,0104	0,7505	0,9981	0,9982
<b>TOTAL</b>	3,6628	4,0987	4,3158	0,1387	0,1875	0,1791

Al calcular el porcentaje de mejora, se observa que el algoritmo de *Q-learning* logra una mejora del 35 % en el promedio de probabilidad de acceso en comparación con el método de asignación fija de preámbulos. Por otro lado, la RNA logra una mejora del 29 % en el promedio de probabilidad de acceso en comparación con el método de asignación fija de preámbulos.

#### 5.4.2. Probabilidad de acceso en escenario múltiples

En esta sección, nuestro enfoque se orienta hacia la exploración de escenarios más diversos con el propósito de analizar la probabilidad de acceso. Los escenarios que serán objeto de simulación se encuentran definidos en la Tabla 5.10. La selección de estos escenarios se basa en la exploración de modelos de asignación, comenzando desde el primer escenario donde la cantidad de arribos esperados por cada segmento se mantiene por debajo de la capacidad de acceso del PRACH. Luego, los escenarios posteriores aumentan la cantidad de arribos hasta llegar al último escenario en el cual la cantidad de arribos supera la capacidad de acceso.

Tabla 5.10: Parámetros de Simulación.

	Total de UEsM	Total de UEsB	Total de UEsH
<b>Escenario 1</b>	6000	200	8000
<b>Escenario 2</b>	14000	450	16000
<b>Escenario 3</b>	22000	700	24000
<b>Escenario 4</b>	30000	950	32000
<b>Escenario 5</b>	38000	1200	40000

En estos escenarios, se varía el número de usuarios en los segmentos UEsM, UEsB, y UEsH, desde situaciones de baja demanda hasta situaciones donde la capacidad

de acceso se ve superada.

La Figura 5.15 muestra la probabilidad de acceso exitoso  $P_{sM}$  en función de los diferentes enfoques. La simulación de asignación fija, que siempre asigna 43 preámbulos, logra una probabilidad de acceso más alta. Sin embargo, este enfoque conduce a una menor probabilidad de acceso exitoso  $P_{sB}$  debido a la asignación excesiva de preámbulos. Tanto el algoritmo de *Q-learning* como el de red neuronal muestran una mayor probabilidad de acceso exitoso  $P_{sB}$  (ver Figura 5.16).

Recordando lo descrito en la sección 4.6.2, la cantidad máxima de UEsM que pueden acceder dentro de la celda en un periodo de 2000 RAOs es de 20,000. Para UEsB, la cantidad máxima permitida es de 8,000 (teniendo en cuenta que cada UE realiza un promedio de 10 actualizaciones por arribo). Es por esta razón que en la Figura 5.16, la probabilidad de acceso se acerca al 100 % para los escenarios en los que el tráfico no ocupa ni la mitad de la capacidad permitida por el segmento, por ejemplo, cuando UEsB es de 200 o 450. A medida que aumenta el número de peticiones, la capacidad del canal reduce la probabilidad de acceso; sin embargo, las estrategias de ML logran obtener un mejor resultado en comparación con una asignación fija.

En la Figura 5.17, se presenta la probabilidad de acceso exitoso total, definida por la Ecuación (5.1). Esta figura revela una característica importante: los modelos de ML estudiados muestran un mejor rendimiento en escenarios de sobrecarga, donde la cantidad esperada de usuarios supera la capacidad de la celda. Esto subraya la efectividad de las estrategias de ML para gestionar el tráfico en situaciones de alta demanda, en comparación con la asignación fija. Sin embargo, cuando la cantidad de peticiones no supera la capacidad de la celda, la asignación de preámbulos fija muestra una mejor probabilidad de acceso. La capacidad total de la celda es de 300,000 usuarios durante un periodo de 2000 RAOs.

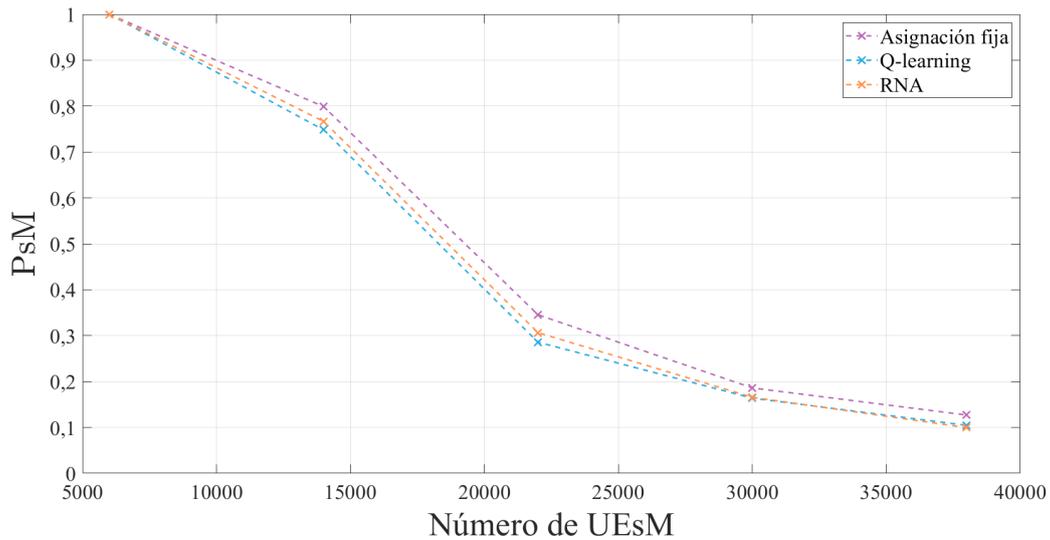


Figura 5.15: Probabilidad de acceso exitoso vs número de UEsM esperados.

Para verificar la exactitud de los resultados generados con diferentes métodos de asignación de preámbulos, se ha calculado intervalos de confianza del 95 % para las diferentes cantidades de UEs del servicio mMTC detalladas en la Tabla 5.10. En la Tabla 5.11 se puede observar que todos los valores de intervalos de confianza obtenidos, tanto con la asignación fija de preámbulos, el enfoque de *Q-learning* y RNA son valores pequeños próximos a cero. Esto nos indica que se tiene una alta confianza en los resultados producidos por los tres métodos para la asignación de preámbulos de acceso en mMTC. Sin embargo, cabe mencionar que al comparar los algoritmos ML, la RNA demuestra ser el método de mayor precisión debido a que presenta intervalos de confianza de menor magnitud en todas cantidades de UEs empleadas, en contraste con el enfoque de *Q-learning*.

Tabla 5.11: Intervalos de confianza del 95 % para mMTC obtenidos con diferentes métodos de asignación de preámbulos.

	UEsM				
	6000	14000	22000	30000	38000
<b>Preámbulos fijos</b>	8,7652E-07	0,00281139	0,00248414	0,0023453	0,00031238
<b>Q-learning</b>	0,000238	0,00583103	0,00657479	0,00467573	0,00465094
<b>Red Neuronal</b>	8,76523E-07	0,00526082	0,00211722	0,002518847	0,000991672

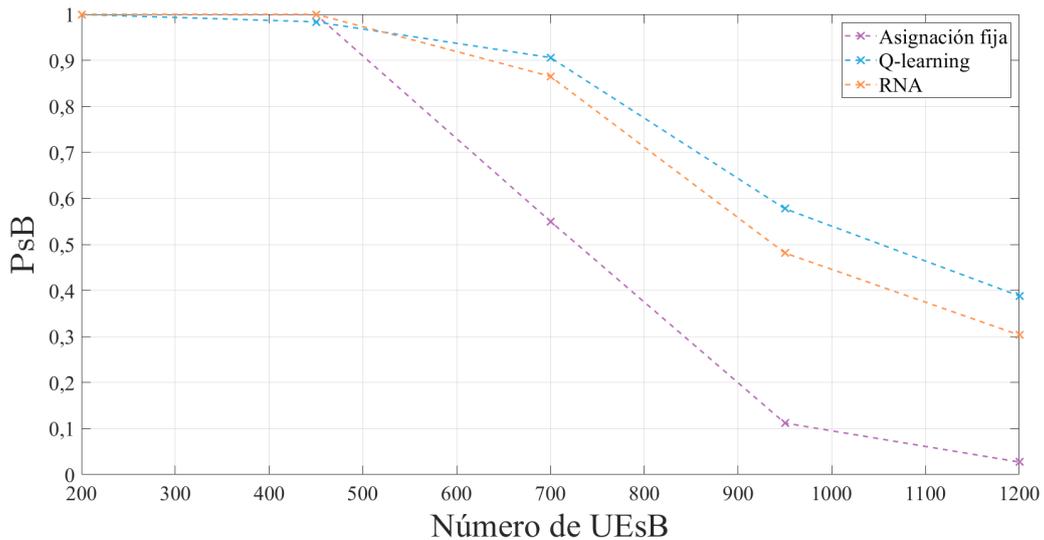


Figura 5.16: Probabilidad de acceso exitoso vs número de UEsB esperados.

En la tabla 5.12 se presentan los intervalos de confianza al 95 % para diferentes cantidades de UE en el segmento emBB, especificadas en los parámetros de simulación de la Tabla 5.10. Los valores obtenidos mediante con la asignación de preámbulos fijos y con RNA demuestran tener menor variabilidad de la probabilidad de acceso en múltiples simulaciones, a comparación del método de *Q-learning*. En el caso de la RNA, se presenta mejores resultados experimentalmente con los valores de 200, 450 y 700 UEs. Mientras, con la asignación fija de preámbulos los mejores resultados son visualizados en los últimos casos de 950 y 1200 UEs. Por otro lado, *Q-learning* se encuentra en último lugar de los tres métodos para asignación preámbulos en el segmento emBB, ya que presenta intervalos de confianza mayores en todo el escenario de simulación planteado.

Tabla 5.12: Intervalos de confianza del 95 % para emBB obtenidos con diferentes métodos de asignación de preámbulos.

	UEsB				
	200	450	700	950	1200
<b>Preámbulos fijos</b>	3,96669E-05	0,000461128	0,018871519	0,005573197	0,001750301
<b>Q-learning</b>	0,000297502	0,001596592	0,011508354	0,009415926	0,015142832
<b>Red Neuronal</b>	8,76523E-07	7,93338E-05	0,0122769	0,006629328	0,003004766

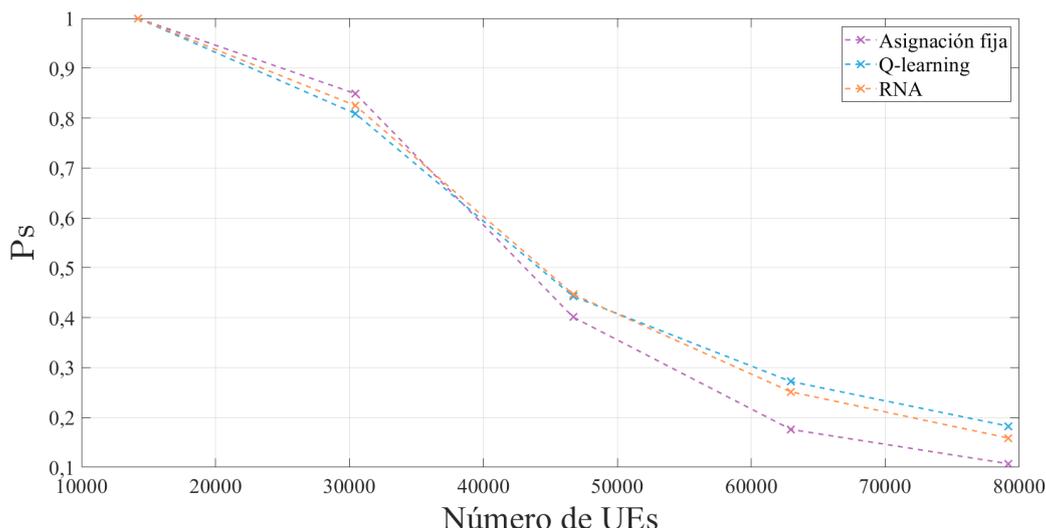


Figura 5.17: Probabilidad de acceso exitoso vs número de UEs esperados.

Los intervalos de confianza mostrados en la Tabla 5.13 proporcionan una medida de la variabilidad en los resultados de la probabilidad total de acceso en la red, es decir, la suma del número de arribos de usuarios de los segmentos mostrados en la Tabla 5.10. Se destaca que tanto los intervalos derivados de la utilización de la RNA como los obtenidos mediante la asignación fija de preámbulos muestran un desempeño sobresaliente. En particular, los intervalos de confianza más reducidos se observan en el caso de RNA con 14200 y 46700 UEs, mientras que la asignación fija de preámbulos exhibe valores más pequeños en los casos de 30450, 62950 y 79200 UEs. Por otro lado, en congruencia con los resultados de las Tablas 5.11 y 5.12, el enfoque basado en *Q-learning* presenta una variabilidad mayor en la probabilidad de acceso en comparación de los otros dos métodos. No obstante, es importante tener en cuenta que los valores de intervalos de confianza analizados se encuentran en una escala pequeña, lo que sugiere que incluso la variabilidad observada es de magnitud limitada.

Tabla 5.13: Intervalos de confianza del 95 % obtenidos con diferentes métodos de asignación de preámbulos.

	UEs				
	14200	30450	46700	62950	79200
<b>Preámbulos fijos</b>	1,98334E-05	0,002008136	0,006292159	0,002727098	0,000609878
<b>Q-learning</b>	0,124200286	0,09862822	0,055144299	0,033961956	0,026168974
<b>Red Neuronal</b>	8,76523E-07	0,003966688	0,002791557	0,003163434	0,001348674

A partir de la Figura 5.17, se procede a evaluar el porcentaje de mejora de los algoritmos de ML en comparación con el método de asignación fija. Estos resultados se resumen en la Tabla 5.14. Es importante destacar que, dado que la capacidad de la celda es de 30000 UEs, se observa una mejora significativa causada por los algoritmos de ML cuando la cantidad total de usuarios que llegan supera esta capacidad.

Por ejemplo, en un escenario donde se esperan 79200 usuarios, el algoritmo de *Q-learning* logra una mejora del 70.1 % en comparación con el resultado obtenido por la asignación fija de preámbulos. Del mismo modo, la RNA puede lograr una mejora del 47.94 % en este escenario. De la Tabla 5.14, se deduce que a medida que la cantidad de usuarios supera la capacidad de la celda, también aumenta la mejora observada en la aplicación de algoritmos de ML.

Tabla 5.14: Porcentaje de mejora con respecto a la asignación fija de preámbulos

<b>Total UEs</b>	<b>Mejora provocada con Q-learning</b>	<b>Mejora provocada con RNA</b>
<b>14200</b>	-0,01	0
<b>30450</b>	-4,72	-2,82
<b>46700</b>	10,37	11,42
<b>62950</b>	54,54	42,78
<b>79200</b>	70,24	47,94

## Capítulo 6 Conclusiones y recomendaciones

### 6.1. Conclusiones

Este trabajo presenta la implementación de dos tipos de algoritmos de ML: supervisado y reforzado, con el propósito de lograr una asignación dinámica de grupos de preámbulos en un entorno de celda única que segmenta la red en tres segmentos de servicios 5G: mMTC, emBB y H2H. Las soluciones planteadas adaptan la asignación de preámbulos a cambios repentinos en la intensidad del tráfico, provocando que tanto algoritmo *Q-learning* como la RNA mejoran en promedio el desempeño del sistema en situaciones de sobrecarga (ver Tabla 5.14). Estos algoritmos trabajan en conjunto considerando métricas como la probabilidad exitosa de acceso a los recursos de la red y el número de transmisiones de preámbulos, lo que a su vez reduce las colisiones de preámbulos.

Se han obtenido resultados concluyentes que demuestran que, cuando se aplica el método de asignación fija de preámbulos, se genera una disparidad entre los distintos segmentos. Esto se debe a que asignar un valor fijo de preámbulos provoca una probabilidad de acceso alta en un segmento, pero a expensas de un decremento de la probabilidad de acceso en otros segmentos. Limitar la asignación con una cantidad fija de preámbulos, incluso en situaciones de alto tráfico en un servicio en particular, ilustra esta limitación.

Si bien se obtuvieron resultados similares en términos de las métricas de rendimiento mencionados en la Sección 5.1 con los dos métodos ML. El algoritmo de *Q-learning* mostró una mejora en comparación con la RNA en cuanto a la probabilidad de acceso del sistema cuando arriban un alto tráfico de UEs a la red. Esta diferencia se debe a que el aprendizaje reforzado es más adecuado para situaciones en las que el entorno es dinámico y evoluciona con el tiempo, permitiendo que el sistema se adapte continuamente a estos cambios mediante la interacción con el entorno. No obstante, esta adaptabilidad constante también puede considerarse una desventaja a comparación con la RNA. Donde, el entrenamiento se basa en datos controlados y etiquetados, lo que conduce a una convergencia más rápida del entrenamiento. Esto contrasta con el enfoque de *Q-learning*, donde los primeros datos están sujetos a exploración de

acciones al azar, lo que reduce la probabilidad de acceso en estas etapas iniciales.

Se ha evaluado la relevancia del tiempo de entrenamiento en el contexto de la implementación de los algoritmos considerando las circunstancias específicas del caso de estudio. En este escenario particular, se enfrenta a una situación en la que una celda experimenta una interrupción en la señalización, lo que genera la necesidad de que un grupo de usuarios restablezca la conexión con la red. En este contexto, el tiempo de entrenamiento adquiere una importancia crítica para garantizar una respuesta rápida y efectiva. En contraste con *Q-learning*, la RNA produce resultados favorables, especialmente considerando el menor tiempo de entrenamiento requerido. Mientras *Q-learning* necesita un extenso proceso de aprendizaje de 400 segundos para operar de manera efectiva, la RNA se entrena en tan solo 14 segundos. En situaciones en las que se requiere una respuesta rápida durante una evaluación de tiempo muy corto, la RNA emerge como la opción preferente. No obstante, si el objetivo es alcanzar los mejores resultados en términos de rendimiento a largo plazo, la elección ideal recae en la propuesta de *Q-learning*.

En el contexto de la red 5G, el enfoque de aprendizaje *Q-learning* se recomienda en mayor medida. Esto se debe a que no exige una gran cantidad de ejemplos etiquetados para entrenar el modelo como lo hace la RNA (2000 datos), lo que es especialmente ventajoso, ya que no siempre se dispone de tales ejemplos existentes.

Los resultados tanto en *Q-learning* como en RNA demuestran que, al trabajar en escenarios de bajo esfuerzo, no se puede aprovechar al máximo las capacidades de ambas políticas en comparación con la asignación estática de preámbulos (Figura 5.17), debido a que no se tienen retransmisiones masivas de preámbulos en estos escenarios. Sin embargo, para casos de moderado y alto esfuerzo, los resultados obtenidos fueron visiblemente mejores. Este hecho permite generalizar el uso de ambos algoritmos de ML a situaciones de alta congestión de tráfico donde los UEs que intentan acceder a los recursos de radio de la red superan la mitad de la capacidad permitida por segmento mMTC, eMBB y H2H.

La estrategia de diseño implementada en el *Q-learning* opera de manera efectiva para mitigar el efecto de bloqueo de acceso causado por las transmisiones masivas del Msg1. Esta política diseñada aborda las transmisiones masivas del Msg1 mediante un

enfoque punitivo que conlleva a una reducción en la asignación de preámbulos. El propósito detrás de esta estrategia es establecer un entorno que priorice las solicitudes de llegada iniciales en lugar de aquellas inducidas por retransmisiones. A lo largo de la simulación, se observa cómo el enfoque de *Q-learning* produce resultados superiores en términos de asignación de enlaces ascendentes.

## 6.2. Recomendaciones

En el caso del *Q-learning*, si se busca lograr resultados más precisos y mejorados, se recomienda reducir la estrategia de reducción de estados. Esta estrategia implica agrupar las métricas que constituyen un estado en grupos, lo que a su vez reduce la cantidad de pares estado-acción. Al optar por grupos más pequeños o incluso por eliminarlos por completo, se logra generar más estados con valores más cercanos entre sí. No obstante, es importante señalar que el incremento en la cantidad de estos estados implica un aumento considerable en la complejidad de la política de recompensas para cada estado. Además, esta acción también afecta al proceso de entrenamiento de datos, ya que se requerirá más tiempo para completar la tabla  $Q(s, a)$  con los datos obtenidos en la fase de exploración.

En lo que respecta a la RNA, se sugiere enriquecer la base de datos, ya que esto puede tener un impacto significativo en la capacidad de aprendizaje de la red. Al disponer de un mayor número de datos etiquetados que representen diversos escenarios de estudio, la RNA podrá adaptarse de manera más efectiva y mejorar su rendimiento.

## 6.3. Trabajos Futuros

- Durante el análisis de los resultados, se ha destacado la relevancia del fenómeno de las retransmisiones. Cuando un preámbulo experimenta una colisión o una decodificación incorrecta, se desencadenan retransmisiones repetidas hasta que el preámbulo se decodifica exitosamente sin colisiones. Estas retransmisiones masivas pueden llevar a un estado de bloqueo en el acceso, ya que tienden a acumularse con el tiempo. Como posible dirección para investigaciones futuras, se sugiere explorar el efecto de las retransmisiones masivas y con-

siderar el desarrollo de un algoritmo de asignación dinámica para el número de retransmisiones permitidas. Esto podría ayudar a evitar el fenómeno de bloqueo y mejorar el rendimiento general del sistema.

- Este proyecto de titulación se ha centrado en un escenario de trabajo que abarca tres segmentos de red. Se sugiere llevar a cabo un estudio adicional que se enfoque en otros segmentos con diversos requisitos de calidad.
- Un proveedor de red no siempre se basará únicamente en la probabilidad de acceso exitoso como métrica para determinar el mejor método de asignación de preámbulos. Dado que existen otras métricas, como los parámetros de calidad de experiencia, se pueden llevar a cabo estudios que se enfoquen en políticas de asignación de preámbulos que consideren las necesidades específicas de cada segmento de red o los requisitos determinados por el proveedor de servicios.

### Referencias

- [1] M. Alain Sultan, “5G System Overview,” 2022. [En línea]. Disponible: <https://www.3gpp.org/technologies/5g-system-overview>
- [2] T. de Cuentas Europeo, “Naturaleza e importancia de la tecnología 5g,” 2022.
- [3] “5G, Internet de las cosas (IoT) y Accesorios tecnológicos ¿Cómo afectan los nuevos usos de las tecnologías inalámbricas a las emisiones electromagnéticas?” *GSMA*, p. 32, Sep. 2017.
- [4] C.-C. Tseng, H.-C. Wang, J.-R. Chang, L.-H. Wang, y F.-C. Kuo, “Design of Two-Step Random Access Procedure for URLLC Applications,” *Wireless Personal Communications*, vol. 121, num. 2, pp. 1187–1219, Nov. 2021. [En línea]. Disponible: <https://doi.org/10.1007/s11277-021-09060-4>
- [5] L. Tello-Oquendo, I. Leyva-Mayorga, V. Pla, J. Martinez-Bauset, J. R. Vidal, V. Casares-Giner, y L. Guijarro, “Performance Analysis and Optimal Access Class Barring Parameter Configuration in LTE-A Networks with Massive M2M Traffic,” *IEEE Transactions on Vehicular Technology*, vol. 67, num. 4, pp. 3505–3520, apr 2018.
- [6] D. Witkowski, “Bridging the gap—21st century wireless telecommunications handbook,” 2019.
- [7] “5G Network Architecture. A High-Level Perspective,” *Huawei Technologies Co. LTD.*, p. 21, Dic. 2006.
- [8] M. Mendieta y I. Minchala, *Teoría Redes Neuronales Artificiales* -, U. de Cuenca, Ed.
- [9] L. Tello-Oquendo, D. Pacheco-Paramo, V. Pla, y J. Martinez-Bauset, “Reinforcement learning-based ACB in LTE-A networks for handling massive M2M and H2H communications,” *IEEE International Conference on Communications*, vol. 2018-May, jul 2018.
- [10] 3rd Generation Partnership Project (3GPP), “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation,” 3GPP, Tech. Rep. TS 36.211, 2017.

- [11] A. U. Gawas, “An Overview on Evolution of Mobile Wireless Communication Networks: 1G-6G,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, num. 5.
- [12] K. Tachikawa, “A Perspective on the Evolution of Mobile Communications,” *IEEE Communications Magazine*, 2003.
- [13] H. Shen, Q. Ye, W. Zhuang, W. Shi, G. Bai, y G. Yang, “Drone-Small-Cell-Assisted Resource Slicing for 5G Uplink Radio Access Networks,” *IEEE Transactions on Vehicular Technology*, vol. 70, num. 7, pp. 7071–7086, Jul. 2021. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9440683/>
- [14] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, y T. Miyazawa, “Consideration On Automation of 5G Network Slicing with Machine Learning,” in *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*. Santa Fe: IEEE, Nov. 2018, pp. 1–8. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8597639/>
- [15] S. Zhang, “An overview of network slicing for 5g,” *IEEE Wireless Communications*, vol. 26, num. 3, pp. 111–117, 2019.
- [16] G. Zhu, J. Zan, Y. Yang, y X. Qi, “A Supervised Learning Based QoS Assurance Architecture for 5G Networks,” *IEEE Access*, vol. 7, pp. 43 598–43 606, 2019. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8673765/>
- [17] V. A. Figueroa Castillo, J. E. Chóez Calle, J. X. Barreto Pin, y C. A. Villacreses Parrales, “5G tecnología inalámbrica que cambiará el mundo por completo,” *UNESUM-Ciencias. Revista Científica Multidisciplinaria. ISSN 2602-8166*, vol. 6, num. 3, pp. 39–48, May 2022. [En línea]. Disponible: <https://revistas.unesum.edu.ec/index.php/unesumciencias/article/view/393>
- [18] Roy Kumer Saha y Chaodit Aswakul, “FUNDAMENTALS OF 5G MOBILE NETWORK Architecture, Requirement, Densification, Cooperation, and Resource Management.”
- [19] Julio Cerezo, Pepe Cerezo, Eugenio Fontán, y Marimar Jiménez, “El Impacto de 5G,” *Evoca Comunicación e Imagen*, vol. 2, p. 63.

- [20] 3rd Generation Partnership Project (3GPP), “5G; NR; NR and NG-RAN Overall description,” 3GPP, Tech. Rep. TS 38.300, 2020.
- [21] X. Wang y T. Zhang, “Reinforcement Learning Based Resource Allocation for Network Slicing in 5G C-RAN,” in *2019 Computing, Communications and IoT Applications (ComComAp)*. Shenzhen, China: IEEE, Oct. 2019, pp. 106–111. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9018774/>
- [22] O. Vikhrova, C. Suraci, A. Tropeano, S. Pizzi, K. Samouylov, y G. Araniti, “Enhanced Radio Access Procedure in Sliced 5G Networks,” in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. Dublin, Ireland: IEEE, Oct. 2019, pp. 1–6. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8970776/>
- [23] S. Vural, N. Wang, G. Foster, y R. Tafazolli, “Success Probability of Multiple-Preamble-Based Single-Attempt Random Access to Mobile Networks,” *IEEE Communications Letters*, vol. 21, num. 8, pp. 1755–1758, Ago. 2017. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/7908954/>
- [24] S. Sundus Zehra, M. Magarini, R. Qureshi, S. Muhammad, N. Mustafa, y F. Farooq, “Proactive approach for preamble detection in 5G-NR PRACH using supervised machine learning and ensemble model,” 123, ISBN: 0123456789.
- [25] L. P. Tello Oquendo, “Design and Performance Analysis of Access Control Mechanisms for Massive Machine-to-Machine Communications in Wireless Cellular Networks,” Ph.D. dissertation, Universitat Politècnica de València, Valencia (Spain), Jul. 2018. [En línea]. Disponible: <https://riunet.upv.es/handle/10251/107946>
- [26] S. M. Sinche Minchalo y L. Tello Oquendo, “Diseño y análisis del canal de acceso aleatorio en redes IoT Celular para compensar la antigüedad de la información en escenarios de comunicaciones masivas.” Thesis, ESPOL. FIEC., 2021, accepted: 2022-06-06T14:31:17Z. [En línea]. Disponible: <http://www.dspace.espol.edu.ec/handle/123456789/53645>

- [27] 3rd Generation Partnership Project (3GPP), “Technical Specification Group Radio Access Network; NR; Medium Access Control (MAC) protocol specification,” 3GPP, Tech. Rep. TS 38.321, 2020.
- [28] —, “5G; 5GS; UE conformance specification; PART 1: Protocol,” 3GPP, Tech. Rep. TS 38.523, 2019.
- [29] S. I. C. López, “ANÁLISIS Y DISEÑO DE UN MECANISMO DE CONTROL DE CONGESTIÓN EN LA TECNOLOGÍA NB-IoT PARA DESPLIEGUES DE COMUNICACIÓN MASIVA DE INTERNET DE LAS COSAS.”
- [30] D. Chu, “Polyphase codes with good periodic correlation properties (corresp.),” *IEEE Transactions on Information Theory*, vol. 18, num. 4, pp. 531–532, 1972.
- [31] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, y J. Pineau, “An Introduction to Deep Reinforcement Learning,” *Foundations and Trends® in Machine Learning*, vol. 11, num. 3-4, pp. 219–354, 2018. [En línea]. Disponible: <http://www.nowpublishers.com/article/Details/MAL-071>
- [32] J. Aranda, E. J. Sacoto-Cabrera, D. Haro-Mendoza, y F. Astudillo-Salinas, “5g networks: A review from the perspectives of architecture, business models, cybersecurity, and research developments,” *Revista Digital Novasinerugia*, vol. 4, pp. 6–41, 6 2021. [En línea]. Disponible: [http://scielo.senescyt.gob.ec/scielo.php?script=sci\\_arttext&pid=S2631-26542021000100006&lng=en&nrm=iso&tlng=es](http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S2631-26542021000100006&lng=en&nrm=iso&tlng=es)  
[http://scielo.senescyt.gob.ec/scielo.php?script=sci\\_abstract&pid=S2631-26542021000100006&lng=en&nrm=iso&tlng=es](http://scielo.senescyt.gob.ec/scielo.php?script=sci_abstract&pid=S2631-26542021000100006&lng=en&nrm=iso&tlng=es)
- [33] J. S. Martín, “Impacto en la productividad por el uso de tecnologías 5G en Ecuador,” 2020.
- [34] I.-S. Comsa, A. De-Domenico, y D. Ktenas, “QoS-Driven Scheduling in 5G Radio Access Networks - A Reinforcement Learning Approach,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. Singapore: IEEE, Dic. 2017, pp. 1–7. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/8254926/>
- [35] “Machine learning, una pieza clave en la transformación de los modelos de negocio.”

- [36] S. Prasanna, "Machine Learning," *Machine Learning*.
- [37] N. Modina, "Physical Random Access Channel for New Radio."
- [38] S. Safavian y D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, num. 3, pp. 660–674, Jun. 1991. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/97458/>
- [39] C. A. Ruiz, M. S. Basualdo, y D. J. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones."
- [40] I. Roldán Bocanegra, "Aprendizaje semisupervisado," Jun. 2020, accepted: 2021-07-06T11:39:15Z. [En línea]. Disponible: <https://idus.us.es/handle/11441/115242>
- [41] L. P. Kaelbling, M. L. Littman, y A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996. [En línea]. Disponible: <https://www.jair.org/index.php/jair/article/view/10166>
- [42] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, y L.-C. Wang, "Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges," *IEEE Vehicular Technology Magazine*, vol. 14, num. 2, pp. 44–52, Jun. 2019. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8683970/>
- [43] V. Yajnanarayana, H. Ryden, y L. Hevizi, "5G Handover using Reinforcement Learning," in *2020 IEEE 3rd 5G World Forum (5GWF)*. Bangalore, India: IEEE, Sep. 2020, pp. 349–354. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9221072/>
- [44] Y. Shi, Y. E. Sagduyu, T. Erpek, y M. C. Gursoy, "How to Attack and Defend NextG Radio Access Network Slicing with Reinforcement Learning," Sep. 2022, arXiv:2101.05768 [cs]. [En línea]. Disponible: <http://arxiv.org/abs/2101.05768>
- [45] J. Tanveer, A. Haider, R. Ali, y A. Kim, "An Overview of Reinforcement Learning Algorithms for Handover Management in 5G Ultra-Dense Small Cell Networks," *Applied Sciences*, vol. 12, num. 1, p. 426, Ene. 2022. [En línea]. Disponible: <https://www.mdpi.com/2076-3417/12/1/426>

- [46] A. Sreedevi y T. Rama Rao, "Reinforcement learning algorithm for 5G indoor device-to-device communications," *Transactions on Emerging Telecommunications Technologies*, vol. 30, num. 9, Sep. 2019. [En línea]. Disponible: <https://onlinelibrary.wiley.com/doi/10.1002/ett.3670>
- [47] M. R. Raza, C. Natalino, P. Ohlen, L. Wosinska, y P. Monti, "Reinforcement Learning for Slicing in a 5G Flexible RAN," *Journal of Lightwave Technology*, vol. 37, num. 20, pp. 5161–5169, Oct. 2019. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8743435/>
- [48] Y. Hua, R. Li, Z. Zhao, X. Chen, y H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," *IEEE Journal on Selected Areas in Communications*, vol. 38, num. 2, pp. 334–349, Feb. 2020. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8931561/>
- [49] C. Ssengonzi, O. P. Kogeda, y T. O. Olwal, "A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization," *Array*, vol. 14, p. 100142, Jul. 2022. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S2590005622000133>
- [50] N. Akkari y N. Dimitriou, "Mobility management solutions for 5G networks: Architecture and services," *Computer Networks*, vol. 169, p. 107082, Mar. 2020. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1389128619306346>
- [51] A. A. Barakabitze, A. Ahmad, R. Mijumbi, y A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, Feb. 2020. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1389128619304773>
- [52] G. Baldoni, M. Melita, S. Micalizzi, C. Rametta, G. Schembra, y A. Vassallo, "Video Broadcasting Services Over SDN-NFV Enabled Networks: A Prototype," *Procedia Computer Science*, vol. 98, pp. 560–565, 2016. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1877050916322311>
- [53] M. Liyanage, P. Porambage, A. Y. Ding, y A. Kalla, "Driving forces for Multi-Access Edge Computing (MEC) IoT integration in 5G," *ICT*

- Express*, vol. 7, num. 2, pp. 127–137, Jun. 2021. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S2405959521000631>
- [54] M. L. F. Sindjoug, M. Velepini, y A. B. Bomgni, “A MEC architecture for a better quality of service in an Autonomous Vehicular Network,” *Computer Networks*, vol. 219, p. 109454, Dic. 2022. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1389128622004881>
- [55] D. Alotaibi, V. Thayananthan, y J. Yazdani, “The 5G network slicing using SDN based technology for managing network traffic,” *Procedia Computer Science*, vol. 194, pp. 114–121, 2021. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1877050921021050>
- [56] A. Rago, S. Martiradonna, G. Piro, A. Abrardo, y G. Boggia, “A tenant-driven slicing enforcement scheme based on Pervasive Intelligence in the Radio Access Network,” *Computer Networks*, vol. 217, p. 109285, Nov. 2022. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S1389128622003413>
- [57] P. Popovski, K. F. Trillingsgaard, O. Simeone, y G. Durisi, “5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View,” Ago. 2018, arXiv:1804.05057 [cs, math]. [En línea]. Disponible: <http://arxiv.org/abs/1804.05057>
- [58] K. Boutiba, A. Ksentini, B. Brik, Y. Challal, y A. Balla, “NRflex: Enforcing network slicing in 5G New Radio,” *Computer Communications*, vol. 181, pp. 284–292, Ene. 2022. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S0140366421003716>
- [59] S. Bakri, P. A. Frangoudis, y A. Ksentini, “Dynamic Slicing of RAN Resources for Heterogeneous Coexisting 5G Services,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, HI, USA: IEEE, Dic. 2019, pp. 1–6. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9013954/>
- [60] S. Zhang, “An Overview of Network Slicing for 5G,” *IEEE Wireless Communications*, vol. 26, num. 3, pp. 111–117, Jun. 2019. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8685766/>

- [61] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, y X. Costa-Perez, "A Machine Learning Approach to 5G Infrastructure Market Optimization," *IEEE Transactions on Mobile Computing*, vol. 19, num. 3, pp. 498–512, Mar. 2020. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8632676/>
- [62] M. R. Raza, C. Natalino, L. Wosinska, y P. Monti, "Machine Learning Methods for Slice Admission in 5G Networks," in *2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC)*. Fukuoka, Japan: IEEE, Jul. 2019, pp. 1–3. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/8817990/>
- [63] R. Kountchev, A. Mahanti, S. Chong, S. Patnaik, y M. Favorskaya, Eds., *Advances in Wireless Communications and Applications: Smart Communications: Interactive Methods and Intelligent Algorithms, Proceedings of 3rd ICWCA 2019*, ser. Smart Innovation, Systems and Technologies. Singapore: Springer Singapore, 2021, vol. 190. [En línea]. Disponible: <http://link.springer.com/10.1007/978-981-15-5697-5>
- [64] D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs, y X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," *IEEE Communications Magazine*, vol. 58, num. 6, pp. 32–38, Jun. 2020. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/9141212/>
- [65] H. Zhou, M. Elsayed, y M. Erol-Kantarci, "RAN Resource Slicing in 5G Using Multi-Agent Correlated Q-Learning," Jun. 2021, arXiv:2107.01018 [cs, eess]. [En línea]. Disponible: <http://arxiv.org/abs/2107.01018>
- [66] Y. Shi, Y. E. Sagduyu, y T. Erpek, "Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing," Sep. 2020, arXiv:2009.06579 [cs]. [En línea]. Disponible: <http://arxiv.org/abs/2009.06579>
- [67] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, y X. Costa-Pérez, "Resource sharing efficiency in network slicing," *IEEE Transactions on Network and Service Management*, vol. 16, num. 3, pp. 909–923, 2019.
- [68] A. Laya, L. Alonso, y J. Alonso-Zarate, "Is the random access channel of lte and

- lte-a suitable for m2m communications? a survey of alternatives,” *IEEE Communications Surveys & Tutorials*, vol. 16, num. 1, pp. 4–16, 2014.
- [69] S. Vural, N. Wang, P. Bucknell, G. Foster, R. Tafazolli, y J. Muller, “Dynamic preamble subset allocation for ran slicing in 5g networks,” *IEEE Access*, vol. 6, pp. 13 015–13 032, 2018.
- [70] O. Vikhrova, C. Suraci, A. Tropeano, S. Pizzi, K. Samouylov, y G. Araniti, “Enhanced radio access procedure in sliced 5g networks,” in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2019, pp. 1–6.
- [71] H. Althumali, M. Othman, N. K. Noordin, y Z. M. Hanapi, “Priority-based load-adaptive preamble separation random access for qos-differentiated services in 5g networks,” *Journal of Network and Computer Applications*, vol. 203, p. 103396, 2022.
- [72] H.-Y. Hwang, S.-M. Oh, C. Lee, J. H. Kim, y J. Shin, “Dynamic rach preamble allocation scheme,” in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2015, pp. 770–772.
- [73] A. M. Gedikli, M. Koseoglu, y S. Sen, “Deep reinforcement learning based flexible preamble allocation for ran slicing in 5g networks,” *Computer Networks*, vol. 215, p. 109202, 2022.
- [74] 3rd Generation Partnership Project (3GPP), “Technical Specification Group Radio Access Network; NR; Multiplexing and channel coding,” 3GPP, Tech. Rep. TS 38.212, 2017.
- [75] —, “Technical Specification Group Radio Access Network; Study on RAN Improvements for Machine-type Communications,” 3GPP, Tech. Rep. TR 37.868, 2011.
- [76] J. Ferrer, “DISTRIBUCIÓN DE PROBABILIDAD BETA Y SUS USO PARA MODELAR CURVA “S” DE UN PROYECTO,” may 2023. [En línea]. Disponible: <https://es.linkedin.com/pulse/distribuci{ó}n-de-probabilidad-beta-y-sus-uso-para-modelar-jos{é}-ferrer>

- [77] S. M. Sinche Minchalo, “Diseño y análisis del canal de acceso aleatorio en redes IoT Celular para compensar la antigüedad de la información en escenarios de comunicaciones masivas.” Ph.D. dissertation, Escuela Superior Politécnica del Litoral, 2021.
- [78] S. Calixto Aldama y R. M. Valdovinos Rosas, “ESTUDIO COMPARATIVO DE HERRAMIENTAS PARA REDES NEURONALES ARTIFICIALES (RNA): WEKA, MATLAB Y NEUROSOLUTIONS,” Tesis de Licenciatura, Universidad Autónoma del Estado de México, 2018. [En línea]. Disponible: <http://hdl.handle.net/20.500.11799/98901>
- [79] M. Sequeira, V. Cortínez, y A. Azzurro, “Caracterización Acústica de Ambientes Urbanos Mediante el Uso de Redes Neuronales | Sequeira | Mecánica Computacional,” *Asociación Argentina de mecánica computacional*, vol. 01, pp. 97–105, 2019.
- [80] J. U. R. A. A. P. Estrada y R. G. Martínez, “Empleo de redes neuronales artificiales para la obtención de espectros de respuesta de registros sísmicos de eventos de subducción y de falla normal,” num. September, p. 10, 2017.
- [81] H. Tabares y J. Hernández, “Mapeo curvas típicas demanda de energía eléctrica del sector residencial, comercial e industrial de la ciudad de Medellín, usando redes neuronales artificiales y algoritmos de interpolación,” *Revista Facultad de Ingeniería*, num. 46, pp. 110–118, 2008.
- [82] ———, “Mapeo curvas típicas demanda de energía eléctrica del sector residencial, comercial e industrial de la ciudad de Medellín, usando redes neuronales artificiales y algoritmos de interpolación,” *Revista Facultad de Ingeniería*, num. 46, pp. 110–118, 2008.
- [83] J. Mca, “Aplicación de diferentes arquitecturas de RNA sobre datos de paquetes de red en Matlab con y sin reducción de,” num. October, 2014.
- [84] Veintimilla-Reyes Jaime y Cisneros Felipe, “Redes Neuronales Artificiales (RNA) aplicadas en la predicción de caudales para intervalos de tiempo horarios,” *Revista Tecnológica ESPOL – RTE*, vol. 27, num. Diciembre, pp. 37–49, 2014.

- [85] A. M. Andrew, “Reinforcement learning: An introduction by richard s. sutton and andrew g. barto, adaptive computation and machine learning series, mit press (bradford book), cambridge, mass., 1998, xviii+ 322 pp, isbn 0-262-19398-1,(hardback,£ 31.95).” *Robotica*, vol. 17, num. 2, pp. 229–235, 1999.
- [86] V. Raykar y P. Agrawal, “Sequential crowdsourced labeling as an epsilon-greedy exploration in a Markov Decision Process,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Kaski y J. Corander, Eds., vol. 33. Reykjavik, Iceland: PMLR, 22–25 Apr 2014, pp. 832–840. [En línea]. Disponible: <https://proceedings.mlr.press/v33/raykar14.html>

**Apéndice A Configuración de Acceso Aleatorio para formatos de preámbulo  
0-3**

Tabla A.1: Configuración de acceso aleatorio para formatos de preámbulo 0-3 (Tabla tomada de [10])

<b>Índice de configuración PRACH</b>	<b>Formato de Preámbulo</b>	<b>Número de trama del Sistema</b>	<b>Número de Subtrama</b>
0	0	Par	1
1	0	Par	4
2	0	Par	7
3	0	Cualquiera	1
4	0	Cualquiera	4
5	0	Cualquiera	7
6	0	Cualquiera	1, 6
7	0	Cualquiera	2, 7
8	0	Cualquiera	3, 8
9	0	Cualquiera	1, 4, 7
10	0	Cualquiera	2, 5, 8
11	0	Cualquiera	3, 6, 9
12	0	Cualquiera	0, 2, 4, 6, 8
13	0	Cualquiera	1, 3, 5, 7, 9
14	0	Cualquiera	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
15	0	Igual	9
16	1	Igual	1
17	1	Igual	4
18	1	Igual	7
19	1	Cualquiera	1
20	1	Cualquiera	4
21	1	Cualquiera	7

*(Continúa en la página siguiente)*

Indice de configuración PRACH	Formato de Preámbulo	Número de trama del Sistema	Número de Subtrama
22	1	Cualquiera	1, 6
23	1	Cualquiera	2, 7
24	1	Cualquiera	3, 8
25	1	Cualquiera	1, 4, 7
26	1	Cualquiera	2, 5, 8
27	1	Cualquiera	3, 6, 9
28	1	Cualquiera	0, 2, 4, 6, 8,
29	1	Cualquiera	1, 3, 5, 7, 9
30	N/A	N/A	N/A
31	1	Igual	9
32	2	Igual	1
33	2	Igual	4
34	2	Igual	7
35	2	Cualquiera	1
36	2	Cualquiera	4
37	2	Cualquiera	7
38	2	Cualquiera	1, 6
39	2	Cualquiera	2, 7
40	2	Cualquiera	3, 8
41	2	Cualquiera	1, 4, 7
42	2	Cualquiera	2, 5, 8
43	2	Cualquiera	3, 6, 9
44	2	Cualquiera	0, 2, 4, 6, 8
45	2	Cualquiera	1, 3, 5, 7, 9
46	N/A	N/A	N/A
47	2	Igual	9
48	3	Igual	1
49	3	Igual	4

*(Continúa en la página siguiente)*

Indice de configuración PRACH	Formato de Preámbulo	Número de trama del Sistema	Número de Subtrama
50	3	Igual	7
51	3	Cualquiera	1
52	3	Cualquiera	4
53	3	Cualquiera	7
54	3	Cualquiera	1, 6
55	3	Cualquiera	2, 7
56	3	Cualquiera	3, 8
57	3	Cualquiera	1, 4, 7
58	3	Cualquiera	2, 5, 8
59	3	Cualquiera	3, 6, 9
60	N/A	N/A	N/A
61	N/A	N/A	N/A
62	N/A	N/A	N/A
63	3	Igual	9

**Apéndice B Recompensas y penalizaciones asociadas a cada estado del sistema**

<b>Matriz de recompensas para <math>CV_{N_{UEs}} = 2</math> y <math>\Delta N_{UEs} = 2</math></b>										
		$mN_{UEs}$								
		0	1	2-3	4-7	8-11	12-14	15-19	20-25	26-30
		UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs
$N_{PA}$	<b>1-3 preámbulos</b>	100	50	-20	-40	-50	-60	-70	-80	-90
	<b>4-8 preámbulos</b>	-20	90	20	-30	-40	-50	-60	-70	-80
	<b>9-14 preámbulos</b>	-30	-20	80	10	-30	-40	-50	-60	-70
	<b>15-18 preámbulos</b>	-40	-30	-10	70	-10	-30	-40	-50	-60
	<b>19-24 preámbulos</b>	-50	-40	-30	0	60	-20	-30	-40	-50
	<b>25-30 preámbulos</b>	-60	-50	-40	-30	20	50	-10	-30	-40
	<b>31-38 preámbulos</b>	-70	-60	-50	-40	-20	10	40	-20	-30
	<b>39-49 preámbulos</b>	-80	-70	-60	-50	-30	-10	0	30	-10
	<b>50-52 preámbulos</b>	-90	-80	-70	-60	-50	-40	-30	0	20
<b>Matriz de recompensas para <math>CV_{N_{UEs}} = 2</math> y <math>\Delta N_{UEs} = 3</math></b>										
		$mN_{UEs}$								
		0	1	2-3	4-7	8-11	12-14	15-19	20-25	26-30
		UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs
	<b>1-3 preámbulos</b>	100	20	-30	-50	-60	-70	-80	-90	-100

	<b>4-8 preámbulos</b>	-20	90	-10	-30	-50	-60	-70	-80	-90
	<b>9-14 preámbulos</b>	-30	-10	80	-20	-40	-50	-60	-70	-800
	<b>15-18 preámbulos</b>	-40	-30	0	70	-30	-40	-50	-60	-70
	<b>19-24 preámbulos</b>	-50	-40	-30	10	60	-30	-40	-50	-60
	<b>25-30 preámbulos</b>	-60	-50	-40	-20	30	50	-30	-40	-50
	<b>31-38 preámbulos</b>	-70	-60	-50	-40	-30	20	40	-30	-40
	<b>39-49 preámbulos</b>	-80	-70	-60	-50	-40	-20	10	30	-20
	<b>50-52 preámbulos</b>	-90	-80	-70	-60	-50	-40	-30	10	20
<b>Matriz de recompensas para <math>CV_{N_{UEs}} = 2</math> y <math>\Delta N_{UEs} = 1</math></b>										
		$mN_{UEs}$								
		<b>0 UEs</b>	<b>1 UEs</b>	<b>2-3 UEs</b>	<b>4-7 UEs</b>	<b>8-11 UEs</b>	<b>12-14 UEs</b>	<b>15-19 UEs</b>	<b>20-25 UEs</b>	<b>26-30 UEs</b>
$N_{PA}$	<b>1-3 preámbulos</b>	100	20	-30	-50	-60	-70	-80	-90	-100
	<b>4-8 preámbulos</b>	-20	90	-10	-30	-50	-60	-70	-80	-90
	<b>9-14 preámbulos</b>	-30	-10	80	-20	-40	-50	-60	-70	-800
	<b>15-18 preámbulos</b>	-40	-30	0	70	-30	-40	-50	-60	-70
	<b>19-24 preámbulos</b>	-50	-40	-30	10	60	-30	-40	-50	-60
	<b>25-30 preámbulos</b>	-60	-50	-40	-20	30	50	-30	-40	-50

	<b>31-38 preámbulos</b>	-70	-60	-50	-40	-30	20	40	-30	-40
	<b>39-49 preámbulos</b>	-80	-70	-60	-50	-40	-20	10	30	-20
	<b>50-52 preámbulos</b>	-90	-80	-70	-60	-50	-40	-30	10	20

**Matriz de recompensas para  $CV_{N_{UEs}} = 1$  y  $\Delta N_{UEs} = 2$**

		$mN_{UEs}$								
		0	1	2-3	4-7	8-11	12-14	15-19	20-25	26-30
		UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs
$N_{PA}$	<b>1-3 preámbulos</b>	90	40	-20	-40	60	-70	-80	-90	-100
	<b>4-8 preámbulos</b>	-20	80	10	-30	-40	-60	-70	-80	-90
	<b>9-14 preámbulos</b>	-30	30	70	0	-30	-50	-60	-70	-80
	<b>15-18 preámbulos</b>	-40	-30	20	60	-20	-40	-50	-60	-70
	<b>19-24 preámbulos</b>	-50	-40	-30	20	50	-30	-40	-50	-60
	<b>25-30 p reámbulos</b>	-60	-50	-40	-30	30	40	-20	-30	-50
	<b>31-38 preámbulos</b>	-70	-60	-50	-40	-10	20	30	-20	-40
	<b>39-49 preámbulos</b>	-80	-70	-60	-50	-30	-10	10	20	-20
	<b>50-52 preámbulos</b>	-90	-80	-70	-60	-50	-40	-30	10	10

**Matriz de recompensas para  $CV_{N_{UEs}} = 1$  y  $\Delta N_{UEs} = 3$**

		$mN_{UEs}$								
		0	1	2-3	4-7	8-11	12-14	15-19	20-25	26-30
		UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs	UEs

$N_{PA}$	<b>1-3 preámbulos</b>	90	30	-20	-40	60	-70	-80	-90	-100
	<b>4-8 preámbulos</b>	-10	80	0	-30	-50	-60	-70	-80	-90
	<b>9-14 preámbulos</b>	-30	40	70	-10	-40	-50	-60	-70	-80
	<b>15-18 preámbulos</b>	-40	-30	30	60	-30	-40	-50	-60	-70
	<b>19-24 preámbulos</b>	-50	-40	-30	20	50	-30	-40	-50	-60
	<b>25-30 preámbulos</b>	-60	-50	-40	-30	30	40	-30	-40	-50
	<b>31-38 preámbulos</b>	-70	-60	-50	-40	-10	30	30	-30	-40
	<b>39-49 preámbulos</b>	-80	-70	-60	-50	-30	-10	20	20	-30
	<b>50-52 preámbulos</b>	-90	-80	-70	-60	-50	-40	-30	20	10
<b>Matriz de recompensas para <math>CV_{N_{UEs}} = 1</math> y <math>\Delta N_{UEs} = 1</math></b>										
		$mN_{UEs}$								
		<b>0 UEs</b>	<b>1 UEs</b>	<b>2-3 UEs</b>	<b>4-7 UEs</b>	<b>8-11 UEs</b>	<b>12-14 UEs</b>	<b>15-19 UEs</b>	<b>20-25 UEs</b>	<b>26-30 UEs</b>
$N_{PA}$	<b>1-3 preámbulos</b>	90	10	-30	-50	60	-70	-80	-90	-100
	<b>4-8 preámbulos</b>	-10	80	-10	-40	-50	-60	-70	-80	-90
	<b>9-14 preámbulos</b>	-40	40	70	-30	-40	-50	-60	-70	-80
	<b>15-18 preámbulos</b>	-50	-40	30	60	-30	-40	-50	-60	-70

<b>19-24 preámbulos</b>	-60	-50	-30	20	50	-30	-40	-50	-60
<b>25-30 preámbulos</b>	-70	-60	-40	-30	30	40	-30	-40	-50
<b>31-38 preámbulos</b>	-80	-70	-50	-40	0	30	30	-30	-40
<b>39-49 preámbulos</b>	-90	-80	-60	-50	-20	0	20	20	-30
<b>50-52 preámbulos</b>	-100	-90	-80	-70	-60	-50	-40	20	10