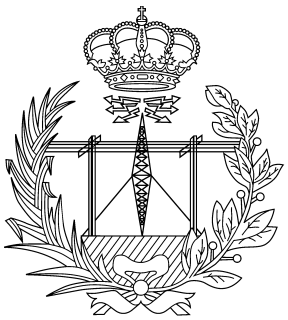
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



***Trabajo Fin de Grado***

# Aplicación de técnicas de aprendizaje de refuerzo profundo para la gestión de recursos radio en redes 5G

(On the use of DRL for radio resource management in 5G networks)

Para acceder al Título de

***Graduado en***
***Ingeniería de Tecnologías de Telecomunicación***

Autor: Enrique Lahuerta López-Lapuente

Septiembre- 2025

UNIVERSIDAD DE CANTABRIA

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

# GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

**Realizado por:** Enrique Lahuerta López-Lapuente
**Director del TFG:** Ramón Agüero Calvo, Luis Francisco Diez Fernández
**Título:** "Aplicación de técnicas de aprendizaje de refuerzo profundo para la gestión de recursos radio en redes 5G"
**Title:** "On the use of DRL for radio resource management in 5G networks"

**Presentado a examen el día:** 17 de septiembre de 2025

para acceder al Título de

# GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del tribunal:
Presidente (Apellidos, Nombre): Ruiz Robredo, Gustavo
Secretario (Apellidos, Nombre): Ramírez Terán, Franco Ariel
Vocal (Apellidos, Nombre): García Arranz, Marta

Este Tribunal ha resulto otorgar la calificación de: .....

| | |
|---|---|
| Fdo.: El Presidente | Fdo.: El Secretario |
| Fdo.: El Vocal | Fdo.: El Director del TFG<br>(solo si es distinto del Secretario) |
| V$^{o}$ B$^{o}$ del Subdirector | Trabajo Fin de Grado N$^{o}$<br>(a asignar por Secretaría) |

**Resumen**

Este trabajo fin de grado presenta el desarrollo de un agente inteligente basado en aprendizaje por refuerzo profundo para optimizar la asignación de recursos en la capa MAC de redes móviles 5G. Se ha diseñado un entorno simulado que permite representar distintos estados del sistema y se ha generado un conjunto de datos tabulados para entrenar modelos con los algoritmos PPO y DQN. El estudio incluye un proceso de optimización automatizada de hiperparámetros mediante la librería Optuna, aplicado a ocho escenarios distintos. Los resultados obtenidos muestran que los agentes son capaces de aprender políticas efectivas, generalizar su comportamiento en fase de verificación y mejorar métricas como la recompensa media y la estabilidad del sistema. El enfoque propuesto demuestra la viabilidad de aplicar técnicas de inteligencia artificial en procesos internos de gestión de redes móviles de nueva generación.

**Abstract**

This bachelor thesis presents the development of an intelligent agent based on deep reinforcement learning to optimize resource allocation in the MAC layer of 5G mobile networks. A simulated environment was designed to represent various system states, and a tabulated dataset was generated to train models using the PPO and DQN algorithms. The study includes an automated hyperparameter optimization process using the Optuna library, applied across eight different scenarios. The results show that the agents are capable of learning effective policies, generalizing their behavior during testing, and improving metrics such as average reward and system stability. The proposed approach demonstrates the feasibility of applying artificial intelligence techniques to internal management processes in next-generation mobile networks.

# Índice general

# Índice de figuras

# Índice de tablas

*Capítulo 1*

# Introducción

La consolidación de las redes móviles Fifth Generation (5G) ha marcado un punto de inflexión en el desarrollo de sistemas de comunicación inalámbrica. Esta tecnología no solo mejora drásticamente la velocidad de transmisión y la latencia, sino que también permite conectar simultáneamente millones de dispositivos, abriendo la puerta a aplicaciones avanzadas en sectores como la automoción, la sanidad, la industria o el entretenimiento. Sin embargo, esta evolución también implica una mayor complejidad en la gestión interna de la red, especialmente en lo que respecta a la asignación dinámica de recursos.

En este contexto, la gestión eficiente de los recursos radioeléctricos —como el ancho de banda, el tiempo de transmisión o la potencia transmitida— se convierte en un aspecto crítico para garantizar el rendimiento global del sistema. La toma de decisiones en tiempo real, en entornos densos y variables, requiere soluciones que sean capaces de adaptarse de forma continua y optimizar múltiples objetivos simultáneamente.

## 1.1. Contexto general del proyecto

Este trabajo se sitúa en la capa Media Access Control (MAC) de una red 5G, donde se gestionan las decisiones relacionadas con el acceso al medio por parte de los usuarios. En particular, se estudia cómo ajustar ciertos parámetros internos del planificador (scheduler) para mejorar el comportamiento del sistema en función del estado de la red. La investigación se desarrolla sobre un entorno simulado, utilizando datos que representan distintas configuraciones y métricas de rendimiento.

El enfoque adoptado combina simulación, inteligencia artificial y análisis de datos para abordar un problema complejo desde una perspectiva experimental. El objetivo no es replicar un sistema comercial, sino explorar el potencial de técnicas avanzadas para mejorar la toma de decisiones dentro de un componente clave de la arquitectura 5G.

## 1.2. Motivación y enfoque

La motivación principal de este proyecto radica en la necesidad de encontrar soluciones adaptativas que respondan eficazmente a las condiciones cambiantes del entorno. Frente a los métodos tradicionales, que suelen basarse en reglas estáticas o estrategias heurísticas, se propone el uso de aprendizaje por refuerzo

profundo como herramienta para entrenar agentes capaces de aprender directamente de los conocimientos previos.

Este enfoque permite modelar la asignación de recursos como un proceso de decisión, en el que el agente observa el estado del sistema, toma una acción y recibe una recompensa que guía su aprendizaje. La capacidad de adaptación y mejora continua convierte esta técnica en una candidata idónea para entornos como la capa MAC, donde la variabilidad es constante y las decisiones deben tomarse en tiempo real.

## 1.3. Objetivo del trabajo

El objetivo de este trabajo fin de grado es desarrollar y evaluar un agente inteligente capaz de optimizar la asignación de recursos en la capa MAC de una red 5G. Para ello, se ha diseñado un entorno simulado, se ha generado un dataset, y se han aplicado algoritmos de aprendizaje por refuerzo profundo junto con técnicas de optimización automatizada de hiperparámetros.

El estudio se centra en analizar el comportamiento del agente, su capacidad de aprendizaje y su rendimiento en distintos escenarios, con el fin de validar la viabilidad del enfoque propuesto y extraer conclusiones útiles para futuras investigaciones.

## 1.4. Estructura del documento

El documento se organiza en cinco capítulos. Tras esta introducción, el Capítulo 2 presenta los fundamentos teóricos necesarios para contextualizar el estudio. El Capítulo 3 describe el diseño experimental, incluyendo el entorno, el agente y el proceso de entrenamiento. El Capítulo 4 recoge los resultados obtenidos y su análisis, y el Capítulo 5 expone las conclusiones finales y posibles líneas de mejora.

*Capítulo 2*

# Antecedentes

En las siguientes secciones se hace un breve resumen de la evolución de las diferentes generaciones de tecnologías celulares. Posteriormente, se presenta la problemática de gestión de radio en redes 5G y, finalmente, se realiza una introducción de los conceptos básicos de Deep Reinforcement Learning (DRL).

## 2.1. Historia de las Redes Inalámbricas

La evolución de las redes celulares ha sido fundamental en la transformación del panorama actual de las telecomunicaciones. De forma general, están compuestas por elementos de acceso y el núcleo de red. Los elementos de acceso o estaciones celulares se distribuyen estratégicamente a lo largo de amplias áreas geográficas, dando lugar a zonas delimitadas denominadas celdas. Cada celda cuenta con uno o varios transceptores que proporcionan cobertura radioeléctrica en su área correspondiente, generalmente ubicados en estaciones base.

Con cada nueva generación de redes celulares, surgen nuevos requisitos, tecnologías y soluciones, lo que ha impulsado avances significativos en el sector. En la actualidad, las redes celulares se han consolidado como una infraestructura esencial para numerosas aplicaciones, entre ellas los dispositivos del Internet of Things (IoT) y la robótica. Actualmente, con el desarrollo incipiente de las tecnologías 6G, se abren nuevas perspectivas y oportunidades para el futuro.

A continuación se resumen los principales avances que marcaron y siguen marcando la evolución de las redes móviles, y sus principales características. [1]

### 2.1.1. Primera Generación (1G)

La primera generación (First Generation (1G)) apareció en los años 80 y fue totalmente analógica. Basada en Frequency Division Multiple Access (FDMA), cada usuario ocupaba un canal de frecuencia. Aunque esta tecnología permitió la proliferación inicial de la telefonía móvil, presentaba numerosas limitaciones tanto en calidad de voz, capacidad o seguridad.

### 2.1.2. Segunda Generación (2G)

La segunda generación (Second Generation (2G)), introducida a inicios de los 90 con el estándar Global System for Mobile Communications (GSM), supuso la transición a sistemas digitales. Se utilizó Time

Division Multiple Access (TDMA) como esquema de acceso múltiple, permitiendo compartir un canal entre varios usuarios en diferentes intervalos de tiempo. Además, se incorporó mensajería Short Message Service (SMS) y un uso más eficiente del espectro.

### 2.1.3. Tercera Generación (3G)

La tercera generación (Third Generation (3G)) generalizó los servicios de datos móviles, tales como videollamadas, navegación o streaming. Basada en Universal Mobile Telecommunications System (UMTS), introdujo Code Division Multiple Access (CDMA) como técnica de acceso múltiple, aumentando capacidad y robustez frente a interferencias. Conforme a la recomendación International Mobile Telecommunications (IMT)-2000, ofrecía velocidades de hasta 2 Mbps y compatibilidad con redes 2G en el núcleo de la red, facilitando una transición tecnológica progresiva para los operadores.

### 2.1.4. Cuarta Generación (4G)

La cuarta generación (Fourth Generation (4G)), que se desplegó mayoritariamente no el nombre de Long Term Evolution (LTE), nació para superar las limitaciones de 3G en velocidad y latencia. Gracias al esquema de acceso múltiple basado en Orthogonal Frequency Division Multiplexing (OFDM) y otras técnicas como Multiple-Input Multiple-Output (MIMO), se alcanzaron velocidades superiores a 20 Mbps y se dio soporte a aplicaciones en tiempo real (videoconferencias, juegos en línea), mejorando notablemente la experiencia de usuario.

### 2.1.5. Quinta Generación (5G)

La red 5G surge como respuesta a las limitaciones del 4G ante la creciente demanda de velocidad, capacidad, baja latencia y eficiencia energética. De acuerdo a los requisitos establecidos por la Unión Internacional de Telecomunicaciones (UIT) bajo el estándar IMT-2020, las tecnologías 5G están diseñadas para soportar aplicaciones modernas con necesidades cada vez más exigentes.

A nivel de arquitectura 5G introduce grandes cambios que se pueden resumir como:

- Separación entre plano de control y plano de usuario, lo que mejora la gestión del tráfico y la escalabilidad.
- Arquitectura basada en servicios, que habilita la modularidad, integración sencilla de nuevas funciones y personalización de servicios según necesidades específicas.

En la Figura 2.1 se presenta una visión general de los componentes de la red 5G, incluyendo tanto los elementos de acceso como el núcleo o core. Entre los componentes clave del core de 5G destacan:

- **UPF (User Plane Function)**: gestiona el tráfico de datos entre el dispositivo y la red.
- **SMF (Session Management Function)**: controla las sesiones del usuario e IPs asignadas.
- **AMF (Access and Mobility Management Function)**: gestiona las conexiones y movilidad de los dispositivos.

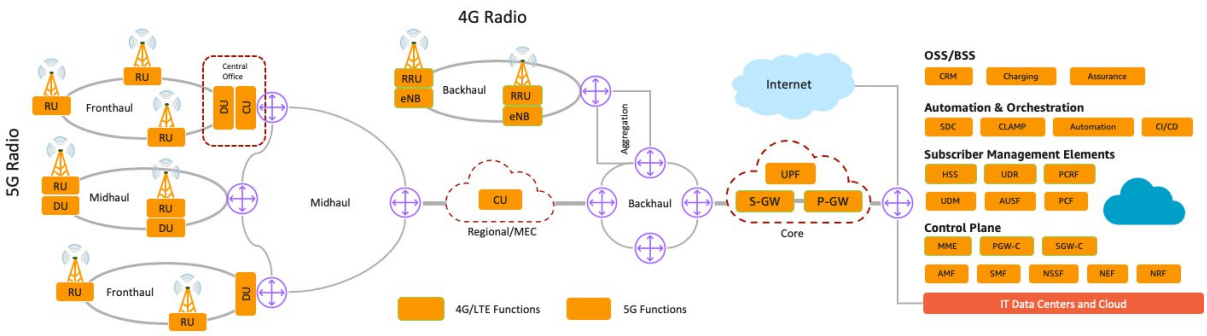



Figura 2.1: Esquema de la arquitectura 5G [2]

El core 5G también incluye funciones como Network Exposure Function (NEF), Network Repository Function (NRF) y Policy Control Function (PCF), que optimizan la exposición, gestión de recursos y políticas de red. Además, tal como se muestra en la Figura 2.2, la tecnología 5G se ha diseñado para cubrir las necesidades de varios tipos de servicio que se agrupan en las siguientes categorías:

- **eMBB (enhanced Mobile Broadband)**: se caracterizan por altas tasas de datos para aplicaciones como vídeo 4K/8K o realidad virtual.
- **URLLC (Ultra Reliable Low Latency Communications)**: comunicaciones críticas con ultra baja latencia, como en automatización industrial o vehículos conectados.
- **mMTC (massive Machine-Type Communications)**: conectividad masiva de dispositivos IoT de bajo consumo.

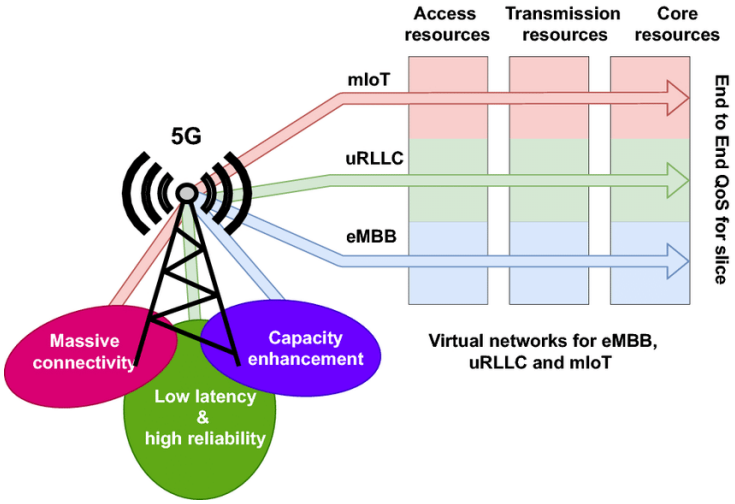



Figura 2.2: Principales servicios soportados por 5G: eMBB,URLLC y mMTC [3].

5G representa un salto tecnológico que prepara el camino hacia Sixth Generation (6G), permitiendo nuevas aplicaciones en sectores como la industria, salud, transporte o el Internet de las Cosas (IoT). La evolución continúa, con investigaciones ya centradas en definir las redes del futuro.

Una de las líneas de trabajo activas en la actualidad es la correcta asignación de recursos de red. Este trabajo se centra en este tema, en el que se utilizan técnicas de Inteligencia Artificial, como el DRL, para

realizar esta asignación de recursos entre los propios usuarios de la red de manera óptima en función del estado de las colas de tráfico de cada uno de ellos, de los recursos disponibles, así como del nivel de *Quality of Service (QoS)* requerido, siendo estos los diferentes parámetros sobre los que se profundizará más adelante.

### 2.1.6. Sexta Generación (6G)

La futura red 6G, prevista para su despliegue en torno al año 2030, promete superar ampliamente las capacidades de 5G, ofreciendo mayor ancho de banda, latencias aún más bajas y mayor eficiencia en la gestión de datos masivos y complejos. Grandes empresas del sector como Ericsson, Samsung y Nokia ya lideran la investigación en esta área. [4] [5]

Uno de los principales impulsores de 6G es el crecimiento exponencial del big data y la necesidad de acceso inalámbrico ubicuo en una sociedad cada vez más digitalizada. La arquitectura 6G partirá de la base de 5G, pero se cree que incorporará elementos disruptivos entre los que destacan:

- **Integración nativa de Inteligencia Artificial (IA)**: redes auto-optimizadas, capaces de aprender y adaptarse en tiempo real para mejorar la asignación de recursos, la movilidad y la eficiencia energética.
- **Comunicación integrada con sensado (ISAC, Integrated Sensing and Communication)**: fusión de capacidades de radar y comunicación en la misma infraestructura, habilitando aplicaciones como localización de alta precisión, monitorización ambiental y soporte a vehículos autónomos.
- **Redes descentralizadas y sostenibles**: diseñadas para reducir el consumo energético y favorecer la resiliencia de la conectividad.

Un ejemplo claro del potencial de 6G es su aplicación en vehículos autónomos, que requieren conectividad en tiempo real, ultra baja latencia y redes adaptativas para reaccionar al entorno y procesar datos instantáneamente.

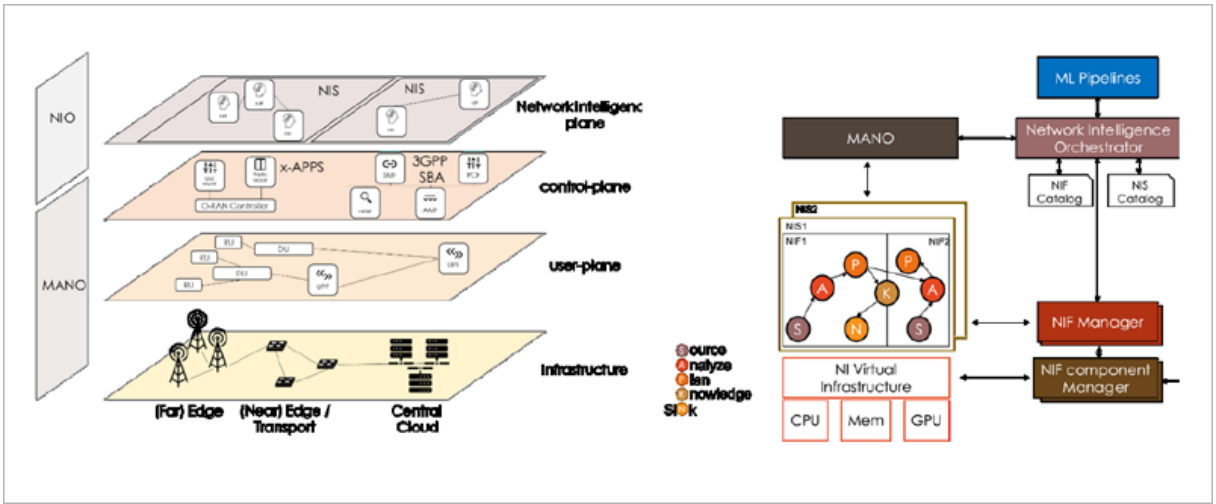



Figura 2.3: Modelo arquitectónico 6G (Proyecto Daemon) [6]

Con el objetivo principal de demostrar cómo la Inteligencia Artificial (IA) puede integrarse de forma práctica en las arquitecturas de redes móviles a nivel de producción y ayudar a automatizar la gestión de los sistemas más allá de 5G y 6G, el proyecto europeo aDAptive and sElf-Learning MObile Networks (Daemon) ha desarrollado un modelo arquitectónico nativo de inteligencia artificial (ver Figura 2.3), que ha sido adoptado por los grupos de trabajo de arquitectura 5G PPP y 6G IA. La arquitectura nativa Network Intelligence (NI) de Daemon introduce la inteligencia directamente en el plano del usuario, creando una jerarquía de instancias de NI para la gestión de red.

## 2.2. Los Recursos en Redes 5G: Gestión de los PRBs y su Optimización

En las redes 5G, uno de los elementos fundamentales para garantizar el correcto funcionamiento del sistema y el cumplimiento de los requisitos de QoS es la gestión eficiente de los recursos radio, en particular los denominados *Physical Resource Block (PRB)*. Estos bloques de recursos físicos representan unidades mínimas de espectro que pueden ser asignadas a diferentes flujos de datos y su correcta distribución es clave para lograr un alto rendimiento de la red [1].

### 2.2.1. Definición de PRB y asignación

Los PRBs son las unidades básicas en las que se divide la banda de frecuencia disponible en 5G para poder transmitir datos. Cada uno de estos bloques se puede asignar a un flujo de tráfico de un usuario y la cantidad de PRBs disponibles en cada instante depende de la configuración del sistema, tales como el ancho de banda o la numerología usada, entre otros parámetros.

Como se puede observar en la Figura 2.4, un PRB es un conjunto de frecuencias portadoras que se le pueden asignar a un usuario con el fin de aumentar su tasa binaria y, consecuentemente, reducir el tamaño actual de las colas.

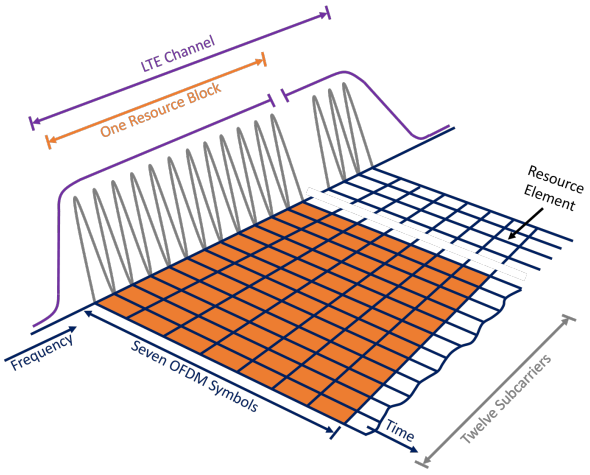



Figura 2.4: PRB (Physycal Resource Block) [7]

En el contexto de la capa MAC, el proceso de planificación (*scheduling*) se encarga de decidir cómo se reparten estos PRBs entre los distintos usuarios y servicios. Este proceso se realiza en intervalos muy

cortos de tiempo (del orden de milisegundos), por lo que debe ser muy eficiente y adaptativo. Además, debe tener en cuenta el estado del canal, la calidad del enlace y las necesidades particulares de cada flujo.

### 2.2.2. PRBs y calidad de servicio (QoS)

Uno de los grandes desafíos en 5G es que los servicios son muy diversos y presentan por tanto requisitos muy diferentes. Algunos flujos necesitan latencias muy bajas, otros priorizan la tasa de datos o la fiabilidad. Para atender esta variedad, 5G extiende el concepto de QoS a nivel de flujo, identificando cada flujo de datos con un *5G QoS Identifier (5QI)*. Cada flujo puede tener un *Guaranteed Flow Bit Rate (GFBR)*, es decir, una tasa de bits mínima garantizada, que debe ser respetada.

Para garantizar esta calidad de servicio, el sistema debe asegurar que cada flujo reciba suficientes PRBs a lo largo del tiempo. Pero dado que los recursos son limitados, es necesario encontrar un equilibrio entre el uso eficiente de los PRBs y el cumplimiento de los requisitos de los distintos flujos.

### 2.2.3. Algoritmos de planificación y uso de PRBs

En esta sección se presenta un *scheduling* basado en teoría de Lyapunov, al que se le ha incorporado un agente de aprendizaje por refuerzo profundo (DRL)[8], que tiene como objetivo ajustar dinámicamente los parámetros que controlan la asignación de PRBs. Este modelo busca minimizar el uso total de recursos mientras se estabilizan las colas de tráfico y se cumplen los requisitos de tasa de bits garantizada. Este trabajo se basa en la arquitectura presentada en [8], ampliando el alcance de sus resultados.

En cada Intervalo de Tiempo de Transmisión (TTI), la capa MAC de las redes 5G ejecuta procesos de planificación para determinar qué usuarios recibirán recursos de radio y en qué cantidad. Esta asignación busca optimizar simultáneamente diversos indicadores de rendimiento, como el throughput, la latencia y la equidad en el reparto de recursos.

El algoritmo toma decisiones sobre cuántos PRBs asignar a cada flujo ($\alpha_n(t)$), teniendo en cuenta el estado de cada cola de tráfico ($Q_n(t)$), el rendimiento conseguido en el instante actual ($\rho_n(t)$) y el déficit respecto al GFBR. Para ello, se definen colas virtuales que reflejan este déficit y el objetivo es mantenerlas lo más estables posible, manteniendo el déficit lo más bajo posible. La asignación se formula como un problema de optimización, cuyo objetivo es minimizar una función que combina:

- La cantidad total de PRBs usados.
- La desviación de las colas de tráfico respecto a su estabilidad.
- El déficit acumulado respecto al GFBR.

Se asume que el tiempo está ranurado y que el número total de bloques de recursos asignados a todos los usuarios no puede superar la cantidad disponible en el sistema. Esto significa que el planificador debe tomar decisiones teniendo en cuenta los límites físicos de la red. Así, se asegura que la distribución de recursos sea realista y sostenible. Este principio es esencial para que el sistema funcione de manera eficiente, ya que evita que se sobreasignen recursos y se comprometa la estabilidad de la red.

### 2.2.4. Influencia de los parámetros de configuración

El sistema introduce tres parámetros clave para ajustar el comportamiento del algoritmo ($V$, $\omega_Q$ y $\omega_G$) que se definen a continuación:

- $V$ representa el peso que se le da a la eficiencia en el uso de PRBs.
- $\omega_Q$ controla la prioridad de mantener estables las colas reales (Radio Link Control (RLC)).
- $\omega_G$ ajusta el enfoque hacia el cumplimiento del GFBR (a través de colas virtuales).

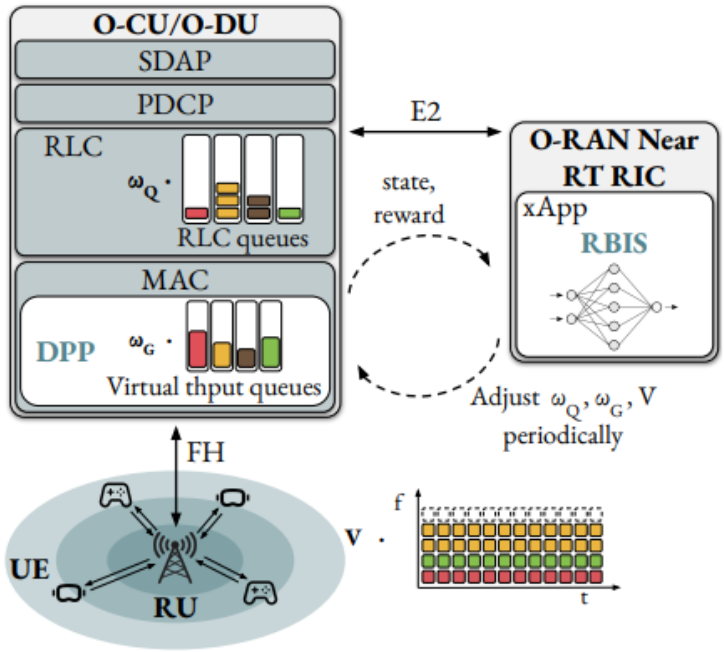



Figura 2.5: Comportamiento MAC Scheduler [8]

Por ejemplo, si el tráfico es muy alto y las colas se llenan rápidamente, puede ser necesario subir $\omega_Q$ para priorizar la evacuación de datos. En cambio, si el sistema tiene capacidad suficiente, se puede bajar $\omega_Q$ y aumentar $V$ para ahorrar recursos sin comprometer el rendimiento. Como observamos en la Figura 2.5, el ajuste dinámico de estos parámetros permite que el sistema responda de forma autónoma a los cambios en el canal o a la demanda de tráfico, manteniendo un uso eficiente de los PRBs.

### 2.2.5. Resultados previos sobre el uso de PRBs

En [8] se aplicó el esquema de DRL sobre el simulador 5G-LENA , demostrando que el sistema propuesto es capaz de reducir el uso de PRBs sin perjudicar el rendimiento. Mientras que una selección aleatoria de los parámetros puede causar un consumo excesivo de PRBs o la falta de asignación, el enfoque basado en DRL logra mantener el rendimiento deseado y al mismo tiempo reducir el número de recursos usados.

Los resultados muestran que el uso de PRBs se estabiliza y se adapta a las condiciones del canal. En particular, en situaciones con canal de calidad baja, el sistema logra mantener el rendimiento de forma más estable aumentando la prioridad del parámetro $\omega_G$, lo que evita picos de asignación innecesarios.

En resumen, los PRBs son el recurso más valioso en una red 5G, y su gestión eficiente es fundamental para garantizar un servicio de calidad a múltiples usuarios y aplicaciones. La asignación dinámica e inteligente de estos recursos, considerando el estado del canal, el tipo de tráfico y los requisitos de QoS, es clave para sacar el máximo rendimiento a la red.

Por ello en este trabajo se extiende el análisis presentado en [8] para optimizar esta asignación de forma autónoma y adaptativa, reduciendo el consumo de recursos sin sacrificar el rendimiento. Como se ha comentado anteriormente, este tipo de enfoques sienta las bases para la evolución hacia redes 6G aún más inteligentes y eficientes.

## 2.3. Conceptos básicos de IA

En esta sección se contextualizan las técnicas de IA y se justifica la elección del DRL como enfoque óptimo para el problema de MAC scheduling.

### 2.3.1. Inteligencia Artificial y sus tipos

La IA es un campo de las ciencias de computación que busca desarrollar sistemas capaces de realizar tareas que normalmente requieren inteligencia humana, como el reconocimiento de patrones, la toma de decisiones y la resolución de problemas complejos. Dependiendo de su enfoque y nivel de autonomía, la IA se puede clasificar en varias categorías principales:

- **IA basada en reglas (o simbólica):** Los sistemas siguen reglas predefinidas para tomar decisiones. Su ventaja es la explicabilidad, pero su capacidad de adaptación es limitada frente a entornos dinámicos.
- **Aprendizaje supervisado:** Los sistemas aprenden a partir de ejemplos etiquetados, buscando generalizar patrones y realizar predicciones sobre datos no vistos. Es ampliamente usado en clasificación, regresión y visión por computadora.
- **Aprendizaje no supervisado:** Los algoritmos buscan estructuras ocultas en datos sin etiquetas, como agrupamientos (*clustering*) o reducción de dimensionalidad.
- **Aprendizaje por refuerzo (RL):** Los agentes aprenden a tomar decisiones secuenciales mediante prueba y error, optimizando una señal de recompensa a lo largo del tiempo. Esta categoría incluye tanto técnicas clásicas como aquellas que incorporan redes neuronales profundas, dando lugar al *DRL*.

Dentro de este marco, el **Deep Reinforcement Learning** se sitúa en la intersección del aprendizaje profundo y el aprendizaje por refuerzo. Mientras que el aprendizaje supervisado o no supervisado se centra en encontrar patrones en los datos, el DRL se orienta a aprender políticas de acciones óptimas en entornos dinámicos, especialmente cuando el espacio de estados y acciones es grande y/o su relación es compleja. Esto lo convierte en una opción natural para problemas de control, planificación y optimización de recursos en tiempo real, como es el caso del MAC scheduling en redes 5G y 6G.

### 2.3.2. DRL en profundidad

El aprendizaje profundo por refuerzo, DRL, es una rama emergente del aprendizaje automático que combina el Deep Learning (DL) con la capacidad de toma de decisiones secuenciales del Reinforcement Learning (RL). Esta integración permite que los agentes aprendan comportamientos óptimos mediante la interacción directa con su entorno, alcanzando logros destacados en áreas como juegos, robótica, visión por computadora y procesamiento de lenguaje natural [9].

Históricamente, el aprendizaje por refuerzo se ha fundamentado en disciplinas como la psicología, la teoría del control y la estadística. Su principio central es que un agente aprende mediante prueba y error, tomando acciones en un entorno, observando los resultados y ajustando su comportamiento para maximizar una señal de recompensa acumulada. En términos formales, este proceso se modela como un Proceso de Decisión de Markov (MDP), que incluye cuatro componentes esenciales: Estados ($s$), Acciones ($a$), Función de Transición ($T$) y Función de Recompensa ($R$). El entorno se considera markoviano si el siguiente estado y la recompensa dependen únicamente del estado y la acción actuales.

Cuando se combina con aprendizaje profundo, el DRL permite representar políticas y funciones de valor complejas mediante redes neuronales. Este enfoque *end-to-end* permite que el agente aprenda directamente a partir de datos de entorno sin necesidad de diseñar manualmente características intermedias. Un ejemplo paradigmático de su éxito es *AlphaGo* y posteriormente *AlphaGo Zero*, desarrollado por DeepMind, donde se utilizaron estas soluciones para alcanzar niveles superiores a los obtenidos por jugadores humanos en el juego del *go* [9].

En el ciclo de aprendizaje del DRL, el agente observa el estado actual del entorno y selecciona una acción basada en una política $\pi(a|s)$. El entorno responde con una nueva observación del estado y una recompensa. Este ciclo lo podemos observar en la Figura 2.6, donde a través de múltiples iteraciones, el agente ajusta su política para maximizar la suma total de recompensas a lo largo del tiempo.

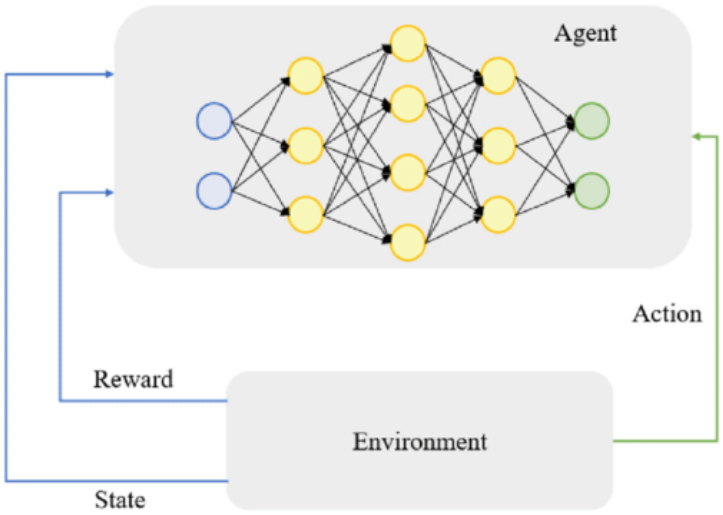



Figura 2.6: Ciclo Agent-Enviroment

En este proceso existen diferentes enfoques, como los métodos basados en valor (por ejemplo, Deep Q-Network (DQN)), los que lo hacen en política (por ejemplo, Proximal Policy Optimization (PPO)) y los métodos actor-crítico, que combinan ambos. También existen enfoques con y sin modelo, según si el agente aprende una representación explícita de la dinámica del entorno.

En resumen, el DRL representa una herramienta prometedora para el desarrollo de inteligencia artificial general, ya que permite a los agentes aprender directamente de datos del entorno, tomando decisiones autónomas complejas. Su desarrollo sigue siendo un campo activo de investigación con el potencial de revolucionar múltiples industrias.

El **MAC scheduling** en redes 5G consiste en decidir, en cada TTI, qué usuarios recibirán recursos de radio y en qué cantidad, buscando optimizar simultáneamente múltiples indicadores de rendimiento, como el *throughput*, la equidad (*fairness*) y la latencia. Esta tarea presenta tres características clave:

- **Alta complejidad y dinamismo**: la carga de tráfico, las condiciones del canal y las prioridades de los usuarios cambian de forma continua.
- **Objetivos conflictivos**: maximizar *throughput* puede perjudicar la equidad y garantizar baja latencia puede reducir eficiencia espectral.
- **Espacio de decisión muy grande**: en escenarios con un número elevado de usuarios en cada TTI existen combinaciones posibles muy numerosas de asignación de recursos.

El procedimiento que seguirá nuestro modelo DRL para gestionar los recursos de la red, tendrá un ciclo como el que se puede ver en la Figura 2.7. El Agente DRL observa el estado de red dado por el estado del buffer y el QoS de cada usuario y genera una acción basada en los parámetros explicados en 2.2.4, utilizados para la asignación de los PRBs en los TTIs correspondientes. Esta acción genera un cambio de estado, y llega al entorno 5G, el cual recompensa o penaliza al agente según qué tan buena haya sido la acción realizada. Seguidamente, el agente actualiza su política según la recompensa obtenida para volver a observar de nuevo el estado de red y continuar con el ciclo el número especificado de pasos.

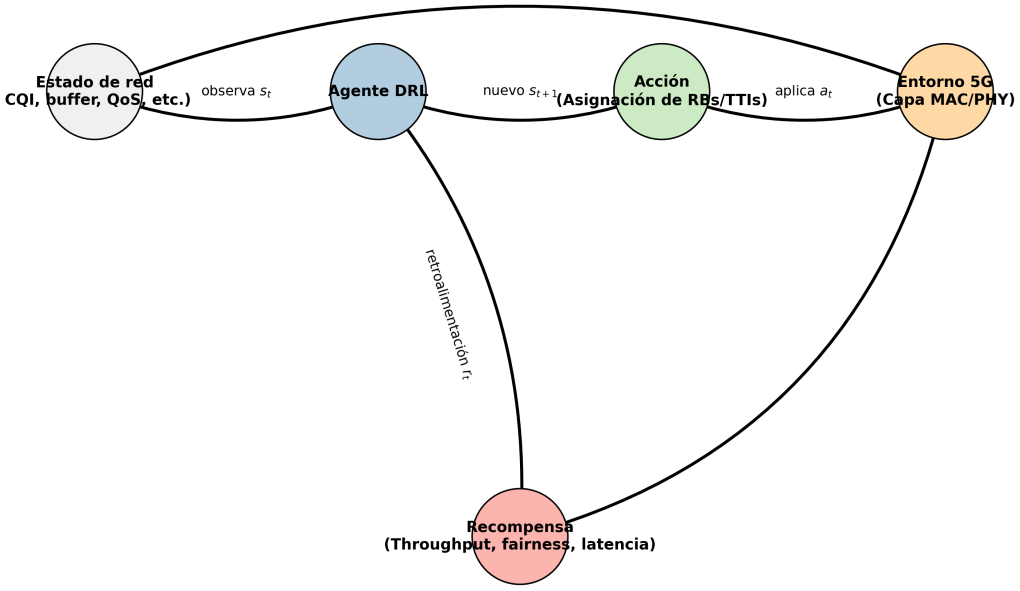



Figura 2.7: Ciclo DRL

*Capítulo 3*

# Implementación

## 3.1. Introducción

En el Capítulo 2 se han descrito los fundamentos de las redes móviles de quinta generación (5G) y los retos asociados a la gestión eficiente de sus recursos. Como se ha mencionado, uno de los mecanismos más relevantes es el MAC *scheduling*, encargado de decidir cómo se asignan los recursos radio a los distintos usuarios conectados a la red. Este problema, aunque pueda parecer sencillo en apariencia, resulta especialmente complejo debido a la gran variabilidad de las condiciones del canal, la heterogeneidad de los usuarios y la necesidad de tomar decisiones en tiempo real.

En este capítulo se aborda la implementación de la solución propuesta para este problema, basada en el uso de técnicas de DRL (2.3). El objetivo es mostrar cómo se ha llevado a la práctica la idea de emplear algoritmos de aprendizaje por refuerzo profundo para aprender políticas de asignación de recursos que superen las limitaciones de los métodos tradicionales.

Para ello, se presenta en primer lugar una descripción detallada del problema del MAC *scheduling* y de las razones que justifican la elección del DRL como alternativa viable. A continuación, se presenta el estudio realizado, incluyendo la construcción de los entornos de simulación, la definición de los estados, las acciones y las recompensas, así como la metodología seguida durante el entrenamiento de los agentes. Finalmente, se expone el enfoque adoptado para la evaluación de la implementación, estableciendo las métricas y criterios que permitirán analizar su rendimiento en comparación con otras estrategias.

Con este planteamiento se busca no solo poner a prueba la validez del uso de DRL en el contexto del MAC *scheduling*, sino también extraer conclusiones que puedan servir de base para futuros trabajos y mejoras en la optimización de recursos en redes 5G y 6G.

## 3.2. Descripción del Problema

### 3.2.1. El problema del MAC *scheduling* en redes 5G

En una red 5G, la capa MAC es la encargada de gestionar cómo se asignan los recursos radioeléctricos, en particular los bloques de recursos físicos (PRBs), entre los distintos usuarios conectados a una estación base. Este proceso, conocido como MAC *scheduling*, es uno de los componentes clave en el rendimiento

global de la red, ya que influye de manera directa tanto en la calidad de servicio percibida por los usuarios como en la eficiencia espectral del sistema [10, 11].

El *scheduler* o planificador debe tomar decisiones de asignación en un entorno extremadamente dinámico. Factores como la movilidad de los usuarios, la variabilidad del canal radio, la interferencia entre celdas y la naturaleza fluctuante del tráfico complican notablemente esta tarea. A ello se suma la heterogeneidad de los requisitos de los dispositivos: algunos priorizan un elevado *throughput*, mientras que otros demandan latencia ultra-baja o fiabilidad extrema. Como resultado, el MAC *scheduling* en 5G se plantea como un problema de optimización multi-objetivo, en el que deben equilibrarse criterios contrapuestos como la eficiencia en el uso de los recursos, la equidad en la asignación y el cumplimiento de los requisitos de calidad de servicio.

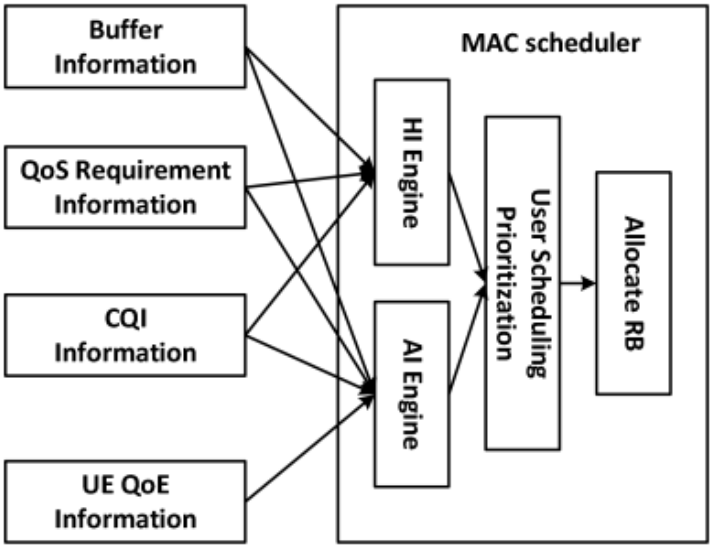



Figura 3.1: Funcionamiento del MAC *scheduling* [12]

### 3.2.2. Limitaciones de los enfoques tradicionales

A lo largo de las últimas décadas, se han propuesto distintos algoritmos para abordar el problema del MAC *scheduling*. Entre los más conocidos se encuentran *Round Robin*, que reparte los recursos de manera equitativa entre todos los usuarios sin considerar el estado del canal o *Proportional Fair*, que busca un compromiso entre equidad y rendimiento aprovechando parcialmente la información del canal [13]. También se han desarrollado variantes más avanzadas que incluyen métricas adicionales, como el historial de asignaciones o el estado de los *buffers*.

Estos métodos han demostrado ser útiles en generaciones previas de redes móviles, especialmente en LTE, donde ofrecieron un equilibrio aceptable entre complejidad y eficiencia. Sin embargo, presentan limitaciones claras en el contexto de 5G y futuras generaciones. Su principal debilidad radica en que se basan en reglas fijas, lo que les impide adaptarse de manera efectiva a escenarios dinámicos y heterogéneos. Cuando la red debe gestionar simultáneamente aplicaciones con requisitos muy diversos, como comunicaciones de realidad aumentada, transmisión masiva de vídeo en alta definición o conexiones de dispositivos del Internet de las Cosas (IoT), los algoritmos clásicos tienden a mostrar un peor comportamiento, degradando la eficiencia y, en algunos casos, la experiencia de usuario.

Diversos estudios recientes apuntan a la necesidad de introducir técnicas más flexibles y adaptativas que sean capaces de aprender patrones y anticipar la evolución del tráfico. En este sentido, enfoques basados en inteligencia artificial y, en particular, en DRL, se han planteado como una alternativa prometedora para la gestión de recursos en redes 5G y futuras generaciones [14, 15]. Estos enfoques permiten superar la rigidez de los algoritmos tradicionales y ofrecen la posibilidad de optimizar el rendimiento global de la red en entornos altamente cambiantes.

## 3.3. Justificación de la Solución Propuesta

### 3.3.1. Motivación para usar Inteligencia Artificial

La creciente complejidad de las redes 5G hace que los enfoques tradicionales de planificación de recursos resulten insuficientes para dar respuesta a escenarios dinámicos y heterogéneos. En este contexto, la inteligencia artificial (IA) se presenta como una herramienta capaz de abordar problemas de alta dimensionalidad y de aprender patrones directamente a partir de los datos o de la interacción con el entorno.

En particular, las técnicas de *machine learning* han demostrado ser útiles para tareas de optimización en comunicaciones inalámbricas, ya que permiten adaptar el comportamiento de los sistemas a condiciones cambiantes sin depender de reglas fijas. El aprendizaje automático puede mejorar la conectividad y seguridad en entornos altamente dinámicos como los vehículos conectados [16]. Esto mismo se traslada al caso del MAC Scheduling en 5G, donde las condiciones de canal y los requisitos de los usuarios varían de manera continua.

### 3.3.2. Ventajas del *Deep Reinforcement Learning* (DRL)

Dentro de las técnicas de IA, el aprendizaje por refuerzo profundo (DRL) destaca como una de las más prometedoras para problemas de asignación de recursos en redes móviles. A diferencia del RL clásico, que se limita a espacios de estado y acción reducidos, el DRL combina RL con redes neuronales profundas, lo que le permite manejar entornos complejos y de gran dimensionalidad.

En este contexto, el DRL se posiciona como la técnica de IA más adecuada para el MAC *scheduling* por varias razones:

- **Aprendizaje por interacción sin necesidad de modelos exactos:** El DRL aprende directamente de la experiencia, interactuando con el entorno y recibiendo realimentación mediante una función de recompensa. Esto elimina la necesidad de contar con un modelo matemático exacto de la red o de diseñar reglas fijas, lo que lo hace especialmente aplicable en escenarios reales con gran incertidumbre.

- **Optimización multiobjetivo y a largo plazo:** Gracias a la definición de la función de recompensa, el DRL puede integrar múltiples Key Performance Indicator (KPI) (como *throughput*, equidad o latencia) y aprender a balancearlos de manera eficiente. De este modo, el agente no solo toma decisiones inmediatas, sino que busca maximizar el rendimiento global del sistema a lo largo del tiempo.

- **Adaptabilidad ante cambios del entorno:** A diferencia de los algoritmos clásicos (p. ej., Round Robin o Proportional Fair), que siguen reglas estáticas, el DRL ajusta sus decisiones en función del estado actual de la red. Esto le permite adaptarse a un entorno cambiante, debido a variaciones en el tráfico o condiciones de canal, sin necesidad de llevar a cabo una reprogramación manual.

- **Generalización y escalabilidad:** Un agente entrenado en entornos simulados con diversas configuraciones puede aplicar lo aprendido a escenarios desconocidos, lo que evita rediseñar el *scheduler* para cada nueva configuración de red. Esta capacidad de generalización resulta clave en redes tan dinámicas como 5G y, aún más, en las futuras 6G.

- **Separación entre entrenamiento y despliegue seguro:** El proceso de aprendizaje puede realizarse en simuladores que emulen el comportamiento de la red sin afectar a usuarios reales. Una vez entrenado, el modelo puede desplegarse de manera controlada en la red en producción, garantizando seguridad y robustez.

- **Resultados empíricamente superiores:** Estudios recientes demuestran que el DRL supera a los algoritmos tradicionales en métricas como *throughput*, retardo y *fairness* [17], lo que valida su capacidad para tomar decisiones más eficientes en entornos complejos y dinámicos.

### 3.3.3. Selección de Algoritmos (PPO y DQN)

Dentro del abanico de algoritmos de DRL, en este trabajo se han seleccionado dos soluciones concretas: DQN y PPO.

Por un lado, DQN constituye uno de los algoritmos más representativos del aprendizaje por refuerzo profundo. Su principal aportación fue demostrar que una red neuronal puede aproximar la función de valor, permitiendo a un agente aprender políticas en problemas con espacios de acción discretos [18]. Esta característica lo hace especialmente adecuado para el MAC scheduling, donde la asignación de recursos puede modelarse como un conjunto de decisiones [17].

Por otro lado, PPO se sitúa dentro de la familia de los métodos de *policy gradient* y ha destacado en la literatura por su robustez y estabilidad durante el entrenamiento [19]. A diferencia de DQN, que aprende una función de valor, PPO optimiza directamente la política, lo que facilita la exploración en entornos con alta dimensionalidad y dinámicas cambiantes, como los que caracterizan a las redes móviles de nueva generación.

La comparación entre DQN y PPO permite analizar dos perspectivas distintas dentro del DRL: mientras que DQN se centra en la estimación de valores de acción en entornos discretos, PPO plantea una aproximación más flexible basada en la optimización directa de políticas. Considerar ambos algoritmos en paralelo resulta útil para evaluar no solo la viabilidad del DRL en el MAC scheduling, sino también las ventajas y limitaciones que cada técnica puede mostrar en escenarios de red de próxima generación.

## 3.4. Diseño de la solución de DRL

### 3.4.1. Enfoque inicial y prueba de concepto

El estudio comenzó con un enfoque exploratorio cuyo objetivo era comprobar si un agente de aprendizaje por refuerzo profundo (DRL) podía aprender a ajustar dinámicamente los parámetros del scheduler Lyapunov, introducido en el Capítulo 2, en la capa MAC de una red 5G. Estos parámetros —$v$, $w_Q$, $w_G$— regulan el compromiso entre eficiencia espectral, estabilidad de colas y cumplimiento del GFBR, y su ajuste óptimo puede mejorar significativamente el rendimiento del sistema.

Para ello, se diseñó un primer entorno denominado `MACEnv`, junto con un agente basado en PPO y DQN, realizado con Gymnasium [20], una librería en Python que proporciona entornos simulados estandarizados para entrenar agentes de aprendizaje por refuerzo. Este entorno se validó sobre un dataset tabulado generado mediante simulaciones, en el que cada fila representaba una combinación de estado, acción y recompensa. Como aún no se disponía del siguiente estado proporcionado por el simulador, se implementó una lógica interna en el entorno para simular la evolución del sistema tras aplicar una acción. Esta lógica modificaba los valores de Modulation and Coding Scheme (MCS) y *buffers* de forma probabilística, en función de la acción tomada, con el objetivo de introducir dinamismo y evitar que el agente memorizara el dataset.

Este primer entorno y agente se concibieron como una prueba de concepto, para validar que el enfoque DRL era viable y que el agente podía aprender a seleccionar combinaciones de parámetros que maximizasen la recompensa.

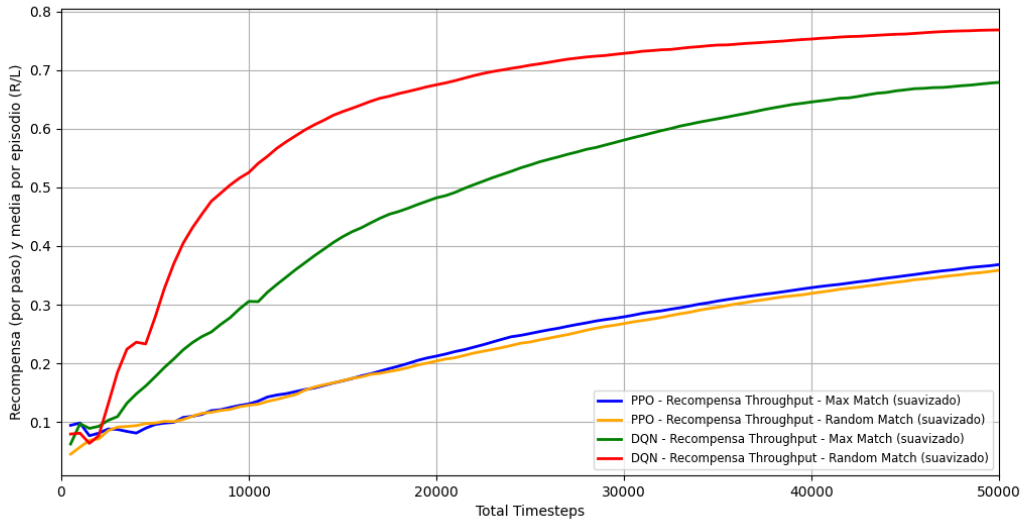



Figura 3.2: Evolución de la Recompensa por Throughput por Timestep en el entorno `MACEnv` con el dataset tabulado.

En las Figuras 3.2 y 3.3, se muestra cómo con esta lógica interna del estado, se obtenían resultados positivos, al comparar la evolución de la recompensa a lo largo del entrenamiento de 50 000 timesteps, para cada uno de los escenarios propuestos, sobre cada una de las columnas de recompensas. Se obtiene así una primera aproximación de cómo se comportarían los algoritmos propuestos y cuál de ellos aprendería más rápido y mejor, identificando además la estrategia que daría recompensas más altas.

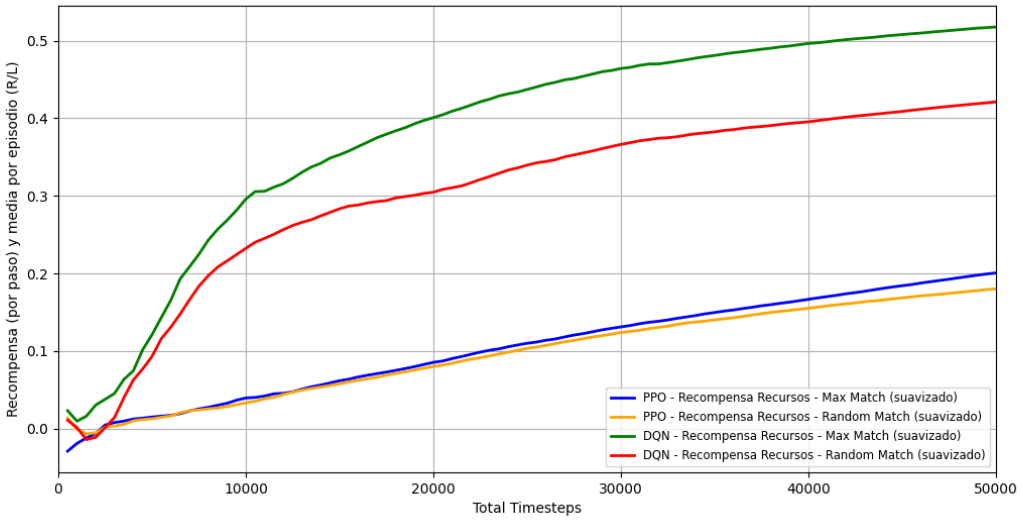



Figura 3.3: Evolución de la Recompensa por Recursos por Timestep en el entorno `MACEnv` con el dataset tabulado.

### 3.4.2. Definición técnica del entorno inicial

En el dataset inicial existen varios elementos que definían con detalle el estado de la red. Estos se podrían separar en dos grupos: constantes y variables. Los elementos constantes del dataset proporcionan información necesaria para la red, pero en este estudio no serían relevantes para el agente, debido a su invariabilidad, por lo que no formaron parte del entorno.

Estos elementos son los siguientes:

- `num_ues`: número de usuarios en la simulación. Fijado a 4.
- `gfbr_1` a `gfbr_4`: tasa de bit asegurada para cada uno de los usuarios. Fijada a [30, 30, 15, 15].
- `max_prbs`: número máximo de PRBs. Fijado a 426.

El entorno `MACEnv` se estructuró con los siguientes elementos:

**Estado:** vector de 8 variables discretas:

- `mcs_1` a `mcs_4`: calidad del canal por usuario, con valores entre 0 (degradado) y 2 (óptimo).
- `avg_buffer_1` a `avg_buffer_4`: ocupación media de las colas RLC, con valores entre 0 (vacía) y 2 (saturada).

**Acción:** combinación discreta de los tres parámetros del *scheduler*:

- $v \in [0, 10]$: parámetro de configuración relacionado con la reducción de los PRBs usados. Cuanto mayor sea, más se reducen los PRBs usados por los usuarios.

- $w_Q \in [0, 3]$: parámetro de configuración relacionado con el peso de estabilizar los buffers. Cuanto mayor sea, más importancia se le da a reducir las colas de los usuarios.
- $w_G \in [0, 3]$: parámetro de configuración relacionado con el peso del GFBR. Cuanto mayor sea, más importancia se le da a cumplir con la tasa de bits requerida de los usuarios

Como se puede observar, incluso en un escenario sencillo y con codificaciones de parámetros simplificadas, el número total de acciones es $11 \times 4 \times 4 = 176$.

**Recompensa:** esta se obtiene buscando el valor de la recompensa en las filas que se corresponden al estado destino. Como puede haber varias filas coincidentes, se definieron dos estrategias de selección de fila:

- `max`: se selecciona la recompensa más alta entre las filas coincidentes.
- `random`: se elige una recompensa aleatoria entre las disponibles.

Además, en el dataset se indican dos valores de recompensa:

- `reward_thput`: recompensa basada en el throughput actual de los usuarios de la red.
- `reward_thput_resources`: recompensa considerando ambos parámetros: el throughput actual y los PRBs usados por los usuarios de la red.

**Transición:** como el dataset no incluía el siguiente estado, se implementó una lógica probabilística que modificaba los MCS y buffers en función de la acción tomada, simulando la evolución del sistema.

Esta lógica de transición de estado se basa en la actualización de los elementos que lo componen para cada uno de los usuarios (`i`): `mcs_i` y `avg_buffer_i`.

La lógica para `mcs` está basada en `v` (reducción de PRBs) y `wg` (prioridad GFBR). Cuando `v` = 6-10, y `mcs` no es bajo, `new_mcs` disminuye 1; si no, cuando `wg` = 2 o 3, y un usuario con GFBR = 30, y `mcs` no es alto, `new_mcs` aumenta 1. Si no cumple ninguna de las dos, hay una probabilidad del 20 % de que `new_mcs` aumente o disminuya 1.

La lógica para `avg_buffer` está basada en `wq` (prioridad a colas) y `v` (reducción de PRBs). Cuando `wq` = 2 o 3, y `avg_buffer` es alto, `new_avg_buffer` disminuye 1; si no, cuando `v` = 6-10, y `avg_buffer` no es alto, `new_avg_buffer` aumenta 1. Si no cumple ninguna de las dos, hay una probabilidad del 20 % de que `new_avg_buffer` aumente o disminuya 1.

Esta lógica interna de transición de estado busca reflejar el comportamiento que utilizaría el simulador ns-3, concretamente con el módulo 5G LENA.

### 3.4.3. Limitaciones observadas y mejora del entorno

Durante el entrenamiento con `MACEnv`, se detectaron dos limitaciones clave:

- **Combinaciones inexistentes:** el agente caía con frecuencia en combinaciones estado–acción no representadas en el dataset. Se penalizó con una recompensa negativa y se reiniciaba el estado aleatoriamente.

- **Duplicados:** múltiples filas con el mismo estado y acción pero distinta recompensa. Se gestionaron mediante las estrategias `max` y `random` previamente comentadas.

Aunque el agente mostraba señales de aprendizaje, la falta de transiciones reales limitaba su capacidad de generalización. Por ello, se generó un segundo dataset, esta vez incluyendo el `next_state` definido por el simulador como resultado de aplicar una acción en un estado dado. Este cambio permitió construir un entorno más realista, denominado `MACSchedulerEnv`, basado en un MDP completo.

En este caso, el agente DRL interactúa con el entorno siguiendo el ciclo clásico de aprendizaje por refuerzo:

1. Observa el estado actual: MCS + buffers.
2. Selecciona una acción: combinación de $v$, $w_Q$, $w_G$.
3. El entorno consulta el dataset:
   - Si existe la combinación: se aplica la recompensa y se transita al siguiente estado.
   - Si no existe: se penaliza y se reinicia el estado aleatoriamente.
4. El agente actualiza su política (PPO) o su red de valor (DQN) en función de la recompensa obtenida.

Este proceso se repite durante miles de timesteps, permitiendo al agente aprender qué combinaciones de parámetros son más efectivas en cada contexto. Para evaluar el rendimiento del agente, se definieron ocho escenarios de entrenamiento, combinando algoritmo, estrategia de selección y tipo de recompensa, tal como se indica a continuación:

Tabla 3.1: Combinaciones de algoritmo, estrategia de asignación y función de recompensa utilizadas.

| Algoritmo | Estrategia | Función de recompensa |
|---|---|---|
| PPO | Max | `reward_thput` |
| PPO | Random | `reward_thput` |
| PPO | Max | `reward_thput_resources` |
| PPO | Random | `reward_thput_resources` |
| DQN | Max | `reward_thput` |
| DQN | Random | `reward_thput` |
| DQN | Max | `reward_thput_resources` |
| DQN | Random | `reward_thput_resources` |

Cada modelo se entrenó durante 100 000 timesteps. Se registraron las recompensas por paso y para su representación se aplicó suavizado con una media móvil, y se generaron gráficas comparativas. También se contabilizó el número de penalizaciones por no encontrar estado destino (“no match”, lo que permitió evaluar la cobertura efectiva del dataset.

Para comprobar que los agentes habían aprendido políticas útiles, se desarrolló un script de validación que evaluaba su comportamiento en estados críticos. El agente tomaba decisiones en función del estado observado, sin actualizar su política, y se analizaban las acciones seleccionadas y las recompensas obtenidas.

Los resultados mostraron que los modelos entrenados con el dataset modificado y con la representación simplificada del estado presentaban una mayor capacidad de adaptación. En particular, se observó una reducción significativa en la tasa de penalización y una mayor coherencia entre las decisiones del agente y las que el simulador consideraba óptimas.

### 3.4.4. Optimización de hiperparámetros con Optuna

Para mejorar el rendimiento de los modelos PPO y DQN, se integró un estudio de optimización de hiperparámetros utilizando la librería `Optuna` [21]. Esta herramienta permite realizar búsquedas automáticas y eficientes en espacios de hiperparámetros mediante técnicas como la optimización bayesiana, el muestreo aleatorio o el método de árbol de Parzen estimado. Su principal ventaja es que reduce significativamente el tiempo y el esfuerzo necesarios para encontrar configuraciones óptimas, especialmente en entornos donde el número de combinaciones posibles es elevado y el coste computacional de cada entrenamiento es considerable.

En este estudio, Optuna se utilizó para explorar sistemáticamente distintas configuraciones de hiperparámetros, evaluando el rendimiento de cada una en función de la recompensa media obtenida tras un entrenamiento breve. Se realizaron un total de 30 ejecuciones por algoritmo, lo que permitió obtener una muestra representativa del espacio de búsqueda y seleccionar los valores más adecuados para cada modelo. Los hiperparámetros explorados fueron los siguientes:

- **Para PPO**:
  - `learning_rate`: tasa de aprendizaje del optimizador.
  - `n_steps`: número de pasos antes de actualizar la política.
  - `batch_size`: tamaño del lote de datos utilizado en cada actualización.
  - `gamma`: factor de descuento aplicado a las recompensas futuras.
  - `clip_range`: rango de recorte para la actualización de la política.
- **Para DQN**:
  - `learning_rate`: tasa de aprendizaje del optimizador.
  - `buffer_size`: tamaño del buffer de experiencia.
  - `batch_size`: tamaño del lote de datos extraído del buffer.
  - `gamma`: factor de descuento aplicado a las recompensas futuras.
  - `exploration_fraction`: proporción del entrenamiento dedicada a la exploración.

La integración de Optuna permitió ajustar con precisión los modelos sin necesidad de intervención manual, adaptándolos específicamente al entorno `MACSchedulerEnv` y al dataset utilizado. Los valores óptimos obtenidos fueron aplicados directamente en los entrenamientos finales, contribuyendo a mejorar la estabilidad del aprendizaje, la eficiencia en la selección de acciones válidas y la recompensa media obtenida en todos los escenarios evaluados.

*Capítulo 4*

# Evaluación

A lo largo de este capítulo se presentarán los principales resultados obtenidos de la evaluación de rendimiento del agente de DRL. En primer lugar, se estudiará la adecuación del dataset para facilitar el aprendizaje del agente y los cambios realizados. Posteriormente, se presentará el rendimiento del agente ante diferentes configuraciones y la mejora mediante la optimización de hiperparámetros.

## 4.1. Análisis de cobertura y representación del dataset

Antes de presentar los resultados del entrenamiento, se realiza una evaluación cuantitativa de la cobertura del dataset original utilizado en el entorno `MACSchedulerEnv`. El objetivo es determinar el grado de representación de las combinaciones estado–acción disponibles y su impacto en la interacción del agente con el entorno.

En un primer análisis se estudió la adecuación del dataset al objetivo de aprendizaje. Para ello, durante el entrenamiento en los ocho escenarios definidos (combinando algoritmo, estrategia de selección y tipo de recompensa) se registró el número total de veces que el agente seleccionó una combinación estado–acción no presente en el dataset durante los 100 000 timesteps con los que se entrena cada escenario. En estos casos, el entorno responde con una penalización y transita a un estado aleatorio.

Tabla 4.1: Número de penalizaciones por "no match" en cada escenario utilizando el dataset original.

| Algoritmo | Estrategia | Columna de Recompensa | No match | Porcentaje |
|---|---|---|---|---|
| PPO | max | `reward_thput_resources` | 14158 | 14.158 % |
| PPO | max | `reward_thput` | 20586 | 20.586 % |
| PPO | random | `reward_thput_resources` | 14703 | 14.703 % |
| PPO | random | `reward_thput` | 14594 | 14.594 % |
| DQN | max | `reward_thput_resources` | 5832 | 5.832 % |
| DQN | max | `reward_thput` | 9921 | 9.921 % |
| DQN | random | `reward_thput_resources` | 12230 | 12.230 % |
| DQN | random | `reward_thput` | 12448 | 12.448 % |

Como se observa en la Tabla 4.1, el porcentaje de veces que selecciona una combinación estado-acción no presente en el dataset es muy alto, llegando a valores de incluso más de 20 % en algún caso.

Las gráficas obtenidas muestran una evolución irregular, con interrupciones frecuentes en la curva de aprendizaje y una recompensa media inferior. Esto refleja el efecto de la baja cobertura del dataset sobre

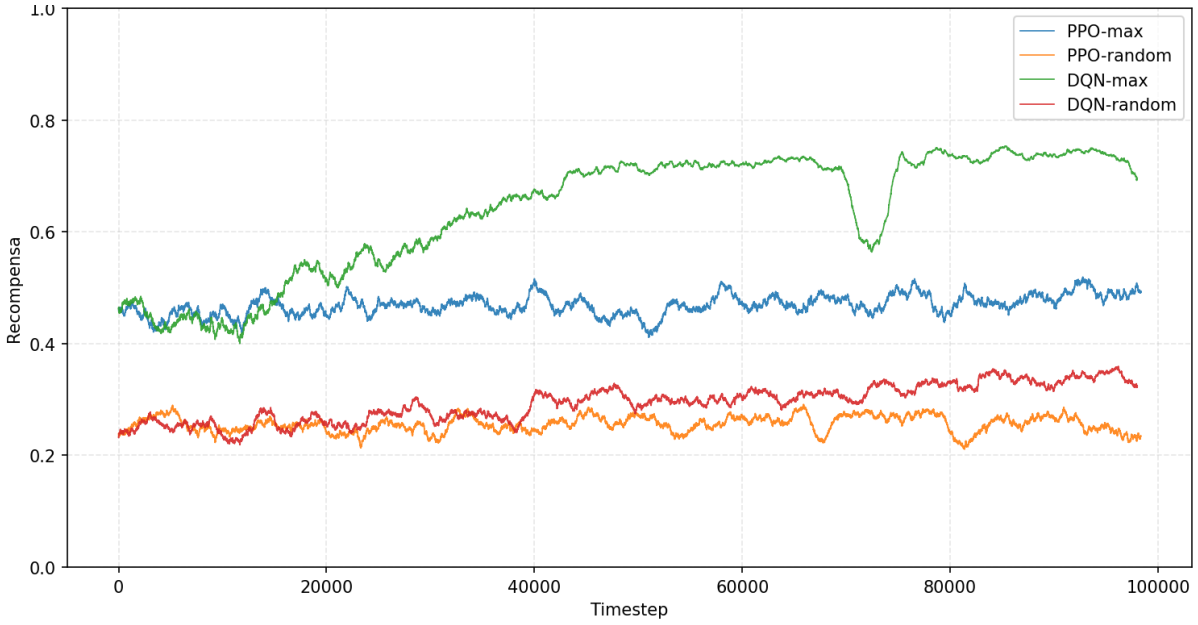



Figura 4.1: Recompensa media por timestep utilizando `reward_thput_resources` y dataset original. Media móvil con ventana de 2000.

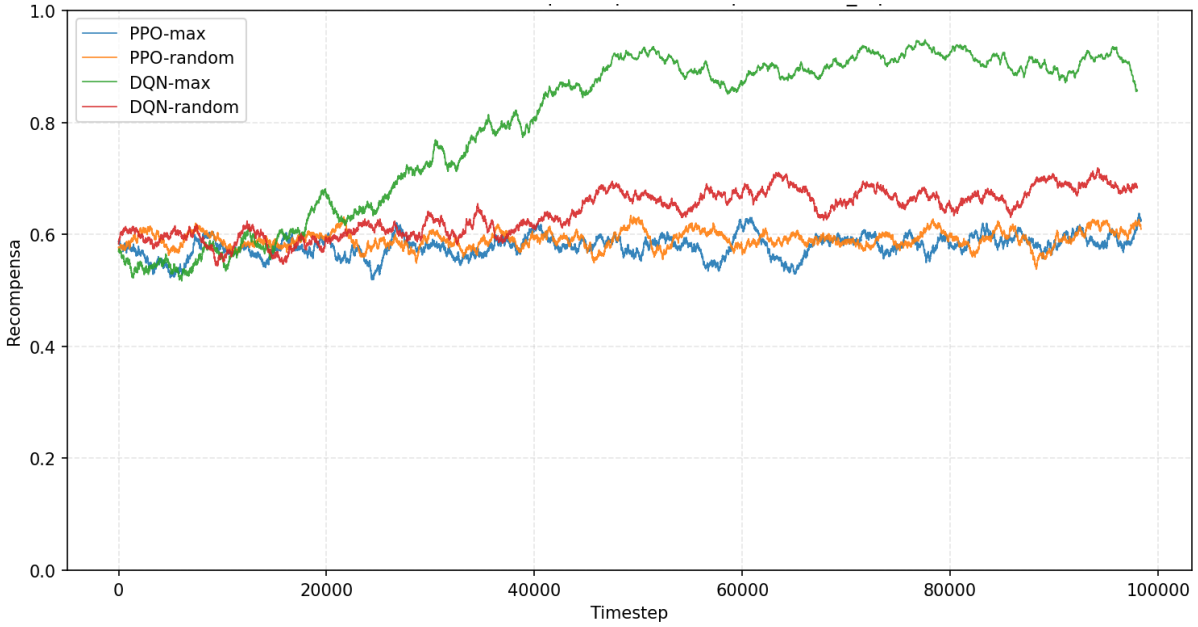



Figura 4.2: Recompensa media por timestep utilizando `reward_thput` y dataset original. Media móvil con ventana de 2000.

el rendimiento del agente.

Con el objetivo de mejorar la representación de combinaciones válidas y reducir la fragmentación del espacio de observación, se aplicaron las siguientes modificaciones:

- Agrupación de los valores altos de $v$ (8, 9 y 10) como un único valor $v = 8$, manteniendo la granularidad en los valores bajos.
- Sustitución de los MCS individuales por su suma total (`sum_mcs`) como indicador global de

calidad de canal. Se aplicó la misma transformación al siguiente estado, generando la columna `sum_next_mcs`.

Tabla 4.2: Número de penalizaciones por “no match” en cada escenario utilizando el dataset modificado.

| Algoritmo | Estrategia | Columna de Recompensa | No match | Porcentaje |
|---|---|---|---|---|
| PPO | max | `reward_thput_resources` | 5132 | 5.132 % |
| PPO | max | `reward_thput` | 8956 | 8.956 % |
| PPO | random | `reward_thput_resources` | 5369 | 5.369 % |
| PPO | random | `reward_thput` | 5339 | 5.339 % |
| DQN | max | `reward_thput_resources` | 1965 | 1.965 % |
| DQN | max | `reward_thput` | 3243 | 3.243 % |
| DQN | random | `reward_thput_resources` | 3029 | 3.029 % |
| DQN | random | `reward_thput` | 3214 | 3.214 % |

Los nuevos datos muestran unos porcentajes mucho más bajos, llegando como máximo a casi el 9 % y reduciendo otros hasta casi el 2 %.

El agente interactúa con el entorno de forma más eficiente, consolidando políticas adaptativas. Este análisis confirma que las modificaciones aplicadas al dataset mejoran la cobertura efectiva del entorno y permiten entrenar agentes con mayor estabilidad y rendimiento.

## 4.2. Recompensa media por timestep — `reward_thput_resources`

En esta sección se presentan los resultados de las gráficas obtenidas tras los cambios introducidos y explicados en la sección 4.1, tras reducir significativamente el número de penalizaciones por "no match.[en] cada escenario al utilizar el dataset modificado. Se puede observar a simple vista la mejoría en la calidad de las recompensas y la suavidad de las curvas en ambas gráficas.

La gráfica de la Figura 4.3 muestra la evolución de la recompensa media obtenida por los agentes PPO y DQN bajo las estrategias `max` y `random`. Se observa que DQN tiene un aprendizaje mucho más marcado que PPO sobre todo los primeros 50 000 steps, y que con estrategia `max`, se alcanza la recompensa más alta y sostenida. Las configuraciones con estrategia `random` presentan menor rendimiento.

La curva de DQN–max destaca por su estabilidad y convergencia clara, lo que indica que el agente ha aprendido una política eficaz y consistente. Esta métrica, que combina throughput con eficiencia en el uso de PRBs, parece favorecer a DQN frente a PPO.

## 4.3. Recompensa media por timestep — `reward_thput`

En esta segunda métrica que observamos en la gráfica de la Figura 4.4, centrada únicamente en el throughput, se mantiene la superioridad de DQN con estrategia `max`, aunque la diferencia con PPO se hace más evidente. Las curvas solo parecen mostrar una evolución con DQN, y al igual que antes, las configuraciones con `random` siguen generando aún peores resultados.

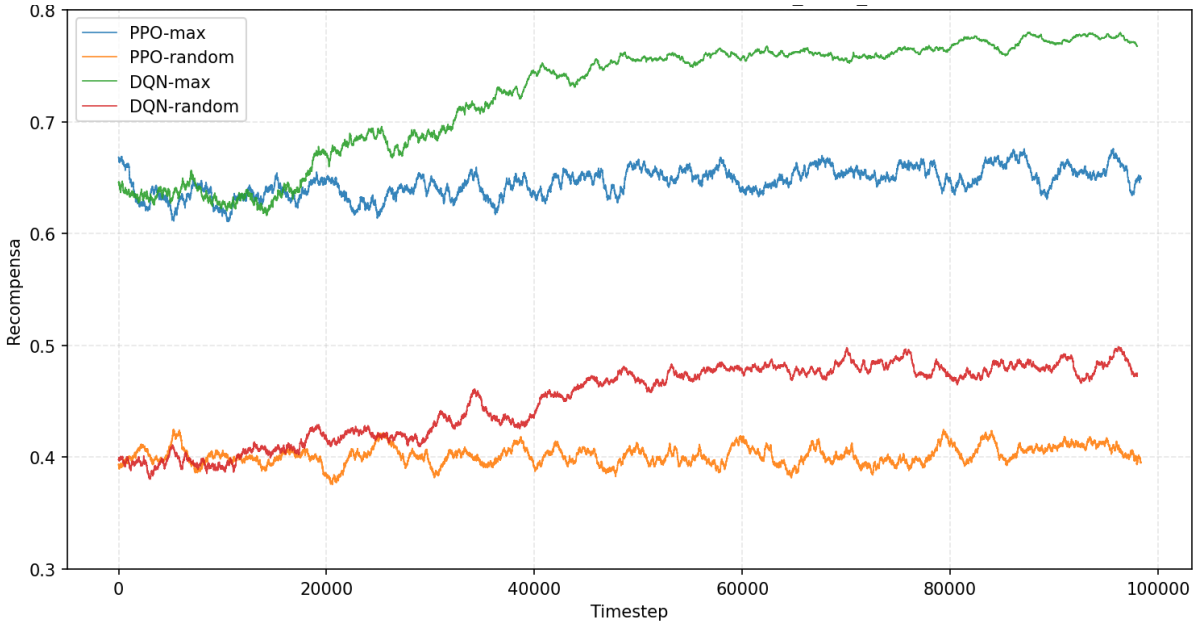



Figura 4.3: Recompensa media por timestep utilizando `reward_thput_resources`. Media móvil con ventana de 2000.

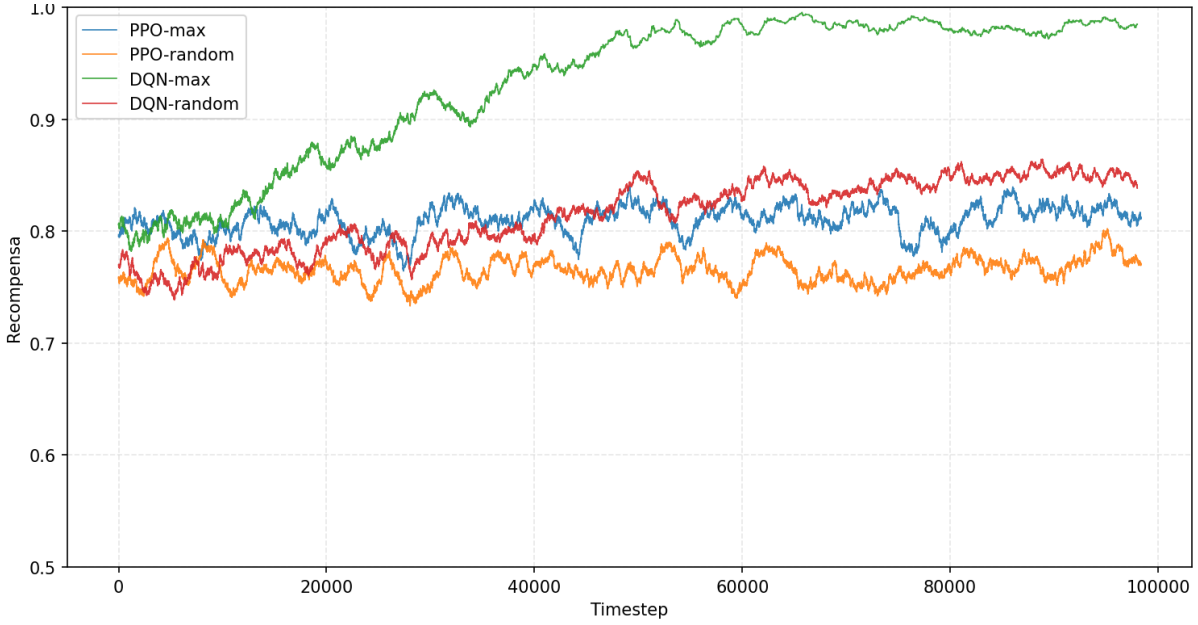



Figura 4.4: Recompensa media por timestep utilizando `reward_thput`. Media móvil con ventana de 2000.

La recompensa media obtenida es superior a la del caso anterior, lo que sugiere que la métrica `reward_thput` proporciona un objetivo más equilibrado para el agente, o que esta segunda columna de recompensas tiene valores más altos.

## 4.4. Optimización de hiperparámetros con Optuna

Con el objetivo de afinar el rendimiento de los modelos PPO y DQN, se realizó un estudio de optimización automatizada de hiperparámetros utilizando la librería `Optuna`.

A diferencia de una única optimización global, en este trabajo se ejecutó un estudio independiente para cada uno de los ocho escenarios definidos, combinando algoritmo, estrategia de selección (`max` o `random`) y tipo de recompensa. Para cada configuración, se realizaron 30 ejecuciones (*trials*), entrenando el modelo durante 20 000 pasos y evaluando su rendimiento sobre 5 000 pasos adicionales en modo determinista.

Los hiperparámetros optimizados incluyeron, entre otros: tasa de aprendizaje, tamaño de batch, factor de descuento, parámetros de exploración (en DQN), y coeficientes de regularización (en PPO). En la Tabla 4.3 se recogen los valores óptimos obtenidos para algunos escenarios representativos.

Tabla 4.3: Hiperparámetros óptimos seleccionados mediante Optuna para cuatro escenarios representativos.

| Escenario | Hiperparámetro | Valor óptimo |
|---|---|---|
| DQN–max–`reward_thput_resources` | `learning_rate` | 6.23e-4 |
| | `buffer_size` | 100000 |
| | `learning_starts` | 5000 |
| | `batch_size` | 256 |
| | `gamma` | 0.959 |
| | `target_update_interval` | 1500 |
| | `train_freq` | 4 |
| | `exploration_fraction` | 0.316 |
| | `exploration_initial_eps` | 0.802 |
| | `exploration_final_eps` | 0.011 |
| DQN–random–`reward_thput` | `learning_rate` | 2.38e-5 |
| | `buffer_size` | 100000 |
| | `learning_starts` | 3000 |
| | `batch_size` | 128 |
| | `gamma` | 0.984 |
| | `target_update_interval` | 250 |
| | `train_freq` | 1 |
| | `exploration_fraction` | 0.547 |
| | `exploration_initial_eps` | 0.854 |
| | `exploration_final_eps` | 0.016 |
| PPO–max–`reward_thput_resources` | `learning_rate` | 1.52e-5 |
| | `n_steps` | 2048 |
| | `batch_size` | 256 |
| | `gamma` | 0.998 |
| | `clip_range` | 0.179 |
| | `gae_lambda` | 0.883 |
| | `ent_coef` | 0.0024 |
| | `vf_coef` | 0.407 |
| PPO–random–`reward_thput` | `learning_rate` | 2.48e-4 |
| | `n_steps` | 2048 |
| | `batch_size` | 64 |
| | `gamma` | 0.980 |
| | `clip_range` | 0.193 |
| | `gae_lambda` | 0.984 |
| | `ent_coef` | 0.0052 |
| | `vf_coef` | 0.389 |

A continuación se muestran las gráficas de recompensa obtenidas tras aplicar los hiperparámetros optimizados y aplicado sobre 500 000 timesteps.

En comparación con las gráficas obtenidas antes de la optimización y aumento de timesteps, se observa una mejora clara en la estabilidad del aprendizaje, una mayor recompensa media sostenida y una reducción

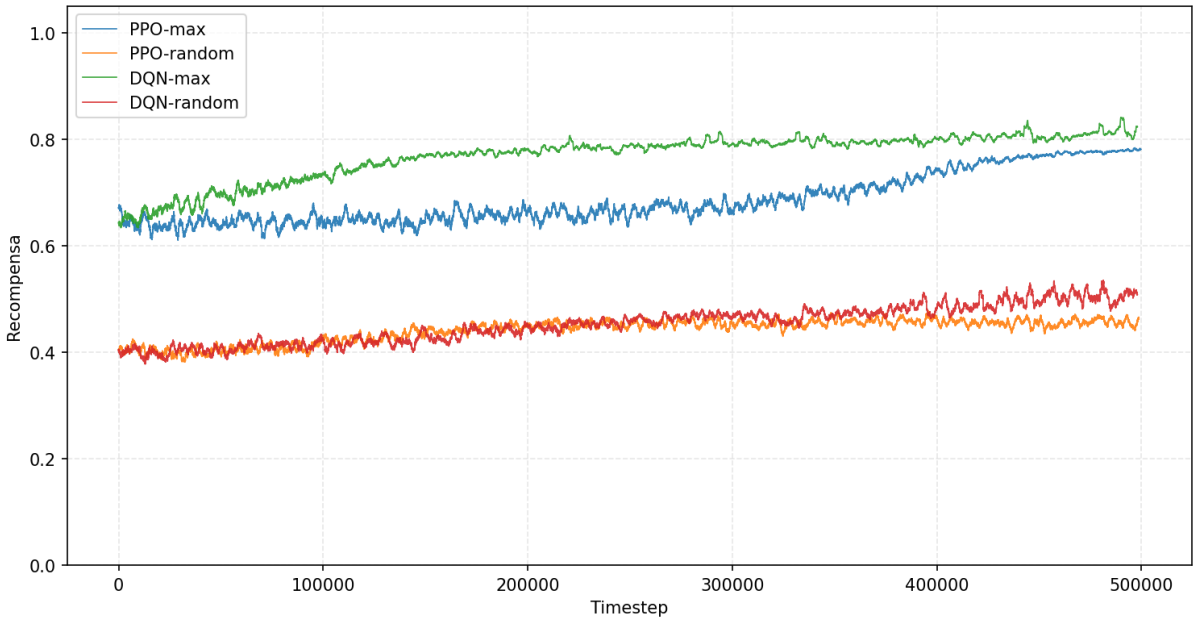



Figura 4.5: Evolución de la recompensa media por timestep tras la optimización con Optuna con recompensa recursos.

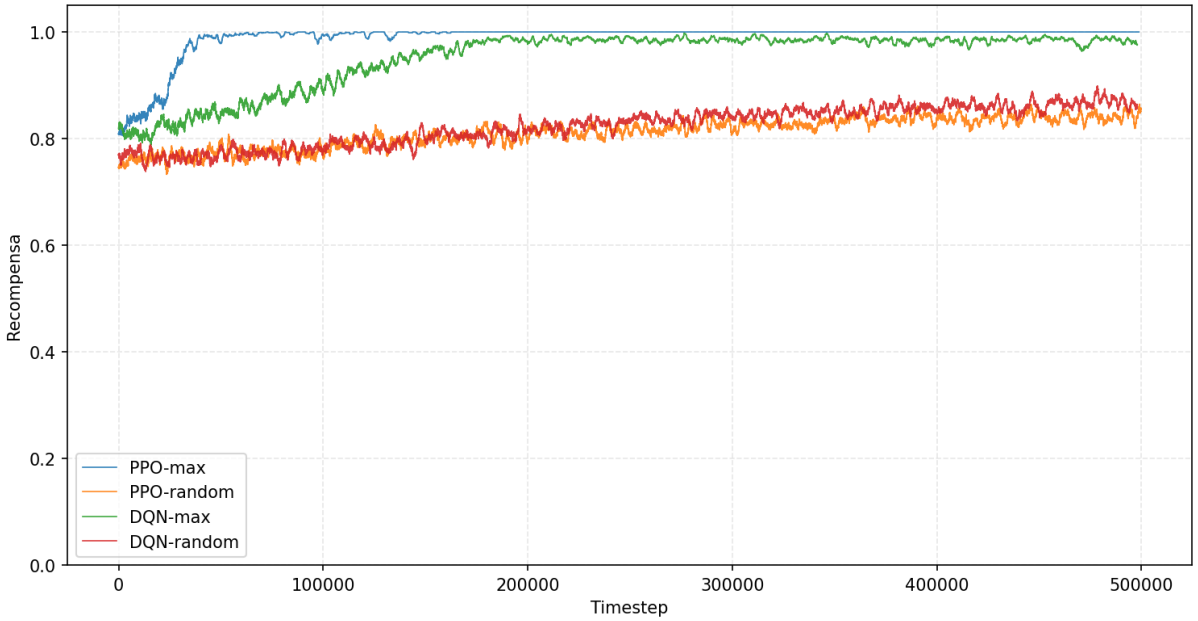



Figura 4.6: Evolución de la recompensa media por timestep tras la optimización con Optuna con recompensa throughput.

en la variabilidad entre episodios. Además, al aumentar los timesteps, descubrimos que los escenarios que utilizan PPO, finalmente parecen converger. Igualmente, los mejores resultados los siguen dando los modelos que utilizan la columna de recompensas `reward_thput`, y la estrategia `max`.

Como podemos observar en ambas gráficas (Ver Figuras 4.5 y 4.6), aunque algunas lo hagan con mayor brevedad, todas las líneas de todos los modelos ascienden. Estos resultados confirman que la optimización automatizada ha sido eficaz para ajustar los parámetros más sensibles de cada algoritmo, mejorando el rendimiento global del agente en todos los escenarios evaluados.

## 4.5. Evaluación de decisiones del agente

Una vez completado el entrenamiento, se evaluó el comportamiento del agente en modo verificación (*testing*), durante 50 000 timesteps en cada escenario, y utilizando la política aprendida sin realizar actualizaciones adicionales. El objetivo de esta fase es validar que el modelo generaliza correctamente y mantiene un rendimiento estable fuera del proceso de entrenamiento.

Las pruebas se realizaron en modo determinista, registrando la recompensa acumulada por episodio y observando las decisiones tomadas por el agente en distintos estados. A continuación se muestran las gráficas obtenidas durante el *testing* (Ver Figuras 4.7 y 4.8).

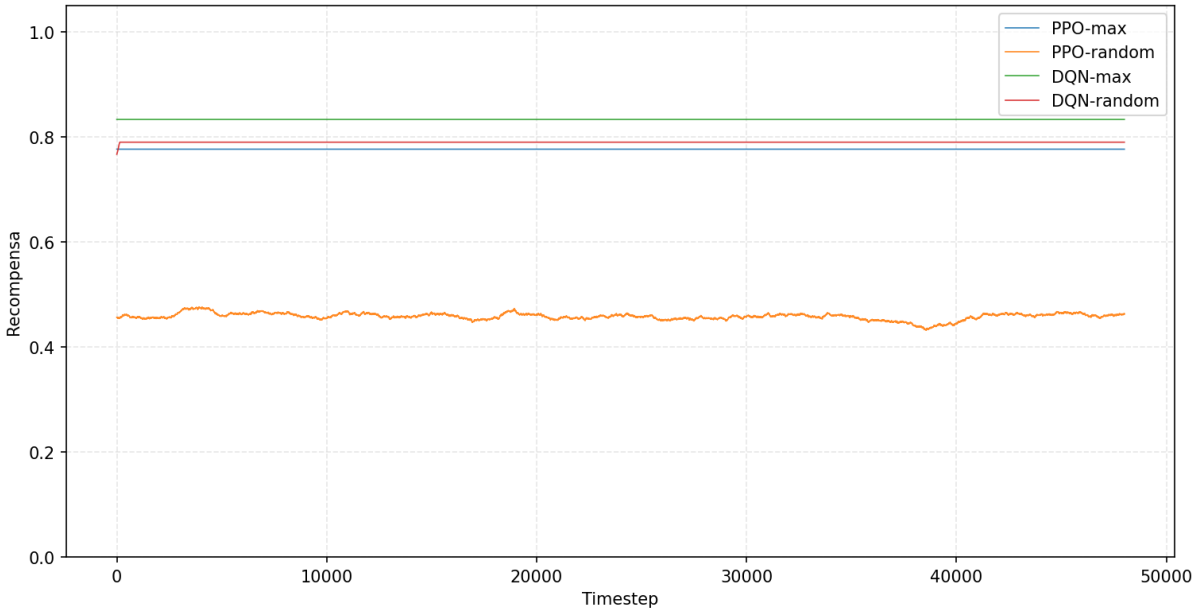



Figura 4.7: Evolución de la recompensa de recursos por timestep en fase de verificación.

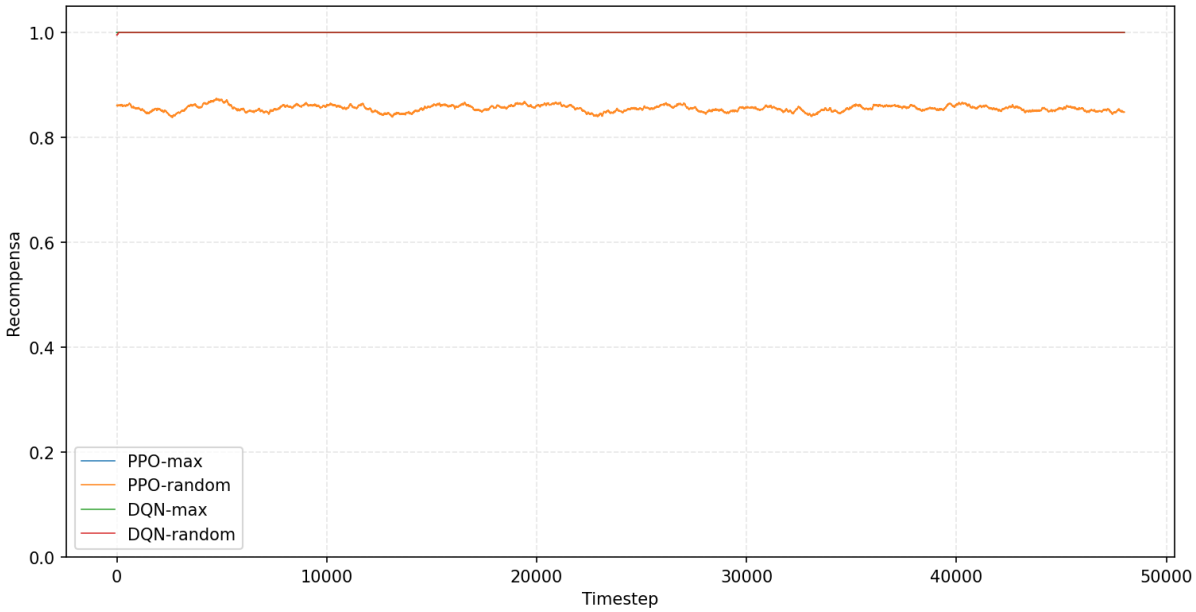



Figura 4.8: Evolución de la recompensa de throughput por timestep en fase de verificación

Los resultados muestran que ambos agentes son capaces de mantener una recompensa positiva de forma

sostenida, lo que indica que la política aprendida es válida y generaliza correctamente. En el caso de DQN, se observa una mayor consistencia en la recompensa acumulada, con menor dispersión entre episodios. PPO, por su parte, con la estrategia `random` parece que no ha terminado de aprender bien la política y tiene más picos en la gráfica, lo que nos indica que a veces falla.

En comparación con las curvas obtenidas durante el entrenamiento, las gráficas de verificación reflejan una mejora en la estabilidad del comportamiento del agente.

*Capítulo 5*

# Conclusiones

El presente trabajo ha demostrado que es posible aplicar técnicas de aprendizaje por refuerzo profundo para optimizar la asignación de recursos radio en redes móviles 5G. A través del diseño de un entorno simulado y la implementación de agentes basados en los algoritmos PPO y DQN, se ha evaluado el comportamiento de los modelos en distintos escenarios, utilizando métricas de recompensa como criterio principal.

Los resultados obtenidos muestran que los agentes son capaces de aprender políticas que mejoran el rendimiento del sistema en términos de estabilidad y eficiencia. En particular, los modelos entrenados con la estrategia de selección `max` han mostrado una evolución más estable durante el entrenamiento, mientras que los modelos con estrategia `random` exploran configuraciones más diversas, aunque con mayor variabilidad.

También se ha llegado a la conclusión de que usando recompensas basadas en tasa (`reward_thput`) se obtiene un rendimiento más alto en general, y que DQN y PPO, independientemente de la velocidad a la que aprenden, acaban convergiendo a valores de recompensas medias muy similares.

La aplicación de Optuna para la optimización de hiperparámetros ha permitido ajustar los modelos de forma precisa, obteniendo mejoras significativas en la recompensa media y en la consistencia del aprendizaje. Los resultados obtenidos confirman que los agentes generalizan correctamente y mantienen un comportamiento coherente fuera del entorno de entrenamiento.

En conjunto, el estudio ha permitido validar un enfoque basado en inteligencia artificial para la toma de decisiones en sistemas de gestión de recursos y ha proporcionado una base sólida para futuras investigaciones en este ámbito.

# Bibliografía


[1] E. Zontou. *Unveiling the Evolution of Mobile Networks: From 1G to 7G*. 2023. URL: https://arxiv.org/abs/2310.19195.

[2] A. W. Services. *Documento técnico de AWS Integración y entrega continuas para redes 5G en AWS*. 2025. URL: %7Bhttps://docs.aws.amazon.com/es_es/whitepapers/latest/cicd_for_5g_networks_on_aws/cicd_for_5g_networks_on_aws.html%7D.

[3] T. Mumtaz, S. Muhammad, M. I. Aslam e I. Ahmed. "Inter-slice resource management for 5G radio access network using markov decision process". En: 79.4 (2022), págs. 541-557. DOI: 10.1007/s11235-021-00877-9.

[4] R. W. News. *Huawei, Ericsson, and Nokia are the most active companies contributing to 5G 3GPP standardization*. Accessed: 2025-09-09. 2023. URL: https://www.rcrwireless.com/20230329/opinion/huawei-ericsson-and-nokia-are-the-most-active-companies-contributing-to-5g-3gpp-standardization-analyst-angle.

[5] T. Trainer. *Magic Quadrant for 5G Network Infrastructure: Leaders, Challengers, and Visionaries*. Accessed: 2025-09-09. 2021. URL: https://www.telecomtrainer.com/magic-quadrant-for-5g-network-infrastructure-leaders-challengers-and-visionaries/.

[6] Casadomo. *Proyecto DAEMON: Marco NI para redes 6G*. Accessed: 2025-09-09. 2024. URL: https://static.casadomo.com/media/2024/06/proyecto-daemon-marco-ni-red-6g.png.

[7] Metaswitch. *5G Slicing and PRB Allocation Diagram*. Accessed: 2025-09-09. 2023. URL: https://www.metaswitch.com/blog/5g-slicing-and-prb-allocation.

[8] N. Villegas, J. L. Herrera, L. Diez, D. Scotece, L. Foschini y R. Agüero. "DRL-based Dynamic MAC Scheduler Reconfiguration in O-RAN". En: *IEEE International Conference on Communications (ICC)*. Montreal, Canada: IEEE, 2025.

[9] K. Yu, K. Jin y X. Deng. "Review of Deep Reinforcement Learning". En: *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE. 2022, págs. 41-48.

[10] *NR; NR and NG-RAN Overall Description*. Inf. téc. TS 38.300. Release 17. 3GPP, 2023.

[11] M. Shafi, A. F. Molisch, P. J. Smith et al. “5G: A tutorial overview of standards, trials, challenges, deployment, and practice”. En: *IEEE Journal on Selected Areas in Communications* 35.6 (2017), págs. 1201-1221.

[12] Z. Liu y S. Han. “A Novel MAC Scheduling Based on Cross-layer Algorithm and Deep Learning”. En: *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*. 2022, págs. 333-338. DOI: 10.1109/ICCC56324.2022.10065651.

[13] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia y P. Camarda. “Downlink packet scheduling in LTE cellular networks: Key design issues and a survey”. En: *IEEE Communications Surveys & Tutorials* 15.2 (2013), págs. 678-700.

[14] K. Yang, N. Zhang et al. “Intelligent network slicing in 5G and beyond: Automating the management of network slices with AI”. En: *IEEE Communications Magazine* 57.6 (2019), págs. 94-100.

[15] M. Bennis, M. Debbah y H. V. Poor. “Ultra-reliable and low-latency wireless communication: Tail, risk, and scale”. En: *Proceedings of the IEEE* 106.10 (2018), págs. 1834-1853.

[16] U. Challita, A. Ferdowsi, M. Chen y W. Saad. “Machine learning for wireless connectivity and security of cellular-connected UAVs”. En: *IEEE Wireless Communications* 26.1 (2019), págs. 28-35.

[17] Y. Liu, X. Liu, C. Chen y Z. Han. “Deep reinforcement learning for resource allocation in 5G heterogeneous networks”. En: *IEEE Internet of Things Journal* 7.10 (2020), págs. 9721-9732.

[18] V. Mnih, K. Kavukcuoglu, D. Silver et al. “Human-level control through deep reinforcement learning”. En: *Nature* 518.7540 (2015), págs. 529-533.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford y O. Klimov. “Proximal policy optimization algorithms”. En: *arXiv preprint arXiv:1707.06347* (2017).

[20] F. Foundation. *Gymnasium Documentation*. Accessed: 2025-09-11. 2025. URL: https://gymnasium.farama.org/.

[21] O. Developers. *Optuna: A Hyperparameter Optimization Framework*. Accessed: 2025-09-12. 2025. URL: https://optuna.org/.