



MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

# TRABAJO DE FIN DE MÁSTER

## *MODELO DE CORRELACIÓN QoS/QoE EN ENTORNOS UNIVERSITARIOS*



**Estudiante:** Barghuthi Barrena, Sara

---

**Directora:** Ibarrola Armendariz, Eva

**Curso:** 2022-2023



## Resumen

El Machine Learning es una rama de la inteligencia artificial que ha cobrado gran importancia en los últimos años debido a su capacidad para aprender patrones y tomar decisiones en base a datos. En la actualidad, esta tecnología se utiliza en una amplia variedad de aplicaciones, desde el procesamiento de imágenes y el reconocimiento de voz hasta la predicción del comportamiento del consumidor y la toma de decisiones empresariales.

El objetivo específico del presente trabajo de final de máster es desarrollar un modelo de correlación entre la QoE y la QoS utilizando técnicas de Machine Learning. Para ello, se llevará a cabo un estudio en entornos universitarios de diferentes perfiles, en concreto en la ESI de Bilbao y en la Facultad de Educación de Bilbao. El propósito es determinar si existe una correlación entre la calidad experimentada y la calidad del servicio objetivo en estos entornos mediante el uso de técnicas de Machine Learning, que permitirá desarrollar un modelo preciso y eficiente para este propósito.

## Laburpena

Machine Learning, adimen artifizialaren adar bat da, azken urteetan oso garrantzitsu bihurtu da, datuen patronak ikasteko eta erabakiak hartzeko gaitasuna dela eta. Gaur egun, teknologia hori aplikazio anitzetan erabiltzen da, irudiak prozesatzeko eta ahotsaren ezagutza lortzeko, bezeroaren jokabidea eta enpresarako erabakiak hartzea barne.

TFM honetako helburu espezifikoena Machine Learning teknikak erabiliz QoE eta QoS arteko korrelazio-metodo bat garatzea da. Horretarako, perfila ezberdin duten unibertsitate inguruetan azterketa bat burutuko da, bai eta Bilboko ESI eta Hezkuntza Fakultatean ere. Helburua da erabiltzaileen bizi emandako kalitatea eta objektibo duten zerbitzuaren kalitatea artean korrelazio bat existitzen den edo ez jakitea eta zerbitzuaren kalitatea hobetzeko zein jardun ahal den erabiltzaileen bizi emandako kalitatea hobetzeko. Machine Learning teknikak erabiliz, helburu hau lortzeko eredu zehatza eta eraginkorra garatuko da.

## Abstract

Machine Learning is a branch of Artificial Intelligence that has gained significant importance in recent years due to its ability to learn patterns and make decisions based on data. Currently, this technology is used in a wide variety of applications, from image processing and voice recognition to predicting consumer behavior and making business decisions.

The specific objective of this Master's Thesis is to develop a model for correlating QoE (experienced quality) and QoS (objective service quality) using Machine Learning techniques. To achieve this, a study will be conducted in university environments of different profiles, specifically at the ESI of Bilbao and the Faculty of Education of Bilbao. The purpose is to determine if there is a correlation between the experienced quality and objective service quality in these environments. The use of Machine Learning techniques will allow for the development of an accurate and efficient model for this purpose.

**Palabras clave: Machine Learning, QoS, QoE, correlación.**



## Tabla de contenido

<b>1. INTRODUCCIÓN</b>	<b>1</b>
<b>2. CONTEXTO</b>	<b>2</b>
<b>3. OBJETIVOS</b>	<b>3</b>
<b>4. BENEFICIOS</b>	<b>4</b>
4.1. Beneficios técnicos	4
4.2. Beneficios económicos	4
4.3. Beneficios sociales	4
<b>5. ESTADO DEL ARTE</b>	<b>6</b>
5.1. Historia y evolución	6
5.2. Análisis de algoritmos de ML	6
5.2.1. Aprendizaje supervisado	7
5.2.2. Aprendizaje no supervisado	7
5.2.3. Aprendizaje por refuerzo	9
5.2.4. Aprendizaje profundo	10
5.3. Análisis de entornos de desarrollo y lenguajes de programación para el ML	11
5.3.1. Entornos de desarrollo	11
5.3.1. Lenguajes de programación	14
5.4. Análisis de parámetros QoS en entornos WiFi	16
5.4.1. Ancho de banda	17
5.4.2. Throughput	18
5.4.3. Latencia	18
5.4.4. Jitter	19
5.4.5. Pérdida de paquetes	20
5.4.6. SNR	21
5.4.7. Priorización de tráfico	22
5.5. Análisis factores de influencia para la QoE en entornos WiFi	22
5.5.1. Factores de influencia humana	24
5.5.2. Factores de influencia del sistema	25
5.5.3. Factor de influencia del contexto	25
5.6. Análisis de metodologías de valoración	26
<b>6. ANÁLISIS DE ALTERNATIVAS</b>	<b>30</b>
6.1. Identificación del parámetro QoS	30
6.1.1. Throughput	31
6.1.2. SNR	31
6.1.3. Criterios de selección	32
6.2. Identificación del algoritmo de ML	33
6.2.1. Regresión lineal	34
6.2.2. Regresión Logística	34
6.2.3. Criterios de selección	35
6.2.4. Redes neuronales feedforward	36



6.2.5.	Redes neuronales convolucionales .....	38
6.2.6.	Redes neuronales recurrentes .....	38
6.2.7.	Criterios de selección.....	39
<b>6.3.</b>	<b>Factores de influencia .....</b>	<b>40</b>
6.3.1.	Factores humanos .....	41
6.3.2.	Factores del sistema .....	43
6.3.3.	Factores de contexto .....	44
6.3.4.	Criterios de selección.....	46
<b>6.4.</b>	<b>Entorno de desarrollo .....</b>	<b>48</b>
6.4.1.	Jupyter Notebooks.....	49
6.4.2.	Google Colab.....	49
6.4.3.	DataSpell.....	50
6.4.4.	Criterios de selección.....	50
<b>6.5.</b>	<b>Lenguaje de programación .....</b>	<b>51</b>
6.5.1.	Python.....	51
6.5.2.	R.....	52
6.5.3.	Criterios de selección.....	52
<b>6.6.</b>	<b>Metodología de valoración.....</b>	<b>53</b>
<b>7.</b>	<b>METODOLOGÍA.....</b>	<b>55</b>
<b>7.1.</b>	<b>Descripción de la solución .....</b>	<b>55</b>
7.1.1.	Desarrollo del modelo de correlación QoS/QoE .....	55
<b>7.2.</b>	<b>Análisis de resultados .....</b>	<b>81</b>
<b>7.3.</b>	<b>Descripción de tareas.....</b>	<b>84</b>
<b>7.4.</b>	<b>Diagrama de Gantt.....</b>	<b>86</b>
<b>8.</b>	<b>ASPECTOS ECONÓMICOS.....</b>	<b>87</b>
<b>8.1.</b>	<b>Horas internas .....</b>	<b>87</b>
<b>8.2.</b>	<b>Amortizaciones .....</b>	<b>87</b>
<b>8.3.</b>	<b>Gastos .....</b>	<b>87</b>
<b>8.4.</b>	<b>Resumen del coste del proyecto .....</b>	<b>87</b>
<b>9.</b>	<b>ANÁLISIS DE RIESGOS.....</b>	<b>89</b>
<b>9.1.</b>	<b>Demoras (A).....</b>	<b>89</b>
<b>9.2.</b>	<b>Fallo tecnológico (B).....</b>	<b>89</b>
<b>9.3.</b>	<b>Demoras en la aprobación de encuestas (C).....</b>	<b>89</b>
<b>9.4.</b>	<b>Falta de datos suficientes (D).....</b>	<b>90</b>
<b>9.5.</b>	<b>Matriz probabilidad – impacto .....</b>	<b>90</b>
<b>10.</b>	<b>CONCLUSIONES .....</b>	<b>91</b>
<b>11.</b>	<b>BIBLIOGRAFÍA.....</b>	<b>92</b>



## Índice de ilustraciones

ILUSTRACIÓN 1: EJEMPLO DE UN ALGORITMO DE "CLUSTERING" .....	8
ILUSTRACIÓN 2: EJEMPLO DE REDUCCIÓN DE DIMENSIONALIDAD .....	8
ILUSTRACIÓN 3: APRENDIZAJE POR REFUERZO .....	9
ILUSTRACIÓN 4: RED NEURONAL.....	10
ILUSTRACIÓN 5: ARQUITECTURA JERÁRQUICA QUE CLASIFICA LOS FI .....	23
ILUSTRACIÓN 6: MOS.....	27
ILUSTRACIÓN 7: DIAGRAMA DE LA ESTRUCTURA TÍPICA DE UNA RNAF.....	37
ILUSTRACIÓN 8: RED NEURONAL RECURRENTE .....	39
ILUSTRACIÓN 9: EJEMPLO DE DATOS OBJETIVOS DE LA RED WIFI EN LA ESCUELA DE INGENIERÍA.....	56
ILUSTRACIÓN 10: MAPA DE CALOR DE LA CORRELACIÓN ENTRE EL THROUGHPUT Y EL MOS EN LA FACULTAD DE INGENIERÍA .....	65
ILUSTRACIÓN 11: MAPA DE CALOR DE LA CORRELACION ENTRE EL THROUGHPUT Y EL MOS EN LA FACULTAD DE EDUCACIÓN .....	65
ILUSTRACIÓN 12: THROUGHPUT VS MOS: PREDICCIONES.....	71
ILUSTRACIÓN 13: THROUGHPUT VS. MOS: REALES.....	71
ILUSTRACIÓN 14: THRPUGHPUT VS. MOS: PREDICCIONES .....	71
ILUSTRACIÓN 15: THROPUGHPUT VS. MOS: REALES.....	72
ILUSTRACIÓN 16: DIFERENCIAS ENTRE LAS DOS FACULTADES: VALORES REALES Y PREDICHOS .....	73
ILUSTRACIÓN 17 THROUGHPUT VS. MOS: PREDICCIONES.....	78
ILUSTRACIÓN 18: THROUGHPUT VS. MOS: VALORES REALES.....	79
ILUSTRACIÓN 19: THROUGHPUT VS. MOS: PREDICCIONES.....	79
ILUSTRACIÓN 20: THRPUGHPUT VS. MOS: VALORES REALES .....	79
ILUSTRACIÓN 21: DIFERENCIAS ENTRE LAS DOS FACULTADES: VALORES REALES Y PREDICHOS .....	81
ILUSTRACIÓN 22: DIAGRAMA DE GANTT.....	86



## Índice de tablas

TABLA 1: MATRIZ DE ALTERNATIVAS DE LOS PARÁMETROS QOS.....	32
TABLA 2: MATRIZ DE ALTERNATIVAS DE ALGORITMOS DE REGRESIÓN .....	36
TABLA 3: MATRIZ DE ALTERNATIVAS DE LOS ALGORITMOS DE ML .....	40
TABLA 4: MATRIZ DE ALTERNATIVAS DE FACTORES HUMANOS .....	47
TABLA 5: MATRIZ DE ALTERNATIVAS DE FACTORES DE SISTEMA .....	47
TABLA 6: MATRIZ DE ALTERNATIVAS DE LOS FACTORES DE CONTEXTO.....	48
TABLA 7: MATRIZ DE ALTERNATIVAS DE LOS ENTORNOS DE DESARROLLO .....	51
TABLA 8: MATRIZ DE ALTERNATIVAS DE LOS LENGUAJES DE PROGRAMACIÓN .....	53
TABLA 9: ESTADÍSTICAS DESCRIPTIVAS.....	63
TABLA 10: MÉTRICAS DE EVALUACIÓN DEL MODELO .....	69
TABLA 11: VALORES DE COEFICIENTE E INTERCEPTO .....	70
TABLA 12: MÉTRICAS PARA LA EVALUACIÓN DEL MODELO.....	78
TABLA 13: PRESUPUESTO CORRESPONDIENTE A LAS HORAS INTERNAS .....	87
TABLA 14: PRESUPUESTO CORRESPONDIENTE A LAS AMORTIZACIONES.....	87
TABLA 15: PRESUPUESTO CORRESPONDIENTE A LOS GASTOS .....	87
TABLA 16: RESUMEN DE LOS COSTES DEL PROYECTO.....	87
TABLA 17: MATRIZ PROBABILIDAD-IMPACTO .....	90



## LISTA DE ACRÓNIMOS

**TFM:** Trabajo de Fin de Máster

**ML:** Machine Learning

**QoS:** Quality of Service

**QoE:** Quality of Experience

**KPI:** Key Performance Indicator

**KQI:** Key Quality Indicator

**SNR:** signal-to-noise ratio

**FNN:** Feed Forward Neural Network

**CNN:** Convolutional Neural Network

**RNN:** Recurrent Neural Network



## 1. INTRODUCCIÓN

En la actualidad, la calidad de los servicios de red se ha convertido en un aspecto fundamental para las organizaciones que desean ofrecer un servicio de calidad a sus usuarios. En este sentido, la calidad de experiencia (QoE) es imprescindible para medir la satisfacción del usuario final. La QoE se refiere a la percepción subjetiva del usuario sobre la calidad del servicio prestado por la red, mientras que la calidad de servicio (QoS) se refiere a los parámetros objetivos de rendimiento de la red.

Por otro lado, la aplicación de técnicas de Machine Learning (ML) está revolucionando muchos campos, incluido el de las redes y los servicios de comunicación. Con la creciente complejidad de las redes y la creciente cantidad de datos que se generan, el ML se ha convertido en una herramienta esencial para el análisis y la optimización de los servicios de red.

En este contexto, el objetivo de este Trabajo de fin de Máster es desarrollar un modelo de correlación de la QoS/QoE utilizando técnicas de ML en entornos universitarios de diferentes perfiles. Se realizará un estudio en los entornos universitarios de ciencias y humanidades de la Universidad del País Vasco (UPV/EHU), con el fin de analizar los parámetros de QoS objetivos que pueden influir en las redes 802.11, así como los aspectos de influencia en la percepción de los usuarios de las redes WiFi. Se identificarán los indicadores de calidad (KQI) más interesantes para cada tipología y perfil de usuario y se analizarán metodologías de valoración óptimas para la evaluación de la QoE. Finalmente, se identificarán las técnicas de ML más adecuadas para la correlación de la QoS/QoE y se validarán mediante la comparación de los dos escenarios planteados para el caso de estudio.

Este trabajo contribuirá al desarrollo de un modelo de correlación de la QoS/QoE que permita mejorar la calidad de los servicios de red en los entornos planteados, así como en otros ámbitos en los que la calidad de la red es fundamental para la satisfacción del usuario.

El presente Trabajo Fin de Máster se ha realizado en el grupo de investigación NQaS, un grupo de investigación de la UPV/EHU, que se especializa en el estudio de las tecnologías utilizadas en las redes de datos en aspectos relacionados con la QoS y la seguridad.





## 2. CONTEXTO

Para entender mejor la naturaleza de este proyecto, es esencial situarlo dentro de su contexto apropiado y analizar las circunstancias específicas bajo las cuales se está llevando a cabo. Esta contextualización permitirá una comprensión más profunda y completa de los objetivos, alcance y desafíos particulares de este trabajo.

Este proyecto se enmarca en el trabajo del grupo de investigación QoXfera, centrado en la correlación entre la QoE y la QoS en redes inalámbricas 802.11, específicamente en entornos universitarios. Este estudio se basa en desarrollar un modelo utilizando técnicas de Machine Learning para esta correlación.

Para realizar el modelo, se dispone de datos etiquetados derivados de encuestas de satisfacción, los cuales ofrecen una perspectiva directa sobre la QoE percibida por los usuarios. Por otro lado, se cuenta con datos recolectados directamente de las mediciones de red en entornos universitarios, que suministran información valiosa en torno a los parámetros técnicos que caracterizan la QoS.

No obstante, es crucial mencionar que se han experimentado retrasos en la recepción del permiso requerido por parte de la Comisión de Ética de Investigación del País Vasco (CEISH). Como consecuencia, la realización de las encuestas no ha sido posible dentro del tiempo inicialmente estimado. Ante esta situación, se ha optado por identificar y aplicar un método alternativo para recopilar los datos de QoE.

La utilidad y aplicabilidad de este modelo puede tener un impacto significativo en la optimización de redes WiFi en universidades, en términos de proporcionar una mejor experiencia de usuario y una gestión más eficiente del servicio.

Por último, es importante destacar que todos los datos utilizados en este proyecto han sido aprobados por el Comité de Ética de la Universidad del País Vasco (UPV/EHU), en el marco de la línea de investigación de la memoria presentada M10/2023/184, titulada "Análisis para la gestión global de la QoS en red WiFi en entorno universitario".



### 3. OBJETIVOS

El objetivo principal de este trabajo es el desarrollo de un modelo para correlacionar QoE y QoS en diferentes entornos universitarios y perfiles de estudiantes, incluyendo áreas de ciencias y humanidades, utilizando técnicas ML. Para ello, se establecen los siguientes objetivos específicos:

1. **Realizar una revisión bibliográfica** sobre las técnicas de machine learning más adecuadas para la correlación de la QoS/QoE en redes inalámbricas, así como sobre los parámetros de QoS objetivos y los aspectos que influyen en la percepción de los usuarios de las redes WiFi en entornos universitarios.
2. **Identificar los indicadores de calidad (KQI, KPI)** más relevantes para cada tipología y perfil de usuarios en los entornos de redes inalámbricas 802.11 de los casos de estudio (ESI de Bilbao y Facultad de Educación de Bilbao).
3. **Realizar un estudio de las metodologías de valoración** óptimas para la evaluación de la QoE y desarrollar los formularios adecuados, en base a los resultados de los factores de influencia identificados para la evaluación de las satisfacciones de los usuarios con el servicio y los requerimientos y expectativas que ellos tienen frente al mismo.
4. **Identificar las técnicas de ML** más adecuadas para la correlación de la QoS/QoE en los casos de estudio planteados y validar los resultados obtenidos mediante la comparación de los dos escenarios.



## 4. BENEFICIOS

### 4.1. Beneficios técnicos

En primer lugar, la implementación de este modelo permite mejorar la calidad de los servicios de red a través de una medición más precisa y objetiva de la calidad de la red, la identificación de los KQI más relevantes para cada tipología de usuario y el uso de técnicas de ML para la correlación de la QoS y la QoE. Esto significa que el modelo puede detectar problemas de calidad de servicio en la red de forma más eficiente y precisa, lo que permite una resolución más rápida y eficiente de estos problemas.

Además, la identificación de las necesidades específicas de los usuarios permite a los investigadores enfocar sus recursos en los aspectos más importantes para los mismos, lo que resulta en una optimización de la configuración de la red y una mejora general de la eficiencia de los servicios.

Otro beneficio técnico es la utilización de técnicas de machine learning para la correlación de la QoS y la QoE. Esto permitiría un análisis más preciso de los datos de la red (con la posibilidad de hacerlo en tiempo real), lo que facilitaría la detección de problemas y la toma de decisiones. Además, el modelo podría actualizarse continuamente para adaptarse a los cambios en la red y a las necesidades de los usuarios.

### 4.2. Beneficios económicos

El desarrollo de este proyecto tiene importantes beneficios económicos, especialmente en el sector comercial, específicamente en el sector de las telecomunicaciones. El enfoque del estudio en la QoE de las redes de telecomunicaciones, en particular en los entornos de redes inalámbricas IEEE 802.11, permite a los proveedores de estas redes mejorar la satisfacción del cliente y su fidelización.

La metodología desarrollada permite complementar el conocimiento existente sobre los clientes, lo que puede emplearse en campañas de marketing, servicios de atención al cliente, servicios de resolución de incidencias, y otros procesos de interacción con el cliente. Al incrementar la productividad, el resultado económico de las empresas también se ve incrementado. Además, el estudio de calidad permite conocer los factores clave que afectan la percepción de calidad, lo que evita sobredimensionar la red en puntos donde los usuarios ya están satisfechos con los niveles de calidad que reciben.

Asimismo, el modelo de correlación de la QoS/QoE permite una mejor identificación de los problemas de calidad de servicio, lo que reduce la necesidad de realizar costosas reparaciones y reemplazos en la infraestructura de la red lo que se traduce en una reducción de los costes a largo plazo.

### 4.3. Beneficios sociales

Este trabajo también ofrece una serie de beneficios sociales. Uno de los principales beneficios es la mejora de la calidad de vida de los usuarios de las redes de telecomunicaciones,



ya que permite enfocar los recursos en los aspectos más importantes para los usuarios, lo que se traduce en una mejor experiencia de uso y una mayor satisfacción general.

Es indudable que la mejora de la red WiFi en los campus universitarios puede suponer un cambio significativo en la calidad de la educación. El acceso a una conexión de red de alta calidad facilita el acceso a la información, permite una comunicación fluida y mejora la eficiencia de las herramientas digitales.

Además, este proyecto alinea sus esfuerzos con los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas, específicamente con el ODS 4, que busca garantizar una educación inclusiva, equitativa y de calidad, y promover oportunidades de aprendizaje durante toda la vida para todos. Mejorar la infraestructura de red de las universidades es un paso importante hacia la creación de un ambiente educativo más inclusivo y accesible.

Además de las mencionadas implicaciones educativas y sociales, este trabajo también está directamente alineado con el Objetivo de Desarrollo Sostenible (ODS) 7 de las Naciones Unidas: "Energía asequible y no contaminante". Esta alineación se refleja en el potencial del proyecto para reducir el consumo de energía y minimizar el impacto ambiental de la infraestructura de red de las universidades.

La optimización de la QoE y la QoS a través de técnicas de Machine Learning puede llevar a una asignación más eficiente de los recursos de la red. Al concentrarse en los aspectos más importantes para los usuarios, se puede evitar la asignación de recursos en áreas que son menos relevantes o que no contribuyen significativamente a la experiencia del usuario. Esto puede resultar en un consumo de energía reducido, ya que menos energía es desperdiciada en procesos y operaciones innecesarias.



## 5. ESTADO DEL ARTE

La inteligencia artificial (IA) y Machine Learning (ML)[1] son dos campos cada vez más populares en la tecnología actual. El ML es una rama de la inteligencia artificial que se dedica al estudio de algoritmos y técnicas que permiten que las máquinas aprendan a partir de datos. Su objetivo es desarrollar sistemas que puedan aprender automáticamente y mejorar su rendimiento a medida que adquieren más datos.

En este estudio de estado del arte, se abordará primero una breve reseña histórica y evolutiva del ML, para continuar con un análisis detallado de las distintas tecnologías y modelos existentes en la actualidad en esta disciplina.

### 5.1. Historia y evolución

Los primeros hechos que sentaron las bases de esta rama de la inteligencia artificial se remontan al siglo XIX. En 1812 el Teorema de Bayes se convirtió en el primer sistema de predecir un evento futuro en base a una serie de condiciones previas.

En el siglo XX, con el avance de la programación informática y la pregunta planteada por Alan Turing sobre si las máquinas eran capaces de pensar, surgieron las redes neuronales artificiales y los primeros programas diseñados para ofrecer soluciones "creativas" a problemas dados.

A medida que se desarrollaron las tecnologías, los expertos en computación comenzaron a enfocarse en el ML mediante la utilización de datos y dejando que los programas desarrollaran reglas en base a ellos. Sin embargo, la escasez de datos y la falta de potencia computacional eran problemas que limitaban el avance en esta disciplina.

Con el paso del tiempo, el avance tecnológico ha permitido el procesamiento de mayores volúmenes de datos, lo que ha permitido desarrollar sistemas de ML cada vez más complejos y eficientes. En la actualidad, ML es una disciplina en constante evolución y tiene un gran impacto en una variedad de campos, permitiendo la automatización y mejora de procesos, así como la toma de decisiones automatizadas.

### 5.2. Análisis de algoritmos de ML

En el campo del ML, existen diferentes técnicas de ML que se utilizan para resolver problemas específicos. Cada uno de estos tipos de aprendizaje tiene sus propias características y metodologías, y es importante elegir el adecuado para cada problema en particular. En este apartado, se describirán y analizarán los principales tipos de ML, incluyendo el aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje por refuerzo y el aprendizaje profundo.



### 5.2.1. Aprendizaje supervisado

El aprendizaje supervisado [2] es una técnica de aprendizaje automático en la que se utiliza un conjunto de datos etiquetados para entrenar un modelo. El objetivo es que el modelo aprenda a hacer predicciones precisas sobre nuevos datos a partir de los patrones presentes en los datos de entrenamiento. Los algoritmos de aprendizaje supervisado se dividen en dos categorías principales: clasificación y regresión.

- **Clasificación:** La clasificación es una tarea de aprendizaje supervisado en la que se busca asignar una etiqueta o categoría a una entrada. Por ejemplo, en el caso de un sistema de reconocimiento de imágenes, la entrada podría ser una imagen y las etiquetas podrían ser "perro" o "gato". Los algoritmos de clasificación más comunes incluyen árboles de decisión, redes neuronales y algoritmos basados en vecinos más cercanos.
- **Regresión:** La regresión es una tarea de aprendizaje supervisado en la que se busca predecir un valor numérico a partir de una entrada. Por ejemplo, en el caso de un sistema de predicción de precios de acciones, la entrada podría ser información financiera sobre una empresa y el valor a predecir podría ser el precio de la acción. Los algoritmos de regresión más comunes incluyen regresión lineal, regresión logística y redes neuronales.

En el aprendizaje supervisado, se divide el conjunto de datos en dos conjuntos: el conjunto de entrenamiento y el conjunto de prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se utiliza para evaluar el rendimiento del modelo. Es importante evaluar el rendimiento del modelo en un conjunto de datos independiente para evitar sesgos y sobrestimaciones.

Una de las principales ventajas del aprendizaje supervisado es su capacidad para manejar grandes cantidades de datos y su facilidad de uso. Sin embargo, también tiene algunas desventajas. En primer lugar, requiere un conjunto de datos etiquetado, lo que puede ser costoso. En segundo lugar, el modelo solo es capaz de realizar predicciones dentro del rango de los datos de entrenamiento, lo que lo hace vulnerable a las situaciones fuera de este rango. Por último, el modelo puede ser sensible a los ruidos y las outliers presentes en los datos de entrenamiento, lo que puede afectar negativamente su rendimiento.

### 5.2.2. Aprendizaje no supervisado

El aprendizaje no supervisado [3] es una técnica de aprendizaje automático en la que se utiliza un conjunto de datos no etiquetados para encontrar patrones y estructuras en los datos. El objetivo es descubrir relaciones ocultas y características relevantes en los datos. A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado el modelo no tiene acceso a las etiquetas o respuestas correctas de los datos, por lo que debe descubrir las estructuras y patrones de forma automática.

Existen varios algoritmos de aprendizaje no supervisado disponibles, cada uno de los cuales se enfoca en resolver problemas diferentes. Algunos ejemplos incluyen:



- **Agrupamiento:** El agrupamiento es una técnica de aprendizaje no supervisado en la que se busca dividir un conjunto de datos en grupos o "clusters" de forma que los datos dentro de cada cluster sean similares entre sí, tal y como se puede ver el ejemplo de la Ilustración 1. Los algoritmos de agrupamiento más comunes incluyen k-means, agrupamiento jerárquico y DBSCAN.

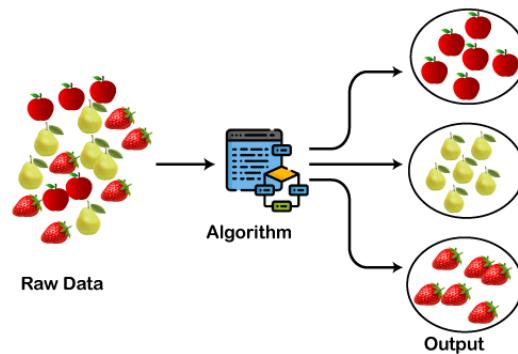


Ilustración 1: Ejemplo de un algoritmo de "clustering"

- **Reducción de dimensionalidad:** La reducción de dimensionalidad es una técnica de aprendizaje no supervisado en la que se busca representar un conjunto de datos en un espacio de menor dimensión (Ilustración 2), manteniendo la información relevante. Los algoritmos de reducción de dimensionalidad más comunes incluyen PCA, LLE y t-SNE.

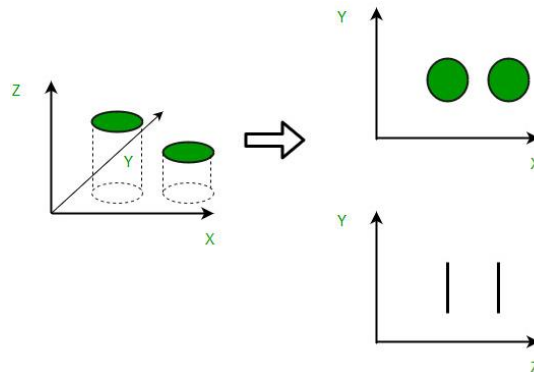


Ilustración 2: Ejemplo de reducción de dimensionalidad

- **Detección de anomalías:** La detección de anomalías es una técnica de aprendizaje no supervisado en la que se busca identificar datos que se desvían significativamente de los patrones normales. Los algoritmos de detección de anomalías más comunes incluyen algoritmos basados en distancias, algoritmos basados en distribuciones y algoritmos basados en redes neuronales.
- **Aprendizaje automático por asociación:** El aprendizaje automático por asociación es una técnica en la que se busca encontrar relaciones entre diferentes variables en un conjunto de datos. Por ejemplo, en un supermercado, podrían encontrar relaciones entre los productos que se compran juntos con frecuencia. Los algoritmos de aprendizaje automático por asociación más comunes incluyen el algoritmo Apriori y el algoritmo FP-Growth.



El aprendizaje no supervisado tiene varias ventajas en comparación con el aprendizaje supervisado. En primer lugar, no requiere un conjunto de datos etiquetados, lo que puede ser una ventaja en situaciones en las que las etiquetas no están disponibles o son costosas de obtener. En segundo lugar, el aprendizaje no supervisado permite explorar los datos y descubrir patrones y estructuras que de otra forma podrían pasar desapercibidos. También es útil para la comprensión y el resumen de datos, como la reducción de dimensionalidad y la generación de características.

Sin embargo, el aprendizaje no supervisado también tiene algunas desventajas. En primer lugar, el rendimiento del modelo puede ser difícil de evaluar ya que no hay una respuesta correcta o etiqueta. En segundo lugar, puede ser difícil interpretar los resultados obtenidos, especialmente en tareas complejas. Por último, algunos algoritmos de aprendizaje no supervisado son sensibles a la calidad y la cantidad de los datos.

### 5.2.3. Aprendizaje por refuerzo

El aprendizaje por refuerzo [4] es una técnica de aprendizaje automático en la cual un agente interactúa con su entorno y aprende a tomar acciones para maximizar una recompensa. El agente recibe una recompensa o un castigo en función de las acciones que toma, y utiliza esta información para mejorar su comportamiento en el futuro (Ilustración 3).

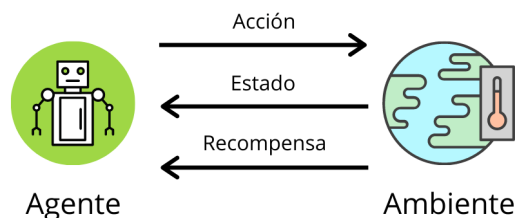


Ilustración 3: Aprendizaje por refuerzo

En el aprendizaje por refuerzo, el agente aprende a través de la experimentación y la retroalimentación. El agente se encuentra en un estado y toma una acción, lo que resulta en un nuevo estado y una recompensa. El agente utiliza esta información para actualizar sus conocimientos sobre cómo debe actuar en el futuro.

La política es el conjunto de reglas que el agente sigue para tomar acciones. El objetivo del aprendizaje por refuerzo es encontrar la política óptima que maximiza la recompensa a largo plazo. Una política óptima es aquella que toma las acciones que maximizan la esperanza de la recompensa futura.

Existen varios algoritmos de aprendizaje por refuerzo, como el algoritmo Q-learning, SARSA, y el algoritmo de política gradiente. El algoritmo Q-learning es uno de los más utilizados, y se basa en la idea de aprender una función de valor Q, que es la esperanza de la recompensa futura si el agente toma una acción en un estado dado. El algoritmo Q-learning actualiza la función Q utilizando la ecuación de Bellman, que es una ecuación recursiva que describe cómo la función Q debe actualizarse en función de la recompensa y la función Q de los estados futuros.





El algoritmo SARSA es similar al algoritmo Q-learning, pero utiliza la función Q para tomar acciones en lugar de la política óptima. El algoritmo de política gradiente es otro algoritmo de aprendizaje por refuerzo utilizado para encontrar políticas óptimas. Este algoritmo utiliza gradientes para actualizar la política, y se basa en la idea de que la política óptima se encuentra en la dirección del gradiente de la esperanza de la recompensa.

#### 5.2.4. Aprendizaje profundo

El aprendizaje profundo [5] es una técnica de aprendizaje automático que utiliza redes neuronales de varias capas para aprender representaciones de los datos. Una red neuronal es un modelo matemático inspirado en el cerebro humano que se compone de capas de nodos o neuronas interconectadas. Cada neurona recibe una entrada y produce una salida mediante una función de activación.

Las redes neuronales profundas se componen de varias capas de neuronas (**Error! Reference source not found.**), y se utilizan para aprender representaciones de los datos a medida que se procesan a través de las capas. Cada capa de una red neuronal profunda aprende una representación más abstracta de los datos, y las representaciones aprendidas se utilizan para realizar tareas como la clasificación, la generación de imágenes y el procesamiento del lenguaje natural.

Las redes neuronales profundas se componen de varios tipos de capas, como las capas de entrada, las capas ocultas y las capas de salida. La capa de entrada recibe los datos de entrada y los alimenta a través de la red neuronal. Cada capa oculta aprende una representación más abstracta de los datos a medida que se procesan a través de ella. La capa de salida produce la salida final de la red neuronal. En la Ilustración 4, se puede observar la representación de una red neuronal.

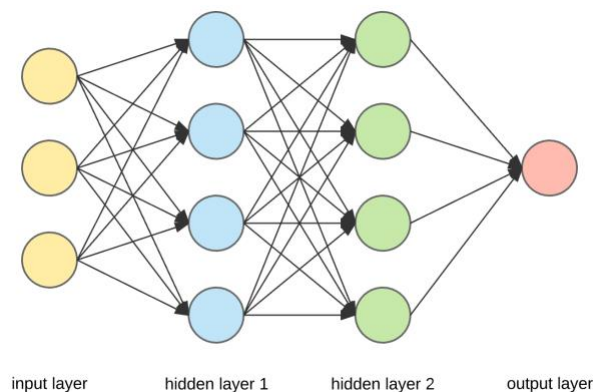


Ilustración 4: Red neuronal

Existen varios tipos de redes neuronales profundas, cada una con sus propias características y aplicaciones. Algunos de los más comunes incluyen:

- **Redes neuronales feedforward:** son las redes neuronales más simples y utilizadas. Son aquellas en las que las conexiones entre las neuronas solo van de una capa hacia la siguiente, sin retroalimentación.



- **Redes neuronales recurrentes:** son aquellas en las que las conexiones entre las neuronas van de una capa hacia la siguiente y también de una capa hacia una capa anterior. Esto permite que las redes neuronales recurrentes procesen secuencias de datos.
- **Redes neuronales convolucionales:** son un tipo especial de redes neuronales feedforward que se utilizan para procesar imágenes. Estas redes neuronales aprenden patrones en las imágenes mediante el uso de filtros y capas de pooling.
- **Redes neuronales autoencoder:** son un tipo de red neuronal feedforward que se utilizan para aprender una representación comprimida de los datos. Estas redes neuronales se componen de dos partes: un codificador que comprime los datos y un decodificador que los reconstruye.

### 5.3. Análisis de entornos de desarrollo y lenguajes de programación para el ML

Los entornos de desarrollo son software que proporcionan herramientas y funcionalidades para facilitar la escritura, prueba y depuración de código. Estos entornos de desarrollo son esenciales para los desarrolladores de ML, ya que permiten aumentar la eficiencia y productividad en el desarrollo de aplicaciones.

Asimismo, los lenguajes de programación son la base fundamental para el desarrollo de aplicaciones de ML. Existen varios lenguajes de programación populares para ML, como Python, R, Java, entre otros. Cada uno de estos lenguajes de programación tiene sus propias características y ventajas, lo que los hace más adecuados para diferentes tareas y aplicaciones.

En este apartado, se profundizará en cada uno de estos aspectos para entender mejor las características y ventajas de las diferentes herramientas y tecnologías disponibles para el desarrollo de aplicaciones de ML.

#### 5.3.1. Entornos de desarrollo

En la actualidad, existen varios entornos de desarrollo que se utilizan comúnmente en el desarrollo de proyectos de ML. Estos entornos ofrecen una variedad de características y herramientas diseñadas específicamente para ayudar a trabajar con bibliotecas y marcos de ML. A continuación, se presenta un análisis de los entornos de desarrollo más utilizados para el ML.

##### 5.3.1.1. *Jupyter Notebooks*

Jupyter [6] es una plataforma de código abierto que permite trabajar con código en una amplia gama de lenguajes de programación, incluyendo Python, R, Julia, y otros. La plataforma se utiliza comúnmente en proyectos de ML, ya que ofrece una interfaz de usuario fácil de usar para trabajar con modelos de ML, y cuenta con una amplia gama de herramientas y características para construir modelos de alta calidad.

Una de las características más destacadas de Jupyter es su capacidad para trabajar con notebooks interactivos. Estos notebooks permiten escribir y ejecutar código en tiempo real, lo que permite explorar datos, experimentar con diferentes modelos de ML, y visualizar los



resultados de sus análisis. Los notebooks también permiten documentar el trabajo en línea, lo que hace que sea más fácil compartir y colaborar en proyectos de ML.

Otra característica útil de Jupyter es su capacidad para integrarse con una amplia gama de bibliotecas y frameworks de ML. Por ejemplo, la plataforma es compatible con TensorFlow, PyTorch, Scikit-learn, Keras, y muchos otros. Esto significa se puede trabajar con una variedad de herramientas y frameworks de ML dentro de un entorno coherente y fácil de usar.

Jupyter también cuenta con una amplia gama de herramientas y funciones para la exploración y visualización de datos. Cuenta con herramientas integradas de visualización de datos, que permiten crear gráficos y visualizaciones para ayudar a entender mejor los patrones y tendencias en los datos. Jupyter también es compatible con una amplia gama de bibliotecas de análisis de datos, lo que posibilita realizar modelos con datos de una variedad de fuentes.

En términos de colaboración, Jupyter es una plataforma ideal para proyectos de ML. Los notebooks pueden ser compartidos y colaborados en tiempo real, lo que hace que sea fácil trabajar con otros desarrolladores y compartir el trabajo con colegas y colaboradores. Además, los notebooks son compatibles con Git y otros sistemas de control de versiones, que facilita la colaboración en proyectos de código abierto y ayuda a asegurar que los cambios en el código sean rastreados y documentados adecuadamente.

#### 5.3.1.2. *Google Colab*

Google Colab [7], abreviación de Google Colaboratory, es una plataforma de código abierto basada en la nube que permite a los usuarios escribir y ejecutar código en Python. Colab se ejecuta en el entorno de Google Drive y proporciona un entorno de desarrollo interactivo similar a Jupyter Notebook.

Una de las ventajas clave de Google Colab es que se ejecuta completamente en la nube, lo que significa que no requiere una configuración local. Los usuarios pueden acceder a Colab a través de un navegador web, lo que les permite trabajar en proyectos de ML sin necesidad de instalar software adicional en su ordenador. Esto hace que Colab sea especialmente conveniente para aquellos que no tienen acceso a recursos de hardware potentes o que desean colaborar y compartir su trabajo fácilmente.

Google Colab ofrece muchas características similares a Jupyter Notebook. Los usuarios pueden crear y ejecutar celdas de código en tiempo real, lo que facilita la experimentación y la iteración rápida. Colab también es compatible con varios lenguajes de programación, pero se centra principalmente en Python.

Una de las características más destacadas de Google Colab es su integración con Google Drive y otras herramientas de Google. Esto permite a los usuarios importar y exportar datos directamente desde y hacia su Google Drive, lo que facilita el acceso a conjuntos de datos y archivos de proyecto. Además, Colab ofrece una amplia gama de bibliotecas y frameworks populares de ML preinstalados, como TensorFlow, Keras y PyTorch, lo que facilita comenzar a trabajar en proyectos de ML sin tener que preocuparse por la instalación y configuración de las bibliotecas.



Al igual que Jupyter, Google Colab ofrece herramientas y funciones de visualización de datos para explorar y visualizar los resultados de los análisis de datos. Los usuarios pueden crear gráficos y visualizaciones interactivas directamente en las celdas de código, lo que facilita la comprensión y la presentación de los resultados.

En términos de colaboración, Google Colab también es una plataforma ideal. Los usuarios pueden compartir sus notebooks de Colab con otros usuarios, permitiendo la colaboración en tiempo real y la revisión conjunta del código y los resultados. Además, Colab es compatible con GitHub, lo que facilita la integración con sistemas de control de versiones y permite a los usuarios realizar un seguimiento de los cambios y colaborar en proyectos de forma efectiva.

#### 5.3.1.3. *DataSpell*

JetBrains DataSpell [8] es un entorno de desarrollo integrado (IDE) diseñado para facilitar el trabajo en proyectos de ML. Está específicamente diseñado para trabajar con lenguajes de programación populares para ML, como Python, y viene equipado con una amplia gama de herramientas y características útiles para la construcción de modelos de ML efectivos.

En términos generales, DataSpell ofrece una interfaz de usuario intuitiva y fácil de usar que permite trabajar de manera más eficiente. Asimismo, la plataforma viene equipada con una serie de herramientas y funciones para automatizar muchas de las tareas repetitivas asociadas con el desarrollo de modelos de ML.

Una de las características más destacadas de DataSpell es su integración con los frameworks de ML más populares, incluyendo TensorFlow, PyTorch, Scikit-learn y MXNet. Esto significa que se pueden realizar proyectos que utilizan estos frameworks de manera más eficiente, lo que puede ayudar a reducir el tiempo de desarrollo y acelerar el proceso de iteración del modelo.

Otra característica útil de DataSpell es su capacidad para ejecutar modelos de ML en la plataforma de nube de JetBrains. Esto implica que se pueden ejecutar modelos en un entorno de nube seguro y escalable sin tener que preocuparse por configurar su propia infraestructura de nube. Esto puede ser especialmente útil para proyectos que requieren recursos de computación escalables, como el entrenamiento de modelos de gran tamaño.

Por último, DataSpell también viene con una serie de herramientas de visualización y análisis de datos integradas para la exploración y visualización de datos. Esto puede ayudar a identificar patrones y tendencias en los datos, lo que a su vez puede mejorar la calidad del modelo de ML resultante.

#### 5.3.1.4. *PyCharm*

PyCharm [9] es un entorno de desarrollo integrado (IDE) desarrollado por JetBrains que está diseñado para trabajar con proyectos de Python. Aunque PyCharm se usa comúnmente para proyectos de desarrollo de software, también es una opción popular para proyectos de ML debido a su conjunto de herramientas y características diseñadas específicamente para ese propósito.



PyCharm tiene capacidad para integrarse con una amplia gama de bibliotecas de ML. La plataforma es compatible con TensorFlow, PyTorch, Scikit-learn, Keras y otras bibliotecas de ML, así pues, se puede trabajar con una variedad de herramientas y frameworks de ML dentro de un entorno coherente y fácil de usar.

Además, PyCharm cuenta con una amplia gama de herramientas y funciones para el desarrollo de modelos de ML. Por ejemplo, la plataforma cuenta con herramientas de depuración y análisis de datos integradas, por lo que facilita la identificación de errores en el código y la detección de patrones en los datos.

Otra característica útil de PyCharm es su capacidad para trabajar con notebooks de Jupyter. Se pueden crear y ejecutar notebooks de Jupyter dentro de PyCharm, en consecuencia, se pueden explorar datos, experimentar con diferentes modelos de ML y documentar su trabajo en línea.

En términos de colaboración, PyCharm es una buena plataforma para proyectos de ML. La plataforma es compatible con Git y otros sistemas de control de versiones, lo que facilita la colaboración en proyectos de código abierto y ayuda a asegurar que los cambios en el código sean rastreados y documentados adecuadamente. Además, PyCharm tiene la posibilidad de trabajar en equipos de manera efectiva mediante la asignación de tareas y el seguimiento del progreso del proyecto.

### 5.3.1. Lenguajes de programación

Además de la necesidad de utilizar un entorno de desarrollo óptimo para realizar proyectos de ML, es indispensable hacer uso de un lenguaje de programación que sea eficiente y adecuado para los mismos. Existen varios lenguajes de programación que se utilizan comúnmente en el desarrollo de proyectos de ML. A continuación, se presenta un análisis de los lenguajes de programación más utilizados.

#### 5.3.1.5. Python

Python [10] es un lenguaje de programación de alto nivel que se ha convertido en uno de los lenguajes más populares para el desarrollo de aplicaciones de ML. A continuación, se describen sus características principales:

- **Lenguaje de programación de alto nivel:** fácil de leer y escribir, lo que acelera el proceso de desarrollo.
- **Lenguaje de código abierto:** acceso al código fuente y modificaciones posibles.
- **Facilidad de uso:** fácil de aprender y utilizar, ideal para los que se están iniciando en el desarrollo de aplicaciones.
- **Portabilidad:** multiplataforma y se puede ejecutar en una amplia variedad de sistemas operativos y arquitecturas.
- **Multiparadigma:** admite varios paradigmas de programación, incluidos programación orientada a objetos, programación funcional y programación imperativa, permitiendo elegir el paradigma que mejor se adapte a sus necesidades.
- **Librerías de ML:** gran cantidad de librerías de ML disponibles en Python, lo que facilita el desarrollo de modelos y reduce el tiempo de desarrollo.



- **Flexibilidad:** Python es un lenguaje flexible y multiparadigma, lo que permite elegir el paradigma que mejor se adapte a las necesidades.
- **Comunidad activa:** Python tiene una comunidad de desarrolladores muy activa, por tanto, significa que hay una gran cantidad de recursos disponibles en línea y una gran cantidad de personas dispuestas a ayudar.
- **Facilidad de integración:** Python se integra fácilmente con otras tecnologías y herramientas, por lo que facilita la integración de aplicaciones de ML en otros sistemas.

#### 5.3.1.6. R

R [11] es un lenguaje de programación y un entorno de software de código abierto utilizado en análisis estadísticos y ML. R es un lenguaje que se enfoca en el análisis de datos y, por lo tanto, tiene una gran cantidad de librerías específicas para el análisis de datos y la creación de modelos de ML. A continuación, se describen sus características principales:

- **Lenguaje de programación de alto nivel:** R es un lenguaje de programación de alto nivel que es fácil de leer y escribir. Esto hace que el proceso de desarrollo sea más rápido y eficiente.
- **Lenguaje de código abierto:** R es un lenguaje de código abierto, lo que significa que se puede acceder al código fuente y modificarlo según sea necesario.
- **Facilidad de uso:** R es un lenguaje fácil de aprender y utilizar, lo que lo hace ideal para aquellos que se están iniciando en el desarrollo de aplicaciones.
- **Las librerías de ML:** R tiene una gran cantidad de librerías de ML, lo que facilita el desarrollo de modelos y reduce el tiempo de desarrollo.
- **Amplias capacidades estadísticas:** R es un lenguaje especialmente diseñado para análisis estadístico, lo que significa que tiene una amplia gama de capacidades estadísticas incorporadas.
- **Flexibilidad:** R es un lenguaje flexible y multiparadigma, lo que permite elegir el paradigma que mejor se adapte a sus necesidades.
- **Visualización de datos:** R tiene capacidades incorporadas de visualización de datos, lo que facilita la exploración y análisis de grandes conjuntos de datos.
- **Comunidad activa:** R tiene una comunidad de desarrolladores muy activa, lo que significa que hay una gran cantidad de recursos disponibles en línea y una gran cantidad de personas dispuestas a ayudar.

#### 5.3.1.7. Java

Java [12] es un lenguaje de programación orientado a objetos de alto nivel utilizado en una amplia gama de aplicaciones de software. Java también se ha utilizado cada vez más en el campo del ML. A continuación, se describen sus características principales:

- **Orientación a objetos:** Java es un lenguaje orientado a objetos, lo que significa que las aplicaciones de ML pueden ser programadas en una estructura de clases y objetos, que facilita la comprensión y el mantenimiento del código.
- **Plataforma independiente:** Java se ejecuta en una máquina virtual, es decir, que el código Java puede ser ejecutado en cualquier plataforma que tenga una máquina virtual de Java instalada.
- **Seguridad:** Java tiene un modelo de seguridad sólido que lo hace interesante para el desarrollo de aplicaciones que manejan datos sensibles.



- **Escalabilidad:** Java es un lenguaje escalable y se puede utilizar para desarrollar aplicaciones de cualquier tamaño.
- **Lenguaje de programación sólido:** Java es un lenguaje de programación sólido y confiable que se ha utilizado en una amplia gama de aplicaciones de software.
- **Librerías y marcos de trabajo de ML:** Java tiene una amplia gama de librerías y marcos de trabajo de ML, lo que permite el desarrollo de modelos de ML sofisticados.
- **Rendimiento:** Java es conocido por tener un buen rendimiento y puede manejar grandes conjuntos de datos y modelos de ML complejos.
- **Facilidad de integración:** Java se integra fácilmente con otras tecnologías y herramientas, lo que facilita la integración de aplicaciones de ML en otros sistemas.
- **Comunidad activa:** Java tiene una comunidad de desarrolladores muy activa, lo que significa que hay una gran cantidad de recursos disponibles en línea y una gran cantidad de personas dispuestas a ayudar.

#### 5.3.1.8. C++

C++ [13] es un lenguaje de programación de alto nivel utilizado en una amplia variedad de aplicaciones de software, incluyendo el campo del ML. A continuación, se describen sus características principales:

- **Velocidad y eficiencia:** C++ es un lenguaje de programación conocido por su alta velocidad y eficiencia. Esto lo hace ideal para aplicaciones de ML que involucran grandes conjuntos de datos y modelos de ML complejos.
- **Orientación a objetos:** C++ es un lenguaje de programación orientado a objetos, lo que significa que las aplicaciones de ML pueden ser programadas en una estructura de clases y objetos, lo que facilita la comprensión y el mantenimiento del código.
- **Portabilidad:** C++ es un lenguaje de programación altamente portátil y puede ser utilizado en una variedad de plataformas y sistemas operativos.
- **Rendimiento:** C++ es conocido por tener un excelente rendimiento y puede manejar grandes conjuntos de datos y modelos de ML complejos.
- **Librerías y marcos de trabajo de ML:** C++ tiene una amplia gama de librerías y marcos de trabajo de ML, lo que permite el desarrollo de modelos de ML sofisticados.
- **Acceso a hardware de bajo nivel:** C++ permite un acceso directo al hardware de la computadora, lo que lo hace ideal para aplicaciones de ML que requieren el uso de hardware de bajo nivel, como aceleradores de hardware de GPU.
- **Flexibilidad:** C++ es un lenguaje de programación altamente flexible que puede ser utilizado para una variedad de aplicaciones de ML, incluyendo el desarrollo de sistemas de aprendizaje profundo, redes neuronales y algoritmos de clasificación.

## 5.4. Análisis de parámetros QoS en entornos WiFi

La calidad de servicio (QoS) es un conjunto de parámetros que determinan el rendimiento y la calidad de las redes. Los parámetros de QoS son especialmente importantes en redes WiFi, donde se enfrentan a muchos desafíos en términos de capacidad, alcance y seguridad. Este análisis se centrará en los principales parámetros de QoS en redes WiFi y su impacto en el rendimiento de la red.



#### 5.4.1. Ancho de banda

El ancho de banda es uno de los parámetros más importantes en la QoS de los entornos WiFi. Según la Recomendación UIT-T Y.1540 el ancho de banda se define como la cantidad de datos que se pueden transmitir en una unidad de tiempo, medido en bits por segundo (bps), y es esencial para la capacidad de una red para soportar aplicaciones de alta velocidad como el streaming de video y audio, la descarga de archivos grandes, y la navegación web rápida.

El ancho de banda disponible en una red WiFi depende de varios factores, incluyendo la tecnología utilizada (por ejemplo, WiFi 4, 5 o 6), la frecuencia de operación (2,4 GHz o 5 GHz), el número de antenas, la calidad de la señal, el número de usuarios conectados, y la cantidad de datos que se están transmitiendo en ese momento.

A continuación, se detallan algunos aspectos importantes que condicionan el ancho de banda en la QoS en entornos WiFi:

- **Tecnología WiFi:** El ancho de banda en una red WiFi está limitado por la tecnología utilizada. Por ejemplo, WiFi 4 (802.11n) ofrece una velocidad máxima de hasta 600 Mbps, mientras que WiFi 5 (802.11ac) puede alcanzar velocidades de hasta 1,3 Gbps y WiFi 6 (802.11ax) puede llegar a velocidades de hasta 9,6 Gbps. Por lo tanto, la elección de la tecnología WiFi es crucial para determinar el ancho de banda disponible y la QoS que se puede esperar de una red.
- **Frecuencia de operación:** La frecuencia de operación también influye en el ancho de banda disponible. El espectro de 2,4 GHz tiene menos ancho de banda disponible en comparación con el espectro de 5 GHz. Por lo tanto, las redes que operan en el espectro de 5 GHz pueden proporcionar una mayor capacidad y, por tanto, una mejor QoS.
- **Número de antenas:** La cantidad de antenas en un punto de acceso WiFi también puede afectar el ancho de banda disponible. Cuantas más antenas tenga el punto de acceso, más capacidad tendrá para transmitir y recibir datos. Por lo tanto, un punto de acceso con múltiples antenas puede proporcionar una mayor QoS que un punto de acceso con una sola antena.
- **Calidad de la señal:** La calidad de la señal es un factor importante en la determinación del ancho de banda disponible. Una señal débil o interferencias pueden disminuir la calidad de la señal y limitar la cantidad de datos que se pueden transmitir. Por lo tanto, es importante que la señal WiFi sea lo más fuerte y clara posible para obtener el máximo ancho de banda disponible.
- **Número de usuarios conectados:** El número de usuarios conectados a una red WiFi puede afectar la QoS y el ancho de banda disponible. Cuantos más usuarios se conecten a la red, mayor será la carga de la red y menor será el ancho de banda disponible para cada usuario. Por lo tanto, es importante limitar el número de usuarios conectados y distribuirlos en diferentes canales y frecuencias para mejorar la QoS.
- **Cantidad de datos que se están transmitiendo:** La cantidad de datos que se están transmitiendo en la red también afecta el ancho de banda disponible.





#### 5.4.2. Throughput

El throughput un parámetro muy importante en la QoS de los entornos WiFi, y se define como la cantidad de datos que se pueden transmitir con éxito en una unidad de tiempo determinada. El throughput se mide típicamente en bits por segundo (bps).

Según la Recomendación ITU-T Y.1540, el throughput es uno de los parámetros clave para evaluar la calidad de una red, se define como la cantidad de datos que se pueden transmitir en una unidad de tiempo, medido en bits por segundo (bps). Además, se establece que el throughput se ve afectado por varios factores, como la capacidad de la red, la congestión de la red y los requisitos de ancho de banda de las aplicaciones.

A continuación, se detallan algunos aspectos importantes que condicionan el throughput en la QoS en entornos WiFi:

- **Tecnología WiFi:** Al igual que con el ancho de banda, la tecnología WiFi utilizada es un factor clave en la determinación del throughput disponible en una red. Las redes WiFi más nuevas, como WiFi 6 (802.11ax), ofrecen velocidades más altas y mayor eficiencia espectral, lo que puede conducir a un mayor throughput.
- **Frecuencia de operación:** La frecuencia de operación también puede influir en el throughput disponible. En general, las redes WiFi que operan en el espectro de 5 GHz tienen un mayor throughput que las que operan en el espectro de 2,4 GHz. Esto se debe a que el espectro de 5 GHz tiene más canales disponibles y es menos susceptible a interferencias.
- **Número de antenas:** Al igual que con el ancho de banda, la cantidad de antenas en un punto de acceso WiFi también puede afectar el throughput disponible. Los puntos de acceso con múltiples antenas pueden proporcionar un mayor throughput que los puntos de acceso con una sola antena, ya que tienen más capacidad para transmitir y recibir datos.
- **Calidad de la señal:** La calidad de la señal es un factor importante en la determinación del throughput disponible. Una señal débil o interferencias pueden disminuir el throughput y limitar la cantidad de datos que se pueden transmitir con éxito. Por lo tanto, es importante que la señal WiFi sea lo más fuerte y clara posible para obtener el máximo throughput disponible.
- **Número de usuarios conectados:** El número de usuarios conectados a una red WiFi también puede afectar el throughput disponible. Cuantos más usuarios se conecten a la red, mayor será la carga de la red y menor será el throughput disponible para cada usuario. Por lo tanto, es importante limitar el número de usuarios conectados y distribuirlos en diferentes canales y frecuencias para mejorar la QoS.
- **Cantidad de datos que se están transmitiendo:** La cantidad de datos que se están transmitiendo en la red también afecta el throughput disponible. Cuanto más alto sea el tráfico de red, menor será el throughput disponible para cada usuario.

#### 5.4.3. Latencia

La latencia es un parámetro clave en la QoS de los entornos WiFi. Según la recomendación ITU-T Y.1541 la latencia es el tiempo de retraso que transcurre desde que se envía un paquete de datos desde un origen hasta que se recibe en su destino. La latencia tiene



un impacto significativo en la QoS de una red, ya que afecta directamente la velocidad y la capacidad de respuesta de la red. Una latencia alta puede hacer que las aplicaciones y servicios en línea se vuelvan lentos e ineficientes, que puede resultar en una experiencia de usuario pobre y en una disminución de la productividad. La recomendación ITU-T Y.1540 establece que la latencia debe ser controlada y mantenida dentro de un rango específico para garantizar una buena QoS para los usuarios.

A continuación, se detallan los aspectos más importantes relacionados con la latencia en la QoS en entornos WiFi:

- **Tecnología WiFi:** La tecnología WiFi utilizada puede afectar significativamente a la latencia en una red inalámbrica. Las tecnologías WiFi más recientes, como WiFi 6 (802.11ax), están diseñadas para reducir la latencia en comparación con las tecnologías WiFi anteriores, como WiFi 4 (802.11n) o WiFi 5 (802.11ac). La tecnología WiFi 6 utiliza una técnica llamada OFDMA (Acceso Múltiple por División de Frecuencia Ortogonal) que permite una mejor asignación del ancho de banda y una menor latencia en redes con muchos usuarios.
- **Congestión de la red:** La congestión de la red es un factor que puede aumentar la latencia en una red WiFi. Si muchos dispositivos están conectados a la red y están transmitiendo datos al mismo tiempo, la red puede estar sobrecargada y la latencia aumentará. Para evitar la congestión, es importante limitar el número de dispositivos conectados a la red y distribuirlos en diferentes canales y frecuencias.
- **Distancia entre los dispositivos:** La distancia entre los dispositivos también puede afectar la latencia en una red WiFi. Cuanto mayor sea la distancia entre el punto de acceso y el dispositivo, mayor será la latencia debido al tiempo que tarda la señal en viajar de ida y vuelta.
- **Interferencia de señal:** La interferencia de señal es otro factor que puede aumentar la latencia en una red WiFi. Si hay otras redes inalámbricas cerca que utilizan la misma frecuencia, la señal WiFi puede sufrir interferencias y la latencia aumentará. Para evitar la interferencia de señal, es importante elegir una frecuencia que no esté siendo utilizada por otras redes y utilizar canales que tengan la menor interferencia posible.
- **Tipo de aplicación:** El tipo de aplicación que se está utilizando también puede afectar la latencia en una red WiFi. Las aplicaciones que requieren una conexión rápida y estable, como las aplicaciones de juegos en línea y las videollamadas, pueden ser más sensibles a la latencia que otras aplicaciones. Para mejorar la QoS, es importante utilizar una tecnología WiFi adecuada y asegurarse de que la red esté diseñada para proporcionar una latencia mínima.

#### 5.4.4. Jitter

El jitter es un parámetro que está estrechamente relacionado con la latencia en una red inalámbrica, por lo que también lo hace un factor importante en la QoS de los entornos WiFi. Como se ha explicado en el apartado anterior, la latencia es el tiempo que tardan los paquetes de datos en viajar desde el origen al destino a través de la red, mientras que el jitter se refiere a la variación en la latencia de los paquetes de datos que se transmiten.

Es importante tener en cuenta que la recomendación Y.1540 establece que el jitter máximo permitido para un servicio de calidad aceptable es de 30 milisegundos en la mayoría de



las situaciones, y de 10 milisegundos en situaciones críticas. Por lo tanto, es esencial que los proveedores de servicios de red monitoreen y gestionen cuidadosamente el jitter para garantizar que cumplan con estas especificaciones y brinden una QoS adecuada.

La latencia y el jitter están relacionados porque una variación significativa en la latencia puede causar un aumento en el jitter. A continuación, se detallan algunos factores importantes relacionados con el jitter:

- **Tecnología WiFi:** La tecnología WiFi utilizada puede afectar significativamente al jitter en una red inalámbrica. Las tecnologías Wi-Fi más recientes, como WiFi 6 (802.11ax), están diseñadas para reducir el jitter en comparación con las tecnologías Wi-Fi anteriores, como WiFi 4 (802.11n) o WiFi 5 (802.11ac). La tecnología WiFi 6 utiliza una técnica llamada "OFDMA" (Acceso Múltiple por División de Frecuencia Ortogonal) que permite una mejor asignación del ancho de banda y una menor variación en la latencia en redes con muchos usuarios.
- **Gestión de la red:** La congestión de la red es un factor que puede aumentar el jitter en una red WiFi. Si muchos dispositivos están conectados a la red y están transmitiendo datos al mismo tiempo, la red puede estar sobrecargada y la variación en la latencia aumentará. Para evitar la congestión, es importante limitar el número de dispositivos conectados a la red y distribuirlos en diferentes canales y frecuencias.
- **Interferencia de señal:** La interferencia de señal es otro factor que puede aumentar el jitter en una red WiFi. Si hay otras redes inalámbricas cerca que utilizan la misma frecuencia, la señal WiFi puede sufrir interferencias y la variación en la latencia aumentará. Para evitar la interferencia de señal, es importante elegir una frecuencia que no esté siendo utilizada por otras redes y utilizar canales que tengan la menor interferencia posible.
- **Calidad de la conexión del dispositivo:** La calidad de la conexión del dispositivo también puede afectar el jitter en una red WiFi. Si el dispositivo tiene una conexión débil con el punto de acceso, la variación en la latencia aumentará. Para mejorar la QoS es importante asegurarse de que el dispositivo tenga una conexión estable y de que esté lo más cerca posible del punto de acceso.

#### 5.4.5. Pérdida de paquetes

La pérdida de paquetes es un parámetro crítico en la QoS de los entornos WiFi. Se refiere a la tasa de paquetes de datos que se pierden en la transmisión a través de una red. La pérdida de paquetes puede afectar significativamente la QoS en una red inalámbrica. Cuando los paquetes de datos se pierden, la información que contienen no se puede transmitir correctamente, lo que puede afectar la calidad de la conexión. En algunos casos, la pérdida de paquetes puede hacer que la conexión se interrumpa por completo. Según la recomendación Y.1540, establece un límite máximo del 1% de pérdida de paquetes como aceptable para la mayoría de las aplicaciones en tiempo real.

A continuación, se detallan algunos aspectos importantes relacionados con la pérdida de paquetes en la QoS en entornos WiFi:

- **Gestión de la red:** La congestión de la red es un factor clave que puede aumentar la tasa de pérdida de paquetes en una red WiFi. Si muchos dispositivos están conectados



a la red y están transmitiendo datos al mismo tiempo, la red puede estar sobrecargada y la tasa de pérdida de paquetes aumentará.

- **Interferencia de señal:** La interferencia de señal es otro factor que puede aumentar la tasa de pérdida de paquetes en una red WiFi. Si hay otras redes inalámbricas cerca que utilizan la misma frecuencia, la señal WiFi puede sufrir interferencias y la tasa de pérdida de paquetes aumentará.
- **Calidad de la conexión del dispositivo:** La calidad de la conexión del dispositivo también puede afectar la tasa de pérdida de paquetes en una red WiFi. Si el dispositivo tiene una conexión débil con el punto de acceso, la tasa de pérdida de paquetes aumentará. Para mejorar la QoS, es importante asegurarse de que el dispositivo tenga una conexión estable y de que esté lo más cerca posible del punto de acceso.
- **Tasa de errores de transmisión:** La tasa de errores de transmisión es otro factor que puede aumentar la tasa de pérdida de paquetes en una red WiFi. Si hay muchos errores en la transmisión de los paquetes de datos, la tasa de pérdida de paquetes aumentará. Para reducir la tasa de errores de transmisión, es importante utilizar tecnologías WiFi avanzadas, como WiFi 6 (802.11ax), que ofrecen una mayor fiabilidad en la transmisión de datos.

#### 5.4.6. SNR

La relación señal-ruido (SNR, por sus siglas en inglés) es otro parámetro crítico para evaluar la QoS en los entornos WiFi. Este parámetro, medido en decibelios (dB), define el nivel de una señal deseada en comparación con el nivel de ruido de fondo, que incluye todo tipo de interferencias y distorsiones indeseables. Un SNR más alto implica una señal más clara y, por lo tanto, una mayor calidad de la conexión.

En el marco de la Recomendación ITU-T Y.1541, el SNR es un indicador crucial para determinar la calidad de una red. Es esencial para mantener la integridad de los datos transmitidos a través de la red, ya que una SNR baja puede resultar en una mayor tasa de errores y una disminución general en la calidad de la red.

Aquí se mencionan algunos factores que pueden influir en la SNR en la QoS en entornos WiFi:

- **Distancia al punto de acceso:** La distancia entre el dispositivo del usuario y el punto de acceso WiFi puede tener un impacto significativo en la SNR. Cuanto mayor sea la distancia, más probable es que la señal se debilite, lo que resulta en un SNR más bajo.
- **Obstáculos físicos:** Los obstáculos físicos como paredes, muebles, y otros objetos pueden atenuar la señal WiFi y aumentar el ruido, lo que reduce la SNR. Por lo tanto, el diseño y la ubicación de los puntos de acceso deben considerarse cuidadosamente para minimizar la cantidad de obstáculos entre el punto de acceso y los dispositivos del usuario.
- **Interferencias:** Las interferencias de otras redes WiFi, dispositivos electrónicos, y fuentes de ruido electromagnético pueden afectar significativamente la SNR. La selección cuidadosa de canales y frecuencias, así como la utilización de tecnologías como la cancelación de ruido, pueden ayudar a minimizar las interferencias.
- **Tecnología WiFi:** Las tecnologías WiFi más nuevas, como WiFi 6 (802.11ax), ofrecen técnicas avanzadas de gestión de interferencias que pueden resultar en un SNR más alto en entornos ruidosos.



- **Ancho de banda:** Un ancho de banda más amplio puede aumentar la cantidad de ruido, lo que reduce la SNR. Sin embargo, también puede permitir una mayor capacidad de datos, por lo que debe haber un equilibrio entre el ancho de banda y la SNR.

#### 5.4.7. Priorización de tráfico

Por último, la recomendación ITU-T Y.1540 también describe la necesidad de priorizar el tráfico de red en función de las necesidades específicas de las aplicaciones. Se establecen diferentes clases de servicio y se definen parámetros de QoS para cada una de ellas. La priorización del tráfico es el proceso de asignar diferentes niveles de prioridad a diferentes tipos de tráfico, con el fin de garantizar que los datos críticos tengan un mejor rendimiento que los datos no críticos. La priorización del tráfico es importante en los entornos WiFi donde se transmiten diferentes tipos de datos, como voz, video y datos.

A continuación, se detallan los métodos utilizados para implementar la priorización del tráfico en entornos WiFi:

- **Clasificación de tráfico:** La clasificación de tráfico es el proceso de identificar y separar diferentes tipos de tráfico en una red inalámbrica. Por ejemplo, el tráfico de voz y video se considera crítico y debe tener una prioridad más alta que el tráfico de datos no críticos. La clasificación de tráfico permite asignar diferentes niveles de prioridad a diferentes tipos de tráfico y garantizar que los datos críticos tengan un mejor rendimiento que los datos no críticos.
- **Marcado de tráfico:** El marcado de tráfico es el proceso de etiquetar los paquetes de datos con información de prioridad. El marcado de tráfico permite que los dispositivos de red identifiquen y traten los paquetes de datos de manera diferente, según su nivel de prioridad. Los protocolos de marcado de tráfico comunes incluyen el protocolo Differentiated Services (DiffServ) y el protocolo Integrated Services (IntServ).
- **Asignación de ancho de banda:** La asignación de ancho de banda es el proceso de asignar diferentes porcentajes de ancho de banda a diferentes tipos de tráfico en una red inalámbrica. Por ejemplo, el tráfico de voz y video puede tener un porcentaje más alto de ancho de banda asignado que el tráfico de datos no críticos. La asignación de ancho de banda permite garantizar que los datos críticos tengan un mejor rendimiento que los datos no críticos.

### 5.5. Análisis factores de influencia para la QoE en entornos WiFi

El concepto de calidad de experiencia (QoE) es una medida subjetiva que se utiliza para evaluar la calidad que experimenta un usuario cuando utiliza un servicio de telecomunicaciones. A diferencia de la QoS, que se basa en parámetros objetivos para optimizar el funcionamiento de la red, la QoE tiene en cuenta la satisfacción que un usuario recibe en términos del contenido y uso de las aplicaciones.

No hay métricas bien definidas de QoE cuando se compara con QoS, ya que depende de la percepción de cada usuario. Para ello, se tienen en cuenta factores que repercuten en la percepción de calidad. Ickin et al. [14] denominan a estos Factores de Influencia (FI) de la Calidad de la Experiencia, y los definen como "cualquier característica de un usuario, sistema, servicio,



aplicación o contexto cuyo estado o configuración actual puede influir en la Calidad de la Experiencia para el usuario". Varios trabajos han propuesto clasificaciones para estos FI. A continuación, se presentan algunos de ellos:

- Stankiewicz y Jajszczyk [15] clasifican los FI de QoE en tres grupos: factores de QoS, factores del Grado de Servicio (GoS), y factores de la Calidad de Resiliencia (QoR). Los autores destacan que la implementación de cada factor en el grupo correspondiente es crucial para lograr una alta QoE.
- Matulin y Š. Mrvelj [16] presentan una evaluación de los factores que afectan la QoE en cuatro niveles: la red central, la red de acceso, la calidad del hardware, y el usuario.
- Ickin et al. [14] organizan los FI de QoE en dos partes: aplicaciones y contextos. Recogen los factores de QoE en trece categorías basadas en aplicaciones móviles, y también mencionan: Diseño de la Interfaz de la Aplicación, Rendimiento de la Aplicación y Batería.
- Callet et al. [17] organizan los FI en tres categorías: Factores Humanos, Factores del Sistema y Factores de Contexto.
- Song et al. [18] resumen siete marcos dividiendo todos los parámetros en tres categorías: Usuario, Producto/Sistema/Servicio, y Contexto.

Teniendo en cuenta las clasificaciones propuestas por los autores mencionados, en [19] se establece un modelo jerárquico que clasifica los FI, tal y como se puede observar en la Ilustración 5.

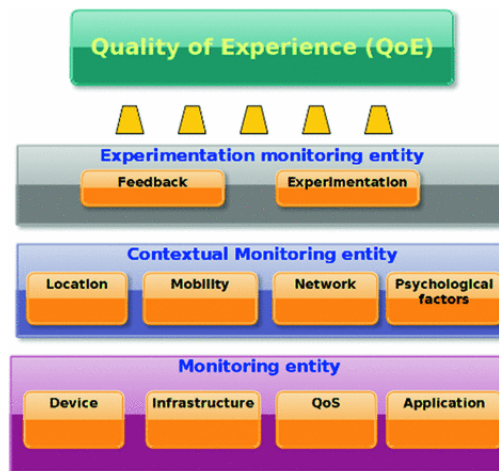


Ilustración 5: Arquitectura jerárquica que clasifica los FI

En el contexto de redes wifi, se puede abordar la QoE a través de tres FI: **Factores Humanos, Factores de Sistema y Factores de Contexto**. Cada uno de estos factores tiene una importancia particular en el caso de las redes WiFi, dada su naturaleza dinámica y diversa. Este enfoque abarca multitud de factores relevantes y permite considerar un amplio espectro de variables que pueden impactar en la QoE. A continuación, se procederá a realizar un análisis detallado y meticuloso de estos factores.



### 5.5.1. Factores de influencia humana

Los factores humanos son aquellos que tienen que ver con las características individuales de los usuarios y cómo estos influyen en su percepción de la calidad de experiencia en redes WiFi. Entre los principales factores humanos que pueden influir en la percepción de los usuarios se encuentran los siguientes:

- **Conocimientos técnicos:** La comprensión y habilidades técnicas de los usuarios en relación con las redes WiFi pueden influir en su percepción de la QoE. Por ejemplo, un usuario con conocimientos técnicos avanzados puede ser más crítico en la detección de problemas en la red, mientras que un usuario con menos conocimientos técnicos puede ser menos crítico.
- **Expectativas previas:** Las expectativas previas de los usuarios en cuanto a la calidad de la red WiFi pueden influir en su percepción de la QoE. Si los usuarios esperan una conexión rápida y estable, pueden ser más críticos si la red no cumple con esas expectativas.
- **Uso previsto:** El uso previsto de la red WiFi puede influir en la percepción de los usuarios. Por ejemplo, si un usuario utiliza la red para actividades que requieren una alta velocidad de conexión, como el streaming de video, puede ser más crítico si la red no cumple con esas necesidades.
- **Perfil de usuario:** El perfil de usuario también puede influir en la percepción de la de QoE. Hay diferentes perfiles de usuarios en redes WiFi, cada uno con diferentes necesidades y preferencias en cuanto a la QoE. Algunos de los perfiles más comunes son:
  - **Usuarios casuales:** Son aquellos que utilizan la red WiFi de forma ocasional y para tareas simples como navegación web, correo electrónico y redes sociales. Este perfil de usuario puede estar menos preocupado por la velocidad de conexión y la calidad de señal y puede ser más tolerante a problemas de calidad.
  - **Usuarios empresariales:** Este perfil de usuario suele utilizar la red WiFi para fines comerciales y empresariales. La calidad de la señal y la estabilidad de la conexión son muy importantes para ellos, ya que necesitan una conexión estable y confiable para trabajar en proyectos importantes.
  - **Usuarios de entretenimiento:** Este perfil de usuario utiliza la red WiFi principalmente para actividades de entretenimiento, como streaming de videos, juegos en línea y descarga de archivos. La velocidad de conexión y el ancho de banda son cruciales para ellos, ya que necesitan una conexión rápida y estable para disfrutar de sus actividades de entretenimiento sin interrupciones.
  - **Usuarios móviles:** Este perfil de usuario se conecta a la red WiFi desde dispositivos móviles como smartphones y tablets. La calidad de la señal y la velocidad de conexión son especialmente importantes para ellos, ya que necesitan una conexión confiable mientras están en movimiento.
  - **Usuarios técnicos:** Este perfil de usuario tiene un conocimiento técnico avanzado y puede estar más interesado en la configuración de la red y la optimización de la QoE. La calidad de la señal, la velocidad de conexión y la seguridad de la red son muy importantes para ellos.



Es importante tener en cuenta que estos factores humanos pueden interactuar entre sí para influir en la percepción de los usuarios. Por lo tanto, es importante tener en cuenta estas variables al momento de diseñar encuestas para evaluar la QoE en redes WiFi.

### 5.5.2. Factores de influencia del sistema

El factor de influencia del sistema es una de las categorías más estudiadas en relación con la QoE en redes WiFi. Este factor hace referencia a todas las características técnicas del sistema que influyen en la calidad percibida por el usuario, es decir, tiene una estrecha relación con la QoS de la red, ya que, la QoS influye directamente en la experiencia del usuario. A continuación, se analizarán los principales subfactores que influyen al sistema:

- **Relacionados con el contenido:** los aspectos técnicos relacionados con el contenido transmitido son fundamentales para garantizar una buena QoE en redes wifi. Entre ellos se incluyen la codificación de los datos, la resolución de la imagen o el sonido, la frecuencia de muestreo y la tasa de bits. Si estos factores no se ajustan adecuadamente, se pueden producir fallos en la transmisión, retrasos, cortes en la señal, etc.
- **Relacionados con la red:** la calidad de la red wifi es otro aspecto técnico clave para garantizar una buena QoE. En este sentido, la velocidad de conexión, el ancho de banda disponible, el retardo de la señal y la fluctuación de fase son algunos de los aspectos técnicos que influyen en la calidad percibida por el usuario.
- **Dispositivos:** los dispositivos utilizados para conectarse a la red wifi también tienen un impacto en la QoE. La resolución de la pantalla, el tamaño y la calidad de los altavoces o auriculares son algunos de los aspectos que influyen en la calidad percibida por el usuario.

Es relevante destacar que estos factores técnicos están estrechamente interrelacionados entre sí y, por lo tanto, cualquier problema técnico en uno de ellos puede afectar negativamente a los demás.

### 5.5.3. Factor de influencia del contexto

El factor de influencia del contexto en la QoE en redes wifi se refiere a cómo el entorno en el que se utiliza el servicio o sistema de WiFi puede afectar la calidad de la experiencia del usuario. Este factor se divide en varios subfactores:

- **Contexto físico:** el contexto físico se refiere a la ubicación y al espacio en el que se utiliza el servicio o sistema de wifi. Por ejemplo, si un usuario está en un área con una mala señal wifi debido a una mala cobertura, esto afectará su QoE. Además, la presencia de interferencias, ruido, obstáculos o una alta concentración de dispositivos que utilizan wifi también puede influir en la QoE del usuario.
- **Contexto temporal:** el contexto temporal se refiere a la hora del día, la frecuencia de uso y otros factores temporales que pueden influir en la QoE del usuario. Por ejemplo, si hay un mayor número de usuarios utilizando la red WiFi en ciertos momentos del día, la QoE puede disminuir debido a una sobrecarga de la red.
- **Contexto social:** el contexto social se refiere a las relaciones interpersonales que pueden influir en la QoE del usuario. Por ejemplo, si hay una gran cantidad de usuarios





que utilizan la misma red wifi en un espacio compartido, es posible que algunos usuarios experimenten una QoE deficiente debido a la competencia por el ancho de banda.

- **Contexto económico:** el contexto económico se refiere al poder adquisitivo de los usuarios y a cómo esto puede influir en la QoE. Por ejemplo, un usuario con una conexión de internet de alta velocidad puede tener una mejor QoE en comparación con un usuario que utiliza una conexión de baja velocidad.
- **Contexto de tarea:** el contexto de la tarea se refiere al tipo de tarea que se está realizando en el sistema o servicio de wifi y cómo esto puede influir en la QoE del usuario. Por ejemplo, si un usuario está realizando una tarea que requiere una conexión de alta velocidad, como descargar o cargar archivos grandes, es posible que experimente una QoE deficiente si la velocidad de la red es baja.
- **Contexto técnico:** El contexto técnico se refiere a la relación entre sistemas en la red wifi, como el enrutador, los dispositivos de acceso, la infraestructura de red y los protocolos utilizados. Estos factores pueden influir en la QoE del usuario, por ejemplo, una red WiFi con un router de baja calidad o una infraestructura de red obsoleta puede tener una mala QoS, lo que puede afectar negativamente la experiencia del usuario.

## 5.6. Análisis de metodologías de valoración

Para medir la QoE de los usuarios, se pueden utilizar diversas metodologías de valoración. Estas metodologías pueden ser objetivas o subjetivas. Los métodos objetivos se basan en medidas técnicas y estadísticas para evaluar la calidad de la red. Los métodos objetivos son especialmente útiles para identificar problemas en la red y permiten la monitorización constante de la calidad de la red. Sin embargo, Los métodos subjetivos se basan en la percepción del usuario y su experiencia de uso. Los métodos subjetivos se utilizan generalmente para evaluar la QoE percibida por el usuario final.

Este proyecto se enfocará exclusivamente en analizar los métodos subjetivos para medir la QoE. Esto se debe a que para este proyecto es fundamental obtener información directamente de los usuarios, comprender su perspectiva y tomar medidas efectivas para mejorar su experiencia en la red. A continuación, se procederá a explicar en detalle las distintas metodologías existentes para la valoración de la QoE de los usuarios mediante métodos subjetivos.

La recomendación ITU-T P.800, presenta el **Método de la Escala de Opinión Media (MOS)** [20]. Es uno de los métodos más utilizados para evaluar la calidad subjetiva de los servicios multimedia, como la voz y el video. Este método se basa en la idea de que la QoE puede ser medida mediante la recopilación de opiniones de los usuarios en una escala subjetiva.

La P.800 establece una escala de opinión subjetiva que se utiliza para medir la calidad de los servicios multimedia en función de la percepción de los usuarios. Esta escala es una escala ordinal de 1 a 5, donde 1 es la peor calidad y 5 es la mejor calidad. Cada valor de la escala se asigna a una descripción de calidad, que se utiliza para orientar a los usuarios a la hora de realizar su valoración subjetiva. La escala se divide en cinco categorías: pobre, justa, buena, muy buena y excelente, tal y como se puede ver en la Ilustración 6.



MOS	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Ilustración 6: MOS

El método MOS se basa en la idea de que, a pesar de que las opiniones de los usuarios pueden ser subjetivas, existe un cierto grado de consenso en cuanto a lo que se considera una experiencia de calidad. Por lo tanto, el MOS utiliza la opinión media de un grupo de usuarios para determinar la calidad de la experiencia del usuario.

El MOS se utiliza ampliamente en la industria de las telecomunicaciones para evaluar la calidad de los servicios multimedia y para comparar diferentes soluciones tecnológicas. La ventaja del MOS es que es una medida fácil de usar y de entender, y que se puede aplicar a diferentes tipos de servicios multimedia. Además, los resultados obtenidos con MOS son fáciles de comparar entre diferentes servicios y soluciones.

Sin embargo, el sistema MOS tiene algunas limitaciones. Una de las principales limitaciones es que no tiene en cuenta las características específicas del servicio multimedia evaluado. Además, los resultados del MOS pueden estar sesgados por factores como el tamaño de la muestra, la forma en que se selecciona a los usuarios para la evaluación, y la experiencia previa de los usuarios con el servicio multimedia evaluado.

Por otro lado, la recomendación ITU-T P.910, proporciona un enfoque más amplio y completo para evaluar la calidad de la experiencia del usuario de servicios multimedia. Se basa en la identificación de las características clave de la experiencia del usuario y en la recopilación de información a través de métodos cualitativos y cuantitativos, incluyendo entrevistas y encuestas.

Dentro de esta recomendación, se encuentran diversos métodos de evaluación subjetiva, entre ellos el método ACR (Absolute Category Rating), ACR-H (ACR with Hidden Reference) y DCR (Degradation Category Rating) y el método de pares.

El **método ACR** [21] es el método más utilizado en la evaluación subjetiva de la calidad. Este método implica que los usuarios califiquen la calidad de un servicio en una escala de categorías absolutas, como excelente, bueno, regular, pobre o malo. Los usuarios deben elegir una sola categoría que mejor describa la calidad percibida del servicio.

En este método, se presentan secuencias a los espectadores uno a uno y se les pide que califiquen la calidad de cada secuencia en una escala de categoría. Después de cada presentación, los espectadores evalúan la calidad de la secuencia mostrada en función de la escala de categoría. La presentación del estímulo en el método ACR puede ser constante o variable.

Una mejora en el método ACR es el método **ACR con referencia oculta (ACR-HR)** [21]. En este método, se incluye una versión de referencia de cada secuencia de prueba que se



muestra como cualquier otro estímulo de prueba, pero de forma oculta. Durante el análisis de los datos, se calcula una puntuación diferencial de opinión media (DMOS) de calidad entre cada secuencia de prueba y su correspondiente (oculta) referencia. Este procedimiento se conoce como "referencia oculta". El cálculo de la DMOS tiene en cuenta la calidad de la secuencia de prueba en relación con la calidad de la referencia oculta.

El método ACR-HR tiene la ventaja de eliminar el sesgo del espectador hacia la referencia al calcular la DMOS, lo que ayuda a obtener una medida más precisa de la calidad de la secuencia de prueba. El método ACR-HR es útil en experimentos a gran escala, siempre y cuando todas las referencias sean de al menos "buena" calidad.

Por otro lado, el método de **Categoría de Degradación (DCR)** [21] es una técnica subjetiva utilizada para evaluar la calidad de la imagen en sistemas de video. Es un método que implica la presentación de dos secuencias de video, siendo la primera una secuencia de referencia y la segunda una secuencia procesada por el sistema bajo evaluación. El evaluador debe determinar el grado de degradación de la segunda secuencia en relación con la primera.

El método DCR utiliza una escala de cinco niveles para calificar el grado de degradación de la segunda secuencia: 5 imperceptible, 4 perceptible pero no molesto, 3 ligeramente molesto, 2 molesto, 1 muy molesto. El evaluador debe calificar cada par de secuencias presentadas durante el experimento en esta escala.

La ventaja del método DCR es que permite evaluar la fidelidad de la transmisión con respecto a la señal de origen. Por lo tanto, es adecuado para evaluar la calidad de imagen de sistemas de alta calidad en videoconferencia.

El método DCR puede presentar algunas limitaciones, por ejemplo, puede ser difícil de aplicar en sistemas que tienen múltiples fuentes de degradación simultáneas o que presentan degradaciones no uniformes. Además, el método DCR no tiene la misma discriminación que otros métodos, como el ACR-HR, ya que solo se evalúa la degradación en relación con una referencia y no se realiza una evaluación comparativa directa.

Por último, el método de **comparación por pares (PC)** implica que las secuencias de prueba se presentan en pares, consistiendo en que la misma secuencia se presente primero a través de un sistema de prueba y luego a través de otro sistema. Los sistemas de prueba (A, B, C, etc.) se combinan generalmente en todas las posibles combinaciones  $n(n - 1)$  como AB, BA, CA, etc. Por lo tanto, todos los pares de secuencias se deben mostrar en ambos los órdenes posibles (por ejemplo, AB, BA). Este método puede ser más difícil de implementar en comparación con otros métodos de evaluación, ya que requiere una comparación directa de pares de secuencias y puede requerir una mayor cantidad de tiempo de evaluación por parte de los usuarios.

Es importante saber cuándo implementar cada uno de estos métodos para obtener el resultado más preciso posible. Esta recomendación comenta que el método DCR es recomendado cuando se desea evaluar la fidelidad de la transmisión con respecto a la señal de origen, especialmente en sistemas de alta calidad. Esta metodología ha sido ampliamente utilizada en la evaluación de calidad de imágenes de televisión en el contexto de la



videotelefonía y videoconferencia. El uso de la escala DCR es particularmente valioso cuando se requiere discriminar entre la percepción de una degradación imperceptible o perceptible.

Por otro lado, el método ACR es fácil y rápido de implementar, y la presentación de estímulos es similar a la de uso común en los sistemas multimedia. Esto lo hace adecuado para realizar pruebas de calificación.

ACR-HR tiene todas las ventajas de ACR en cuanto a la presentación y velocidad, y además tiene la capacidad de eliminar el impacto perceptual del video de referencia en las puntuaciones subjetivas. Esto reduce el impacto del sesgo del escenario (por ejemplo, los espectadores que les gustan o no les gustan un video de referencia), la calidad del video de referencia (por ejemplo, pequeñas diferencias en la calidad de la cámara) y el monitor (por ejemplo, calidad profesional versus grado de consumidor) en las puntuaciones finales. ACR-HR es adecuado para realizar experimentos con un gran número de observadores, siempre y cuando todos los videos de referencia sean de al menos "buena" calidad. Sin embargo, ACR-HR puede ser insensible a algunas degradaciones que son fácilmente detectadas por métodos diferenciales directos, como DCR.

Por último, el método PC se destaca por su alta capacidad discriminativa, lo que es particularmente valioso cuando varios elementos de prueba son de calidad similar. Sin embargo, este método tiende a ser más largo cuando se desean evaluar un gran número de elementos de prueba.



## 6. ANÁLISIS DE ALTERNATIVAS

Después de haber realizado una exhaustiva revisión de las tecnologías existentes en el estado del arte, es importante llevar a cabo un análisis de alternativas para definir la selección de la solución en cuanto a: **la identificación del parámetro QoS, la identificación del algoritmo de ML, los factores de influencia relevantes a tener en cuenta, la metodología de valoración a realizar y el entorno de desarrollo y el lenguaje de programación para el desarrollo del modelo de ML.**

Para ello, teniendo en cuenta el contexto del proyecto (apartado 2), se puede determinar cuáles de las opciones revisadas en el estado del arte son comparables para su realización. Después, se podrá llevar a cabo una evaluación rigurosa de las diferentes alternativas y seleccionar aquellas que mejor se adapten a las necesidades y objetivos del proyecto. En este sentido, se establecen una serie de criterios de selección que se adapten de manera efectiva al objetivo en cuestión. Cada criterio deberá tener un porcentaje asociado en función de su importancia relativa. Para evaluar cada alternativa, se calificará cada criterio del 1 al 5, y se seleccionará la opción u opciones que obtengan la mayor puntuación global en base a los criterios establecidos.

### 6.1. Identificación del parámetro QoS

En la sección 5.4 del estado del arte, se ha llevado a cabo un análisis exhaustivo de los diversos parámetros objetivos que influyen en la calidad de servicio de las redes WiFi. Sin embargo, para este caso de estudio en particular, se ha decidido seleccionar un único parámetro objetivo que se utilizará para entrenar el modelo y establecer una correlación con la QoE. Esta decisión se ha tomado en base a varias razones:

- **Simplicidad del Modelo:** Comenzar con un único parámetro permite simplificar el modelo y facilitar su interpretación. Es más sencillo entender y comunicar los efectos de un único parámetro en la QoE que lidiar con múltiples parámetros y las posibles interacciones entre ellos.
- **Enfoque en el Parámetro más Importante:** Algunos parámetros pueden tener un impacto más significativo en la QoE que otros. El foco en un único parámetro permite centrarse en aquel que se considera más influyente para el caso de estudio.

Para lograr esto, hay que realizar un análisis de parámetros objetivos que influyan en la red wifi y que sean óptimas para realizar el modelo de correlación QoS/QoE. En el apartado 5.4, se han analizado diversos parámetros relevantes en relación con la QoS, sin embargo, este análisis se centrará en el throughput y el SNR (relación señal-ruido). La selección de estos parámetros se debe a dos razones: primero, son características que están disponibles en el conjunto de datos existente; segundo, poseen una relevancia indiscutible en el estudio de la QoS.

El throughput es esencial porque representa la cantidad de datos que pueden ser transmitidos desde un punto a otro en un periodo de tiempo determinado. Su importancia radica en el hecho de que un mayor throughput a menudo se traduce en una red más rápida y eficaz, lo que contribuye a una mejor QoE.



La relación señal-ruido (SNR), por otro lado, es un parámetro que puede ofrecer información valiosa sobre la integridad de la señal de la red. La SNR puede ser particularmente útil para entender cómo los obstáculos físicos, como paredes y mobiliario, o la densidad de usuarios en ciertos espacios, como aulas o auditorios, pueden afectar la calidad de la conexión WiFi. Un SNR bajo puede indicar una señal débil o mucha interferencia, lo que se traduce en una peor experiencia para el usuario.

Con el objetivo de elegir el parámetro más relevante para realizar la comparación entre los dos entornos universitarios, se aplicarán varios criterios y en base a ellos, se seleccionará el parámetro que mejor se ajuste a la naturaleza del proyecto.

#### 6.1.1. Throughput

Como se ha explicado ya en el apartado 5.4.2, el throughput hace referencia a la cantidad de datos que se pueden transmitir o recibir a través de una red en un determinado período de tiempo. Es una medida de la velocidad de transferencia de datos y se expresa típicamente en bits por segundo (bps) o en unidades más grandes como kilobits por segundo (Kbps).

En este caso de estudio, el conjunto de datos con el que se va a trabajar contiene el throughput medio de una sesión de un usuario específico. Por lo tanto, el throughput se presenta como un parámetro importante a ser correlacionado con la experiencia de los usuarios. Esto se debe a que el throughput proporciona una indicación general de la velocidad de la red WiFi, que a su vez influye en la calidad de la red experimentada por los usuarios durante una sesión determinada.

Un alto throughput implica una mayor velocidad de transferencia de datos, lo que generalmente se asocia con una mejor calidad de la red WiFi y una experiencia de usuario más satisfactoria. Por otro lado, un bajo throughput indica una velocidad de transferencia de datos más lenta, lo que puede resultar en una menor calidad de la red y una experiencia de usuario menos óptima.

Dado que el throughput es una medida directa de la velocidad de la red WiFi, tiene un impacto significativo en la calidad percibida por los usuarios. Por lo tanto, correlacionar el throughput medio de una sesión con la experiencia de los usuarios puede proporcionar información valiosa sobre cómo la velocidad de la red influye en su satisfacción y percepción de la calidad del servicio.

#### 6.1.2. SNR

La SNR, como analizado en el apartado 5.4.6, representa la relación entre la potencia de la señal recibida y el nivel de ruido presente en el entorno de la red WiFi. Proporciona información sobre la calidad de la señal transmitida y recibida.

La SNR alta indica que la señal recibida es fuerte en comparación con el ruido, lo que se traduce en una mejor calidad de la señal y menos interferencias. Esto conduce a una mayor



capacidad de transferencia de datos y una menor probabilidad de errores en la comunicación. En consecuencia, una alta SNR está asociada con una mejor calidad de la red WiFi.

Por otro lado, una SNR baja indica que la señal recibida es débil en comparación con el ruido. Esto puede resultar en una calidad de señal deficiente, pérdida de paquetes de datos y una mayor probabilidad de errores de transmisión. Una baja SNR puede afectar negativamente la calidad de la red WiFi y la experiencia del usuario al reducir la velocidad de conexión y generar interrupciones en la comunicación.

Al correlacionar la SNR con la experiencia de los usuarios, se obtiene información valiosa sobre cómo la calidad de la señal transmitida y recibida influye en su satisfacción y percepción de la calidad del servicio.

### 6.1.3. Criterios de selección

- 1. Importancia teórica (35%):** La importancia teórica es un criterio crucial ya que respalda la relación entre el parámetro y la QoE, por lo que se asigna un 25% a este criterio.
- 2. Disponibilidad de datos (25%):** La disponibilidad de datos es un factor clave ya que se necesita contar con un conjunto de datos completo y confiable para realizar la correlación, por tanto, se le asigna un 20%.
- 3. Significado práctico (25%)** El significado práctico considera la utilidad y aplicabilidad de los resultados obtenidos al correlacionar cada parámetro con la QoE.
- 4. Facilidad de medición (15%):** La facilidad de medición se refiere a la facilidad y precisión con la que se puede medir cada parámetro en el entorno universitario.

Criterios	Throughput	SNR	Peso (%)
Importancia teórica	5	4	35
Disponibilidad de los datos	5	5	25
Significado práctico	5	4	25
Facilidad de medición	4	4	15
<b>TOTAL</b>	<b>4,85</b>	<b>4,25</b>	

Tabla 1: Matriz de alternativas de los parámetros QoS

Observando la Tabla 1, el throughput ha obtenido una puntuación total más alta. Esto se debe a su importancia teórica, ya que mide directamente la velocidad de transferencia de datos en la red WiFi, lo cual está relacionado con la calidad de servicio. Además, el throughput es más relevante en términos prácticos para representar la QoS, ya que proporciona una indicación clara de la capacidad de la red para entregar datos de manera eficiente. También tiene un significado práctico más directo, ya que una correlación significativa con la QoE indica que una mayor velocidad de transferencia de datos se asocia con una mejor experiencia del usuario.

En consecuencia, el throughput se selecciona como el parámetro más adecuado para correlacionar con la QoE en este estudio. Su medición directa de la velocidad de transferencia de datos y su significado práctico en términos de la QoE lo convierten en la mejor opción en comparación con el SNR.



## 6.2. Identificación del algoritmo de ML

En este proyecto, se busca desarrollar un modelo de correlación entre la calidad experimentada por los usuarios (QoE) y la calidad de servicio objetiva (QoS) en redes WiFi. Para lograr este objetivo, es necesario aplicar técnicas de ML que permitan analizar grandes cantidades de datos y establecer relaciones complejas entre ellos.

En particular, se han considerado algoritmos supervisados de ML ya que se cuenta con un conjunto de datos previamente etiquetados (encuestas que evalúan la satisfacción de los usuarios con el servicio) y se desea entrenar un modelo para que pueda correlacionar la calidad experimentada por los usuarios a partir de cierto indicador objetivo de calidad de servicio (KPI). El KPI seleccionado para este estudio es el throughput, como se ha establecido anteriormente. Al tener datos etiquetados de las encuestas de satisfacción, se utilizan técnicas de ML supervisado para entrenar un modelo capaz de predecir la calidad experimentada por los usuarios a partir del valor del throughput.

En este proyecto se ha optado por analizar dos tipos de algoritmos supervisados (concretamente de regresión): **la regresión lineal y la regresión logística**. La elección de estos dos se debe a varias razones:

- **Regresión lineal:** Este algoritmo es uno de los más básicos y fundamentales en el aprendizaje supervisado. Se ha elegido por su simplicidad y eficiencia, lo que lo convierte en un buen punto de partida para entender y modelar la correlación entre los datos de entrada (las medidas de QoS) y los de salida (la QoE percibida por los usuarios).
- **Regresión logística:** Aunque la regresión logística se utiliza comúnmente para problemas de clasificación binaria, también puede ser aplicada a problemas de regresión multiclase al considerar la QoE como un valor discreto dentro de un conjunto de clases (por ejemplo, mala, aceptable, buena). Se ha escogido la regresión logística por su capacidad para manejar este tipo de problemas y porque puede proporcionar probabilidades asociadas a las predicciones, lo que puede ser útil para entender la confianza del modelo en sus predicciones.

Además, se explorarán los tres tipos de algoritmos principales supervisados dentro de las redes neuronales: **Redes neuronales feedforward Redes Neuronales Convolucionales y Redes Neuronales Recurrentes**. Tras realizar un análisis detallado de los diferentes algoritmos, se seleccionarán aquellos que resulten más interesantes y adecuados para el proyecto en cuestión, en base a diversos criterios establecidos. Esto implicará elegir el algoritmo de regresión óptimo entre la regresión lineal y la regresión logística, así como el tipo de red neuronal (feedforward, convolucional o recurrente) que mejor se ajuste a los datos y objetivos del estudio.

Se desarrollarán dos modelos distintos: uno basado en el algoritmo de regresión seleccionado, que proporciona una aproximación más sencilla y fácilmente interpretable de la relación entre el throughput y la experiencia de usuario; y otro basado en una red neuronal seleccionada, que permite una modelización más profunda y la captura de relaciones más complejas y no lineales.





### 6.2.1. Regresión lineal

La regresión lineal [22] es un modelo de regresión en el que se establece una relación lineal entre una única variable independiente (predictor) y una variable dependiente (respuesta). Se utiliza cuando queremos predecir o evaluar la influencia de una sola variable en la variable de respuesta. La forma matemática de un modelo de regresión lineal simple se puede expresar como:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Donde:

- $Y_i$  es el valor de la variable dependiente en la observación  $i$ .
- $\beta_0$  es el término de intercepción, que representa el valor de  $Y$  cuando la variable independiente  $X_i$  es igual a cero.
- $\beta_1$  es el coeficiente de regresión, que indica cómo influye la variable independiente  $X_i$  en la variable dependiente  $Y$ .
- $X_i$  es el valor de la variable independiente en la observación  $i$ .
- $\varepsilon_i$  es el término de error, que representa la variabilidad no explicada por el modelo.

Durante el entrenamiento del modelo de regresión lineal simple, el algoritmo ajusta los coeficientes de regresión ( $\beta_0$  y  $\beta_1$ ) para minimizar el error cuadrático medio entre las predicciones del modelo y los valores reales de  $Y$  en el conjunto de entrenamiento.

Una vez entrenado el modelo, se puede utilizar para predecir el valor de  $Y$  en función de nuevos valores de  $X_i$ . La regresión lineal simple es un modelo sencillo y fácil de interpretar, ya que proporciona una estimación directa de la relación lineal entre la variable independiente y la variable dependiente. El coeficiente de regresión  $\beta_1$  indica cómo cambia  $Y$  en promedio cuando  $X_i$  aumenta en una unidad.

Sin embargo, es importante tener en cuenta que la regresión lineal simple asume una relación lineal y aditiva entre la variable independiente y la variable dependiente. Esto implica que la relación entre las variables es constante en todo el rango de valores. Si la relación no es lineal o si hay otras variables que influyen en la variable dependiente, el modelo de regresión lineal simple puede no ser apropiado.

### 6.2.2. Regresión Logística

La regresión logística [23] es un modelo estadístico utilizado en el ML para analizar la relación entre una variable dependiente binaria ( $Y$ ) y un conjunto de variables independientes continuas o categóricas ( $X_1, X_2, X_3$ , etc.). El objetivo de la regresión logística es estimar la probabilidad de que  $Y$  sea igual a 1 en función de los valores de las variables independientes.

En la regresión logística, se utiliza una función logística (también conocida como sigmoide) para modelar la relación entre las variables independientes y la probabilidad de que



Y sea igual a 1. La función logística transforma la suma ponderada de las variables independientes mediante los coeficientes de regresión en un valor entre 0 y 1.

Matemáticamente, el modelo de regresión logística se expresa como:

$$p(Y = 1) = 1 / (1 + \exp(-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)))$$

Donde  $p(Y=1)$  es la probabilidad de que, Y sea igual a 1,  $\beta_0$  es el término de intercepción y  $\beta_1$  a  $\beta_n$  son los coeficientes de regresión para cada variable independiente.

Durante el entrenamiento del modelo de regresión logística, se ajustan los valores de los coeficientes de regresión ( $\beta_1$  a  $\beta_n$ ) para maximizar la verosimilitud de los datos de entrenamiento. Este proceso implica encontrar los valores de los coeficientes que hacen que el modelo se ajuste mejor a los datos observados.

Una vez entrenado el modelo, se puede utilizar para predecir la probabilidad de que la variable dependiente sea igual a 1 en función de nuevos valores de las variables independientes. Si la probabilidad estimada es mayor que un umbral predeterminado (por ejemplo, 0.5), se clasifica la instancia como perteneciente a la clase 1; de lo contrario, se clasifica como perteneciente a la clase 0.

La regresión logística es adecuada para variables dependientes binarias y se utiliza comúnmente en problemas de clasificación binaria. Sin embargo, también puede ser extendida a problemas de clasificación multiclase utilizando técnicas como la regresión logística multinomial o la regresión logística ordinal.

### 6.2.3. Criterios de selección

Para la evaluación de los algoritmos se han utilizado los siguientes criterios:

1. **Naturaleza de los datos (25%):** La naturaleza de los datos hace referencia al tipo de datos y al nivel de complejidad que presentan. Es crucial seleccionar un modelo adecuado que permita procesar estos datos de manera efectiva y obtener resultados precisos.
2. **Tamaño del conjunto de datos (20%):** se refiere a la cantidad de datos que se utilizan para crear un modelo de ML. Es fundamental elegir un algoritmo que se adapte al tamaño del conjunto de datos en cuestión, ya que el rendimiento y la precisión del modelo pueden verse afectados significativamente si se utiliza un algoritmo inadecuado.
3. **Facilidad de implementación (20%):** se refiere a la facilidad con la que se puede implementar y utilizar el algoritmo. Esto incluye la disponibilidad de bibliotecas de código abierto, la documentación y la complejidad del algoritmo en sí mismo.
4. **Precisión (20%):** se refiere a la precisión del algoritmo en la tarea de predicción de la QoE a partir de la QoS.
5. **Escalabilidad (15%):** se refiere a la capacidad del algoritmo para manejar grandes cantidades de datos.



Criterios	Regresión lineal	Regresión logística	Peso (%)
Naturaleza de los datos	4	3	25
Tamaño del conjunto de datos	3	3	20
Facilidad de implementación	5	4	20
Escalabilidad	3	3	20
Precisión	4	3	15
<b>TOTAL</b>	<b>3,75</b>	<b>3,2</b>	

Tabla 2: Matriz de alternativas de algoritmos de regresión

Teniendo en cuenta estos resultados, la regresión lineal emerge como la opción preferida debido a su mejor puntuación total y su ventaja en la naturaleza de los datos y la facilidad de implementación. Si bien la regresión lineal se selecciona como el método inicial, es importante resaltar que no se debe asumir de manera predefinida una relación lineal entre el throughput y la experiencia de usuario. Dicha relación podría no ser estrictamente lineal, y la suposición de una relación lineal puede ser una simplificación de la realidad.

En este sentido, aunque la regresión lineal proporciona un primer enfoque para modelar la relación entre el throughput y la experiencia del usuario, se plantea (como ya se ha estipulado anteriormente) también la realización de un modelo basado en redes neuronales. Este tipo de algoritmo puede capturar de manera más precisa la relación potencialmente compleja entre estas variables. Mediante la comparación de estos dos algoritmos, se determina la verdadera naturaleza de la relación entre las variables.

#### 6.2.4. Redes neuronales feedforward

Las redes neuronales feedforward (FNN) [24][25] son un tipo de red neuronal artificial que se utiliza en el aprendizaje supervisado. Este tipo de red está compuesta por una serie de capas de neuronas, en la que cada neurona está conectada con las neuronas de la capa siguiente, y todas las neuronas de la misma capa están conectadas. Se las denomina feed-forward ya que la información no puede ir más que desde las entradas hasta las salidas, sin volver atrás.

Cada capa consta de un conjunto de nodos o neuronas, que reciben entradas de la capa anterior y generan salidas para la capa siguiente. Los pesos entre los nodos se ajustan durante el entrenamiento de la red neuronal para minimizar el error en la predicción de la salida.

Una FNN se caracteriza por ser un conjunto de neuronas que reciben información multivariable, la procesan y dan una respuesta que puede ser multivariable también. En la arquitectura feedforward la topología del arreglo de neuronas y sus interconexiones hace fluir la información de forma unidireccional para que nunca pueda pasar más de una vez a través de una neurona antes de generarse la respuesta de salida. Tal como muestra la.

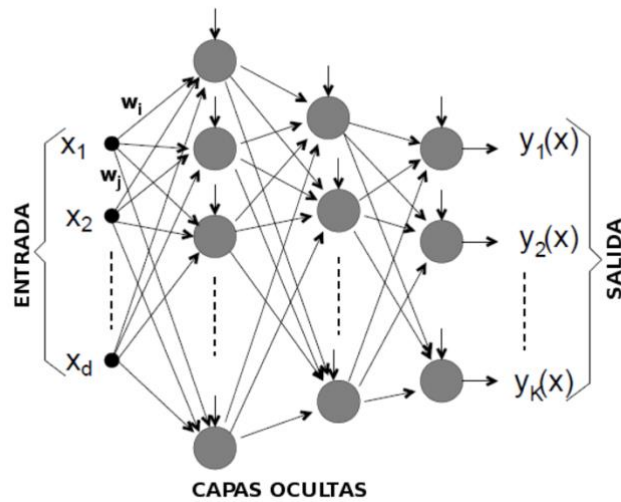


Ilustración 7: Diagrama de la estructura típica de una RNAf

A muy grandes rasgos, el trabajo de la red es ingresar los datos a través de las neuronas de la capa de entrada, y al pasar la información a la siguiente capa, llamada oculta, cada una de las neuronas receptoras recibe la suma ponderada de todas las entradas conectadas a ella. Pues cada conexión entre neuronas representa un peso de conexión. Lo que matemáticamente se puede resumir como:

$$\sum_j w_{ij}x_j$$

Donde  $w_{ij}$  son los pesos de cada conexión de las  $x_j$  neuronas a la neurona receptora. Pero, al igual que sucede con las neuronas biológicas reales, a menos que el estímulo tenga cierta magnitud, la información se procesa, de lo contrario se ignora. En las FNN se logra mediante un valor umbral  $m_i$ , y a menos que la suma ponderada supere un valor dado, se considera la salida efectiva de la información procesada. Nuevamente, se puede expresar como:

$$\sum_j w_{ij}x_j - m_i$$

El proceso de aprendizaje de una FNN puede ser del tipo conocido como supervisado, lo que significa que se utiliza un conjunto de valores objetivo como patrón de comparación para la salida que produce la red neuronal. Durante este proceso, se calcula la diferencia entre la salida real y la salida deseada, y se utiliza esta diferencia para ajustar los valores de peso en las conexiones de las neuronas de la red. El objetivo es minimizar el error en la siguiente iteración. Este proceso se repite hasta que se alcance un nivel de precisión aceptable en la red neuronal.

Diversos algoritmos se utilizan para el aprendizaje de las FNN. Sin embargo, el más popular es el de propagación reversa (backpropagation algorithm), y se fundamenta en una estrategia simple para reducir el error cuadrático medio, similar al método generalizado de regresión.



Utilizando este tipo de algoritmo, se pueden aprender patrones no lineales complejos en los datos de QoS y QoE, lo que podría llevar a una mejor correlación entre las dos variables que otros modelos más simples.

A pesar de las ventajas que ofrecen las FNN supervisadas hay algunas desventajas que se deben tener en cuenta. En primer lugar, estas redes requieren un gran conjunto de datos de entrenamiento para producir resultados precisos. Si no hay suficientes datos de QoS y QoE disponibles para el entrenamiento de la red, esto podría ser un problema.

#### 6.2.5. Redes neuronales convolucionales

Una red neuronal convolucional (CNN) [26] es un tipo de red neuronal que se utiliza comúnmente en aplicaciones de visión por computadora y procesamiento de imágenes. En lugar de utilizar capas completamente conectadas como una FNN, las CNN utilizan capas convolucionales que aplican filtros a la entrada para extraer características relevantes.

En la primera capa convolucional, los filtros se aplican a la imagen de entrada para detectar características simples, como bordes y curvas. En las capas convolucionales posteriores, los filtros se utilizan para detectar características más complejas, como texturas y patrones.

Después de las capas convolucionales, la salida se alimenta a través de capas completamente conectadas para realizar la clasificación final. En general, las CNN son muy efectivas para tareas de clasificación de imágenes, como identificar objetos en una imagen.

Al igual que con las FNN, las CNN también se pueden entrenar mediante el algoritmo de propagación hacia atrás. La diferencia es que, en las CNN, el proceso de entrenamiento implica ajustar los pesos de los filtros de convolución en lugar de los pesos de las neuronas completamente conectadas.

Este tipo de red neuronal es especialmente eficiente en el procesamiento de imágenes grandes, ya que las capas convolucionales comparten los mismos pesos en toda la imagen de entrada. Esto permite que las CNN procesen imágenes más grandes de manera más eficiente.

En cuanto a la utilidad para este proyecto, el uso de una CNN puede tener sentido si se trabaja con datos de imagen relacionados con la calidad de la red. Asimismo, se requiere una gran cantidad de datos de entrenamiento para producir resultados precisos y pueden ser más complejas de entrenar que las FNN, ya que requieren ajustar los pesos de los filtros de convolución en lugar de los pesos de las neuronas completamente conectadas.

#### 6.2.6. Redes neuronales recurrentes

Una red neuronal recurrente (RNN) [27] es un tipo de red neuronal que se utiliza en aplicaciones de procesamiento de lenguaje natural y series de tiempo. A diferencia de las FNN y las CNN, las RNN utilizan la salida anterior como entrada para la siguiente iteración, lo que permite que la red tenga memoria y pueda capturar patrones en secuencias de datos.

Las RNN se componen de células de memoria que mantienen una memoria interna y que pueden aceptar una entrada y producir una salida. En cada paso de tiempo, la célula de



memoria procesa la entrada actual y la memoria anterior para producir una nueva salida y actualizar la memoria interna.

Por lo tanto, la decisión del paso temporal  $t-1$  afecta a la decisión tomada en el paso temporal  $t$ , es por eso por lo que se dice que las redes neuronales recurrentes tienen memoria. Por ejemplo, si se tuviera que clasificar que tipo de evento está pasando en cada momento de una película, no está claro como una red neuronal tradicional podría usar su razonamiento sobre eventos anteriores en la película para informar eventos posteriores. Mientras que las RNN pueden hacerlo gracias a que tienen bucles; permitiendo que la información persista.

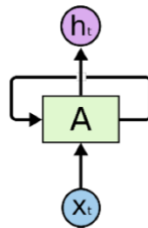


Ilustración 8: Red neuronal recurrente

Una ventaja de las RNN es que son altamente efectivas para el procesamiento de secuencias y, por lo tanto, pueden ser una buena opción para modelar la correlación de la QoS y la QoE en el tiempo. Sin embargo, las RNN también pueden ser propensas al sobreajuste y pueden requerir una gran cantidad de datos de entrenamiento para producir resultados precisos. Asimismo, las RNN pueden ser más complejas de entrenar y pueden requerir más tiempo de entrenamiento y recursos computacionales.

#### 6.2.7. Criterios de selección

Para la evaluación de los algoritmos se han utilizado los siguientes criterios:

6. **Naturaleza de los datos (25%):** La naturaleza de los datos hace referencia al tipo de datos y al nivel de complejidad que presentan. Es crucial seleccionar un modelo adecuado que permita procesar estos datos de manera efectiva y obtener resultados precisos.
7. **Tamaño del conjunto de datos (20%):** se refiere a la cantidad de datos que se utilizan para crear un modelo de ML. Es fundamental elegir un algoritmo que se adapte al tamaño del conjunto de datos en cuestión, ya que el rendimiento y la precisión del modelo pueden verse afectados significativamente si se utiliza un algoritmo inadecuado.
8. **Facilidad de implementación (20%):** se refiere a la facilidad con la que se puede implementar y utilizar el algoritmo. Esto incluye la disponibilidad de bibliotecas de código abierto, la documentación y la complejidad del algoritmo en sí mismo.
9. **Precisión (20%):** se refiere a la precisión del algoritmo en la tarea de predicción de la QoE a partir de la QoS.
10. **Escalabilidad (15%):** se refiere a la capacidad del algoritmo para manejar grandes cantidades de datos.



Criterios	Redes neuronales feedforward	Redes neuronales convolucionales	Redes neuronales recurrentes	Peso (%)
Naturaleza de los datos	4	3	2	25
Tamaño del conjunto de datos	4	4	4	20
Facilidad de implementación	4	4	3	20
Escalabilidad	4	4	5	20
Precisión	4	4	4	15
<b>TOTAL</b>	<b>4</b>	<b>3,75</b>	<b>3,15</b>	

Tabla 3: Matriz de alternativas de los algoritmos de ML

La red neuronal feedforward ha obtenido la mayor puntuación total, dado que las FNN son útiles para datos numéricos y categóricos, lo que las convierte en una mejor opción que las redes neuronales convolucionales, cuyo propósito suele ser el procesamiento de imágenes, o las redes neuronales recurrentes, que se utilizan principalmente para datos secuenciales.

Asimismo, esta red neuronal ha obtenido mayor puntuación en facilidad de implementación y tamaño del conjunto de datos. Esto se debe a que las FNN son más simples y fáciles de implementar que las otras dos opciones, y son efectivas en problemas de clasificación y regresión en conjuntos de datos medianos, como se ha mencionado anteriormente.

### 6.3. Factores de influencia

Para desarrollar un modelo de correlación QoS/QoE preciso y eficiente, es necesario realizar una clasificación de los aspectos de influencia de cara a la gestión de la QoE y se deben identificar los indicadores de calidad (KQI) que pueden ser más interesantes para luego poder realizar una encuesta óptima y efectiva. De esta forma, se podrán obtener datos más precisos y útiles para el desarrollo de un modelo de correlación más riguroso.

Como se ha mencionado previamente en la sección 5.5, para evaluar la QoE, es necesario tener en cuenta tres factores generales: **factores humanos, factores del sistema y factores de contexto**. Cada uno de estos factores contiene una variedad de subfactores que son relevantes para diferentes perfiles de usuarios, sistemas y situaciones. Es esencial tener en cuenta que la selección de los factores relevantes para la evaluación de la QoE debe estar basada en la naturaleza y las necesidades del caso de estudio. Por lo tanto, en este análisis de alternativas, se llevará a cabo una selección cuidadosa de los factores más relevantes para este caso en particular. Una vez que se hayan identificado los factores adecuados, se realizará una evaluación rigurosa en base a diversos criterios para garantizar que se seleccionen los más apropiados y que la evaluación de la QoE sea lo más precisa posible.



Para llevar a cabo este análisis, resulta imprescindible centralizarlo en el entorno inalámbrico IEEE 802.11 (WiFi), además de tener en cuenta que la encuesta se va a lanzar en dos entornos universitarios muy distintos: facultad de ciencias y facultad de educación.

### 6.3.1. Factores humanos

Para comenzar, se van a analizar los diversos factores humanos relevantes que pueden influir en la experiencia de calidad de los usuarios, pero en este caso, se centrará el análisis en el estudio de este proyecto en particular.

#### 6.3.1.1. Perfil de usuario

El perfil de usuario puede incluir diferentes variables, como la edad, el sexo, el área de estudio, el género, entre otros.

En este caso de estudio, es importante tener en cuenta el perfil de usuario, ya que las diferentes tipologías de usuarios pueden tener necesidades y expectativas diferentes con respecto al servicio de red inalámbrica. Por ejemplo, los estudiantes de ciencias pueden tener necesidades más exigentes en cuanto a la velocidad de conexión y la disponibilidad de ancho de banda debido a que necesitan acceder a recursos pesados como software de simulación, bases de datos o bibliografías digitales. En cambio, los estudiantes de humanidades pueden tener una menor exigencia en términos de ancho de banda, pero pueden ser más sensibles a la disponibilidad de la red y a la calidad de la señal debido a que pueden necesitar acceder a la red en zonas con menor cobertura como bibliotecas, salas de lectura, entre otros.

Por lo tanto, es necesario identificar las diferentes tipologías de usuarios y sus necesidades específicas para poder gestionar de forma eficiente la QoE. Esto puede permitir tomar decisiones informadas y adaptar el servicio a las necesidades y expectativas de los usuarios, lo que puede mejorar la percepción de calidad del servicio y, por lo tanto, la QoE.

#### 6.3.1.2. Conocimientos técnicos

El conocimiento técnico de los usuarios es otro de los factores de influencia clave que puede afectar a la QoE. El conocimiento técnico de los usuarios se refiere a la habilidad que tienen los usuarios para utilizar dispositivos tecnológicos y comprender el funcionamiento de las redes inalámbricas.

En este caso, es importante tener en cuenta el conocimiento técnico de los usuarios ya que esto puede influir en la percepción de la QoS de red inalámbrica. Por ejemplo, si los usuarios tienen un bajo conocimiento técnico, pueden experimentar problemas al conectarse a la red, configurar la conexión, solucionar problemas técnicos, entre otros. Esto puede generar una percepción negativa de la calidad del servicio y, por lo tanto, reducir la QoE.

Por ello, es necesario identificar el nivel de conocimiento técnico de los usuarios y ofrecer soluciones y recursos adecuados para ayudar a los usuarios a superar las dificultades técnicas que puedan experimentar. Esto puede incluir la provisión de manuales de uso, guías de solución de problemas y asistencia técnica. De esta manera, los usuarios pueden sentirse más





seguros y confiados en el uso de la red inalámbrica, lo que puede mejorar su percepción de calidad del servicio y, por lo tanto, la QoE.

Además, ofrecer recursos y soluciones adecuados puede ser especialmente relevante para aquellos usuarios con menor conocimiento técnico, como pueden ser los estudiantes de humanidades que pueden tener menos familiaridad con las tecnologías inalámbricas.

#### 6.3.1.3. *Expectativas previas*

Las expectativas previas de los usuarios se refieren a las percepciones que tienen los usuarios sobre el servicio de red inalámbrica antes de utilizarlo, ya sea por experiencias previas o por información recibida de otros usuarios o fuentes.

Es importante tener en cuenta las expectativas previas de los usuarios ya que esto puede influir en su percepción de la QoS. Si los usuarios tienen expectativas muy altas sobre el servicio, pueden experimentar una percepción negativa si la calidad del servicio no cumple con sus expectativas. Por otro lado, si los usuarios tienen expectativas bajas, pueden experimentar una percepción positiva si la calidad del servicio supera sus expectativas.

Por ello, es necesario identificar las expectativas previas de los usuarios para poder establecer estrategias para satisfacerlas. Esto puede incluir la comunicación clara de los niveles de QoS ofrecidos, el establecimiento de compromisos y garantías de calidad, y la provisión de información clara y detallada sobre el servicio de red inalámbrica. Así, los usuarios pueden tener una percepción más realista del servicio y ajustar sus expectativas en consecuencia, lo que puede mejorar su percepción de calidad del servicio.

#### 6.3.1.4. *Experiencias previas*

Las experiencias previas de los usuarios hacen referencia a las interacciones que han tenido los usuarios con el servicio de red inalámbrica anteriormente. Estas experiencias pueden moldear la percepción de la QoS que los usuarios tienen en interacciones futuras con la red.

Es fundamental considerar las experiencias previas de los usuarios, ya que éstas pueden influir en su evaluación de la QoS. Si las experiencias previas con el servicio fueron satisfactorias, los usuarios pueden ser más comprensivos si la calidad del servicio disminuye temporalmente. Sin embargo, si las experiencias previas fueron negativas, los usuarios pueden ser más críticos con la calidad del servicio, incluso si ésta mejora.

Por ello, es importante reconocer y entender las experiencias previas de los usuarios para poder desarrollar estrategias que mejoren su percepción de la calidad del servicio. Esto puede implicar aprender de errores y problemas pasados, implementar mejoras en la red basadas en el feedback de los usuarios, y proporcionar un servicio consistente y confiable que refuerce experiencias positivas.



#### 6.3.1.5. *Uso previsto*

El uso previsto se refiere al tipo de actividad que los usuarios tienen previsto realizar en la red inalámbrica, así como a la cantidad de tiempo que estarán utilizando la red.

En este caso de estudio, es importante tener en cuenta el uso previsto de los usuarios dado que puede influir en la percepción de la QoS de la red. Si los usuarios tienen previsto realizar actividades que requieren un alto consumo de datos, como la descarga o el streaming de vídeos, pueden experimentar una percepción negativa de la calidad del servicio si no se les proporciona suficiente ancho de banda. De igual forma, si los usuarios van a utilizar la red durante largos periodos de tiempo, pueden experimentar una percepción negativa si se producen interrupciones o problemas de conectividad.

Por tanto, se debe identificar el uso previsto de los usuarios y adaptar el servicio de red inalámbrica a sus necesidades específicas. Por ejemplo, se puede asignar los recursos de ancho de banda en función del tipo de actividad que se va a realizar, la provisión de soluciones para garantizar la disponibilidad y la calidad del servicio durante largos periodos de tiempo, etc. De esta manera, los usuarios pueden realizar sus actividades de forma satisfactoria y experimentar una percepción positiva de la calidad del servicio.

#### 6.3.2. Factores del sistema

A continuación, se va a realizar el análisis de los factores de sistema relevantes que pueden influir en la experiencia de calidad de los usuarios, pero en este caso, se centrará el análisis en el estudio de este proyecto en particular.

##### 6.3.2.1. *Factores relacionados con la red*

Los factores relacionados con la red son un aspecto clave a considerar al identificar los factores de influencia en este caso de estudio. Estos factores pueden incluir aspectos técnicos (objetivos) relacionados con la calidad de la conexión, como la velocidad de transferencia de datos, la disponibilidad de ancho de banda y la estabilidad de la señal, así como otros aspectos relacionados con la infraestructura de red, como el número de dispositivos conectados a la red y la congestión de la red.

Es uno de los aspectos más importantes a tener en cuenta, ya que la calidad de la conexión es un aspecto crítico para la percepción de QoS, y a su vez para la QoE. Si la conexión es lenta o inestable, los usuarios pueden experimentar una percepción negativa de la calidad del servicio y, por lo tanto, reducir su QoE. Por otro lado, si la conexión es rápida y estable, los usuarios pueden experimentar una percepción positiva de la calidad del servicio y, por lo tanto, aumentar su QoE.

Es un aspecto crítico a tener en cuenta para identificar los problemas relacionados y así poder mejorarlos. Esto puede incluir la optimización de la infraestructura de red o la implementación de medidas de gestión de tráfico. Asimismo, es importante monitorear de forma regular la calidad de la conexión y adoptar medidas preventivas para evitar congestiones y otros problemas técnicos que puedan afectar a la calidad del servicio.



### 6.3.2.2. Factores relacionados con el dispositivo

Uno de los factores de influencia relevantes a considerar es el dispositivo utilizado por parte de los usuarios. Depende del dispositivo utilizado, puede haber diferencias respecto al procesamiento de los datos o la compatibilidad con los protocolos de la red.

La calidad del dispositivo utilizado puede tener un impacto significativo en la percepción de calidad del servicio de la red por parte de los usuarios. Si el dispositivo no tiene suficientes recursos para manejar la información y los datos, los usuarios pueden experimentar una percepción negativa de la calidad del servicio, lo que puede reducir su QoE.

Por ende, es importante identificar los factores técnicos relacionados con el dispositivo utilizado y tomar medidas para optimizar su rendimiento. Esto puede incluir actualizar el software y el hardware del dispositivo, administrar las aplicaciones que se ejecutan en el dispositivo y adoptar medidas de seguridad para proteger la información transmitida a través de la red inalámbrica.

### 6.3.2.3. Factores relacionados con el contenido

Los factores relacionados con el contenido tienen un impacto significativo en la percepción de calidad del servicio de red por parte de los usuarios.

Estos factores se refieren al tipo de acción realizada a través de la red (descarga/subida de archivos, navegación web...). Si el contenido es muy complejo o requiere una gran cantidad de recursos de la red, los usuarios pueden experimentar una percepción negativa de la calidad del servicio, lo que puede reducir su QoE. Por otro lado, si el contenido es fácil de manejar y no requiere grandes cantidades de ancho de banda, los usuarios pueden experimentar una percepción positiva de la calidad del servicio, lo que puede aumentar su QoE.

Es esencial identificar los factores relacionados con el contenido utilizado en la red. Al identificar las actividades más realizadas dentro de la red, se pueden asignar los recursos de manera óptima y eficiente, lo que puede mejorar la percepción de calidad del servicio de red inalámbrica por parte de los usuarios.

## 6.3.3. Factores de contexto

Para finalizar, se va a realizar el análisis de los factores de contexto de mayor importancia que pueden influir en la experiencia de calidad de los usuarios, pero de nuevo, se centrará el análisis en el estudio de este proyecto en particular.

### 6.3.3.1. Contexto físico

Un factor importante relacionado con el contexto específico, es el contexto físico en el que se utiliza la red WiFi. Este factor se refiere al lugar donde se encuentra el usuario y cómo puede afectar la percepción de calidad del servicio.

El contexto físico puede influir en la QoE de los usuarios de diferentes maneras. Por ejemplo, si el lugar donde se utiliza la red inalámbrica tiene una alta densidad de usuarios o está



rodeado de objetos que interfieren con la señal inalámbrica, puede haber una disminución en la calidad del servicio. De igual manera, si el lugar cuenta con una buena infraestructura de red y está libre de interferencias, puede haber una percepción positiva de la calidad del servicio.

Es importante considerar el contexto físico al identificar los factores de influencia en el caso de estudio, ya que esto puede ayudar a identificar los lugares con más o menos señal dentro del campus universitario. Por tanto, en los lugares donde no haya buena cobertura, se pueden realizar una serie de mejoras, tales como, la instalación de equipos adicionales para mejorar la cobertura de la señal y la identificación de interferencias en la señal para mitigar su impacto en la calidad del servicio, entre otros.

#### 6.3.3.2. Contexto temporal

Entre los factores de contexto, se encuentra el contexto temporal, es decir, el momento del día en el que se utiliza la red inalámbrica.

El contexto temporal puede afectar la calidad del servicio de red WiFi de diversas maneras. Por ejemplo, durante las horas pico, cuando hay una gran cantidad de usuarios conectados a la red, puede haber una menor disponibilidad de ancho de banda, lo que puede afectar negativamente la calidad del servicio. Por otro lado, durante las horas de menor uso, la calidad del servicio puede mejorar debido a una menor cantidad de usuarios conectados a la red.

Para comprender mejor la percepción de calidad del servicio por parte de los usuarios, es esencial tener en cuenta el contexto temporal en el que se utiliza la red. Al conocer la hora del día en la que se utiliza la red inalámbrica, se pueden tomar medidas para mejorar la calidad del servicio y aumentar la satisfacción de los usuarios.

Por lo tanto, se pueden identificar las horas pico en el que se utiliza la red inalámbrica y tomar medidas para optimizar su rendimiento. Por ejemplo, se puede llevar a cabo la implementación de medidas de gestión de tráfico para garantizar una distribución equitativa de los recursos de la red.

#### 6.3.3.3. Contexto social

Otro factor de influencia a considerar es el contexto social en el que se utiliza la red inalámbrica. Este factor se refiere a cómo las interacciones sociales pueden afectar la percepción de calidad del servicio de red inalámbrica.

El contexto social puede influir en la QoE de los usuarios de diferentes maneras. Por ejemplo, si el lugar donde se utiliza la red inalámbrica es un espacio público, como una biblioteca o un café, la presencia de otros usuarios puede afectar la calidad del servicio debido a la sobrecarga de la red. Por otro lado, en un contexto social más cerrado, como una oficina o un aula, es posible que haya menos usuarios conectados a la red, lo que puede mejorar la calidad del servicio.

Es importante considerar el contexto social al identificar los factores de influencia en el caso de estudio, ya que esto puede ayudar a comprender mejor la percepción de calidad del servicio por parte de los usuarios. Al conocer el contexto social en el que se utiliza la red



inalámbrica, se pueden tomar medidas para mejorar la QoS y, por lo tanto, aumentar la satisfacción de los usuarios.

#### 6.3.3.4. Contexto técnico

El contexto técnico es otro de los factores importantes a considerar en la evaluación de la calidad del servicio en una red WiFi. Este factor se refiere a la relación entre los distintos sistemas que componen la red, como el enrutador, los dispositivos de acceso, la infraestructura de red y los protocolos utilizados.

Es importante tener en cuenta que estos factores pueden influir en la QoE del usuario de diferentes maneras. Por ejemplo, una red WiFi con un router de baja calidad o una infraestructura de red obsoleta puede tener una mala QoS, lo que puede afectar negativamente la experiencia del usuario al reducir la velocidad de conexión o causar interrupciones en la conexión.

Por lo tanto, es necesario evaluar el contexto técnico de la red WiFi para identificar posibles problemas que puedan afectar la calidad del servicio y tomar medidas para solucionarlos. Esto puede incluir la actualización de la infraestructura de red, la implementación de nuevos protocolos o la optimización de la configuración del enrutador y otros dispositivos de acceso.

#### 6.3.4. Criterios de selección

1. **Impacto en la QoE (25 %):** se deben seleccionar los factores de influencia que tengan un impacto directo en la QoE de los usuarios al utilizar la conexión WiFi en los campus universitarios.
2. **Relevancia para los usuarios (20 %):** es importante seleccionar los factores de influencia que más afectan la calidad de la conexión WiFi desde la perspectiva de los usuarios en cada uno de los campus universitarios.
3. **Diferencias significativas entre los campus universitarios (20 %):** se deben seleccionar los factores de influencia que muestren diferencias significativas entre los dos campus universitarios para poder comparar la calidad de la conexión WiFi de manera efectiva.
4. **Facilidad de medición (20 %):** se deben seleccionar los factores de influencia que sean medibles de manera fácil y precisa para poder realizar el estudio de manera efectiva y obtener resultados confiables.
5. **Posibilidad de mejora (15 %):** se deben seleccionar los factores de influencia que tengan un mayor potencial para ser mejorados y, por lo tanto, para mejorar la calidad de la conexión WiFi en los campus universitarios.



Crterios	Perfil de usuario	Conocimientos técnicos	Expectativas previas	Uso previsto	Porcentaje (%)
Relevancia para los usuarios	3	3	3	3	20
Diferencia entre los campus universitarios	5	5	5	4	20
Facilidad de medición	4	4	4	5	20
Impacto en la QoE	4	4	4	4	25
Posibilidad de mejora	2	2	2	2	15
<b>TOTAL</b>	<b>3,7</b>	<b>3,7</b>	<b>3,7</b>	<b>3,7</b>	

Tabla 4: Matriz de alternativas de factores humanos

En este caso, se ha determinado que todos los factores humanos analizados obtienen la misma puntuación, concluyendo que todos tienen la misma importancia en el caso de estudio. Esto sugiere que todos ellos deben ser considerados cuidadosamente durante la planificación y ejecución de la encuesta para obtener un resultado más preciso de la QoE de los usuarios.

Crterios	Factores relacionados con la red	Factores relacionados con el contenido	Factores relacionados con el dispositivo	Porcentaje (%)
Relevancia para los usuarios	5	5	4	20
Diferencia entre los campus universitarios	3	4	3	20
Facilidad de medición	4	3	5	20
Impacto en la QoE	5	5	5	25
Posibilidad de mejora	3	3	3	15
<b>TOTAL</b>	<b>4,1</b>	<b>4,1</b>	<b>4,1</b>	

Tabla 5: Matriz de alternativas de factores de sistema



De la misma manera, se ha determinado que todos los factores de sistema analizados obtienen la misma puntuación, lo que sugiere que cada uno de ellos es esencial para garantizar una realización óptima de la encuesta y obtener conclusiones rigurosas sobre la percepción de los usuarios en cuanto a los factores del sistema de la red.

Crterios	Contexto físico	Contexto temporal	Contexto social	Contexto técnico	Porcentaje (%)
Relevancia para los usuarios	4	4	4	4	20
Diferencia entre los campus universitarios	3	3	3	3	20
Facilidad de medición	4	4	3	2	20
Impacto en la QoE	4	4	4	4	25
Posibilidad de mejora	3	3	3	2	15
<b>TOTAL</b>	<b>3,65</b>	<b>3,65</b>	<b>3,45</b>	<b>3,25</b>	

Tabla 6: Matriz de alternativas de los factores de contexto

En este caso, se ha observado que los factores con mayor puntuación son el contexto físico y temporal. Estos factores son especialmente importantes debido a su facilidad de medición y su impacto en la QoE de los usuarios. Aunque el contexto social también es un aspecto relevante, en este caso podría ser redundante, ya que algunos de sus parámetros, como los lugares con mayor o menor cobertura y la cantidad de usuarios conectados, pueden medirse con los factores mencionados: contexto temporal y físico.

En cuanto al contexto técnico, aunque es un factor importante, su medición puede resultar un poco complicada. En este caso, se ha decidido que es prescindible para la elaboración de la encuesta, ya que otros factores tienen una mayor relevancia en este proyecto.

#### 6.4. Entorno de desarrollo

En la sección 5.3.1, se han analizado varios entornos de desarrollo para la creación de modelos de ML, incluyendo Jupyter Notebooks, Google Colab, Data Spell, PyCharm y Visual Studio Code. Sin embargo, en este análisis se ha decidido enfocarse en los tres primeros entornos, ya que son los que mejor se adaptan a las necesidades específicas del caso de estudio. Jupyter Notebooks, Google Colab y Data Spell son especialmente adecuados para el análisis de datos y la creación de modelos de ML. Son entornos flexibles y versátiles que permiten la colaboración y la experimentación con diferentes enfoques de análisis y modelos. Además, ofrecen herramientas útiles para la visualización de datos y la comunicación de resultados.



#### 6.4.1. Jupyter Notebooks

Jupyter Notebooks, como analizado anteriormente, es una plataforma de código abierto y gratuita que permite la integración de diferentes lenguajes de programación como Python, R y Julia. Se trata de un entorno de desarrollo interactivo que permite la exploración y prototipado de modelos de ML y análisis de datos.

Entre las ventajas de Jupyter Notebooks, destaca su carácter de código abierto y gratuito, lo que la hace accesible a cualquier usuario interesado en la ciencia de datos. Además, Jupyter Notebooks es altamente personalizable y permite la integración de diferentes lenguajes de programación, lo que la convierte en una herramienta versátil y flexible. Otra ventaja es que los notebooks pueden ser compartidos fácilmente a través de plataformas como GitHub o Google Collab.

Entre las desventajas de Jupyter Notebooks, destaca su complejidad para usuarios nuevos o que no estén familiarizados con el lenguaje de programación. Además, la gestión de versiones de los notebooks puede ser complicada y puede llevar a problemas de compatibilidad entre diferentes versiones. Otra desventaja importante de Jupyter Notebooks es que, aunque es un entorno de desarrollo interactivo, no es un IDE completo y no cuenta con opciones avanzadas de “debugging”. Esto puede hacer que la identificación y corrección de errores en el código sea más complicada en comparación con otras herramientas de programación.

#### 6.4.2. Google Colab

Google Colab, como explicaco en el apartado 5.3.1.2, es una plataforma de código abierto y gratuita que permite el desarrollo de proyectos de ciencia de datos y ML. Sin embargo, Google Colab se ejecuta en la nube y ofrece algunas características adicionales debido a su integración con los servicios de Google.

Una de las ventajas más destacadas de Google Colab es su facilidad de acceso. Al ser una herramienta basada en la nube, no requiere ninguna instalación o configuración local, lo que la hace muy conveniente para usuarios que no deseen lidiar con la configuración de un entorno de desarrollo en sus propias máquinas. Además, Google Colab está estrechamente integrado con otros servicios de Google, como Google Drive, lo que facilita el almacenamiento y acceso a los datos y notebooks.

Google Colab también ofrece una amplia gama de recursos computacionales, incluyendo unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPU). Estos recursos son especialmente útiles para acelerar el entrenamiento de modelos de ML y procesamiento de datos a gran escala.

Google Colab es compatible con múltiples lenguajes de programación, con un enfoque principal en Python. Esto permite a los usuarios trabajar con bibliotecas populares de ML como TensorFlow, PyTorch y scikit-learn.





### 6.4.3. DataSpell

Tal y como se ha visto en 13, DataSpell es un entorno de desarrollo integrado (IDE) específico para la ciencia de datos, que permite el trabajo con diferentes lenguajes de programación como Python, R y SQL, entre otros. Es una herramienta de pago que ofrece una amplia gama de características y herramientas para la exploración y análisis de datos, y para la creación de modelos de aprendizaje automático.

Entre las ventajas de DataSpell, destaca su amplio conjunto de herramientas de análisis de datos y modelado, que incluyen bibliotecas de aprendizaje automático integradas y herramientas de visualización avanzadas. Además, ofrece una interfaz de usuario intuitiva y fácil de usar, lo que la hace accesible para usuarios con diferentes niveles de experiencia. También ofrece opciones avanzadas de “debugging” y un sistema de gestión de versiones integrado, lo que facilita la colaboración en proyectos de ciencia de datos en equipo.

Entre las desventajas de DataSpell, destaca su carácter de pago, lo que puede limitar su accesibilidad a usuarios con menos recursos o presupuestos más reducidos. Además, al ser una herramienta de propósito específico, puede requerir una curva de aprendizaje mayor en comparación con otras herramientas de programación.

### 6.4.4. Criterios de selección

1. **Capacidad de procesamiento (25 %):** es uno de los aspectos más importantes a tener en cuenta. Se debe seleccionar un entorno de desarrollo que tenga una alta capacidad de procesamiento para manejar la gran cantidad de datos y ejecutar los algoritmos de ML de manera eficiente.
2. **Flexibilidad de la plataforma (25 %):** también es uno de los aspectos más importantes ya que, para trabajar de forma eficiente es importante seleccionar una plataforma que sea flexible y permita la implementación de diferentes algoritmos de ML para analizar y correlacionar los datos.
3. **Interfaz de usuario amigable (20 %):** es un aspecto con un poco de menos peso, pero también es relevante seleccionar una plataforma que tenga una interfaz de usuario amigable y fácil de usar para facilitar la manipulación y análisis de datos.
4. **Compatibilidad con las herramientas existentes (20 %):** este aspecto tiene el mismo peso que el anterior, ya que es igual de importante seleccionar un entorno de desarrollo que sea compatible con las herramientas de análisis y visualización de datos existentes para poder integrarlas fácilmente en el proceso de análisis.
5. **Coste (10 %):** este es el aspecto con el menor peso, ya que, aunque es importante seleccionar un entorno de desarrollo que se adapte al presupuesto disponible, es más importante que sea una plataforma óptimo en los aspectos mencionados.



Crterios	Jupyter Notebooks	Google Colab	DataSpell	Peso (%)
Capacidad de procesamiento	4	5	5	25
Flexibilidad de la plataforma	4	5	5	25
Interfaz de usuario amigable	4	5	4	20
Compatibilidad con las herramientas existentes	5	4	4	20
Costo	5	5	2	10
<b>TOTAL</b>	<b>4,3</b>	<b>4,8</b>		

Tabla 7: Matriz de alternativas de los entornos de desarrollo

En general, se puede observar que las tres plataformas cumplen con los criterios de seleccin para el proyecto. Sin embargo, Google Colab obtiene la puntuacin ms alta en los criterios evaluados. Esto la convierte en la opcin preferida en trminos generales. Ofrece una capacidad de procesamiento slida, flexibilidad en la plataforma, una interfaz de usuario amigable y una buena compatibilidad con herramientas existentes. Google Colab es una opcin adecuada para desarrollar el modelo de correlacin, ya que proporciona todas las herramientas necesarias integradas y se caracteriza por su facilidad de uso.

## 6.5. Lenguaje de programacin

Para desarrollar un modelo de ML de manera eficiente, es fundamental elegir un lenguaje de programacin que se adapte mejor a las necesidades especficas del proyecto en cuestin. Como se ha analizado previamente en la seccin 5.3.1, aunque existen varios lenguajes de programacin que pueden ser utilizados, ya que el proyecto se enfoca en programar un modelo de ML, los lenguajes ms comnmente utilizados y extendidos para este propsito son Python y R. Por lo tanto, en este anlisis se compararn estos dos lenguajes para determinar cul es el ms adecuado para el desarrollo del modelo de ML.

### 6.5.1. Python

Tal y como se ha explicado en la seccin 5.3.1.5, Python es un lenguaje de programacin popular y ampliamente utilizado para la implementacin de soluciones de ML. Es un lenguaje de programacin de alto nivel que se enfoca en la legibilidad del cdigo y en la facilidad de uso.

Python tiene una amplia gama de bibliotecas de ML disponibles, como TensorFlow, Keras, Scikit-learn, y PyTorch, entre otras. Estas bibliotecas permiten la implementacin de modelos de ML de manera sencilla y rpida.

Python es un lenguaje de programacin eficiente en tiempo de ejecucin y escalable. Adem{s, cuenta con una comunidad de desarrollo muy activa y una gran cantidad de recursos disponibles on-line, lo que facilita el aprendizaje y la solucin de problemas. Tambi{en es un



lenguaje de programación compatible con diferentes sistemas operativos y arquitecturas de hardware.

Otra ventaja importante de Python es su capacidad para procesar grandes conjuntos de datos de manera eficiente. Python cuenta con bibliotecas de procesamiento de datos, como Pandas y Numpy, que permiten la manipulación y el análisis de grandes conjuntos de datos.

A pesar de sus muchas ventajas, Python también tiene algunas desventajas. Una de ellas es que, como es un lenguaje interpretado, puede ser menos eficiente en términos de velocidad de ejecución en comparación con lenguajes compilados. Además, aunque Python es fácil de aprender y usar, es posible que programadores más experimentados prefieran lenguajes más complejos y de mayor rendimiento.

### 6.5.2. R

Tal y como se ha analizado en la sección 5.3.1.6, R es un lenguaje de programación enfocado en el análisis estadístico y la visualización de datos. Es ampliamente utilizado en la industria y la academia para la implementación de soluciones de ML y análisis de datos.

Al igual que Python, R cuenta con una amplia gama de bibliotecas de ML, como Caret, Random Forest y XGBoost. Además, R cuenta con bibliotecas especializadas en estadísticas y análisis de datos, como ggplot2 y dplyr, que permiten la visualización y el análisis de datos de manera sencilla y efectiva.

R es un lenguaje interpretado, lo que significa que su velocidad de ejecución puede ser menor que la de lenguajes compilados. Sin embargo, R cuenta con bibliotecas optimizadas, como data.table y dplyr, que permiten el procesamiento eficiente de grandes conjuntos de datos.

Una ventaja importante de R es su amplia comunidad de desarrollo y su gran cantidad de recursos en línea. Esto facilita el aprendizaje y la solución de problemas. Además, R es un lenguaje de programación gratuito y de código abierto, lo que lo hace accesible para todos.

Sin embargo, una de las desventajas de R es que su sintaxis puede ser menos intuitiva y más complicada que la de otros lenguajes de programación, como Python. Esto puede dificultar su aprendizaje para programadores que no tienen experiencia previa en programación.

### 6.5.3. Criterios de selección

- 1. Soporte de bibliotecas de ML (20%):** el soporte de bibliotecas de ML es un criterio crucial para el proyecto ya que se necesitan bibliotecas especializadas para la implementación de modelos para la correlación de QoS/QoE. Se ha asignado un 20% de importancia a este criterio debido a su gran relevancia para el proyecto.
- 2. Escalabilidad (20%):** la escalabilidad es importante porque se necesita manejar grandes conjuntos de datos y modelos complejos. Por tanto, también se ha asignado un 20% de importancia a este criterio.
- 3. Eficiencia en tiempo de ejecución (15%):** la eficiencia en tiempo de ejecución es importante porque se busca que el modelo se ejecute de manera eficiente y en tiempo



real. Se ha asignado un 15% de importancia a este criterio porque, aunque es relevante, no es tan crucial como otros criterios como el soporte de bibliotecas de ML.

4. **Facilidad de uso (15%):** la facilidad de uso es importante porque puede afectar la curva de aprendizaje y la eficiencia en el desarrollo del proyecto. Se ha asignado un 15% de importancia a este criterio porque no es imprescindible para la realización del proyecto.
5. **Comunidad de desarrollo (15%):** la comunidad de desarrollo es importante porque puede proporcionar soporte y ayuda en caso de problemas o dificultades durante el desarrollo del proyecto. Al igual que en el caso anterior, se ha asignado un 15% de importancia a este criterio.
6. **Portabilidad (10%):** La portabilidad es importante porque es interesante buscar que el modelo se pueda implementar en diferentes plataformas. Se ha asignado un 10% de importancia a este criterio porque, aunque sea relevante, no tiene la misma importancia que los criterios anteriores para el proyecto en cuestión.
7. **Coste (5%):** algunos lenguajes de programación son de código abierto y gratuitos, mientras que otros requieren una licencia para su uso. El coste puede ser un factor importante a considerar, especialmente si hay un presupuesto limitado.

Criterios	Python	R	Peso (%)
Soporte de bibliotecas ML	5	4	20
Escalabilidad	4	3	20
Eficiencia en tiempo de ejecución	5	3	15
Facilidad de uso	5	4	15
Comunidad de desarrollo	4	3	15
Portabilidad	5	4	10
Costo	5	5	10
<b>TOTAL</b>	<b>4,65</b>	<b>3,55</b>	

Tabla 8: Matriz de alternativas de los lenguajes de programación

Al analizar la tabla comparativa, se puede observar que tanto Python como R obtienen puntuaciones altas en la mayoría de los criterios evaluado, sin embargo, Python obtiene una puntuación ligeramente más alta que R.

Se podría decir que Python obtiene una puntuación más alta debido a su amplia gama de bibliotecas de ML disponibles, su facilidad de uso, su comunidad de desarrollo activa y su capacidad para procesar grandes conjuntos de datos. Además, Python es ampliamente utilizado y cuenta con una gran cantidad de recursos y tutoriales on-line, lo que lo hace más accesible para aprenderlo de forma más ágil. Por lo tanto, se ha seleccionado Python para el desarrollo del modelo ML.

## 6.6. Metodología de valoración

El objetivo de este proyecto es recolectar valoraciones subjetivas por parte de los usuarios acerca de su experiencia con la red WiFi. En el estado del arte se han analizado diversas metodologías subjetivas existentes. Sin embargo, en este caso específico, la medición de la



experiencia subjetiva de los usuarios se llevará a cabo mediante una encuesta on-line utilizando la plataforma Microsoft Forms, según lo establecido por el CEISH y en el marco de la memoria M10/2023/184 aprobada por este comité en la que se contempla la utilización de datos y la posibilidad de encuestación en el marco del proyecto de investigación "Análisis para la gestión global de la Calidad de Servicio (QoS) en red WiFi en entorno universitario" del grupo de investigación NQaS.

Es importante destacar que algunas metodologías de valoración, como ACR, ACR-HR, DCR y pares, no son adecuadas para este tipo de encuesta en línea, ya que su naturaleza se basa en experimentos específicos con los usuarios en entornos específicos y condiciones controladas.

Por este motivo, no tiene sentido realizar un análisis comparativo entre estas metodologías, dejando como la única opción adecuada el sistema MOS. MOS se utiliza comúnmente para valorar la experiencia de los usuarios en diversos escenarios, incluyendo la calidad de video y audio, pero no está limitado a estas modalidades. Por lo tanto, es posible emplear este método para evaluar la calidad de la red inalámbrica en cuestión.

Como ya se ha explicado en el apartado 5.6, el sistema MOS es una técnica de valoración subjetiva utilizada para medir la QoE de un servicio en función de las opiniones subjetivas de los usuarios. Esta técnica se basa en preguntar a los usuarios sobre su opinión en una escala numérica o verbal, como una escala de 1 a 5 o de "muy malo" a "excelente". El resultado final se obtiene mediante la media aritmética de las respuestas de los usuarios, lo que proporciona una puntuación global que representa la percepción subjetiva de los usuarios sobre la calidad del servicio.

En el caso del presente TFM, el sistema MOS se utilizará para medir la QoE de los usuarios en relación a la red WiFi. Para ello, se tiene que diseñar un cuestionario que pregunte a los usuarios sobre su experiencia en la red WiFi, incluyendo preguntas sobre la velocidad de conexión, la calidad de la señal, la facilidad de uso y la satisfacción general. Cada pregunta podría ser valorada por los usuarios en una escala de 1 a 5, donde 1 significa "muy malo" y 5 significa "excelente".

Una vez que se hayan recopilado las respuestas de los usuarios, se calcula la puntuación global de la QoE mediante la media aritmética de las respuestas de los usuarios a todas las preguntas del cuestionario. Esta puntuación global representaría la percepción subjetiva de los usuarios sobre la calidad de la red WiFi.

Es importante destacar que, para que los resultados del sistema MOS sean precisos y relevantes, es necesario que se realice una muestra representativa (es necesario realizar un muestreo adecuado que permita seleccionar a los participantes de manera que reflejen de manera precisa la población objetivo) de usuarios y se obtenga una cantidad significativa de respuestas. Además, también es importante considerar la naturaleza subjetiva de esta técnica, ya que los resultados pueden variar según la percepción individual de cada usuario.



## 7. METODOLOGÍA

### 7.1. Descripción de la solución

Esta sección proporciona una visión detallada de cómo se ha implementado y desarrollado el modelo de correlación de QoS/QoE. En esta sección, se describen los pasos específicos que se han seguido en la implementación del modelo, junto con los resultados obtenidos y su análisis.

En primer lugar, se aborda la recolección de datos tanto objetivos como subjetivos, los cuales serán fundamentales para la realización del modelo. A continuación, se presenta la implementación del modelo de correlación de QoS/QoE, donde se describen los pasos desde la preparación de los datos hasta el entrenamiento y evaluación del modelo, incluyendo las herramientas y lenguajes de programación utilizados.

#### 7.1.1. Desarrollo del modelo de correlación QoS/QoE

El desarrollo del modelo de correlación de QoS/QoE es un componente fundamental en el marco de este proyecto. El objetivo principal de este modelo es establecer una relación sólida entre los parámetros objetivos de QoS, medidos en términos de indicadores KPI, y la QoE percibida por los usuarios. En este contexto específico, se busca crear un gráfico que relacione un único parámetro objetivo de QoS con los resultados de las MOS obtenidas de los usuarios.

Para lograr el desarrollo óptimo y efectivo del modelo de correlación de QoS/QoE, es necesario llevar a cabo una serie de pasos clave. A continuación, se enumeran los pasos correspondientes, los cuales serán detallados en mayor profundidad posteriormente:

- 1. Recopilación de datos:** Obtener los datos necesarios para el modelo, que incluyen los parámetros objetivos de QoS y las puntuaciones de MOS de los usuarios.
- 2. Preparación del entorno de ejecución:** Este paso implica configurar el entorno de ejecución donde se desarrollará y entrenará el modelo de ML.
- 3. Preparación de los datos:** Realizar una limpieza y preparación de los datos para asegurar su calidad y consistencia. Esto puede incluir la eliminación de datos irrelevantes o duplicados, el manejo de valores atípicos y la normalización de los datos.
- 4. Desarrollo del modelo:** En este paso, se procede a desarrollar el modelo de ML utilizando las bibliotecas y herramientas adecuadas.
- 5. Evaluación del modelo:** Utilizar el conjunto de prueba para evaluar el rendimiento del modelo. Esto implica calcular métricas de evaluación.

##### 7.1.1.1. Recopilación de datos

La recopilación de datos es una etapa fundamental en el desarrollo del modelo. En este paso, se busca obtener los datos necesarios para alimentar el modelo y permitir el análisis de la QoS y la QoE de los usuarios. Estos datos proporcionarán la base para comprender la relación entre los parámetros objetivos de QoS y la percepción subjetiva de los usuarios.



Durante la fase de recopilación de datos, el objetivo es obtener información relevante sobre el parámetro objetivo de QoS (throughput). Para lograrlo, se han recopilado conjuntos de datos que representan los dos escenarios distintos de red. En la Ilustración 9, se presenta un ejemplo parcial de los datos recopilados en la Facultad de Ingeniería, que proporciona una visión de los resultados recopilados de la red.

AP Name	Radio Type	Device Name	Map Location	SSID	Profile	VLAN ID	Protocol	Session Duration	Policy Type	Avg. Session Throughput (Kbps)	Host Name	Client Type	Speed
AP-ETSilyT-1702-U022636	802.11a/n/ac	WLC-HA_UPV_Leioa	BIZ-Bilbao > Ingenieros > Sotano 1	eduroam	eduroam	35	802.11n(5GHz)	16 hrs 10 min 32 sec	WPA2	0.1	NaN	Unified Wireless	NaN
AP-ETSilyT-1702-U022345	802.11a/n/ac	WLC-HA_UPV_Leioa	BIZ-Bilbao > Ingenieros > Planta Baja	eduroam	eduroam	35	802.11ac	1 hrs 40 min 22 sec	WPA2	247.9	NaN	Unified Wireless	NaN
AP-ETSilyT-1702-U022345	802.11a/n/ac	WLC-HA_UPV_Leioa	BIZ-Bilbao > Ingenieros > Planta Baja	eduroam	eduroam	35	802.11ac	50 min 11 sec	WPA2	89.3	NaN	Unified Wireless	NaN
AP-ETSilyT-1702-U022388	802.11a/n/ac	WLC-HA_UPV_Leioa	BIZ-Bilbao > Ingenieros > Planta 3	eduroam	eduroam	35	802.11ac	50 min 10 sec	WPA2	13.3	NaN	Unified Wireless	NaN
AP-ETSilyT-1602E-U022766	802.11a/n	WLC-HA_UPV_Leioa	BIZ-Bilbao > Ingenieros > Planta 2	eduroam	eduroam	35	802.11n(5GHz)	50 min 11 sec	WPA2	<0.1	NaN	Unified Wireless	NaN

Ilustración 9: Ejemplo de datos objetivos de la red WIFI en la escuela de Ingeniería

Además, se considera esencial recopilar datos de la QoE, que proporcionan información subjetiva sobre la calidad experimentada por los usuarios al utilizar los servicios de la red. Para obtener estos datos subjetivos, se tenía previsto realizar una encuesta para recopilar opiniones y percepciones de los usuarios. Sin embargo, lamentablemente, surgieron problemas relacionados con la comisión ética que han impedido la realización y lanzamiento de las encuestas planificadas. Como alternativa, se ha decidido realizar una simulación de los datos de MOS para obtener información subjetiva sobre la calidad de experiencia de los usuarios. Los detalles y el enfoque de esta simulación se explicarán más adelante.

### Recopilación de datos objetivos

Los conjuntos de datos recopilados corresponden a los dos entornos universitarios distintos de este caso de estudio: la Facultad de Educación de Bilbao (en la UPV/EHU) y la Escuela Superior de Ingeniería (ESI). Ambos conjuntos de datos fueron registrados el 3 de mayo de 2023 a las 17:48:49, con el objetivo de analizar y comparar el rendimiento de las redes WiFi en estos entornos de estudio. Cada conjunto de datos captura información detallada sobre las conexiones WiFi y las interacciones de los usuarios en la red.

La duración de las sesiones se registra en los conjuntos de datos, proporcionando información sobre la temporalidad de las conexiones WiFi de los usuarios. Esto es relevante para evaluar la estabilidad y la continuidad de las conexiones en los entornos universitarios. Además, los conjuntos de datos incluyen medidas objetivas de rendimiento, como el throughput medio



en las sesiones de cada usuario. A continuación, se realiza una breve descripción de la información presente en los conjuntos de datos:

1. **Client IP Address:** La dirección IP asignada al cliente (dispositivo) que se conecta a la red WiFi. Este dato es importante para identificar cada dispositivo en la red.
2. **Client MAC Address:** La dirección MAC (Media Access Control) del cliente, que es un identificador único asignado a la tarjeta de red del dispositivo. Esta información es esencial para identificar de manera precisa y unívoca cada dispositivo conectado a la red.
3. **id:** Identificador único del cliente. Esta columna proporciona un identificador único para cada cliente en los conjuntos de datos, lo cual facilita su seguimiento y análisis individual.
4. **Association Time:** El momento en que el cliente se ha asociado (conectado) a la red WiFi. Este dato permite analizar la duración de las conexiones y comprender el comportamiento de los clientes en relación con el tiempo.
5. **Global Unique:** Identificador único global. Esta columna puede contener identificadores únicos adicionales asociados a los clientes, los cuales pueden ser útiles para realizar análisis más detallados.
6. **Local Unique:** Identificador único local. Al igual que el campo anterior, esta columna puede contener identificadores únicos adicionales para los clientes.
7. **Link Local:** Dirección IP de enlace local asignada al cliente. Esta dirección IP se utiliza para la comunicación dentro de la red local y es relevante para el enrutamiento de datos.
8. **Endpoint Type:** Tipo de punto final del cliente (por ejemplo, cliente móvil, cliente fijo, etc.). Esta información permite diferenciar los diferentes tipos de dispositivos que se conectan a la red WiFi.
9. **CCX:** Cisco Compatible Extensions, una especificación de Cisco para mejorar las capacidades de los dispositivos inalámbricos. Esta columna indica si los clientes son compatibles con las extensiones de Cisco y puede ser relevante para análisis específicos relacionados con la compatibilidad de dispositivos.
10. **Mobility Oracle:** Oracle de movilidad. Esta columna puede contener información adicional relacionada con la movilidad de los dispositivos en la red, proporcionada por un sistema de movilidad específico.
11. **Mobility Controller:** Controlador de movilidad. Esta columna indica el controlador de movilidad utilizado para gestionar los movimientos de los dispositivos en la red.
12. **Anchor Mobility Controller:** Controlador de movilidad ancla. Esta columna indica el controlador de movilidad utilizado como punto de referencia para gestionar la movilidad de los dispositivos en la red.
13. **Switch Peer Group:** Grupo de pares de conmutadores. Esta columna puede contener información sobre los grupos de pares de conmutadores utilizados en la red para mejorar la eficiencia y la redundancia.
14. **Anchor Switch Peer Group:** Grupo de pares de conmutadores ancla. Esta columna indica el grupo de pares de conmutadores utilizado como punto de referencia para mejorar la eficiencia y la redundancia en la red.
15. **E2E:** End-to-End (extremo a extremo), que indica que se refiere a un parámetro o estado que abarca todo el trayecto de comunicación. Esta columna puede indicar la medición de parámetros de extremo a extremo relacionados con la calidad de servicio.





16. **PMIP State:** Estado de PMIP (Protocolo de movilidad de IP). Esta columna puede contener información sobre el estado del Protocolo de Movilidad de IP, que se utiliza para administrar la movilidad de los dispositivos en redes IP.
17. **PMIP Connected Interface:** Interfaz conectada PMIP. Esta columna indica la interfaz de red utilizada para establecer la conexión PMIP, lo cual es relevante para el enrutamiento y la comunicación de los dispositivos.
18. **Home Address:** Dirección IP de inicio del cliente. Esta columna muestra la dirección IP asignada como "hogar" para cada cliente en la red, lo cual puede ser útil para rastrear y administrar las conexiones de los dispositivos.
19. **Access Technology Type:** Tipo de tecnología de acceso utilizada por el cliente (por ejemplo, 802.11a, 802.11n, etc.). Esta columna indica la tecnología inalámbrica utilizada por los dispositivos para acceder a la red WiFi.
20. **Local Link Identifier:** Identificador de enlace local. Este campo puede contener un identificador único asignado a cada enlace local en la red, lo cual puede ser relevante para el enrutamiento y la gestión de la conectividad.
21. **LMA:** Local Mobility Anchor (ancla local de movilidad). Esta columna indica el ancla local de movilidad utilizado en la red, que es un punto de referencia para el manejo de la movilidad de los dispositivos.
22. **Vendor:** Fabricante o proveedor del cliente o dispositivo. Esta columna muestra el nombre del fabricante o proveedor del dispositivo cliente, lo cual puede ser útil para identificar las características y especificaciones de los dispositivos utilizados en la red.
23. **AP Name:** Nombre del punto de acceso (Access Point). Esta columna indica el nombre asignado a cada punto de acceso en la red WiFi.
24. **Radio Type:** Tipo de radio del punto de acceso (por ejemplo, 2.4 GHz, 5 GHz). Esta columna muestra el tipo de radio utilizado por cada punto de acceso, lo cual es relevante para comprender las características de la red inalámbrica.
25. **Device Name:** Nombre del dispositivo. Esta columna muestra el nombre asignado a cada dispositivo cliente en la red.
26. **Map Location:** Ubicación en un mapa. Esta columna puede contener información geográfica o de ubicación asociada a cada dispositivo o punto de acceso en la red.
27. **SSID:** Service Set Identifier, el nombre de la red WiFi a la que el cliente se está conectando. Esta columna muestra el nombre de la red inalámbrica a la que cada cliente se ha conectado.
28. **Profile:** Perfil asociado al cliente. Esta columna indica el perfil de configuración o ajustes específicos aplicados a cada cliente en la red.
29. **VLAN ID:** ID de VLAN (Virtual LAN) asociada al cliente. Esta columna muestra el identificador único de la VLAN a la que cada cliente está asignado, lo cual es relevante para la segmentación y el enrutamiento de la red.
30. **Protocol:** Protocolo utilizado (por ejemplo, TCP, UDP). Esta columna indica el protocolo de comunicación utilizado en cada conexión de red.
31. **Session Duration:** Duración de la sesión del cliente. Esta columna muestra el tiempo de duración de la sesión de cada cliente en la red WiFi.
32. **Policy Type:** Tipo de política aplicada al cliente. Esta columna indica el tipo de política de red o conjunto de reglas aplicadas a cada cliente.
33. **Avg. Session Throughput (Kbps):** Promedio del rendimiento de la sesión en kilobits por segundo. Esta columna muestra el rendimiento promedio de la sesión de cada cliente, es decir, la velocidad de transferencia de datos promedio durante el tiempo de conexión.



34. **Host Name:** Nombre del host o dispositivo. Esta columna muestra el nombre asignado al host o dispositivo cliente en la red.
35. **Client Type:** Tipo de cliente (por ejemplo, cliente corporativo, cliente de invitado). Esta columna indica el tipo de cliente conectado a la red WiFi, lo cual puede ser relevante para analizar diferentes perfiles de usuarios.
36. **Speed:** Velocidad de conexión del cliente. Esta columna muestra la velocidad de conexión de cada cliente en la red, es decir, la capacidad de transferencia de datos que se puede alcanzar.
37. **AP MAC Address:** Dirección MAC del punto de acceso al que se conecta el cliente. Esta columna muestra la dirección MAC del punto de acceso WiFi al que cada cliente está conectado.
38. **AP IP Address:** Dirección IP del punto de acceso al que se conecta el cliente. Esta columna muestra la dirección IP asignada al punto de acceso WiFi al que cada cliente está conectado.
39. **Device IP:** Dirección IP del dispositivo cliente. Esta columna muestra la dirección IP asignada a cada dispositivo cliente en la red.
40. **Controller Port:** Puerto del controlador utilizado. Esta columna indica el puerto del controlador utilizado para gestionar la conexión del cliente.
41. **Anchor Controller:** Controlador de anclaje. Esta columna indica el controlador de anclaje utilizado en la red, que se encarga de la gestión de la movilidad de los dispositivos.
42. **Association ID:** ID de asociación. Esta columna muestra el identificador único asociado a la asociación entre el cliente y el punto de acceso WiFi.

Los conjuntos de datos analizados contienen una amplia variedad de parámetros que proporcionan información detallada sobre cada sesión de usuario en la red WiFi. Es esencial destacar la importancia de manejar estos datos de manera cuidadosa y respetuosa, siguiendo los principios de protección de datos establecidos en la Ley Orgánica de Protección de Datos (GDPR). En este contexto, es fundamental la pseudoanonimización (enmascarar o sustituir los datos personales con pseudónimos para prevenir la identificación directa o indirecta de los individuos) de datos personales, como los identificadores de usuario y las direcciones IP, lo que garantiza la privacidad de los usuarios. A pesar de esta transformación, la estructura de los datos se mantiene intacta, lo que permite la realización de análisis y modelizaciones. Es crucial recordar que el manejo de datos personales debe realizarse siempre con respeto y de acuerdo con las leyes y regulaciones vigentes, garantizando la confidencialidad y la integridad de los datos de los usuarios.

Además, es importante mencionar que se encuentra en proceso de aprobación la memoria M10/2023/184. Este documento está enmarcado dentro de nuestro proyecto actual y esperamos obtener su aprobación en breve. Esta memoria abarca y consolida la visión y las estrategias que se siguen en el tratamiento de los datos.

Por otro lado, si bien todos los parámetros son importantes para tener una visión completa y contextualizada de cada sesión de usuario, en este caso de estudio se ha decidido centrarse en un parámetro específico, como ya se ha establecido en el apartado 6: el throughput medio de las sesiones. Esta elección se debe a que el throughput es un indicador clave de la QoS en una red WiFi. El throughput medio se define como la velocidad promedio de transferencia de datos durante el tiempo de conexión de cada usuario a la red WiFi. Es un indicador directo de la



capacidad de la red para satisfacer las demandas de transmisión de datos de los usuarios. Un alto throughput indica una mayor capacidad de transmisión de datos y, por lo tanto, una mejor calidad de servicio.

Este enfoque se justifica por varias razones. La norma IEEE 802.11, que rige las redes WiFi, menciona que el throughput es una métrica esencial para medir la capacidad de la red. Las actualizaciones sucesivas de esta norma, como la IEEE 802.11n y la IEEE 802.11ac, se han centrado en gran medida en mejorar el throughput. Además, un informe de Cisco, líder en el sector de la tecnología de redes, titulado "Understanding QoS in Wireless LANs" [28], identifica el throughput como una de las tres métricas claves de la QoS, junto con la latencia y el jitter.

En el caso de los conjuntos de datos, el "avg. throughput" (tal y como se denomina la columna) de una sesión se refiere al promedio de la tasa de transferencia de datos durante la duración de una sesión de conexión a la red WiFi. Representa la velocidad media a la que se transmiten los datos entre el usuario (dispositivo conectado a la red WiFi) y el punto de acceso (AP) durante el tiempo en que se mantiene la conexión.

El "avg. throughput" se expresa en kilobits por segundo (Kbps) y se calcula dividiendo la cantidad total de datos enviados y recibidos durante la sesión por la duración de la sesión. Este valor proporciona una medida promedio del rendimiento de la conexión durante el tiempo en que el usuario estuvo conectado a la red WiFi.

El throughput desempeña un papel crucial en la calidad de la experiencia del usuario en una red WiFi. Una conexión de alta velocidad, representada por un alto throughput, permite una transferencia rápida y eficiente de datos, lo que se podría traducir en una experiencia de usuario mejorada. Los usuarios pueden disfrutar de una navegación web fluida, una reproducción de video sin interrupciones, descargas rápidas y una mayor capacidad para realizar múltiples tareas en línea. Por otro lado, una conexión lenta con un bajo throughput puede resultar en retrasos en la carga de páginas web, interrupciones en la transmisión de video y una experiencia general insatisfactoria.

Al correlacionar el throughput con las puntuaciones MOS, se obtendrá una comprensión más profunda de cómo la calidad de la red WiFi, representada por el throughput, afecta la percepción y satisfacción de los usuarios. Esta información puede ser invaluable para la gestión de la red, permitiendo identificar áreas de mejora y tomar medidas para optimizar el rendimiento de la red y mejorar la calidad de la experiencia del usuario.

### **Recopilación de datos subjetivos**

Como se ha comentado anteriormente, debido a problemas de plazo con la comisión ética, se ha tomado la decisión de simular los resultados de las valoraciones MOS. Esta elección se ha tomado tras una cuidadosa consideración de los posibles riesgos y beneficios. Los estudios de investigación en este ámbito normalmente dependen de las respuestas de los usuarios para obtener estos datos. Sin embargo, estos problemas éticos pueden generar complicaciones significativas.

Por lo tanto, para garantizar la continuidad de la investigación y la obtención de datos relevantes, se optó por la simulación de los resultados MOS. Esta decisión fue informada y



respaldada por diversos estudios previos en este campo. Un estudio particularmente relevante es "Effect of quality of service parameters on quality of experience for YouTube service in mobile networks" [29]. En este artículo, los investigadores llevaron a cabo un análisis exhaustivo de cómo las variaciones en los parámetros de la QoS, como el throughput, afectan la QoE (Quality of Experience), medida a través de los resultados MOS. La simulación de los resultados MOS se utilizó en este estudio para prever y evaluar el impacto de los parámetros de QoS en la QoE del usuario.

Aunque esta estrategia no reemplaza completamente los datos recopilados a través de encuestas directas a los usuarios, ofrece una solución viable y éticamente responsable en la situación donde la recopilación de datos primarios no es factible.

Sin embargo, es importante señalar que, aunque los resultados obtenidos a través de la simulación proporcionan información valiosa, no pueden capturar todas las complejidades y factores contextuales que podrían surgir de los datos recopilados directamente de los usuarios. Como tal, los resultados deben interpretarse con cierta cautela, y es esencial considerar la realización de investigaciones adicionales una vez que se resuelvan los problemas éticos relacionados con la encuesta a los usuarios.

Para simular los datos de las puntuaciones MOS, se ha comenzado con el análisis de los datos de throughput disponibles en los conjuntos de datos existentes. A partir de estos, se ha asignado una puntuación MOS para cada valor de throughput. Para introducir un elemento de variabilidad y simular el factor de contexto que se experimenta en situaciones reales, se ha añadido una cierta aleatoriedad a la asignación de las puntuaciones MOS.

Teniendo en cuenta las posibles diferencias entre las necesidades y expectativas de los usuarios en diferentes facultades, se ha ajustado la simulación de los datos MOS para los conjuntos de datos correspondientes a las facultades de Educación e Ingeniería. Para reflejar las necesidades de la Facultad de Educación, que presumiblemente son menos exigentes en términos de calidad de red que la Facultad de Ingeniería, se han generado los datos MOS de tal manera que reflejen esta diferencia. Este ajuste ha sido el resultado de una investigación adicional sobre las necesidades específicas y expectativas de los usuarios en estas dos facultades. Además, se espera que será validado una vez se cuente con la aprobación del comité de ética y se lleven a cabo las encuestas planificadas.

De esta manera, se han generado los datos MOS simulados, que, aunque no son un sustituto directo de las encuestas de usuario, representan una estimación informada y basada en la investigación de las puntuaciones MOS que se podrían haber obtenido a través de dichas encuestas.

Una vez que se han recopilado todos los datos de throughput correspondientes a cada sesión de usuario junto con sus respectivas puntuaciones MOS simuladas, se puede proceder a desarrollar el modelo. De nuevo destacar que las puntuaciones MOS son simuladas, y en consecuencia, una vez que se puedan lanzar las encuestas a los usuarios, el modelo se entrenará con las MOS obtenidas directamente de los usuarios para obtener un modelo aún más preciso y representativo de la experiencia real del usuario. No obstante, para los propósitos de este estudio, las puntuaciones MOS simuladas permiten extraer conclusiones bastante significativas.



### 7.1.1.2. Preparación del entorno de ejecución

Como ya se ha establecido en el apartado 5.3.1, se va a utilizar la plataforma Google Colab para la realización del modelo. Google Colab es una plataforma basada en la nube que ofrece un entorno de desarrollo gratuito y colaborativo para ejecutar código Python, incluyendo la implementación y evaluación de modelos de ML. Para ello, se han seguido los siguientes pasos:

1. **Acceso a Google Colab:** Para comenzar, se accede a la plataforma Google Colab a través de un navegador web. El sitio web oficial de Google Colab se encuentra en <https://colab.research.google.com/>. Este acceso permite crear y abrir cuadernos de Colab, que son documentos interactivos donde se desarrolla y ejecuta el código.
2. **Creación de un nuevo cuaderno:** En Google Colab, se ha creado un nuevo cuaderno seleccionando la opción "Nuevo cuaderno" en la página de inicio o a través del menú "Archivo". También existe la opción de abrir cuadernos existentes desde Google Drive o desde repositorios en GitHub.
3. **Configuración del entorno de ejecución:** Una vez dentro del cuaderno, se ha configurado el entorno de ejecución. En la barra de herramientas superior, se encuentra un menú desplegable llamado "Entorno de ejecución". En esta sección, se han realizado ajustes como la selección de la versión de Python, la asignación de recursos de procesamiento (CPU o GPU) y la cantidad de memoria RAM destinada al entorno.
  - En particular, se ha seleccionado una CPU para el entrenamiento del modelo de ML, ya que el conjunto de datos es relativamente pequeño y no requiere una gran cantidad de recursos de computación para su entrenamiento.
4. **Importación de bibliotecas:** En un principio, se han importado las bibliotecas de Python que más se van a utilizar para el proyecto:
  - **scikit-learn:** scikit-learn es una biblioteca de que ofrece una amplia gama de algoritmos y herramientas para tareas de clasificación, regresión, agrupación, reducción de dimensionalidad y más. En este caso, scikit-learn proporciona una implementación robusta y optimizada del algoritmo de regresión lineal, lo que permite realizar fácilmente el entrenamiento y la predicción utilizando modelos de regresión lineal.
  - **numpy:** numpy es una biblioteca fundamental para el cálculo numérico en Python. Proporciona un soporte eficiente para matrices multidimensionales y funciones matemáticas que permiten realizar operaciones numéricas de manera rápida y eficiente. En este caso, numpy se utiliza para manipular y procesar los datos necesarios para el entrenamiento del modelo.
  - **pandas:** pandas es una biblioteca de análisis de datos que proporciona estructuras de datos flexibles y herramientas para la manipulación y análisis de conjuntos de datos. Ofrece estructuras de datos como DataFrame, que es una tabla bidimensional con etiquetas en columnas y filas, similar a una hoja de cálculo. En este caso, pandas se utiliza para cargar y manipular los datos de entrada. Se utiliza para leer los datos desde archivos.
  - **matplotlib:** matplotlib es una biblioteca de visualización de datos en Python que proporciona una interfaz flexible para la generación de una amplia variedad de gráficos y diagramas. matplotlib se integra bien con otras bibliotecas de la pila científica de Python, como numpy y pandas, lo que la convierte en la opción



natural para la visualización de datos en muchos análisis y flujos de trabajo de ML.

### 7.1.1.3. Preparación de los datos

En este paso, se lleva a cabo un conjunto de tareas para garantizar la calidad, coherencia y adecuación de los datos recopilados antes de utilizarlos en el análisis y entrenamiento del modelo.

La preparación de los datos implica varios procesos, que incluyen la limpieza, transformación y procesamiento de los datos recopilados. El objetivo principal es asegurarse de que los datos sean consistentes, completos y estén en el formato adecuado para su uso en el modelo de correlación.

En este caso, el modelo se entrenará utilizando los datos del throughput medio de cada sesión junto con su valoración MOS correspondiente. Antes de comenzar a desarrollar el modelo, es crucial obtener información relevante sobre los datos disponibles para determinar su consistencia y características. Con ese fin, se han recopilado una serie de estadísticas descriptivas, que incluyen el valor más común, el valor máximo, la media y la desviación estándar para ambos conjuntos de datos.

Estos datos proporcionan información valiosa sobre la distribución y tendencia central de los valores de throughput. El valor más común nos indica cuál es el valor dominante o más frecuente en los conjuntos de datos. La media da una idea del valor promedio o central, mientras que la desviación estándar indica la dispersión o variabilidad de los datos alrededor de la media. A continuación, se indican los valores obtenidos en ambos conjuntos de datos:

	Valor más común	Valor máximo	Media	Desviación estándar
Facultad de Ingeniería	0,1	2.9413,4	134,286	811,543
Facultad de Educación	0,1	10.269,7	199,980	591,388

Tabla 9: Estadísticas descriptivas

Al analizar los datos proporcionados, se observa que tanto en la Facultad de Ingeniería como en la Facultad de Educación existen valores extremos con una desviación estándar considerablemente grande. Esta alta variabilidad en los datos puede plantear desafíos al desarrollar el modelo, ya que los datos abarcan un rango muy amplio. Para obtener resultados más precisos al entrenar el modelo, es importante realizar una limpieza de los datos. Esto implica identificar y tratar los valores atípicos o extremos. Para ello, se implementa un método de eliminación de *outliers*<sup>1</sup> basado en la desviación estándar. Este método consta de los siguientes pasos:

- Calcular la media y la desviación estándar de la columna que contiene los datos.

---

<sup>1</sup> Un outlier es un valor atípico que difiere significativamente de la tendencia general de los datos y puede afectar la precisión de los modelos de Machine Learning.



- Definir un umbral multiplicando la desviación estándar por un factor para determinar el rango dentro del cual los valores se considerarán no *outliers*.
- Filtrar los datos del DataSet manteniendo solo aquellos valores que estén dentro del rango definido.

A continuación, se detalla el código utilizado para la realización de este proceso:

```
media = data['Avg. Session Throughput (Kbps)'].mean()
desviacion_estandar = data ['Avg. Session Throughput (Kbps)'].std()

umbral = 3
data_filt = data[(data ['Avg. Session Throughput (Kbps)'] >= media -
umbral * desviacion_estandar) & (data_ingenieria['Avg. Session
Throughput (Kbps)'] <= media
```

Como se puede observar en el fragmento de código, se define un umbral para determinar el rango dentro del cual los valores se considerarán no *outliers*. A continuación, se realiza un proceso de filtrado en el DataFrame original seleccionando únicamente aquellos valores de la columna "Avg. Session Throughput (Kbps)" que caen dentro del rango determinado por la media más/menos el umbral multiplicado por la desviación estándar. Esto implica eliminar los valores atípicos del conjunto de datos, lo que resulta en un DataFrame filtrado y más consistente para su posterior entrenamiento.

#### 7.1.1.4. Análisis exploratorio

El análisis exploratorio de datos es una etapa fundamental en el proceso de ML. Consiste en examinar y comprender los datos antes de aplicar cualquier modelo o algoritmo. Su objetivo principal es obtener información y conocimientos clave sobre los datos, identificar patrones, relaciones y características relevantes. Como se ha mencionado anteriormente, las columnas que se utilizan para la realización del modelo son: "Avg. Session Throughput (Kbps)" y "MOS" en ambos conjuntos de datos. Por lo tanto, es crucial comprender la relación entre estas dos variables para obtener una comprensión más profunda de su naturaleza y lograr un modelo más preciso.

Para establecer la relación entre estas dos variables, se emplean diferentes métodos. En primer lugar, se realiza un mapa de calor que muestra la correlación entre "Avg. Session Throughput (Kbps)" y "MOS". Posteriormente, se utiliza un gráfico de dispersión para visualizar la relación entre ambas variables.

A continuación, se muestran los mapas de calor obtenidos de la Facultad de Ingeniería y Educación, respectivamente, que indican la correlación entre las dos variables:

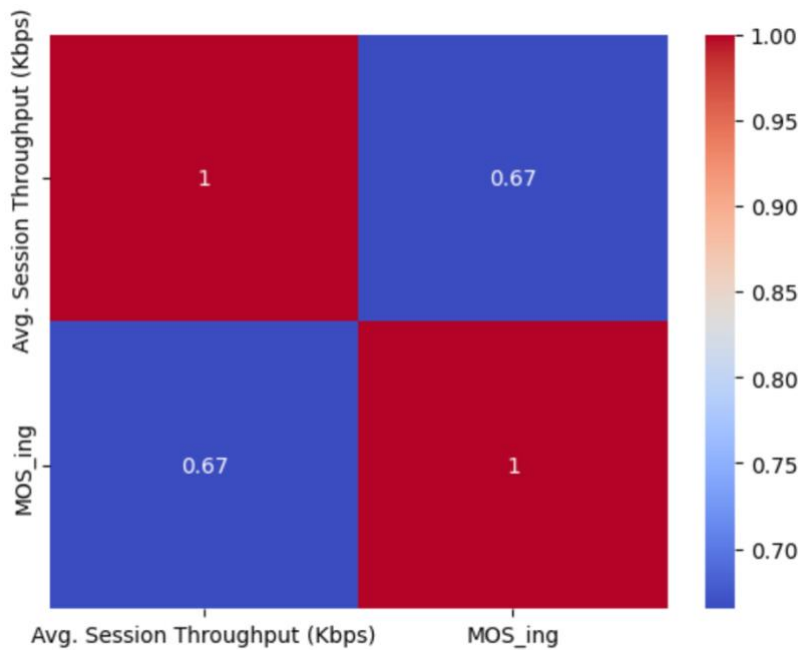


Ilustración 10: Mapa de calor de la correlación entre el throughput y el MOS en la Facultad de Ingeniería

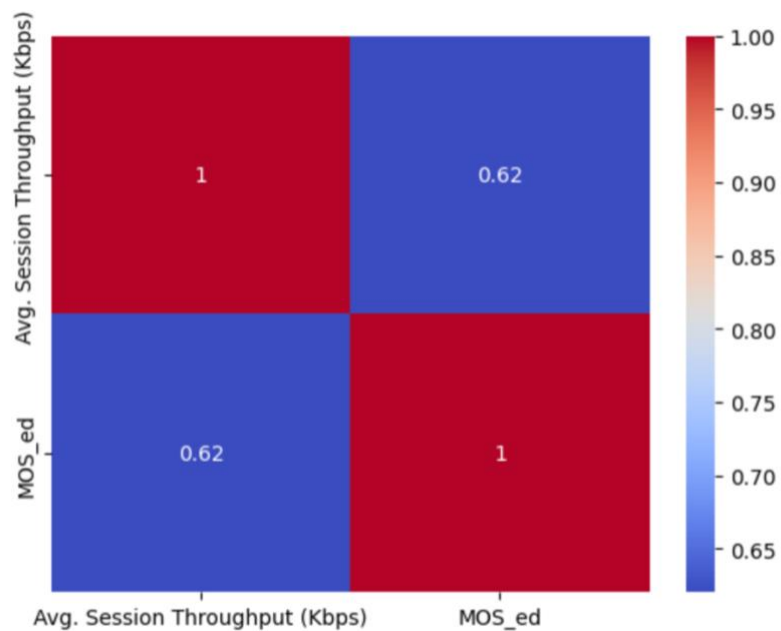


Ilustración 11: Mapa de calor de la correlación entre el throughput y el MOS en la Facultad de Educación

Tal y como se puede ver en la Ilustración 10 y Ilustración 11, se obtiene un coeficiente de correlación de 0.67 y 0.62 en el conjunto de datos de la Facultad de Ingeniería y Educación, respectivamente, que indican en los dos casos una correlación positiva moderada entre el throughput y el MOS. Esto significa que existe una tendencia general de que a medida que el throughput promedio de una sesión aumenta, la valoración MOS también tiende a ser mayor. Sin embargo, es importante destacar que la correlación no implica causalidad, es decir, no se puede afirmar que el aumento del throughput sea la causa directa de una mayor valoración MOS.





Además, es relevante tener en cuenta que una correlación de esta magnitud indica que la relación entre estas dos variables no es perfecta y puede existir cierta variabilidad en los datos. Otros factores también pueden influir en la valoración MOS (factores de influencia) y es posible que haya casos en los que el throughput sea alto pero la valoración MOS sea baja, o viceversa.

#### 7.1.1.5. *Desarrollo del modelo: regresión lineal*

Una vez se ha adquirido un mayor conocimiento sobre las relaciones entre las variables en las dos facultades y se han observado las diferencias existentes, se puede construir un modelo de ML que capture de manera óptima la relación entre estas variables. El objetivo es utilizar este modelo para predecir la puntuación de los usuarios en función del throughput de una red WiFi. En otras palabras, el modelo permitirá tener una estimación de la experiencia de los usuarios basada en los datos objetivos de la red (throughput en este caso).

Este enfoque brinda la ventaja de utilizar datos históricos recopilados de las valoraciones de los usuarios y el rendimiento del throughput para entrenar el modelo. Una vez entrenado, el modelo podrá generalizar y hacer predicciones precisas sobre la calidad de la experiencia de los usuarios en función del throughput medido. Esto resulta útil para la gestión y mejora de redes WiFi, ya que permite tomar decisiones informadas para optimizar el rendimiento y satisfacer las necesidades de los usuarios.

Siguiendo la decisión tomada en el análisis de alternativas, se ha optado por utilizar la regresión lineal como modelo para este propósito. La regresión lineal busca establecer una relación lineal entre las variables independientes y la variable dependiente, en este caso, la puntuación MOS que puede tomar continuos del 1 al 5. El objetivo es entrenar este modelo para identificar las relaciones entre las variables y predecir la puntuación MOS correspondiente a cada valor de throughput. A través del análisis de la relación lineal, se buscará encontrar una ecuación que permita estimar la puntuación MOS en función del throughput.

La regresión lineal es una opción adecuada debido a su capacidad para modelar relaciones lineales entre variables y realizar predicciones basadas en esas relaciones. Al entrenar este modelo con los datos disponibles, se espera que aprenda los patrones y las características que relacionan el throughput con la puntuación MOS. La regresión lineal permite establecer una ecuación que estima la puntuación MOS en función del throughput, lo que proporciona una forma de entender la relación entre estas variables y predecir los resultados en base a nuevos valores de throughput. En comparación con técnicas de clasificación, que usa clases predefinidas, los modelos de regresión proporcionan una mayor granularidad del resultado previsto y, por lo tanto, permiten cubrir una gama más amplia de posibles casos de uso.

Una vez que el modelo de regresión lineal haya sido entrenado y validado, estará preparado para predecir nuevas puntuaciones MOS en función de los valores de throughput. Esto proporcionará una herramienta útil para estimar la satisfacción de los usuarios en función de las mediciones de throughput, lo que permitirá tomar decisiones informadas y realizar mejoras pertinentes para brindar una experiencia óptima a los usuarios. A continuación, se describen los pasos llevados a cabo para la construcción del modelo de regresión lineal.



## División de los datos

En esta etapa, se divide el conjunto de datos recopilados en diferentes subconjuntos que serán utilizados para entrenar, validar y evaluar el rendimiento del modelo.

El objetivo principal de la división de los datos es asegurar que el modelo sea capaz de generalizar y realizar predicciones precisas en datos no vistos previamente. Para lograr esto, es necesario separar los datos en conjuntos independientes y representativos.

La división de los datos generalmente se realiza en tres conjuntos principales: entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, es decir, para ajustar los parámetros del modelo según los datos de entrada y las salidas esperadas. El conjunto de validación se utiliza para ajustar los hiperparámetros<sup>2</sup> del modelo y optimizar su rendimiento. Finalmente, el conjunto de prueba se utiliza para evaluar el rendimiento final del modelo en datos no vistos previamente. Sin embargo, en este caso, dado que el conjunto de datos es pequeño, se opta por realizar únicamente la división en conjuntos de entrenamiento y prueba.

La división de los datos en ambos conjuntos de datos se realiza de la siguiente manera: el 80% de los datos se utilizan para entrenar el modelo y el 20% restante se reservan para realizar pruebas y evaluar la precisión y el rendimiento del modelo. A continuación, se presenta el código desarrollado.

```
X_entrenamiento, X_prueba, y_entrenamiento, y_prueba =
train_test_split(X, y, test_size=0.2, random_state=42)
```

El código utiliza la función **train\_test\_split** para dividir los datos en conjuntos de entrenamiento y prueba. Aquí se muestra cómo se realiza la división:

- **X** es la matriz de características o variables independientes (throughput).
- **y** es el vector de la variable objetivo o variable (MOS).
- **test\_size = 0.2** indica que se desea asignar el 20% de los datos para el conjunto de prueba y el 80% restante para el conjunto de entrenamiento.
- **random\_state= 42** es una semilla utilizada para la generación de números aleatorios, lo cual asegura que la división de datos sea reproducible.

El resultado de esta línea de código es la generación de cuatro conjuntos de datos:

**X\_entrenamiento, X\_prueba, y\_entrenamiento y y\_prueba.**

- **X\_entrenamiento** contiene los datos de características de entrenamiento.
- **X\_prueba** contiene los datos de características de prueba.
- **y\_entrenamiento** contiene los valores objetivo de entrenamiento.
- **y\_prueba** contiene los valores objetivo de prueba para la facultad de ingeniería.

---

<sup>2</sup> Los hiperparámetros son variables que se establecen antes del proceso de entrenamiento del modelo y afectan el rendimiento y comportamiento del algoritmo de Machine Learning.



## Entrenamiento del modelo

Durante este paso, se utilizan los dos conjuntos de datos de entrenamiento para ajustar los parámetros del modelo y aprender los patrones y relaciones subyacentes entre los parámetros objetivos de QoS y la QoE percibida. El objetivo principal del entrenamiento del modelo es lograr que el modelo sea capaz de realizar predicciones precisas y generalizar correctamente en datos no vistos previamente.

Este proceso se desarrolla de la siguiente manera para el caso de ambas facultades:

```
modelo = LinearRegression()
modelo.fit(X_entrenamiento, y_entrenamiento)
predicciones = modelo.predict(X_prueba)
```

Como se puede observar en este fragmento, primero, se crea una nueva instancia de la clase **LinearRegression()** proporcionada por scikit-learn. Después, se llama al método **fit()** del objeto **modelo**, que realmente entrena el modelo. El método **fit()** toma dos argumentos: el conjunto de datos de características **X\_entrenamiento** y el conjunto de datos de etiquetas **y\_entrenamiento**. Cuando se llama al método **fit()**, el modelo de regresión lineal se ajusta a los datos de entrenamiento. Esto implica encontrar la línea (o hiperplano en múltiples dimensiones) que mejor se ajuste a los datos. En términos matemáticos, se busca minimizar la suma de los cuadrados de las diferencias (residuos) entre los valores reales y los predichos por la línea de regresión. Este proceso se conoce como **mínimos cuadrados ordinarios**.

Por último, una vez que el modelo ha sido entrenado, se puede utilizar para hacer predicciones en nuevos datos. Aquí, el método **predict()** toma un conjunto de datos de características (en este caso, **X\_prueba**, que son los datos de prueba del throughput) y devuelve las predicciones del modelo para esos datos.

### 7.1.1.6. Evaluación del modelo: regresión lineal

La evaluación del modelo es una parte esencial del proceso de ML. Involucra el uso de varias métricas y técnicas para determinar cuán bien el modelo aprendió de los datos de entrenamiento y cuán bien puede generalizar a datos nuevos o no vistos. Para ello, en este caso se van a calcular las métricas comúnmente utilizadas en la regresión para evaluar el rendimiento del modelo. Cada métrica aporta una visión distinta, y es interesante analizar cada una de ellas para obtener una visión más global del funcionamiento del modelo. A continuación, se detallan las métricas que se van a emplear:

1. **MSE (Error Cuadrático Medio)**: Esta es una buena métrica si los errores grandes son particularmente indeseables, ya que, debido al cuadrado, los errores grandes se penalizan más.
2. **RMSE (Raíz del Error Cuadrático Medio)**: Esta métrica es una extensión del MSE y presenta la ventaja de estar en las mismas unidades que la variable objetivo. Es útil para comparar errores con la magnitud de la variable de destino.
3. **MAE (Error Absoluto Medio)**: Esta métrica da una idea de cuánto se está equivocando el modelo en promedio, sin preocuparse tanto por los errores grandes.



4. **R<sup>2</sup> (Coeficiente de determinación)**: Esta es una métrica que proporciona una medida del ajuste general del modelo, en términos de qué porcentaje de la variabilidad total en los datos es explicado por el modelo.

En este caso particular, los resultados obtenidos son los siguientes:

	MSE	RMSE	MAE	R <sup>2</sup>
Facultad de Ingeniería	0,214	0,462	0,362	0,455
Facultad de Educación	0,323	0,569	0,457	0,675

Tabla 10: métricas de evaluación del modelo

En primer lugar, El Error Cuadrático Medio (MSE) obtenido sugiere que los modelos cometen un error cuadrático promedio al predecir el MOS. Aunque este valor es relativamente bajo, lo que indica un rendimiento aceptable, señala que existe una oportunidad para mejorar y optimizar los modelos. Este optimismo se ve reforzado por la observación del Error Cuadrático Medio de la Raíz (RMSE), que proporciona una métrica más interpretable al estar en la misma escala que el MOS. Aunque la discrepancia promedio no es insignificante, queda dentro de un rango tolerable dada la escala de MOS.

Además, el Error Absoluto Medio (MAE) confirma que las predicciones de los modelos se desvían de los valores reales en una cantidad promedio moderada. Este resultado, aunque indica una cierta capacidad del modelo para acercarse a los valores reales, también apunta a la necesidad de un mayor refinamiento del modelo.

En última instancia, el Coeficiente de Determinación (R<sup>2</sup>) revela que los modelos pueden explicar una porción considerable, aunque no completa, de la variación en los valores de MOS en función del throughput. Aunque estos modelos proporcionan una capacidad explicativa sustancial, queda una parte significativa de la variación en la calidad de experiencia del usuario (MOS) que aún no está siendo capturada por los modelos.

Una vez evaluadas las métricas de rendimiento, es esencial también considerar otros componentes claves de nuestro modelo de regresión lineal para una comprensión más completa: los **coeficientes de regresión** y el término del **intercepto**.

Los coeficientes de regresión en un modelo de regresión lineal representan la variación esperada en la variable dependiente (en este caso, MOS) por cada unidad de cambio en las respectivas variables independientes, manteniendo constantes todas las demás variables. Estos coeficientes revelan el grado de impacto que las variables de entrada (en este contexto, el throughput) tienen en las predicciones del MOS. Un coeficiente más elevado indica que la variable correspondiente tiene un efecto más pronunciado en la predicción del MOS. Así, se puede apreciar una relación cuantitativa entre el throughput y el MOS gracias a estos coeficientes.

El término de intercepto en el modelo de regresión lineal, por otro lado, representa el valor predicho de la variable dependiente cuando todas las variables independientes se sitúan en cero. En este contexto, se podría interpretar como el valor del MOS que se predice cuando el throughput es cero. Aunque en situaciones prácticas, un throughput de cero es improbable (o



indicaría una calidad de experiencia insatisfactoria), el intercepto puede carecer de una interpretación práctica directa en este contexto. No obstante, es crucial para la precisión de las predicciones del modelo.

En este caso, se han obtenido los siguientes resultados para estas variables:

	<b>Coefficiente</b>	<b>Intercepto</b>
<b>Facultad de Ingeniería</b>	0,0017	1,152
<b>Facultad de Educación</b>	0,003	1,236

*Tabla 11: valores de coeficiente e intercepto*

El coeficiente, que se asocia con el predictor throughput, proporciona una medida de cuánto se espera que cambie el MOS para un cambio unitario en el throughput, manteniendo todo lo demás constante. Para la facultad de Ingeniería, el coeficiente es de 0.0017, mientras que para la facultad de Educación, es ligeramente mayor, a 0.003. Estos valores relativamente pequeños sugieren que los modelos predicen un cambio moderado en el MOS por cada incremento unitario en el throughput. El hecho de que el coeficiente sea más alto para la facultad de Educación sugiere que, en coherencia con las expectativas, los estudiantes de esta facultad pueden ser menos exigentes en términos de throughput, asignando calificaciones de MOS más altas a pesar de un menor throughput en comparación con los usuarios de la facultad de Ingeniería.

El intercepto de un modelo de regresión lineal indica el valor esperado de la variable dependiente (en este caso, el MOS) cuando todas las variables independientes (en este caso, el throughput) son cero. En los modelos para ambas facultades, el intercepto es mayor que 1 (1.152 para Ingeniería y 1.236 para Educación). Estos valores indican que, incluso con un throughput de cero, el MOS predicho no sería de cero. Esta afirmación es coherente con la naturaleza del MOS, cuyo valor mínimo está establecido en 1.

Una vez examinados en profundidad todos los parámetros pertinentes del modelo, resulta útil explorar estos hallazgos de manera más visual para consolidar lo que se ha analizado. Para ello, se construyen una serie de gráficos de dispersión, los cuales ofrecen una representación visual de las relaciones entre las variables estudiadas (throughput y MOS) en ambas facultades (reales y predichas), así como de las discrepancias identificadas entre ellas. Estos gráficos proporcionan una perspectiva intuitiva y accesible de los resultados analíticos obtenidos. A continuación, se muestran los gráficos generados.



- Facultad de Ingeniería

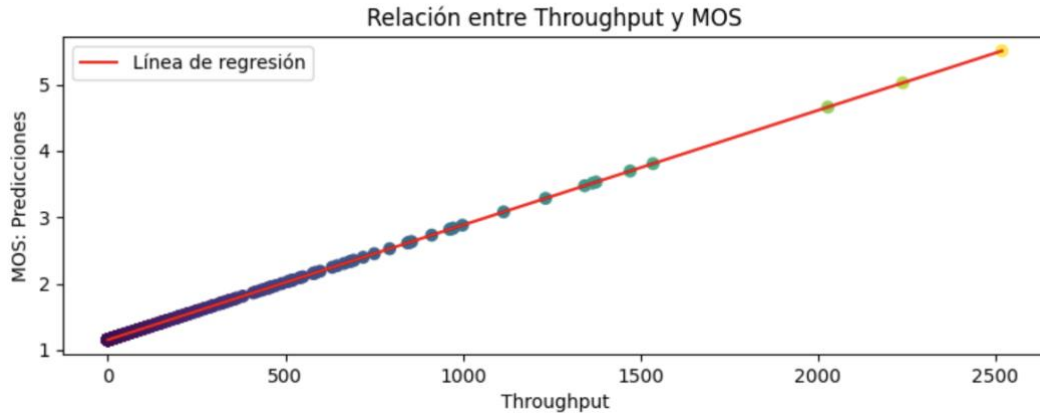


Ilustración 12: Throughput vs MOS: predicciones

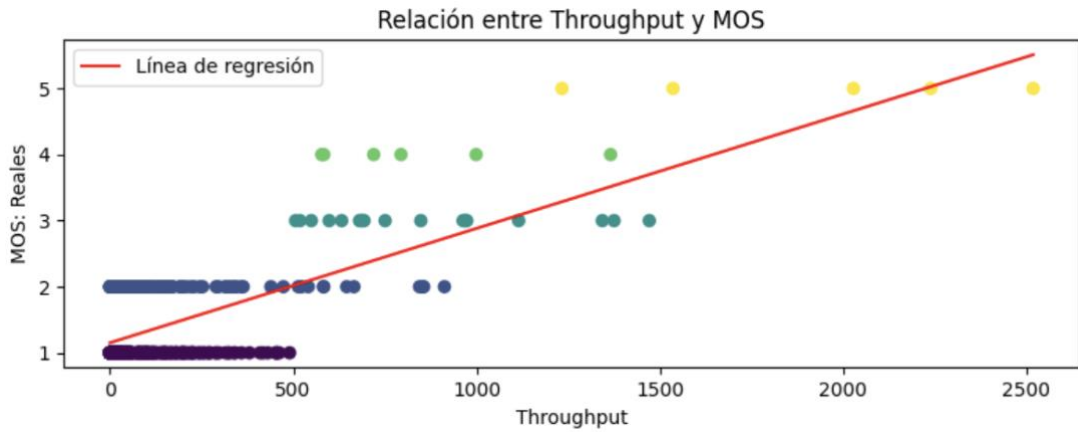


Ilustración 13: Throughput vs. MOS: Reales

- Facultad de educación

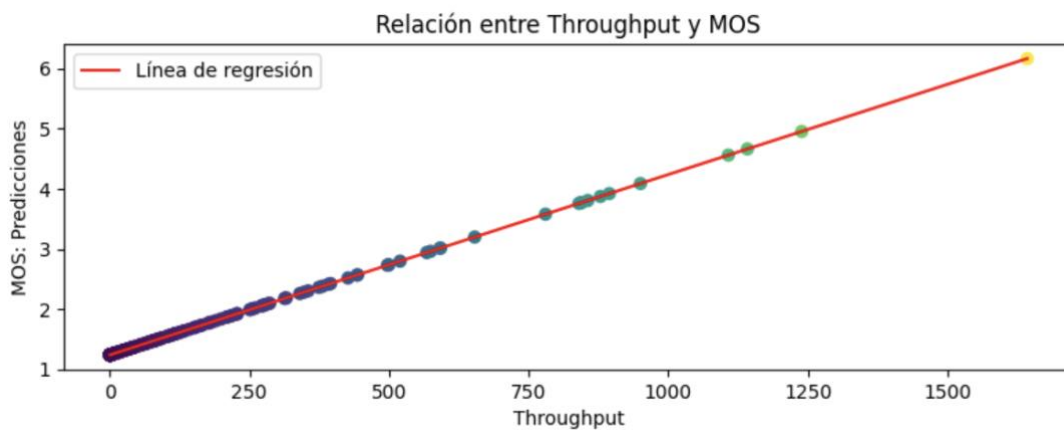
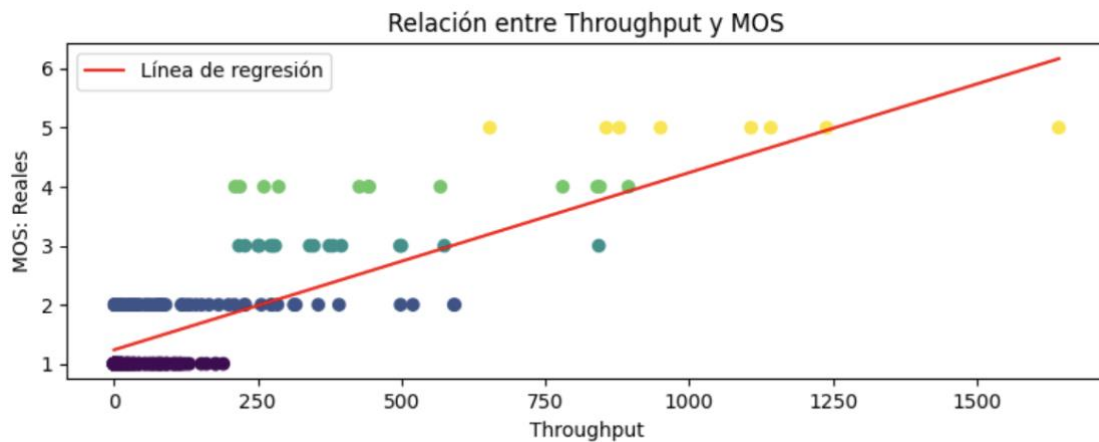


Ilustración 14: Throughput vs. MOS: predicciones



Como se puede apreciar en la Ilustración 12 y Ilustración 14, que representan la relación entre el throughput y el MOS en la Facultad de Ingeniería y Educación respectivamente (predichas), se puede observar una correlación lineal positiva entre ambas variables. Sin embargo, en la Ilustración 13 y Ilustración 15 (valores reales) se puede intuir que la relación entre las variables se asemeja más a una correlación logarítmica. En estos casos, se llega a un punto en el que, a pesar de aumentar el throughput, el MOS se estabiliza en su valor máximo, lo que indica que no se perciben mayores mejoras en la calidad de la red.

Esta discrepancia entre los valores reales y predichos puede atribuirse a la suposición de linealidad en el modelo de regresión. A pesar de esta limitación, el uso de la regresión lineal sigue siendo útil por las siguientes razones:

- **Simplificación del modelo:** La regresión lineal es un modelo sencillo y ampliamente utilizado en el análisis de datos. Aunque pueda no capturar completamente la forma logarítmica de la relación puede proporcionar una aproximación aceptable y una forma más fácil de interpretar los resultados.
- **Interpretación y comprensión:** El modelo de regresión lineal permite una interpretación más intuitiva de los coeficientes y la relación entre las variables. Los coeficientes de regresión lineal proporcionan estimaciones directas de la asociación entre MOS y throughput, lo que facilita la comprensión de la dirección y magnitud de la relación. Aunque pueda haber casos en los que el MOS se sature y el modelo prediga valores más altos que el máximo real, aún es posible extraer conclusiones relevantes a partir de los resultados obtenidos.
- **Predicción:** A pesar de que la relación sea más logarítmica en los valores reales, el modelo de regresión lineal todavía puede usarse para hacer predicciones razonables dentro del rango de datos observados. Si los valores de throughput se encuentran dentro del rango de datos utilizados para ajustar el modelo, la regresión lineal puede proporcionar estimaciones razonables del MOS correspondiente.
- **Linealidad aproximada:** Incluso si la relación es más logarítmica en los valores reales, es posible que exista una aproximación lineal dentro de cierto rango o en ciertas condiciones específicas. En estos casos, la regresión lineal puede ser útil para modelar y analizar la relación en esas circunstancias particulares.



Dicho esto, lo crucial es la identificación de esta relación positiva entre el throughput y el MOS, donde se evidencia que a medida que el throughput mejora, las puntuaciones MOS tienden a incrementarse. Este patrón respalda una lógica intuitiva: a medida que la calidad de la red se mejora, la satisfacción de los usuarios tiende a incrementarse.

Además, es relevante resaltar las diferencias significativas que emergen entre las dos facultades en estudio. A continuación, se muestran dos gráficos de dispersión que comparan las dos facultades, tanto para los valores reales y los predichos.

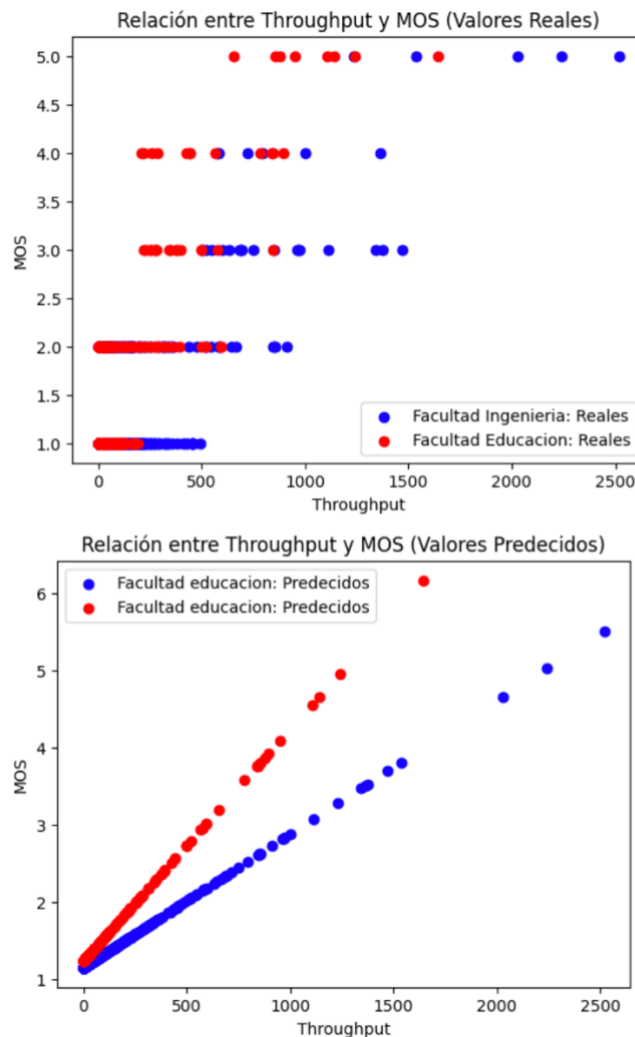


Ilustración 16: Diferencias entre las dos facultades: valores reales y predichos

Como se exploró anteriormente, existen múltiples factores que influyen en cómo los usuarios califican su experiencia con la red. Esto se puede observar en la Ilustración 16, donde se observa que las calificaciones asignadas por los usuarios de la Facultad de Educación tienden a ser un poco superiores, a pesar de tener niveles de throughput menores, en comparación con la Facultad de Ingeniería. Esta distinción también se refleja en las predicciones generadas por el modelo, lo que sugiere que el modelo ha capturado eficazmente estas diferencias. Como se indicó anteriormente, estas variaciones pueden deberse a una variedad de razones. No





obstante, una de las más convincentes podría ser que los estudiantes de ingeniería requieren una mayor calidad de red debido a la naturaleza específica de las actividades que desarrollan.

Por otro lado, al analizar y predecir las calificaciones de los usuarios en ambas facultades, el modelo puede demostrar su capacidad para capturar las diferencias en las experiencias de red y proporcionar predicciones y relaciones precisas. La realización del modelo en diferentes contextos, como distintas facultades universitarias, ayuda a evaluar su generalización y capacidad para adaptarse a diversas situaciones.

En conclusión, en términos generales, de los resultados obtenidos en la evaluación del modelo de regresión lineal, se podría decir que el modelo tiene un rendimiento moderado en la predicción. A pesar de ello, es cierto que se ha logrado identificar una correlación bastante pronunciada entre ambas variables en cada una de las facultades. No obstante, la limitada precisión del modelo puede atribuirse a una variedad de factores:

1. **Insuficiente cantidad de datos:** el conjunto de datos utilizado para entrenar y evaluar el modelo no es lo suficientemente grande o representativo de todas las clases existentes. Si hay una falta de diversidad en los datos o una cantidad limitada de muestras, el modelo puede tener dificultades para capturar la variabilidad y generalizar correctamente.
2. **Características insuficientes:** Es importante considerar si las características utilizadas para entrenar el modelo (en este caso, el throughput) son suficientes para capturar toda la información relevante que influye en las puntuaciones MOS. Puede haber otras variables o características que sean importantes y que no se estén teniendo en cuenta en el modelo actual.
3. **Relación no lineal:** El modelo de Regresión Lineal asume relaciones lineales entre las características y las etiquetas. Sin embargo, en el caso de la correlación entre el throughput y las puntuaciones MOS, es posible que existan relaciones no lineales o complejas que el modelo no pueda capturar adecuadamente. En este caso, podría ser necesario explorar modelos más sofisticados como las redes neuronales.

Por lo tanto, para mejorar la precisión del modelo de regresión lineal y abordar posibles problemas de baja precisión, se pueden considerar varias estrategias. En primer lugar, recopilar más datos de diferentes fuentes o contextos puede enriquecer el conjunto de entrenamiento y permitir al modelo capturar mejor la variabilidad. Además, explorar características adicionales relacionadas con la calidad de la red puede proporcionar información adicional para mejorar las predicciones. También se pueden probar diferentes modelos y técnicas más complejas, que podrían ser más adecuados para capturar relaciones complejas en los datos.

#### *7.1.1.7. Desarrollo del modelo: Red neuronal feedforward*

En base a las deducciones obtenidas a través de la implementación del modelo de regresión lineal, se ha determinado que, a pesar de su simplicidad, este modelo puede presentar limitaciones al explorar relaciones y efectuar predicciones complejas. Por lo tanto, siguiendo la decisión que se adoptó en la sección 6.2, se procederá con la implementación del algoritmo de red neuronal feedforward. Esta técnica, debido a su inherente complejidad, es notablemente más competente en discernir y modelar relaciones intrincadas de manera precisa y eficaz.



En consecuencia, se procederá a la implementación de este modelo para determinar la relación entre el throughput y el MOS en los dos escenarios de red propuestos. A partir de los resultados obtenidos, se realizará una comparación con el modelo de regresión lineal previamente desarrollado. Dicha comparativa permitirá ilustrar las potenciales ventajas y desventajas que ambos modelos podrían presentar en este contexto en particular.

Es importante destacar que, en este caso, a diferencia de la regresión lineal, se ha decidido no eliminar los valores extremos o atípicos durante la implementación del modelo. La intención es reflejar una correlación lo más realista posible, la cual se ha observado que tiene una tendencia logarítmica en la realidad. Se pretende entrenar el modelo con todos los valores existentes, permitiéndole así aprender que el MOS tiende a estabilizarse con valores de throughput muy altos. Esta decisión se fundamenta en la necesidad de capturar completamente las dinámicas inherentes a los datos y de permitir que el modelo aprenda y se adapte a estas dinámicas de la forma más precisa posible.

Esta decisión se toma en el caso de la red neuronal debido a su habilidad para aprender y modelar relaciones no lineales y complejas, a diferencia de los modelos más simples como la regresión lineal. Esta capacidad de las redes neuronales las hace más adecuadas para manejar datos en los que existen relaciones no lineales y complejas, como es el caso en esta situación.

A continuación, se describen los pasos llevados a cabo para la construcción del modelo de la red neuronal feedforward.

### División de datos

Evitando la redundancia en la descripción de los procesos, la división de los datos se llevará a cabo de la misma manera que se efectuó para el modelo de regresión lineal, lo cual se puede revisar en el apartado 7.1.1.5.

### Construcción del modelo

La construcción de un modelo en redes neuronales se basa en la estructura y el funcionamiento del cerebro humano, específicamente en las neuronas y las conexiones que se encuentran en él. Una red neuronal artificial es un modelo computacional inspirado en este sistema biológico, diseñado para procesar información y realizar tareas de aprendizaje y predicción. Se elige una estructura de red neuronal bastante sencilla, ya que los datos disponibles son de tamaño reducido y el problema a solucionar es bastante directo. Esta elección favorece la eficiencia y la optimización del proceso de aprendizaje de la red. A continuación, se detalla el proceso de construcción del modelo:

```
modelo = Sequential()
modelo.add(Dense(64, input_dim=1, activation='relu'))
modelo.add(Dense(32, activation='relu'))
modelo.add(Dense(1))
```

- 1. Sequential():** Esta línea de código crea un modelo secuencial (feedforward), que es la forma más común de construir una red neuronal. Un modelo secuencial es una pila



lineal de capas donde la salida de una capa se convierte en la entrada de la siguiente capa.

2. `modelo.add(Dense(64, input_dim=1, activation='relu'))`: Aquí se añade a la red neuronal una capa densa (fully-connected) con 64 neuronas. Esta capa se conecta a la capa de entrada, la cual espera una única dimensión de entrada. Se utiliza la función de activación 'relu' (Rectified Linear Unit), la cual es muy usada en redes neuronales debido a sus propiedades de introducir no linealidades y su eficiencia computacional.
3. `modelo1.add(Dense(32, activation='relu'))`: Esta línea de código añade otra capa densa con 32 neuronas a la red neuronal. Se sigue utilizando la función de activación 'relu'. Al no especificar `input_dim`, esta capa se conecta automáticamente a la capa anterior, recibiendo como entrada sus 64 salidas.
4. `modelo.add(Dense(1))`: se añade la capa de salida con una sola neurona. Esta capa no tiene función de activación ya que se trata de un problema de regresión donde se busca predecir un valor numérico continuo (MOS).

### Compilación del modelo

La compilación en redes neuronales se refiere al paso en el que se configuran los aspectos adicionales del modelo antes de iniciar el proceso de entrenamiento. En este paso, se especifica el optimizador, la función de pérdida y las métricas de evaluación que se utilizarán durante el entrenamiento y la evaluación del modelo.

1. **Optimizador**: El optimizador es un algoritmo que se encarga de ajustar los pesos y los biases de las neuronas durante el entrenamiento de la red neuronal. El objetivo del optimizador es minimizar la función de pérdida o de error, que cuantifica la diferencia entre las predicciones del modelo y los valores reales del objetivo. En este caso, se utiliza el optimizador Adam, que es un optimizador popular y eficiente que se adapta dinámicamente a la tasa de aprendizaje durante el entrenamiento.
2. **Función de pérdida**: La función de pérdida o de error es una medida que cuantifica la discrepancia entre las predicciones del modelo y los valores reales del objetivo. En este caso, se utiliza la función de pérdida 'mean\_squared\_error', que calcula el error cuadrático medio entre las predicciones y los valores reales. El objetivo es minimizar esta función de pérdida durante el entrenamiento para que las predicciones del modelo se acerquen lo más posible a los valores reales.
3. **Métricas de evaluación**: Las métricas de evaluación son medidas adicionales que se utilizan para evaluar el rendimiento del modelo durante el entrenamiento y la evaluación. En este caso, se utiliza la métrica 'accuracy' o precisión para evaluar la precisión del modelo en la clasificación de datos. Sin embargo, en el caso de la regresión, la precisión no es relevante, por lo que esta métrica puede no ser significativa en el contexto específico de la regresión.

A continuación, se presenta el código utilizado para la compilación del modelo:

```
modelo.compile(optimizer=Adam(learning_rate=0.001), loss=
'mean_squared_error', metrics = 'MAE')
```



En esta línea de código, se lleva a cabo la compilación del modelo neuronal. Se especifica el optimizador utilizando el algoritmo Adam con una tasa de aprendizaje de 0.001. La función de pérdida se establece como `'mean_squared_error'`, que es una medida comúnmente utilizada para problemas de regresión donde se busca minimizar el error cuadrático medio entre las predicciones del modelo y los valores reales. Además, se elige la métrica MAE como una medida adicional para evaluar el modelo y complementar la evaluación basada en la pérdida de error cuadrático medio.

### Entrenamiento del modelo

El entrenamiento en un modelo de red neuronal consiste en ajustar los pesos y los “biases” de las neuronas para que el modelo aprenda a realizar predicciones precisas. Durante el entrenamiento, para cada lote de datos, se realiza una pasada hacia adelante en la red neuronal para obtener las predicciones, se calcula la función de pérdida para medir la discrepancia entre las predicciones y los valores objetivo, y luego se realiza una pasada hacia atrás utilizando el algoritmo de retropropagación (backpropagation) del error para actualizar los pesos y los “biases” de las neuronas. Este proceso se repite para cada época hasta que se alcanza el número especificado de épocas.

El objetivo del entrenamiento es encontrar los mejores valores para los pesos y los “biases” de la red neuronal que minimicen la función de pérdida y permitan hacer predicciones precisas sobre datos no vistos. A medida que el modelo se entrena, se espera que vaya mejorando su rendimiento y su capacidad para generalizar a partir de los datos de entrenamiento hacia datos nuevos.

A continuación, se muestra el código utilizado para este propósito:

```
modelo.fit(X_entrenamiento, y_entrenamiento, epochs=100,  
batch_size=32)
```

Durante este proceso, el parámetro **epochs** se establece en 100, lo que significa que el modelo realizará 100 pasadas completas a través de todo el conjunto de datos de entrenamiento. Cada pasada consta de una fase de propagación hacia adelante (forward pass) y una fase de retropropagación del error (backward pass), en la que se ajustan los pesos y los sesgos de las neuronas.

El parámetro **batch\_size** se utiliza para dividir el conjunto de datos de entrenamiento en lotes más pequeños. En lugar de procesar todo el conjunto de datos de entrenamiento en una sola vez, los lotes permiten que el modelo ajuste sus parámetros de manera más eficiente. En este caso, se ha establecido un tamaño de lote de 32, lo que significa que se toman 32 ejemplos de entrenamiento a la vez para calcular las actualizaciones de los pesos. Posteriormente, los pesos se actualizan después de cada lote, en lugar de después de cada ejemplo individual, lo que agiliza el proceso de entrenamiento.

Por último, una vez que el modelo ha sido entrenado, se puede utilizar para hacer predicciones en nuevos datos.



### 7.1.1.8. Evaluación del modelo: Red neuronal feedforward

Para evaluar la precisión del modelo de red neuronal, se han utilizado dos métricas de error comúnmente adoptadas en el campo del ML para problemas de regresión: el Error Cuadrático Medio y el Error Absoluto Medio. Así, podrá compararse con mayor facilidad con el modelo de regresión lineal.

Los resultados de las métricas del modelo realizado son los siguientes:

	Loss (MSE)	MAE
<b>Facultad de Ingeniería</b>	0,210	0,318
<b>Facultad de educación</b>	0.226	0.370

Tabla 12: Métricas para la evaluación del modelo

Para la Facultad de Ingeniería, los valores obtenidos fueron un MSE de 0,210 y un MAE de 0,318, mientras que, para la Facultad de Educación, el MSE y el MAE fueron de 0,226 y 0,370 respectivamente. Estos resultados son mejores a los obtenidos con el modelo de regresión lineal. Esto es esperable ya que las redes neuronales tienen a aprender mejor las relaciones complejas entre las variables.

Ahora, al igual que el caso anterior, se construyen una serie de gráficos de dispersión, los cuales ofrecen una representación visual de las relaciones entre las variables estudiadas (throughput y MOS) en ambas facultades (reales y predichas), así como de las discrepancias identificadas entre ellas. Estos gráficos proporcionan una perspectiva intuitiva y accesible de los resultados analíticos obtenidos. A continuación, se muestran los gráficos generados.

- **Facultad de Ingeniería:**

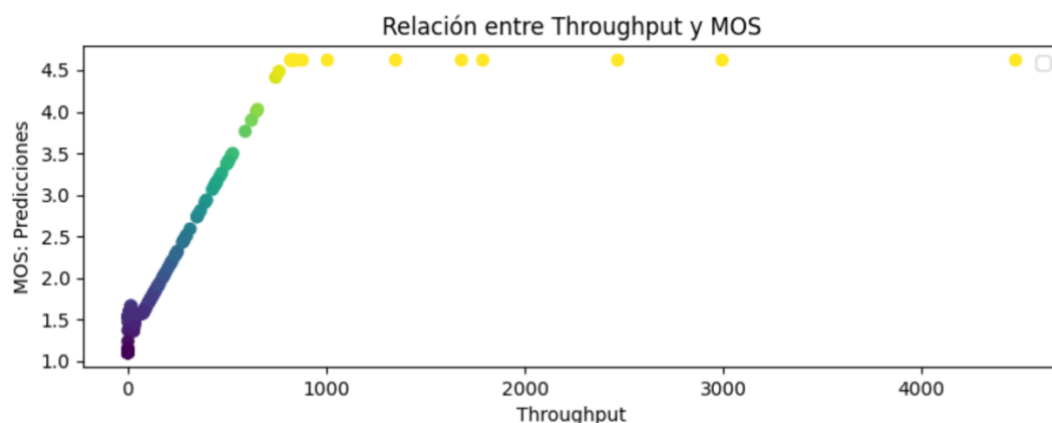


Ilustración 17 Throughput vs. MOS: predicciones

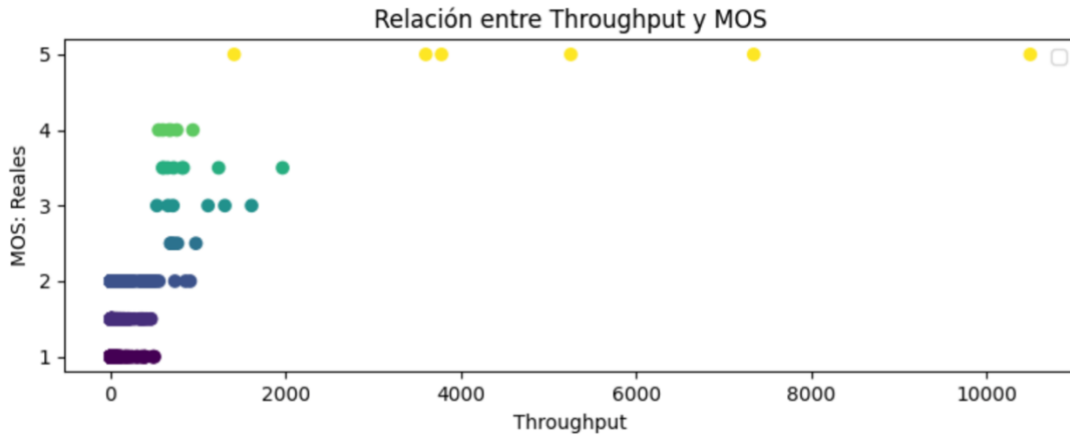


Ilustración 18: Throughput vs. MOS: valores reales

- Facultad de educación:

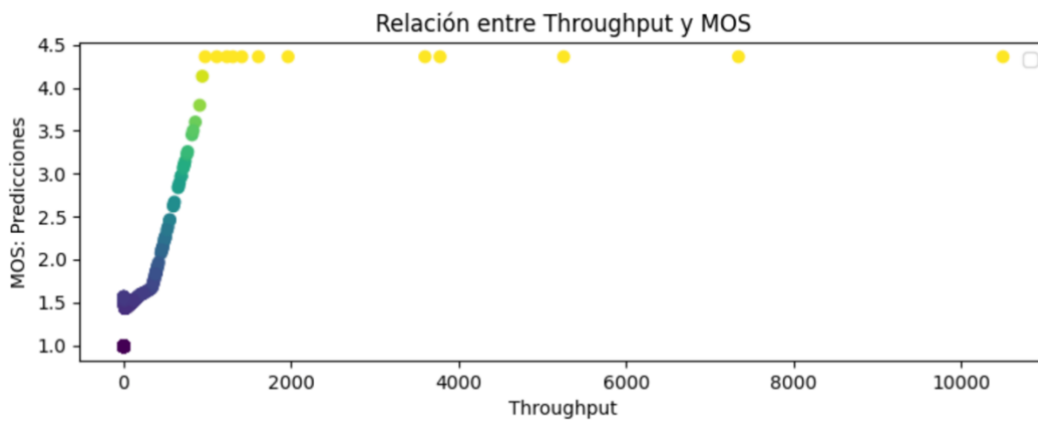


Ilustración 19: Throughput vs. MOS: predicciones

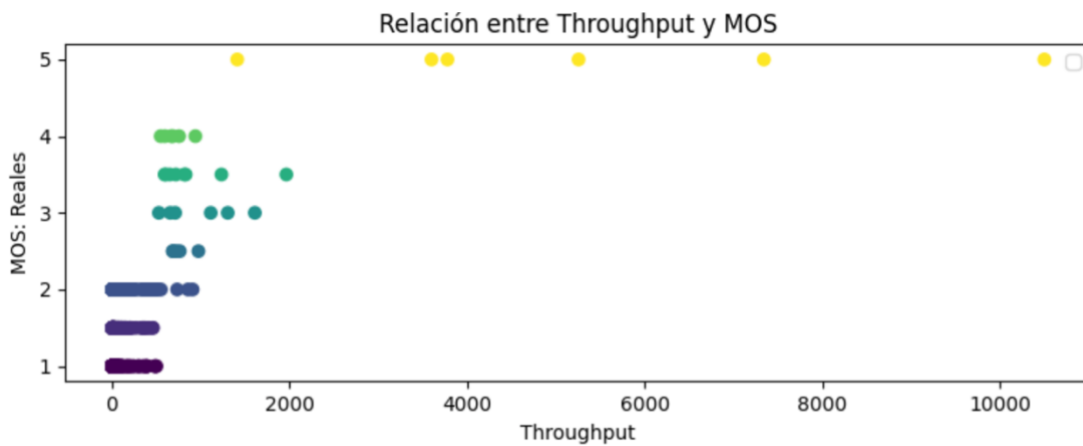


Ilustración 20: Throughput vs. MOS: valores reales

Comparando estos gráficos con los obtenidos para el modelo de regresión lineal, sí que se observan diferencias en los gráficos obtenidos.



Reiterando algunas observaciones realizadas en el caso del modelo anterior, aquí también se observa una correlación positiva entre estas dos variables. Sin embargo, es importante tener en cuenta que una red neuronal ha tenido la capacidad de identificar y aprender patrones más complejos en los datos. En este caso específico, la red neuronal ha podido capturar **patrones logarítmicos**, como se puede apreciar en la Ilustración 17 y Ilustración 19. Esto significa que el modelo ha aprendido una relación no lineal entre las variables de throughput y MOS. A medida que el valor de throughput sigue subiendo, a partir de ciertos valores, el valor de MOS se estanca en el valor máximo. Esto es una característica lógica y representa un comportamiento deseado, ya que indica que incluso cuando el rendimiento sigue mejorando, el puntaje de calidad percibida ya ha alcanzado su límite máximo.

La capacidad de capturar patrones logarítmicos es una de las ventajas de las redes neuronales en comparación con modelos lineales más simples. Estas redes son capaces de aprender relaciones no lineales en los datos y capturar efectos más complejos. Al entrenar el modelo con valores extremos, se le permite aprender y generalizar estos patrones logarítmicos en el conjunto de datos.

En este caso, también es conveniente analizar las discrepancias entre las dos facultades. A continuación, se muestran dos gráficos de dispersión que comparan las dos facultades, tanto para los valores reales y los predichos.

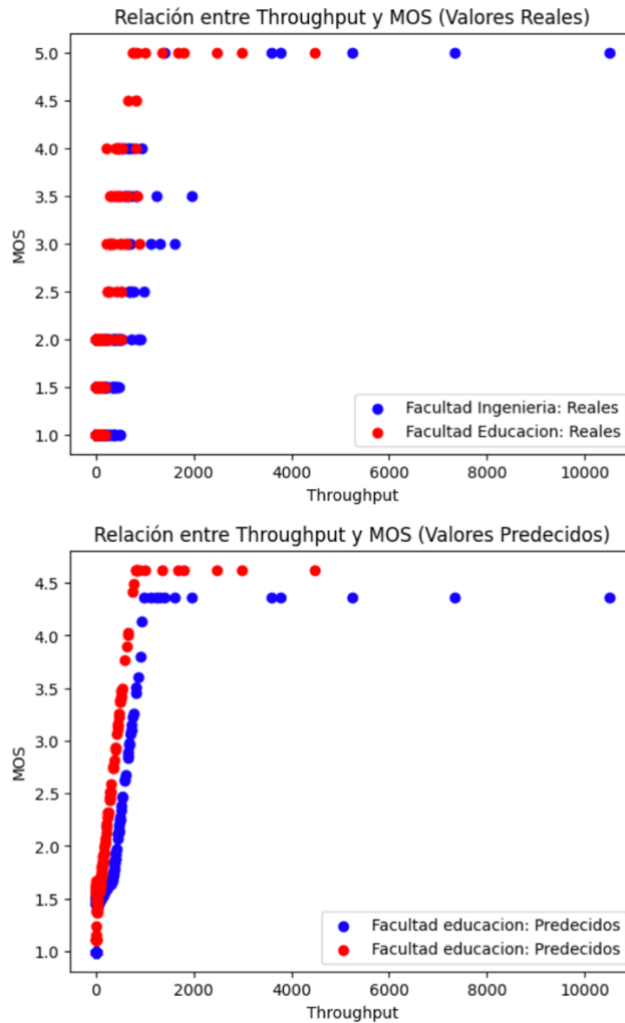


Ilustración 21: Diferencias entre las dos facultades: valores reales y predichos

Como se puede apreciar en los gráficos, se detectan las mismas discrepancias en las puntuaciones del MOS entre las dos facultades. Esta diferencia se atribuye principalmente a las expectativas más elevadas de los estudiantes de ingeniería respecto a la calidad de la red, las cuales están vinculadas a la naturaleza de sus actividades diarias.

## 7.2. Análisis de resultados

Por último, se realiza un análisis de los resultados obtenidos a partir de la correlación de QoS/QoE y de las puntuaciones MOS. Se presentan los hallazgos significativos y se extraen conclusiones relevantes en relación con la satisfacción de los usuarios y los indicadores de calidad identificados. Se discuten posibles implicaciones y se resaltan los aspectos más relevantes para la gestión de la QoE en los entornos universitarios estudiados.

El propósito principal de este proyecto era establecer una correlación entre un parámetro objetivo y las calificaciones MOS derivadas de encuestas a usuarios. Sin embargo, debido a ciertas complicaciones en relación con temas éticos en la realización de estas encuestas, no se pudieron llevar a cabo. Por lo tanto, como se explicó previamente, con base en





varios estudios que presentaban datos de throughput y MOS, se tomó la decisión de simular los datos MOS de la forma más realista posible. Este enfoque permitió desarrollar el modelo tal como se ha descrito. Es importante tener en cuenta la aleatoriedad inherente debido a los diversos factores de influencia (detallados en la sección 5.5) que pueden afectar la experiencia de los usuarios. Incluso si la calidad de la red en términos de throughput es alta, la experiencia del usuario puede no ser necesariamente satisfactoria, subrayando la complejidad y la naturaleza de las percepciones de los usuarios respecto a la calidad de la red.

No obstante, se ha establecido una correlación bastante sólida en ambos escenarios de red entre el throughput y las puntuaciones MOS simuladas. Esto concluye que, en general, **a medida que el throughput aumenta, las puntuaciones MOS tienden a mejorar**. Este patrón se modeló inicialmente utilizando el algoritmo de regresión lineal, que permite identificar de forma efectiva relaciones lineales entre la variable independiente (throughput) y la variable objetivo (MOS). Posteriormente, este patrón fue replicado con el algoritmo de red neuronal feedforward. A diferencia de la regresión lineal, las redes neuronales feedforward tienen la capacidad de capturar y modelar relaciones más complejas, gracias a su estructura de múltiples capas y nodos interconectados. La realización de este modelo permite predecir la experiencia de los usuarios únicamente a partir del dato objetivo, el throughput. De esta forma, se pueden tomar decisiones más informadas y adecuadas para mejorar la calidad de la red.

Como se mencionó anteriormente, en el caso de la regresión lineal se captura una relación lineal entre el MOS y el throughput. Esto no es totalmente incorrecto, ya que, dentro de un rango de valores, se puede considerar que la relación es lineal, es decir, a medida que el throughput aumenta, el MOS también aumenta de manera proporcional. Sin embargo, al utilizar una red neuronal feedforward, se ha logrado obtener una relación más logarítmica entre las variables.

La utilización de una red neuronal feedforward ha permitido obtener un modelo con un entrenamiento más preciso y una mayor capacidad para aprender relaciones complejas. Esto se debe a que las redes neuronales tienen la capacidad de capturar patrones más allá de las relaciones lineales simples. El entrenamiento de la red neuronal con valores extremos ha sido clave para capturar esta relación logarítmica. Al proporcionar al modelo datos con valores más altos de throughput, se le ha permitido aprender y generalizar los patrones logarítmicos presentes en el conjunto de datos.

Sin embargo, es evidente que siempre hay margen para mejorar la precisión del modelo y reducir los errores. Una forma prometedora de validar y mejorar el modelo en el futuro será mediante la implementación de encuestas directas a los usuarios.

La recopilación de datos personales y detalles específicos sobre la red utilizada a través de encuestas permitirá obtener una comprensión más profunda de la tipología de usuarios en cada facultad y revelar diferencias en sus expectativas, usos y otros factores relevantes. Esto será fundamental para interpretar de forma más precisa la correlación entre la QoS y la QoE. Al tener acceso a un conjunto de datos relacionados con la tipología de usuarios y su puntuación respectiva en cuanto a la calidad de la red, será posible elaborar soluciones más específicas y orientadas a mejorar la experiencia del usuario.



Al comprender mejor las necesidades y expectativas individuales de los usuarios, se podrán diseñar soluciones más eficaces para aumentar la satisfacción del usuario y mejorar la calidad de la experiencia de red. La implementación de encuestas directas aportará información valiosa y permitirá realizar ajustes más precisos en el modelo existente, garantizando así una aproximación más realista y efectiva en la evaluación y mejora de la calidad de la red.

Resumiendo todo lo discutido, es crucial destacar que, el modelo desarrollado a partir de las puntuaciones MOS simuladas consigue establecer una correlación significativa con el parámetro objetivo, el throughput de la red. Este resultado refuerza la idea de que la calidad objetiva de la red tiene un impacto directo en la experiencia del usuario, aunque el grado de este impacto puede variar dependiendo de diversos factores de contexto.

El modelo revela que a medida que mejora la calidad de la red (mayor throughput), la calidad de la experiencia del usuario tiende a mejorar en ambos escenarios, aunque los usuarios de la Facultad de Educación parecen ser algo menos exigentes. Este hallazgo es de gran valor para los proveedores de red, ya que con un modelo de este tipo pueden predecir la satisfacción general de cada facultad, analizando únicamente los parámetros objetivos de la red y haciendo las mejoras necesarias si se requiere. Esto permitiría optimizar la gestión de los recursos de red y mejorar la experiencia del usuario de manera más eficiente.

En conclusión, se puede sintetizar la relevancia y utilidad de este modelo, así como los hallazgos significativos obtenidos, en los siguientes puntos clave:

1. **Identificación de relaciones:** Este modelo permite identificar y cuantificar la correlación existente entre el parámetro objetivo (en este caso, el throughput de la red) y las puntuaciones MOS de los usuarios.
2. **Evaluación de la calidad de la red:** Dada la relación identificada entre el throughput y las puntuaciones MOS, este modelo puede servir como herramienta para evaluar la calidad de la red desde la perspectiva de la experiencia del usuario.
3. **Predicción de la satisfacción del usuario:** A partir de los parámetros objetivos de la red, el modelo puede predecir en términos generales la satisfacción del usuario, lo que puede ser útil para ajustar y mejorar la calidad de la red.
4. **Comprensión de las diferencias entre grupos de usuarios:** Al analizar las diferencias en las puntuaciones MOS entre las dos facultades, este modelo también proporciona información sobre cómo diferentes grupos de usuarios pueden experimentar y valorar la red de maneras diferentes, probablemente debido a diferencias en sus necesidades y expectativas.
5. **Mejora de la toma de decisiones:** Con la información proporcionada por este modelo, los proveedores de red pueden tomar decisiones más informadas sobre dónde y cómo mejorar la red para mejorar la experiencia del usuario.



### 7.3. Descripción de tareas

En el presente proyecto se han definido diferentes paquetes de trabajo, cada uno con una serie de tareas que permitirán alcanzar los objetivos planteados. A continuación, se describe brevemente cada uno de los paquetes de trabajo definidos, incluyendo las tareas integradas, la carga de trabajo, la persona responsable y los recursos técnicos.

#### 1. Definición de los objetivos

En este paquete de trabajo se definirá el alcance y los objetivos del proyecto, se establecerán los requerimientos y se definirán los entregables del TFM.

- Tareas:
  - Identificación de los objetivos y alcance del proyecto.
  - Establecimiento de los requerimientos del proyecto.
  - Definición de las etapas a seguir para favorecer el buen desarrollo de la solución.
- Recursos humanos: directora del proyecto, proyectista.
- Duración: 10 días.

#### 2. Estado del arte

En esta fase del trabajo se busca y se recopila información relevante para conocer las diferentes tecnologías que se van a utilizar para desarrollar el proyecto.

- Tareas:
  - Realización de una revisión bibliográfica sobre las técnicas de ML más adecuadas para la correlación de la QoS/QoE.
  - Investigación de los diferentes entornos de desarrollo y lenguajes de programación existentes para el desarrollo de proyectos de ML.
  - Investigación de los parámetros de QoS objetivos y los factores que influyen en la percepción de los usuarios de las redes WiFi.
  - Investigación de metodologías de valoración para capturar la experiencia de los usuarios en el uso de redes WiFi.
- Recursos humanos: proyectista.
- Duración: 25 días

#### 3. Análisis de alternativas

Comparación de las tecnologías consideradas y selección de las más adecuadas para el desarrollo de la solución propuesta.

- Tareas:
  - Identificación de diferentes alternativas de solución.
  - Análisis de la viabilidad y eficacia de las alternativas.
  - Selección de las alternativas más adecuadas para el proyecto.
- Recursos humanos: proyectista.
- Duración: 40 días



#### 4. Familiarización con el entorno de desarrollo

En este paquete de trabajo se familiarizará con las herramientas y entorno de desarrollo que se van a utilizar para el desarrollo del proyecto.

- **Tareas:**
  - Familiarización con los entornos de desarrollo.
  - Familiarización con algoritmos y herramientas para la utilización del ML.
  - Familiarización con los conjuntos de datos de los dos escenarios para el estudio.
  - Identificación de posibles perfiles de usuario y servicios para el estudio.
- **Recursos humanos:** proyectista.
- **Duración:** 35 días

#### 5. Simulación de los resultados MOS

En esta fase del trabajo se realiza la simulación de los resultados aproximados de MOS que proporcionarían los alumnos de ambas facultades.

##### Tareas:

- Realización de un estudio de resultados de la Calidad de Experiencia (MOS) basados en parámetros objetivos de la red en entornos similares a los de este estudio.
- Identificar los parámetros de red (KQI y KPI) que sean relevantes para la experiencia de los usuarios en la red.
- Establecer relaciones entre las variables objetivas y la experiencia de los usuarios.
- Diseño de una función que permita simular los resultados MOS.
- **Recursos humanos:** proyectista.
- **Duración:** 20 días

#### 6. Diseño del modelo de correlación QoS/QoE

En este paquete de trabajo se diseñará el modelo que permita correlacionar la QoS y la QoE en los casos de estudio planteados.

- **Tareas:**
  - Captura de resultados objetivos de la red WiFi y resultados subjetivos mediante la simulación de los MOS.
  - Diseño y desarrollo del modelo de correlación QoS/QoE.
- **Recursos humanos:** proyectista
- **Duración:** 25 días.

#### 7. Análisis de resultados

En este paquete de trabajo se analizarán los resultados obtenidos en el estudio de la correlación de los parámetros QoS y QoE.

- **Tareas:**
  - Validación del modelo propuesto mediante la comparación de los resultados obtenidos en los casos de estudio.
  - Conclusiones del estudio.
- **Recursos humanos:** proyectista.



- **Duración:** 7 días.

## 8. Documentación

En este paquete de trabajo se redactará la memoria del TFM, se elaborarán los manuales necesarios y se preparará la presentación del proyecto.

- **Tareas:**
  - Redacción de la memoria del TFM.
  - Elaboración de los manuales necesarios.
- **Recursos humanos:** proyectista
- **Duración:** 140 días.

## 7.4. Diagrama de Gantt

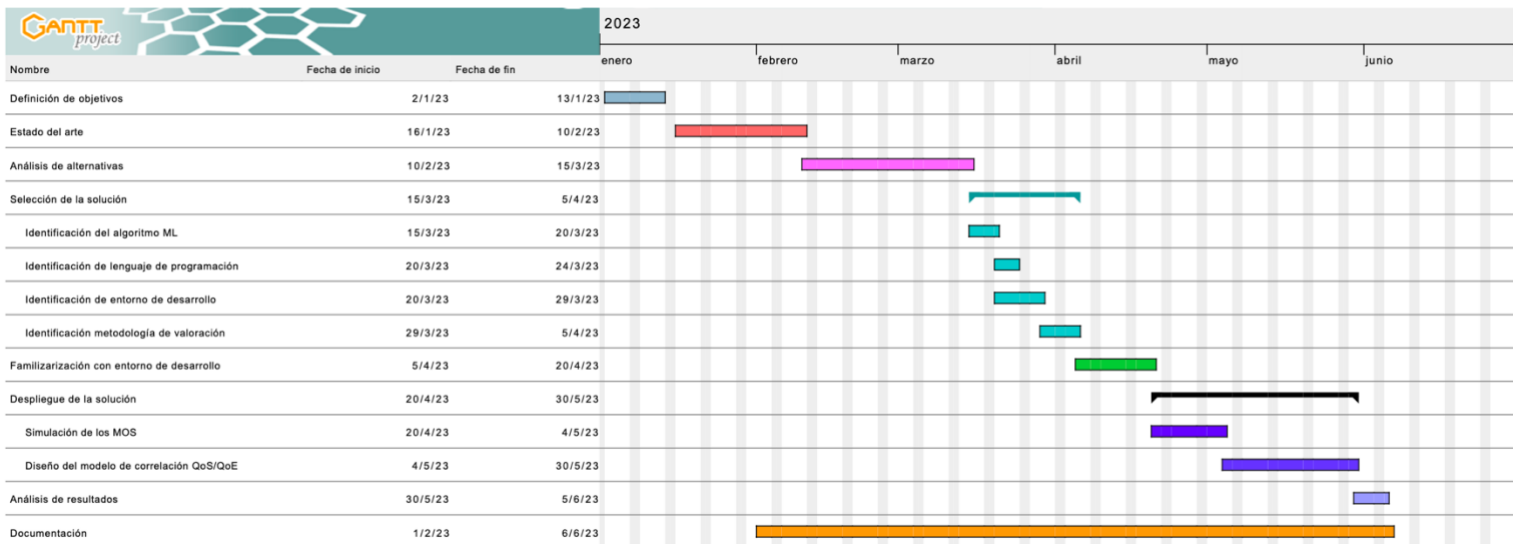


Ilustración 22: Diagrama de Gantt



## 8. ASPECTOS ECONÓMICOS

En este capítulo, se abordará una de las dimensiones fundamentales de cualquier proyecto, los aspectos económicos, que juegan un papel esencial en el marco de este trabajo. Este análisis proporciona un entendimiento más profundo de la viabilidad económica y financiera del proyecto, y su consideración es imprescindible para cualquier toma de decisiones futura. Los costos directos del proyecto se dividen en tres grupos: horas internas, amortizaciones y gastos.

- **Horas internas:** se definen como las horas de los miembros del equipo de trabajo para la realización del proyecto.
- **Amortizaciones:** los activos fijos que se utilizan para el proyecto.
- **Gastos:** costes asociados a lo utilizado en el proyecto y que no puede ser utilizado posteriormente.

### 8.1. Horas internas

Concepto	Número de horas (h)	Coste horario (€/h)	Coste total (€)
Directora del proyecto	30	50	1.200,00
Proyectista	600	20	12.000,00

Tabla 13: Presupuesto correspondiente a las horas internas

### 8.2. Amortizaciones

Concepto	Coste	Vida útil	Uso (meses)	Coste total (€)
MacBook Pro	1.400	60	6	116,6

Tabla 14: Presupuesto correspondiente a las amortizaciones

### 8.3. Gastos

Concepto	Coste unitario (€)	Número de unidades	Coste total (€)
Fungibles	-	-	26

Tabla 15: Presupuesto correspondiente a los gastos

### 8.4. Resumen del coste del proyecto

El análisis de costes del presente TFM detalla los costes asociados a la ejecución del proyecto. A continuación, se proporciona un desglose de los costes totales, que suman un importe de 7.345,16 euros.

Concepto	Coste total (€)
Subtotal horas internas	13.200
Subtotal amortizaciones	116,16
Subtotal de gastos	26
<b>TOTAL</b>	<b>13.342,16</b>

Tabla 16: Resumen de los costes del proyecto



Analizando las tablas de los costes del proyecto, el costo más significativo proviene de las "horas internas", valorado en €13.200. Este costo refleja el tiempo y esfuerzo dedicado al proyecto por parte de los investigadores. La relevancia de este gasto subraya la importancia del componente humano en la realización del TFM, siendo el tiempo y esfuerzo invertidos en la investigación y desarrollo del proyecto los factores más cruciales.

En cuanto a las "amortizaciones", que ascienden a €116,16, este valor representa la depreciación de los activos tangibles e intangibles utilizados en el proyecto. En este caso, se refiere al ordenador utilizado para la realización del proyecto.

Por otro lado, se han considerado "gastos adicionales", que suman un total de €26. Este monto cubre gastos diversos incurridos durante el desarrollo del proyecto. Aunque la cifra pueda parecer pequeña en comparación con los otros costos, refleja la importancia de tener en cuenta todos los gastos adicionales que pueden surgir en el transcurso de la investigación.

Al sumar todos estos componentes, se obtiene el coste total del proyecto, que asciende a **€13.342,42**. Este valor proporciona una visión completa y transparente del presupuesto requerido para la realización de este TFM. Esta suma total se convierte en una base significativa para futuras comparaciones y análisis de costos en proyectos similares.



## 9. ANÁLISIS DE RIESGOS

El análisis de riesgos tiene como objetivo aumentar la probabilidad y el impacto de los eventos positivos y disminuir la probabilidad e impacto de los eventos negativos que puedan afectar el desarrollo y los resultados del proyecto. A continuación, se presentan los posibles riesgos para este proyecto junto con las estrategias de mitigación para cada uno.

### 9.1. Demoras (A)

En el desarrollo de cualquier proyecto, pueden surgir retrasos imprevistos que amenacen la finalización del mismo en los plazos establecidos inicialmente. Por ende, es fundamental implementar una planificación estratégica donde se delinee de manera precisa las tareas a llevar a cabo, así como las fechas estimadas de finalización para cada una de ellas.

La probabilidad de que surjan tales retrasos en este proyecto es relativamente baja, gracias a la guía continúa proporcionada por la directora del proyecto. Esta supervisión constante asegura que las tareas se identifiquen y se sigan correctamente, permitiendo un seguimiento regular de su progreso.

Además, es importante señalar que el alcance de este TFM no es particularmente extenso, lo que disminuye aún más la posibilidad de que surjan imprevistos significativos que puedan impactar el progreso del proyecto de manera considerable.

### 9.2. Fallo tecnológico (B)

El ordenador es una herramienta esencial en la elaboración de la solución propuesta para este TFM. Todo el desarrollo del modelo, desde su concepción hasta su implementación y pruebas, se realiza utilizando esta herramienta tecnológica. A pesar de que los ordenadores generalmente pasan por rigurosos controles de calidad y cumplen con estándares elevados, la probabilidad de una falla o mal funcionamiento del hardware no puede descartarse completamente, catalogándola como de riesgo medio.

Sin embargo, el impacto de una potencial falla del ordenador se ve significativamente mitigado gracias al uso de herramientas basadas en la nube, como Google Colaboratory (Colab). Este enfoque de trabajo en la nube ofrece un nivel adicional de seguridad y conveniencia. Cualquier trabajo realizado se guarda automáticamente en la nube, y, en caso de un fallo del ordenador, se puede acceder a él y continuar trabajando desde cualquier otro dispositivo con acceso a Internet. Esto no solo reduce el riesgo de pérdida de datos, sino que también minimiza la interrupción del flujo de trabajo, ya que se puede retomar rápidamente en otro dispositivo.

### 9.3. Demoras en la aprobación de encuestas (C)

En cualquier proyecto que involucre la recopilación de datos mediante encuestas, existe el riesgo de demoras causadas por la necesidad de obtener aprobación de comités éticos u otras entidades de supervisión. Estas demoras pueden alterar el cronograma del proyecto, causando retrasos significativos en la ejecución y finalización del mismo.





En el contexto de este TFM este riesgo se materializó en la fase de recopilación de datos. Como se ha comentado a lo largo del documento, la aprobación necesaria para llevar a cabo las encuestas para evaluar la QoE no se obtuvo a tiempo. El impacto es medio, ya que se logró implementar el modelo realizando una simulación de los resultados del MOS.

Para mitigar este tipo de riesgo, es fundamental anticipar y planificar potenciales demoras en la fase de planificación del proyecto. Esto puede incluir iniciar el proceso de solicitud de aprobación con suficiente antelación, tener un plan B para la recopilación y análisis de datos en caso de retrasos, y mantener una comunicación fluida con todas las partes externas involucradas en el proyecto.

Además, en el caso de que se produzcan retrasos inesperados, es útil contar con estrategias de contingencia, como la simulación de datos, para garantizar que el proyecto pueda continuar avanzando a pesar de estos obstáculos.

#### 9.4. Falta de datos suficientes (D)

El riesgo de falta de datos suficientes se refiere a la posibilidad de no poder recopilar una cantidad adecuada de datos de QoS y QoE que serían necesarios para entrenar de manera efectiva los modelos de ML. Este riesgo es particularmente relevante en proyectos basados en ML, ya que la calidad y cantidad de datos disponibles para el entrenamiento puede tener un impacto directo en la precisión y eficacia del modelo resultante.

La probabilidad de ocurrencia de este riesgo puede ser moderada, ya que a menudo depende de factores que pueden estar fuera del control del equipo del proyecto, como la disponibilidad de fuentes de datos o la disposición de los usuarios a participar en las encuestas.

El impacto de la falta de datos suficientes puede ser significativo. Puede retrasar el desarrollo del proyecto, ya que se tendría que buscar y recopilar datos adicionales, y puede comprometer la calidad de los resultados, ya que un modelo de ML entrenado con datos insuficientes puede tener una precisión y eficacia más baja.

#### 9.5. Matriz probabilidad – impacto

La gestión efectiva de los riesgos se lleva a cabo a través de una herramienta estratégica conocida como la matriz de probabilidad-impacto. En esta matriz, se catalogan y visualizan de manera clara todos los riesgos previamente mencionados. Esta matriz facilita la comprensión de los riesgos, permite evaluar su potencial efecto en el proyecto, y proporciona una base sólida para establecer prioridades en la mitigación de riesgos.

PROBABILIDAD		IMPACTO		
		BAJO 0,2	MEDIO 0,5	ALTO 0,8
BAJO 0,2	BAJO 0,2	Bajo 0,04	Bajo 0,1 (B)	Moderado 0,16
	MEDIO 0,5	Bajo 0,1	Moderado 0,25 (A) (D)	Alto 0,4
	ALTO 0,8	Moderado 0,16	Alto 0,4 (C)	Alto 0,64

Tabla 17: Matriz probabilidad-impacto



## 10. CONCLUSIONES

El objetivo principal de este proyecto fue desarrollar un modelo que correlacione la QoS y QoE utilizando técnicas de ML en entornos universitarios específicos. Este objetivo no solo ha sido alcanzado, sino que ha aportado una serie de beneficios y avances significativos en el campo de estudio.

En primer lugar, a través de este TFM, se ha llevado a cabo un estudio detallado y exhaustivo de los parámetros de QoS y los indicadores de calidad (KQI) que influyen en la percepción de los usuarios de las redes WiFi, específicamente en redes 802.11. La clasificación de estos aspectos y la identificación de los KQI más relevantes representan un avance importante en la comprensión y la gestión de la QoE.

Además, la implementación de técnicas de ML para correlacionar la QoS y la QoE ha demostrado ser una herramienta eficaz y poderosa. Esta aplicación innovadora de ML en la gestión de la QoE podría tener un impacto significativo en la mejora de la satisfacción del usuario en entornos universitarios y más allá.

Se tiene previsto realizar encuestas directas a los usuarios en un futuro próximo para recopilar datos reales y validar los resultados de la simulación. Estas encuestas permitirán obtener información más precisa y detallada sobre las experiencias de los usuarios y su percepción de la calidad de la red. La validación de los resultados de la simulación con datos reales será fundamental para mejorar y ajustar el modelo, asegurando así su precisión y utilidad en la evaluación y mejora de la calidad de la red.

En conclusión, este TFM no solo ha alcanzado sus objetivos, sino que también ha contribuido a avances significativos en la correlación de la QoS y la QoE en redes 802.11 utilizando técnicas de ML. Los resultados obtenidos pueden servir como base para futuras investigaciones y como recurso valioso para mejorar la QoE en entornos universitarios u otros entornos.

Aunque se han obtenido resultados prometedores, se reconoce la necesidad de continuar con la investigación y el desarrollo en este campo. Esto incluye la exploración de más técnicas de ML, la recopilación y análisis de más datos y la consideración de más factores que pueden influir en la QoS y la QoE. Con estos esfuerzos continuos, es posible que se pueda mejorar aún más la precisión y la eficacia del modelo y profundizar la comprensión de la correlación entre la QoS y la QoE.



## 11. BIBLIOGRAFÍA

- [1] «Machine learning. Tipos, modelos, técnicas y usos - IAT». <https://iat.es/tecnologias/inteligencia-artificial/machine-learning/> (accedido 18 de abril de 2023).
- [2] «Supervised Machine learning - Javatpoint». <https://www.javatpoint.com/supervised-machine-learning> (accedido 18 de abril de 2023).
- [3] «Unsupervised Machine learning - Javatpoint». <https://www.javatpoint.com/unsupervised-machine-learning> (accedido 18 de abril de 2023).
- [4] S. Bhatt, «Reinforcement Learning 101», *Medium*, 19 de abril de 2019. <https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292> (accedido 18 de abril de 2023).
- [5] «What is Deep Learning and How Does It Works [Updated]», *Simplilearn.com*. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning> (accedido 18 de abril de 2023).
- [6] R. Python, «Jupyter Notebook: An Introduction – Real Python». <https://realpython.com/jupyter-notebook-introduction/> (accedido 14 de abril de 2023).
- [7] «Google Colaboratory». <https://colab.research.google.com/?hl=es> (accedido 26 de mayo de 2023).
- [8] «JetBrains DataSpell: el IDE para ingenieros de datos», *JetBrains*. <https://www.jetbrains.com/es-es/dataspell/> (accedido 14 de abril de 2023).
- [9] «PyCharm: el IDE de Python para desarrolladores profesionales, por JetBrains», *JetBrains*. <https://www.jetbrains.com/es-es/pycharm/> (accedido 14 de abril de 2023).
- [10] «The Python Tutorial», *Python documentation*. <https://docs.python.org/3/tutorial/index.html> (accedido 14 de abril de 2023).
- [11] «R: What is R?» <https://www.r-project.org/about.html> (accedido 14 de abril de 2023).
- [12] «What is Java? | IBM». <https://www.ibm.com/topics/java> (accedido 14 de abril de 2023).
- [13] «C++ Tutorial | Learn C++ Programming - javatpoint», *www.javatpoint.com*. <https://www.javatpoint.com/cpp-tutorial> (accedido 14 de abril de 2023).
- [14] «Factors influencing quality of experience of commonly used mobile applications | IEEE Journals & Magazine | IEEE Xplore». <https://ieeexplore.ieee.org/document/6178833> (accedido 3 de mayo de 2023).
- [15] R. Stankiewicz y A. Jajszczyk, «A survey of QoE assurance in converged networks», *Comput. Netw.*, vol. 55, n.º 7, pp. 1459-1473, may 2011, doi: 10.1016/j.comnet.2011.02.004.
- [16] M. Matulin y Š. Mrvelj, «State-of-the-Practice in Evaluation of Quality of Experience in Real-Life Environments», *Promet - TrafficTransportation*, vol. 25, n.º 3, pp. 255-263, jun. 2013, doi: 10.7307/ptt.v25i3.1195. (accedido el 30 de abril de 2023)
- [17] «document.pdf». Accedido: 3 de junio de 2023. [En línea]. Disponible en: <https://hal.science/hal-00977812/document>
- [18] D. Tjondronegoro, *Mobile Multimedia: User and Technology Perspectives*. BoD – Books on Demand, 2012. (accedido el 15 de mayo de 2023)
- [19] «A Hierarchical Classification Model of QoE Influence Factors | SpringerLink». [https://link.springer.com/chapter/10.1007/978-3-319-22572-2\\_16#Sec5](https://link.springer.com/chapter/10.1007/978-3-319-22572-2_16#Sec5) (accedido 3 de junio de 2023).
- [20] «P.800 : Methods for subjective determination of transmission quality». <https://www.itu.int/rec/T-REC-P.800-199608-I/en> (accedido 24 de abril de 2023).
- [21] «P.910 : Subjective video quality assessment methods for multimedia applications».



<https://www.itu.int/rec/T-REC-P.910/en> (accedido 24 de abril de 2023).

[22] K. MALI, «Everything you need to Know about Linear Regression!», *Analytics Vidhya*, 4 de octubre de 2021. <https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/> (accedido 25 de mayo de 2023).

[23] S. Swaminathan, «Logistic Regression — Detailed Overview», *Medium*, 18 de enero de 2019. <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc> (accedido 30 de mayo de 2023).

[24] «Qué es Red neuronal feedforward Concepto y definición. Glosario». <https://gamco.es/glosario/red-neuronal-feedforward/> (accedido 2 de mayo de 2023).

[25] «pt406.pdf». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: <https://www.imt.mx/archivos/Publicaciones/PublicacionTecnica/pt406.pdf>

[26] M. J. M. Ayora, «Clasificación de imágenes usando redes neuronales convolucionales en Python».

[27] «tfg\_Pau\_Agustin\_memoria.pdf». Accedido: 2 de mayo de 2023. [En línea]. Disponible en: [https://diposit.ub.edu/dspace/bitstream/2445/124249/1/tfg\\_Pau\\_Agustin\\_memoria.pdf](https://diposit.ub.edu/dspace/bitstream/2445/124249/1/tfg_Pau_Agustin_memoria.pdf)

[28] «Quality of Service in a Wireless Environment - Cisco». <https://www.cisco.com/c/en/us/td/docs/routers/access/wireless/software/guide/QualityOfService.html#wp1062496> (accedido 5 de junio de 2023).

[29] A. Vilaikruad, P. Anchuen, P. Uthansakul, y M. Uthansakul, «Effect of quality of service parameters on quality of experience for YouTube service in mobile networks», jun. 2017, pp. 568-571. doi: 10.1109/ECTICon.2017.8096301. (accedido el 10 de mayo de 2023)