COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

SISTEMA DE REUTILIZACIÓN DE AGUAS RESIDUALES Y PREDICCIÓN DE LA DEMANDA DE AGUA UTILIZANDO APLICACIONES DE INTELIGENCIA ARTIFICIAL

Autor: Javier González-Gancedo del Hoyo

Director: Jaime Navarro Ocón

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Sistema de reutilización de aguas residuales y predicción de la demanda de agua utilizando aplicaciones de inteligencia artificial

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

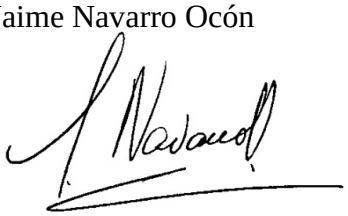
tomada de otros documentos está debidamente referenciada.

Fdo.: Javier González-Gancedo del Hoyo Fecha: 14/07/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Jaime Navarro Ocón Fecha: 21/07/2025



SISTEMA DE REUTILIZACIÓN DE AGUAS RESIDUALES Y PREDICCIÓN DE LA DEMANDA DE AGUA UTILIZANDO APLICACIONES DE INTELIGENCIA ARTIFICIAL

Autor: González-Gancedo del Hoyo, Javier.
Director: Navarro Ocón, Jaime.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

Este Trabajo Fin de Máster desarrolla un sistema integrado que combina tecnologías de reutilización de aguas residuales con modelos de inteligencia artificial para predecir la demanda hídrica. La solución propuesta optimiza la operación de plantas de tratamiento mediante un modelo predictivo desarrollado en TensorFlow, validado con simulaciones que incluyen escenarios extremos. Los resultados muestran el potencial de esta integración para mejorar la eficiencia hídrica y contribuir a la sostenibilidad.

Palabras clave: Reutilización de agua, inteligencia artificial, predicción de demanda, TensorFlow, sostenibilidad, red neuronal, eficiencia hídrica.

1. Introducción

La escasez de agua es uno de los retos ambientales más urgentes del siglo XXI. Frente a esta situación, la reutilización de aguas residuales y la predicción de la demanda mediante inteligencia artificial (IA) emergen como estrategias complementarias clave para una gestión hídrica eficiente y sostenible. Este proyecto propone una solución híbrida que integra ambas tecnologías en un sistema modular, escalable y adaptable a diversos entornos (urbanos, agrícolas, industriales).

2. Definición del proyecto

El objetivo del proyecto es diseñar una plataforma conceptual capaz de anticipar la demanda de agua regenerada y activar automáticamente los procesos de tratamiento necesarios. Esta integración permite ajustar la producción en función de la demanda prevista, optimizando el uso de recursos y minimizando pérdidas y vertidos.

3. Descripción del sistema y metodología

La arquitectura general del sistema está compuesta por cinco módulos funcionales:

- Entrada de datos: consumo histórico, meteorología, indicadores estacionales y normativos.
- Predicción de demanda: red neuronal desarrollada en TensorFlow.
- Planificación del tratamiento
- Sistema de control: automatización de los procesos de tratamiento.
- Salida: indicadores de eficiencia y cumplimiento normativo.

4. Modelo predictivo

Se ha implementado un modelo de red neuronal artificial con TensorFlow, entrenado con datos sintéticos simulados que reproducen variaciones estacionales de temperatura, precipitaciones y consumo de agua. El modelo utiliza capas densas con activaciones ReLU y optimizador Adam. Las simulaciones incluyen escenarios de alta demanda y condiciones extremas.

El entrenamiento del modelo ofrece resultados cuantificables en términos de:

- MAE (Error absoluto medio)
- RMSE (Raíz del error cuadrático medio)
- R^2 (Coeficiente de determinación)

5. Resultados

Con el objetivo de validar el rendimiento del modelo predictivo de demanda de agua regenerada, se ha diseñado una serie de simulaciones utilizando datos sintéticos que representan distintas condiciones climáticas y operativas. Se han entrenado redes neuronales densas con TensorFlow, evaluadas mediante tres métricas clave: MAE (Error Absoluto Medio), RMSE (Raíz del Error Cuadrático Medio) y R^2 (Coeficiente de Determinación).

5.1. Simulación base

El modelo inicial se entrena con un conjunto de datos que simula el comportamiento estacional y aleatorio del clima. Los resultados obtenidos muestran un rendimiento limitado:

- MAE: 12,58

- RMSE: 15,08
- R^2: −0,70

Este resultado refleja que el modelo no logra capturar bien la relación entre variables meteorológicas y consumo. Sirve como base funcional para desarrollar mejoras posteriores.

5.2. Escenarios extremos simulados

Se han probado seis escenarios adicionales para analizar la robustez del sistema:

Escenario	MAE	RMSE	R^2
Base	12,58	15,08	-0,7
Ola de calor	12,46	14,16	-0,11
Sequía extrema	12,82	14,97	0,04
Lluvias intensas localizadas	11,99	14,2	0,74
Restricción normativa	11,53	13,19	-0,37
Fuga oculta	12,49	14,5	-0,49

Figura I. Comparativa escenarios simulados

Estos resultados evidencian que el mejor rendimiento se logra bajo el escenario de lluvias intensas localizadas, donde el coeficiente R^2 alcanza 0,74, lo que indica un ajuste notable. El modelo explica correctamente la variabilidad del consumo, manteniendo errores bajos.

Por el contrario, el peor comportamiento aparece en el escenario base y el de fuga oculta, donde el modelo no logra anticipar cambios inesperados, lo que reduce su capacidad explicativa. A pesar de eso, los errores absolutos (MAE y RMSE) se mantienen

relativamente estables en todos los casos, lo que sugiere una cierta robustez estructural del modelo, aunque no siempre precisión contextual.

5.3. Análisis global

El modelo muestra estabilidad ante condiciones extremas, pero tiene limitaciones a la hora de capturar eventos anómalos o cambios estructurales no modelados. Para mejorar esto, se podría añadir nuevas variables (festivos, humedad, restricciones), o entrenar con más datos reales para mejorar la generalización.

En conclusión, el sistema predictivo tiene una base sólida, pero requiere optimización para su aplicación en contextos reales de planificación hídrica automatizada.

6. Conclusiones

El sistema propuesto demuestra la viabilidad de integrar IA en la gestión de aguas regeneradas. Aunque el modelo inicial es mejorable, las pruebas realizadas confirman su capacidad para anticipar la demanda hídrica y adaptar la operación de tratamiento de forma automatizada. Este enfoque puede contribuir significativamente al cumplimiento de los ODS 6, 11 y 13, ofreciendo una herramienta útil para municipios, operadores y empresas del ciclo del agua.

WASTEWATER REUSE SYSTEM AND WATER DEMAND FORECASTING USING ARTIFICIAL INTELLIGENCE APPLICATIONS

Author: González-Gancedo del Hoyo, Javier.
Supervisor: Navarro Ocón, Jaime.
Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

This Master's thesis presents an integrated system that combines wastewater reuse technologies with artificial intelligence to forecast water demand. The proposed solution optimizes treatment plant operation through a predictive model developed in TensorFlow, validated through simulations including extreme scenarios. The results demonstrate the potential of this approach to improve water efficiency and contribute to sustainability goals.

Keywords: Water reuse, artificial intelligence, demand forecasting, TensorFlow, sustainability, neural networks, water efficiency.

1. Introduction

Water scarcity is one of the most pressing environmental challenges of the 21st century. In this context, wastewater reuse and artificial intelligence (AI)-based demand prediction emerge as complementary strategies for efficient and sustainable water management. This project proposes a hybrid solution that integrates both technologies into a modular, scalable system adaptable to urban, industrial, or agricultural settings.

2. Project definition

The objective is to design a conceptual platform capable of forecasting reclaimed water demand and automatically adjusting the treatment process accordingly. This integration enables proactive production planning based on predicted demand, minimizing losses and optimizing resource use.

3. System description and methodology

The proposed system architecture consists of five functional modules:

- Data input: historical consumption, weather, seasonal and regulatory variables.
- Demand prediction module: neural network model trained using TensorFlow.
- Treatment planning module

- Control system: manages operation level based on forecasted flow.
- Output indicators: water savings, compliance, efficiency.

4. Predictive model training

A neural network model was implemented using TensorFlow, trained on synthetic data simulating seasonal variations in temperature, precipitation, and water consumption. The model architecture includes dense layers with ReLU activation and the Adam optimizer.

Simulation scenarios included sudden demand increases, water restrictions, and fault detection. Model performance was evaluated using:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- R^2 (Coefficient of Determination)

5. Results

To validate the performance of the reclaimed water demand prediction model, a series of simulations were designed using synthetic data representing different climatic and operational conditions. Dense neural networks were trained with TensorFlow and evaluated using three key metrics: MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and R^2 (Coefficient of Determination).

5.1. Baseline Simulation

The initial model is trained with a dataset that simulates the seasonal and random behavior of the weather. The results obtained show limited performance:

- MAE: 12.58
- RMSE: 15.08
- R^2: −0.70

This result reflects that the model fails to adequately capture the relationship between meteorological variables and consumption. It serves as a functional basis for developing subsequent improvements.

5.2. Simulated extreme scenarios

Six additional scenarios were tested to analyze the system's robustness:

Scenario	MAE	RMSE	R^2
Baseline	12,58	15,08	-0,7
Heat wave	12,46	14,16	-0,11
Extreme drought	12,82	14,97	0,04
Localized heavy rains	11,99	14,2	0,74
Regulatory restriction	11,53	13,19	-0,37
Hidden leak	12,49	14,5	-0,49

Figure II. Comparative simulated scenarios

These results show that the best performance is achieved under the localized heavy rainfall scenario, where the R^2 coefficient reaches 0,74, indicating a remarkable fit. The model correctly explains consumption variability, maintaining low errors.

In contrast, the worst performance appears in the baseline and hidden leakage scenarios, where the model fails to anticipate unexpected changes, reducing its explanatory capacity. Despite this, the absolute errors (MAE and RMSE) remain relatively stable in all cases, suggesting a certain structural robustness of the model, although not always contextual accuracy.

5.3. Global analysis

The model shows stability under extreme conditions but has limitations when capturing anomalous events or unmodeled structural changes. To improve this, new variables (holidays, humidity, restrictions) could be added or training with more real data could be used to improve generalization.

In conclusion, the predictive system has a solid foundation but requires optimization for its application in real-world automated water planning contexts.

6. Conclusions

The proposed system validates the feasibility of integrating AI into reclaimed water management. Although the initial predictive model requires refinement, results confirm its capacity to forecast demand and operate treatment dynamically. This contributes directly to achieving SDGs 6, 11, and 13 by offering a practical tool for municipalities, utilities, and water managers.

Índice de la memoria

Capítulo 1. Introducción 13

1.1. Motivación del proyecto 14

Capítulo 2. Descripción de las tecnologías 15

2.1 Tecnologías de reutilización de aguas residuales 15

2.2 Caudales de mantenimiento y normativa aplicable: El caso del río Llobregat 17

2.3 Caso de estudio: la reutilización de agua en el área metropolitana de Barcelona 19

2.4 Caso práctico: Visión operativa de la reutilización del agua en el Área Metropolitana de Barcelona – Aportación técnica de Aigües de Barcelona 22

2.5 Modelos de Inteligencia Artificial para la Predicción de la Demanda de Agua 26

Capítulo 3. Estado de la cuestión 28

3.1 Reutilización de aguas residuales 28

3.2 Aplicación de la Inteligencia Artificial en la gestión del agua 29

Capítulo 4. Definición del trabajo 31

4.1 Justificación 31

4.2 Objetivos 33

4.3 Metodología 34

Capítulo 5. Sistema propuesto de reutilización y predicción demanda agua 37

5.1 Arquitectura general del sistema 37

5.2 Diseño del sistema de tratamiento 39

5.3 Modelo de predicción de demanda 40

5.4 Integración operativa Predicción - Decisión 41

5.5 Desarrollo técnico del modelo de predicción de demanda de agua regenerada 42

5.6 Simulación inicial con modelo predictivo en TensorFlow 43

5.7 Iteración sobre modelo inicial y simulación de escenarios extremos 54

Capítulo 6. Análisis económico 78

6.1 Evaluación rentabilidad proyecto 82

Capítulo 7. Alineación con los objetivos de desarrollo sostenible 85

Capítulo 8. Conclusiones y trabajos futuros..... 87

Capítulo 9. Bibliografía..... 89

Capítulo 1. INTRODUCCIÓN

La gestión sostenible del agua es un desafío global que adquiere especial relevancia en contextos de creciente escasez hídrica y aumento de la demanda. La reutilización de aguas residuales emerge como una estrategia clave para garantizar el suministro de agua en diversas regiones. Por ejemplo, en la provincia de Alicante, expertos destacan la importancia de combinar inteligentemente todos los recursos hídricos disponibles, incluyendo la reutilización de aguas regeneradas, para asegurar el abastecimiento futuro.

Paralelamente, la aplicación de tecnologías avanzadas, como la inteligencia artificial (IA), está revolucionando la gestión del agua. Proyectos como iRAIN desarrollan sistemas innovadores que integran la monitorización y la IA para regenerar y reutilizar aguas residuales y sus nutrientes en el sector agrícola de forma segura.

Este Trabajo Fin de Máster se centra en el desarrollo de un sistema predictivo que, mediante técnicas de IA, anticipe la demanda de agua y evalúe la viabilidad de implementar tecnologías de reutilización de aguas residuales. La combinación de estas herramientas permitirá optimizar la planificación y gestión de los recursos hídricos, contribuyendo a la sostenibilidad ambiental y económica.

1.1. MOTIVACIÓN DEL PROYECTO

La creciente presión sobre los recursos hídricos debido al cambio climático y al incremento de la demanda exige soluciones innovadoras y sostenibles. La reutilización de aguas residuales no solo reduce la dependencia de fuentes tradicionales, sino que también minimiza el impacto ambiental asociado al vertido de efluentes. Regiones como Aragón han avanzado significativamente en la depuración y reutilización de aguas residuales, contribuyendo al desarrollo sostenible local.

Además, la implementación de sistemas predictivos basados en IA ofrece la posibilidad de anticipar patrones de consumo y optimizar la distribución del agua. Iniciativas como el proyecto Darrow, que desarrolla soluciones basadas en IA para transformar las plantas de tratamiento en instalaciones de recuperación de recursos del agua, hacen que las plantas de tratamiento sean más autónomas y eficientes energéticamente.

La integración de estas tecnologías responde a la necesidad de una gestión más eficiente y sostenible del agua, alineándose con los Objetivos de Desarrollo Sostenible, en particular los ODS 6 (Agua limpia y saneamiento), ODS 11 (Ciudades y comunidades sostenibles) y ODS 13 (Acción por el clima). Este proyecto busca aportar soluciones prácticas y aplicables que contribuyan a enfrentar los desafíos hídricos actuales y futuros. Se explorará el estado del arte en la reutilización de aguas residuales y la predicción de la demanda hídrica mediante IA, se detallará el diseño del sistema propuesto y se evaluará su impacto potencial. Finalmente, se presentarán los resultados obtenidos y las conclusiones derivadas del estudio, sentando las bases para futuras investigaciones en la gestión inteligente del agua.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 TECNOLOGÍAS DE REUTILIZACIÓN DE AGUAS RESIDUALES

La reutilización de aguas residuales es esencial para una gestión hídrica sostenible, especialmente en regiones afectadas por la escasez de agua. Este proceso implica tratar el agua residual para que cumpla con los estándares necesarios para su reutilización en diversas aplicaciones, como la agricultura, la industria y el consumo humano. A continuación, se describen las principales tecnologías empleadas en este ámbito:

- Tratamiento Terciario Avanzado: Este nivel de tratamiento incluye procesos como la ultrafiltración, la ósmosis inversa, la ozonización y la desinfección por rayos ultravioleta (UV). Estas técnicas permiten eliminar contaminantes residuales y patógenos, produciendo agua de alta calidad apta para su reutilización en aplicaciones industriales y agrícolas.
- Biofiltración: Esta tecnología emergente utiliza lechos filtrantes biológicos por los que pasa el agua residual. Los microorganismos adheridos al medio filtrante degradan contaminantes orgánicos e inorgánicos, transformándolos en productos menos nocivos.
- Sistemas de Oxigenación y Adición de Bacterias Beneficiosas: Implementados en cuerpos de agua estancados, estos sistemas mejoran la calidad del agua al prevenir su putrefacción y la acumulación de lodos. La oxigenación y la introducción de bacterias beneficiosas ayudan a mantener la salud de la flora y fauna locales, reduciendo malos olores y la proliferación de mosquitos.
- Automatización y Control: La incorporación de sistemas automatizados y tecnologías de control en las plantas de tratamiento de aguas residuales ha optimizado la eficiencia y seguridad de los procesos. La convergencia entre la

tecnología operacional (OT) y la tecnología de la información (IT) permite una gestión más eficaz y una respuesta rápida a incidencias.

- Internet de las Cosas (IoT): La implementación de dispositivos IoT, como medidores inteligentes, en puntos clave de las plantas de tratamiento permite la recopilación continua de datos en tiempo real. Esta información es vital para monitorear y optimizar los procesos de tratamiento, mejorando la eficiencia operativa.
- Big Data y Analítica Avanzada: El análisis de grandes volúmenes de datos históricos y en tiempo real facilita la identificación de patrones y tendencias, optimizando los procesos de tratamiento y permitiendo la predicción de posibles problemas. Esto contribuye a una toma de decisiones proactiva y a la implementación de medidas correctivas oportunas.

Frente a otras estrategias como la desalación o la captación de aguas superficiales, la reutilización presenta ventajas importantes: menor coste energético, menor huella de carbono, y mayor proximidad al punto de uso. Además, permite cerrar el ciclo del agua, reduciendo el impacto ambiental y mejorando la autosuficiencia hídrica del sistema.

2.2 CAUDALES DE MANTENIMIENTO Y NORMATIVA APLICABLE: EL CASO DEL RÍO LLOBREGAT

En el contexto de la reutilización de aguas residuales y la gestión de la demanda hídrica, uno de los factores normativos clave es el cumplimiento de los caudales ecológicos o de mantenimiento establecidos por las autoridades competentes en materia de agua. Estos caudales representan los valores mínimos de flujo que deben mantenerse en los cursos de agua para preservar los ecosistemas acuáticos y garantizar la calidad ambiental del río.

En el caso concreto de Cataluña, la Agència Catalana de l'Aigua (ACA) define estos caudales a través de distintos documentos de planificación hidrológica. Uno de los más relevantes es el Plan de Gestión del Distrito de Cuenca Fluvial de Cataluña (PDG), que en su Anexo II establece los caudales de mantenimiento en condiciones de normalidad hídrica. Por ejemplo, para el tramo del río Llobregat situado aguas abajo de la ETAP de Sant Joan Despí, se especifican los valores mínimos de caudal que deben respetarse en situaciones ordinarias.

Sin embargo, la planificación hidrológica también contempla escenarios extraordinarios como los períodos de sequía. Para estas situaciones, la ACA ha aprobado el Pla Especial d'Actuació en Situació d'Alerta i Eventual Sequera (conocido como Pla de Sequera), donde se definen niveles de actuación progresivos (normalidad, prealerta, alerta, excepcionalidad y emergencia) en función de la disponibilidad de agua en los sistemas. En cada fase se establecen restricciones y modificaciones de los caudales mínimos permitidos, adaptando la gestión a la situación climática y de recursos. La información relativa a estos umbrales puede encontrarse en el DOGC núm. 8039.

Esta normativa es especialmente relevante para proyectos como el presente, ya que condiciona directamente la viabilidad técnica y operativa de los sistemas de reutilización de aguas residuales. Por ejemplo:

- En situaciones de normalidad hídrica, el sistema puede operar con regularidad, vertiendo aguas tratadas según estándares ambientales establecidos.

- En situaciones de sequía o alerta, la planta debe adaptarse a restricciones más exigentes, lo cual puede afectar tanto el volumen de agua reutilizable como la calidad de tratamiento requerida.

- La existencia de estas limitaciones refuerza la necesidad de un modelo predictivo de demanda que anticipe los cambios de consumo y permita ajustar dinámicamente la operación del sistema.

Además, estas regulaciones constituyen una oportunidad para demostrar el valor de soluciones inteligentes que integren IA y reutilización: sistemas capaces de reaccionar automáticamente a restricciones ambientales y optimizar sus procesos en función de los escenarios normativos vigentes.

2.3 CASO DE ESTUDIO: LA REUTILIZACIÓN DE AGUA EN EL ÁREA METROPOLITANA DE BARCELONA

La reutilización del agua constituye una de las estrategias más eficaces para hacer frente al desafío creciente de la escasez hídrica en entornos urbanos densamente poblados. En este sentido, el Área Metropolitana de Barcelona (AMB) ha desarrollado en las últimas décadas una infraestructura sólida y moderna para el aprovechamiento del agua regenerada, posicionándose como un referente en la gestión sostenible del ciclo urbano del agua en España.

2.3.1. Infraestructura de regeneración en el AMB

La AMB dispone actualmente de un sistema consolidado de depuración y regeneración compuesto por 12 estaciones depuradoras de aguas residuales (EDAR) que dan servicio a más de 3 millones de habitantes, y 4 estaciones de regeneración de agua (ERA) estratégicamente localizadas:

- ERA del Prat
- ERA de Sant Feliu
- ERA de Gavà-Viladecans
- ERA de Montcada i Reixac

Estas instalaciones permiten tratar el agua residual hasta niveles superiores a los requeridos por la normativa vigente, mediante tecnologías avanzadas como la filtración, desinfección UV y ozonización. La infraestructura está diseñada para integrarse dentro del ciclo urbano y ambiental del agua, ofreciendo agua regenerada para múltiples usos no potables: limpieza viaria, riego de zonas verdes, usos agrícolas e industriales, recarga de acuíferos y mantenimiento del caudal ecológico de los ríos. Así, este sistema de regeneración permite aprovechar el agua depurada que, de otro modo, se vertería al mar o al cauce del río, dotándola de una segunda vida útil.

2.3.2. Producción anual de agua regenerada

Según datos oficiales de Aigües de Barcelona, en los últimos años la producción de agua regenerada ha experimentado un crecimiento significativo, tal y como se muestra en la siguiente imagen extraída del portal de Aigües de Barcelona:

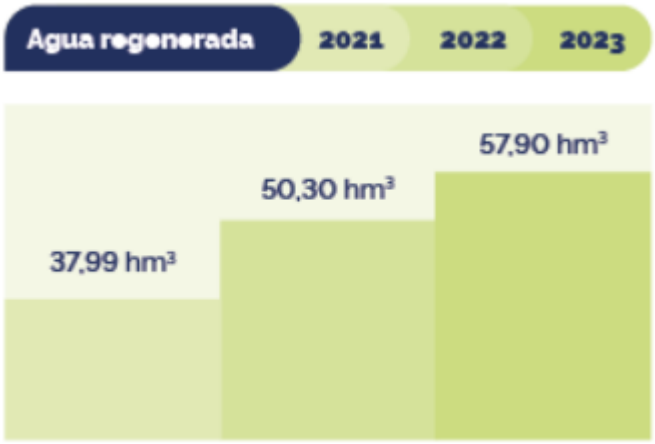


Figura III. Agua regenerada últimos años

Este incremento sostenido en la producción demuestra una clara apuesta institucional por el reaprovechamiento del recurso hídrico, y pone de manifiesto la viabilidad técnica y económica del modelo.

Este caso de estudio aporta una base sólida para justificar la aplicabilidad del sistema propuesto en este TFM. Algunos de los aprendizajes clave que se pueden extraer son, por un lado, la necesidad de contar con infraestructuras de regeneración descentralizadas y adaptadas a distintos usos. También, la utilidad de la monitorización y gestión inteligente del agua regenerada, que puede potenciarse mediante modelos predictivos, y la importancia de alinear la producción de agua regenerada con la demanda real prevista, especialmente en contextos de sequía o restricciones hídricas, lo cual refuerza la utilidad de aplicar técnicas de inteligencia artificial como las propuestas en este trabajo.

Así, el modelo desarrollado en este TFM no solo es técnicamente viable, sino que se alinea con las tendencias y estrategias reales de planificación hídrica que ya se están aplicando con éxito en territorios como el área metropolitana de Barcelona.

2.4 CASO PRÁCTICO: VISIÓN OPERATIVA DE LA REUTILIZACIÓN DEL AGUA EN EL ÁREA METROPOLITANA DE BARCELONA – APORTACIÓN TÉCNICA DE AIGÜES DE BARCELONA

Con el objetivo de enriquecer este trabajo con una perspectiva práctica, se mantuvo una conversación técnica con un profesional de Aigües de Barcelona, entidad operadora del ciclo urbano del agua en el Área Metropolitana de Barcelona (AMB). Esta aportación permite comprender de forma directa algunos de los retos reales en la gestión y reutilización del agua tratada en un entorno urbano e intermunicipal altamente desarrollado.

2.4.1. Usos prioritarios del agua regenerada: peso del componente ambiental

Según el profesional consultado, entre el 80 % y el 85 % del agua regenerada en el AMB se destina a usos ambientales. Esta elevada proporción refleja el compromiso institucional con el mantenimiento de los ecosistemas fluviales, especialmente durante los periodos de sequía prolongada. Uno de los usos principales es la devolución controlada de caudales ecológicos a los ríos, una práctica regulada por los planes de gestión hidrológica y condicionada por los niveles de disponibilidad de recursos.

2.4.2. Dificultades en la predicción del caudal a aportar a los ríos

El cálculo del caudal que debe devolverse a los ríos no está exento de complejidad. El procedimiento habitual consiste en medir el caudal en un punto de aforo aguas abajo, y estimar cuánto caudal debe añadirse en el punto de vertido aguas arriba para alcanzar el valor ecológico establecido.

Sin embargo, este sistema presenta limitaciones importantes:

- Existe un margen de error elevado, especialmente en situaciones de caudal alto, donde el volumen aportado puede diferir significativamente del requerido.

- No siempre es posible conocer con precisión la cantidad real de agua que llega al punto de control, lo que conlleva el riesgo de sobredimensionar el aporte y desperdiciar recursos.
- La incertidumbre operativa puede generar ineficiencias tanto ambientales como energéticas.

Esta situación pone de manifiesto la necesidad de desarrollar sistemas de predicción y optimización del caudal ambiental a partir de modelos de inteligencia artificial, como el propuesto en este proyecto. Estos modelos permitirían anticipar la evolución del caudal natural del río y ajustar dinámicamente la cantidad de agua regenerada que debe aportarse.

2.4.3. Limitaciones en usos agrícolas: pérdidas en transporte

En el ámbito agrícola, se destaca otro reto significativo: la pérdida de agua durante el transporte a través de canales abiertos. En muchos casos, el canal actúa como infraestructura de distribución desde la estación regeneradora hasta los campos de riego. No obstante, las pérdidas por infiltración, evaporación y fugas son considerables, y no existe un sistema fiable para cuantificar el volumen exacto que se pierde, ni para ajustar automáticamente el caudal de envío.

Este tipo de ineficiencia puede corregirse parcialmente con herramientas de predicción del uso real y sistemas de control en campo, lo cual refuerza la utilidad de soluciones integradas de IA en la gestión del agua regenerada.

2.4.4. Usos urbanos descentralizados: logística mediante cisternas

En el entorno urbano, la reutilización del agua regenerada se realiza a menor escala, principalmente mediante el uso de camiones cisterna (cuba). Varios ayuntamientos del AMB cargan agua regenerada en dos estaciones depuradoras y la utilizan para riego de jardines, limpieza de calles y baldeo.

Este sistema, aunque efectivo a pequeña escala, depende de la logística local y no está completamente automatizado ni optimizado desde el punto de vista del consumo energético o la planificación de recursos.

2.4.5. Marco institucional y competencias

La estructura de gestión en el AMB se caracteriza por un modelo de gestión descentralizada con titularidad supramunicipal. El AMB ostenta la titularidad de las estaciones de depuración y regeneración, mientras que los ayuntamientos son los responsables del uso final del agua regenerada. Aigües de Barcelona actúa como operador técnico para este ente supramunicipal.

Uno de los principales obstáculos identificados para ampliar o modernizar las infraestructuras existentes es el acceso a financiación pública, especialmente en contextos de sequía o necesidad de respuesta rápida. Este escollo limita la capacidad de inversión en digitalización, sensores o tecnologías predictivas avanzadas.

2.4.6. Sistemas de tratamiento disponibles

En cuanto a los tratamientos de regeneración utilizados en el AMB, se encuentran sistemas que abarcan desde el biorreactor de membrana (MBR) clásico, hasta tecnologías avanzadas como la ultrafiltración, la microfiltración y la ósmosis inversa. Estas tecnologías permiten alcanzar calidades de agua que cumplen con los requisitos legales para usos agrícolas, industriales y ambientales.

No obstante, el entorno urbano e industrial de Barcelona introduce riesgos de contaminación específica (por ejemplo, compuestos emergentes), lo que obliga a mantener una vigilancia constante sobre la calidad del agua tratada.

Este conjunto de observaciones prácticas ofrece un panorama realista de los desafíos y limitaciones que enfrenta actualmente la reutilización del agua en una gran metrópolis. A su vez, refuerza la justificación del presente proyecto al destacar la necesidad de herramientas predictivas, flexibles e inteligentes que optimicen la gestión de los caudales y la planificación del uso de agua regenerada en un contexto regulado, complejo y vulnerable al cambio climático.

2.5 MODELOS DE INTELIGENCIA ARTIFICIAL PARA LA PREDICCIÓN DE LA DEMANDA DE AGUA

La aplicación de la inteligencia artificial (IA) en la gestión del agua ha revolucionado la capacidad de predecir la demanda hídrica, permitiendo una planificación y asignación más eficiente de los recursos. A continuación, se presentan algunos modelos y aplicaciones destacadas:

- Modelos Basados en la Arquitectura 'Transformer': Investigadores de la Universidad de Córdoba han desarrollado modelos de IA basados en la arquitectura 'Transformer' capaces de predecir la demanda real de agua de riego con un margen de error inferior al 2% a una semana vista. Estos modelos facilitan una gestión más eficiente del agua y la energía en la agricultura.
- Plataformas Web Integradas: El desarrollo de plataformas web que integran modelos predictivos mejora la gestión del agua en cuencas hidrográficas. Estas plataformas permiten prever variables meteorológicas y del ciclo hidrológico, facilitando la toma de decisiones en la gestión de recursos hídricos y la prevención de sequías e incendios.
- Aplicaciones Empresariales de IA: Empresas especializadas han implementado soluciones de IA para optimizar el tratamiento del agua, incluyendo la predicción de la demanda, la mejora de la eficiencia energética y la detección de fugas. Estas aplicaciones contribuyen a una gestión más sostenible y rentable del agua.
- Modelos Holísticos de Riego de Precisión: El proyecto HOPE ha desarrollado modelos que, mediante IA, proporcionan estimaciones precisas de la cantidad de agua necesaria para satisfacer las necesidades de los cultivos. Estos modelos permiten predecir la demanda real de agua de riego con alta precisión, facilitando una gestión eficiente de los recursos hídricos en la agricultura.

- Gestión del Agua y Medioambiente: La IA se utiliza cada vez más en la gestión del agua para mejorar la eficiencia, la sostenibilidad y la rentabilidad de los servicios de abastecimiento y saneamiento, así como para gestionar fenómenos climatológicos extremos asociados a inundaciones y sequías.

La integración de estas tecnologías y modelos de IA en la gestión del agua representa un avance significativo hacia la sostenibilidad y eficiencia en el uso de los recursos hídricos, permitiendo anticipar y responder de manera más efectiva a las necesidades y desafíos asociados al suministro de agua.

Capítulo 3. ESTADO DE LA CUESTIÓN

En este capítulo se entra en detalle sobre los trabajos y soluciones existentes en el ámbito de la reutilización de aguas residuales y la aplicación de inteligencia artificial para la predicción de la demanda de agua. Este análisis permitirá identificar qué se ha desarrollado hasta la fecha y establecer la base para la justificación del proyecto propuesto.

3.1 REUTILIZACIÓN DE AGUAS RESIDUALES

La reutilización de aguas residuales se ha convertido en una estrategia clave para garantizar la sostenibilidad hídrica, especialmente en regiones afectadas por la escasez de agua. Diversos proyectos y soluciones han sido implementados para mejorar el tratamiento y aprovechamiento de estas aguas:

- Proyecto THERESA PCP: Liderado por el Hospital Universitario de Navarra y el centro de investigación Navarrabiomed, este proyecto europeo, con una financiación de 5,2 millones de euros, busca desarrollar tecnologías innovadoras para el tratamiento in situ de aguas residuales hospitalarias. El objetivo es eliminar antibióticos, bacterias resistentes y otros contaminantes, contribuyendo a la iniciativa "One Health" de la Comisión Europea, que promueve el equilibrio entre la salud de personas, animales y ecosistemas.
- Región de Murcia: Reconocida como un modelo mundial en gestión sostenible del agua, la Región de Murcia, a través de la Entidad de Saneamiento y Depuración de la Región de Murcia (ESAMUR), ha implementado sistemas avanzados para maximizar el uso del agua. En colaboración con instituciones como el Instituto Murciano de Investigación y Desarrollo Agrario y Medioambiental (IMIDA), se han desarrollado proyectos agrícolas que optimizan el rendimiento hídrico, destacando la importancia de la reutilización en la agricultura.

3.2 APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN LA GESTIÓN DEL AGUA

La inteligencia artificial ha emergido como una herramienta poderosa para optimizar la gestión del agua, permitiendo una mayor eficiencia y sostenibilidad en el sector. Además, engloba diferentes niveles de modelado. Dentro del aprendizaje automático (machine learning), se encuentran métodos como la regresión, los árboles de decisión o las redes neuronales. Cuando estas redes alcanzan cierta complejidad, con muchas capas ocultas, se habla de aprendizaje profundo (deep learning), que es especialmente potente para modelar grandes volúmenes de datos y relaciones no lineales. Algunos proyectos y aplicaciones relevantes de la IA en gestión hídrica son los siguientes:

- Proyecto iRAIN: Este proyecto tiene como objetivo desarrollar un sistema innovador de gestión de recursos hídricos mediante la aplicación de herramientas avanzadas que combinan monitorización e inteligencia artificial. Utiliza datos de satélites, drones, robots agrícolas y redes de sensores para regenerar y reutilizar aguas residuales y sus nutrientes en el sector agrícola de manera segura.

- Optimización del Tratamiento de Aguas Residuales: La IA se está utilizando para mejorar la eficiencia en las plantas de tratamiento de aguas residuales. Algoritmos avanzados permiten ajustar en tiempo real procesos críticos, como la dosificación de reactivos y el control de líneas de tratamiento, garantizando el cumplimiento de estándares ambientales más estrictos y reduciendo costos operativos.

- Predicción de la Demanda de Agua: Mediante el análisis de grandes volúmenes de datos históricos y en tiempo real, la IA facilita la predicción precisa de la demanda de agua. Esto permite a los proveedores ajustar la distribución y almacenamiento, asegurando un suministro constante y eficiente.

- Detección de Fugas y Reducción de Pérdidas: La implementación de sistemas basados en IA permite identificar fugas en la red de suministro mediante el análisis de datos de sensores. La detección temprana posibilita reparaciones rápidas, reduciendo la cantidad de agua perdida y mejorando la eficiencia en la distribución.

- Modelos Predictivos en Cuencas Hidrográficas: Un estudio aplicado a la Cuenca Hidrográfica del Segura en España integró modelos hidrológicos avanzados, técnicas de teledetección y algoritmos de IA para predecir la disponibilidad de agua, estimar la demanda y optimizar la asignación de recursos a corto y largo plazo. Esta metodología demostró beneficios ambientales significativos y redujo las emisiones de CO2 al optimizar la distribución de recursos.

A diferencia de los modelos deterministas clásicos, que dependen de fórmulas fijas, los modelos de IA permiten adaptar el sistema en tiempo real a partir de los datos. Esto mejora la eficiencia, reduce el riesgo de sobredimensionamiento y permite gestionar la incertidumbre de manera más efectiva, siendo especialmente útil en contextos de alta variabilidad climática y demanda.

En resumen, la IA permite una anticipación más precisa de la demanda, optimización del tratamiento y distribución, reducción de pérdidas, y respuesta más ágil ante condiciones extremas. Frente a métodos tradicionales basados en reglas fijas, los modelos de IA aprenden del comportamiento pasado y pueden adaptarse a contextos cambiantes, haciendo la gestión más eficiente y sostenible.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

A pesar de los avances significativos en el ámbito de la reutilización de aguas residuales y la aplicación de inteligencia artificial en la gestión hídrica, el análisis crítico del estado de la cuestión revela un vacío claro: la falta de soluciones integradas que combinen en una sola plataforma operativa la predicción de la demanda de agua con un sistema optimizado de reutilización de aguas residuales, capaz de adaptarse dinámicamente a las condiciones reales de consumo y a variables climáticas o estacionales.

Si bien existen iniciativas centradas en tecnologías de tratamiento (como en la Región de Murcia o el proyecto THERESA) y otras orientadas a la predicción mediante IA (caso del proyecto iRAIN o los modelos aplicados al riego agrícola), no se ha identificado ninguna solución que integre de forma funcional ambos enfoques en una herramienta práctica, automatizada y escalable para entornos urbanos o industriales.

Además, muchos de los desarrollos actuales están orientados a sectores específicos (agricultura de precisión, hospitales, cuencas hidrográficas), mientras que existe una carencia real de herramientas orientadas a gestores públicos o privados que desean tomar decisiones basadas en datos para optimizar el tratamiento, almacenamiento y distribución del agua reciclada.

Por tanto, este proyecto se justifica como una propuesta innovadora en los siguientes aspectos:

Valor diferencial técnico: se desarrollará un sistema híbrido que une tecnologías de tratamiento de aguas residuales con modelos de predicción basados en IA, lo cual permite la

automatización en la toma de decisiones sobre cuándo y cuánto agua tratar y reutilizar en función de la demanda prevista.

Innovación en la integración de tecnologías: mientras que la mayoría de los proyectos actuales abordan problemas específicos, esta propuesta se enfoca en el desarrollo de una plataforma interoperable y modular, con potencial para ser aplicada tanto en municipios como en instalaciones industriales o agrícolas.

Alineación con el mercado y las necesidades reales: en un contexto donde la sequía, la presión regulatoria y la necesidad de eficiencia hídrica son crecientes, los gestores necesitan herramientas que faciliten el cumplimiento normativo, la reducción de costes operativos y la mejora en la planificación hídrica.

Oportunidad de escalabilidad y replicabilidad: el enfoque adoptado permite que el modelo pueda adaptarse a distintas escalas y contextos geográficos, lo que lo convierte en una solución comercialmente viable para empresas de utilities, ayuntamientos, parques tecnológicos, complejos turísticos y más.

Apoyo a los Objetivos de Desarrollo Sostenible (ODS): el sistema propuesto tiene un impacto directo en al menos tres ODS: el ODS 6 (Agua limpia y saneamiento), el ODS 11 (Ciudades sostenibles) y el ODS 13 (Acción por el clima), lo cual refuerza su valor estratégico ante inversores, administraciones públicas y entidades medioambientales.

En definitiva, este proyecto no solo representa una mejora técnica frente a soluciones existentes, sino que responde a una demanda concreta del mercado, proponiendo una solución integral, automatizada y sostenible para la gestión inteligente del agua.

4.2 OBJETIVOS

El objetivo principal de este Trabajo Fin de Máster es el diseño y desarrollo de un sistema integrado que permita reutilizar aguas residuales de forma eficiente y, a su vez, predecir con precisión la demanda de agua en distintos escenarios utilizando herramientas de inteligencia artificial. Se trata de ofrecer una solución innovadora y práctica para optimizar la gestión hídrica en entornos urbanos, industriales o agrícolas.

Este objetivo general se concreta en los siguientes objetivos específicos:

- Diseñar un sistema de tratamiento y reutilización de aguas residuales adaptado a usos concretos.
    - Identificar tecnologías viables de tratamiento fisicoquímico, biológico y avanzado para garantizar la calidad del agua tratada.
    - Definir las etapas del proceso y sus condiciones operativas óptimas.
    - Evaluar la aplicabilidad del sistema a distintos fines: riego, procesos industriales, recarga de acuíferos, etc.

- Desarrollar un modelo predictivo de demanda de agua basado en técnicas de inteligencia artificial.
    - Recopilar y preprocesar datos históricos de consumo, datos meteorológicos y variables socioeconómicas.
    - Aplicar modelos de Machine Learning (como redes neuronales, modelos autoregresivos, etc.) para anticipar la demanda hídrica.
    - Validar la precisión del modelo mediante indicadores estadísticos como MAE, RMSE o R^2.

- Integrar el sistema de tratamiento con la herramienta de predicción en una plataforma operativa.

- Diseñar una arquitectura funcional donde el modelo predictivo oriente la operación del sistema de tratamiento.
- Plantear una interfaz básica (dashboard) que permita visualizar datos, realizar simulaciones y tomar decisiones operativas.
- Proponer mecanismos de automatización para activar o ajustar procesos de tratamiento en función de la demanda prevista.

- Evaluar la viabilidad técnica, económica y ambiental del sistema desarrollado.
  - Estimar los costes de implantación, operación y mantenimiento del sistema.
  - Analizar los beneficios ambientales derivados de la reducción del consumo de agua dulce y vertidos.
  - Valorar la escalabilidad del sistema a otras regiones o sectores.

4.3 METODOLOGÍA

El desarrollo de este Trabajo Fin de Máster se estructura en una secuencia de etapas interrelacionadas que permiten alcanzar de forma sistemática los objetivos definidos. La metodología combina técnicas de investigación aplicada, desarrollo de modelos predictivos y análisis técnico-económico. A continuación, se describe el enfoque seguido:

Fase 1: Revisión del estado del arte y definición del alcance

- Investigación bibliográfica sobre tecnologías de tratamiento de aguas residuales, sistemas de reutilización y aplicaciones de inteligencia artificial en predicción de la demanda de agua.
- Análisis de estudios de caso existentes a nivel nacional e internacional.
- Delimitación de los casos de uso más relevantes para el desarrollo del sistema: entornos urbanos, industriales y/o agrícolas.
- Definición del alcance técnico del proyecto y requisitos funcionales del sistema.

Fase 2: Diseño del sistema de reutilización de aguas residuales

- Selección de tecnologías de tratamiento apropiadas según calidad del agua de entrada y uso final (riego, limpieza urbana, procesos industriales...).
- Definición de una cadena de tratamiento modular (decantación, filtración, desinfección, etc.).
- Simulación básica del rendimiento del sistema con parámetros realistas.
- Representación del sistema mediante diagramas de flujo y esquemas de proceso.

Fase 3: Desarrollo del modelo de predicción de demanda de agua

- Recopilación de datos históricos de consumo de agua, meteorología y variables estacionales (de fuentes públicas como AEMET, organismos locales o datasets abiertos).
- Limpieza y preprocesamiento de datos para su adaptación al modelo (normalización, detección de valores atípicos, etc.).
- Entrenamiento de modelos de aprendizaje automático: regresión lineal, árboles de decisión, y redes neuronales recurrentes (RNN o LSTM).
- Evaluación del rendimiento con métricas como MAE (Mean Absolute Error), RMSE (Root Mean Square Error) y R^2.
- Selección del modelo más robusto y preciso.

Fase 4: Integración del sistema de tratamiento y el modelo predictivo

- Diseño de una arquitectura funcional que vincule la predicción de la demanda con la activación dinámica del sistema de tratamiento (por ejemplo: predicción de alta demanda = activar tratamiento completo).
- Propuesta de una plataforma de visualización de resultados: dashboard o interfaz que muestre predicciones, datos históricos y alertas.
- Validación teórica de la operatividad del sistema integrado.

Fase 5: Evaluación de la viabilidad técnico-económica y ambiental

- Estimación de costes de implementación y operación del sistema propuesto.
- Cálculo del ahorro potencial de agua y costes frente a alternativas tradicionales.
- Análisis del impacto ambiental positivo en términos de reducción del uso de agua potable, emisiones de CO_2 y vertidos.
- Alineación del sistema con los Objetivos de Desarrollo Sostenible (ODS).

Fase 6: Redacción del documento y presentación

- Organización de resultados, esquemas, tablas y gráficos.
- Redacción de la memoria en formato académico.
- Preparación de la presentación oral de defensa del TFM.

Capítulo 5. SISTEMA PROPUESTO DE REUTILIZACIÓN Y PREDICCIÓN DEMANDA AGUA

En este capítulo se presenta el sistema desarrollado que integra dos módulos principales: un sistema de tratamiento y reutilización de aguas residuales optimizado para usos no potables, y un modelo predictivo de demanda hídrica basado en técnicas de inteligencia artificial (IA). El objetivo es diseñar una plataforma conceptual capaz de predecir la demanda de agua regenerada y ajustar dinámicamente el funcionamiento de la planta de tratamiento, mejorando la eficiencia operativa y reduciendo el desperdicio de recursos hídricos.

5.1 ARQUITECTURA GENERAL DEL SISTEMA

El sistema propuesto se basará en una arquitectura modular y escalable, compuesta por los siguientes bloques funcionales:

- Entrada de datos: recopilación y gestión de históricos de consumo de agua, variables meteorológicas (temperatura, precipitación, humedad), factores estacionales y normativos (por ejemplo, caudales mínimos ambientales).
- Módulo de predicción de demanda: modelo de IA entrenado con datos históricos que estima la demanda futura en horizontes temporales variables (horarios, diarios o semanales).

- Módulo de planificación del tratamiento: cálculo del caudal necesario a regenerar en función de la predicción de demanda, considerando pérdidas, usos específicos y eficiencia del sistema.
- Sistema de control y operación: determina la activación y el grado de operación de los procesos de tratamiento según el caudal estimado.
- Salida: volúmenes de agua regenerada por tipo de uso, ahorro en consumo de agua potable, cumplimiento normativo y evaluación de eficiencia.

5.2 DISEÑO DEL SISTEMA DE TRATAMIENTO

El sistema de tratamiento considerado en este proyecto se basa en tecnologías comúnmente utilizadas en entornos urbanos, como tratamiento terciario con desinfección UV, filtración avanzada mediante microfiltración o ultrafiltración, y una opción de tratamiento por ósmosis inversa en zonas sensibles o para usos agrícolas con alta exigencia.

El sistema se configura para generar agua regenerada apta para usos no potables, especialmente riego agrícola, mantenimiento de caudales ecológicos y limpieza urbana.

Se considerará un sistema modular, adaptable a distintos tamaños de EDAR y escalable según la demanda local.

5.3 MODELO DE PREDICCIÓN DE DEMANDA

El núcleo innovador del sistema es el modelo predictivo de demanda de agua regenerada, que se desarrolla siguiendo las siguientes fases:

- Recopilación de datos: histórica de consumo por usos, meteorología, calendario agrícola, indicadores de sequía y otros parámetros relevantes.
- Preprocesamiento: normalización, limpieza, imputación de valores faltantes y análisis exploratorio para determinar correlaciones significativas.
- Modelado: construcción y entrenamiento de modelos de inteligencia artificial con TensorFlow, utilizando arquitecturas de redes neuronales adaptadas a la naturaleza y volumen de los datos.

La elección de TensorFlow se fundamenta en su flexibilidad, escalabilidad y robustez, permitiendo probar desde modelos simples hasta redes profundas o recurrentes, así como integrar el entrenamiento, validación y simulación en un mismo entorno.

5.4 INTEGRACIÓN OPERATIVA PREDICCIÓN - DECISIÓN

Una característica diferencial del sistema es la integración funcional entre el modelo predictivo y el sistema de regeneración. La predicción anticipada de la demanda permite planificar con antelación el caudal a regenerar, optimizando los procesos y minimizando pérdidas o vertidos innecesarios.

Esta estrategia es especialmente relevante en escenarios de restricciones hídricas, donde la precisión en la planificación y el control son críticos para mantener la eficiencia operativa. La plataforma puede aplicarse en contextos urbanos, agrícolas y ambientales, facilitando la sincronización entre demanda prevista y oferta de agua regenerada.

5.5 DESARROLLO TÉCNICO DEL MODELO DE PREDICCIÓN DE DEMANDA DE AGUA REGENERADA

Para la validación y desarrollo técnico del modelo se utilizarán técnicas de aprendizaje automático implementadas en TensorFlow. Se iniciará con la generación y simulación de datos que reflejen las condiciones reales del sistema, incluyendo variables meteorológicas y consumo.

Se evaluarán diversas arquitecturas y configuraciones de redes neuronales, optimizando parámetros y funciones de pérdida para mejorar la precisión predictiva. La evaluación se realizará mediante métricas estándar como el error absoluto medio (MAE), la raíz del error cuadrático medio (RMSE) y el coeficiente de determinación (R^2).

Posteriormente, se implementarán simulaciones de escenarios extremos, como aumentos súbitos de demanda, restricciones de agua o detección de fugas, para evaluar la robustez y adaptabilidad del modelo predictivo. Estas simulaciones permitirán analizar el impacto en la operación del sistema de tratamiento y su capacidad para ajustar dinámicamente la producción de agua regenerada.

Las redes neuronales artificiales utilizadas en este modelo imitan el funcionamiento de las redes neuronales biológicas, y se componen de capas de nodos o “neuronas” interconectadas. Estas redes son especialmente eficaces para capturar relaciones no lineales y complejas entre variables, como ocurre con el consumo de agua en función de factores meteorológicos. Se han preferido frente a otros métodos de regresión debido a su capacidad para aprender automáticamente patrones ocultos a partir de los datos.

5.6 SIMULACIÓN INICIAL CON MODELO PREDICTIVO EN TENSORFLOW

Para iniciar el desarrollo del módulo predictivo de demanda de agua regenerada, se ha diseñado una simulación básica utilizando variables climatológicas y de consumo diarias. El objetivo es construir un modelo que relacione la temperatura y la precipitación con el consumo hídrico, emulando la influencia de estas variables sobre la demanda.

Las variables consideradas serán las siguientes:

- Temperatura (°C): Representa la temperatura diaria promedio, con un comportamiento estacional simulado mediante una función sinusoidal y ruido aleatorio para reflejar variaciones reales.
- Precipitación (mm): Se genera a partir de una distribución gamma combinada con un filtro probabilístico para simular días sin lluvia y días con precipitación variable.
- Consumo de agua (m^3/día): Variable objetivo, calculada como función lineal de temperatura y precipitación con ruido aleatorio añadido, para reflejar el consumo real con fluctuaciones e incertidumbre.

A continuación, se presenta un fragmento del código implementado en Python utilizando TensorFlow para entrenar una red neuronal densa que prediga el consumo a partir de las variables independientes:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Generación de datos simulados
np.random.seed(0)
dias = np.arange(365)
temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)
precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)
consumo = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)

data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo
})

X = data[['temperatura', 'precipitacion']]
y = data['consumo']

# División en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Definición del modelo
modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])

modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
```

Figura IV. Código Python primer modelo predictivo demanda agua

Esta simulación, ejecutada en Python con la herramienta TensorFlow, devolverá las variables MAE (Error absoluto medio), RMSE (Raíz del Error Cuadrático Medio) y R^2 (Coeficiente de Determinación).

En primer lugar, se importan las librerías necesarias:

- ***numpy*** para operaciones numéricas y generación de datos aleatorios.
- ***pandas*** para manejo de datos en forma de tablas.
- ***tensorflow*** y ***keras*** para construir y entrenar el modelo de red neuronal.

- ***sklearn*** para dividir el conjunto de datos en entrenamiento y prueba, y calcular métricas de evaluación.
- ***train_test_split*** y las métricas MAE, RMSE y R^2 son útiles para validar el rendimiento del modelo.

Después, se fija la semilla de generación aleatoria de números ***np.random.seed(0)***, y posteriormente se genera el vector de días de un año completo (***días = np.arrange(365)***). Seguidamente, se simula la temperatura diaria, que viene dada por la siguiente fórmula sinusoidal a la que se añade un componente aleatorio para simular la variabilidad diaria:

$$T = 20 + 5 \times \sin(\frac{2 \times \pi \times d}{365}) + Ruido$$

Donde:

- *d* es el número del día en el año (de 0 a 364)
- *20* es la temperatura media anual en ºC.
- *5* es la amplitud de la variación estacional
- *Ruido* es una distribución normal con media 0 y desviación típica 1,5 °C para introducir variaciones diarias.

Esta fórmula genera un comportamiento térmico realista con máximos en verano y mínimos en invierno, adaptado a climas templados típicos de zonas urbanas en España.

Después se simula la precipitación diaria, mediante una combinación de distribución gamma y una condición de ocurrencia probabilística para reflejar su naturaleza esporádica y asimétrica:

$$P = \Gamma(1{,}5; 1{,}0) \times I(U_d > 0{,}7)$$

Donde en primer lugar se extrae un valor aleatorio de la distribución Gamma con forma 1,5 y escala 1, adecuada para modelar precipitaciones positivas con alta variabilidad, y este valor

se multiplica por la condición de probabilidad de si llueve o no ese día (valores 0 o 1), suponiendo que aproximadamente sólo llueve un 30% de los días del año en la zona mediterránea que nos ocupa.

Esta última función se explicaría así: para cada día del año (365 días), se genera un número aleatorio entre 0 o 1. Si este número es mayor que 0,7 (lo que ocurre el 30% del tiempo), se considera que ese día llueve, y en este caso la fórmula genera una cantidad de lluvia usando la distribución Gamma inicial. En cambio, si el número es 0 y por tanto menor que 0,7, significa que no llueve ese día, tomando la fórmula un valor de precipitación de 0.

La distribución Gamma es adecuada para simular precipitaciones porque solo da valores positivos y es asimétrica, como la lluvia real.

Después, se calcula el consumo de agua diario como función lineal de la temperatura y la precipitación, con coeficientes que reflejan su influencia positiva (temperatura) o negativa (precipitación):

consumo = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)

Se añade ruido normal con desviación 3 para simular variabilidad e incertidumbre en el consumo real.

Una vez calculadas estas variables, se procede a crear el DataFrame, organizando las variables simuladas (temperatura, precipitación y consumo) en una tabla para facilitar su manejo y análisis posterior.

Se distingue entre las variables explicativas (la temperatura y la precipitación diarias), que son las variables de entrada al modelo de predicción, y la variable objetivo (el consumo diario).

Las variables explicativas representan las condiciones meteorológicas que influyen o ayudan a predecir la otra variable, el consumo diario, que es la variable que se quiere encontrar con el modelo. Así, las variables explicativas se denotan con X, y la objetivo con Y:

X = data[['temperatura', 'precipitacion']]

y = data['consumo']

Seguidamente, se pasa a la división del conjunto de datos en datos de entrenamiento y prueba; se separan el 80% de los datos para entrenamiento y el 20% restante a prueba (test). Esto se debe a que para entrenar un modelo de inteligencia artificial de manera adecuada, es fundamental dividir los datos disponibles en dos conjuntos principales:

- Conjunto de entrenamiento: Datos que el modelo utiliza para aprender patrones y ajustar sus parámetros.
- Conjunto de prueba (test): Datos que el modelo no ha visto durante el entrenamiento y que se usan para evaluar su capacidad de generalización, es decir, qué tan bien predice con datos nuevos.

Esta división ayuda a evitar el sobreajuste, que ocurre cuando el modelo memoriza los datos de entrenamiento y no funciona bien con datos nuevos.

En el código, esta división se realiza con la función ***train_test_split*** de la biblioteca Scikit-learn:

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

En ella, se especifica que el tamaño del test será del 20% (0,2), dejando el resto para entrenamiento, y el parámetro ***random_state*** fija la semilla del generador de números aleatorios que selecciona aleatoriamente las muestras para cada conjunto. Se establece en un valor específico (en este caso 42) para garantizar que la división del conjunto de datos sea reproducible. Es decir, si ejecutas el código varias veces o alguien más lo hace, la misma

división se obtendrá siempre. Esto es importante para comparar resultados y hacer análisis consistentes, ya que evita que los resultados cambien debido a diferentes divisiones aleatorias.

El número 42 es un valor muy usado convencionalmente en programación y ciencia de datos porque es una "broma" cultural (la "respuesta a la vida, el universo y todo" según el libro "La guía del autoestopista galáctico"), pero puede ser cualquier número entero.

El próximo paso será definir el modelo de red neuronal mediante el siguiente código:

```
modelo = keras.Sequential([

keras.layers.Dense(32, activation='relu', input_shape=(2,)),

keras.layers.Dense(16, activation='relu'),

keras.layers.Dense(1)

])
```

Se crea una red neuronal secuencial con una capa densa de 32 neuronas con función de activación ReLU, que recibe 2 variables de entrada, una segunda capa densa con 16 neuronas y activación ReLU para extraer características más complejas, y por último una capa de salida con una neurona para predecir un valor continuo (el consumo).

Por último, se compilará el modelo con la siguiente línea de código:

```
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
```

Definiendo el optimizador Adam para el entrenamiento, la función de pérdida como el error cuadrático medio (MSE) y monitorizando el error absoluto medio (MAE) como métrica.

El optimizador Adam es un algoritmo de optimización muy utilizado en el entrenamiento de redes neuronales. Su nombre viene de "Adaptive Moment Estimation" (estimación

adaptativa de momentos). Durante el entrenamiento de un modelo de redes neuronales, este optimizador es el encargado de ajustar los pesos (o parámetros) del modelo para minimizar la función de pérdida (por ejemplo, el error cuadrático medio entre las predicciones y los valores reales). Es decir, el optimizador guía el aprendizaje del modelo paso a paso.

Adam utiliza dos ideas clave:

1. Estimación del primer momento (media del gradiente): calcula la media de los gradientes pasados, como hace el método del momentum.

2. Estimación del segundo momento (varianza del gradiente): calcula la varianza de los gradientes, como hace RMSProp.

El método Momentum permite acelerar el descenso por gradiente suavizando la dirección del gradiente con una media exponencial, mejorando la convergencia. Por su parte, RMSProp adapta la tasa de aprendizaje individual de cada parámetro dividiendo el gradiente por la raíz cuadrada de una media móvil de su magnitud, lo que permite mayor estabilidad en escenarios no estacionarios.

Con estas dos estimaciones, Adam adapta automáticamente la tasa de aprendizaje para cada parámetro, lo que permite una convergencia más rápida, mayor estabilidad durante el entrenamiento y un buen rendimiento en la práctica.

Así, se utiliza en la línea de código indicándole al compilador que use Adam para compilar el modelo, minimizando el error cuadrático medio (MSE) y mostrando como métrica el error absoluto medio (MAE).

```
# Entrenamiento del modelo
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

# Evaluación
predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

print(f"\nResultados de la predicción:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura V. Entrenamiento del modelo

Una vez compilado el modelo se procede al entrenamiento. Se entrena durante 50 épocas, usando el 20% del conjunto de entrenamiento para validación interna. La expresión ***verbose=1*** muestra el progreso durante el entrenamiento.

Seguidamente, en la evaluación, se produce la predicción sobre los datos de prueba. El modelo genera predicciones de consumo para los datos no vistos durante el entrenamiento. La expresión ***flatten()*** convierte el resultado en un vector unidimensional.

Finalmente ocurre el cálculo de las métricas de evaluación:

- MAE (Error Absoluto Medio): promedio del valor absoluto de la diferencia entre predicción y valor real.
- RMSE (Raíz del Error Cuadrático Medio): refleja la magnitud promedio del error, penalizando más errores grandes.
- R^2 (Coeficiente de determinación): indica la proporción de varianza explicada por el modelo (valor máximo 1).

Y su posterior impresión en pantalla, mostrando las métricas calculadas con dos decimales para interpretar la calidad del modelo.

Así, se compila el modelo en TensorFlow y se obtienen los siguientes resultados:

```
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 235.1723 - mae: 13.0272 - val_loss: 237.6960 - val_mae: 12.6248
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 222.0896 - mae: 12.5949 - val_loss: 237.3075 - val_mae: 12.6906
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 228.2333 - mae: 12.7420 - val_loss: 237.1360 - val_mae: 12.7096
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 241.1689 - mae: 13.4590 - val_loss: 236.9915 - val_mae: 12.7166
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 258.7359 - mae: 13.9388 - val_loss: 236.8279 - val_mae: 12.7163
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 247.6266 - mae: 13.5493 - val_loss: 236.6631 - val_mae: 12.6868
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 234.0791 - mae: 13.0354 - val_loss: 236.5016 - val_mae: 12.6986
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 225.6360 - mae: 12.6821 - val_loss: 236.3335 - val_mae: 12.6970
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 236.9487 - mae: 13.2514 - val_loss: 236.1585 - val_mae: 12.7155
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 241.3000 - mae: 13.3424 - val_loss: 235.9887 - val_mae: 12.6952
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 217.3479 - mae: 12.5590 - val_loss: 235.8301 - val_mae: 12.6953
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 232.7265 - mae: 13.0972 - val_loss: 235.6007 - val_mae: 12.6893
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 233.7269 - mae: 13.0135 - val_loss: 235.3805 - val_mae: 12.7026
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 244.1965 - mae: 13.4575 - val_loss: 235.1810 - val_mae: 12.7052
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 228.8562 - mae: 13.0395 - val_loss: 234.9549 - val_mae: 12.6832
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 211.0163 - mae: 12.2634 - val_loss: 234.7338 - val_mae: 12.6905
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step

Resultados de la predicción:
MAE: 12.58
RMSE: 15.08
R²: -0.70

C:\Users\javie\OneDrive\Escritorio\ICAI 24-25\TFM>
```

Figura VI. Resultados primera simulación

Los resultados obtenidos son un MAE de 12.58, un RMSE de 15.08 y un coeficiente R^2 de -0.70. A continuación se interpretan los resultados:

- MAE (Mean Absolute Error): 12.58. Significa que, en promedio, el modelo se equivoca 12.58 unidades al predecir el consumo de agua. Es una medida intuitiva del error medio: cuanto menor, mejor.

- RMSE (Root Mean Squared Error): 15.08. Es similar al MAE pero penaliza más los errores grandes. Un RMSE de 15.08 indica una desviación típica del error algo superior al MAE, lo que sugiere que hay algunos errores elevados.

- R^2 (Coeficiente de determinación): -0.70. Esta métrica evalúa qué porcentaje de la varianza de los datos se explica por el modelo. Un valor negativo indica que el modelo predice peor que simplemente usar la media de los valores reales. En general:
    - R^2 cercano a 1 significa un muy buen ajuste.
    - R^2 cercano a 0 supone un modelo sin capacidad predictiva.
    - R^2 negativo implica un modelo peor que un modelo trivial.

Si bien los errores absolutos son razonables en términos relativos, el valor negativo del R^2 indica que el modelo no logra generalizar adecuadamente la relación entre variables meteorológicas y la demanda de agua. Este comportamiento puede deberse a la simplicidad del modelo, al número limitado de variables predictoras y al uso de un conjunto de datos sintético con alto nivel de ruido.

Los próximos pasos que se pueden seguir para optimizar y mejorar el modelo son los siguientes:

- Añadir más variables: Día de la semana, estación del año, humedad, evapotranspiración, nivel del embalse, datos históricos de consumo, etc.
- Aumentar los datos: Simular varios años (más de 365 días) para tener un conjunto de entrenamiento más robusto.
- Mejorar el modelo: Aumentar la arquitectura: más capas o más neuronas o entrenar más épocas.
- Simular escenarios extremos, como los siguientes:
    - Aumento extremo de consumo (por ejemplo por ola de calor, festividades, etc.).
    - Restricciones de agua (escenarios de sequía donde se reduce la oferta).

- Introducción de fugas o pérdidas en la red (ver cómo se comporta la predicción con anomalías).
- Cambios estacionales marcados: ver cómo cambia la demanda en verano vs invierno.

Todo esto se podrá simular modificando artificialmente los datos de entrada y observando si el modelo sigue siendo fiable o si se rompe.

5.7 ITERACIÓN SOBRE MODELO INICIAL Y SIMULACIÓN DE ESCENARIOS EXTREMOS

Una vez implementado el primer modelo de predicción de demanda de agua utilizando técnicas de aprendizaje automático con TensorFlow, se procedió a evaluar su rendimiento sobre un conjunto de datos simulados. Los resultados obtenidos fueron un MAE de 12.58, un RMSE de 15.08 y un R^2 de -0.70.

Estos indicadores muestran que el modelo, en su forma inicial, presenta una capacidad predictiva limitada. El valor negativo del coeficiente de determinación indica que el modelo no logra captar adecuadamente la relación entre las variables meteorológicas (temperatura y precipitación) y el consumo de agua. A pesar de ello, este primer resultado constituye una base funcional sobre la cual construir y optimizar el modelo predictivo.

A partir de este análisis, se planteó la simulación de escenarios extremos de estrés hídrico o variabilidad en la demanda para validar la robustez y utilidad del modelo en condiciones límite. Se proponen los siguientes escenarios:

- Aumento de la demanda: simular olas de calor, aumento de población temporal o usos intensivos.
- Restricción de recursos: simular condiciones de sequía o reducción de la disponibilidad hídrica.
- Fugas o pérdidas: introducir ruido artificial en los datos para simular anomalías operativas.
- Cambios estacionales marcados: evaluar la capacidad del modelo para adaptarse a las diferencias entre estaciones húmedas y secas.

Por cada nueva versión del modelo o escenario simulado, se hará un análisis de sensibilidad: se registrarán métricas de rendimiento (MAE, RMSE, R^2) y se analizarán los resultados de

forma crítica. Esto permitirá identificar las configuraciones más eficientes y comparar los beneficios obtenidos frente al modelo base.

Así, se procede a simular un escenario extremo de aumento de demanda debido a una ola de calor prolongada. Se aumentará la temperatura 7ºC durante 30 días seguidos, lo que afectará significativamente al consumo de agua. A continuación, se expone el código usado en Python:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Simulación de datos con ola de calor (días 180 a 210)
np.random.seed(1)
dias = np.arange(365)
temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)
temperatura[180:210] += 7  # Ola de calor: aumento de 7ºC durante 30 días

precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)
consumo = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)

# Crear DataFrame
data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo
})

# Separación de variables independientes y dependiente
X = data[['temperatura', 'precipitacion']]
y = data['consumo']

# División en entrenamiento y test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Modelo de predicción
modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])
```

```
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Entrenamiento del modelo
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

# Evaluación
predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

# Resultados
print("\nResultados del escenario con ola de calor:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura VI. Código simulación ola de calor

En primer lugar, se importan las librerías necesarias igual que se hizo con la simulación inicial. Seguidamente, **se** inicializa el generador aleatorio para reproducibilidad ***np.random.seed(1)***. En este segundo escenario se cambia la semilla de 0 a 1 para garantizar que la serie aleatoria de ruido y precipitación sea diferente, representando así un escenario alternativo. Si se repite la misma semilla, el modelo tendría exactamente el mismo ruido aleatorio en ambas simulaciones. Después, se crea el vector de días de un año completo (***días = np.arrange(365)***).

A continuación, se escriben las fórmulas para la temperatura, la precipitación y el consumo. La de la temperatura se expresa de igual manera que en el primer escenario (***temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)***), añadiendo la ola de calor artificial durante un mes, más o menos en época veraniega (días del año 180 a 210, sumándole 7ºC a la temperatura normal. Esto se consigue con la siguiente expresión:

temperatura[180:210] += 7

La precipitación se expresa con la misma fórmula que en el caso inicial, teniendo en cuenta la suposición de que sólo lloverá el 30% de los días (***precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)***), y lo mismo con el consumo (***consumo = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)***). Se espera que con el aumento de calor, aumente el consumo.

El próximo paso es organizar la información en un DataFrame para facilitar su manipulación, definir las variables explicativas (temperatura y precipitación) y la variable objetivo (consumo), y dividir el conjunto de datos en 80% para entrenamiento del modelo y el resto para test.

De igual manera que en el primer caso, se crea la red neuronal y se compila. A continuación, se entrena el modelo durante 50 épocas, reservando un 20% del conjunto de entrenamiento como validación (***modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)***).

Finalmente se hace la evaluación; se obtienen las predicciones del modelo sobre los datos de prueba, convirtiéndolas a un array unidimensional (***predicciones = modelo.predict(X_test).flatten()***), seguido del cálculo de las tres métricas de estudio, imprimiéndolas como paso final.

Así, se compila el modelo con TensorFlow, obteniéndose los siguientes resultados:

```
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 243.0638 - mae: 12.8303 - val_loss: 233.0722 - val_mae: 13.0028
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 240.2194 - mae: 12.4986 - val_loss: 232.8329 - val_mae: 12.9972
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 254.1382 - mae: 13.4806 - val_loss: 232.0497 - val_mae: 13.0525
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 223.5910 - mae: 12.2716 - val_loss: 232.0395 - val_mae: 13.0186
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 225.9811 - mae: 12.5049 - val_loss: 231.8114 - val_mae: 13.0125
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 238.8544 - mae: 12.8198 - val_loss: 231.1168 - val_mae: 13.0771
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 249.4789 - mae: 12.9330 - val_loss: 231.0414 - val_mae: 13.0381
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 240.4427 - mae: 12.7745 - val_loss: 230.9536 - val_mae: 13.0146
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 240.4707 - mae: 12.8151 - val_loss: 230.4535 - val_mae: 13.0497
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 212.9247 - mae: 12.0434 - val_loss: 230.1446 - val_mae: 13.0693
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 250.4082 - mae: 13.0077 - val_loss: 229.8982 - val_mae: 13.0665
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 228.6366 - mae: 12.4965 - val_loss: 229.6372 - val_mae: 13.0796
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 247.2570 - mae: 13.3929 - val_loss: 229.4854 - val_mae: 13.0509
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 230.5944 - mae: 12.7119 - val_loss: 229.4178 - val_mae: 13.0182
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 236.8389 - mae: 12.9301 - val_loss: 229.0043 - val_mae: 13.0559
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - loss: 254.0065 - mae: 13.4961 - val_loss: 228.8217 - val_mae: 13.0387
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step

Resultados del escenario con ola de calor:
MAE: 12.46
RMSE: 14.16
R²: -0.11
```

Figura VIII. Resultados simulación caso extremo ola de calor

Los resultados obtenidos para este escenario son un Error Absoluto Medio de 12.46, una Raíz del Error Cuadrático Medio de 14.16 y un Coeficiente de determinación de -0.11. La tabla comparativa con los resultados obtenidos en la primera simulación se expone a continuación:

Métrica	Escenario base	Ola de calor	Cambio
MAE	12,58	12,46	Ligera mejora
RMSE	15,08	14,16	Mejora
R^2	-0,7	-0,11	Mejora notable

Tabla I. Comparativa resultados escenario inicial – ola de calor

A la vista de los resultados, aunque ambos escenarios tienen un R^2 negativo, lo cual indica que el modelo aún no generaliza bien en los datos de prueba, el valor se acerca más a cero en el escenario de ola de calor. Además, tanto el MAE como el RMSE han disminuido, lo que indica que, en términos absolutos, el modelo ha cometido menos errores en la predicción bajo condiciones de calor extremo. Esto sugiere que el modelo responde ligeramente mejor cuando la temperatura es más elevada y estable, posiblemente porque la temperatura tiene un peso mayor en la fórmula de generación del consumo (***consumo = 80 + 3 * temperatura - 1.5 * precipitación + ruido***).

Las conclusiones que se sacan de este modelo son que el modelo tiene margen de mejora, ya que el coeficiente de determinación (R^2) sigue siendo negativo en ambos escenarios. Aun así, se observa como el modelo se comporta mejor en condiciones extremas de calor que en condiciones medias simuladas, probablemente porque el patrón de consumo es más lineal y predecible. No se ha introducido precipitación extrema, por lo que el efecto de eventos hidrológicos anómalos (sequía, lluvias intensas) aún no ha sido evaluado, pero es probable que se obtengan más resultados reveladores.

Se probará a realizar una simulación de escenario de sequía extrema, con precipitaciones nulas o muy bajas y temperaturas normales, para observar cómo reacciona el modelo, y posteriormente si hiciera falta se podrá probar a introducir más variables explicativas: humedad, calendario agrícola, festivos, caudales mínimos legales o restricciones por sequía.

Así, el código propuesto para el escenario de sequía extrema es el siguiente:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Simulación: Sequía prolongada (temperaturas altas, baja precipitación)
np.random.seed(2)  # Distinto de los anteriores para diversificar la muestra
dias = np.arange(365)
temperatura = 22 + 6 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.2, 365)
precipitacion = np.random.gamma(0.5, 0.8, 365) * (np.random.rand(365) > 0.85)  # Muy poca precipitación
consumo = 85 + 3.5 * temperatura - 0.5 * precipitacion + np.random.normal(0, 3, 365)

# Crear DataFrame
data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo
})

# Variables independientes y dependiente
X = data[['temperatura', 'precipitacion']]
y = data['consumo']

# División del conjunto
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Modelo
modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])
# Compilación y entrenamiento
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

# Predicción y evaluación
predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

# Resultados
print("\nResultados del escenario de sequía extrema:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura IX. Código Python para escenario sequía extrema

En este caso, se fija la semilla aleatoria número 2 para asegurar que la serie aleatoria sea diferente a la de los escenarios anteriores. Por lo demás, el código se inicializa de la misma manera que en los casos previos.

En cuanto a la temperatura, se simulará de forma que sea más elevada que en un caso normal, debido al escenario de sequía. Se usará la siguiente fórmula:

temperatura = 22 + 6 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.2, 365)

Se usa un valor de temperatura base mayor en este caso (22), representando un calentamiento generalizado del año, con una amplitud de la variación estacional de 6 en vez de 5, lo cual amplía el rango de temperaturas entre estaciones. Esto se hace porque en escenarios extremos, los veranos tienden a ser más cálidos e intensos, y los inviernos pueden seguir siendo templados. Este aumento de amplitud representa mejor el impacto de una ola de calor prolongada o un verano inusualmente cálido, afectando directamente la demanda de agua.

Además, se incluye la componente aleatoria de ruido con desviación estándar de 1,2 ºC en vez de los 1,5 ºC del escenario base. Mientras que en el caso inicial este valor de desviación representaba una variabilidad atmosférica moderada, como días nublados, cambios bruscos de temperatura, etc., en el escenario extremo de sequía se reduce debido a que en los períodos de sequía prolongada, el tiempo suele ser más estable y seco. Las masas de aire cálido y seco tienden a reducir la variabilidad térmica diaria, produciendo jornadas repetitivas con altas temperaturas y pocas fluctuaciones.

En cuanto a la precipitación, también se modifica la expresión ligeramente en relación al primer escenario. Para este caso de sequía, la fórmula será la siguiente:

precipitacion = np.random.gamma(0.5, 0.8, 365) * (np.random.rand(365) > 0.85)

Simulando lluvias más escasas que en el escenario inicial, por un lado suponiendo que en vez de llover el 30% de los días, sólo lloverá el 15%. Además, la distribución gamma ahora tendrá forma 0,5 y escala 0,8, diferente a la forma de 1,5 y escala 1 que presentaba el primer escenario. Reducir la forma un tercio hace que la mayoría de los valores sean muy pequeños, representando lluvias escasas, y si a eso le sumas reducir la escala esto hace que la magnitud de las precipitaciones se reduzca también.

Por último, el consumo se modelará mediante la siguiente expresión:

consumo = 85 + 3.5 * temperatura - 0.5 * precipitacion + np.random.normal(0, 3, 365)

Esta expresión se diferencia de aquella del escenario base en lo siguiente:

- Valor base de consumo diario medio de 85 en vez de 80. Se parte de un consumo estructural más alto debido al aumento de necesidades generales.
- Coeficiente de temperatura de 3,5 en vez de 3. Cuanto más alta sea la temperatura, más aumentará el consumo. Este mayor factor representa mayor sensibilidad: en olas de calor/sequía se consume más agua por persona, cultivos, refrigeración, etc.
- Coeficiente de precipitación de 0,5 en vez de 1,5, con signo negativo porque si llueve, se consume menos. Este menor valor significa que en escenarios secos, aunque llueva, la reducción en el consumo es menor (el terreno sigue seco).
- Se elige mantener la variabilidad de ruido en 3, sin asumir más incertidumbre.

De esta forma, una vez modeladas las tres variables de estudio, se procede de igual manera que en los casos anteriores. Se construye el DataFrame para el análisis y modelado, se separan las tres variables en las dos explicativas y la objetivo, se extrae el 80% de los datos para entrenamiento y el 20% restante para prueba, y se construye la red neuronal.

Por último, se compila y se entrena el modelo, procediendo a su evaluación y obtención de los resultados. Así, se obtienen los siguientes resultados:

```
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 1083.3685 - mae: 31.0704 - val_loss: 805.5838 - val_mae: 25.9405
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 802.6750 - mae: 25.8006 - val_loss: 601.0632 - val_mae: 21.3550
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 584.9431 - mae: 20.9672 - val_loss: 463.0694 - val_mae: 17.7849
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 477.9837 - mae: 18.0226 - val_loss: 373.4788 - val_mae: 15.3205
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 347.3196 - mae: 14.8328 - val_loss: 320.2499 - val_mae: 14.2035
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 337.8661 - mae: 15.1639 - val_loss: 290.4998 - val_mae: 13.6836
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 296.2810 - mae: 14.3570 - val_loss: 275.9778 - val_mae: 13.5252
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 266.9045 - mae: 13.9175 - val_loss: 270.0191 - val_mae: 13.5213
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 250.6210 - mae: 13.5888 - val_loss: 268.1077 - val_mae: 13.5403
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 265.4572 - mae: 14.0815 - val_loss: 268.0620 - val_mae: 13.5673
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - loss: 253.2607 - mae: 13.8033 - val_loss: 268.6940 - val_mae: 13.6327
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 257.2620 - mae: 14.0221 - val_loss: 269.3442 - val_mae: 13.6812
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 265.5478 - mae: 14.2560 - val_loss: 269.9536 - val_mae: 13.7211
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 255.3385 - mae: 14.0288 - val_loss: 269.8062 - val_mae: 13.7152
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 251.6640 - mae: 13.7570 - val_loss: 269.7528 - val_mae: 13.7151
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - loss: 261.6523 - mae: 13.9905 - val_loss: 269.5071 - val_mae: 13.7031
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step

Resultados del escenario de sequía extrema:
MAE: 12.82
RMSE: 14.97
R²: 0.04
```

Figura X. Resultados simulación caso sequía extrema

Los resultados del escenario de sequía extrema obtenidos con el modelo de predicción de demanda de agua son los siguientes:

- Error Absoluto Medio: 12.82
- Raíz del Error Cuadrático Medio: 14.97
- Coeficiente de Determinación: 0.04

El MAE (12.82) indica que, en promedio, el modelo se equivoca unos 12.82 litros por unidad de consumo simulada. El RMSE (14.97) mide el mismo error, pero penalizando más los errores grandes. Es ligeramente superior, como es habitual.

Estos valores son similares a los obtenidos en el escenario de ola de calor (MAE = 12.46 y RMSE = 14.16), lo que sugiere que el modelo mantiene una estabilidad relativa bajo condiciones extremas, aunque no logra mejorar la precisión. Por otro lado, el R^2 mide qué porcentaje de la variabilidad de los datos reales está explicada por el modelo. Un valor de 0.04 implica que el modelo solo explica un 4 % de la variabilidad de la demanda de agua en este escenario, lo que es muy bajo. Aunque es mejor que en los escenarios iniciales (donde llegó a ser negativo), sigue siendo un modelo poco explicativo, especialmente en situaciones más extremas como la sequía prolongada.

La comparativa con los escenarios anteriores se expone a continuación:

Métrica	Escenario base	Ola de calor	Sequía extrema
MAE	12,58	12,46	12,82
RMSE	15,08	14,16	14,97
R^2	-0,7	-0,11	0,04

Tabla II. Comparativa resultados simulaciones

A la vista de los resultados, el modelo reduce su error cuadrático en el escenario de ola de calor (mejor RMSE). En el escenario de sequía, aunque el R^2 mejora, sigue indicando que el modelo no es suficientemente bueno para capturar relaciones complejas bajo condiciones extremas. Por otro lado, el MAE se mantiene constante, lo que indica que el modelo es robusto pero no suficientemente preciso.

A partir de los resultados obtenidos, se propone continuar el desarrollo del sistema explorando nuevos escenarios extremos, incorporando variables adicionales relevantes y probando algoritmos alternativos que permitan mejorar la precisión y capacidad de generalización del modelo. Asimismo, se plantea analizar el impacto económico y operativo del sistema predictivo en la planificación del tratamiento de aguas regeneradas, así como

evaluar su comportamiento en situaciones de emergencia, normativa restrictiva o fallo técnico. Todo ello permitirá consolidar un sistema robusto, flexible y adaptado a las condiciones cambiantes de la gestión hídrica real.

Para conseguir esto, se probará con tres escenarios: lluvias intensas localizadas, para ver cómo afecta un exceso de precipitación al consumo de agua regenerada, restricciones normativas severas, para simular un escenario con limitación externa de caudal de reutilización, y fuga oculta o fallo en planta, esto es, crear un modelo donde parte del consumo se ve afectado por una pérdida no detectada.

Estos escenarios reflejarán condiciones reales que el sistema deberá gestionar dinámicamente.

Así, se simulará primero el escenario de lluvias intensas con el siguiente código:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

np.random.seed(3)
dias = np.arange(365)

temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)

# Lluvias localizadas intensas
precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)
eventos_intensos = np.random.choice(dias, size=20, replace=False)
precipitacion[eventos_intensos] += np.random.gamma(10.0, 4.0, size=20)  # lluvias intensas puntuales

# Consumo: disminuye más con lluvia intensa, pero con ruido alto
consumo = 80 + 3 * temperatura - 2.5 * precipitacion + np.random.normal(0, 3.5, 365)

data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo
})

X = data[['temperatura', 'precipitacion']]
y = data['consumo']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

print("\nResultados del escenario de lluvias intensas localizadas:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura XI. Código Python para simulación lluvias intensas

El código establece la misma expresión de temperatura que en el caso inicial, introduciendo novedades en la precipitación y el consumo.

Para la precipitación, se inicializa un vector de precipitación que suponga que llueve el 30% de los días del año, como en casos anteriores. Posteriormente, se seleccionarán 20 días

aleatorios en los que lloverá intensamente. Para estos días, la precipitación se genera usando una distribución Gamma para simular lluvias intensas y puntuales; forma 10 y escala 4, esto es, media de precipitación intensa (40 mm/día) en los 20 días.

El consumo se modela de forma similar a los casos anteriores, con un consumo base sumado del aumento lineal que tiene con la temperatura y una disminución intensa debido a las altas precipitaciones, añadido de un ruido.

De esta forma, la simulación en Python queda de la siguiente manera:

```
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 1189.6772 - mae: 33.0348 - val_loss: 963.1177 - val_mae: 29.4553
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 915.0734 - mae: 28.4139 - val_loss: 744.7039 - val_mae: 25.3827
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 736.6740 - mae: 24.9784 - val_loss: 576.4595 - val_mae: 21.6743
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 555.9706 - mae: 20.8475 - val_loss: 451.9445 - val_mae: 18.5024
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 455.5126 - mae: 18.2121 - val_loss: 361.9621 - val_mae: 15.8010
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 377.1063 - mae: 16.1338 - val_loss: 299.0678 - val_mae: 14.0141
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 280.9146 - mae: 13.2805 - val_loss: 257.4044 - val_mae: 12.9962
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 239.5323 - mae: 12.2232 - val_loss: 229.4107 - val_mae: 12.3394
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 254.9756 - mae: 13.1039 - val_loss: 211.0967 - val_mae: 11.9502
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step - loss: 223.4195 - mae: 12.3149 - val_loss: 200.5883 - val_mae: 11.6912
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 213.3674 - mae: 12.2967 - val_loss: 194.4162 - val_mae: 11.5507
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 200.3341 - mae: 11.8876 - val_loss: 191.1519 - val_mae: 11.4885
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 197.5820 - mae: 11.7437 - val_loss: 189.6014 - val_mae: 11.4789
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 206.7243 - mae: 12.3609 - val_loss: 188.8580 - val_mae: 11.4760
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 190.8703 - mae: 11.7859 - val_loss: 188.4043 - val_mae: 11.4702
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 210.6547 - mae: 12.2061 - val_loss: 188.4689 - val_mae: 11.4760
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step

Resultados del escenario de lluvias intensas localizadas:
MAE: 11.99
RMSE: 14.20
R²: 0.74

C:\Users\javie\OneDrive\Escritorio\ICAI 24-25\TFM>
```

Figura XII. Resultados simulación lluvias intensas localizadas

Se obtiene un MAE de 11,99, un RMSE de 14,2 y un R^2 de 0,74, lo que comparado con los tres escenarios anteriores supone una clara mejora en el rendimiento del modelo predictivo. El menor MAE supone un mejor ajuste promedio, mientras que una raíz del error cuadrático

medio de 14,2 es de las menores que han salido en los cuatro escenarios lo que implica que los errores grandes son penalizados menos. Por otro lado, un coeficiente de determinación de 0,74 supone una excelente mejora respecto a los otros escenarios donde este parámetro era muy bajo o incluso negativo, lo que indicaba que el modelo que hacía no era muy bueno.

De esta manera, se aprecia que el modelo mejora con este escenario de lluvias intensas localizadas. Esto puede ser debido a que, por ejemplo, el patrón entre lluvia y consumo es más marcado y predecible: cuando llueve mucho, el consumo baja más claramente.

Además, en los otros escenarios (ola de calor o sequía), las variables tienen menos variabilidad o el efecto es más complejo o no lineal (por ejemplo, una ola de calor puede aumentar o reducir consumo según el comportamiento humano). La precipitación intensa puntual ayuda a diferenciar mejor los días anómalos, lo cual puede facilitar el aprendizaje del modelo.

De esta manera, el escenario de lluvias intensas localizadas no solo mejora todas las métricas, sino que es el único con un coeficiente de determinación claramente positivo y alto. Esto sugiere que el modelo está captando mucho mejor las relaciones entre las variables en ese escenario.

Así, se propone un nuevo escenario que simule restricciones normativas severas, ya que es distinto a los anteriores en el sentido de que no es meteorológico y que refleja condiciones reales de operación de una planta regeneradora, como son los caudales mínimos legales o límites por normativa sanitaria, añadiendo una variable política y de gestión al modelo predictivo. Todo esto puede ilustrar cómo no basta solamente con diseñar un modelo predictivo sino que debe integrarse también con criterios regulatorios.

Así, se creará el código para simular este escenario con una limitación impuesta sobre el caudal máximo que puede regenerar la planta para por ejemplo la época de verano, y se muestra a continuación:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Semilla reproducibilidad
np.random.seed(4)
dias = np.arange(365)

# Temperatura estacional típica
temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)

# Precipitación normal
precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)

# Consumo sin restricciones
consumo_real = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)

# Restricciones normativas entre los días 150 y 250 (verano)
# Se impone un límite máximo al caudal regenerado, simulando normativa externa
limite_regeneracion = 140

# Inicialmente el consumo restringido es igual al real
consumo_restringido = consumo_real.copy()

# Aplica el límite de caudal solo entre los días 150 y 250
consumo_restringido[150:251] = np.minimum(consumo_real[150:251], limite_regeneracion)


# DataFrame
data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo_restringido
})
```

```
# Separar variables predictoras y objetivo
X = data[['temperatura', 'precipitacion']]
y = data['consumo']

# División del conjunto
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Modelo de red neuronal
modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])

# Compilar y entrenar
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

# Evaluación
predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

# Resultados
print("\nResultados del escenario con restricciones normativas:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura XIII. Código Python para escenario restricciones normativas

El modelo simula un año normal, pero entre los días 150 y 250 (verano), se limita el consumo de agua a un máximo de 140 unidades. Esto simula, por ejemplo, una limitación legal sobre el volumen que puede tratar la planta o una orden de la autoridad hidráulica por razones sanitarias o ecológicas.

Las tres variables de estudio se modelan de igual manera que en el escenario inicial, pero para el consumo se simula una restricción normativa externa de 140 m³/día, como un límite legal de caudal regenerado, para la época veraniega (días 150 a 250 del año).

Este valor de consumo de 140 m³/día representa el caudal máximo diario que la normativa externa permite regenerar y suministrar, por ejemplo, durante un periodo de restricción como es el verano. El consumo formulado en el modelo puede subir fácilmente a 130 o 150 m³/día

para los meses cálidos. De esta forma, el valor de 140 impone una restricción "ligera pero efectiva", ya que crea una zona de conflicto entre la demanda y la normativa, lo cual es interesante para evaluar la respuesta del modelo.

Así, se ejecuta el código y se obtienen los siguientes resultados de simulación:

```
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 246.5687 - mae: 13.3095 - val_loss: 233.9484 - val_mae: 13.2634
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 247.1171 - mae: 13.2198 - val_loss: 233.8235 - val_mae: 13.2731
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 252.3991 - mae: 13.4170 - val_loss: 233.6034 - val_mae: 13.2769
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 240.3220 - mae: 13.0637 - val_loss: 233.5038 - val_mae: 13.2853
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 243.3630 - mae: 13.2860 - val_loss: 232.6639 - val_mae: 13.2644
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 239.2582 - mae: 13.0715 - val_loss: 232.2173 - val_mae: 13.2580
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 252.5262 - mae: 13.5169 - val_loss: 232.2342 - val_mae: 13.2729
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 235.1691 - mae: 12.8682 - val_loss: 231.5343 - val_mae: 13.2548
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 229.3416 - mae: 12.8769 - val_loss: 230.9368 - val_mae: 13.2383
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 250.9185 - mae: 13.3400 - val_loss: 230.4592 - val_mae: 13.2308
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 230.8207 - mae: 12.7348 - val_loss: 230.2081 - val_mae: 13.2342
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 238.7273 - mae: 13.0392 - val_loss: 229.9409 - val_mae: 13.2359
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 243.0802 - mae: 13.2891 - val_loss: 229.6037 - val_mae: 13.2339
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 225.1862 - mae: 12.5998 - val_loss: 229.2548 - val_mae: 13.2306
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 225.8576 - mae: 12.7335 - val_loss: 229.1095 - val_mae: 13.2379
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step

Resultados del escenario con restricciones normativas:
MAE: 11.53
RMSE: 13.19
R²: -0.37

C:\Users\javie\OneDrive\Escritorio\ICAI 24-25\TFM>
```

Figura XIV. Resultados simulación restricciones normativas

Los resultados de la simulación para restricciones normativas de consumo en verano son los siguientes:

- MAE: 11,53
- RMSE: 13,19
- R^2: -0,37

El error absoluto medio y la raíz del error cuadrático medio han mejorado respecto a los escenarios anteriores, lo que refuerza aún más el modelo. Esto sugiere que, a pesar de la restricción artificial en el verano, el modelo se adapta mejor cuando la anomalía está localizada en el tiempo; comete pocos errores grandes en términos absolutos, y en los tramos sin restricción acierta bastante. Por otro lado, el coeficiente de determinación es negativo, ha caído de 0,74 en el escenario de lluvias intensas a -0,37 en este.

Esto se puede deber a que no hay una variable que indique la causa del cambio; el consumo ha sido manipulado artificialmente. No existe ninguna variable de entrada que diga que en verano hay una limitación externa. Solamente se usó *np.minimum()* para forzar un límite en los días 150–250, lo que podría haber generado errores estructurales, que son muy penalizados en el cálculo de R^2, ya que este parámetro es sensible a desviaciones sistemáticas; si el modelo se equivoca mucho en una zona (verano), aunque acierte en otras, el R^2 baja drásticamente.

En resumen, un coeficiente de determinación negativo significa que el modelo carece de explicación para esta desviación estructural del consumo en el tramo veraniego (no sabe que existe esa normativa).

Visto este escenario, se probará con el último escenario propuesto inicialmente, un escenario más técnico como es el de una fuga oculta o fallo en la planta. El objetivo será simular una situación en la que parte del agua regenerada se pierde por una fuga no detectada o mal funcionamiento de un proceso de la planta, lo que hará que el consumo real será menor de lo esperado.

Se busca analizar si el modelo puede detectar una caída en el consumo en estas condiciones, y la magnitud de cuánto afecta este fallo oculto a las variables de estudio. De esta forma, se modificará el consumo real de forma selectiva, restando un caudal constante (simulando fuga o pérdida de eficiencia), sin que el modelo lo sepa. Por ejemplo, se establecerá una fuga de

agua de 10 m^3/día para una época del año, antes de que este fallo técnico se localice y se arregle. Para el resto del año, el consumo será normal.

De esta forma, el código implementado se muestra a continuación:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Semilla reproducibilidad
np.random.seed(5)
dias = np.arange(365)

# Temperatura
temperatura = 20 + 5 * np.sin(2 * np.pi * dias / 365) + np.random.normal(0, 1.5, 365)

# Precipitación
precipitacion = np.random.gamma(1.5, 1.0, 365) * (np.random.rand(365) > 0.7)

# Consumo estimado sin fugas
consumo_real = 80 + 3 * temperatura - 1.5 * precipitacion + np.random.normal(0, 3, 365)

# Simulación fuga técnica entre los días 180 y 220
consumo_con_fuga = consumo_real.copy()
consumo_con_fuga[180:221] -= 10  # pérdida de 10 m3/dia

# DataFrame con la fuga oculta
data = pd.DataFrame({
    'temperatura': temperatura,
    'precipitacion': precipitacion,
    'consumo': consumo_con_fuga
})

# Separación de variables
X = data[['temperatura', 'precipitacion']]
y = data['consumo']
```

```
# División del conjunto de datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Definición del modelo
modelo = keras.Sequential([
    keras.layers.Dense(32, activation='relu', input_shape=(2,)),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(1)
])

# Compilación y entrenamiento
modelo.compile(optimizer='adam', loss='mse', metrics=['mae'])
modelo.fit(X_train, y_train, epochs=50, verbose=1, validation_split=0.2)

# Evaluación
predicciones = modelo.predict(X_test).flatten()
mae = mean_absolute_error(y_test, predicciones)
rmse = np.sqrt(mean_squared_error(y_test, predicciones))
r2 = r2_score(y_test, predicciones)

# Mostrar resultados
print("\nResultados del escenario con fuga oculta o fallo en planta:")
print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Figura XV. Código Python para simulación fuga oculta

Así, el código se desarrolla de igual manera que los casos anteriores con la novedad del consumo con fuga. Se simula una fuga oculta entre los días 180 y 220, periodo durante el cual el consumo cae 10 m^3/día. Se trata de una fuga invisible ya que se modela de forma inesperada.

Así, se ejecuta el código con estas variables y los resultados se muestran a continuación:

```
Epoch 33/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 206.7845 - mae: 11.9673 - val_loss: 189.7554 - val_mae: 11.5466
Epoch 34/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 220.1672 - mae: 12.1523 - val_loss: 189.6790 - val_mae: 11.5464
Epoch 35/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 210.2477 - mae: 11.7762 - val_loss: 189.7129 - val_mae: 11.5616
Epoch 36/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - loss: 222.5582 - mae: 12.2206 - val_loss: 189.7170 - val_mae: 11.5713
Epoch 37/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 187.7557 - mae: 11.1928 - val_loss: 189.7398 - val_mae: 11.5809
Epoch 38/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 211.4297 - mae: 11.8529 - val_loss: 189.5842 - val_mae: 11.5738
Epoch 39/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 201.5123 - mae: 11.6414 - val_loss: 189.3483 - val_mae: 11.5582
Epoch 40/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - loss: 212.8414 - mae: 11.9994 - val_loss: 189.1416 - val_mae: 11.5439
Epoch 41/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step - loss: 214.7546 - mae: 12.0351 - val_loss: 189.0452 - val_mae: 11.5407
Epoch 42/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 209.9171 - mae: 11.9379 - val_loss: 188.8898 - val_mae: 11.5305
Epoch 43/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 189.0822 - mae: 11.0865 - val_loss: 188.7868 - val_mae: 11.5280
Epoch 44/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 204.1903 - mae: 11.6196 - val_loss: 188.7349 - val_mae: 11.5322
Epoch 45/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 194.0269 - mae: 11.3480 - val_loss: 188.6683 - val_mae: 11.5345
Epoch 46/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 214.1924 - mae: 11.9408 - val_loss: 188.5683 - val_mae: 11.5311
Epoch 47/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 214.4761 - mae: 12.1205 - val_loss: 188.5275 - val_mae: 11.5366
Epoch 48/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 220.3495 - mae: 12.4748 - val_loss: 188.4855 - val_mae: 11.5415
Epoch 49/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step - loss: 203.7349 - mae: 11.7643 - val_loss: 188.2045 - val_mae: 11.5199
Epoch 50/50
8/8 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 200.9832 - mae: 11.7000 - val_loss: 188.0705 - val_mae: 11.5138
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step

Resultados del escenario con fuga oculta o fallo en planta:
MAE: 12.49
RMSE: 14.50
R²: -0.49

C:\Users\javie\OneDrive\Escritorio\ICAI 24-25\TFM>
```

Figura XVI. Resultados simulación fuga oculta

Los resultados para este escenario han sido un MAE de 12,49, un RMSE de 14,5 y un R^2 de -0,49. El error absoluto medio y raíz del error cuadrático medio son moderados, lo que indica que el modelo sigue en su línea de no cometer grandes errores. Como la fuga está localizada en el tiempo, el error se diluye y no se ve muy afectado en valor absoluto. Sin embargo, el coeficiente de determinación negativo indica que el modelo no explica bien la variabilidad. El modelo no sabe que ha habido una fuga, por lo que predice valores más altos de los reales en ese periodo. Esta desviación sistemática reduce mucho el coeficiente de determinación porque el modelo falla al capturar el cambio estructural inesperado.

De esta forma, una vez analizados los seis escenarios de estudio, se aporta una tabla comparativa de resultados a continuación:

Escenario	MAE	RMSE	R^2
Base	12,58	15,08	-0,7
Ola de calor	12,46	14,16	-0,11
Sequía extrema	12,82	14,97	0,04
Lluvias intensas localizadas	11,99	14,2	0,74
Restricción normativa	11,53	13,19	-0,37
Fuga oculta	12,49	14,5	-0,49

Tabla III. Comparación seis escenarios de estudio

A la vista de los escenarios simulados está que el mejor resultado general lo da el caso de lluvias intensas localizadas. Un coeficiente de determinación alto como es 0,74 indica que el modelo explica correctamente la variabilidad del consumo en este contexto, y un error absoluto medio y raíz del error cuadrático medio bajos indica que se trata de un modelo muy ajustado.

Por otro lado, los peores resultados vienen del escenario base y del de fuga oculta en la planta. En el escenario base hay un coeficiente de determinación de -0,7, lo que significa que el modelo no logra explicar bien la variabilidad natural, quizás por falta de variables.

En el escenario de fuga técnica no detectada, el modelo tampoco identifica la caída de consumo, con un $R^2 = -0{,}49$. El MAE y RMSE son razonables porque el fallo está acotado en el tiempo, pero el modelo no aprende la causa.

Seguidamente, en el escenario con restricciones normativas, aunque el modelo comete menos errores absolutos (MAE y RMSE más bajos), el coeficiente de determinación sigue siendo negativo. Esto demuestra que el modelo predice con precisión general, pero no explica bien el patrón anómalo en verano debido a la restricción externa no modelada.

Por último, en los escenarios climáticos extremos (ola de calor y sequía), el modelo se comporta de forma bastante estable: MAE y RMSE similares al caso base, mientras que el coeficiente de determinación mejora un poco: la sequía tiene un patrón que el modelo puede aprender, aunque no sea fuerte.

La conclusión que se saca de aquí es que en escenarios donde se introducen condiciones operativas o técnicas que el modelo no puede observar directamente (como restricciones normativas o fallos técnicos), el R^2 cae por debajo de cero, indicando que el modelo no logra explicar estructuralmente esos cambios. A pesar de ello, los errores absolutos (MAE y RMSE) se mantienen bajos, lo que sugiere que el modelo sigue siendo útil para la estimación general, aunque no para situaciones no explícitamente representadas en los datos de entrada.

En cuanto a propuestas de mejora del modelo, a partir de las simulaciones realizadas, se propone como mejora ampliar la base de datos con más años y variables (como humedad, días festivos o nivel de embalses), probar otras arquitecturas más profundas o recurrencias (como LSTM), y ajustar hiperparámetros mediante búsqueda automática (grid search o random search). Asimismo, se recomienda implementar técnicas de validación cruzada para mejorar la generalización del modelo.

CAPÍTULO 6. ANÁLISIS ECONÓMICO

En este capítulo se estimarán los costes asociados al sistema de reutilización de aguas residuales y la integración del modelo predictivo basado en IA, diferenciando entre:

- CAPEX: costes de inversión o construcción
- OPEX: costes de operación y mantenimiento
- Coste unitario del agua regenerada
- Comparación económica con agua potable
- Reflexión sobre escalabilidad y retorno

En primer lugar, se tienen los costes de construcción, los cuales dependen del tamaño, el tipo de tecnologías incluidas y el grado de automatización del sistema modular de tratamiento de agua. Para este trabajo, se considera una planta tipo para un entorno urbano o agrícola de tamaño medio, capaz de tratar entre 500 y 1.000 m^3/día. Una estimación de estos CAPEX se ve reflejada en la siguiente tabla:

CAPEX	Coste estimado (€)
Pretratamiento, tanques	70.000
Sistema de filtración	120.000
Desinfección	50.000
Instrumentación y control y eléctrico	35.000
Obra, montaje e instalación pruebas y puesta en marcha	100.000
Desarrollo, integración sistema predictivo	15.000
Total	390.000

Tabla IV. CAPEX

Seguidamente, se estimarán los costes operativos, que incluyen energía, personal, mantenimiento, sustitución de consumibles, reactivos (en caso de cloración) y costes indirectos. Para el consumo energético, se considera un valor promedio de 0,6 kWh/m^3 para sistemas terciarios compactos, automatizados y orientados a usos no potables. Así, considerando la planta en cuestión:

OPEX	Coste anual (€)
Consumo energético	30.000
Mantenimiento, repuestos	12.000
Personal técnico	18.000
Gestión, control digital	5.000
Total	65.000

Tabla V. OPEX

De esta forma, una vez estimados los costes de inversión y de operación, se puede obtener el coste unitario para el agua regenerada. Estimando una producción anual de agua regenerada de 150.000 $m^3/año$, lo que equivale a algo más de 400 $m^3/día$, el coste total del primer año sería una suma de los CAPEX y los OPEX, unos 455.000€.

Para los años siguientes, el coste sería de los OPEX, unos 65.000€ al año. Con esto se puede calcular el coste unitario del agua regenerada:

$$Coste\ unitario\ primer\ año = \frac{455.000€}{150.000m^3} = 3{,}03€/m^3$$

$$Coste\ unitario\ siguientes\ años = \frac{65.000€}{150.000m^3} = 0{,}43€/m^3$$

Lo cual supone un ahorro considerable con respecto al coste de agua potable en entornos agrícolas o urbanos, que oscila entre los 0,8 y 1,2 €/m^3.

El sistema propuesto, al ser modular, permite adaptar la capacidad y reducir costes por economía de escala: una planta mayor, de aproximadamente 3.000 m^3/día, podría reducir el coste unitario a menos de 0,30 €/m^3. En entornos con alta presión sobre el agua potable o

restricciones legales, el uso de agua regenerada puede generar ahorro neto, especialmente en usos como riego de zonas verdes, limpieza viaria o caudales ecológicos.

Además, la integración del modelo predictivo de demanda permite optimizar el uso energético, reducir el sobredimensionamiento y minimizar el vertido de excedentes, mejorando la eficiencia económica de la planta.

6.1 EVALUACIÓN RENTABILIDAD PROYECTO

Además del análisis de costes de inversión y operación, resulta fundamental evaluar la viabilidad económica a medio plazo del sistema propuesto. Para ello, se calculan tres indicadores clásicos de rentabilidad: el Valor Actual Neto (VAN), la Tasa Interna de Retorno (TIR) y el Periodo de Retorno (Pay-back).

Para ello, se parten de unos supuestos de análisis para un horizonte temporal de 10 años:

- Inversión inicial (CAPEX): 390.000 €
- Costes de operación y mantenimiento (OPEX anual): 65.000 €
- Producción anual de agua regenerada: 150.000 m³
- Ahorro generado por sustituir agua potable (1 €/m³): 150.000 €/año
- Ahorro neto anual (beneficio económico): 85.000 €/año
- Tipo de descuento aplicado: 5 %

De esta forma, se procede al cálculo de los indicadores de rentabilidad:

1. **Valor Actual Neto (VAN):**

Se calcula con la siguiente fórmula:

$$VAN = \sum_{t=1}^{n} \frac{F_t}{(1+r)^t} - I_o$$

Siendo:

- F_t: flujo de caja neto en el año t
- r: tasa de descuento

- n: número de años del proyecto
- I_o: inversión inicial del proyecto

De esta forma, para las hipótesis propuestas, el VAN se obtendría de la siguiente manera:

$$VAN = \sum_{t=1}^{10} \frac{85.000}{(1+0{,}05)^t} - 390.000 = 266.255€$$

El Valor Actual Neto da positivo, lo que indica que el proyecto genera valor económico.

2. Tasa Interna de Retorno (TIR)

La TIR es la tasa de descuento que hace que el VAN sea cero. Para los flujos considerados:

$$VAN = \sum_{t=1}^{10} \frac{85.000}{(1+TIR)^t} - 390.000 = 0 \rightarrow TIR = 17\%$$

Por lo general, una TIR superior al 5% implica que el proyecto genera una rentabilidad muy atractiva.

3. Periodo de Retorno (Pay-back)

El Pay-back indica cuántos años son necesarios para recuperar la inversión inicial, y se calcula usando la siguiente fórmula:

$$Pay - back = \frac{I_o}{F_t}$$

Lo que para los datos en cuestión queda de la siguiente manera:

$$Pay - back = \frac{390.000€}{85.000 \frac{€}{año}} = 4{,}6\ años$$

Lo que indica que en menos de 5 años se habrá recuperado la inversión inicial, lo que resulta razonable para un proyecto de estas características.

Las conclusiones son, por tanto, que los resultados muestran que el sistema propuesto de reutilización de aguas residuales y predicción de la demanda presenta una viabilidad económica sólida. Con un VAN positivo, una TIR elevada y un período de retorno moderado, el proyecto no solo resulta técnicamente viable, sino también rentable desde el punto de vista financiero.

CAPÍTULO 7. ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

El presente trabajo se enmarca dentro de los principios de la Agenda 2030 de Naciones Unidas y contribuye directamente a varios de los Objetivos de Desarrollo Sostenible (ODS), especialmente en el ámbito de la gestión del agua, la sostenibilidad urbana y la acción climática.

A continuación, se detallan los principales ODS con los que el sistema propuesto guarda una relación directa:

1. **ODS 6: Agua limpia y saneamiento.** Garantizar la disponibilidad de agua y su gestión sostenible y el saneamiento para todos.

El sistema desarrollado propone una solución concreta para la reutilización segura y eficiente de aguas residuales tratadas, contribuyendo al uso racional del recurso hídrico. A través del diseño de tecnologías de tratamiento terciario y la incorporación de un modelo predictivo de demanda, se optimiza la regeneración y se minimizan vertidos innecesarios, lo cual se alinea con las metas:

- **Meta 6.3**: Mejorar la calidad del agua reduciendo la contaminación, eliminando vertidos y aumentando el reciclado y la reutilización en condiciones de seguridad.
- **Meta 6.4**: Aumentar considerablemente el uso eficiente del agua en todos los sectores.

2. **ODS 11: Ciudades y comunidades sostenibles.** Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles.

La propuesta permite integrar sistemas descentralizados de reutilización en contextos urbanos y periurbanos, facilitando el riego de zonas verdes, la limpieza viaria o el mantenimiento de caudales ecológicos sin depender del agua potable. Además, al incorporar herramientas de predicción con IA, se favorece una gestión urbana inteligente de los recursos.

- **Meta 11.6**: Reducir el impacto ambiental negativo per cápita de las ciudades, prestando especial atención a la calidad del aire y a la gestión de los residuos líquidos.
- **Meta 11.b**: Aumentar el número de ciudades resilientes mediante la adopción de políticas integradas de mitigación y adaptación al cambio climático.

3. **ODS 13: Acción por el clima.** Adoptar medidas urgentes para combatir el cambio climático y sus efectos.

El proyecto contribuye a la adaptación frente a escenarios de escasez hídrica derivados del cambio climático, especialmente en zonas urbanas con presión sobre el recurso. Asimismo, al sustituir parcialmente el uso de agua potable por regenerada y optimizar los procesos operativos mediante IA, se reducen indirectamente las emisiones de gases de efecto invernadero ligadas a la captación, transporte y bombeo de agua convencional.

- **Meta 13.1**: Fortalecer la resiliencia y la capacidad de adaptación a los riesgos relacionados con el clima.
- **Meta 13.3**: Mejorar la educación, la sensibilización y la capacidad institucional sobre mitigación del cambio climático y su reducción.

CAPÍTULO 8. CONCLUSIONES Y TRABAJOS FUTUROS

El presente Trabajo Fin de Máster ha desarrollado un sistema conceptual completo para la reutilización de aguas residuales combinada con un modelo de predicción inteligente de la demanda, aplicando técnicas de inteligencia artificial basadas en redes neuronales. El proyecto se ha estructurado en torno a dos pilares: un sistema modular y escalable de tratamiento y regeneración de aguas para usos no potables, y un modelo predictivo diseñado para anticipar la demanda de agua a partir de variables climáticas.

Entre los logros alcanzados, destacan:

- El diseño de una arquitectura funcional integrada, que enlaza la predicción de la demanda con la planificación del caudal a regenerar, optimizando el funcionamiento de la planta y reduciendo el desperdicio.
- La implementación y entrenamiento de modelos predictivos con TensorFlow, evaluando su comportamiento bajo seis escenarios representativos (condiciones normales, ola de calor, sequía extrema, lluvias intensas, restricciones normativas y fallos técnicos).
- La obtención de resultados comparativos mediante métricas como MAE, RMSE y R^2, que han permitido analizar la sensibilidad del modelo ante diferentes condiciones operativas y climáticas.
- El desarrollo de un análisis económico estimado, mostrando que el sistema puede ser económicamente competitivo frente al suministro de agua potable convencional.
- La identificación de una alineación clara con varios Objetivos de Desarrollo Sostenible (ODS), en particular el ODS 6 (agua limpia y saneamiento), ODS 11 (ciudades sostenibles) y ODS 13 (acción por el clima).

Las aportaciones personales del autor incluyen el diseño completo del sistema, la programación de los modelos en Python, la simulación de escenarios extremos, el análisis crítico de resultados, y la propuesta de integración entre IA y control operativo del sistema de regeneración.

En cuanto a las líneas de trabajo futuro, a partir del trabajo realizado, se identifican varias oportunidades de mejora y desarrollo:

- Ampliar la base de datos utilizada para el entrenamiento del modelo, incorporando múltiples años, diferentes zonas geográficas y variables adicionales como humedad, consumo por sectores o nivel de embalses.
- Explorar arquitecturas más complejas, como redes neuronales recurrentes (RNN, LSTM) que puedan modelar mejor la secuencia temporal del consumo de agua.
- Aplicar métodos de optimización de hiperparámetros, como búsqueda en rejilla (grid search) o algoritmos genéticos.
- Incluir variables de control operativo y normativo directamente en el modelo predictivo, para mejorar el rendimiento en escenarios donde hay restricciones externas o fallos internos.
- Evaluar la viabilidad técnica y regulatoria de la implantación real de este tipo de sistemas en entornos urbanos o agrícolas específicos.

En conjunto, este trabajo muestra que la aplicación de inteligencia artificial en la gestión del agua puede contribuir significativamente a la eficiencia, sostenibilidad y resiliencia de los sistemas hídricos, especialmente en un contexto de cambio climático y presión creciente sobre los recursos naturales.

Capítulo 9. BIBLIOGRAFÍA

[1] Cadena SER, Marzo 2025. "Los expertos apuestan por combinar de manera inteligente todos los recursos hídricos existentes para garantizar el agua en la provincia de Alicante".

[2] Eurecat, Mayo 2024. "Aplican inteligencia artificial para la reutilización de aguas residuales y sus nutrientes en el sector agrícola".

[3] Cadena SER, Marzo 2025. "El gran reto: depurar las aguas residuales".

[4] TecnoAqua, Noviembre 2022. "Inteligencia artificial para optimizar la recuperación de compuestos orgánicos de aguas residuales".

[5] Fluence, Julio 2023. "Explorando los métodos y tecnologías de tratamiento de aguas residuales industriales".

[6] Química ecológica, Mayo 2023. "Tecnologías emergentes para el tratamiento de aguas residuales".

[7] Cadena SER, Marzo 2025. "Canals quiere este verano un "Gorg de l'Omet" sin mosquitos y sin malos olores".

[8] Idrica, Agosto 2023. "La revolución tecnológica en las plantas de tratamiento de aguas residuales".

[9] Óleo Revista, Abril 2024. "Modelo 'Transformer' para la predicción de demanda de agua en agricultura".

[10] Phytoma España, Sanidad Vegetal, Abril 2024. "La UCO desarrolla un modelo de IA para predecir la demanda real de agua y energía".

[11] Universidad de Córdoba, Junio 2023. "Inteligencia artificial para prever el agua de riego a 7 días vista".

[12] Ciencia Gandía, Septiembre 2024. "Inteligencia artificial para gestionar mejor el agua en cuencas hidrográficas".

[13] La Ecuación Digital, Marzo 2025. "Cinco aplicaciones de inteligencia artificial que revolucionan la gestión del agua en 2025".

[14] Sener, Marzo 2024. "IA y Machine Learning en las disciplinas del agua y el medioambiente: del hidrólogo al "hidroinformático".

[15] Cadena SER, Marzo 2025. "El HUN y Navarrabiomed lideran un proyecto europeo sobre el tratamiento de aguas residuales de los hospitales".

[16] Cadena SER, Marzo 2025. "Esamur celebra el Día Mundial del Agua con un compromiso por la gestión sostenible de este recurso en la Región de Murcia".

[17] Aquacorp, Enero 2025. "Proyecto iRain".

[18] Idrica, Marzo 2025. "Cinco aplicaciones claves de la IA para transformar la gestión del agua en 2025".

[19] iAgua, Abril 2023. "¿Cómo se aplica la inteligencia artificial en la gestión del agua?".

[20] arXiv, Noviembre 2024. "Integrated Water Resource Management in the Segura Hydrographic Basin: An Artificial Intelligence Approach".

[21] IMNovation Hub. "IA para una gestión inteligente del ciclo del agua".

[22] IMNovation Hub. "Cómo conservar la biodiversidad por medio de la innovación".

[23] Plan de gestión del distrito de cuenca fluvial de Catalunya 2022-2027, 2021. "Anexo II. Caudales de mantenimiento o ecológicos y requerimientos hídricos".

[24] Diario oficial Generalitat Cataluña, 2024. "Plan de sequía – DOGC 8039".

[25] Aigües de Barcelona. "Agua regenerada, la solución más sostenible ante la emergencia hídrica".

[26] Área Metropolitana de Barcelona. "Reutilización".

[27] Revista digital UCE, 2016. "Modelo para el pronóstico de la demanda real de agua potable en Quito".

[28] Universidad Técnica del Norte, 2023. "Modelo para el pronóstico de la demanda de agua potable aplicando modelos de inteligencia artificial en la portada del cantón Mira".

[29] Universidad Politécnica Salesiana, 2023. "Aplicación de inteligencia artificial para pronosticar la demanda energética del sector de Primavera 2 con su respectivo alimentador de la subestación eléctrica Durán Sur con nivel de tensión de 69/13,8 KV".

[30] Universidad Nacional Federico Villareal, 2024. "Modelo predictivo basado en redes neuronales artificiales para pronosticar el consumo de agua potable en la ciudad de Iquitos".

[31] Sener, 2024. "IA y Machine Learning en las disciplinas del agua y el medioambiente: del hidrólogo al "hidroinformático"".

[32] Phytoma, 2024. "La UCO desarrolla un modelo de IA para predecir la demanda real de agua y energía".

[33] Cadena SER, 2024. "Arranca 'Canalbrain'. Grupo operativo para el Control Inteligente de Canales".

[34] 42 Madrid, 2020. "El sentido de la vida, el universo y todo lo demás: la respuesta es 42".

[35] Scikit Learn. "train_test_split".

[36] GeeksforGeeks, 2024. "What is Scikit-learn Random State in Splitting Dataset?".

[37] Medium (Kishan Modasiya), 2022. "What is random_state?".

[38] The Clever Programmer, 2020. "Why Random_state=42 in Machine Learning?".

[39] Stack Overflow, 2021. "What is "random-state" in sklearn.model_selection.train_test_split example?".

[40] arXiv, 2014. "Adam: A Method for Stochastic Optimization".

[41] TensorFlow, 2024. "tf.keras.optimizers.Adam".

[42] GeeksforGeeks, 2025. "What is Adam Optimizer?".

[43] ScienceDirect, 1999. "On the momentum term in gradient descent learning algorithms".

[44] Neural Networks for Machine Learning, 2012. "RMSProp: Divide the gradient by a running average of its recent magnitude".

[45] Inter-Research Science Publisher, 2005. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance".

[46] European Geosciences Union, 2014. "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature".

[47] Advancing Earth and Space Sciences, 2015. "Water Resources Research. What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models".

[48] Journal of Hydroinformatics, 2000. "Forecasting high waters at Venice Lagoon using chaotic time series analysis and nonlinear neural networks".

[49] ScienceDirect, 2020. "Land use optimization for nutrient reduction under stochastic precipitation rates".

[50] NumPy. "numpy.random.seed".

[51] Scikit-learn. "Glossary of Common Terms and API Elements".

[52] Machine Learning Mastery, 2021. "Regression Metrics for Machine Learning".

[53] ScienceDirect, 2022. "Urban flood numerical simulation: Research, methods and future perspectives".

[54] Journal of Hydroinformatics, 2023. "Coupled sink and flow accumulation analyses with single flow direction and multiple flow direction algorithms".

[55] The Intergovernmental Panel on Climate Change, 2022. "IPCC Sixth Assessment Report".

[56] ScienceDirect, 2020. "Certain trends in uncertainty and sensitivity analysis: An overview of software tools and techniques".

[57] TensorFlow, 2022. "Pronóstico de series de tiempo".

[58] ScienceDirect, 2002. "Comparison of streamflow between pre and post timber harvesting in Catamaran Brook (Canada)".

[59] Canal de Isabel II, 2024. "Informe de Sostenibilidad 2024".

[60] Scikit-learn. "3.4. Metrics and scoring: quantifying the quality of predictions".

[61] Journal of Chemical Information and Modeling, 2015. "Beware of R^2: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models".

[62] ScienceDirect, 2019. "Assessing the cover crop effect on soil hydraulic properties by inverse modelling in a 10-year field trial".

[63] European Geosciences Union, 2018. "A global approach to estimate irrigated areas – a comparison between different data and statistics".

[64] TensorFlow, 2022. "Regresion Basica: Predecir eficiencia de gasolina".

[65] Rob J Hyndman and George Athanasopoulos, 2018. "Forecasting: Principles and Practice".

[66] Boletín Oficial del Estado, 2007. "Real Decreto 1620/2007, de 7 de diciembre, por el que se establece el régimen jurídico de la reutilización de las aguas depuradas".

[67] European Union, 2020. "Reglamento (UE) 2020/741 del Parlamento Europeo y del Consejo de 25 de mayo de 2020 relativo a los requisitos mínimos para la reutilización del agua (Texto pertinente a efectos del EEE)".

[68] Jake VanderPlas, 2016. "Python Data Science Handbook".

[69] Machine Learning Mastery, 2020. "Metrics To Evaluate Machine Learning Algorithms in Python".

[70] FAO, 2010. "Reutilización del agua en la agricultura: ¿Beneficios para todos?".

[71] INE, 2022. "Estadística sobre el Suministro y Saneamiento del Agua".

[72] MITECO, 2022. "Planificación Hidrológica".

[73] Naciones Unidas. "Objetivo 6: Garantizar la disponibilidad de agua y su gestión sostenible y el saneamiento para todos".

[74] Naciones Unidas. "Objetivo 11: Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles".

[75] Naciones Unidas. "Lucha contra el cambio climático".

[76] Naciones Unidas. "Transforming our world: the 2030 Agenda for Sustainable Development".

[77] World Health Organization, 2019. "Safety and Quality of Water Used in Food Production and Processing".

[78] UN Habitat. "Water and Sanitation".

[79] Machine Learning Mastery, 2016. "Need Help Getting Started with Applied Machine Learning? These are the Step-by-Step Guides that You've Been Looking For!".

[80] Cornell University, 2017. "Adam: A Method for Stochastic Optimization".

[81] UNESCO, 2018. "The United Nations world water development report 2018: nature-based solutions for water".

[82] OECD, 2021. "Biodiversity, water and ecosystems".