Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

Predicción de la demanda de agua en las ETAPs de Valladolid

Autor:

Hernández Gómez, Diego

Tutoras:

De La Fuente Aparicio, María Jesús

Departamento de Ingeniería de Sistemas y Automática

González Peña, María José

Aquavall

Valladolid, julio 2025.

ÍNDICE

RESUMEN

Este trabajo tiene como objetivo desarrollar un modelo capaz de predecir la demanda diaria de agua potable en la ciudad de Valladolid, utilizando redes neuronales del tipo LSTM (Long Short-Term Memory). Para ello, se ha recopilado y tratado una base de datos histórica de caudal suministrado por las estaciones de tratamiento de agua (ETAP), complementada con información meteorológica. Se ha llevado a cabo un proceso completo de análisis, preprocesamiento, diseño del modelo, entrenamiento y evaluación. Se han comparado distintas configuraciones de red y enfoques predictivos (univariado y multivariado, con predicción a uno y varios días vista). Los resultados muestran que la inclusión de variables climáticas y un buen tratamiento de los datos iniciales mejoran significativamente la precisión del modelo. Este trabajo demuestra el potencial de la inteligencia artificial para apoyar la planificación y gestión eficiente del recurso hídrico en entornos urbanos.

Palabras clave: predicción de demanda, redes neuronales, LSTM, agua potable, series temporales.

ABSTRACT

This project aims to develop a model capable of predicting the daily demand for drinking water in the city of Valladolid using Long Short-Term Memory (LSTM) neural networks. For this purpose, a historical dataset of water flow from treatment plants (ETAPs) has been compiled and processed, along with complementary meteorological data. A complete pipeline was carried out, including analysis, preprocessing, model design, training, and evaluation. Different network configurations and forecasting approaches were compared (univariate and multivariate, with one-day and multi-day horizons). The results show that including climate variables and properly preparing the data significantly improves the model's accuracy. This work highlights the potential of artificial intelligence to support efficient water resource planning and management in urban environments.

Keywords: demand prediction, neural networks, LSTM, drinking water, time series

CAPÍTULO I: INTRODUCCIÓN

1. Introducción

1.1. Contexto

El agua potable es un recurso fundamental para la vida, la salud y el desarrollo económico. Una gestión eficiente de este recurso esencial se ha convertido en una prioridad global, especialmente en un nuevo contexto marcado por diversos factores como el cambio climático y el crecimiento demográfico y urbanístico. Asegurar el suministro y la calidad de agua potable en la cantidad requerida y de forma continua representa un desafío técnico y logístico para las entidades encargadas del ciclo y tratamiento del agua. En este contexto, las Estaciones de Tratamiento de Agua Potable (ETAPs) desempeñan el papel fundamental en la transformación del agua bruta en agua apta para el consumo humano, garantizando siempre el cumplimiento de todos los estándares de calidad y sanitarios establecidos por la normativa vigente.

En España, el consumo medio de agua potable en hogares se encuentra en torno a 128 litros por habitante y día, según datos del INE (Instituto Nacional de Estadística) del año 2022. Sin embargo, en la comunidad autónoma de Castilla y León, el consumo asciende alcanzando los 146 litros por habitante y día. Con estos datos queda reflejado la necesidad de implementar y desarrollar estrategias de gestión más eficientes y sostenibles. [1]

En la ciudad de Valladolid, la entidad pública Aquavall es la encargada de gestionar el sistema de abastecimiento de agua, la cual suministra agua a más de 344.600 hogares e industrias, este suministro abarca a la ciudad de Valladolid y varias poblaciones del Alfoz. El consumo medio de agua potable al día en la ciudad de Valladolid, según datos de Aquavall, es de 76 millones de litros, lo que equivale a un consumo medio cercano a 220 litros por habitante y día, una cifra que supera tanto la media nacional como la autonómica. Esta elevada demanda subraya la importancia de optimizar los procesos de producción y distribución de agua potable en la ciudad. [2]

En Valladolid existen dos plantas principales para el tratamiento del agua: la ETAP de “Las Eras” y la ETAP de “San Isidro”. La ETAP de “Las Eras” fue construida en el año 1955, capta el agua del Canal de Castilla y se encarga de producir el 70% del agua consumida en Valladolid. Por su parte, la ETAP de “San Isidro” es la más antigua y data del año 1866, capta el agua del Canal del Duero y produce el 30% restante de la producción de agua potable. Ambas plantas disponen de una capacidad conjunta de producción cercana a los 30 millones de metros cúbicos de agua potable al año. [2]

Figura 1.1: ETAP de "Las Eras"

Ser capaces de predecir de forma precisa el futuro consumo de agua permite optimizar el uso de los recursos hídricos, crear una planificación de la producción del agua tratada, reducir costes operativos y disminuir el impacto ambiental ligado al tratamiento y distribución del agua. Sin embargo, la predicción de la demanda de agua es una tarea compleja, esta predicción implica el análisis de datos históricos, la identificación de patrones en el consumo y ser capaces de anticipar el comportamiento en función de múltiples variables.

Tradicionalmente, esta tarea de predicción se ha abordado mediante modelos estadísticos lineales, como la regresión lineal, también mediante el uso de modelos autorregresivos como los modelos ARIMA [3] o técnicas de suavizado exponencial [4]. El uso de modelos determinísticos, como aquellos basados en el número de población y emplear tasas de consumo histórico por habitante, y el empleo de algunos softwares de simulación hidráulica como EPANET. Otros métodos empleados son los empíricos con el uso de curvas de demanda históricas basadas en medias y patrones anuales [5]. Estas herramientas tradicionales mencionadas son útiles en contextos simples, sin embargo, presentan limitaciones en patrones no lineales y dinámicos, por ejemplo ante cambios en factores como temperatura, humedad o el día de la semana, además de la necesidad de ajustes manuales de manera frecuente. En respuesta a estas limitaciones, surge la tendencia e interés en los últimos años por la aplicación de técnicas de inteligencia artificial para el análisis de series temporales en el ámbito de la predicción y gestión del agua.

Dentro de las distintas aproximaciones basadas en el aprendizaje automático, las redes neuronales recurrentes (RNN) han demostrado ser eficaces en la predicción

de las secuencias temporales. En concreto, las redes de tipo Long Short-Term Memory (LSTM) son capaces de modelar relaciones temporales más complejas, ya que se encuentran diseñadas para recordar información relevante a lo largo del tiempo y aplicar filtros a datos menos significativos. Estas características convierten a las redes LSTM en una herramienta para la predicción de la demanda de agua con precisión, teniendo en cuenta datos históricos y variables externas que puedan influir en el comportamiento de la demanda.

1.2. Objetivos del trabajo

El objetivo principal de este Trabajo Fin de Grado es el desarrollo de un modelo de predicción de la demanda de agua diaria en la ciudad de Valladolid empleando redes neuronales de tipo LSTM (Long Short-Term Memory), una técnica de aprendizaje automático diseñada especialmente para el análisis de series temporales. Dicho modelo tiene como principal propósito proporcionar una anticipación de la cantidad de agua requerida y demandada en las Estaciones de Tratamiento de Agua Potable (ETAPs) de Valladolid mediante el uso de datos históricos de consumo, proporcionados por Aquavall, y otros factores externos considerados relevantes. Este modelo busca ofrecer una herramienta más precisa y versátil frente a las técnicas tradicionales, integrando conocimientos de ingeniería, tratamiento de datos y aprendizaje automático.

A mayores del valor académico, este Trabajo Fin de Grado pretende dar respuesta a la necesidad real y actual de la gestión hídrica, como disponer de predicciones fiables que permitan optimizar la planificación de los recursos, reducir el desperdicio de agua, ser capaz de anticiparse y dar respuesta a picos de consumo y mejorar la eficiencia energética del proceso de obtención de agua potable. En una sociedad cada vez más condicionada por una variabilidad climática y aumento demográfico, la capacidad de previsión de la demanda de agua se convierte en un factor crítico tanto en la toma de decisiones como en la reducción de costes. Las redes LSTM, a diferencia de los modelos más clásicos, permiten modelar fenómenos no lineales o con dependencia de otros factores, lo que las convierte en candidatas apropiadas para el desarrollo de este modelo.

De manera más concreta, este trabajo plantea los siguientes objetivos específicos:

- **Analizar** los datos históricos de consumo de agua en Valladolid, así como otro tipo de datos, como los meteorológicos, que puedan estar relacionados con la demanda de agua.
- **Preprocesar y transformar** los datos para adecuarlos al formato requerido por los modelos de redes neuronales, lo cual incluye la normalización de

variables, el tratamiento de valores ausentes o no significativos y un tratamiento final de los datos.

- **Diseñar e implementar** un modelo de red LSTM en Python, utilizando librerías como TensorFlow o Keras, implementando diferentes variables para formar una arquitectura en función de los objetivos del trabajo.
- **Entrenar y validar** el modelo con una parte de los datos disponibles, ajustando parámetros y utilizando métricas de evaluación como la raíz del error cuadrático medio (RMSE) para medir su rendimiento.
- **Comparar** los resultados del modelo LSTM con el resto de los datos proporcionados, para observar como de precisa es la predicción.
- **Reflexionar** sobre los resultados obtenidos e implementar mejoras para incrementar la exactitud de las predicciones y así obtener mejores modelos para la posterior integración en sistemas reales de gestión del agua.

En conclusión, el objetivo es sentar las bases para optimizar la gestión del agua mediante inteligencia artificial, asumiendo sus limitaciones actuales pero demostrando su capacidad para revolucionar los sistemas predictivos en el ámbito urbano.

1.3. Organización de la memoria

Este documento se estructura en cinco capítulos que recogen de forma ordenada el desarrollo completo del trabajo.

En el primer capítulo se presenta una introducción general al proyecto, incluyendo el contexto en el que se enmarca, los objetivos perseguidos y una explicación de la organización del propio documento.

El segundo capítulo aborda el marco teórico necesario para entender las técnicas empleadas. Se explican los procesos de tratamiento de agua en una ETAP, los fundamentos de la predicción de series temporales, y se introducen los conceptos clave sobre redes neuronales, haciendo especial énfasis en las redes LSTM.

El tercer capítulo describe las fuentes de datos utilizadas y el proceso de preprocesamiento llevado a cabo, desde la limpieza y transformación de la serie temporal hasta el escalado y preparación de los datos para su uso en el modelo.

En el cuarto capítulo se desarrolla el diseño del modelo LSTM y se explican las distintas simulaciones realizadas. Se presentan y comparan los resultados obtenidos

en función de distintas configuraciones, incluyendo enfoques univariados y multivariados, así como predicciones a uno o varios días vista.

El quinto capítulo recoge las conclusiones generales del trabajo, destacando los resultados más relevantes, las limitaciones encontradas y posibles líneas futuras de mejora o ampliación.

Por último, se muestra la bibliografía que se ha consultado para la realización de este trabajo.

Cabe añadir que, para facilitar la redacción del documento y mejorar su claridad, se ha hecho uso puntual de herramientas de inteligencia artificial como apoyo.

CAPÍTULO II: MARCO TEÓRICO

2. Marco teórico

2.1. Tratamiento del agua en una ETAP

El tratamiento de agua en una Estación de Tratamiento de Agua Potable (ETAP) tiene como objetivo fundamental transformar el agua bruta, procedente de fuentes naturales como ríos, embalses o canales, en agua potable apta para el consumo humano. Para lograrlo, se recurre a un proceso secuencial compuesto por distintas etapas que combinan métodos físicos, químicos y, en algunos casos, biológicos. Cada una de estas etapas está diseñada para eliminar diferentes tipos de impurezas (desde sólidos en suspensión hasta microorganismos patógenos), con el fin de cumplir con los estándares de calidad exigidos. [2]

Este proceso no solo garantiza que el agua sea segura desde el punto de vista sanitario, sino que también tenga unas condiciones adecuadas de calidad para el usuario, como un sabor neutro, un olor agradable y un aspecto claro y limpio. En este apartado se describen en detalle los principales procesos de tratamiento en una ETAP, así como los factores que pueden influir en la demanda de agua en una ciudad como Valladolid.

2.1.1. Procesos en una ETAP

Los procesos que forman el ciclo urbano del agua en la ciudad de Valladolid son (Figura 2.1): captación, potabilización, distribución, abastecimiento, saneamiento, depuración y retorno.

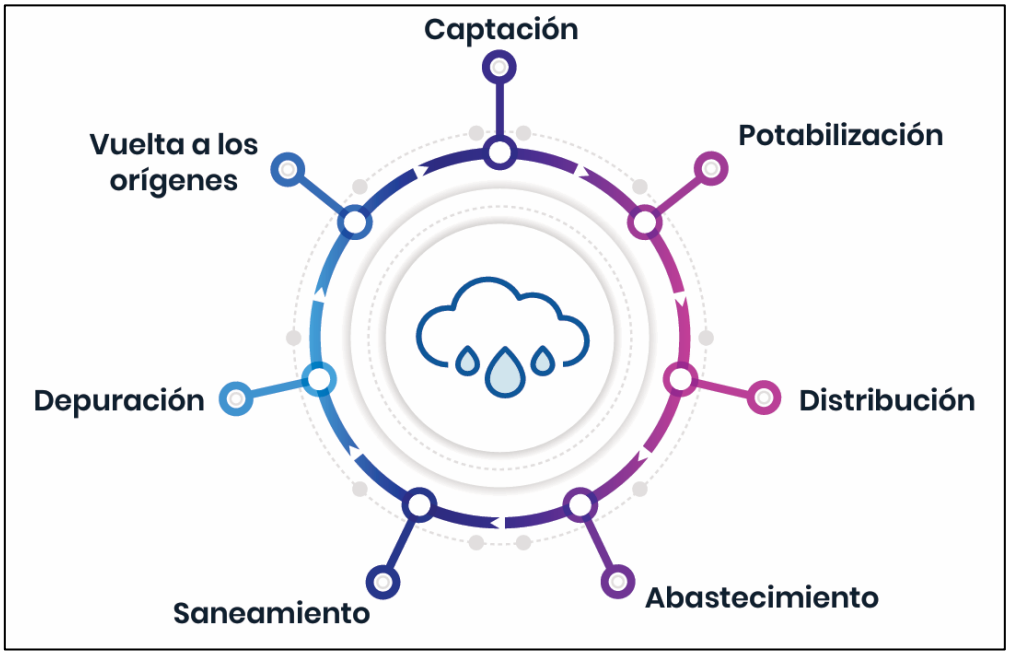

Figura 2.1: Ciclo urbano del agua en Valladolid

De los procesos mencionados aquellos que se llevan a cabo en las Estaciones de Tratamiento de Agua son la captación de agua, la potabilización de esta y el punto de partida para el abastecimiento.

Captación del agua

El proceso comienza con la captación del agua desde fuentes naturales (ríos, embalses o canales). En el caso de Valladolid, la ETAP de "Las Eras" se abastece normalmente desde el Canal de Castilla por gravedad aunque cuenta con una toma impulsada de agua desde el río Pisuerga para casos de emergencia, el abastecimiento de agua para la ETAP de "San Isidro" se toma habitualmente del Canal del Duero, aunque también dispone de una toma de emergencia impulsada desde el río Duero. Estas tomas de emergencia mencionadas apenas han sido utilizadas, ya que las condiciones de captación habituales han resultado suficientes. [2]

Se puede apreciar la distribución de volúmenes captados en diferentes años en Valladolid en la Tabla 1:

VOLUMEN CAPTADO (m^3)						
	2018	2019	2020	2021	2022	2023
Volumen Canal de Castilla	21.130.061	21.779.998	21.123.025	22.089.364	14.673.253	20.829.944
Volumen Canal del Duero	9.757.491	9.099.075	8.946.506	7.738.232	14.720.859	8.638.429
Volumen Puente Mayor - Pisuerga	0	0	0	0	0	0
Volumen Boecillo - Duero	0	0	0	0	0	0
TOTAL VOLUMEN CAPTADO	**30.887.552**	**30.879.073**	**30.069.531**	**29.827.596**	**29.394.112**	**29.468.373**

Tabla 1: Volúmenes captados en diferentes años en Valladolid

En esta tabla se observa como el volumen total captado en Valladolid siempre se encuentra sobre los 30 millones de metros cúbicos, con una disminución en los últimos años. También se confirma que, al menos en estos años que disponemos de

datos, no ha sido necesario recurrir a la captación de agua de las tomas de emergencia del rio Pisuerga y Duero.

Potabilización del agua

Una vez se ha captado el agua de las fuentes mencionadas, debe ser potabilizada para que sea apta para el consumo humano. Para realizar esta función en Valladolid se dispone de las dos plantas de tratamiento físico-químico mencionadas anteriormente como son la ETAP de "Las Eras" y la ETAP de "San Isidro". Juntas son capaces de abastecer a la ciudad con una producción cerca de 30 millones de metros cúbicos de agua potable anualmente.

La ETAP de "Las Eras", en funcionamiento desde 1955, lleva prestando servicio a la ciudad de Valladolid durante 70 años. Dispone de una capacidad de producción de 4.500 metros cúbicos por hora, además de una capacidad de depósito de agua de 1.500 metros cúbicos. A su vez esta Estación de Tratamiento cuenta con un laboratorio acreditado para el análisis del agua tratada, lo que garantiza un control riguroso y continuo de la calidad del agua suministrada.

Construida en 1886, la ETAP de "San Isidro" representa una de las infraestructuras más antiguas y emblemáticas del sistema de abastecimiento de Valladolid. Con casi 140 años de historia, ha experimentado múltiples mejoras para adaptarse a las exigencias actuales en materia de calidad y suministro. Esta planta cuenta con una capacidad de producción de 4.500 metros cúbicos por hora y dispone de depósitos de agua con una capacidad de 74.000 metros cúbicos, lo que permite asegurar una gran reserva para el abastecimiento de la ciudad. [2]

Se observa las diferencias de volúmenes captados y suministrados en diferentes años en Valladolid en la Tabla 2.

El proceso de potabilización del agua en las Estaciones de Tratamiento de Agua Potable de Valladolid consta de diversas etapas:

1. Se recibe el agua mediante una serie de tuberías, en esta toma se aplica el tratamiento de desbaste de gruesos, una operación que consiste en la eliminación de elementos de gran tamaño como ramas, hojas, plásticos u otros residuos sólidos. Esta tarea se suele realizar mediante rejas o tamices metálicos, que actúan como filtros evitando que dichos materiales entren en el sistema de tratamiento, protegiendo así el correcto funcionamiento de los equipos.

2. A continuación, el agua se dirige a las torretas de mezcla o de llegada, donde comienza el primer tratamiento químico. En esta etapa de pre-desinfección se añaden agentes oxidantes como el cloro y el oxígeno con el objetivo de reducir los microorganismos patógenos (bacterias, virus, protozoos)

presentes en el agua. También se añaden en esta etapa diferentes tipos de elementos coagulantes y floculantes, como el cloruro férrico y el sulfato de alúmina, que agrupan las partículas en suspensión mediante la desestabilización de estas, generando así flóculos de mayor tamaño lo que facilita su eliminación con un proceso de decantación. [2]

3. Desde las torretas de mezcla el agua pasa a uno de los diferentes decantadores. En la ETAP de "Las Eras" se dispone de 6 decantadores, 3 de tipo aceleradores estáticos, 2 con puente rascador y uno lamelar, en cambio en la ETAP de "San Isidro" disponemos de 4 decantadores, 2 son de tipo acelerador dinámico y los 2 restantes son Superpulsator. Gracias a la ayuda de los coagulantes y floculantes añadidos anteriormente la materia coloidal o en suspensión se puede eliminar del agua fácilmente en los decantadores mencionados [2].

4. Una vez el agua se ha clarificado mediante el anterior paso esta pasa a uno de los 14 filtros de arena y posteriormente a uno de los 6 filtros de carbón activo granular. Los filtros de arena se encargan de eliminar partículas suspendidas mediante un proceso de tamizado, donde las partículas más grandes quedan atrapadas. La función del carbón activo granular en los filtros es eliminar contaminantes químicos y mejorar las características organolépticas del agua [6]. Mediante el proceso de adsorción los contaminantes se adhieren a la superficie del carbón, eliminado así contaminantes orgánicos, compuestos químicos (cloro) y también el posible mal olor y sabor del agua [7]. Existe la posibilidad de reforzar la oxidación en este paso con el empleo de ozono, un gas altamente oxidante con propiedades desinfectantes.

5. En el último paso de potabilización, se añade hipoclorito sódico como agente desinfectante. Esta sustancia, al disolverse en el agua, actúa liberando cloro activo, cuya función es inactivar los microorganismos patógenos que pudieran persistir tras las etapas anteriores del proceso de potabilización. [8]

Aquavall dispone de un laboratorio donde lleva a cabo las tareas de vigilancia y control oportunas marcadas y establecidas por los criterios sanitarios de la calidad del agua de consumo humano, garantizando así las condiciones óptimas. Según datos facilitados por Aquavall más de 2.500 analíticas para el control del agua potable son realizadas cada año.

Una vez completado todo el proceso de potabilización, se almacena el agua en los diferentes depósitos (con capacidad para proporcionar casi 3 días de autonomía) para su posterior transporte a la ciudad mediante un bombeo directo a la red. [2]

	2018		2019		2020	
	Volumen suministrado (m^3)	Volumen captado (m^3)	Volumen suministrado (m^3)	Volumen captado (m^3)	Volumen suministrado (m^3)	Volumen captado (m^3)
Volumen Canal de Castilla (Eras)	20.701.573	21.130.061	19.573.267	21.779.998	19.638.989	21.123.025
Volumen Canal del Duero (San Isidro)	7.719.623	9.757.491	8.094.616	9.099.075	8.145.542	8.946.506
Bypass Eras -> San Isidro	-247.803	-	-454.866	-	-263.906	-
TOTAL VOLUMEN	**28.173.393**	**30.887.552**	**27.213.017**	**30.879.073**	**27.520.625**	**30.069.531**
Rendimiento de producción (%)	**91,21 %**		**88,13 %**		**91,52 %**	

Tabla 2: Comparativa de volúmenes suministrados y captados en diferentes años en Valladolid

2.1.2. Factores que influyen en la demanda de agua

La demanda de agua potable en entornos urbanos se encuentra directamente relacionada con la interacción de múltiples factores, tanto naturales como sociales, que deben ser tomados en cuenta para una correcta estimación y predicción. En el caso de la ciudad de Valladolid, una ciudad de tamaño considerable, el abastecimiento debe ser adaptable a las posibles variaciones diarias y/o estacionales del consumo de agua potable.

Uno de los principales factores determinantes es el factor climático. La temperatura ambiente, la humedad relativa y las precipitaciones influyen directamente sobre la cantidad de agua consumida. En general, las temperaturas elevadas y la escasez en las precipitaciones se traducen en un incremento significativo de la demanda de agua, principalmente en los períodos de verano. El aumento del consumo, en estas condiciones climatológicas mencionadas, se puede explicar debido a la mayor frecuencia de duchas, lavado y uso del agua en elementos externos al hogar, apreciando así una tendencia más acusada en aquellas viviendas o zonas residenciales que disponen de zonas ajardinadas, piscinas o grandes terrazas. [9]

La Agencia Estatal de Meteorología (AEMET) ha señalado que las previsiones climáticas para el siglo XXI apuntan a una mayor irregularidad en las precipitaciones

y un aumento de las temperaturas, lo que podría traducirse en una mayor presión sobre los recursos hídricos urbanos. [10]

Otro aspecto relevante es la población residente y visitante. El número de habitantes conectados a la red influye directamente en el volumen de agua demandado. Por ello, la presencia de visitantes o el turismo puede provocar picos puntuales en determinadas épocas del año. Según datos del INE (2023), Valladolid recibe más de 400.000 turistas anuales, lo cual impacta estacionalmente en el consumo, sobre todo eventos culturales de la ciudad [11]. Se aprecia un incremento de la ocupación hotelera de la ciudad en los meses desde marzo (eventos relacionados con la Semana Santa) hasta finales de verano en septiembre (Fiestas populares de Valladolid) lo que se traduce en un aumento de la demanda de agua potable a tener en cuenta para la predicción. [12]

Los hábitos de consumo, que están ligados a los factores culturales y sociodemográficos, juegan también un papel muy importante. Factores como el nivel de renta, la concienciación ambiental o un acceso a tecnologías de ahorro de agua pueden modificar los comportamientos de los usuarios. También el precio del agua influye significativamente, incrementos en la tarifa provocan descensos en el consumo, especialmente en hogares con menor renta.

Junto con los factores ya mencionados, la estructura urbana y el tipo de edificación influye en el patrón de consumo. Las zonas con viviendas unifamiliares tienden a tener un consumo significativamente superior, debido al uso de agua para el riego de jardines y llenado de piscinas privadas, en comparación con viviendas en bloques de pisos donde el consumo se centra en los usos domésticos básicos.

Otro componente es el estado de la infraestructura de distribución. Las pérdidas en la red, causadas por fugas o averías, pueden suponer una importante cantidad de agua desperdiciada. Según el informe del INE (Instituto Nacional de Estadística) de 2020, se cifran las pérdidas de agua en el abastecimiento urbano debido a fugas, roturas y averías en la red pública en un 15,4% del caudal, equivalente a 652 hm^3. A este dato también se podrían añadir las pérdidas derivadas de errores o consumos ilegales, que suponen 413 hm^3.

En el caso de Valladolid, la empresa municipal Aquavall gestiona directamente la red de abastecimiento, y ha impulsado mejoras en la digitalización y control de fugas para reducir estas pérdidas. Según el Plan Estratégico 2022–2025 de Aquavall, las pérdidas técnicas se sitúan por debajo del 10 %, un dato favorable en comparación con la media nacional. Esto implica que las predicciones deben considerar no solo el consumo directo de los usuarios, sino también una estimación de pérdidas estructurales para dimensionar adecuadamente la producción en las ETAPs.

También es importante tener en cuenta las campañas institucionales o restricciones puntuales, como en situaciones de sequía que tienen efectos en una reducción evidente del consumo y demanda [9]. En períodos de escasez hídrica, los ayuntamientos pueden emitir ordenanzas que limiten usos no prioritarios, lo que se

traduce en caídas puntuales en la demanda. Por ejemplo, en julio de 2022, ante el descenso del nivel de los embalses, varias localidades de Castilla y León, incluida Valladolid, promovieron campañas de ahorro de agua e implementaron medidas restrictivas en el riego de zonas verdes.

En los últimos años, también se han incorporado factores ligados al desarrollo de ciudades inteligentes (smart cities), donde se utilizan sensores, plataformas digitales y sistemas de análisis de datos para monitorizar en tiempo real el comportamiento de la red de agua. Este nuevo uso de tecnologías permite una gestión más precisa de la demanda y facilita la predicción mediante técnicas de inteligencia artificial. La ciudad de Valladolid forma parte de la Red Española de Ciudades Inteligentes (RECI), y ha sido beneficiaria de fondos europeos para proyectos de digitalización del ciclo urbano del agua. [13]

Por último, es fundamental destacar el impacto de fenómenos excepcionales, como las pandemias. Durante la crisis del COVID-19, el confinamiento general y los cambios en la rutina provocaron un aumento en el consumo residencial, al tiempo que disminuyó el consumo comercial e industrial, lo que nos indica que también se debe estar preparado para estas circunstancias excepcionales que generan cambios en el consumo.

Como conclusión, la demanda de agua potable urbana está condicionada por una combinación de factores climáticos, demográficos, sociales, técnicos y regulatorios. La comprensión de estos elementos es clave para desarrollar modelos predictivos eficaces que permitan anticipar las necesidades de la red de abastecimiento. En el caso de Valladolid, la estacionalidad, los hábitos de consumo, el estado de la infraestructura y la digitalización del sistema son variables que deben considerarse en el desarrollo de sistemas de predicción basados en redes neuronales, como las que se abordan en este trabajo.

2.2. Predicción con series temporales

El análisis y estudio de las series temporales es una técnica imprescindible en el ámbito de la inteligencia artificial y estadística, empleada para anticipar el comportamiento futuro de variables que evolucionan en el tiempo, como la demanda de agua en contextos urbanos. Una serie temporal se define como un conjunto de datos u observaciones que se encuentran en un orden cronológico, tomadas en intervalos periódicos o regulares, lo que permiten reflejar la evolución de una variable a lo largo del tiempo [14]. Las series temporales pueden ser discretas (diarias, mensuales, anuales, etc.) o continuas, pero su estructura temporal debe implicar que los datos estén relacionados entre sí. [15]

El uso de series temporales permite la posibilidad de identificar patrones históricos dentro de la serie que genere la predicción de futuros valores. Esto adquiere una gran relevancia en la gestión del agua potable, donde disponer de una predicción precisa permite una mejor planificación y garantizar el suministro sostenible.

Las series temporales están formadas de varios componentes que pueden superponerse:

- Tendencia: representa la evolución o los cambios a largo plazo de la serie. La tendencia puede ser ascendente o descendente, mostrando por ejemplo un incremento o decremento en la demanda de agua, o mantenerse estable. Puede estar ligada y reflejar factores estructurales como un crecimiento poblacional u otros factores influyentes en el consumo de agua.

- Estacionalidad: se entiende como los patrones que se repiten en intervalos regulares, un ejemplo serían los picos de consumo de agua en verano.

- Ciclos: son fluctuaciones más amplias que la estacionalidad, pero no necesariamente regulares. Los cambios suelen estar relacionados con factores económicos o sociales de largo plazo.

- Ruido: engloba irregularidades o variaciones aleatorias sin un patrón especifico, por ello no se puede atribuir concretamente a ningún factor de los mencionados.

Entender estos componentes permite aplicar técnicas adecuadas para su modelado y predicción. Box y Jenkins (1976) fueron pioneros en este campo al proponer los modelos ARIMA (AutoRegressive Integrated Moving Average), que han sido ampliamente utilizados como base para las predicciones univariantes. [16]

La presencia de los componentes mencionados influye directamente en la elección del modelo predictivo. Por ejemplo, los modelos clásicos ARIMA requieren que la serie sea estacionaria, es decir, que sus propiedades estadísticas no cambien en el tiempo [15]. Para ello, se suelen aplicar transformaciones como la diferenciación, que elimina tendencias, o la transformación Box-Cox, que estabiliza la varianza.

La predicción de series temporales es fundamental en muchos sectores, incluyendo la energía, las finanzas y la gestión de recursos como el agua. En el caso concreto del abastecimiento de agua potable, permite anticipar la demanda y planificar con antelación la producción, el almacenamiento y la distribución del agua [17]. Una predicción precisa contribuye a:

- Reducir pérdidas económicas y energéticas por sobreproducción o subabastecimiento.

- Mejorar la gestión de infraestructuras hídricas.

- Reaccionar ante eventos inesperados como olas de calor o restricciones por sequía.

Existen dos tipos principales de predicción: a corto plazo, empleada para la operación diaria y semanal del sistema, y a largo plazo, que apoya decisiones estratégicas como inversiones en infraestructuras o gestión de recursos hídricos. [18]

Se han desarrollado distintos enfoques estadísticos para la predicción de series temporales. Entre los más representativos destacan los siguientes:

- Medias móviles y suavizado exponencial

Los modelos de media móvil (moving average) calculan el futuro valor de una serie a partir del promedio de un número determinado de observaciones pasadas. Este método es útil para series con poco ruido y sin una marcada estacionalidad. Sin embargo, su capacidad predictiva es limitada, ya que no considera tendencias ni relaciones complejas entre los datos.

El suavizado exponencial (exponential smoothing) es un método mejorado que aplica pesos decrecientes a las observaciones pasadas. El método de Holt-Winters, por ejemplo, permite modelar series con tendencia y estacionalidad [19]. Estas técnicas son simples y fáciles de implementar, pero no son efectivas cuando los datos presentan un comportamiento no lineal o múltiples variables externas.

- Modelo AR

El modelo AR (Autorregrresivo) se basa principalmente en el uso de observaciones pasadas para la predicción de observaciones a futuro, empleando un proceso donde un valor en un momento determinado puede calcularse en función de los valores en momentos pasados de la misma serie. [15]

Estos modelos trabajan correctamente bajo el uso de una serie estacionaria y aquí se encuentra su principal limitación.

- Modelos ARIMA y SARIMA

El modelo ARIMA (AutoRegressive Integrated Moving Average) propuesto en 1976 por Box y Jenkins ha sido durante años un modelo base en la predicción de series temporales. En este modelo se combinan componentes de autorregresión, de integración y diferenciación, para convertir la serie en estacionaria, además de medias móviles. [16]

La versión del modelo SARIMA incluye componentes estacionales para la mejor captura de patrones periódicos.

La mayor limitación de estos modelos se encuentra en la necesidad de trabajar con una serie estacionaria, teniendo así dificultades en series no lineales. [14]

- Modelos VAR

El modelo VAR (Vector Autorregresivo) está basado en la creación de un modelo AR (Autorregresivo) para una serie temporal de diversas variables, donde cada una de las variables genera una dependencia e influencia en el resto de ellas.

Al igual que en los modelos autorregresivos la mayor limitación se encuentra en las dificultades cuando no se trata con una serie estacionaria. [15]

Los modelos tradicionales son adecuados para los problemas simples, donde hay patrones estables y pocas variables externas. Sin embargo, presentan dificultad para adaptarse a cambios abruptos o comportamientos no lineales. Además, su ajuste requiere un conocimiento profundo de la estructura interna de la serie, como el número de retardos óptimos, que se suelen determinar mediante criterios de información como AIC, BIC o HQC. [15]

Observando las limitaciones que tienen estos modelos clásicos, se hacen poco compatibles con el objetivo de este trabajo ya que la predicción de demanda de agua requiere de múltiples variables externas y los datos obtenidos son normalmente no lineales.

En las últimas décadas, ha cobrado relevancia el uso de modelos de aprendizaje automático, especialmente redes neuronales, para la predicción de series temporales. Estas técnicas no requieren que la serie sea estacionaria y permiten modelar relaciones no lineales y de múltiples variables.

Algunos estudios consideran que los modelos estadísticos siguen siendo superiores en ciertos contextos, otros trabajos demuestran que las redes neuronales ofrecen mejores resultados en tareas complejas y con grandes volúmenes de datos. Por ello, se ha generalizado el uso combinado de ambos enfoques mediante modelos híbridos como el ES-RNN (Exponential Smoothing – Recurrent Neural Network), que combina el suavizado exponencial con las redes neuronales. [15]

Las redes neuronales recurrentes (RNN), y especialmente las redes LSTM (Long Short-Term Memory), han demostrado una excelente capacidad para predecir series temporales con alta dependencia temporal y ruido estructural, como sucede en sistemas complejos como el tráfico urbano o el consumo energético. En particular, redes LSTM fueron empleadas por Zhao et al. (2017) para el pronóstico del tráfico a corto plazo, logrando precisión superior frente a modelos tradicionales gracias a su capacidad para capturar dinámicas temporales prolongadas. [20]

La posibilidad de integrar múltiples variables, como la temperatura, la humedad o los días de la semana, convierte a estos modelos en herramientas muy adecuadas para aplicaciones reales como la gestión del agua potable. Su implementación, aunque es más compleja que los modelos clásicos, ofrece mayor precisión y capacidad de adaptación ante cambios o patrones no lineales.

2.3. Redes neuronales para predicción

El desarrollo de modelos de predicción basados en inteligencia artificial (IA) ha experimentado un crecimiento notable en los últimos años, especialmente en ámbitos donde los métodos estadísticos tradicionales presentan limitaciones frente a relaciones no lineales, dependencias temporales complejas y conjuntos de datos multivariados. Una de las técnicas más consolidadas dentro de la IA son las redes neuronales artificiales (RNA), cuyo diseño se inspira en el funcionamiento de las conexiones sinápticas del cerebro humano. Estas redes son capaces de aprender patrones subyacentes en grandes volúmenes de datos históricos y generalizar dicho aprendizaje para realizar predicciones en nuevos contextos.

En el ámbito de las series temporales, las RNA han demostrado su eficacia frente a los enfoques clásicos como los modelos ARIMA, especialmente cuando los datos presentan estacionalidad, ruido y múltiples factores externos. Se ha destacado que estas redes pueden adaptarse a contextos dinámicos y mejorar la precisión predictiva cuando se combinan con estrategias de preprocesamiento y evaluación continua del modelo. Su aplicación a la predicción de la demanda de agua, un hecho condicionado por múltiples variables como la meteorología, el calendario o los hábitos de consumo, ha demostrado resultados prometedores. Estas redes permiten anticipar comportamientos de alta variabilidad y optimizar así la gestión de infraestructuras hidráulicas urbanas con una precisión superior a la de modelos lineales convencionales. [21]

2.3.1. Introducción a las redes neuronales artificiales

Las redes neuronales artificiales son modelos computacionales diseñados para reconocer patrones y aprender relaciones complejas que existan entre las entradas y salidas, todo esto se lleva a cabo mediante un proceso de entrenamiento. Estas estructuras de redes se componen de diversas unidades de procesamiento llamadas neuronas artificiales, las cuales se encuentran interconectadas en capas. El modelo está inspirado, como su propio nombre indica, en las redes de neuronas biológicas del sistema nervioso, aunque en este caso con un funcionamiento únicamente matemático. [22, capítulo 1]

Cada neurona que forma parte de la red neuronal se considera una unidad algoritmica, que recibe un vector de entrada ($\vec{x}$) y genera como salida otro valor o vector ($\vec{y}$). Para conseguir dicha salida se aplica una operación de transformacion empleando la ecuación:

$$\boldsymbol{z = \vec{w} \cdot \vec{x} + b} \qquad \textbf{(1)}$$

Las entradas x_n se multiplan por sus respectivos pesos (w) y se suma un término de sesgo $(bias, b)$. El cálculo de estas variables de peso y bias se calculan por medio del proceso de entrenamiento.

Una vez el valor ha sido transformado se introduce z en una función de activación no lineal que permite capturar relaciones más complejas (Figura 2.2).

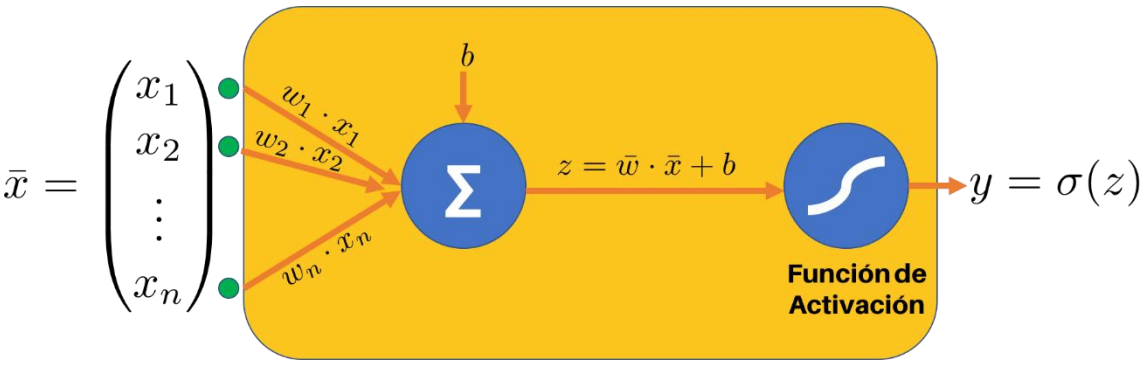

Figura 2.2: Elementos que forman un neurona artificial

Una red neuronal básica está compuesta por tres tipos de capas (Figura 2.3):

- **Capa de entrada (input layer):** recibe los datos de entrada.
- **Capas ocultas (hidden layers):** realizan el procesamiento intermedio mediante combinaciones lineales y funciones no lineales.
- **Capa de salida (output layer):** entrega la predicción final o la clasificación, dependiendo del tipo de problema.

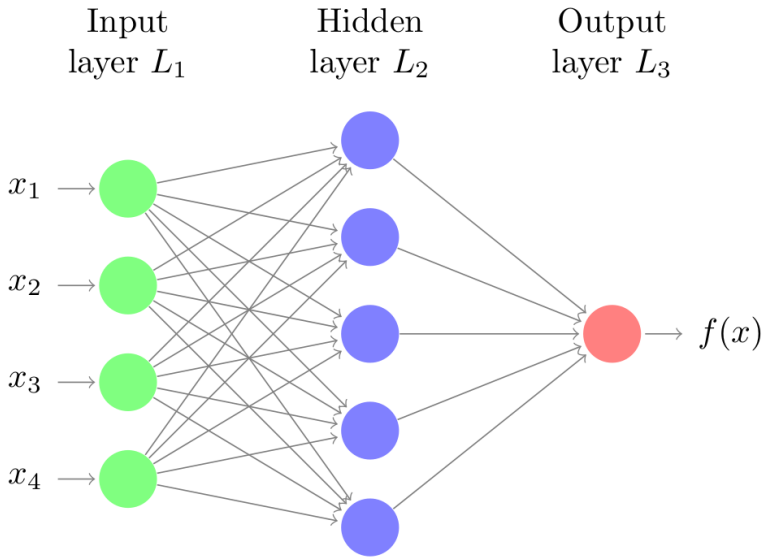

Figura 2.3: Red neuronal simple

Una red neuronal simple se compone de una capa de entrada, una capa oculta y una capa de salida. Cuando se añaden múltiples capas ocultas, se habla de una red neuronal profunda (deep neural network, DNN) (Figura 2.4). Estas estructuras profundas permiten extraer representaciones jerárquicas y detectar relaciones no lineales entre las variables, lo que resulta esencial en problemas complejos. [22, capítulo 6]

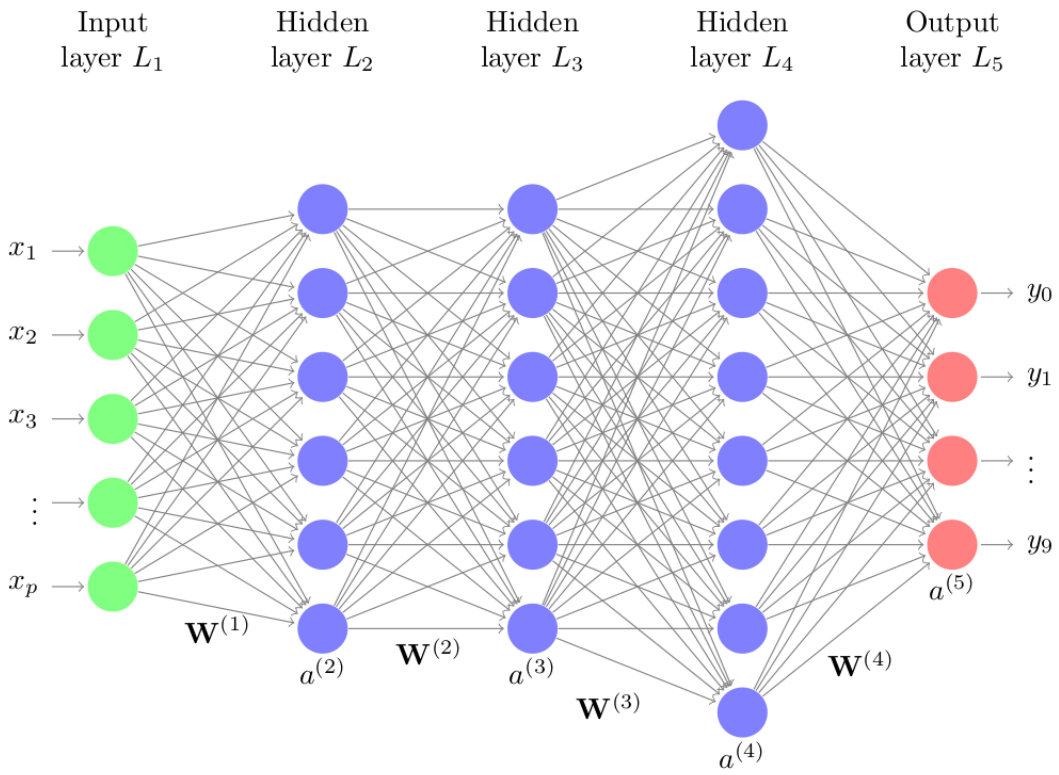

Figura 2.4: Red neuronal profunda

Cada una de las conexiones entre neuronas se encuentra asociada a un peso (w), cuya función es regular la información que se propaga de una neurona a otra

Cada neurona aplica una función de activación empleada para introducir la no linealidad en el modelo, como su propio nombre indica esta función se encarga de determinar si esa neurona debe o no activarse, es decir, si debe transmitir información a las neuronas siguientes que se encuentran ligadas a ella. [23]

Sin el uso de estas funciones de activación una red neuronal solo sería capaz de realizar transformaciones lineales de los datos de entrada. Por ello es necesario incluir la no linealidad mediante estas funciones para así poder aprender y representar las relaciones más complejas que encontremos entre la entrada y salida del proceso.

Dentro de las funciones de activación se encuentran diferentes tipos, las más destacadas y empleadas son [23, 24]:

- Función sigmoidal (σ):
  Devuelve valores entre 0 y 1. Es útil para representar probabilidades, especialmente en la capa de salida de problemas de clasificación binaria. Su principal limitación es que tiende a provocar el problema del desvanecimiento del gradiente, ya que sus derivadas son pequeñas para valores extremos.

$$\sigma(\mathrm{x}) = \frac{1}{1+e^{-x}} \tag{2}$$

- Función tangente hiperbólica (tanh):
  Similar a la sigmoidal, pero su rango va de -1 a 1, lo que la hace más centrada en cero y más útil en capas ocultas. También puede verse afectada por el desvanecimiento del gradiente.

$$\tanh(\mathrm{x}) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3}$$

- Función *ReLU (Rectified Linear Unit):*
  Se ha convertido en la más popular en redes profundas por su simplicidad y eficiencia computacional. Devuelve cero para valores negativos y el propio valor para positivos. Aunque evita el desvanecimiento del gradiente, puede sufrir el problema de "neurona muerta" si recibe muchas entradas negativas.

$$f(x) = \max(0, x) \tag{4}$$

- Función Softmax:
  Se emplea típicamente en la capa de salida de redes de clasificación multiclase. Convierte un vector de valores en un vector de probabilidades normalizadas que suman 1.

$$\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{5}$$

- Función Swish:
  Propuesta por investigadores de Google, combina propiedades de la ReLU y la sigmoide. Ha mostrado mejor rendimiento en algunas tareas de aprendizaje profundo, especialmente en redes profundas y convolucionales.

$$\mathrm{Swish}(\mathrm{x}) = \mathrm{x} \cdot \sigma(\mathrm{x}) \tag{6}$$

- Función ELU *(Exponential Linear Unit)*:
  Mejora la ReLU permitiendo valores negativos, lo que ayuda a centrar las activaciones en torno a cero y a acelerar la convergencia del entrenamiento.

$$\begin{aligned} f(x) &= x && si\ x > 0 \\ f(x) &= \alpha(e^{x} - 1) && si\ x \leq 0 \end{aligned} \tag{7}$$

- Función lineal:
  Esta función devuelve el valor de entrada sin alteración. Es útil en la capa de salida de redes neuronales para problemas de regresión, ya que permite que el modelo prediga valores reales sin restricciones.

$$f(x) = x \tag{8}$$

El proceso de aprendizaje en una red neuronal consiste en el ajuste de los pesos de sus conexiones para que la salida generada se aproxime lo máximo posible al resultado esperado. Este ajuste se realiza mediante un proceso iterativo de optimización llamado retropropagación del error (*backpropagation*), combinado con un algoritmo de descenso del gradiente.

Durante el entrenamiento, la red recibe ejemplos etiquetados (pares entrada–salida) y compara la salida que produce con la real. La diferencia entre ambas se mide mediante una función de pérdida, y este error se propaga hacia atrás a través de la red para ajustar los pesos en función de su contribución al error global [25]. Este proceso se repite durante múltiples épocas fijadas hasta que la red converge a un mínimo del error aceptable.

El algoritmo más común para minimizar la función de pérdida durante el entrenamiento es el descenso del gradiente estocástico (SGD), aunque existen variantes más avanzadas como Adam, RMSprop o Adagrad, que adaptan automáticamente la tasa de aprendizaje y mejoran la eficiencia del proceso. Estas son útiles en casos con grandes cantidades de datos.

Uno de los objetivos principales del entrenamiento es lograr que la red no solo memorice los datos de entrenamiento, sino que generalice bien nuevos datos. Cuando una red es demasiado compleja o se entrena durante demasiadas iteraciones, puede ocurrir en un fenómeno llamado sobreajuste (overfitting), que consiste en un rendimiento muy bueno sobre los datos ya conocidos, pero malo sobre datos no vistos.

Para evitar este problema, se aplican técnicas como:

- **Regularización** (L1, L2): añaden una penalización a los pesos excesivamente grandes en la función de pérdida, favoreciendo modelos más simples y generalizables.
- **Dropout**: desconecta aleatoriamente un porcentaje de neuronas durante cada iteración del entrenamiento, obligando a la red a no depender de neuronas concretas y reduciendo el sobreajuste.
- **Early stopping**: monitoriza el error de validación y detiene el entrenamiento cuando este comienza a empeorar, evitando que la red memorice el conjunto de entrenamiento.

- **División del conjunto de datos**: consiste en separar los datos en subconjuntos de entrenamiento, validación y prueba. Esto permite evaluar el rendimiento de la red de forma objetiva y detectar si hay sobreajuste.

Las principales ventajas de las redes neuronales son:

- Capacidad para modelar relaciones no lineales complejas.
- Flexibilidad para poder trabajar con datos ruidosos o incompletos.
- Se adaptan a múltiples tipos de datos.

Sin embargo, también presentan algunas necesidades:

- Necesidad de grandes volúmenes de datos para un correcto entrenamiento.
- Requieren una mayor potencia de cálculo y tiempo de entrenamiento.

A pesar de estas necesidades, su buen rendimiento en las tareas de predicción complejas ha reforzado su presencia en múltiples sectores industriales y científicos.

Aunque las redes neuronales fueron en un principio empleadas en tareas de clasificación o regresión en contextos estáticos, su capacidad se ha extendido con éxito al campo de las series temporales. En estos casos, la red adquiere la capacidad de predecir valores futuros de una secuencia a partir de datos pasados.

Las redes neuronales han sido aplicadas en múltiples sectores para predicción temporal: desde el consumo energético y el tráfico de vehículos, hasta el mercado financiero. Por ejemplo, Zhao et al. (2017) emplearon una red LSTM para predecir el tráfico en autopistas urbanas, obteniendo resultados superiores frente a modelos tradicionales en términos de precisión [20]. Del mismo modo, Fischer y Krauss (2018) demostraron que redes neuronales profundas pueden detectar patrones relevantes en series temporales bursátiles, mejorando la toma de decisiones en inversiones financieras de corto plazo. [26]

Existen múltiples arquitecturas de redes neuronales diseñadas para diferentes problemas de predicción en función de la naturaleza de los datos. El perceptrón multicapa (MLP) es una de las estructuras más elementales dentro las redes neuronales profundas. Está formado por varias capas de neuronas completamente conectadas y es recomendable su uso en tareas de clasificación y regresión sobre conjuntos de datos estáticos. Aunque no fue diseñado originalmente para tratar datos secuenciales, puede aplicarse a problemas temporales si los datos se reorganizan adecuadamente, por ejemplo, mediante el uso de ventanas deslizantes que simulan una estructura secuencial.

Las redes convolucionales (CNN), pensadas inicialmente para el procesamiento de imágenes, también han sido adaptadas al tratamiento de series temporales, Gracias

a su capacidad para identificar estructuras espaciales o temporales mediante filtros, han demostrado un rendimiento competitivo en tareas como el análisis de señales o el control del tráfico. [27]

Por otro lado, las redes neuronales recurrentes (RNN) están diseñadas específicamente para trabajar con datos secuenciales, ya que incorporan conexiones recurrentes que permiten mantener información de pasos anteriores. Esta característica les proporciona una memoria dinámica que resulta fundamental en tareas donde destaca el contexto temporal. Sin embargo, las RNN tradicionales presentan algunas limitaciones como el desvanecimiento del gradiente, lo que ha llevado al desarrollo de arquitecturas más avanzadas, como las redes LSTM. [22, capítulo 10]

Esta diversidad de arquitecturas ofrece un conjunto de herramientas versátiles para abordar distintos problemas reales, como la predicción temporal en sistemas complejos, donde la secuencialidad de los datos es clave.

2.3.2. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) son un tipo especializado de arquitectura neuronal diseñada específicamente para tratar con datos secuenciales o temporales. A diferencia de las redes neuronales tradicionales, donde la información fluye en una única dirección desde la entrada a la salida, las RNN introducen conexiones internas que les permiten obtener un estado oculto que depende de los pasos anteriores, otorgando así una "memoria" de datos pasados. Esta capacidad hace que las RNN sean particularmente útiles en tareas como el reconocimiento de voz, el modelado del lenguaje, la predicción financiera y, especialmente, la predicción de series temporales como en el caso de estudio sobre la demanda de agua. [22, capítulo 10]

Este estado oculto que aporta la particularidad a las RNN se calcula en el instante t como:

$$h_t = \sigma \cdot (b_1 + W \cdot h_{t-1} + U \cdot x_t) \quad (9)$$

Siendo x_t la entrada actual, h_{t-1} el estado oculto en el estado anterior, b_1 es el vector del sesgo, W y U son las matrices de pesos asociadas al estado anterior y a la entrada, respectivamente. Se aplica σ que representa la función de activación.

La salida de la red en el instante t se calcula de la siguiente manera:

$$o_t = b_2 + V \cdot h_t \quad (10)$$

Siendo h_t el estado oculto ya calculado en ese instante, V otra matriz de peso y b_2 otro vector de sesgo. [28]

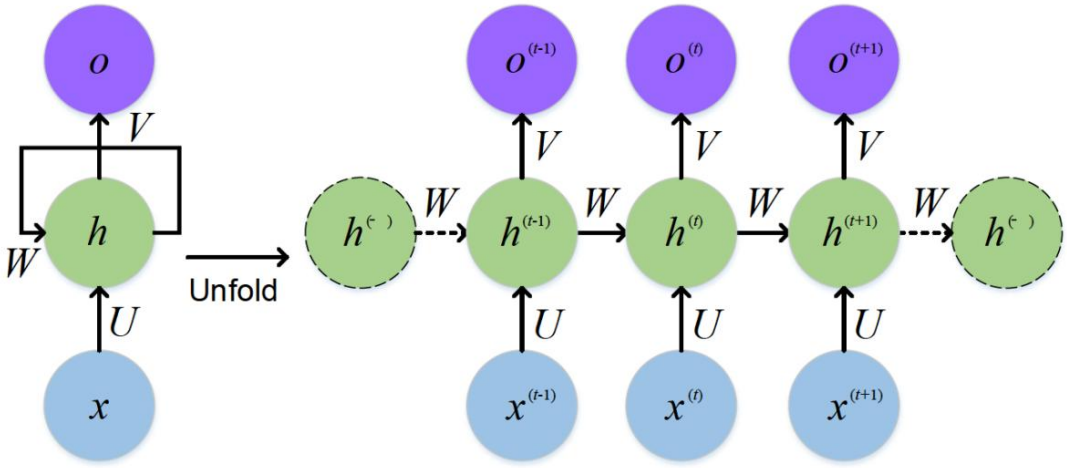

Figura 2.5: Arquitectura RNN

La arquitectura básica de una RNN incorpora el concepto de recurrencia, la cual se basa en una conexión que retroalimenta la salida de una neurona hacia sí misma o hacia otras neuronas de la misma capa en el siguiente paso de tiempo (Figura 2.5). Esto permite que la red tenga en cuenta no solo el dato actual, sino también los estados anteriores, proporcionando así la memoria a corto plazo.

Las RNN son particularmente adecuadas para el análisis y la predicción de series temporales, ya que permiten capturar dependencias entre los valores pasados y presentes de una secuencia. En el contexto de la predicción de demanda de agua, donde los valores de consumo dependen fuertemente de los días previos, las condiciones climáticas o eventos estacionales, las RNN ofrecen una solución mucho más potente que los modelos clásicos, como ARIMA o regresiones lineales multivariantes. [29]

Además, las RNN pueden integrarse con variables exógenas (como temperatura o día de la semana) para realizar predicciones multivariadas, lo que las hace útiles en sistemas de abastecimiento urbano y planificación de recursos.

A pesar de su potencial, las RNN tradicionales presentan limitaciones importantes en su funcionamiento. Uno de los problemas más conocidos es el desvanecimiento del gradiente (vanishing gradient), que ocurre durante el proceso de entrenamiento mediante retropropagación a través del tiempo. A medida que el algoritmo actualiza los pesos de las conexiones para minimizar el error, los gradientes pueden decrecer exponencialmente, haciendo que las actualizaciones de pesos sean insignificantes. Esto impide que la red aprenda dependencias a largo plazo, ya que la información relevante del pasado se pierde antes de influir en la predicción futura. [30]

El gradiente desvanecido limita seriamente la capacidad de las RNN para capturar relaciones temporales más allá de unas pocas unidades de tiempo. Para problemas

donde la información relevante se encuentra lejos en el pasado representa un obstáculo considerable.

Como solución a estas limitaciones, surgieron arquitecturas mejoradas como las redes LSTM (Long Short-Term Memory), desarrolladas por Hochreiter y Schmidhuber en 1997. Estas redes incorporan una estructura interna más compleja basada en celdas de memoria y compuertas que permiten controlar el flujo de información a lo largo del tiempo, y así retener o descartar información de forma más eficaz. [31]

No obstante, es importante destacar que las RNN estándar siguen siendo útiles en contextos donde las dependencias temporales son cortas, o cuando se combinan con otras técnicas como el suavizado exponencial o el preprocesado diferenciado de datos, lo que puede mitigar en parte el desvanecimiento del gradiente. [15]

2.3.3. Redes LSTM

Las redes neuronales Long Short-Term Memory (LSTM) se consideran una evolución importante dentro de la familia de redes neuronales recurrentes (RNN), orientadas a dar solución a uno de sus principales inconvenientes, la incapacidad para aprender dependencias a largo plazo en series temporales. Las LSTM fueron propuestas por Hochreiter y Schmidhuber en 1997 como una solución a los problemas de desvanecimiento y explosión del gradiente que dificultaban el entrenamiento efectivo de las RNN clásicas. [31]

Las LSTM están diseñadas para recordar información durante intervalos largos de tiempo, lo cual resulta crucial en tareas donde el contexto temporal influye notablemente en la predicción, como ocurre en la demanda de agua, la predicción meteorológica, la traducción automática o la generación de texto. A diferencia de las RNN estándar, las redes LSTM incorporan una estructura interna compleja basada en celdas de memoria y compuertas que controlan el flujo de la información.

La unidad principal de una red LSTM es su celda de memoria (Figura 2.6), la cual está compuesta por tres compuertas principales: la compuerta de olvido, la compuerta de entrada y la compuerta de salida. Estas compuertas utilizan funciones sigmoides y tangentes hiperbólicas para decidir qué información se debe mantener, actualizar o descartar del estado interno de la celda.

- **Compuerta de olvido (forget gate)**: determina qué información del estado anterior de la celda debe eliminarse. Utiliza una función sigmoide que recibe como entrada el estado anterior de la celda y la nueva entrada.

- **Compuerta de entrada (input gate):** controla la cantidad de nueva información que se incorporará al estado de la celda. Está formada por una función sigmoide i_t, que determina qué parte del nuevo contenido debe añadirse, y

una compuerta de candidatura g_t, generada mediante una tangente hiperbólica, que representa los nuevos valores propuestos para actualizar el estado.

- **Compuerta de salida (output gate)**: decide qué parte del estado interno de la celda debe transmitirse a la siguiente unidad de la red. También utiliza una función sigmoide y una tangente hiperbólica para generar la salida.

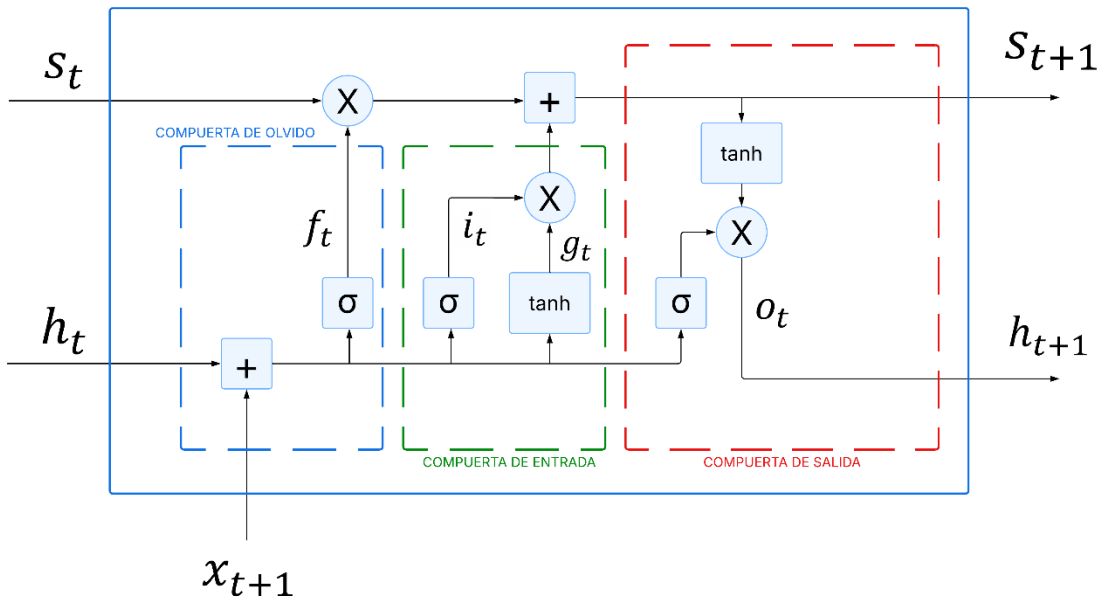

Figura 2.6: Esquema de celda de memoria LSTM

- s_t y s_{t+1} : estado actual y estado siguiente de la celda
- x_t : es la entrada a la celda en el instante t
- h_{t+1} : información que se envía a la siguiente capa oculta
- f_t : compuerta de olvido
- i_t : compuerta de entrada
- g_t : compuerta de candidatura
- o_t : compuerta de salida
- $tanh$: tangente hiperbólica
- σ : función sigmoidal

El funcionamiento completo de la celda LSTM puede expresarse mediante las siguientes ecuaciones:

$$s_{t+1} = f_{t+1} \cdot s_t + i_{t+1} \cdot g_{t+1} \quad (11)$$

$$h_{t+1} = o_{t+1} \cdot tanh(s_{t+1}) \quad (12)$$

$$f_{t+1} = \sigma \cdot (R_f h_t + W_f x_{t+1} + b_f) \quad (13)$$

$$i_{t+1} = \sigma \cdot (R_i\, h_t + W_i\, x_{t+1} + b_i) \quad (14)$$

$$g_{t+1} = \tanh \cdot (R_g\, h_t + W_g\, x_{t+1} + b_g) \quad (15)$$

$$o_{t+1} = \sigma \cdot (R_o\, h_t + W_o\, x_{t+1} + b_o) \quad (16)$$

A mayores de los factores y variables ya vistos, W son los pesos sinápticos, R los pesos recurrentes y b los sesgos.

Estas operaciones permiten que la LSTM mantenga de manera efectiva la información relevante y descarte la irrelevante, una capacidad que resulta vital para las tareas de predicción a largo plazo.

Las LSTM han demostrado ser especialmente eficaces en problemas de series temporales, donde los datos presentan una dependencia secuencial que no puede ser capturada adecuadamente por arquitecturas más tradicionales como las redes neuronales artificiales (ANN) o incluso por RNN simples. Algunos ejemplos destacados de los usos de redes LSTM incluyen:

- Predicción del consumo energético o demanda de agua en función de patrones históricos y condiciones climáticas. [32]
- Reconocimiento de voz y procesamiento del lenguaje natural, donde la secuencia de palabras determina el significado.
- Análisis financiero y detección de fraudes, aprovechando la capacidad de modelar fluctuaciones temporales.
- Diagnóstico médico a partir de señales fisiológicas (como ECG o EEG) que requieren análisis de secuencias temporales complejas [33].

Las redes LSTM presentan varias ventajas frente a las redes recurrentes convencionales:

- **Memoria a largo plazo**: gracias a sus compuertas, pueden mantener información durante muchos pasos temporales sin que el gradiente se desvanezca.
- **Flexibilidad estructural**: permiten arquitecturas profundas o bidireccionales, donde las secuencias se procesan en ambos sentidos temporales.
- **Robustez frente al ruido**: al poder descartar la información considerada como irrelevante, pueden manejar mejor datos ruidosos o incompletos.

Sin embargo, las redes LSTM también tienen ciertas desventajas. Requieren una mayor capacidad computacional, y su entrenamiento es más complejo debido al número de parámetros y funciones involucradas. Es común aplicar regularización

mediante técnicas como *dropout* para evitar el sobreajuste y mejorar la generalización del modelo.

Una red LSTM completa suele componerse de varias capas secuenciales que permiten capturar dependencias temporales en los datos:

1. **Capa de entrada secuencial (Sequence Input):** introduce la serie temporal como una secuencia de vectores ordenados en el tiempo.
2. **Capas LSTM ocultas:** constituyen el núcleo del modelo. Estas capas procesan las secuencias mediante celdas de memoria capaces de retener información a corto y largo plazo.
3. **Capas de regularización (Dropout):** desconectan aleatoriamente conexiones durante el entrenamiento, lo que ayuda a reducir el sobreajuste.
4. **Capa *Fully Connected* o *Dense Layer*:** transforma la salida de la última capa LSTM en un vector adecuado para la predicción.
5. **Capa de salida:** genera el valor final de salida, que puede ser un valor continuo (regresión) o una clase (clasificación), dependiendo de la tarea.

Esta arquitectura puede adaptarse en el número de unidades según la complejidad del problema y la naturaleza de los datos [34].

En los últimos años, se han desarrollado variantes como las redes GRU (Gated Recurrent Unit), que simplifican la estructura LSTM manteniendo un rendimiento competitivo [34], y las redes bidireccionales (BiLSTM), que procesan la información tanto en sentido directo como inverso [35]. También han surgido modelos híbridos, como las ConvLSTM para datos espaciotemporales [36], y arquitecturas más recientes basadas en mecanismos de atención, como los Transformers [37], que han redefinido el enfoque hacia el procesamiento secuencial.

Gracias a su capacidad para modelar dinámicas temporales complejas, las LSTM son hoy una de las herramientas más utilizadas en el aprendizaje profundo aplicado a series temporales, y constituyen una base robusta para sistemas de predicción inteligente en entornos urbanos.

CAPÍTULO III:
DATOS Y PREPROCESADO

3. Datos y preprocesado

3.1. Fuentes de datos utilizadas

Para el desarrollo del modelo predictivo basado en redes neuronales LSTM, se han recopilado y organizado diferentes conjuntos de datos que reflejan el comportamiento histórico de la demanda de agua en Valladolid, así como factores externos que pueden influir en dicha demanda.

La información relacionada con el suministro de agua fue proporcionada por Aquavall, entidad pública responsable de la gestión del ciclo urbano del agua en Valladolid. La base de datos proporcionada incluye múltiples variables asociadas a las diferentes operaciones de las Estaciones de Tratamiento de Agua Potable (ETAP), como caudales captados, caudales suministrados, niveles de turbidez, volumen almacenado, consumo energético, entre otras. Sin embargo, para el presente trabajo se han empleado exclusivamente las variables de caudales impulsados a la red de abastecimiento, al ser la más representativa del comportamiento de la demanda y la más adecuada como serie objetivo del modelo predictivo. En concreto se ha empleado la variable abreviada como *"Q_Red_Eras"* que refleja el caudal de impulsión directa a la red de la ciudad por la ETAP de Las Eras.

Esta y todas las variables abarcan un periodo comprendido entre el 1 de enero de 1999 y el 30 de octubre de 2024, con una frecuencia diaria. Obteniendo un total de 9.435 series diarias de todas las variables proporcionadas.

Además, se han incorporado variables meteorológicas para mejorar el funcionamiento del modelo con información ambiental relevante. En concreto, se utilizaron:

- La temperatura media diaria (en grados Celsius).
- La humedad relativa media diaria (en porcentaje).

Estos datos fueron obtenidos a través de la API histórica de Open-Meteo [38], un servicio de acceso libre que proporciona registros meteorológicos pasados basados en las coordenadas geográficas. Para este proyecto, se consultaron y obtuvieron los datos correspondientes a la ciudad de Valladolid, generando un conjunto con igual frecuencia y periodo que la serie de consumo de agua.

Todos los conjuntos de datos fueron integrados en un único marco temporal, sincronizados día a día y filtrados para eliminar registros incompletos o erróneos.

3.2. Limpieza y preparación de los datos

Una vez se han recopilado e incorporado las diferentes fuentes de información, se lleva a cabo un proceso de limpieza, transformación y preparación de los datos, con el objetivo de obtener una estructura homogénea, libre de errores, alineada temporalmente y adecuada para ser utilizada en el modelo de predicción.

En primer lugar, se importaron los datos proporcionados por Aquavall desde un archivo CSV. Se convierte la columna de fechas al tipo *datetime*, estableciéndola como índice temporal del conjunto de datos. Esta transformación permite ordenar cronológicamente los registros y facilitar su posterior fusión con las variables meteorológicas.

De entre todas las columnas disponibles del archivo CSV proporcionado por Aquavall, se selecciona la correspondiente al caudal de impulsión directa a la red de la ciudad por la ETAP de Las Eras, llamada *Q_Red_Eras* (correspondiente a la columna con el índice 8) al ser la variable que se utilizaría como serie objetivo. A continuación, se aplica una limpieza de esta columna, reemplazando todos los valores nulos o iguales a cero (que no reflejan un caudal real) por la media de la serie calculada o por un valor obtenido mediante la interpolación, con el fin de evitar distorsiones en el entrenamiento del modelo.

A su vez, se cargan los datos meteorológicos históricos desde otro archivo CSV. Estos incluyen la temperatura media diaria y la humedad relativa. Al igual que con la serie del caudal, se transforma la columna de fechas al formato *datetime* y se establece como índice. Por último, se separan las dos columnas de temperatura y humedad para poder aplicar más adelante una función a todos los datos.

A continuación, se filtran ambas series (la hídrica y la meteorológica) para mantener únicamente los registros correspondientes a fechas comunes. Esto garantiza que cada fila del conjunto final represente los mismos días completos con la información correcta y correspondiente a ese día de las variables de caudal, temperatura y humedad.

Como paso adicional antes de unir las tres variables en un mismo *DataFrame*, se aplica sobre las tres columnas (caudal, temperatura y humedad) una función de media deslizante con una ventana de 7 días para suavizar la serie, reducir el ruido y que los grandes picos en los datos del caudal no afecten al resultado final. Esta técnica ayuda a suavizar las fluctuaciones puntuales manteniendo las tendencias relevantes de la serie, lo que facilita la detección de patrones temporales por parte del modelo LSTM.

Finalmente, se unifican todas las variables en un único *DataFrame* ordenado y listo para su posterior uso como entrada en el modelo. Las columnas finales se renombran como: *Q_Red_Eras, temperature* y *relative_humidity*.

3.3. Escalado y normalización

Las redes neuronales, en particular las LSTM, son sensibles en lo referido a la escala de los datos, por ello es necesario aplicar técnicas de normalización que garanticen que todas las variables de entrada contribuyen de forma equilibrada al proceso de aprendizaje. En este caso se ha optado por un escalado de tipo Min-Max en el rango de [-1, 1], aplicado a todas las variables del modelo: caudal, temperatura y humedad.

Este escalado se ha llevado a cabo mediante la función *escalar_dataset*, que permite escalar por separado cada una de las variables (features) utilizando un objeto *MinMaxScaler* de la librería *scikit-learn*. En total, se emplea un escalador independiente para cada variable, todos ajustados únicamente con los datos del conjunto de entrenamiento, para evitar cualquier tipo de fuga de información. Posteriormente, estos escaladores se aplican también al conjunto de validación y prueba, para garantizar la coherencia entre particiones.

Este procedimiento se aplica a los tres subconjuntos de datos generados tras la partición temporal:

- *x_tr, y_tr*: conjunto de entrenamiento
- *x_vl, y_vl*: conjunto de validación
- *x_ts, y_ts*: conjunto de prueba

Además, se almacena el escalador empleado para la variable objetivo para así poder invertir el escalado después de la predicción y poder evaluar los resultados en las unidades reales e iniciales.

El escalado uniforme empleado mejora la estabilidad del entrenamiento y facilita que el modelo aprenda de forma más eficiente, al evitar que las diferencias de magnitud entre variables distorsionen el cálculo de los gradientes.

CAPÍTULO IV:

IMPLEMENTACIÓN Y EXPERIMENTACIÓN DEL MODELO

4. Implementación y experimentación del modelo

4.1. Descripción del entorno de desarrollo

El desarrollo del modelo de predicción se ha realizado utilizando el lenguaje de programación Python, dentro del entorno de trabajo proporcionado por *Anaconda*, concretamente mediante su interfaz integrada *Spyder*. Esta combinación facilita la escritura de código, la ejecución controlada de scripts, y la visualización de datos.

Las principales bibliotecas utilizadas a lo largo del proyecto fueron:

- **pandas**: para la lectura, manipulación y limpieza de datos estructurados en formato tabular.
- **numpy**: para operaciones matemáticas sobre matrices y vectores, y para optimizar cálculos numéricos.
- **matplotlib.pyplot**: para la generación de gráficos y visualización de series temporales.
- **scikit-learn (sklearn)**:
    - MinMaxScaler: para el escalado de variables al rango [-1, 1].
    - mean_squared_error: para la evaluación cuantitativa del modelo.
- **tensorflow.keras**:
    - Sequential, LSTM, Dense y Dropout: para la definición y construcción de la arquitectura de la red neuronal.
    - RMSprop: como optimizador para el entrenamiento de la red.
- **tensorflow**: como backend (motor encargado de ejecutar las ooperaciones) para el entrenamiento de modelos en CPU.

La combinación de estas herramientas ha permitido implementar de forma eficiente todo el flujo de trabajo del proyecto, desde la carga y preprocesamiento de los datos hasta la construcción, entrenamiento y evaluación del modelo LSTM.

4.2. Diseño del modelo LSTM

Tras el preprocesamiento de los datos (incluyendo la sustitución de los valores nulos o cero por la media de la serie de datos o un valor obtenido por interpolación, el escalado de variables y el suavizado mediante una media deslizante de 7 días), se continua con el diseño del modelo de red neuronal.

Para el desarrollo de la red neuronal se emplearon redes LSTM (Long Short-Term Memory), debido a su idoneidad para modelar series temporales con dependencias a medio y largo plazo, como es el caso de la demanda de agua potable.

Se han creado cuatro variantes del modelo, combinando enfoques univariantes y multivariantes, así como predicción a un paso (unistep) y a varios pasos (multistep):

- Univariado+unistep:
  Solo se ha empleado la serie histórica del caudal de agua como la entrada, y se predice el valor de un solo paso, es decir, un solo día.
- Univariado+multistep:
  También se emplea solo como entrada los datos del caudal, pero ahora la salida consiste en una predicción a varios pasos, en este caso de una semana, 7 días.
- Multivariado+unistep:
  En este caso se incorporan a la entrada las variables climáticas de temperatura media y humedad relativa media, además del caudal. Se vuelve a predecir el valor de un solo día.
- Multivariado+multistep:
  En este caso se siguen incorporando las tres variables, de clima y caudal, y se predice los valores para los siguientes 7 días.

Al crear estos cuatro modelos se ha podido comparar el rendimiento de cada uno de ellos con diferentes parámetros y variantes que dispone la red, y así poder obtener el mejor modelo posible para la predicción de la demanda de agua.

Las redes LSTM requieren que los datos de entrada estén organizados como secuencias temporales. Para ello, se divide la serie en bloques temporales consecutivos (ventanas de entrada) que actúan como muestras para entrenar el modelo. Se han realizado simulaciones con dos tamaños de ventana:

- Una semana (7 días) de historial como entrada.
- Un mes (30 días) de historial como entrada.

En cada caso particular, el modelo recibe como entrada la secuencia de valores de los días anteriores y genera una predicción de 1 o varios días futuros, según el tipo de modelo.

El modelo fue construido mediante la API Sequential de *Keras*, permitiendo dos configuraciones principales:

- Una arquitectura más simple con una sola capa LSTM con 128 unidades, una capa intermedia *Dropout* para reducir el riesgo de sobreajuste y finalmente una capa *Dense* que genera la salida.

- Una arquitectura más profunda, con dos capas LSTM apiladas y dos capas intermedias *Dropout* para también reducir el riesgo de sobreajuste. La primera capa LSTM tiene 128 unidades y emplea *return_sequences=True* para devolver la secuencia completa. A continuación dispone de la capa *Dropout* con un valor de 0,3, que desactiva aleatoriamente el 30% de las conexiones durante el entrenamiento. Seguidamente se encuentra la segunda capa LSTM con 64 unidades con otra capa *Dropout*, y finalmente la capa de salida *Dense*.

Las capas *Dropout* mencionadas tienen como finalidad reducir el sobreajuste, obligando a la red a no depender excesivamente de ciertas neuronas durante el entrenamiento. Esta técnica mejora la generalización del modelo a datos no vistos.

La capa de salida es una capa totalmente conectada (*Dense*) con función de activación lineal, lo cual es apropiado en tareas de regresión continua como esta. La función lineal simplemente devuelve el valor calculado por la neurona sin aplicar ninguna transformación, permitiendo que el modelo produzca cualquier valor real dentro del rango de la variable a predecir, en este caso el caudal.

Durante el entrenamiento del modelo se han usado dos algoritmos de optimización: *RMSprop* (*Root Mean Square Propagation*) y *Adam* (*Adaptive Moment Estimation)*. Ambos son variantes del descenso del gradiente, una técnica base en el aprendizaje de redes neuronales. Un optimizador es el componente del modelo encargado de ajustar los pesos de las neuronas en cada iteración del entrenamiento, con el objetivo de minimizar la función de pérdida. Actúa como una guía y se encarga de determinar cuánto y en qué dirección deben modificarse los pesos del modelo tras evaluar el error cometido.

Los algoritmos utilizados fueron:

- *RMSprop*:
  Diseñado para redes neuronales recurrentes y secuenciales, *RMSprop* ajusta la tasa de aprendizaje de cada parámetro en función de la magnitud de sus gradientes. Esto le permite adaptarse bien a los datos con ruido o con dinámicas temporales complejas.

- *Adam*:
  Combina las ventajas de *RMSprop* y del optimizador *Momentum*. *Adam* no solo adapta la tasa de aprendizaje para cada parámetro, sino que también

incorpora el historial de los gradientes para acelerar la convergencia y amortiguar las oscilaciones.

La posibilidad de alternar entre ambos algoritmos permite realizar comparaciones lo que favorece el análisis del rendimiento del modelo con cada algoritmo.

4.3. Configuración y entrenamiento del modelo

Una vez definida la arquitectura del modelo, se procede a su entrenamiento sobre el conjunto de datos procesados. Para garantizar una buena evaluación del rendimiento y evitar el sobreajuste, se dividieron los datos en tres subconjuntos:

- 80% de los datos para entrenamiento (*Train*),
- 10% de los datos para validación (*Validation*),
- 10% de los datos para prueba (*Test*).

Esta división permitió ajustar los hiperparámetros y evaluar el modelo sobre datos no vistos en el entrenamiento ni en la validación.

El modelo fue entrenado durante 100 épocas, valor suficiente para asegurar la convergencia sin provocar sobreajuste. Se empleó un tamaño de lote (*batch_size*) de 32, que proporcionó un mejor equilibrio entre la velocidad de predicción y la estabilidad del aprendizaje en comparación con otros valores más grandes.

La función de pérdida empleada ha sido la raíz del error cuadrático medio (RMSE), que penaliza más los errores grandes y proporciona una métrica más interpretable al estar en las mismas unidades que la variable objetivo a predecir.

Se genera una gráfica para poder observar el comportamiento del modelo mediante la evolución de la pérdida (*loss*) en los conjuntos de entrenamiento y validación. Estas curvas permiten verificar que no se produce sobreajuste (*overfitting*) durante el entrenamiento.

Finalmente, se comparan los valores reales y los valores predichos tanto gráficamente como mediante valores relevantes obtenidos en el modelo. Se muestra por pantalla los resultados de evaluación en los conjuntos de entrenamiento, validación y prueba, incluyendo valores como el RMSE (raíz del error cuadrático medio), la media del error absoluto (MAE) y el error relativo medio en porcentaje. Estos indicadores permiten analizar el comportamiento general del modelo y facilitar la comparación entre las distintas configuraciones implementadas.

4.4. Predicción univariada

En este apartado se presentan los resultados obtenidos al entrenar modelos LSTM utilizando exclusivamente la variable de caudal como entrada. Este enfoque, conocido como predicción univariada, permite evaluar si la serie temporal contiene suficiente información para predecir su evolución sin recurrir a variables externas.

Se han considerado dos configuraciones distintas de salida:

- Predicción a un paso (*unistep*), en la que el modelo predice únicamente el valor del día siguiente.
- Predicción a múltiples pasos (*multistep*), en la que el modelo predice de forma simultánea los valores de los siguientes 7 días.

Estas variantes permiten valorar cómo influye el horizonte temporal de predicción en el rendimiento del modelo cuando solo se utiliza la información interna de la serie.

4.4.1. Predicción univariada-unistep

En esta configuración, el modelo LSTM recibe como entrada únicamente los valores históricos del caudal durante una ventana de tiempo fija (7 o 30 días), y genera como salida el caudal del día siguiente.

La simplicidad del modelo permite evaluar la capacidad de la serie de caudal para autorreproducirse en el corto plazo, sin necesidad de otras fuentes de información.

A continuación se presentan los resultados obtenidos en esta configuración, así como las gráficas de comparación entre los valores reales y los predichos.

MODELO 1.1: 1 capa, 64 unidades, media, optimizador *Adam*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 7 días. (Figura 4.1)

Resultados de simulación obtenidos:

RMSE train: 0.126

RMSE val: 0.134

RMSE test: 0.175

Media del error absoluto (MAE): 4992.715

Error en porcentaje (%): 12.79 %

RMSE: 7934.709

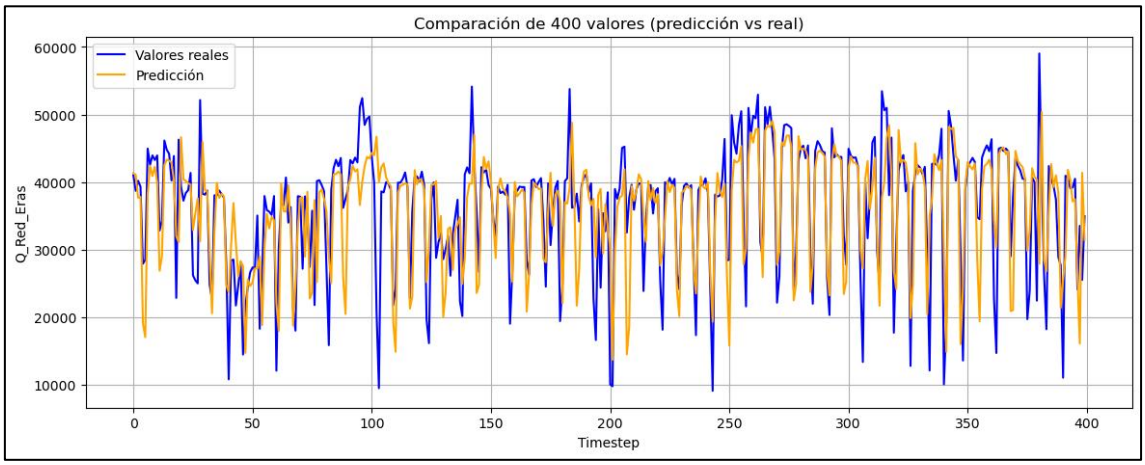

Figura 4.1: Comparación de los valores reales y predichos por el modelo 1.1

MODELO 1.2: 1 capa, 64 unidades, interpolación, optimizador *Adam*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *Adam* y una ventana de entrada de 7 días. (Figura 4.2)

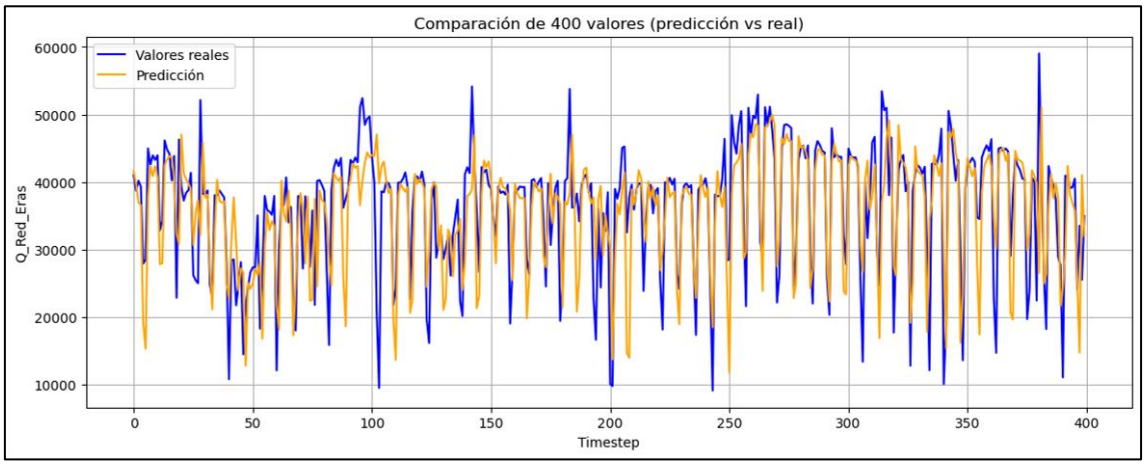

Figura 4.2: Comparación de los valores reales y predichos por el modelo 1.2

Resultados de simulación obtenidos:

RMSE train: 0.128

RMSE val: 0.136

RMSE test: 0.168

Media del error absoluto (MAE): 4921.628

Error en porcentaje (%): 12.61 %

RMSE: 7585.327

MODELO 1.3: 1 capa, 64 unidades, media, optimizador *RMSprop*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *RMSprop* y una ventana de entrada de 7 días. (Figura 4.3)

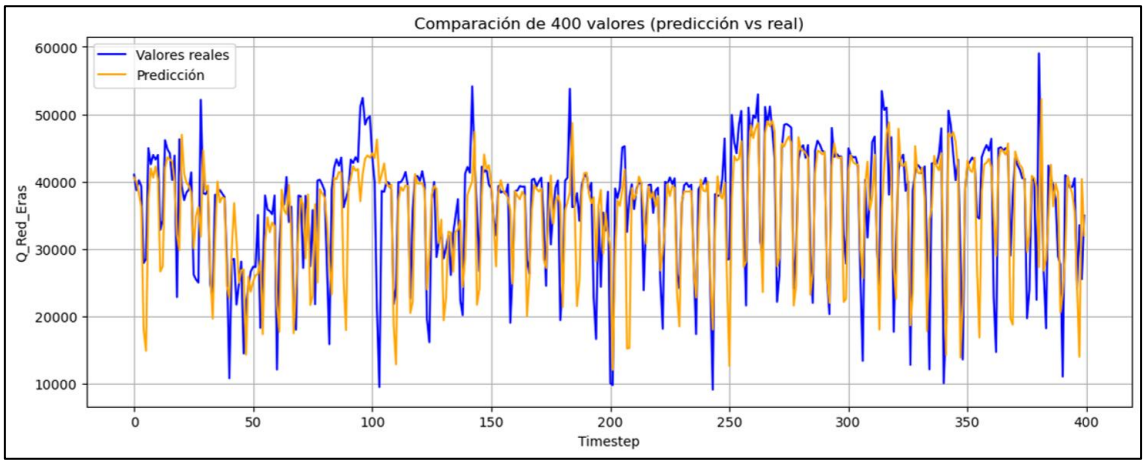

Figura 4.3: Comparación de los valores reales y predichos por el modelo 1.3

Resultados de simulación obtenidos:

RMSE train: 0.127

RMSE val: 0.137

RMSE test: 0.180

Media del error absoluto (MAE): 5140.488

Error en porcentaje (%): 13.17 %

RMSE: 8109.704

MODELO 1.4: 1 capa, 64 unidades, interpolación, optimizador *RMSprop*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 7 días. (Figura 4.4)

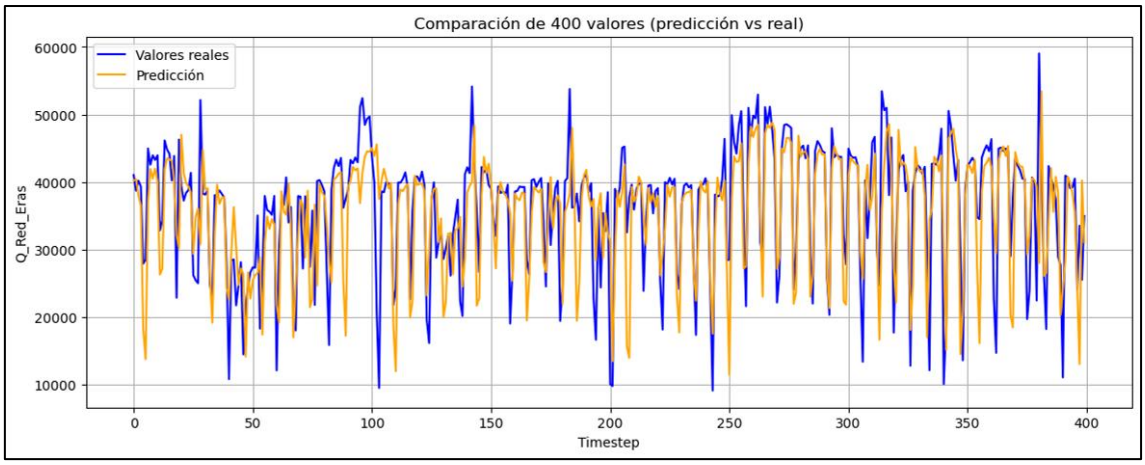

Figura 4.4: Comparación de los valores reales y predichos por el modelo 1.4

Resultados de simulación obtenidos:

RMSE train: 0.128

RMSE val: 0.137

RMSE test: 0.169

Media del error absoluto (MAE): 4949.256

Error en porcentaje (%): 12.68 %

RMSE: 7586.522

MODELO 1.5: 1 capa, 64 unidades, media, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.5)

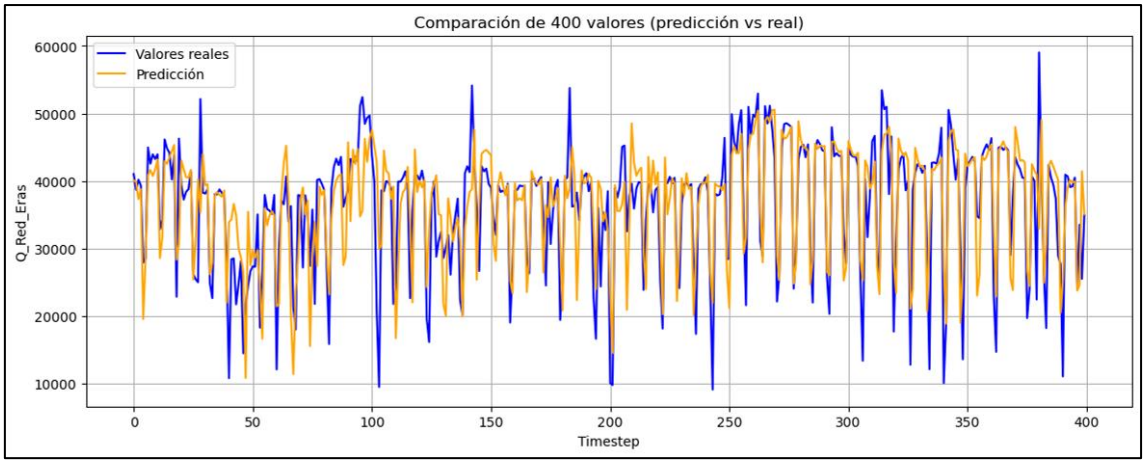

Figura 4.5: Comparación de los valores reales y predichos por el modelo 1.5

Resultados de simulación obtenidos:

RMSE train: 0.108

RMSE val: 0.127

RMSE test: 0.165

Media del error absoluto (MAE): 4877.597

Error en porcentaje (%): 12.49 %

RMSE: 7550.969

MODELO 1.6: 1 capa, 64 unidades, interpolación, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.6)

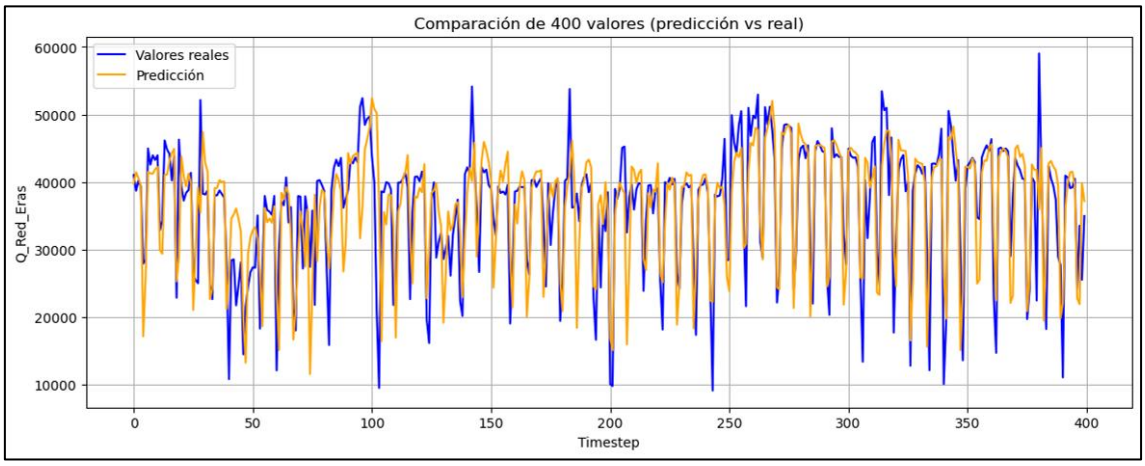

Figura 4.6: Comparación de los valores reales y predichos por el modelo 1.6

Resultados de simulación obtenidos:

RMSE train: 0.108

RMSE val: 0.126

RMSE test: 0.164

Media del error absoluto (MAE): 4866.68

Error en porcentaje (%): 12.47 %

RMSE: 7395.145

MODELO 1.7: 1 capa, 64 unidades, media, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.7)

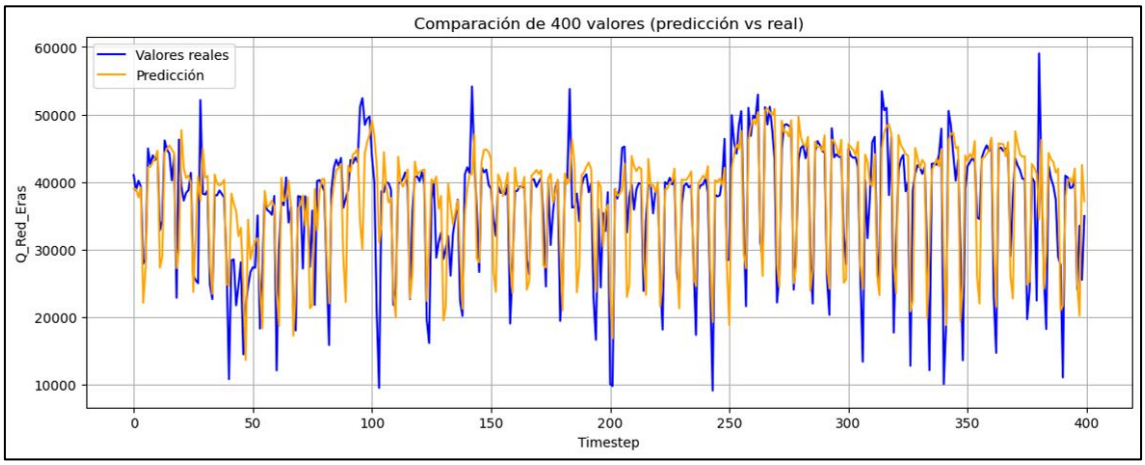

Figura 4.7: Comparación de los valores reales y predichos por el modelo 1.7

Resultados de simulación obtenidos:

RMSE train: 0.120

RMSE val: 0.124

RMSE test: 0.157

Media del error absoluto (MAE): 4541.014

Error en porcentaje (%): 11.63 %

RMSE: 7102.97

MODELO 1.8: 1 capa, 64 unidades, interpolación, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.8)

Resultados de simulación obtenidos:

RMSE train: 0.118

RMSE val: 0.125

RMSE test: 0.159

Media del error absoluto (MAE): 4711.812

Error en porcentaje (%): 12.07 %

RMSE: 7200.885

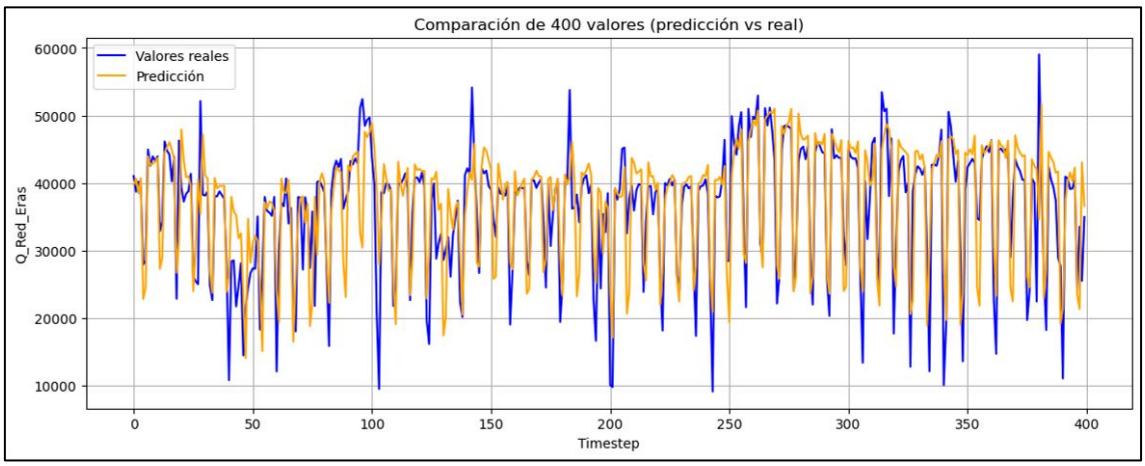

Figura 4.8: Comparación de los valores reales y predichos por el modelo 1.8

Una vez realizadas todas las simulaciones posibles para una red LSTM de una sola capa de 64 unidades, se llega a la conclusión de que el modelo que mejores resultados ha proporcionado es el Modelo 7. Aunque cabe destacar que ninguno de los modelos implementados hasta ahora es capaz de ajustarse a los picos del caudal, obteniendo un error no aceptable en todos.

Con la información obtenida con estas simulaciones, se implementaron nuevos modelos, ahora con una sola capa y 128 unidades. Simulando ahora modelos con esta nueva característica junto con las variables que mejores resultados otorgaron en las anteriores simulaciones.

MODELO 1.9: 1 capa, 128 unidades, interpolación, optimizador *Adam*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *Adam* y una ventana de entrada de 7 días. (Figura 4.9)

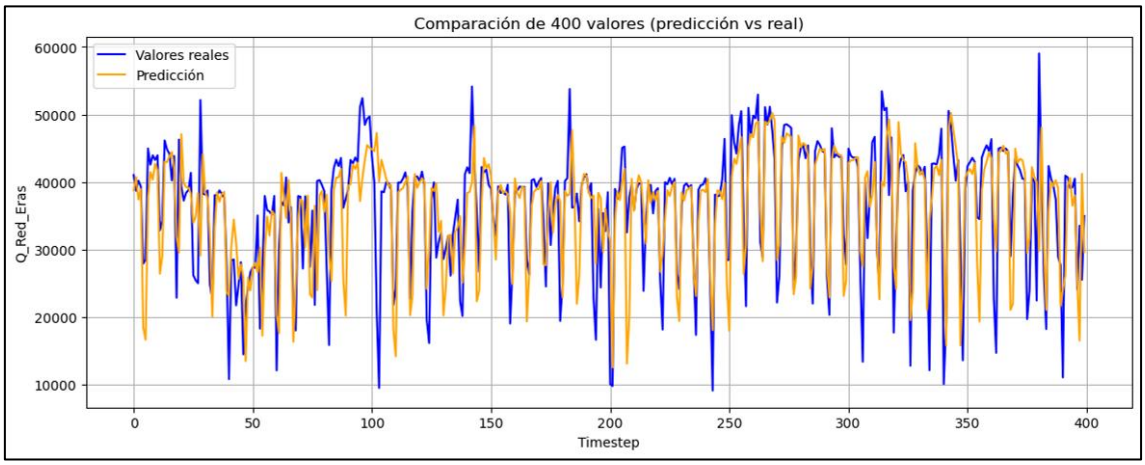

Figura 4.9: Comparación de los valores reales y predichos por el modelo 1.9

Resultados de simulación obtenidos:

RMSE train: 0.123

RMSE val: 0.133

RMSE test: 0.165

Media del error absoluto (MAE): 4738.452

Error en porcentaje (%): 12.14 %

RMSE: 7437.765

MODELO 1.10: 1 capa, 128 unidades, media, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.10)

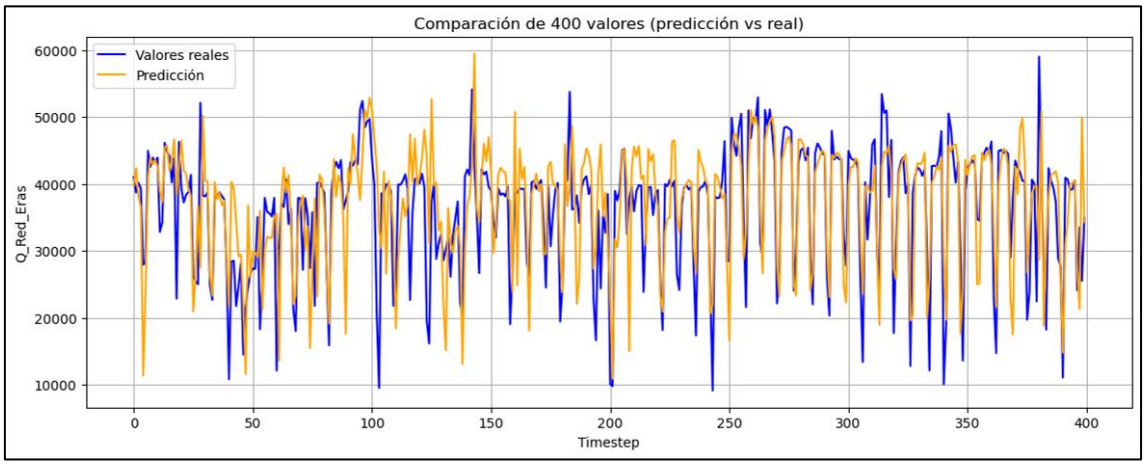

Figura 4.10: Comparación de los valores reales y predichos por el modelo 1.10

Resultados de simulación obtenidos:

RMSE train: 0.093

RMSE val: 0.142

RMSE test: 0.186

Media del error absoluto (MAE): 5702.188

Error en porcentaje (%): 14.61 %

RMSE: 8429.598

MODELO 1.11: 1 capa, 128 unidades, media, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.11)

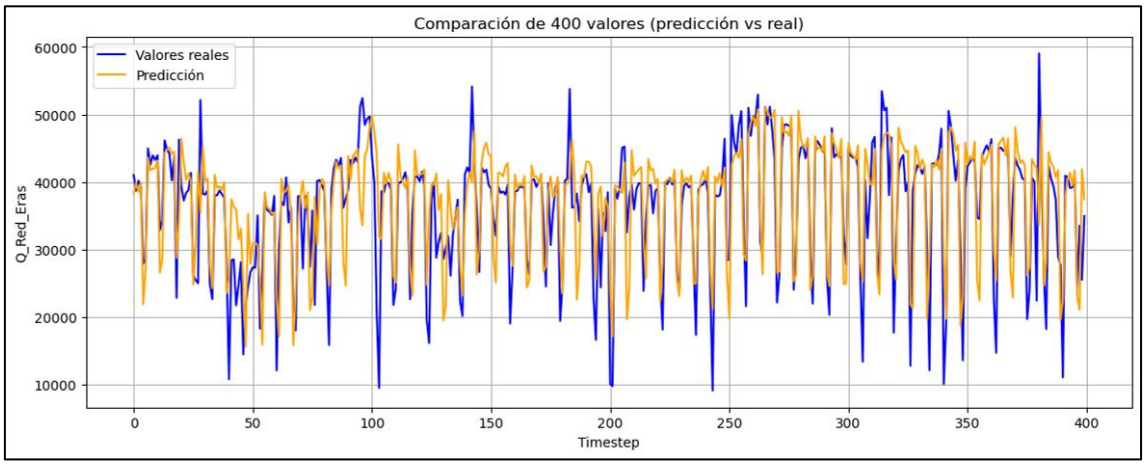

Figura 4.11: Comparación de los valores reales y predichos por el modelo 1.11

Resultados de simulación obtenidos:

RMSE train: 0.120

RMSE val: 0.125

RMSE test: 0.161

Media del error absoluto (MAE): 4739.252

Error en porcentaje (%): 12.14 %

RMSE: 7314.61

MODELO 1.12: 1 capa, 128 unidades, interpolación, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.12)

Resultados de simulación obtenidos:

RMSE train: 0.119

RMSE val: 0.125

RMSE test: 0.159

Media del error absoluto (MAE): 4710.431

Error en porcentaje (%): 12.07 %

RMSE: 7220.852

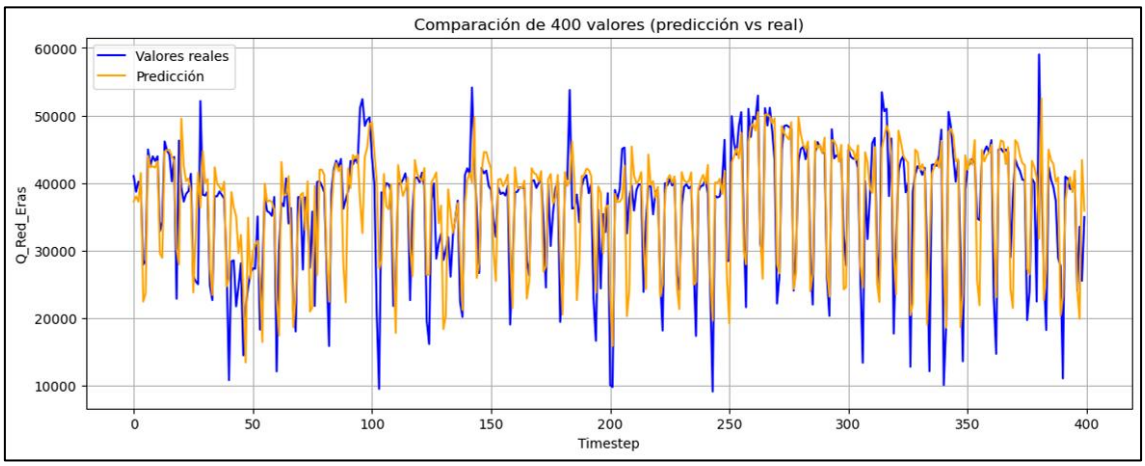

Figura 4.12: Comparación de los valores reales y predichos por el modelo 4.12

Una vez terminadas las simulaciones para una red LSTM de una capa de 128 unidades, se llega a la conclusión de que aunque la mayoría de las predicciones se ajustan a la tendencia, ninguna genera una predicción fiable ni aceptable por el alto error que se obtiene.

Por último se realiza un modelo con dos capas LSTM, la primera de 128 unidades y la segunda de 64. Se emplean las variables que menor error proporcionaron en las últimas simulaciones de una capa de 128 unidades.

MODELO 1.13: 2 capas, 128 y 64 unidades, interpolación, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.13)

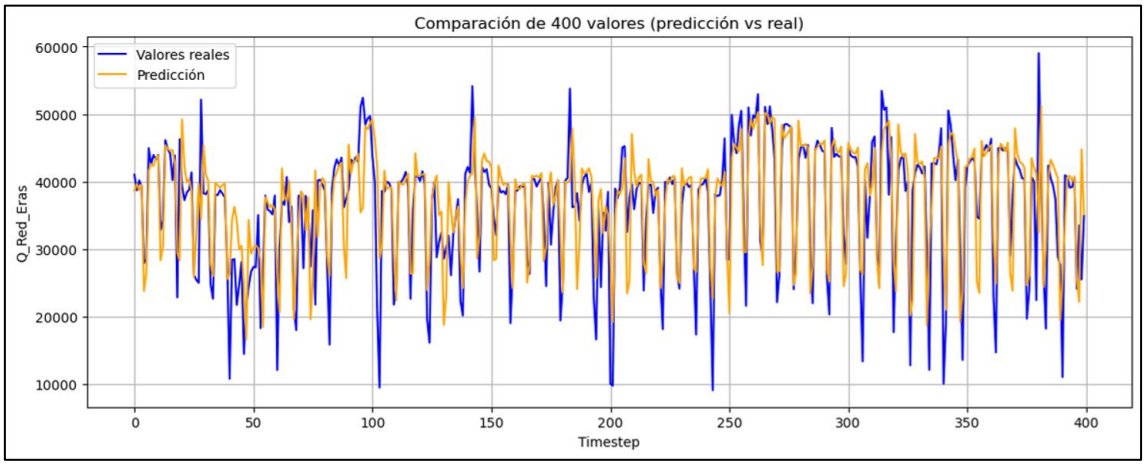

Figura 4.13: Comparación de los valores reales y predichos por el modelo 4.13

Resultados de simulación obtenidos:

RMSE train: 0.122

RMSE val: 0.122

RMSE test: 0.156

Media del error absoluto (MAE): 4550.084

Error en porcentaje (%): 11.65 %

RMSE: 7016.128

Tras analizar los resultados obtenidos en todas las simulaciones, se observa que el modelo no es capaz de seguir adecuadamente los picos de caudal, lo que se traduce en errores porcentuales elevados y predicciones poco fiables en momentos críticos. Ante esta limitación, se probaron nuevas técnicas de preprocesamiento de la serie temporal con el objetivo de mejorar el rendimiento del modelo. En concreto, se incorporaron dos funciones adicionales:

- Una función de media deslizante, con ventana de 7 días, destinada a suavizar la serie eliminando pequeñas oscilaciones sin alterar su estructura general.
- Una función de diferenciación, que calcula la diferencia entre valores consecutivos de la serie, permitiendo así resaltar los cambios bruscos y mejorar la detección de picos o tendencias.

A continuación se muestran los resultados obtenidos aplicando estas transformaciones, utilizando las combinaciones de parámetros que habían ofrecido mejores resultados en las simulaciones previas.

MODELO 1.14: 2 capas, 128 y 64 unidades, interpolación, optimizador *RMSprop*, 30 días. Empleando la función diferenciada.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.14)

Resultados de simulación obtenidos:

RMSE train: 0.088

RMSE val: 0.09

RMSE test: 0.114

Media del error absoluto (MAE): 4579.914

Error en porcentaje (%): 11.73 %

RMSE: 7223.7

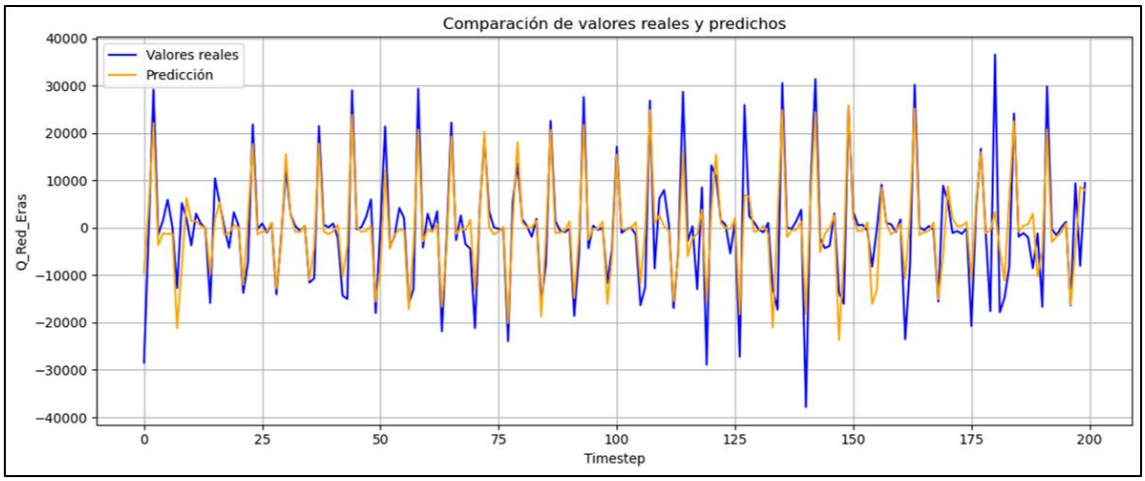

Figura 4.14: Comparación de los valores reales y predichos por el modelo 4.14

Se observa que este modelo 14 tiene una mejor predicción en los picos de caudal, aunque no se refleja en el error total producido, por ello no resulta aceptable su uso para la predicción.

MODELO 1.15: 2 capas, 128 y 64 unidades, interpolación, optimizador *RMSprop*, 30 días. Empleando la función media con ventana deslizante de 7 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.15)

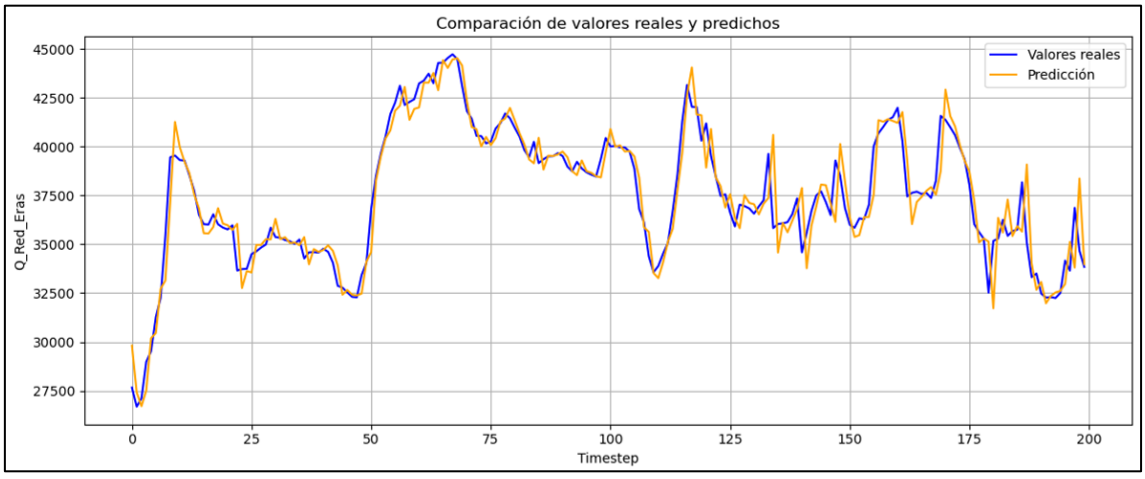

Figura 4.15: Comparación de los valores reales y predichos por el modelo 4.15

<u>Resultados de simulación obtenidos:</u>

RMSE train: 0.027

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 851.57

Error en porcentaje (%): 2.18 %

RMSE: 1246.246

En este caso, el modelo 1.15 muestra un rendimiento notablemente superior al observado en las simulaciones iniciales, con un error porcentual considerablemente más bajo. Esta mejora se debe en gran parte a la aplicación de la media deslizante, que permite suavizar la serie sin eliminar su estructura esencial, facilitando así que el modelo LSTM pueda aprender los patrones de forma más estable. A diferencia de los casos anteriores, el modelo es ahora capaz de adaptarse mucho mejor a los picos de caudal, lo que se traduce en una predicción más precisa tanto en los valores extremos como en la tendencia general.

Se puede observar la diferencia en el error obtenido en las Tablas 3 y 4, donde se aprecia que en los modelos que no emplean el filtrado de la función de la media deslizante se obtiene siempre un error similar y elevado, en cambio se obtiene un error mucho menor al aplicar dicho filtrado.

MODELO	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
Error en porcentaje (%)	12,79	12,61	13,17	12,68	12,49	12,47	11,63	12,07

Tabla 3: Comparación de los resultados con modelos univariados unistep

MODELO	1.9	1.10	1.11	1.12	1.13	1.14	1.15
Error en porcentaje (%)	12,14	14,61	12,14	12,07	11,65	11,73	2,18

Tabla 4: Comparación de los resultados con modelos univariados unistep

4.4.2. Predicción univariada-multistep

En este apartado se presentan los resultados obtenidos al aplicar un modelo LSTM en configuración univariada multistep, es decir, utilizando únicamente la serie histórica del caudal como entrada y generando como salida los valores previstos para los siete días siguientes. Para estas simulaciones se ha empleado la configuración que mejores resultados ofreció en el apartado anterior, la cual consiste en:

- Un modelo con dos capas LSTM (128 y 64 unidades respectivamente)
- Uso de interpolación para el tratamiento de valores nulos
- Optimizador RMSprop

- La función de la media con ventana deslizante de 7 días, se emplea esta técnica ya que mostró una mejora y buena adaptación a los picos de caudal.

A continuación, se muestran los resultados obtenidos para cada una de las siete salidas generadas por el modelo, correspondientes a las predicciones del caudal para los días 1 al 7 tras la ventana de entrada.

En la Figura 4.23, se muestra el error RMSE cometido en cada uno de los siete días predichos por el modelo, se observa una tendencia creciente del error a medida que se avanza en el horizonte de predicción. Este comportamiento es habitual en modelos multistep, ya que el nivel de incertidumbre aumenta cuanto más lejos se encuentra el punto de predicción respecto al último dato observado. En las primeras salidas el modelo dispone de información reciente que todavía conserva una fuerte correlación temporal, lo que permite mantener una alta precisión. Sin embargo, en las salidas correspondientes a los días más alejados el modelo debe extrapolar patrones sin apoyo en datos reales recientes, lo que dificulta la precisión y se traduce en mayores errores. Esta degradación del rendimiento con el horizonte de predicción es una limitación en los enfoques multistep, especialmente cuando se trabaja con una única variable de entrada, como en este caso.

MODELO 1.16: 2 capas, 128 y 64 unidades, interpolación, optimizador *RMSprop*, 30 días. Empleando la función media con ventana deslizante de 7 días.

Resultados de simulación general obtenidos:

RMSE train: 0.080

RMSE val: 0.063

RMSE test: 0.084

RMSE: 3311.212

A continuación se muestran los resultados y las gráficas de los 7 días de predicción.

DIA 7: (Figura 4.16)

MAE: 3404.572; RMSE: 4802.133

Error en porcentaje (%): 8.72 %

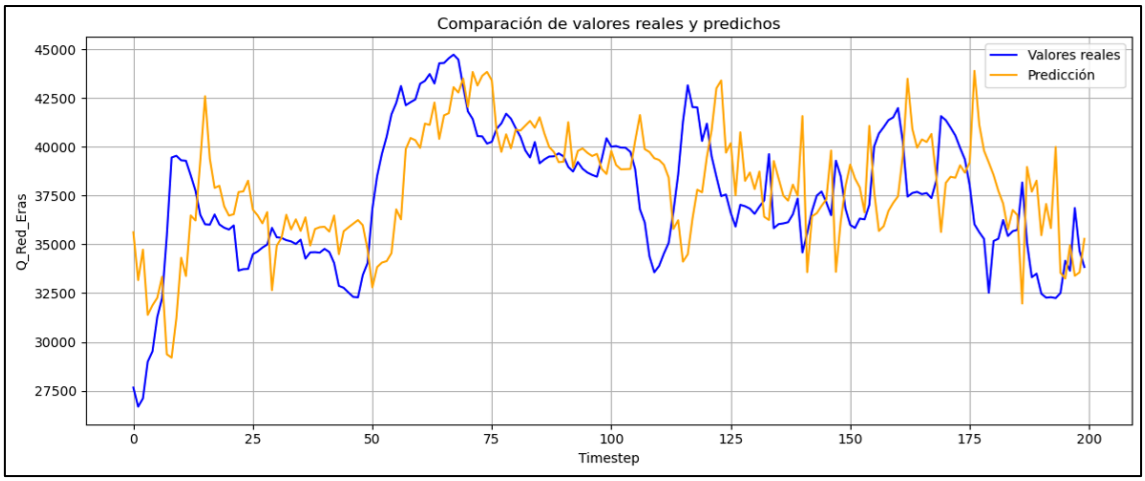

Figura 4.16: Comparación de los valores reales y predichos en el día 7

DIA 6: (Figura 4.17)

MAE: 3018.463; RMSE: 4245.575

Error en porcentaje (%): 7.73 %

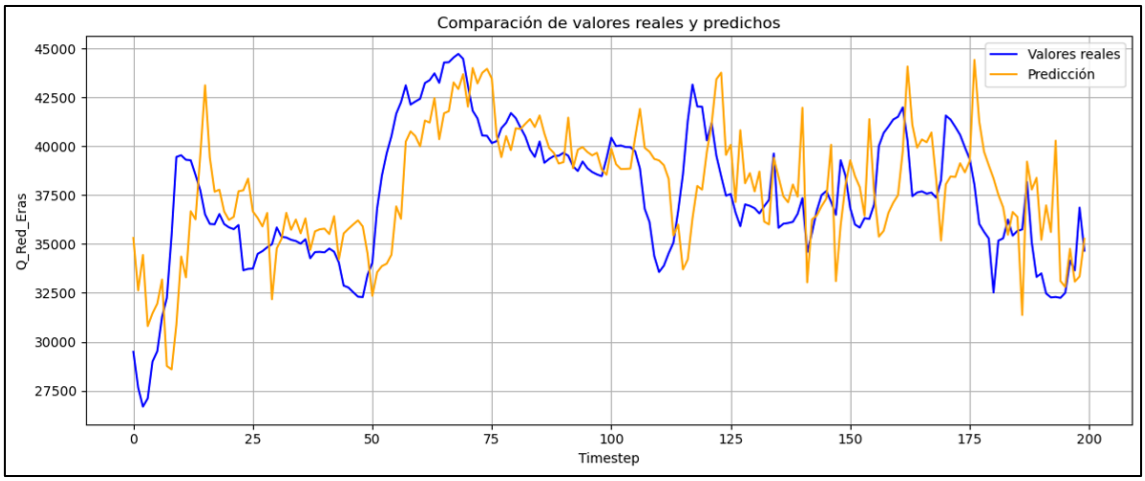

Figura 4.17: Comparación de los valores reales y predichos en el día 6

DIA 5: (Figura 4.18)

MAE: 2634.214; RMSE: 3672.049

Error en porcentaje (%): 6.75 %

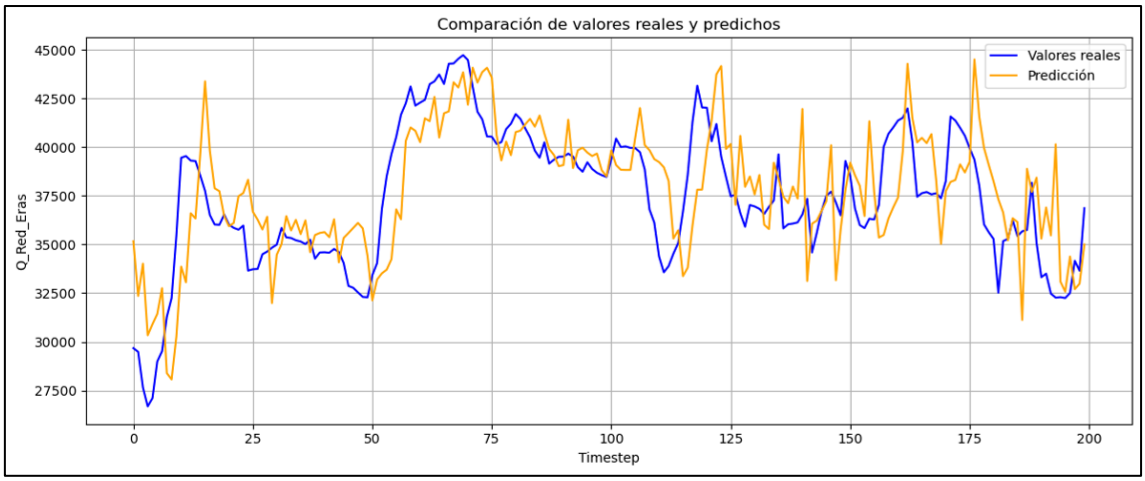

Figura 4.18: Comparación de los valores reales y predichos en el día 5

DIA 4: (Figura 4.19)

MAE: 2265.55; RMSE: 3122.377

Error en porcentaje (%): 5.8 %

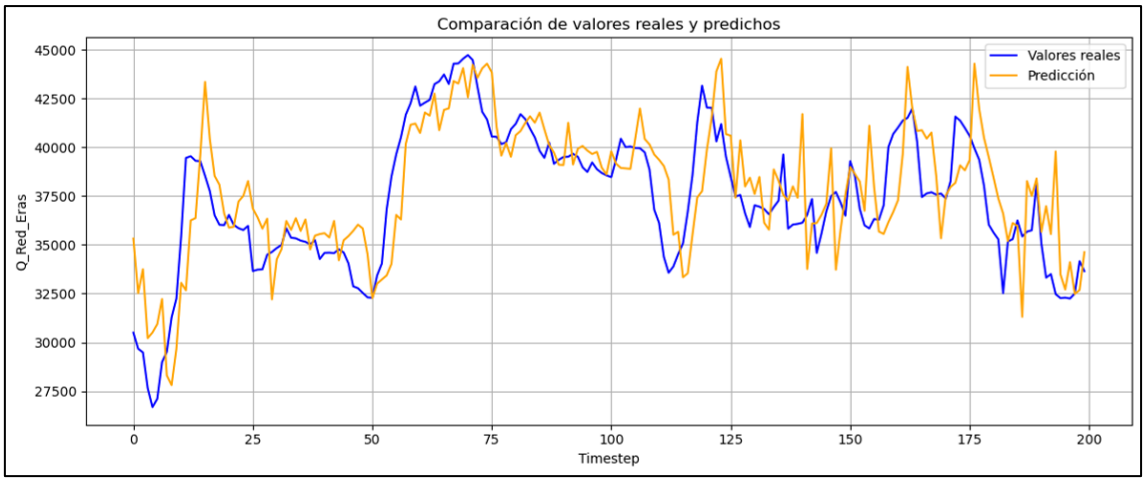

Figura 4.19: Comparación de los valores reales y predichos en el día 4

DIA 3: (Figura 4.20)

MAE: 1874.667; RMSE: 2549.404

Error en porcentaje (%): 4.8 %

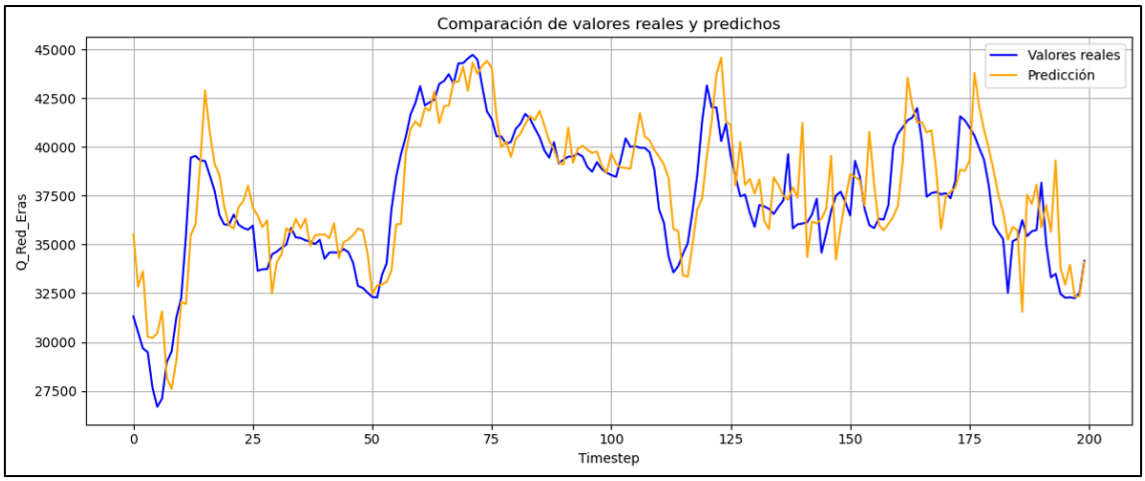

Figura 4.20: Comparación de los valores reales y predichos en el día 3

DIA 2: (Figura 4.21)

MAE: 1446.779; RMSE: 1985.463

Error en porcentaje (%): 3.71 %

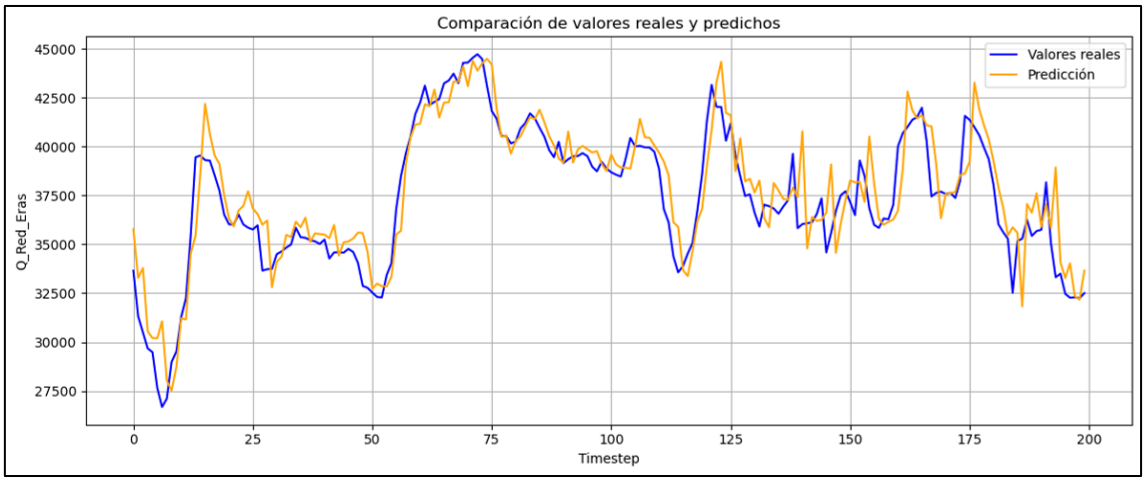

Figura 4.21: Comparación de los valores reales y predichos en el día 2

DIA 1: (Figura 4.22)

MAE: 1011.150; RMSE: 1410.241

Error en porcentaje (%): 2.59 %

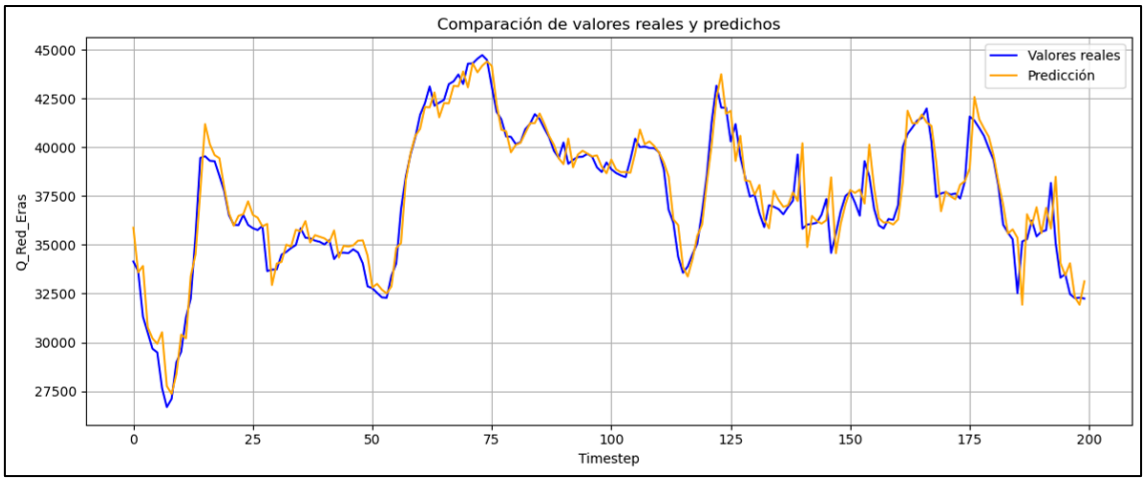

Figura 4.22: Comparación de los valores reales y predichos en el día 1

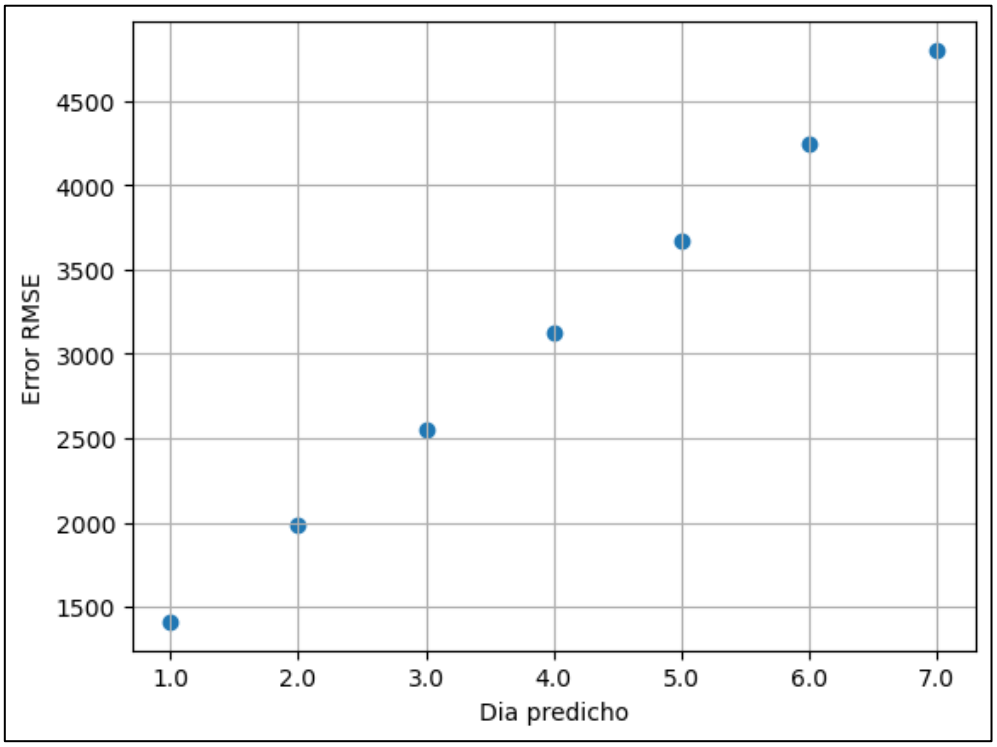

Figura 4.23: Evolución de los errores en los 7 días de salida con el Modelo 1.16

En la Tabla 5 se observa de forma resumida los errores obtenidos en cada una de las siete salidas del modelo. Al igual que la información otorgada por la Figura 4.23, se confirma la degradación progresiva del rendimiento a medida que aumenta el horizonte de predicción. Estos resultados reflejan la dificultad que presenta la predicción a largo plazo cuando se utiliza únicamente la serie del caudal como entrada, especialmente en contextos donde los patrones de demanda pueden variar de forma brusca. Sin embargo, el modelo mantiene un nivel de precisión aceptable en los dos primeros días de la predicción, lo que lo resulta útil para aplicaciones de planificación operativa a corto plazo.

	DIA 7	DIA 6	DIA 5	DIA 4	DIA 3	DIA 2	DIA 1
RMSE	4802.133	4245.575	3672.049	3122.377	2549.404	1985.463	1410.241
Error en porcentaje (%)	8.72 %	7.73 %	6.75 %	5.8 %	4.8 %	3.71 %	2.59 %

Tabla 5: Comparación de los resultados obtenidos por días

4.5. Predicción multivariada

Una vez analizados los resultados de los modelos univariados, se amplía la complejidad del modelo incluyendo variables externas que puedan influir en la demanda de agua. En concreto, se incorporan dos variables meteorológicas diarias: la temperatura media y la humedad relativa, obtenidas a partir de los registros históricos extraídos de Open-Meteo [38].

Este enfoque multivariado permite al modelo LSTM tener en cuenta no solo la evolución pasada del caudal, sino también factores externos que, como se ha visto en estudios previos, tienen un efecto directo sobre el consumo de agua, especialmente en determinadas épocas del año.

Al igual que en el caso univariado, se han planteado dos configuraciones principales de predicción:

- Predicción multivariada unistep: donde el modelo recibe como entrada el historial combinado de caudal, temperatura y humedad, y devuelve la predicción del caudal del día siguiente.
- Predicción multivariada multistep: en la que el modelo predice simultáneamente los valores de caudal para los siete días siguientes.

Todas las simulaciones multivariadas se han llevado a cabo empleando el preprocesamiento mediante media deslizante de 7 días. En lugar de reutilizar directamente la configuración óptima del modelo univariado, se ha vuelto a probar todas las combinaciones posibles de arquitectura, optimizador, tratamiento de valores nulos y longitud de la ventana de entrada, con el objetivo de comprobar si las tendencias observadas anteriormente se mantenían al introducir las variables externas. Este nuevo análisis comparativo permite evaluar el verdadero impacto de incorporar factores climáticos en la capacidad predictiva del modelo.

4.5.1. Predicción multivariada-unistep

A continuación se presentan los resultados obtenidos para la predicción multivariada unistep, donde el modelo LSTM recibe como entrada las series combinadas de caudal, temperatura media y humedad relativa, y genera como salida el caudal del día siguiente. Al igual que en el caso univariado, se han probado múltiples configuraciones del modelo con el fin de identificar la combinación que ofrece el mejor rendimiento en este nuevo contexto multivariable.

MODELO 2.1: 1 capa, 128 unidades, media, optimizador *Adam*, 7 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 7 días. (Figura 4.24)

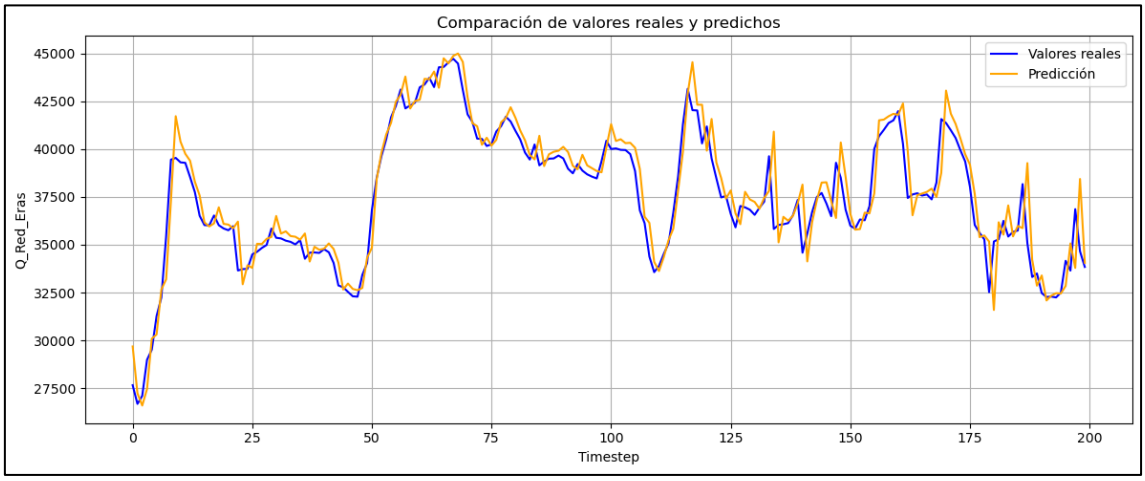

Figura 4.24: Comparación de los valores reales y predichos por el modelo 2.1

Resultados de simulación obtenidos:

RMSE train: 0.026

RMSE val: 0.026

RMSE test: 0.035

Media del error absoluto (MAE): 895.769

Error en porcentaje (%): 2.29 %

RMSE: 1324.075

MODELO 2.2: 1 capa, 128 unidades, media, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.25)

Resultados de simulación obtenidos:

RMSE train: 0.025

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 842.466

Error en porcentaje (%): 2.16 %

RMSE: 1257.66

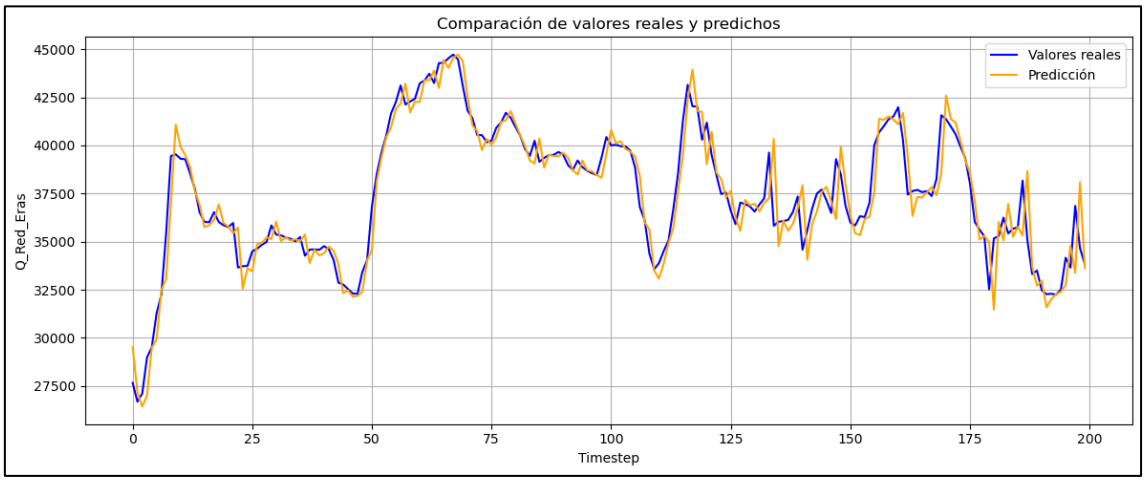

Figura 4.25: Comparación de los valores reales y predichos por el modelo 2.2

MODELO 2.3: 1 capa, 128 unidades, interpolación, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.26)

Resultados de simulación obtenidos:

RMSE train: 0.025

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 848.051

Error en porcentaje (%): 2.17 %

RMSE: 1239.754

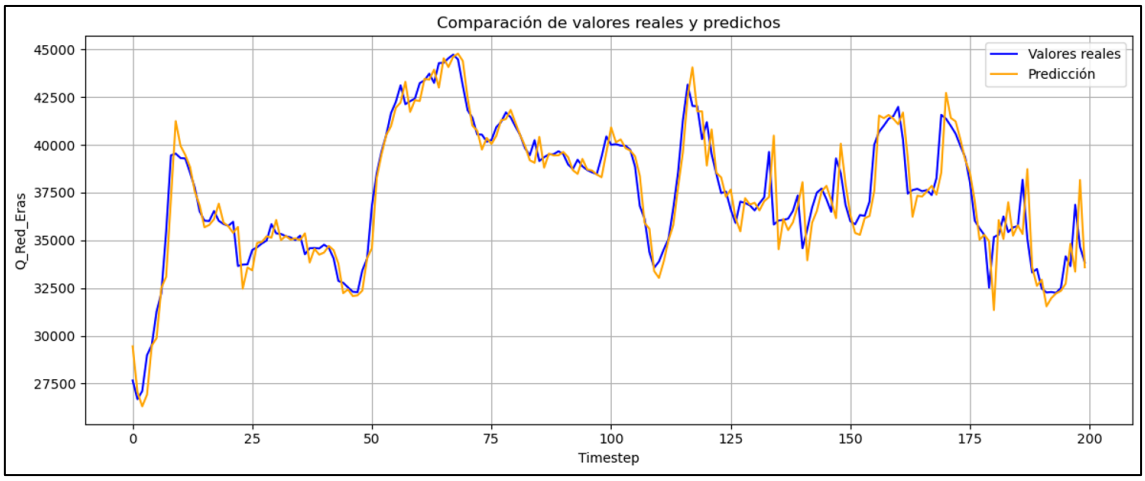

Figura 4.26: Comparación de los valores reales y predichos por el modelo 2.3

MODELO 2.4: 1 capa, 128 unidades, interpolación, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.27)

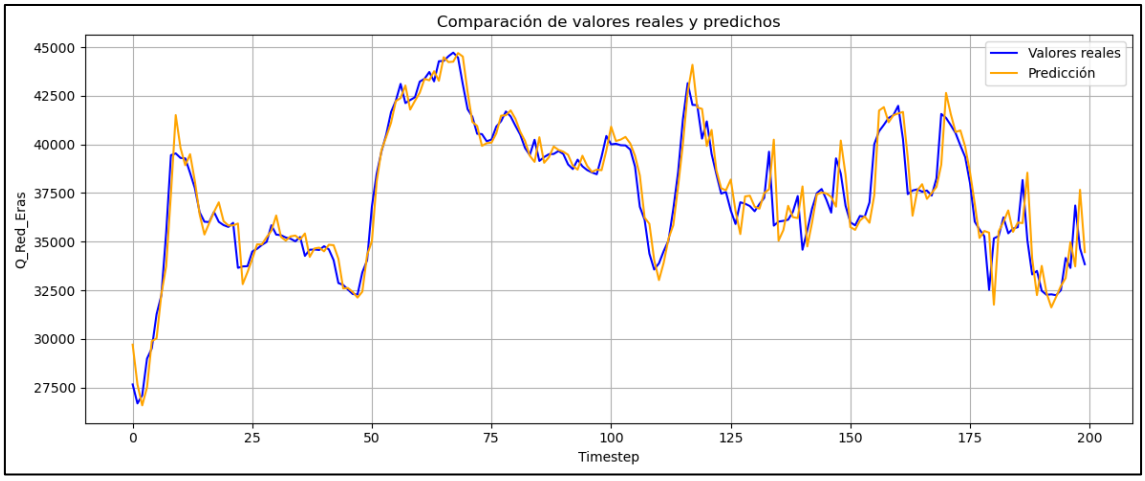

Figura 4.27: Comparación de los valores reales y predichos por el modelo 2.4

Resultados de simulación obtenidos:

RMSE train: 0.024

RMSE val: 0.024

RMSE test: 0.032

Media del error absoluto (MAE): 829.108

Error en porcentaje (%): 2.12 %

RMSE: 1188.695

MODELO 2.5: 1 capa, 128 unidades, media, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura simple de una sola capa LSTM con 128 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.28)

Resultados de simulación obtenidos:

RMSE train: 0.024

RMSE val: 0.024

RMSE test: 0.031

Media del error absoluto (MAE): 802.107

Error en porcentaje (%): 2.05 %

RMSE: 1173.607

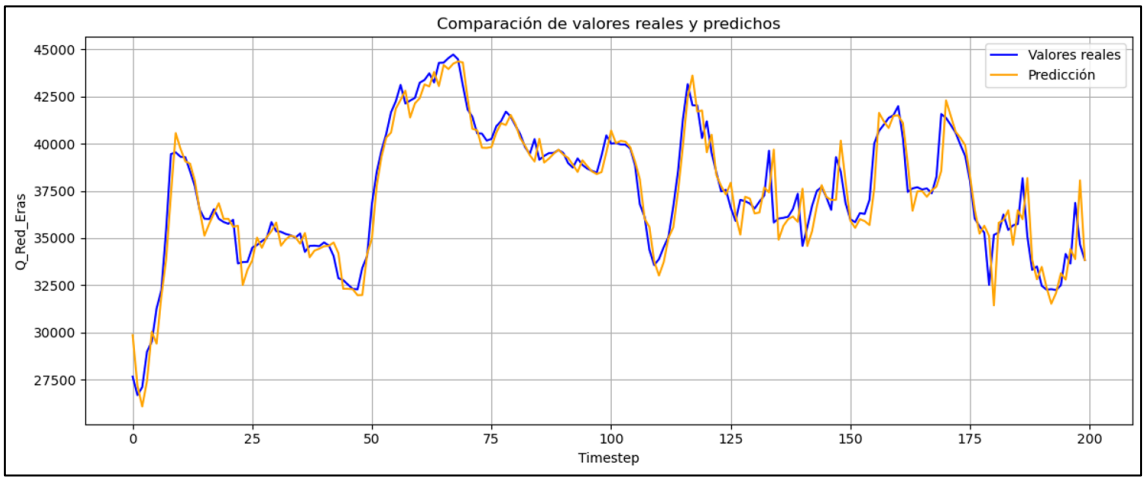

Figura 4.28: Comparación de los valores reales y predichos por el modelo 2.5

MODELO 2.6: 2 capas, 128 y 64 unidades, interpolación, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.29)

Resultados de simulación obtenidos:

RMSE train: 0.026

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 837.956

Error en porcentaje (%): 2.14 %

RMSE: 1240.932

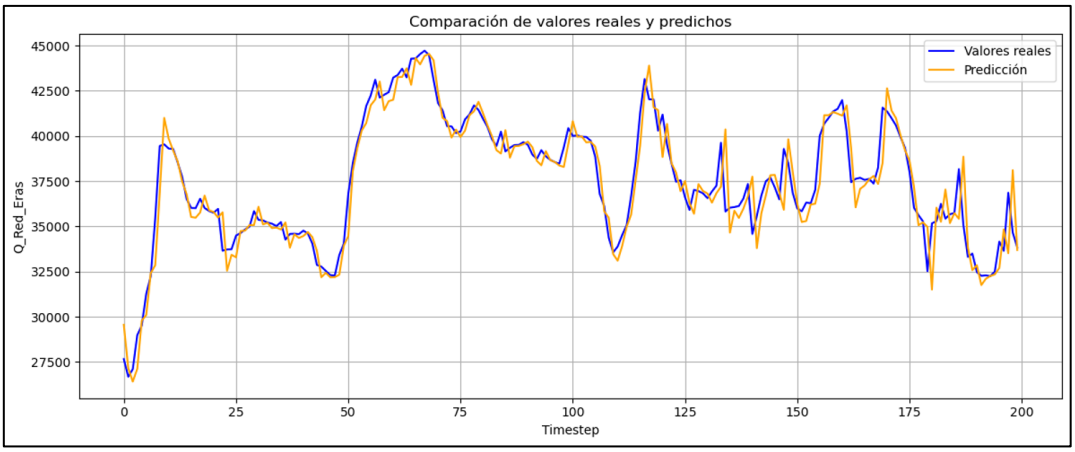

Figura 4.29: Comparación de los valores reales y predichos por el modelo 2.6

MODELO 2.7: 2 capas, 128 y 64 unidades, media, optimizador *Adam*, 7 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 7 días. (Figura 4.30)

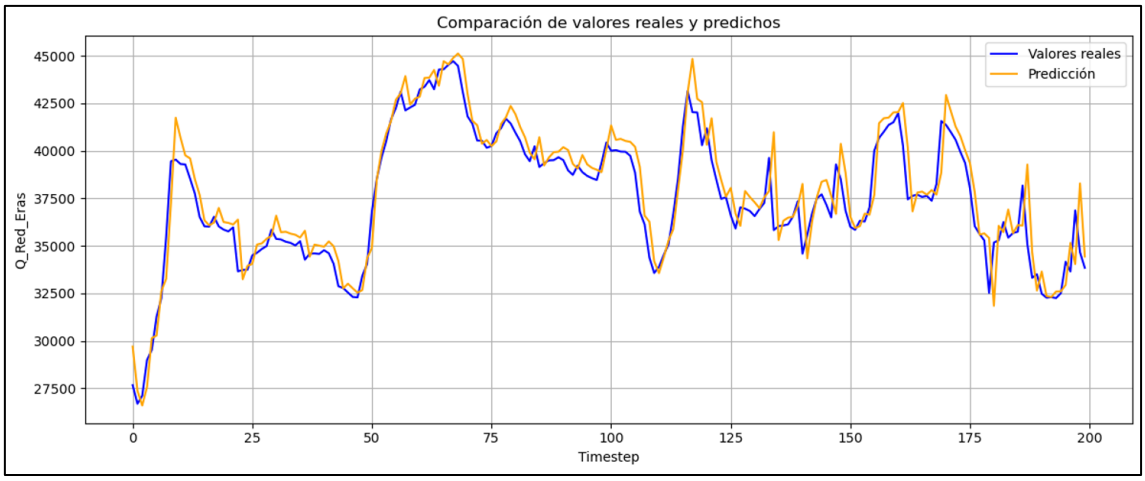

Figura 4.30: Comparación de los valores reales y predichos por el modelo 2.7

Resultados de simulación obtenidos:

RMSE train: 0.028

RMSE val: 0.027

RMSE test: 0.036

Media del error absoluto (MAE): 934.023

Error en porcentaje (%): 2.39 %

RMSE: 1347.786

MODELO 2.8: 2 capas, 128 y 64 unidades, media, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.31)

Resultados de simulación obtenidos:

RMSE train: 0.026

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 864.172

Error en porcentaje (%): 2.21 %

RMSE: 1220.101

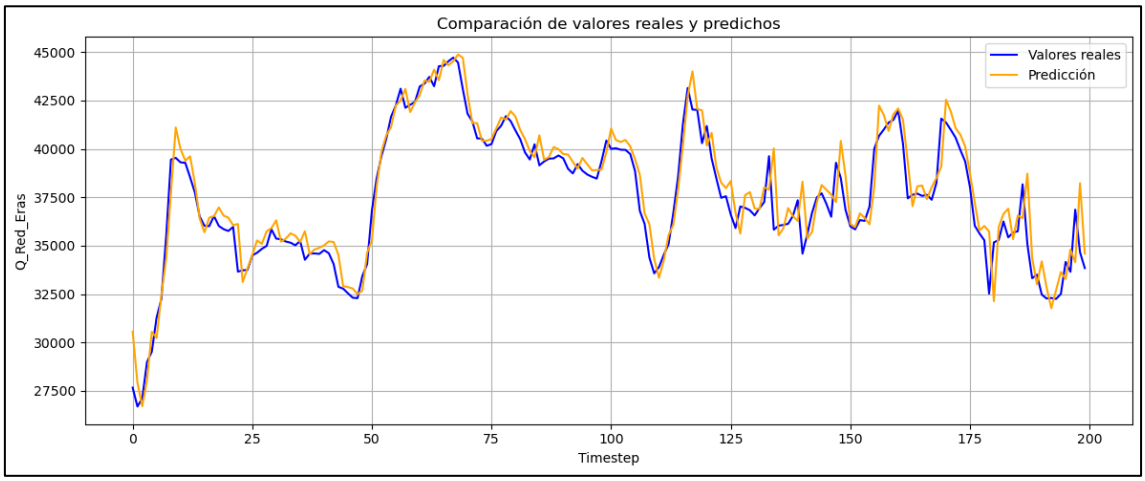

Figura 4.31: Comparación de los valores reales y predichos por el modelo 2.8

MODELO 2.9: 2 capas, 128 y 64 unidades, interpolación, optimizador *Adam*, 30 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por un valor obtenido por interpolación, se emplea el optimizador *Adam* y una ventana de entrada de 30 días. (Figura 4.32)

Resultados de simulación obtenidos:

RMSE train: 0.026

RMSE val: 0.024

RMSE test: 0.034

Media del error absoluto (MAE): 925.928

Error en porcentaje (%): 2.37 %

RMSE: 1287.101

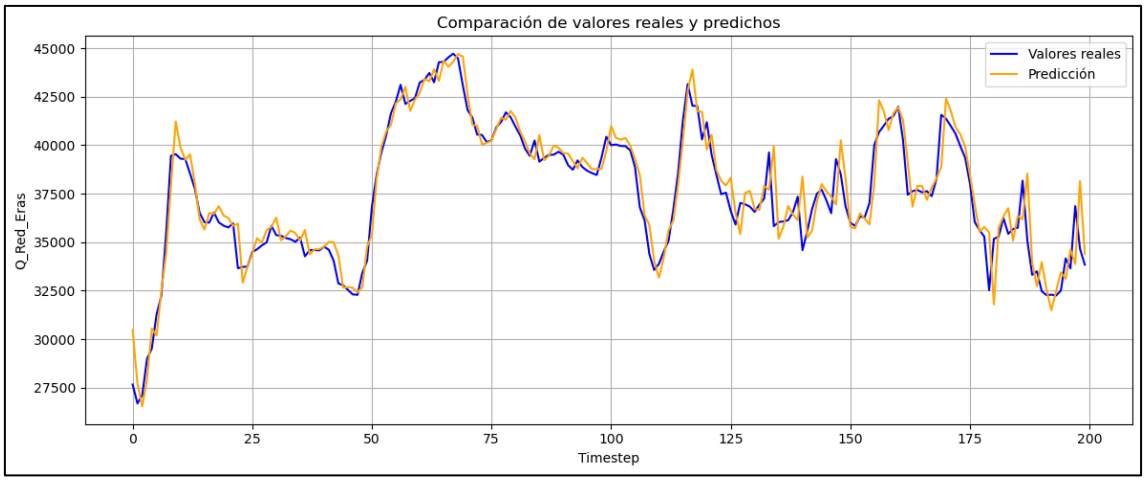

Figura 4.32: Comparación de los valores reales y predichos por el modelo 2.9

MODELO 2.10: 2 capas, 128 y 64 unidades, media, optimizador *RMSprop*, 30 días.

Este modelo emplea una arquitectura de dos capas LSTM con 128 y 64 unidades, se sustituyen los valores nulos por la media de la serie, se emplea el optimizador *RMSprop* y una ventana de entrada de 30 días. (Figura 4.33)

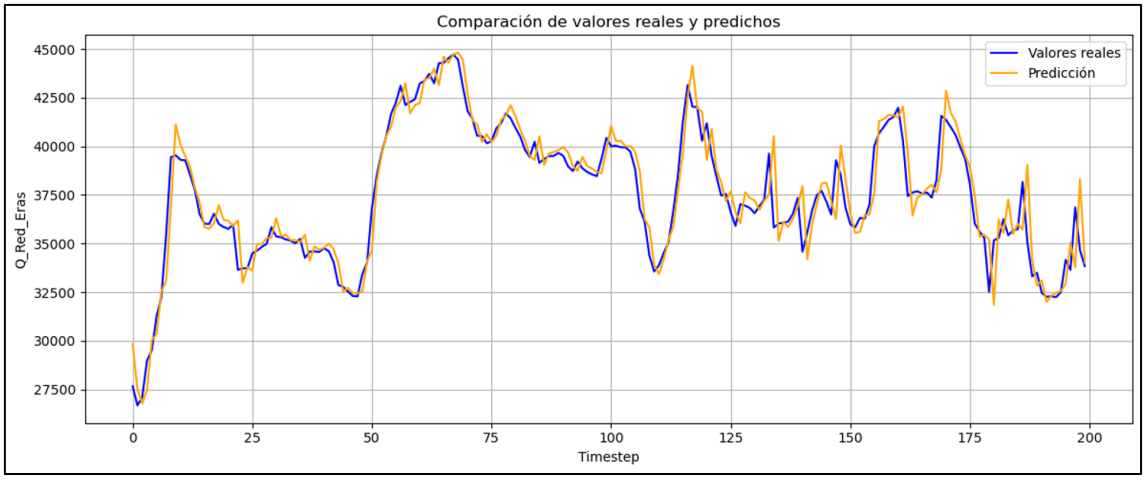

Figura 4.33: Comparación de los valores reales y predichos por el modelo 2.10

Resultados de simulación obtenidos:

RMSE train: 0.026

RMSE val: 0.025

RMSE test: 0.033

Media del error absoluto (MAE): 850.704

Error en porcentaje (%): 2.18 %

RMSE: 1270.75

Tras comparar todas las configuraciones probadas se observa que todas ellas consiguen una tendencia muy cercana a los valores reales. Aunque, se concluye que el mejor resultado para la predicción multivariada unistep ha sido el modelo 2.5, con una sola capa LSTM de 128 unidades, utilizando la función de media deslizante, el optimizador *Adam*, una ventana de entrada de 30 días, y reemplazo por la media en los valores nulos. Esta configuración logró el error porcentual más bajo (2.05 %) entre todas las simulaciones (Tabla 6).

MODELO	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.10
Error en porcentaje (%)	2,29	2,16	2,17	2,12	2,05	2,14	2,39	2,21	2,37	2,18

Tabla 6: Comparación de los resultados con modelos multivariados unistep

Se observa que el uso de dos capas LSTM no mejora el rendimiento, e incluso en algunos casos lo empeora. Una posible explicación es que el modelo más profundo tiende a sobreajustar los datos cuando el tamaño del conjunto de entrenamiento no es suficientemente grande. Además, el uso de múltiples capas puede hacer que la red pierda sensibilidad a los patrones más inmediatos, lo que en una predicción a corto plazo como el unistep puede resultar contraproducente.

4.5.2. Predicción multivariada-multistep

En este apartado se analiza el comportamiento del modelo en una configuración multivariada multistep, es decir, utilizando como entrada las series temporales de caudal, temperatura media y humedad relativa, y generando como salida los valores de caudal correspondientes a los siete días siguientes. Ya que en el apartado anterior se identificó una combinación de parámetros con los mejores resultados en la predicción a un día, para este nuevo modelo se ha optado por mantener dicha configuración para evaluar su rendimiento en un horizonte de predicción más amplio.

A continuación, se muestran los resultados obtenidos para algunas de las siete salidas generadas por el modelo. En la Figura 4.34, se muestra el error RMSE cometido en cada uno de los siete días predichos por el modelo, se observa una tendencia creciente del error a medida que se avanza en el horizonte de predicción.

MODELO 2.11: 1 capa, 128 unidades, media, optimizador *Adam*, 30 días. Empleando la función media con ventana deslizante de 7 días.

Resultados de simulación general obtenidos:

RMSE train: 0.057

RMSE val: 0.072

RMSE test: 0.087

RMSE: 3394.527

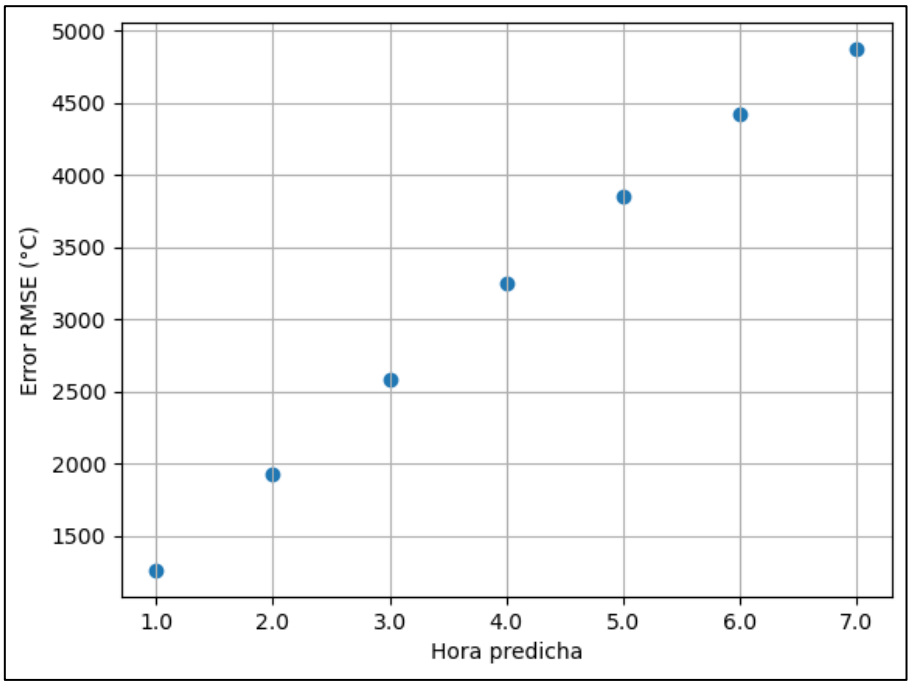

Figura 4.34: Evolución de los errores en los 7 días de salida con el Modelo 2.11

A continuación se muestran los resultados del primer, segundo y último día de predicción.

DIA 1: (Figura 4.35)

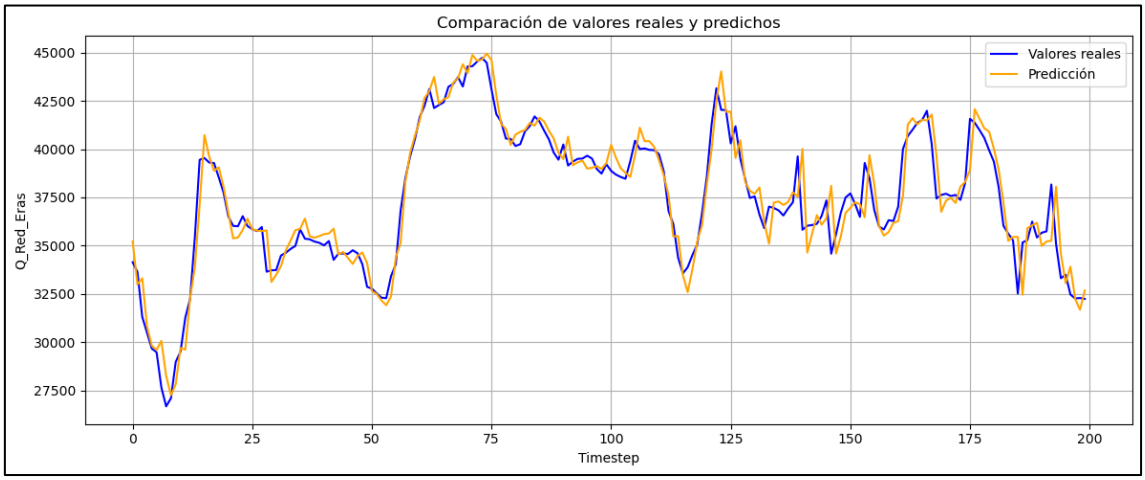

Figura 4.35: Comparación de los valores reales y predichos en el día 1

MAE: 871.026; RMSE: 1259.455

Error en porcentaje (%): 2.23 %

DIA 2: (Figura 4.36)

MAE: 1308.056; RMSE: 1896.682

Error en porcentaje (%): 3.35 %

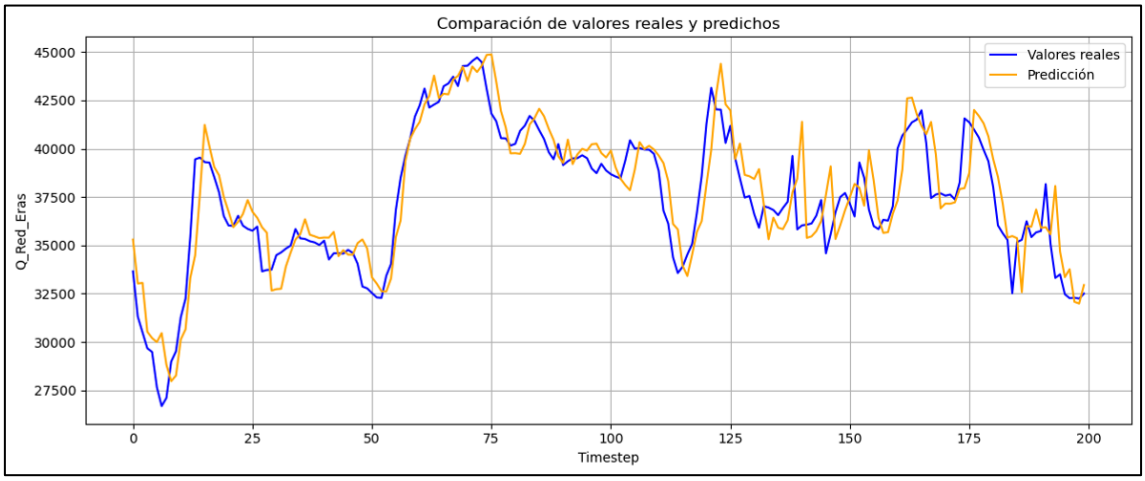

Figura 4.36: Comparación de los valores reales y predichos en el día 2

DIA 7: (Figura 4.37)

MAE: 3318.27; RMSE: 4811.496

Error en porcentaje (%): 8.5 %

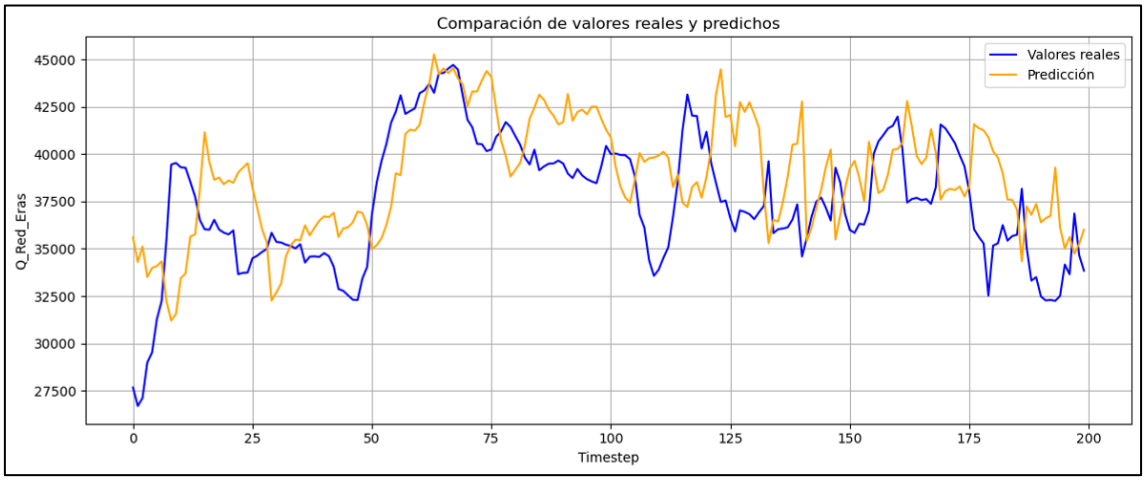

Figura 4.37: Comparación de los valores reales y predichos en el día 7

Se confirma, al igual que en el caso del modelo univariado, la degradación progresiva del rendimiento a medida que aumenta el horizonte de predicción. Los resultados obtenidos con la configuración multivariada multistep son ligeramente peores que los alcanzados en el caso multivariado unistep. Aunque ambos modelos utilizan la misma arquitectura y el mismo preprocesamiento, se observa un incremento en el error porcentual. Esta diferencia puede deberse al mayor riesgo de sobreajuste (overfitting) en la predicción multistep. El modelo intenta aprender patrones válidos para predecir varios días a la vez, pero al hacerlo puede ajustarse demasiado a los datos de entrenamiento, perdiendo capacidad de generalización.

4.6. Resumen de resultados

A lo largo de este capítulo se han presentado los resultados obtenidos con distintas configuraciones del modelo de red LSTM, evaluando su rendimiento en las tareas de predicción de la demanda de agua en la ciudad de Valladolid. El análisis ha constado tanto de modelos univariados como multivariados, así como configuraciones de unistep y multistep, permitiendo comparar cada enfoque sobre la precisión final del modelo.

Primero se comenzó con la predicción univariada unistep utilizando únicamente la serie de caudal y sin aplicar un filtrado adicional a los datos. Aunque el modelo era capaz de reproducir de forma aproximada la tendencia general de la demanda, los resultados reflejaban un error porcentual elevado, especialmente en los periodos donde se producían picos o caídas bruscas del caudal. Estos resultados mostraban que la red no era capaz de ajustarse adecuadamente a la variabilidad de la serie.

Como solución, se implementaron dos funciones de transformación adicionales: una basada en aplicar media deslizante de 7 días y otra de diferenciación entre valores consecutivos. Tras realizar nuevas simulaciones con dichas funciones, se comprobó que la aplicación de la media deslizante mejoró de forma significativa los resultados, suavizando la serie sin eliminar su estructura, facilitando el aprendizaje del modelo y obteniendo un error mucho más bajo que los obtenidos anteriormente. Esta transformación fue empleada en todas las simulaciones posteriores.

A continuación, se llevó a cabo la predicción univariada multistep, donde el modelo debía anticipar los valores del caudal para los siete días siguientes. Como era de esperar, se observó que el error aumentaba progresivamente con el horizonte de predicción, siendo el séptimo día el que presentaba mayor error. Esta disminución progresiva del rendimiento es común en modelos multistep, ya que la incertidumbre se acumula y el modelo dispone de menos contexto reciente para generar predicciones fiables. Como consecuencia, aunque el modelo obtenía buenos resultados para el primer día, el resultado global era inferior al obtenido en la versión unistep.

El siguiente paso consistió en incorporar variables externas, la temperatura media diaria y la humedad relativa, con el objetivo de observar su impacto en la precisión del modelo. En esta configuración, se observó una mejora respecto a los enfoques univariados, obteniendo los mejores resultados de todo el estudio. La introducción de variables climáticas permitió al modelo adaptarse mejor a ciertos patrones estacionales o meteorológicos que influyen directamente en la demanda de agua, especialmente en épocas más cálidas o secas.

En el caso de la predicción multivariada multistep los resultados también mejoraron respecto al modelo univariado equivalente, gracias a las variables externas. Sin embargo, no se logró superar el rendimiento obtenido en la versión multivariada unistep. Esto puede deberse a la mayor dificultad que supone predecir varios días a la vez, ya que el modelo tiene que anticipar el comportamiento futuro sin conocer información real de esos días. Además, al intentar ajustarse a tantas salidas al mismo tiempo, es posible que el modelo haya terminado aprendiendo demasiado los datos de entrenamiento, perdiendo así eficacia cuando se enfrenta a datos nuevos. Esta situación de sobreajuste hace que aunque el modelo parezca funcionar bien durante el entrenamiento, no se comporte igual de bien al hacer predicciones reales, lo que afecta negativamente a su precisión final.

En resumen, el mejor rendimiento se obtuvo con el modelo 2.5 multivariado unistep, configurado con las siguientes características:

- Una sola capa LSTM con 128 unidades
- Uso de la función media deslizante de 7 días
- Optimización mediante *Adam*
- Ventana de entrada de 30 días
- Sustitución de valores nulos por la media de la serie de caudal

Esta combinación logró minimizar el error y capturar con mayor precisión tanto la tendencia general como los picos de caudal, lo que la convierte en la configuración más robusta y eficaz para el problema planteado.

CAPÍTULO V:
CONCLUSIONES

5. Conclusiones

Este Trabajo Fin de Grado se ha centrado en el desarrollo de un modelo capaz de predecir la demanda en el caudal de agua potable para la ciudad de Valladolid, aplicando redes neuronales de tipo LSTM. En este proceso se han llevado a cabo varios tipos de trabajo, desde la recopilación y preparación de los datos hasta el diseño, entrenamiento y evaluación de diferentes configuraciones en los modelos.

En una primera fase, se trabajó con modelos univariados y predicción a un solo día, los cuales resultaron útiles para captar la tendencia general, pero con limitaciones en la predicción de los picos de demanda. Ante estos resultados, se introdujeron funciones de filtrado de la serie, como la media deslizante de siete días, que ayudó a suavizar las fluctuaciones y permitió al modelo identificar patrones más estables sin eliminar los cambios de comportamiento. Esta mejora en el preprocesamiento permitió reducir de forma notable el error, haciendo que la predicción estuviese más ajustada a los valores reales.

Más adelante, se avanzó hacia modelos multivariados, incorporando información climática diaria (temperatura media y humedad relativa), con el objetivo de proporcionar al modelo información adicional. Esta ampliación resultó beneficiosa ya que en las configuraciones multivariadas unistep se alcanzaron los mejores resultados del proyecto, con errores bajos y una buena capacidad de adaptación a los picos de la demanda. La comparación entre modelos reveló que la presencia de variables externas mejoraba claramente el rendimiento frente a los enfoques univariados.

En los modelos multistep, diseñados para predecir simultáneamente varios días futuros, se observó un aumento progresivo del error conforme se ampliaba el horizonte de predicción. Esto es un comportamiento habitual, ya que el modelo dispone de menos información real y debe anticipar patrones con mayor incertidumbre. Aunque los resultados del modelo multivariado multistep fueron mejores que los obtenidos en su versión univariada, no llegaron a superar la precisión alcanzada por el modelo multivariado unistep. Esto puede deberse a que, al intentar predecir varios días a la vez, el modelo tiende a sobreajustarse o a perder sensibilidad en los primeros días, lo que afecta a la precisión global de las salidas. Aun así, la incorporación de variables externas demostró ser útil para obtener un mejor rendimiento.

A modo de resumen, se puede afirmar que la arquitectura LSTM ha demostrado ser adecuada para este tipo de problemas, siempre que se acompañe de un buen tratamiento de datos y la incorporación de variables externas para aportar más información al modelo. El enfoque multivariado ofrece predicciones más precisas, especialmente a corto plazo, y representa una herramienta útil para tareas de planificación y la toma de decisiones relacionadas con la gestión del agua.

En cuanto a futuras líneas de trabajo, se pueden plantear varias propuestas:

- Introducir nuevas variables externas, como los días festivos, o datos relativos al consumo por sectores.
- Explorar otras arquitecturas como GRU, redes bidireccionales o modelos híbridos.
- Integrar el modelo en un sistema que reciba datos en tiempo real, permitiendo una actualización continua de las predicciones.

Con todo ello, este trabajo sienta una base para seguir explorando la aplicación de las redes neuronales en el ámbito de la predicción de la demanda de agua, con posibilidades reales de mejora y futura implementación.

6. Bibliografía

[1] **Instituto Nacional de Estadística (INE)**
https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176834&menu=ultiDatos&idp=1254735976602

[2] **Aquavall**
https://aquavall.es/

[3] **Zhou, S. L., McMahon, T. A., Walton, A., & Lewis, J. (2002)**. *"Forecasting daily urban water demand: a case study of Melbourne"*. **Journal of Water Resources Planning and Management**, 128(1), 26-36.

[4] **Alvisi, S., Franchini, M., & Marinelli, A. (2007). "A short-term, pattern-based model for water-demand forecasting". Journal of Hydroinformatics, 9(1), 39-50.**

[5] **García, V. J., García-Bartual, R., & Cabrera, E. (2015)**. *"Water Demand Forecasting in Small Communities"*. **Water Resources Management**, 29(9), 3215-3231.

[6] **Sustainable Sanitation and Water Management**
https://sswm.info/es/gass-perspective-es/tecnologias-de-agua-y-saneamiento/tecnologias-de-abastecimiento-de-agua/filtraci%C3%B3n-r%C3%A1pida-de-arena

[7] **Mathiesen: Carbón Activado**
https://www.grupomathiesen.com/application/carbon-activado-adsorbente-eficaz-en-tratamiento-de-aguas/

[8] **AEAS (Asociación Española de Abastecimientos de Agua y Saneamiento) Control de cloratos en el tratamiento**
https://www.daquas.es/images/explora/TripticoA3_RECOMENDACIONES_CONTROL_DE_CLORATOS.pdf

[9] **Morote Seguido, Á. F. (2017).** Factores que inciden en el consumo de agua doméstico. Estudio a partir de un análisis bibliométrico. *Estudios Geográficos.* Vol. 78(282), pp: 257-281.
https://doi.org/10.3989/estgeogr.201709

[10] **AEMET (2015).** *Proyecciones Climáticas para el siglo XXI en España*.
http://www.aemet.es/es/serviciosclimaticos/cambio_climat

[11] Instituto Nacional de Estadística (INE) Encuesta ocupación hotelera https://www.ine.es/jaxiT3/Tabla.htm?t=2074

[12] EpData (2025) https://www.epdata.es/datos/ocupacion-hotelera-hoteles-datos-graficos-municipios/143/valladolid/7617

[13] Ministerio de Transportes, Movilidad y Agenda Urbana (2023). Proyectos de digitalización del ciclo urbano del agua.

[14] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts. https://otexts.com/fpp2/

[15] Fierro (2020) Predicción de Series Temporales con Redes Neuronales, TFG, Universidad Nacional de la Plata, Argentina, 2020 https://sedici.unlp.edu.ar/bitstream/handle/10915/114857/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

[16] Box, G. E. P., & Jenkins, G. M. (1976). Time Series Analysis: Forecasting and Control. Holden-Day.

[17] Agaj, T., Budka, A., Janicka, E. *et al.* Using ARIMA and ETS models for forecasting water level changes for sustainable environmental management. *Sci Rep* 14, 22444 (2024). https://doi.org/10.1038/s41598-024-73405-9

[18] Koutsoyiannis, D. (2000). *A generalized mathematical framework for stochastic simulation and forecast of hydrologic time series*. *Water Resources Research*, 36(6), 1519–1533. https://doi.org/10.1029/2000WR900044

[19] Chatfield, C. (2004). *The Analysis of Time Series: An Introduction*

[20] Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). *LSTM network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems*, 11(2), 68–75. https://doi.org/10.1049/iet-its.2016.0208

[21] González Velázquez, Miguel (2020). Mejora de la calidad de un proceso mediante la detección de anomalías basada en datos. TFG, Universidad de Valladolid.

[22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning. MIT Press. Capítulos 1, 6 y 10.* https://www.deeplearningbook.org/

[23] INNOVATIANA: Función de activación https://es.innovatiana.com/post/activation-function-in-ai#:~:text=Las%20funciones%20de%20activaci%C3%B3n%20determinan,filtran%20las%20se%C3%B1ales%20menos%20relevantes [Último acceso: Julio 2025].

[24] Codificando Bits: La función de activación https://codificandobits.com/blog/funcion-de-activacion/ [Último acceso: Julio 2025].

[25] Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Pearson Education. https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf

[26] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. Vol 270(2), pp:654-669.

[27] LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature 521, 436-444 (2015) https://www.researchgate.net/publication/277411157_Deep_Learning

[28] W. Feng, N. Guan, Y. Li, X. Zhang and Z. Luo, "Audio visual speech recognition with multimodal recurrent neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 681-688, doi: 10.1109/IJCNN.2017.7965918.

[29] Gamboa, J. C. B. (2017). Deep learning for time-series análisis https://arxiv.org/pdf/1701.01887

[30] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* Vol 5(2), pp: 157-166. https://www.researchgate.net/publication/5583935_Learning_long-term_dependencies_with_gradient_descent_is_difficult

[31] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-Term Memory. Neural Computation Vol.9 (8), pp: 1735-1780. https://www.researchgate.net/publication/13853244_Long_Short-Term_Memory

[32] Xu, Junhua. (2024). Forecasting Water Demand With the Long Short-Term Memory Deep Learning Mode. International Journal of Information Technologies and Systems Approach, Vol. 17(1), pp:1-18 https://www.researchgate.net/publication/378317910_Forecasting_Water_Demand_With_the_Long_Short-Term_Memory_Deep_Learning_Mode

[33] Oliver Faust, Yuki Hagiwara, Tan Jen Hong, Oh Shu Lih, U Rajendra Acharya, (2018), Deep learning for healthcare applications based on physiological signals: A review, Computer Methods and Programs in Biomedicine, Vol: 161, pp:1-13 https://shura.shu.ac.uk/21073/1/Deep%20learning%20for%20healthcare%20applications%20based%20on%20physiological%20signals%20a%20review.pdf

[34] Chollet, F. (2021). *Deep learning with Python* (2.ª ed.). Manning Publications.

[35] Schuster, M., & Paliwal, K. K. (1997). *Bidirectional recurrent neural networks*. IEEE Transactions on Signal Processing, 45(11), 2673–2681. https://www.researchgate.net/publication/3316656_Bidirectional_recurrent_neural_networks

[36] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). *Convolutional LSTM network: A machine learning approach for precipitation nowcasting*. In *Advances in Neural Information Processing Systems*, 28. https://papers.nips.cc/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html

[37] Vaswani, A., et al. (2017). *Attention is all you need*. *Advances in Neural Information Processing Systems*, 30, 5998–6008

[38] *API histórica de Open-Meteo* https://open-meteo.com/en/docs/historical-weather-api [Último acceso: Julio 2025].