



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Graduado en Ingeniería Informática

Aplicación de Modelos Predictivos basados en Inteligencia Artificial para la Gestión Sostenible del Agua en Embalses de Andalucía

Application of an Artificial Intelligence-based Predictive Models for Sustainable Water Management of Andalusian Reservoirs

Modelos Predictivos para la Prospección de Escenarios Dinámicos en la Gestión Hídrica

Predictive Models for Prospecting Dynamic Scenarios in Water Management

Realizado por
Patricia Martín Castillo

Tutorizado por
Jamal Toutouh El Alamin
Christian Cintrano López

Departamento
LENGUAJE Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA

MÁLAGA, julio 2025



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA

**Aplicación de Modelos Predictivos basados en Inteligencia Artificial
para la Gestión Sostenible del
Agua en Embalses de Andalucía**

**Application of an Artificial Intelligence-based Predictive Models for
Sustainable Water Management of Andalusian Reservoirs**

**Modelos Predictivos para la Prospección de Escenarios Dinámicos
en la Gestión Hídrica**

**Predictive Models for Prospecting Dynamic Scenarios in Water
Management**

Realizado por
Patricia Martín Castillo

Tutorizado por
Jamal Toutouh El Alamin
Christian Cintrano López

Departamento
LENGUAJE Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JULIO DE 2025

Fecha de defensa: septiembre de 2025

Abstract

This Final Degree Project delivers a practical solution to support reservoir management in Andalusia through forecasting and scenario exploration. The goal is to anticipate storage evolution and provide fast, accessible insights for practitioners and non-expert users.

To this end, a data pipeline is built that integrates historical reservoir data with meteorological variables, resolving naming inconsistencies, mapping reservoirs to provinces, and generating temporal aggregates and derived features. The information is consolidated in the InfluxDB time-series database and used to train deep learning models (GRU) capable of capturing seasonality and lagged effects. The final phase uses these predictions to generate What-If scenarios (e.g., changes in precipitation or temperature).

The result is a system that combines time-tested data, inference-ready models, and a web interface for viewing historical series and their evolution.

Keywords: water management; reservoirs; Andalusia; time series; GRU; What-If; InfluxDB; deep learning.

Resumen

Este Trabajo Final de Grado plantea una solución práctica para apoyar la gestión de embalses en Andalucía mediante predicción y exploración de escenarios. El trabajo se centra en anticipar la evolución de las reservas y facilitar consultas ágiles para técnicos y usuarios no expertos.

Para ello se construye un *pipeline* de datos que integra datos históricos de embalses con variables meteorológicas, resolviendo inconsistencias de nombres, mapeando embalses a provincia y generando agregados temporales y variables derivadas. La información se consolida en la base de datos de series temporales InfluxDB y se emplea para entrenar modelos de aprendizaje profundo (GRU) capaces de capturar estacionalidad y efectos retardados. La última fase usa estas predicciones para generar escenarios *What-If* (p. ej., cambios en precipitación o temperatura).

El resultado es un sistema que combina: datos depurados temporales, modelos listos para inferencia y una interfaz web para visualizar series históricas y su evolución.

Palabras clave: gestión del agua; embalses; Andalucía; series temporales; GRU; *What-If*; InfluxDB; aprendizaje profundo.

Índice

1. Introducción	7
1.1. Motivación	7
1.2. Objetivos	8
1.3. Estructura del documento	9
2. Estado del arte	13
2.1. Modelos computacionales aplicados al análisis y gestión del agua	13
2.2. Modelos analizados en este trabajo	15
2.3. Contexto del estudio	22
2.3.1. Red de embalses	23
2.3.2. Red de estaciones agroclimáticas	24
2.4. Métricas empleadas en este Trabajo Final de Grado	27
2.5. Tecnologías empleadas	28
2.5.1. Python	28
2.5.2. InfluxDB (Influx DataBase)	28
2.5.3. JavaScript	29
2.5.4. Visual Studio Code	30
2.5.5. Google Colaboratory	30
2.5.6. Git + Github	30
2.5.7. Docker	31
2.5.8. Node.js	31
2.5.9. Bootstrap	32
2.5.10. Express	33
2.5.11. HTML (HyperText Markup Language)	33
2.5.12. CSS (Cascade Style Sheets)	34
2.5.13. Algunas librerías destacables	34
3. Análisis experimental	37

3.1.	Entrenamiento de modelos predictivos	38
3.1.1.	Conjuntos de datos y pre-procesado	38
3.1.2.	<i>Feature selection</i> (selección de características)	40
3.1.3.	Pasos del entrenamiento de los modelos	49
3.2.	Entrenamiento y validación de escenarios <i>What-If</i>	51
3.2.1.	Pasos del desarrollo de los escenarios <i>What-If</i>	52
3.3.	Resultados	53
3.3.1.	Resultados del entrenamiento de los modelos	54
3.3.2.	Resultados de escenarios <i>What-If</i>	80
3.4.	Discusión general	94
4.	Aplicación	97
4.1.	Ingeniería del software	97
4.1.1.	Introducción a la plataforma web	97
4.1.2.	Estructura del proyecto	98
4.1.3.	Despliegue de servicios	99
4.1.4.	Funcionamiento de la aplicación web	100
4.1.5.	Problemas encontrados y soluciones	101
4.2.	Procesos ETL	102
4.2.1.	Fase de extracción	102
4.2.2.	Fase de transformación	106
4.2.3.	Fase de carga	108
4.2.4.	Problemas encontrados y soluciones	110
4.3.	Metodología de trabajo	110
5.	Conclusiones y trabajos futuros	113
	Bibliografía	115
	Apéndice A. Manual de Usuario	121
	Apéndice B. Contexto del estudio (Tablas)	133

1

Introducción

1.1. Motivación

En la década de 1860, el físico John Tyndall identificó el efecto invernadero natural de la Tierra y sugirió que ligeros cambios en la composición atmosférica podrían provocar variaciones climáticas. En 1896, un artículo fundamental del científico sueco Svante Arrhenius predijo por primera vez que los cambios en los niveles de dióxido de carbono atmosférico podrían alterar sustancialmente la temperatura de la superficie a través del efecto invernadero [1].

En 1979, en la primera Conferencia Mundial sobre el clima, se identificó el cambio climático como un problema global urgente, con lo que empezó la lucha contra el cambio climático [2]. Actualmente, en España, y concretamente en Andalucía, se hacen evidentes los efectos del cambio climático y el impacto directo sobre el medio ambiente. Muchos de los problemas están directamente relacionados con la gestión hídrica, como las sequías, cada vez más frecuentes y prolongadas, aumentos de temperatura y tormentas y olas de calor inesperadas entre otros muchos problemas. Además, el incremento en la demanda de agua por parte de los sectores agrícola, ganadero, turístico e industrial ejerce una fuerte presión sobre los embalses, que constituyen la principal fuente de abastecimiento de agua dulce en Andalucía. Garantizar y optimizar el abastecimiento de agua es de importante relevancia, tanto para la preservación del medio ambiente como el crecimiento socioeconómico. En este sentido, diversos estudios han señalado la necesidad de implementar estrategias de adaptación y mitigación para una gestión eficiente de los recursos hídricos [3].

Este proyecto surge de la necesidad por encontrar una solución efectiva que garantice el suministro de agua a largo plazo mediante la correcta gestión del agua en los embalses

andaluces. Con este fin, el Trabajo Final de Grado (TFG) consta de una parte común y dos líneas de trabajo: la primera, centrada en el desarrollo de modelos predictivos para escenarios *What-If* (línea particular de este TFG) y la segunda, centrada en el uso de *Transfer Learning* para la reutilización de modelos entre embalses similares. Para ello, se hará uso de la Inteligencia Artificial (IA), un conjunto de tecnologías basadas principalmente en el aprendizaje automático (ML, de sus siglas en inglés) y el aprendizaje profundo (DL, de sus siglas en inglés). Estas técnicas se aplican en campos como el reconocimiento de voz e imagen, traducción, modelado predictivo, análisis de datos, seguridad cibernética y especialmente en áreas donde se requieren procesar grandes cantidades de datos [4].

1.2. Objetivos

La Inteligencia Artificial ofrece diversas soluciones para la predicción y gestión eficiente de los recursos hídricos. Los modelos predictivos permiten anticipar la disponibilidad de agua y tomar decisiones informadas, alineándose con los Objetivos de Desarrollo Sostenible (ODS) de la ONU. Entre los más relevantes se encuentran el ODS 13 (Acción por el Clima), el ODS 12 (Producción y Consumo Responsables) y el ODS 11 (Ciudades y Comunidades Sostenibles).

- **Implementación de una herramienta automática para la extracción de datos:**

Mediante la técnica de *web scraping* y el uso de herramientas y librerías se han implementado códigos en Python para extracción automática de datos. Dichos datos se han extraído de distintas webs de la Junta de Andalucía. Este proceso de extracción forma parte de un conjunto de procesos denominados ETL (Extracción, Transformación y Carga). Posterior a la extracción de los datos, se hizo una limpieza y transformación de los mismos.

- **Creación de una base de datos de series temporales para el almacenamiento de la información extraída:**

Este objetivo agrupa tanto la creación de la base de datos InfluxDB como la fase de carga de los procesos ETL. La creación de la base de datos incluye la creación y configuración de "buckets" la configuración de la propia base de datos. Por otro lado, la carga de los datos se realiza de forma automática mediante un código Python.

- **Aplicación de modelos predictivos personalizables para analizar escenarios *What-If*:**

En este objetivo se pretende diseñar, entrenar y validar modelos predictivos basados en *Machine Learning* y *Deep Learning* para analizar la evolución de las reservas de los embalses de Andalucía. Para ello se integra información climática e hidrológica procedente de la Red de Información Agroclimática de Andalucía (RIA) [5] y la Red de Información Ambiental de Andalucía (REDIAM) [6] en una base de datos de series temporales (InfluxDB). La información histórica de esta base de datos pasa antes por un flujo de extracción y limpieza de datos (ETL/*web scraping*) que mantiene el histórico actualizado y consistente.

Sobre estos datos se entrenan modelos que permiten analizar escenarios *What-If*. Estos escenarios responderán a preguntas como “¿qué pasaría si aumentara la temperatura media debido al cambio climático?” o “¿cómo afectaría una mayor extracción de agua para riego a la disponibilidad del embalse a lo largo del año?”. De esta manera, se optimizará la gestión del agua y se implementarán estrategias de apoyo en la toma de decisiones, fomentando el uso adecuado del agua en embalses y garantizando el bienestar social, económico y medioambiental en Andalucía.

- **Desarrollo de una aplicación web para la visualización de la evolución de los embalses y la meteorología:**

El último objetivo es el desarrollo de una plataforma web para visualización de las series históricas (gráficas y tablas) que sirve como apoyo a usuarios expertos e interesados en la gestión del agua en Andalucía.

1.3. Estructura del documento

Este documento se organiza del modo que sigue, guiando al lector desde el contexto y los objetivos hasta la experimentación, resultados y despliegue de la solución:

- **Sección 1: Introducción** Presenta la motivación del trabajo, los objetivos generales y específicos, así como esta descripción de la estructura del documento.

- **Sección 2: Estado del arte.** Revisa los antecedentes y trabajos relacionados con la predicción en recursos hídricos, series temporales y el uso de técnicas de *Machine Learning* y *Deep Learning*. Se señalan líneas abiertas y tendencias que motivan la realización de este TFG.
- **Sección 3: Análisis experimental** Recoge la metodología y los experimentos realizados. Incluye:
 1. **Entrenamiento y validación:** definición de conjuntos de datos y preprocesado, selección de características (*feature selection*), arquitectura de redes neuronales y protocolo de entrenamiento y validación.
 2. **Escenarios *What-If*:** formulación de escenarios, procedimiento de generación y evaluación de impactos en las variables de interés.
 3. **Resultados:** se presentan y analizan los resultados de ambos objetivos (métricas, comparativas, gráficas y tablas) y se incluye una **discusión general** integrando hallazgos y limitaciones.
- **Sección 4: Aplicación:** Describe la plataforma software desarrollada para consulta y visualización. Se estructura en:
 - **Tecnologías:** resumen de las herramientas de desarrollo empleadas (p. ej., Python, InfluxDB, JavaScript, etc.).
 - **Ingeniería del software:** arquitectura de la plataforma web (frontend y backend), estructura del proyecto, despliegue de servicios y funcionamiento de la aplicación.
 - **Procesos ETL:** detalle de las fases de extracción, transformación y carga de los datos, junto con problemas encontrados y soluciones adoptadas.
- **Sección 5: Conclusiones y trabajos futuros:** Resume las principales aportaciones del trabajo, limitaciones y posibles líneas de mejora y continuidad (incluyendo la ampliación de escenarios *What-If*).
- **Sección A: Apéndice A** Se incluye como material de apoyo un **Manual de usuario** de la plataforma web.

- **Sección B: Apéndice B** Se incluyen las tablas de clasificación de embalses y estaciones meteorológicas.

2

Estado del arte

En esta sección repasamos los antecedentes y trabajos relacionados con la predicción y la gestión de recursos hídricos usando series temporales y técnicas de *Machine Learning* y *Deep Learning*. Explicamos de forma breve qué modelos computacionales utilizamos en este Trabajo Final de Grado (TFG) y con qué propósito y resumimos las métricas con las que evaluamos los modelos. Situamos el caso de estudio en Andalucía. Por último, describimos las tecnologías empleadas tanto en el análisis experimental como en el desarrollo de la aplicación web.

2.1. Modelos computacionales aplicados al análisis y gestión del agua

El análisis y la gestión de embalses se apoyan cada vez más en datos abiertos y en modelos computacionales capaces de explotar series temporales hidrometeorológicas. El flujo de trabajo habitual parte de la limpieza y unificación de fuentes de datos heterogéneas (estaciones con históricos desiguales, nomenclaturas distintas, huecos de medida), y generación de agregados temporales. En la literatura hidrológica se han empleado tanto enfoques clásicos de series temporales como modelos de aprendizaje automático y profundo para pronosticar caudales, niveles y aportes a embalses. Los modelos **ARIMA/SARIMA** se han aplicado a predicción de niveles y aportes en distintos contextos, mostrando capacidad para capturar estacionalidad y dependencias de corto plazo [7]. En el ámbito de *machine learning*, se han usado *Random Forest* y también *gradient boosting* (XGBoost) para predicción de caudales y métricas relacionadas, con buenos resultados frente a métodos base [8, 9]. En *deep learning*, las LSTM y GRU destacan por modelar dependencias de largo plazo y estacionalidad hidrológica. Varios trabajos comparativos y de referencia muestran su rendimiento competitivo e incluso superior a modelos tradicionales en problemas de *rainfall-runoff* y caudales en múltiples cuencas [10, 11]. Revisiones recientes sintetizan estas evidencias y tendencias en el uso de *Deep Learning* y *Machine Learning* en recursos hídricos [12]. En paralelo, el uso de bases de datos de series

temporales, como InfluxDB, simplifica la ingesta continua, el cálculo de agregados, el análisis y la alimentación de modelos computacionales.

En cuanto a proyectos y soluciones ya consolidadas, **WEAP (Water Evaluation and Planning, del Stockholm Environment Institute)** es una plataforma de planificación de recursos hídricos que diseña sistemas de cuenca utilizando un esquema de nodos y enlaces (embalses, ríos, captaciones y demandas) y los distribuye por orden de prioridad. Ofrece un "gestor de escenarios" muy cómodo para el análisis *What-If* y permite la comparación entre las reglas operativas de embalses, los escenarios de cambio climático y el crecimiento de la demanda. Puede integrar series climáticas externas, añadir pérdidas por evapotranspiración y hasta combinarse con modelos de acuíferos (*Modflow*) o módulos de calidad del agua. Esto lo convierte en una herramienta muy útil para las entidades de cuenca que desean hacer una evaluación de políticas antes de implementarlas [13].

A nivel regional, la Junta de Andalucía ha establecido el Centro de Gestión de Recursos Hídricos "**Smart Water**", que emplea inteligencia artificial y análisis avanzados para consolidar información meteorológica e hidrológica, crear modelos computacionales y escenarios climáticos en las cuencas andaluzas y ayudar a tomar decisiones frente a sequías u otros contratiempos. De acuerdo con la comunicación institucional, el objetivo es contar con un ambiente operativo que incluya cuadros de mando, ingesta constante de datos y herramientas de simulación para los técnicos. Esto permitirá que puedan comparar opciones de gestión con más rapidez y trazabilidad [14].

En la vertiente de modelos centrados en datos, **HydroForecast (Upstream Tech)** muestra que es posible predecir caudales en la dirección de los modelos centrados en datos, al combinar las señales históricas, la teledetección y los forzantes meteorológicos (observados y pronosticados) con modelos de aprendizaje automático. Es un servicio operativo que ofrece entregas a través de API y paneles web, diseñado para la gestión de riesgos, caudales ecológicos o hidroelectricidad. Los horizontes son de días a meses dependiendo del tipo de cuenca y la disponibilidad de datos. Su perspectiva demuestra que los modelos de ML "basin-agnostic" pueden generalizarse a diversas cuencas y proporcionar valor en la explotación diaria [15].

En resumen, todas estas soluciones concuerdan con dos conceptos fundamentales: la comunicación de resultados a través de visores o paneles web es indispensable para que sean adoptados; y se puede utilizar series históricas junto con forzantes externos para examinar impactos climáticos o alternativas operativas. Nuestro trabajo integra la lógica de comparación de políticas y escenarios de WEAP, la predicción basada en datos mediante modelos de aprendizaje profundo (similar a HydroForecast) y el enfoque operativo de proyectos como "Smart Water", todo ello adaptado a su escala. En particular, combinamos datos históricos de embalses andaluces con variables meteorológicas (RIA/REDIAM), los consolidamos en una base de series temporales y entrenamos modelos GRU utilizando variables externas y derivadas para capturar la estacionalidad y los efectos retardados. A partir de esas predicciones, creamos escenarios *What-If*. Todo esto se ofrece en una plataforma web que permite la consulta de datos históricos y su evolución a través de gráficos, diseñada para usuarios técnicos y para aquellos que estén interesados. Al igual que en los trabajos relacionados, partimos de la hipótesis de que los modelos son correlacionales (no demuestran causalidad) y que hay mecanismos físicos no detectados, por ejemplo la infiltración al suelo/acuífero, retornos o decisiones operativas, que pueden influir en el ajuste; por lo tanto, especificamos supuestos, contrastamos líneas base y mantenemos la posibilidad de incluir nuevas variables en el futuro.

2.2. Modelos analizados en este trabajo

▪ Redes neuronales recurrentes (RNN)

Es una red neuronal que se entrena con datos secuenciales o de series temporales cuyo fin es el de crear modelos de *machine learning* que haga predicciones basadas en entradas secuenciales [16].

Algunos de los usos mas comunes:

- Ya que los datos meteorológicos son datos secuenciales, esta red neuronal se podría usar para **predecir los niveles diarios de inundación** basándose en datos de inundaciones y meteorológicos pasados.
- **Procesamiento de lenguaje natural (NLP).**
- **Reconocimiento de voz.**

- **Subtitulado de imágenes.**

A diferencia de las CNN (redes neuronales convolucionales) donde las entradas y salidas son independientes entre sí, las RNN toman información de las entradas anteriores influyendo en las entradas y salidas actuales, por lo que se distinguen por su memoria. [16]

Las RNN comparten el mismo peso dentro de cada capa de la red, a diferencia de las redes feedforward cuyo peso es diferente en cada nodo. Además, usa algoritmos de retropropagación a través del tiempo (BPTT) para determinar los gradientes. Este algoritmo difiere de los algoritmos de retropropagación tradicionales ya que es específico de los datos secuenciales. [16]

- **¿Cómo funciona el algoritmo BPTT?**

El modelo se entrena a sí mismo calculando los errores de su capa de salida a su capa de entrada. Estos errores nos permiten ajustar los parámetros del modelo correctamente. Como los parámetros se comparten dentro de cada capa de la red el algoritmo BPTT suma los errores en cada paso de una secuencia.

- **¿Cuáles son los problemas del RNN?**

Los dos principales problemas de las RNN son el cálculo de los gradientes de explosión y los gradientes de desaparición. El gradiente se define como la pendiente de la función de pérdida a lo largo de la curva de error. Cuando el gradiente es muy pequeño tiende a seguir haciéndose pequeño, por lo tanto los pesos de los nodos terminan siendo insignificantes y como consecuencia el algoritmo deja de aprender. Por el contrario, cuando el gradiente es muy grande los pesos crecen mucho y terminan representándose como NaN (not a number) y por lo tanto el modelo que resulta es inestable y surgen dificultades para procesar dependencias a largo plazo.[16]

Existen varios tipos de RNN, véase la figura 1:

- **One to One:** Se trata de una red neuronal simple que se usa para problemas de aprendizaje automático que solo tienen una entrada y una salida.

- **One to Many:** Una entrada y múltiples salidas. Se utiliza para generar títulos de imágenes.
- **Many to One:** Toma muchas entradas y predice una única salida. Se usa para la clasificación de sentimientos donde la entrada es un texto y la salida es una categoría.
- **Many to Many:** Múltiples entradas y salidas. Se usa comúnmente para la traducción automática

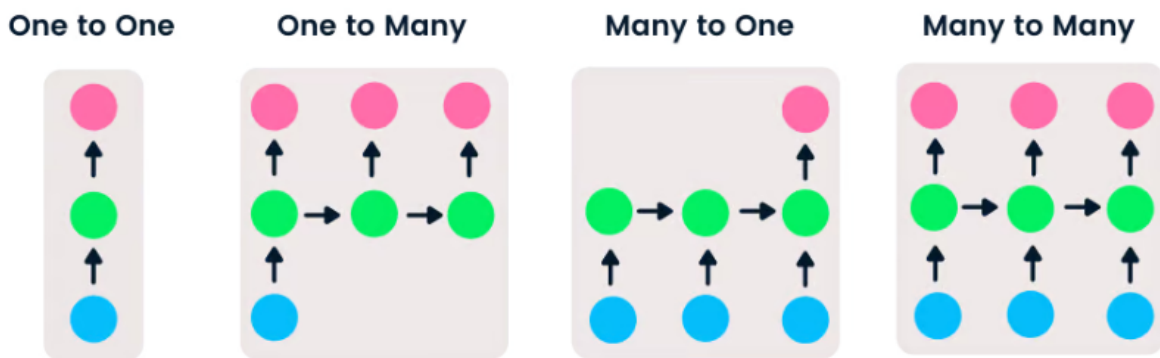


Figura 1: Tipos de redes neuronales recurrentes

Este tipo de arquitectura neuronal presenta algunas variantes, entre ellas:

- **Redes Neuronales recurrentes bidireccionales (BRRN)**
 - **Memoria a corto plazo (LSTM)**
 - **Unidades recurrentes con compuertas (GRU)**
- **LSTM (Long Short-Term Memory)**

Es una variante de la arquitectura RNN que surge como una solución al problema del gradiente de desaparición. Introduce una novedad llamada celda de memoria. Esta celda permite almacenar y acceder a la información durante un periodo de tiempo. [17]

Algunos de los usos mas comunes:

- **Traducción automática.**

- **Generar texto.**
- **Reconocimiento de voz.**
- **Predicción y clasificación de secuencias:** predicción de series temporales, análisis de sentimientos en textos, detección de anomalías...etc.

Esta arquitectura presenta algunas desventajas:

- Más difícil de entrenar por un mayor número de parámetros.
- Necesidad de más datos de entrenamiento.
- Necesidad de mayor potencia de cálculo para obtener buenos resultados.
- Sensible al sobreentrenamiento.
- **¿Cómo funciona todo esto?**

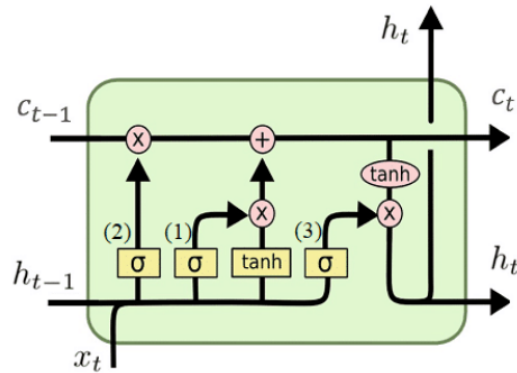
El mecanismo que sigue una LSTM es el siguiente (véase la figura 2):

La compuerta 1 de entrada usa la entrada x_t y el estado previo de la celda de memoria $h_{(t-1)}$ y genera un vector de activación que representa la información que se añadirá a la celda $c_{(t-1)}$. Esta adición es una operación matemática entre el vector de activación y el estado previo $c_{(t-1)}$.

La compuerta de olvido (2) usa tanto la entrada actual (x_t) como el estado previo para genera un vector de activación que determinará qué información se eliminará. Esta adición también es una operación matemática entre este segundo vector de activación y el estado previo ($c_{(t-1)}$). A partir de las dos operaciones previas sobre $c_{(t-1)}$ se obtiene el estado actual c_t .

Finalmente, la compuerta de salida 3 usa la entrada actual x_t y el estado actual de la celda de memoria c_t para generar otro vector de activación que representa la salida del LSTM (h_t).

De esta forma el LSTM mantiene un flujo constante de información a través del tiempo durante la retropropagación del gradiente solucionando así el problema del gradiente de desaparición.



LSTM
(Long-Short Term Memory)
Diagrama de una célula de memoria LSTM

Figura 2: Estructura general de LSTM

▪ **GRU (Gated Recurrent Units)**

Surgen para resolver algunos problemas que tenían las RNNs tradicionales, por ejemplo: dependencias a largo plazo en datos secuenciales. Esta arquitectura procesa datos secuenciales (texto o datos de series temporales) usando mecanismos de compuertas especializados. Dichos mecanismos gestionan el flujo de información almacenándola, actualizándola o descartándola en cada paso de una secuencia. De esta forma detectan dependencias y patrones temporales en los datos. [18]

Algunos de los usos mas comunes:

- **Procesamiento de lenguaje natural (NLP):** Traducción automática, síntesis de texto y análisis de sentimientos.
- **Reconocimiento de voz:** Procesamiento de datos asociados a series temporales para el desarrollo de modelos que transforman el audio en texto.
- **Análisis de series temporales:** Detección de tendencias de mercado de valores y patrones climáticos.

Respecto a las otras dos arquitecturas anteriores, esta presenta algunas ventajas:

- Entrenamiento más rápido.
- Mejor rendimiento a la vez que menos demanda computacional.

- Necesita menos parámetros por lo que es una arquitectura más simple.
- Compatible con grandes volúmenes de datos.
- **¿Cómo funciona todo esto?**

Esta arquitectura está compuesta por 2 mecanismos de compuertas, como se muestra en la figura 3:

- **Compuerta de actualización:** Determina el grado en el que la arquitectura conserva la información del pasado. En cada paso de la secuencia el modelo decide si actualiza su estado oculto con nuevas entradas o no.
- **Compuerta de reseteo:** Su función es decidir cuanta información del pasado se descarta (se olvida) porque es irrelevante.

El estado actual se calcula a partir de las dos compuertas anteriores, mezclando la nueva entrada con la información retenida del estado anterior. Este estado captura las dependencias a corto y largo plazo.

La arquitectura GRU es una simplificación de la LSTM ya que combina la celda de estado con las compuertas de entrada y salida en un único mecanismo de forma que la cantidad de parámetros a aprender es menor.

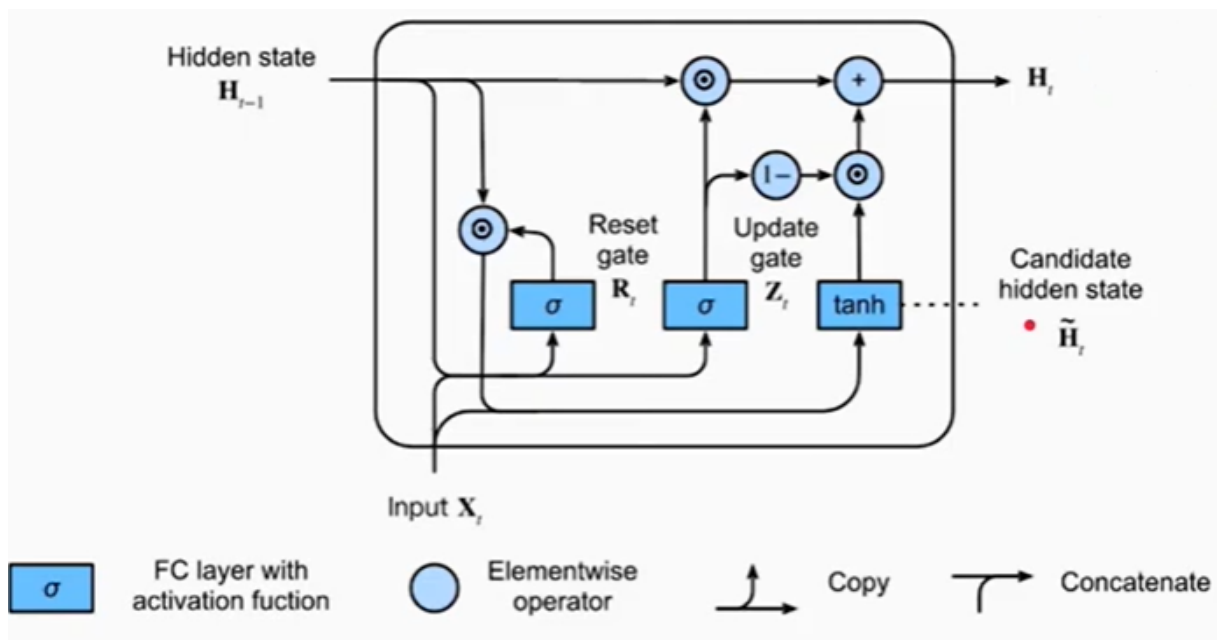


Figura 3: Estructura general de GRU

En este trabajo se ha optado por GRU porque:

- Presenta un 30–40 % menos de parámetros que la LSTM equivalente, lo que reduce el riesgo de sobreajuste;
- Converge más rápido, importante al re-entrenar con ventanas de 48 meses y 73 embalses;
- Ofrece métricas (MAE, RMSE) muy próximas a LSTM en ensayos preliminares, con menor coste computacional.

Estas arquitecturas las aplicaremos a modelos de aprendizaje automático. Estos modelos se pueden clasificar en tres tipos: [19]

- **Supervisados:** usan datos etiquetados para el entrenamiento.
- **No supervisados:** busca patrones en datos no etiquetados.
- **Por refuerzo:** un agente aprende a tomar decisiones interactuando con el entorno.

En nuestro caso, debido al contexto del TFG y del análisis usaremos modelos supervisados. Dichos modelos se usan para contextos que requieren predecir resultados basados en características de entrada. El algoritmo aprende a hacer predicciones al transformar los datos de entradas en etiquetas conocidas.

Algunos de los modelos supervisados más populares son:

- **Regresión lineal**
- **Regresión logística**
- **Árboles de decisión**
- **Bosque aleatorio** (*Random Forest*)
- **Redes neuronales**
- **Naïve Bayes**
- **K-NN** (K-vecinos más cercanos)
- **SVM** (Máquinas de soporte vectorial)

2.3. Contexto del estudio

Andalucía es una región amplia y muy diversa: al norte limita con Sierra Morena, en el centro se extiende el valle del Guadalquivir y al sur y este se elevan las cordilleras Béticas, con un litoral que mira tanto al Atlántico como al Mediterráneo. Este marco físico, unido a un clima mediterráneo con lluvias irregulares (concentradas en otoño–invierno) y veranos largos y secos, condiciona fuertemente la disponibilidad de agua. La evapotranspiración estival es elevada, la variabilidad interanual de precipitaciones es grande y, por tanto, el almacenamiento en embalses resulta esencial para asegurar abastecimiento urbano, riego, actividades económicas y conservación ambiental.

La importancia del agua en Andalucía es hoy, además, un asunto social. En los últimos años, diferentes encuestas y barómetros señalan que la sequía se percibe como uno de los principales problemas medioambientales de la región, con un claro aumento de la preocupación ciudadana. La AEMET ha calificado varios periodos recientes como “extremadamente secos”, y los otoños e inviernos, claves para la recarga, han sido deficitarios en precipitaciones, lo que se ha traducido en niveles de embalses por debajo de la media histórica [20, 21].

A esta presión hidrológica se suman los usos del territorio. El sector agrario concentra la mayor parte del consumo de agua (más del 70 % en la mayoría de demarcaciones), con especial peso del regadío. Los requerimientos anuales orientativos por hectárea varían según cultivo: forrajeros como la alfalfa o el maíz forrajero pueden superar los 7 000–10 000 m³/ha; cítricos y frutales suelen situarse entre 3 000–6 000 m³/ha y el olivar en regadío entre 1 500–3 000 m³/ha. Por provincias, Almería y Huelva destacan por invernaderos y frutales, Jaén y Córdoba por el olivar (con riego localizado en crecimiento). En paralelo, el turismo incrementa la demanda estival, precisamente cuando hay menos agua, y concentra consumos en la franja litoral (Málaga, Cádiz, Almería). Diversos informes recuerdan que el gasto por turista puede multiplicar el consumo doméstico medio, lo que obliga a reforzar medidas de eficiencia y reutilización en alojamientos y servicios municipales [22, 23].

También la dinámica demográfica importa: Andalucía ha crecido de forma sostenida desde 1970, con mayores incrementos en áreas urbanas y costeras (Málaga, Sevilla, Almería). Esto exige más capacidad de abastecimiento, saneamiento y depuración, y una planificación fina de la demanda. Por demarcaciones, Guadalquivir y Guadiana presentan balances más holgados,

mientras que Cuencas Mediterráneas Andaluzas registran déficit, y Guadalete-Barbate opera con márgenes ajustados. Todo ello hace evidente la necesidad de gestionar el agua con criterios de eficiencia y resiliencia [24].

En este contexto, la gestión del agua en Andalucía depende de una red densa de embalses y de sistemas de aforo y observación meteorológica. Las principales fuentes de datos empleadas en este trabajo son la Red de Información Agroclimática de Andalucía (RIA) [5], que proporciona series temporales de variables meteorológicas medidas en estaciones distribuidas por la región, y el visor de embalses de la Red de Información Ambiental de Andalucía (REDIAM) [6], que ofrece información histórica de almacenamiento y porcentaje de llenado por embalse. Estas series permiten estudiar patrones estacionales y tendencias, y sirven de base para el entrenamiento de modelos predictivos.

En la figura 4 se incluye un mapa del relieve principal de Andalucía [25].



Figura 4: Datos Espaciales de Referencia de Andalucía (DERA)

2.3.1. Red de embalses

A continuación se clasificarán los embalses Andaluces usados en este proyecto para el entrenamiento de los modelos.

Los embalses se han clasificado por demarcaciones. Una demarcación hidrográfica es una agrupación de cuencas hidrográficas para la gestión hídrica de los recursos.

En total, Andalucía consta de 73 embalses clasificados en 4 demarcaciones.

Cada uno de los embalses cuenta con dos variables que se almacenan en una base de datos de series temporales:

- **Reserva** (hm^3): Es el volumen de agua almacenada en el embalse dedicado a abasteci-

miento o riego. Se mide en hectómetros cúbicos.

- **Porcentaje de reserva (%)**: Es la relación entre la reserva y la capacidad total del embalse. Se mide en tantos por cien.

2.3.2. Red de estaciones agroclimáticas

Una estación agroclimática es un conjunto de dispositivos cuya función es la de recoger datos meteorológicos como la Temperatura, Radiación, Precipitación, etc. Estas estaciones se usan en diversos sectores, como la agricultura [26].

Para la instalación de estas estaciones agroclimáticas se necesita un estudio previo para determinar la ubicación, los instrumentos y la adecuada disposición [26].

Existen dos tipos de estaciones agroclimáticas:

- **Estaciones manuales**: se componen de instrumentos, dispositivos analógicos.
- **Estaciones automáticas**: se componen de sensores. Estos son dispositivos electrónicos o digitales que miden y transmiten la información medida automáticamente. Las estaciones automáticas surgen de la necesidad de tomar medidas en sitios de difícil acceso, o estaciones donde se necesitaban tomar medidas fuera del horario laboral del personal.

Elementos que componen una estación agroclimática: [26]

- **Trípode**: Se trata de un mástil de 2 metros de acero galvanizado con 3 patas. En el mástil se incorporan los brazos con los piranómetros y los sensores de viento. El sensor de dirección del viento apunta al norte mientras que el piranómetro, sensor de radiación, apunta al sur.
- **Caja intemperie**: El modelo ENC 12/14 aloja la unidad de proceso, el módem y un regulador de batería. El material de la caja permite proteger estos elementos del agua, la corrosión y el polvo. En el interior se instala la alimentación, el registrador de datos (datalogger) y los periféricos mientras que en la base se instalan los conectores para los sensores y el cable de toma de tierra.
- **Alimentación**: Es la fuente de alimentación externa que suministra energía al datalogger y al módem GSM. Se emplea una batería modelo DELTA80 que se recargará con un panel solar de 20W.

- **Sonda HR y TA:** Este dispositivo consta de un circuito que detecta los cambios, los linealiza, los amplifica y compensa la humedad relativa con la temperatura. Además, consta de un protector modelo URS1 para la radiación que protege al sensor de temperatura de la radiación directa y difusa de forma que las temperaturas que se tomen sean reales. Dicho dispositivo también protege al sensor de humedad relativa del rocío, la lluvia o la condensación evitando así el efecto de histéresis.

La temperatura se mide con un sensor de termorresistencias Pt1000 que cada vez que cambia la temperatura se modifica la resistencia eléctrica [27].

Para medir la humedad relativa se usa un sensor capacitivo de estado-sólido HUMICAP 180 que modifica sus características eléctricas cada vez que varía la humedad, disminuyendo la capacidad cada vez que absorbe la humedad.

- **Pluviómetro:** Es un dispositivo que mide el volumen de precipitación en un lugar en un tiempo determinado.
- **Anemoveleta:** Sirve para medir la velocidad y la dirección del viento.
- **Piranómetro:** Sirve para medir la radiación solar.
- **Unidad central:** Este dispositivo recibe la señal de todos los anteriores sensores y los procesa. Usa un registrador de datos que se sitúa en una caja a la intemperie.
- **Módem:** Es una terminal de GSM que transmite datos, fax, SMS y voz.

Al igual que los embalses, las estaciones meteorológicas cuentan con doce variables que se almacenan en la base de datos de series temporales:

- **Temperatura máxima (°C):** temperatura máxima registrada en un día en particular. Se mide en grados celsius.
- **Temperatura mínima (°C):** temperatura mínima registrada en un día en particular. Se mide en grados celsius.
- **Temperatura media (°C):** temperatura media de un día en concreto. Se mide en grados celsius.

- **Humedad máxima (%)**: Es la cantidad máxima de vapor de agua que se encuentra en un determinado volumen de aire en un día en concreto. En este caso se mide en tantos por cien.
- **Humedad mínima (%)**: Es la cantidad mínima de vapor de agua que se encuentra en un determinado volumen de aire en un día en concreto. En este caso se mide en tantos por cien.
- **Humedad media (%)**: Es la cantidad media de vapor de agua que se encuentra en un determinado volumen de aire en un día en concreto. En este caso se mide en tantos por cien.
- **Velocidad del viento (m/s)**: Es el desplazamiento del aire en un espacio e instante determinado de un día en concreto. Se mide en metros por segundo.
- **Velocidad del viento máxima (m/s)**: Es el desplazamiento máximo del aire en un espacio e instante determinado en un día en concreto. Se mide en metros por segundo.
- **Dirección del viento (°)**: Nos indica de donde viene el viento. Se mide en grados, desde a 1° a 360°. Valores cercanos a 1° y 360° indican vientos procedentes del norte, a 90° vientos procedentes del este, a 180° vientos procedentes del sur y 270° del oeste [28].
- **Radiación (MJ/m²)**: Es la energía emitida por el sol que llega a la tierra en forma de ondas electromagnéticas. Se mide en megajulios por metro cuadrado.
- **Precipitación (mm)**: Es la cantidad de lluvia que se registra en un día en concreto. Se mide en milímetros.
- **Evapotranspiración de referencia (ETO)**: La evapotranspiración es la combinación de dos sucesos, la evaporación del agua del suelo y la transpiración de los cultivos. Esto nos ayuda a medir la cantidad de agua que se pierde por estos dos sucesos [29].

ETO viene del término evapotranspiración de referencia y es la cantidad de agua demandada de la atmósfera usando, como su nombre indica, un cultivo de referencia. Esto se usa para calcular la ETc, que es la evapotranspiración real de ese cultivo, junto con un coeficiente Kc. Cada tipo de planta tiene su Kc. Además, para calcular el ETO se usan

la radiación, la velocidad del viento, la temperatura y la humedad.

Las estaciones se han clasificado por provincias. Andalucía cuenta con 110 estaciones, número que varía a lo largo del tiempo por bajas, altas y periodos de inactividad.

Las tablas detalladas con el listado de embalses por demarcación 12, 13, 14, 15 y las relaciones completas de estaciones por provincia 16, 17 se han desplazado al **Apéndice B** para no sobrecargar la lectura en esta sección (véanse los cuadros correspondientes).

2.4. Métricas empleadas en este Trabajo Final de Grado

Para medir la eficiencia de los experimentos se han usado las siguientes métricas:

- **MAE (Mean Absolute Error):** Nos permite saber cuanto es la diferencia entre el valor predicho y el valor real. Se utiliza el valor absoluto para que un error valor negativo no anule un error con valor positivo. Además, se usa el promedio ya que nos interesa saber el error de todas las observaciones, no de una en concreto. La fórmula es la siguiente:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i|$$

- **RMSE (Root Mean Square Error):** Nos permite saber cuan cerca está la línea de la regresión a las observaciones. Esta métrica nos da los valores en las unidades originales gracias a la raíz cuadrada. La fórmula es la siguiente:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2}$$

- **R² (Coeficiente de determinación):** Nos ayuda a determinar la variación de la variable dependiente que queremos predecir desde la variable independiente, es decir, que tan bien se ajusta nuestro modelo a las observaciones reales. La fórmula es la siguiente:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - x_i)^2}{\sum_{i=1}^n (y_i - \mu_y)^2}$$

- **MAPE (Mean Absolute Percentage Error):** Mide la precisión del modelo. En este caso, el resultado se muestra en tantos por ciento. La fórmula es la siguiente:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{y_i} \right|$$

- **Pérdida en entrenamiento:** es el MAE promedio del conjunto de datos de entrenamiento (80 %).
- **Pérdida en validación:** es el MAE promedio del conjunto de datos de validación (20 %).

2.5. Tecnologías empleadas

En esta sección se presentan las tecnologías empleadas tanto para el análisis experimental, entrenamiento de modelos y escenarios *What-If*, como para el desarrollo de la aplicación web. En el desarrollo de este proyecto se han utilizado principalmente los lenguajes de programación Python, InfluxQL y JavaScript. Además, se ha usado InfluxDB, Visual Studio Code y Google Colaboratory entre otras tecnologías que se detallarán a continuación.

2.5.1. Python

Python es un lenguaje de alto nivel de programación comúnmente utilizado en aplicaciones web, ciencia de datos y *machine learning*, entre otros usos. En este proyecto se ha usado para la automatización de *scripts* de *web scraping* y para las tareas relacionadas con las fases de *machine learning* [30].

Las características más importantes de este lenguaje son:

- **Lenguaje interpretado:** el programa se ejecutará con un intérprete, no con un compilador.
- **Tipo dinámico:** no es necesario asignar el tipo de dato previamente a la variable, se asignará según el tipo que le asignemos en la ejecución.
- **Fuertemente tipado:** no se pueden mezclar distintos tipos de datos.
- **Multiplataforma:** el intérprete es compatible con Linux, Windows, etc.
- **Orientado a objetos:** se basa en datos y objetos en lugar de usar funciones y lógica.

2.5.2. InfluxDB (Influx DataBase)

InfluxDB es una base de datos de código abierto que se usa especialmente para almacenar y analizar grandes cantidades de datos temporales. Está escrito en el lenguaje de programación

de Google, Golang. Comenzó a desarrollarse a finales de 2013 por la empresa Errplane, que más tarde pasó a llamarse InfluxData. La primera versión de código abierto (InfluxDB 0.5) se lanzó en 2014. [31]

Algunas de las características más relevantes y por la que se ha decidido usar esta base de datos son:

- **Orientado a series temporales:** cada dato que se almacena cuenta con un timestamp (marca de tiempo) que facilita el análisis de tendencias a lo largo del tiempo.
- **Alto rendimiento:** capacidad alta de escritura y consultas en tiempo real.
- **Integración con *machine learning*:** compatible con Python, Tensorflow, Pandas y Grafana, entre otros.

InfluxDB admite dos lenguajes de consulta:

- **InfluxQL (Influx Query Language):** es un lenguaje de consulta similar a SQL orientado a trabajar con datos y series temporales. Se usa para versiones 1.x de influxDB y en algunos entornos de InfluxDB 2.0.
- **Flux:** es el lenguaje de consultas para InfluxDB 2.0, diseñado para ser más flexible y potente que InfluxQL. A partir de la versión 1.8 de InfluxDB se integra con InfluxQL para uso de producción. Es el que se ha usado en este proyecto.

2.5.3. JavaScript

JavaScript es un lenguaje de programación dinámico e interpretado, usado principalmente para crear páginas web interactivas y con ello contenido dinámico en la propia página web [32].

Fue creado por Netscape Communications por Brendan Eich en 1995 como lenguaje de programación para páginas web en el navegador de la propia empresa, Netscape Navigator. A lo largo de los años, el resto de navegadores gráficos han ido adoptando dicho lenguaje [33].

Algunas de las características más relevantes de este lenguaje son:

- **Lenguaje de alto nivel** fácil de aprender y usar.
- **Basado en eventos** por lo que es ideal para interacciones dinámicas en la web.

- **Multiparadigma**, por lo que soporta programación imperativa, funcional y orientada a objetos.
- **No requiere compilación**, es decir, se ejecuta directamente en el navegador.
- Se integra eficientemente con APIs, base de datos, frameworks y otras tecnologías.

2.5.4. Visual Studio Code

Visual Studio Code es un editor de código fuente con licencia de código abierto. Esta aplicación nos permite editar desde texto plano a editar código escrito en cualquier lenguaje de programación. Integra herramientas de depuración de código y herramientas para explorar y trabajar archivos y carpetas. Además, se integra con Git, un sistema de control de versiones muy popular que nos permite trabajar con repositorios. En definitiva, integra múltiples herramientas y funcionalidades lo que hace que sea una aplicación muy potente y popular.[34]

Fue desarrollado por Microsoft y se lanzó al público en 2015. Es compatible con Linux y Windows.

2.5.5. Google Colaboratory

Es una herramienta gratuita alojada en Jupyter Notebook y basada en la nube que permite al usuario escribir, ejecutar y compartir código Python a través de los llamados notebooks o cuadernos, sin necesidad de configuración ni instalación en su máquina local. Es una gran solución especialmente en áreas como la ciencia de datos y el aprendizaje automático ya que nos permite entrenar modelos de *machine learning* y *deep learning* en CPU, GPU y TPU.[35]

Dicha herramienta fue desarrollada en 2017 por Google Research.

2.5.6. Git + Github

Git es un sistema de control de versiones que permite al usuario realizar un seguimiento de los cambios en los archivos. En este proyecto se usará para el desarrollo de la web para la visualización de los resultados ya que, al tratarse de un TFG grupal, habrá varias personas haciendo cambios en los mismos archivos al mismo tiempo. [36]

Los archivos son almacenados en un repositorio de Git y a través de distintas ramas que se crean a partir de la copia principal se realizarán las modificaciones necesarias. Una vez hechas las modificaciones en cada rama, se deberán fusionar con la principal (rama master).

Github es la plataforma donde se alojan los repositorios de Git. Además, nos permite compartir código y trabajar conjuntamente con otras personas.

El uso de estas dos tecnologías nos asegura un desarrollo de código controlado y seguro.

2.5.7. Docker

Es una de las plataformas de contenedores más usadas y conocidas en la actualidad. Fue fundada en 2010 por Solomon Hykes, Kamel Founadi y Sebastien Pahl, aunque se lanzó al público en 2013 como código abierto.[37]

Docker nos permite crear, gestionar y desplegar contenedores de forma eficiente. Los contenedores son una forma de virtualización ligera a nivel de sistema operativo que permite al usuario empaquetar todas las dependencias del proyecto y ejecutarlas fácilmente en cualquier entorno.[38]

Algunas de las ventajas de Docker frente a otras plataformas como Red Hat OpenShift o Podman son:

- **Modularidad:** separa las partes de la aplicación para actualizarlas sin necesidad de deshabilitarla completamente.
- **Implementación rápida:** reducción de los tiempos de implementación del sistema con respecto a la puesta en marcha de sistemas hardware antiguos.
- **Capas y control de versiones de imágenes:** cada imagen Docker se compone de varias capas, por lo que cuando el usuario hace cualquier cambio, se crea una nueva capa en la imagen, haciendo el control de versiones mucho más eficiente y ágil.

2.5.8. Node.js

Es un entorno de ejecución JavaScript de código abierto que permite crear servidores y aplicaciones web.

Dicho entorno es asíncrono y su arquitectura está basada en eventos. Además usa una arquitectura Single Threaded Event Loop, lo que hace que se puedan procesar varios clientes

al mismo tiempo sin necesidad de enviar una respuesta después de cada solicitud individual. Esto es una gran ventaja respecto a otras arquitecturas ya que usa menos recursos de memoria y el tiempo de ejecución de las tareas es menor. [39]

Algunas de las características más destacables son:

- **Escalable:** puede manejar un gran número de conexiones eficientemente.
- **Fácil de usar:** es ideal para aquellos usuarios que comienzan en el mundo del desarrollo web por su facilidad de uso, además de su amplia comunidad.
- **Velocidad:** su arquitectura hace que los tiempos de ejecución sean menores.
- **Multiplataforma**
- **Paquetes:** cuenta con un amplio conjunto de paquetes que facilitan el desarrollo web.

2.5.9. Bootstrap

Es un framework orientado a la creación de sitios web adaptables y aplicaciones móviles, lo que garantiza que se adapte a todo tipo de pantallas en cualquier dispositivo. Consta de dos versiones, precompilada y basada en código fuente. Fue creado por Mark Otto y Jacob Thornton, trabajadores de Twitter. Se lanzó el 19 de Agosto de 2011 y hasta el día de hoy cuenta con más de 20 versiones.[40]

Algunas de sus ventajas son: [41]

- **Facilidad de uso:** tiene una curva de aprendizaje baja. Además, cuenta con una gran comunidad con tutoriales y foros.
- **Rejilla responsive (*grid*):** es el sistema de columnas que se adapta a la pantalla con el que construyes una página.
- **Mutiplataforma**
- **Sistema de imágenes:** consta de reglas HTML y CSS predefinidas para gestionar las imágenes.

2.5.10. Express

Es el framework estándar de backend para Node.js. Se utiliza para crear sitios web minimalistas de una única página, multipáginas e híbridos escalables. Consta de herramientas para el enrutamiento y middleware para crear y desplegar aplicaciones a gran escala, entre otras. Fue desarrollado en 2010 por TJ Holowaychuk. [42]

Algunas de las ventajas son:

- **Flexible y rápido**
- **Escalable:** apenas requiere configuración y cuenta con un gran número de módulos y recursos para facilitar a los usuarios la creación de páginas web fiables.
- **Parte de la Stack MEAN:** se integra con MongoDB, Angular (o React/Vue) y Node.js para un desarrollo full-stack en JavaScript.
- **Compatible con el Motor V8 de Google** lo que le hace aún más potente para crear páginas webs con alto rendimiento y escalables.
- **Gran comunidad**

2.5.11. HTML (HyperText Markup Language)

Es el lenguaje estándar para el desarrollo de páginas web, aunque no se considera realmente un lenguaje de programación. Dicho estándar define los contenidos que se muestran en la web. Además, se considera el lenguaje principal de la WWW (World Wide Web). Sus orígenes se remontan a 1980 cuando Tim Berners-Lee quién desarrolló la idea, aunque fue en 1991 cuando salió el primer documento HTML. [43]

Algunas de las características destacables son:

- **Lenguaje de marcado:** usa etiquetas (tags) para definir la forma en la que los contenidos se muestran en la web.
- **Independiente de la plataforma:** se puede ver en cualquier navegador web de cualquier sistema operativo.
- **Basado en texto plano:** se puede editar con cualquier editor de texto.

- **Integrable:** se integra con CSS, JavaScript, SVG, entre otros.

2.5.12. CSS (Cascade Style Sheets)

Es el lenguaje de estilos que definen la forma en la que se renderizan los documentos HTML, como el color, márgenes y estilos de las letras, etc. Su nombre, Hojas de estilo en cascada, viene porque el código se aplica de arriba abajo. Fue propuesto en 1994 por Håkon Wium Lie aunque fue en 1995 cuando realmente se crea dicho lenguaje. [44]

Un aspecto fundamental de CSS es el de separar la gestión del contenido, para lo que usamos HTML, de la gestión de la presentación, para lo que se usa CSS. Esto da lugar a varias ventajas:

- Los cambios visuales lo hacemos en el mismo lugar y se aplica a toda la página.
- No se duplican los estilos en distintos lugares.
- El contenido que se transmite es menor, lo que agiliza la carga de las páginas.
- A la hora de llevarlo a otras plataformas, como móviles, tablets, es más fácil de adaptarlo y crear versiones diferentes.

2.5.13. Algunas librerías destacables

- **Tensorflow**

Es una biblioteca de código abierto para cálculos numéricos, aprendizaje automático a gran escala, el *deep learning* y otras actividades de análisis estadístico y predictivo. Esta herramienta facilita la implementación de modelos de *machine learning* para los desarrolladores, asistiéndoles en el proceso de adquisición de datos, formulación de predicciones a gran escala y el afinamiento de los resultados.

Puede entrenar y ejecutar redes neuronales profundas para tareas como la clasificación de dígitos manuscritos, el reconocimiento de imágenes, la representación distribuida de palabras (word embedding) y el procesamiento del lenguaje natural (NLP). El código contenido en sus bibliotecas puede integrarse en cualquier aplicación para ayudarla a aprender a llevar a cabo estas tareas. Las aplicaciones de TensorFlow pueden ejecutarse tanto en CPU convencionales (unidades centrales de procesamiento) como en GPU

(unidades de procesamiento gráfico de alto rendimiento). Al ser desarrollado por Google, TensorFlow también funciona en las unidades de procesamiento tensorial (TPU) de la compañía, diseñadas específicamente para acelerar las operaciones de TensorFlow.

¿Por qué tensorflow y no pytorch?

- a. Mejor integración con producción y despliegue.
- b. Compatibilidad: TensorFlow incluye Keras como su API principal, facilitando la implementación de modelos de *Machine Learning*. Además, permite definir modelos de forma más intuitiva y rápida, en comparación con la API de bajo nivel de Pytorch. Por lo tanto, para una sintaxis sencilla sin sacrificar rendimiento, TensorFlow y Keras es una buena combinación.
- c. Mayor soporte en aplicaciones empresariales.
- d. Herramientas avanzadas para modelos complejos.
- e. Comunidad más grande y documentación detallada.

■ **Pandas**

Es una librería centrada en el manejo y el análisis de estructuras de datos. En este proyecto en particular, se usa para manipular archivos CSV y DataFrames.

■ **Scikit-learn (sklearn)**

Es una librería de python de código abierto usado para el aprendizaje automático. En este proyecto se ha usado tanto para la fase de selección de características, con RandomForestRegressor, como para la evaluación del modelo entrenado con mean_absolute_error, entre otras muchas funcionalidades.

■ **Matplotlib**

Es una librería de python orientada a la creación de gráficos, tanto en 2D como en 3D. Dicha biblioteca se usa en este proyecto para visualizar los resultados del entrenamiento de los modelos así como los escenarios *What-If*.

■ **Joblib**

Es una librería de python que permite que ejecutar tareas que requieren un alto esfuerzo computacional. En este proyecto se ha usado para guardar objetos como los modelos entrenados, scalars, etc. de forma que lo podamos reusar sin necesitar de volver a reentrenar desde 0.

- **Selenium + Webdriver Manager**

Se usan para el proceso de automatización de descarga de ficheros de la web cuando no existe una API. Webdriver manager instala el chromedriver oportuno sin necesidad de buscarlo manualmente. En este proyecto se ha usado para la fase de web scrapping.

- **React**

Es una librería de código abierto de Javascript usada para el desarrollo web. Fue desarrollada en 2011 por un trabajador de Facebook para solventar la necesidad de mejorar el rendimiento de dicha red social. Una vez se consolidó la librería, en 2013 la hicieron en código abierto.

Entre sus características principales encontramos que está basada en componentes, es decir, cada componente representa una interfaz como una pieza independiente, con su propia lógica. A diferencia de otras librerías, React nos permite renderizar y cargar las páginas más rápido gracias a que las carga desde los componentes.

La sintaxis que comunmente se usa para desarrollar aplicaciones con React es JSX (JavaScript XML), una extensión del lenguaje JavaScript original.

Por último, esta librería usa Virtual DOM. Esta característica hace que en lugar de usar el binding de datos para actualizar las vistas, use una copia del DOM que se compara con el original para mantener los cambios sincronizados.

- **Chart.js**

Es una librería de JavaScript de código abierto que nos permite la creación de múltiples tipos de gráficas a partir de datos.

3

Análisis experimental

En esta sección se exponen los numerosos experimentos realizados para evaluar el rendimiento y la capacidad de los modelos predictivos, así como, los resultados de cada uno de estos.

Los experimentos se dividirán según el rango de años de los datos, tanto meteorológicos como de los embalses, y la cantidad de embalses.

Los objetivos principales son:

1. Estudio de la evolución de los resultados de los modelos entrenados en función de dos ejes:
 - a. Histórico de años empleados para entrenar el modelo: 5, 15, 25 años.
 - b. Conjunto de embalses: 25, 50, 73 embalses.
2. Obtención de un modelo óptimo para su uso en la fase de generación de escenarios *what-if*.

Por otro lado, los experimentos se dividieron en dos etapas:

1. **Etapa 1 (fase de base y pruebas):** se entrenó el modelo sin el estudio previo de *feature selection*, con solo 5 años de datos y 73 embalses. En total, en esta etapa se realizaron 3 experimentos:
 - a. 5 años de datos y 25 embalses.
 - b. 5 años de datos y 50 embalses.
 - c. 5 años de datos y 73 embalses.

2. **Etapa 2:** se realizó un estudio previo de las variables tanto meteorológicas como de los embalses para ver su correlación y posteriormente se entrenó el modelo con 15 y 25 años de datos y 25, 50 y 73 embalses. En total, en esta etapa se realizaron 6 experimentos:

- a. 15 años de datos y 25 embalses.
- b. 15 años de datos y 50 embalses.
- c. 15 años de datos y 73 embalses.
- d. 25 años de datos y 25 embalses.
- e. 25 años de datos y 50 embalses.
- f. 25 años de datos y 73 embalses.

En total surgen 9 experimentos, de los cuales se incluirán los más interesantes: el de 5 años con 73 embalses y los 6 totales de la Etapa 2.

En cuanto al segundo objetivo, se ha usado el modelo más completo, con 25 años de datos y 73 embalses. Este ha servido para crear un nuevo conjunto de experimentos de escenarios *what-if*.

3.1. Entrenamiento de modelos predictivos

En esta sección se abordarán los detalles referentes al objetivo uno, entrenamiento de modelos predictivos.

3.1.1. Conjuntos de datos y pre-procesado

En este apartado se concretarán los detalles de los conjuntos de datos usados para los distintos experimentos.

- **Rango de datos:** 2000 – 2025
- **Disponibilidad de los datos:** la disponibilidad de los datos es diaria, aunque para el entrenamiento se han agregado mensualmente.
- **Variables originales:**

- **Meteorológicas:** precipitación, temperatura máxima, mínima y media, evapotranspiración, humedad relativa máxima, mínima y media, velocidad del viento media y máxima, dirección del viento y radiación.
 - **Embalses:** reserva (hm^3) y porcentaje de reserva.
- **Variables agregadas:** estas variables han sido las que se han usado para entrenar los modelos, ya que en los embalses no se notan los efectos de las variables meteorológicas de inmediato, sino tras semanas o meses.
 - Precipitación acumulada (6 meses): `precipitacion_sum_6m`
 - Humedad relativa media acumulada (6 meses): `humedad_media_sum_6m`
 - Velocidad media del viento (3 meses): `vel_viento_mean_3m`
 - Velocidad del viento acumulada (6 meses): `vel_viento_sum_6m`
 - Radiación global acumulada (3 meses): `radiacion_sum_3m`
 - Evapotranspiración media (2 meses): `evapotransp_mean_2m`
 - Temperatura media (2 meses): `temp_media_mean_2m`
 - Humedad relativa máxima del mes actual: `humedad_max`
- **Fuentes de datos:**
 - **Embalses:** [REDIAM – Embalses de Andalucía](#)
 - **Estaciones meteorológicas:** [Junta de Andalucía – Red RIA](#)
- **Normalización y codificación:**
 - **Variables numéricas:** se precisa la normalización de estas variables, ya que cada una se mide en una escala distinta, lo que provoca que el modelo sea mucho menos eficiente. Para ello se utiliza la función *MinMaxScaler* de la librería *sklearn*.
 - **Embalses:** originalmente están codificados con sus respectivos nombres, lo que impide que el modelo entienda su significado. Por ello, se ha convertido la columna del nombre de los embalses a números utilizando codificación *One-Hot* de la librería *sklearn*.

▪ **Configuración de hiperparámetros según el experimento:**

La tabla 1 muestra las configuraciones que se han ido probando para los experimentos.

ID Experimento	Años	Embalses	Feature selection	Ventana (meses)	Dropout	GRU Units	LR	Optimizador
E1	5	73	NO	60	0.5	64	5e-4	Adam
E2	15	25	SÍ	48	0.1	128	0.01	Adam
E3	15	50	SÍ	36	0.1	128	0.01	Adam
E4	15	73	SÍ	48	0.1	128	0.01	Adam
E5	25	25	SÍ	48	0.1	128	0.01	Adam
E6	25	50	SÍ	55	0.1	128	0.01	Adam
E7	25	73	SÍ	48	0.1	128	0.01	Adam

Tabla 1: Configuración de hiperparámetros por experimento

Descripción de hiperparámetros

- **Learning rate (LR):** tamaño de los pasos (*forward pass*) con los que el optimizador actualiza los pesos.
- **Ventana:** tiempo que recibe la GRU como entrada, es decir, el número de meses que se usa para predecir la reserva del mes siguiente.
- **Tasa de dropout:** porcentaje de neuronas que se apagan durante el entrenamiento en cada *forward-pass* para evitar el sobreajuste del modelo.
- **GRU Units:** número de neuronas en cada celda GRU, es decir, la cantidad de memoria interna que puede guardar la red.
- **Optimizador:** algoritmo que decide cuánto y cómo mover, en cada paso hacia atrás, los pesos para minimizar el error entre las predicciones y los valores reales. En todos los experimentos se ha usado el optimizador Adam (Adaptive Moment Estimation).

3.1.2. Feature selection (selección de características)

En este apartado abordaremos la fase de *feature selection* previa a los experimentos E2–E7.

El *feature selection* es un método para seleccionar las variables meteorológicas que tienen más peso a la hora de predecir la reserva de los embalses. Esto permite mejorar la precisión de los modelos, su capacidad de generalización y el rendimiento computacional.

Pasos seguidos durante esta fase

■ Preprocesamiento de los datos

- Se han unido los datos de las variables meteorológicas y de los embalses por fechas.
- Se han tratado los valores faltantes.
- Se han normalizado las variables necesarias.
- Se ha revisado que tanto los datos meteorológicos como los de los embalses tuvieran la misma agregación (diaria, semanal, etc.).

■ Análisis de correlaciones lineales

El análisis de correlaciones lineales es una técnica estadística que mide qué tan fuerte y en qué dirección se relacionan dos variables numéricas de manera lineal.

El objetivo en este caso fue cuantificar la relación *lineal* entre la **reserva de los embalses** y distintas **variables meteorológicas**. Para ello se calculó el coeficiente de correlación de **Pearson** (r) entre la serie de *reserva* y las series de precipitación, temperaturas, humedad, viento, radiación y evapotranspiración.

A continuación se realiza un estudio inicial de correlación entre las variables relevantes del conjunto de datos y la reserva. Las figuras 5, 6, 7, 8, 9, 10, 11 muestran las correlaciones de la reserva con la radiación, la precipitación, la evapotranspiración, la temperatura media, la humedad máxima, la temperatura máxima y temperatura mínima.

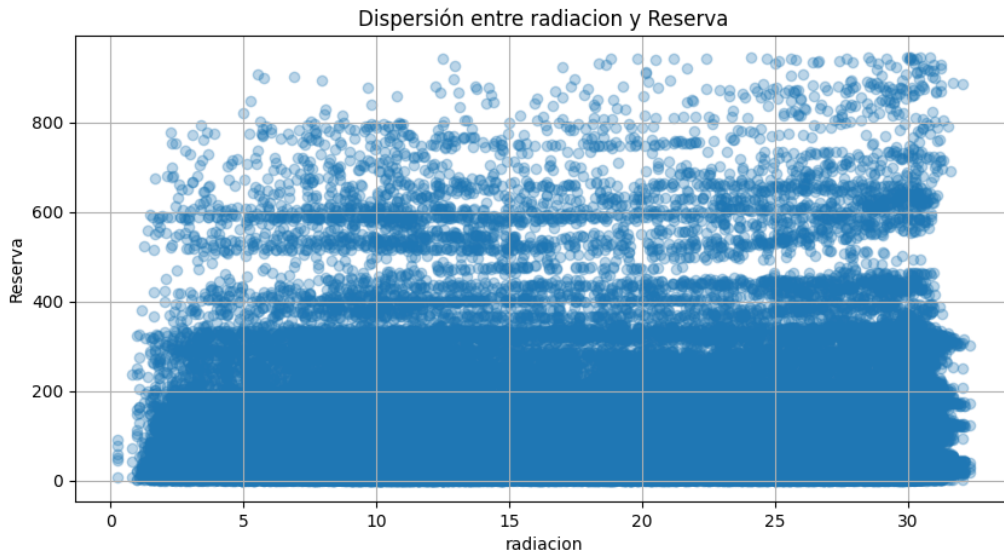


Figura 5: Correlación lineal entre reserva y radiación

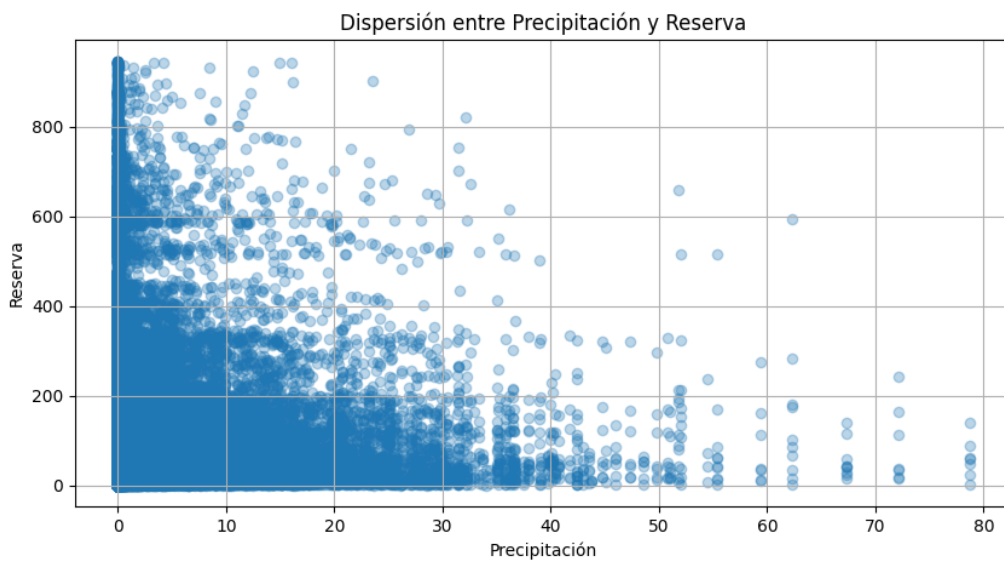


Figura 6: Correlación lineal entre reserva y precipitación

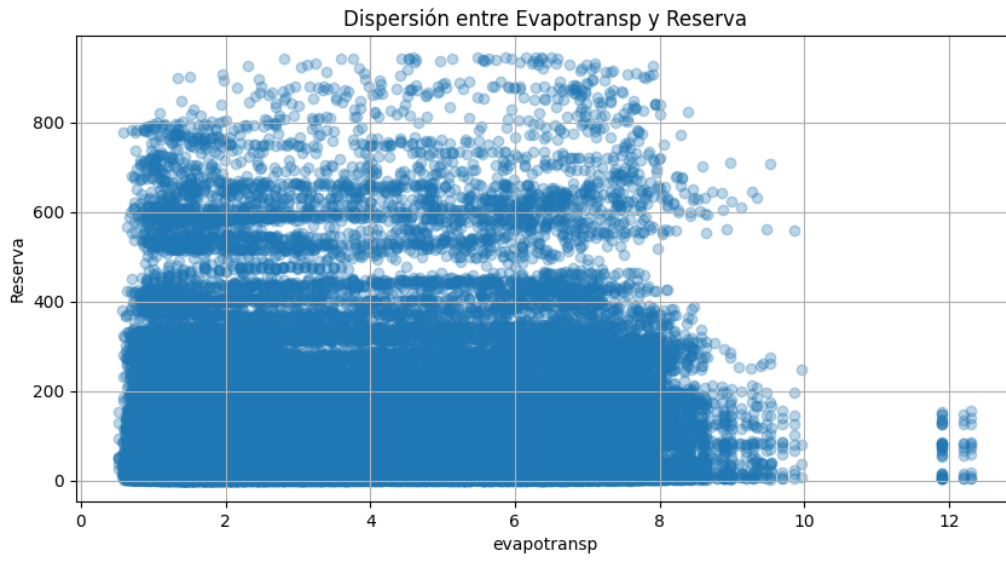


Figura 7: Correlación lineal entre reserva y evapotranspiración

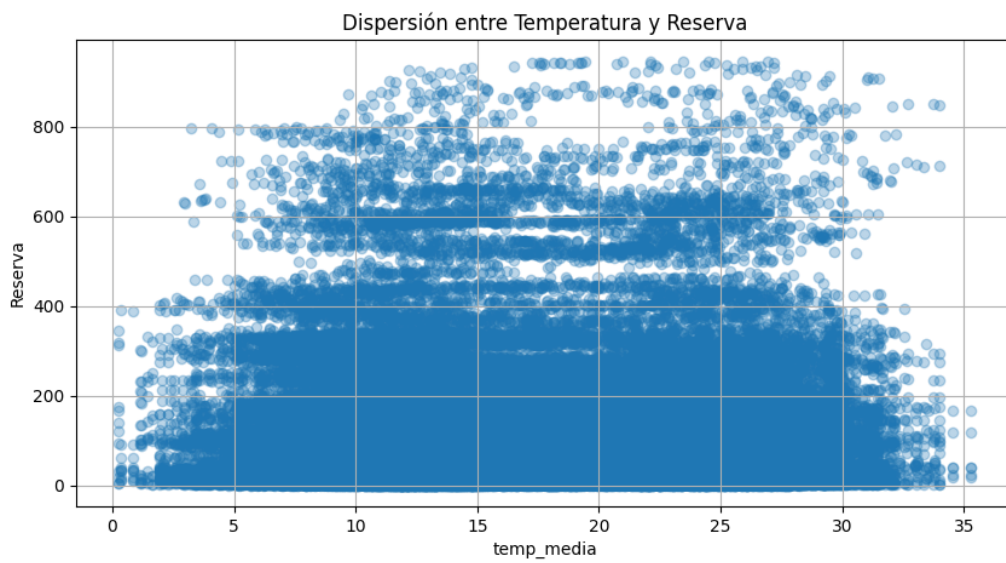


Figura 8: Correlación lineal entre reserva y temperatura media

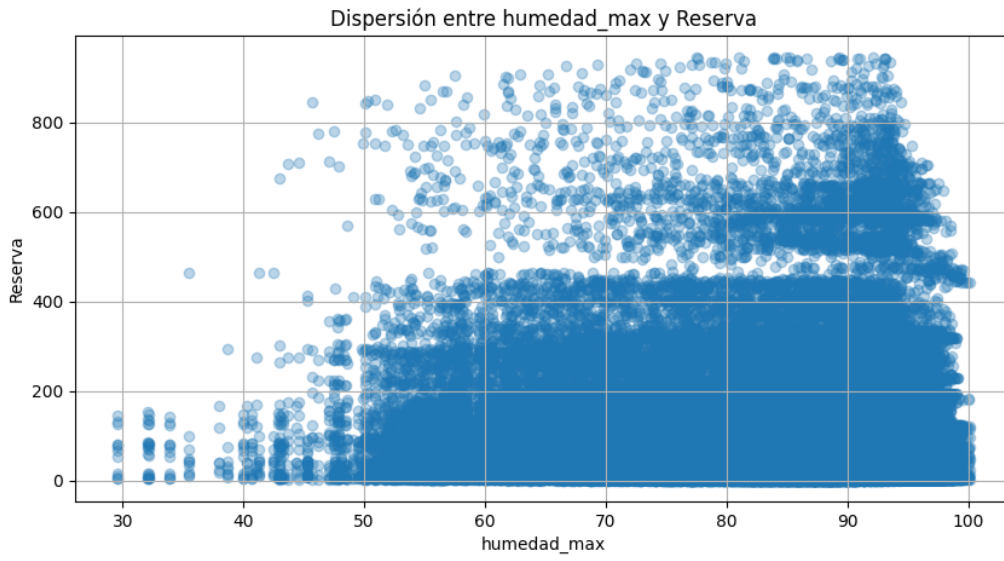


Figura 9: Correlación lineal entre reserva y humedad máxima

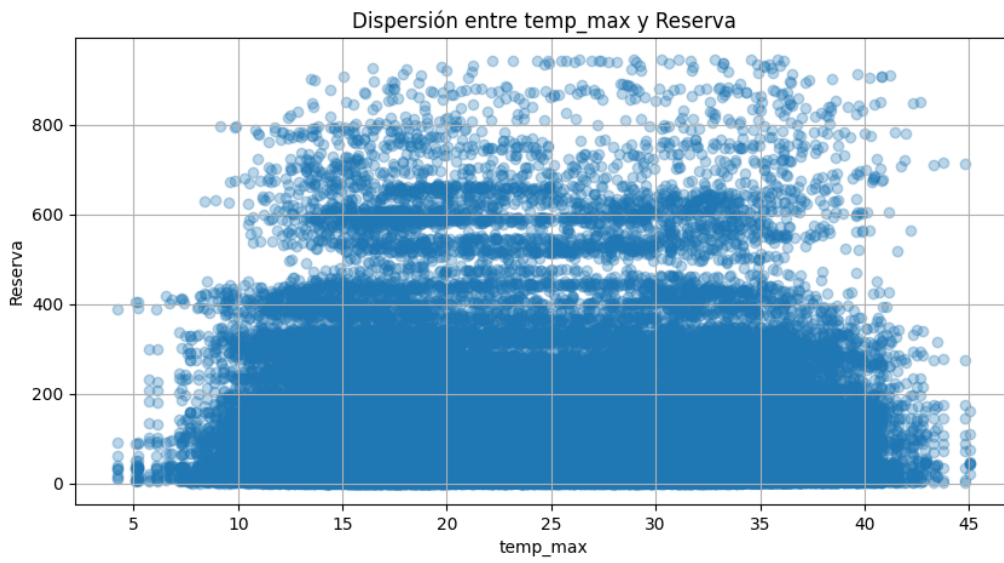


Figura 10: Correlación lineal entre reserva y temperatura máxima

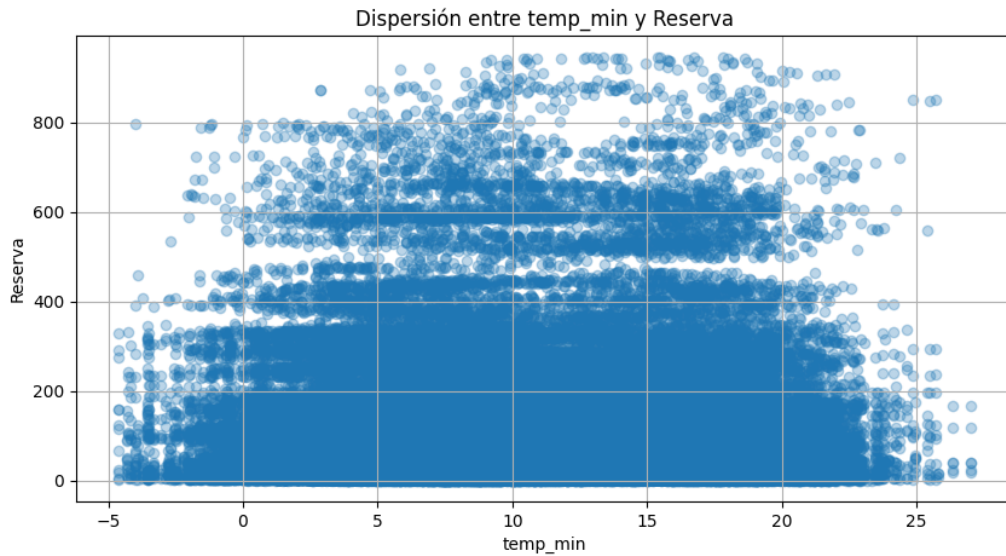


Figura 11: Correlación lineal entre reserva y temperatura mínima

Variable	Coef. Pearson ρ con reserva
Evapotranspiración	0.040479
Humedad relativa máxima	-0.010218
Precipitación	-0.002430
Radiación	0.044845
Temperatura máxima	0.025829
Temperatura media	0.014492
Temperatura mínima	-0.005367

Tabla 2: Correlación lineal entre variables meteorológicas y la reserva de los embalses.

La tabla 2 resume los resultados de la correlación. La correlación lineal instantánea entre la reserva de los embalses y las variables meteorológicas es, en general, **muy débil o nula**. Los coeficientes obtenidos se sitúan todos en el entorno de $|\rho| < 0,05$:

- **Ligeramente positivas (muy débiles):** radiación ($\rho \approx 0,045$), evapotranspiración ($\rho \approx 0,040$), temperatura máxima ($\rho \approx 0,026$) y temperatura media ($\rho \approx 0,014$).

- **Prácticamente nulas/ligeramente negativas:** precipitación ($\rho \approx -0,002$), humedad máxima ($\rho \approx -0,010$) y temperatura mínima ($\rho \approx -0,005$).

Estos valores tan bajos indican que: el efecto de la meteorología sobre la reserva es *diferido* (acumulación y propagación de agua en cuenca y explotación de presas), hay *no linealidades* y umbrales (p. ej., eventos intensos frente a lloviznas), y la *gestión humana* (desembalses, demandas, trasvases) añade variabilidad que rompe correlaciones lineales simples.

Las correlaciones a t no capturan el fenómeno; por ello es necesario: trabajar con *agregados* y *desfases* (medias/sumas a 2–6 meses) y emplear modelos secuenciales *no lineales* (GRU/LSTM) que aprendan dependencias temporales. Esto justifica las variables agregadas que se usan después (p. ej., *precipitacion_sum_6m*, *evapotransp_mean_2m*) y el uso de una GRU para explotar efectos retardados y patrones no lineales.

■ Análisis de correlaciones cruzadas

En nuestro Trabajo Final de Grado se ha llevado a cabo un análisis de correlación cruzada para ver cuán relacionadas están las variables con distintos desfases temporales (*lags*). Los resultados se presentan en la Figura 12 y la Figura 13

Este método es crucial en este caso, ya que los embalses no ven las consecuencias de la meteorología de manera inmediata, sino con desfases temporales.

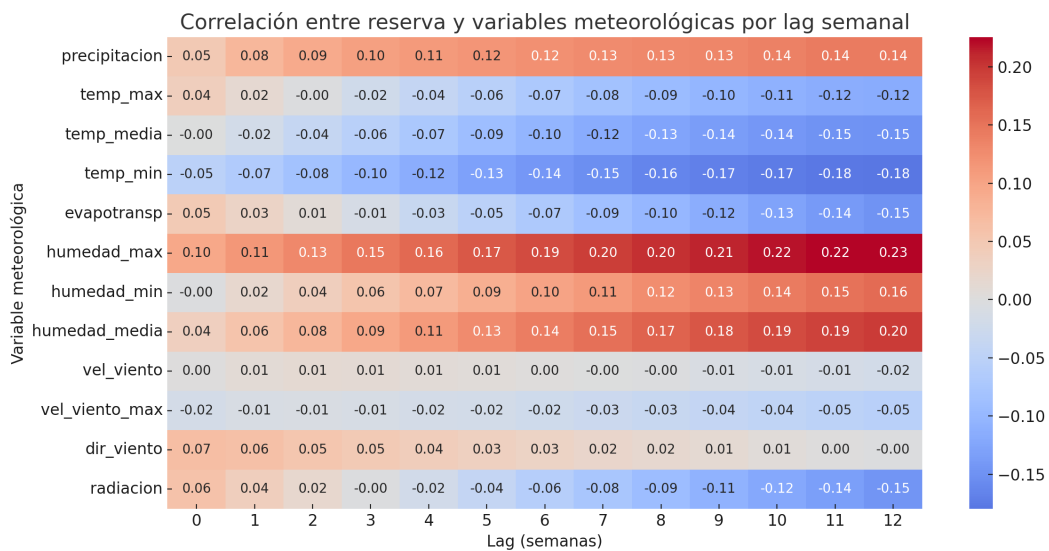


Figura 12: Heatmap de correlaciones cruzadas por provincia

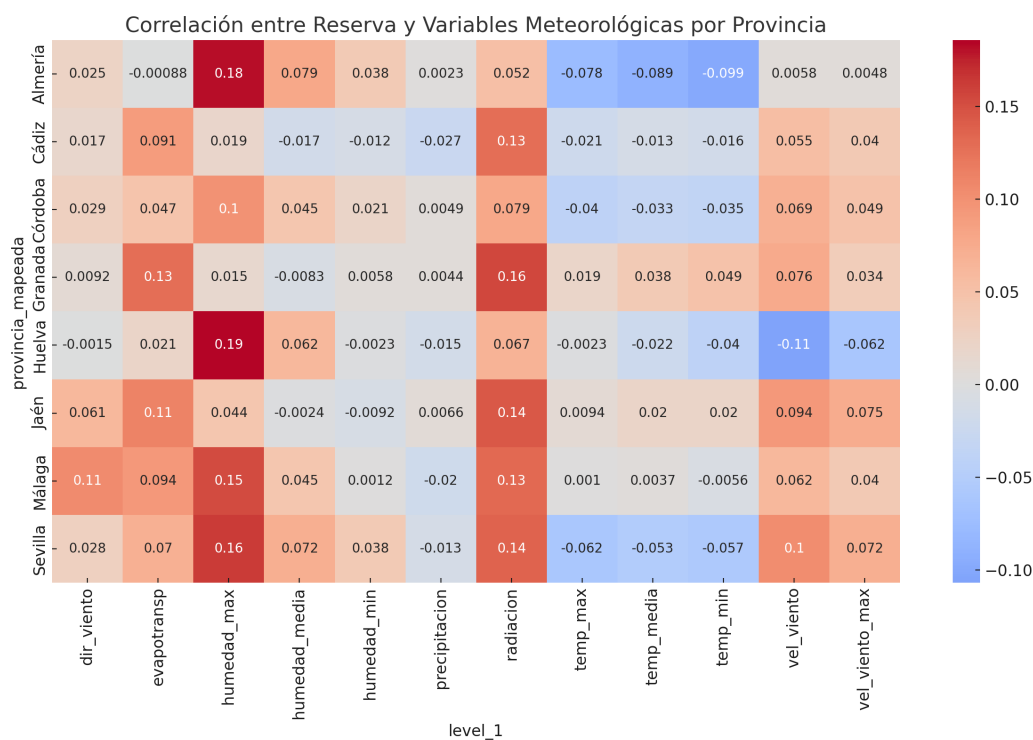


Figura 13: Heatmap correlaciones cruzadas diarias

Las correlaciones máximas en un desfase temporal de 12 semanas (3 meses) se muestran en la tabla 3.

Variable	Lag óptimo (semanas)	Correlación
Precipitación	12	0.1423
Temperatura máxima	12	-0.1180
Temperatura media	12	-0.1501
Temperatura mínima	12	-0.1798
Evapotranspiración	12	-0.1475
Humedad relativa máxima	12	0.2256
Humedad relativa mínima	12	0.1576
Humedad relativa media	12	0.1983
Velocidad viento media	12	-0.0188
Velocidad viento máxima	12	-0.0548
Dirección del viento	0	0.0668
Radiación global	12	-0.1457

Tabla 3: Correlación cruzada entre la reserva y las variables meteorológicas (lag óptimo)

- **Evaluación de la importancia de las variables con *Random Forest***

Random Forest es un modelo de aprendizaje automático que indica la importancia que tiene cada variable en el modelo.

Los resultados de este análisis se muestran en la figura 14:

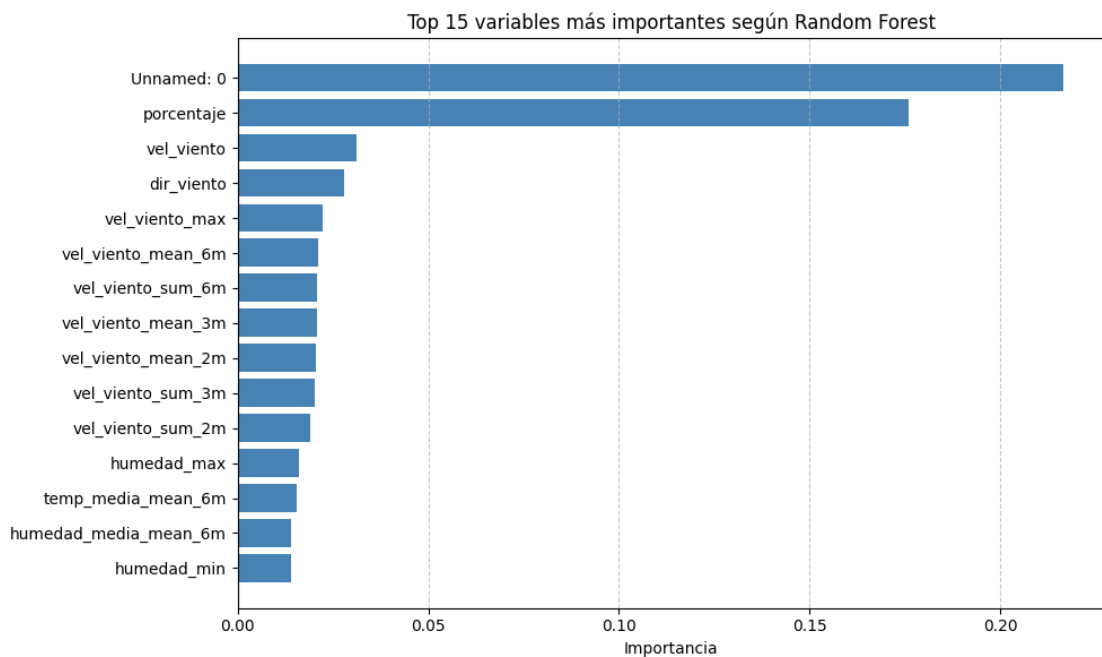


Figura 14: Top 15 variables más importantes según *Random Forest*.

- **Selección final de variables**

Según los resultados de ambos métodos, se han elegido las siguientes variables por su alta correlación o importancia dentro del modelo. Véase la tabla 4:

Variable	Justificación
precipitacion_sum_6m	Alta correlación cruzada + importancia.
humedad_media_sum_6m	Mayor correlación cruzada.
vel_viento_mean_3m	Muy alta importancia en <i>Random Forest</i> .
vel_viento_sum_6m	Alta importancia en <i>Random Forest</i> .
radiacion_sum_3m	Aporte relevante según <i>Random Forest</i> .
evapotransp_mean_2m	Mejora el modelo según <i>Random Forest</i> .
temp_media_mean_2m	Representa efecto térmico acumulado.
humedad_max	Alta importancia directa en <i>Random Forest</i> .

Tabla 4: Variables seleccionadas con mayor relevancia según análisis cruzado y *Random Forest*.

3.1.3. Pasos del entrenamiento de los modelos

A continuación se detallan los pasos generales seguidos por todos los experimentos de la fase de entrenamiento de los modelos.

1. Preparación y reproducibilidad.

Se activa el uso eficiente de GPU (crecimiento dinámico de memoria) y se fijan semillas en Python, NumPy y TensorFlow (42) para hacer reproducibles los resultados.

2. Adquisición y sincronización de datos.

Se extraen las series de *embalses* (reserva y porcentaje) y las series *meteorológicas*. Se corrigen problemas de codificación de texto y se mapea cada embalse a su provincia para poder integrar ambas fuentes por fecha y ámbito geográfico. Las estaciones se agregan por provincia (estadístico robusto) para obtener una única serie meteorológica representativa por fecha.

3. Limpieza y selección de columnas.

Se conservan únicamente las variables relevantes (tiempo, nombre del embalse, reserva y métricas meteorológicas) y se tratan los nulos en campos críticos. Se guarda un conjunto limpio que servirá como base del modelado.

4. Agregación temporal.

Como los efectos meteorológicos sobre la reserva son retardados, las series se *agregan a frecuencia mensual*. Las variables de *flujo* (p. ej., precipitación) se agregan por suma; las de *estado* (p. ej., temperaturas, humedades) por media.

5. Ingeniería de características temporales.

Se generan estadísticas móviles y acumuladas en distintas ventanas (W meses), por ejemplo medias móviles o acumulados a 2–6 meses, para capturar efectos diferidos y suavizar ruido.

6. Selección de variables.

Se combinan criterios estadísticos y de aprendizaje: correlaciones (simples y cruzadas con desfase) entre cada variable y la reserva; e importancia de variables con un modelo de referencia (*Random Forest*). A partir de ambos, se fija un subconjunto final de predictores.

7. Preparación del conjunto de aprendizaje.

- *Escalado de entradas*: normalización conjunta de las variables numéricas.
- *Identificador del embalse*: codificación categórica (p. ej., one-hot) para permitir que el modelo aprenda diferencias estructurales entre embalses.
- *Transformación del objetivo*: se estabiliza la varianza de la *reserva* (p. ej., transformación logarítmica) y se aplica un escalado *por embalse* para respetar rangos propios.

8. Construcción de ventanas secuenciales.

Se formulan pares (X, y) con ventanas deslizantes de longitud W (meses) y horizonte $H = 1$ mes: $X_t = \{\text{features}_{t-W+1}, \dots, \text{features}_t\}$, $y_{t+1} = \text{reserva}_{t+1}$. Se realiza una partición temporal entrenamiento/validación (sin barajado) para no romper la estructura de serie.

9. Arquitectura del modelo.

Se emplea una red neuronal recurrente para series temporales (GRU/LSTM) con:

- una o varias *capas recurrentes* que procesan la secuencia;
- *regularización* (p. ej., dropout) para mitigar sobreajuste;
- un *cabezal denso* que proyecta el estado final a la salida escalar (reserva).

La pérdida es de regresión (MAE o MSE) y se usa un optimizador de gradiente adaptativo.

10. Estrategia de entrenamiento.

Se entrena durante varias épocas con:

- *parada temprana* sobre la pérdida de validación, restaurando los mejores pesos;
- *ajuste automático de la tasa de aprendizaje* cuando la validación se estanca;
- registro de curvas *loss/val_loss* para diagnóstico.

11. Evaluación y visualización.

Para validar interpretación, se invierte la transformación del objetivo (incluido el escalado por embalse) y se reportan métricas por embalse y globales: MAE, RMSE, R^2 y MAPE. Se generan gráficos de *predicción vs. real* por embalse y una gráfica global con varias series concatenadas.

12. Persistencia y reproducibilidad.

Se guardan modelo, escaladores y codificador, así como los conjuntos derivados y las figuras. De este modo se garantiza la reproducción del entrenamiento y su uso posterior (p. ej., en escenarios *what-if*).

3.2. Entrenamiento y validación de escenarios *What-If*

En esta sección se describirá cómo, partiendo del modelo óptimo E7 (25 años + 73 embalses), se construyen y evalúan los distintos escenarios *What-If*.

Para ponernos en contexto, primero debemos saber que es un escenario *What-If*. Los escenarios *What-If* son un enfoque de análisis usado para evaluar cómo ciertos cambios en las variables de entrada pueden afectar los resultados de un sistema. Estos escenarios se usan para la toma de decisiones estratégicas en áreas como la economía, la planificación urbana, y en este caso, en la gestión del agua en embalses.[45]

En el caso de la gestión del agua, estos escenarios nos ayudan a prever y gestionar riesgos usando modelos basados en datos históricos y predicciones:

- Optimizar la distribución del agua según distintas condiciones climáticas.

- Planificar medidas de prevención ante sequías o inundaciones.
- Tomar decisiones basadas en datos en lugar de suposiciones.

Para comenzar con el análisis de dichos escenarios, primero necesitamos un modelo predictivo que permita ajustar las variables de entrada y observar cómo cambia la salida.[46]

Algunos de los modelos de *machine learning* que se pueden usar son:

- **Regresión Lineal o Polinómica:** para estimar tendencias a partir de datos históricos.
- **Redes Neuronales (LSTM, RNN, GRU):** para predicciones más avanzadas de series temporales.
- **Modelos de Árboles de Decisión:** para analizar varios factores simultáneamente.

El objetivo de este paso es observar el impacto en las predicciones si modificamos las variables de entrada.

Ejemplos de escenarios:

- ¿Qué pasaría si la precipitación disminuye un 30 % en los próximos 3 meses?
- ¿Cómo afectaría un aumento del consumo agrícola en un 20 %?

3.2.1. Pasos del desarrollo de los escenarios *What-If*

A continuación se van a enumerar los pasos específicos que se han seguido para generar los escenarios entrando en detalle respecto al código python que se ha desarrollado:

1. Carga de objetos

En este paso hemos cargado los objetos que generamos en la fase de entrenamiento del modelo, como el propio modelo GRU, los escaladores de entrada y los escaladores específicos de reserva, y el objeto codificador (*OneHotEncoder*) para los embalses.

Con este paso nos aseguramos de que lo que estamos usando para generar los escenarios son los mismos que en la fase de entrenamiento, haciendo todo el proceso mucho más eficiente.

2. Construcción del dataframe para el escenario

Con el dataframe original que usamos para entrenar el modelo, hacemos una copia pero con las columnas modificadas y multiplicadas según el escenario que queremos simular.

3. Configuración de las ventanas de entrada

Haciendo uso de los escaladores, el codificador se han generado las ventanas temporales de 48 meses. Estas ventanas avanzan de mes en mes, respetando la forma de entrada que usó el GRU al entrenar.

4. Desescalado a hm^3

Para cada embalse se ha desescalado a hm^3 para que los resultados se puedan comparar con los valores reales.

5. Cálculo de métricas

Este paso es común tanto en la generación de escenarios *What-If* como en el entrenamiento de los modelos.

Las métricas que hemos usado son MAE, RMSE, MAPE y R^2 .

Nos ayudan a medir, en este caso, las diferencias entre el escenario simulado y la climatología observada. Además, se generan dichas medidas tanto globalmente como por embalses.

6. Guardado de los archivos

En este paso simplemente guardamos los resultados de los procesos anteriores, como los datos ponderados respecto a los reales, las gráficas generadas de los escenarios nuevos etc, en formato csv y png.

3.3. Resultados

En esta sección se mostrarán los resultados de todos los experimentos realizados durante la fase general de *machine learning*, tanto la parte de entrenamiento de los modelos como la de los escenarios *What-If*.

Además, se reflejarán los experimentos que han salido bien y los que han salido mal.

3.3.1. Resultados del entrenamiento de los modelos

Todos estos experimentos constan de gráficas del modelo global que se entrena y de gráficas por embalses. Como el total de embalses asciende a 73, no se adjuntarán imágenes de cada uno de ellos.

Además, se ha sacado una gráfica comparativa de las pérdidas en la validación y las pérdidas en el entrenamiento.

E1: 5 años y 73 embalses

Al comienzo de esta fase se entrenaron modelos con pocos años de datos y todos los embalses pero sin estudio previo para la selección de características (*Feature selection*). Este es el caso del experimento 1.

Este experimento no tiene gráficas por embalses, solo el resultado del modelo completo ya que simplemente era de prueba para los posteriores.

El resultado se presenta en la figura 15.

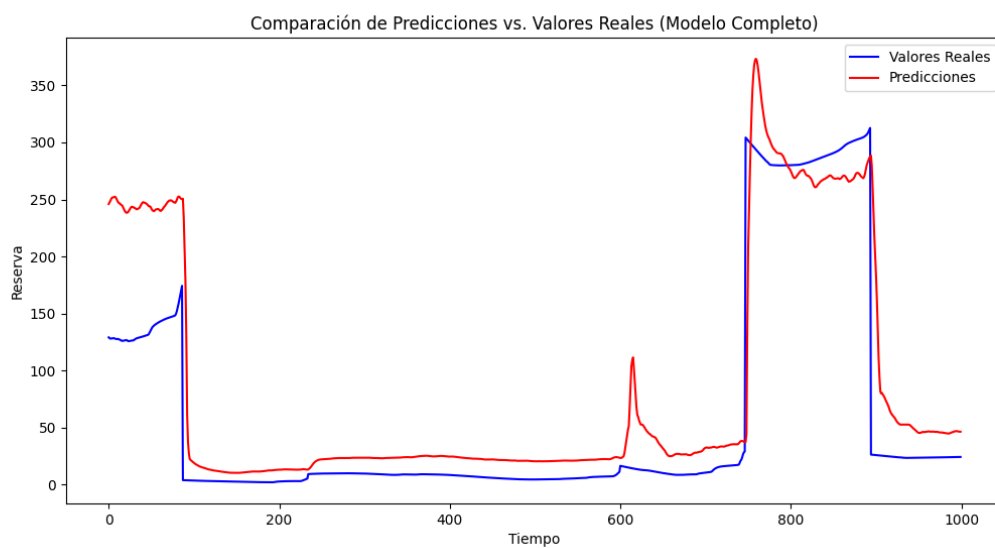


Figura 15: Gráfica modelo completo E1

E2: 15 años y 25 embalses

A partir de este experimento, incluido este, todos incluyen la fase previa de selección de características (*Feature selection*).

Una vez validadas las configuraciones con conjuntos reducidos, se procedió a entrenar el modelo predictivo utilizando información de 25 embalses andaluces distintos. El objetivo era comprobar la capacidad generalizadora del modelo al aumentar la diversidad espacial y temporal de los datos. El modelo utilizado fue una red basada en dos capas GRU combinadas con capas densas intermedias, diseñada para captar tanto patrones secuenciales como relaciones no lineales entre las variables meteorológicas y el volumen de reserva. Tras distintas pruebas con configuraciones de hiperparámetros, se establecieron los siguientes ajustes finales:

- **Ventana temporal:** 48 pasos temporales (equivalente a 48 meses).
- **Unidades GRU:** 128 y 64 en cada capa respectivamente.
- **Capas densas:** [64, 16, 1] con activaciones relu y linear.
- **Dropout:** 0.1 tras cada capa GRU.
- **Learning rate:** 0.01 (valor ajustado manualmente tras experimentar con valores menores).
- **Optimizador:** Adam
- **Criterio de parada:** EarlyStopping con patience=35 sobre la val_loss.

El resultado de aplicar estas configuraciones se puede observar en la figura 16. Esta figura es la representación gráfica del modelo completo.

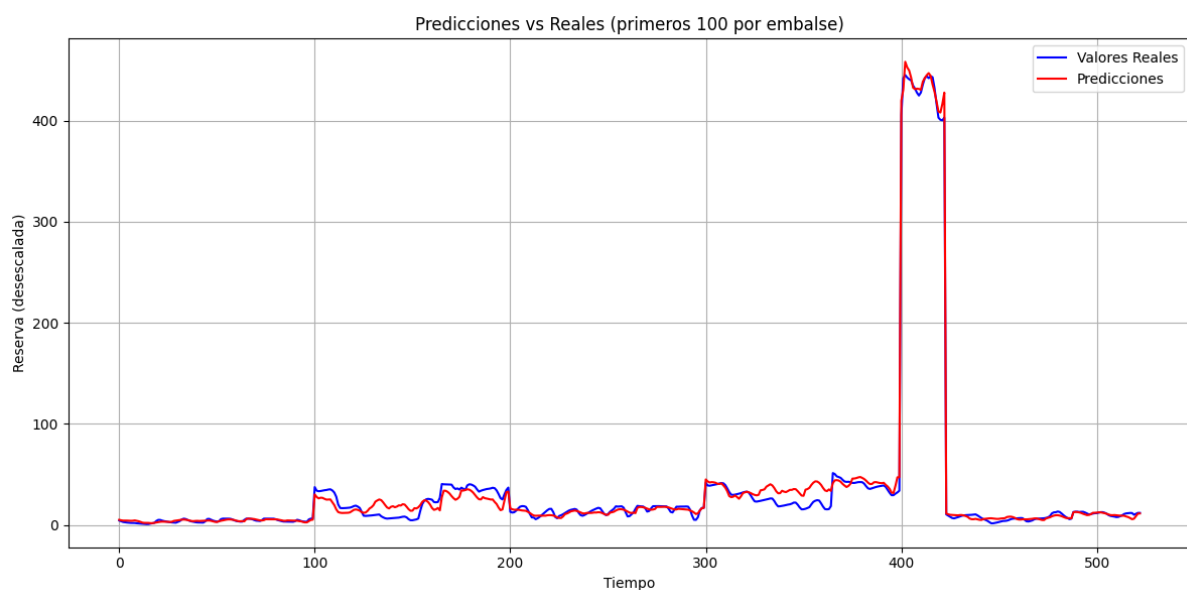


Figura 16: Gráfica modelo completo E2

Las figuras 17 y 18 son las representaciones gráficas por embalse.

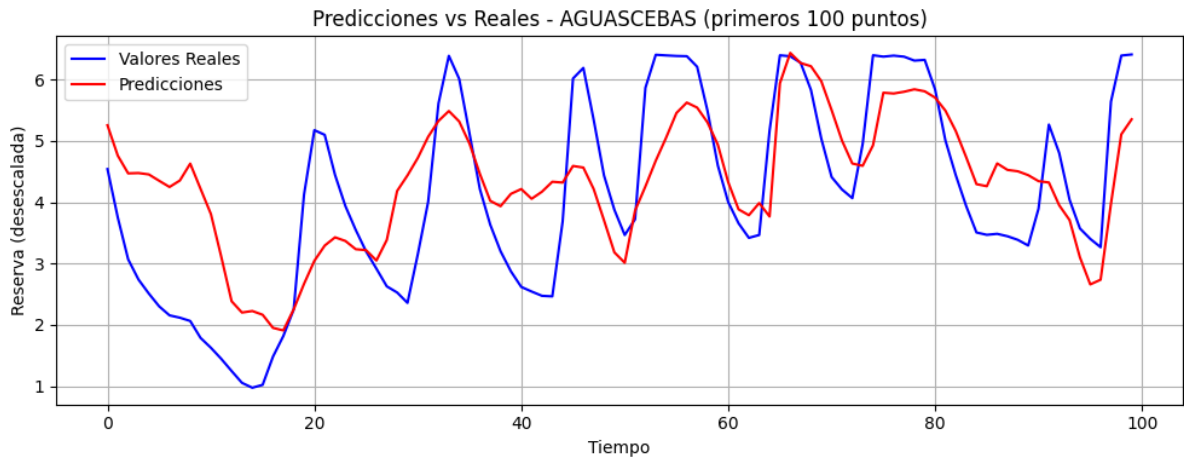


Figura 17: Gráfica embalse de AGUASCEBAS E2

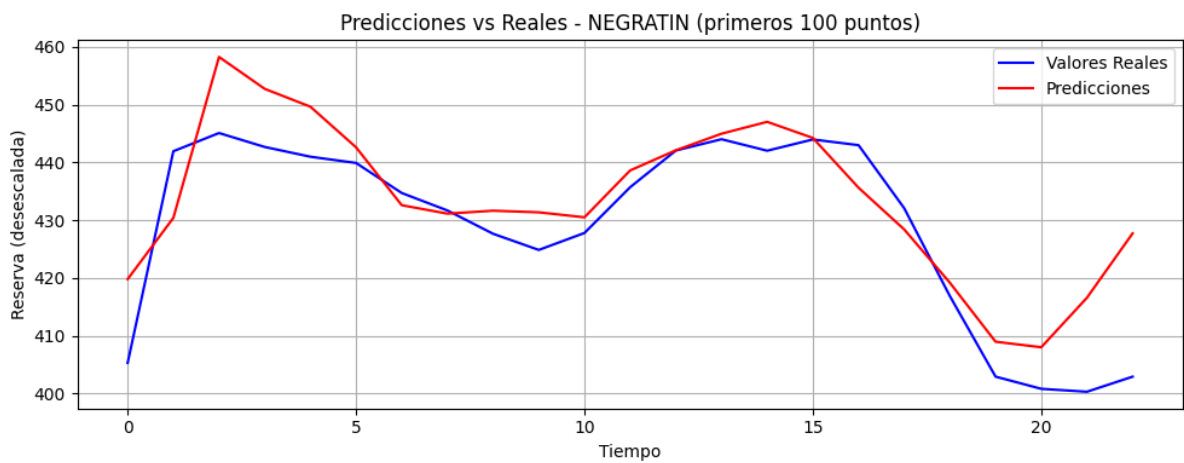


Figura 18: Gráfica embalse de NEGRATIN E2

Y por último, la figura 19 muestra la gráfica de las pérdidas en validación y en entrenamiento.

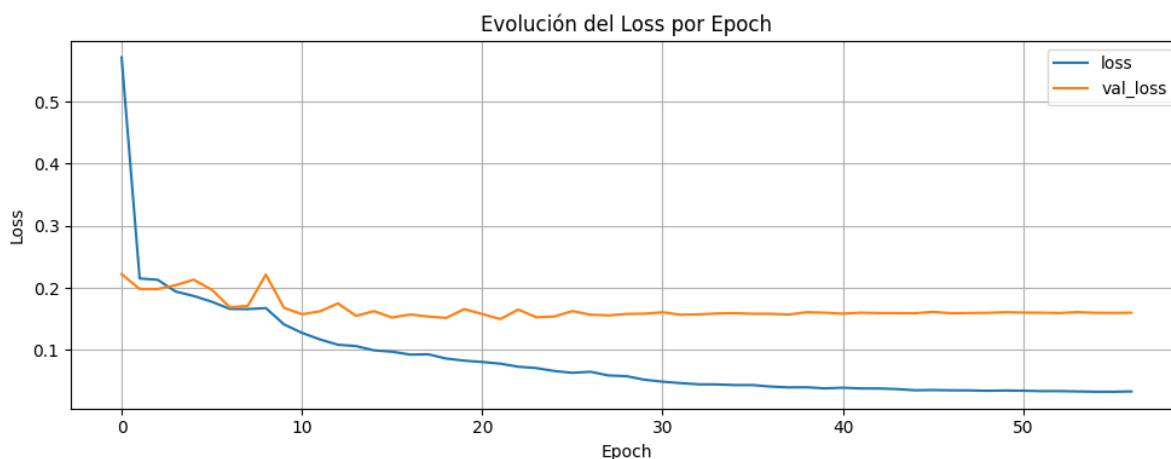


Figura 19: Gráfica de pérdidas E2

La tabla 5 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 19.

Embalse	MAE	RMSE	R^2	MAPE (%)
Aguascebas	0.91	1.09	0.50	28.90 %
Negratín	6.66	8.95	0.67	1.58 %

Tabla 5: Resultados por embalse E2

El ajuste final obtenido presenta una gran capacidad para aproximarse a la serie real, incluyendo cambios bruscos y picos de reserva. En la figura se muestra la comparación entre las predicciones generadas por el modelo y los valores reales desescalados para los primeros 100 registros de cada embalse. Se observa un comportamiento notablemente fiel a las tendencias reales, con una mejora significativa respecto a configuraciones anteriores, especialmente en las zonas de mayor pendiente o variabilidad.

E3: 15 años y 50 embalses

- **Ventana temporal:** 48 pasos temporales (equivalente a 48 meses).
- **Unidades GRU:** 128 y 64 en cada capa respectivamente.
- **Capas densas:** [64, 16, 1] con activaciones *relu* y *linear*.
- **Dropout:** 0.1 tras cada capa GRU.

- **Learning rate:** 0.01 (valor ajustado manualmente tras experimentar con valores menores).
- **Optimizador:** Adam.
- **Criterio de parada:** *EarlyStopping* con *patience*=35 sobre la *val_loss*.

El resultado de aplicar estas configuraciones se puede observar en la figura 20. Esta figura es la representación gráfica del modelo completo. El modelo reproduce bien los ciclos de llenado-vaciado y los cambios de tendencia. Los mayores desajustes aparecen en picos muy pronunciados y transiciones rápidas.

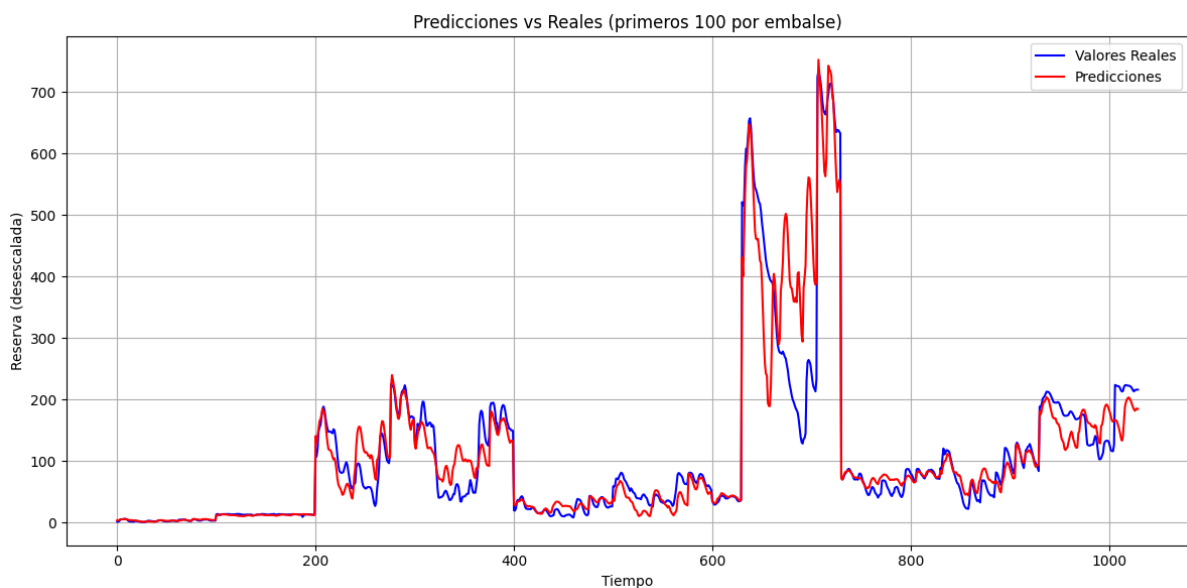


Figura 20: Gráfica modelo completo E3

Las figuras 21, 22 y 23 son las representaciones gráficas por embalse.

En la figura 21 la predicción sigue la estacionalidad y las subidas/bajadas principales; se observan pequeñas desviaciones en máximos.

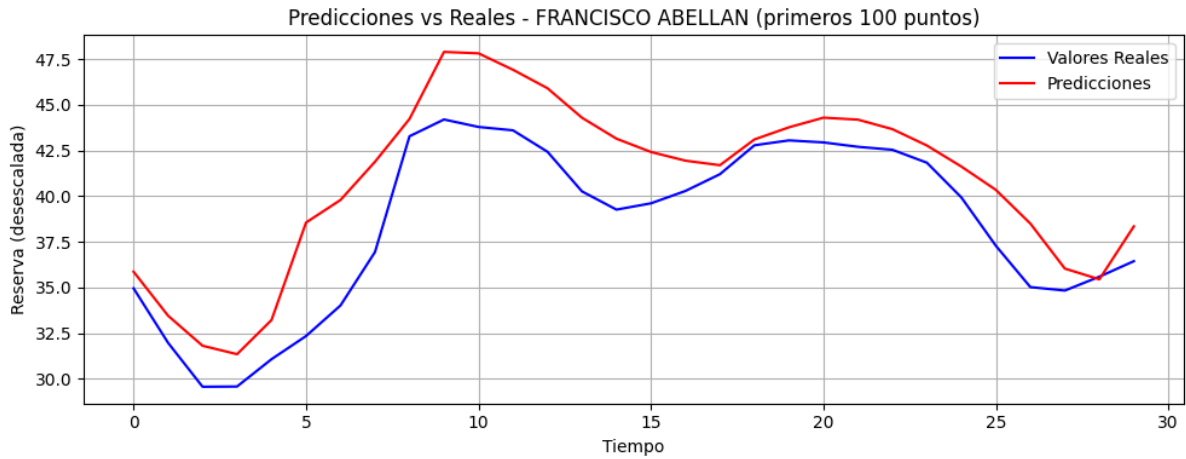


Figura 21: Gráfica embalse FRANCISCO ABELLAN E3

En la figura 22 la forma general y la amplitud se capturan. Los picos más agudos son los que más cuestan.

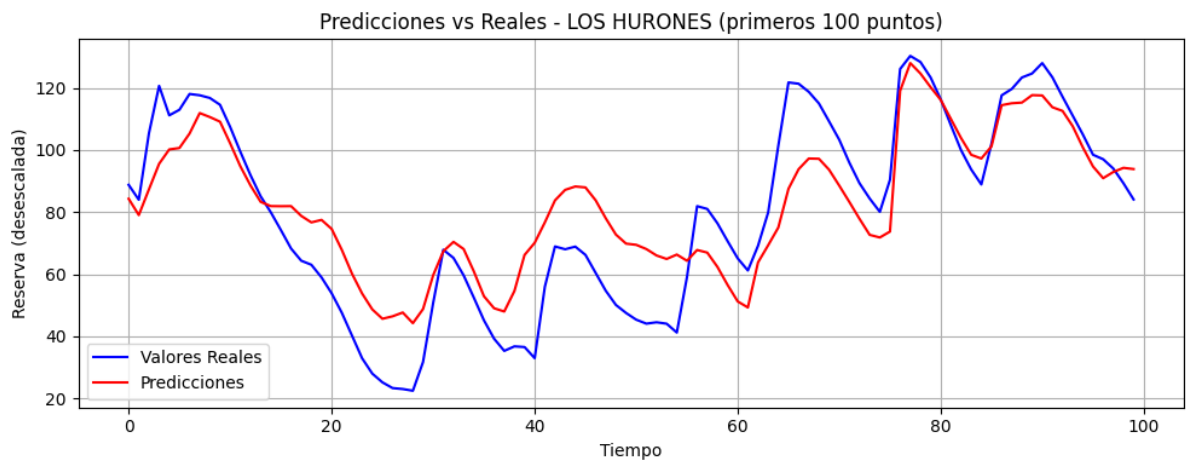


Figura 22: Gráfica embalse LOS HURONES E3

En la figura 23 el porcentaje incrementa cuando los niveles reales son muy bajos por lo que conviene priorizar MAE/RMSE.

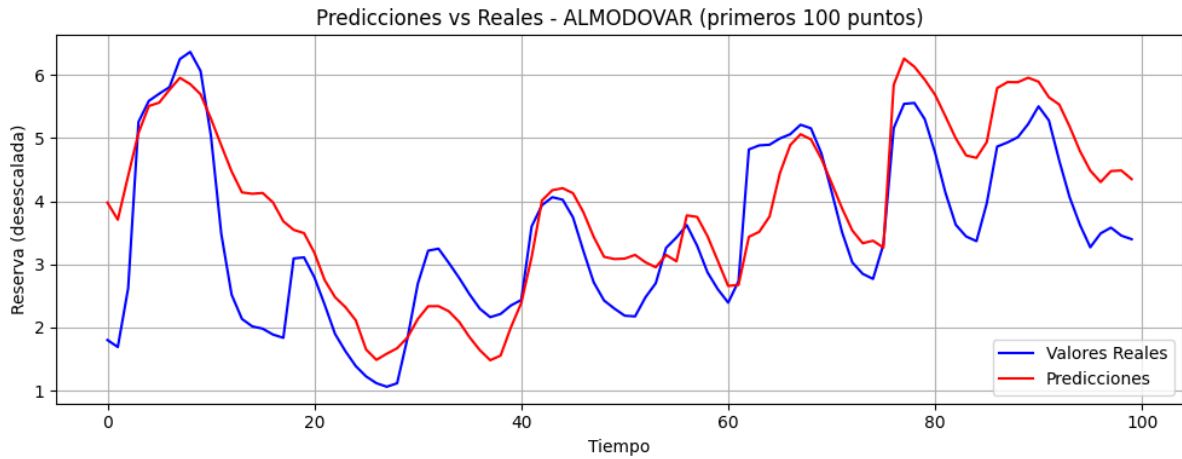


Figura 23: Gráfica embalse ALMODOVAR E3

Y por último, la figura 24 muestra la gráfica de las pérdidas en validación y en entrenamiento. Las curvas de entrenamiento y validación descienden y se estabilizan sin separarse en exceso. No hay sobreajuste y la parada temprana fija el mejor punto.

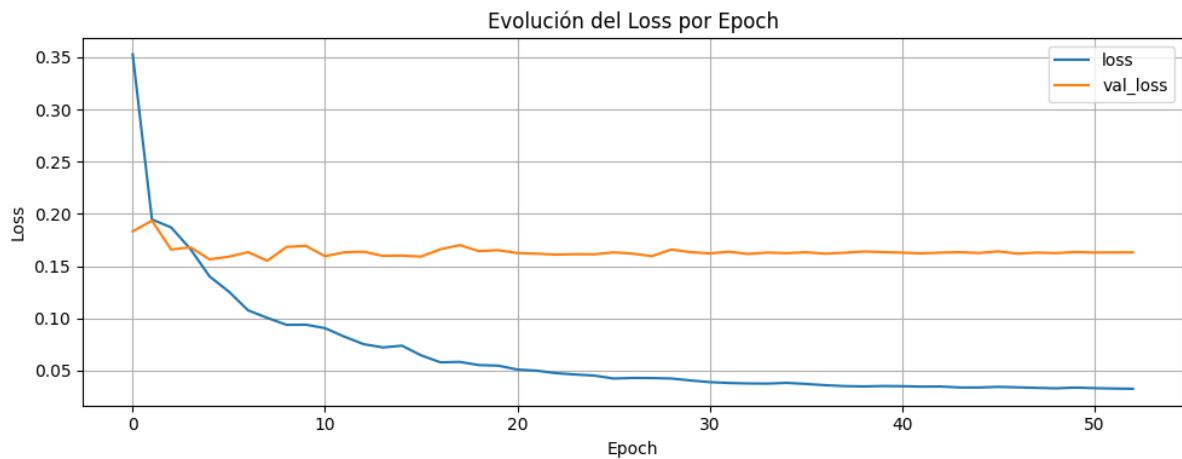


Figura 24: Gráfica de pérdidas E3

La tabla 6 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 24.

Embalse	MAE	RMSE	R^2	MAPE (%)
Francisco Abellan	2.37	2.85	0.60	6.35 %
Los Hurones	12.70	15.25	0.74	21.47 %
Almodovar	0.71	0.87	0.53	24.94 %

Tabla 6: Resultados por embalse E3

E4: 15 años y 73 embalses

- **Ventana temporal:** 48 pasos temporales (equivalente a 48 meses).
- **Unidades GRU:** 128 y 64 en cada capa respectivamente.
- **Capas densas:** [64, 16, 1] con activaciones *relu* y *linear*.
- **Dropout:** 0.1 tras cada capa GRU.
- **Learning rate:** 0.01 (valor ajustado manualmente tras experimentar con valores menores).
- **Optimizador:** Adam.
- **Criterio de parada:** *EarlyStopping* con *patience*=35 sobre la *val_loss*.

El resultado de aplicar estas configuraciones se puede observar en la figura 25. Esta figura es la representación gráfica del modelo completo. Al ampliar cobertura espacial aumenta algo la dispersión, pero se conservan ciclos y tendencias.

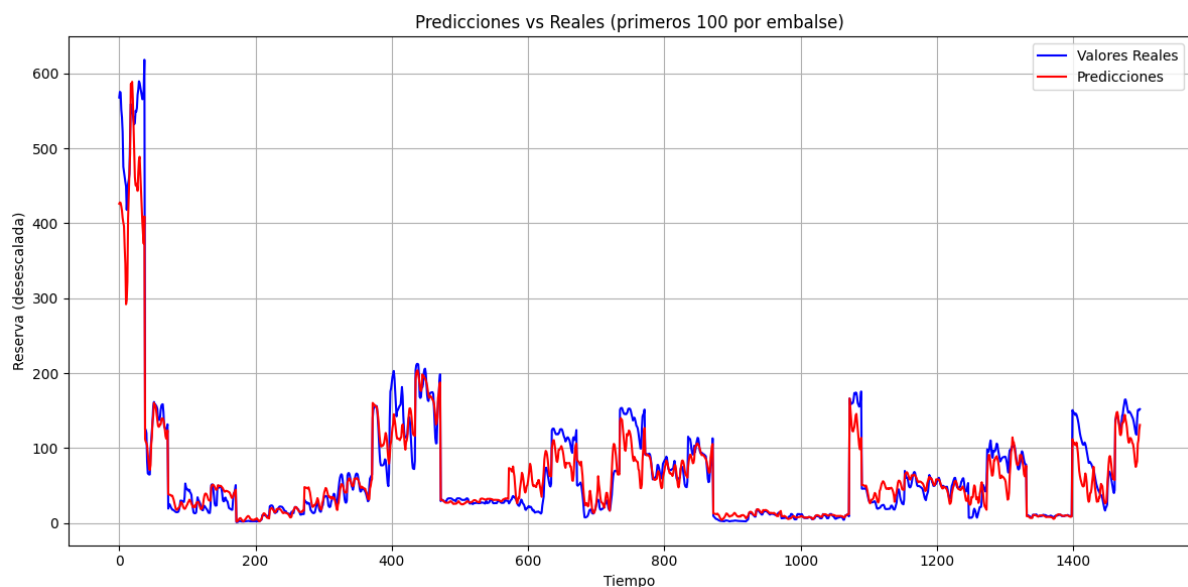


Figura 25: Gráfica modelo completo E4

Las figuras 26, 27, 28 y 29 son las representaciones gráficas por embalse.

En la figura 26 se captan bien valles y picos con ligeros desfases temporales en máximos.

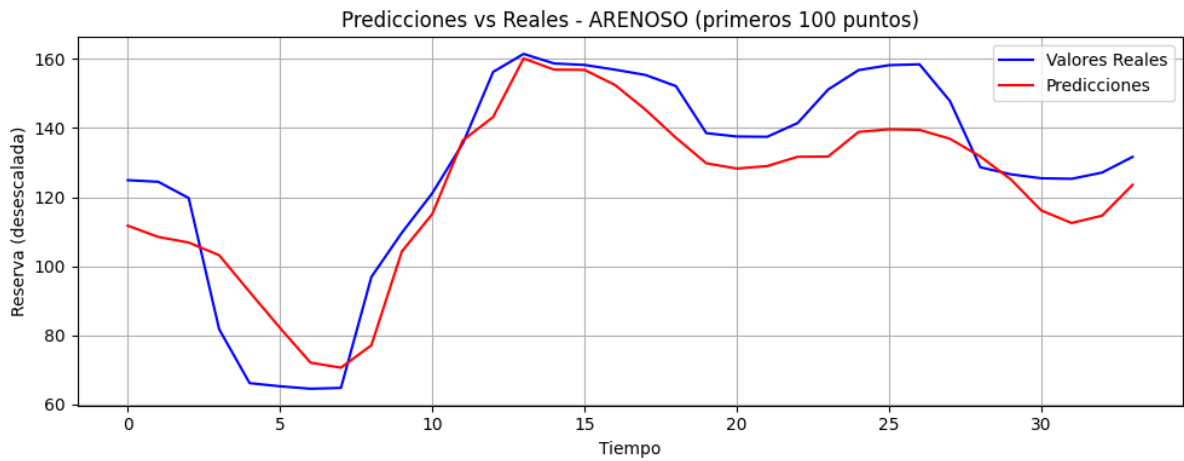


Figura 26: Gráfica embalse ARENOSO E4

En la figura 27 la forma de la serie está bien reproducida. El MAPE elevado se explica por periodos con valores reales pequeños.

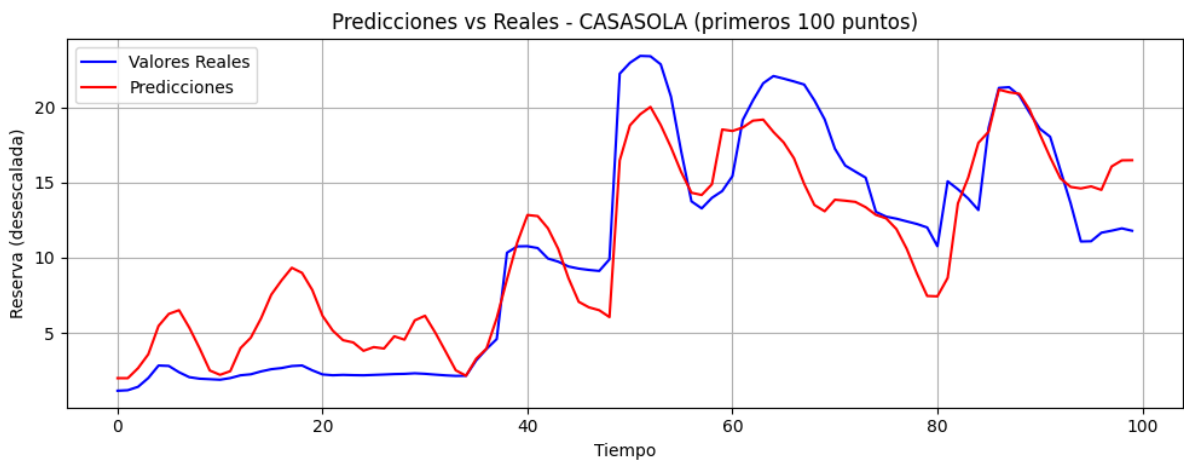


Figura 27: Gráfica embalse CASASOLA E4

En la figura 28 se siguen las tendencias, pero en episodios de cambios bruscos aparecen sobreestimaciones y mayor error relativo.

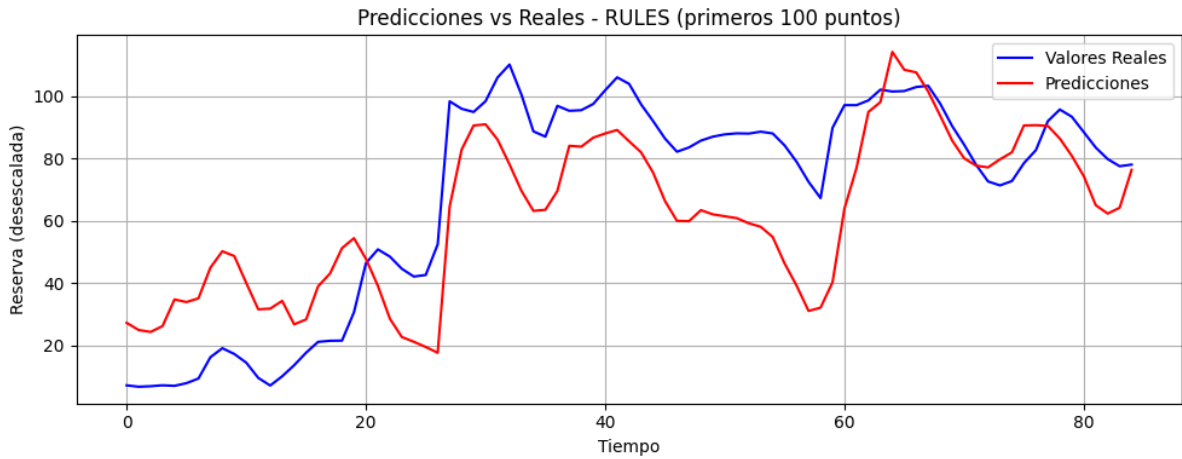


Figura 28: Gráfica embalse RULES E4

En la figura 29 el modelo refleja la dinámica general y los escalones principales. Los extremos muestran mayor error.

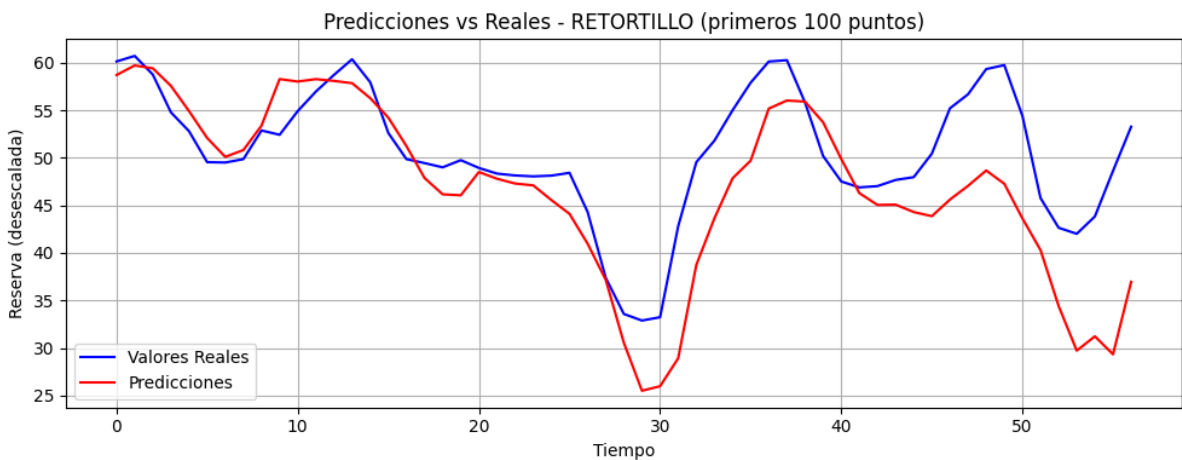


Figura 29: Gráfica embalse RETORTILLO E4

Y por último, la figura 30 muestra la gráfica de las pérdidas en validación y en entrenamiento. Ambas decrecen y convergen con una separación moderada. Generaliza de forma razonable en el conjunto ampliado.

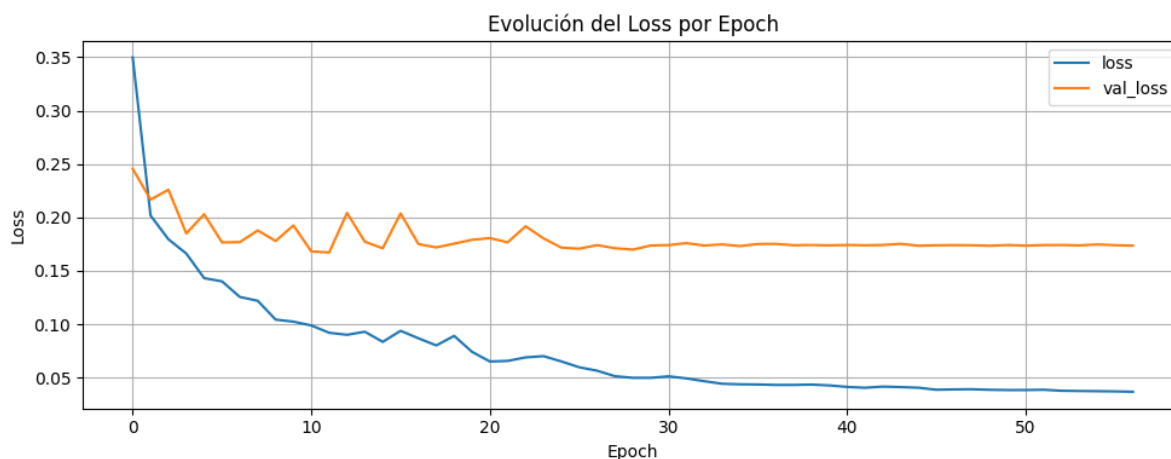


Figura 30: Gráfica pérdidas E4

La tabla 7 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 30.

Embalse	MAE	RMSE	R^2	MAPE (%)
Arenoso	10.84	12.65	0.82	9.61 %
Casasola	2.45	3.04	0.82	48.18 %
Rules	19.06	21.85	0.59	66.03 %
Retortillo	4.84	6.63	0.70	9.88 %

Tabla 7: Resultados por embalse E4

E5: 25 años y 25 embalses

- **Ventana temporal:** 48 pasos temporales (equivalente a 48 meses).
- **Unidades GRU:** 128 y 64 en cada capa respectivamente.
- **Capas densas:** [32, 16, 1] con activaciones *relu* y *linear*.
- **Dropout:** 0.1 tras cada capa GRU.
- **Learning rate:** 0.01.
- **Optimizador:** Adam.
- **Criterio de parada:** *EarlyStopping* con *patience*=45 sobre la *val_loss*.

El resultado de aplicar estas configuraciones se puede observar en la figura 31. Esta figura es la representación gráfica del modelo completo. Se aprecian trayectorias coherentes y amplitudes plausibles.

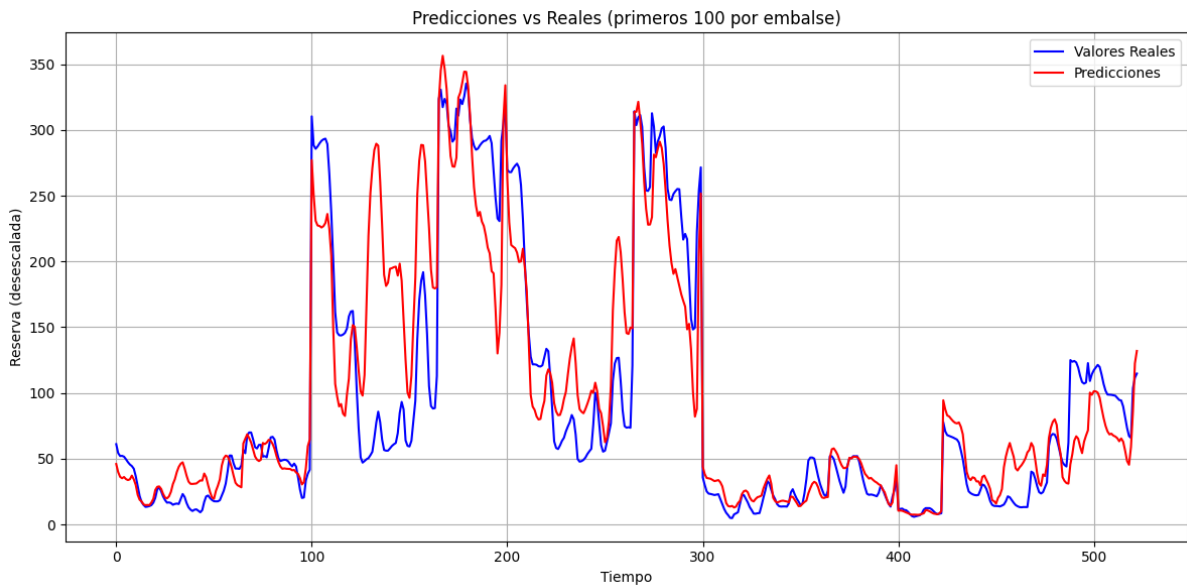


Figura 31: Gráfica modelo completo E5

Las figuras 32, 33 y 34 son las representaciones gráficas por embalse.

En la figura 32 la serie predicha no recoge bien la variabilidad. Posible influencia de regulación específica o cambios de régimen no observados.

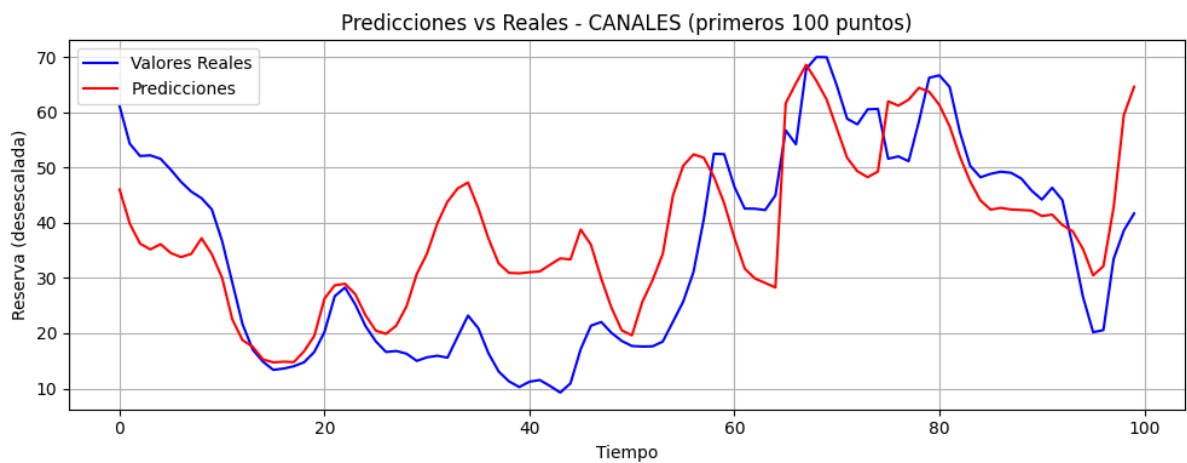


Figura 32: Gráfica embalse CANALES E5

En la figura 33 se reproduce la amplitud y los giros principales. Los errores se concentran en rampas muy rápidas.

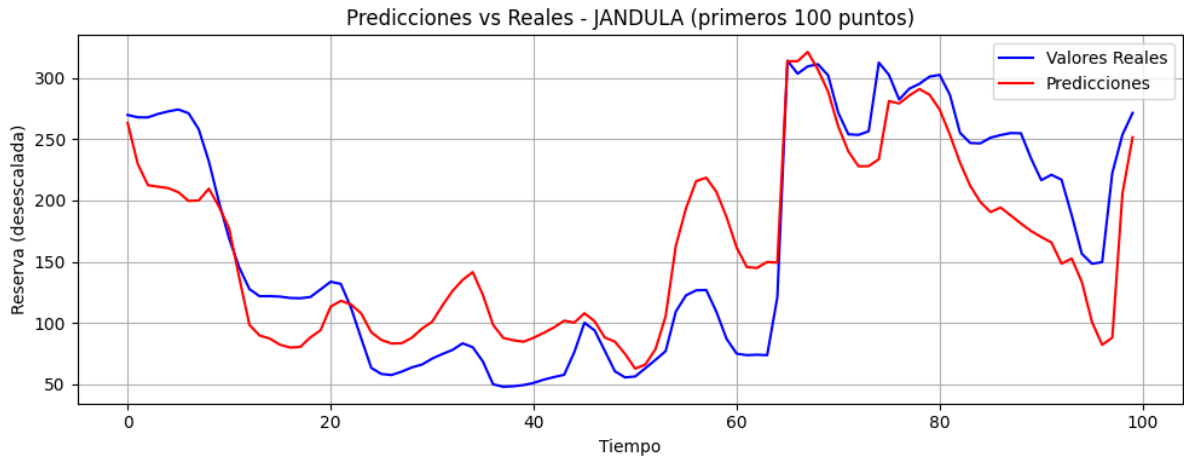


Figura 33: Gráfica embalse JANDULA E5

En la figura 34 se capta la estacionalidad y la mayor parte de crestas, con cierto alisado en picos.

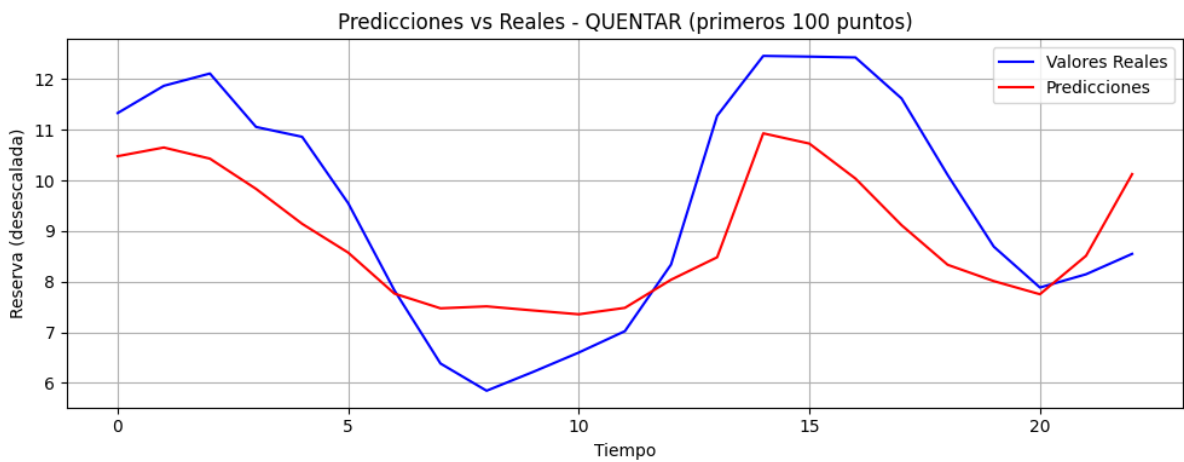


Figura 34: Gráfica embalse QUENTAR E5

Y por último, la figura 35 muestra la gráfica de las pérdidas en validación y en entrenamiento. La pérdidas son estables con poca brecha entre entrenamiento y validación. La parada temprana evita sobreajuste.

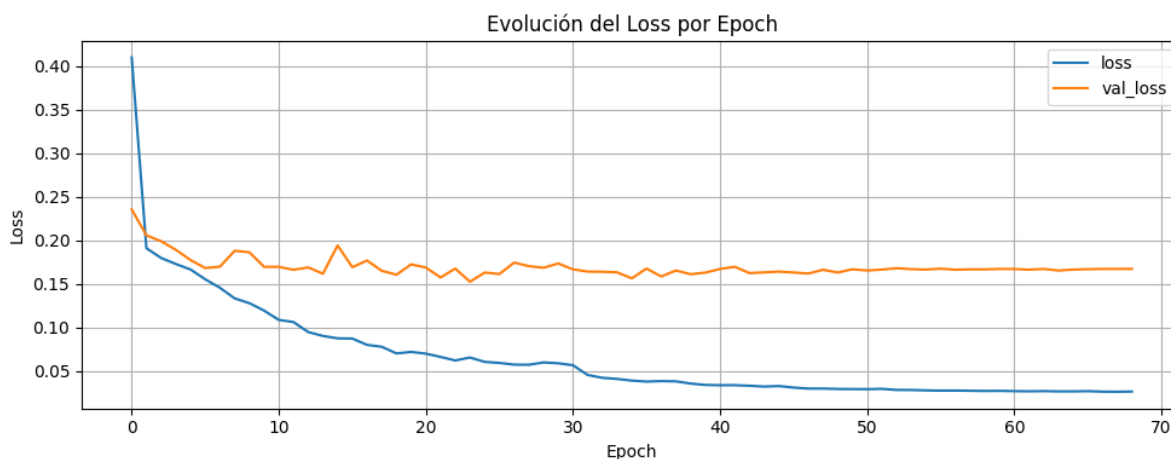


Figura 35: Gráfica de pérdidas E5

La tabla 8 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 35.

Embalse	MAE	RMSE	R^2	MAPE (%)
Canales	13.92	16.79	-0.09	40.14 %
Jándula	29.17	37.05	0.80	25.53 %
Quéntar	1.25	1.45	0.57	12.96 %

Tabla 8: Resultados por embalse E5

E6: 25 años y 50 embalses

- **Ventana temporal:** 55 pasos temporales (equivalente a 55 meses).
- **Unidades GRU:** 128 y 64 en cada capa respectivamente.
- **Capas densas:** [32, 16, 1] con activaciones *relu* y *linear*.
- **Dropout:** 0.1 tras cada capa GRU.
- **Learning rate:** 0.01.
- **Optimizador:** Adam.
- **Criterio de parada:** *EarlyStopping* con *patience*=45 sobre la *val_loss*.

El resultado de aplicar estas configuraciones se puede observar en la figura 36. Esta figura es la representación gráfica del modelo completo.

En la figura 36 la diversidad de embalses introduce variabilidad, pero la señal principal se mantiene.

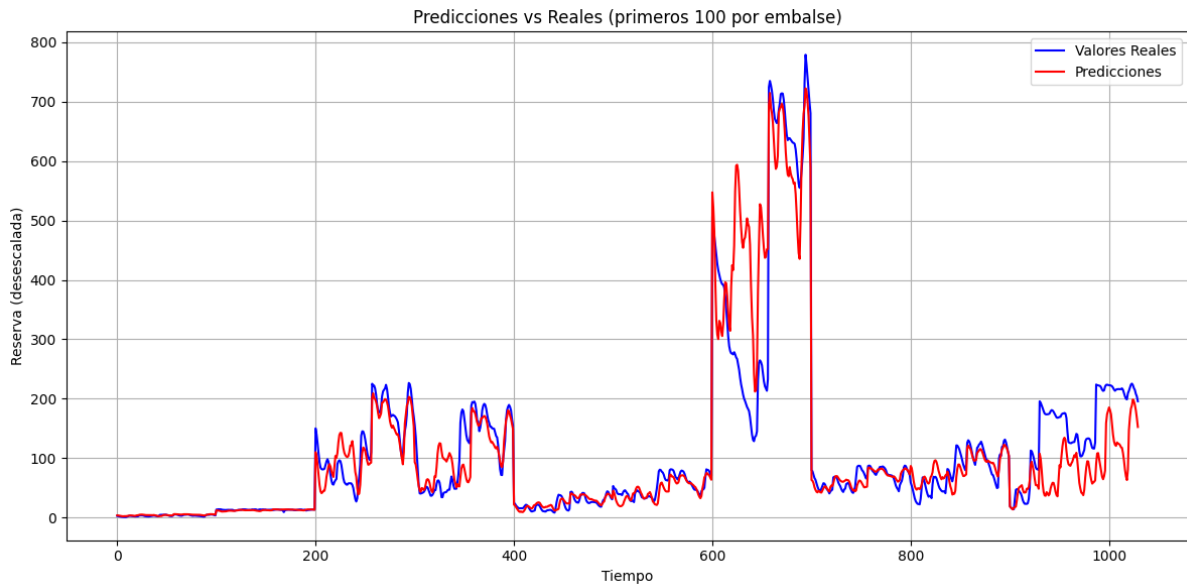


Figura 36: Gráfica modelo completo E6

Las figuras 37, 38, 39 y 40 son las representaciones gráficas por embalse.

En la figura 37 se aprecia un buen seguimiento de tendencias, picos atenuados (suavizado) y leves retrasos en máximos.

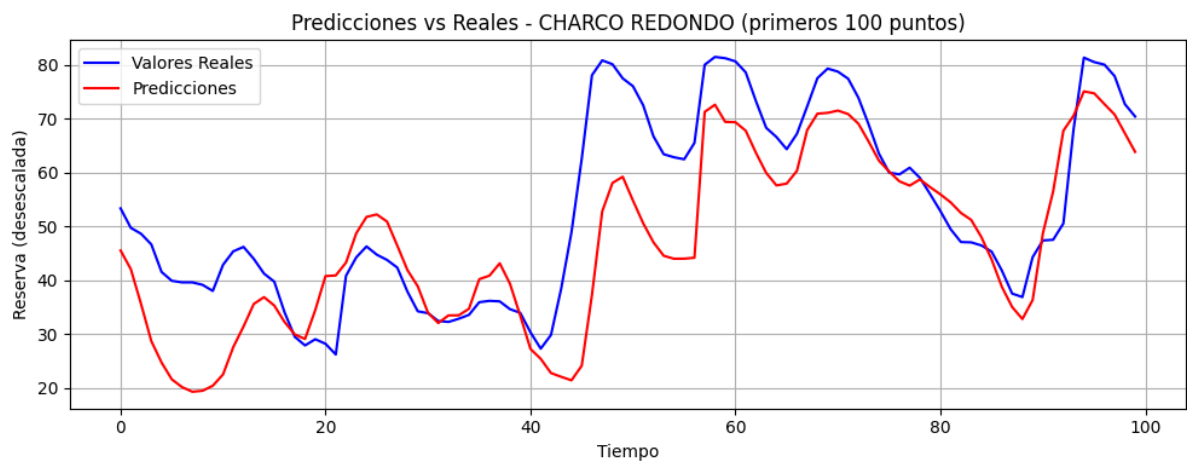


Figura 37: Gráfica embalse CHARCO REDONDO E6

En la figura 38 el modelo recoge las oscilaciones con algún desfase y errores mayores en cambios abruptos.

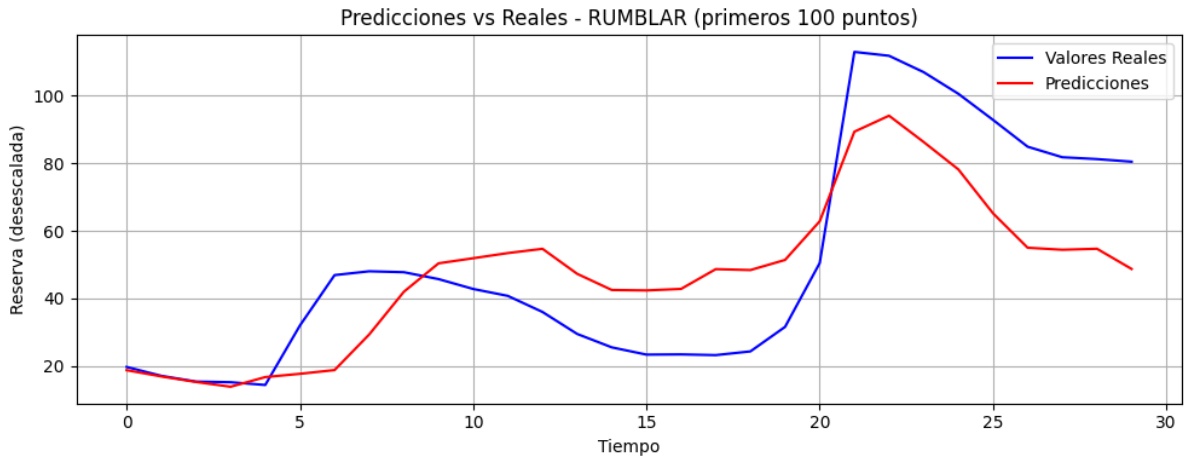


Figura 38: Gráfico embalse RUMBLAR E6

En la figura 39 se captan los niveles relativos y la forma general. Hay mayor incertidumbre en extremos.

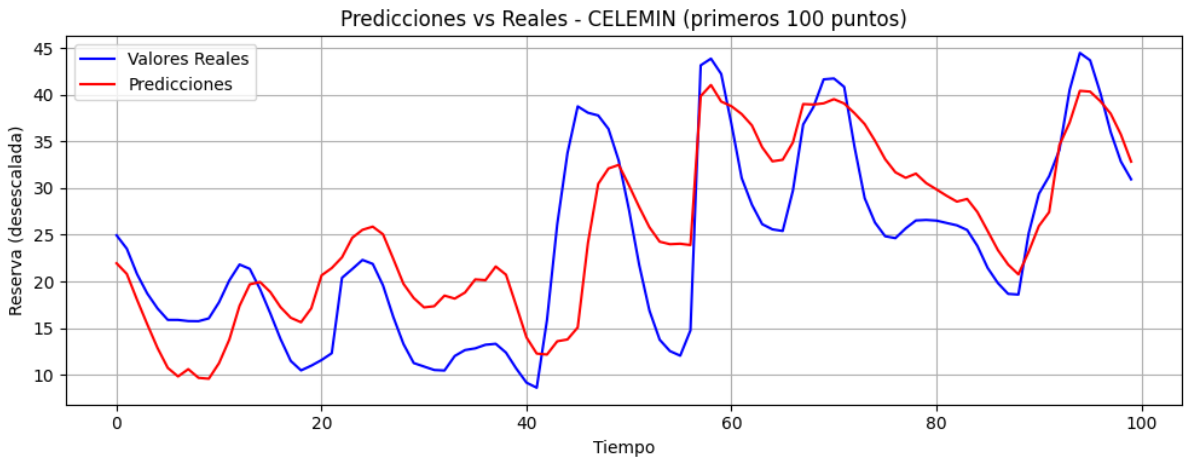


Figura 39: Gráfica embalse CELEMIN E6

En la figura 40 la amplitud y la tendencia están bien representadas aunque hay discrepancias localizadas en picos.

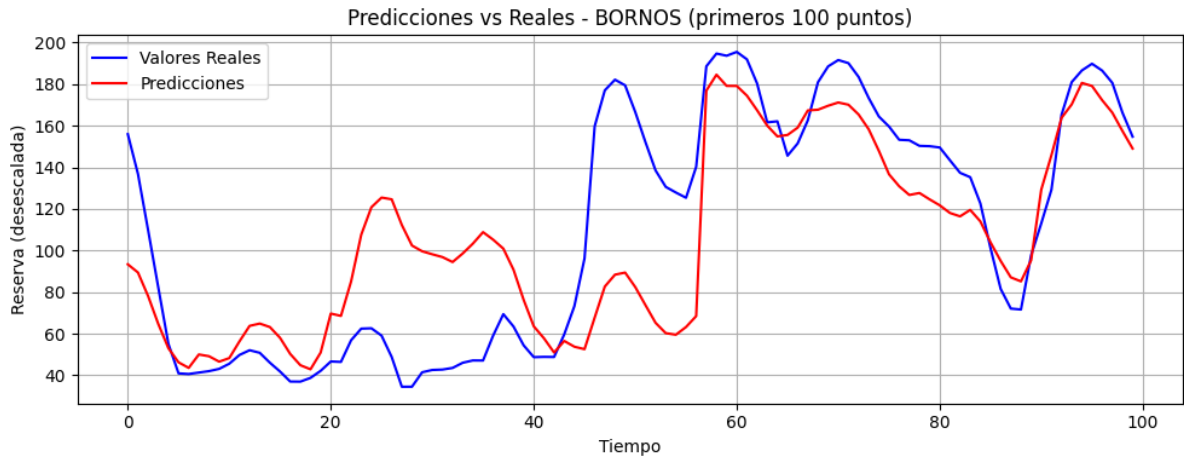


Figura 40: Gráfica embalse BORNOS E6

Y por último, la figura 41 muestra la gráfica de las pérdidas en validación y en entrenamiento. La curvas son consistentes pero con ligera separación. La generalización es adecuada en un conjunto de 50 embalses.

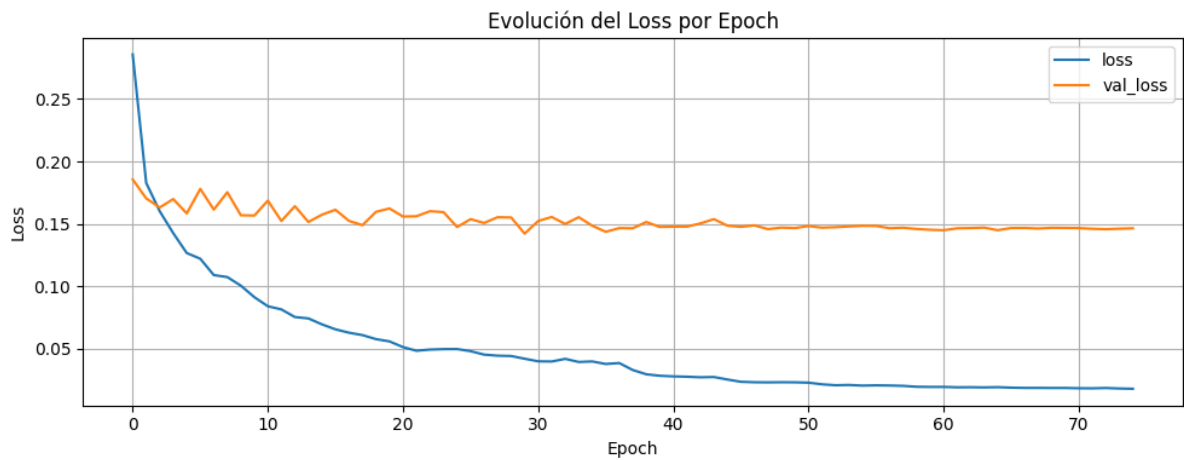


Figura 41: Gráfica de pérdidas E6

La tabla 9 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 41.

Tabla 9: Resultados por embalse E6

Embalse	MAE	RMSE	R^2	MAPE (%)
Charco Redondo	9.03	11.78	0.56	23.88 %
Rúmblar	16.65	19.18	0.63	37.88 %
Celemín	5.26	6.78	0.58	37.59 %
Bornos	19.08	27.74	0.71	27.80 %

E7: 25 años y 73 embalses

- **Ventana temporal:** 48 pasos temporales (equivalente a 48 meses).
- **Unidades GRU:** 128 y 64 en cada capa, respectivamente.
- **Capas densas:** [64, 16, 1] con activaciones *relu*, *relu* y *linear*.
- **Dropout:** 0.1 tras cada capa GRU.
- **Learning rate:** 0.01 (ajustado manualmente tras probar valores menores).
- **Optimizador:** Adam.
- **Criterio de parada:** *EarlyStopping* sobre *val_loss* con *patience=35* y *restore_best_weights=True*.

El resultado de aplicar estas configuraciones se puede observar en la figura 42. Esta figura es la representación gráfica del modelo completo. Muestra un comportamiento lógico a escala regional y sirve de base para escenarios What-If. Aun así, persisten desajustes en episodios extremos.

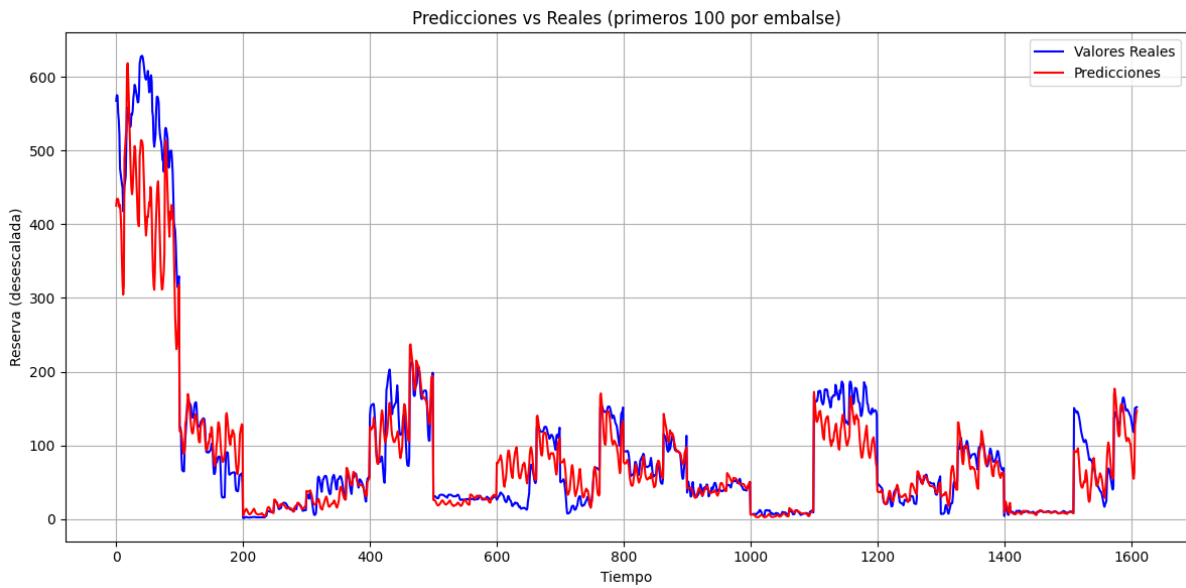


Figura 42: Gráfica modelo completo E7

Las figuras 43, 44, 45 y 46 son las representaciones gráficas por embalse.

En la figura 43 se siguen las tendencias de fondo, pero con errores relevantes en amplitud/temporización de picos.

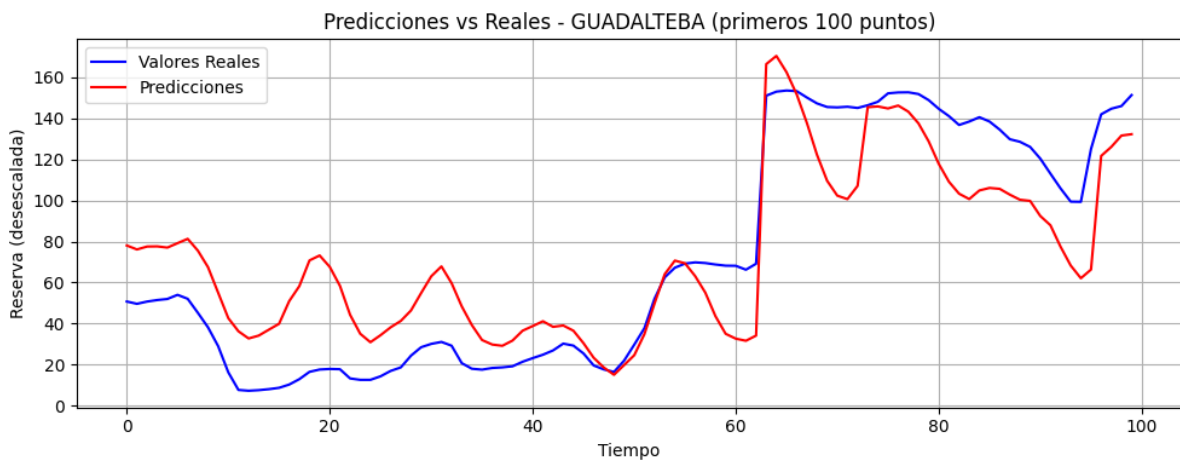


Figura 43: Gráfica embalse GUADALTEBA E7

En la figura 44 la serie predicha refleja la dirección general con poca precisión. Hay que tener precaución al interpretar valores puntuales.

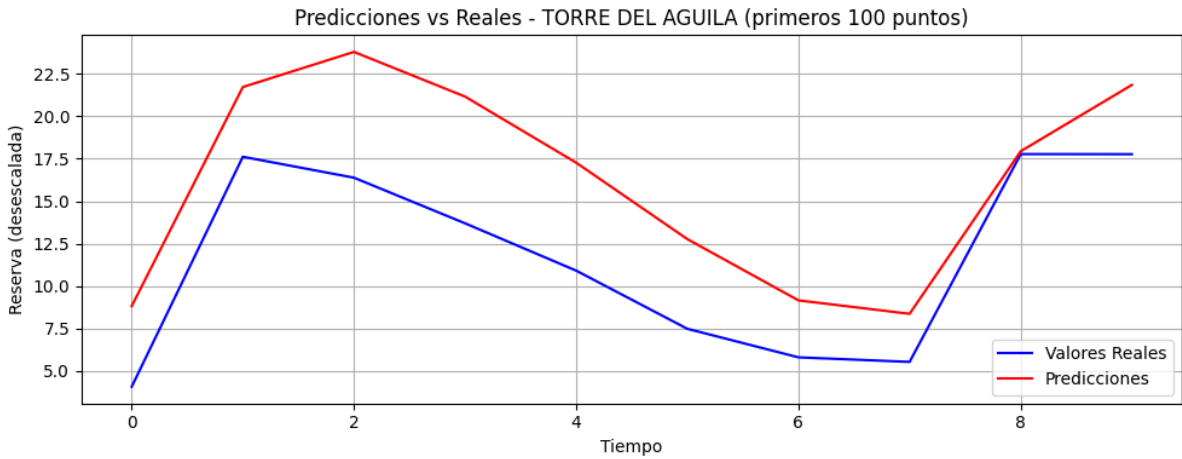


Figura 44: Gráfica embalse TORRE DEL AGUILA E7

En la figura 45 la dinámica estacional se reproduce. Los errores están concentrados en transiciones rápidas.

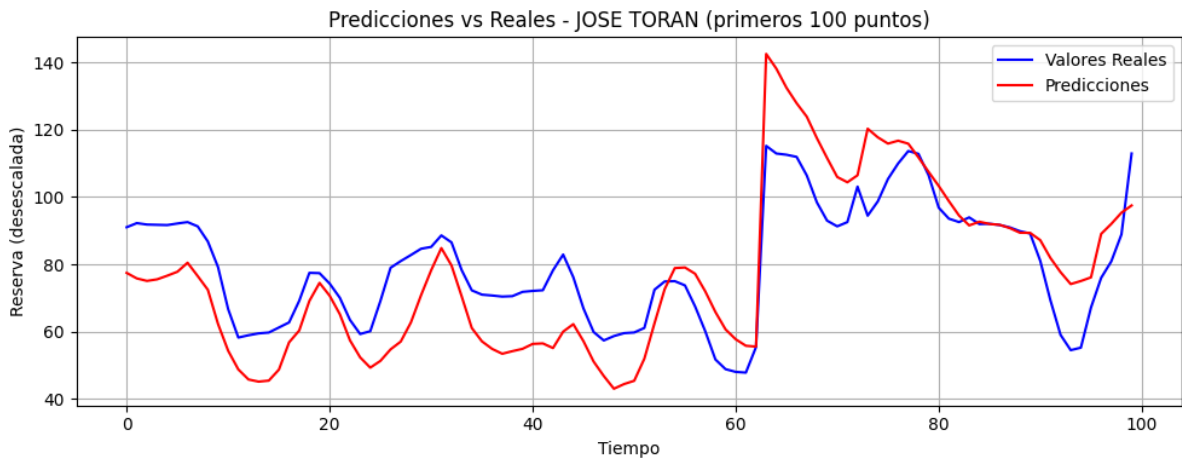


Figura 45: Gráfica embalse JOSE TORAN E7

En la figura 46 la forma es razonable. El porcentaje se distorsiona cuando los valores reales son muy bajos.

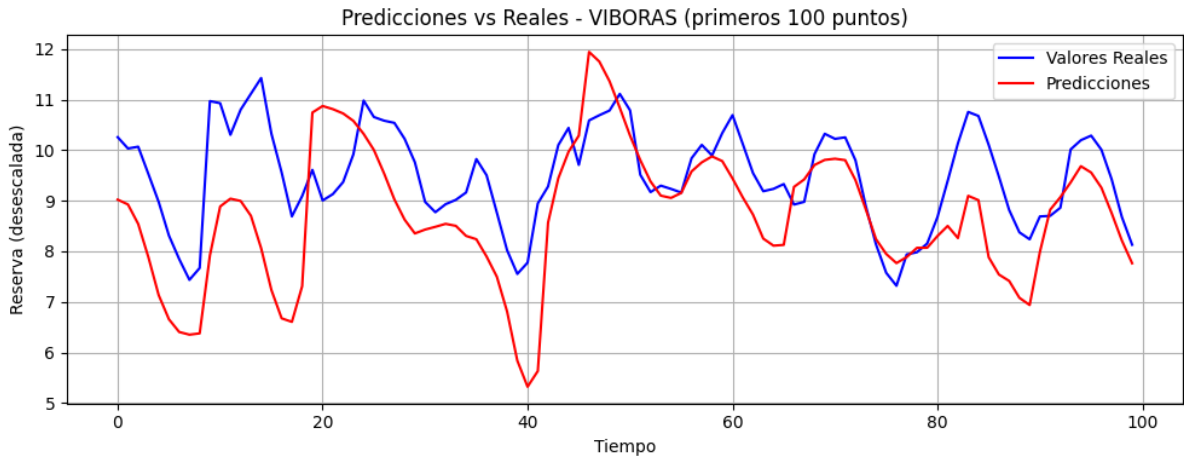


Figura 46: Gráfica embalse VIBORAS E7

Y por último, la figura 47 muestra la gráfica de las pérdidas en validación y en entrenamiento. Ambas descienden moderadamente. La parada temprana estabiliza el modelo. La generalización es algo menor que en conjuntos más pequeños, acorde al aumento de complejidad.

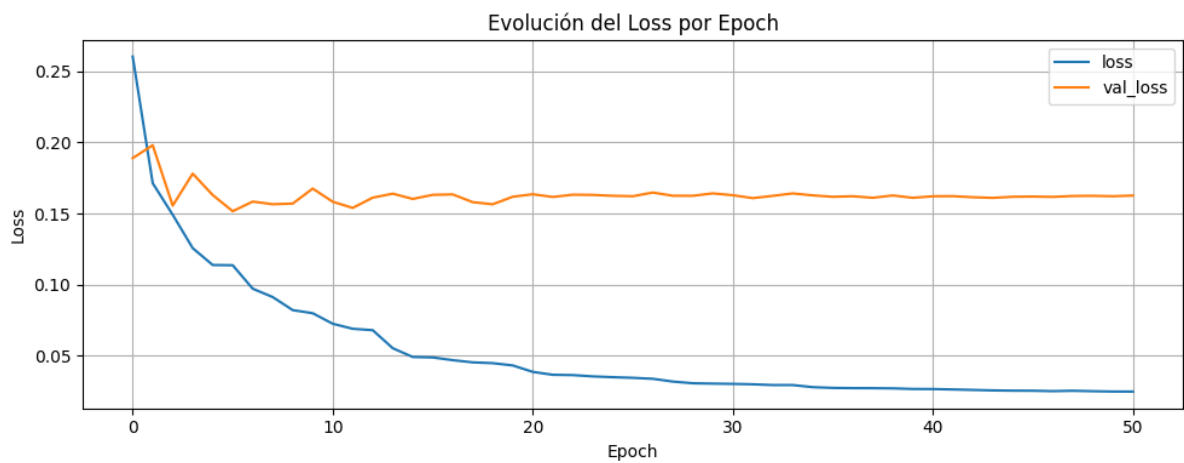


Figura 47: Gráfica de pérdidas E7

La tabla 10 muestra los valores de las pérdidas en el entrenamiento y en la validación. Dicha tabla va acompañada de la figura 47.

Embalse	MAE	RMSE	R^2	MAPE (%)
Guadalteba	32.04	39.20	0.29	56.39 %
Torre del Águila	4.58	5.04	0.10	50.08 %
Jose Torán	15.03	17.85	0.56	34.01 %
Víboras	1.16	1.40	0.29	13.50 %

Tabla 10: Resultados por embalse E7

En la tabla 11 se resumen todos los resultados de los experimentos que han salido bien:

Años	Embalses	MAE	RMSE	R^2	MAPE	Pérdida en entrenamiento	Pérdida en validación
	25 embalses	4.3273	6.4672	0.9931	28.45 %	0.0332682840526104	0.15989470481872559
15 años	50 embalses	23.7428	49.4387	0.8902	25.97 %	0.03250793367624283	0.16716589033603668
	73 embalses	14.6855	26.6914	0.9051	37.52 %	0.036749813705682755	0.17358006536960602
	25 embalses	25.3548	37.2978	0.8130	41.38 %	0.02614147774875164	0.16707968711853027
25 años	50 embalses	24.6041	52.1749	0.8423	30.41 %	0.01795930601656437	0.14646942913532257
	73 embalses	21.8100	37.4800	0.8461	49.21 %	0.024726377800107002	0.16267670691013336

Tabla 11: Resultados de los experimentos por número de años y embalses.

Experimentos fallidos

Todos los experimentos anteriores han sido resultados de muchos intentos fallidos. Las figuras 48, 49, 50, 51, 52, 53, 54 y 55 muestran los casos que salieron mal.

- En la figura 48 se solaparon trozos de series de distintos embalses uno detrás de otro. No hay continuidad temporal real y cada embalse tiene su escala.

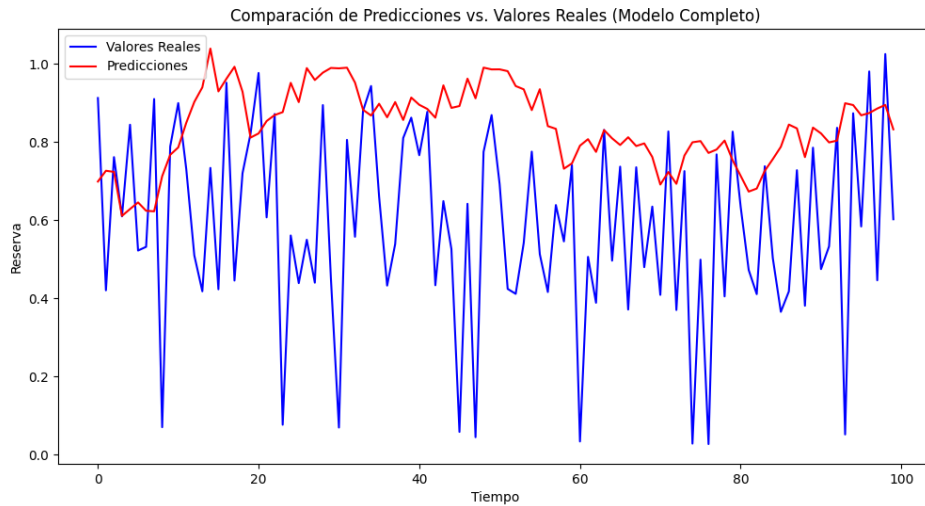


Figura 48: Modelo fallido 1

- En la figura 49 la línea de predicción sale casi plana alrededor de un valor medio. Es un caso claro de subajuste: el modelo no recibió la reserva pasada y sólo vio meteorología agregada, por eso no reproduce saltos ni mesetas.

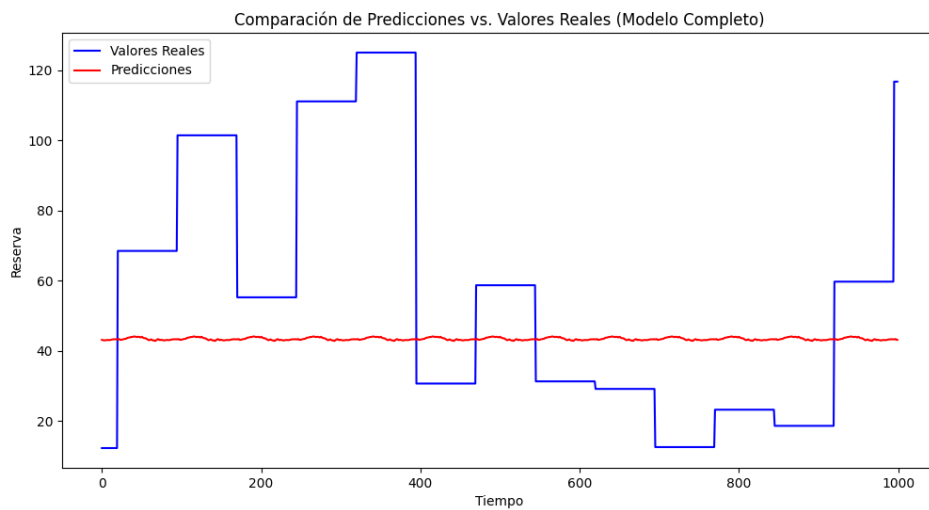


Figura 49: Modelo fallido 2

- En el caso de la figura 50 el problema es similar al anterior: predicción muy suavizada y sin picos. Falta memoria útil (sin la reserva rezagada) y la capacidad/ventana no alcanzan para captar cambios bruscos. Además, mezclar embalses en una única línea empeora la lectura.

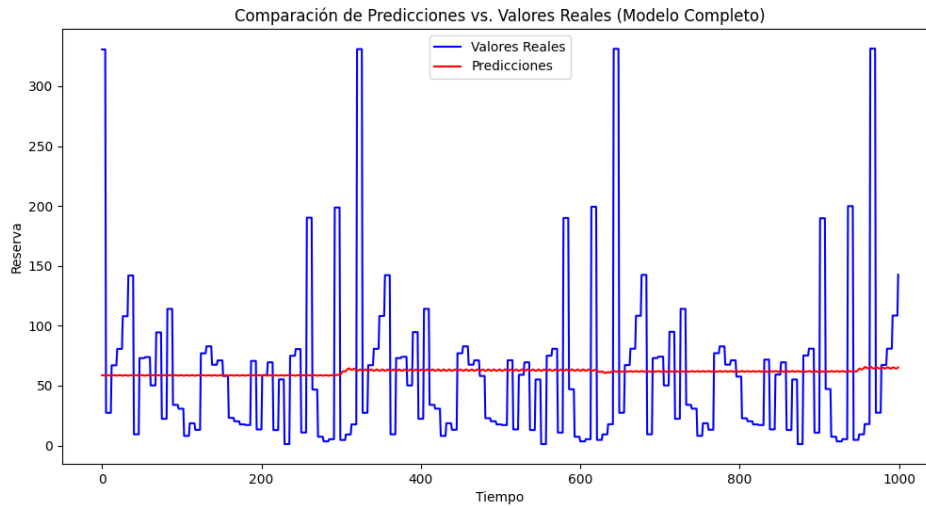


Figura 50: Modelo fallido 3

- En la figura 51 todo está normalizado (0–1) y se han mezclado embalses. La predicción parece “mal” porque sigue una tendencia media normalizada, pero no puede clavar oscilaciones rápidas sin la señal de reserva histórica.

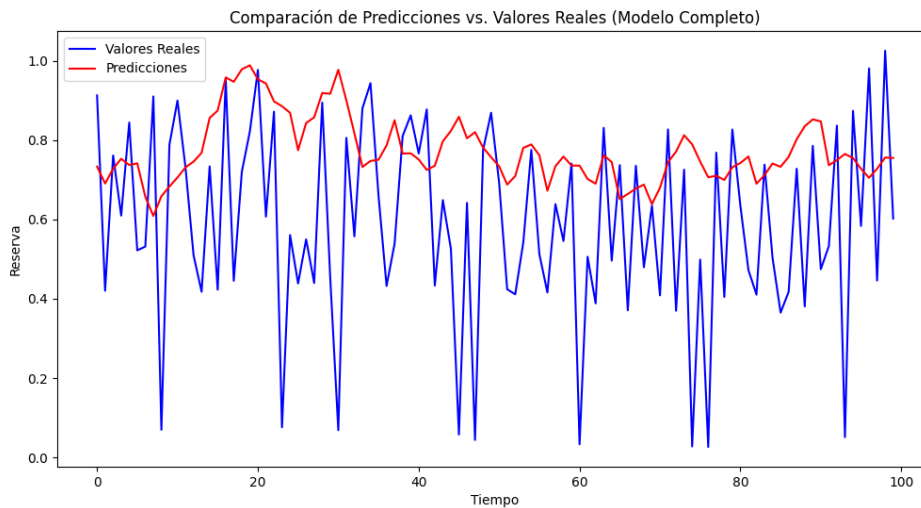


Figura 51: Modelo fallido 4

- En la figura 52 el rojo va casi recto mientras el azul hace escalones. El modelo aprendió el nivel medio por embalse (gracias a la codificación del embalse), pero no las dinámicas dentro del mes por usar agregación mensual y no incluir la reserva pasada.

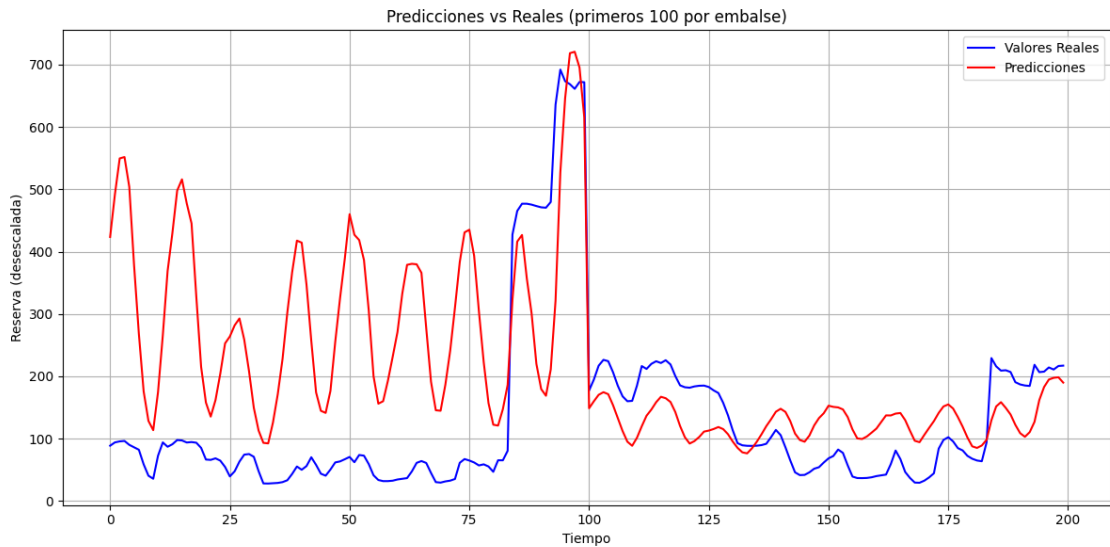


Figura 52: Modelo fallido 5

- En la figura 53 la predicción desescalada muestra amplitud y ondas poco realistas. Se usó el scaler equivocado al desescalar (global o de otro embalse), por eso las unidades y la variabilidad no cuadran con la realidad.

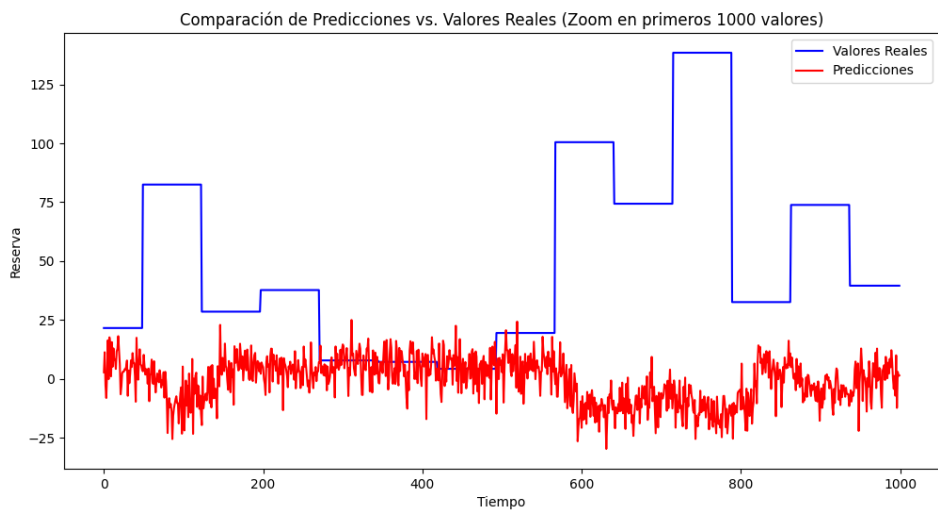


Figura 53: Modelo fallido 6

- En la figura 54 aparecen valores negativos y rangos raros por un fallo en el posprocesado: el orden entre *inverse_transform* y *expm1* no fue el correcto (o faltó uno de ellos). El error es del desescalado, no de la GRU.

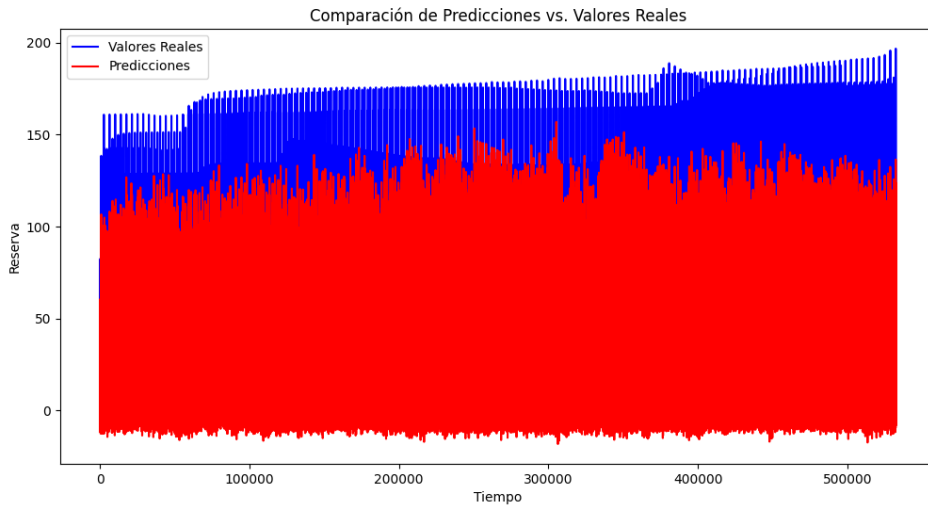


Figura 54: Modelo fallido 7

- En la figura 55 la predicción está alrededor de cero con mucha dispersión. Se desescaló con un scaler global y se concatenaron embalses en el eje tiempo. Como cada embalse tiene su propio rango, las predicciones quedaron aplastadas y desplazadas.

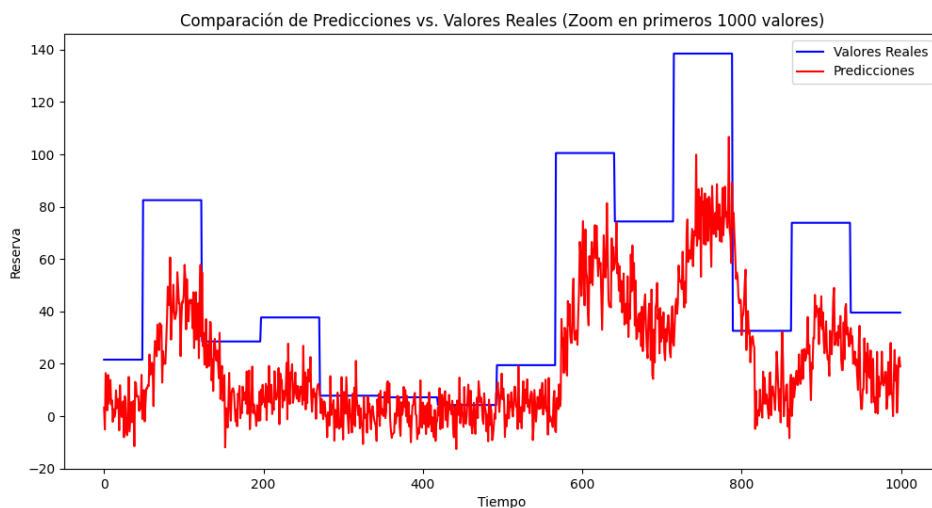


Figura 55: Modelo fallido 8

3.3.2. Resultados de escenarios *What-If*

Escenario 1: Lluvias y humedad más altas

En este escenario incrementamos la lluvia acumulada a seis meses (*precipitacion_sum_6m*) y los niveles de humedad (especialmente *humedad_max*). El modelo reacciona elevando la reserva en los periodos de aportes y manteniendo la tendencia física esperable. En la figura 56 se aprecia que los máximos no aparecen exactamente a la vez que en la serie real, sino con un ligero retraso, coherente con el tiempo que tarda la lluvia en transformarse en almacenamiento.

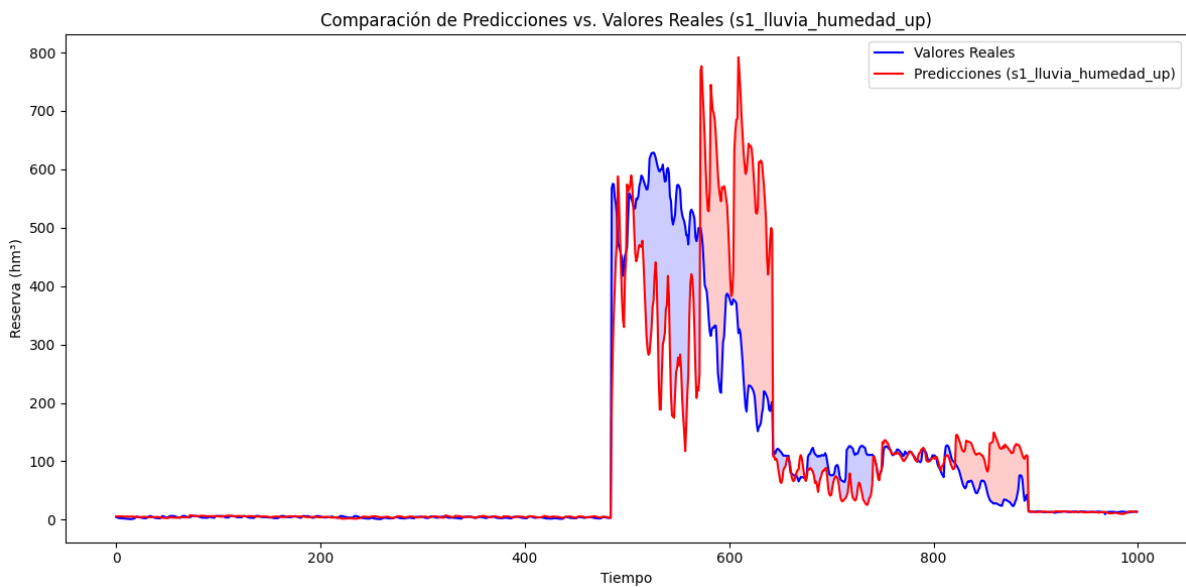


Figura 56: Escenario 1: Gráfica global

Las figuras 57, 58 y 59 son las representaciones gráficas del escenario uno por embalse.

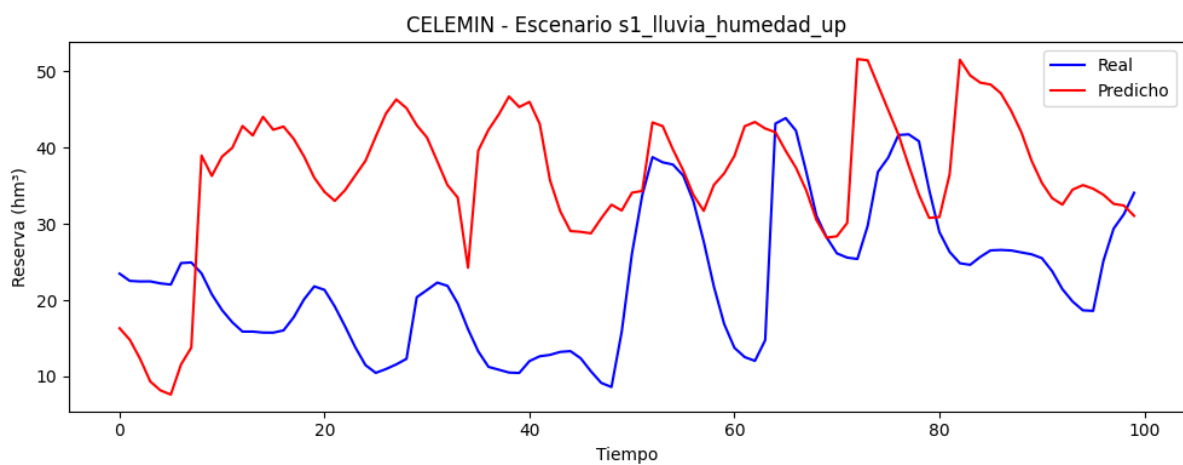


Figura 57: Escenario 1: Gráfica Celemin

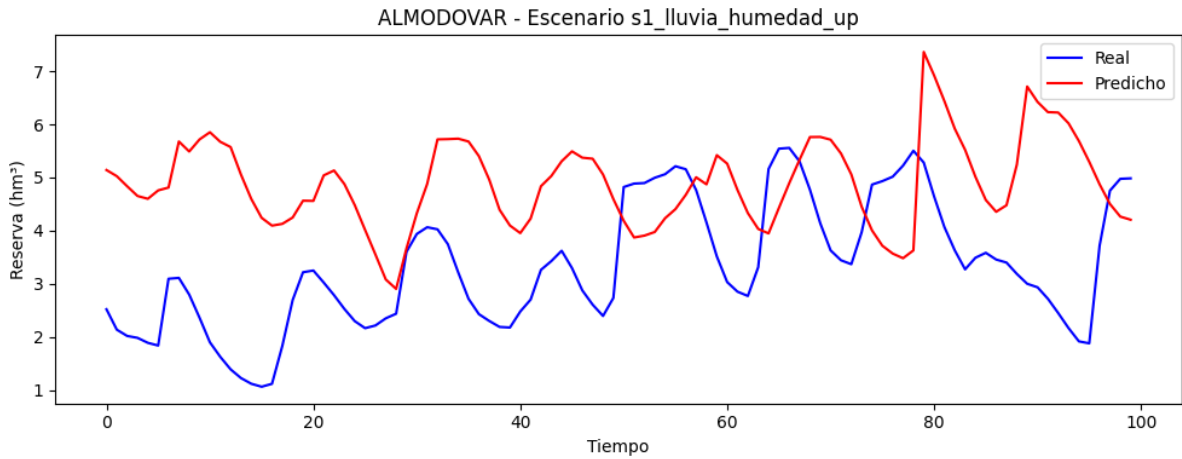


Figura 58: Escenario 1: Gráfica Almodovar

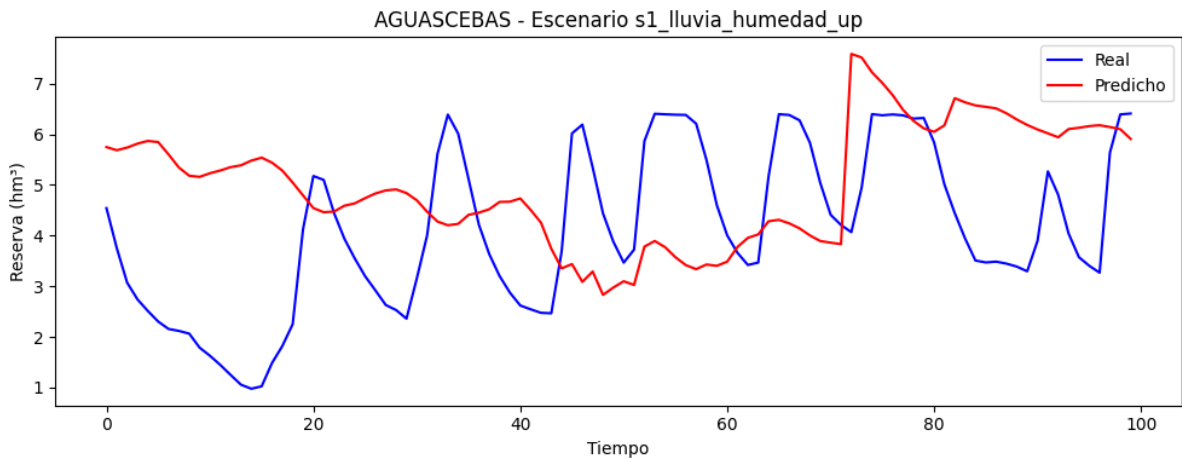


Figura 59: Escenario 1: Gráfica Aguascebas

Escenario 2: Ola de calor y sequía

Aquí reducimos lluvia y humedad y aumentamos temperatura y radiación (más evaporación). El sistema responde con descensos de reserva y recuperaciones más lentas, justo lo que esperaríamos en un episodio cálido y seco. Al combinar varios “factores de pérdida” a la vez, en algunos tramos de la figura 60 el modelo tiende a bajar un poco de más, pero el comportamiento global es coherente: menos entradas y más evaporación implican menos agua almacenada.

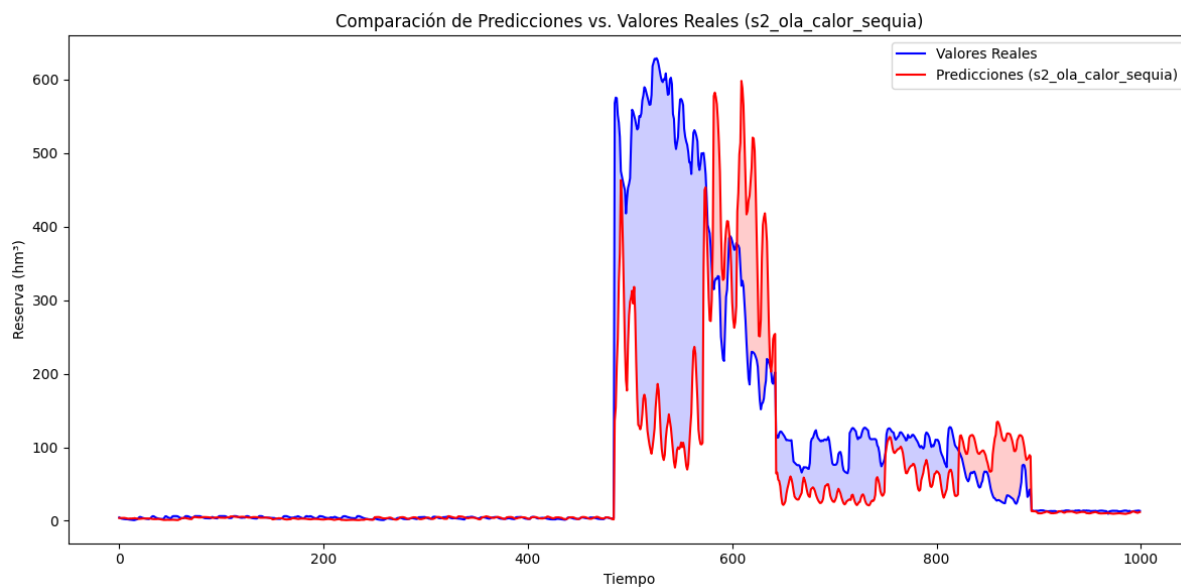


Figura 60: Escenario 2: Gráfica global

Las figuras 61, 62 y 63 son las representaciones gráficas del escenario dos por embalse.

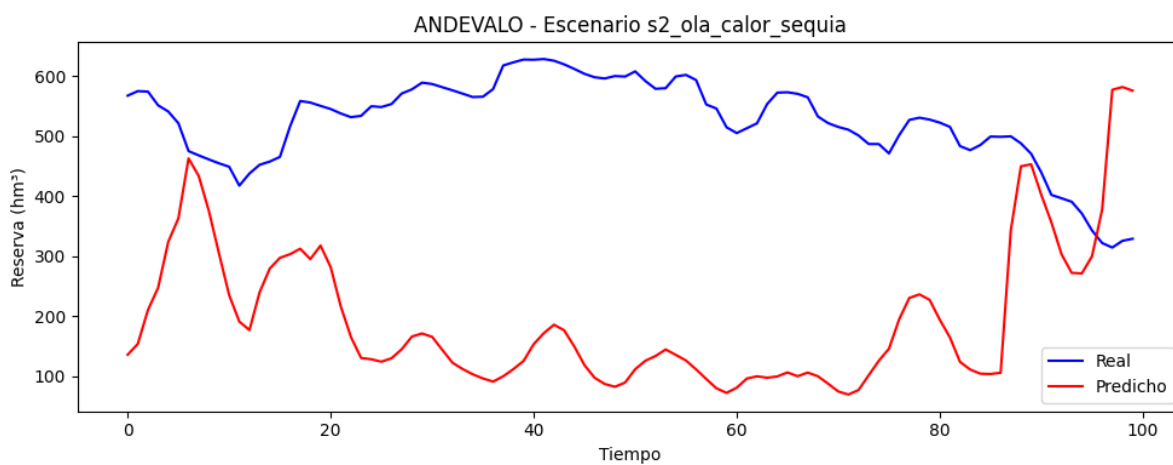


Figura 61: Escenario 2: Gráfica Andevalo

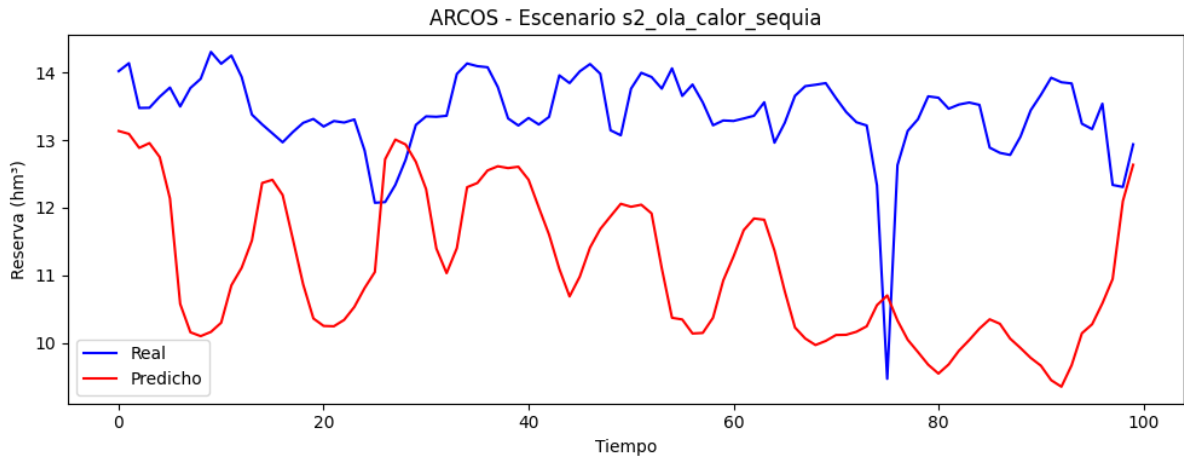


Figura 62: Escenario 2: Gráfica Arcos

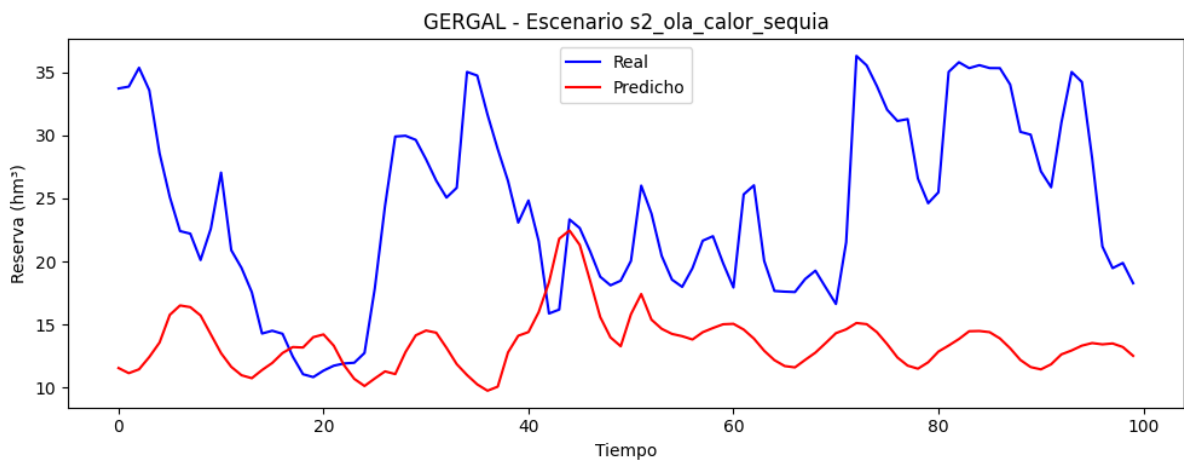


Figura 63: Escenario 2: Gráfica Gergal

Escenario 3: Episodio extremo local en Iznájar

Modificamos de forma localizada la lluvia acumulada en el embalse de Iznájar (multipli-cándola por un factor alto). La reserva simulada aumenta con especial claridad en las ventanas temporales que afectan a ese embalse, sin desbordar al resto, lo que muestra que el modelo es sensible a cambios locales y mantiene la “separación por embalse”. La figura 64 es un buen ejemplo de cómo un incremento sostenido de aportes se traduce en picos más altos y una descarga algo más lenta, dentro de límites realistas.

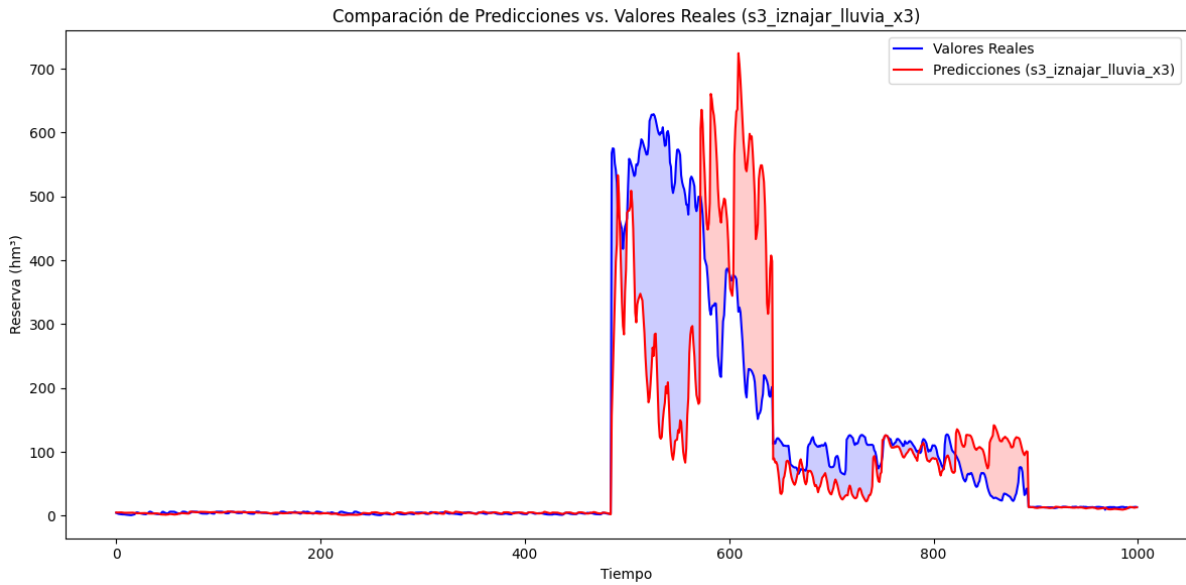


Figura 64: Escenario 3: Gráfica local iznajar

Escenario 4: Sequía prolongada (gran déficit hídrico)

Forzamos un periodo largo con menos lluvia y humedad y mayor evaporación. En la figura 65 la reserva permanece baja durante más tiempo y la recuperación se desplaza hacia más adelante. Es un caso exigente para el modelo, porque históricamente no abundan sequías tan continuadas, pero aun así mantiene la dirección correcta: alargamos la escasez y el sistema lo refleja. Útil como prueba de estrés.

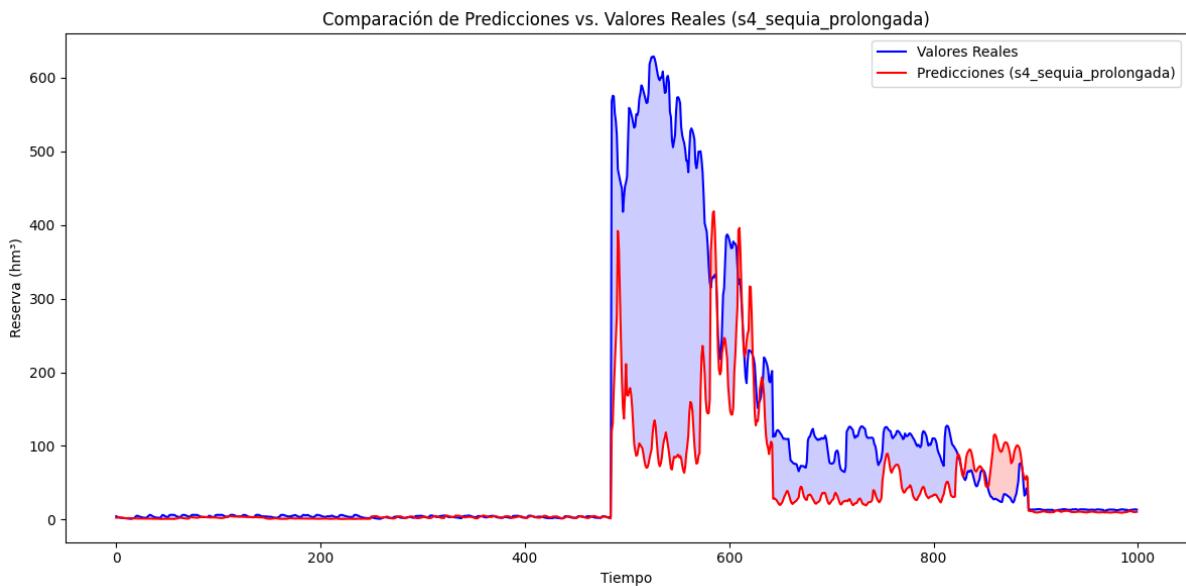


Figura 65: Escenario 4: Gráfica global

Las figuras 66, 67 y 68 son las representaciones gráficas del escenario cuatro por embalse.

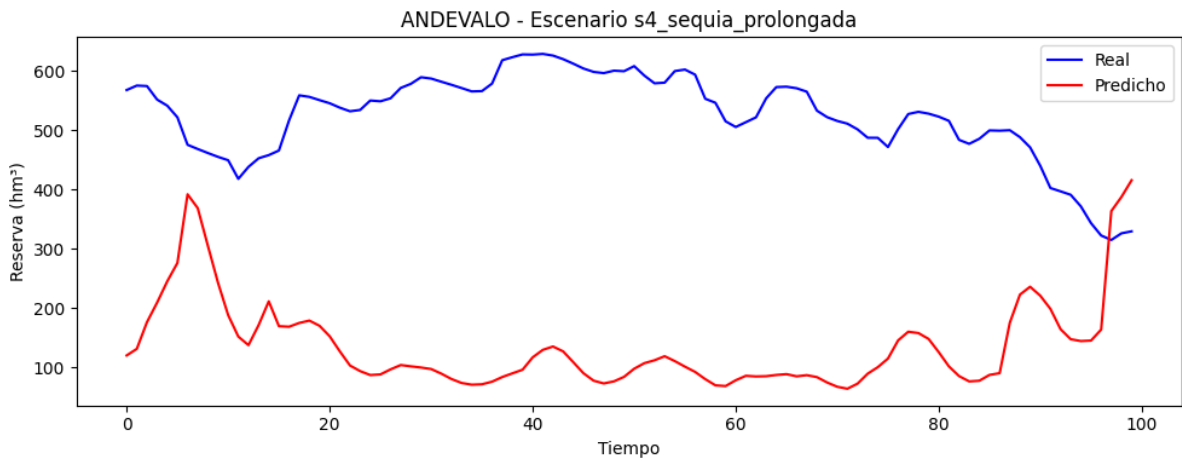


Figura 66: Escenario 4: Gráfica Andevalo

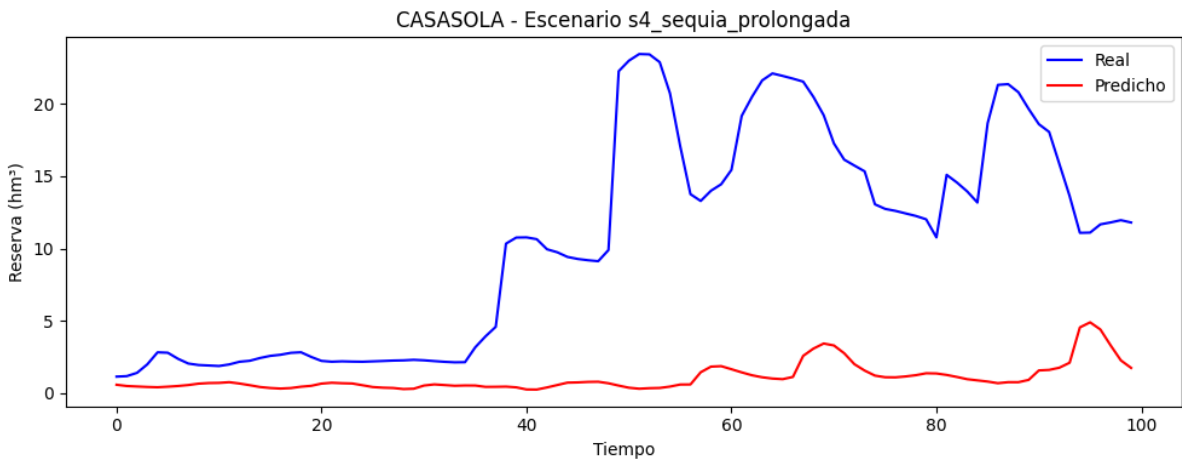


Figura 67: Escenario 4: Gráfica Casasola

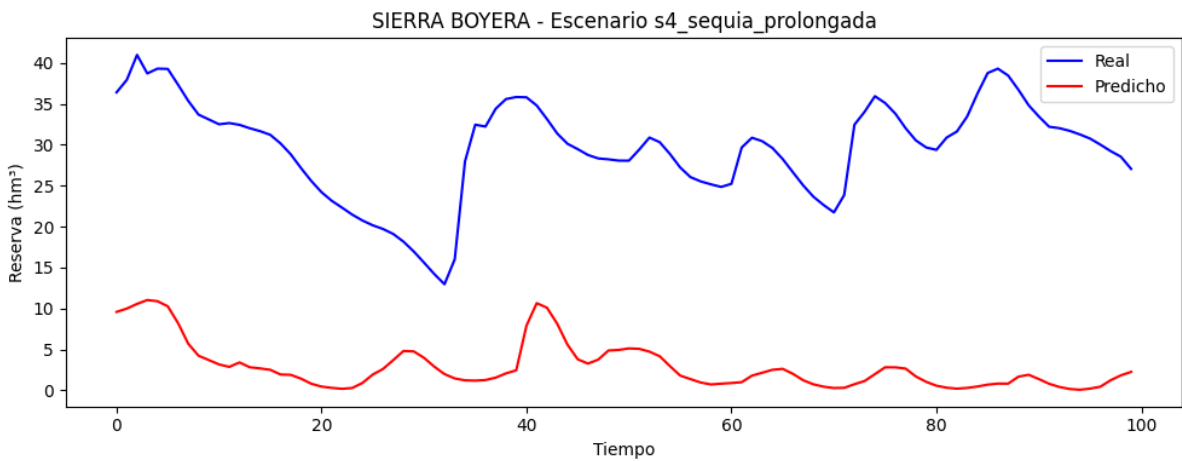


Figura 68: Escenario 4: Gráfica Sierra Boyera

Escenario 5: Enfriamiento húmedo (frentes suaves)

Reducimos temperatura y radiación (menor evaporación) y elevamos la humedad. El resultado son reservas algo más altas y “mesetas” más sostenidas después de los periodos de entrada de agua. En algunos tramos de la figura 69 el modelo tiende a conservar algo más de agua de la que retuvo la serie real, pero esto es consistente con un ambiente más fresco y húmedo que favorece la retención.

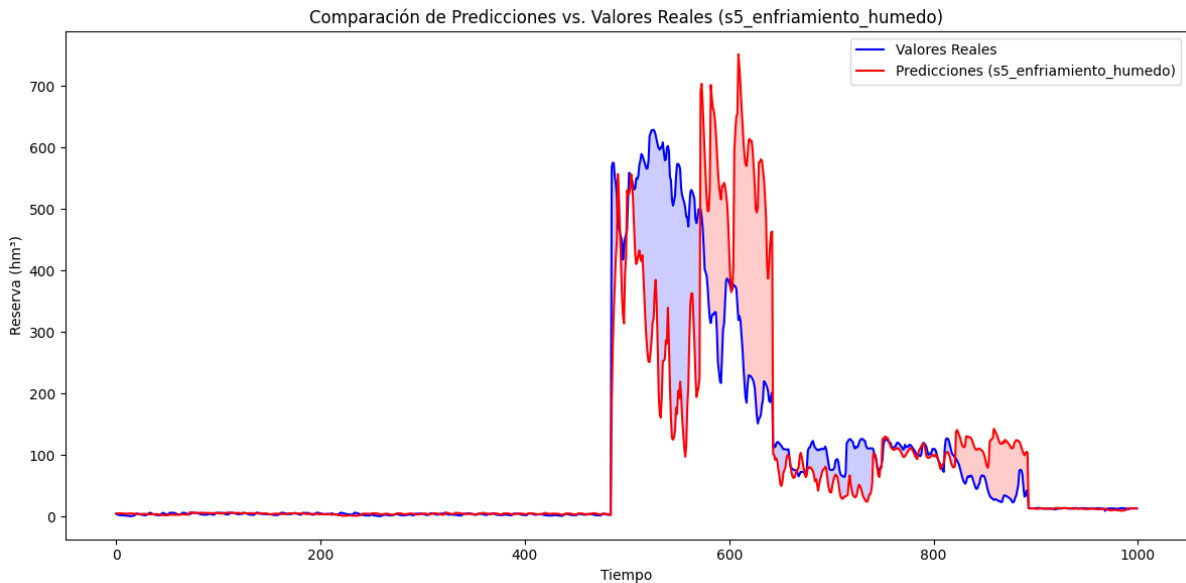


Figura 69: Escenario 5: Gráfica global

Las figuras 70, 71 y 72 son las representaciones gráficas del escenario cinco por embalse.

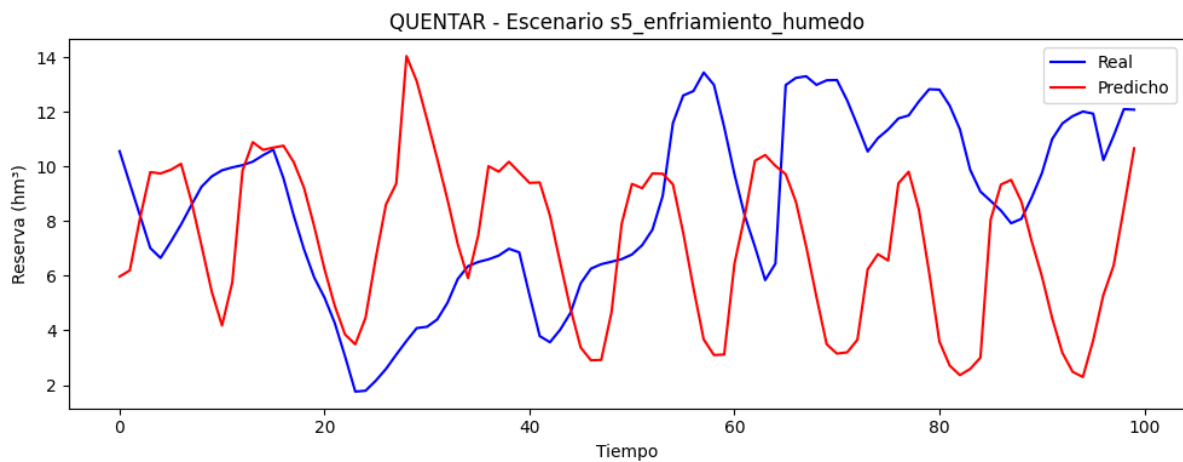


Figura 70: Escenario 5: Gráfica Quéntar

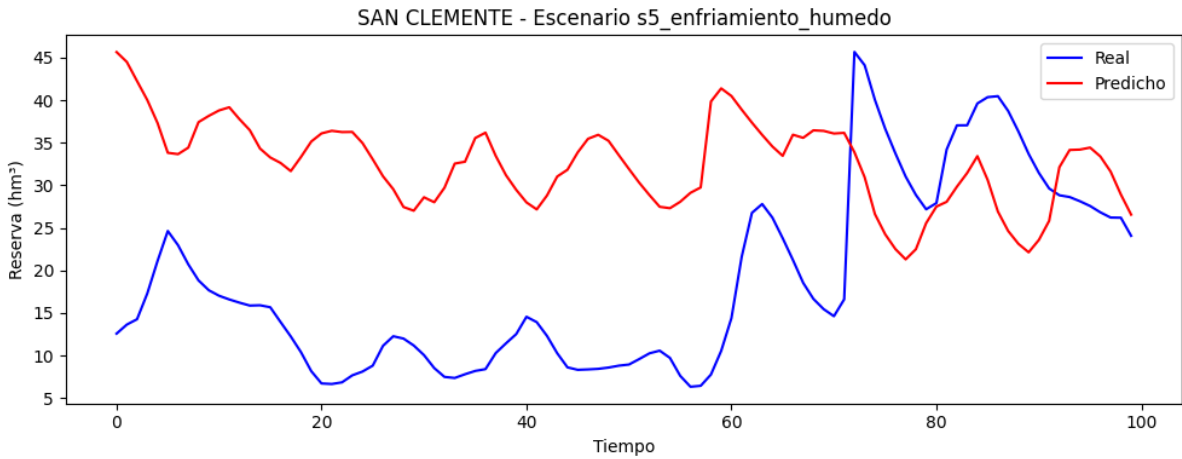


Figura 71: Escenario 5: Gráfica San Clemente

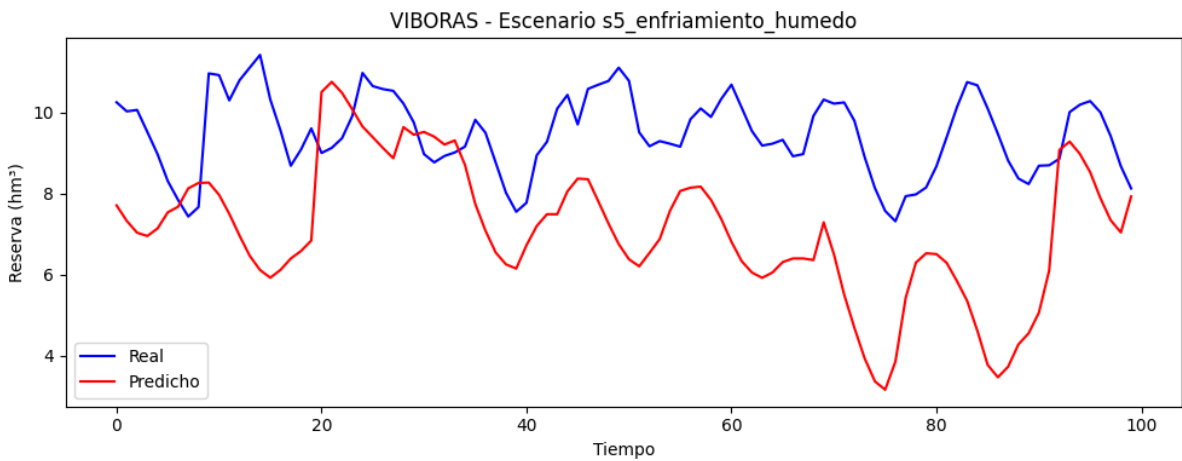


Figura 72: Escenario 5: Gráfica Víboras

Escenario 6: Viento anómalo

Incrementamos la intensidad y el acumulado de viento. De acuerdo con el análisis de correlaciones, el viento apenas explica variación en la reserva a esta escala temporal; el modelo actúa en consecuencia y apenas modifica su salida. En la figura 73 esta estabilidad es positiva: indica que el sistema no “sobrerreacciona” a variables con poco peso hidrológico en el conjunto de datos.

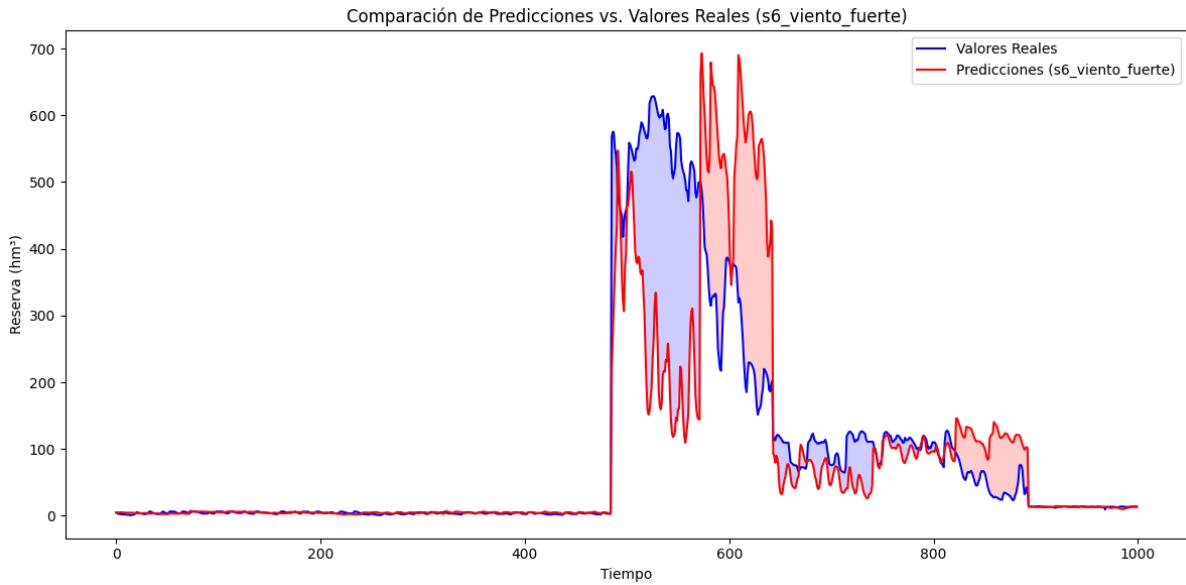


Figura 73: Escenario 6: Gráfica global

Escenario 7: Humedad máxima alta y sostenida

Aumentamos humedad_max, la variable de humedad que mejor se correlaciona con los incrementos de reserva a varias semanas vista. La figura 74 muestra llenados algo más rápidos y transiciones más suaves hacia niveles altos, especialmente en los periodos de recarga. Es un escenario muy plausible porque la humedad alta suele anticipar aportes efectivos a los embalses.

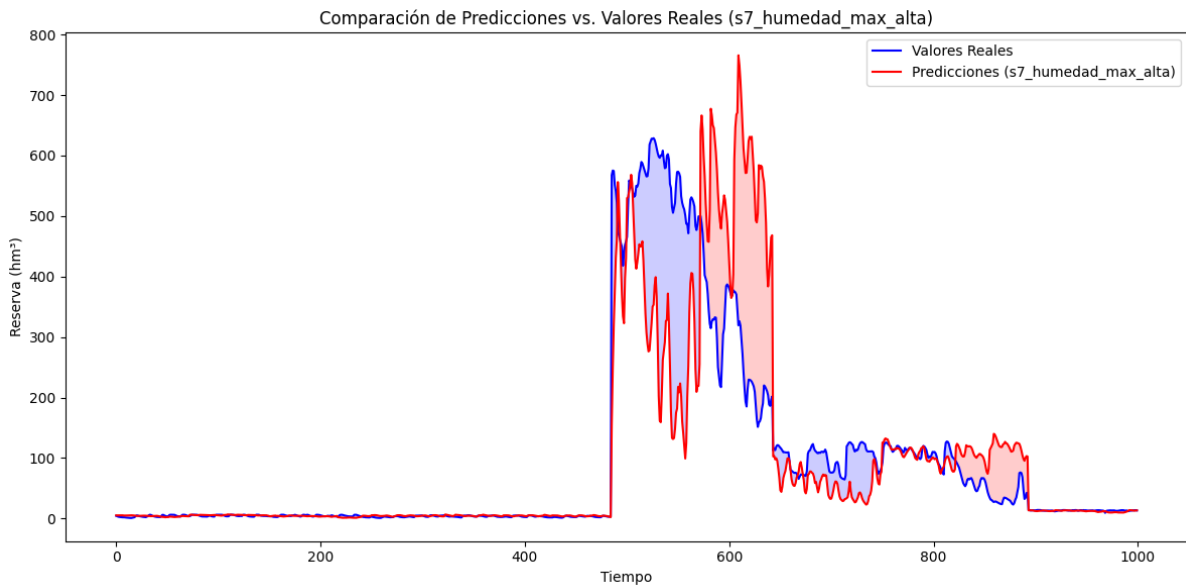


Figura 74: Escenario 7: Gráfica global

Las figuras 75, 76 y 77 son las representaciones gráficas del escenario siete por embalse.

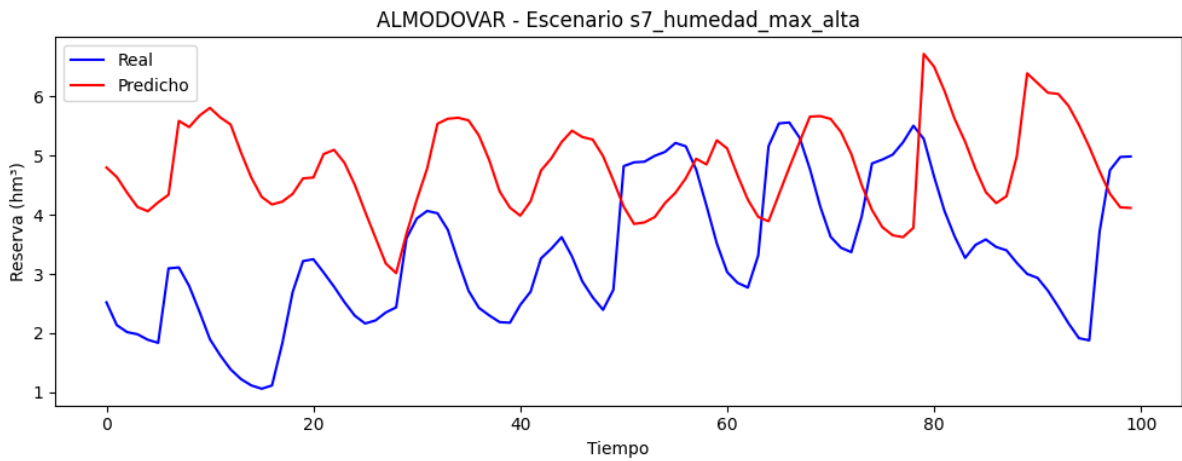


Figura 75: Escenario 7: Gráfica Almodovar

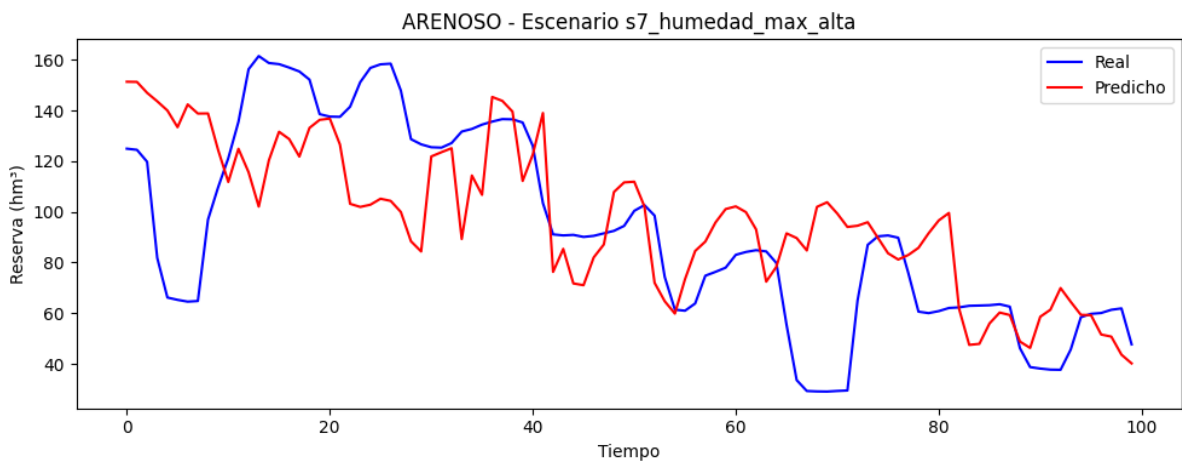


Figura 76: Escenario 7: Gráfica San Arenoso

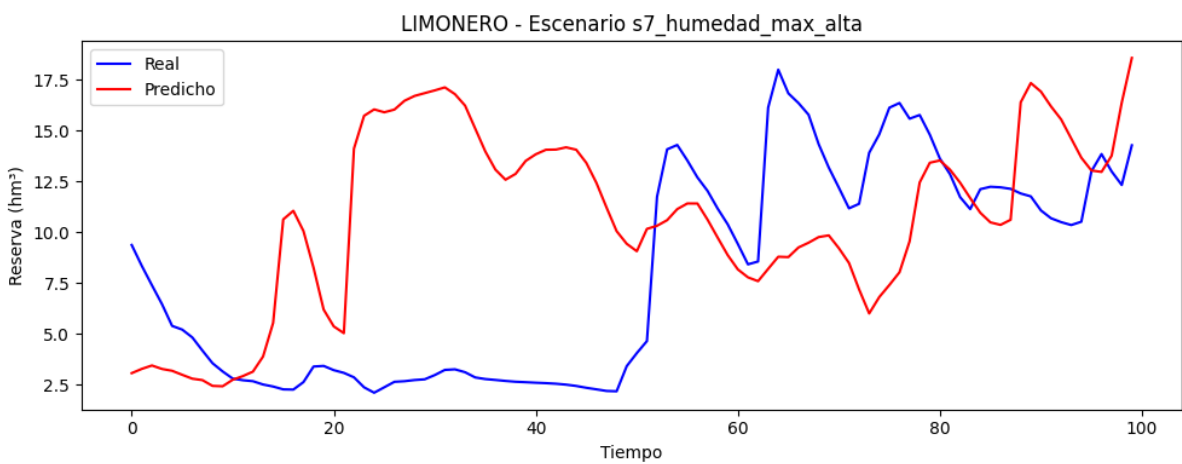


Figura 77: Escenario 7: Gráfica Limonero

Experimentos fallidos

La fase de escenarios *What-If* también tienen varios intentos fallidos.

A continuación se detallan los problemas que han generado dichos resultados:

- **Desescalado incorrecto de la reserva:** En entrenamiento se aplica $y_{\log} = \log(1 + y)$ y luego un `MinMaxScaler` por embalse. En los escenarios no se aplicó la inversa correcta $(\exp(\cdot) - 1)$ y/o se usó un scaler global distinto del del embalse. Resultado: curvas en escala $[0, 1]$ o casi planas frente a hm^3 reales como se observa en la figura 78.

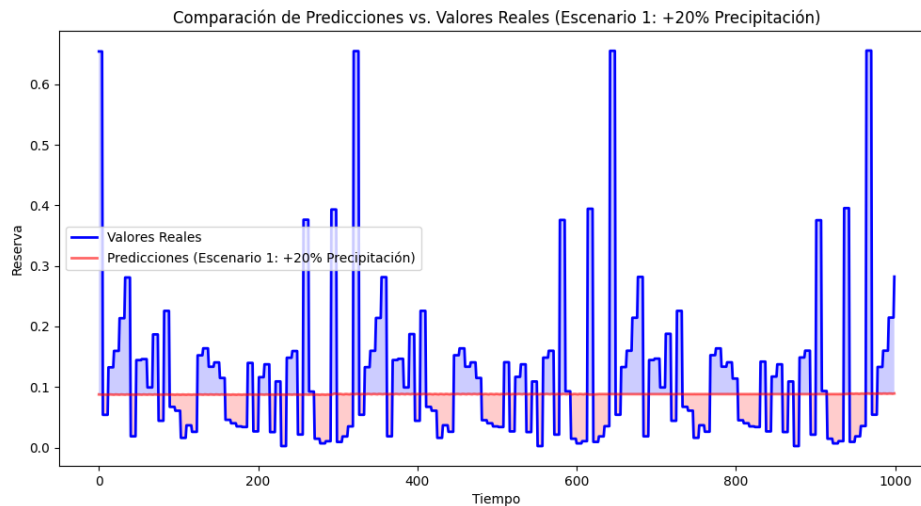


Figura 78: Modelo fallido *What-If* 1

- **Se alteró una variable que el modelo no usa:** Se aplicó el shock a `vel_viento_max` u otras columnas fuera de `variables_finales`; el GRU no las ve, así que la predicción no cambia.

Las figuras 79, 80 y 81 son distintos escenarios modificando las variables de precipitación y temperatura media. Como se puede observar, no importa cuanto modifiquemos las variables ni cuáles porque siempre salía el mismo resultado.

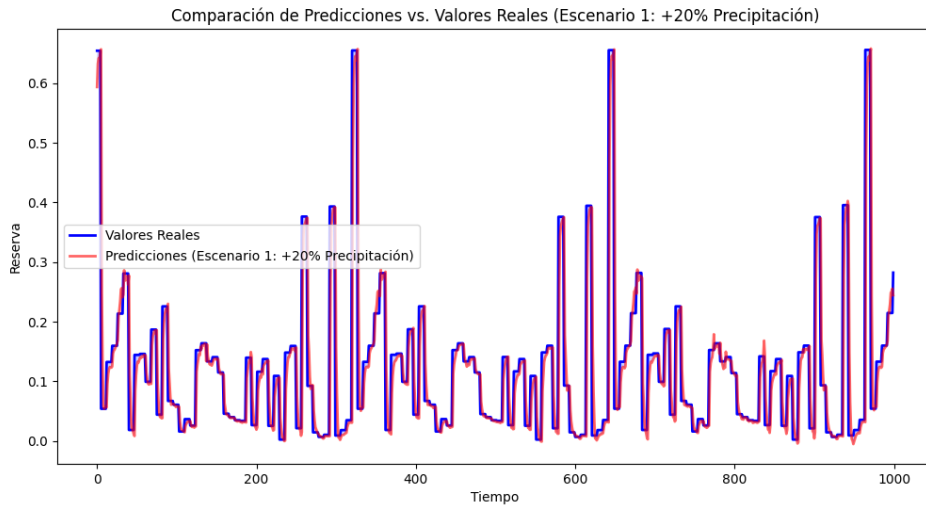


Figura 79: Modelo fallido *What-If 2*

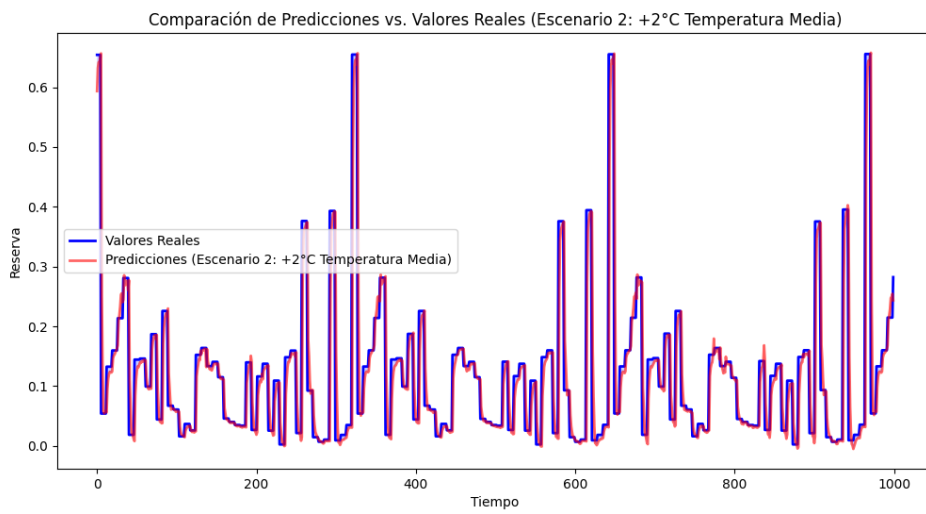


Figura 80: Modelo fallido *What-If 3*

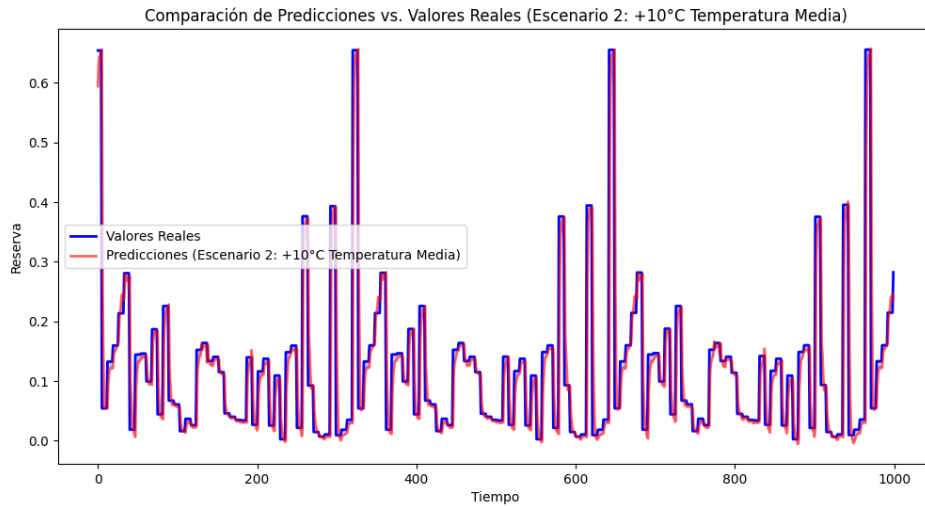


Figura 81: Modelo fallido *What-If* 4

- **No se recomputaron las features derivadas tras el shock.** El modelo consume agregados (`_mean_`, `_sum_`); si se cambia la señal base y no se rehacen esos campos, la entrada al GRU queda igual y el escenario no tiene efecto, como se puede observar en la figura 82.

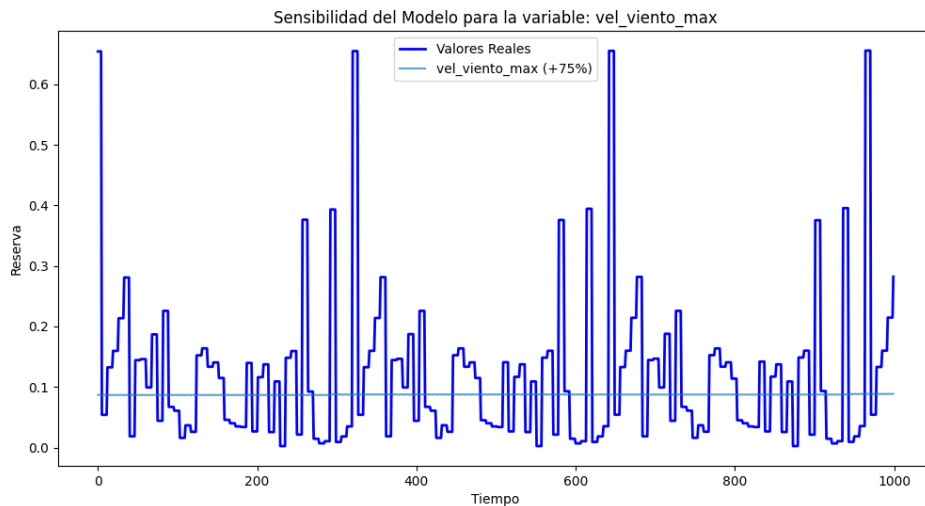


Figura 82: Modelo fallido *What-If* 5

- **Reajuste de scalers/encoders en inferencia.** Se hizo `fit_transform` en lugar de cargar y sólo `transform` (`scaler_entrada.pkl`, `encoder_embalse.pkl`). Cambia la escala y se rompe la coherencia con el entrenamiento, como se puede observar en la figura 83.

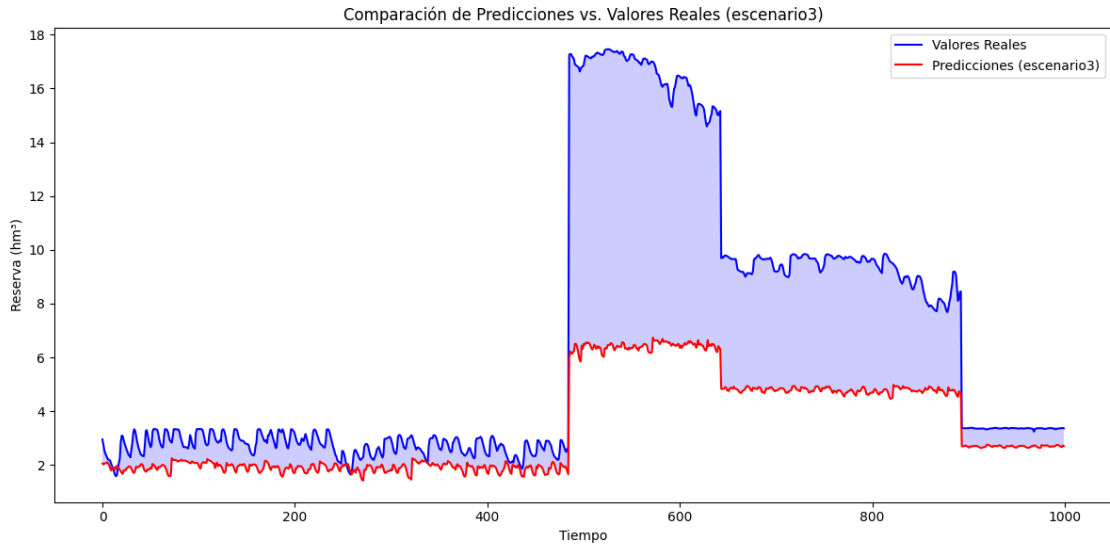


Figura 83: Modelo fallido *What-If 6*

- Mezcla de embalses en la misma gráfica.** Se concatenaron ventanas de embalses diferentes. Cada uno tiene su scaler y dinámica, lo que introduce escalones y artefactos visuales como se puede ver en la figura 84.

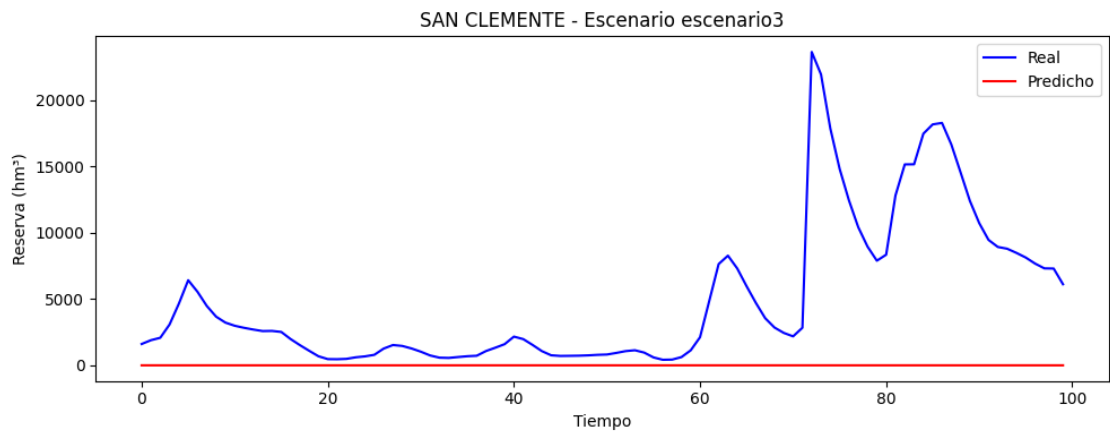


Figura 84: Modelo fallido *What-If 7*

- Inconsistencia en la ventana.** Se prepararon entradas con una ventana (p. ej., 36) distinta a la usada para entrenar (p. ej., 48), desalineando el patrón temporal aprendido, como se puede ver en la figura 85.

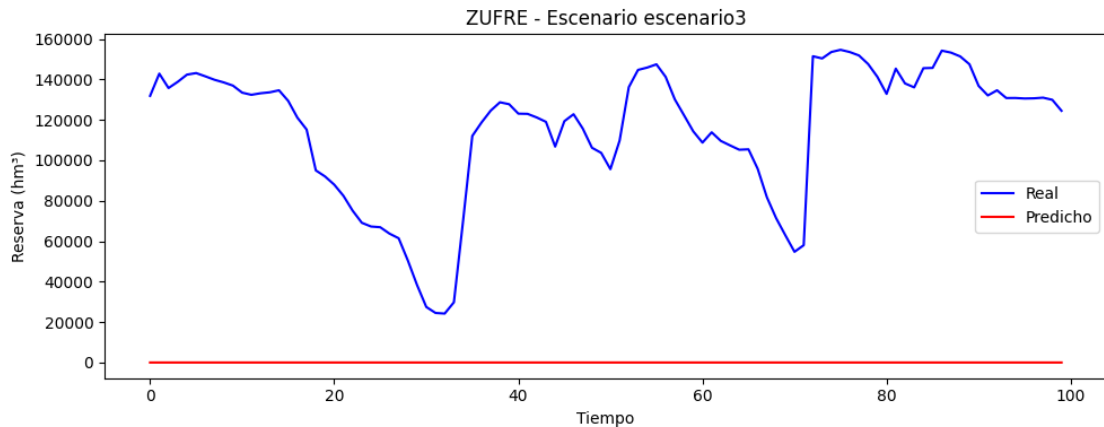


Figura 85: Modelo fallido *What-If* 8

3.4. Discusión general

En este trabajo hemos construido un marco predictivo para la evolución de la reserva en embalses andaluces combinando ingeniería de variables temporales y redes GRU. El plan experimental se dividió en dos etapas: una inicial de base con 5 años y 73 embalses sin selección de variables, y una segunda etapa completa con selección de características (*Feature selection*), 15–25 años de histórico y 25–73 embalses, que culmina en el modelo E7 (25 años, 73 embalses) usado en los escenarios *What-If*. Esta progresión nos permitió pasar de prototipos que evidenciaban subajuste y errores de post-proceso a un modelo coherente y estable para análisis de sensibilidad.

El estudio de correlaciones justificó la necesidad de trabajar con agregados y desfases. Las correlaciones lineales “instantáneas” entre meteorología y reserva fueron muy bajas, lo que confirma que el embalse integra señales meteorológicas con retardo y está mediado por decisiones de explotación. Al analizar correlaciones cruzadas por *lags* semanales, aparecieron patrones más informativos (p. ej., humedad máxima y humedades medias con correlaciones positivas de 12 semanas; temperatura, radiación y evapotranspiración con relaciones negativas y retardadas), aunque siguen siendo moderadas en magnitud. De ahí que las variables finales fueran acumulados/medias móviles (2–6 meses) y la humedad máxima del mes actual, que resultaron entre las más útiles también por importancia en *Random Forest*. Este punto es clave: el fenómeno es débilmente lineal y diferido, por lo que tenía sentido una arquitectura secuencial no lineal (GRU) con ventana de 48 meses.

En la construcción del dataset tomamos decisiones prácticas que mejoraron la estabilidad: agregación mensual (para capturar efectos retardados), escalado por embalse y transformación logarítmica de la reserva para estabilizar varianza; además, codificamos el embalse (*one-hot*) para que la red aprendiera diferencias estructurales entre cuencas. La tabla de hiperparámetros consolidada (ventana, dropout, unidades GRU y learning rate con Adam) refleja las configuraciones que ofrecieron el mejor equilibrio entre capacidad y generalización. Los primeros intentos que salieron “planos” o con amplitudes absurdas nos enseñaron el coste de romper la coherencia de escalados (usar scalars globales, invertir mal \log_{1p}/\exp_{m1}) o de ocultar al modelo su propia memoria de reserva, errores que se documentan en la sección de “experimentos fallidos”. Tras corregirlos, las curvas predichas empezaron a reproducir tendencias, mesetas y picos con mayor fidelidad en la mayoría de embalses.

Los resultados de entrenamiento muestran un comportamiento razonable a escala regional y heterogéneo por embalse. Es lo normal porque cada sistema tiene capacidades, reglas de explotación y demandas distintas. Aun así, el modelo capta bien los ciclos de llenado-vaciado y responde a cambios de régimen; allí donde la R^2 es más pequeña, suele coincidir con embalses fuertemente regulados o con series más ruidosas/escasas. También observamos que métricas como el MAPE se distorsionan cuando la reserva real es muy baja (divisiones por valores próximos a cero), de modo que la comparación visual y MAE/RMSE son más informativas.

Con este modelo, los escenarios *What-If* se comportaron de forma físicamente aceptable: más lluvia/humedad elevó reservas y extendió mesetas; olas de calor y sequía intensificaron descensos y retardaron recuperaciones; choques locales (Iznájar $\times 3$ precipitación) afectaron selectivamente al embalse objetivo sin “contagiar” al resto; y variaciones en viento (variable débil en el análisis) apenas alteraron la salida, señal de que el sistema no sobre-reacciona a predictores de poco peso. Esta consistencia entre señales de entrada y respuesta del modelo respalda su uso como herramienta de apoyo a la decisión para explorar sensibilidades y priorizar políticas (p. ej., impacto de episodios húmedos o de periodos secos prolongados).

Ahora bien, los límites también quedan claros. Primero, las correlaciones bajas nos recuerdan que la reserva no depende sólo del clima: desembalses programados, demandas agrícolas/urbanas, trasvases, restricciones ambientales y pérdidas no medidas (filtraciones a acuíferos, evaporación real) introducen variabilidad que no está en el dataset y que un modelo puramente meteorológico nunca podrá explicar del todo. Segundo, hay desajustes espaciales (meteorolo-

gía provincial vs. cuenca específica) y de escala temporal (agregación mensual) que suavizan señales relevantes (picos de avenida, maniobras operativas intrames). Tercero, faltan predictores hidrológicos básicos (área de cuenca, aportes medidos, balance de entrada-salidas, curvas cota-volumen) que tradujeran lluvia-escurrimiento y reglas de operación, y por tanto reduzcan la carga que hoy asume la red. Todo esto explica por qué, a pesar de la buena coherencia de los escenarios, los resultados podrían ser mejores si se incorporan estas fuentes de información.

En conjunto, el proyecto demuestra que:

- Con ingeniería temporal adecuada y una GRU bien regularizada se puede capturar una parte sustancial de la dinámica de reservas a gran escala.
- Los escenarios *What-If* generados son consistentes con la hidrología básica y útiles para explorar impactos relativos.
- La precisión absoluta está limitada por variables no observadas y por la granularidad espacial-temporal disponible.

4

Aplicación

4.1. Ingeniería del software

4.1.1. Introducción a la plataforma web

Como fase final del proyecto se ha desarrollado una página web con el fin de proporcionar acceso centralizado a datos históricos y recientes sobre los embalses y la meteorología de Andalucía. La plataforma ofrece tanto datos en formato de tablas con filtros, como gráficas interactivas, ayudando al usuario a ver la evolución de los datos a través del tiempo.

La solución cuenta con un *frontend* y un *backend* que conecta con la base de datos de series temporales InfluxDB. Dicha base de datos cuenta con un total de 25 años de datos.

En el desarrollo de la plataforma se ha usado JavaScript tanto en el *frontend* como en el *backend*, junto con herramientas dedicadas a la gestión de datos temporales y el despliegue de servicios.

El objetivo principal es el de proporcionar una herramienta fácil de usar, moderna y útil para:

- Estudiantes e investigadores.
- Técnicos en gestión de embalses y gestión hidráulica.
- Ciudadanos con intereses por la evolución de los embalses y la meteorología en Andalucía.

En las siguientes secciones se detallará más en profundidad los aspectos técnicos de dicha plataforma web.

4.1.2. Estructura del proyecto

La Figura 86 resume el flujo completo: a la izquierda, la ingesta y el procesamiento de datos (ETL y *web scraping* con Selenium) que normalizan y cargan la información en InfluxDB; en el centro, los modelos predictivos (Python, scikit-learn y TensorFlow/GRU) donde se entrena y se guarda el modelo; y, a la derecha, la capa de visualización web en React, desplegada con Docker. Esta figura sirve como guía para lo que se explica a continuación en esta sección y en las siguientes.

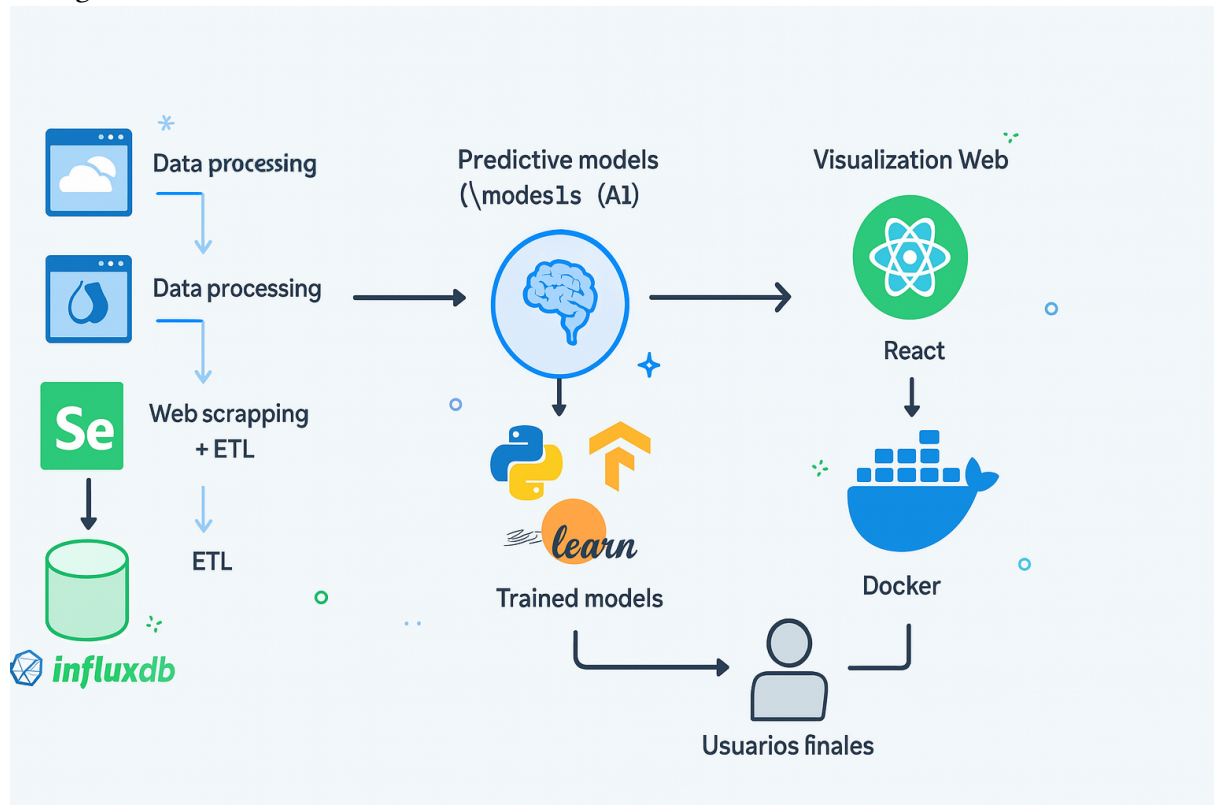


Figura 86: Arquitectura general del proyecto

Frontend

El frontend de una página web es la parte con la que los usuarios interactúan.

En este caso, se ha usado principalmente **React**. Esta biblioteca es la que nos ha permitido crear una interfaz dinámica e interactiva donde los usuarios pueden acceder fácilmente a tablas y gráficas. Por una parte, las tablas cuentan con filtros de provincias, fecha única, rango de fechas y estaciones. Mientras que las gráficas cuentan con filtros de selección múltiples y menús desplegables de forma que el usuario pueda comparar en la misma gráfica la evolución

de los embalses y la meteorología.

En cuanto a la estructura del proyecto, el frontend usa React, lo que quiere decir que sigue una lógica basada en componentes. Estos componentes permiten separar la lógica de datos de la presentación.

Backend

El backend es la parte que se encarga de llevar la lógica del servidor, gestionar las peticiones e interactuar con la base de datos.

En cuanto al *backend*, se ha construido una API REST usando Node.js y Express. Esta API es la que ha permitido gestionar peticiones y aplicar filtros actuando como intermediario entre la base de datos y la interfaz de usuario.

Para gestionar las peticiones, tanto generales como específicas, se han creado varios *endpoints*, como por ejemplo, la obtención de datos para las gráficas o las consultas filtradas por estaciones, provincias, etc.

Otro aspecto importante es la conexión entre la base de datos y el backend. Dicha conexión se ha hecho a través de una librería oficial de InfluxDB para Node.js mediante consultas escritas en lenguaje Flux.

En cuanto a la estructura del proyecto, el backend tiene un único archivo (*server.js*) donde se definen los *endpoints* ya mencionados. Este archivo es el servidor y usa Express para conectarse con InfluxDB. Algunas de sus funciones son: consultar datos de los embalses y la meteorología, filtrar por medidas, provincias, estaciones y fechas, proporcionar los datos en forma de tablas y gráficas y gestionar errores. Para levantar el servicio de backend el proyecto cuenta con un archivo (*docker-compose.yml*) que permite ejecutarse en un contenedor Docker.

Por último, para añadir seguridad al proyecto, se ha creado un archivo *.env* que guarda los datos de configuración, como el token de InfluxDB y la URL. Este archivo se lee en tiempo de ejecución de forma que no se exponen las credenciales en el código fuente .

4.1.3. Despliegue de servicios

Inicialmente, la base de datos se ejecutaba manualmente en local pero para evitar errores de configuración y portabilidad y separar los servicios se optó finalmente por usar Docker.

El archivo docker-compose.yml mencionado anteriormente es el que se encarga de iniciar todos los servicios. Se han creado contenedores para el backend (node), la base de datos InfluxDB y para el frontend.

4.1.4. Funcionamiento de la aplicación web

Las funcionalidades principales de la web son la visualización de gráficas y la visualización de tablas. Ambas están implementadas para que de manera intuitiva el usuario pueda interactuar con ellas.

Una funcionalidad secundaria es la visualización del mapa de Andalucía con la localización de todos los embalses y estaciones meteorológicas.

Con respecto a las tablas, se distinguen dos independientes:

- **Tabla de embalses:** Esta tabla cuenta con filtros por embalses y selección de fechas, tanto fechas únicas como rango de fechas.
- **Tabla de datos meteorológicos:** En el caso de esta tabla, cuenta con un filtro más. Filtros de selección de fechas, únicas y rango de fechas, filtros de provincias (Málaga, Sevilla, Cádiz, Huelva, Córdoba, Jaén y Almería) y filtros por estaciones meteorológicas.

Para obtener estos datos, al pulsar el botón de "Buscar", se lanza una petición al backend, el backend le envía la consulta a InfluxDB y este nos devuelve los datos solicitados según los filtros. Esto sigue una lógica sin endpoints específicos permitiendo interacciones rápidas y basadas en los datos únicamente.

Por otro lado, también se distinguen dos gráficas independientes, la gráfica de la evolución de los embalses y la gráfica de la evolución de la meteorología.

- **Gráfica de embalses:** Esta gráfica cuenta, como en el caso de las tablas con filtros. Filtros de selección de fechas (fecha de inicio y fecha de fin), filtros en forma de checkboxes para seleccionar uno o más embalses y filtros de selección de medidas. En el caso del último filtro, los embalses cuentan con dos medidas, la reserva en hectómetros cúbicos y el porcentaje de reserva.

Ambas gráficas siguen la lógica de los endpoints para obtener los datos de InfluxDB. En este caso, el endpoint es /api/embalses-grafica.

Una vez que nos devuelven los datos, la gráfica nos muestra una línea distinta por cada embalse, en el caso de que seleccionemos más de uno. De esta forma, el usuario puede comparar los datos de distintos embalses en el mismo espacio de tiempo.

- **Gráfica de meteorología:** En el caso de esta gráfica, cuenta con un filtro más. Filtros de selección de fechas (fecha de inicio y fecha de fin), filtro de provincias (Málaga, Sevilla, Cádiz, Huelva, Córdoba, Jaen y Almería), filtro de medidas (precipitación, temperatura máxima, mínima y media, evapotranspiración, humedad relativa máxima, mínima y media, velocidad del viento media y máxima, dirección del viento y radiación) cada una con su unidad de medida y filtro por estaciones meteorológicas con checkboxes.

El endpoint que hace la consulta a la base de datos es `/api/meteo-grafica`. Y al igual que en la anterior gráfica, se visualizan tantas líneas como estaciones se hayan seleccionado y en distintos colores.

4.1.5. Problemas encontrados y soluciones

En el desarrollo de la plataforma han surgido numerosos inconvenientes tanto en la configuración de las herramientas como en el propio desarrollo de las funciones de la web.

A continuación, se detallan algunos de los más relevantes y sus soluciones:

- **Uso erróneo de `range()` en InfluxDB:**

La obtención de datos de InfluxDB para mostrarlos en la web se hace a través de consultas. Cuando las consultas se hacían para obtener etiquetas, como provincias, estaciones, etc, la función `range` provocaba errores ya que dichas etiquetas carecían de una columna `_time`.

La solución fue simple. Para este tipo de consultas con funciones relacionadas con metadatos como `schema.tagValues()` se evitó el uso de `range()`.

- **Datos no cargados tras la migración a Docker:**

Inicialmente la web se ejecutó sobre el propio sistema operativo, es decir, en local. Para solucionar problemas de dependencias, finalmente se optó por migrar la aplicación a Docker. Esto ocasionó algunos problemas ya que la lógica que cargaba desde InfluxDB y mostraba los datos en la web dejó de funcionar.

La solución no fue tan simple. Tuvo que cambiarse el enfoque completamente, obteniendo ahora los datos de una solo vez. De esta forma, ahora el filtrado se hace directamente desde el frontend en lugar de depender de endpoints específicos.

▪ **Mal funcionamiento de los checkboxes en las gráficas:**

Para seleccionar las provincias, embalses y estaciones se hace uso de checkboxes. Por defecto cuando el usuario entraba en la web, se seleccionaban todas por defecto lo que recargaba masivamente la web y afectaba al rendimiento.

Esto se solucionó modificando el comportamiento por defecto de los checkboxes de forma que ninguna estación apareciese seleccionada. De esta forma, el usuario selecciona manualmente las estaciones a su gusto.

4.2. Procesos ETL

Los procesos ETL (*Extraction-Transformation-Loading*) son los todos las fases por las que los datos pasan hasta cargarse en una base de datos.

4.2.1. Fase de extracción

La base de este proyecto son los datos históricos de los embalses y las estaciones meteorológicas. Dichos datos proceden de distintas páginas webs de la Junta de Andalucía. Para la obtención de estos y su posterior uso para el aprendizaje de los modelos de *machine learning* se ha hecho uso de *Web Scraping*.

El *Web Scraping* es una técnica utilizada para extraer información de manera automática a partir de sitios web. A través de un programa o script, se navega por páginas web y se extra la información necesaria desde el código HTML, almacenándolo en un formato específico, como hojas de cálculo o bases de datos. [47]

La principal ventaja del *web scraping* es la capacidad de extraer una gran cantidad de datos en muy poco tiempo y de forma automática.

El funcionamiento general de esta técnica se resume en tres pasos:

1. **Solicitud a un servidor:**

El primer paso es la conexión entre el web scraper y el servidor que contiene la información a través del protocolo HTTP.

2. **Extraer y analizar el código HTML:**

El scraper lee y analiza el código HTML o XML de la página web para extraer los elementos predefinidos .

3. **Guardar los datos relevantes:**

Tras extraer la información, esta se almacena en un repositorio estructurado.

Respecto a las herramientas, hemos optado por usar Python. Python ofrece una gran cantidad de bibliotecas de código abierto que facilitan la extracción de datos a través de código. Algunas de ellas son:

- **Scrapy:** Framework basado en Python muy completo especialmente usado para la minería de datos y en el rastreo de datos históricos.
- **Pandas:** Utilizada para manipular e indexar datos. Se integra perfectamente con BeautifulSoup, de manera que se puede realizar todo el proceso en un solo lenguaje de programación.
- **Parsehub:** Es una herramienta online que permite extraer información de otras webs sin necesidad de código. Aunque la versión gratuita tiene funcionalidad limitada, es ideal para iniciarse y hacer pruebas.
- **Selenium:** Permite automatizar navegadores web y acceder al contenido dinámico generado por JavaScript. Es muy útil cuando las páginas requieren interacción con formularios, botones o inicios de sesión.

En nuestro caso, hemos usado tanto **Pandas** como **Selenium**.

- **Automatización:** Ahorro de tiempo y esfuerzo al automatizar el proceso de extracción de datos de manera automática.
- **Eficiencia:** Ideal para proyectos donde es necesario el procesamiento masivo de datos.

- **Acceso a datos en tiempo real:** Facilita la obtención de datos actualizados, lo que es crucial para el análisis de datos en tiempo real.
- **Flexibilidad:** Se puede aplicar a una gran variedad de tipos de datos y páginas web.
- **Recopilación de datos estructurados:** Conversión de datos no estructurados provenientes de páginas web en formatos estructurados, como CSV o JSON, haciendo el proceso de análisis más sencillo.

Extracción de datos meteorológicos

1. **Configuración del navegador y de las descargas.** Se inicializa Chrome en modo sin interfaz (headless) y con opciones para ejecución estable en entornos locales: desactivar *sandbox*, uso compartido de memoria del sistema, GPU, barras de información y extensiones; además de abrir maximizado. Se fijan las preferencias de descarga para: directorio por defecto, no pedir confirmación, permitir actualización de directorios y protección de *safebrowsing*.
2. **Arranque del WebDriver.** Se crea la instancia de Chrome usando un gestor automático del controlador (ChromeDriver) y un *WebDriverWait* de 30 segundos para sincronizar las esperas sobre la página.
3. **Acceso al listado de estaciones y cierre de cookies.** Se navega a la página de estaciones de la Red RIA y se espera a que cargue la tabla principal. Si aparece el aviso de cookies, se localiza el contenedor y se pulsa el botón de aceptar; si no está disponible, se continúa sin bloquear el flujo.
4. **Raspado de enlaces de estaciones.** Se recorre cada fila de la tabla, extrayendo el nombre de la estación y el enlace a su ficha individual. Para cubrir todas las estaciones, se itera por la paginación pulsando el botón “Siguiente” hasta que aparece deshabilitado. El resultado es una lista completa de pares (*nombre*, *URL*).
5. **Descarga de ficheros por estación.** Para cada estación:
 - a) Se abre su página y se accede a la sección “Descarga de Ficheros”.

- b) Se marcan las medidas a descargar (temperaturas, humedades, viento, radiación, precipitación y evapotranspiración), asegurando además la opción “Marcar todas”.
- c) Se pulsa el botón de “Descargar datos” y se espera unos segundos a que el archivo se guarde en el directorio de descargas.

6. **Gestión de los archivos descargados.** En la carpeta de descarga se identifica el archivo más reciente con extensión CSV o TXT. Se limpia el nombre de la estación (sustituyendo caracteres no permitidos en nombres de fichero) y se renombra el archivo con el patrón *<NombreEstacion>.csv*.
7. **Tolerancia a fallos y cierre.** Cualquier error puntual en una estación se captura y se registra, continuando con el resto para no interrumpir la extracción global. Finalmente, se cierra el navegador y se libera la sesión del WebDriver.

Extracción de datos de los embalses

1. **Inicialización del navegador.** Se configura Chrome con la opción de mantener la ventana abierta (detach) y se arranca el controlador de forma automática mediante la instalación del driver correspondiente.
2. **Acceso al portal.** Se navega a la página de embalses de REDIAM y se espera a que cargue el buscador principal de embalses.
3. **Obtención del listado de embalses.** Desde la lista de sugerencias asociada al cuadro de búsqueda se extraen todos los nombres de embalses disponibles, generando un listado completo para iterar.
4. **Utilidades de apoyo.** Se define una rutina para:
 - Detectar la finalización de una descarga en la carpeta configurada, comparando el contenido antes y después durante una ventana de tiempo.
 - Normalizar textos (acentos, mayúsculas, signos) con el fin de hacer comparaciones robustas entre el nombre solicitado y los títulos del mapa.
5. **Bucle de descarga con reintentos.** Para cada embalse del listado, y hasta un máximo de cinco intentos por elemento:

- a) Se escribe el nombre en el buscador y se confirma la búsqueda.
- b) Se localizan los marcadores del mapa y se selecciona el que coincide con el nombre del embalse tras la normalización de textos.
- c) Se hace clic en el marcador y se verifica, en la ventana emergente informativa, que el nombre mostrado corresponde al embalse buscado.
- d) Se pulsa el enlace de descarga del CSV asociado al embalse y se espera a que el archivo aparezca en la carpeta de descargas.
- e) Si la descarga se confirma, se contabiliza como éxito; en caso contrario, se vuelve a intentar dentro del límite establecido.

6. **Registro de resultados.** Se lleva la cuenta del número total de descargas correctas y se construye una lista con los embalses que no han podido descargarse tras agotar los reintentos, mostrándolos al final del proceso.

7. **Cierre ordenado.** Una vez recorridos todos los embalses, se cierra el navegador y se informa por consola del resumen: descargas exitosas frente al total y, en su caso, el listado de embalses fallidos.

4.2.2. Fase de transformación

Esta fase comprende tanto la transformación de los datos como la limpieza de los mismos. El objetivo es convertir el CSV bruto de embalses y datos meteorológicos en un fichero CSV compatible con InfluxDB 2.x, normalizando fechas, corrigiendo nombres con comas y tipando correctamente etiquetas (*tags*) y valores (*fields*).

Transformación de datos meteorológicos

A continuación se detallan los pasos y transformaciones por las que los datos meteorológicos han pasado:

1. **Normalización de nombres y detección de duplicados.** Se unifican nombres de fichero (acentos, guiones) y se marcan versiones duplicadas con sufijos (1), (2), etc. Ejemplos de renombrado: *Almunecar* → *Almuñécar (1)*, *Puerto_de_Santa_Maria* → *Puerto de Santa María (1)*, *Villanueva_del_Rio_y_Minis* → *Villanueva del Río y Minas (1)*.

2. **Fusión segura por estación.** Para cada nombre base (*Linares, Palma del Río, ...*) se localizan sus CSV con una expresión regular y:

- Si hay un único fichero, se parsea la columna *FECHA* (formato día/mes/año) y se guarda como *base_fusionado.csv*.
- Si hay varios, se leen todos, se igualan las cabeceras, se concatenan, se eliminan duplicados, se ordenan por *FECHA* y se guarda el resultado en *base_fusionado.csv*. Los originales del grupo se eliminan.

3. **Reorganización y renombrado final.** Los ficheros que terminan en *_fusionado.csv* se reescriben respetando el delimitador “;” y se renombran eliminando el sufijo *_fusionado* para dejar el nombre limpio (p. ej., *Linares.csv*).

4. Adaptación a InfluxDB

Para cada CSV final:

- 4.1. *Estandarización de cabeceras* a mayúsculas.
- 4.2. *Detección del prefijo* de estación a partir del nombre de las columnas (patrón [A-Z]+[0-9]+) y creación de *station_id*.
- 4.3. *Mapeo de provincia* con el prefijo de letras (*AL, CA, CO, GR, HU, JA, MA, SE*).
- 4.4. *Limpieza de nombres de variables*: eliminación del prefijo para conservar *TMAX, TMED, PRECIP*, etc.
- 4.5. *Normalización de valores*: las cadenas “n/d” y vacíos se pasan a valores nulos; en columnas de *fields* se emiten como nulos.
- 4.6. *Enriquecimiento de metadatos (tags)*: *station_id, provincia, station_name* y *time* (ISO-8601 UTC a partir de *FECHA*); si un tag falta, se rellena con “unknown”.
- 4.7. *Serialización*: se genera un CSV con línea de cabecera de tipos y filas del tipo *weather_data,station_id,provincia,station_name,TMAX,...,ETO,time*.

Transformación de datos de los embalses

A continuación se detallan los pasos y transformaciones por las que los datos de los embalses han pasado:

1. **Lectura robusta.** Se lee el CSV original con codificación ISO-8859-1 para preservar acentos y eñes.
2. **Reparación de nombres con comas.** Si tras separar por comas una fila contiene más de cinco elementos, se concatenan desde la quinta posición para reconstruir el nombre del embalse (que puede venir entrecomillado con comas internas).
3. **Limpieza de campos.** Se eliminan comillas sobrantes en *codigo* y *nombre*. La columna *fecha* se convierte a ISO-8601 UTC (YYYY-MM-DDTHH:MM:SSZ); las filas con fechas inválidas se omiten.
4. **Construcción de registros Influx.** Para cada fila válida se genera un registro con:
 - **measurement:** *embalses*.
 - **tags:** *codigo, nombre*.
 - **fields:** *reserva, porcentaje*.
 - **timestamp:** derivado de *fecha* en formato ISO-8601 UTC.
5. **Serialización** Se escribe un fichero con cabecera de tipos

#datatype measurement,tag,tag,field,field,timestamp

seguido de la línea de nombres y todas las observaciones.

4.2.3. Fase de carga

Esta es la fase de inserción en la base de datos. Según la implementación realizada, se emplean dos patrones de inserción diferenciados:

- **Datos meteorológicos:** se realiza una *carga masiva por fichero* (bulk). Para cada CSV se generan todas las líneas en formato Line Protocol y se envían en una única operación de escritura por fichero. Este enfoque reduce el número de peticiones y agrupa de forma explícita por archivo.
- **Embalses:** se usa *carga por lotes asíncrona* (batching). La escritura se configura con lotes de hasta 500 puntos o cada 10 s (lo que ocurra antes); al finalizar se fuerza el envío

pendiente y se cierra la conexión. De este modo, la librería acumula puntos en memoria y los envía en bloques, mejorando el rendimiento y la tolerancia a latencias.

Inserción de datos meteorológicos

A continuación se detallan los pasos para la inserción de los datos meteorológicos en InfluxDB:

1. **Preparación:** se recorren las carpetas con los CSV anotados y se llama a *insertar_datos_en_influxdb*.
2. **Lectura:** se leen los ficheros en UTF-8 y se saltan las dos primeras líneas (cabeceras).
3. **Tags:** se extraen *station_id*, *provincia* y *station_name*; se validan y se escapan espacios y comas (Line Protocol).
4. **Fields:** se generan pares *clave=valor* con variables meteorológicas; los vacíos no se incluyen y las filas sin campos válidos se omiten.
5. **Tiempo:** la columna temporal ISO-8601 Z se convierte a nanosegundos.
6. **Line Protocol:** se construye una línea por observación con *measurement*, *tags*, *fields* y *timestamp*. (*measurement* en el código: *weather_data*.)
7. **Escritura:** se envía el bloque de líneas con *write_api.write* al *bucket* y *org* configurados.
8. **Cierre:** se cierra la conexión con *client.close()*.

Inserción de datos de los embalses

A continuación se detallan los pasos para la inserción de los datos de embalses en InfluxDB:

1. **Entrada:** se utiliza el CSV ya convertido a CSV (*formato_influxdb_2025.csv*), con dos líneas de cabecera.
2. **Lectura:** se leen las líneas con *csv.reader* y se saltan las dos primeras (tipos y nombres).
3. **Extracción:** por fila válida (6 columnas) se obtienen *measurement*, *codigo*, *nombre*, *reserva*, *porcentaje* y *timestamp*. El *measurement* es *embalses*.

4. **Tags:** se limpian *codigo* y *nombre* (escape de espacios y comas para Line Protocol) y se forman los tags *codigo=...* y *nombre=...*
5. **Fields:** con *build_fields* se incluyen únicamente campos no vacíos: *reserva* y *porcentaje*.
6. **Tiempo:** *timestamp* (ISO-8601 en UTC) se convierte a nanosegundos para InfluxDB.
7. **Line Protocol:** se genera una línea por observación con la forma
embalses, codigo=..., nombre=... reserva=..., porcentaje=... timestamp_ns.
8. **Escritura:** se envían las líneas al *bucket* con *write_api.write(...)* usando escritura por lotes.
9. **Flush y cierre:** se fuerza el envío con *write_api.flush()*, se cierra el API de escritura y la conexión con *client.close()*.

4.2.4. Problemas encontrados y soluciones

El principal inconveniente que se encontró fue el cambio total de la web de los embalses de la Junta de Andalucía. Esto supuso un gran problema ya que la base de datos aún no constaba con los datos. La solución fue el desarrollo de un código totalmente distinto para la extracción automática de los datos de la nueva web.

Por lo tanto, hay dos código funcionales a día de hoy para los datos meteorológicos y los datos de los embalses y otro código que fue funcional hasta el cambio de la web.

4.3. Metodología de trabajo

Para el desarrollo de este TFG se siguió una metodología ágil de tipo Scrum, con ciclos de trabajo cortos (*sprints*) de aproximadamente dos semanas. Al inicio de cada sprint se planificaban las tareas priorizando aquellas con mayor importancia para el avance del proyecto. Este enfoque iterativo permitió incorporar cambios y ajustes con rapidez, apoyándose en reuniones quincenales con los tutores para revisar progreso, recoger retroalimentación y redefinir el *backlog* cuando era necesario.

Los primeros sprints se enfocaron en preparar el entorno y las bases del sistema: configuración de repositorios y control de versiones, definición del entorno de ejecución, y construcción del flujo de datos. En esta fase se integraron las fuentes (embalses y meteorología),

se resolvieron inconsistencias de nombres, se mapeó cada embalse a su provincia y se cargó la información en la base de series temporales InfluxDB. Paralelamente se establecieron validaciones básicas y pequeños tests para asegurar la calidad del dato antes de continuar.

En los sprints intermedios el foco pasó al procesamiento y a la parte de análisis: generación de agregados temporales (diario–mensual), creación de variables derivadas y preparación de conjuntos de entrenamiento. A partir de ahí se desarrollaron y compararon modelos predictivos basados en GRU, ajustando hiperparámetros. Cada iteración incluía experimentación controlada, evaluación con métricas y análisis de errores para orientar las siguientes mejoras.

Finalmente, los últimos sprints se dedicaron a la integración y entrega del producto: serialización de modelos y *scalers*, generación de predicciones, y construcción de una interfaz web en React para visualizar los datos históricos y sus gráficas. Se preparó un contenedor con Docker para facilitar el despliegue y la reproducibilidad del sistema. Al cierre de cada sprint se realizaba una revisión de lo conseguido y una breve retrospectiva para identificar mejoras del proceso de trabajo, manteniendo la documentación actualizada del proyecto y del código.

Este esquema ágil e incremental permitió avanzar de forma ordenada desde la obtención y tratamiento de datos hasta el proceso de entrenamiento de los modelos, incorporando de manera continua el feedback de tutores y las limitaciones en el desarrollo.

5

Conclusiones y trabajos futuros

Este proyecto se ha centrado en la gestión de los embalses andaluces: recolección y limpieza de datos (ETL y *web scraping*), consolidación en InfluxDB, creación de variables temporales (acumulados y medias móviles), entrenamiento de modelos GRU y una web que permite consultar datos históricos y su evolución. A nivel práctico, el modelo aprende patrones de llenado–vaciado y reacciona de forma coherente ante escenarios simulados de precipitación, temperatura o humedad, lo que lo convierte en una herramienta útil para explorar sensibilidades.

Todo esto presenta unos límites. La reserva no depende solo del clima: influyen decisiones de explotación (desembalses, demandas, trasvases), así como procesos hidrológicos que no se han tenido en cuenta, como la infiltración en el suelo, pérdidas a acuíferos y un balance entrada–salida detallado. También hay sensibilidad al post-procesado (escalados por embalse, transformaciones) y a la calidad de los datos extraídos de la web. En embalses muy regulados o con datos escasos y series ruidosas, el ajuste es razonable pero mejorable. Con todo, el trabajo demuestra que, con ingeniería de variables adecuada, una GRU bien regularizada y una interfaz accesible, es posible ofrecer una base funcional para previsión y escenarios que sirva de apoyo a la decisión.

A pesar de que este proyecto es totalmente funcional algunas de los trabajos futuros que se podrían realizar para mejorar los resultados son:

- **Scraping automatizado diario:** Programar las descargas con un orquestador (cron, Airflow o GitHub Actions), validación automática de ficheros y alertas si cambian las páginas fuente o faltan datos.

- **Más variables y mejor señal hidrológica:** Incorporar información de infiltración y pérdidas: humedad del suelo (satélite), índices lluvia–escorrentía, aportes/caudales, evaporación real, demandas y reglas de operación, área de cuenca y usos del suelo. Esto permitiría correlacionar mejor las variables que ya tenemos y por lo tanto mejorar la precisión de los escenarios what-if.
- **Arquitecturas mixtas:** Probar enfoques híbridos (modelo lluvia–escorrentía simple + GRU), stacking con árboles de gradiente, CNN+GRU para extraer patrones.
- **Web: noticias automáticas:** Añadir un módulo que agregue automáticamente noticias y avisos de embalses y meteorología en Andalucía con filtrado por provincia y embalse.
- **Web: descarga de datos:** Botón para exportar a CSV/JSON los datos filtrados y la imagen de las gráficas (PNG), con metadatos básicos (fuente, rango de fechas, versión del modelo).
- **Web: simulación interactiva:** Un panel in situ para generar escenarios what-if con modificadores y ver al momento la respuesta del embalse seleccionado.

Bibliografía

- [1] *Evidencia del cambio climático*. NASA. URL: <https://ciencia.nasa.gov/cambio-climatico/evidencia/#:~:text=En%201896%2C%20un%20art%C3%ADculo%20fundamental,a%20trav%C3%A9s%20del%20efecto%20invernadero> (visitado 01-09-2025).
- [2] *El proceso internacional de lucha contra el cambio climático: Naciones Unidas*. Ministerio para la Transición Ecológica y el Reto Demográfico (MITECO). URL: <https://www.miteco.gob.es/es/cambio-climatico/temas/el-proceso-internacional-de-lucha-contr-el-cambio-climatico/naciones-unidas.html#:~:text=En%201979%2C%20la%20primera%20Conferencia,internacionales%20para%20tratar%20el%20tema> (visitado 01-09-2025).
- [3] José Romero González y Ana M. Camarasa Belmonte. *Evidencias científicas sobre cambio climático y territorio en el Mediterráneo ibérico: Efectos, estrategias y políticas públicas. Principales recomendaciones*. Publicacions de la Universitat de València, 2025. URL: <https://roderic.uv.es/items/f51e77c6-8eec-43f4-bc9c-bd421b474ac4> (visitado 01-09-2025).
- [4] *¿Qué es la Inteligencia Artificial?* Google Cloud. URL: <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=es-419#what-is-artificial-intelligence-ai> (visitado 01-09-2025).
- [5] *Red de Información Agroclimática de Andalucía (RIA): Estaciones*. IFAPA (Junta de Andalucía). URL: <https://www.juntadeandalucia.es/agriculturaypesca/ifapa/riaweb/web/estaciones> (visitado 01-09-2025).
- [6] *Visor de embalses de Andalucía*. REDIAM (Junta de Andalucía). URL: <https://portalrediam.cica.es/embalses/> (visitado 01-09-2025).
- [7] Thomas Guerrero Gloria Amaris Humberto Ávila. “Aplicación de modelo ARIMA para el análisis de series de volúmenes anuales en el río Magdalena”. En: *Hydrology* (2017). URL: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-921X2017000200007 (visitado 01-09-2025).

- [8] Gabriel Koran, Tanya F. Heikkila et al. *Using Random Forest To Estimate Daily Base-Flow Metrics For California*. Scientific Investigations Report 2022–5005. U.S. Geological Survey, 2022. URL: <https://pubs.er.usgs.gov/publication/sir20225005> (visitado 01-09-2025).
- [9] Muhammad Afiq Zakaria, Muhammad Mubashar Yasin, Ahmad Fairuz Omar et al. “Application of XGBoost Algorithm towards River Streamflow and Water Level Prediction: A Systematic Literature Review”. En: *Water* 15.22 (2023), pág. 3942. DOI: [10.3390/w15223942](https://doi.org/10.3390/w15223942). URL: <https://www.mdpi.com/2073-4441/15/22/3942> (visitado 01-09-2025).
- [10] Frederik Kratzert et al. “Toward Improved Predictions in Ungauged Basins: Exploiting the Power of Large Scale Data and Deep Learning”. En: *Water Resources Research* 55.11 (2019). URL: <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019WR026065> (visitado 01-09-2025).
- [11] Rohit Pradhan, Subhankar Mishra, Manoj Kumar et al. “Streamflow forecasting with deep learning models: A side-by-side evaluation”. En: *Water Supply* 24.9 (2024), e13633. URL: <https://doi.org/10.2166/ws.2024.024> (visitado 01-09-2025).
- [12] M. Sit et al. “A comprehensive review of deep learning applications in hydrology and water resources”. En: *Water Science and Technology* 82.12 (2020), págs. 2635-2670. URL: <https://iwaponline.com/wst/article/82/12/2635/77673/A-comprehensive-review-of-deep-learning> (visitado 01-09-2025).
- [13] *WEAP (Water Evaluation And Planning System)*. Stockholm Environment Institute (SEI). URL: <https://www.weap21.org/> (visitado 01-09-2025).
- [14] *Centro de Gestión de Recursos Hídricos “Smart Water” en Andalucía*. Nota de prensa. Europa Press. 27 de mar. de 2023. URL: <https://www.iagua.es/noticias/junta-andalucia/andalucia-aprueba-implantacion-centro-gestion-recursos-hidricos-smart-water?> (visitado 01-09-2025).
- [15] *HydroForecast. Pronóstico de caudales con Machine Learning*. Upstream Tech. URL: <https://www.upstream.tech/hydroforecast> (visitado 01-09-2025).

- [16] *Redes neuronales recurrentes (RNN)*. IBM. URL: <https://www.ibm.com/es-es/topics/recurrent-neural-networks> (visitado 01-09-2025).
- [17] *LSTM: memoria a largo y corto plazo*. DataScientest. URL: <https://datascientest.com/es/memoria-a-largo-plazo-a-corto-plazo-lstm> (visitado 01-09-2025).
- [18] *¿Qué es una GRU?* VPN Unlimited. URL: <https://www.vpnunlimited.com/es/help/cybersecurity/gru?srsltid=AfmB0op3nJFeHvDexJEvIZ95bnvL11G2fpfuJ7289RY6BydJ5s4Qvc4g> (visitado 01-09-2025).
- [19] *Modelos de Machine Learning: tipos y usos*. QuestionPro. URL: <https://www.questionpro.com/blog/es/modelos-de-machine-learning/> (visitado 01-09-2025).
- [20] Redacción iAgua. *Sequía en Andalucía: haciendo frente a una crisis hídrica en escalada*. iAgua. URL: <https://www.iagua.es/noticias/redaccion-iagua/sequia-andalucia-haciendo-frente-crisis-hidrica-escalada> (visitado 01-09-2025).
- [21] *La sequía en la comunidad andaluza no es solo por la falta de lluvias*. Ecologistas en Acción. URL: <https://www.ecologistasenaccion.org/308259/la-sequia-en-la-comunidad-andaluza-no-es-solo-por-la-falta-de-lluvias/> (visitado 01-09-2025).
- [22] *Día Mundial del Turismo 2019*. eldiario.es (branded). URL: <https://branded.eldiario.es/dia-mundial-turismo-2019> (visitado 01-09-2025).
- [23] Redacción iAgua. *Turismo y sostenibilidad hídrica: una relación dependiente*. iAgua. URL: <https://www.iagua.es/noticias/redaccion-iagua/turismo-y-sostenibilidad-hidrica-relacion-dependiente> (visitado 01-09-2025).
- [24] *Ciclo integral del agua de uso urbano*. Junta de Andalucía. URL: <https://www.juntadeandalucia.es/medioambiente/portal/areas-tematicas/agua/infraestructura-hidraulica-y-actuaciones/ciclo-integral-del-agua-de-uso-urbano> (visitado 01-09-2025).
- [25] *Datos Espaciales de Referencia de Andalucía (DERA)*. Instituto de Estadística y Cartografía de Andalucía (IECA). URL: <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/datos-espaciales-de-referencia-de-andalucia-dera> (visitado 01-09-2025).

- [26] *Red de Información Agroclimática de Andalucía (RIA) — ¿Quieres saber más?* IFAPA (Junta de Andalucía). URL: <https://www.juntadeandalucia.es/agriculturaypesca/ifapa/riaweb/web/quieressabermas> (visitado 01-09-2025).
- [27] *Sensor PT1000: características y funcionamiento.* Resistencias Eléctricas. URL: <https://www.resistencias-electricas.com/productos/control-de-temperatura/resistencias-termicas/pt1000/#:~:text=El%20sensor%20PT1000%20tiene%20un,el%20valor%20C3%B3hmico%20del%20cable> (visitado 01-09-2025).
- [28] *Dirección del viento: definición.* Meteo Navarra. URL: <https://meteo.navarra.es/definiciones/viento.cfm#:~:text=Direcci%C3%B3n%20del%20viento&text=Se%20mide%20en%20grados%2C%20desde,sur%20y%20270%C2%BA%20del%20oeste> (visitado 01-09-2025).
- [29] *¿Qué es la evapotranspiración (ETO) y cómo interpretar el dato?* Sencrop. URL: <https://faq.sencrop.com/es/articles/6103455-que-es-la-evapotranspiracion-y-como-interpretar-el-dato> (visitado 01-09-2025).
- [30] *¿Qué es Python?* Amazon Web Services. URL: <https://aws.amazon.com/es/what-is/python/> (visitado 01-09-2025).
- [31] *¿Qué es InfluxDB y primeros pasos.* OpenWebinars. URL: <https://openwebinars.net/blog/que-es-influxdb-y-primeros-pasos> (visitado 01-09-2025).
- [32] *¿Qué es JavaScript?* OpenWebinars. URL: <https://openwebinars.net/blog/que-es-javascript/> (visitado 01-09-2025).
- [33] *Introduction to JavaScript (traducción al español).* Launch School. URL: https://launchschool-com.translate.google/books/javascript/read/introduction?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq (visitado 01-09-2025).
- [34] *What's the origin of VS Code?* Microsoft Learn. URL: <https://learn.microsoft.com/es-es/shows/one-dev-minute/whats-the-origin-of-vs-code> (visitado 01-09-2025).
- [35] *Google Colab: qué es y cómo usarlo.* Arsys. URL: <https://www.arsys.es/blog/google-colab-que-es-y-como-usarlo> (visitado 01-09-2025).

- [36] *About GitHub and Git*. GitHub. URL: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git> (visitado 01-09-2025).
- [37] *Docker en AWS (introducción)*. Amazon Web Services. URL: <https://aws.amazon.com/es/docker> (visitado 01-09-2025).
- [38] *Docker: todo lo que hay que saber*. DataScientest. URL: <https://datascientest.com/es/docker-todo-que-saber> (visitado 01-09-2025).
- [39] *¿Qué es Node.js?* Kinsta. URL: <https://kinsta.com/es/base-de-conocimiento/que-es-node-js> (visitado 01-09-2025).
- [40] *¿Qué es Bootstrap? Funciones básicas de Bootstrap*. Hostinger. URL: https://www.hostinger.com/es/tutoriales/que-es-bootstrap#Funciones_basicas_de_Bootstrap (visitado 01-09-2025).
- [41] *Qué es Bootstrap: historia y tu primer Hola Mundo*. Nubecolectiva. URL: <https://nubecolectiva.com/blog/que-es-bootstrap-historia-y-tu-primer-hola-mundo> (visitado 01-09-2025).
- [42] *¿Qué es Express?* Kinsta. URL: <https://kinsta.com/es/base-de-conocimiento/que-es-express> (visitado 01-09-2025).
- [43] *HTML: Lenguaje de marcas de hipertexto*. MDN Web Docs. URL: <https://developer.mozilla.org/es/docs/Web/HTML> (visitado 01-09-2025). Véase también: Hostinger, «¿Qué es HTML?», <https://www.hostinger.com/es/tutoriales/que-es-html>.
- [44] *¿Qué es CSS? Introducción*. Lenguaje CSS. URL: <https://lenguajecss.com/css/introduccion/que-es-css> (visitado 01-09-2025).
- [45] *What-If Tool para explicabilidad: revisión*. EITCA Academy. URL: <https://es.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/google-cloud-ai-platform/using-the-what-if-tool-for-explainability/examination-review-using-the-what-if-tool-for-explainability/how-does-the-what-if-tool-help-users-understand-the-behavior-of-their-machine-learning-models/> (visitado 01-09-2025).

- [46] *Previsión de escenarios: cómo utilizar el análisis What-If*. FasterCapital. URL: <https://fastercapital.com/es/contenido/Prevision-de-escenarios--como-utilizar-el-analisis-What-If-para-planificar-diferentes-resultados-y-riesgos.html> (visitado 01-09-2025).
- [47] *Web scraping: qué es, funcionamiento y ventajas*. ESIC. URL: <https://www.esic.edu/rethink/tecnologia/web-scraping-que-es-funcionamiento-ventajas-c#:~:text=El%20web%20scraping%20es%20una,c%C3%A1culo%20o%20bases%20de%20datos> (visitado 01-09-2025).

Apéndice A

Manual de Usuario

La finalidad de este manual es el de guiar el usuario para que explore las distintas funcionalidades de la plataforma.

1. Página de Inicio

La página de inicio (figura 87) es la página principal de la web. Contiene un carrusel de imágenes de distintos embalses andaluces. Además, contiene una barra de búsqueda donde el usuario podrá buscar por palabras, noticias referentes a la gestión del agua y la meteorología.

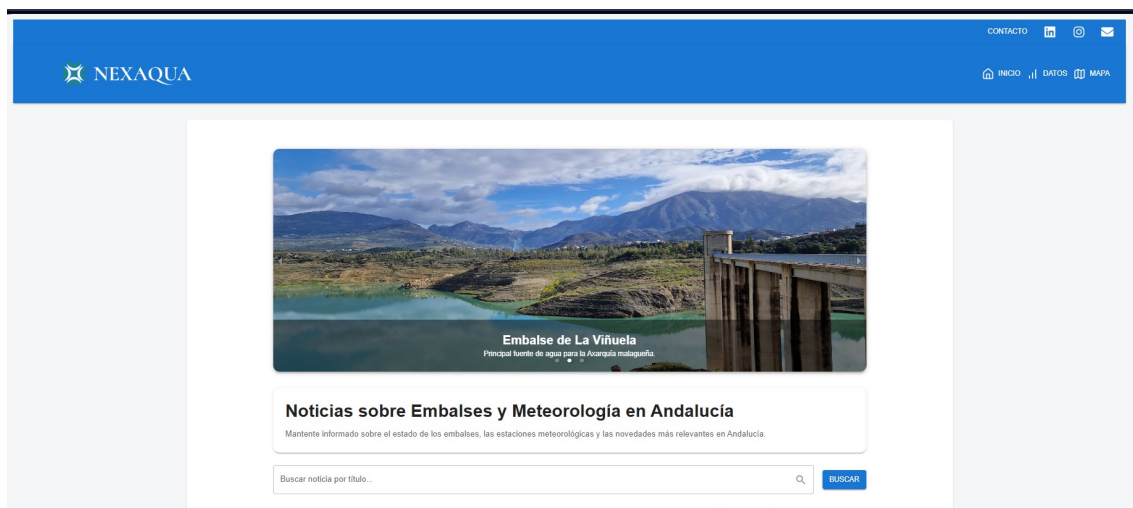


Figura 87: Ventana de inicio

En la figura 88 se pueden ver las noticias actuales que hay en la web y un botón de "Ver más" que llevará al usuario a la correspondiente noticia.

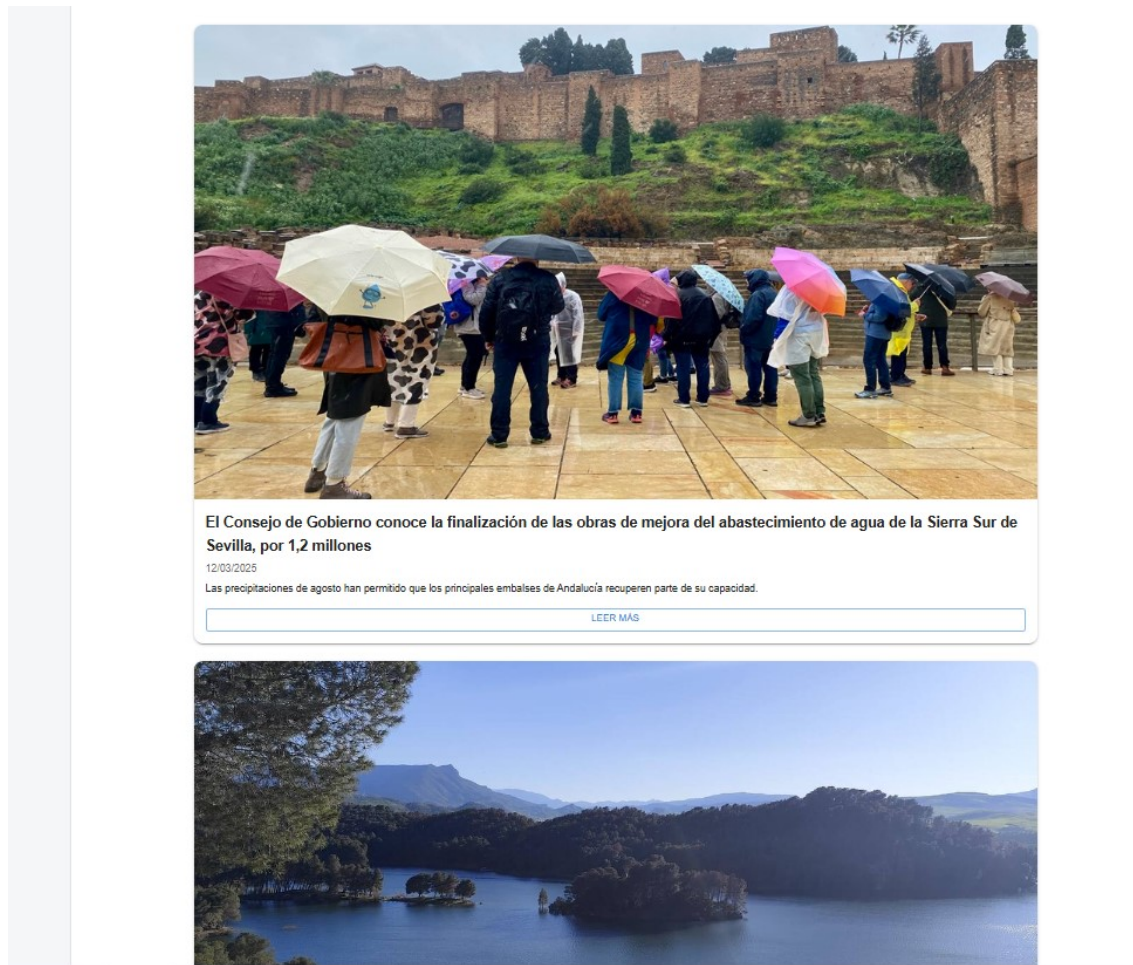


Figura 88: Ventana de inicio noticias

2. Páginas de avisos legales

En el footer de la web, figura 89, se incluyen las pestañas de avisos legales, políticas de cookies, accesibilidad y protección de datos. Estas páginas proporcionan al usuario información sobre las políticas legales de la web.

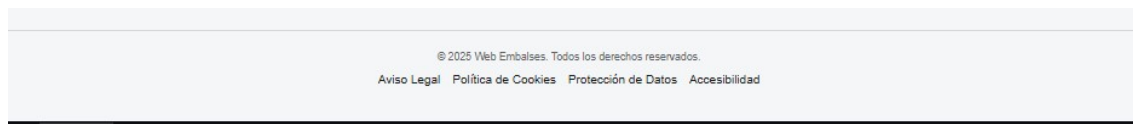


Figura 89: Pestañas de avisos legales

Cada una de las siguientes imágenes 90 91 92 93 muestran las pestañas personalizadas de nuestras políticas legales.

Aviso Legal

El presente sitio web, **NEXAQUA**, tiene como finalidad ofrecer información y herramientas relacionadas con la gestión de embalses y la meteorología en Andalucía.

Titularidad

Este sitio web es propiedad de **Patricia Martín Castillo** y **Rodrigo Hernández Barba**, con correos de contacto en: 0619993622@uma.es y 0619894427@uma.es.

Condiciones de uso

La utilización de este sitio web implica la aceptación plena de las condiciones aquí expuestas. El usuario se compromete a utilizar el portal de conformidad con la legislación vigente y con el presente Aviso Legal, absteniéndose de realizar cualquier actividad que pueda resultar ilícita o lesiva para terceros.

Propiedad intelectual e industrial

Los contenidos de esta web, incluidos textos, imágenes, logotipos, diseño gráfico y código fuente, están protegidos por los derechos de propiedad intelectual e industrial, y no podrán ser reproducidos, distribuidos ni transformados sin autorización expresa del titular.

Responsabilidad

NEXAQUA no se hace responsable del uso indebido de los contenidos de la web, ni de posibles daños o perjuicios derivados de la falta de disponibilidad o continuidad del sitio.

Figura 90: Ventana de aviso legal

Protección de Datos y Privacidad

En cumplimiento con el Reglamento (UE) 2016/679 de Protección de Datos (RGPD) y la normativa española aplicable, se informa a los usuarios de lo siguiente:

Responsable del tratamiento

Patricia Martín Castillo y **Rodrigo Hernández Barba**.

Correo electrónico de contacto: 0619993622@uma.es y 0619894427@uma.es.

Finalidad del tratamiento

Los datos personales recogidos a través de los formularios de contacto o comunicación serán utilizados únicamente para atender las consultas planteadas por los usuarios y mejorar la experiencia de navegación en el sitio web.

Legitimación

La base legal para el tratamiento de los datos es el **consentimiento del usuario**, otorgado al remitir su consulta o al utilizar el sitio.

Destinatarios

No se cederán datos a terceros, salvo en los casos en que exista una **obligación legal**.

Derechos de los usuarios

Los usuarios pueden ejercer sus derechos de acceso, rectificación, supresión, limitación y oposición escribiendo a las direcciones de correo electrónico indicadas: 0619993622@uma.es o 0619894427@uma.es.

Para procesar la solicitud será necesario adjuntar una copia de un documento acreditativo de identidad.

Conservación de los datos

Los datos personales se conservarán únicamente durante el tiempo necesario para atender la consulta o petición realizada, y mientras el usuario no solicite su supresión.

Figura 91: Ventana de protección de datos

Política de Cookies

Este sitio web utiliza cookies propias y de terceros con el objetivo de mejorar la experiencia de usuario, analizar el tráfico y ofrecer un servicio más personalizado.

¿Qué son las cookies?

Las cookies son pequeños archivos de texto que se almacenan en el navegador del usuario al visitar un sitio web. Sirven para reconocer al usuario en futuras visitas, recordar sus preferencias o recopilar información estadística sobre la navegación.

Tipos de cookies utilizadas

- **Cookies técnicas:** necesarias para el correcto funcionamiento de la web y la prestación de los servicios solicitados por el usuario.
- **Cookies de análisis:** nos ayudan a comprender cómo interactúan los usuarios con la página, midiendo y analizando la audiencia con fines estadísticos.
- **Cookies de personalización:** permiten recordar preferencias de navegación como idioma, región o ajustes elegidos por el usuario.

Gestión de cookies

El usuario puede configurar, bloquear o eliminar las cookies instaladas en su equipo a través de las opciones de configuración de su navegador:

- [Google Chrome](#)
- [Mozilla Firefox](#)
- [Microsoft Edge](#)
- [Safari](#)

Actualizaciones de la política

La presente política de cookies puede ser actualizada para adaptarse a nuevas exigencias normativas o a cambios en la configuración del sitio web. Recomendamos a los usuarios revisarla de forma periódica.

Figura 92: Ventana de política de cookies

Accesibilidad

En NEXAQUA estamos comprometidos con la accesibilidad digital y el diseño universal, promoviendo la igualdad de oportunidades y la no discriminación. Nuestro objetivo es que cualquier persona pueda utilizar este sitio web con independencia de sus capacidades, su dispositivo o su contexto de uso.

Marco y nivel de conformidad

Trabajamos para cumplir con los criterios de conformidad de las **Pautas de Accesibilidad para el Contenido Web (WCAG) 2.1**, nivel **AA**, y la normativa vigente en materia de accesibilidad digital.

Medidas adoptadas

- **Estructura semántica** del contenido (encabezados, listas y etiquetas correctamente anidadas).
- **Texto alternativo** en imágenes y controles no textuales relevantes.
- **Contraste de color** suficiente para garantizar la legibilidad.
- **Navegación por teclado** y **focus** visible en elementos interactivos.
- **Uso de etiquetas y descripciones** en formularios e inputs.
- Evitamos contenido que **parpadee** o pueda provocar molestias.
- Compatibilidad progresiva con **lectores de pantalla** y ayudas técnicas habituales.

Contenidos y tecnologías

Este sitio emplea HTML5, CSS, JavaScript y componentes React/Material UI. Intentamos mantener la compatibilidad con los navegadores modernos más utilizados y con diferentes tamaños de pantalla.

Observaciones y contacto

La accesibilidad es un proceso de mejora continua. Si encuentras alguna barrera o necesitas la información en un formato alternativo, escríbenos a:

- 0619993622@uma.es
- 0619894427@uma.es

Limitaciones conocidas

Aunque realizamos revisiones periódicas, podrían existir páginas o componentes con:

- Etiquetas ARIA o textos alternativos pendientes de revisión puntual.
- Tablas con gran volumen de datos que requieren mejoras de navegación para lectores de pantalla.
- Elementos de terceros (por ejemplo, mapas incrustados) con controles no totalmente personalizables.

Si detectas algún problema, por favor avísanos para priorizar su corrección.

Fecha de la declaración

Esta declaración de accesibilidad se actualizó el **1 de septiembre de 2025**.

Figura 93: Ventana de accesibilidad

3. Cabecera de la web

En la cabecera (figura 94) se encuentran los botones que llevan al usuario a las distintas páginas de tablas, mapas y gráficas. Además, se han añadido botones en la parte superior del *header* para que el usuario pueda contactar con nosotros mediante un bloque de contacto (Figura 95), junto con nuestras redes sociales para mantenerse al día de las últimas noticias de nuestra web.

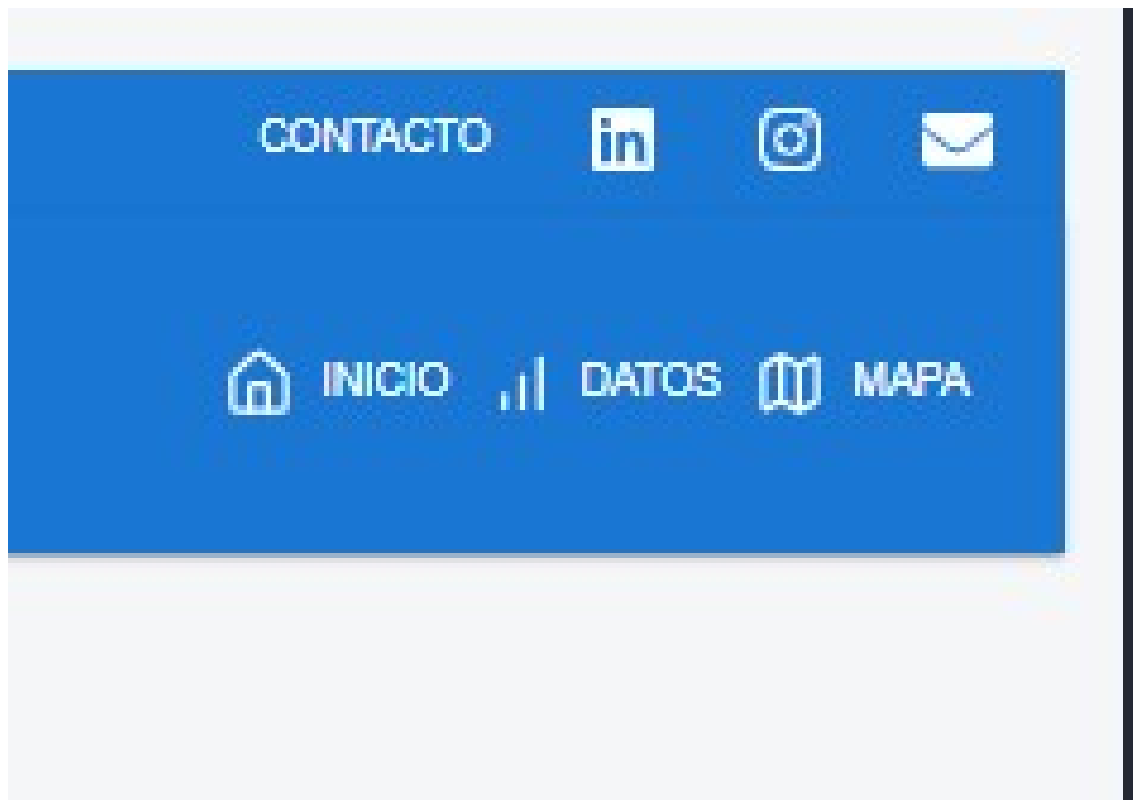


Figura 94: Header

The image shows a contact form titled "Formulario de Contacto" in bold black text. Below the title is a short paragraph: "Si tienes dudas, sugerencias o necesitas información, rellena el formulario y nos pondremos en contacto contigo." The form contains four input fields: "Nombre:" with a text box, "Correo Electrónico:" with a text box, "Asunto:" with a text box, and "Mensaje:" with a larger text area. At the bottom of the form is a dark blue button with the word "Enviar" in white text.

Figura 95: Ventana de contacto

Las páginas más importantes de la web son las pestañas de "Datos". La figura 96 muestra el desplegable que contiene las páginas de las tablas (embalses y datos meteorológicos) y las gráficas.



Figura 96: Desplegable

Por último, recalcar que la web cuenta con un logo y un nombre personalizado que el usuario puede clicar para volver a la página de inicio [97](#).



Figura 97: Logo y nombre de la página web

4. Páginas de datos

En estas páginas se encuentran las tablas con las que el usuario puede interactuar para filtrar datos y gráficas a su gusto.

En las imágenes [98](#) y [99](#) podemos observar las tablas y los filtros. Cada tabla está en páginas distintas y tiene sus filtros propios.

Como las tablas se pueden hacer bastante grandes, si el usuario quiere ir abajo del todo o arriba sin esfuerzo, se han incluido dos flechas abajo a la derecha que puede clicar para llevarlo directamente.

Además, el usuario puede elegir la cantidad de filas a mostrar en la tabla junto con un sistema de paginación para navegar entre la información.

Todas estas características aplican tanto a la tabla de embalses como a la de datos meteorológicos.

Datos de Embalses

RANGO DE FECHAS FECHA ÚNICA
 Fecha Inicio: Fecha Fin:

Fecha	Embalse	Reserva (hm³)	Porcentaje (%)
2024-04-12	AGUASCEBAS	6.39 hm³	100%
2024-04-13	AGUASCEBAS	6.39 hm³	100%
2024-04-12	ALMODÁVAR	4.66 hm³	82%
2024-04-13	ALMODÁVAR	4.85 hm³	82%
2024-04-12	ANDÁVALO	220.47 hm³	35%
2024-04-13	ANDÁVALO	220.42 hm³	35%
2024-04-12	ARACENA	75.28 hm³	56%
2024-04-13	ARACENA	75.33 hm³	56%
2024-04-12	ARCOS	12.86 hm³	88%
2024-04-13	ARCOS	12.86 hm³	88%

Filas por página: 10 - 1-10 of 148 < >

© 2025 Web Embalses. Todos los derechos reservados.
 Aviso Legal Política de Cookies Protección de Datos Accesibilidad

Figura 98: Ventana tabla de embalses

Datos Meteorológicos

RANGO DE FECHAS FECHA ÚNICA
 Fecha Inicio: Fecha Fin:
 Filtrar por Provincia Filtrar por Estación

Fecha	Provincia	Estación	Temp. Media (°C)	Temp. Máx (°C)	Temp. Mín (°C)	Humedad Media (%)	Humedad Máx (%)	Humedad Mín (%)
2024-04-12	Almería	Adra	18.70°C	21.10°C	12.20°C	66.2%	86.3%	46.6%
2024-04-13	Almería	Adra	18.10°C	22.40°C	13.70°C	57.9%	72.4%	36.5%
2024-04-12	Jaén	Alcaudete	18.90°C	27.50°C	10.70°C	41.6%	61.7%	23.1%
2024-04-13	Jaén	Alcaudete	18.90°C	26.90°C	11.80°C	38.3%	55.5%	20.9%
2024-04-12	Almería	Almería	17.00°C	21.40°C	11.00°C	61.0%	86.1%	36.1%
2024-04-13	Almería	Almería	19.00°C	25.40°C	12.40°C	53.2%	81.1%	30.3%
2024-04-12	Huelva	Almonte	19.00°C	26.30°C	10.20°C	63.5%	91.6%	28.8%
2024-04-13	Huelva	Almonte	19.40°C	27.80°C	10.80°C	60.9%	80.6%	28.3%
2024-04-12	Huelva	Almonte bajo plástico	19.00°C	36.80°C	6.20°C	66.3%	100.0%	20.9%
2024-04-13	Huelva	Almonte bajo plástico	19.90°C	37.20°C	6.70°C	63.7%	100.0%	15.6%

Filas por página: 10 - 1-10 of 150 < >

Figura 99: Ventana tabla de datos meteorológicos

Dentro del desplegable de datos también encontramos la página de las gráficas. El funcionamiento es idéntico al de las tablas. El usuario puede seleccionar los valores que prefiera en los filtros para mostrar las gráficas.

En las figuras 100 y 101 se muestra un ejemplo tras rellenar los filtros.

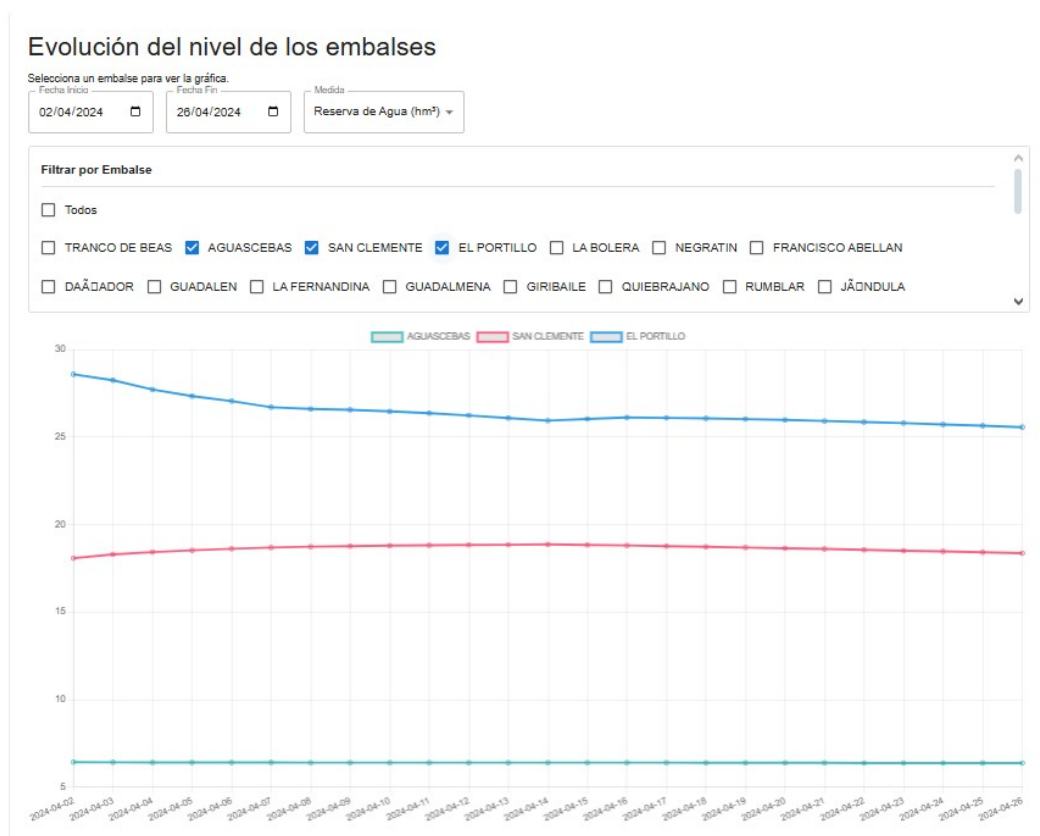


Figura 100: Ventana de grafica de embalses

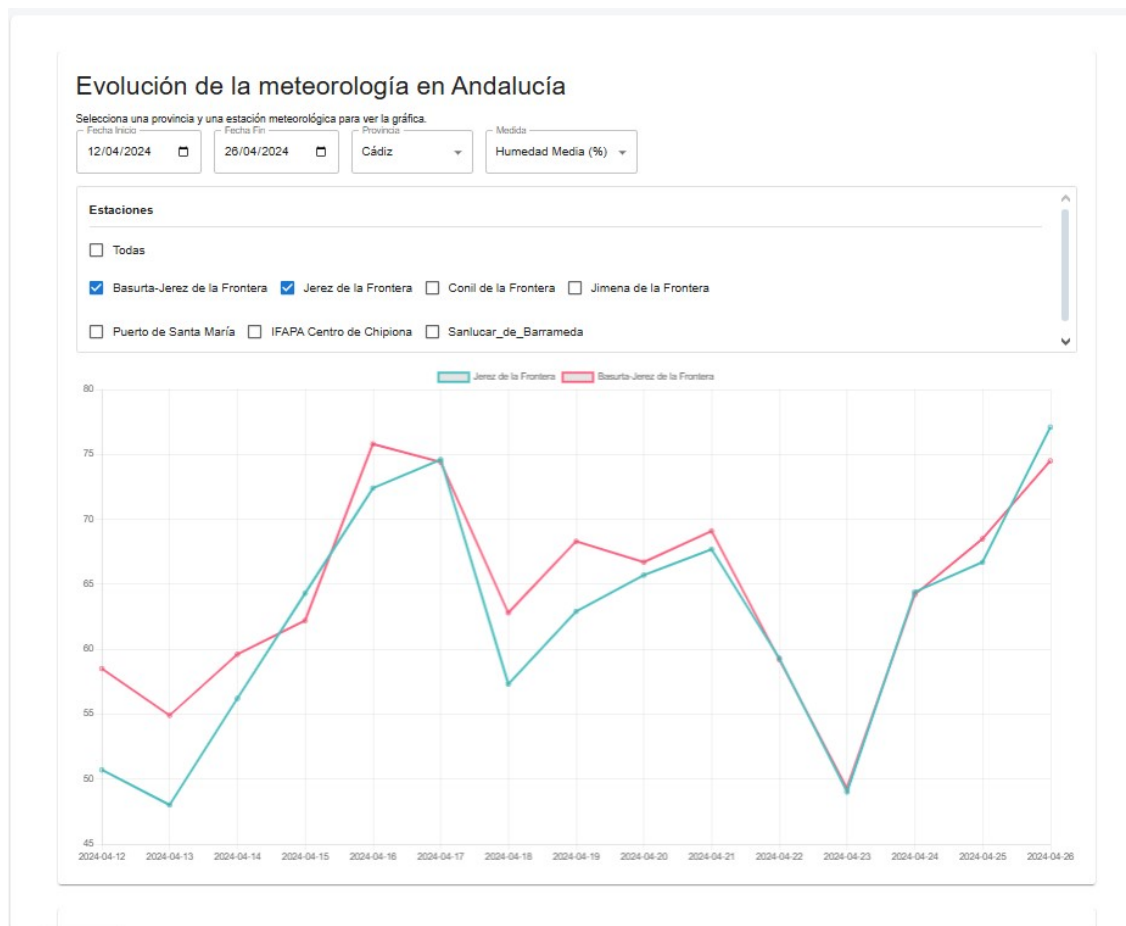


Figura 101: Ventana de grafica de datos meteorologicos

5. Mapa de Andalucía con embalses y estaciones meteorológicas

Entre los botones del header se encuentra el del mapa (figura 102). Este mapa contiene la localización exacta de todos de los embalses y de las estaciones meteorológicas que hay en Andalucía. Los iconos rojos representan las estaciones meteorológicas y los azules los embalses. Estos iconos son clicables por lo que el usuario puede hacer clic en todos para obtener información.



Figura 102: Ventana de mapa

Apéndice B

Contexto del estudio (Tablas)

Estas tablas presentan la clasificación de los embalses de Andalucía por demarcaciones:

▪ Demarcación del Guadalquivir

Embalse	Uso	Río	Provincia	Sistema
Aguascebas	Abastecimiento	Arroyo De Las Aguascebas Chico	Jaén	Sistema De Regulacion General
Aracena	Abastecimiento, Hidroeléctrico	Rivera De Huelva	Huelva	Abastecimiento De Sevilla
Arenoso	Abastecimiento, Hidroeléctrico, Riego	Arenoso	Córdoba	Sistema De Regulacion General
Bembezar	Abastecimiento, Defensa Frente Avenidas, Riego, Ganadero	Bembezar	Córdoba	Bembezar- Retortillo
Cala	Abastecimiento, Hidroeléctrico, Ganadero	Rivera De Cala	Sevilla	Abastecimiento De Sevilla

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Canales	Abastecimiento, Defensa Frente Avenidas, Ganadero, Hidroelectico, Riego	Genil	Granada	Alto Genil
Colomera	Abastecimiento, Defensa Frente Avenidas, Riego, Ganadero	Colomera	Granada	Alto Genil
Cubillas	Riego, Hidroelectico, Abastecimiento	Cubillas	Granada	Alto Genil
Dañador	Abastecimiento	Dañador	Jaén	Sistema De Regulacion General
El Agrio	Riego, Industrial	Cripinejo O Agrio	Sevilla	Almonte- Marismas
El Pintado	Abastecimiento, Riego	Viar	Sevilla	Viar
El Portillo	Abastecimiento, Defensa Frente Avenidas, Riego	Castril	Granada	Alto Guadiana Menor
Francisco Abellán	Abastecimiento, Riego	Fardes	Granada	Hoya De Guadix
Gergal	Abastecimiento	Rivera De Huelva	Sevilla	Abastecimiento De Sevilla
Giribaile	Abastecimiento, Defensa Frente Avenidas, Riego, Hidroeléctrico	Guadalimar	Jaén	Sistema De Regulacion General

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Guadalen	Abastecimiento, Defensa Frente Avenidas, Riego, Hidroeléctrico, Ganadero	Guadalen	Jaén	Sistema De Regulacion General
Guadalmellato	Abastecimiento, Hidroeléctrico, Riego	Guadalmellato	Córdoba	Sistema De Regulacion General
Guadalmena	Abastecimiento, Hidroeléctrico, Riego	Guadalmena	Jaén	Sistema De Regulacion General
Guadalupe	Abastecimiento, Defensa Frente Avenidas	Guadalupe	Córdoba	Sistema De Regulacion General
Huesna	Abastecimiento, Defensa Frente Avenidas, Ganadero	Rivera De Huesna	Sevilla	Rivera Del Huesna
Iznajar	Abastecimiento, Defensa Frente Avenidas, Hidroeléctrico, Riego, Ganadero	Genil	Córdoba	Sistema De Regulacion General
Jándula	Abastecimiento, Riego, Hidroeléctrico, Ganadero	Jándula	Jaén	Sistema De Regulacion General
Jose Toran	Abastecimiento, Defensa Frente Avenidas, Riego	Arroyo De Guadalbacar	Sevilla	Sistema De Regulacion General

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
La Bolera	Abastecimiento, Defensa Frente Avenidas, Riego	Guadalentin	Jaén	Alto Guadiana Menor
La Breña	Abastecimiento, Defensa Frente Avenidas, Hidroeléctrico, Riego	Guadiato	Córdoba	Sistema De Regulacion General
La Fernandina	Abastecimiento, Hidroeléctrico, Riego	Guarriza	Jaén	Sistema De Regulacion General
La Minilla	Abastecimiento, Defensa Frente Avenidas	Rivera De Huelva	Sevilla	Abastecimiento De Sevilla
Los Bermejales	Abastecimiento, Defensa Frente Avenidas, Riego, Hidroeléctrico, Ganadero	Cacin	Granada	Alto Genil
Martin Gonzalo	Abastecimiento, Regulacion	Arroyo De Martin Gonzalo	Córdoba	Sistema De Regulacion General
Melonares	Abastecimiento, Riego	Viar	Sevilla	Abastecimiento De Sevilla
Negratín	Abastecimiento, Defensa Frente Avenidas, Riego, Hidroeléctrico	Guadiana Menor	Granada	Sistema De Regulacion General

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Puebla De Cazalla	Riego	Corbones	Sevilla	Sistema De Regulacion General
Puente Nuevo	Abastecimiento, Hidroeléctrico, Riego	Guadiato	Córdoba	Sistema De Regulacion General
Quéntar	Riego, Hidroelectico, Abastecimiento	Aguas Blancas	Granada	Alto Genil
Quiebrajano	Abastecimiento	Quiebrajano	Jaén	Abastecimiento De Jaén
Retortillo	Abastecimiento, Riego	Retortillo	Córdoba	Bembezar- Retortillo
Rumblar	Abastecimiento, Hidroeléctrico, Riego	Rumblar	Jaén	Rumblar
San Clemente	Riego	Guardal	Granada	Alto Guadiana Menor
San Rafael De Navallana	Abastecimiento, Defensa Frente Avenidas, Riego	Guadalmellato	Córdoba	Sistema De Regulacion General
Sierra Boyera	Riego, Hidroeléctrico	Guadiato	Córdoba	Sistema De Regulacion General
Torre Del Aguila	Riego	Arroyo Salado De Moron	Sevilla	Sistema De Regulacion General
Tranco De Beas	Abastecimiento, Hidroeléctrico, Riego	Guadalquivir	Jaén	Sistema De Regulacion General

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Vadomojon	Riego, Hidroeléctrico	Guadajoz	Córdoba	Sistema De Regulacion General
Viboras	Abastecimiento	Viboras	Jaén	Abastecimiento De Jaén
Yeguas	Riego, Hidroeléctrico	De Las Yeguas	Córdoba	Sistema De Regulacion General
Zufre	Abastecimiento, Hidroeléctrico	Rivera De Huelva	Huelva	Abastecimiento De Sevilla

Tabla 12: Embalses de la demarcación GUADALQUIVIR

■ **Demarcación de Guadalete-Barbate**

Embalse	Uso	Río	Provincia	Sistema
Almodovar	Abastecimiento, Defensa Frente Avenidas	Almodovar	Cadiz	Barbate
Arcos	Abastecimiento, Defensa Frente Avenidas	Guadalete	Cadiz	Guadalete
Barbate	Abastecimiento	Barbate	Cadiz	Barbate
Bornos	Abastecimiento, Defensa Frente Avenidas	Guadalete	Cadiz	Guadalete
Celemin	Abastecimiento	Celemin	Cadiz	Barbate

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Guadalcacín	Abastecimiento, Defensa Frente Avenidas	Majaceite	Cádiz	Guadalete
Los Hurones	Abastecimiento, Defensa Frente Avenidas, Ganadero	Majaceite	Cádiz	Guadalete
Zahara-El Gastor	Abastecimiento	Guadalete	Cádiz	Guadalete

Tabla 13: Embalses de la demarcación GUADALETE–BARBATE

■ **Demarcación de Tinto-Odiel-Piedras**

Embalse	Uso	Río	Provincia	Sistema
Andevalo	Abastecimiento, Defensa Frente Avenidas	Rivera De Malagon	Huelva	Tinto-Odiel- Piedras (Cuenca Del Guadiana)
Chanza	Abastecimiento, Defensa Frente Avenidas, Ganadero	Rivera De Chanza	Huelva	Tinto-Odiel- Piedras (Cuenca Del Guadiana)
Corumbel Bajo	Abastecimiento, Defensa Frente Avenidas	Corumbel	Huelva	Tinto-Odiel- Piedras (Tinto)
Jarrama	Abastecimiento, Defensa Frente Avenidas	Rivera Del Jarrama	Huelva	Tinto-Odiel- Piedras (Tinto)
Machos	Abastecimiento, Defensa Frente Avenidas	Piedras	Huelva	Tinto-Odiel- Piedras (Piedras)

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Piedras	Abastecimiento, Defensa Frente Avenidas	Piedras	Huelva	Tinto-Odiel- Piedras (Piedras)

Tabla 14: Embalses de la demarcación TINTO–ODIEL–PIEDRAS

■ **Demarcación del Mediterráneo**

Embalse	Uso	Río	Provincia	Sistema
Beninar	Abastecimiento, Defensa Frente Avenidas	Adra	Almeria	Sierra Nevada
Beznar	Abastecimiento, Defensa Frente Avenidas, Ganadero	Izbor	Granada	Sierra Nevada
Casasola	Abastecimiento, Defensa Frente Avenidas	Campanillas	Málaga	Serranía De Ronda
Charco Redondo	Abastecimiento, Defensa Frente Avenidas, Ganadero	Las Cañas O Palmones	Cadiz	Serranía De Ronda
Concepcion	Abastecimiento	Verde	Málaga	Serranía De Ronda
Conde De Guadalhorce	Abastecimiento	Turon	Málaga	Serranía De Ronda
Cuevas De Almanzora	Abastecimiento, Defensa Frente Avenidas, Ganadero	Almanzora	Almeria	Sierra de Los Filabres- Estancias

Continúa en la siguiente página

Embalse	Uso	Río	Provincia	Sistema
Guadalhorce	Abastecimiento, Defensa Frente Avenidas, Ganadero	Guadalhorce	Málaga	Serranía De Ronda
Guadalteba	Abastecimiento, Defensa Frente Avenidas, Ganadero	Guadalteba	Málaga	Serranía De Ronda
Guadarranque	Abastecimiento, Defensa Frente Avenidas, Ganadero	Guadarranque	Cádiz	Serranía De Ronda
Limonero	Abastecimiento, Defensa Frente Avenidas	Guadalmedina	Málaga	Serranía De Ronda
Rules	Abastecimiento, Defensa Frente Avenidas, Ganadero	Guadalfeo	Granada	Sierra Nevada
Viñuela	Abastecimiento, Defensa Frente Avenidas	Guaro	Málaga	Sierra de Tejeda Y Almirajara

Tabla 15: Embalses de la demarcación MEDITERRÁNEO

Esta tabla presenta las estaciones meteorológicas Andaluzas clasificadas por provincias:

Almería	Cádiz	Córdoba	Granada
Adra	Basurta–Jerez de la Frontera	Adamuz	Almuñécar
Almería	Conil de la Frontera	Baena	Baza

Continúa en la siguiente página

Almería	Cádiz	Córdoba	Granada
Cuevas de Almanzora	IFAPA Centro de Chipiona	Belmez	Cádiar
Fiñana	Jerez de la Frontera	Córdoba	Huéneja
Huércal-Overa	Jimena de la Frontera	El Carpio	IFAPA Centro Camino del Purchil
La Mojonera	Puerto de Santa María	Hornachuelos	Iznalloz
Níjar	Sanlúcar de Barrameda	IFAPA Centro de Cabra	Jerez del Marquesado
Tabernas	Villamartín	IFAPA Centro de Hinojosa del Duque	Loja
Virgen de Fátima–Cuevas de Almanzora	Vejer de la Frontera	Palma del Río	Puebla de Don Fabrique
Tíjola		Santaella	Pinos Puente Zafarraya Padul Pinos Puente Casanueva

Tabla 16: Estaciones por provincia (I)

Huelva	Jaén	Málaga	Sevilla
Almonte	Alcaudete	Antequera	Aznalcázar
Almonte bajo plástico	Chiclana de Segura	Archidona	Ecija
Almonte bajo plástico frb	Huesa	Cártama	Guillena

Continúa en la siguiente página

Huelva	Jaén	Málaga	Sevilla
Aroche	IFAPA Centro Mengíbar	Estepona	IFAPA Centro Las Torres–Tomejil
El Campillo	Jaén	IFAPA Centro de Campanillas	IFAPA Centro Las Torres–Tomejil. Finca Tomejil
El Tojalillo–Gibraleón	Jódar	IFAPA Churriana	IFAPA Centro de Los Palacios
Gibraleón	La Higuera de Arjona	Málaga	Isla Mayor
Gibraleón – Manzorales	Linares	Vélez–Málaga	La Luisiana
IFAPA Centro de Huelva. El Cebollar bajo plástico	Mancha Real	Sierra Yeguas	La Puebla del Río
IFAPA Centro Huelva. Finca el Cebollar	Marmolejo	Pizarra	La Puebla del Río II
La Palma del Condado	Mengíbar		La Rinconada
La Puebla de Guzmán	Pozo Alcón		Las Cabezas de San Juan
Lepe	San José de los Propios		Lebrija I
Lepe bajo plástico	Sabiote		Lebrija II
Moguer	Torreblascopedro		Lora del Río
Moguer El Fresno	Ubeda		Los Molares
Niebla	Santo Tomé		Los Palacios y Villafranca
Palos de la Frontera bajo plástico	Torreperogil		Villanueva del Río y Minas

Continúa en la siguiente página

Huelva	Jaén	Málaga	Sevilla
	Villacarrillo		Villafranco del Guadalquivir Osuna Sanlúcar la Mayor Puebla Cazallar

Tabla 17: Estaciones por provincia (II)



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA