



Universidad Politécnica  
de Madrid

**Escuela Técnica Superior de  
Ingenieros Informáticos**



Grado en Matemáticas e Informática

Trabajo Fin de Grado

**Desarrollo de Modelos de Deep  
Learning para la Implementación de  
Soft Sensor de Variables en Procesos  
Industriales**

Autor: Paula Ramírez Rodríguez  
Tutora: Belén Ríos Sánchez  
Cotutor: Freddy José Perozo Rondón

Madrid, Junio 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

*Trabajo Fin de Grado*  
*Grado en Matemáticas e Informática*

*Título:* Desarrollo de Modelos de Deep Learning para la Implementación  
de Soft Sensor de Variables en Procesos Industriales

Junio 2025

*Autor:* Paula Ramírez Rodríguez  
*Tutora:* Belén Ríos Sánchez  
Departamento de Matemática Aplicada a las Tecnologías de la  
Información y las Comunicaciones  
Escuela Técnica Superior de Ingenieros Informáticos  
Universidad Politécnica de Madrid  
*Cotutor:* Freddy José Perozo Rondón  
Departamento de Matemáticas Avanzadas  
Repsol Technology Lab

# Resumen

El auge de la inteligencia artificial en la última década ha revolucionado la forma en la que se abordan los procesos industriales. Su aplicación en sectores como la energía o la petroquímica ha permitido optimizar operaciones, reducir costes y mejorar la calidad de los productos. Uno de los desafíos existentes en estos entornos es la monitorización de variables críticas que, en ocasiones, por motivos técnicos, económicos o de seguridad, no pueden ser medidas en tiempo real. En instalaciones complejas como las de la empresa Repsol, existen casos en los que el acceso directo a algunas variables es limitado. En consecuencia, se ha generado la necesidad de estimar ciertas variables indirectamente, y es en este contexto donde surgen los llamados *soft sensors*.

Un *soft sensor* o sensor virtual, permite inferir el valor de variables críticas o no accesibles a partir de otras disponibles de forma continua. Se trata de modelos que pueden desarrollarse de dos maneras, desde un enfoque físico, basado en principios fundamentales del proceso, o puramente basado en datos. Este último está cada vez más extendido gracias a la disponibilidad de históricos operativos y al avance de técnicas de inteligencia artificial basadas en Machine Learning y Deep Learning. A pesar de su potencial, se plantean una serie de desafíos al tratar con datos de procesos industriales reales: datos ruidosos o incompletos, relaciones complejas entre variables, retardos temporales y cambios en condiciones de operación.

El objetivo principal de este proyecto es desarrollar un *soft sensor* basado en modelos de Deep Learning para predecir en línea variables de control en una columna de destilación de C3-C4 de una planta industrial de Repsol. La solución propuesta se apoya en datos operativos reales proporcionados por la empresa.

La primera fase del proyecto incluye un análisis exploratorio detallado, selección de variables relevantes, limpieza de datos, interpolación de valores atípicos y escalado de todas las señales. Se han definido diez variables clave, agrupadas según su función operativa, y se ha seleccionado un intervalo reciente de 45 días, representativo del comportamiento habitual de la planta, con un total de 64.800 observaciones.

En la segunda fase se ha realizado el modelado de la solución propuesta. Se han implementado distintas arquitecturas buscando abarcar diferentes enfoques, con el fin de compararlos posteriormente. El proceso comienza con modelos *baseline*, que sirven como referencia para establecer una base. A partir de

---

ellos, se continúa con algoritmos de Machine Learning basados en árboles de decisión como Random Forest y XGBoost, así como redes neuronales del estado del arte para problemas temporales: LSTM, TCN, DLinear, NLinear y Temporal Fusion Transformer (TFT).

Todos estos modelos son evaluados en distintos horizontes de predicción (hasta 45 minutos) sobre el conjunto de *test*, aplicando métricas estándar de regresión. Además, se generan visualizaciones temporales que permiten identificar la alineación entre las predicciones y los valores reales, tanto en tramos amplios como en ventanas puntuales. Este análisis no solo valora el rendimiento global, sino que también facilita identificar cuándo y por qué cada modelo resulta más o menos preciso, así como detectar patrones de error según el tipo de variable o la dinámica del proceso.

Las arquitecturas LSTM, DLinear y NLinear han demostrado un mejor rendimiento que modelos más avanzados como la TFT. Estas han alcanzado valores de MAPE de entre un 0,36% y 1,5%, y coeficientes de determinación ( $R^2$ ) muy cercanos a 1 en horizontes de predicción cortos, aunque con cierta degradación a medida que se amplía el rango de predicción. Los resultados obtenidos muestran que hay un gran potencial en el uso de estos modelos en entornos industriales. Se refleja la capacidad real del sistema para anticipar el comportamiento del proceso, sin necesidad de recurrir a modelos más sofisticados.

El trabajo proporciona una metodología completa y replicable para el desarrollo de *soft sensors* basados en datos, adaptada a las características específicas de un entorno industrial real. Además, abre la puerta a futuras líneas de trabajo orientadas a la incorporación de modelos probabilísticos, la integración directa de estos sistemas en planta o su aplicación en simulaciones de procesos.

**Palabras claves:** *Soft sensors, Deep Learning, series temporales, optimización de procesos, inteligencia artificial industrial, modelado basado en datos.*

# Abstract

The rise of artificial intelligence in the last decade has revolutionized the way in which industrial processes are approached. Its application in sectors such as energy or petrochemicals has enabled the optimization of operations, cost reduction and improvement in product quality. One of the existing challenges in these environments is the monitoring of critical variables that, due to technical, economic or safety reasons, cannot always be measured in real time. In complex facilities such as those operated by Repsol, there are cases where direct access to certain variables is limited. Consequently, the need has arisen to estimate some variables indirectly, which is where so-called soft sensors come into play.

A soft sensor, or virtual sensor, infers the value of complex or inaccessible variables based on others that are continuously available. These models can be developed in two main ways: using a physics based approach, grounded in fundamental process principles, or purely data-driven. The latter has become increasingly widespread due to the availability of operational historical data and advances in artificial intelligence techniques based on Machine Learning and Deep Learning techniques. Despite their potential, several challenges arise when working with real industrial process data: noisy or incomplete data, complex relationships between variables, time delays and changes in operating conditions.

The main objective of this project is to develop a soft sensor based on Deep Learning models to predict control variables in real time in a C3-C4 distillation column from Repsol industrial plant. The proposed solution relies on real operational data provided by company.

The first phase of the project includes a detailed exploratory data analysis, selection of relevant variables, data cleaning, outlier interpolation and signal scaling. Ten key variables are defined, grouped according to their operational function, and a recent 45-day interval is selected, representative of the plant's usual behavior, with a total of 64,800 observations.

In the second phase, the proposed solution is modeled. Several architectures are implemented to cover different approaches, allowing for subsequent comparison. The modeling begins with baseline models that serve as a reference. From there, the work progresses to classical Machine Learning algorithms based on decision trees, such as Random Forest and XGBoost, as well as state-of-the-art neural network architectures for time series problems: LSTM, TCN, DLinear, NLinear and Temporal Fusion Transformer (TFT).

---

All these models are evaluated across different prediction horizons (up to 45 minutes) on the test set, using standard regression metrics. Temporal visualizations are also generated to assess alignment between predictions and actual values, both across longer sequences and within specific time windows. This made it possible not only to evaluate overall performances but also to identify when each model is more or less accurate and to analyze error patterns depending on variable type or process dynamics.

LSTM, DLinear, NLinear architectures demonstrate better performance than more advanced models like TFT. These models achieved MAPE values between 0.36% and 1.5%, and determination coefficients ( $R^2$ ) close to 1 for short prediction horizons, though performance declined as the forecast range increased. The results highlight the strong potential of these models in industrial environments. The system proves capable of anticipating process behavior without the need for more sophisticated architectures.

This work provides a complete and replicable methodology for developing data-based soft sensors, tailored to the specific characteristics of a real industrial setting. Moreover, it opens the door to future work focused on incorporating probabilistic models, integrating these systems directly into the plant, or applying them in process simulations.

**Keywords:** *Soft sensors, Deep Learning, time series, process optimization, industrial artificial intelligence, data-driven modeling.*

# ÍNDICE

<b>1. Introducción y objetivos</b>	<b>1</b>
1.1. Introducción	1
1.2. Objetivos	4
1.3. Estructura	4
<b>2. Estado del arte</b>	<b>7</b>
2.1. Técnicas de modelado	8
2.1.1. Enfoques basados en ML y DL	8
2.1.2. Enfoques estadísticos y clásicos	9
2.2. Aplicaciones en la industria	11
<b>3. Fundamento teórico</b>	<b>15</b>
3.1. Series temporales	15
3.2. Modelos estudiados	18
3.2.1. Modelos <i>baseline</i>	18
3.2.2. Clásicos de Machine Learning	19
3.2.3. Arquitecturas de Deep Learning	21
3.3. Estrategias de entrenamiento y predicción	28
3.3.1. Teacher forcing	28
3.3.2. Predicción autorregresiva	29
<b>4. Definición del problema</b>	<b>31</b>
4.1. Columna de destilación	31
4.2. Datos históricos	34
4.2.1. Librerías empleadas en el análisis exploratorio	34
4.2.2. Descripción del conjunto de datos inicial	34
4.2.3. Características temporales	36
4.2.4. Análisis exploratorio de las variables	38
4.2.5. Visualizaciones complementarias	43
<b>5. Modelado de la solución</b>	<b>49</b>
5.1. Preparación de los datos	49
5.1.1. Preprocesamiento y limpieza	49
5.1.2. División y escalado de datos	52
5.2. Librerías para el desarrollo de los modelos	53
5.2.1. Optimización de hiperparámetros	54

5.3. Evaluación del modelo . . . . .	54
5.3.1. Estrategia de predicción . . . . .	54
5.3.2. Métricas de evaluación y visualizaciones . . . . .	55
<b>6. Resultados</b>	<b>59</b>
6.1. Configuración común de los modelos . . . . .	59
6.2. Modelos <i>baseline</i> y Random Forest . . . . .	60
6.3. XGBoost . . . . .	62
6.4. LSTM . . . . .	65
6.5. TCN . . . . .	69
6.6. DLinear y NLinear . . . . .	72
6.7. TFT . . . . .	76
6.8. Comparación conjunta . . . . .	79
<b>7. Conclusiones y trabajo futuro</b>	<b>83</b>
<b>8. Análisis de impacto</b>	<b>87</b>
<b>Bibliografía</b>	<b>91</b>
<b>Anexos</b>	<b>99</b>
<b>A. Gráficas adicionales</b>	<b>99</b>
A.1. XGBoost . . . . .	99
A.2. LSTM . . . . .	100
A.3. TCN . . . . .	101
A.4. DLinear y NLinear . . . . .	102
A.5. TFT . . . . .	103

# ÍNDICE DE TABLAS

4.1. Variables seleccionadas del <i>dataset</i> . . . . .	36
4.2. Estadísticas descriptivas de las variables . . . . .	39
4.3. Valores nulos y negativos por variable . . . . .	40
4.4. Valores atípicos por variable . . . . .	43
6.1. Métricas de evaluación de NaiveCopy . . . . .	62
6.2. Comparación de métricas entre XGBoost optimizado y su configuración inicial . . . . .	63
6.3. Comparación de métricas entre LSTM optimizado y su configuración inicial . . . . .	67
6.4. Comparación de métricas entre TCN optimizado y su configuración inicial . . . . .	70
6.5. Comparación de métricas entre DLinear y NLinear optimizados y sus configuraciones iniciales . . . . .	74
6.6. Métricas de evaluación de TFT . . . . .	79
6.7. Métricas de evaluación del mejor modelo en cada caso . . . . .	80



# ÍNDICE DE FIGURAS

1.1. Evolución del uso de IA en el sector empresarial en los últimos 7 años . . . . .	1
1.2. Estimación del mercado de la IA . . . . .	2
3.1. Ejemplo de representación de una serie temporal . . . . .	16
3.2. Descomposición de una serie temporal . . . . .	17
3.3. Arquitectura de Random forest . . . . .	19
3.4. Arquitectura de XGBoost . . . . .	20
3.5. Diagrama de una red neuronal recurrente sencilla . . . . .	21
3.6. Arquitectura de una celda LSTM . . . . .	22
3.7. Capa convolucional de una dimensión . . . . .	23
3.8. Capa convolucional causal y dilatada . . . . .	23
3.9. Red TCN básica . . . . .	24
3.10. Funcionamiento del modelo DLinear . . . . .	25
3.11. Predicción con múltiples horizontes . . . . .	26
3.12. Arquitectura de TFT . . . . .	27
3.13. <i>Teacher forcing</i> y predicción autorregresiva . . . . .	28
4.1. Esquema simplificado del proceso industrial . . . . .	32
4.2. Series temporales completas de las diez variables seleccionadas . . . . .	37
4.3. Datos atípicos en cada una de las series temporales . . . . .	42
4.4. Mapa de calor de correlaciones . . . . .	44
4.5. Distribuciones de las variables seleccionadas . . . . .	45
4.6. Estacionalidad anual de las variables controladas . . . . .	46
4.7. Estacionalidad mensual de las variables controladas . . . . .	47
4.8. Estacionalidad diaria de las variables controladas . . . . .	47
5.1. Series temporales tras el preprocesado . . . . .	51
5.2. Ejemplo de la división en los conjuntos de entrenamiento, validación y test, y escalado de la variable 'PC_tope' . . . . .	53
6.1. Predicciones de RF en tres horizontes . . . . .	61
6.2. Predicción puntual de modelos baseline y RF sobre una ventana de 45 minutos . . . . .	61
6.3. Predicciones de XGBoost en tres horizontes . . . . .	64
6.4. Predicción puntual de XGBoost sobre una ventana de 45 minutos . . . . .	65
6.5. Predicciones de LSTM en tres horizontes . . . . .	67

## ÍNDICE DE FIGURAS

---

6.6. Predicción puntual de LSTM sobre una ventana de 45 minutos . . .	68
6.7. Predicciones de TCN base en tres horizontes . . . . .	71
6.8. Predicciones de TCN optimizada en tres horizontes . . . . .	71
6.9. Predicción puntual de TCN sobre una ventana de 45 minutos . . .	72
6.10 Predicciones de DLinear en tres horizontes . . . . .	75
6.11 Predicciones de NLinear en tres horizontes . . . . .	75
6.12 Predicción puntual de DLinear y NLinear sobre una ventana de 45 minutos . . . . .	76
6.13 Predicciones de TFT en tres horizontes . . . . .	77
6.14 Predicciones de TFT con regularización en tres horizontes . . . . .	78
6.15 Predicción puntual de TFT sobre una ventana de 45 minutos . . . .	78
A.1. Primera ventana de 45 minutos adicional de XGBoost . . . . .	99
A.2. Segunda ventana de 45 minutos adicional de XGBoost . . . . .	100
A.3. Tercera ventana de 45 minutos adicional de XGBoost . . . . .	100
A.4. Primera ventana de 45 minutos adicional de LSTM . . . . .	100
A.5. Segunda ventana de 45 minutos adicional de LSTM . . . . .	101
A.6. Tercera ventana de 45 minutos adicional de LSTM . . . . .	101
A.7. Primera ventana de 45 minutos adicional de TCN . . . . .	101
A.8. Segunda ventana de 45 minutos adicional de TCN . . . . .	102
A.9. Tercera ventana de 45 minutos adicional de TCN . . . . .	102
A.10 Primera ventana de 45 minutos adicional de DLinear y NLinear . .	102
A.11 Segunda ventana de 45 minutos adicional de DLinear y NLinear . .	103
A.12 Tercera ventana de 45 minutos adicional de DLinear y NLinear . .	103
A.13 Primera ventana de 45 minutos adicional de TFT . . . . .	103
A.14 Segunda ventana de 45 minutos adicional de TFT . . . . .	104
A.15 Tercera ventana de 45 minutos adicional de TFT . . . . .	104

# Capítulo 1

## Introducción y objetivos

### 1.1. Introducción

La revolución de la **inteligencia artificial** (IA) en los últimos años ha cambiado y transformado drásticamente la industria moderna. Actualmente, el 77% de las empresas a nivel mundial están utilizando o explorando la IA [1]. El mercado global de esta tecnología experimentó un crecimiento del 33% en 2024, con una proyección del 38% para este año [2].

Encuestas realizadas en 2024 reflejan claramente este aumento en su uso en el sector empresarial (Figura 1.1).

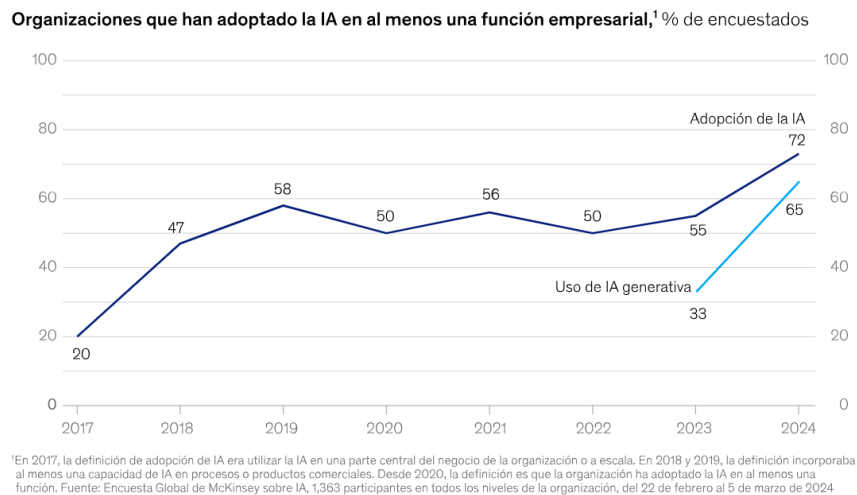


Figura 1.1: Evolución del uso de IA en el sector empresarial en los últimos 7 años [3]

El interés en la IA tiene un alcance verdaderamente global. Las inversiones en inteligencia artificial siguen en aumento, impulsadas por la automatización y la demanda de modelos más sofisticados [1].

- El mercado global de IA superará los 407 mil millones de dólares en 2027,

## Capítulo 1. Introducción y objetivos

---

con un crecimiento anual del 36,8%.

- Las empresas tecnológicas han triplicado su gasto en IA desde 2020, centrándose en infraestructura, modelos de lenguaje y automatización.

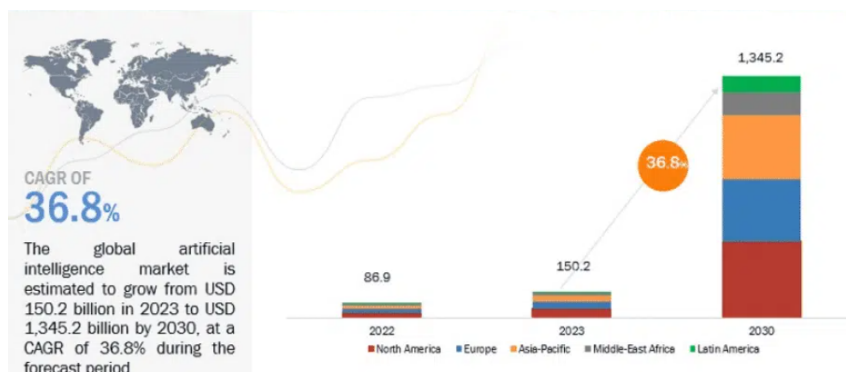


Figura 1.2: Estimación del mercado de la IA [1]

Hoy en día, está presente en sectores muy diversos: automoción, sanidad, finanzas, entre otros [4]. Se ha convertido en un factor fundamental en el ámbito industrial, no solo por su capacidad para mejorar la eficiencia y la seguridad en las operaciones, sino también por su impacto en la sostenibilidad y la reducción de costos [5].

Su aplicación va mucho más allá de las funciones tradicionales, como la generación de texto o el reconocimiento de imágenes, alcanzando áreas estratégicas como el control de sistemas. Proporciona soluciones innovadoras que abarcan desde la automatización de tareas rutinarias hasta la optimización de procesos pesados y complejos [5].

A raíz de esto, se está desarrollando un nuevo modelo industrial basado en la digitalización, que combina los principales avances en la automatización industrial con las tecnologías de la información [6]. Transformar las estructuras tradicionales permite vincular el mundo físico con el virtual, facilitando una toma de decisiones más precisa basada en datos en tiempo real.

Dentro de esta transformación, las simulaciones de entornos industriales se han convertido en un aspecto clave. Los conocidos como gemelos digitales, o *digital twins*, son un modelo virtual que refleja con exactitud un objeto físico, proceso o sistema. Esta tecnología permite operar y analizar un sistema virtual con la misma precisión que con uno real [6]. Se pueden realizar pruebas, optimizar procesos y predecir el comportamiento del sistema sin tener que intervenir de manera directa en el entorno físico, reduciendo significativamente posibles errores.

Además, las empresas están experimentando importantes beneficios financieros, una toma de decisiones más informada y un notable incremento en la productividad, consolidando así el valor estratégico de los gemelos digitales en diversos sectores.

Se prevé que el despliegue de estos experimente un crecimiento medio del 36% en cinco años [7].

En el sector energético, empresas como Repsol están integrando dichas tecnologías con el objetivo de mejorar la eficiencia operativa, la calidad del producto y reducir costos. Uno de los mayores desafíos a los que se enfrentan en este tipo de entornos industriales complejos es la sensorización y monitorización de variables críticas en procesos. En algunos casos, el uso de los sensores físicos necesarios para medir dichas variables puede resultar inviable, ya sea por limitaciones técnicas, económicas o temporales. En otras ocasiones, algunos de estos parámetros no pueden ser medidos en línea y necesitan un análisis de laboratorio, lo que implica retardos que afectan al proceso [8]. Para remediar esta problemática, se ha impulsado el uso de sensores virtuales, conocidos como ***soft sensors***.

Los *soft sensors* [9], son un mecanismo que permite inferir y estimar (en línea) variables difíciles de medir directamente, esenciales para el control y mantenimiento de los procesos. Son modelos virtuales que pueden desarrollarse de una forma básica, teniendo en cuenta estados ideales, o más avanzada, utilizando datos más completos que representan todo el sistema. Se emplean diferentes métodos para predecir las variables deseadas, desde modelos matemáticos tradicionales hasta técnicas avanzadas de aprendizaje automático.

Gracias a los avances mencionados en aprendizaje automático y minería de datos, la aplicación de estos sensores virtuales se ha afianzado como una solución eficaz [8]. Sin embargo, aunque presentan grandes ventajas, también se enfrentan a retos significativos. La calidad y disponibilidad de los datos necesarios para desarrollarlos se pueden ver afectadas por diferentes factores: la presencia de datos incompletos, ruido en las señales, retardos en métricas, entre otros aspectos. Además, la extracción de características relevantes suele conllevar cierta complejidad. Procesos como pueden ser el refinamiento de petróleo o la producción de polímeros presentan dinámicas altamente no lineales y dependientes de factores externos, lo que dificulta la selección de las variables de entrada óptimas [10].

Por otra parte, cambios en las condiciones operativas, como variaciones en las características de la materia prima o fluctuaciones en la demanda, pueden afectar a la estabilidad y capacidad de generalización de los modelos [9]. En este sentido, el preprocesamiento de los datos es fundamental; la forma en la que se limpian y transforman puede impactar directamente en la precisión de los modelos predictivos específicamente seleccionados [11].

En este contexto, el presente trabajo plantea el desarrollo e implementación de un sistema de *soft sensor* basado en modelos de Deep Learning. Su propósito es estimar en tiempo real variables críticas en un entorno industrial operativo. Para ello, se trabajará con datos reales de proceso proporcionados por Repsol, obtenidos de una columna de destilación fraccionada diseñada para separar propano (C3) de butano (C4) basándose en sus diferentes puntos de ebullición.

### 1.2. Objetivos

A partir de esta base, se establecen una serie de objetivos orientados a cubrir todas las etapas del desarrollo del sistema propuesto, desde la preparación de los datos hasta la validación del modelo en condiciones cercanas a la operación real. Se incluye tanto el tratamiento y modelado de **series temporales**, como la evaluación comparativa de distintas **arquitecturas de Deep Learning**, con el fin de obtener un modelo robusto, preciso y aplicable a datos de similares características a los empleados. A continuación, se detallan las principales líneas de trabajo:

- Realizar un estudio del proceso industrial objeto de análisis, lo que facilitará tanto la elaboración de los modelos como la posterior interpretación de los resultados.
- Desarrollar un enfoque sólido para el preprocesamiento y la preparación de datos de series temporales industriales.
- Explorar distintas arquitecturas de Deep Learning, como redes neuronales recurrentes (RNN) o redes convolucionales temporales (TCN), comparando su capacidad para modelar series temporales industriales complejas. Validar dichos modelos utilizando conjuntos de datos reales de un proceso industrial, definiendo métricas específicas para su evaluación.
- Determinar modelos robustos y precisos que permitan inferir variables difíciles de medir directamente, mejorando así la eficiencia y el control del proceso.
- Identificar las limitaciones y posibles áreas de mejora del modelo, sentando las bases para líneas futuras.

Con este enfoque, se pretende cubrir todas las fases necesarias para el desarrollo del sensor, aplicando una metodología rigurosa que garantice su correcta implementación.

### 1.3. Estructura

Este documento está organizado para facilitar la comprensión del trabajo realizado, desde los antecedentes hasta los resultados obtenidos, combinando teoría y aplicación práctica. A continuación, se describe cómo se estructura el resto del contenido de la memoria.

En el primer y actual capítulo se presenta el tema que se va a abordar y se introducen brevemente algunos conceptos básicos. Seguidamente, se presentan los objetivos que se pretenden alcanzar y, finalmente, se muestra un esquema de cómo se estructura la memoria que recoge todo el trabajo realizado.

El Capítulo 2 muestra el estado del arte, en el que se realiza una revisión exhaustiva de la literatura, analizando aplicaciones previas de estos sensores en la industria y las principales técnicas empleadas para el desarrollo de *soft sensors*.

En el Capítulo 3 se presenta el marco teórico necesario, se abordan conceptos clave de series temporales y se explican los modelos de Machine y Deep learning aplicados a este tipo de datos. Además, se presentan las estrategias de entrenamiento y predicción que se emplearán en el desarrollo.

En el Capítulo 4, por su parte, se establece la definición del problema, desde una descripción del proceso industrial en el que se enmarca el estudio, hasta la adquisición y análisis exploratorio de los datos que serán empleados.

Posteriormente, el Capítulo 5 muestra todo el modelado de la solución. En él se abarca el preprocesado de los datos y su preparación para el entrenamiento de los modelos, junto con la presentación de las herramientas que se usarán para desarrollar dichos modelos y las métricas y visualizaciones para su evaluación.

Tras esto, el Capítulo 6 recoge los resultados obtenidos. Para cada arquitectura se presentan los hiperparámetros ajustados, las métricas alcanzadas y las gráficas de evaluación. Posteriormente, se incluye una comparativa global que permite identificar los enfoques más adecuados en el problema planteado.

El Capítulo 7 presenta las conclusiones extraídas, analiza las limitaciones encontradas durante el desarrollo y presenta posibles líneas de investigación futura.

Por último, en el Capítulo 8 se aborda el impacto del proyecto, evaluando su alcance a nivel personal y académico, así como sus implicaciones en los ámbitos empresarial, económico, sociocultural y medioambiental. Además, se analiza su alineación con los Objetivos de Desarrollo Sostenible (ODS).



## Capítulo 2

# Estado del arte

**Soft sensor** es un término recientemente utilizado para referirse y describir sensores basados en *software*, que permiten estimar el valor de ciertas variables de un sistema o proceso a partir de la medición de otras [6]. Este tipo de sensores suele emplearse en situaciones en las que hay parámetros que son difíciles de medir directamente, como pueden ser variables de calidad de los procesos, fundamentales para el control de los mismos. Existen muchas razones por las cuales puede ocurrir esto, como alto coste de sensores físicos, imprecisiones en sus medidas, averías temporales o que, simplemente, no existan sensores físicos capaces de desarrollar una tarea concreta. Los *soft sensors* se presentan como una solución tecnológica que permite superar esas limitaciones.

Los modelos de *soft sensing* se pueden dividir en dos clases: basados en modelos (Model-driven) o basados en datos (Data-driven) [12].

Los **basados en modelos** suelen desarrollarse a partir de un conocimiento de los fundamentos físicos, químicos y/o biológicos de los procesos industriales. Se diseñan empleando ecuaciones derivadas de esos principios (relacionados con el proceso considerado), por lo que también son conocidos como modelos de primeros principios [12], [13]. Se consideran “de caja blanca” pues conocen los antecedentes y mecanismos internos del proceso. Sin embargo, la obtención de un dispositivo preciso como este suele resultar difícil debido a la complejidad y dinamicidad de los procesos [12].

En contraste con estos, los modelos **basados en datos** dependen en menor medida de un conocimiento exacto de la dinámica del procedimiento y ofrecen mayor flexibilidad [9]. En lugar de basarse en ecuaciones derivadas de principios físicos, este tipo de *soft sensors* utilizan grandes volúmenes de datos reales, históricos del proceso, para entrenar a diferentes modelos. Estos modelos son capaces de extraer patrones y relaciones entre las variables involucradas [12]. Así, a partir de técnicas como el Machine Learning (ML), Deep Learning (DL) u otras metodologías estadísticas, se logran realizar estimaciones precisas sin necesidad de conocer con profundidad las bases teóricas del proceso.

El desarrollo y aplicación de ambos tipos de *soft sensors* han adquirido gran relevancia en el ámbito industrial, donde la optimización y el control son esenciales.

## Capítulo 2. Estado del arte

---

Se han implementado en diversos sectores: químico, petroquímico, farmacéutico, energético, así como otros ámbitos similares.

En este estado del arte se analizan las principales técnicas de desarrollo, aplicaciones, ventajas y limitaciones de los *soft sensors* actuales.

### 2.1. Técnicas de modelado

Como se ha mencionado, existen diversos enfoques metodológicos para el desarrollo de *soft sensors*. Principalmente, estos se pueden construir empleando una serie de técnicas, entre las que destacan las siguientes: modelos basados en ML y DL, procesos estocásticos más clásicos u observadores de estado como los filtros de Kalman [10].

#### 2.1.1. Enfoques basados en ML y DL

En los últimos años, el principal avance se ha dado en el uso de técnicas de ML y, especialmente, DL en el desarrollo de sensores virtuales basados en datos. Estas han demostrado ser eficaces en la predicción de variables críticas en diferentes sectores, gracias a su capacidad para identificar patrones complejos y dinámicas no lineales.

#### Modelos sencillos

Los modelos empleados engloban desde los más sencillos hasta arquitecturas más avanzadas. Por ejemplo, la **regresión lineal**, uno de los más simples, ha sido una de las primeras aproximaciones en la construcción de estos sensores. Este modelo [14] establece una relación lineal entre una variable dependiente y una o más variables independientes, permitiendo realizar predicciones basadas en una ecuación ajustada a los datos observados. Su principal ventaja radica en su interpretabilidad y su bajo coste computacional. Sin embargo, su aplicación es muy restringida, sólo puede utilizarse en aquellos problemas en los que la relación entre las variables sea aproximadamente lineal.

Para solventar esta limitación, se introdujo el uso de árboles de decisión. Concretamente, se empezó a utilizar ensamblados de ellos con modelos como **Random forest**. Este modelo genera diversas particiones de los datos y combina sus resultados para obtener predicciones más robustas y generalizables, reduciendo la varianza y mitigando el sobreajuste de los datos. Stanislav Vakaruk [14] estudió y comparó el desempeño de estos dos modelos. Gracias a ello, se pudo comprobar que, efectivamente, si se utilizan datos más complejos, Random forest obtiene predicciones más acertadas que la regresión lineal.

#### Redes neuronales

A medida que las necesidades industriales han requerido modelos más avanzados, el uso de **redes neuronales artificiales (ANN)** ha cobrado gran relevancia.

Compuestas por capas de neuronas interconectadas que procesan la información ajustando sus pesos, permiten capturar patrones en los datos. Son múltiples los estudios que han desarrollado *soft sensors* empleando este tipo de modelos [6], [10], [11], [15]. En ellos se destaca su capacidad para representar relaciones no lineales. Sin embargo, cabe tener en cuenta que su entrenamiento requiere grandes volúmenes de datos y un ajuste de los hiperparámetros para evitar problemas como el sobreajuste.

Dentro de las redes neuronales, se han introducido arquitecturas más especializadas y eficientes para tareas específicas como estas. Entre ellas, las **redes neuronales convolucionales (CNN)** han demostrado ser especialmente útiles cuando los datos de entrada presentan estructuras espaciales relevantes, como sucede en aplicaciones relacionadas con imágenes o señales temporales. Emplean capas de convolución que permiten extraer características locales y patrones. Su capacidad para identificarlos con un menor número de parámetros en comparación con redes densamente conectadas ha favorecido su aplicación. J.R. Rodríguez et al. [6] evaluaron la aplicación de este tipo de redes unidimensionales (1DCNN) para la predicción del caudal de un proceso, obteniendo resultados prometedores. En particular, se observó que el sensor virtual basado en CNN fue capaz de replicar la tendencia de las mediciones reales, pero con alguna desviación.

En este mismo estudio, también se exploró el uso de **redes neuronales recurrentes (RNN)**, concretamente, la arquitectura **LSTM**. Estas redes han mostrado una gran eficacia en el desarrollo de *soft sensors* debido a su capacidad de capturar dependencias temporales, algo fundamental para modelar fenómenos que evolucionan en el tiempo. Las LSTM incorporan mecanismos de control que mejoran la retención de información en secuencias largas. C. Xie et al. [9] emplearon una LSTM bidireccional para imitar la estructura espacial del proceso sobre el que se aplicó. El artículo apoya la idea de que modelar sistemas complejos utilizando redes neuronales con una estructura mimética de la del proceso puede ser muy efectivo para modelos basados en datos.

J.R. Rodríguez et al. [6] realizaron una comparación de estas tres últimas arquitecturas para el desarrollo de un sensor de caudal: redes neuronales densas, redes convolucionales unidimensionales y redes LSTM. En su estudio, evaluaron el rendimiento de cada modelo en términos de error absoluto medio (MAE) y error cuadrático medio (RMSE). Llegaron a la conclusión de que la LSTM fue la que obtuvo el mejor desempeño con un MAE de 0,3844 y un RMSE de 0,2132. Pudo captar mejor la dinámica del sistema gracias a su capacidad para modelar dependencias temporales. No obstante, su mayor complejidad estructural implica un entrenamiento más costoso y una mayor necesidad de datos para evitar problemas de sobreajuste.

### 2.1.2. Enfoques estadísticos y clásicos

Una vez revisadas las principales técnicas de ML y DL, es importante señalar que en el ámbito de los sensores virtuales también se han empleado métodos más clásicos y estadísticos. Si bien su capacidad para capturar dinámicas no lineales

puede resultar más limitada, ofrecen ventajas en cuanto a interpretabilidad, menor coste computacional y robustez en entornos con datos limitados.

### Técnicas estadísticas

Una de las técnicas estadísticas más utilizadas en el desarrollo de los sensores es la **regresión por componentes principales (PCR)**. Surge como alternativa a la regresión lineal múltiple cuando las variables de entrada están fuertemente correlacionadas [15]. Utiliza análisis de componentes principales (PCA) para reducir la dimensionalidad del conjunto de datos, extrayendo variables latentes que capturan la mayor varianza. Luego, estas se emplean en un modelo de regresión para predecir la variable de interés. Se pueden aplicar versiones robustas que reducen el impacto de valores atípicos. Aunque en algunos casos, PCR puede requerir más componentes y, por tanto, introducir algo más de ruido, sigue siendo eficaz en contextos industriales complejos [13].

Otra técnica estadística también destacada es la **regresión por mínimos cuadrados parciales (PLS)**. A diferencia de PCR, PLS también reduce la dimensionalidad, pero optimiza la predicción al considerar tanto la varianza de las variables de entrada como su correlación con la salida. Esto permite obtener buenos resultados con menos componentes que la anterior. Además, puede extenderse a modelos dinámicos (DPLS) incorporando retardos temporales, lo que mejora la capacidad de seguimiento en sistemas con dinámica temporal [13]. PLS ha demostrado ser eficaz en aplicaciones como sensores blandos para emisiones, especialmente cuando se combinan con técnicas de detección de valores atípicos.

Tanto PCR como PLS han sido aplicadas con éxito en numerosos estudios, especialmente en contextos industriales donde los datos presentan alta colinealidad. Bao Lin et al. [13] propusieron un enfoque sistemático para el diseño de *soft sensors* en el que se comparan ambas técnicas. En particular, se aplican los modelos para predecir variables de calidad no directamente medibles, como la concentración de productos intermedios. El estudio destaca que, si bien ambos métodos ofrecen buenos resultados, PLS tiende a superar a PCR en términos de precisión con un menor número de variables latentes, gracias a su capacidad para considerar la correlación con la variable objetivo durante la construcción del modelo.

### Observadores y modelos clásicos

Además de estas técnicas de regresión, otra familia de métodos clásicos empleados son los observadores de estado [8]. Entre ellos destacan el **observador de Luenberger y el filtro de Kalman**. El primero se utiliza en sistemas lineales para estimar variables internas a partir de entradas y salidas observadas, corrigiendo su estimación mediante la diferencia entre la salida real y la estimada. Por su parte, el filtro de Kalman introduce una ventaja clave al considerar de forma explícita el ruido del proceso y de las mediciones, lo que mejora la precisión de las estimaciones en entornos con incertidumbre. Su versión extendida permite, además, su aplicación a sistemas no lineales.

Por último, cabe mencionar el modelo **ARIMA** (Autoregressive Integrated Moving Average), otra técnica clásica utilizada para la predicción de series temporales univariantes. Combina componentes autorregresivos, de media móvil y de diferenciación, lo que le permite capturar patrones como tendencias o estacionalidades [16]. Su principal ventaja es su simplicidad y bajo coste computacional, siendo útil en procesos estables donde se dispone de un historial suficiente. Sin embargo, presenta limitaciones en entornos multivariantes o con dinámicas altamente no lineales, lo que restringe su aplicabilidad directa en procesos industriales más complejos [17].

En resumen, el desarrollo de modelos de *soft sensors* en el ámbito industrial se apoya en una amplia gama de técnicas, cuya elección depende del tipo de proceso, la disponibilidad de datos y los requisitos del sistema. Aunque estas metodologías han permitido grandes avances en la estimación de variables difíciles de medir, cada una presenta también ciertas limitaciones, como la necesidad de grandes volúmenes de datos, la complejidad computacional o la dificultad de interpretación.

En el caso particular abordado en este trabajo, una columna de destilación de la compañía Repsol, se dispone de un conjunto de datos operativos reales. Este tipo de proceso genera datos multivariantes y secuenciales, con dinámicas temporales complejas, de las cuales no se conocen las ecuaciones o fundamentos que lo rigen. Por este motivo, el desarrollo de un *soft sensor* de primeros principios resulta poco viable. En cambio, la disponibilidad de dichos datos históricos hace más accesible la construcción de un modelo basado en datos. Dentro de este enfoque, el uso de técnicas de DL es una opción especialmente adecuada para capturar las relaciones no lineales y modelar las dependencias temporales.

## 2.2. Aplicaciones en la industria

Los sensores virtuales están presentes en distintos ámbitos de la automatización. Concretamente, en el ámbito industrial existen numerosas implementaciones, incluyendo refinerías de petróleo, plantas químicas, hornos de cemento, centrales eléctricas, industrias de papel y plantas nucleares, entre otras [18].

### Sector cementero

En la industria del cemento, se han convertido en herramientas muy útiles para optimizar el control de los hornos de clinkerización. En ellos, variables críticas, como la temperatura del horno o la composición química del material, son difíciles de medir en tiempo real. Bao Lin et al. [13] propusieron un enfoque basado en técnicas estadísticas para desarrollar modelos inferenciales. Se incluyeron dos casos de estudio: la estimación de la concentración de cal libre en el producto y la predicción de emisiones de óxidos de nitrógeno (NO<sub>x</sub>).

La evaluación del rendimiento de los modelos se hizo utilizando la suma de errores cuadráticos de predicción (PRESS). Mide el error acumulado en las pre-

## Capítulo 2. Estado del arte

---

dicciones hechas; cuanto menor sea este, mejor será el ajuste. Para la predicción de la cal libre, el modelo obtuvo un PRESS de 38, siendo este un buen resultado para los valores de la variable; el error es relativamente bajo y, por tanto, una estimación precisa. Por otra parte, para la predicción de las emisiones, el PRESS fue  $8,39 \times 10^7$ . Este valor es mucho mayor, pero eso se debe a la magnitud de esta variable. La métrica de evaluación debe interpretarse considerando los datos implicados y las escalas numéricas de estos.

### Sector energético

En cuanto al sector energético, están desempeñando un papel fundamental. Por ejemplo, en plantas termoeléctricas se han desarrollado diversos sensores virtuales para estimar parámetros como la eficiencia térmica y la composición de gases de combustión, así como para diagnosticar fallos en sensores físicos y optimizar el control del sistema.

Un ejemplo destacado de esto último es su aplicación en el sistema de diagnóstico de fallos y sensores virtuales (SDFSV) descrito en [19], diseñado para mejorar la confiabilidad operativa. Además de contribuir a la detección de fallos (en sensores físicos), este sistema permite estimar variables críticas como la presión en distintas etapas de la turbina y la potencia eléctrica generada. Su efectividad para mantener el desempeño del sistema de control incluso ante fallos se demostró en un simulador de una central termoeléctrica real.

En el ámbito de las energías renovables, han sido aplicados en parques eólicos y plantas solares fotovoltaicas. En el caso de la energía solar, se han aplicado para estimar parámetros clave como la potencia de salida, la temperatura del panel y la irradiación solar (sin necesidad de sensores físicos adicionales) [18]. Su integración con diversas técnicas permite aumentar la frecuencia de muestreo y optimizar algoritmos de Maximum Power Point Tracking (MPPT), ajustando en tiempo real la orientación de los paneles para maximizar la generación de energía [11].

Para la producción de biogás, han sido empleados en la optimización del proceso de digestión anaerobia, facilitando el monitoreo de parámetros clave como el pH. Un estudio [14] desarrolló un sensor basado en redes neuronales LSTM para predecir este valor en un biodigestor. Alcanzó una alta precisión con un error cuadrático medio (MSE) de 0,0016 y un error porcentual absoluto medio (MAPE) de 0,4446%. Los valores de estas métricas indican una predicción precisa, con desviaciones inferiores al 0,5% respecto al valor real, lo que permite controlar el proceso de digestión con fiabilidad comparable a la de mediciones directas.

### Sector químico y petroquímico

En los sectores químico y petroquímico se han implementado en diferentes tipos de procesos. J. C. B. Gonzaga et al. [10] estudiaron su uso para monitorear propiedades como la viscosidad en procesos de polimeración. Concretamente se aplicó para la producción de polietileno tereftalato, en el que la viscosidad es un parámetro fundamental de control. Usando redes neuronales se consiguió inferir

dicha variable en tiempo real, obteniendo un error relativo de aproximadamente el 0,3%. Así, la diferencia entre el valor predicho y el real representa solo una fracción mínima del valor verdadero. Además, se calculó y reportó también el error absoluto máximo observado, que fue de 4 poise (unidad de viscosidad dinámica). Este valor supone un margen de error reducido ya que en estos procesos industriales las viscosidades habituales suelen encontrarse entre 100 y 200 poise. La precisión demuestra que el modelo es capaz de seguir las variaciones de viscosidad.

Al-Jamimi et al [20] revisaron soluciones basadas en diferentes técnicas de aprendizaje automático (ANN, SVM) para la desulfuración, eliminación de azufre, de productos derivados del petróleo. Los resultados fueron evaluados utilizando métricas estadísticas de ajuste y precisión, incluyendo el coeficiente de determinación ( $R^2$ ), el error cuadrático medio (MSE) y el error relativo medio (MRE). El análisis de estos indicadores mostró que ambas técnicas tienen una capacidad predictiva satisfactoria, con una fuerte concordancia entre las predicciones y los valores reales. Los errores relativos se mantuvieron dentro de rangos aceptables para aplicaciones industriales, lo que permitió identificar parámetros que impactan de manera significativa en la eficiencia del proceso de desulfuración.

En los reactores químicos, estos sensores se han empleado para el control de parámetros vinculados a la conversión de reactivos o la concentración de productos intermedios. En concreto, se han aplicado en redes de reactores donde el acceso a mediciones en tiempo real es más limitado. C. Xie et al. [9] evaluaron la efectividad de diferentes modelos de redes neuronales para la predicción de variables en este tipo de sistemas. Consideraron dos casos de estudio, el primero consistió en una simulación y el segundo en una aplicación a un caso industrial real.

Simularon una red de reactores constituida por una cascada de tres reactores de tanque agitado continuo con enfriamiento a contracorriente. El objetivo del desarrollo del sensor era predecir las variables de concentración en los tres reactores. Para evaluar los modelos se usaron dos métricas, el RMSE (raíz de error cuadrático medio) y el  $R^2$ . En este caso se logró obtener un RMSE menor de 0,003 y un  $R^2$  mayor de 0,99. Un RMSE tan bajo indica que los errores de predicción en comparación con los valores reales son prácticamente despreciables. Por otro lado, un  $R^2$  cercano a 1 sugiere que el modelo es capaz de explicar casi toda la variabilidad de los datos. En conjunto, estas métricas confirman que el sensor virtual desarrollado es confiable y adecuado.

En el segundo caso, se aplicó a un proceso industrial real que consistía en siete reactores conectados sucesivamente para producir dinitrobenceno. Se buscó desarrollar un *soft sensor* para predecir la concentración de salida del componente en el cuarto reactor, considerada como la variable de calidad crítica para la seguridad del proceso. Los mejores resultados mostraron que un modelo basado en atención alcanzó un RMSE de 0,07 y un  $R^2$  de 0,71.

El estudio también resaltó, a partir de los dos casos de estudio, la importancia de la calidad y complejidad de los datos en el desempeño de los modelos. Los resultados en la simulación fueron significativamente mejores que en la aplicación

real. El proceso era más controlado y predecible, la no linealidad era consistente, el sistema operaba en condiciones normales sin anomalías y el ruido añadido seguía una distribución gaussiana, lo que facilitaba el aprendizaje del modelo. En contraste, en un proceso industrial real se presentan mayores desafíos, lo que dificulta en cierta medida el aprendizaje.

Por otra parte, también se han aplicado en columnas de destilación. A. K. Pani et al. [15] desarrollaron un *soft sensor* en una columna de desbutanizador con el objetivo de conseguir una monitorización en línea del contenido de butano en el fondo de la columna. Empleando y comparando diferentes modelos de aprendizaje profundo, se logró estimar una precisión alta. En el mejor caso se obtuvo un MAE de 0,055, un RMSE de 0,076 y un coeficiente de determinación ( $R^2$ ) de 0,856, lo que refleja un rendimiento adecuado del modelo. Los errores cometidos en la predicción del contenido son bajos en magnitud (MAE y RMSE), y el  $R^2$  evidencia una fuerte relación entre las predicciones del modelo y los valores reales.

Hay otros ejemplos aplicados a distintos tipos de columnas de destilación, como una para la separación de agua y etanol, en la que se busca predecir la concentración de etanol en el tope [21].

En definitiva, la versatilidad y capacidad adaptativa de los *soft sensors* ha permitido su consolidación como herramientas clave en numerosos sectores industriales, contribuyendo a una mayor eficiencia operativa y toma de decisiones basada en datos.

Estos sensores ofrecen numerosas ventajas que han impulsado su uso, como se ha podido comprobar. Su bajo coste, flexibilidad y facilidad de integración los convierten en una opción eficiente frente a sensores físicos, especialmente en entornos hostiles o donde las mediciones directas no son viables [18]. Permiten estimaciones en tiempo real de variables no medibles directamente, mejoran la continuidad operativa y optimizan el rendimiento de los procesos al integrarse en sistemas de control automático [22]. Su implementación contribuye a la sostenibilidad al reducir la cantidad de *hardware* necesario. Ofrecen funciones complementarias [11] como diagnóstico de fallos, compensación del desgaste de sensores físicos y predicción multi-paso.

No obstante, el desarrollo y mantenimiento de sensores virtuales aún enfrenta desafíos significativos [18], [22]. Su desempeño depende en gran medida de la calidad y cantidad de los datos empleados para su entrenamiento y validación, siendo sensibles a valores atípicos, ruido o cambios en las condiciones del proceso. Requieren un diseño cuidadoso, conocimientos especializados en modelado o inteligencia artificial, y ajustes regulares para evitar la degradación del modelo. La falta de estandarización y la dificultad para interpretar algunos modelos complejos también son obstáculos para su adopción generalizada. A pesar de ello, su potencial sigue creciendo, impulsado por los avances en aprendizaje automático y procesamiento de datos.

## Capítulo 3

# Fundamento teórico

Para abordar adecuadamente el desarrollo de modelos de Deep Learning aplicados a series temporales industriales para la implementación de *soft sensors*, es fundamental primero establecer los fundamentos teóricos que lo sustentan. Es necesario entender tanto la naturaleza de las series temporales en entornos industriales como los modelos de aprendizaje profundo aplicables a este contexto.

En los procesos industriales, muchas de las variables de interés evolucionan de forma continua en el tiempo. Esta característica hace que su análisis requiera técnicas específicas capaces de capturar la dinámica temporal y las posibles relaciones entre diferentes señales del sistema. Las series temporales constituyen, por tanto, una herramienta clave para modelar y predecir el comportamiento de estas variables [23].

Este capítulo presenta una descripción general del concepto de series temporales, su estructura y características principales, y los retos asociados a su modelado en contextos industriales. A continuación, se abordan los principales modelos que serán utilizados en el desarrollo del *soft sensor*, junto con las estrategias empleadas para su entrenamiento y validación.

### 3.1. Series temporales

Las **series temporales** son secuencias ordenadas cronológicamente de valores, donde cada punto de datos representa una observación de una variable en un momento concreto. Las observaciones suelen registrarse a intervalos regulares de tiempo: días, horas, minutos . . . Formalmente, una serie se representa matemáticamente mediante una sucesión de valores  $Y = \{y_1, y_2, \dots, y_t\}$ , donde  $y_t$  es la observación de la variable en el instante  $t$  [23].

Están presentes en numerosos campos y ámbitos, la economía y el marketing, la demografía, el medioambiente o la industria. Por ello, se han convertido en una herramienta para estudiar el desarrollo y la evolución de una variable a lo largo del tiempo [24]. La clave del análisis de estas reside en su naturaleza secuencial, donde el orden es importante y observaciones pasadas influyen en las futuras [23].

### Capítulo 3. Fundamento teórico

---

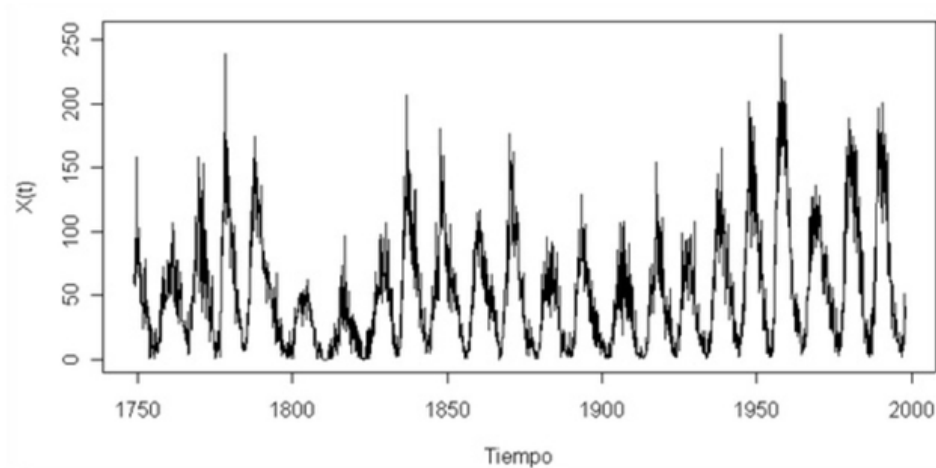


Figura 3.1: Ejemplo de representación de una serie temporal [24]

Las series temporales se pueden clasificar según el número de variables que se observan y evolucionan en el tiempo [24]. Las series univariantes son aquellas en las que únicamente se estudia una variable. Al contrario, en las multivariantes se analiza más de una variable, recogiendo el valor de cada una de estas. Normalmente, el interés reside en la dependencia o influencia que hay entre ellas.

Desde una perspectiva teórica, para analizar y comprender el comportamiento de las series es necesario tener en cuenta que estas se pueden entender como la combinación de cuatro elementos fundamentales [23], [25]:

- **Tendencia:** patrón general que adopta una serie temporal a lo largo de un periodo prolongado. Puede ser un crecimiento, decrecimiento o estabilización, representándose como una trayectoria suave, no necesariamente lineal.
- **Ciclo:** fluctuaciones que se repiten en el tiempo pero que no siguen periodicidad fija, su duración varía según el fenómeno que se analice. Generalmente la componente cíclica suele agruparse con la tendencia.
- **Estacionalidad:** repeticiones sistemáticas en la serie con una frecuencia regular y conocida, pudiendo ser diaria, semanal o anual, entre otras.
- **Componente irregular o residual:** variaciones impredecibles que no pueden ser explicadas por la tendencia, estacionalidad o ciclos. Recoge el ruido o aleatoriedad inherente del sistema.

La Figura 3.2 muestra un ejemplo de esta descomposición aplicada a una serie temporal, en cada uno de sus componentes principales.

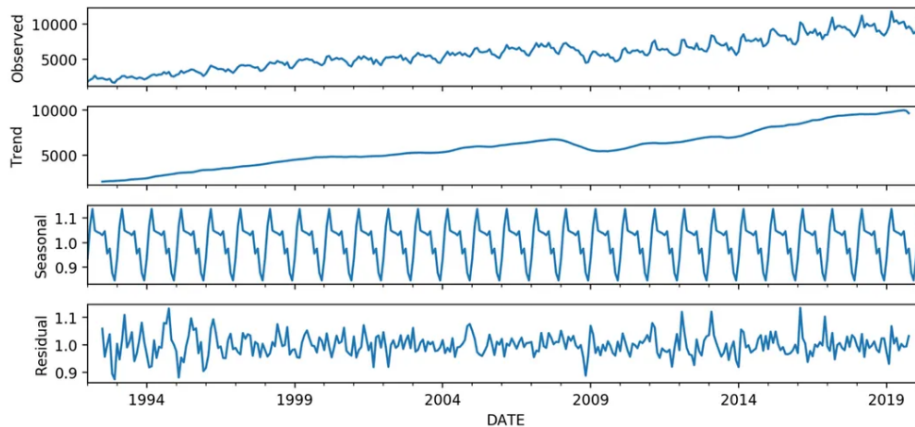


Figura 3.2: Descomposición de una serie temporal [25]

En contextos industriales, estos patrones rara vez se presentan de forma aislada o estable, suelen superponerse y variar dinámicamente lo que supone una dificultad en el modelado. Además, son entornos complejos donde influyen factores humanos, técnicos y ambientales, que afectan a la evolución de las variables. Esta situación conlleva diversos desafíos asociados al análisis y tratamiento de las series temporales en este ámbito particular [26]. Entre los más frecuentes destacan:

- **No estacionalidad o estacionalidades múltiples e irregulares.** Muchas variables industriales suelen mostrar tendencias, estacionalidad u otros comportamientos no constantes, limitando la aplicabilidad de técnicas clásicas. Asimismo, las series industriales suelen presentar estacionalidades múltiples o irregulares, que derivan de ciclos operativos, mantenimiento de maquinaria, turnos de trabajo o condiciones ambientales, introduciendo mayor complejidad.
- **Falta de datos o registros incompletos.** Es frecuente encontrarse con falta de datos, que se puede dar por fallos en los sensores, interrupciones operativas o errores en la recopilación de estos. Esta situación, requiere un análisis detallado, ya que la forma en que se traten los datos puede afectar la precisión de los modelos de predicción.
- **Valores atípicos o anomalías.** Se trata de errores de medición, fallos puntuales o situaciones de transición que pueden distorsionar los patrones y estructura temporal.
- **Escasez de datos históricos de calidad.** En ciertos casos, no se dispone de historiales suficientes o estos contienen ruido e inconsistencias. Limita la capacidad de los modelos para generalizar.

Teniendo en cuenta lo anterior, el tratamiento de series temporales en entornos industriales requiere métodos más allá de los tradicionales. Es imprescindible aplicar un preprocesamiento meticuloso y seleccionar modelos que sean capaces de capturar las relaciones no lineales y complejas, además de adaptarse a entornos dinámicos y gestionar la incertidumbre inherente en ellos.

### 3.2. Modelos estudiados

Para el desarrollo de un *soft sensor* aplicado al proceso industrial en el que se centra el presente trabajo, una columna de destilación de C3-C4, se desarrollarán diversos modelos de ML y DL. Dada la naturaleza compleja de este proceso y la dificultad que supone simularlo mediante métodos tradicionales, un enfoque basado en datos que utilice este tipo de técnicas avanzadas es la solución más adecuada. A continuación, se explica brevemente el fundamento que sustenta a cada una de las arquitecturas que se evaluarán en este estudio.

La selección de estos modelos no es arbitraria, responde a una estrategia fundamentada en la literatura reciente [27], donde se enfatiza en la importancia de realizar evaluaciones comparativas. Con este objetivo, se ha tratado de integrar modelos de diversa naturaleza, enfoques clásicos, más sencillos, y métodos más avanzados, pero con una base conceptual ya consolidada.

Así, se incluyen modelos **baseline** y técnicas clásicas de ML basadas en **árboles de decisión**, que permiten establecer puntos de comparación básicos. También se contempla el uso de redes recurrentes, **LSTM**, por su gran capacidad para modelar secuencias temporales y capturar las dependencias a largo plazo; y modelos convolucionales, en concreto **TCN**, que permite detectar patrones locales en las series temporales industriales.

Se incorporan asimismo modelos basados en perceptrones multicapa, con un enfoque más reciente que busca simplificar la complejidad de otras arquitecturas profundas. Finalmente, como representación de modelos basados en atención se incluye el **Temporal Fusion Transformer (TFT)**, dado que presenta gran capacidad para manejar dependencias temporales complejas.

Cabe destacar que la elección realizada no trata arquitecturas más experimentales y de reciente aparición, sino que se ha priorizado el uso de modelos que cuentan con una trayectoria sólida en la literatura y validados en diferentes estudios. Se busca construir un análisis basado en arquitecturas maduras y robustas, con las que ofrecer un marco comparativo completo que permita identificar cuál de las estrategias resulta más adecuada.

#### 3.2.1. Modelos *baseline*

Como punto de referencia, antes de comenzar a implementar técnicas más avanzadas y complejas, es fundamental desarrollar modelos *baseline*, caracterizados por su sencillez. Este tipo de modelos *naive* utilizan aproximaciones muy básicas para realizar predicciones [28]. A partir de ellos, se podrá evaluar el rendimiento de arquitecturas más elaboradas. En este trabajo se consideran tres modelos simples [29]:

- **NaiveCopy:** replica el último valor medido durante el entrenamiento como estimación.
- **NaiveMean:** utiliza el valor promedio histórico de los datos de entrenamiento como predicción.

- **NaiveDrift:** ajusta una línea recta entre la primera y última instancia del conjunto de entrenamiento y la prolonga en el tiempo para realizar las predicciones. Si tenemos una serie de entrenamiento de longitud  $T$  se tiene:

$$\hat{y}_{T+h} = y_T + h \left( \frac{y_T - y_1}{T - 1} \right) \quad (3.1)$$

### 3.2.2. Clásicos de Machine Learning

Tras establecer los modelos *naive* como base, este estudio implementa modelos más sofisticados que pueden capturar relaciones más complejas en los datos de una columna de destilación. Se desarrollarán modelos clásicos de ML basados en árboles de decisión. Concretamente, se estudiará la aplicación de técnicas de Random forest y XGBoost en los datos disponibles.

#### Random forest (RF)

El Bosque aleatorio o Random forest es un algoritmo de aprendizaje automático supervisado basado en el *ensamble*, es decir, en la combinación de un conjunto de modelos. En particular, se trata de una agrupación de árboles de decisión, en la que se combina la salida de estos para generar un estimador con el que obtener una predicción [30].

Es un algoritmo que utiliza el *bagging* como método de ensamblado de dichos modelos. Se seleccionan múltiples muestras aleatorias con reemplazo en el conjunto de datos de entrenamiento. Para cada una de estas muestras de datos se crea un árbol de decisión y se entrenan de manera independiente en paralelo. Posteriormente, se predice una observación en cada árbol, empleando el promedio de esas predicciones para obtener la estimación (Figura 3.3). De esta manera, se consigue reducir la varianza en un conjunto ruidoso de datos.

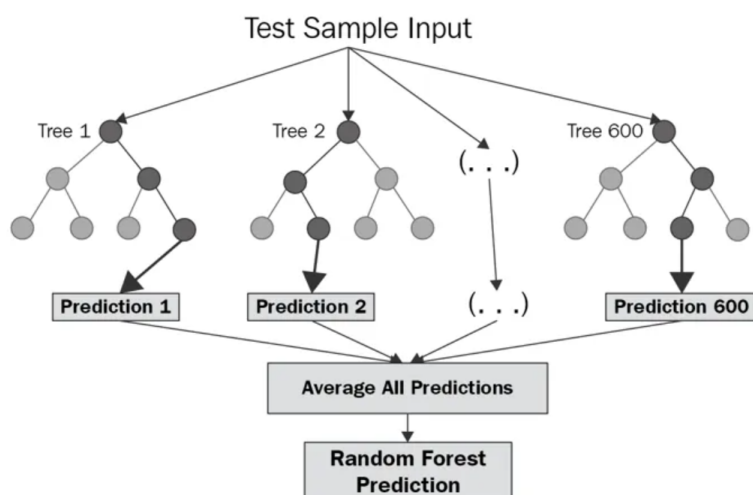


Figura 3.3: Arquitectura de Random forest [31]

Existen una serie de factores que afectan a la precisión del modelo: el número

### Capítulo 3. Fundamento teórico

de estimadores, es decir, el número de árboles que se crean, o su profundidad, el número máximo de nodos de cada uno de los árboles [30]. Un mayor número de estimadores suele mejorar la estabilidad del modelo, reduciendo la varianza al contar con mayor número de predicciones. De igual forma, una mayor profundidad en los árboles permite capturar patrones más complejos. Sin embargo, se ha de tener cuidado ya que aumenta el riesgo de sobreajuste si el modelo se adapta demasiado a los datos de entrenamiento.

#### XGBoost

XGBoost es un algoritmo de aprendizaje automático basado también en un *ensemble* de árboles de decisión. A diferencia del Random forest, este modelo utiliza el **boosting** como método de ensamblado. Aun así, comparten el mismo enfoque, unificar varios estimadores débiles para obtener un estimador mejor. Se emplea principalmente para reducir el sesgo y la varianza [30].

Mientras que en el caso anterior los estimadores se entrenaban en paralelo y simultáneamente, en este algoritmo el entrenamiento es secuencial.

Se parte de un conjunto de datos del cual se extrae una muestra para entrenar el primer árbol del modelo. Una vez entrenado, este árbol se utiliza para predecir todo el conjunto de datos original, y se identifica qué observaciones han sido clasificadas correctamente y cuáles no. A continuación, las muestras mal clasificadas se emplean para entrenar un segundo árbol. Este nuevo modelo vuelve a predecir el conjunto completo y, nuevamente, se seleccionan las observaciones incorrectas para entrenar un tercer árbol. Este procedimiento se repite de forma iterativa hasta alcanzar el número deseado de estimadores. Finalmente, todos los árboles generados se combinan para formar un modelo más robusto y preciso. Este proceso se ilustra en la siguiente imagen:

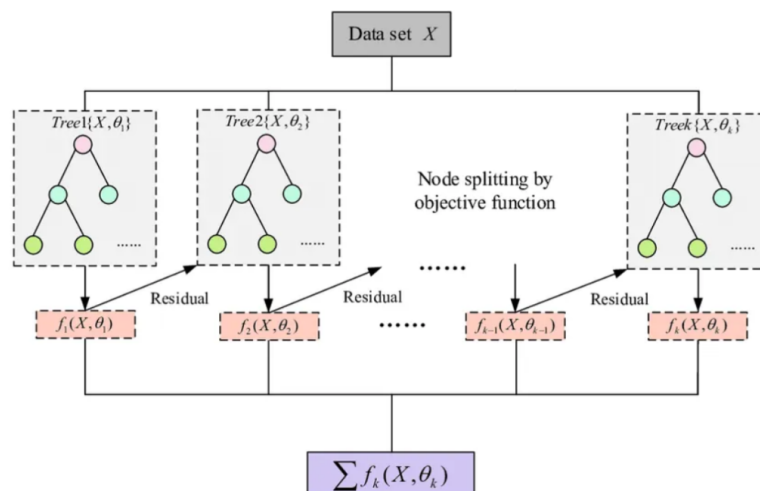


Figura 3.4: Arquitectura de XGBoost [32]

### 3.2.3. Arquitecturas de Deep Learning

Además de los modelos ya expuestos, para hacer frente a la complejidad inherente de un proceso industrial como el que se está enfocando, es necesario implementar arquitecturas más avanzadas. Estas son capaces de modelar en los datos relaciones temporales complejas y patrones no lineales. Las redes neuronales, especialmente aquellas diseñadas para series temporales, ofrecen esas capacidades para capturar estas dinámicas [33]. En el presente trabajo se explorará el uso de diferentes arquitecturas de este tipo que han demostrado buenos resultados en procesos similares.

Estas técnicas son en particular muy útiles porque permiten modelar interacciones entre variables, así como incorporar memoria a largo plazo, basándose en el tratamiento de series temporales [20], [33]. Se adaptan automáticamente a la estructura subyacente de los datos, lo que resulta valioso en un proceso tan complejo, donde intervienen múltiples variables con relaciones no triviales entre ellas [20].

A continuación, se muestran los modelos que se probarán.

#### Long Short-Term Memory (LSTM)

Una red de memoria a corto-largo plazo (LSTM) es un tipo de red neuronal recurrente.

Las redes neuronales recurrentes (RNN) son una clase de aprendizaje profundo, basada en los trabajos de David Rumelhart en 1986 [34]. Son un tipo de red neuronal diseñada específicamente para procesar y obtener información de datos que siguen un orden secuencial. De esta forma, estas arquitecturas son usadas principalmente en el procesamiento del lenguaje natural, análisis de videos, música y, como en este caso, de series temporales [35].

En las redes neuronales artificiales, la información fluye en una dirección desde la entrada hasta la salida de la red, pasando por las capas ocultas, sin considerar una estructura secuencial de los datos. Sin embargo, las RNN mantienen estados internos que se actualizan, transmiten y comparten, lo cual les permite recordar información pasada y con ello predecir valores futuros en la secuencia [35], [36].

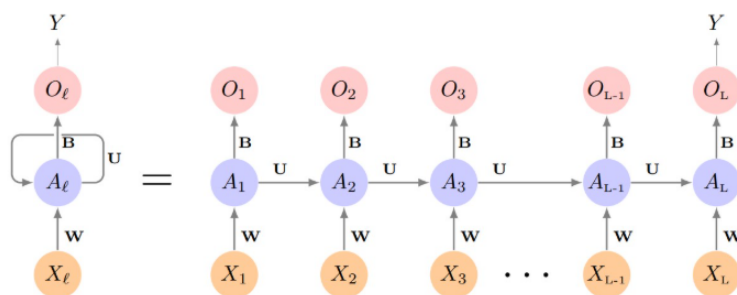


Figura 3.5: Diagrama de una red neuronal recurrente sencilla [35]

## Capítulo 3. Fundamento teórico

Como se puede observar en la Figura 3.5, cada neurona, o célula, toma dos entradas: la entrada actual y el estado oculto previo. Este último es ese estado interno, que puede entenderse como una memoria que permite a la red retener información pasada. Ambas entradas se combinan para calcular la salida y el nuevo estado oculto.

Estas redes presentan ciertas limitaciones para capturar patrones a más largo plazo. Para dar solución a este problema, surge este tipo de RNN especializado, la LSTM. Presentan una arquitectura similar, aunque más compleja, con unidades de memoria y mecanismos de puertas que regulan el flujo de información. Concretamente, cada celda tiene una estructura modular con tres puertas principales [35]:

- **Puerta de olvido** (*Forget gate*): regula la información que se debe olvidar y la que se mantiene, combinando la entrada actual y la salida anterior con una función sigmoide.
- **Puerta de entrada** (*Input gate*): decide cuánta información debe añadirse a la memoria a largo plazo.
- **Puerta de salida** (*Output gate*): determina cuánta información de la memoria actual se utilizará para la salida final.

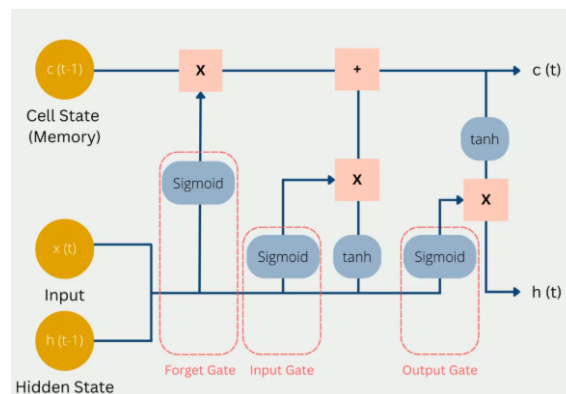


Figura 3.6: Arquitectura de una celda LSTM [35]

### Redes Convolucionales Temporales (TCN)

Junto con las redes neuronales recurrentes, otra de las principales arquitecturas que permite el modelado de secuencias es la red convolucional temporal (TCN) [37]. Este tipo de modelo hace referencia a la familia de arquitecturas que constan de capas convolucionales de una dimensión, causales, dilatadas y con la misma longitud de entrada y salida [38]. A continuación, se detalla qué significa cada uno de estos aspectos.

- **Red convolucional de una sola dimensión:** procesa secuencias mediante el uso de filtros que se deslizan a lo largo del eje temporal. Cada capa toma una secuencia como entrada, generando otra de salida de igual longitud (salvo el número de canales, que depende del número de filtros). Para obte-

ner cada elemento de salida se realiza un producto escalar entre la ventana de entrada y el filtro [37]. Se muestra en la siguiente imagen.

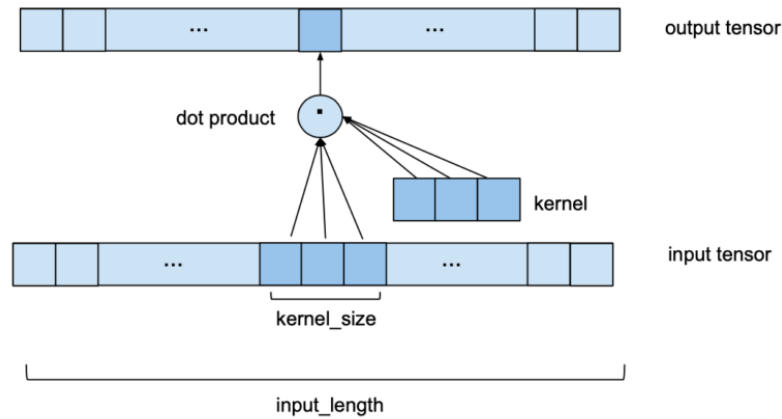


Figura 3.7: Capa convolucional de una dimensión [38]

- **Capas convolucionales causales:** para poder aplicar este tipo de arquitecturas a series temporales, donde el momento en el que se considera cada instante y dato tiene gran relevancia, es necesario que las capas convolucionales sean causales [38]. Se trata de capas como las anteriores, pero con la limitación de que cada elemento en la secuencia de salida solo puede obtenerse de los elementos que vienen antes de él en la de entrada. Para asegurar que el tensor de salida tiene el mismo tamaño que el tensor de entrada, es necesario añadir ceros al inicio de este, cuya cantidad dependerá del tamaño del filtro.
- **Dilatación:** es un mecanismo que permite aumentar el campo receptivo, y que los elementos de salida dependan de más valores de entrada, sin aumentar el coste computacional [37]. En lugar de aplicar los filtros de forma contigua, se dejan espacios entre los elementos de entrada considerados en cada cálculo. Sin embargo, para evitar que haya valores que no se tengan en cuenta, es necesario mantener una relación entre el tamaño del filtro y la base de dilatación [38].

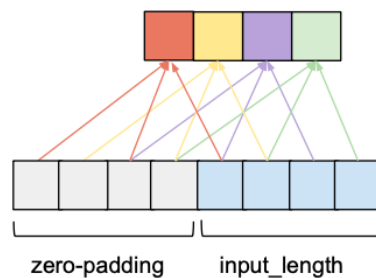


Figura 3.8: Capa convolucional causal y dilatada [38]

## Capítulo 3. Fundamento teórico

La red TCN básica tendría un aspecto similar al de la siguiente imagen:

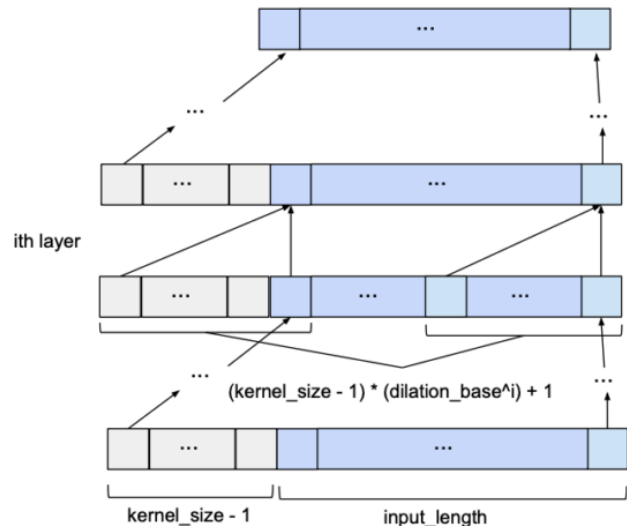


Figura 3.9: Red TCN básica [38]

### DLinear y NLinear

En el presente trabajo, además de las arquitecturas ya expuestas, también se van a desarrollar dos modelos lineales propuestos recientemente para la predicción de series temporales a largo plazo, DLinear y NLinear. Pertenecen al conjunto de modelos LTSF-Linear (*Long-term series forecasting*), basados en arquitecturas extremadamente simples. Según estudios recientes, han conseguido superar el rendimiento obtenido con modelos basados en Transformers (véase el siguiente modelo) en diferentes conjuntos de datos reales [39]. Se caracterizan por aplicar una única capa lineal sobre la secuencia de entrada histórica para predecir la secuencia futura. Son especialmente útiles en situaciones en las que la preservación de la información temporal es importante.

**DLinear** [39], [40] combina un esquema de descomposición de tendencia estacional con capas lineales. Primero, realiza la descomposición explícita de la serie temporal en dos componentes: una componente de tendencia (mediante un núcleo de media móvil) y otra componente residual o estacional (diferencia entre la serie original y la tendencia).

A cada una de las dos componentes anteriores, se les aplica una capa lineal independiente, sumando sus resultados para obtener una predicción, como se puede ver en la Figura 3.10. Este tipo de red permite capturar tendencias en los datos, mejorando la predicción en aquellos con comportamientos direccionales.

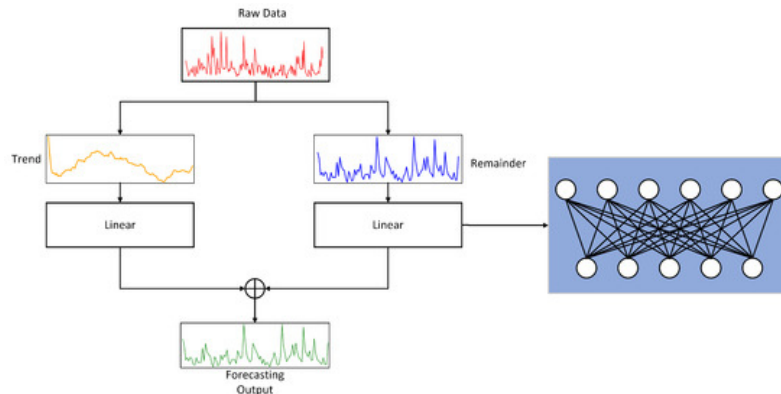


Figura 3.10: Funcionamiento del modelo DLinear [40]

**NLinear** [39], [40] por su parte, realiza una normalización sencilla a la secuencia de entrada. Extrae el último valor de esta antes de aplicar una capa lineal, para posteriormente, una vez se haya aplicado, sumarlo de nuevo. Con esta técnica se ayuda a estabilizar el modelo cuando hay un cambio de distribución en el conjunto de datos.

Ambos modelos presentan alternativas precisas y eficientes frente a arquitecturas más complejas, con menores costes computacionales y sin perder capacidad de modelar las series temporales [39].

#### Temporal Fusion Transformer (TFT)

Otro avance reciente en el campo del aprendizaje profundo ha sido el modelo Temporal Fusion Transformer (TFT). Desarrollado por investigadores de la Universidad de Oxford y Google [41], el TFT ha sido específicamente diseñado para la predicción de series temporales multihorizonte y multivariantes. Integra las características y flexibilidad de las RNN con la capacidad de modelado de atención de los Transformers [42].

Los Transformers son una arquitectura de red neuronal que destaca por el uso de mecanismos de atención para aprender las relaciones entre las distintas posiciones en una secuencia de datos. Su estructura básica consta de bloques de atención, capas de normalización y capas de alimentación hacia adelante. Cada bloque de atención calcula una matriz que pondera la importancia de cada elemento de la secuencia con respecto a los demás. Las otras dos capas ayudan a estabilizar y transformar la salida de cada uno de estos bloques, consiguiendo capturar las dependencias temporales [41].

Los modelos anteriores al TFT manejaban los datos de forma menos flexible, pues solo se centraban en datos históricos conocidos o en futuros. Sin embargo, este modelo incluye múltiples tipos de variables, desde conocidas hasta desconocidas a lo largo de distintos horizontes temporales (Figura 3.11), lo que permite mejorar la precisión e interpretabilidad [41], [42].

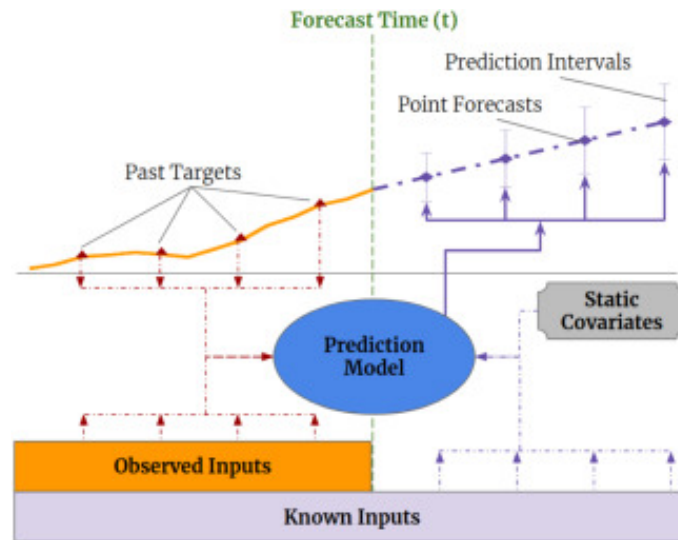


Figura 3.11: Predicción con múltiples horizontes [41]

A continuación, se muestra una descripción de la arquitectura del modelo, incluyendo únicamente una visión general de sus componentes principales debido a su complejidad [41], [42]. Todo el conjunto puede observarse en la Figura 3.12.

- **Mecanismos de control** (*Gating mechanisms*): permiten al modelo omitir ciertos componentes de la arquitectura cuando no son necesarios. Se implementan mediante Redes Residuales con Puerta (GRN), como un bloque de la arquitectura, permitiendo al modelo aprender cuando algunas capas de procesamiento no lineales son innecesarias.
- **Redes de selección de variables** (*Variable selection networks*): a la hora de entrenar un modelo, se dispone de múltiples variables de entrada. Sin embargo, no todas ellas tienen porque ser relevantes y la contribución al valor de salida es desconocida. TFT contiene este componente para seleccionar las variables más relevantes en cada paso de tiempo. Permite eliminar entradas ruidosas que pudieran afectar negativamente el rendimiento y ayuda a mejorar la interpretabilidad del modelo.
- **Codificadores de covariables estáticas** (*Static covariate encoders*): integran las características estáticas en la red. Las transforman en representaciones contextuales que el modelo puede usar en el proceso de predicción. Proporciona un contexto adicional que puede influir en el comportamiento de la serie temporal.
- **Procesamiento temporal**: permite al modelo aprender relaciones temporales entre las variables de entrada observadas y conocidas a lo largo de distintos horizontes temporales. Dichas relaciones pueden variar en el tiempo y pueden representar tanto dependencias a corto como largo plazo. Hay dos mecanismos principales con los que se logra.

Por una parte, una capa secuencia a secuencia identifica patrones temporales locales, relaciones que surgen entre valores cercanos en el tiempo.

### 3.2. Modelos estudiados

Esta capa sustituye la codificación posicional de los Transformers, y permite modelar la secuencia mediante conexiones recurrentes, habitualmente a través de una estructura codificador-decodificador basada en LSTM. Los vectores de contexto se usan en este bloque para inicializar el estado de la celda y el oculto de la primera unidad LSTM.

En contraposición, por otra parte, un mecanismo de atención temporal es el que permite capturar las dependencias a largo plazo, identificando los momentos pasados más relevantes para la predicción. Otros métodos utilizan ventanas fijas, pero este asigna pesos dinámicos a los pasos temporales anteriores según su importancia. Además, el modelo ajusta la atención multicabezal clásica compartiendo los pesos entre las cabezas, facilitando la interpretación de qué entradas influyen más en la predicción.

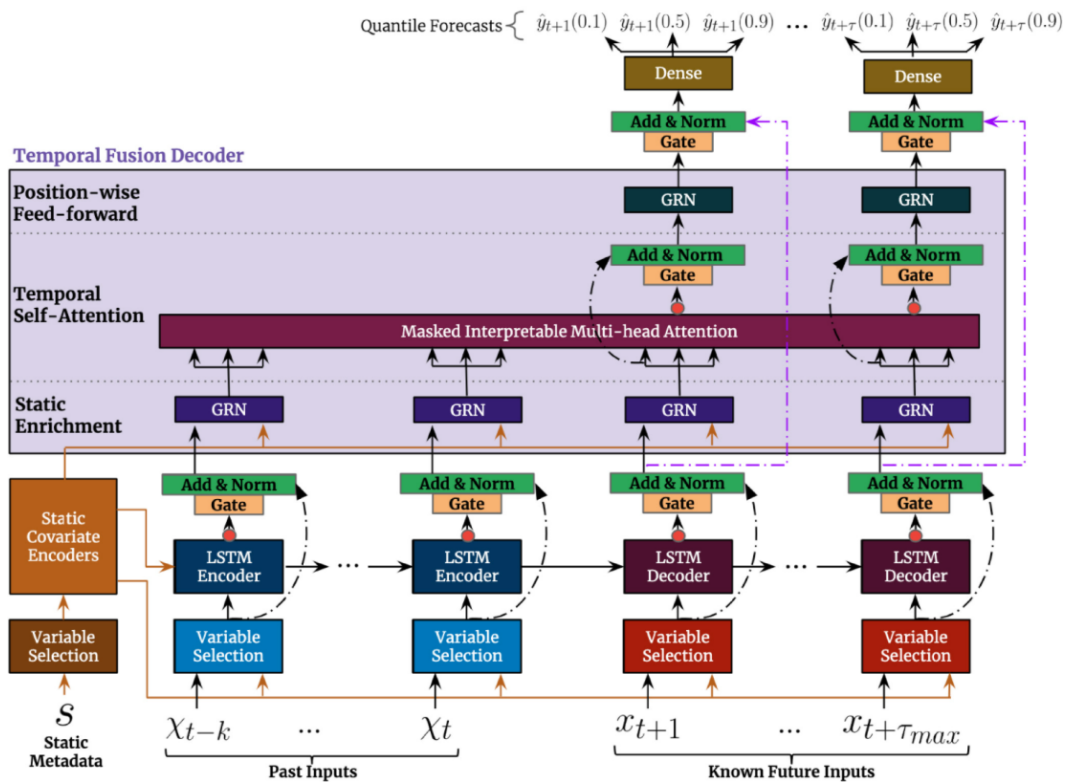


Figura 3.12: Arquitectura de TFT [41]

Se puede concluir que los TFT son modelos potentes, adaptables a diferentes situaciones y ámbitos. Además, son interpretables, permitiendo examinar la importancia de cada variable de entrada en la salida, visualizar patrones temporales e identificar eventos que producen cambios en la dinámica [42].

Toda la formulación matemática se puede consultar en [41].

### 3.3. Estrategias de entrenamiento y predicción

La efectividad en la predicción de series temporales no depende únicamente de la arquitectura y el modelo seleccionado, también es fundamental definir la estrategia que se empleará durante el entrenamiento y la predicción. Establecer cómo se presenta la información al modelo en cada una de las etapas impacta directamente en su rendimiento final. Mientras que en el entrenamiento se dispone de la secuencia completa de datos históricos, en la fase de predicción real se deben obtener valores futuros sin contar con referencias reales. Esta diferencia permite adoptar enfoques específicos en cada una de las fases.

En el presente trabajo se emplean dos técnicas utilizadas en el contexto de series temporales (Figura 3.13).

- **Teacher forcing**, o predicción no autorregresiva, para el entrenamiento.
- **Predicción autorregresiva** para su evaluación.

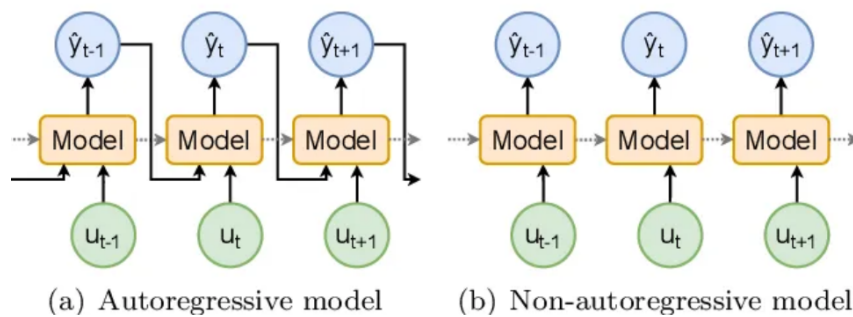


Figura 3.13: *Teacher forcing* y predicción autorregresiva [43]

#### 3.3.1. Teacher forcing

El **teacher forcing** [44] es una técnica de entrenamiento empleada en modelos secuenciales, especialmente en aquellos basados en arquitecturas recurrentes o con mecanismos de atención. Consiste en proporcionar al modelo los valores reales en cada momento, en lugar de su propia predicción durante el aprendizaje. Es decir, en cada paso temporal  $t$ , se proporciona como entrada el valor real en  $t - 1$ , en vez de utilizar la predicción que realizó el instante anterior.

Este enfoque consigue reducir el problema de la propagación de errores, ya que impide que los fallos cometidos al principio de una secuencia afecten a las predicciones posteriores. De esta forma, se consigue que no se acumulen durante el entrenamiento, permitiendo al modelo aprender de forma estable, con una convergencia más rápida.

No obstante, es importante considerar que esto puede introducir un sesgo conocido como sesgo de exposición. Al entrenar con secuencias reales en cada momento, los modelos pueden no desarrollar suficiente robustez a sus errores predictivos, lo que afectaría al rendimiento durante la evaluación.

#### 3.3.2. Predicción autorregresiva

La **predicción autorregresiva** [45], a diferencia del anterior, es una estrategia en la cual el modelo genera predicciones futuras usando sus propias estimaciones en los pasos e instantes realizados. Se inicia con un conjunto de valores reales observados y, a partir de ellos, cada predicción se incorpora en la entrada para obtener el próximo valor. Utilizar esta técnica permite extender el horizonte de predicciones más allá de datos disponibles, lo cual es esencial para aplicaciones prácticas como la de este trabajo.

Sin embargo, este tipo de enfoque conlleva importantes desafíos. El más destacado es la propagación de errores: cualquier desviación en una predicción se transmite a las estimaciones siguientes. A medida que se incrementa el horizonte temporal, el impacto de estas imprecisiones se intensifica, provocando una degradación progresiva del modelo con el paso del tiempo. Para mitigar este efecto, la aplicación de métodos probabilísticos que representen la incertidumbre inherente a las predicciones permite mejorar el rendimiento en escenarios autorregresivos de largo alcance.



## Capítulo 4

# Definición del problema

En el contexto de los entornos industriales modernos, concretamente en el sector petroquímico, la monitorización de variables de difícil acceso en tiempo real se ha convertido en un reto para mejorar la eficiencia y calidad [8]. Los *soft sensors* basados en datos ofrecen una solución a este problema. Permiten inferir dichas magnitudes a partir de variables disponibles, más accesibles del proceso, y sus datos históricos [9].

El presente trabajo plantea el desarrollo de un sensor de este tipo aplicado a un proceso industrial real, con el objetivo de simular el comportamiento de determinadas variables de interés mediante modelos de aprendizaje automático. Para ello, se busca modelar las dinámicas temporales, superando desafíos como el ruido en los datos, la no linealidad y los retardos en las señales, mediante el uso de técnicas avanzadas de Deep Learning. Esta propuesta se alinea con los desarrollos actuales revisados y da respuesta a una necesidad específica en el caso real de un proceso proporcionado por Repsol.

### 4.1. Columna de destilación

El caso de estudio que se aborda en este trabajo corresponde a una planta industrial dedicada a la separación de hidrocarburos mediante la destilación. En concreto, el proceso se centra en una **columna de destilación o separación fraccionada de C3-C4** (propano/propileno y butano), ubicada en el complejo petroquímico de dicha empresa.

Para poder emplear los datos provenientes del proceso y desarrollar los modelos predictivos, primero es esencial conocer y comprender, al menos en términos generales, el funcionamiento de la unidad. De esta manera, se puede entender de forma clara de dónde provienen y, sobre todo, qué representan. Disponer de esta visión de la planta y comprender el significado de sus datos, no solo permite mejorar el preprocesamiento de ellos, sino también interpretar mejor los resultados y tomar decisiones más acertadas durante el desarrollo.

Con este propósito, en la Figura 4.1 se muestra una representación esquemática del proceso con el que se va a trabajar. Aunque no corresponde al diagrama

## Capítulo 4. Definición del problema

exacto de la planta de la empresa, por cuestiones de confidencialidad, recoge y refleja su funcionamiento general. A partir de esta imagen, se explicará, a continuación, el principio de operación de este tipo de columnas y los componentes principales del sistema, necesarios para comprender los datos registrados. Cabe destacar que se llevará a cabo una explicación del funcionamiento sin entrar en los detalles matemáticos o físicos, al quedar este alcance fuera del trabajo.

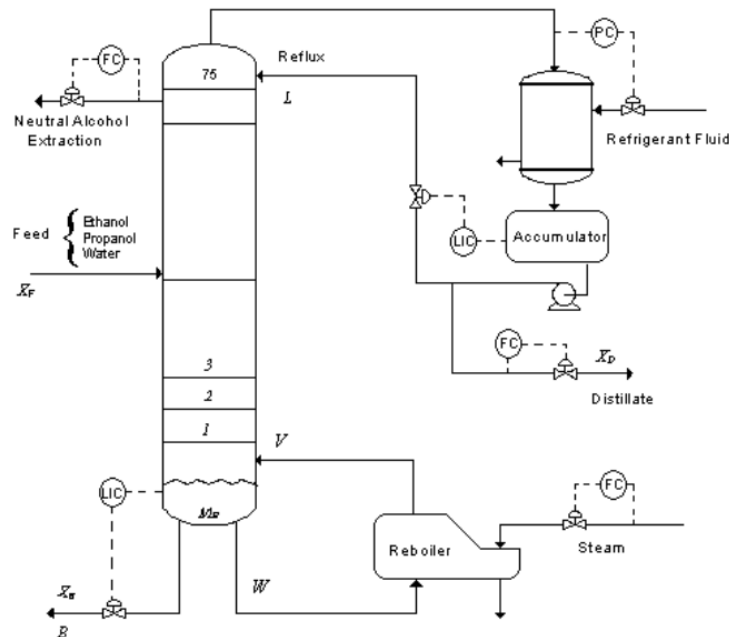


Figura 4.1: Esquema simplificado del proceso industrial [46]

Una de las operaciones más empleadas en el sector químico y petroquímico es la destilación. Se trata de un proceso que permite separar diferentes componentes de una corriente y su uso abarca una gran variedad de aplicaciones [46]. Su funcionamiento se basa en las diferencias de volatilidad de los compuestos que forman la mezcla, utilizando el calentamiento y enfriamiento para conseguir esa separación [47]. En este caso concreto, como se ha mencionado, el estudio se centra en una columna de destilación que trata de separar compuestos de C3 (propano, propileno) de compuestos de C4 (butano) a partir de una entrada de gas licuado de petróleo (GLP).

Para conseguir este propósito, una planta de estas características tiene los siguientes componentes [47], como se puede observar en la imagen:

- **Columna de fraccionamiento:** es el principal elemento del sistema en el que se lleva a cabo la separación. Está formada por unas bandejas o platos que favorecen el intercambio de materia entre el líquido y el vapor dentro de esta, mejorando así la eficacia del proceso.
- **Rehervidor:** conocido como *reboiler*, es un intercambiador de calor. Proporciona el calor necesario que hace que parte de la entrada a la columna se vaporice. En la planta específica con la que se trabaja está diseñado

con dos compartimentos: una cámara superior donde se acumula el líquido a calentar, y una inferior con la fuente de calor, normalmente vapor o aceite térmico. Desde la parte inferior se transfiere el calor al líquido del intercambiador, el cual se evapora y entra a la columna.

- **Condensador:** se encuentra en la parte superior y enfría el vapor de la alimentación separado que sale de la columna, condensándolo antes de llegar al tambor de reflujo.
- **Tambor de reflujo:** situado justo después del condensador, recoge el vapor condensado que sale de este y crea dos corrientes: la salida final del proceso, conocido como destilado, y una segunda que recibe el nombre de reflujo. La segunda corriente vuelve de nuevo a la columna de separación. Su función es mejorar la calidad del destilado ya que al introducirse por la parte superior condensa parte del vapor ascendente, lo que facilita una mejor separación de los componentes más volátiles.

Con todos los componentes principales identificados, lo siguiente es analizar el **funcionamiento** global de la planta y cómo se relacionan e interactúan entre sí [47], [48].

El líquido o corriente de alimentación, GLP en este caso, entra en la columna de fraccionamiento por el plato correspondiente, generalmente situado en el centro o cerca del mismo. La mezcla desciende por gravedad a través de los diferentes platos, interactuando con el vapor que asciende desde la base. Parte de la alimentación cae y se acumula en la base, donde pasa al *reboiler*. En él se produce la vaporización parcial del líquido, que regresa de nuevo a la columna en forma de vapor ascendiendo por ella.

En su camino, el vapor atraviesa las diferentes bandejas entrando en contacto con el líquido descendente en cada una. Este intercambio favorece una **transferencia de masa eficiente**. El componente más ligero, con menor punto de ebullición, se enriquece; mientras que los más pesados, con mayor punto de ebullición, permanecen líquidos. A medida que se avanza hacia el tope, el vapor se compone del elemento más volátil. En la base de la columna, sin embargo, se concentran los productos de menor volatilidad, que no logran evaporarse, y se extraen como producto de fondo.

El vapor que alcanza el tope se dirige al condensador, en el que se enfría para ser convertido de nuevo en líquido. Este líquido continúa hasta el tambor de reflujo, donde se divide en las dos corrientes: el reflujo, que se recicla y vuelve de nuevo a la columna entrando por la parte superior, y el resto, extraído como producto final y componente destilado. El reflujo es una parte esencial del funcionamiento, ya que mantiene la eficiencia de la separación. Permite condensar parte del vapor ascendente enriqueciendo aún más la mezcla.

En definitiva, el funcionamiento conjunto del sistema permite separar con precisión una corriente de GLP. Se busca obtener propano y propileno (C3) como destilado, y butano (C4) como producto de fondo. Esta separación se basa en las diferencias de volatilidades de los productos, ajustable mediante variables operativas como la temperatura en la columna, la presión o la tasa de reflujo.

### 4.2. Datos históricos

Una vez se ha comprendido el proceso industrial en el que se basa este trabajo, el siguiente paso consiste en familiarizarse con el conjunto de datos disponibles. Este análisis resulta imprescindible para construir una base sólida sobre la que desarrollar los modelos en etapas posteriores.

El análisis comienza con una revisión detallada de las variables incluidas en el **dataset**, identificando el significado de cada una de ellas. De estas, únicamente se seleccionan aquellas que resultan relevantes y aportan información al estudio. Con las variables necesarias definidas, se procede a un análisis de las principales características de los datos. El análisis descriptivo se complementa con visualizaciones que permiten obtener información adicional, facilitando la interpretación de patrones o comportamientos en las series.

El objetivo de esta fase es consolidar una buena base sobre la que desarrollar los modelos y construir el *soft sensor*. Se pretende adquirir una comprensión exhaustiva del comportamiento de las variables disponibles y el contexto en el que se han registrado.

#### 4.2.1. Librerías empleadas en el análisis exploratorio

Durante la realización de esta primera fase de comprensión y análisis de los datos se han empleado las siguientes librerías en el lenguaje de programación Python. Todas ellas facilitan la exploración detallada del conjunto inicial, además de su visualización.

- **Pandas:** una librería muy empleada y fundamental para la manipulación y análisis de estructuras de datos, concretamente aquellas tabulares. Ofrece diversas funcionalidades, desde la carga de los datos, hasta la limpieza, el filtrado o funciones estadísticas básicas [49].
- **Numpy:** junto con Pandas, otra de las librerías comúnmente conocida y usada en este lenguaje. Permite realizar operaciones vectorizadas y manejar *arrays* multidimensionales. Su utilidad se encuentra en el tratamiento de los valores numéricos y en los cálculos matemáticos [50].
- **Matplotlib:** herramienta principal para la representación visual de datos, diseñada para realizar gráficos con los que facilita identificar tendencias, distribuciones o relaciones entre variables [51].
- **Seaborn:** construida sobre Matplotlib, genera visualizaciones estadísticas más sofisticadas y elaboradas, de forma sencilla [52].

El uso conjunto de estas librerías ha permitido realizar una exploración profunda de los datos, mostrada a continuación.

#### 4.2.2. Descripción del conjunto de datos inicial

Los datos proporcionados para este estudio contienen registros operativos reales de una columna de destilación C3-C4 perteneciente a una unidad de Repsol, co-

mo se mencionó anteriormente. Abarcan un período de aproximadamente cinco años y medio, siendo el primer registro del 12 de junio de 2019 y el último del 14 de enero de 2025 a las 23:59. La frecuencia de muestreo es de un minuto, lo que implica que se dispone de un valor para cada variable correspondiente a cada minuto comprendido en ese intervalo.

El *dataset* utilizado incluye información relativa a una amplia serie de mediciones obtenidas a través de sensores situados en todo el sistema. Entre las variables registradas se encuentran temperaturas en diferentes posiciones de la columna, presiones (en el fondo y cabeza), caudales de entrada y salida del proceso y concentraciones de los productos separados.

Inicialmente, con el apoyo de los expertos conocedores del sistema, además del uso de manuales técnicos e información complementaria, se realizó una selección y depuración de las variables disponibles. El archivo original contenía un total de 40 variables. A partir de este conjunto inicial, se seleccionaron aquellas consideradas más relevantes para los objetivos planteados. Dicha selección, fundamentada en criterios técnicos de los especialistas del proceso que trabajan con el sistema, redujo el número a un total de 10.

Hay tres grupos principales entre las que se dividen esas variables según su naturaleza dentro del contexto del control de procesos:

- VARIABLES DE PERTURBACIÓN: reflejan condiciones externas que afectan al proceso y su comportamiento. Influyen en el sistema, pero no pueden ser modificadas ni controladas por los operarios.
- VARIABLES MANIPULADAS: pueden ser modificadas por el sistema de control para alcanzar los objetivos deseados en la planta.
- VARIABLES CONTROLADAS: generalmente corresponden a indicadores de calidad o estabilidad, son aquellas que se desean mantener dentro de ciertos rangos o condiciones. Para ello se actúa sobre las variables manipuladas. En este caso, el propósito del trabajo es poder estimar estas variables en tiempo real a partir de sus propios registros, y los de los otros dos tipos de variables. Comprobar cómo afectan modificaciones en los dos primeros tipos de variables a estas últimas.

A continuación, se presentan las 10 variables con las que se seguirá trabajando, agrupadas según su función en el sistema.

Nombre de la variable	Descripción breve
<b>Perturbación</b>	
<b>FC_alimentación_splitter</b>	Caudal de alimentación hacia la columna de separación. Refleja la carga del sistema.
<b>T_amb_C</b>	Temperatura ambiente. Influye en el equilibrio térmico del sistema.

## Capítulo 4. Definición del problema

---

<b>LC_separador</b>	Nivel de líquido en el separador. Es la condición previa al ingreso en la columna.
<b>Manipulación</b>	
<b>TI_plato_6</b>	Temperatura en el plato 6, puede ajustarse para controlar la eficiencia del corte.
<b>FC_reflujo</b>	Caudal de reflujo. Es un parámetro clave para el control de la pureza en la cabeza de la torre.
<b>FC_salida_C3</b>	Caudal de salida del producto C3. Es el flujo extraído en la cabeza.
<b>Controlada</b>	
<b>PC_tope</b>	Presión en la parte superior de la columna.
<b>FI_salida_C4</b>	Caudal de salida del producto C4. Es el flujo extraído por el fondo.
<b>TI_plato_26</b>	Temperatura en el plato 26, está vinculada a la separación de los componentes.
<b>TI_cabeza</b>	Temperatura en la cabeza, también relacionada con la composición del producto destilado.

Tabla 4.1: Variables seleccionadas del *dataset*

De esta forma, las diez variables presentadas constituyen la base sobre la que se desarrolla el análisis posterior y el sensor. Una vez clasificadas, se definen los conjuntos de entrada (*inputs*) y salida (*outputs*) de los modelos. En este sentido, las variables de perturbación, junto con las manipuladas y el historial reciente de las controladas, serán la entrada de los modelos. Por otra parte, las de salida corresponden a las variables controladas en los minutos que se deseen estimar, desde el instante actual hasta el horizonte temporal definido.

En los siguientes apartados se analizan en profundidad las características temporales, así como aspectos clave de su calidad y estructura.

### 4.2.3. Características temporales

En esta sección se pretende describir las características temporales relevantes del *dataset* seleccionado. Dado que se trata de registros recopilados de forma continua, y van a ser tratados como series temporales, resulta imprescindible conocer e identificar comportamientos que puedan condicionar el desarrollo. Es fundamental describir aspectos como el intervalo temporal que se va a emplear, la frecuencia de muestreo y el volumen de datos.

Aunque el periodo cubierto por los datos y su frecuencia han sido introducidos en el apartado previo, conviene retomarlos brevemente para mostrar y visuali-

## 4.2. Datos históricos

zar las series temporales en conjunto. Se cuenta con casi cinco años y medio de registros, desde el 12 de junio de 2019 hasta el 14 de enero de 2025, con una frecuencia de un minuto. Este nivel de granularidad da lugar a un total de 2.943.065 observaciones por variable.

Los datos presentan una estructura de serie temporal multivariante, donde cada una de las variables evoluciona a lo largo del tiempo y puede estar influenciada por las demás. Para explorar la dinámica temporal, en la Figura 4.2 se representan gráficamente todas las variables, lo que permite visualizar tanto su comportamiento como las características descritas hasta el momento. La frecuencia de muestreo con la que se cuenta permite capturar dinámicas a corto plazo, pero introduce a su vez ruido y posibles valores atípicos debido a condiciones operativas anómalas. Esto también se puede observar en la imagen.

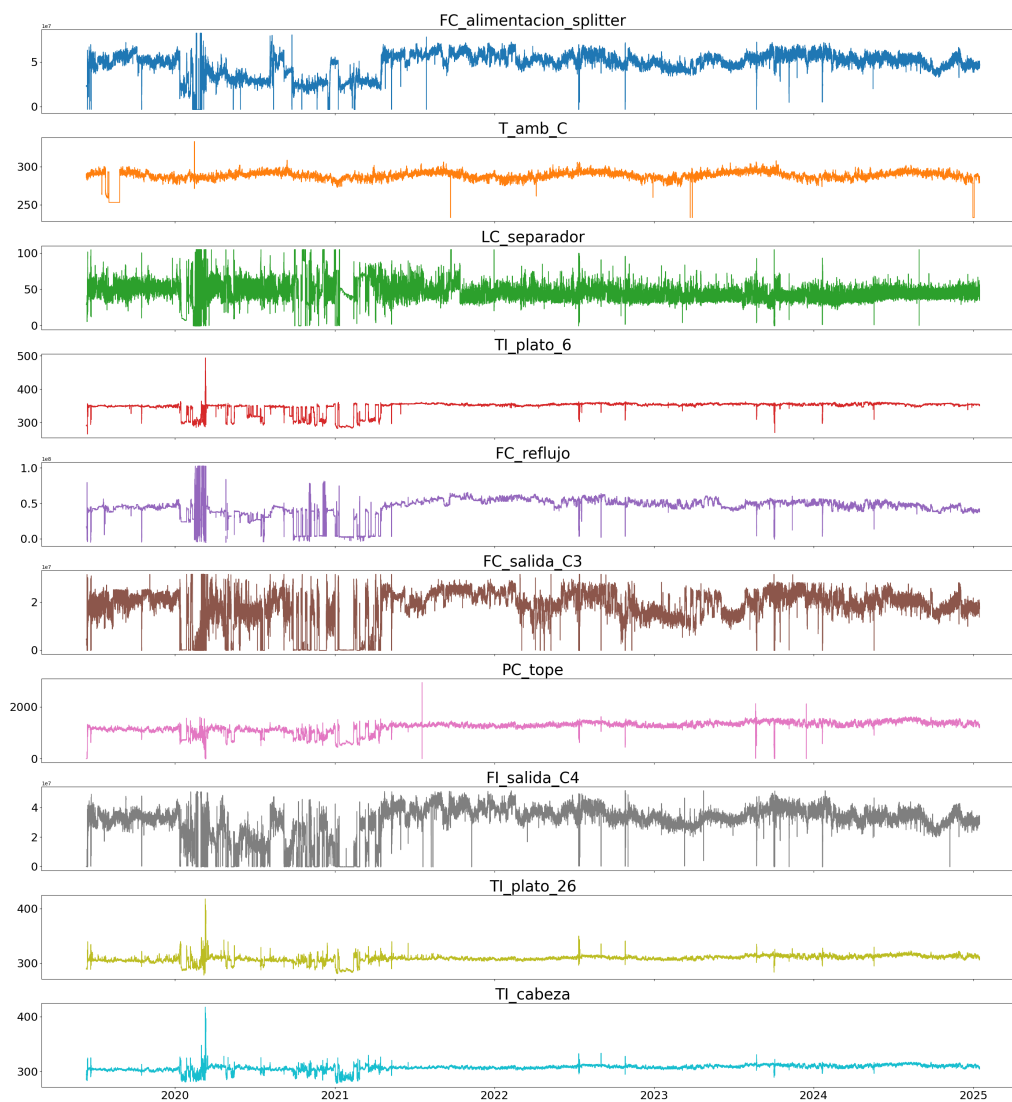


Figura 4.2: Series temporales completas de las diez variables seleccionadas

## Capítulo 4. Definición del problema

---

Al mostrar y visualizar todas las series temporales que van a ser usadas, se puede observar que, en la primera parte de los datos hay situaciones anómalas. Entre 2020 y la mitad, aproximadamente junio, de 2021, las variables parecen estar descontroladas, con alteraciones significativas, en comparación con el resto del histórico. Durante toda la secuencia se aprecia que hay valores atípicos, pero en ese intervalo parece haber una mayor variabilidad de lo normal. Esas anomalías justo coinciden con la pandemia de COVID-19, por lo que es razonable suponer que corresponden a condiciones operativas especiales tomadas durante ese contexto.

Como se cuenta con una amplia cantidad de datos, se ha decidido prescindir de esa parte, ya que reflejan situaciones que actualmente no son representativas del funcionamiento normal. Se trata de condiciones que, a menos que se dé otra pandemia, en principio no tendrían por qué ocurrir de nuevo de esa forma. Así, el nuevo *dataset* abarcará desde el 1 de julio de 2021 hasta el final, suponiendo un total de 1.863.183 datos por variable.

Por otra parte, a simple vista se aprecian posibles tendencias y estacionalidades en algunas de las variables, como la temperatura ambiente (serie naranja), la salida de C3 (serie marrón) o la salida de C4 (serie gris). Aparte de las tendencias, es evidente la presencia de ruido a lo largo de todo el histórico. Estas cuestiones, los patrones periódicos (anuales, mensuales y diarios) y la presencia de *outliers*, se analizarán con más detalle en las próximas secciones.

### 4.2.4. Análisis exploratorio de las variables

Una vez definido el conjunto total de los datos finales con los que se va a trabajar (tanto las variables como el intervalo de tiempo en las que las vamos a considerar) se procede a realizar un análisis más exhaustivo. Se va a llevar a cabo una exploración con el fin de evaluar la calidad, completitud y comportamiento general. Cabe destacar que en esta fase únicamente se va a realizar el análisis, puesto que el tratamiento y limpieza se abordarán posteriormente en la etapa de preparación para los modelos.

#### Tipo de dato y coherencia general

Cuando se emplean arquitecturas de aprendizaje automático, es fundamental considerar y controlar el tipo de los datos empleados. Para poder trabajar correctamente y entrenar los modelos, estos tienen que tomar valores numéricos, ya que los algoritmos no pueden operar con variables categóricas. En el caso del *dataset* elegido, todas las variables son de tipo *'float'*, como se ha observado en la Figura 4.2. Tampoco se han detectado codificaciones erróneas ni tipos inconsistentes. Todo ello facilita el tratamiento posterior, ya que no será necesario transformar ni realizar ninguna codificación de los datos.

#### Estadísticas descriptivas

Para comprender mejor cada una de las variables y conocer información más específica sobre sus rangos de operación y dispersión, es interesante estudiar las

## 4.2. Datos históricos

estadísticas descriptivas más relevantes. En la Tabla 4.2 se muestran las más relevantes: mínimo, máximo, media, desviación típica y los percentiles principales (25, 50, 75).

Variable	FC_aliment.	T_amb_C	LC_separador	TI_plato_6	FC_reflujo
<b>Media</b>	$5,25 \times 10^7$	288,59	46,17	354,17	$5,02 \times 10^7$
<b>D. Típica</b>	$6,76 \times 10^6$	5,23	8,28	3,84	$6,28 \times 10^6$
<b>Mín</b>	$-3,20 \times 10^6$	233,15	-0,76	269,38	$-8,63 \times 10^5$
<b>Máx</b>	$7,78 \times 10^7$	307,79	104,85	362,15	$6,54 \times 10^7$
<b>25 %</b>	$4,78 \times 10^7$	285,78	41,22	352,37	$4,63 \times 10^7$
<b>50 %</b>	$5,28 \times 10^7$	288,92	45,45	354,56	$5,05 \times 10^7$
<b>75 %</b>	$5,74 \times 10^7$	291,77	50,18	356,46	$5,45 \times 10^7$

Variable	FC_salida_C3	PC_tope	FI_salida_C4	TI_plato_26	TI_cabeza
<b>Media</b>	$2,02 \times 10^7$	1347,22	$3,51 \times 10^7$	310,70	309,18
<b>D. Típica</b>	$4,26 \times 10^6$	101,96	$5,23 \times 10^6$	3,20	2,78
<b>Mín</b>	$-1,84 \times 10^5$	0,16	-0,76	283,41	289,21
<b>Máx</b>	$3,15 \times 10^7$	307,79	2943,27	349,70	333,34
<b>25 %</b>	$1,76 \times 10^7$	285,78	1279,27	308,16	307,01
<b>50 %</b>	$2,05 \times 10^7$	288,92	1343,41	310,56	309,03
<b>75 %</b>	$2,35 \times 10^7$	291,77	1412,57	312,98	311,207

Tabla 4.2: Estadísticas descriptivas de las variables

A partir de las estadísticas mostradas, se pueden extraer una serie de observaciones sobre el comportamiento y la calidad general.

- Homogeneidad y baja simetría: Algunas de las variables, aquellas relacionadas con la temperatura, parecen presentar distribuciones relativamente homogéneas y acotadas. Las medias son próximas a las medianas (percentil 50), por lo que la asimetría es baja. Por ejemplo, 'TI\_plato\_6' muestra una media de 354,17°C y una mediana de 354,56°C, con una desviación de 3,84, baja en comparación con su rango. Sugiere estabilidad en las mediciones y un funcionamiento controlado de esas partes.
- Rangos amplios y asimetría marcada en otros casos: Ciertas magnitudes, especialmente los caudales tanto de entrada como de salida muestran un rango amplio. Contienen grandes diferencias entre los mínimos y máximos, con considerables desviaciones típicas. En el caso de la presión, 'PC\_tope',

## Capítulo 4. Definición del problema

---

el rango abarca desde 0,15 hasta 3000, y, aunque la media y mediana están alineadas, la diferencia con el máximo sugiere la existencia de colas en la distribución. Todo ello se debe a variaciones operativas intensas o a presencia de valores atípicos, lo cual será analizado en esta misma sección.

- **Presencia de valores extremos o inconsistentes:** Existen parámetros donde hay una gran diferencia entre mínimos y máximos, incluso alcanzando valores negativos incoherentes. En la alimentación de la columna el mínimo es -3 millones, pero la media y mediana se encuentran en torno a 52 millones, reflejando una presencia de valores claramente anómalos. Con el resto de flujos y caudales se produce la misma situación. Los valores extremos, debidos probablemente a errores de sensorización o medición, deberán tratarse antes de la modelización.
- **Cohesión entre temperaturas:** Las mediciones de temperaturas se sitúan dentro de los márgenes operativos razonables, manteniendo una progresión coherente. La baja dispersión, desviación típica y la similitud entre las medias de estas señales refleja esa consistencia operativa.
- **Relación entre caudales:** Todas las medidas de caudales (variables que comienzan por F) tienen una magnitud similar. Como sería lógico este proceso, sugiere que están relacionadas entre sí dentro del balance de masa del sistema. El flujo de entrada se repartiría entre los de salida de C3 y C4, mientras que como estabilizador interno se encontraría el reflujo.

### Valores nulos y negativos

Se ha verificado la presencia de valores nulos y negativos en cada una de las variables consideradas. En la Tabla 4.3 se recogen la cantidad total de estos.

Variable	Valores nulos	Valores negativos
<b>FC_alim.</b>	22	333
<b>T_amb_C</b>	64	0
<b>LC_separador</b>	60	1116
<b>TI_plato_6</b>	40	0
<b>FC_reflujo</b>	34	2
<b>FC_salida_C3</b>	38	7140
<b>PC_tope</b>	25	0
<b>FI_salida_C4</b>	871	1017
<b>TI_plato_26</b>	36	0
<b>TI_cabeza</b>	36	0

Tabla 4.3: Valores nulos y negativos por variable

Respecto a los valores **nulos**, todos los parámetros contienen una cierta cantidad, aunque su proporción respecto al total en general es muy baja. Tomando como ejemplo la salida de C4 ('FI\_salida\_C4') como la que más valores nulos presenta con un total de 871 y considerando el total de datos de los que se dispone, poco más de 1.800.000, la proporción de valores nulos supone menos del 0,05%. Aunque la existencia de estos valores no es deseada, no debería suponer un problema grave al ser una proporción tan pequeña.

Por otra parte, la presencia de valores **negativos** en algunas variables no tiene sentido físico. No es posible que, por ejemplo, el caudal de salida de la columna, tanto de C3 como de C4 ('FC\_salida\_C3' y 'FI\_salida\_C4'), o el de entrada sean negativos. 'FC\_salida\_C3' es quizás la variable más problemática, pues es aquella que presenta un mayor número, 7140 en total. Aun así, también suponen una proporción muy pequeña de los registros totales. Para este proyecto, se asume que se trata de errores en la adquisición de los datos. En muchos casos, estos valores aparecen próximos a cero, lo que sugiere oscilaciones o errores en la calibración de los sensores. Las observaciones mencionadas deberán tenerse en cuenta para el tratamiento en la siguiente fase.

### Datos faltantes o no registrados

Además de los valores nulos explícitos, existen registros del *dataset* que no presentan valor en alguna de las variables, por lo que se decidió comprobar posibles periodos faltantes. Analizando la continuidad de los datos con la frecuencia de un minuto establecida se ha descubierto que hay tres periodos de tiempo en los que faltan mediciones.

- 22 de junio de 2022 a las 01:58 hasta las 02:56 (59 minutos)
- 22 de junio de 2023 a las 02:57 hasta las 03:55 (59 minutos)
- 21 de junio de 2024 a las 03:56 hasta las 04:54 (59 minutos)

Los tres periodos presentan la misma duración y tienen lugar en fechas muy similares cada año. El parón en cada uno de ellos, en momentos tan parecidos y 59 minutos más tarde (exactamente la misma cantidad de valores faltantes), sugiere la posibilidad de una parada programada del sistema o tareas de mantenimiento recurrentes. En el tratamiento posterior habrá que comprobar si forman parte efectivamente de una tendencia periódica para solventarlo de la manera más conveniente.

### Detección de outliers

Finalmente, se ha realizado una identificación de posibles valores atípicos. Para ello, se ha utilizado un enfoque basado en el rango intercuartílico (IQR), identificando observaciones anómalas alejadas de la distribución central. Como puntos atípicos han sido considerados aquellos situados por debajo del cuartil 10 y por encima del 90, ampliando los límites con un pequeño factor multiplicativo para mejorar la sensibilidad.

## Capítulo 4. Definición del problema

Prácticamente todas las variables presentan cierta proporción de *outliers*, aunque en diferente medida. En la siguiente imagen, Figura 4.3, se representa gráficamente la presencia de dichos valores, marcados en rojo aquellos detectados para cada una de las series temporales.

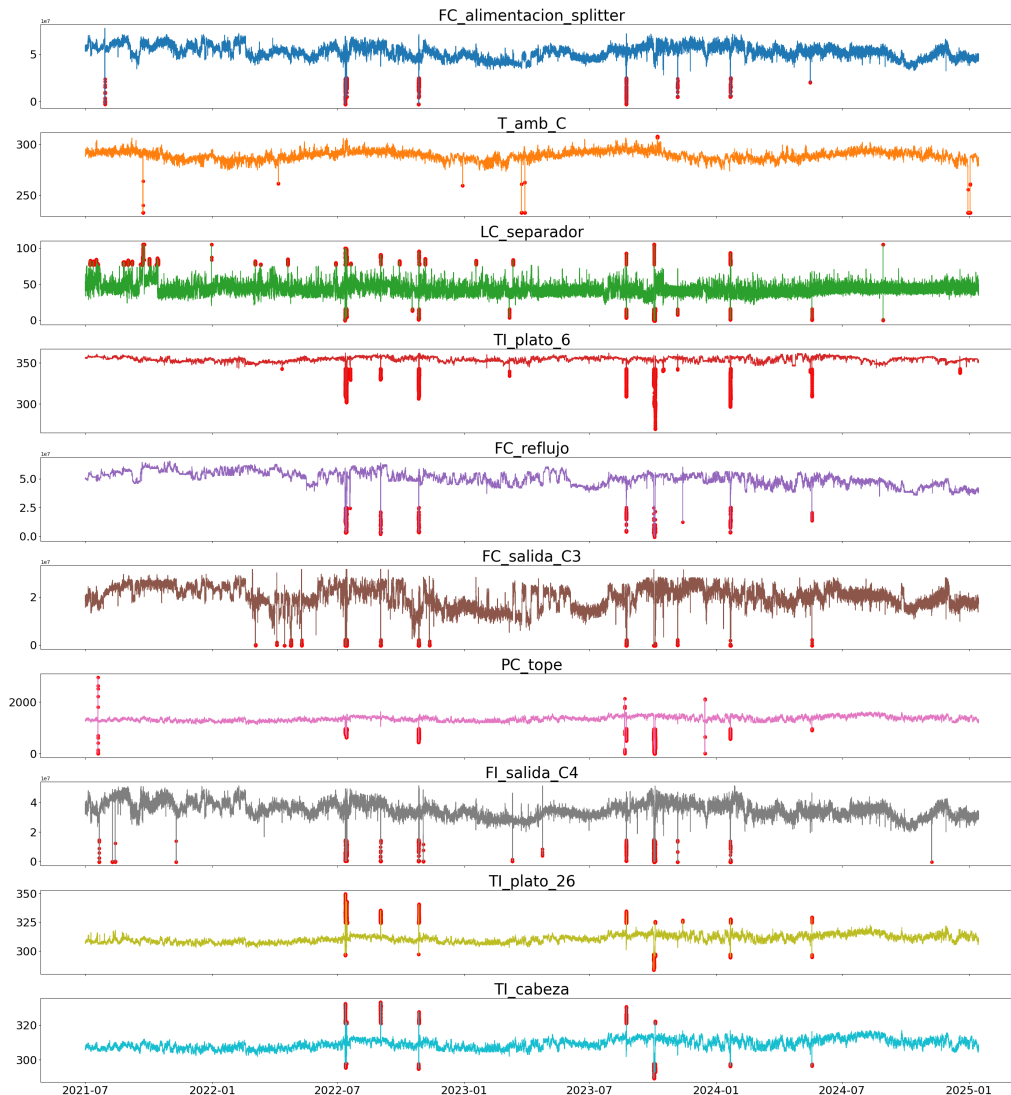


Figura 4.3: Datos atípicos en cada una de las series temporales

Como era de esperar con la primera observación que se hizo de las series en la Figura 4.3, todas las señales contienen un cierto número de valores atípicos. Su presencia no se concentra únicamente en un momento específico, sino que están distribuidos a lo largo de todo el horizonte temporal. La dispersión por tanto sugiere que no se deben a eventos aislados, sino a fluctuaciones que pueden repetirse a lo largo de las fases del proceso.

La frecuencia de estos puntos varía entre las diferentes variables. Aquellas relacionadas con el caudal o flujo ('LC\_separador', 'FI\_salida\_C4') acumulan un ele-

vado número de observaciones fuera del rango común, conteniendo más variabilidad e inestabilidades. Sin embargo, las mediciones de temperatura muestran una presencia mucho menor, con comportamientos más estables y controlados. Se incluye una tabla con el número total de outliers identificados de esta manera para cuantificar estas diferencias.

Variable	Número de outliers
<b>FC_alim.</b>	3678
<b>T_amb_C</b>	4764
<b>LC_separador</b>	10842
<b>TI_plato_6</b>	7138
<b>FC_reflujo</b>	6433
<b>FC_salida_C3</b>	7713
<b>PC_tope</b>	5229
<b>FI_salida_C4</b>	6774
<b>TI_plato_26</b>	1732
<b>TI_cabeza</b>	2254

Tabla 4.4: Valores atípicos por variable

### 4.2.5. Visualizaciones complementarias

Con el objetivo de ampliar el análisis ya realizado, más centrado en la calidad y completitud de los datos, se incluye una serie de visualizaciones adicionales que permiten extraer patrones, relaciones y comportamientos. Las siguientes representaciones gráficas aportan una perspectiva más intuitiva sobre la dinámica y las dependencias entre las variables.

En particular, se ha añadido un mapa de calor para conocer las correlaciones, las distribuciones individuales y un análisis de la estacionalidad (anual, mensual y semanal).

#### Mapa de calor de correlaciones

Antes de profundizar en aspectos individuales, se ha elaborado un mapa de calor (*heatmap*) de las correlaciones de Pearson para explorar la posible existencia de relaciones entre las diferentes variables. Permite identificar pares con asociaciones fuertes, ya sean positivas o negativas.

## Capítulo 4. Definición del problema

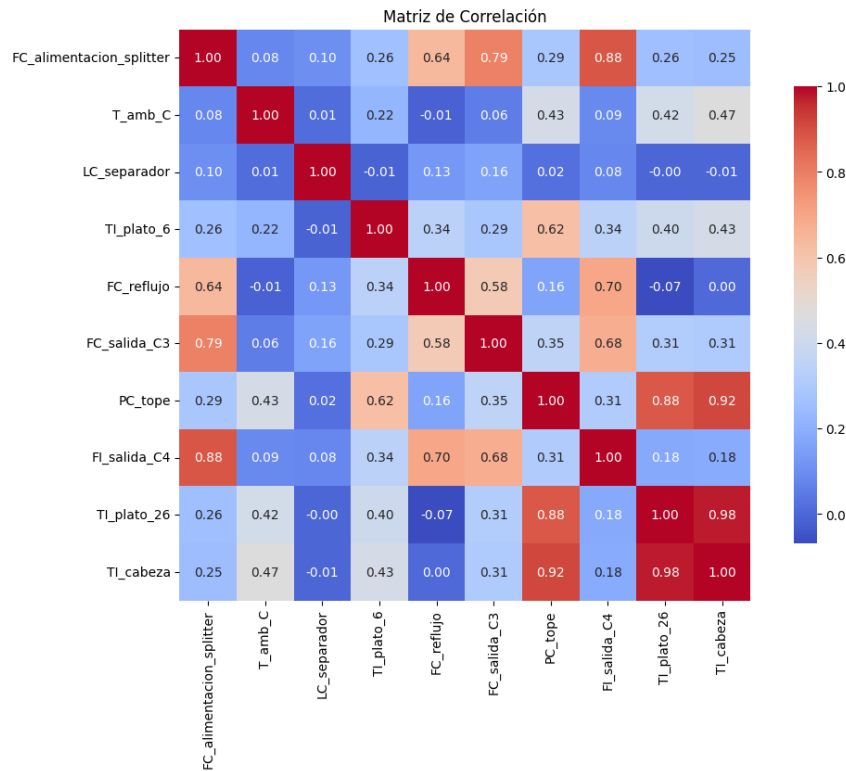


Figura 4.4: Mapa de calor de correlaciones

Tal como se observa en la imagen anterior, Figura 4.4, existen correlaciones destacadas entre algunas de las variables, reforzando las relaciones esperadas en el comportamiento del sistema. Las temperaturas, en la cabeza, en el plato 26 y en el plato 6 muestran correlaciones altas, lo que es coherente con el perfil térmico a lo largo de la columna. Del mismo modo, 'PC\_tope' también presenta correlaciones considerables con estas tres, en línea con su sensibilidad a las variaciones térmicas producidas en los distintos niveles. Otra relación también interesante que ya se ha mencionado, es la dada entre 'FC\_alimentacion', 'FC\_salida\_C3' y 'FI\_salida\_C4', visible en la imagen. Estas tres variables también están relacionadas de manera muy significativa con el reflujo, lo que es coherente por su función de control de caudal en el tope de la torre.

Por otro lado, algunas variables como 'T\_amb\_C' o 'LC\_separador' muestran correlaciones generalmente bajas con el resto, salvo en alguna excepción. Su impacto parece ser más indirecto o dependiente de otros factores.

Con este análisis se aporta una visión de las dependencias lineales existentes, lo que valida el comportamiento del sistema.

### Distribuciones individuales

A continuación, se analizan las distribuciones individuales de las variables mediante histogramas. Facilitan observar la forma general de los datos, identificar sesgos, concentraciones o colas. Lo visto anteriormente en el análisis estadísti-

co se puede reforzar con esta visualización, además de complementarlo con la detección de patrones poco habituales.

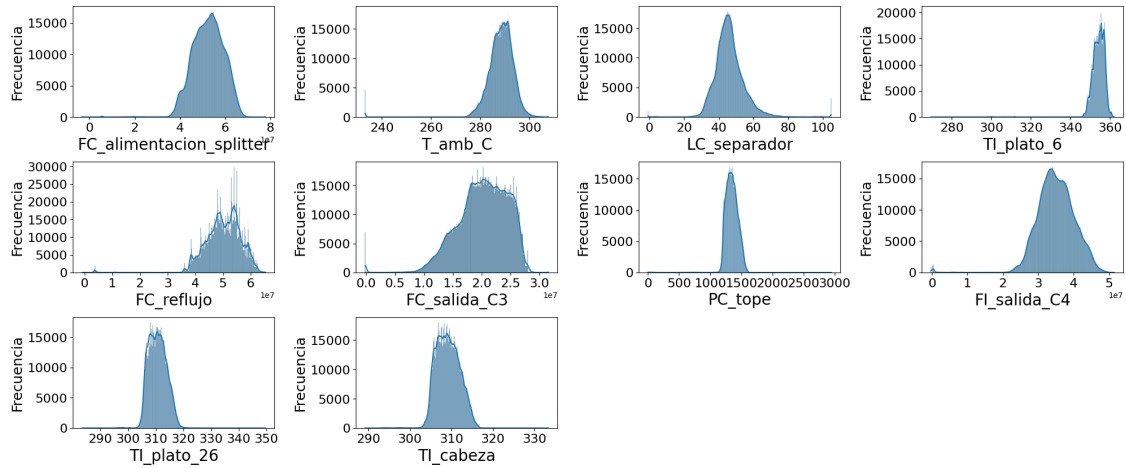


Figura 4.5: Distribuciones de las variables seleccionadas

En general, se aprecian distribuciones con comportamientos esperables según el tipo de variable y su significado. Las temperaturas tienden a agruparse a valores centrales con distribuciones unimodales y simétricas, consistentes con un funcionamiento estable. En concreto, 'T\_amb\_C' contiene una distribución más ancha por su carácter estacional.

Las variables de flujo, a diferencia de las anteriores, muestran colas más prolongadas y asimetrías. Sugiere y se confirma la presencia de oscilaciones y valores extremos. 'LC\_separador' también presenta cierta dispersión, aunque en márgenes coherentes con lo visto en el análisis exploratorio anterior.

### Autocorrelación

En esta sección se analiza la autocorrelación de las variables controladas, objetivo del modelo, es decir, aquellas que se desea estimar mediante el desarrollo de *soft sensor*. Son el foco del modelado por lo que estudiar su comportamiento interno resulta relevante.

La autocorrelación permite evaluar la forma en la que una variable está relacionada con sus instantes anteriores. En series temporales resulta especialmente interesante debido a la secuencialidad de los datos. Puede revelar patrones de persistencia, repetición o dependencia que influyen en la capacidad de predicción. En este caso, se ha decidido estudiar en las variables controladas, es decir, aquellas que se desea predecir. El objetivo es conocer la información que influye en el instante actual y con ello definir el tamaño de la ventana temporal necesaria.

Los resultados muestran que las cuatro variables tienen una autocorrelación muy elevada, sobre todo en los primeros retardos. El valor de ellas en un instante está condicionado en gran medida por su propio historial reciente, lo cual

## Capítulo 4. Definición del problema

es coherente con la situación dada. En los entornos industriales, los sistemas suelen presentar dinámicas continuas en tiempo con variaciones suaves y progresivas, lo que unido a la frecuencia de muestreo tan pequeña, confirma los resultados obtenidos. Esta fuerte autocorrelación refuerza la idea de que la historia reciente de una variable contiene información útil para predecir su evolución.

### Estacionalidad

En procesos continuos, las condiciones de operación pueden verse influidas por factores estacionales. Puede resultar de ayuda examinar si las variables objetivo tienen patrones que tienden a repetirse cíclicamente a lo largo del tiempo. Se han descompuesto las series temporales de interés según diferentes granularidades temporales. A continuación, se muestran los resultados en los que se puede observar cómo varían los valores medios a lo largo del año, los distintos meses y las horas del día.

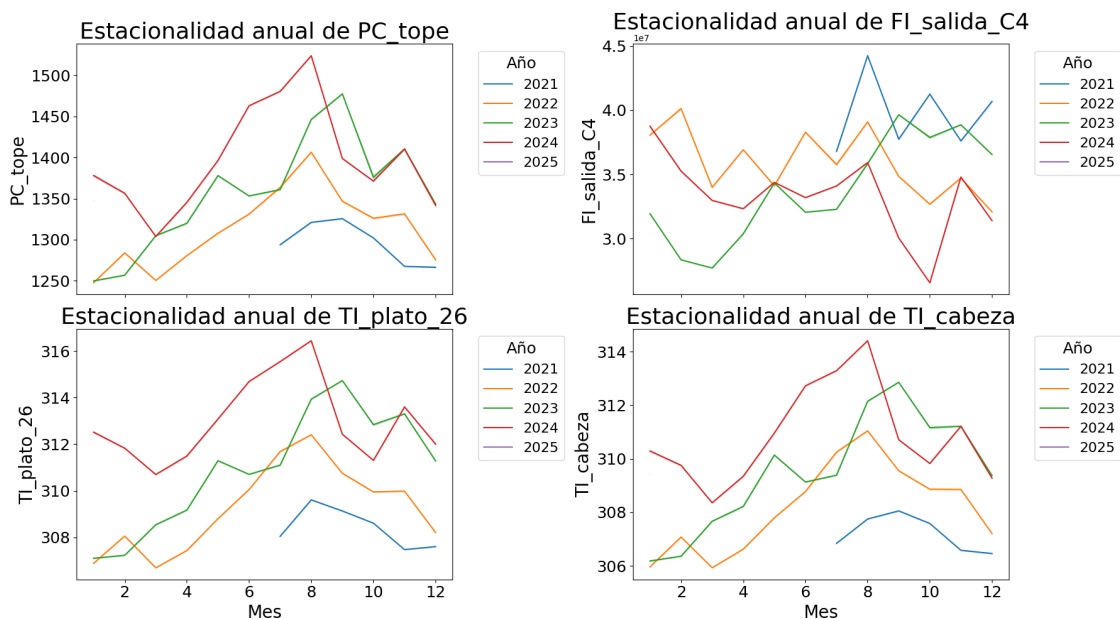


Figura 4.6: Estacionalidad anual de las variables controladas

En la Figura 4.6 se observa una componente anual en tres de las cuatro variables. Tanto la presión como las temperaturas evolucionan de forma ascendente desde principios de año hasta los meses de verano, y descendente desde ahí hasta finales de año. La dinámica refleja la influencia de la temperatura ambiente y condiciones externas en el equilibrio térmico de la columna. Sin embargo, 'FI\_salida\_C4' no presenta ningún patrón estacional, con mayor dispersión entre los años.

## 4.2. Datos históricos

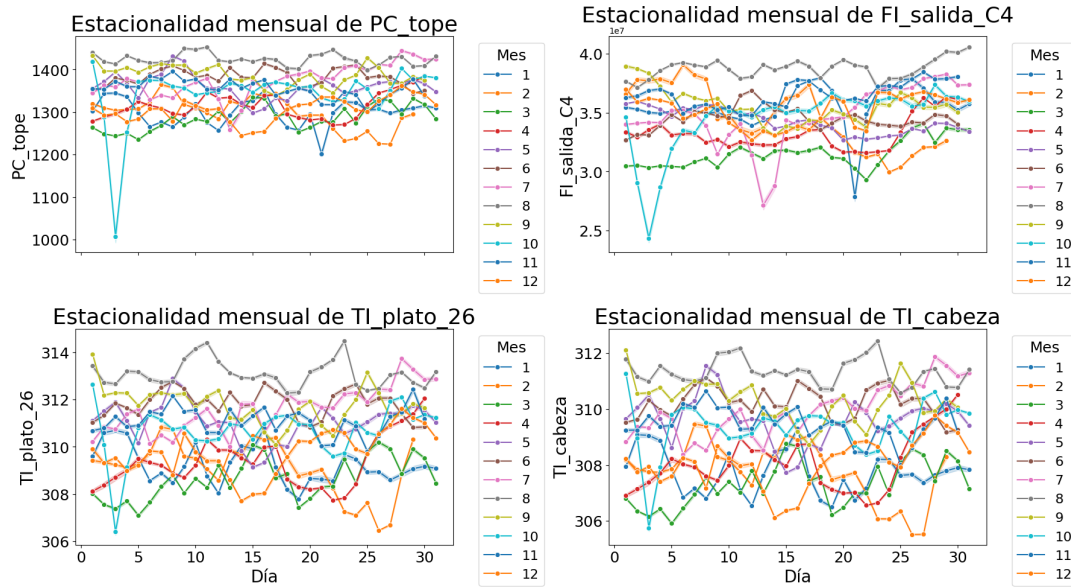


Figura 4.7: Estacionalidad mensual de las variables controladas

En la Figura 4.7, el análisis mensual no refleja ninguna estacionalidad a lo largo de los meses. Hay una alta variabilidad, sin comportamientos repetitivos mes a mes. Aun así, de forma general, se puede intuir lo observado en la estacionalidad anual. Durante los meses de verano, las variables térmicas y de presión alcanzan los valores más altos, y en los primeros y últimos meses del año ocurre lo contrario.

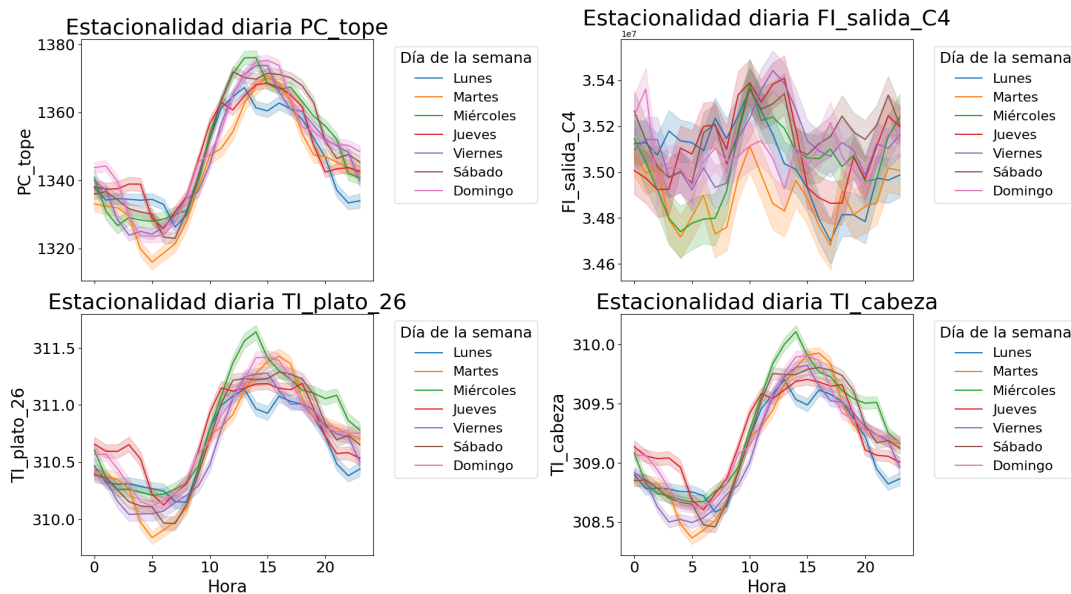


Figura 4.8: Estacionalidad diaria de las variables controladas

Por último, la Figura 4.8 muestra la estacionalidad diaria. Las variables térmicas

## Capítulo 4. Definición del problema

---

cas y la presión contienen un ciclo claro a lo largo del día, alcanzando valores máximos durante el mediodía y mínimos de madrugada. La oscilación corresponderá probablemente a cambios en la operación de la planta, el perfil térmico ambiental o los patrones de carga del sistema. En el caso de la cuarta variable, el caudal de salida de C4, aunque el comportamiento es más irregular, se puede apreciar una ligera tendencia, valores crecientes durante el día y decrecientes por la tarde.

Se confirma así la existencia de estacionalidades a distintas escalas en las series temporales objetivo, especialmente en las térmicas y la presión.

En conjunto, el análisis realizado a lo largo de este capítulo ha permitido comprender mejor el proceso y las características de los datos con los que se quiere trabajar. Identificando la estructura temporal de las variables, su calidad, relaciones internas y patrones generales, se sientan las bases necesarias para afrontar el desarrollo de los modelos predictivos. A partir de este punto, se detallará la preparación de dichos datos para su uso en las diferentes técnicas de aprendizaje automático, y la construcción del *soft sensor*.

## Capítulo 5

# Modelado de la solución

Con las características operativas del proceso comprendidas y los datos disponibles analizados en detalle, se procede al diseño de la solución basada en modelos predictivos. En este capítulo se aborda la preparación del conjunto de datos para las técnicas de aprendizaje automático, así como las herramientas y estrategias empleadas para construir, optimizar y evaluar los modelos.

Se describe todo el preprocesamiento aplicado para la organización de las secuencias, la limpieza, el escalado y la división de los datos. Además, se introducen las librerías y recursos empleados para el desarrollo de los modelos y el ajuste de sus parámetros. Finalmente, se incluyen las métricas y visualizaciones seleccionadas para medir el rendimiento, y la forma en la que se llevará a cabo la evaluación de dichos modelos. Todo ello forma la base sobre la que se construye el sistema de *soft sensor* propuesto.

### 5.1. Preparación de los datos

Antes de poder aplicar cualquiera de los modelos y construir el sistema de predicción, resulta fundamental la preparación de los datos. La limpieza y transformaciones aplicadas en esta fase no solo permiten cumplir con los requisitos de las arquitecturas empleadas, sino que garantizan un entrenamiento sobre una base sólida, coherente y representativa del sistema real. Basándose en el análisis exhaustivo realizado, se abordan los aspectos relacionados con la limpieza, depuración y ajuste del conjunto de datos. Todo ello impacta directamente en el rendimiento final y es imprescindible para obtener unos resultados fiables.

En este trabajo, se ha dividido la tarea en dos bloques principales: la limpieza y transformación de los datos, y su división y escalado.

#### 5.1.1. Preprocesamiento y limpieza

En un primer momento, se contaba con un conjunto de datos que abarcaban desde el 12 de junio de 2019 hasta el 14 de enero de 2025. Como se explicó en el análisis, los registros de los dos primeros años se descartaron desde el

## Capítulo 5. Modelado de la solución

---

inicio, ya que estaban condicionados por situaciones externas concretas y no reflejaban el funcionamiento normal de la planta. A partir de ese momento, todo el análisis se limitó al periodo comprendido entre el 1 de julio de 2021 y el final de los registros, con más de 1.800.000 observaciones.

Dado que el volumen de datos es excesivamente elevado y su procesamiento no es viable a nivel computacional, se ha decidido seleccionar un intervalo más manejable, representativo y reciente. No obstante, esta decisión no se basa únicamente en criterios de tamaño; se busca centrarse en un tramo temporal con un modo de operación estable, consistente y habitual, con el que poder evaluar el comportamiento de los modelos en condiciones controladas. En futuras fases se podría ampliar el enfoque a periodos más amplios con mayor variabilidad.

Analizando más en detalle los registros comprendidos en el último año, se han seleccionado aquellos que contienen menos ruido. Ciertos meses en 2024, aunque presentan variaciones (inevitables dada la naturaleza del proceso), parecen ser más estables. De esta forma, se emplean los datos comprendidos entre el 1 de marzo de 2024 y el 14 de abril de ese mismo año, obteniendo una muestra suficientemente amplia con un total de 64.800 observaciones.

Una vez el intervalo de trabajo ha sido definido, se procede a la limpieza de los datos. El objetivo es dejarlos preparados para el entrenamiento de los modelos, eliminando valores que afecten o interfieran en el rendimiento. Se abordan una serie de problemas: valores nulos, negativos, extremos o atípicos y posibles minutos faltantes.

- Nulos: En el periodo que se va a usar para el entrenamiento y validación de los modelos, no se encuentra ningún valor nulo, lo cual facilita el procesamiento. En las series temporales todos los valores tienen que estar definidos, y en este caso, por la frecuencia de muestreo, cada minuto debe haber un registro. Por ello, eliminar las filas que contuvieran valores nulos, no sería una opción porque se perdería la estructura regular, y deberían ser imputados. Sin embargo, en este caso, no hay que preocuparse por esta situación.
- Valores negativos: Al igual que ocurre con los anteriores, en el mes y medio seleccionado, ninguna de las variables contiene este tipo de valores. De estar presentes, deberían ser tratados de igual forma aplicando un tratamiento de eliminación seguido de una imputación, ya que no tienen sentido físico. No obstante, esto no es necesario puesto que no se da el problema.
- Datos faltantes: No se ha detectado ninguno, no hay instantes que no estén registrados. Esta problemática durante el análisis se observó en los meses de junio, pero al prescindir de esas fechas, no es necesario aplicar ninguna corrección.
- Valores atípicos: Se utiliza el mismo enfoque empleado durante el análisis exploratorio, basado en el rango intercuartílico, para detectar valores extremos. Algunas variables presentan observaciones fuera de rango, pero su número es limitado y están distribuidos de forma dispersa en el intervalo temporal. Dado que son escasos, en comparación con la cantidad total dis-

## 5.1. Preparación de los datos

ponible, se decide eliminarlos e imputarlos. Para mantener la continuidad temporal y que las series mantengan la estructura de un valor por minuto, se aplica una interpolación lineal basada en el tiempo. La técnica resulta adecuada ya que los valores eliminados están rodeados de datos válidos, facilitando así la reconstrucción sin introducir distorsiones relevantes en la dinámica de las señales.

En la siguiente imagen se muestran las series temporales correspondientes a cada una de las variables tras aplicar el proceso de limpieza:

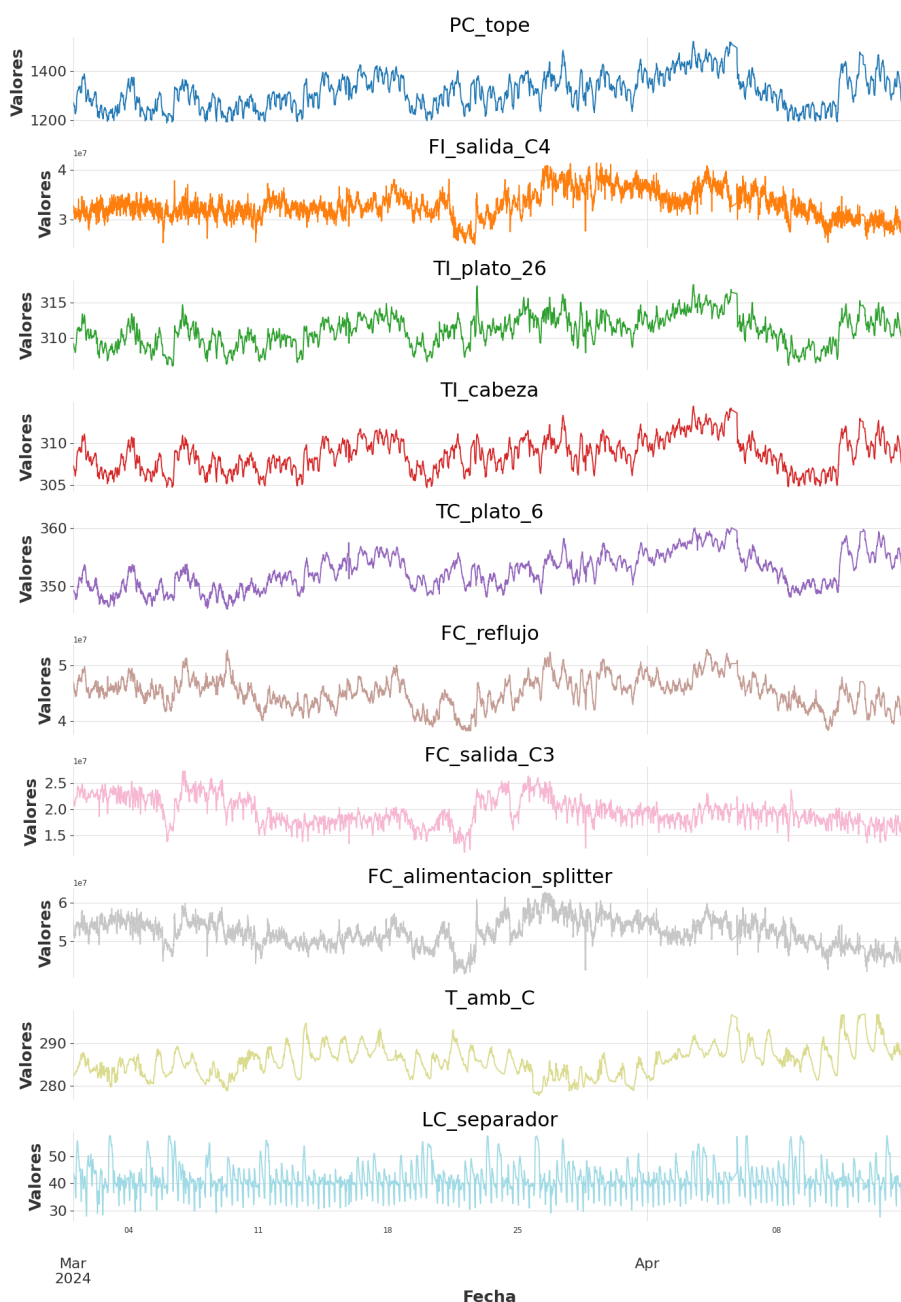


Figura 5.1: Series temporales tras el preprocesado

## Capítulo 5. Modelado de la solución

---

Como resultado del procesamiento aplicado sobre las señales, las series temporales resultantes en la Figura 5.1 muestran comportamientos más estables. Si bien algunas siguen presentando pequeñas oscilaciones, se debe en parte al ajuste de escalas realizado para homogeneizar la visualización. Aun así, al tratarse de un proceso industrial, es natural que existan ciertas fluctuaciones y situaciones ligeramente más inestables. De hecho, si las señales fuesen completamente uniformes, se estaría eliminando la dinámica propia del sistema, lo que supondría perder información significativa para el modelado.

Una vez se dispone del conjunto de datos depurado, se procede a su división y escalado para prepararlos completamente para los modelos de aprendizaje automático. Tales aspectos se abordan a continuación.

### 5.1.2. División y escalado de datos

Con los datos limpios y preparados, la siguiente tarea es organizar y estructurar correctamente las variables para aplicarlas a los modelos seleccionados. Antes de hacer la división en los tres conjuntos de entrenamiento, validación y test, es necesario establecer y distinguir con claridad aquellas variables que se usarán solo como entrada y aquellas que, además, se desea estimar.

Tal como se explicó en el capítulo anterior, las variables disponibles se dividen en tres grupos: de perturbación, manipuladas y controladas. Las dos primeras son aquellas que se utilizarán como entrada al modelo, aportándole información complementaria y contexto sobre el estado. En conjunto, esas seis variables forman lo que se conoce como covariables o variables auxiliares. Por otro lado, las controladas también se incorporan como entrada a través de su historial reciente, pero son el objetivo de la predicción. Forman las cuatro señales que se tratará de estimar a partir de las covariables y de su propio histórico. Cada modelo recibirá valores pasados de todas las variables con el fin de predecir las controladas.

Con la organización establecida, se realiza la **división del conjunto de datos**. Sobre el intervalo total (01/03/2024 – 14/04/2025) se definen tres subconjuntos:

- Entrenamiento: 1 de marzo – 1 de abril (aproximadamente el 70%)
- Validación: 1 – 8 de abril (15%)
- Test: 8 – 15 de abril (15%)

Tras la división, se procede al **escalado de las variables**. Este proceso es fundamental en datasets similares al que se dispone, en los que las variables presentan magnitudes muy diferentes entre sí. El escalado permite homogeneizar los rangos numéricos evitando que parámetros con una amplitud mucho mayor dominen sobre las demás.

En este trabajo, se utiliza una normalización tipo MinMaxScaler. Se transforman los valores de cada una de las variables para que se ajusten a un intervalo dado, (0, 1). De esta forma, se conservan las distribuciones originales, pero

## 5.2. Librerías para el desarrollo de los modelos

comprimiendo los valores al rango específico. Para evitar filtración de datos e información, primero se calculan los parámetros del escalado (mínimo y máximo) sobre el conjunto de entrenamiento, y posteriormente se aplican a los otros dos conjuntos.

A modo ilustrativo, en la Figura 5.2, se muestra el resultado de esta división y escalado sobre la variable 'PC\_tope'.

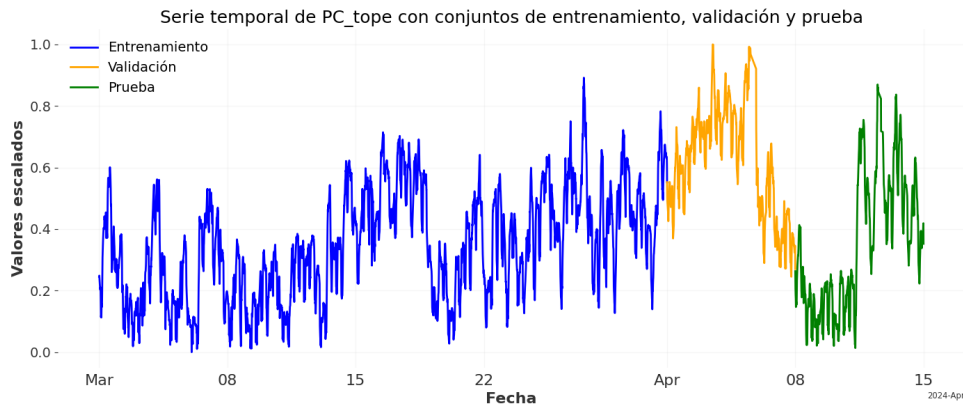


Figura 5.2: Ejemplo de la división en los conjuntos de entrenamiento, validación y test, y escalado de la variable 'PC\_tope'

Como se puede observar, con el escalado, los valores de la variable quedan comprendidos entre 0 y 1. También se aprecia la división de los datos a lo largo del tiempo sin reordenarlos y manteniendo la secuencialidad, lo que permite entrenar y validar los modelos respetando el flujo temporal original.

Con la estructura definida y los datos escalados, ya se dispone de todo lo necesario para poder implementar y desarrollar los modelos de predicción. En los siguientes apartados se introduce la herramienta empleada para llevar a cabo el desarrollo y se describen las métricas y visualizaciones que se usarán para evaluar los modelos.

## 5.2. Librerías para el desarrollo de los modelos

Para la implementación y desarrollo de los modelos predictivos se ha recurrido al uso de herramientas específicas para series temporales. Estas permiten modelar las series de forma flexible y eficiente, gestionando los datos y el proceso de entrenamiento. Además, facilitan tareas clave como la definición de ventanas temporales, el escalado, el entrenamiento supervisado y la evaluación.

En este sentido, la principal herramienta utilizada ha sido la librería **Darts** [29], desarrollada en Python y orientada específicamente al modelado de series temporales. Su principal ventaja se encuentra en la integración de un marco unificado que contiene tanto modelos estadísticos tradicionales como arquitecturas modernas de aprendizaje profundo. Facilita la experimentación y el uso de diferentes enfoques sin necesidad de modificar la lógica del desarrollo. Además,

## Capítulo 5. Modelado de la solución

---

a partir de su estructura de datos propia, *TimeSeries*, ofrece funcionalidades como el escalado, la sincronización temporal entre variables, la incorporación de covariables o la generación de ventanas para el entrenamiento. Todo ello es adaptable a diferentes frecuencias de muestreo y horizontes de predicción, lo que permite ajustarse al problema concreto sin dificultad.

De esta manera, Darts ofrece un entorno de desarrollo flexible y uniforme en el que implementar los modelos de predicción. Su diseño permite comparar de forma homogénea diferentes enfoques y analizar el comportamiento y rendimiento de cada uno de ellos.

### 5.2.1. Optimización de hiperparámetros

Durante el desarrollo de modelos de predicción, un aspecto fundamental es la correcta selección de sus **hiperparámetros**. Se trata de parámetros que no se aprenden durante el entrenamiento, sino que tienen que definirse previamente y afectan a la actuación del modelo. Tienen un impacto en el comportamiento, el aprendizaje, la capacidad de generalización y el rendimiento final. Por ello, es necesario disponer de un método eficaz para ajustarlos.

Con este propósito, en el trabajo se hace uso de **Optuna** [53], una herramienta de optimización automática de hiperparámetros. En lugar de realizar búsquedas aleatorias o por cuadrícula (*grid search*), esta librería utiliza una estrategia basada en optimización bayesiana, adaptando la búsqueda en función de los resultados obtenidos en cada iteración.

Está completamente integrada en Darts, lo que facilita su aplicación sobre los modelos seleccionados, automatizando el proceso de ajuste sin necesidad de hacer pruebas exhaustivas.

## 5.3. Evaluación del modelo

Otro aspecto fundamental es la definición de una estrategia de evaluación adecuada y coherente con la naturaleza del problema, para evaluar su rendimiento. No se debe considerar únicamente el ajuste numérico de las predicciones, sino también la capacidad de generalización en diferentes escenarios y el comportamiento dinámico a lo largo del tiempo.

En este apartado se describen tanto la estrategia seguida para generar las predicciones y estructurar el proceso de evaluación, como las métricas y visualizaciones que permiten analizar los comportamientos en todo el conjunto de prueba.

### 5.3.1. Estrategia de predicción

Con el fin de evaluar de forma consistente la capacidad predictiva de las diferentes arquitecturas, se plantea una estrategia de predicción en dos niveles. Analizar únicamente algunos casos concretos no es suficiente. Puede dar una impresión puntual del comportamiento, pero no permite obtener una valoración

sólida y generalizable. Por tanto, se opta por combinar dos enfoques complementarios: predicciones localizadas en ventanas concretas para realizar un análisis cualitativo, y una evaluación estadística con predicciones más sistemáticas sobre sectores amplios o el *test* completo.

Por una parte, se seleccionan distintas **ventanas temporales de 45 minutos** del conjunto de datos de *test*. El objetivo de las predicciones localizadas en estas ventanas es analizar la actuación del modelo en casos y contextos concretos del proceso. La elección de los 45 minutos corresponde a un equilibrio entre ofrecer una secuencia suficientemente extensa como para validar el desempeño del modelo y mantener un horizonte dentro de la utilidad de la aplicación en este proceso. Estas ventanas permiten observar de forma más detallada la evolución de las predicciones a lo largo del tiempo, mostrando errores, desviaciones o cambios puntuales.

Sin embargo, estas observaciones puntuales no son suficientes para evaluar la consistencia del modelo, ya que solo representan momentos específicos. Por ello, se emplea una segunda estrategia basada en una evaluación sistemática a lo largo de tramos más amplios, pudiendo llegar a evaluar el conjunto de prueba completo. Dado que este conjunto contiene un total de 10.000 registros, no sería viable ni fiable realizar predicciones autorregresivas de largo alcance, puesto que los errores se propagarían y acumularían a cada paso. Además, el sistema no busca anticipar a largo plazo, sino realizar estimaciones fiables a más corto o medio plazo.

Para cubrir estas limitaciones, se definen **horizontes fijos de 15, 30 y 45 minutos** y se recorren múltiples puntos consecutivos, generando predicciones completas para un intervalo temporal más amplio. En cada uno de los puntos, el modelo genera una predicción para el horizonte definido. Por ejemplo, si el horizonte es de 15 minutos, el modelo predice en cada uno de los instantes los próximos 15 valores.

Con todas esas predicciones generadas, se selecciona únicamente el último valor de cada una de ellas, es decir, el valor correspondiente al minuto 15. Así, se obtiene una nueva serie de predicciones, en las que cada una representa la estimación del valor 15 minutos después del punto temporal de cada una de las observaciones del *test*. Esta serie se compara directamente con los valores reales desplazados esos 15 minutos, generando una evaluación específica para ese horizonte temporal.

El mismo procedimiento se sigue para el resto de horizontes, 30 y 45 minutos, permitiendo analizar el rendimiento de los modelos en diferentes plazos. Se evita la acumulación de errores que suele producirse al predecir de forma autorregresiva en intervalos largos, y se obtiene una estimación más estable.

#### 5.3.2. Métricas de evaluación y visualizaciones

Una vez definida la estrategia de predicción, lo siguiente es establecer cómo se va a medir el rendimiento de los modelos. Para ello, se presentan una serie de métricas y visualizaciones que permiten interpretar el comportamiento y analizar

## Capítulo 5. Modelado de la solución

---

la calidad de las estimaciones de cada arquitectura.

El primer método empleado son las **métricas de evaluación**, que ofrecen una medida cuantitativa del error cometido en las predicciones. En el estudio se emplean las siguientes:

- **Error absoluto medio (MAE)**: mide el error absoluto medio entre los valores estimados y los reales observados. Refleja la magnitud media del error sin penalizar de forma excesiva los valores extremos [54]. Su fórmula es la siguiente:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - Y_i| \quad (5.1)$$

Donde:

- $n$ : número de muestras
  - $y_i$ : valor observado
  - $Y_i$ : valor predicho
  - $|y_i - Y_i|$ : error absoluto entre los valores predichos y los observados
- **Error absoluto porcentual medio (MAPE)**: calcula el error en términos relativos, no absolutos, expresándolo en un porcentaje. Proporciona una visión del error proporcional respecto al valor real, siendo independiente de las escalas [54].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \frac{|y_i - Y_i|}{y_i} \quad (5.2)$$

Donde:

- $n$ : número de muestras
  - $y_i$ : valor observado
  - $Y_i$ : valor predicho
  - $\frac{|y_i - Y_i|}{y_i}$ : error absoluto relativo, que indica la desviación proporcional de la predicción respecto del valor real
- **Error cuadrático medio (MSE)**: mide la media de los errores al cuadrado, es decir, la diferencia entre los valores reales y estimados. Al elevar al cuadrado, penaliza los errores grandes. Un valor cercano a 0 significa que el modelo predice correctamente los valores esperados [55].

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - Y_i)^2 \quad (5.3)$$

Donde:

- $n$ : número de muestras
- $y_i$ : valor observado
- $Y_i$ : valor predicho

- **Raíz del error cuadrático medio (RMSE):** se trata de la raíz cuadrada del error cuadrático medio. Cuantifica al igual que el anterior, la diferencia media entre los valores reales y las predicciones [54]. Para calcularlo, se aplica la siguiente fórmula:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - Y_i)^2} \quad (5.4)$$

Donde:

- $n$ : número de muestras
  - $y_i$ : valor observado
  - $Y_i$ : valor predicho
- **Coefficiente de determinación ( $R^2$ ):** mide la variabilidad de la variable objetivo explicada por el modelo. Un valor cercano a 1 indica un buen ajuste, mientras que un valor más cerca del 0 refleja una baja capacidad predictiva del modelo [54].

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (y_i - \bar{Y})^2} \quad (5.5)$$

Donde:

- $n$ : número de muestras
- $y_i$ : valor observado
- $\hat{Y}_i$ : valor predicho
- $\bar{Y}$ : media de todas las observaciones
- $\sum_{i=1}^n (y_i - \hat{Y}_i)^2$ : suma de los errores cuadrados de predicción
- $\sum_{i=1}^n (y_i - \bar{Y})^2$ : suma de las desviaciones cuadradas de los valores observados respecto a su media

Por otro lado, las **visualizaciones complementarias** permiten tener una perspectiva más intuitiva, facilitando la identificación de patrones, errores o desviaciones. Para realizar un análisis cualitativo de los resultados se utilizan **gráficas temporales**, que comparan visualmente los valores reales frente a las predicciones del modelo a lo largo del tiempo. Permiten visualizar la evolución de las variables, facilitando detectar si el modelo capta la tendencia general. Se emplean dos tipos de visualizaciones, ambas gráficas temporales, en línea con las estrategias de predicción.

- Una representación global de un tramo más extenso del conjunto de *test*, para evaluar la consistencia de los modelos a diferentes horizontes definidos.
- Una ventana puntual de 45 minutos, seleccionada como caso concreto para observar con mayor detalle la diferencia entre predicción y realidad.

## **Capítulo 5. Modelado de la solución**

---

Gracias al uso de todas estas herramientas, tanto visuales como numéricas, es posible hacer un análisis detallado del comportamiento de los modelos, desde su rendimiento hasta limitaciones que puedan presentar.

Con todo lo desarrollado en este capítulo, se dispone del entorno completo para construir el sensor mediante el entrenamiento y evaluación de los modelos propuestos. A partir de este punto, el estudio se centra en la aplicación de cada uno de ellos al problema concreto definido, incluyendo el ajuste de sus hiperparámetros y el análisis de su comportamiento. El siguiente capítulo recoge los resultados obtenidos para las diferentes arquitecturas analizadas, tanto en términos cuantitativos como cualitativos, permitiendo valorar el enfoque más adecuado.

## Capítulo 6

# Resultados

Una vez se ha definido y realizado el preprocesamiento de los datos que se van a emplear, establecidas las herramientas para el desarrollo y las métricas de evaluación, se muestra la implementación de los diferentes modelos con los resultados obtenidos tras su entrenamiento y validación. Todos ellos han sido evaluados usando los mismos criterios, explicados en el capítulo anterior, y sobre el mismo conjunto de datos, lo que permite hacer una comparación objetiva entre los diferentes enfoques.

Para cada arquitectura se detallan los hiperparámetros seleccionados y ajustados, junto con los valores obtenidos en las métricas de evaluación definidas y las visualizaciones que permiten analizar gráficamente su comportamiento. Finalmente, se incluye una conclusión individual sobre el rendimiento de cada arquitectura y una comparativa global que permite identificar el enfoque que más se adecúa al problema planteado.

### 6.1. Configuración común de los modelos

Antes de presentar los resultados, es necesario explicar una serie de hiperparámetros que son comunes a todas las arquitecturas empleadas, a excepción de las *baseline*. Se trata de un problema de series temporales, por lo que es fundamental la forma en la que se estructura la entrada y salida de los modelos. Por este motivo, se describen al comienzo de forma general.

- **'Input\_chunk\_length'**: define la cantidad de valores pasados que recibe el modelo como entrada para hacer la predicción. Aunque este valor es adaptado y ajustado a cada modelo, en todos los casos se parte de una configuración base con el mismo número de pasos históricos. Tras realizar un estudio preliminar sobre una porción pequeña del conjunto, se llegó a la conclusión que 25 era el número de pasos óptimos como punto de partida.

En los modelos basados en árboles (RF y XGBoost) este parámetro se define con un nombre distinto, pero su función es equivalente. En lugar de 'input\_chunk\_length' se emplea 'lags' para el historial de las variables ob-

jetivo y 'lags\_past\_covariates' para las covariables o variables auxiliares. En ambos casos se usa también el valor 25 como inicial.

- '**Output\_chunk\_length**': representa cuántos valores se predicen simultáneamente en cada iteración. En este caso, el valor se ha fijado siempre a 1 ya que se busca que las predicciones se hagan punto a punto mediante un enfoque autorregresivo. Es decir, el modelo genera un único valor como estimación que se utiliza de entrada para la siguiente estimación, y así, sucesivamente, como se explica en la Sección 3.3.2 (Predicción autorregresiva).

Además, cabe destacar que para todos los modelos de redes neuronales se parte y emplean otros hiperparámetros comunes. En todos los casos se utiliza como función de pérdida el Error Cuadrático Medio (MSE), y como optimizador el Adam con una tasa de aprendizaje o *learning rate* fija de 0,001. Estas configuraciones han sido ampliamente validadas en tareas de regresión en el ámbito del Deep Learning y se consideran robustas y efectivas para esta aplicación.

Estos parámetros conforman la estructura básica de entrada y salida de los modelos y estarán presentes a lo largo de todo el capítulo.

### 6.2. Modelos *baseline* y Random Forest

Como base y referencia para comparar el resto de las arquitecturas más complejas, se incluyen una serie de modelos sencillos, conocidos como *baseline*. Inicialmente se parte de tres variantes distintas: **NaiveCopy**, **NaiveMean** y **NaiveDrift**. No disponen de hiperparámetros ajustables, aunque al igual que el resto de las arquitecturas requieren de un proceso de entrenamiento en el que aprenden únicamente a partir de la evolución de las propias variables objetivo.

Junto a estos tres modelos iniciales se incorpora otro que, aunque entraña una mayor complejidad, es considerado también como punto de partida para el desarrollo del resto: **Random forest (RF)**. A pesar de que el modelo introduce más capacidad de aprendizaje respecto a los *baseline*, su rendimiento ha sido limitado, especialmente en horizontes amplios, lo que justifica su agrupación en este bloque como referencia inicial.

RF, a diferencia del resto, si emplea información y contexto de las variables auxiliares para realizar las predicciones. Contiene una serie de hiperparámetros principales que impactan directamente en su rendimiento. Además de los datos históricos de cada variable, explicados en el apartado anterior, se encuentran los siguientes:

- '**n\_estimators**': número total de árboles empleados. Se parte de una configuración con 100 estimadores.
- '**max\_depth**': profundidad máxima de cada uno de los árboles. Se establece 15 como valor base con el fin de evitar un posible sobreajuste.

Durante las primeras pruebas, estos cuatro modelos obtuvieron resultados poco prometedores. Realizando predicciones a lo largo del conjunto de prueba, con

## 6.2. Modelos *baseline* y Random Forest

los tres horizontes definidos (15, 30, 45 minutos), se observó que, en general, tienden a copiar y seguir el último valor conocido a lo largo de sus estimaciones.

A continuación, se incluye una imagen que muestra este hecho. Corresponde a un tramo del conjunto de *test*, con las predicciones en los diferentes horizontes para el modelo RF (el resto obtienen resultados similares, aunque menos ruidosos).

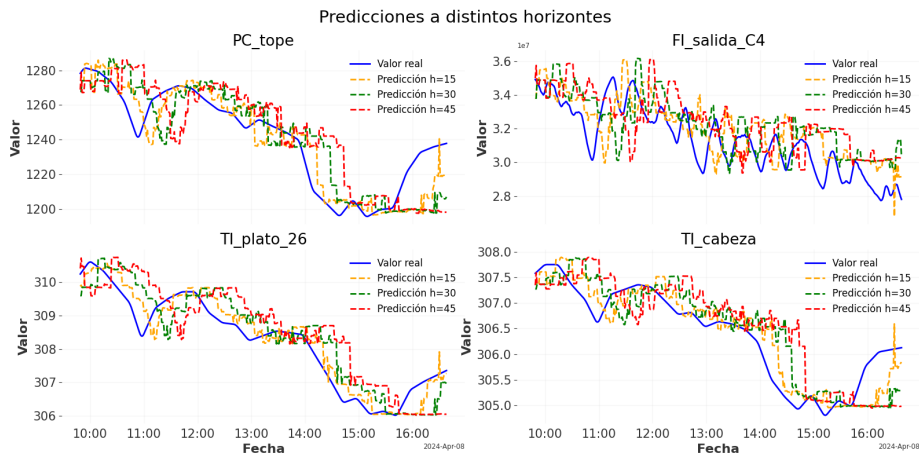


Figura 6.1: Predicciones de RF en tres horizontes

Se analizaron predicciones puntuales en ventanas de 45 minutos, donde se pueden observar estas claras limitaciones en todos los modelos. En la Figura 6.2 se puede apreciar cómo NaiveCopy repite el último valor observado, NaiveMean calculando la media se desvía de la serie, NaiveDrift sigue la misma trayectoria que el primero, y RF apenas mejora los resultados anteriores.

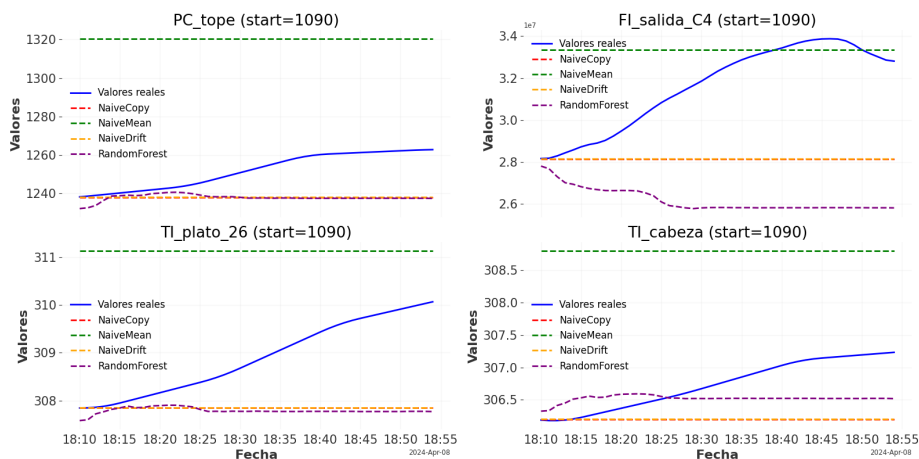


Figura 6.2: Predicción puntual de modelos *baseline* y RF sobre una ventana de 45 minutos

El rendimiento de estos cuatro modelos disminuye conforme se amplía el horizonte de predicción. Su capacidad de adaptación y generalización es muy li-

## Capítulo 6. Resultados

mitada, lo que hace evidente la necesidad de aplicar modelos más avanzados, capaces de extraer información de este tipo de datos complejos. Dada la simplicidad y las limitaciones que presentan, no se ha considerado realizar ajustes o pruebas adicionales, ya que no se espera una mejora significativa en los resultados. Aun así, en conjunto, todos ellos establecen una línea base sobre la que analizar las mejoras alcanzadas por el resto de las arquitecturas en los siguientes apartados.

Para poder llevar a cabo esta comparativa con la que medir los avances o empeoras de cada uno de los modelos frente a estos, se consideran como referencia las métricas incluidas en la Tabla 6.1. Corresponden a los resultados obtenidos por el modelo NaiveCopy sobre el conjunto de *test* completo para los tres horizontes de predicción definidos (15, 30, 45 minutos). En cada una de las variables objetivo se muestran las cinco métricas de evaluación descritas. Se emplea únicamente este modelo, al ser el más representativo a nivel base.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_tope	15 min	6,01	0,46%	76,27	8,73	0,98
	30 min	10,89	0,83%	234,25	15,31	0,95
	45 min	15,47	1,18%	437,18	20,91	0,92
FI_salida_C4	15 min	7,89x10 <sup>5</sup>	2,59%	1,04x10 <sup>12</sup>	1,02x10 <sup>6</sup>	0,65
	30 min	8,02x10 <sup>5</sup>	2,64%	1,17x10 <sup>12</sup>	1,08x10 <sup>6</sup>	0,60
	45 min	8,95x10 <sup>5</sup>	2,94%	1,39x10 <sup>12</sup>	1,18x10 <sup>6</sup>	0,53
TI_plato_26	15 min	0,22	0,07%	0,09	0,29	0,98
	30 min	0,41	0,13%	0,28	0,53	0,94
	45 min	0,56	0,18%	0,52	0,72	0,90
TI_cabeza	15 min	0,19	0,06%	0,05	0,23	0,98
	30 min	0,30	0,10%	0,17	0,41	0,96
	45 min	0,43	0,14%	0,32	0,57	0,93

Tabla 6.1: Métricas de evaluación de NaiveCopy

### 6.3. XGBoost

Tras establecer como primera referencia los anteriores modelos, se incorpora XGBoost. Comparte con RF la base de árboles de decisión, pero introduce mejoras con el entrenamiento secuencial, lo que permite capturar relaciones más complejas dentro de la serie temporal. Contiene exactamente los mismos hiperparámetros que RF: 'lags', 'lags\_past\_covariates', 'n\_estimators' y 'max\_depth', por lo que se parte de la misma configuración inicial para poder establecer una comparación objetiva entre sus rendimientos.

De esta forma, los 25 pasos históricos como entrada y los 100 estimadores con una profundidad máxima de 15, se definen como punto de partida para el entrenamiento. A partir de esta base, se lleva a cabo un proceso de optimización automática con Optuna para ajustar los hiperparámetros y obtener mejores re-

sultados maximizando el rendimiento. Se construye un nuevo modelo con las siguientes características:

- La ventana de entrada se mantiene igual, tanto para las variables auxiliares, covariables, como para las objetivo. Aunque para esta arquitectura concreta se podrían considerar tamaños de entrada distintos para cada grupo de variables, se ha decidido mantener la misma longitud de ventana para ambos. La razón corresponde, por un lado, a la lógica del problema, ya que el modelo debe disponer del mismo contexto temporal para todas las señales, y por otro, a la coherencia con el resto de arquitecturas.
- El número de estimadores, es decir árboles, se disminuye a 94, con el fin de reducir ligeramente el ruido que produce el modelo de lo aprendido.

La Tabla 6.2 presenta las métricas obtenidas por este último modelo sobre el conjunto completo de *test* para los tres horizontes de predicción. En cada una de las variables objetivo se incluyen las cinco métricas de evaluación descritas, acompañadas de la diferencia obtenida respecto al modelo base con la configuración inicial. Esta se representa en color verde cuando el ajuste ha supuesto una mejora, y en rojo en caso contrario. De tal forma, se facilita visualmente la comparación, pudiendo identificar el impacto del proceso de optimización.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_tope	15 min	6,03 (-1,28)	0,45% (-0,09%)	76,51 (-51,74)	8,75 (-2,58)	0,98 (+0,00)
	30 min	10,64 (-1,71)	0,81% (-0,12%)	222,73 (-94,86)	14,92 (-2,90)	0,96 (+0,01)
	45 min	14,87 (-2,10)	1,13% (-0,15%)	410,32 (-158,29)	20,26 (-3,59)	0,92 (+0,02)
FI_salida_C4	15 min	7,19x10 <sup>5</sup> (+1,52x10 <sup>4</sup> )	2,38% (+0,05%)	8,29 x10 <sup>11</sup> (+2,6 x10 <sup>10</sup> )	9,11 x10 <sup>5</sup> (+1,5x10 <sup>4</sup> )	0,72 (+0,00)
	30 min	7,56 x10 <sup>5</sup> (-4,4 x10 <sup>3</sup> )	2,50% (-0,01%)	9,36 x10 <sup>11</sup> (-2,3 x10 <sup>10</sup> )	9,68 x10 <sup>5</sup> (-1,2 x10 <sup>4</sup> )	0,69 (+0,00)
	45 min	8,23 x10 <sup>5</sup> (-1,45 x10 <sup>4</sup> )	2,73% (-0,04%)	1,08 x10 <sup>12</sup> (-8,9 x10 <sup>10</sup> )	1,04 x10 <sup>6</sup> (-4,2x10 <sup>4</sup> )	0,64 (+0,03)
TI_plato_26	15 min	0,18 (-0,00)	0,06% (-0,00%)	0,06 (-0,00)	0,24 (-0,01)	0,98 (+0,00)
	30 min	0,35 (-0,01)	0,11% (-0,01%)	0,22 (-0,03)	0,47 (-0,03)	0,95 (+0,00)
	45 min	0,50 (-0,01)	0,16% (-0,01%)	0,45 (-0,03)	0,67 (-0,02)	0,91 (+0,00)
TI_cabeza	15 min	0,15 (-0,01)	0,05% (-0,00%)	0,05 (-0,01)	0,22 (-0,01)	0,98 (+0,00)
	30 min	0,27 (-0,02)	0,09% (-0,01%)	0,14 (-0,02)	0,37 (-0,03)	0,97 (+0,00)
	45 min	0,39 (-0,02)	0,13% (-0,01%)	0,27 (-0,04)	0,52 (-0,04)	0,94 (+0,00)

Tabla 6.2: Comparación de métricas entre XGBoost optimizado y su configuración inicial

## Capítulo 6. Resultados

Es importante señalar que, aunque las métricas obtenidas reflejen valores bastante buenos, con errores bajos y  $R^2$  muy cercanos a 1, hay que interpretarlos con cuidado. Si las predicciones fueran erróneas no se alcanzarían valores tan positivos. Sin embargo, hay que tener en cuenta que se calculan sobre el *test* completo, lo que suaviza errores al tratarse de 10.000 puntos. Además, la alta correlación temporal de estas series suele producir valores de  $R^2$  altos. Es por todo esto por lo que también se deben considerar las visualizaciones complementarias a la hora de evaluar modelos en problemas como el dado.

En la Figura 6.3 se muestran las predicciones hechas en los tres horizontes por el mejor de los modelos. Se selecciona únicamente un tramo concreto de 500 puntos (poco más de 8 horas) del primer día de *test*, en lugar del conjunto completo, para permitir una mejor interpretación y poder ver de forma más concreta su actuación. En ella se aprecia que el modelo no consigue anticiparse por completo al comportamiento de las variables, sino que tiende a copiar los valores recientes observados. Esto genera un desplazamiento en la señal predicha en función del horizonte empleado. Una excepción es la variable 'FI\_salida\_C4' en la que no parece haber un desfase temporal tan marcado. Al igual que ocurría con el RF, la predicción sigue la forma general de la curva, pero con cierto ruido y variabilidad añadida.

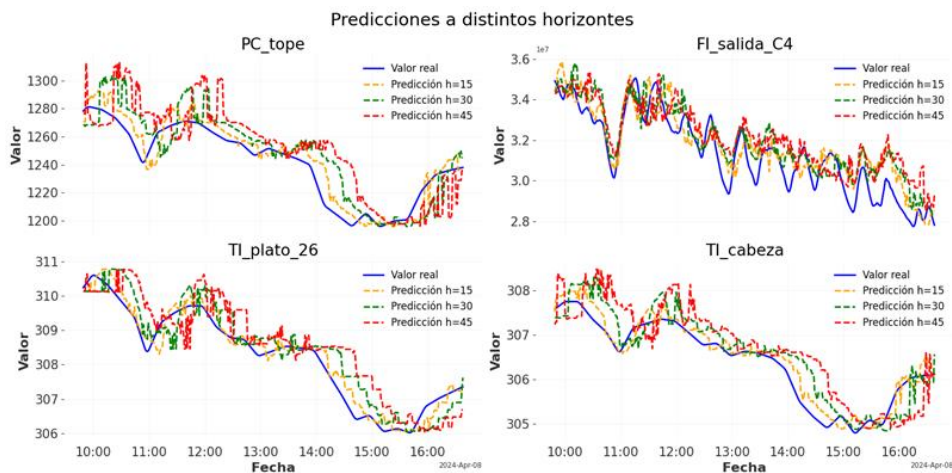


Figura 6.3: Predicciones de XGBoost en tres horizontes

Se incluye una figura adicional con una predicción de 45 minutos, seleccionada como ejemplo específico para observar en detalle la alineación entre las estimaciones y los valores reales a lo largo del tiempo. Se trata de un periodo de tiempo específico, cuyas predicciones obtenidas se muestran también en el resto de los modelos para poder establecer una comparación. Sin embargo, con el fin de poder analizar otros momentos, se facilitan los resultados de otras ventanas temporales en el Anexo A.

En este caso, en la Figura 6.4 se ve con claridad la limitación del modelo, tiende a generar estimaciones poco dinámicas que no capturan la evolución de las señales. Aunque en las tres primeras variables el modelo logra imitar parcialmente

la tendencia ascendente, no consigue alcanzar los valores reales, mostrando una separación notable entre ellos. En el caso de 'TI\_cabeza', la predicción se queda plana, replicando el último valor aprendido.

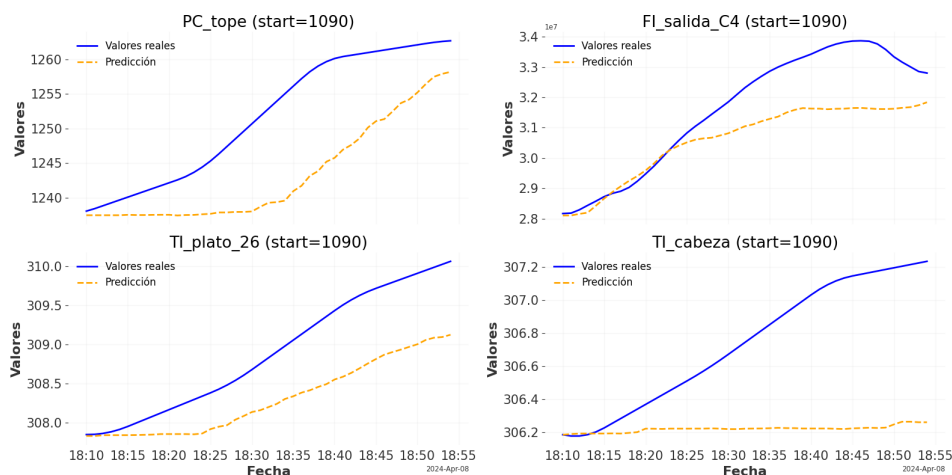


Figura 6.4: Predicción puntual de XGBoost sobre una ventana de 45 minutos

Aunque en un primer momento se pensó que XGBoost podría ofrecer mejores resultados que los modelos ya vistos, motivo por el que se incluye en una sección aparte, finalmente presenta limitaciones similares. No tiene capacidad suficiente para capturar las dinámicas propias del sistema. Las métricas pueden parecer razonables, pero esto se debe, en parte, a que como el modelo está tendiendo a copiar el último valor aprendido en cada momento, la diferencia con el valor actual es reducida. Al tratarse de horizontes no excesivamente amplios, sobre todo en un conjunto de prueba como el que se está tratando con 10.000 puntos, el error aparente es bajo. No obstante, esto no implica que la predicción sea útil, si el modelo no anticipa los cambios, su aplicación práctica es limitada.

En comparación con RF, presenta una ligera mejora, con predicciones más suaves. En algunos casos concretos, como en la Figura 6.4, el modelo logra captar la tendencia de subida en casi todas las variables, aunque con un claro error. Aun así, este comportamiento no es el adecuado para un *soft sensor* como el caso de estudio. Su rendimiento real sigue sin ser el esperado, lo que refleja la limitación de aplicar modelos de este tipo. Los algoritmos basados en árboles de decisión pueden resultar útiles en problemas de clasificación o regresión estática, pero no parecen estar diseñados para trabajar con series temporales.

## 6.4. LSTM

Como primer modelo de Deep Learning, se ha evaluado la red LSTM. Se trata de una arquitectura especialmente diseñada para trabajar con secuencias temporales por su capacidad para mantener información a lo largo del tiempo. Además de los parámetros comunes ('input\_chunk\_length' y 'output\_chunk\_length'), el modelo incluye también los siguientes (se añade junto a ellos la configuración

## Capítulo 6. Resultados

inicial de partida seleccionada):

- 'hidden\_dim': número de unidades ocultas por capa (se parte de 25, para una representación básica en la primera fase).
- 'n\_rnn\_layers': número de capas recurrentes (1, manteniendo una red sencilla con la que evaluar su comportamiento elemental).
- 'batch\_size': tamaño de cada lote de entrenamiento (1024, tamaño considerable dada la cantidad de datos con la que se va a trabajar).
- 'n\_epochs': número total de épocas de entrenamiento (60).
- 'dropout': proporción de neuronas desactivadas durante el entrenamiento (0.0).

Esta configuración se ha escogido como punto de partida para no formar una red muy profunda y compleja desde el inicio. A partir de ella, se lleva a cabo el proceso de optimización, con el objetivo de mejorar el rendimiento general, sin caer en sobreajuste. El modelo resultante incorpora una serie de ajustes relevantes:

- 'input\_chunk\_length' aumenta a 48, por lo que el modelo parece necesitar y mejorar cuando accede a un mayor historial temporal.
- 'n\_rnn\_layers': 2 y 'hidden\_dim': 197, ampliando el tamaño y capacidad del modelo para representar patrones más complejos.
- 'dropoup': 0,07, añadiendo una pequeña regularización.
- 'batch\_size' se mantiene igual y 'n\_epochs' se amplía a 80 favoreciendo un aprendizaje más completo para un modelo más complejo.

La Tabla 6.3 recoge los resultados obtenidos por este modelo optimizado en los tres horizontes de predicción (15, 30 y 45 minutos). Al igual que en el apartado anterior, se incluyen las cinco métricas para cada variable objetivo, acompañadas de la diferencia respecto al modelo base.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
<b>PC_tope</b>	15 min	5,59 (-4,04)	0,42% (-0,30%)	48,98 (-94,65)	6,99 (-4,99)	0,99 (+0,02)
	30 min	10,59 (-4,53)	0,81% (-0,32%)	172,63 (-185,31)	13,14 (-5,78)	0,97 (+0,03)
	45 min	15,19 (-6,18)	1,16% (-0,44%)	351,77 (-358,63)	18,76 (-7,89)	0,94 (+0,06)
<b>FI_salida_C4</b>	15 min	7,70x10 <sup>5</sup> (-1,04x10 <sup>5</sup> )	2,57% (-0,34%)	9,19x10 <sup>11</sup> (-1.91x10 <sup>5</sup> )	9,59x10 <sup>5</sup> (-9,54x10 <sup>5</sup> )	0,69 (+0,06)
	30 min	1,04x10 <sup>6</sup> (-2,11x10 <sup>5</sup> )	3,50% (-0,69%)	1,75x10 <sup>12</sup> (-5,65x10 <sup>5</sup> )	1,32x10 <sup>6</sup> (-1,97x10 <sup>5</sup> )	0,41 (+0,19)

	45 min	1,17x10 <sup>6</sup> (-3,25x10 <sup>5</sup> )	3,91% (-1,09%)	2,22x10 <sup>12</sup> (-1,03x10 <sup>6</sup> )	1,49x10 <sup>6</sup> (-3,12x10 <sup>5</sup> )	0,26 (+0,33)
TI_plato_26	15 min	0,22 (-0,14)	0,07% (-0,04%)	0,08 (-0,10)	0,28 (-0,14)	0,98 (+0,02)
	30 min	0,42 (-0,22)	0,14% (-0,07%)	0,28 (-0,31)	0,53 (-0,24)	0,94 (+0,06)
	45 min	0,60 (-0,26)	0,19% (-0,08%)	0,56 (-0,50)	0,75 (-0,28)	0,89 (+0,09)
TI_cabeza	15 min	0,15 (+0,01)	0,05% (+0,00%)	0,04 (+0,00)	0,19 (+0,01)	0,99 (+0,00)
	30 min	0,27 (+0,01)	0,09% (+0,00%)	0,12 (+0,01)	0,34 (+0,01)	0,97 (+0,00)
	45 min	0,40 (+0,00)	0,13% (+0,00%)	0,25 (+0,00)	0,50 (+0,00)	0,94 (+0,00)

Tabla 6.3: Comparación de métricas entre LSTM optimizado y su configuración inicial

Junto con la tabla, se añade la siguiente imagen, que contiene estas predicciones exclusivamente para el modelo que obtuvo un mejor resultado. Se selecciona únicamente el mismo tramo de 500 puntos del primer día del grupo de datos de test que se mostró en Figura 6.3 (apartado 6.3 XGBoost), y que se mostrará para el resto de los modelos. Como se puede observar en ella, en los horizontes más cortos, especialmente  $h=15$ , la red logra seguir con bastante precisión la evolución de los datos reales. A medida que se incrementa, la precisión se reduce aumentando los errores, visible también en las métricas.

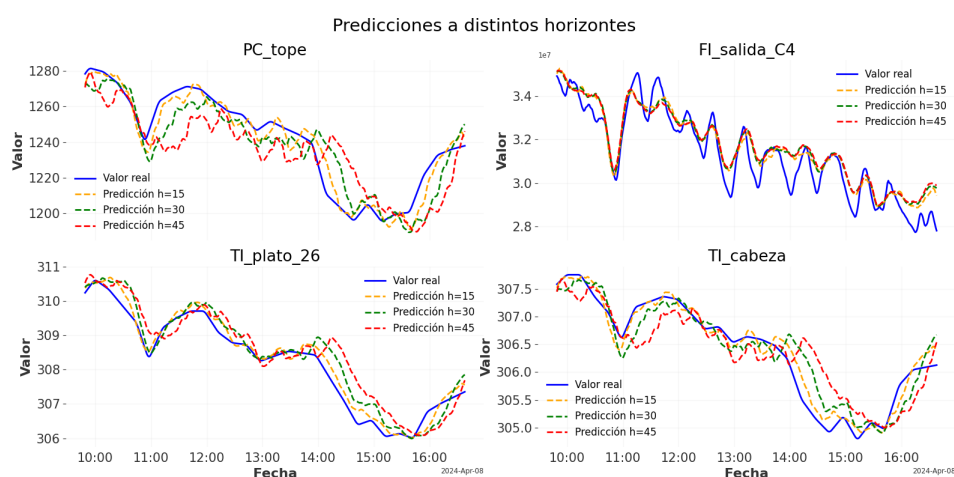


Figura 6.5: Predicciones de LSTM en tres horizontes

De forma complementaria, se incluye una predicción puntual sobre la ventana de 45 minutos como ejemplo concreto para observar cómo evolucionan las estimaciones frente al valor real con mayor detalle. Otras ventanas adicionales se

## Capítulo 6. Resultados

incorporan en el Anexo A.

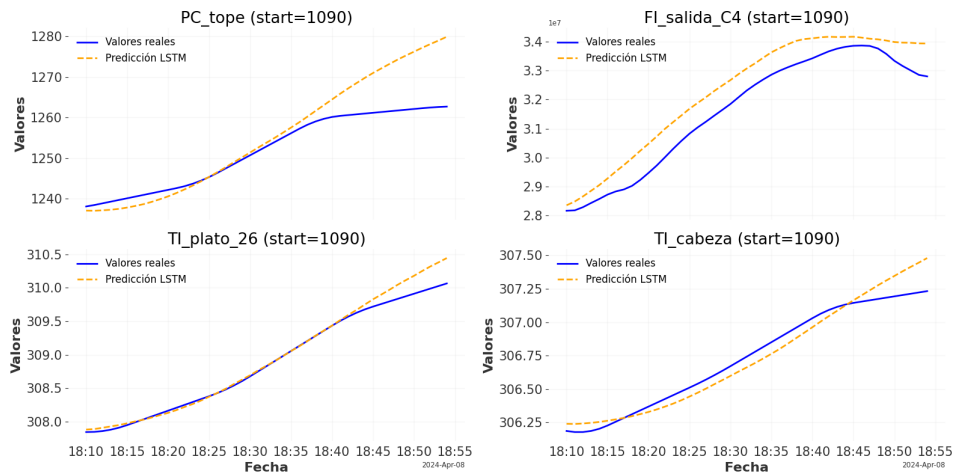


Figura 6.6: Predicción puntual de LSTM sobre una ventana de 45 minutos

En general, los resultados reflejan una mejora clara tras la optimización del modelo, especialmente en horizontes más largos, donde se consigue reducir en mayor cantidad el error cometido. La red muestra una gran capacidad para ajustarse a las dinámicas de las series, manteniendo su tendencia y capturando parte de la variabilidad.

Aun así, al igual que los modelos anteriores, el rendimiento parece degradarse a medida que el horizonte de predicción aumenta. Este deterioro es más notable en la variable 'FI\_salida\_C4', en la cual las métricas indican una pérdida más marcada de precisión conforme aumentan los minutos de predicción. Aunque visualmente en la Figura 6.5 pueda parecer que el modelo ajusta bastante bien, ya que las predicciones no se desplazan tanto en el tiempo, la realidad es que no consigue anticipar con precisión los picos y valles de la serie. Esto se debe a la influencia de un controlador situado en la parte inferior de la columna, cuya acción afecta a la evolución de esta variable y no está completamente reflejado en la entrada del modelo.

En cambio, en las otras tres variables, se mantiene una mayor coherencia. Aunque se observa una cierta degradación en la precisión temporal a medida que aumenta el horizonte, el modelo conserva correctamente la tendencia general de las señales, en particular, las transiciones de subida y bajada. En tramos de predicción más reducidos, el modelo sí consigue ajustar tanto el momento en el que se producen los cambios como la dirección que siguen, con un comportamiento mucho más preciso.

En la segunda imagen (Figura 6.6) se puede observar toda esta tendencia de forma clara: en los primeros minutos, la red sigue con exactitud la realidad y las predicciones se ajustan correctamente, pero a medida que avanza el tiempo, las diferencias entre los valores reales y las estimaciones se van acumulando. A pesar de ello, el rendimiento del modelo en general es sólido y mejora considerablemente respecto a las arquitecturas más básicas presentadas anteriormente.

## 6.5. TCN

La arquitectura TCN, basada en redes convolucionales, es otra de las propuestas generales para el problema de predicción de series temporales. Al igual que el resto de los modelos, se mantiene la misma estructura inicial de entrada y salida definida. Se le añaden los siguientes hiperparámetros, con la configuración base detallada a continuación:

- 'kernel\_size': tamaño del filtro de las capas convolucionales (3).
- 'num\_filters': número de filtros por capa (30, cantidad inicial suficiente como para captar patrones principales).
- 'num\_layers': número de capas convolucionales (1, partiendo de una arquitectura sencilla para evaluar el rendimiento básico del modelo).
- 'dilatation\_base': factor de dilatación que controla la separación entre los valores considerados por el filtro (2).
- 'weight\_form': indica si se aplica normalización de pesos en la capa de entrenamiento (falso, es decir, no se aplica, con el objetivo de no introducir restricciones adicionales en esta primera fase).
- 'dropout': 0.0
- 'batch\_size': 1024
- 'n\_epochs': 60

Respecto a los parámetros comunes con otras arquitecturas, se decide partir del mismo valor inicial para luego adaptarlo a este caso según corresponda, y comparar con un punto de partida común. Así, con esta base, se aplica, mediante Optuna, un ajuste de todos los parámetros presentados. Este proceso da lugar a la siguiente configuración final, ligeramente distinta y más compleja que la de partida:

- 'input\_chunk\_length': 58 pasos temporales de entrada. El modelo optimizado se beneficia de una mayor ventana de contexto histórico, captando patrones de mayor duración.
- 'num\_layers': 2, 'num\_filters': 122 y 'kernel\_size': 4, creando una red con una mayor capacidad y tamaño.
- 'dilatation\_base': 4, ampliando el campo temporal efectivo.
- 'dropout': 0,048 y 'weight\_form' activado, lo que permite aplicar una ligera regularización.
- 'batch\_size' se mantiene igual, y 'num\_epochs' se aumenta en 20, para extender el tiempo de entrenamiento y compensar la complejidad del nuevo modelo.

Los resultados obtenidos para la arquitectura, tanto en su versión inicial como optimizada, se exponen en la Tabla 6.4, recogiendo los tres horizontes de

## Capítulo 6. Resultados

predicción establecidos.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_tope	15 min	7,53 (-3,93)	0,57% (-0,29%)	90,79 (-87,32)	9,53 (-3,82)	0,98 (+0,01)
	30 min	15,02 (-2,73)	1,16% (-0,19%)	356,79 (-84,03)	18,89 (-2,11)	0,94 (+0,01)
	45 min	22,43 (-1,74)	1,74% (-0,10%)	791,68 (-47,33)	28,14 (-0,83)	0,86 (+0,01)
FI_salida_C4	15 min	8,24x10 <sup>5</sup> (+1,50x10 <sup>5</sup> )	2,75% (+0,53%)	1,03 x10 <sup>12</sup> (+3,10x10 <sup>11</sup> )	1,02 x10 <sup>5</sup> (+1,66x10 <sup>5</sup> )	0,66 (-0,10)
	30 min	1,02 x10 <sup>6</sup> (+3,70x10 <sup>5</sup> )	3,41% (+1,27%)	1,54 x10 <sup>12</sup> (+8,28x10 <sup>11</sup> )	1,24 x10 <sup>6</sup> (+3,97x10 <sup>5</sup> )	0,49 (-0,27)
	45 min	1,13 x10 <sup>6</sup> (+4,50x10 <sup>5</sup> )	3,80% (+1,54%)	1,91 x10 <sup>12</sup> (+1,13x10 <sup>12</sup> )	1,38 x10 <sup>6</sup> (+4,99x10 <sup>5</sup> )	0,36 (-0,38)
TI_plato_26	15 min	0,17 (-0,10)	0,05% (-0,04%)	0,05 (-0,06)	0,21 (-0,12)	0,99 (+0,01)
	30 min	0,34 (-0,12)	0,11% (-0,04%)	0,20 (-0,12)	0,45 (-0,12)	0,96 (+0,02)
	45 min	0,56 (-0,10)	0,18% (-0,03%)	0,55 (-0,11)	0,74 (-0,07)	0,89 (+0,02)
TI_cabeza	15 min	0,19 (+0,01)	0,06% (-0,00%)	0,06 (-0,00)	0,25 (+0,01)	0,99 (+0,00)
	30 min	0,43 (+0,07)	0,14% (+0,02%)	0,30 (+0,10)	0,55 (+0,10)	0,94 (-0,02)
	45 min	0,67 (+0,13)	0,22% (+0,04%)	0,71 (+0,26)	0,84 (+0,17)	0,85 (-0,05)

Tabla 6.4: Comparación de métricas entre TCN optimizado y su configuración inicial

Como puede apreciarse, los resultados obtenidos por la TCN optimizada no muestran una mejora significativa. Aunque hay variables en las que se observan pequeñas mejoras, estas diferencias no parecen suponer un cambio importante. De hecho, en el caso de 'FI\_salida\_C4' y 'TI\_cabeza' los errores se mantienen igual o incluso aumentan tras este proceso.

Este comportamiento puede deberse a la complejidad del espacio de búsqueda en los hiperparámetros de la TCN, dificultando encontrar combinaciones óptimas. A pesar de emplear Optuna, es posible que la búsqueda haya convergido hacia regiones subóptimas dado el alto número de combinaciones y el limitado número de evaluaciones disponibles realizadas.

Para analizar estas diferencias, a continuación, se incluyen las gráficas de las predicciones de ambas arquitecturas sobre el tramo de 500 puntos del *test* con los tres horizontes.

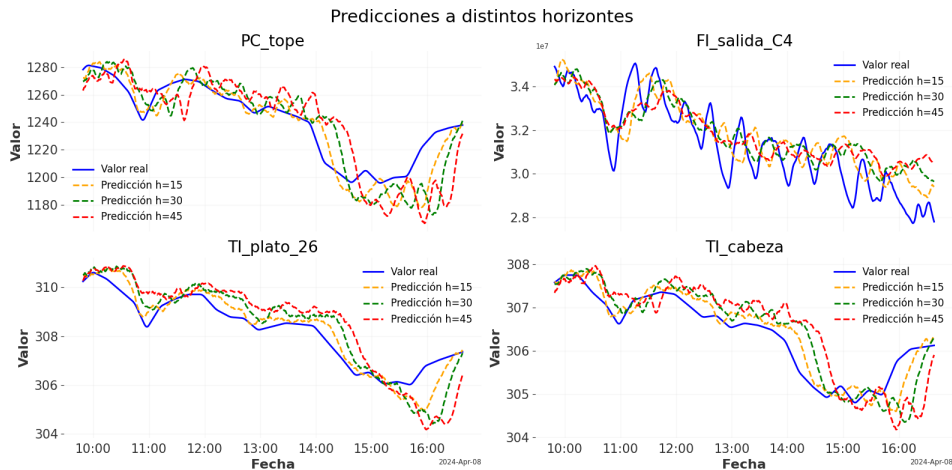


Figura 6.7: Predicciones de TCN base en tres horizontes

En la Figura 6.7 (modelo inicial), la predicción es razonablemente buena en tramos cortos, pero en general comete considerables errores. Aunque la tendencia general suele seguirse, el modelo tiende a exagerar tanto subidas como bajas, lo cual se ve pronunciado a medida que aumenta el horizonte. Esto último no ocurre con 'FI\_salida\_C4', donde la señal predicha se aplana sin captar las variaciones reales.

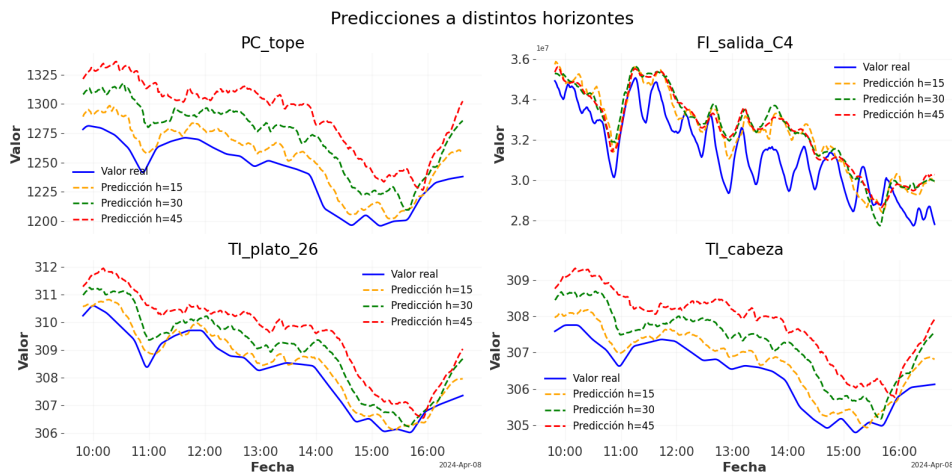


Figura 6.8: Predicciones de TCN optimizada en tres horizontes

Por otro lado, en la Figura 6.8 se observa como el modelo optimizado mejora la alineación temporal, no hay tanto desfase, y sigue de manera correcta la tendencia, pero presenta un problema con la escala. A medida que crece el rango de predicción, los valores generados tienden a aumentar progresivamente, alejándose de los reales.

Para completar el análisis, se añade la predicción de ambos modelos en una ventana de 45 minutos (la misma usada en secciones anteriores), para comprobar

## Capítulo 6. Resultados

el comportamiento puntual de ellos en ese momento concreto. Otras ventanas se añaden en el Anexo A.

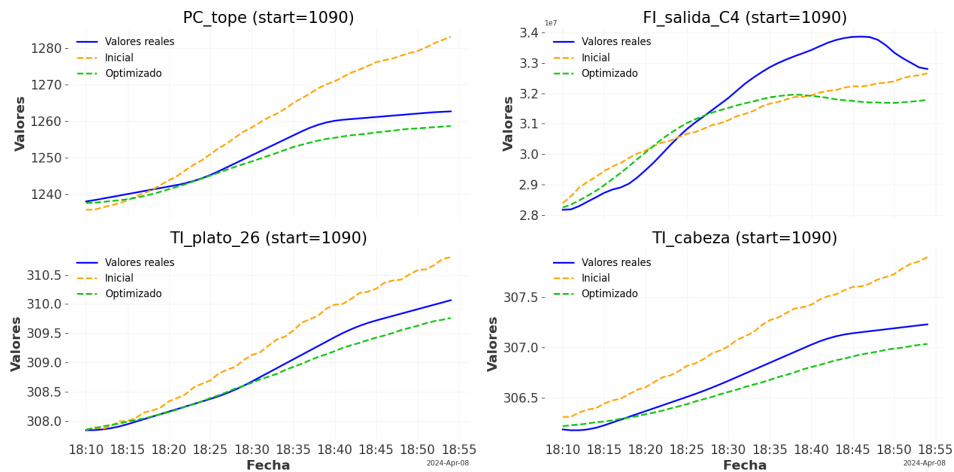


Figura 6.9: Predicción puntual de TCN sobre una ventana de 45 minutos

En este caso, el modelo optimizado parece ajustar mejor la predicción, especialmente en las temperaturas y presión, donde el error cometido es menor. No obstante, este comportamiento depende del tramo que se observe, ya que esa separación en escala está más presente en algunos tramos, mientras que en otros es más sutil.

En general, la TCN no termina de ofrecer un rendimiento del todo satisfactorio. Las predicciones en tramos cortos parecen aceptables, pero siguen presentando tanto problemas de escala como de alineación temporal según el modelo. Como ocurría en casos anteriores, la variable 'FI\_salida\_C4' continúa sin ajustarse correctamente, predecible por lo explicado. En las otras tres variables, dependiendo del momento y contexto, el modelo es capaz de captar ciertos comportamientos.

Por tanto, se trata de una arquitectura con potencial, pero que en este caso necesita ajustes adicionales, como una pequeña corrección de la escala, para ser aplicable en este contexto.

### 6.6. DLinear y NLinear

Estas dos arquitecturas surgen como alternativas simples y directas para predicciones de este tipo. Como se explicó en el Capítulo 3, su funcionamiento se basa en la aplicación de una serie de transformaciones sobre la secuencia de entrada, seguidas de capas lineales que simplifican el aprendizaje. Ambos modelos comparten la misma estructura de entrada y salida que el resto (con los hiperparámetros 'input\_chunk\_length' y 'output\_chunk\_length') y, además, presentan los siguientes:

- 'learning\_rate': define la velocidad de actualización del modelo durante el

## 6.6. DLinear y NLinear

entrenamiento, controla el tamaño de los pasos al minimizar el error. Se parte de un valor inicial estándar de 0,001, siendo este el valor por defecto y típicamente empleado. Como se estableció al comienzo, todas las arquitecturas de redes neuronales emplean este valor. En el caso de los modelos DLinear y NLinear se ha decidido intentar optimizar este parámetro para ampliar el espacio de búsquedas más óptimas, al tratarse de arquitecturas sencillas.

- 'batch\_size': al que se le da el mismo valor inicial que el resto de los vistos, 1024.
- 'n\_epochs': al igual que en las anteriores arquitecturas se parte del mismo valor, 60.

En el caso de DLinear, se incorpora un parámetro más, 'kernel\_size', que define el tamaño de la ventana utilizada para el suavizado mediante media móvil, técnica empleada para obtener la componente de la tendencia. El valor base que se ha usado es 25, ya que es el valor inicial por defecto que propone la librería.

Tras esta configuración, ambos son optimizados siguiendo el mismo procedimiento que en el resto de las arquitecturas. En el modelo DLinear los hiperparámetros que son modificados son el tamaño del *kernel*, aumentando en 5, es decir, con un valor de 30, y el '*learning rate*' a 0,0878, el resto se mantienen igual. En cuanto a NLinear, se reduce el tamaño de la ventana de entrada, 'input\_chunk\_length' a 10, y se ajusta el '*learning rate*' a 0,0027.

Los resultados obtenidos con esta nueva configuración se muestran en la Tabla 6.5, recogiendo todas las métricas para cada uno de ellos en los tres horizontes.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_topo	15 min DL	6,67 (-1,01)	0,51% (-0,03%)	68,59 (-10,27)	8,28 (-1,01)	0,98 (+0,00)
	15 min NL	4,76 (-0,12)	0,36% (-0,00%)	45,01 (-1,26)	6,70 (-0,09)	0,99 (+0,00)
	30 min DL	13,41 (-1,04)	1,02% (-0,08%)	267,07 (-21,47)	16,34 (-0,67)	0,95 (+0,00)
	30 min NL	9,5 (-0,08)	0,73% (-0,01)	165,63 (-18,03)	12,87 (-0,41)	0,97 (+0,00)
	45 min DL	21,61 (-2,02)	1,64% (-0,13%)	668,09 (-54,82)	25,85 (-2,04)	0,88 (+0,01)
	45 min NL	14,24 (-0,03)	1,08% (-0,00%)	341,81 (-26,94)	18,49 (-0,72)	0,94 (+0,00)
FI_salida_C4	15 min DL	4,54x10 <sup>5</sup> (-4,78x10 <sup>4</sup> )	1,49% (-0,00%)	3,30 x10 <sup>11</sup> (-1,52 x10 <sup>9</sup> )	5,74 x10 <sup>5</sup> (-1,32x10 <sup>3</sup> )	0,88 (+0,00)
	15 min NL	7,76 x10 <sup>5</sup> (-6,44x10 <sup>4</sup> )	2,59% (-0,15%)	8,80 x10 <sup>11</sup> (-1,06x10 <sup>11</sup> )	9,38 x10 <sup>5</sup> (-5,51x10 <sup>4</sup> )	0,70 (+0,03)

## Capítulo 6. Resultados

	30 min DL	4,49 x10 <sup>5</sup> (-6,93x10 <sup>3</sup> )	1,48% (-0,02%)	3,28 x10 <sup>11</sup> (-1,64x10 <sup>10</sup> )	5,73 x10 <sup>5</sup> (-1,41x10 <sup>4</sup> )	0,89 (+0,00)
	30 min NL	1,05 x10 <sup>6</sup> (-1,32x10 <sup>5</sup> )	3,53% (-0,33%)	1,53 x10 <sup>12</sup> (-3,03x10 <sup>11</sup> )	1,24 x10 <sup>6</sup> (-1,17x10 <sup>5</sup> )	0,48 (+0,10)
	45 min DL	5,03 x10 <sup>5</sup> (-9,54x10 <sup>3</sup> )	1,66% (-0,03%)	4,16 x10 <sup>11</sup> (-2,13x10 <sup>10</sup> )	6,45 x10 <sup>5</sup> (-1,77x10 <sup>4</sup> )	0,86 (+0,00)
	45 min NL	1,25 x10 <sup>6</sup> (-2,83x10 <sup>5</sup> )	4,21% (-0,81%)	2,12 x10 <sup>12</sup> (-8,89x10 <sup>11</sup> )	1,45 x10 <sup>6</sup> (-2,79x10 <sup>5</sup> )	0,29 (+0,29)
<b>TI_plato_26</b>	15 min DL	0,22 (-0,03)	0,07% (-0,01%)	0,07 (-0,00)	0,26 (-0,05)	0,98 (+0,00)
	15 min NL	0,15 (+0,02)	0,05% (-0,00)	0,03 (+0,01)	0,19 (+0,04)	0,99 (+0,00)
	30 min DL	0,49 (+0,07)	0,16% (+0,01%)	0,32 (+0,07)	0,57 (+0,07)	0,93 (-0,01)
	30 min NL	0,29 (-0,06)	0,09% (-0,01%)	0,13 (-0,04)	0,36 (-0,05)	0,97 (+0,00)
	45 min DL	0,74 (+0,06)	0,24% (+0,00%)	0,74 (+0,09)	0,86 (+0,04)	0,85 (+0,02)
	45 min NL	0,45 (-0,13)	0,15% (-0,04%)	0,29 (-0,09)	0,54 (-0,09)	0,94 (+0,01)
<b>TI_cabeza</b>	15 min DL	0,12 (-0,01)	0,04% (-0,00%)	0,03 (-0,00)	0,16 (-0,01)	0,99 (+0,00)
	15 min NL	0,12 (-0,00)	0,04% (-0,00%)	0,02 (-0,00)	0,16 (-0,01)	0,99 (+0,00)
	30 min DL	0,31 (-0,00)	0,10% (-0,00%)	0,15 (-0,00)	0,38 (-0,01)	0,96 (+0,00)
	30 min NL	0,24 (-0,01)	0,08% (-0,00%)	0,10 (-0,00)	0,32 (-0,00)	0,97 (+0,00)
	45 min DL	0,50 (-0,01)	0,16% (-0,00%)	0,37 (-0,01)	0,61 (-0,01)	0,92 (+0,00)
	45 min NL	0,37 (-0,02)	0,12% (-0,00%)	0,22 (-0,01)	0,47 (-0,01)	0,95 (+0,00)

Tabla 6.5: Comparación de métricas entre DLinear y NLinear optimizados y sus configuraciones iniciales

Ambos modelos presentan unas métricas positivas, con errores generalmente bajos y un  $R^2$  muy cercano a 1 en gran parte de las variables y horizontes. Los dos muestran un rendimiento y resultado muy similares. Para complementar esta información, del mismo modo que en los apartados anteriores, se incluyen las siguientes visualizaciones.

En primer lugar, la Figura 6.10 recoge el comportamiento del modelo DLinear. Se aprecia un ajuste visualmente muy preciso, sobre todo en el horizonte de 15 minutos, donde las predicciones son prácticamente iguales a la señal real. En 'TI\_plato\_26' los tres horizontes se mantienen muy alineados y próximos a la realidad a lo largo del tramo. En 'FI\_salida\_C4', a pesar de las dificultades

expuestas en modelar esta variable, los resultados son razonablemente buenos. El modelo ajusta bien varias de las caídas y responde de forma coherente a los cambios. Aun así, al igual que ocurría en modelos anteriores, en los horizontes de 30 y, especialmente 45 minutos, se empieza a notar un ligero desfase, aunque no muy marcado.

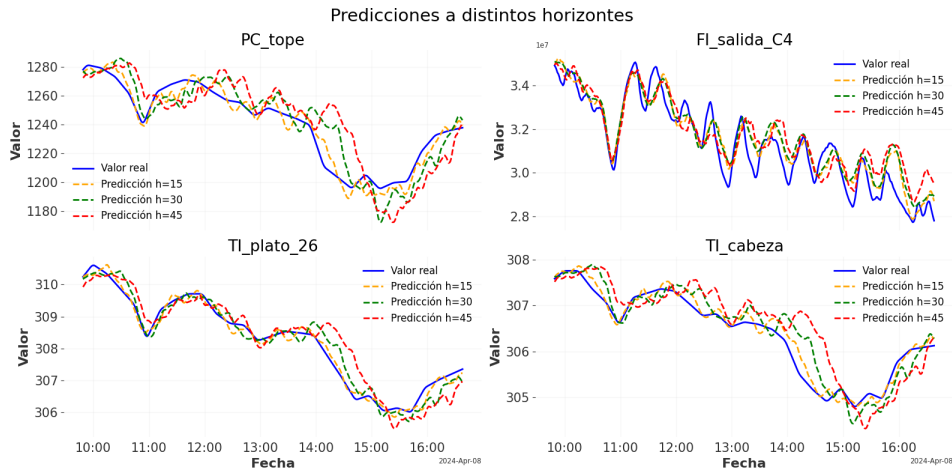


Figura 6.10: Predicciones de DLinear en tres horizontes

En segundo lugar, el modelo NLinear, Figura 6.11, también muestra un comportamiento positivo. Como el DLinear, en el menor de los horizontes, las predicciones son bastante exactas y la tendencia general se sigue. Los errores empiezan a aumentar en los otros dos casos, donde se nota en mayor medida el desfase temporal, aunque no afecta de forma crítica al ajuste global. Sin embargo, en la variable 'FI\_salida\_C4', aunque el modelo predice muy bien la primera caída, en el resto se aprecia una degradación alejada de la realidad. Aun así, se mantiene un buen comportamiento general.

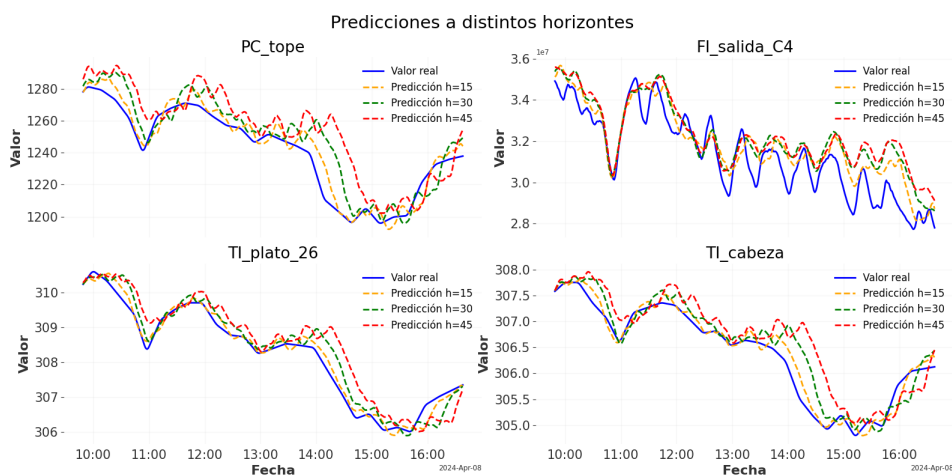


Figura 6.11: Predicciones de NLinear en tres horizontes

## Capítulo 6. Resultados

Por otro lado, se incluye en la Figura 6.12 la predicción puntual de ambos modelos en la ventana de 45 minutos seleccionada, facilitando la comparación directa entre las arquitecturas en una misma situación temporal. Se incorporan otras ventanas alternativas en el Anexo A.

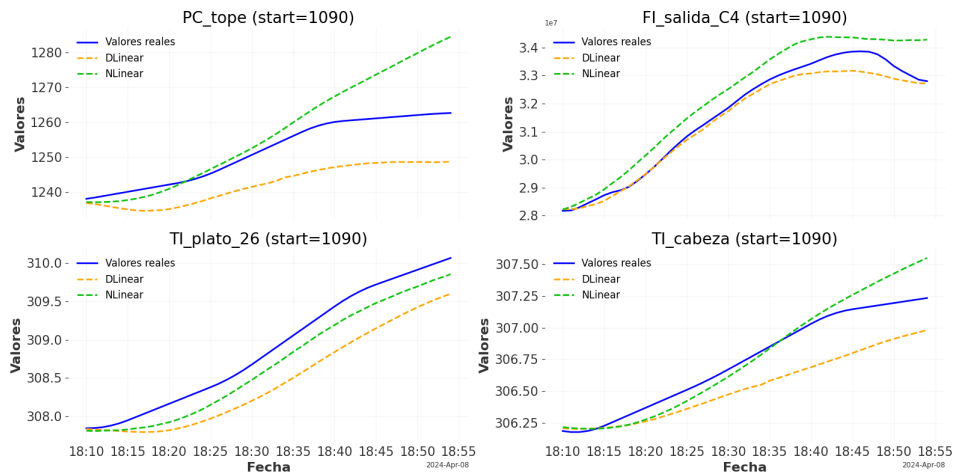


Figura 6.12: Predicción puntual de DLinear y NLinear sobre una ventana de 45 minutos

En esta ventana concreta se observa lo analizado previamente en ambos modelos. Los dos presentan un ajuste considerablemente bueno, y son capaces de captar la tendencia general. En ciertas ocasiones, NLinear tiende a sobreestimar, especialmente en los últimos puntos de predicción, aunque mantiene un trazo muy similar a DLinear. Las diferencias entre ellos en este tramo son sutiles, lo que refuerza su buena adaptación.

Ambos destacan por su sencillez y bajo coste, ofreciendo al mismo tiempo resultados sólidos, especialmente en horizontes cortos. NLinear ha mostrado un comportamiento más consistente en las métricas, pero DLinear parece conseguir un ajuste muy preciso en varias variables en las visualizaciones. En general, son modelos eficientes que pueden ofrecer un rendimiento notable sin necesidad de recurrir a arquitecturas más complejas y con mayor coste computacional.

### 6.7. TFT

La última arquitectura a evaluar es la Temporal Fusion Transformer o TFT, basada en los modelos de atención. Se incluye en la comparativa como una de las propuestas más avanzadas dentro del conjunto estudiado. Su diseño, en principio, permite combinar diferentes fuentes de información, identificar patrones temporales y adaptar las predicciones a contextos cambiantes.

Dada la complejidad de este modelo, se ha decidido partir de la arquitectura base de la librería empleada. Esta contiene los siguientes hiperparámetros:

- 'input\_chunk\_length': 25, y 'output\_chunk\_length': 1 (establecidos como

el resto para mantener la coherencia).

- 'hidden\_size': tamaño del espacio de representación interno del modelo.
- 'lstm\_layers': número de capas LSTM en la parte secuencial del codificador (1).
- 'num\_attention\_heads': número de cabezas utilizadas en el mecanismo de atención múltiple (4).
- 'dropout': 0.0
- 'batch\_size': 8192
- 'n\_epochs': 100

Haciendo pruebas iniciales con esta configuración se obtienen los siguientes resultados, mostrados en la Figura 6.13.

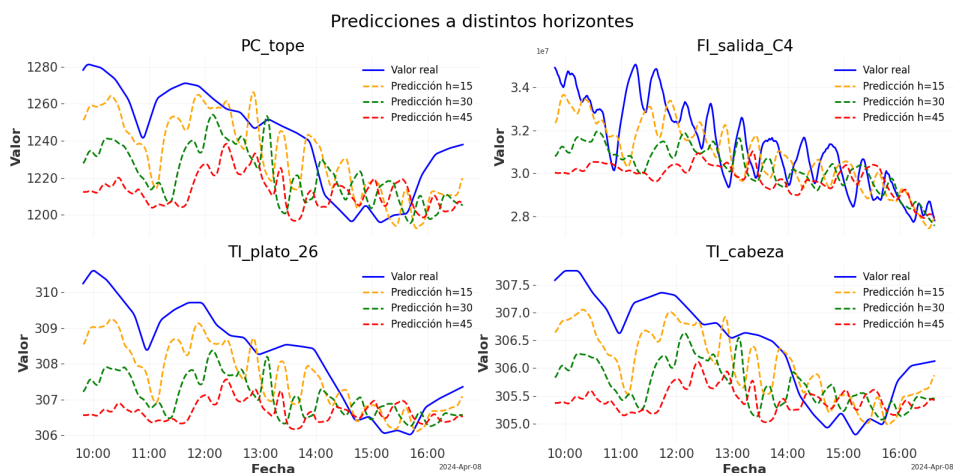


Figura 6.13: Predicciones de TFT en tres horizontes

Como se puede observar en la imagen, el modelo produce unas señales bastante ruidosas, alejadas de los valores reales. A medida que avanzan los horizontes, los errores se acumulan y las estimaciones pierden la forma de las señales originales.

Para hacer una pequeña prueba con la que analizar si el modelo puede ofrecer resultados más útiles, se decide considerar el de partida con un ajuste. Para evitar unas predicciones tan ruidosas, generadas probablemente por la complejidad y el aprendizaje de ciertas tendencias durante el entrenamiento, se introduce un 'dropout' de 0,2. Esto añade una regularización con la que se consiguen las siguientes predicciones.

## Capítulo 6. Resultados

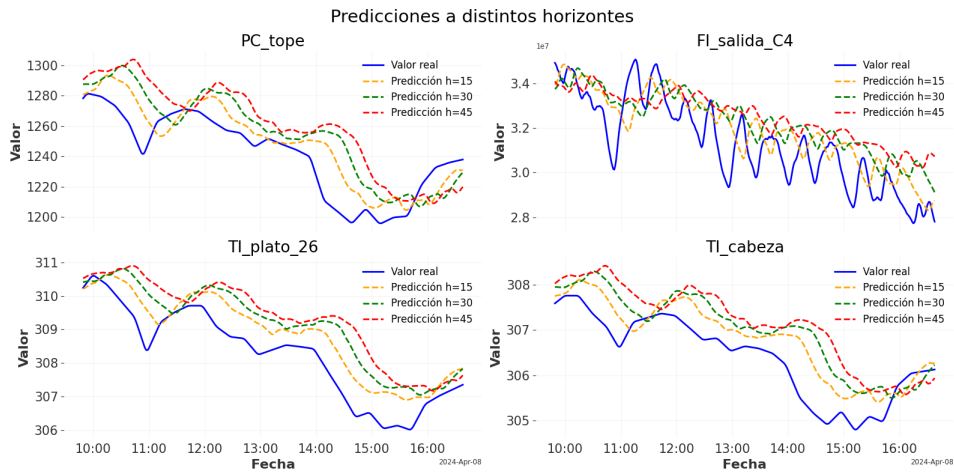


Figura 6.14: Predicciones de TFT con regularización en tres horizontes

En la Figura 6.14 se aprecia cómo esta normalización produce mejoras considerables. Las señales resultantes ya no muestran ese ruido inicial, con estimaciones mucho más suaves y estructuradas. Consiguen mantener la forma coherente y adaptarse en cierta medida a las tendencias reales. A pesar de esta mejora, el modelo sigue sin lograr ajustar completamente las predicciones con el valor real. Se producen desviaciones marcadas, tanto temporales como de escala. Para analizar esta situación de manera más detallada, se incluyen las predicciones obtenidas para la ventana de 45 minutos empleada en el resto de los modelos.

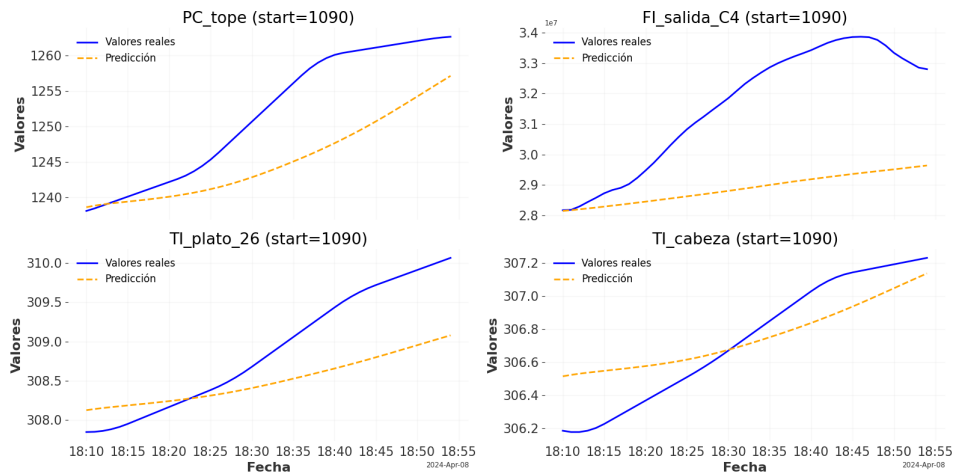


Figura 6.15: Predicción puntual de TFT sobre una ventana de 45 minutos

La Figura 6.15 refuerza lo observado en la imagen anterior. Aunque el modelo ya no presenta un ruido excesivo como ocurría en el primer caso, las predicciones siguen sin adaptarse y seguir la realidad. En las variables de temperatura y presión se nota cómo ligeramente la tendencia de crecimiento está presente en las estimaciones. Sin embargo, en 'FI\_salida\_C4' no es capaz de capturar la

## 6.8. Comparación conjunta

dinámica, y se mantiene prácticamente lineal.

A la vista de los resultados, TFT no ha ofrecido el rendimiento esperado en el problema planteado. Su alto coste computacional no se compensa ni justifica con la complejidad y dificultad que presenta para adaptarse a las dinámicas de las variables. Por ello, no se considera oportuno llevar a cabo una optimización de sus hiperparámetros. Aun así, se presenta a continuación la tabla con las métricas obtenidas por este modelo para realizar un análisis cuantitativo y poder compararlo con el resto de arquitecturas.

Variable	Horizonte	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_tope	15 min	14,86	1,12%	381,17	19,52	0,93
	30 min	20,22	1,51%	674,63	25,97	0,88
	45 min	24,75	1,86%	998,60	31,61	0,83
FI_salida_C4	15 min	8,17x10 <sup>5</sup>	2,71%	1,06x10 <sup>12</sup>	1,03 x10 <sup>6</sup>	0,64
	30 min	9,62x10 <sup>5</sup>	3,19%	1,51x10 <sup>12</sup>	1,23 x10 <sup>6</sup>	0,49
	45 min	1,10x10 <sup>6</sup>	3,67%	1,95x10 <sup>12</sup>	1,40 x10 <sup>6</sup>	0,35
TI_plato_26	15 min	0,39	0,13%	0,23	0,48	0,95
	30 min	0,53	0,17%	0,43	0,66	0,92
	45 min	0,66	0,21%	0,68	0,82	0,87
TI_cabeza	15 min	0,34	0,11%	0,19	0,44	0,95
	30 min	0,48	0,16%	0,37	0,61	0,92
	45 min	0,61	0,20%	0,58	0,76	0,87

Tabla 6.6: Métricas de evaluación de TFT

## 6.8. Comparación conjunta

Una vez evaluadas todas las arquitecturas, se establece una comparación global que permite identificar las principales diferencias en cuanto a rendimiento, estabilidad, eficiencia y adaptabilidad en cada uno de los horizontes. Se incluye la siguiente tabla en la que se recoge el mejor modelo para cada caso, variable y horizonte, con sus métricas correspondientes. La selección de dicho mejor modelo se ha realizado a partir de las métricas de evaluación, comparando el MSE y RMSE, buscando minimizar el error cometido en las predicciones.

Variable	Horizonte	Modelo	MAE	MAPE	MSE	RMSE	R <sup>2</sup>
PC_tope	15 min	NLinear	4,76	0,36%	45,01	6,70	0,99
	30 min	NLinear	9,5	0,73%	165,63	12,87	0,97
	45 min	NLinear	14,24	1,08%	341,81	18,49	0,94
FI_salida_C4	15 min	DLinear	4,54x10 <sup>5</sup>	1,49%	3,30x10 <sup>11</sup>	5,74x10 <sup>5</sup>	0,88
	30 min	DLinear	4,49x10 <sup>5</sup>	1,48%	3,28x10 <sup>11</sup>	5,73x10 <sup>5</sup>	0,89

## Capítulo 6. Resultados

	45 min	DLinear	5,03x10 <sup>5</sup>	1,66%	4,16x10 <sup>11</sup>	6,45x10 <sup>5</sup>	0,86
<b>TI_plato_26</b>	15 min	NLinear	0,15	0,05%	0,03	0,19	0,99
	30 min	NLinear	0,29	0,09%	0,13	0,36	0,97
	45 min	NLinear	0,45	0,15%	0,29	0,54	0,94
<b>TI_cabeza</b>	15 min	NLinear	0,12	0,04%	0,02	0,16	0,99
	30 min	NLinear	0,24	0,08%	0,10	0,32	0,97
	45 min	NLinear	0,37	0,12%	0,22	0,47	0,95

Tabla 6.7: Métricas de evaluación del mejor modelo en cada caso

A partir de los datos recogidos, puede observarse cómo NLinear es el mejor modelo, especialmente en las variables de temperatura y presión. Su rendimiento destaca, sobre todo en horizontes cortos, al igual que ocurría con el resto de arquitecturas, donde se logra ajustar la escala de forma precisa, consiguiendo errores bajos. Aun así, en los horizontes más amplios también muestra una gran consistencia, a diferencia de otros que comienzan a degradarse.

En el caso de la variable 'FI\_salida\_C4', el modelo que obtiene los mejores resultados es el DLinear. A pesar de que es la más difícil de predecir debido a su comportamiento irregular y posible información faltante en la entrada, consigue valores de error aceptables y, sobre todo, una mejora clara frente al resto.

Esta comparación conjunta permite comprobar que no es siempre necesario recurrir a arquitecturas muy complejas y elaboradas para obtener predicciones precisas y de calidad [39]. En este caso, los modelos (de Deep Learning) menos complejos planteados y con menor coste computacional han superado el rendimiento de otros más avanzados como TCN y TFT. Parte del buen rendimiento de estas arquitecturas se puede deber, en parte, a la incorporación de técnicas de procesamiento de señales temporales, como las descritas en la Sección 3.1, donde se analizó la estructura típica de las series temporales.

Por otra parte, aunque no haya obtenido mejores métricas que estos dos, la LSTM también ha destacado, con resultados muy similares. En un horizonte de 15 minutos ha demostrado una buena capacidad de ajuste y adaptación. A pesar de no ser considerado el mejor, podría aplicarse al problema tratado como una alternativa perfectamente válida. El buen resultado mostrado de este tipo de modelos es esperado visto el éxito y uso extenso en la literatura de series temporales con Deep Learning.

En caso contrario, modelos que en principio parecían prometedores por las avanzadas técnicas que incorporan, no han conseguido alcanzar el rendimiento esperado. La TCN y la TFT han presentado dificultades para generalizar, cometiendo errores considerables en cada uno de los horizontes, sin compensar su complejidad ni alto coste computacional en este contexto. Sin embargo, es posible que estas arquitecturas sofisticadas requieran de conjuntos de datos más amplios y variados para mostrar su potencial.

## **6.8. Comparación conjunta**

---

Igualmente, todos los modelos, al menos sus versiones optimizadas, han logrado superar a los modelos base de referencia. Esto demuestra que, aunque sus capacidades parezcan ser limitadas en ciertos casos, suponen una mejora significativa.

Se puede concluir con que es fundamental adaptar el modelo al tipo de señal y horizonte que se desee predecir, sin existir una única arquitectura válida para todos los casos. Es por ello que la elección del modelo que mejor se ajuste dependerá de las características del sistema, los recursos disponibles y los objetivos concretos del proyecto.



## Capítulo 7

# Conclusiones y trabajo futuro

El trabajo desarrollado sienta una base sólida para la implementación de sistemas de *soft sensors* basados en datos en entornos industriales. Los modelos de Deep Learning ofrecen un gran potencial para la estimación de variables críticas en procesos. A través del uso de datos operativos reales proporcionados por Repsol, y centrados en una columna de destilación fraccionada, se ha demostrado el potencial que pueden ofrecer estas técnicas para el desarrollo de sensores virtuales.

Algunas de las principales ventajas de este tipo de modelado son que no solo permite reducir la dependencia de instrumentación física, sino que también mejora la eficiencia operativa. Permite anticiparse a desviaciones del proceso y apoya la toma de decisiones en tiempo real [6].

Durante todo el proceso se han probado distintas arquitecturas, desde modelos más sencillos y clásicos a otros más avanzados. Se ha comprobado que no siempre es necesario recurrir a las arquitecturas más complejas y novedosas para obtener los mejores resultados. Modelos como DLinear y NLinear, con estructuras mucho más simples, han superado en gran parte de los casos a alternativas como los Transformers, muy aclamadas últimamente. Por ejemplo, NLinear logró un MAPE de tan solo 0,36% y un  $R^2$  de 0,99 en la predicción de 'PC\_tope' a 15 minutos, obteniendo métricas similares en las variables de temperatura ('TI\_plato\_26' y 'TI\_cabeza'). Por su parte, el modelo DLinear destacó en la estimación de 'FI\_salida\_C4', más compleja y con mayor variabilidad, con MAPEs en torno al 1,5% y un  $R^2$  de hasta 0,89. Demuestran así su robustez y capacidad de generalización en situaciones como la estudiada.

Todo ello, por tanto, refuerza la idea de que la complejidad del modelo no es fundamental ni proporcional a su rendimiento, sino que la clave es cómo se adapta a la naturaleza de los datos y el horizonte de predicción. La elección más apropiada depende de cada caso, los objetivos, características, necesidades y propósitos definidos.

En esta línea, también se ha comprobado que la calidad de los datos y un preprocesamiento detallado impactan de forma significativa en el rendimiento de los

## Capítulo 7. Conclusiones y trabajo futuro

---

modelos. Una preparación de los datos facilita el aprendizaje y, además, puede maximizar la eficacia de estos.

A partir de esta base, si bien es cierto que los modelos desarrollados se enfocan en la problemática planteada, la metodología empleada puede extenderse fácilmente a otros sectores industriales con dinámicas similares como la química, farmacéutica o alimentaria. En estos contextos, un *soft sensor* como el implementado permitiría realizar estimaciones fiables de variables optimizando el proceso real. No sería necesario incorporar sensores adicionales o interrumpir la operación.

No obstante, este enfoque representa un primer paso dentro de un campo amplio y en constante evolución. A partir de los resultados obtenidos y las limitaciones observadas, se abren diversas líneas de trabajo futuro. Están orientadas a mejorar el rendimiento, ampliar la aplicabilidad del sistema y facilitar su integración en entornos industriales complejos. Se proponen a continuación algunas direcciones de investigación que permitan consolidar los resultados obtenidos en el trabajo.

- Ampliación de la cantidad de datos y su calidad. Contar con mayor cantidad de datos, provenientes de diferentes condiciones de operación podría mejorar la capacidad de generalización y adaptación del modelo. En el estudio para su desarrollo únicamente se emplearon dos meses del registro completo dada la variabilidad y ruido que contenían. Sin embargo, ampliar ese rango podría favorecer un mayor aprendizaje.
- Análisis profundo de las variables empleadas. La elección de variables correctas a emplear o la incorporación de variables adicionales podrían enriquecer la representación del proceso. Se partió del conjunto propuesto por los expertos del sistema, pero con los resultados obtenidos volver a revisar el conjunto podría aportar valor a las estimaciones.
- Exploración de nuevas arquitecturas. Aunque los modelos actuales han demostrado buen rendimiento, sería interesante estudiar otras variantes de redes híbridas, arquitecturas novedosas desarrolladas para series temporales con Transformers. Estos últimos no se han podido probar con profundidad, pero quizás ciertas arquitecturas y configuraciones lograrían adaptarse mejor.
- Incorporación de modelos probabilísticos. Probar enfoques que proporcionen estimaciones de incertidumbre mediante bandas de confianza y no únicamente predicciones puntuales podría aportar gran valor. Permitiría evaluar la fiabilidad de las predicciones, especialmente útil en entornos donde la seguridad es prioritaria.
- Implantación en planta industrial. Una aplicación directa del trabajo realizado sería su implantación como *soft sensor* en la planta correspondiente. Permitiría sustituir sensores físicos en caso de fallo, mantenimiento o para estimar variables que no disponen de sensor.
- Predicción a corto plazo para la toma de decisiones. El uso de estas pre-

---

dicciones podría formar parte de un sistema de control o recomendación que ayude a una optimización en tiempo real, anticipando desviaciones o detectando comportamientos anómalos.

- Simulación del proceso. Otra posible línea futura de trabajo, ligada a las anteriores, sería emplear los modelos desarrollados como base para construir simuladores del proceso. Estos entornos virtuales permitirían probar estrategias de optimización. A través de agentes entrenados sobre los modelos, sería posible diseñar políticas de control eficientes, mejorando el rendimiento y autonomía del sistema.

En resumen, el trabajo realizado no solo ha permitido validar la utilidad de los modelos de Deep Learning como *soft sensors*, sino que también ha abierto múltiples caminos para investigaciones futuras. Se ha validado una metodología completa, desde la selección y tratamiento de los datos, hasta la evaluación de diferentes arquitecturas, mostrando cómo incluso enfoques sencillos pueden superar a alternativas complejas si se ajustan al problema.

Todo ello conforma las bases para avanzar a una industria más inteligente donde la monitorización virtual, el control avanzado y la toma de decisiones basada en datos formen parte de la rutina operativa. Con el crecimiento del interés por la digitalización y la inteligencia artificial en los últimos años, representa una apuesta sólida, escalable y transformadora en la optimización de procesos en tiempo real.



## Capítulo 8

# Análisis de impacto

### Impacto personal y académico

El desarrollo de este Trabajo de Fin de Grado ha supuesto un reto significativo al mismo tiempo que una oportunidad de crecimiento tanto a nivel personal como académico. La elección de un tema como los *soft sensors* ha exigido el estudio de este tipo de herramientas emergentes y la aplicación práctica de los conocimientos adquiridos durante la carrera, pero sobre todo durante los meses de prácticas en Repsol. Un aspecto especialmente enriquecedor ha sido la posibilidad de trabajar con datos reales provenientes de una instalación industrial de esta empresa. Ha añadido realismo, complejidad y relevancia al proceso.

Desde el punto de vista académico, el proyecto contribuye al conocimiento sobre el diseño y validación de sensores virtuales. Se ha podido realizar una comparación crítica que ha permitido una mejor adaptación del sensor a las condiciones reales del proceso gracias a la evaluación de múltiples enfoques de modelado, con diferentes modelos y arquitecturas del estado del arte. Asimismo, este trabajo me ha brindado la oportunidad de crecer profesionalmente al colaborar con profesionales del sector en Repsol, aprendiendo de su experiencia, sus conocimientos técnicos y sus metodologías de trabajo. El proceso completo ha sido un aprendizaje tanto de habilidades técnicas como competencias transversales (gestión del tiempo, documentación rigurosa, trabajo autónomo, entre otras).

### Impacto empresarial y económico

La aplicación de un *soft sensor* en un entorno industrial puede tener un impacto económico significativo. Esto se debe a que permite reducir o incluso evitar el uso de sensores físicos, que en la mayoría de ocasiones van acompañados de un coste elevado. Además, se ahorran otros costes relacionados con el mantenimiento o la calibración de estos sensores. Todo esto ayuda a que el proceso sea más rentable. Asimismo, el sensor virtual permite monitorear variables críticas en tiempo real, lo que facilita la toma de decisiones informadas y mejora la eficiencia de los procesos.

Desde una perspectiva empresarial, el uso de esta tecnología es especialmente

## Capítulo 8. Análisis de impacto

---

útil en sectores donde es importante tener un buen control del proceso y aprovechar al máximo los recursos. También se abre la posibilidad de aplicar el uso de esta solución en diferentes líneas de producción o en diversos sectores, lo cual amplía las oportunidades de aplicación y puede contribuir a una mayor rentabilidad y diferenciación en el mercado.

### **Impacto sociocultural**

La creación e implementación de un *soft sensor* tiene un impacto social que, aunque pueda parecer indirecto, es significativo en varios niveles. Su capacidad para optimizar procesos industriales y reducir costes puede traducirse en una mejora de los productos o servicios finales. Además, al aumentar la eficiencia y precisión en sistemas de control, se mejora la seguridad en entornos de trabajo, lo que repercute positivamente en las condiciones laborales.

Desde un punto de vista cultural, el proyecto también supone un cambio en la forma en que valoramos la tecnología y el conocimiento científico. Utilizar modelos virtuales en lugar de sensores físicos representa una evolución en nuestra manera de aplicar y entender la tecnología. Este enfoque favorece una cultura más enfocada en la eficiencia, la sostenibilidad y el trabajo conjunto entre distintas disciplinas.

### **Impacto medioambiental**

Uno de los beneficios más notables de los sensores virtuales es su contribución a la sostenibilidad ambiental. Al eliminar la necesidad de fabricar y desechar sensores físicos, se reduce el consumo de materiales y la generación de residuos electrónicos. Asimismo, al permitir una monitorización más precisa y continua, el *soft sensor* puede contribuir a optimizar procesos energéticos, reducir emisiones contaminantes y disminuir el uso de recursos naturales.

Este impacto se vuelve especialmente relevante en industrias de alto consumo energético o potencialmente contaminantes, donde una pequeña mejora en eficiencia puede traducirse en grandes beneficios medioambientales. La capacidad de este tipo de sensores para integrar métricas ambientales y anticipar comportamientos inusuales lo convierte también en una herramienta útil para cumplir normativas ecológicas y avanzar hacia una producción más responsable.

### **Alineación con los ODS**

Con el objetivo de avanzar hacia un mundo más equitativo a nivel económico, social y medioambiental, en el año 2015 se aprobó la Agenda 2030. Esta agenda establece una serie de compromisos orientados al desarrollo sostenible, entre los cuales destacan los 17 Objetivos de Desarrollo Sostenible (ODS), que se desglosan en un total de 169 metas con el horizonte puesto en el año 2030.

En relación con este proyecto, los ODS implicados en mayor medida son los siguientes:



**ODS 9 Industria, innovación e infraestructura:** El proyecto se encuentra alineado de forma directa con este ODS al tratarse de una solución tecnológica que impulsa la transformación digital en procesos industriales. Este tipo de innovación mejora la eficiencia operativa, reduce costes y permite a las empresas adoptar modelos más sostenibles y avanzados, fomentando la investigación y el desarrollo de estas tecnologías.



**ODS 11 Ciudades y comunidades sostenibles:** Los sensores virtuales pueden contribuir a una gestión más eficiente de servicios urbanos, especialmente en aplicaciones como la monitorización ambiental, la gestión energética o el control de procesos en infraestructuras críticas. Estas aplicaciones ayudan a mejorar la calidad de vida urbana, reduciendo el impacto ambiental y optimizando los recursos disponibles.

De forma indirecta, el proyecto se podría vincular con el **ODS 8 Trabajo decente y crecimiento económico**, ya que el uso de tecnologías como los *soft tensors* impulsa la creación de empleo cualificado y fomenta la productividad industrial. Asimismo, el proyecto también refleja aspectos del **ODS 17 Alianzas para lograr los objetivos**, en tanto que promueve la colaboración interdisciplinar entre distintas áreas del conocimiento y puede integrarse en proyectos conjuntos con instituciones académicas o empresariales que compartan el mismo objetivo que este trabajo.

### **Decisiones tomadas en base al impacto**

Durante el desarrollo del proyecto se tomaron decisiones orientadas a maximizar su utilidad para futuras aplicaciones. Se optó por trabajar con datos reales y diversos modelos para asegurar una solución robusta y con resultados contrastados. Además, se priorizó el uso de herramientas accesibles y fácilmente replicables, evitando dependencias de *softwares* exclusivos, lo que favorece futuros desarrollos de este trabajo. Estas decisiones responden al objetivo de generar un impacto positivo no solo en el ámbito técnico, sino también en el económico, social y medioambiental.



# Bibliografía

- [1] S. López Blanco. «Estadísticas Esenciales de Inteligencia Artificial para 2025: Quién la usa y para qué». (2025), dirección: <https://www.iebschool.com/hub/estadisticas-esenciales-de-inteligencia-artificial-para-2025-quien-la-usa-y-para-que-tecnologia/> (visitado 20-04-2025).
- [2] J. Pearce. «AI: Key stats you need to know in 2025». (2025), dirección: <https://techinformed.com/global-ai-market-and-key-stats/> (visitado 20-04-2025).
- [3] A. Singla, A. Sukkarevsky, L. Yee y M. Chui, «El estado de la IA a principios de 2024: la adopción de la IA generativa aumenta y comienza a generar valor», *McKinsey Global Institute*, 2024.
- [4] Á. S. de Bedoya, «Avances y desafíos de la inteligencia artificial», Tesis doct., Master's thesis, Escuela Técnica Superior de Ingeniería (ICAI), Madrid, 2022.
- [5] P. A. C. Rivero y M. d. P. E. Mota, «Evolución de las Habilidades Laborales en la Industria 4.0 y su Impacto Financiero», *Revista Innova ITFIP*, vol. 6, n.º 1, págs. 106-119, 2020.
- [6] J. R. Rodríguez Ossorio, R. González Herbón, G. González Mateos et al., «Sensor virtual de caudal basado en técnicas de deep learning», en *Domínguez González, M., Cabrera Santana, PJ, Irigoyen Gordo, E.(eds.)(2022). XVII Simposio CEA de Control Inteligente: Reunión anual del grupo de Control Inteligente del comité español de automática (CEA). Libro de Actas, León, 27-29 de junio de 2022*, Universidad de León, págs. 81-86.
- [7] Hexagon. «Estadísticas de gemelos digitales en 2025». (2025), dirección: <https://hexagon.com/es/resources/insights/digital-twin/statistics> (visitado 20-04-2025).
- [8] S. Carlos y A. Morales, «Fundamentos y Aplicaciones de los Sensores Virtuales», en *Simposio de Metrología*, vol. 1659, 2008.
- [9] J. Gonzaga, L. Meleiro, C. Kiang y R. Maciel Filho, «ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process», *Computers & chemical engineering*, vol. 33, n.º 1, págs. 43-49, 2009.

- [10] C. Xie, R. Yao, L. Zhu, H. Gong, H. Li y X. Chen, «Soft-sensor development through deep learning with spatial and temporal feature extraction of complex processes», *Industrial & Engineering Chemistry Research*, vol. 62, n.º 1, págs. 519-534, 2022.
- [11] R. Ayadi, R. M. Abd El-Aziz, A. I. Taloba, H. Aljuaid, N. O. Hamed y M. A. Khder, «Deep learning-based soft sensors for improving the flexibility for automation of industry», *Wireless Communications and Mobile Computing*, vol. 2022, n.º 1, pág. 5450473, 2022.
- [12] Y. S. Perera, D. Ratnaweera, C. H. Dasanayaka y C. Abeykoon, «The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review», *Engineering Applications of Artificial Intelligence*, vol. 121, pág. 105988, 2023.
- [13] B. Lin, B. Recke, J. K. Knudsen y S. B. Jørgensen, «A systematic approach for soft sensor development», *Computers & chemical engineering*, vol. 31, n.º 5-6, págs. 419-425, 2007.
- [14] S. Vakaruk, «Contribuciones a la Aplicación de Machine Learning en Escenarios Novedosos de Tiempo Real», Tesis doct., ETSI\_Sistemas\_Infor, 2023.
- [15] A. K. Pani, K. G. Amin y H. K. Mohanta, «Soft sensing of product quality in the debutanizer column with principal component analysis and feed-forward artificial neural network», *Alexandria Engineering Journal*, vol. 55, n.º 2, págs. 1667-1674, 2016.
- [16] D. Montalvo García et al., «Construcción de algoritmos de aprendizaje automático para predicción de consumos energéticos en edificios inteligentes», 2020.
- [17] T. Lyu, Y. Gan, R. Zhang, S. Wang, D. Li e Y. Zhuo, «Development of a Real-Time NOx Prediction Soft Sensor Algorithm for Power Plants Based on a Hybrid Boost Integration Model», *Energies*, vol. 17, n.º 19, pág. 4926, 2024.
- [18] O. Savytskyi, M. Tymoshenko, O. Hramm y S. Romanov, «Application of soft sensors in the automated process control of different industries», en *E3S Web of Conferences*, EDP Sciences, vol. 166, 2020, pág. 05003.
- [19] G. Madrigal-Espinosa, C. M. Astorga-Zaragoza, M. Adam-Medina, G.-L. Osorio-Gordillo y M. Vázquez-Román, «Diagnóstico de fallos en el control y compensación de la medición por sensores virtuales en una central termoeléctrica», *DYNA*, vol. 92, n.º 5, págs. 545-551, 2017.
- [20] H. A. Al-Jamimi, S. Al-Azani y T. A. Saleh, «Supervised machine learning techniques in the desulfurization of oil products for environmental protection: A review», *Process Safety and Environmental Protection*, vol. 120, págs. 57-71, 2018.
- [21] J. E. Cote-Ballesteros, V. H. Grisales Palacios y J. E. Rodríguez-Castellanos, «A Hybrid Approach Variable Selection Algorithm Based on Mutual Information for Data-Driven Industrial Soft-Sensor Applications», *Ciencia e Ingeniería Neogranadina*, vol. 32, n.º 1, págs. 59-70, 2022.

- [22] C. Abeykoon, «Design and applications of soft sensors in polymer processing: A review», *IEEE Sensors Journal*, vol. 19, n.º 8, págs. 2801-2813, 2018.
- [23] D. Montalvo García et al., «Construcción de algoritmos de aprendizaje automático para predicción de consumos energéticos en edificios inteligentes», 2020.
- [24] M. de León y A. Gómez Corral. «Series temporales». (2020), dirección: <https://www.madrimasd.org/blogs/maticas/2020/06/01/148025> (visitado 20-04-2025).
- [25] R. Dey. «Time Series Decomposition». (2024), dirección: <https://medium.com/@roshmitadey/time-series-decomposition-62cbf31ab65e> (visitado 30-05-2025).
- [26] F. Capital. «Análisis de series temporales exploración del vínculo entre el análisis de series temporales y la precisión de las previsiones». (2024), dirección: <https://fastercapital.com/es/contenido/Analisis-de-series-temporales--exploracion-del-vinculo-entre-el-analisis-de-series-temporales-y-la-precision-de-las-previsiones.html> (visitado 20-04-2025).
- [27] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long y J. Wang, «Deep time series models: A comprehensive survey and benchmark», *arXiv preprint arXiv:2407.13278*, 2024.
- [28] V. Ylvén. «Why naive models are still relevant in the age of complex AI». (2024), dirección: <https://www.algorithmia.se/our-latest-thinking/why-naive-models-are-still-relevant-in-the-age-of-complex-ai#:~:text=Naive%20models%20are%20significantly%20simpler,and%20gain%20valuable%20insights%20sooner.> (visitado 20-04-2025).
- [29] J. Herzen, F. LÃssig, S. G. Piazzetta et al., «Darts: User-Friendly Modern Machine Learning for Time Series», *Journal of Machine Learning Research*, vol. 23, n.º 124, págs. 1-6, 2022. dirección: <http://jmlr.org/papers/v23/21-1177.html>.
- [30] M. S. Sardaña, «Modelo predictivo de venta cruzada en productos de Vida y Salud: Random Forest vs XGBoost», *no publication details provided*, 2022.
- [31] D. Gunay. «Random Forest Regression». (2020), dirección: <https://medium.com/@denizgunay/random-forest-af5bde5d7e1e> (visitado 20-04-2025).
- [32] F. Omarzai. «XGBoost Regression in Depth». (2024), dirección: <https://medium.com/@fraidoonomarzai99/xgboost-regression-in-depth-cb2b3f623281> (visitado 20-04-2025).
- [33] C. A. Meneses Agudo et al., «Análisis y predicción de series temporales provenientes de un sistema SCADA de una planta de fabricación industrial», 2019.
- [34] C. Arana, «Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales», *Serie Documentos de Trabajo, inf. téc.*, 2021.

- [35] J. Amat Rodrigo y F. Carazo. «Deep Learning para la predicción de series temporales: Redes Neuronales Recurrentes (RNN) y Long Short-Term Memory (LSTM)». (2024), dirección: <https://cienciadedatos.net/documentos/py54-forecasting-con-deep-learning> (visitado 20-04-2025).
- [36] D. Rubio Adeva, «Red neuronal: análisis de series temporales de entidades bancarias», 2021.
- [37] J. R. Guerrero Meza y B. E. Renteros Parra, «Predicción de la demanda eléctrica de los edificios de la Facultad de Derecho y Edificio E de la UDEP mediante el uso de redes neuronales LSTM y TCN»,
- [38] F. Lässig. «Temporal Convolutional Networks and Forecasting». (2021), dirección: <https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/> (visitado 20-04-2025).
- [39] A. Zeng, M. Chen, L. Zhang y Q. Xu, «Are transformers effective for time series forecasting?», en *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, págs. 11 121-11 128.
- [40] F. Peng, X. Ji, L. Zhang, J. Wang, K. Zhang y W. Wu, «Interpretable Mixture of Experts for Decomposition Network on Server Performance Metrics Forecasting.», *Electronics (2079-9292)*, vol. 13, n.º 20, 2024.
- [41] B. Lim, S. Ö. Arik, N. Loeff y T. Pfister, «Temporal fusion transformers for interpretable multi-horizon time series forecasting», *International Journal of Forecasting*, vol. 37, n.º 4, págs. 1748-1764, 2021.
- [42] N. V. Muñoz, «Previsión del Precio SPOT en el Mercado Eléctrico Español con Redes Neuronales», 2024.
- [43] N. Ovchinnikov. «Non-Autoregressive Models Hideout». (2024), dirección: <https://www.getpeech.com/blog/non-autoregressive-models-hideout> (visitado 10-05-2025).
- [44] N. V. Otten. «Teacher Forcing In Recurrent Neural Networks (RNNs): An Advanced Concept Made Simple». (2023), dirección: <https://spotintelligence.com/2023/10/12/teacher-forcing-in-recurrent-neural-networks-rnns-an-advanced-concept-made-simple/> (visitado 01-05-2025).
- [45] L. Mao. «Autoregressive Model and Autoregressive Decoding for Sequence to Sequence Tasks». (2021), dirección: <https://leimao.github.io/blog/Autoregressive-Model-Autoregressive-Decoding/> (visitado 01-05-2025).
- [46] S. Cristea y A. Rueda Ferreiro, «Control predictivo no lineal aplicado a columnas de destilación», *2ª Reunión de Usuarios de EcosimPro, UNED, Madrid*, págs. 24-25, 2003.
- [47] Savree. «Distillation Column Explained». (2022), dirección: <https://www.savree.com/en/encyclopedia/distillation-column> (visitado 04-05-2025).
- [48] S. Miller. «The Ultimate Guide to Distillation and Distillation Columns». (2024), dirección: <https://www.gustawater.com/blog/distillation-column.html> (visitado 04-05-2025).

- 
- [49] T. pandas development team, *pandas-dev/pandas: Pandas*, ver. v2.2.3, sep. de 2024. DOI: 10.5281/zenodo.13819579. dirección: <https://doi.org/10.5281/zenodo.13819579>.
- [50] C. R. Harris, K. J. Millman, S. J. van der Walt et al., «Array programming with NumPy», *Nature*, vol. 585, n.º 7825, págs. 357-362, sep. de 2020. DOI: 10.1038/s41586-020-2649-2. dirección: <https://doi.org/10.1038/s41586-020-2649-2>.
- [51] J. D. Hunter, «Matplotlib: A 2D graphics environment», *Computing in Science & Engineering*, vol. 9, n.º 3, págs. 90-95, 2007. DOI: 10.1109/MCSE.2007.55.
- [52] M. L. Waskom, «seaborn: statistical data visualization», *Journal of Open Source Software*, vol. 6, n.º 60, pág. 3021, 2021. DOI: 10.21105/joss.03021. dirección: <https://doi.org/10.21105/joss.03021>.
- [53] T. Akiba, S. Sano, T. Yanase, T. Ohta y M. Koyama, «Optuna: A Next-generation Hyperparameter Optimization Framework», en *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [54] P. Buestán, P. Carrión, A. Chamba y J. Pazmiño, «Una evaluación integral de las técnicas de ia para predecir el índice de calidad del aire: RNN y transformers», 2025.
- [55] N. Arrijoja Landa Cosio. «Métricas en regresión». (2021), dirección: <https://medium.com/@nicolasarrijoja/m%C3%A9tricas-en-regresi%C3%B3n-5e5d4259430b> (visitado 11-05-2025).



# **Anexos**



# Apéndice A

## Gráficas adicionales

En este anexo se proporcionan visualizaciones complementarias correspondientes a los resultados obtenidos con la implementación de los distintos modelos. Si bien en el Capítulo 6 (Resultados) solo se muestra un ejemplo representativo por modelo, aquí se incluyen tres ventanas adicionales de 45 minutos para cada arquitectura. Se trata de intervalos de tiempo a lo largo del conjunto de *test*, siendo los mismos para todos los modelos.

El objetivo es analizar sus comportamientos en distintos momentos del proceso, evaluando su capacidad de predicción en varias situaciones y reforzando las conclusiones extraídas.

### A.1. XGBoost

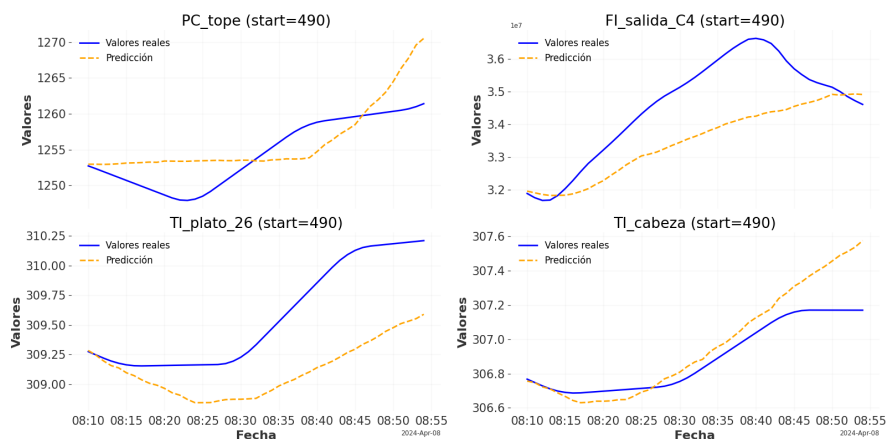


Figura A.1: Primera ventana de 45 minutos adicional de XGBoost

## Capítulo A. Gráficas adicionales

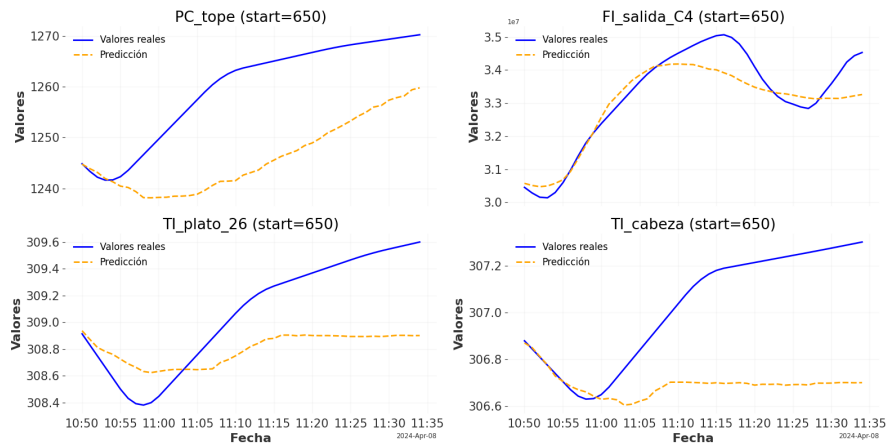


Figura A.2: Segunda ventana de 45 minutos adicional de XGBoost

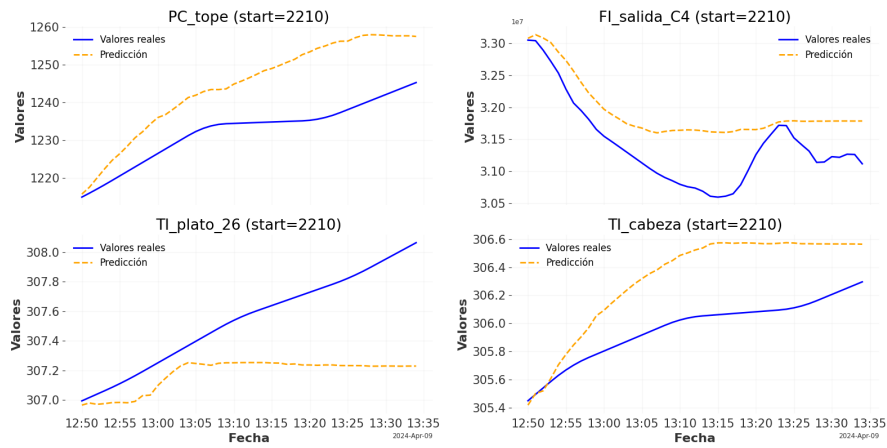


Figura A.3: Tercera ventana de 45 minutos adicional de XGBoost

## A.2. LSTM

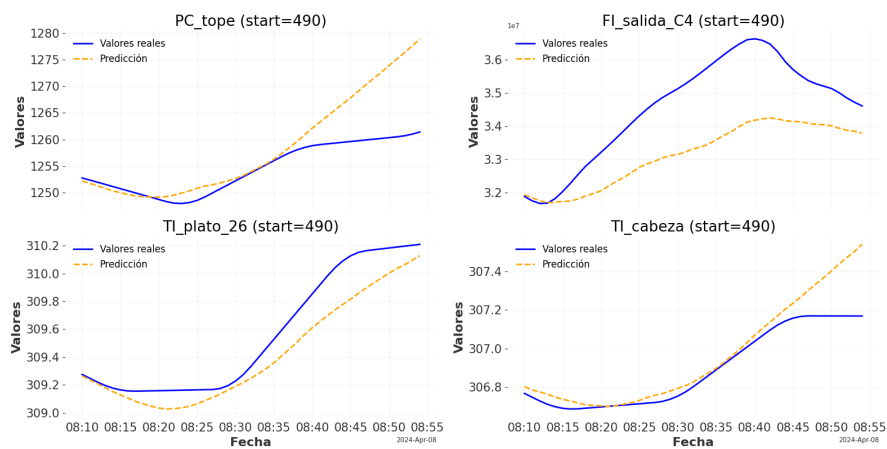


Figura A.4: Primera ventana de 45 minutos adicional de LSTM

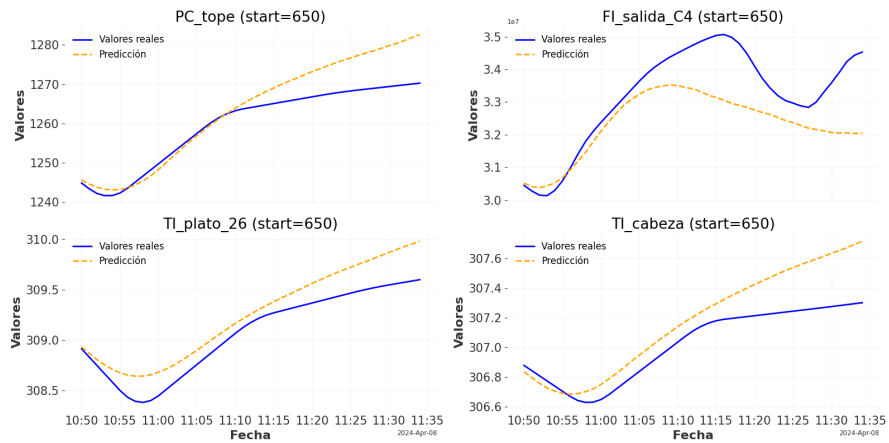


Figura A.5: Segunda ventana de 45 minutos adicional de LSTM

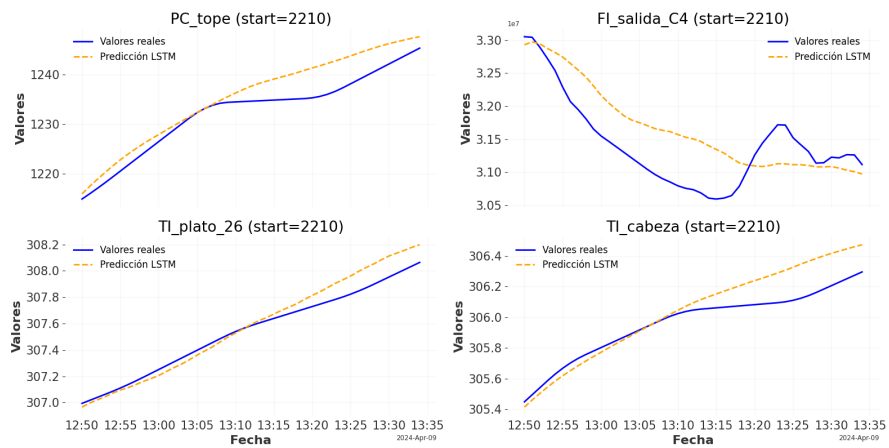


Figura A.6: Tercera ventana de 45 minutos adicional de LSTM

### A.3. TCN

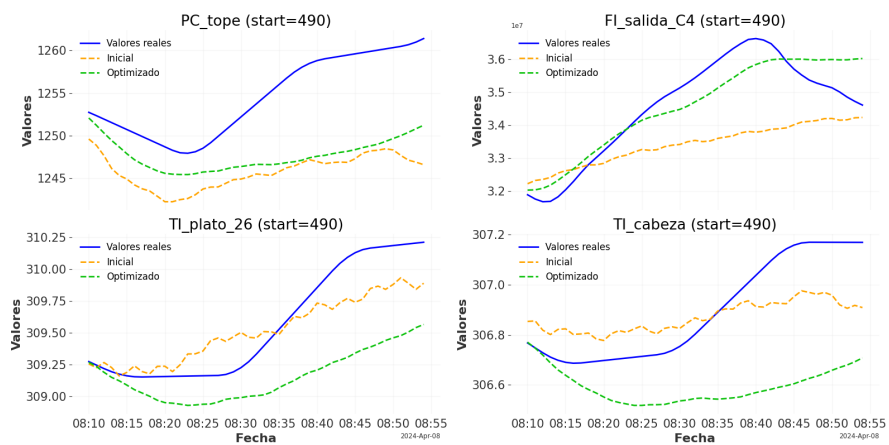


Figura A.7: Primera ventana de 45 minutos adicional de TCN

## Capítulo A. Gráficas adicionales

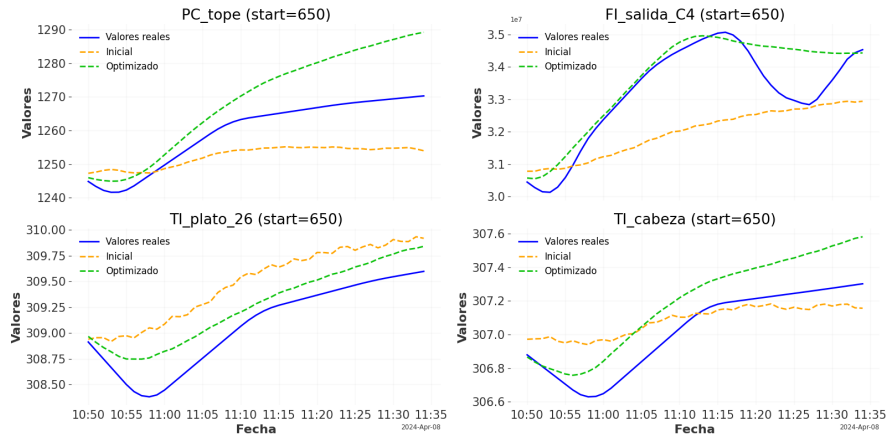


Figura A.8: Segunda ventana de 45 minutos adicional de TCN

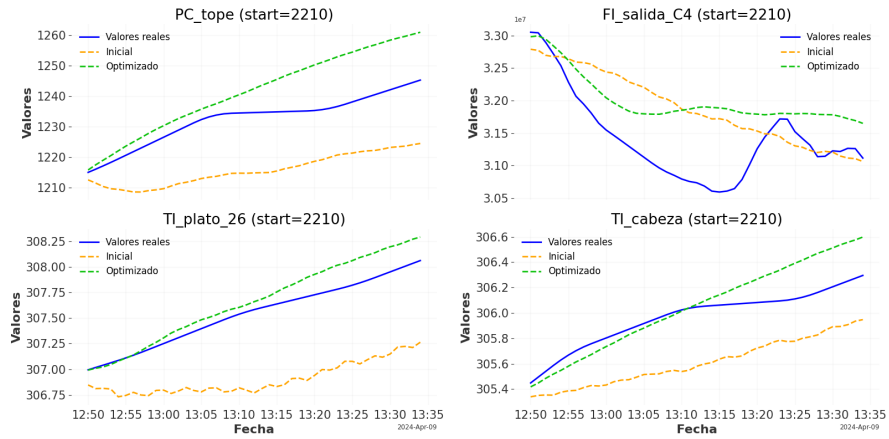


Figura A.9: Tercera ventana de 45 minutos adicional de TCN

### A.4. DLinear y NLinear

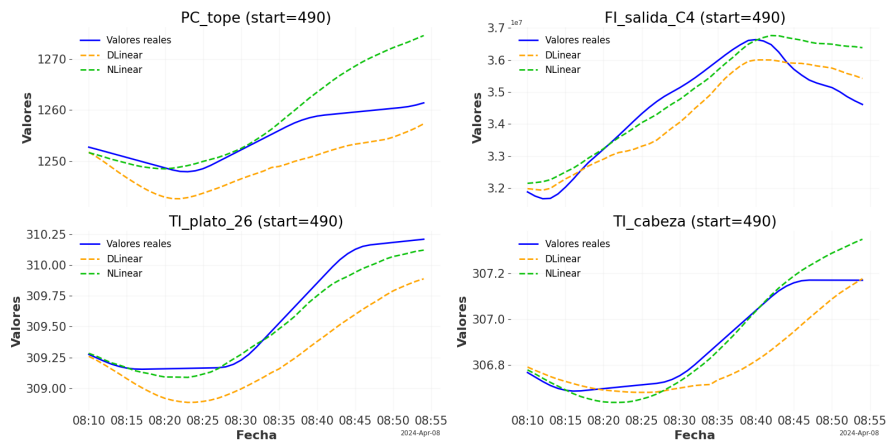


Figura A.10: Primera ventana de 45 minutos adicional de DLinear y NLinear

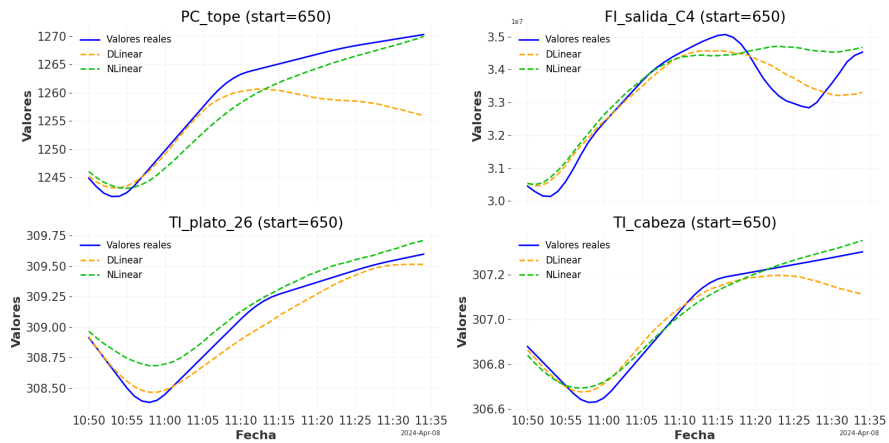


Figura A.11: Segunda ventana de 45 minutos adicional de DLinear y NLinear

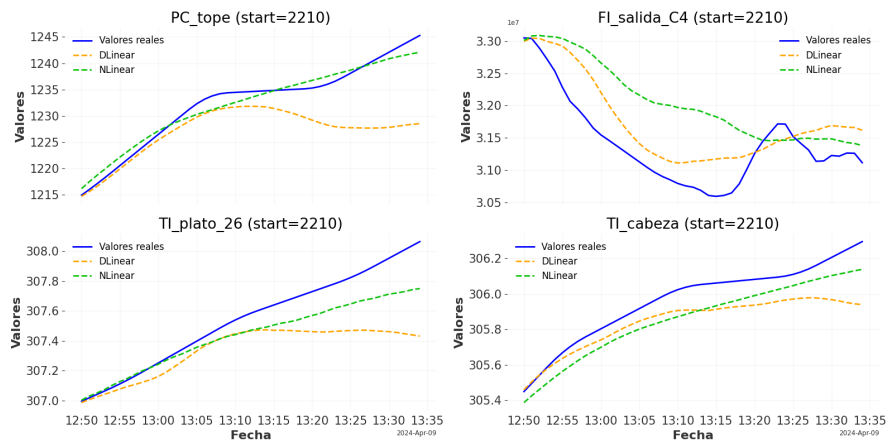


Figura A.12: Tercera ventana de 45 minutos adicional de DLinear y NLinear

**A.5. TFT**

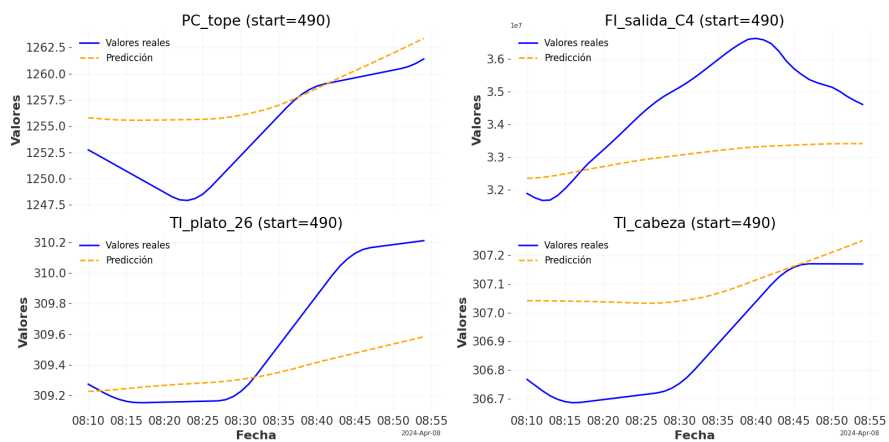


Figura A.13: Primera ventana de 45 minutos adicional de TFT

## Capítulo A. Gráficas adicionales

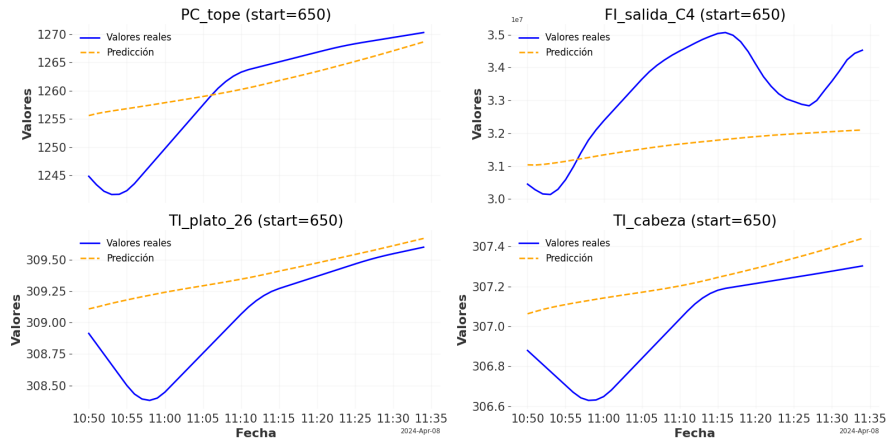


Figura A.14: Segunda ventana de 45 minutos adicional de TFT

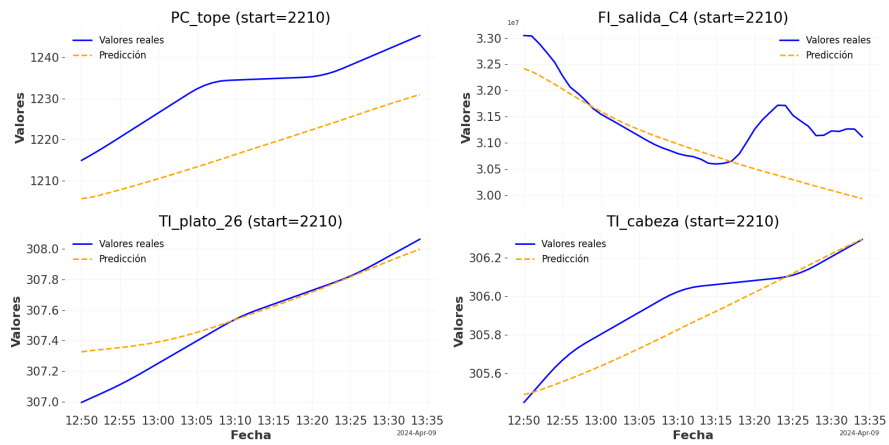



Figura A.15: Tercera ventana de 45 minutos adicional de TFT

Este documento esta firmado por

	<b>Firmante</b>	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	<b>Fecha/Hora</b>	Tue Jun 03 23:09:31 CEST 2025
	<b>Emisor del Certificado</b>	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	<b>Numero de Serie</b>	561
	<b>Metodo</b>	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)