



# DESARROLLO DE UN GEMELO DIGITAL DE MEDICIONES MEDIOAMBIENTALES

PROYECTO FIN DE GRADO

GRADO EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

AUTOR:

ANA MADRID SERRANO

TUTORES:

FRANCISCO SERRADILLA (UPM)

LUIS REDONDO (MTP)

FECHA:

05/04/2024

# INDICE

INDICE DE FIGURAS .....	4
RESUMEN .....	6
ABSTRACT .....	7
1. Revisión de la Literatura .....	8
1.1 Definición de Gemelos Digitales .....	8
1.2 Antecedentes Históricos .....	8
1.3 Tipos de Gemelos Digitales .....	10
1.4 Aplicaciones de Gemelos Digitales .....	13
1.5 Tecnologías y Herramientas clave .....	14
1.6 Desafíos y Limitaciones .....	16
1.7 Casos de estudio y ejemplos destacados .....	19
1.8 Experiencias previas con Gemelos Digitales en dispositivos de mediciones medioambientales .....	21
1.9 Futuro de los Gemelos Digitales .....	22
2. Marco Teórico .....	24
2.1 Componentes del Gemelo Digital .....	24
2.2 Herramientas y Tecnologías Relevantes .....	24
2.3 Definición de modelos explorados .....	27
2.4 Definición de métricas de validación empleadas .....	28
3. Metodología .....	32
3.1 Etapas del desarrollo .....	32
3.2 Arquitectura del software .....	33
3.3 Exploración de modelos .....	34
4. Recopilación de datos .....	41
4.1 Configuración del sensor .....	41
4.2 Conexión sensor base de datos .....	43
5. Implementación .....	46
5.1 Bases de datos .....	46
5.2 Procesar los datos .....	47

5.3 Predicciones .....	47
5.4 Graficar .....	50
5.5 Conexión con el sensor real .....	51
5.6 Transformer .....	51
5.7 Clases sensores.....	51
5.8 Sistema del gemelo digital .....	51
5.9 Interfaz de usuario .....	53
6. Validación y verificación.....	58
6.1 Comparación con el objeto físico .....	58
6.2 Datos de entrada .....	58
6.3 Modelo de simulación .....	59
6.4 Predicciones futuras .....	68
6.5 Sincronización en tiempo real .....	74
6.6 Seguridad cibernética .....	75
7. Conclusiones .....	77
8. Consideraciones sociales, económicas y medio ambientales.....	79
9. Trabajos futuros .....	80
10. Referencias.....	81

# INDICE DE FIGURAS

FIGURA 1: Primer gemelo digital (Apolo XIII) .....	9
FIGURA 2: Ejemplo de gemelo digital de sistema.....	10
FIGURA 3: Ejemplo de gemelo digital de producto .....	11
FIGURA 4: Ejemplo de gemelo digital de ciudad .....	12
FIGURA 5 : Ejemplo de gemelo digital de paciente.....	13
FIGURA 6:Ejemplo de gemelo digital de hospital .....	12
FIGURA 7: Ejemplo de gemelo digital de energía .....	13
FIGURA 8: Representación de la unión de IoT, IA, RV y Cloud.....	16
FIGURA 9: Diagrama de clases UML simplificado.....	34
FIGURA 10: Grafica de las predicciones de temperatura usando walk forward .....	37
FIGURA 11: Grafica de las predicciones de temperatura usando RNN .....	38
FIGURA 12: Grafica de las predicciones de temperatura usando transformers.....	40
FIGURA 13: Visualización de la configuración de Node-RED.....	44
FIGURA 14: Visualización de la aplicación de InfluxDB .....	44
FIGURA 15: Representación del sistema de recolección de datos .....	45
FIGURA 16: Interfaz de usuario al conectarse con el sensor real.....	53
FIGURA 17: Desplegable de la interfaz para seleccionar el tipo.....	55
FIGURA 18 Desplegable de la interfaz para seleccionar la función .....	54
FIGURA 19: Aparición del cuadro de fecha.....	56
FIGURA 20: Interfaz de usuario para predicciones futuras .....	55
FIGURA 21: Grafico de humedad generado al seleccionar comparación de predicciones .....	55
FIGURA 22: Grafico de temperatura al seleccionar tendencia .....	56
FIGURA 23: Apartados al seleccionar simulateSensor .....	56
FIGURA 24: Valores ingresados para simulateSensor.....	57
FIGURA 25: Suscriptor MQTT para los valores ingresados .....	57
FIGURA 26: Grafica de temperatura predicciones frente a test.....	61
FIGURA 27: Grafica de humedad predicciones frente a test.....	63
FIGURA 28: Grafica de presión predicciones frente a test .....	65
FIGURA 29: Grafica de luminosidad predicciones frente a test .....	68
FIGURA 30: Grafica de temperatura para el modelo1 con sus predicciones a futuro ...	69
FIGURA 31: Grafica de temperatura test frente a predicciones .....	70
FIGURA 32: Grafica de temperatura para el modelo2 con sus predicciones a futuro ...	70
FIGURA 33: Comparación datos de validación predicciones futuras temperatura .....	72
FIGURA 34: Comparación datos de validación predicciones futuras humedad .....	73
FIGURA 35: Comparación datos de validación predicciones futuras presión .....	74



## **RESUMEN**

En este proyecto realizado junto a la empresa MTP, desarrollaremos un gemelo digital para un sensor de mediciones medioambientales que captura datos de temperatura, humedad, presión y luminosidad. A través de un sensor físico, recolectaremos datos reales para alimentar y comparar con nuestro gemelo digital. Analizaremos y aplicaremos modelos de machine learning y deep learning para predecir futuras condiciones medioambientales basándonos en los datos históricos obtenidos. Además, crearemos una aplicación de software que dará funcionalidad a nuestro gemelo digital, incluyendo distintas características y herramientas para la interacción del usuario. Para integrar los datos del sensor real con nuestro sistema digital, estableceremos una infraestructura que recopile, almacene y haga accesibles los datos en una base de datos, la cual estará conectada directamente con el gemelo digital. Esta infraestructura permitirá una interacción fluida a través de una interfaz de usuario, facilitando el acceso a la información y la gestión de las mediciones ambientales.

# **ABSTRACT**

In this project carried out together with the company MTP, we will develop a digital twin for an environmental measurement sensor that captures data on temperature, humidity, pressure, and luminosity. Through a physical sensor, we will collect real data to feed and compare with our digital twin. We will analyze and apply machine learning and deep learning models to predict future environmental conditions based on the historical data obtained. Additionally, we will create a software application that will provide functionality to our digital twin, including various features and tools for user interaction. To integrate the data from the real sensor with our digital system, we will establish an infrastructure that collects, stores, and makes the data accessible in a database, which will be directly connected to the digital twin. This infrastructure will allow for smooth interaction through a user interface, facilitating access to information and management of environmental measurements.

# **1. Revisión de la Literatura**

## **1.1 Definición de Gemelos Digitales**

Se define un gemelo digital como una representación virtual precisa y dinámica de un sistema físico, proceso o entidad. Este modelo digital abarca tanto la estructura como el comportamiento del objeto de estudio, permitiendo una simulación detallada y en tiempo real. Se destaca por su versatilidad y aplicabilidad en diversas áreas, brindando una herramienta poderosa para comprender, analizar y optimizar sistemas complejos.

La relevancia de los gemelos digitales se manifiesta en su capacidad para proporcionar un análisis predictivo en tiempo real. Su aplicación va más allá de un campo específico, abriendo oportunidades para mejorar la eficiencia y la toma de decisiones informadas en diversos contextos.

## **1.2 Antecedentes Históricos**

El desarrollo de los gemelos digitales se remonta a mucho antes de lo que creemos, muchos creen que empezaron a evolucionar en 2002, pero fue la NASA los que emplearon el concepto mucho antes en 1960.

La empresa espacial creó un gemelo digital que replicaba y evaluaba las condiciones del Apollo 13. Cuando se lanzó la nave, hubo una explosión en un tanque de oxígeno, el fallo se tuvo que resolver desde miles de kilómetros, pero por suerte tenían el gemelo digital físico, lo que evitó una catástrofe al poder regresar a la Tierra con éxito.



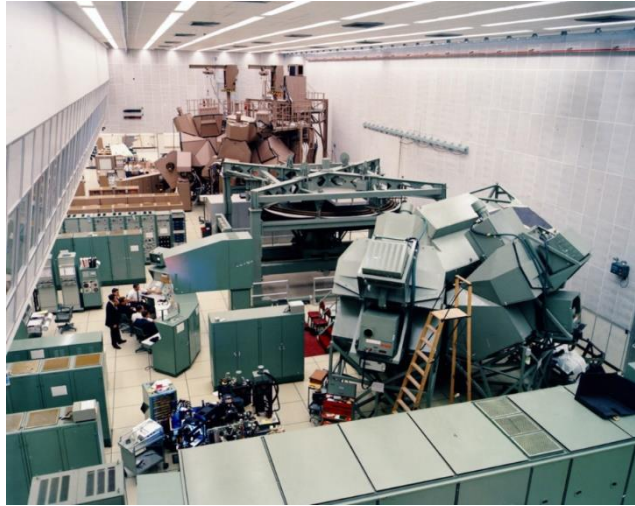


FIGURA 1: Primer gemelo digital (Apolo XIII)

La palabra de gemelo digital se mencionó por primera vez en 1998, y se refería a una copia de la voz del actor Alan Alda. Desde 2002 se han empezado a conocer, y en 2011 el termino se expandió, y se refirió a él como “Information Mirroring Model” en el libro “Virtually Perfect: Driving Innovation and Lean Products through Product Lifecycle Management”. Pero no ha sido hasta 2017 que se han convertido en una de las mayores estrategias tecnológicas, ya que la evolución de Internet of Things (IoT), el big data, y sensores de tiempo real, han abaratado significativamente los costes y han permitido un mejor desarrollo.

El crecimiento de Internet of Things (IoT), big data y sensores en tiempo real ha tenido un impacto significativo en el desarrollo y la viabilidad de los gemelos digitales. Aquí hay una comparación de cómo estos avances tecnológicos han influido en los gemelos digitales.

Respecto al IoT ha sido un habilitador clave. La capacidad de conectar dispositivos y sensores a la red y recopilar datos en tiempo real desde el mundo físico ha permitido su creación de una manera más precisa y actualizada. La información recopilada a través de IoT se utiliza para alimentar los modelos digitales y mantenerlos sincronizados con sus contrapartes físicas. Esto ha mejorado la capacidad de monitoreo y control, lo que a su vez ha impulsado la eficiencia y la toma de decisiones en tiempo real.

Gracias al auge del big data, se ha permitido el procesamiento y análisis de grandes volúmenes de datos, lo que es esencial para el funcionamiento de los gemelos digitales. Estos dependen de datos detallados y en tiempo real para reflejar con precisión el

comportamiento de los sistemas físicos. El big data ha facilitado la recopilación, almacenamiento y análisis de datos necesarios para impulsar los gemelos digitales y mejorar su capacidad de modelado y predicción.

Los sensores en tiempo real son los "ojos y oídos" de los gemelos digitales. Proporcionan datos continuos sobre las condiciones físicas y el rendimiento de los objetos en el mundo real. La evolución de los sensores ha mejorado la calidad y la frecuencia de los datos recopilados, lo que a su vez ha aumentado la precisión de los gemelos digitales. Esto es esencial para la toma de decisiones en tiempo real y la detección de problemas antes de que se conviertan en crisis.

### 1.3 Tipos de Gemelos Digitales

Existen múltiples tipos de gemelos digitales dependiendo de la finalidad que le quieras otorgar y las necesidades del proyecto. Los principales tipos son:

Los gemelos digitales industriales representan una revolución tecnológica en la gestión y optimización de procesos productivos. Estos modelos digitales son réplicas virtuales de sistemas físicos, maquinaria o procesos industriales, permitiendo simulaciones precisas y análisis en tiempo real. Facilitan la predicción de fallos, el mantenimiento preventivo, y la mejora continua mediante la recolección y análisis de datos. Su implementación conduce a una mayor eficiencia, reducción de costos y tiempos de inactividad, y una mejor toma de decisiones basada en datos. Los gemelos digitales se convierten en una herramienta esencial para la Industria 4.0, impulsando la innovación y la sostenibilidad en el sector industrial.

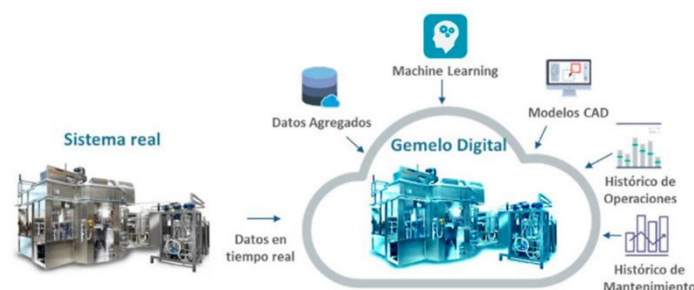


FIGURA 2: Ejemplo de gemelo digital de sistema

Los gemelos digitales de productos son modelos virtuales que replican las características, el comportamiento y la evolución de un producto físico a lo largo de su ciclo de vida. Estos modelos permiten a diseñadores, ingenieros y gestores de producto simular, analizar y optimizar el producto desde su concepción hasta su desecho, pasando por la fabricación, el uso y el mantenimiento. Facilitan la experimentación en un entorno virtual seguro, reduciendo los costes y tiempos asociados a prototipos físicos y pruebas reales. Al integrar datos en tiempo real y retroalimentación de los usuarios, los gemelos digitales de productos mejoran la toma de decisiones, la personalización y la innovación, convirtiéndose en una herramienta clave para el desarrollo ágil y la mejora continua de productos en diversas industrias.



*FIGURA 3:* Ejemplo de gemelo digital de producto

Los gemelos digitales de ciudades son representaciones virtuales avanzadas de áreas urbanas, que combinan datos en tiempo real con modelos 3D para simular y analizar el funcionamiento y la evolución de las ciudades. Estos modelos permiten a urbanistas, gobiernos y empresas monitorear y gestionar infraestructuras, tráfico, servicios públicos, y recursos energéticos de manera eficiente. Al integrar información de diversas fuentes, como sensores IoT, sistemas de información geográfica (GIS) y datos demográficos, los gemelos digitales urbanos facilitan la planificación urbana, la respuesta a emergencias, la sostenibilidad y la participación ciudadana. Se convierten en herramientas clave para la toma de decisiones basadas en datos, ayudando a crear ciudades más inteligentes, resilientes y adaptadas a las necesidades futuras.



FIGURA 4: Ejemplo de gemelo digital de ciudad

Los gemelos digitales en el sector de la salud abarcan tanto a modelos virtuales personalizados de pacientes como a representaciones digitales de centros de salud y hospitales. Estos modelos ofrecen una simulación detallada y análisis del estado físico y biológico de los pacientes, así como la operatividad y gestión de las instalaciones de salud. Permiten diagnósticos precisos, tratamientos personalizados y mejor planificación y administración de recursos sanitarios. La integración de datos médicos, genéticos, comportamentales y operativos mejora la eficiencia de los servicios de salud, optimiza la gestión de inventarios, y facilita la programación de procedimientos. Los gemelos digitales en salud son cruciales para la investigación, el desarrollo de medicamentos, y la implementación de estrategias preventivas, promoviendo un enfoque holístico que incluye tanto la atención personalizada al paciente como la optimización de los procesos hospitalarios, impulsando así la transformación hacia sistemas de salud más avanzados y centrados en el paciente.



FIGURA 5 : Ejemplo de gemelo digital de paciente



FIGURA 6: Ejemplo de gemelo digital de hospital

Los gemelos digitales energéticos son modelos virtuales avanzados que replican sistemas y redes de energía, desde plantas de generación hasta infraestructuras de distribución y consumo. Estos modelos permiten simular y analizar el flujo de energía, la eficiencia de los sistemas, la gestión de la demanda, y el impacto de las energías renovables en tiempo

real. Al integrar datos operacionales, climáticos y de consumo, los gemelos digitales energéticos facilitan la optimización de la producción, la distribución de energía más eficiente y la reducción de las emisiones de carbono. Son herramientas clave para la transición hacia una economía baja en carbono, permitiendo a operadores y gestores de redes adaptarse a las fluctuaciones de la demanda y a la integración de fuentes renovables. Además, promueven la innovación en el diseño de sistemas energéticos más resilientes y sostenibles, mejorando la planificación estratégica y la toma de decisiones para enfrentar los retos del cambio climático y la seguridad energética.



FIGURA 7: Ejemplo de gemelo digital de energía

## 1.4 Aplicaciones de Gemelos Digitales

Los gemelos digitales son una gran herramienta, pero su uso no siempre es adecuado para todos los usos. Muchas veces no vale la pena a nivel económico, invertir las grandes cantidades de dinero que conlleva tener una réplica exacta digital, ya que esto puede encarecer el precio final del producto. No obstante, hay multitud de aplicaciones que se benefician de los GD. Algunos de estos ejemplos son, proyectos que físicamente son muy grandes como pueden ser puentes, monumentos. En general grandes obras de ingeniería con reglas muy estrictas y precisas.

Un gemelo digital en la industria y la fabricación permite diseñar, probar y simular productos, como motores y turbinas, antes de su producción física. También facilita la predicción de reparaciones, tiempos de mantenimiento y el rendimiento de piezas. Además, se puede aplicar a la simulación de procesos de fabricación, reduciendo costos, optimizando operaciones y detectando posibles fallos. En resumen, esta tecnología ahorra recursos y mejora la eficiencia en la creación de productos y procesos industriales.

En el sector de la salud, los gemelos digitales son esenciales. Pueden representar digitalmente hospitales, proporcionando información en tiempo real sobre pacientes y flujos de trabajo. Los sensores monitorean pacientes, permiten la coordinación del personal y mejoran el análisis de procesos. Los gemelos digitales reducen los tiempos de espera en emergencias, optimizan el flujo de pacientes y disminuyen costos operativos. Además, mejoran la experiencia del paciente al garantizar una atención más eficiente y oportuna.

Las ciudades inteligentes están adoptando la tecnología de gemelos digitales para el control del tráfico, la planificación urbana y la gestión de infraestructuras como torres de redes móviles y placas solares. Ejemplo, el Departamento de Transporte de Los Ángeles utiliza un gemelo digital basado en datos de transporte para administrar redes de bicicletas y electricidad, y planea expandir su uso para gestionar viajes compartidos y taxis drones en el futuro. Esto demuestra cómo los gemelos digitales están transformando la eficiencia y la planificación en las ciudades inteligentes.

## **1.5 Tecnologías y Herramientas clave**

Los GD siguen tres procesos principales, la adquisición de datos, la modelización de datos y la aplicación de estos datos recogidos y posteriormente procesados. Para estos aspectos se utilizan cuatro tecnologías que han crecido exponencialmente en los últimos años, lo que ha permitido la rápida evolución de los gemelos. Las tecnologías mencionadas incluyen Internet de las Cosas(IoT), Inteligencia Artificial(IA), Realidad Extendida (XR) y la Nube.

-El Internet de las Cosas consta de una red de dispositivos conectados y la tecnología que permite la comunicación entre estos y la nube. Los GD precisan de la tecnología de IoT para obtener la información del mundo real y pasarla a la digital. Los datos son almacenados y es posible ver cómo actúan en tiempo real.

-La Inteligencia Artificial es la que se ocupa de resolver problemas asociados a la inteligencia humana como puede ser el aprendizaje, la resolución de problemas o reconocimiento de patrones. Dentro de la IA tenemos el Machine Learning(ML) que trata de una técnica que desarrolla modelos estadísticos y algorítmicos, de esta manera la maquina es capaz de realizar múltiples tareas basándose en patrones detectados sin necesidad de recibir instrucciones. En lo que a GD acontece utiliza estos algoritmos del

ML para procesar los datos que le llegan de los sensores de IoT y poder identificar patrones.

-Cuando hablamos de la Realidad Virtual podemos ver que se puede confundir fácilmente con los GD, pero hay algunas diferencias clave. La RV es más usada para diseñar y representar cosas sin ningún cambio. Sin embargo, los gemelos son entornos virtuales complejos que van cambiando en tiempo real.

Un ejemplo para ayudarnos a visualizar esto podría ser cuando nos subimos a una simulación de una nave espacial, estamos ante una simulación inmersiva, pero los escenarios no están vinculados a una nave real. En cambio, un GD sabe todos los componentes y acciones de esa nave y los posibles problemas que se pueden plantear.

Aunque como hemos dicho estas dos tecnologías no son las mismas, la RV desempeña un papel importante en el desarrollo de los GD ya que proporciona una plataforma inmersiva y altamente interactiva para crear, visualizar y simular objetos, sistemas y entornos virtuales que se asemejen a sus contrapartes en el mundo real.<sup>3</sup>

-La computación en la nube es una poderosa herramienta de ayuda para el desarrollo, esto se debe a que los GD requieren de grandes cantidades de datos. La nube proporciona almacenamiento escalable que permite almacenar y acceder a grandes volúmenes de datos de manera eficiente. También destaca su capacidad para proporcionar recursos escalables y acceso remoto a datos. Esto nos permite tener GD gestionados desde la nube lo que nos da un acceso instantáneo a una plataforma segura y continuamente actualizada para supervisar y probar las operaciones.





FIGURA 8: Representación de la unión de IoT, IA, RV y Cloud

## 1.6 Desafíos y Limitaciones

A la hora de desarrollar gemelos digitales nos topamos con ciertos desafíos tanto económicos, tecnológicos y sociales.

En cuanto a los técnicos nos encontramos con la falta de protocolos de comunicación lo que dificulta la integración de sistemas y fuentes de datos heterogéneas. Se espera que los GD crezcan en tamaño y complejidad, lo que requerirá gran cantidad de recursos computacionales, lo que supone un desafío para mantener un entorno de alto rendimiento.

Sin ir a problemas más técnicos, uno de los principales es garantizar un reflejo preciso del mundo real y producir resultados confiables. Esto requiere modelado preciso, validación continua y actualización de los datos.

Otro problema que concierne al sector es la seguridad, ya que estos sistemas pueden ser vulnerables a ataques cibernéticos lo que compromete la integridad de los datos.

Hablando de los problemas económicos los gemelos digitales implican una gran inversión tanto de tecnología, software y recursos humanos. El mantenimiento y actualización regular también supone grandes gastos, y requiere de gran trabajo asegurarse de que siguen siendo fieles a la realidad con el tiempo.

Determinar si un GD va a retornar la inversión es muy complicado, por lo que conseguir inversores también lo es.



Cuando hablamos de nuevas tecnologías siempre aparecen problemas éticos y los gemelos digitales no son una excepción. Nos enfrentamos a desafíos sobre el uso responsable de los datos, considerar la privacidad de los usuarios cuyos datos se usen. Controlar los algoritmos usados para evitar sesgos en los entrenamientos que impliquen una desigualdad. Las decisiones sobre la toma de decisiones y el empleo de los datos debe ser transparentes para poder garantizar la confianza de la comunidad.

Actualmente la tecnología se encuentra frente a un desafío que comparten tanto las tecnologías de IA e IoT. Estos problemas van desde aspectos tales como la estandarización de datos, la gestión de estos y su seguridad. También podemos incluir en los desafíos las barreras para su implementación y la transformación de sistemas heredados. Otro de los muros con el que nos encontramos es la necesidad de modernizar infraestructuras de TI antiguas, dificultades en la conectividad, la privacidad, la seguridad de los datos confidenciales, y la inexistencia de un enfoque de modelado estándar

Para sumar a todos los desafíos mencionados anteriormente, se presentan nuevos retos que probablemente ralentice el crecimiento del mercado de GD, tales como el elevado coste de su implementación, el aumento de la demanda energética. La complejidad de su arquitectura también supone un impedimento.

La implementación de GD resulta muy costosa, este se debe a que requiere una gran inversión en plataformas tecnológicas, el desarrollo de infraestructura, el control de calidad de datos y soluciones de seguridad. A todo esto, le sumamos el coste y la complejidad de mantener la infraestructura, lo que supone una alta inversión en operaciones.

La validación y verificación de los gemelos digitales son unos de los desafíos clave en su implementación y uso efectivo. La verificación y la validación (V&V) son procesos esenciales para garantizar la corrección y credibilidad de los modelos de gemelos digitales. La verificación implica la confirmación de que los modelos se construyeron según las especificaciones y requisitos establecidos, evaluando la lógica, funcionalidad y estructura, y corrigiendo cualquier error detectado. Por otro lado, la validación implica la comparación de los modelos con la realidad que representan, evaluando su precisión y fidelidad mediante la comparación de resultados y comportamientos con datos y observaciones reales. Ambos procesos son complementarios y cruciales para asegurar que los gemelos digitales sean confiables y útiles. La verificación se enfoca en la construcción

correcta de los modelos, mientras que la validación verifica su capacidad para representar con precisión el mundo real. Estos procedimientos ayudan a garantizar que los gemelos digitales sean herramientas efectivas para la toma de decisiones y simulaciones precisas. Aquí hay algunos desafíos específicos relacionados con la validación de los gemelos digitales:

La implementación exitosa de gemelos digitales presenta una serie de desafíos técnicos y operativos de validación que deben abordarse con cuidado. En primer lugar, la precisión de los datos de entrada es fundamental. Los gemelos digitales dependen en gran medida de datos precisos y actualizados de sensores y dispositivos para representar con precisión el mundo real. Garantizar la confiabilidad de estos datos es un desafío constante.

La calibración y el ajuste de sensores y dispositivos son esenciales para mantener la precisión del gemelo digital, pero esto puede ser complicado y requerir un mantenimiento continuo. Además, la validación experimental, que a menudo implica pruebas en el mundo real, es costosa y consume tiempo, pero es necesaria para verificar la precisión del gemelo digital.

Comparar el rendimiento del gemelo digital con datos históricos conocidos puede ser útil, pero puede ser insuficiente si las condiciones cambian con el tiempo. La incertidumbre y la variabilidad en los datos de entrada y el comportamiento del objeto físico agregan complejidad a la validación, y los modelos deben ser capaces de manejar estas incertidumbres.

Los sistemas físicos evolucionan con el tiempo debido al desgaste, el envejecimiento y otros factores, lo que requiere que el gemelo digital se mantenga actualizado y preciso. En entornos extremos, como la industria aeroespacial o la energía nuclear, la validación puede ser particularmente difícil debido a la falta de oportunidades para pruebas prácticas.

Modelar sistemas altamente complejos, como plantas de energía o sistemas de transporte, presenta desafíos adicionales debido a la interconexión de múltiples componentes y variables. Cumplir con normativas y estándares de la industria también es esencial, ya que los gemelos digitales deben cumplir con requisitos específicos.

Finalmente, el mantenimiento y la actualización continua son fundamentales para que los gemelos digitales sigan siendo precisos, pero esto puede ser costoso y requerir recursos significativos.

La validación adecuada es esencial para que los gemelos digitales sean herramientas confiables y efectivas en aplicaciones críticas. Abordar estos desafíos de validación requiere una combinación de tecnología avanzada, protocolos de prueba rigurosos y procesos de gestión de datos sólidos.

## **1.7 Casos de estudio y ejemplos destacados**

Vamos a abordar ciertos casos de estudio y ejemplos que los gemelos digitales se han implementado con éxito en distintas empresas y organizaciones.

En REPSOL está llevando a cabo un ambicioso programa de digitalización para transformar la empresa, impulsar un cambio cultural y desarrollar soluciones digitales para diversas áreas de negocio. Este programa es crucial para lograr los objetivos establecidos en la estrategia de REPSOL. El enfoque principal de esta transformación es el gemelo digital para Repsol E&P, que tiene como objetivo proporcionar herramientas digitales a las operaciones en campo y proyectos para mejorar la eficiencia de los activos y optimizar la producción

Un componente esencial de este gemelo digital es el proyecto de Gestión Automatizada de la Producción (APM) y Modelos Integrados de Flujo (IFM), desarrollado en colaboración con OVS Group como parte del Programa Digital Corporativo de REPSOL. El APM se centra en mejorar la eficiencia operativa y la producción diaria en un entorno integrado y estandarizado, utilizando datos de diversas fuentes dentro de los activos de REPSOL. Los flujos de trabajo de APM permiten a los operadores e ingenieros identificar, clasificar y dar seguimiento a oportunidades de optimización de manera centralizada.

En resumen, REPSOL está liderando un proceso de digitalización estratégica con un enfoque central en el gemelo digital, específicamente en el proyecto APM e IFM, que busca mejorar la eficiencia y la producción en toda la organización, lo que contribuirá significativamente a sus objetivos y metas.

Rolls-Royce, la renombrada empresa aeroespacial y de defensa, ha implementado con éxito tecnología de gemelos digitales para monitorear de manera precisa los motores que fabrica. Este enfoque avanzado permite un seguimiento detallado del rendimiento de cada motor en condiciones reales de vuelo, teniendo en cuenta las variables específicas de uso

y las necesidades individuales de mantenimiento. Stuart Hughes, director de Información y Digital de Rolls-Royce, destaca que este enfoque se ajusta a la vida útil real de cada motor en lugar de depender exclusivamente de las recomendaciones de los manuales.

Esta adaptación de gemelos digitales ha llevado a Rolls-Royce a ofrecer un servicio altamente personalizado a sus clientes, extendiendo significativamente los intervalos de mantenimiento en algunos motores hasta un 50%. Esto ha resultado en una drástica reducción del inventario de piezas de repuesto y ha contribuido a una mayor eficiencia de los motores, con un ahorro de 22 millones de toneladas de carbono hasta la fecha. Gracias a la implementación de gemelos digitales Rolls-Royce ha conseguido optimizar su enfoque de mantenimiento y mejorar su huella ambiental.

La empresa de confitería, cuidado de mascotas y alimentos, Mars, ha adoptado una estrategia digital innovadora al crear un gemelo digital de su cadena de suministro de fabricación lo que le permite procesar y analizar datos generados por las máquinas de producción en sus 160 instalaciones de fabricación.

La implementación de esta tecnología se ha centrado en mejorar la eficiencia operativa y el control de procesos. Mars utiliza el servicio Azure Digital Twins IoT de Microsoft en colaboración con consultores de operaciones y fabricación digital de Accenture. Esto ha resultado en simulaciones de software que optimizan la capacidad de producción y reducen los residuos al prevenir cantidades inconsistentes de productos en el envasado.

Además, el enfoque en gemelos digitales ha permitido a Mars crear una "tienda de aplicaciones de casos de uso" virtual que se puede aplicar en todas sus líneas comerciales. En el futuro, la empresa tiene planes de utilizar datos de gemelos digitales para tener en cuenta variables como el clima y otras consideraciones situacionales que afectan a sus productos, mejorando así la visibilidad de su cadena de suministro desde la producción hasta el consumidor final. Esta estrategia digital no se limita a ser una innovación tecnológica, sino que se ve como un catalizador para impulsar el negocio de Mars hacia un futuro más eficiente y orientado al cliente.

## **1.8 Experiencias previas con Gemelos Digitales en dispositivos de mediciones medioambientales**

Gemelos Digitales en Edificaciones: La creciente tendencia de utilizar gemelos digitales en edificaciones ha permitido la simulación y análisis de respuestas ante diversas circunstancias. En el ámbito de oficinas, por ejemplo, la gestión de variables ambientales es esencial para crear condiciones de confort óptimas. Sensores que miden temperatura, humedad, concentración de CO<sub>2</sub>, partículas en suspensión (PM<sub>10</sub> y PM<sub>2.5</sub>), compuestos orgánicos volátiles (COVs) y ozono supervisan estos factores. La integración de estos sensores con la representación digital del edificio, junto con datos relevantes como gasto energético o consumo de agua, posibilita la optimización del espacio de trabajo y la reducción de costos operativos.

Aunque tecnologías como Building Information Modeling (BIM) comparten objetivos similares, la distinción fundamental radica en la inclusión del factor humano por parte de los gemelos digitales. Estos no solo ofrecen información sobre subsistemas, sino que también responden a cuestiones relacionadas con el comportamiento previsible de los ocupantes frente a fallos en sistemas o necesidades de ventilación en eventos específicos.

Gemelos Digitales en Entornos Urbanos: En el ámbito urbano, la aplicación de gemelos digitales ha sido vital para abordar los desafíos relacionados con la calidad del aire. Su potencial para desarrollar políticas urbanas holísticas, considerando las interacciones entre los diferentes sistemas, se destaca.

Un ejemplo revelador proviene de Cambridge, donde se ha iniciado un proyecto de gemelo digital con el objetivo de reducir la congestión del tráfico y mejorar la calidad del aire. Este proyecto, precursor del National Digital Twin a nivel nacional, incorpora diversos datos, incluida la calidad del aire de múltiples fuentes. Además, se incluyen variables socioeconómicas y patrones de movilidad de las personas recurrentes en Cambridge. Este enfoque integral permite recrear escenarios, como inversiones en transporte, distribución del empleo y opciones de trabajo flexible, evidenciando el potencial de los gemelos digitales como herramientas de gobernanza para mejorar la calidad de vida urbana.

Aplicación en Optimización Energética: En el ámbito industrial, empresas como Semantic Systems emplean gemelos digitales como herramientas clave para anticiparse a

la generación y análisis de datos con el objetivo de optimizar la energía en una planta. A través de modelos y escenarios probables, estos gemelos digitales permiten la optimización de horarios de encendido y apagado del sistema de climatización (HVAC). Se generan modelos de predicción de la temperatura interna de la planta basados en datos históricos, y posteriormente, se crea un modelo prescriptivo para determinar un horario óptimo de encendido/apagado del sistema de climatización, contribuyendo así a la eficiencia energética.

## **1.9 Futuro de los Gemelos Digitales**

El futuro de los gemelos digitales es casi ilimitado, debido al hecho de que se dedican constantemente cantidades cada vez mayores de potencia cognitiva a su uso. Por lo tanto, los gemelos digitales están constantemente aprendiendo nuevas habilidades y capacidades, lo que significa que pueden continuar generando la información necesaria para mejorar los productos y tornar los procesos más eficientes.

La pandemia de COVID-19 ha impulsado significativamente el crecimiento del mercado de Gemelos Digitales en diversas aplicaciones. Esto ha generado perspectivas de crecimiento prometedoras, con varios países planeando implementar soluciones de Gemelos Digitales como parte de sus esfuerzos de reforma económica.

Las tecnologías subyacentes a los gemelos digitales, como el IoT, la IA, el big data y la computación en la nube, continúan evolucionando constantemente. Esto sugiere que los Gemelos Digitales también evolucionarán en paralelo, y su potencial real aún no se ha realizado por completo. Además, muchas organizaciones están recurriendo a la tecnología de gemelos digitales para optimizar sus cadenas de suministro y procesos operativos como parte de su recuperación económica postpandemia.

La aparición de la Industria 4.0 y el IoT ha acelerado aún más la adopción de la tecnología de gemelos digitales en diversas aplicaciones. La Industria 4.0 utiliza tecnologías avanzadas como la computación en la nube, el IoT, la inteligencia artificial y la impresión 3D, con los gemelos digitales desempeñando un papel central en estas iniciativas. Cada vez más industrias están aprovechando las soluciones de gemelos digitales para gestionar el ciclo de vida de sus activos y productos, permitiéndoles tomar decisiones informadas de manera anticipada.

Este aumento en la adopción de gemelos digitales se ha facilitado por la reducción de los costos de tecnologías que mejoran tanto el Internet de las Cosas (IoT) como los gemelos digitales. En los últimos años, los gemelos digitales han demostrado su valor en aplicaciones comerciales vitales, y se espera que su alcance se expanda a más casos de uso y diferentes industrias, lo que ha impulsado un crecimiento exponencial en su aplicación. Las proyecciones del mercado global indican un crecimiento impresionante, con una tasa prevista del 58% y una estimación de alcanzar los 48.2 mil millones de dólares para el año 2026. En resumen, los gemelos digitales se han convertido en una herramienta esencial en la era digital, con un potencial de crecimiento sustancial a medida que continúan evolucionando y expandiéndose en diversas industrias y aplicaciones.

## 2. Marco Teórico

### 2.1 Componentes del Gemelo Digital

En el contexto del gemelo digital, centrado en un sensor de temperatura, humedad, luminosidad y presión, se identifican los siguientes componentes:

1. **Sensor Físico:** Este componente representa el dispositivo real que mide la temperatura, humedad, presión y luminosidad en entornos físicos. Para nuestro gemelo digital usaremos un sensor de la marca liberium.
2. **Gestión de datos:** Para la administración de los datos que nutrirán nuestro gemelo digital, emplearemos una solución basada en contenedores Docker. Esta configuración nos permitirá integrar InfluxDB y Node-RED, creando un puente eficaz hacia MQTT. Con esta infraestructura, facilitaremos la conexión y el almacenamiento de los datos recogidos, garantizando un flujo constante de información hacia nuestro gemelo digital.
3. **Gemelo Digital:** El propio gemelo digital, que actúa como una representación virtual del sensor y su entorno. Incluye modelos, algoritmos y datos que replican las condiciones del mundo real para permitir la simulación y análisis. Usaremos python como lenguaje de programación.
4. **Interfaz de Usuario Interactiva:** Diseñada para permitir a los usuarios interactuar con el gemelo digital. Permite la observación, análisis y configuración de parámetros del gemelo digital.

### 2.2 Herramientas y Tecnologías Relevantes

En la implementación del gemelo digital del sensor de presión, luminosidad, temperatura y humedad, se emplearon herramientas y tecnologías específicas que contribuyeron al éxito del proyecto. Entre ellas se destacan:

1. **Wasmote para programación del sensor:** La programación del sensor se llevó a cabo mediante la plataforma Wasmote, proporcionando un entorno robusto y especializado para la configuración y operación del dispositivo.



2. **Protocolo MQTT para transmisión de datos:** Se implementó el protocolo MQTT para la transmisión eficiente de datos generados por el sensor. Esta elección, respaldada por su eficacia en entornos IoT, facilitó la conectividad y la interoperabilidad del gemelo digital.
3. **QoS para gestionar el tráfico de datos:** se utiliza para gestionar el tráfico de datos en redes informáticas. Permite priorizar ciertos tipos de datos sobre otros, asegurando ancho de banda suficiente, baja latencia y fiabilidad
4. **Docker para la conexión con la base de datos:** Usamos Docker esencialmente para crear y desplegar consistentemente aplicaciones como InfluxDB y Node-RED. Facilita la conexión y gestión de estos servicios, garantizando entornos reproducibles y eficientes.
5. **Se utiliza Node-RED para integrar dispositivos IoT:** Con Node-RED conectamos servicios y creamos una lógica de control sin necesidad de programación extensa. Consta de una plataforma visual que simplifica la creación de flujos de automatización.
6. **InfluxDB como base de datos:** Es una base de datos de series temporales de alto rendimiento. Se emplea para almacenar y consultar datos temporales, como lecturas de sensores IoT, facilitando análisis y visualización eficientes.
7. **Python como Lenguaje de Programación:** El desarrollo del gemelo digital se realizó utilizando Python, aprovechando su versatilidad y eficiencia. Se emplearon diversas bibliotecas y frameworks específicos de Python para optimizar el manejo de datos y la integración de funcionalidades.

#### **Librerías de Python Utilizadas:**

- **Pandas y NumPy** para el eficiente manejo y análisis de datos, respectivamente.
- **TensorFlow y Keras** ha sido esencial para la implementación de modelos de aprendizaje automático.
- **Torch:** Biblioteca de aprendizaje automático. Se ha utilizado para construir y entrenar modelos de aprendizaje profundo con PyTorch.

- **Scikit-learn (sklearn)** ha enriquecido la capacidad predictiva mediante la implementación de diversos algoritmos de machine learning.
- **Matplotlib** ha facilitado la visualización clara y comprensible de datos, contribuyendo a la interpretación de resultados.

**PyQt5** es una biblioteca de Python diseñada para el desarrollo de aplicaciones de escritorio con interfaces gráficas de usuario (GUI). Con una amplia gama de herramientas y funcionalidades, PyQt5 facilita la creación de interfaces modernas e interactivas de manera eficiente.

PyQt5 se destaca por su atractivo visual moderno, ofreciendo estilos y temas avanzados que permiten crear interfaces más atractivas y profesionales que Tkinter, la biblioteca de GUI estándar de Python. Además, cuenta con una documentación detallada y una comunidad activa de desarrolladores, lo que brinda un amplio soporte y recursos útiles para los desarrolladores, facilitando el aprendizaje y la resolución de problemas durante el desarrollo de aplicaciones. Con una amplia variedad de widgets y herramientas para la creación de interfaces gráficas, PyQt5 proporciona una mayor flexibilidad y personalización en el diseño de aplicaciones, permitiendo a los desarrolladores crear interfaces más complejas y funcionales. Además, su compatibilidad con múltiples plataformas, incluyendo Windows, macOS y Linux, facilita el desarrollo de aplicaciones multiplataforma sin necesidad de modificar el código fuente. Finalmente, PyQt5 es conocido por su potencia y rendimiento, siendo ideal para proyectos con requisitos exigentes en términos de procesamiento de datos o renderizado de gráficos, lo que lo convierte en una opción sólida para proyectos que necesitan un alto rendimiento.

**Entornos de Desarrollo de Python:** Jupyter Notebook ha sido el entorno de desarrollo principal, brindando flexibilidad y capacidad de análisis interactivo durante la creación del gemelo digital.

La combinación de Waspmote, el protocolo MQTT, Python y sus librerías especializadas ha permitido no solo la programación eficiente del sensor, sino también la creación de un gemelo digital robusto y preciso, listo para aplicaciones diversas en el ámbito de la presión, temperatura y humedad.

## **2.3 Definición de modelos explorados**

### **2.3.1 ARMA ARIMA**

ARMA (Autoregressive Moving Average) y ARIMA (Autoregressive Integrated Moving Average) son modelos estadísticos utilizados para el análisis y predicción de series temporales. ARMA combina dos componentes: autoregresión (AR), que predice valores futuros basándose en valores pasados, y promedio móvil (MA), que incorpora el error de pronóstico en términos anteriores. ARIMA añade un elemento de diferenciación (I) para hacer la serie temporal estacionaria, es decir, para estabilizar la media de la serie a lo largo del tiempo, mejorando la predicción en series no estacionarias.

### **2.3.2 WALK FORWARD**

El método "Walk Forward" es una técnica de validación utilizada en la modelización de series temporales, especialmente en el trading y la predicción financiera. Se basa en la idea de 'caminar hacia adelante' en el tiempo, entrenando el modelo en un segmento de datos y luego probándolo en el siguiente segmento para evaluar su rendimiento. Esta técnica ayuda a comprender cómo el modelo se desempeñará en condiciones cambiantes a lo largo del tiempo, proporcionando una evaluación más realista de su capacidad predictiva en escenarios futuros.

### **2.3.3 RNN**

Las Redes Neuronales Recurrentes (RNN) son una clase de redes neuronales diseñadas para manejar secuencias de datos, como series temporales o secuencias lingüísticas. Se caracterizan por tener conexiones recurrentes, lo que les permite mantener un estado o memoria sobre la información previa en la secuencia. Esto les hace idóneas para tareas como el reconocimiento de voz, la generación de texto y otras aplicaciones donde el contexto y el orden de los datos son importantes. Las RNN pueden capturar patrones temporales y secuenciales que otras redes neuronales no pueden.

## **8.3.4 TRANSFORMERS**

Los Transformers, inicialmente desarrollados para tareas de procesamiento de lenguaje natural, se están adaptando con éxito para el análisis de series temporales. Su característica distintiva es el mecanismo de atención, que permite al modelo enfocarse en

diferentes segmentos de los datos de entrada, captando así dependencias a largo plazo y patrones complejos. En el contexto de series temporales, esta capacidad les permite analizar y predecir secuencias temporales con gran eficacia, identificando tendencias y correlaciones que otros modelos podrían pasar por alto. Su estructura paralela facilita un procesamiento más rápido y eficiente de datos secuenciales, lo que los convierte en una herramienta potente para predecir eventos futuros y analizar dinámicas temporales en una variedad de campos.

## 2.4 Definición de métricas de validación empleadas

### 2.4.1 MAE Error Absoluto Medio

El MAE, o Error Absoluto Medio por sus siglas en inglés (Mean Absolute Error), es una métrica comúnmente utilizada en problemas de regresión para evaluar la precisión de un modelo. Este error se calcula tomando la media de las diferencias absolutas entre los valores reales y las predicciones del modelo para cada punto de datos en el conjunto de prueba.

La fórmula del MAE se expresa de la siguiente manera:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i|$$

Donde:

- $n$  es el número total de puntos de datos en el conjunto de prueba.
- $y_i$  representa el valor real del objetivo para el punto de datos  $i$ .
- $y'_i$  es la predicción del modelo para el punto de datos  $i$ .

El MAE es una medida de la magnitud promedio de los errores en las predicciones del modelo, sin considerar su dirección. Cuanto menor sea el valor del MAE, mejor será el rendimiento del modelo, ya que indicará que las predicciones están más cerca de los valores reales en promedio. Esta métrica es útil porque es fácil de interpretar y no se ve afectada por valores atípicos o extremos en los datos. Sin embargo, dado que trata todos los errores por igual, puede no ser adecuada para todos los escenarios, especialmente si algunos errores son más críticos que otros. En tales casos, otras métricas como el Error

Cuadrático Medio (MSE) o el Error Cuadrático Medio de la Raíz (RMSE) pueden ser más apropiadas.

#### 8.4.2 MSE Error Cuadrático Medio

El MSE es una medida fundamental en la evaluación de modelos de regresión. Proporciona información sobre la magnitud promedio de los errores entre las predicciones del modelo y los valores reales.

La fórmula para calcular el MSE es la siguiente:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Donde:

- $n$  es el número total de puntos de datos en el conjunto de prueba.
- $y_i$  representa el valor real del objetivo para el punto de datos  $i$ .
- $\hat{y}_i$  es la predicción del modelo para el punto de datos  $i$ .

El MSE se calcula tomando la media de los cuadrados de las diferencias entre los valores reales y las predicciones del modelo para cada punto de datos en el conjunto de prueba. Cuanto menor sea el valor del MSE, mejor será el rendimiento del modelo, ya que indicará que las predicciones están más cerca de los valores reales en promedio.

En términos prácticos, un MSE cercano a 0 sugiere que el modelo se ajusta bien a los datos, es decir, que las predicciones del modelo son muy similares a los valores reales. Por otro lado, un MSE más alto indica que las predicciones del modelo están más alejadas de los valores reales, lo que puede indicar una menor calidad del modelo.

MSE bajo, pero aun así tener predicciones inexactas en regiones críticas del espacio de características.

#### 8.4.2 RMSE Error Cuadrático Medio de la Raíz

El Error Cuadrático Medio de la Raíz (RMSE, por sus siglas en inglés, Root Mean Squared Error) es otra métrica comúnmente utilizada en problemas de regresión para evaluar la precisión de un modelo. A diferencia del MAE, el RMSE penaliza de manera

más significativa los errores grandes, ya que toma la raíz cuadrada de la media de los errores cuadráticos entre los valores reales y las predicciones del modelo.

La fórmula del RMSE se expresa de la siguiente manera:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Donde:

- $n$  es el número total de puntos de datos en el conjunto de prueba.
- $y_i$  representa el valor real del objetivo para el punto de datos  $i$ .
- $\hat{y}_i$  es la predicción del modelo para el punto de datos  $i$ .

El RMSE se calcula tomando la raíz cuadrada de la media de los errores cuadráticos entre los valores reales y las predicciones del modelo. Esto significa que los errores más grandes tendrán un impacto más significativo en la métrica en comparación con el MAE. Al igual que con el MAE, cuanto menor sea el valor del RMSE, mejor será el rendimiento del modelo, ya que indicará que las predicciones están más cerca de los valores reales en promedio.

El RMSE es una métrica útil para evaluar la precisión de un modelo de regresión, ya que proporciona una idea de la magnitud promedio de los errores en las predicciones, teniendo en cuenta tanto la dirección como la magnitud de los errores. Sin embargo, al cuadrar los errores, el RMSE puede verse afectado por valores atípicos o extremos en los datos, lo que puede influir en su interpretación en ciertos casos.

#### **8.4.3 $R^2$ Coeficiente de determinación**

El coeficiente de determinación, comúnmente conocido como  $R^2$  o R-cuadrado, es una métrica fundamental en la evaluación de modelos de regresión. Proporciona información

sobre la proporción de la varianza en la variable dependiente que es explicada por el modelo.

La fórmula para  $R^2$  se define de la siguiente manera:

$$R^2 = 1 - \frac{SS_{tot}}{SS_{res}}$$

Donde:

- $SS_{res}$  es la suma de los cuadrados de los residuos o errores, es decir, la suma de los cuadrados de las diferencias entre los valores reales  $y_i$  y las predicciones del modelo  $\hat{y}_i$
- $SS_{tot}$  es la suma total de los cuadrados, que representa la variabilidad total de los datos con respecto a su media.

Un  $R^2$  de 1 indica que el modelo explica perfectamente la variabilidad de los datos, mientras que un  $R^2$  de 0 indica que el modelo no explica nada de la variabilidad.

En términos prácticos, un  $R^2$  cercano a 1 sugiere que el modelo se ajusta bien a los datos, es decir, que las predicciones del modelo se acercan bastante a los valores reales. Por otro lado, un  $R^2$  cercano a 0 indica que el modelo no explica adecuadamente la variabilidad de los datos y puede necesitar mejoras o ser inadecuado para el conjunto de datos específico.

Es importante tener en cuenta que el  $R^2$  por sí solo no indica si un modelo es bueno o malo. Es esencial considerar otras métricas y evaluar el contexto específico del problema para una evaluación completa del modelo. Por ejemplo, un modelo puede tener un  $R^2$  alto, pero aun así tener predicciones inexactas en regiones críticas del espacio de características.

## 3. Metodología

### 3.1 Etapas del desarrollo

#### Etapa 1: Diseño

- **Objetivo:** Preparar la base para el desarrollo del gemelo digital, configurando la adquisición de datos, el procesamiento de datos y la arquitectura del software.

- **Desarrollo:**

1. **Adquisición y Procesamiento de Datos**

El procesamiento de datos es una parte crítica de esta fase. Los datos brutos recopilados pueden ser voluminosos y ruidosos, por lo que se requiere una serie de operaciones para limpiar, filtrar y preparar los datos para su uso. Esto incluye la eliminación de datos atípicos o irrelevantes, la interpolación de datos faltantes y la normalización de unidades de medida, si es necesario.

2. **Arquitectura del Software**

Estructura y arquitectura de software desarrollada con clases y componentes necesarios para organizar y gestionar el modelo y los datos de manera efectiva.

Se realizarán dos esquemas de arquitectura de software utilizando lenguaje UML:

- **Esquema de Alto Nivel (Diagrama):** Un diagrama que proporciona una vista general de los componentes y su interacción a nivel abstracto.
- **Esquema de Bajo Nivel (Diagrama de Clases):** Un diagrama detallado que describe las clases, relaciones y atributos a nivel de implementación.

#### Etapa 2: Implementación

- **Objetivo:** Desarrollar e implementar las clases y componentes necesarios, incluyendo el motor de simulación, la interfaz de usuario y el procesamiento de datos, para completar la implementación del Sistema de Gemelo Digital.
- **Desarrollo:**



1. **Implementación de las clases:** Se completará el desarrollo del Sistema de Gemelo Digital, logrando un conjunto de componentes plenamente funcionales. Se creará el GD de los sensores. Se implementará un motor de simulación en la clase Predicción, permitiendo la creación, entrenamiento y predicción de valores con exactitud usando un modelo de IA. La clase Graficas generara gráficos efectivos que mostraran tendencias y comparaciones de las predicciones con los valores reales. Además, la clase DatosProcesados garantiza que los datos sean procesados de manera correcta y estén listos para su utilización en las predicciones y gráficas
2. **Interfaz de usuario:** En esta etapa también se creará una interfaz para que el usuario pueda interactuar con el gemelo.

### **Etapas 3: Validación del Sistema de Gemelo Digital**

- **Objetivo:** Verificar y validar la funcionalidad, precisión y rendimiento del Sistema de Gemelo Digital para asegurar que cumple con los requisitos y expectativas.
- **Desarrollo:** En esta etapa, se llevará a cabo una serie de pruebas exhaustivas y análisis para evaluar el Sistema de Gemelo Digital en condiciones diversas. Se realizarán pruebas de predicciones con datos reales, análisis de tendencias a largo plazo y evaluación del rendimiento en tiempo real. Se identificarán y corregirán posibles problemas y errores, y se ajustarán los modelos de predicción según sea necesario.

## **3.2 Arquitectura del software**

La arquitectura del software presentada en este trabajo, diseñada para la recolección, procesamiento y análisis de datos ambientales, se beneficia significativamente del uso del lenguaje de modelado UML (Unified Modeling Language) para su diagrama de clases. UML permite una representación visual clara y estandarizada de la estructura del software, facilitando la comprensión de las relaciones y dependencias entre las distintas clases. Cada clase, especializada en tareas específicas que van desde la recolección de datos mediante sensores hasta la visualización de información a través de una interfaz de usuario, es detalladamente descrita. La clase central, `Digital_Twin_System`, juega un

papel crucial al coordinar las operaciones entre las distintas clases, asegurando un flujo de datos eficiente y la ejecución correcta de tareas. Este enfoque modular y escalable, representado a través de UML, es esencial para enfrentar los desafíos complejos asociados con el monitoreo y análisis de datos ambientales, proveyendo una solución robusta y flexible adaptada a diversas aplicaciones. La utilización de UML en este contexto no solo mejora la documentación del proyecto, sino que también facilita la fase de diseño y el desarrollo futuro del software.

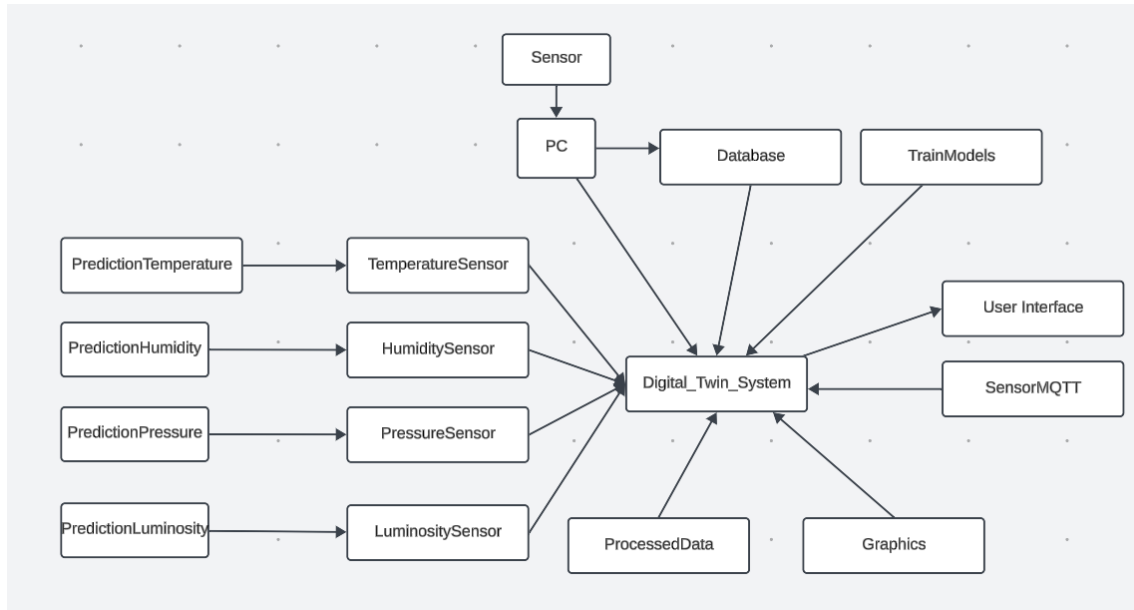


FIGURA 9: Diagrama de clases UML simplificado

### 3.3 Exploración de modelos

En esta sección del trabajo, exploramos y evaluamos varios modelos de aprendizaje automático y aprendizaje profundo (ML y DL, respectivamente) para comprender mejor cómo pueden contribuir al análisis y procesamiento de nuestros datos ambientales recopilados. Esta evaluación nos permite identificar el modelo más adecuado que mejora la precisión y eficiencia de nuestras predicciones y análisis. Comparamos diferentes algoritmos y configuraciones, analizando sus ventajas y limitaciones en el contexto específico de nuestro sistema. Este análisis es crucial para optimizar la capacidad predictiva y operativa de nuestra arquitectura de software.

### **3.3.1 ARMA ARIMA:**

Se exploraron inicialmente métodos estadísticos ARMA y ARIMA como posibles enfoques para la predicción de las mediciones del sensor multifunción. Se llevó a cabo una prueba de estacionariedad utilizando el test ADF (Augmented Dickey-Fuller) en las observaciones de temperatura, confirmando la estacionariedad de la serie temporal.

Se utilizó el criterio de información de Akaike (AIC) para seleccionar el orden óptimo del modelo autorregresivo (AR) en ARIMA. La selección del orden AR se realizó automáticamente con la función `ar_select_order` de statsmodels, utilizando el método de AIC. Sin embargo, a pesar de estos esfuerzos iniciales, los resultados obtenidos no fueron satisfactorios, lo que condujo a la consideración de otras opciones.

Posteriormente, se realizaron pruebas con diferentes modelos ARIMA, variando el orden AR, con el objetivo de identificar el modelo que proporcionara un mejor ajuste a los datos y una mayor capacidad predictiva.

Dado el carácter secuencial y frecuente de las mediciones, la necesidad de obtener resultados precisos requería la consideración de valores elevados para los parámetros  $p$  y  $q$ . Estos parámetros corresponden al orden de la parte autorregresiva (AR) y la parte de media móvil (MA) del modelo ARIMA, respectivamente.

El aumento de los valores de  $p$  y  $q$ , aunque podría mejorar la precisión de la predicción, también conllevaba un aumento significativo en el tiempo de ejecución de los modelos. Este incremento en el tiempo de procesamiento no solo afectaba la eficiencia computacional, sino que también planteaba consideraciones prácticas en términos de rentabilidad.

### **3.3.2 WALK FORWARD:**

En la aplicación del método de "walk forward", se llevó a cabo una fase inicial de limpieza y preparación de los datos. Este proceso incluyó la selección de la variable objetivo, en este caso, las mediciones. Además, se realizó una configuración de la variable de predicción y se establecieron los lags (rezagos) relevantes para el modelo. Se implementó un enfoque de "walk forward" utilizando la clase `ForecasterAutoreg`, la cual utilizó un regresor basado en Random Forest. Este regresor se entrenó con datos de entrenamiento,

y se especificó un número de lags, representando la cantidad de periodos temporales anteriores considerados para predecir el siguiente valor.

La medida de error absoluto medio (MAE) obtenida fue de aproximadamente 7.07. En este caso, la cifra indicaba que, en promedio, las predicciones del modelo diferían alrededor de 7.07 grados de las mediciones reales. Una puntuación más baja en el MAE generalmente sugería un mejor rendimiento del modelo por lo que se aplicaron unos cambios.

Para mejorar el MAE del modelo anterior se le aplicó un nuevo enfoque. En el primer bloque de código, se inicializó un `ForecasterAutoreg` con un regresor `RandomForestRegressor` y se realizó un bucle `walk-forward` sobre el conjunto de datos de entrenamiento. En cada iteración, se creó una nueva instancia de `ForecasterAutoreg`, y el conjunto de entrenamiento se actualizó para incluir la última observación junto con el error previo. Este enfoque permitió una adaptación continua del modelo a las condiciones cambiantes de los datos.

Dentro del bucle, se ajustó el modelo al conjunto de entrenamiento actualizado y se realizó una predicción para un solo paso de tiempo. La predicción se almacenó en una lista para su posterior visualización. Además, se calculó y actualizó el error actual, que se utilizó como ajuste en el siguiente paso para mejorar la capacidad del modelo para anticipar las variaciones en las mediciones.

En esta evolución se logró una mejora sustancial en la precisión de las predicciones. Se observó una reducción significativa en el Mean Absolute Error (MAE), disminuyendo de 7.07 a 2.54. Este progreso demostró la eficacia del nuevo enfoque para adaptarse dinámicamente a los cambios en los datos y generar predicciones más cercanas a las mediciones reales.

A pesar de esta mejora, se reconoció que el MAE de 2.54 aún no alcanzaba el nivel de precisión deseado. Ante este escenario, se exploraron y evaluaron otros modelos con el objetivo de mejorar aún más la capacidad predictiva.

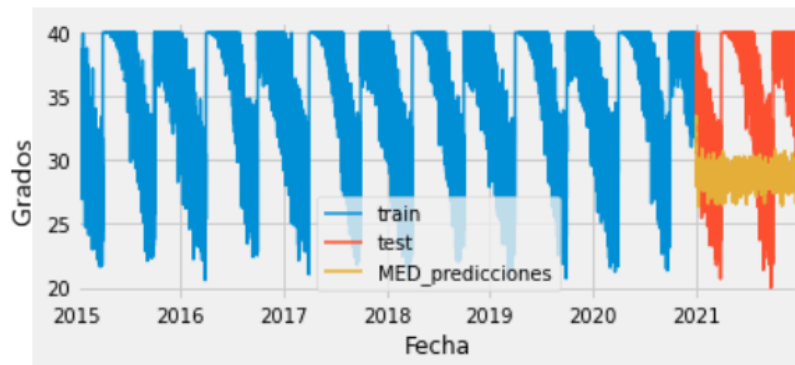


FIGURA 10: Grafica de las predicciones de temperatura usando walk forward

### 3.3.3 RNN:

Se comenzó a implementar una función que permitiera crear secuencias temporales a partir de la serie de mediciones y sus fechas correspondientes. Este proceso implicaba la normalización de las mediciones para asegurarse de que estuvieran dentro de un rango específico y la generación de secuencias temporales con salidas asociadas.

En la siguiente etapa, se dividió el conjunto de datos en conjuntos de entrenamiento y prueba, ajustando la forma de los datos para que fueran compatibles con la entrada del modelo RNN. Aunque proporcionó resultados iniciales, se observaron limitaciones en términos de rendimiento predictivo durante la evaluación del modelo.

Ante estas limitaciones, se exploraron otros modelos más efectivos. Se implementó una RNN con una capa LSTM bidireccional, una arquitectura más avanzada que permitía capturar patrones temporales en ambas direcciones. Esta mejora significativa se tradujo en una mayor capacidad del modelo para comprender la complejidad de la serie temporal de mediciones.

En la construcción y entrenamiento de este nuevo modelo, se incorporaron técnicas adicionales, como el uso de Dropout para regularizar y mejorar la generalización del modelo. La entrada de este modelo mejorado abarcaba secuencias de mediciones y fechas, y el proceso de compilación y entrenamiento se llevó a cabo con un optimizador Adam y una función de pérdida de error absoluto medio.

A pesar de obtener un Mean Squared Error (MAE) de 0.76 y un coeficiente de determinación  $R^2$  de 0.87 en los datos de prueba, se observó que el modelo actual, aunque preciso, tenía limitaciones en cuanto a la interpretación de la estacionalidad. En lugar de capturar las variaciones estacionales de las mediciones, parecía enfocarse en predecir

mediciones con una alta variabilidad, lo que podría indicar una falta de sensibilidad hacia los patrones cíclicos o estacionales presentes en la serie temporal.

Era crucial tener en cuenta estas observaciones al evaluar el rendimiento del modelo, ya que un buen desempeño en métricas generales como el MAE y  $R^2$  no garantizaba la capacidad del modelo para comprender aspectos específicos de la serie temporal, como la estacionalidad.

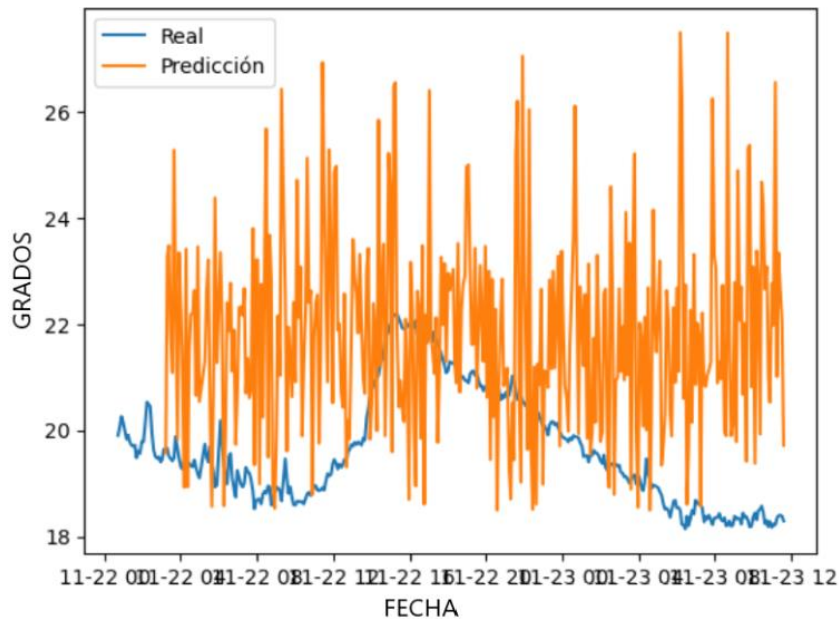


FIGURA 11: Grafica de las predicciones de temperatura usando RNN

### 3.3.4 TRANSFORMERS:

En esta etapa, se procedió a implementar y evaluar la eficacia de modelos basados en la arquitectura Transformer en la tarea de predicción de series temporales. Antes de introducir los datos al modelo, se llevó a cabo una preparación meticulosa. Esto incluyó la estandarización de los datos para garantizar que estuvieran en una escala uniforme y la creación de secuencias temporales, estructurando así la información de manera óptima para la entrada en el modelo.

El modelo es una implementación innovadora de la arquitectura Transformer diseñada específicamente para abordar la predicción en series temporales. Con una estructura modular y escalable, este modelo se inició con una capa de embedding que transformó las entradas a un espacio de representación oculta. A continuación, se incorporaron múltiples capas de atención Multihead, cada una dotada con un número de cabezas y una función de dropout para regularizar el proceso. De manera paralela, capas lineales, cada

una seguida de una función de activación ReLU y otra de dropout, fueron implementadas para procesar la información. La capa de salida llevó a cabo la proyección de las características aprendidas a las dimensiones de salida, proporcionando así las predicciones finales. Cabe destacar que, durante el flujo hacia adelante (forward), se realizó una permutación del tensor de entrada para adecuarlo a la estructura requerida por las capas de atención, y al final, se realizó un promedio a lo largo de la dimensión temporal para obtener la salida final. Este diseño permitió capturar patrones complejos en series temporales, proporcionando así una herramienta potente para la predicción precisa de mediciones futuras.

Con la implementación del modelo, se logró obtener un Mean Absolute Error (MAE) cercano a 0.12 y un coeficiente de determinación ( $R^2$ ) de aproximadamente 0.988. Estos resultados sugirieron un rendimiento significativamente positivo en la capacidad del modelo para predecir mediciones futuras en comparación con las predicciones anteriores. Al examinar visualmente las predicciones en el conjunto de prueba junto a las mediciones reales, se observó que el modelo era capaz de identificar de manera precisa y adecuada los patrones inherentes a los datos. Este desempeño destacado respaldó la eficacia del modelo en la tarea de predicción de series temporales, brindando un enfoque avanzado y preciso para comprender y anticipar las variaciones en las mediciones.

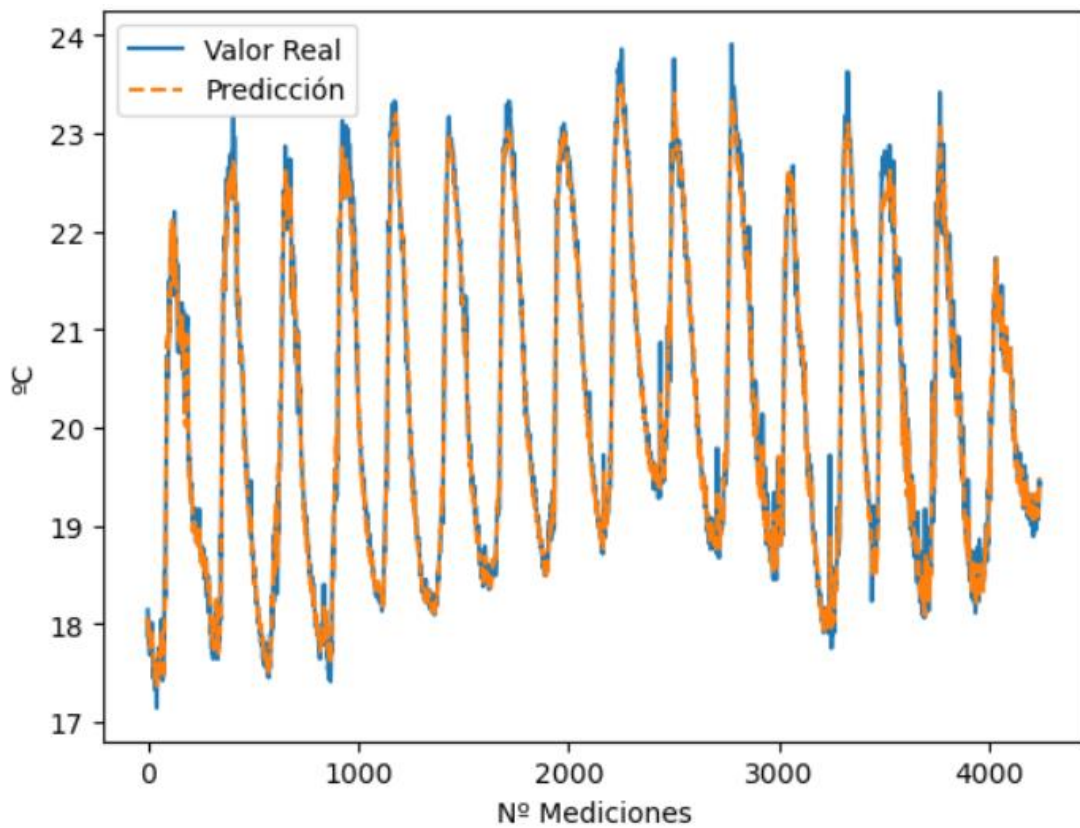


FIGURA 12: Grafica de las predicciones de temperatura usando transformers

Con la evaluación de diversos enfoques, desde métodos estadísticos hasta modelos avanzados como RNN y Transformer, se ha observado una evolución en la precisión de las predicciones para las mediciones del sensor multifunción. Considerando el rendimiento destacado del modelo Transformer, con un MAE cercano a 0.12 y un R2 de aproximadamente 0.988, se ha decidido implementarlo como la opción preferida para la tarea de predicción. El Transformer demostró ser eficaz al identificar patrones complejos en las series temporales, ofreciendo así una solución avanzada y precisa para anticipar y comprender las variaciones en las mediciones.



## **4. Recopilación de datos**

### **4.1 Configuración del sensor**

Para recolectar los datos se hace uso del sensor real liberium waspmote. Se carga al sensor un script para realizar la lectura de datos provenientes de un sensor de temperatura, humedad, presión y luminosidad y enviar estos datos a un servidor o broker MQTT a través de una conexión WiFi. En este proceso, se utilizan principalmente dos protocolos: WiFi y MQTT, cada uno desempeñando roles específicos en la transmisión de datos desde el dispositivo físico hacia un sistema de almacenamiento o monitoreo remoto.

#### **4.1.1 Protocolo WiFi**

El protocolo WiFi se utiliza para establecer la conexión inalámbrica entre el dispositivo (que alberga el sensor) y la red de internet local. Esto permite que el dispositivo tenga acceso a la red y, por ende, pueda comunicarse con servicios externos como un servidor MQTT. La configuración WiFi requiere de ciertos parámetros básicos como el SSID (nombre de la red WiFi) y una contraseña, dependiendo de la seguridad de la red. Este protocolo es fundamental para cualquier aplicación que requiera enviar o recibir datos a través de internet sin necesidad de una conexión física directa.

#### **4.1.2 Protocolo MQTT**

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero diseñado para la comunicación en dispositivos de Internet de las Cosas (IoT). Se utiliza para enviar (publicar) datos de sensores a un servidor o broker MQTT, que actúa como intermediario, recibiendo los mensajes y distribuyéndolos a los dispositivos o servicios suscritos (clientes) interesados en esos datos específicos. Este protocolo es especialmente útil en entornos IoT debido a su eficiencia, bajo ancho de banda y requisitos mínimos de energía, lo que lo hace ideal para dispositivos con recursos limitados. Además, MQTT facilita la implementación de comunicaciones en tiempo real entre dispositivos y aplicaciones, permitiendo una amplia gama de aplicaciones como monitoreo remoto, automatización del hogar, y más.

### 4.1.3 QoS

El QoS es un parámetro clave en MQTT que determina la garantía de entrega de los mensajes entre el emisor (publicador) y el receptor (suscriptor) a través del broker MQTT.

- QoS 0 (A lo sumo una vez): Este nivel garantiza una entrega máxima de una vez. No hay confirmación de que el mensaje haya sido recibido, y no se hacen intentos de reenvío. Es el modo más rápido, pero también el menos fiable.
- QoS 1 (Al menos una vez): Garantiza que el mensaje llegue al menos una vez al receptor. Puede haber duplicados si el mensaje es reenviado, pero se asegura que al menos uno llegue.
- QoS 2 (Exactamente una vez): Este es el nivel más alto de garantía, donde se asegura que el mensaje se reciba exactamente una vez por el destinatario. Esto se logra mediante un proceso de handshake de cuatro pasos entre el emisor y el receptor, lo que lo hace más fiable pero también más lento y con mayor consumo de recursos.

En nuestro proyecto, hemos optado deliberadamente por utilizar el nivel 2 de Calidad de Servicio (QoS 2) para la transmisión de datos desde el sensor físico al servidor MQTT. Esta decisión se fundamenta en la premisa de que la integridad y exactitud de los datos son primordiales, superando en importancia a la velocidad de transmisión. El uso de QoS 2 asegura que cada mensaje se entregue exactamente una vez al receptor, eliminando cualquier posibilidad de mensajes duplicados o perdidos, lo cual es esencial para mantener la fiabilidad y precisión de los datos en aplicaciones críticas. Este nivel de QoS, aunque puede introducir una mayor latencia debido a su mecanismo de confirmación de cuatro pasos, ofrece beneficios significativos al garantizar la entrega de datos sin errores. Tal garantía es vital para la limpieza de datos y el análisis posterior, ya que simplifica estos procesos al eliminar la necesidad de gestionar duplicados o verificar la integridad de los datos recibidos.

Adicionalmente, estos datos se envían a través de un publicador de MQTT, un mecanismo que facilita la entrega eficiente y segura de información desde el dispositivo hacia el broker MQTT. Este método de publicación es crítico para el flujo de datos en tiempo real, permitiendo que nuestro sistema distribuya la información recopilada a cualquier

suscriptor interesado en ella, manteniendo un alto nivel de fiabilidad y precisión en la comunicación.

## 4.2 Conexión sensor base de datos

En el sistema implementado, se utiliza Docker Compose para desplegar dos servicios esenciales: InfluxDB y Node-RED. Estos servicios están diseñados para trabajar en conjunto y gestionar los datos provenientes de un sensor en tiempo real.

### Docker Compose:

**InfluxDB Service:** El servicio influxdb se basa en la imagen de InfluxDB y expone el puerto **8086**. Este servicio actúa como la base de datos central para almacenar y recuperar datos del sensor. El puerto **8086** sirve como interfaz de la API HTTP de InfluxDB, permitiendo la comunicación con otras aplicaciones.

**Node-RED Service:** El servicio nodered utiliza la imagen oficial de Node-RED y expone el puerto **1880**. Node-RED es una plataforma de desarrollo visual que facilita la integración de dispositivos y servicios en la web mediante flujos lógicos. En este caso, se emplea para procesar y gestionar los datos provenientes del sensor.

### Configuraciones:

**Node-Red:** La configuración de Node-RED delineada establece un flujo para recibir y almacenar datos de un sensor mediante un broker MQTT y una base de datos InfluxDB. Se emplean tres nodos MQTT In para subscribirse a los temas "temp", "hum" y "pres". Actúan como puntos de entrada para datos del sensor, conectándose al broker MQTT con la dirección IP "192.168.1.32" y el puerto "1883". Se incorporan tres nodos InfluxDB Out para escribir datos en la base "database" de InfluxDB. Cada nodo se asocia a un tipo de dato específico ("Temperatura", "Humedad", "Presión") y realiza la escritura en la base de datos dentro de la organización "GD" y el cubo "GDbucket". Estos nodos están conectados a los nodos MQTT In respectivos para recibir datos del sensor.

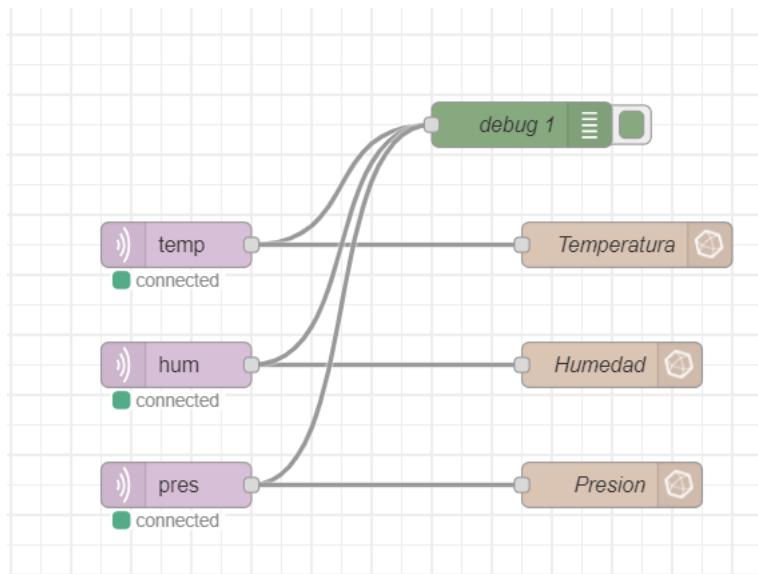


FIGURA 13: Visualización de la configuración de Node-RED

**InfluxDB:** En la configuración de InfluxDB, se ha creado una base de datos destinada a recibir la información proveniente de Node-RED. Esta base de datos, denominada "database," se encuentra configurada para almacenar los datos recopilados por los nodos correspondientes en el flujo de Node-RED. Asimismo, esta base está asociada a la organización "GD" y al cubo "GDbucket," estableciendo un entorno organizado y específico para la gestión de los datos capturados por el sistema.

Q Filter tables...	_start	_stop	_time	_value
_field = value _measurement = hum	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 14:00:00 GM...	35,10
_field = value _measurement = pres	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 15:00:00 GM...	34,24
_field = value _measurement = temp	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 16:00:00 GM...	34,25
	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 17:00:00 GM...	36,88
	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 18:00:00 GM...	37,14
	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 19:00:00 GM...	38,76
	2024-01-30 16:39:41 GMT...	2024-02-29 16:39:41 GM...	2024-02-06 20:00:00 G...	41,67

FIGURA 14: Visualización de la aplicación de InfluxDB

En el contexto del gemelo digital, se realiza un acceso a la base de datos de InfluxDB desde Python. Utilizando consultas, se descarga la información en formato CSV, que posteriormente se integra en la base de datos del gemelo digital. Este proceso permite la transferencia de datos entre el entorno de InfluxDB y el sistema del gemelo digital, asegurando la actualización y coherencia de la información almacenada en ambas plataformas.

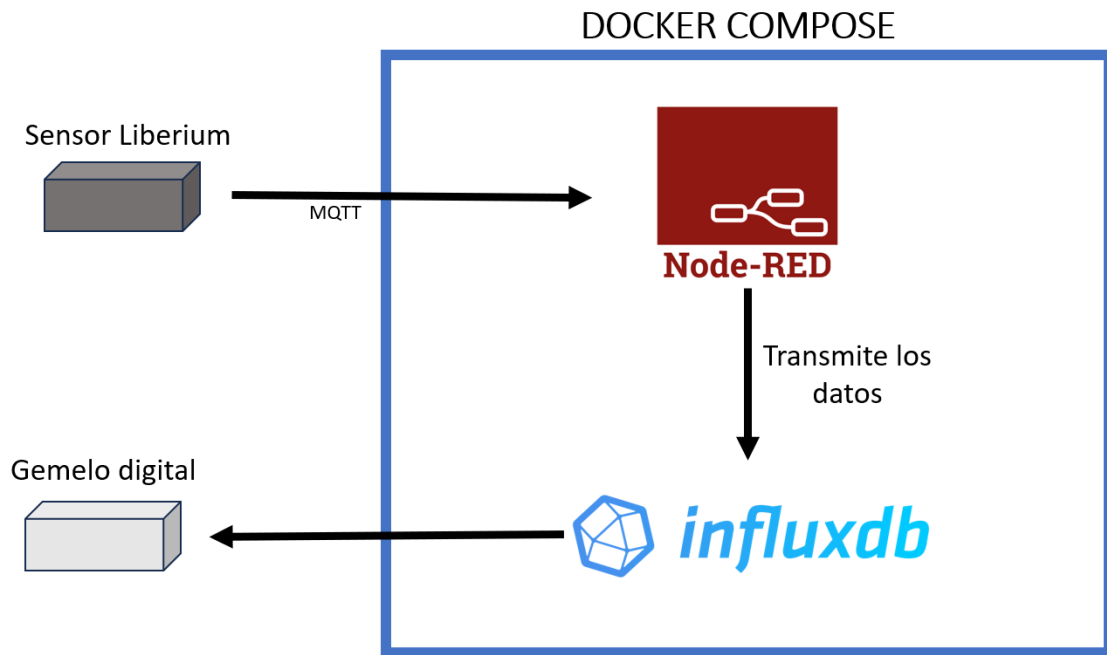


FIGURA 15: Representación del sistema de recolección de datos

## 5. Implementación

En la fase de implementación, se ha optado por una estructura organizada y modulada mediante clases de Python. En este diseño, las clases se interconectan para formar un sistema integral, donde cada clase desempeña un papel específico y contribuye al funcionamiento general del modelo. Una clase principal actúa como el controlador principal, coordinando la interacción entre las demás clases. Este enfoque proporciona una implementación clara y eficiente, facilitando la comprensión y la gestión del código en el desarrollo del modelo del GD. A lo largo de esta explicación detallada, se explorarán las funcionalidades y relaciones de cada clase para comprender a fondo la estructura y el dispositivo creado.

### 5.1 Bases de datos

En la fase inicial de la implementación, se han creado clases específicas para gestionar las bases de datos. En particular, se han desarrollado dos clases esenciales: una dedicada al manejo de los datos recopilados por el sensor y otra diseñada para almacenar modelos previamente entrenados.

La clase **Database** representa una base de datos que puede almacenar diferentes tipos de datos temporales, como temperatura, luminosidad, presión y humedad. Cada tipo de dato se asocia con un archivo CSV obtenido de InfluxDB que contiene la información temporal correspondiente. La clase ofrece métodos para agregar, obtener y eliminar archivos CSV de la base de datos.

La clase **TrainModels** representa una funcionalidad dentro del sistema que se encarga de gestionar y almacenar modelos preentrenados. Al igual que la clase **Database**, esta clase tiene asociados diferentes tipos de datos, como temperatura, luminosidad, presión y humedad. Proporciona métodos para agregar y obtener modelos preentrenados. Al agregar un modelo, se carga desde un archivo y se almacena en la base de datos de modelos. La clase brinda así una interfaz para la gestión de modelos dentro del Gemelo Digital.

## 5.2 Procesar los datos

La clase **ProcessedData** se encarga de procesar la información del sensor según el tipo de datos proporcionados, como temperatura, luminosidad, presión o humedad. Esta clase modifica la estructura de los datos para adaptarla al formato requerido por el modelo. Dependiendo del tipo de datos, se realizan distintas operaciones.

Para el tipo 'luminosity', se convierte la columna Fecha a un formato de fecha, se renombran las columnas y se ajusta el índice.

Para el tipo 'temperature', se realiza una copia del conjunto de datos original, se convierte la columna de fecha a un formato específico, se ajusta el índice y se eliminan columnas innecesarias.

Para el tipo 'pressure', se convierte la columna de fecha, se ajusta el índice y se selecciona un rango específico de datos.

Para el tipo 'humidity', se realiza un proceso similar al de 'temperature', eliminando columnas innecesarias y seleccionando un rango específico.

Cada vez que se modifica la estructura, la clase almacena el DataFrame resultante en un atributo. La clase retorna el DataFrame modificado.

## 5.3 Predicciones

Para las clases de predicciones **PredictionTemperature**, **PredictionHumidity**, **PredictionLuminosity** y **PredictionPressure** tenemos los siguientes métodos:

**\_\_init\_\_**: El método de inicialización de la clase. Define varios atributos, como el tipo de datos (**data\_type**), el valor predicho (**predicted\_value**), los datos procesados (**processed\_df**), y otros atributos relacionados con la preparación y entrenamiento del modelo.

**create\_sequences**: Esta función toma una secuencia de datos y crea secuencias de entrada y objetivos para el entrenamiento del modelo. Específicamente, toma una secuencia de longitud **seq\_length** y crea secuencias de entrada y salida.

**prepareData:** Esta función se encarga de preparar los datos para el entrenamiento del modelo. Escala los datos utilizando **StandardScaler**, divide los datos en conjuntos de entrenamiento y prueba, y crea secuencias para el modelo.

**createModel:** Crea una instancia del modelo Transformer (**ImprovedTimeSeriesTransformer**) con los hiperparámetros especificados. Llama a **prepareData** para asegurarse de que los datos estén listos para el entrenamiento.

**trainModel:** Entrena el modelo utilizando los datos preparados. Utiliza el optimizador Adam y la función de pérdida Mean Squared Error (MSELoss) para entrenar el modelo durante un número específico de épocas.

**predictValue:** Realiza predicciones para un tiempo específico después del conjunto de prueba. Itera en incrementos de tiempo y utiliza el modelo entrenado para predecir.

**validateModel:** Valida el modelo utilizando el conjunto de prueba y calcula métricas como el Error Absoluto Medio (MAE) y el coeficiente de determinación ( $R^2$ ).

**graphicPredictions:** Muestra gráficamente las predicciones comparadas con los valores reales en el conjunto de prueba.

**predictSequence:** Realiza predicciones secuenciales en intervalos de tiempo especificados por el usuario. Pide al usuario el intervalo y el tiempo total de predicción, y hace predicciones en cada intervalo.

**sendPredictionMQTT:** Publica las predicciones en un servidor MQTT especificado por el usuario.

Estas funciones trabajan juntas para cargar datos, entrenar el modelo, realizar predicciones y validar el rendimiento del modelo. El uso principal es instanciar la clase, entrenar el modelo con **trainModel**, y luego realizar predicciones con funciones como **predictValue** o **predictSequence**.

La clase **PredictionTemperature** está diseñada para realizar predicciones de temperatura utilizando un modelo Transformer. Como se evidenció anteriormente, la variable de temperatura exhibe características estacionarias, lo que sugiere que puede ser abordada mediante el uso de secuencias para realizar predicciones futuras. Las secuencias se han definido con una longitud de 288, ya que las mediciones se registran cada 5 minutos, y



por lo tanto, 288 mediciones equivalen a un día completo. La estrategia adoptada por el modelo consiste en entrenarse para predecir el siguiente valor en la secuencia.

Durante el proceso de predicción en el futuro, se utiliza la última secuencia disponible. La predicción del siguiente valor se realiza sobre esta secuencia, y luego se actualiza la secuencia incorporando esta predicción y eliminando el valor más antiguo, el más distante en el tiempo. Este procedimiento se repite iterativamente hasta alcanzar la fecha solicitada por el usuario.

La clase **PredictionHumidity** está diseñada para realizar predicciones de humedad utilizando un modelo Transformer. La variable de humedad exhibe una menor estacionariedad en comparación con la temperatura, y su comportamiento está fuertemente influenciado por esta última. En consecuencia, hemos adaptado nuestro enfoque de predicción, utilizando secuencias más cortas para reducir el tiempo de ejecución. El modelo de predicción se diseñará para recibir secuencias que contienen mediciones tanto de temperatura como de humedad, y proporcionará predicciones específicamente para la humedad.

Para realizar predicciones futuras, primero invocaremos la función de predicción de temperatura para obtener valores anticipados hasta la fecha deseada. Utilizaremos estos valores para actualizar la última secuencia de humedad y la correspondiente secuencia de temperatura. Siguiendo una lógica similar a la aplicada en la predicción de temperatura, continuaremos ajustando estas secuencias hasta alcanzar la fecha requerida por el usuario. Este enfoque nos permitirá obtener predicciones precisas y sincronizadas de la humedad en función de las predicciones previas de temperatura.

La clase **PredictionPressure** es un Transformer diseñado para hacer predicciones de presión basándose en la temperatura y humedad. La variable de presión, al igual que la humedad, muestra una dependencia con las dos variables previamente mencionadas. Por lo tanto, aplicaremos la misma lógica utilizada para la predicción de humedad en el entrenamiento del modelo. En este caso, crearemos secuencias que incluyan las mediciones de humedad, temperatura y presión, con el objetivo de predecir el siguiente valor de presión.

Para realizar predicciones futuras, seguiremos un procedimiento similar al implementado para la variable de humedad. Inicialmente, invocaremos el modelo de predicción de

humedad para obtener los valores anticipados de humedad y temperatura hasta la fecha deseada. Posteriormente, utilizaremos estos valores para realizar predicciones de presión, actualizando iterativamente las secuencias correspondientes hasta alcanzar la fecha final especificada por el usuario. Este enfoque nos permitirá incorporar la influencia de las variables de humedad y temperatura en las predicciones de presión, proporcionando así estimaciones más precisas y contextualizadas.

La clase **PredictionLuminosity** es un Transformer diseñado para hacer predicciones de luminosidad basándose en valores anteriores. Nos encontramos con dificultades en los sensores al momento de recoger los datos de luminosidad, por lo que hemos procedido a medir esta variable de forma independiente de las demás. Hemos generado secuencias de 288 valores, cada una correspondiente a un día completo, y a partir de ellas hemos realizado predicciones para el valor subsiguiente.

## 5.4 Graficar

La clase **Graphics** ha sido diseñada para simplificar la tarea de visualizar y comparar datos asociados a las distintas variables. Su funcionalidad se centra en dos tipos principales de gráficas. En primer lugar, ofrece la capacidad de representar gráficamente las tendencias temporales de los datos mediante la aplicación del filtro HP (Hodrick-Prescott). Este método, denominado **displayTrends**, ajusta su operación según el tipo de variable que se está analizando.

El segundo tipo de gráfica, realizado mediante el método **comparePredictions**, se centra en la comparación de predicciones generadas por los modelos específicos con los valores reales del conjunto de prueba. Aquí, la clase **Graphics** utiliza las clases de predicción correspondientes (por ejemplo, **PredictionTemperature** para la temperatura) para generar predicciones y luego emplea el método **graphicPredictions** de esas clases para visualizar la comparación entre las predicciones y los valores reales.

## 5.5 Conexión con el sensor real

La clase **SensorMQTT** está diseñada para establecer conexión y recibir datos de un sensor en tiempo real a través de un servidor MQTT. Su objetivo principal es permitir al gemelo digital que pueda mostrar tanto los valores predichos por los modelos como los valores reales provenientes del sensor en sí. Al conectarse al sensor real, esta clase facilita la integración de datos en tiempo real.

## 5.6 Transformer

La clase **ImprovedTimeSeriesTransformer** define la arquitectura del modelo Transformer para la predicción de series temporales. Esta estructura es empleada en el contexto de las clases de predicción, donde se entrena y utiliza para hacer predicciones futuras.

## 5.7 Clases sensores

Las clases de sensores (**LuminositySensor**, **TemperatureSensor**, **HumiditySensor**, **PressureSensor**) actúan como intermediarios entre el **DigitalTwinSystem** y las clases de predicción (**PredictionTest**, **PredictionHumidity**, **PredictionPressure**, **PredictionLuminosity**). Estas clases de sensores gestionan el traspaso de datos y modelos entrenados entre la clase principal y la correspondiente clase de predicción.

Cada una de estas clases tiene una función **get\_value(self)** para gestionar las predicciones, y otra **get\_train\_model(self)** para el entrenamiento de los modelos.

## 5.8 Sistema del gemelo digital

La clase **DigitalTwinSystem** funciona como la clase principal que controla y coordina las diferentes funcionalidades y clases relacionadas con el sistema digital gemelo. A continuación, se proporciona una descripción general de las funciones de esta clase:

1. **\_\_init\_\_(self)**: El método de inicialización define varias variables de instancia para almacenar datos sensoriales, datos procesados, modelos de predicción y el tipo de sensor seleccionado.

2. **select\_sensor\_type(self, selected\_type):** Este método permite seleccionar el tipo de sensor que se utilizará y actualiza la variable `self.type` en consecuencia.
3. **getData(self, type):** Este método obtiene los datos del sensor del tipo especificado. Si los datos ya han sido cargados, los devuelve; de lo contrario, los carga desde la base de datos.
4. **getModel(self, type):** Este método obtiene el modelo de predicción del tipo especificado. Si el modelo ya ha sido cargado, lo devuelve; de lo contrario, carga el modelo utilizando la función **get\_model**.
5. **getProcessedData(self, type):** Este método obtiene los datos procesados del tipo especificado utilizando la clase **ProcessedData**. Si los datos ya han sido procesados, los devuelve; de lo contrario, los procesa y los almacena para uso futuro.
6. **getPredictions(self):** Este método realiza predicciones utilizando el modelo correspondiente al tipo de sensor seleccionado. Invoca las clases **LuminositySensor**, **TemperatureSensor**, **HumiditySensor**, o **PressureSensor** según el tipo de sensor para obtener y mostrar las predicciones.
7. **getTrainModel(self, type):** Este método carga o entrena un modelo para el tipo de sensor especificado. Invoca las clases correspondientes de entrenamiento si el modelo aún no ha sido entrenado.
8. **getGraphics(self):** Este método permite al usuario seleccionar el tipo de gráfico a visualizar: tendencia o comparación de predicciones. Luego, utiliza la clase **Graphics** para mostrar el gráfico correspondiente.
9. **validateModel(self):** Este método valida el modelo actualmente cargado para el tipo de sensor seleccionado. Utiliza las clases de predicción (**PredictionTemperature**, **PredictionHumidity**, **PredictionPressure**, **PredictionLuminosity**) para realizar la validación del modelo.
10. **getSequences(self):** Este método realiza predicciones de tal manera que simula el comportamiento real del sensor. Invoca las clases **PredictionTemperature**, **PredictionHumidity**, **PredictionPressure** o **PredictionLuminosity** según el tipo de sensor.

11. **callSensor(self):** Este método inicia la conexión al sensor MQTT utilizando la clase `SensorMQTT` en un hilo separado. La conexión se realiza mediante el protocolo MQTT, y los datos del sensor se reciben y procesan en tiempo real.

La clase **DigitalTwinSystem** actúa como la interfaz principal para controlar las operaciones del sistema, coordinando la obtención de datos, la selección y entrenamiento de modelos, la visualización de gráficos y la interacción con el sensor en tiempo real.

## 5.9 Interfaz de usuario

Se implementa una interfaz gráfica de usuario (GUI) en PyQt5 para interactuar con el sistema del gemelo digital. Se utiliza un hilo de ejecución (`QThread`) para realizar mediciones sin bloquear la interfaz, permitiendo operaciones asincrónicas. Como es el caso de conectarnos al sensor real y poder ver las mediciones en tiempo real.



FIGURA 16: Interfaz de usuario al conectarse con el sensor real

La interfaz proporciona múltiples opciones, incluyendo la selección de tipos de sensor (luminosidad, temperatura, humedad, presión) y funciones específicas (sensor real,

validación de modelo, gráficos, predicciones, simulación de sensor) mediante menús desplegables.

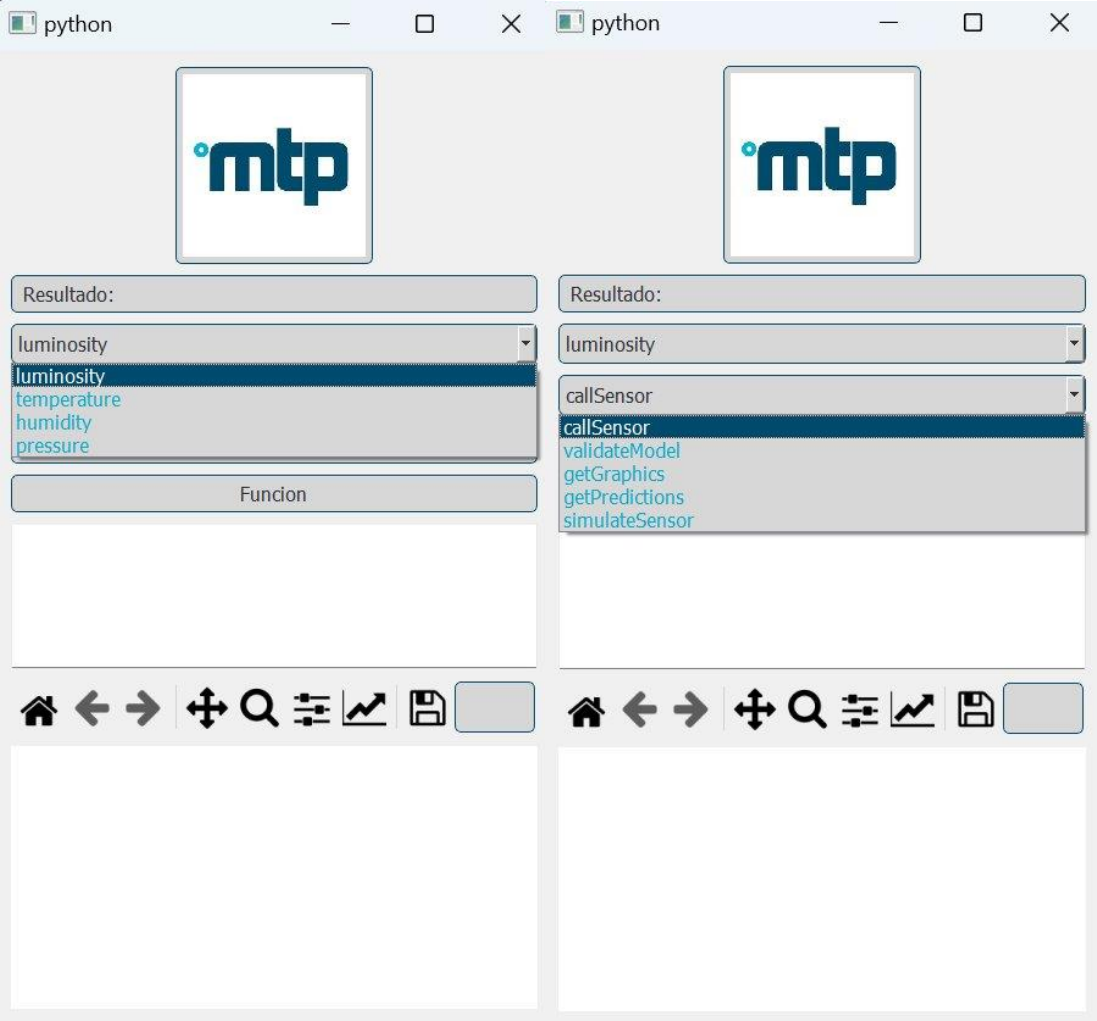


FIGURA 17: Desplegable de la interfaz para seleccionar el tipo

FIGURA 18 Desplegable de la interfaz para seleccionar la función

Los usuarios pueden especificar detalles para las predicciones, como fecha y hora de la predicción deseada.

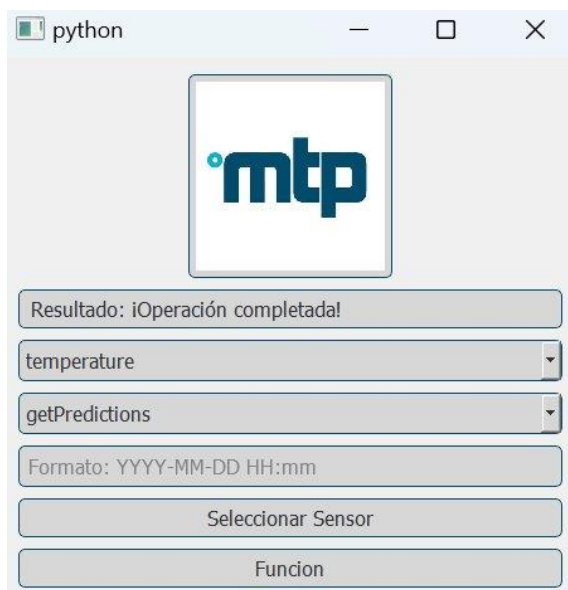


FIGURA 19: Aparición del cuadro de fecha



FIGURA 20: Interfaz de usuario para predicciones futuras

Se pueden visualizar gráficos que no solo permitirán comparar las predicciones con los datos reales, sino también analizar las tendencias a lo largo del tiempo. Esta dualidad en la visualización de datos es esencial para una comprensión profunda del rendimiento del modelo y la precisión de las predicciones.

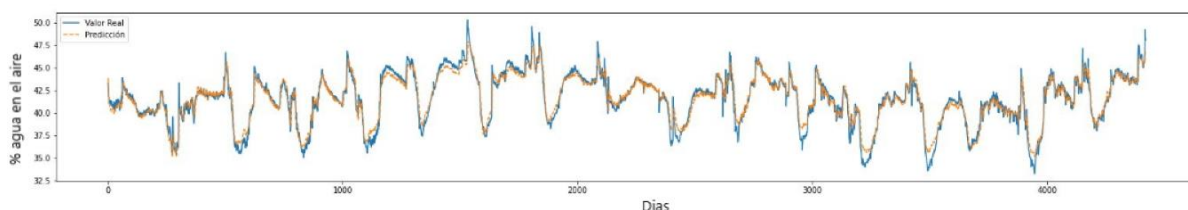


FIGURA 21: Grafico de humedad generado al seleccionar comparación de predicciones

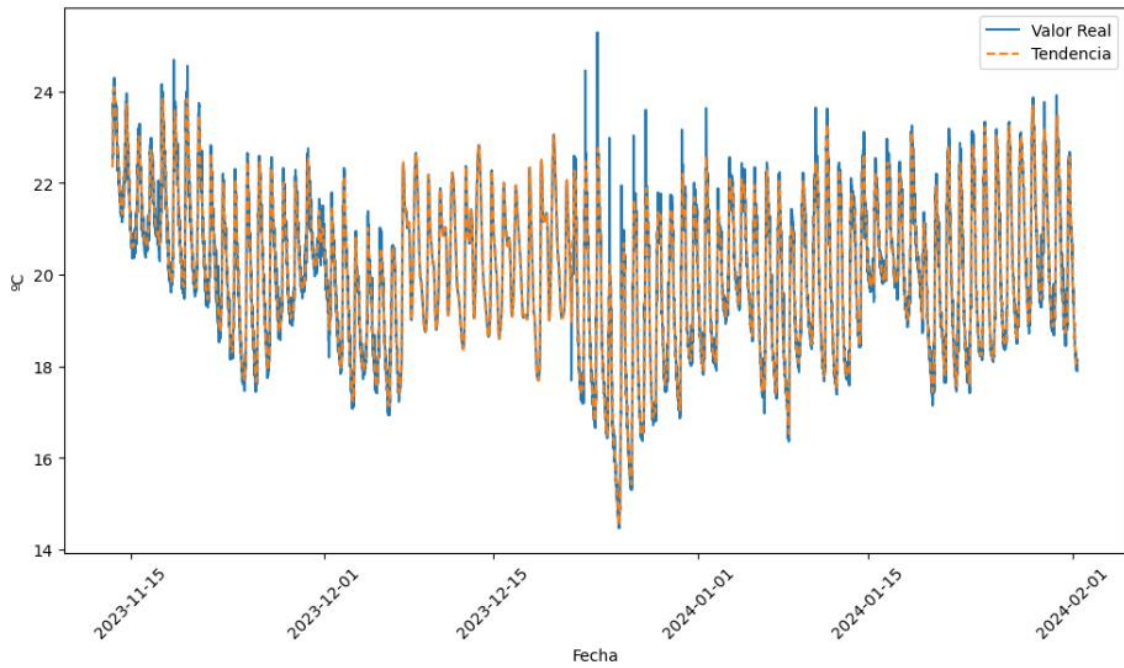


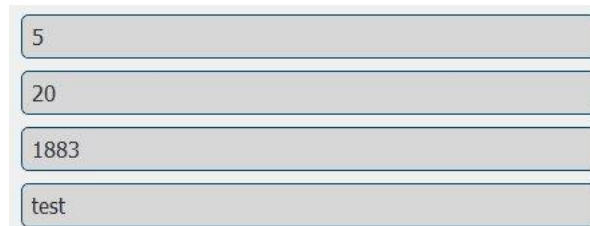
FIGURA 22: Grafico de temperatura al seleccionar tendencia

Para simular las acciones del sensor real se nos abre unas opciones para poder escribir los parámetros deseados.

FIGURA 23: Apartados al seleccionar simulateSensor



Ingresamos las características deseadas y las predicciones se irán publicando en el topic y puerto deseado.



A vertical stack of four input fields. The first field contains the number '5'. The second field contains the number '20'. The third field contains the number '1883'. The fourth field contains the text 'test'.

FIGURA 24: Valores ingresados para simulateSensor

```
C:\Program Files\mosquitto>mosquitto_sub -t test -p 1883
Prediction at Value 20.65591
Name: 2024-02-06 13:49:00, dtype: float64: 20.655909554362296
Prediction at Value 20.655483
Name: 2024-02-06 13:54:00, dtype: float64: 20.655482951402664
Prediction at Value 20.655057
Name: 2024-02-06 13:59:00, dtype: float64: 20.65505718410015
Prediction at Value 20.654632
Name: 2024-02-06 14:04:00, dtype: float64: 20.654631834626198
```

FIGURA 25: Suscriptor MQTT para los valores ingresados

## **6. Validación y verificación**

La validación de un Gemelo Digital es un proceso esencial para garantizar que la representación digital de un objeto o sistema refleje con precisión su estado y comportamiento en el mundo real. Para llevar a cabo una validación efectiva, es crucial seguir una serie de criterios que abarquen desde la precisión del modelo de simulación hasta la integridad de los datos de entrada. Este proceso implica una exhaustiva revisión de la coherencia entre el Gemelo Digital y su contraparte física, así como pruebas exhaustivas en diversos escenarios.

### **6.1 Comparación con el objeto físico**

Para maximizar la precisión de nuestro gemelo digital, lo hemos diseñado para replicar de manera fiel las funcionalidades de nuestro sensor físico. Esto incluye la integración de capacidades avanzadas, como la conexión en tiempo real con el sensor, permitiéndonos acceder a mediciones precisas al instante y de forma continua. Una característica clave que hemos implementado es el uso del protocolo MQTT, el mismo que utiliza nuestro sensor físico para publicar datos. Esta funcionalidad es esencial, ya que permite que nuestro gemelo digital reproduzca fielmente las operaciones del sensor, incluida la publicación de predicciones en los intervalos programados. Esta réplica exacta de las funcionalidades básicas asegura que nuestro gemelo digital no solo simule el comportamiento del sensor en tiempo real, sino que también ofrezca una herramienta invaluable para la monitorización y análisis de datos, reforzando así nuestra capacidad para realizar ajustes y mejoras basadas en información precisa y actualizada.

### **6.2 Datos de entrada**

A la hora de obtener los datos tenemos que ser cuidadosos, nuestros datos deben ser ordenados y limpios, sin ruido, por lo que seguiremos unas ciertas medidas. A la hora de programar el sensor real para recopilar datos, se integran unas ciertas medidas de seguridad como puede ser QoS. La implementación de QoS permite optimizar el rendimiento de la red, proporcionar diferentes niveles de rendimiento y prioridad a diferentes tipos de tráfico de datos. La elección de QoS 2 refleja nuestro compromiso con la calidad de los datos recopilados, proporcionando una base sólida y confiable para la

toma de decisiones basada en datos y el monitoreo preciso del entorno que nuestro gemelo digital busca representar. Esta aproximación subraya la importancia de la fiabilidad por encima de la velocidad en el contexto de nuestra aplicación, asegurando que la calidad y exactitud de la información no se vean comprometidas en ningún momento.

### 6.3 Modelo de simulación

Vamos a ver diferentes métodos y técnicas empleados para validar modelos. Para cada medición hemos entrenado modelos distintos como ya hemos visto anteriormente.

- **TEMPERATURA:** En el desarrollo de este modelo, hemos empleado secuencias para generar predicciones. Para su validación, dividimos el conjunto de datos en grupos de entrenamiento (train) y prueba (test), aplicando las métricas de validación más reconocidas en análisis de regresión, tal como se detalla en la sección 8.4.

#### **MAE Global (Mean Absolute Error):**

Valor: 0.12189

En este caso, el MAE indica que, en promedio, las predicciones del modelo se desvían aproximadamente 0.12189 unidades de los valores reales en el conjunto de prueba. Un MAE bajo indica que el modelo tiene un buen rendimiento en términos de precisión, ya que las predicciones tienden a estar muy cerca de los valores reales en promedio. Por lo que, un MAE de 0.12189 sugiere que el modelo tiene una precisión bastante alta.

#### **MSE Global (Mean Squared Error):**

Valor: 0.02861

En este caso, el MSE indica que, en promedio, el cuadrado de la diferencia entre las predicciones y los valores reales es de aproximadamente 0.02861 unidades. Un MSE bajo indica que el modelo tiene un buen rendimiento en términos de precisión, ya que las predicciones tienden a estar muy cerca de los valores reales. Por lo que, un MSE de 0.02861 sugiere que el modelo tiene una muy buena precisión.

#### **RMSE Global (Root Mean Squared Error):**

Valor: 0.16916

En este caso, el RMSE indica que, en promedio, las predicciones del modelo se desvían aproximadamente 0.16916 unidades de los valores reales en el conjunto de prueba. Al igual que el MAE y el MSE, un RMSE bajo indica que el modelo tiene un buen rendimiento en términos de precisión. Por lo que, un RMSE de 0.16916 sugiere que el modelo tiene una precisión bastante alta.

### **R<sup>2</sup> Global (Coeficiente de determinación):**

Valor: 0.98870

En este caso, un R<sup>2</sup> de 0.98870 indica que aproximadamente el 98.87% de la variabilidad en la variable dependiente está explicada por el modelo. Un R<sup>2</sup> cercano a 1 sugiere que el modelo se ajusta muy bien a los datos y que las predicciones del modelo son muy cercanas a los valores reales. En este caso, un R<sup>2</sup> de 0.98870 indica que el modelo tiene un ajuste excelente a los datos y una capacidad de predicción muy fuerte.

### **Conclusiones:**

Los resultados de las métricas de evaluación del modelo son altamente alentadores. El modelo ha demostrado un rendimiento excepcionalmente alto en términos de precisión y capacidad predictiva en el conjunto de prueba evaluado.

- **Precisión de las predicciones:** El modelo ha demostrado tener una alta precisión en sus predicciones, con valores bajos de MAE, MSE y RMSE. Esto indica que, en promedio, las predicciones del modelo están muy cerca de los valores reales en el conjunto de prueba.
- **Baja dispersión de errores:** El valor bajo del RMSE sugiere que la dispersión de los errores del modelo es pequeña, lo que significa que las predicciones tienden a estar muy cerca de los valores reales y hay una consistencia en la precisión de las predicciones en todo el conjunto de datos.
- **Capacidad explicativa:** El alto R<sup>2</sup> indica que el modelo explica una gran proporción de la variabilidad en la variable dependiente. Esto sugiere que el modelo captura bien las relaciones subyacentes en los datos y es capaz de predecir con precisión la variable objetivo.

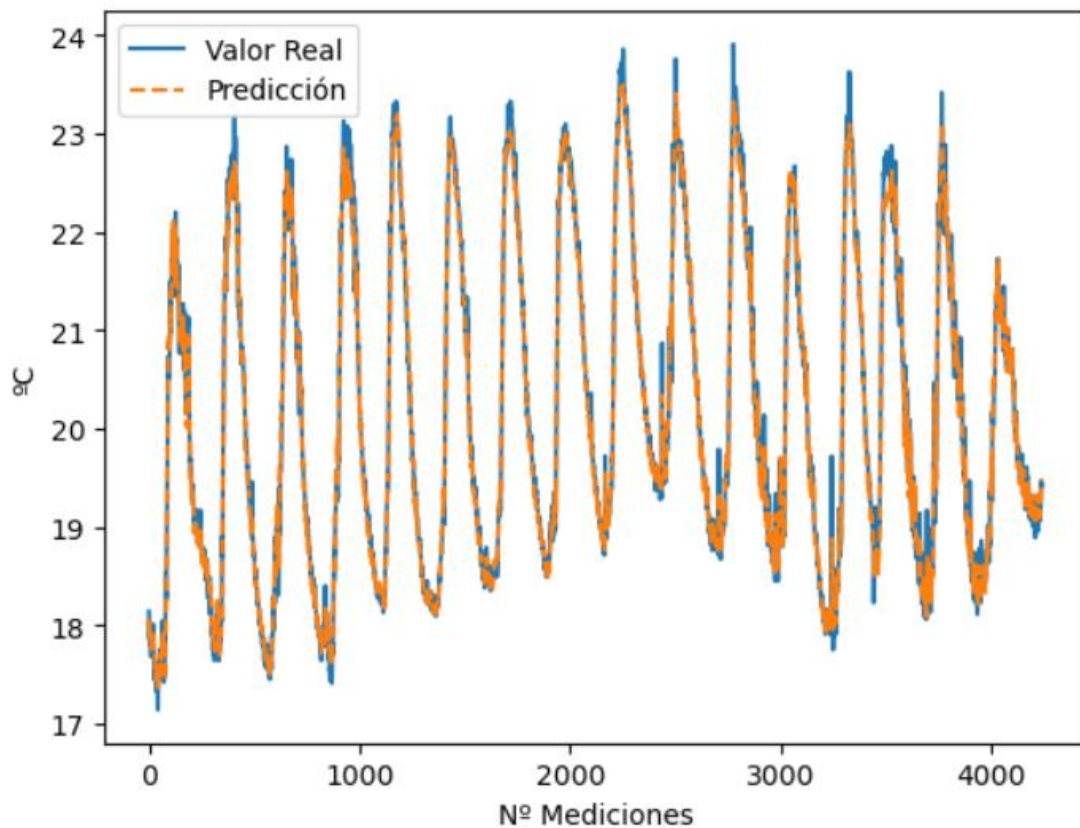


FIGURA 26: Grafica de temperatura predicciones frente a test

Podemos ver como nuestras predicciones se ajustan a la perfección a los valores reales.

- **HUMEDAD:** Las predicciones de humedad se realizan no solo en base a valores anteriores de humedad sino también considerando registros históricos de temperatura. Para evaluar la precisión de este modelo, hemos aplicado idénticas métricas de validación utilizadas previamente en el análisis del modelo de temperatura.

#### **MAE Global (Mean Absolute Error):**

Valor: 0.41955

En este caso, el MAE indica que, en promedio, las predicciones del modelo se desvían aproximadamente 0.41955 unidades de los valores reales en el conjunto de prueba. Un MAE bajo indica que el modelo tiene un buen rendimiento en términos de precisión, ya que las predicciones tienden a estar muy cerca de los valores reales en promedio. Por lo que, un MAE de 0.41955 sugiere que el modelo tiene una precisión bastante alta.

#### **MSE Global (Mean Squared Error):**

Valor: 0.37066

Descripción: En este caso, el MSE indica que, en promedio, el cuadrado de la diferencia entre las predicciones y los valores reales es de aproximadamente 0.37066 unidades. Un MSE bajo indica que el modelo tiene un buen rendimiento en términos de precisión, ya que las predicciones tienden a estar muy cerca de los valores reales. Por lo que, un MSE de 0.37066 sugiere que el modelo tiene una muy buena precisión.

#### **RMSE Global (Root Mean Squared Error):**

Valor: 0.60882

En este caso, el RMSE indica que, en promedio, las predicciones del modelo se desvían aproximadamente 0.60882 unidades de los valores reales en el conjunto de prueba. Al igual que el MAE y el MSE, un RMSE bajo indica que el modelo tiene un buen rendimiento en términos de precisión. Por lo que, un RMSE de 0.60882 sugiere que el modelo tiene una precisión bastante alta.

#### **R<sup>2</sup> Global (Coeficiente de determinación):**

Valor: 0.96620

En este caso, un R2 de 0.96620 indica que aproximadamente el 96.62% de la variabilidad en la variable dependiente está explicada por el modelo. Un R2 cercano a 1 sugiere que el modelo se ajusta muy bien a los datos y que las predicciones del modelo son muy cercanas a los valores reales. En este caso, un R2 de 0.96620 indica que el modelo tiene un ajuste excelente a los datos y una capacidad de predicción muy fuerte.

Los resultados obtenidos de las métricas de evaluación del modelo son sumamente prometedores. El modelo ha demostrado un rendimiento extraordinariamente alto en cuanto a precisión y habilidad predictiva en el conjunto de datos de prueba analizado.

- **Precisión en las Predicciones:** El modelo ha exhibido una precisión notable en sus predicciones, evidenciada por los valores bajos de MAE, MSE y RMSE. Estos indicadores reflejan que, en promedio, las predicciones del modelo se encuentran altamente alineadas con los valores reales presentes en el conjunto de datos de prueba.
- **Escasa Dispersión de Errores:** La magnitud reducida del RMSE sugiere que la dispersión de los errores del modelo es mínima. Esto implica que las predicciones del modelo mantienen una proximidad considerable con los valores reales a lo

largo de todo el conjunto de datos, lo que indica coherencia en la precisión de las predicciones.

- **Capacidad Explicativa:** El elevado valor de  $R^2$  indica que el modelo es capaz de explicar una proporción significativa de la variabilidad presente en la variable dependiente. Esta capacidad explicativa subraya la efectividad del modelo para capturar las relaciones fundamentales en los datos y para predecir con precisión la variable objetivo.

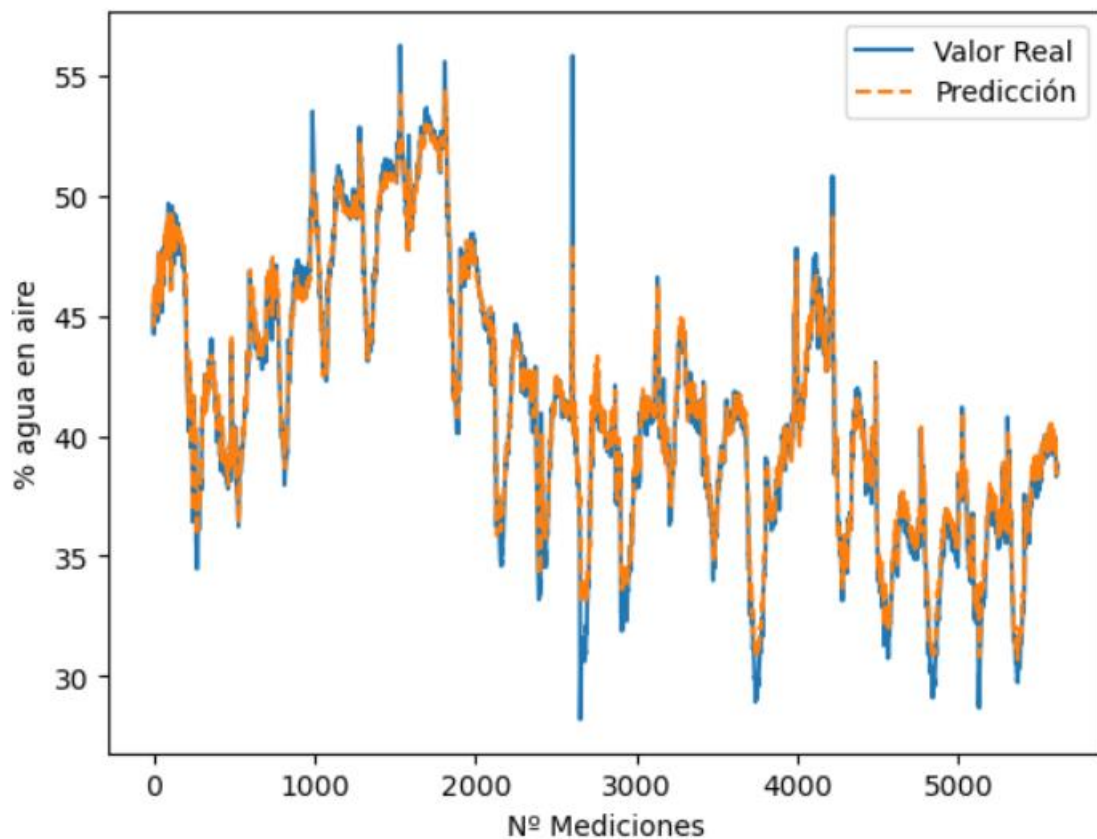


FIGURA 27: Grafica de humedad predicciones frente a test

Como se observa, el modelo enfrenta mayores desafíos al intentar predecir valores extremos. Sin embargo, logra replicar con gran precisión los valores reales.

- **PRESIÓN:** Para las predicciones de presión nos basamos en mediciones antiguas de temperatura y humedad.

#### MAE Global (Mean Absolute Error):

Valor: 1.42603

En este escenario, el MAE señala que, en promedio, las predicciones del modelo desvían aproximadamente 1.42603 unidades de los valores reales en el conjunto de prueba. Un MAE bajo sugiere que el modelo exhibe un buen desempeño en términos de precisión, dado que las predicciones tienden a acercarse considerablemente a los valores reales. Por ende, un MAE de 1.42603 indica una precisión notablemente alta del modelo.

#### **MSE Global (Mean Squared Error):**

Valor: 2.97312

En este contexto, el MSE indica que, en promedio, el cuadrado de la diferencia entre las predicciones y los valores reales es de alrededor de 2.97312 unidades. Un MSE bajo implica que el modelo presenta un rendimiento adecuado en términos de precisión, ya que las predicciones tienden a mantenerse muy cerca de los valores reales. Por lo tanto, un MSE de 2.97312 sugiere una precisión bastante alta del modelo.

#### **RMSE Global (Root Mean Squared Error):**

Valor: 1.72427

Dentro de este análisis, el RMSE indica que, en promedio, las predicciones del modelo se desvían aproximadamente 1.72427 unidades de los valores reales en el conjunto de prueba. De manera similar al MAE y al MSE, un RMSE bajo indica que el modelo exhibe un buen desempeño en términos de precisión. Por lo tanto, un RMSE de 1.72427 sugiere una precisión considerablemente alta del modelo.

#### **R<sup>2</sup> Global (Coeficiente de determinación):**

Valor: 0.97581

En este caso, un R<sup>2</sup> de 0.97581 indica que aproximadamente el 97.58% de la variabilidad en la variable dependiente es explicada por el modelo. Un R<sup>2</sup> cercano a 1 sugiere que el modelo se ajusta de manera óptima a los datos y que las predicciones del modelo son muy precisas en comparación con los valores reales. Por consiguiente, un R<sup>2</sup> de 0.97581 denota un ajuste excepcional del modelo a los datos y una capacidad predictiva muy fuerte.

- **Precisión en las Predicciones:** El modelo ha demostrado una precisión sobresaliente en sus estimaciones, evidenciada por los valores reducidos de MAE,



MSE y RMSE. Estos indicadores revelan que, en promedio, las predicciones del modelo están estrechamente alineadas con los valores reales en el conjunto de datos de prueba.

- **Dispersión de Errores Limitada:** El tamaño reducido del RMSE sugiere que la variabilidad en los errores del modelo es mínima. Esto implica que las estimaciones del modelo se mantienen consistentemente cercanas a los valores reales en todos los puntos del conjunto de datos, lo que sugiere una estabilidad en la precisión de las predicciones.
- **Capacidad Explicativa:** El alto  $R^2$  señala que el modelo tiene la habilidad de explicar una proporción significativa de la variabilidad en la variable dependiente. Esta capacidad de explicación resalta la habilidad del modelo para capturar las relaciones subyacentes en los datos y para predecir con precisión la variable objetivo.

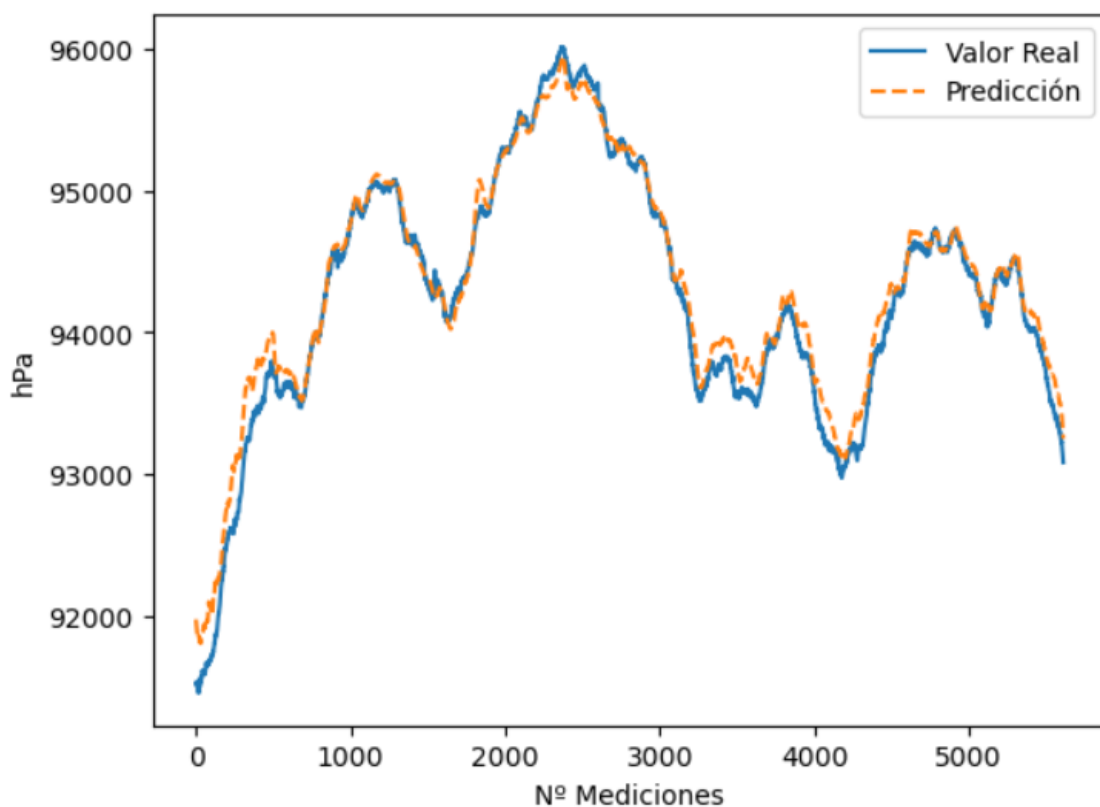


FIGURA 28: Grafica de presión predicciones frente a test

Aunque no lo parezca en esta grafica hay representados tres días y medio, como podemos ver los datos no son nada estacionarios. Por lo que consideramos que nuestro modelo recoge muy bien las variaciones de presión.

- **LUMINOSIDAD:** La recopilación de datos para este modelo presentó mayores desafíos, debido a la ausencia de un sensor que registrara de manera continua. La limitada disponibilidad de datos, sumada al amplio rango de valores que oscilan entre 0 y 1200, resultó en las dificultades que enfrentamos.

**MAE Global (Mean Absolute Error):**

Valor: 76.12

Este valor indica que, en promedio, las predicciones del modelo se desvían aproximadamente 76.12 unidades del valor real. Dado el extenso rango de valores objetivo, este nivel de error absoluto medio es relativamente bajo, lo que sugiere que el modelo es capaz de realizar predicciones con un alto grado de precisión en una escala amplia.

**MSE Global (Mean Squared Error):**

Valor: 12617.42

El MSE, al elevar al cuadrado los errores antes de promediarlos, penaliza más fuertemente las predicciones que están lejos del valor real. Un MSE de 12617.42, mientras parece alto, debe considerarse en el contexto del rango de valores objetivo. Este valor nos indica que, aunque hay casos de predicciones con desviaciones significativas, en general, el modelo gestiona bien la variabilidad en un conjunto de datos con una amplia gama de valores.

**RMSE Global (Root Mean Squared Error):**

Valor: 112.33

El RMSE proporciona una medida de la dispersión de los errores en las mismas unidades que los valores predichos. Un RMSE de 112.33, en relación con el rango total de 0 a 1200, demuestra que el modelo mantiene una dispersión de error limitada, lo que refleja su robustez y la consistencia de sus predicciones a través de este amplio espectro de valores.

**R<sup>2</sup> Global (Coeficiente de determinación):**

Valor: 0.9371

Este coeficiente, con un valor cercano a 1, indica que el modelo tiene una capacidad explicativa excepcionalmente alta. Un R<sup>2</sup> de 0.937 sugiere que aproximadamente el 93.7% de la variabilidad en los datos observados puede ser explicada por el modelo. Este

nivel de ajuste es particularmente impresionante dada la complejidad inherente a predecir comportamientos en un rango tan amplio de valores, reafirmando la eficacia del modelo en capturar y simular las dinámicas subyacentes del comportamiento del usuario.

- **Precisión en las Predicciones:** Nuestro modelo ha demostrado una precisión sobresaliente en sus estimaciones, evidenciada por un MAE Global de 76.1173, un MSE Global de 12617.41796875, y un RMSE Global de 112.3273. A pesar del amplio rango de valores objetivo de 0 a 1200, estos indicadores revelan que, en promedio, las predicciones del modelo están estrechamente alineadas con los valores reales en el conjunto de datos de prueba. Esta alineación sugiere que el modelo es altamente efectivo en predecir el comportamiento del usuario con una gran precisión.
- **Dispersión de Errores Limitada:** La magnitud del RMSE, siendo 112.3273, sugiere que la variabilidad en los errores del modelo es mínima, especialmente considerando el amplio rango de valores objetivo. Esto indica que las estimaciones del modelo se mantienen consistentemente cercanas a los valores reales a lo largo de todo el conjunto de datos, demostrando una estabilidad notable en la precisión de las predicciones. Tal consistencia es indicativa de un modelo confiable, capaz de generar estimaciones precisas en una variedad de escenarios de uso.
- **Capacidad Explicativa:** Con un  $R^2$  Global de 0.9371, nuestro modelo exhibe una capacidad excepcional para explicar una proporción significativa de la variabilidad en la variable dependiente. Este alto coeficiente de determinación resalta la habilidad del modelo para capturar fielmente las relaciones subyacentes dentro de los datos y predecir con precisión la variable objetivo. La capacidad del modelo para explicar aproximadamente el 93.7% de la variabilidad en los datos es particularmente impresionante, considerando la diversidad y complejidad de las interacciones del usuario que intenta modelar.

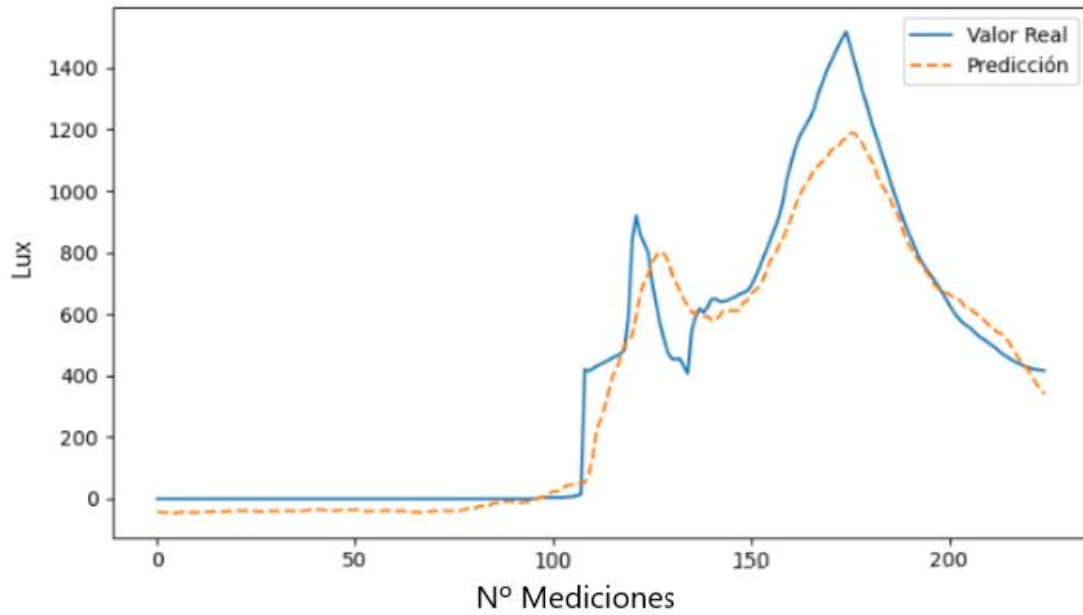


FIGURA 29: Grafica de luminosidad predicciones frente a test

A pesar de la escasez de datos de temperatura mencionada anteriormente, nuestro modelo demuestra una capacidad excepcional para capturar con precisión la tendencia de la luminosidad en función de los datos disponibles.

## 6.4 Predicciones futuras

Al validar nuestras predicciones futuras, enfrentamos ciertas limitaciones inherentes al proceso de proyectar datos hacia el futuro, dado que no disponemos de información real para esas fechas futuras específicas. Para superar este desafío, adoptamos un enfoque más cualitativo y estructurado. Inicialmente, generamos predicciones que se extienden aproximadamente diez días hacia adelante. Luego, procedemos a visualizar estas predicciones mediante gráficos, lo que nos permite realizar un análisis visual crítico de los datos proyectados. Este análisis se centra en evaluar si las predicciones mantienen una distribución lógica y coherente con los patrones observados en los datos históricos reales. Además, examinamos cuidadosamente si las proyecciones siguen la tendencia general identificada en el conjunto de datos existente. Basándonos en esta evaluación meticulosa, que combina el análisis visual con nuestro conocimiento acumulado, determinamos la validez y precisión de las predicciones generadas por el modelo para períodos futuros. Este proceso nos permite calibrar la fiabilidad de nuestras proyecciones y ajustar nuestros modelos conforme sea necesario para asegurar predicciones futuras más acertadas.

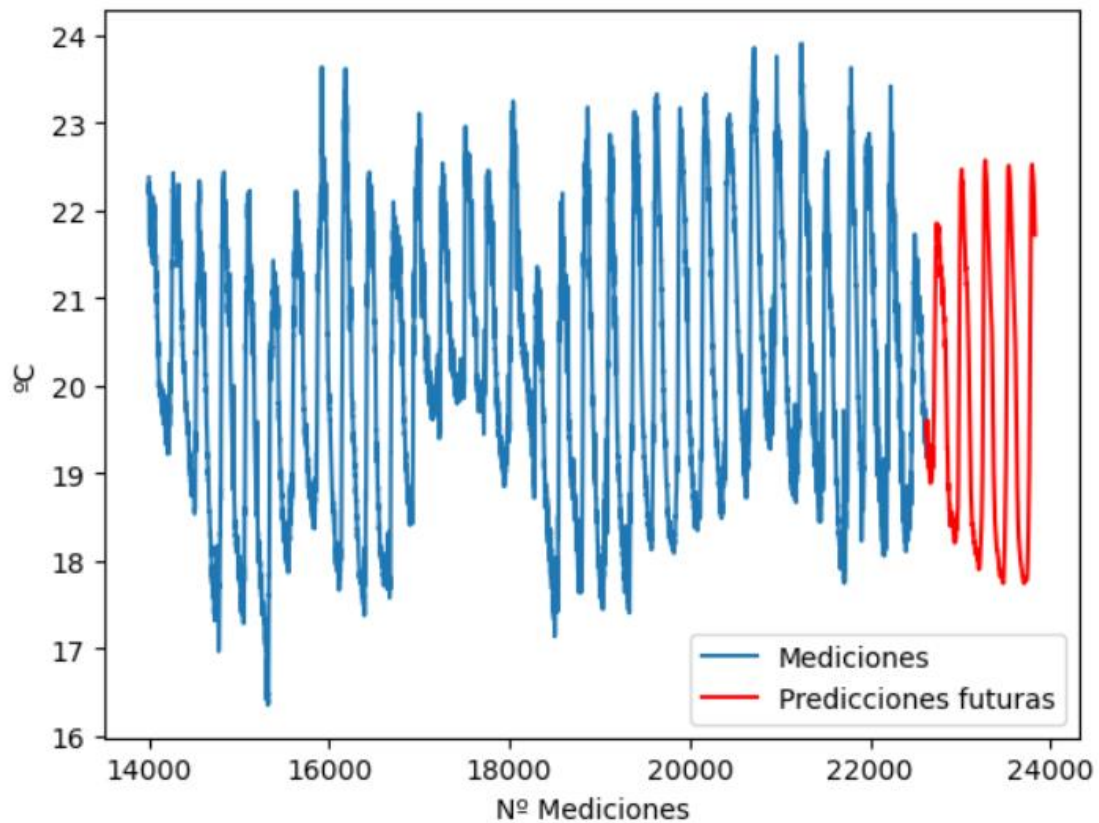


FIGURA 30: Grafica de temperatura para el modelo1 con sus predicciones a futuro

Al observar las predicciones de temperaturas de color rojo, notamos que estas se alinean adecuadamente con el patrón y tendencia observados en los datos. Por lo tanto, consideramos que este modelo es acertado, dado que exhibe excelentes métricas de validación y, además, las proyecciones futuras muestran una notable concordancia con las expectativas.

Por otro lado, tenemos un modelo con una configuración de hiperparámetros similar, con solo algunos ajustes menores. Este modelo también exhibe métricas de validación sobresalientes: un MAE Global de 0.1675, un MSE Global de 0.0463, un RMSE Global de 0.2152, y un  $R^2$  Global de 0.9814. Las predicciones realizadas sobre el conjunto de datos de prueba reflejan esta alta precisión.

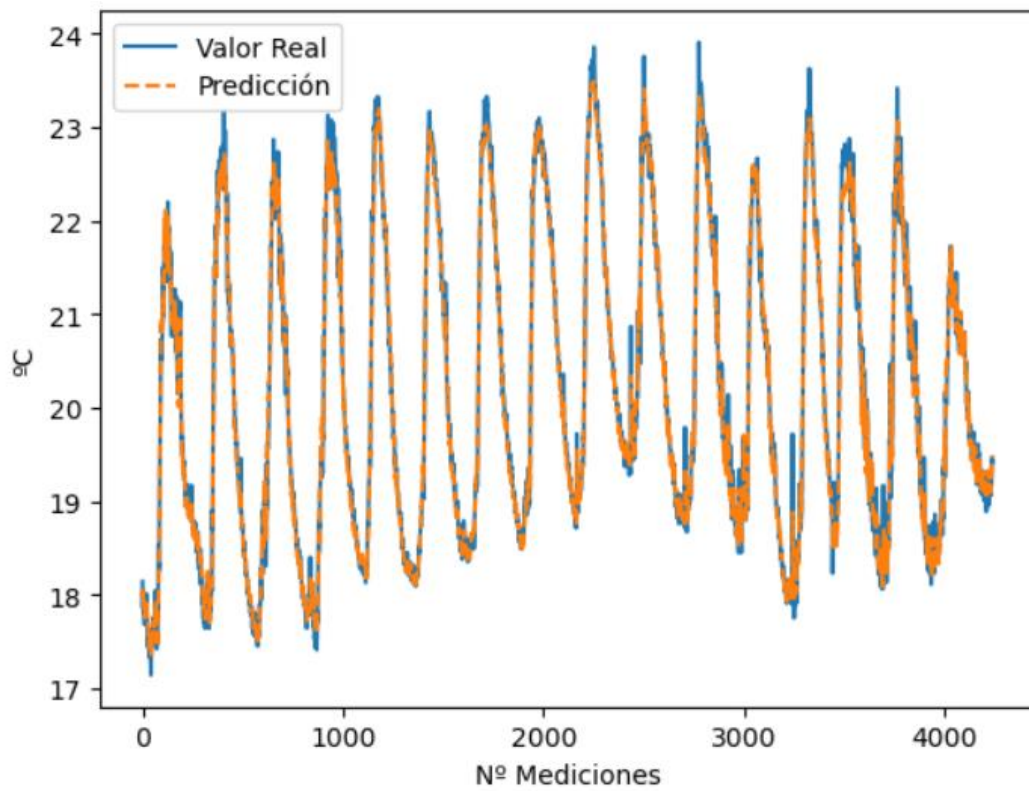


FIGURA 31: Grafica de temperatura test frente a predicciones

Sin embargo, al realizar predicciones a futuro obtenemos lo siguiente:

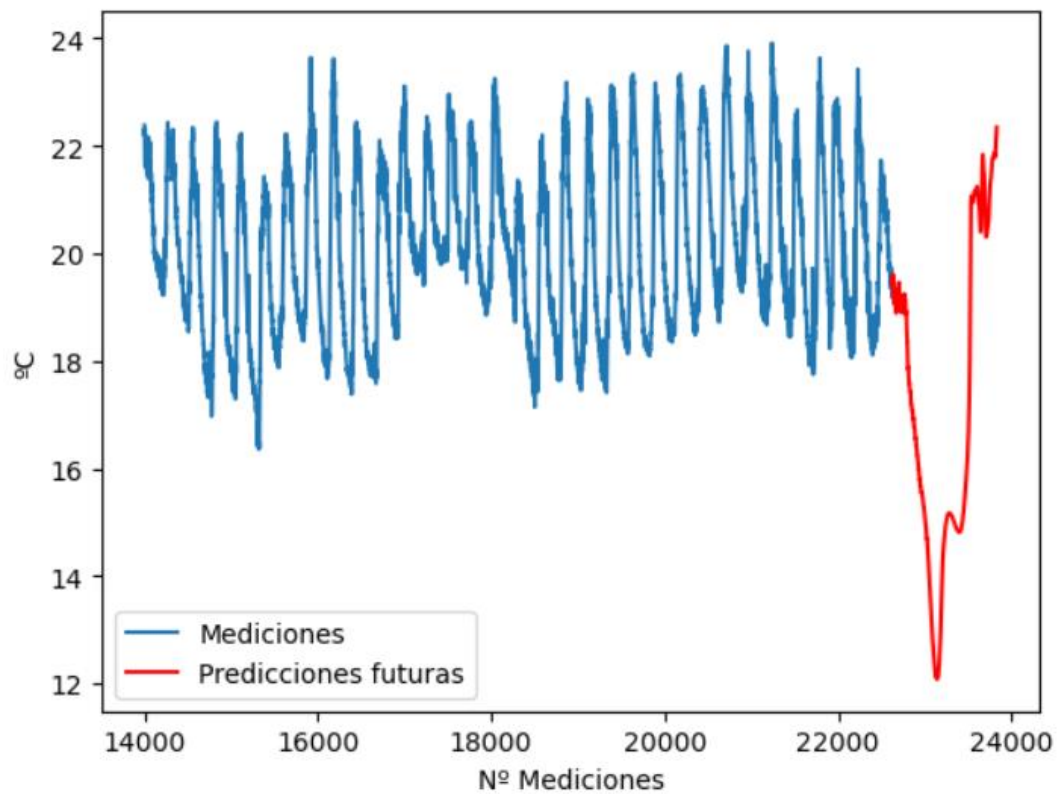


FIGURA 32: Grafica de temperatura para el modelo2 con sus predicciones a futuro

De color rojo, al revisar nuevamente las predicciones a futuro, se observa en la gráfica que, a simple vista, estas no siguen los patrones ni capturan la estacionalidad de las mediciones de temperatura.

Esta situación se presenta debido a que el modelo, en lugar de generalizar correctamente a partir de los datos disponibles, ha incurrido en sobreajuste (overfitting). Esto significa que, aunque el modelo ha aprendido a replicar con alta precisión los datos de entrenamiento, ha perdido la capacidad de predecir de manera precisa en situaciones no vistas anteriormente, como se evidencia en las predicciones a futuro que no siguen los patrones o la estacionalidad observados en las mediciones históricas. Este fenómeno de sobreajuste no se limita solo a las mediciones de temperatura, sino que se extiende a las demás variables analizadas. Por tanto, es crucial ejercer un escrutinio detallado de las predicciones futuras que genera nuestro modelo, asegurándonos de que estas mantengan una coherencia lógica con los datos históricos y evitando caer en el ciclo de sobreajuste que compromete la capacidad del modelo para realizar proyecciones fiables y útiles.

#### **6.4.1 Validación a futuro usando la Transformada de Furier**

Para evaluar la precisión de las predicciones futuras generadas por nuestros modelos, hemos organizado nuestros datos de la siguiente manera: un 80% se ha destinado al entrenamiento (train), un 10% a la prueba (test) y el restante 10% para la validación de las predicciones futuras.

##### **TEMPERATURA:**

Para el modelo de temperatura que mejores resultados nos ha dado en las métricas aplicadas al test haremos predicciones a futuro, es decir añadiendo la propia predicción a la secuencia para la siguiente predicción.

A continuación, contrastamos estas predicciones futuras con el 10% de los datos reales reservados para la validación. Para ello, aplicamos la Transformada de Fourier a ambas series de datos —la predicha y la real— y procedemos a calcular el Error Absoluto Medio (MAE) de las series transformadas. Además, examinamos las fases de las series transformadas para evaluar la similitud en los ángulos entre las predicciones y los valores reales.

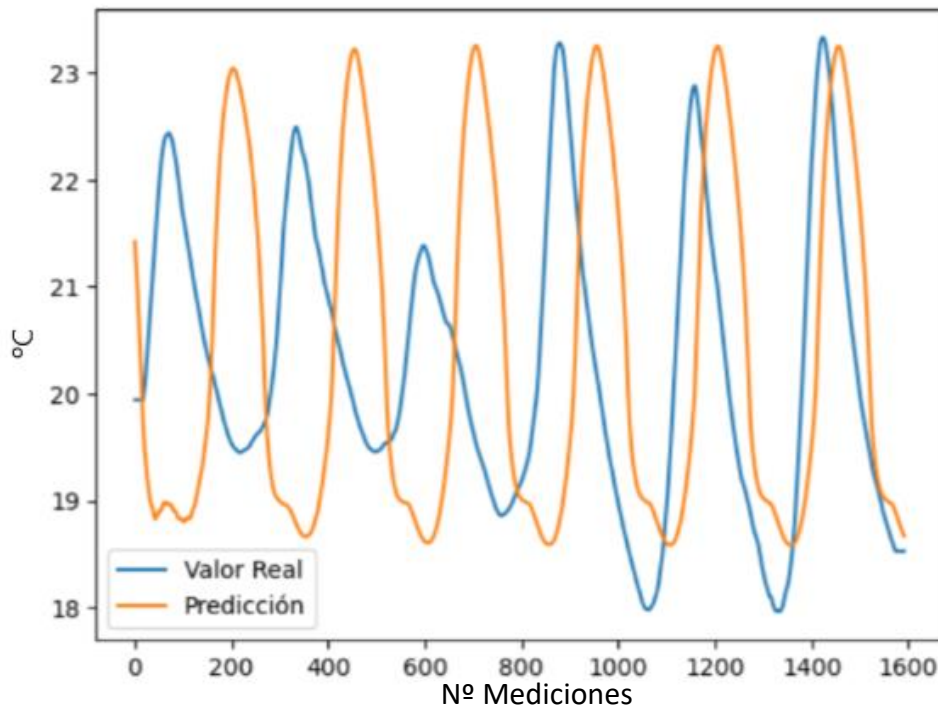


FIGURA 33: Comparación datos de validación predicciones futuras temperatura

Resultados:

- MAE de la Transformada: 4.56
- MAE de la fase(ángulo): 0.13 radianes (12 grados)

Conclusión:

Al realizar predicciones futuras con nuestro modelo, se identifican discrepancias en la predicción exacta de la temperatura. No obstante, el modelo es altamente efectivo en replicar la tendencia general observada en los datos de temperatura reales puesto que solo difiere 12 grados en el ángulo.

HUMEDAD:

Hacemos el mismo proceso para humedad, realizaremos las predicciones futuras y comparamos sus Transformadas de Furier con las de los datos reales de validación.



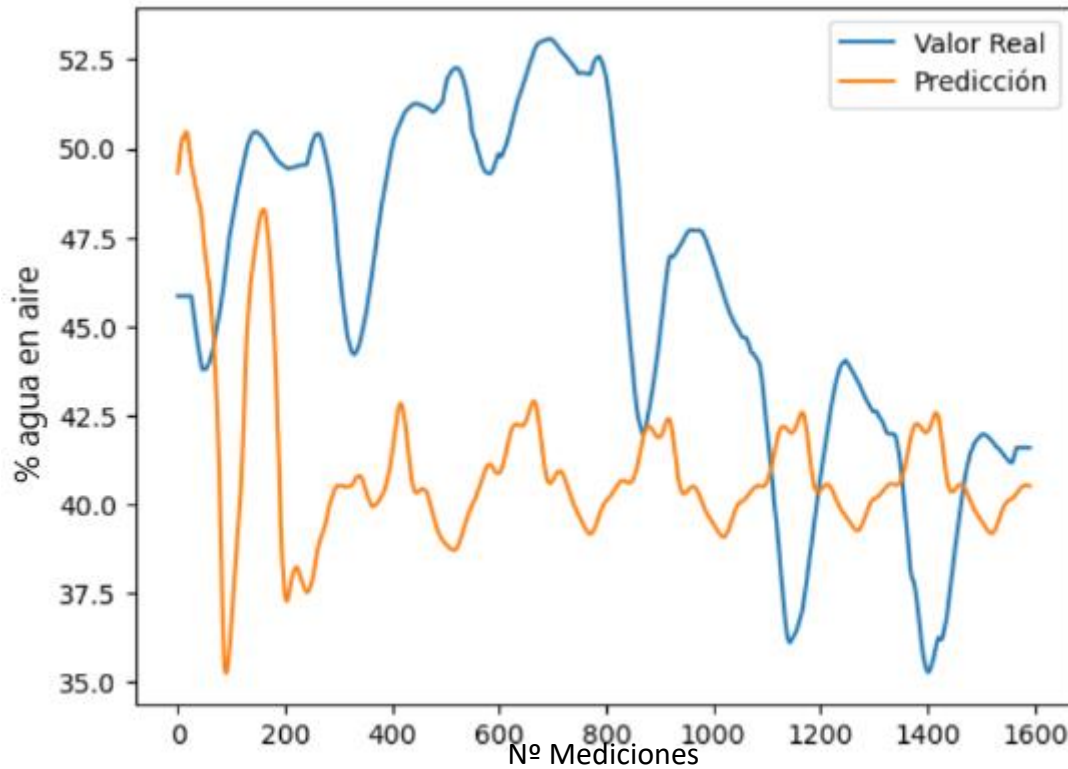


FIGURA 34: Comparación datos de validación predicciones futuras humedad

Resultados:

- MAE de la Transformada: 24.27
- MAE de la fase(ángulo): 0.15 radianes (8.6 grados)

Conclusión:

En el modelo de humedad nos pasa como el de temperatura si bien le cuesta dar los valores correctos no difiere mucho la fase predicha de la real.

PRESIÓN:

Para presión seguimos el mismo procedimiento que anteriormente, realizamos la Transformada de Fourier a las siguientes series.

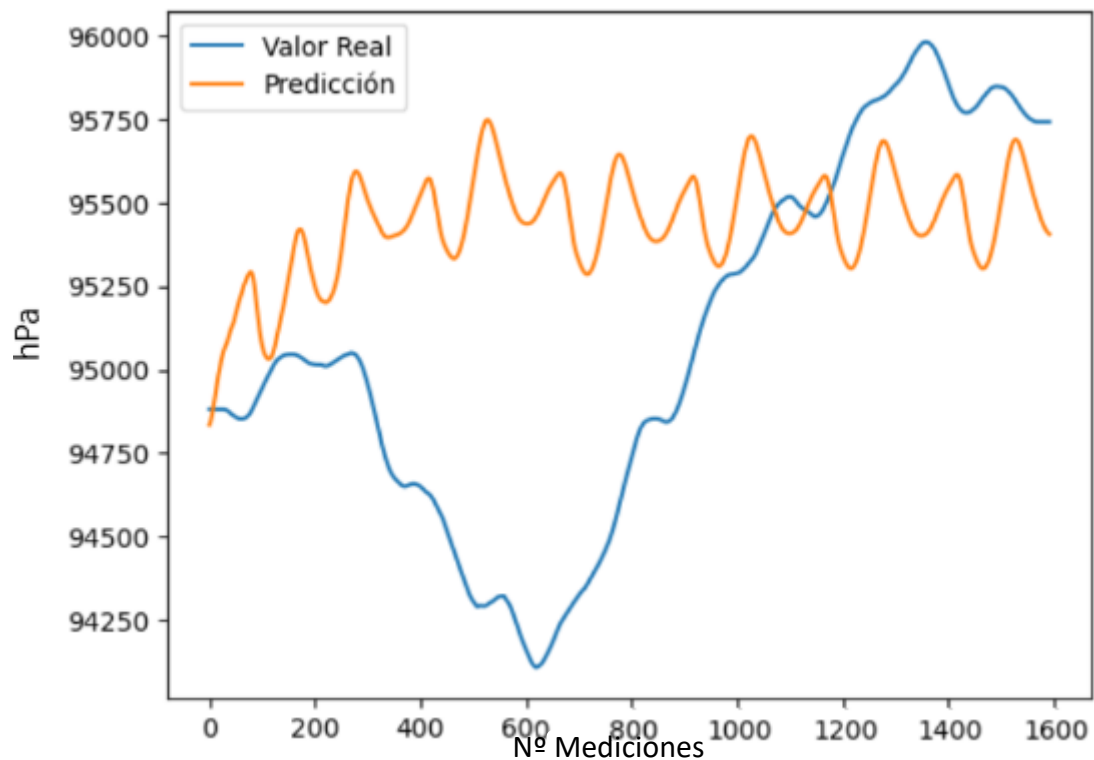


FIGURA 35: Comparación datos de validación predicciones futuras presión

Resultados:

- MAE de la Transformada: 1575.06
- MAE de la fase(ángulo): 0.05 radianes (2.8 grados)

Conclusión:

En cuanto a la presión vemos que el MAE de la transformada nos da un valor muy alto pero hay que tener en cuenta que la presión oscila en un rango entre 92000 y 96000 unidades. También obtenemos un MAE de la fase muy bajo por lo que nuestro modelo capta bien la variación de ángulo.

## 6.5 Sincronización en tiempo real

En la validación de la sincronización en tiempo real del Gemelo Digital, se confirma que la actualización del Gemelo ocurra sin demoras. Este proceso se logra mediante consultas que extraen datos en tiempo real desde la base de datos InfluxDB, alimentando así de manera instantánea la representación digital del objeto físico. Además, durante la validación, se asegura la posibilidad de conectarse en tiempo real al sensor físico,

permitiendo la visualización inmediata de los datos en el Gemelo Digital, siempre y cuando el sensor real esté conectado al mismo WiFi que el dispositivo que sostiene al Gemelo Digital. Este enfoque integral fortalece la fidelidad y la interactividad entre la representación digital y el objeto físico correspondiente.

Por otro lado, la actualización de los modelos de predicción no está disponible en tiempo real, a día de hoy cada diez días aproximadamente se reentrenan con nuevos datos obtenidos hasta esa fecha.

En cuanto a la actualización de nuestros modelos de predicción, estos no se reajustan en tiempo real. En la práctica, llevamos a cabo un proceso de reentrenamiento cada diez días aproximadamente, utilizando los nuevos datos acumulados hasta ese momento. Este enfoque periódico asegura que nuestros modelos se mantengan actualizados con la información más reciente, permitiéndonos mejorar continuamente su precisión y eficacia. Al incorporar regularmente nuevos conjuntos de datos, podemos ajustar y optimizar los algoritmos de predicción para reflejar mejor las tendencias actuales y las variaciones en los datos observados. Este ciclo de reentrenamiento estructurado es fundamental para mantener la relevancia y la precisión de los modelos frente a la evolución de los patrones de datos a lo largo del tiempo.

## **6.6 Seguridad cibernética**

En lo que respecta a la seguridad de nuestro Gemelo Digital, aunque no se han implementado medidas de seguridad específicas a nivel de software, se ha prestado especial atención a la protección a nivel de datos. La transmisión de datos se realiza a través de MQTT, un protocolo que ofrece características de seguridad robustas, incluyendo la capacidad de utilizar el cifrado TLS/SSL para asegurar la comunicación entre el cliente y el servidor, autenticación de cliente mediante certificados y la opción de usar nombres de usuario y contraseñas para una capa adicional de seguridad. Estas características ayudan a garantizar que los datos transmitidos sean protegidos contra interceptaciones no autorizadas y manipulaciones.

Por otro lado, el almacenamiento de los datos se gestiona mediante InfluxDB, una base de datos diseñada para optimizar la seguridad de los datos almacenados. InfluxDB admite la autenticación y autorización de usuarios, limitando el acceso a los datos a usuarios verificados y permitiendo la definición de permisos específicos para controlar las

operaciones que los usuarios pueden realizar. Además, InfluxDB puede configurarse para usar el cifrado HTTPS para la transmisión de datos, así como el cifrado en reposo, asegurando que los datos almacenados estén protegidos tanto durante su transmisión como cuando están almacenados.

En conjunto, estas medidas de seguridad incorporadas en las herramientas que utilizamos para la transmisión y almacenamiento de datos aseguran una capa robusta de protección para nuestro Gemelo Digital, centrando nuestros esfuerzos de seguridad en la integridad y confidencialidad de los datos manejados.

## 7. Conclusiones

Después de completar el proyecto, procedemos a analizar los resultados obtenidos. Logramos implementar un gemelo digital que puede predecir con precisión mediciones de temperatura, presión y humedad. Sin embargo, debido a problemas técnicos con el sensor de luminosidad, no obtuvimos un gemelo tan preciso para esta medición, aunque sí lo suficientemente preciso para analizar las tendencias.

Gracias al sistema creado con Docker, podemos obtener datos en tiempo real y almacenarlos en una base de datos a la que accede nuestro gemelo. Una vez obtenidos los datos, nuestro gemelo procede a limpiarlos y prepararlos para su uso. Para cada medición, se verifica si hay un modelo entrenado; de lo contrario, se procede a entrenarlo, y este nuevo modelo entrenado se guarda para predicciones y usos futuros.

Desde nuestra interfaz gráfica podemos realizar diversas acciones para las cuatro mediciones. Podemos realizar predicciones para el futuro, ingresando el día y la hora deseados. También podemos conectarnos con el sensor real en tiempo real, siempre que el sensor esté en la misma red WiFi que el dispositivo en el que se ejecute el gemelo digital.

Una de las opciones más interesantes y visuales es la capacidad de generar gráficos. Podemos elegir entre dos tipos de gráficos: uno que muestra la tendencia de los datos y otro que muestra las predicciones reales frente a las predichas.

Por último, nuestro gemelo es capaz de hacer publicaciones MQTT, similar al sensor real. Podemos seleccionar la frecuencia con la que queremos realizar las predicciones y el tiempo total de estas predicciones. El gemelo realizará las publicaciones en el puerto y el tema especificados.

Cuando validamos nuestros modelos de predicción, dividimos los datos en tres conjuntos, como se explicó en la sección de validación. Observamos que nuestros modelos tienen un alto rendimiento al predecir los datos de prueba, ya que se le van pasando al modelo las secuencias reales de ese momento. Sin embargo, conforme realizamos predicciones para el futuro y estas predicciones se integran en las secuencias de datos utilizadas para

avanzar en el tiempo, la precisión de nuestros modelos al dar un valor exacto disminuye. A pesar de ello, siguen siendo capaces de captar la tendencia general de las mediciones.

## **8. Consideraciones sociales, económicas y medio ambientales.**

El desarrollo del gemelo digital representa un avance significativo hacia la sostenibilidad y eficiencia en el manejo de recursos naturales y energéticos. Socialmente, esta tecnología promete mejorar la calidad de vida, al permitir una gestión más inteligente y adaptativa de los entornos urbanos y naturales, reduciendo así el impacto ambiental de actividades humanas.

Económicamente, la implementación de gemelos digitales puede ofrecer ahorros significativos a largo plazo, optimizando el consumo de recursos y minimizando costos operativos mediante la predicción precisa y la prevención de situaciones adversas. Además, puede fomentar la creación de nuevos mercados y empleos vinculados a la tecnología verde y la economía circular.

Desde una perspectiva medioambiental, el proyecto del gemelo digital no solo contribuye a la conservación de recursos naturales y minimiza las emisiones contaminantes, sino que también reduce la necesidad de fabricar sensores físicos. Al simular digitalmente el comportamiento y las condiciones que los sensores físicos monitorearían, se disminuye la demanda de materiales y energía requeridos para su producción, lo cual representa un ahorro significativo de recursos y una reducción en la generación de residuos. Este enfoque sustentable refuerza el compromiso con prácticas de desarrollo sostenible, alineándose con esfuerzos globales para mitigar el impacto ambiental de la tecnología.

En resumen, el proyecto del gemelo digital se alinea con los objetivos de desarrollo sostenible, ofreciendo un enfoque holístico que abarca consideraciones sociales, económicas y medioambientales, esenciales para el progreso y bienestar de las sociedades contemporáneas.

## 9. Trabajos futuros

Para mejorar nuestro gemelo digital en el futuro, planeamos centrarnos en varios aspectos clave. El más importante es perfeccionar las mediciones de luminosidad, estableciendo un flujo de datos continuo que alimente nuestro modelo específico para este parámetro. Esto permitirá obtener predicciones más precisas y relevantes.

Además, pretendemos enriquecer la interfaz de usuario con la incorporación de gráficas adicionales y mejoras generales. Esto hará que los datos y predicciones sean más fáciles de interpretar, facilitando así la toma de decisiones en tiempo real.

Un cambio significativo que buscamos implementar es la automatización del reentrenamiento de nuestros modelos. Actualmente, este proceso se realiza manualmente, pero nuestro objetivo es que se ejecute automáticamente a intervalos regulares. De esta forma, el modelo se actualizará constantemente con los últimos datos recopilados por los sensores, mejorando su precisión y relevancia.

Para proyectos más avanzados, aspiramos a evolucionar de un gemelo digital basado en un único sensor a una red interconectada de sensores. Esto permitirá que los dispositivos compartan datos entre sí, mejorando significativamente la precisión de nuestras predicciones al tener en cuenta múltiples fuentes de información. Este enfoque de red de sensores marca un paso adelante hacia predicciones más precisas y decisiones más informadas.



## 10. Referencias

- [1] Miskinis, C. "The history and creation of the digital twin concept." Challenge.org. Disponible en: <https://www.challenge.org/insights/digital-twin-history/> (Fecha de consulta: 15 de octubre de 2023)
- [2] Scholarly Community Encyclopedia. "Evolution of Digital Twins." Encyclopedia.pub. Disponible en: <https://encyclopedia.pub/entry/9513> (Fecha de consulta: 7 de noviembre de 2024)
- [3] Open Data Science. "Evolution of Digital Twins." Medium.com. Disponible en: <https://odsc.medium.com/evolution-of-digital-twins-5df92006878b> (Fecha de consulta: 20 de noviembre de 2024)
- [4] Repsol. "¿Qué es un gemelo digital?" Repsol.com. Disponible en: <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/gemelos-digitales/index.cshmtl> (Fecha de consulta: 10 de noviembre de 2023)
- [5] Siemens. "Gemelo digital de productos." Siemens.com. Disponible en: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/digital-twin/24465> (Fecha de consulta: 5 de diciembre de 2024)
- [6] IBM. "¿Qué es un gemelo digital?" Ibm.com. Disponible en: <https://www.ibm.com/es-es/topics/what-is-a-digital-twin> (Fecha de consulta: 18 de diciembre de 2023)
- [7] Open Sistemas. "Gemelo digital: ejemplos reales de uso, conoce todo su potencial." Opensistemas.com. Disponible en: <https://opensistemas.com/gemelo-digital-ejemplos/> (Fecha de consulta: 9 de noviembre de 2024)
- [8] AWS Amazon. "¿Qué es la tecnología de gemelos digitales?" Amazon Web Services. Disponible en: <https://aws.amazon.com/es/what-is/digital-twin/> (Fecha de consulta: 28 de noviembre de 2024)
- [9] Attaran, M. "Digital Twin: Benefits, use cases, challenges, and opportunities." ScienceDirect. Disponible en: <https://www.sciencedirect.com/science/article/pii/S277266222300005X#sec3> (Fecha de consulta: 17 de octubre de 2024)
- [10] LinkedIn. "What are the best practices for testing and validating your digital twin models before deploying them?" Disponible en:

<https://www.linkedin.com/advice/0/what-best-practices-testing-validating-your-2e>

(Fecha de consulta: 12 de noviembre de 2023)

[11] CSO. "Desafíos y oportunidades de ciberseguridad de los gemelos digitales." Computerworld.es. Disponible en: <https://cso.computerworld.es/tendencias/desafios-y-oportunidades-de-ciberseguridad-de-los-gemelos-digitales> (Fecha de consulta: 21 de diciembre de 2023)

[12] AWS IoT. "AWS IoT TwinMaker." Amazon Web Services. Disponible en: <https://aws.amazon.com/es/iot-twinmaker/> (Fecha de consulta: 4 de febrero de 2024)

[13] Conde Diaz, J. "Contribución al Desarrollo de arquitecturas y soluciones para la implementación de gemelos digitales basados en datos abiertos." Universidad Politécnica de Madrid. Disponible en: [https://oa.upm.es/74176/1/JAVIER\\_CONDE\\_DIAZ.pdf](https://oa.upm.es/74176/1/JAVIER_CONDE_DIAZ.pdf) (Fecha de consulta: 8 de enero de 2024)

[14] Fernández Álvarez, G. "Repsol deploys its production management digital twin." LinkedIn.com. Disponible en: <https://www.linkedin.com/pulse/repsol-deploys-its-production-management-digital-twin-guillermo/> (Fecha de consulta: 16 de noviembre de 2024)

[15] Becerra Pozas, J. L. "Gemelos digitales: 5 casos de éxito." Cio.com.mx. Disponible en: <https://cio.com.mx/gemelos-digitales-5-casos-de-exito/> (Fecha de consulta: 3 de diciembre de 2024)

[16] "Gemelos digitales, una tecnología que no para de crecer." Forbes Argentina. Disponible en: <https://www.forbesargentina.com/innovacion/gemelos-digitales-una-tecnologia-crecer-n20830> (Fecha de consulta: 26 de diciembre de 2023)

[17] Baron Rufete, J. M. "Metodología para el Desarrollo de un gemelo digital." Universidad Politécnica de Madrid. Disponible en: [https://oa.upm.es/72603/1/TFM\\_JOSE MARIA BARON RUFETE.pdf](https://oa.upm.es/72603/1/TFM_JOSE MARIA BARON RUFETE.pdf) (Fecha de consulta: 23 de noviembre de 2023)

[18] Sothis. "Gemelo Digital: El proceso de digitalización al detalle." Disponible en: <https://www.sothis.tech/gemelo-digital-proceso/> (Fecha de consulta: 14 de enero de 2024)

[19] NumPy Developers. "Routines: Discrete Fourier Transform (numpy.fft)." NumPy v1.22 Manual. Disponible en: <https://numpy.org/doc/stable/reference/routines.fft.html> (Fecha de consulta: 5 de diciembre de 2023)

- [20] Gómez Moreno, A. "Desarrollo de un gemelo digital para la gestión de almacenes." Universidad de Sevilla, 2021. Disponible en: <https://idus.us.es/bitstream/handle/11441/127622/TFG-3799-GOMEZ%20MORENO.pdf?sequence=1&isAllowed=y> (Fecha de consulta: 7 de enero de 2024)
- [21] Norlean. "Fases para crear un gemelo digital de éxito." Disponible en: <https://norlean.com/es/fases-para-crear/> (Fecha de consulta: 19 de enero de 2024)
- [22] scikit-learn Developers. "Decision Trees." scikit-learn 0.24.1 documentation. Disponible en: <https://scikit-learn.org/stable/modules/tree.html#regression> (Fecha de consulta: 9 de enero de 2023)
- [23] "Gemelos Digitales: qué son, ventajas y aplicaciones." Iberdrola. Disponible en: <https://www.iberdrola.com/innovacion/gemelos-digitales> (Fecha de consulta: 28 de noviembre de 2023)
- [24] PyTorch. "PyTorch Documentation." Disponible en: <https://pytorch.org/docs/stable/index.html> (Fecha de consulta: 17 de diciembre de 2023)
- [25] "¿Qué son los Digital Twins o Gemelos Digitales?" Computer Hoy. Disponible en: <https://computerhoy.com/reportajes/tecnologia/son-digital-twins-gemelos-digitales-1053559> (Fecha de consulta: 18 de octubre de 2024)
- [26] TensorFlow. "Keras overview." TensorFlow 2.4.0 documentation. Disponible en: <https://www.tensorflow.org/guide/keras?hl=es-419> (Fecha de consulta: 11 de noviembre de 2024)
- [27] "Qué son los gemelos digitales y por qué pueden ser el futuro de la medicina." Infobae. Disponible en: <https://www.infobae.com/america/ciencia-america/2023/04/07/que-son-los-gemelos-digitales-y-por-que-pueden-ser-el-futuro-de-la-medicina/#:~:text=B%C3%A1sicamente%20es%20un%20gemelo%20de,a%20trav%C3%A9s%20de%20la%20pantalla.&text=O%20el%20proyecto%20que%20encar%C3%B3,su%20coraz%C3%B3n%20en%20realidad%20virtual>. Fecha de consulta: 25 de noviembre de 2023)
- [28] Maulshree. "Digital Twin: Origin to Future." MDPI. Disponible en: <https://www.mdpi.com/2571-5577/4/2/36> (Fecha de consulta: 13 de enero de 2024)
- [29] "Qué son los Gemelos digitales y cómo permitirán progresar a las industrias." La Tercera. Disponible en: <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/gemelos->

[digitales/index.cshtml#:~:text=Los%20gemelos%20digitales%20o%20%E2%80%9Cdigital%20twins%E2%80%9C%20son%20un%20modelo%20virtual,real%20de%20una%20manera%20eficiente.](#) (Fecha de consulta: 14 de noviembre de 2024)

[30] "Gemelos digitales: ¿Qué podrían hacer por su negocio?" McKinsey. Disponible en: <https://www.mckinsey.com/featured-insights/destacados/gemelos-digitales-que-podrian-hacer-por-su-negocio/es> (Fecha de consulta: 20 de diciembre de 2024)

[31] "La era de los gemelos digitales." La Vanguardia. Disponible en: <https://www.lavanguardia.com/vida/20210428/7374253/la-era-de-los-gemelos-digitales.html> (Fecha de consulta: 2 de enero de 2024)

[32] scikit-learn Developers. "Neural network models (supervised)." scikit-learn 0.24.1 documentation. Disponible en: [https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html#regression](https://scikit-learn.org/stable/modules/neural_networks_supervised.html#regression) (Fecha de consulta: 11 de diciembre de 2023)

[33] "Digital twins: qué son, para qué sirven y cuáles son los beneficios y problemas de los gemelos digitales." Xataka. Disponible en: <https://www.xataka.com/pro/digital-twins-que-sirven-cuales-beneficios-problemas-gemelos-digitales#:~:text=Un%20gemelo%20digital%20se%20crea,de%20an%C3%A1lisis%2C%20monitoreo%20y%20predicci%C3%B3n> (Fecha de consulta: 3 de febrero de 2024)

[34] "¿Qué son los gemelos digitales y cómo funcionan?" Unity. Disponible en: <https://unity.com/solutions/digital-twins> (Fecha de consulta: 26 de enero de 2024)

[35] PyTorch. "torch.nn — Neural network layers and functions." PyTorch Documentation. Disponible en: <https://pytorch.org/docs/stable/nn.html> (Fecha de consulta: 16 de noviembre de 2024)

[36] "Gemelos Digitales: su uso para medir la calidad del aire." Envira IoT. Disponible en: <https://enviraiot.es/gemelos-digitales-uso-medir-calidad-aire/> (Fecha de consulta: 22 de noviembre de 2024)

[37] "Gemelos digitales al servicio de la eficiencia energética." Semantic Systems. Disponible en: <https://www.semantic-systems.com/semantic-noticias/articulos-tecnologicos/gemelos-digitales-al-servicio-de-la-eficiencia-energetica/> (Fecha de consulta: 5 de diciembre de 2023)