



**Universitat de les
Illes Balears**

Escuela Politécnica Superior

Memoria del Trabajo de Fin de Grado

Digital Twins en entornos domésticos

Jaume Ferrer Mayol

Grado en Ingeniería Informática

Año académico 2020-2021

DNI del alumno: 43228873M

Trabajo tutelado por Dr. Bartolomé J. Serra Cifre
Departamento de Matemáticas e Informática

Se autoriza a la Universidad a incluir este trabajo en el Repositorio Institucional para su consulta en acceso abierto y su difusión por línea, con finalidades exclusivamente académicas y de investigación.	Autor		Tutor	
	Sí	No	Sí	No
	X		X	

Palabras clave del trabajo:

Gemelo digital, tiempo real, modelo, bomba de calor, mantenimiento predictivo, simulación

Agradecimientos

A mi tutor, Tomeu, por ayudarme pacientemente a planificar el trabajo,

A Toni, por presentarme el entorno real de investigación,

A mi familia, por ser el apoyo incondicional que todo universitario necesita,

A mis compañeros, por los increíbles momentos vividos durante esta etapa,

A ti, por dejar a un lado tus muchas otras tareas y leer este trabajo.

ÍNDICE

1. Introducción.....	6
2. Contextualización	9
2.1. Historia	9
2.2. Concepto.....	11
2.3. Valor	12
2.4. Retos	13
3. Estado del arte	14
3.1. Tecnologías relacionadas con los datos.....	14
3.1.1. Recolección de los datos.....	15
3.1.2. Transmisión de los datos	18
3.1.3. Procesamiento de los datos.....	26
3.1.4. Almacenamiento de los datos	28
3.2. Tecnologías para modelos de alta fidelidad	32
3.2.1. Modelos físicos.....	32
3.2.2. Modelos semánticos	33
3.3. Tecnologías para la simulación basada en modelos de alta fidelidad	35
3.3.1. Plataformas de simulación en tiempo real.....	35
3.3.2. Interacción bidireccional	36
4. Bomba de calor.....	38
5. Entorno de trabajo	41
5.1. El problema	42
5.2. Limitaciones del entorno	43
6. Planteamiento y análisis de requisitos	44
6.1. ¿Por qué digital twins?	44
6.2. Análisis de requisitos.....	45
7. Metodología.....	47
7.1. Gestión del proyecto.....	47
7.2. Control de versiones	47
8. Diseño.....	49
8.1. Gemelo físico.....	50
8.2. Gemelo digital	50
8.2.1. Servidor	51
8.2.2. Cliente.....	52
8.2.3. Componentes externos.....	53

9. Implementación	56
9.1. Gemelo físico.....	58
9.1.1. Recolección de los datos.....	59
9.1.2. Sistema de control del entorno físico	60
9.2. Gemelo digital	61
9.2.1. Módulo de comportamiento del sistema - Servidor flask.....	61
9.2.2. Desarrollo del cliente.....	75
10. Resultados y trabajos futuros.....	77
10.1. Resultados de la interacción bidireccional entre gemelos	77
10.2. Visualización en tiempo real del comportamiento del gemelo físico.....	79
10.3. Resultados de la simulación en el comportamiento futuro del sistema.....	82
10.4. Trabajo futuros	86
11. Conclusiones.....	87
12. Bibliografía.....	88

1. Introducción

Este trabajo está centrado en el mundo de los Digital Twins (DT), un paradigma que se ha estado extendiendo en los últimos años y que involucra la labor de numerosas áreas de la informática para su aplicación. Los gemelos digitales son réplicas virtuales autónomas de un producto, servicio o sistema real que se basan en información proporcionada por su homólogo físico para poder reflejar el estado actual del sistema y adaptarse de manera predictiva a los diferentes episodios de su ciclo de vida.

Cada año, se estima un gasto de veinte mil millones de dólares dedicado a mantener equipamiento industrial. Los gemelos digitales permiten digitalizar la manipulación de la información que se conoce de estos sistemas y generar nueva información predictiva para garantizar que estos activos funcionen de manera más predecible y eficiente, lo que supone un ahorro económico mayúsculo [1].

Según Gartner, en el 2019 el 13% de las empresas que implementaban entornos de IoT ya hacían uso de gemelos digitales, otro 62% planeaba hacerlo en cuestión de un año [2]. Para este 2021, las predicciones apuntan a un crecimiento de la eficiencia en un 10 por ciento para la mitad de compañías industriales mediante el uso de gemelos digitales [3].

Un gran número de empresas, principalmente de fabricación industrial han apostado por el desarrollo de gemelos digitales para evitar la necesidad de prototipos físicos, lo que permite reducir el tiempo de desarrollo y mejorar la calidad del producto terminado. Mediante softwares de generación de gemelos digitales como CAD NX [4], marcas de prestigio como Masseratti pudieron reducir tanto el coste como el tiempo de desarrollo en un 30 por ciento. Para estos automóviles de lujo, se redujo el tiempo de comercialización en 16 meses [5].

A la hora de implementar gemelos digitales, se pueden encontrar desde sistemas tan minúsculos como un sensor hasta representaciones complejas como una ciudad. Además, se espera de ellos poder establecer una sociedad de gemelos digitales, donde todo se encuentran en constante intercambio de información para así operar de manera proactiva con el conocimiento sobre su entorno [6].

Aunque las posibilidades para implementar gemelos digitales son muchas, este trabajo se centra el caso de uso concreto de los entornos domésticos. Trabajar con los gemelos digitales en entornos domésticos permite comprender en tiempo real cómo funciona un edificio. Como resultado, se obtendrá un edificio inteligente más rentable, sencillo y sostenible [7].

El entorno sobre el que se desarrolla este proyecto se sitúa en el Centro de Innovación Ca Ses Llúcies, ubicado en el mismo campus universitario, que tiene el fin de convertirse en un ejemplo de espacio sostenible, inteligente y saludable. Uno de los criterios de este proyecto es la calidad ambiental y eficiencia en el uso de recursos, por ejemplo, el sistema de climatización, sistemas de ventilación y calidad de aire, suministro energético, entre otros [8]. Concretamente, se ha planteado el uso de *digital twins* sobre el sistema de bomba de calor con el fin de mejorar su eficiencia y mantenimiento así como el confort del propio usuario.

El objetivo de este estudio es diseñar un *digital twin* para una bomba de calor aire-agua que permita monitorizar el estado del gemelo físico y tomar decisiones a partir de los datos en crudo que este mismo y fuentes exteriores proporcionan. Este proceso deberá tener implicaciones directas en cuestiones como la eficiencia energética, la transparencia sobre su funcionamiento y el confort del usuario.



Fig. 1-A. De izquierda a derecha: unidades interior y exterior del sistema de bomba de calor Panasonic WH-SDC05H3E.

Este documento ha sido estructurado de la siguiente manera:

En el primer capítulo se explora sobre el contexto y el motivo por el que surge la idea de los gemelos digitales. Se presentan sus antecesores, representaciones idénticas moldeadas físicamente para reflejar el comportamiento del gemelo en producción y principalmente desarrollados en el sector aeroespacial.

En el segundo capítulo se exponen algunas definiciones históricas que se le otorgan desde diferentes puntos de vista y para diferentes campos de la ciencia. Se definen los retos a los que la comunidad de desarrollo se enfrenta, así como el valores añadido que aporta confiar en estos nuevos entes virtuales e inteligentes, respectivamente.

En el tercer capítulo se determina, dando un repaso al estado del arte, el camino y las herramientas actuales que permiten transformar estas grandes cantidades de datos salientes de la entidad real en información útil para el usuario y en un funcionamiento más eficiente y fiable de la máquina. Posteriormente, en el Capítulo 4 se definen los componentes, características y propiedades de una bomba de calor genérica en un sistema de climatización para el hogar.

Tras explicar el estado del arte en los *digital twins* así como el funcionamiento genérico del sistema con el que se trabajará en el proyecto, en el capítulo 5 se conoce el entorno real que será caso de estudio.

Una vez presentado el caso de estudio y recogidos los aspectos de la instalación de la bomba cuya funcionalidad se verá optimizada mediante el uso de gemelos digitales, se describe en el capítulo 7 la metodología utilizada para realizar el desarrollo de la solución.

En el capítulo 8 se realiza un diseño de una aplicación que permita cumplir con las expectativas que se espera del gemelo digital enumeradas en el previo capítulo 6 y que se explicará su implementación sobre el entorno energético del proyecto en el capítulo 9.

En el capítulo 10, se muestra la batería de pruebas utilizada para validar el cumplimiento de cada uno de los requisitos que se han preestablecido.

Finalmente, en el capítulo 11 se presentan las conclusiones extraídas a partir de los resultados obtenidos.

2. Contextualización

En los siguientes apartados se realiza un análisis del contexto en el que aparecen los *digital twin*. Primeramente, se resumirán los principales hitos que han marcado la historia de esta tecnología. Después, se redactan los aspectos fundamentales que los definen y el valor que aporta al usuario apostar por estas soluciones. Por último se enumerarán los retos principales a la hora de conseguir esta semejanza en el comportamiento del gemelo digital y su homólogo físico.

2.1. Historia

En este capítulo se describen algunos de los aspectos más característicos de los antecedentes de los gemelos digitales: ¿en qué se basan? ¿de dónde proviene este concepto? ¿cuáles son sus expectativas para el mundo actual y futuro? Las próximas líneas resumen los diferentes hitos en la historia que han contribuido a plantear la utilidad que tiene implementar clones de la maquinaria real en producción y las ventajas que suponen transformar estos clones en un primer momento físicos a una naturaleza digital.

Existen fechas en la historia que han significado un antes y un después para lo que se puede entender como tecnología, pero no es hasta la segunda mitad del siglo XVIII que se viraliza este término. La llegada de la denominada revolución industrial permitió a zapateros, herreros, constructores, agricultores y un largo lista de otras profesiones, cambiar sus manos por máquinas las cuales, irónicamente, hacían el mismo trabajo pero más rápido, más preciso y sin sufrir fatiga. Desde ese momento, la tecnología en los sistemas de producción ha estado cada vez más presente, pero sin duda, este protagonismo no se adquirió de la noche a la mañana.

Según algunos historiadores, dentro de la historia de la humanidad aparecen tres grandes revoluciones: la cognitiva, la agrícola y la industrial [9]. ¿Y por qué no encontrar los hitos que permitan seccionar esta última etapa, la llamada revolución industrial y tecnológica?

Se conoce que la mecanización surgió hace más de cinco mil años, en la civilización Sumeria, con la invención de la rueda, sin embargo, la etapa más famosa no llegó hasta el 1440 d.C. con la imprenta. Considerando la revolución industrial como el auge de los sistemas de producción mecanizados, se considera que realmente el punto de inflexión fue allá sobre la segunda mitad del S.XIX, en Gran Bretaña. Se comienza a utilizar una fuente de suministro energético basado en agua y en vapor para reemplazar de manera parcial o total el trabajo humano o animal, la máquina de vapor. Sin embargo, estas máquinas requieren de instrucciones y de información que todavía debe ser proporcionada por los humanos, a diferencia de la maquinaria automatizada, que tardará todavía unos siglos en aparecer [10].

En segundo lugar, a principios del siglo XX, concretamente el 1901, se inauguró la primera fábrica de montaje en cadena, también conocido como “producción en masa”,

lo que significó un aumento mayúsculo de la productividad de las fábricas junto a un gran decremento de los costes de producción.

Tras nacimiento de la computación tras las 2ª Guerra Mundial, aparecen los primeros programas que permiten que las máquinas obedezcan a una serie de instrucciones dadas de manera automática, se habla de la automatización de los sistemas de producción . Esto no dotó a las máquinas de pensamiento propio, pero sí transfirió los mandos de su funcionamiento a nuevas entidades que esta vez fueros los ordenadores, en lugar de los humanos, a pesar de que no tuvieran la capacidad de, por sí solas, adaptarse a cambios de entorno o automejorarse.

La cuarta revolución industrial, también llamada Industria 4.0, permite que estas ya “piensen” por sí solas. Con el auge de tecnologías como IoT, el Big Data y Machine Learning, las máquinas son capaces de tomar decisiones y prevenir errores y averías de manera autónoma a partir de un análisis masivo de datos en tiempo real que recogen de su entorno. Los sistemas ciberfísicos (CPS) son un concepto clave de esta nueva revolución, unos dispositivos que abarcan dos realidades, objetos que tienen la capacidad de interactuar con allí donde se encuentre otro ente virtual o físico a través de las red. Por tanto, estas entidades físicas tendrán su respectiva representación digital, su “Digital Twin”, que permitirá monitorizar, controlar, reflejar, predecir y optimizar su estado y funcionamiento de manera sumamente precisa.

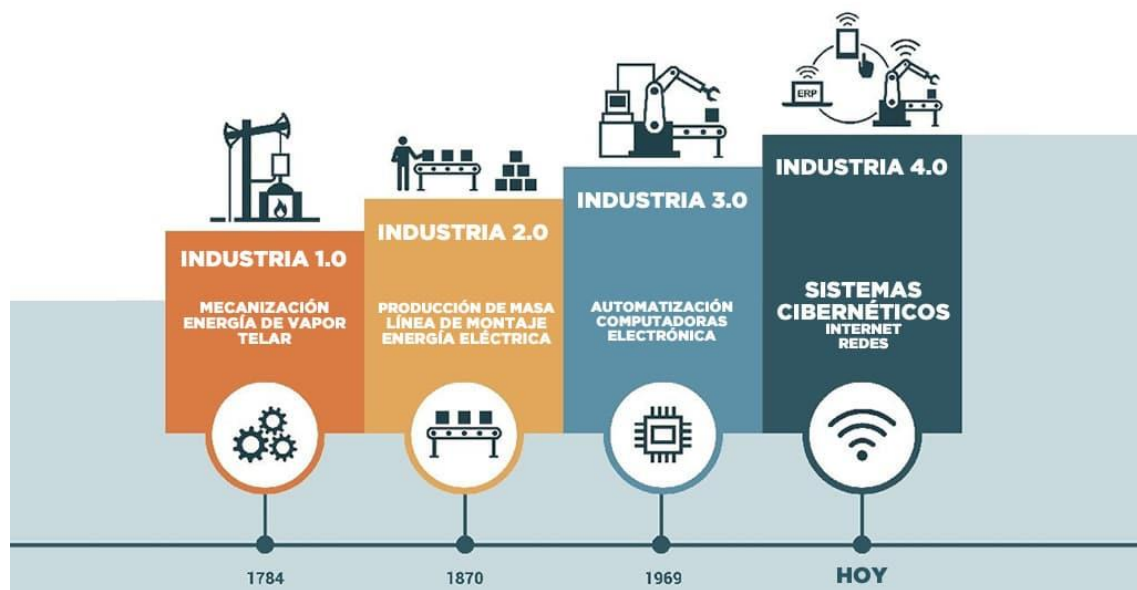


Fig. 2.1-A. Etapas de la mecanización [11].

La fabricación digital, una de las bases de los CPS, una idea orientada construir modelos numéricos y simulaciones de productos y procesos de desarrollo a través de la representación virtual de entidades como fábricas, recursos y mano de obra, ha estado aportando valor continuamente al sector de la industria durante las últimas décadas.

Una de las iniciativas que significó el comienzo del auge de este concepto fue el programa espacial Apollo, de la NASA. Se construyeron, como mínimo, dos vehículos espaciales idénticos que permitieron que se reflejara sobre los situados en la tierra el

comportamiento del del enviado al espacio exterior. El que permaneció en la tierra fué bautizado como “gemelo”. Este se utilizó durante el entrenamiento de la misión y, durante su desarrollo, permitió simular condiciones y ayudar a los astronautas a resolver situaciones críticas [12].

Sin embargo, la NASA descubrió que, a pesar de contar con una réplica exacta del modelo original en la Tierra, las decisiones correctivas no deberían basarse en el modelo original puesto que, a ciertas alturas de la misión, este se expondría a cambios significativos dada la hostilidad del espacio vacío. Mientras que estos primeros gemelos consistían en combinaciones de modelos físicos y matemáticos, el avance tecnológico hasta hoy en día ha permitido convertirlos en representaciones puramente virtuales, dinámicas y de una precisión asombrosa [13].

2.2. Concepto

Ya se ha comentado cómo el gemelo digital nace a partir de la base del concepto de “gemelo”. En un primer lugar estos gemelos fueron físicos, famosos por intervenciones como en el vuelo del Apolo 11 o Apolo 13, sobre los que los ingenieros se respaldaron para poder clonar el sistema que se encontraba en la misión.

Aunque ya se encargó David Gelernter en *Mirror Worlds* (1991) de postular por primera vez la simulación digital del mundo real como realidad digital [14], no fue hasta 2003 cuando Michael Grieves puso en práctica este concepto en el ciclo de vida de procesos de fabricación [15].

Según Wang et al. (2019), un *digital twin* (DT) pueden considerarse como un paradigma mediante el cual se integran, dinámicamente y a través de la red, una serie de mediciones seleccionadas en un aparato físico con el mundo simulado, donde el modelo de simulación sirve también como guía para un mundo real adaptable a sus recomendaciones [16].

Trabajar sobre modelo virtuales ofrece la posibilidad de no dedicar los recursos a la intervención directa sobre el sistema físico. Por ende, se conoce cómo responderán los componentes que lo forman a diferentes situaciones reales a través de simulaciones digitales relializadas previamente. También, se da feedback instantáneo de estas predicciones al modelo físico, lo que permitirá un control inteligente en tiempo real del sistema; existe una comunicación bidireccional entre ente físico y ente digital.

Actualmente, se diseñan gemelos digitales para realizar tareas específicas. En cuanto a la maquinaria de producción industrial, su utilidad apunta a monitorizar condiciones operativas, prevenir errores o realizar mantenimientos predictivos. Si se habla de integrar el uso de DT y las personas, cabe referirse a temas como servicios relacionados con la salud, por ejemplo, la prevención de enfermades utilizando datos personales [6].

Mirando al futuro, los avances sobre los DT llevarán a la demanda de simulaciones a gran escala, en los que se deberán integrar y combinar diferentes entidades que hasta el momento servían para realizar operaciones concretas. Por lo tanto, una de las dificultades de los DT, pero a la vez una de sus motivaciones, es la interacción entre dos

o varias representaciones digitales. Desde este punto de vista, uno de los mayores retos es establecer un sistema normalizado de interacción entre gemelos; actualmente las estructuras de datos que manejan son heterogéneas e incompatibles con esta idea [17]. Aunque realizar una representación digital de un sistema físico de manera precisa no sea una tarea muy ardua, recrear una sociedad digital en la que varios de ellos puedan intercambiar información y mapear estos nuevos conocimientos a la realidad no es algo tan sencillo.

2.3. Valor

¿En qué beneficia al usuario este nuevo paradigma respecto a las visiones convencionales? En esta sección, se listan las aportaciones de valor para el usuario que quiera desplegar estas soluciones en entornos industriales y/o domésticos que los DT son capaces de generar [18]:

- 1- El control y monitorización en tiempo real no solo permite que se monitorice en tiempo real el rendimiento del sistema sino que además da acceso a él de manera remota, con total disponibilidad y con feedback inmediato.
- 2- Dota de autonomía a los sistemas, lo que aporta un mayor nivel de eficiencia y seguridad. Se ceden tareas que podría comprometer la salud de las personas a las máquinas mientras estas son controladas de manera remota.
- 3- Un análisis inteligente y preciso de las enormes cantidades de datos que aportan los múltiples sensores que formarían parte el sistema físico puede resultar en conclusiones que determinen un momento de fallo próximo del equipo. Estos errores podrían detectarse con antelación y para habilitar una planificación y mantenimiento predictivos.
- 4- La evaluación de riesgos y escenarios ayuda a imaginar situaciones y estados del sistemas actualmente desconocidos, para así estudiar la manera de actuar en consecuencia.
- 5- El alto grado de accesibilidad de la información y una mayor autonomía del sistema hace que la dedicación del los equipos pueden centrarse más en mejorar su sinerga y colaboración interna y externa para así mejorar su productividad.
- 6- La disponibilidad de datos cuantitativos y análisis avanzados en tiempo real conducirán a poder tomar decisiones de manera más eficiente y sabia.
- 7- En la fabricación, aumentarán las demandas de productos y servicios personalizados debido al acceso a un amplio historial de requisitos, preferencias de los interesados y al seguimiento detallado las tendencias del mercado.

2.4. Retos

En esta última sección del módulo dedicado a la contextualización de los *digital twins*, se enumeran los principales retos que se encuentran a la hora de ofrecer las aportaciones de valor a las cuales se refiere el apartado 2.3.

En el mundo de los gemelos digitales, la lista de retos es muy amplia y las soluciones muy diversas. Según A. Rashed, la principal cualidad a lograr sobre los *digital twins*, es conseguir que sean indistinguibles respecto a sus gemelos físicos, para ello, la comunidad encargada de modelar los DT debe asegurarse de cumplir ciertas expectativas [18]:

- El éxito de los DT depende de su conexión bidireccional en tiempo real entre la entidad física y su gemelo digital para permitir realismo físico.
- Conforme que el sistema físico evolucione en el tiempo, también se requiere que los componentes que se hayan modelado por computador vayan evolucionando y se mantenga su correspondencia.
- Garantizar la seguridad de sistemas físicos que dispongan de un clon virtual no es una cuestión trivial; se requerirá una mayor transparencia y entendimiento de las decisiones tomadas basadas en DT.
- Por último, mencionar que el DT debe presentarse al usuario final de manera que parezca indistinguible de la entidad física y más sencillo e intuitivo de trabajar.

En la siguiente sección se dan a conocer algunas de las tecnologías actuales que pueden ser apropiadas para implementar una solución que pueda cumplir con estos retos.

3. Estado del arte

Así como se menciona al principio de este documento, el objetivo de este estudio es diseñar un *digital twin* para una bomba de calor aire-agua que permita monitorizar el estado del gemelo físico y tomar decisiones a partir de los datos en crudo que este mismo y fuentes exteriores proporcionan. Este proceso deberá tener implicaciones directas en cuestiones como la eficiencia energética y el confort del usuario.

Aunque hay diferentes maneras de definir el estado del arte, es interesante mencionar la caracterización que realiza Liu et. al.(2021) en diferentes perspectivas de las tecnologías clave para el desarrollo de *digital twins* [12]:

- Tecnologías relacionadas con los datos
- Tecnologías para modelos de alta fidelidad
- Tecnologías para la simulación basada en modelos

Partiendo de esta diferenciación, en estos próximos párrafos se introduce cada una de las partes para posteriormente ser exploradas en profundidad:

Tecnologías relacionadas con los datos. Para identificar cuál es la interacción entre los gemelos digitales y sus respectivas representaciones físicas, es necesario examinar que tipos de datos contienen y cuáles de ellos necesitan transmitirse entre ellos. Este apartado a conocer los datos que se obtienen al realizar una recolección masiva, cómo se procesan, de qué manera son transmitidos en tiempo real a la vez que el receptor sepa interpretarlos y dónde se almacenan para posteriormente ser estudiados.

Tecnologías para modelos de alta fidelidad. Los datos recolectados de un entorno no tienen significado per se. No serían más que números y símbolos triviales si mediante algún tipo de tecnología no se convirtieran en información útil para el usuario. En esta sección se enumeran las tecnologías que permiten obtener información visual del estado y rendimiento del gemelo físico a partir de los dataset recibidos.

Tecnologías para la simulación basada en modelos. Tomar decisiones para mejorar el funcionamiento es el siguiente paso de la representación de este que menciona el punto anterior. Aquí, se verá qué metodologías son las más usadas para realizar predicciones a partir de los datos de los modelos y sistemas externos para determinar de qué manera se podría reajustar la configuración de control del gemelo físico para que estas se cumplan.

3.1. Tecnologías relacionadas con los datos

Los datos son el elemento básico de los gemelos digitales. El gemelo físico debe ser la fuente original de todas estas mediciones y debe transmitirlos en tiempo real a su homólogo digital. Se utiliza un sistema virtualizado del gemelo físico para procesar toda esta información en un plano externo al entorno de trabajo real y analizar el comportamiento y la correctitud de este con ánimos de mejorar su eficiencia, rendimiento y transparencia.

En este apartado, se mencionan las tecnologías actuales utilizadas para virtualizar el estado de estos entornos. Partiendo de la premisa de que debe quedar plasmada la actividad de un gemelo real, los capítulos se dedicarán a explicar cómo varía la naturaleza de estos datos según fluye en el sistema binario.

3.1.1. Recolección de los datos

La captura de datos es el primer paso para la construcción de un gemelo digital. Hay que asegurarse de que las fuentes de alimentación de los modelos numéricos son fuentes fiables y aportan datos relevantes en tiempo real.

Los modelos digitales permiten representar y simular condiciones presentes y futuras de del clon físico. Por tanto, se debe contar con una infraestructura que permita trabajar y comprender a nivel físico el sistema que se representa. Las redes de sensores son la solución. Una WSN (*Wireless Sensor Network*) es una estructura que compuesta por sensores, unidades de procesamiento y elementos de comunicación que tienen el propósito de grabar, observar y reaccionar a diferentes eventos o fenómenos del entorno real. [19].

Esta red de sensores interconectados, normalmente, sigue el siguiente esquema:

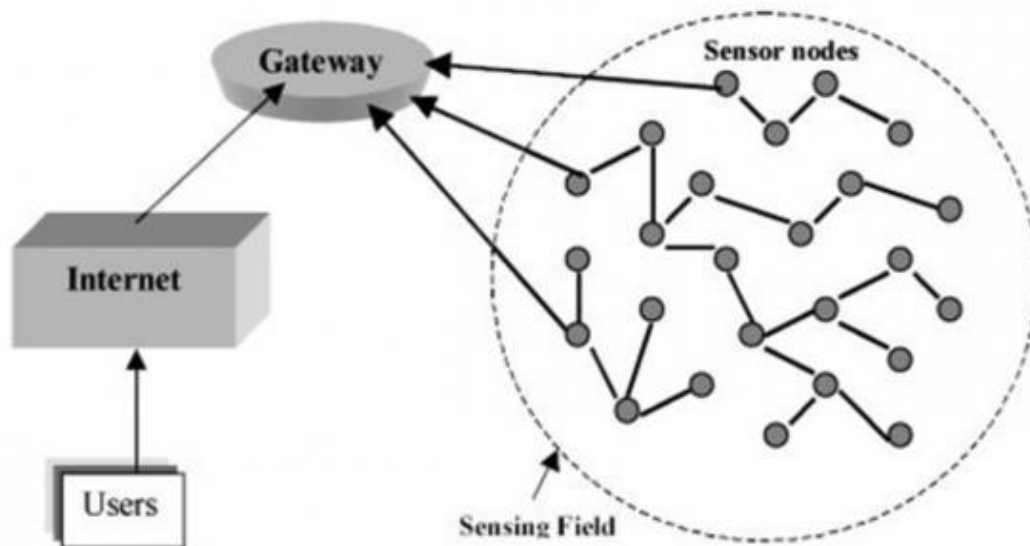


Fig. 3.1.1-A. Estructura de una red de sensores.

La Fig. 3.1.1-A muestra la estructura de una red de sensores genérica. Se aprecian algunos componentes en el entorno local como la puerta de enlace (*gateway*) que permite enviar los datos procesados a través de internet y los nodos sensores, que

recogen los datos del ambiente y los transmiten a esta puerta de enlace para poder ser distribuidos.

Esta red interna de sensores puede adoptar varias formas en arquitectónicas según las conveniencias y los requisitos de la aplicación. Se distinguen las topologías en estrella, malladas y en árbol, para concetarse a una estación base que trabaja como puerta de enlace.

- **Topología en estrella.** Los nodos sensores se conectan directamente a la estación central mediante una única conexión (*single-hop*).
- **Topología mallada.** Los nodos están interconectados entre si además de la estación central. El camino que conecta dos nodos no es único ya que, además de estar conectados la estación central, existen rutas entre ellos a través de otros puntos.
- **Topología en árbol.** Se trata de una estructura jerárquica. Los nodos se conectan al nodo de su nivel inmediatamente superior.

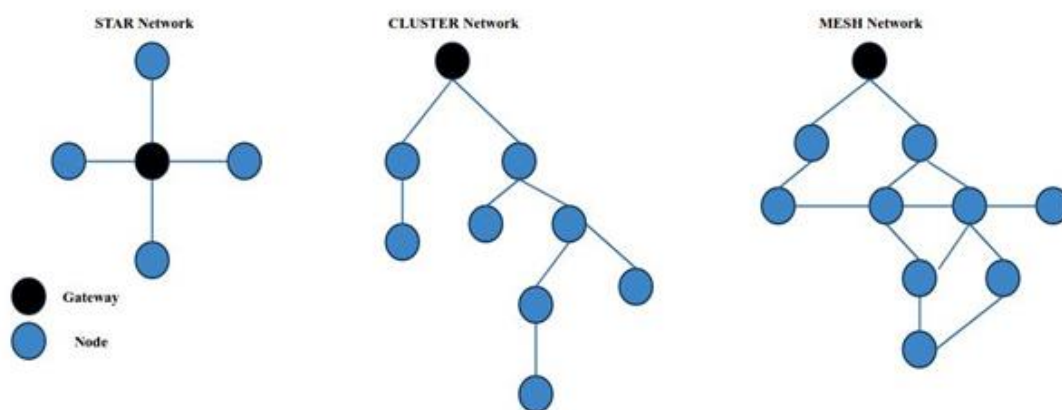


Fig.3.1.1-B. Topologáis de redes de sensores (WSN) [20].

En los próximos párrafos se resume el funcionamiento de estos dos componentes que forman la red: los nodos sensores y las puertas de enlace.

Nodo sensor

Uno de los componentes de las WSN son los nodos sensores. Para que la puerta de enlace reciba información fiable y la retransmita por la red se necesitan dos unidades distinguidas para transformar condiciones ambientales externas en información etiquetada: un sistema de sensorización y un sistema de evaluación de los datos [21].

El sistema de sensorización se encarga de convertir eventos medioambientales en datos digitales. Mediante conversores, por ejemplo, A/D, y un pequeño sistema de *buffering* se convierten señales eléctricas analógicas en paquetes de datos digitales que posteriormente podrán ser procesados por un microcontrolador.

Por otro lado, el sistema de evaluación está conectado a múltiples sistema de sensorización de los que recibe los mencionados paquetes digitales. La tarea consiste en convertir los datos en crudo de los sistemas de sensorización en datos fiables y temporalmente consistentes mediante teoría de juegos cooperativa o la lógica Fuzzy [21] [22].

Puerta de enlace (*gateway*)

Las puertas de enlace son elementos que permiten enrutar paquetes de datos recolectados a destinatarios que se encuentren más allá del perímetro de la red local.

La versatilidad de los *gateway* debe permitir conexiones entre nodos sensor y dispositivos que no se encuentran directamente conectados a la red, la limpieza de los datos, filtrado, preprocesado, optimización y almacenamiento temporal hasta la visualización y análisis básico de datos a través de aplicaciones implementadas en el propio dispositivo [23].

Estos elementos tienen la capacidad de filtrar los datos heterogéneos y convertirlos a un formato interpretable por la aplicación de destino, como la que se ejecuta en un servidor ubicado en la nube. Reciben información tanto de manera cableada como inalámbrica. En la recolección de datos, los sensores transmiten paquetes de datos sin procesar, normalmente obedeciendo protocolos de bajo consumo energético como ZigBee o BLE. Por otro lado, otra interfaz permite conectarse a la red exterior y trabaja con protocolos de más alto nivel como protocolos REST, MQTT o CoAP [21].

A lo largo del punto 3.1.1 se ha revisado la actividad de los nodos sensores como las *gateways*. Se ha pues en manifiesto la necesidad de procesar datos, limpiarlos, filtrarlos, e implantar funciones lógicas. Estos procedimiento se programan sobre microcontroladores. En la Tabla 3.1.1-A se muestran las características de los más utilizados, entre otros:

	Arduino UNO Rev 3	ESP32	Raspberry Pi 4
Código abierto	Sí	Sí	Sí
Ventajas	<ul style="list-style-type: none"> -Bajo coste. -Librerías online. -Fácil de aprender. -Multiplataforma. 	<ul style="list-style-type: none"> -Bajo coste. -Bajo consumo. -Compatibilidad con nodos sensor. -Diseño robusto. 	<ul style="list-style-type: none"> -Alta capacidad de procesamiento. -Alta potencia gráfica. -Ancho de banda sin limitaciones. -Cifrado SSL/TLS.

Inconvenientes	-Baja capacidad de procesamiento. -Baja capacidad de almacenamiento. -Librerías poco eficientes. [24] -Componente extra para cifrado. (Arduino Shield)	-Cifrado WPA/WPA2 en Wi-fi. -Novedosa respecto ESP8266 y poco testada	-Coste elevado. -Coge temperatura. -Coste energético.
Aplicaciones	-Control y monitorización de condiciones ambientales. -Róbotica básica.	-Microcontrolador para IoT.	-Domótica. -Multimedia.
Lenguajes	Arduino IDE	IDE Arduino, MicroPython, Mongoose OS, Espruino, RTOS	Python, Java, Javascript...

Tabla 3.1.1-A. Comparativa entre microcontroladores

A partir de la unidades caracterizadas en la tabla 3.1.1-A, se le permite al gemelo físico preparar los datos recogidos por la red de sensores para ser enviados en el formato deseado y a través del protocolo de comunicación adecuado al gemelo digital. Las tecnologías que permitirán enlazar los dos gemelos tanto a largas distancias como cortas se describen en la próxima sección.

3.1.2. Transmisión de los datos

¿Qué requisitos exigen las comunicaciones entre los gemelos físicos y digitales? Un *Digital Twin* deja de ser un *Digital Twin* si no puede reflejar el estado actual del gemelo físico. Se debe escoger la tecnología adecuada para poder consumir la información generada en el sistema real y actualizar de manera continua el estado del DT.

Esta transmisión debe ser preferiblemente inalámbrica, debido a la cantidad de sensores que pueden llegar a formar parte de la red. También, en su artículo distingue tres categorías para de las aplicaciones que trabajan con sistemas ciberfísicos [25]:

- Aplicaciones PPP en tiempo real, en las que el estado del dispositivo se debe transmitir de forma fiable y con baja latencia.
- Aplicaciones dedicadas a monitorizar sistemas, las cuales tienen como imperativo una recolección periódica de datos. En esta categoría se prima el bajo consumo energético de los nodos sensores y el uso eficiente del ancho de banda.

- Aplicaciones dedicadas al intercambio de datos, es decir, que exigen una comunicación bidireccional para actualizar no solo los DT, sino también las configuraciones del modelo físico.

Los gemelos digitales mantienen una relación estrecha con la tecnología IoT. Los datos analizados y manipulados por el gemelo digital se basan en la actividad registrada por los sensores en el gemelo físico. Es por ello que se requiere del sistema de comunicación adecuado que permita enlazar en tiempo real ambos agentes y habilitar el flujo de estas cantidades ingentes de datos recogidas.

En primer lugar, se enumeran protocolos de capa de enlace indicados para soportar transmisiones en un entorno IoT. Los sensores con compatibilidad IP, es decir, capaces de conectarse de manera autónoma a su servidor o broker, se asociarán directamente con los protocolos WAN (Wide Area Networks). En contraposición, los dispositivos que necesiten hacer uso de puertas de enlace (gateways) para transmitir información utilizarán habitualmente redes de menor rango de alcance. Destacan las redes 4G, 5G, SigFox y LoRa WAN en el largo alcance y Bluetooth, Z-Wave, Wi-fi (802.11ah)[26], ZigBee y HART en redes WLAN.

En segundo lugar, se indican las principales características de los protocolos más usados a nivel de capa de aplicación para las comunicaciones de los datos originados en los sensores. Estos protocolos serán HTTP y MQTT. Se mencionan otros protocolos menos populares que se también son utilizados y que se usan en casos más específicos que los dos anteriores. Estos serán SDTP, OPC UA, AMQP y CoAP.

Capa de enlace

La capa de enlace es la encargada de definir la naturaleza física de la comunicación establecida entre nodos. Estas redes pueden clasificarse según el rango de cobertura que ofrecen. Entre las redes de menor alcance, que estarían basadas en la utilización de *gateways*, existen algunos ejemplos como Bluetooth, Z-Wave, Wi-Fi HaLow, Zigbee, HART, entre otras. En los siguientes párrafos se indican las principales características de las más utilizadas:

- **Bluetooth.** Desarrollado por Ericsson se recoge en el estándar IEEE 802.15.1 [27]. Fue diseñado originalmente para la transferencia inalámbrica de datos en redes PAN (*Personal Area Network*). Trabaja en las bandas no licenciadas de 2.4GHz y tiene un coste energético elevado, requiere de un proceso de configuración de conexión previo y su alcance es relativamente corto. Posteriormente, Nokia desarrolló una nueva versión para combatir estas limitaciones: *Bluetooth Low-Energy* (BLE) [28]. BLE ofrece topologías de malla y broadcast además de dar soporte a funciones de descubrimiento automático de nuevos dispositivos [25].

Ciertamente, para BLE, las tasas de transferencia no son superiores a otras tecnologías como Wi-fi (802.11) o Bluetooth Clásico, pero su bajo consumo permite que los dispositivos que lo utilizan trabajen durante períodos centenas de veces mayores [29]. Los autores de [30] proponen la arquitectura *Bluetooth Low-Energy* como la solución más eficiente para la transferencia de información

coleccionada entre los dispositivos de recolección de datos y las puertas de enlace (*gateways*).

- **Z-Wave.** Fue desarrollado por Zensys y está orientado a las comunicaciones IoT especialmente en *Smart Homes* y pequeños dominios comerciales [25]. Z-Wave trabaja en la banda ISM licenciada de 900MHz y también 2.4GHz y ofrece un tasa máxima de transferencia de 40kbps y 200kbps, respectivamente.

Z-Wave nació como una solución para no saturar las redes de comunicación inalámbrica actuales como Wifi o Bluetooth. Además, forma topologías de redes en malla, lo que permite conectar múltiples dispositivos entre ellos y que, a su vez, su carácter jerárquico de maestro-esclavos no haga depender la posición del dispositivo de la estación base. Por otro lado, Z-Wave es una buena solución en redes heterogéneas al permitir la comunicación entre dispositivos de proveedores diferentes. Otro punto de vista es la ganancia en fiabilidad en las redes malladas debido a la redundancia de caminos posibles por los que enrutar los paquetes de datos.

- **802.11ah.** También llamado Wi-Fi HaLow, es un protocolo de red inalámbrica que se terminó en 2017. A diferencia de actuar en las bandas de 2.4GHz y 5GHz como los protocolos Wi-Fi convencionales, actúa en las bandas de 900MHz.

Wi-Fi HaLow nace de la familia del estándar 802.11 poniendo su punto de mira en las tecnologías IoT, así desafía a los principales retos de Wi-fi. Esta nueva rama del estándar Wi-Fi ha sido adaptada para convertirse en una solución óptima a diferencia del protocolo convencional [31]:

- Aumenta el rango de cobertura.
 - Solventa los retrasos por la congestión de la banda de 2.4GHz.
 - Facilita la coexistencia de gran número dispositivos incorporando *Restricted Access Window* (RAW).
 - Reduce el consumo energético mediante transmisiones más cortas.
- **Zigbee.** Se trata de un conjunto de protocolos de alto nivel basada en el estándar IEEE 802.15.4. Se caracteriza por ser un protocolo de alto nivel orientado a las comunicaciones inalámbricas WPAN y para dispositivos de bajo consumo.

Trabaja en las capas físicas y MAC. En cuanto a la capa física, mencionar que trabaja en las bandas ISM para usos industriales 868 MHz (Europa) y 2.4Ghz para todo el mundo. Ofrece una velocidad de un máximo de 250Kbs. Según la función que tengan en la red, existen tres tipos de dispositivos Zigbee:

- a. **Coordinador.** Define las rutas que deben seguir los dispositivos para interconectarse entre ellos.

- b. Router. Interconecta grupos de dispositivos que no tienen visibilidad entre ellos. Se ejecuta el código a nivel de aplicación.
 - c. Endpoint. Tiene las funciones básicas y suficientes para conectarse con el enrutador o el coordinador, un ejemplo de endpoint podría ser un interruptor inteligente.
- **HART.** La versión dedicada a comunicaciones inalámbricas de estándar HART, WirelessHART (*Highway Addressable Remote Transducer Protocol*) fue desarrollada por la Fundación HART y publicada en el 2007. Está diseñada para ejecutarse en entornos dedicados a la automatización de procesos y control de maquinaria industrial [25]. Al trabajar en su capa física con el estándar WPAN 802.15.4, opera en las bandas de 868MHz en Europa y con una tasa de transferencia básica de 250kbps y cobertura a 10 metros.

WirelessHART puede adoptar tanto topologías en estrella como malladas. El uso de AP (*Access Points*) permite conectar los dispositivos con su puerta de enlace e instalar múltiples de estos intermediarios lo acompañan beneficios como una reducción de la congestión de tráfico de red, la reducción de latencia, un aumento de la productividad y una mejora en la fiabilidad de la red. Para cumplir con los requisitos de las aplicaciones de control industrial, WirelessHART adopta un carácter centralizado utilizando un administrador de red central [32].

Por otro lado, en cuanto a la redes WAN de largo alcance destacan los protocolos 5G, Sigfox, LoRaWAN, NB-IoT, entre otros:

- **5G.** Es una red celular que prevé consolidarse en el mercado a alturas del 2025, según la Asociación GSM, incompatible con dispositivos enfocados solo en recibir cobertura 4G. La tecnología 5G opera en tres bandas de frecuencia diferentes [33]:

	Frecuencia	Tasa de transferencia	Rango de cobertura
Low Band 5G	600-700MHz	30-250Mbps	100+ km2
Mid Band 5G	2.5-3.5GHz	100-900Mbps	1-10 km2
High Band 5G	24-39GHz	1-3Gbps	1~ km2

Tabla 3.1.2-A. Bandas de la comunicación 5G

Un punto a favor de las redes 5G en el entorno de los DT es, gracias a su reducida latencia y gran velocidad, dar la oportunidad de incluir experiencias AR y VR.

Por otro lado, la red 5G es ideal para operar con productos domésticos que requieran una conectividad constante y fiable. Además, la tecnología 5G ofrece un plus de seguridad que no ofrecía 4G, esto es por su cifrado de 256 bits, más allá de los 128 bits anteriormente establecidos[34].

- **SigFox.** Una solución que provee la compañía francesa Sigfox destinada a las aplicaciones IoT que requieran comunicar menor cantidad de datos e incluso de manera intermitente mediante una cobertura de largo alcance y bajo consumo. Sigfox actúa en bandas no licenciadas de 868 MHz en Europa. En un primer momento se consolidó para dar soporte solo a conexiones uplink y después fue mejorado para ofrecer también servicio downlink.

SigFox es una tecnología UNB (*Ultra Narrow Band*), ya que los dispositivos que se conectan mediante ella lo hacen en franjas de apenas 100Hz lo que se traduce en un ruido y consumo de energía mínimo. El inconveniente que la tasa de transferencia no supera los 100bps.

- **LoRaWAN.** Es un protocolo de comunicaciones basado en LoRa estandarizado por la LoRa Alliance en 2015, aunque su desarrollo se remonta al 2009 en la start-up Cycleo. LoRaWAN opera en bandas inferiores al GHz con un ratio de transferencia entre 300bps y 50kpbs.

Hace uso una tecnología llamada *Factor Spreading* o Factores de Dispersión (*SF7-SF12*); los *SF* dan la posibilidad de regular la tasa de transferencia, entre estos, por ejemplo, las tarjetas LongRa y ECO Power [35]. Cuanto más alto este *spreading factor*, mayor rango de cobertura y menor velocidad.

En un entorno doméstico, puesto que los sensores que permiten monitorizar el entorno priman el *low-cost* y la duración de su batería y, por otro lado, no requieren *QoS*, hacen tanto de Sigfox como LoRaWAN dos soluciones atractivas [36].

- **NB-IoT.** También llamado LTE Cat NB1 e introducido por la 3GPP en el 2016 desafía la popularidad de otras redes WAN como LoRa o Sigfox. Su principal característica es el reducido ancho de banda sobre los 200KHz y la posible coexistencia de MBB (*Mobile Broadband*) con dispositivos IoT. Esto es, en parte, por la inclusión de eDRX, que permite que no deban mantenerse constantemente las conexiones abiertas.

Que NB-IoT actúe sobre bandas licenciadas se traduce en una mayor fiabilidad, sin embargo o, esta está sujeta a la aprobación del operador para desplegarse. Si algo garantiza NB-IoT es la entrega de la información transmitida, eso sí, a coste de un mayor consumo de los dispositivos con peor cobertura, ya que deberán retransmitir más veces [37][38].

Capa de aplicación

Los protocolos de aplicación permiten etiquetar el tipo de información que está viajando en el paquete durante el proceso de comunicación. Existen un gran número para la capa de aplicación, aunque los más utilizados son: HTTP, MQTT, DDS, OPC UA, AMQP, CoAP. A continuación, se mencionan sus principales características:

- **HTTP (*Hypertext Transfer Protocol*)**. Es un protocolo orientado a transacciones, especialmente de documentos. Fue desarrollado por Tim Berners-Lee y posteriormente publicado como estándar en 1997 por IETF y W3C. Se basa en un modelo del tipo cliente/servidor compatible con los métodos CRUD que habilitan la interacción RESTful, en el que se establece conexión solo cuando el cliente quiere enviar una petición al servidor. Por otro lado, mencionar que el flujo de datos en la comunicación es half-duplex, es decir, que la entidad emisora no puede enviar nueva información mientras esté recibiendo de otra.

La especificación del protocolo HTTP es realmente intrincada y es apto para aplicaciones que requieran mayor control de la comunicación [25]. Las características básicas de su primera versión se recogen en los archivos RFC 7230 al RFC 7235, aunque con los años se ha ido extendiendo a otras especificaciones y versiones: HTTP/2 en el RFC 7540 o el RFC 2818, que incorpora TLS para mejora su seguridad. Su última versión es la HTTP/3, sucesor de HTTP/2 en la que se establece UDP como protocolo subyacente en lugar del anterior TCP, a pesar de todavía ser compatible con este último.

Respecto a la estructura de su trama, HTTP cuenta con un gran inconveniente para las conexiones entre dispositivos que envían una gran cantidad de paquetes pero de poco tamaño a su servidor, y es su cabecera. El tamaño de la cabecera del protocolo HTTP puede provocar serios problemas en el consumo de recursos de red así como grandes retrasos [39].

HTTP puro es no-persistente. Cada vez que se realiza una petición de información al servidor, se establece una conexión a través de un protocolo determinado. Por otro, HTTP/1.1 y sus sucesores ya incorporan configuraciones de conexión persistentes, lo que significa una menor congestión de la red y consumo de CPU.

HTTP/1 y sus sucesores también incorporan *pipelining* [40], un modo que permite enviar varios mensajes sin tener que recibir una confirmación de reconocimiento del receptor.

- **MQTT (*Message Queueing Telemetry Transport Protocol*)**. MQTT es un protocolo orientado a la transmisión de datos a nivel de byte. Fue desarrollado por IBM, presentado en 1999 y es uno de los protocolos más antiguos para la comunicación M2M (*Machine-to-Machine*). Se trata de un protocolo basado en el modelo de publicaciones/suscripciones y permite conexiones *one-to-one*, *one-to-many* y *many-to-many*. La gestión de los mensajes se realiza mediante un intermediario llamado “*broker*”. El flujo de los datos es full-duplex, lo que

significa que el destinatario no tiene que esperar a recibir todas las tramas de un emisor para enviar él de nuevas.

Actualmente, MQTT se encuentra en su versión 3.1.1 y se recoge en los archivos del estándar OASIS así como en la ISO 20922 [41]. Existen una serie de conceptos que hay que entender independientemente para entender el funcionamiento de las conexiones MQTT.

- **Publicaciones/Suscripciones:** Sistema en el que se transmiten mensajes clasificados entre un productor que publica en el canal un mensaje tipado y un consumidor (suscriptor) que señala al intermediario de qué tipo de mensajes se va a querer aliementar durante esa sesión.
- **Mensajes:** Es la información que se publica referente a un tema; se trata de comandos o datos.
- **Temas:** Son la manera de mostrar interés en una temática de mensajes concreta, o también, desde el punto de vista de emisor, la manera de especificar dónde quieres publicar el mensaje.
- **Broker:** Es el intermediario entre emisor y receptor. Recibe todos los mensajes publicados, los filtra y publica de nuevo a los clientes que se suscriban a los temas en cuestión.

A diferencia de las primeras versiones HTTP, MQTT es persistente en su conexión. Cuenta con tres niveles preliminares de QoS: *At most once (QoS0)*, *At least once (QoS1)*, *Exactly Once (QoS2)*.

- **DDS (*Data Distribution Service*).** Es un estándar desarrollado por OMG (Object Management Group) cuyo objetivo es equilibrar predictibilidad y eficiencia [37][42]. Dota a sus aplicaciones con unas políticas de calidad de servicio (QoS) excelentes además de una alta fiabilidad de los datos. DDS hace uso del modelo publicación/suscripción, pero, a diferencia de otros protocolos como MQTT, no cuenta con *Broker*.

Este protocolo puede ejecutarse tanto sobre TCP como UDP, aunque estos dos no son interoperables [25]. Su arquitectura se define por dos capa: DCPS (*Data-Centric Publish-Subscribe*), que se responsabiliza de entregar la información publicada a los suscriptores, y la DLR (*Data-Local Reconstruction Layer*), una capa más bien opcional, que ofrece una interfaz a nivel de aplicación para el acceso a los datos y funcionalidades de la DCPS [43].

DDS se caracteriza por su alta escalabilidad e inserción automática de dispositivos en la red. La suma de la baja latencia con su alta predictibilidad significa que DDS es una solución óptima para aplicaciones en tiempo real. Sin embargo, existen inconvenientes en su consumo tanto de memoria como de recursos computacionales.

- **OPC UA.** Estándar de código abierto de la IEC (International Electrotechnical Commission) desarrollado por la Fundación OPC en colaboración con investigadores, fabricantes y usuarios. Es una arquitectura orientada a servicio publicada en el 2008 con vistas a responder ante las emergentes necesidades del sector industrial mediante un framework que además integra todas las especificaciones del estándar OPC clásico.

OPC UA se caracteriza por estar centrado en la intercomunicación de equipamiento industrial y de sistemas orientados a la recolección de datos y control. También destacar su compatibilidad multi-plataforma y robustez, aunque existen también inconvenientes como su baja eficiencia al tipar los datos ocasionalmente con XML. Es comunmente utilizado a alto nivel para monitorizar y controlar sistemas automatizados y para transmitir y representar los datos en redes heterogéneas [44].

Es compatible con modelos del tipo cliente/servidor como publicaciones/suscripciones en forma de *websockets* y *https* sobre TCP.

- **AMQP (*Advanced Message Queuing Protocol*).** Es un protocolo de capa de aplicación de código abierto centrado en entornos de comunicación mediante mensajes. Fue desarrollado por John O'Hara en la empresa financiera estadounidense JPMorgan Chase. Este protocolo está diseñado para que se garantice la fiabilidad, seguridad, aprovisionamiento e interoperabilidad. Emergió del sector financiero para poder liberar a los usuarios de los sistemas de mensajería no interoperables [45]. A diferencia de los demás protocolos descritos en este documento, AMQP es un protocolo orientado ser implementado en conexiones cableadas y no inalámbricas. AMQ se ejecuta sobre TCP y da soporte tanto a arquitecturas Petición/respuesta como Publicación/suscripción.

Su sistema de comunicación requiere que tanto publicante como con consumidor creen un *exchange* con un nombre determinado que posteriormente difundan entre los demás nodos. Estas entidades Publicante/Suscriptor se darán a conocer entre ellos a través de este nombre. Posteriormente, el consumidor crea una cola asociado al *exchange* instanciado. Los mensajes recibidos de ese *exchange* serán introducidos en la cola mediante un proceso llamado *binding* [46].

Para las seguridad, utiliza TLS o SASL. Provee de tres niveles de QoS, al igual que MQTT: *at most once*, *at least once*, *exactly once*.

- **CoAP (*Constrained Application Protocol*).** Es un protocolo M2M ligero desarrollado por IETF CoRE (*Constrained RESTful Environments*) que se encuentra redactado en el RFC 7252 [47]. Fue diseñado para ser un protocolo REST compatible con las operaciones CRUD. A pesar de que la mayoría de

protocolos REST se ejecutan sobre TCP, el subyacente de CoAP es UDP, que es menos cargado, aunque también es compatible con TCP y SMS.

CoAP tiene el objetivo de permitir que dispositivos de bajo consumo, bajas capacidades de cómputo y comunicación implementen interacciones REST. Se puede dividir en dos subcapas: mensaje y petición respuesta [43].

De manera similar a MQTT, los recursos en CoAP se identifican y localizan mediante URIs en interactúan mediante una técnica denominada “observación de recursos”.

Aunque los protocolos que se ha descrito tanto para la capa de aplicación como para la capa de enlace no son los únicos, sí que son los más utilizados. Una vez transmitidos los datos, la infraestructura que forma el gemelo digital debe contemplar un lugar donde procesarlos.

3.1.3. Procesamiento de los datos

En esta sección se definirá el paradero de toda la información que vuelca el gemelo físico sobre el gemelo digital. Estos datos, aparentemente, no tienen ningún significado por si solos, el gemelo digital dibujará la realidad de su homólogo físico a partir de ellos, por lo tanto, se requerirá de un lugar de procesamiento para realizar los cálculos del comportamiento del sistema. En los párrafos siguientes se plantean algunas de las opciones que son apropiadas para este cometido.

- Procesamiento local (*On-premise*)
- Procesamiento en la nube (*Cloud computing*)
- Procesamiento en el borde (*Edge computing*)

Procesamiento local (*On-premise*)

El procesamiento de datos local se basa en realizar todas las operaciones dedicadas al servicio desplegado en un lugar que el usuario o la organización controle. En este entorno, las personas que proporcionan el servicio no dependen de proveedores y tiene control total sobre el hardware instalado.

Tener las unidades de procesamiento “en casa”, significa que los servidores *on-premise* pueden ser mantenidos por el mismo usuario que los despliega, eso supone que la labor de mantenimiento sea más ágil. Sin embargo, estadísticamente estos tipos de despliegue son más vulnerables, ya que la cantidad de recursos invertida en ellos en referencia a labores de seguridad es menor. Por ello, los favoritos de los atacantes son este tipo de servidores, desatendidos y que rara vez incorporan los últimos parches.

Otro factor a tener en cuenta es el coste. Realmente existe un coste oculto cuando la red cuenta con un grupo de servidores alojados *on-premise*. El coste de oportunidad es el

resultado de no aprovechar aplicaciones o servicios más adecuados mientras sus sistemas quedan obsoletos. Sin lugar a duda, este será una cuestión que el *cloud computing* permite satisfacer, ya que en este segundo caso, el proveedor mantendrá al día su infraestructura [48].

Cloud computing

La computación en la nube es un paradigma de procesamiento de los datos que consiste en procesar, almacenar y acceder a datos a través de la red, atacando un sistema que se haya desplegado fuera del perímetro local en vez de hacerlo a través de dispositivos on-premise o instalados el propio entorno informático de la empresa.

Si bien la infraestructura completa de estos centros de computación ha pertenecido tradicionalmente al mismo que despliega el servicio, hoy en día también existe la posibilidad de “alquilar” la capacidad de procesamiento de las infraestructuras de terceros.

Se conoce como nube privada a aquella infraestructura que, a pesar de no encontrarse físicamente en el entorno informático de la empresa, es administrada por esta misma y solo la propia empresa tiene la capacidad de acceder para llevar a cabo su gestión interna.

Se habla de la nube pública cuando se realiza un despliegue de los servicios fuera del entorno local, pero, a diferencia de la privada, la empresa no dispone de acceso a la gestión interna de la infraestructura.

Una tercera opción es el cloud híbrido, que se trata de una combinación de ambas nubes descritas. Aquí, se aprovecha la seguridad y disponibilidad del cloud privado para desplegar los sistemas más críticos y, por otro lado, también la ágil escalabilidad y rentabilidad de la nube pública.

Edge computing

Es una manera de destinar el cómputo de los datos a centros de procesamiento que estén lo más cercanos posible del lugar donde se generan. La intención es reducir la latencia computacional que provoca las lejanas comunicaciones que presenta el cloud. En el *edge computing* se busca olvidar los centros de procesamiento de datos centralizados y resumir la generación de datos, almacenamiento, procesamiento y comunicación prácticamente a nivel de LAN.

Este sistema de procesar los datos permiten erradicar algunos de los principales inconvenientes de la computación en el *cloud*:

- Seguridad: Desplegar servicios para procesar grandes cantidades de datos en la nube significa depender de que la transmisión de datos a larga distancia sea segura.

- Latencia: La localización remota de los centros de datos significa un coste temporal extra durante el cómputo.

Como se ha mencionado en el apartado dedicado a *cloud computing*, los contenedores permiten desplegar servicios en la nube de una manera más eficiente e intuitiva. El uso de herramientas de gestión de contenedores como Kubernetes, facilita la realización de esta tarea de distribución entre los servidores *edge*. Sin embargo, con la llegada de los *Digital Twins*, se deberá encontrar la manera de distribuir las cargas de trabajo entre contenedores considerando el crecimiento exponencial de la magnitud de nodos sensores; para 2021 se prevé un total de ciento cincuenta mil millones de dispositivos conectados a la red. Este número hace inviable gestionar las cargas de trabajo de manera manual [49].

3.1.4. Almacenamiento de los datos

La arquitectura utilizada para almacenar cantidades ingentes de datos, popularmente denominadas *Big Data*, debe tener la capacidad de almacenar y computar datasets masivos. Los datos en sí se suelen generar de manera desestructurada, lo que se denomina “*raw data*” y, aunque no se definen como “*Big*” un volumen y una capacidad específicos, el almacenamiento del *Big data* normalmente se refiere a volúmenes que crecen exponencialmente a escalas de terabytes o petabytes [50].

Las bases de datos no-relacionales demuestran ser una alternativa viable respecto a los sistemas de bases de datos relacionales usados tradicionalmente [51]. En contraposición a estas, según Oracle [52]:

- Las bases de datos relacionales están basadas en modelos relacionales; almacenan y proporcionan acceso a puntos de datos relacionados entre sí.
- Respecto a las no-relacionales, no existe un esquema exacto predefinido que indique qué tipos de datos se van a almacenar. Como estos no están identificados, no hay una relación directa entre un conjunto de datos u otro.

En los siguientes subapartados se enumeran algunas de las tecnologías más usadas para gestionar grandes cantidades de datos entrantes.

SQL, NoSQL y NewSQL

- **SQL**, un sistema de gestión de base de datos relacional que se caracteriza por su alta escalabilidad vertical. Responde correctamente a aplicaciones que solicitan peticiones complejas (*queries*). El objetivo de SQL es satisfacer el concepto de ACID (atomicidad, consistencia, aislamiento, durabilidad), una serie de propiedades cuya intención es garantizar la confiabilidad de las transacciones en las bases de datos.

Algunos ejemplos de herramientas de gestión SQL son Oracle 12c, MySQL o Microsoft SQL Server.

- **Not only SQL**, es un sistema de gestión de bases de datos de esquema flexible que está orientado a usarse bajo aplicaciones relacionados con el *Big Data* y de tiempo real. Se caracterizan por su gran escalabilidad horizontal para hacer frente a la llegada masiva de datos. NoSQL cubre algunas de las limitaciones de las bases de datos relacionales, por ejemplo, la variabilidad de los datos [51].

Algunos ejemplos de sistemas NoSQL son MongoDB, Redis, CouchDB y RavenDB.

- **NewSQL**, está basada en los sistemas de bases de datos relacionales, mas ofrece soluciones a algunas de las limitaciones de SQL, especialmente sobre aplicaciones dedicadas al análisis de datos y posterior toma de decisiones que tienen como imperativo una alta consistencia de los datos; se implementa un sistema de recuperación basado en redundancias.

NewSQL significa una nueva versión escalable y de alto rendimiento de las bases de datos, que no solo está capacitada para almacenar grandes volúmenes de datos, sino que también respeta ACID de las bases de datos tradicionales.

Algunos ejemplos de sistemas NewSQL son ClustrixDB, VoltDB, NuoDB y Apache Trafodion.

La siguiente figura es una tabla comparativa entre las tres tecnologías en la que se representan las respuestas en su rendimiento ante diversos factores:

	SQL	NoSQL	NewSQL
ACID	SÍ	NO	SÍ
Escalabilidad	Vertical	Horizontal y vertical	Horizontal
Latencia	Alta	Baja	Baja
Consistencia	Alta	Baja	Alta
Disponibilidad	Baja	Alta	Alta
Tolerancia a particiones	Alta	Alta	Baja
Vel. Lectura	Alta	Baja	Alta
Vel. Escritura	Baja	Alta	Alta

Vel. Borrado	Alta	Baja	Alta
---------------------	------	------	------

Tabla 3.1.4-A. Comparativa entre bases de datos SQL, NoSQL y NewSQL

En resumen, las bases de datos SQL respetan modelos de relaciones además de las propiedades del concepto de ACID. Por otro lado, a NewSQL suma a lo anterior un mejor rendimiento y mayor escalabilidad vertical. Por último, bases de datos no-relacionales NoSQL, al no requerir un esquema predefinido, cuentan con mayor escalabilidad horizontal a pesar no concernir ACID.

Sistemas de archivos distribuidos (DFS)

Un sistema de ficheros establece una correspondencia lógica entre los ficheros y directorios y los dispositivos. Si se extrapola esta idea a la red, se podría interpretar como extensión del sistema de fichero local, uno que permita el acceso a los ficheros de un servidor utilizando un determinado protocolo [53].

En un sistema de ficheros distribuidos, los usuarios acceden y operan sobre un sistema de archivos común pese a que estos recursos se encuentren distribuidos físicamente a lo ancho de toda la red local; en implementaciones más ambiciosas, donde por ejemplo interesa interconectar varias sedes ubicadas en lugares diferente, esta idea también pueden ser trasladarse a entornos WAN.

Una de las ventajas de este tipo de sistemas es que se puede optimizar la carga de la red para que los nodos con mucho tráfico deriven recursos compartidos a otras ubicaciones de la red con lo cual se minimiza el riesgo de cuello de botella y se optimiza la velocidad de acceso a la información [54].

En esta solución, los datos son almacenados bajo una distribución en forma de árbol que ofrece mayor flexibilidad y tolerancia a errores a la hora de almacenar la información, ya que podría por ejemplo replicarse u ofrecer autoescalado añadiendo más recursos físicos al dominio.

HBase es la base de datos distribuida más caracterísitica. Se trata de una base no relacional de código abierto, distribuida, escalable y capacitada par almacenar grandes cantidades de datos que forma parte del proyecto Hadoop de la fundación Apache. Esta versión está modelada sobre BigTable (Google).

Almacenamiento en la nube

El almacenamiento en “cloud” no es considerado como un sistema de gestión de bases de datos en si, más bien como otro enfoque donde se hace uso la infraestructura física de un tercero para almacenar la información. En este paradigma, los datos se almacenan más allá del perímetro de la red local sobre una infraestructura que puede estar dedicada a un solo cliente (cloud privado) o a varios (cloud público).

La migración de datos al cloud público también puede realizarse de manera parcial: Existe los denominados “cloud híbridos” que, por ejemplo, una posible escenario consistiría en situar los datos más críticos en la nube privada y los datos más accedidos por el personal del cliente en la nube pública.

Enfatizando sobre el concepto de cloud público, cabe destacar su permisividad ante el ajuste de parámetros como tiempo de despliegue o espacio contratado, entre otros, mediante un sistema bajo demanda.

Nodos de caché estándar – Generación actual	Precio por hora
cache.t3.micro	0,018 USD
cache.t3.small	0,036 USD
cache.t3.medium	0,072 USD
cache.t2.micro	0,018 USD
cache.t2.small	0,036 USD
cache.t2.medium	0,073 USD
cache.m6g.large	0,164 USD
cache.m6g.xlarge	0,327 USD
cache.m6g.2xlarge	0,654 USD
cache.m6g.4xlarge	1,308 USD
cache.m6g.8xlarge	2,615 USD

Fig. 3.1.4 -A. Coste de servicio de Amazon Elastic Cache

La Fig. 3.1.4-A es una instantánea de los diferentes precios que ofrece Amazon Web Services de su sistema “Amazon ElasticCache” para el motor de base de datos relacional “Redis”. Cada una de las filas representa el precio de un nodo de caché estándar de una potencia, capacidad de almacenamiento y desempeño de la red determinados. Por poner un ejemplo, el primer nodo “cache.t3.micro” consiste en el alquiler de pago por hora de utilización de una instancia de BBDD de las siguientes características:

Cache Node Type	CPU virtual	Memoria (GiB)	Desempeño de la red
Estándar – Generación actual			
cache.t2.micro	1	0,555	De bajo a moderado
cache.t2.small	1	1,55	De bajo a moderado
cache.t2.medium	2	3,22	De bajo a moderado
cache.t3.micro	2	0,5	Hasta 5 Gigabits
cache.t3.small	2	1,37	Hasta 5 Gigabits

Fig. 3.1.4-B. Características del servicio de almacenamiento en AWS de instancias `cache.t3.micro`

Otra de las ventajas del almacenamiento en la nube es su alta disponibilidad. Raramente se producirán fallas en el acceso a los datos cuya responsabilidad recaiga en el proveedor. Normalmente, estas entidades garantizan una disponibilidad anual superior al 99%. El 1% restante es un riesgo el cual también existen algunas maneras de cubrirlo. Por ejemplo, *DRaaS* (Data Recovery as a Service) es una categoría de computación en cloud que responde con $RTO = 0$ (tiempo de inactividad) a estos episodios. Un ejemplo es la solución de *Veeam*.

Considerar que los datos están protegidos una vez los se migran al cloud es erróneo. El proveedor del cloud ofrece una infraestructura donde almacenar esta información pero en ningún momento dedicará recursos implícitos en el servicio para proteger los datos del cliente [55].

3.2. Tecnologías para modelos de alta fidelidad

Los gemelos digitales se definen como representaciones virtuales de sistemas físicos. Estos entes deben reflejar el comportamiento real del sistema de manera precisa a partir de la física que se conoce del mismo sistema como a partir de modelos numéricos que representen su funcionamiento construidos a partir de registros de actividad en tiempo real. En esta sección, se enumeran algunas tecnologías adecuadas para construir este tipo de modelos.

3.2.1. Modelos físicos

Un modelo físico consiste en reconstruir sobre un gemelo digital la física de la que se compone la entidad real a la que representa. Partiendo de su apariencia visual mediante modelado 3D hasta las leyes y comportamientos que obedece el objeto, es decir, las ecuaciones que gobiernan su funcionamiento. La profundidad de detalle con la que se modelan algunos objetos es uno de los retos de este campo. En algunas situaciones, podría desconocerse parcialmente la física que hay detrás del componente que se representa. También, podría ocurrir lo contrario, encontrarnos con la necesidad de simplificar el modelo para optimizar su eficiencia a costa de determinadas imprecisiones en sus cálculos.

Los modelos físicos permiten conocer visualmente como se está comportando el sistema. Existen gran número de herramientas que permiten integrar los cálculos de comportamiento a nivel numérico con su impacto real en el sistema a nivel visual, entre las más demandadas para el modelado 3D están Blender, ANSYS Digital Twin, Unity 3D y el software CAD NX, de Siemens.

3.2.2. Modelos semánticos

A diferencia de los modelos físicos, los modelos semánticos se caracterizan por manifestar a través de los datos tanto la física conocida así como obtener nuevos conocimientos sobre el sistema [18]. Por ello, permiten al sistema o usuario tomar decisiones a partir del análisis, interpretación y predicción de los datos recogidos mediante diversas técnicas de generación (p.e. sensorización).

Guiarse por las observaciones y no por leyes predefinidas otorga la ventaja, respecto a la estaticidad de los modelos físicos, de remodelar la ley automáticamente en caso de la adición de nuevos items en su catálogo [56].

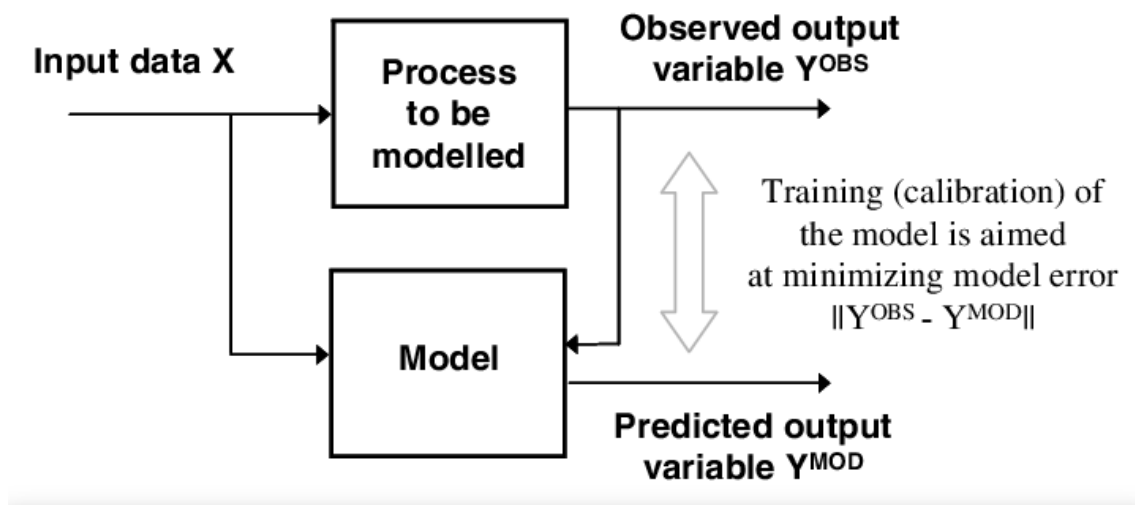


Fig. 3.2.2-A. Comportamiento de los modelos semánticos [57].

La figura 3.2.2-A muestra esquemáticamente el funcionamiento de los modelos semánticos. A medida que se suman nuevas observaciones sobre los datos entrantes (*Yobs*), se modifica ligeramente esta ley (*Model*) para predecir con mayor precisión las salidas (*Ymod*) dada una entrada cualquiera. Este proceso se denomina “entrenamiento del modelo” o “calibración”.

El *Machine Learning* es la tecnología que permite realizar este tipo de predicciones guiadas por datos. Mediante estos algoritmos los ordenadores tienen la capacidad de realizar tareas específicas y tomar decisiones de manera autónoma sin que estas se hayan programado explícitamente [18]. Se distingue el aprendizaje automático de las máquinas en tres categorías:

- **Aprendizaje supervisado**

En el aprendizaje supervisado, se parte de un conjunto de datos de entrada y su respectiva salida conocidos. A continuación se establece una ley interna que permita determinar una salida dada la entrada según las observaciones previas. Se diferencia el tipo de ley según el dominio de la salida: si son valores continuos se habla de modelos de regresión, si es un dominio discreto, de clasificación.

Entre los algoritmos de clasificación más comunes se encuentran las máquinas de vectores de soporte (SVM), árboles de decisión, redes neuronales, *random forest*... Entre los algoritmos de regresión incluimos la regresión de tipo lineal, Lasso, logística y múltiple [58].

- **Aprendizaje no supervisado**

En este ámbito únicamente se conocen las entradas y es el propio algoritmo el que se encarga de categorizarlas en grupos de datos o *clusters*.

Entre los algoritmos más comunes de aprendizaje no supervisado se encuentran los algoritmos de clusterización como *k-means*, los mapas autoorganizados (SOM), clustering jerárquico, el análisis de componentes principales (PCA), *Fuzzy c-Means*... Este último es de gran utilidad cuando un valor de entrada puede pertenecer a varias categorías diferentes [59].

- **Aprendizaje por refuerzo**

Tanto modelos de aprendizaje supervisado como no supervisado pueden ofrecer resultados muy precisos cuando se cuenta con una gran cantidad de datos que entrenen y calibren tales modelos. Pero... ¿qué ocurre cuando el conjunto de datos de aprendizaje es más bien modesto? ¿Pueden realizar predicciones precisas con pocos datos?

En el aprendizaje por refuerzo se actúa como actuaría una mente humana inexperta en solucionar algún tipo de problema. Estos se resuelven mediante prueba y error, buscando optimizar el resultado continuamente. El agente cada vez que realiza una acción se ve recompensado. Esta retribución puede ser tanto negativa como positiva. El objetivo es que conforme el tiempo avanza, se maximice el valor de lo acumulado, ofreciendo una respuesta al problema cada vez más óptima y finalmente, encontrar un conjunto de decisiones que determinen un valor máximo para la acumulación de recompensas.

En el aprendizaje reforzado, se busca encontrar un modelo ideal que se genere a partir de la comparativa entre todas las diferentes soluciones encontradas

En el entorno de los DT, optar por introducir modelos de aprendizaje reforzado parece ser interesante ya que, en muchas de sus aplicaciones, aprender acerca de su comportamiento por prueba y error desde un sistema virtual resulta ser mucho menos costoso y seguro que bien desde el gemelo físico dadas todas las consecuencias que los errores reales podrían conllevar [60].

3.3. Tecnologías para la simulación basada en modelos de alta fidelidad

Una simulación de un sistema físico consiste en determinar las implicaciones de diferentes efectos sobre este mediante una representación digital que permita anticipar la respuesta futura del sistema físico bajo las mismas circunstancias.

Mientras que una simulación básica el objeto simulado se ve limitado por poder ser representado tal y como se conoce en la realidad y de manera totalmente inflexible, en las simulaciones basadas en *Digital Twins*, al conocer todas las características y partes del sistema de una manera muy precisa, existe la posibilidad de predecir cuál será su respuesta o funcionamiento ante diversos eventos futuros, como la adición de nuevos componentes [61].

3.3.1. Plataformas de simulación en tiempo real

Actualmente, tanto en entornos industriales como domésticos, se utilizan los simuladores de software para prototipar y calcular el rendimiento de los sistemas de calefacción, refrigeración y aire acondicionado de un edificio o espacio. Según Ansys [62], estas plataformas permiten:

- Investigar diseños alternativos.
- Entender mejor los elementos que podrían mejorar el rendimiento.
- Explorar soluciones más innovadoras.

En los próximos párrafos se detallan algunos proveedores de simuladores de software que destacan hoy en día en el mercado:

- **ANSYS** permite impulsar a través de su software productos basados en simulación donde tanto diseñadores como ingenieros tienen la posibilidad de analizar el comportamiento de los productos sobre diferentes escenarios y situaciones sin tener que fabricarlos físicamente.
- **AnyLogic** provee de un software dedicado a generar gemelos digitales. Además de la posibilidad de reflejar el comportamiento en tiempo real del sistema, permite realizar pronósticos del estado de la maquinaria a partir de soluciones basadas técnicas de Machine Learning. Integrando los modelos numéricos que definen el sistema con estas técnicas de ML, aseguran satisfacer las tres políticas básicas del mantenimiento [63]:
 1. Reemplazar los componentes según lo programado
 2. Reemplazar los componentes según sus condiciones
 3. Reemplazar los componentes según sus previsiones.

- **NX CAE** es un software de Siemens enfocado a la simulación 3D de productos industriales. Este software se ha desarrollado con el objetivo de solucionar los problemas de análisis estructurales en términos de tensiones, vibraciones, dinámico, acústico, transferencia térmica y aeroelasticidad [64].

Los gemelos digitales son una herramienta que debe permitir entender y responder ante las condiciones presentes de la máquina así como las de un futuro cercano y lejano. Su uso, ofrece la posibilidad de analizar y comparar situaciones a partir de la información recogida históricamente de el sistema físico y su entorno. En base a esto, mediante los softwares de simulación se permite comprobar rápidamente cómo cambios en él y su entorno podrían afectar a su disponibilidad, rendimiento y eficiencia [63].

3.2.2. Interacción bidireccional

Una de las características fundamentales de los *Digital Twins* es que exista una comunicación en ambas direcciones entre la entidad física y digital. A diferencia de las sombras digitales o simples modelos digitales, la información que el DT genera a partir de las enormes cantidades de datos entrantes no solo debe destinarse a un uso puramente digital; las simulaciones digitales deben servir para predecir episodios futuros de la máquina y tomar decisiones de manera automática con el fin de optimizar su rendimiento, por tanto, necesitan una vía de comunicación para ofrecer su feedback al PT.

Hasta ahora, se ha descrito qué hacer con los datos que viajan en un primer momento en sentido al mundo digital, en este apartado, se enumeran una serie de tecnologías que permitan llevar nueva información de vuelta a la realidad y tomar decisiones de manera automática y en tiempo real sobre la propia configuración del gemelo físico.

Por un lado, gran parte de los proveedores de los sistemas domóticos instalados en la viviendas cuentan con su propia plataforma a la que los clientes pueden acceder para controlar de manera remota sus productos:

Panasonic ofrece a sus clientes **Panasonic Smart Cloud**, un servicio en tiempo real que permite visualizar el estado de sus productos de calefacción y refrigeramiento, como su consumo, así como realizar ajustes de manera remota en su temperatura o temporizador. También, la entrega de Panasonic posibilita a los técnicos ver y predecir averías y errores desde sin una asistencia presencial.

Zehnder, proveedor de sistemas de ventilación y calidad de aire, ha desarrollado **IoT Zehnder Cloud**, recoge un histórico de datos que permite analizar y mejorar la calidad del aire del interior, mostrando información al usuario a través de un *dashboard*. Además, notificará cuando la calidad disminuya e informará sobre cómo se debería responder ante el imprevisto.

Siemens, por otro lado, proveedor de electrodomésticos a nivel mundial, convierte tu casa en un hogar conectado mediante **Home Connect**. Este servicio permite desde

consultar los alimentos que hay en tu frigorífico, ajustar la temperatura y programar su bloqueo a programar el arranque de tu cafetera inteligente desde tu smartphone.

Muchos de estos entornos utilizan herramientas de comunicación específicas para establecer una conexión que permita comandar los controladores lógicos de manera remota:

- **Modbus.** Es un protocolo público basado en la arquitectura maestro/esclavo, utilizado para permitir la comunicación con los PLC (*Programmable Logic Controller*). Es fácil de implementar además sin suponer restricciones al manejar los bloques de datos. Modbus trabaja a nivel de aplicación y puede implementarse sobre diferentes capas físicas. En arquitecturas cliente/servidor, se ejecuta sobre TCP/IP. A diferencia de otros protocolos, como OPC UA, Modbus no cumple las exigencias de seguridad que requieren las redes IoT. Sin embargo, existen soluciones que permiten integrarlo en el mundillo, como el *gateway* OPC UA Modbus Server, que encripta los mensajes, entre otras cosas, manteniendo sus funcionalidades nativas [65].

Si el dispositivo con el que se quiere establecer conexión no reconoce el protocolo Modbus, existen traductores que todavía se benefician de algunas interfaces estándar, como *Barionet 1000*, de la productora QIBIXX [66].

- **KNX.** Es un estándar internacional de comunicación de dispositivos industriales. KNX garantiza la interoperabilidad tanto cableada como inalámbrica (KNX-RF) diferentes fabricantes certificados a los que se les exige el cumplimiento de la ISO 9001. La asociación KNX desarrolló la herramienta ETS para programar y supervisar los dispositivos conectados a la red desde el *gateway* IP.

KNX es uno de los protocolos principales en las industrias dedicadas a la climatización, ventilación y HVACs. Cubre las capas de enlace, red y transporte del protocolo OSI y puede direccionarse hasta 65.000 dispositivos intercomunicados debido a su naturaleza peer-to-peer.

Dada la naturaleza IPv4 de KNX, el paso a la compatibilidad eficiente con IPv6 es todo un reto pero, debido al crecimiento exponencial de dispositivos IoT y, por consiguiente, la ocupación de las direcciones IPv4, el salto es necesario. Seifried et al. proponen un rediseño de la arquitectura para un próximo estándar KNX IP, una versión mejorada de los KNXnet / IP y KNX IP actuales, en los que se hace un uso inadecuado del conjunto de funciones IPv6 [67].

Una vez presentadas las tecnologías más utilizadas a la hora de desarrollar los gemelos digitales, cabe introducir de manera conceptual el gemelo físico sobre el que se trabaja en este proyecto. El Capítulo 4 se dedica a explicar el funcionamiento de un sistema bomba de calor genérico, que será posteriormente llevado al caso concreto del entorno de energético del proyecto en el Capítulo 5.

4. Bomba de calor

Recordemos que en este proyecto se va a desarrollar un digital twin que represente el funcionamiento del sistema de bomba de calor instalado en el Centro de Innovación Ca Ses Llúcies. Para facilitar la comprensión del entorno energético, se piensa que es interesante realizar previamente un breve repaso al concepto de sistema bomba de calor, sus componentes y las transformaciones energéticas que ocurren en él. En el caso que el lector ya esté familiarizado con el comportamiento interno de los sistemas de climatización domésticos, podría obviarse este capítulo.

Los sistemas de climatización basados en bombas de calor aire-agua, utilizan la energía absorbida del aire ambiente para proporcionar frío o calor a través de la unidad interior, independientemente de si se trata de aire o agua sanitaria. Las bombas de calor son sistemas de refrigeración reversibles, es decir, que se utiliza el mismo procedimiento tanto para proporcionar frío como calor, esto se consigue invirtiendo los intercambios de energía térmica realizados en cada una de sus unidades.

Los tipos más comunes de bombas de calor son de aire como fuente y tierra. Las bombas de calor de aire como fuente son las más usadas para el calentar y refrigerar viviendas, transfieren calor entre el aire de dentro de la casa y el exterior. Por otro lado, las bombas de calor geotérmicas, transfieren el calor entre el aire del interior de la vivienda y la tierra del exterior. En cuanto a estos últimos, se tratan de sistemas más caros de instalar, pero a la vez más eficientes y con menores costes operativos ya que la diferencia de temperatura del suelo exterior entre temporadas es más moderada que en el caso del aire exterior. De todas formas, en este estudio se centra en trabajar específicamente sobre las bombas de calor aire-agua.

Estos sistemas están compuestos por dos unidades: interior y exterior. Por otro lado, un circuito hermético que las conecta por el que circulará el fluido refrigerante en diferentes estados, definidos por propiedades como la temperatura, presión, etc.

Una bomba de calor tiene la labor, de manera simplificada, de trasladar calor desde la entrada de una unidad a la salida de la otra a través de un circuito:

- En verano, la bomba de calor extraerá el calor del interior de la vivienda para trasladarlo a la unidad exterior y cederlo al aire ambiente. De esta manera se refresca el aire una habitación absorbiendo su propia energía (*refrigeración*).
- En invierno, se absorberá el calor del exterior para cederlo al interior de la casa, sea para calentar un depósito de agua o el aire de una habitación (*calefacción*).

Desde el punto de vista del refrigerante que circula por el sistema, absorber calor del exterior significa calentar el refrigerante, ceder calor al exterior significa enfriarlo. Cuando se desea provocar un calentamiento del refrigerante mediante este intercambio energético, se realiza mediante un evaporador, cuando el intercambio lo enfría, estará circulando por el condensador.

A continuación, con la ayuda de la Fig. 4.1-A y Fig 4.1-B se redacta detalladamente el funcionamiento para cada uno de los dos casos descritos (*calefacción y refrigeración*).

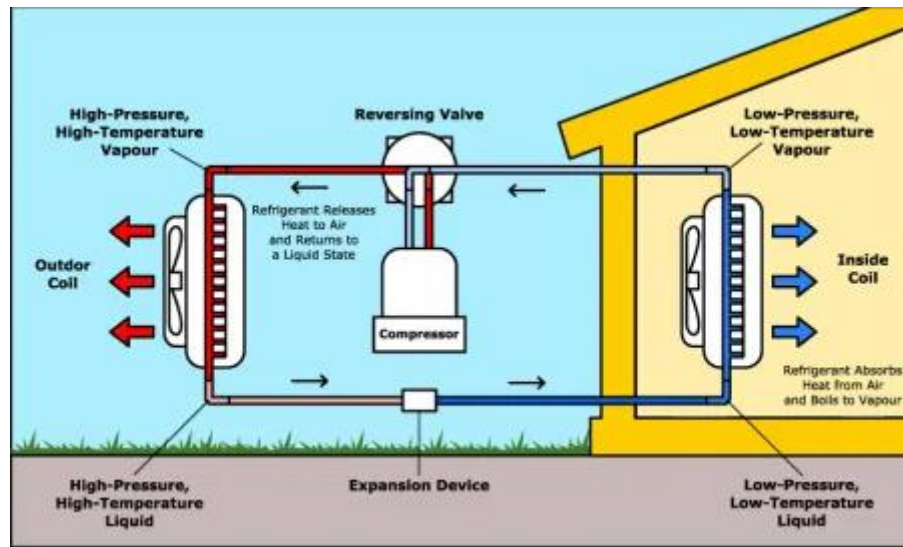


Fig. 4.1-A. Funcionamiento de una bomba de calor en modo refrigeración.

- En modo refrigeración 4.1-A, el refrigerante absorbe el calor del fluido que circula por el interior de la vivienda a través de la unidad interior y conduce esta energía al evaporador, que la libera en la unidad exterior.

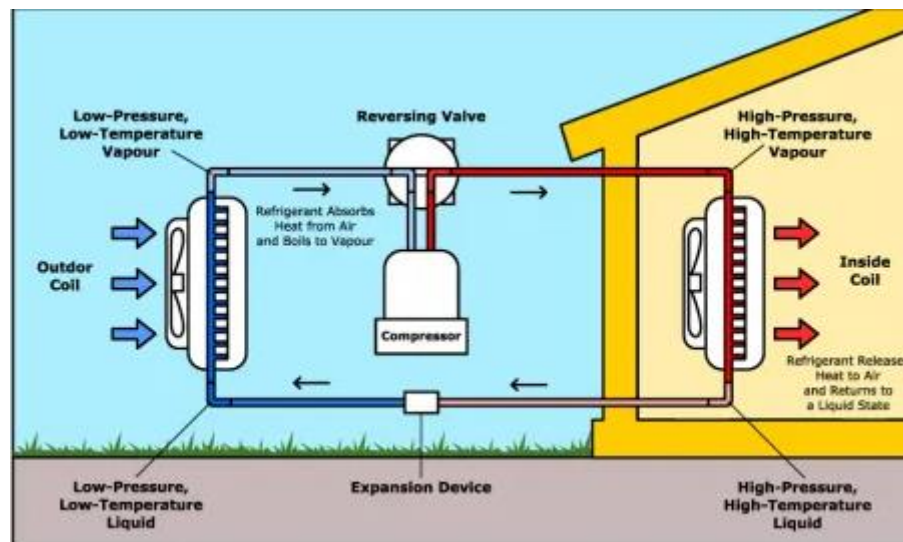


Fig. 4.1-B. Funcionamiento de una bomba de calor en modo calefacción.

- En modo calor (Fig. 4.1-B), el refrigerante absorbe el calor del aire exterior, se evapora y transmite esta energía calorífica a la unidad interior, donde se cede al fluido que circula por el interior de la vivienda a través del condensador. En el

entorno con el que se trabaja en este proyecto, el fluido interior consiste en agua corriente sanitaria.

Existen otros elementos que no están involucrados de manera directa, a diferencia de evaporador y condensador, con el intercambio energético. Pero, estos elementos, ayudan de manera interna al sistema a que el refrigerante adquiera una serie de propiedades para sacar el máximo provecho a esta absorción y cesión de temperatura.

- **Valvula de expansión.** Disminuye la presión del fluido refrigerante para controlar la entrada líquida al evaporador.
- **Compresor.** Aumenta la presión del refrigerante a su paso por el circuito así como su temperatura. La entrada de compresor debe ser totalmente gas, de allí la necesidad de un leve sobrecalentamiento para evitar averías.

El estado del refrigerante lo definen su temperatura, presión y volumen. Dado que los procesos vistos afectan únicamente a su presión y temperatura, es fundamental que tanto la compresión del gas como la expansión a líquido sean isocóricas, es decir, que no haya un modificación en su volumen (*p.e.* pérdidas de gas). Este debe estar, por lo tanto, totalmente aislado del exterior en todo momento.

Una vez conocidas las bases del comportamiento de un sistema de climatización genérico, próximo capítulo está dedicado a presentar el caso concreto con el que se trabaja en este proyecto.

.

5. Entorno de trabajo

El entorno de trabajo de este proyecto se sitúa en la innovadora instalación de Ca Ses Llúcies, ubicada en el mismo campus universitario. El sistema de HVAC al que se hace referencia es un sistema de bomba de calor aire-agua del proveedor Panasonic y modelo WH-SDC05H3E5.

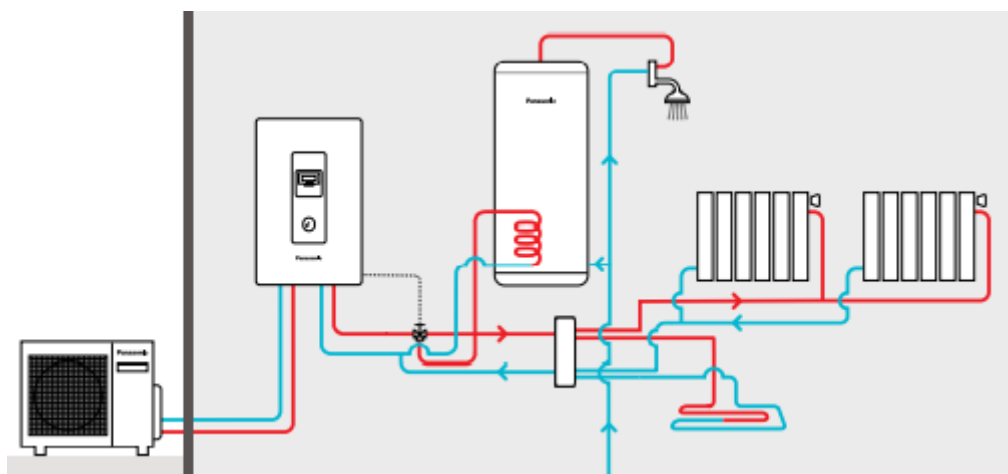


Fig. 5-A. Esquema interno del sistema de bomba de calor Panasonic WH-SDC05H3E.

En la figura 5-A se muestran las unidades exterior e interiores del sistema Panasonic Aquarea instalado en Ca Ses Llúcies. En la mitad izquierda y de izquierda a derecha aparecen: unidad exterior (evaporador), unidad interior (condensador), tanque de agua corriente sanitaria. En la mitad derecha aparecen los elementos que calientan el domicilio: radiadores, suelo radiante, duchas, etc.

Esta solución está desplegada parcialmente sobre un entorno privado. Por ello, la información en tiempo real generada internamente por la maquinaria así como la unidad de control solo pueden ser manipulados a través de las herramientas que el proveedor ofrece, en este caso, equipamiento como el mismo cuadro de mandos o a través de aplicaciones *cloud-based* como Aquarea Service Cloud.

Afortunadamente, la *Universitat de les Illes Balears* ha optado por instalar por motu propio varios sensores que permiten medir de manera periódica algunos de los valores que definen el estado actual de la bomba. Estos son:

1. Temperatura en planta
2. Temperatura exterior
3. Consumo energético del sistema
4. Temperatura consignada del tanque de agua
5. Temperatura de impulsión del tanque de agua
6. Temperatura de retorno al tanque de agua

Esta información es visible a través de la intranet basada en la plataforma Grafana que dispone la universidad. Se puede acceder a través de la red del mismo campus o través el servicio de VPN que también proporciona la universidad a sus alumnos vía solicitud.

La red de sensores recoge información del entorno a todas horas, por lo que se puede obtener información de las mediciones también en momentos del día en que la sistema no esté operativo.

En cuanto a la programación establecida por la propia universidad para el arranque de la bomba, solo se contempla su uso en los días fríos, por lo que únicamente ejercerá su función en modo calentamiento. El arranque se ha establecido de manera fija a las 8:00AM de lunes a viernes.

Haciendo referencia a la enumeración anterior, existe una funcionalidad que permite exportar en formato .csv desde la intranet el histórico de datos generado por cada uno de los sensores filtrando por fecha de inicio y fin. De esta manera, se ofrece poder descargarlos manualmente y trabajarlos de manera local.

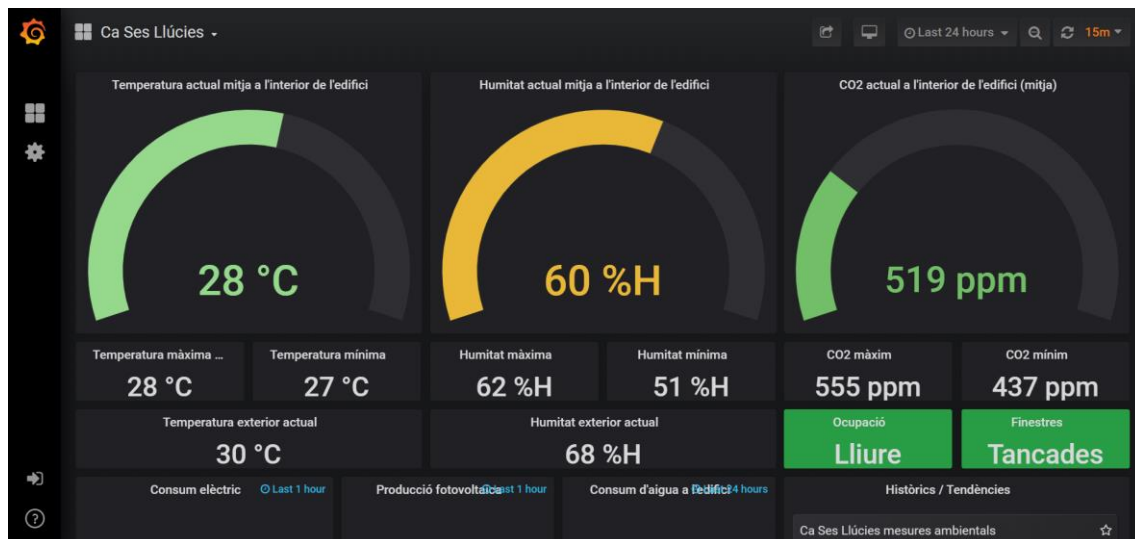


Fig. 5-B. Vista inicial del portal unviersitario de Grafana

Con esta herramienta, el usuario tiene a su disposición el histórico completo de datos recogido por la red de sensores del campus hasta la fecha

5.1. El problema

En este proyecto, se busca mejorar la eficiencia y el rendimiento del comportamiento del sistema de bomba de calor mediante el uso de los conceptos introducidos.

Las plataformas desplegadas por el proveedor, no suelen incorporar un alto nivel de transparencia del funcionamiento del sistema a nivel de usuario. Tampoco estos agentes gozan de la autonomía suficiente para adaptar el mismo sistema físico al que representa

a los diferentes escenarios registrados durante su tiempo de uso; el gemelo digital permitirá que el sistema aprenda de manera autónoma de los escenarios registrados previamente con el fin de optimizar su comportamiento.

A través de la aplicación desarrollada, el usuario deberá poder realizar un seguimiento del sistema instalado a través de una interfaz gráfica así como interactuar con este. Por otro lado, el flujo de información entre el gemelo físico y el gemelo digital debe ser bidireccional y constante, donde el sistema digital deberá estar sincronizado con la actividad registrada a partir de la red sensorial que incorpora su homólogo físico.

5.2. Limitaciones del entorno

Para llevar a cabo este proyecto se deben de tener en cuenta algunos factores que limitarán el servicio que pueda ofrecer el gemelo digital.

- No se ha tenido la posibilidad de acceder remotamente a los controladores lógicos (PLC) del sistema HVAC. Por tanto, la implementación de la solución no incorpora un sistema que ofrezca bidireccionalidad directa en la información que fluye entre gemelo físico y el gemelo digital. Sin embargo, este punto no disminuye la efectividad del proyecto; los beneficios del uso de gemelos digitales pueden visualizarse sin manipular directamente el sistema PLC.
- La generación de los datos a través de los sensores será en diferido. Se deberá establecer una fecha sobre la que empezar a simular esta recolección ya que la única vía de acceso a estos es mediante el histórico exportable desde la intranet. De todas formas, la utilización de un registro de actividad pasado no impide obtener un reflejo del funcionamiento de bomba basado en datos reales.
- El carácter privado del software de *Panasonic* obliga a implementar una solución completamente desde cero y a desplegarse en entornos externos al del proveedor. Sin embargo, dado que el software está totalmente administrado por el cliente, se agiliza el proceso de desarrollo de nuevas funcionalidades en el gemelo digital. Además, obtener un gemelo digital del sistema no suministrado por el proveedor facilita el control y el mantenimiento sobre su software.

De cara al futuro, la libre instalación de nuevos tipos sensores IoT que registren la actividad del sistema permite enviar nueva información al gemelo digital y, por consiguiente, un mayor conocimiento sobre la máquina y su entorno. El entorno IoT desplegado con el fin de monitorizar el estado de las instalaciones del Centro de Ca Ses Llúcies como en el resto del campus universitario permite considerar en trabajos futuros la creación de un ecosistema de gemelos digitales interconectados.

6. Planteamiento y análisis de requisitos

Una vez conocido el problema que presenta el entorno energético, en este capítulo se especifica el motivo y los beneficios de aplicar un *digital twin* sobre el sistema de climatización con el que se va a trabajar. Posteriormente, se detallan los requisitos que debe satisfacer el producto diseñado. Estos deben de estar alineados con los objetivos del proyecto y resolver el problema planteado.

6.1. ¿Por qué digital twins?

Entonces, ¿por qué se deberían utilizar digital twins en sistemas de refrigeración y bombas de calor? ¿Qué servicios se potencian con esta tecnología? ¿Qué puede extraerse de los modelos que ofrecen los DT?

A la hora de crear un gemelo digital, se trata de disminuir el esfuerzo en crear las representaciones a través de modelos que estén orientados al servicio. Estos modelos deben verificar que los sistemas cumplan con los servicios que se esperan de ellos, que sean modulares, reutilizables y con un equilibrio entre su complejidad y fiabilidad.

Viéndolo desde otra perspectiva, los DT permitirán encontrar nuevos conocimientos sobre el sistema y cómo se deben usar, constituyen la base para una variedad de servicios:

- El proceso de seguimiento del sistema, que implica analizar en tiempo real el funcionamiento, el rendimiento de los componentes y del sistema como un todo.
- El diagnóstico de futuros errores en sus etapas tempranas a partir del un monitoreo continuo de los mecanismos y los conocimientos técnicos sobre el sistema.
- La optimización del sistema de operaciones traído por una configuración menos conservadora y más óptima de parámetros operativos debido al mejor conocimiento del sistema y su estado que ofrecen los modelos numéricos del DT.

Hacer uso de estos modelos permite comparar situaciones reales, situaciones teóricas y situaciones simuladas. Monitorizar el estado real de, por ejemplo, la temperatura de salida del tanque de agua corriente sanitaria y comparándolo con el modelo teórico permite encontrar desviaciones en su rendimiento en tiempo real e incluso predecir valores futuros.

Gracias a esta solución inteligente, se tiene a disposición un funcionamiento adaptado a la realidad del usuario, dejando atrás la limitación que suponen las configuraciones por defecto, así como la virtud de un sistema autoregulable y constantemente comunicado con su entorno.

6.2. Análisis de requisitos

Requisito 0

El canal de comunicación entre el gemelo físico y digital debe soportar el flujo bidireccional de información.

Requisito 1

El gemelo digital muestra en tiempo real la actividad registrada por los sensores.

Requisito 2

El gemelo digital informará en tiempo real al usuario si existe un comportamiento no deseado del sistema.

Requisito 3

El gemelo digital informará en tiempo real al usuario si existe un cambio de estado esperado del sistema.

Requisito 4

El gemelo digital registra las modificaciones en el gemelo físico llevadas a cabo por el usuario.

Requisito 5

El gemelo digital está preparado para almacenar grandes cantidades de datos heterogéneos.

Requisito 6

El gemelo digital debe predecir el comportamiento del gemelo físico a partir del histórico de actividad de los sensores del gemelo físico.

Requisito 7

El gemelo digital se apoyará en sistemas ajenos al entorno energético para mejorar el conocimiento presente y futuro del entorno energético.

Requisito 8

Se debe mantener un control de las versiones, cambios y correcciones del producto durante la implementación de la solución.

Requisito 9

La estructura de la solución diseñada debe permitir una detección ágil de los errores.

Requisito 10

El producto debe ser fácil de supervisar y mantener.

Requisito 11

La planificación del proyecto debe de ser tolerante a cambios y modificaciones sobre el diseño de la solución.

7. Metodología

En este capítulo se describen el conjunto de métodos y técnicas de gestión del proyecto que se aplicarán para llevar a cabo el desarrollo de la solución. Trabajar con una metodología de desarrollo permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar [68].

7.1. Gestión del proyecto

Para poder gestionar el desarrollo de soluciones complejas, es habitual hacer uso de herramientas que permitan planificar el trabajo a realizar y distribuir en una ventana temporal las tareas dedicadas a construir progresivamente cada los módulos en los que se trabaja. En el presente proyecto, se opta por una metodología de desarrollo ágil basada en la metodología incremental. Este enfoque permite construir el gemelo digital a partir de porciones mínimas y de manera progresiva, añadiendo nuevas funcionalidades cada poco tiempo a la solución final. Por otro lado, este flujo de trabajo es tolerante a la aparición de nuevos requisitos del proyecto así como a incidencias que supongan reprogramar el calendario.

Entre un abanico de distintas tecnologías, para gestionar el desarrollo del gemelo digital se hará uso la aplicación de administración de tareas Microsoft Planner. Este software permitirá al desarrollador planificar las diferentes tareas a realizar para encaminar el proyecto, definir una serie de momentos en el tiempo para establecer fechas límites en el desarrollo, así como la posibilidad de reorganizar la planificación conforme surgen incidencias y se realizan cambios en el diseño.

Y es que los incidentes y los cambios de sentido en el proyecto son un factor fundamental a tener en cuenta para elaborar soluciones complejas, es por ello que una solución que permite tener esa flexibilidad y tolerancia a los cambios como la de Microsoft se ajustará al plan de desarrollo incremental que se ha adoptado en este proyecto.

7.2. Control de versiones

En un metodología de desarrollo incremental, se añaden progresivamente nuevas funcionalidades al producto final. Es aconsejable hacer uso de herramientas que permitan tener un control sobre las nuevas versiones de la solución final tras la introducción de nuevos elementos, preservar el histórico de los cambios y tener la posibilidad de revertir las acciones realizadas en caso de cambios en los requerimientos.

Los repositorios son sistemas de información que preservan y organizan el material a la vez de garantizar el acceso a esta información [69]. A la hora de implementar una solución en código, será necesario hacer uso de un sistema que permita dejar constancia del historial de cambios que se hayan realizado sobre el programa así como la

posibilidad de revertirlos. Otra de las ventajas de utilizar repositorios públicos como GitHub, es que se garantizará el acceso al material del proyecto desde cualquier dispositivo con acceso a la red. Un tercer factor a tener en cuenta es la seguridad. En desarrollos largos como el caso de este trabajo, existirá el riesgo de sufrir algún tipo de pérdida de información en la estación utilizada o, simplemente, que esta deje de funcionar [\[70\]](#).

Para llevar a cabo el desarrollo de esta solución, en este proyecto deberá volcarse la información del workspace sobre el que se trabaje en los repositorios públicos de GitHub.

8. Diseño

En este capítulo se definirá el diseño escogido para la solución al problema: desarrollar un gemelo digital que refleje, prediga y optimice el funcionamiento de una bomba de calor aire-agua instalada en un entorno doméstico. A diferencia de los modelos digitales, los gemelos digitales deben ser diseñados con vistas a poder garantizar una comunicación entre la unidad de control del gemelo digital y la unidad de control del sistema físico, además de la comunicación entre la red sensores y el lugar de almacenamiento de los datos que se analizan.

Puesto que un *digital twin* se compone de varios elementos donde cada uno contribuye parcialmente en la solución, se ha dividido este problema complejo en problemas más asequibles. Esta enfoque modular permite afrontar el reto de integrar los diversos campos de la informática que acogen los *Digital Twins*.

En la figura 8-A se puede observar la estructura que se ha elegido para diseñar el gemelo digital, la cual se compone de varios módulos.

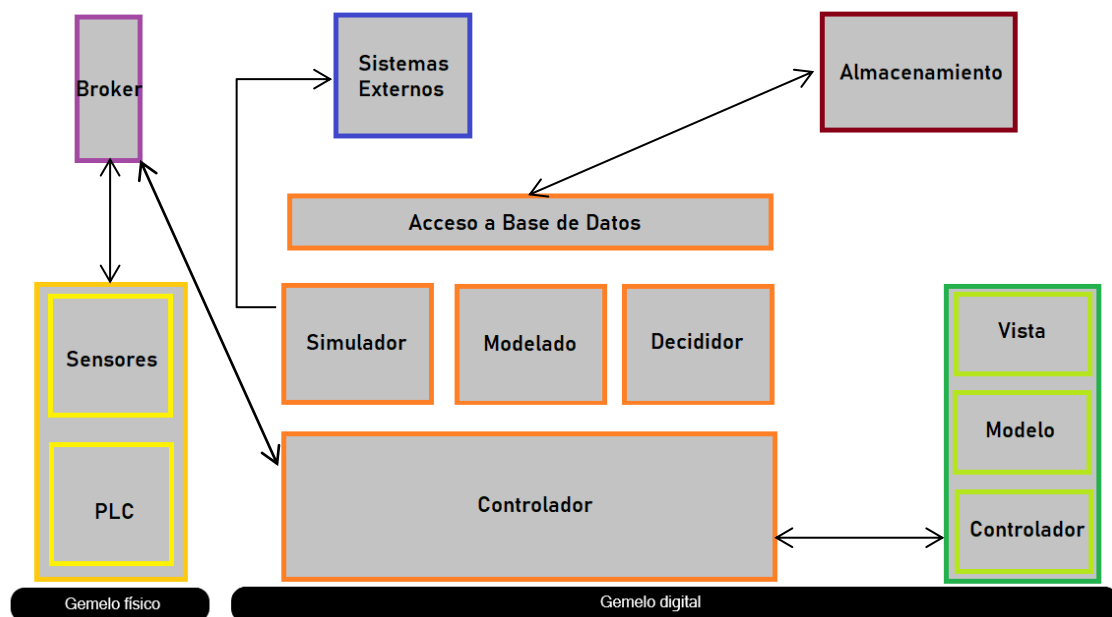


Fig. 8-A. Esquema modular del gemelo digital

En la Fig. 8-A se representan los diferentes sistemas que compondrán la estructura de la pareja digital y física. Se observa que esta estructura está dividida en dos grandes bloques, el gemelo físico y el gemelo digital. Estos dos módulos están unidos, por un lado, por la parte de almacenamiento, mediante la que el gemelo físico comunica al gemelo digital los datos recogidos por los sensores y, por otro lado, la parte de control, que permite al gemelo digital comandar el funcionamiento de su homólogo físico.

Se describe separadamente las función de cada uno de los bloques:

- El **gemelo físico** será el sistema real al cual le corresponderá su clon digital. En el entorno de este trabajo, es el sistema de bomba de calor instalado en el hogar y cuyas características se han detallado en la Sección 4. Este módulo constará de un sistema que registra determinadas mediciones en su funcionamiento y el estado de su entorno por medio de una red de **sensores**. Por otro lado, contará con un sistema de control de la configuración de la maquinaria orquestrado por un usuario a través de una **PLC** tanto presencialmente como de manera remota.
- El **gemelo digital** será la representación virtual creada a imagen y semejanza del gemelo físico. Su labor consistirá en establecer una comunicación en tiempo real con el gemelo físico y extraer información acerca del comportamiento real y futuro del sistema con el fin de monitorizar y mejorar su rendimiento y disponibilidad. También presentará una amigable interfaz para permitir la interacción directa entre usuario y sistema.

En los siguientes subapartados se entra en detalle respecto a las características de los submódulos que forman el PT y el DT y la comunicación que existe entre ellos.

8.1. Gemelo físico

En esta sección se enumeran y describen los componentes que conformarán la estructura del gemelo físico, este bloque representa la figura del sistema real instalado en el entorno doméstico del proyecto. En la figura 8-A se distinguen dos módulos diferentes:

- **Sensores.** A nivel conceptual, se trata de una red mallada de sensores que envía datos preprocesados al módulo de almacenamiento de datos ubicado en el DT. En la práctica de este proyecto, dado el entorno de trabajo y la consecuente imposibilidad de obtener en tiempo real el valor actual de los sensores, se desplegará una solución que simule el envío periódico de documentos al ente virtual a partir de un histórico de datos previamente guardado. Este procedimiento se realizará a través de un script en código *Python*, que recogerá y transmitirá periódicamente los registros del histórico de datos, almacenados de manera local.
- **Controladora.** Este actuador se encargará de atribuir nuevas configuraciones a la máquina según las indicaciones del sistema de decisión al que se encuentra directamente conectado. Se puede entender este módulo como un conversor de los comandos que llegan del DT en acciones reales de la maquinaria.

8.2. Gemelo digital

El gemelo digital es un modelo matemático que simula el funcionamiento del gemelo físico. Este reflejará en tiempo real la situación registrada por su homólogo físico. Además su carácter virtual le permite simular escenarios que todavía no se hayan

contemplado y con ello poder adaptar el comportamiento real del sistema de manera autónoma a partir de estas especificaciones. De acuerdo con el esquema 8-A los módulos en los que se ha dividido este bloque son:

8.2.1. Servidor

El servidor será la unidad de procesamiento y análisis de los datos que analizados y procesados por el gemelo. El despliegue de este servidor se secciona en diversas capas:

- **Capa de control:**

- **Controlador.** El controlador tendrá la función de inicializar los principales servicios del gemelo digital: los sistemas de interacción tanto con el usuario final así como con el sistema físico y los módulos de la capa de servicio, dedicados a realizar los cálculos del comportamiento del sistema real y simulado.

Se implementará mediante un servidor *flask* [71] basado en *Python*, el cual ofrecerá un listado de rutas a las que el cliente podrá apuntar. Este servidor inicializará también un cliente *mqtt* mediante la librería *paho mqtt* [72], mediante el cual se establecerá una sesión con el *broker* para permitir interactuar con el sistema físico.

- **Capa de servicio:**

- **Modelado.** Se tratará de una representación numérica del sistema que permite describir su naturaleza y estado con exactitud a partir de mediciones e información conocida de este y su entorno. Los modelos generados son la base de los cálculos del elemento simulador. Este módulo tendrá como inputs el histórico de todos los registros generados por el gemelo físico y podrá ofrecer al simulador todo un abanico de modelos constantemente actualizados a partir de los registros más recientes del gemelo físico.

Este módulo correrá en segundo plano para mantener el control en la ejecución del servidor y se implementará mediante la librería de código abierto de análisis y manipulación de datos *pandas* [73].

- **Simulador.** Será el servicio encargado de reproducir condiciones del sistema que todavía no se hayan experimentado a partir de la línea del comportamiento observado. Este módulo será clave para que el gemelo digital se aleje de la proactividad y pueda tomar medidas preventivas que impulsen mejoras sobre su rendimiento.

Estas operaciones se realizarán a través de los modelos generados por el módulo de “modelado” a través de la mismas funciones de predicción que ofrece la librería *pandas*.

El código fuente de ambos módulos (Modelado y Simulador) utilizará tecnologías que permitan el análisis de datos y creación de modelos de *Machine Learning*. Estas herramientas deberán permitir:

- **Modelar** → Crear modelos numéricos basados en el histórico de datos recogidos por el gemelo físico hasta la fecha, entrenarlos y mantenerlos actualizados conforme a los nuevos datos que vaya recibiendo el gemelo digital.
- **Simular** → Predecir la respuesta del sistema ante determinados inputs que puedan registrarse en un futuro haciendo uso del abanico de modelos numéricos que ofrezca el módulo de entrenamiento.
- **Decididor**. Este módulo permitirá calcular el comportamiento real del sistema a partir de un enumerado de reglas estáticas que definen el correcto funcionamiento del gemelo físico y provocar que el gemelo digital actúe en consecuencia.
- **Capa de acceso a datos**. Se tratará de la capa de acceso a base de datos, encargada de ofrecer a los servicios de análisis de datos del servidor un medio de conexión con el sistema de almacenamiento. Esta interfaz será común para todos los módulos que operen en el lado del servidor y quieran acceder al histórico.
- **Unidad de almacenamiento**. Servicio de base de datos que almacenará todos los registros recogida de la red de sensores instalada en el gemelo físico. Se tratará de un sistema que pueda almacenar grandes cantidades de datos y que no depende de la tipología de estos, por lo tanto, una base de datos no relacional como MongoDB será una tecnología interesante a la hora de desplegar este servicio.

En el contexto de las arquitecturas del tipo Cliente-Servidor, desarrollar el lado servidor en *Python* pondrá a disposición del desarrollador un abanico de librerías públicas dedicadas a la manipulación y el análisis de datos, así como la posibilidad de trabajar con numerosos protocolos de comunicación.

8.2.2. Cliente

El cliente será el encargado de mostrar visualmente al usuario cuál es la realidad del problema e ofrece una interfaz gráfica amigable para que este pueda visualizar la información e interactuar con el sistema. Se optará por una arquitectura modular del tipo Modelo-Vista-Controlador (MVC) en cliente. Esta formación permitirá dividir de manera modular el almacenamiento estructurado de los datos en local, la lógica de control y la interfaz- vista con la que interactúe en el usuario.

- **Modelo.** Permitirá almacenar localmente de manera estructurada la información recibida desde el servidor del gemelo digital. Enviará a la vista a través del controlador la información que se solicita ser mostrada.
- **Vista.** Se tratará de la interfaz gráfica con la que interactúa directamente el usuario. Permitirá al usuario hacer un seguimiento de la información que fluye entre los gemelos e interactuar con ella.
- **Controlador.** Será la capa que responderá a los eventos que invoquen las acciones del usuario sobre la vista. Su función consistirá en dinamizar la interfaz gráfica que se le presenta al usuario y, por otro lado, inicializar el enlace de comunicación entre el cliente y el servidor.

Se trabajará bajo un marco de desarrollo web, en el que lenguajes como *javascript* y *html* permitirán dinamizar la información mostrada por pantalla al usuario así como integrar el módulo cliente con el módulo servidor descrito en el punto 8.2.1.

8.2.3. Componentes externos

Existirán otros componentes fundamentales para desarrollar un gemelo digital que operan al margen de los módulos cliente y servidor. En la figura 8-A se distinguen los siguientes:

- **Brokers.** Los *brokers* serán servicios que trabajarán como intermediarios en una comunicación entre dos partes. Permiten dotar de persistencia y bidireccionalidad a una conexión y son comúnmente utilizados cuando se trabaja en entornos que requieren intercambios de información en tiempo real. Para los gemelos digitales, este mediador se desplegará entre los gemelos físico y digital para establecer esta comunicación.
- **Sistemas externos.** Comunicarse a sistemas externos que almacenen información de diversa tipología ofrecerá la posibilidad de añadir información al contexto del sistema que se encuentren fuera del alcance local y no se haya contemplado a partir de los registros captados por la unidad física.

En este capítulo se han detallado las características de los módulos que formarán parte de la solución así como las tecnologías utilizadas. A modo resumen, la tabla 8-A muestra como hay un cubrimiento total de los requerimientos del proyecto mediante las tecnologías utilizadas para desarrollar la solución:

Núm. requisito	Requisito	Tecnología / metodología
0	El canal de comunicación entre el gemelo físico y digital debe soportar el flujo bidireccional de información.	Servidor Mosquitto
1	El gemelo digital muestra en tiempo real la actividad registrada por los sensores.	Servidor Mosquitto
2	El gemelo digital informará al usuario si existe un comportamiento no deseado del sistema.	Flask Server (Decididor)
3	El gemelo digital informará al usuario si existe un cambio de estado esperado del sistema.	Flask Server (Decididor)
4	El gemelo digital registra las modificaciones en el gemelo físico llevadas a cabo por el usuario.	HTML-Javascript-CSS (Cliente)
5	El gemelo digital está preparado para almacenar grandes cantidades de datos heterogéneos.	MongoDB (Almacenamiento)
6	El gemelo digital debe predecir el comportamiento del gemelo físico a partir del histórico de actividad de los sensores del gemelo físico.	Pandas - Linear Regression
7	El gemelo digital se apoyará en sistemas ajenos al entorno energético para mejorar el conocimiento presente y futuro del entorno energético.	Pandas – Linear Regression (Modelado, Simulador)
8	Se debe mantener un control de las versiones, cambios y correcciones del producto durante la implementación de la solución.	GitHub

9	La estructura de la solución diseñada debe permitir una detección ágil de los errores.	Diseño modular
10	El producto debe ser fácil de supervisar y mantener.	Diseño modular
11	La planificación del proyecto debe de ser tolerante a cambios y modificaciones sobre el diseño de la solución.	Desarrollo incremental

Tabla 8-A. Cubrimiento de los requerimientos

En el siguiente capítulo se describe cómo se ha llevado a cabo la implementación del gemelo digital digital que responde a los requisitos del proyecto. Esta solución estará basada en la estructura presentada en este capítulo e implementada mediante las tecnologías escogidas.

9. Implementación

- **Recolección de los datos e interacción bidireccional.** Para cumplir esta

- 



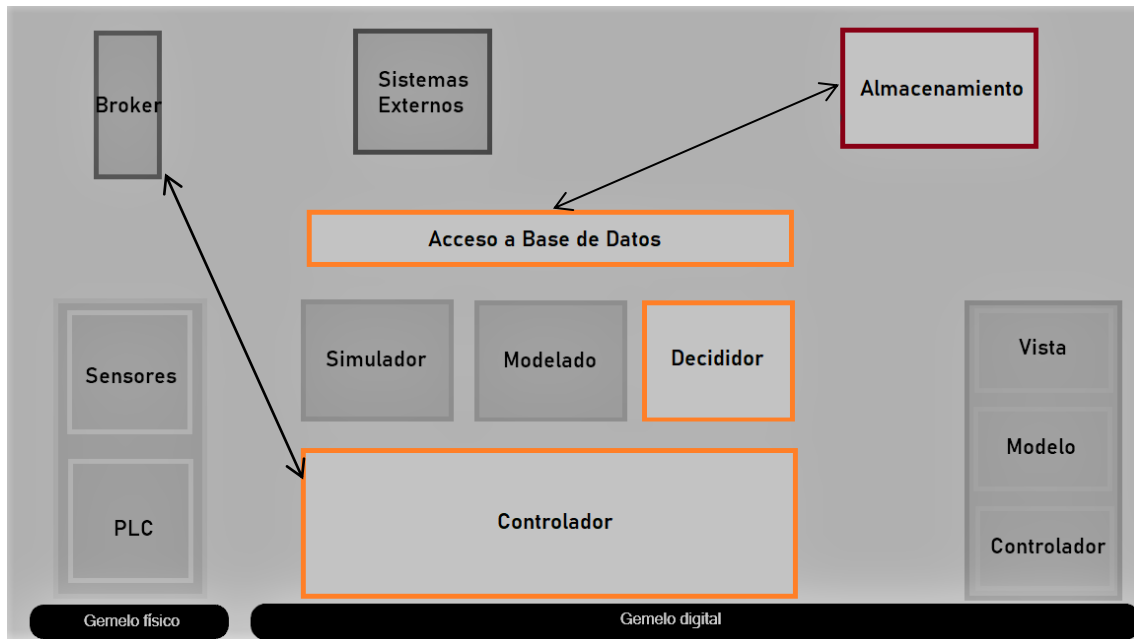


Fig. 9-B. Componentes del gemelo digital que participan en la descripción del estado del sistema real.

- Cálculo del comportamiento del sistema virtual.** Las tecnologías de simulación de escenarios basadas en el *Machine Learning* permiten al gemelo digital anticiparse a futuros sucesos sobre el sistema físico. En el servidor, se han implementado los componentes de la capa de servicio “modelado” y “simulador” para poder realizar predicciones del comportamiento de la máquina ante futuros escenarios.

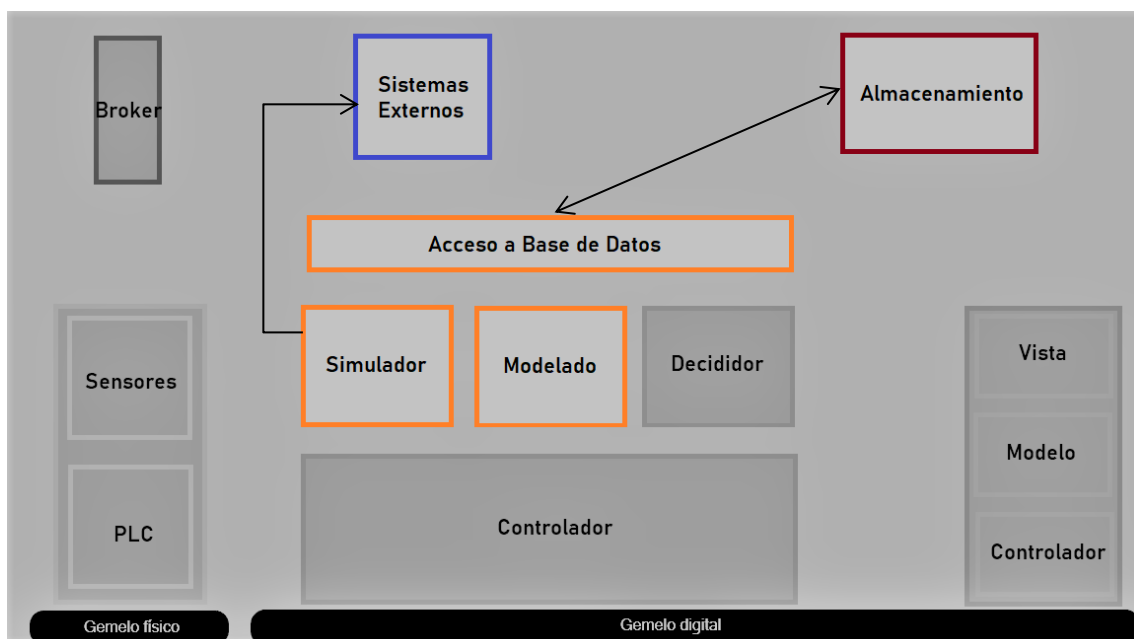


Fig. 9-C. Componentes del gemelo digital que conforman el sistema de cálculo de comportamiento basado en simulación.

- **Desarrollo de la interfaz gráfica de usuario.** A partir de la información analizada por el gemelo digital, el usuario debe disponer visualmente de toda la información relevante que gemelo digital pueda ofrecerle de su homólogo física para poder realizar un seguimiento de su funcionamiento. Se desarrolla un entorno basado en API REST que permite comunicar el cliente con el servidor con el fin de mostrar la información a través del navegador.

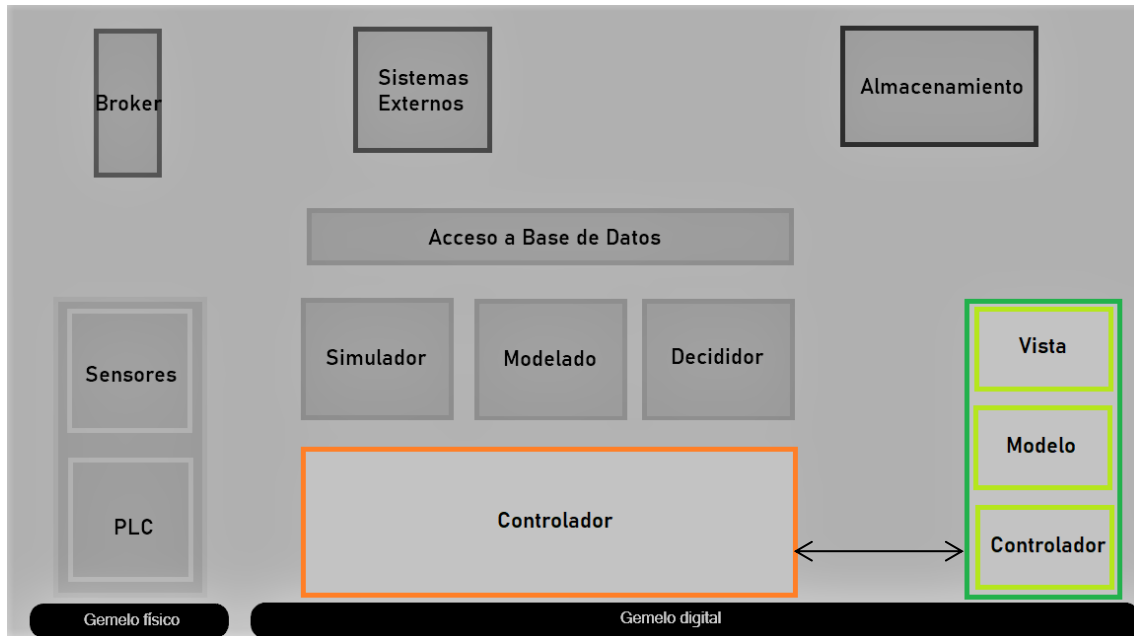


Fig. 9-D. Componentes del gemelo digital que forman el sistema de interacción con el usuario y visualización.

Entre diferentes entornos de desarrollo, se escoge el IDE *Pycharm* para redactar el código fuente del gemelo digital. Este software está enfocado a trabajar con soluciones basadas en código Python, a la vez que considera su integración en marcos de trabajo de desarrollo de código web. Otro aspecto a considerar sobre *Pycharm* es que dinamiza el flujo de trabajo, permitiendo saltar ágilmente entre los procedimientos y las clases del código fuente.

9.1. Gemelo físico

El gemelo físico es la unidad física y real de la pareja de gemelos. En este proyecto, se ha implementado la solución sobre el sistema de bomba de calor Panasonic instalado en el mismo campus universitario y mostrado a través de la figura 1-A, cuyo esquema interno ha sido mostrado posteriormente mediante la figura 5-A. Estos dos bloques físico y digital están estar intercomunicados y sincronizados así como muestra la figura 8-A.

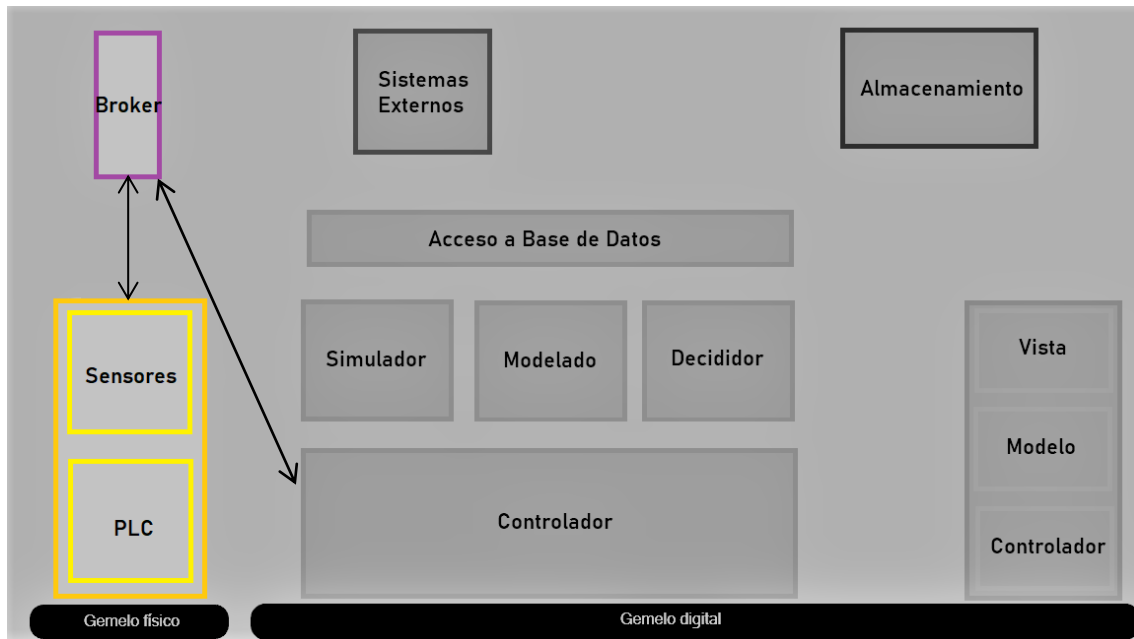


Fig. 9.1-A – Componentes que forman el gemelo físico

Para que el gemelo digital pueda extraer información relevante de la bomba de calor que se estudia, se implementa un mecanismo que recoge mediciones en tiempo real tanto de la bomba como del ambiente en el que trabaja

Por otro lado, a partir de los resultados, deducciones y predicciones que pueda generar el ente virtual, la bomba de calor tiene la capacidad de ajustar su configuración según las indicaciones del *digital twin*.

El gemelo físico pues, es la fuente del input del gemelo digital y se beneficia de la información de output que este genere.

9.1.1. Recolección de los datos

Los próximos párrafos se dedican a detallar cómo se ha desarrollado el componente que hace referencia a la recolección de los datos extraídos del gemelo físico y su transmisión al gemelo digital. Se denomina como el módulo de sensores en la figura 9.1-A.

Dado que no se cuenta con acceso directo al sistema de bomba de calor, se decide simular las transmisiones llevadas a cabo por la red de sensores con la que cuenta. En el apartado 5.1 se mencionan que la generación de datos debe realizarse en diferido, esto es, mediante la descarga de un histórico de datos delimitado entre fechas a partir del entorno de Grafana que proporciona la Universidad:

<https://smartgraph.uib.es/d/9CpqX5MMk/ca-ses-llucies?orgId=1&refresh=15m>

Una vez obtenido el conjunto de datos, se crea un script en código Python que simula el envío continuo de información cada cuarenta y cinco segundos. Se hace uso de la librería Pandas para poder manipular los datos y enviar los documentos en el formato correcto.

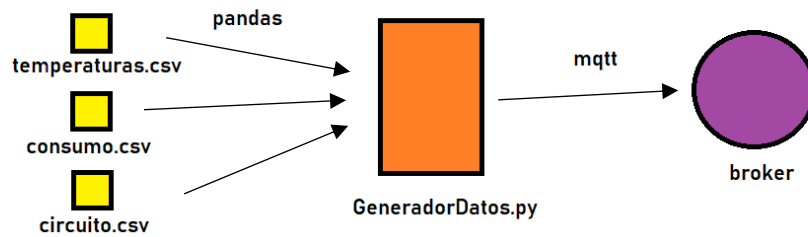


Fig. 9.1.1-A. Simulación de envío del registro de actividad en el gemelo físico.

En el paradigma de los *digital twins*, el trasvase de información en tiempo real es un factor primordial. Es por ello que debe desplegarse un sistema de comunicación persistente que permita enviar información al mundo digital tan pronto como el sistema real lo desee.

Se despliega un servidor Mosquitto que trabaje de intermediario entre ambos gemelos y que opera con sus cliente mediante el protocolo *mqtt*. Para que el gemelo físico pueda enviar los datos que se recogen, se abre un cliente *mqtt* desde el mismo script de generación de datos haciendo uso de la librería de código abierto *paho.mqtt*.

Este punto permite describir cómo se soluciona el problema de la recolección de los datos y su transmisión al mundo virtual. El código fuente en referencia a este sistema de generación de datos en diferido se documenta en el Anexo I. En el siguiente punto se hace referencia a la comunicación en sentido opuesto y retroalimentación de la máquina.

9.1.2. Sistema de control del entorno físico

En el anterior apartado se detallan las herramientas de la comunicación entre los gemelos físico y digital que se ha tenido en cuenta al implementar la solución a este problema y cómo se ha implementado el módulo funcional referente a la actividad de los sensores. Esto ha sido mediante el envío simulado y periódico de un histórico predelimitado de los registros que son visibles y descargables por el usuario desde el portal corporativo.

Con este sistema se ha implementando parcialmente la actividad del gemelo físico, pues el paradigma de los gemelos digitales contempla la bidireccionalidad en el flujo de los datos entre ambos entes. Sin embargo, este segundo módulo en referencia a la actividad del controlador PLC, recipiente de los resultados de los cálculos realizados desde el gemelo digital, no se ha implementado debido a las limitaciones que se contempla el entorno de trabajo. Sin embargo, la solución implementada no disminuye la efectividad del proyecto, pues los beneficios del uso de gemelos digitales en cuestiones de eficiencia y rendimiento de la máquina pueden visualizarse sin manipular directamente el sistema PLC.

El diseño llevado a cabo para implementar esta solución contempla de manera teórica la implementación de un módulo incorporado en el gemelo físico que realice la labor directa de control de la maquinaria. Puesto que se ha implementado un servidor *broker* que soporte las transmisiones de datos en ambos sentidos de manera constante, se facilita la integración de este módulo PLC en futuros trabajos.

9.2. Gemelo digital

En este apartado se muestra la manera en la que se ha implementado el sistema digital que simula el comportamiento de su homólogo físico. El objetivo de esta unidad es monitorizar y mejorar el rendimiento a través de la representación del comportamiento del sistema real y la predicción del comportamiento futuro del sistema, respectivamente.

Se debe trabajar de manera bidireccional e ininterrumpida con el gemelo físico, alimentándose y analizando de la información que este le pueda proporcionar con el fin de mejorar la transparencia, su rendimiento y eficiencia.

Haciendo énfasis en estos dos últimos factores, rendimiento y eficacia, se despliega una solución que permite una comunicación directa con el mundo real y, dada la imagen, interpreta el estado en el que el sistema se encuentre en la actualidad y cómo operaría en nuevos escenarios. Por otro lado, se ha desarrollado un entorno amigable desde el que el cliente pueda realizar un seguimiento del estado su maquinaria así como reajustar la configuración de manera manual.

9.2.1. Módulo de comportamiento del sistema - Servidor flask

Con este componente, que forma parte del gemelo digital, se representa el comportamiento de la máquina en un momento concreto y ofrecer tolerancia ante eventos futuros. El módulo de comportamiento del sistema tiene la misión de operar con los datos que provengan del gemelo físico y convertirlos en información útil tanto para el usuario final como para la misma máquina.

En Fig 9.2.1-A se observan los componentes diseñados que forman parte del servidor y que se dedican a analizar el comportamiento del sistema en tiempo real así como predecir futuros escenarios a partir de la actividad previa y de fuentes externas.

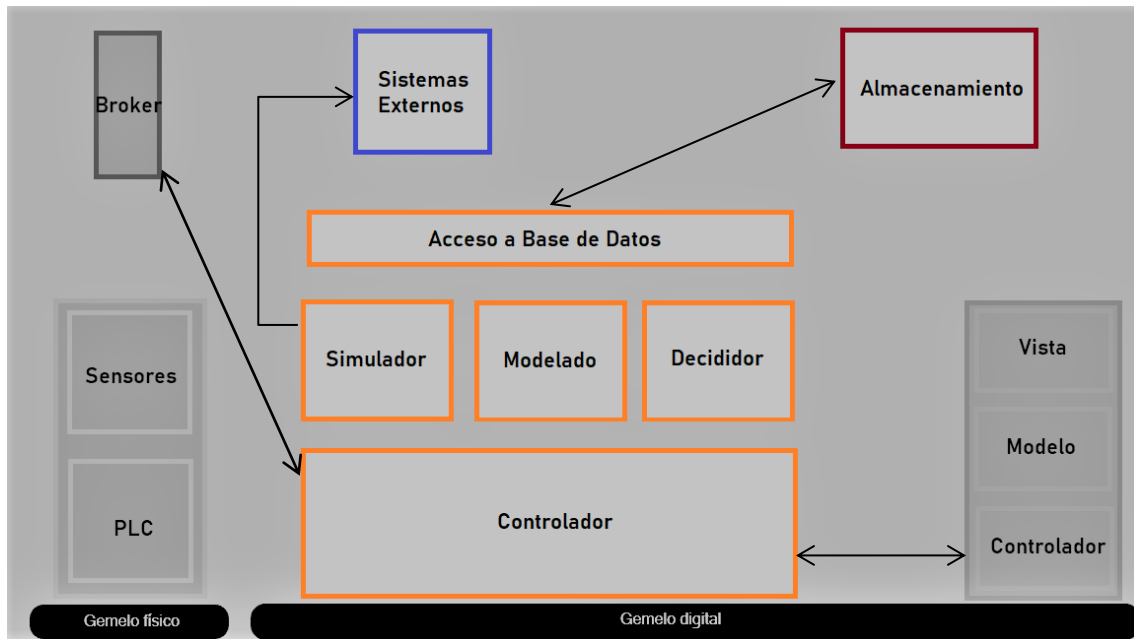


Fig.9.2.1-A. Componentes que forman el servidor

La información debe fluir de manera bidireccional entre el *digital twin* y el *physical twin*. Tal y como se ha implementado en el gemelo físico, en este entorno también es necesario dedicar ciertos recursos a mantener la visibilidad entre los dos gemelos de manera persistente.

Por otro lado, desde el punto de vista del usuario final, se debe determinar una fuente desde la que el usuario pueda descargar en su dispositivo la información referente al estado del gemelo digital. Además, desde el mismo nivel, el usuario también podría sentir la necesidad de establecer nuevas configuraciones de la bomba.

En la figura 9.2.1-B se visualiza a bajo nivel los diferentes servicios que ofrece el servidor desplegado:

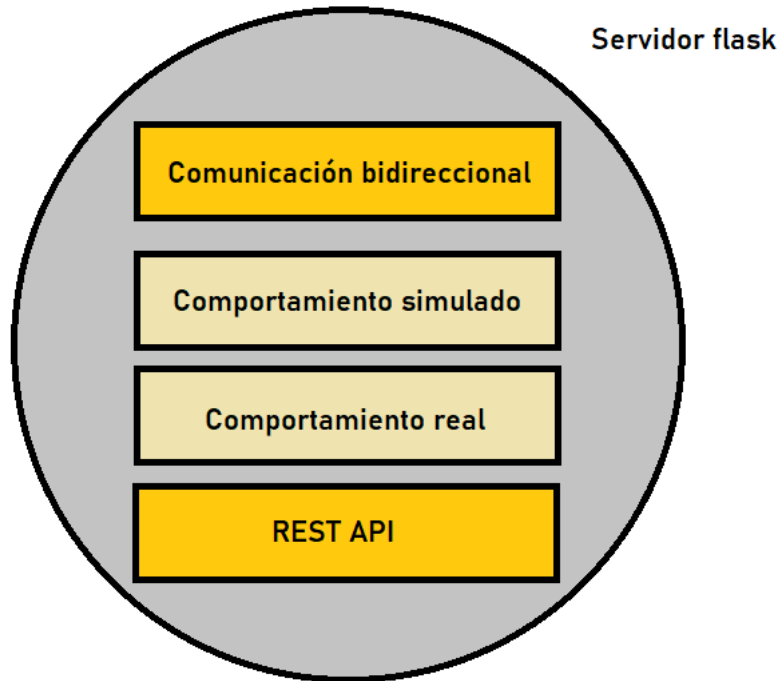


Fig. 9.2.1-B. Módulos funcionales del servidor flask

Para poder aglutinar todas estas funcionalidades introducidas en la Figura 9.2.1-B en una única unidad se ha desplegado en local una app en Python basada en *flask* [71] que permite ejecutar paralelamente varios módulos funcionales:

- El módulo dedicado a establecer una comunicación bidireccional y constante con el gemelo físico.
- La unidad de cálculo del comportamiento previsto del sistema a partir de simulaciones basadas en modelos numéricos. Estos modelos describen el funcionamiento del sistema físico a partir de los datos recogidos por los sensores.
- El módulo encargado de describir el estado del sistema en tiempo real a partir de las reglas estáticas de los sistemas de bomba de calor.
- Módulo de interacción con el usuario final vía REST API. Que ofrece una tabla de rutas a las que el cliente apunta para solicitar información del sistema.

A continuación, se resumirá la naturaleza de cada uno de estos módulos y se justifica la relación de sus funcionalidades con los objetivos del servidor.

Comunicación bidireccional

En esta sección se detalla cómo se ha implementado el servicio encargado de establecer la comunicación con el sistema físico. Tal y como muestra la figura 9.2.1-B, la comunicación bidireccional será un proceso, entre otros, que lleva a cabo el servidor *flask* desplegado. Sin implementar esta comunicación en ambos sentidos, el sistema físico no puede recibir el feedback tras los cálculos de su comportamiento que realiza el gemelo digital y las instrucciones del mismo usuario.

Para establecer esta comunicación, tal y como se muestra en la figura 9.2.1-A se ha utilizado un *broker* basado en *Mosquitto* que permita levantar una conexión *mqtt* entre el gemelo físico y digital. Puesto que en figura 9.2.1-A enseña cómo se hace uso de este *broker* para que el gemelo físico envíe los datos recogidos por la red de sensores, en la siguiente imagen se muestra cómo se adapta el gemelo digital a este esquema:

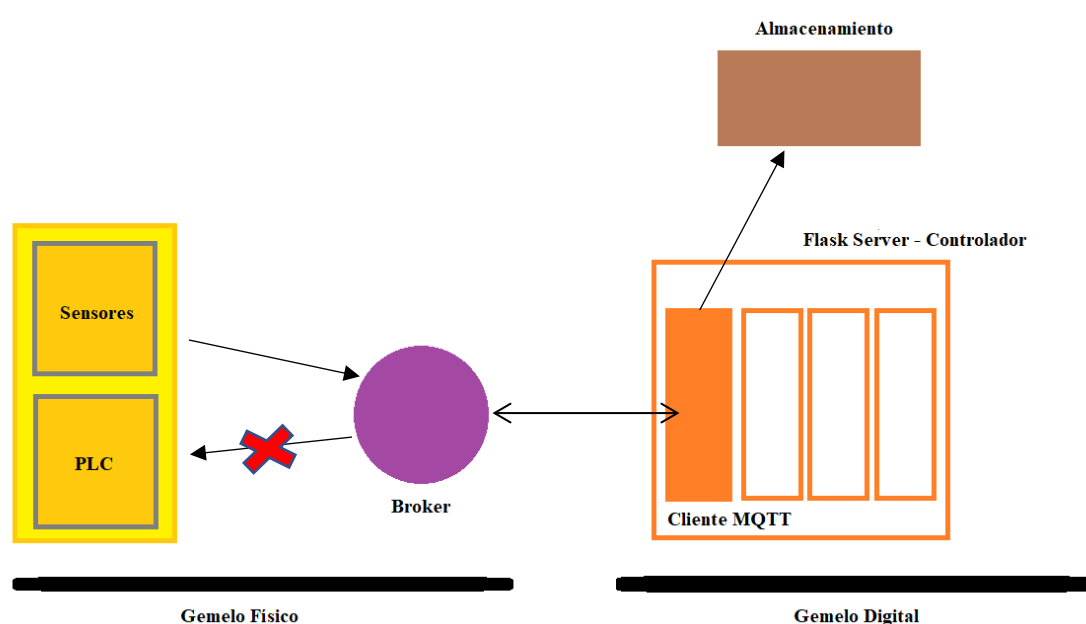


Fig. 9.2.1-C. Esquema de comunicación bidireccional entre gemelos

En la Figura 9.2.1-B se detalla cómo se ha implementado la funcionalidad de interacción con el sistema desde el lado servidor del gemelo digital. Se inicializa un cliente MQTT mediante el cual se permite abrir una sesión de comunicación entre gemelos físico y digital. El desarrollo de este componente de interacción con el gemelo mediante una comunicación bidireccional permitirá llevar a cabo la primera de las funciones fundamentales del gemelo digital mostradas en la figura 9.2.1-B.

- **Comunicación de Gemelo Físico a Digital.** Una vez realizado el procedimiento establecido en la figura 9.1.1-A, los datos son enviados a través del canal *mqtt* abierto entre el *broker* y el Cliente MQTT desde la capa de control del servidor. Posteriormente, estas cantidades ingentes de datos que recibe el DT son almacenadas en el módulo de almacenamiento MongoDB, visto anteriormente en la Figura 9.2.1-A

- **Comunicación de Gemelo Digital a Físico.** A causa de las limitaciones del entorno energético con el que se trabaja en el proyecto, el servidor no implementa ninguna funcionalidad dedicada al control de la PLC de la bomba de calor dado que no se cuenta con acceso a este entorno. Sin embargo, en esta implementación sí se ha considerado a nivel conceptual mediante el desarrollo de una comunicación *mqtt*, que soporte la comunicación en ambos sentidos.

El código fuente en referencia al módulo de interacción entre gemelos físico y digital en el lado de servidor forma parte del programa principal de la aplicación *flask*. Este puede verse en el Anexo II – Controller.py.

Descripción del estado del sistema real

En el esquema de la figura 8-A se visualizan, entre otros, los módulos pertenecientes a la capa de servicio del servidor: el simulador, el módulo de modelado y el decididor. En el punto 8.2.1 se ha anticipado que los módulos de simulación y modelado realizan las tareas de cálculo del comportamiento del sistema virtual con el fin de que el sistema pueda anticiparse a escenarios futuros. El módulo decididor, sin embargo, es el encargado de evaluar el comportamiento real del sistema y atiende a las condiciones presentes en las que se encuentra el gemelo físico.

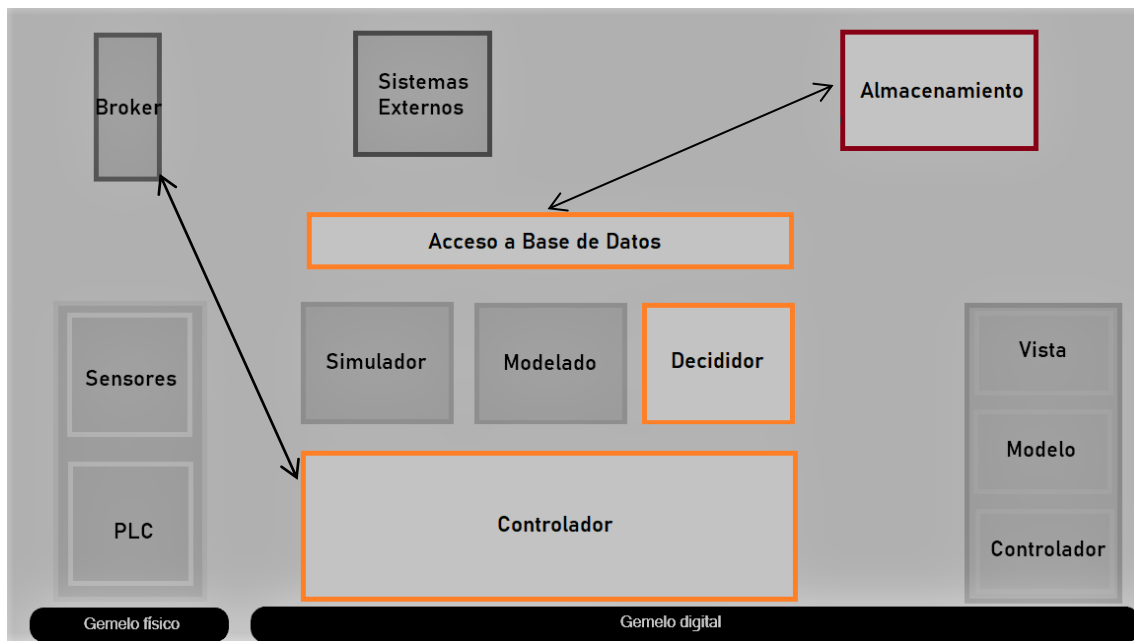


Fig. 9.2.1-D – El decididor es el componente que analiza el comportamiento real del sistema.

En estos entornos domésticos, los gemelos digitales son una solución de valor porque permiten al usuario conocer fallas en el funcionamiento presente y futuro de la instalación. Concretamente, la función principal del módulo que describe el estado de sistema será asegurar el correcto funcionamiento de este en tiempo real. Para poder

notificar situaciones indeseadas que se deban considerar, se parte de los datos obtenidos durante el proceso de recolección realizado por los sensores y transmitido al gemelo digital.

En la figura 8-A se visualiza cómo el gemelo físico hace uso de un intermediario o *broker* para comunicarse en tiempo real con el gemelo digital. También, se distingue un módulo de almacenamiento donde el gemelo digital deposita los datos que recibe de la red de sensores. Obteniendo los registros más recientes de la bomba, se determina mediante una serie de reglas estáticas propias de un sistema de calefacción si existen condiciones numéricas que podrían suponer que la disponibilidad y el buen rendimiento de sistema se estuviera viendo comprometido.

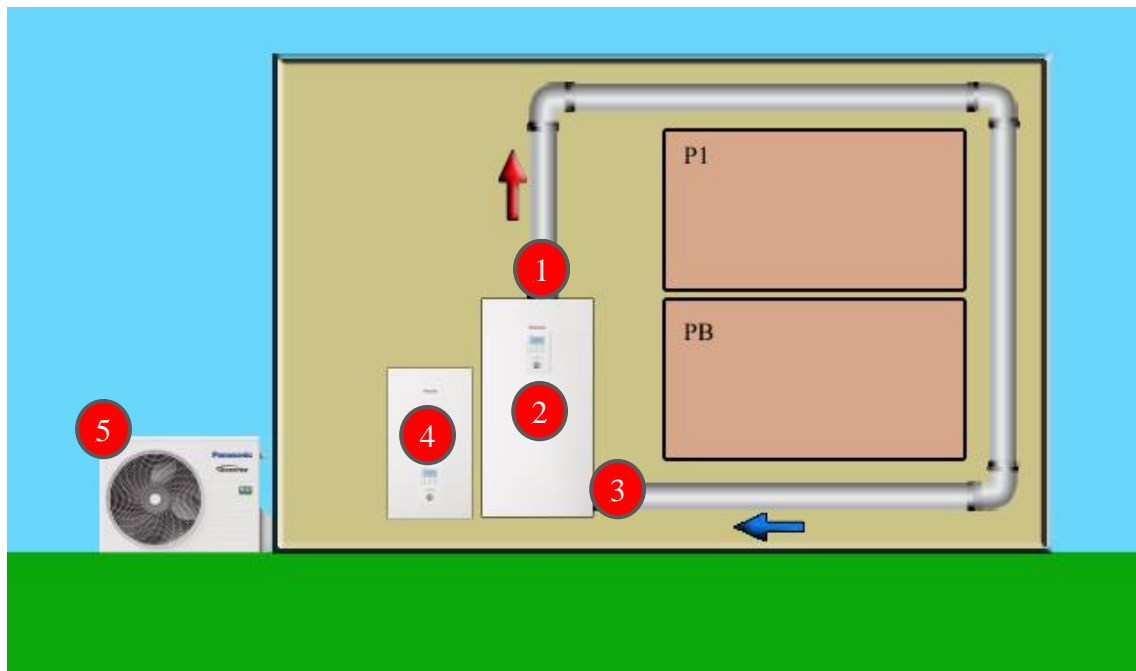


Fig. 9.2.1-E. Puntos de fallo de la bomba de calor

En la Fig 9.2.1-E, se reflejan los puntos críticos del sistema HVAC que podrían registrar condiciones indeseadas:

- 1- Temperatura de impulso del agua
- 2- Estado del tanque de agua (temperatura)
- 3- Temperatura de retorno del agua
- 4- Tensión de la bomba
- 5- Temperatura del exterior

Tan pronto como las mediciones de la red sensores se comunican al gemelo digital, el servidor debe procesarlos en tiempo real para determinar si estas cumplen con las condiciones ideales o bien existen algún punto de fallo. El estado ideal puede

determinarse tanto por las reglas estáticas propias de un sistema de bomba de calor genérico en modo calor como por rasgos y recomendaciones concretadas detalladas por el fabricante. Las condiciones a las que atiende esta solución son las siguientes:

- La temperatura de impulso del agua desde el tanque no debe superar la temperatura consignada por el usuario. Incumplir esta condición supondría tener un sobrecalentamiento del tanque.

$$T_{\text{impulso}} \leq T_{\text{consignada}}$$

- En un sistema de calefacción la temperatura impulsada por el circuito no debe retornar al tanque de agua más caliente de lo que salió puesto que eso significa que no se está realizando su función de cesión de energía calorífica.

$$T_{\text{retorno}} < T_{\text{impulso}}$$

- El sistema de bomba de calor siempre debe de recibir cierto suministro eléctrico:

$$\text{Consumo} > 0 \text{ (W)}$$

Por otro lado, también se destacan algunas recomendaciones facilitadas por el mismo fabricante:

- El sistema de bomba de calor deja de operar eficientemente si la temperatura exterior se encuentra fuera de ciertos márgenes:

$$T_{\text{exterior}} > -20^{\circ}\text{C}$$

$$T_{\text{exterior}} < 35^{\circ}\text{C}$$

- La potencia suministrada al sistema no debe superar los límites energéticos establecidos por el fabricante:

$$\text{Consumo} < 2.35 \text{ (W)}$$

- La temperatura del agua en el tanque no debe superar los límites establecidos por el proveedor:

$$T_{\text{impulso}} > 20^{\circ}\text{C}$$

$$T_{\text{impulso}} < 55^{\circ}\text{C}$$

El esquema del diseño de la solución (Fig. 9.2.1-D), contempla el desarrollo de un módulo cliente dedicado a la interacción entre el sistema y el usuario. Para que el usuario pueda obtener un feedback inmediato si alguna de estas condiciones no se cumpliera, se ha implementado un sistema de registro de alertas que el *digital twin* almacena a nivel local cuando, durante un período de tiempo, no se cumplan algunas de las reglas anteriores. El usuario podrá visualizar en la misma GUI del módulo cliente sobre qué componentes del sistema real no se está registrando una actividad adecuada.

Por otro lado, a partir de la misma metodología basada en estas reglas de comportamiento, también se despliega un registro de eventos benignos durante la actividad de la máquina. Algunos de las mediciones que se lleven a cabo también puede ser utilizadas para detectar cambios de estado en el sistema.

Algunos eventos que quedan registrados en el *digital twin* son:

- Arranque de la bomba de calor:

	Registro anterior	Registro actual
Consumo	0	>0

Tabla 9.2.1-A. Relación del evento de arranque de la bomba con el valor de consumo

- Detención de la bomba de calor:

	Registro anterior	Registro actual
Consumo	>0	0

Tabla 9.2.1-B. Relación del evento de apagado de la bomba con el valor de consumo

El entorno energético con el que se trabaja no permite ofrecer el feedback al gemelo físico dadas las interpretaciones de los resultados de los procesos de simulación. Se mencionó en el apartado anterior cómo el gemelo digital se apoyaría de un *broker* para permitir ese flujo bidireccional y la consecuente visibilidad directa con la unidad e control de la máquina (PLC) que se muestra en la figura 8-A.

Este punto detalla cómo se ha llevado a cabo la implementación del sistema monitorización en tiempo real de la bomba de calor con la que se trabaja en el proyecto para que el usuario final pueda realizar un seguimiento continuo del estado de este.

En este módulo del servidor, solo se tendrían en cuenta escenarios con el que el usuario podría encontrarse en el momento del seguimiento, se han diseñado otros módulos en el mismo servidor que permiten simular escenarios futuros que, mediante la realización de acciones predictivas, mejoren el rendimiento y eficiencia de la bomba anticipando futuros sucesos.

El código fuente en referencia al cálculo de estado del sistema real forma parte de la clase Decididor de la aplicación *flask*. Este puede consultarse en el Anexo II – Decider.py.

Cálculo del comportamiento del sistema simulado

La evaluación de riesgos y escenarios permite recrear situaciones y estados del sistemas actualmente desconocidos, para así estudiar la manera de actuar en consecuencia. Las tecnologías de simulación de escenarios son una herramienta permiten al gemelo digital anticiparse a futuros sucesos sobre el sistema físico que puedan traducirse en fallos en el funcionamiento del este o adaptarse a condiciones predecidas del entorno para maximizar la eficiencia y rendimiento del sistema.

En la sección 3.2, donde se estudia el estado del arte de las tecnologías de modelado y simulación basada en modelos de alta fiabilidad, se hace referencia a las herramientas de modelado numérico que permitirían simular este tipo de comportamientos de manera predictiva. Los modelos de *Machine Learning* basados en el aprendizaje supervisado, ofrecen un marco de trabajo que permite predecir futuros escenarios del entorno mediante los modelos numéricos formados a partir de conjuntos de datos que se han ido registrando históricamente.

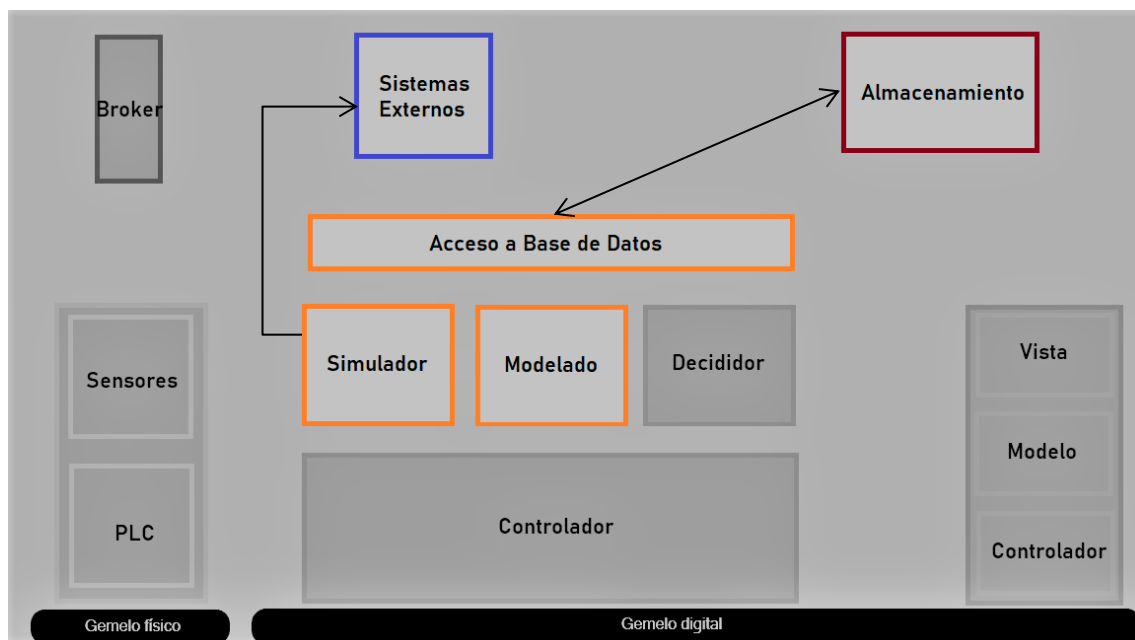


Fig.9.2.1-F. Principales componentes del gemelo digital dedicados al cálculo de comportamiento del sistema simulado.

En la la figura 9.2.1-F se observar los componentes principales del gemelo digital que se utilizan para implementar el cálculo del comportamiento del sistema mediante la simulación de escenarios, se han desarrollado los módulos “modelador” y “simulador” vistos en la figura 8-A. La herramienta escogida para el tratamiento de los datos ha sido la librería *pandas*, basada en *Python*. Aunque estos dos componentes mencionados son los que están específicamente dedicados a realizar las funciones predictivas, el proceso completo de simulación involucra también la actividad de otras unidades del gemelo digital.

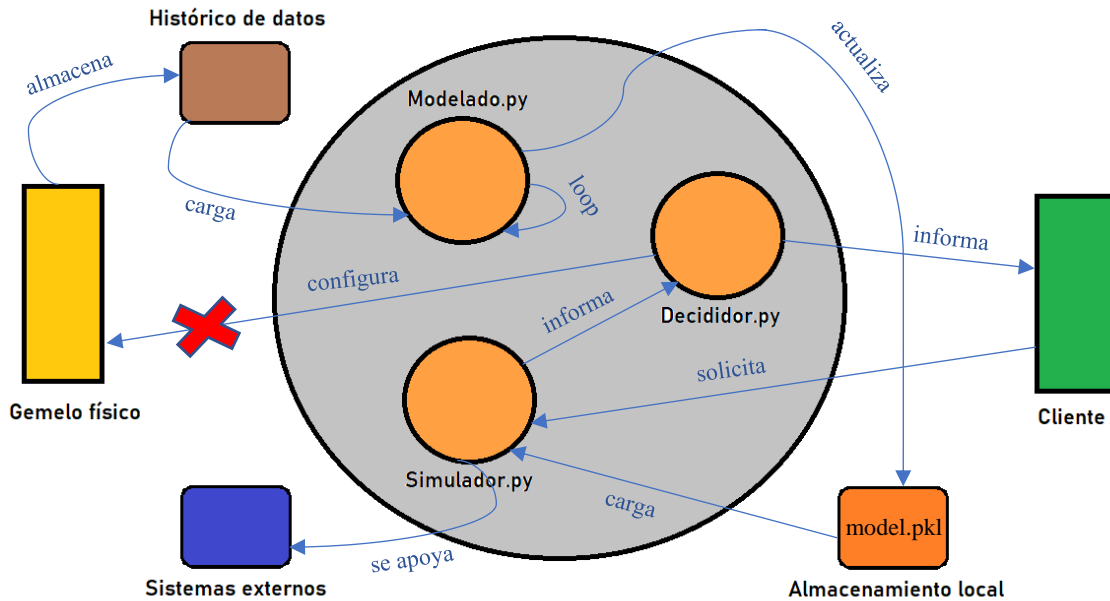


Fig. 9.2.1-G. Comunicación entre módulos en el proceso de simulación

La Fig 9.2.1-G muestra una serie de intercambios de información que se producen entre los módulos que intervienen en el el proceso de simulación de escenarios. Los siguientes puntos se dedican a poder interpretar este procedimiento:

- El **gemelo físico** carga periódicamente los datos registrados por los sensores que incorpora en un dispositivo de almacenamiento orientado a almacenar cantidades ingentes de datos.
- La base de datos que almacena el **histórico de datos** en crudo recogidos por el gemelo físico se basa en la tecnología MongoDB y hace referencia al módulo de Almacenamiento visto en la figura 9.2.1-F. La librería utilizable en código Python *pymongo* [74] permite gestionar este sistema de base de datos no relacional desde el servidor.
- El módulo de **modelado** y el **simulador** se ejecuta en segundo plano para no interferir sobre las operaciones básicas del servidor *flask*.
- Este servicio de **modelado** opera con los datos registrados en el histórico, crea y actualiza periódicamente los modelos semánticos que serán posteriormente almacenados en el **almacenamiento local**, en este caso, en formato *.pkl*.
- Tanto el **cliente** como el propio **decididor** del servidor, este último de manera autónoma, pueden solicitar al **simulador** que opere con los modelos del **almacenamiento local** del servidor para predecir escenarios no contemplados anteriormente.
- El **simulador**, en segundo plano, puede apoyarse de fuentes externas, como por ejemplo información referente a predicciones meteorológica, para trabajar con los modelos semánticos que ha cargado desde los archivos generados en el

almacenamiento local del servidor. En el caso concreto de este proyecto, se simula el funcionamiento de una base de datos externa que aporte predicciones meteorológicas a partir de la misma actividad registrada históricamente por los sensores de temperatura exteriores instalados en el entorno.

- El **simulador** informa únicamente al **decididor** al obtener los resultados del proceso. En el decididor se habrán convertido estos resultados en posibles eventos o alertas.
- El **decididor**, que no interviene activamente en el proceso de simulación, tiene la potestad de informar al **cliente** sobre la interpretación que se ha dado de estos resultados, si este la solicita previamente.

El entorno energético con el que se trabaja no permite al gemelo digital ofrecer feedback al gemelo físico a partir de los resultados de los procesos de simulación. El punto 9.2.1-*Comunicación bidireccional* habla de cómo el entorno ya estaba preparado para llevar a cabo este proceso. Será aprovechando el despliegue de *Mosquito* la manera en la que se implemente el sistema de feedback entre el servidor *flask* y la controladora embebida en el sistema de bomba de calor.

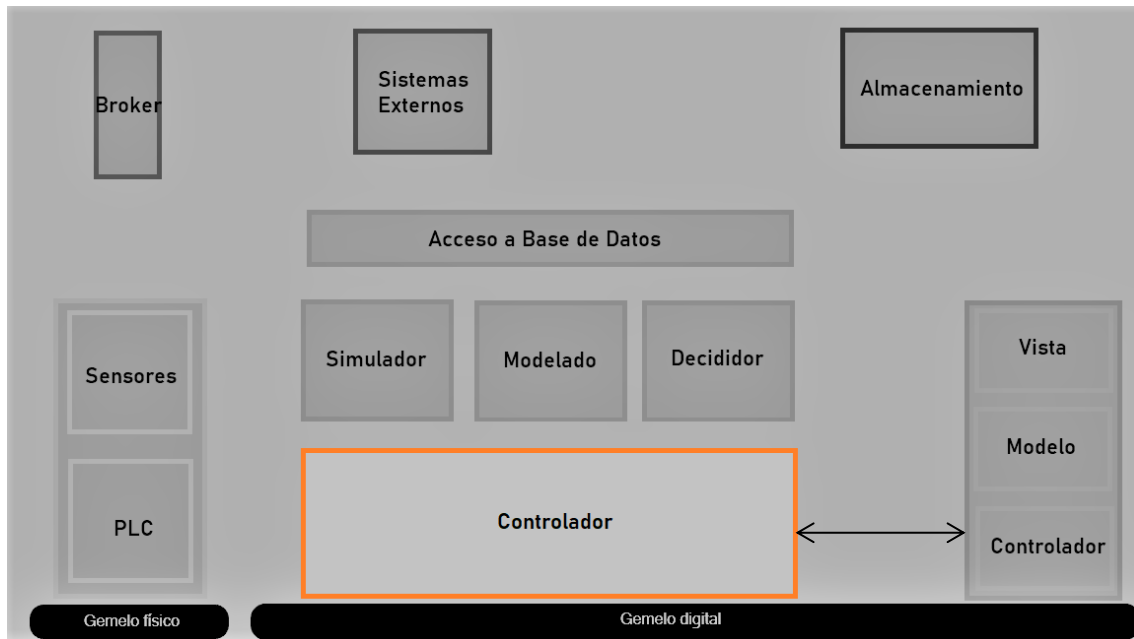
En estos párrafos se ha resumido el procedimiento de cálculo del futuro comportamiento del gemelo físico a partir de un sistema virtual, en este caso basado fundamentalmente en modelos semánticos; se ha implementado una funcionalidad específica para anticipar el momento de arranque bajo dada la fecha y hora de utilización.

El código fuente en referencia al cálculo del comportamiento del sistema simulado forma parte de la clase “Modelado” y “Simulador” de la aplicación *flask*. Ambas clases se han documentado en el Anexo II.

Así como se establece un sistema de interacción entre *physical twin* y el *digital twin* permitiendo ese feedback, el usuario final requiere un trato similar. En el próximo punto se describe cómo se implementa el sistema que permitirá al cliente interactuar con el servidor para poder visualizar en su dispositivo la información extraída a partir de estos módulos de análisis de datos.

Interacción con usuario – REST API

Una de las principales fuentes de valor de los *digital twins* es la posibilidad para el usuario de controlar y monitorizar en tiempo real el sistema físico. La figura 8-A presentada en el diseño, anticipa la existencia de un sistema de interacción entre el dispositivo del usuario sobre el que visualiza los datos (cliente) y el servidor, encargado de analizar el comportamiento actual y futuro de la bomba de calor.



F. 9.2.1-H. El controlador como sistema de interacción con el cliente

Así como muestra la figura 9.2.1-H, se despliega sobre la capa de control del servidor el sistema que permita conectar el entorno de visualización con esta unidad de procesamiento de los datos. Se trata de un servicio REST API mediante el cual el cliente puede consultar información del estado del sistema a partir de un listado de rutas. La comunicación entre el servidor y el cliente que se ejecute en el dispositivo del usuario final se basa en la metodología CRUD. Puesto que las consultas y acciones realizadas sobre el sistema se realizan de manera esporádica, no se ha visto la necesidad de desplegar otro tipo de sistema que diese continuidad a la sesión, a diferencia del mecanismo de comunicación entre los dos gemelos.

Para poder realizar estas operaciones se define un listado de rutas en servidor *flask* a las que el usuario pueda referirse para extraer o depositar información en el gemelo digital vía peticiones *http/https*.

Algunos ejemplos de consulta GET son:

- https://127.0.0.0:5000/sensor_values/ → Solicita el valor de las mediciones de los sensores más reciente hasta la fecha.
- <https://127.0.0.1:5000/alerts/> → Solicita un listado de alertas a considerar dado el estado actual de la máquina y su entorno.
- <https://127.0.0.1:500/events/> → Solicita la lista de las últimas operaciones realizadas sobre el sistema.
- <https://127.0.0.1:500/status/> → Consulta el valor de los parámetros más significativos del sistema.

Por otro lado, el usuario también debe tener la opción de intervenir proactivamente sobre las condiciones del sistema. Algunos ejemplos de peticiones POST son los siguientes:

- <https://127.0.0.1:5000/events/> → Solicita un cambio específico en los parámetros de configuración del sistema.
- https://127.0.0.1:500/predict_boot/ → Solicita el arranque automático de la máquina bajo cierta parametrización.

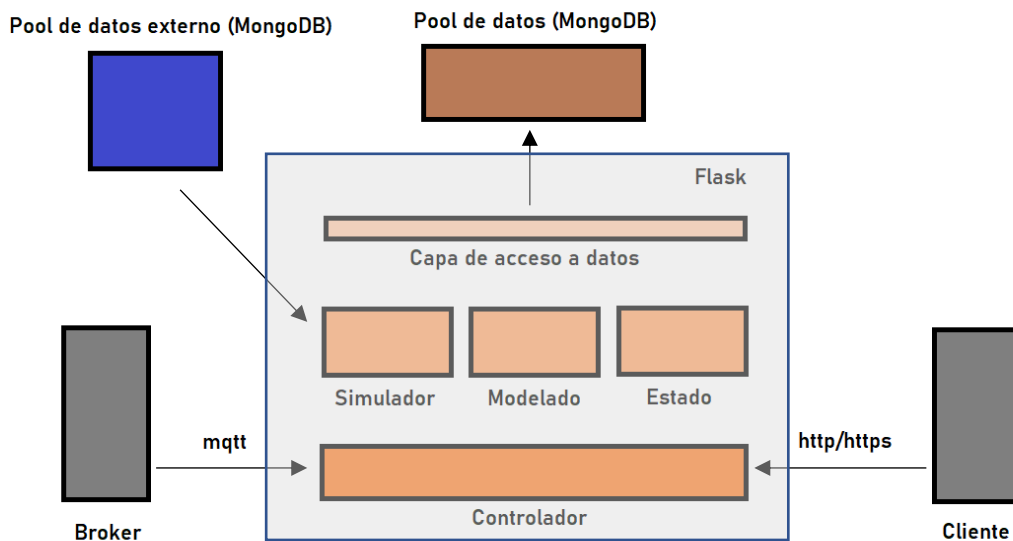


Fig. 9.2.1-I. Esquema general del lado servidor.

La Figura 9.2.1-I muestra desde otra perspectiva la manera en la que se ha desplegado el servidor en el host. En ella, se observan los diferentes sistemas que interactúan directamente con él.

A la izquierda, se aprecia la comunicación establecida con el *broker*. Inicializada por el controlador, se establece una sesión *mqtt* que dota de bidireccionalidad y continuidad a la sesión entre gemelos físico y virtual

A la derecha, se facilita al cliente a través de una comunicación *http/https*, la información que se ha recogido a partir de los cálculos de comportamiento que realiza el servidor así como datos actuales de la bomba.

En el centro, se detalla cómo el servidor *flask* se ha desplegado en una serie de capas, alineando este desarrollo con el diseño mostrado en la figura 9.2.1-A:

- La capa de control, encargada de inicializar las capas de servicio superiores y los sistemas de interacción tanto con el gemelo físico a través de un *broker* como con el cliente alojado en el dispositivo del usuario.

- La capa de servicio, en la que se desplieguen ambas funcionalidades de cálculo de comportamiento de real y virtual a través del módulo decididor y el módulo de modelado y simulación, respectivamente. También muestra cómo estos servicios tienen la capacidad de alimentarse de los datos provenientes de sistemas externos y del *pool* de datos local.
- La capa de acceso a datos, que permite abrir una conexión con el *pool* de datos local donde se almacena los registros que el gemelo físico va recogiendo.

El código fuente en referencia al módulo de interacción con el cliente forma parte del programa principal de la aplicación *flask*. Este servicio de *routing* que lo conforma se puede ver en el Anexo II – Controller.py.

La siguiente sección se dedica al lado del cliente, este el módulo que correrá en el mismo dispositivo del usuario desde el que podrá realizar el seguimiento del comportamiento de la bomba.

9.2.2. Desarrollo del cliente

Los *digital twins* son sistemas autónomos. Son una representación digital de su homólogo físico y por ello se enfatiza respecto a esta similitud en la palabra “*twin*”. Ambos se encuentran en un estado de comunicación continuo y, parcialmente, eso permite que sean semejantes.

El usuario del sistema de bomba de calor debe de disponer visualmente de toda la información relevante que gemelo digital pueda ofrecerle de su homólogo física para poder realizar un seguimiento de su funcionamiento; esta interfaz también debe permitir que pueda intervenir activamente en el comportamiento del mismo. El cliente es el módulo del gemelo digital encargado de dibujar la realidad del sistema a través de una interfaz gráfica amigable con la que este pueda interactuar a través de su dispositivo.

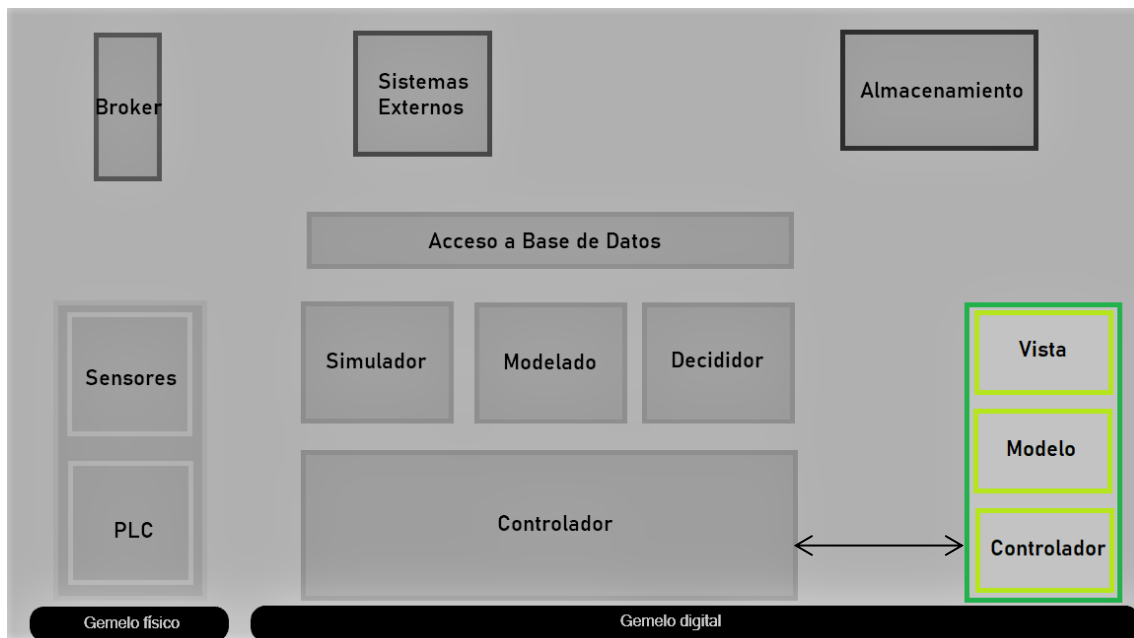


Fig. 9.2.2-A. El cliente como sistema de visualización de los datos e interacción con el usuario.

Respetando la arquitectura Modelo-Vista-Controlador (MVC) propuesta en la sección de diseño 8.2.2, se han implementado tres submódulos de la siguiente manera para que el usuario pueda monitorizar la bomba de el mismo navegador.

- **Modelo.** Es una clase inicializada desde el controlador que permite almacenar en caché la información referente al gemelo digital a partir de una estructura de variables de clase predefinida.
- **Vista.** Se trata del documento HTML que, abierto desde el navegador del usuario, ofrece gráficamente una visión del estado y comportamiento del sistema de bomba de calor. En su desarrollo, se ha hecho uso de la herramienta de bootstrap Atlantis Lite, especialmente dedicado a la monitorización de entornos IoT.

- Controlador. Programa basado en Javascript que permite controlar el DOM de la vista, dinamizando la información que aparece en ella. Su funcionalidad de dinamizar el la vista se complementa con el sistema de peticiones HTTP a través de una clase *HTTPGateway.js* que opera como puerta de enlace entre módulos. El servidor responde con la información referente al comportamiento de la bomba de calor que sea de utilidad para el usuario y la presenta en la vista.

Integrando estos tres submódulos, el cliente ofrece al usuario la total visibilidad de la información que procesa el gemelo digital así como la oportunidad de interactuar con la máquina de manera proactiva.

El código fuente en referencia a la visualización del estado del sistema viene recogido en el Anexo III. Este documento se estructura según las partes que forman el módulo del cliente: Modelo, Vista y Controlador.

En este capítulo se ha presentado el desarrollo de una aplicación web que permite monitorizar y predecir el comportamiento del sistema físico con el que se trabaja a partir del conjunto de registros de actividad recogidos por una red sensores. Aunque para explicar el código se ha optado por realizar una explicación a alto de nivel de la aplicación diseñada, este puede consultarse con detalle en los anexos I, II y III.

En el capítulo 10 se presenta una batería de pruebas que se han llevado a cabo para demostrar la capacidad del gemelo digital de procesar e interpretar la información en tiempo real registrada por el gemelo físico, así como realizar de manera autónoma los cálculos a través del sistema virtual que permitan mejorar la eficiencia, el rendimiento de la máquina y, visto el entorno de trabajo, el confort del usuario que la instala.

10. Resultados y trabajos futuros

En esta sección se definen una serie de pruebas que, sin ser exhaustivos, permiten cubrir algunas de las expectativas principales de los gemelos digitales y que dan una idea completa del funcionamiento del sistema. Entre estas expectativas se encuentra:

- Establecer una comunicación en tiempo real con el gemelo físico.
- Reflejar el comportamiento y monitorizar el estado del sistema real.
- Predecir el comportamiento del sistema físico ante escenarios futuros a partir del conocimiento sobre su actividad previa.
- Incorporar una interfaz gráfica desde la que el usuario puede realizar su seguimiento e interactuar con el sistema.

Se ha dividido el capítulo cuatro secciones. Las tres primeros puntos incorporan una serie de pruebas que permitirían cubrir una o varias de las expectativas mencionadas anteriormente. En la última sección, se enumeran algunos trabajos futuros que permitan seguir poniendo en manifiesto las utilidades de los gemelos digitales en entornos domésticos.

10.1. Resultados de la interacción bidireccional entre gemelos

Con el fin de permitir al gemelo digital establecer una comunicación en tiempo real con el gemelo físico, se ha desplegado un servidor servidor *Mosquitto* que ejerce de intermediario en la comunicación entre el gemelo digital y el gemelo físico. En este entorno se permite que cada cliente trabaje tanto de emisor como de receptor de mensajes información.

Los mensajes que viajan por el canal pueden fluir en ambos sentidos y están etiquetados. Los clientes podrán recibir los mensajes cacheados en el mediador o *broker* que otro emisor haya depositado anteriormente.

Se ejecutado el intermediario sobre una máquina virtual independiente basada en Ubuntu. Este servidor *Mosquitto* escucha desde el puerto 1883.

```
jfm174@jfm174: ~  
jfm174@jfm174:~$ mosquitto -c /etc/mosquitto/mosquitto.conf  
1625423884: mosquitto version 1.4.10 (build date 2021-05-02 19:01:36+0200) starting  
1625423884: Config loaded from /etc/mosquitto/mosquitto.conf.  
1625423884: Opening websockets listen socket on port 9001.  
1625423884: Opening ipv4 listen socket on port 1883.  
1625423884: Opening ipv6 listen socket on port 1883.  
1625424148: New connection from 192.168.1.126 on port 1883.  
1625424149: Socket error on client <unknown>, disconnecting.  
1625424149: New connection from 192.168.1.126 on port 1883.  
1625424150: New client connected from 192.168.1.126 as mosquitto_client_02 (c1, k65).  
1625424150: Sending CONNACK to mosquitto_client_02 (0, 0)  
1625424151: Received SUBSCRIBE from mosquitto_client_02  
1625424151: sensor_values (QoS 0)  
1625424151: mosquitto_client_02 0 sensor_values  
1625424151: Sending SUBACK to mosquitto_client_02  
1625424171: New connection from 192.168.1.126 on port 1883.  
1625424172: New client connected from 192.168.1.126 as mosquitto_client_01 (c1, k65).  
1625424172: Sending CONNACK to mosquitto_client_01 (0, 0)  
1625424173: Received PUBLISH from mosquitto_client_01 (d0, q0, r0, m0, 'sensor_values', ... (308 bytes))  
1625424174: Sending PUBLISH to mosquitto_client_02 (d0, q0, r0, m0, 'sensor_values', ... (308 bytes))  
1625424174: Received SUBSCRIBE from mosquitto_client_01  
1625424174: plc_control (QoS 0)  
1625424174: mosquitto_client_01 0 plc_control  
1625424174: Sending SUBACK to mosquitto_client_01
```

Fig. 10.1-A. Sesión MQTT con Mosquitto

En la Fig. 10.1-A se establece una sesión *mqtt* entre el gemelo físico y el gemelo digital. Algunos aspectos a considerar antes son los nombres que han recibido cada uno de ellos para establecer la sesión:

- Gemelo físico: *mosquitto_client_01*
- Gemelo digital: *mosquitto_client_02*

Durante el procedimiento, en primer lugar, el cliente *mosquitto_client_02* se inicializa tanto como suscriptor de mensajes etiquetados como “*sensor_values*”, a lo que se refiere a los registros enviados por el gemelo físico, por lo tanto, estará preparado para recibir ese tipo de mensajes.

Por otro, *mosquitto_client_01* también se inicializará como suscriptor, en este caso, para poder recibir información del tipo “*plc_control*” en sentido contrario y desde el gemelo digital.

En el registro 1625424173 del servidor *Mosquitto*, el cliente *mosquitto_client_01* deposita en el *broker* información referente a la actividad de los sensores. Posteriormente, el mismo *broker* entrega la información al gemelo digital (*mosquitto_client_02*).

Con esta intercambio de información, se valida la visibilidad entre gemelo físico y digital y el flujo de información entre los entes en tiempo real.

10.2. Visualización en tiempo real del comportamiento del gemelo físico

En esta sección se demuestra la capacidad del gemelo digital de reflejar el comportamiento del sistema real y dar feedback al usuario respecto a las condiciones actuales de la bomba de calor. Para conseguir este objetivo, se alimentará de manera periódica al gemelo digital mediante una modificación del registro original de datos de la bomba de calor de Ca Ses Llúcies entre las siguientes fechas:

- 22/03/2021 | 13:00h
- 26/03/2021 | 17:00h

En el Anexo VI se visualiza cómo la actividad registrada originalmente (AnexoIV) ha sido modificada con el fin de representar situaciones indeseadas del entorno, por ejemplo:

```
Series;Time;Value
Bomba de calor;2021-03-22T13:00:00+01:00;0
Bomba de calor;2021-03-22T13:15:00+01:00;0
Bomba de calor;2021-03-22T13:30:00+01:00;0
Bomba de calor;2021-03-22T13:45:00+01:00;0
Bomba de calor;2021-03-22T14:00:00+01:00;0
...
```

```
Series;Time;Value
Bomba de calor;2021-03-22T13:00:00+01:00;0
Bomba de calor;2021-03-22T13:15:00+01:00;0
Bomba de calor;2021-03-22T13:30:00+01:00;200
Bomba de calor;2021-03-22T13:45:00+01:00;200
Bomba de calor;2021-03-22T14:00:00+01:00;0
...
```

Ejemplo: Se modifica el valor del consumo energético de la bomba de calor a un nuevo valor por encima de los límites deseados por el fabricante.

Al iniciar el entorno, el gemelo digital registrará en tiempo real los valores generados por el gemelo físico:

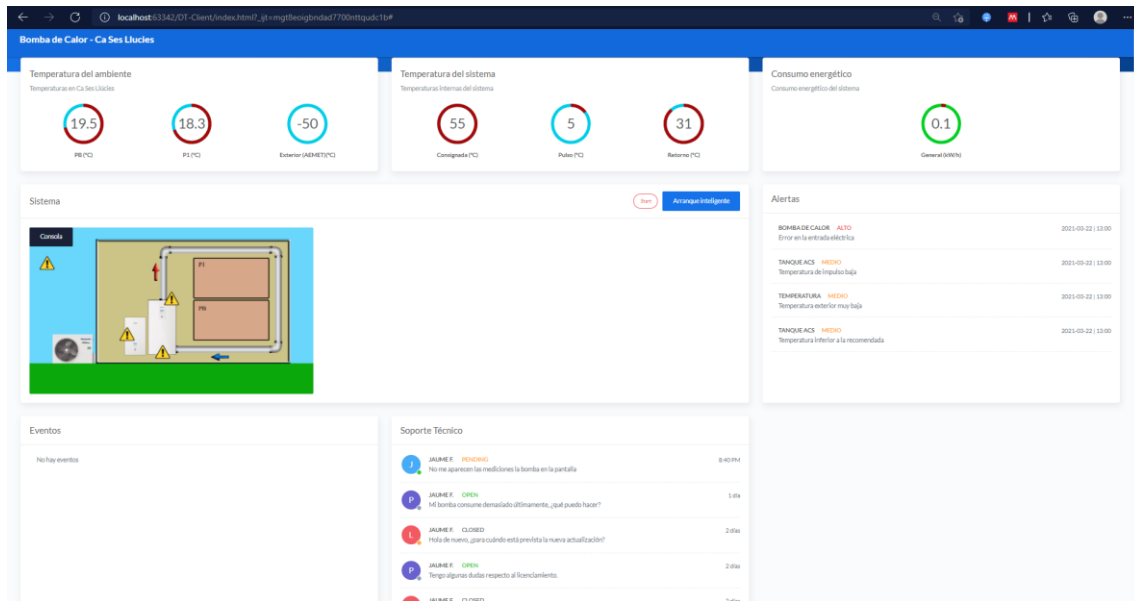


Fig. 10.2-A. Vista del estado del sistema 22-03-2021 / 13:00.

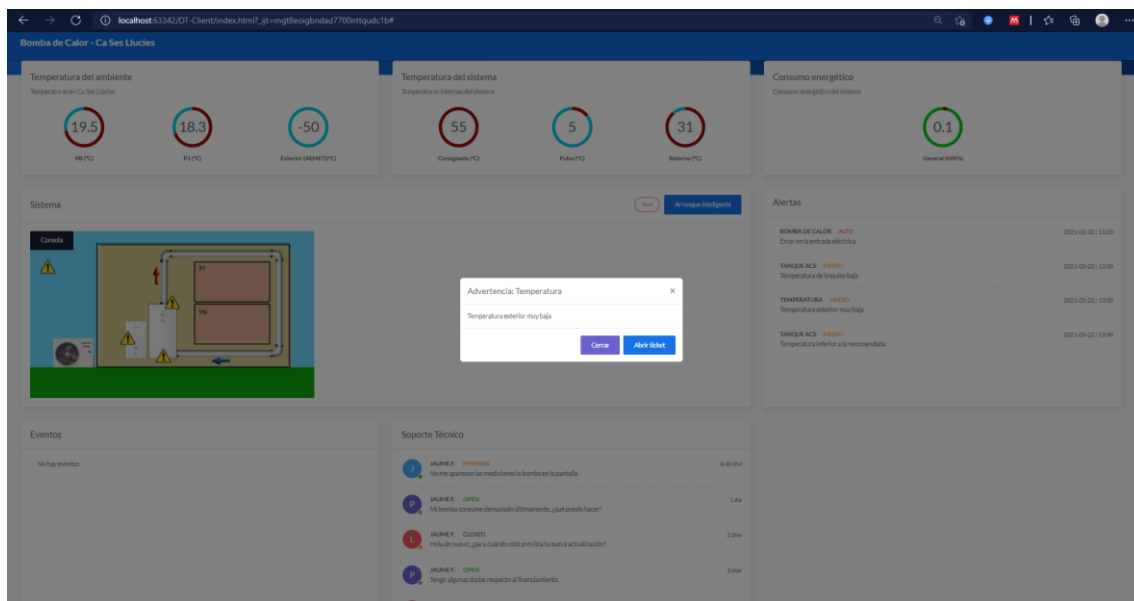


Fig. 10.2-B. Detalle de alerta – Temperatura exterior.

Tanto las alertas como los eventos que se le muestren al usuario dependerán del cumplimiento de las reglas definidas en el apartado “9.2.1-Descripción del estado del sistema real”. En el primer registro de la actividad de la bomba de calor, se detectan numerosas alertas respecto a su comportamiento que se deben considerar:

- Bomba de calor - Error en la entrada eléctrica
- Tanque ACS - Temperatura de impulso baja

- Temperatura - Temperatura exterior muy baja
- Tanque ACS - Temperatura inferior a la recomendada

Conforme se van procesando nuevos registros, la información sobre el estado del sistema de bomba de calor ha variado:

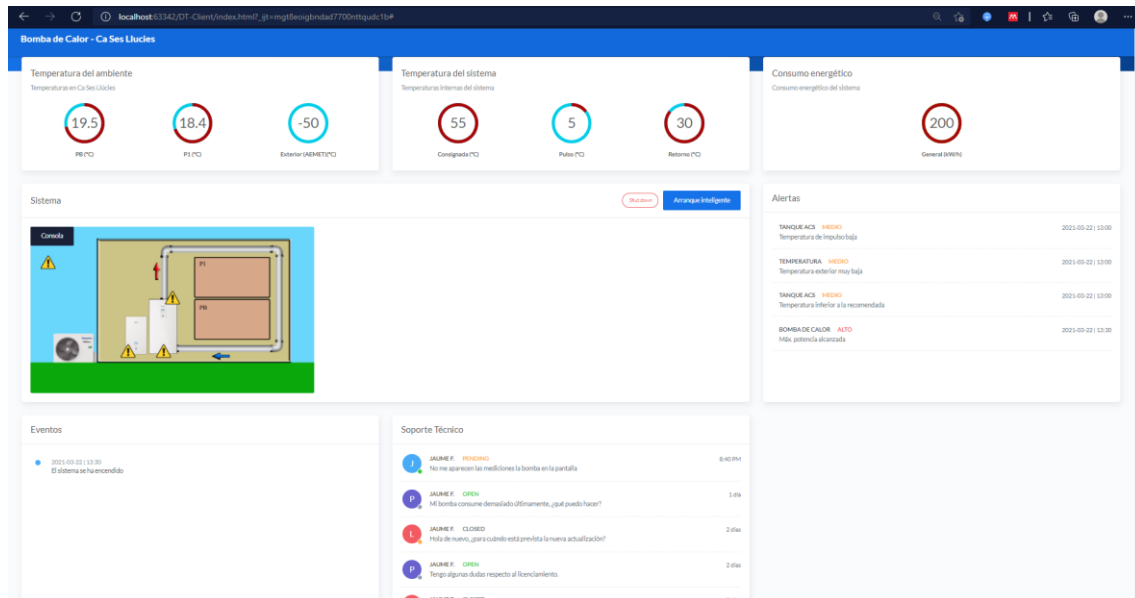


Fig. 10.2-C. Vista del estado del sistema 22-03-2021 / 13:30.

En este nuevo escenario, se elimina la alerta en referencia al error en la entrada eléctrica, sin embargo, aparecen nuevas alertas por considerar:

- Tanque ACS - Temperatura de impulso baja
- Temperatura - Temperatura exterior muy baja
- Tanque ACS - Temperatura inferior a la recomendada
- Bomba de calor – Máx. potencia alcanzada

Por otro lado, se visualizan otros cambios de estado benignos de la bomba que no se habían contemplado anteriormente. En este caso, se registra un evento de arranque de la bomba en la sección de “Eventos”.

- 2021-03-22 | 13:00 – El sistema se ha encendido

En el registro del 22-03-2021 | 15:00, todos los componentes ya se encuentran en correcto funcionamiento y, además, vuelven a aparecer nueva actividad en relación a los eventos de sistema:

- 2021-03-22 | 13:00 – El sistema se ha encendido
- 2021-03-22 | 14:00 – El sistema se ha detenido
- 2021-03-22 | 14:00 – El sistema se ha encendido
- 2021-03-22 | 15:00 – El sistema se ha detenido

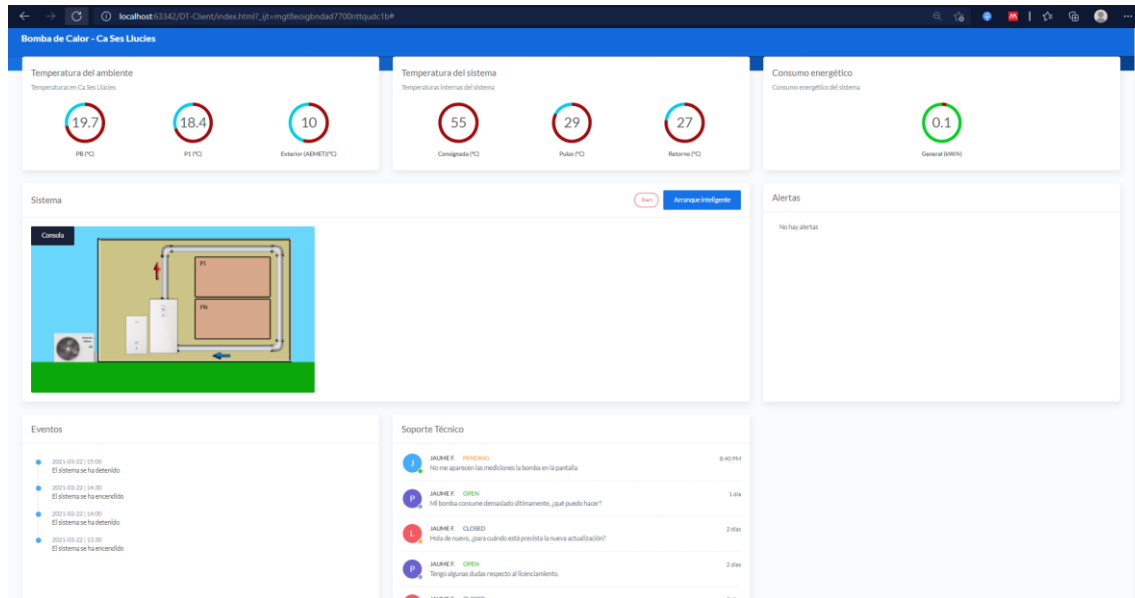


Fig. 10.2-C. Vista del estado del sistema 22-03-2021 | 15:30.

Con esta batería de pruebas, se refleja el ejercicio de interpretación por parte del gemelo digital del comportamiento del sistema a partir de las mediciones llevadas a cabo por el gemelo físico. El análisis de los datos entrantes en tiempo real aporta al usuario una solución para poder conocer la correctitud del funcionamiento del sistema a partir del sistema de alertas y eventos.

10.3. Resultados de la simulación en el comportamiento futuro del sistema

El gemelo digital debe tener la capacidad de adaptarse a escenarios futuros del sistema físico a partir de su actividad previa con la intención de mejorar la eficiencia y el rendimiento. La prueba realizada demuestra la capacidad del sistema para predecir su tiempo de arranque a partir de unas condiciones temporales y térmicas determinadas. Esta funcionalidad permite ajustar la temperatura del tanque deseada a la hora de entrada del usuario en la vivienda.

Se ha hecho uso de dos subconjuntos del histórico de datos original registrado por la bomba de calor en sus fechas de operatividad. Estos subconjuntos vienen recogidos en el Anexo V. La siguiente tabla muestra cómo se ha realizado el corte en el dataset original según la fecha y hora del registro:

	Fecha Inicial	Hora Inicial	Fecha Fin	Hora Fin
Dataset original	22-03-2021	13:00	26-03-2021	17:30
Dataset de entrenamiento (train)	22-03-2021	13:00	25-03-2021	22:00
Dataset de pruebas (test)	25-03-2021	22:00	26-03-2021	17:30

Tabla 10.3-B. Valor de la fecha de inicio y fin del dataset de prueba (Req_03)

A partir de la información que aparecen en la Tabla 10.3-B, el escenario planteado es el siguiente:

El modelo de aprendizaje supervisado planteado en el apartado 9.4.1 – *Cálculo del comportamiento del sistema simulado* permite predecir el arranque de la bomba de calor a partir de las experiencias pasadas durante los procesos de arranque.

El cliente deberá introducir el momento en el que requerirá del uso del ACS, para ese momento, la temperatura del circuito deberá ser la deseada (55°C). Para realizar este proceso el gemelo digital ha hecho uso del modelo entrenado a partir del del histórico de datos perteneciente al “Dataset de entrenamiento” y se ha apoyado en los sistema externos de predicciones meteorológicas mencionados en el apartado 9.2.1, también simulado mediante un pequeño sistema de base de datos alojado de manera local.

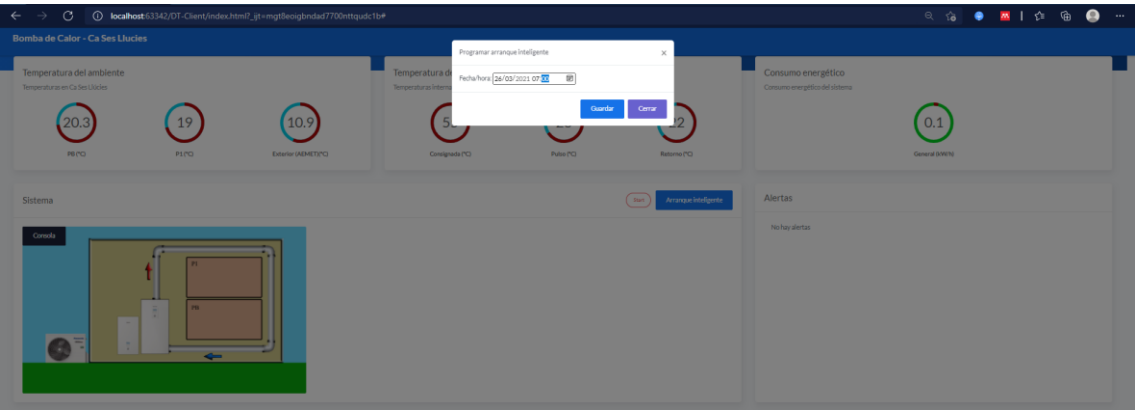


Fig. 10.3-A. Solicitud de arranque automático.

Una vez realizada la solicitud, el gemelo digital retorna un evento de confirmación al cliente para ofrecer feedback de la acción realizada. Se ha programado el inicio de la

bomba para satisfacer la necesidad del usuario, esta deberá estar operativa bajo en la hora deseada.

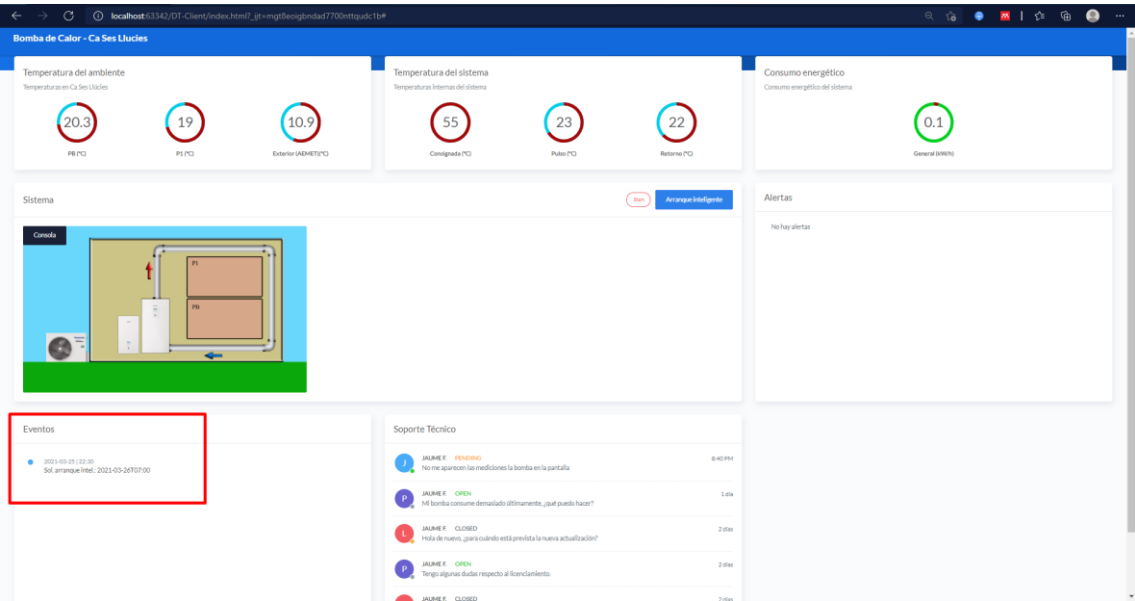


Fig. 10.3-B. Confirmación de solicitud de arranque automático

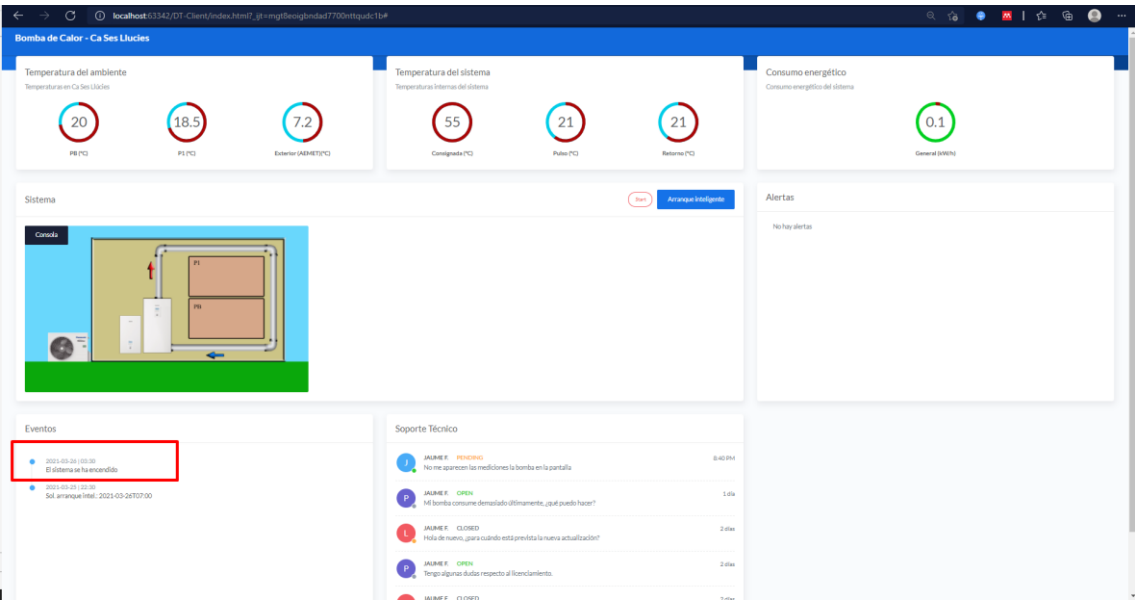


Fig. 10.3-C. Confirmación de arranque del sistema

La Fig 10.3-C muestra la confirmación de arranque de la bomba de calor. Esta se ajusta su comportamiento de manera autónoma a partir de las condiciones que el usuario le solicitó. Esta programación se ha establecido únicamente en el gemelo digital; en trabajos futuros y en un entorno energético con menores limitaciones se comandará al gemelo físico que el evento se debe producir al momento que el gemelo digital lo considere. El diagrama temporal de esta demostración se reproduce en el siguiente imagen:

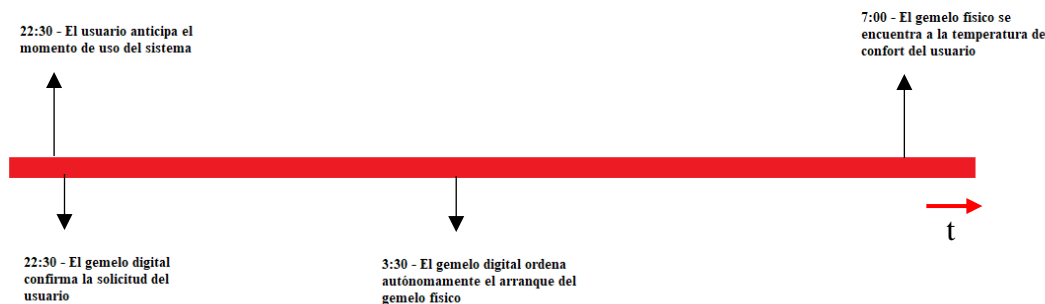


Fig 10.3-D. Esquema temporal del proceso de arranque inteligente.

Para demostrar la mejora de la eficiencia que supone esta solución, se ha comparado el esquema de la figura Fig. 10.3-D con el funcionamiento habitual de la máquina. En la siguientes imágenes, se adjunta el momento de arranque real que tuvo el sistema físico en la misma fecha sobre la que se ha realizado la simulación:

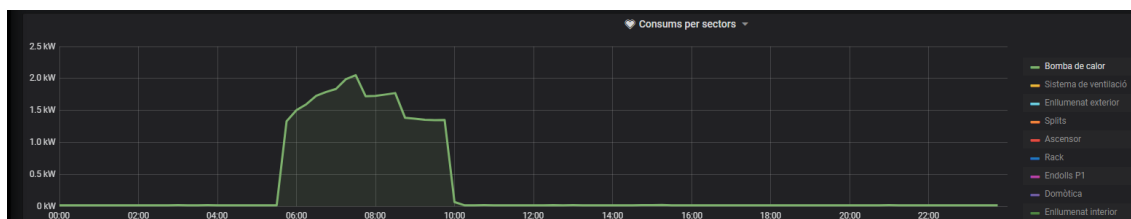


Fig. 10.3-E. Entrada eléctrica real de la bomba de calor durante la jornada del 26-03-2021.

La figura Fig. 10.3-E muestra la curva del consumo real de la bomba de calor durante la jornada del 26-03-2021. En la imagen se aprecia el arranque del sistema real a las 5:45AM. Este no alcanzó la temperatura deseada por el usuario (55°C) hasta las 10:00AM.

A diferencia de la actividad llevada a cabo utilizando la solución del gemelo digital descrita en la Fig. 10.3-D, la curva de consumo de la jornada real muestra la ineficiencia en el funcionamiento de la bomba. Durante el intervalo de tiempo entre las 7:00AM y las 10:00AM el sistema de calefacción no está cumpliendo con las expectativas del usuario.

En resumen, esta batería de pruebas tiene el fin de poner en manifiesto, en un entorno concreto, las carencias de los sistemas domóticos en términos de transparencia y eficiencia actualmente desplegados en entornos domésticos y valorar las ventajas que la visión de los *digital twins* puede presentar. La interacción entre gemelo digital y gemelo físico en tiempo real permite que el sistema virtual se aproveche al momento de la información que su homólogo registre de su entorno. Así, además de la función de reflejar el comportamiento real de este, pueda hacer su función propia de ente virtual simulando escenarios que todavía no se hayan registrado y trabajar proactivamente para anticipar futuros sucesos. Se ha presentado al usuario una interfaz gráfica amigable que transparenta este flujo de datos entre ambos entes y la interpretación que el gemelo digital les da.

La sección siguiente redacta una serie de trabajos futuros que se han contemplado para seguir poniendo en manifiesto las utilidades que el paradigma de los gemelos digitales puede ofrecer.

10.4. Trabajo futuros

El modelo desarrollado es plenamente funcional y cumple con los requerimientos planteados. El presente trabajo, sin embargo, abre la puerta a otras soluciones más complejas que permitirían aumentar el alcance del proyecto.

Con vistas a la realización de nuevos trabajos futuros, se propone:

- Implementar un módulo de control sobre el gemelo físico que permita controlar el comportamiento del sistema real con vistas a aumentar el alcance del proyecto presente.
- Realizar la migración del gemelo digital a un entorno *cloud*, más escalable, seguro y fiable.
- Trabajar en la normalización de las actuales redes heterogéneas de los sistemas ciberfísicos con el fin de facilitar la integración y comunicación entre múltiples agentes.
- Desarrollar nuevas soluciones basadas en gemelos digitales sobre otros sistemas del entorno doméstico (ventilación, suministro energético, etc.) con vistas a establecer una sociedad digital e intercomunicada que permita mejorar las prestaciones individuales de cada unidad.

11. Conclusiones

Se han alcanzado los objetivos planteados al principio del presente trabajo. Se ha diseñado, implementado y testado un gemelo digital utilizando las tecnologías existentes que representa el funcionamiento de la bomba de calor aire-agua que opera en el Centro de Innovación Ca Ses Llúcies.

En este gemelo digital se ha visto cómo se permite monitorizar el estado del gemelo físico y tomar decisiones de manera predictiva a partir de los datos en crudo que este mismo y fuentes exteriores proporcionan.

Este proceso ha tenido implicaciones directas sobre la visibilidad y transparencia en el comportamiento del sistema, su eficiencia energética, el rendimiento y sobre el confort del mismo usuario. Se ha descrito cómo la solución implementada cumple con algunas de las expectativas esperadas de los gemelos digitales a partir de la batería de pruebas realizada.

12. Bibliografía

- [1] Priya Patra, “Digital twin: the physical world meets the digital world,” 2018. [Online]. Available: <https://www.capgemini.com/2018/08/digital-twin-physical-meets-digital-world/>.
- [2] G. O. Katie Costello, “Gartner Survey Reveals Digital Twins Are Entering Mainstream Use,” *Gartner*, 2019.
- [3] Open Sistemas, “DIGITAL TWIN, LA INTELIGENCIA OPERACIONAL.” [Online]. Available: <https://opensistemas.com/caso-uso/digital-twin-la-inteligencia-operacional/>.
- [4] Siemens, “NX.” [Online]. Available: <https://www.plm.automation.siemens.com/global/en/products/nx/>.
- [5] Siemens, “Getting to market quickly,” 2015. [Online]. Available: <https://new.siemens.com/global/en/company/stories/industry/getting-to-market-quickly.html>.
- [6] DTC Innovation Forum, “Digital Twin Computing,” 2019. [Online]. Available: https://www.rd.ntt/e/dtc/DTC_Whitepaper_en_2_0_0.pdf.
- [7] Siemens, “Building Twin.” [Online]. Available: <https://new.siemens.com/global/en/products/buildings/digital-building-lifecycle/building-twin.html>.
- [8] “Centro de innovación y transferencia Ca ses Llúcies,” *Universitat de les Illes Balears*. <https://smart.uib.es/Activitats/Projectes/Ca-ses-Llucies/Presentacio-Ca-ses-Llucies/>.
- [9] Y. N. Harari, *Sapiens, De Animales a dioses*. 2011.
- [10] “Mecanización,” *Wikipedia*. [Online]. Available: <https://es.wikipedia.org/wiki/Mecanización>.
- [11] alvaroh, “La IA en la Industria 4.0,” *PublicaTIC*, 2020.
- [12] M. Liu, S. Fang, H. Dong, and C. Xu, “Review of digital twin about concepts, technologies, and industrial applications,” *J. Manuf. Syst.*, vol. 58, no. October 2019, pp. 346–361, 2021, doi: 10.1016/j.jmsy.2020.06.017.
- [13] Jarrett Hendricks, “Rise of the Digital Twin: How Lessons Learned from NASA Are Changing the Way Supply Chains Are Managed.” Expeditors, 2020, [Online]. Available: <https://info.expeditors.com/horizon/rise-of-the-digital-twin>.
- [14] @ge, “The Democratization of the Digital Twin for Tool Assemblies,” *MachiningCloud*.
- [15] M. Grieves, “Product Lifecycle Management: Driving the Next Generation of Lean Thinking,” 2005. [Online]. Available: https://www.researchgate.net/publication/247267862_Product_Lifecycle_Management_Driving_the_Next_Generation_of_Lean_Thinking.
- [16] Q. Y. Peng Wang, Mei Yang, Yong Peng, Jiancheng Zhu, Rusheng Ju, “Sensor Control in Anti-Submarine Warfare—A Digital Twin and Random Finite Sets Based Approach,” 2019. [Online]. Available: <https://www.mdpi.com/1099-4300/21/8/767/htm>.
- [17] P. Andre, F. Azzi, O. Cardin, and P. André, “Heterogeneous Communication Middleware for Digital Twin Based Cyber Manufacturing Systems Heterogeneous Communication Middleware for Digital Twin Based Cyber Manufacturing Systems. Service Oriented, Holonic and Multi-agent Manufacturing Systems for Indust,” vol. 853, p. 10, 2019, [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02382463>.

- [18] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.
- [19] "Basics of Wireless Sensor Networks (WSN) | Classification, Topologies, Applications." <https://www.electronicshub.org/wireless-sensor-networks-wsn/>.
- [20] Priya, "Wireless Sensor Networks (WSN) : IoT Part 34," 2018. [Online]. Available: <https://www.engineersgarage.com/wireless-sensor-networks-wsn-iot-part-34/>.
- [21] P. Maiti, B. Sahoo, A. K. Turuk, and S. Satpathy, "Sensors data collection architecture in the Internet of Mobile Things as a service (IoMTaaS) platform," *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, no. February, pp. 578–582, 2017, doi: 10.1109/I-SMAC.2017.8058245.
- [22] S. S. Bhunia, J. Pal, and N. Mukherjee, "Fuzzy assisted event driven data collection from sensor nodes in sensor-cloud infrastructure," *Proc. - 14th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2014*, pp. 635–640, 2014, doi: 10.1109/CCGrid.2014.77.
- [23] "IoT Gateways – Powering the Industrial Internet of Things." <https://openautomationsoftware.com/open-automation-systems-blog/what-is-an-iiot-gateway/>.
- [24] A. Burlinson, "Why I'm ditching the Arduino software platform," 2012. http://bleaklow.com/2012/02/29/why_im_ditching_the_arduino_software_platform.html.
- [25] Riku Ala-Laurinaho, "Sensor Data Transmission from a Physical Twin to a Digital Twin," p. 105, 2019, [Online]. Available: www.aalto.fi.
- [26] I. C. Society, "IEEE 802.11ah-2016 - IEEE Standard for Information technology," 2016. https://standards.ieee.org/standard/802_11ah-2016.html.
- [27] "IEEE 802.15 WPAN Task Group 1 (TG1)," 2002, [Online]. Available: <https://www.ieee802.org/15/pub/TG1.html>.
- [28] M. C. Sastre, "Bluetooth para IoT, mucho más que tecnología de manos libres," 2016. <https://empresas.blogthinkbig.com/https-empresas-blogthinkbig-com-bluetooth-para-iiot-mucho-mas-que-tecnologia-de-manos-libres/>.
- [29] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67. Elsevier B.V., pp. 104–122, Jul. 04, 2014, doi: 10.1016/j.comnet.2014.03.027.
- [30] A. E. Boualouache, O. Nouali, S. Moussaoui, and A. Derder, "A BLE-based data collection system for IoT," *NTIC 2015 - 2015 1st Int. Conf. New Technol. Inf. Commun. Proceeding*, 2015, doi: 10.1109/NTIC.2015.7368748.
- [31] V. Baños-González, "IEEE 802.11ah. A Technology to Face the IoT Challenge." 2016.
- [32] J. Song *et al.*, "WirelessHART: Applying wireless technology in real-time industrial process control," *Proc. IEEE Real-Time Embed. Technol. Appl. Symp. RTAS*, no. May, pp. 377–386, 2008, doi: 10.1109/RTAS.2008.15.
- [33] J. Horwitz, "The definitive guide to 5G low, mid, and high band speeds | VentureBeat," 2019, [Online]. Available: <https://venturebeat.com/2019/12/10/the-definitive-guide-to-5g-low-mid-and-high-band-speeds/>.
- [34] A. Hodgson, "Cómo el 5G hará que tu hogar sea más inteligente." [https://www.intel.es/content/www/es/es/it-managers/5g-smart-home.html#:~:text=Pero%2C ¿qué otro tipo de,5G para una casa inteligente%3F&text=Los dispositivos domésticos inteligentes son,los 128 bits](https://www.intel.es/content/www/es/es/it-managers/5g-smart-home.html#:~:text=Pero%2C%20¿qué otro tipo de,5G para una casa inteligente%3F&text=Los dispositivos domésticos inteligentes son,los 128 bits)

- del 4G.
- [35] RadioShuttle, “Nota sobre los Factores de Dispersión SF7 y SF12,” [Online]. Available: <https://www.radioshuttle.de/es/medias-es/informaciones-tecnicas/nota-sobre-los-factores-de-dispersion-sf7-y-sf12/>.
 - [36] F. M. Kais Mekki, Eddy Bajic, Frederic Chaxel, “Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT,” 2018.
 - [37] Ejaz, “The IoT - WAN,” 2016. <https://www.netmanias.com/en/post/blog/10702/iot-lora/the-iot-wan>.
 - [38] H. Joe, H. An, W. Wang, W. Lee, and H. Park, “In-band cellular IoT for smart home applications,” in *2017 IEEE International Conference on Consumer Electronics, ICCE 2017*, Mar. 2017, pp. 336–337, doi: 10.1109/ICCE.2017.7889344.
 - [39] T. Yokotani and Y. Sasaki, “Comparison with HTTP and MQTT on required network resources for IoT,” *ICCEREC 2016 - Int. Conf. Control. Electron. Renew. Energy, Commun. 2016, Conf. Proc.*, pp. 1–6, 2017, doi: 10.1109/ICCEREC.2016.7814989.
 - [40] Mozilla, “Gestión de la conexión en HTTP/1.x.” https://developer.mozilla.org/es/docs/Web/HTTP/Connection_management_in_HTTP_1.x.
 - [41] “ISO/IEC 20922:2016 Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1,” 2016. [Online]. Available: <https://www.iso.org/standard/69466.html>.
 - [42] S. R. Christian Esposito, “Performance Assessment of OMG compliant Data Distribution Middleware,” Napoly.
 - [43] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
 - [44] A. Øvern, “Industry 4.0 - Digital Twins and OPC UA,” no. June, p. 173, 2018, [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2561319>.
 - [45] A. Foster, “Messaging Technologies for the Industrial Internet and the Internet of Things,” no. July, p. 26, 2017, [Online]. Available: <http://www.prismtech.com/sites/default/files/documents/Messaging-IIoT-IoT-Whitepaper-v2-1.pdf>.
 - [46] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” *2017 IEEE Int. Symp. Syst. Eng. ISSE 2017 - Proc.*, pp. 1–7, 2017, doi: 10.1109/SysEng.2017.8088251.
 - [47] C. B. Z Shelby, K Hartke, “The Constrained Application Protocol (CoAP),” 2014. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7252>.
 - [48] F. J. Ruiz, “On Premise, servidores y problemas frente a solución Cloud,” 2016. [Online]. Available: <https://blog.dataprius.com/index.php/2016/06/17/on-premise-servidores-problemas-solucion-cloud/>.
 - [49] IBM, “What is edge computing?” https://www.ibm.com/cloud/what-is-edge-computing?cm_mmc=OSocial_Youtube_-_Cloud+and+Data+Platform_Management+and+Platform_-_WW_WW_-_CDAIS+What+is+edge+computing+edge+computing+landing+page+YT+Description&cm_mmca1=000020LH&cm_mmca2=10012473.
 - [50] G. Kranz, “big data storage.”
 - [51] T. N. Khasawneh, M. H. Al-Sahlee, and A. A. Safia, “SQL, NewSQL, and NOSQL Databases: A Comparative Survey,” *2020 11th Int. Conf. Inf. Commun.*

- Syst. ICICS 2020*, pp. 13–21, 2020, doi: 10.1109/ICICS49469.2020.239513.
- [52] ORACLE, “El significado de base de datos relacional.” [Online]. Available: <https://www.oracle.com/es/database/what-is-a-relational-database/>.
 - [53] Félix García Carballeira, “Sistemas de ficheros distribuidos y paralelos.” [Online]. Available: <https://www.arcos.inf.uc3m.es/infodsd/wp-content/uploads/sites/38/2017/02/8.pdf>.
 - [54] F. de A. L. Fuentes, *Sistema Distribuidos*. 2014.
 - [55] Vasanth Makam, “Why Organizations Need to Pay Attention to Cloud Security,” 2020. [Online]. Available: <https://www.red-gate.com/simple-talk/cloud/security-and-compliance/how-organizations-can-optimize-cloud-security/>.
 - [56] IBM, “Data Driven Modeling.” [Online]. Available: <https://www.ibm.com/docs/en/configurepricequote/10.0?topic=overview-data-driven-modeling>.
 - [57] R. A. D. Solomatiné, L. See, “Chapter 2 Data Driven Modelling : Concepts, Approaches and Experiences,” 2017. [Online]. Available: <https://www.semanticscholar.org/paper/Chapter-2-Data-Driven-Modelling-%3A-Concepts-%2C-and-Solomatiné-See/5432246b7dc55a105a687339f404701406987d16>.
 - [58] Analytics India Magazine, “Top 6 Regression Algorithms Used In Data Mining And Their Applications In Industry,” 2017.
 - [59] F. Sanchez, “Unsupervised Learning Algorithms: Explanaiton and Simple code,” 2020. [Online]. Available: <https://fesan818181.medium.com/unsupervised-learning-algorithms-explanaiton-and-simple-code-b7f695a9e2cd>.
 - [60] F. Jaensch, A. Csiszar, C. Scheifele, and A. Verl, “Digital Twins of Manufacturing Systems as a Base for Machine Learning,” *Proc. 2018 25th Int. Conf. Mechatronics Mach. Vis. Pract. M2VIP 2018*, 2019, doi: 10.1109/M2VIP.2018.8600844.
 - [61] Carlos Miskinis, “Normal simulation models versus a digital twin based simulation,” 2018.
 - [62] Walter Schwarz, “How HVAC Simulation Can Improve Safety in Public Spaces,” 2020. <https://www.ansys.com/blog/how-hvac-simulation-can-improve-safety>.
 - [63] Anylogic, “CNH Industrial. Digital Twin of Manufacturing Line: Helping Maintenance Decision-Making.” [Online]. Available: <https://www.anylogic.com/digital-twin-of-a-manufacturing-line-helping-maintenance-decision-making/>.
 - [64] Avantek, “NX CAE para la simulación.”
 - [65] Satoshi, “¿Es Modbus un protocolo IoT?,” 2017. [Online]. Available: <https://www.opiron.com/es-modbus-un-protocolo-iot/>.
 - [66] Qibixx, “Modbus. ¿Tal vez el estándar ‘IoT’?,” 2017. [Online]. Available: <https://www.qibixx.com/es/compania/blogs/news/modbus-tal-vez-el-estandar-iot>.
 - [67] W. K. Stefan Seifried, Günther Gridling, “KNX IPv6: Design issues and proposed architecture,” 2017. [Online]. Available: <https://0-ieeeexplore-ieee-org.llull.uib.es/document/7991951>.
 - [68] “Metodologías de desarrollo de software: ¿qué son?,” *Santander Universidades*, 2020. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.
 - [69] D. G. P. M. Elanie Duperet Cabrera, “Importancia de los repositorios para preservar y recuperar la información,” 2015. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1029-

30192015001000014.

- [70] D. Laballós, “64. ¿Qué es un repositorio de código?,” 2016. [Online]. Available: <https://diegolaballos.com/podcast/64-que-es-un-repositorio-de-codigo/>.
- [71] “Flask.” [Online]. Available: <https://flask.palletsprojects.com/en/2.0.x/>.
- [72] E. Foundation, “paho.mqtt.” [Online]. Available: <https://pypi.org/project/paho-mqtt/>.
- [73] AQR Capital Management, “pandas.” 2008, [Online]. Available: <https://pandas.pydata.org/>.
- [74] MongoDB, “pymongo.” [Online]. Available: <https://pypi.org/project/pymongo/>.