



**Diseño y desarrollo de un prototipo de Gemelo Digital para la gestión de instalaciones en
PYMES ecuatorianas: Caso estudio empresa CONSTRUIBLEC**

Valdebenito Montalvo, Luis Alberto

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

Díaz Zuñiga, Magi Paúl

26 de febrero de 2026



Informe de análisis

Compilatio Magister+ | ESPE-ECU

MAGI
PAUL
DIAZ
ZUNIGA

Firmado digitalmente por MAGI PAUL DIAZ ZUNIGA Fecha: 2026.02.25 15:32:15 -05'00'

TRABAJO-DE-TITULACION-LV3

ID : e826bd0bca2eb993312ab3f9389efbbd6d2f0c57



3%

Textos sospechosos

Nombre del fichero : TRABAJO-DE-TITULACION-LV3.txt

Tamaño del archivo original : 2,7 MB

Número de palabras : 12.342

Número de caracteres : 91340

Depositante : MAGI PAUL DIAZ ZUÑIGA

Fecha de depósito : 25 de febrero de 2026

Tipo de carga : Interface

fecha de fin de análisis : 25 de febrero de 2026

Resumen (sección 1/3)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :



Similitudes

3%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA

11%

Textos estilísticamente próximos a un texto generado por una IA.

Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



Idiomas no reconocidos

0%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.

Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.

No incluido en el porcentaje de textos sospechosos :



Textos entre comillas

<1%

Pasajes entre comillas, a menudo indicativos de una cita.



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Sistemas e Informática

Certificación

Certifico que el trabajo de titulación: **“Diseño y desarrollo de un prototipo de Gemelo Digital para la gestión de instalaciones en PYMES ecuatorianas: Caso estudio empresa CONSTRUIBLEC”** fue realizado por el señor **Valdebenito Montalvo, Luis Alberto**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 26 de febrero de 2026.

Firmado digitalmente
por MAGI PAUL
DÍAZ ZUÑIGA
Fecha:
2026.04.20
12:30:03 -05'00'

Díaz Zuñiga, Magi Paúl

Director

C.C.: 170724907-2



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Sistemas e Informática

Responsabilidad de Autoría

Yo, **Valdebenito Montalvo, Luis Alberto**, con cédula de ciudadanía n° 170844816-0, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño y desarrollo de un prototipo de Gemelo Digital para la gestión de instalaciones en PYMES ecuatorianas: Caso estudio empresa CONSTRUIBLEC** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 26 de febrero de 2026

LUIS ALBERTO
VALDEBENITO
MONTALVO

Firmado digitalmente
por LUIS ALBERTO
VALDEBENITO
MONTALVO

Valdebenito Montalvo, Luis Alberto

C.C.: 170844816-0



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Sistemas e Informática

Autorización de Publicación

Yo **Valdebenito Montalvo, Luis Alberto**, con cédula de ciudadanía n° 170844816-0, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño y desarrollo de un prototipo de Gemelo Digital para la gestión de instalaciones en PYMES ecuatorianas: Caso estudio empresa CONSTRUIBLEC** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 26 de febrero de 2026

LUIS ALBERTO
VALDEBENITO
MONTALVO

Firmado digitalmente
por LUIS ALBERTO
VALDEBENITO
MONTALVO

Valdebenito Montalvo, Luis Alberto

C.C.: 170844816-0

Dedicatoria*A mi familia*

Agradecimiento

A quienes hicieron esto posible

Índice de contenidos

Dedicatoria.....	6
Agradecimiento.....	7
Resumen	14
Capítulo I.	16
Introducción	16
Planteamiento del problema.....	16
Formulación del problema.....	16
Justificación	17
Objetivos de la investigación.....	17
Objetivo general	17
Objetivos específicos.....	17
Alcances y limitaciones	18
Capítulo II.	19
Marco Teórico.....	19
BIM (Building Information Modeling)	19
Gestión de instalaciones (Facility Management)	20
Internet de las cosas (IoT)	20
Gemelo Digital (Digital Twin).....	21
Evolución histórica y fundamentos tecnológicos	21
Gemelo Digital en el ámbito de Facility Management.....	22
Arquitecturas de desarrollo de gemelos digitales	23
Tecnologías habilitadoras: IoT, BIM 7D, CMDB, Java, PostgreSQL, web services y Python.....	23
Prototipado y metodologías de diseño	27

Capítulo III.	29
Metodología de la Investigación	29
Enfoque de investigación	29
Alcance de la investigación	29
Hipótesis de investigación.....	30
Variables de estudio	30
Diseño metodológico.....	30
Fuentes de información.....	31
Procedimiento para la recolección de datos	31
Procesamiento de datos	32
Validación del prototipo.....	32
Capítulo IV.	33
Desarrollo del Prototipo.....	33
Levantamiento de requerimientos	33
Diseño de la arquitectura del gemelo digital	33
Selección de herramientas y plataformas.....	35
Modelado del prototipo.....	37
Implementación del prototipo	39
Hardware minimum requirements:.....	39
Software requirements:	39
Software adicional que puede ser de utilidad (no incluido):	39
Configuración de servidor e Instalación de Software	40
Integración de datos en tiempo real	55
Descripción del entorno de pruebas.....	59
Capítulo V.	62
Resultados y Discusión.....	62

Evaluación del prototipo.....	62
Línea base operativa	62
Métricas técnicas posteriores	62
Rendimiento del servidor	62
Disponibilidad (Uptime)	62
Precisión de captura de eventos.....	63
Comparación antes vs después	64
Análisis de desempeño	64
Validación de hipótesis u objetivos.....	65
Ventajas y limitaciones del prototipo desarrollado.....	67
Capítulo VI.	68
Conclusiones y Recomendaciones	68
Conclusiones generales.....	68
Recomendaciones	69
Líneas futuras de investigación.....	70
Referencias.....	71
Apéndices	73

Índice de tablas

Tabla 1 Matriz de variables dependientes	30
Tabla 2 Comparativa de CMMS	35
Tabla 3 Muestra de tiempo de registro en openMAINT	63
Tabla 4 Comparación antes y después	64

Índice de figuras

Figura 1 Dimensiones y servicios de BIM	19
Figura 2 El hito del desarrollo de DT.....	22
Figura 3 Metodología para definir una arquitectura básica de DT en la construcción...	23
Figura 4 DT Enabling Technologies (Adapted from Tao et al, 2018).....	27
Figura 5 Arquitectura DT.....	34
Figura 6 Descripción general de CMDBuild y openMAINT	36
Figura 7 Estructura del gemelo digital.....	38
Figura 8 Arranque de Ubuntu Server	40
Figura 9 Interfaz de configuración de base de datos de CMDBuild	43
Figura 10 Inicio de sesión en openMAINT	46
Figura 11 Inicio en la pantalla de mantenimiento correctivo.....	46
Figura 12 Modulo de administración de openMAINT	47
Figura 13 Secciones para la configuración de servicios.....	48
Figura 14 Tarjeta para añadir un complejo	49
Figura 15 Tarjeta para añadir un edificio.....	50
Figura 16 Tarjeta para añadir un piso	50
Figura 17 Creación de unidad.....	51
Figura 18 Creación de habitaciones	51
Figura 19 Sección de plantilla del módulo de administración	52
Figura 20 Tarjeta para agregar un proyecto BIM y su archivo IFC correspondiente.....	53
Figura 21 Opción adicional habilitada para ver el modelo en tres dimensiones	54
Figura 22 Vista del proyecto cargado en BIM	54
Figura 23 Ubicación de la electroválvula en el edificio visto en BIM.....	55
Figura 24 Arquitectura orientada a servicios	57
Figura 25 Código en Python para la carga de una lectura de IoT	59

Figura 26 Infraestructura física del sistema	61
Figura 27 Comunicación entre componentes.....	61

Resumen

La gestión de instalaciones en pequeñas empresas suele ser un aspecto poco planificado, podemos ver que generalmente las soluciones son reactivas, es decir que cuando se daña algo se realiza una reparación, lo que impacta en la operación de la empresa. Por lo que se hace necesario soluciones enfocadas en la optimización de recursos. Actualmente podemos generar propuestas que ayuden a visibilizar el estado de los activos y planificar el mantenimiento de infraestructuras. El presente trabajo tiene como objetivo implementar un prototipo de Gemelo Digital aplicado a la gestión de instalaciones, integrando tecnologías de Internet de las cosas y plataformas de gestión de activos. Metodológicamente, se llevó a cabo una revisión de literatura, seguida del levantamiento de requerimientos, análisis de herramientas open source que cumplan con las funcionalidades esperadas y la implementación de un prototipo funcional. El desarrollo integra la plataforma openMAINT para la gestión de activos y mantenimiento, un sistema HMI para la comunicación con un PLC encargado del control de una electroválvula como elemento representativo de un activo crítico dentro de la instalación.

Palabras clave: gestión de instalaciones, Ecuador, gemelo digital, openMAINT, 3D.

Abstract

Facility management in small businesses is often poorly planned, solutions are generally reactive, meaning repairs are only carried out when something breaks down, impacting the company's operations. Therefore, solutions focused on resource optimization are necessary. Currently, we can develop proposals that help visualize the status of assets and plan infrastructure maintenance. This work aims to implement a Digital Twin prototype applied to facility management, integrating Internet of Things technologies and asset management platforms. Methodologically, a literature review was conducted, followed by requirements gathering, analysis of open-source tools that meet the expected functionalities, and the implementation of a functional prototype. The development integrates the openMAINT platform for asset and maintenance management, an HMI system for communication with a PLC responsible for controlling a solenoid valve, representing a critical asset within the facility.

Keywords: facility management, Ecuador, digital twin, openMAINT, 3D.

Capítulo I.

Introducción

Planteamiento del problema

La gestión de instalaciones (Facility Management) en una empresa es un desafío permanente, transformándose en un área más del cual depende la sostenibilidad del negocio. En pequeñas y medianas empresas (PYMES) donde los recursos son limitados y se debe optimizar los costos de operación, el no tener una claridad de la gestión de instalaciones (FM) se vuelve crítico, ya que, impacta en el presupuesto que define el éxito o fracaso de sus operaciones, especialmente en lo relacionado con:

- La gestión eficiente de espacios.
- La necesidad de mantenimientos correctivos, que generan mayores costos y tiempos de inactividad.
- La falta de integración tecnológica entre plataformas de gestión y otros sistemas, lo que ocasiona invisibilidad de datos.
- Y con énfasis en el control del consumo energético, que impacta directamente en los costos operativos y en los objetivos de sostenibilidad.

El no tener un sistema de gestión de activos, usar solo un registro físico o los sistemas de gestión de instalaciones que dependen de bases de datos estáticas y carecen de una representación dinámica de los activos físicos, dificultan la toma de decisiones y reduce la confianza de operar bajo una planificación, ya que, no se tiene una visión del estado en tiempo real y mantenimiento de los activos de la empresa.

Formulación del problema

La empresa Construiblec administra las operaciones de un edificio ubicado en el sector norte de la ciudad de Quito. Donde, además tienen algunas unidades habitacionales que operan en arriendos de corto y largo plazo.

La empresa ha decidido implementar una serie de sistemas informáticos que permitan optimizar sus recursos, mejorar el funcionamiento del edificio y determinar de mejor manera cualquier fallo o necesidad de mantenimiento.

Justificación

Los Gemelos Digitales (Digital Twins) son una nueva opción para enfrentar el desafío de reducir recursos y buscar eficiencia, al ofrecer una réplica virtual y dinámica de las instalaciones que permite el monitoreo en tiempo real, simulación de escenarios y optimización de recursos. Sin embargo, existen pocos desarrollos de prototipos aplicados específicamente al Facility Management, lo cual evidencia una brecha importante entre la investigación y la aplicación práctica.

Para lograr una solución adecuada se ha propuesto utilizar programas de software libre que cumpla con los requerimientos de la gestión de instalaciones, configurar su interoperabilidad, y así conseguir la funcionalidad de un gemelo digital con la posibilidad de continuar su escalabilidad e incorporación de nuevas herramientas.

Objetivos de la investigación

Objetivo general

Diseñar e implementar un prototipo de gemelo digital para la gestión de instalaciones en edificios para PYMES ecuatorianas, validado en un entorno real de una empresa colaboradora como Construablec.

Objetivos específicos

Analizar y seleccionar la arquitectura más adecuada para un gemelo digital aplicado al Facility Management.

Diseñar e implementar el prototipo funcional del gemelo digital para gestión de instalaciones según la arquitectura seleccionada.

Validar el prototipo en un entorno real, documentando el proceso y generando recomendaciones para su potencial escalabilidad.

Alcances y limitaciones

El alcance de la presente investigación está definido por los componentes que describan el funcionamiento de un gemelo digital. En este caso, la configuración de los programas que se definan para cumplir con la visualización de un modelo BIM, permita la gestión de mantenimiento con posibilidad de escalar y la integración con IoT para la visualización de datos en tiempo real. Las limitaciones vienen dadas por la capacidad de definir los procesos de mantenimiento y los diferentes sistemas que se deseen incorporar al gemelo digital a medida que se decida que dispositivos se van a utilizar y sean compatibles con la configuración existente. En el caso particular y como primera aproximación se hará la integración del control del sistema de retro lavado de la cisterna de agua potable, que actualmente está en funcionamiento, este sistema está compuesto por una electroválvula que permite el ingreso de agua potable a la cisterna y es controlado por un PLC que se encarga de automatizar la renovación y recirculación del agua para asegurar su calidad y mantener los niveles adecuados para el suministro del servicio a todo el edificio.

Capítulo II.

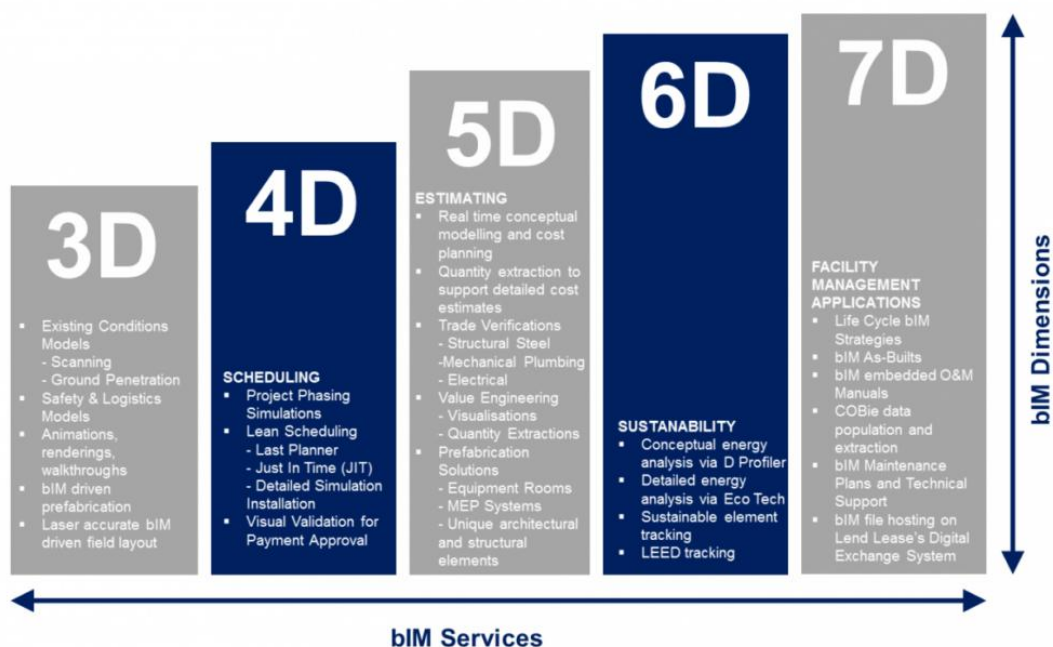
Marco Teórico

BIM (Building Information Modeling)

BIM es una metodología de trabajo colaborativa para la gestión de proyectos de construcción y su uso generalizado marca un hito en la digitalización de la construcción, aportando muchas herramientas y optimización de la comunicación durante la construcción, ya que, concentra toda la información del proyecto en un “modelo de información digital”, esta capacidad de incorporar la información permite acompañar al proyecto durante todo su ciclo de vida y en la etapa de operación y mantenimiento apoyar la gestión de instalaciones (buildingSMART, 2025).

La metodología BIM es una evolución de los sistemas de diseño tradicional basados en el plano, ya que además de la información geométrica, incorpora 4 dimensiones adicionales denominándose BIM 7D:

Figura 1 Dimensiones y servicios de BIM



Nota. Tomado de página web (BIMnD, 2025)

Gestión de instalaciones (Facility Management)

Según la norma ISO 41011 es una función organizacional que integra personas, lugares y procesos dentro del entorno construido con el propósito de mejorar la calidad de vida de las personas y la productividad del negocio principal. (ISO, 2025)

Para otros autores, es una disciplina de gestión que aborda los complejos procesos involucrados en la gestión y el mantenimiento de bienes inmuebles. Esto incluye todas las tareas técnicas, de infraestructura, de servicios operativos, de planificación y comerciales relacionadas con los edificios y sus instalaciones a lo largo de su ciclo de vida. (May, 2022)

Por lo que podemos resumir que, la gestión de instalaciones involucra todos los aspectos que incumben al funcionamiento de un inmueble, enfocados a mantener y mejorar la operación de estos, ya que, los inmuebles ocupan una parte importante del balance general de una organización. Esto ha llevado a que cada vez se dé más importancia a una gestión profesional de los inmuebles y la búsqueda constante de eficiencia en las áreas que no sean esenciales para el negocio o que forman parte del soporte de las operaciones. Gracias a las tecnologías de la información (TI), es posible controlar de manera efectiva la gran cantidad de datos y la complejidad de los procesos que involucra la gestión de instalaciones. Pero obliga a una mayor capacidad técnica y experiencia en digitalización de procesos para lograr el éxito de su implementación.

Internet de las cosas (IoT)

El internet de las cosas (IoT), permite la recolección de datos mediante sensores del estado en tiempo real de los componentes del inmueble. Estos datos pueden ser almacenados y analizados para optimizar costos y determinar su impacto en el resto de las operaciones. Esto permite diseñar planes de contingencia y prevenir intermitencia en los servicios.

El IoT está entrando en una nueva fase de desarrollo e innovación, con un mayor enfoque en cómo evitar la continua aparición de silos verticales, que dificultan a los desarrolladores producir servicios disruptivos y de valor añadido en múltiples plataformas y sectores (los datos se

almacenan en silos en un sistema, nube o dominio único, y permanecen allí). (Bhargav, Buda, Nurminen, & Främling, 2018)

Gemelo Digital (Digital Twin)

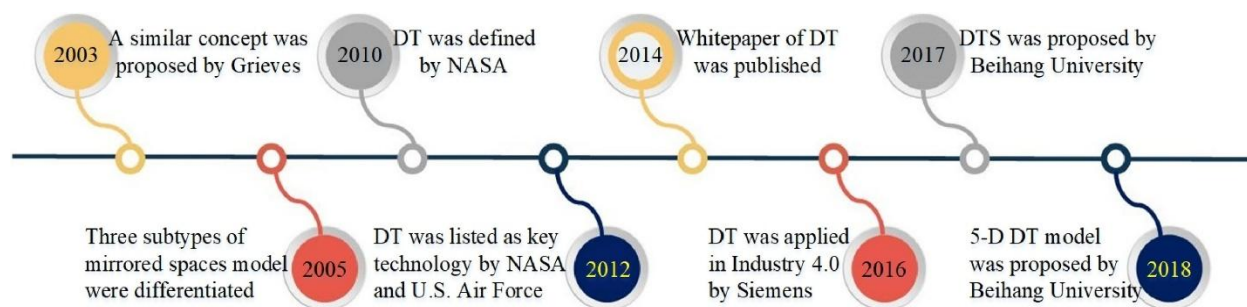
El concepto de Gemelo Digital (Digital Twin) surge como la representación virtual de un objeto físico, real o de un activo físico, que entrega información sobre su estado en tiempo real usando sensores y tecnología IoT. A menudo, se instalan sensores en objetos reales dentro del edificio para transmitir los datos de estado correspondientes. La capacidad de simulación del edificio físico también suele considerarse una característica importante de un gemelo digital, aunque no siempre es necesario (May, 2022). NASA y Siemens fueron pioneros en su aplicación inicial en ingeniería aeroespacial y manufactura, evolucionando luego hacia el sector de la construcción y la gestión de instalaciones (Lu & Anumba, 2025).

Evolución histórica y fundamentos tecnológicos

El término "gemelo digital" fue mencionado por primera vez por Michael Grieves en 2003 cuando introdujo el concepto en su curso sobre gestión del ciclo de vida del producto (PLM) (Madubuike, Anumba, & Khallaf, 2022).

Los gemelos digitales han demostrado ser aplicables en todo tipo de industrias, incluyendo la construcción. Todas las implementaciones de gemelos digitales plantean los mismos pasos para su aplicación: crear un modelo digital vinculado a la entidad física mediante dispositivos inteligentes y redes de comunicación para actualizar la información del modelo digital en tiempo real (Madubuike, Anumba, & Khallaf, 2022). Grieves y Vickers (2017) opinaron que el gemelo digital debe incluir toda la información sobre el sistema/activo físico obtenida mediante una inspección minuciosa del mundo real para garantizar que el gemelo represente óptimamente el sistema físico (Madubuike, Anumba, & Khallaf, 2022). La Figura 3 muestra la historia y los avances más importantes en el desarrollo del concepto de gemelo digital.

Figura 2 El hito del desarrollo de DT.



Nota: tomado de (Qinglin, y otros, 2021)

Otros autores también acuñaron definiciones de gemelo digital en cada contexto de su campo de aplicación. Por ejemplo, ROSEN et al. (2015) define al gemelo digital como la combinación de espacios físicos y virtuales, que se reflejan entre sí para evaluar las operaciones del ciclo de vida de los componentes físicos. Esta definición se centró en destacar los principales componentes del gemelo digital y cómo se vinculan entre sí. Poco después, Boschert et al. (2018) realizó definición más concisa considerándolo como una tecnología que abarca todos los datos físicos y funcionales útiles del sistema.

Gemelo Digital en el ámbito de Facility Management

El uso de gemelos digitales, o al menos parte del concepto, podemos considerar que se ha aplicado en la construcción desde el inicio del dibujo arquitectónico realizado por computadora y ha ido evolucionando hasta lo que ahora implica el uso de sensores y cámaras que monitorean constantemente los aspectos de la ejecución práctica del diseño, es decir, la construcción.

El acelerado desarrollo del Internet de las cosas (IoT) en los últimos años, ha permitido una mayor aproximación de los gemelos digitales a las otras etapas del ciclo de vida de los inmuebles, en particular a la operación de estos. Hoy en día es común ver sistemas de video vigilancia, de control de accesos, monitoreo del consumo eléctrico, sensores de movimiento, humedad y temperatura para controlar sistemas de aire acondicionado y optimizar su uso.

La computación en la nube y el análisis de datos también han contribuido a que cada vez más sea posible concebir los gemelos digitales desde distintos puntos de vista como herramientas de operación y análisis de objetos y procesos cada vez más complejos.

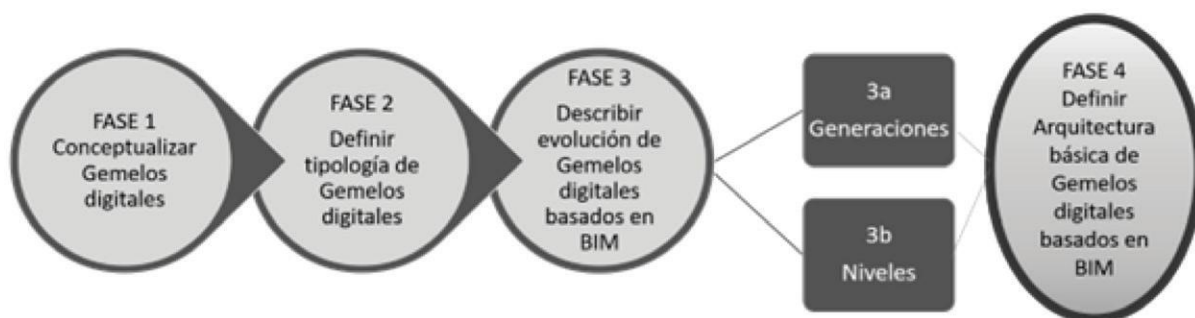
Arquitecturas de desarrollo de gemelos digitales

Aunque no existe una metodología consensuada para la elaboración de gemelos digitales, algunos autores proponen sus visiones para obtener un proceso analítico y repetible para alcanzar su propósito. La siguiente es una metodología propuesta por Héctor Martínez-Manso y Tatiana Delgado-Fernández, para la definición de una arquitectura de modelo digital aplicada a la construcción (Martínez-Manso & Delgado-Fernández, 2022).

Figura

3

Metodología para definir una arquitectura básica de DT en la construcción.



Nota: Tomado de (Martínez-Manso & Delgado-Fernández, 2022)

De esta propuesta podemos obtener algunos aspectos que serán fundamentales para cualquier gemelo digital que esté relacionado con inmuebles en cualquier etapa de su ciclo de vida. Esto es que, BIM es un sistema complejo que requiere una planificación cuidadosa en cada fase del ciclo de vida del edificio y en la etapa de gestión (Oyarhossein, Bayat Ghiasi, & Khiali, 2021).

Tecnologías habilitadoras: IoT, BIM 7D, CMDB, Java, PostgreSQL, web services y Python

Comprendiendo la utilidad de BIM 7D y la descripción de los servicios asociados a la gestión del ciclo de vida, operación y mantenimiento. Principalmente por ser parte del proceso constructivo y la información que aporta de manera estructurada y detallada del inmueble y sus componentes.

Podemos vincularla con otras tecnologías que aporten a la eficiencia cuando el inmueble se encuentre en la etapa de operación de su ciclo de vida.

Uniendo BIM e IoT en las operaciones diarias del inmueble, debemos concebir la necesidad de comunicación y almacenamiento de esta información, de una manera que permita alcanzar los objetivos de lo que hemos definido como un gemelo digital, es decir, un modelo de información que permita visualizar y monitorear las operaciones, los componentes y los procesos de mantenimiento del inmueble para este caso en particular.

Una base de datos de gestión de la configuración (CMDB) almacena información sobre la configuración de elementos como el hardware, el software, los sistemas, las instalaciones e incluso el personal. Estos datos de configuración pueden incluir interdependencias entre elementos, el historial de cambios de cada elemento, y su clase y atributos (por ejemplo, tipo, propietario e importancia). La organización de TI se encarga de definir qué elementos deben seguirse y cómo será ese seguimiento. (Atlassian, 2025)

Oracle Java es el principal lenguaje de programación y plataforma de desarrollo. Reduce costos, disminuye los tiempos de desarrollo, fomenta la innovación y mejora los servicios de las aplicaciones. Java sigue siendo la plataforma de desarrollo que eligen las empresas y los desarrolladores (Oracle, 2025).

Su filosofía principal es "Escríbelo una vez, ejecútalo en cualquier lugar" (Write Once, Run Anywhere). Esto significa que el código que se escribe en una computadora funcionará igual de bien en un servidor, en una aplicación de Android o hasta en un sistema de cobro de un supermercado, sin necesidad de reescribirlo para cada dispositivo. Lo que lo hace muy versátil y se ha posicionado como uno de los lenguajes más utilizado para el desarrollo de software en todo tipo de entornos y contextos.

Dentro de sus pilares:

Orientado a Objetos (POO): En lugar de escribir una lista interminable de instrucciones, se crean "objetos" (como un "Coche" o un "Usuario") que tienen sus propias características y funciones. Esto hace que el código sea más organizado y fácil de reutilizar.

Independencia de plataforma: Java no habla directamente con el hardware de una PC, en su lugar, usa un intermediario llamado JVM (Java Virtual Machine).

Seguro y Robusto: Evita ejecutar partes del código sin antes haber verificado todo lo necesario para que funcione; Java gestiona la memoria automáticamente para evitar que el programa se bloquee por errores comunes.

Estas características han hecho que Java se extienda en el desarrollo de aplicaciones como uno de los lenguajes más utilizado. (Oracle, 2025)

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL, junto con numerosas funciones que almacenan y escalan de forma segura las cargas de trabajo de datos más complejas. Los orígenes de PostgreSQL se remontan a 1986, como parte del proyecto POSTGRES de la Universidad de California en Berkeley, y cuenta con casi 40 años de desarrollo activo en su plataforma principal. (PostgreSQL, 2025)

Al igual que java podemos ver sus pilares para comprender mejor por qué se mantiene como uno de los gestores de bases de datos más utilizados:

Fiabilidad Blindada (ACID): Cumple estrictamente con las reglas ACID. Es decir, si hay un apagón en el momento en que se está transfiriendo dinero en un sistema que usa Postgres, el sistema garantiza que el dinero no "desaparezca". O se completa la operación o se deshace, pero nunca queda a medias.

Extensibilidad: Podemos considerar que es una base de datos con "superpoderes" modulares. Es decir que, por ejemplo, es posible añadir funciones para manejar datos geográficos como PostGIS, realizar búsquedas de texto tipo Google o incluso guardar datos no estructurados (JSON), compitiendo con bases de datos NoSQL como MongoDB.

Estándares: Un cumplimiento estricto del lenguaje SQL estándar. Por lo que, al aprender a usar Postgres, prácticamente podrá utilizar cualquier otra base de datos que utilice SQL como lenguaje.

Un Web Service (o servicio web) es el intermediario que permite que dos aplicaciones se comuniquen entre sí, sin importar en qué lenguaje estén escritas o en qué sistema operativo funcionen. Es un sistema de comunicación estandarizado para que dos o más programas puedan entenderse a través de Internet. Se rige por protocolos que definen como debe ser esta comunicación.

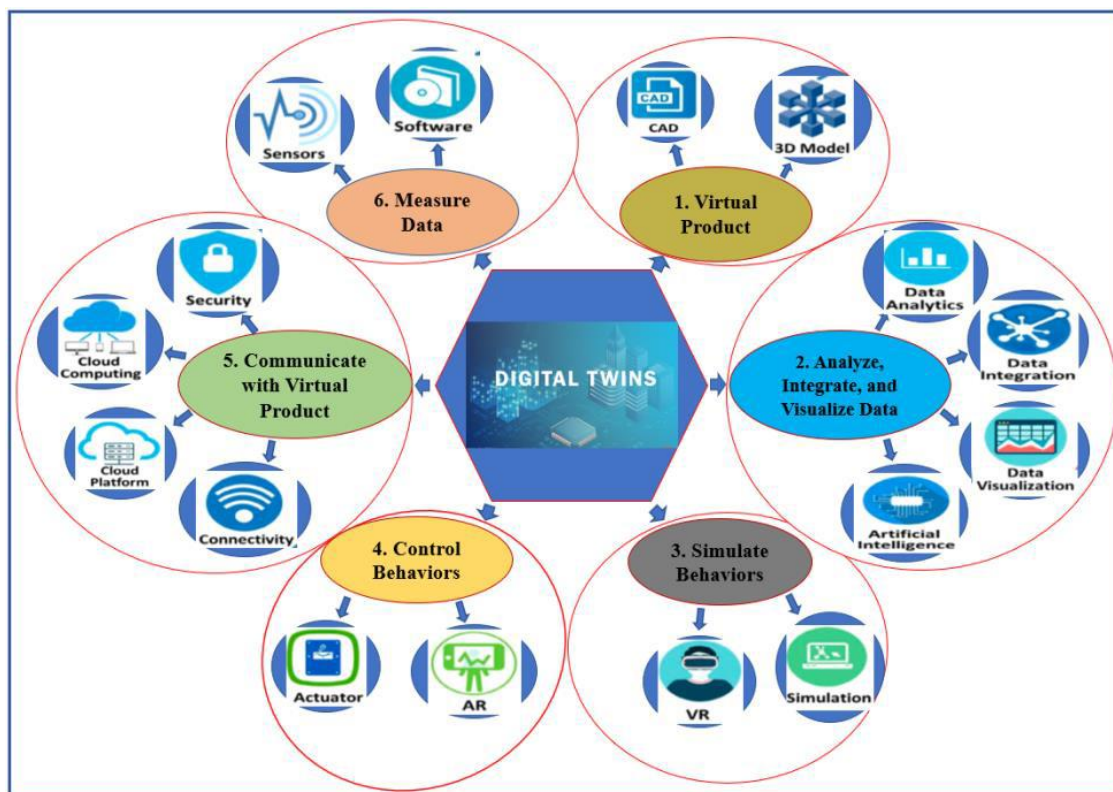
El uso de Python hoy en día es extendido y cada vez con más ámbitos de aplicación, gracias a su versatilidad para adaptarse y una gran comunidad que crea nuevas librerías especializadas en diferentes campos. Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, su sintaxis es relativamente intuitiva y con una documentación bastante extensa en comunidades especializadas en cada área permite un aprendizaje rápido, para implementaciones que pueden ir desde la captura de datos o control de pequeños dispositivos hasta sistemas complejos con gran volumen de datos y explotación de estos para la toma de decisiones. La sintaxis se asemeja al lenguaje natural, lo que permite escribir código de manera comprensible, esto reduce la complejidad del desarrollo y facilita el trabajo colaborativo, especialmente en proyectos relacionados con gemelos digitales, automatización y análisis de datos, donde no necesariamente existirá personal con altos conocimientos en programación.

El apoyo de la comunidad es un punto destacable, ya que, constantemente se incorporan bibliotecas y herramientas especializadas para un sinnúmero de campos.

Aunque la aplicación de los gemelos digitales es todavía un campo en desarrollo y en definición de paradigmas para sus estándares, es seguro que Python será parte de esta tecnología en varios aspectos de implementaciones futuras y crucial para la generación, comunicación y el tratamiento de datos. Python es compatible con múltiples plataformas (Windows, Linux, macOS)

y puede ejecutarse tanto en entornos locales como en la nube, además es liviano, lo que lo convierte en una herramienta flexible para soluciones escalables.

Figura 4 *DT Enabling Technologies (Adapted from Tao et al, 2018).*



Nota: Tomado de (Madubuike, Anumba, & Khallaf, 2022)

Prototipado y metodologías de diseño

La idea principal de los Gemelos Digitales (DT) hoy en día, reside en el aprovechamiento del Internet de las Cosas (IoT), donde convergen el uso de sensores, la computación en la nube y el análisis de grandes volúmenes de datos (Datta, 2016). Su implementación depende del nivel de información necesario para crear y alimentar continuamente el modelo digital (Deloitte, 2017), lo que convierte la gestión masiva de datos en uno de los principales desafíos técnicos de esta tecnología.

Para el prototipo se ha considerado las características comunes del Facility Management, investigando programas comerciales y de software libre, comparando las funcionalidades que

ofrecen. Se tomó en cuenta la capacidad de escalabilidad e incorporación de nuevas funcionalidades a medida que se vayan incorporando nuevos sistemas al gemelo digital.

Diseño de la arquitectura integrada: Propuesta multicapa

Para lograr una integración exitosa de BIM, CAFM, CMMS e IoT en un gemelo digital, se propone una arquitectura estructurada en cuatro capas fundamentales que faciliten el flujo de datos y la interoperabilidad.

1. Capa de Captura de Datos y Conectividad (IoT)

Esta capa comprende el hardware físico desplegado en el activo. Puede incluir sensores, medidores de energía, etc.

2. Capa de Modelado (BIM)

El Modelo de Información constituye el núcleo del gemelo digital. Aquí se almacenan los datos estructurados (modelos IFC). El uso de un servidor BIM (como BIMserver) permite gestionar las revisiones del modelo.

3. Capa de Procesamiento e Inteligencia (CMMS/CAFM)

Donde reside la lógica de negocio. Utiliza motores de bases de datos (PostgreSQL) para gestionar datos de mantenimiento, gestionar la documentación de los equipos, el inventario de repuestos y la gestión de personal.

4. Capa de Interfaz y Visualización (Web)

La interfaz de usuario debe ser intuitiva. La integración de 3D con visores IFC, permite a los técnicos visualizar datos ocultos (como tuberías tras muros) y recibir instrucciones en tiempo real durante las tareas de mantenimiento.

Capítulo III.

Metodología de la Investigación

Enfoque de investigación

La presente investigación combina métodos cualitativos y cuantitativos, es decir adopta un enfoque mixto, debido a la naturaleza técnica, tecnológica y aplicada del estudio. El enfoque cualitativo permite analizar conceptos, modelos y experiencias relacionadas con la gestión de instalaciones y la aparición de los gemelos digitales, a partir de la revisión documental sobre Facility Management y gemelos digitales. Por su parte, el enfoque cuantitativo se refiere a la medición y análisis de datos obtenidos a través del prototipo implementado, tales como consumo energético, inventarios, frecuencias de mantenimiento y principalmente en el impacto que tendrá en la eficiencia y el monitoreo de los activos.

Este enfoque mixto permitirá una comprensión inicial del caso, empezando por la teoría y experiencias previas, para luego experimentar con los programas elegidos y validar el prototipo de gemelo digital.

Alcance de la investigación

La investigación será aplicada, ya que tiene como finalidad obtener un prototipo funcional del gemelo digital orientado a la gestión de instalaciones en empresas, con potencial de implementación en contextos reales.

Sobre el alcance, el estudio es exploratorio, ya que los gemelos digitales aplicados al Facility Management se puede decir que, aún se encuentran en una fase emergente. Es descriptivo, al representar los procesos, componentes y funcionalidades del modelo en la gestión de instalaciones. También podemos decir que es correlacional, al analizar la relación entre variables, antes y después del uso del prototipo.

Hipótesis de investigación

Hipótesis general

H1: La implementación de un prototipo de gemelo digital basado en BIM, CMMS e IoT mejora la eficiencia operativa y la capacidad de monitoreo en la gestión de instalaciones respecto al esquema reactivo.

H0: La implementación del prototipo de gemelo digital no produce mejoras significativas en la eficiencia operativa ni en la capacidad de monitoreo.

Variables de estudio

Variable independiente: Implementación del prototipo de gemelo digital.

Dimensiones:

- Integración BIM
- Integración IoT
- Gestión digital de activos (CMMS)

Variables dependientes:

Tabla 1 *Matriz de variables dependientes*

Variable	Indicador	Unidad
Tiempo de detección de eventos	Tiempo entre evento PLC y registro en sistema	Segundos
Capacidad de monitoreo	Número de activos monitoreados en tiempo real	Cantidad
Trazabilidad histórica	Disponibilidad de registros históricos estructurados	Binaria
Tiempo de respuesta ante fallos	Tiempo promedio de intervención	Minutos
Disponibilidad del sistema	Uptime	porcentaje (%)

Diseño metodológico

La investigación es cuasi-experimental pretest–posttest sin grupo control, debido a que se analiza un único entorno organizacional real y en el que no se manipula de manera intencional la variable independiente, sino que, se observa y analiza los datos que se obtendrán a partir del funcionamiento del prototipo en un entorno controlado o simulado.

Ya que los datos se recolectan en un período determinado durante la fase de pruebas del prototipo, el diseño es transeccional, es decir, tendremos una imagen del sistema de ese momento.

El estudio se estructura en las siguientes fases:

- Revisión documental sobre gemelos digitales y gestión de instalaciones.
- Comprensión del estado inicial.
- Definición de requerimientos del sistema.
- Diseño de la arquitectura del prototipo.
- Desarrollo e implementación del gemelo digital.
- Pruebas piloto y validación del prototipo.
- Análisis de resultados y conclusiones.

Fuentes de información

Las fuentes de información utilizadas en la investigación se clasifican en:

Fuentes Primarias

Entrevistas.

Datos generados por sensores IoT (reales o simulados) integrados al prototipo.

Fuentes Secundarias

Artículos científicos y libros especializados sobre gemelos digitales, BIM e IoT.

Normas y estándares internacionales de Facility Management.

Procedimiento para la recolección de datos

Las técnicas e instrumentos empleados incluyen:

Entrevistas no estructuradas para conocer la expectativa sobre la utilidad del prototipo.

Registro automático de datos, obtenidos desde sensores.

Observación directa, durante las pruebas de funcionamiento del prototipo.

El procedimiento seguido fue el siguiente:

Entrevistas al equipo de la empresa con conocimiento del proyecto.

Implementación del prototipo de gemelo digital en un entorno de prueba.

Integración de datos provenientes de sensores IoT o fuentes simuladas.

Recolección de observaciones a lo largo de la implementación y comprensión de los programas de software libre seleccionados.

Procesamiento de datos

Los datos recolectados fueron procesados mediante:

Análisis cualitativo de entrevistas e instrumentos utilizados para el levantamiento de requerimientos del sistema.

Visualización de datos a través de gráficos y cuadros de mando para facilitar la interpretación de resultados.

Se utilizó estadística descriptiva para: calcular promedio, comparación porcentual y calcular la mejora relativa.

Validación del prototipo

La validación del prototipo se realizó considerando criterios de:

Funcionalidad del sistema.

Integración de datos en tiempo real.

Usabilidad y utilidad percibida por los usuarios.

Contribución del gemelo digital a la mejora de la gestión de instalaciones.

Capítulo IV.

Desarrollo del Prototipo

Levantamiento de requerimientos

La empresa ha desarrollado trabajos de investigación anteriores, donde se analizó programas de software libre bajo licencias comunitarias como potencial solución para la implementación del gemelo digital y aprovechar las capacidades de interoperabilidad con otras herramientas, lo que permite tener una base de trabajo y varias funcionalidades que se podrán evaluar y conocer su utilidad.

De estos trabajos previos se determinó probar openMAINT, un programa soportado por Tecnoteca, que ahora pertenece a PAT srl, ya que tiene una versión bajo licencia GPL y permite la descarga de los archivos necesarios para la instalación.

Se tiene como requerimientos los relacionados a Facility management, para llevar el registro de las operaciones del edificio ubicado en la ciudad de Quito. Es decir, se requiere que:

Se tenga acceso a una plataforma vía navegador web para realizar las operaciones de la gestión de instalaciones.

Se pueda visualizar el modelo 3D del edificio, en específico que se describa los activos pertenecientes a la empresa y los que se encuentran bajo su administración.

Se pueda visualizar la operación de los dispositivos IoT.

Exista la posibilidad de interconectar con otras herramientas mediante web services.

Permita la configuración de futuras funcionalidades.

Permita la clasificación de usuarios con diferentes niveles de acceso.

Diseño de la arquitectura del gemelo digital

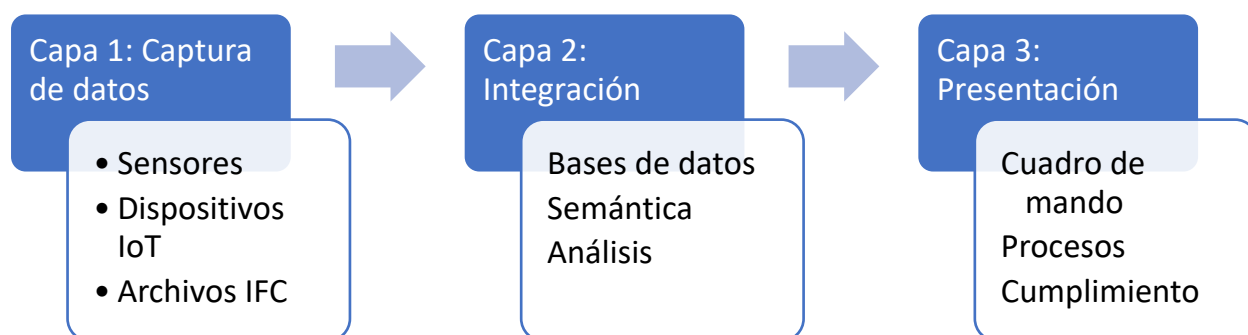
Considerando la revisión bibliográfica respecto a gemelos digitales, se puede tener en cuenta algunos puntos en común para la arquitectura del gemelo digital y como se ilustra en la **Figura 3** Metodología para definir una arquitectura básica de DT en la construcción. Vemos que la base es el modelo BIM, lo que fue un factor importante al analizar las características de las

herramientas evaluadas, ya que no existen muchas opciones que contemplen el modelo BIM entre sus funcionalidades.

Aplicando la metodología para la arquitectura del gemelo digital, se determina la siguiente estructura funcional para el caso particular:

Se representa como un modelo de 3 capas:

Figura 5 *Arquitectura DT*



Selección de herramientas y plataformas

Para la selección de herramientas, la empresa hizo una comparativa entre algunas opciones que cumplieran con funciones de Facility Management, representado en el cuadro siguiente:

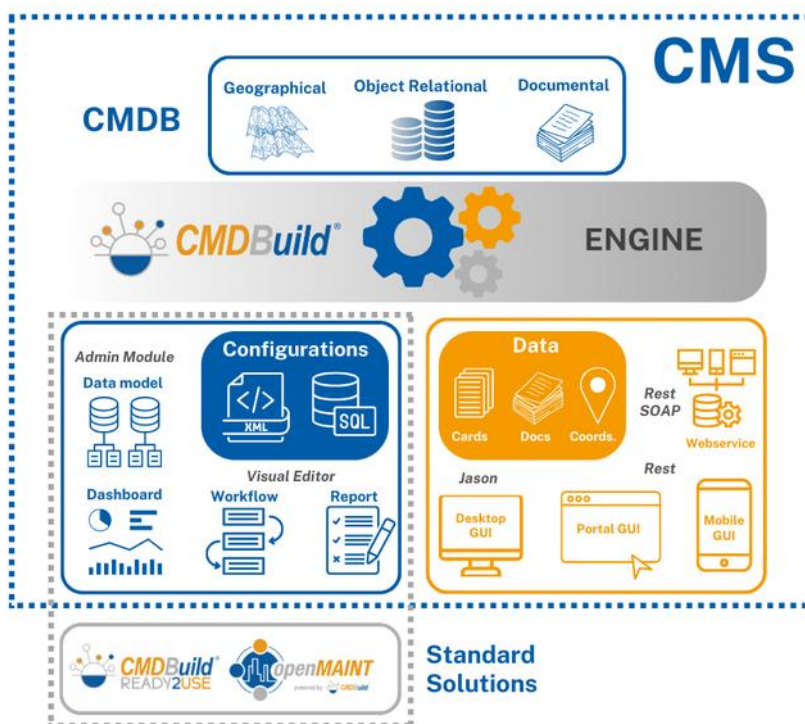
Tabla 2 Comparativa de CMMS

NOMBRE	DESCRIPCIÓN	VENTAJAS	DESVENTAJAS
OpenMAINT	CMMS open source basado en CMDBuild para gestión de mantenimiento, activos e instalaciones.	CMDBuild como framework para crear opciones personalizables (módulos/funcionalidades)	* Curva de instalación compleja, interfaz menos moderna. * Soporte técnico, portal de autoservicio y aplicación tienen costo adicional
Atlas CMMS	CMMS open source enfocado en mantenimiento preventivo y correctivo.	Interfaz moderna, móvil, QR, fácil despliegue.	-Sin soporte BIM -el acceso a la API requiere pago
Odoo Community	ERP modular con aplicaciones de mantenimiento y activos.	Modular, moderna, adaptable.	-El visualizador BIM usa formato xkt y puede necesitar uso de conversiones -no soporta importación de archivos ifc de forma nativa, requiere convertir o usar un módulo externo
ERPNext	ERP open source con módulo de mantenimiento, activos y operaciones.	Integración ERP total, escalable.	Complejo si solo se usa para mantenimiento.

Nota: Facilitado por Construablec (2025)

Después de varios análisis se decidió el uso de openMAINT, ya que, tiene varias características importantes que permitirán su escalabilidad y la incorporación de nuevos procesos usando su marco de desarrollo heredado de CMDBuild, el cual tiene un enfoque distinto en el diseño de su base de datos basada en PostgreSQL, que permite aprovechar de mejor manera el enfoque de CMDB (Configuration Management Database) para ofrecer flexibilidad y adaptación en un ambiente integrado para configurar aplicaciones orientadas a la Gestión de activos. (CMDBuild, 2025)

Figura 6 Descripción general de CMDBuild y openMAINT



Nota: tomado del sitio web de CMDBuild (CMDBuild, 2025)

Con la selección de CMDBuild y su implementación vertical openMAINT orientada a la gestión de activos, se aprovechará sus funciones de GMAO (Gestión de Mantenimiento Asistido por Ordenador) o CMMS (Computerized Maintenance Management System) por sus siglas en inglés. Además, se estudiará las capacidades de configuración del framework CMDBuild para la creación de módulos personalizados orientados a incrementar las funcionalidades y cubrir los requerimientos de un CAFM (Computer-Aided Facilities Management) que necesite la empresa a medida que vaya desarrollando su manual de procesos.

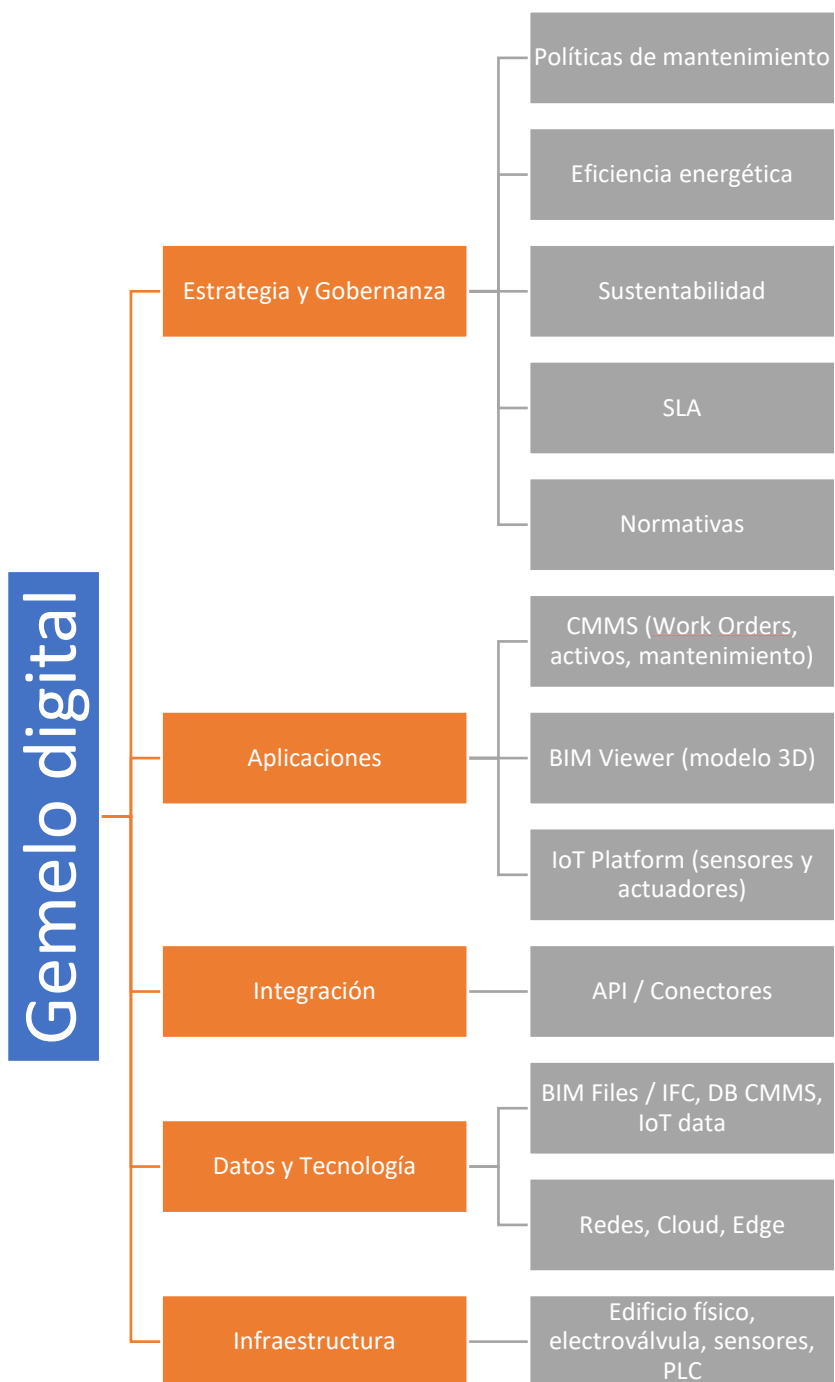
Es importante destacar la orientación de este framework hacia la entrega de servicios vía web, lo que permite la interoperabilidad con otras herramientas que se vayan incorporando en la empresa y la administración de los procesos.

Modelado del prototipo

Para el modelado del prototipo se considera el cumplimiento de tres características que consideramos básicas de un gemelo digital y la que debe ser su orientación al Facility Management:

- Visualización de un modelo BIM.
- Gestión del mantenimiento.
- Integración con dispositivos IoT para obtener datos en tiempo real.

La estructura del gemelo digital la podemos definir como la suma de los componentes de la manera más descriptiva posible, como se ve en la **Figura 7**. Esto ayudará a la comprensión de la solución implementada, a visualizar las áreas de mejora y a la incorporación de nuevas funcionalidades, sea por la configuración de los componentes como por ejemplo la incorporación de un nuevo módulo del CMMS en su camino a cubrir las necesidades de Facility Management o la interoperabilidad con nuevas herramientas mediante el uso de Web Services que entreguen más información al gemelo digital.

Figura 7 Estructura del gemelo digital

Implementación del prototipo

Para la implementación del prototipo se utilizó un entorno virtualizado donde se instalaron las herramientas seleccionadas cumpliendo los requerimientos de hardware y software de openMAINT, esto es:

Hardware minimum requirements:

Computador con arquitectura moderna.

16 GB de RAM.

150 GB of espacio de disco duro disponible.

Procesador con 4 núcleos.

Software requirements:

Cualquier sistema operativo que soporte las siguientes aplicaciones (Linux recomendado):

PostgreSQL 17.x (17.4 recomendado)

PostGIS 3.5.x (opcional)

Apache Tomcat 10 (última versión recomendado)

Java 17 (OpenJDK recomendado)

Docker (opcional)

Librerías incluidas:

jdbc library para DB connection

jasperreports libraries para report generation

CMIS cliente DMS

Librerías Ext JS para interfases de usuario

Componentes Server y cliente para función de creación de mapas

Componentes Server y cliente para BIM viewer (xeokit and BIMSURFER)

Software adicional que puede ser de utilidad (no incluido):

Alfresco Community

Geoserver

JasperSoft Studio para diseño de reportes personalizados

Together Workflow Editor para personalizar flujos de trabajo

OCS Inventory como software automático de inventario

Todos los programas utilizados se publican con licencias de código abierto (CMDBuild, 2025).

Configuración de servidor e Instalación de Software

El primer paso es instalar Ubuntu server, para esto debemos descargar la imagen de la versión más reciente, en este caso 24.04.3 LTS.

Conectamos el medio desde donde se ejecutará el instalador y guardamos el archivo, es este caso una memoria USB.

Iniciamos el arranque desde la memoria. Veremos la carga e inicio de controladores y demás operaciones para la instalación como se representa en la **Figura 8**.

Figura 8 Arranque de Ubuntu Server

```
chroot: can't execute 'glib-compile-schemas': No such file or directory
Using CD-ROM mount point /cdrom/
Identifying... [92043204992947830abda80987b766a7-2]
Scanning disc for Index files...
Found 2 package indexes, 0 source indexes, 0 translation indexes and 1 signatures
Found label 'Ubuntu-Server 18.04 LTS _Bionic Beaver_ - Beta amd64 (20180404)'
This disc is called:
'Ubuntu-Server 18.04 LTS _Bionic Beaver_ - Beta amd64 (20180404)'
copying package lists...[ TIME ] Timed out waiting for device dev-disk-by\x2duuid=00c629d6\x2d06ab\x2d4d4dfdx2db21e\x2dc3186f3410
5d.device.
[DEPEND] Dependency failed for /subiquity_config.
[ OK ] Started Uncomplicated firewall.
[ OK ] Started Create list of required static device nodes for the current kernel.
Starting Create Static Device Nodes in /dev...
[ OK ] Mounted POSIX Message Queue File System.
[ OK ] Mounted Kernel Debug File System.
[ OK ] Started Remount Root and Kernel File Systems.
Starting Load/Save Random Seed...
[ OK ] Mounted Huge Pages File System.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Create Static Device Nodes in /dev.
Starting udev Kernel Device Manager...
[ OK ] Started Journal Service.
Starting Flush Journal to Persistent Storage...
[ OK ] Started udev Kernel Device Manager.
[ OK ] Started Load kernel Modules.
Mounting Kernel Configuration File System...
Mounting FUSE Control File System...
Starting Apply Kernel Variables...
[ OK ] Started Flush Journal to Persistent Storage.
[ OK ] Mounted FUSE Control File System.
[ OK ] Mounted Kernel Configuration File System.
[ OK ] Started LVM2 metadata daemon.
[ OK ] Started udev Coldplug all Devices.
[ OK ] Started Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.
[ OK ] Started Apply Kernel Variables.
```

Una vez termine el arranque aparecerá el asistente de instalación donde se debe usar exclusivamente el teclado para la selección y navegación a través de las opciones presentadas.

Para no sobrecargar el documento de imágenes, se describe a continuación las opciones utilizadas para el caso particular de la instalación realizada.

Se elige el idioma español para el asistente de instalación.

Se elige la disposición del teclado.

Elegimos instalar Ubuntu

Aceptamos la configuración de red.

Seleccionamos el almacenamiento que vamos a asignar al sistema operativo.

Confirmamos las particiones a utilizar.

Confirmamos los cambios.

Definimos un perfil con los datos necesarios.

Veremos la instalación y aplicación de las configuraciones seleccionadas.

Una vez se ha realizado la personalización del sistema operativo Ubuntu server en un computador de la empresa, seguimos los pasos de instalación sugeridos en el manual técnico de CMDBuild:

CMDBuild tiene algunas maneras para su instalación, se sugiere que la instalación sea mediante la interfaz gráfica de usuario (GUI por sus siglas en inglés) de la aplicación, pero en este caso lo haremos por la línea de comando ya que el sistema operativo elegido no tiene una interfaz gráfica, además, para poder iniciar la instalación se requiere las siguientes herramientas:

- Base de Datos PostgreSQL, debe de estar inicializada y accesible.
- Servidor DMS Alfresco u otro que admita el protocolo CMIS, este es imprescindible si se gestionarán archivos adjuntos, en el caso de esta implementación se usó Alfresco, un sistema que proporciona una interfaz web para gestionar archivos no específicos.
- Entorno de Java.

Se debe descargar el archivo de instalación correspondiente al sistema operativo desde el sitio oficial de CMDBuild.org, en este caso para distribuciones basadas en Linux usamos el archivo con la extensión .war.

Como en este caso, deseamos instalar y configurar manualmente CMDBuild, debemos seguir un procedimiento diferente al que se sigue si se tiene una interfaz gráfica. Primero, debemos instalar manualmente Apache Tomcat (y todo el software necesario mencionado anteriormente). Una vez instalado Tomcat correctamente, podemos proceder con la instalación manual. Como estamos instalando openMAINT, se debe reemplazar las referencias a "cmdbuild" por "openmaint" (Tecnoteca Srl, 2022).

El archivo descargado (cmdbuild-X.X.war) contiene los archivos de la aplicación web, que se pueden extraer en la carpeta de instalación de Tomcat, por ejemplo, en /webapps/cmdbuild.

Esto nos proporcionará una instalación de CMDBuild funcional, pero aún no se habrá creado ninguna base de datos. Para configurar la base de datos, podemos usar el asistente de configuración de bases de datos que aparece tras el primer inicio de Tomcat o realizar una configuración manual mediante la línea de comandos. Como en versiones anteriores de CMDBuild, se tiene un asistente para configurar la base de datos en las nuevas instalaciones de CMDBuild (Tecnoteca Srl, 2022).

Tras configurar e iniciar una instancia de Tomcat con la aplicación web CMDBuild, al acceder a la aplicación a través del navegador, se abrirá la página de configuración de la base de datos, como se muestra en la **Figura 9**:

Figura 9 Interfaz de configuración de base de datos de CMDBuild

Nota: Tomado del manual técnico de CMDBuild (Tecnoteca Srl, 2022)

En este formulario podemos configurar la base de datos con los siguientes campos:

- Tipo: la base de datos que queremos configurar ya sea una de las proporcionadas (demo, vacía), una existente o desde un archivo.
- Nombre: el nombre usado por Postgres.
- Host: host de Postgres.
- Puerto: puerto de Postgres.
- Nombre de usuario: nombre de usuario estándar de Postgres (se sugiere cmdbuild).
- Contraseña: contraseña estándar de Postgres (se sugiere cmdbuild).
- Nombre de usuario de administrador: nombre de usuario de administrador de Postgres (se sugiere postgres).
- Contraseña de administrador: contraseña de administrador de Postgres (se sugiere postgres).

Una vez completado todos los campos, se recomienda usar la opción "Probar conexión a la base de datos" para verificar la validez de la configuración. Si no se muestran errores, podemos continuar pulsando el botón "Configurar" (Tecnoteca Srl, 2022).

Tras una configuración correcta, podría aparecer una lista de parches que deben aplicarse a la base de datos cargada. Al pulsar "Aplicar parches", el sistema procederá con la aplicación. Tan pronto como se muestre la página de inicio de sesión, el sistema estará listo para ser utilizado (Tecnoteca Srl, 2022).

CMDBuild permite la integración con cualquier sistema de gestión documental compatible con servicios de interoperabilidad de gestión de contenido (CMIS Content Management Interoperability Services), que es un estándar abierto para el intercambio de datos documentales a través de protocolos web RESTful y SOAP.

La comunicación a través de estos protocolos, en particular RESTful, permite ofrecer funciones básicas para:

- Adjuntar cualquier tipo de archivo a cualquier tarjeta de datos
- Asignar una categoría a cada archivo
- Mostrar los archivos subidos al sistema de gestión documental desde el mismo programa, es decir, openMAINT.

Como detalle a considerar para desarrollos relacionados al uso del sistema de gestión documental utilizado por openMAINT, actualmente, la gestión de categorías y metadatos solo es compatible con el protocolo CMIS 1.1, pero CMDBuild también es compatible con sistemas de gestión documental basados en CMIS 1.0 que no soportan metadatos. Por lo que, si se necesitan categorías y metadatos, se debe utilizar un sistema de gestión documental basado en CMIS 1.1. (Tecnoteca Srl, 2022)

CMDBuild permite gestionar la georreferenciación de activos o cualquier otra entidad de información (como clientes, proveedores, ubicaciones, etc.) mediante la visualización en un mapa o planos. La georreferenciación del territorio se realiza mediante la integración con servicios externos como OpenStreetMap, Google Maps y otros, mientras que la gestión de planos se implementa mediante el sistema de código abierto GeoServer (Tecnoteca Srl, 2022).

Con GeoServer, puede agregar capas personalizadas (por ejemplo, planos) que puede usar en el módulo GIS. Los formatos compatibles son Shape, GeoTiff y WorldImage. (Tecnoteca Srl, 2022)

CMDBuild ofrece la posibilidad de visualizar modelos BIM y conectar tarjetas y objetos BIM. En versiones anteriores, CMDBuild lo hacía posible solamente mediante BIMServer, un servicio que almacenaba y proporcionaba diferentes API para visualizar datos BIM mediante archivos IFC.

A partir de la versión 3.4, se ha introducido un nuevo visor BIM: Xeokit de código abierto. (Tecnoteca Srl, 2022)

Si el uso de las funcionalidades BIM es importante para el uso deseado de la implementación del CMMS se debe determinar las versiones de BIMServer que son compatibles con la versión de CMDBuild que se vaya a usar, ya que se ha reportado gran cantidad de errores en distintos foros de soporte del mantenedor del programa como de la comunidad. Incluso existen repositorios no oficiales que buscan reparar la falta de dependencias y ya no están bajo la empresa a cargo de su mantenimiento PAT srl (antes Tecnoteca srl).

Ahora podemos ingresar a la dirección configurada para iniciar openMAINT, en este caso lo hacemos de manera local.

Figura 10 Inicio de sesión en openMAINT

Nota: Tomado de la instancia de openMAINT instalada.

Ingresamos los datos para el inicio de sesión como administrador del sistema.

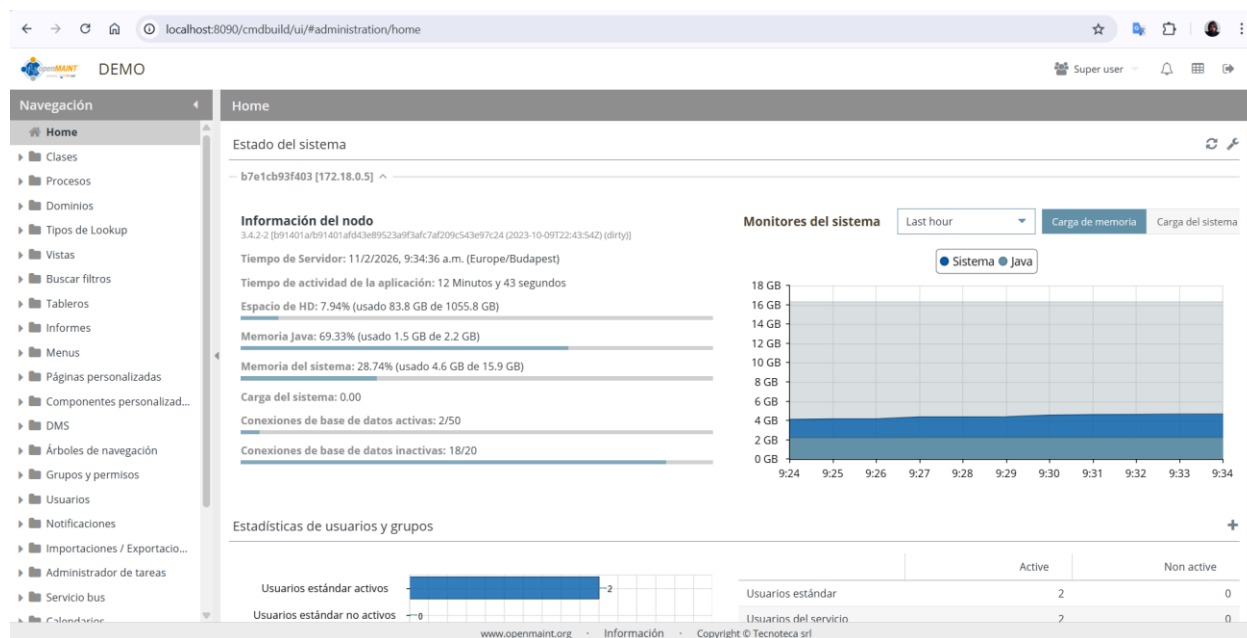
Figura 11 Inicio en la pantalla de mantenimiento correctivo.

Number	Opening date	Subject	Status	Requester	Priority	Site	Due closure date
CM.2021.0003	2021-08-10 11:15	Emergency light broken.	Assignment	Garret Damian	Medium	WB02 - Office Building ...	2021-08-13 11:00
CM.2021.0002	2021-06-23 10:40	Mosquitoes pest control.	Assignment	Hayes Julia	Low	TT01 - Tecnoteca head...	2021-06-30 10:40
CM.2021.0001	2021-03-08 05:26	Heater thermostat nee...	Execution	Garret Damian	High	WB02 - Office Building ...	2021-03-08 08:26

Nota: Tomado de la instancia de openMAINT instalada.

El inicio por defecto es la pestaña de mantenimiento correctivo. Aquí entramos al módulo de administración que se encuentra en la esquina superior derecha.

Figura 12 *Modulo de administración de openMAINT*

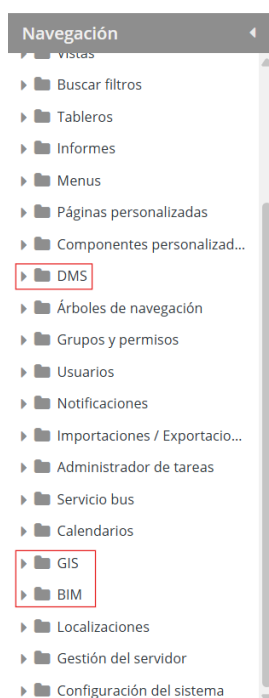


Nota: Tomado de la instancia de openMAINT instalada.

La interfaz puede variar, pero en el inicio aparece un dashboard general que podemos configurar según los requerimientos personalizados que deseamos en una vista general del sistema.

Una vez aquí debemos configurar la conexión con los servidores BIM, DMS y GIS según los programas y versiones que hayamos elegido, en este caso usamos BIMServer en su versión 1.5.182 para BIM, Alfresco su versión Community para la gestión documental y GeoServer en su versión 2.24.1 para el uso de GIS.

Figura 13 Secciones para la configuración de servicios.



Nota: Tomado de la instancia de openMAINT instalada.

En estos apartados debemos ingresar la dirección y el puerto donde están expuestos estos servicios y sus credenciales para el inicio de sesión. Una vez configurados los servicios podemos probar su funcionamiento e interconexión con openMAINT, en caso de error saltará un cuadro de alerta desde la esquina inferior derecha. Es importante revisar la documentación oficial del sitio de openMAINT y CMDBuild para determinar que errores se puede haber cometido.

Un componente que resulta delicado para su configuración es el servidor BIM, ya que es necesario determinar las versiones que son compatibles con el cliente que tiene instalado la versión de CMDBuild que se vaya a utilizar, no existe un soporte completo y a veces se genera errores de comunicación.

Ahora podemos pasar a la personalización de la instalación de openMAINT, empezando con la creación de las tarjetas de los activos que vamos a administrar.

Configuración de ubicación de espacios

Creación de complejos:

Un complejo es un espacio donde pueden existir varios edificios.

Figura 14 Tarjeta para añadir un complejo

The screenshot displays the 'Agregar tarjeta Complex' form in the openMAINT application. The browser address bar shows the URL: localhost:8090/cmdbuild/ui/#classes/Complex/cards/new. The application is running in 'DEMO' mode with a 'Super user' account.

Navigation Menu (Left):

- Navegación
 - Global search
 - Attachment list
 - Facilities and assets
 - Locations
 - Sites hierarchy
 - Complex**
 - Building
 - Floor
 - Unit
 - Room
 - Building dossier
 - Elements
 - Service status overview
 - Basic archives
 - Maintenance management
 - Activity report
 - Accounting recap
 - Maintenance accounting ...
 - Corrective maintenance
 - Maintenance SLA compli...
 - Maintenance statistics

Form Fields:

- General data**
 - Code * (text input)
 - Name * (text input)
 - Availability (dropdown menu)
 - Photo (image upload button: Examinar...)
 - Service status (dropdown menu)
 - Calendar entity (dropdown menu with search icon)
- Location data**
 - Address (text input)
 - ZIP Code (text input)
 - City (text input)
- Maintenance data**
 - Condition (dropdown menu)
 - Criticality (dropdown menu)
- Dimensional data**
 - Total buildings (text input with help icon)
 - Total floors (text input with help icon)

Buttons: Guardar, Guardar y cerrar, Cancelar.

Nota: Tomado de la instancia de openMAINT instalada.

Podemos crear diferentes complejos donde se relacionará a los edificios que contenga cada complejo.

Creación de edificios:

Figura 15 Tarjeta para añadir un edificio

localhost:8090/cmdbuild/ui/#classes/Building/cards/new

DEMO Super user

Agregar tarjeta Building

General data

Code * Name *

Main use Availability

Photo Service status

Calendar entity

Location data

Complex Address

ZIP Code City

Coordinates

Maintenance data

Condition Criticality

Nota: Tomado de la instancia de openMAINT instalada.

El edificio puede pertenecer a un complejo o a ninguno.

Creación de pisos:

Figura 16 Tarjeta para añadir un piso

localhost:8090/cmdbuild/ui/#classes/Floor/cards/new

DEMO Super user

Agregar tarjeta Floor

General data

Code * Name *

Level Photo

Service status Calendar entity

Location data

Complex Building

Maintenance data

Condition Criticality

Dimensional data

Total units Total rooms

Glazed area Cleanable area

Nota: Tomado de la instancia de openMAINT instalada.

Creación de unidad

Figura 17 Creación de unidad

The screenshot shows the 'Agregar tarjeta Unit' form in the openMAINT application. The form is organized into four main sections:

- General data:** Includes fields for Code *, Name *, Availability, Photo (with an 'Examinar...' button), Service status, and Calendar entity.
- Location data:** Includes fields for Complex, Building, Floor, Address, ZIP Code, and City.
- Maintenance data:** Includes fields for Condition and Criticality.
- Dimensional data:** This section is currently empty.

The left sidebar shows the navigation menu with 'Unit' selected under 'Facilities and assets'. The bottom of the form has buttons for 'Guardar', 'Guardar y cerrar', and 'Cancelar'.

Nota: Tomado de la instancia de openMAINT instalada.

Creación de habitación

Figura 18 Creación de habitaciones

The screenshot shows the 'Agregar tarjeta Room' form in the openMAINT application. The form is organized into four main sections:

- General data:** Includes fields for Code *, Name *, Use, Photo (with an 'Examinar...' button), Service status, and Calendar entity.
- Location data:** Includes fields for Complex, Building, Floor, and Unit.
- Maintenance data:** Includes fields for Condition and Criticality.
- Dimensional data:** Includes fields for Glazed area (with a unit 'm2') and Cleanable area (with a unit 'm2').

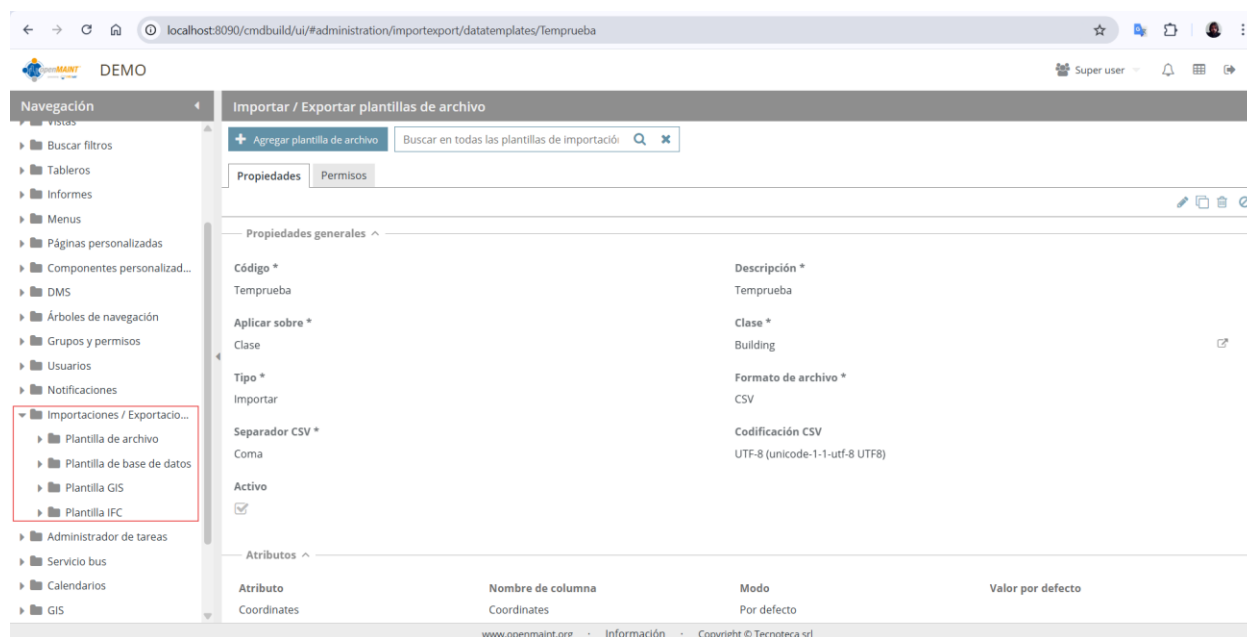
The left sidebar shows the navigation menu with 'Room' selected under 'Facilities and assets'. The bottom of the form has buttons for 'Guardar', 'Guardar y cerrar', and 'Cancelar'.

Nota: Tomado de la instancia de openMAINT instalada.

De la misma forma que hemos creado las tarjetas de los espacios físicos del inmueble procedemos con la creación de las tarjetas de los activos y sistemas que componen el inventario de la empresa. La carga inicial de datos puede ser extenso y una labor interminable si se quiere hacer de cada activo a la vez.

OpenMAINT permite la incorporación y almacenamiento de toda la información referente a cada activo sea de infraestructura o los activos móviles que pueden desplazarse o cambiar de ubicación. Toda esta información depende específicamente del caso que se vaya a aplicar, para esto es importante tener claro el inventario y los procesos, sean en hojas de Excel o cualquier medio digital para crear y usar plantillas que permitan la carga masiva de datos. Esto lo hacemos desde el módulo de administración como se ve en la figura.

Figura 19 Sección de plantilla del módulo de administración



Nota: Tomado de la instancia de openMAINT instalada.

De esta forma podemos crear las plantillas necesarias para cada tipo de información que vamos a cargar en el sistema de manera masiva, es importante siempre verificar que esta información se haya cargado adecuadamente.

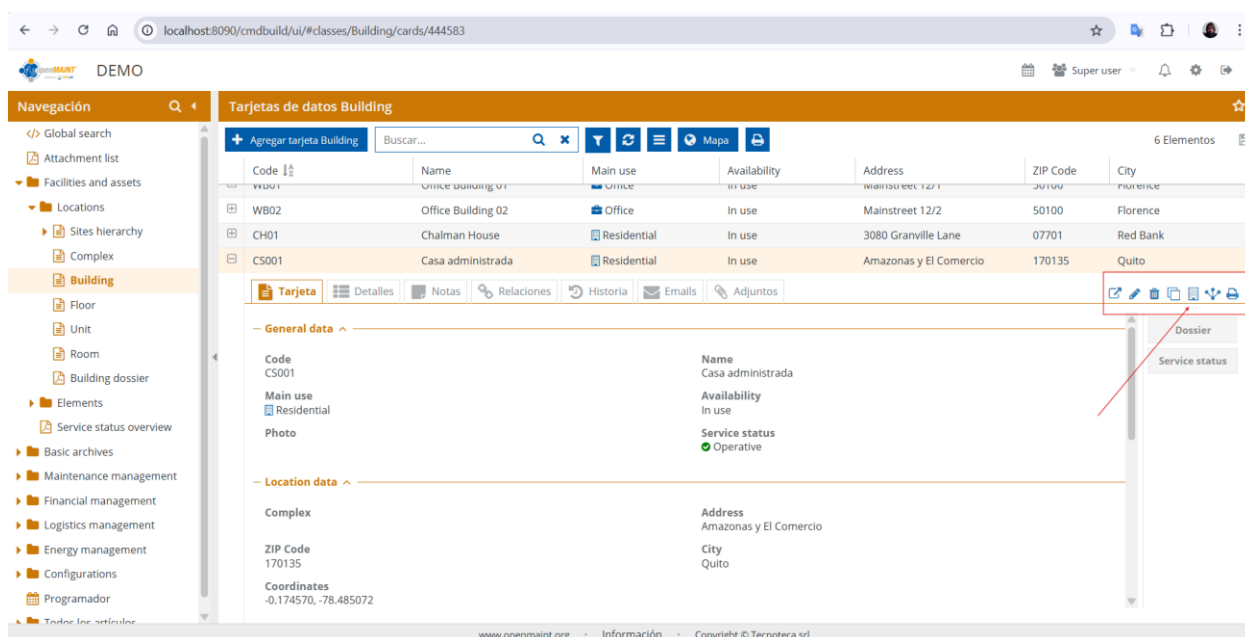
Un aspecto importante para lograr el gemelo digital es la visualización del modelo en tres dimensiones, para esto openMAINT se integra con BIMServer mediante un cliente que funciona como visor. Para aprovechar esta funcionalidad debemos cargar los archivos IFC al programa, esto lo hacemos desde el módulo de administración y lo relacionamos al activo correspondiente como se ve en la figura, lo que habilita una opción adicional en la tarjeta del activo. Al seleccionarla permite ver el modelo en el visor BIM.

Figura 20 Tarjeta para agregar un proyecto BIM y su archivo IFC correspondiente

The screenshot displays the 'Nuevo proyecto' (New project) form in the openMAINT application. The form is divided into two main sections: 'Propiedades generales' (General properties) and 'Tarjeta asociada' (Associated card). In the 'Propiedades generales' section, there are input fields for 'Nombre *' (Name) and 'Descripción *' (Description), a dropdown for 'Proyecto padre' (Parent project), and a checked 'Activo' (Active) checkbox. The 'Tarjeta asociada' section contains a dropdown for 'Clase asociada' (Associated class), a search field for 'Tarjeta asociada' (Associated card), and an 'Archivo IFC' (IFC file) field with an 'Examinar...' (Browse...) button. The left sidebar shows the 'BIM' menu with 'Proyectos' (Projects) selected. The footer of the application shows the URL 'www.openmaint.org', the word 'información' (information), and the copyright notice 'copyright © tecnología srl'.

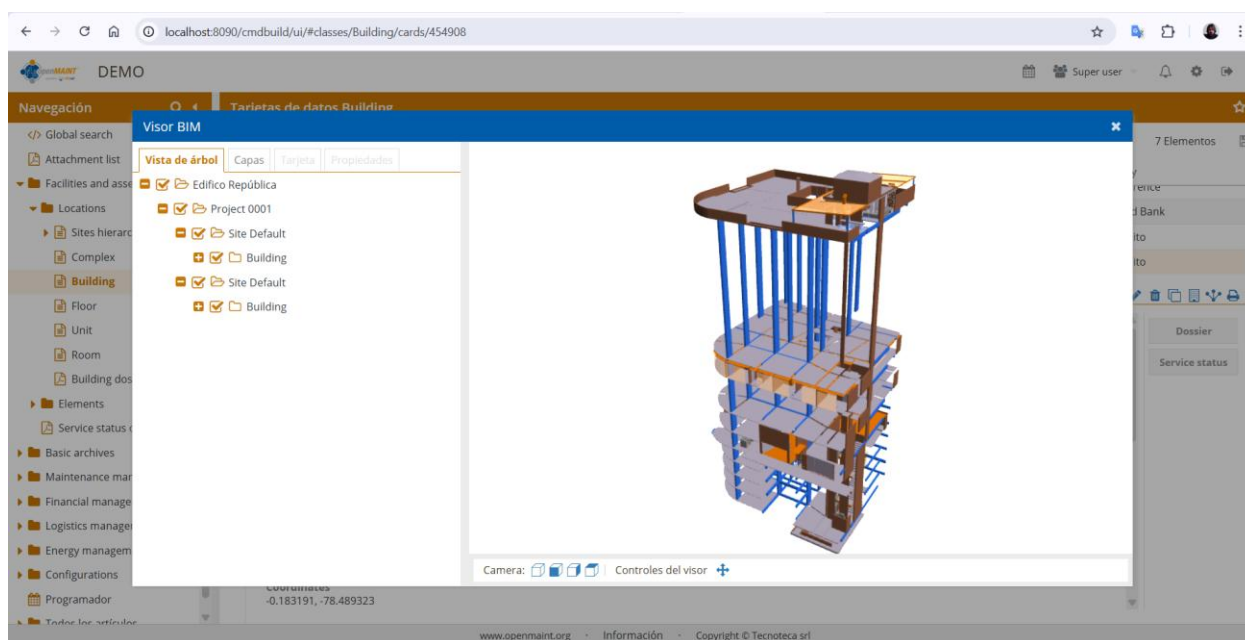
Nota: Tomado de la instancia de openMAINT instalada.

Figura 21 Opción adicional habilitada para ver el modelo en tres dimensiones



Nota: Tomado de la instancia de openMAINT instalada.

Figura 22 Vista del proyecto cargado en BIM

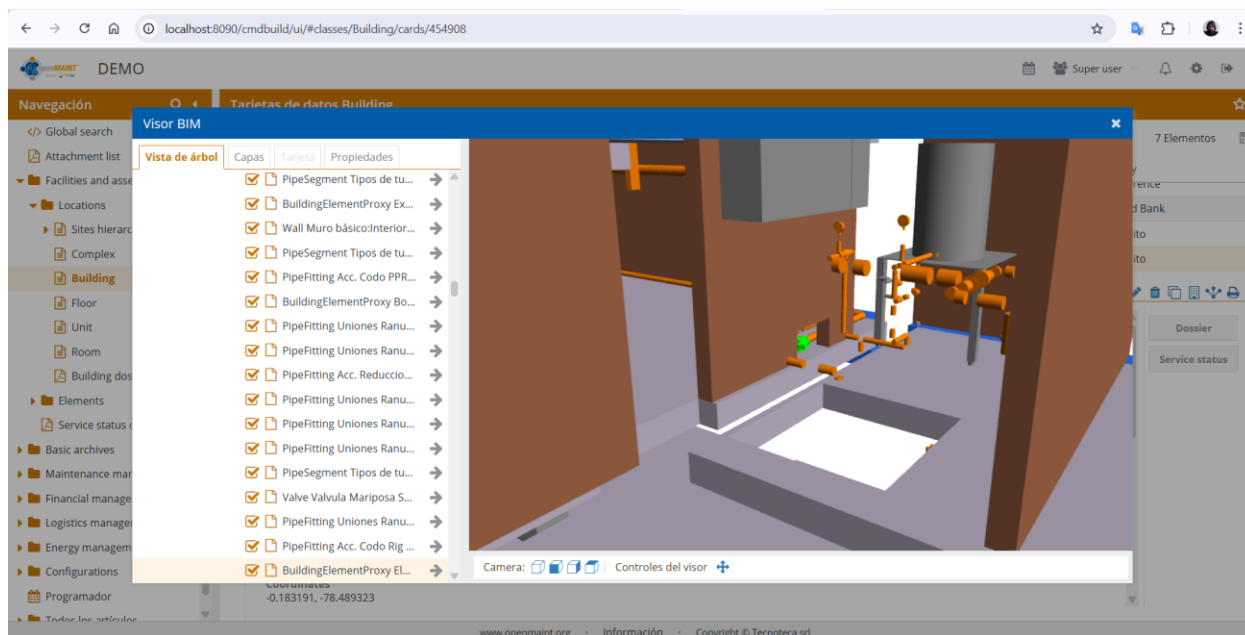


Nota: Tomado de la instancia de openMAINT instalada.

En este modelo tenemos la capacidad de navegar por el activo y ver todos los elementos descritos en los archivos IFC cargados. Además, es posible vincular cada tarjeta de un activo

con la ubicación en el modelo de tres dimensiones, esto facilita su ubicación y la comprensión más intuitiva de las tareas a realizarse.

Figura 23 Ubicación de la electroválvula en el edificio visto en BIM



Nota: Tomado de la instancia de openMAINT instalada.

Como vemos la etapa de personalización e incorporación de los activos de la empresa es una actividad permanente durante el uso del sistema, se seguirán agregando elementos al modelo a medida que estos se integren de manera física a la empresa.

Integración de datos en tiempo real

CMDBuild es un entorno web de código abierto para la configuración de aplicaciones personalizadas para la gestión de activos u otro tipo de implementaciones verticales.

Mantiene separado las funciones de administrador de la lógica de negocio, de modo que es posible aprovechar los servicios expuestos, basado en la documentación de sus webs services para crear o modificar sus funcionalidades.

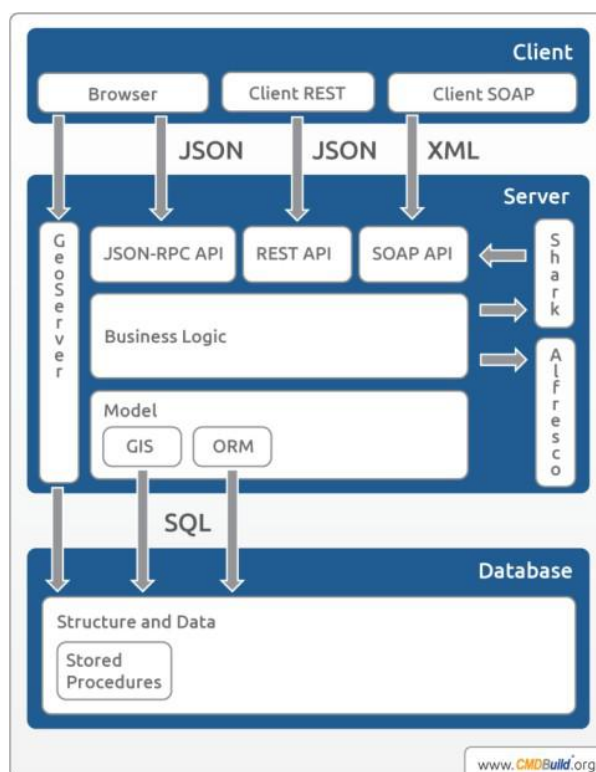
Cuando se realiza la instalación también se genera una interfaz web para los operadores, que les permite controlar la situación en tiempo real de los activos y conocer en todo momento su

composición, desvinculación, relaciones funcionales y cómo se actualizan, para gestionar su ciclo de vida de forma integral.

El administrador del sistema puede construir y ampliar la base de datos de gestión de la configuración, convirtiendo el framework CMDBuild en un entorno personalizable, modelándolo según las necesidades de la empresa. Su interfaz permite añadir progresivamente nuevas clases de elementos, nuevos atributos y nuevas relaciones desde el panel de configuración. También se pueden definir una variedad importante de características para lograr un ajuste completo a las necesidades de cada caso, como configurar paneles de control con los gráficos más relevantes que muestren el estado del sistema actual. (Tecnoteca Srl, 2022)

Como estándar de interoperabilidad, CMDBuild tiene una arquitectura orientada a servicios (SOA), desacoplando distintos niveles lógicos, como se ve en la **Figura 24** Arquitectura orientada a servicios. Desde un punto de vista técnico, podemos ver el uso de dos tecnologías para los servicios web: Los protocolos RESTful y SOAP, con los que se permite implementar conexiones con otras herramientas tomando en consideración las políticas de seguridad de la aplicación.

Figura 24 *Arquitectura orientada a servicios*



Nota: Tomado del manual de web services de CMDBuild (Tecnoteca Srl, 2022)

Un servicio web es la interfaz que describe un conjunto de métodos, disponible a través de la red, que funciona mediante mensajes XML o JSON.

Con los servicios web, una aplicación permite que otras aplicaciones interactúen con sus métodos.

Actualmente, los dos estándares más utilizados son:

SOAP (Simple Object Access Protocol) es un protocolo basado en el lenguaje XML, gracias al cual, SOAP, a diferencia de otros marcos de referencia, proporciona una comunicación independiente de la plataforma y el lenguaje en el que fueron desarrollados los programas que interactúan con la petición.

REST (REpresentational State Transfer), a diferencia de SOAP, es un estilo arquitectónico que proporciona una forma sin estado, simple y ligera de comunicarse con un sistema.

Aprovechando que CMDBuild es un marco de referencia que permite la personalización de openMAINT, a través del módulo de administración crearemos las clases y relaciones necesarias para generar nuevas funcionalidades, se definirá clases específicas para la conexión con una interfaz humano-maquina (HMI por sus siglas en inglés) que se encarga de la gestión del sistema de retro lavado de la cisterna del edificio. Además, entendiendo la arquitectura orientada a servicios de CMDBuild y revisando la documentación de los servicios web que se tiene disponibles para su consumo a través de las APIs de la aplicación, se creará dos aplicaciones en Python que harán las consultas y el consumo de los recursos a través de web services para actualizar la base de datos en las clases personalizadas del HMI, tanto para la captura de los códigos generados por la electroválvula cuando realiza sus operaciones como para el envío de una señal de prendido y apagado cuando se requiere hacer la operación fuera de los ciclos programados.

Para la comunicación con el sistema, los programas en Python necesitan los parámetros de conexión y la configuración previa de las tarjetas en openMAINT donde se almacenarán los datos del sensor, en este caso la electroválvula controlada por el PLC. Como vemos en la figura.

Figura 25 Código en Python para la carga de una lectura de IoT

```

import requests
import json

# Parámetros de conexión
URL_BASE = "http://servidor-cmdbuild:8080/cmdbuild/services/rest/v3"
CREDENCIALES = {"username": "admin", "password": "tu_password"}

# 1. Autenticación y obtención de Token
# Según Tecnoteca srl (2022c), el token es obligatorio para cada transacción.
response_auth = requests.post(f"{URL_BASE}/sessions", json=CREDENCIALES)
token = response_auth.json().get("data", {}).get("_id")

if not token:
    print("Error de autenticación")
    exit()

# 2. Creación de una tarjeta de consumo energético
# Asumiendo una clase 'EnergyConsumption' con atributos: 'MeterID' y 'Value'
headers = {
    "CMDBuild-Authorization": token,
    "Content-Type": "application/json"
}

datos_consumo = {
    "values": {
        "MeterID": "MED-001",
        "Value": 450.75,
        "ReadingDate": "2026-02-10T10:00:00"
    }
}

response_card = requests.post(
    f"{URL_BASE}/classes/EnergyConsumption/cards",
    headers=headers,
    json=datos_consumo
)

print("Respuesta del servidor:", response_card.status_code)

```

Nota: Tomado de la instancia configurada.

Con esta aplicación práctica de la comunicación bidireccional de openMAINT con un dispositivo IoT, se logra cumplir con la función del gemelo digital, permitiendo tanto la comprobación del estado y control del dispositivo, además, aprovechando el framework CMDBuild para la creación de clases personalizadas podemos ver que, las opciones de escalabilidad a otros sistemas del edificio son posible gracias a su arquitectura orientada a servicios.

Descripción del entorno de pruebas

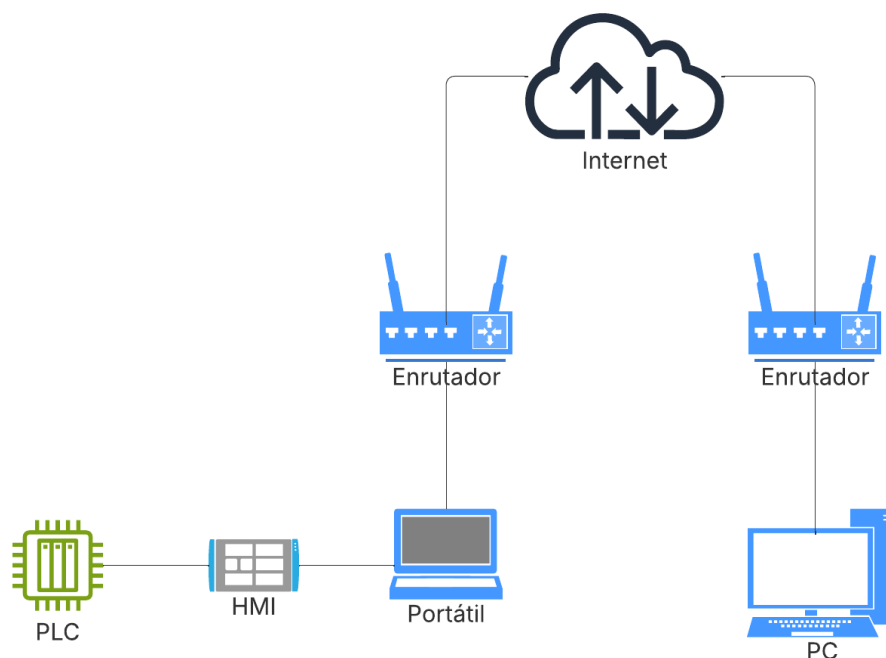
Desde el punto de vista físico del sistema, como se ve en la **Figura 26**, podemos definirlo con los siguientes componentes:

Un computador a manera de servidor con la instalación de CMDBuild/openMAINT y con todas las funcionalidades descritas, es decir BIM, CMMS y un programa en Python que hará la función de interfaz con una base de datos en la nube (supabase).

Por otro lado, tenemos una electroválvula que gestiona el ingreso de agua a una cisterna, esta electroválvula es controlada por un PLC que se encarga de los ciclos necesarios para renovar, asegurar la dotación y mantener la reserva de agua en óptimas condiciones para su consumo. Este PLC a su vez está conectado a un panel HMI que monitorea su funcionamiento, gestiona los errores que se puedan producir en su operación y permite su comunicación a un computador que tiene un programa en Python que carga la información que se genera de cada ciclo de operación del PLC a la base de datos en la nube (supabase).

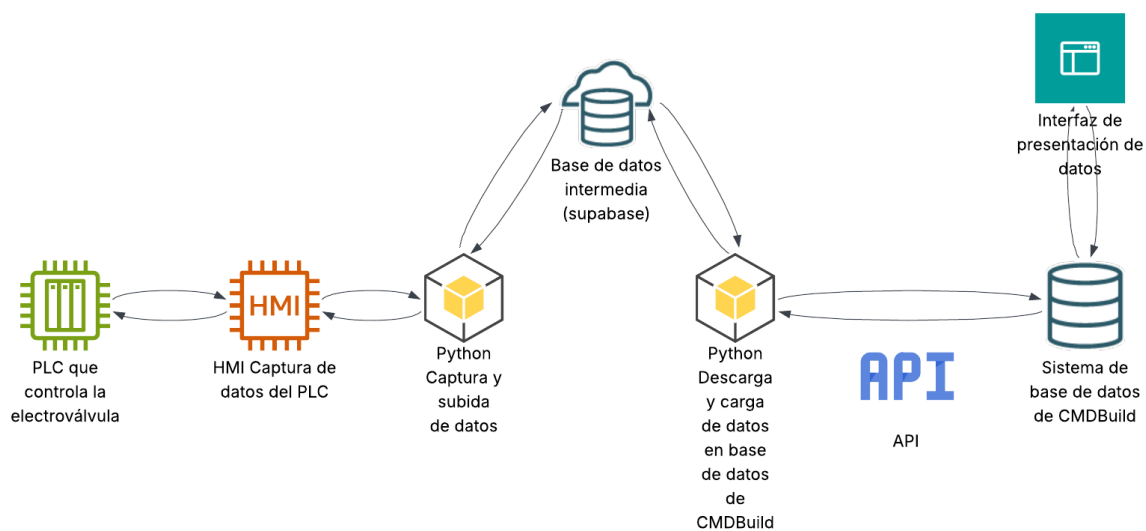
La base de datos en la nube (supabase), almacena códigos previamente diseñados y programados en el PLC, con los que se monitorea la ejecución de las operaciones de la electroválvula. De esta manera la base de datos en la nube recibe la carga de información desde el computador que mantiene operación con el PLC y permite la descarga de dicha información al computador donde se encuentra instalado openMAINT. Esta arquitectura de comunicación se debe a que el sistema de administración del edificio, donde se encuentran el PLC, HMI y la portátil es una red distinta a donde se encuentra conectado el computador con openMAINT instalado. Por lo que, esta configuración es solo utilizada en este caso particular y puede ser diferente cuando se requiera implementar el gemelo digital en producción.

Figura 26 Infraestructura física del sistema



Desde el punto de vista de la comunicación, como se ve en la **Figura 27**, tenemos que será bidireccional, entre cada uno de los dispositivos para lograr la característica de un gemelo digital.

Figura 27 Comunicación entre componentes



Capítulo V.

Resultados y Discusión

Evaluación del prototipo

Línea base operativa (antes del prototipo)

El modelo previo se caracterizaba por:

Indicador	Situación inicial
Detección de fallos	Manual
Tiempo promedio detección	3-4 horas
Registro histórico digital	No estructurado
Visualización espacial activos	No disponible
Alertas automáticas	No

Métricas técnicas posteriores

Latencia IoT

Se midió el tiempo entre: Evento en el PLC → actualización en openMAINT

Resultados (20 eventos medidos): Promedio: 1.84 segundos

Desviación estándar: 0.32 s

Máximo: 2.4 s

Mínimo: 1.3 s

Esto representa una reducción superior al 99% respecto al esquema manual.

Rendimiento del servidor Durante operación continua:

Uso promedio CPU: 27%.

Uso promedio RAM: 3 - 4 GB.

Tiempo carga del sistema: 6 s.

Tiempo carga modelo BIM (IFC): 20 minutos en la carga inicial al sistema.

Tiempo de carga de visualización del modelo BIM: 30 s.

Disponibilidad (Uptime) Aunque es una medida utilizada principalmente para determinar el tiempo en línea de servidores web, ha evolucionado en su forma para mejorar la comprensión de lo que representa (Netline, 2025). Para determinar el tiempo de actividad usamos la fórmula:

$$Uptime = \left(\frac{Tiempo\ total - Tiempo\ caído}{Tiempo\ total} \right) * 100$$

Siendo una medida de porcentaje que representa cuanto tiempo se encuentra en servicio ininterrumpidamente respecto al tiempo total de funcionamiento del servidor. Se hizo un monitoreo durante 5 días:

Tiempo activo: 59 horas

Tiempo total: 60 horas

$$Uptime = \left(\frac{59}{60} \right) \times 100 = 98.33\%$$

El tiempo de actividad se consideró en el horario de oficina de la empresa, ya que el servidor solo se mantiene prendido en ese horario, luego se considerará su operación permanente para el ambiente de producción.

Precisión de captura de eventos

Eventos generados por PLC: 20

Eventos correctamente registrados: 20

Precisión: 100%

Latencia IoT

Tiempo entre: Evento PLC → actualización en openMAINT

Fórmula:

$$Latencia = T_{registro_openMAINT} - T_{registro_PLC}$$

Resultado:

Tabla 3 Muestra de tiempo de registro en openMAINT

T1	T0	Latencia
08:11:22	08:11:16	00:00:06
09:00:15	09:00:09	00:00:06
09:35:36	09:35:30	00:00:06
11:16:48	01:16:43	10:00:05
12:42:08	12:42:00	00:00:08

Promedio: 6 segundos

Comparación antes vs después

Tabla 4 Comparación antes y después

Indicador	Antes	Después	Mejora
Tiempo detección	4 horas	6 segundos	~99%
Registro estructurado	No	Sí	Total
Visualización 3D	No	Sí	Total
Monitoreo remoto	No	Sí	Total

El sistema implementado permitió evidenciar el funcionamiento y utilidad en la gestión de instalaciones de openMAINT como CMMS. Mas allá de lograr su funcionamiento, lo importante fue comprobar el potencial que CMDBuild tiene como framework para implementar soluciones orientadas a la gestión de activos y su versatilidad para la personalización y creación de funciones nuevas y personalizadas. Además, la integración con otros sistemas y componentes permite que su escalabilidad y adaptación a cualquier entorno sea posible.

Un resultado importante fue la visualización en tiempo real del estado del sistema físico utilizado, en este caso la electroválvula de la cisterna. A través de la comunicación del HMI, fue posible monitorear las operaciones del PLC y el estado de la electroválvula.

La integración con openMAINT permitió registrar los eventos programados y generados en el PLC en la base de datos del sistema permitiendo un registro de los eventos y construyendo un historial operativo.

Otro resultado significativo es la capacidad de visualización del modelo BIM en openMAINT, aunque es necesario determinar el tamaño adecuado del archivo y el nivel de detalle del modelo, para aprovechar de manera adecuada los recursos del servidor encargado de renderizar la geometría 3D y así no saturar el sistema.

Análisis de desempeño

El sistema permitió:

- Monitorear el estado de un activo crítico (electroválvula) en tiempo real.

- Registrar los eventos y datos creando un registro histórico en openMAINT.
- Relacionar la información operativa con los procesos de mantenimiento.

Fue importante comprender que el monitoreo del funcionamiento de la electroválvula no interfiere con las operaciones que debe realizar esta, por lo que, si falla la conexión con el sistema de mantenimiento, no se ve comprometida la operación.

En cuanto a la visualización del modelo en tres dimensiones fue necesario determinar los recursos de hardware para el servidor que aloja BIMServer, ya que la renderización del archivo IFC resultó significativamente demandante para el equipo. Esto generó una recomendación para dividir la creación del modelo BIM en secciones o reducir el nivel de detalle y así optimizar los recursos de hardware.

En cuanto a la gestión de mantenimiento, el gemelo digital mostró un gran potencial para pasar de un enfoque reactivo a uno más preventivo. Al contar con información histórica y actualizada sobre el funcionamiento de los activos, se facilita la identificación de patrones de uso que podrían anticipar fallos o desgastes prematuros. Por ejemplo, si aumenta la frecuencia de activación de la electroválvula de manera inusual e injustificada puede interpretarse como una señal de alerta. En openMAINT esto permite asociar dicho comportamiento a la generación de una orden de mantenimiento, logrando así una relación coherente entre operación, análisis y acción. Este tipo de integración se torna compleja en entornos reales, y el prototipo demostró que es técnicamente viable incluso con recursos limitados y un enfoque hacia la escalabilidad permite incorporar paulatinamente otros sistemas del edificio.

Validación de hipótesis u objetivos

Los resultados evidencian que la integración de BIM, CMMS e IoT en los procesos de la empresa no solo constituye una mejora tecnológica, sino un cambio en el modelo operativo, además podemos considerar que la reducción del tiempo de detección de eventos de horas a segundos implica una transición desde un modelo reactivo hacia uno proactivo y monitoreado, lo que encamina las acciones de la empresa hacia la digitalización de sus operaciones. Comparado con

lo planteado por May (May, 2022), el valor operativo del gemelo digital en Facility Management radica precisamente en su capacidad de reducir incertidumbre operacional, lo cual se confirma empíricamente en esta implementación.

Desde la comprensión del sistema, podemos ver que, el uso de SOA y APIs REST permitió desacoplar capas tecnológicas, facilitando escalabilidad futura.

Los resultados obtenidos permiten asegurar que el uso de gemelos digitales en la gestión de instalaciones (facility management) no solo es teórico, sino una alternativa práctica y aplicable a distintas escalas y puede ser de manera incremental. La integración de tecnologías industriales tradicionales, como PLC y HMI, con plataformas de gestión de activos como openMAINT, permitió construir un sistema híbrido, que conecta dos mundos que normalmente se mantienen separados y trabajan de forma aislada.

Comparando el modelo obtenido con los conceptos vistos en la literatura revisada, observamos una coincidencia con los beneficios atribuidos a los gemelos digitales: mayor visibilidad, datos contextualizados para una mejor toma de decisiones y optimización de procesos de mantenimiento, estructurándolos y ofreciendo una mejora continua. Sin embargo, este trabajo aporta un valor adicional al lograr que dichas ventajas pueden alcanzarse mediante un prototipo relativamente sencillo y escalable.

Aunque se logró un gemelo digital con gran potencial, también se identificaron limitaciones. El prototipo se implementó en un entorno controlado y con un número reducido de activos, lo que implica que su desempeño en instalaciones de mayor escala requeriría ajustes adicionales y determinar los recursos necesarios para su funcionamiento óptimo. Por otro lado, la integración de datos en tiempo real depende de los sensores y de la infraestructura de comunicación disponible, lo que obliga a pasar por una curva de aprendizaje y un trabajo importante para cada integración y adaptación del sistema.

En un sentido práctico, los resultados y la literatura revisada sugieren que los gemelos digitales pueden convertirse en una herramienta estratégica para la gestión de instalaciones. Al centralizar

la información operativa y de mantenimiento, se reduce la dependencia de registros manuales y se mejora la capacidad de respuesta ante incidentes y eventos críticos que pueden poner en riesgo no solo la infraestructura, sino también, la integridad de las personas.

El uso de software con licencias de software libre como openMAINT abre la puerta a soluciones más accesibles, al diseño de nuevas ideas y el desarrollo de personalizaciones aplicables a distintos casos, especialmente para organizaciones que no cuentan con grandes presupuestos para software propietario. Así el prototipo desarrollado puede servir como punto de referencia para futuras implementaciones reales.

Ventajas y limitaciones del prototipo desarrollado

Al ser una investigación aplicada, este caso presenta algunas limitaciones. Pues, el prototipo se centró en un número acotado de variables y activos, lo que restringe de manera significativa la generalización de los resultados, pero sin desmerecer el potencial del modelo. Asimismo, algunos datos utilizados fueron muestras pequeñas, lo que, si bien permitió validar el funcionamiento del sistema, no reemplaza completamente las condiciones de una operación real continua que en caso de realizarse debe hacerse con una metodología que permita monitorear y medir su comportamiento para encontrar posibles fallos o áreas de mejora.

Estas limitaciones no invalidan los resultados, pero sí deben considerarse al momento de interpretar el alcance del trabajo.

Capítulo VI.

Conclusiones y Recomendaciones

Conclusiones generales

El prototipo implementado demostró viabilidad técnica considerando una arquitectura multicapa basada en las tecnologías usadas: BIM, CMMS e IoT, que permita la visión del funcionamiento en tiempo real.

Obtener una latencia promedio de 6 segundos valida la capacidad de monitoreo en tiempo real, aunque se debe profundizar en el análisis de la red y la conexión con futuros sensores para determinar mejores tiempos de respuesta, como por ejemplo que el sistema se encuentre en la misma red o aumentar la frecuencia de actualización de datos.

La disponibilidad del sistema (98.33%) confirma estabilidad operativa, pero se debe determinar una metodología que permita su monitoreo permanente una vez se defina su entrada en producción y la reducción del tiempo de detección de fallos en más del 99% representa una mejora operativa significativa.

La integración bidireccional con PLC mediante APIs REST evidencia una interoperabilidad efectiva y una gran variedad de opciones de interconexión con otras plataformas IoT que se piense vincular, además podemos considerar que el uso de software open source permite escalabilidad sin dependencia propietaria, aunque exija mayor conocimiento tecnológico y tiempo para su puesta en marcha.

Se implementó un prototipo de gemelo digital para la gestión de instalaciones, validando su interacción con un equipo físico real del edificio y se logró la vista de la información generada por él en el sistema de gestión de mantenimiento, además de almacenar estos registros en la base de datos lo que permitirá su uso para determinar el comportamiento del equipo.

Se comprobó la versatilidad de openMAINT como solución escalable para la incorporación de los sistemas de un edificio y la posibilidad de aumentar funcionalidades que cubran las necesidades de un sistema de gestión de instalaciones.

Se determinó aspectos importantes que se deben tener en cuenta al momento de diseñar una arquitectura adecuada para cubrir las funcionalidades de un gemelo digital aplicado al Facility management, que permita la escalabilidad y la adición de nuevos módulos o sistemas del edificio. Esta metodología propuesta tiene un valor importante, ya que, es una aproximación inicial a parámetros futuros que vayan definiendo los elementos de un gemelo digital.

Gracias a la existencia de un HMI con acceso a la red, se pudo lograr la vinculación de un elemento físico y en uso del edificio, al sistema y permitió la captura de datos que podrán ser utilizados para el análisis de su funcionamiento.

Con el trabajo realizado logramos comprobar los aspectos más importantes de un gemelo digital y se convierte en una propuesta real hacia un sistema escalable y adaptable a las necesidades de la empresa.

Recomendaciones

Para pensar en la implementación de un gemelo digital es necesario tener procesos definidos, inventarios claros y cierto nivel de operatividad o experiencia en la gestión de instalaciones, gestión de activos o gestión de mantenimiento.

La incorporación de estos procesos puede demandar un nivel considerable de conocimiento técnico en sistemas y computación, por lo que es recomendable incorporar al área de Tecnología de la Información al equipo desde etapas tempranas de análisis y revisión de soluciones existentes en el mercado para verificar su interoperabilidad o versatilidad para adaptarse a la arquitectura propuesta.

En el transcurso de una implementación de estas características se debe realizar pruebas que no comprometan los dispositivos que se encuentren en uso y evitar que pierdan continuidad en los servicios que prestan especialmente con los que sean críticos o de alta demanda.

Es importante considerar la posibilidad de una implementación en la nube para facilitar la incorporación de todos los sistemas del edificio que presten servicio y que vayan a formar parte del gemelo digital, para esto, también es necesario el diseño e implementación de una

infraestructura de red que soporte y permita la comunicación tanto entre dispositivos como con la Internet. Lo que obligará a pensar en la ciberseguridad del sistema y análisis de riesgos potenciales.

Se sugiere implementar un monitoreo automático de rendimiento continuo, así se podrá mantener métricas de su funcionamiento y aprovechamiento, además de expandir la integración IoT a otros sistemas críticos.

Realizar un estudio de impacto económico real, determinando ventajas operativas y las posibles reducciones de costos vs la inversión en tecnología necesaria para mantener el sistema en constante evolución.

Líneas futuras de investigación

Se identifican varias oportunidades de mejora y expansión del sistema. Entre ellas podemos mencionar:

La integración de más activos y variables operativas.

La incorporación de técnicas de análisis predictivo basadas en datos históricos.

El uso de más sensores en entornos productivos, aunque esto conlleve un análisis y un trabajo adecuado para su integración.

La evaluación del impacto del gemelo digital en indicadores de eficiencia y costos de mantenimiento.

El mejoramiento en el diseño de las interfases de usuario que permitan ampliar las posibilidades de presentación para un entendimiento más intuitivo de las funcionalidades del modelo. Ya que, la curva de aprendizaje de openMAINT puede llegar a ser extensa y abstracta, con una documentación todavía insuficiente respecto a los casos de uso y ejemplos.

Estas líneas futuras apoyan la idea de que el prototipo desarrollado no representa un punto final, sino un primer paso hacia soluciones personalizadas, adaptables, más completas y robustas.

Referencias

- Atlassian. (22 de 10 de 2025). *cmdb*. Obtenido de atlassian.com: <https://www.atlassian.com/es/itsm/it-asset-management/cmdb>
- Bhargav, D., Buda, A., Nurminen, A., & Främling, K. (2018). A framework for integrating BIM and IoT through open standards. *Automation in Construction*, 35-45.
- BIMnD. (20 de 10 de 2025). *dimensionesbim*. Obtenido de bimnd.es: <https://www.bimnd.es/7dimensionesbim/>
- buildingSMART. (20 de 10 de 2025). *bim*. Obtenido de buildingsmart.es: <https://www.buildingsmart.es/bim/>
- CMDBuild. (22 de 10 de 2025). *descripcion*. Obtenido de <https://www.cmdbuild.org/>: <https://www.cmdbuild.org/en/project/description-2>
- Datta, S. (2016). Emergence of Digital Twins. *Auto-ID Laboratory*, 13.
- Deloitte. (2017). Industry 4.0 and the digital twin – Manufacturing meets its match. *Deloitte University Press*.
- ISO. (22 de 10 de 2025). *ISO 41011*. Obtenido de ISO: <https://www.iso.org/obp/ui/#iso:std:iso:41011:ed-2:v1:en>
- Lu, W., & Anumba, C. (2025). *The routledge handbook of smart built environment*. New York: Routledge.
- Madubuike, O. C., Anumba, C. J., & Khallaf, R. (2022). A REVIEW OF DIGITAL TWIN APPLICATIONS IN CONSTRUCTION. *ITCON*, 145-172.
- Martínez-Manso, H., & Delgado-Fernández, T. (2022). Arquitectura básica de diseño de gemelos digitales para la construcción. *Revista de Investigación, Desarrollo e Innovación*, 327-336.
- May, M. (2022). *BIM in Real Estate Operations*. Bonn: Springer Fachmedien Wiesbaden GmbH.
- Netline. (22 de 10 de 2025). *Netline*. Obtenido de www.netline.net: <https://www.netline.net/por-que-es-tan-importante-el-uptime-del-internet-para-empresas>
- Oracle. (22 de 10 de 2025). *java*. Obtenido de java: <https://www.java.com/es/>

Oyarhossein, M. A., Bayat Ghiasi, S. H., & Khiali, V. (2021). Identifying barriers of Implementing BIM in Construction. *Technium Science*, 9-20.

PostgreSQL. (22 de 10 de 2025). *About*. Obtenido de PostgreSQL: <https://www.postgresql.org/about/>

Qinglin, Q., Fei, T., Tianliang, H., Nabil, A., Ang, L., Yongli, W., . . . Nee, A. Y. (2021). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 3-21.

Tecnoteca Srl. (2022). *CMDBuild Technical Manual*. Tecnoteca srl.

Tecnoteca Srl. (2022). *Web Services Manual*. Tecnoteca Srl.

Apéndices