

Interreg



Cofinanciado por
la Unión Europea
Cofinanciado pela
União Europeia

España – Portugal

Twin **NavAux**



Arquitectura estándar para la implementación de un gemelo digital de producto

Dic 2024





Contenido

1

Funcionalidades de un gemelo digital

2

Arquitectura básica de un gemelo digital

3

IoT: datos, sensores, protocolos de comunicación

4

Desarrollo de modelos de simulación

5

Almacenamiento de datos

6

Visualización



Funcionalidades de un gemelo digital



La definición más completa del concepto de gemelo digital la da M. Singh cuando afirma que un gemelo digital es un modelo virtual o digital (simulación) dinámico de un objeto real (una máquina, un componente, un proceso, etc.). Este modelo debe representar el estado exacto de su gemelo físico (real) en cualquier momento y ante cualquier situación a partir del intercambio de datos reales (los datos fluyen del real al virtual y viceversa) y conservando los datos históricos, al mismo tiempo. Cualquier cambio que se produzca en una de las partes del gemelo, tanto la real como la virtual, debe reflejarse en el otro, de forma que la réplica real-virtual sea exacta.

Existen, sin embargo, gemelos digitales que toman esta definición de forma parcial, dependiendo del momento de creación del gemelo o de la funcionalidad buscada. De aquí que se pueda hablar de gemelos digitales según el nivel de integración o automatización. La versión más básica sería un modelo digital, donde los datos del objeto físico se trasladan al modelo virtual manualmente por lo que cualquier cambio en la parte real no se refleja automáticamente en la parte virtual. El siguiente nivel sería la sombra digital donde los datos fluyen de forma autónoma desde el objeto real a su modelo virtual pero no al revés. En este caso cualquier cambio en la parte real se refleja de forma automática en la parte virtual, pero no en sentido contrario. Y el nivel más alto sería el propio gemelo digital, tal y como aparece en la definición anterior.

Por otro lado, se pueden realizar gemelos (o sombras) digitales de partes de un sistema e ir integrando esos gemelos parciales para obtener el gemelo del sistema completo (gemelo de sistema). Y si integramos varios sistemas tendríamos un gemelo de sistema de sistemas (SoS).

Un gemelo digital y no su nivel de automatización vendrá determinado por la aplicación que se le quiera dar.

Los gemelos se pueden emplear para:

- Monitorización de un equipo, sistema o proceso. Dado que replica fielmente el estado operativo del sistema físico al que representa, y recopila datos continuos de sensores y otros dispositivos, este monitoreo continuo le permite identificar cambios sutiles en el rendimiento o parámetros críticos.
- Mantenimiento predictivo de equipos y sistemas. El gemelo digital sugiere planes de mantenimiento personalizados, reduciendo costos asociados a intervenciones innecesarias y minimizando el tiempo de inactividad.
- Testeo de prototipos y simulación de escenarios. Probando virtualmente diferentes condiciones operativas e identificando los impactos en el rendimiento del sistema. Esto permite predecir el comportamiento de los equipos en situaciones de estrés e identificar los componentes que tienen mayor probabilidad de fallar.

Todo ello con la posibilidad de interconexión del gemelo digital con los sistemas de gestión de mantenimiento (CMMS) que automatizan la emisión de alertas y permiten dar una respuesta más rápida y eficiente.

En el caso del gemelo digital para monitorización es necesario crear el modelo virtual de los que se quiere replicar de la realidad y que exista un flujo de datos entre la parte real y la parte virtual en tiempo real. De no ser así, no es posible realizar la monitorización en tiempo real. El modelo virtual debe estar basado en un modelo de simulación que replique el comportamiento del equipo/sistema real de forma fidedigna. Este modelo de simulación puede ser 1D o algoritmos de machine Learning. Para la transferencia de datos entre la parte real y la virtual es necesaria la colocación de sensores que transmitan las variables físicas a controlar. Los datos procedentes de los sensores en tiempo real se comparan con los datos obtenidos de la simulación, lo que permite garantizar el correcto funcionamiento del equipo/sistema o bien detectar cualquier anomalía en ese funcionamiento. El almacenamiento continuo de estos datos permitirá generar datos históricos que pueden ser útiles en posteriores análisis.

Estos datasets almacenados con datos de las variables físicas medidas con los sensores, son la herramienta de trabajo de los gemelos digitales de mantenimiento predictivo.

Estos datasets permiten entrenar los modelos de simulación en todas las casuísticas del funcionamiento de un equipo/sistema, de forma que dadas unas condiciones de contorno iniciales sea capaz de predecir los fallos que va a tener el equipo después de unas determinadas horas de trabajo. Los modelos de simulación se pueden ejecutar en tiempo real (monitorización) o bien ejecutarse para un determinado período de tiempo, donde el step de tiempo se acelera con respecto al tiempo real. En estos gemelos se debe actualizar el estado del equipo/sistema virtual a medida que cambie el estado del equipo/sistema real, es decir, la condición de deterioro del equipo o las tareas de mantenimiento realizadas que restauran la condición del equipo. Esto es, el propio gemelo requiere de un mantenimiento específico.

En el caso de los gemelos que se crean en el momento del diseño de un equipo, la parte de sensórica (IoT) no es posible aplicarla puesto que aún no existe la parte real. En estos casos se parte de un modelo CAD 3D y se crea un modelo de simulación basado en un modelo matemático. Estos modelos incorporan ecuaciones de la física, la termodinámica, la dinámica, etc., capaces de reproducir el comportamiento mecánico de un equipo. De esta forma, con la parte virtual del gemelo se pueden simular diferentes escenarios de uso para ese equipo y conseguir así una optimización del diseño. En estos gemelos es útil disponer de datasets históricos de equipos similares para ajustar los modelos de simulación.



Arquitectura básica de un gemelo digital

2



Entendemos por arquitectura de un gemelo digital los diferentes componentes que integrarán el despliegue de dicho gemelo. Estos componentes pueden variar dependiendo de las funcionalidades o las necesidades que se presenten en ese despliegue.

Los componentes a combinar, en cualquier caso, serían los siguientes:

1. Sensórica - IoT
2. Entroncos de Simulación
3. Entornos de ejecución: plataformas IoT, cloud platforms, Edge computing
4. Almacenamiento de datos
5. Analítica (Big data, ML/DL)
6. Visualización (RA, HMI)
7. Ciberseguridad

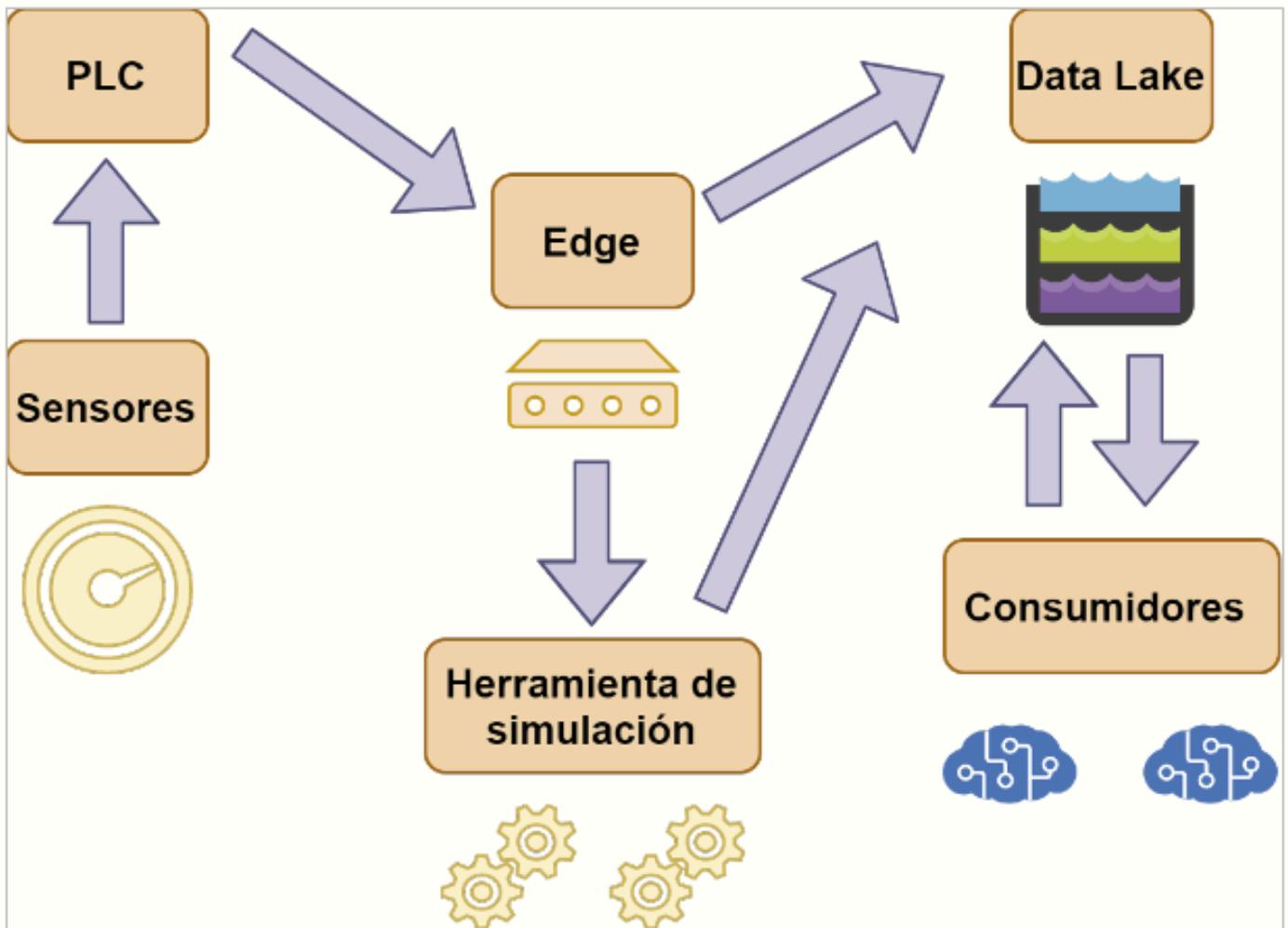


Figura 1. Esquema básico de una arquitectura de gemelo digital

La arquitectura de un gemelo digital es fundamentalmente la arquitectura de los modelos de simulación en los que se basa. Por eso es necesario comenzar por definir el dominio que se va a modelar, como procesos industriales, sistemas de equipos, etc. La definición del dominio ayuda a delinear los límites del sistema, las entidades involucradas y los procesos o comportamientos subyacentes que se van a simular.

El sistema se puede representar mediante ecuaciones diferenciales, modelos estadísticos, modelos de aprendizaje automático u otras formulaciones matemáticas. La elección de la herramienta o el entorno de simulación depende de la naturaleza del sistema (determinista, probabilístico, híbrido). Además, se pueden utilizar solucionadores numéricos (solucionadores de EDO, algoritmos de optimización, etc.) para calcular el comportamiento del sistema a lo largo del tiempo.

El siguiente paso es ajustar la entrada y la salida del modelo. Por lo general, los modelos se basan en varias fuentes de datos de entrada, como datos de sensores, datos históricos, parámetros y bases de datos externas. Los resultados de los modelos de simulación se envían a los usuarios u otros sistemas para su análisis y toma de decisiones. Estos resultados pueden ser métricas de rendimiento, predicciones o visualizaciones.

Las arquitecturas de simulación modernas suelen depender de flujos de datos en tiempo real o conjuntos de datos históricos para proporcionar entradas precisas y dinámicas. La integración con dispositivos IoT/IIoT, sensores o bases de datos permite que el sistema se mantenga actualizado y responda a las condiciones cambiantes del mundo real. La capa de integración de datos se encarga de la limpieza de datos, el formato y la comunicación entre el modelo de simulación y los sistemas externos.

Los motores de simulación son el componente central que impulsa la ejecución del modelo. Procesa los datos de entrada, ejecuta los algoritmos subyacentes y genera resultados. El motor debe gestionar la progresión temporal (discreta, continua o híbrida), la sincronización de eventos o estados del sistema y la gestión de recursos (CPU, memoria, etc.). Estos motores de simulación pueden ser en tiempo real (para aplicaciones como sistemas de control) o no en tiempo real (para fines analíticos).

Las simulaciones pueden requerir un uso intensivo de recursos computacionales, lo que requiere la orquestación de recursos en varias máquinas o clústeres. La paralelización divide las tareas entre las CPU/GPU para ejecutar partes del modelo simultáneamente, lo que mejora el rendimiento, especialmente para simulaciones a gran escala (por ejemplo, el modelo de un sistema completo). Kubernetes y marcos de orquestación similares pueden gestionar modelos de simulación distribuidos en contenedores.

Algunas arquitecturas implican modelos híbridos que combinan diferentes enfoques de modelado (por ejemplo, discretos y continuos) o integran múltiples modelos de simulación para representar conjuntos complejos de sistemas. Los marcos de co-simulación se utilizan para orquestar la interacción entre diferentes modelos (por ejemplo, simulación de sistemas mecánicos y eléctricos).

Finalmente, los algoritmos de control se pueden utilizar dentro de la arquitectura del modelo de simulación para habilitar mecanismos de retroalimentación que ajusten el comportamiento del modelo en función de los resultados. Y también, incluir capas de optimización que permitan al modelo buscar la mejor solución dentro de restricciones definidas (por ejemplo: minimizar el uso de energía o maximizar el rendimiento).

Podemos clasificar la arquitectura de los modelos en función de cómo se conectan entre sí. Podemos hablar de arquitectura monolítica, modular, distribuída, federada o híbrida. En la arquitectura monolítica, todos los componentes de la simulación están contenidos en un único modelo o sistema. Es simple, pero carece de flexibilidad y escalabilidad. En una arquitectura modular, la simulación se divide en módulos independientes y reutilizables (los sistemas grandes se pueden dividir en subsistemas), lo que permite flexibilidad y escalabilidad. En la arquitectura distribuida, el modelo se distribuye entre varios sistemas o procesadores, lo que permite simulaciones a gran escala y de alto rendimiento. En la arquitectura federada, se ejecutan varios modelos en paralelo, a menudo creados en diferentes herramientas o entornos. Un administrador de federación central coordina las interacciones entre ellos. Una arquitectura híbrida combina diferentes técnicas o enfoques de simulación (por ejemplo, algoritmos de aprendizaje automático y simulaciones continuas) en un modelo. Existen múltiples estándares disponibles para la implementación de todas estas arquitecturas, como FMI/FMU (Functional Mockup Interface/Unit) o HLA (High Level Architecture).

Las arquitecturas que tienen que manejar un flujo de datos deben incluir un conjunto de servicios y herramientas:

- Almacenamiento y procesamiento de datos.
- Ingesta de datos: recopilación de datos de varias fuentes, ya sea en lotes o en streaming.
- Orquestación de datos: gestión de la programación del flujo de trabajo y las dependencias.
- Transformación de datos: aplicación de canalizaciones de transformación de datos para extraer valor de los mismos, por ejemplo, mediante la aplicación de lógica empresarial, agregaciones y enriquecimientos.
- Inteligencia empresarial y análisis: consulta y generación de informes de datos para comunicar de forma eficaz el valor extraído.
- Calidad de datos: coherencia de los datos, la latencia y los errores de los datos.
- Gestión de acceso: control de acceso a los datos y gestión de permisos.

Los entornos basados en la nube proporcionan plataformas flexibles y escalables para el despliegue de modelos, con la ventaja de contar con recursos informáticos distribuidos. Este es el caso de AWS (Amazon Web Services) que ofrece servicios EC2 a demanda para recursos informáticos, AWS Batch o la ejecución de simulaciones a gran escala, y AWS S3 para el almacenamiento. Las simulaciones se pueden ejecutar en paralelo utilizando la potencia de la computación en la nube, lo que resulta útil para simulaciones de gran capacidad computacional. Un caso similar es Microsoft Azure, que proporciona recursos informáticos como Azure Virtual Machines y Azure Batch para el despliegue de modelos de simulación. Los gemelos digitales de Azure son especialmente adecuados para crear y ejecutar simulaciones de entornos físicos, lo que permite simulaciones de ecosistemas IoT y entornos inteligentes. El otro gran proveedor de este entorno de nube es Google. Google Cloud Platform (GCP) ofrece Google Compute Engine para ejecutar simulaciones personalizadas, Google Kubernetes Engine (GKE) para modelos de simulación en contenedores y Google Cloud AI para ejecutar simulaciones impulsadas por IA.

Para aplicaciones en tiempo real o que requieren una latencia mínima, las plataformas de computación en el edge permiten que los modelos se ejecuten cerca de donde se generan los datos, lo que mejora la eficiencia. Para este caso, existen diferentes entornos disponibles como Azure IoT Edge, AWS IoT Greengrass o NVIDIA Jetson. Azure permite ejecutar simulaciones directamente en dispositivos edge, procesando los datos cerca de donde se generan. Esto es útil cuando la toma de decisiones en tiempo real basada en simulaciones es esencial. De la misma manera, AWS IoT Greengrass permite ejecutar simulaciones en el edge, brindando capacidades para el procesamiento local de datos que permiten que los modelos de simulación operen en tiempo real en dispositivos IoT. En cambio, NVIDIA proporciona capacidades de simulación aceleradas por GPU en dispositivos edge, lo que los hace adecuados para simulaciones en tiempo real en drones o vehículos autónomos.

El punto intermedio entre los dos casos anteriores serían los entornos híbridos. Este tipo de entorno conecta la nube y los edges para gestionar el procesamiento en tiempo real y, al mismo tiempo, mantener la conectividad en la nube para actualizaciones, entrenamiento de modelos e inferencia por lotes. Las simulaciones se pueden ejecutar en parte en la nube y en parte en recursos locales para equilibrar la latencia, la seguridad y el costo. Este es el caso de Google Anthos, que permite la implementación de aplicaciones en contenedores, incluidos los modelos de simulación, en infraestructuras locales y en la nube híbrida. Es compatible con Kubernetes para la orquestación y garantiza la coherencia en todos los entornos. Las simulaciones en la nube híbrida también son posibles en AWS Outposts.

Por lo tanto, es necesario garantizar al menos un subconjunto de estos servicios siempre que se implementen modelos basados en datos en contextos empresariales.

Los entornos de implementación más comunes son locales, en la nube, en contenedores y en una solución edge.

Mientras que en una implementación local las aplicaciones se ejecutan en servidores locales de una organización, la implementación en la nube aprovecha las plataformas de nube externas que hacen que las infraestructuras y los servicios escalables estén disponibles, ofreciendo un conjunto diferenciado de servicios según los modelos de implementación de IaaS, PaaS o SaaS. En cambio, la contenerización es popular para las arquitecturas de microservicios: los contenedores incluyen aplicaciones y dependencias empaquetadas juntas, lo que mejora la portabilidad. Finalmente, la implementación en el edge permite alojar servicios livianos en el edge de la red, por ejemplo, cerca de las fuentes de datos, lo que reduce la latencia y aplica escenarios en tiempo real.

La arquitectura relevante para implementar modelos en el edge se puede ver en el contexto general de las arquitecturas de referencia de Internet Industrial de las cosas (IIoT). IIoT se ocupa de los procesos relacionados con la adquisición, transformación, intercambio y análisis de datos.

Análisis de los entornos disponibles para la arquitectura de modelos

Para seleccionar un entorno para desplegar una arquitectura de modelo de simulación en el caso de sistemas complejos es necesario elegir el entorno adecuado que admita un cálculo eficiente, escalabilidad e integración perfecta con otros componentes. Los diferentes entornos atienden distintos aspectos del despliegue de modelos, como la ejecución en tiempo real, la integración en la nube y en el edge, la orquestación y la interacción con dispositivos IoT.

Para las simulaciones que requieren computación distribuida e implementación modular, las herramientas de orquestación como Kubernetes y Docker brindan flexibilidad y escalabilidad. Los modelos de simulación se pueden contenerizar con Docker y orquestar a través de Kubernetes, lo que permite un escalado eficiente, tolerancia a fallas y administración en todos los clústeres. Kubeflow se puede utilizar para gestionar flujos de trabajo de simulación y aprendizaje automático en un entorno de Kubernetes, lo que simplifica el entrenamiento, la validación y la implementación de modelos en el caso de modelos basados en datos. Para gestionar los procesos de aprendizaje automático y automatizar los procesos de entrenamiento, implementación y reentrenamiento de modelos, Apache Airflow podría ser útil para orquestar flujos de trabajo.

Para ciertas industrias o casos de uso, la implementación de modelos en servidores locales o clústeres de computación de alto rendimiento (HPC) garantiza un mejor control, seguridad o asignación de recursos. Por lo tanto, las soluciones HPC se utilizan para ejecutar modelos de simulación de alto rendimiento computacional que requieren control directo sobre el hardware o deben cumplir con estrictos requisitos de seguridad. Se puede implementar software como OpenFOAM (para simulaciones CFD) o Ansys Fluent en un entorno HPC para lograr una computación de alta velocidad.





1. SENSÓRICA - IoT

Los sensores son dispositivo que miden una magnitud física. Pueden ser analógicos o digitales. En caso de generarse un valor analógico, en el proceso de lectura se debe transformar el valor analógico a un valor digital para poder ser almacenado en una variable. Se habla de sensores inteligentes cuando no sólo permiten la lectura de la magnitud, sino también el procesado del dato.

Ejemplo: sonda de temperatura.

El gemelo digital está fuertemente influenciado por los requisitos del sensor. La representación virtual de modelos necesita información variada del activo real. Esta información puede estar disponible con la instalación de los sensores adecuados en los equipos y su posterior y correcto almacenamiento de los datos.

Para determinar qué sensor es el adecuado es necesario especificar algunas características básicas:

- Rangos de amplitud del sensor. Dependiendo del sistema y equipo medido, la capacidad de alcance varía. Es decir, mediciones de rpm de 0 a 200 cuando el límite diseñado en rpm del eje es 150 rpm, sería correcto. Si el rango del sensor de medición del sensor es de 0 a 100 rpm sería incorrecto. Una mala selección podría resultar en la imposibilidad de capturar el fenómeno físico que nos interesa.
- Naturaleza de la medición. Dependiendo de la magnitud física, la tecnología de sensores puede variar. Las propiedades físicas pueden determinar la complejidad del sensor y su medición.
- Tasas de frecuencia. Determinar el nivel de resolución temporal requerido para la aplicación. La cantidad medida puede tener una frecuencia baja, media o alta, por lo que el sensor necesita adaptarse a los fenómenos para que se midan los datos necesarios, lo que podría resultar en demasiados datos inútiles, unidades sensoriales más caras y otro tipo de ineficiencias. Los Hz de adquisición del sensor van a determinar el volumen de datos adquiridos. Si la frecuencia con la que varía una variable no es alta, sensores que miden a 1 Hz deberían ser más que suficiente, por ejemplo.

1.1 Tipos de sensores, características, cuántos, para qué?

A continuación, se realiza una descripción del tipo de señal y las diferentes tecnologías de sensores disponibles para su medición.

1. Intensidad. Esta señal se puede obtener mediante diferentes tecnologías.

- a. Resistencia de derivación + Amplificador de aislamiento/ADC. Este método calcula el valor de la intensidad a partir del voltaje a través de una resistencia conocida. La caída de voltaje a través de la resistencia en derivación es proporcional a la corriente que fluye a través de ella. Ampliamente utilizado, es adecuado para detectar pequeñas corrientes y no genera mucho calor. Requiere componentes adicionales para aislamiento y una idoneidad limitada para grandes corrientes.
- b. Sensores de efecto Hall. Basado en el fenómeno del efecto Hall. Mide el campo magnético generado por la corriente. Adecuado para corriente alterna, continua o pulsante. Detección de corriente de uso general.
- c. Bobinas de Rogowski. Mide corriente alterna (CA) o pulsos de corriente de alta velocidad. Consiste en una bobina flexible enrollada alrededor del conductor. Apto para corriente alterna, no intrusivo y compacto. Monitoreo de la calidad de la energía, gestión de la energía.
- d. Sensores Fluxgate. Proporciona alta precisión. Mide el campo magnético generado por la corriente. Medición de corriente de alta precisión.
- e. Sensores de corriente sin núcleo. Mide corriente sin núcleo magnético. Adecuado para dispositivos pequeños y aplicaciones rentables. Rendimiento comparable al de los sensores de corriente con núcleo. Es interesante para aplicaciones donde el tamaño y el costo importan. La medida de corriente no está sujeta a compensación por histéresis magnética y no depende de la temperatura de desmagnetización de la ferrita. La frecuencia y el ancho de banda no están limitados por la saturación inherente del elemento magnético del núcleo.

2. Energía eléctrica. Utiliza sensores de corriente y voltaje.

3. Flujo. Estas variables se pueden obtener mediante diferentes tecnologías.

- Sensor de flujo de presión diferencial. Estos sensores calculan la diferencia de presión entre dos puntos en un sistema de fluidos. Una constricción (como un tubo Venturi o una placa de orificio) acelera el fluido, creando un diferencial de presión. Al medir la presión en ambos puntos, el caudal se puede determinar mediante la ecuación de Bernoulli. Estos sensores son versátiles y ampliamente utilizados en la industria.
- Sensores de flujo másico térmico. Estos sensores se basan en las propiedades térmicas de un fluido para medir el flujo. Se utilizan comúnmente en aplicaciones gaseosas, de bajo flujo y alta precisión (por ejemplo, fabricación de semiconductores). Los sensores de flujo másico térmico son precisos y adecuados para escenarios de bajo flujo.

- Sensores de flujo electromagnéticos. Estos sensores generan un campo magnético a través del fluido y miden el voltaje inducido a medida que el fluido fluye a través del campo. El voltaje inducido es proporcional a la velocidad del fluido y, en consecuencia, al caudal. Funcionan bien para fluidos conductores.
- Sensores de flujo ultrasónicos. Los sensores ultrasónicos utilizan ondas sonoras para medir el flujo. Se emplean técnicas de tiempo de tránsito o Doppler. Los sensores de tiempo de tránsito miden el tiempo que tardan las ondas ultrasónicas en viajar aguas arriba y aguas abajo en el fluido. Los sensores Doppler analizan los cambios de frecuencia causados por el movimiento de los fluidos. No son intrusivos y adecuados para diversas aplicaciones.
- Sensores de flujo de turbina. Utilice una turbina giratoria para medir el caudal.
- Sensores de flujo térmico MEMS. Sensores pequeños, de bajo coste y con alta sensibilidad para caudales muy bajos.

4. Frecuencia. Estas variables se pueden obtener mediante diferentes tecnologías.

- a. Sensores de corriente de efecto Hall. Hasta 100 KHz.
- b. Sensores de bobina Rogowski.

5. Composición de los gases. Estas variables se pueden obtener mediante diferentes tecnologías.

- Cromatografía de gases (GC). La cromatografía de gases es una técnica poderosa que separa y analiza mezclas de gases. Utiliza una fase estacionaria (como gel de sílice o alúmina) y una fase gaseosa móvil. La GC se utiliza ampliamente para análisis de composición precisos en laboratorios.
- Espectroscopía Infrarroja (IR). Los sensores IR detectan la absorción de luz infrarroja por moléculas de gas. Cada gas tiene un patrón de absorción único, lo que permite la identificación y cuantificación de componentes específicos en una mezcla.
- Sensores de gas capacitivos. Estos sensores miden cambios en la capacitancia debido a la adsorción de gas en una superficie sensible. Se utilizan comúnmente para detectar gases específicos como dióxido de carbono o metano.

6. Energía mecánica. Utiliza sensores de par y rpm.

7. Presión. Estas variables se pueden obtener mediante diferentes tecnologías.

- a. Sensores de presión diferencial. Estos sensores calculan la presión midiendo la diferencia de presión entre dos puntos. A menudo utilizan un diafragma o membrana para detectar la diferencia de presión. Los tipos comunes incluyen medidores Venturi (basados en la ecuación de Bernoulli, utilizan una constricción para acelerar el fluido y medir la caída de presión), placas de orificio (similares a los medidores Venturi, pero con un orificio circular simple), tubos de Pitot (miden la presión dinámica en el flujo de fluido) y sensores capacitivos de presión diferencial (usa cambios en la capacitancia debido a la desviación del diafragma).
- b. Sensores de presión extensométrica. Las galgas extensométricas cambian la resistencia cuando se someten a tensiones mecánicas. Unidos a un diafragma, miden la tensión causada por la presión. Estos sensores se utilizan ampliamente en aplicaciones industriales.
- c. Sensores de presión capacitivos. Estos sensores miden cambios en la capacitancia debido a la deflexión del diafragma causada por la presión. Son sensibles y adecuados para aplicaciones de baja presión.
- d. Sensores de presión MEMS. Los sensores de presión de sistemas microelectromecánicos (MEMS) son dispositivos miniaturizados e integrados. Utilizan varios principios (capacitivos, piezorresistivos, etc.) y se encuentran comúnmente en electrónica de consumo, sistemas automotrices y dispositivos médicos.
- e. Sensores de presión piezoeléctricos. Estos sensores convierten la presión en una carga eléctrica utilizando materiales piezoeléctricos. Son adecuados para mediciones dinámicas y pueden soportar altas presiones.



8. RPM. Estas variables se pueden obtener mediante diferentes tecnologías.

- a. Codificadores de eje (tipo rotativo). Estos sensores proporcionan alta resolución, que normalmente oscila entre 1000 y 5000 pulsos por revolución (PPR). Generan impulsos simétricos y claramente definidos al girar el eje. Los codificadores de eje son confiables y ampliamente utilizados para mediciones precisas de RPM.
- b. Sensores fotoeléctricos (tipo óptico). Los sensores fotoeléctricos detectan un objetivo reflectante en el eje giratorio. Son adecuados para detección de resolución media a baja, dependiendo del número de pulsos medidos por revolución. Estos sensores funcionan bien en diversas aplicaciones.
- c. Sensores magnéticos de velocidad de rotación (tipo de proximidad). Estos sensores utilizan varios principios de medición de proximidad magnética para monitorear velocidades en el rango de 0 a 30.000 RPM. Pueden detectar las velocidades de los componentes de la máquina sin contacto directo con el eje giratorio. Los sensores magnéticos son versátiles y rentables.
- d. Tacómetros. Diseñado para medir la velocidad de rotación o superficial. Pueden funcionar en modo de contacto o sin contacto. Los tacómetros mecánicos analizan electrónicamente las revoluciones del sensor mecánico. Este método se usa comúnmente para resoluciones bajas entre 20 y 20.000 RPM. Los tacómetros digitales proporcionan lecturas precisas de RPM y se usan ampliamente en diversas aplicaciones.

9. Temperatura. Estas variables se pueden obtener mediante diferentes tecnologías.

- a. Termómetros de resistencia (RTD). Estos sensores utilizan el cambio en la resistencia eléctrica de un metal (generalmente platino) con la temperatura. Los RTD ofrecen alta precisión y estabilidad, lo que los hace adecuados para diversas aplicaciones.
- b. Termopares (TC). Los TC constan de dos cables metálicos diferentes unidos por un extremo. La diferencia de temperatura entre la unión y el otro extremo genera un voltaje que se correlaciona con la temperatura del fluido. Los TC son resistentes y se utilizan ampliamente en entornos industriales.
- c. Sensores de diodo o transistor. Los diodos semiconductores o transistores presentan características de tensión o corriente que dependen de la temperatura. Estos sensores son compactos, económicos y adecuados para aplicaciones donde el espacio es limitado.
- d. Termografía infrarroja. Los sensores infrarrojos detectan la radiación térmica emitida por el fluido o las superficies sólidas. No tienen contacto y pueden medir la distribución de temperatura en una superficie. Sin embargo, es posible que no funcionen bien en fluidos ópticamente opacos.

10. Par. Esta variable se puede obtener mediante diferentes tecnologías.

- a. Sensores de par de galgas extensométricas. Las galgas extensométricas están adheridas a la superficie del eje. Cuando se aplica torsión, el eje se deforma, provocando tensión en los medidores. La tensión se convierte en una señal eléctrica que puede medirse y analizarse. Utilizado en diversas aplicaciones industriales y científicas para una medición precisa del par.
- b. Sensores de par magnetoelásticos. Estos sensores detectan cambios en la permeabilidad midiendo alteraciones en su propio campo magnético. Cuando se aplica un par, el material experimenta tensión, lo que afecta sus propiedades magnéticas. Muy adecuado para entornos hostiles (altas temperaturas, vibraciones y corrosión).
- c. Sensores de par piezoeléctricos. Un material piezoeléctrico se comprime cuando se aplica un par. Esta compresión genera una carga eléctrica, que se mide mediante un amplificador de carga. Se utiliza en pruebas de motores y transmisiones, dinamómetros y análisis de motores eléctricos.
- d. Sensores de par giratorio. Estos sensores miden directamente el par en un componente giratorio. Utilizan transductores giratorios para proporcionar lecturas precisas. Comúnmente utilizado en pruebas automotrices, desarrollo de motores y maquinaria industrial.
- e. Sensores de par de reacción. Mide el par estático o dinámico sin provocar rotación. Son transductores estacionarios o no giratorios. Útil para aplicaciones donde el eje permanece fijo.

11. Vibración. Esta variable se puede obtener mediante diferentes tecnologías.

- a. Sensores ultrasónicos. Los sensores ultrasónicos utilizan ondas sonoras para medir distancias o detectar objetos. Emiten pulsos ultrasónicos y miden el tiempo que tardan los ecos en regresar. Estos sensores se utilizan comúnmente en aplicaciones de automatización industrial, robótica y automoción.
- b. Acelerómetros piezoeléctricos. Estos sensores miden la vibración y la aceleración. Generan una carga eléctrica cuando se someten a esfuerzos mecánicos. Los acelerómetros piezoeléctricos se utilizan ampliamente en el monitoreo del estado estructural, pruebas automotrices y aplicaciones aeroespaciales.
- c. Sensores MEMS (Sistemas Microelectromecánicos). La tecnología MEMS permite la miniaturización de sensores, haciéndolos adecuados para diversas aplicaciones. Estos sensores pueden medir la aceleración, la presión e incluso la frecuencia. Los acelerómetros MEMS, por ejemplo, pueden detectar vibraciones y cambios de movimiento.

12. Voltaje. Esta variable se puede obtener mediante diferentes tecnologías.

- a. Transductores de tensión. Estos dispositivos permiten la medición aislada de tensión. LEM ofrece una amplia gama de sensores de voltaje, incluidas tecnologías como efecto Hall, Fluxgate y aislamiento digital. Estos sensores pueden medir voltajes desde 10 V hasta 6,4 kV. Los sensores de efecto Hall, por ejemplo, utilizan el campo magnético para generar un potencial eléctrico, lo que permite una medición precisa del voltaje.
- b. Divisores de voltaje. En los sensores de tensión, la medición se basa en un divisor de tensión. Hay dos tipos principales de sensores de voltaje disponibles: sensores de voltaje de tipo capacitivo y sensores de voltaje de tipo resistivo. Estos sensores dividen el voltaje para proporcionar una salida proporcional.
- c. Transformadores de Corriente (CT) y Transformadores de Potencial (PT). Si bien se utilizan principalmente para medir corriente, los CT también pueden medir voltaje indirectamente. Se basan en la ley de Faraday y se utilizan comúnmente en sistemas de energía.

Una de las primeras conclusiones es que los gemelos digitales tienen un fuerte componente de sensorización y requieren un gran esfuerzo a la hora de establecer los estándares de comunicación y las necesidades de hardware asociadas. Normalmente no se requieren muchas tecnologías de sensores diferentes. Pero sí definir correctamente los sensores necesarios. Para ello es importante conocer los modelos de simulación a desarrollar y las variables futuras con las que se compararán los modelos o que alimentan a estos modelos.

Esto hace que se deban considerar algunas condiciones de contorno como las siguientes:

- Capacidad para obtener una medición en el barco de la variable deseada.
- Selección de sensores para establecer qué técnica de acondicionamiento de señal utilizar, qué frecuencia de datos tener, etc.
- Si los sensores son intrusivos y necesitan ser instalados durante la propia instalación del equipo o sistema. En ese caso el gemelo y su despliegue deben tenerse en cuenta desde la fase de diseño del equipo o sistema.
- Vida útil prevista y mantenimiento requerido.

1.2 PLC

El nivel del PLC (Programmable Logic Controller) es el nivel más bajo y el más cercano a las máquinas. Es un ordenador ruggedizado, adaptado para su instalación en procesos industriales y máquinas. Un PLC puede tener decenas de entradas/salidas para las señales, se puede instalar uno solo o puede estar montado en un rack y combinarse con otros PLC's. Los sensores están conectados al PLC y miden magnitudes físicas de forma que se mantienen los valores actualizados para poder ser consultados desde otros dispositivos o niveles superiores a través de un protocolo de campo como MODBUS o PROFINET. Pero también existe la posibilidad de usar TCP/IP. Existe variedad en los protocolos de comunicación no facilita los temas de seguridad en la red de comunicación, es por esto que han estandarizado el protocolo con el OPC-UA (Open Platform Communications Unified Architecture), que está aceptado y es usado de forma bastante masiva por la industria. Usualmente estos protocolos de campo permiten un modelo de suscripción en la que se envía el dato actualizado cada vez que existe un cambio.

Ejemplo: Siemens S7-1500.

Existe también la posibilidad de utilizar un SCADA (Supervisory Control And Data Acquisition). Es un dispositivo hardware de control que incluye procesadores, una red de comunicación de datos y una interface de usuario para la supervisión de máquinas y procesos. La red de comunicación permite el acceso remoto a los datos recogidos por el SCADA.

PLC y SCADA no son incompatibles, se pueden combinar, todo depende de las necesidades de conexión entre sensores y el entorno en que se procesen los datos. Y ambos se pueden eliminar de la arquitectura si trabajamos con sensores IoT que se conectan vía wifi a un cloud.



1.3 IoT Gateway

Si trabajamos con sensores directamente conectados a una red, debemos emplear lo que se conoce como IoT gateways. Estos dispositivos eran inicialmente hardware, pero ahora ya están disponibles en versión software (plataforma virtual) para alojarlos en la nube o en un servidor local.

Es un dispositivo que conecta sensores, módulos de IoT y dispositivos inteligentes a la nube. Sirven como portal de entrada inalámbrica para dar acceso de internet a diversos dispositivos

Pero tienen la ventaja de que en ellos se puede realizar cierto procesamiento de datos, tales como agregación de datos, traducción de protocolos, filtrado y compresión de datos. De esta manera se reduce la cantidad de datos transmitidos hacia el cloud y mejora la eficiencia de la red.

Dependiendo de múltiples factores, tendremos unos tipos u otros de pasarela IoT. Ya sea por cuestiones físicas o de recursos, como por motivos de su aplicación específica, podemos establecer varias categorías o tipos de gateway. Como siempre, la última palabra sobre cuál elegir dependerá de los requisitos específicos de cada red y de los diferentes tipos de dispositivos conectados (y su número).

1. Los gateways IoT de tipo industrial, que están, lógicamente, pensadas y diseñadas para aplicaciones industriales. Se caracterizan por su resistencia, ya que pueden (y suelen) funcionar en entornos difíciles, y ofrecen diversas opciones de conectividad.
2. En entornos en los que basta con una solución de conectividad inalámbrica, fundamentalmente porque se trate de áreas de conectividad reducidas, la opción es la de los gateway IoT inalámbricos. Se utilizan protocolos de comunicación como WiFi, pero también otros como Zigbee, y otros. Es una solución eficiente, de bajo consumo y coste.
3. Las pasarelas integradas en los propios dispositivos conectados forman una solución compacta e integrada para redes IoT pequeñas o muy pequeñas.
4. Las pasarelas multiprotocolo permiten la comunicación entre dispositivos que utilizan distintos protocolos de comunicación, una solución flexible y versátil.
5. Pasarelas cloud, alojadas en la nube, que dan una solución de acceso remoto para gestionar y procesar los datos de los dispositivos conectados.

1.4 Edge

Los dispositivos en este nivel se encargan de leer los datos de los sensores del PLC y de enviarlos a un Data Lake. Son dispositivos preparados para funcionar en condiciones poco favorables a pie de máquina (vibraciones, polvo, suciedad, salpicaduras). En este nivel también se podrían realizar algunos cálculos ligeros sobre los datos, como, por ejemplo, filtrado de señales.

Mover parte de la carga computacional a estos dispositivos que suelen situarse cerca de la fuente de datos, permite reservar la nube para tareas de computación intensivas. Estas tareas pueden incluir el análisis de conjuntos de datos más grandes de una gama más amplia de fuentes y el uso de análisis avanzados para obtener información sobre tendencias. Estos dispositivos que están asociados a lo que se conoce como Edge computing solo necesitan proporcionar los resultados de los datos que se procesaron localmente, después de lo cual normalmente se descartan. Solo la información útil y post procesada se envía a la nube y se almacena.



2. ENTORNOS DE SIMULACIÓN

El núcleo central de un gemelo digital son los modelos de simulación que van a replicar el comportamiento del equipo real. Existen múltiples herramientas para el desarrollo de modelos de simulación. Aquí se van a describir las más usadas para desarrollos de modelos de producto. Entre ellas destacan:

- Simcenter Amesim
- Matlab&Simulink
- OpenModelica
- Dymola

Simcenter Amesim

Simcenter es un conjunto de soluciones de software ofrecidas por Siemens Digital Industries Software que permiten a los ingenieros realizar simulaciones y pruebas de sistemas en varios dominios. Dentro de Simcenter se encuentra Amesim, que es una herramienta para modelar y simular sistemas multidominio, como componentes mecánicos, hidráulicos, neumáticos, térmicos y eléctricos. Este software tiene licencia propietaria.

Las características clave son:

- Simulación basada en la física para múltiples dominios y escalas.
- Simulación basada en datos para mejorar la información y la optimización.
- Integración de pruebas y simulaciones para una validación y verificación más rápidas.
- Proceso de simulación y gestión de datos para mejorar la colaboración y la eficiencia.

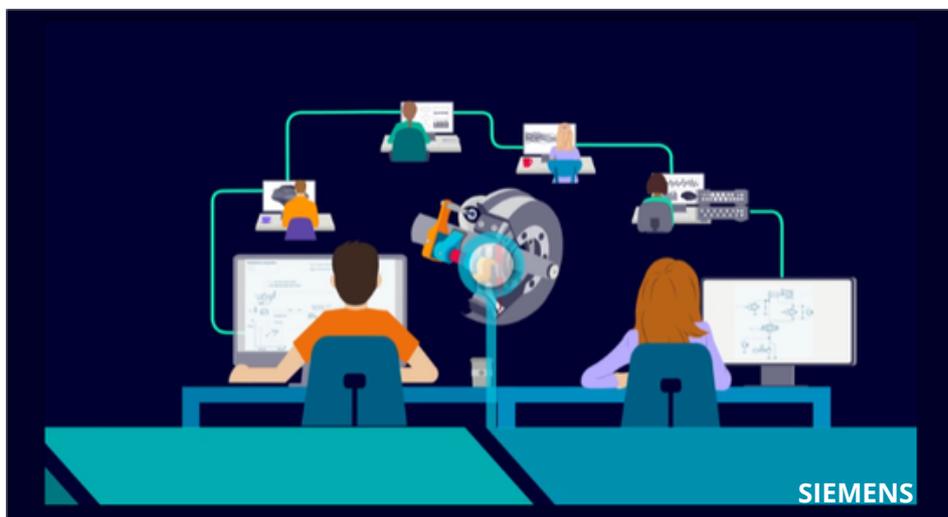
Esta herramienta permite a los usuarios crear y validar modelos a nivel de sistema, analizar el comportamiento y el rendimiento del sistema y optimizar el diseño y el funcionamiento de dicho sistema. Simcenter Amesim proporciona una biblioteca de más de 6.500 componentes listos para usar (drag & drop), que cubren una amplia gama de dominios de ingeniería, como la automoción, la aeroespacial, y la naval. Simcenter Amesim también admite la cosimulación con otras soluciones de Simcenter, como Simcenter 3D o Simcenter STAR-CCM+ (Simulación Computacional de Fluidos), así como con software de terceros, como Matlab/Simulink, Modelica y Functional Mock-up Interface (FMI).

Esta herramienta tiene un motor de simulación con solvers de paso fijo y variable patentados. Esto hace que se requiera de una instalación mínima de su entorno para ejecutar los modelos, aunque estén encapsulados.

Permite, además, la reutilización y personalización de los componentes de las librerías utilizando Modelica o C/C++.

Simcenter Amesim cumple con los requisitos mínimos generales para las herramientas de modelado, es decir, una ejecución rápida y robusta, con una ventana gráfica y una amplia documentación para facilitar su uso.

Simcenter Amesim combina bibliotecas multifísicas listas para usar con soluciones orientadas a la industria. Esto permite a los usuarios crear modelos rápidamente y realizar análisis con precisión.



Las bibliotecas integradas se organizan en tres categorías: bibliotecas base, bibliotecas de aplicaciones y bibliotecas de industria.

Las bibliotecas base proporcionan los bloques de construcción fundamentales para modelar y simular sistemas multidominio. Incluyen componentes para dominios mecánicos, hidráulicos, neumáticos, térmicos, eléctricos y de control. Los usuarios también pueden crear sus propios componentes o bibliotecas personalizados utilizando el lenguaje Modelica o el código C/C++. Las bibliotecas base son:

- **Mecánica:** Esta biblioteca contiene componentes para modelar cuerpos rígidos y flexibles, articulaciones, engranajes, cojinetes, resortes, amortiguadores y más.
- **Hidráulica:** Esta biblioteca contiene componentes para modelar sistemas de potencia de fluidos, como bombas, válvulas, cilindros, motores, tuberías y acumuladores.
- **Neumático:** Esta biblioteca contiene componentes para modelar sistemas de dinámica de gases, como compresores, boquillas, válvulas, cámaras e intercambiadores de calor.
- **Térmica:** Esta biblioteca contiene componentes para modelar sistemas de transferencia de calor y termodinámica, como fuentes, sumideros, conductores, convectores, radiadores y fluidos térmicos.
- **Eléctrico:** Esta biblioteca contiene componentes para modelar sistemas eléctricos y electrónicos, como fuentes, cargas, resistencias, condensadores, inductores, transformadores, diodos, transistores e interruptores.
- **Control:** Esta biblioteca contiene componentes para modelar sistemas de control, como sensores, actuadores, controladores, filtros, lógica y funciones matemáticas.

Las bibliotecas de aplicaciones proporcionan componentes y herramientas especializadas para modelar y simular aplicaciones específicas, como sistemas de propulsión, combustión, aire acondicionado, o sistema eléctrico. Las bibliotecas de aplicaciones son:

- **Tren motriz:** Esta biblioteca contiene componentes para modelar y simular motores de combustión interna, transmisiones, líneas de transmisión y máquinas eléctricas.
- **Combustión:** Esta biblioteca contiene componentes para modelar y simular procesos de combustión, como la ignición por chispa, la ignición por compresión, la turbina de gas y los motores de cohetes.
- **Aire acondicionado:** Esta biblioteca contiene componentes para modelar y simular sistemas de aire acondicionado, como compresores, evaporadores, condensadores, válvulas de expansión y refrigerantes.
- **Pila de combustible:** Esta biblioteca contiene componentes para modelar y simular sistemas de pilas de combustible, como la membrana de intercambio de protones, el óxido sólido y las pilas de combustible de metanol directo.
- **Batería:** Esta biblioteca contiene componentes para modelar y simular sistemas de baterías, como baterías de plomo-ácido, níquel-hidruro metálico, iones de litio y litio-azufre.

Las bibliotecas de la industria proporcionan componentes y herramientas a medida para modelar y simular sistemas en industrias específicas, como la de automoción, aeroespacial, marina, energética y de bienes de consumo. También incluyen estándares de la industria y mejores prácticas para garantizar el cumplimiento y la calidad. Las bibliotecas de la industria son:

- Automoción: Esta biblioteca contiene componentes y herramientas para modelar y simular sistemas de automoción, como la dinámica del vehículo, el chasis, la suspensión, los frenos, la dirección y la asistencia al conductor.
- Aeroespacial: Esta biblioteca contiene componentes y herramientas para modelar y simular sistemas aeroespaciales, como la dinámica de vuelo, la aerodinámica, el tren de aterrizaje, el control de vuelo y la aviónica.
- Marino: Esta biblioteca contiene componentes y herramientas para modelar y simular sistemas marinos, como la propulsión, la maniobra, la hidrodinámica y el movimiento de los barcos.
- Energía: Esta biblioteca contiene componentes y herramientas para modelar y simular sistemas de energía, como la generación, distribución, conversión y almacenamiento de energía.
- Bienes de consumo: Esta biblioteca contiene componentes y herramientas para modelar y simular sistemas de bienes de consumo, como electrodomésticos, dispositivos personales y productos inteligentes.

Amesim no sólo es capaz de desarrollar modelos KBM (Knowledge Based Models) sino que proporciona algunas funcionalidades para el desarrollo de DDMs (Data Driven Models). Es posible importar Redes Neuronales a Amesim. Es posible construir un ROM (Reduce Order Model), por ejemplo. Esta herramienta permite crear ROMs y DDM con el uso de algoritmos de regresión lineal y no lineal, redes neuronales y una adecuada descomposición ortogonal.



Simcenter Amesim ofrece varias opciones de visualización de datos para facilitar el análisis y la comprensión de los resultados de la simulación. Algunas de las opciones son:

- **Gráficas.** Permite crear gráficos de diferentes tipos, como XY, polar, barra, circular, superficie, etc. Los gráficos se pueden personalizar con diferentes opciones de color, escala, leyenda, etiqueta, etc. Los gráficos se pueden exportar a diferentes formatos, como PNG, JPG, PDF, etc.
- **Dashboards.** Permite crear paneles interactivos con diferentes elementos gráficos, como indicadores, medidores, termómetros, interruptores, botones, etc. Los cuadros de mando se pueden vincular a los parámetros del modelo y mostrar su evolución en tiempo real. Los paneles se pueden exportar a diferentes formatos, como HTML, PNG, JPG, etc.
- **Análisis lineal.** Permite realizar análisis lineales de los sistemas, como análisis de frecuencia, análisis modal, análisis de estabilidad, etc. Los resultados se pueden visualizar en diferentes gráficos, como Bode, Nyquist, Nichols, etc. Los gráficos se pueden exportar a diferentes formatos, como PNG, JPG, PDF, etc.
- **Animación.** Permite crear animaciones 2D o 3D de los sistemas, utilizando imágenes, modelos CAD o geometrías sencillas. Las animaciones se pueden sincronizar con los resultados de la simulación y muestran el comportamiento dinámico de los sistemas. Las animaciones se pueden exportar a diferentes formatos, como AVI, MP4, GIF, etc.

Por otro lado, puede exportar datos de simulaciones e incluso postprocesar datos de entrada para exportarlos o reutilizarlos en una simulación. Son posibles diferentes formatos de archivo.

Simcenter Amesim también se puede implementar como un servicio en la nube, lo que permite a los usuarios hacer frente a los picos de uso de la simulación del sistema y acceder a sus modelos desde cualquier lugar. Simcenter Amesim también se puede integrar con Simcenter Webapp Server, que proporciona acceso basado en la web a resultados de simulación específicos para los usuarios finales. Simcenter Webapp Server permite a los usuarios almacenar y compartir modelos, ejecutar simulaciones y trazar resultados relevantes a través de un navegador web. Simcenter Webapp Server también permite a los usuarios crear aplicaciones web de ingeniería personalizadas y fáciles de usar que se pueden compartir con cualquier persona de la organización, como equipos de ventas, preventas y técnicos.



Matlab/Simulink

Matlab y Simulink son potentes herramientas de software desarrolladas por MathWorks que permiten a ingenieros y científicos realizar computación numérica, simulación y diseño basado en modelos. Matlab es una plataforma de programación y computación numérica que permite a los usuarios analizar datos, desarrollar algoritmos y crear modelos. Simulink es un entorno de diagrama de bloques que permite a los usuarios diseñar sistemas con modelos multidominio, simular antes de pasar al hardware y desplegar sin necesidad de escribir código.

Los avances en la tecnología de propulsión y simulación marina, demostrados a través de varios estudios de MATLAB/Simulink, subrayan el papel fundamental que desempeñan estas herramientas en el avance de la ingeniería marítima. MATLAB y Simulink se han vuelto indispensables al permitir simulaciones en tiempo real de motores diésel y sistemas de propulsión de buques navales, incorporar el aprendizaje automático para optimizar el mantenimiento y facilitar la creación de gemelos digitales para un análisis exhaustivo del sistema. Estas capacidades subrayan la aplicación integral de MATLAB/Simulink en propulsión y simulación marinas, mejorando la comprensión teórica y práctica de los sistemas marinos. Contribuyen significativamente al sector marítimo al mejorar la eficiencia operativa, la sostenibilidad y la seguridad.

Matlab es un lenguaje de programación multiparadigma propietario y un entorno de computación numérica. Matlab permite la manipulación de matrices, el trazado de funciones y datos, la implementación de algoritmos, la creación de interfaces de usuario y la interfaz con programas escritos en otros lenguajes. Los ingenieros y científicos utilizan Matlab porque les permite expresar directamente las matemáticas de matrices y arrays. El álgebra lineal en Matlab es intuitiva y concisa. Aplicable en varios campos de trabajo, como análisis de datos, procesamiento de señales e imágenes, diseño de controles y otras aplicaciones.

Las características clave son:

- Computación numérica integral y análisis de datos.
- Visualización y exploración de datos.
- Creación de scripts, funciones y clases.
- Creación de aplicaciones para implementación web y de escritorio.
- Computación paralela y ejecución en la nube.



Simulink es un entorno de diagrama de bloques para simulación multidominio y diseño basado en modelos. Es compatible con el diseño a nivel de sistema, la simulación, la generación automática de código y las pruebas y verificaciones continuas de sistemas integrados. Simulink proporciona un editor gráfico, bibliotecas de bloques personalizables y solvers para modelar y simular sistemas dinámicos. Está integrado con Matlab, lo que le permite incorporar algoritmos de Matlab en modelos y exportar los resultados de la simulación a Matlab para su posterior análisis.

Se presenta una descripción resumida de Simulink.

- Editor gráfico para la construcción y gestión de diagramas de bloques jerárquicos.
- Librerías de bloques predefinidos para el modelado de sistemas de tiempo continuo y tiempo discreto.
- Motor de simulación con solucionadores ODE de paso fijo y paso variable.
- Visualización de alcances y datos para ver los resultados de la simulación.
- Herramientas de gestión de proyectos y datos para la gestión de archivos y datos de modelos.
- Herramientas de análisis de modelos para refinar la arquitectura del modelo y aumentar la velocidad de simulación.
- Bloque de función de Matlab para importar algoritmos de Matlab en modelos.
- Herramienta de código heredada para importar código C y C++ en modelos.
- Ingeniería de sistemas basada en modelos y diseño de arquitecturas.

Matlab/Simulink cumple con los requisitos mínimos generales para las herramientas de modelado, es decir, una ejecución rápida y robusta, con una ventana gráfica y una amplia documentación para facilitar su uso (este software tiene licencia propietaria).

Consta de las siguientes herramientas de aplicación general en el modelado de sistemas:

- Matlab: Está diseñado para ser aplicable en varios campos de trabajo, como el análisis de datos, el procesamiento de señales e imágenes, el diseño de controles y otras áreas. El software proporciona herramientas de desarrollo útiles que mejoran el mantenimiento y el rendimiento del código, además de integrarse con otros lenguajes de programación como Fortran, C/C++, .NET o Java. Otra característica útil es que Matlab soporta la programación orientada a objetos, que permite definir objetos que combinan datos con funciones que operan sobre esos mismos datos. Esta característica hace que sea mucho más fácil mantener y ampliar las capacidades y el código de los componentes.
- Simulink: Es una interfaz gráfica de usuario avanzada y fácil de usar que proporciona a los usuarios una creación de modelos fácil de usar, edición de conexiones, simulación de modelos y análisis y trazado de resultados. Simulink puede comunicarse con Matlab.

Simulink incorpora extensas bibliotecas integradas. Al mismo tiempo, muchas funcionalidades de Simulink se proporcionan mediante el uso de Toolbox propietario. Con la extensión de software correspondiente se proporciona una lista de áreas de aplicación multifísicas.

- Aplicaciones generales. Para sistemas dinámicos generales.
- Aplicaciones automotrices. Simulink permite modelar y simular una gran variedad de sistemas de automoción. Se puede modelar y simular el vehículo y el entorno y desarrollar algoritmos de control para aplicaciones automotrices.
- Aplicaciones de automatización industrial.
- Procesamiento de señales. Para modelos de sistemas de procesamiento de señales y comunicaciones.
- Diseño de control. Diseñar y analizar sistemas de control modelados en el entorno de Simulink.
- Modelado físico. Para modelar sistemas físicos en el entorno de Simulink.
- Simscape amplía Simulink con herramientas para modelar sistemas que abarcan dominios mecánicos, eléctricos, hidráulicos y otros dominios físicos como redes físicas. Proporciona bloques de creación fundamentales de estos dominios para permitirle crear modelos de componentes personalizados.
- Ingeniería de sistemas. Diseñar y analizar arquitecturas de sistemas en Simulink.

System composer permite la especificación y el análisis de arquitecturas para la ingeniería de sistemas basada en modelos y el modelado de arquitecturas de software.

Matlab/Simulink permite cargar y utilizar datos desde el espacio de trabajo de Matlab, cargar datos de diferentes formatos de archivo, acceder a servidores de datos.

Es posible visualizar datos a partir de una simulación. Matlab ofrece una amplia gama de tipos de gráficos incorporados como gráficos de líneas, gráficos de dispersión, gráficos de distribución y gráficos geográficos para visualizar conjuntos de datos de un conjunto diverso de aplicaciones. Estos gráficos permiten explorar visualizaciones de datos, anotar y personalizar visualizaciones de datos, exportar visualizaciones de datos, integrar la visualización de datos con el análisis de datos o incluso conectar Matlab con otras herramientas de visualización de datos (ejemplos son Tableau, Power BI o Plotly).

Los datos se pueden exportar desde simulaciones, e incluso postprocesar los datos de entrada para exportarlos o reutilizarlos en una simulación. Son posibles diferentes formatos de archivo.

SIMULINK®

Es posible crear modelos ROM utilizando Simulink. Se pueden utilizar varios métodos, dependiendo del tipo y la complejidad del modelo. Algunos de estos métodos son:

- Simplificación del modelo. Reducción del modelo mediante la eliminación de estados variables que no tienen ningún efecto en la respuesta general del modelo. Este método se utiliza para reducir la precisión física del modelo. Para realizar esta simplificación, se pueden utilizar funciones integradas de Matlab que se aplican a las representaciones de espacio de estados y a las funciones de transferencia. En cambio, mientras se sigue modelando con ecuaciones físicas, se deben utilizar algunos conocimientos sobre la física subyacente para realizar una simplificación manual del modelo en sí.
- Reducción de modelos basados en datos. Algunas toolboxes se pueden aplicar a la industria de los gemelos digitales y la creación de modelos basados en datos. Algunas técnicas disponibles son el ajuste de curvas, las tablas de búsqueda, las redes neuronales o NARX.

Hay opciones web de la suite de software MathWorks, que permiten la simulación.

- Simulink Online. Se trata de una funcionalidad que proporciona acceso a Simulink desde cualquier navegador estándar en cualquier lugar donde se tenga acceso a Internet. Se puede utilizar Simulink Online para diseñar, simular y desplegar sus modelos sin necesidad de descargas ni instalaciones. También es posible colaborar con otros a través del uso compartido en línea y el almacenamiento en la nube.
- Compilador de Simulink. Es un producto que permite compartir simulaciones como ejecutables independientes, aplicaciones web y FMU. Se puede utilizar Simulink Compiler para generar aplicaciones web a partir de sus modelos de Simulink y desplegarlas en Matlab Web App Server, donde se pueden ejecutar e interactuar con ellas a través de un navegador web.
- Vista web de Simulink. Es una función que permite ver y explorar los modelos de Simulink en un navegador web. Se puede utilizar Simulink Web View para compartir los modelos con otras personas que no tengan instalado Simulink, o para incrustar los modelos en páginas web o documentos. También puede utilizar Simulink Web View para ejecutar simulaciones y ajustar los parámetros de los modelos.

OpenModelica

Open Source Modelica Consortium (OSMC) es una organización no gubernamental sin fines de lucro y un instituto de investigación en el área de sistemas ciberfísicos, con el objetivo de desarrollar y promover el desarrollo y el uso de la implementación de código abierto OpenModelica del lenguaje informático Modelica y los estándares relacionados, incluidos [FMI](#), [SSP](#), [DCP](#), así como las herramientas y bibliotecas de código abierto asociadas a OpenModelica, denominadas colectivamente OpenModelica Environment.

OpenModelica está disponible para uso comercial y no comercial bajo las licencias GPLv3 y bajo las condiciones de la Licencia Pública de OSMC, tal y como se define en los Estatutos de OSMC. El objetivo de OSMC, dentro de las limitaciones de sus recursos, es proporcionar soporte y mantenimiento de OpenModelica, apoyar la publicación en la web y coordinar las contribuciones a OpenModelica.

OpenModelica utiliza el lenguaje Modelica, que es adecuado para el modelado multidominio de sistemas grandes, complejos y heterogéneos. Es adecuado para subcomponentes multidominio (mecánicos, eléctricos, electrónicos, magnéticos, hidráulicos, térmicos, de control, de energía eléctrica y orientados a procesos), que admiten la conexión causal de componentes gobernados por ecuaciones matemáticas para facilitar el modelado a partir de principios básicos.

Los modelos se describen matemáticamente mediante ecuaciones diferenciales, algebraicas y discretas. No es necesario resolver ninguna variable manualmente. Una herramienta de Modelica tendrá suficiente información para decidir eso automáticamente. Modelica está diseñado de tal manera que se pueden utilizar algoritmos especializados disponibles para permitir el manejo eficiente de modelos grandes con miles de ecuaciones.

La semántica del lenguaje Modelica se especifica por medio de un conjunto de reglas para traducir cualquier clase descrita en el lenguaje Modelica a una estructura plana de Modelica. Una clase pensada para ser simulada por sí misma se denomina modelo de simulación. La estructura plana de Modelica también se define para otros casos que no sean los modelos de simulación; incluyendo funciones (se pueden usar para proporcionar contenidos algorítmicos), paquetes (utilizados como mecanismo de estructuración) y modelos parciales (utilizados como modelos base).

Existen restricciones semánticas específicas para un modelo de simulación con el fin de garantizar que el modelo esté completo; permiten que su estructura plana de Modelica se transforme aún más en un conjunto de ecuaciones diferenciales, ecuaciones algebraicas y discretas.

OpenModelica

Se presenta una descripción resumida del lenguaje Modelica.

- Lenguaje declarativo de tipos estáticos.
 - Las ecuaciones y las funciones matemáticas permiten la modelización causal.
 - Especificación de alto nivel y comprobación de tipo estática para una mayor corrección.
- Modelado multidominio.
 - Combina sistemas eléctricos, mecánicos, termodinámicos, hidráulicos, biológicos, de control, de eventos, en tiempo real, etc.
- Todo es una clase.
 - Prácticas de ingeniería seguras mediante lenguaje orientado a objetos de tipo estático, concepto general de clase.
- Programación de componentes visuales.
 - Capacidades de arquitectura de sistemas jerárquicos.
- Eficiente, no propietario.

OpenModelica consta de las siguientes herramientas de aplicación general en el modelado de sistemas:

- Compilador interactivo avanzado de OpenModelica (OMC). Es un compilador de Modelica que traduce Modelica a código C. Está soportado por una tabla de símbolos que contiene definiciones de clases, funciones y variables. OMC también incluye un intérprete de Modelica para el uso interactivo y la evaluación constante de expresiones. El subsistema también incluye facilidades para construir ejecutables de simulación vinculados con solvers numéricos ODE o DAE seleccionados [Fuente : [Advanced Interactive OpenModelica Compiler \(OMC\)](#)].
- OpenModelica Connection Editor (OMEdit): Es una interfaz gráfica de usuario avanzada de código abierto y fácil de usar que proporciona a los usuarios la creación de modelos fáciles de usar, la edición de conexiones, la simulación de modelos y el trazado de resultados. OMEdit se comunica con OMC a través de una API interactiva, solicita información del modelo y crea modelos/diagramas de conexión basados en la versión estándar de anotaciones de Modelica. Desarrollado con bibliotecas Qt y C++ [Fuente: [OpenModelica Connection Editor \(OMEdit\)](#)].

El lenguaje Modelica promueve la reutilización ordenada de modelos de componentes mediante paquetes que contienen bibliotecas estructuradas de modelos reutilizables. El ejemplo más destacado es la Biblioteca Estándar Modelica (MSL).

OpenModelica permite la inclusión de datos externos mediante la carga de datos en formato de tabla desde un archivo externo proporcionado, así como el acceso a archivos y rutas a través de scripting. Se puede acceder a las rutas del repositorio con fines de modelado y simulación con datos actualizados.

Es posible visualizar los resultados de la simulación a través de gráficos, como el gráfico paramétrico, el gráfico de matriz y el gráfico paramétrico de matriz. Se pueden mostrar todas las variables guardadas.

Por otro lado, es posible visualizar la geometría simulada. Al proporcionar un CAD, asociado a los componentes abiertos en OMEdit, se puede activar la simulación con animación (Simulation > Simular con Animation); El indicador +d=VISXML se incluye automáticamente, de modo que se generan todos los resultados de las variables que se necesitan para la animación.

Los archivos de salida se pueden generar en diferentes formatos en el momento de la simulación, exportar las variables seleccionadas después de la simulación o exportar las variables trazadas a un archivo. Los formatos de archivo generales son MAT, CSV y PLT.

Dymola

Dymola es una herramienta de software comercial de modelado y simulación diseñada para modelar sistemas físicos complejos. Desarrollado por Dassault Systèmes, se utiliza a menudo en el campo de la ingeniería y la simulación de sistemas.

Se presenta una descripción resumida de Dymola :

- Soporte del lenguaje Modelica: Dymola se basa en el lenguaje de modelado Modelica, que es un lenguaje de estándar abierto diseñado para modelar sistemas físicos complejos.
- Modelado multidominio: Dymola admite el modelado y la simulación de sistemas multidominio, lo que permite a los usuarios integrar componentes de varios dominios de ingeniería, como sistemas mecánicos, eléctricos, térmicos y de control.
- Compatibilidad con Open Modelica: Dymola es compatible con la plataforma de código abierto OpenModelica, lo que facilita la interoperabilidad entre diferentes herramientas de modelado y entornos compatibles con el lenguaje Modelica.
- Editor de modelos gráficos: El software suele incluir un editor gráfico que permite a los usuarios crear modelos ensamblando componentes gráficamente. Esto lo hace fácil de usar para ingenieros y científicos sin amplios conocimientos de programación.
- Capacidades de simulación: Dymola proporciona capacidades de simulación avanzadas, lo que permite a los usuarios realizar simulaciones dinámicas de sistemas complejos. Admite simulaciones de tiempo continuo y eventos discretos, lo que permite el estudio del comportamiento del sistema a lo largo del tiempo.

- Bibliotecas de modelos: Dymola a menudo viene con un amplio conjunto de bibliotecas de modelos integradas que contienen componentes predefinidos y modelos que representan varios sistemas físicos. Esta función de biblioteca facilita el ensamblaje de modelos complejos.
- Cosimulación e integración: Dymola admite la cosimulación, lo que permite a los usuarios combinar modelos de diferentes dominios o herramientas para un análisis más completo a nivel de sistema. También se integra con otras herramientas de ingeniería, como el software CAD y las herramientas de optimización.
- Personalización y extensibilidad: Los usuarios pueden ampliar la funcionalidad de Dymola mediante la creación de bibliotecas, componentes o funciones personalizadas. Esto permite la personalización para cumplir con los requisitos específicos de modelado y simulación.
- Optimización y estimación de parámetros: Dymola proporciona herramientas para la optimización y la estimación de parámetros, lo que permite a los usuarios refinar los parámetros del modelo para que coincidan mejor con los datos experimentales u optimizar el rendimiento del sistema.
- Simulación en tiempo real: Algunas versiones de Dymola admiten la simulación en tiempo real, lo que la hace útil para aplicaciones en las que la simulación debe ejecutarse en sincronía con procesos del mundo real, como las pruebas de hardware-in-the-loop (HIL).

Dymola es compatible con una amplia gama de bibliotecas que cubren varios dominios, lo que permite a los usuarios modelar y simular sistemas complejos en diferentes industrias. A continuación, se muestran algunos ejemplos:

- Modelica Standard Library: Dymola suele incluir la Modelica Standard Library, que proporciona un conjunto fundamental de componentes para modelar fenómenos físicos fundamentales. Esta biblioteca está estandarizada en todos los entornos de simulación compatibles con Modelica.
- Bibliotecas de sistemas mecánicos: Dymola a menudo admite bibliotecas para modelar sistemas mecánicos, incluidos componentes como engranajes, resortes, amortiguadores y otros elementos mecánicos.
- Bibliotecas eléctricas: Las bibliotecas para modelar sistemas eléctricos son comúnmente compatibles. Estos pueden incluir componentes para resistencias, condensadores, inductores, transformadores y más.
- Bibliotecas térmicas: Se incluyen bibliotecas para sistemas térmicos, que cubren componentes como intercambiadores de calor, tuberías, válvulas y otros elementos relevantes para el modelado de la transferencia de calor y la dinámica térmica.

- Bibliotecas de sistemas de fluidos: Dymola suele admitir bibliotecas para modelar sistemas de fluidos, que abarcan componentes como bombas, válvulas, tuberías y elementos de flujo de fluidos.
- Bibliotecas de sistemas de control: A menudo están disponibles bibliotecas para modelar sistemas de control, lo que permite a los usuarios incluir controladores, sensores, actuadores y otros elementos en sus modelos.
- Bibliotecas automotrices: Dymola se usa comúnmente en la industria automotriz y puede incluir bibliotecas especializadas para modelar la dinámica de vehículos, motores, transmisiones y otros sistemas automotrices.
- Bibliotecas aeroespaciales: Es posible que se admitan bibliotecas específicas de la industria aeroespacial que ofrezcan componentes para modelar la dinámica de las aeronaves, los sistemas de propulsión y otros sistemas relacionados con la industria aeroespacial.
- Bibliotecas hidráulicas: A menudo se incluyen bibliotecas para sistemas hidráulicos, que proporcionan componentes como bombas, válvulas, cilindros y propiedades del fluido hidráulico.
- Bibliotecas de sistemas de potencia: Dymola puede admitir bibliotecas para modelar sistemas de potencia, incluidos componentes para generadores, transformadores, líneas de transmisión y otros elementos.
- Bibliotecas de energía renovable: Es posible que haya bibliotecas relacionadas con sistemas de energía renovable, como paneles solares, turbinas eólicas y componentes de almacenamiento de energía.

Dymola permite cargar y utilizar datos de diferentes formatos de archivo, accediendo a servidores de datos.

Dymola proporciona varias herramientas y funciones para visualizar datos de simulaciones. Los usuarios pueden analizar e interpretar los resultados de la simulación a través de representaciones gráficas, gráficos y otras técnicas de visualización. También puede integrarse con otras herramientas de visualización.

Los datos se pueden exportar desde simulaciones, e incluso posprocesar los datos de entrada para exportarlos o reutilizarlos en una simulación. Son posibles diferentes formatos de archivo, incluidos archivos .csv, FMU y archivos MAT.

Dymola es una herramienta de modelado y simulación basada en escritorio, y no está diseñada como una aplicación basada en la web.





3. ENTORNOS DE EJECUCIÓN

Python

Python es un lenguaje de programación versátil accesible tanto para programadores experimentados como para estudiantes novatos. Su adaptabilidad y robusto desarrollo lo hacen adecuado para proyectos de diversas escalas. Ampliamente utilizado por su eficiencia, Python simplifica la depuración con sus características integradas, mejorando así la productividad y la eficiencia del programador.

Desde 2014, Python se ha clasificado constantemente entre los principales lenguajes de codificación y se usa ampliamente en los cursos de informática en las universidades de los Estados Unidos. El dominio de Python se ha vuelto casi indispensable para obtener títulos que requieran conocimientos de codificación o informática, particularmente en campos como el análisis de datos. Su prevalencia en los programas académicos asegura su relevancia en el mercado laboral posterior a la graduación.

La adopción generalizada de Python en las instituciones educativas garantiza su impacto duradero en los alumnos. Ya sea que se trate de su primer lenguaje de programación o que se haya adquirido en una institución de nivel superior, Python tiene un valor significativo debido a su prominencia en la educación de nivel universitario. Incluso si los estudiantes cursan diferentes carreras o aprenden otros lenguajes de codificación, Python sigue siendo muy apreciado por su influencia y utilidad en entornos académicos y profesionales.

Comparativa con Matlab

Python ofrece varias ventajas a los programadores. En primer lugar, es de libre acceso para cualquier persona, lo que permite a las personas motivadas utilizarlo como deseen. Además, Python es conocido por su simplicidad y legibilidad, lo que facilita el aprendizaje y la traducción de ideas en código en comparación con Matlab. Las extensas bibliotecas, listas y diccionarios de Python facilitan la programación organizada, mientras que su estructura modular permite un inicio rápido y una introspección sencilla. La manipulación de cadenas también se simplifica en Python, lo que contribuye a su eficiencia.

Además, la compatibilidad de Python en diferentes sistemas, incluidos Windows, Linux y OSX, garantiza una accesibilidad generalizada. Los programadores pueden definir funciones y clases según sea necesario, y Python proporciona una variedad de kits de herramientas GUI para crear aplicaciones personalizadas. A diferencia de Matlab, el código Python es más compacto, legible y menos propenso a errores debido a su estructura de bloques basada en sangría y a su sintaxis clara para la indexación y las funciones.

Python supera a Matlab en varios aspectos. En particular, la indexación de base cero de Python se alinea con la mayoría de los lenguajes de programación, lo que simplifica la traducción para los usuarios. La programación orientada a objetos (POO) de Python es más sencilla en comparación con el complejo esquema de Matlab. Además, la naturaleza de código abierto de Python fomenta las contribuciones de la comunidad y ofrece una gama más amplia de paquetes gráficos y conjuntos de herramientas, a diferencia del modelo de desarrollo cerrado de Matlab. Además, la declaración de importación de Python carece de una contraparte de Matlab, lo que mejora aún más su versatilidad y adaptabilidad.

Una ventaja significativa es el enfoque especializado de Matlab en la computación numérica y el análisis técnico. Matlab se desarrolló inicialmente como un paquete de manipulación de matrices, lo que lo hace particularmente hábil en el manejo de operaciones matemáticas y tareas de análisis de datos comunes en ingeniería, física y otras disciplinas técnicas. Además, Matlab proporciona un conjunto completo de herramientas y bibliotecas específicamente adaptadas a las necesidades de ingenieros y científicos. Estas herramientas cubren una amplia gama de funcionalidades, incluido el procesamiento de señales, el procesamiento de imágenes, el diseño de sistemas de control y la optimización, entre otros. Esta amplia colección de herramientas agiliza el proceso de desarrollo y permite a los usuarios abordar problemas técnicos complejos de manera eficiente. Además, Matlab ofrece una interfaz gráfica de usuario (GUI) integrada que simplifica la visualización de datos, el desarrollo de aplicaciones y la exploración interactiva de algoritmos. Esta GUI permite a los usuarios crear gráficos interactivos, diseñar interfaces gráficas de usuario para sus aplicaciones y visualizar datos de forma clara e intuitiva, sin necesidad de bibliotecas o herramientas externas adicionales. Además, Matlab proporciona un sólido soporte técnico a través de su amplia documentación, tutoriales, foros en línea y servicios de atención al cliente dedicados. Esta infraestructura de soporte ayuda a los usuarios a solucionar problemas, aprender nuevas funciones y técnicas, y mantenerse actualizados con los últimos desarrollos en Matlab.

Finalmente, el rendimiento y la eficiencia de Matlab en términos de ejecución en tiempo de ejecución y utilización de recursos son notables. Para ciertos tipos de tareas computacionales, Matlab puede superar a Python y otras alternativas, especialmente cuando se trata de cálculos numéricos a gran escala o aplicaciones críticas en el tiempo.

Entorno de co-simulación SIMULINK

Si bien Simulink se utiliza principalmente para simulación independiente, también se puede emplear en un entorno de co-simulación donde se integra con otras herramientas de simulación o plataformas de software para modelar sistemas complejos que abarcan múltiples dominios. Estos modelos pueden ser de diferentes dominios o disciplinas, como sistemas de control, sistemas mecánicos, sistemas eléctricos, etc.

Se desarrollan interfaces para facilitar la comunicación entre Simulink y otras herramientas de simulación. Estas interfaces manejan el intercambio de datos, la sincronización y la coordinación entre los diferentes entornos de simulación.

El intercambio de datos entre Simulink y otras herramientas de simulación ocurre durante la simulación. Esto incluye el intercambio de señales de entrada, parámetros y variables de estado entre los modelos en cada entorno de simulación.

Durante la co-simulación, cada herramienta de simulación ejecuta sus respectivos modelos mientras intercambia datos con otras herramientas. La simulación avanza de forma iterativa, y cada herramienta avanza el tiempo de simulación y actualiza su estado en función de los datos entrantes de otras herramientas.

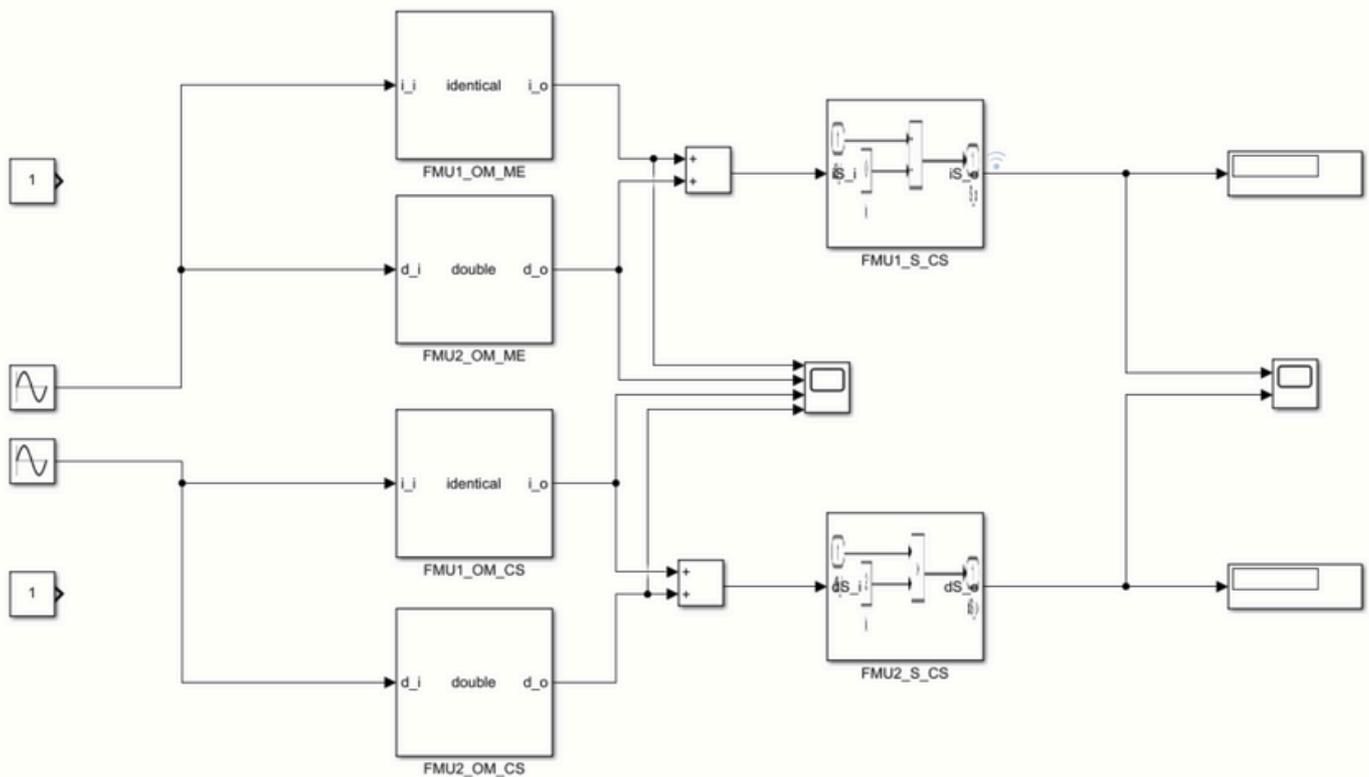
Se emplean mecanismos de sincronización para garantizar que los pasos de tiempo de simulación se alineen en todos los entornos de simulación. Esto garantiza que los datos intercambiados entre las diferentes herramientas sean consistentes y precisos.

El control de la simulación generalmente lo gestiona un entorno de simulación maestro o coordinador. Este entorno organiza la ejecución de cada herramienta de simulación, supervisa el progreso de la simulación y maneja los errores o excepciones que pueden ocurrir durante la co-simulación.

En general, el uso de Simulink como entorno de co-simulación permite a los usuarios modelar y simular sistemas complejos que involucran componentes de diferentes dominios o disciplinas. Este enfoque permite un análisis integral y la optimización del rendimiento del sistema en múltiples dominios.



```
src > Components > C3 > Index.js > ...
1 import {use} from "react"
2 import React useCallba... function React.useCallback() {
3 import Theme useContext
4 useDebugValue
5 function C3 useEffect
6 const the useImperativeHandle
7 return ( useLayoutEffect
8   <div> useMemo
9     <p>So useReducer
10    <p>En useRef
11    <butt useState
12    Cli useCallback
13  </but useContext
```



Estándares para ejecución en cualquier entorno. Encapsulamiento.

Los entornos de desarrollo de los modelos de simulación no son la opción más flexible para la conexión con el PLC o la nube. Por lo que es necesario ejecutar los modelos en entornos distintos a los de desarrollo. Para esto resulta muy útil poder encapsular los modelos en “bloques” cerrados e independientes del entorno de ejecución.

Análisis de FMU, FME.

En co-simulación, las Unidades de Maqueta Funcional (FMUs) son componentes software que encapsulan modelos de simulación o partes de modelos, permitiendo su intercambio e integración con otras herramientas de simulación. Por tanto, una FMU es una representación empaquetada de un modelo que incluye información sobre la estructura del modelo, las variables y el comportamiento de la simulación. Estas FMUs pueden desarrollarse en diversos entornos de modelado y simulación como Amesim, Modelica, Simulink o Dymola.

Amesim puede exportar FMU de cosimulación e intercambio de modelos. Las FMU de cosimulación contienen su propio solver y se comunican con el algoritmo maestro en puntos de comunicación discretos, mientras que las FMU de intercambio de modelos se basan en el solver del algoritmo maestro y se comunican en cada paso de integración. Amesim puede importar FMU de otras herramientas y usarlas como submodelos en una simulación de sistema más grande, así como manejar diferentes versiones y plataformas de FMI. Hay una interfaz gráfica para crear y exportar FMU dentro de Simcenter Amesim. Finalmente, Amesim puede validar la correcta conexión de varios FMU utilizando FMU Compliance Checker.

Simulink cuenta con un bloque FMU Import incorporado, que admite tanto el intercambio de modelos como las FMU de cosimulación. El bloque de importación de FMU selecciona automáticamente el modo de FMU en función de la FMU existente que desea importar. También permite ajustar parámetros, guardar estados y avanzar y retroceder en la simulación, siempre que la propia FMU admita estas funciones. El bloque de importación de FMU puede generar un informe de conectividad para cualquier intercambio de modelos o FMU de cosimulación.

En el caso de Modelica el estándar FMI es independiente. OpenModelica admite la exportación de FMU de acuerdo con FMI 1.0 y FMI 2.0 para FMU de intercambio de modelos y FMI 2.0 para FMU de co-simulación. Además, admite la importación de FMU de acuerdo con FMI 1.0 y 2.0 para el intercambio de modelos y FMI 1.0 independiente para la cosimulación. Cuando se desea una simulación de una sola FMU independiente o una conexión de varias FMU, la herramienta recomendada es OMSimulator.

Dymola también es compatible con el estándar Functional Mock-up Interface (FMI). A continuación, se muestra cómo trabaja Dymola con el estándar FMI:

- Exportación de modelos como FMUs: Dymola permite a los usuarios exportar modelos como Unidades de Maquetas Funcionales (FMUs) de acuerdo con el estándar FMI.
- Versiones de FMI: Dymola es compatible con varias versiones del estándar FMI, incluidas FMI 1.0 y FMI 2.0. Los usuarios pueden elegir la versión de FMI adecuada en función de sus requisitos y de las capacidades de la herramienta de simulación de destino.
- Estructura y variables del modelo: Al exportar un modelo como FMU, Dymola incluye información sobre la estructura del modelo, las variables y sus dependencias. Esta información es esencial para que el entorno de cosimulación interprete y ejecute correctamente el modelo.
- Intercambio de modelos y cosimulación: Dymola admite los modos de intercambio de modelos y cosimulación definidos por el estándar FMI. Las FMU de intercambio de modelos encapsulan todo el comportamiento del modelo, mientras que las FMU de cosimulación proporcionan un enfoque más modular y dinámico, lo que permite la interacción con entornos externos.
- Parametrización e inicialización: Dymola permite a los usuarios parametrizar las FMU durante la exportación, lo que permite crear FMU reutilizables con parámetros configurables. Además, Dymola proporciona mecanismos para especificar las condiciones iniciales a fin de garantizar la inicialización adecuada en escenarios de cosimulación.
- Exportación e importación de FMU: Una vez que se exporta un modelo como FMU desde Dymola, el archivo FMU resultante se puede utilizar en otras herramientas de simulación compatibles con el estándar FMI. Por el contrario, Dymola puede importar FMU creadas por otras herramientas para su posterior análisis o cosimulación.

- Mapeo de variables FMU: Dymola proporciona mecanismos para mapear variables entre el modelo Dymola y el FMU, lo que facilita el intercambio de datos entre el entorno Dymola y las herramientas de cosimulación externas.
- Configuración de simulación FMU: Los usuarios pueden configurar varios parámetros y opciones de simulación al exportar modelos como FMU, incluida la elección de la configuración del solucionador y otras configuraciones relacionadas con la simulación.

Documentación y soporte: Dymola proporciona documentación y soporte para trabajar con FMI, incluidas las pautas sobre la exportación de modelos como FMU, la configuración de los ajustes de exportación y la integración de FMU en escenarios de cosimulación



4. ALMACENAMIENTO

El registro de los datos reales adquiridos de los sensores y de los datos de las simulaciones es fundamental para generar un conjunto de datos históricos valioso para diferentes análisis (¿qué pasó? ¿qué va a ocurrir?). Su almacenamiento en bases de datos accesibles para aplicar técnicas de analítica de datos es importante. Por ello, es necesario definir qué tipo de almacenamiento conviene más dependiendo de los tipos de datos que se van a almacenar. También es necesario

Data Lake

El Data Lake es un repositorio que almacena datos en crudo, sin procesar, de diferente tipo:

- Datos estructurados: se almacenan en una base de datos relacional, como, por ejemplo: MySQL, MariaDB, PostgreSQL, Microsoft SQL Server.
- Datos no estructurados: imágenes o video, texto, manuales, documentos técnicos en PDF.
- Datos semi estructurados: datos sin un esquema fijo, pero con etiquetas, por ejemplo, datos en formato JSON, XML.

El Data Lake estaría formado por diferentes bases de datos, optimizadas para almacenar un tipo de dato en concreto.

En el caso particular de almacenar resultados de simulación, son datos estructurados ya que en los modelos de simulación están definidas cada una de las variables del sistema, tanto entradas como salidas. Los resultados de simulación además tienen asociada una marca de tiempo con lo que cada variable se puede considerar como una serie temporal. Existen bases de datos optimizadas para ser eficientes tanto en espacio de almacenamiento como en velocidad de lectura y escritura almacenando series temporales, como, por ejemplo, InfluxDB.

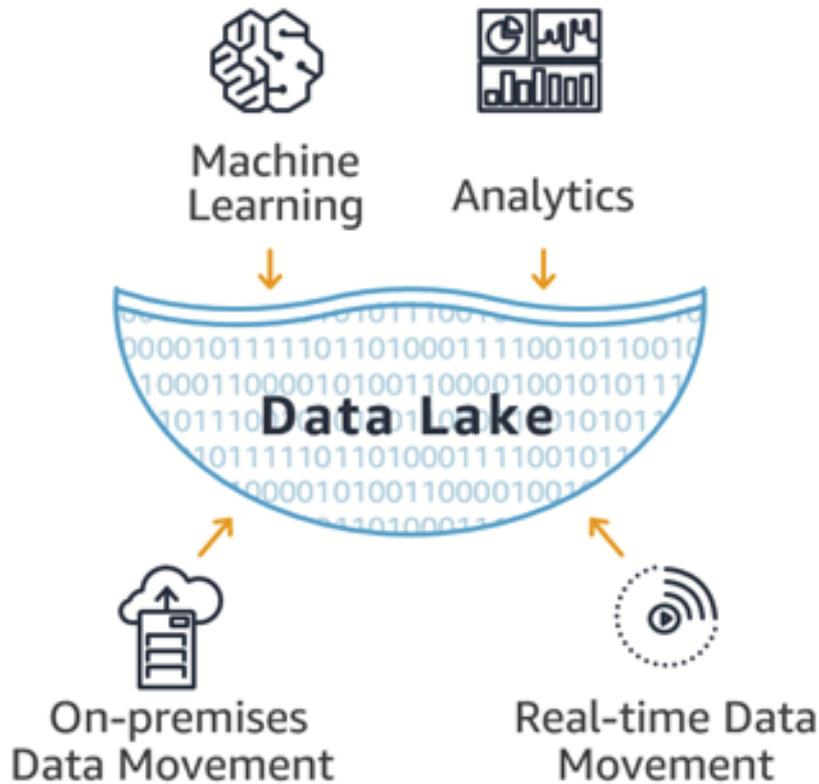


Ilustración . Esquema de un proceso ETL

Consumidores

Los consumidores del Data Lake son las aplicaciones que van a consultar los datos que están almacenados. Por ejemplo, un modelo de Machine Learning puede extraer un volumen de datos grande para usar de conjunto de entrenamiento a través de consultar el Data Lake o consultar los últimos datos en tiempo real que pueden servir como entrada al modelo. El proceso de extraer los datos para crear un conjunto de entrenamiento no es directo, pero una vez que los datos están en el Data Lake, podría automatizarse.

Otro ejemplo de consumidor de los datos sería una aplicación con dashboards para visualizar los datos de la simulación. Estos dashboards se encargarían de realizar las consultas a los datos almacenados y la visualización en gráficas de estos.

La arquitectura del modelo de simulación desempeña un papel fundamental a la hora de determinar la eficiencia, la escalabilidad y la precisión de las simulaciones en varios dominios. La arquitectura adecuada depende de factores como la complejidad del sistema, los requisitos en tiempo real, la integración de datos y las necesidades computacionales.

Analítica (Big data, ML/DL)

El componente analítico de un gemelo digital se basa en tecnologías de Big Data y algoritmos de aprendizaje automático (ML) y aprendizaje profundo (DL) para recopilar, procesar y analizar grandes volúmenes de datos. Estas tecnologías le permiten transformar los datos en información procesable y precisa, esencial para una toma de decisiones informada.

En el contexto del Big Data, las “8 V” (volumen, variedad, velocidad, veracidad, variabilidad, validación, valor y visualización) resumen los desafíos y oportunidades en la gestión de datos. El volumen se refiere a la cantidad de datos generados por diferentes fuentes, mientras que la variedad abarca los diferentes tipos y formatos de datos. La velocidad resalta la rapidez con la que se generan y procesan los datos, y la veracidad evalúa su calidad y precisión. La variabilidad refleja cambios constantes en los datos, la validación asegura su adecuación y precisión, el valor representa la utilidad extraída de esos datos y la visualización facilita la interpretación de grandes conjuntos de información.

Al integrar ML y DL, los gemelos digitales pueden automatizar tareas y realizar análisis avanzados. Estas tecnologías permiten aplicaciones como reconocimiento de patrones, predicción de tendencias, detección de anomalías, optimización de procesos en tiempo real, procesamiento del lenguaje natural y segmentación de clientes. La sinergia entre las “8 V” y estas herramientas permite que los gemelos digitales sean soluciones robustas y adaptables para afrontar la complejidad y escala de los datos en el contexto actual.

Tecnologías Big Data

Para abordar los desafíos asociados al Big Data, varias herramientas desempeñan un papel crucial:

Hadoop se utiliza ampliamente para el almacenamiento distribuido y el procesamiento de grandes cantidades de datos. Este enfoque es esencial para superar las limitaciones de las arquitecturas centralizadas en términos de eficiencia, tiempo y costos. Spark, a su vez, ofrece un entorno potente para análisis, compatible con SQL, procesamiento en tiempo real y aprendizaje automático. Utiliza un modelo de programación en memoria que mejora significativamente el rendimiento al minimizar las operaciones de lectura y escritura en disco. Finalmente, Kafka es un sistema de mensajería diseñado para ser escalable, rápido, confiable y duradero. Este sistema es capaz de gestionar grandes volúmenes de datos en tiempo real, con una arquitectura distribuida que garantiza el balanceo de carga y tolerancia a fallos.

En conjunto, estas tecnologías transforman grandes volúmenes de datos en información útil, ayudando a las organizaciones y a los investigadores a obtener información relevante para la toma de decisiones.

Machine Learning

Los modelos de aprendizaje automático juegan un papel fundamental en el análisis y exploración de big data. Estos algoritmos están diseñados para extraer información, predecir tendencias y optimizar operaciones en función de patrones previamente identificados en los datos. En aplicaciones como los gemelos digitales, los modelos ML son esenciales para simular el comportamiento de los sistemas físicos, predecir fallas y mejorar la eficiencia de los recursos.

	Regression	Classification	Clustering	Type
Linear Regression	X			Supervised Learning
Logistic Regression	X			Supervised Learning
Naive Bayes		X		Supervised Learning
Decision trees	X	X		Supervised Learning
K-Means			X	Unsupervised Learning
K-nearest neighbor (KNN)	X	X		Unsupervised Learning
Support Vector Machine (SVM)		X		Unsupervised Learning
DBSCAN (Density-Based Spatial Clustering of Applications with Noise)			X	Unsupervised Learning

Modelos de machine learning (ML)

Los modelos varían según el tipo de problema que se quiera resolver. En la tabla anterior se muestran ejemplos de algoritmos categorizados según las técnicas y el tipo de aprendizaje que utilizan.

Deep Learning

Deep Learning (DL), una subárea del Machine Learning, se destaca por el uso de redes neuronales profundas, compuestas por múltiples capas interconectadas. Estas redes tienen la capacidad de aprender representaciones complejas de datos de forma jerárquica, lo que les permite abordar problemas como el reconocimiento de imágenes, la traducción automática y la previsión de series de tiempo.

A diferencia de los enfoques tradicionales que se basan en la ingeniería de características manual, DL automatiza este proceso identificando patrones ocultos directamente en los datos sin procesar. A pesar de requerir infraestructuras computacionales robustas y grandes volúmenes de datos, el DL continúa revolucionando áreas como la visión artificial y la robótica, gracias a su capacidad de generalizar y lograr precisión en tareas complejas.

Sistemas de Base de Datos para Big Data

Los sistemas de bases de datos juegan un papel central en el manejo de grandes volúmenes de datos. Soluciones como Cassandra y MongoDB destacan por su flexibilidad y escalabilidad, siendo especialmente útiles en entornos que requieren procesamiento distribuido. Estas bases de datos NoSQL ofrecen ventajas como la manipulación ágil de datos, adaptación a cambios frecuentes en la estructura de datos y tiempos de respuesta rápidos en escenarios de alta demanda. Si bien las bases de datos relacionales son indispensables en contextos que requieren una consistencia estricta, las soluciones NoSQL han ganado popularidad por su capacidad de manejar la diversidad y el volumen de los datos modernos.



5. CIBERSEGURIDAD

Las aplicaciones del gemelo digital introducen retos a nivel de la ciberseguridad debido a su dependencia de los dispositivos IoT, el cloud computing o la IA.

Es por esto que se debe garantizar la integridad del dato y su autenticidad cuando se trabaja con datos procedentes de sensores en tiempo real. Se debe considerar la posibilidad de ataques MitM (Man in the middle) que pueden manipular las comunicaciones entre el sistema físico y el sistema virtual.

La gestión de los accesos es fundamental para minimizar los ataques. Puede haber múltiples accesos al gemelo digital (operadores, fabricantes, terceras partes, etc.), pero cada uno de ellos debe hacerlo con un rol diferentes y con las restricciones adecuadas para que el control del acceso sea estricto. Los métodos de autenticación juegan un papel importante en estos casos.

Se debe prestar especial atención a la seguridad de los dispositivos IoT que formen parte del gemelo puesto que carecen de encriptación, en general, y resultan el punto más vulnerable del sistema.

Hoy en día los despliegues en la nube son cada vez más seguros, minizándose la vulnerabilidad de las APIs, desconfiguración del almacenamiento, etc. Estas plataformas poseen sus propios sistemas de seguridad que impiden las habituales amenazas a la seguridad de los datos, etc.

Sin embargo, se pueden incrementar los niveles de seguridad aplicando un conjunto de buenas prácticas:

- **Comunicación y cifrado seguros**

Se puede utilizar un cifrado de extremo a extremo (TLS 1.3, AES-256) para datos en tránsito y en almacenados. Así como APIs seguras con OAuth 2.0 y JWT para realizar la autenticación. Además, se puede implementar lo que se conoce como una Arquitectura de confianza cero (ZTA), donde se verifica el acceso de forma continua, asumiendo que no existe una confianza inherente.

- **Gestión de identidad y de acceso (IAM)**

Se puede aplicar una autenticación multifactor (MFA) para todos los puntos de acceso del gemelo digital. El control de acceso basado en roles (RBAC) y el principio de privilegio mínimo (PoLP) son herramientas eficaces para la gestión del acceso.

- **Detección y monitoreo de amenazas**

Se pueden implementar sistemas de detección de intrusiones (IDS) para monitorear anomalías de la red. Y utilizar herramientas SIEM (gestión de eventos e información de seguridad) para alertas en tiempo real.

- **Desarrollo e implementación seguros de gemelos digitales**

Una buena práctica es auditar y reparar vulnerabilidades de software y firmware de forma regular. O bien encapsular modelos o aplicaciones en contenedores (por ejemplo, Docker, Kubernetes) con estrictos controles de acceso.

- **Integridad de datos y seguridad de la IA**

La integridad del dato debe estar garantizada y se puede conseguir a través del Blockchain o con firmas digitales para verificar la autenticidad de las fuentes de datos.

También es importante disponer de conjuntos de datos de entrenamiento robustos y detección de anomalías.





**IoT: datos, sensores,
protocolos de
comunicación**

3





Formato de los datos

La elección del formato de datos para la adquisición de sensores depende de varios factores, incluido el tipo de sensor, la naturaleza de los datos, la compatibilidad con los sistemas existentes y la facilidad de procesamiento. Algunos formatos de datos comunes utilizados para la adquisición de sensores podrían ser:

I. CSV (valores separados por comas):

Un formato simple y ampliamente utilizado basado en texto donde cada línea representa un registro de datos y los valores están separados por comas. CSV es fácil de leer y se puede abrir con software de hoja de cálculo común.

II. JSON (notación de objetos JavaScript):

Un formato de intercambio de datos liviano y legible para humanos. JSON se usa a menudo para datos estructurados y se analiza fácilmente con varios lenguajes de programación. Admite estructuras anidadas y es adecuado para representar datos complejos de sensores.

III. XML (lenguaje de marcado extensible):

Otro formato basado en texto que permite la representación de datos jerárquicos y estructurados. XML es más detallado en comparación con JSON, pero aún se usa ampliamente, especialmente en ciertas industrias y aplicaciones.

IV. Formatos binarios (p. ej., Protocol Buffers, Avro):

Los formatos binarios son más compactos y eficientes para grandes conjuntos de datos en comparación con los formatos basados en texto. Protocol Buffers y Avro son ejemplos de formatos de serialización binaria que ofrecen eficiencia y velocidad.

V. HDF5 (Hierarchical Data Format versión 5):

Un formato de archivo y un conjunto de herramientas para gestionar datos complejos. HDF5 es particularmente útil para almacenar grandes conjuntos de datos con múltiples tipos de datos y jerarquías. Se utiliza comúnmente en aplicaciones científicas y de ingeniería.

VI. Colas de mensajes (p. ej., MQTT, Apache Kafka):

Para datos de sensores en tiempo real o en streaming, se pueden emplear protocolos de cola de mensajes como MQTT o Apache Kafka. Estos protocolos permiten una comunicación eficiente entre sensores y consumidores de datos, y admiten varios formatos de datos.

VII. Parquet:

Un formato de almacenamiento en columnas que es altamente eficiente para grandes conjuntos de datos. Parquet se utiliza a menudo en marcos de procesamiento de big data como Apache Spark y Apache Hive.

VIII. Protocolos binarios de sensores (p. ej., Modbus, OPC-UA):

Dependiendo del tipo de sensores y los protocolos de comunicación que admitan, se pueden utilizar protocolos binarios de sensores como Modbus u OPC-UA para un intercambio de datos eficiente y estandarizado.

IX. Formatos de bases de datos (p. ej., SQL, NoSQL):

El almacenamiento de datos de sensores en bases de datos, ya sean relacionales (SQL) o no relacionales (NoSQL), es común. La elección depende de los requisitos específicos de la aplicación, las necesidades de escalabilidad y la complejidad de los datos.

X. Formatos binarios personalizados:

En algunos casos, los formatos binarios personalizados están diseñados para satisfacer necesidades específicas, optimizando el tamaño de los datos, la velocidad u otros requisitos específicos de la aplicación.

XI. STEP (ISO 10303):

STEP, que significa Estándar para el intercambio de datos de modelos de productos, es un estándar internacional para la representación e intercambio de datos de productos. El estándar es desarrollado y mantenido por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC). STEP se utiliza ampliamente en las industrias de fabricación e ingeniería para facilitar la interoperabilidad de los sistemas de diseño asistido por computadora (CAD), ingeniería asistida por computadora (CAE) y fabricación asistida por computadora (CAM). Este formato es fundamental para la gestión de datos y procesos de simulación (SPDM) y la gestión del ciclo de vida del producto (PLM).

Al seleccionar un formato de datos, es esencial tener en cuenta factores como el tamaño de los datos, la facilidad de integración, la interoperabilidad y las herramientas y plataformas utilizadas para el análisis y procesamiento de datos en su aplicación específica. Además, se debe considerar el cumplimiento de los estándares de la industria y la interoperabilidad con los sistemas existentes.



Requisitos de adquisición

La frecuencia de adquisición del sensor se refiere a la frecuencia con la que un sensor recopila datos en un período de tiempo determinado. La frecuencia de muestreo adecuada del sensor depende de varios factores relacionados con la aplicación específica y las características del fenómeno que se mide. Algunas consideraciones para determinar la frecuencia de adquisición del sensor podrían ser:

- **Frecuencia de muestreo**

La frecuencia de adquisición del sensor se refiere a la frecuencia con la que un sensor recopila datos en un período de tiempo determinado. La frecuencia de muestreo adecuada del sensor depende de varios factores relacionados con la aplicación específica y las características del fenómeno que se está midiendo. Algunas consideraciones para determinar la frecuencia de adquisición del sensor podrían ser:

- **Naturaleza del fenómeno:**

Es necesario considerar la dinámica y las características del fenómeno que se está midiendo. Los fenómenos que cambian o fluctúan rápidamente pueden requerir frecuencias de muestreo más altas para capturar detalles importantes. Para abordar esto, un criterio podría ser el teorema de muestreo de Nyquist-Shannon que establece que la frecuencia de muestreo debe ser al menos el doble del componente de frecuencia más alta de la señal para evitar el aliasing.

- **Rango de amplitud**

Otra variable que influye en la selección de un sensor son los rangos de amplitud. Este rango de amplitud en un sensor se refiere al rango de valores que el sensor es capaz de medir. Representa el intervalo entre los valores mínimos y máximos que el sensor puede detectar o medir con precisión. Este rango suele especificarse en la hoja de datos o las especificaciones técnicas del sensor y es un parámetro importante para comprender las capacidades y limitaciones del sensor.

Es fundamental elegir un sensor con un rango de amplitud que se adapte a los requisitos específicos del equipo. Si el sensor está expuesto a valores que superan su rango especificado, puede proporcionar mediciones inexactas o poco confiables.

Para obtener detalles específicos sobre el rango de amplitud de un sensor, se recomienda consultar la hoja de datos del sensor o la documentación técnica proporcionada por el fabricante. La hoja de datos generalmente incluye información sobre el rango operativo del sensor, la sensibilidad y otras características de rendimiento. Una mala selección podría resultar en la imposibilidad de capturar los fenómenos.

- **Requisitos de ancho de banda:**

Evalúe el ancho de banda de la señal de interés. Si el fenómeno tiene un amplio rango de frecuencias, se necesita una mayor tasa de muestreo para representar fielmente la señal sin aliasing.

- **Almacenamiento y transmisión de datos:**

Es necesario considerar la capacidad de almacenamiento y transmisión de datos. Las tasas de muestreo más altas generan más datos, lo que puede plantear desafíos para el almacenamiento, procesamiento y transmisión, especialmente en barcos que son entornos con recursos limitados.

- **Requisitos en tiempo real:**

Es necesario determinar si es necesario el monitoreo o control en tiempo real. Algunas aplicaciones, como los sistemas de control de retroalimentación, pueden requerir una alta tasa de muestreo para responder rápidamente a los cambios en el entorno.

- **Filtrado y procesamiento de señales:**

En algunos casos, las tasas de muestreo más bajas combinadas con el procesamiento de señales adecuado pueden ser suficientes para extraer información relevante. Por lo tanto, las técnicas de filtrado y procesamiento de señales deben estar disponibles para algunos modelos. En este caso, tenemos que considerar la gestión de los datos como una combinación de dos fuentes diferentes (gestión de conjuntos de datos híbridos).



Sensorización de equipos

Los barcos modernos están equipados con sistemas de monitoreo avanzados que rastrean continuamente una amplia gama de parámetros operativos. Para los equipos accionados eléctricamente, las mediciones clave incluyen voltaje, corriente y potencia, que ayudan a garantizar que los sistemas eléctricos estén funcionando dentro de límites seguros y eficientes.

Para los equipos que funcionan con fluidos o gases, los sensores monitorean el caudal y la presión, proporcionando datos cruciales, que ayudan a mantener el funcionamiento adecuado de sistemas como el suministro de combustible, la refrigeración y las operaciones hidráulicas. Este monitoreo asegura que el equipo funcione dentro de los parámetros deseados, evitando posibles problemas como fugas o caídas de presión que podrían comprometer el rendimiento.

Las máquinas rotativas, como bombas y motores, se monitorean a través de sus revoluciones, potencia de salida y torque. Estas mediciones son fundamentales para evaluar la eficiencia y la salud mecánica de la maquinaria. Por ejemplo, las variaciones en el torque o las revoluciones pueden indicar posibles problemas mecánicos, como desgaste o desalineación, que se pueden abordar de manera proactiva antes de que se conviertan en problemas más importantes.

Se hace evidente que una sola pieza de equipo puede presentar una combinación de las características mencionadas anteriormente. Por ejemplo, una bomba puede estar equipada con sensores que controlan su consumo de energía eléctrica, caudal, presión, así como sus revoluciones y par. Al agregar y analizar estos datos completos, los operadores de barcos pueden mejorar el rendimiento de sus equipos, optimizar la eficiencia energética y garantizar la confiabilidad y seguridad de los sistemas del barco.

Este enfoque holístico de la supervisión se hace posible mediante la integración de sensores, actuadores y software de control, que facilita la recopilación de datos en tiempo real, el análisis y la operación remota desde el puente o la sala de control. Además, esta infraestructura de supervisión detallada sirve como una base sólida para implementar un gemelo digital, una réplica virtual de los sistemas del barco. Un gemelo digital se puede utilizar para la supervisión, el control y el mantenimiento de los equipos sin necesidad de instalar nuevos sensores. Al aprovechar los datos existentes, un gemelo digital puede simular diferentes escenarios, predecir las necesidades de mantenimiento y optimizar las operaciones, mejorando aún más la eficiencia y la seguridad de las operaciones marítimas.



Protocolos de comunicación: sensores - dispositivos IoT, dispositivos - nube

Al trabajar con datos de sensores en tiempo real o en streaming, es fundamental utilizar los protocolos adecuados para una comunicación de datos eficiente y fiable. A continuación, se indican algunos protocolos habituales que se utilizan para gestionar datos de sensores en tiempo real o en streaming:

1. MQTT (Message Queuing Telemetry Transport):

MQTT es un protocolo de mensajería ligero y eficiente diseñado para redes de bajo ancho de banda, alta latencia o poco fiables. Se utiliza ampliamente en Internet de las cosas (IoT) para la comunicación en tiempo real entre sensores y aplicaciones. MQTT es un protocolo estándar ISO ampliamente adoptado para IoT. Es un protocolo de mensajería extremadamente sencillo y ligero, ideal para redes limitadas. Estas características lo convierten en una buena opción para recopilar datos de barcos de varios sensores, integrar datos de barcos con sistemas de logística, mantenimiento y cadena de suministro en tierra, generar alertas y notificaciones, y alimentar sistemas de análisis de datos. La capacidad de MQTT para gestionar conexiones de red intermitentes o inestables es inestimable para mantener la comunicación y el intercambio de datos en condiciones marítimas difíciles.

2. WebSocket:

WebSocket es un protocolo de comunicación que proporciona canales de comunicación full-duplex a través de una única conexión de larga duración. Se utiliza habitualmente para aplicaciones web en tiempo real, incluidas aquellas que implican la transmisión de datos de sensores.

3. CoAP (Constrained Application Protocol):

CoAP es un protocolo diseñado para dispositivos y redes con recursos limitados, lo que lo hace adecuado para aplicaciones de IoT. Es un protocolo ligero que se puede utilizar para la comunicación en tiempo real entre sensores y servidores.

4. AMQP (Advanced Message Queuing Protocol):

AMQP es un protocolo de mensajería que permite la comunicación entre diferentes componentes en un sistema distribuido. Admite mensajería asíncrona y es adecuado para el intercambio de datos en tiempo real en varias aplicaciones.

5. HTTP/2 y HTTP/3:

Las últimas versiones del protocolo HTTP, HTTP/2 y HTTP/3, introducen mejoras en términos de rendimiento y compatibilidad con la comunicación en tiempo real. Permiten la multiplexación, lo que reduce la latencia y mejora la eficiencia.

6. DDS (Data Distribution Service):

DDS es un protocolo de middleware diseñado para sistemas de misión crítica y en tiempo real. Proporciona un modelo de publicación-suscripción para distribuir y suscribirse a flujos de datos de manera escalable y confiable.

7. RTP (Real-time Transport Protocol):

RTP es un protocolo utilizado para entregar audio y video a través de redes IP. Se emplea comúnmente en aplicaciones en tiempo real que requieren la entrega oportuna de datos de sensores, como transmisión de video o VoIP.

8. Protocolo Kafka:

Apache Kafka es una plataforma de transmisión distribuida y tiene su propio protocolo de comunicación. Kafka se usa ampliamente para construir aplicaciones de transmisión y canalizaciones de datos en tiempo real.

9. STOMP (Simple Text Oriented Messaging Protocol):

STOMP es un protocolo de mensajería simple diseñado para middleware orientado a mensajes. Se puede utilizar para la comunicación en tiempo real entre sensores y aplicaciones, proporcionando un formato basado en texto para el intercambio de mensajes.

10. AMT (Asynchronous Management Protocol):

AMT está diseñado para la comunicación en tiempo real en sistemas de automatización industrial. Admite modelos de comunicación de solicitud-respuesta y publicación-suscripción.

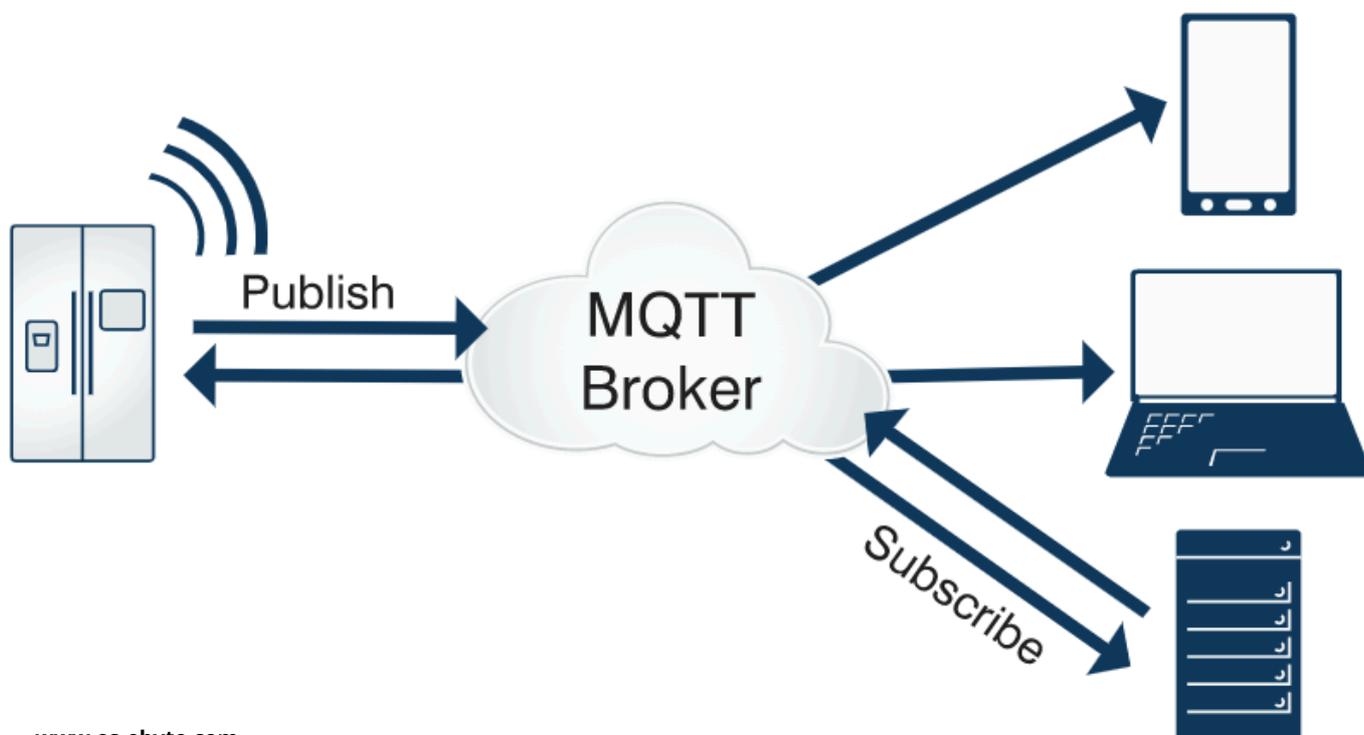
11. XMPP (Extensible Messaging and Presence Protocol):

XMPP es un protocolo para la comunicación en tiempo real que suele asociarse con aplicaciones de mensajería instantánea. Se puede adaptar para su uso en escenarios que requieren el intercambio de datos de sensores en tiempo real.

12. OPC UA (Unified Architecture):

OPC UA es un estándar para la comunicación industrial y la interoperabilidad. Se utiliza ampliamente en la automatización industrial para el intercambio de datos en tiempo real entre sensores y sistemas de control. Este protocolo admite una amplia gama de tipos y estructuras de datos, incluidas estructuras jerárquicas, matrices y tipos definidos por el usuario. Esto permite una fácil integración con diferentes dispositivos y sistemas y de diferentes fabricantes. OPC UA admite la comunicación en tiempo real, lo que es esencial en aplicaciones que requieren tiempos de respuesta rápidos. El intercambio de datos, incluida la multidifusión y el procesamiento por lotes, se realiza dentro del protocolo.

Para seleccionar un protocolo para datos de sensores en tiempo real o en streaming, se deben considerar factores como la latencia, la confiabilidad, la eficiencia del ancho de banda y los requisitos específicos de los sistemas. Además, el protocolo elegido debe estar alineado con las capacidades de los sensores y las capacidades de procesamiento de los modelos de simulación utilizados.





Desarrollo de modelos de simulación

4



El modelo de simulación se puede abordar de diferentes maneras dependiendo de los requisitos del elemento físico o sistema y dependiendo de los resultados esperados. Esto permite elegir diferentes grados de simulación. Una simulación de alto nivel significa desarrollar una simulación con un alto grado de abstracción, dejando de lado muchos detalles y centrándose en tendencias generales. Mientras que las simulaciones de nivel inferior se centran en interacciones detalladas.

Una simulación de alto nivel podría ser rentable si estamos interesados en resultados generales, patrones o dinámicas generales en lugar de componentes individuales detallados. Las simulaciones de alto nivel a menudo simplifican o agregan detalles de bajo nivel para modelar sistemas complejos de una manera más eficiente computacionalmente. Se utilizan en varios campos, como modelos climáticos o económicos, para comprender tendencias, comportamientos o sistemas a gran escala. Las simulaciones militares y estratégicas a menudo utilizan modelos de alto nivel para simular el resultado de batallas, guerras o decisiones estratégicas a gran escala, utilizando juegos de guerra para simular el movimiento de unidades enteras, como batallones o flotas, en lugar de soldados o vehículos individuales. El enfoque está en estrategias generales, asignación de recursos y logística, no en los detalles minuciosos de enfrentamientos o tácticas individuales.

Las simulaciones de alto nivel ayudan a los diseñadores de barcos a evaluar el rendimiento general de un buque, centrándose en resultados generales en lugar de interacciones detalladas a nivel de componente. En lugar de modelar cada interacción de fluidos con el casco a nivel molecular, las simulaciones de alto nivel analizan la resistencia y el arrastre generales que experimenta un barco a medida que se mueve a través del agua. Estas simulaciones evalúan la eficiencia de diferentes formas y tamaños de casco, evaluando métricas como el consumo de combustible, la velocidad y la estabilidad.

Este grado de simulación puede ser útil para modelar el consumo general de combustible y los niveles de emisión del barco en diferentes condiciones operativas, incluidas las velocidades de crucero, la carga y los estados del mar. Al mismo nivel, esta simulación se puede combinar con otro modelo de alto nivel para simular rutas de barcos. La optimización de la ruta de los barcos puede ayudar a reducir el tiempo de viaje y el uso de combustible al identificar las rutas más eficientes a través de los océanos, teniendo en cuenta las variaciones estacionales y los factores ambientales. Al mismo tiempo, las simulaciones de alto nivel pueden evaluar cómo las diferentes condiciones climáticas, como tormentas, vientos fuertes o mares agitados, afectan el viaje de un barco. Estas simulaciones agregan datos meteorológicos y predicen el impacto en el consumo de combustible, la velocidad y la seguridad, lo que permite realizar ajustes en el rumbo o la velocidad del barco.

Este nivel de abstracción aplicado a la simulación podría ser suficiente para obtener resultados significativos en el mantenimiento, la gestión del ciclo de vida o la gestión energética. En este último caso, es posible modelar el uso general de energía del barco considerando cómo interactúan entre sí los distintos sistemas. El enfoque podría estar en la identificación de oportunidades de ahorro de energía en lugar de modelar el flujo de energía detallado a través de cada sistema.

En el mismo nivel se encuentran las simulaciones como la evacuación, el control de daños, la estabilidad del barco y el comportamiento en el mar, donde el enfoque se centra en cómo se comporta el barco en diferentes condiciones del mar o cómo responde el barco a un fallo crítico en lugar de en la hidrodinámica detallada o la mecánica específica de cada componente o sistema.

Las simulaciones de alto nivel son computacionalmente menos intensivas porque no requieren modelar cada pequeño detalle. Esto las hace ideales para simular sistemas a gran escala con muchas partes interactuando, así como para probar diferentes escenarios y comprender sus posibles efectos en un sistema en su conjunto. Por otro lado, este tipo de simulación se basa en simplificaciones y suposiciones que pueden llevar a predicciones inexactas. La agregación de comportamientos o procesos a un alto nivel puede conducir en ocasiones a resultados menos precisos, especialmente en casos en los que las interacciones de bajo nivel tienen efectos significativos en el sistema.

En el caso de la simulación de bajo nivel lo que se hace es una simulación de grano fino que se centra en componentes individuales o interacciones entre sistemas. Este nivel de simulación puede ser más intensivo computacionalmente pero también más preciso en la representación del sistema que se está estudiando. Un claro ejemplo de esto son las simulaciones de dinámica de fluidos (CFD) donde es posible modelar el comportamiento de cualquier flujo (agua de mar, aire, petróleo, etc.). Otro ejemplo son las simulaciones de sensores, donde es necesario modelar cómo interactúan los sensores con su entorno, haciendo un seguimiento de la señal para probar algoritmos de control o rendimiento, y para replicar las condiciones de entornos complejos.

Las simulaciones de bajo nivel en el contexto de los barcos implican un modelado muy detallado de los diversos procesos físicos, mecánicos y ambientales que afectan al funcionamiento del barco. Estas simulaciones se centran en aspectos complejos del comportamiento del barco y sus interacciones con su entorno, desde la dinámica de fluidos y los sistemas de propulsión hasta las tensiones estructurales y los desafíos de navegación. Probablemente la aplicación más conocida son los modelos CFD. Se trata de una potente herramienta capaz de simular la resistencia de las olas, la estela de la hélice, el desprendimiento de vórtices o la cavitación. Al centrarse en estos fenómenos específicos a un nivel de detalle, las formas del casco, los diseños de la hélice o las configuraciones del timón pueden mejorar su rendimiento.

Las simulaciones de bajo nivel también son aquellas basadas en el análisis de elementos finitos (FEA), que se centran en las tensiones y deformaciones que experimentan el casco, la superestructura y otros componentes del barco. Estas tensiones pueden ser inducidas por las olas, debido al impacto de cargas o vibraciones.

En lo que respecta al sistema de propulsión, las simulaciones de bajo nivel abordan la termodinámica del motor, el seguimiento de la transferencia de calor dentro de los cilindros o las fuerzas mecánicas que afectan al eje. Se pueden realizar simulaciones detalladas para modelar la interacción entre las palas de la hélice y el agua, modelando las fuerzas de sustentación y arrastre generadas en cada punto de la superficie de la pala. Lo mismo ocurre con las maniobras del barco, donde los modelos simulan el movimiento del buque en respuesta a los ángulos del timón y los factores ambientales. Para ello, la simulación ambiental que incluye alturas y frecuencias de olas, velocidad y dirección del viento o corrientes puede ser muy útil.

En el campo de los materiales, las simulaciones de bajo nivel tratan de predecir cómo se degradarán los materiales de un barco con el tiempo debido a la exposición a entornos marítimos hostiles. La modelización en detalle de los fenómenos de corrosión puede ayudar a analizar cómo el agua salada, la humedad y la temperatura afectan al casco de acero y los componentes del barco con el tiempo, prediciendo cómo se desarrollará la corrosión e identificando áreas críticas que necesitan protección o mantenimiento. O, en el caso de los recubrimientos, las simulaciones de bajo nivel pueden modelar la eficacia de diferentes tipos de recubrimientos protectores (por ejemplo, pintura anticorrosión o capas antiincrustantes) para prevenir daños en el casco y los sistemas del barco.

Las simulaciones de bajo nivel pueden brindarnos resultados precisos y modelos realistas de sistemas complejos. Esto las convierte en herramientas valiosas para aplicaciones en las que pequeñas diferencias en el comportamiento pueden tener impactos significativos (daños en el casco, inundaciones, etc.). Aparte de eso, estas simulaciones pueden revelar información que las abstracciones de nivel superior podrían pasar por alto (comportamiento de fluidos dentro de un sistema) y, por lo tanto, obtener predicciones más precisas. Por otro lado, este tipo de modelos pueden ser computacionalmente costosos, especialmente para sistemas grandes con muchos componentes interactuando. Y también, su complejidad puede dificultar su desarrollo y validación. Por estas razones, se suelen utilizar cuando los detalles finos son importantes para comprender el comportamiento del sistema.

Los modelos basados en el conocimiento (knowledge-based), a menudo asociados con sistemas expertos o sistemas basados en reglas, se basan en un conjunto predefinido de reglas, lógica y conocimiento específico del dominio para tomar decisiones.

En algunos casos, los modelos basados en el conocimiento pueden incorporar datos históricos para refinar las reglas o realizar adaptaciones. Esto es especialmente relevante en sistemas que aprenden y evolucionan con el tiempo.

Los modelos basados en conocimiento que se incluyen en los gemelos digitales son modelos dinámicos, por lo que los datos de series temporales son un componente importante en el modelo de simulación. Los datos de series temporales implican observaciones o mediciones recopiladas a lo largo del tiempo; estos datos serán datos de entrada para los modelos de simulación que tienen como objetivo replicar procesos del mundo real a través de representaciones matemáticas o computacionales. Cuando trabajamos con datos de series temporales, necesitamos algunos requisitos:

I. Recopilación y preprocesamiento de datos:

Los datos de series temporales deben limpiarse y marcarse con la fecha y hora adecuadas. Este último problema dependerá de la frecuencia de muestreo de cada parámetro. El problema surge cuando las señales de los sensores tienen diferentes frecuencias de muestreo, pero necesitamos la misma marca de tiempo para todas ellas en el conjunto de datos de un modelo.

II. Validación y calibración del modelo:

Es muy recomendable validar los modelos de simulación, y esto es posible mediante el uso de datos de series temporales históricas. Esto ayuda a garantizar que el modelo represente con precisión la dinámica y los comportamientos del sistema subyacente. Lo mismo se aplica a la calibración de los modelos de simulación utilizando datos de series temporales. Esto es necesario para ajustar los parámetros y mejorar la precisión.

III. Estimación y pronóstico de parámetros

En ocasiones, los datos de series temporales se pueden utilizar para estimar parámetros en modelos de simulación. Esto se puede hacer mediante diversas técnicas, como métodos de optimización o enfoques estadísticos, para mejorar la fidelidad del modelo. De la misma manera, podemos aprovechar los datos de series temporales para desarrollar modelos de pronóstico que predigan valores futuros. Los modelos de simulación pueden proporcionar información sobre los procesos subyacentes que impulsan las tendencias observadas. En cualquier caso, la misma taxonomía de datos que los datos de series temporales se debe aplicar a estos conjuntos de datos con diferentes fuentes.

IV. Integración de modelos:

La gestión de datos cuando se tienen varios modelos integrados puede ser todo un desafío. El enfoque más simple puede ser priorizar los datos para su uso selectivo por parte de cada modelo. Esto lleva a definir una jerarquía de datos según el tipo de integración.

V. Clasificación de datos por escenarios

Los modelos de simulación se pueden utilizar para explorar diferentes escenarios y sus posibles impactos en el sistema. Por lo tanto, es necesario clasificar los datos de series temporales (u otro tipo de datos) por escenarios (funcionamiento normal, falla, etc.). Cada clase debe tener características definidas como condiciones iniciales, FTA, FMCA, etc.

Dependiendo de la orquestación del modelo y la implementación de la arquitectura, podemos definir relaciones entre diferentes elementos de datos. Esto podría implicar relaciones jerárquicas o dependencias.

Ejemplo:

Se puede desarrollar un modelo KBM con el objetivo de realizar predicciones de consumo de combustible a partir de ciertos datos de entrada. Para esto hay que disponer de varios sensores en el motor propulsor. El trabajo de selección de los sensores que estarán disponibles se puede optimizar con la ayuda de los requisitos del modelo KBM.

El modelo KBM, en este caso, está construido de tal manera que puede reproducir el funcionamiento de un motor diésel a partir de mediciones de potencia mecánica. Esta potencia mecánica se obtiene con sensores de par y velocidad en el motor. A continuación se muestra una lista de entradas y variables de estado disponibles para el modelo.

Entradas:

- Potencia mecánica: requiere la instalación de sensores de par y velocidad para el eje del motor del generador en la embarcación.

VARIABLES DE ESTADO:

- Consumo de combustible.
- Potencia eléctrica generada.
- Carga del motor del generador.
- Par y revoluciones del motor.
- Temperatura de escape.
- Otras variables de estado. El modelo puede mostrar otras variables de estado como aceleraciones del eje, variables del motor y variables del modelo del generador. Además, este modelo de generador se puede ampliar, con el objetivo de tener acceso a un mayor número de variables de estado. Una de las mayores ventajas de los modelos KBM (Knowledge Based Model) es la posibilidad de acceder a un mayor número de variables de estado que un modelo DDM. Además, todas las variables de estado a las que se puede acceder desde un modelo DDM requieren que se hayan realizado medidas previas mediante sensores colocados en el equipo real.

ML MODELS

El aprendizaje automático (ML) representa un subárea de la inteligencia artificial que tiene como objetivo permitir que los sistemas informáticos aprendan y mejoren a partir de los datos, sin depender de una programación explícita para cada tarea. Este campo se basa en el desarrollo de algoritmos capaces de identificar patrones en datos y, por tanto, realizar predicciones o tomar decisiones. Esta capacidad de aprendizaje automático se ha utilizado ampliamente en aplicaciones que van desde sistemas industriales hasta servicios de recomendación personalizados.

Además, el uso de modelos ML en gemelos digitales permite la exploración de “escenarios hipotéticos”, como la simulación de condiciones específicas y el impacto que podrían tener en el rendimiento del sistema. Estas capacidades dan como resultado decisiones más informadas, mayor eficiencia operativa, reducción de costos y mejora de la competitividad de las organizaciones.

Aprendizaje supervisado

El aprendizaje supervisado es un subárea del aprendizaje automático que utiliza datos etiquetados para entrenar algoritmos para realizar predicciones o clasificaciones. Este tipo de aprendizaje se basa en conjuntos de datos de entrenamiento donde las variables de entrada están asociadas con una etiqueta o valor conocido. Durante el proceso de entrenamiento, el modelo ajusta sus parámetros internos para minimizar el error entre las predicciones y las etiquetas conocidas. Una vez entrenado, el modelo se puede aplicar a nuevos datos para realizar predicciones con un alto nivel de precisión.

El aprendizaje supervisado se utiliza ampliamente en una variedad de aplicaciones, como el diagnóstico médico, la previsión de la demanda, la detección de fraudes y el mantenimiento predictivo. Este método se puede subdividir en dos tipos principales de modelos: regresión y clasificación.



Modelos de Regresión

Los modelos de regresión se utilizan para predecir variables continuas, como la temperatura de un motor o la vida útil restante de un equipo. Estos modelos analizan la relación entre variables de entrada (independientes) y una variable de salida (dependiente), buscando identificar patrones o tendencias que puedan ser descritos por una relación funcional (lineal o no lineal).

Por ejemplo, la regresión lineal supone que la relación entre variables puede expresarse mediante una línea recta, mientras que los modelos más avanzados, como la regresión polinomial o los modelos basados en redes neuronales, capturan relaciones más complejas.

La aplicación de estos modelos es común en escenarios donde existe la necesidad de realizar previsiones cuantitativas, como la previsión financiera o el análisis del desgaste de equipos industriales.



Modelos de Clasificación

Los modelos de clasificación, por otro lado, están diseñados para predecir variables categóricas. El objetivo es asignar etiquetas o categorías a los datos de entrada según patrones aprendidos durante el entrenamiento. Estos modelos utilizan técnicas estadísticas, probabilidad y algoritmos de aprendizaje automático, como árboles de decisión, máquinas de vectores de soporte (SVM) y redes neuronales, para diferenciar entre categorías.

Por ejemplo, en el mantenimiento predictivo, se pueden utilizar modelos de clasificación para predecir si un equipo está en un estado “normal”, “degradado” o “falla inminente”. Este tipo de modelos son esenciales en aplicaciones de detección de fraude, reconocimiento de imágenes y clasificación de documentos.



Aprendizaje no supervisada

A diferencia del aprendizaje supervisado, el aprendizaje no supervisado funciona con datos no etiquetados. Este método analiza de forma única las variables de entrada para identificar patrones, grupos o distribuciones ocultos en los datos. El aprendizaje no supervisado es particularmente útil en situaciones donde no hay suficiente información para etiquetar los datos o donde el volumen de datos es tan alto que el etiquetado manual se vuelve poco práctico.

Estos métodos se utilizan a menudo para el análisis exploratorio, ayudando a los investigadores e ingenieros a comprender mejor la estructura subyacente de los datos. El aprendizaje no supervisado también juega un papel esencial en la reducción de la dimensionalidad, donde se eliminan las variables redundantes o irrelevantes para mejorar el rendimiento de los modelos posteriores.



Modelos de agrupamiento

Los modelos de agrupamiento están diseñados para dividir los datos en grupos (conglomerados) en función de similitudes o proximidad. El objetivo es maximizar la similitud entre elementos del mismo clúster y minimizar la similitud entre diferentes clústeres. Las técnicas comunes incluyen K-Means, que asigna datos al grupo más cercano en función de la distancia promedio, y DBSCAN, que identifica grupos en función de la densidad de datos.

Estos modelos tienen aplicaciones en marketing (segmentación de clientes), biología (clasificación de especies) y mantenimiento predictivo (identificación de patrones operativos anómalos). Su capacidad para identificar grupos ocultos los hace ideales para explorar conjuntos de datos complejos.



Modelos de series temporales

Los modelos de series de tiempo están diseñados para manejar datos organizados cronológicamente, analizando tendencias, patrones estacionales y dependencias temporales. Estos modelos son ampliamente utilizados para predecir valores futuros como ventas mensuales, consumo de energía o fallas de equipos.

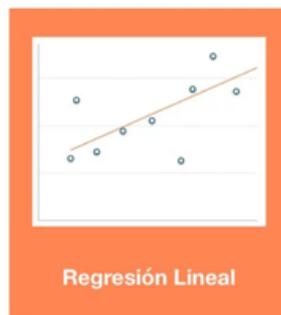
Los modelos clásicos como ARIMA (promedio móvil integrado autorregresivo) se utilizan a menudo para capturar dependencias lineales en series de tiempo. Más recientemente, se han aplicado técnicas basadas en redes neuronales como LSTM (memoria a largo plazo a corto plazo) para abordar dependencias no lineales y complejas. Estos modelos son esenciales en dominios como la previsión meteorológica, el análisis del mercado financiero y la monitorización de equipos industriales.



Modelos de redes neuronales profundas

Las redes neuronales profundas representan una clase avanzada de modelos que utilizan múltiples capas de neuronas artificiales para aprender representaciones jerárquicas de datos. Cada capa aprende progresivamente características más abstractas, lo que permite al modelo identificar patrones complejos en datos sin procesar, como imágenes, sonido o texto.

Los modelos descritos anteriormente se utilizan ampliamente en visión artificial, procesamiento del lenguaje natural y aplicaciones de detección de anomalías. Por ejemplo, en el mantenimiento predictivo, se pueden utilizar redes neuronales profundas para detectar patrones imperceptibles que indican fallas inminentes en los equipos. La capacidad de estas redes para manejar grandes volúmenes de datos y complejidad las hace indispensables en muchas áreas de la ciencia y la ingeniería.





Almacenamiento

5



La implementación de un gemelo digital requiere una infraestructura de almacenamiento de datos robusta para capturar, procesar y hacer que la información esté disponible en tiempo real. La elección de la tecnología de almacenamiento depende del volumen, la velocidad y la variedad de los datos generados. A continuación, se describen las principales opciones utilizadas en este contexto, con detalles sobre su aplicabilidad, ventajas y desafíos.



Base de Datos

Las bases de datos juegan un papel crucial en el almacenamiento y la gestión de datos estructurados y semiestructurados de un gemelo digital. Garantizan la integridad y disponibilidad de los datos para el análisis y la toma de decisiones.

Bases de datos relacionales (SQL)

Las bases de datos relacionales son ideales para almacenar datos organizados en tablas, garantizando la consistencia y la integridad referencial. Son ampliamente utilizadas en aplicaciones industriales y empresariales. Existen varias tecnologías que permiten la implementación y almacenamiento de datos relacionales:

- PostgreSQL: de código abierto, robusto y extensible, admite JSON y procesamiento avanzado de consultas.
- MySQL: Ampliamente utilizado en aplicaciones web y escalable para grandes volúmenes de datos.
- Microsoft SQL Server: Se integra bien con los productos de Microsoft y ofrece alta disponibilidad.
- Oracle Database: Recomendado para sistemas críticos que requieren alto rendimiento y seguridad.



Bases de Datos NoSQL

Para aplicaciones que requieren mayor flexibilidad y escalabilidad horizontal, las bases de datos NoSQL son una excelente opción. Son ideales para manejar grandes volúmenes de datos no estructurados y para aplicaciones que requieren alta disponibilidad.

- MongoDB: Basado en documentos JSON, facilita el modelado flexible de datos.
- Cassandra: Excelente para aplicaciones distribuidas que necesitan tolerancia a fallas.
- Redis: se utiliza como caché de alto rendimiento y base de datos de clave-valor.
- Neo4j: Basado en gráficos, adecuado para modelar relaciones complejas.



Data Lake y Datacenter

El concepto de Data Lake y Datacenter es esencial para gestionar grandes volúmenes de datos, garantizando su accesibilidad y seguridad.

Los Data Lakes son repositorios escalables donde los datos se almacenan en su forma original, lo que permite realizar análisis avanzados, aprendizaje automático e integración con diversas fuentes de datos.

- Apache Hadoop: Plataforma distribuida para almacenar y procesar big data.
- Amazon S3: almacenamiento de objetos escalable y de alta disponibilidad.
- Azure Data Lake: optimizado para análisis avanzados e integración con servicios de IA.
 - Google Cloud Storage: solución flexible integrada en el ecosistema de aprendizaje automático de Google.

Los Centros de Datos son infraestructuras físicas dedicadas al procesamiento y almacenamiento de datos, garantizando una alta seguridad y confiabilidad.

- Dell EMC: Soluciones de almacenamiento empresarial escalables.
- IBM Storage: Alto rendimiento para almacenamiento de datos críticos.
- NetApp: almacenamiento híbrido para la nube y en las instalaciones.



Cloud Computing

Las soluciones de almacenamiento en la nube ofrecen escalabilidad, seguridad e integración con servicios de análisis y aprendizaje automático, lo que permite la construcción de gemelos digitales más eficientes.

Servicios web de Amazon (AWS)

AWS proporciona una infraestructura confiable y escalable para almacenar y procesar datos.

AWS S3: almacenamiento de objetos distribuidos.

·AWS RDS: Bases de datos administradas para SQL.

·AWS DynamoDB: NoSQL de alta disponibilidad.

·AWS IoT Core: Conectividad y procesamiento de datos de sensores.



Microsoft Azure

La plataforma Azure proporciona soluciones integradas de almacenamiento y procesamiento de datos para gemelos digitales.

- Azure Blob Storage: almacenamiento de objetos escalable.
- Azure SQL Database: base de datos relacional administrada.
- Cosmos DB: NoSQL con soporte para diferentes modelos de datos.
- Azure Digital Twins: Plataforma para modelar y simular gemelos digitales.



Google Cloud Platform (GCP)

GCP ofrece servicios avanzados para el almacenamiento y análisis de datos.

- Google Cloud Storage: almacenamiento de objetos confiable.
- BigQuery: Procesamiento y análisis de grandes volúmenes de datos.
- Firestore: base de datos NoSQL para aplicaciones escalables.
- IoT Core: Procesamiento de datos de sensores conectados.

IBM Cloud y Oracle Cloud

Estas plataformas ofrecen soluciones híbridas y empresariales con soporte para bases de datos escalables y seguridad avanzada.

- IBM Cloud Object Storage: almacenamiento distribuido para big data.
- Oracle Autonomous Database: Base de datos autónoma de alto rendimiento y seguridad.

La elección de la solución de almacenamiento ideal dependerá de las necesidades del gemelo digital, el volumen de datos generados, las demandas de seguridad y la infraestructura existente. La integración eficiente entre bases de datos, lagos de datos y la nube es esencial para garantizar un flujo continuo de información y permitir un análisis predictivo eficaz. Además, la elección adecuada de las tecnologías afectará directamente la capacidad de escalabilidad y la eficiencia operativa del sistema.





Visualización

6



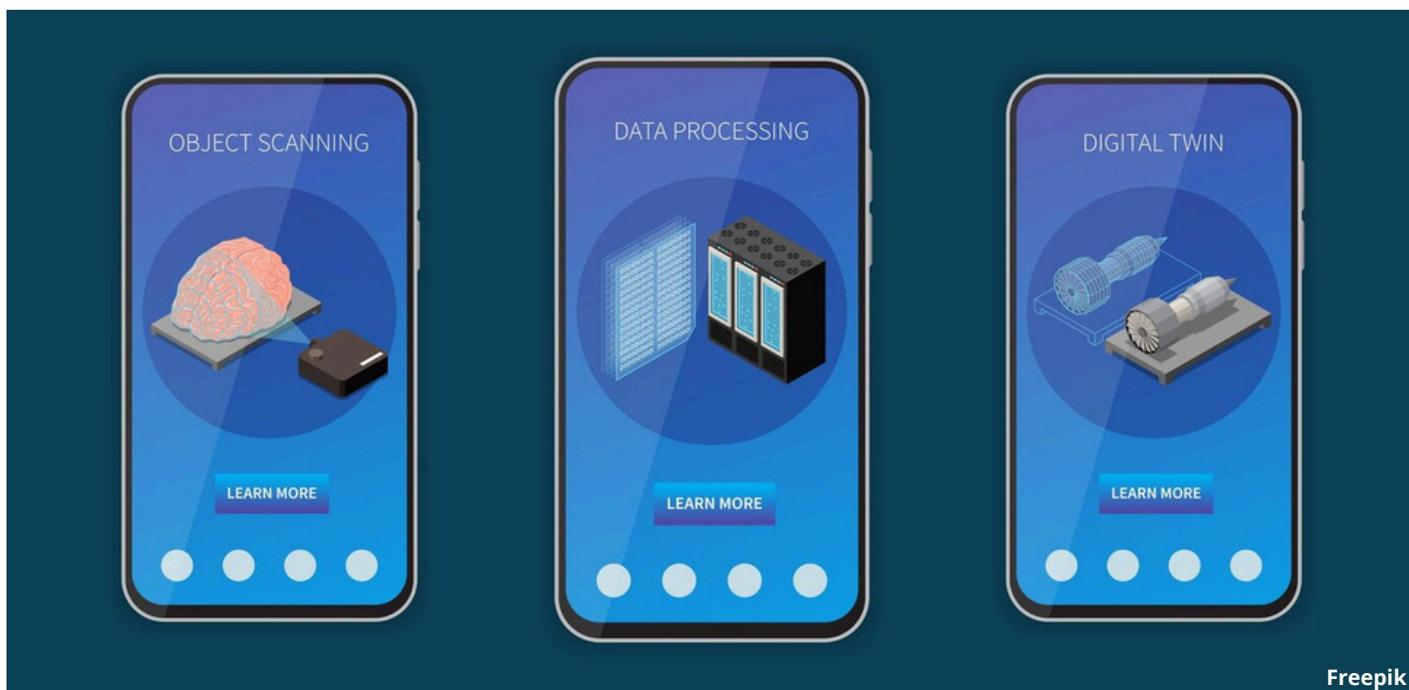
Introducción

La integración de la realidad aumentada en los gemelos digitales representa una evolución significativa en la forma en que interactuamos con los modelos virtuales de sistemas físicos. Este apartado explora cómo la Realidad Aumentada (R.A.) mejora la funcionalidad y el valor de los gemelos digitales, proporcionando beneficios tangibles en múltiples sectores industriales.

Los gemelos digitales han revolucionado la forma en que monitorizamos, analizamos y optimizamos sistemas físicos complejos. La incorporación de la realidad aumentada añade una nueva dimensión a esta tecnología, permitiendo una interacción más intuitiva y eficiente con los datos y modelos virtuales.

Fundamentos tecnológicos

La R.A. superpone información digital sobre el mundo real, creando una experiencia interactiva donde los elementos virtuales y físicos coexisten. En el contexto de los gemelos digitales, la R.A. actúa como puente entre el mundo físico y su representación digital.



Freepik

Beneficios de la integración gemelo digital - realidad aumentada

Visualización Mejorada

- Superposición de datos relevantes en tiempo real sobre equipos físicos
- Visualización tridimensional de parámetros y métricas
- Lo que permite la identificación inmediata de anomalías y desviaciones

Operaciones Optimizadas

- Creación y utilización de guías de mantenimiento interactivas
- Instrucciones paso a paso superpuestas sobre los equipos reales
- Esto lleva a una reducción de errores en procedimientos complejos

Colaboración Mejorada

- Compartición de visualizaciones en tiempo real entre equipos de trabajo
- Asistencia remota con superposición de indicaciones
- Formación inmersiva para nuevo personal o en el manejo de nuevos equipos



Retorno de la inversión

Beneficios Cuantificables

- Reducción de tiempos de mantenimiento
- Disminución de errores operativos
- Optimización de los recursos

Beneficios Cualitativos

- Mayor satisfacción del personal
- Mejora en la toma de decisiones
- Transferencia de conocimiento más efectiva



Implementación y consideraciones técnicas

REQUISITOS DE SOFTWARE

Existen tres arquitecturas principales para el desarrollo de aplicaciones de realidad virtual, que van asociadas a una mayor o menor complejidad en su implementación, y que se relacionan a continuación.

1. MOTOR GRÁFICO + LIBRERÍA DE R.A.

Como se puede intuir, para utilizar esta arquitectura necesitamos principalmente dos softwares diferentes: por un lado necesitamos un motor gráfico, que permite la visualización de contenido 3D en los dispositivos y la programación de las interacciones necesarias en la aplicación. Por otro, se precisa de una librería de realidad aumentada que se integra con el motor gráfico anterior y permite, a través de diferentes sensores que dependerán del dispositivo de visualización a utilizar, calcular la posición dónde se tiene que mostrar el contenido aumentado, y mostrar el mismo.

Esta opción es la más versátil de las tres ya que tenemos la libertad de elegir tanto el motor como la librería de realidad aumentada que mejor se adapten a nuestro proyecto.

La desventaja de esta arquitectura con respecto a otras es que se hace preciso contar con un programador experimentado que dote de interactividad a los objetos virtuales.

Con respecto a los motores gráficos los dos más utilizados en la eurorregión son **Unreal Engine** y **Unity 3D**. Ambos permiten generar aplicaciones de R.A. si se combinan con una librería R.A. compatible.

Tanto Unity como Unreal distribuyen una versión gratuita pero muy capaz del software a autónomos y pequeñas organizaciones, hasta que éstos alcancen un determinado umbral de ingresos con los desarrollos de estas aplicaciones virtuales.

La librería de realidad aumentada más extendida es **PTC Vuforia**, la cual puede transformar los datos CAD e IoT existentes en experiencias detalladas que brindan información crítica a los trabajadores de primera línea cuando y donde la necesitan.

Vuforia utiliza internamente las plataformas ARCore (Google) y ARKit (Apple), lo que permite a los desarrolladores utilizar ambas librerías, a bajo nivel, para dotar de más potencia a sus aplicaciones de realidad aumentada.

Presenta una serie de módulos:

- Vuforia Expert Capture, para captura de contenidos
- Vuforia Studio, herramienta de creación de contenidos
- Vuforia Chalk, para comunicación y colaboración
- Vuforia Engine, que permite que los clientes vena y personalicen los productos.

VENTAJAS

- Potencia gráfica
- Motores gráficos con mucha comunidad
- Versiones gratuitas potentes
- Exportación a múltiples plataformas (web, gafas R.A., dispositivos móviles)
- Las librerías de realidad aumentada alcanzan prácticamente todas las posibilidades (reconocimiento de marcadores, posición espacial, reconocimiento de entornos, posicionamiento por GPS, etc.)
- Capacidades de desarrollo ilimitadas.

INCONVENIENTES

- Necesidad de personal técnico especializado (programadores)
- Necesidad de hardware de desarrollo potente
- Mayor tiempo/complejidad de desarrollo
- Desarrollos con partes comunes para cada plataforma (iOS, Android, Hololens, MagicLeap) pero hay partes específicas para cada una de ellas.

2. TECNOLOGÍAS “TODO-EN-UNO”

Existen librerías que engloban los dos aspectos del punto anterior (motor gráfico + librería de realidad aumentada) en una sola herramienta de desarrollo, están “encapsuladas” y se utilizan de forma conjunta sin posibilidad de elegir cada parte por separado.

Este concepto elimina la posibilidad de elegir utilizar un motor gráfico a libre elección con una librería que se adapte a las necesidades particulares, pero ofrece una mejor integración y, sobre todo, una mayor facilidad de desarrollo.

Generalmente estas tecnologías “todo-en-uno” están basadas en estándares web, como ocurre por ejemplo en el caso de WebAR o AR.js.

WebAR es la abreviatura que se utiliza para referirse a Web-based Augmented Reality, la realidad aumentada para web. Básicamente, es una tecnología que permite acceder a experiencias de R.A. directamente desde el navegador, sin necesidad de descargar ninguna aplicación a mayores.

Hasta hace poco, la mayoría de las experiencias de realidad aumentada se tenían que disfrutar a través aplicaciones móviles (app-based AR), lo que significaba que el usuario tenía que acceder a un mercado de apps como Google Play o App Store y descargarse la app específica para ese contenido, con la pérdida de tiempo y el consumo de memoria móvil que eso suponía.

Sin embargo, en los últimos años, el gran desarrollo de los navegadores para móviles y la capacidad de las cámaras que llevan incorporadas, han permitido crear con facilidad páginas con contenidos de R.A. a las que se accede desde una dirección URL, bien desde una url en el navegador web o mediante un código QR.

De este modo se aprovecha la libertad y flexibilidad de la tecnología web para crear experiencias basadas en reconocimiento de imágenes, reconocimiento de superficies y experiencias geo-localizadas, y lo hace utilizando la cámara del dispositivo móvil a través de un navegador como puede ser Google Chrome, Firefox, Safari, Microsoft Edge, etc.

Con webAR se puede abrir una dirección web, como la de cualquiera de las muchas páginas que se visitan a diario, y en un momento dado, a través de la cámara del teléfono, se puede mostrar contenido aumentado sobre una revista, una mesa, un equipo industrial, etc.

¿Qué tecnologías se usan para ejecutar webAR?

WebAR usa una combinación de tecnologías para proporcionar experiencias de AR inmersivas e interactivas en la web. Estas tecnologías incluyen:

- ARCore y ARKit: estos son kits de desarrollo de software creados respectivamente por Google y Apple, que permiten experiencias de AR en sus respectivos dispositivos. Usan varias tecnologías, como visión por computadora y aprendizaje automático, para reconocer y rastrear objetos físicos en el entorno físico del usuario, permitiendo que el contenido virtual se superponga al mundo real de una manera fluida e inmersiva.
- WebGL: este es un estándar web que permite renderizar gráficos en 3D en un navegador web. WebAR utiliza WebGL para renderizar el contenido virtual que se superpone al mundo real, permitiendo que se muestre de una manera realista e interactiva.
- HTML5: este es un estándar web que define la estructura y contenido de una página web. WebAR utiliza HTML5 para crear y mostrar la interfaz de usuario para la experiencia de AR, permitiendo a los usuarios interactuar con el contenido virtual de manera natural e intuitiva.
- JavaScript: este es un lenguaje de programación ampliamente utilizado en la web. WebAR utiliza JavaScript para unir todas las tecnologías mencionadas anteriormente, permitiendo que la experiencia de AR se ejecute y controle dentro del navegador web.

AR.js es una biblioteca de JavaScript popular para la visualización de WebAR. Juntas, estas tecnologías permiten que WebAR proporcione experiencias de AR inmersivas e interactivas en la web, sin requerir una aplicación dedicada.

AR.js es una biblioteca gratuita y de código abierto de JavaScript para crear experiencias de realidad aumentada en la web. Está construida sobre los estándares web WebGL y WebRTC, y utiliza los marcadores de ARToolKit y AR.js para habilitar ellas experiencias de R.A.

AR.js está diseñado para ser fácil de usar y ligero, lo que lo convierte en una opción popular entre los desarrolladores que desean crear experiencias de AR en la web. Es compatible con una amplia gama de navegadores web y dispositivos de visualización, incluyendo teléfonos inteligentes, tabletas y computadoras de escritorio.

Los marcadores de AR pueden ser cualquier imagen o patrón, aunque para AR.js, se recomienda usar códigos QR en blanco y negro.

VENTAJAS

- Totalmente gratuitas, sin versión limitada
- Sencillez de desarrollo (generalmente Javascript)
- Utilización desde múltiples dispositivos sin necesidad de exportación (Android e iOS)
- Utilización a través de navegador web sin necesidad de instalar ningún complemento
- No es necesario hardware demasiado potente

INCONVENIENTES

- Necesidad de personal técnico especializado (programadores)
- Menor potencia gráfica que aplicaciones dedicadas
- Necesidad de un hospedaje para alojar la aplicación
- Conexión a internet estable

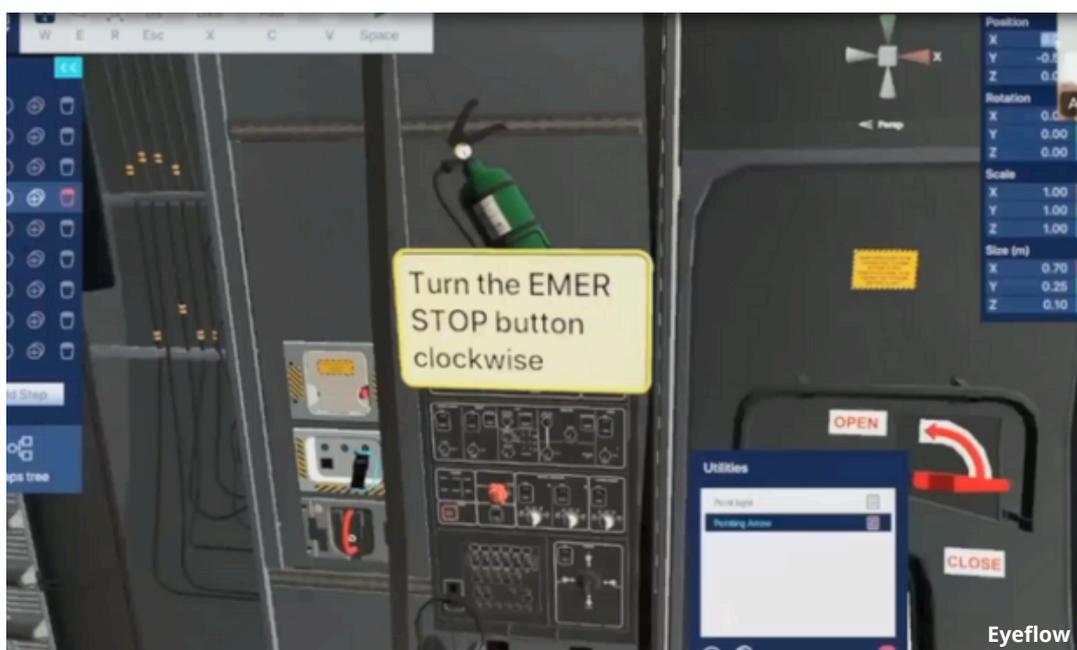
3. PLATAFORMAS "NO-CODE"

También se puede encontrar una tercera vía, que la constituyen empresas que han creado herramientas de generación de aplicaciones de R.A. y R.V. de forma sencilla, como las españolas Viroo o Eyeflow, que también hacen que incluso personal no experimentado pueda generar aplicaciones de RV/RA con unas características avanzadas.

Estas herramientas disponen de software que permite generar aplicaciones de R.A. mediante la técnica del "arrastrar y soltar".

Eyeflow es una plataforma de generación de contenidos de realidad virtual y aumentada que destaca por su versatilidad y facilidad de uso, ya que no requiere conocimientos de programación. El sistema ofrece la flexibilidad de funcionar tanto en modo online como offline, e incluye integración con servicio remoto de soporte para asistencia técnica.

Una de sus características más destacadas es la capacidad de integrar contenidos en plataformas de gestión de aprendizaje (LMS), lo que la hace ideal para entornos educativos y de formación. La plataforma es altamente accesible, permitiendo la conexión desde diversos dispositivos como smartphones, computadoras personales y una amplia gama de dispositivos de realidad virtual y aumentada, con especial énfasis en el uso de Microsoft HoloLens. Además, cuenta con capacidades de conexión IoT (Internet de las Cosas), lo que amplía significativamente sus posibilidades de aplicación y funcionalidad.



ViROO es una plataforma de desarrollo basada en Unity que destaca por su enfoque "sin código", permitiendo crear aplicaciones sin necesidad de programación, pero que también ofrece la flexibilidad de incorporar código avanzado personalizado por usuarios experimentados. La plataforma sobresale por su capacidad de crear entornos multiusuario y su compatibilidad con una amplia gama de equipos. Todos los contenidos se gestionan directamente dentro del ecosistema ViROO, que incluye Virtual Desk, un portal de formación dedicado. Una característica particularmente valiosa es su capacidad para importar y trabajar con modelos 3D provenientes de software CAD, lo que la hace especialmente útil para aplicaciones industriales.



VENTAJAS

- Desarrollo de aplicaciones accesible a cualquiera (no sólo programadores)
- Posibilidad de usarlo en diferentes dispositivos sin exportación
- Herramientas estables y probadas

INCONVENIENTES

- Coste. Son herramientas de pago sin versión gratuitas
- Optimización. Al incluir una capa de abstracción a mayores no están tan optimizadas como las soluciones anteriores.

Dynamic 365 Guides y Remote Assist. Una tercera vía consiste en el uso del ecosistema Microsoft 365. Estos dos softwares mencionados permiten generar experiencias de realidad aumentada y actividades de mantenimiento remoto respectivamente, utilizando como dispositivo de visualización las gafas Microsoft HoloLens.

También este es un conjunto que permite desarrollar ambos tipos de tareas a personal no experto, sin conocimientos de programación e incluso sin conocimientos de 3D, con la consiguiente flexibilidad que eso aporta a la empresas usuaria.

Dynamic 365 Guides permite crear una guía compuesta de instrucciones paso a paso, imágenes, vídeos y hologramas 3D. Después de crear la guía ésta se pasa a las gafas HoloLens para conectar las tarjetas de instrucciones (virtuales) y los hologramas al espacio de trabajo físico, moviéndolas a la ubicación correcta. Se puede usar la biblioteca predeterminada de hologramas 3D y también se pueden importar objetos 3D propios.



Por su parte Dynamics 365 Remote Assist, por medio de dispositivos HoloLens 2, Android o iOS, permite aportar a los trabajadores informaciones de expertos y colaboradores remotos mediante videollamadas de Teams, compartiendo la visualización de los equipos de realidad aumentada mencionados.

Lo negativo de estos sistemas es que Microsoft ha comunicado recientemente que dejará de dar soporte a estas herramientas a partir de 2027.

REQUISITOS DE HARDWARE

De nuevo podemos encontrar diferentes niveles de hardware a la hora de ejecutar aplicaciones de realidad aumentada, desde los más sencillos hasta los más complejos.

DISPOSITIVOS MÓVILES

Los dispositivos móviles se han convertido en una plataforma fundamental para la realidad aumentada, gracias a su ubicuidad y a la integración de cámaras, sensores y procesadores cada vez más potentes. La mayoría de smartphones y tablets modernos son capaces de ejecutar aplicaciones de RA a través de frameworks como ARCore (Android) y ARKit (iOS), que permiten el reconocimiento del entorno, el seguimiento de movimiento y la superposición precisa de elementos virtuales sobre el mundo real.

Entre las principales ventajas del uso de dispositivos móviles para R.A. destaca su accesibilidad, ya que la mayoría de personas ya disponen de un smartphone, eliminando la necesidad de adquirir hardware específico y, por otro lado, la curva de aprendizaje es menor pues los usuarios están familiarizados con la manipulación de estos dispositivos. El coste de desarrollo y distribución también es más bajo, aprovechando las infraestructuras existentes de las tiendas de aplicaciones. La portabilidad y la autonomía de batería son otros puntos fuertes, junto con la constante mejora en capacidades de procesamiento y calidad de los sensores incorporados.

Sin embargo, existen limitaciones significativas. La principal es que el usuario debe sostener el dispositivo constantemente, lo que puede resultar incómodo en sesiones largas o cuando se requieren las dos manos para otras tareas. El campo de visión está limitado por el tamaño de la pantalla y el ángulo de la cámara, ofreciendo una experiencia menos inmersiva que las gafas de R.A.. Los sensores, aunque capaces, no son tan precisos como los de dispositivos especializados, lo que puede afectar al tracking y la estabilidad de los elementos virtuales. También hay que considerar las limitaciones de procesamiento, e incluso térmicas ya que las aplicaciones de R.A. son exigentes y pueden causar sobrecalentamiento del dispositivo en sesiones prolongadas.

En el contexto industrial y profesional, los dispositivos móviles pueden ser una excelente opción para aplicaciones puntuales o de corta duración, como visualización de modelos 3D, navegación aumentada o inspecciones breves. Sin embargo, para tareas que requieren uso prolongado o manos libres, como mantenimiento industrial o formación técnica, las gafas de R.A. siguen siendo la opción más adecuada a pesar de su mayor coste.

DISPOSITIVOS DEDICADOS

Microsoft HoloLens y Magic Leap representan dos importantes hitos en el desarrollo del hardware de realidad aumentada, con especial énfasis en HoloLens como referente en el sector.

Las HoloLens, desarrolladas por Microsoft, se establecieron como una tecnología pionera que revolucionó las aplicaciones de realidad aumentada y asistencia remota, destacando por su facilidad de implementación y su compatibilidad con el motor Unity para el desarrollo de aplicaciones. A pesar de su éxito inicial y su posición como una opción ideal para adentrarse en el mundo de la realidad aumentada y el mantenimiento industrial, especialmente por su relación calidad-precio, Microsoft ha anunciado que sólo proporcionará actualizaciones de software para problemas críticos de seguridad y errores hasta el año 2027, lo que marca un punto de inflexión en su ciclo de vida.

Las **HoloLens** sobresalieron particularmente por sus aplicaciones Dynamic Guides y Remote Assist, herramientas que facilitan significativamente la implementación de soluciones de realidad aumentada en entornos profesionales. Esta tecnología, aunque aparentemente no continuará su desarrollo para el gran público, dejó un importante legado en la industria y estableció estándares para futuros desarrollos en el campo de la realidad aumentada.

Lo que si continúa Microsoft es el trabajo con el ejército de los EEUU para suministrarle un casco de R.A. realmente operativo, el Army's Integrated Visual Augmentation System (IVAS), que dote de mayor valor añadido a sus soldados, basado en el HoloLens.



Las **Magic Leap** son unas gafas de realidad aumentada que han tenido dos versiones principales: Magic Leap One (lanzada en 2018), con muchas carencias, y las Magic Leap 2 (lanzada en 2022) que sí están a la altura de lo que se espera de un equipo de estas características.

Las Magic Leap 2 representan una mejora significativa sobre su predecesora, siendo más ligera y compactas, con un campo de visión más amplio (70 grados en diagonal) y una tecnología única de atenuación dinámica que puede oscurecer selectivamente partes del visor para mejorar la visibilidad de los hologramas en entornos brillantes. Esta característica las hace especialmente útiles en entornos industriales y sanitarios.

A diferencia de HoloLens, que se centró principalmente en aplicaciones empresariales desde el principio, Magic Leap comenzó con ambiciones en el mercado de consumo, aunque posteriormente pivotó hacia el sector empresarial. Finalmente la compañía se ha enfocado en sectores como la sanidad, la manufactura y la defensa (donde también se trabaja con las HoloLens).

Las Magic Leap 2 cuentan con controladoras hápticas que permiten una interacción más precisa con los elementos virtuales. La plataforma admite desarrollo con Unity 3D y Unreal Engine, lo que facilita la creación de aplicaciones, aunque en este caso si son necesarios conocimientos de programación.

Sin embargo, al igual que en el caso de las HoloLens, el precio ha sido una barrera para su adopción masiva, manteniéndose principalmente en el ámbito empresarial y de desarrollo.



METAVERSO

El Metaverso representa la siguiente evolución de Internet, conceptualizado como un espacio virtual tridimensional persistente y compartido que combina múltiples elementos de tecnología, incluyendo realidad virtual y realidad aumentada, donde los usuarios pueden interactuar con un entorno generado por ordenador y con otros usuarios, todo ello en tiempo real.

La definición actual del Metaverso engloba varios elementos clave: persistencia (el mundo continúa existiendo y evolucionando incluso cuando los usuarios no están conectados), sincronía (permite interacciones en tiempo real), interoperabilidad (capacidad de trasladar activos digitales entre diferentes plataformas), y economía digital (permite a los usuarios crear, poseer, invertir y comerciar con activos digitales).

Entre las **ventajas** más significativas del Metaverso encontramos:

- Eliminación de barreras geográficas para la interacción social y profesional
- Reducción de la necesidad de viajes físicos para reuniones y eventos
- Nuevas oportunidades económicas a través de la economía digital
- Experiencias más inmersivas y envolventes que las plataformas digitales actuales
- Mayor potencial para la creatividad y la autoexpresión

Sin embargo, también presenta **desafíos** importantes:

- Preocupaciones sobre privacidad y seguridad de datos
- Necesidad de infraestructura tecnológica robusta
- Riesgos de adicción y aislamiento social
- Cuestiones de accesibilidad y brecha digital
- Desafíos de interoperabilidad entre diferentes plataformas

La evolución del Metaverso está apenas comenzando, y su desarrollo continuo promete transformar fundamentalmente la forma en que interactuamos, trabajamos, aprendemos y nos entretenemos en el mundo digital. Sin embargo, su éxito dependerá en gran medida de la capacidad de la industria para abordar los desafíos técnicos, sociales y éticos que surjan en el camino.

Los gemelos digitales se benefician de las mejoras en la interoperabilidad de datos impulsadas por marcos de datos abiertos como OpenUSD, gráficos avanzados, IA generativa y computación acelerada, lo que lleva al surgimiento de una nueva clase de gemelos digitales basados en la física y habilitados por IA.

En la industria estos gemelos satisfacen la necesidad de visibilidad de datos en tiempo real, mantenimiento predictivo, optimización de procesos, operaciones remotas, mejora en la calidad, gestión de riesgos y reducción de costos en las operaciones de fabricación.

En el ámbito puramente industrial el metaverso que se encuentra más enfocado y evolucionado es el de la empresa tecnológica NVIDIA, llamado Omniverse.

Omniverse

NVIDIA Omniverse™ es una plataforma de API, SDK y servicios que permiten a los desarrolladores integrar Open USD, las tecnologías de renderizado NVIDIA RTX™ y la IA física generativa en las herramientas de software existentes y la simulación de los flujos de trabajo de para casos de uso industriales y robóticos.

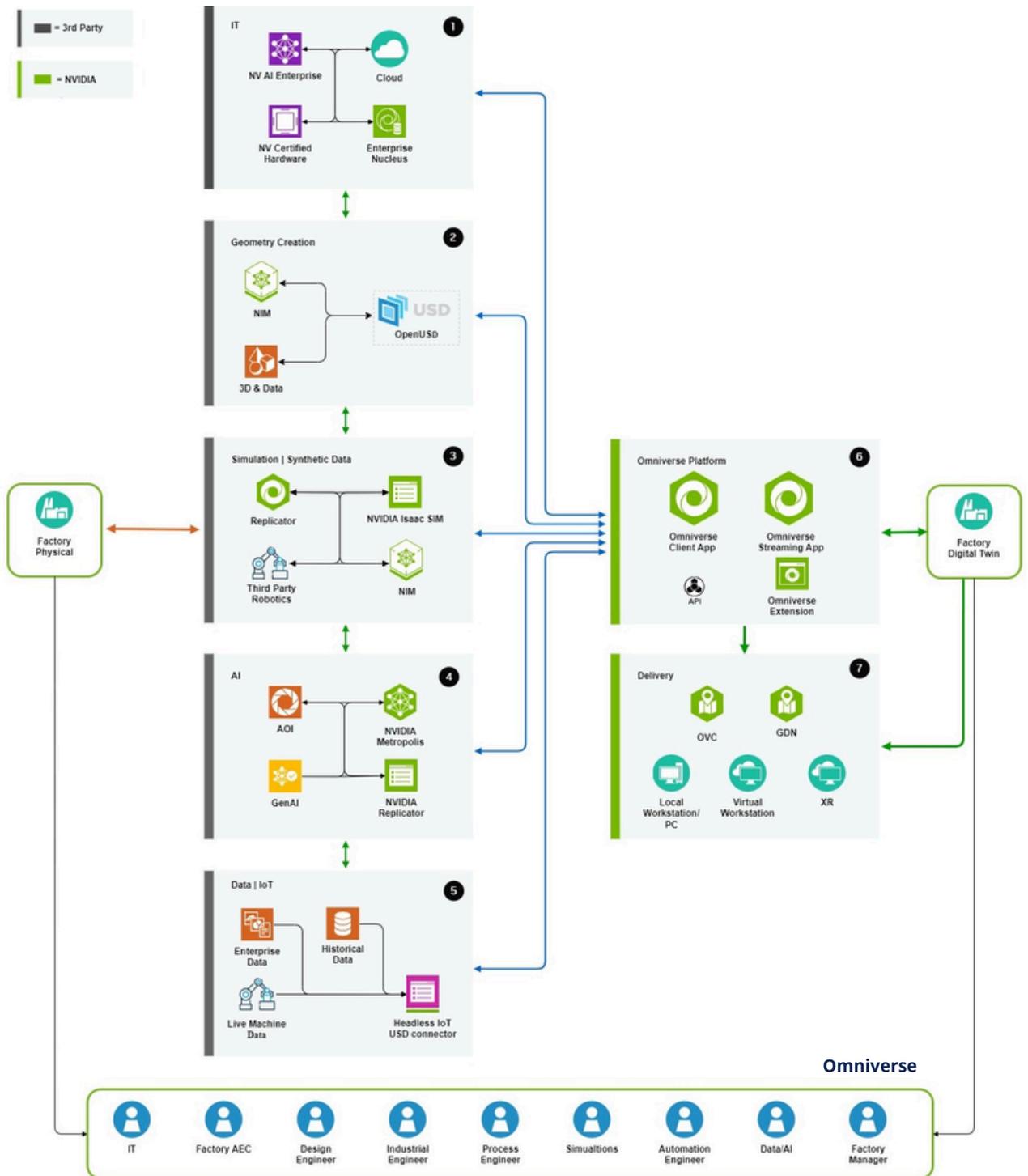
El formato OpenUSD (Universal Scene Description) es independiente del sistema de archivos y tiene una API para crear, editar, consultar, renderizar, colaborar y simular mundos virtuales. Diferentes usuarios pueden modificar la representación de un equipo o de una fábrica, en diferentes capas, al mismo tiempo y sus ediciones no dañarán el trabajo de los demás. Se conecta directamente con toda una serie de sistemas de CAD del mercado.

RTX™ es la plataforma más avanzada para el trazado de rayos y las tecnologías de renderizado neuronal en tiempo real, aumentando los FPS, reduciendo la latencia y mejorando la calidad de imagen.

La IA física permite que las máquinas autónomas, como por ejemplo robots o automóviles autónomos, perciban, comprendan y realicen acciones complejas en el mundo real (físico). También se le suele llamar “IA física generativa” debido a su capacidad de generar conocimientos y acciones a ejecutar. Esta IA amplía la IA generativa actual con la comprensión de las relaciones espaciales y el comportamiento físico del mundo 3D en el que vivimos, lo que se logra proporcionando datos adicionales que contienen información sobre las relaciones espaciales y las reglas físicas del mundo real durante el proceso de entrenamiento de la IA.

En cuanto al IoT, los sensores y dispositivos de Internet de las cosas (IoT) desempeñan un papel crucial a la hora de proporcionar datos en tiempo real que mantienen a los gemelos digitales precisos y actualizados, lo que permite interacciones dinámicas bidireccionales entre los ámbitos físico y digital. Omniverse permite integrar extensiones personalizadas como, por ejemplo: conectar fuentes de datos de IoT (CSV, agente de mensajes, etc.), incorporar datos de IoT al modelo USD, visualizar datos de IoT mediante una extensión OmniUI o modificar el rendimiento de la geometría mediante datos de IoT.

En la imagen siguiente podemos ver la arquitectura de referencia del gemelo digital de fábrica en Omniverse:



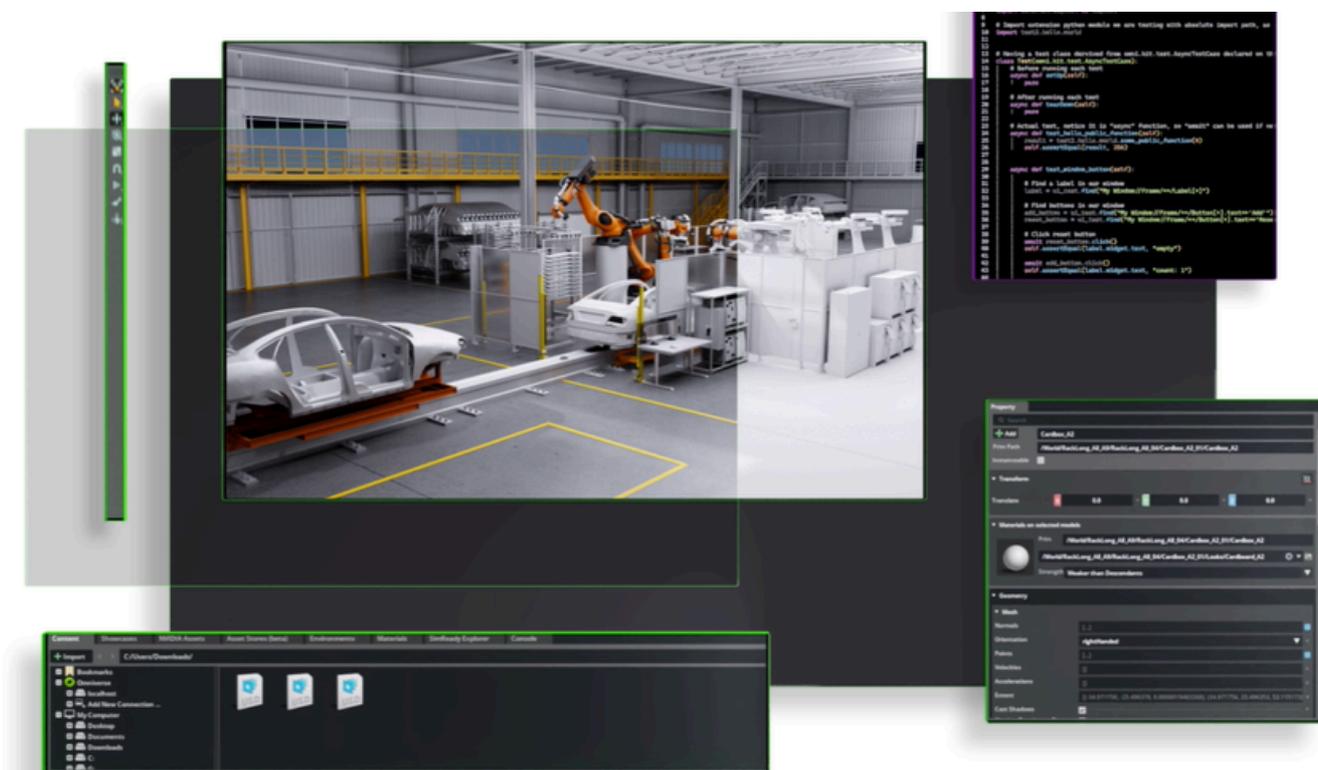
En cuanto al licenciamiento de esta herramienta, nos encontramos con varias posibilidades:

- NVIDIA Omniverse Enterprise, anual o trianual
- Educational Program, con un menor coste y licencias anuales o trianuales
- Inception Program con licencia anual
- Free limitada en uso

La versión Enterprise contiene diversos módulos:

- Omniverse Kit SDK
- Omniverse Base Editor
- Omniverse Extensions
- Omniverse RTX Renderer
- NVIDIA PhysX
- App Streaming API
- Omniverse Nucleus.

Salvo que la empresa usuaria posea una infraestructura propia muy potente, para trabajar con Omniverse deberá hacerlo a través de una nube como AWS o Azure, para las que existen los productos NVIDIA necesarios.



Unity

Incluimos este apartado referente a Unity, porque en los últimos tiempos se ha enfocado con fuerza en el mundo industrial y ha creado el **Unity Industry**, que se define como un conjunto de productos y servicios para que los desarrolladores creen experiencias 3D personalizadas y en tiempo real para AR, VR, dispositivos móviles, computadoras de escritorio y la web, aplicado en cualquier industria.

Incluye:

- La última versión de Unity
- Pixyz para importación optimizada de geometrías
- Colaboración en la Nube

Los productos **Pixyz** permiten optimizar los flujos de trabajo de datos 3D y prepararlos para cualquier uso de visualización:

- Optimiza modelos 3D complejos para crear activos listos para aplicaciones en tiempo real.
- Automatiza la preparación y el procesamiento de datos 3D, CAD y de nubes de puntos.
- Habilita la revisión instantánea y la colaboración fluida.

PiXYZ es accesible independientemente o con Unity Industry, y cuenta con la siguiente estructura de soluciones:



Pixyz Studio

Características:

- Importa/convierte datos CAD, de malla y de nube de puntos
- Optimiza/exporta datos
- Automatiza tareas a través de la API de Python



Pixyz Plugin

Incluido en [Unity Industry](#)

Complemento de Unity para importación de datos

Características:

- Importa datos CAD/3D/BIM en Unity
- Conserva la jerarquía y los metadatos
- Automatiza scripts con Rule Engine



Pixyz SDK

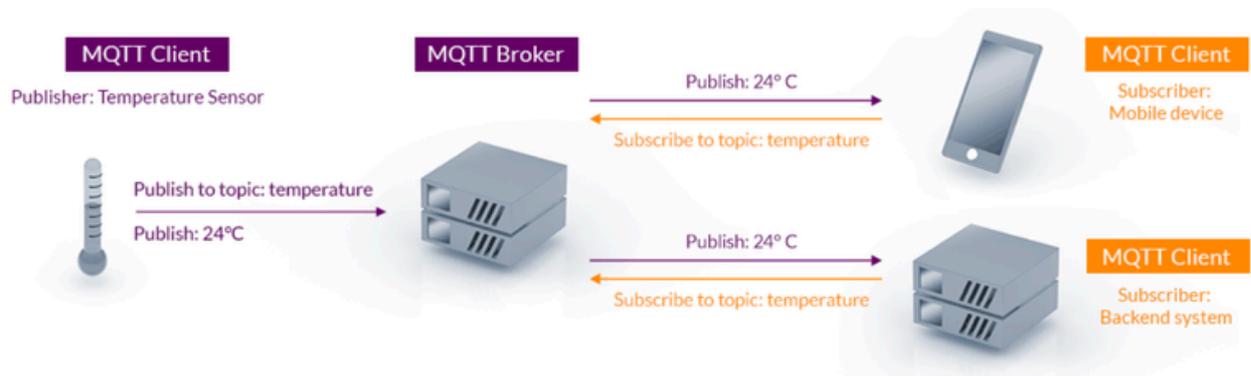
Características:

- Convierte y optimiza datos en lotes
- Se ejecuta en instalaciones locales o en la nube privada
- Libera el poder de Pixyz en cualquier lugar y a escala

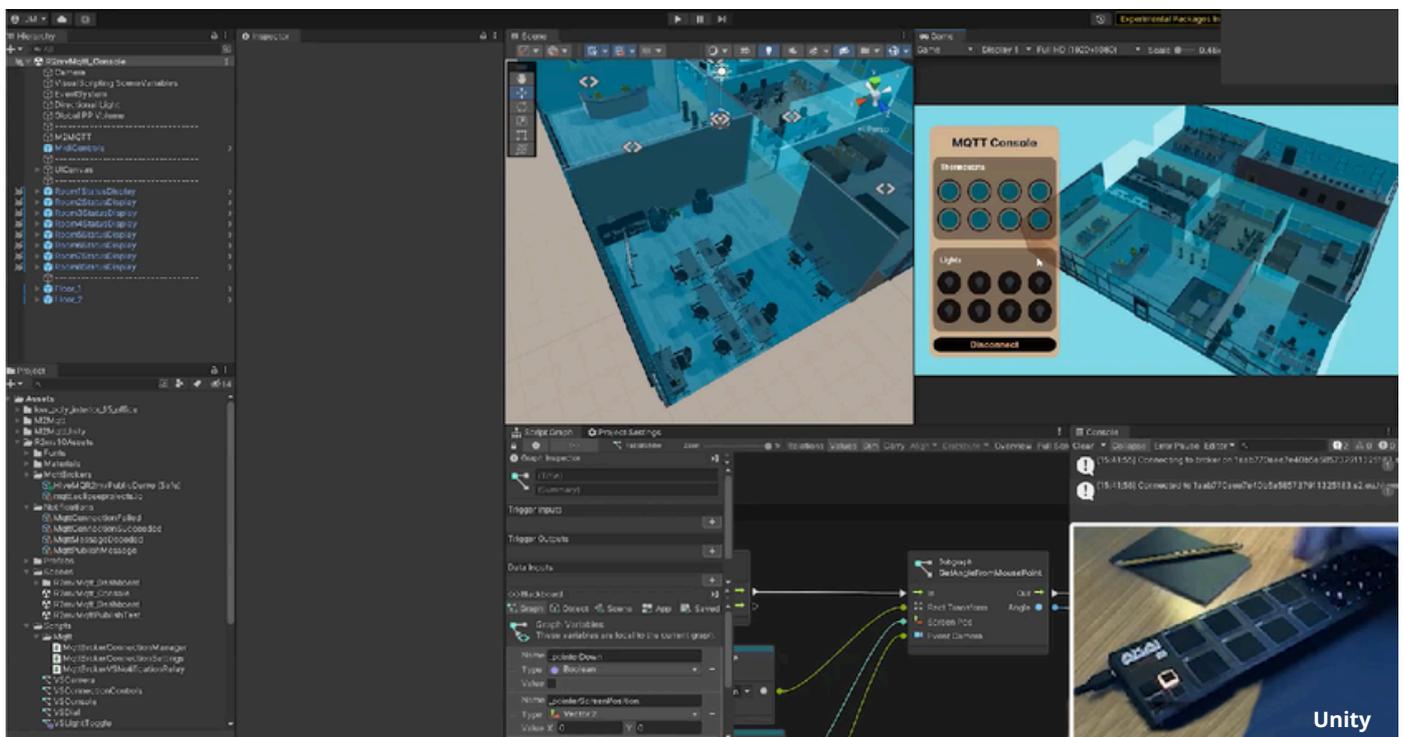
Por su parte, las herramientas disponibles en la nube de Unity (Unity Cloud) permiten a los equipos colaborar, compartir y administrar archivos en un proyecto o entre proyectos, de una forma sencilla e intuitiva.

Pero además Unity presenta actualmente capacidades avanzadas de intercomunicación bidireccional entre simulación virtual y equipos industriales por medio de **IoT**, lo que permite acercar a más empresas al mundo de los gemelos digitales y el metaverso.

MQTT Publish / Subscribe Architecture

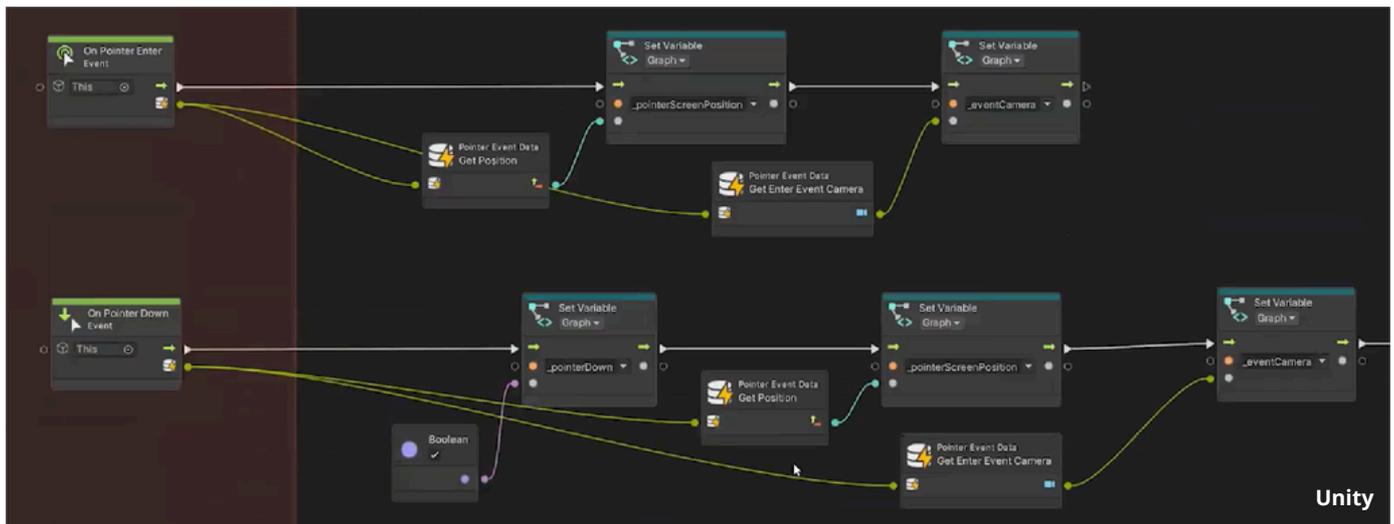


En la imagen siguiente se puede ver el interface de Unity incluyendo una simulación y actuadores virtuales y físicos, que están interconectados para poder monitorizar y actuar sobre la iluminación y la temperatura de unas oficinas.



La conexión se hace por medio de MQTT, un popular protocolo para IoT, que se descarga gratuitamente desde <https://github.com/gpvigano/M2MqttUnity>.

Dentro de los brokers (sistemas intermedios de comunicación) podemos encontrar el Mosquitto, muy habitual entre la comunidad maker, y otros más profesionales como HMQTT, en Python, o EMQTT, todos ellos Open Source. Pero existen muchos más, y habrá que seleccionarse uno u otro según el lenguaje a utilizar y la tipología del proyecto a desarrollar.



Custom event. Fuente: Unity

Ahondando en el enfoque de Unity hacia el gemelo digital, también existe un asset de Unity para comunicarse con las placas Arduino, el Uduino.

Se descarga de la Unity Store y se instala en apenas unos segundos. En Unity tiene el aspecto de la parte derecha de la imagen anterior y el hardware al que se conecta presenta un interface dentro del IDE de Arduino, que se selecciona dentro de su menú Ejemplos: Uduino.

Esta capacidad facilita que las empresas puedan acceder a estas tecnologías de forma barata y accesible, pues existe una gran cantidad de técnicos formados en Unity y en Arduino.

Interreg



Cofinanciado por
la Unión Europea
Cofinanciado pela
União Europeia

España – Portugal

TwinNavAux



AXENCIA
GALEGA DE
INNOVACIÓN



Asociación Cluster del Naval
Gallego (**ACLUNAGA**)



Universidad de da
Coruña (**UDC**)



Industrias Ferri SA



Ibercisa Deck Machinery SA



Electrorayma SL



Centro de Apoio Tecnológico à
Indústria Metalomecânica (**CATIM**)



Universidade Portucalense
Infante Dom Henrique