

## Diseño de una arquitectura para sistemas y aplicaciones en Industria 4.0 basada en computación en la nube y análisis de datos

Carlos E. Belman-López\*, José A. Jiménez-García\*, José A. Vázquez-Lopez, Karla A. Camarillo-Gómez

*Departamento de Doctorado en Ciencias de la Ingeniería, Tecnológico Nacional de México en Celaya.*

**To cite this article:** Belman-Lopez, C.E., Jiménez-García, J.A., Vázquez-Lopez, J.A., Camarillo-Gómez, K.A. 2023. Design of an architecture for systems and applications in Industry 4.0 based on cloud computing and data analysis. *Revista Iberoamericana de Automática e Informática Industrial* 20, 124-136. <https://doi.org/10.4995/riai.2022.17791>

### Resumen

El término Industria 4.0 se ha convertido en prioridad y objeto de estudio para empresas y centros de investigación, pero aún se encuentra dentro de sus primeras etapas de implementación. Además, las compañías enfrentan dificultades al desarrollar soluciones para Industria 4.0, sin estar seguras de cómo afrontar sus requerimientos básicos. El diseño de una arquitectura de referencia aborda explícitamente este problema, apoya a los profesionales en la implementación de soluciones siendo la base del desarrollo y proporciona un soporte ante los desafíos que la Industria 4.0 representa. Por lo tanto, la contribución de este documento se centra en diseñar una arquitectura de referencia para sistemas y aplicaciones en Industria 4.0 basada en computación en la nube y análisis de datos, mostrando su viabilidad a través de la implementación en un caso de uso: Agricultura 4.0. Mediante esta arquitectura, ingenieros e investigadores podrán enfrentar los desafíos actuales de la producción inteligente, así como investigar, desarrollar e implementar soluciones (aplicaciones y sistemas) guiadas, estandarizadas y a costos accesibles, que cumplan los requerimientos que gobiernan Industria 4.0.

*Palabras clave:* Industria 4.0, arquitectura de sistemas, computación en la nube, análisis de datos, desarrollo de aplicaciones.

### Design of an architecture for systems and applications in Industry 4.0 based on cloud computing and data analysis

### Abstract

Industry 4.0 has become a priority and object of study for companies and research centers, although it is still in its early stages of implementation. In addition, companies face difficulties in developing solutions for Industry 4.0, without being sure how to address its essential requirements. A reference architecture explicitly addresses this problem, supports professionals in implementing solutions, being the basis of development, and providing support to face the challenges that Industry 4.0 represents. Therefore, the contribution of this document focuses on designing a reference architecture for systems and applications in Industry 4.0 based on cloud computing and data analysis, demonstrating its applicability through the implementation of a use case. Through this architecture, engineers and researchers will face the current challenges of smart production, in addition to researching, developing, and implementing guided, standardized solutions (applications and systems) at affordable costs that meet the requirements that govern Industry 4.0.

*Keywords:* Industry 4.0, system architecture, cloud computing, data analysis, applications development.

### 1. Introducción

La Industria 4.0 combina métodos avanzados de producción y tecnologías innovadoras para facilitar la colaboración de organizaciones y personas con activos físicos y virtuales (Liu, y otros, 2022). Además, las empresas deben identificar las tecnologías que mejor satisfacen sus necesidades y las de sus clientes para invertir en nuevos

desarrollos tecnológicos que les permita desarrollar servicios de valor agregado y mejorar la eficiencia y productividad de los recursos a costos accesibles (Karatas et al., 2022). Entre los desafíos existentes para Industria 4.0, destacan la creación de arquitecturas o modelos de referencia, desarrollo de plataformas que integren múltiples datos de producción, la colaboración hombre-máquina, la implementación de casos de uso exitosos, entre otros (Wankhede & Vinodh, 2021).

\*Autor para correspondencia: carlosbelman@gmail.com; josealfredo.jimenez@itcelaya.edu.mx  
Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

Una arquitectura posee un enfoque exterior centrándose en el sistema y su entorno, mientras que el diseño se enfoca hacia el interior una vez que se han establecido los límites del sistema (ISO/IEC/IEEE 42010, 2007). El estándar IEEE 42010 define “arquitectura del sistema” como los conceptos o propiedades fundamentales de un sistema en su entorno, materializados en sus elementos (estructuras y subsistemas), relaciones (internas o externas) y principios de su diseño y evolución. El pensamiento arquitectónico es fundamental en Industria 4.0 pero la mayoría de los ingenieros de producción son expertos en áreas como la mecánica, mecatrónica, entre otras, por lo cual, la noción del pensamiento arquitectónico ha encontrado resistencia (Wang et al., 2017).

En este contexto, es necesaria una arquitectura de referencia que cumpla con los requisitos que gobiernan Industria 4.0, proporcione una descripción de sus elementos, y facilite la implementación y el despliegue de los nuevos sistemas y aplicaciones empresariales (Zhong et al., 2017; López Martínez et al., 2021). Hermann et al. (2015) identificó seis drivers arquitectónicos o requerimientos arquitectónicamente significantes (ASR) para que las empresas logren los beneficios de la Industria 4.0. Estos son la interoperabilidad, virtualización, descentralización, respuesta en tiempo real, orientación al servicio y modularidad. Posteriormente, Ghobakhloo (2018) presentó doce requerimientos para Industria 4.0. Aunque en una investigación más reciente, Belman-Lopez et al. (2020) detalla de forma amplia diecisiete requerimientos para las empresas en Industria 4.0. Estos requerimientos utilizados durante esta investigación son modularidad, integración, colaboración, flexibilidad y escalabilidad, virtualización, arquitectura distribuida y descentralizada, holística, personalizable, ubicua, robustez, información en tiempo real, autonomía e inteligencia, decisiones optimizadas por datos, seguridad, equilibra la vida laboral y mejora la eficiencia y productividad.

No obstante, la Industria 4.0 aún se encuentra dentro de las primeras etapas de implementación tanto en la industria como en la investigación científica, donde no está claro cómo incorporar muchos conceptos tecnológicos (Sony et al., 2021; Liu et al., 2017; Hermann et al., 2015). Además, las arquitecturas existentes aún tienen un largo camino por recorrer, pero la comunidad de la Industria 4.0 tiene razón al invertir en arquitecturas de referencia considerando el futuro de la Industria 4.0 (Nakagawa et al., 2021). Adicionalmente, muchas de las compañías aún dudan en realizar una implementación sobre Industria 4.0 porque consideran que requieren de una inversión inicial bastante elevada (adquirir servidores, configurarlos y un lugar para almacenarlos, acondicionar el espacio en cuanto a temperatura, energía, conexión a Internet, seguridad, entre otros gastos), más elevada aún si los servicios y aplicaciones que se quieren implementar necesitan de redundancia (para garantizar la suficiente capacidad de procesamiento, escalabilidad y alta disponibilidad) mediante su despliegue a través de diferentes centros distribuidos en varias zonas geográficas.

La computación en la nube ofrece una solución a estos desafíos cambiando la forma de implementar servicios, permitiendo a compañías de cualquier tamaño construir aplicaciones poderosas y escalables (Pallathadka et al., 2021). Entre las ventajas que ofrece la computación en la nube se encuentran: (1) disminuir la inversión inicial, sin gastar tiempo y dinero, comprando y administrando servidores e

infraestructura. (2) No es necesario predecir cuánta capacidad necesitará la infraestructura antes de implementar una aplicación. (3) Alcanzar un costo variable más bajo comparado con el que se obtendría por cuenta propia gracias a que los proveedores en la nube hacen uso de las economías de escala. Además, la arquitectura e infraestructura tecnológica de Industria 4.0 incluye el acoplamiento de dispositivos inteligentes como sistemas ciber-físicos (CPS) enriquecidos en sus capacidades mediante computación en la nube. Esto implica máquinas reales que operan en conjunto con una plataforma en la nube, simulando las condiciones de un conocimiento basado en algoritmos impulsados por datos, así como de otros conocimientos físicos disponibles (Tao et al., 2019). Por estos motivos, la contribución de este documento se centra en diseñar una arquitectura de referencia basada en computación en la nube y análisis de datos que cumpla los requerimientos que gobiernan Industria 4.0, demostrando su viabilidad a través de la implementación de un caso de uso en Agricultura. Mediante esta arquitectura, los ingenieros podrán implementar soluciones (aplicaciones y sistemas) guiadas, estandarizadas y a costos accesibles en el contexto de Industria 4.0.

La Sección 2 presenta trabajos relacionados a esta investigación. La Sección 3 muestra la metodología. La Sección 4 expone la arquitectura propuesta. La Sección 5 muestra la implementación del caso de uso. La Sección 6 presenta costos y discusión de los resultados. La Sección 7 presenta las conclusiones y el trabajo futuro.

## 2. Trabajos relacionados

La arquitectura es el principal portador de las cualidades del sistema, como el rendimiento, la flexibilidad, la seguridad, entre otras, las cuales no se pueden lograr sin una visión integral y unificada de la misma (Carnell, 2017). Las arquitecturas de referencia se han adoptado cada vez más para guiar a los ingenieros sobre cómo deben interactuar y estructurarse sus sistemas, abordando el uso de las tecnologías y herramientas que podrían respaldar su implementación, permitiendo desarrollar aplicaciones flexibles, reutilizables y robustas (Nakagawa et al., 2021). Los sistemas que carecen de una arquitectura formal están estrechamente acoplados, son frágiles, difíciles de cambiar, no poseen una visión o dirección clara, contienen módulos desorganizados, carecen de roles, responsabilidades y relaciones claras entre sí (Richards, 2015). Una arquitectura representa la base de la investigación y el desarrollo, permitiendo modelar la tecnología que describe el sistema objetivo antes de ser llevado a cabo (Liu et al., 2017). Además, permite estimar los recursos necesarios para la implementación, aumenta la calidad, favorece la reutilización de componentes y servicios, mejorando los tiempos de entrega y reduciendo los costos (Cervantes Maceda et al., 2016).

En la literatura existen una serie de arquitecturas que se han propuesto para Industria 4.0. Dintén et al. (2021) y López Martínez et al. (2021) presentaron arquitecturas (metamodelos) para el desarrollo en entornos locales (on promise) pero escalables y distribuidos enfocándose en la configuración y despliegue de nodos. Predix es una plataforma para el Internet Industrial desarrollada por GE, que emplea una arquitectura en microservicios y un enfoque modular para el desarrollo de aplicaciones (Azeem et al., 2021). Predix

conecta a dispositivos y proveedores, permitiendo la gestión del rendimiento y posee servicios para la optimización de operaciones (GE, 2018). Nakagawa et al. (2021) realizaron una revisión de las arquitecturas de referencia existentes para Industria 4.0, analizando su viabilidad y destacando IIRA, SITAM, IVRA, RAMI 4.0, IBM Industry 4.0 y LASFA.

RAMI 4.0 utiliza una arquitectura SOA donde los componentes del sistema proporcionan servicios en red de forma independiente a la tecnología. RAMI 4.0 define ciertas capas, como activos, integración, comunicación, información, función y negocios, considerando todo el ciclo de vida del producto (ISA, 2019). OPC UA utiliza una arquitectura SOA centrada en aspectos de adquisición de datos, modelado de comunicación (entre planta y aplicaciones) e interoperabilidad (Yang et al., 2019). Liu et al. (2019) desarrollaron una propuesta de sistema de producción ciberfísico (CPPS) basado en OPC UA y MTConnect. Aunque ya diversos autores han señalado las desventajas de arquitecturas basadas en SOA resultantes en aplicaciones complejas, pesadas y costosas, de difícil mantenimiento, requiriendo gran capacitación y entrenamiento (Niknejad, y otros, 2020).

Algunas otras propuestas han sido orientadas a dominios más específicos relacionados con Industria 4.0. Geest et al. (2021) propusieron una arquitectura de referencia para el desarrollo de almacenes inteligentes en Industria 4.0 aplicando un enfoque de diseño basado en el dominio. Sahba et al. (2021) desarrollaron una arquitectura de mantenimiento predictivo para Industria 4.0 y la cadena de radiodifusión personalizando el modelo RAMI 4.0. Aheleroff et al. (2021) desarrollaron una arquitectura de referencia para el gemelo digital en Industria 4.0, incluyendo características como servicios para mantenimiento inteligente, monitoreo en tiempo real, control remoto y predicción.

Miny et al. (2021) presentaron una vista funcional del término "Asset Administration Shell (AAS)" para la virtualización de activos e implementación del gemelo digital (DT) en aplicaciones para Industria 4.0. El AAS está basado en dos conceptos clave: la capacidad informática (red de nodos informáticos para el despliegue y operación de las aplicaciones) y los servicios de software (funcionalidad ofrecida mediante la capacidad informática) pero sin prescribir una forma determinada de despliegue e implementación. Bader et al. (2021) definieron interfaces de programación de aplicaciones (APIs) para permitir el acceso a la información que proporciona un AAS. La especificación es neutral a la tecnología y sigue un enfoque orientado a recursos muy similar a REST mediante tres principios de diseño: operaciones independientes o sin estado (stateless), recursos definidos por sustantivos y verbos para describir los métodos. Sin embargo, aún falta desarrollar un modelo con una imagen completa del entorno de la aplicación e incluya requerimientos no funcionales como la seguridad, robustez, flexibilidad y escalabilidad (Miny et al., 2021).

Además, existen otros retos que si no se consideran pueden limitar el rendimiento y efectividad de las arquitecturas existentes, tales como, el volumen de datos a ser administrados, la capacidad de procesamiento y manejo de cargas para producir respuestas en tiempo real, entre otros (Macías et al., 2019). Tampoco son claros el total de los requerimientos que las arquitecturas satisfacen, patrones de diseño que utilizan o los costos para su implementación y despliegue en ambientes productivos.

### 3. Metodología

La arquitectura define cómo resolver el problema del cliente de manera que la solución satisfaga todos los requerimientos funcionales y no funcionales (atributos de calidad y restricciones) sin definir detalles de implementación (López Martínez et al., 2021). El diseño de la arquitectura se inicia al completar la especificación de requerimientos, pero antes de la fase de implementación. El diseño define las estructuras que componen la arquitectura mediante la toma de decisiones y selecciones de patrones de diseño. La implementación permite demostrar la viabilidad y los beneficios de la arquitectura, y se desarrollan, prueban, integran y despliegan los módulos y servicios identificados.

Para realizar el diseño de la arquitectura se utilizó el método de diseño guiado por atributos (ADD). El ADD es un método sistemático diseñado por el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon. El método ADD está basado en los ASR que incluyen requerimientos funcionales, atributos de calidad y restricciones. El ADD sigue un proceso de descomposición iterativo, donde en cada etapa de descomposición se eligen patrones de diseño para satisfacer los ASR (Software Engineering Institute, 2018). Los pasos del método ADD se describen a continuación.

1. Asegurar que existe suficiente información sobre los ASR asociados al sistema.
2. Elegir un elemento del sistema para descomponer, el elemento puede ser el sistema completo (si es un desarrollo nuevo) o un elemento obtenido de una iteración anterior.
3. Identificar los ASR asociados al elemento.
4. Elegir conceptos de diseño que satisfagan los ASR.
5. Crear elementos derivados de los patrones y definir sus responsabilidades.
6. Definir interfaces y propiedades para los elementos identificados.
7. Verificar si se han satisfecho los ASR y en caso necesario refinar los elementos identificados.
8. Repetir los pasos 2-7 para elementos que requieran un refinamiento mayor hasta cubrir los ASR.

El primer paso consistió en confirmar que existe suficiente información sobre los ASR, donde se utilizaron los requerimientos identificados en Belman-Lopez et al. (2020) (*requerimientos no funcionales*). Además, el caso de uso se enfoca en agricultura y consiste en desarrollar una aplicación que detecte plagas y plantas enfermas a partir del video de la cámara de dispositivos móviles (*requerimientos funcionales*). La Tabla 1 muestra el resumen de las iteraciones. En el segundo paso, se tomó el sistema como elemento a descomponer (dado que se está creando el sistema desde cero). En el tercer paso se identificaron los ASR a tratar durante la iteración. Dado que se trata de la iteración inicial, el enfoque consiste en la estructuración general del sistema. En el cuarto paso, fueron elegidos conceptos de diseño para dar forma al elemento seleccionado. La arquitectura debe ser modular, es decir, desacoplada en subsistemas con pocas dependencias (López Martínez et al., 2021). Por lo cual, para la estructura inicial del sistema se utilizó un estilo arquitectónico por capas, ideal para alcanzar la separación de intereses, aislando (desacoplando) de forma lógica las distintas responsabilidades del sistema (Macías et al., 2019) y aspectos relacionados a la interacción con el usuario (capa de presentación al cliente), captura de datos (capa de captura y comunicación), conexión

de sistemas externos (capa de integración), desarrollo del gemelo digital clave para la Industria 4.0, manejo de la lógica de negocio (capa de servicios). Además, la aplicación prioriza un modelo de distribución de software específicamente diseñado para entrega en red (capa en la nube) que permita satisfacer aumentos en la demanda de manera flexible, ágil, descentralizada y que utilice los recursos de forma óptima y segura (ciberseguridad) (Wang et al, 2017).

Tabla 1. Resumen de las iteraciones utilizando el método ADD.

Concepto	Opciones tecnológicas / Justificación	ASR que satisface
<b>Elemento: El sistema (estructura general) – Iteración 1</b>		
Estilo arquitectónico por capas	Alta cohesión y bajo acoplamiento entre subsistemas, aislando de forma lógica distintas responsabilidades	Modularidad
<b>Elemento: clientes y entidades ciber-físicas – Iteración 2</b>		
Entorno de interacción entre las entidades y la aplicación. La aplicación brinda a las entidades soporte en la toma de decisiones, muestra los conocimientos adquiridos y diagnósticos generados. Los usuarios pueden ejecutar servicios y establecer configuraciones. Las entidades incluyen usuarios, sistemas legados, diferentes tipos de CPS como dispositivos, productos y procesos. CPPS, la fábrica y su ambiente.		
<b>Elemento: capa de comunicación – Iteración 3</b>		
Computación ubicua e IoT	Sensores, Imágenes, Etiquetas RFID, Kinect, Teléfonos móviles, Localización (GPS), Drones, AGVs	Ubicua Tiempo real
Protocolos IoT	Bluetooth, Ultra, Wideband, Redes de largo alcance, Fibra óptica, ZigBee, Zwave, WiFi, Telefonía móvil.	Holística
<b>Elemento: integración – Iteración 4</b>		
REST	Desacoplamiento de las aplicaciones mediante HTTP como plataforma para integrar aplicaciones y servicios (RedHat, 2021).	Integración Modularidad y bajo acoplamiento
JSON y AVRO	Estructura de mensajes e intercambio de datos estructurados y no estructurados	
<b>Elemento: orientación a servicios – Iteración 5</b>		
Gemelo Digital	Reflejar entidades físicas en el mundo digital con respecto a su forma, posición, estados y movimientos.	Virtualización
<b>Elemento: orientación a servicios – Iteración 6</b>		
Servicios	Servicios de valor agregado y asistencia inteligente para monitoreo, diagnóstico, predicción, simulación, administración de la salud de activos y procesos, entre otros (Tao et al., 2019).	Orientación a servicios Decisiones en base a datos Modularidad Tiempo real Equilibrio de la vida laboral
<b>Elemento: capa en la nube – Iteración 7</b>		
Computación en la nube	Los datos, modelos, servicios son desplegados brindando soporte a las decisiones de forma descentralizada, distribuida, en una plataforma con la capacidad de manejar altos volúmenes de transacciones, datos estructurados y no estructurados, alta disponibilidad, escalabilidad y rendimiento en la ejecución de servicios (R & R, 2017).	Rendimiento en tiempo real Robustez Flexibilidad y escalabilidad Descentralización
<b>Elemento: ciberseguridad – Iteración 8</b>		
NIST CSF	Enfoque prioritario, flexible, repetible y rentable para identificar, evaluar y gestionar riesgos cibernéticos	Ciberseguridad Robustez

En el quinto paso se generaron los elementos nuevos mediante los patrones utilizados y se asignan sus responsabilidades, dando como resultado estructuras que en este caso son tanto físicas como lógicas. En el paso 6 se definieron relaciones y propiedades para los elementos instanciados. Las iteraciones en el método ADD se llevan a cabo mientras sea necesario tomar decisiones de diseño adicionales para satisfacer los ASR

o hasta que termine el tiempo estipulado para diseñar. Para esta investigación fueron necesarias 8 iteraciones para satisfacer los requerimientos funcionales y no funcionales especificados. La Figura 1 muestra la estructura general de la arquitectura propuesta.

#### 4. Arquitectura de referencia para Industria 4.0

En esta sección se describe la arquitectura de referencia propuesta, diseñada bajo los principios de diseño mencionados en la sección anterior para cumplir con los requisitos de la Industria 4.0.

##### 4.1. Clientes y entidades ciber-físicas

La Industria 4.0 propone integrar tecnología innovadora en procesos industriales para alcanzar una fábrica digital e inteligente, personalizable (Almada-Lobo, 2015), dirigida por datos mediante servicios de valor agregado (Karatas et al., 2022). A diferencia de los sistemas de manufactura tradicionales que constan de la colección de varios procesos de manufactura y transformación de la materia prima (materiales) mediante equipo (máquinas, energía) y capital humano, con el objetivo de fabricar productos (Groover, 2001). En Industria 4.0, estos recursos típicos de producción deben ser convertidos en dispositivos inteligentes capaces de sentir, actuar y tener comportamiento dentro de su ambiente, evolucionando en la forma de CPS (Rosen, 2019). Los CPS son divididos en 3 niveles, de acuerdo con la implementación de sus tareas según sus reglas físicas, funcionalidad, estructura e incertidumbre dentro de su ambiente. Estos niveles son unidad, sistema y sistema de sistemas (Tao et al., 2019). En la Figura 2, se observan los niveles jerárquicos de los distintos CPS en el sistema de producción de acuerdo con su funcionalidad y estructura hasta alcanzar la fábrica inteligente. Mediante estos elementos ciber-físicos trabajando en conjunto mediante la arquitectura y plataforma adecuada, los ingenieros podrán rediseñar fábricas, procesos y aplicaciones para adaptarse a los nuevos desarrollos tecnológicos y a la manufactura de productos en instalaciones cada vez más complejas e inteligentes (Belman-López et al., 2020).

##### 4.2. Comunicación

La conectividad brinda a las empresas la oportunidad de agregar valor a productos y servicios, desarrollando nuevas ofertas y funciones para sus mercados (PwC Middle East, 2018). La Industria 4.0 requiere de un medio de comunicación que brinde seguridad, confiabilidad y soporte a la toma de decisiones en el momento justo (Kagermann et al., 2013). La capa de comunicación permite la conexión en tiempo real entre entidades físicas y virtuales, incluyendo dispositivos relacionados con la captura y transferencia para su posterior almacenamiento y cómputo en un entorno distribuido (Xu & Duan, 2018). La computación ubicua y el IoT habilitan a recursos humanos y físicos, el acceso y captura de datos e información sin restricciones de momento y lugar (tiempo real), siendo la mejor forma de obtener datos provenientes del entorno (Tian & Zhao, 2015). La *capa de captura ubicua (en tiempo real)* está conformada por tecnología que incluye sensores ubicuos, etiquetas RFID, imágenes, videos, Kinect, GPS (localización), entre otras, que aumentan la visibilidad (holística) y generan nuevas oportunidades de negocio (Chen & Tsai, 2016).

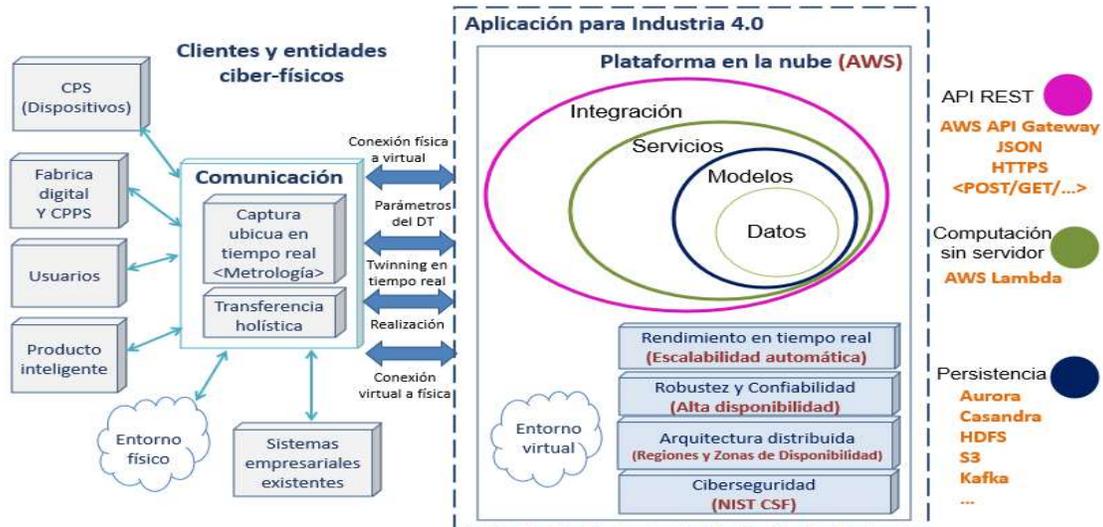


Figura 1. Estructura general de la arquitectura.



Figura 2. Jerarquía de los CPS de acuerdo con su funcionalidad y estructura.

La *capa de transferencia* transmite los datos capturados y comunica a todas las entidades físicas mediante diferentes protocolos IoT. Entre los protocolos de transferencia más utilizados se encuentran Bluetooth (para redes de vehículos y aplicaciones de detección), ZigBee (protocolo para redes WSN con bajo consumo de energía), Zwave (para casas inteligentes y áreas de la salud con bajo consumo de energía), WiFi (redes inalámbricas basadas en IEEE 802.11), NFC (sistemas de pagos mediante teléfonos celulares), redes de largo alcance (SigFox, Neul, LoRaWAN) y telefonía móvil (para ciudades inteligentes y aplicaciones ambientales que transmiten datos en muchos kilómetros) (Yang et al., 2019).

### 4.3. Integración

La integración es importante porque las aplicaciones no pueden vivir aisladas unas de otras. La *capa de integración* permite descubrir y exponer servicios, aplicaciones y datos valiosos a través de APIs (Hohpe & Woolf, 2004). Las aplicaciones web utilizan HTTP por definición, pero en sus inicios, los desarrolladores dedicaban mucho esfuerzo en implementar características de tecnologías distribuidas como CORBA y DCOM sobre HTTP. Esto llevó a tecnologías como SOAP. La experiencia demostró que estas tecnologías eran más complejas, pesadas y frágiles. La idea que reemplazó a SOAP fue que se podía usar HTTP de manera más directa y con menos tecnología involucrada. Las aplicaciones web modernas son más simples que SOAP y se implementan de forma más ligera sobre HTTP. La especificación para HTTP

es uno de los estándares mejor redactados, diseñados y universalmente aceptados. REST es el nombre que se le ha dado al estilo arquitectónico mediante HTTP. HTTP es la realidad, REST es un conjunto de ideas de diseño que le dieron forma y nos ayuda a comprender cómo pensar y usar HTTP. El término RESTful ha surgido para aplicaciones web que combinan conceptos nativos adicionales de HTTP, siendo el estilo dominante para el desarrollo de aplicaciones y servicios. REST ha demostrado escalabilidad y encaja bien con modelos de dominios, pero sobre todo minimiza el acoplamiento (facilidad con la que se puede cambiar una parte sin romper otra) entre aplicaciones distribuidas. Minimizar el acoplamiento es una de las cosas más difíciles de lograr en el diseño de aplicaciones y la eficacia de HTTP a este respecto es innegable (admitiendo clientes HTTP escritos hace 20 años). El análisis de cómo HTTP logra este nivel de desacoplamiento es de los aspectos más destacados de REST, y una de las principales razones para utilizar HTTP como plataforma para integrar aplicaciones y servicios (Google Cloud, 2018).

Cuando el cliente envía una solicitud a través de RESTful, esta transfiere una representación del estado del recurso requerido a quien lo haya solicitado. Este mensaje se entrega por medio de HTTP. El estándar utilizado para la estructura de estos mensajes es JSON, ya que tanto las máquinas como las personas lo comprenden y no depende de ningún lenguaje (RedHat, 2021). Por su parte, Avro permite el control de versiones en la estructura de los mensajes. De esta manera, si la estructura del mensaje cambia (nueva versión) las aplicaciones y servicios que consumen los datos pueden continuar procesando mensajes sin requerir ningún cambio o actualización (The Apache Software Foundation., 2020).

### 4.4. Virtualización (desarrollo del gemelo digital)

Hoy en día, el diseño de productos no puede limitarse únicamente al diseño de un dispositivo físico. Más bien, el diseño debe abarcar el dispositivo, los servicios habilitados por el dispositivo y la infraestructura que respalda esos servicios. Una de las arquitecturas más utilizadas para modelar el DT es proporcionada por Tao et al. (2018), que consiste en modelar el DT utilizando 5 dimensiones. Estas son entidades físicas, modelos virtuales, servicios, datos y conexiones. Stark

et al. (2019) analizaron el desarrollo y operación del DT en 8 dimensiones: amplitud de integración, modos de conectividad, frecuencia de actualización, inteligencia del CPS, capacidades de simulación, riqueza del modelo digital, interacción humana y ciclo de vida del producto. Aunque el modelado del DT en 5 y 8 dimensiones propuestos por Tao et al. (2018) y Stark et al. (2019) describen gran parte de los elementos principales del DT, hay ciertas características que quedan a la deriva. Jones et al. (2020) caracterizó el DT en 13 características fundamentales: entidad física, entidad virtual, entorno físico, entorno virtual, estado y parámetros, realización, metrología, hermanamiento (twinning), tasa de hermanamiento, conexión física a virtual, conexión virtual a física, procesos físicos y procesos virtuales. Una entidad física representa un elemento físico en el mundo real, como un vehículo autónomo, un dispositivo, un producto, un sistema o una fábrica. Una entidad o modelo virtual es la réplica de la entidad física en el mundo virtual (Qi et al., 2019). El entorno físico se refiere al espacio medible y cuantificable del mundo real donde se encuentra la entidad física. Los parámetros se refieren a tipos de datos (geometría, funcionalidad, ubicación, actividades, tiempos, rendimiento, entre otros) e información, transmitidos entre la entidad física y virtual. El estado representa las condiciones actuales del DT y parámetros medidos, mientras que la fidelidad representa la precisión y el grado de replicación con respecto a la entidad física, describiendo la cantidad de parámetros transferidos entre ambos mundos. El hermanamiento consiste en la sincronización del estado y parámetros físicos y virtuales, por ejemplo, medir los parámetros actuales en la entidad física y llevar esos mismos valores o estado al modelo virtual. La tasa de hermanamiento es la frecuencia en la que se produce la sincronización, ya sea del mundo físico al virtual o viceversa, generalmente expresada en tiempo real. Esto implica que cualquier cambio en un gemelo (físico o virtual) debe reflejarse instantáneamente en el otro. La conexión física a virtual mide los parámetros físicos (metrología), transfiere los valores y los realiza en el entorno y modelo virtual. La conexión virtual a física cuantifica, transfiere y realiza la información del modelo virtual a la entidad física. Esta conexión implica la funcionalidad para llevar a cabo el cambio en el estado físico. Por lo tanto, los procesos (o servicios) virtuales determinan las medidas óptimas para los parámetros, mientras que la conexión virtual a física (junto con la metrología y la realización) asegura que esos valores óptimos se apliquen y lleven a cabo de manera efectiva en la entidad física. Finalmente, los procesos (o servicios) virtuales se refieren a actividades realizadas utilizando el modelo virtual dentro del entorno virtual. Estos servicios permiten realizar actividades de predicción, simulación, experimentación, optimización virtual y verificación apoyados en tecnologías en la nube, algoritmos de aprendizaje automático, aprendizaje profundo y técnicas de Big Data. Cabe destacar que el modelado del DT utilizado en esta investigación es el propuesto por Jones et al. (2020).

#### 4.5. Servicios basados en datos y computación sin servidor

La Industria 4.0 abre nuevas oportunidades de negocio mediante servicios innovadores, donde el cambio del enfoque centrado en el producto al enfoque servicios mediante el producto es clave. Desde la perspectiva del DT, los servicios se refieren a actividades o funcionalidades realizadas

utilizando el modelo virtual dentro del entorno virtual y deben ser proporcionados sin restricciones de tiempo y lugar a todas las compañías bajo una composición modular, flexible y basada en datos (Wiesner & Thoben, 2016).

La *capa de servicios* implementa funcionalidades para optimizar, simular o monitorear activos físicos y virtuales, incluyendo servicios de gestión de la salud de activos (Lee et al., 2015). Los servicios brindan información del sistema en cuestión, generan predicciones y diagnósticos mediante métodos de inteligencia artificial y presentan el conocimiento adquirido a los usuarios, brindando un soporte para la toma de decisiones informadas (Jones et al., 2020). Los servicios de gestión de la salud monitorean el rendimiento, la vida útil y desgaste, predicen fallas, detectan degradación en el rendimiento, descubren ineficiencias y programan mantenimientos sin interrumpir las operaciones (Caggiano, 2018; Lu et al., 2019).

Los servicios están basados en análisis de datos, generalmente clasificados en tres tipos: análisis descriptivos (mediante funciones de estadística descriptiva, análisis de correlación, análisis clúster, entre otros), análisis predictivos y análisis prescriptivos. Los análisis predictivos se han convertido en la mejor fuente para extraer conocimientos relacionados con la producción (Kusiak, 2017), categorizados en 5 tipos: regresión, árboles de decisión, redes neuronales artificiales, máquinas de vector de soporte y análisis bayesianos. Finalmente, los análisis prescriptivos son capaces de buscar el plan óptimo con el costo total más bajo, existiendo dos tipos principales: programación matemática y búsquedas heurísticas. La programación matemática (como la programación lineal, programación entera, programación dinámica, programación no lineal o el método de ramificación y acotamiento) está diseñada para encontrar la solución óptima global mientras las búsquedas heurísticas (algoritmo hill climbing, optimización por colonias, algoritmos genéticos, búsqueda tabú, recocido simulado) están diseñadas para encontrar soluciones óptimas locales en tiempos cortos (Xu & Duan, 2018). Para implementar los servicios, se utilizó el paradigma de computación sin servidor. El término "sin servidor" significa que su código se ejecuta (en la forma de funciones) en servidores sin la necesidad de aprovisionar ni administrar estos y se factura por el tiempo en que el código está en ejecución (Mishra, 2019). La computación sin servidor permite centrarse en innovar nuevos productos y funciones en lugar de mantener los servidores. Otro beneficio de la computación sin servidor es la flexibilidad para escalar funciones, servicios y aplicaciones automáticamente, además de permitir ajustar las capacidades modificando recursos y memoria si es necesario (Poccia, 2016). El código se activa en respuesta a eventos para una gran variedad de escenarios, como la carga de un archivo en un almacenamiento, un cambio en una tabla o la llegada de datos. Además, es altamente escalable y capaz de ejecutar instancias en paralelo, proporcionando un muy alto rendimiento y disponibilidad en respuesta a múltiples eventos concurrentes mientras la plataforma en la nube administra el aprovisionamiento de recursos en segundo plano.

#### 4.6. Arquitectura distribuida, escalable y descentralizada

Industria 4.0 requiere desarrollar sistemas y servicios descentralizados, donde la descentralización no es necesariamente física, pero sí lógica, por ejemplo, una

aplicación o un CPS con la capacidad de identificarse y conectarse a un sistema donde la capacidad de cómputo está en otra parte (Almada-Lobo, 2015). Además, la gran variabilidad en la demanda requiere una plataforma productiva ágil y escalable en la que aplicaciones y servicios puedan ser reconfigurados de forma flexible y la escasez de recursos sea recompensada en corto tiempo para satisfacer las nuevas demandas (Angulo et al., 2016).

La computación en la nube permite el desarrollo de aplicaciones y servicios distribuidos, modulares y desacoplados que poseen una alta disponibilidad (robustez), escalabilidad y alto rendimiento mediante la ejecución de múltiples instancias en paralelo administradas de forma automática por el proveedor en la nube. Además, la computación en la nube administra de forma automática la redundancia necesaria para la ejecución del código, el balanceo de cargas entre las diferentes instancias y la segregación de la infraestructura en varias ubicaciones. De esta forma, los datos, modelos y servicios son desplegados sobre una plataforma en la nube brindando soporte a decisiones complejas de manera descentralizada y distribuida (Xu et al., 2018), con la capacidad de manejar altos volúmenes de transacciones (R & R, 2017), datos estructurados y no estructurados, poseer alto rendimiento en tiempo real, ofrecer una latencia baja en la ejecución de funciones y servicios, poseer alta escalabilidad, disponibilidad, confiabilidad y consistencia en los datos (Gorton & Klein, 2015).

#### 4.7. Ciberseguridad

La ciberseguridad es primordial para el progreso de Industria 4.0 dado el gran incremento en el número de conexiones y vulnerabilidades que no requieren un ataque físico, tal como sería el robo, uso indebido o acceso a información (que aplicaciones y servicios poseen) por personas no autorizadas (Kusiak, 2017). Los ataques a la ciberseguridad explotan la complejidad y conectividad de sistemas e infraestructura, poniendo en riesgo la economía, reputación y salud de la empresa. Además, estos ataques pueden aumentar los costos, reducir los ingresos y dañar la capacidad de una organización para innovar, ganar y mantener clientes (Tuptuk & Hailes, 2018).

Para abordar estos riesgos, el NIST desarrolló un enfoque prioritario, flexible, repetible y rentable, que incluye medidas de seguridad para identificar, evaluar, y gestionar los riesgos cibernéticos conocido como el marco de seguridad cibernética (NIST CSF, por sus siglas en inglés). La arquitectura propuesta utiliza el NIST CSF para proteger las entidades y los datos que estas producen, así como los modelos y servicios que estas utilizan contra el uso indebido y el acceso no autorizado. El NIST CSF se ha convertido en un marco reconocido internacionalmente, que puede ser utilizado por cualquier organización, independientemente del sector o tamaño. El NIST CSF consta de tres elementos: el núcleo, los niveles y los perfiles. El núcleo establece un conjunto fundamental de actividades organizadas en cinco funciones continuas (identificar, proteger, detectar, responder, recuperar) para ayudar a mejorar la seguridad, la gestión de riesgos, la robustez y resiliencia de la organización.

Los niveles caracterizan la aptitud y madurez de una organización para administrar las funciones y controles del CSF, y los perfiles están destinados a definir las posturas de seguridad de la organización "actual" y "futura".

Originalmente, el CSF estaba destinado al sector de infraestructura crítica, pero ha sido respaldado por gobiernos e industrias de todo el mundo como referencia recomendada para organizaciones de todos los tipos y tamaños. La lista de los primeros en adoptarlo a nivel mundial incluye a Japón, Israel, Reino Unido, Uruguay, entre otros. De hecho, en febrero de 2018, la Organización Internacional de Normalización publicó el "ISO / IEC 27103: 2018", un estándar que proporciona orientación para implementar ciberseguridad aprovechando los estándares existentes. ISO 27103 promueve los mismos conceptos y mejores prácticas reflejados en el NIST CSF. Cabe resaltar que estas funciones no están destinadas a conducir a un estado final estático deseado, más bien, las funciones deben realizarse de forma simultánea y continua para formar una cultura operativa que aborde el riesgo dinámico de ciberseguridad (NIST, 2018).

## 5. Caso de uso: Agricultura 4.0

La arquitectura es la base para la construcción del sistema mientras la implementación transforma casos de uso en aplicaciones para los usuarios, evitando desviaciones respecto de la arquitectura definida. Además, la implementación permite demostrar la viabilidad y los beneficios de la arquitectura. El caso de uso está enfocado en el área de la agricultura y consiste en desarrollar una aplicación que en tiempo real detecte plagas y plantas enfermas a partir del video de la cámara de dispositivos móviles (*requerimientos funcionales*). Adicionalmente, la aplicación debe cumplir con la arquitectura diseñada y satisfacer los requerimientos que gobiernan Industria 4.0 (*requerimientos no funcionales*). La detección de plagas tiene gran importancia en términos de productividad y reducción en el uso de plaguicidas, además los métodos convencionales de inspección visual son ineficaces para cierto tamaño del cultivo y a menudo omiten hallazgos por diversas razones. La Industria 4.0 y el aprendizaje profundo (DL) juegan un rol fundamental para incrementar la confiabilidad, calidad y productividad en la agricultura. El tiempo de respuesta es clave para inspeccionar una mayor cantidad de plantas debido a la corta duración de baterías en dispositivos como robots, drones o vehículos autónomos (AGV) que permitan la implementación de sistemas ubicuos y omnipresentes en esta área (Kakani et al., 2020). Uno de los componentes principales de la arquitectura es la plataforma en la nube, donde se eligió Amazon Web Services (AWS), por ser la plataforma en la nube más destacada dada la amplia variedad de servicios que ofrece.

### 5.1. Captura y transferencia

La entrega de valor a partir de los datos incluye desde imágenes, videos, propiedades, parámetros, entre otros (Kusiak, 2017). Las imágenes son el tipo de dato utilizado en este caso de uso, capturadas mediante cámaras que pueden pertenecer a dispositivos como teléfonos móviles, cámaras IP, un drone o un AGV. La imagen es transmitida como una cadena Base64 mientras que para la transferencia de la imagen es posible utilizar la red móvil como 5G o una red WiFi.

### 5.2. Desarrollo del modelo

Para el entrenamiento del modelo se utilizó el conjunto de datos "Plant Doc" para el reconocimiento de plantas enfermas. Este conjunto fue publicado por investigadores del Instituto

Tecnológico de la India en Gandhinagar, representando el esfuerzo de aproximadamente 300 horas humanas (Singh et al., 2019). Pero en este caso se utilizó la versión publicada en Roboflow que corrige algunos errores y puede ser descargada en varios formatos incluyendo YOLOv5. El conjunto posee 2874 imágenes y está dividido en un subconjunto para entrenamiento (90%) y otro para validación (10%). Para las plagas de insectos se utilizó la base de datos creada por Huang & Chuang (2020), conformada por 299 imágenes y ocho plagas comunes del tomate. Este conjunto está originalmente diseñado para clasificación, así que para su uso en detección se realizó el etiquetado de las imágenes de forma manual usando Roboflow-Annotate. En total entre los dos conjuntos de datos, se analizaron 38 clases y 3173 imágenes, 2874 para entrenamiento y 299 para validación.

El modelo para la detección de plagas y enfermedades fue desarrollado utilizando Python en conjunto con PyTorch y YOLOv5m. Para acelerar el entrenamiento de los modelos, se utilizó Google Colaboratory para acceder de manera gratuita a una GPU Tesla K80. PyTorch es una biblioteca ampliamente utilizada para desarrollar modelos de DL. YOLOv5 (Jocher et al., 2021) es una familia de arquitecturas y modelos de detección de objetos previamente entrenados en el conjunto de datos COCO y desarrollados por Ultralytics. YOLOv5 es increíblemente rápido. En Google Colaboratory con una GPU Tesla P100, YOLOv5m reporta una velocidad de 0.0082 segundos por imagen con un  $mAP_{0.5}^{val}$  de 63.9, lo que significa más de 120 imágenes por segundo (FPS), superando a su antecesor YOLOv4 que reporta hasta 50 FPS después de haber sido convertido a la misma biblioteca PyTorch de Ultralytics. Para la fase de entrenamiento se utilizaron imágenes con una resolución de 416, tamaño de lote de 64, 300 épocas, el modelo yolov5m para establecer la arquitectura de la red, configuración y pesos iniciales, el descenso de gradiente estocástico (SGD) como optimizador y un ritmo de entrenamiento de 0.01. El modelo final obtuvo un peso de 42.7MB, una  $mAP_{0.5}^{val} = 0.486$  y una  $mAP_{0.5}^{val} = 0.635$ , superior al 0.389 de los autores de PlantDoc en 2019 mediante la red Faster R-CNN con InceptionResnetV2. El modelo fue almacenado en Amazon Simple Storage Service (S3).

### 5.3. Servicios basados en datos y computación sin servidor

Los servicios deben ser escalables y robustos con la capacidad de procesar miles de peticiones al instante (alta eficiencia en tiempo real). AWS Lambda es un servicio de computación sin servidor (atractivo para aplicaciones escalables, rentables y confiables), que permite ejecutar código sin la necesidad de administrar servidores. AWS Lambda tiene una escalabilidad automática que ejecuta código en respuesta a eventos (arquitectura orientada a eventos), reduciendo la sobrecarga operativa, además de poseer un esquema de facturación de pago por uso. El código es ejecutado en paralelo, procesando cada evento individualmente y escalando en automático según la carga de trabajo, logrando así un alto rendimiento (eficiencia) y confiabilidad (robustez) (Poccia, 2016). El código y todas sus dependencias fueron empaquetadas como una imagen de contenedor utilizando Docker. Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de forma rápida y confiable de un entorno computacional (local o en la nube) a otro. El software en contenedores siempre se ejecutará de la

misma manera, independientemente de la infraestructura o sistema operativo. Los contenedores aíslan el software de su entorno y garantizan que funcione de manera uniforme a pesar de las diferencias (Docker, 2021). AWS permite un tamaño máximo para la imagen de 10 GB mediante Amazon Elastic Container Registry (Amazon ECR). Posteriormente, se creó la función Lambda a partir de la imagen almacenada en Amazon ECR desde donde se puede recuperar el modelo almacenado en S3.

### 5.4. Integración

La capa de integración permite descubrir los servicios y aplicaciones a través de un API capaz de realizar predicciones en tiempo real utilizando los servicios desarrollados mediante computación sin servidor (AWS Lambda). Para desarrollar la API se utilizó Amazon API Gateway para convertir la función Lambda en servicios RESTful que ejecutan código sin servidor en respuesta a eventos HTTP. De esta manera se pueden construir sistemas “backend” de aplicaciones completas, sin aprovisionar un solo servidor (Mishra, 2019). El API Gateway facilita a los desarrolladores la creación, publicación, monitoreo y seguridad de APIs y servicios a cualquier escala. Las APIs actúan como la puerta de entrada para que las aplicaciones accedan a los datos y a la lógica empresarial en tiempo real. El API Gateway gestiona la aceptación y procesamiento de hasta cientos de miles de llamadas simultáneas, la administración del tráfico, compatibilidad con CORS, el control de autorizaciones y accesos, la limitación controlada, el monitoreo y la administración de versiones de los servicios. Finalmente, cuando el cliente envía una solicitud (en este caso de tipo POST) a través de la API RESTful, está transfiere una representación del estado del recurso requerido a quien lo haya solicitado. Este mensaje se entrega por medio de HTTPS (cifrando los datos en tránsito), utilizado JSON como estructura para los mensajes.

### 5.5. Ciberseguridad

La implementación utiliza los servicios de AWS para apoyar al NIST CSF. Para la función **identificar** AWS gestiona los activos que conforman la nube, reduciendo significativamente la gestión de activos. Para la función **proteger**, se utilizó Identity and Access Management (IAM) para administrar las diferentes identidades que interactúan con la nube desde un lugar centralizado, además de grupos, roles y políticas implementando el principio de privilegio mínimo. Además, se debe proteger los servicios utilizados por la aplicación y evitar el acceso no autorizado. Todas las llamadas hacia los servicios deben pasar por una forma de autenticación y autorización. AWS Cognito permite gestionar, proteger, controlar e informar sobre los usuarios que utilizan la aplicación, validando los derechos de acceso y proporcionando un token JWT necesario para acceder a la API.

Para la función **detectar**, CloudWatch monitorea en tiempo real la infraestructura de AWS, aplicaciones y servicios y AWS CloudTrail para proporcionar un registro de las acciones realizadas por un usuario, rol o servicio. Al detectar acciones indebidas, los permisos son inmediatamente revocados en IAM y Cognito (función **responder**). Finalmente, la infraestructura de AWS en diferentes regiones y zonas de disponibilidad permite una alta disponibilidad

(función **recuperar**) de los servicios en la nube como Lambda, S3, API Gateway, Cognito e IAM.

### 6. Resultados y discusión

Una de las ventajas de utilizar una plataforma en la nube es acelerar el tiempo de desarrollo accediendo a recursos computacionales de manera rápida a costos más bajos (beneficio de las economías de escala), sin requerir una inversión inicial significativa. Además, libera a los desarrolladores de invertir tiempo en tareas que no agregan valor para concentrarse en las características que se desea desarrollar. El caso de uso fue desarrollado utilizando la plataforma AWS mediante servicios como Lambda, S3, Cognito, API Gateway, entre otros. La Figura 3 muestra imágenes de la aplicación desarrollada ejecutada en campos de cultivo en tiempo real y la Figura 4 muestra el diagrama de la arquitectura de la implementación, recursos AWS utilizados y sus relaciones.

A continuación, se realiza un desglose de los costos de cada uno de estos servicios. Para los costos de S3 influyen aspectos como la cantidad de almacenamiento ocupado mensualmente, el número de recuperaciones de los objetos almacenados y la cantidad de datos transferidos. Para Lambda el costo se calcula en base a dos aspectos: el tiempo de procesamiento (en milisegundos) y el número de invocaciones a la función. Los costos para AWS API Gateway son en base a cada millón de solicitudes y los precios de Cognito son en base al número mensual de usuarios activos.

La Tabla 2 muestra el desglose de costos para los servicios utilizados en la aplicación durante 1 millón de imágenes analizadas, donde se puede apreciar que tales costos son menores a 5 USD mensuales, una cantidad muy inferior a los costos requeridos para desarrollar estos servicios de forma local (on promise) que adicionalmente debe incluir gastos de Internet, electricidad, seguridad, renta de inmuebles, entre otros. Además, se evita tener que predecir la capacidad requerida por aumentos repentinos en la demanda de la aplicación y de los servicios cognitivos desarrollados.



Figura 3. Aplicación desarrollada y ejecutada en tiempo real en campos de cultivo.

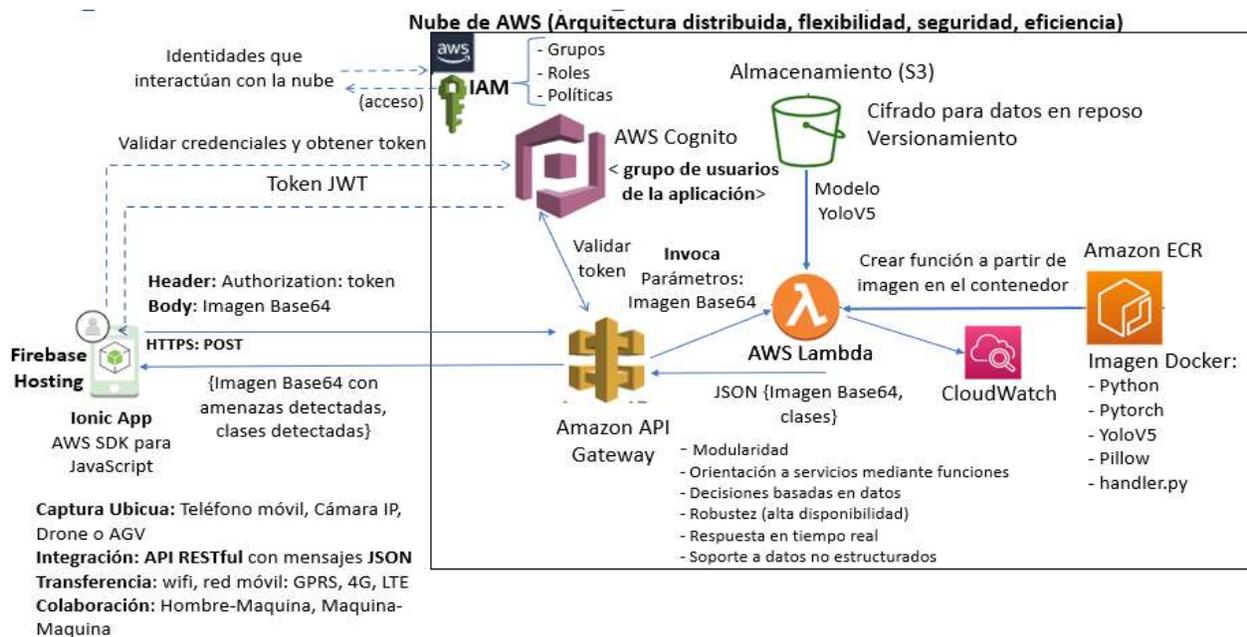


Figura 4. Arquitectura de la solución (implementación).

Tabla 2. Costos de los servicios utilizados para la implementación.

Servicio	Medida	Tarifa (USD)	Implementación	Total (USD)
<b>S3 Standard</b>				
Almacenamiento	50 TB/mes	0.023	50 MB	0.023
Recuperaciones	1000 solicitudes	0.0004	solo en cold start	< 1
Transferencia	1 GB	0.033	solo en cold start	< 1
<b>Lambda</b>				
	Configuración: x86, 1024MB	Tiempo de Procesamiento = Duración de la solicitud (600ms) * Solicitudes - Capa Gratuita		
Tiempo de Procesamiento	GB memoria / milisegundo	0.0000000167	200,000,000	3.34
Solicitudes	1 millón de solicitudes	0.2	1,000,000	0.2
<b>API Gateway</b>				
Llamadas	1 millón de llamadas	1	1,000,000	1
<b>Cognito</b>				
Usuarios activos	1 a 50 mil usuarios activos mensuales	0	1 usuario	0
<b>CloudWatch</b>				
	Funcionalidad de la capa gratuita	0		0
<b>Firebase</b>				
	Capa gratuita	10 GB	1.8 MB	0
			Total	4.563

El diseño de una arquitectura consiste en desarrollar una solución que cumple con los requerimientos, enfocándose generalmente en los atributos de calidad (requerimientos no funcionales) que habilita. A continuación, se analiza la forma en que la implementación llevada a cabo cumple con los 17 principios (requerimientos) que gobiernan Industria 4.0.

**Personalización** mediante el desarrollo de aplicación que utiliza un modelo personalizado para detectar plagas y plantas enfermedades a través de la red YOLOv5m que detecta 13 especies de plantas y 30 clases (entre enfermas y sanas), además de 8 plagas del tomate.

**Colaborativa:** la arquitectura habilita la cooperación entre múltiples disciplinas (computación en la nube, tecnología ubicua, CPS, análisis de datos, entre otras) y permite el trabajo conjunto entre personas y dispositivos como CPS (colaboración humano-máquina) enriquecidos en sus capacidades computacionales a través de modelos y servicios desplegados en plataformas en la nube (colaboración máquina-máquina).

**Decisiones basadas en datos:** esta propuesta basada en los conjuntos de datos Plant Doc y Mendelay Data permite desarrollar un modelo (basado en YOLOv5m) consumido en funciones (sin servidor) robustas para la detección temprana y en tiempo real de plagas y plantas enfermas para reducir hallazgos que pueden ser omitidos, incrementar la calidad, la satisfacción del cliente y la productividad en los campos de cultivo.

**Modular:** la aplicación está dividida en capas separadas (separation of concerns) que poseen alta cohesión como son la interfaz de usuario, el controlador (API Gateway), servicios que incluyen la lógica de negocio y persistencia (S3). Las interfaces de comunicación fueron estandarizadas mediante un API tipo REST para exponer servicios. **Reutilización:** se utilizó la red YOLOv5m para el desarrollo ágil del modelo de aprendizaje profundo, aumentando la fiabilidad, al ser código intensamente probado y que ha sido utilizado en otras investigaciones científicas.

**Orientación a servicios** utilizando el paradigma de computación sin servidor mediante AWS Lambda para desarrollar un servicio que no requiere una inversión inicial para su despliegue, capaz de procesar miles de peticiones al instante, posee escalabilidad automática, alta disponibilidad sin restricciones de lugar o momento y que es facturado sobre demanda a un costo de \$0.0000000167 USD por petición.

**Arquitectura descentralizada y distribuida** donde la computación en la nube proporciona una solución efectiva a este desafío (Xu et al., 2018). La API y los servicios desarrollados se almacenan en la nube, brindando soporte a la toma de decisiones. La región AWS utilizada para el despliegue de los servicios es el Norte de Virginia (us-east-1) que posee 6 zonas de disponibilidad (AZ). En AWS, una región es una ubicación física en el mundo donde se agrupan centros conocidos como zonas de disponibilidad (AZ). Cada región consta de varias AZ aisladas y separadas físicamente dentro de un área geográfica. Una AZ consta de uno o más centros discretos con alimentación, redes y conectividad redundantes. Las AZ brindan niveles de disponibilidad, tolerancia a errores y escalabilidad mayores que el que ofrecería un centro de datos único. El tráfico entre las AZ está cifrado. Las AZ están físicamente separadas entre sí por una distancia significativa, aunque todas están dentro de un rango de 100 km (60 millas) de separación (AWS, 2022).

**Manejo de información en tiempo real:** la recopilación de datos mediante dispositivos ubicuos como el teléfono celular garantiza una colección en tiempo real y baja latencia de transmisión mediante la red móvil. Las regiones de AWS y su infraestructura ofrecen alto rendimiento de red mediante fibra de 100Gb Ethernet redundantes entre regiones (Amazon Web Services, 2022). Finalmente, el tiempo promedio de respuesta de la función Lambda (servicio) mediante 2048 Mb de memoria y el modelo basado en YOLOv5m es de 600 ms.

**Robusta y confiable:** AWS sostiene que mediante su infraestructura desarrollada en regiones compuestas por diferentes zonas de disponibilidad, servicios como Amazon S3 tienen una disponibilidad garantizada del 99.99% y una durabilidad del 99.999999999%. Mediante AWS CloudWatch se programó un evento que invoca la API desarrollada cada 3 minutos. Este evento ejecutado por 1 mes (14880 invocaciones) encontró una disponibilidad del 100 %.

**Flexibilidad:** los clientes solían utilizar redundancia para asegurar la suficiente capacidad para responder al máximo nivel de actividad. La nube (AWS) permite lograr niveles de flexibilidad y escalabilidad muy altos, aprovisionando la cantidad de recursos que realmente se necesitan, aumentando o disminuyendo instantáneamente junto con las necesidades, reduciendo costos y mejorando la capacidad del cliente para satisfacer las demandas de usuarios e incrementos en la carga de trabajo (Amazon Web Services, 2022). La concurrencia se refiere a la cantidad de instancias atendiendo solicitudes. Para el tráfico inicial, la concurrencia es de 3000 según la región utilizada. Cuando las solicitudes llegan más rápido comparado a la escalabilidad las solicitudes fallan por un error de limitación (código http 429). Para probar la escalabilidad de AWS se utilizó la herramienta artillery.io, realizando una simulación de 5 nuevos usuarios accediendo a la aplicación (prueba de carga) cada segundo durante 10 min, resultando en 3000 invocaciones y un 100% de ejecuciones exitosas.

**Ubicua:** la aplicación posee la capacidad de ejecución sin restricciones de tiempo y lugar mediante el uso de tecnología ubicua como teléfonos celulares (e incluso un dron o AGV), redes de tecnología móvil, servicios REST y computación en la nube.

**Integración:** se utilizó Amazon API Gateway para convertir la función Lambda en un servicio RESTful y un API que ejecuten código sin servidor en respuesta a eventos HTTP. La API desarrollada actúa como puerta de entrada para que las aplicaciones accedan a los datos y a la lógica empresarial en tiempo real. El API Gateway facilita la creación, publicación, monitoreo y seguridad de APIs y servicios. Además, gestiona la aceptación y procesamiento de miles de llamadas simultáneas, la administración del tráfico, compatibilidad con CORS, el control de autorizaciones y accesos, la limitación controlada, el monitoreo y la administración de versiones.

**Holística:** la capa en la nube permite tener una visión holística de todos los procesos computacionales (AWS CloudWatch). La aplicación desarrolla una visión holística del proceso físico, integrando imágenes de campo en tiempo real a través de dispositivos móviles con acceso temporal y espacialmente independiente a funciones y servicios. Esto permite una evaluación y mejora de la salud de los cultivos y la integración de dispositivos, sistemas y la toma de decisiones. El flujo de información está disponible cuando y donde es necesario.

**Eficiencia:** mayor eficiencia de los recursos computacionales invertidos en la modalidad de pago-por-uso, además de acelerar el tiempo de desarrollo accediendo a recursos rápidamente. El costo de la implementación mediante computación sin servidor, que además elimina gastos y tiempos relacionados a la adquisición y administración de infraestructura, es de 4.6 dólares por millón de imágenes analizadas. Para analizar la eficiencia se consideró la implementación del servicio en instancias de Amazon Elastic Compute Cloud (EC2) equivalentes y la cantidad de tiempo de cómputo necesario para procesar el millón de imágenes. AWS Lambda brinda 6 vCPU y hasta 10 Gb de RAM. Las instancias EC2 más cercanas como c5.2xlarge con 8 vCPU y 16 Gb de RAM a un costo de \$0.34 USD por hora o t3.xlarge con 4 vCPU y 16 Gb de RAM a un costo de \$0.1664 USD por hora generarían un costo de \$66.11 y \$32.35 USD respectivamente, muy superior a los \$4.6 utilizando AWS Lambda. Esto sin considerar que con Amazon EC2, el usuario es responsable de aprovisionar la capacidad necesaria, monitorizar el estado y desempeño de recursos y servicios, además de diseñar la escalabilidad y la tolerancia a errores. Finalmente, el modelo desarrollado tiene un mayor rendimiento en la detección con una  $mAP_{0.5}^{val} = 0.635$ , superior al 0.389 reportado por los autores originales de PlantDoc mediante la red Faster R-CNN con InceptionResnetV2.

**Virtualización:** implementación del DT utilizando el modelo propuesto por Jones et al. (2020) basado en 13 características; entidad física: plantas enfermas y plagas; entidad virtual: modelo YOLOv5m almacenado en S3; entorno físico: campos de cultivo; entorno virtual: plataforma en la nube de AWS; metrología: imágenes capturadas mediante cámaras de teléfonos móviles, cámaras IP, Drones o un AGV; parámetros: imágenes capturadas y transmitidas como cadenas Base64; hermanamiento y tasa de hermanamiento: cada 0.5 segundos; realización: envío de una solicitud (POST) mediante HTTPS (cifrando los datos en tránsito) y JSON como estructura para los mensajes; conexión física a virtual: API desplegada en la

nube y disponible mediante WiFi o telefonía móvil; procesos físicos: detección de plagas y plantas enfermas; procesos virtuales: servicio para detección mediante AWS Lambda; conexión virtual a física: cliente ejecutando acciones y tomando decisiones en base a la información proporcionada por la aplicación y el servicio de detección en tiempo real.

**Ciberseguridad:** La implementación utiliza los recursos de AWS para llevar a cabo el NIST CSF, donde se realizaron acciones para las funciones identificar, proteger, detectar, responder y recuperar, desarrollando un enfoque prioritario, flexible y repetible, que incluye medidas de seguridad para identificar, evaluar, y gestionar los riesgos cibernéticos.

**Equilibra la vida laboral:** la aplicación mediante servicios para análisis de imágenes sin restricciones de tiempo y lugar buscar ser un apoyo en la toma de decisiones, reducir el número de hallazgos que pueden ser omitidos y el tiempo de capacitación para identificar plagas y plantas enfermas, además del ahorro en esfuerzo físico al utilizar los servicios desde un Drone o AGV.

**Inteligencia:** el servicio desarrollado mediante la red YOLOv5m basada en redes neuronales convolucionales genera un conocimiento del proceso monitoreado, brinda diagnósticos y soporte en la toma de decisiones. Este servicio junto con la conexión virtual-a-física del modelado del DT, permiten cerrar el ciclo entre las hipótesis y decisiones generadas en el entorno virtual y las consecuencias llevadas a cabo en el entorno físico. Un aspecto que se omite con frecuencia en la literatura es el rol de los operadores en el ciclo del DT. Si alguien utiliza el DT para predecir la salud de una maquinaria usando servicios implementados por modelos de inteligencia artificial para finalmente enviar ingenieros a reemplazar los componentes necesarios en la maquinaria. El ingeniero en este escenario realiza el proceso de conexión virtual a física, realización y configuración, cerrando el ciclo del DT. Este es el enfoque utilizado en esta implementación, donde las decisiones apoyadas en los servicios de inteligencia artificial son llevadas a cabo por los ingenieros correspondientes, cerrando el ciclo de DT y cumpliendo las definiciones en el modelado de DT propuesto por Jones et al. (2020).

## 7. Conclusiones y trabajo futuro

Aunque muchas compañías aún dudan en realizar una implementación para Industria 4.0, dado el desafío de la inversión inicial, la computación en la nube ofrece una solución viable, rentable y a costos accesibles para tal desafío, cambiando la forma de implementar servicios y permitiendo a compañías de cualquier tamaño construir aplicaciones poderosas y escalables. La arquitectura de referencia tiene la capacidad de integrar el enorme abanico de tecnologías involucradas desde análisis de datos, aprendizaje profundo, visión artificial, ciberseguridad, CPS, sensores ubicuos, computación en la nube, entre otras. Además, el diseño de esta arquitectura desarrolla una solución que cumple con los requerimientos, enfocándose principalmente en los atributos de calidad (requerimientos no funcionales) que habilita. La arquitectura y no solo las aplicaciones pueden convertirse en activos dentro de la organización. Además, la arquitectura transforma a entidades físicas individuales en una "empresa conectada", proporcionando un marco que estimula la creación de valor y difusión del conocimiento a través de los

datos. La API y los servicios desarrollados mediante el paradigma de computación sin servidor poseen alto rendimiento, disponibilidad, escalabilidad, robustez, modularidad, seguridad, y pueden ser utilizados por otros colaboradores para intercambiar información y tomar decisiones en tiempo real. La colaboración entre entidades físicas, humanas y virtuales forma parte de un ecosistema conectado que proporciona: reducción en el tiempo de las operaciones, menor número de defectos, conocimientos del sistema monitoreado, visibilidad en tiempo real de extremo a extremo, virtualización de las operaciones de producción e incrementa la productividad. Además, la computación en la nube permite la visión y evaluación de todos los recursos y estructuras virtuales (holística) permitiendo el monitoreo y la mejora continua.

La mejora en las redes de comunicación como la telefonía móvil permite incorporar desarrollos tecnológicos innovadores conectando un mayor número de dispositivos (como drones o AGVs) debido a su mayor velocidad y alcance permitiendo a las aplicaciones tener una capacidad de datos mucho mayor con tiempos de respuesta más cortos en una diversidad más amplia de lugares. Aunque el rendimiento y estabilidad de la infraestructura de comunicación aún tiene impacto directo en el rendimiento de las aplicaciones holísticas ejecutadas en lugares lejanos o de difícil acceso.

En cuanto al trabajo futuro destaca la implementación de los servicios de autoconfiguración que permitan llevar a cabo la ejecución de servicios de forma autónoma e independiente, además de la inclusión de personas y su conexión con el taller. La representación de una persona, incluidos datos personales, de salud, actividades y estado emocional, puede ayudar a establecer modelos para comprender el bienestar personal y las condiciones laborales. En base a la comprensión del estado de los trabajadores es posible diseñar estrategias de colaboración humano-máquina (centradas en el trabajador) para mejorar la salud física y psicológica de los trabajadores, y mejorar el rendimiento de la producción.

## Agradecimientos

Los autores agradecen al Consejo Nacional de Ciencia y Tecnología de México (CONACYT) por financiar esta investigación mediante una beca para estudios de posgrado (CVU 773443), al TecNM por el apoyo recibido a través de la convocatoria "Proyectos de Desarrollo Tecnológico e Innovación 2022" y al Dr. José Enrique Botello Álvarez por sus valiosas observaciones en el desarrollo de esta investigación.

## Referencias

- Aheleroff, S., Xu, X., Zhong, R., & Lu, Y. (2021). Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model. *Advanced Engineering Informatics*, 1-15. doi:https://doi.org/10.1016/j.aei.2020.101225
- Almada-Lobo, F. (2015). The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES). *Journal of Innovation Management*, 16-21.
- Amazon Web Services. (2022). Infraestructura Global. Obtenido de AWS: <https://aws.amazon.com/es/about-aws/global-infrastructure/>
- Angulo, P., Guzmán, C., Jiménez, G., & Romero, D. (2016). A service-oriented architecture and its ICT infrastructure to support eco-efficiency performance monitoring in manufacturing enterprises. *International*

- Journal of Computer Integrated Manufacturing*, 202-214. doi:https://dx.doi.org/10.1080/0951192X.2016.1145810
- AWS. (2022). Regiones y zonas de disponibilidad. Obtenido de AWS: [https://aws.amazon.com/es/about-aws/global-infrastructure/regions\\_az/](https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/)
- Azeem, M., Haleem, A., Shashi, B., Javaid, M., Suman, R., & Nandan, D. (2021). Big data applications to take up major challenges across manufacturing industries: A brief review. *Materials Today: Proceedings*, 1-10. doi:https://doi.org/10.1016/j.matpr.2021.02.147
- Bader, S., Berres, B., Boss, B., Gatterburg, A., & Hoffmeister, M. (Noviembre de 2021). Plattform Industrie 4.0. Obtenido de Details of the Asset Administration Shell - Interoperability at Runtime – Part 2: Exchanging Information via Application Programming Interfaces: [https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details\\_of\\_the\\_Asset\\_Administration\\_Shell\\_Part2\\_V1.html](https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html)
- Bauer, J. (24 de Feb de 2021). Using container images to run TensorFlow models in AWS Lambda. Obtenido de AWS: <https://aws.amazon.com/es/blogs/machine-learning/using-container-images-to-run-tensorflow-models-in-aws-lambda/>
- Belman-Lopez, C., Jiménez-García, J., & Hernández-González, S. (2020). Análisis exhaustivo de los principios de diseño en el contexto de Industria 4.0. *Revista Iberoamericana de Automática e Informática Industrial*, 432-447. doi:https://doi.org/10.4995/riai.2020.12579
- Belman-López, C., Jiménez-García, J., Vázquez-López, J., Hernández-González, S., & Franco-Barrón, J. (2020). Elementos fundamentales del sistema de manufactura inteligente en la era de Industria 4.0. *Revista Internacional de Investigación e Innovación Tecnológica*, 1-26.
- Caggiano, A. (2018). Cloud-based manufacturing process monitoring for smart diagnosis services. *International Journal of Computer Integrated Manufacturing*, 31(7), 612-623. doi:https://doi.org/10.1080/0951192X.2018.1425552
- Carnell, J. (2017). *Spring Microservices in Action*. NY: Manning Publications Co.
- Cervantes Maceda, H., Velasco-Elizondo, P., & Castro Careaga, L. (2016). *Arquitectura de Software. Conceptos y ciclo de desarrollo*. Ciudad de México, México: CENGAGE Learning.
- Charro, A., & Schaefer, D. (2018). Cloud Manufacturing as a new type of Product-Service System. *International Journal of Computer Integrated Manufacturing*, 1018-1033.
- Chen, T., & Tsai, H.-R. (2016). Ubiquitous manufacturing: Current practices, challenges, and opportunities. *Robotics and Computer-Integrated Manufacturing*, 1-7. doi:https://dx.doi.org/10.1016/j.rcim.2016.01.001
- Dintén, R., López Martínez, P., & Zorrilla, M. (2021). Arquitectura de referencia para el diseño y desarrollo de aplicaciones para la Industria 4.0. *Revista Iberoamericana de Automática e Informática Industrial*, 300-311. doi:https://doi.org/10.4995/riai.2021.14532
- Docker. (2021). Obtenido de Docker: <https://www.docker.com/>
- Francalanza, E., Borg, J., & Constantinescu, C. (2018). Approaches for handling wicked manufacturing system design problems. *Procedia CIRP*, 67, 134-139. doi:https://doi.org/10.1016/j.procir.2017.12.189
- GE. (01 de Noviembre de 2018). Predix Platform | GE Digital. Obtenido de GE: <https://www.ge.com/digital/iiot-platform>
- Geest, M., Tekinerdogan, B., & Catal, C. (2021). Design of a reference architecture for developing smart warehouses in Industry 4.0. *Computers in Industry*, 1-21. doi:https://doi.org/10.1016/j.compind.2020.103343
- Ghobakhloo, M. (2018). The future of manufacturing industry: a strategic road toward Industry 4.0. *Journal of Manufacturing Technology Management*, 910-936. doi:https://doi.org/10.1108/JMTM-02-2018-0057
- Google Cloud. (2018). *Web API Design: The Missing Link*. Google LLC.
- Gorton, I., & Klein, J. (2015). *Distribution, Data, Deployment, Software Architecture Convergence in Big Data Systems*. IEEE COMPUTER SOCIETY, 78-85.
- Groover, M. (2001). *Automation, Production Systems, and Computer-Integrated Manufacturing*. USA: Prentice Hall.
- Hermann, M., Otto, B., & Pentek, T. (2015). *Design Principles for Industrie 4.0 Scenarios: A Literature Review*. ResearchGate, 1-16. doi:10.13140/RG.2.2.29269.22248
- Hohpe, G., & Woolf, B. (2004). *Enterprise Integration Patterns*. Boston, MA: Addison-Wesley.
- Huang, M.-L., & Chuang, T. (2020). A database of eight common tomato pest images. *Mendeley Data*, V1. doi:10.17632/s62zm6djd.1
- ISA. (Octubre de 2019). *Automation IT: RAMI 4.0 Reference Architectural Model for Industrie 4.0*. Obtenido de International Society of Automation (ISA): <https://www.isa.org/intech/20190405/>
- ISO/IEC/IEEE 42010. (10 de Jul de 2007). *ISO/IEC/IEEE 42010: Defining "architecture"*. Obtenido de ISO/IEC/IEEE 42010: <http://www.iso-architecture.org/ieee-1471/defining-architecture.html>

- Jocher, G., Stoken, A., Chaurasia, A., Borovec, J., NanoCode012, TaoXie, . . . AlexWang1900. (2021). ultralytics/yolov5: v6.0. Zenodo. doi:https://doi.org/10.5281/zenodo.5563715
- Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 36-52. doi:https://doi.org/10.1016/j.cirpj.2020.02.002
- Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group. National Academy of Science and Engineering (acatech.), 1-82.
- Kakani, V., Nguyen, V., Kumar, B., Kim, H., & Pasupuleti, V. (2020). A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 1-12. doi:doi.org/10.1016/j.jafr.2020.100033
- Karatas, M., Eriskin, L., Deveci, M., Pamucar, D., & Garg, H. (2022). Big D3333ata for Healthcare Industry 4.0: Applications, challenges and future perspectives. *Expert Systems with Applications*, 1-13. doi:https://doi.org/10.1016/j.eswa.2022.116912
- Kusiak, A. (2017). Smart manufacturing. *International Journal of Production Research*, 508-517. doi:https://doi.org/10.1080/00207543.2017.1351644
- Lee, J., Ardakani, H., Yang, S., & Bagheri, B. (2015). Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia CIRP*, 3-7.
- Lie, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods*, 1-18. doi:https://doi.org/10.1186/s13007-021-00722-9
- Liu, C., Vengayil, H., Lu, Y., & Xu, X. (2019). A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect. *Journal of Manufacturing Systems*, 1-14. doi:https://doi.org/10.1016/j.jmsy.2019.04.006
- Liu, Y., Peng, Y., Wang, B., Yao, S., & Liu, Z. (2017). Review on Cyber-physical Systems. *Journal of Automatica Sinica*, 27-40. doi:10.1109/JAS.2017.7510349
- Liu, Z., Sampaio, P., Pishchulov, G., Mehandjiev, N., Cisneros-Cabrera, S., Schirrmann, A., . . . Bnouhanna, N. (2022). The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration. *Computers in Industry*, 1-12. doi:https://doi.org/10.1016/j.compind.2022.103623
- López Martínez, P., Dintén, R., Drake, J., & Zorrilla, M. (2021). A big data-centric architecture metamodel for Industry 4.0. *Future Generation Computer Systems*, 263-284. doi:https://doi.org/10.1016/j.future.2021.06.020
- Lu, Y., Liu, C., Wang, K.-K., Huang, H., & Xu, X. (2019). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer Integrated Manufacturing*, 1-14. doi:https://doi.org/10.1016/j.rcim.2019.101837
- Macías, A., Navarro, E., & González, P. (2019). A Microservice-Based Framework for Developing Internet of Things and People Applications. *Proceedings*, 1-13. doi:10.3390/proceedings2019031085
- Malathi, V., & Gopinath, M. (2021). Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica, Section B — Soil & Plant Science*. doi:10.1080/09064710.2021.1874045
- Miny, T., Stephan, G., Usländer, T., & Vialkowitz, J. (Abril de 2021). Plattform Industrie 4.0. Obtenido de Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Functional-View.html
- Mishra, A. (2019). *Machine Learning in the AWS Cloud*. Indianapolis, Indiana: John Wiley & Sons, Inc.
- Nakagawa, E. Y., Antonino, P. O., Schnicke, F., Capilla, R., Kuhn, T., & Liggesmeyer, P. (2021). Industry 4.0 reference architectures: State of the art and future trends. *Computers & Industrial Engineering*, 1-13. doi:https://doi.org/10.1016/j.cie.2021.107241
- Niknejad, N., Ismail, W., Ghani, I., Nazari, B., Bahari, M., & Hussin, A. (2020). Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation. *Information Systems*, 1-27. doi:https://doi.org/10.1016/j.is.2020.101491
- NIST. (16 de Abril de 2018). Framework for Improving Critical Infrastructure Cybersecurity. Obtenido de National Institute of Standards and Technology: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
- Pallathadka, H., Sajja, G., Phasinam, K., Ritonga, M., Naved, M., Bansal, R., & Quiñonez-Choquecota, J. (2021). An investigation of various applications and related challenges in cloud computing. *Materials Today: Proceedings*, 1-5. doi:https://doi.org/10.1016/j.matpr.2021.11.383
- Pereira, A., & Romero, F. (2017). A review of the meaning and the implications of the Industry 4.0 concept. En P. Manufacturing (Ed.), *Manufacturing Engineering Society International Conference* (págs. 1206-1214). Vigo, España: Elsevier.
- Poccia, D. (2016). *AWS Lambda in Action*. Manning.
- PwC Middle East. (23 de 10 de 2018). Big investments with big impacts and rapid returns. Obtenido de PwC Middle East : https://www.pwc.com/m1/en/publications/industry-40-survey/big-investments.html
- Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., . . . Nee, A. (2019). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 1-19. doi:https://doi.org/10.1016/j.jmsy.2019.10.001
- R, S., & R, S. (2017). *Data Mining with Big Data. Intelligent Systems and Control (ISCO)* (págs. 246-250). Coimbatore, India: IEEE. doi:10.1109/ISCO.2017.7855990
- RedHat. (2021). ¿Que es una api rest? Obtenido de RedHat: https://www.redhat.com/es/topics/api/what-is-a-rest-api#rest
- Richards, M. (2015). *Software Architecture Patterns*. Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc.
- Rosen, D. (2019). Thoughts on Design for Intelligent Manufacturing. *Engineering*, 1-6. doi:https://doi.org/10.1016/j.eng.2019.07.011
- Sahba, R., Radfar, R., Ghatari, A. R., & Ebrahimi, A. P. (2021). Development of Industry 4.0 predictive maintenance architecture for broadcasting chain. *Advanced Engineering Informatics*, 1-11. doi:https://doi.org/10.1016/j.aei.2021.101324
- Singh, D., Jain, N., Jain, P., & Kayal, P. (2019). PlantDoc: A Dataset for Visual Plant Disease Detection. *arXivLabs*, 1-5. doi:arXiv:1911.10317
- Software Engineering Institute. (04 de May de 2018). Attribute-Driven Design - Create software architectures using architecturally significant requirements. Obtenido de Software Engineering Institute at Carnegie Mellon University: https://resources.sei.cmu.edu/asset\_files/FactSheet/2018\_010\_001\_5139\_30.pdf
- Sony, M., Antony, J., Mc Dermott, O., & Garza-Reyes, J. (2021). An empirical examination of benefits, challenges, and critical success factors of industry 4.0 in manufacturing and service sector. *Technology in Society*, 1-12. doi:https://doi.org/10.1016/j.techsoc.2021.101754
- Tao, F., Qi, Q., Wang, L., & Nee, A. (2019). Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, 653-661. doi:https://doi.org/10.1016/j.eng.2019.01.014
- The Apache Software Foundation. (2020). *Apache Avro*. Obtenido de Apache Avro: https://avro.apache.org/
- Tian, W., & Zhao, Y. (2015). Optimized Cloud Resource Management and Scheduling. *Morgan Kaufmann*. doi:https://doi.org/10.1016/C2013-0-13415-0
- Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. *Journal of Manufacturing Systems*, 93-106. doi:https://doi.org/10.1016/j.jmsy.2018.04.007
- Wang, X., Ong, S., & Nee, A. (2017). A comprehensive survey of ubiquitous manufacturing research. *International Journal of Production Research*, 604-628. doi:https://doi.org/10.1080/00207543.2017.1413259
- Wankhede, V. A., & Vinodh, S. (2021). Analysis of Industry 4.0 challenges using best worst method: A case study. *Computers & Industrial Engineering*, 1-13. doi:https://doi.org/10.1016/j.cie.2021.107487
- Wiesner, S., & Thoben, K.-D. (2016). Requirements for models, methods and tools supporting servitisation of products in manufacturing service ecosystems. *International Journal of Computer Integrated Manufacturing*, 1-12. doi:http://dx.doi.org/10.1080/0951192X.2015.1130243
- Xu, L. D., & Duan, L. (2018). Big data for cyber physical systems in industry 4.0: a survey. *Enterprise Information Systems*, 1-23. doi:10.1080/17517575.2018.1442934
- Xu, L., Xu, E., & Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, 56, 2941-2962. doi:https://doi.org/10.1080/00207543.2018.1444806
- Yang, H., Kumara, S., Bukkapatnam, S., & Tsung, F. (2019). The Internet of Things for Smart Manufacturing: A Review. *IIE Transactions*, 1-36. doi:10.1080/24725854.2018.1555383
- Zhong, R., Xu, X., Klotz, E., & Newman, S. (2017). Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering*, 616-630. doi:http://dx.doi.org/10.1016/J.ENG.2017.05.015