

Application of Industrial Agents for Availability and OEE Improvement in Cyber-Physical Production Systems

Juan Pablo Rueda Loaiza¹; Anderson Steven Ortiz²; Luis Alberto Cruz Salazar³; David Santiago Delgadillo Leguizamón⁴; Libis Valdez Cervantes⁵; Juan Sebastián Sánchez-Gómez⁶; Juan Manuel Cuéllar Mendoza⁷
^{1,2,3,4}Universidad Antonio Nariño, Colombia, jrueda58@uan.edu.co, aortiz39@uan.edu.co, luicruz@uan.edu.co, ddelgadillo142@uan.edu.co
⁵Fundación Universitaria Antonio de Arévalo (Unitecnar), Colombia; LACCEI, USA, libis.valdez@unitecnar.edu.co
⁶Universidad El Bosque, Colombia, jusesago@gmail.com
⁷Intenalco Educación Superior, Colombia, redeslab@intenalco.edu.co

Abstract— *Unplanned downtimes are a key cause of loss of availability and OEE in industrial systems. This article develops and implements a Cyber-Physical Production System with distributed control based on Multi-Agent Systems, where Resource Agents (RA) interact with the variables of the physical processes and their functionalities. The development of the CPPS is structured through model-based engineering with SysML for the description and analysis of the system, in a way that facilitates conceptual transfer and reusability. The cyber layer is integrated in a non-intrusive way with an industrial controller through the OPC UA protocol, preserving control determinism. Faced with induced faults in sensors (physical layer), the RA detects inconsistencies, classifies criticality and, when the fault is compensable, executes resilience through soft-sensors and override flags. The aforementioned allows temporarily substituting the physical signal, maintaining operational continuity and registering maintenance alerts, improving the availability variable of the OEE. The proposal is validated in an emulated Pick & Place cell and subjected to 40 independent tests, each operating during a batch of 20 continuous production cycles with random perturbations, against a centralized industrial control and with the multi-agent system activated. Considering an ideal performance and quality ($\approx 90\%$), the results demonstrate an availability of at least 70% when the multi-agent system is activated, proving that the proposed CPPS architecture manages to increase the OEE in traditional industrial systems.*

Keywords— *Availability, Industrial Agents, CPPS, Multi-Agent Systems, OEE.*

Aplicación de Agentes Industriales para mejorar la Disponibilidad y el OEE en Sistemas de Producción Ciberfísicos

Juan Pablo Rueda Loaiza¹; Anderson Steven Ortiz²; Luis Alberto Cruz Salazar³; David Santiago Delgadillo Leguizamón⁴; Libis Valdez Cervantes⁵; Juan Sebastián Sánchez-Gómez⁶; Juan Manuel Cuéllar Mendoza⁷

^{1,2,3,4}Universidad Antonio Nariño, Colombia, jrueda58@uan.edu.co, aortiz39@uan.edu.co, luicruz@uan.edu.co, ddegadillo142@uan.edu.co

⁵Fundación Universitaria Antonio de Arévalo (Unitecnar), Colombia, libis.valdez@unitecnar.edu.co

⁶Universidad El Bosque, Colombia, jusesago@gmail.com

⁷Intenalco Educación Superior, Colombia, redeslab@intenalco.edu.co

Resumen – Las paradas no planificadas son una causa clave de pérdida de disponibilidad y de OEE en sistemas industriales. Este artículo desarrolla e implementa un Sistema de Producción Ciberfísico con control distribuido basado en Sistemas Multiagente, donde Agentes Recurso (RA) interactúan con las variables de los procesos físicos y sus funcionalidades. El desarrollo del CPPS se estructura mediante ingeniería basada en modelos con SysML para la descripción y el análisis del sistema, de manera que se facilite la transferencia conceptual y la reusabilidad. La capa ciber se integra de forma no intrusiva con un controlador industrial mediante protocolo OPC UA, preservando el determinismo del control. Ante fallas inducidas en sensores (capa física), el RA detecta inconsistencias, clasifica criticidad y cuando la falla es compensable, ejecuta resiliencia mediante sensores virtuales (soft-sensors) y marcas de anulación (override flags). Lo anterior permite sustituir temporalmente la señal física, manteniendo la continuidad operativa y registrando alertas de mantenimiento, mejorando la variable de disponibilidad del OEE. La propuesta se valida en una celda Pick & Place emulada y se somete a 40 pruebas independientes, operando cada una durante un lote de 20 ciclos de producción continua con perturbaciones aleatorias, frente a un control industrial centralizado y con el sistema multiagente activado. Considerando un rendimiento y calidad ideales (≈90%), los resultados demuestran una disponibilidad de al menos 70% cuando el sistema multiagente está activado, comprobando que la arquitectura CPPS propuesta logra aumentar el OEE en sistemas industriales tradicionales.

Palabras clave – Agentes Industriales, CPPS, Disponibilidad, Sistemas Multiagente, OEE.

I. INTRODUCCIÓN

Las paradas no planificadas continúan siendo una de las principales causas de pérdida de la capacidad del sistema para mantenerse operativo sin fallos durante la mayor parte del tiempo [1]. Estas interrupciones afectan directamente la continuidad del proceso productivo, degradando el flujo de producción, la estabilidad operativa y el desempeño global del sistema [2], [3]. En este contexto, la disponibilidad de los activos industriales se consolida como un indicador clave, dado su impacto directo sobre la métrica de desempeño *Overall Equipment Effectiveness* (OEE), la cual representa el porcentaje de tiempo de fabricación que es realmente productivo [4]. El OEE es un indicador clave de desempeño (KPI) ampliamente adoptado en la industria, que integra desempeño, calidad y disponibilidad, permitiendo identificar pérdidas operativas, comparar el progreso y orientar estrategias de mejora [5]. De manera particular, la disponibilidad está estrechamente relacionada con el *Mean Time Between Failures* (MTBF), que

permite cuantificar la probabilidad de que un sistema opere sin fallos durante un periodo determinado, manteniendo la continuidad frente a interrupciones no planificadas [6]. A pesar de la relevancia de estas métricas, los sistemas de producción industrial continúan apoyándose mayoritariamente en arquitecturas de control centralizadas basadas en controladores lógicos programables (en inglés, PLC). Si bien estas arquitecturas han demostrado confiabilidad en entornos estables, presentan limitaciones significativas ante escenarios dinámicos, fallas no planificadas y cambios en las condiciones del proceso [1]. Entre estos factores, la disponibilidad refleja el porcentaje de tiempo en que el equipo se encuentra realmente operativo respecto al tiempo planificado, y es directamente influenciada por la relación entre el MTBF y el *Mean Time To Repair* (MTTR), según (1):

$$\text{Disponibilidad (\%)} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} * 100 \quad (1)$$

En este contexto, la evolución temporal de la disponibilidad de un activo industrial pone de manifiesto la necesidad de incorporar capacidades de predicción, autonomía y toma de decisiones distribuidas para optimizar las estrategias de mantenimiento y asegurar la continuidad de un sistema [4]. En la Fig. 1, la dependencia de un único punto de control incrementa el impacto de las averías sobre KPI como la disponibilidad, así el MTTR en consecuencia, afecta el OEE [6].

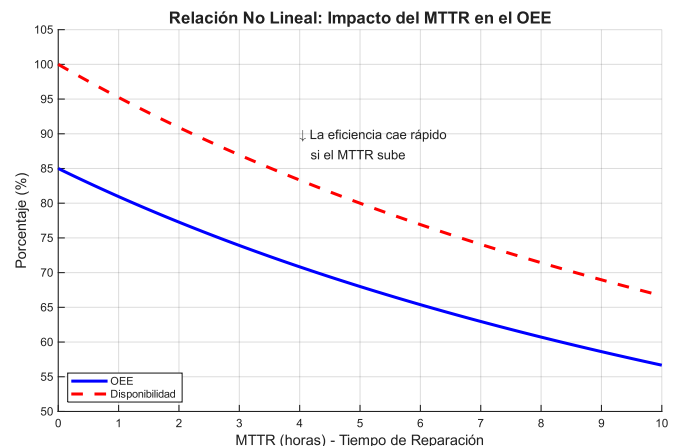


Fig. 1 Relación entre el MTTR y la OEE. Valores simulados con un MTTR constante de 1 h, rendimiento de 0.85 y calidad de 0.95

Con el establecimiento de la Industria 4.0 [7], se ha impulsado la evolución hacia Sistemas de Producción Ciberfísicos (en

inglés, CPPS) integrando digitalización, interconectividad y capacidades de supervisión avanzada [8]. Un CPPS actúa como enlace entre el mundo físico y el virtual, permitiendo la representación en tiempo real de los activos industriales y facilitando la gestión de eventos operativos [9].

Sin embargo, en numerosas celdas industriales y entornos educativos, estas capacidades se implementan aún sobre esquemas de control centralizado basados en PLC, limitando el potencial de adaptación y resiliencia del sistema productivo [10]. Como muestra la Fig. 2, el OEE es un KPI que mide la eficiencia total de un sistema de manufactura [5], y se calcula como el producto de tres factores fundamentales como en (2).

$$OEE = Disponibilidad \times Rendimiento \times Calidad \quad (2)$$



Fig. 2. Diagrama de cálculo del OEE: disponibilidad, rendimiento y calidad.

La disponibilidad, es la capacidad de un sistema para operar sin fallos la mayor parte del tiempo, es uno de los factores de desempeño más importantes en los sistemas de producción automatizados debido a que afecta directamente el rendimiento y, por lo tanto, la productividad del sistema [3].

El rendimiento mide la velocidad real de operación del equipo respecto a su capacidad máxima teórica, por lo que sus pérdidas se asocian comúnmente a micro paradas, ralentizaciones del proceso y tiempos de funcionamiento en vacío [11]. Por su parte, la calidad corresponde al porcentaje de unidades que cumplen con las condiciones críticas de calidad (CTQ) frente al total producido, viéndose afectada por productos defectuosos, reprocesos y desperdicios generados durante la operación. En el contexto del OEE, ambos factores dependen directamente de la disponibilidad, puesto que una detención del sistema impide la producción y reduce simultáneamente la velocidad efectiva y la generación de unidades conformes. En consecuencia, la disponibilidad se establece como la variable crítica del modelo, debido a su impacto multiplicador sobre la eficiencia global del equipo y la productividad final de la estación [12].

Para abordar estas limitaciones, los sistemas multiagente (en inglés, MAS) han sido propuestos como un paradigma para el control industrial distribuido y el desarrollo de CPPS [9], [13], [14], permitiendo modelar entidades autónomas y cooperativas

capaces de tomar decisiones locales y coordinarse para alcanzar objetivos globales del sistema en arquitecturas basadas en MAS [15], [16]. No obstante, persiste una brecha de investigación relevante: la ausencia de enfoques estandarizados que integren arquitecturas de control basadas en sistemas MAS con métricas industriales consolidadas, como el OEE, particularmente en su componente de disponibilidad, que tiene repercusiones en la eficiencia y calidad de la estación.

II. ANTECEDENTES Y REVISIÓN DE LA LITERATURA

A. Categorización de agentes industriales en CPPS

Los agentes industriales se analizan comúnmente a partir de sus clases y tipología, que dependen de sus características funcionales de inteligencia artificial (IA) industrial, arquitectura, y tiempos de respuesta que permiten comprender su rol dentro de un MAS y el comportamiento global de un CPPS [10]. Un agente industrial es una entidad encapsulada (hardware/software) con objetivos específicos definidos [17]. Un agente industrial procura alcanzar dichos objetivos mediante su comportamiento autónomo, interactuando con su entorno y con otros agentes (definición traducida del inglés según la norma VDI-Standard 2653) [18].

A.1 Clases de agentes industriales en CPPS

La literatura en CPPS con MAS propone clases según su rol dentro de la arquitectura [8]. Esta diferenciación permite estructurar responsabilidades y delimitar claramente la interacción entre el dominio físico y el dominio ciber [10].

- **Clase I: Agentes de Acceso Físico**, encargados de abstraer y conectar equipos de producción heterogéneos con el MAS, actuando como representación digital de activos físicos. En esta categoría se instancian comúnmente los Agentes Recurso (RA), que encapsulan funcionalidades y estados operativos de los recursos industriales [18].
- **Clase II: Agentes Organizacionales**, orientados a ofrecer servicios integrados para la gestión y coordinación del sistema productivo, incluyendo la planificación, programación y supervisión de órdenes y procesos [18], [19].
- **Clase III: Agentes de Interfaz**, responsables de facilitar la comunicación efectiva entre agentes y sistemas, convirtiendo interfaces propietarias en múltiples protocolos y garantizando la interoperabilidad dentro del CPPS [6].
- **Clase IV: Agentes Humanos**, que permiten integrar al operador dentro del sistema MAS bajo el concepto *human-in-the-loop*, apoyando tareas de supervisión, validación y asistencia en la toma de decisiones [14].

Entre los agentes de Clase I, los RA adquieren un papel fundamental al encapsular activos físicos y representar su estado operativo dentro del sistema. En consecuencia, la Tabla I formaliza los niveles de responsabilidad y autoridad asociados específicamente al RA, estableciendo un marco jerarquizado que define sus obligaciones, capacidades y restricciones operativas dentro del MAS [10].

TABLA I. NIVELES DE CUMPLIMIENTO Y COMPORTAMIENTO DEL RA EN SISTEMAS CPPS CON MAS. TOMADO DE [10], [20].

Nivel	Etiqueta	Definición Técnica	Función del RA en sistemas MAS
Nivel 5	Debe	Requisito obligatorio estricto; no se permiten desviaciones.	Sincronización determinista: El RA debe mantener la coherencia de datos con el PLC en tiempo real para evitar inconsistencias críticas. El agente debe suprimir la escritura si la señal de activación es baja.
Nivel 4	Debería	Curso de acción recomendado como particularmente adecuado.	Diagnóstico proactivo: El RA debería analizar tendencias en variables OPC UA para anticipar fallos Notificación Jerárquica: El RA debería reportar eventos al AMS y HMI para facilitar la trazabilidad.
Nivel 3	Puede	Acción permisible dentro de los límites del estándar.	Inyección de <i>soft-sensors</i> : El sistema permite la sustitución de datos de sensores físicos mediante el uso de RA en sensores físicos con valores virtuales para mantener la disponibilidad operativa.
Nivel 2	Es capaz de	Declaración de posibilidad técnica o capacidad causal.	Reconfiguración dinámica: El RA es capaz de adaptar sus parámetros de compensación si la topología física de la celda cambia.
Nivel 1	Generalmente no	Nivel mínimo de logro indica funciones inesperadas.	Control de seguridad crítica: El RA generalmente no ejecuta funciones de seguridad funcional (Safety), las cuales permanecen en el PLC por normativa.

A.2 Características fundamentales de los agentes industriales

Desde la perspectiva de la *Foundation for Intelligent Physical Agents* (FIPA), un agente industrial debe presentar un conjunto de características esenciales que definen su comportamiento dentro de un sistema MAS [19]. Estas características son patrones que han sido adaptados al dominio industrial por estándares como VDI/VDE 2653-4 y el IEEE Std 2660.1-2020, los cuales enfatizan la necesidad de una correspondencia directa entre los agentes y los activos físicos de planta, así como la integración con sistemas de automatización existentes [18].

La Tabla II resume las principales características de IA industrial de los agentes industriales y su definición para distintas clases de agentes dentro de un CPPS [20].

TABLA II. DEFINICIONES DE CARACTERÍSTICAS EN AGENTES INDUSTRIALES

Característica	Definición
C1. Autonomía [12]	Grado en el que un agente industrial puede dominar de forma independiente condiciones inciertas de manera delimitada y automatizada, logrando sus objetivos sistemáticamente; es decir, sin intervención externa o humana.
C2. Reactividad [21]	Grado en el que un agente industrial puede responder a una solicitud para el procesamiento de la información de su entorno (capacidad de respuesta de observación y comunicación en tiempo real).
C3. Proactividad [22]	Grado en el que un agente industrial toma la iniciativa para decidir y procesar información mientras persigue un objetivo (razonamiento para tareas deliberativas).
C4. Predictibilidad [1], [23]	Grado en el que un agente industrial puede predecir (capacidad predictiva [24]) los resultados de acciones futuras dadas las acciones en tareas previas y el autoaprendizaje (a partir de información histórica).
C5. Cooperatividad humana [14]	Grado en el que un agente industrial puede aplicar el concepto de humano en el bucle (<i>Human-in-the-loop</i>).

Estas características constituyen la base funcional para el diseño de arquitecturas MAS en entornos CPPS, permitiendo encapsular recursos físicos, soportar la toma de decisiones distribuida, incorporar capacidades predictivas y facilitar la cooperación entre agentes y operadores humanos [10].

B. Requerimientos técnicos de los CPPS

Para llevar estas capacidades a un CPPS real y cumplir con los estándares de la Industria 4.0 [25], se han definido los requerimientos técnicos que se muestran en la Tabla III.

TABLA III. REQUERIMIENTOS TÉCNICOS PARA SISTEMAS MAS EN CPPS

No. de Requerimiento y Descripción
R1. Control descentralizado. El sistema soporta una arquitectura de control descentralizada que permita la toma de decisiones distribuida [26]
R2. Reusabilidad. El sistema permite la reutilización de agentes y comportamientos en diferentes recursos y configuraciones productivas [8], [27]
R3. Implementación independiente de la plataforma. El sistema se implementa utilizando tecnologías independientes de la plataforma, lo que garantiza interoperabilidad, escalabilidad y evita la dependencia de proveedor [28]
R4. Mejora de indicadores KPI industriales. El sistema incorpora mecanismos orientados a la mejora de los KPI industriales, ej. se busca reducir paradas no planificadas y optimizar el desempeño del sistema productivo [29]

C. Correspondencia de MAS con requisitos CPPS

La digitalización ha transformado profundamente los procesos industriales [30]. Este fenómeno, conocido como Industria 4.0, tiene como objetivo incrementar la flexibilidad, la disponibilidad y la resiliencia de los sistemas de manufactura (R4) [13], [14]. decisiones descentralizadas y los MAS se han consolidado como enfoques clave para habilitar la toma de decisiones descentralizada, la autoorganización y la adaptación dinámica en entornos industriales complejos (R1)[20].

Para operar de forma consistente y estandarizada, los CPPS se apoyan en modelos de referencia y marcos normativos ampliamente aceptados. Entre ellos destaca el RAMI 4.0 (del inglés, *Reference Architecture Model Industrie 4.0*) [31], oficializado en la norma DIN SPEC 91345 [32], así como el estándar internacional ISA 95, cuyo propósito es garantizar que dispositivos, aplicaciones y sistemas heterogéneos puedan interoperar y funcionar de manera conjunta dentro de arquitecturas industriales modernas (R3)[32].

En este esquema, los agentes industriales operan bajo un modelo de control distribuido, alejándose del paradigma centralizado tradicional (R1). Su función principal es encapsular recursos físicos y lógicos del sistema productivo, otorgándoles capacidades de comunicación, cooperación y negociación autónoma, lo cual resulta especialmente relevante en entornos heterogéneos con tecnologías y plataformas de distintos fabricantes (R3) [26]. En esta línea, la norma IEEE Std

2660.1-2020 introduce una arquitectura basada en agentes industriales interoperables, promoviendo la estandarización de interfaces, modelos de información y mecanismos de descubrimiento, gestión y cooperación entre agentes [27]. La norma reporta agentes con roles específicos dentro de los CPPS.

Por ejemplo, los *Agentes Producto* negocian la asignación de recursos para optimizar la ejecución del proceso productivo y el cumplimiento de plazos (R1) (R4) [22]. Los agentes *MeiA* actúan como intermediarios entre los niveles de control, facilitando la cooperación mediante estándares como OPC UA (R3) [33]. Arquitecturas en Agentes Industriales de Fabricación Autoconfigurables (SMIA) se apoyan en estos agentes y en el modelo *Asset Administration Shell* de RAMI 4.0 para estandarizar activos físicos y habilitar su representación como gemelos digitales [34] [35], [36], permitiendo un despliegue ágil en celdas de manufactura inteligente (R2) (R3) [37].

En recursos físicos, la estructura del RA de la Fig. 3. busca garantizar que la arquitectura MAS sea escalable y cumpla con los estándares de interoperabilidad industrial en CPPS (R3).

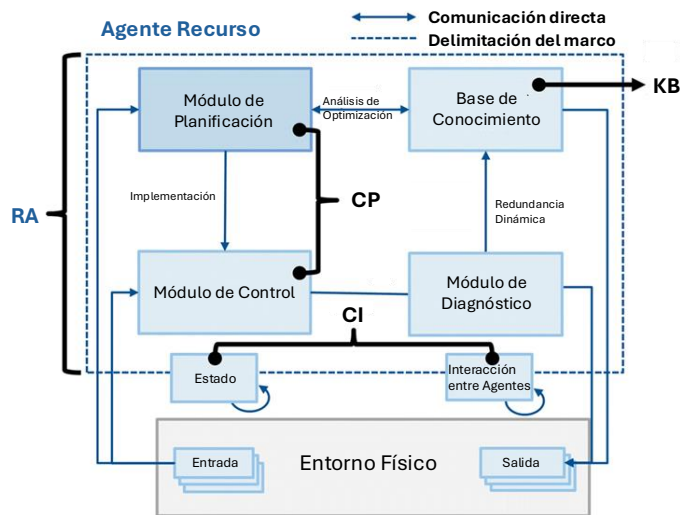


Fig. 3 Arquitectura del agente recurso RA presentada por Wannagat [38].

Desde una perspectiva arquitectónica, el RA presenta la estructura interna (ver Fig. 3), en la cual se integran módulos de planificación, control y diagnóstico soportados por una base de conocimiento (en inglés, *Knowledge Base KB*) (R2). Una interfaz de comunicación (del inglés, *Communication Interface CI*) permite la interacción con otros agentes y con el entorno físico mediante (R3), asimismo incorpora un proceso de coordinación (del inglés, *Coordination Process CP*) (R1), encargado de gestionar la cooperación y sincronización con otras entidades del sistema [38]. En el caso de estudio de este trabajo se aplica una arquitectura MAS basada en RA, alineándose con enfoques experimentales reportados en la literatura de CPPS [10].

D. Plataformas de desarrollo MAS para CPPS

A continuación, se contrastan las plataformas más relevantes de desarrollo de MAS en la industria y se justifica la elección

tecnológica desde una perspectiva de ingeniería de software. La Tabla IV resume la comparativa de plataformas para MAS con Java y Python entre JADE, PADE y SPADE basándose en criterios de arquitectura y desempeño de estándares.

TABLA IV. COMPARATIVA DE PLATAFORMAS MAS

Criterio	JADE[39]	PADE [40]	SPADE [41]
Lenguaje de programación	Java	Python	Python
Cumplimiento extender FIPA	Completo (ACL nativo)	Parcial	Parcial (ACL sobre XMPP)
Modelo de comunicación	Message passing (RMI/sockets)	Message passing TCP	XMPP (Distribuido)
Modelo de concurrencia	Multithreading	Asincrónico	Asincrónico (Asyncio)
Arquitectura de agentes	Agente + comportamientos	Agente + comportamientos	Agente + comp. + FSM*
Soporte distribuido	Sí (Contenedores JADE)	Limitado	Sí (Nativo)
Integración con APIs	Media	Alta	Alta
Escalabilidad práctica	Alta	Baja-media	Alta
Madurez del framework	Alta	Media-baja	Media-alta
Orientación de uso	Académico / formal	Prototipos baja escala	Aplicado / investigación

*FSM: Finite State Machine, máquina finita de estado

JADE: Es una plataforma madura y ampliamente validada que ofrece una implementación exhaustiva del estándar FIPA. Sin embargo, su dependencia de la Máquina Virtual de Java (JVM) y su arquitectura rígida basada en hilos (*multi-threading*) incrementan la complejidad en la integración con sistemas industriales modernos y APIs externas (R3)[39].

PADE: Representa una alternativa liviana en Python que implementa con éxito la representación de mensajes FIPA-ACL. No obstante, su modelo de concurrencia y su baja escalabilidad en redes distribuidas complejas suelen restringir su uso a prototipos académicos de pequeña escala (R1)[40].

SPADE: Combina la versatilidad de Python con un modelo asincrónico basado en *asyncio*. A diferencia de JADE y PADE, utiliza XMPP como middleware de comunicación que permite una gestión nativa de presencia y una escalabilidad distribuida mejorada, ideal para entornos de Industria 4.0 (R1-R3) [41].

En paralelo, el *Industrial Internet of Things* (IIoT) extiende el Internet de las Cosas al ámbito industrial, habilitando la interconexión de máquinas, sensores y sistemas para el análisis de datos en tiempo real (R4). En la práctica, facilita la comunicación mediante protocolos abiertos como MQTT, OPC UA, HTTP REST o LoRaWAN, permitiendo la integración de recursos distribuidos, controladores y servicios digitales. Esta convergencia constituye la base de sistemas interoperables y distribuidos, donde los sistemas MAS actúan como capa de coordinación entre los componentes del sistema (R1) (R3) [42].

III. METODOLOGÍA

Con base en la literatura revisada, se propone una metodología experimental para implementar y evaluar un

sistema CPPS basado en un MAS, con énfasis en los RA para reducir el MTTR y mejorar la disponibilidad en una celda Pick & Place emulada. Esta celda se refiere a una planta en Factory IO denominada *Assembler*¹ que tiene como objetivo principal ensamblar piezas de tapas y bases mediante un sistema de selección y colocación de dos ejes. La metodología en este trabajo se estructura en cuatro fases alineadas con un caso de estudio y su evaluación, como se muestra en la Fig. 4.

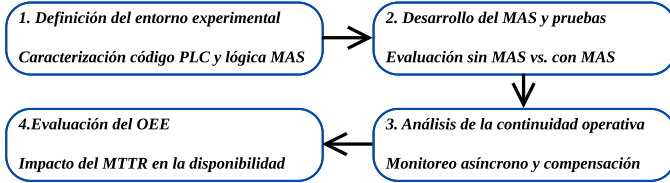


Fig. 4 Las cuatro fases de la metodología propuesta.

En la primera fase se define el entorno experimental, caracterizando el código PLC, su lógica en los sensores críticos y los escenarios de fallo controlado, así como la integración no intrusiva del RA mediante OPC UA. En la segunda fase se busca la reusabilidad (R2) mediante modelos hacia entornos industriales, considerando su relación con OEE y las implicaciones técnicas de implementación. Luego, se implementa el MAS y se ejecutan pruebas bajo fallos inducidos, validando la capacidad de los RAs para detectar inconsistencias y compensar señales sin alterar la lógica determinista del PLC sobre el control de la planta. La tercera fase consiste en el análisis de continuidad operativa y funcionamiento del MAS mediante el monitoreo asíncrono y la compensación que realiza el RA. Finalmente, en la cuarta fase se evalúa la OEE del CPPS y el análisis cuantitativo del impacto del MAS, comparando el escenario base y el escenario con RAs para evaluar la reducción del MTTR y su efecto en la disponibilidad.

En el CPPS del caso de estudio se distribuyen agentes proactivos y reactivos en función del dominio en el que operan, ya sea digital o físico, como se ilustra en la Fig. 5 [20]. Los agentes proactivos se sitúan principalmente en el dominio ciber [34], donde desempeñan funciones de planificación, coordinación y toma de decisiones a nivel global del sistema [10]. En contraste, los agentes reactivos operan en el dominio físico del proceso, estando directamente vinculados a sensores y actuadores, lo que les permite responder de manera inmediata a los eventos que ocurren durante la operación. Se busca la reusabilidad conceptual del modelo CPPS (R2) hacia una estación industrial real tipo Festo² MPS D Pick and Place, evaluada en Factory IO.

Para fundamentar las cuatro fases del caso de estudio, se aplica *Model-Based Systems Engineering*, utilizando SysML como lenguaje formal para modelar de manera estructurada sistemas, incluidos los CPPS basados en MAS [10].

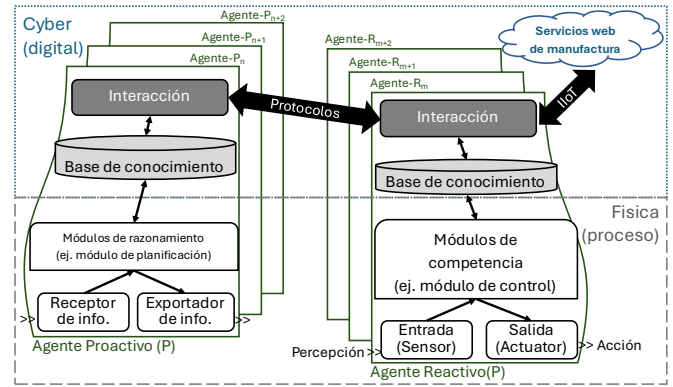


Fig. 5 Interacción entre agentes proactivos y reactivos en un CPPS.

SysML extiende su alcance al modelado integral de sistemas siguiendo un proceso incremental inspirado en el marco *waterfall* [43]. Se desarrollaron así: un diagrama de casos de uso, dos diagramas de secuencias, un *Block Definition Diagram* (BDD), un diagrama paramétrico y un diagrama de actividades. Estos modelos SysML permiten definir requisitos, arquitectura y comportamiento del CPPS de forma coherente y trazable.

IV. CASO DE ESTUDIO: CPPS BASADO EN AGENTE RECURSO

A. Definición del entorno experimental

Se crean modelos SysML para su implementación independientemente de la plataforma (R3). En la planta emulada existen interacciones entre actores humanos, el sistema de control (físico) y la capa inteligente como en la Fig. 6. El modelo delimita el alcance funcional del CPPS, separando entre control, supervisión inteligente y gestión operativa (ciber).

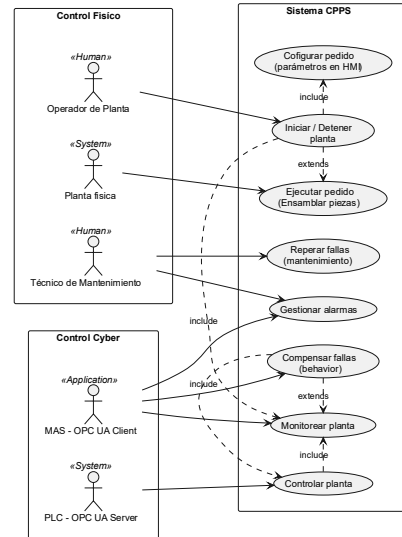


Fig. 6 Diagrama de casos de uso SysML en el CPPS.

La inicialización de la planta se hace mediante un esquema de activación bajo demanda por set point. En el estado inicial, el PLC y el MAS mantienen desacoplamiento operativo hasta la configuración de *setpoints* por el operador. La instanciación del

¹ La planta de *Assembler* de Factory IO está disponible online en: <https://docs.factoryio.com/manual/scenes/assembler/>

² Festo MPS D Pick and Place station disponible online en www.festo.com

MAS y el consumo de ancho de banda difieren hasta el evento *Enable_Agent* como se observa en la Fig. 7. Esta señal dispara una secuencia automática para la comunicación OPC UA, finalizando en un estado de monitoreo pasivo.

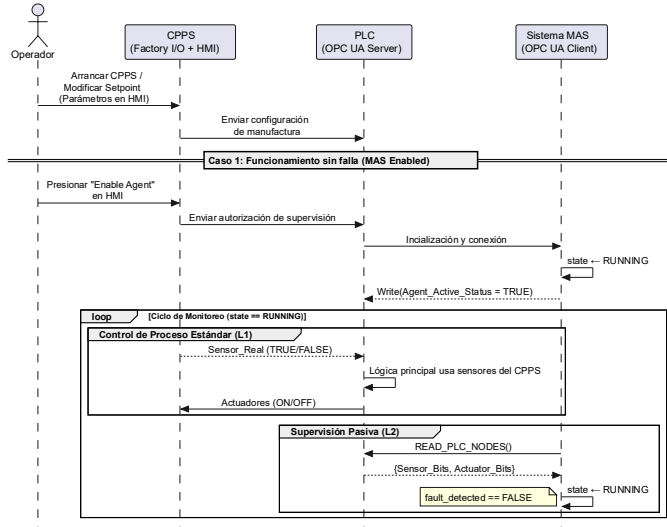


Fig. 7 Diagrama de secuencias: Inicialización Dinámica y Configuración.

La Fig. 8 ilustra la dinámica de resiliencia ante fallos de sensor. Al detectar una anomalía, el MAS transita al estado *COMPENSATING*, ejecutando un *override* lógico mediante la variable *Override_Flag*. Un RA sustituye la señal física por un valor inferido, manteniendo la continuidad operativa. La transición de retorno al control del PLC está condicionada a una validación de estabilidad postcompensación, garantizando la integridad del proceso de manufactura. La compensación ocurre cuando la evaluación del RA corresponde con los casos de compensación viables en la base de conocimiento. El mapeo entre las variables de proceso y la lógica de control evidencia que la capa ciber (inteligente) se acopla de forma no intrusiva a la estructura existente.

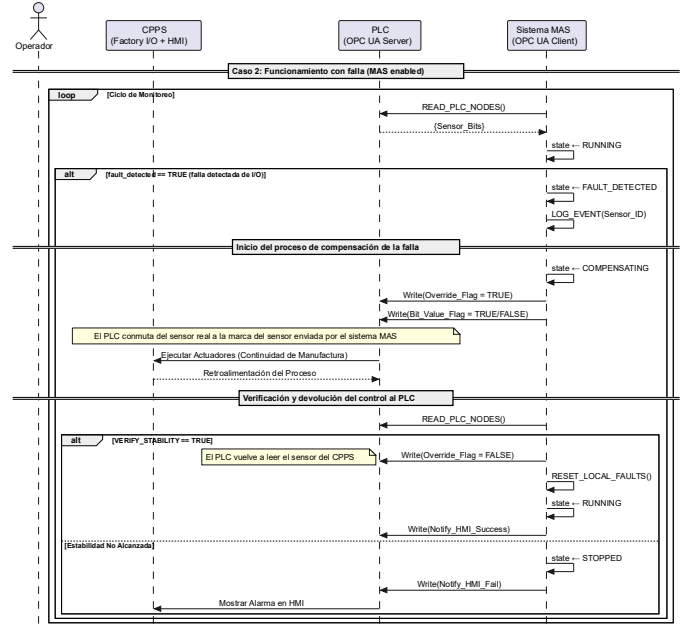


Fig. 8 Diagrama de secuencias: Funcionamiento MAS con falla.

En la Fig. 9 se identifican seis elementos funcionales directamente transferibles a entornos industriales referenciados por números en cada estación y su interfaz de usuario: el control de movimiento del robot *Gantry* en sus ejes X y Z (1, 2), la operación del actuador que atrapa la pieza (3), los sistemas de alimentación de tapas y bases (4, 5) y el módulo de sujeción de tapas y bases (6). La Fig. 10 presenta un diagrama paramétrico SysML que formaliza las dependencias matemáticas entre el MTTR, la disponibilidad operativa y el OEE. Este enfoque permite el aislamiento de variables de pérdida, optimizando la resiliencia del sistema mediante la continuidad del flujo productivo bajo condiciones de falla. Considerando requerimientos de mejora del OEE (R4), se selecciona SPADE para la implementación del MAS por las siguientes razones:

- Arquitectura no bloqueante: El uso de corrutinas asíncronas permite que el RA procese a la par flujos de datos de OPC UA y comportamientos de diagnóstico sin las latencias (R1) o condiciones de carrera típicas de los modelos basados en hilos.

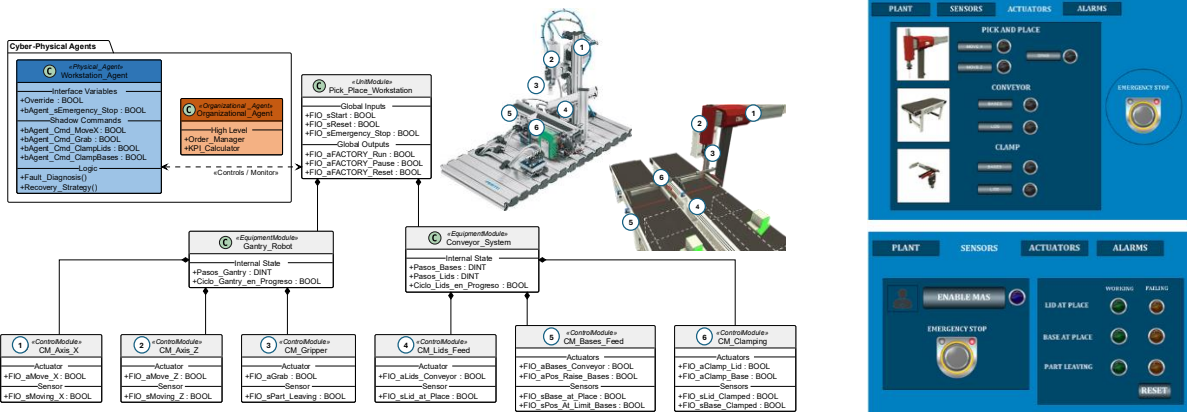


Fig. 9 BDD (izquierda) de la correspondencia estructural entre los bloques del sistema y la estación Pick and Place en Factory IO. Interfaz de usuario (derecha).

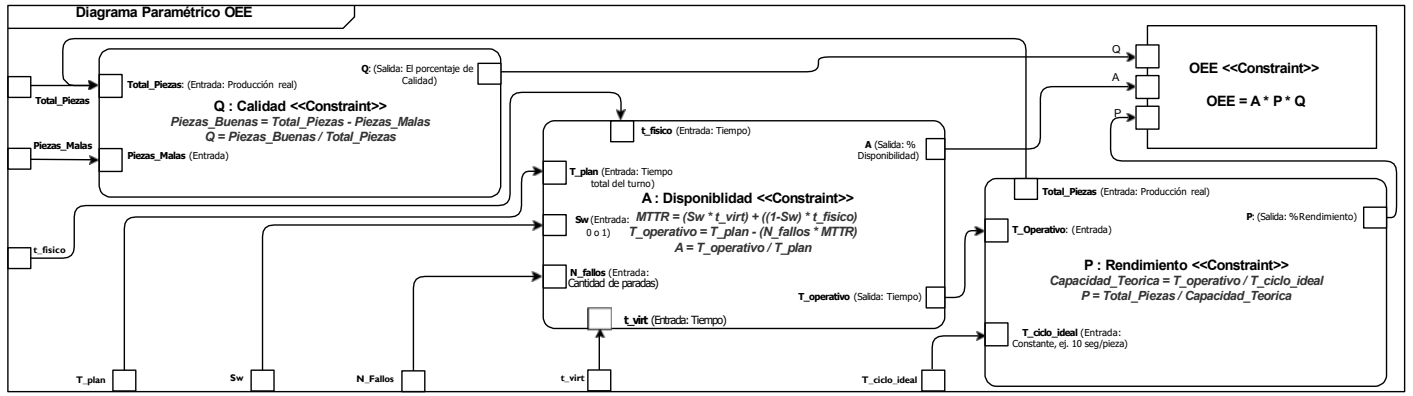


Fig. 10 Diagrama paramétrico SysML del cálculo del OEE.

• Interoperabilidad FIPA-XML: SPADE cumple con los estándares FIPA-ACL encapsulando los mensajes en fragmentos XML. Esto facilita la depuración y asegura que la capa de inteligencia sea independiente de la plataforma de control PLC (R2-R3) y una integración no intrusiva [41].

B. Criterio de ajuste del parámetro de calidad.

En el entorno simulado, el rendimiento del sistema CPPS se fija contra métricas normalizadas de plantas automatizadas reales, alcanzando un 93.37% de calidad en comparación con el 99.99% de estándares de clase mundial en control de calidad industrial [44], [45]. Las fallas se modelan mediante una distribución de probabilidad normal con nivel de significancia del 6,7% ($\alpha=0.067$), simulando eventos realistas en sistemas de producción lineal bajo normas ISO 9001 para gestión de calidad. Este enfoque estadístico presume operación estándar, incorporando herramientas como muestreo por atributos/variables y control de procesos para detectar desviaciones tempranas, similar a estrategias distribuidas en agentes industriales que negocian fallos en celdas flexibles [11].

En producción lineal, las fallas de calidad impactan intrínsecamente el rendimiento a través de la disponibilidad, modulada por MTBF y MTTR, como se evidencia en modelos CPPS que priorizan resiliencia vía agentes proactivos/reactivos [12]. Esta interdependencia justifica la extrapolación del

modelo simulado a entornos PLC reales, reduciendo latencias y residuos mediante interoperabilidad OPC UA/XMPP.

C. Implementación del sistema MAS y ejecución de fallos

Con base en la experimentación realizada, se analizan los eventos de fallo gestionados en el sistema MAS, evaluando el impacto de las estrategias de simulación de respuesta sobre la continuidad operativa del sistema. Siguiendo la metodología de Wannagat y Vogel-Heuser [12], se comparan dos escenarios:

- Escenario base (sin MAS): Un fallo en un sensor detiene inmediatamente la secuencia del PLC, reduciendo la disponibilidad hasta la intervención física de un técnico.
- Escenario propuesto (con MAS): El MAS detecta la inconsistencia mediante el análisis de estados en una FSM asíncrona. Si la falla es compensable, un RA inyecta una señal virtual al PLC, permitiendo la continuidad del proceso y registrando el evento como una alerta de mantenimiento.

D. Análisis de la continuidad operativa

Esta etapa permite identificar las ventajas y restricciones del enfoque distribuido frente a arquitecturas de control centralizado y sienta las bases para trabajos futuros orientados a la incorporación de otras clases de agentes y validaciones experimentales como también la relación de parámetros de cálculo en entornos industriales reales.

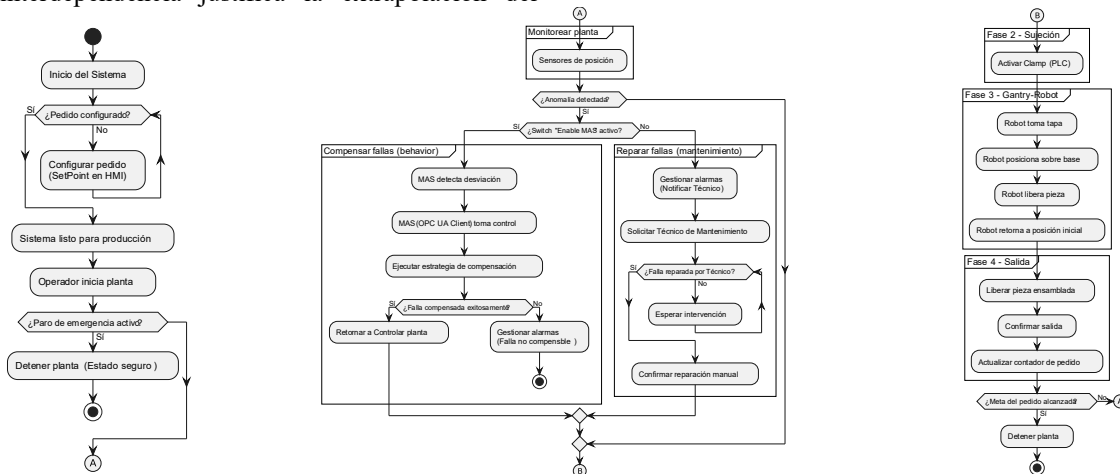


Fig. 11 Diagrama de actividades SysML del CPPS.

De manera complementaria al análisis paramétrico del OEE (ver Fig. 10), y con el fin de vincular el modelo matemático con el comportamiento operacional del sistema, el diagrama de actividades explica el flujo normal. En la Fig. 11 se ven las secuencias que vigilan las variables críticas y sus efectos cuando falla un sensor (*soft-sensor* [10]), asimismo la integración entre la lógica del PLC (*ladder*) y la intervención del sistema MAS en el proceso Pick & Place. Asimismo, la arquitectura de software del CPPS propuesto se fundamenta en código³ híbrido (bajo y alto nivel). El pseudocódigo en la Fig. 12 describe resumidamente el proceso de supervisión y la estrategia de mitigación de errores del MAS mediante señales virtuales (temporales tipo flag) en código Python (alto nivel).

Entrada: *PLC.Nodes* (Sensores/Actuadores), *Sample.Time*
Salida: *Override.Signal*, *HMI.Notification*, *System.State*

```

state ← INIT;
Call Connect_OPC-UA(PLC_Server);
if Agent_Enabled = TRUE then // Esperar habilitación desde HMI
    Write (PLC.Agent.Active_Status ← TRUE);
    state ← RUNNING;
while state ≠ SHUTDOWN do
    // --- Fase de Monitoreo (Ciclo Sincrono) ---
    Sensor_Bits ← Read(PLC.Nodes);
    if ¬Fault_Detected(Sensor_Bits) then
        state ← RUNNING;
    ; // El agente mantiene supervisión pasiva (L2)
    else
        state ← FAULT_DETECTED;
        Call LOG_EVENT (Sensor_ID);
        // --- Inicio de Compensación (Override) ---
        state ← COMPENSATING;
        Corrected_Value ← Get_Strategy(Sensor_Bits);
        Write (PLC.Override_Flag ← TRUE);
        Write (PLC.Bit_Value_Flag ← Corrected_Value);
        ; // El PLC ejecuta lógica con sensor virtualizado
        Wait (Process_Reaction_Time);
        // --- Verificación y Devolución de Control ---
        Post_Action_Data ← Read(PLC.Nodes);
        if VERIFY_STABILITY (Post_Action_Data) is TRUE then
            Write (PLC.Override_Flag ← FALSE);
            Call RESET_LOCAL_FAULTS ();
            Write (PLC.Notify_HMI_Success ← TRUE);
            state ← RUNNING;
        else
            // Fallo en compensación: Parada Segura
            state ← STOPPED;
            Write (PLC.Notify_HMI_Fail ← TRUE);
            break;
        Wait (Sample.Time);

```

Fig. 12 Pseudocódigo de supervisión y compensación del sistema MAS.

E. Evaluación de la disponibilidad

Para el análisis de resultados experimentales se somete el CPPS a 40 pruebas independientes, dado que cada prueba comprende un lote de 20 ciclos permite que la distribución de los promedios de disponibilidad satisfaga los criterios de normalidad asintótica del Teorema del Límite Central ($n \geq 30$) [46], inyectando perturbaciones aleatorias para elevar artificialmente el número de fallos. La Fig. 13 presenta los datos recolectados en tiempo real. Los resultados evidencian la sensibilidad del CPPS ante la acumulación de errores. A medida que el número de fallos incrementa durante la prueba (pasando de 0 a 6 eventos), la disponibilidad desciende progresivamente

del 96.9% al 72.9%, asimismo se observa que las métricas de Rendimiento (~90%) permanecen estable y desacopladas de las fallas.

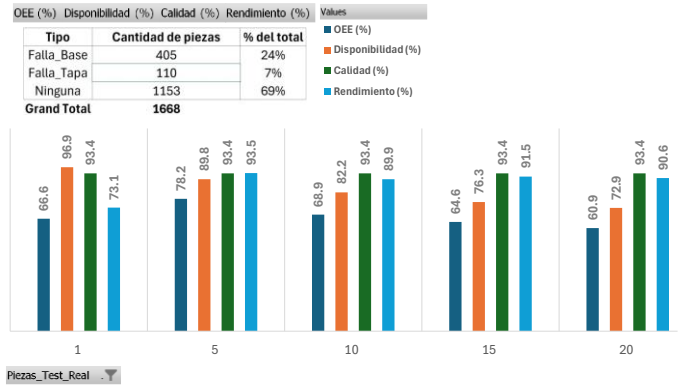


Fig. 13 Evolución del OEE y disponibilidad por número de piezas con MAS.

La medida del rendimiento valida que el sistema mide correctamente la velocidad de procesamiento cuando la máquina está en estado RUN. Esta caída confirma la ecuación paramétrica donde el sistema penaliza el OEE estrictamente en función de la cantidad de interrupciones y su duración, sin afectar falsamente a los otros indicadores (ver Fig. 10).

Para evaluar el impacto del enfoque propuesto bajo condiciones de estrés (inyección controlada de 1 falla por cada intervalo de 20 piezas producidas), se analizó la distribución del OEE. Dado que los datos registrados presentan una distribución fuertemente asimétrica (concentraciones atípicas en los extremos del 0% y valores cercanos al 100%), se comprobó que no cumplen con el supuesto de normalidad requerido para pruebas paramétricas. Por consiguiente, se seleccionó la prueba estadística no paramétrica U de Mann-Whitney, la cual es el método adecuado para comparar las medianas y las distribuciones de dos muestras independientes (“Sin MAS” y “Con MAS”) bajo estas condiciones.

El nivel de significancia para determinar diferencias estadísticas se estableció en un alfa de 0.05. El análisis estadístico reveló una diferencia significativa en el rendimiento operativo entre ambas configuraciones ($p < 0.05$). Como se ilustra en la Fig. 14, la distribución de los datos de OEE bajo estrés muestra patrones de comportamiento drásticamente distintos en sus rangos inferiores. El sistema tradicional “Sin MAS” presenta una alta concentración de densidad de probabilidad en el límite inferior absoluto (0%). En contraste, el sistema “Con MAS” reduce de forma notable la densidad en el 0%, desplazando la masa de los datos hacia rangos operativos intermedios (aproximadamente entre el 20% y el 60%). Si bien ambas configuraciones lograron alcanzar picos máximos de eficiencia cercanos al 100%, la respuesta del sistema ante las fallas en el cuartil inferior constituyó el diferenciador principal.

Los resultados obtenidos en la emulación demuestran que la lógica de los agentes es robusta. Esta validación controlada es un paso previo esencial que garantiza la efectividad de los

³ Código disponible en: <https://github.com/siulzurc/agent4.0/>

sensores virtuales (soft-sensors) ante fallas operativas, previo a una implementación en planta física.

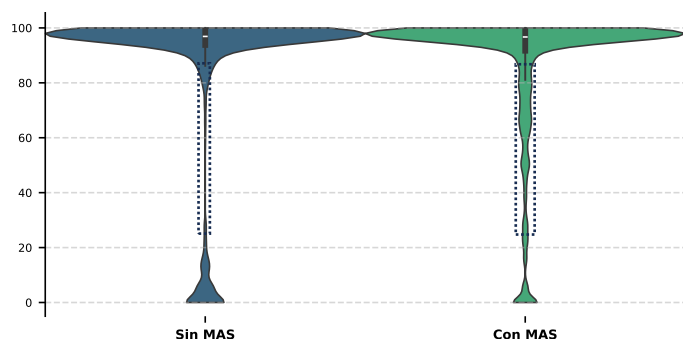


Fig. 14 Prueba estadística del OEE global con o sin MAS.

V. CONCLUSIONES

Este trabajo demuestra que una arquitectura CPPS basada en MAS con RAs transforma la disponibilidad del sistema desde un esquema reactivo y centralizado hacia un comportamiento resiliente y distribuido frente a fallas no planificadas. En este caso de estudio, una celda Pick & Place se integra el MAS al CPPS—no intrusiva al código del PLC—se logra mediante OPC UA. RAs permiten detectar inconsistencias en sensores, clasificar su criticidad e inyectar señales virtuales cuando la falla es compensable, manteniendo la continuidad operativa sin alterar el determinismo del PLC. El MAS evalúa y desplaza el efecto de una falla desde una parada de línea hacia un estado de operación degradada controlada, reduce el impacto de las paradas no planificadas sobre el MTTR y eleva la disponibilidad, y con ello el componente de disponibilidad del OEE. Asimismo, el RA en su base de conocimiento define con claridad qué decisiones pueden descentralizarse de forma segura y cuáles deben permanecer en el PLC (ej. compensar variables), fortaleciendo la trazabilidad técnica del enfoque.

A pesar de la agresividad de las pruebas, donde se fuerza una tasa de fallo aproximada del 30 % (6 fallos en 20 ciclos), el sistema demuestra una notable resiliencia operativa. El hecho de que la disponibilidad final se mantenga por encima del 70 % indica que, aunque el número de fallos es alto, el MTTR permanece bajo. Esto implica que la arquitectura CPPS permite una detección rápida y una recuperación eficiente de los estados de error. El modelo paramétrico OEE y demás modelos SysML se consolida como una herramienta eficaz no solo para el cálculo de indicadores, sino también para caracterizar la naturaleza de las pérdidas: en este escenario, las pérdidas de OEE se atribuyen exclusivamente a la frecuencia de paradas, descartando problemas de velocidad o calidad en la línea.

No obstante, los resultados están condicionados por varias limitaciones. La validación se realiza en un entorno de laboratorio emulado con fallas inducidas principalmente en sensores, por lo que su generalización a escenarios industriales con fallas multicausales requiere más evidencias. La mejora observada se interpreta a partir de la reducción técnica del MTTR, pero el MTBF debe analizarse con cautela para mejorar la compensación. Además, la escritura controlada en el PLC

con variables *override* introduce desafíos asociados a seguridad funcional, ciberseguridad, certificación y gobernanza, lo que exige análisis formales de riesgo, evaluación de latencias y criterios estrictos para determinar cuándo una falla es compensable sin afectar la calidad ni la seguridad.

Como trabajo futuro, se propone (i) completar la validación cuantitativa con campañas experimentales repetibles que reporten distribuciones de tiempos de detección, respuesta, MTTR, disponibilidad y OEE (no solo promedios), e incorporar pruebas de significancia o análisis por escenarios; (ii) ampliar el repertorio de fallas hacia eventos de mayor realismo industrial, incluyendo degradación progresiva, fallas intermitentes, pérdidas de comunicación y fallas en actuadores, integrando criterios de “degradación segura” que consideren calidad y desempeño; (iii) extender el CPPS desde RAs “unitarios” hacia un ecosistema MAS con otras clases de agentes con negociación/redistribución de funciones entre recursos disponibles, evaluando escalabilidad (número de agentes, carga de red, latencia) y coordinación; (iv) incorporar un gemelo digital o modelo SysML más completo para cerrar el lazo entre eventos, políticas de compensación y KPI, habilitando optimización basada en datos; y (v) avanzar hacia pilotos en entornos industriales controlados, la escalabilidad de esta arquitectura hacia una planta real, lo cual permitirá evaluar el desempeño de los agentes ante variables físicas no deterministas. Esta validación en entornos de producción a gran escala será fundamental para consolidar la viabilidad técnica y económica del CPPS propuesto en la Industria 4.0.

AGRADECIMIENTOS

Este trabajo ha sido apoyado por la Universidad Antonio Nariño, la Universidad El Bosque y la UNITECNAR. Los autores expresan su gratitud a sus familias y profesores, extendiendo un reconocimiento especial a nuestro profesor, Luis Cruz. Asimismo, agradecemos a Katherine Ramos, cuyo apoyo incondicional permitió el enfoque y el arduo trabajo necesarios para culminar este proyecto.

† Este trabajo está dedicado a la memoria de un joven y prometedor investigador, el Profesor **Luis Fernando Castañeda Melo, PhD**.

REFERENCIAS

- [1] G. Eads *et al.*, “Predictability, Repeatability, and Models for Cyber-Physical Systems,” in *Foundations and Applications of Component-based Design (WFCDD) ArtistDesign Workshop organized within Embedded Systems Week*, Scottsdale, AZ, USA, 2010, pp. 1–8.
- [2] J. Mund, M. Junker, S. Bougouffa, S. Cha, and B. Vogel-Heuser, “Model-based availability analysis for automated production systems,” in *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design*, New York, NY, USA: ACM, Sep. 2017, pp. 46–55. doi: 10.1145/3127041.3127051.
- [3] A. Immonen and E. Niemelä, “Survey of reliability and availability prediction methods from the viewpoint of software architecture,” *Softw. Syst. Model.*, vol. 7, no. 1, pp. 49–65, Nov. 2007, doi: 10.1007/s10270-006-0040-x.
- [4] J. Berganzo, “OEE: La guía completa y definitiva sobre el indicador industrial por excelencia,” ZEO Technology. Accessed: Nov. 07, 2025. [Online]. Available: <https://www.sistemasoe.com/oee-guia-completa/>
- [5] I. C. Fantozzi, A. Santolamazza, G. Loy, and M. M. Schiraldi, “Digital Twins: Strategic Guide to Utilize Digital Twins to Improve Operational Efficiency in Industry 4.0,” *Future Internet*, vol. 17, no. 1, Jan. 2025, doi: 10.3390/fi17010041.
- [6] Jenna Koo, “MTBF vs MTTF vs MTTR: Comprender las métricas de incidencias en el contexto de las operaciones,” Tulip. Accessed: Nov. 07, 2025. [Online]. Available: <https://tulip.co/es/blog/mtbf-mttf-mttr-incident-metrics/>

- [7] J. S. Sánchez-Gómez, A. Hernandez-Fuentes, and L. Valdez, "The closing of the gender gap in the fourth industrial revolution with a feminist focus in Colombia," in *PAEE ALE 2024*, 2024. doi: 10.5281/zenodo.14062556.
- [8] L. A. Cruz Salazar, D. Ryashentseva, A. Lüder, and B. Vogel-Heuser, "Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns," *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 4005–4034, Dec. 2019, doi: 10.1007/s00170-019-03800-4.
- [9] B. Vogel-Heuser, J. Lee, and P. Leitão, "Agents enabling cyber-physical production systems," *at - Automatisierungstechnik*, vol. 63, no. 10, pp. 777–789, Oct. 2015, doi: 10.1515/auto-2014-1153.
- [10] L. A. Cruz Salazar and B. Vogel-Heuser, "A CPPS-architecture and workflow for bringing agent-based technologies as a form of artificial intelligence into practice," *At-Automatisierungstechnik*, vol. 70, no. 6, pp. 580–598, Jun. 2022, doi: 10.1515/auto-2022-0008.
- [11] X. Ye and S. H. Hong, "An AutomationML/OPC UA-based Industry 4.0 Solution for a Manufacturing System," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Sep. 2018, pp. 543–550. doi: 10.1109/ETFA.2018.8502637.
- [12] S. Karnouskos, L. Ribeiro, P. Leitao, A. Luder, and B. Vogel-Heuser, "Key Directions for Industrial Agent Based Cyber-Physical Production Systems," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, IEEE, May 2019, pp. 17–22. doi: 10.1109/ICPHYS.2019.8780360.
- [13] B. Vogel-Heuser, M. Seitz, L. A. Cruz Salazar, F. Gehlhoff, A. Dogan, and A. Fay, "Multi-agent systems to enable Industry 4.0," *At-Automatisierungstechnik*, vol. 68, no. 6, pp. 445–458, Jun. 2020, doi: 10.1515/auto-2020-0004.
- [14] S. Karnouskos, P. Leitao, L. Ribeiro, and A. W. Colombo, "Industrial Agents as a Key Enabler for Realizing Industrial Cyber-Physical Systems: Multiagent Systems Entering Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 14, no. 3, pp. 18–32, Sep. 2020, doi: 10.1109/MIE.2019.2962225.
- [15] D. S. B. V.-H. S. Ulewicz, "Flexible real time communication between distributed automation software agents," *International Conference on Production*, 2013.
- [16] K. S. Leitão P., *Industrial agents: emerging applications of software agents in industry*, 1st ed. 2015.
- [17] 2660.1-2020 - IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level Automation Functions. IEEE, 2021.
- [18] VDI/VDE, "Multi-agent systems in industrial automation - Selected patterns for field level control and energy systems." Accessed: Jan. 18, 2026. [Online]. Available: <https://www.vdi.de/mitgliedschaft/vdi-richtlinien/details/vdivde-2653-blatt-4-multi-agent-systems-in-industrial-automation-selected-patterns-for-field-level-control-and-energy-systems>
- [19] H. Suguri, "A standardization effort for agent technologies: The Foundation for Intelligent Physical Agents and its activities FIPA," in *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, IEEE Comput. Soc, p. 10. doi: 10.1109/HICSS.1999.773093.
- [20] L. A. Cruz Salazar and B. Vogel-Heuser, "Industrial Artificial Intelligence: A Predictive Agent Concept for Industry 4.0," in *IEEE International Conference on Industrial Informatics (INDIN)*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 27–32. doi: 10.1109/INDIN51773.2022.9976159.
- [21] Stuart J. Russell and Peter Norvig, *Artificial Intelligence A Modern Approach*, Fourth Edition. PEARSON.
- [22] M. Cossentino, S. Lopes, G. Renda, L. Sabatucci, and F. Zaffora, "A Metamodel of a Multi-Paradigm Approach to Smart Cyber-Physical Systems Development," 2019.
- [23] L. Ribeiro and M. Hochwallner, "On the Design Complexity of Cyberphysical Production Systems," *Complexity*, Jan. 2018, doi: 10.1155/2018/4632195.
- [24] A. R. Robinson, P. J. Haley, P. F. J. Lermusiaux, and W. G. Leslie, "Predictive skill, predictive capability and predictability in ocean forecasting," in *Oceans '02 MTS/IEEE*, IEEE, pp. 787–794. doi: 10.1109/OCEANS.2002.1192070.
- [25] J. S. Sánchez-Gómez, C. Cadenas Anaya, V. E. Peters Rada, and S. Milena Rojas Tolosa, "STEM Vocations for Industry 4.0," in *2024 IEEE Colombian Conference on Communications and Computing (COLCOM)*, IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/COLCOM62950.2024.10720278.
- [26] D. Maldonado, E. Cruz, J. Abad Torres, P. J. Cruz, and S. D. P. Gamboa Benitez, "Multi-Agent Systems: A Survey about Its Components, Framework and Workflow," *IEEE Access*, vol. 12, pp. 80950–80975, 2024, doi: 10.1109/ACCESS.2024.3409051.
- [27] 2660.1-2020 - IEEE Recommended Practice for Industrial Agents : Integration of Software Agents and Low-Level Automation Functions. IEEE, 2021.
- [28] L. A. Cruz Salazar, F. Mayer, D. Schütz, and B. Vogel-Heuser, "Platform Independent Multi-Agent System for Robust Networks of Production Systems," Elsevier B.V., Jan. 2018, pp. 1261–1268. doi: 10.1016/j.ifacol.2018.08.359.
- [29] A. Lopez et al., "Technology-Independent Demonstrator for Testing Industry 4.0 Solutions," in *IEEE International Conference on Industrial Informatics (INDIN)*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 21–26. doi: 10.1109/INDIN51773.2022.9976144.
- [30] J. S. Sánchez Gómez, S. M. Chaparro Cajamarca, and A. Godoy Gonzalez, "Agile methodologies for the digital transformation of a Colombian telecommunications company," *Transactions on Energy Systems and Engineering Applications*, May 2025, doi: 10.32397/tesea.vol6.n1.725.
- [31] H. R. Albis et al., "Remote Educational Environment through MOOC for Industrial Automation and RAMI 4.0 Integration using Digital Twins in Colombia," in *2025 IEEE Digital Education and MOOCs Conference (DEMOcon)*, IEEE. doi: 10.1109/DEMOcon65705.2025.11282874.
- [32] DIN and P. Anna, "ReferenceArchitectureModelIndustrie40 RAMI4.0," Apr. 2016, Accessed: Sep. 30, 2025. [Online]. Available: <https://www.din.de/en/wdc-beuth:din21:250940128>
- [33] R. Sierra, I. Sarachaga, A. Burgos, N. Iriondo, and M. L. Álvarez, "Collaboration between industrial agents of encapsulated control components," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 22, no. 1, pp. 81–91, 2025, doi: 10.4995/riai.2024.21953.
- [34] L. A. Cruz Salazar et al., "CyberTwinNet - Colombia: A collaborative network of Remote and Online Labs for Industry 4.0 Education for Latin America," in *Proceedings of the 23rd LACCEI International Multi-Conference for Engineering, Education and Technology (LACCEI 2025)*, Latin American and Caribbean Consortium of Engineering Institutions, 2025.
- [35] J. S. D. Rodríguez et al., "Digital Twin and Cyber-Physical System Architecture for Musical Instruments and Their Accessories," in *2025 IEEE Colombian Caribbean Conference (C3)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/C366505.2025.11340232.
- [36] E. Chacón, L. A. Cruz Salazar, E. M. Cruz, L. Valdez-Cervantes, J. S. Sánchez-Gómez, and M. M. Larrondo-Petrie, "Towards a Digital Twin-Based Evaluation Framework for Cyber-Physical Production Systems: Latin America SMEs Guideline," in *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/ETFA65518.2025.11205706.
- [37] E. Hurtado, A. Burgos, A. Armentia, and O. Casquero, "Self-configurable Manufacturing Industrial Agents (SMIA): a standardized approach for digitizing manufacturing assets," *J. Ind. Inf. Integr.*, vol. 47, Sep. 2025, doi: 10.1016/j.jii.2025.100915.
- [38] A. Wannagat, "Development and evaluation of agent-based automation systems in order to increase the flexibility and reliability of manufacturing plants," Technische Universität München, 2010.
- [39] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE a FIPA2000 compliant agent development environment," in *Proceedings of the fifth international conference on Autonomous agents*, New York, NY, USA: ACM, May 2001, pp. 216–217. doi: 10.1145/375735.376120.
- [40] L. S. Melo, R. F. Sampaio, R. P. S. Leão, G. C. Barroso, and J. R. Bezerra, "Python-based multi-agent platform for application on power grids," *International Transactions on Electrical Energy Systems*, vol. 29, no. 6, Jun. 2019, doi: 10.1002/2050-7038.12012.
- [41] P. javi, "Smart Python Agent Development Environment (SPADE)." Accessed: Feb. 11, 2026. [Online]. Available: <https://github.com/javipalanca/spade>
- [42] S. Gil, G. D. Zapata-Madrigal, R. García-Sierra, and L. A. Cruz Salazar, "Converging IoT protocols for the data integration of automation systems in the electrical industry," *Journal of Electrical Systems and Information Technology*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s43067-022-00043-4.
- [43] B. Vogel-Heuser, "1 Introduction," in *System Engineering with SysML*, De Gruyter, 2025, pp. 1–12. doi: 10.1515/9783111442907-001.
- [44] C. E. Belman-López, J. A. Jiménez-García, J. A. Vázquez-Lopez, and K. A. Camarillo-Gómez, "Diseño de una arquitectura para sistemas y aplicaciones en Industria 4.0 basada en computación en la nube y análisis de datos," *Revista Iberoamericana de Automática e Informática industrial*, vol. 20, no. 2, pp. 137–149, Sep. 2022, doi: 10.4995/riai.2022.17791.
- [45] L. Y. Becerra Sánchez, J. E. Herrera Arroyave, L. H. H. Morris Molina, and A. Toro Lazo, "Tecnologías de la cuarta revolución industrial utilizadas en la manufactura para mejorar los indicadores de productividad: Una revisión," *Entre Ciencia e Ingeniería*, Jul. 2024, doi: 10.31908/19098367.3149.
- [46] S. G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean J. Anesthesiol.*, vol. 70, no. 2, pp. 144–156, Apr. 2017, doi: 10.4097/kjae.2017.70.2.144.