

## MÁSTER UNIVERSITARIO EN INGENIERÍA DE CONTROL, AUTOMÁTICA Y ROBÓTICA

# TRABAJO FIN DE MÁSTER

### **INTEGRACIÓN DE ROBOT COLABORATIVO EN SISTEMA DE FABRICACIÓN FLEXIBLE EN EL CONTEXTO DE LA INDUSTRIA 4.0**

**Estudiante**  
**Directoras**

**Departamento**  
**Curso académico**

*Aragón Isidro, Alazne*  
*Sarachaga González, María Isabel*  
*Armentia Diaz de Tuesta, Aintzane*  
*Ingeniería de Sistema y Automática*  
*2021/2022*

*Bilbao, 24, junio, 2022*



## ***AGRADECIMIENTOS***

---



*Quería comenzar este apartado agradeciendo el trato recibido por todos los profesores durante el máster. También agradecer a mis compañeros por estos dos años, no ha sido nada fácil comenzar un curso después de una pandemia y sin poder vernos la mitad de la cara, pero hemos conseguido pasar buenos momentos quedándonos con anécdotas para recordar.*

*Agradecer también a todos aquellos que han seguido al lado mío a pesar de haberme tenido que desplazar a Bilbao. Este desplazamiento ha conseguido que los fines de semana los disfrutásemos de una mejor forma. Por ello, gracias a esas amigas y amigos que siempre están, aunque no nos veamos todos los días.*

*Agradecer el esfuerzo de mis directoras, Isabel y Aintzane, el cual ha servido para poder sacar adelante este proyecto. Gracias por el trato que me habéis dado y por haber mostrado en todo momento vuestro apoyo ante cualquier situación. Sin olvidarme de Alejandro y Oskar, quienes han estado dispuestos a ayudarme en todo momento.*

*Por último, agradecer a mi familia por haberme dejado adentrarme en esta nueva experiencia, la cual suma una etapa más de mi vida. A pesar de ser unos años duros han servido para madurar como estudiante y también como persona. Gracias por seguir a mi lado aun no poder pasar tanto tiempo juntos.*



## ***RESUMEN***

---



## Resumen

Los sistemas de fabricación convencionales se encuentran inmersos en una transición hacia un modelo de producción más flexible que permita cubrir las demandas actuales del mercado. El paradigma de la Industria 4.0 plantea el uso de herramientas y tecnologías que aumenten la eficiencia y la capacidad de adaptación de los procesos productivos para hacer frente a un mercado globalizado y con una alta competitividad. Para lograr estos objetivos se pueden identificar dos factores clave: la robotización y la descentralización.

Una de las tecnologías determinantes en este nuevo modelo de producción es la robótica colaborativa, ya que aporta fuerza, resistencia y precisión en la realización de tareas, y, sobre todo, capacidad para que personas y robots puedan colaborar en la realización de dichas tareas. Así, el uso de la robótica colaborativa permite maximizar la flexibilidad, la productividad y la calidad de los procesos.

Los sistemas multiagente han demostrado ser adecuados para satisfacer los requisitos de conectividad y descentralización de los activos de fabricación en diferentes entornos industriales, hasta el punto de que el concepto de Agente Industrial se concibe como una implementación del concepto de Componente I4.0 (una particularización del concepto de Sistema Ciberfísico alineado con la arquitectura de referencia RAMI 4.0) mediante agentes.

Este este contexto, este trabajo contribuye a la integración de una estación de ensamblado (formada por un robot colaborativo y un PLC) como agente industrial dentro FlexManSys, una plataforma de gestión de aplicaciones de fabricación desarrollada por el grupo de investigación GCIS de la UPV/EHU. Se sigue una arquitectura en 4 capas que permite la integración de la estación de ensamblado como componente I4.0 dentro de la plataforma inteligente. Esta integración comprende: la identificación y programación de las operaciones de la estación, de forma modular para maximizar la flexibilidad del sistema; la definición de los mecanismos necesarios para el intercambio de datos entre el activo y su agente asociado, y la definición de una interfaz, a nivel de sistema, que permita la gestión virtual de la estación.

**Palabras clave:** Robótica colaborativa, agente industrial, componente I4.0, RAMI 4.0 y fabricación flexible.

## Laburpena

Ohiko fabrikazio-sistemak ekoizpen-eredu malguago baterako trantsizioan murgilduta daude, merkatuaren egungo eskaerei erantzun ahal izateko. 4.0 Industriaren paradigmak produkzio-prozesuen eraginkortasuna eta egokitzeko gaitasuna handituko duten tresnak eta teknologiak erabiltzea planteatzen du, merkatu globalizatuari eta lehiakortasun handiari aurre egiteko. Helburu horiek lortzeko, funtsezko bi faktore identifika daitezke: robotizazioa eta deszentralizazioa.

Ekoizpen-eredu berri honetan, teknologia erabakigarrietako bat elkarlaneko robotika da; izan ere, indarra, erresistentzia eta zehaztasuna ematen du zereginak egitean, eta, batez ere, pertsonak eta robotek zeregin horietan laguntzeko gaitasuna ematen du. Horrela, elkarlaneko robotikaren erabilerak prozesuen malgutasuna, produktibitatea eta kalitatea maximizatzea ahalbidetzen du.

Eragile anitzeko sistemek erakutsi dute egokiak direla fabrikazio-aktiboaren konektibitate- eta deszentralizazio-baldintzak hainbat industria-inguruetan betetzeko; izan ere, eragile industrialaren kontzeptua I4.0 osagaiaren kontzeptuaren inplementazio gisa ulertzen da (sistema ziberfisikoaren kontzeptuaren berezitasuna, RAMI 4.0 erreferentziazko arkitekturarekin lerrokatuta), agenteen bidez.

Testuinguru horretan, lan honek muntai-estazio bat (elkarlaneko robot batek eta PLC batek osatua) agente industrial gisa integratzen laguntzen du FlexManSysen barruan, UPV/EHUko GCIS ikerketa-taldeak garatutako fabrikazio-aplikazioak kudeatzeko plataforma. 4 geruzako arkitektura bat jarraitzen du, muntai-estazioa I4.0 osagai gisa plataforma adimendunaren barruan integratzea ahalbidetzen duena. Integrazio horren barruan sartzen dira: geltokiko eragiketak identifikatzea eta programatzea, modu modularrean, sistemaren malgutasuna maximizatzeke; aktiboaren eta hari lotutako eragilearen artean datuak trukatzeko beharrezko mekanismoak definitzea; eta sistema mailako interfaze bat definitzea, estazioaren kudeaketa birtuala ahalbidetzeko.

**Hitza gakoak:** elkarlaneko robotika, Industria-agentea, I4.0 osagaia, RAMI 4.0 eta fabrikazio malgua.

## Abstract

Conventional manufacturing systems are immersed in a transition towards a more flexible production model that allows them to meet current market demands. The Industry 4.0 paradigm proposes the use of tools and technologies that increase the efficiency and adaptability of production processes to deal with a globalized and highly competitive market. To achieve these objectives, two key factors can be identified: robotization and decentralization.

One of the determining technologies in this new production model is collaborative robotics, considering that it provides strength, resistance and precision in carrying out tasks, and, above all, the ability for people and robots to collaborate in carrying out these tasks. Thus, the use of collaborative robotics allows maximizing the flexibility, productivity and quality of the processes.

Multi-agent systems have proven to be adequate to meet the connectivity and decentralization requirements of manufacturing assets in different industrial environments, to the point that the Industrial Agent concept is conceived as an implementation of the Component I4.0 concept (a particularization of the Cyberphysical System concept aligned with the RAMI 4.0 reference architecture) through agents.

In this context, this work contributes to the integration of an assembly station (formed by a collaborative robot and a PLC) as an industrial agent within FlexManSys, a manufacturing application management platform developed by the GCIS research group of the UPV/ EHU. A 4-layer architecture is followed allows the integration of the assembly station as an I4.0 component within the intelligent platform. This integration comprises: the identification and programming of station operations, in a modular way to maximize the flexibility of the system; the definition of the necessary mechanisms for the exchange of data between the asset and its associated agent, and the definition of an interface, at the system level, that allows the virtual management of the station.

**Key words:** collaborative robotics, industrial agent, I4.0 component, RAMI 4.0 and FlexManSys



## **ÍNDICE DE CONTENIDOS**

---



# ÍNDICE DE CONTENIDO

<b>ÍNDICE DE FIGURAS .....</b>	<b>VII</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>XI</b>
<b>1 INTRODUCCIÓN .....</b>	<b>3</b>
1.1 Contexto actual.....	5
<b>2 OBJETIVOS Y ALCANCE .....</b>	<b>11</b>
2.1 Objetivos.....	11
2.2 Alcance.....	12
<b>3 ESTADO DE LA TECNOLOGÍA.....</b>	<b>15</b>
3.1 Industria 4.0 .....	15
3.1.1 Historia de la Automatización Industrial.....	16
3.1.2 RAMI 4.0.....	17
3.2 Robótica Colaborativa .....	21
3.2.1 UR3e de Universal Robots.....	22
3.3 Comunicaciones.....	24
3.3.1 Ethernet.....	25
3.3.2 MQTT.....	27
<b>4 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>33</b>
4.1 AAS de la estación de ensamblado .....	35
4.1.1 Estructura de datos intercambiadas entre el AAS y el asset .....	38
4.2 Integración AAS – asset de la estación de ensamblado .....	41
4.2.1 Diseño del agente pasarela.....	41
4.2.2 Conexión con el bróker MQTT.....	42
4.3 Asset de la estación de ensamblado .....	48
4.3.1 Piezas del ensamblaje.....	48
4.3.2 Descripción de la mesa de trabajo.....	50
4.3.3 Descripción de los servicios .....	52
4.3.4 Diseño del programa del robot .....	54

<b>5</b>	<b>PRUEBAS Y RESULTADOS</b>	<b>61</b>
5.1	Pruebas unitarias	62
5.1.1	Pruebas unitarias del robot	62
5.1.2	Pruebas de conexión del PLC y el bróker	62
5.1.3	Pruebas de conexión del PLC y el robot	63
5.1.4	Pruebas de conexión entre el agente máquina y el agente pasarela	64
5.1.5	Pruebas de conexión entre el bróker y el agente pasarela	65
5.2	Pruebas de integración	65
5.2.1	Pruebas de integración del asset	65
5.2.2	Pruebas de integración del bróker	67
5.2.3	Pruebas de integración del agente máquina y el agente pasarela	70
5.3	Pruebas de verificación y validación	71
<b>6</b>	<b>METODOLOGÍA DE DESARROLLO DEL TRABAJO</b>	<b>77</b>
6.1	Descripción de las tareas	77
6.1.1	Formación	78
6.1.2	Programación del robot	78
6.1.3	Conexión PLC – Robot	78
6.1.4	Comunicación MQTT	78
6.1.5	Agentes	78
6.1.6	Integración	79
6.1.7	Documentación	79
6.2	Fases del proyecto	79
6.3	Diagrama de Gantt	81
<b>7</b>	<b>ASPECTOS ECONÓMICOS</b>	<b>85</b>
<b>8</b>	<b>CONCLUSIONES</b>	<b>89</b>
<b>9</b>	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>93</b>

## **ÍNDICE DE FIGURAS**

---



## Índice de figuras

Figura 1. 1. Correlación entre la arquitectura de cuatro capas y un Componente I4.0 [7] .....	7
Figura 3. 1. Evolución de la industria [50] .....	17
Figura 3. 2. Arquitectura RAMI 4.0 [51] .....	19
Figura 3. 3. Componente I4.0 [52].....	20
Figura 3. 4. UR3e - Universal Robots [23].....	23
Figura 3. 5. PolyScope [27] .....	24
Figura 3. 6. Arquitectura MQTT [45] .....	28
Figura 4. 1. Representación del proyecto .....	34
Figura 4. 2. Mensajería entre el agente máquina y el agente pasarela.....	36
Figura 4. 3. Diagrama de secuencia de la realización de un servicio .....	37
Figura 4. 4. Bloque de datos <b>AASassetDB</b> declarado en TIA Portal .....	41
Figura 4. 5. Definición de la dirección IP y el puerto del bróker .....	43
Figura 4. 6. Permitir acceso al bróker de dispositivos externos.....	43
Figura 4. 7. Comunicación entre el PLC y el bróker MQTT.....	44
Figura 4. 8. Cliente MQTT (TIA Portal).....	45
Figura 4. 9. Diagrama de secuencia para la comunicación entre el agente máquina y el bróker.....	47
Figura 4. 10. Diagrama de secuencia para la comunicación del bróker y el PLC.....	47
Figura 4. 11. Conjunto de piezas .....	49
Figura 4. 12. Palé.....	49
Figura 4. 13. Disposición de la mesa de trabajo.....	50
Figura 4. 14. Descripción de las rampas.....	51
Figura 4. 15. Elementos dispensadores de la mesa de trabajo .....	51
Figura 4. 16. Posibles montajes finales .....	54
Figura 4. 17. Posición de los ítems sobre el palé .....	57
Figura 5. 1. Entorno de pruebas .....	61
Figura 5. 2. Arranque de Mosquitto .....	63
Figura 5. 3. Configuración Profinet en la instalación del robot.....	63
Figura 5. 4. Dispositivos y redes empleados en las pruebas de conexión del PLC y el robot .....	64
Figura 5. 5. Selección de tópicos (TIA Portal) .....	68
Figura 5. 6. Bloque de deserializar (TIA Portal) .....	69
Figura 5. 7. Bloque de serializar (TIA Portal) .....	70
Figura 5. 8. Definición de los datos de fabricación.....	71
Figura 5. 9. HMI.....	73
Figura 6. 1. Diagrama de Gantt.....	81



## ***ÍNDICE DE TABLAS***

---



## Índice de tablas

<i>Tabla 3. 1. Características de los robots de Universal Robots.....</i>	<i>23</i>
<i>Tabla 4. 1. Parámetros de control de intercambio de información (lectura/escritura).....</i>	<i>38</i>
<i>Tabla 4. 2. Datos enviados del agente máquina al PLC.....</i>	<i>39</i>
<i>Tabla 4. 3. Datos enviados del PLC al agente máquina.....</i>	<i>40</i>
<i>Tabla 4. 1. Conjunto de servicios proporcionados por el asset.....</i>	<i>53</i>
<i>Tabla 4. 2. Fases requeridas para cada servicio.....</i>	<i>55</i>
<i>Tabla 6. 1. Desglose de la ejecución de las tareas por fases.....</i>	<i>80</i>
<i>Tabla 7. 1. Aspectos económicos.....</i>	<i>86</i>



*CAPÍTULO 1*

# **INTRODUCCIÓN**

---



# 1 Introducción

En la actualidad, las empresas manufactureras se enfrentan a una serie de desafíos que están transformando la manera en que se diseñan y despliegan los sistemas de producción. Como consecuencia, son necesarios sistemas de producción inteligentes y personalizables, capaces de asegurar una gestión eficiente de los recursos de fabricación, y de adaptar su producción a la demanda y a las incidencias en planta de forma dinámica. El paradigma de la Industria 4.0 (I4.0) propone el uso de tecnologías como el big data, los gemelos digitales, los Sistemas Ciber-Físicos (Cyber-Physical Systems, CPS) o el Internet de las Cosas para permitir una fabricación inteligente y descentralizada. En este sentido, numerosas instituciones gubernamentales y organizaciones empresariales han considerado este paradigma como un factor clave en sus estrategias de desarrollo industrial para dar soporte a la innovación de los procesos de fabricación basados en la interconexión total entre activos de fabricación. Esto ha dado lugar a iniciativas como Industrie 4.0 [1] en Alemania e Industrial Internet [2] en EE. UU.

El concepto de fábrica inteligente puede llevarse a cabo mediante una red de CPS en la que la comunicación tiene lugar tanto horizontal como verticalmente, siendo necesaria la abstracción o virtualización de los activos de fabricación en cuanto a sus funcionalidades proporcionadas como servicios de red. Éste es el enfoque seguido por RAMI 4.0 (Reference Architecture Model Industrie 4.0) [3], la arquitectura de referencia de Industrie 4.0, que define el concepto de Componente I4.0 como un participante de un Sistema I4.0 que comprende un activo (asset) y un administrador del activo (Asset Administration Shell – AAS). El AAS se define como una representación virtual del Componente I4.0 en el sistema, que contiene submodelos de información a los que se puede acceder a través de un Gestor de Componentes. Los Componentes I4.0 deben cumplir ciertos requisitos: proporcionar capacidades de comunicación para garantizar la interoperabilidad entre los Componentes I4.0 del sistema (interoperabilidad); identificar de forma exclusiva tanto el AAS como el activo (identificación); representar el estado y los servicios del activo en submodelos (representación); gestionar el acceso de otros componentes a los submodelos del Componente I4.0 (gestión de

submodelos); integrar los AAS y los activos (integración); y, por último, gestionar la automatización y ejecución de los servicios realizados por el activo (gestión del activo).

Los agentes software son entidades autónomas capaces de competir o colaborar con otros agentes del sistema para alcanzar sus objetivos. Los agentes representan una alternativa adecuada para la implementación del AAS, ya que cumplen con los requisitos de interoperabilidad, identificación, representación y gestión de submodelos, y proporcionan características adicionales, como la capacidad de toma de decisiones distribuida. Sin embargo, la tecnología de agentes no contempla los requisitos de integración y gestión de activos. El paradigma de agentes industriales surge para hacer frente a esta limitación, proponiendo marcos, como la norma IEEE 2660.1-2020 [4] sobre prácticas recomendadas para agentes industriales, para la integración de un agente con un activo físico. Sin embargo, el principal reto de este paradigma se encuentra en la diversidad de activos físicos presentes en una fábrica en cuanto a la heterogeneidad de las interfaces de comunicación de los activos físicos y la ubicación de los agentes.

En este sentido, este trabajo parte de los resultados previos del grupo de investigación "Control e Integración de Sistemas" (GCIS) del Departamento de Ingeniería de Sistemas y Automática, UPV/EHU, para la implementación de agentes industriales mediante una arquitectura de cuatro capas compatible con la norma IEEE 2660.1-2020: una capa superior o de inteligencia, que aborda las capacidades de comunicación e interoperabilidad, además de otras capacidades asociadas a los agentes, como la negociación y la toma de decisiones; una capa intermedia o de gestión, que transforma los datos recibidos del activo en información útil para los Componentes I4.0 en el sistema; y dos capas inferiores, operativa y funcional, que gestionan la llamada, la ejecución y el estado de los servicios del activo.

En este contexto, este trabajo contribuye a la integración de una estación de ensamblado (basada en un robot UR3e y automatizada con un PLC S7-1500) como un agente industrial dentro de una arquitectura inteligente para fabricación flexible. Esta integración consta de la programación de las operaciones de la estación (capa funcional); la definición de una interfaz, a nivel de servicio, para la estación (capa operativa); la provisión de mecanismos para el intercambio de datos entre el activo y

su agente asociado (capa de gestión); y la definición de una interfaz, a nivel de sistema, para la gestión virtual de la estación (capa de inteligencia).

La descripción del proyecto se recoge en esta memoria organizada en ocho capítulos. El primer capítulo se dedica a la introducción y el contexto en el que se enmarca el proyecto. En el segundo capítulo se presenta, por un lado, el objetivo principal con los objetivos parciales, y, por otro lado, el alcance del proyecto. El siguiente capítulo aborda el estado de las tecnologías utilizadas en el proyecto. En el cuarto capítulo se detalla la solución propuesta para el agente industrial de la estación de ensamblado, mientras que en el quinto capítulo se presentan las pruebas desarrolladas y los resultados obtenidos. En los dos capítulos siguientes se describe la metodología de desarrollo del proyecto y los aspectos económicos del mismo, para finalizar con las conclusiones en el capítulo 8.

## **1.1 Contexto actual**

El trabajo que se presenta en esta memoria se ha desarrollado en el marco del proyecto "Arquitectura inteligente para la industria 4.0: realimentación a través del fog (SMARTFOG)". Concretamente, este trabajo se integra en el demostrador de la plataforma de gestión de aplicaciones de fabricación FlexManSys.

FlexManSys está diseñada para lograr flexibilidad a través de un conjunto de agentes distribuidos que se pueden personalizar para diferentes dominios por medio de modelos de dominio [5]. Los sistemas multiagente se han utilizado con frecuencia para el desarrollo de sistemas de fabricación flexibles con inteligencia distribuida. La interacción entre estos agentes permite la trazabilidad del proceso de fabricación y la capacidad de hacer frente a situaciones no deseadas. Así, la flexibilidad se logra a través de la cooperación y competencia entre agentes.

FlexManSys se basa en un middleware basado en agentes cuyo núcleo consiste en el 'System Agent' (SA), que proporciona una interfaz de programación de aplicaciones para dar de alta, iniciar y detener aplicaciones de fabricación, así como consultar y actualizar el estado del sistema de fabricación (almacenado en el System Model, SM) durante todo el ciclo de ejecución de la aplicación. El SM representa el estado global del sistema de fabricación con la información de uso para otros agentes, es decir, el estado de los recursos y los servicios que ofrecen, y el conjunto de planes de fabricación registrados definidos por sus órdenes.

En FlexManSys cada recurso está representado por su Agente de Recursos (Resource Agent, RA) correspondiente. Un RA gestiona la operación de una estación, ofrece servicios y entra en negociaciones con otras estaciones para realizar operaciones, actualizando su plan en consecuencia. Además, las aplicaciones de manufactura son administradas por Agentes de Planes de Manufactura (MPA), Agentes de Pedidos (OA) y Agentes de Lote (BA). Un MPA supervisa un conjunto de OA, cada uno de los cuales es responsable de gestionar el pedido de un cliente. Un OA, a su vez, supervisa un conjunto de BA. Por lo tanto, se pueden distinguir dos grandes tipos de agentes. Por un lado, están los agentes máquina que representan los servicios y necesidades de los recursos disponibles en la fábrica, encargados de interactuar con su activo, representando su estado y ofreciendo sus servicios al resto de agentes del sistema. Por otro lado, están los agentes de producto los cuales se encargan de la gestión del producto o lotes del producto [6], pudiendo llevar a cabo su trazabilidad. En el caso de los agentes de producto, se dedican a realizar la trazabilidad de su activo. Una trazabilidad basada en agentes permite aumentar la agilidad tanto en la interacción con otras entidades de la fábrica como en la capacidad de toma de decisión, pudiendo asumir algunas de las funcionalidades tradicionalmente asociadas al MES (Manufacturing Execution System).

El trabajo realizado por los autores en [7], propone una arquitectura de 4 capas para la implementación de diferentes aspectos de la integración entre AASs basados en agentes máquina y los activos. La división realizada en capas potencia la separación de responsabilidades, encapsulando los diferentes aspectos de integración en diferentes niveles de abstracción (véase en la Figura 1.1).

- **Capa de inteligencia** (Intelligence Layer). Implementa la interfaz orientada a servicios del Componente I4.0, permitiendo la interacción con otros Componentes I4.0 del sistema. También identifica al AAS y al activo para que puedan ser reconocidos en las comunicaciones. Además, implementa las capacidades de toma de decisiones en base a la información recibida del exterior y la información que reside en la capa de gestión.
- **Capa de gestión** (Management Layer). Corresponde al gestor de componentes del AAS. Contiene los submodelos que representan el estado y los servicios del activo, y se encarga de mantenerlos actualizados. Para ello, gestiona las solicitudes realizadas por la capa de inteligencia para acceder a los

datos y funciones en los submodelos, interactuando con la capa operativa cuando es necesario. Por tanto, en este nivel, la ejecución de los servicios se reduce a la transmisión y actualización de información en los submodelos.

- **Capa operativa (Operative Layer).** Implementa la lógica detrás de la ejecución de los servicios, es decir, interpreta las solicitudes recibidas de la capa de gestión y desencadena las acciones necesarias para poder entregar una respuesta. Esta capa interactúa con la capa funcional en caso de que se requiera la ejecución de un servicio para responder a una solicitud de la capa de gestión.
- **Capa funcional (Functional Layer).** Realiza los servicios que ofrece el Componente I4.0. Recibe información de la capa operativa a través de los parámetros de entrada al invocar el servicio. Del mismo modo, los resultados de la ejecución del servicio se devuelven como parámetros de salida.

Aunque el uso de capas permite resolver los requisitos por separado, esto no significa que las capas sean independientes entre sí. Todas las capas son necesarias para cumplir con la funcionalidad.

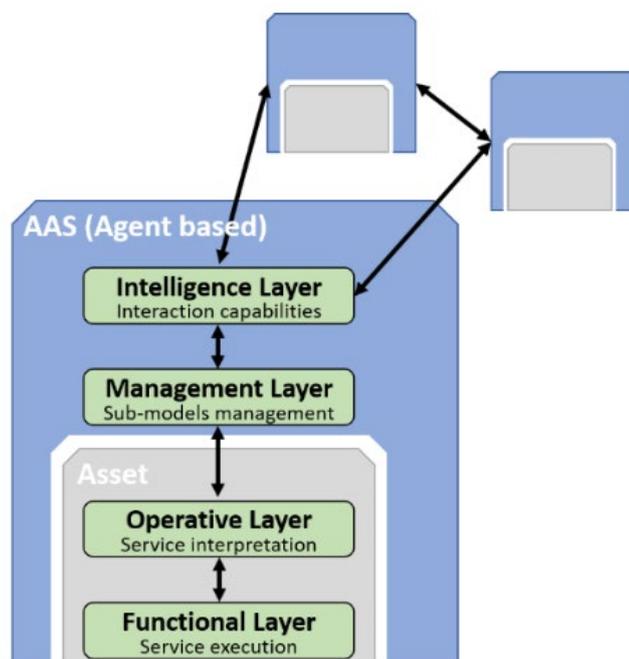


Figura 1. 1. Correlación entre la arquitectura de cuatro capas y un Componente I4.0 [7]



*CAPÍTULO 2*

# **OBJETIVOS Y ALCANCE**

---



## 2 Objetivos y alcance

En este capítulo de la memoria se presenta el objetivo principal del proyecto y los objetivos parciales que permitirán la consecución del mismo. Además, se describe el alcance del trabajo.

### 2.1 Objetivos

El objetivo principal de este trabajo consiste en la integración de un asset físico en un sistema de fabricación flexible conforme al paradigma de la Industria 4.0. Concretamente, se realiza la integración de un robot colaborativo UR3e (que forma parte de una estación de ensamblado) como un Componente I4.0, dentro de una arquitectura inteligente basada en agentes industriales.

Para poder alcanzar el objetivo principal definido, es necesario el cumplimiento de los objetivos parciales que se describen a continuación. El primero estaría relacionado con el desarrollo del asset físico, compuesto por el robot colaborativo y un PLC. Mientras que los dos siguientes estarían orientados al desarrollo de su ASS, necesario para su integración en la arquitectura inteligente. Dichos objetivos parciales están definidos en los siguientes puntos.

1. **Modularidad en los servicios ofrecidos por el robot:** tanto la identificación de los servicios ofrecidos por el robot como su programación deben estar orientados no sólo a cubrir necesidades de producción normal, sino también a poder hacer frente a situaciones excepcionales de producción (e.g.: llegada de pedidos de prioridad alta, recuperación ante el fallo de un robot que ofrece algún servicio similar, etc).
2. **Protocolo de comunicación a nivel de aplicación:** para lograr una correcta integración del asset físico en la arquitectura inteligente, es necesaria una comunicación bidireccional en la que el robot, por un lado, recibirá las órdenes necesarias y, por otro lado, deberá proporcionar información del estado de evolución de sus tareas. Para el desarrollo de la comunicación se hace uso de un protocolo asíncrono, el cual permite que

cada una de las partes no tenga la necesidad de esperar a recibir la información.

3. **Arquitectura de integración:** teniendo en cuenta las capacidades de comunicación de las tecnologías empleadas en este trabajo (agentes industriales y una estación de ensamblado con robot UR3e) y el protocolo de nivel de aplicación definido, será necesario definir los elementos de integración necesarios, así como sus interfaces de comunicación.

## 2.2 Alcance

Este trabajo se enmarca dentro del ámbito de la Industria 4.0 y los sistemas de fabricación flexible, centrándose en la integración de un nuevo activo en la arquitectura inteligente del sistema de fabricación. En concreto, el sistema de fabricación inteligente es implementado mediante la arquitectura basada en agentes FlexManSys, mientras que el activo con el que se trabaja es un robot colaborativo UR3e de Universal Robot.

Los agentes inteligentes de FlexManSys decidirán qué tarea debe realizar el robot que, además, proporcionará información de estado sobre su evolución. Como cualquier otro agente inteligente, la implementación del agente máquina se realizará conforme a la arquitectura de 4 capas. No obstante, no está contemplado entre los objetivos del proyecto la integración de dicho agente con los restantes agentes de FlexManSys. Por tanto, la capa superior se implementará parcialmente, siendo objeto de trabajo a futuro complementario dotando al agente de las capacidades de negociación y toma de decisiones en el marco de la citada arquitectura inteligente.

Los servicios que debe desempeñar el robot colaborativo UR3e deben coincidir con los realizados por otro robot Kuka existente en el demostrador, con el objetivo de permitir validar los aspectos de flexibilidad de la plataforma de gestión de aplicaciones de fabricación FlexManSys. Por este motivo, ambos robots presentan características similares en cuanto a alcance y capacidad de carga, y el layout del entorno de trabajo coincide.

*CAPÍTULO 3*

# **ESTADO DE LA TECNOLOGÍA**

---



### 3 Estado de la tecnología

La evolución de la humanidad contempla 4 revoluciones industriales hasta el momento. A consecuencia de las líneas de montaje de la segunda revolución industrial, se han introducido diferentes adelantos tecnológicos relacionados con la producción en masa. A partir de la tercera revolución industrial, la introducción de robots dio pie a la automatización, la informática y la electrónica en el sector industrial.

Esta evolución hace evidente la necesidad de comunicar los dispositivos que se encuentran en una planta de trabajo. Con el paso de los años, las comunicaciones han ido evolucionando de manera que hoy en día se dispone de una variedad amplia de protocolos de comunicaciones industriales, lo cual permite elegir el que mejor se ajusta a la hora de realizar el trabajo. Estos protocolos son los que hacen posible el intercambio de datos entre los dispositivos.

A lo largo de este capítulo se presentan las diferentes tecnologías base, las cuales son necesarias para comprender la propuesta realizada en el trabajo. Antes de comenzar con la descripción de las tecnologías, se define la Industria 4.0 (I4.0), sus antecedentes históricos y la arquitectura de referencia que sigue este trabajo. Dado que para el desarrollo del proyecto se dispone de una estación de ensamblado compuesta por un PLC y un robot colaborativo UR3e, el capítulo continúa definiendo este tipo de robots y, concretamente, el robot utilizado. Se finaliza el capítulo con las comunicaciones industriales, centrándose en las dos utilizadas durante el proyecto. Por un lado, se ha hecho uso de Profinet (protocolo de Ethernet Industrial) para la comunicación entre el PLC y el robot. Por otro lado, el resto de comunicaciones del proyecto se realizan mediante MQTT.

#### 3.1 Industria 4.0

La Industria 4.0, también conocida como la cuarta revolución industrial, es un término que se escucha tanto en el sector industrial como en la vida cotidiana. Esta revolución ha modificado las características básicas de la industria hasta el momento [8]. Este término fue introducido por primera vez el año 2011 en Alemania, siendo más

exactos, en la feria de Hannover. La Industria 4.0 se refiere a la inteligencia de la red de trabajo de máquinas y procesos para la industria con la ayuda de las tecnologías de la información y la comunicación (TIC) [9], [10].

### **3.1.1 Historia de la Automatización Industrial**

Para entender mejor el concepto de Industria 4.0 es necesario conocer de dónde proviene (véase en la Figura 3.1), requiriendo definir algunos términos sobre las tres revoluciones industriales anteriores. La Primera Revolución Industrial (Industria 1.0) está relacionada con la mecanización de la producción y la introducción de la máquina de vapor. La Segunda Revolución Industrial (Industria 2.0), en cambio, comenzó con el descubrimiento de la electricidad y la producción en línea de ensamblaje. Henry Ford trasladó los principios de la producción en masa a la industria del automóvil, consiguiendo producciones más rápidas con menores costes [11].

La Tercera Revolución Industrial (Industria 3.0, I3.0), también conocida como la revolución científico-tecnológica (RCT), se dio alrededor de los años 70. Esta revolución se centra en la producción en masa y el uso masificado de la lógica digital, concretamente, en los transistores MOS y chips de circuitos integrados, y todas las tecnologías que derivan de estos dos. El uso de estos productos trajo innovaciones que transformaron los métodos tradicionales de producción, permitiendo introducir en la industria la automatización parcial, usando controladores con memoria programable y computadoras. Lo que supuso el comienzo de la introducción de robots en el sector industrial.

El origen de la Cuarta Revolución Industrial (Industria 4.0, I4.0) representa la combinación de las nuevas formas en la que la tecnología puede ser integrada tanto en la industria como en la vida cotidiana, mediante la fusión de las tecnologías actuales. Dentro de esta revolución se definen avances tecnológicos como la inteligencia artificial, la nanotecnología y la impresión 3D entre otros [12].

En los siguientes puntos se muestran algunas de las diferencias que existen entre la Industria 3.0 y la Industria 4.0 [13]:

- La I3.0 se limita a los sistemas físicos, mientras que la I4.0 está enfocada a los sistemas ciberfísicos, compuestos por sistemas reales y virtuales al mismo tiempo.

- La Inteligencia Artificial (Artificial Intelligence, AI) es la precursora de la I4.0. A través de ella se integran conceptos como Machine Learning y manejo de grandes cantidades de datos (BigData), entre otros.
- La I4.0 se basa en las comunicaciones rápidas entre los diferentes elementos de la industria para una adaptación igualmente veloz a los cambios.

Resumiendo, la principal diferencia entre las dos últimas revoluciones industriales se encuentra a nivel tecnológico. La I3.0 surgió hace varias décadas, lo que supone que las tecnologías utilizadas durante esa revolución sean menos satisfactorias en comparación con las de la I4.0. Es decir, la cuarta revolución da un paso más, no solo temporal, sino también en la digitalización del dato. Lo cual, acompañado del gran desarrollo de las tecnologías actuales de procesamiento y comunicación permite hacer frente a las exigentes demandas de flexibilidad de los actuales sistemas de fabricación.

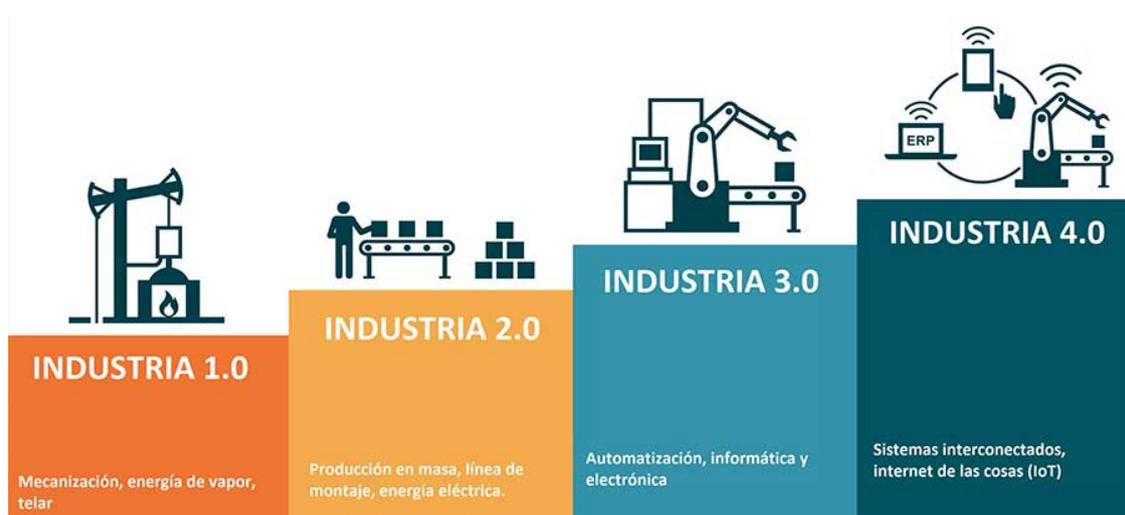


Figura 3. 1. Evolución de la industria [50]

### 3.1.2 RAMI 4.0

La evolución de la industria conlleva la necesidad de la generación de modelos de arquitectura de referencia, que permitan aplicar buenas prácticas a los diseños sin el estorbo de restricciones innecesarias y arbitrarias. Las arquitecturas más conocidas en el sector industrial son las siguientes: RAMI 4.0 (Reference Arquitectural Model Industrie, Europa), NIST (National Institute of Standards and Technology), IIRA

(Industrial Internet Reference Architecture, Industrial IoT Consortium – IIC) y IMSA (Intelligent Manufacturing System Framework).

En Europa, un grupo de corporaciones y asociaciones (BITKOM – Asociación Federal de Tecnologías de la Información, Telecomunicación y Nuevos Medios, VDMA – Asociación Alemana de Fabricantes de Maquinaria y ZVEI – Asociación Central de Ingeniería Eléctrica e Industrial Digital Alemana) de diferentes puntos geográficos sugirieron una arquitectura de referencia para la I4.0, la citada RAMI 4.0 [8], [14], que es la que se adoptará en el desarrollo del presente proyecto.

En el año 2015 se presentó la RAMI 4.0 que, como tal arquitectura, define una estructura organizada de un sistema o componente, sus relaciones, sus principios y sus pautas controlando el diseño y la evolución en el tiempo. Se trata de una arquitectura orientada a servicios, en la cual unos componentes proporcionan servicios a otros a través de un protocolo de comunicación y una red, ayudando a la clasificación e identificación dentro de la Industria 4.0. El propósito de la RAMI 4.0 es representar el objeto técnico y todos los aspectos relevantes para él, desde su desarrollo, producción y uso hasta su eliminación [15].

La RAMI 4.0 es un mapa tridimensional, lo que hace que esté formada por tres ejes: niveles de jerarquía, ciclo de vida y flujo de valor, y varias capas verticales (véase en la Figura 3.2) [15], [16]. Mediante esta estructura tridimensional se facilita un modelo que puede ser visto desde diferentes perspectivas, con lo que se busca abordar la organización de la empresa desde tres puntos de vista principales que engloban varios aspectos del proceso de fabricación. La primera dimensión propone una jerarquía funcional para todos los componentes de la Industria 4.0 [17]. La segunda dimensión propone un modelo de datos consistente durante todo el ciclo de vida y toda la cadena de valor del producto. Por último, la tercera dimensión propone un modelo de capas que permita integrar fácilmente diferentes tecnologías.

- **Jerarquía.** Representa las características funcionales de los componentes basándose en las normativas IEC 62264-1 y IEC 61512-1. Consiste en siete niveles individuales: Producto, Dispositivo de campo, Dispositivo de control, Estación, Centros de trabajo, Empresa y El mundo conectado. Estos niveles representan las diferentes funcionalidades dentro de una empresa [3].

- **Ciclo de Vida del Producto.** Describe el proceso de tipo a instancia desde una perspectiva de ciclo de vida. El modelo RAMI 4.0 puede representar los datos recopilados durante el ciclo de vida, desde el desarrollo hasta su eliminación.
- **Capas.** Un Componente I4.0 consiste en seis capas: Negocios y Funcional para gestionar el proceso de manufactura, Información para recopilar los datos relativos al producto, y Comunicación, Integración y Activos las cuales hacen referencia a los componentes de la planta. La suma de estas capas permite obtener el Componente I4.0.

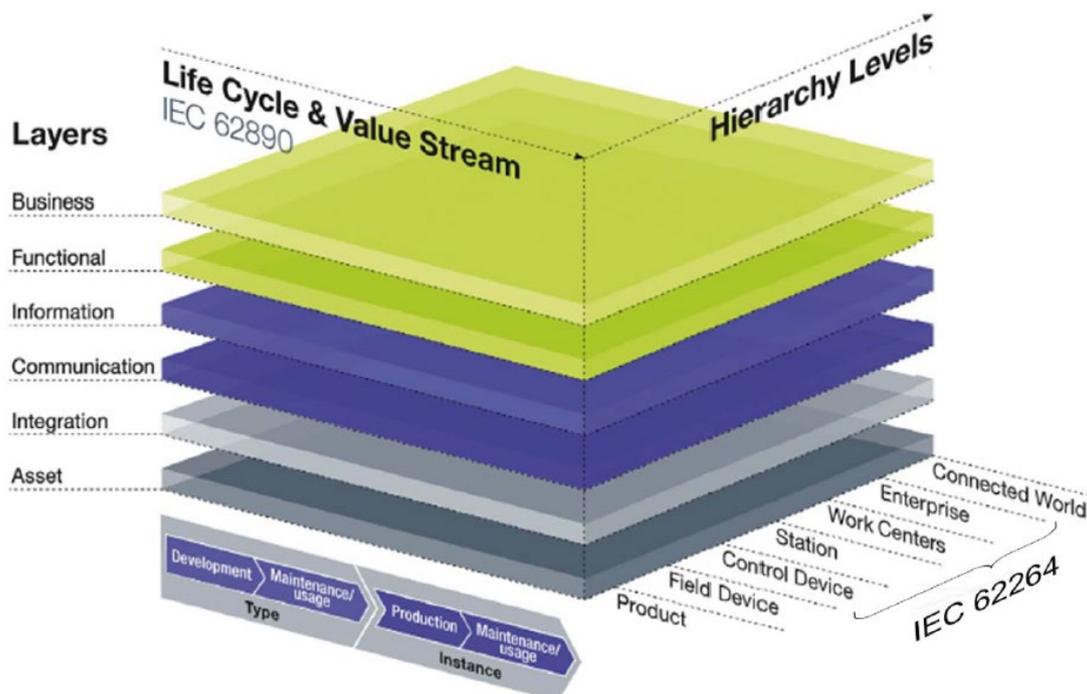


Figura 3. 2. Arquitectura RAMI 4.0 [51]

### 3.1.2.1 Componente I4.0

La RAMI 4.0 integra diferentes perspectivas de usuarios y proporciona una comprensión común de la I4.0. La característica especial de RAMI 4.0 es su combinación de ciclo de vida y flujo de valor con un enfoque estructurado jerárquicamente para la definición de Componentes I4.0.

Dentro de la RAMI 4.0, un Componente I4.0 es aquel participante globalmente identificable con capacidad de comunicación que consiste de un activo (asset) y su ASS

dentro de un sistema I4.0 (véase en la Figura 3.3). Estos componentes ofrecen servicios con características definidas de calidad de servicio (QoS).

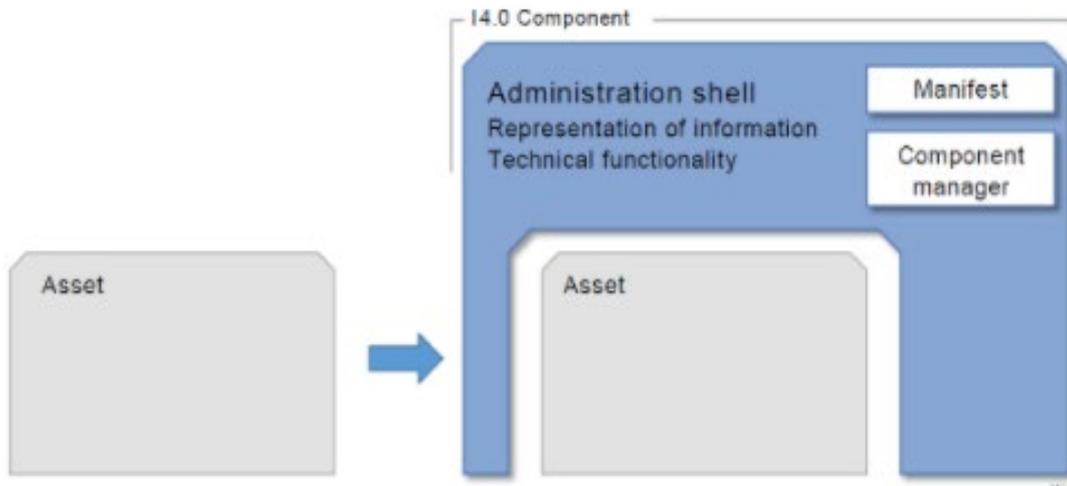


Figura 3. 3. Componente I4.0 [52]

Se denomina activo a cualquier entidad que participa en el proceso de fabricación. A pesar de que la idea de un activo sea algo físico, es posible que el activo sea algo digital como un documento, plano... Un activo no tiene que ser necesariamente un Componente I4.0. No obstante, es posible convertir un activo en un Componente I4.0 haciendo uso de un Shell de Administración.

El ASS es la representación virtual, digital y activa de un Componente I4.0, que está compuesto por un cuerpo y una cabecera. La cabecera contiene información de identificación, mientras el cuerpo comprende una cierta cantidad de submodelos y una caracterización específica de los activos de AAS [17], [18].

El AAS integra el activo en la comunicación de I4.0: proporciona un acceso controlado a toda la información del activo; es la interfaz de comunicación estandarizada y segura; y es direccionable en la red e identifica el activo inequívocamente. No obstante, también puede integrar activos 'pasivos' sin una interfaz de comunicación [19], incluyendo,

- Modelos desde diferentes puntos de vista.
- Enlaces a otros activos.

- Servicios para la cooperación y el diálogo entre activos y con gestores de producción.

### 3.2 Robótica Colaborativa

Como se ha mencionado, la I3.0 supuso la introducción de robots industriales en el entorno de la fabricación, consiguiendo que los robots realizaran los trabajos de mayor esfuerzo.

Los robots industriales eran sistemas mecatrónicos avanzados que integraban sinérgicamente el diseño mecánico, la electrónica, el software y el control. Estos robots no tenían conciencia real de lo que sucedía a su alrededor, por lo que se mantenían separados mediante una valla de seguridad o disponían de sensores para la seguridad [20]. Los robots operaban de manera aislada. Para mayor precaución, estos robots tenían un color llamativo para que los usuarios pudiesen conocer la posición actual del robot. Cada robot perteneciente a la fabricación definía una celda de fabricación aislada, con una tarea de fabricación específica. Estas celdas estaban conectadas al resto del proceso mediante sistemas de transporte, habitualmente cintas transportadoras. Cada uno de los robots que componían la estación de trabajo era programado de manera independiente, incluso aquellos robots que trabajaban juntos en una misma pieza de trabajo [21].

Una de las tecnologías que mayor repercusión e impacto ha tenido en el contexto de la I4.0 ha sido la Robótica Colaborativa. Término general que transmite la idea de proximidad entre máquinas y humanos para algunas tareas útiles en un espacio compartido [22]. La Robótica Colaborativa pretende ser complementaria a la robótica convencional, consiguiendo aumentar la participación humana, en términos de tiempo y de espacio compartido. Los robots colaborativos, también conocidos como cobots, describen a un robot diseñado para asistir a un humano en una tarea específica. Tienen la capacidad de interactuar con los humanos sin generarles ningún tipo de daño físico [23]. Este tipo de robots permiten el trabajo simultáneo en la misma zona de trabajo entre un robot y un humano.

Los robots colaborativos entran dentro del grupo de robots de servicio. El objetivo de estos robots es ayudar y facilitar el trabajo del usuario, de manera que puede realizar los trabajos monótonos sin ningún tipo de fatiga. Dentro de los robots

de servicio se dividen dos subcategorías, robots de servicio profesional y robots de servicio de uso personal. Los robots de servicio profesionales son los que interactúan junto a los trabajadores (usuarios). Los robots de servicio de uso personal son aquellos que poseen un nivel de autonomía complejo, rico y próximo al HRI (Human-Robot Interaction). En este proyecto se dispone de un robot de servicio profesional.

Las principales características de un cobot son la colaboración, la automatización y la versatilidad.

- **Colaboración.** Los cobots fueron creados para ayudar a las personas, por lo tanto, la interacción con los operarios es fundamental.
- **Automatización.** Un cobot es capaz de realizar tareas de automatización, logrando disminuir el tiempo que se necesita para efectuar el trabajo a realizar [24].
- **Versatilidad.** Los cobots pueden ser integrados en cualquier tipo de empresa llevando a cabo distintos procesos industriales.

Dentro de la industria de robots colaborativos se encuentran diferentes empresas que se dedican a la fabricación de dichos robots, tales como ABB, Universal Robots o KUKA, entre otras.

### 3.2.1 UR3e de Universal Robots

El robot de la estación de ensamblado utilizada en este trabajo es el UR3e de Universal Robots. Universal Robots [25] es una empresa de Robótica Industrial que se dedica a fabricar brazos robóticos colaborativos de 6 ejes de todos los tamaños. Estos robots están definidos como flexibles, seguros y fáciles de usar en una empresa.

Los robots se distinguen por la capacidad de carga que pueden soportar. Esto significa que, dependiendo del nombre definido para cada uno de los robots, estos serán capaces de soportar dicha carga. Entre estos robots están el UR3e, UR5e, UR10e y UR16e.

Cada una de las características propias de los robots mencionados anteriormente se recopilan en la Tabla 3.1. En ella se muestra tanto el nombre de cada uno de los robots, como su peso, alcance, carga útil y la rotación de sus articulaciones.

Tabla 3. 1. Características de los robots de Universal Robots

Nombre del robot	Alcance	Peso del robot	Carga útil	Rotación de las articulaciones
UR3e	500mm	11kg	3kg	$\pm 360^\circ$
UR5e	850mm	20.6kg	5g	$\pm 360^\circ$
UR10e	1300mm	33.5kg	12.5kg	$\pm 360^\circ$
UR16e	900mm	33.1kg	16kg	$\pm 360^\circ$

El brazo robótico s-Series Universal Robots UR3e, es el robot colaborativo más pequeño de la gama de Universal Robots (véase en la Figura 3.4). Es un cobot industrial ultraligero y compacto, adecuado para implementar directamente dentro de maquinaria u otros espacios pequeños de trabajo. Tiene un peso de 11kg, una carga útil de 3kg, un radio de trabajo de 500mm desde la junta base y todas sus articulaciones pueden llegar a tener una rotación de  $\pm 360^\circ$ . Está definido como un robot perfecto para tareas ligeras de ensamblaje y para trabajos que requieren precisión [26].

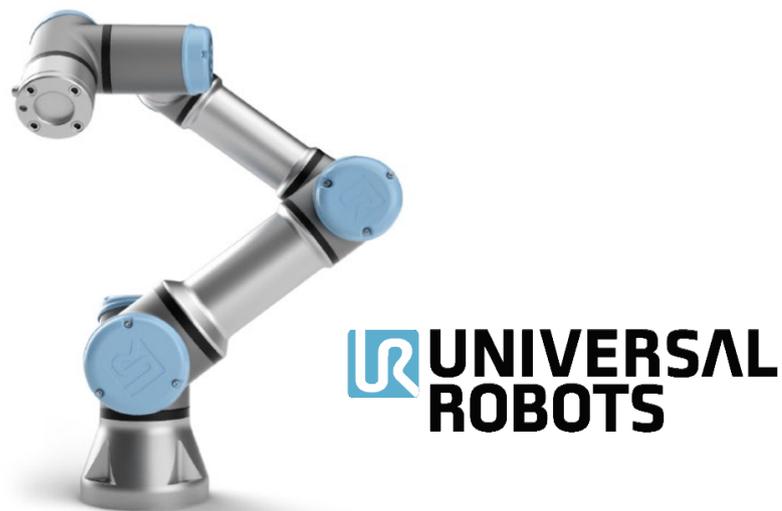


Figura 3. 4. UR3e - Universal Robots [23]

El robot dispone de un puerto de Ethernet adicional que permite comunicar mediante MODBUS, EtherNet/IP y PROFINET, permitiendo un acceso y un control remoto del dispositivo [27].

Para programarlo se dispone de PolyScope (véase en la Figura 3.5) que es una Interfaz Gráfica de Usuario (Graphical User Interface, GUI) en el Teach Pendant. Esta interfaz permite operar con el brazo robótico y la controladora, permitiendo ejecutar programas.

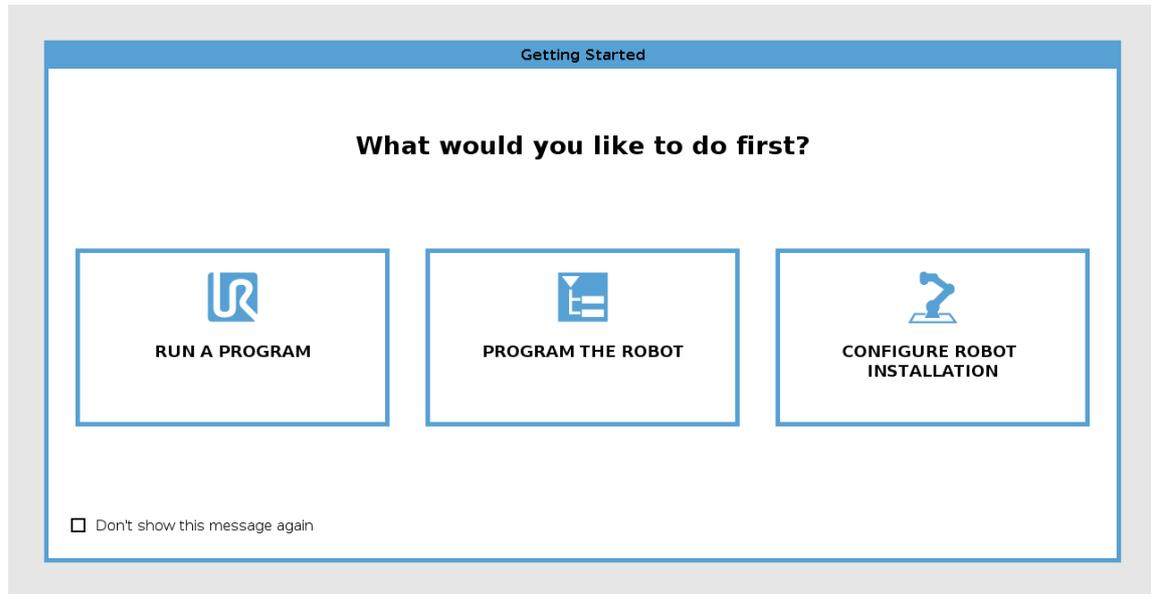


Figura 3. 5. PolyScope [27]

### 3.3 Comunicaciones

Con el paso de los años, las funcionalidades de los robots colaborativos van mejorando, consiguiendo que realicen trabajos más complejos y con mayor perfección. La necesidad de ser más eficientes y conseguir reducir el tiempo de producción, obliga a compartir tareas entre robots. Esto conlleva la necesidad de tener que realizar comunicaciones entre ellos [28].

Los protocolos de comunicación industriales hacen posible el intercambio de información entre diferentes dispositivos durante los procesos, consiguiendo formar una red de comunicación con todos los dispositivos que estén comunicados entre sí [29].

Por un lado, se ha generado la necesidad de evolucionar los antiguos protocolos ante la evolución de la industria, y, por otro lado, surge la necesidad de crear nuevos estándares basados en nuevas tecnologías. Entre los protocolos antiguos y los nuevos estándares, destacan protocolos como Ethernet Industrial, PROFIBUS, MQTT y AS-I. A

la hora de desarrollar el proyecto se ha hecho uso de los protocolos MQTT y PROFINET, siendo este último un estándar de Ethernet.

### 3.3.1 Ethernet

Ethernet [30] es uno de los protocolos más utilizados en el mundo, que surge de un esfuerzo colaborativo de la corporación XEROX. Es una tecnología tradicional que permite conectar dispositivos en una red de área local (LAN, Local Area Network) o una red de área amplia (Wide Area Network, WAN) por cable. Permite comunicar dos dispositivos a través de un protocolo, el cual define un lenguaje de red común [31].

La red de área local Ethernet proporciona una comunicación para el intercambio de datos de alta velocidad entre computadoras y otros dispositivos digitales, ubicados dentro de un área geográfica de tamaño moderado que cubre casas, oficinas... La IEEE define a la Ethernet como el Protocolo 802.3.

Esta red es usada mundialmente debido a las ventajas que ofrece: bajo coste, compatibilidad con versiones anteriores, resistente al ruido, velocidad, fiabilidad y seguridad de los datos.

#### 3.3.1.1 Ethernet Industrial

Industrial Ethernet es el uso de Ethernet en un entorno industrial con protocolos que proporcionan determinismo y control en tiempo real [32]. Estos protocolos incluyen:

- **EtherCAT.** Tecnología de Ethernet Industrial en tiempo real. Es un protocolo informático de código abierto y alto rendimiento creado y desarrollado por Beckhoff Automation [33].
- **EtherNET/IP.** Basado en un modelo de objeto llamado CIP (Control and Information Protocol), es un protocolo de red industrial.
- **PROFINET.** Responde a la necesidad de interoperabilidad entre los dispositivos de automatización y los subsistemas, los cuales están enlazados mediante Ethernet.
- **POWERLINK.** Protocolo de comunicación en tiempo real basado en hardware estándar Ethernet, que es capaz de expandir Ethernet mediante un mecanismo mixto de sondeo y distribución de tiempo [34].

- **SERCOS III.** Estandarización abierta a interfaces digitales para comunicaciones de tiempo real a alta velocidad entre controladores industriales, dispositivos de movimiento y dispositivos de entradas/salidas [35].
- **CC-Link IE.** Red con alto nivel de velocidad y con gran capacidad de cubrir largas distancias, que proporciona tanto comunicación asíncrona como comunicación síncrona determinista [36].

Teniendo en cuenta los diferentes protocolos que incluye Ethernet Industrial, en este proyecto se ha realizado la comunicación entre el PLC y el robot mediante el estándar abierto PROFINET. Dicho estándar es uno de los más utilizados mundialmente de Ethernet Industrial. El objetivo principal de este protocolo está en la reducción de los costes de ingeniería mediante la integración con plantas de automatización existentes, incluso estando basadas en buses de campos tradicionales [37], [38].

Este protocolo de comunicación considerado muy rápido a nivel de bus, permite intercambiar datos de forma rápida y determinista entre controladores y dispositivos de automatización [39]. La comunicación mediante Profinet puede realizarse mediante cable Ethernet, cable de fibra óptica, cable de alimentación a través de Ethernet o inalámbrico [40]. Profinet permite el intercambio de datos de una manera rápida, precisa y organizada.

Profinet es un estándar de Ethernet Industrial desarrollado por Profibus International (PI) para ingeniería de plantas y máquinas modulares o E/S distribuidas. Consiste en varios temas como la automatización distribuida (Profinet CBA), los dispositivos de campo descentralizados, la gestión de redes, las pautas de instalación y la integración de web [41].

Dependiendo del tipo de información que se quiera transmitir existen diferentes mecanismos de transmisión, los cuales conforman los 'canales de comunicación'. Dentro de Profinet existen tres canales de comunicación: en tiempo real (RT), en tiempo no real (NRT) y en tiempo real isócrono (IRT).

- **Tiempo real, RT.** Define la comunicación en tiempo real manejando el intercambio de datos de tiempo crítico. Consigue tiempos de actualización mayores a los 250 $\mu$ s.

- **Tiempo no real, NRT.** Utiliza todas las capas del modelo OSI y cuenta con direcciones IP, permitiendo acceder a los dispositivos a través de enrutamiento o internet. Este canal es utilizado mayormente para las comunicaciones menos sensibles, como la configuración entre el dispositivo y el controlador.
- **Tiempo real isócrono, IRT.** Define una comunicación en tiempo real con ancho de banda reservado y ciclos sincronizados. Consigue tiempo de actualización mayores a los 125µs.

### 3.3.2 MQTT

El protocolo MQTT, creado en 1999, es un protocolo de red de mensajería ligero estándar de OASIS para IoT (Internet of Things) y M2M (Machine to Machine). Está diseñado para el transporte de mensajería de publicación-suscripción entre dispositivos remotos, consiguiendo publicar y suscribir datos en tópicos. MQTT permite la mensajería entre un dispositivo y la nube y viceversa, consiguiendo facilitar el envío de datos en las dos direcciones. Sus siglas hacen referencia a 'Message Queing Telemetry Transport' [42], [43].

Este protocolo de publicación-suscripción (pubsub) está diseñado pensando en dispositivos pequeños [44]. Los sistemas Pubsub funcionan como un bus de mensajes. Un dispositivo proporciona un mensaje con información a un tema (tópico) y cualquier software con una suscripción en ese tema obtiene una copia de su mensaje.

Son varias las ventajas que ofrece este protocolo, pero la principal está relacionada con la seguridad de una empresa. Este protocolo permite aumentar la seguridad, esto es debido a que los equipos están conectados a los servidores, de tal manera que se consigue almacenar los datos en una entidad.

A día de hoy, MQTT es uno de los protocolos que más se utiliza en una amplia variedad de entornos industriales. Por ejemplo: automóvil, logística, fabricación, viviendas inteligentes (domótica), gas, petróleo y transporte.

El protocolo MQTT como mínimo puede realizar 4 acciones: publicar, suscribirse, cancelar la suscripción y hacer ping.

- **Publicar.** Envío de un bloque de datos.
- **Suscribirse.** Suscripción de un cliente en un tópico.

- **Cancelar la suscripción.** Desconexión del bróker, lo puede realizar tanto el suscriptor como el editor.
- **Ping.** Es utilizado para comprobar si la conexión sigue funcionando. El cliente manda un paquete al bróker y este le contesta con otro paquete.

### 3.3.2.1 Arquitectura MQTT

MQTT es un protocolo que sigue el paradigma publicista/suscriptor. Este mecanismo desacopla al cliente que envía mensajes (publicista) de uno o varios clientes que reciben los mensajes (suscriptores). Un bróker es un servidor con el que se comunican los clientes, estos clientes no se comunican entre sí. Los clientes que se conectan al bróker pueden ser publicistas, suscriptores o ambos (véase en la Figura 3.6).

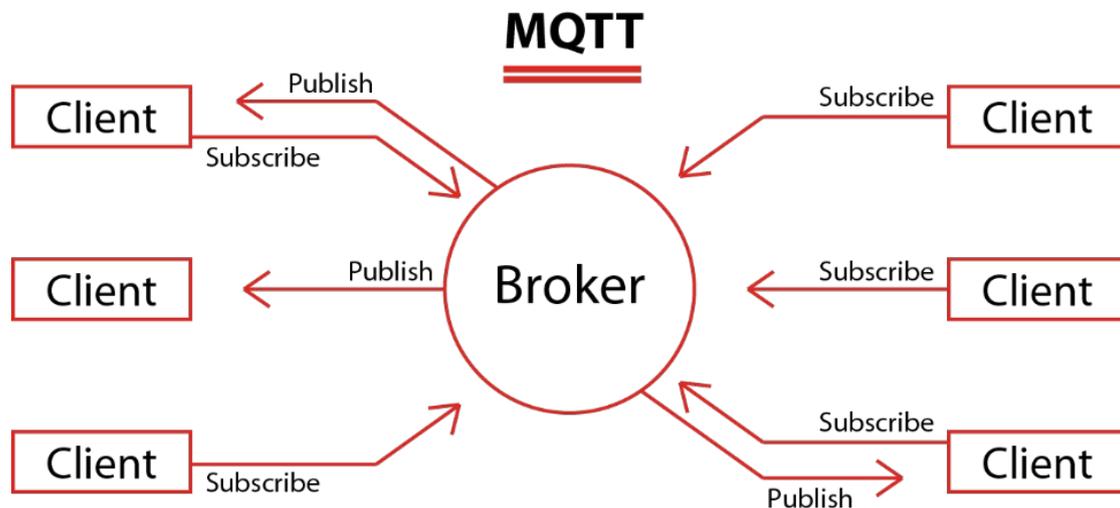


Figura 3.6. Arquitectura MQTT [45]

MQTT es un protocolo controlado por eventos, donde no hay transmisión de datos periódica o continua.

### 3.3.2.2 Tópicos MQTT

Los mensajes MQTT se organizan en tópicos. Un tópico está compuesto por uno o más niveles. Los temas y las estructuras de temas son uno de los principales puntos de diseño de MQTT. A diferencia de los buses de mensajes más 'empresariales', en MQTT los temas no están registrados previamente. Los clientes MQTT pueden suscribirse o cancelar la suscripción a los tópicos. El remitente de un mensaje (mecanismo de publicación) se encarga de definir el contenido y el tópico al enviar el

mensaje. Un bróker MQTT puede tener millones de tópicos consiguiendo una estructura ligera y dinámica.

A la hora de suscribirse a un tópico se deben tener en cuenta los siguientes dos comodines.

- # - Comodín multinivel. Permite acceder a todos los subtópicos del tópico al que se suscribe. El suscriptor estará suscrito a todo lo que esté después de este carácter, evitando suscribirse a cada uno de los tópicos.
- + - Comodín de un solo nivel. Solamente coincide hasta la siguiente barra invertida '/'. Puede ir situado en la mitad o al final del tópico.



*CAPÍTULO 4*

**DESCRIPCIÓN DE LA SOLUCIÓN  
PROPUESTA**

---



## 4 Descripción de la solución propuesta

Este capítulo describe la solución propuesta para la integración de un activo de fabricación, una estación de ensamblado, en un sistema de fabricación flexible, implementado mediante una arquitectura inteligente basada en agentes industriales llamada FlexManSys. La solución presentada es conforme al paradigma de la Industria 4.0, por lo que el activo se integra en FlexManSys como un Componente I4.0.

Por un lado, los agentes son entidades software autónomas capaces de competir y colaborar con otros agentes para lograr sus objetivos, lo cual les dota de la capacidad para tomar decisiones y negociación necesaria para la implementación de sistemas de fabricación flexibles. Los agentes industriales son agentes que, además, son capaces de integrar assets físicos. Es por ello que suelen emplearse para la implementación de AASs. Este es el caso de la arquitectura de FlexManSys, donde todas las máquinas de una planta (assets físicos) están supervisadas por su correspondiente agente máquina (AAS basado en agentes), capaz de interactuar con el resto de agentes de la arquitectura para tomar decisiones sobre los servicios que la máquina debe realizar. Para esta toma de decisiones el agente máquina debe conocer el estado de las tareas que el asset físico esté llevando a cabo.

Por otro lado, el asset físico es el responsable de realizar los servicios de fabricación que la arquitectura inteligente considere necesarios en cada momento. Y también es el encargado de informar acerca de la evolución de las tareas solicitadas. En este trabajo, el asset físico se corresponde con una estación de ensamblado compuesta de un PLC y un robot colaborativo. Este último será el que realice los servicios de fabricación relacionados con el ensamblaje de piezas.

En este contexto, y tal y como se muestra en la Figura 4.1, el agente máquina (AAS) representa las capas superiores de la arquitectura de integración propuesta en este trabajo (capa de inteligencia y gestión), mientras que la estación de ensamblado (asset) se corresponde con las capas inferiores (capas operativas y funcional). A pesar de la capacidad de los agentes industriales para integrar assets físicos, esta integración

no es directa, siendo necesaria una capa intermedia para la integración AAS-asset. Esta capa intermedia debe adaptar los mecanismos de comunicación propios de agentes a las interfaces de comunicación de la estación de ensamblado. Con el objetivo de desacoplar al máximo a los dos extremos de la comunicación, AAS y asset, este trabajo propone el uso del paradigma publicista/suscriptor, basado en la existencia de un bróker a través del cual se intercambian información en forma de tópicos. En concreto, se ha hecho uso del protocolo MQTT.

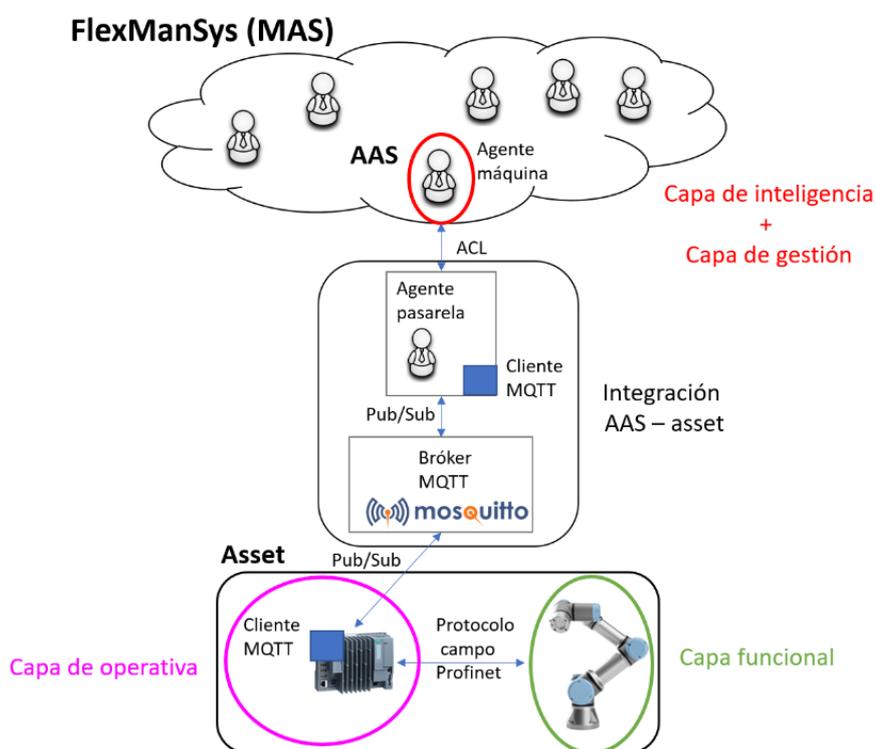


Figura 4. 1. Representación del proyecto

Como se muestra en la figura anterior, existe relación entre las capas de la arquitectura de 4 capas y los elementos del Componente I4.0. La capa de inteligencia es la encargada de gestionar las interacciones y procesas las peticiones de servicio y la capa de gestión se encarga de gestionar los submodelos que representan el activo. Teniendo en cuenta eso, estas dos primeras capas están directamente relacionadas con el agente máquina. En cambio, al hacer referencia a las capas definidas en el activo se dispone de la siguiente diferenciación. Por un lado, se dispone de la capa operativa que se sitúa en el PLC, encargado de recibir las peticiones y realizar la llamada del programa del robot parametrizándolo. Por último, se dispone de la capa funcional situada en el robot UR3e, siendo el encargado de efectuar los servicios solicitados.

Es importante señalar que el concepto de agente máquina, es decir, su funcionamiento e interfaz dentro de FlexManSys, es único y común a todos los asset que puedan existir en la planta. Por lo tanto, estos agentes sólo soportan el protocolo de comunicación propio de agentes, no siendo compatible con el paradigma publicista/suscriptor. Es por ello que resulta necesario incluir un nuevo elemento en la arquitectura de integración propuesta: el llamado agente pasarela, encargado de convertir el protocolo de comunicación usado por el agente máquina a MQTT.

#### 4.1 AAS de la estación de ensamblado

El AAS del Componente I4.0 correspondiente a la estación de ensamblado se implementa mediante un agente máquina. Para su desarrollo se ha hecho uso de Java como lenguaje de programación, siendo programado en IntelliJ IDEA [46], un Entorno de Desarrollo Integrado (Integrated Development Environment, IDE) que se utiliza para el desarrollo de programas informáticos. A la hora de definir el funcionamiento de un agente es necesario basarse en JADE (*Java Agente Development Framework*), un software que facilita el desarrollo de sistemas multiagentes inteligentes interoperables. Asimismo, se trata de una implementación completa del *estándar Foundation Intelligent Physical Agents (FIPA)*, en el lenguaje Java. FIPA promueve la tecnología necesaria y establece estándares para la interacción y desarrollo de sistemas basados en agentes inteligentes.

La estructura de los mensajes enviados por un agente se basa en *Agent Communication Language* (lenguaje ACL) definido por FIPA [47]. Los mensajes enviados mediante este lenguaje son de confianza, es decir, los mensajes están bien formados y llegan correctamente a su destino. Además, son fiables y ordenados, consiguiendo que los mensajes se reciban tal y como se envían. Dentro de los mensajes ACL existen diferentes tipos dependiendo de la función que se les asignen. En este proyecto se ha hecho uso de tres tipos de mensajes:

- **INFORM** (información). Dar información al receptor.
- **REQUEST** (solicitud). Solicita al receptor la ejecución de una acción.
- **CONFIRM** (confirmación). El emisor confirma al receptor la validez de una proposición.



necesaria para poder comenzar con el servicio del proceso de ensamblaje que debe realizar el robot de la estación. Dicha información se guarda en una estructura de datos definida como HashMap en JADE, permitiendo generar una estructura de datos que permite enviar la información de manera estructurada al resto de componentes del proyecto.

El primer elemento en recibir la estructura de datos es el agente pasarela, el cual recibe dichos datos en forma de solicitud. Una vez le llegue la confirmación desde el agente máquina, este queda bloqueado hasta que los mensajes que indican la evolución del proceso (mensajes informativos) lleguen desde el agente pasarela.

La integración del Componente I4.0 precisa una buena comunicación entre los dispositivos que lo forman. Por ello, definir una buena estructura de comunicación entre cada uno de los dispositivos es primordial. En el diagrama de secuencia que se muestra en la Figura 4.3, se recopila el flujo de datos requerido entre el AAS y el asset de la estación de ensamblado. También se muestran los datos que se comparten entre los dos dispositivos que componen el asset (PLC y UR3e).

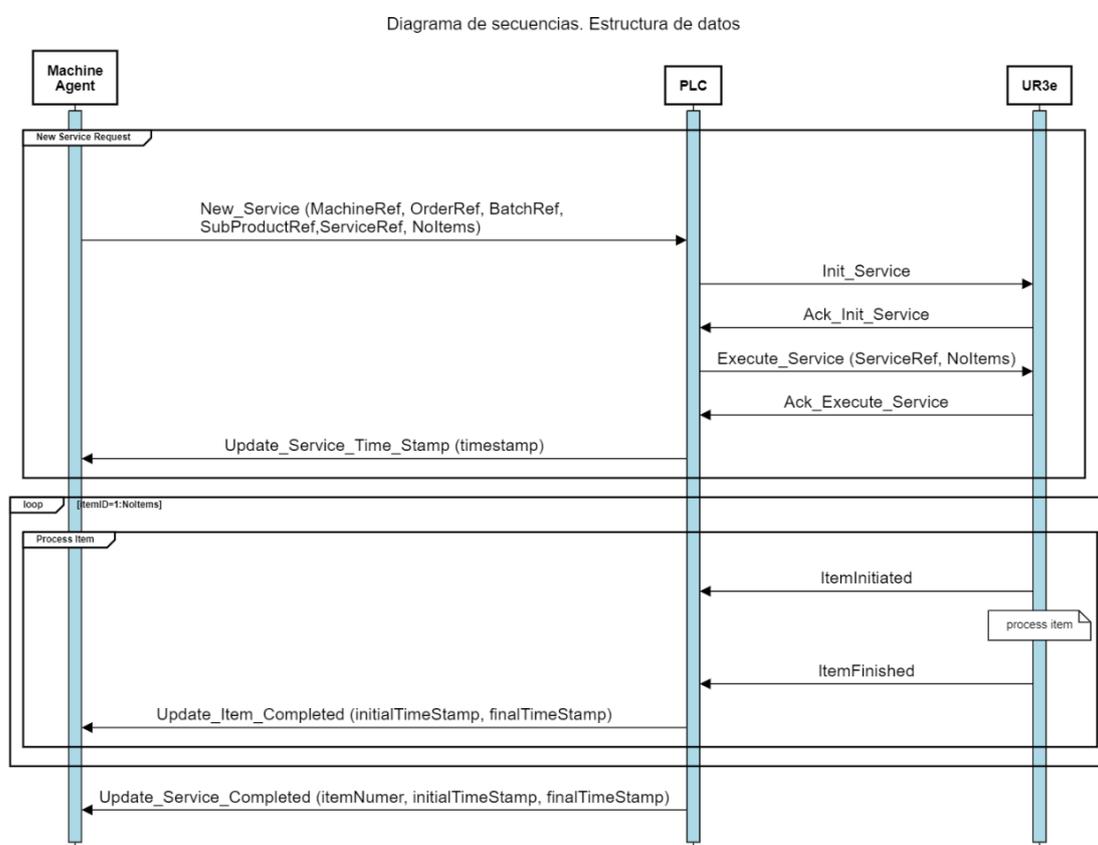


Figura 4. 3. Diagrama de secuencia de la realización de un servicio

Como se muestra en la figura, el comienzo del proceso es responsabilidad del agente máquina. Este es el encargado de recopilar la información necesaria del servicio de fabricación a solicitar y enviarla hasta el PLC. Para ello, hace uso de un agente pasarela y un bróker, los cuales trabajan como puente entre el AAS y el asset. Una vez los datos lleguen al PLC, este se encarga de que la solicitud llegue adecuadamente hasta la controladora del robot. Durante la ejecución del servicio solicitado, el robot se encarga de enviar datos sobre la evolución del proceso al PLC, el cual es responsable del envío de los datos recopilados durante el servicio para que lleguen adecuadamente hasta el agente máquina mediante mensajes informativos.

En el siguiente subapartado se define la estructura de datos utilizada a la hora de intercambiar información entre el AAS y el asset, mostrando en diferentes tablas los datos necesarios para el correcto funcionamiento.

#### 4.1.1 Estructura de datos intercambiadas entre el AAS y el asset

La información intercambiada entre el AAS y el asset se puede dividir en 4 grupos diferentes: Control, Operation, Id y Data. La información recopilada en el grupo de Control hace referencia a la solicitud de un nuevo servicio y la evolución del mismo (tanto a nivel de servicio, como de ítem). En el grupo de Operation se recopila la información del servicio a realizar. En el caso de Id, se recopila la información con la que se identifica el proceso. Por último, en lo que respecta a Data, se guardan los datos relacionados con los tiempos obtenidos durante el proceso de ensamblaje.

La Tabla 4.1 hace referencia a los datos bidireccionales que se agrupan dentro del grupo Control. En esta estructura de datos se recopila la solicitud de un nuevo servicio y los indicadores que muestran cuándo un ítem o el servicio solicitado ha finalizado.

Tabla 4.1 . Parámetros de control de intercambio de información (lectura/escritura)

Nombre	Tipo	Descripción
<b>Control.Flag_New_Service</b>	BOOL	Solicitud de nuevo servicio
<b>Control.Flag_Item_Completed</b>	BOOL	Indicador de ítem completado
<b>Control.Flag_Service_Completed</b>	BOOL	Indicador de servicio completado

Al definirse como variables bidireccionales pueden ser escritas tanto por el agente máquina como por el robot o PLC. En el caso de Flag\_New\_Service, será escrita por el agente y leída por el PLC o el robot. En el caso de las otras dos variables (Flag\_Item\_Completed y Flag\_Service\_Completed) puede ser tanto el agente máquina como el PLC el que escriba sobre ellas. En el caso de que las escriba el agente, el PLC leerá su valor. Si el PLC escribe sobre estas dos variables, el agente máquina será el que se encargue de leerlas.

En relación con las variables unidireccionales utilizadas en el proyecto, existen dos sentidos de comunicación. Por un lado, se han definido las variables que contienen los datos que son enviados desde el agente máquina al PLC. Dichas variables se muestran en la Tabla 4.2. Dentro de este grupo se definen dos tipos de variables. Por un lado, están las relacionadas con la identificación del proceso (Id). Por otro lado, las variables relacionadas con el servicio solicitado (Operation).

Tabla 4. 2. Datos enviados del agente máquina al PLC

Nombre	Tipo	Descripción
<b>Id.Machine_Reference</b>	UDINT	Referencia de la máquina
<b>Id.Order_Reference</b>	UDINT	Referencia del pedido
<b>Id.Batch_Reference</b>	UDINT	Referencia del lote
<b>Id.Ref_Subproduct_Type</b>	UDINT	Referencia del subproducto
<b>Operation.Ref_Service_Type</b>	UDINT	Referencia del servicio (operación)
<b>Operation.No_of_Items</b>	USINT	Número de elementos del servicio

Por último, se definen las variables que se deben enviar desde el PLC al agente máquina. En el caso de las variables de identificación (Id), al trabajar con la misma solicitud de ensamblaje, estas variables tienen el mismo valor durante todo el proceso

de ensamblaje. Por ello, una vez que las variables de identificación llegan desde el agente máquina al PLC se copian sobre las variables de identificación que se envían desde el PLC al agente máquina. El resto de las variables van actualizando su valor a medida de la evolución del proceso de ensamblaje. El conjunto de variables se observa en la Tabla 4.3, donde se recopilan las variables de identificación y las variables que recogen los tiempos de ensamblaje.

Tabla 4. 3. Datos enviados del PLC al agente máquina

Nombre	Tipo	Descripción
<b>Id.Machine_Reference</b>	UDINT	Referencia de la máquina
<b>Id.Order_Reference</b>	UDINT	Referencia del pedido
<b>Id.Batch_Reference</b>	UDINT	Referencia del lote
<b>Id.Ref_Subproduct_Type</b>	UDINT	Referencia del subproducto
<b>Id.Ref_Service_Type</b>	UDINT	Referencia del servicio
<b>Id.Item_Number</b>	USINT	Número del ítem completado
<b>Data.Initial_Time_Stamp</b>	LDT	TimeStamp de inicio del ítem
<b>Data.Final_Time_Stamp</b>	LDT	TimeStamp de final del ítem
<b>Data.Service_Time_Stamp</b>	LDT	TimeStamp de inicio del servicio

Considerando las tablas anteriores, se ha generado un bloque de datos en el programa de control del PLC desarrollado en TIA Portal (véase en la Figura 4.4), donde se consigue recopilar todos los datos con los que se va a trabajar durante el proceso de ensamblaje. Mediante estructuras se consigue definir tanto los datos bidireccionales como los unidireccionales, diferenciando cada uno de los sentidos en el caso de los unidireccionales.

AASassetDb		
	Nombre	Tipo de datos
1	Static	
2	Control	Struct
3	Flag_New_Service	Bool
4	Flag_Item_Completed	Bool
5	Flag_Service_Completed	Bool
6	Agent2PLC	Struct
7	Id	Struct
8	Machine_Reference	UDInt
9	Order_Reference	UDInt
10	Batch_Reference	UDInt
11	Ref_Subproduct_Type	UDInt
12	Operation	Struct
13	Ref_Service_Type	UDInt
14	No_of_Items	USInt
15	PLC2Agent	Struct
16	Id	Struct
17	Machine_Reference	UDInt
18	Order_Reference	UDInt
19	Batch_Reference	UDInt
20	Ref_Subproduct_Type	UDInt
21	Ref_Service_Type	UDInt
22	Item_Number	USInt
23	Data	Struct
24	Initial_Time_Stamp	LDT
25	Final_Time_Stamp	LDT
26	Service_Time_Stamp	LDT

Figura 4. 4. Bloque de datos AASassetDB declarado en TIA Portal

## 4.2 Integración AAS – asset de la estación de ensamblado

Este subapartado del capítulo se centra en la comunicación entre los elementos del Componente I4.0, el AAS y el asset, que precisa un agente pasarela y un bróker MQTT. Por tanto, se comienza con una explicación sobre el diseño del agente pasarela, el cual se utiliza para que los datos procedentes del agente máquina lleguen hasta el bróker. Posteriormente, se presenta el bróker MQTT, basado en la comunicación mediante publicista/suscriptor, en el cual se organizan los datos en tópicos.

### 4.2.1 Diseño del agente pasarela

El segundo agente que se utiliza en el proyecto es el agente pasarela, el cual es el encargado de comunicar el agente máquina con el resto de los componentes del

proyecto. Este agente, como indica su nombre, es capaz de mandar y recibir mensajes ACL actuando de puente con otros protocolos. En este proyecto, el agente pasarela hace de intermediario entre el protocolo de comunicación de los agentes y el protocolo de comunicación MQTT.

Siguiendo la misma estructura a la hora de definir un agente, se comienza definiendo el comportamiento de dicho agente. En este caso, se dispone de un agente pasarela con comportamiento cíclico. Su funcionamiento se basa en enviar la estructura de datos procedente del agente máquina al bróker y viceversa. Lo que significa que debe estar en todo momento en funcionamiento, a la espera de que lleguen mensajes por alguno de los dos sentidos de transmisión.

El agente pasarela diseñado dispone de dos modos de trabajo. Por un lado, se define el funcionamiento cuando recibe información desde el agente máquina. Por el otro, se define el funcionamiento cuando recibe datos desde el bróker. En referencia al primero de los funcionamientos, una vez recibe una estructura de datos procedente del agente máquina, se encarga de enviar un mensaje de confirmación, permitiendo asegurar que la recepción de datos se ha realizado adecuadamente. Tras ello, procesa los datos y realiza el envío de la estructura de datos, consiguiendo que lleguen correctamente hasta el bróker MQTT. En cambio, si los datos provienen del bróker, el agente pasarela se debe encargar de procesarlos y enviarlos hasta el agente máquina. En este segundo caso los datos enviados se definen como informativos.

Para poder realizar una comunicación publicista/suscriptor es necesario hacer uso de un cliente MQTT, el cual es el encargado de publicar los datos procedentes del agente máquina y de suscribirse al tópico necesario para poder recibir la información procedente del asset. Por tanto, dentro del agente pasarela se debe generar un cliente MQTT que haga el trabajo de publicar en el tópico indicado o suscribirse al tópico del cual se debe conseguir la información del proceso de ensamblaje.

#### **4.2.2 Conexión con el bróker MQTT**

La conexión entre el AAS y el asset también requiere el uso de un bróker, permitiendo que los datos procedentes del agente lleguen hasta el asset pasando por él. En este proyecto se ha hecho uso de Mosquitto como bróker MQTT, el cual permite que la estructura de datos llegue desde el agente máquina hasta el PLC y viceversa.

Eclipse Mosquitto [48] es un bróker MQTT de código abierto. Debido a su ligereza, es posible su funcionamiento en dispositivos de baja potencia. El bróker tiene la funcionalidad de recibir los mensajes de ambos sentidos, siendo el encargado de gestionar los tópicos (temas). Las funciones que cumple el bróker utilizado durante el proyecto son las siguientes:

- Aceptar la solicitud de conexiones del cliente MQTT.
- Recibir mensajes de un cliente MQTT.
- Aceptar las solicitudes de suscripción de los clientes MQTT.
- Reenviar mensajes recibidos a los clientes MQTT atendiendo a las suscripciones.

Para poder utilizar el bróker en el proyecto, es necesario su previa configuración. En este caso, se han realizado dos cambios en el archivo de configuración del bróker. El primero de ellos está relacionado con la definición de la dirección IP y el puerto en el que se encuentra el bróker (véase en la Figura 4.5). Para este proyecto se le ha asignado la dirección 192.168.2.151 y el puerto 1883.

```
# listener port-number [ip address/host name/unix socket path]
listener 1883 192.168.2.151
```

Figura 4. 5. Definición de la dirección IP y el puerto del bróker

El segundo cambio de la configuración está relacionado con las conexiones. En este proyecto se trabaja con dispositivos definidos dentro del PC local y dispositivos que están fuera del PC, por ello es necesario permitir el acceso al bróker desde dispositivos que se encuentran fuera del PC local (véase en la Figura 4.6).

```
# Defaults to false, unless there are no listeners defined in the configuration
# file, in which case it is set to true, but connections are only allowed from
# the local machine.
allow_anonymous true
```

Figura 4. 6. Permitir acceso al bróker de dispositivos externos

La conexión con el bróker se hace en dos sentidos. Por un lado, existe la conexión del bróker y el agente pasarela, la cual permite que los datos enviados desde el agente máquina lleguen hasta el bróker y viceversa. Por otro lado, existe la conexión entre el bróker y el asset. En este segundo caso, se consigue que los datos del bróker lleguen hasta el PLC y los datos del robot lleguen hasta el bróker pasando por el PLC. Por tanto,

se consigue una comunicación en ambas direcciones teniendo el bróker como intermediario.

Cada vez que se quiera realizar la conexión con el bróker es necesario el uso de un cliente MQTT, por ello, en este proyecto es necesario el uso de dos clientes MQTT (véase en la Figura 4.1). El primero de ellos se define en el programa de control del PLC y el segundo en el agente pasarela.

En el caso del PLC, el desarrollo del sistema de control se ha hecho con la herramienta TIA Portal V16. Esta herramienta permite el uso de librerías, facilitando tanto la comunicación como la manipulación de las estructuras de datos utilizadas en el proyecto. El Cliente MQTT que establece la comunicación MQTT entre el bróker y el PLC es el encargado de realizar tanto la suscripción como las publicaciones con el bróker (véase en la Figura 4.7). Dicho cliente se define mediante un Bloque de Función (Function Block, FB) que permite el intercambio de datos con el bróker. Al comunicar mediante este protocolo surge la necesidad de hacer uso de la librería 'Libraries for Communication for SIMATIC Controllers', la cual dispone de los bloques necesarios para realizar la comunicación con el bróker, y entre ellos está el bloque del cliente MQTT.

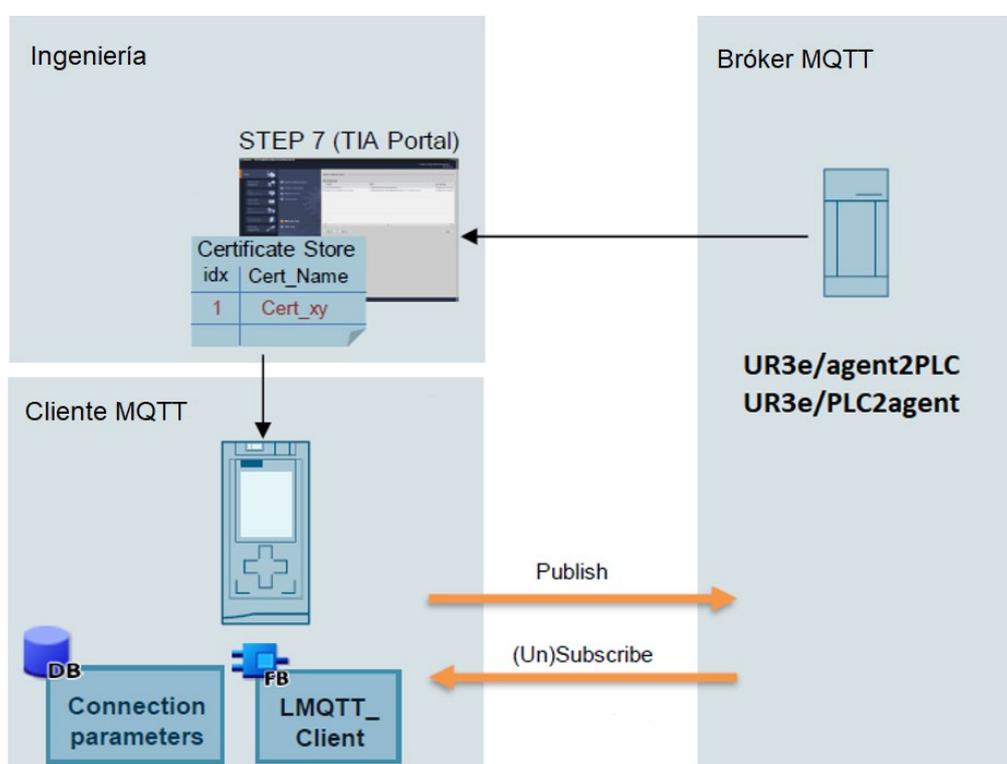


Figura 4. 7. Comunicación entre el PLC y el bróker MQTT

El Bloque de Función del Cliente MQTT es necesario instanciarlo en el programa generado en TIA Portal, consiguiendo la estructura que se muestra en la Figura 4.8. Una vez instanciado es necesario vincular cada una de sus salidas y entradas con una variable del programa. Para agrupar cada una de las variables relacionadas con la comunicación del cliente, se ha generado un Bloque de Datos (Data Block, DB) donde se recopilan todas las variables que se necesitan asignar para el bloque del Cliente MQTT.

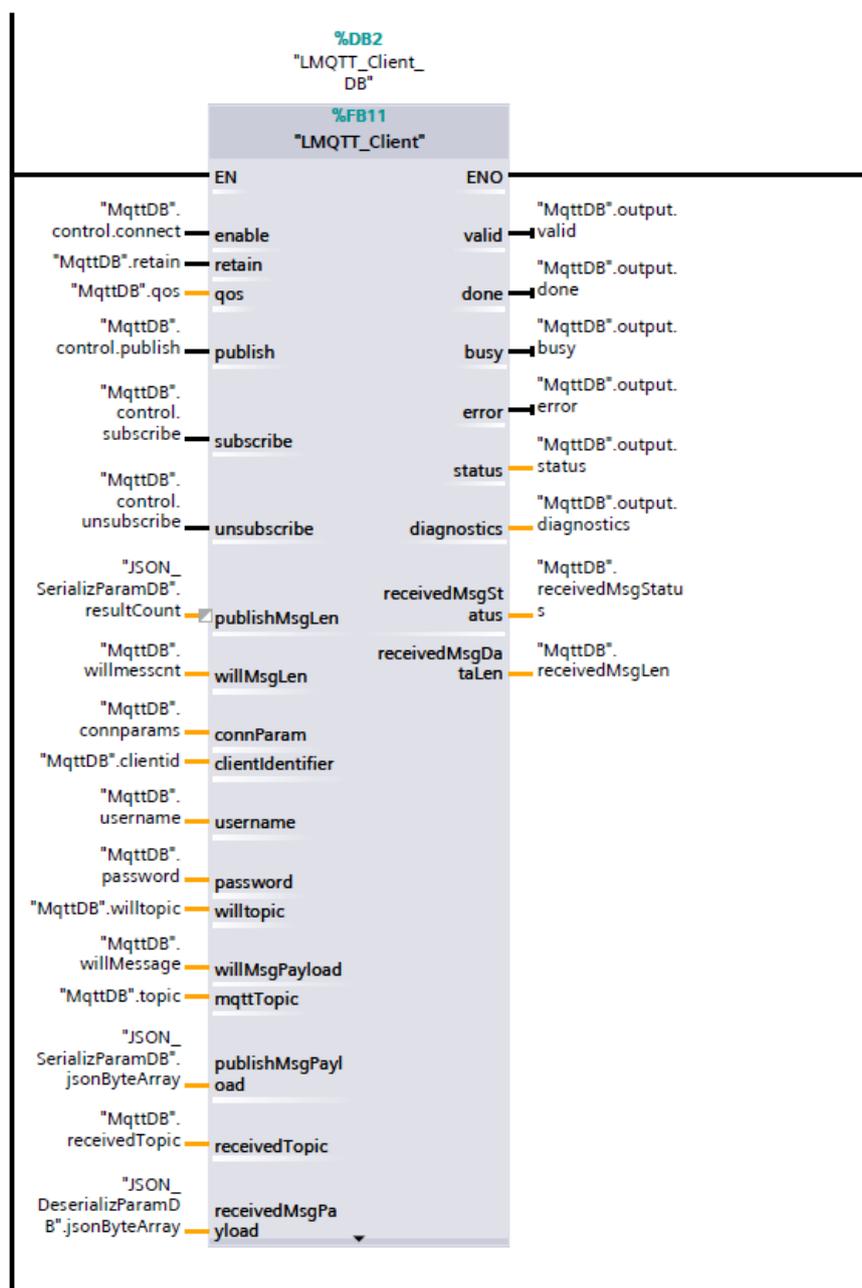


Figura 4. 8. Cliente MQTT (TIA Portal)

Este cliente permite tanto la suscripción a un tópico del bróker como la publicación de datos en él. Entre las variables que ofrece el bloque se encuentran las que recopilan los datos relacionados con la conexión, publicación y suscripción al bróker.

- **enable.** Parámetro de control que permite establecer y mantener la conexión con el bróker mientras su valor sea TRUE.
- **publish.** Con conexión existente al bróker permite publicar cada vez que haya un flanco positivo.
- **subscribe.** Con conexión existente al bróker permite suscribirse al tópico tras obtener un flanco positivo.
- **mqttTopic.** Tópico al que se desea publicar/suscribir/cancelar la suscripción.
- **publishMsgPayload.** Mensaje MQTT que se desea publicar en un tópico.
- **receivedMsgPayload.** Mensaje MQTT que se recibe al suscribirse en un tópico.

El segundo cliente MQTT se ha definido en el agente pasarela. Este cliente se encarga de publicar los datos necesarios en el bróker cuando estos provienen del agente máquina mediante el tópico 'UR3e/agente2PLC' (véase en la Figura 4.9). Este tópico define el robot con el que se trabaja y el sentido del envío de los datos. Es decir, define el envío de datos que se realiza desde el agente máquina hasta el PLC. En cambio, si se desea obtener los datos que provienen del PLC, es necesario realizar una suscripción a uno de los tópicos definidos en este proyecto, 'UR3e/PLC2agent'. En este segundo caso, el tópico define el sentido contrario del envío de datos, en otras palabras, precisa el envío de datos desde el PLC hasta el agente máquina. Mediante esta suscripción, cada vez que se escriba un nuevo dato en este tópico, los datos llegan hasta el cliente y se guardan en el agente pasarela para después poder enviarlos al agente máquina.

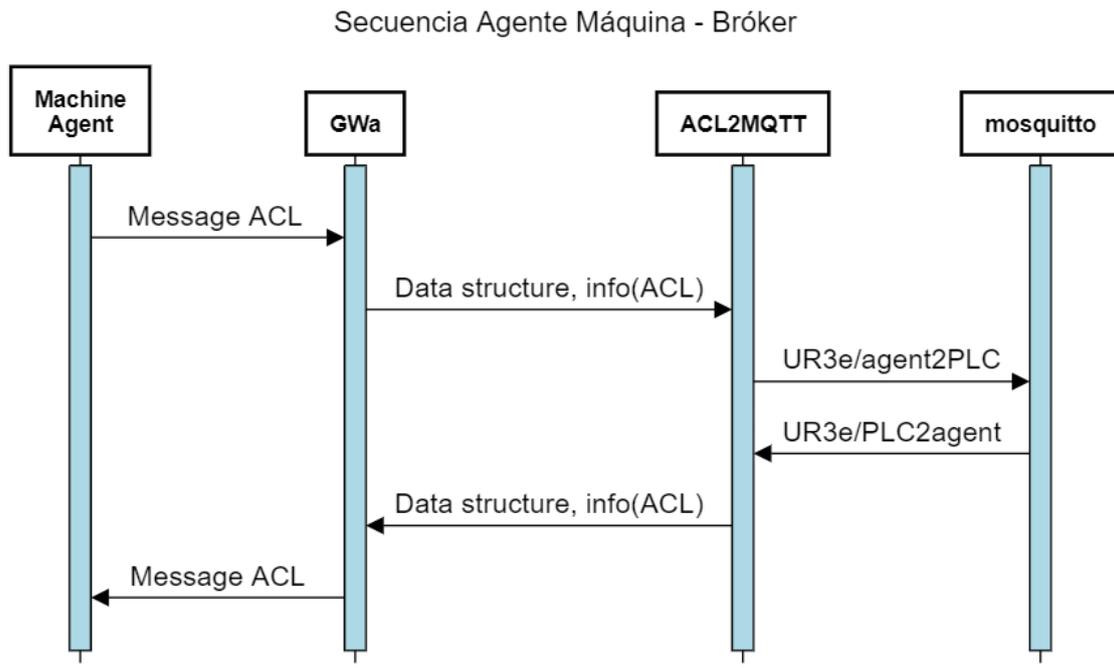


Figura 4. 9. Diagrama de secuencia para la comunicación entre el agente máquina y el bróker

En el caso del cliente MQTT desarrollado en el programa de control del PLC, tiene el mismo funcionamiento, pero definiendo los tópicos a la inversa (véase en la Figura 4.10). En el caso de querer recibir información desde el agente, el cliente se debe suscribir al tópico 'UR3e/agent2PLC'. Por el contrario, si se desea publicar los datos de la evolución del proceso, los datos se deben publicar en el tópico 'UR3e/PLC2agent'. En este caso se hace uso de dos tópicos, uno para cada sentido de la comunicación.

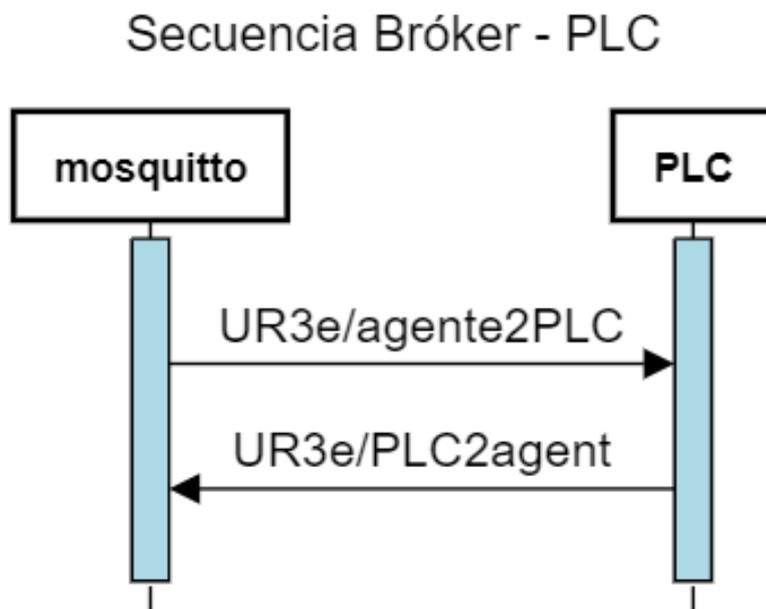


Figura 4. 10. Diagrama de secuencia para la comunicación del bróker y el PLC

### 4.3 Asset de la estación de ensamblado

El asset de la estación de ensamblado consta de dos dispositivos: el PLC y el robot. El PLC será responsable de obtener y procesar la información procedente del bróker. Una vez se haya procesado dicha información, debe enviar los datos necesarios a la controladora del robot. Para ello es imprescindible la conexión entre los dos dispositivos que componen el asset y la realización de un programa que vincule cada uno de los datos del proceso a realizar. La comunicación entre el PLC y el robot se ha efectuado mediante el protocolo Profinet. Este protocolo permite intercambiar datos de forma rápida y determinista.

Dado que el robot es capaz de comunicarse mediante diferentes protocolos (Modbus, Profinet y Ethernet/IP) es necesario especificar en la instalación del robot el protocolo que se va a utilizar durante la comunicación.

El robot de la estación de ensamblado es el encargado de llevar a cabo el servicio solicitado, por ello se necesita un conocimiento previo de las piezas con las que se trabaja y de la mesa de trabajo. El conocimiento de estos elementos permite entender los servicios a realizar, como sus situaciones iniciales y finales. Para finalizar con este apartado del capítulo se especifican tanto los servicios a realizar como el diseño de programación realizado para el robot.

#### 4.3.1 Piezas del ensamblaje

La estación de ensamblado ha sido diseñada para realizar el ensamblado de ítems. Dichos ítems están compuestos por varias piezas a colocar sobre las bases con la ayuda del robot disponible en la estación de trabajo. Las piezas que lo componen son las siguientes: base, rodamiento, bulón, tapa interior y tapa exterior (véase en la Figura 4.11).

A la hora de realizar los servicios, no es necesario utilizar todos los elementos que componen el conjunto, lo cual dependerá del servicio que se haya solicitado. Disponer de una variedad de servicios aumenta la flexibilidad de la estación, permitiendo actuar ante averías o paros por mantenimiento en colaboración con otra estación disponible de características similares.

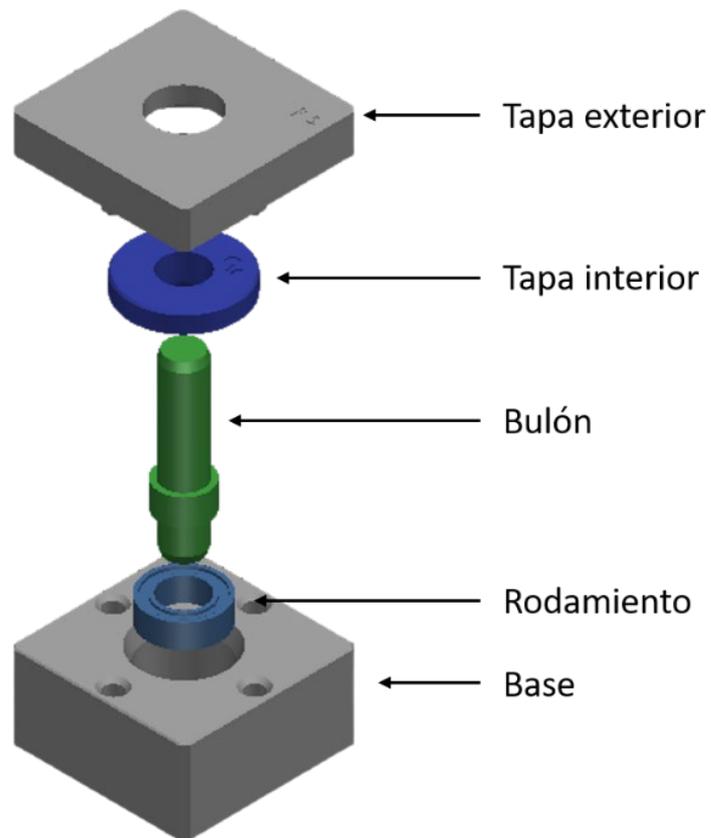


Figura 4. 11. Conjunto de piezas

Las bases sobre las cuales se ensamblan las piezas vienen ordenadas en un palé, el cual está formado por 6 posiciones y en las cuales se sitúan las bases (véase en la Figura 4.12).

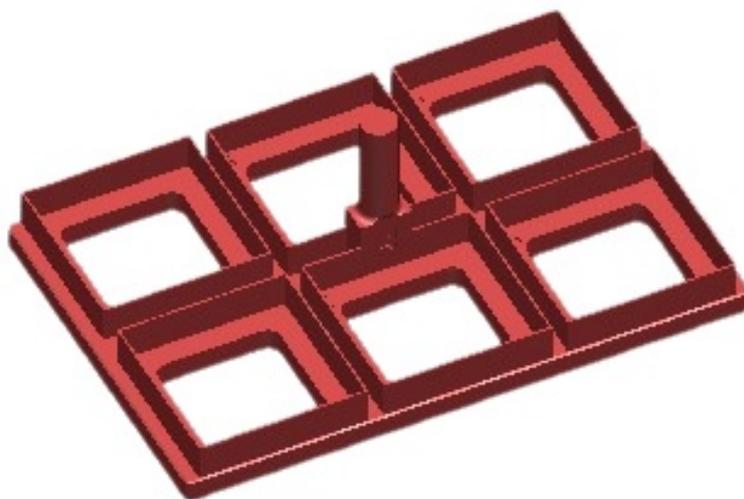


Figura 4. 12. Palé

### 4.3.2 Descripción de la mesa de trabajo

La estación de ensamblado está formada por una mesa de trabajo y el robot UR3e (véase en la Figura 4.13). El robot está situado en el punto más óptimo, consiguiendo que llegue a alcanzar todas las posiciones de la estación. Como se observa en la figura, la mesa está compuesta por 5 zonas (cada una de ellas identificada mediante un color diferente).

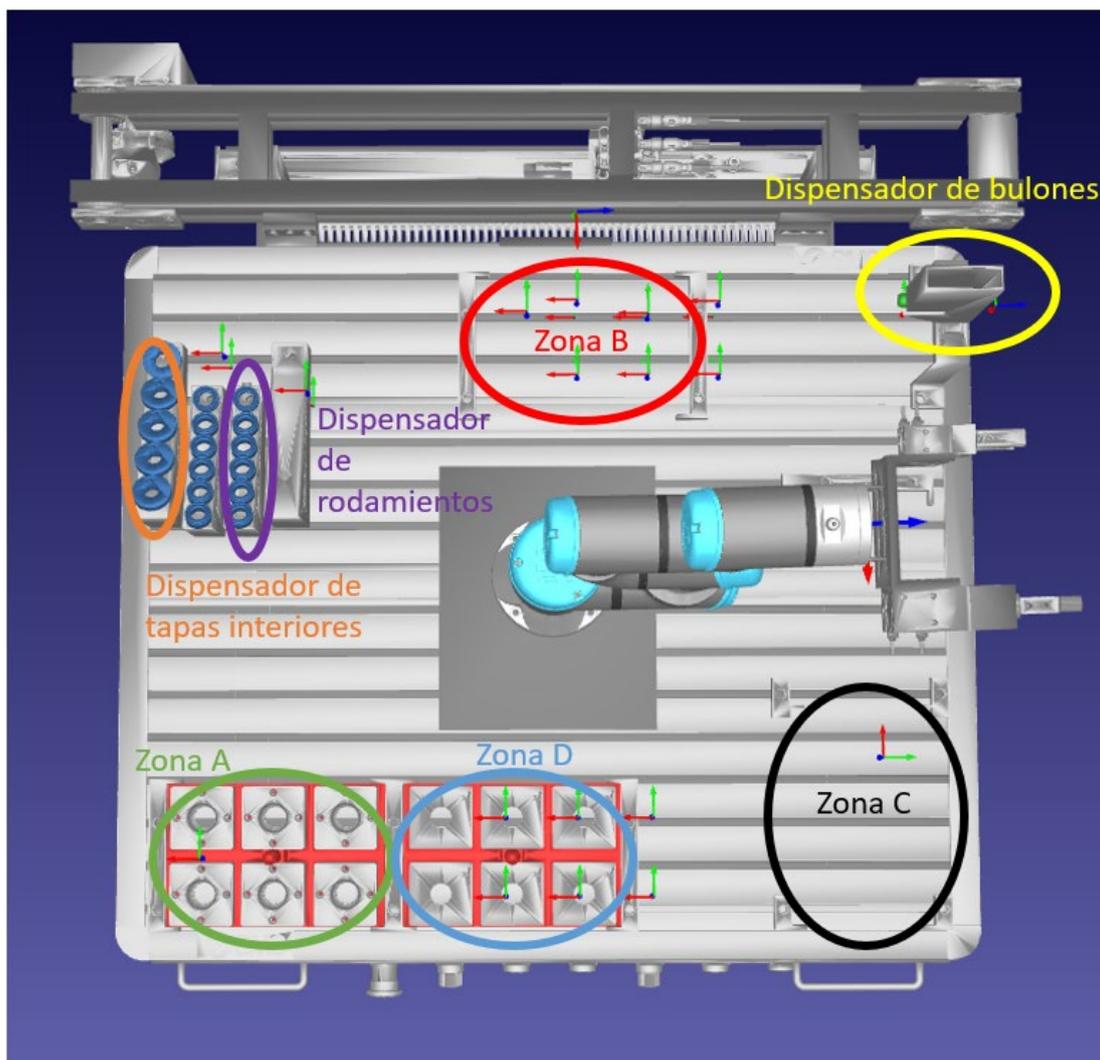
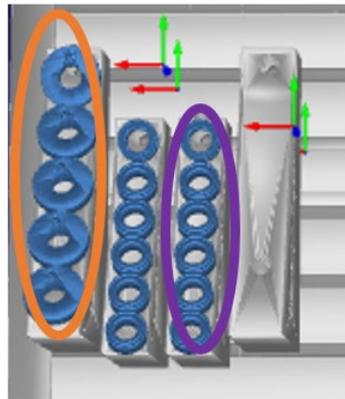


Figura 4. 13. Disposición de la mesa de trabajo

La zona A (verde) corresponde a la zona de entrada, es decir, el lugar en donde se sitúa el palé al inicio del servicio. Al comienzo de cada ensamblaje el robot debe coger el palé situado en esta zona y llevarlo hasta la zona de montaje (zona B, rojo). Tras realizar todas las fases que definen el servicio solicitado y finalizar los ítems, el robot debe transportar el palé a la zona de salida (zona C, negro), donde se depositan los palés

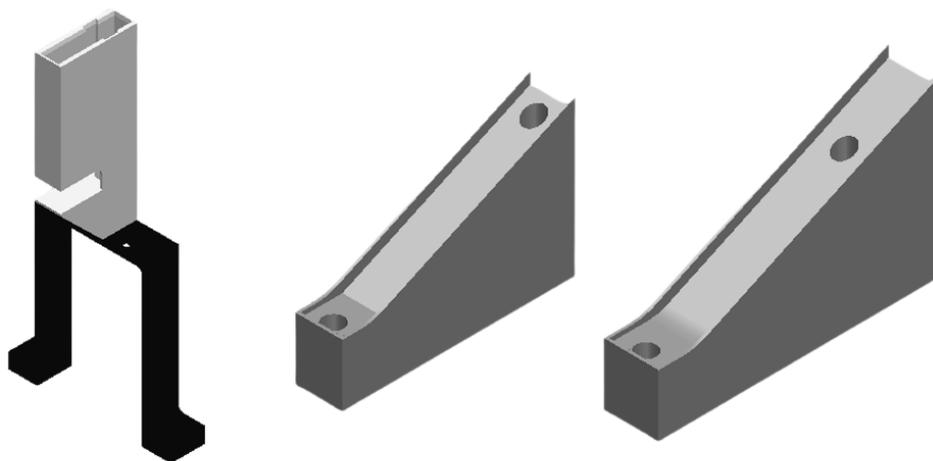
finalizados. En color amarillo se identifica el dispensador de bulones, en color morado el dispensador de rodamientos y en color naranja el dispensador de tapas interiores (véase en la Figura 4.14). Por último, se ha definido una zona de entrada de tapas exteriores (zona D, azul).



Rampa de tapas interiores  
Rampa de rodamientos

Figura 4. 14. Descripción de las rampas

Tanto las piezas del proceso como los elementos dispensadores instalados en la mesa de trabajo han sido generados mediante impresión 3D. Los elementos dispensadores consisten en dos rampas, una de ellas para dispensar los rodamientos (véase en la Figura 4.15b) y la otra para dispensar las tapas interiores (véase en la Figura 4.15c), y un dispensador de bulones (véase en la Figura 4.15a).



a) Dispensador de bulones      b) Rampa de rodamientos      c) Rampa de tapas interiores

Figura 4. 15. Elementos dispensadores de la mesa de trabajo

### **4.3.3 Descripción de los servicios**

El robot instalado en la mesa de trabajo ha sido programado para que manipule diferentes piezas situadas en la mesa, consiguiendo completar el servicio solicitado por el sistema multiagente. Cada operación a realizar se identifica como un servicio. El servicio que debe llevar a cabo el robot y la cantidad de ítems a completar llegan desde el agente máquina, pasando por el agente pasarela, bróker y el PLC.

Dependiendo del servicio que se haya solicitado, el robot debe colocar las piezas que corresponden sobre las bases o el conjunto de piezas de partida. En caso de un montaje completo, el conjunto consta de un rodamiento, un bulón, una tapa interior y una tapa exterior. No obstante, es posible, dependiendo del servicio solicitado, que solo se requiera el uso de algunos de estos elementos en el ensamblaje. Por ello, el servicio requerido indica los elementos que se deben ensamblar y las zonas de la mesa de trabajo sobre las cuales debe trabajar el robot. Es decir, dependiendo del servicio solicitado el robot hace uso de unas u otras zonas.

Cada uno de los servicios representa una situación del producto inicial y final. Basándose en las diferentes situaciones iniciales en las que se encuentra el conjunto, los servicios se clasifican en 4 grupos. En el primer grupo, el palé llegará con la base vacía. En el segundo grupo, el palé llegará con la base y el rodamiento montado. En el tercer grupo, el palé llega con la base con un rodamiento y un bulón montados. En el cuarto grupo, el palé llega con la base de la pieza a montar con un rodamiento, un bulón y la tapa interior pequeña puesta.

En la Tabla 4.4 se muestran todas las opciones posibles de servicios que se pueden realizar, comenzando desde el servicio 1 y finalizando en el 10. El trabajar con diferentes servicios ofrece mayor flexibilidad a la estación de ensamblado.

Tabla 4. 4. Conjunto de servicios proporcionados por el asset

SERVICIOS	SITUACIÓN DE ENTRADA	TAREAS A REALIZAR
1		Rodamiento
2		Rodamiento Bulón
3	Base	Rodamiento Bulón Tapa interior
4		Rodamiento Bulón Tapa interior Tapa exterior
5	Base + Rodamiento	Bulón
6		Bulón Tapa interior
7		Bulón Tapa interior Tapa exterior
8		Tapa interior
9	Base + Rodamiento + Bulón	Tapa interior Tapa exterior
10	Base + Rodamiento + Bulón + Tapa interior	Tapa exterior

A modo de ejemplo, en la Figura 4.16 se muestran dos posibles montajes finales. El montaje final de la izquierda (Figura 4.16a) se obtiene como resultado de la ejecución de los servicios 3, 6 y 8, correspondiendo también al montaje inicial del servicio 10. El montaje final de la derecha (Figura 4.16b) es el resultado de la ejecución de los servicios 4, 7, 9 y 10.

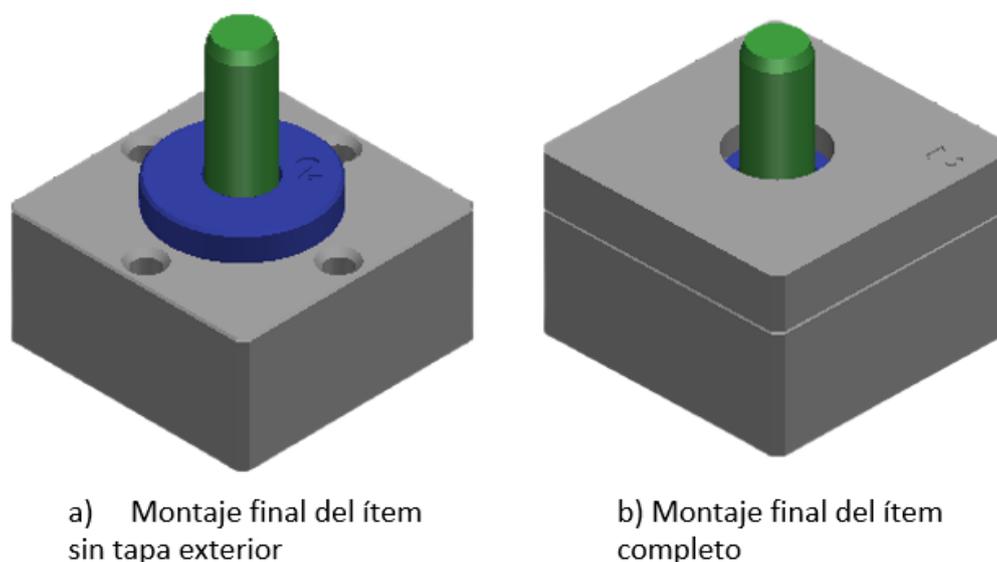


Figura 4. 16. Posibles montajes finales

El palé que se dispone en la zona de inicio estará siempre compuesto por las bases como mínimo. Esto no significa que en todas las solicitudes sea necesario trabajar sobre los seis ítems que componen el palé. Dependiendo de la solicitud, la cantidad de ítems a realizar puede variar entre uno y seis.

#### 4.3.4 *Diseño del programa del robot*

Con objeto de realizar la programación necesaria para que el robot pueda llevar a cabo todos los servicios, se ha utilizado un programa generado anteriormente en RoboDK (software de programación y simulación), el cual ha proporcionado información sobre los movimientos del robot en el proceso de ensamblaje de piezas.

La programación del robot sigue los principios de modularidad para dotar de mayor flexibilidad a la estación de ensamblado, tanto en producción normal como en situaciones excepcionales de producción, y permitir la reutilización de las distintas fases en las que se estructuran los servicios.

Con la solicitud de servicio procedente del sistema multiagente, el PLC envía tanto el servicio solicitado como la cantidad de ítems a realizar. El programa cíclico del robot evalúa los datos y ejecuta unas fases u otras según corresponda al servicio

solicitado. En la Tabla 4.5. se han destacado en color naranja las fases necesarias en cada servicio.

Tabla 4. 5. Fases requeridas para cada servicio

		Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6
<b>Caso 1</b>	Servicio 1						
	Servicio 2						
	Servicio 3						
	Servicio 4						
<b>Caso 2</b>	Servicio 5						
	Servicio 6						
	Servicio 7						
<b>Caso 3</b>	Servicio 8						
	Servicio 9						
<b>Caso 4</b>	Servicio 10						

Como se muestra en la tabla, los servicios necesitarán unas fases u otras, pero tanto la fase 1 como la fase 6 serán necesarias en todos los servicios. Estas dos fases hacen referencia a los movimientos que se deben realizar tanto al comienzo como al final de cada uno de los servicios.

En los siguientes puntos se proporciona una breve explicación de las acciones que se realizan en cada una de las fases. Todas las explicaciones están basadas en la Figura 4.14, donde se muestra la disposición de la mesa de trabajo.

- Fase 1 → El robot comienza desde la posición inicial, se mueve hasta la zona inicial del palé (zona A) donde coge el palé y lo lleva hasta la zona de montaje (zona B). Tras depositar el palé en la zona de montaje, el robot vuelve a la posición inicial.
- Fase 2 → El robot coloca un rodamiento en cada una de las bases que está colocada sobre el palé. Para ello, el robot se desplaza, desde la posición inicial, hasta la rampa en la cual están colocados los rodamientos (véase en la Figura 4.14, rampa de la derecha). Una vez coja un rodamiento en la pinza, la deposita en la primera base. Se realiza el mismo movimiento hasta dejar el último rodamiento en la última base que se haya solicitado. Finalmente, el robot vuelve a la posición inicial.
- Fase 3 → El robot se desplaza, desde la posición inicial, hasta el dispensador de bulones, del cual coge un bulón y lo deposita en el ítem inicial que le corresponde en ese momento. El robot realiza la misma secuencia de movimientos hasta completar todos los ítems que se han solicitado. Finalmente, el robot vuelve a la posición inicial.
- Fase 4 → El robot se desplaza, desde la posición inicial, hasta la rampa en la cual están las tapas interiores (rampa de la izquierda en la Figura 4.14), el robot coge el primer tapón interior y lo coloca en el ítem correspondiente del palé. Tras depositar el primer tapón, realiza la misma secuencia de movimientos hasta completar todos los ítems solicitados. Finalmente, el robot vuelve a la posición inicial.
- Fase 5 → El robot se desplaza, desde la posición inicial, a la zona en la cual está el palé de las tapas exteriores (zona D). En ella coge una de las tapas y la deposita en el ítem correspondiente del palé que está situado en la zona de montaje (zona B). Estas tapas se irán colocando una a una en orden, comenzando por el primer ítem y finalizando en el último que se haya solicitado. Finalmente, el robot vuelve a la posición inicial.
- Fase 6 → Esta última fase corresponde al final del montaje. El robot se desplaza, desde la posición inicial, hasta la zona de montaje, coge el palé y lo deja en la zona final (zona C). Finalmente, el robot vuelve a la posición inicial.

En la Figura 4.17 se muestra el orden que sigue el robot a la hora de realizar el montaje de los ítems. Siempre comenzará desde el ítem 1 y finalizará en el ítem que se defina en el pedido de servicio, como máximo 6 ítems por cada servicio solicitado.

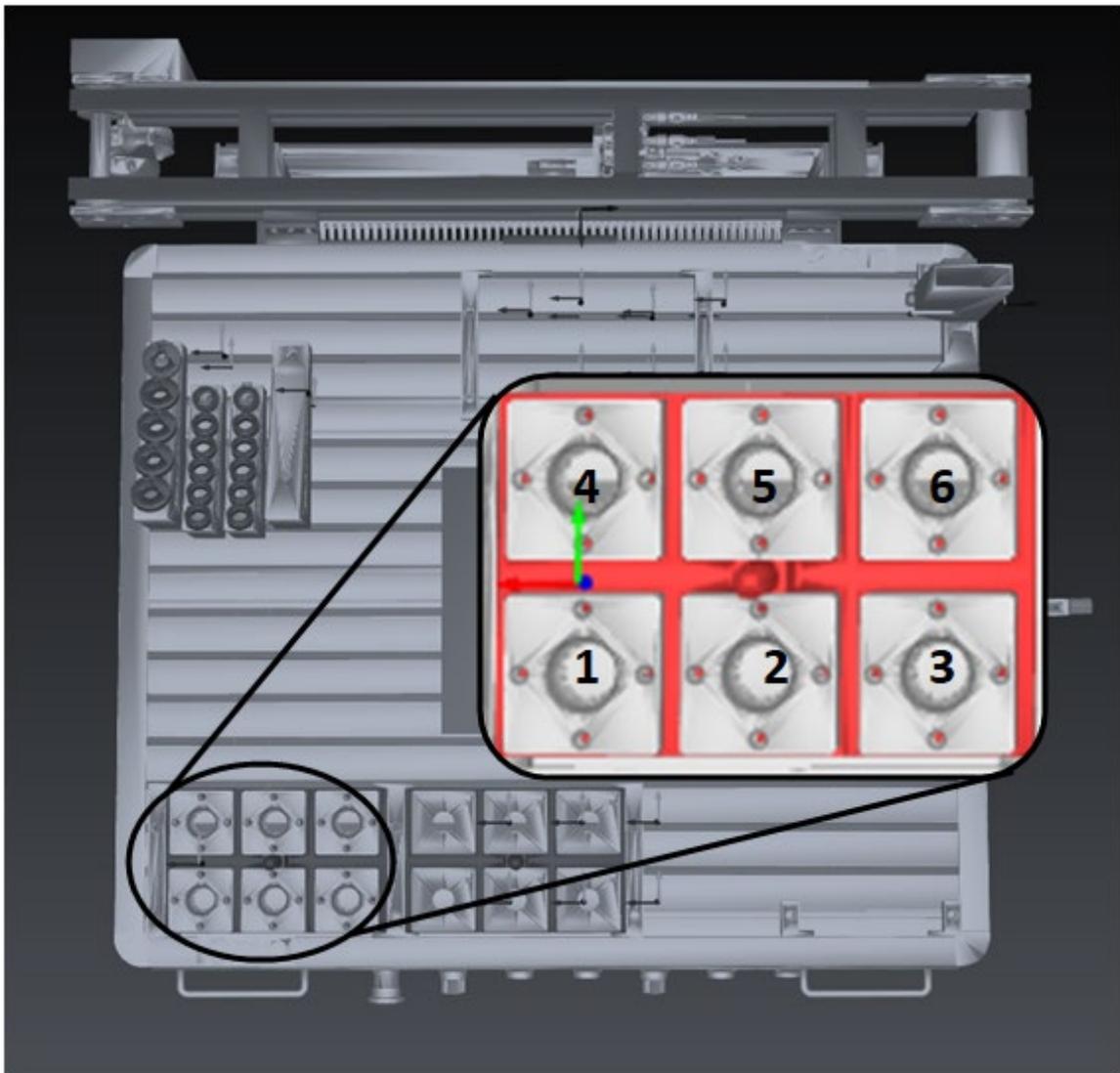


Figura 4. 17. Posición de los ítems sobre el palé



*CAPÍTULO 5*

# **PRUEBAS Y RESULTADOS**

---



## 5 Pruebas y resultados

En este capítulo se describen las pruebas realizadas y los resultados que se han obtenido. Dada la variedad de dispositivos y tecnologías presentes en este proyecto, se ha realizado un desarrollo escalonado, comenzando por el funcionamiento de cada dispositivo y/o tecnología para después realizar una integración ordenada de los mismos. Por ello, este capítulo se organiza en tres subapartados. Un primer apartado que aborda las pruebas unitarias de los dispositivos y las pruebas de conexión entre dispositivos para validar las tecnologías empleadas. Un segundo apartado referente a las pruebas de integración con estrategia bottom-up para comprobar que la comunicación vertical en ambos sentidos proporciona una transferencia de datos correcta. Un tercer apartado con las pruebas de verificación y validación del Componente I4.0 de la estación de ensamblado.

En la Figura 5.1 se presenta el escenario de pruebas. Por un lado, se tienen el bróker (Mosquitto) y los agentes (máquina y pasarela) dentro del PC local. Por otro lado, se encuentra el PLC físico, con el cual se realiza el control del robot UR3e.

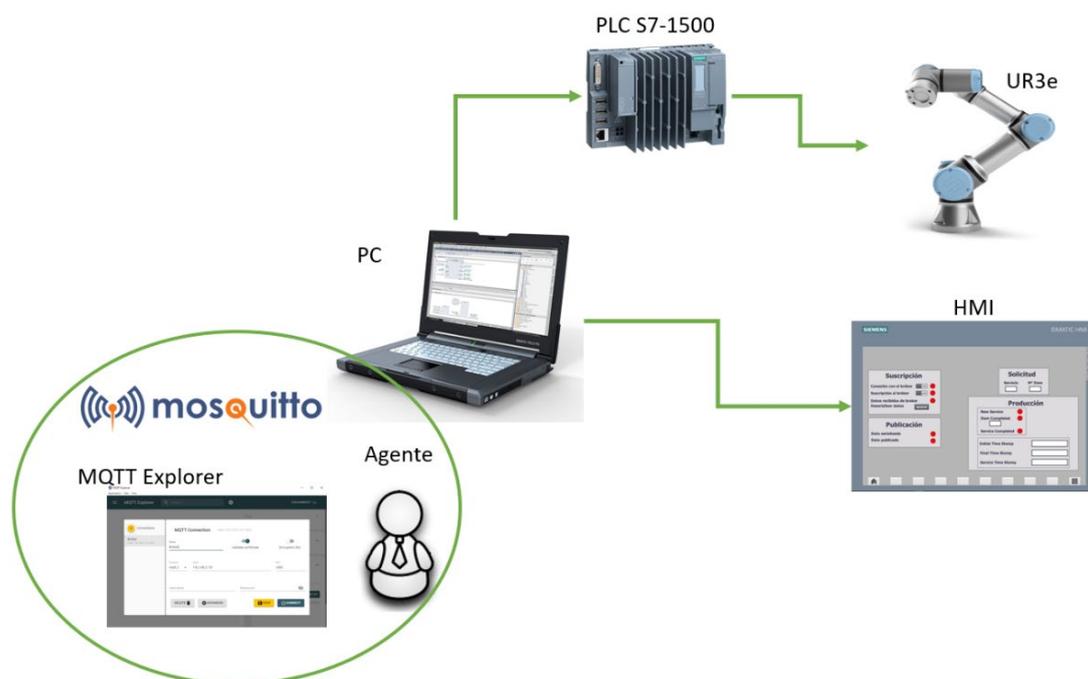


Figura 5. 1. Entorno de pruebas

Como se puede observar, en el escenario de pruebas intervienen dos aplicaciones que se ejecutan en el PC local. Por un lado, se hace uso del software MQTT Explorer que permite visualizar los datos que llegan al bróker en ambos sentidos. Por otro lado, se ha diseñado un HMI que permite controlar las variables del proceso y mostrar su evolución.

## **5.1 Pruebas unitarias**

Las pruebas unitarias hacen referencia a cada una de las pruebas que se han realizado durante el proyecto con los dispositivos y la conexión entre dispositivos. Por ellos, este subapartado se ha dividido en las pruebas realizadas en relación con la programación del robot, pruebas de conexión del PLC y el bróker, pruebas de conexión del PLC y el robot, pruebas de conexión entre el agente pasarela y el agente máquina, y pruebas de conexión entre el bróker y el agente pasarela.

### **5.1.1 Pruebas unitarias del robot**

Las pruebas unitarias del robot están directamente relacionadas con la comprobación de la correcta programación de los movimientos del robot. Conocida la distribución de la mesa, se comprueba que el robot llega adecuadamente y sin ningún impedimento a cada una de las posiciones requeridas. Una vez las posiciones están definidas y guardadas, se realiza un programa en el cual se define la secuencia de movimientos que debe realizar el robot. Para el desarrollo de este programa se toma de referencia el servicio 4, el cual realiza a parte de realizar el montaje con todas las piezas disponibles para el ensamblado de los ítems, también trabaja con el máximo número de ítems posibles (6 ítems).

### **5.1.2 Pruebas de conexión del PLC y el bróker**

Las simulaciones requeridas para estas pruebas precisan la utilización de la herramienta PLCSim Advance. Esta herramienta permite simular un PLC real realizando una instancia de PLC virtual, siendo necesario asignarle una dirección IP en la misma subred que se encontraba el bróker.

Una vez se comprueba que la conexión entre PLCSim Advance y el bróker es correcta y permite transferir datos básicos de un dispositivo a otro, se plantean pruebas en las que es necesario arrancar el bróker (véase en la Figura 5.2) y realizar la conexión

mediante el cliente MQTT definido en el PLC. Para ello se hace uso de la variable 'enable', la cual permite que se realice la conexión. Mientras esta variable se mantenga activa, se mantendrá la conexión. Si se desea suscribirse a un tópic, es suficiente con activar la variable 'subscribe'. En el caso de querer publicar en el bróker, es necesario que la opción de 'subscribe' este desactivada y generar un flanco positivo en la variable 'publish'. Todas estas variables hacen referencia a las variables definidas en el cliente MQTT de la Figura 4.9.

```
Microsoft Windows [Versión 10.0.19044.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\alazn>cd "c:\Program Files\mosquitto"

c:\Program Files\mosquitto>mosquitto -v -c mosquitto.conf
1653903507: mosquitto version 2.0.14 starting
1653903507: Config loaded from mosquitto.conf.
1653903507: Opening ipv4 listen socket on port 1883.
1653903507: mosquitto version 2.0.14 running
```

Figura 5. 2. Arranque de Mosquitto

### 5.1.3 Pruebas de conexión del PLC y el robot

Dado que a conexión entre el PLC y el robot se realiza con el protocolo Profinet, es necesario habilitar la opción de Profinet en el UR3e, tal y como se muestra en la Figura 5.3. Una vez esté la opción habilitada, el led se pondrá en amarillo indicando que está preparado para comunicar. Posteriormente, cuando se realice la conexión con el PLC, el led se cambia a verde confirmando que se ha realizado la conexión adecuadamente y el robot está preparado para realizar la comunicación.



Figura 5. 3. Configuración Profinet en la instalación del robot

Las primeras pruebas realizadas para comprobar la conexión entre el PLC y el robot, permitieron identificar la incapacidad de conexión de PLCSim Advance mediante Profinet. Por ello, en estas pruebas es necesario hacer uso del PLC físico.

Trabajar con el PLC físico precisa crear su instancia dentro del TIA Portal y realizar la conexión con el robot. En la Figura 5.4 se muestran los tres componentes (incluido el HMI para el control del proceso), la comunicación realizada entre ellos y la dirección IP de cada uno de los dispositivos. Una vez instanciado el PLC físico, la conexión entre el PLC y el robot se ha realizado adecuadamente, consiguiendo el intercambio de datos.

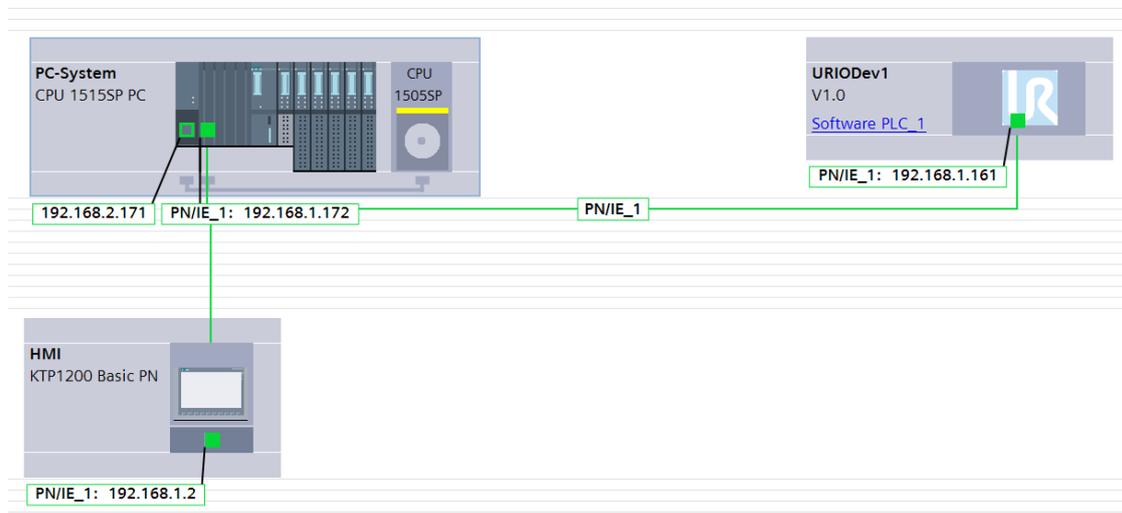


Figura 5. 4. Dispositivos y redes empleados en las pruebas de conexión del PLC y el robot

#### 5.1.4 Pruebas de conexión entre el agente máquina y el agente pasarela

Al no tener el agente máquina integrado en la arquitectura FlexManSys, las pruebas correspondientes a la conexión entre el agente máquina y el agente pasarela se realizan con ayuda de una consola que permite escribir manualmente la información que se necesita para el comienzo del proceso de ensamblaje.

Para poder comprobar que la conexión entre ambos agentes funciona correctamente, se realiza el envío de datos con una sola variable. Este envío permite asegurarse de que la variable enviada desde el agente máquina llega correctamente hasta el agente pasarela. Tras ello, el agente pasarela envía un mensaje de confirmación.

### **5.1.5 Pruebas de conexión entre el bróker y el agente pasarela**

Con objeto de realizar la conexión entre el bróker y el agente pasarela se debe tener en cuenta que se deben arrancar en este orden: primero el bróker y después el agente pasarela. Una vez estén arrancados, será posible arrancar el agente máquina, que posee las variables definidas con unos valores fijos. Durante estas pruebas se realiza el envío de dichas variables hasta el bróker, y mediante la consola que ofrece MQTT Explorer se realiza el envío desde el bróker hasta el agente pasarela.

En sentido contrario, para poder recibir los datos procedentes del bróker, el cliente MQTT definido en el agente pasarela debe estar suscrito al tópico correspondiente.

## **5.2 Pruebas de integración**

Las pruebas de integración se realizan según una estrategia bottom-up. En primer lugar, se realiza la integración de los dispositivos que conforman el asset. Una vez obtenida una comunicación correcta en ambos sentidos entre el PLC y el robot, el siguiente paso en la integración consiste en añadir el bróker. Para finalizar con las pruebas de integración se agregan los agentes pasarela y máquina.

Cabe destacar que, si bien las pruebas unitarias se ciñen a la conexión entre dispositivos, las pruebas de integración se focalizan en el correcto intercambio de las estructuras de datos asociadas con las solicitudes de servicios e información sobre el desempeño de los mismo.

### **5.2.1 Pruebas de integración del asset**

Para poder llevar a cabo todas las transferencias de datos entre el PLC y el robot se debe utilizar la estructura de datos que ofrece el robot a la hora de instanciarlo en TIA Portal. Para ello es necesaria la previa instalación del GSD de Universal Robots. Concretamente, un GSD es un archivo de texto ASCII que contiene las especificaciones del dispositivo para la comunicación. Sin la instalación de este archivo no sería posible la configuración de red del robot. Una vez se han instanciado, se debe configurar la misma subred a los dos dispositivos a comunicar, para que se puedan ver entre ellos. Teniendo en cuenta la estructura de la que dispone el PLC, se han definido las siguientes dos direcciones IP para los dispositivos.

- PLC → 192.168.1.172

- UR3e → 192.168.1.161

Gracias a GSD de Universal Robots se consigue acceder a los diferentes registros que se encuentran en la controladora del robot desde el PLC, haciendo más fácil el intercambio de datos entre los dos dispositivos. Dependiendo del registro que se necesite en la comunicación se hace uso de un tipo de estructura de datos u otra. Es decir, si se desea leer datos del registro en el que se guardan los datos que salen del robot, se hace uso del tipo 'UR\_T2O'. En el caso de querer escribir sobre las variables del robot se define el tipo de datos 'UR\_O2T'.

- UR\_T2O → tipo de datos de salida.
- UR\_O2T → tipo de datos de entrada.

Para que la controladora del robot sea consciente de los cambios o los valores que se le están mandando se deben usar las siguientes instrucciones. La instrucción comenzado por 'read' representa la lectura desde el robot al programa de TIA Portal, lo que significa que esta lectura se le deberá asignar a una variable que se defina en el programa del robot. En cambio, en el caso de tener 'write', se escribirá sobre un registro del robot el valor deseado, consiguiendo que el valor enviado se muestre en el programa de TIA Portal. Las siguientes líneas de comandos son las que se han utilizado en el programa del robot.

- read\_input\_boolean\_register(address)
- read\_input\_integer\_register(address)
- write\_output\_boolean\_register(address, value)
- write\_output\_integer\_register(address, value)

Las pruebas de integración iniciales consisten en el intercambio de datos booleanos entre los dos dispositivos, con el objetivo de poner en marcha el programa del robot cuando el PLC active el bit que corresponde con la variable MARCHA. Esta variable se ha generado para poder arrancar el proceso de ensamblaje cuando dicha variable se ponga a *true* (verdadero). En el caso de enviar bits desde el robot, se definen dos bits diferentes, el primero de ellos se activa cada vez que finaliza el ensamblaje de un ítem. El proceso de ensamblaje se realiza ítem a ítem, es decir, una vez comienza a realizar un servicio en un ítem debe finalizar con el ítem antes de seguir con el siguiente. Por ello, cada vez que se termine con el ensamblaje de un ítem se debe activar el led que lo indique.

El segundo se activa cuando se haya finalizado con el proceso completo, esto significa que se han realizado todos los ítems solicitados y que el palé está en la posición final de la mesa de trabajo.

Tras conseguir un resultado satisfactorio de estas pruebas se continúa con el envío de valores enteros. Así, se permite definir el servicio que se quiere realizar y la cantidad de ítems. Para ello se generan dos variables en el PLC donde se guarda el valor de estos dos datos. Como todavía no se ha integrado la parte del agente, se utiliza una variable fija que se define al comienzo de la simulación.

Al terminar con esta parte de la integración se consigue que el robot comience a funcionar cuando la variable booleana del PLC se active. De esta manera, aunque el programa del robot esté en ejecución el robot no se moverá hasta que la variable lo indique. Una vez la variable se activa, los datos que hacen referencia al servicio y la cantidad de ítems llegan hasta la controladora del robot, permitiendo que el robot realice el servicio solicitado.

### **5.2.2 Pruebas de integración del bróker**

El siguiente paso está relacionado con la integración del bróker, de modo que los datos procedentes del bróker deben llegar hasta la controladora del robot. Para realizar las pruebas de integración se emplea MQTT Explorer que es un visualizador de la información que llega al bróker. Al no tener el agente máquina integrado, los datos que deben llegar del agente se escriben en la consola que ofrece MQTT Explorer. En ella, se deben definir todos los datos del proceso para que después lleguen hasta la controladora del robot de manera ordenada.

La comunicación en sentido contrario requiere suscribirse al tópico indicado para conseguir que los datos lleguen desde el PLC hasta el cliente MQTT. Para ello se han generado las siguientes instrucciones de código (véase en la Figura 5.5). En ella se muestra el cambio que se realiza a la hora de definir el tópico en el cliente MQTT dependiendo de si se quiere suscribir o publicar en uno.

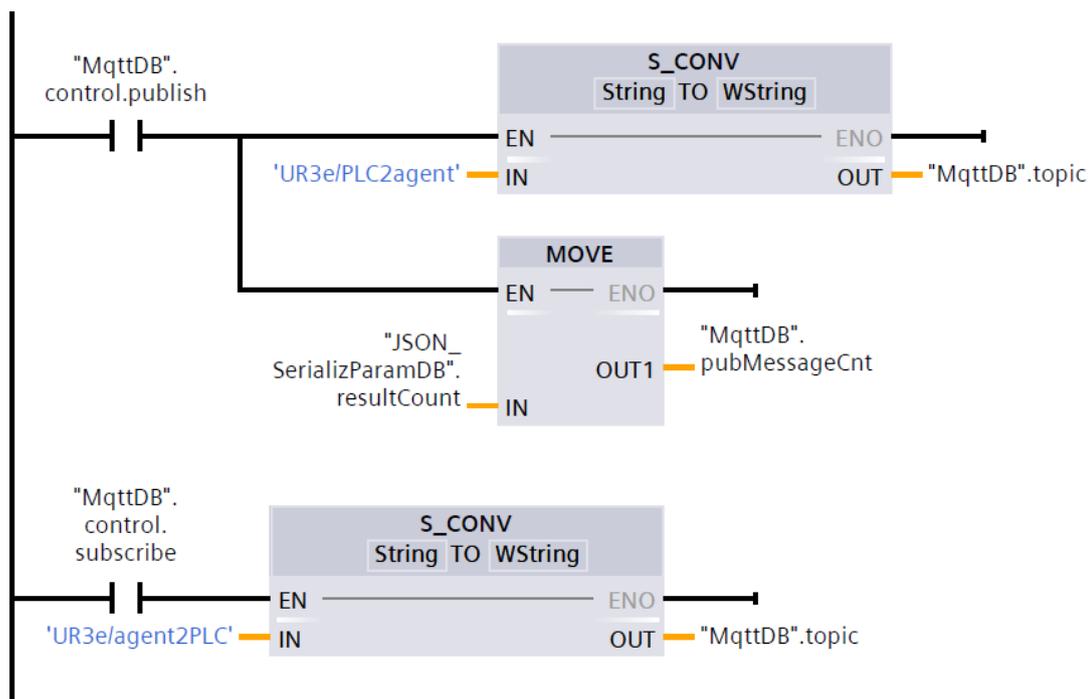


Figura 5. 5. Selección de tópicos (TIA Portal)

Considerando que se ha realizado la suscripción adecuadamente, los datos que se reciben llegan definidos en un array de bytes (matriz de bytes), que se debe transformar en una estructura de datos JSON. Para esta conversión se hace uso de la librería para el Flujo de Datos ('Library for Data Streams', LStream) [49], la cual ofrece una serie de bloques que facilitan la conversión de una matriz de bytes a una estructura de datos y viceversa.

En el caso de querer convertir los datos procedentes del bróker, se utiliza el bloque 'LStream\_jsonDeserializer' (véase en la Figura 5.6) que proporciona dicha librería, permitiendo la conversión de la matriz de bytes recibida que corresponde al formato de datos JSON en un formato que pueda ser procesado por un controlador SIMATIC S7. Este bloque hace uso de la matriz de bytes ('raw') procedente del cliente MQTT, una variable booleana ('execute') con la cual se indica el momento de la deserialización en el instante que hay un flanco positivo en esta variable, y la variable estructurada ('tree'), en la cual se recopila la estructura de datos deserializados de la información de la matriz recibida del bróker.

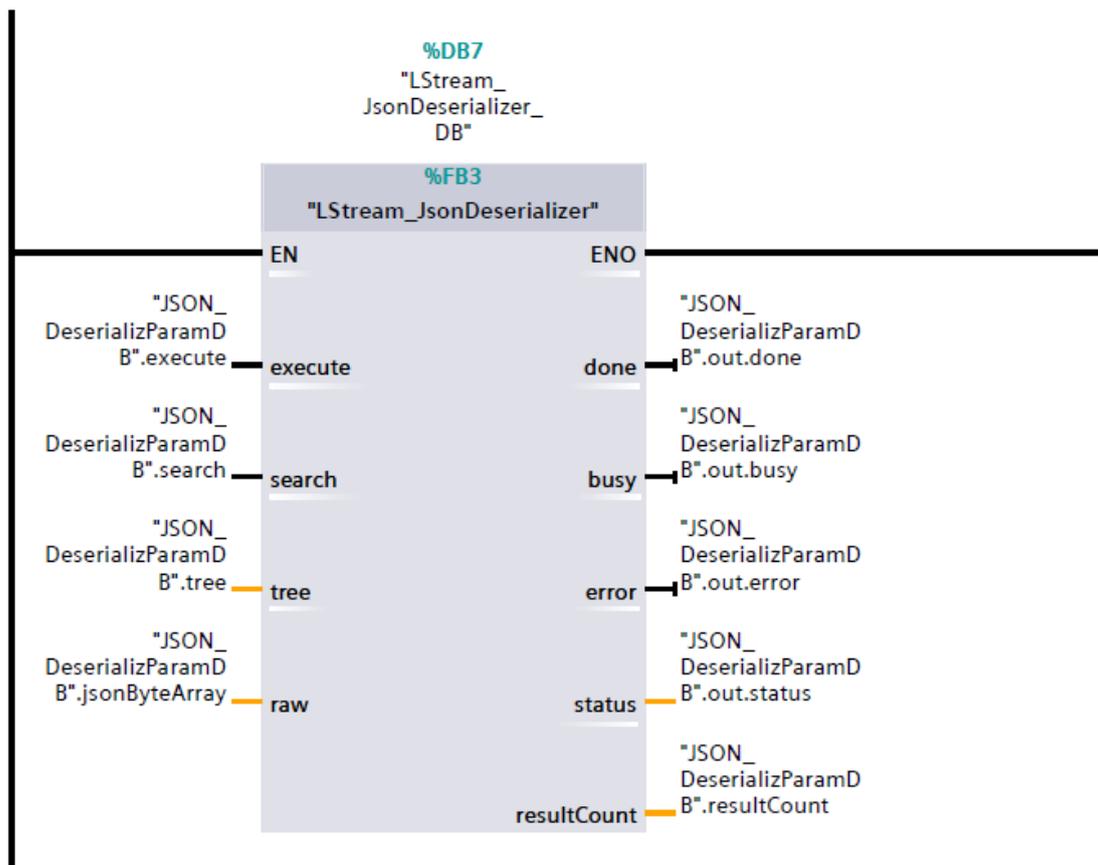


Figura 5. 6. Bloque de deserializar (TIA Portal)

Para la estructura de datos procedente el robot, también es necesario serializar los datos. Lo que significa que de una estructura de datos se generará una matriz de bytes. Esta matriz será la que se envíe al bróker mediante el cliente MQTT. Para la serialización se hace uso del bloque 'LStream\_JSONSerializer' mostrado en la Figura 5.7. A este bloque se le debe facilitar la estructura ('tree') de datos a serializar y la variable donde guardar la matriz de bytes ('jsonByteArray'). Una vez generado un flanco positivo en la variable 'execute', se obtiene la matriz de bytes y el número de bytes que hay en el array ('count'). Ambos datos se facilitan al cliente MQTT, el cual se encarga de enviar la información al bróker.

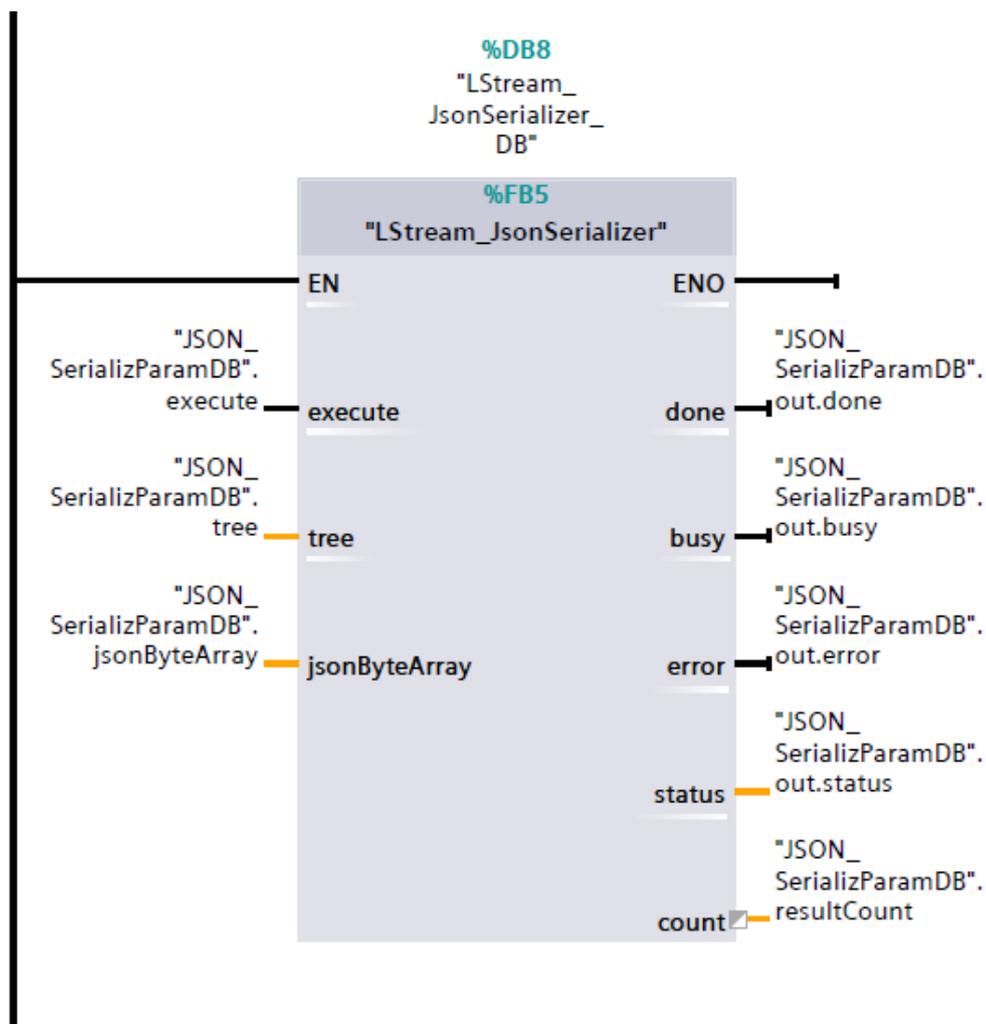
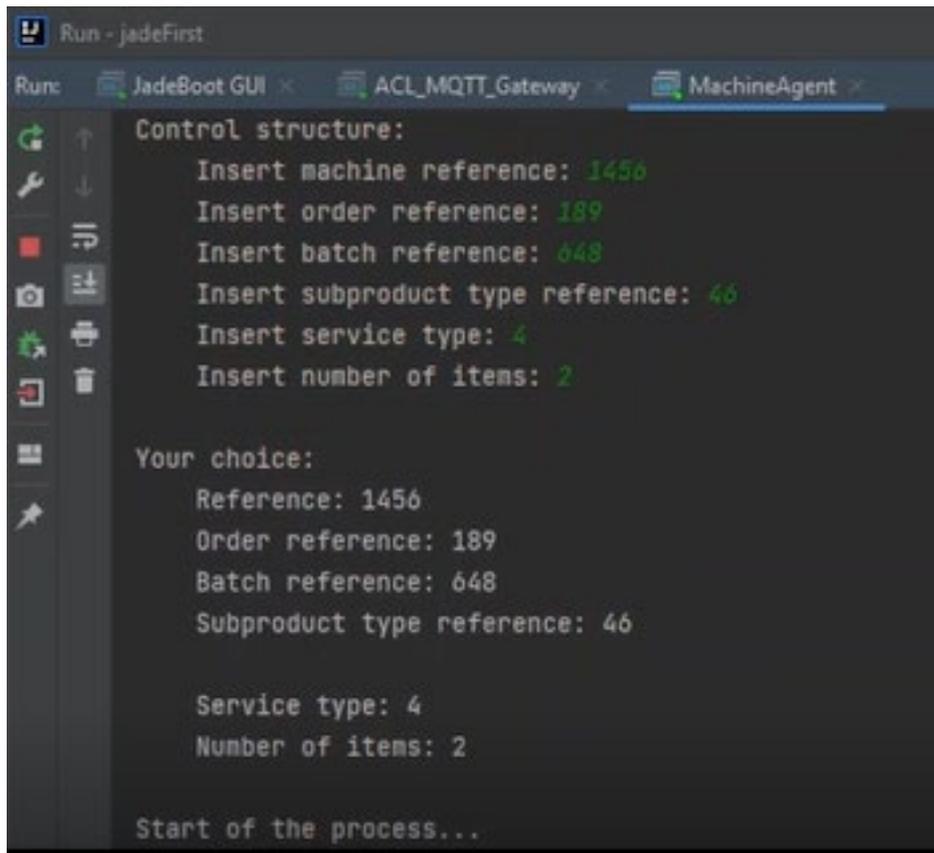


Figura 5. 7. Bloque de serializar (TIA Portal)

### 5.2.3 Pruebas de integración del agente máquina y el agente pasarela

Para finalizar con las pruebas de integración se debe añadir el agente máquina con el agente pasarela a las pruebas realizadas hasta el momento. Como se ha comentado anteriormente, al tener una versión simplificada del agente máquina, el proceso de ensamblaje comienza por la solicitud de datos realizada por consola. El usuario debe introducir manualmente los siguientes datos de fabricación (véase en la Figura 5.8): referencia de la máquina, referencia del pedido, referencia del lote, referencia del tipo del subproducto, tipo de servicio a realizar y cantidad de ítems.



```
Run - jadeFirst
JadeBoot GUI x ACL_MQTT_Gateway x MachineAgent x
Control structure:
  Insert machine reference: 1456
  Insert order reference: 189
  Insert batch reference: 648
  Insert subproduct type reference: 46
  Insert service type: 4
  Insert number of items: 2

Your choice:
  Reference: 1456
  Order reference: 189
  Batch reference: 648
  Subproduct type reference: 46

  Service type: 4
  Number of items: 2

Start of the process...
```

Figura 5. 8. Definición de los datos de fabricación

El envío de la estructura de datos solamente se realiza en el momento que se arranca el agente máquina. Una vez se haya realizado dicho envío, el agente se queda bloqueado a la espera de recibir un mensaje desde el agente pasarela. Dicho mensaje contiene la estructura de datos procedente del proceso de ensamblaje realizado por el robot. Al recibir dicho mensaje, se muestran los datos en la consola del agente, tras ello el agente se vuelve a bloquear a la espera de un nuevo mensaje.

### 5.3 Pruebas de verificación y validación

Finalizada la integración del agente máquina, se procede a realizar las pruebas de verificación y validación del Componente I4.0 de la estación de ensamblado. Para visualizar la secuencia del proceso y conseguir un mayor control, se ha diseñado un HMI en el cual se muestran los datos más relevantes. El HMI se ha dividido en 4 secciones (véase en la Figura 5.9).

### 1. **Suscripción** (arriba a la izquierda)

Este apartado está compuesto por dos interruptores que permiten la conexión y la suscripción del Cliente MQTT al bróker con sus correspondientes leds. En el momento que se desea conectar con el bróker se debe poner en ON el interruptor 'Conexión con el bróker', lo que conlleva a que se ponga el led en verde si la conexión se ha hecho correctamente. Tras ello se debe cambiar la posición del interruptor de 'Suscripción al bróker', en este caso también se pondrá el led en verde si la suscripción se hace adecuadamente.

Una vez hecha la suscripción al tópic correspondiente se espera hasta que lleguen los datos procedentes del agente. Cuando los datos hayan llegado, el led que indica 'Datos recibidos del bróker' comienza a parpadear y continuará parpadeando hasta que se presione el pulsado de 'Deserializar datos'. Tras pulsarlo los datos llegan hasta la controladora del robot y se comienza el proceso de ensamblaje.

### 2. **Publicación** (abajo a la izquierda)

Este apartado está compuesto por dos leds que indican (cambiando de rojo a verde) en qué momento se serializan los datos que se deben enviar al bróker y el momento en el que se realiza la publicación en él.

### 3. **Solicitud** (arriba a la derecha)

Este apartado muestra el servicio y el número de ítems solicitados por el usuario tras haber deserializado los datos mediante el pulsador que hay en el apartado suscripción.

### 4. **Producción** (abajo a la derecha)

Este último apartado muestra todos los datos necesarios de la producción. Estos datos irán actualizándose a medida que el proceso vaya ejecutándose. En él se muestra el led que indica que ha llegado un nuevo servicio a realizar ('New service'), la cantidad de ítems completados hasta el momento ('Item Completed'), valor que va cambiando con la evolución del ensamblaje, un led que se activa cada vez que se finaliza el ensamblaje de un ítem y un led que indica el momento en el que el servicio ha finalizado por

completo. Por otro lado, se muestran los tiempos de producción. Concretamente, se muestra la hora y día del comienzo del servicio, así mismo la hora y día de comienzo y final del ensamblaje de cada uno de los ítems solicitados.



Figura 5. 9. HMI



*CAPÍTULO 6*

**METODOLOGÍA DE DESARROLLO  
DEL TRABAJO**

---



## 6 Metodología de desarrollo del trabajo

En este capítulo se define la metodología utilizada para el desarrollo del proyecto, teniendo en cuenta el cómputo de horas a dedicar. Al comienzo de este proyecto se hizo una estimación aproximada de un cómputo de 300 horas, comenzando en septiembre y finalizando en junio.

Para el progreso del proyecto ha sido necesario seguir un orden en el desarrollo de cada una de las tareas. A pesar de necesitar una base de conocimiento de las tareas previas para poder continuar con las siguientes, no ha sido necesario tenerlas completamente finalizadas para avanzar en el proyecto. Esta es la razón por la cual las tareas aparecerán fraccionadas en el diagrama de Gantt mostrado en uno de los siguientes subapartados.

Las diferentes tareas a realizar durante el proyecto son las siguientes:

1. Formación
2. Programación del robot
3. Conexión PLC - Robot
4. Comunicación MQTT
5. Agentes
6. Integración
7. Documentación

Este capítulo se divide en tres apartados. En el primero se realiza una descripción de cada una de las tareas. En el segundo se identifican las fases del proyecto que agrupan dichas tareas y se recoge en una tabla el tiempo dedicado a las tareas de cada fase. Por último, en el tercer apartado, se presenta la planificación del proyecto con el cronograma de las tareas en un Diagrama de Gantt.

### 6.1 Descripción de las tareas

En este apartado de la metodología se reúnen las explicaciones de cada una de las tareas del proyecto.

### **6.1.1 Formación**

Al comienzo de un proyecto es necesario recopilar información sobre el tema que se va a tratar, recopilando toda la información en los artículos seleccionados. Dada la variedad de tecnologías utilizadas durante el proyecto, ha sido necesario realizar una formación modulada dependiendo de la tarea que se estaba realizando. En concreto, tecnologías relacionadas con el robot, TIA Portal y los agentes. El tiempo dedicado a la formación en cada una de las tecnologías ha dependido del conocimiento previo.

### **6.1.2 Programación del robot**

El programa del robot precisa flexibilizar al máximo el proceso a realizar. El código referente a la programación del robot ha sido desarrollado desde cero en el Teach Pendant, requiriendo aprender su manejo.

### **6.1.3 Conexión PLC – Robot**

Esta tarea hace referencia a la necesidad de comunicar el PLC y el robot, consiguiendo realizar la transferencia de datos en ambos sentidos. Para ello, se ha utilizado Profinet, siendo necesario el uso de librerías específicas que ofrece SIMATIC S7 para poder realizar la conexión.

### **6.1.4 Comunicación MQTT**

La comunicación MQTT permite que los datos enviados desde el agente hasta el bróker lleguen al PLC, y viceversa. Esta tarea aborda la gestión del bróker y el tratamiento realizado a los datos que se envían, identificando los siguientes tres aspectos:

- Configuración del bróker.
- Conexión entre el PLC y el bróker.
- Conexión entre el bróker y el agente.

### **6.1.5 Agentes**

Esta fase se dedica a la programación de los agentes (máquina y pasarela) y las estructuras de datos enviadas en ambos sentidos que han sido meticulosamente comprobadas mediante las simulaciones necesarias.

### **6.1.6 Integración**

Una vez finalizadas las tareas, es necesario realizar la integración de cada una de las tecnologías, consiguiendo verificar y validar su correcto funcionamiento. Para ello se deben hacer pruebas tanto de comprobación de las conexiones entre los diferentes dispositivos, como del funcionamiento del robot.

Esta tarea engloba pruebas de integración, de validación y de verificación. Por lo tanto, esta fase se dedica a la comprobación de que todas las tareas funcionan correctamente y realizan lo solicitado.

### **6.1.7 Documentación**

A medida que se va realizando el proyecto, se recopila información, comenzando por buscar información para la parte más teórica del documento y finalizando por documentar la parte del desarrollo del proyecto.

## **6.2 Fases del proyecto**

Para el conocimiento real del periodo de tiempo dedicado al proyecto se ha generado la Tabla 6.1 que agrupa las tareas en 3 fases: Formación, Desarrollo y Documentación. En la tabla se muestran las fechas aproximadas de comienzo y fin de cada una de las tareas. Aunque se indique el primer día y el último que se ha dedicado a esa tarea, eso no significa que todos los días que fija ese periodo de tiempo se hayan dedicado a esa tarea. Por ello, se ha generado una columna que indica la duración en días de cada una de las tareas, permitiendo fijar los días exactos dedicados a cada una de las tareas, siendo posible dedicar un mismo día a dos tareas diferentes. Tras calcular los días dedicados a cada una de las tareas se ha conseguido obtener los días exactos dedicados al proyecto, exactamente 198 días.

Tabla 6. 1. Desglose de la ejecución de las tareas por fases

<b>Formación</b>			
Tarea	Fecha de inicio	Fecha final	Días dedicados a la tarea
Entorno	15-10-2021	19-10-2021	3
Robot	16-10-2021	21-10-2021	4
TIA Portal	26-10-2021	27-10-2021	2
Agentes	18-10-2021	11-01-2022	11
<b>Desarrollo</b>			
Tarea	Fecha de inicio	Fecha final	Días dedicados a la tarea
Programación del robot	04-11-2021	06-04-2022	25
Conexión PLC - Robot	20-01-2022	22-03-2022	23
Comunicación MQTT	23-11-2021	19-05-2022	42
Agente	2022-05-19	27-05-2022	12
Integración	26-04-2022	27-05-2022	34
<b>Documentación</b>			
Tarea	Fecha de inicio	Fecha final	Días dedicados a la tarea
Documentación	27-10-2021	24-06-2022	90
<b>Total</b>			<b>198</b>

### 6.3 Diagrama de Gantt

El la Figura 6.1 se muestra el Diagrama de Gantt con la planificación del proyecto. En dicho diagrama se pueden observar las fases descritas, así como las fechas de comienzo y fin de cada una de las tareas.

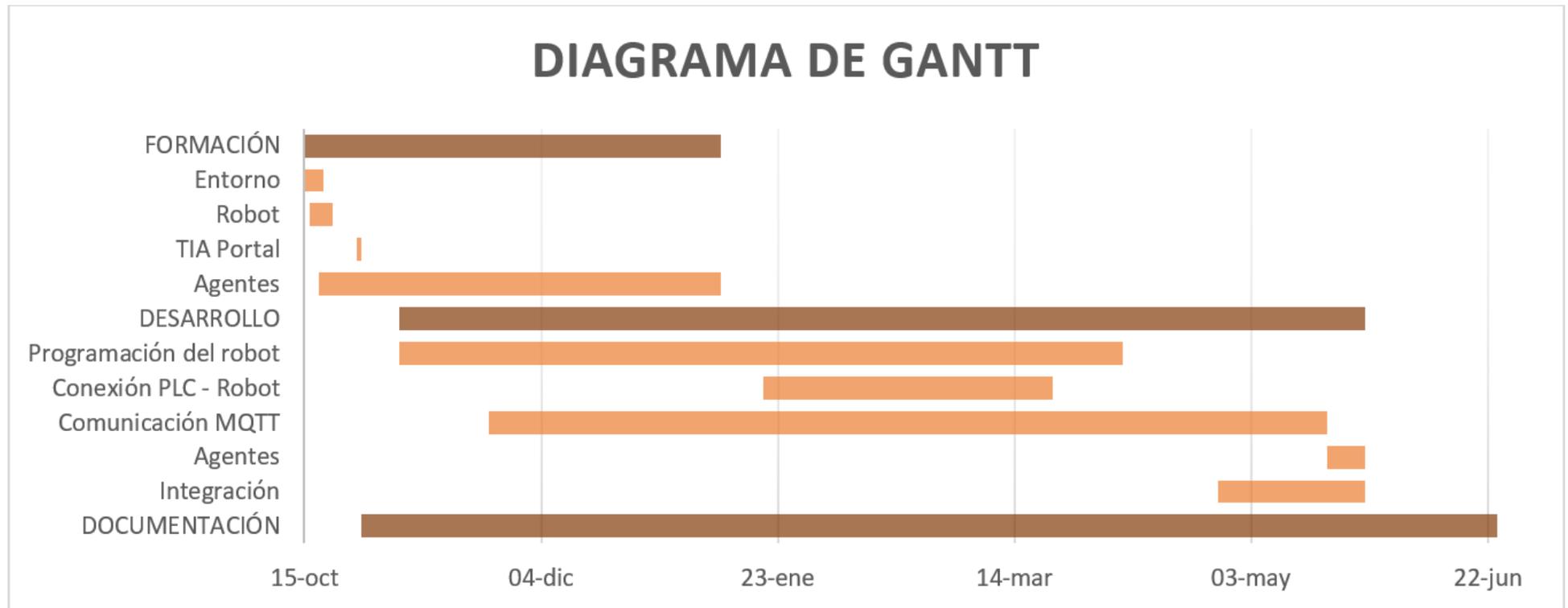


Figura 6. 1. Diagrama de Gantt



*CAPÍTULO 7*

# **ASPECTOS ECONÓMICOS**

---



## 7 Aspectos económicos

En este capítulo del documento se presentan los aspectos económicos del proyecto. En los siguientes puntos se presenta una descripción de cada una de las partidas identificadas, para después mostrar todos los gastos resumidos en la Tabla 7.1.

- **Horas internas.** Este apartado incluye los gastos asociados al personal involucrado en el proyecto. En él han tomado parte una alumna, calculado como ingeniero junior con un coste horario de 20€/h; y dos profesoras, consideradas ingeniero senior, con una retribución de 40€/h. Esta partida supone la gran mayoría de los gastos.
- **Amortizaciones.** En este apartado se considera el coste de activos reutilizables, de forma relativa a su tiempo de uso y vida útil. En este proyecto, los activos utilizados estaban en el puesto de trabajo. Los gastos que generan las licencias requeridas por los programas no gratuitos, como el ordenador utilizado y sus licencias, el PLC empleado para las pruebas y las licencias que conllevan el uso de ello, el robot, la mesa, y otros dispositivos se introducen en esta partida.
- **Gastos de desarrollo.** Esta partida recopila los gastos que ha generado el desarrollo del proyecto. En concreto, los gastos de los materiales generados por la impresora 3D.
- **Costes indirectos.** En esta partida se tienen en cuenta los costes no imputables de un proyecto, como gastos eléctricos, limpieza o mantenimiento de equipos. Se calcula como un 10% de la suma de los demás gastos.

Tabla 7. 1. Aspectos económicos

<b>HORAS INTERNAS</b>				
Concepto	Coste por hora	Horas totales		Coste
Ingeniero Senior (directoras)	40€/h	35h		1400€
Ingeniero Junior (alumna)	20€/h	300h		6000€
<b>Total</b>				<b>7400€</b>
<b>AMORTIZACIONES</b>				
Concepto	Coste unitario	Vida útil (años)	Tiempo de uso (meses)	Total amortizaciones
UR3e	16000€	15	7	622,22€
Herramienta del robot	1200€	15	7	46,66€
Mesa de trabajo	1500€	15	7	58,33€
TIA Portal V16	4000€	1	7	2333,33€
PLC	1800€	10	7	105€
Ordenador	1500€	3	7	15,55€
Cables de red	30€	2	7	8,75€
<b>Total</b>				<b>3203,47€</b>
<b>GASTOS DE DESARROLLO</b>				
Concepto	Precio	Cantidad		Coste
Elementos 3D	20€/kg	650g		13€
<b>Total</b>				<b>13€</b>
<b>Subtotal:</b>				<b>10616,47€</b>
Costes indirectos e imprevistos		10% del subtotal		1061,65€
<b>Total:</b>				<b>11678,12€</b>

Analizando el gasto total obtenido se observa que el mayor gasto se genera en las horas internas dedicadas al proyecto.

*CAPÍTULO 8*

# **CONCLUSIONES**

---



## 8 Conclusiones

El trabajo realizado en este proyecto Fin de Máster ha contribuido a la integración de un robot colaborativo UR3e en un entorno de fabricación flexible. En él se ha realizado desde la programación de cada uno de los servicios del robot hasta su integración como Componente I4.0 en un sistema multiagente. Como se ha comentado en la introducción, este trabajo está dentro de un proyecto de investigación, por lo que se ha partido del conocimiento del propio grupo de investigación. Por la misma razón, los progresos que se obtengan durante el desarrollo del trabajo servirán de base para futuros proyectos del grupo de investigación.

En lo que se refiere al activo de fabricación (la estación de ensamblado), se ha logrado programar el robot de manera que se ha conseguido mayor eficacia en su funcionamiento. La estructura modular definida para el programa del robot permite aumentar la eficiencia y flexibilidad del proceso de ensamblado, tanto en producción normal como en situaciones excepcionales de producción. Así, el robot no sólo es capaz de realizar las operaciones de ensamblaje desde diferentes montajes iniciales (sólo base, base con rodamiento, etc.), sino que también es capaz de procesar diferente número de ítems. Por otro lado, la conexión establecida entre el robot y el PLC, el segundo componente del activo, ha permitido descargar al robot tareas de gestión relativas a la comunicación con el agente máquina. De esta forma, el robot se ha centrado en la realización de los servicios de fabricación y del envío de información relativa a su evolución, mientras que el PLC ha sido el encargado de atender las solicitudes del agente máquina y del tratamiento de la información recibida del robot para convertirla a la estructura demandada por el agente.

Además, el hecho de disponer de un PLC también ha permitido ampliar las capacidades de comunicación propias del robot, siendo posible hacer uso del paradigma publicista/suscriptor para la comunicación AAS-activo, siendo el bróker mosquitto el intermediario de dicha comunicación. Así, haciendo uso de la herramienta de ingeniería TIA Portal se ha dispuesto de un cliente MQTT para la conexión PLC-bróker, tanto para publicar como para suscribirse a tópicos. Ha sido

necesario desarrollar otro cliente MQTT como parte del agente pasarela para la conexión agente-bróker. De esta forma se ha conseguido la integración del activo y el AAS del componente I4.0

Finalmente, es importante destacar que gracias al HMI y a la consola desarrollados y a las pruebas realizadas se ha podido comprobar el correcto funcionamiento de la arquitectura de integración. Es decir, se ha podido comprobar que diferentes solicitudes de servicio para diferente cantidad de ítems han sido correctamente realizadas por el robot. Y que la información relativa a la evolución de dichos servicios ha sido actualizada en el agente.

Se puede decir, por lo tanto, que mediante el desarrollo de cada una de las tareas realizadas durante el proyecto se ha conseguido cumplir con cada uno de los objetivos propuestos a comienzo del proyecto. Cabe destacar que se han tenido que superar varios problemas que han ido surgiendo durante el proyecto tanto a nivel de comunicaciones como a nivel de código.

En el futuro, se espera el uso de un PC industrial (IPC127e), consiguiendo instanciar tanto el agente como el bróker en él. Además, se pretende la integración total del Componente I4.0 en la arquitectura de FlexManSys, de tal manera que no sea necesario el uso de la consola. Finalmente, para realizar la supervisión del proceso de fabricación sería necesario diseñar un HMI nuevo que permita visualizar la evolución del proceso.

*CAPÍTULO 9*

## **REFERENCIAS BIBLIOGRÁFICAS**

---



## 9 Referencias bibliográficas

- [1] M. Hermann, T. Pentek, B. Otto, and T. \* Pentek, “Design Principles for Industrie 4.0 Scenarios: A Literature Review”, doi: 10.13140/RG.2.2.29269.22248.
- [2] J. Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, “Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1504–1526, Jul. 2017, doi: 10.1109/COMST.2017.2691349.
- [3] “RAMI 4.0, el Modelo de arquitectura de referencia para la industria 4.0 -.” <https://blog.tenea.com/rami-4-0-el-modelo-de-arquitectura-de-referencia-para-la-industria-4-0/> (accessed Jun. 09, 2022).
- [4] *2660.1-2020 - IEEE Recommended Practice for Industrial Agents Integration of Software Agents and Low-Level Automation Functions*. IEEE, 2021.
- [5] A. Lopez, E. Estevez, O. Casquero, and M. Marcos, “Using industrial standards for modeling flexible manufacturing systems,” in *Proceedings - 2020 IEEE Conference on Industrial Cyberphysical Systems, ICPS 2020*, Jun. 2020, pp. 41–46. doi: 10.1109/ICPS48405.2020.9274785.
- [6] A. López, E. Estévez Estévez, and M. Marcos, “Trazabilidad de la producción basada en agentes industriales,” in *XLII JORNADAS DE AUTOMÁTICA: LIBRO DE ACTAS*, Servizo de Publicacións da UDC, 2021, pp. 717–723. doi: 10.17979/spudc.9788497498043.717.
- [7] A. Lopez, O. Casquero, E. Estevez, P. Leitao, and M. Marcos, “Towards the generic integration of agent-based AASs and Physical Assets: a four-layered architecture approach,” Oct. 2021, pp. 1–6. doi: 10.1109/indin45523.2021.9557568.
- [8] “Pesi | Artículo sobre el modelo RAMI de Industria 4.0.” [http://www.pesi-seguridadindustrial.org/frontend/pNoticia/articulo-sobre-el-modelo-rami-de-industria-40\\_537](http://www.pesi-seguridadindustrial.org/frontend/pNoticia/articulo-sobre-el-modelo-rami-de-industria-40_537) (accessed Jan. 30, 2022).
- [9] “Industry 4.0 and the fourth industrial revolution explained.” <https://www.i-scoop.eu/industry-4-0/> (accessed Nov. 29, 2021).
- [10] “¿Qué es la Industria 4.0?” <https://www.factoriadelfuturo.com/que-es-la-industria-4-0/> (accessed Nov. 10, 2021).
- [11] “Revolución Industrial – De Industria 1.0 a Industria 4.0 - Desoutter Industrial Tools.” <https://www.desouttertools.mx/industria-4-0/noticias/1015/revolucion-industrial-de-industria-1-0-a-industria-4-0> (accessed Mar. 13, 2022).
- [12] “Short history of manufacturing: from Industry 1.0 to Industry 4.0 - KFactory.” <https://kfactory.eu/short-history-of-manufacturing-from-industry-1-0-to-industry-4-0/> (accessed Oct. 27, 2021).

- [13] “The differences between Industry 3.0 and 4.0: technology and innovation.” [https://audaces.com/en/industry-3-0-industry-4-0-2/#Differences\\_between\\_Industry\\_30\\_and\\_Industry\\_40](https://audaces.com/en/industry-3-0-industry-4-0-2/#Differences_between_Industry_30_and_Industry_40) (accessed May 25, 2022).
- [14] M. A. Pisching, M. A. O. Pessoa, F. Junqueira, D. J. dos Santos Filho, and P. E. Miyagi, “An architecture based on RAMI 4.0 to discover equipment to process operations required by products,” *Computers and Industrial Engineering*, vol. 125, pp. 574–591, Nov. 2018, doi: 10.1016/j.cie.2017.12.029.
- [15] S.-I. Ao and International Association of Engineers, *Topological Approach for Mapping Technologies in Reference Architectural Model Industrie 4.0 (RAMI 4.0)*.
- [16] “Open Source Control Device for Industry 4.0 Based on RAMI 4.0 \_ Enhanced Reader”.
- [17] P. Marcon *et al.*, “The Asset Administration Shell of Operator in the Platform of Industry 4.0; The Asset Administration Shell of Operator in the Platform of Industry 4.0,” 2018.
- [18] Panepistēmio Kyprou, IEEE Industrial Electronics Society, and Institute of Electrical and Electronics Engineers, *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation : September 12-15, 2017, Limassol, Cyprus*.
- [19] “Plataforma Industria 4.0 - Página de inicio.” <https://www.plattform-i40.de/IP/Navigation/DE/Home/home.html> (accessed May 17, 2022).
- [20] C. S. Franklin, E. G. Dominguez, J. D. Fryman, and M. L. Lewandowski, “Collaborative robotics: New era of human–robot cooperation in the workplace,” *Journal of Safety Research*, vol. 74, pp. 153–160, Sep. 2020, doi: 10.1016/j.jsr.2020.06.013.
- [21] A. Grau, M. Indri, L. lo Bello, and T. Sauter, “Robots in Industry: The Past, Present, and Future of a Growing Collaboration with Humans,” *IEEE Industrial Electronics Magazine*, vol. 15, no. 1, pp. 50–61, Mar. 2021, doi: 10.1109/MIE.2020.3008136.
- [22] F. Vicentini, “Collaborative Robotics: A Survey,” 2020, doi: 10.1115/1.4046238.
- [23] “Robots Colaborativos | Universal Robots.” <https://www.universal-robots.com/es/> (accessed Nov. 10, 2021).
- [24] “▷Todo sobre los Robots Colaborativos | Marcas y Precios.” [https://www.neobotik.com/robots-colaborativos/#Caracteristicas\\_de\\_los\\_robots\\_colaborativos](https://www.neobotik.com/robots-colaborativos/#Caracteristicas_de_los_robots_colaborativos) (accessed Jun. 10, 2022).
- [25] “UR3e - Fiche técnica.”
- [26] “Robots UR3e: Robot Colaborativo Universal Robot.” <https://cfzcobots.com/productos/ur3e/> (accessed Nov. 10, 2021).
- [27] “Universal Robots e-Series User Manual UR3e Original instructions (en) UR3e User Manual,” 2009.
- [28] A. Climent Jiménez, “Comunicación Inteligente Robots | Universal Robots.” <https://www.universal-robots.com/es/blog/comunicacion-inteligente-robots/> (accessed Apr. 19, 2022).

- [29] “Cuáles son los protocolos de comunicación industrial - AUTYCOM.” <https://www.autycom.com/protocolos-de-comunicacion-industrial/> (accessed Apr. 19, 2022).
- [30] M. A. Maynard, “Digital Equipment Corporation The Ethernet A Local Area Network Data Link Layer and Physical Layer Specifications intel,” 1980.
- [31] J. Burke, A. Irei, and W. Chai, “Ethernt.” <https://www.computerweekly.com/es/definicion/Ethernet> (accessed May 04, 2022).
- [32] P. S. Marshall, *Industrial Ethernet*.
- [33] “EtherCAT Technology Group | ¿Por qué utilizar EtherCAT?” [https://www.ethercat.org/es/why\\_use\\_ethercat.htm](https://www.ethercat.org/es/why_use_ethercat.htm) (accessed May 04, 2022).
- [34] “Ethernet Powerlink Descripción general y Estandarización.” [https://hmong.es/wiki/Ethernet\\_Powerlink](https://hmong.es/wiki/Ethernet_Powerlink) (accessed May 04, 2022).
- [35] “What is Sercos?” <https://www.sercos.org/technology/what-is-sercos/> (accessed May 04, 2022).
- [36] “CC-Link IE Field Network | Network technology | CC-Link Partner Association.” [https://www.cc-link.org/en/cclink/cclinkie/cclinkie\\_f.html](https://www.cc-link.org/en/cclink/cclinkie/cclinkie_f.html) (accessed May 04, 2022).
- [37] “¿Qué es PROFINET? - Universidad PROFINET.” <https://profinetuniversity.com/profinet-basico/que-es-profinet/> (accessed May 12, 2022).
- [38] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, “Experimental evaluation of PROFINET performance,” in *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 2004, pp. 331–334. doi: 10.1109/wfcs.2004.1377739.
- [39] C. Felser, M. Felser, and H. Kaghazchi, “Improved architecture for Profinet IRT devices,” 2012. doi: 10.1109/ETFA.2012.6489562.
- [40] “PROFINET: Qué es y cómo funciona | Comunicaciones Industriales.” <https://www.cursosaula21.com/profinet-que-es-y-como-funciona/> (accessed Jun. 10, 2022).
- [41] J. Feld, “Profinet - Scalable factory communication for all applications,” in *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 2004, pp. 33–38. doi: 10.1109/wfcs.2004.1377673.
- [42] “Protocolo MQTT - Ditel Diseños y Tecnología S.A.” <https://www.ditel.es/protocolo-mqtt/> (accessed Nov. 30, 2021).
- [43] “MQTT - The Standard for IoT Messaging.” <https://mqtt.org/> (accessed Nov. 30, 2021).
- [44] “Using MQTT to send and receive data for your next project | Opensource.com.” <https://opensource.com/article/18/6/mqtt> (accessed Apr. 05, 2022).
- [45] “MQTT vs CoAP, the battle to become the best IoT protocol.” <https://www.pickdata.net/es/noticias/mqtt-vs-coap-mejor-protocolo-iot> (accessed Jun. 22, 2022).
- [46] “IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains.” <https://www.jetbrains.com/idea/promo/?source=google&medium=cpc&campaign=973>

- 6637745&term=intelij&gclid=Cj0KCQjwzLCVBhD3ARIsAPKYTcRILxabVYxvw313W6R\_cKKGp0TgbmwYltkmOvxj8PmPhKgE7x5sLJwaAkHCEALw\_wcB (accessed Jun. 17, 2022).
- [47] “FIPA ACL Message Structure Specification.” <http://www.fipa.org/specs/fipa00061/SC00061G.html> (accessed Jun. 17, 2022).
- [48] “Eclipse Mosquitto.” <https://mosquitto.org/> (accessed Jun. 11, 2022).
- [49] “Library for Data Streams (LStream),” 2021. [Online]. Available: <http://www.siemens.com/industrialsecurity>.
- [50] “INDUSTRIA 4.0 - Qué es, ventajas y sus aplicaciones.” <https://pyxis.cat/es/industria-4-0/> (accessed Nov. 13, 2021).
- [51] G. Koschnick, M. Hankel, and B. Rexroth, “RAMI 4.0-Structure The Reference Architectural Model Industrie 4.0 (RAMI 4.0) Contact: Reference Architectural Model Industrie 4.0,” 2015, Accessed: Nov. 20, 2021. [Online]. Available: [www.zvei.org/](http://www.zvei.org/)
- [52] “Modularidad en la industria de procesos con I4.0 (Parte 1) - infoPLC.” <https://www.infoplcn.net/plus-plus/tecnologia/item/104784-modularidad-en-la-industria-de-procesos> (accessed Mar. 13, 2022).