

MÁSTER UNIVERSITARIO EN INGENIERÍA DE CONTROL,  
AUTOMATIZACIÓN Y ROBÓTICA

# TRABAJO FIN DE MÁSTER

***DISEÑO E IMPLEMENTACIÓN DEL  
GEMELO DIGITAL DE CÉLULA  
ROBOTIZADA PARA ENTORNO  
INTELIGENTE Y VALIDACIÓN CON  
SOFTWARE IN THE LOOP (SIL)  
DISTRIBUIDO***

**Estudiante:** Aresti Muñoz, Irati

**Director:** Orive Revillas, Darío

**Departamento:** Ingeniería de Sistemas y Automática

**Curso académico:** 2020 – 2021

Bilbao, 10, septiembre 2021



## LABURPENA

Industria 4.0-ren garai honetan, produktuak pertsonalizatzeagatik bezeroek egiten duten eskariak eragin du fabrikazio-prozesuak malguagoak izatea. Hori dela eta, Industria 4.0 teknologietan izandako aurrerapenen ondorioz, Biki Digitalaren kontzeptua manufaktura-industriaren barruan aintzatespena hartzen joan da. Prozesuen edota makina errealeen portaera simulatzen duten erreplika birtualak dira. Gauzen Internet (IoT) bezalako teknologiekin Biki Digitalak modelatzea ahalbidetzen dute denbora errealeko aktiboak simulatzeko eta prozesu industrialen eraginkortasuna handitzeko.

Zentzu honetan, dokumentu honek inguru adimentsu baten barnean robotizatutako zelula malgu baten Biki Digitalaren modelatze eta honen inplementazio birtual eta errealerako gauzatutako metodologia aurkezten du. Tecnomatix Process Simulate V16.0.1 ingeniari-tresnan egindako Bikia modelatzeko jarraitu diren urratsak deskribatzen dira. Ondoren, kontrol-programaren diseinua eta HMI interfazea gainbegiratze-sistema gisa zehazten dira. Ondoren, programak eta Bikia araztu egiten dira, *Software in the Loop* (SiL) arkitektura banatuarekin modu birtualean martxan jarritz. Modu birtualean behar bezala funtzionatzen duela egiaztatu ondoren, zelularen estazioetako bat errealtatean martxan jartzen da.

**Gako Hitzak:** Industria 4.0, Biki Digitala, Zelula robotizatu malgu, Abiarazte birtual banandua, Tecnomatix Process Simulate, Software in the Loop (SiL)

## **RESUMEN**

En esta era de la Industria 4.0 la demanda de los clientes por la personalización de los productos, ha llevado a que los procesos de fabricación sean más flexibles. Por ello, los avances en las tecnologías de la Industria 4.0 han hecho que el concepto de Gemelo Digital haya ido cogiendo reconocimiento dentro de la industria manufacturera. Se puede definir como réplicas virtuales de procesos y/o máquinas reales que simulan su comportamiento. Con tecnologías como el Internet de las Cosas (IoT) hacen posible modelar Gemelos Digitales para simular activos en tiempo real y aumentar la eficiencia de los procesos industriales.

En este sentido, este documento presenta la metodología llevada a cabo para el modelado e implementación virtual y real del Gemelo Digital de una célula robotizada flexible dentro de un entorno inteligente. Se describe los pasos seguidos para el modelado del Gemelo realizado en la herramienta de ingeniería Tecnomatix Process Simulate V16.0.1. Posteriormente, se detalla el diseño del programa de control y la interfaz HMI como sistema de supervisión. Después, se realiza un depurado de los programas y del Gemelo mediante la puesta en marcha virtual con la arquitectura de *Software in the Loop* (SiL) distribuida. Una vez comprobado el funcionamiento correcto de manera virtual, se realiza la puesta en marcha real de una de las estaciones de la célula.

**Descriptores:** Industria 4.0, Gemelo Digital, Célula robotizada flexible, Puesta en marcha virtual distribuida, Tecnomatix Process Simulate, Software in the Loop (SiL)

## **ABSTRACT**

In this era of Industry 4.0, customer demand for product customization has led to more flexible manufacturing processes. For this reason, advances in Industry 4.0 technologies have led to the concept of the Digital Twin gaining recognition within the manufacturing industry. It can be defined as virtual replicas of real processes and/or machines that simulate their behaviour. With technologies such as the Internet of Things (IoT), they make it possible to model Digital Twins to simulate assets in real time and increase the efficiency of industrial processes.

In this sense, this document presents the methodology carried out for the virtual and real modelling and implementation of the Digital Twin of a flexible robotic cell within an intelligent environment. The steps followed for the Twin modelling performed in the Tecnomatix Process Simulate V16.0.1 engineering tool are described. Subsequently, the design of the control program and the HMI interface as a supervision system are detailed. Afterwards, a debugging of the programs and the Twin is carried out by means of the virtual commissioning with the distributed Software in the Loop (SiL) architecture. Once the correct operation has been verified in a virtual way, the real commissioning of one of the stations of the cell is carried out.

**Key words:** Industry 4.0, Digital Twin, Flexible robotic cell, Distributed virtual commissioning, Tecnomatix Process Simulate, Software in the Loop (SiL)



# ÍNDICE GENERAL

<b>1 MEMORIA .....</b>	<b>1</b>
1.1 Introducción .....	1
1.2 Contexto .....	3
1.3 Objetivos y alcance del proyecto.....	7
1.3.1 Montaje de los conjuntos.....	7
1.3.2 Gemelo Digital de la célula flexible.....	8
1.3.3 Programa de control y puesta en marcha virtual distribuida.....	8
1.3.4 Implantación real .....	8
1.4 Beneficios que aporta el trabajo .....	10
1.4.1 Beneficios económicos.....	10
1.4.2 Beneficios sociales .....	10
1.4.3 Beneficios técnicos.....	10
1.5 Análisis de alternativas .....	13
1.5.1 Software de desarrollo de Gemelos Digitales .....	13
1.5.2 Puesta en marcha virtual.....	17
1.6 Análisis de Riesgos.....	19
1.6.1 Error en el diseño y modelado del Gemelo Digital .....	19
1.6.2 Error en el programa de control.....	20
1.6.3 Error en la compatibilidad de las herramientas de ingeniería .....	20
1.6.4 Error en la puesta en marcha real.....	20
1.7 Descripción de la solución propuesta. Diseño .....	22
1.7.1 Descripción funcional de la célula robotizada flexible .....	22
1.7.2 Estaciones robotizadas .....	22
1.7.3 AMR.....	29
<b>2 METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO .....</b>	<b>31</b>
2.1 Descripción de fases, tareas, equipos o procedimientos .....	31
2.1.1 Servicios .....	31
2.1.2 Gemelo Digital .....	33
2.1.3 Programa de control para la puesta en marcha virtual.....	65
2.1.4 Puesta en marcha virtual distribuida.....	89
2.1.5 Configuración Hardware del robot.....	96
2.1.6 Programa de control para la puesta en marcha real.....	103
2.1.7 Puesta en marcha real de una estación.....	103
2.2 Diagrama de Gantt/Cronograma .....	105
<b>3 ASPECTOS ECONÓMICOS .....</b>	<b>110</b>

---

<b>3.1 Descripción del presupuesto y/o del presupuesto ejecutado .....</b>	<b>110</b>
3.1.1 <i>Presupuesto de los elementos de la estación real.....</i>	<i>110</i>
3.1.2 <i>Presupuesto de las herramientas de ingeniería .....</i>	<i>112</i>
3.1.3 <i>Grupo del proyecto .....</i>	<i>113</i>
3.1.4 <i>Presupuesto total .....</i>	<i>113</i>
 <b>4 CONCLUSIONES .....</b>	 <b>115</b>
 <b>5 BIBLIOGRAFÍA .....</b>	 <b>117</b>
 <b>6 ANEXO I.....</b>	 <b>119</b>
<hr/>	
6.1 Código de los programas del Robot.....	119
 <b>7 ANEXO II .....</b>	 <b>321</b>
<hr/>	
7.1 Código del programa de control.....	321

# ÍNDICE DE IMÁGENES

Figura 1: Gemelo Digital célula robotizada flexible .....	2
Figura 2: Tecnologías de la Industria 4.0.....	4
Figura 3: Cloud, Fog y Edge Computing en la pirámide de automatización.....	5
Figura 4: Elementos de los conjuntos a montar .....	7
Figura 5: Entorno de Tecnomatix Process Simulate.....	13
Figura 6: Entorno de KUKA.Sim 4.0.....	14
Figura 7: Entorno RoboDK .....	14
Figura 8: Zonas de la mesa de trabajo .....	23
Figura 9: Alimentador rodamientos.....	24
Figura 10: Alimentador de bulones.....	24
Figura 11: Alimentador de tapas interiores .....	24
Figura 12: Alimentadores piezas pequeñas .....	25
Figura 13: Gráfica del campo de trabajo.....	26
Figura 14: Modelo de la herramienta del robot .....	27
Figura 15: Pinza MHK2-16D.....	27
Figura 16: Pinza MHKL2-25D .....	28
Figura 17: Válvula de solenoide de puerto 5/2 SMC serie SY3000 .....	28
Figura 18: Nodo EtherCAT .....	29
Figura 19: Modelo del AMR.....	30
Figura 20: Posición de los ítems en el pallet .....	32
Figura 21: Configuración de la cinemática del robot .....	36
Figura 22: Configuración de la cinemática de la herramienta del robot .....	36
Figura 23: Dependencia del enlace j2 respecto del j1.....	37
Figura 24: Puntos TCP de la herramienta del robot (naranja) .....	38
Figura 25: Bloque lógico de la herramienta del robot .....	38
Figura 26: Tipos de operaciones .....	39
Figura 27: Tipos de señales de activación de las operaciones .....	40
Figura 28: Non-Sim operation para crear una de las tapas exteriores .....	41
Figura 29: Operación de flujo de materiales del rodamiento de la posición 1 .....	42
Figura 30: Características del sensor fotoeléctrico del alimentador de bulones.....	43
Figura 31: Operación flujo de materiales TapaInt_Abajo_Pos1 .....	43
Figura 32: Comandos Add Location.....	44
Figura 33: Operación PICK_BULON en el Path Editor.....	45

Figura 34: Configuración correcta (arriba) e incorrecta (abajo) de los ejes de un robot.....	46
Figura 35: Comandos OLP cierre pinza.....	47
Figura 36: Fragmento de los comandos OLP de la operación OP200.....	47
Figura 37: Comandos OLP del programa principal MAIN .....	50
Figura 38: Fragmento de los comandos OLP del subprograma Service 2.....	51
Figura 39: Ruta del AMR sobre la célula robotizada .....	53
Figura 40: AMR original (izq.) y AMR con plataforma (dcha.).....	54
Figura 41: Configuración de la cinemática del AMR.....	54
Figura 42: Configuración de la pose Arriba en el AMR.....	55
Figura 43: Sensor fotoeléctrico plataforma AMR.....	56
Figura 44: Configuración de la operación de dispositivo .....	56
Figura 45: Operaciones del dispositivo AMR.....	57
Figura 46: Evento de fijación en la operación Bajar_E1 .....	57
Figura 47: Evento de separación en la operación Subir_E2.....	58
Figura 48: Traslado de la estación mediante el Placement Manipulator .....	58
Figura 49: Modificaciones necesarias en las operaciones de la segunda estación.....	59
Figura 50: Esquema del Material Flow Parte 1 .....	62
Figura 51: Esquema del Material Flow Parte 2 .....	63
Figura 52: Activación de la simulación mediante CEE.....	64
Figura 53: Vista de redes de los PLCs y el HMI.....	66
Figura 54: Programación OB100 del PLC del AMR.....	68
Figura 55: Secuencia funcionamiento automático AMR .....	70
Figura 56: Configuración de los parámetros de la conexión en "TSEND_C" y "TRCV_C" .....	72
Figura 57: Configuración de los parámetros del bloque "TSEND_C" .....	73
Figura 58: Configuración de los parámetros del bloque "TRCV_C" .....	74
Figura 59: Programación OB100 del PLC de las estaciones.....	76
Figura 60: Archivo *.scl y los FBs creados .....	77
Figura 61: Comunicación ODK.....	79
Figura 62: Variables compartidas entre PLC y la unidad de control KR C4 .....	81
Figura 63: Lógica de la generación del bulón de la posición 2.....	82
Figura 64: Lógica del movimiento por gravedad de la tapa interior posición 1 .....	82
Figura 65: Parámetros de la conexión de los bloques "TSEND_C" y "TRCV_C" de las estaciones .....	83
Figura 66: Parámetros del bloque "TSEND_C" de las estaciones .....	84
Figura 67: Parámetros del bloque "TRCV_C" de las estaciones .....	84
Figura 68: Comunicación S7 entre los PLCs y el HMI.....	85

Figura 69: Interfaz HMI.....	88
Figura 70: Comunicación distribuida vía Ethernet.....	90
Figura 71: Transferencia de información entre las diferentes herramientas.....	91
Figura 72: Propiedades Protocolo de Internet Versión 4 (TCP/IPv4) .....	92
Figura 73: Instancia del PLC de la estación 1 en PLCSIM Advanced de un PC Remoto .....	93
Figura 74: Ubicación archivo *.dll en las instancias PLC.....	94
Figura 75: Conexión Tecnomatix PS con las instancias PLC de los PC Remotos.....	95
Figura 76: Signal Viewer.....	95
Figura 77: Simulation Panel durante la simulación del Gemelo.....	96
Figura 78: Interfaces KR C4 compact .....	97
Figura 79: Interface de conexiones del robot KR3 Agilus.....	98
Figura 80: Conexiones del sistema de robot .....	99
Figura 81: Estructura del proyecto de KUKA.WorkVisual.....	100
Figura 82: Ventana Circuito EA para el mapeo entre señales .....	101
Figura 83: Definición de los nombres de las variables .....	102
Figura 84: Vista de redes del control de la puesta en marcha real de una estación .....	103
Figura 85: Diagrama de Gantt del proyecto .....	109

# ÍNDICE DE TABLAS

Tabla I: Procesos multicriterio. Software de simulación de Gemelos .....	16
Tabla II: Matriz de probabilidad e impacto de riesgos .....	19
Tabla III: Datos técnicos KR 3 R540 [17] .....	25
Tabla IV: Servicios a realizar.....	32
Tabla V: Modos de operación de Tecnomatix PS.....	33
Tabla VI: Tipos de elementos que se utilizarán en el proyecto y su descripción.....	34
Tabla VII: Casos de los Pallets de Entrada .....	35
Tabla VIII: Operaciones de robot.....	44
Tabla IX: Señales de robot .....	48
Tabla X: Resumen de operaciones Pick & Place de cada Servicio.....	50
Tabla XI: Variables AMR.....	52
Tabla XII: Variables PLC del AMR.....	66
Tabla XIII: Variables que se reciben desde la primera estación .....	74
Tabla XIV: Variables que se reciben desde la segunda estación.....	75
Tabla XV: Variables que se envían desde el AMR.....	75
Tabla XVI: Índice de las diferentes tablas de variables .....	75
Tabla XVII: Información fichero ODK2PLC .....	78
Tabla XVIII: Variables “TelegramaAgente” .....	78
Tabla XIX: Información del fichero PLC2ODK.....	80
Tabla XX: Fragmento de la Tabla de las variables HMI.....	85
Tabla XXI: Señales de robot que se transfieren.....	102
Tabla XXII: Modos de servicio del robot .....	104
Tabla XXIII: Pasos necesarios para el desarrollo del proyecto.....	105
Tabla XXIV: Precio de los elementos de la estación real.....	110
Tabla XXV: Presupuesto de los elementos de la estación real.....	111
Tabla XXVI: Precios de las herramientas de ingeniería .....	112
Tabla XXVII: Presupuesto de las herramientas de ingeniería.....	112
Tabla XXVIII: Presupuesto grupo de trabajo .....	113
Tabla XXIX: Presupuesto total.....	114

# 1 MEMORIA

## 1.1 Introducción

Actualmente, la transformación digital está en auge, donde el Gemelo Digital, se define como la representación virtual de máquinas o procesos, está revolucionando la industria. Por ello, muchos sectores y empresas apuestan por la utilización del Gemelo Digital para detectar fallos e incrementar la eficiencia de los procesos. Sin embargo, el concepto de Gemelo Digital no es nuevo, su origen reside en tecnologías ya existentes como la simulación, el modelado 3D, etc.

Con el avance de las tecnologías de la información impulsadas por la Industria 4.0, tales como el Internet de las cosas (IoT), *Cloud Computing* (computación en la nube), análisis de *Big Data* y la inteligencia artificial (IA), el proceso de digitalización se está acelerando enormemente [1].

Esta creciente popularidad del Gemelo Digital refleja la tendencia de que el mundo virtual y el mundo físico se vinculan cada vez más entre sí y se integran como un todo [1].

En este contexto, este Trabajo Fin de Máster pertenece y parte del desarrollo del proyecto de investigación financiado por el Ministerio de Ciencia, Innovación y Universidades: “Arquitectura inteligente para la Industria 4.0: Realimentación a través del *Fog*” del departamento de Ingeniería de Sistemas y Automática donde se usa un sistema multiagente para dotar de inteligencia al sistema. Este trabajo es una parte del demostrador en el que se van a validar los desarrollos realizados.

En este proyecto se presenta el desarrollo de un Gemelo Digital de una célula robotizada flexible (véase Figura 1). Esta célula está formada por dos estaciones robotizadas en las cuales se realiza el montaje total o parcial de un conjunto de piezas. Además, el tráfico de pallets entre las dos estaciones se realiza mediante un robot móvil autónomo (AMR, Autonomous Mobile Robot).

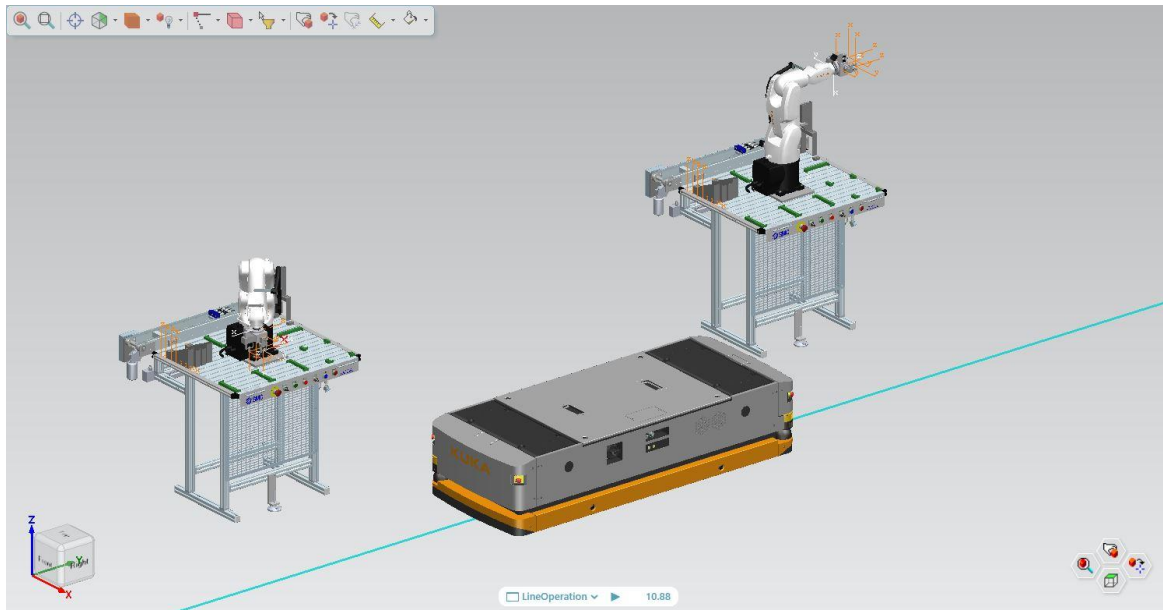


Figura 1: Gemelo Digital célula robotizada flexible

Para explicar el desarrollo del proyecto y comprender mejor su descripción se han desarrollado los apartados que se detallan a continuación.

En el primer apartado denominado Memoria, se explican los siguientes apartados. Se describe el Contexto donde se establece este trabajo y en el apartado de Objetivos y Alcance se definen los objetivos para el desarrollo del mismo. A continuación, se indican los Beneficios Sociales, Económicos y Técnicos que conlleva el proyecto. Se han analizado las diferentes tecnologías y elementos disponibles tal y como se explica en el apartado de Análisis de alternativas. En el apartado Análisis de Riesgos se describen los posibles riesgos que hay que valorar a la hora de llevar a cabo este proyecto. En el último apartado de la Memoria, Descripción de la solución propuesta es el diseño que se plantea para la realización de este proyecto.

En el segundo apartado, Metodología y diseño seguidos en el progreso del trabajo, se detallan y describen las diferentes fases trabajadas. Además, se explica la planificación ejecutada, que también se puede encontrar resumida en un diagrama de Gantt.

En el tercer apartado, denominado Aspectos Económicos, se detalla el Presupuesto del proyecto.

El cuarto apartado recoge las Conclusiones donde se expone si se alcanzan los objetivos establecidos, cuales son las dificultades encontradas y los trabajos futuros.

En el apartado final se detalla la Bibliografía, donde se detallan las fuentes de toda la información utilizada en el desarrollo del trabajo.

## 1.2 Contexto

Desde hace varios años, la sociedad avanzada se encuentra en plena cuarta revolución industrial, comúnmente denominada Industria 4.0.

Desde el inicio de la industrialización, los saltos tecnológicos han dado lugar a cambios de paradigma denominados “Revoluciones industriales”.

La primera revolución industrial, entre los siglos XVIII y XIX, fue la derivada de la invención de la máquina de vapor. La energía generada por esta máquina, promovió la productividad en masa y la mecanización de la industria, transformando la economía agrícola y artesanal [2]. Sin embargo, esa energía tenía deficiencias, por lo que, para un mayor crecimiento industrial, era necesario buscar nuevas fuentes de energía alternativas. Ante esta necesidad, se dio comienzo a la segunda revolución industrial.

La segunda revolución comenzó en el siglo XIX, caracterizada por el aprovechamiento de recursos y nuevas fuentes de energía [2]. El crecimiento de la industria ferroviaria originó la utilización de la electricidad o el petróleo. Además, la fabricación en línea complementó esta segunda revolución.

En los años 70, aparece la tercera revolución, donde surgieron diversos desarrollos tecnológicos, como los microprocesadores y consecuentemente, la programación. Otro de los hitos de esta revolución fue la creación del internet en los años 90.

Actualmente nos encontramos en la cuarta revolución industrial, una nueva revolución que une diferentes tecnologías para poder innovar creando procesos de fabricación inteligentes y más eficientes. Esta evolución ha creado nuevos mercados basados en la personalización y la creación de nuevos productos y servicios.

La tecnología más destacable de la Industria 4.0 es el Internet de las Cosas (IoT) que brindando un entorno de fabricación colaborativa totalmente conectada con la recopilación y el tratamiento de datos [2] y, por ende, la extracción de conocimiento sobre los procesos.

La Industria 4.0 también engloba un conjunto de tecnologías que se refieren a la simulación, la fabricación aditiva, los sistemas de integración horizontal y vertical, la ciberseguridad, la realidad aumentada, el *Cloud Computing*, los robots autónomos, el *Big Data* y la analítica avanzada (véase Figura 2).

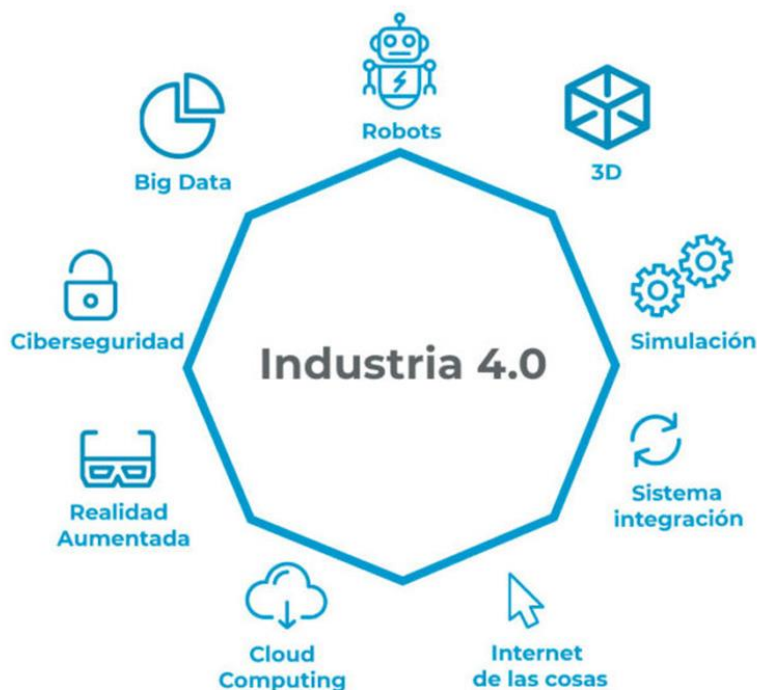


Figura 2: Tecnologías de la Industria 4.0

En la visión de la Industria 4.0, las máquinas serán capaces de comunicarse entre sí para recibir y/o transmitir información y ejecutar acciones; ya que, con el IoT, muchos sensores estarán conectados a la red. Estos sensores generarán grandes cantidades de datos de diferentes tipos (*Big Data*), y el *Cloud Computing* proveerá de almacenamiento y procesamiento a estos grandes volúmenes de datos [3].

A parte del *Cloud Computing*, se han desarrollado el *Fog* y el *Edge Computing*.

Por un lado, el *Cloud Computing* es la utilización de las instalaciones propias de un servidor web albergadas por un proveedor de Internet para almacenar, desplegar y ejecutar aplicaciones [23]. Cuando a nivel industrial se necesita tiempo real, la nube no es adecuada ya que es necesario bajar los servicios.

Por otro lado, en el *Fog Computing* los recursos, datos y aplicaciones se sitúan en algún lugar lógico entre el *Cloud* y la fuente generadora de datos [4]. Se encuentra a nivel de planta y/o proceso (véase Figura 3) y permite a la empresa trabajar en su propia red local y no externamente como el *Cloud*.

En el *Edge Computing* los recursos y aplicaciones están situados en los dispositivos más cercanos a la fuente de datos (véase Figura 3). Por lo que los datos no se envían a una nube remota u otros sistemas centralizados para su procesamiento.

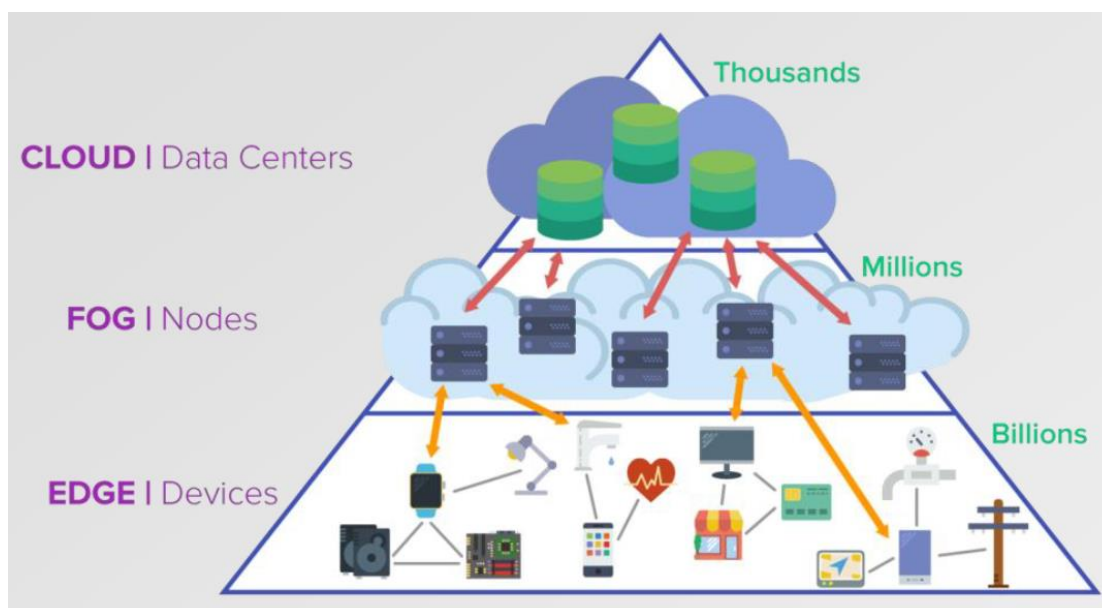


Figura 3: Cloud, Fog y Edge Computing en la pirámide de automatización

Una de las tecnologías de la Industria 4.0 destacable en este proyecto son los robots autónomos fijos y móviles. En los procesos industriales el término autómatas no es nuevo, sin embargo, la robótica ha evolucionado y los robots incorporan nuevas capacidades para trabajar coordinadamente para automatizar tareas logísticas y de producción sin ningún tipo de supervisión.

Por otro lado, otro pilar de la Industria 4.0 es la simulación que permite crear entornos virtuales donde se replica el comportamiento de máquinas o procesos. De esta forma, la simulación permite realizar pruebas y configuraciones óptimas para la fabricación en planta antes de decidir producir el producto [22]. Por lo que aparecen nuevos conceptos como los Gemelos Digitales y el *Virtual Commissioning*.

En los últimos años muchas empresas están evolucionando hacia la transformación digital de las plantas industriales. Los Gemelos Digitales se presentan como una tecnología para la simulación y análisis de procesos industriales, capaz de exprimir al máximo los beneficios de esta transformación [5]. Los Gemelos Digitales son representaciones digitales animadas de máquinas y procesos que componen una fábrica, los cuales permiten obtener unos modelos cuyo comportamiento se asemeja al del sistema real.

Consecuentemente, se consigue un entorno seguro para la experimentación, que junto con el *Virtual Commissioning* permite realizar la puesta en marcha virtual de una máquina o proceso para realizar pruebas en una fase temprana del diseño y desarrollo. Así, se pueden llegar a detectar problemas o errores antes de que sucedan, nuevos planes de fabricación y validar la seguridad, incluso

antes de que exista la máquina y/o el proceso real. Gracias al *Virtual Commissioning*, el tiempo total requerido para la puesta en marcha de la máquina real puede verse reducido considerablemente, eliminando la necesidad de prototipos físicos y reduciendo significativamente el tiempo de comercialización de las máquinas de producción [6].

## 1.3 Objetivos y alcance del proyecto

El objetivo de este proyecto es realizar el montaje de una serie de conjuntos mediante una célula robotizada flexible.

Para ello, se realiza el diseño y el modelado del Gemelo Digital de dicha célula y, posteriormente, se valida su funcionamiento por medio de su puesta en marcha virtual con *Software in the Loop* (SiL) distribuido. Además, se realiza la validación parcial de la célula en una estación real; ya que, actualmente no existen dos estaciones robotizadas reales.

### 1.3.1 Montaje de los conjuntos

Los conjuntos a montar pueden estar formados por los siguientes elementos: base, rodamiento, bulón, tapa interior y tapa exterior (véase Figura 4). El objetivo de este trabajo es implementar todos los montajes posibles. De esta forma se abarcan todas las posibilidades de montajes de los conjuntos desde simplemente montar el rodamiento sobre la base hasta el montaje del conjunto completo.

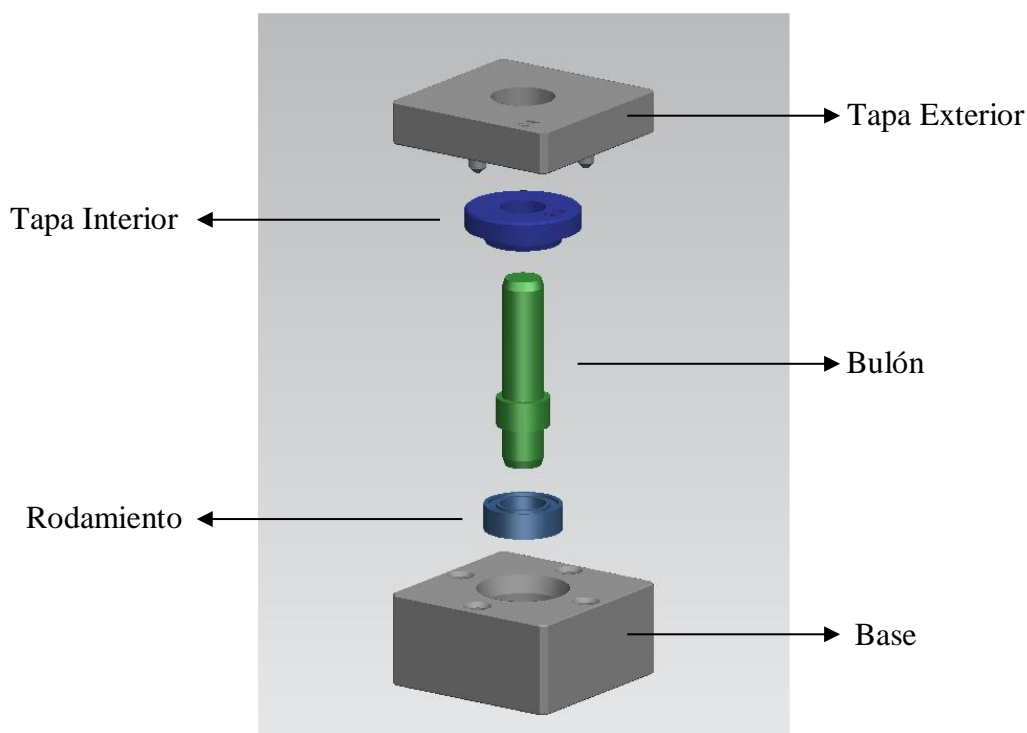


Figura 4: Elementos de los conjuntos a montar

### **1.3.2 Gemelo Digital de la célula flexible**

Uno de los objetivos transversales de este proyecto es modelar el Gemelo Digital de la célula flexible compuesta por dos estaciones robotizadas en las cuales se realiza el montaje total o parcial de un conjunto de piezas. Para la transferencia de pallets de una estación a otra también se modela un AMR.

El objetivo fundamental de este Gemelo Digital es representar fielmente de manera virtual el comportamiento de la célula robotizada flexible, por ello, las estaciones están modeladas minuciosamente. De esta forma, se valida el funcionamiento de la célula para detectar errores y solucionarlos antes de realizar la implementación real. Esta validación se efectúa independientemente del control y posteriormente, dicho Gemelo se automatiza mediante un control externo.

### **1.3.3 Programa de control y puesta en marcha virtual distribuida**

Una vez validado el Gemelo sin un programa de control, el siguiente objetivo es diseñar el sistema de control capaz de controlar el funcionamiento de toda la célula.

Cada robot está controlado por su unidad de control en el cual mediante el programa robot se controlan las trayectorias del robot, así como el cierre y la apertura de las pinzas.

Por ello, la finalidad del programa de control es automatizar estas unidades de control para verificar el correcto funcionamiento del control de la célula antes de la implantación real. Se pretende que los tres componentes del Gemelo Digital (las dos estaciones y el AMR) funcionen en *Software in the Loop* cada uno con un controlador PLC virtual. Asimismo, la puesta en marcha virtual será distribuida, puesto que, los PLC virtuales se ejecutarán en PCs diferentes. Con esto se pretende disminuir el peso del uso del procesador en el PC donde se ejecuta el Gemelo Digital mejorando el rendimiento de la simulación y disminuyendo el coste computacional del mismo.

### **1.3.4 Implantación real**

Una vez realizada la comprobación y solución de los posibles errores durante la puesta en marcha virtual se pasa a la implantación real cuyo propósito es la validación parcial de las estaciones robotizadas en una estación real. Esto es debido, como se ha comentado anteriormente, a que solo se dispone de una estación real.

En esta fase, se comprueba la gestión desde el sistema multiagente de las solicitudes de montaje con el PLC real y si el Gemelo Digital está modelado correctamente o hay que realizar algún pequeño ajuste para que el montaje de conjuntos sea adecuado.

## 1.4 Beneficios que aporta el trabajo

A continuación, se describen los beneficios económicos, sociales y técnicos que aporta este proyecto.

### 1.4.1 Beneficios económicos

En cuanto a los beneficios económicos, debido a la introducción de la Industria 4.0, hay una creciente demanda de las fábricas inteligentes orientadas a producto. Para ello, es necesario que los sistemas de producción sean más flexibles, y en consiguiente, es necesario que los tiempos de desarrollo y producción se minimicen.

El Gemelo Digital permite poder realizar una simulación de un proceso de producción sin que éste haya sido aún implantado, es posible detectar errores en los procesos y corregirlos antes de la puesta en marcha real. Por lo tanto, se consiguen procesos con mayor rendimiento y se reducen los costes, ya que una vez realizada la implantación real del proceso corregir los errores es económicamente costoso.

### 1.4.2 Beneficios sociales

La automatización de procesos dentro de la Industria 4.0 supone mayor seguridad y comodidad para los/as trabajadores/as, porque no es necesario realizar esfuerzos físicos y poner en riesgo su integridad física en entornos hostiles. Además, la introducción de nuevas tecnologías permite monitorizar los datos del proceso en tiempo real lo que posibilita la reducción de errores en el proceso de producción debido a que cualquier fallo se detecta al instante para poder solucionarse lo antes posible.

El uso de células robotizadas permite hacer tareas muy repetitivas sin ningún tipo de fallo y con un ritmo de producción sin interrupciones.

### 1.4.3 Beneficios técnicos

Para llevar a cabo este proyecto se han utilizado diferentes tecnologías y herramientas de última generación, las cuales aportan diferentes beneficios.

El Gemelo Digital se utiliza para la verificación de procesos de manufactura en un entorno 3D [8]. Entre los beneficios que aporta se encuentran [7]:

- La mejora de la productividad del proceso de planificación
- Validación exhaustiva de los procesos de fabricación
- Optimización de los tiempos de ciclo de los procesos
- Minimización de los riesgos de producción gracias a la simulación

La incorporación de un robot móvil autónomo (AMR) a la célula robotizada permite la evolución de la flexibilización de la producción industrial, concepto impulsor de la Industria 4.0. Los AMRs se han convertido en herramientas indispensables para el traslado de mercancías en planta de forma rápida, segura y eficiente. Ya que se pueden adaptar al nivel de flexibilidad de cualquier sector industrial y garantizan mayor seguridad laboral y menor riesgo de accidentes [24].

KUKA.WorkVisual ofrece un entorno de desarrollo offline homogéneo, un entorno de diagnóstico online y de mantenimiento. El cual permite configurar, programar, poner en servicio o diagnosticar los robots de KUKA. Este entorno ofrece una configuración integrada de módulos E/S de bus de campo, así como diagnóstico a Robot Control y PLC real entre ellos. Además, permite la edición de programas de control directamente en el entorno de ingeniería [9].

Para el control de la estación real se utiliza el SIMATIC ET 200SP Open Controller, la CPU 1515SP PC2 (F), es un sistema de control compacto y robusto que combina la función de un controlador de la familia ET 200SP con una plataforma basada en PC. Gracias a esta CPU que soporta Windows se puede usar la aplicación Siemens ODK para el sistema multiagente. Además, permite conectar periferia distribuida con el adaptador PROFINET del sistema ET 200SP. [10]

Para la puesta en marcha virtual se emplea el PLC S7-1500 con CPU 1518-4 PN/DP ODK. Al igual que el Open Controller, esta CPU permite utilizar Siemens ODK para la comunicación entre el PLC y Windows para comunicarse con el sistema multiagente y recibir las peticiones de montaje a ejecutar.

Para el desarrollo del sistema de control se ha utilizado la herramienta TIA Portal V16, la cual permite dar una solución de automatización eficiente para los procesos de ingeniería. Al tratarse de una aplicación modular se pueden ir añadiendo módulos según las necesidades. Por lo que en una sola plataforma se pueden tener diferentes elementos de automatización y ofrece una puesta en marcha rápida y un diagnóstico de detección de errores [12]. Igualmente, el paquete de TIA Portal incluye PLCSIMAdvanced, gracias al cual se ha realizado la puesta en marcha virtual, ya que permite la simulación de la funcionalidad de un PLC o varios, incluyendo la comunicación, así como la

interacción con los modelos de planta. Asimismo, permite que esta puesta en marcha virtual sea distribuida, ya que mediante el PLCSIM Virtual Ethernet Adapter las instancias de PLC pueden correr en diferentes PCs.

## 1.5 Análisis de alternativas

Para satisfacer las especificaciones del proyecto, en este apartado se analizarán diferentes opciones o soluciones. Para ello, se investigarán diferentes aspectos y se seleccionarán las mejores soluciones. Los diferentes aspectos que se analizarán serán los siguientes: el software para el desarrollo de Gemelos Digitales y tipos de arquitecturas para la puesta en marcha virtual.

### 1.5.1 Software de desarrollo de Gemelos Digitales

#### 1.5.1.1 Software de desarrollo de Gemelos Digitales. Descripción

A la hora de realizar el Gemelo Digital, es imprescindible realizar un análisis entre los diferentes softwares que existen para modelar y simular Gemelos Digitales. Seguidamente, se describen las alternativas que se contemplan para dicho propósito.

- **Tecnomatix Process Simulate:**

Siemens Tecnomatix Process Simulate es un software para diseño y verificación de células robotizadas multirobot en un entorno 3D. También simula los procesos de ensamblaje del producto, las interacciones con las herramientas complementarias al robot y valida la lógica del PLC de la instalación interactuando con la programación offline de los robots [7].

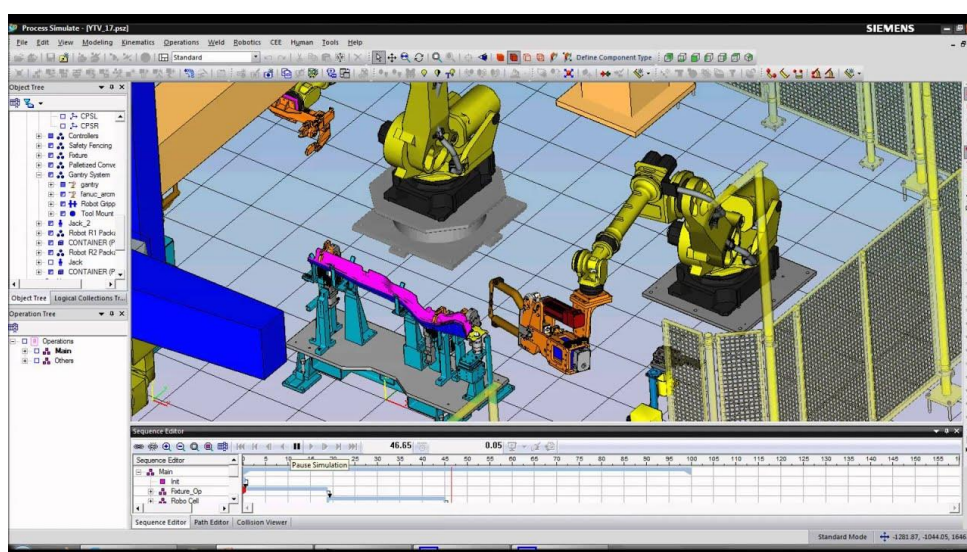


Figura 5: Entorno de Tecnomatix Process Simulate

## - KUKA.Sim 4.0:

KUKA.Sim 4.0 es un software permite solamente realizar la programación offline de los robots KUKA de forma eficiente. KUKA.Sim es capaz de realizar gemelos digitales y abarca toda la planificación desde el diseño del proceso, la visualización de los flujos de materiales hasta el código del PLC [16].

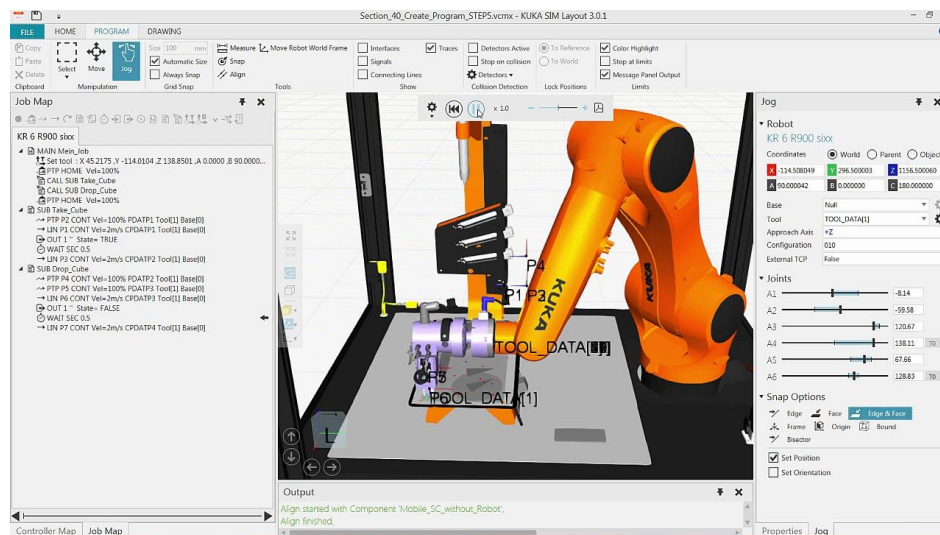


Figura 6: Entorno de KUKA.Sim 4.0

## - RoboDK

RoboDK es un software potente y rentable para realizar Gemelos de procesos de manufactura con robots industriales de diferentes fabricantes y su programación. RoboDK permite la programación offline de los robots, eliminando el tiempo de inactividad de la producción causado por la programación online del taller [17].

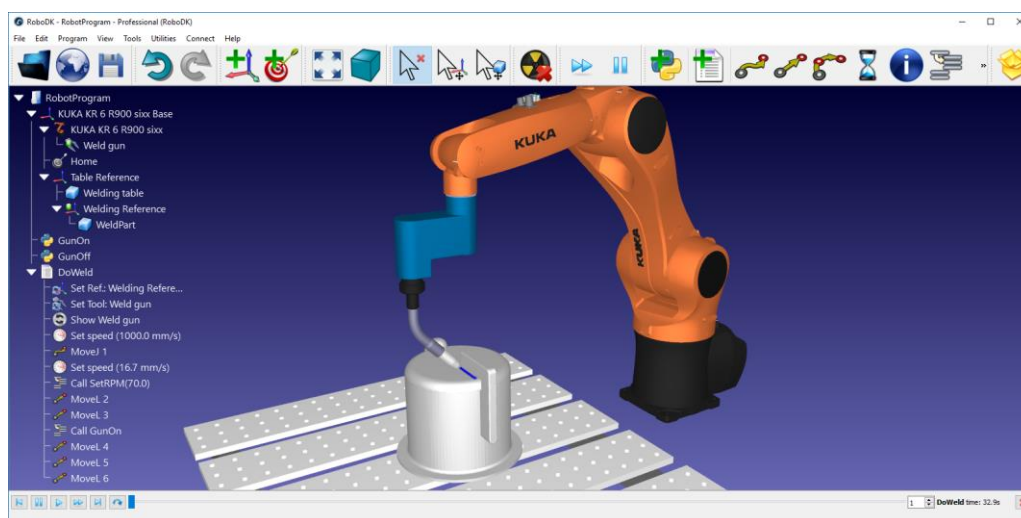


Figura 7: Entorno RoboDK

### 1.5.1.2 Software de simulación de Gemelos Digitales. Análisis

Para determinar la opción más adecuada se utilizará el método de Unificación Ponderada. En este método se tienen en cuenta todos los factores de ejecución del proyecto, incluidos los financieros. Posteriormente, se fijará la supuesta importancia a cada factor. Finalmente, a los factores preestablecidos se les asigna una calificación y, aplicando la siguiente fórmula, se elige la opción más adecuada.

$$SP = \sum_1^n p_i \cdot x_i$$

donde:

p: Peso de los factores

x: Calificación de los factores

Para la elección del software de simulación se tendrá en cuenta los siguientes factores: versatilidad, la facilidad de uso y el coste de esta tecnología (véase Tabla I).

Para establecer los pesos de los factores de la Tabla I, se ha empezado con el factor más importante que es la conectividad, atribuyéndole un peso del 40%. La conectividad es importante puesto que la célula está compuesta por dos estaciones robotizadas y por un AMR, el control de éstos está soportado por un programa control realizado en TIA Portal. Tecnomatix es capaz de usar varias instancias de PLCSIM Advanced tanto en local como en remoto, por esta razón su calificación es la más alta con un 9. Seguidamente, KUKA.Sim es capaz de interaccionar con el programa control que corre en PLCSIM Advanced, pero a través de la plataforma SIMIT por lo que es necesario el uso de otra plataforma, mientras que en Tecnomatix PS la conexión es directa, por ello su calificación es un 8. Por último, RoboDK no permite hacer *Virtual Commisioning* mediante PLCSIM Advanced. Debido a esto su calificación es de 5.

Por otro lado, la versatilidad en este tipo de softwares también es muy importante porque están pensados para representar en un entorno 3D un proceso de producción, así pues, el peso del factor de versatilidad es del 40%. Por ello, Tecnomatix PS y KUKA.Sim son mejores que RoboDK, debido a que este último no puede representar con cierta similitud la realidad de un flujo de materiales. Sin embargo, KUKA.Sim sólo puede realizar Gemelos Digitales con robots KUKA, pero tal y como se ha

dicho anteriormente, el objetivo es replicar un proceso de producción y éstos normalmente pueden tener robots de diferentes fabricantes. En conclusión, la calificación de RoboDK es de 5, la de KUKA.Sim 4.0 un 7 y la de Tecnomatix PS un 8.

Por último, está la facilidad de uso con un peso del 20%. Este peso es debido a que realizar el modelado y la simulación de un Gemelo Digital es de por sí complejo por lo que es preferible elegir una herramienta de uso sencillo. Los softwares más complejos de utilizar son Tecnomatix PS y KUKA.Sim 4.0, ya que para hacer una tarea sencilla de Pick and Place hay que tener en cuenta no solo la trayectoria, sino también que el flujo de materiales sea correcto, cosa que no ocurre con el RoboDK. Por ello, la calificación de Tecnomatix es de un 5 y la calificación de RoboDK es de 7.

Tabla I: Procesos multicriterio. Software de simulación de Gemelos

FACTORES	PESOS	CALIFICACIÓN		
		RoboDK	KUKA.Sim 4.0	Tecnomatix PS
Conectividad	%40	5	8	9
Versatilidad	%40	5	7	8
Facilidad de uso	%20	7	5	5
SP		5.4	7	7.8

A la vista de los resultados de la Unificación Ponderada (SP) de la Tabla I, se puede decir que **Tecnomatix Process Simulate** es aquel que se adapta mejor a las demandas del proyecto.

## 1.5.2 Puesta en marcha virtual

A la hora de realizar la puesta en marcha virtual es necesario definir qué tipo de arquitecturas existen y cuál de ellas es la adecuada para este proyecto.

### 1.5.2.1 Tipos de puesta en marcha virtual. Descripción

#### - Hardware in the Loop (HiL)

La simulación HiL es una técnica para realizar simulaciones a nivel de sistema de sistemas integrados de una manera integral, rentable y repetible cuando esos sistemas no se pueden probar fácil, completa y repetidamente en sus entornos operativos. La simulación HiL requiere el desarrollo de una simulación en tiempo real que modele ciertas partes del sistema integrado y las interacciones significativas con su entorno operativo. Las salidas del sistema integrado son las entradas para la simulación y las salidas que genera la simulación son las entradas para el sistema integrado.

Por lo tanto, para poder realizar HiL es necesario el hardware E/S y el software para realizar el modelado de simulación en tiempo real; es decir, se necesita, en este caso, el PLC físico para poder realizar la simulación del sistema [13].

#### - Software in the Loop (SiL)

La simulación SiL es una técnica que se puede realizar en las primeras etapas del desarrollo del software. Ofrece la posibilidad de ejecutar pruebas antes de que el hardware esté disponible y así detectar errores [14]. Además, permite probar y modificar directa e iterativamente el código fuente, conectando directamente el software a un modelo de planta digital en sustitución de sistemas, prototipos o bancos de prueba más costosos.

Por lo tanto, para poder realizar SiL no es necesario el hardware E/S por lo que la simulación se puede realizar en un PLC virtual, ya que tanto la aplicación software como la simulación están en la misma plataforma PC.

### **1.5.2.2 Tipos de puesta en marcha virtual. Análisis**

Para realizar la elección de qué tipo de arquitectura de puesta en marcha virtual es adecuada para este proyecto, es necesario tener en cuenta varios factores. El SiL permite probar el software antes de la inicialización de la fase de creación de prototipos de hardware, lo que acelera significativamente el ciclo de desarrollo. Mientras que la simulación HiL se ejecuta sobre un sistema embebido, o en este caso, un PLC físico. Por otro lado, la simulación en SiL ofrece la ventaja de la flexibilidad, no se requieren equipos de hardware costosos, pero su principal inconveniente es que el tiempo de simulación es completamente diferente al esperado de un sistema en tiempo real, sin embargo, la simulación HiL es en tiempo real.

Por otro lado, hay que tener en cuenta de que no se dispone del PLC real para poder realizar una simulación con su hardware. Por lo que es necesario un PLC virtual para realizar la puesta en marcha virtual de este proyecto, por consiguiente, en este proyecto se usará **SiL** para realizar la puesta en marcha virtual.

## 1.6 Análisis de Riesgos

Entre los posibles riesgos, que podría conllevar este proyecto, se encuentran un diseño y modelado inexacto del Gemelo Digital de la célula, ya que el gemelo tiene que ser muy parejo a la célula. Errores en el código del programa de control, la posible incompatibilidad entre las herramientas de ingeniería utilizadas para realizar el Gemelo Digital y el software de control y los posibles fallos que se pueden dar al realizar la puesta en marcha real de la célula robotizada.

En este apartado se describen dichos riesgos, la probabilidad de que ocurran, su nivel de impacto y en consiguiente, el nivel de riesgo que podrían llegar a producir (véase Tabla II).

Tabla II: Matriz de probabilidad e impacto de riesgos

		IMPACTO				
PROBABILIDAD		Muy Bajo	Bajo	Medio	Alto	Muy Alto
	Muy Baja	Bajo	Bajo	Bajo	Medio	Medio
	Baja	Bajo	Bajo	Medio	Medio	Alto
	Media	Bajo	Medio	Medio	Alto	Alto
	Alta	Bajo	Medio	Medio	Alto	Alto
	Muy Alta	Medio	Medio	Alto	Alto	Alto

### 1.6.1 Error en el diseño y modelado del Gemelo Digital

El Gemelo Digital ha de ser lo más fiel posible a la célula real. Lograr una exactitud total es casi imposible, por lo que siempre va a existir alguna diferencia entre el Gemelo Digital y la célula real. Sin embargo, si esa diferencia es alta, podría causar errores graves en la puesta en marcha real tales como el daño del utillaje y la fabricación errónea y/o con desperfectos del producto final.

Por lo tanto, al tratarse de un riesgo con probabilidad Muy Alta e impacto Muy Alto, la puntuación de este riesgo es **Muy Alto** (véase Tabla II).

El plan de contingencia sería realizar un modelado del Gemelo Digital minucioso para que el error entre el gemelo y la célula real sea el mínimo posible. Además de comprobar que no haya ningún tipo de colisión entre los elementos de la célula.

### **1.6.2 Error en el programa de control**

Puede ocurrir que haya un error en el código del programa de control y no haya sido detectado durante su depuración en la puesta en marcha virtual cuya probabilidad es media. Debido a un fallo de código, la ejecución del código puede ser errónea, lo que podría ocasionar desperfectos en los elementos de la célula real.

Por lo tanto, al tratarse de un riesgo con probabilidad Media e impacto Muy Alto, la puntuación de este riesgo es **Muy Alto** (véase Tabla II).

El plan de contingencia sería realizar la depuración del programa mediante puesta en marcha virtual teniendo en cuenta todos los posibles casos para así disminuir la probabilidad de una mala ejecución del programa en la puesta en marcha real.

### **1.6.3 Error en la compatibilidad de las herramientas de ingeniería**

En este proyecto, las herramientas de ingeniería utilizadas son Tecnomatix Process Simulate, TIA Portal V16 y PLCSIM Advanced V3 Upd 2, las tres pertenecen al fabricante Siemens por lo que la incompatibilidad de estas herramientas es muy improbable. En cambio, si existiera alguna incompatibilidad esto supondría no poder realizar la simulación y validación del gemelo por lo que su impacto sería Alto.

Por lo tanto, al tratarse de un riesgo con probabilidad Muy Baja e impacto Alto, la puntuación de este riesgo es **Medio** (véase Tabla II).

El plan de contingencia sería realizar una búsqueda exhaustiva de las diferentes herramientas para realizar el Gemelo Digital y el programa de control para cerciorarse que son compatibles.

### **1.6.4 Error en la puesta en marcha real**

En el momento de hacer la puesta en marcha real, los programas robot realizados en Tecnomatix Process Simulate se deben cargar correctamente en la unidad de control del robot para que ésta pueda ejecutarlos. Para ello, es necesario tener una buena comunicación entre la célula y la

unidad de control además de su configuración hardware. Estas configuraciones podrían no realizarse correctamente debido a la cantidad de variables que hay en el proyecto por ello la probabilidad de que ocurra este error es alta. En cuanto al impacto del error, se califica como medio, porque en el caso que suceda, el robot no se podría mover.

Por lo tanto, al tratarse de un riesgo con probabilidad Alta e impacto Medio, la puntuación de este riesgo es **Medio** (véase Tabla II).

El plan de contingencia sería realizar un Excel con las variables mapeadas en el PLC y en WorkVisual.

## 1.7 Descripción de la solución propuesta. Diseño

En este apartado se describen cuáles son los elementos que forman la célula robotizada flexible. Así como la descripción funcional de la célula.

### 1.7.1 Descripción funcional de la célula robotizada flexible

La célula robotizada está compuesta por dos estaciones robotizadas y un AMR (véase Figura 1). En cada una de estas estaciones hay un único robot que realiza el montaje total o parcial de diferentes conjuntos de piezas. Ambas estaciones tienen la misma funcionalidad, por lo que pueden montar los mismos conjuntos. Mientras que el AMR realiza el tráfico de pallets entre estaciones.

Los montajes a realizar por las estaciones pueden tener los siguientes componentes: base, rodamiento, bulón, tapa interior y tapa exterior (véase Figura 4).

El AMR transporta en primer lugar, el pallet desde el almacén de entrada a la primera estación, una vez terminado el montaje de la primera estación, transporta el montaje parcial a la segunda estación y una vez terminado el montaje de la segunda estación, el montaje final pasa al almacén de salida. Finalmente, el AMR vuelve vacío al almacén de entrada, y se queda esperando a nuevas peticiones de montaje.

### 1.7.2 Estaciones robotizadas

La célula está formada por dos estaciones robotizadas idénticas. Cada una de ellas tiene un robot para la manipulación de los elementos. Este robot está situado en el punto óptimo para que pueda alcanzar todos los elementos de la estación, incluidos los pallets de la plataforma del AMR. Además, estas estaciones están dotadas de los alimentadores de las piezas para realizar los montajes.

Seguidamente se detallan los elementos que conforman ambas estaciones.

#### 1.7.2.1 Mesa de trabajo

La mesa de trabajo del fabricante SMC está dividida en varias zonas (véase Figura 8).

La zona de entrada, donde el robot transporta la entrada de pallets desde el AMR a la mesa de trabajo.

La zona de montaje, donde el robot realiza las tareas pertinentes de *Pick and Place* para completar el montaje que se le haya asignado desde el sistema multiagente.

La zona de entrada de tapas exteriores, dependiendo el montaje a realizar, en ocasiones es necesario montar las tapas exteriores en el conjunto, por ello, en la mesa de trabajo existe una zona donde se alimentan estas tapas en un pallet.

Por último, la zona de salida donde una vez completado el montaje, el robot coloca el pallet. Posteriormente, el robot coloca nuevamente el pallet en el AMR para que éste pueda llevar el pallet a la siguiente estación o al almacén de salida de la célula.

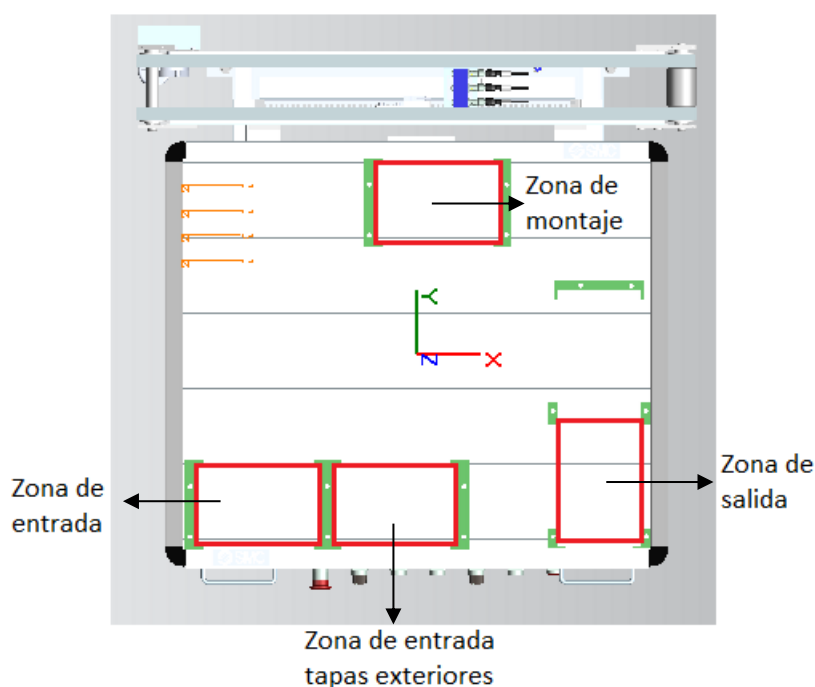


Figura 8: Zonas de la mesa de trabajo

### 1.7.2.2 Alimentadores de piezas

Además de las diferentes zonas de la mesa de trabajo, en la mesa también se dispone de diferentes alimentadores de los elementos que pueden constituir los conjuntos montados en la estación. Estos elementos son los rodamientos, bulones y tapas interiores.

El alimentador de rodamientos es de gravedad con rampa, por la cual se deslizan los rodamientos (véase Figura 9).

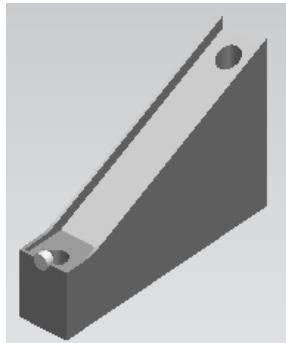


Figura 9: Alimentador rodamientos

El alimentador de bulones es vertical por gravedad. Está construido de tal forma en que la garra del robot no pueda dañar el alimentador (véase Figura 10).

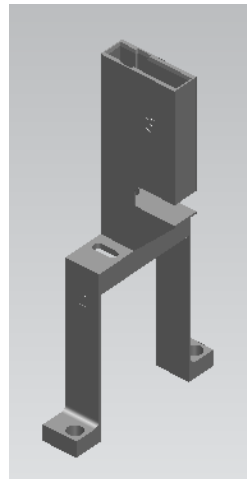


Figura 10: Alimentador de bulones

Al igual que el alimentador de rodamientos, el alimentador de tapas interiores es por gravedad con rampa (véase Figura 11). Sin embargo, este alimentador tiene la forma interior diferente para poder adaptarse a la forma de las tapas interiores para que su deslizamiento sea óptimo y las tapas interiores no se caigan del alimentador.

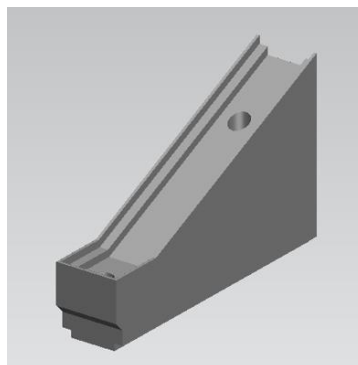


Figura 11: Alimentador de tapas interiores

A parte de estos alimentadores, también se tienen otros dos alimentadores de rodamientos y tapas interiores (véase Figura 12) para futuros trabajos donde se empleen conjuntos de piezas más pequeñas.

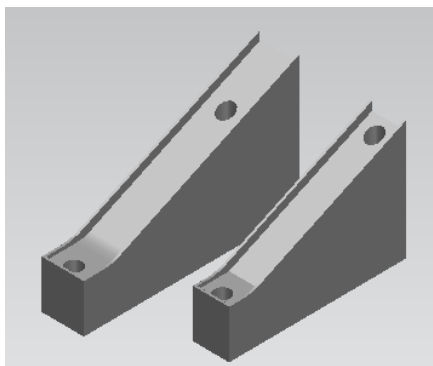


Figura 12: Alimentadores piezas pequeñas

### 1.7.2.3 Robot KUKA KR 3 R540

Este robot está colocado en las dos estaciones robotizadas, su función es la de transportar del AMR a la mesa de trabajo el pallet correspondiente y montar los rodamientos, bulones, tapas interiores y/o tapas exteriores. Por ello, el robot seleccionado es el robot del fabricante KUKA de la gama Agilus: KR 3 R540 (véase Tabla III y Figura 13), el cual es un robot antropomórfico de seis ejes. Cuya capacidad de carga de 3 Kg es suficiente para la manipulación de los elementos de los conjuntos, ya que éstos tienen un muy poco peso. Además, el alcance máximo que tiene el robot es adecuado para llegar a los componentes de la célula incluido el AGV.

Este robot ha sido fabricado para producir las piezas más pequeñas de forma ágil, eficiente y con la máxima precisión [17] gracias a su buena repetitividad de posición  $\pm 0.02$  mm, lo hace perfecto para cumplir su propósito.

Tabla III: Datos técnicos KR 3 R540 [17]

Alcance máximo	541 mm
Carga máxima	3 kg
Repetitividad de posición	$\pm 0.02$ mm
Número de ejes	6
Peso	Aprox. 26.5 kg

Unidad de control

KR C4 compact  
KR C5

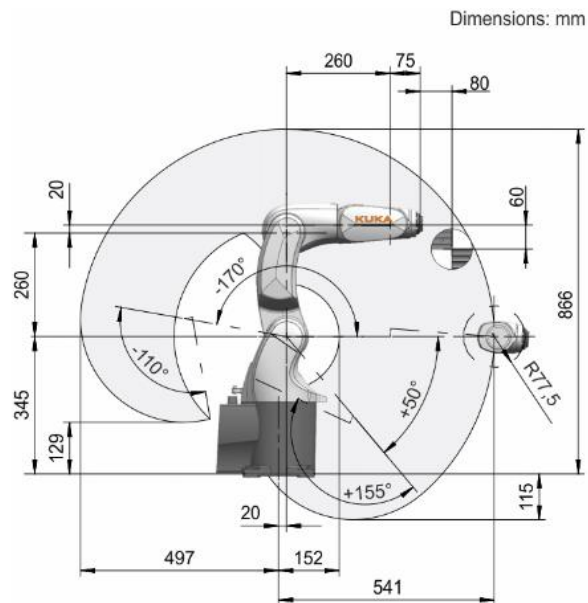


Figura 13: Gráfica del campo de trabajo

#### 1.7.2.3.1 Unidad de control del robot KR C4 compact

La unidad de control que se utiliza en este proyecto es la KR C4 Compact, puesto que su diseño es el más compacto de toda la gama, es la unidad de control más idónea. La arquitectura del software KR C4 integra Robot Control, PLC Control, Motion Control (por ejemplo, KUKA.CNC) y Control de seguridad [18]. Igualmente, comprende lenguajes de PLC, factor indispensable debido a que en este proyecto es necesario el intercambio de datos entre el PLC y la unidad de robot a la hora de realizar la puesta en marcha real.

#### 1.7.2.4 Herramienta del robot

La herramienta del robot ha de realizar el montaje de los diferentes elementos, para ello, la herramienta está formada por dos pinzas neumáticas (véase Figura 14). Ambas pinzas disponen de un par de sensores inductivos para dar información acerca de la posición en la que se encuentran las pinzas.

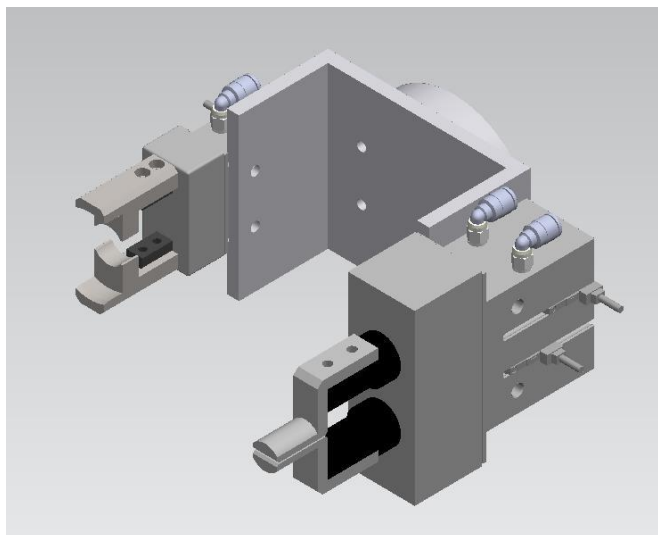


Figura 14: Modelo de la herramienta del robot

Una de las pinzas es el modelo MHK2-16D del fabricante SMC (véase Figura 14 o Figura 14, izquierda). Esta pinza neumática de apertura paralela de doble efecto ha sido diseñada para aplicaciones de gran precisión. Esta pinza Normalmente Abierta (NA) con los dedos adecuados, es la encargada de manipular los pallets y los bulones, ya que es de amarre externo y la carrera del dedo es estándar.



Figura 15: Pinza MHK2-16D

La otra pinza es el modelo MHKL2-25D del fabricante SMC (véase Figura 16 o Figura 14, izq.). Al igual que la pinza MHK2-16D está pensada para aplicaciones de gran precisión, es de doble efecto y destaca por su repetitividad de  $\pm 0.01\text{mm}$ . Esta pinza es Normalmente Cerrada (NC) y es la encargada de manipular los rodamientos, tapas interiores y exteriores gracias a su amarre interno y su carrera del dedo que es de carrera larga de 22 mm.

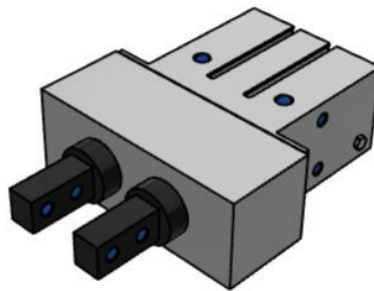


Figura 16: Pinza MHKL2-25D

#### 1.7.2.4.1 Electroválvulas

Para el control de la apertura/cierre de las pinzas de doble efecto se utilizan las electroválvulas SY3100-5UD del fabricante SMC (véase Figura 17). La válvula de solenoide de puerto 5/2 SMC serie SY3000 tiene una presión de funcionamiento que oscila entre 0,1 MPa y 0,7 MPa. Su tensión de solenoide es 24VDC y es una interfaz entre controladores electrónicos y sistemas neumáticos [19].

Al accionarse el solenoide entra aire en el circuito cerrando las pinzas neumáticas, la apertura de las pinzas se lleva a cabo porque las electroválvulas tienen retorno por muelle.

#### 2-position single

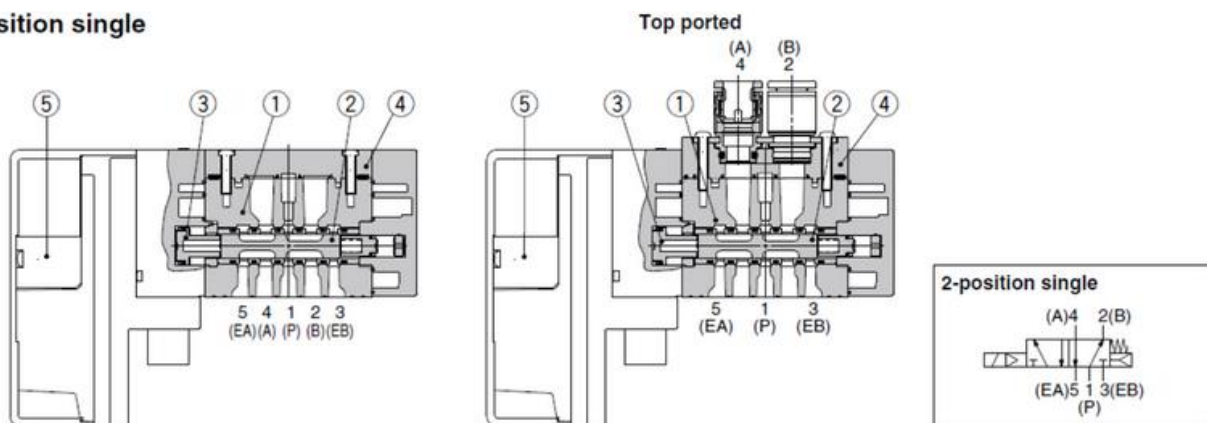


Figura 17: Válvula de solenoide de puerto 5/2 SMC serie SY3000

#### 1.7.2.4.2 Nodo EtherCAT

Este nodo está compuesto por el acoplador y los terminales EtherCAT de E/S digitales.

El acoplador EtherCAT EK1100 es el enlace entre el protocolo EtherCAT a nivel de bus de campo y los terminales EtherCAT (véase Figura 18). Estos terminales constan de 16 canales son el terminal de entradas digitales EL1809 y el terminal de salidas digitales EL2809. El terminal de entradas digitales está conectado a los sensores de las pinzas y el terminal de salidas digitales al solenoide de las electroválvulas de las pinzas de la herramienta del robot. El acoplador está conectado a la unidad de control del robot puesto que mediante el programa robot se activan las variables que accionan la electroválvula abriendo/cerrando las pinzas y recibiendo información acerca de los sensores de las pinzas.

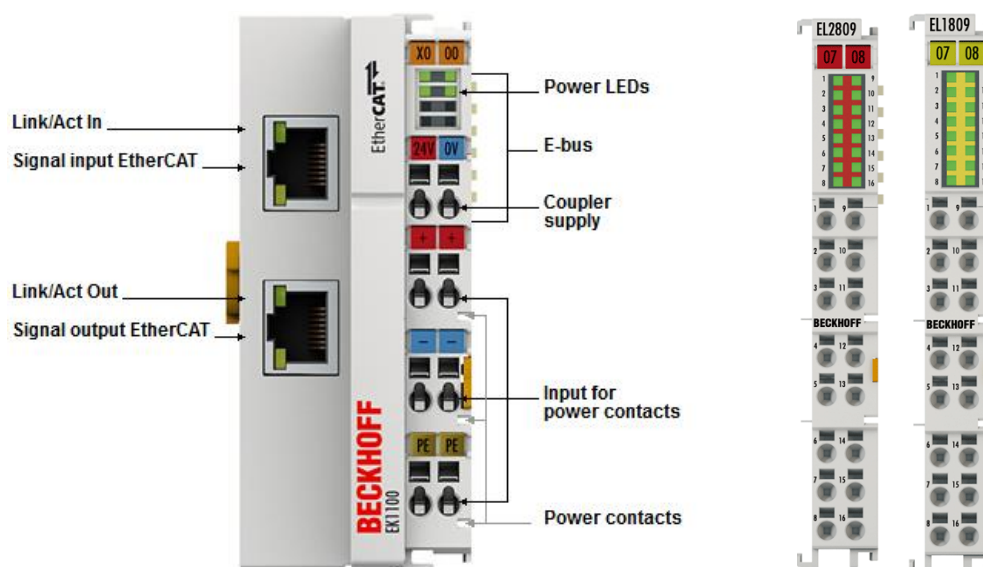


Figura 18: Nodo EtherCAT

### 1.7.3 AMR

Para el tráfico entre las piezas se utiliza un AMR (véase Figura 19), el cual dispone de una plataforma elevadora para que el robot tenga acceso a los pallets para realizar el montaje del conjunto.

Como se ha dicho anteriormente, este AMR sólo estará en el modelo del Gemelo Digital, es decir, no será implementado físicamente, debido a que en el proyecto del departamento se está trabajando paralelamente con AMRs reales, pero aún están en fase de desarrollo.

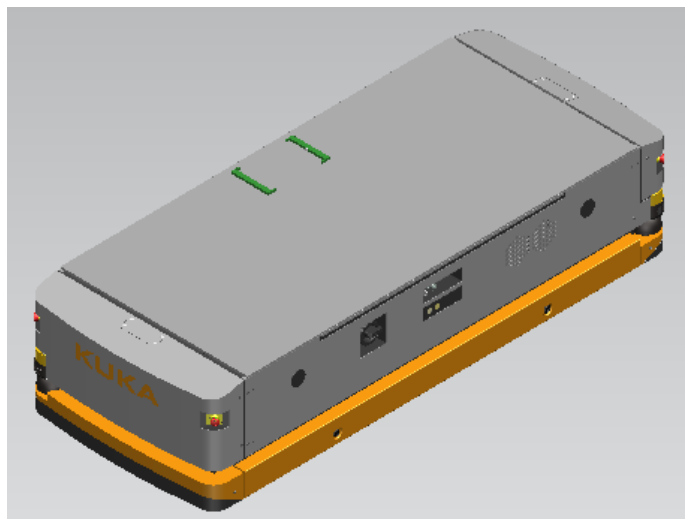


Figura 19: Modelo del AMR

## 2 METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO

En este documento, en primer lugar, se realiza una descripción de las fases y procedimientos seguidos para el desarrollo de la solución del proyecto y se indican los recursos empleados para ello. Asimismo, se presenta el diagrama de Gantt que indica cuál ha sido la planificación de este Trabajo Fin de Máster donde se incluyen las tareas, fases y cronograma.

### 2.1 Descripción de fases, tareas, equipos o procedimientos

En este apartado se describen los diferentes procedimientos que se han tenido que ejecutar para desarrollar este proyecto.

Se ha formado el Gemelo Digital de una estación robotizada resaltando el diseño de los servicios. Estos servicios han sido validados en puesta en marcha virtual. También se explica la configuración hardware la estación y su posterior puesta en marcha real.

A continuación, una vez terminado el modelado de los servicios del Gemelo Digital de una de las estaciones, se detalla cómo se ha formado la célula robotizada añadiendo otra estación y para el tráfico entre pallets se ha modelado un AMR. Una vez finalizado, se pasa a realizar el programa de control del mismo y se valida el modelo creado mediante la puesta en marcha virtual distribuida. En este caso, no se puede realizar la puesta en marcha real porque solo existe una única estación real.

#### 2.1.1 Servicios

Las estaciones han de llevar a cabo unas operaciones para completar el montaje de los conjuntos. Por ello, cada operación a realizar se identifica con un servicio.

Los servicios indican qué montaje se va a realizar en cada una de las estaciones. Es decir, los servicios definen qué tareas u operaciones tienen que realizar los robots de las estaciones. Existen diez servicios, los cuales abarcan todas las combinaciones de montaje posibles de un conjunto que puede estar formado por los elementos: rodamiento, bulón, tapa interior y tapa exterior. Mediante estos servicios se realiza el montaje de al menos un elemento de los mencionados y el montaje final puede ser completo o parcial.

Tabla IV: Servicios a realizar

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

La solicitud de qué servicio se quiere realizar viene dada por un sistema multiagente. Este sistema no sólo dice qué servicio se debe realizar en cada estación, sino también qué número de ítems se deben montar. El número máximo de ítems por cada pallet es de seis (véase Figura 20) y el pallet de entrada puede llegar a cada estación en diferentes situaciones (véase Tabla IV, Situaciones de entrada) dependiendo del servicio que se haya solicitado. Sin embargo, en todas estas situaciones es necesario que en el pallet estén las bases montadas.

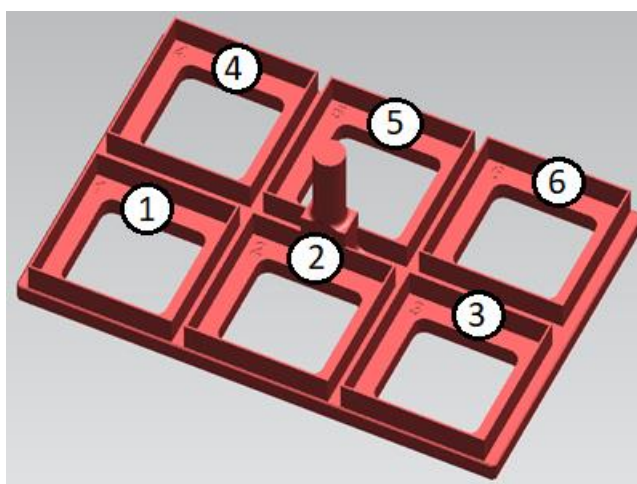


Figura 20: Posición de los ítems en el pallet

Por ejemplo, si se solicita que la estación debe de completar el servicio siete, el pallet llegará a la estación con el pallet con las bases y los rodamientos ya montados y las tareas a realizar por el robot será el montaje de los bulones, las tapas interiores y tapas exteriores.



## 2.1.2 Gemelo Digital

El modelado del Gemelo Digital de la célula robotizada se ha realizado con la herramienta Tecnomatix Process Simulate, debido a que, en el Análisis de Alternativas, se optó Tecnomatix PS como la mejor herramienta para la simulación de Gemelos Digitales en los que intervienen robots.

### 2.1.2.1 Entorno de Tecnomatix PS

Tecnomatix PS tiene dos modos de operación: *Standard Mode* y *Line Simulation Mode* (véase Tabla V).

Tabla V: Modos de operación de Tecnomatix PS

<i>Standard Mode</i>	
	El modo basado en tiempo permite simular un ciclo de producción único de principio a fin con una secuencia predefinida de operaciones. Los tiempos pueden ser implementados y testeados. Se puede evaluar el tiempo de ciclo de una célula.
<i>Line Simulation Mode</i>	
	Simulación en línea o modo CEE ( <i>Cyclical Event Evaluation</i> , evento cíclico evaluación) permite simular diversas dinámicas y ciclos de producción basados en eventos a diferencia de una secuencia predefinida.

Ambos modos de operación son necesarios para el desarrollo del Gemelo Digital de este proyecto. Por un lado, el modo *Line Simulation Mode* se utiliza para validar el Gemelo mediante simulación ya que, las estaciones no siguen una secuencia predefinida de operaciones, sino que según las entradas/salidas del robot se realizan unas determinadas operaciones; es decir, el ciclo de producción está basado en eventos. Por otro lado, el *Standard Mode* se utiliza para modelar la estación debido a que es necesario para la inserción de los componentes que conforman el modelo.

### 2.1.2.2 Modelado de la estación

Como se comenta en apartados previos, se ha partido de un Gemelo Digital de una de las estaciones, el cual fue modelado en un anterior Trabajo de Fin de Máster donde el escenario ya estaba modelado. En este trabajo, no obstante, se implementan todos los servicios mencionados anteriormente.

A continuación, se realiza una descripción de cómo se ha modelado dicha estación.

#### 2.1.2.2.1 Inserción de componentes

En primer lugar, se ha de crear una librería donde estarán los modelos 3D de los elementos que formarán el Gemelo Digital. La librería ha de ubicarse en el directorio C:\SYSROOT. Dentro de esta librería se distinguen dos tipos elementos (véase Tabla VI):

- “Parts” (Partes): componentes que se manipulan en la célula robotizada.
- “Resources” (Recursos): resto de componentes que conforman el escenario.

Tabla VI: Tipos de elementos que se utilizarán en el proyecto y su descripción

Tipo de elemento	Subtipo	Descripción
Resource	Device	Mecanismo que no sea un robot y tiene una cinemática definida
	EquipmentPrototype	Sirve para agrupar otros recursos
	Gripper	La(s) pinza(s) del robot
	Robot	El propio robot
	ToolPrototype	Elemento general. Se usa para definir los elementos estáticos del Gemelo.
Parts	Parts	Componentes manipulables
	Compound Parts	Parts compuestas

Por un lado, los “Resources” que se ha utilizado en la elaboración de este Gemelo son el robot KUKA KR 3 R540 definido como tipo “Robot”, las pinzas de la herramienta del robot se definen como “Gripper” y el resto de elementos como los alimentadores y la mesa serán “ToolPrototype”.

Por otro lado, las “Parts” del Gemelo serán los elementos que se utilizan en los montajes: los pallets, rodamientos, bulones, tapas interiores y tapa exteriores.

Se han creado seis rodamientos, bulones, tapas interiores y tapa exteriores debido a que el número máximo de ítems a montar por cada pallet es de seis.

Para catalogar las diferentes situaciones de entrada de los pallets (véase Tabla IV), se han creado lo que se identifica como “Casos”. Existen cuatro “Casos” diferentes según los elementos que lleguen ya montados a las estaciones (véase Tabla VII).

Tabla VII: Casos de los Pallets de Entrada

Nº de Caso	Situación Pallets de Entrada	Servicios
1	Bases vacías	1
		2
		3
		4
2	Bases con rodamientos	5
		6
		7
3	Bases con rodamientos y bulones	8
		9
4	Bases con rodamientos, bulones y tapas interiores	10

Estos “Casos” están definidos como “Compound Parts” denominadas como “IN\_CASOX\_Y” donde X es el número de “Caso” (véase Tabla IV) e Y es el número de ítems a montar solicitado.

#### 2.1.2.2.2 Cinemática de los elementos

Tecnomatix PS da la capacidad de dotar a los componentes móviles de cinemática. Estos componentes son el propio robot y su herramienta.

La cinemática del robot viene dada en el propio modelo 3D del robot obtenido por el soporte técnico del fabricante KUKA (véase Figura 21). Se puede observar que cada elemento que conforma el robot está de un color diferente (denominado ki) y entre un elemento y otro están los enlaces (denominado ji) que representan las articulaciones de revolución del robot.

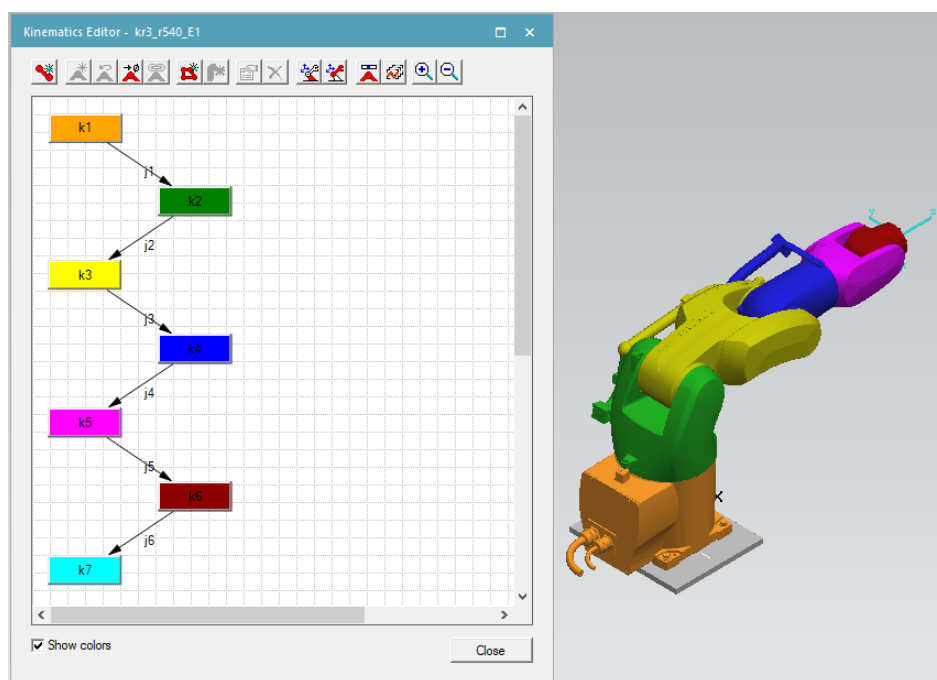


Figura 21: Configuración de la cinemática del robot

Por otro lado, ha sido necesario configurar la cinemática de la herramienta del robot. Para ello, seleccionando la herramienta y en el editor de cinemática (*Kinematics Editor*) se han definido sus diferentes elementos (véase Figura 22). En primer lugar, se define el elemento “FIJO” compuesto por todas las piezas de la herramienta que son inmóviles. Por otra parte, se definen los elementos móviles, que son los dedos de las dos pinzas neumáticas.

Una vez definidos los cinco elementos, se crean los enlaces del elemento fijo con los móviles. Estos enlaces pueden ser prismáticos o de revolución. En este caso, todos los enlaces son prismáticos.

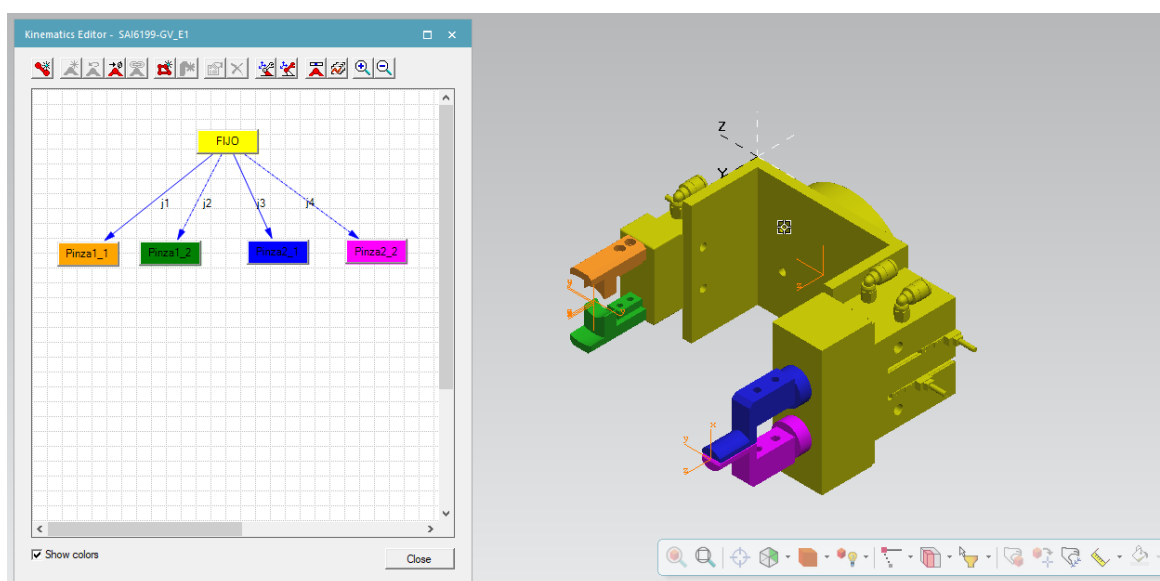


Figura 22: Configuración de la cinemática de la herramienta del robot

Para crear el movimiento de los dedos se define su eje de translación, así como los límites superior e inferior de acuerdo con los finales de carrera de las pinzas reales. Para que cada dedo de las pinzas vaya en un sentido se establece la dependencia entre los enlaces (véase Figura 23).

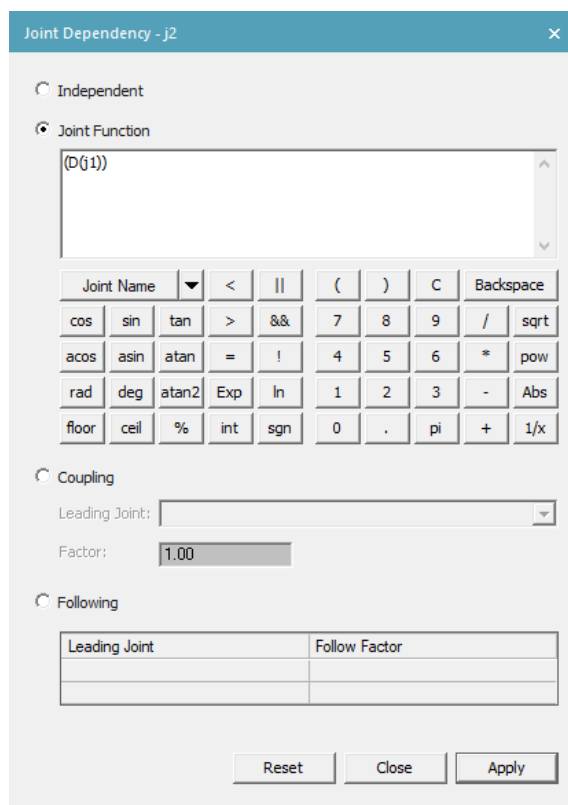


Figura 23: Dependencia del enlace j2 respecto del j1

Para que las pinzas puedan abrirse y cerrarse es necesario crear unas señales mediante el “Editor de Poses” (*Pose Editor*) y así se definen las diferentes poses. Estas poses representan las posiciones que pueden tener cada una de las pinzas. La pose “Rodamiento\_Holding” determina que la pinza está agarrando un rodamiento, tapa interior o exterior, la pose “Bulon\_Closed” determina que la pinza está agarrando un bulón o un pallet. La pose “HOME” determina que ambas pinzas están abiertas.

Para finalizar el modelado de la pinza, se crean los puntos centrales de la herramienta (TCP, *Tool Center Point*). Se crean tres uno en la base de la herramienta y los otros dos en las puntas de las pinzas (véase Figura 24). Los TCP se utilizan para el posicionamiento del robot en coordenadas cartesianas (coordenadas de traslación y rotación).

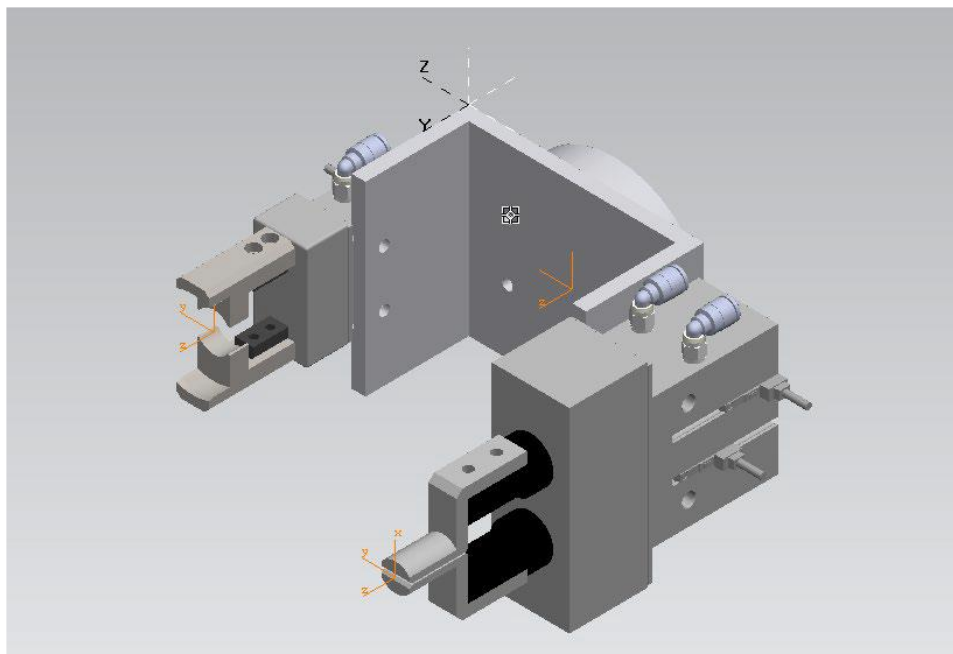


Figura 24: Puntos TCP de la herramienta del robot (naranja)

#### 2.1.2.2.3 Bloque lógico para la herramienta de robot

Una vez realizada la configuración cinemática de la herramienta, para poder controlar los movimientos de apertura/cierre de ésta, es necesario crear lo que se denomina un Bloque Lógico de Pose y Sensores (*LB Pose and Sensors*). Este comando permite seleccionar poses creadas en el “Editor de Poses” para las cuales el sistema crea acciones de pose y sensores, así como las entradas y salidas requeridas.

Las entradas del bloque lógico están conectadas a las órdenes de apertura/cierre de las pinzas y las salidas de este bloque a los sensores generados automáticamente (véase Figura 25). Por ejemplo, la señal “rmtp\_Bulon\_CLOSED” es la orden de cierre de la pinza de bulones y la señal del sensor que indica que la pinza está cerrada es “at\_Bulon\_CLOSED”.

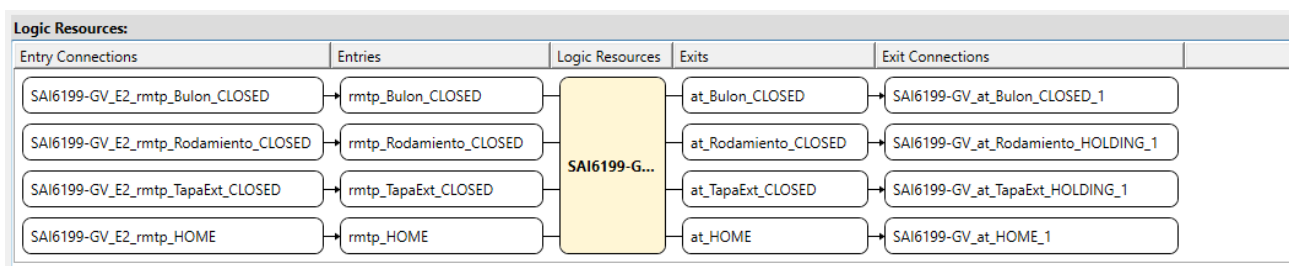


Figura 25: Bloque lógico de la herramienta del robot

### 2.1.2.3 Operaciones

Una vez colocado los componentes de la célula en el Gemelo Digital y una vez dotado a los componentes móviles de su cinemática, se pasa a realizar las operaciones de simulación.

La herramienta Tecnomatix PS permite definir varios tipos de operaciones (véase Figura 26). Seguidamente, se explican los tipos de operaciones que se han utilizado en el proyecto para entender mejor su uso.

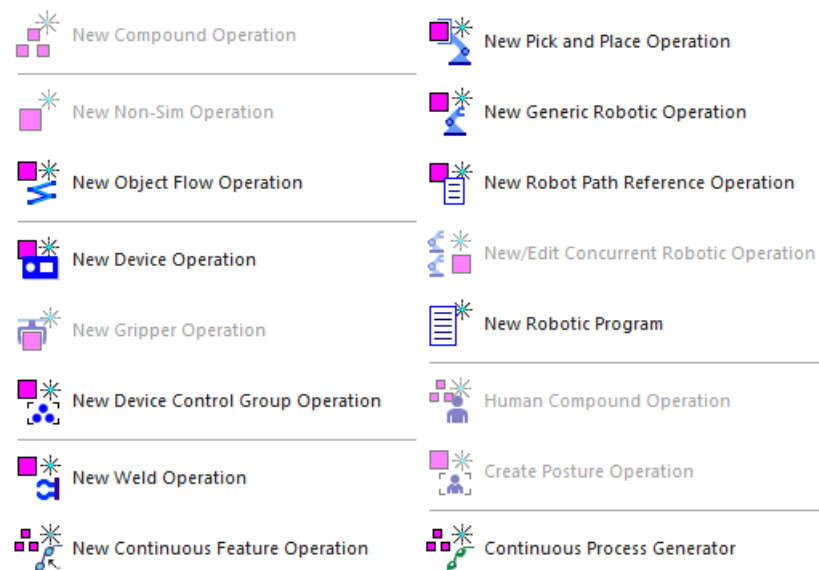


Figura 26: Tipos de operaciones

- *New Compound Operation* (Operación Compuesta): su función es agrupar las operaciones en grupos.
- *New Non-Sim Operation* (Operación No Simulada): permite crear operaciones no simuladas. Este tipo de operación es una operación vacía que se puede utilizar para marcar un intervalo de tiempo especificado antes de continuar con otra operación o para marcar el lugar de una operación que se creará más adelante. Se usa para la creación de los pallets de entrada.
- *New Object Flow Operation* (Operación de Flujo de Objetos): permite crear movimiento de un objeto de un lugar a otro.
- *New Device Operation* (Operación de Dispositivo): permite mover un dispositivo de una pose a otra. Una vez que se ha definido un dispositivo, se pueden definir sus poses mediante el Editor de Poses (*Pose Editor*).
- *New Pick and Place Operation* (Operación de Coger y Colocar): permite mover un objeto de un lugar a otro mediante el robot.

- *New Generic Robotic Operation* (Operación General del Robot): permite generar una operación de robot de uso genérico.
- *New Robotic Program* (Programa Robot): permite crear programas para el robot. Por ejemplo: el programa principal del robot.

Una vez definidas las operaciones que debe de realizar el Gemelo, es necesario que éstas tengan señales asociadas para que se puedan activar. Debido a esto, se deben crear lo que se denomina *Operation Signals* (Señales de Operación). Existen diferentes tipos según la operación (véase Figura 27).

- *Device Start Signals*: crea la señal que activa la “Operación de Dispositivo”
- *Non-Sim Start Signals*: crea la señal que activa la “Operación No Simulada”
- *Robot Operation Start Signals*: crea señales de inicio para todas las operaciones robóticas asociadas con los robots
- *All Flow Start Signals*: crea señales de inicio para todas las “Operaciones de Flujo de Objetos”



Figura 27: Tipos de señales de activación de las operaciones

En los siguientes apartados se describen las operaciones que se han empleado para poder realizar la simulación del Gemelo Digital. Estas operaciones se han dividido en dos tipos: las “operaciones de materiales” donde se encuentran las operaciones relacionadas con el utillaje de las estaciones, y las “operaciones de robot”, las cuales están relacionadas con el robot.

#### 2.1.2.3.1 Operaciones de materiales

Los pallets de entrada y las tapas exteriores se crean mediante operaciones del tipo *New Non-Sim Operation*. Dentro de estas operaciones se configura qué piezas se deben generar (véase Figura 28).

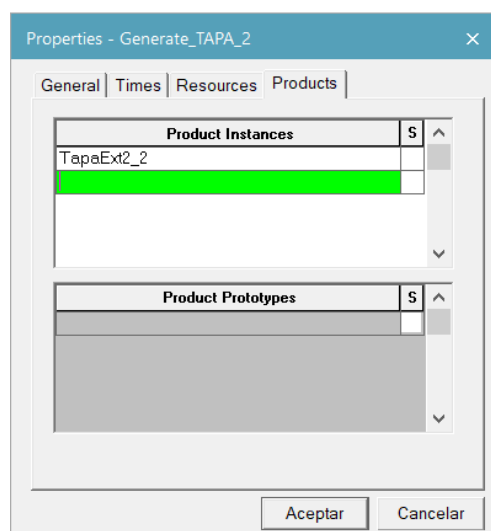


Figura 28: *Non-Sim operation* para crear una de las tapas exteriores

Dependiendo del tipo de servicio que se solicite, se genera la apariencia de un “Caso” o situaciones de los pallets de entrada (véase Tabla VII). El nombre estas operaciones es “GeneratePartsX\_Y” donde X pertenece al número de “Caso” (véase Tabla VII) e Y pertenece al número de ítems solicitado en el servicio.

Asimismo, si se solicita un servicio donde se montan las tapas exteriores (véase Tabla IV), éstas aparecen en la estación en la zona de entrada de tapas exteriores (véase Figura 8). El nombre de estas operaciones es “Generate\_TAPA\_Z” donde Z es la posición de las tapas exteriores en el pallet (véase Figura 20).

Por otro lado, también se utiliza este tipo de operación *Non-Sim* para realizar las operaciones denominadas “KILL”. Esta operación sirve para eliminar cualquier apariencia que esté en el Gemelo Digital durante la simulación, de esta forma cuando se solicite un nuevo servicio no habrá ningún elemento dentro la célula.

Para simular el desplazamiento de los rodamientos, bulones y tapas interiores por los alimentadores, se utilizan las operaciones tipo *Object Flow Operation*. Mediante estas operaciones se crean las apariencias de los elementos y, además, se recrea el movimiento que hacen los elementos en sus alimentadores (véase Figura 29).

Por cada alimentador se han realizado seis operaciones diferentes ya que existen seis bulones, seis rodamientos y seis tapas interiores, uno por cada posición del pallet (véase Figura 20).

Las operaciones de los rodamientos se denominan “RodB\_MF\_PosZ”, las de los bulones “BulonLargoP\_MF\_PosZ”, “TapaInt2\_1\_PosZ”. Donde Z representa la posición de los anteriores elementos sobre el pallet.

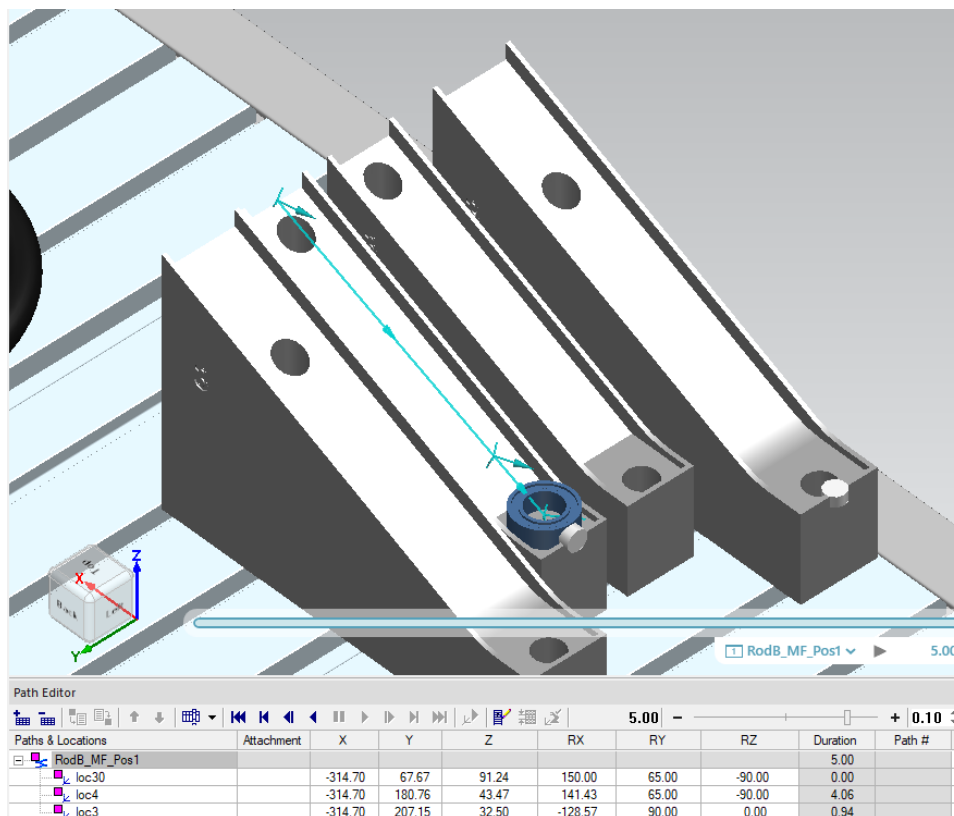


Figura 29: Operación de flujo de materiales del rodamiento de la posición 1

Para activar las señales para ejecutar estas operaciones, se ha creado un sensor fotoeléctrico por cada alimentador. Cuando no hay pieza en el alimentador, la señal del sensor se desactiva y, por consiguiente, se activa la señal de la operación y dependiendo el valor del contador de ítems, se crea un nuevo elemento.

Entre las características que hay que configurar en el sensor fotoeléctrico están el área de detección y la lista de elementos con los que se comprueba la interferencia (véase Figura 30).

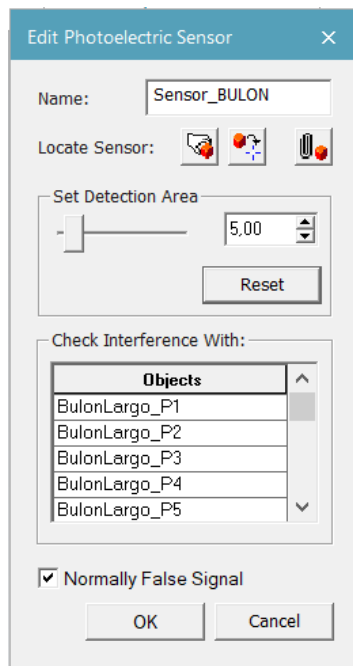


Figura 30: Características del sensor fotoeléctrico del alimentador de bulones

Tecnomatix PS carece de dinámica; es decir, que no existe gravedad. Cuando el robot coloca las tapas interiores y exteriores, éstas se quedan en el aire. Para que esto no ocurra, se han creado unas *Object Flow Operations* para simular la caída de estas tapas debido a la gravedad (véase Figura 31). El nombre de las operaciones para las tapas interiores es “TapaInt\_Abajo\_PosZ” y el de las tapas exteriores es “TapaExt\_PosZ”.

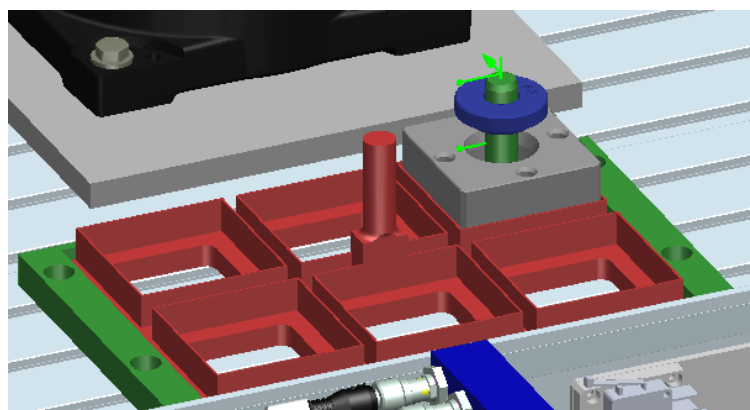


Figura 31: Operación flujo de materiales TapaInt\_Abajo\_Pos1

#### 2.1.2.3.2 Operaciones de Robot. Trayectorias

Todas las operaciones que realiza el robot son de Coger y Colocar comúnmente denominadas en inglés *Pick & Place* (véase Tabla VIII).

Tabla VIII: Operaciones de robot

Operación	Descripción
OP100	Traslada el pallet de entrada desde la zona de entrada de la estación a la zona de montaje
OP200	Traslada el conjunto montado desde la zona de montaje a la zona de salida
PICK_RODAMIENTO	Coge los rodamientos de su alimentador
PICK_BULON	Coge los bulones de su alimentador
PICK_TAPA_INT	Coge las tapas interiores de su alimentador
PICK_TAPA_EXT_POSZ	Coge la tapa exterior correspondiente de la zona de entrada de las tapas exteriores
PLACE_RODA_POSZ	Coloca los rodamientos en la posición del pallet correspondiente
PLACE_BULON_POSZ	Coloca los bulones en la posición del pallet correspondiente
PLACE_TAPA_INT_POSZ	Coloca las tapas interiores en la posición del pallet correspondiente
PLACE_TAPA_EXT_POSZ	Coloca las tapas exteriores en la posición del pallet correspondiente
OP1000	Traslada el pallet de entrada desde la plataforma del AMR a la zona de entrada de pallets de la estación
OP2000	Traslada el montaje desde la zona de salida de la estación a la plataforma del AMR

Para generar las anteriores operaciones, se han utilizado operaciones tipo *New Generic Robotic Operation* y se han utilizado los comandos de *Add Location* (Añadir Localización) para añadir las localizaciones para crear las trayectorias que realizará el robot (véase Figura 32).

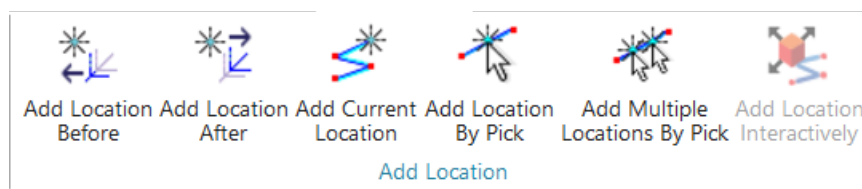


Figura 32: Comandos Add Location

Para realizar una operación es necesario que ésta se añada al *Path Editor* (Editor de Ruta) para ir añadiendo localizaciones a la operación para crear trayectorias. Además, permite simular cada operación comprobando si la trayectoria se realiza de manera correcta (véase Figura 33).

Las operaciones genéricas aparecen vacías y hay que ir añadiendo localizaciones. Para ello, mediante el comando *Add Location By Pick* (Añadir Localización Para *Pick*), se añade a la operación la localización en la cual el robot coge los diferentes elementos. Para completar las trayectorias se han

de añadir localizaciones intermedias y de aproximación con los comandos *Add Location Before* (Añadir Localización Antes) y *Add Location After* (Añadir Localización Después).

Con el comando *Add Current Location* (Añadir Localización Actual) se han creado las localizaciones “HOME” y “HOME\_MONTAJE”. La localización “HOME” es la posición de reposo del robot y la localización “HOME\_MONTAJE” es la posición de reposo del robot cuando está en la zona de montaje.

A parte de las localizaciones, también hay que tener en cuenta otras características para que el robot haga una trayectoria correcta. El tipo de movimiento define cómo se desplaza el robot de una localización a otra. En este proyecto se han utilizado dos tipos: “LIN” para las trayectorias lineales; es decir, para las localizaciones *Pick*, *Place* y las de aproximación y “PTP” para las trayectorias Punto a Punto; es decir, para las localizaciones intermedias.

Paths & Locations	X	Y	Z	RX	RY	RZ	Duration	OLP Commands	Robot	Configur...	Config	Speed	Motion
PICK_BULON							20.21		kr3_r540...				
via3	373.00	-0.00	625.00	180.00	-90.00	0.00	0.00			✓	S 2 T 35	100 %	PTP
HOME_MONTAJE	-66.80	294.80	294.80	180.00	0.00	90.00	2.00			✓	S 2 T 43	100 %	PTP
via161	284.97	280.16	489.25	90.00	0.00	90.00	2.61			✓	S 6 T 27	100 %	PTP
via160	284.97	280.15	194.25	90.00	0.00	90.00	3.95			✓	S 6 T 27	0.1 m/s	LIN
via153	334.97	280.15	194.25	90.00	0.00	90.00	1.41			✓	S 6 T 27	0.1 m/s	LIN
pick_bulon	364.97	280.15	194.25	90.00	0.00	90.00	1.16	\$OUT[1]=TRUE #Destinatio		✓	S 6 T 27	0.1 m/s	LIN
via153	334.97	280.15	194.25	90.00	0.00	90.00	1.10			✓	S 6 T 27	0.1 m/s	LIN
via160	284.97	280.15	194.25	90.00	0.00	90.00	1.41			✓	S 6 T 27	0.1 m/s	LIN
via161	284.97	280.16	489.25	90.00	0.00	90.00	3.95			✓	S 6 T 27	0.1 m/s	LIN
HOME_MONTAJE	-66.80	294.80	294.80	180.00	0.00	90.00	2.61			✓	S 2 T 43	100 %	PTP

Figura 33: Operación PICK\_BULON en el Path Editor

Otra característica a tener en cuenta es la configuración de ejes del robot (véase Figura 34). Es importante realizar una buena configuración de ejes porque si la configuración es incorrecta, el robot podría realizar movimientos que podrían dar lugar a sobreesfuerzos. Para evitarlo, en la ventana de configuración se “enseña” (*Teach*) al robot su configuración idónea.

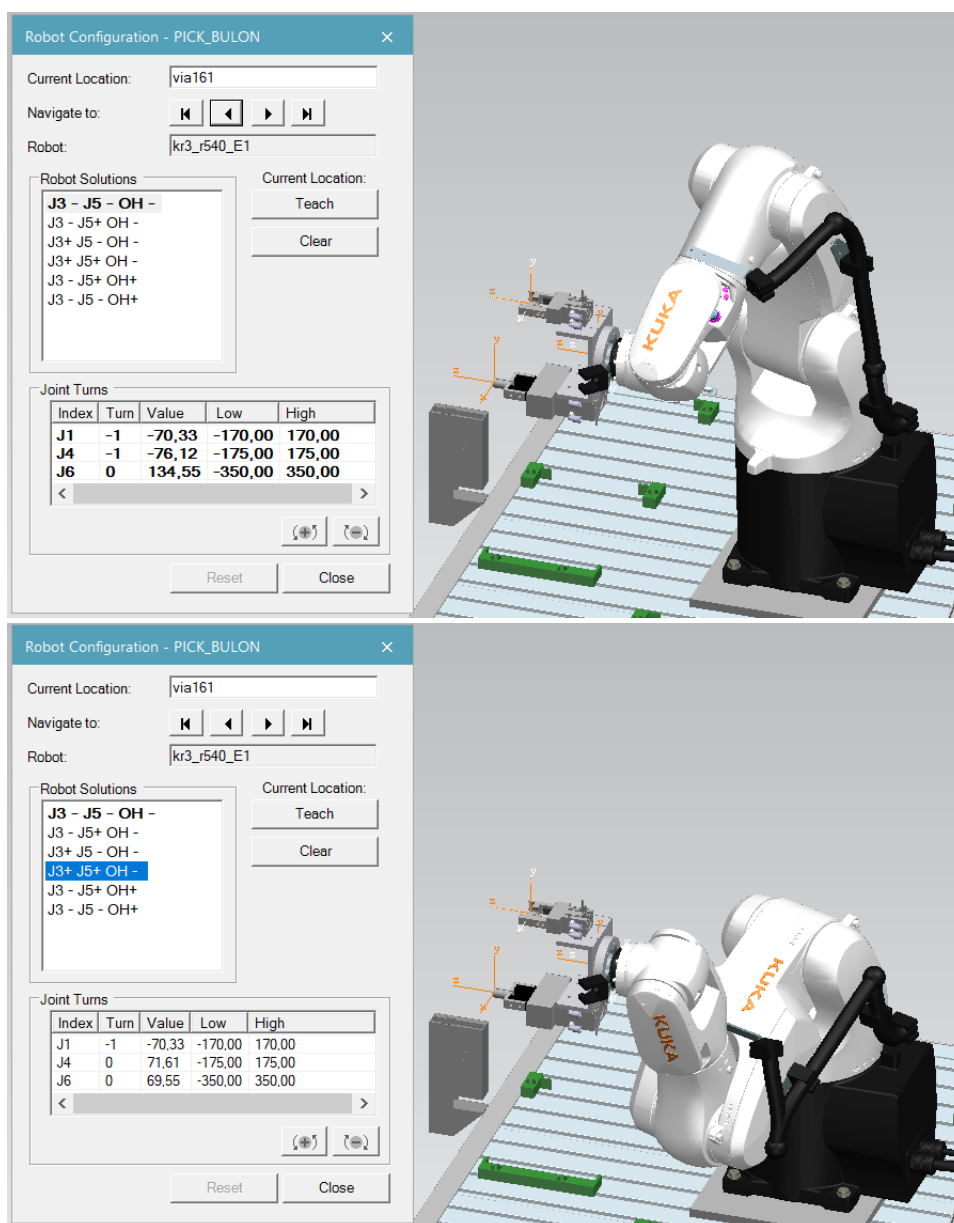


Figura 34: Configuración correcta (arriba) e incorrecta (abajo) de los ejes de un robot

Para completar las operaciones del robot, es necesario que las pinzas se abran y se cierren. Por ello, en las localizaciones donde se realizan los *Pick* y *Place*, se tiene que añadir en los “comandos OLP” (*Off Line Program commands*) la apertura y cierre de las pinzas (véase Figura 35). Para ello, se elige la pinza y la posición de ésta; es decir, abierta o cerrada. Finalmente, se añade una espera hasta que se detecte que la pinza está en la posición deseada.

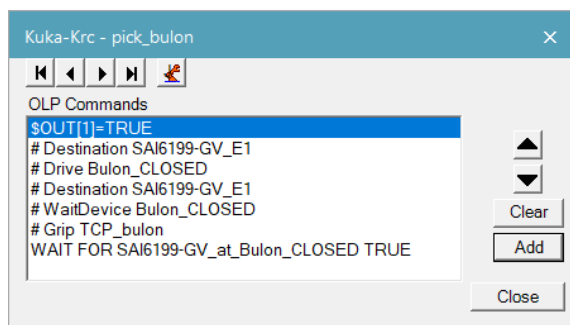


Figura 35: Comandos OLP cierre pinza

Cabe destacar que en los “comandos OLP” de la operación OP200 (véase Figura 36) se utiliza el comando denominado *Attach* (Fijación) mediante el cual se realiza el ensamblaje de los montajes. Por lo que se utiliza un *Switch-Case* con todas las posibilidades posibles dependiendo el servicio y el número de ítems que se haya solicitado.

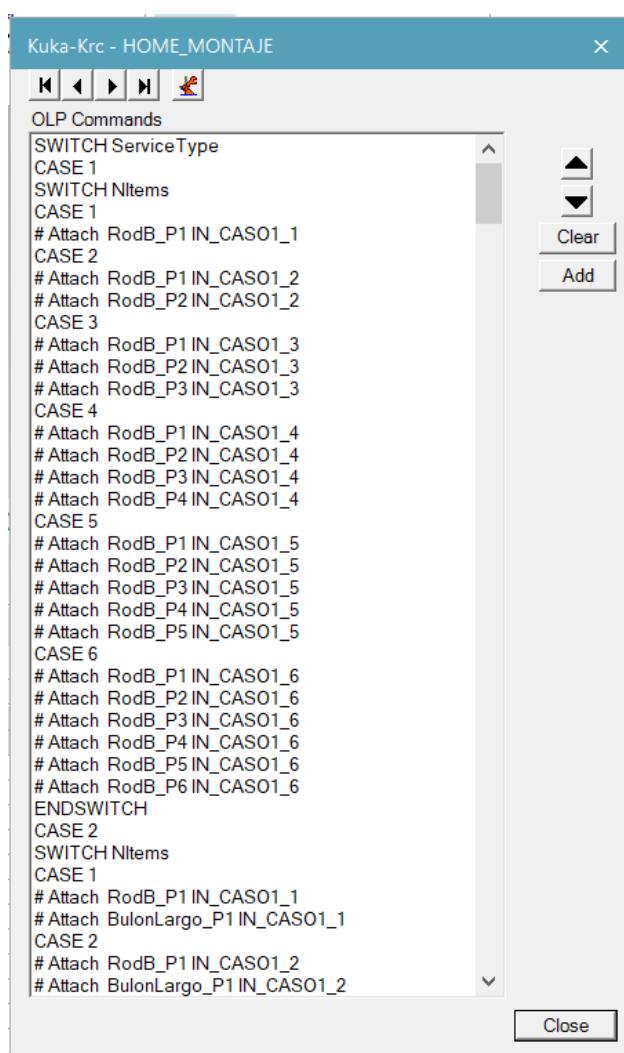


Figura 36: Fragmento de los comandos OLP de la operación OP200

#### 2.1.2.4 Programación del robot

Después de haber realizado las operaciones de las trayectorias que realiza el robot, se realiza la programación *offline* del robot.

Para poder realizar la programación, es necesario incluir la unidad de control virtual del robot, ya que el robot tiene unas señales de control internas y es necesario traducir los programas realizados en Tecnomatix al lenguaje propio del robot KUKA. La controladora virtual añadida es del tipo KUKA-Krc, la cual pertenece al mismo tipo que la Unidad de Control real: KR C4 Compact.

Seguidamente, se crean las señales de control del robot. Tecnomatix PS crea por defecto unas señales de control. Sin embargo, el usuario puede añadir las señales de robot que le sean necesarias. Por ello se han creado las siguientes señales (véase Tabla IX). Hay ciertas variables que sólo se han creado porque son necesarias para el flujo de materiales, sin embargo, se ha intentado que el número de estas sea el menor posible.

Tabla IX: Señales de robot

Nombre de la Señal	I/O	Función	Tipo
\$EXT_START	Q	Arranque externo	BOOL
\$MOVE_ENABLE	Q	Marcha habilitada	BOOL
\$DRIVES_OFF	Q	Accionamientos conectados	BOOL
\$DRIVES_ON	Q	Accionamientos desconectados	BOOL
\$IN_HOME1	Q	El robot se encuentra en la posición inicial	BOOL
kr3_r540_programNumber	Q	Número de programa de robot a ejecutar	BYTE
kr3_r540_NewService	Q	Orden de que ha llegado una nueva solicitud de servicio	BOOL
kr3_r540_ServiceType	Q	Tipo de servicio solicitado para que ejecute el robot	BYTE
kr3_r540_NItems	Q	Número de ítems solicitado	BYTE
ItemStarted	I	Ítem iniciado	BOOL
ItemCompleted	I	Ítem completado	BOOL
ServiceStarted	I	Servicio iniciado	BOOL
Service_Completed	I	Servicio completado	BOOL
kr3_r540_Counter	I	Contador de ítems montados	BYTE
kr3_r540_TapaInt	I	El robot está montando las tapas interiores (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_TapaExt	I	El robot está montando las tapas exteriores (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_at_Tapa_P1	I	El robot está en la posición 1 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_at_Tapa_P2	I	El robot está en la posición 2 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL

kr3_r540_at_Tapa_P3	I	El robot está en la posición 3 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_at_Tapa_P4	I	El robot está en la posición 4 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_at_Tapa_P5	I	El robot está en la posición 5 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL
kr3_r540_at_Tapa_P6	I	El robot está en la posición 6 del pallet para dejar la tapa (necesario para la simulación del flujo de materiales)	BOOL

Una vez creadas las señales para el control, se realiza el programa robot. El programa principal del robot se llama “MAIN” (véase Figura 37). Este programa está definido como el programa por defecto del robot. Por ello, la señal de “kr3\_r540\_programNumber” se ha de poner a 1 para que la ejecución del “MAIN” pueda realizarse.

Primero, el programa espera a que se habilite la marcha del robot y se queda esperando a una solicitud nueva de servicio. A continuación, el programa entra en un bucle donde que el arranque externo no esté deshabilitado, en el caso de que lo esté, el programa saldrá del bucle. En este mismo bucle se ejecuta la operación OP1000 para coger el pallet del AMR y dependiendo del servicio llama a los diferentes subprogramas, que son los encargados de controlar los servicios. Cuando el servicio se termine, se ejecuta la OP2000 para pasar el montaje realizado en la estación al AMR y se queda a la espera de un nuevo servicio.

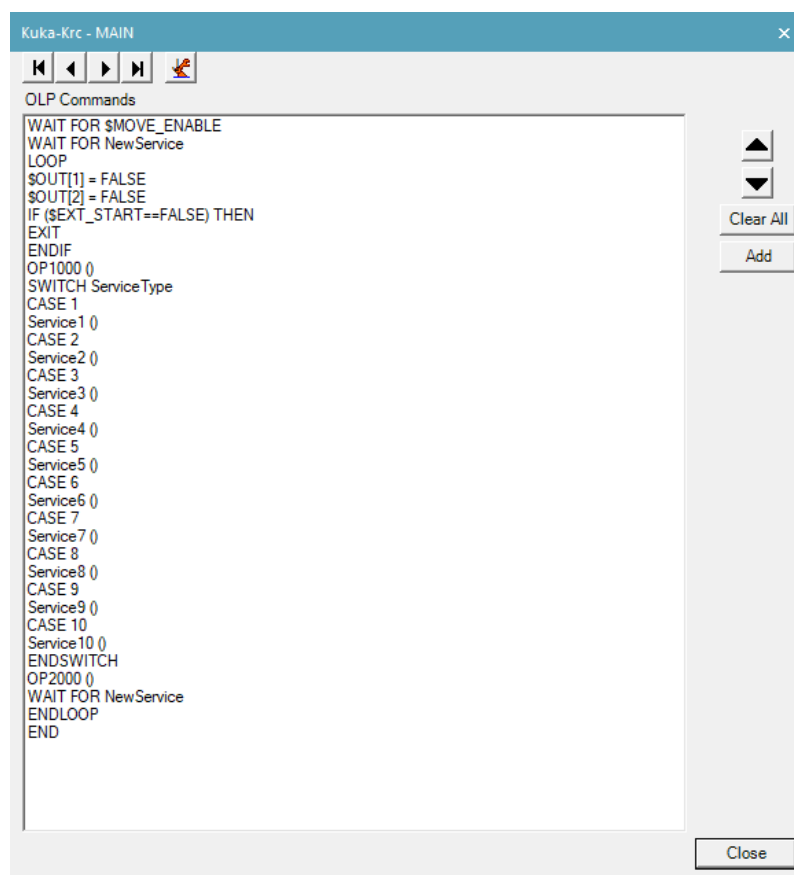


Figura 37: Comandos OLP del programa principal MAIN

Una vez evaluado en el programa MAIN cuál es el servicio solicitado, se entra en el subprograma donde se ejecutan las operaciones *Pick & Place* correspondientes a ese servicio (véase Tabla X).

Tabla X: Resumen de operaciones Pick & Place de cada Servicio

		PICK & PLACE Rodamiento	PICK & PLACE Bulón	PICK & PLACE Tapa Int	PICK & PLACE Tapa Ext
MAIN	Service 1				
	Service 2				
	Service 3				
	Service 4				
	Service 5				
	Service 6				
	Service 7				
	Service 8				
	Service 9				
	Service 10				

Como ejemplo de subprograma se ha cogido el Servicio 2 (véase Figura 38). Al principio de todos los subprogramas se evalúa mediante un *Switch-Case* cuántos ítems hay que montar.

En primer lugar, se lleva el pallet de entrada a la zona de montaje mediante la operación OP100 y a continuación, se inicializa el montaje del primer ítem y esto se refleja mediante un pulso de 3 segundos en la señal “Item\_Started”. Se llama a las operaciones correspondientes de montaje y mediante el contador “kr3\_r540\_Counter” se controla el flujo de materiales de los rodamientos, bulones y tapas interiores. Cuando se ha completado un ítem se activa durante 3 segundos la señal “Item\_Completed” y actualiza el valor del “kr3\_r540\_Counter”. Este procedimiento se repite hasta que se terminan de montar todos los ítems.



Figura 38: Fragmento de los comandos OLP del subprograma Service 2

### 2.1.2.5 Modelado del AMR

Para modelar el AMR se optó por una de las soluciones de movilidad que KUKA ofrece: KMP 1500 omniMove. Una vez descargado el archivo 3D del servicio técnico de KUKA, éste se inserta dentro de Tecnomatix PS.

#### 2.1.2.5.1 Creación AMR

Tecnomatix PS permite que cualquier tipo de recurso sea un AMR, en este caso se ha definido como *Device*, simplemente hay que seleccionarlo y en la pestaña de “Control” darle a crear un AMR (*Create AMR*). Una vez definido el archivo 3D del KMP 1500 omniMove como AMR, se crea automáticamente la lógica AMR necesaria donde aparecen las señales asociadas al AMR (véase Tabla XI).

Tabla XI: Variables AMR

Entrada	Tipo	Uso
MotionPlannerType	INT	Define si el AMR usa la alfombra para ir a su destino o la ignora. 0 – utiliza la alfombra 1 – ignora la alfombra
PathPlannerType	INT	Define si el AMR se dirige inmediatamente a nuevos objetivos a medida que se asignan o los almacena en búfer. 0 – búfer 1 – anular el objetivo con el nuevo
RotationMethod	INT	Establece el método en el que el AMR se acerca a un objetivo: 0 – avance 1 – hacia atrás 2 – mantener orientación
Speed	REAL	Establece la velocidad del AMR en mm/s
TargetIndex	INT	Establece el siguiente objetivo del AMR
Salida	Tipo	Uso
Busy	BOOL	Se obtiene el estado del AMR. 0 – libre 1 – ocupado
CurrentTarget	INT	Se obtiene el objetivo actual del AMR
OnTarget	BOOL	Se obtiene la posición del AMR. 0 – no está en objetivo 1 – está en objetivo
TargetReached	BOOL	Se obtiene un pulso cuando se alcanza el objetivo

#### 2.1.2.5.2 Situar los objetivos/paradas del AMR

Después, se crean los “Objetivos” (*Target*) del AMR, éstos serán las paradas que debe de hacer el AMR por la célula. Se definen cuatro objetivos: estación 1, estación 2 y el almacén de entrada y salida. Cada uno de estos “Objetivos” tiene un número para su identificación denominado “AMRTarget\_ID”. Esos números son necesarios para poder establecer el siguiente objetivo del AMR mediante el “TargetIndex” (Índice del Objetivo). Por lo tanto, el “AMRTarget\_ID” de la estación 1 es 1, el de la estación 2 es 2, el del almacén de entrada es 3 y el del almacén de salida es 4.

Seguidamente, se crea la ruta del AMR. Esta alfombra es un componente 2D formado por líneas y curvas que representa el camino por el cual el AMR va a cada objetivo. En este proyecto, la trayectoria que realiza el AMR es recta (véase Figura 39).

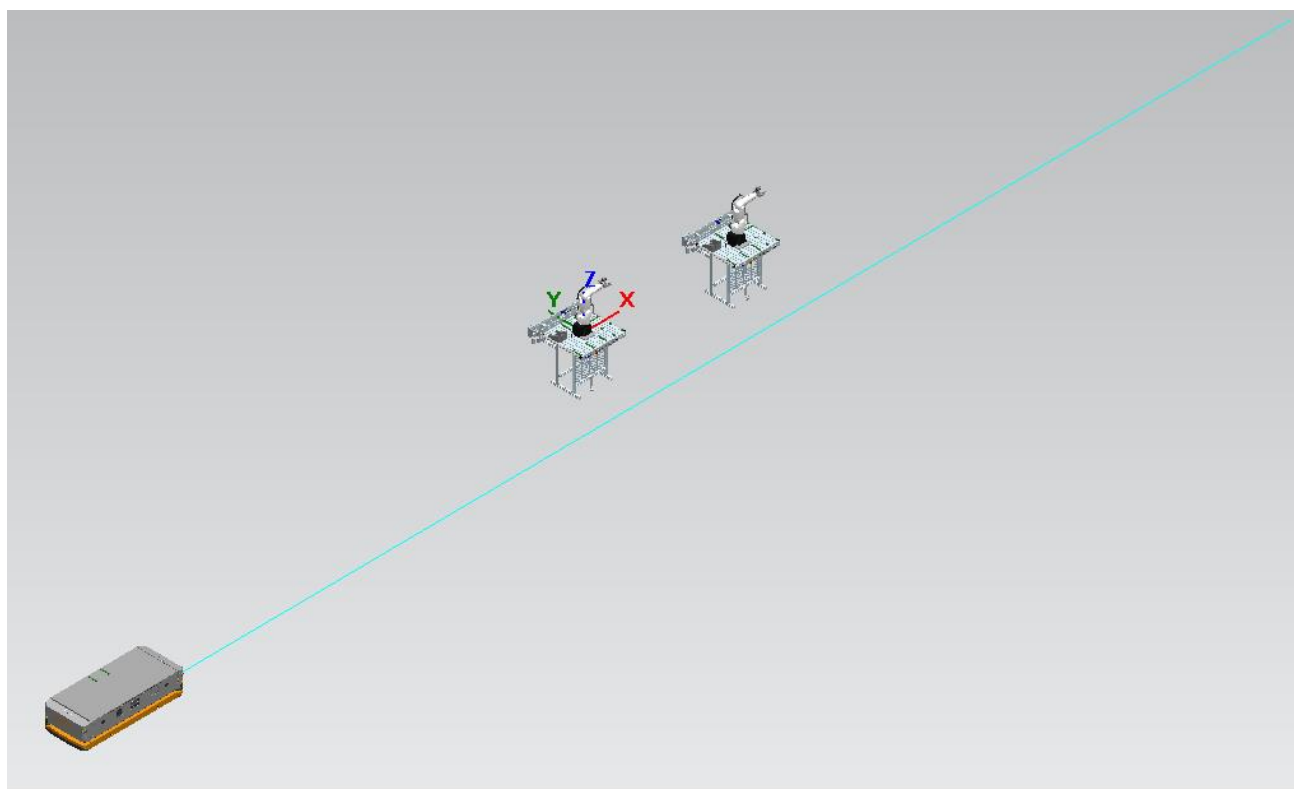


Figura 39: Ruta del AMR sobre la célula robotizada

#### 2.1.2.5.3 Plataforma del AMR

Una vez modelado el AMR, se ha añadido encima de éste una plataforma elevadora para que el robot pueda alcanzar los pallets de entrada en cada estación. Para crear dicha plataforma, es necesario establecer el AMR para modelar (*Set Modeling Scope*) y a continuación, se crean la plancha y los marcos para los pallets mediante el editor de geometría del propio Tecnomatix PS (véase Figura

40). Este editor permite crear sólidos con diferentes geometrías. Para crear la plataforma se ha creado un cubo especificando sus medidas (alto, ancho y profundidad) y se ha colocado encima del AMR. Este mismo procedimiento se ha realizado para la creación de los marcos de los pallets.

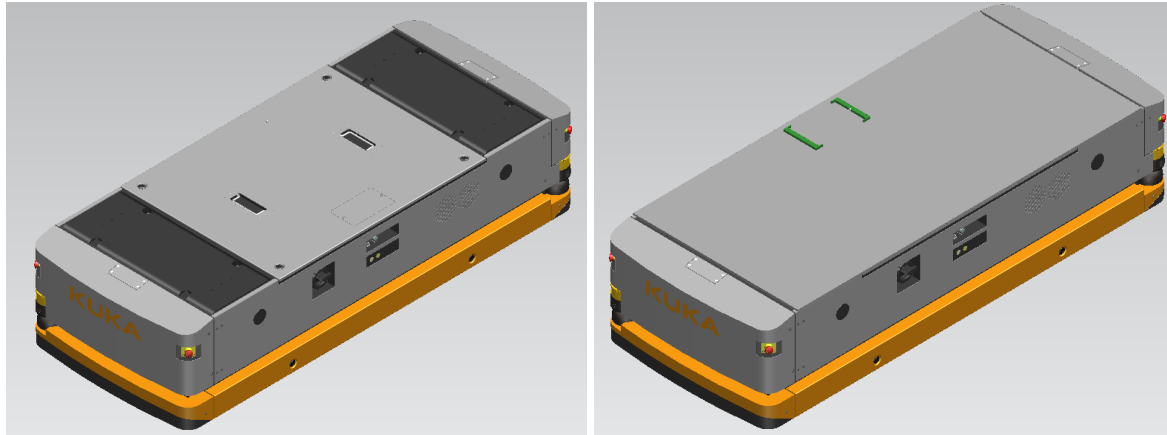


Figura 40: AMR original (izq.) y AMR con plataforma (dcha.)

Para que la plataforma se pueda elevar, es necesario crear su cinemática. Mediante el *Kinematics Editor*, se dota de cinemática a la plataforma seleccionando al AMR como la parte fija (véase Figura 42, lnk1) y a la plataforma como la parte móvil (véase Figura 41, lnk2). El tipo de movimiento que tiene la plataforma es prismático y se definen dos puntos y la línea que éstos forman es el eje de movimiento vertical de la plataforma (véase Figura 41, j1). Asimismo, se establecen los límites superior e inferior de la altura de la plataforma.

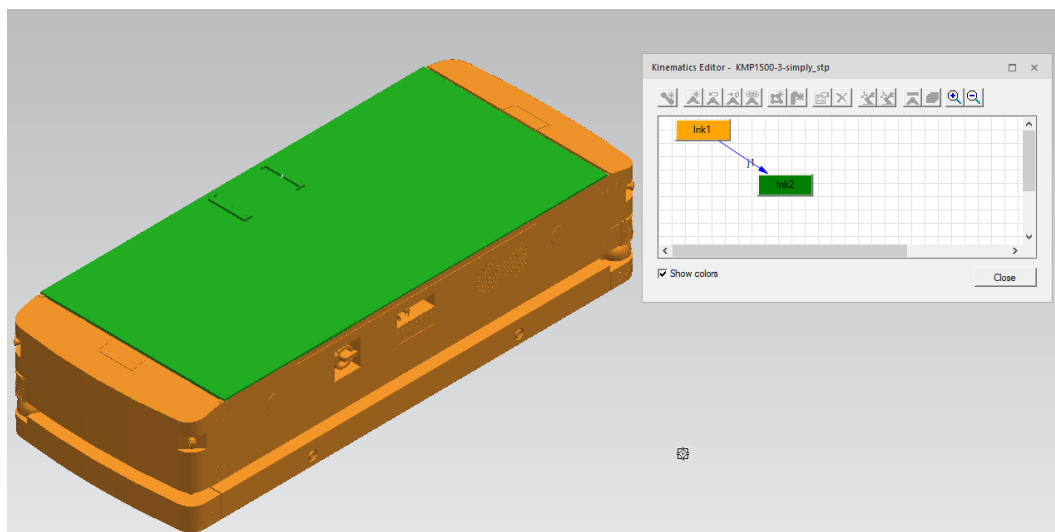


Figura 41: Configuración de la cinemática del AMR

Una vez definida la cinemática, es necesario definir las poses de la plataforma para poder elevar/bajar la plataforma. Para ello se utiliza el *Pose Editor* tal y como se ha usado con la herramienta del robot. En este caso, se han definido la pose “HOME” que indica que el AMR está abajo por lo que el valor de  $j1$  es 0 mm. Por otro lado, la pose “Arriba” indica que la plataforma está elevada donde el valor del enlace  $j1$  es de 496,04 mm (véase Figura 42), este valor es idóneo para que el robot pueda alcanzar el pallet sin problemas.

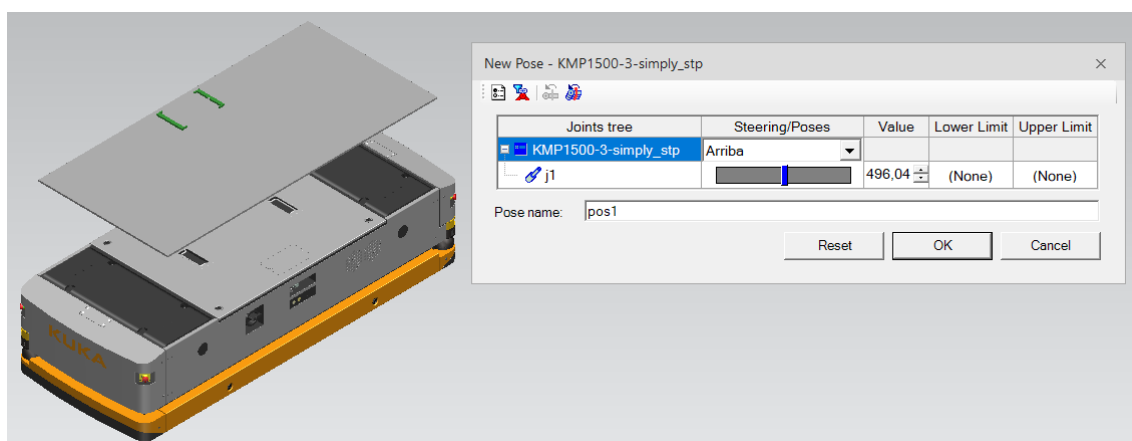


Figura 42: Configuración de la pose Arriba en el AMR

#### 2.1.2.5.4 Sensor fotoeléctrico

Para saber si hay algún pallet sobre la plataforma del AMR, se ha colocado un sensor fotoeléctrico en uno de los lados del marco de la plataforma (véase Figura 43). Con este sensor se detecta si hay un pallet sobre la plataforma del AMR. El procedimiento para configurar el sensor es definir el área de detección y la lista de elementos con los que se comprueba la interferencia. En esta lista están todos los “Casos” de los pallets de entrada.

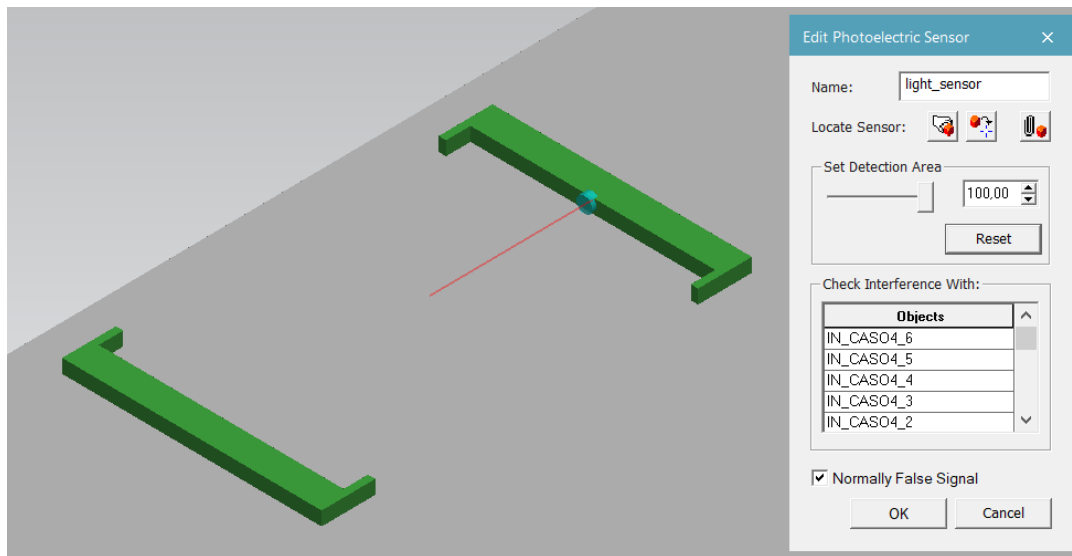


Figura 43: Sensor fotoeléctrico plataforma AMR

#### 2.1.2.5.5 Operaciones del AMR

Para poder controlar la plataforma del AMR es necesario crear un “Bloque Lógico” tal y como se ha realizado para la herramienta del robot para crear las entradas y salidas de las poses creadas en el *Pose Editor*.

Para accionar la subida y bajada de la plataforma, se han creado operaciones de dispositivo (*Device Operation*). En este tipo de operaciones se elige el dispositivo y después se define desde que pose a que pose es el movimiento. Como anteriormente se han definido las dos posiciones del AMR: “HOME” y “Arriba”; éstas se pueden utilizar para crear los movimientos para que la plataforma suba y baje (véase Figura 44).

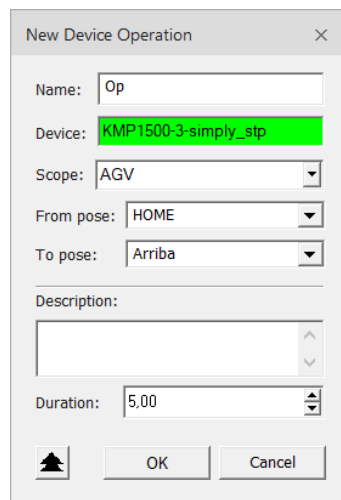
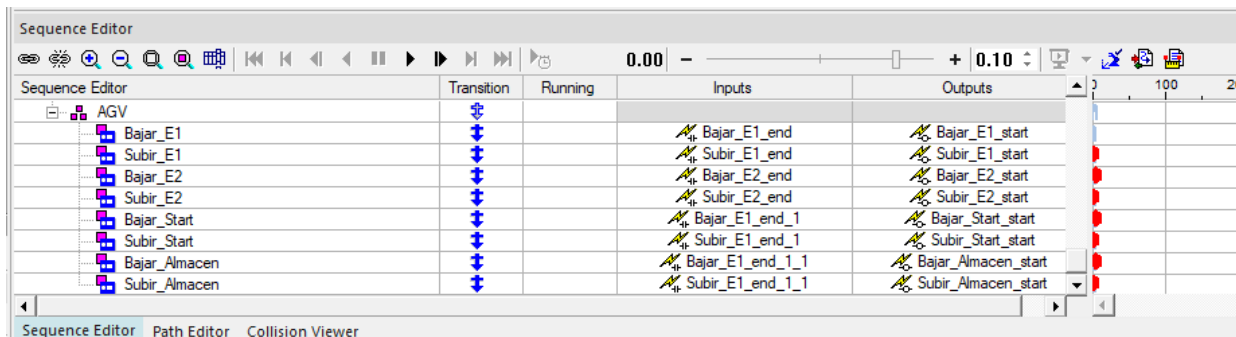


Figura 44: Configuración de la operación de dispositivo

Para que la plataforma suba y baje en todos los “Objetivos”, se han creado las operaciones de dispositivo en cada uno de ellos (véase Figura 45). Por ejemplo, “Bajar\_Start” sirve para bajar la plataforma cuando el AMR está en el almacén inicial.



Sequence Editor	Transition	Running	Inputs	Outputs
AGV				
Bajar_E1			Bajar_E1_end	Bajar_E1_start
Subir_E1			Subir_E1_end	Subir_E1_start
Bajar_E2			Bajar_E2_end	Bajar_E2_start
Subir_E2			Subir_E2_end	Subir_E2_start
Bajar_Start			Bajar_E1_end_1	Bajar_Start_start
Subir_Start			Subir_E1_end_1	Subir_Start_start
Bajar_Almacen			Bajar_E1_end_1_1	Bajar_Almacen_start
Subir_Almacen			Subir_E1_end_1_1	Subir_Almacen_start

Figura 45: Operaciones del dispositivo AMR

Como se ha dicho en apartados anteriores, el AMR transporta los pallets, esto quiere decir, que los pallets y el AMR se tienen que mover al unísono. Para que esto suceda, es necesario crear lo que se denomina como “Evento de Fijación” (*Attach Event*) en cada una de las operaciones de “Bajar” de la plataforma después de que empiece la ejecución de éstas. Sin embargo, sólo se puede crear un evento de fijación por cada “Caso” por lo que, se ha tenido que crear un “Evento de Fijación” por cada “Caso” (véase Figura 46).

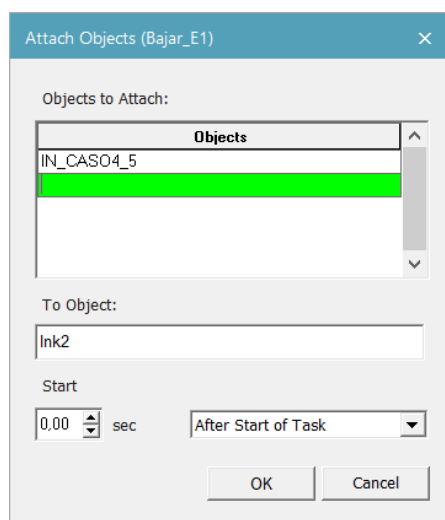


Figura 46: Evento de fijación en la operación Bajar\_E1

Por otro lado, cuando el AMR ha llegado a uno de los objetivos y la plataforma está en la posición “Arriba”, el robot tiene que poder cogerlas por lo que hay que “separar” los pallets del AMR. Para esto, se utiliza los “Eventos de Separación” (*Detach Event*) antes de que finalice la ejecución de

las operaciones de subida de la plataforma (véase Figura 47). Este tipo de evento sí deja seleccionar todos “Casos” posibles, por lo que sólo haría falta un único evento por cada operación de “Subir”.

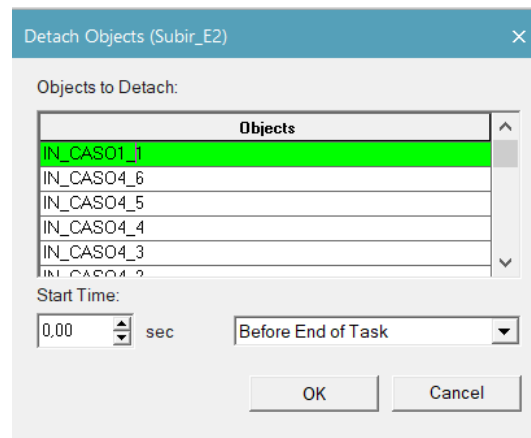


Figura 47: Evento de separación en la operación Subir\_E2

### 2.1.2.6 Duplicado de la estación

La célula está compuesta por dos estaciones robotizadas. Para hacer la segunda simplemente hay que duplicar la primera.

En el *Objetct Tree*, se selecciona la estación y simplemente con copiarlo y pegarlo se crea una réplica de la estación. Sin embargo, las estaciones se quedan superpuestas. Para mover una, se utiliza el comando *Placement Manipulator* (Manipulador de Ubicación) y se ha trasladado la nueva estación 2m en el eje X y el marco de referencia es el *Self* (véase Figura 48).

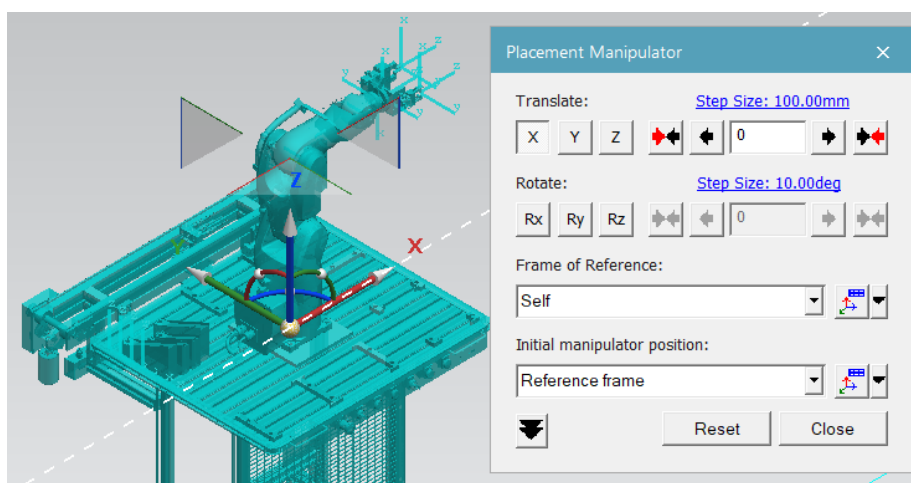


Figura 48: Traslado de la estación mediante el Placement Manipulator

Para distinguir entre un robot de una estación y otra, se les ha cambiado el nombre a “kr3\_r540\_E1” (robot de la primera estación) y “kr3\_r540\_E2” (robot de la segunda estación) y lo mismo con la herramienta de los robots.

Las operaciones que se tienen que ejecutar en la segunda estación son las mismas que en la primera, por lo que, se copian y pegan todas las operaciones en el *Operation Tree*. Estas nuevas operaciones se han creado también sobrepuestas a las de la primera estación. En este caso se utiliza el comando *Single or Multiple Locations Manipulator* para trasladar las operaciones a la otra estación en el eje X 2 m.

También hay que realizar un duplicado de las *Parts* para que se puedan crear los elementos de la segunda estación y se les ha cambiado el nombre. Por ejemplo, el rodamiento de la posición 4 en la primera estación se llama “RodB\_P4” y el de la segunda “RodB\_P4\_2”. Por lo que todas las operaciones de Flujo de Materiales de la nueva estación, hay que cambiarles los *Product Instances* (Instancias de Producto) por las *Parts* de esta segunda estación (véase Figura 49, izq.).

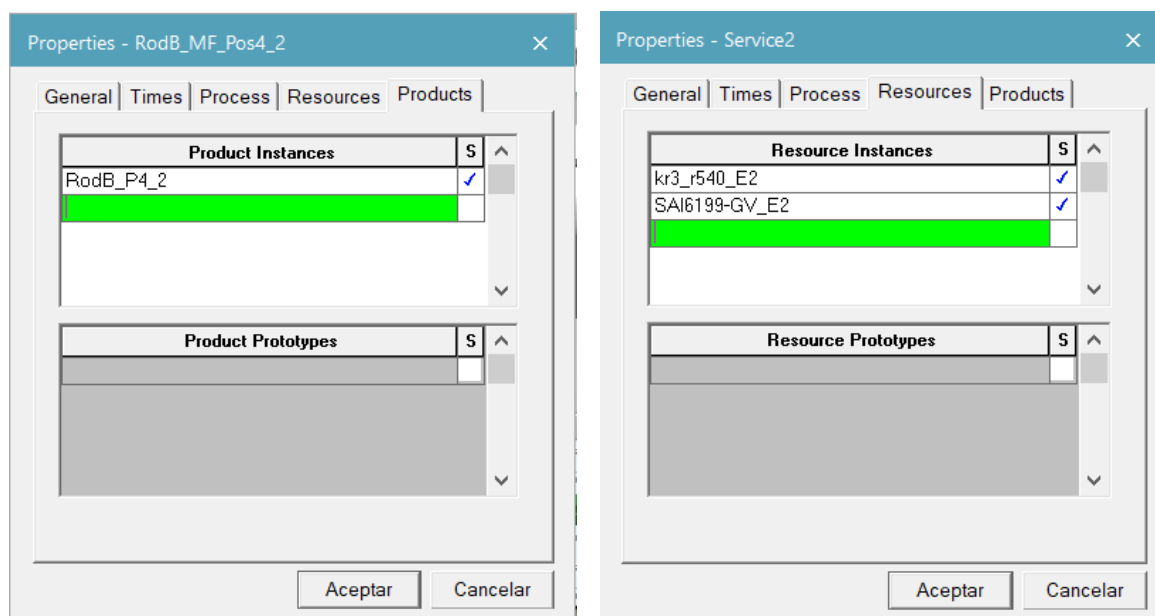


Figura 49: Modificaciones necesarias en las operaciones de la segunda estación

Además de esto, en las operaciones de robot también hay que modificar las instancias de los recursos al nuevo robot y herramienta (véase Figura 49, dcha.) para que estas nuevas operaciones puedan ser realizadas por el robot de la segunda estación.

Sin embargo, los nombres de las señales de control de ambas estaciones serán las mismas debido a que cada robot va a ser controlado por un PLC diferente.

### 2.1.2.7 Flujo de materiales

Una vez creadas todas las operaciones necesarias para la simulación del Gemelo Digital, se debe crear el esquema que representa el camino que realizan los elementos del montaje y los pallets. Este esquema se dibuja en el *Material Flow Viewer* (Visor de Flujo de Materiales) (véanse Figuras 50 y 51).

En primer lugar, se generan los pallets según la solicitud de servicio mediante las operaciones de *Non-Sim* “GeneratePartsCasoX\_Y”. Estas operaciones están unidas con la operación de “Bajar\_Start” que es la encargada de bajar la plataforma en el almacén de entrada, esta unión es debida a que en la operación “Bajar\_Start” está el *Attach Event* donde se fijan los pallets de entrada con el AMR (véase Figura 50, izq.).

Por otro lado, se generan los rodamientos, bulones, tapas interiores y exteriores necesarios de la primera estación para poder realizar el montaje de conjuntos (véase Figura 50, dcha.). Las operaciones de generación de las tapas interiores y exteriores están unidas con las operaciones que simulan la caída de éstas debido a la gravedad cuando son colocadas por el robot en el montaje.

Todas estas operaciones están unidas en la operación OP200 donde mediante los “comandos OLP”, se realiza el ensamblaje (*Attachment*) entre los pallets de entrada y los elementos de montaje. De esta operación se pasa a realizar el descenso de la plataforma del AMR mediante la operación “Bajar\_E1”.

Igual que en la primera estación, en la segunda el esquema es muy parecido (véase Figura 51) ya que, la parte de las operaciones de generación de los elementos de montaje son los mismos. Sin embargo, en esta parte del esquema no se realiza la generación de los pallets de entrada porque el pallet de entrada a esta estación es el montaje realizado en la primera estación. Por lo que, en la operación OP200 de la segunda estación se ensambla el montaje de la primera estación con los elementos colocados en la segunda.

Finalmente, se baja la plataforma con la operación “Bajar\_E2” y se realiza el “KILL” donde se eliminan todos los elementos de la célula.

Cabe destacar que no es necesario colocar las operaciones de subida de la plataforma del AMR porque en estas operaciones no se realiza ningún *Attach*, mientras que en las de bajada sí.

Asimismo, se puede observar que en el esquema de flujo de materiales hay flechas discontinuas. Estas flechas indican caminos alternativos; es decir, en ambas estaciones se puede dar el caso que no se deban montar rodamientos, bulones y/o tapas interiores. Al haberse creado

automáticamente pero no han sido utilizadas, es necesario que al final del proceso sean eliminadas mediante la operación “KILL”, por esta razón, las flechas de estas operaciones tienen por un lado la opción de ser fijadas en la OP200 o de ser eliminadas por el “KILL” de su estación.

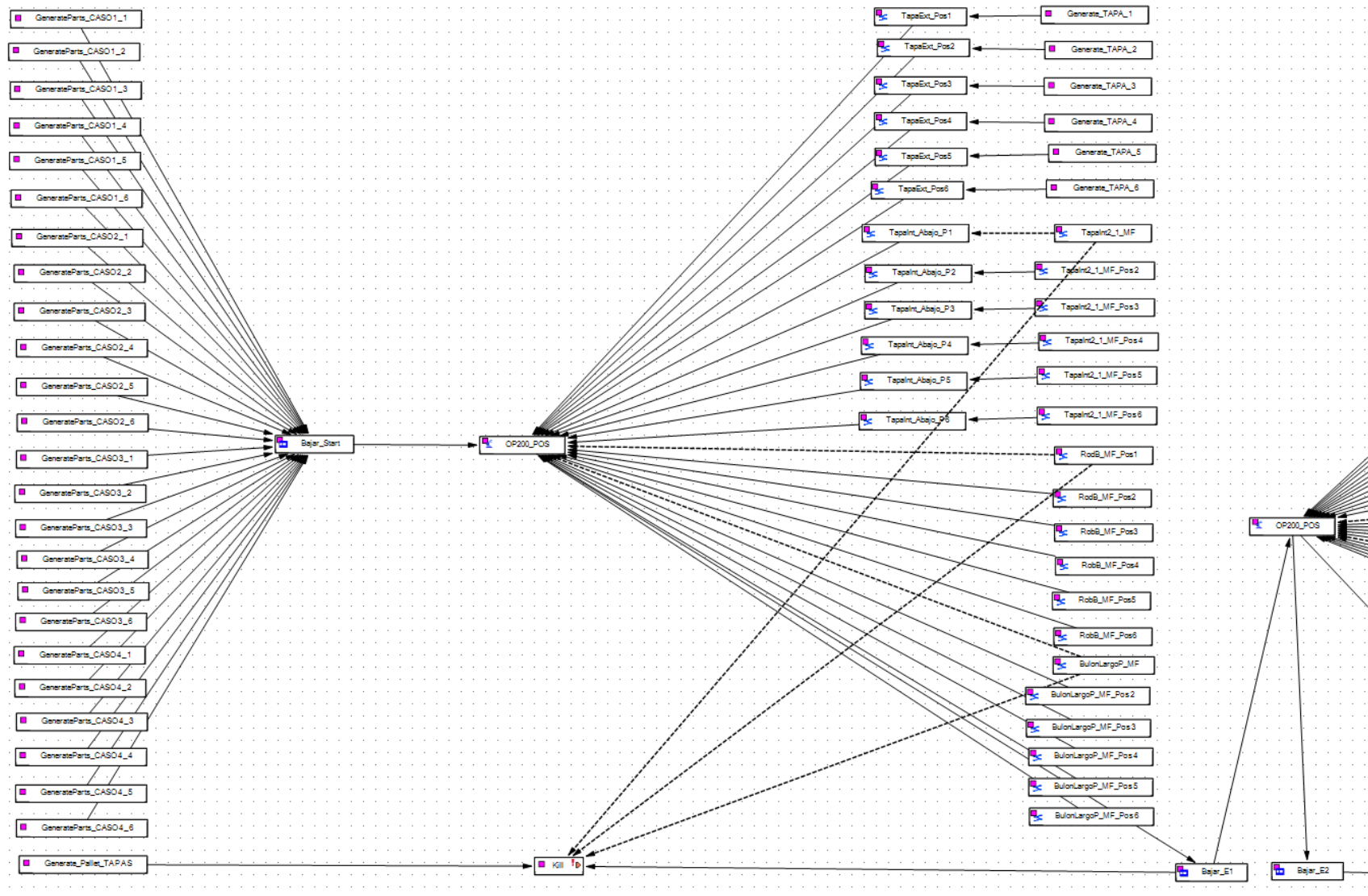


Figura 50: Esquema del Material Flow Parte 1

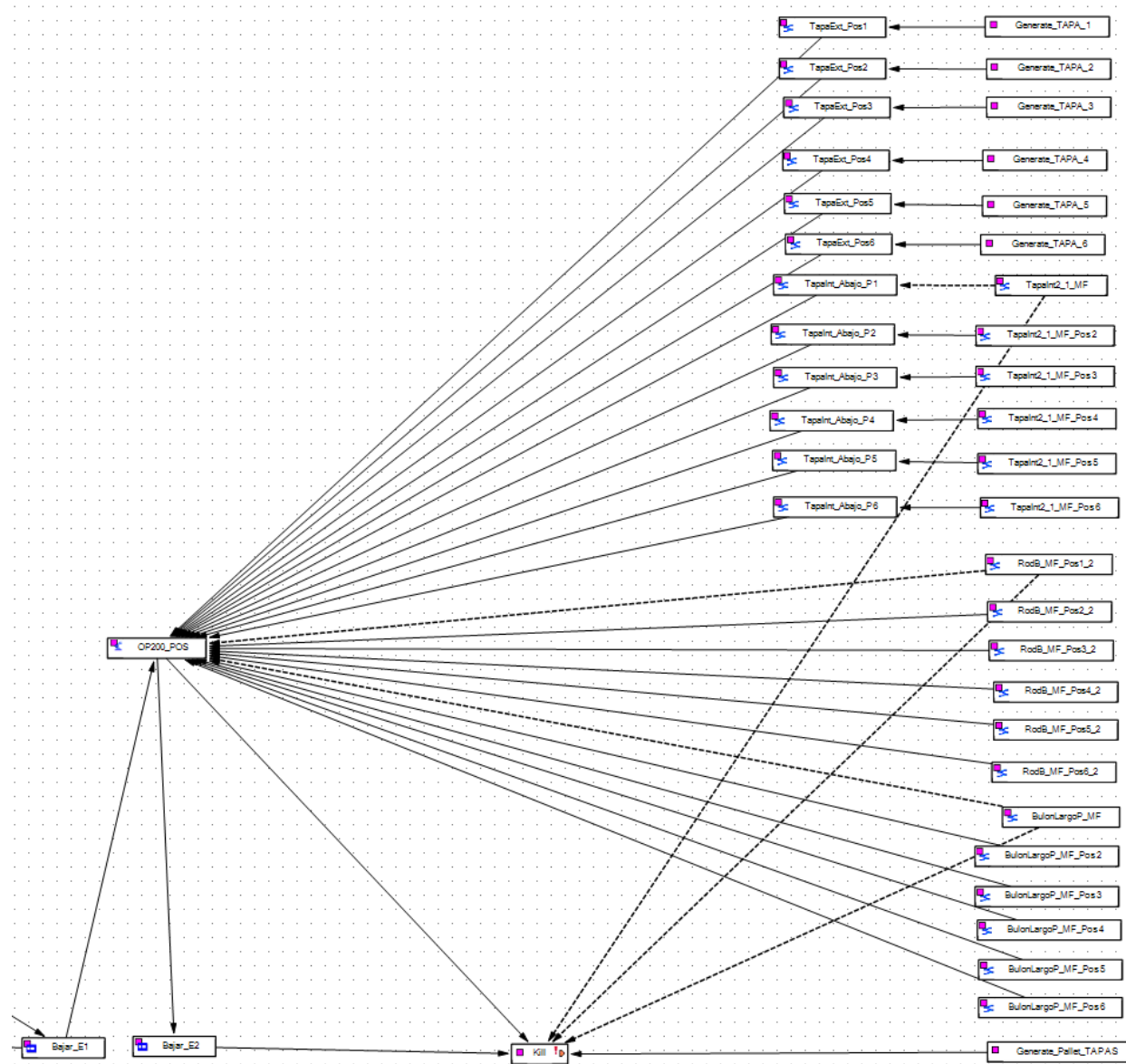


Figura 51: Esquema del Material Flow Parte 2

### 2.1.2.8 Validación del Gemelo Digital en Tecnomatix PS

Después de terminar el modelo de la célula y antes de realizar su software de control, se valida el modelo mediante el “Evaluador de Eventos Cíclicos” (CEE) de la propia herramienta de Tecnomatix PS.

El CEE actúa como centro de control para la simulación. En cada ciclo de simulación, el CEE evalúa las señales de activación de las operaciones para determinar el flujo de la simulación. Debido a que el CEE funciona cíclicamente, la simulación es continua e infinita, empieza al hacer clic en “Reproducir” y finaliza solo cuando se detiene la simulación dando clic al “Stop”.

Este tipo de simulación basada en eventos permite reflejar con precisión el sistema de fabricación propuesto. Asimismo, permite probar y validar la sincronización del robot.

Normalmente este tipo de simulación está activada por defecto, para activarlo, en las opciones de Tecnomatix en el apartado PLC se elige el tipo de simulación CEE (véase Figura 52). La opción PLC se utilizará una vez realizado el software de control para poder realizar la puesta en marcha distribuida mediante *Software in the Loop*.

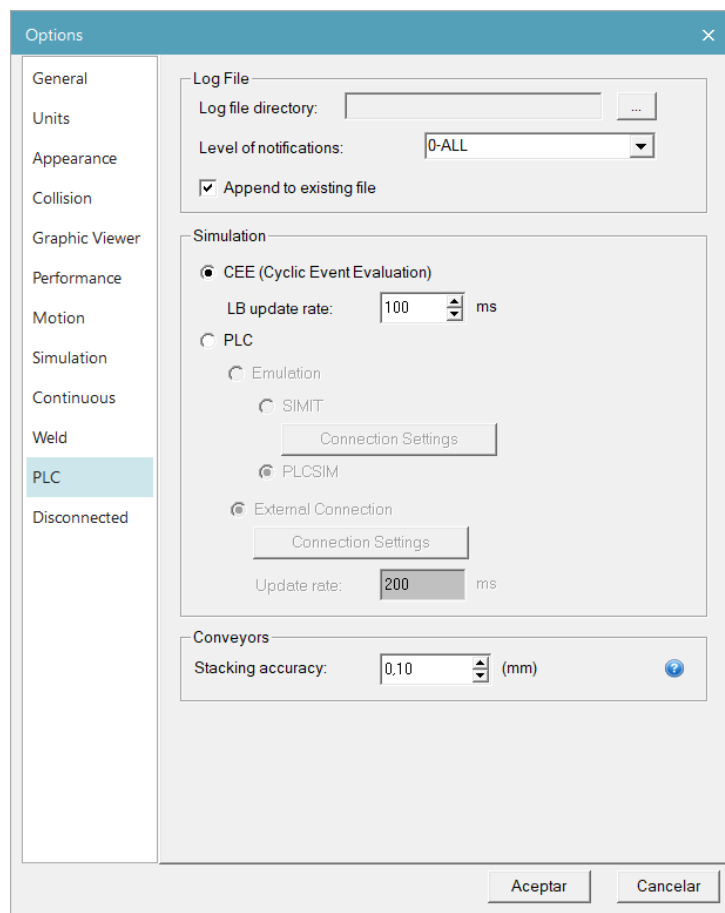


Figura 52: Activación de la simulación mediante CEE

### **2.1.3 Programa de control para la puesta en marcha virtual**

Después de realizar la verificación del Gemelo Digital mediante Tecnomatix PS, se pasa a realizar el programa de control.

En este apartado se describe la realización del programa de control de la célula robotizada mediante la herramienta de ingeniería TIA Portal V16.

En primer lugar, se explica la configuración hardware de la comunicación entre los dispositivos. Seguidamente, se explican los diferentes apartados del software de control de cada PLC y, por último, la configuración del sistema de supervisión HMI.

#### **2.1.3.1 Configuración Hardware de los dispositivos**

En primer lugar, se realiza la configuración hardware de los dispositivos. El proyecto de control de la célula robotizada está compuesto por tres PLCs. Uno por cada estación robotizada y un tercero para el control del AMR. Al mismo tiempo, la supervisión del estado del proceso se realiza mediante un HMI (véase Figura 53).

Dichos PLCs pertenecen a la gama SIMATIC S7-1500 concretamente 1518-4 PN/DP ODK. Se ha elegido esta CPU ya que es la única que soporta el sistema de comunicación ODK, necesario para la comunicación con el sistema multiagente para tramitar las solicitudes de servicio.

En la “Vista de redes” de TIA Portal se configuran las conexiones entre los dispositivos. Para la comunicación de los PLCs con el HMI, se realiza la conexión mediante el puerto X1 de las interfaces Profinet de los PLCs.

A cada dispositivo se le asigna una dirección Ethernet que pertenezca a la misma subred. Por consiguiente, la dirección del puerto X1 del PLC Estación 1 es 192.168.0.120, la del PLC Estación 2 es 192.168.0.121 y la del PLC AMR es 192.168.0.122 y la dirección del HMI es 192.168.0.200 (véase Figura 53).

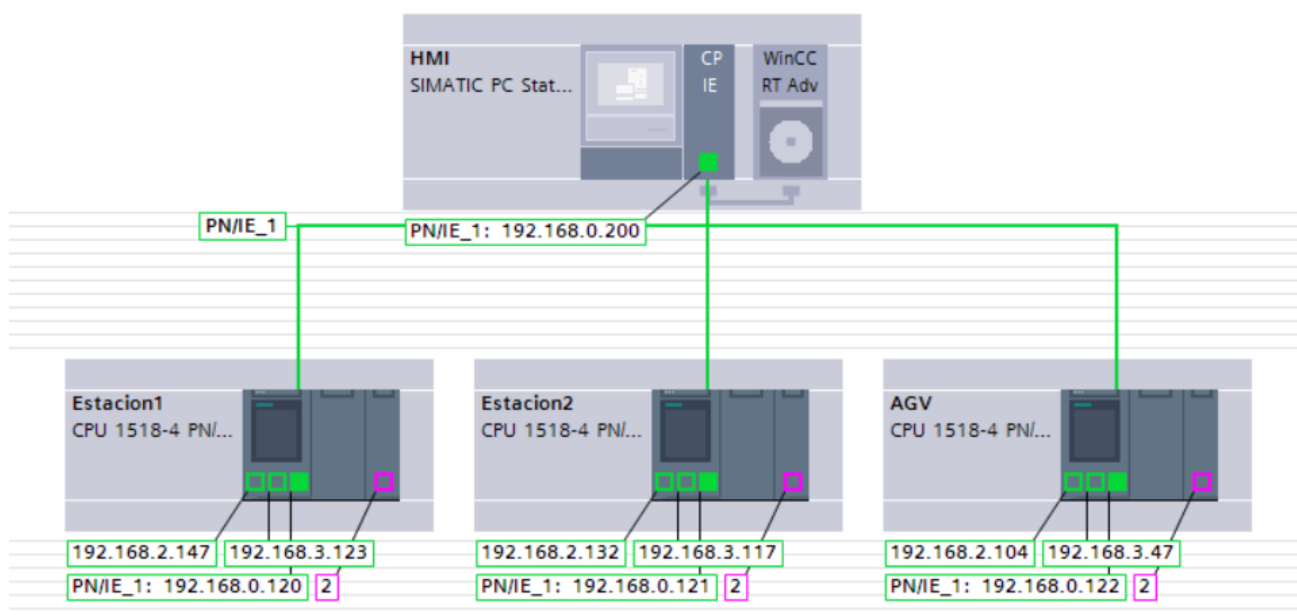


Figura 53: Vista de redes de los PLCs y el HMI

Los PLCs Estación 1 y Estación 2 necesitan enviar y recibir información del PLC del AMR, por ello, entre ellos se realiza una conexión TCP/IP utilizando el mismo puerto que para la conexión con el HMI debido a que estos PLCs tienen un *switch* en el puerto X1.

### 2.1.3.2 Programa de control del PLC del AMR

Para crear al software de control del PLC del AMR se ha creado la tabla de variables (véase Tabla XII). Estas variables serán las mismas que hay en Tecnomatix y las marcas auxiliares se utilizarán para poder completar el control del AMR. Seguidamente, se implementan los siguientes bloques de programa.

Tabla XII: Variables PLC del AMR

Nombre	Tipo de dato	Dirección	Nombre	Tipo de dato	Dirección
Motion planner type	Int	%QW0	Marca_OnTarget	Bool	%M0.0
Path planner type	Int	%QW2	Marca_CurrentTarget1	Int	%MW1
Rotation method	Int	%QW4	Marca_Busy_Aux	Bool	%M0.1
Speed	Real	%QD7	Marca_Busy	Bool	%M0.2
Target index	Int	%QW12	Marca_mOnTarget_fn	Bool	%M0.4
Subir_E1	Bool	%Q6.1	Subir_HMI	Bool	%M4.0
Bajar_E1	Bool	%Q6.2	Bajar_HMI	Bool	%M4.1
Subir_E2	Bool	%Q6.4	Bajar_End_fp_1	Bool	%M1001.2
Bajar_E2	Bool	%Q6.3	Bajar_End_fp_2	Bool	%M1001.3

Bajar_Start	Bool	%Q6.5	Bajar_End_fp_3	Bool	%M1001.4
Bajar_Almacen	Bool	%Q6.6	TCP/IP_Start_AGV	Bool	%M5.2
Subir_Start	Bool	%Q6.7	Marca_Req	Bool	%M5.7
Subir_Almacen	Bool	%Q11.0	Aux_TargetIndex	Bool	%M6.0
Subir_End	Bool	%I0.0	Aux_Arriba	Bool	%M6.1
Bajar_End	Bool	%I0.1	Aux_Kill	Bool	%M6.2
OnTarget	Bool	%I0.2	Aux_onTarget	Bool	%M6.3
CurrentTarget1	Int	%IW1			
Busy	Bool	%I0.4			
Sensor_Caso	Bool	%I0.5			

### 2.1.3.2.1 Startup [OB100]

Este bloque de inicialización se utiliza para cargar valores de inicialización de ciertas variables del AMR (véase Figura 54):

- *Speed* (Velocidad) a 500 mm/s.
- *Motion planner type* (Tipo de planificador de movimiento) a 0; usa la ruta definida en Tecnomatix PS para realizar los movimientos.
- *Path planner type* (Tipo de planificador de trayectoria) a 0; mediante buffer de objetivos (*targets*).
- *Rotation method* (Tipo de rotación) a 2; mantiene la orientación.
- *Target index* (Índice de objetivo) a 3; el AMR empieza la simulación en el almacén de entrada. El *Target index* se transfiere a la "Marca\_CurrentTarget1" que se utilizará a lo largo del programa.
- Si la plataforma del AMR no está subida, ésta ha de subirse.

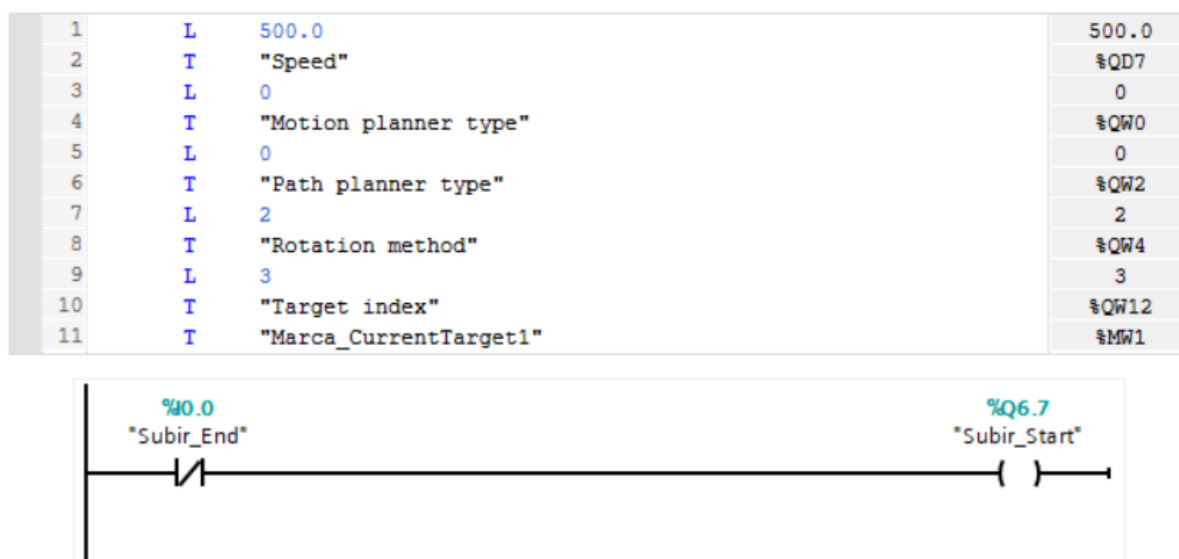


Figura 54: Programación OB100 del PLC del AMR

#### 2.1.3.2.2 Main [OB1]

En el programa principal se instancian los FCs: Automático y TCP/IP. Asimismo, se establecen las marcas que se van a utilizar para la visualización de variables en el HMI. Por último, se realiza el funcionamiento en manual de la plataforma del AMR que funcionará mediante pulsadores en el HMI.

#### 2.1.3.2.3 Automático [FC]

En esta función se programa el funcionamiento automático del AMR. Esta función activa y desactiva las *Device Operation* creadas en Tecnomatix PS para subir y bajar la plataforma y también se establece el camino que debe realizar el AMR. Esta parte del programa se realiza mediante una secuencia (véase Figura 55).

Cuando el AMR esté en el almacén de entrada la plataforma sube ("Subir\_Start" se activa) hasta que haya llegado arriba (el sensor "Subir\_End" se activa). Cuando el sensor fotoeléctrico detecte que hay un pallet en la plataforma ("Sensor\_Caso" se activa), ésta baja hasta que el sensor indique que la plataforma está abajo (el sensor "Bajar\_End" se activa). Cuando la plataforma haya bajado, se actualiza el "Target index" a 1 para que el AMR vaya a la primera estación.

Cuando el AMR esté en la primera estación la plataforma del AMR sube ("Subir\_E1" se activa) hasta que haya llegado arriba. Cuando la primera estación haya terminado de realizar el servicio, la plataforma baja y cuando haya bajado, se actualiza el "Target index" a 2 para que el AMR vaya a la

segunda estación. Lo mismo ocurre en la segunda estación y después de terminar su servicio se actualiza el “Target index” a 4 para que el AMR vaya al almacén de salida.

Cuando el AMR llegue al almacén de salida, se hace la operación “KILL” con la que desaparecen todos elementos de Tecnomatix y se actualiza el “Target index” a 3 para que el AMR vaya al almacén de entrada.

En esta misma función se actualiza la información que se transmite a los PLCs de las estaciones mediante TCP/IP. Los datos que se envían son la marca indicando si el AMR está en algún *target*, si la plataforma está subida, la actualización del “Target index” y si el sensor de la plataforma indica que hay un pallet sobre la plataforma.

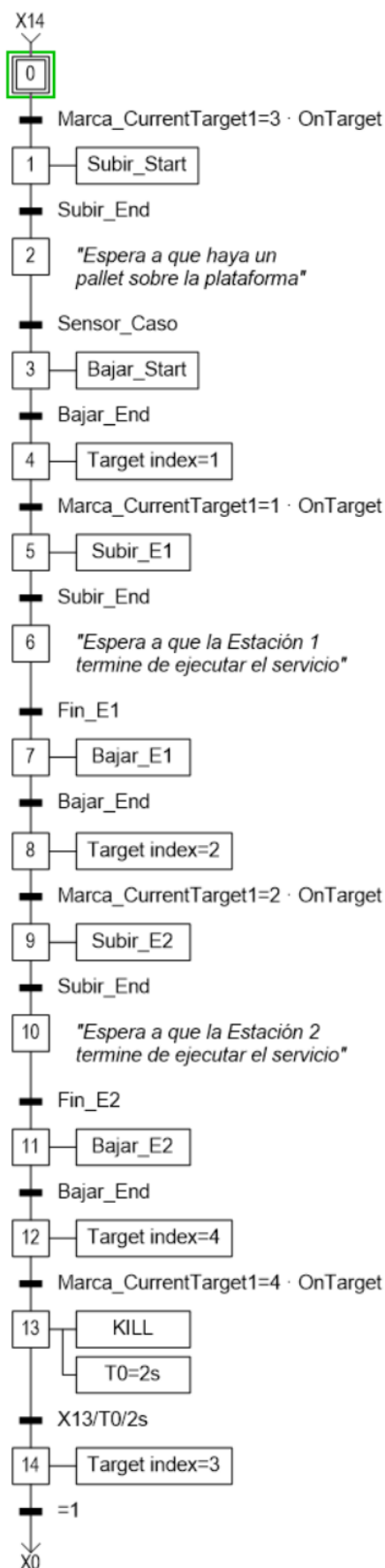


Figura 55: Secuencia funcionamiento automático AMR

#### 2.1.3.2.4 TCP/IP [FC]

En esta función se encuentra la configuración de la comunicación vía TCP/IP. Para realizar la conexión se utilizan los bloques: “TSEND\_C” y “TRCV\_C”. Estos bloques se ejecutan de forma asíncrona y tiene las siguientes funciones:

- Configurar y establecer una conexión
- TSEND\_C se envían datos a través de la conexión existente
- TRCV\_C se reciben datos a través de la conexión existente
- Deshacer o inicializar la conexión

Para realizar la conexión de los PLCs de las estaciones con el PLC del AMR, es imprescindible tener cuatro enlaces de conexión desde el PLC del AMR a los PLCs de las estaciones. Por lo tanto, se necesitan dos bloques “TSEND\_C” para enviar información del PLC del AMR a los PLCs de las estaciones y otros dos “TRCV\_C” para recibir información de las estaciones al AMR.

La configuración de los parámetros de la conexión de todos bloques es la misma (véase Figura 56).

La CPU local es la del AMR y el interlocutor son las CPUs de las estaciones. El tipo de conexión es TCP y el modo de configuración que se emplea es utilizar bloque de programa donde se generan los DBs donde están los datos de la conexión. En este caso el establecimiento activo a la conexión está en la CPU del PLC del AMR. En la conexión entre las CPUs del AMR y de la Estación 1 el puerto del interlocutor utilizado en el bloque “TSEND\_C” es el 2000 y en “TRCV\_C” 2001. En la conexión entre las CPUs del AMR y de la Estación 2 el puerto del interlocutor utilizado en el bloque “TSEND\_C” es el 3000 y en “TRCV\_C” 3001.

Figura 56: Configuración de los parámetros de la conexión en “TSEND\_C” y “TRCV\_C”

A continuación, se configuran los parámetros del bloque de envío “TSEND\_C” (véase Figura 57). El inicio de la petición de envío (denominado “REQ” en el bloque “TSEND\_C”) de la información del AMR a las estaciones se da cuando se cumple alguna de las siguientes condiciones:

- El “Target index” es el mismo que el “CurrentTarget”
- La plataforma del AMR está arriba
- Se ha realizado el “KILL”
- El AMR está en algún objetivo; es decir, “OnTarget” está activo
- El sensor de la plataforma indica que hay un pallet; es decir, “Sensor\_Caso” está activo

La petición de envío se desactiva cuando se indique que la petición se ha ejecutado correctamente mediante la activación de la salida DONE del bloque.

La conexión (denominado “CONT” en el bloque “TSEND\_C”) se realiza cuando se le dé al pulsador del HMI de "TCP/IP\_Start\_AMR" y pasen 0.1s.

Mediante un puntero se especifica el área de datos que quiere transferirse (denominado “DATA” en el bloque “TSEND\_C”), en este caso apunta al primer bit del DB “TCP/IP\_AMR\_SEND” y la longitud de datos a transferir es de 3 bytes.

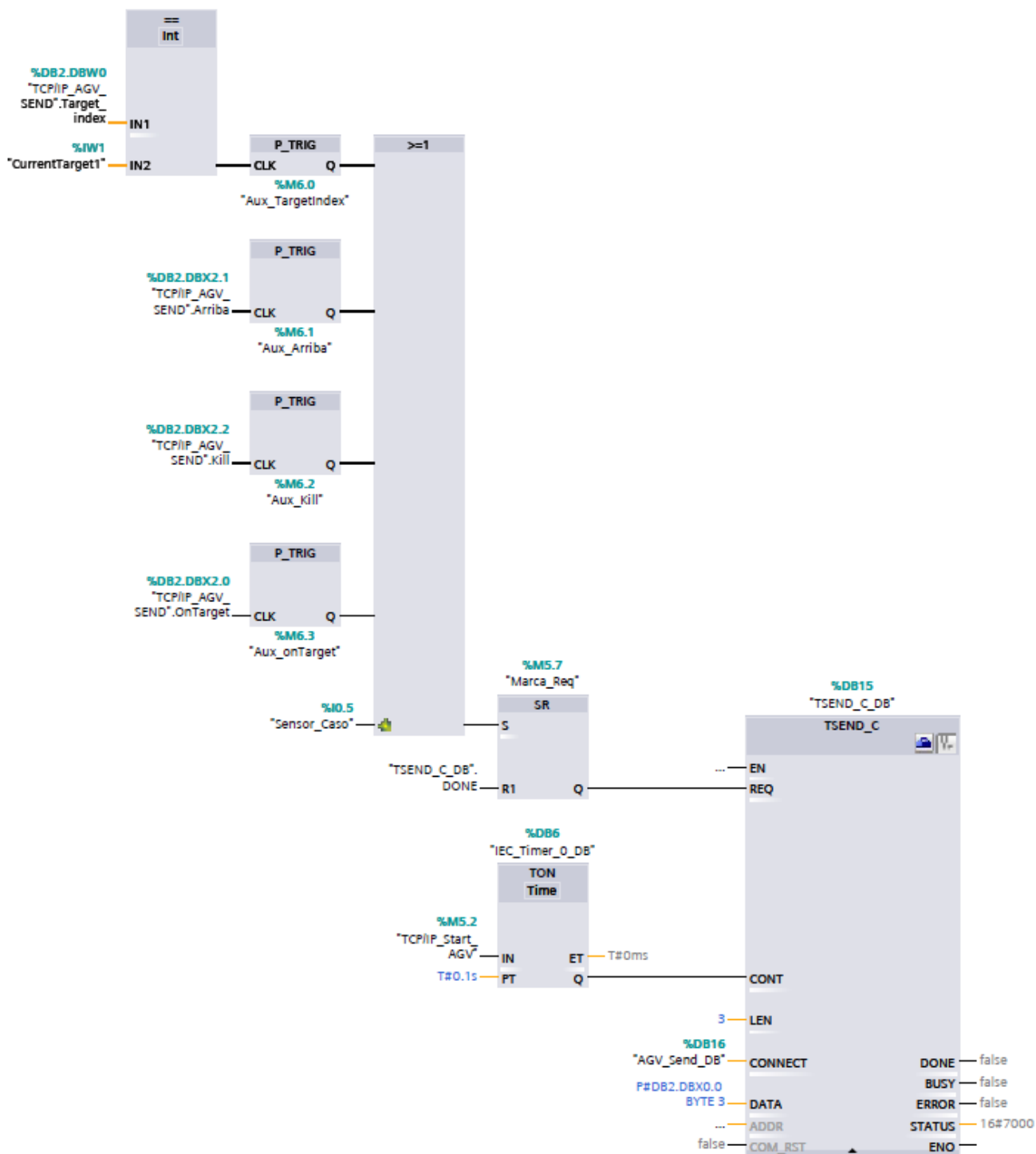


Figura 57: Configuración de los parámetros del bloque “TSEND\_C”

De igual forma, se configuran los parámetros de los bloques para recibir “TRCV\_C” (véase Figura 58). Se habilita la recepción de información y se realiza la conexión una vez pulse el pulsador del HMI de "TCP/IP\_Start\_AMR" y pasen 0.1s.

Mediante un puntero se especifica el área de datos que quiere transferirse (denominado “DATA” en el bloque “TRCV\_C”), en este caso, apunta al primer bit del DBs que se reciben de las estaciones y la longitud de datos a transferir es de 3 bytes.

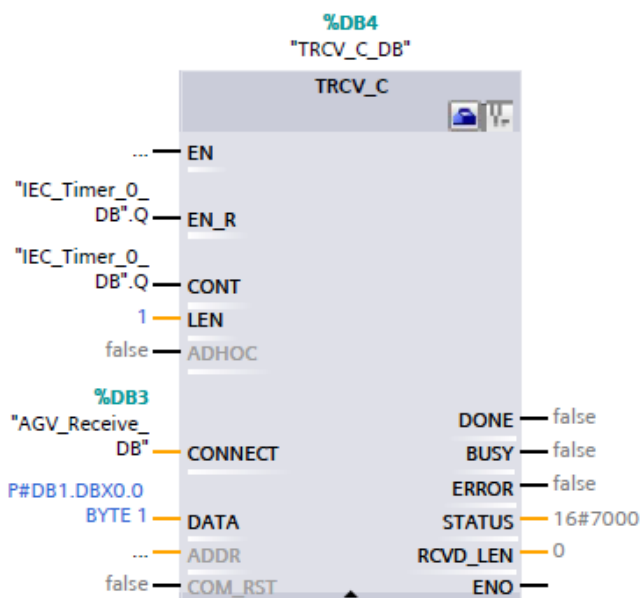


Figura 58: Configuración de los parámetros del bloque “TRCV\_C”

De esta misma forma, se configuran los bloques “TSEND\_C” y “TRCV\_C” de los PLCs de las estaciones.

#### 2.1.3.2.5 TCP/IP\_AMR\_REC\_E1 [DB1]

Estos DBs se utilizan para recibir la información de la CPU Estación 1 (véase Tabla XIII). El valor que se recibe es una variable booleana que indica que se ha terminado el servicio en la primera estación.

Tabla XIII: Variables que se reciben desde la primera estación

Nombre	Tipo de datos	Offset
Fin_E1	BOOL	0.0

#### 2.1.3.2.6 TCP/IP\_AMR\_REC\_E2 [DB14]

Este DB se utiliza para recibir la información de la CPU Estación 2 (véase Tabla XIV). El valor que se recibe es una variable booleana que indica que se ha terminado el servicio en la segunda estación.

Tabla XIV: Variables que se reciben desde la segunda estación

Nombre	Tipo de datos	Offset
Fin_E2	BOOL	0.0

#### 2.1.3.2.7 TCP/IP\_AMR\_SEND [DB2]

Este DB se utiliza para enviar la información del AMR a las estaciones (véase Tabla XV). Se envía el índice del objetivo, si el AMR está en algún objetivo, si la plataforma está arriba, si se ha realizado el “KILL” y si hay algún pallet sobre la plataforma.

Tabla XV: Variables que se envían desde el AMR

Nombre	Tipo de datos	Offset
Target_Index	INT	0.0
OnTarget	BOOL	2.0
Arriba	BOOL	2.1
Kill	BOOL	2.2
Sensor	BOOL	2.3

### 2.1.3.3 Programa de control de las estaciones

Los programas de control de las estaciones son similares, por lo que, lo que se comenta sobre una de las estaciones es atribuible a la otra estación.

En primer lugar, se definen variables PLC que se van a utilizar para el control de las estaciones. Estas variables se han dividido en diferentes tablas (véase Tabla XVI).

Tabla XVI: Índice de las diferentes tablas de variables

Tabla de Variables	Descripción
GenerateParts	Variables que controlan la generación del flujo de materiales
Gripper	Variables Tecnomatix PS de las pinzas

KUKA Signals	Variables de la estación real, las cuales se comunican vía Profinet con la unidad de control KR C4 Compact
ODK Signals	Variables para la comunicación entre el PLC y el Agente Máquina
Robot Signals	Variables del robot que se usan para la simulación en Tecnomatix PS
Stage Sensors	Sensores de los alimentadores
TCP/IP	Variables usadas para la comunicación con el PLC del AMR vía TCP/IP

#### 2.1.3.3.1 Startup [OB100]

Este bloque de inicialización se utiliza para cargar el valor 1 en el número de programa del robot (véase Figura 59), así Tecnomatix PS podrá ejecutar la operación robot MAIN.

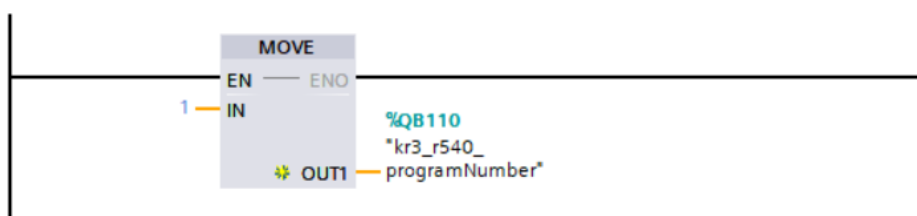


Figura 59: Programación OB100 del PLC de las estaciones

#### 2.1.3.3.2 Main [OB1]

Se instancian las funciones y los bloques de funciones generados en el programa.

#### 2.1.3.3.3 ComunicacionODK [FC] entre PLC y el Agente Máquina

Como se ha indicado anteriormente este trabajo es parte de un proyecto en el cual el control de procesos se realiza mediante un sistema multiagente. En este sistema se encuentra el denominado “Agente Máquina” encargado del control de la solicitud de servicios de la célula. Dicho Agente se ha de comunicar con el PLC. Sin embargo, es preciso destacar que el PLC del AMR no necesita esta comunicación con el Agente Máquina, ya que el AMR no recibe ninguna información de éste. La información que recibe el PLC del AMR viene de los PLCs de las estaciones mediante la comunicación TCP/IP anteriormente descrita.

Para crear las aplicaciones para dicha comunicación se usa la herramienta “Microsoft Visual Studio”. Estas aplicaciones se han desarrollado utilizando SIMATIC ODK 1500S. SIMATIC ODK es un paquete de desarrollo que se utiliza como interfaz para llamar a aplicaciones de lenguaje de alto

nivel dentro del programa de control de la CPU [20]. Esta aplicación es el sistema multiagente desarrollado en C.

Usando estas dos herramientas se generan dos archivos para la comunicación con TIA Portal. Uno de los archivos tiene extensión \*.scl que permite integrar las aplicaciones del sistema multiagente en el código de control del PLC. Este archivo se agrega a la CPU de las estaciones como “Fuentes externas” (véase Figura 60). Después de agregarlos, se generan automáticamente los FBs para la comunicación que se utilizarán en esta función (véase Figura 61).

Los FBs denominados “GetTrace”, “ODKLoad” y “ODKUnload” se encargan de la carga del sistema ODK. Los dos últimos realizan la conexión y desconexión respectivamente, para ello, se utilizan pulsadores del HMI.

También existen otros dos FBs uno para la lectura: “ODKProject2SampleRead” y otro para la escritura: “ODKProject2SampleWrite”.

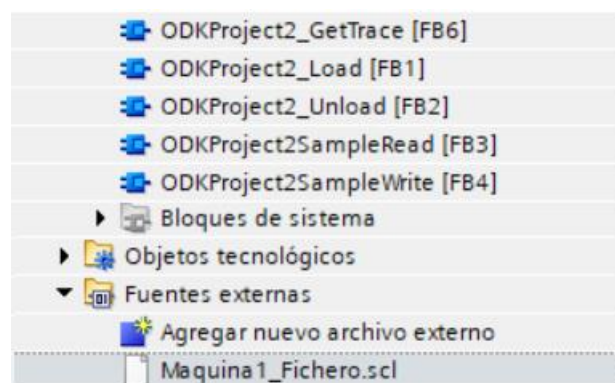


Figura 60: Archivo \*.scl y los FBs creados

El otro archivo tiene extensión \*.dll contiene la aplicación ODK para comunicarse con el entorno Windows y se utiliza en la instancia del PLC virtual en PLCSIM Advanced. En posteriores apartados este procedimiento se describirá más detalladamente.

Mediante esta función, se realiza la comunicación entre el PLC y el Agente Máquina. Para ello, se utilizan los FBs creados a partir del archivo de extensión \*.scl.

Actualmente, el sistema multiagente se encuentra en desarrollo, en consiguiente, no se dispone del “Agente Máquina”. Por lo que, para realizar este proyecto, el propio usuario simula ser el agente y escribirá la solicitud de servicio mediante un fichero de texto denominado “ODK2PLC” (véase Tabla XVII) y el PLC leerá dicha solicitud (véase Figura 61). En esta solicitud aparecen los siguientes datos:

Tabla XVII: Información fichero ODK2PLC

• Id_Machine_Reference	Referencia de la máquina
• Id_Order_Reference	Referencia de la petición u orden
• Id_Batch_Reference	Referencia del lote
• Id_Ref_Subproduct_Type	Referencia del tipo de subproducto
• Operation_Ref_Service_Type	Referencia del tipo de servicio a realizar
• Operation_No_of_Items	Número de ítems a montar

Al mismo tiempo, se realiza la escritura de los datos de la ejecución de los servicios en otro archivo de texto denominado “PLC2ODK” (véase Figura 61), en el siguiente apartado se detallan los datos que se escriben en este fichero.

Para la lectura y escritura de estos ficheros, se utiliza un DB denominado “TelegramaAgente” (véase Tabla XVIII).

Tabla XVIII: Variables “TelegramaAgente”

Nombre	Tipo de datos
str2Info	"ODKProject2control_flags"
• Control_Flag_New_Service	BOOL
• Control_Flag_Item_Completed	BOOL
• Control_Flag_Service_Completed	BOOL
str2PLC	"ODKProject2agent2plc"
• Id_Machine_Reference	UDInt
• Id_Order_Reference	UDInt
• Id_Batch_Reference	UDInt
• Id_Ref_Subproduct_Type	UDInt
• Operation_Ref_Service_Type	UDInt
• Operation_No_of_Items	USInt
str2Agent	"ODKProject2plc2agent"
• Id_Machine_Reference	UDInt
• Id_Order_Reference	UDInt
• Id_Batch_Reference	UDInt
• Id_Ref_Subproduct_Type	UDInt
• Id_Ref_Service_Type	UDInt
• Id_Item_Number	USInt

• Data_Initial_Time_Stamp	LDT
• Data_Final_Time_Stamp	LDT
• Data_Service_Time_Stamp	LDT

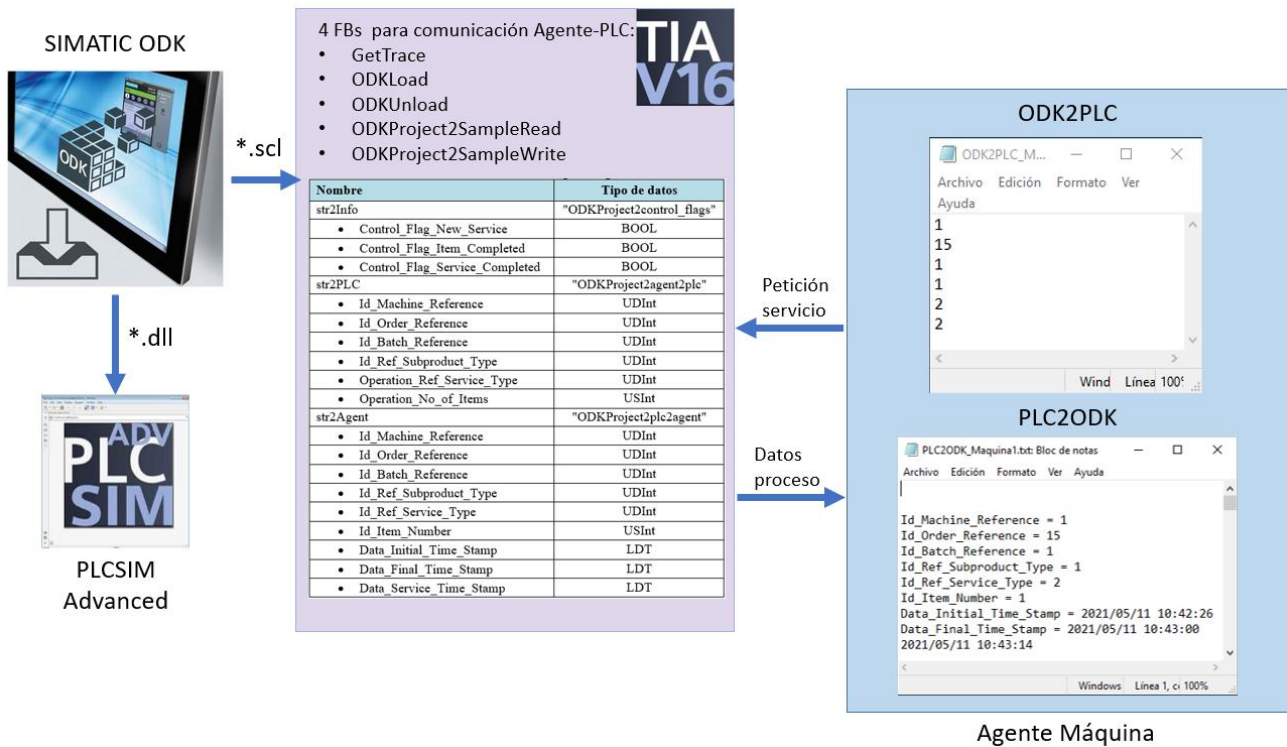


Figura 61: Comunicación ODK

#### 2.1.3.3.4 GestionarPedido [FC]

En esta función se evalúa la solicitud realizada por el “Agente Máquina” sobre el servicio y número de ítems que se han de montar en la estación. Para que el pedido sea ejecutable por el robot se deben cumplir las siguientes condiciones:

- El tipo de servicio tiene un valor entre 1 y 10
- El número de ítems tiene un valor entre 1 y 6

Si no se cumple alguna de estas condiciones, se activa la alarma del HMI de fallo de tipo de servicio y/o fallo en el número de ítems.

En el caso que se cumplan las condiciones, el pedido pasa a ejecutarse. Por ello, los valores del tipo de servicio y el número de ítems se transmiten vía Profinet a la unidad de control para que el robot ejecute el pedido cuando se haga la puesta en marcha real.

### 2.1.3.3.5 *EscribirDatosAgente [FC]*

Mediante esta función, durante la ejecución de los servicios, se guardan en el DB “TelegramaAgente” la información de la ejecución de los servicios. Por medio de la aplicación ODK esta información se copia en el fichero “PLC2ODK” (véase Tabla XIX). Los datos que se escriben son los siguientes:

Tabla XIX: Información del fichero PLC2ODK

• Id_Machine_Reference	Referencia de la máquina
• Id_Order_Reference	Referencia de la petición u orden
• Id_Batch_Reference	Referencia del lote
• Id_Ref_Subproduct_Type	Referencia del tipo de subproducto
• Id_Ref_Service_Type	Referencia del tipo de servicio realizado
• Id_Item_Number	Número de ítems montados
• Data_Initial_Time_Stamp	Hora de inicialización de montaje de un ítem
• Data_Final_Time_Stamp	Hora de finalización de montaje de un ítem
• Data_Service_Time_Stamp	Hora de inicio de la ejecución del servicio

### 2.1.3.3.6 *ControlRobot [FB]*

En este bloque de función, se habilita el movimiento del robot mediante una secuencia de activación de variables del robot mediante temporizadores (véase Figura 62). Para ello, mediante el pulsador de marcha (PM) se habilita el movimiento del robot activando \$MOVE\_ENABLE.

Una vez habilitado el movimiento, se activa la señal \$DRIVES\_OFF que indica que no está activada la desconexión de los accionamientos. Esta señal debe permanecer activa en todo momento ya que, si se desactiva, se desactivan los accionamientos del robot. Seguidamente se activa \$DRIVES\_ON que conecta la señal de los accionamientos y está activa durante 1 segundo (como mínimo ha de estar activada 200ms).

Después se activa la señal \$CONF\_MESS para confirmar los posibles mensajes que hayan aparecido en la unidad de control. Por último, se activa el arranque externo \$EXT\_START para poder utilizar el PLC como control.

Una vez completada esta secuencia, el robot está listo para ejecutar los servicios.

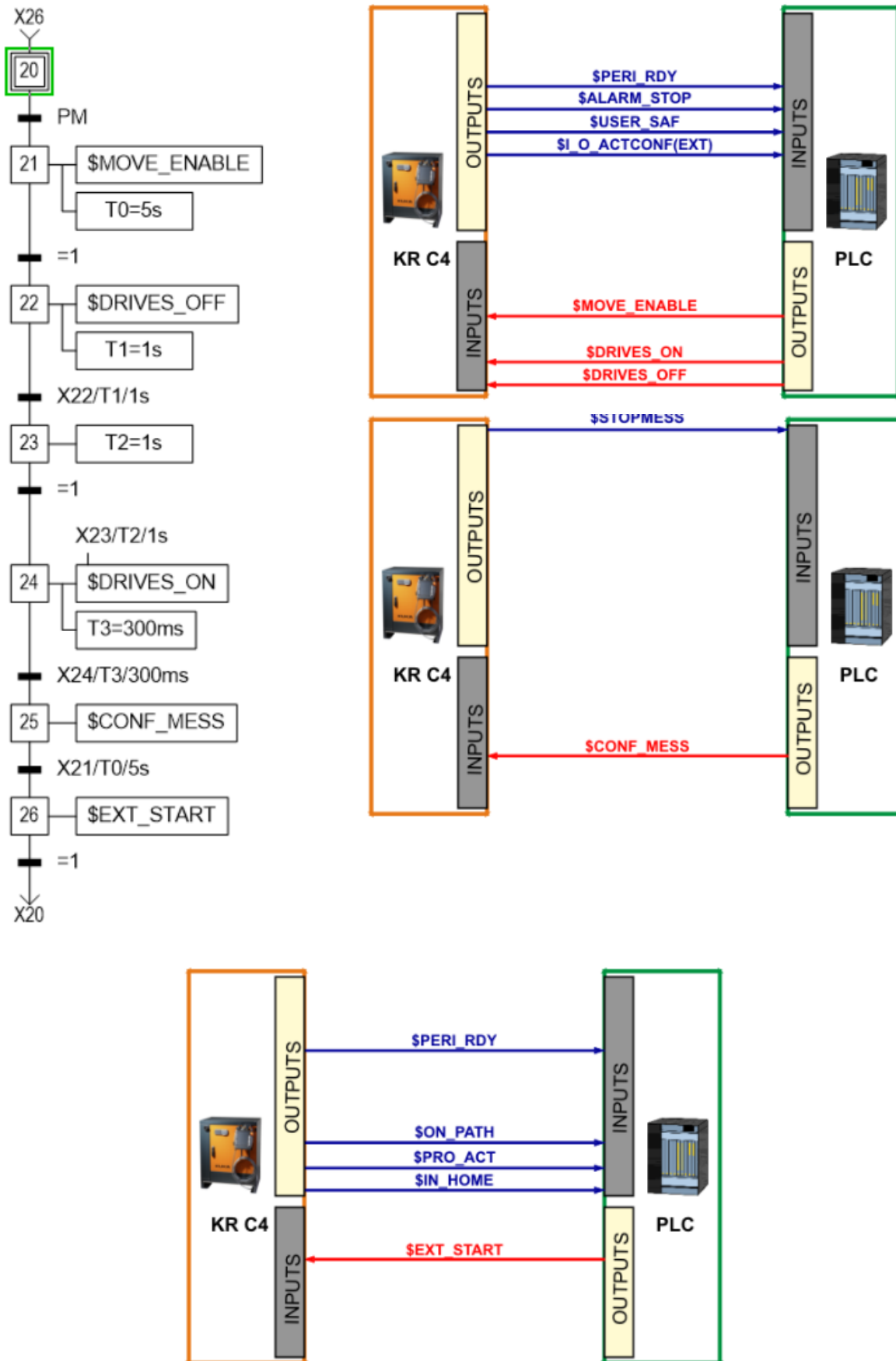


Figura 62: Variables compartidas entre PLC y la unidad de control KR C4

### 2.1.3.3.7 Material Flow [FC]

Esta función solo es necesaria cuando se realiza la puesta en marcha virtual de la célula en Tecnomatix PS, ya que mediante esta función se realiza la generación de los diferentes elementos en el Gemelo Digital.

En función del tipo de servicio solicitado y el número de ítems a montar, se genera un “Caso” en los pallets de entrada (véase Tabla VII). Lo mismo ocurre con la generación de las tapas exteriores.

Por otro lado, la generación de los rodamientos, bulones y tapas interiores depende de la señal del sensor, el valor del contador “kr3\_r540\_Counter” y si ya ha finalizado la generación del anterior elemento (véase Figura 63).

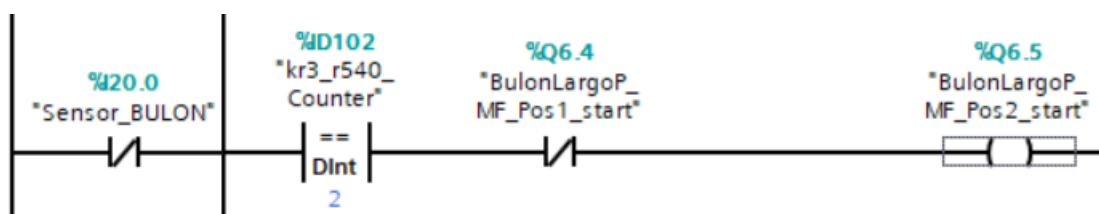


Figura 63: Lógica de la generación del bulón de la posición 2

El movimiento de la caída de las tapas interiores y exteriores simulando la gravedad (véase Figura 64) se hacen si la garra está abierta, si la tapa es interior o exterior y si el robot está en una de las posiciones denominadas “kr3\_r540\_at\_Tapa\_PZ”. El movimiento se resetea cuando el robot regresa a su posición HOME.

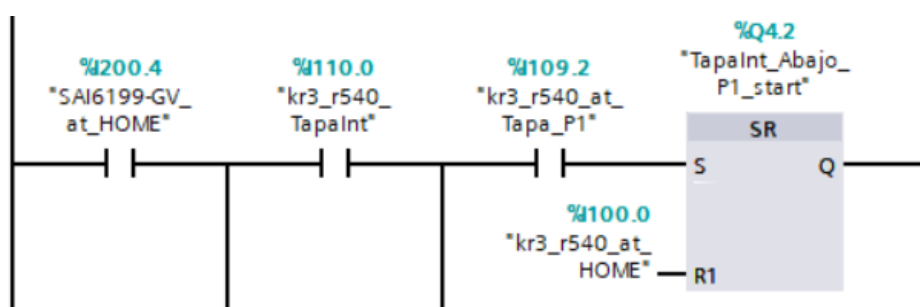


Figura 64: Lógica del movimiento por gravedad de la tapa interior posición 1

Finalmente, se activa la operación “KILL” para hacer desaparecer todos los elementos de la célula. Esta señal se activa cuando el contador se ha reseteado, el robot está en la posición HOME y se ha recibido desde el AMR la orden para hacer el “KILL”.

### 2.1.3.3.8 TCP/IP [FC]

Para la comunicación TCP/IP se utilizan los mismos bloques que en el programa de control del AMR: “TSEND\_C” y “TRCV\_C” para complementar la comunicación con el PLC del AMR. En el caso de las estaciones solo se necesita un bloque para enviar y otro para recibir. Estos bloques se configuran de la misma manera que en el PLC del AMR. A continuación, se muestra la configuración en la primera estación, pero es semejante en la segunda estación.

En este caso, en los parámetros de la conexión de los bloques (véase Figura 65), la CPU local son las CPUs de las estaciones y el interlocutor la CPU del AMR. El tipo de conexión es TCP y el modo de configuración que se emplea es utilizar bloque de programa donde se generan los DBs donde están los datos de la conexión. En este caso el establecimiento activo a la conexión también está en la CPU del PLC del AMR.

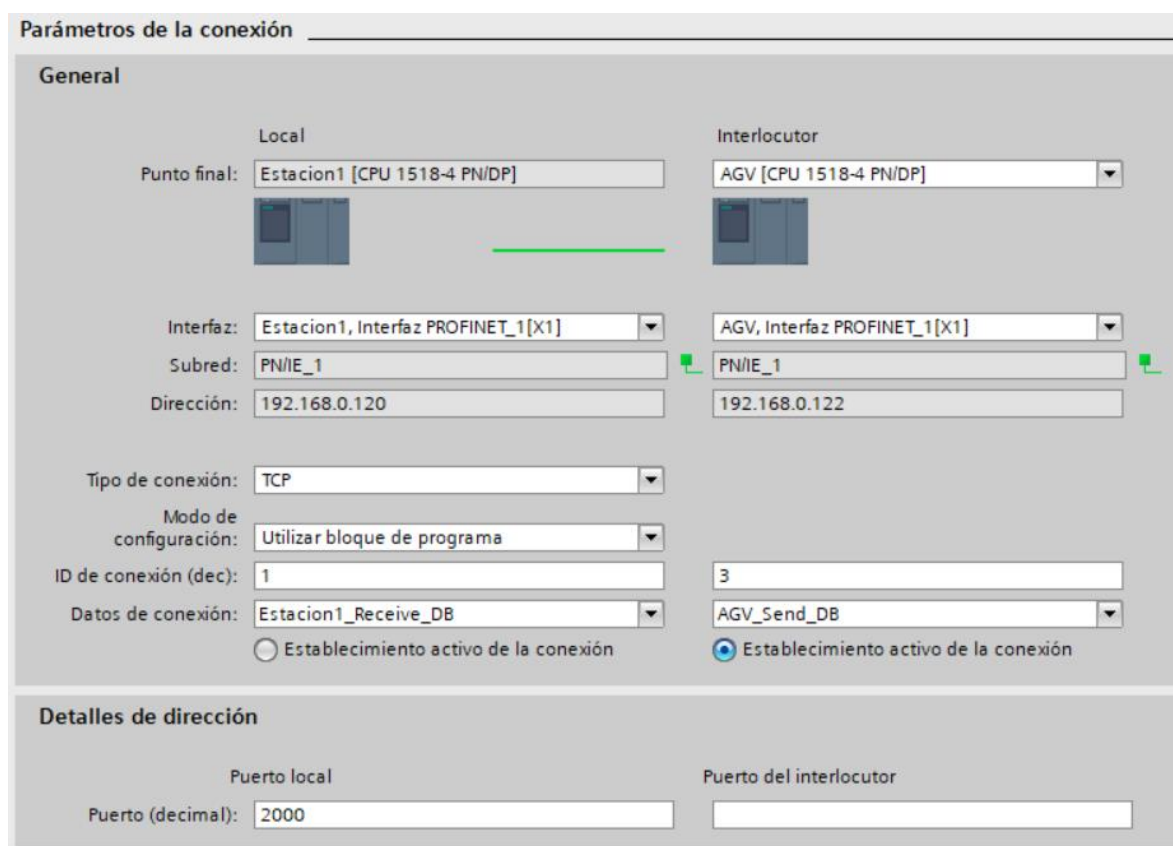


Figura 65: Parámetros de la conexión de los bloques “TSEND\_C” y “TRCV\_C” de las estaciones

A continuación, se configuran los parámetros del bloque “TSEN\_C” (véase Figura 66). El inicio de la petición de envío al AMR se da cuando se ha finalizado la ejecución del servicio de la estación. El dato que se envía es la “Fin\_E1” que indica que la primera estación ha terminado de ejecutar el servicio solicitado.

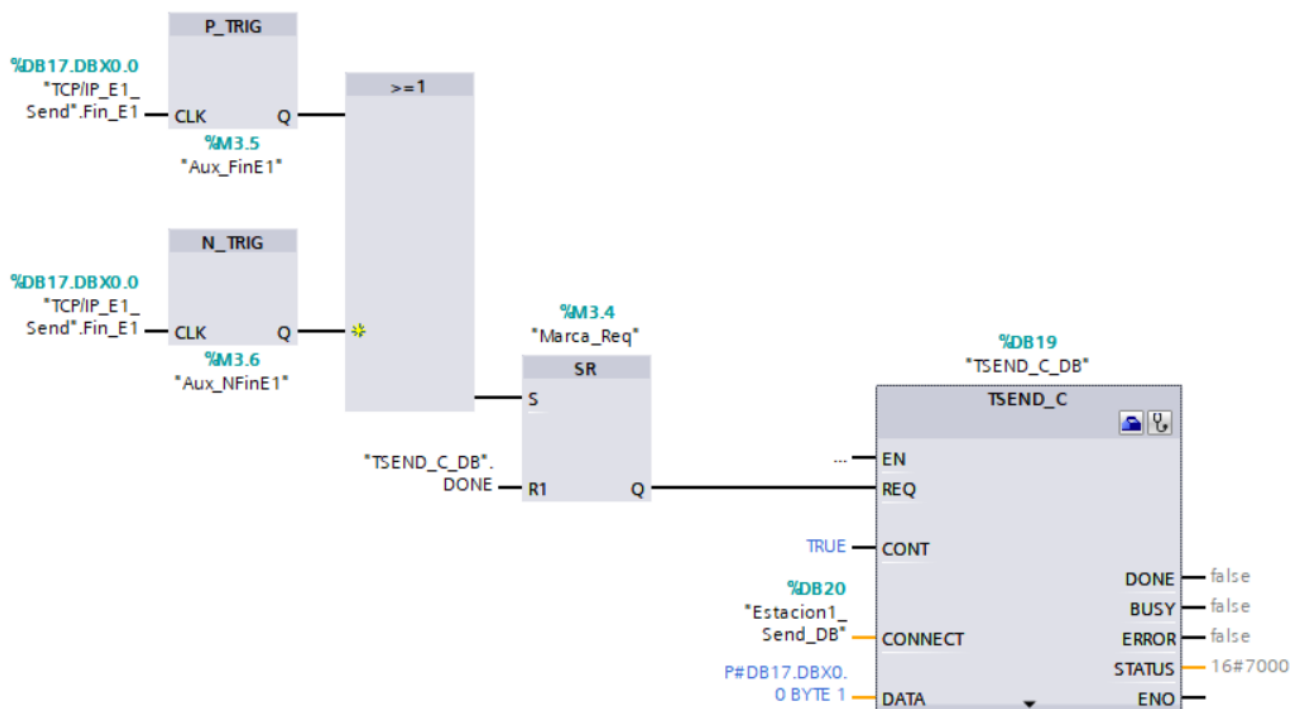


Figura 66: Parámetros del bloque “TSEND\_C” de las estaciones

En la configuración del bloque “TRCV\_C” (véase Figura 67), la conexión de la comunicación se realiza cuando se le dé al pulsador del HMI y mediante un puntero se especifica el área de datos que quiere transferirse. En este caso se recibe la información del AMR (véase Tabla XV).

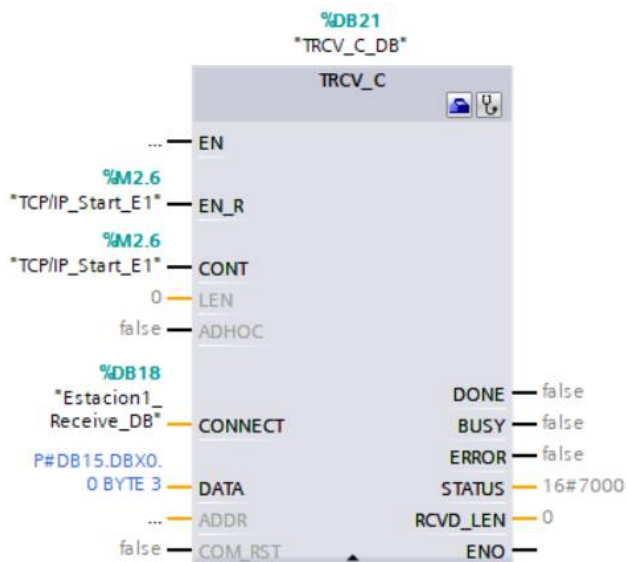


Figura 67: Parámetros del bloque “TRCV\_C” de las estaciones

### 2.1.3.4 HMI

Para realizar el control y la supervisión del proceso se ha diseñado y configurado una interfaz HMI (véase Figura 69). Para ello, es necesario configurar las conexiones entre los PLCs y la interfaz HMI con comunicación S7 (véase Figura 68). En este caso existen tres comunicaciones HMI una con cada PLC del proyecto: PLC de las estaciones y del AMR.

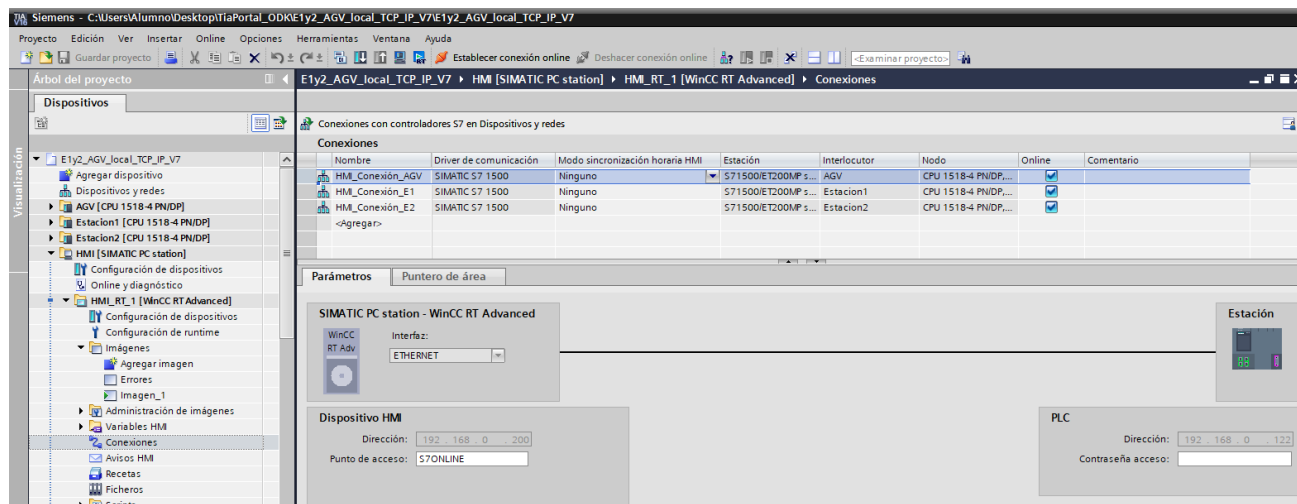


Figura 68: Comunicación S7 entre los PLCs y el HMI

Una vez establecidos los enlaces, se rellena la tabla de variables HMI con las variables de control y de proceso necesarias para poder observar y controlar la ejecución de la célula robotizada (véase Tabla XX). En esta tabla se van asignando nombres a las variables HMI y éstas se conectan a la variable de PLC a la que se corresponde especificando la conexión.

En el caso de la tabla inferior se pueden ver solo las variables de la primera estación y del AMR, puesto que, las variables de la segunda estación son las mismas, pero tienen diferente conexión.

Tabla XX: Fragmento de la Tabla de las variables HMI

Nombre	Conexión	Variable PLC	Tipo de Dato
New service_E1	HMI_Conexión_E1	kr3_r540_NewService	Bool
ServiceType_E1	HMI_Conexión_E1	kr3_r540_ServiceType	Byte
Item_E1	HMI_Conexión_E1	kr3_r540_NItems	Byte
PM_E1	HMI_Conexión_E1	PM	Bool
PP_E1	HMI_Conexión_E1	PP	Bool
ODK load_E1	HMI_Conexión_E1	ODK_LOAD	Bool
ODK_Conectado_E1	HMI_Conexión_E1	ODK_Conectado	Bool
ItemStarted_E1	HMI_Conexión_E1	ItemStarted	Bool
Item_Completed_E1	HMI_Conexión_E1	Item_Completed	Bool

Item_Completed_Number_E1	HMI_Conexión_E1	Item_Completed_Number	USInt
ServiceStarted_E1	HMI_Conexión_E1	ServiceStarted	Bool
Service_Completed_E1	HMI_Conexión_E1	Service_Completed	Bool
Read_File_E1	HMI_Conexión_E1	Marca_Read_File	Bool
Write_File_E1	HMI_Conexión_E1	Marca_Write_File	Bool
ODK_UNLOAD_E1	HMI_Conexión_E1	ODK_UNLOAD	Bool
Alarmas(1)	HMI_Conexión_E1	<No Value>	UInt
BorrarError	HMI_Conexión_E1	BorrarError	Bool
Read_E1	HMI_Conexión_E1	Read	Bool
Nuevo caso_E1	HMI_Conexión_E1	"Nuevo caso"	Bool
Kill_E1	HMI_Conexión_E1	Kill	Bool
TCP/IP_Start_E1	HMI_Conexión_E1	"TCP/IP_Start_E1"	Bool
TCP/IP_End_E1	HMI_Conexión_E1	"TCP/IP_End_E1"	Bool
TSEND_C_DB.CONT_E1	HMI_Conexión_E1	TRCV_C_DB.CONT	Bool
Target index	HMI_Conexión_AGV	"Target index"	Int
Bajar_End	HMI_Conexión_AGV	Bajar_End	Bool
Subir_End	HMI_Conexión_AGV	Subir_End	Bool
OnTarget	HMI_Conexión_AGV	Marca_OnTarget	Bool
CurrentTarget1	HMI_Conexión_AGV	Marca_CurrentTarget1	Int
Subir_HMI	HMI_Conexión_AGV	Subir_HMI	Bool
Bajar_HMI	HMI_Conexión_AGV	Bajar_HMI	Bool
Start_AGV	HMI_Conexión_AGV	"Start AGV"	Bool
Bajar_Start	HMI_Conexión_AGV	Bajar_Start	Bool
TCP/IP_Start_AGV	HMI_Conexión_AGV	"TCP/IP_Start_AGV"	Bool
TCP/IP_End_AGV	HMI_Conexión_AGV	"TCP/IP_End_AGV"	Bool
TSEND_C_DB.CONT_AGV	HMI_Conexión_AGV	TSEND_C_DB.CONT	Bool

Después de rellenar la tabla de variables HMI, se realiza el diseño de la interfaz (véase Figura 69).

En primer lugar, se pueden ver los pulsadores de “Marcha” y “Paro” y un pulsador denominado “Generate”, con el cual se controla la generación de un nuevo pallet en el almacén inicial.

En el apartado de “Comunicaciones” se realiza la monitorización del estado de los dos tipos de comunicaciones que se emplean en el proyecto: ODK y TCP/IP. Mediante los pulsadores “Load” y “UnLoad” se conecta y desconecta la comunicación del sistema ODK. Los indicadores indican si el sistema está conectado o desconectado. En el subapartado de fichero se muestra si se está realizando la lectura o escritura de los ficheros de texto. En el subapartado TCP/IP, el pulsador “Connect” sirve para inicializar la conexión de la comunicación TCP/IP. Con las luces en verde se indica si la conexión se ha realizado en cada uno de los PLCs.

En el apartado de “Estación 1” y “Estación 2”, en la primera columna aparece el servicio solicitado y se indica con luces verdes si éste ha empezado o ya ha sido completado. En la segunda columna se puede ver el número de ítems a montar solicitado y si un ítem se ha empezado a montar o si ya está completado. Además, se visualiza el valor del contador que indica cuantos ítems se han completado.

En el apartado del AMR se muestra el estado de la plataforma si está “Arriba” o “Abajo” y mediante los pulsadores “Subir” y “Bajar” se puede mover la plataforma manualmente. Asimismo, se muestra si el AMR está en algún objetivo y el índice de éste. Para saber dónde se encuentra el AMR se han puesto luces que indican si el AMR está en el almacén de entrada “Inicio”, “Estación 1”, “Estación 2” y en el “Almacén” de salida. Al mismo tiempo hay unos pulsadores para llevar al AMR a esos objetivos.

Mediante el pulsador “Errores” se abre una segunda pantalla donde aparece si ha ocurrido algún error. Finalmente, con el pulsador “Cerrar” se cierra la ejecución en WinCC Advanced de la interfaz HMI.

COMUNICACIONES

☒ **Marcha**

☐ **Paro**

☐ **Generate**

**ODK**

☒ Conectado

☐ Desconectado

**Fichero**

☒ Lectura

☐ Escritura

**TCP/IP**

☒ Estación 1

☒ Estación 2

☒ AGV

ESTACIÓN 1

2. Servicio      2 items

☐ Empezado      ☐ Empezado

☐ Completado      ☐ Completado

N° de completados

ESTACIÓN 2

9. Servicio      2 items

☐ Empezado      ☐ Empezado

☐ Completado      ☐ Completado

N° de completados

AGV

☒ Arriba

☐ Abajo

☒ En objetivo

N° de objetivo

☒

☐

☐

☐

Figura 69: Interfaz HMI

### **2.1.4 Puesta en marcha virtual distribuida**

Una vez diseñado el programa de control, se procede a su validación mediante la puesta en marcha virtual distribuida con *Software in the Loop* (SiL). En esta validación se depura el programa de control desarrollado detectando errores y solucionándolos antes de la implementación real de la estación robotizada.

La puesta en marcha virtual distribuida se va a realizar mediante una conexión remota entre Tecnomatix PS y las instancias de los PLCs virtuales de PLCSIM Advanced donde un único Gemelo Digital es controlado por tres PLCs virtuales diferentes. Esta conexión se realiza entre cuatro PCs diferentes. Se le denomina PC Local al PC donde se ejecutará el Gemelo Digital en Tecnomatix PS junto con su programa de control en TIA Portal y el Runtime del HMI en SIMATIC WinCC Advanced. A los otros tres PCs se les denomina PCs Remotos, en estos se ejecutarán las instancias de los PLCs virtuales en PLCSIM Advanced V3 Upd 2.

El primer paso es conectar mediante cable Ethernet los segundos puertos de los PCs, para ello se utilizará un *switch*. Se utilizarán los segundos puertos debido a que los primeros puertos están conectados a la red de internet de la universidad. Es importante que los cuatro PCs estén en la misma subred y que la dirección IP no esté repetida, sino la conexión no se podrá realizar. Por ello la dirección IP del PC Local es la 192.168.1.50, y la de los PCs Remotos 192.168.1.20, 192.168.1.16 y 192.168.1.10 (véase Figura 70).

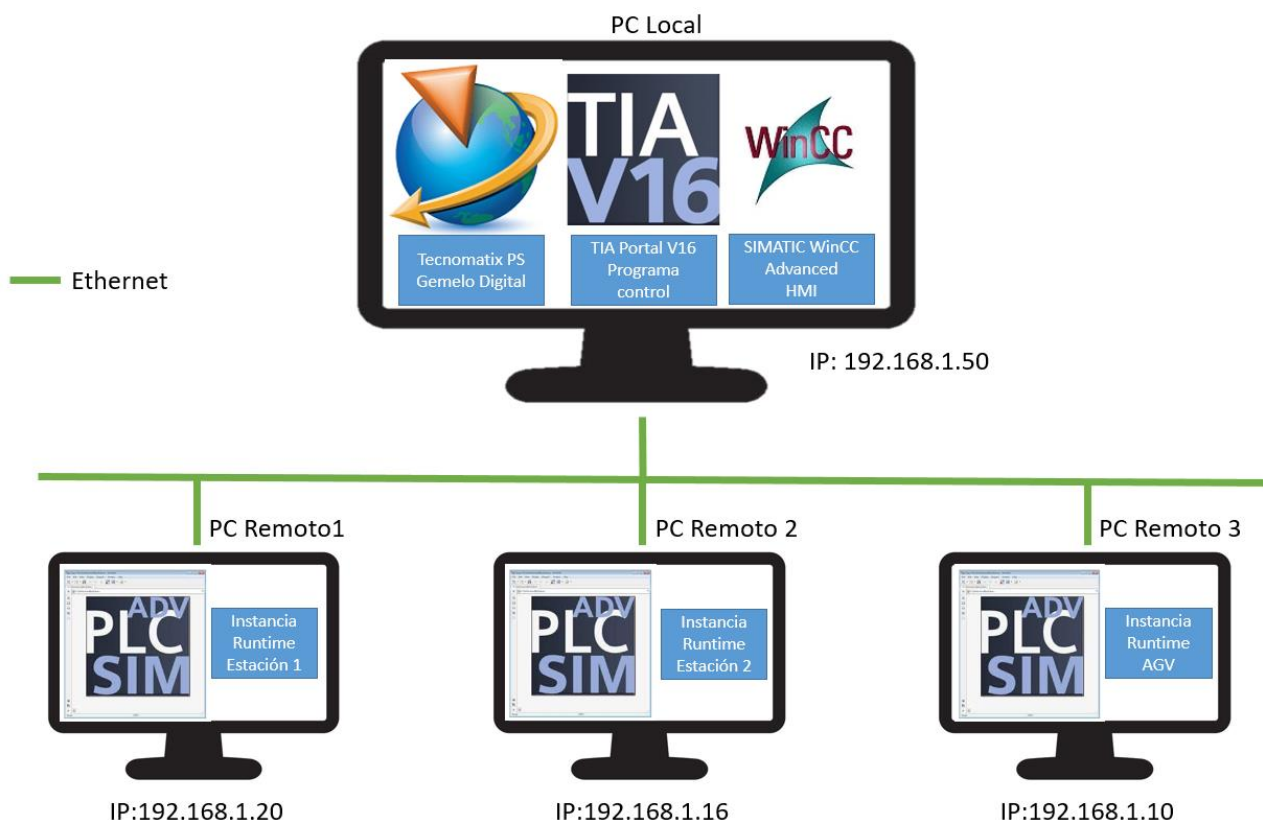


Figura 70: Comunicación distribuida vía Ethernet

En esta comunicación se realiza el intercambio de diferentes tipos de datos (véase Figura 71). En TIA Portal se carga y ejecuta el programa de control de los PLCs y se envía el resultado del estado de las variables de control a las instancias de los PLCs virtuales.

Al mismo tiempo, los PLCs virtuales envían el estado de las variables a Tecnomatix PS para la ejecución del Gemelo Digital. Mientras se ejecuta la simulación del Gemelo, Tecnomatix PS envía a los PLCs virtuales los datos del proceso obtenidos durante la simulación.

En la interfaz HMI se fuerzan y se visualizan las variables de los PLCs virtuales.

El programa de control de TIA Portal, por lo tanto, recibe el estado de las variables del proceso obtenidas por la simulación en Tecnomatix PS y el estado de las variables forzadas en la interfaz HMI. La recepción de esta información se hace por medio de las instancias de los PLCs virtuales.

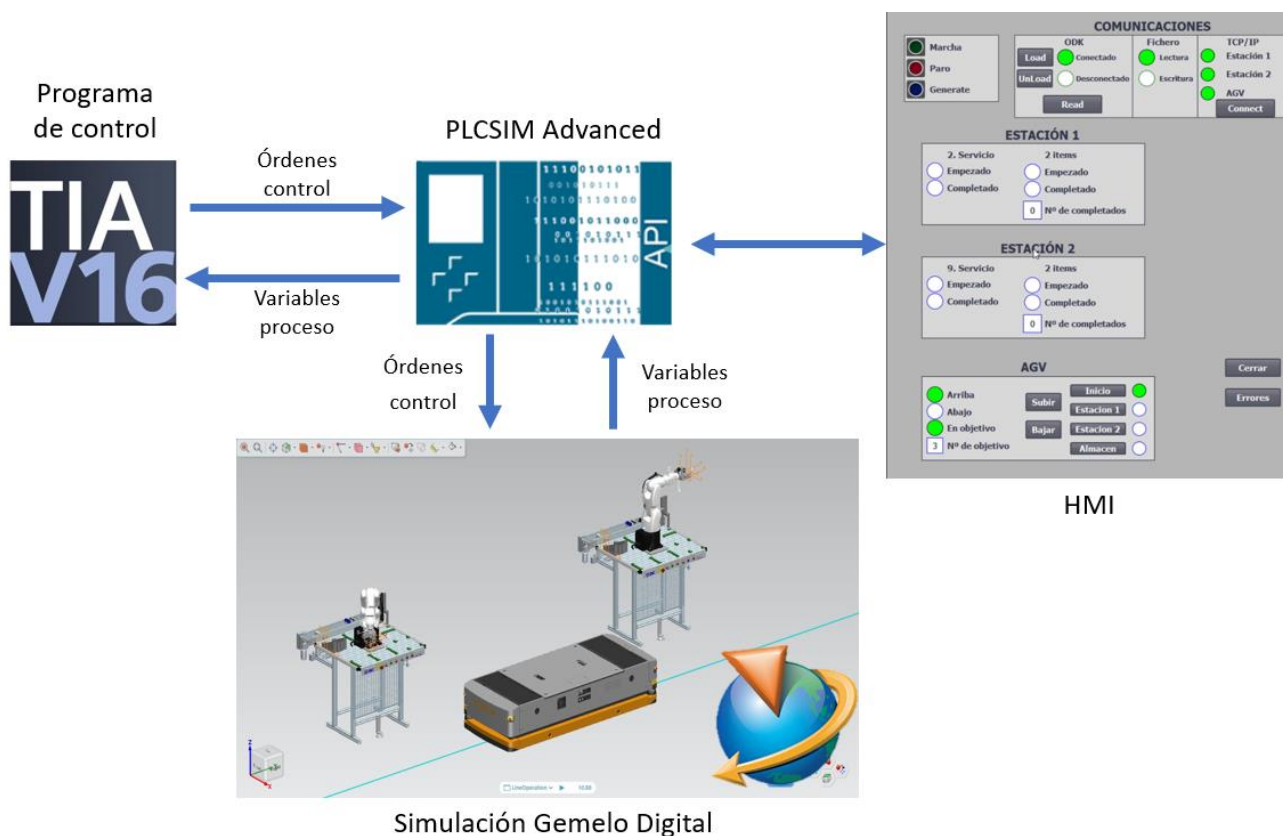


Figura 71: Transferencia de información entre las diferentes herramientas

Además, para hacer visibles las instancias de PLCSIM en la red y llegar a otros dispositivos, es necesario activar el “Siemens PLCSIM Virtual Switch” en la tarjeta que se utiliza para la comunicación (véase Figura 72). Esta configuración se realiza en el Panel de control de todos los PCs involucrados en la comunicación.

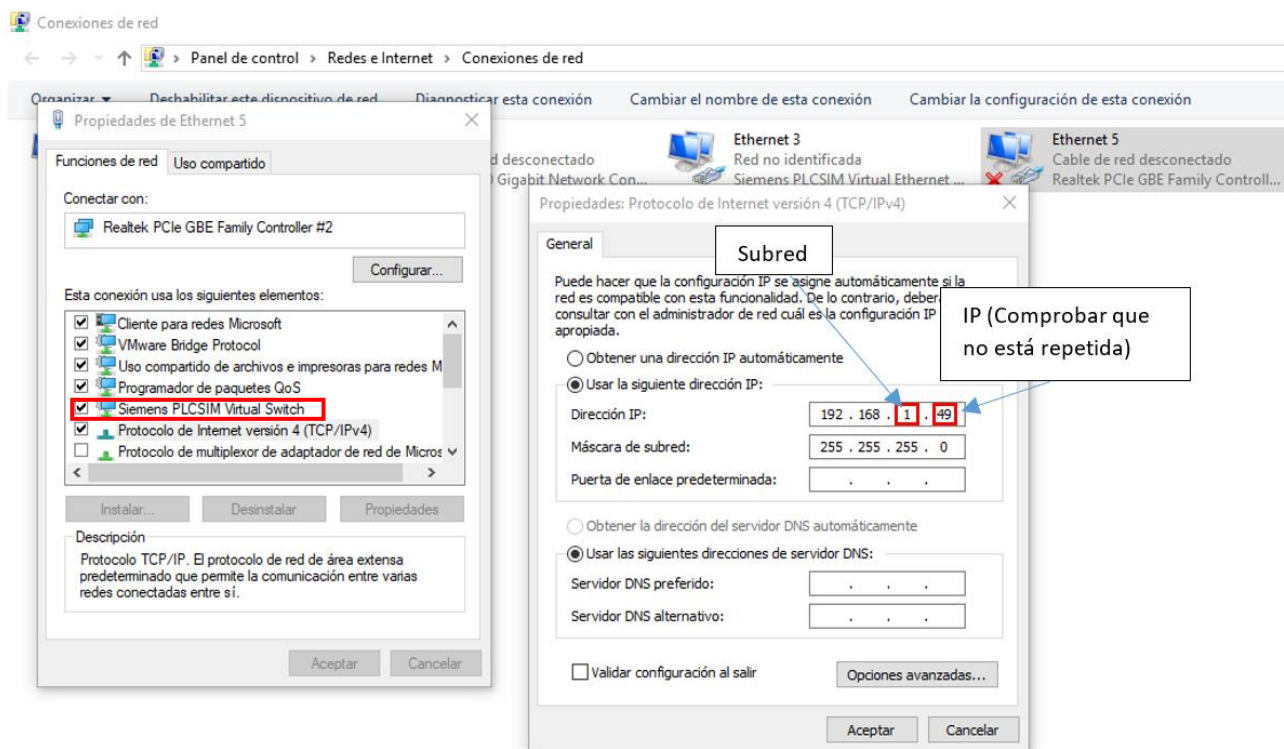


Figura 72: Propiedades Protocolo de Internet Versión 4 (TCP/IPv4)

Posteriormente, se crean las instancias de los PLCs virtuales en PLCSIM Advanced en los PCs Remotos. En cada PC remoto se ejecuta una instancia de los PLCs que controlan la primera estación, la segunda y el AMR.

En PLCSIM Advanced se active el “PLCSIM Virtual Eth. Adapter” y se elige la red de la tarjeta Ethernet conectada a los otros PCs. Se añaden las instancias anteriores, se especifica cada IP y la máscara subred (255.255.255.0). Estas IPs deben coincidir con las del programa de control de TIA Portal. Por ello, la IP instancia del PLC virtual de la primera estación es 192.168.0.120, de la segunda estación 192.168.0.121 y del AMR 192.168.0.122. Para que Tecnomatix pueda ver las instancias es necesario que se active el “*Runtime Manager Port*” (Puerto del administrador de tiempo de ejecución) y se le asigna un número de puerto en este caso es 50000 (véase Figura 73).

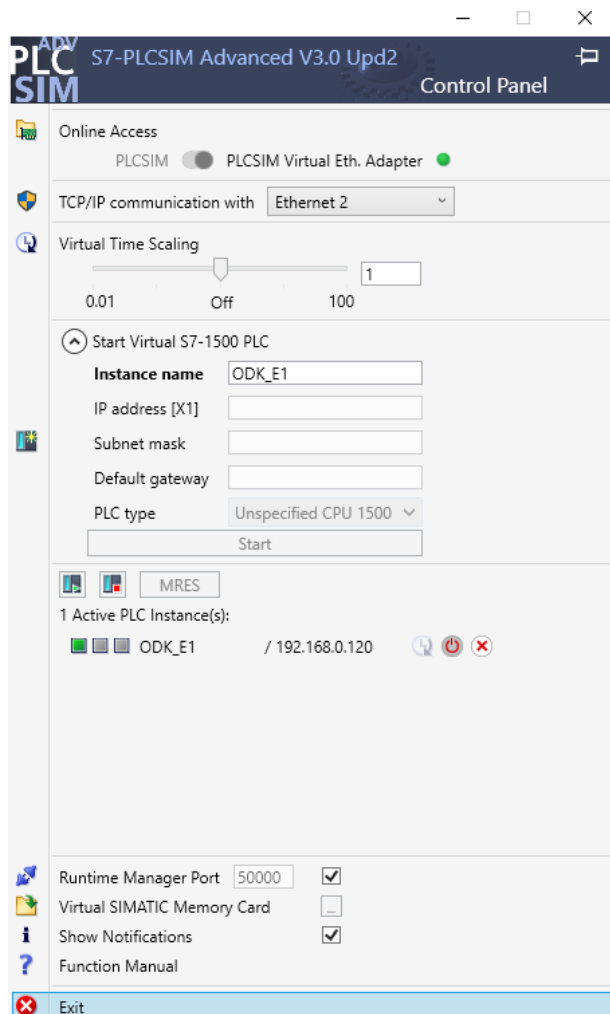


Figura 73: Instancia del PLC de la estación 1 en PLCSIM Advanced de un PC Remoto

Al mismo tiempo, como se ha dicho en anteriores ocasiones, las solicitudes de servicio vienen dadas por el sistema de multiagente para que se puedan tramitar estos pedidos, el archivo creado cuya extensión es \*.dll se ha de insertar dentro de la instancia de los PLCs de las estaciones. Con este fin, en la “Virtual SIMATIC Memory Card” y en cada una de las instancias de las estaciones se insertan los archivos \*.dll (véase Figura 74).

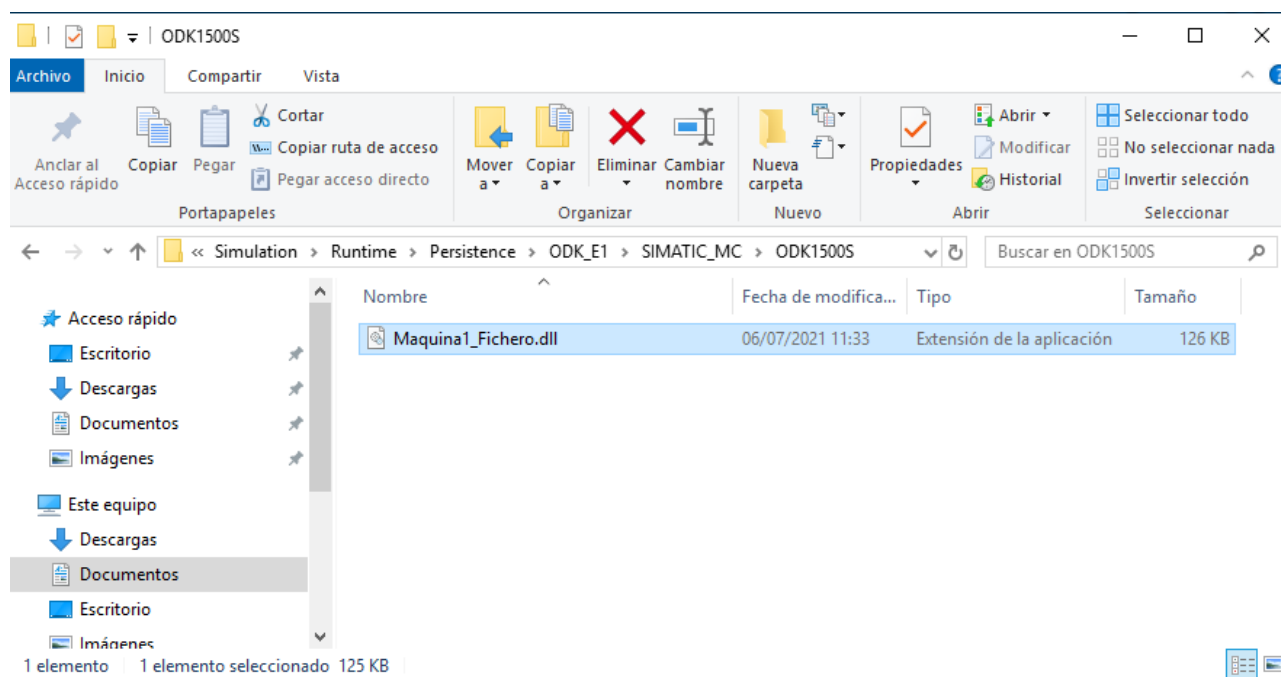


Figura 74: Ubicación archivo \*.dll en las instancias PLC

Sin embargo, para que Tecnomatix pueda encontrar las instancias de PLC es necesario que éstas estén en estado de “RUN”. Para ello, desde TIA Portal se realiza la carga avanzada y se selecciona como interfaz la tarjeta de red; en este caso, “Realtek PCIe GBE Family Controller”.

Finalmente, se realiza la conexión con Tecnomatix PS en el PC Local. A diferencia de la fase donde se realizaba la validación del Gemelo Digital (véase apartado 2.1.2.8. Validación del Gemelo Digital en Tecnomatix PS); en esta ocasión, la simulación se hace desde una conexión externa (véase Figura 75). Se añade una instancia remota de PLCSIM Advanced donde el *Host name* es el nombre de los PCs Remotos y el *Port* es el puerto indicado en PLCSIM Advanced *Runtime Manager Port* cuyo valor es 50000 (véase Figura 73). Después Tecnomatix ha de encontrar las instancias de los PLCs de los dos PCs Remotos y se le pone un nombre a cada instancia.

Por último, se elige el tipo de mapeo de señales. Existen dos tipos: *Server Address* (dirección del servidor) y *Signal Name* (nombre de señal). En este caso se elige el tipo *Server Address* ya que se le ha dado más importancia a que las direcciones de las señales sean iguales en Tecnomatix PS y en TIA Portal, a que tengan el mismo nombre. Debido a que el nombre de las variables de las estaciones son los mismos y, en consecuencia, es más fácil mapear de forma errónea las señales.

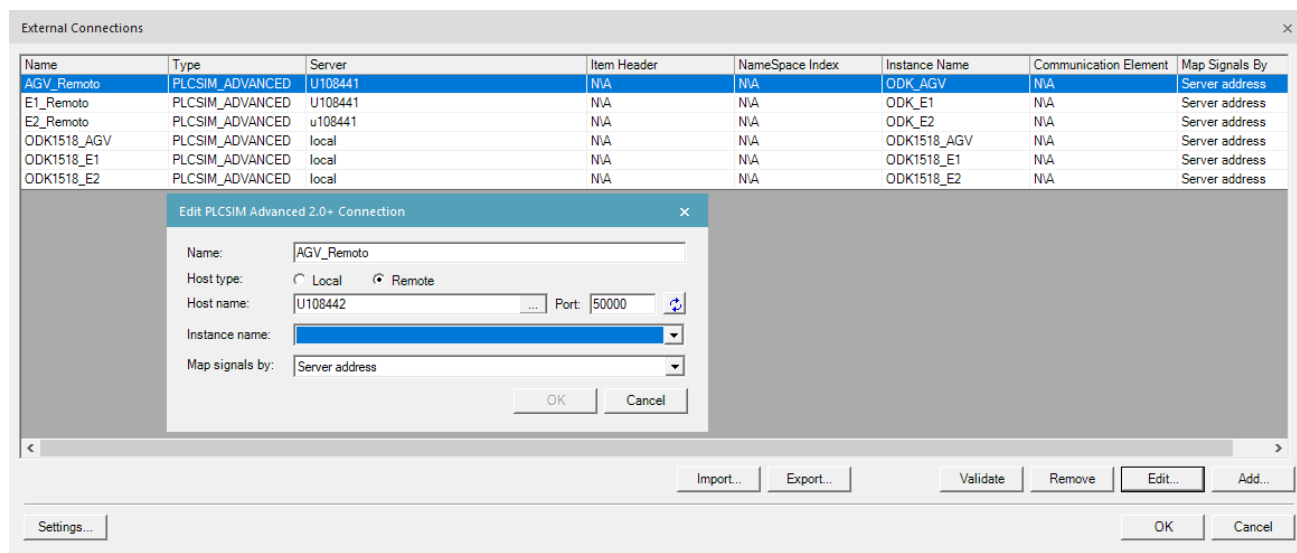


Figura 75: Conexión Tecnomatix PS con las instancias PLC de los PC Remotos

Una vez añadidas las instancias en Tecnomatix PS, se ha de especificar qué señal va con qué instancia de PLC; es decir, se especifica a cada PLC que parte del Gemelo tiene que controlar. Esto se realiza en el *Signal Viewer* (Visor de Señales), la columna de *PLC Connection* (Conexión PLC) se activa si las señales van a estar controladas mediante PLC y en la columna *External Connection* (Conexión externa) se asigna la señal con una instancia de PLC (véase Figura 76).

Signal Viewer									
Signal Name	Memory	Type	Robot Signal Name	Address	IEC Format	PLC Connect	External Connection	Resource	Comment
Sensor_TapaInt_1	<input type="checkbox"/>	BOOL		20.2	I20.2	<input checked="" type="checkbox"/>	E2_Remoto	● Sensor_TapaInt	
Sensor_TapaInt	<input type="checkbox"/>	BOOL		20.2	I20.2	<input checked="" type="checkbox"/>	E1_Remoto	● Sensor_TapaInt	
Sensor_Rodamiento_1	<input type="checkbox"/>	BOOL		20.1	I20.1	<input checked="" type="checkbox"/>	E2_Remoto	● Sensor_Rodamiento	
Sensor_Rodamiento	<input type="checkbox"/>	BOOL		20.1	I20.1	<input checked="" type="checkbox"/>	E1_Remoto	● Sensor_Rodamiento	
Sensor_BULON_1	<input type="checkbox"/>	BOOL		20.0	I20.0	<input checked="" type="checkbox"/>	E2_Remoto	● Sensor_BULON	
Sensor_BULON	<input type="checkbox"/>	BOOL		20.0	I20.0	<input checked="" type="checkbox"/>	E1_Remoto	● Sensor_BULON	
SAI6199-GV_E2_rntp_Bulon_CLOSED	<input type="checkbox"/>	BOOL		200.0	Q200.0	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_rntp_Rodamiento_CLOSED	<input type="checkbox"/>	BOOL		200.1	Q200.1	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_rntp_TapaExt_CLOSED	<input type="checkbox"/>	BOOL		200.2	Q200.2	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_rntp_HOME	<input type="checkbox"/>	BOOL		200.3	Q200.3	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_to_Bulon_CLOSED	<input type="checkbox"/>	BOOL		200.4	Q200.4	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_to_Rodamiento_CLOSED	<input type="checkbox"/>	BOOL		200.5	Q200.5	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_to_TapaExt_CLOSED	<input type="checkbox"/>	BOOL		200.6	Q200.6	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_to_OPEN	<input type="checkbox"/>	BOOL		200.7	Q200.7	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_E2_to_HOME	<input type="checkbox"/>	BOOL		201.0	Q201.0	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_at_Bulon_CLOSED_1	<input type="checkbox"/>	BOOL		200.0	I200.0	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_at_Rodamiento_HOLDING_1	<input type="checkbox"/>	BOOL		200.1	I200.1	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_at_TapaExt_HOLDING_1	<input type="checkbox"/>	BOOL		200.2	I200.2	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_at_HOME_1	<input type="checkbox"/>	BOOL		200.4	I200.4	<input checked="" type="checkbox"/>	E2_Remoto	● SAI6199-GV_E2	
SAI6199-GV_rntp_Bulon_CLOSED	<input type="checkbox"/>	BOOL		200.0	Q200.0	<input checked="" type="checkbox"/>	E1_Remoto	● SAI6199-GV_E1	
SAI6199-GV_rntp_Rodamiento_HOLD	<input type="checkbox"/>	BOOL		200.1	Q200.1	<input checked="" type="checkbox"/>	E1_Remoto	● SAI6199-GV_E1	
SAI6199-GV_rntp_TapaExt_HOLD	<input type="checkbox"/>	BOOL		200.2	Q200.2	<input checked="" type="checkbox"/>	E1_Remoto	● SAI6199-GV_E1	
SAI6199-GV_rntp_HOME	<input type="checkbox"/>	BOOL		200.3	Q200.3	<input checked="" type="checkbox"/>	E1_Remoto	● SAI6199-GV_E1	
SAI6199-GV_to_Bulon_CLOSED	<input type="checkbox"/>	BOOL		200.4	Q200.4	<input checked="" type="checkbox"/>	E1_Remoto	● SAI6199-GV_E1	

Figura 76: Signal Viewer

Finalmente, se inicia de la simulación del proceso de la célula robotizada. La visualización de las variables del Gemelo Digital se puede observar en el *Simulation Panel* (Panel de Simulación) que ofrece Tecnomatix PS (véase Figura 77).

Simulation Panel							
Simulation	Inputs	Outputs	...	Forced	Force...	Address	Robot Signal ...
ALCANCE_ROBOT_MESA							
AGV							
KMP1500-3-simply_stp_ID1	0			<input type="checkbox"/>	0	I	
KMP1500-3-simply_stp_O...	<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	I0.2	
KMP1500-3-simply_stp_C...	0			<input type="checkbox"/>	0	I1	
KMP1500-3-simply_stp_A...		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q	
KMP1500-3-simply_stp_T...		0		<input type="checkbox"/>	1	Q12	
KMP1500-3-simply_stp_S...		0.00		<input type="checkbox"/>	500.00	Q7	
Subir_E1_start		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q6.1	
Bajar_E1_start		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q6.2	
ESTACION 1							
ROBOT 1							
\$EXT_START		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q200...	\$EXT_START
kr3_r540_programNu...		0		<input type="checkbox"/>	1	Q110	programNumber
kr3_r540_at_HOME	<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	I100.0	kr3_r540_at_...
kr3_r540_ServiceType		0		<input type="checkbox"/>	0	Q2246	ServiceType
kr3_r540_NItems		0		<input type="checkbox"/>	0	Q2247	NItems
kr3_r540_Counter	0			<input type="checkbox"/>	0	I102	Counter
kr3_r540_NewService		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q100.1	NewService
\$DRIVES_ON		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q200...	\$DRIVES_ON
\$DRIVES_OFF		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q200...	\$DRIVES_OFF
\$CONF_MESS		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q200...	\$CONF_MESS
\$MOVE_ENABLE		<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Q200...	\$MOVE_ENA...
Service_Completed	<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	I2247.1	Service_Com...

Figura 77: Simulation Panel durante la simulación del Gemelo

## 2.1.5 Configuración Hardware del robot

Para realizar la puesta en marcha real de la estación robotizada, es necesario realizar antes la conexión entre los distintos dispositivos hardware que conforman la estación y la configuración de las señales mediante la herramienta KUKA.WorkVisual que se transfieren de un dispositivo a otro.

### 2.1.5.1 Conexiones

En este apartado se describen las conexiones realizadas entre los equipos que conforman la estación: la unidad de control, el robot KR 3 R540, la unidad manual de programación smartPAD, el dispositivo esclavo EtherCAT y el PLC.

A continuación, se observa el panel de conexiones de la unidad de control KR C4 compact (véase Figura 78). En la interfaz de seguridad X11, se conecta el pulsador de parada de emergencia, indispensable para brindar la seguridad necesaria tanto a los elementos de la estación como para el operario/a. En la X19, se realiza la conexión entre la unidad de control y el smartPAD el cual es el dispositivo de mando manual KUKA. En la extensión Interface X65 se conecta el dispositivo esclavo

EtherCAT Coupler EK1100. Asimismo, este dispositivo está conectado a módulos E/S digitales para el control de la apertura/cierre de las pinzas de la herramienta del robot mediante electroválvulas neumáticas y los sensores de éstas. Mediante el conector X66 se realiza la conexión entre la unidad de control y el PLC vía Profinet y el PC donde se encuentra KUKA.WorkVisual donde se hace la configuración hardware de las entradas y salidas digitales.

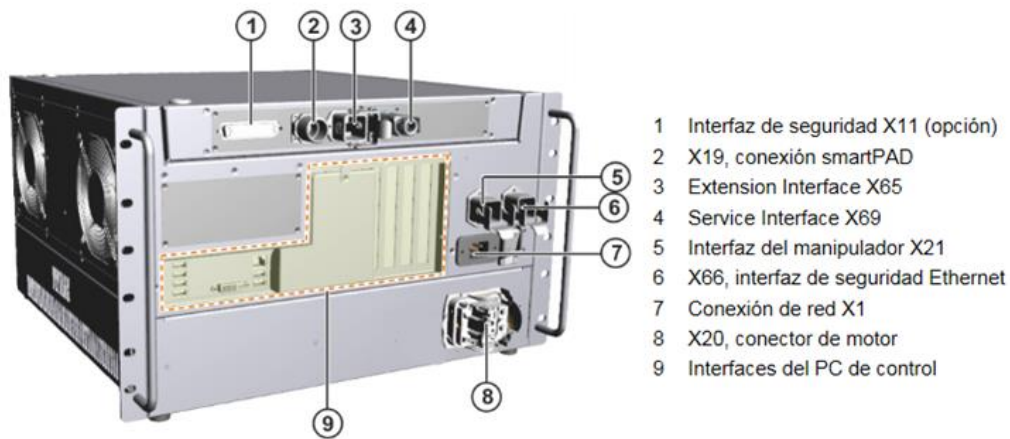


Figura 78: Interfaces KR C4 compact

Por otro lado, está la interfaz de conexiones del robot KUKA KR3 Agilus. Las conexiones de la interfaz A1 (véase Figura 79, izq.) son del cable motor (XM1/XM2-6) al conector de motor X20 de la unidad de control y del cable de datos (X15/X18) a la interfaz del manipulador X21 de la unidad de control. Las conexiones en el punto de conexión del cliente A1 (véase Figura 79, dcha.) se utilizan como la toma de aire de las electroválvulas de las pinzas de la herramienta del robot.

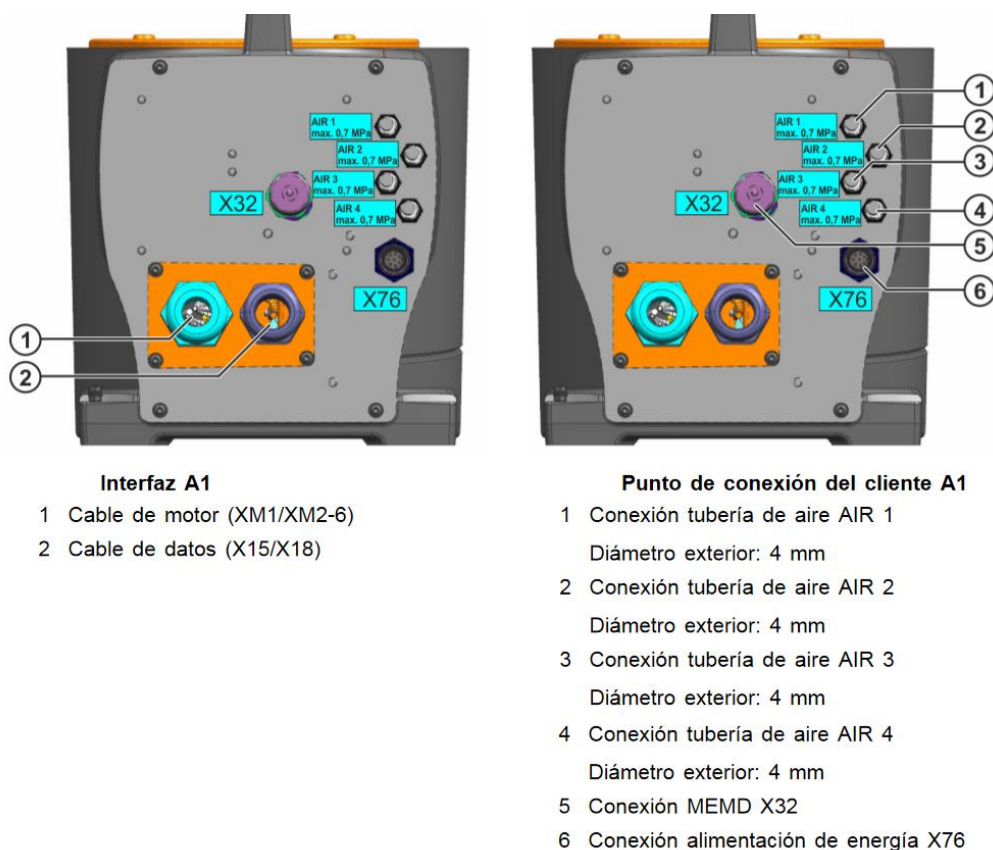
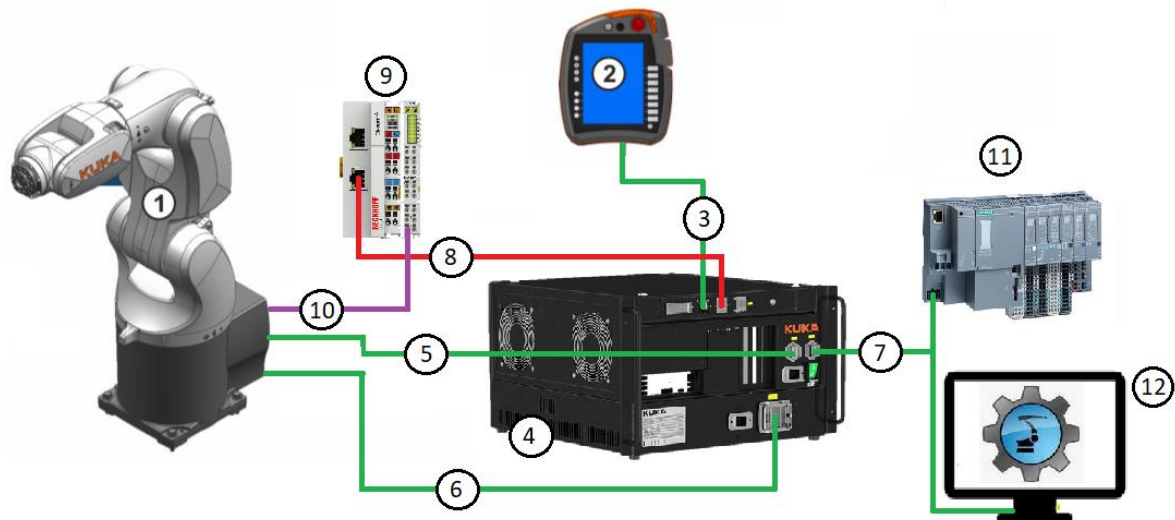


Figura 79: Interface de conexiones del robot KR3 Agilus

Por lo tanto, las conexiones entre todos los dispositivos de la estación, quedan de la siguiente forma (véase Figura 80).



- 1 Robot
- 2 Unidad manual de programación smartPAD
- 3 Cable de conexión/smartPAD
- 4 Unidad de control del robot
- 5 Cable de conexión/cable de datos
- 6 Cable de unión/cable de motor
- 7 Cable de conexión con SoftPLC y PC
- 8 Cable de conexión con dispositivo EtherCAT
- 9 Dispositivo EtherCAT
- 10 Cable de conexión canales E/S
- 11 ET 200SP Open Controller
- 12 PC con KUKA.WorkVisual

Figura 80: Conexiones del sistema de robot

### 2.1.5.2 Configuración de señales del robot

Para realizar la configuración de las señales, se utiliza la herramienta de ingeniería KUKA.WorkVisual. Esta herramienta está ubicada en el PC que está conectado a la unidad de control (véase Figura 8, 7).

Una vez abierto KUKA.WorkVisual se detecta la unidad de control del robot y ésta se establece como “unidad de control activa”, así se puede realizar la configuración hardware para el intercambio de señales. Por ello, se tienen que configurar las señales del robot (véase Figura 81), las del dispositivo EtherCAT y las señales vía Profinet entre la unidad de control y el PLC.

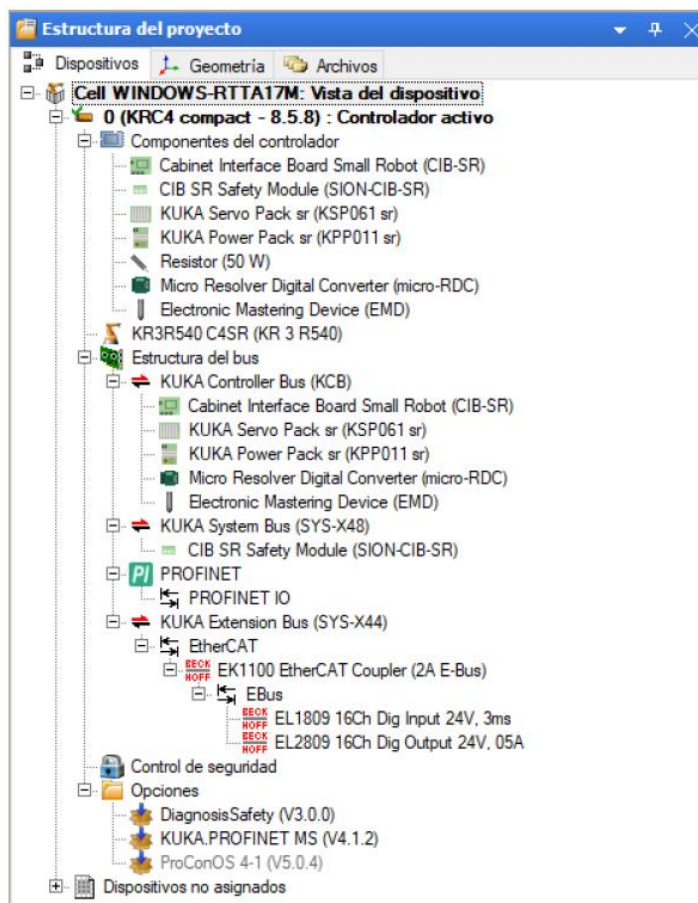


Figura 81: Estructura del proyecto de KUKA.WorkVisual

Después se realiza el mapeado de señales. Para ello, en KUKA.WorkVisual se abre la ventana “Circuito EA” (véase Figura 82). En las dos ventanas superiores se eligen los dispositivos entre los cuales se desea hacer el mapeado. En las ventanas inferiores se observan todas las señales de cada dispositivo y en la ventana del medio las señales ya mapeadas entre los dispositivos seleccionados.

Para mapear las señales de la unidad de control con las de EtherCAT, en las pestañas superiores se eligen por un lado las entradas digitales de la unidad de control (KR C E/S) y, por otro lado, las señales de los 16 canales de entradas digitales de EtherCAT del módulo EL1809 (véase Figura 82, ventana superior). Como se puede ver en la ventana del medio (véase Figura 82, ventana intermedia), se han conectado las primeras 16 señales de entradas digitales de la unidad de control (\$IN[1]...\$IN[16]) con los 16 canales de entradas digitales del dispositivo EtherCAT. De la misma forma, se han conectado las 16 primeras salidas digitales de la unidad de control (\$OUT[1]...\$OUT[16]) con los 16 canales de salidas digitales del módulo EtherCAT EL2809.

En este proyecto, se utilizan las salidas digitales \$OUT[1] y \$OUT[2] que sirven para controlar la apertura y el cierre de las pinzas de la herramienta del robot. Cuando están activas las pinzas se

cierran y cuando se desactivan las pinzas se abren. Las entradas digitales \$IN[1], \$IN[2], \$IN[3] y \$IN[4] son los sensores de las pinzas. Cada pinza tiene dos sensores de posición que indican que la pinza está abierta o cerrada.

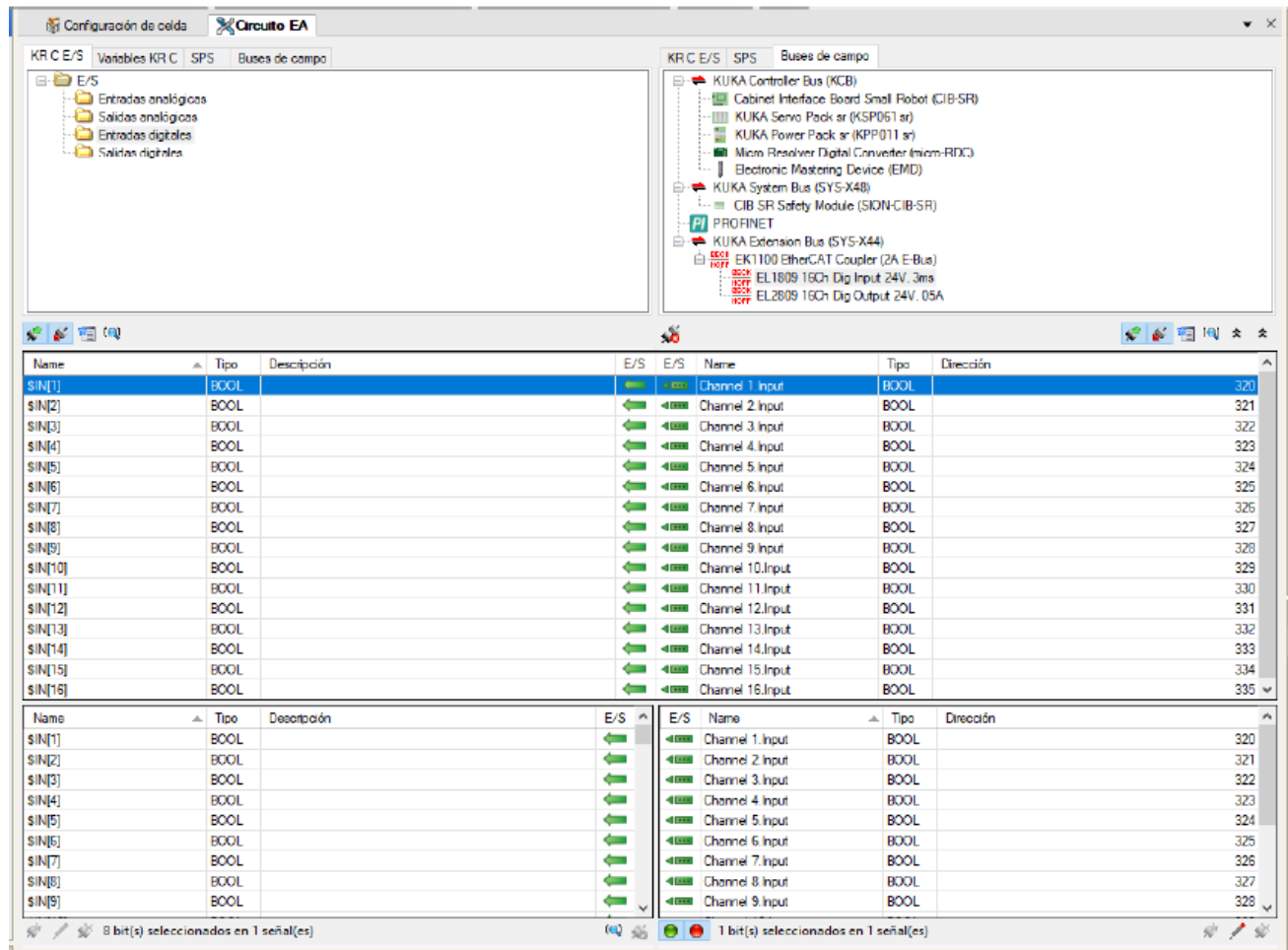


Figura 82: Ventana Circuito EA para el mapeo entre señales

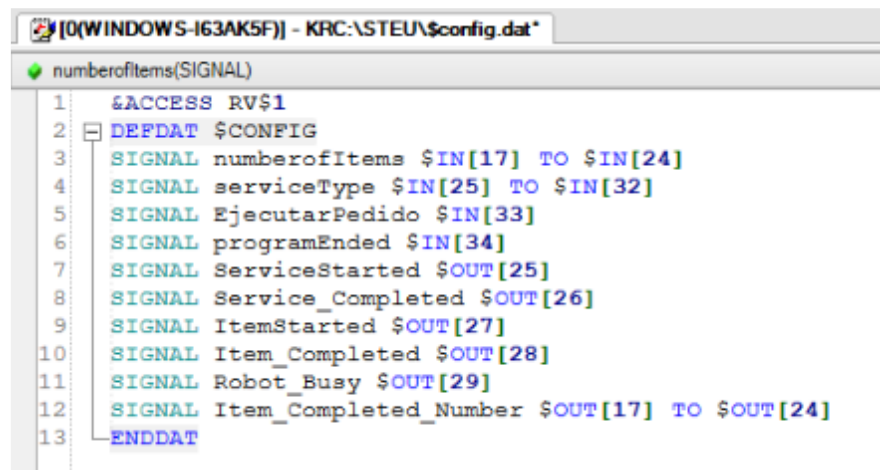
Por otro lado, en los programas de control del robot, se utilizan ciertas variables que han de ser definidas en la unidad de control del robot (véase Tabla XXI). Estas variables se mapean del mismo modo que los canales del dispositivo EtherCAT. El robot tiene 4096 señales E/S digitales y conectan todas las señales a partir de la dirección 2049 con las señales E/S Profinet.

Tabla XXI: Señales de robot que se transfieren

Nombre de la Señal	Dirección Robot	Función	Tipo
\$EXT_START	\$IN[2059]	Arranque externo	BOOL
\$MOVE_ENABLE	\$IN[2060]	Marcha habilitada	BOOL
\$CONF_MESS	\$IN[2061]	Confirmación de mensajes	BOOL
\$DRIVES_OFF	\$IN[2062]	Accionamientos conectados	BOOL
\$DRIVES_ON	\$IN[2063]	Accionamientos desconectados	BOOL
numberOfItems	\$IN[17...25]	Número de ítems solicitado	BYTE
serviceType	\$IN[25...32]	Tipo de servicio solicitado para que ejecute el robot	BYTE
EjecutarPedido	\$IN[33]	Orden de que ha llegado una nueva solicitud de servicio	BOOL
ServiceStarted	\$OUT[25]	Servicio iniciado	BOOL
Service_Completed	\$OUT[26]	Servicio completado	BOOL
ItemStarted	\$OUT[27]	Ítem iniciado	BOOL
ItemCompleted	\$OUT[28]	Ítem completado	BOOL

El dispositivo Profinet tiene 2032 bits E/S; sin embargo, hay variables cuyo tipo es byte, por lo que se agrupan 8 bits en las señales Profinet para crear dichos bytes.

Además, a estas señales se les puede asignar nombres mediante el archivo denominado “\$config.dat” (véase Figura 83).



```

1  &ACCESS RV$1
2  DEFDAT $CONFIG
3  SIGNAL numberOfItems $IN[17] TO $IN[24]
4  SIGNAL serviceType $IN[25] TO $IN[32]
5  SIGNAL EjecutarPedido $IN[33]
6  SIGNAL programEnded $IN[34]
7  SIGNAL ServiceStarted $OUT[25]
8  SIGNAL Service_Completed $OUT[26]
9  SIGNAL ItemStarted $OUT[27]
10 SIGNAL Item_Completed $OUT[28]
11 SIGNAL Robot_Busy $OUT[29]
12 SIGNAL Item_Completed_Number $OUT[17] TO $OUT[24]
13 ENDDAT
  
```

Figura 83: Definición de los nombres de las variables

Después de definir las variables y su mapeado, se transfieren los cambios a la unidad de control del robot para poder ejecutar sus programas.

### 2.1.6 Programa de control para la puesta en marcha real

El programa de control para la puesta en marcha real está basado en el programa realizado para la puesta en marcha virtual de la célula. Sin embargo, debido a que físicamente solo se dispone de una estación, el control de la segunda estación y del AMR se han eliminado. Así como la comunicación TCP/IP y la función relacionada con la simulación del flujo de materiales de la célula en el Gemelo Digital denominada “Material Flow”. Mientras que los bloques y funciones para la ejecución del sistema ODK se han mantenido: “EscribirDatosAgente” y “GestionarPedido”, además del código de “ControlRobot” para mandar al robot las señales que lo accionan.

Igualmente, la configuración hardware del proyecto se ha modificado (véase Figura 84), ya que el PLC virtual utilizado era un 1518, no obstante, el PLC real que se utiliza es un PLC real ET200 SP Open Controller con dos módulos E/S digitales y analógicos. En la “Vista de Redes” se ha agregado como “dispositivo no agrupado” la unidad de control del robot KR C4 Compact añadiéndole un módulo de 2032 bits E/S digitales para el intercambio de datos con el PLC. Las señales de dicho módulo se han configurado para que empiecen a partir de la dirección 2000. Finalmente, se han configurado las direcciones IP de ambos dispositivos y la conexión Profinet entre ellos.

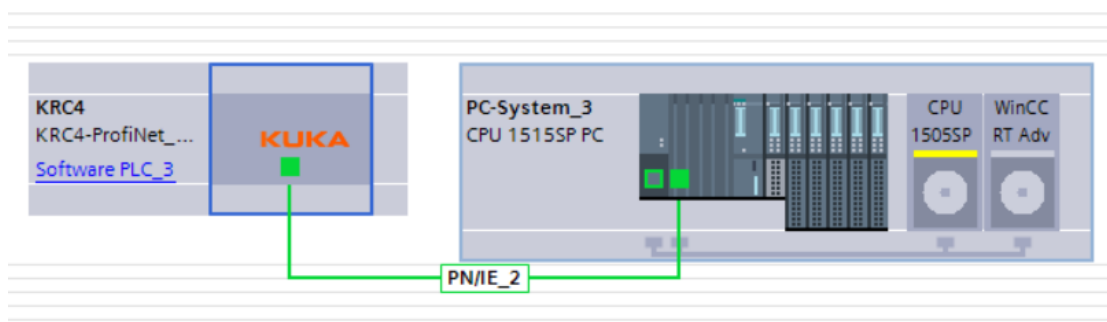


Figura 84: Vista de redes del control de la puesta en marcha real de una estación

### 2.1.7 Puesta en marcha real de una estación

Primero, los programas de robot desarrollados en Tecnomatix PS se han de cargar en el robot, de esta manera, Tecnomatix PS genera los programas de robot en el lenguaje propio de KUKA y se distinguen dos tipos de ficheros con distinta extensión: \*.src y \*.dat. En los ficheros \*.src se encuentran los comandos del código de los programas de robot y en el \*.dat se encuentran las coordenadas de los puntos de las trayectorias que ha de seguir el robot.

Ambos ficheros han de ser cargados en la unidad de control del robot. Para esto, se utilizará un USB con los ficheros y existen dos maneras de proceder. Una manera es insertando el USB

directamente en los puertos de la unidad de control y la otra es insertando el USB en el smartPAD. Se ha optado por insertarlo directamente en la unidad de control.

Seguidamente, se pueden observar los ficheros en la smartPAD. Para poder seleccionarlos, el usuario configurado en la smartPAD ha de ser “experto”. Para realizar diferentes pruebas de las operaciones, se pueden utilizar los diferentes modos de servicio del robot (véase Tabla XXII).

Tabla XXII: Modos de servicio del robot

Modo de servicio	Uso	Velocidades
T1	Para el modo de prueba, y programación por aprendizaje	<ul style="list-style-type: none"> <li>• Verificación del programa</li> </ul> Velocidad máxima de 250 mm/s <ul style="list-style-type: none"> <li>• Modo manual:</li> </ul> Velocidad de desplazamiento manual máximo 250 mm/s
T2	Para el modo de prueba	<ul style="list-style-type: none"> <li>• Verificación del programa</li> </ul> Velocidad programada <ul style="list-style-type: none"> <li>• Modo manual: No es posible</li> </ul>
AUT	Para robots industriales sin unidad de control superior	<ul style="list-style-type: none"> <li>• Funcionamiento del programa:</li> </ul> Velocidad programada <ul style="list-style-type: none"> <li>• Modo manual: No es posible</li> </ul>
AUT EXT	Para robots industriales con unidad de control superior, p. ej. Un PLC	<ul style="list-style-type: none"> <li>• Funcionamiento del programa:</li> </ul> Velocidad programada <ul style="list-style-type: none"> <li>• Modo manual: No es posible</li> </ul>

Mediante el modo de servicio T1 se verifican las operaciones del robot a velocidad muy reducida para asegurar que las trayectorias y los puntos de *Pick & Place* son correctos. En esta fase se han tenido que modificar algunos puntos que no coincidían con el escenario real.

En este proyecto, el robot industrial tiene un control superior a través del PLC real por lo que para que ese control pueda realizarse, se debe establecer el modo de servicio como AUT EXT (Automático Externo) y seleccionar el programa “main”. Al mismo tiempo ha de cargarse el programa de control modificado del apartado anterior sobre el PLC real y a continuación, se pone en marcha la estación robotizada y se ejecutan los servicios solicitados por el sistema multiagente.

## 2.2 Diagrama de Gantt/Cronograma

En este apartado se explica la planificación realizada para llevar a cabo el proyecto, incluyendo el desarrollo de la documentación que acompaña al proyecto. Por la presente se hace referencia a una breve descripción de todas las fases del proyecto, a la duración de cada uno de estas y a los recursos utilizados para su ejecución (véase Tabla XXIII).

El desarrollo del proyecto comenzó el lunes 23 de noviembre de 2020 y ha tenido una duración de 184 días, la jornada de trabajo es de 5 horas de lunes a viernes excluyendo los periodos de fiestas nacionales y vacaciones de Navidad y Semana Santa (véase Figura 85).

Tabla XXIII: Pasos necesarios para el desarrollo del proyecto

FORMACIÓN TECNOMATIX PS	
Duración	17 días
Fechas de inicio/fin	23/11/2020 – 17/12/2020
Descripción	En esta fase se ha realizado una formación sobre el funcionamiento de la herramienta de ingeniería Tecnomatix Process Simulate con la finalidad de coger suficiente autonomía para poder modelar el Gemelo Digital de la célula robotizada
Recursos utilizados	- Tecnomatix PS - Documentación facilitada por Siemens con ejemplos para desarrollar en Tecnomatix PS
GEMELO DIGITAL ESTACIÓN	
<b>Completar servicios estación</b>	
Duración	50 días
Fechas de inicio/fin	11/01/2021 – 19/03/2021
Descripción	En esta fase se completan los servicios que no han sido implementados anteriormente en el Gemelo Digital de la estación
Recursos utilizados	Tecnomatix PS
VALIDACIÓN GEMELO DIGITAL ESTACIÓN	
<b>Validación estación servicios completos</b>	
Duración	8 días
Fechas de inicio/fin	22/03/2021 – 31/03/2021

Descripción	Se valida el funcionamiento del Gemelo Digital de una de las estaciones
Recursos utilizados	Tecnomatix PS simulación mediante CEE
<b>GEMELO DIGITAL CÉLULA</b>	
<b>Duplicado de la estación</b>	
Duración	25 días
Fechas de inicio/fin	12/04/2021 – 14/05/2021
Descripción	Una vez programada una estación, ésta se duplica para formar la célula robotizada
Recursos utilizados	Tecnomatix PS
<b>Inserción del AMR</b>	
Duración	5 días
Fechas de inicio/fin	17/05/2021 – 21/05/2021
Descripción	Se añade el AMR a la célula y se programa la célula completa
Recursos utilizados	Soporte técnico KUKA Tecnomatix PS
<b>DISEÑO SOFTWARE DE CONTROL CÉLULA</b>	
<b>Programación control de la célula</b>	
Duración	20 días
Fechas de inicio/fin	24/05/2021 – 18/06/2021
Descripción	Diseño del código de control de la célula
Recursos utilizados	TIA Portal V16
<b>Diseño pantalla HMI</b>	
Duración	20 días
Fechas de inicio/fin	24/05/2021 – 18/06/2021
Descripción	Diseño y configuración de una pantalla HMI para la supervisión y control de la célula
Recursos utilizados	TIA Portal V16 WinCC Runtime
<b>PUESTA EN MARCHA VIRTUAL CÉLULA</b>	
<b>Mapeado señales</b>	
Duración	1 día
Fechas de inicio/fin	21/06/2021
Descripción	Mapeado de señales entre el Gemelo Digital y el programa de control
Recursos utilizados	TIA Portal V16 Tecnomatix PS

<b>Puesta en marcha virtual distribuida</b>	
Duración	10 días
Fechas de inicio/fin	21/06/2021 – 2/07/2021
Descripción	Validación del funcionamiento de la célula robotizada mediante SiL
Recursos utilizados	TIA Portal V16, WinCC Runtime PLCSIM Advanced V3 Upd 2 Tecnomatix PS
<b>PUESTA EN MARCHA REAL ESTACIÓN</b>	
<b>Mapeado señales</b>	
Duración	3 días
Fechas de inicio/fin	5/07/2021 – 7/07/2021
Descripción	Mapeado de las señales en la unidad de control del robot con las señales del PLC real y de los módulos E/S EtherCAT.
Recursos utilizados	KUKA.WorkVisual
<b>Modificación programas</b>	
Duración	5 días
Fechas de inicio/fin	5/07/2021 – 9/07/2021
Descripción	Se modifican los programas de control y robot para que se puedan ejecutar en la estación real
Recursos utilizados	TIA Portal V16 Tecnomatix PS
<b>Carga de programas</b>	
Duración	2 días
Fechas de inicio/fin	12/07/2021 – 13/07/2021
Descripción	Carga de programas de robot a la unidad de control del robot KR C4 Compact
Recursos utilizados	KUKA smartPAD USB Unidad de control KR C4 Compact
<b>Puesta en marcha real</b>	
Duración	3 días
Fechas de inicio/fin	14/07/2021 – 16/07/2021
Descripción	Se realiza la puesta en marcha real de la estación robotizada física y se realiza la grabación de algunos servicios para la defensa del proyecto

Recursos utilizados	TIA Portal V16 KUKA smartPAD Estación real
<b>DOCUMENTACIÓN</b>	
<b>Memoria</b>	
Duración	54 días
Fechas de inicio/fin	28/06/2021 – 09/09/2021
Descripción	Documentación escrita del desarrollo del proyecto
Recursos utilizados	Microsoft Word Power Point Diferentes fuentes de información

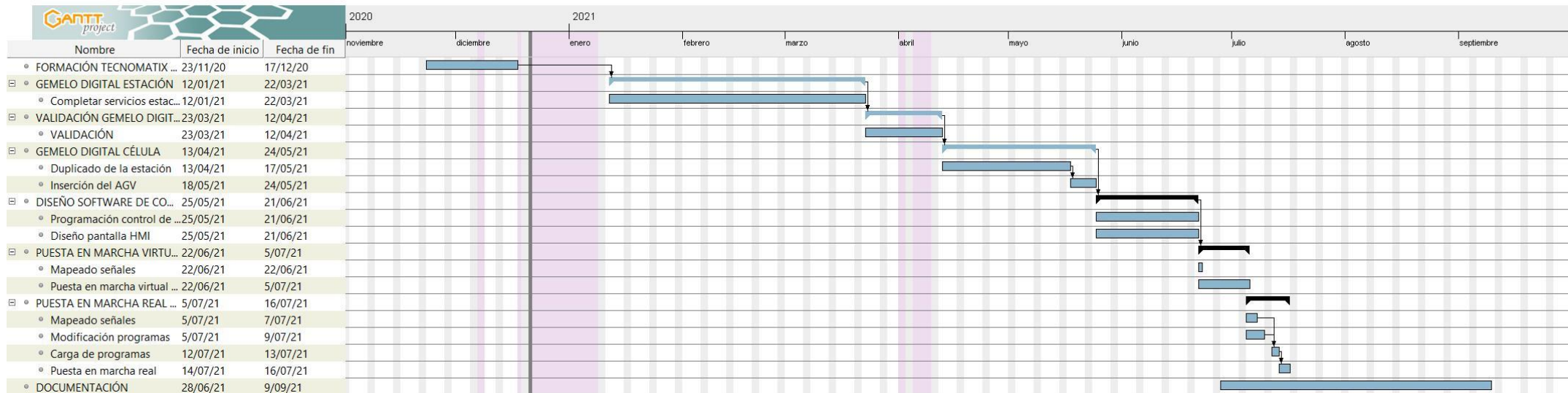


Figura 85: Diagrama de Gantt del proyecto

### 3 ASPECTOS ECONÓMICOS

#### 3.1 Descripción del presupuesto y/o del presupuesto ejecutado

En este apartado se define el presupuesto del proyecto. El presupuesto se divide en diferentes subapartados.

##### 3.1.1 Presupuesto de los elementos de la estación real

A continuación, se detalla el presupuesto de elementos utilizados en la estación robotizada real. Para ello, se presentan los precios de los elementos (véase Tabla XXIV) y después se amortizará sus valores calculando sus valores proporcionales al tiempo que han sido utilizadas.

Tabla XXIV: Precio de los elementos de la estación real

ELEMENTO	MODELO	FABRICANTE	PRECIO UNITARIO (€)	CANTIDAD	PRECIO TOTAL (€)
Robot Serie	AGILUS KR3 R540	KUKA	18.000	1	18.000
Unidad de control	KR C4 Compact			1	
Herramienta robot	Pinza MHK2-16D	SMC	1.200	1	1.200
	Pinza MHKL2-25D			1	
	Sensor inductivo			4	
	Electroválvula SY3100-5UD			2	
Nodo EtherCAT	EK1100 EtherCAT Coupler	Beckhoff	600	1	600
	Terminal EL2809 Digital Output			1	
	Terminal EL1809 Digital input			1	

PLC	SIMATIC ET 200SP	Siemens	1.800	1	1.800
Mesa de trabajo	-	SMC	1.500	1	1.500
PC sobremesa	Precision Tower	DELL	1.500	1	1.500
PCs simulación	Precision Compact	DELL	1.100	3	3.300

Una vez detallado los precios totales de los elementos de la estación se calcula la amortización de éstos (véase Tabla XXV). Se estima que la vida útil de los elementos de la estación; es decir, el robot y su controladora, la herramienta de robot, el nodo EtherCAT, PLC y la mesa de trabajo, es de cinco años. Por otro lado, la vida útil de los PCs del laboratorio se estima en tres años.

Tabla XXV: Presupuesto de los elementos de la estación real

ELEMENTO	PRECIO TOTAL (€)	VIDA ÚTIL (h)	AMORTIZACIÓN (€/h)	Nº HORAS USO (h)	COSTE TOTAL USO (€)
Robot Serie	18.000	43.800	0,41095	150	61,64
Unidad de control					
Herramienta robot	1.200	43.800	0,02739	150	4,11
Nodo EtherCAT	600	43.800	0,01369	150	2,05
PLC	1.800	43.800	0,04109	150	6,16
Mesa de trabajo	1.500	43.800	0,03424	150	5,14
PC sobremesa	1.500	26.280	0,05707	745	42,53
PCs simulación	3.300	26.280	0,12557	55	6,91

**Importe total, elementos de la estación**

**128,54 €**

### 3.1.2 Presupuesto de las herramientas de ingeniería

En este apartado se muestra el presupuesto de las herramientas de ingeniería. Donde se sigue el mismo procedimiento que con los elementos de la estación. Primero se define el precio de las herramientas (véase Tabla XXVI) y después, su automatización.

El paquete de TIA Portal tiene un coste por licencia fijo y la licencia es indefinida. En este proyecto se necesitan cuatro licencias una por cada PC utilizado en la puesta en marcha virtual.

Para el desarrollo del Gemelo Digital se utiliza herramienta de ingeniería Tecnomatix PS V16.0.1, para la cual se requiere una licencia fija de 8.000 € y una renovación anual de 1.000 €.

Tabla XXVI: Precios de las herramientas de ingeniería

TIPO DE HERRAMIENTA	MODELO	FABRICANTE	PRECIO UNITARIO (€)	CANTIDAD	PRECIO TOTAL (€)
Herramienta ingeniería para SW de control	TIA Portal V16	Siemens	1.200	4	4.800
	PLCSIM Advanced V3.0 Upd 2				
WinCC Advanced	SIMATIC WinCC Runtime Advanced	Siemens	500	1	500
Software Gemelo Digital	Tecnomatix Process Simulate V16.0.1	Siemens	8.000	1	8.000

La vida útil de todas las herramientas se ha establecido un año debido a que estas herramientas van actualizándose cada año.

Tabla XXVII: Presupuesto de las herramientas de ingeniería

TIPO DE HERRAMIENTA	PRECIO TOTAL (€)	VIDA ÚTIL (h)	AMORTIZACIÓN (€/h)	Nº HORAS USO (h)	COSTE TOTAL USO (€)
Herramienta ingeniería para SW de control	4.800	8.760	0,54794	320	175,34
WinCC Advanced	500	8.760	0,05707	255	14,55
Software Gemelo Digital	8.000	8.760	0,91324	780	712,32

<b>Importe total, herramientas de ingeniería</b>	<b>902,21 €</b>
--	-----------------

### 3.1.3 Grupo del proyecto

Para llevar a cabo el proyecto, hay que tener en cuenta que el grupo de trabajo está formado por diferentes participantes (véase Tabla XXVIII). Entre ellos, el desarrollador del proyecto; es decir, la alumna y el director del proyecto.

El desarrollo del proyecto incluye la creación de toda la documentación del proyecto. Para el desarrollo de todo el proyecto, tal y como se indica en el diagrama Gantt, han sido necesarios 184 días considerando que se ha trabajado durante 5 horas diarias.

Tabla XXVIII: Presupuesto grupo de trabajo

CONCEPTO	PRECIO DE HORA (€/h)	HORAS (h)	PRECIO TOTAL (€)
Desarrollo del proyecto	20	736	14.720
Director del proyecto	50	90	4.500

<b>Importe total, grupo de trabajo</b>	<b>19.220 €</b>
--	-----------------

### 3.1.4 Presupuesto total

Después de haber definido los presupuestos directos, para obtener cuál es el presupuesto total del proyecto, se realiza la suma de éstos y se suman los costes indirectos (véase Tabla XXIX). Estos costes corresponden a la luz, las comunicaciones, etc. que se estima que son un 10% del presupuesto total.

Tabla XXIX: Presupuesto total

Importe total, elementos de la estación	128,54 €
Importe total, herramientas de ingeniería	902,21 €
Importe total, grupo del proyecto	19.220 €
Subtotal	20.250,75 €
Costes indirectos (% 10)	2.025,07 €
<b>TOTAL</b>	<b>22.275,83 €</b>
<b><u>TOTAL</u></b>	<b>Veinte dos mil doscientos setenta y cinco euros con ochenta y tres céntimos</b>

## 4 CONCLUSIONES

Este Trabajo Fin de Máster es parte de un proyecto de investigación denominado “Arquitectura inteligente para la Industria 4.0: Realimentación a través del *Fog*”. En este trabajo se ha realizado el modelado del Gemelo Digital de una célula robotizada flexible y la puesta en marcha real del funcionamiento de las estaciones de la célula sobre una estación real.

En primer lugar, se ha modelado el Gemelo Digital de la célula robotizada flexible en Tecnomatix Process Simulate.

Mediante la herramienta de ingeniería de TIA Portal V16 se ha diseñado el programa de control de la célula. Para su validación y depuración, se ha realizado la puesta en marcha virtual con *Software in the Loop (SiL)* distribuido con cuatro PCs.

Finalmente, se ha llevado a cabo la puesta en marcha real de la estación real volcando los programas obtenidos y validados en simulación sobre los equipos reales.

Con esta metodología se ha conseguido disponer una célula virtual cuyas estaciones tienen la misma funcionalidad que la estación real para el demostrador del proyecto de investigación del departamento de Ingeniería de Sistemas y Automática.

Sin embargo, en un primer momento se pensó que la célula robotizada podría tener una línea de producción bidireccional y dependiendo del utillaje disponible en cada una de las estaciones, el sistema multiagente tendría que decidir si la producción empezaba por la primera o segunda estación. No obstante, debido a las limitaciones de la herramienta Tecnomatix PS, la línea de producción ha de ser unidireccional; es decir, el montaje de las piezas empieza siempre en la primera estación y termina en la segunda, no se puede hacer a la inversa. A causa de este impedimento, la célula robotizada pierde un grado de flexibilidad.

Otra de las dificultades encontradas desarrollando la documentación ha sido el planteamiento a seguir para explicar de manera adecuada y clara todos los aspectos que se han tenido que tener en cuenta para realizar el modelado del Gemelo Digital de la célula. Esto es debido a que Tecnomatix PS no es una herramienta sencilla puesto que, para completar cualquier operación hay que tener en cuenta varios factores y esto es complejo de reflejar sobre la documentación.

Como trabajo a futuro se ha pensado en la posibilidad de modelar un robot sobre el AMR para que fuera el propio AMR el que transportara el pallet a las estaciones. De esta forma, no haría falta una plataforma elevadora para que los robots de las estaciones tengan que coger los pallets. Esto

supondría un beneficio ya que, las operaciones del traslado de los pallets entre el AMR y la estación están cerca de los límites del alcance del robot.

## 5 BIBLIOGRAFÍA

En este apartado del documento se detallan los documentos, artículos, catálogos y páginas web con los que se ha elaborado el proyecto.

- [1] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, A.Y.C. Nee (2021). *Enabling technologies and tool for digital twin*.
- [2] Isidro Jesús Gonzalez-Hernandez, Berenice Armas-Álvarez, Melanie Coronel-Lazcano, Osmara Vergara-Martínez, Nereida Maldonado-López, Rafael Granillo-Macías (2021). *El desarrollo Tecnológico en las revoluciones industriales*.
- [3] B2A, INDUSTRIAL INTELLIGENCE. *Facilitas de Negocio*. Disponible en: <https://www.b2a.biz/es/industria-4-0/> [Fecha de consulta: 08/07/2021]
- [4] Arsys (12/03/2020). *¿Qué es el Fog Computing y que se diferencia del Edge Computing?*. Disponible en: <https://www.arsys.es/blog/fogcomputing/> [Fecha de consulta: 08/07/2021]
- [5] ITI, Investigate To Innovate. *DIGITAL TWINS: Gemelos Digitales en la transición a la Industria 4.0*. Disponible en: <https://www.iti.es/proyectosidi/proyecto-gemelos-digitales-industria-4-0/> [Fecha de consulta: 08/07/2021]
- [6] Avantek CT Solutions. *Virtual Commissioning para fabricantes de máquinas, ¿el fin de los prototipos?* Disponible en: <https://avantek.es/virtual-commissioning-para-fabricantes-de-maquinas-el-fin-de-los-prototipos/> [Fecha de consulta: 10/07/2021]
- [7] Avantek CT Solutions. *Tecnomatix Process Simulate*. Disponible en: <https://avantek.es/productos/tecnomatix/tecnomatix-process-simulate/> [Fecha de consulta: 10/07/2021]
- [8] Engusa. *Siemens Tecnomatix Process Simulate*. Disponible en: <https://www.engusa.com/es/product/siemens-tecnomatix-process-simulate> [Fecha de consulta: 10/07/2021]
- [9] KUKA. *KUKA.WorkVisual*. Disponible en: [https://www.KUKA.com/es-es/productos-servicios/sistemas-de-robot/software/software-de-sistema/KUKA\\_systemsoftware/KUKA\\_work-visual](https://www.KUKA.com/es-es/productos-servicios/sistemas-de-robot/software/software-de-sistema/KUKA_systemsoftware/KUKA_work-visual) [Fecha de consulta: 10/07/2021]
- [10] Siemens Product Support. *Liberación para la venta del ET 200SP Open Controller V2.0 estándar y con funcionalidad de seguridad ante fallos*. Disponible en: <https://support.industry.siemens.com/cs/document/109743492/liberaci%C3%B3n-para-la-venta-del-et-200sp-open-controller-v2-0-est%C3%A1ndar-y-con>

- [funcionalidad-de-seguridad-ante-fallos-?dti=0&lc=es-ES](#) [Fecha de consulta: 11/07/2021]
- [11] Support.industry.siemens.com (05/2021). *Manual Simatic S7-1500*.
- [12] Tecnopl.com. *Tia Portal: Utilidades del Software*. Disponible en: <https://www.tecnopl.com/tia-portal-utilidades-del-software/> [Fecha de consulta: 11/07/2021]
- [13] Ledin, Jim A. (02/1999). *Hardware-in-the-Loop Simulation*.
- [14] Engineering, Technology & Applied Science Research (10/2017). *Software In the Loop Simulation for Robot Manipulators*. Disponible en: <http://www.etasr.com/index.php/ETASR/article/view/1285> [Fecha de consulta: 11/07/2021]
- [15] Test and Measurement World. *Difference between Software in the loop testing and Hardware in the loop testing*. Disponible en: <https://www.test-and-measurement-world.com/Terminology/Software-in-the-loop-testing-Vs-Hardware-in-the-loop-testing.html> [Fecha de consulta: 11/07/2021]
- [16] KUKA. *KUKA.Sim*. Disponible en: <https://www.KUKA.com/es-mx/productos-servicios/sistemas-de-robot/software/planificaci%C3%B3n-proyecci%C3%B3n-servicio-seguridad/KUKA.-d-.sim> [Fecha de consulta: 15/07/2021]
- [17] RoboDK. *Simulate Robot Applications*. Disponible en: <https://robodk.com/> [Fecha de consulta: 15/07/2021]
- [18] KUKA. *KR 3 AGILUS*. Disponible en: <https://www.KUKA.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/kr-3-agilus> [Fecha de consulta: 18/07/2021]
- [19] KUKA (2016). *The control system of the future\_KR C4*.
- [20] SMC (2017). *Operation Manual. Solenoid Valve SY3000/5000/7000 Series*.
- [21] Siemens. *SIMATIC ODK 1500S – Ejemplos*. Disponible en: <https://support.industry.siemens.com/cs/document/106192387/simatic-odk-1500s-ejemplos?dti=0&lc=es-CO> [Fecha de consulta: 6/08/2021]
- [22] PAPELESdeInteligencia. *Los 9 pilares tecnológicos de la Industria 4.0 que todo CEO debe conocer*. Disponible en: <https://papelesdeinteligencia.com/pilares-tecnologicos-de-la-industria-4-0/> [Fecha de consulta: 6/08/2021]
- [23] Diccionario de la Real Academia de la Ingeniería. *Computación en la nube*.
- [24] Inser Robótica. *Las 7 ventajas de los sistemas AMR para transporte en tu empresa*. Disponible en: <https://www.inser-robotica.com/las-7-ventajas-de-los-sistemas-AMR-para-transporte-en-tu-empresa/> [Fecha de consulta: 10/07/2021]