

GRADO EN TECNOLOGÍA INDUSTRIAL
TRABAJO FIN DE GRADO

***MODELADO Y CONTROL EN SIMULACIÓN DE LA
ESTACION DE TAPAS DE LA CÉLULA FMS-200
PARA SIMULACIÓN DE FALLOS***

Alumno/Alumna: Artetxe Lázaro, Eneko
Director/Directora: Revillas Orive, Darío

Curso: 2018-2019

Fecha: Miércoles, 17/07/2019

Resumen

Los gemelos digitales son uno de los componentes de la llamada Industria 4.0, modelos que no son más que réplicas virtuales de procesos que simulan el comportamiento del sistema original. Este proyecto tiene como objetivo modelar un gemelo digital de la estación FMS-201, así como el análisis, simulación y control de los fallos inyectados en dicho modelo. Para ello, se expone en el documento la metodología utilizada tanto para la identificación y caracterización de fallos, como para su detección y resolución. Asimismo, se detalla la realización del gemelo digital, describiendo paso a paso el modelado y la lógica introducida. Por último, se demuestra que el proyecto cumple todos los objetivos establecidos, siendo posible su aplicación en la industria. La posible implementación de este tipo de tecnologías permitirá acortar los plazos de puesta en marcha, resultando así en una reducción de los costes asociados a esta.

Palabras clave: gemelo digital, puesta en marcha virtual, simulación, inyección de fallos, detección de fallos, control, modelado, NX-MCD.

Laburpena

Biki digitalak 4.0 Industria delakoaren osagarrietako bat dira, jatorrizko sistema baten portaera simulatzen duten prozesuen erreplika birtualak. Proiektu honen helburu nagusiak honakoak ziren: FMS-201 estazioaren biki digital baten eredu burutzea, bai eta modelo horretan injeztatutako akatsen analisia, simulazioa eta kontrola. Horretarako, idazkian adierazten da bai akatsen identifikazio eta ezaugarritzea zein berauen hautemate eta ebazpenerako jarraitutako metodologia. Halaber, biki digitalaren gauzapena zehaztatzen da, pausoz pauso deskribatuz bertan barruratutako eredu-egitea eta logika. Azkenik, baieztatzen da proiektuak hasieran ezarritako helburuak betetzen dituela, bere industriarako aplikazioa ahalbidetu lezakeelarik. Gainera, teknologia mota hauen balizko inplementatzeak abiarazpen-prozesuaren epeak laburtuko ditu; horrenbestez, berari lotutako kostuak murriztuz.

Hitz-gakoak: biki digitala, abiarazpen birtuala, simulazioa, akats-injezioa, akats-detekzioa, kontrola, eredu-egitea, NX-MCD.

Abstract

Digital twins are one of the components belonging to the so-called Industry 4.0, which are nothing but virtual replicas of processes that simulate the behaviour of the original system. The main objectives of this project were not only to model the digital twin of the FMS-201 station, but also to analyse, simulate and control the errors injected in it. In order to achieve this aim, the methodology used for the identification and characterization of the errors is presented in this report, as well as the one employed for the detection and resolution of such errors. Moreover, the execution of the digital twin itself is detailed, describing step by step the modelling and the logic introduced. Lastly, it is shown that the project achieves the objectives stated above, thus being allowed its feasible implementation in the industry. Therefore, the use of this technologies will permit to shorten the deadline of the launching process, turning out into a reduction of the associated costs.

Keywords: digital twin, virtual commissioning, simulation, error injection, error detection, control, modelling, NX-MCD.

ÍNDICE

ÍNDICE DE FIGURAS.....	5
ÍNDICE DE TABLAS	6
LISTA DE ACRÓNIMOS.....	6
1 MEMORIA	7
1.1 INTRODUCCIÓN	7
1.2 CONTEXTO	7
1.3 OBJETIVOS Y ALCANCE.....	10
1.4 BENEFICIOS QUE APORTA EL PROYECTO	11
1.4.1 BENEFICIOS TÉCNICOS.....	11
1.4.2 BENEFICIOS ECONÓMICOS	11
1.4.3 BENEFICIOS SOCIALES	12
1.5 ANÁLISIS DE ALTERNATIVAS.....	12
1.5.1 ALTERNATIVAS PARA SOFTWARE DE GAMALOS DIGITALES	12
1.5.1.1 SELECCIÓN DEL SOFTWARE DE DISEÑO MECATRÓNICO	13
1.5.2 ANÁLISIS PARA SIMULACIÓN.....	13
1.6 ANÁLISIS DE RIESGOS	14
2 METODOLOGÍA	16
2.1 DISEÑO DEL GEMELO DIGITAL EN SIEMENS NX-MCD.....	16
2.1.1 JUNTAS Y RESTRICCIONES.....	16
2.1.2 FÍSICAS BÁSICAS.....	17
2.1.3 SEÑALES, SENSORES Y ACTUADORES	18
2.1.3.1 ACTUADORES.....	18
2.1.3.2 SENSORES.....	20
2.2 PROYECTO DE CONTROL	22
2.2.1 MÓDULO MECATRÓNICO BOTONERA	23
2.2.2 MÓDULO MECATRÓNICO RACK.....	24
2.2.2.1 POSICIÓN INICIAL RACK	24
2.2.2.2 SECUENCIA PRINCIPAL.....	25
2.2.2.3 EXTRAER BASE.....	26
2.2.2.4 VERIFICAR POSICIÓN	26
2.2.2.5 INSERTAR EN PALLET	27
2.2.3 TRANSPORTE.....	28
2.2.3.1 POSICIÓN INICIAL TRANSPORTE.....	28
2.2.3.2 UBICAR PALLET	28

2.2.3.3 RETIRAR PALLET.....	29
2.2.2.4 ESTADO ESTACIÓN	29
2.2.2.5 MOVER CINTA.....	30
2.3 ANÁLISIS PRELIMINAR DE FALLOS.....	30
2.3.1 DETECCIÓN	31
2.3.2 DIAGNÓSTICO	33
2.3.3 ANÁLISIS	34
2.4 INYECCIÓN DE FALLOS	38
2.4.1 MODELADO.....	38
2.4.2 TRATAMIENTO DE SEÑALES EXISTENTES	39
2.5 ANÁLISIS DE FALLOS SELECCIONADOS	42
2.5.1 Fallo N°3	42
2.5.2 Fallo N°5	43
2.5.3 Fallo N°9	44
2.5.4 Fallo N°11	46
2.6 VERIFICACIÓN	48
2.7 DESCRIPCIÓN DE TAREAS Y DIAGRAMA DE GANTT.....	48
3 ASPECTOS ECONÓMICOS	51
3.1 PRESUPUESTO.....	51
4 CONCLUSIONES	53
4.1 LÍNEAS FUTURAS	53
5 BIBLIOGRAFÍA	54
ANEXO	55

ÍNDICE DE FIGURAS

Figura 1: Estación FMS-201.....	8
Figura 2: Esquema gráfico de la estación FMS-201.....	8
Figura 3: Diagrama de las áreas de la Industria 4.0.....	9
Figura 4: Esquema del funcionamiento del software in the loop (SIL) (derecha) y su representación real (izquierda).....	14
Figura 5: Modelo digital de la estación FMS-201.....	17
Figura 6: Tabla de variables de entrada y salida, y código de los cilindros.....	19
Figura 7: Modelado de ventosas.....	20
Figura 8: Tabla de variables de entrada y salida, y código de los selectores.....	21
Figura 9: Tabla de variables de entrada y salida, y código de las ventosas.....	21
Figura 10: Modelo digital de diferentes modelos mecatrónicos.....	22
Figura 11: Esquema con la jerarquía del proyecto de control.....	23
Figura 12: Tabla y diagrama de entradas y salidas y diagrama de módulo básico del FB1.....	23
Figura 13: modelo digital de los conjuntos del módulo mecatrónico Rack.....	24
Figura 14: Tabla y diagrama de entradas y salidas del FB2.....	24
Figura 15: Tabla y diagrama de entradas y salidas, y GRAFCET del FB3.....	25
Figura 16: Tabla y diagrama de entradas y salidas, y GRAFCET del FB4.....	26
Figura 17: Tabla y diagrama de entradas y salidas, y GRAFCET del FB5.....	26
Figura 18: Tabla y diagrama de entradas y salidas, y GRAFCET del FB6.....	27
Figura 19: Tabla y diagrama de entradas y salidas del FB7.....	28
Figura 20: Tabla y diagrama de entradas y salidas, y GRAFCET del FB8.....	28
Figura 21: Tabla y diagrama de entradas y salidas, y GRAFCET del FB9.....	29
Figura 22: Tabla y diagrama de entradas y salidas del FB10.....	29
Figura 23: Tabla y diagrama de entradas y salidas del FB11.....	30
Figura 24: Código de la detección correspondiente al método tiempo de vigilancia.....	31
Figura 25: Código de la detección correspondiente al método tiempo mínimo de etapa.....	32
Figura 26: Código de la detección correspondiente al método coincidencia simultánea.....	32
Figura 27: Código de la detección correspondiente al método detección mediante contador.....	33
Figura 28: GEMMA correspondiente a la secuencia diseñada.....	34
Figura 29: Tabla de variables de entrada/salida, código de lámpara indicadora de los interruptores.....	38
Figura 30: Modelo de los interruptores.....	39
Figura 31: Tabla de variables de entrada y salida, y código de sensor con fallo.....	39
Figura 32: Tabla de variables de entrada y salida, y código de ventosas con fallo.....	40
Figura 33: Tabla de variables de entrada y salida, código de los sensores de los cilindros con fallo.....	41
Figura 34: Tabla de variables de entrada y salida, y código de los cilindros con fallo.....	41
Figura 35: Modificación de código para dar solución a la avería 3.....	42
Figura 36: Modificación de código para dar solución a la avería 5.....	43
Figura 37: Modificación de código para dar solución a la avería 5.....	44
Figura 38: Modificación de código para dar solución a la avería 9.....	45
Figura 39: Modificación de código para dar solución a la avería 9.....	46
Figura 40: Modificación de código para dar solución a la avería 11.....	47
Figura 41: Diagrama de Gantt con las tareas, fechas y duración del proyecto.....	50
Figura 42: Desglose del presupuesto total indicado en porcentajes.....	52
Figura 43: Pantalla inicial del HMI.....	55
Figura 44: Interfaz de la detección y resolución de errores.....	56
Figura 45: Mando manual para fallos con paro de emergencia.....	57

ÍNDICE DE TABLAS

Tabla 1: <i>Puntuación de las diferentes alternativas para software de gemelos digitales.</i>	13
Tabla 2: <i>Matriz de Probabilidad-Impacto.</i>	15
Tabla 3: <i>Diagrama de Gantt con las tareas, fechas y duración del proyecto.</i>	49
Tabla 4: <i>Resumen de presupuesto.</i>	51

LISTA DE ACRÓNIMOS

- CAD:** Computer Aided Design (Diseño asistido por ordenador)
- CPU:** Central Processing Unit (Unidad central de procesado)
- DB:** Data Block (Bloque de datos)
- FB:** Function Block (Bloque de función)
- FC:** Function (Función)
- FMS:** Flexible Modular Station (Estación flexible modular)
- HIL:** Hardware In the Loop
- HMI:** Human Machine Interface (Interfaz humano-máquina)
- IEC:** International Electrotechnical Commission (Comisión internacional de electrotecnia)
- IO:** In & Out (Entradas y salidas)
- IP:** Internet Protocol (Protocolo de internet)
- MCD:** Mechatronic Concept Designer (Diseñador de conceptos mecatrónicos)
- PLC:** Programable Logic Controller (Controlador lógico programable)
- SIL:** Software In the Loop

1 MEMORIA

1.1 INTRODUCCIÓN

Los Gemelos Digitales se presentan como una tecnología disruptiva en la simulación y análisis de procesos industriales, capaces de aprovechar los beneficios de la transformación digital de planta, una transformación en la que ya se encuentran involucradas un número significativo de empresas y que se intensificará en los próximos años. Entre los numerosos usos que poseen los gemelos digitales, existe el del análisis de fallos, estudio que resulta muy importante a la hora de diseñar el control cualquier proceso industrial. De hecho, dicha implementación de fallos permite al usuario observar las consecuencias de los fallos contemplados y comprobar que las estrategias planeadas para su resolución son adecuadas.

Sin embargo, debido a la reciente aparición de esta tecnología, los límites del conocimiento acerca de las posibilidades de los gemelos digitales son aún difusos para los usuarios de este tipo de técnicas. Asimismo, el margen de mejora es amplio para las compañías que diseñan estas herramientas, lo cual facilita la emersión de simbiosis positivas entre universidad y empresa. Muestra de ello es la colaboración vigente entre la empresa Siemens y diferentes universidades, entre las que se encuentra la propia Escuela de Ingeniería de Bilbao (ETSI). Este tipo de sinergias posibilita que, por un lado, la universidad disponga de facilidades para la adquisición de equipos docentes y licencias; y, por otro, la empresa obtiene tanto profesionales cualificados adaptados a cada una de sus herramientas, como *feedback* para el desarrollo de sus productos.

En este marco, la creación de un modelo digital de la estación FMS-201 resulta de gran interés dada la utilidad que supone el hecho de que los fallos estén integrados en ella. En efecto, el uso de dichos fallos permite analizar el comportamiento de la máquina real en su gemelo digital. La mencionada estación FMS-201, es uno de los módulos de la estación didáctica modular FMS-200 de la empresa SMC, la cual está ubicada en el laboratorio de investigación del Departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingeniería de Bilbao (Figuras 1 y 2).

En este documento se expone el proceso seguido para la realización del gemelo digital de una estación, así como la solución dada para la inyección de fallos en dicho gemelo, y la detección y resolución de estos fallos mediante el código de control.

1.2 CONTEXTO

Dado el comienzo en la digitalización dentro de las fábricas, la combinación de tecnologías basadas en Internet y tecnologías prospectivas en el campo de los objetos “inteligentes” (máquinas y productos), se está llevando a cabo un cambio de paradigma fundamental de la producción industrial. La nueva visión en torno al futuro de la producción engloba sistemas de manufactura modulares y

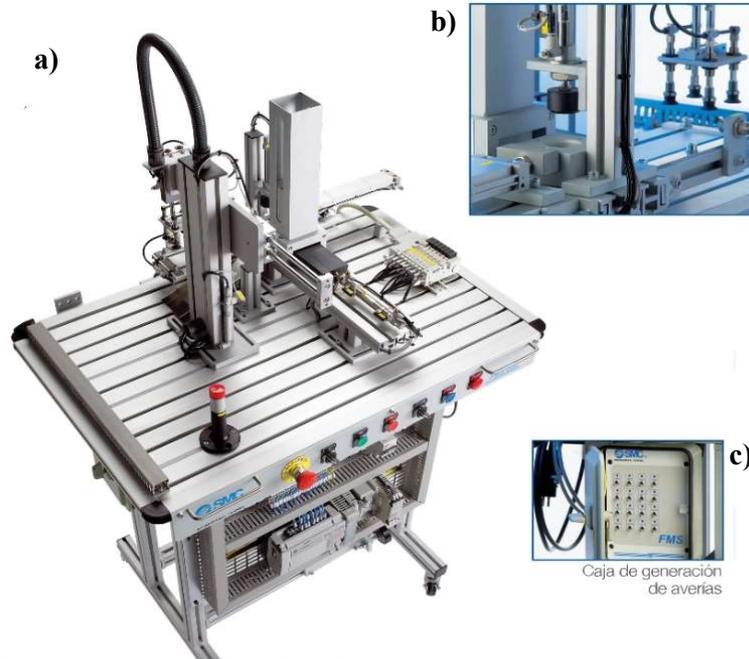


Figura 1: Estación FMS-201: a) Visión frontal de la estación completa, b) detalle de la parte posterior de la inserción en pallet y la verificación posición, c) caja de generación de averías.

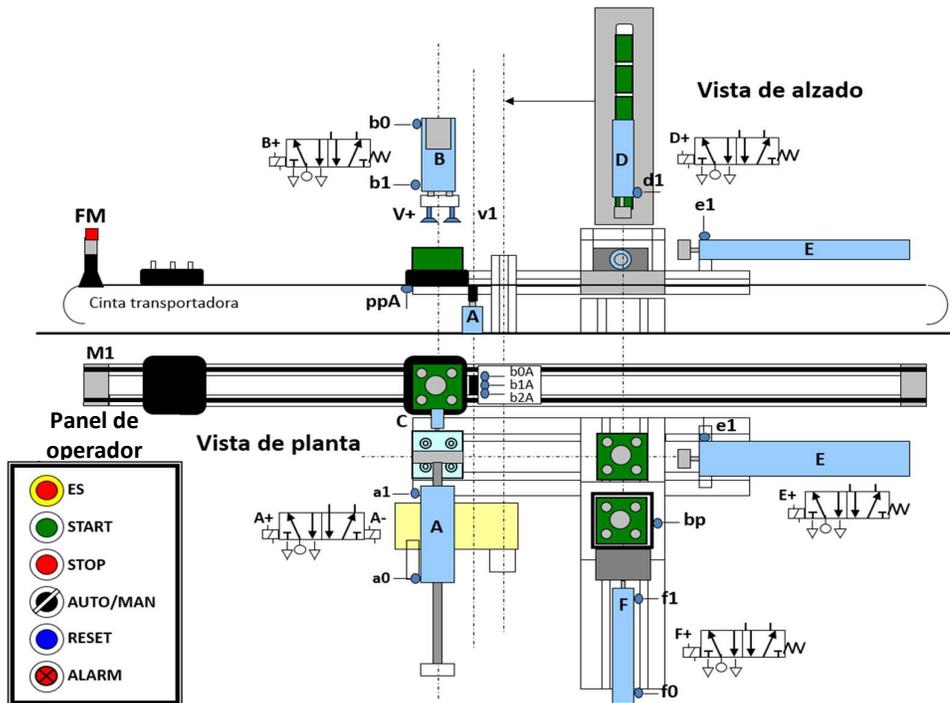


Figura 2: Esquema gráfico de la estación FMS-201. A: Cilindro horizontal de la inserción en pallet. B: Cilindro vertical de la inserción en pallet. C: Cilindro expulsor. D: Cilindro vertical de la verificación de posición. E: Cilindro horizontal de la verificación de posición. F: Cilindro de la extracción de base. FM: Barra luminosa. M1: Motor de accionamiento de cinta.

eficientes, y describe escenarios en los que los procesos de producción reciben un mayor control. Así, se pretende realizar la manufactura de productos individuales conocidos como *batch size of one* (lote de una unidad¹), pero manteniendo las ventajas económicas de la producción en masa. Debido a esta expectativa futura, el término “Industria 4.0” fue acuñado *ex-ante* para la llegada de la “cuarta revolución industrial”, siendo este término una reminiscencia del versionado de software.

La idea principal de esta nueva industria es explotar el potencial de las nuevas tecnologías y conceptos, tales como (Figura 3):

- disponibilidad y uso del internet de las cosas (Internet of Things - IoT),
- mapeado digital y virtualización del mundo real, es decir, sistemas ciber-físicos (cyber-physical systems - CPS),
- empresa “inteligente”, incluyendo recursos inteligentes de la producción industrial y productos “inteligentes”.



Figura 3: Diagrama de las áreas de la Industria 4.0. Fuente: AMETIC. Recuperada de: <https://bit.ly/2Gx7N4H>

La cada vez mayor competitividad existente en el mercado global actual exige minimizar los costes a toda empresa que pretenda mantener su posición. Precisamente, la implementación de estas tecnologías puede ayudar a reducir significativamente los costes en la producción industrial. De hecho, la “industria 4.0” parece ser una solución, ya que, según varias fuentes, puede provocar la reducción de los costes de producción y logísticos en un 10-30 %, y los costes de gestión de calidad en torno a un 10-20 % [2].

Tal y como se ha mencionado anteriormente, uno de los componentes de la nueva industria son los sistemas de simulación, entre los que se incluye el gemelo digital. Según la definición propuesta por Söderberg *et al.* (2017) [3], un gemelo digital consiste en una copia digital de un sistema físico para la realización de una optimización en tiempo real. En otras palabras, esta tecnología no es más que la generación de una réplica virtual de un producto, servicio o proceso que simula el comportamiento de su homólogo físico. Esto permite monitorizarlo, analizar su reacción ante determinados escenarios y

¹ traducción libre al castellano. Alude al cambio de paradigma que implica pasar de la producción en masa a productos individualizados, adaptados al cliente.

mejorar su rendimiento y eficacia. Además, es posible experimentar sin correr riesgos, cualidad muy beneficiosa para los procesos de fabricación.

Hoy en día, gracias a la cuarta revolución industrial y sus avances en computación masiva de datos (*big data*), computación en la nube (*cloud computing*) o el internet de las cosas (*internet of things*), se han logrado nuevas posibilidades en este campo.

Una de esas oportunidades es la puesta en marcha virtual en el gemelo (*virtual commissioning*), la cual es una estrategia que permite modelar un proceso de manufactura en un entorno virtual, y, por tanto, sin necesidad de tener sistema físico durante la fase de desarrollo. Otra de las ventajas es el hecho de que ingenieros de diferentes campos tengan un modelo común con el que todos sean capaces de trabajar. Además, la puesta en marcha virtual posibilita la fácil reconfiguración de un sistema ya existente, ya que pueden realizarse cambios en el modelo digital del sistema a nivel de proceso, software o hardware [4]. Esto permite, a su vez, programar el sistema antes de que el hardware físico esté instalado, siendo el objetivo la detección temprana y corrección de los errores generados durante la planificación, el diseño y la programación. De hecho, a lo largo del desarrollo del proyecto, es posible analizar cualquier fallo para así evitarlo en la puesta en marcha del sistema físico. Por tanto, el *virtual commissioning* puede ser una herramienta muy útil a la hora de validar esta fase del proyecto. En efecto, estudios experimentales muestran tanto el efecto positivo de la puesta en marcha virtual en la tasa de error de la puesta en marcha real [5], como la reducción de hasta un 75 % en el tiempo necesario para la puesta en marcha de una planta [6].

1.3 OBJETIVOS Y ALCANCE

La consecución de los objetivos que se muestran a continuación permitirá poseer, por un lado, un banco de pruebas digital que posibilite la realización de un número ilimitado de ensayos. Por otro lado, hará que la interrupción de la producción ante imprevistos resulte en un menor impacto en la eficiencia de la empresa. Así, los objetivos establecidos son los siguientes:

Objetivo 1: Crear un gemelo digital de la estación FMS-201 plenamente funcional: el modelo debe reproducir el funcionamiento, y ser capaz de recibir y transmitir las señales, tal y como lo hace su proceso real.

Objetivo 2: Adaptar el código de control modular creado con anterioridad a las necesidades de la estación y comprobar, en simulación, el correcto funcionamiento de dicho código.

Objetivo 3: Supeditado a la verificación de los objetivos 1 y 2. Modificar el gemelo digital para la inyección de errores y la simulación de estos. Será necesario que el modelo realice los errores correctamente conforme a su descripción.

Objetivo 4: Analizar los fallos introducidos desarrollando métodos para su detección mediante el código de control y desarrollar respuestas para la resolución de dichos fallos.

Cabe mencionar, que, como objetivo transversal, se establece el uso del estándar IEC 61131-3 para que el código desarrollado pueda ser utilizado independientemente de la marca de PLC.

1.4 BENEFICIOS QUE APORTA EL PROYECTO

1.4.1 BENEFICIOS TÉCNICOS

Las técnicas para la puesta en marcha de equipos pueden beneficiarse enormemente de tecnologías como los gemelos digitales. Estas técnicas aún están por desarrollar en gran medida, debido a su reciente aparición, tal y como se menciona en la introducción. Asimismo, la consecución de este proyecto puede aportar una visión más amplia tanto de las necesidades por desarrollar en este campo, como de los beneficios que, hoy en día, puede aportar esta tecnología.

Las empresas, antes reticentes a estas tecnologías debido a su alto coste, empiezan a interesarse en las posibilidades que ofrecen. Así, grandes empresas como la planta de Mercedes (Vitoria-Gasteiz) utilizan este tipo de tecnología para la formación de sus empleados. Más allá de su uso como herramienta pedagógica, su mayor aportación a la industria actual es la posibilidad de realizar puestas a punto del control y automatización de máquinas sin la necesidad de tenerlas presentes, permitiendo acortar los plazos de puesta en marcha. Este *virtual commissioning* o puesta en marcha virtual, combinado con otras herramientas (como la simulación de procesos y de plantas industriales), permite llevar a cabo la planificación de una planta con cierta antelación.

Por otro lado, los gemelos digitales no solo pueden resultar útiles a la hora de ayudar en la puesta en marcha de la maquinaria, si no que pueden ser beneficiosas también para modificar máquinas ya existentes. Por ejemplo, es posible estudiar alteraciones en la máquina (como insertar y retirar elementos) para optimizar la producción de la misma. Además, todo ello puede realizarse sin necesidad de parar la máquina.

Otro de los beneficios técnicos que aporta el gemelo digital es la flexibilidad que otorga. Esta última es la que hace posible diseñar diferentes configuraciones para el desarrollo del producto.

1.4.2 BENEFICIOS ECONÓMICOS

Pese a lo costoso de la tecnología, la utilización de este tipo de herramientas puede tener beneficios económicos para la empresa. Anteriormente se ha mencionado el posible recorte en los plazos hasta el inicio de la producción, el cual libera de una gran cantidad de costes económicos a esta fase de la producción. Esta reducción de costes hace que la inversión inicial se realice de manera más segura. Además, la producción puede comenzar mucho antes, reduciendo así el tiempo para empezar a recuperar la inversión, y adelantando el tiempo de *payback*.

Cabe mencionar que además de la puesta en marcha virtual, esta herramienta permite el diseño de cambios de producción sin tener que parar la producción en sí misma. Esta funcionalidad sinergiza muy bien con las máquinas de diseño modular como la que se estudia en este proyecto.

Igualmente, esta tecnología permite adelantarse a los fallos que puedan surgir para tener capacidad de respuesta contra ellos. De este modo, la cantidad de horas que la producción puede estar interrumpida a causa de imprevistos puede reducirse drásticamente, y con ello, sus gastos asociados.

1.4.3 BENEFICIOS SOCIALES

La automatización de procesos libera al trabajador de una gran cantidad de carga de trabajo manual pesado, aumentando así su calidad de vida. Otro gran beneficio es el aumento de seguridad de los equipos. Al desarrollar estrategias ante fallos potencialmente peligrosos para la seguridad de los trabajadores, es posible, si bien no erradicar este peligro, reducirlo en lo máximo posible. Ya que con la ayuda de modelos digitales puede comprobarse la eficacia de las medidas adoptadas antes de la puesta en marcha de los equipos.

El aumento de la escasez de recursos, el relacionado aumento de los precios de dichos recursos, y el cambio social en el contexto de los aspectos ecológicos, hacen necesario un enfoque más intensivo en la sostenibilidad de la industria. Siendo el objetivo una mayor eficiencia tanto económica como ecológica, el gemelo digital aporta un claro beneficio social, dado que reduce la cantidad de recursos a utilizar en el ciclo de vida del producto.

Tal y como se ha mencionado en el apartado de beneficios técnicos, la función pedagógica es un punto fuerte de este tipo de tecnologías, no sólo porque mejora el sistema de educación de los alumnos, sino porque, además, aumenta la cualificación de los futuros trabajadores. Todo ello, favorece el mantenimiento de un ecosistema laboral estable y productivo.

1.5 ANÁLISIS DE ALTERNATIVAS

1.5.1 ALTERNATIVAS PARA SOFTWARE DE GAMALOS DIGITALES

NX-MCD

NX es una plataforma de software de la empresa Siemens, la cual pone a disposición del usuario soluciones de diseño, simulación y fabricación de próxima generación. Para ello, ofrece un conjunto integrado de herramientas que coordina distintas disciplinas, preserva la integridad de los datos y la intención del diseño, y optimiza el proceso al completo. Además, da soporte a todos los aspectos del desarrollo de productos, desde el diseño de conceptos hasta la ingeniería y la fabricación. Por una parte, dispone tanto de una herramienta de diseño en CAD, como de la posibilidad de introducir CAD creados en otros programas, haciendo así muy fácil el modelado. Por otra parte, dentro de esta plataforma se encuentra la aplicación *Mechatronic Concept Designer* (NX-MCD), el cual permite la colaboración multidisciplinar, la reutilización del conocimiento existente, la reducción del tiempo de comercialización y la toma de mejores decisiones, desde la creación del concepto hasta la evaluación de la producción.

CIROS® Studio

FESTO CIROS® Studio es la herramienta de trabajo profesional para crear modelos de simulación, así como una plataforma de desarrollo para uso industrial que unifica simulación, modelización y programación en una interfaz común. Además, cuenta con modelado en 3D y simulación 3D en tiempo real, incluyendo la simulación de efectos físicos, operaciones de transporte, errores y sensores, entre otros. A diferencia de los otros dos softwares, este cuenta con una versión reducida orientada a ámbitos más académicos. Aun así, en esta última versión, no existe la posibilidad

de modelar, por lo que no se ajusta a las necesidades de este proyecto y, por tanto, solo se comparará la versión Studio.

Automation studio™

Famic Technologies es una empresa canadiense que ofrece servicios en el campo de la ingeniería de software y la automatización industrial. Automation Studio™ es el más alejado a los otros dos softwares para diseño de sistemas, ya que es una herramienta más enfocada a la multidisciplinariedad y optimización que sólo permite el uso en 2D. Automation Studio™ es un software de diseño y de simulación que cubre todas las necesidades en hidráulica, neumática, electrotecnia, electricidad, automática, HMI y comunicación.

1.5.1.1 SELECCIÓN DEL SOFTWARE DE DISEÑO MECATRÓNICO

Una vez planteadas las opciones de software disponibles, se puntuó a cada una de ellas en diferentes aspectos, ponderando cada una de esas valoraciones dependiendo de la importancia dada a cada apartado escogido para su clasificación. Dichos apartados son: modelado, facilidad de uso, diseño mecatrónico, capacidad de simulación, adaptabilidad y coste económico. Cabe aclarar que cuando se hace referencia a la adaptabilidad del software, se está valorando las posibilidades de este software en diferentes campos, es decir, la capacidad multidisciplinar.

Así, la puntuación total para cada software se obtendrá sumando la multiplicación de la puntuación obtenida para cada categoría con la ponderación de la misma (Tabla 1). Tal y como se observa en la Tabla 1, la puntuación más alta obtenida es para el software Siemens NX-MCD, aunque el Festo CIROS® Studio también podría satisfacer las necesidades de este proyecto dada la corta diferencia existente entre ambos.

Tabla 1: Puntuación de las diferentes alternativas para software de gemelos digitales.

	Ponderación	Siemens NX-MCD	Festo CIROS® Studio	Automation Studio™
Modelado	3	5	4	1
Facilidad de uso	2	3	5	3
Diseño mecatrónico	5	4	4	3
Capacidad de simulación	4	4	3	4
Adaptabilidad	1	5	2	4
Coste económico	2	2	4	3
TOTAL		66	64	50

1.5.2 ANÁLISIS PARA SIMULACIÓN

Para la simulación se ha escogido entre la simulación en *software in the loop (SIL)* y *hardware in the loop (HIL)*. En el primero, la simulación se realiza en un único ordenador, es decir, sin necesidad de un PLC real; en el caso de usar esta técnica, es necesario un software que simule la función del PLC (Figura 4). Debido a que el PLC utilizado en la estación FMS-201 es un dispositivo Siemens, la única opción disponible es usar el software PLCSIM Advanced. En cambio, en el *hardware in the loop (SIL)*,

conectaremos el ordenador a un PLC real mediante hardware SIMIT UNIT PF, permitiendo así un análisis exhaustivo de los tiempos de ciclo y ejecución.

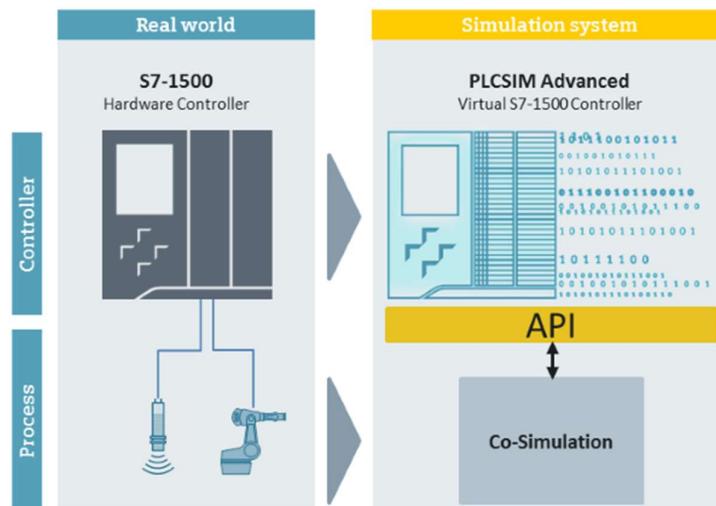


Figura 4: Esquema del funcionamiento del software in the loop (SIL) (derecha) y su representación real (izquierda).

Considerando tanto que el análisis de los tiempos de ciclo y ejecución no es parte central del objeto de estudio de este proyecto, como el alto costo y complejidad del *hardware in the loop (HIL)* respecto a la otra técnica, se escogió el *software in the loop (SIL)* como método de simulación.

1.6 ANÁLISIS DE RIESGOS

- 1. Riesgo.** Alta cualificación necesaria para el uso del software.
 Probabilidad: Muy alta. Para la utilización de estos softwares es necesaria un nivel alto de formación, dado que el conocimiento actual no es muy amplio.
 Impacto: Muy alto. El proyecto no puede completarse sin el conocimiento necesario para el uso de las herramientas.
 Puntuación: Alto (0,72)
 Plan de contingencia: Se pondrá a cargo del proyecto a una persona con una cualificación adecuada.
- 2. Riesgo.** Diseño inexacto del gemelo digital.
 Probabilidad: Alta. Debido a la cantidad de variables a considerar, resulta factible realizar mal algún ajuste tanto en los sensores como en los actuadores.
 Impacto: Alto. El diseño es la base del proyecto y, por ende, un error en este paso producirá un funcionamiento incorrecto.
 Puntuación: Alto (0,28)
 Plan de contingencia: Los errores deben solucionarse nada más sean detectados.
- 3. Riesgo.** Realizar un mapeo incorrecto.
 Probabilidad: Baja. Conectar las variables del gemelo digital a las del PLC no resulta complicado ya que la similitud en los nombres es alta.

Impacto: Medio. Pese a que un fallo en el mapeo tendrá como consecuencia el mal funcionamiento del modelo, su solución es más sencilla, lo cual provoca que el impacto sea menor.

Puntuación: Medio (0,06)

Plan de contingencia: Se usarán identificadores similares en ambas plataformas para evitar este riesgo, y en caso de detectar algún error, se proporcionará una solución lo antes posible.

4. Riesgo: Incompatibilidad en el software.

Probabilidad: Muy baja. Como los softwares que se utilizan pertenecen a Siemens, es poco probable que surjan incompatibilidades entre ellos.

Impacto: Medio. La incompatibilidad de software imposibilita la simulación; aun así, todavía permite realizar el diseño del modelo digital y del proyecto de control por separado.

Puntuación: Bajo (0,02)

Plan de contingencia: Debe comprobarse que las licencias siguen vigentes, que los permisos de usuario han sido concedidos y, como último recurso, contactar con la asistencia de la empresa.

La Tabla 2 representa la Matriz de Probabilidad-Impacto de los riesgos analizados, la cual ayuda en la determinación de la gravedad de los mismos. Los valores asignados a cada riesgo se han calculado siguiendo la Matriz.

Tabla 2: Matriz de Probabilidad-Impacto.

		IMPACTO				
		Muy bajo (0,05)	Bajo (0,1)	Medio (0,2)	Alto (0,4)	Muy alto (0,8)
PROBABILIDAD	Muy baja (0,1)	Bajo 0,005	Bajo 0,01	Bajo 0,02	Medio 0,04	Medio 0,08
	Baja (0,3)	Bajo 0,015	Bajo 0,03	Medio 0,06	Medio 0,12	Alto 0,24
	Media (0,5)	Bajo 0,025	Medio 0,05	Medio 0,1	Alto 0,2	Alto 0,4
	Alta (0,7)	Bajo 0,035	Medio 0,07	Medio 0,14	Alto 0,28	Alto 0,56
	Muy alta (0,9)	Medio 0,045	Medio 0,09	Alto 0,18	Alto 0,36	Alto 0,72

2 METODOLOGÍA

2.1 DISEÑO DEL GEMELO DIGITAL EN SIEMENS NX-MCD

Uno de los objetivos de este proyecto ha sido el diseño e implementación de un gemelo digital que represente adecuadamente la estación real. Un modelo matemático ideal podría servir como análogo de un gemelo digital, modelos matemáticos que se emplean en otros campos de la ingeniería. Por tanto, como en dichos modelos, resulta imposible crear una réplica totalmente idéntica a la original. A pesar de todo, al igual que en cualquier modelo matemático, los resultados de este gemelo digital deben asemejarse en lo máximo posible a su homólogo real. En las siguientes líneas, se describe el proceso seguido para la obtención del gemelo digital.

2.1.1 JUNTAS Y RESTRICCIONES

La creación del gemelo digital comenzó con el montaje de los diferentes componentes del modelo de la estación, piezas que ya había sido previamente modeladas en CAD. La empresa SMC proporcionó dichos componentes de la estación FMS 201. Una vez recibidos los archivos, se comenzaron a crear las restricciones necesarias para asegurar su correcto comportamiento. Para ello, todas las piezas carentes de movimiento fueron fijadas con el elemento fijo o base, garantizando así que no pudieran tener ningún grado de libertad tanto en rotación como en traslación.

A continuación, hubo que dar solución a las restricciones de los componentes que formaban parte de los elementos móviles de la máquina. Para dicho fin, debe observarse qué tipo de movimientos pueden realizar los componentes, y qué elementos llevan a cabo dichos movimientos. Los movimientos eran realizados, en su mayoría, por los actuadores neumáticos, siendo estos, sobre todo, cilindros que únicamente realizan movimientos lineales en un eje. Otros componentes que necesitaron las mencionadas restricciones fueron los pulsadores y selectores de la botonera con el fin de que el usuario pudiese interactuar desde ellos.

En aras de garantizar que el movimiento de los actuadores se produjese únicamente en el eje deseado, se aplicó la restricción de junta deslizante al vástago de cada cilindro. Para ello, se seleccionó como grado de libertad el eje paralelo al eje de revolución del vástago, y se insertó un principio y un final de carrera de cada cilindro. La misma operación fue realizada para los pulsadores, con la particularidad de que a estos se les añadió el efecto de un muelle para permitir el regreso a su posición inicial. Los selectores hicieron uso de una junta de charnela para permitir solamente la rotación sobre el eje deseado y no permitir ningún tipo de traslación.

Una vez especificados los movimientos y los componentes que los realizan, se pudo aplicar una restricción fija al resto de piezas que tenían movimiento en base a dichos componentes. Cabe aclarar, que dichas piezas son solidarias con el movimiento de los componentes; es decir, no tienen movimiento relativo respecto a las piezas móviles, pero sí respecto a la base. Asimismo, la restricción fija mencionada, fue aplicada en vez de con la base, con el elemento que comparte movimiento.

Ha debido tenerse especial cuidado con las restricciones establecidas al sistema de ventosas, ya que el sistema tenía que acompañar tanto el movimiento horizontal del cilindro A, como el vertical del cilindro B. Así, la carcasa del cilindro B tiene definida la restricción de movimiento relativo con el vástago del cilindro A, permitiendo así el movimiento del vástago B, pero haciendo que este acompañe

el movimiento horizontal. A su vez, se restringió el movimiento del sistema de ventosas con el vástago del cilindro B, consiguiendo así el efecto deseado (Figura 2).

Una vez definidas las restricciones de las ventosas arriba descritas, se dio por concluido todo lo referente a las juntas y restricciones, dando así por finalizado el ensamble (Figura 5).

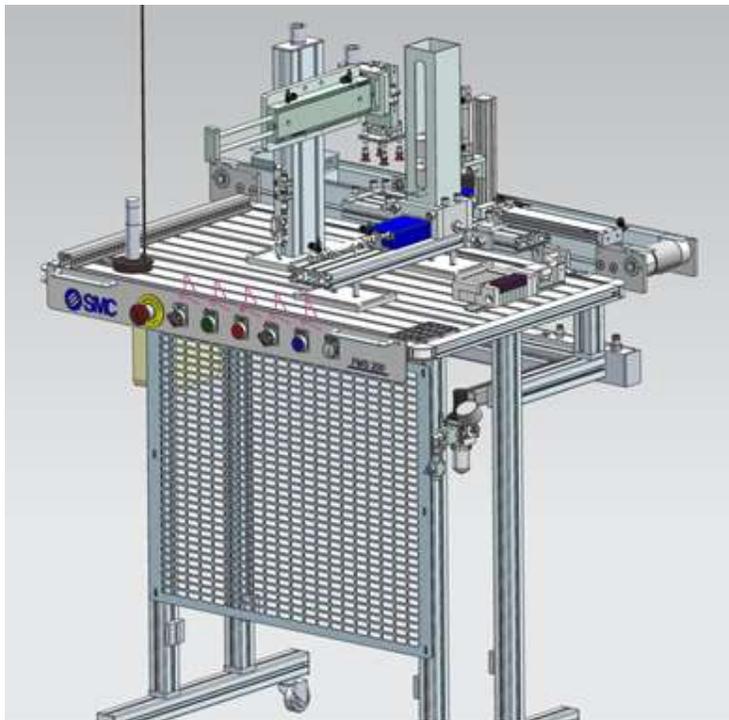


Figura 5: Modelo digital de la estación FMS-201.

2.1.2 FÍSICAS BÁSICAS

El diseño CAD tiene restringidos los movimientos que puede hacer comparado con aquellos de su homólogo en la realidad, pero carece tanto de físicas como de sensores y actuadores. Por ello, el siguiente paso que se realizó fue dotar de características físicas a los elementos. Es una realidad que el software NX-MCD demanda una alta capacidad de procesamiento, a la hora de simular las físicas en especial; por tanto, con el fin de optimizar tanto el tiempo invertido como los recursos del propio ordenador, se decidió aplicar las mismas físicas únicamente a los elementos vitales para el correcto funcionamiento del gemelo digital. Realizando un rápido análisis, se determinó que dichos elementos serían tanto los sensores y actuadores, como las superficies que fueran a tener contacto con la pieza y la propia pieza. Y, por el mismo motivo, se decidió que las superficies de contacto tuvieran las formas simples que soporta NX-MCD, es decir, esferas, ortoedros y cilindros. Además, se trató de evitar particularmente las superficies malladas, ya que consumen muchos más recursos y resultan especialmente problemáticas al interactuar con otras superficies de contacto.

Una vez hechas las determinaciones anteriormente mencionadas, se procedió a introducir las físicas en el modelo. En este proyecto se han usado tres tipos de físicas básicas: cuerpo rígido, cuerpo de colisión y fuente de objetos. Como cuerpos rígidos se declararon tanto aquellos con los que el usuario debía interactuar, como los que necesitaban que se simulase cómo les afecta la gravedad: componentes que actuaban en ejes verticales, las piezas, los pallets y los sensores. Por otro lado, fueron asignadas

como cuerpos de colisión las piezas que interactuasen con los sensores y los elementos que tuviesen colisiones con otros componentes. Por último, las piezas y los pallets se declararon también como fuente de objetos, para así poder añadir más durante la simulación.

2.1.3 SEÑALES, SENSORES Y ACTUADORES

Para finalizar con el modelo del gemelo digital, el último paso fue introducir tanto los sensores como los actuadores, para así poder recibir la información del estado de la máquina y poder dar las ordenes de control, respectivamente. Para la creación de ambos fueron necesarias la introducción de estos elementos en el modelo, por un lado, y realizar la asignación y mapeo de las señales por otro. Este paso es importante, ya que permite al PLC reconocer las entradas del programa de control y ejecutar los comandos necesarios para dar las señales de salida que controlan la estación.

A continuación, se describe la creación de las señales, paso previo a su asignación y mapeo. El número de señales es igual a la suma del número de sensores y del número de mandos que reciben los actuadores. Con el objetivo de realizar un mapeo correcto, fue necesario parametrizar como señal de entrada aquellas señales que actúan como salidas para el PLC; es decir, aquellas señales de mando que llegan a los actuadores. Asimismo, se parametrizaron como salidas las señales que recogían información de los sensores, pulsadores y selectores.

Una vez creadas todas las señales necesarias, se realizó el mapeo con los parámetros del programa de control del PLC, para así poder comprobar que los sensores y actuadores se comportaban adecuadamente. A pesar de lo contraintuitivo que el proceso arriba detallado pudiera parecer, la creación de señales fue el primer objetivo dado que los bloques de expresión necesitan señales previamente realizadas para poder ser creados.

2.1.3.1 ACTUADORES

El siguiente paso fue la creación de los actuadores, ya que, una vez introducidos, sería una tarea mucho más sencilla comprobar que los sensores recogen la información de manera correcta. En la estación (Figura 2) pueden diferenciarse hasta 7 cilindros neumáticos, y ya que todos ellos fueron creados de la misma forma para optimizar y simplificar el trabajo, se explicará a continuación el proceso de modelado.

Antes de profundizar en la explicación, cabe aclarar varios aspectos. Por una parte, así como en los cilindros neumáticos reales, la señal de mando ejercería sobre el preactuador (en este caso la electroválvula) traducándose esta señal en el movimiento del actuador, en el modelo se representa directamente dicha señal. Por otra parte, para aliviar la carga de trabajo del ordenador, se hizo uso de controles de posición en vez de los actuadores neumáticos disponibles en el catálogo de Siemens. Este uso indistinto es posible dado que el objeto de estudio de este proyecto no se centra en la simulación de los actuadores neumáticos, si no en el análisis de fallos.

El proceso de diseño comenzó con las juntas deslizantes, una de las restricciones anteriormente especificadas, a las que se les aplicaron controles de posición. Seguidamente, en los cuadros de diálogo de dichos controles, se indicaron las velocidades de traslación de los vástagos, dejando el campo de la posición como el parámetro a controlar.

Para realizar el control de posición fueron añadidos bloques de expresión. Como entradas de los bloques fueron parametrizados las señales de mando provenientes del mapeo con el PLC, excepto

el cilindro de doble efecto que necesitó dos entradas, una para la extensión de vástago y otra para la retracción. Asimismo, como salidas se establecieron los valores de posición que se dejaron libres previamente. Así, las expresiones que controlaban los valores marcaban cero cuando no había señal de mando, e indican el valor de final de carrera cuando dicho mando estaba activo. Fue posible utilizar directamente los valores cero y de final de carrera debido a que los vástagos estaban colocados inicialmente en su posición original; de hecho, de no ser este el caso y si la posición inicial fuera una intermedia, habría que calcular los valores necesarios. En el caso del cilindro de doble efecto, la expresión varió un poco, resultando en que el valor sería cero cuando el mando que indica la retracción estuviese activado, la posición de fin de carrera cuando el mando de extensión estuviese activo y la última orden recibida en las demás ocasiones. En el lenguaje de programación de NX-MCD se expresó de la siguiente forma (Figura 6):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
contr pos	real	salida	0.0	Valor de la salida control posición al actuador
Mando (cil. simple)	bool	entrada	false	Señal de mando desde PLC
pos fin carr	real	constante		Valor de la posición de fin de carrera
Retr (cil. doble efecto)	bool	entrada	false	Señal retraer desde PLC
Ext (cil. doble efecto)	bool	entrada	false	Señal extraer desde PLC

Código:

Cilindro efecto simple → ***contr pos= If (mando) Then (pos fin carr) Else (0.0)***

Cilindro doble efecto → ***contr pos= If (retr=false & ext=true) Then (pos fin carr) Elseif (retr=true & ext=false) Then (0.0) Else (contr pos)***

Figura 6: Tabla de variables de entrada y salida, y código de los cilindros.

Para modelar el efecto del pallet sobre la cinta, se declararon superficies de transporte ambas partes de la cinta. Debe tenerse en cuenta que, para poder llevar a cabo esta tarea, se les confirió la condición de cuerpos de colisión, tal y como se ha apuntado en el punto 1.2. A continuación, en el cuadro de diálogo se especificó tanto la velocidad de la cinta, como la dirección en la que debía transportar el pallet.

Debido a la limitación de los tipos de actuadores disponibles en NX-MCD, las ventosas tuvieron que ser modeladas mediante una junta fija. Dicha junta fija se controló por medio de un sensor de colisión y la orden de activación de las ventosas. Para ello, hubo que dejar libre el campo “asociación” en el cuadro de diálogo de la junta y seleccionar las ventosas como base. Posteriormente, se asoció el sensor de contacto de la ventosa con la junta fija, para controlar el campo “asociación”. El funcionamiento es el siguiente: como el sensor de colisión no detecta ningún elemento, el campo “asociación” contiene la información *null*, indicando que no hay ningún elemento asociado. Cuando la pieza entra en contacto con el sensor de la ventosa, y el mando está dado, la pieza se convierte en el elemento asociado, activando la junta fija y, por tanto, simulando el efecto de las ventosas (Figura 7).

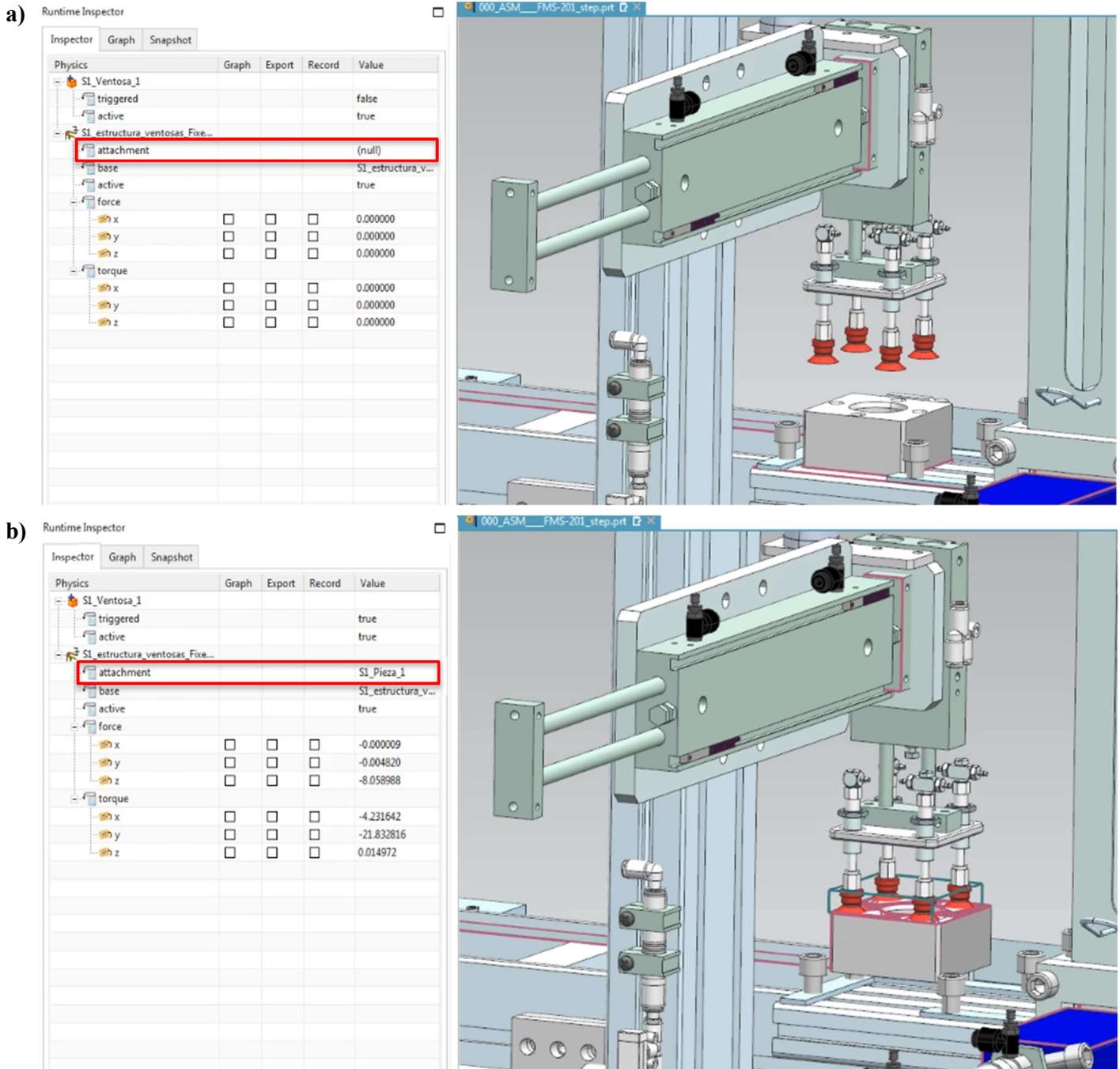


Figura 7: Modelado de ventosas. a) Antes de la activación de ventosas b) después de la activación. En ambas figuras se representa enmarcado el parámetro base.

2.1.3.2 SENSORES

Una vez modelados los actuadores, se ha procedido al modelado e implementación de los sensores. La mayoría de los sensores eran detectores magnéticos, inductivos o capacitivos; sin embargo, el software NX-MCD no dispone en su catálogo estos tipos de sensores. Ante esta problemática, se decidió utilizar sensores de contacto que cumpliesen la misma función, de manera que para el PLC fuese imposible detectar diferencia alguna. No obstante, esta solución no resultó válida para los

selectores, por lo que hubo que utilizar bloques de expresión para conseguir imitar el funcionamiento las señales de los selectores.

Debido a las limitaciones de procesamiento del ordenador, desventaja expuesta ya en el apartado de las físicas, se modelaron unas pequeñas esferas para declararlas a continuación como sensores de colisión. Con la intención de que las esferas mantuvieran la colocación correcta en todo momento, se restringió su movimiento con los elementos propietarios de cada sensor. El funcionamiento de un sensor es simple: cuando otro elemento declarado como cuerpo de colisión entra en contacto con él, cambia el estado de la señal asociada al sensor. Los sensores no tenían que tener colisión necesariamente, por lo que podían ser traspasados por otros objetos sin modificar ningún movimiento. Cada sensor fue mapeado con su señal correspondiente y, finalmente, se certificó su correcta interacción con el resto de los elementos.

Por último, se crearon los bloques de expresión para los selectores. El parámetro de entrada fue el valor del ángulo de la junta charnela creada en el apartado restricciones y juntas. En cambio, se estableció como parámetro de salida el parámetro correspondiente al PLC en el mapeo. De esta manera, al superar el ángulo la mitad de su arco total, el estado de la señal cambiaría. El código correspondiente sería el siguiente (Figura 8):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
Señal sensor	bool	salida	false	Estado de señal del sensor
ángulo	real	entrada	0.0	Ángulo del selector en un instante
Arco total	real	constante		Valor del arco máximo posible

Código:

señal sensor=If (ángulo>arco total/2) Then (true) Else (false)

Figura 8: Tabla de variables de entrada y salida, y código de los selectores.

Cabe destacar que, ante la imposibilidad de replicar un sensor como el vacuostato por limitaciones de software, se decidió activar la señal de este sensor mediante un bloque de expresión simple mientras las ventosas estuvieran activadas. Permitiendo así el correcto funcionamiento del código de control. Como entrada se declaró la señal de mando de las ventosas, y como salida la señal del sensor. El código fue el siguiente (Figura 9):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
vacuostato	bool	salida	false	Señal del sensor vacuostato
ventosas	bool	entrada	false	Señal de activación paralas ventosas

Código:

vacuostato=If (ventosas) Then (true) Else (false)

Figura 9: Tabla de variables de entrada y salida, y código de las ventosas.

Como aclaración final, cabe especificar que todos los sensores y actuadores se han descrito para el caso de un estado normalmente abierto (NA), en estos casos, en el estado de reposo el sensor se mostrará desactivado. En el caso de que fuesen normalmente cerrados (NC), los códigos se deberían ajustar a este hecho, únicamente teniendo que intercambiar entre sí los *True* y *False*.

2.2 PROYECTO DE CONTROL

El código de control se ha realizado de modo que siguiese una estructura modular en componentes mecatrónicos. De este modo, la estación se dividió en tres módulos mecatrónicos diferentes: la Botonera, la cual contiene todos los pulsadores y selectores para que el operario interactúe con la máquina (Figura 10a); el Rack, módulo que se encarga de realizar todas las operaciones de la estación (Figura 10b); y el Transporte (Figura 10c), que se encarga de enviar los pallets de una estación a otra.

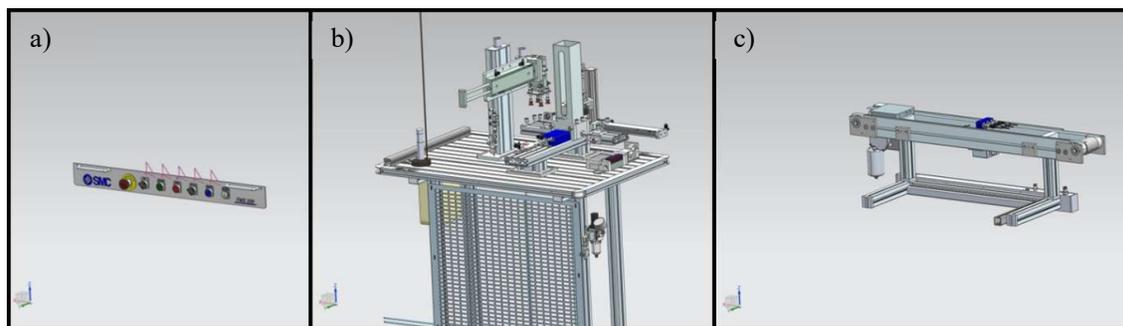


Figura 10: Modelo digital de diferentes modelos mecatrónicos. a) Botonera, b) Rack y c) Transporte.

El hardware se organizó de similar manera. A cada módulo mecatrónico se le asignó un dispositivo IO en Profinet que recogiese las señales, siendo el PLC el que actuaba como controlador IO de los tres dispositivos. Todas las conexiones se hicieron mediante el protocolo de comunicación Profinet IO creando un *bus* de comunicaciones E/S distribuidos.

Para la creación del código de control, se creó una secuencia principal donde se estableció el orden en el que serían llamadas las subsecuencias, a fin de completar correctamente la operación. Sobre el diseño realizado en proyectos anteriores, las subsecuencias del código sólo se tuvieron que adaptar al proyecto actual. Todos los bloques de programa relativos a las secuencias fueron declarados como FB (*function block*) y con parámetros formales dentro de su respectivo módulo. Posteriormente, son llamados en la FC (*function*) general de su módulo, y, por último, cada FC general de cada módulo serán llamados en el OB1 (*main*). La información que tenía que procesarse por los diferentes FB de cada módulo, fueron guardados en DB (*data block*) globales denominados *DB_local* dentro de cada módulo. Asimismo, la información que se debía comunicar entre diferentes módulos mecatrónicos fue guardada en DB globales denominados *DB_inout* (en la Figura 11) se recoge esta información de forma esquemática). A continuación, se detallan las secuencias y subsecuencias programadas. Para la implementación del código, se hizo uso del manual facilitado por Siemens [7].

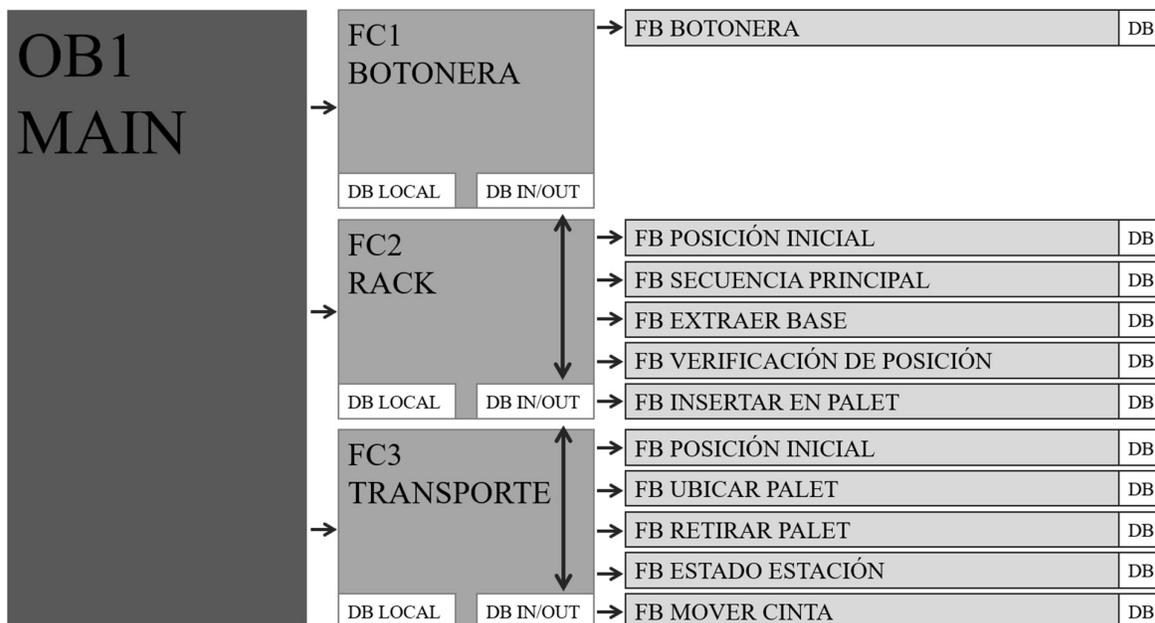
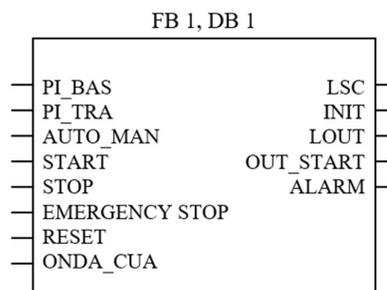


Figura 11: Esquema con la jerarquía del proyecto de control.

2.2.1 MÓDULO MECATRÓNICO BOTONERA

Parámetros I/O del FB1

IN	PI_BAS	BOOL
IN	PI_TRA	BOOL
IN	AUTO_MAN	BOOL
IN	START	BOOL
IN	STOP	BOOL
IN	EMERGENCY_STOP	BOOL
IN	RESET	BOOL
IN	ONDA_CUA	BOOL
OUT	LSC	BOOL
OUT	INIT	BOOL
OUT	LOUT	BOOL
OUT	OUT_START	BOOL
OUT	ALARM	BOOL



Módulo básico

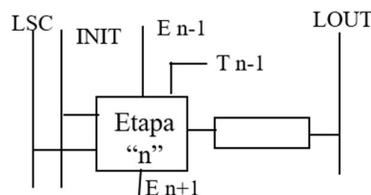


Figura 12: Tabla y diagrama de entradas y salidas y diagrama de módulo básico del FB1.

En este FB se programaron todas las operaciones de las formas de mando de la estación. Recibe la información de entrada tanto de los pulsadores de *start* y *stop* como del estado del selector de *auto/man*. Por otra parte, también se encarga de comprobar la posición inicial de todos los módulos y las secuencias mediante la salida *INIT*; así como de la liberación de cadena secuencial y la liberación de las salidas, *LSC* y *LOUT*, respectivamente (Figura 12). Cabe destacar que el hecho de que el parámetro *LSC* esté a 1 es condición necesaria para evolucionar cada etapa en las secuencias, pero, en

adelante, no se incluirá en las imágenes y explicaciones en aras de simplificar la comprensión. Tanto *LSC* como *LOUT* se activan o bien mientras el selector está en automático, o bien se pulsa *start* mientras estando en manual.

2.2.2 MÓDULO MECATRÓNICO RACK

El rack se dividió en tres diferentes conjuntos, dependiendo de su función. Estos conjuntos representan los actuadores, sensores y demás elementos que hacen la función de extraer la base del alimentador, verificar su orientación e insertar la pieza en el pallet (Figura 13).

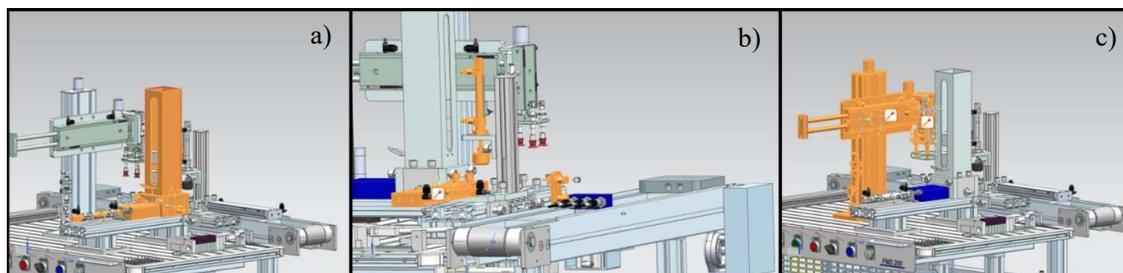


Figura 13: modelo digital de los conjuntos del módulo mecatrónico Rack. a) extracción de pallet, b) verificación de posición y c) inserción en pallet

2.2.2.1 POSICIÓN INICIAL RACK

Parámetros I/O del FB2

IN	f0	BOOL
IN	f1	BOOL
IN	d1	BOOL
IN	e1	BOOL
IN	a0	BOOL
IN	a1	BOOL
IN	b0	BOOL
IN	b1	BOOL
OUT	PI_Bastidor	BOOL

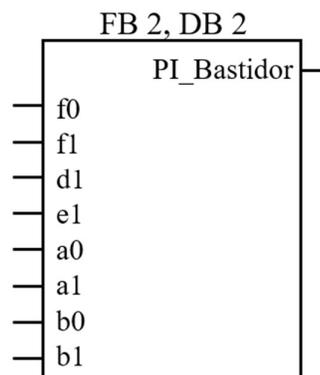


Figura 14: Tabla y diagrama de entradas y salidas del FB2.

El FB fue programado para reconocer cuándo se encontraba la máquina en posición inicial para así poder comenzar el resto de las secuencias. Esta información se comunica con el módulo de la botonera para activar el parámetro *INIT*. Para ello, todos los actuadores del módulo mecatrónico deben estar retraídos y las ventosas desactivadas (Figura 14).

2.2.2.3 EXTRAER BASE

Parámetros I/O del FB4

IN	LSC	BOOL
IN	INIT	BOOL
IN	LOUT	BOOL
IN	INI_EX_BA	BOOL
IN	bp	BOOL
IN	f0	BOOL
IN	f1	BOOL
OUT	END_EX_BA	BOOL
OUT	F	BOOL

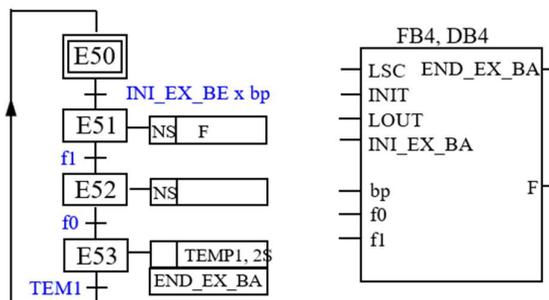


Figura 16: Tabla y diagrama de entradas y salidas, y GRAFCET del FB4.

En este FB se programó la secuencia para extraer las bases del alimentador y colocarlas en posición, para comprobar así su correcta orientación. La secuencia comienza al recibir la orden desde la secuencia principal. Si se encuentra alguna pieza en el alimentador, el cilindro se extiende hasta el final de carrera y a continuación se contrae hasta la posición inicial. En el caso de no encontrar ninguna pieza, esperará hasta que el alimentador reciba una. Por último, antes de volver a la primera etapa, al cabo de dos segundos, se comunica a la secuencia principal que la secuencia ha finalizado, mediante la activación de la señal correspondiente (Figura 16).

2.2.2.4 VERIFICAR POSICIÓN

Parámetros I/O del FB5

IN	LSC	BOOL
IN	INIT	BOOL
IN	LOUT	BOOL
IN	INI_VE_PO	BOOL
IN	d1	BOOL
IN	e1	BOOL
OUT	END_VE_OK	BOOL
OUT	END_VE_DE	BOOL
OUT	C	BOOL
OUT	D	BOOL
OUT	E	BOOL

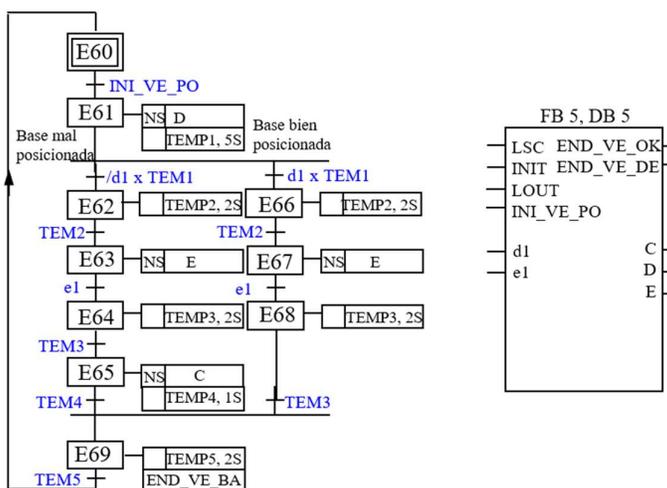


Figura 17: Tabla y diagrama de entradas y salidas, y GRAFCET del FB5.

Este FB se diseñó para identificar si la pieza está colocada correctamente o no. Las piezas adecuadamente situadas continuarán la secuencia, mientras que las que estén mal posicionadas se retirarán de la producción. El FB comienza al recibir la orden de la secuencia principal, momento en el que el cilindro vertical se extenderá durante 5 segundos, los suficientes para llegar al final de carrera.

A continuación, la secuencia se separa en dos ramas. Si el cilindro llega al final de su recorrido, el sensor de final de carrera se activa dando a entender que la pieza está bien colocada. Posteriormente, el cilindro vertical se retrae, y el cilindro horizontal se extiende hasta su final de carrera para colocar la pieza en posición para la siguiente secuencia. Seguidamente, el cilindro horizontal vuelve a su posición original. Por el contrario, si el sensor del cilindro vertical nunca llega a encenderse, la pieza estaría mal posicionada, y se repetiría la operación anterior, con la salvedad de que cuando el cilindro horizontal regrese a su posición inicial, la pieza será expulsada por el expulsor. Antes de volver a la primera etapa, el FB envía la información del estado a la secuencia principal activando la salida *END_VE_DE* o *END_VE_OK*, en función de si está mal o bien colocada, respectivamente (Figura 17).

2.2.2.5 INSERTAR EN PALLET

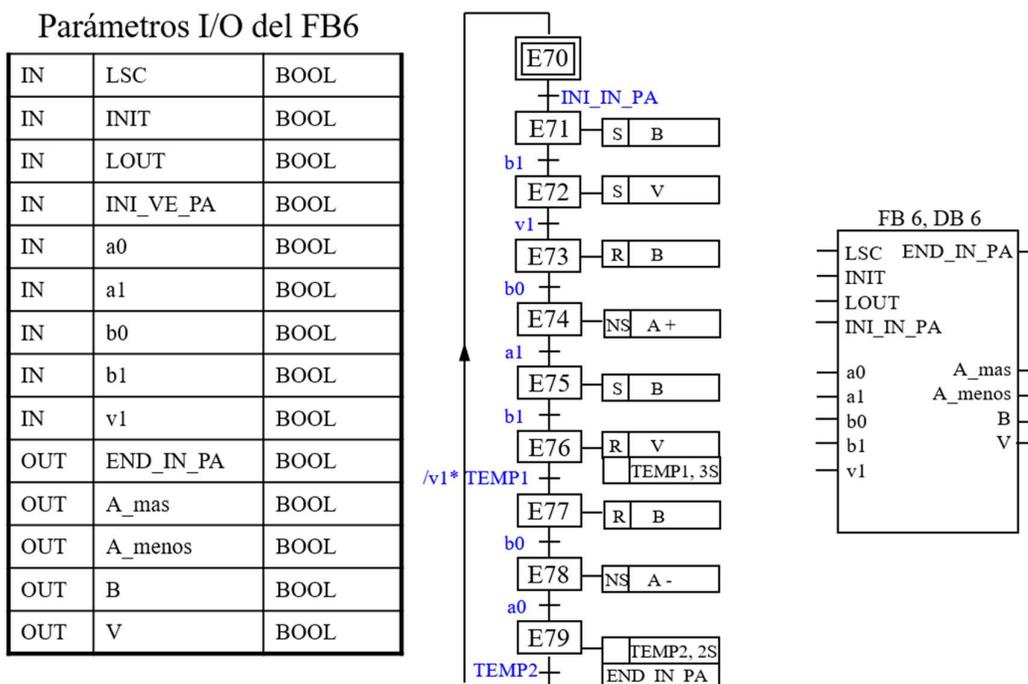


Figura 18: Tabla y diagrama de entradas y salidas, y GRAFCET del FB6.

En caso de estar la pieza correctamente orientada, este FB ordena la inserción de la pieza al pallet de transporte. La secuencia comienza al recibir la orden de la secuencia principal, al igual que las anteriores. El cilindro vertical desciende y, a continuación, las ventosas reciben la orden de provocar el vacío. Cuando el vacuostato activa la señal de que las ventosas han hecho el vacío, el cilindro vertical asciende y, al llegar a la posición inicial, el cilindro horizontal avanza hasta su final de carrera. Después, el cilindro vertical se extiende hasta su final de carrera, y al finalizar su recorrido, las ventosas dejan de hacer el vacío depositando así la pieza en el pallet. En las dos etapas de activación y desactivación de las ventosas hay un temporizador para garantizar el correcto funcionamiento de ambas. Una vez depositada la pieza, el cilindro vertical asciende, y después el cilindro horizontal se retrae, quedando así en su posición inicial. Por último, el FB envía la información de que su secuencia a finalizado y vuelve a la primera etapa (Figura 18).

2.2.3 TRANSPORTE

2.2.3.1 POSICIÓN INICIAL TRANSPORTE

Parámetros I/O del FB 7

IN	ppA	BOOL
IN	b0A	BOOL
IN	b1A	BOOL
IN	b2A	BOOL
IN	Sensores_Palet_IN	BYTE
OUT	PI_MM_TRANSPORT	BOOL
OUT	Sensores_Palet_OUT	BYTE

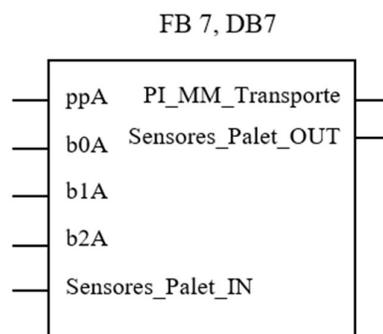


Figura 19: Tabla y diagrama de entradas y salidas del FB7.

Al igual que en el Rack, este FB se programó para estudiar si el módulo de transporte se encuentra en posición inicial. Para ello, se comprueba que tanto el sensor del pallet como los sensores que recogen el código del pallet están en 0, que el cilindro está extendido y que la cinta no está en movimiento. Además, se guarda la información del código del pallet para enviarlo al Rack. Se debe recordar, que este es un código modular, por lo que, aunque esta estación no necesite el código del pallet, otras sí lo requieren (Figura 19).

2.2.3.2 UBICAR PALLET

Parámetros I/O del FB 8

IN	LSC	BOOL
IN	INIT	BOOL
IN	LOUT	BOOL
IN	INI_UBI_PA	BOOL
IN	ppM	BOOL
IN	E500	BOOL
IN	pp	BOOL
IN	CO_PA_b0	BOOL
IN	CO_PA_b1	BOOL
IN	CO_PA_b2	BOOL
OUT	END_UBI_PA	BOOL
OUT	M1	BOOL

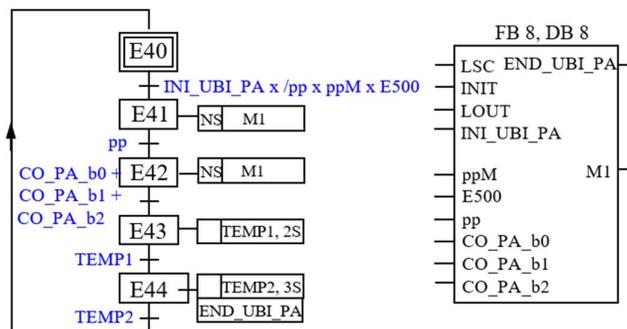


Figura 20: Tabla y diagrama de entradas y salidas, y GRAFCET del FB8.

El **FB Ubicar Pallet** fue programado para llevar el pallet desde el inicio de la cinta hasta la posición de recepción de la pieza. El FB se inicia al pulsar el pulsador *start*, siempre y cuando no haya un pallet en posición. La cinta se activa en ese momento hasta que tanto el sensor de posición como los del código se activan. Tras esperar un segundo para dar tiempo a colocar perfectamente el pallet se manda la señal de que la secuencia ha terminado, señal que se envía a su vez a la secuencia principal del rack para empezar con el **FB extraer base** (Figura 20).

2.2.3.3 RETIRAR PALLET

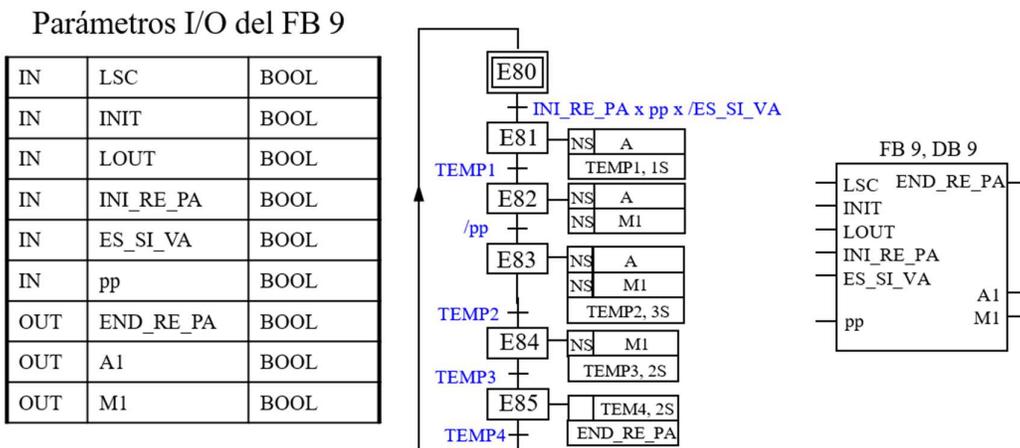


Figura 21: Tabla y diagrama de entradas y salidas, y GRAFCET del FB9.

Este FB fue programado para, una vez recibida la pieza mandar el pallet con la pieza a la siguiente estación. La secuencia comienza cuando en la secuencia principal del Rack se finaliza la etapa insertar en pallet. El cilindro que contiene los sensores que leen el código del pallet se retrae para dejar pasar el pallet y la pieza. Una vez pasado el tiempo suficiente establecido para que el cilindro complete su acción, se activa la cinta y se mantiene activa hasta tres segundos después de que el sensor de posición se desactive, para dar tiempo al pallet a superar el cilindro retraído. Tras ello, el cilindro vuelve a extenderse mientras la cinta sigue activa otros dos segundos, tiempo necesario para que el pallet llegue a la siguiente estación. Finalmente, se da la señal de que la secuencia ha sido completada (Figura 21).

2.2.2.4 ESTADO ESTACIÓN

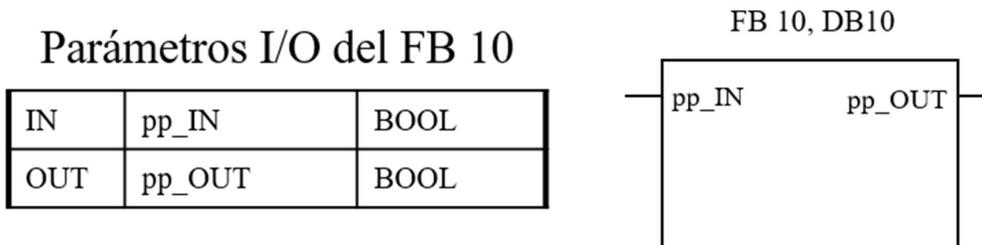


Figura 22: Tabla y diagrama de entradas y salidas del FB10.

El FB se programó con el fin de poder comunicar a otras estaciones si esta está ocupada o libre. Únicamente envía la señal del sensor de posición de la cinta al parámetro *pp_estacion_out* para que este se comunique con otras estaciones (Figura 22).

2.2.2.5 MOVER CINTA

Parámetros I/O del FB 11

IN	M1_UBICAR_PALET	BOOL
IN	M1_RETIRAR_PALET	BOOL
IN	S2T_FREE	BOOL
OUT	M1_Transport	BOOL

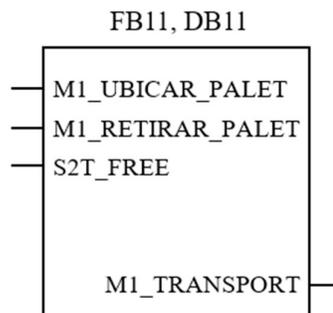


Figura 23: Tabla y diagrama de entradas y salidas del FB11.

Este FB se programó para activar la cinta. Es necesario introducirlo ya que la cinta se mueve tanto en ubicar pallet como en retirar pallet, es decir, dos FB diferentes, al contrario que el resto de los actuadores. Este hecho hace que, si no se introduce este FB, la salida de la cinta recibe información contradictoria de cada FB, ya que, mientras en uno está activo en el otro se mantendrá inactivo, provocando un mal funcionamiento. Por ello, la salida del motor se controla mediante un bloque *or* en este FB. El resto de los actuadores no necesitan este tipo de FB, debido a que únicamente se usan en un FB, lo que evita que reciban información contradictoria (Figura 23).

2.3 ANÁLISIS PRELIMINAR DE FALLOS

Dada la cantidad de fallos detectados, fue una tarea aún más crucial analizar la detección de cada fallo y la resolución del mismo. Para llevar a cabo esa clasificación, se adoptó la guía MeiA [8] como recurso para la planificación de estas respuestas. Se determinó que todos los temporizadores y contadores que necesitaran ser usados, serían del estándar IEC 61131-3 para mantener la modularidad y el programa de control en cualquier PLC.

Para que el operario pudiera ejecutar el aviso y la respuesta, se introdujo un HMI que permitiese llevar a cabo dichas acciones [9]. Este HMI ejecuta en TIA Portal con un *run-time WinCC*. El HMI se diseñó de tal manera que en la página principal se pudiese observar la secuencia y en qué punto de la misma se encontraba la estación. En esta misma imagen aparecería una notificación en caso de haber algún fallo. Asimismo, se añadió una segunda imagen donde se podrían consultar el historial de los fallos, y realizar las operaciones para su resolución. Por último, se agregó una tercera imagen con los mandos manuales de la estación, solamente accesible en caso de paro de emergencia. Además, en todas las imágenes se podrían consultar el estado tanto de los mandos de la botonera, como el de la barra luminosa. Para llevar a cabo el intercambio de parámetros entre el PLC y HMI, se creó un DB global que contuviese toda la información necesaria (ver Anexo 1).

2.3.1 DETECCIÓN

Con el fin de homogeneizar el código, se crearon unos métodos de detección lo más replicables posibles. Se pensaron diferentes modos de detección respondiendo a las necesidades de los fallos a detectar. Sin embargo, aquellos fallos relacionados con defectos en el cableado de los mandos de la botonera resultaban imposibles de detectar mediante el código; por tanto, se creó un método que comparara las diferencias entre las señales que recibía y enviaba el PLC (vía HMI) con lo que ocurriera en la realidad (vía estación real) mediante observación del operario.

Para aquellos fallos en los que los sensores, debido a algún tipo de defecto, nunca llegaban a activarse, o aquellos otros en los que los actuadores se averiaban, se diseñó un detector mediante tiempo de vigilancia o tiempo de etapa máximo (Figura 24). Para ello, se colocaron temporizadores de retardo en las etapas enlazadas a este tipo de fallos; dichos temporizadores se programaron con el doble del tiempo necesario para realizar la acción. Si estos temporizadores se activaban, significaría que el sensor (o actuador) necesario para la evolución de etapa no estaba funcionando correctamente.

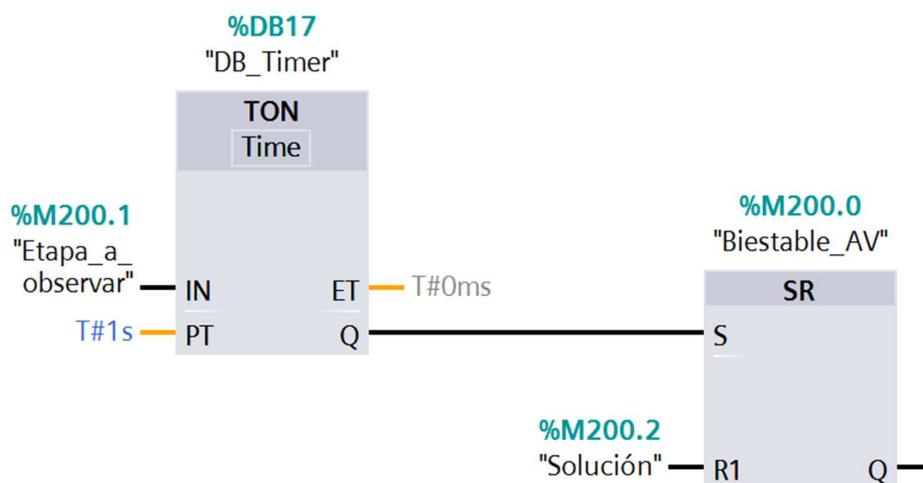


Figura 24: Código de la detección correspondiente al método tiempo de vigilancia.

En cambio, para los defectos en los que sensores estaban activados por error, se diseñó un temporizador de etapa mínimo como método (Figura 25). En este método, mediante un temporizador de pulsos, se envía un pulso de una duración muy inferior al tiempo necesario para completar la acción, pero más largo de la duración de un ciclo. Durante ese pulso, se comprueba si la etapa a observar sigue activa, y, de no ser así, el biestable que indica el fallo se activa, hasta que recibe la información de que el fallo ha sido resuelto.

Otro de los métodos implantados fue la detección por coincidencia simultánea (Figura 26). Este método se usó para detectar la simultaneidad de activación de dos sensores o actuadores que nunca debieran estar activos a la vez. La detección es muy simple: cuando las dos señales están activas, se guarda esa información en el biestable asignado al fallo. No obstante, debe de ponerse especial cuidado con este método, pues puede causar detecciones incorrectas cuando hay varios fallos simultáneos. Este

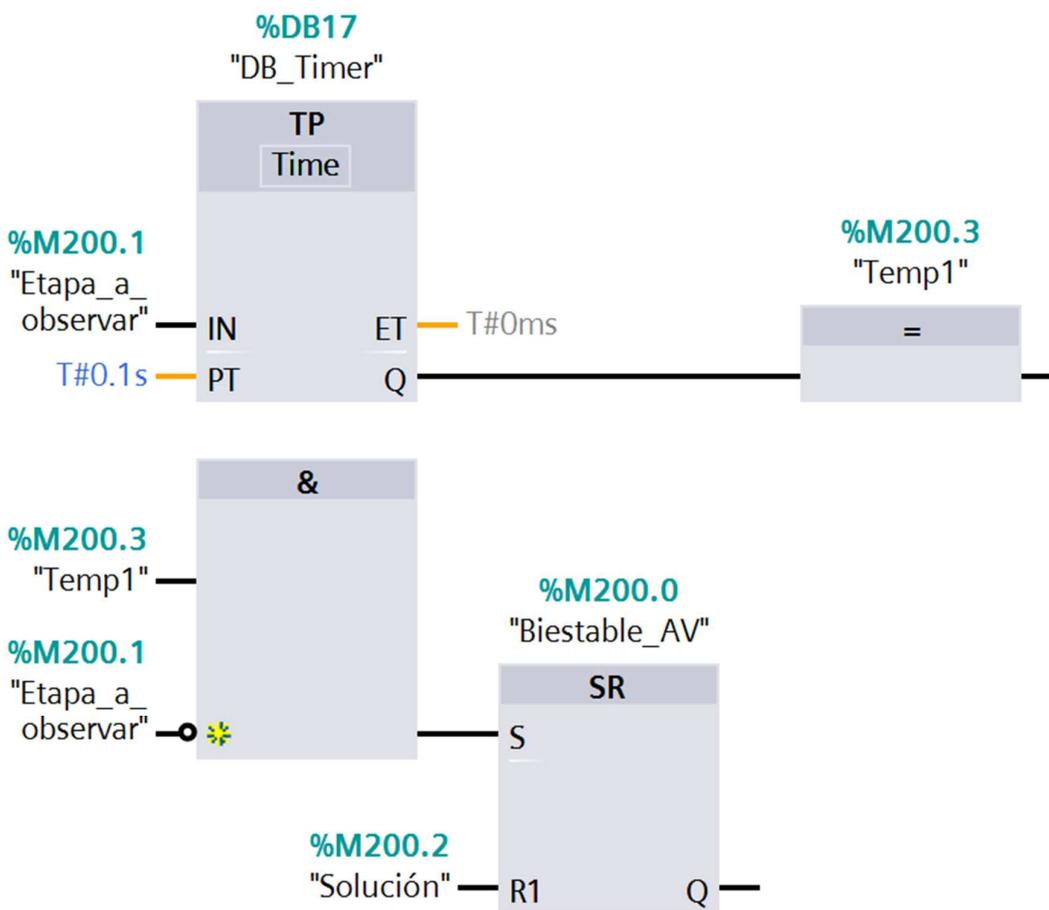


Figura 25: Código de la detección correspondiente al método tiempo mínimo de etapa.

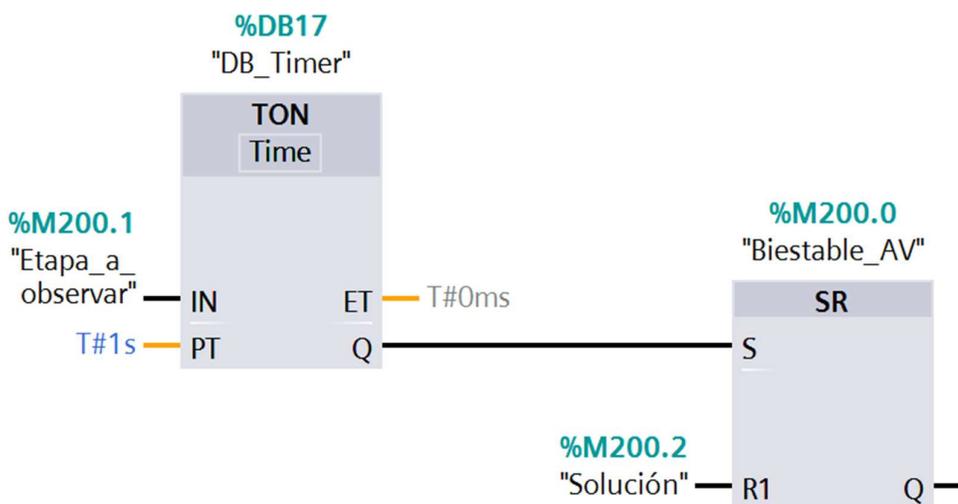


Figura 26: Código de la detección correspondiente al método coincidencia simultánea.

método se solapaba, en ciertos casos, con el temporizador de etapa mínimo, condición que hacía imperativa la elección entre ambos métodos. Debido a la robustez del temporizador de etapa mínimo, en esos casos, se eligió este como método de detección.

El último método que se creó fue la detección mediante contador (Figura 27). En este método se utiliza un contador para contar un evento a observar, de modo que, si este evento ocurre un número seguido de veces, un biestable avisa del fallo. Como ejemplo, cuatro piezas consecutivas mal colocadas activarían el aviso; sin embargo, el paso de una pieza buena resetearía el contador, mientras que solucionar el sensor, resetearía el biestable. Cabe destacar que este método es especialmente útil cuando hay ramificaciones en la secuencia.

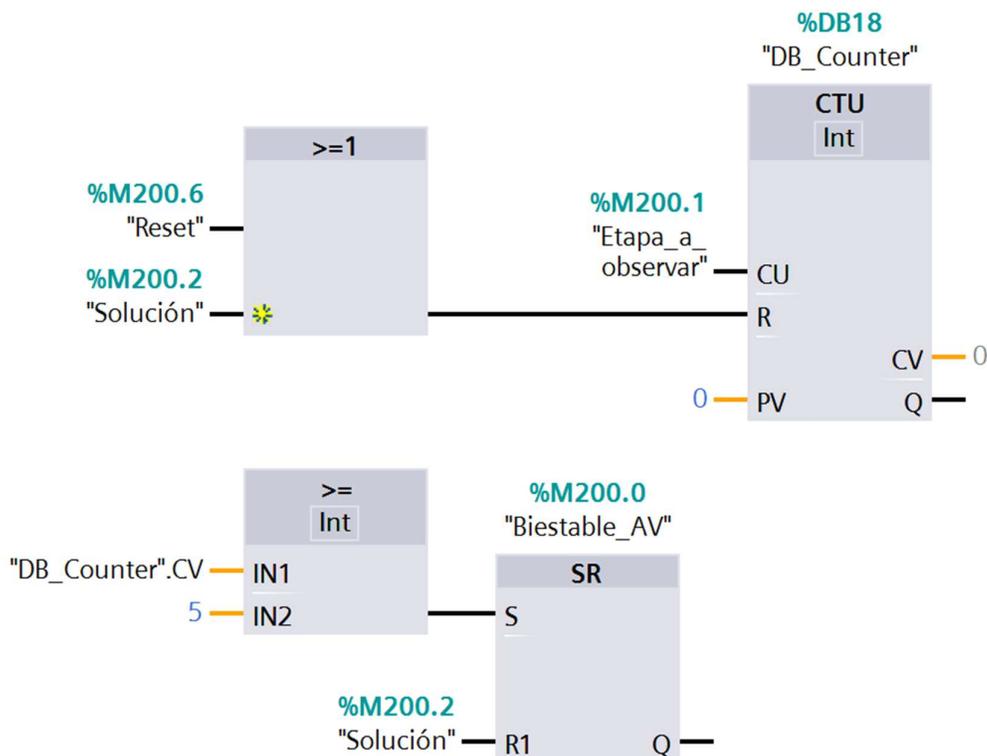


Figura 27: Código de la detección correspondiente al método detección mediante contador.

2.3.2 DIAGNÓSTICO

Los fallos en los actuadores o en los pulsadores y selectores de la estación fueron clasificados como fallos **no solucionables** (Fig. 28), ya que, debido a la naturaleza del fallo, carecen de resolución posible. Esto es, ante el fallo de un actuador, por ejemplo, no hay forma de responder por parte del control. La única solución es intercambiar el elemento en mal estado por uno plenamente operativo, por lo que carece de demasiado interés en lo que al control respecta. A pesar de todo, una vez intercambiado el elemento, la máquina deberá volver a posición inicial de forma manual para poder iniciar un nuevo ciclo.

Por otra parte, aquellos fallos que dañaran la pieza o pusieran en peligro tanto al operario como la integridad de la estación, fueron catalogados como **paros de emergencia** (Fig. 28). En este tipo de fallos la estación cesa toda actividad al instante y manteniendo activos únicamente los actuadores necesarios para garantizar la seguridad (las ventosas cuando la pieza está sujeta). Ante la imposibilidad

de saber cómo está el estado de la máquina tras el fallo, el operario realizará todos los movimientos necesarios manualmente desde el HMI para la devolución de la máquina a posición inicial. Todo ello, garantizando siempre la seguridad del operario, y teniendo preferencia la seguridad del individuo a la de la estación. Tras la resolución del problema la máquina estará lista para continuar la producción.

Por último, aquellos fallos con los que se podía lidiar de manera manual mediante HMI y no pusieran en peligro la seguridad, se catalogaron como **fallo aceptable en producción** (Fig. 28). En este tipo de fallos la secuencia solo evolucionará cuando el operario se cerciore de que la máquina está en estado de pasar a la siguiente etapa, y dé la orden manualmente mediante el HMI. Tras ello, la producción continuará en modo automático o manual dependiendo de con qué modo se haya comenzado.

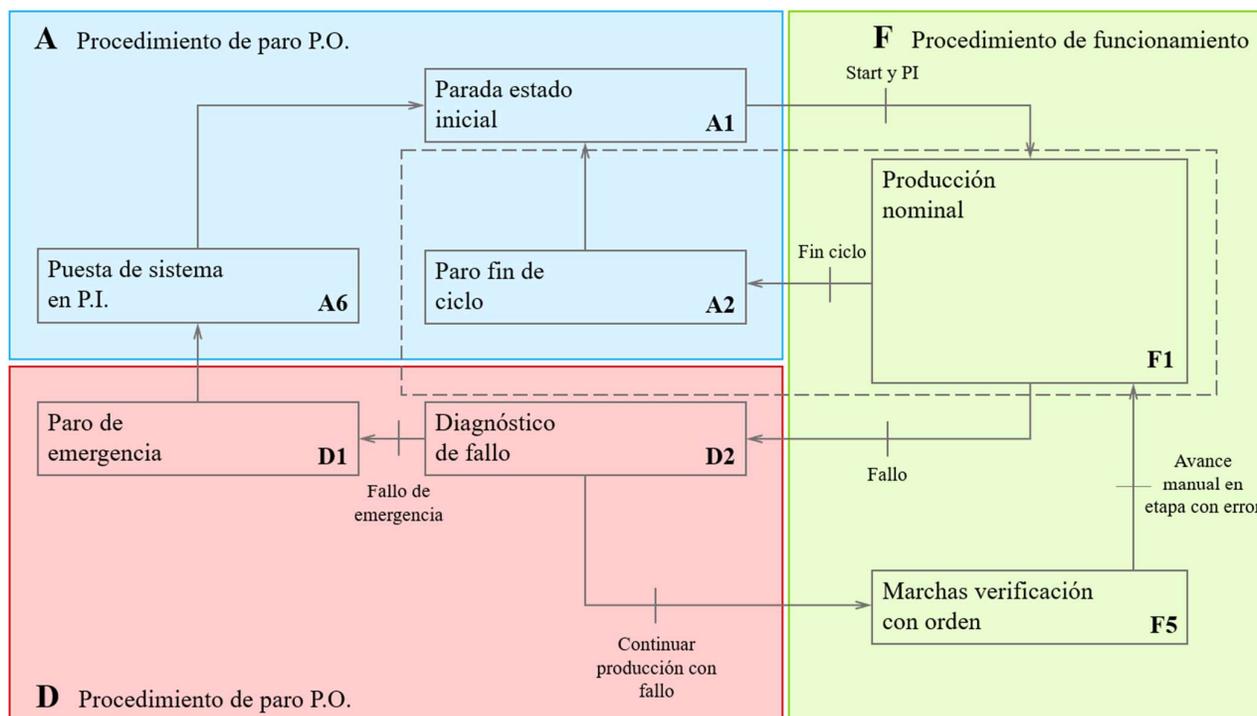


Figura 28: GEMMA correspondiente a la secuencia diseñada.

2.3.3 ANÁLISIS

Para comenzar, se realizó un análisis preliminar de los dieciséis fallos. Para poder identificar estos fallos, se recurrió a la lista ofrecida por el fabricante:

Nº1

Avería La estación no puede ponerse en marcha

Componente Pulsador de marcha

Detección Observación del operario

Aviso Sin aviso

Diagnóstico No solucionable

Resolución Cambiar el componente dañado

Nº2

Avería La estación solo funciona en modo automático

Componente Selector *auto/man*

Detección Observación del operario

Aviso Sin aviso

Diagnóstico No solucionable

Resolución Cambiar el componente dañado

Nº3

Avería El cilindro vertical de la verificación no desciende

Componente Sensor fin de carrera del cilindro que extrae la base

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente

Nº4

Avería El cilindro vertical de la inserción en pallet no asciende

Componente vacuostato

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente

Nº5

Avería Se expulsa el cuerpo, aunque esté bien orientado

Componente detector fin de carrera del cilindro vertical de la verificación

Detección Contador piezas malas consecutivas (4 unidades)

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI se despliega las opciones para introducir si la pieza es buena mala o el fallo se ha solucionado y al pulsar *OK* introducimos la señal manualmente.

Nº6

Avería La estación no se puede poner en marcha en modo automático

Componente Pulsador de paro

Detección Observación del operario

Aviso Sin aviso

Diagnóstico No solucionable

Resolución Cambiar el componente dañado

Nº7

Avería El cilindro horizontal de la verificación no retorna

Componente Sensor fin de carrera del cilindro horizontal de la verificación

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente

Nº8

Avería Se suelta el cuerpo desde arriba

Componente Sensor fin de carrera del cilindro vertical de la inserción en pallet

Detección Tiempo de etapa mínimo

Aviso Barra luminosa y HMI

Diagnóstico Paro de emergencia

Resolución Puesta en posición inicial manual mediante HMI

Nº9

Avería El cilindro vertical de la inserción en pallet desciende mientras el horizontal avanza

Componente Sensor fin de carrera del cilindro horizontal de la inserción en pallet

Detección Tiempo de etapa mínimo

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente

Nº10

Avería El cilindro horizontal de la inserción en pallet no retorna

Componente Sensor principio de carrera del cilindro horizontal de la inserción en pallet

Detección Tiempo de etapa mínimo

Aviso Barra luminosa y HMI

Diagnóstico Paro de emergencia

Resolución Puesta en posición inicial manual mediante HMI

Nº11

Avería El cilindro vertical de la inserción en pallet asciende mientras el horizontal avanza

Componente Sensor principio de carrera del cilindro vertical de la inserción en pallet

Detección Simultaneidad de señales

Aviso Barra luminosa y HMI

Diagnóstico Paro de emergencia

Resolución Puesta en posición inicial manual mediante HMI

Nº12

Avería La estación no comienza

Componente Sensor inicio de carrera del cilindro de la extracción de base

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente (conjunto con la Nº16)

Nº13

Avería No se traslada el cuerpo al finalizar la verificación, en su lugar desciende el cilindro vertical de la inserción

Componente Cableado de ambos actuadores intercambiado

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Fallo no solucionable

Resolución Cambiar el componente dañado

Nº14

Avería No avanza el cilindro horizontal de la inserción

Componente Actuador cilindro horizontal de la inserción

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Fallo no solucionable

Resolución Cambiar el componente dañado

Nº15

Avería No se crea el vacío

Componente Actuador ventosas

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Fallo no solucionable

Resolución Cambiar el componente dañado

Nº16

Avería La estación no inicia el ciclo

Componente Cableado de los sensores de inicio y final de carrera del cilindro vertical de la inserción intercambiados

Detección Tiempo de vigilancia

Aviso Barra luminosa y HMI

Diagnóstico Aceptar producción con fallo

Resolución Al pulsar la avería correspondiente en el HMI y pulsar *OK* introducimos la señal manualmente (conjunto con la Nº12)

2.4 INYECCIÓN DE FALLOS

Con el fin de poder ver el efecto de los fallos y poder simularlos en el gemelo digital fue necesario tanto introducir componentes nuevos en el gemelo digital que permitieran la activación de dichos fallos, como modificar las señales del modelo para adaptarlas a los mismos.

2.4.1 MODELADO

En primer lugar, se procedió a realizar un modelo que permitiese la interacción usuario-gemelo digital durante la simulación. Con el fin de mantenerse lo más fiel a la realidad posible, se introdujo en el diseño una plataforma con interruptores similar a la original, aunque se cambió la posición para hacerlo más funcional. Como el modelo de los interruptores no estaba disponible, se crearon unos modelos desde cero.

Los interruptores contaban con una base, creada mediante extrusión, donde se alojaban tanto el interruptor como una lámpara indicadora. En la abertura donde se aloja el interruptor, se colocó un eje simple sobre el que giraría el interruptor, y sobre este eje el interruptor creado mediante revolución. En cuanto a las restricciones, se fijó el eje a la base y esta última a la plataforma, ambas mediante una junta fija. Al interruptor, sin embargo, se le aplicó una junta charnela para permitir la rotación sobre el eje. Para posibilitar la interacción entre usuario y modelo, el interruptor se declaró como cuerpo rígido, dotándolo de esta cualidad para ahorrar costes de procesamiento únicamente, tal y como se indicó en el apartado físicas básicas. El tratamiento de las señales fue el mismo que el de los selectores mencionado anteriormente en el punto 1.3.2. Esta vez, se sumó un nuevo bloque de expresión para unir la señal del fallo con la activación del color de su respectiva lámpara indicadora. Para poder cambiar el color de la lámpara indicadora, fue necesario introducir el actuador permutador de pantalla, programando a su vez el bloque de expresión que controlaba dicho actuador. Como entrada se parametrizó la señal del interruptor, y como salida, tanto el “modo de ejecución”, como el color. El parámetro “modo de ejecución” debía estar siempre en 1 para poder funcionar, y el color variaba de blanco (0) cuando el interruptor estaba apagado, a rojo (155) cuando el mismo está encendido. El código fue el siguiente (Fig. 29):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
interruptor	bool	entrada	false	Estado del interruptor de la avería
mod ejec	int	salida	0	Valor de salida del modo de ejecución (siempre 1)
color	int	salida	0	Valor del parámetro color (0 blanco, 155 rojo)

Código:

```

mod ejec=If (interruptor) Then (1) Else (1)
color=If (interruptor) Then (155) Else (0)
  
```

Figura 29: Tabla de variables de entrada y salida, y código de lámpara indicadora de los interruptores.

Una vez finalizado el diseño de un interruptor (Fig. 30), se insertaron 16 copias del mismo, cambiando en cada caso el nombre de las señales y bloques de expresión para poder diferenciar claramente cada una entre sí.

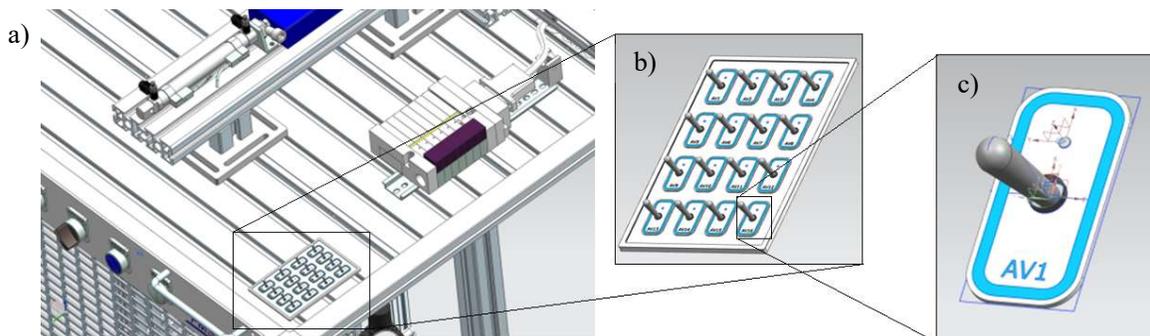


Figura 30: Modelo de los interruptores. a) Posicionamiento del panel de interruptores en la estación. b) Panel de interruptores aislado. c) Detalle de un interruptor.

2.4.2 TRATAMIENTO DE SEÑALES EXISTENTES

Una vez se crearon las 16 señales que indicaban cuándo el usuario activaba una de estas, se procedió a diseñar la lógica que simulara el efecto de cada avería en el modelo. Para ello, solamente hace falta tratar las señales con bloques de expresión que imiten la avería. Con la intención de simplificar el trabajo se trató de que estas soluciones fueran lo más genéricas y replicables posible. Por otro lado, los fallos se dividieron en dos grupos, los que afectan a sensores, y los que afectan a actuadores.

Las averías que afectaban a sensores, pulsadores o selectores se replicaron mediante el uso de un solo bloque de expresión. En dicho bloque se parametrizó como entrada tanto la señal del interruptor de la avería como la señal del propio sensor, y como salida, se declaró una nueva señal que se mapeara sustituyendo a la del sensor original. Esta señal se nombró al igual que la señal del sensor original, añadiendo el número de la avería como sufijo (*fl_AV4*, por ejemplo). En cuanto a la expresión se programó para que mientras el interruptor se mantuviese activo siempre diera 0 (o 1, depende del caso), y cuando el interruptor no esté pulsado se reproduzca la información del sensor. El código es el siguiente (Fig. 31):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
Sensor_AVX	bool	salida	false	Señal del sensor con la señal tratada
interruptor	bool	entrada	false	Estado del interruptor de la avería
sensor	bool	entrada	false	Señal original del sensor

Código:

Avería sensor desactivado → *sensor_AVX=If (interruptor=false & sensor=true) Then (true) Else (false)*
 Avería sensor activado → *sensor_AVX=If (interruptor=false & sensor=false) Then (false) Else (true)*

Figura 31: Tabla de variables de entrada y salida, y código de sensor con fallo.

Sin embargo, dos de las averías necesitaron adaptarse para poder conseguir sus efectos. En primer lugar, la 8ª avería, donde la pieza se soltaba desde arriba, se podría traducir como si el sensor de fin de carrera del cilindro vertical en la inserción del pallet estuviera activo. No obstante, al contrario de los demás, no se podía hacer que ese sensor estuviese siempre activo ya que aparte de incumplir las condiciones iniciales, bloqueando el inicio de la secuencia, provocaría que el primer movimiento de descenso para recoger la pieza no se hiciese, bloqueando la secuencia. Para solucionarlo, se optó por añadir más condiciones al bloque de expresión, incluyendo la necesidad de que las ventosas estuvieran activas, caso que solo se daba en la segunda bajada. Para ello, fue necesario meter la señal de mando de las ventosas como entrada. El código modificado fue el siguiente (Fig. 32):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
Sensor_AVX	bool	salida	false	Señal del sensor con la señal tratada
interruptor	bool	entrada	false	Estado del interruptor de la avería
sensor	bool	entrada	false	Señal original del sensor
ventosas	bool	entrada	false	Señal de mando de las ventosas del PLC

Código:

sensor_AVX=If (interruptor=false & sensor=true) Then (true) Elseif (interruptor=true & ventosas=true) Then (true) Else (false)

Figura 32: Tabla de variables de entrada y salida, y código de ventosas con fallo.

El segundo error que hubo que modificar fue el 9º, donde el cilindro horizontal de la inserción en el pallet avanzaba a la vez que el vertical descendía. De forma similar a la anterior, se podría traducir como si el sensor del cilindro horizontal estuviese siempre activo, solo que ese sensor estuviese a 0 era condición necesaria para que la máquina reconociese la posición inicial. Para solucionarlo, se introdujo la misma fórmula que con la 8ª avería, evitando así el bloqueo al comienzo de la secuencia.

Para los actuadores se siguió una estrategia similar. Aun así, estos componentes tenían la peculiaridad de que, al estar creadas las señales en subconjuntos del ensamble, era imposible comunicar las señales de los interruptores con los controles de posición. Como solución, se introdujeron señales nuevas en los subconjuntos y se conectaron mediante bloques de expresión a la señal del interruptor en el ensamblaje final. La entrada del bloque era la señal del interruptor, y la salida, la señal recién creada. Una vez la información del interruptor estaba en el subconjunto, se procedió a modificar la expresión del actuador, introduciendo la condición de que el interruptor no estuviese activo para que se produjera la acción. En el caso del cilindro de doble efecto, cuando el interruptor estaba activado, se mantenía la posición de ese momento. Las expresiones fueron las siguientes (Fig. 33):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
contr pos	real	salida	false	Valor control posición
interruptor	bool	entrada	false	Estado del interruptor de la avería
mando	bool	entrada	false	Señal de mando del cilindro simple
Pos fin carrera	real	constante		Valor de la posición de final de carrera
retr (cil. Doble efecto)	bool	entrada	false	Orden de retracción del cilindro de doble efecto
Ext (cil. Doble efecto)	bool	entrada	false	Orden de extensión del cilindro de doble efecto
ventosas	bool	entrada	false	Señal de mando de las ventosas

Código:

Cilindro efecto simple → *contr pos= If (mando=true & interruptor=false) Then (pos fin carr) Else (0.0)*
 Cilindro doble efecto → *contr pos= If (retr=false & ext=true & interruptor=false) Then (pos fin carr) Elseif (retr=true & ext=false & interruptor=false) Then (0.0) Else (contr pos)*
 Ventosas → *ventosas=If (mando=true & interruptor=false) Then (true) Else (false)*

Figura 33: Tabla de variables de entrada y salida, y código de los sensores de los cilindros con fallo.

En el caso de la 13ª avería, hubo que realizar una programación diferente. En este fallo, cuando el cilindro horizontal de la verificación de posición debía extenderse, esta acción no se realizaba y, por el contrario, descendía el cilindro vertical de la inserción en el pallet. Es decir, el cableado de los dos actuadores estaba intercambiado. Para simularlo, después de crear las dos señales en los subconjuntos de los actuadores, se añadió este cambio en el bloque de expresión en el que se enlaza el interruptor con estas señales. Para ello, hubo que añadir como entrada ambos mandos y como salidas ambas señales de los subconjuntos. Descrito en el lenguaje de programación (Fig. 34):

VARIABLE	TIPO	E/S	VALOR INICIAL	COMENTARIO
interruptor	bool	entrada	false	Estado del interruptor de la avería
cil horiz	bool	entrada	false	Señal de mando desde el PLC del cilindro horizontal
cil horiz sub	bool	salida	false	Salida del cilindro horizontal
cil vert	bool	entrada	false	Señal de mando desde el PLC del cilindro vertical
cil vert sub	bool	salida	false	Salida del cilindro vertical

Código:

If (interruptor=true & cil horiz=true) Then (cil vert sub=true & cil horiz sub=false) Elseif (interruptor=false & cil horiz=true) Then (cil vert sub=false & cil horiz sub=true) Elseif (interruptor=true & cil vert=true) Then (cil vert sub=false & cil horiz sub=true) Elseif (interruptor=false & cil vert=true) Then (cil vert sub=true & cil horiz sub=false) Else (cil vert sub=false & cil horiz sub=false)

Figura 34: Tabla de variables de entrada y salida, y código de los cilindros con fallo.

2.5 ANÁLISIS DE FALLOS SELECCIONADOS

Para llevar a cabo un análisis más exhaustivo, se seleccionaron varios fallos representativos de todos los métodos de detección y tipo de fallo.

2.5.1 Fallo N°3

En el fallo número 3, el cilindro vertical de la verificación de posición no desciende cuando debiera, ya que el cilindro de extracción de base no ha retornado a su posición original. La causa de dicho comportamiento es el fallo del sensor de fin de carrera de este último cilindro. Por tanto, en el modelo, se debe simular que el sensor siempre se muestra como desactivado ante el PLC. Dado que el fallo ni daña la pieza ni pone en riesgo la seguridad, se aceptará la producción con fallo bajo la supervisión del operario.

Para la detección del fallo descrito, se utilizará el método tiempo de vigilancia. Tal y como se describe en el punto 4.1, se hará un control de la etapa E51 (Fig. 16): si el tiempo de estancia en dicha etapa excede el límite establecido, se activará una variable para reportar el fallo. La información del estado de esta señal se transmitirá al HMI mediante un *byte* donde se codifica el color que debe mostrar, para así indicar el aviso. Para su solución, en paralelo a las condiciones de transición de etapa, se introduce una variable controlada en el HMI por el operario (Fig. 35).

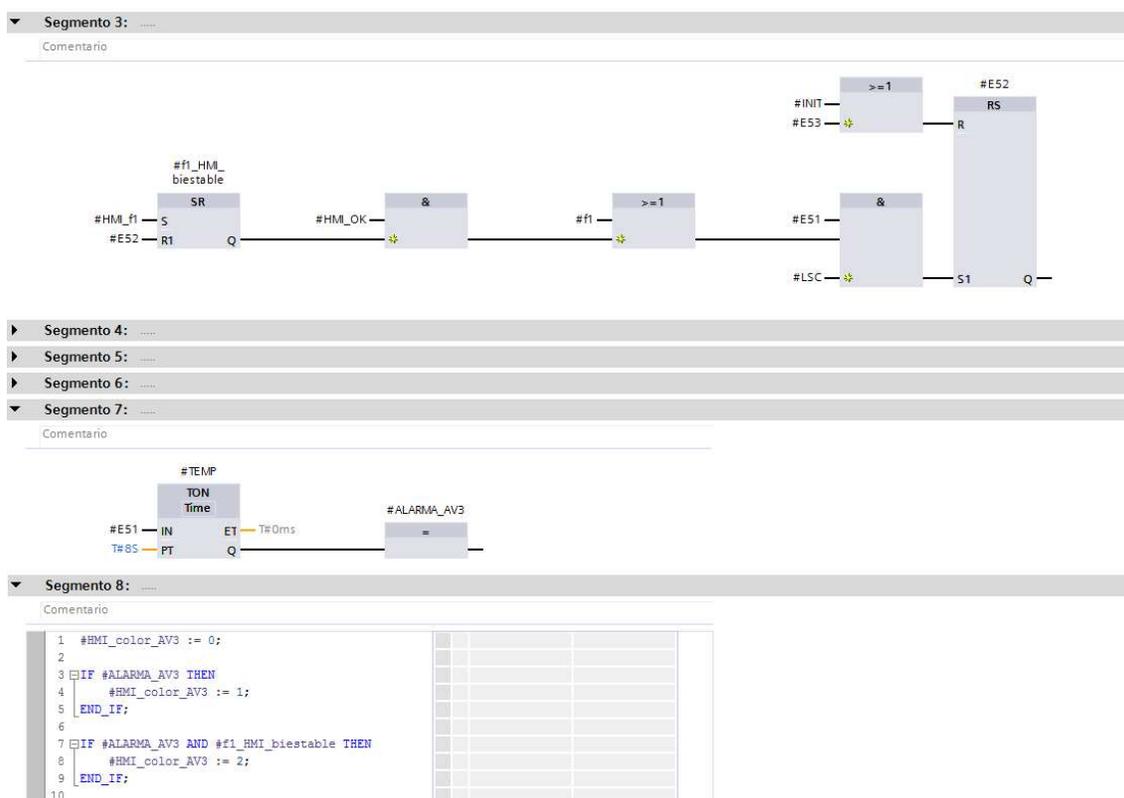


Figura 35: Modificación de código para dar solución a la avería 3.

2.5.2 Fallo N°5

En este segundo fallo, toda pieza es expulsada, aunque su orientación sea correcta. Esto ocurre porque el sensor de fin de carrera del cilindro vertical de la verificación de posición no se activa aunque el cilindro llegue a su final de carrera. Al igual que el fallo previo, el PLC nunca debe detectar dicho sensor como activo. Asimismo, este fallo tampoco daña la pieza ni supone un riesgo para la seguridad; por tanto, se aceptará la producción, en cualquier caso, siempre y cuando esté supervisada por el operario.

La detección del fallo se hará mediante un contador de piezas consecutivas defectuosas. Al detectarse la E62 cuatro veces consecutivas (Fig. 17), el contador activará una variable para avisar del fallo. De nuevo, la información del estado de la señal se transmitirá al HMI a través un *byte* donde se codifica el color que debe revelar, con el fin de señalar el aviso. Al detectar el fallo, se activan dos opciones posibles dependiendo del tipo de pieza (pieza correcta o incorrecta). Seguidamente, el operario deberá seleccionar cuál de las dos es la que corresponde en ese momento, para que así se continúe con la secuencia necesaria (Fig. 36 y 37).

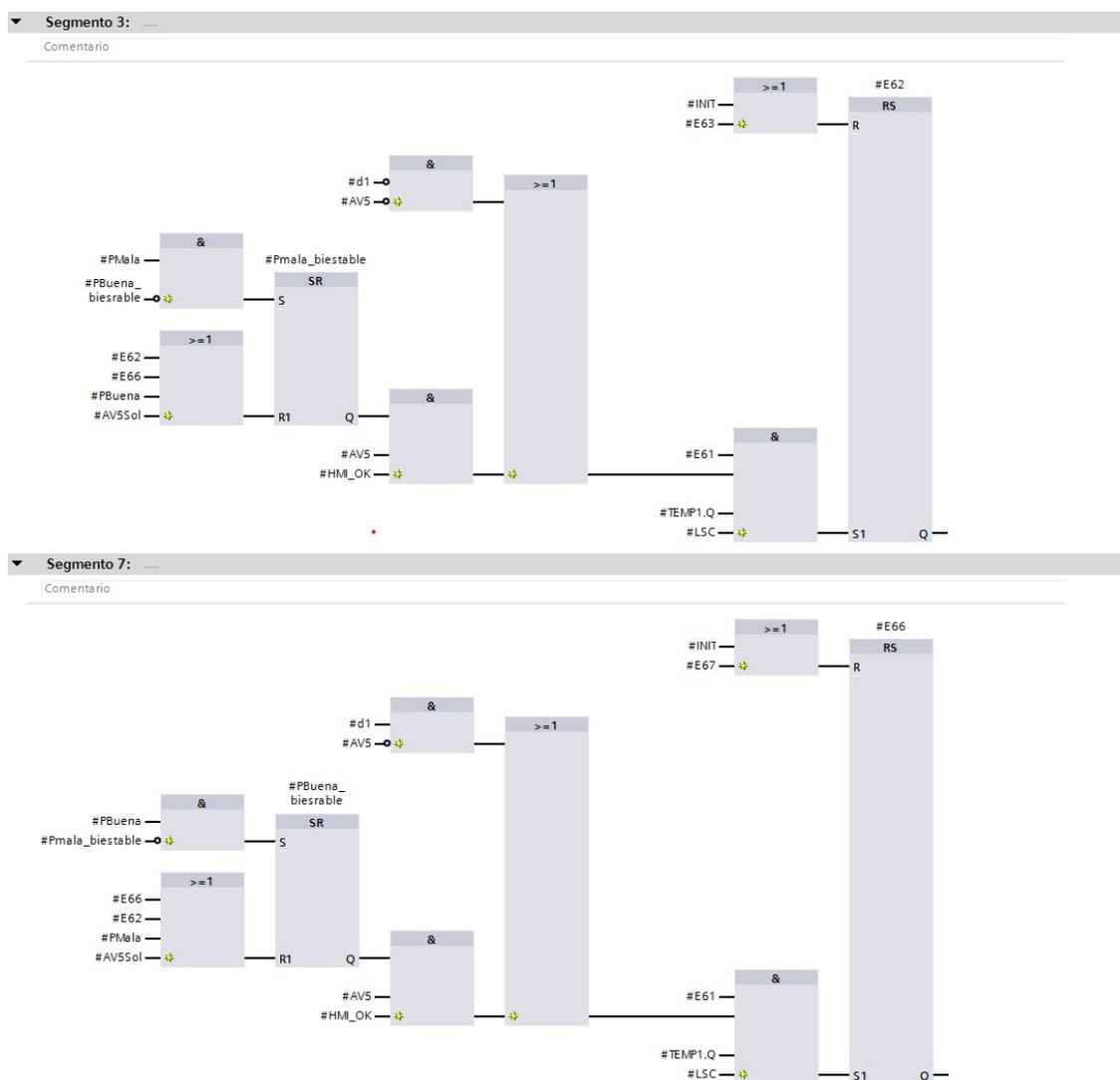


Figura 36: Modificación de código para dar solución a la avería 5.



Figura 37: Modificación de código para dar solución a la avería 5.

2.5.3 Fallo N°9

El fallo número nueve, tras recoger la pieza y ascender el cilindro vertical de la inserción en pallet, este último empieza a descender durante la extensión del cilindro horizontal de la inserción en pallet. En este caso, el componente averiado es el sensor de fin de carrera del cilindro horizontal de inserción en pallet, el cual se activa antes de llegar a la posición. El PLC debe recibir la información de que este sensor está activo; sin embargo, al inicio de cada ejecución, es requisito sine qua non que esté desactivado. Al contrario que en los dos fallos anteriores, cuando el cilindro baja mientras avanza se provoca una colisión con la base, pudiendo así resultar dañada tanto la pieza como la propia base. Es por ello, que este fallo se categorizará como un fallo de emergencia.

La detección se realizará mediante temporizador de mínimo de etapa (Fig. 18). Para ello, al activarse la etapa E74 crearemos un pulso de una duración muy corta, y si durante ese pulso la señal de la etapa se desactiva, un biestable recogerá esa información dando el aviso del fallo. Igual que en los anteriores fallos, la información del estado de la señal se trasladará al HMI por medio de un *byte* donde se codifica el color que debe mostrar, y de esta forma informar del aviso. Nada más detectar el fallo, los actuadores cesarán su actividad lo antes posible y se enviará una señal LSC (descripción en el punto 2.1) para quede imposibilitada la evolución de etapa. De esta manera, se evita la colisión de la pieza. A continuación, el operario pondrá manualmente la posición inicial desde el HMI (Fig. 38 y 39).

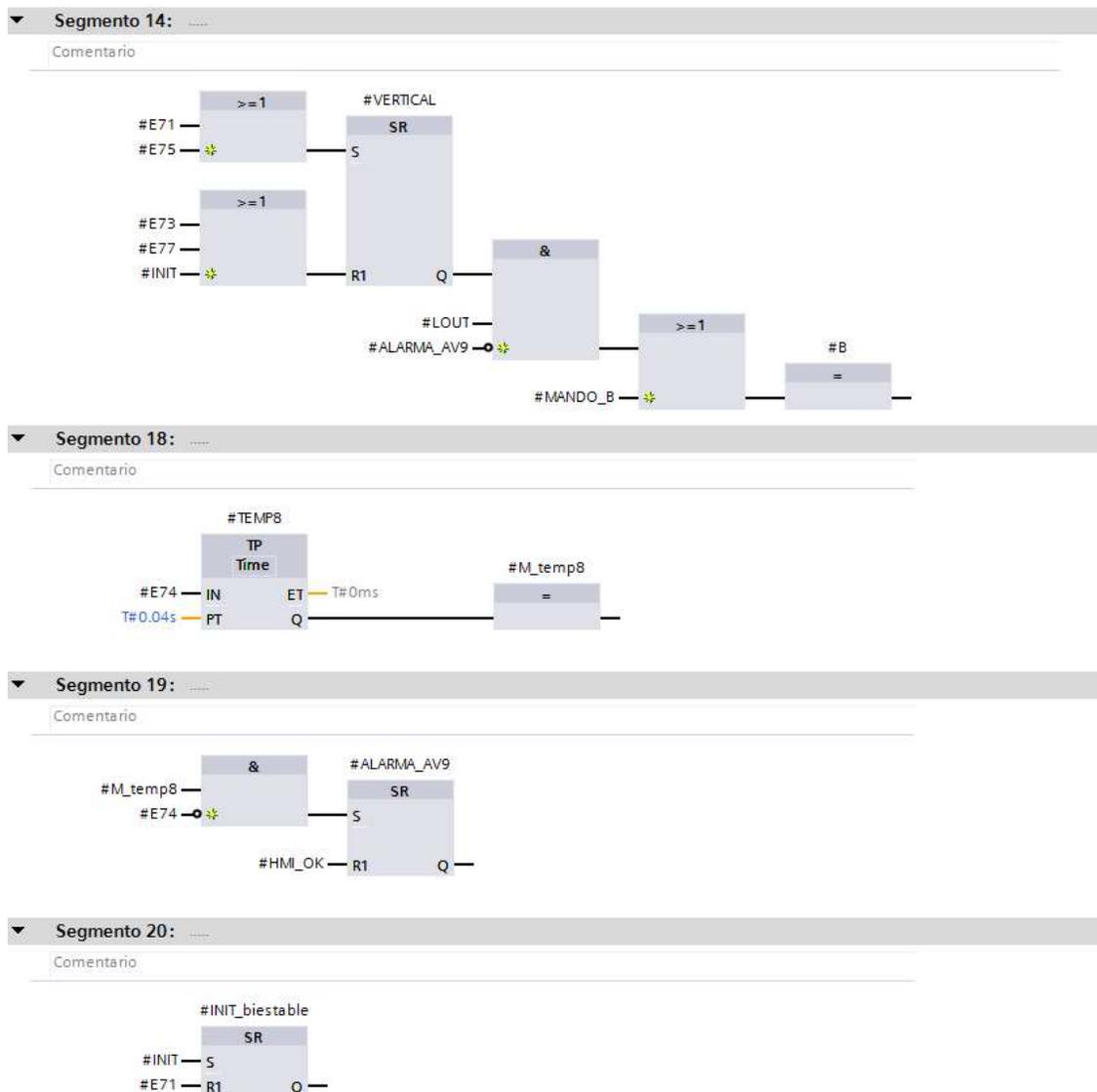


Figura 38: Modificación de código para dar solución a la avería 9.

Segmento 22:

Comentario

1	L	0			0
2	T	#HMI_color_AV9			
3					
4	A	#ALARMA_AV9			
5	JCN	FIN			
6					
7	L	1			1
8	T	#HMI_color_AV9			
9					
10	A	#INIT_biestable			
11	JCN	FIN			
12					
13	L	2			2
14	T	#HMI_color_AV9			
15					
16	FIN:	BEU			
17					

Segmento 5:

Comentario

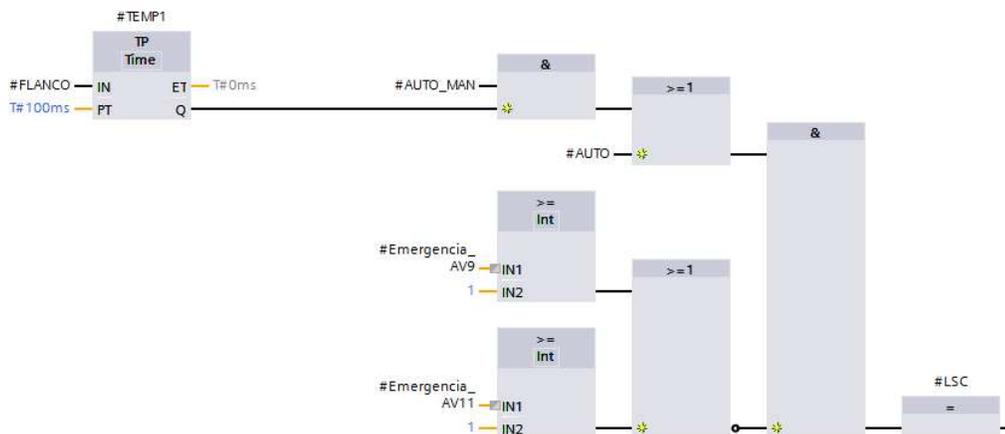


Figura 39: Modificación de código para dar solución a la avería 9.

2.5.4 Fallo N°11

En este caso, el cilindro horizontal de la inserción en pallet avanza mientras el cilindro vertical de la inserción en pallet asciende, provocando así la colisión con el expulsor de la verificación de posición. El elemento averiado es el sensor de inicio de carrera del cilindro horizontal de inserción en pallet, el cual siempre se muestra activo. El PLC ha de recibir la información de que el sensor está activado. Debido a que la pieza puede resultar dañada en la colisión anteriormente descrita, este fallo también se catalogará como de emergencia.

Este fallo es detectado mediante simultaneidad de señales: si tanto el cilindro vertical como el horizontal se encuentran activos a la vez (Fig. 18), un biestable mandará la señal de fallo, ya que nunca deberán estar ambos cilindros activos simultáneamente. De nuevo, se hará uso de un *byte* para transmitir la información al HMI. Tras detectar el fallo, y por su condición de fallo de emergencia, se desactivarán los actuadores y se evitará que avance de etapa. Posteriormente, el operario colocará la máquina en posición inicial a través del HMI (Fig. 40).

2.6 VERIFICACIÓN

Se procederá a la simulación tanto de la detección como de la resolución de los fallos anteriormente enunciados. Así pudiendo permitir la validación de los resultados teóricos esperados con la simulación en el gemelo digital. Para la simulación se usó la previamente explicada técnica de *Software in the loop (SIL)*. Es decir, toda la simulación se realizó en un único ordenador utilizando el software de Siemens PLCSIM Advance para simular el hardware.

Para realizar la simulación primeramente se creó un PLC virtual Siemens S-1500 en el PLCSIM Advanced. La dirección IP que debe tener esta CPU virtual debe coincidir con la del proyecto de TIA Portal y, a su vez, para conectarlo con la herramienta NX-MCD, se nombró de la misma manera tanto a la CPU virtual como al mapeo. Una vez cargado el proyecto en la CPU virtual, y establecida la conexión entre el NX-MCD y el PLCSIM Advanced, es posible hacer la simulación viendo el resultado en el modelo digital en tiempo real. Los resultados obtenidos de dicha simulación satisficieron las expectativas establecidas al comienzo.

2.7 DESCRIPCIÓN DE TAREAS Y DIAGRAMA DE GANTT

Tareas previas

El comienzo del proyecto tiene como fecha el 04/02/2019

Propuesta y objetivos del TFG. Elección del tema del TFG y fijación de los objetivos a completar durante el desarrollo del mismo. Reunión en la fecha de comienzo del proyecto.

Estudio del contexto. Fase de búsqueda de información sobre las herramientas y métodos de trabajo.

Análisis de alternativas. Estudio y elección del software con el que se desarrollará el proyecto.

Familiarización con las herramientas

Debido a que las herramientas a utilizar no se estudian durante el grado, se creó el gemelo digital de una máquina conocida con el fin de conocer el uso del software.

Compra e instalación del software.

Familiarización con célula Festo. Familiarización mediante el modelado de una célula Festo anteriormente vista en la asignatura Automatización industrial. Esta tarea ocupa la mayor parte de este segmento con una duración de 19 días.

Diseño del proyecto

Fase principal del proyecto, en él se recogen todas las tareas para la creación del gemelo digital y la inyección de fallos.

Diseño en Siemens NX-MCD. Creación del gemelo digital, tanto de sus movimientos como sus restricciones, y creación de señales necesarias para el control mediante PLC.

Adaptación del código de control. Realización de los ajustes necesarios en los parámetros del código para ajustarlo al proyecto actual.

Inyección de errores. Introducción de los errores en el modelo de NX.

Modelado. Diseño del modelo capaz de interactuar con el operario, ensamblaje de este con el resto de la estación.

Tratamiento de señales. Creación de bloques de expresión y señales para imitar los errores en la simulación.

Verificación

Verificación de los resultados en la detección y respuesta ante los fallos, realizada mediante simulación de *software in the loop*.

Documentación

Redacción del documento en el que se recoge toda la información derivada del proyecto y preparación del documento para su entrega en la fecha 22/07/2019.

En las siguientes ilustraciones se recoge la información en un diagrama de Gantt (Tabla 3 y 41):

Tabla 3: Diagrama de Gantt con las tareas, fechas y duración del proyecto.

	Nombre de tarea	Duration	Start	Finish
1	Inicio del proyecto	0 days	Mon 04/02/19	Mon 04/02/19
2	▲ Proyecto	121 days	Mon 04/02/19	Mon 22/07/19
3	▲ Tareas previas	10 days	Mon 04/02/19	Fri 15/02/19
4	Propuestas y objetivos	1 day	Mon 04/02/19	Mon 04/02/19
5	Estudio del contexto	4 days	Tue 05/02/19	Fri 08/02/19
6	Análisis de alternativas	5 days	Mon 11/02/19	Fri 15/02/19
7	Hito: fin tareas previas	0 days	Fri 15/02/19	Fri 15/02/19
8	▲ Familiarización con las herramientas	15 days	Mon 18/02/19	Fri 08/03/19
9	Compra e instalación del software	2 days	Mon 18/02/19	Tue 19/02/19
10	Diseño célula Festo	13 days	Wed 20/02/19	Fri 08/03/19
11	Hito: fin familiarización con las herramientas	0 days	Fri 08/03/19	Fri 08/03/19
12	▲ Desarrollo del proyecto	74 days	Mon 11/03/19	Thu 20/06/19
13	Modelado del gemelo digital	27 days	Mon 11/03/19	Tue 16/04/19
14	Adaptación del código	15 days	Wed 17/04/19	Tue 07/05/19
15	Análisis de fallos	3 days	Wed 08/05/19	Fri 10/05/19
16	▲ Inyección de errores	29 days	Mon 13/05/19	Thu 20/06/19
17	Diseño y modelado	15 days	Mon 13/05/19	Fri 31/05/19
18	Tratamiento de señales y modificación del código	14 days	Mon 03/06/19	Thu 20/06/19
19	Hito: fin inyección de errores	0 days	Thu 20/06/19	Thu 20/06/19
20	Hito: Fin desarrollo del proyecto	0 days	Thu 20/06/19	Thu 20/06/19
21	▲ Verificación de resultados	7 days	Fri 21/06/19	Mon 01/07/19
22	Simulación de resultados	7 days	Fri 21/06/19	Mon 01/07/19
23	Hito: fin verificación de resultados	0 days	Mon 01/07/19	Mon 01/07/19
24	▲ Documentación	15 days	Tue 02/07/19	Mon 22/07/19
25	Redacción	14 days	Tue 02/07/19	Fri 19/07/19
26	Entrega	1 day	Mon 22/07/19	Mon 22/07/19
27	Hito: fin documentación	0 days	Mon 22/07/19	Mon 22/07/19
28	Hito: fin proyecto	0 days	Mon 22/07/19	Mon 22/07/19

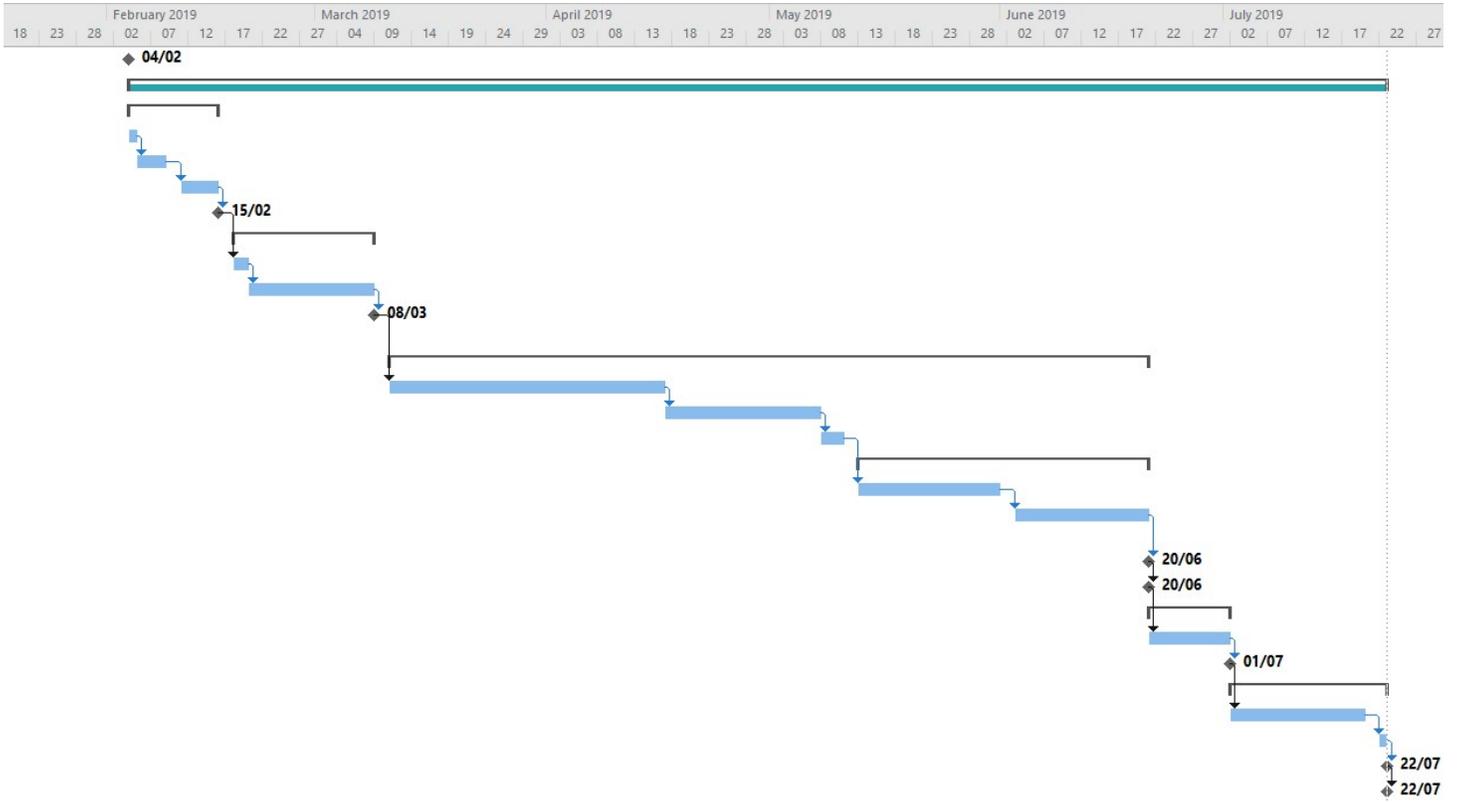


Figura 41: Diagrama de Gantt con las tareas, fechas y duración del proyecto.

3 ASPECTOS ECONÓMICOS

3.1 PRESUPUESTO

En este apartado se realizará el presupuesto necesario para llevar a cabo el proyecto. La mayor parte de este se destinará a las horas internas tanto del alumno como del tutor. Para realizar una aproximación, el precio unitario del alumno ha sido ajustado con la estimación del sueldo de un ingeniero junior, mientras que para el tutor se ha utilizado la estimación de un ingeniero senior. Respecto a las amortizaciones, se han incluido los equipos y licencias que pueden ser usados en otros proyectos. Por una parte, el ordenador, al cual se le ha estimado una vida útil de 5 años siendo utilizado durante dos turnos al día. Por otro lado, las licencias de software tienen una duración aproximada de dos años, debido a que las actualizaciones versiones han de ser adquiridas de nuevo. En lo que a gastos se refiere, se han añadido tanto el material de oficina como el coste de impresión de documentos necesarios para la realización del proyecto. Estos gastos se han estimado en 30€.

La suma de lo anterior forma los costes directos, a los cuales hay que añadir los costes indirectos (electricidad, servicio de limpieza...) estimados en un 8% de los costes directos. Además, se creó una partida del 10% del subtotal para gastos imprevistos, por ejemplo, la compra de periféricos del ordenador en caso de rotura. Por último, se añadieron los costes financieros, estimados en un 5%. El total del proyecto se estimó así en 7.007,73€. A continuación, se muestra la tabla realizada para sintetizar esta información (Fig. 42):

Tabla 4: Resumen de presupuesto.

Horas internas	Nº de horas	Precio unitario	Total
Alumno	200	20 €/h	4.000 €
Director	30	40 €/h	1.200 €
			5.200 €

Amortizaciones	Coste	Cantidad	Nº de horas	vida útil (horas)	Precio unitario	Amortización
PC	1.000,00 €	1	200	13600	0,0735 €/h	14,71 €
TIA Portal	5.000,00 €	1	100	13600	0,3676 €/h	36,76 €
PLCSIM Advance	1.500,00 €	1	50	13600	0,1103 €/h	5,51 €
NX	30.000,00 €	1	150	13600	2,2059 €/h	330,88 €
						387,87 €

Gastos	Coste	Cantidad	Total
Impresión	20 €	-	20 €
Material de oficina	10 €	-	10 €
			30 €

Costes directos	5.617,87 €
Costes Indirectos	449,43 €
Subtotal	6.067,30 €
Imprevistos	606,73 €
Subtotal	6.674,03 €
Costes financieros	333,70 €
TOTALA	7.007,73 €

Observando el desglose se puede apreciar que la mayor parte del presupuesto lo representan las horas internas (Fig. 42). Las amortizaciones también representan una parte importante del presupuesto a causa del alto precio de las licencias. En cuanto a los costes, sin embargo, son prácticamente indiferentes, ya que su impacto relativo es muy pequeño. Por último, los costes indirectos, financieros y la partida destinada a imprevistos tienen un impacto parecido a las amortizaciones.

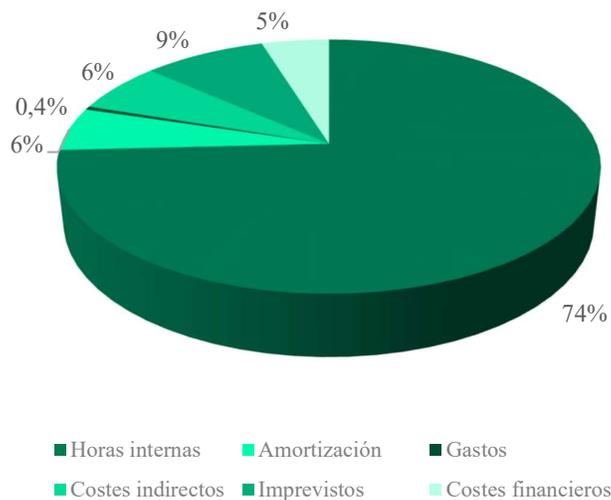


Figura 42: Desglose del presupuesto total indicado en porcentajes.

4 CONCLUSIONES

La primera conclusión que se puede extraer de este proyecto es que la creación de un gemelo digital de la estación FMS-201 plenamente funcional se pudo completar satisfactoriamente. Así, el modelo reproduce los movimientos fielmente, y es capaz de recibir y transmitir las señales. Sin embargo, es posible que, en caso de que los procesos a reproducir fueran más complejos, no se consiguiera una réplica suficientemente adecuada. La razón de dicha limitación es que las herramientas que posee el software NX son finitas; aunque es posible que en futuras actualizaciones del programa se amplíe las posibilidades.

Asimismo, se adaptó el código modular ya creado a las necesidades de la estación y su funcionamiento fue correcto. La estructura modular supuso una complicación más en el proceso de adaptación debido a la gran cantidad de parámetros que deben manejarse, por lo que hubiera sido más sencillo realizar el control en un GRAFCET. Pese a todo, la condición modular de programa permite la implementación en diferentes PLCs, lo que es beneficioso porque no limita su uso al PLC existente hoy en día.

Una vez logrados los objetivos uno y dos, se modificó exitosamente el gemelo digital para la inyección de errores y su simulación, por lo que el modelo puede realizar correctamente los errores conforme a la descripción. Aun así, dada la gran cantidad de elementos necesarios para el correcto funcionamiento, provocó una caída en rendimiento de la simulación.

Como última conclusión puede decirse que la introducción de fallos, así como el desarrollo de métodos para su detección mediante el código fue satisfactoria. Además, el desarrollo de respuestas para la resolución de dichos fallos fue, a su vez, igualmente correcto. Asimismo, pudieron introducirse estos fallos manteniendo la modularidad arriba mencionada.

4.1 LÍNEAS FUTURAS

El siguiente paso natural que debieran realizar las investigaciones futuras, es por una parte el análisis de máquinas más complejas introduciendo elementos como servomotores o robots. Por otra, deberían explorarse otras herramientas, tales como Tecnomatix Plant Simulation o Tecnomatix Process Simulation, las cuales permiten llevar a cabo un análisis más completo del *virtual commissioning*.

En cuanto al software empleado, sería interesante poder ampliar las opciones disponibles para realizar así diseños más complejos y/o precisos. Además, el desarrollo de este tipo de tecnologías permitirá abaratar sus costes de implementación en cada vez más empresas, lo cual provocará un cambio de paradigma en la industria tal y como la conocemos. Por ejemplo, podría llegar a ser tan cotidiano en el futuro mostrar al cliente el producto en simulación como lo es hoy en día presentar los planos del proyecto.

5 BIBLIOGRAFÍA

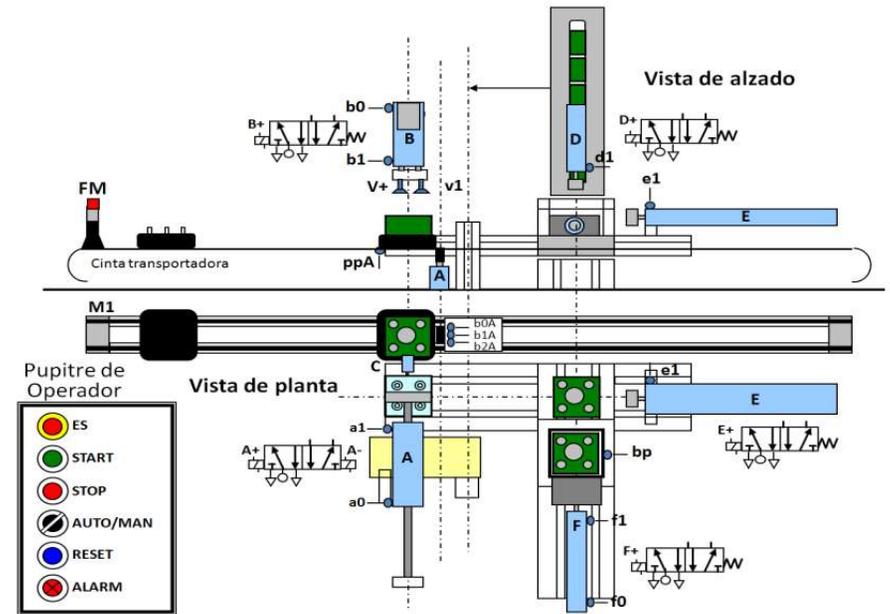
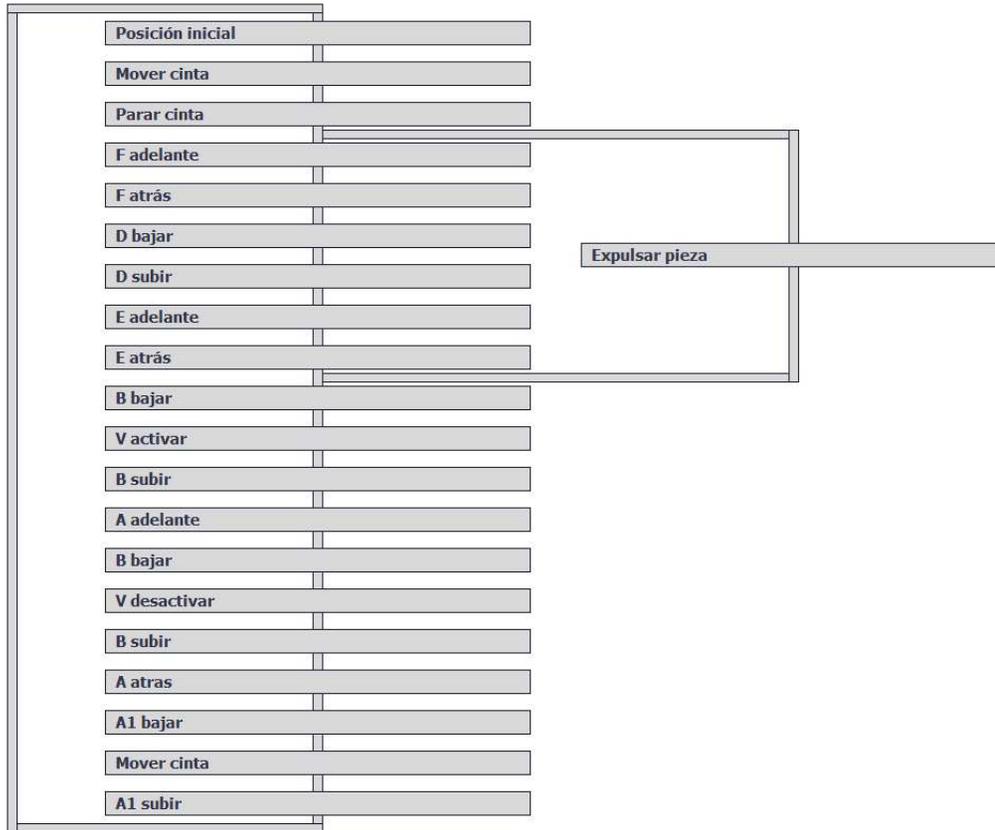
- [1] SMC ® International Training. (n.d.). Configurador de producto y especificaciones técnicas: *FMS-200: Célula de ensamblaje flexible automatizada con transfer lineal*. Recuperado de: <https://www.smctraining.com/es/productlist/productlist>
- [2] Rojko, A. (2017). Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies*, 11, 77-90. doi: g/10.3991/ijim.v11i5.
- [3] Söderberg, R., Wärnefjord, K., Carlson, J. S., Lindkvist, L. (2017). Toward a Digital Twin for real-time geometry assurance in individualized production. *CIRP Annals*, 66, 137-140. doi: 10.1016/j.cirp.2017.04.038.
- [4] Krause, H. (2007). Virtual commissioning of a large LNG plant with the DCS 800XA by ABB. 6th EUROSIM Congress on Modelling and Simulation. Ljubljana, Slovénie.
- [5] Bargiela, A., Ali, S. A., Crowley, D., Kerckhoffs, E. J. H. (2010). *Simulation meets global challenges*. Dudweiler, Germany: Digitaldruck Pirrot GmbH.
- [6] Reinhart, G. & Wunsch, G. (2007). Economic application of virtual commissioning to mechatronic production systems. *Computer Aided Engineering*, 1, 371-379. doi: 10.1007/s11740-007-0066-0.
- [7] Berger, H. (2014). *Automating with SIMATIC S7-1500. Configuring, programming and testing with STEP 7 professional*. Publicis publishing, Erlangen. Berlin, Germany.
- [8] Burgos, A., Álvarez, M. L., Sarachaga, I., Sainz de Murieta, J. (2008). *MeiA. (Methodology for industrial Automation systems/Metodología para ingeniería de Automatización)*. ETSI Bilbao (UPV/EHU). Bilbao, España.
- [9] Martínez, J. & Díez, J. M. (2011). *Aprenda WinCC*. Universitat Politècnica de València. Valencia, Spain.

Anexo 1

En las Figuras 1, 2 y 3 se muestran diferentes capturas del HMI.



7/20/2019 9:45:02 AM



AVERÍAS

Figura 43: Pantalla inicial del HMI. A la izquierda se muestra el GRAFCET de la estación y a la derecha la vista esquemática de los actuadores y sensores.

Auto Man Start Stop Barra luminosa



7/20/2019 9:45:29 AM

AV1

AV2

AV3 - f1

AV4 - v1

AV5 - d1

AV6

AV7 - e1

AV8 - b1

AV9 - a1

AV10 - a0/A-

AV11 - b0

AV12

AV13

AV14

AV15

AV16

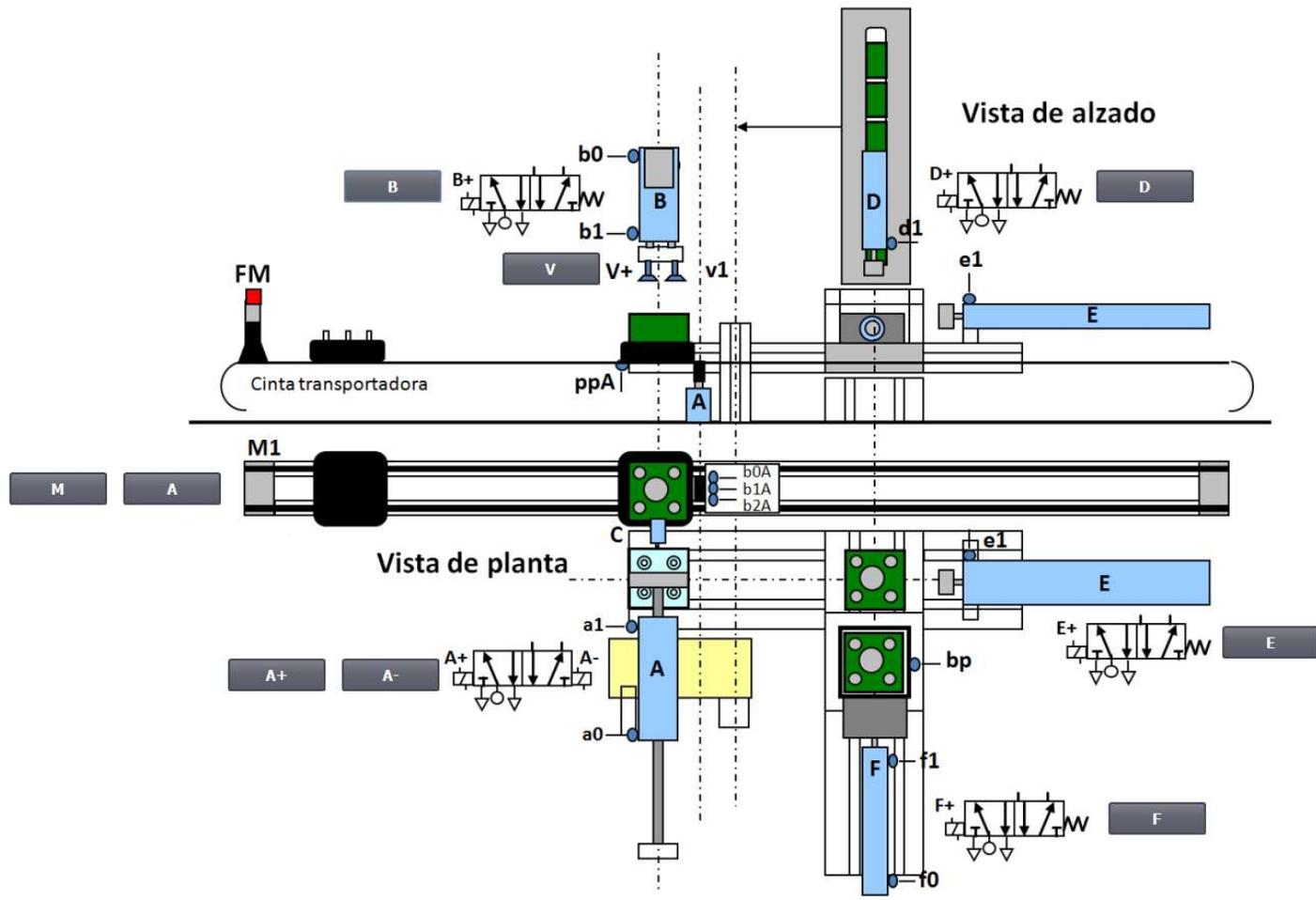
No.	Time	Date	Status	Text	Acknowledge group



INICIO

OK

Figura 44: Interfaz de la detección y resolución de errores.



Posicion inicial

Cerrar

Figura 45: Mando manual para fallos con paro de emergencia.