UNIVERSITAT JAUME I

ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

Uso de gemelos digitales para la validación y verificación de software en el estándar de programación industrial IEC-61131

TRABAJO FINAL DE GRADO

AUTOR/A:

Olga González Gimeno

DIRECTOR/A:

Oscar Miguel Escrig

Agradecimientos

Me gustaría agradecer a todas las personas que me han apoyado durante la elaboración de este proyecto. En primer lugar a mi familia, que aún sin entender muy bien el tema del proyecto, han seguido de cerca su evolución.

En segundo lugar, agradecer a los profesores que me han guiado y han contribuido a mi crecimiento académico. En especial a mi tutor Oscar Miguel Escrig, por su paciencia y numerosas tutorías sin las cuales no habría podido realizar este proyecto.

Por último, agradecer a mi círculo más cercano, en especial a Marina, quien, sin importar la distancia que nos separe, siempre ha estado a mi lado.

Índice

1.Memoria

1.1. Objeto

Durante las últimas décadas, el uso de tecnologías de control y la automatización de procesos se ha vuelto cada vez más común en la industria, evolucionando significativamente desde su aparición en los años 60. Con el fin de controlar los distintos procesos industriales, se introducen los autómatas programables o PLCs (Programmable Logic Controllers), usados para gestionar actuadores (ej.motores) u otros dispositivos en función de la información recibida a través de las entradas (ej. sensores) acorde a la lógica de control que implementan.

Esta creciente digitalización de la industria provoca una tendencia hacia la optimización y el diseño de procesos mediante el uso de tecnologías avanzadas, resultando en una cuarta revolución industrial o Industria 4.0, caracterizada por el uso de máquinas y fábricas inteligentes. Interesa resaltar, entre estas nuevas tecnologías, la utilización de gemelos digitales.

Un gemelo digital es una réplica virtual de un sistema físico, a menudo en formato 3D, que reproduce su comportamiento en tiempo real. A diferencia de las simulaciones clásicas, donde las interacciones están programadas de antemano, en los gemelos digitales, las interacciones entre elementos se basan en leyes físicas básicas, lo que permite una emulación más realista del sistema y la identificación de posibles fallos antes de su implementación en el mundo físico. Esto facilita el estudio del comportamiento del software de forma previa a su implementación en un sistema real.

Para explorar en profundidad cómo los gemelos digitales pueden utilizarse como herramienta para validar y verificar el software de control de sistemas reales, se obtendrá un gemelo digital de un sistema neumático específico (proporcionado por el área de Ingeniería de Sistemas y Automática de la Universitat Jaume I), utilizando la herramienta de simulación Factory I/O. En este programa se recreará virtualmente el entorno del sistema, incluyendo sus componentes y su interacción, en un formato tridimensional inmersivo.

Una vez obtenido el gemelo digital, se procede al desarrollo del software de control necesario para dirigir el comportamiento del sistema. Para esto, se utilizará CODESYS®, un entorno de desarrollo integrado para programar aplicaciones de controlador reconocido a nivel internacional y ampliamente utilizado en la programación de sistemas industriales. Este entorno sigue los estándares de programación establecidos por la normativa IEC-61131.

Cuando se programa un PLC, es común que los principales fabricantes adopten los conceptos del estándar IEC-61131, ampliamente reconocido a nivel internacional como una referencia para la programación de sistemas de control industrial. Entre los cinco lenguajes definidos por esta normativa, el siguiente proyecto se centra en el SFC (Sequential Function Chart) que, como se verá más adelante, permite programar de manera más visual. Previamente a la implementación del programa mediante SFC, se utilizará el lenguaje GRAFCET, definido en la norma IEC-60848, al ser este un lenguaje de modelado gráfico.

El siguiente paso de este proyecto es la utilización del gemelo digital desarrollado para llevar a cabo pruebas del software de control. Esto permite realizar una evaluación precisa del rendimiento del software en un entorno virtual antes de su implementación en el sistema físico real. De este modo, en primer lugar, se verifica el código, asegurándose de que cumple con los requisitos establecidos de forma teórica comprobados en el entorno

virtual. Y, en segundo lugar, se pasa a la validación del código para garantizar que funcione de manera óptima en el sistema real, cumpliendo así con las funciones requeridas.

En este trabajo de final de grado, se pretende discutir las ventajas y limitaciones del uso de gemelos digitales como método para la validación y verificación de software desarrollado según el estándar IEC-61131, así como las herramientas necesarias para su implementación, mediante un ejemplo práctico. Como se ha visto, este método consiste en replicar de la manera más fidedigna posible el sistema en el que se va a implementar el software, de modo que se obtiene una copia virtual en la que se puede evaluar el programa sin asumir riesgos en la realidad. Se busca evaluar la incidencia de los gemelos digitales en la eficacia y la fiabilidad de los procesos de control y automatización industrial, a la vez que se asegura el cumplimiento de los estándares establecidos.

1.2. Alcance

Este proyecto se centra en examinar la eficacia del uso de los gemelos digitales para la validación y verificación de software desarrollado en la norma IEC-61131, además de explorar las ventajas y limitaciones de su implementación en un caso práctico. Para ello se comienza con la puesta a punto del sistema físico proporcionado, para el cual se elabora un programa que permita controlar el funcionamiento de un sistema neumático. Se desarrollará un gemelo digital basado en un sistema físico, y se emplearán las herramientas Factory I/O y CODESYS® para el desarrollo de gemelos digitales, simulación y el desarrollo del software de control.

El gemelo digital se usará como herramienta de detección y prevención de fallos, reduciendo así la probabilidad de fallos al implementar el software sobre el sistema físico. Para el modelado de dicho gemelo se divide el conjunto en distintos módulos, diseñando y programando cada uno de ellos de manera individual. Una vez se haya comprobado que el programa de cada sección o módulo se comporta satisfactoriamente tanto en el gemelo virtual como en el sistema físico, se integran los distintos componentes de software en un único programa para toda la maqueta, cuyo comportamiento se analizará sobre un gemelo digital que agrupe los diversos módulos del sistema real. Además, se realizará un estudio sobre la viabilidad técnica y económica del proyecto.

Ciertos aspectos quedan fuera del alcance del proyecto, como es el diseño del proceso real, tanto en sus aspectos mecánicos como eléctricos, a excepción de la introducción de elementos menores como piezas de repuesto y sensores.

Este proyecto puede usarse como caso práctico de estudio para la aplicación de gemelos digitales en un contexto industrial, que a pesar de haber surgido recientemente, con herramientas como Factory I/O pueden llegar a convertirse en un método efectivo de puesta a punto.

También se puede contemplar su uso en el ámbito educativo ya que permiten a los estudiantes experimentar y comprender los conceptos de automatización de manera más económica y autónoma, dada la reducción de costes materiales al no ser necesarias las maquetas empleadas hasta ahora.

1.3.Antecedentes

1.3.1.Estado actual del uso del estándar IEC-61131

El estándar IEC-61131 es el principal estándar en el ámbito de la programación de autómatas industrialesl. Publicado por la Comisión Electrotécnica Internacional (IEC), esta normativa permite integrar sistemas de control de distintos fabricantes, estandarizando los lenguajes de programación en este campo, permitiendo diseñar e implementar sistemas de producción más flexibles.

Es por esto que dicho estándar es ampliamente utilizado en la industria de automatización para la programación de PLCs. Y es que, aunque los PLCs han mantenido su relevancia gracias a su fiabilidad y capacidad de adaptación, esta longevidad también se ve respaldada por su alta estandarización, que facilita la interoperabilidad entre diferentes marcas y modelos de PLCs, de modo que las empresas pueden confiar en la continuidad de sus sistemas de control a lo largo de los años. Además, existen organizaciones como PLCOpen [7] que dedican grandes esfuerzos a la divulgación, mantenimiento y mejora del estándar.

Los lenguajes de programación incluidos en este estándar se usan para generar el código del PLC y permiten el intercambio de información y bibliotecas entre programadores. Entre los 5 lenguajes de programación dispuestos en la norma, este trabajo se centra en el uso de SFC, ya que permiten la programación de secuencias de código de complejidad variable de forma visual y sencilla.

Los SFC en el estándar IEC-61131 [4] se basan en un conjunto de elementos gráficos estructurales como pasos, transiciones, ramificaciones y puntos de entrada y salida. Estos elementos se utilizan para definir las secuencias de operaciones, las condiciones de activación y las acciones a realizar en cada etapa del proceso. Permiten representar y controlar de manera visual las secuencias de operaciones de un sistema, lo que facilita el diseño, la depuración y el mantenimiento de los programas de control. Además, los SFC brindan una estructura clara y lógica para organizar las etapas de un proceso y las condiciones de transición entre ellas.

1.3.2.Estado actual de los gemelos digitales

Los gemelos digitales son el resultado de la evolución de la simulación, que ha existido durante siglos. El concepto moderno de los gemelos digitales se atribuye a Michael Grieves, quien lo presentó en 2002 durante una conferencia de la Asociación de Ingenieros de Fabricación en la Universidad de Míchigan [1]. Aunque en ese momento se presentaba bajo el título “Concepto Ideal para el PLM”, se incluían todos los elementos del gemelo digital.

Un gemelo digital consta de un espacio real, un espacio virtual, un enlace para el flujo de datos del espacio real al virtual, y viceversa. Estos dos últimos elementos son lo que diferencia al gemelo virtual de una simulación tradicional. A diferencia de una simulación tradicional, donde el sistema virtual replica el comportamiento del sistema físico bajo condiciones preprogramadas, un gemelo digital va más allá al permitir no sólo la reproducción del comportamiento en tiempo real, sino también el intercambio de información entre ambos sistemas, permitiendo una réplica del comportamiento en tiempo real. Mientras que en una simulación el comportamiento y la interacción de sus elementos están previamente programados por el creador del entorno de simulación, en un los gemelos digitales la interacciones entre elementos no están definidas para una configuración

concreta, ya que sólo se implementan leyes físicas básicas que permiten la evolución natural de un sistema, capaz de emular las simplificaciones y casos no considerados por quien programa el simulador. De este modo se recrea un entorno más cercano a la implementación real del sistema.

Inicialmente, los gemelos digitales se aplicaron a productos industriales físicos, pero con el tiempo su alcance se ha ampliado para abarcar una amplia gama de productos y servicios, llegando incluso al campo de la medicina donde se usan para recrear las condiciones biológicas de los pacientes [2]. Gracias a la evolución de los gemelos digitales, estos no solo se usan para representar entidades tangibles sino que también se utilizan para representar procesos e ideas abstractas.

A pesar de las limitaciones tecnológicas, los gemelos digitales han demostrado su versatilidad y aplicabilidad en diversos campos. Hasta el momento, el sector de la fabricación ha liderado la adopción de esta tecnología, con numerosas herramientas de CAE (*Computer Aided Engineering*) que cubren todos los aspectos de los procesos productivos. Sin embargo, a medida que la industria avanza en su transformación digital, acelerada por el uso de modelos remotos debido a la pandemia de COVID-19 [3], las empresas más innovadoras están aceptando cada vez más la idea de que los gemelos digitales sean indispensables en todas las etapas de diseño, construcción y operaciones. De este modo, su impacto se extiende más allá del sector industrial, aplicándose en campos como la salud o en la planificación de infraestructuras urbanas.

1.4.Normas y referencias

1.4.1.Normas aplicadas

En la siguiente sección se abordan las normativas aplicadas en este Trabajo de Fin de Grado. En concreto, se van a aplicar la norma IEC-61131, la norma IEC-60848, la guía GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts) y el estándar OPC (Open Protocol Communication). Cada una de estas normativas será presentada con una explicación introductoria.

Norma IEC-61131

La primera versión de la norma IEC-61131 [4] surge en el año 1993, con el fin de reducir las diferencias entre los entornos de ejecución, sistemas operativos y lenguajes de programación de PLCs de las diversas compañías. Esta norma específica trata de estandarizar los conceptos relativos a los PLCs y sistemas de automatización industrial. Se divide en 10 apartados de forma que en el primero se proporciona una descripción general del conjunto de normas, definiendo los términos y conceptos básicos utilizados en la programación de PLC, el segundo apartado define los requisitos generales de hardware, en la siguiente sección se establecen las características y reglas específicas para cada lenguaje.

Estos lenguajes, abarcados en la sección IEC-61131-3, incluyen: Lista de Instrucciones (IL) y Texto Estructurado (ST) como lenguajes de programación textuales, Diagrama de Bloques de Funciones (FBD) y Diagrama de Escalera (LD) como lenguajes de programación gráficos, y Diagrama de Funciones Secuenciales (SFC) como una herramienta de estructuración y lenguaje de programación de nivel superior.

En los demás apartados se detallan varios aspectos como las guías de usuario, comunicaciones, seguridad funcional, la programación de control difuso o *fuzzy*, las

directrices para la aplicación e implementación de lenguajes de programación, la interfaz de comunicación punto a punto para pequeños sensores y actuadores, y el formato de intercambio XML abierto de PLC.

Norma IEC-60848

Aunque la norma IEC-61131 proporciona las herramientas y estándares para la programación de automatismos, para el modelado de los algoritmos de control se recurre a la norma IEC-60848.

Esta norma propone como herramienta de modelado los diagramas de GRAFCET es un estándar que propone una serie de símbolos y reglas específicas para la descripción formal sin ambigüedades de los procesos, sin embargo, no cubre la implementación de la especificación descrita.

En un diagrama grafcet se puede distinguir, como se muestra en la Figura 0, entre elementos estructurales y de interpretación. De modo que los elementos estructurales incluyen:

- **Etapas**, simbolizadas por un cuadrado, pueden estar activas o inactivas. El conjunto de las etapas activas en un momento dado representa la situación del grafcet en ese instante.
- **Transiciones**, representadas mediante una línea recta horizontal, indican que puede haber una evolución de la actividad entre varias etapas. Esta evolución se lleva a cabo mediante la activación de la condición asociada a la transición.
- **Enlaces Dirigidos**, líneas rectas que conectan una o varias etapas con una transición, o una transición con una o varias etapas.

Por otro lado, los elementos de interpretación involucran:

- **Condiciones de Transición**, es una expresión lógica que puede ser verdadera o falsa y que está compuesta por variables de entrada y/o variables internas.
- **Acciones**, indica la actividad a desarrollar en cada etapa.

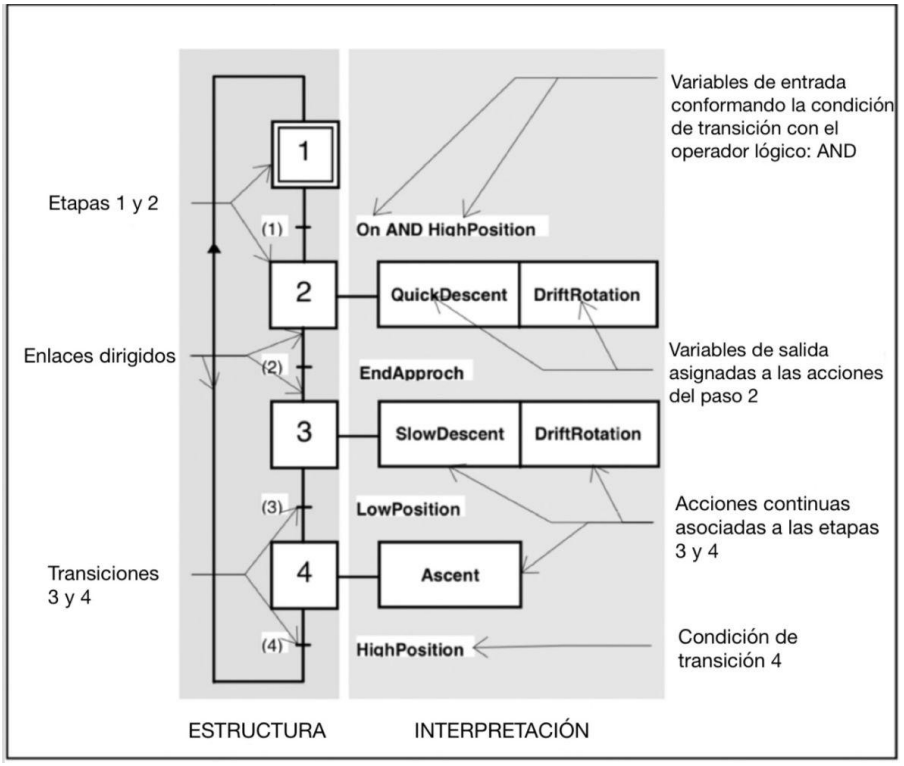


Figura 0. Elementos de la estructura e interpretación de un diagrama grafcet usados para describir el comportamiento de la parte secuencial de un sistema definido por sus entradas y salidas.

Para interpretar correctamente la evolución de un grafcet este se debe leer de arriba a abajo, siguiendo ciertas reglas. En la inicialización de un grafcet se deben activar únicamente las etapas iniciales, marcadas por un doble cuadrado. Para pasar a cualquier etapa, la transición previa debe estar validada (todas las etapas inmediatamente superiores deben estar activas) y se debe cumplir la condición de transición. Además, al franquear una transición todas las etapas inmediatamente superiores serán desactivadas. Si varias transiciones se disparan al mismo tiempo, la activación y desactivación será simultánea, en caso de que esto implique una activación y desactivación simultánea de la misma etapa esta permanecerá activa.

Guía GEMMA

Por otro lado, la guía GEMMA [5] desarrollada por ADEPA organiza de manera exhaustiva todos los modos de Marcha y Parada en los que puede encontrarse una máquina o proceso de producción automatizado. Brinda orientación sobre las transiciones o cambios que pueden ocurrir entre los diferentes modos de funcionamiento.

Uno de los objetivos principales de la guía GEMMA es aplicar una metodología sistemática y estructurada que proporcione información precisa sobre el sistema en términos de los posibles estados. Por tanto, se presenta habitualmente en forma de una descripción completa de todos los estados posibles que el sistema puede alcanzar.

Según la guía GEMMA, una máquina o proceso automatizado puede encontrarse en tres situaciones, en las cuales puede estar produciendo o no: grupo F, en funcionamiento

normal (producción), grupo A., parado o en proceso de parada, grupo D, en defecto, situación en la que puede no estar produciendo o el producto no es aprovechable o solo lo es si se manipula adecuadamente a posteriori.

Estas tres situaciones (funcionamiento, parada y defecto) se representan mediante rectángulos, incluyendo además, un quinto rectángulo para indicar que el sistema productivo está en producción (Figura 1).

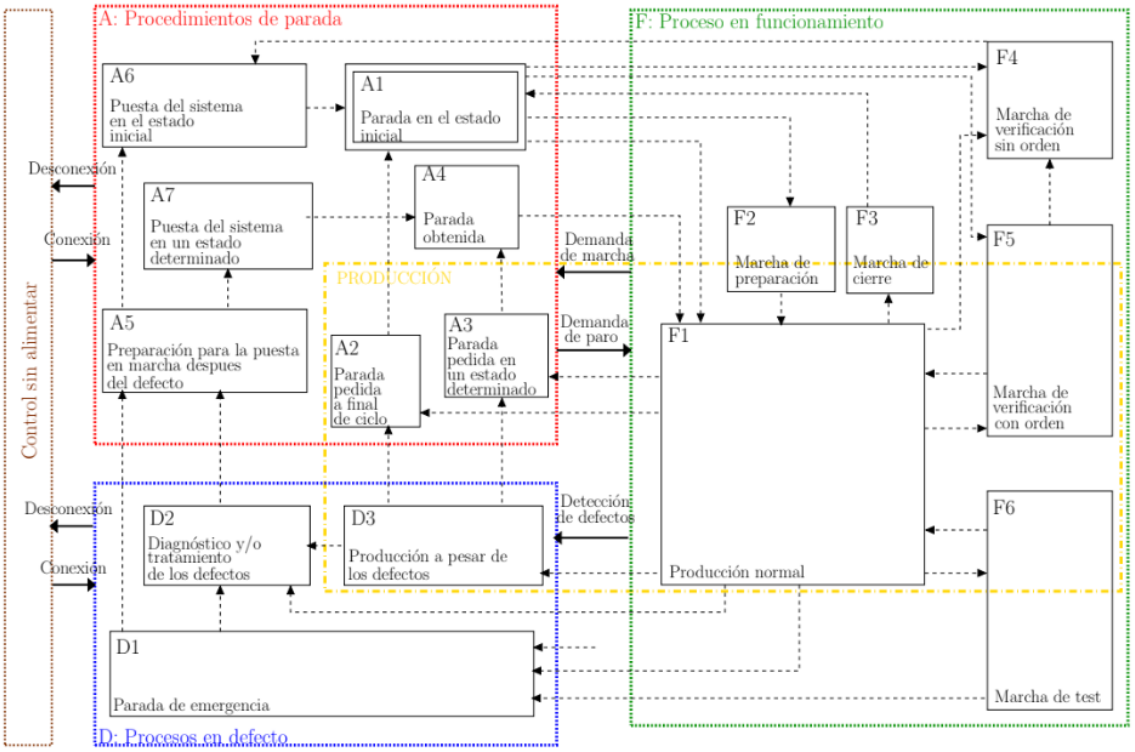


Figura 1.Representación de la guía GEMMA incluyendo sus 16 estados.

Cada una de estas situaciones se subdivide, de manera que la guía GEMMA presenta 16 estados de funcionamiento posibles, de modo que la producción engloba varios estados.

Estándar OPC

Para la comunicación entre los dos programas principales, CODESYS® (para la programación) y Factory I/O (para el modelado y ejecución del entorno virtual), se utiliza el estándar OPC. Este estándar es reconocido por su intercambio seguro y confiable de información en el ámbito de la automatización industrial y la empresa.

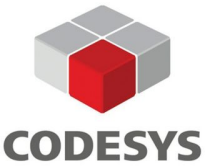
La especificación original de OPC fue desarrollada con el propósito de estandarizar el intercambio de datos entre aplicaciones de software y dispositivos de hardware de automatización industrial. Sus especificaciones definen una interfaz para clientes y servidores, así como entre servidores, lo que permite que componentes del sistema, como PLCs, interfaces hombre-máquina (HMIs) y cualquier dispositivo compatible con OPC, compartan datos sin necesidad de desarrollar aplicaciones personalizadas de interfaz de dispositivo.

Al utilizar OPC, se logra una conectividad abierta entre productos, independientemente de la plataforma de hardware o software utilizada lo que facilita una comunicación fluida y eficiente entre los programas y dispositivos involucrados en el proceso de automatización. Además, proporciona conectividad *plug-and-play*, lo que significa que el sistema informático puede adaptarse fácilmente a los cambios de hardware con una intervención mínima del usuario en la automatización industrial.

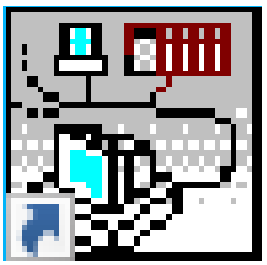
1.4.2.Programas utilizados

Se explican a continuación todos los programas utilizados durante este Trabajo de Fin de Grado, con una breve descripción del uso que se les ha dado.

Factory I/O. Se trata de un programa de simulación 3D con una librería de piezas que recrean las utilizadas en entornos industriales. Con este programa se crea y ejecuta el gemelo digital.



CODESYS®. CODESYS® es el software líder en automatización basado en la norma IEC-61131-3 para el diseño de sistemas de control. Entre los autómatas compatibles con este software se encuentran los utilizados en este proyecto (WAGO), es por esto que se ha utilizado para la programación de aplicaciones,así como para los elementos que las conforman. Para este proyecto en concreto se ha usado la versión V3.5 SP17.30 Patch 3 de CODESYS®.



BootP-DHCP Tool. Este programa ofrece la posibilidad de seleccionar un módulo y asignarle una dirección IP de forma interactiva, lo que se utiliza para poder localizar el PLC conectado mediante ethernet.

1.4.3.Bibliografía

[1] Grieves, Michael. (2016). Origins of the Digital Twin Concept.

[2] Kamel Boulos MN, Zhang P. Digital Twins: From Personalised Medicine to Precision Public Health. Journal of Personalized Medicine. 2021; 11(8):745.

[3] New York, March 20, 2023 (GLOBE NEWSWIRE). "Digital Twin Market - Growth, Trends, COVID-19 Impact, And Forecasts (2023 - 2028)"

[4] IEC 61131-3 International Standard for Programmable Controllers. International Electrotechnical Commission.

[5] ADEPA, “Gemma (Guide d'Etude des Modes de Marches et d'Arrêts),” 1981

[6] Bonfatti, Monari, Sampieri, 1997. IEC-61131-3 Programming methodology.

[7]PLCopen (https://plcopen.org/) [27/07/2023]

[8] Sierra Díaz, Á. L. (2012). MIOOP. Modificación del estándar IEC-61131 para dar soporte al paradigma de programación orientado a objetos. Aplicación al desarrollo del control de procesos industriales.

[9] Grieves, M. W. (2023). Digital Twins: Past, Present, and Future. In The Digital Twin (pp. 97-121). Cham: Springer International Publishing.

[10] "2009 Index IEEE Industrial Electronics Magazine Vol. 3," en IEEE Industrial Electronics Magazine, vol. 3, no. 4, pp. 1-5, Dec. 2009

[11] "Contents," en 2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 2013 pp. 1-19.

[12] IEC. IEC 60848 Ed.2 Specification language GRAFCET for sequential function charts, 2002

[13] Gökalp, E., Şener, U., & Eren, P. E. (2017). Development of an assessment model for industry 4.0: industry 4.0-MM. In Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings (pp. 128-142). Springer International Publishing.

[14] Molina Cortés, D. S., & Alvarino Garzón, J. (2016). Método de programación para PLC'S basado en el estándar IEC61131. Caso de estudio proceso de elaboración de pan.

[15] Yu, W., & Gu, F. (2019). A Review of Modern Industrial Control Strategies: Longevity, Future Trends, and Applicability in Developing Regions. Journal of Control Science and Engineering, 2019.

[16] Gemelo digital: el modelo inteligente de datos, futuro de la edificación. Autodesk. Recuperado de: https://www.autodesk.com/es/design-make/articles/gemelo-digital [27/07/2023]

[17] What is OPC?. OPC foundation. Recuperado de: https://opcfoundation.org/faq/what-is-opc/ [27/07/2023]

[18] Why is OPC important?. OPC foundation. Recuperado de: https://opcfoundation.org/faq/why-is-opc-important/ [27/07/2023]

[19] ¿Qué es la industria 4.0?. IBM. Recuperado de: https://www.ibm.com/es-es/topics/industry-4-0 [14/09/2023]

[20] CODESYS descargas https://www.codesys.com/download.html [13/02/2023]

1.5.Definiciones y abreviaturas

1.5.1.Definiciones

B	
BOOL	Es un tipo de dato que se refiere a variables booleanas. Estas variables pueden tomar únicamente dos valores: 1 (verdadero) o 0 (falso).
E	
Entradas/*Inputs*	Se refiere a las señales de entrada recibidas por el PLC provenientes normalmente de sensores.
Escena	Se trata de simulaciones que se pueden guardar, descargar y compartir dentro del Factory I/O.
G	
GRAFCET	Hace referencia a las siglas *Graphe Fonctionnel de Commande Etape Transition,* y se trata de un lenguaje de modelado que describe la evolución de un proceso automatizado. Nos referiremos al lenguaje empleando su nombre completamente en mayúsculas.
Grafcet	En el contexto de este proyecto se usará para referirse a los diagramas realizados siguiendo el lenguaje GRAFCET. Así, un grafcet es un modelo realizado en lenguaje GRAFCET.
I	
INT	Es un tipo de dato que se refiere a las variables enteras, pudiendo tomar valores enteros tanto negativos como positivos.
M	
Módulo	En este proyecto el término módulo adquiere dos definiciones. 1.Secciones en las que se divide la línea de procesado. 2.Partes que componen el PLC

P

PLC	Se define como un ordenador industrial adaptado para el control de procesos de producción que requieran de alta fiabilidad.

S

Salidas/*Outputs*	Se refiere a las señales de entrada enviadas por el PLC a los actuadores del proceso.

1.5.2.Abreviaturas

C

CC	Corriente contínua	CAE	*Computer Aided Engineering*

D

DA	*Data Access*

F

FIO	Factory I/O	FBD	*Function Block Diagram*

G

GVL	*Global Variable List*	GEMMA	*Guide d'Etude des Modes de Marches et d'Arrêts*
GRAFCET	Gráfico Funcional de Control de Etapas y Transiciones		

H

HMI	*Human - Machine Interface*

I

IL	*Instruction List*	IEC	Comisión Electrotécnica Internacional
IP	*Internet Protocol*		

L

LD	*Ladder Diagram*

M

MAC	*Media Access Control*

O

OPC	*Open Protocol Communication*

P

PLC	*Programmable Logic Controller*	PC	*Personal Computer*

S

SFC	*Sequential Function Chart*	ST	*Structured Text*

U

UA	*Unified Architecture*

1.6.Requisitos de diseño

Como se ha mencionado previamente, en este proyecto se pretende validar y verificar el software de control mediante el uso de gemelos digitales.

Se parte de un sistema neumático (Figura 2) que simula, a escala reducida, una línea de producción como las que se podrían encontrar en la industria, conformada por cuatro módulos o estaciones (distribución, estampado, clasificación y procesado manual). Se dispone también de un controlador PLC que cuenta con 7 módulos de modo que se tienen 14 canales de entrada digital y 10 canales de salida digital, además de un módulo final (sin entradas y/o salidas).

En una primera instancia se revisará el estado del sistema físico, tanto sensores como actuadores, así como el conexionado con el PLC.

Se procede a continuación al diseño del programa de control, siguiendo la norma IEC-60848 para trazar los grafcets correspondientes a cada módulo y al sistema completo, para después implementar mediante alguno de los lenguajes estandarizados (incluídos en la norma IEC-61131-3).

A grandes rasgos, se pretende crear un programa que permita distribuir piezas plásticas y metálicas en orden aleatorio, de modo que al pasar al módulo de estampado se simula el estampado de cada pieza. Tras esto, se descartan las piezas metálicas mientras que las piezas de plástico llegan al último módulo, en el que mediante un robot *pick & place* se recoge una bola y se introduce en la pieza en cuestión.

Los programas implementados tanto para el estudio individual de los módulos como para la maqueta al completo se deben evaluar sobre el gemelo digital previamente a su aplicación en el sistema físico. Con lo que se deben modelar cinco escenas distintas en las que el modelo virtual refleje con la mayor exactitud posible los distintos módulos que componen la maqueta así como la línea de procesado completa.

Tras lograr resultados satisfactorios tanto el sistema digital como en el real, se configura la comunicación de modo que ambos sistemas puedan funcionar simultáneamente, intercambiando información para lograr la sincronía.

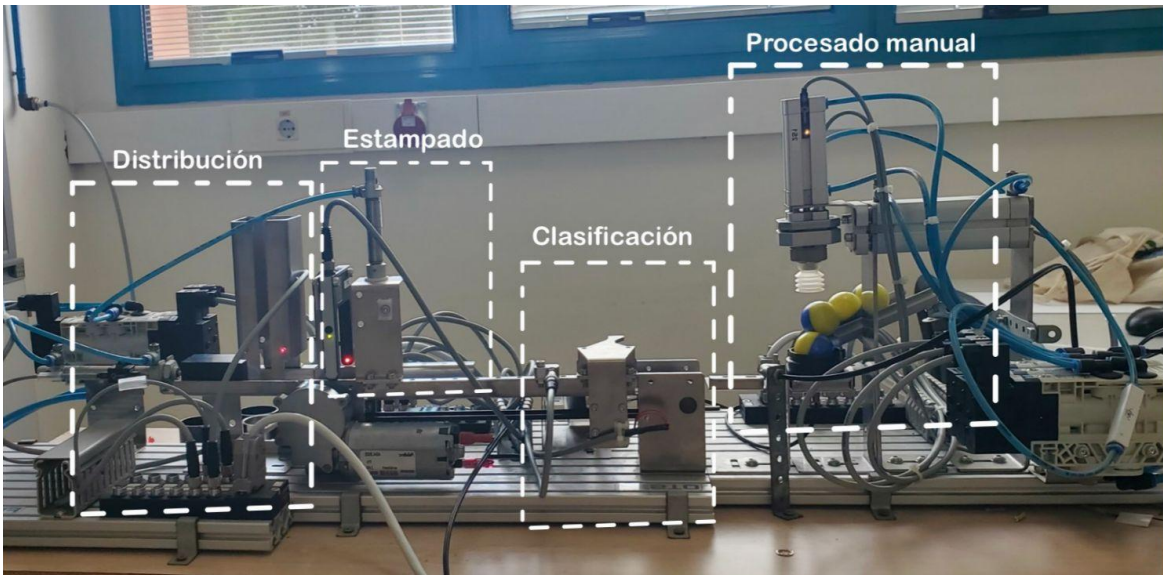


Figura 2. Línea de procesado completa (maqueta real)

1.6.1.Rehabilitación

Previamente a la puesta en marcha de la maqueta se debe comprobar el correcto funcionamiento de todos los componentes, tanto sensores como actuadores, que componen el sistema neumático, reemplazando aquellas piezas que no funcionen adecuadamente.

1.6.2.Requisitos de modelado

1.6.2.1.Modelado del gemelo digital

Se plantea un diseño modular del gemelo digital, que permita la evaluación de cada módulo o estación de manera independiente a los demás, con lo que se requiere crear cuatro escenas, una por cada módulo.

Para la réplica modular del sistema físico es necesario tener una idea clara de los componentes presentes en la maqueta. Clasificados según el módulo al que pertenecen, estos son:

- Distribución. En este módulo se encuentra un pistón de doble efecto y un detector de presencia.

- Estampado. Este módulo cuenta con un sensor de arco y un pistón de simple efecto.
- Clasificación. El módulo de clasificación cuenta con un sensor inductivo, un detector de presencia y una palanca. Además, se considerará la cinta transportadora como parte de este módulo.
- Procesado manual. Este módulo contiene dos detectores de presencia y un robot *pick & place*, compuesto por dos pistones neumáticos.

Se quiere obtener también una última escena que incluya la línea de procesado completa, ensamblando los módulos descritos en única escena, para la evaluación del programa con interacción entre los módulos.

1.6.2.2.Modelado en GRAFCET

El modelado en GRAFCET se divide en dos secciones, siendo estas el modelado de las distintas estaciones o módulos que componen la línea de procesado, y la coordinación entre estos según la guía GEMMA.

Módulos

En lo que respecta a los requisitos de control, la aplicación a programar debe ceñirse a las siguientes condiciones:

- **Distribución.** En esta primera etapa, mostrada en la Figura 3, se suministran los discos que se encuentran en la torre de almacenamiento mediante un cilindro neumático de doble efecto situado al inicio de la plataforma. Para esto primero se comprueba el estado de dicho pistón, recogiéndolo en caso de estar extendido. Tras esto, se extiende por completo llevando a la pieza a la siguiente etapa (estampado).

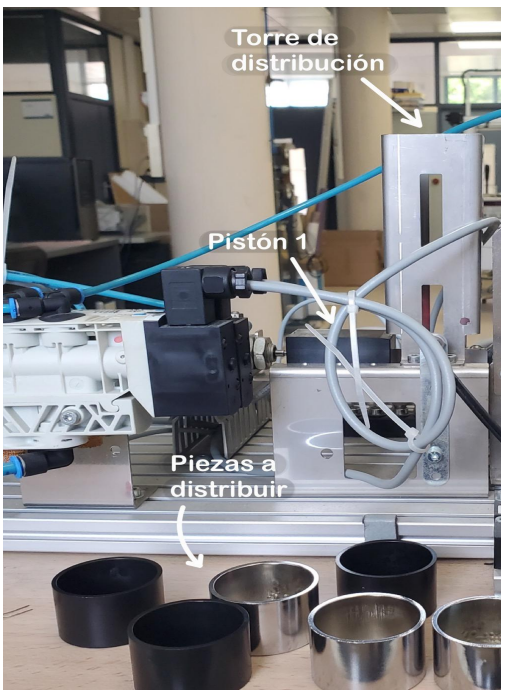


Figura 3. Estación de distribución.

- **Estampado.** La pieza llega hasta el sensor de arco, un detector de barrera colocado al inicio de la cinta, que al detectar presencia pone en marcha la cinta, y que al dejar de detectar presencia (dejándola bajo la unidad de estampado) para la cinta y hace descender un pistón neumático de simple efecto que simula el estampado de la pieza. Una vez retraído el pistón, si no se percibe la presencia de una pieza en la última etapa (procesado manual), se volverá a poner en marcha la cinta para que la pieza pueda ser clasificada. En la Figura 4 se puede observar el módulo de estampado con sus componentes.

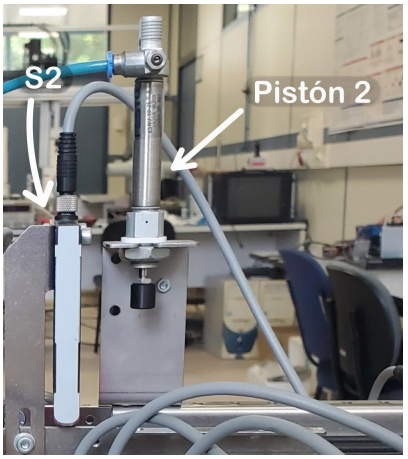


Figura 4. Estación de estampado.

- **Clasificación.** En esta etapa, mostrada en la Figura 5, se separan las piezas en función de su material, aprovechando la capacidad del detector inductivo para distinguir entre piezas metálicas y aquellas fabricadas con plástico. Al activarse el sensor (al detectar metal), se activa la palanca, un mecanismo compuesto por un electroimán y una pieza metálica, de forma que al activarse el electroimán la pieza desciende, desviando las piezas hacia una rampa metálica. La palanca permanecerá activa hasta detectar un impulso de subida en el detector situado al final de la rampa metálica por la que se deslizan las piezas que han sido descartadas. En este caso, esta sería la última etapa del ciclo. Por otro lado, si la pieza es de plástico, no es detectada por el sensor inductivo y sigue circulando por la cinta con normalidad hasta llegar a la etapa final del proceso.

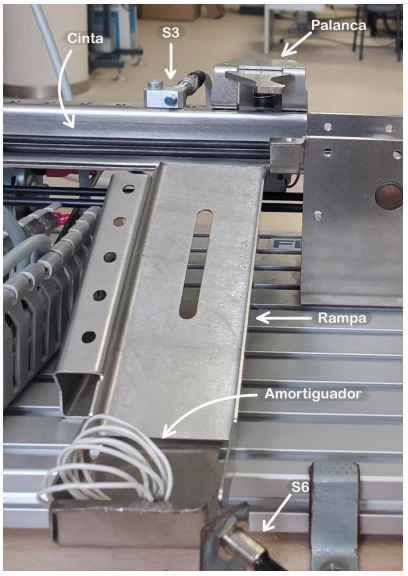


Figura 5. Estación de clasificación.

- **Procesado manual.** Un detector confirma la presencia de una pieza cuando esta alcanza el final de la cinta. A continuación se comprueba mediante otro detector la presencia de una bola en el punto de recogida de bolas. En caso de detectar una pieza en la base de procesado manual pero no una bola en el punto de recogida, simplemente se esperará a que esta sea repuesta. Por el contrario, si hay bola en el punto de recogida, se pone en funcionamiento la secuencia de movimientos de los cilindros neumáticos y la ventosa de absorción que conforman el robot *pick & place*. De esta forma se recoge la bola de su posición inicial y se lleva hasta depositarla en el interior de la pieza de plástico, para después retraer los cilindros neumáticos hasta su posición original. Dado que el dispensador de bolas funciona por gravedad, siempre que haya bolas habrá una en el punto de recogida. Sólo se podrá dar la etapa por terminada una vez se deje de detectar presencia en el primer sensor, ya que esto quiere decir que el operario ha retirado la pieza de la base de procesado manual. Se puede observar el módulo de procesado manual en la Figura 6.

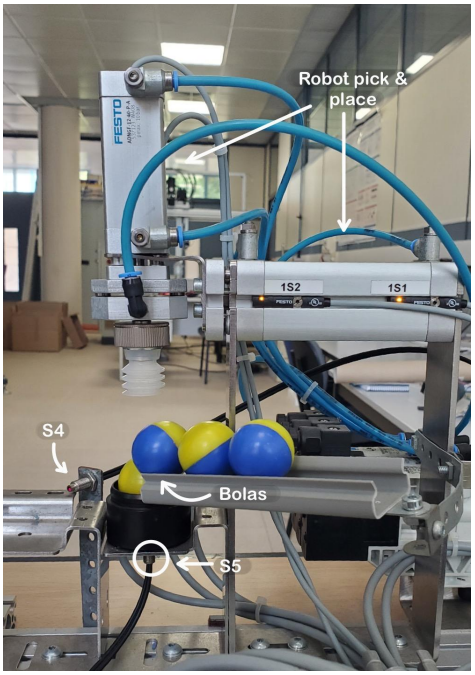


Figura 6. Estación de procesado manual.

Coordinación/alto nivel

Se explican a continuación los modos de funcionamiento relevantes para este proyecto representados según la guía GEMMA, ilustrados en la Figura 8.

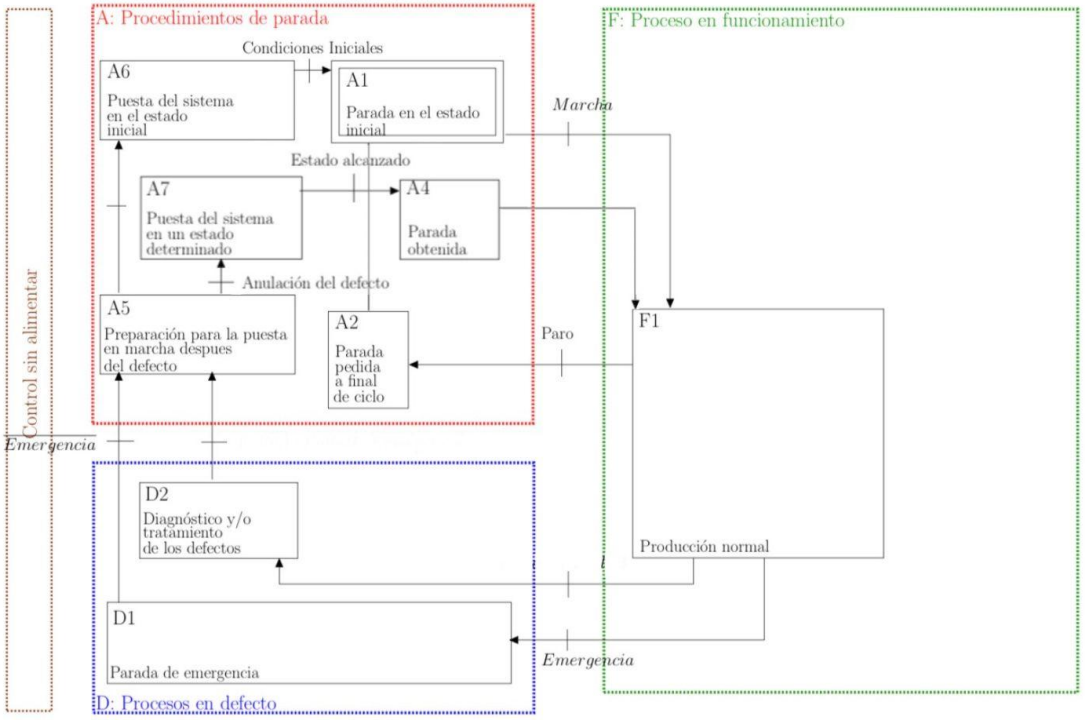


Figura 7. Representación de proyecto en la guía GEMMA

Una vez encendida la máquina, esta permanece en reposo hasta activar el interruptor de marcha. La máquina funciona de forma cíclica, contando con cuatro tareas: distribución, estampado, clasificación y procesado manual, cada una de las cuales se relaciona con las demás atendiendo a diferentes relaciones de causalidad o paralelismo. Este programa, que constituye el proceso de producción normal (F1 en la Figura 7), se ejecutará con normalidad hasta que se desactive la marcha, o hasta que se detecte un fallo, lo que en el diagrama se corresponde con el estado F1.

En caso de desactivarse la marcha, se activa el protocolo de vaciado de la línea, lo que prepara el sistema para entrar en reposo, volviendo al estado inicial a la espera de que se vuelva a activar la marcha. Todo este proceso corresponde al grupo A de los estados de funcionamiento descritos en la guía GEMMA, al estar bajo la sección de “Procedimientos de parada”.

Solo al evaluar el programa simultáneamente en los sistemas físico y virtual, se podrá alcanzar el estado de emergencia o defecto, pudiendo acceder a este de dos maneras distintas. Si se detecta una incongruencia entre la información recibida entre los sensores de la maqueta y del gemelo digital, el sistema se pausa, y el propio usuario debe evaluar y seleccionar de forma manual cómo reanudar el funcionamiento del sistema. Si se trata de un desfase en la recepción de información, el usuario lo solucionará manualmente, y una vez los sensores reciban la misma respuesta de ambos sistemas, se reanudará la marcha con normalidad. Sin embargo, si se trata de otro tipo de fallo, el sistema procederá a efectuar una parada en un estado seguro para corregir los posibles fallos. La detección de dichos fallos se recoge en el grupo D “Procesos en defecto”, mientras que el protocolo a seguir para reanudar el funcionamiento de la línea pertenece al ya mencionado grupo A.

1.6.3.Requisitos de programación

En relación con los requisitos de programación, se utilizará un software apropiado con un lenguaje que permita implementar las aplicaciones de control siguiendo rigurosamente las directrices y estándares establecidos por la norma IEC-61131. Se plantea como requisito la elaboración de aplicaciones modulares, que permitan el control de cada una de las estaciones mencionadas previamente de manera individual. Tras comprobar el correcto funcionamiento de estas aplicaciones modulares tanto en el entorno virtual como en el real, se requiere realizar una aplicación única que posibilite el control simultáneo tanto del sistema físico como de su contraparte virtual. Esta premisa implica un requerimiento esencial: la capacidad de la aplicación para intercambiar información de manera fluida y efectiva con ambos sistemas, facilitando la transferencia de datos en ambas direcciones según lo requiera la operación.

El proceso de programación se llevará a cabo de forma progresiva. Cada programa desarrollado será sometido a una evaluación y análisis de posibles errores en el entorno del gemelo virtual antes de implementarse en la maqueta física. Este enfoque preventivo tiene como objetivo identificar y subsanar cualquier posible fallo antes de que afecte al sistema real. De este modo, se verifica y valida el código siguiendo la metodología en V que se describe más adelante en la sección 1.9.

Una vez que la aptitud del programa ha sido confirmada en el ámbito virtual, se procederá a descargarlo en el sistema físico. Durante esta etapa, se espera que el programa funcione sin dificultades, gracias a la validación previa llevada a cabo en el entorno virtual. A continuación, se dará paso al desarrollo de una aplicación más integral, basada en el programa previamente validado, con la capacidad de controlar ambos

sistemas de manera simultánea y sincronizada. Se debe tener en cuenta que este proceso podría requerir múltiples iteraciones y ajustes para alcanzar un resultado completamente satisfactorio y eficiente en términos de operación y coordinación entre el sistema físico y el virtual.

1.6.4.Requisitos de comunicación

El sistema físico debe comunicar con el programa desarrollado mediante las entradas y salidas del PLC disponible. A su vez, el entorno de simulación Factory I/O debe comunicarse con el PLC para poder llevar a cabo el desarrollo normal de la aplicación.

Para poder establecer una conexión entre el PLC y Factory I/O es esencial evaluar la versión del entorno de desarrollo del programa que se encuentra disponible, consultar qué protocolos de comunicación soporta y, entre estos, determinar cuál es el más adecuado para el proyecto. Dadas las ventajas que ofrece el protocolo de comunicación OPC UA, sería preferible si se pudiera seleccionar como protocolo a utilizar. En caso de que el protocolo OPC UA u OPC DA no estén disponibles en la versión instalada, sería prudente considerar la posibilidad de trabajar con el protocolo de comunicación Modbus TCP/IP como alternativa viable.

Una vez seleccionado el protocolo de comunicación, se debe lograr el intercambio de información con el gemelo digital, con la maqueta, y con ambos simultáneamente, de modo que el gemelo digital y la maqueta puedan comunicarse sin problemas.

1.7.Análisis de soluciones

1.7.1.Soluciones de rehabilitación

El sistema físico proporcionado por la UJI ha sido previamente empleado en otros proyectos de investigación, lo que significa que se partía de un producto ya existente. No obstante, aunque el hecho de contar con un producto preexistente proporciona ciertas ventajas, se requirieron pequeñas adaptaciones para alinear el sistema con los objetivos y requerimientos particulares de la investigación en cuestión.

Examinando la maqueta se observa que no dispone de la junta tórica que compone la cinta, ya que debido a su material se había deteriorado con el tiempo, necesitando un reemplazo. Además, durante la evaluación del programa se hace presente la necesidad de incorporar un conteo de las piezas que abandonan la cinta. Se plantea así la incorporación de un nuevo detector en la maqueta, que permita conocer de manera exacta el momento en que las piezas plateadas abandonan la rampa de descarte. Las diferentes opciones posibles entre los sensores destinados a esta finalidad son:

- Detector de presencia Telemecanique XUD-J003937/ de SCHNEIDER.
- Detector de arco SOOF-P-FL-ST-C50-P/ 553563WS/ 100mA-pnp de FESTO.
- Sensor inductivo SIEN-M5B-PS-S-L/ PNP/ UO W7B/ 150371 de FESTO.

Se concluye también que es necesaria la adición de algún elemento que permita amortiguar la caída de la pieza al ser descartada, evitando la posibilidad de doble conteo y minimizando posibles inexactitudes en el proceso de registro.

1.7.2.Soluciones de modelado

1.7.2.1.Diseño del gemelo digital

Para trabajar con el entorno virtual se escoge el programa Factory I/O, un simulador 3D que permite la conectividad con PLCs de distintos fabricantes mediante protocolos industriales y que, además, cuenta con librerías para trabajar con procesos-tipo ya existentes, así como para crear sistemas propios. Esto proporciona una ventaja sobre otros entornos de simulación que pueden ser 2D, carecer de la facilidad de conectividad de Factory I/O o que conlleven una creación de gemelos digitales más laboriosa en comparación.

El simulador Factory I/O pone a disposición del usuario una serie de sensores, actuadores (incluye los elementos de carga ligera, elementos de carga pesada, operadores, estaciones, dispositivos de alarma), y elementos/piezas (emisor y extractor, pasarelas, piezas), que permiten crear réplicas de sistemas físicos. Sin embargo, no permite la importación o creación de nuevos elementos. Del mismo modo, no es posible modificar las dimensiones de los elementos proporcionados, con lo que la selección queda acotada entre los elementos disponibles. En los siguientes apartados se describirán los elementos utilizados de cada tipo.

1.7.2.1.1. Selección de sensores

En la versión digital, para la selección de sensores se dispone de una variedad de opciones proporcionadas por el Factory I/O, tal como se muestra en la Figura 8. De estos sensores, únicamente serán necesarios los detectores de presencia, los sensores inductivos y las barreras de luz o sensores en arco. Al tratarse de un sistema de eventos discretos, donde la información relevante es de carácter booleano, los sensores usados en su mayoría son detectores en lugar de transductores, a excepción del sensor de arco para el que habrá que convertir la señal. En total, para la construcción del modelo virtual serán necesarios:

- **Sensores difusos**, sensores fotoeléctricos que pueden detectar cualquier tipo de objeto sólido.
- **Sensores inductivos**, se usan para la detección cercana de materiales conductores.
- **Sensores de matriz de luz** o arcos, detecta presencia mediante un conjunto de haces de luz paralelos que crean un campo de detección.

En Factory I/O se incluyen sensores difusos e inductivos que recrean perfectamente el comportamiento de los sensores presentes en la maqueta. A diferencia de los anteriores, el modelo de barrera de luz proporcionado por el Factory I/O es una salida analógica, cuyo resultado depende del número de haces de luz interrumpidos. Este problema se resuelve fácilmente al considerar los resultados como iguales o distintos de cero. Por otro lado, los demás sensores no requieren adaptación adicional, ya que siguen las mismas características que los de la maqueta física.

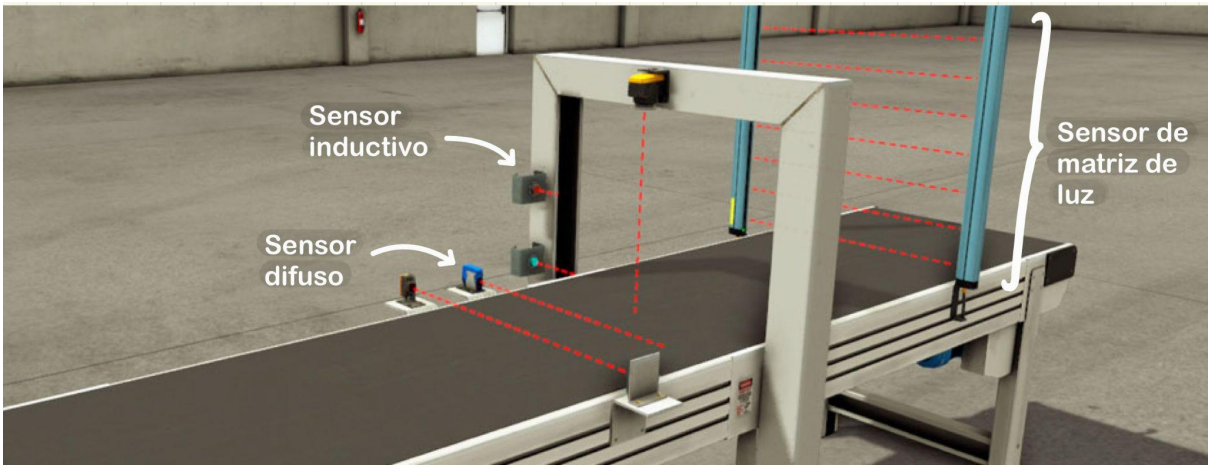


Figura 8. Sensores disponibles en Factory I/O

1.7.2.1.2. Selección de actuadores

Para modelar la maqueta presentada en la Figura 2 son necesarios una serie de componentes:

- **Empujadores neumáticos**, clasificadores equipados con dos sensores de láminas que indican los límites delantero y trasero. Se encarga de distribuir las piezas.
- ***Pick & place* de dos ejes**, como su nombre indica permiten el transporte de elementos en dos ejes (X y Z). Se utiliza para la recogida y colocación de artículos de un lugar a otro.
- **Pistón de simple efecto**, equipados con un sensor de lámina que indica el límite delantero, simula el estampado de las piezas.
- **Palanca o clasificador**, se trata de un desviador con desplazamiento vertical, accionado por un electroimán. Se sitúa tras el sensor inductivo con el propósito de descartar las piezas metálicas.
- **Cinta transportadora de correa**, apta para el transporte de carga ligera. Se usará para transportar las piezas a lo largo del proceso.

A pesar de que estos elementos describen aproximadamente el comportamiento de la maqueta real, dadas las limitaciones de diseño del Factory I/O se han tomado medidas adicionales con el fin de lograr una réplica lo más fiel posible a la configuración original.

Pistones

En la librería del Factory I/O, se encuentran los pistones neumáticos referidos como "empujadores" (Figura 9), los cuales están configurados por defecto como pistones de simple efecto. En consecuencia, se requiere adaptar la programación del pistón de distribución para su correcto funcionamiento en el entorno con el que se comunica, ya sea físico o virtual, dado que en la maqueta física dicho pistón es de doble efecto.

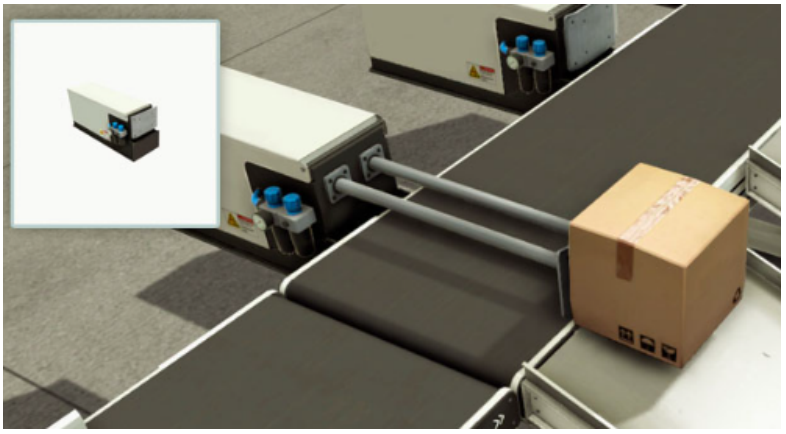


Figura 9. Empujador neumático (PISTÓN 1)

Para llevar a cabo esta adaptación, se establece una relación entre la orden de salida del pistón y la orden de retorno del sistema físico, creando una variable destinada al gemelo virtual que reciba la información de salida y retorno y la transforme en una única señal. Los resultados obtenidos se basan en la Tabla 1, a través de esta tabla de verdad se concluye que para lograr la sincronización entre ambos entornos, es necesario modificar el valor de la variable de salida para el gemelo digital cuando la orden de salida y retorno presentan valores diferentes entre sí. De esta manera, la programación queda definida como se muestra en el Anexo 2.1.

Retorno / Salida	0	1
0	K-1	0
1	1	X

Tabla 1. Estudio del comportamiento deseado en los pistones digitales (en blanco) en función de la información del sistema real (en gris)

En relación al pistón de estampado, no será necesaria ninguna adaptación, dado que se trata de un pistón de simple efecto que no presenta cambios entre el entorno físico y virtual. Sin embargo, en lugar de emplear el pistón neumático utilizado previamente, se optará por utilizar un robot "pick & place" del cual se aprovechará únicamente su capacidad de desplazamiento en vertical. Esta elección se justifica por la mayor similitud que presenta con la pieza real presente en el sistema de procesado.

Palanca

El componente principal del módulo de clasificación es la palanca, una pieza metálica que desciende al activarse un electroimán, desviando la pieza que circula por la cinta transportadora en ese momento siguiendo la ruta predeterminada. Dadas las limitaciones de la librería, no ha sido posible recrear exactamente este mecanismo en el

gemelo virtual, por lo que se ha optado por seleccionar un clasificador de brazo pivotante como alternativa. Este brazo tiene la capacidad de rotar sobre su eje en la dirección elegida, lo que permite empujar las piezas metálicas hacia la rampa designada.

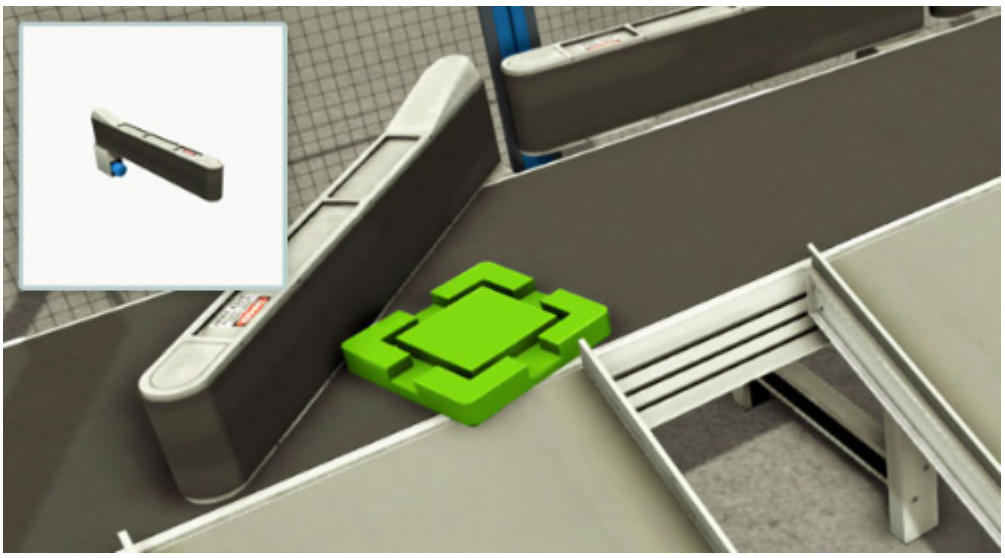


Figura 10. Clasificador de brazo pivotante (PALANCA)

Pick & Place

En la última etapa de la línea se coloca la bola en el interior de la pieza dispensada mediante un robot *pick & place*, que se desplaza mediante movimientos horizontales y verticales. El Factory I/O ofrece en su biblioteca el "*Pick & Place* de dos ejes" que cumple esta misma función. No obstante, se trata de pistones de simple efecto, al contrario que en el sistema real. Para este inconveniente se utiliza la solución aplicada previamente, de forma que la programación resulta como se muestra en el Anexo 2.1.

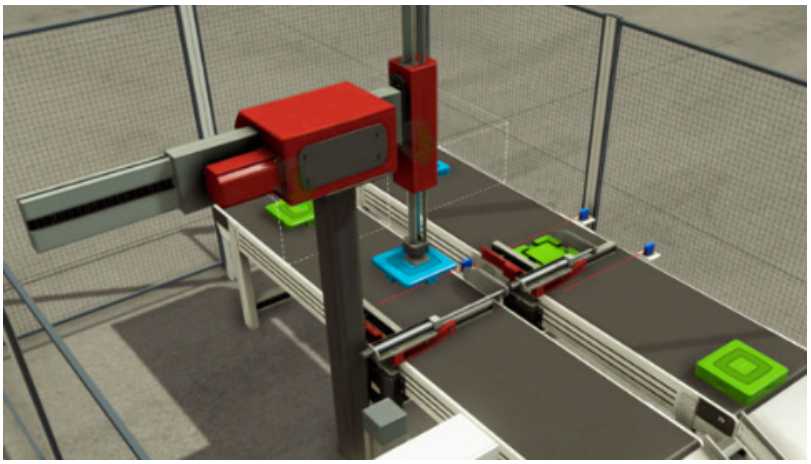


Figura 11. Pick & place de dos ejes

1.7.2.1.3. Selección de piezas y otros elementos

En el sistema físico se dispone de un número limitado de piezas de plástico y de metal, dispensadas al inicio del circuito según estén apiladas en la torre de distribución. Conforme el proceso avanza, una vez que han completado la secuencia de movimientos de selección y ubicación (*pick & place*), o al ser descartadas como piezas metálicas una vez ya han sido detectadas por el sensor situado al final de la rampa de descartes, estas piezas son retiradas de manera manual al final de la cinta transportadora. Para llevar a cabo la simulación de las zonas de entrada y salida de piezas serán necesarios los siguientes elementos:

- **Piezas**, se trata de cilindros huecos, que pueden ser metálicos o de plástico. Además también se utilizan bolas de plástico que se introducirán en el interior de las piezas mencionadas.
- **Rampa o transportador de canal bajo**, se trata de un transportador de tolva recta comúnmente utilizado para despachar elementos.
- **Plataformas**, son estructuras metálicas empleadas para crear suelos a la altura deseada.

A continuación se explica detalladamente el proceso de selección de las piezas a dispensar, así como el razonamiento detrás de la selección de los elementos fijos.

Piezas

Para simular las piezas utilizadas en el sistema real, será necesario disponer de dos tipos de piezas que sean idénticas en forma, pero fabricados con materiales diferentes: unas de plástico y otras de metal.

Inicialmente, se consideró utilizar el elemento referenciado como "materia prima" para estas piezas; sin embargo, durante la simulación se encontró que el proceso en la estación de procesado manual no podía ser replicado de manera precisa. Por ende, se optó por sustituir las piezas distribuidas bajo la etiqueta "materia prima" por las denominadas "base del producto," y las bolas fueron representadas mediante la "tapa del producto". Esta modificación asegura que las piezas encajen correctamente al superponerse, proporcionando así la opción más cercana a la realidad durante la simulación.

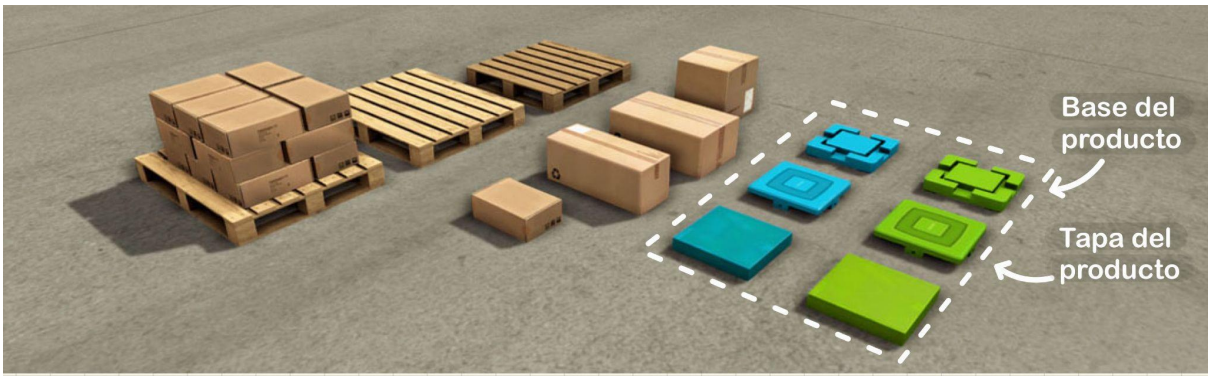


Figura 12. Piezas disponibles para ser dispensadas por el emisor

Emisor

El emisor emite una pieza para ser utilizada en una simulación de forma que no se emiten más elementos mientras todavía haya uno dentro de la zona de emisión de piezas. Se puede elegir qué parte o base emitir, el tiempo entre emisiones, el número de elementos a emitir y si se debe tener en cuenta la posición u orientación aleatoria.

La configuración del emisor no permite seleccionar un orden o patrón para las piezas a dispensar, complicando la sincronización entre sistemas, ya que no se podrá conseguir que el gemelo digital dispense el mismo tipo de piezas que el sistema virtual. Se plantean varias opciones entorno a esta limitación.

En un principio se considera la incorporación de un sensor inductivo que, situado al inicio de la maqueta, permita conocer el material de la pieza seleccionando la pieza a dispensar en el gemelo virtual conforme a la señal recibida. Sin embargo, al haber muchos elementos metálicos en un espacio reducido no se puede obtener una medición fiable hasta más adelante en el proceso, siendo ya demasiado tarde para incorporar la pieza en el proceso virtual. Se pasa entonces a la segunda opción, seleccionar manualmente la pieza a dispensar en Factory I/O en función de la que haya en la maqueta o viceversa.

Dado que cada pieza está asociada a un código en el Factory I/O, se puede editar el código mientras la simulación está en marcha, cambiando así la siguiente pieza emitida para que sea acorde a la de la maqueta. Por otro lado, para modificar la pieza del sistema físico en función de la del gemelo digital simplemente se retiran todas las piezas de la torre dispensadora y se introduce la pieza deseada.

Extractor

Se sitúan dos extractores que retiran el elemento que entra en contacto con la zona delimitada. Se utilizan para simular la retirada de piezas, que en el sistema físico se realiza de forma manual.

Pulsadores

Se introduce, en la última etapa de la programación, un sistema de detección de incongruencias entre las entradas de la maqueta y las del gemelo digital. Se agrega un paso en el que se indica manualmente (por el operario) el motivo del fallo, deberse a diferencias en el número de piezas en cada línea de procesamiento o a retrasos en el funcionamiento de alguno de los dos sistemas. Sin embargo, para evitar las entradas y salidas de los módulos, y teniendo en cuenta que solo se implementan estas funciones cuando está activo el gemelo digital, se añaden dos pulsadores al cuadro eléctrico del proceso virtual, tal y como se muestra en la Figura 13, de modo que cada pulsador está asignado a un tipo de error específico.

Figura 13. Botones disponibles

Interruptor

Solo se requiere un interruptor que indique el estado de MARCHA, con lo que no será necesaria su adaptación al Factory I/O, ya que se pasará esta misma señal al gemelo digital.

Elementos fijos

La maqueta de referencia consta de varios elementos estructurales que pueden ser implementados en el gemelo virtual mediante distintas plataformas y rampas. La librería del Factory I/O proporciona cintas (Figura 14) de 2, 4 y 6 metros de longitud. A pesar de esto, tras evaluar estas tres opciones, se determina que la longitud más adecuada para la cinta sería de aproximadamente 3 metros. Por lo tanto, se agrega un componente adicional denominado "plataforma S" (Figura 16), situado sobre la cinta de 4 metros, de manera que las piezas dispensadas recorran una distancia más corta.

También se incorporan dos plataformas de tamaño S al final de la cinta transportadora que servirán como un banco de apoyo.

En cuanto a la rampa (Figura 15), simplemente se usa el elemento de la librería, ya que el entorno 3D ya tiene implementadas las leyes físicas, con lo que se trata de una caída por gravedad, y su implementación en el gemelo virtual no presenta dificultades, ya que no se ha tenido que programar como se haría en una simulación.

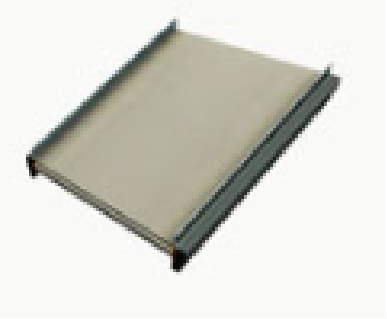


Figura 14. Transportador de tolva baja o rampa

Figura 15. Cinta transportadora

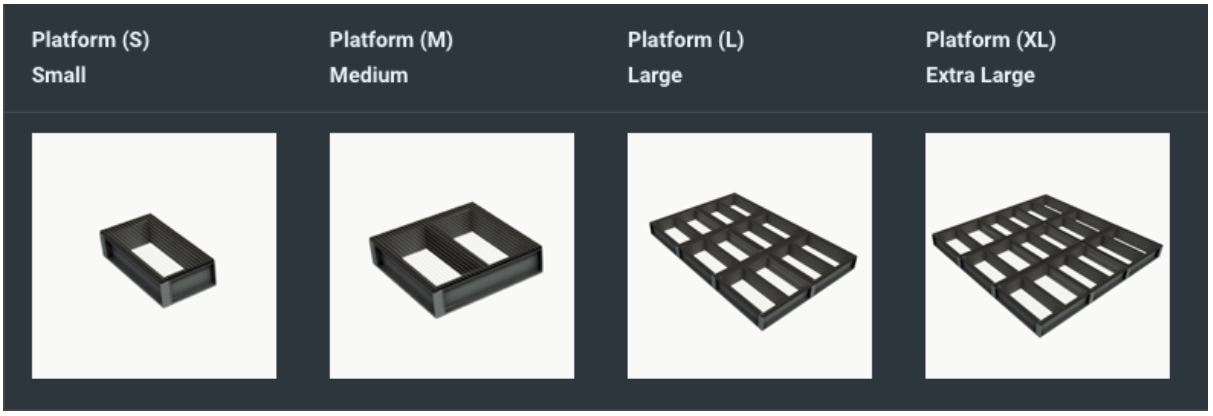


Figura 16. Selección de plataformas disponibles

1.7.2.1.4. Diseño modular

Durante el modelado de las distintas estaciones que componen la línea de procesado se requieren ciertas modificaciones respecto al sistema real. Siguiendo los requisitos de diseño, y utilizando los recursos disponibles en Factory I/O, el módulo de distribución cuenta con unas plataformas en las que se sitúa un empujador neumático, un emisor y un sensor que permita detectar presencia, de modo que su haz quede alineado con la zona de emisión de piezas recreando así la Figura 3.

Así mismo, para el módulo de estampado (mostrado en la Figura 4), se incorpora un sensor que detecte presencia (pudiendo ser un detector, un arco, un sensor retroreflectivo, o un sensor con visión) y, por motivos ya vistos, un robot *pick & place* que sustituye al pistón de estampado, además de una cinta transportadora y un emisor que permiten simular el procesado de las piezas.

Para el módulo de clasificación (Figura 5) se utilizará un sensor que distinga las piezas metálicas, junto con un actuador que permita descartar las piezas, para lo que se considera el uso de un empujador neumático o de un clasificador de brazo pivotante y un sensor de presencia. Se requiere también la incorporación de una cinta transportadora, un emisor y una rampa.

Por último, para el módulo de procesado manual, sobre dos plataformas fijas se sitúa un robot *pick & place*, dos sensores de presencia, junto con dos emisores. Uno de estos emisores simulará la llegada de piezas desde la cinta, mientras que el otro imita las bolas a introducir en las piezas, dando una amplia gama de elementos a distribuir para elegir. El módulo recreado se puede ver en la Figura 6.

1.7.2.1.5. Diseño completo

Aunque durante los ensayos modulares no se encuentran problemas con las escenas diseñadas, al concatenarlas creando el gemelo digital de la maqueta surgen algunos contratiempos.

Se ve que, en caso de dispensar dos piezas muy seguidas, si la primera de ellas es metálica, la palanca que la descarta arrastra ambas piezas al almacén de descartes, lo que supone un gran inconveniente si la segunda pieza es de plástico. Para evitar este fallo, se

plantean dos opciones: estudiar el comportamiento de otros operadores similares a la palanca real y sustituir la seleccionada, o espaciar las piezas aumentando el tiempo de distribución. La sustitución del clasificador de brazo pivotante sería beneficiosa, ya que durante las pruebas realizadas se observa cómo en ocasiones la palanca (debido a su tamaño) desplaza piezas que están siendo procesadas en la estación de procesado manual. Sin embargo, a falta de otro tipo de elemento que se asemeje más a la palanca del sistema real, se opta por espaciar las piezas. Esta adaptación ha sido necesaria para lograr un funcionamiento adecuado del clasificador en el entorno virtual, ya que no se puede replicar de manera idéntica el comportamiento de la palanca presente en el sistema físico, en el que la palanca se desplaza verticalmente en lugar de girar sobre un eje.

En esta misma línea, debido a las limitaciones en la precisión de la réplica virtual, surge una situación imprevista durante la evaluación del sistema completo con los módulos interactuando entre sí. Dado que la cinta transportadora está programada para detenerse bajo determinadas circunstancias, si esta parada sucede justo en el momento en que se ha detectado una pieza metálica pero aún no ha sido descartada, esta pieza podría no ser descartada y seguir circulando. Esto se debe a que la palanca es accionada durante un tiempo limitado tras la activación del sensor inductivo. Para evitar este tipo de fallo se incorpora un detector de presencia al final de la rampa de descartes, como se menciona en el apartado 1.7.1.

Por otro lado, dadas las dimensiones de las piezas dispensadas y el limitado rango de detección del sensor inductivo, se ha modificado su posición para que detecte las piezas desde arriba. Este cambio en la ubicación del sensor inductivo también facilita la adaptación virtual de la palanca del módulo de clasificación, puesto que se pasa a disponer de más espacio entre el final de la palanca y la zona de procesado manual, de modo que permite desplazarla y que no entre en la zona de procesado manual.

Con estas modificaciones se logra obtener una réplica digital del sistema físico que se comporta como es esperado.

1.7.2.2.Modelado en GRAFCET

1.7.2.2.1. Diseño modular

El modelado GRAFCET del programa de control se realiza, en un principio, de forma modular, de modo que permita la implementación y evaluación del software de cada uno de los módulos de manera individual. Se incluyen a continuación cada uno de los grafcets diseñados. Todos ellos parten de una etapa inicial de reposo, de modo que solo se ejecuta el proceso con la activación de la variable marcha.

Siguiendo los requisitos de modelado, para la estación de distribución se obtiene el grafcet mostrado en la Figura 17, encargado de poner en circulación las piezas.

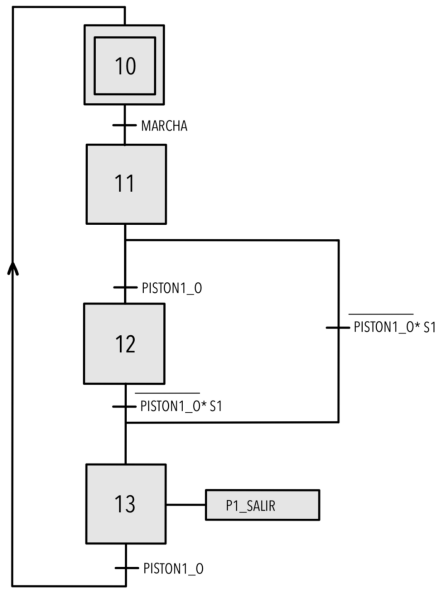


Figura 17. Diagrama grafcet para el diseño modular de la estación de distribución

Al módulo de estampado (Figura 18) se le asigna el control de la cinta transportadora, de manera que esta se desactive durante el estampado.

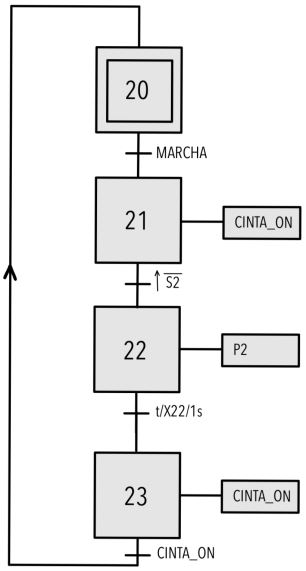


Figura 18. Diagrama grafcet para el diseño modular de la estación de estampado

Sin embargo, para el siguiente módulo de clasificación, se observa en la Figura 19 que no se acciona la cinta en ningún momento, esto es porque se asume una situación en la que la cinta permanece activa, de este modo el módulo controla únicamente la palanca encargada de descartar las piezas.

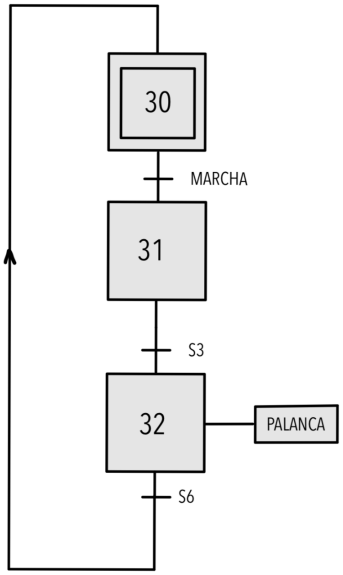


Figura 19. Diagrama grafcet para el diseño modular de la estación de clasificación

Por último se diseña el diagrama grafcet asignado a la estación de procesado manual (Figura 20), en la que se controla el robot *pick & place* situado al final de la línea de procesado.

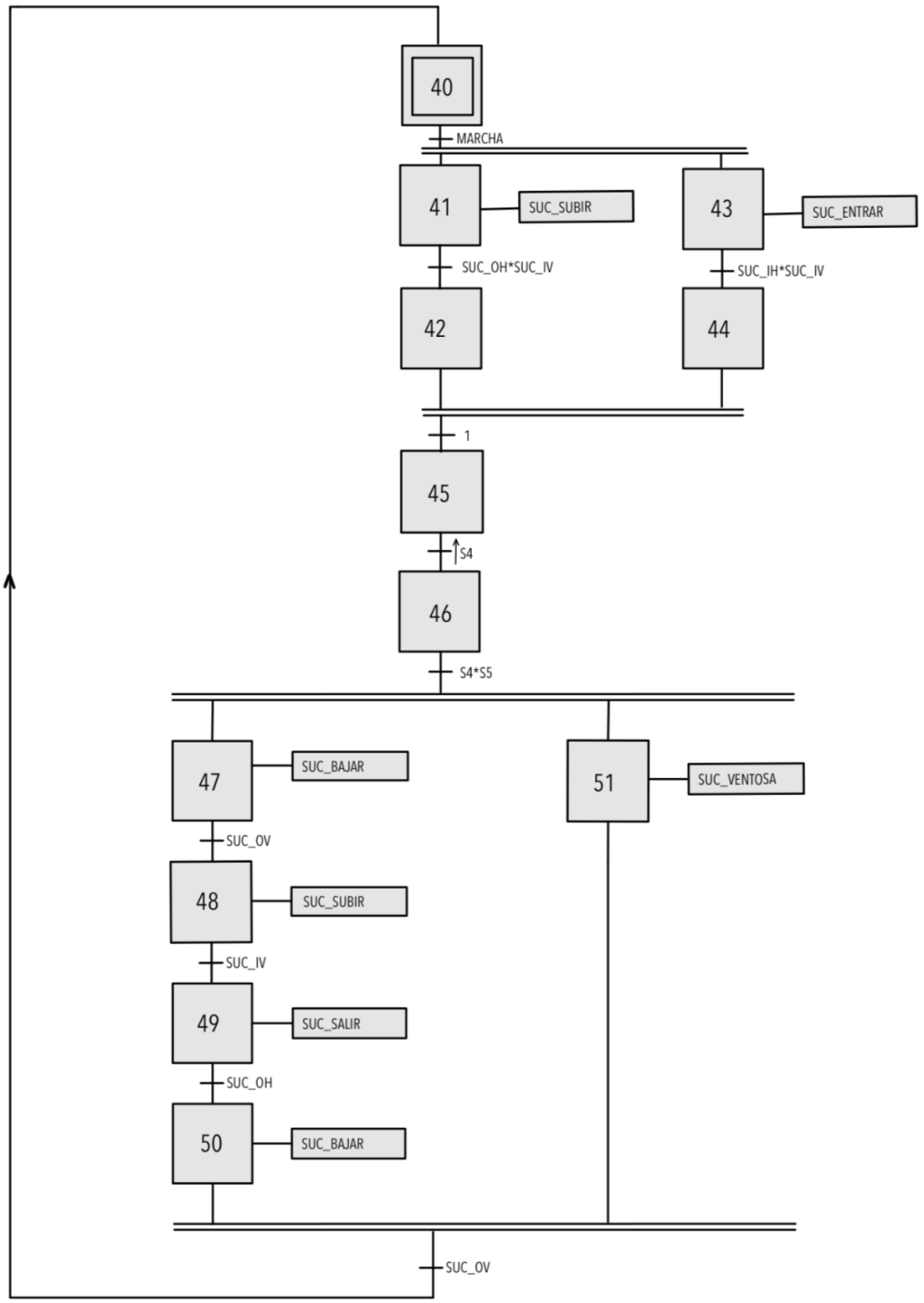


Figura 20. Diagrama grafcet para el diseño modular de la estación de procesado manual

1.7.2.2.2. Diseño completo

Para la integración de los módulos en un grafcet que permita el control de la línea de procesado al completo se plantean dos opciones: la ejecución de los módulos de manera concurrente, o la ejecución simultánea. Se parte de la primera opción, diseñando un grafcet que trabajaría de manera secuencial, lo que implica el procesado de una única pieza a la

vez, de modo que hasta que la pieza no abandone la línea, no se distribuirá una nueva pieza. Para esto se introducen macroetapas que permiten una organización más visual, como se puede comprobar en la Figura 21. El desarrollo de cada una de estas macroetapas se puede encontrar en el Anexo 2.2.1, ya que apenas hay diferencia con los grafcets creados durante el diseño modular.

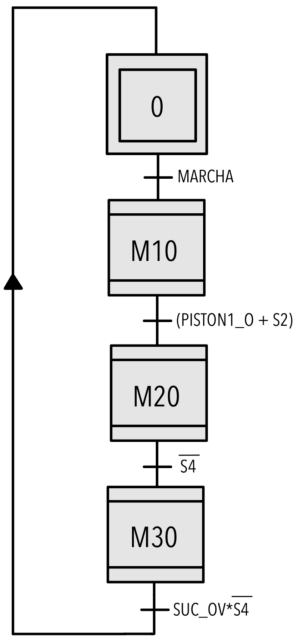


Figura 21. Diagrama grafcet para el diseño secuencial de la línea de procesado

Se observa, sin embargo, que resulta más eficiente el procesado simultáneo de las piezas, con lo que se diseña un grafcet que ejecute los módulos en paralelo. Esto requiere de modificaciones adicionales ya que con esta configuración los módulos interaccionan entre ellos.

Se modela un grafcet maestro que, al activar la variable marcha, ejecutará simultáneamente los cuatro módulos en los que se ha dividido el sistema, como se muestra en la Figura 22.

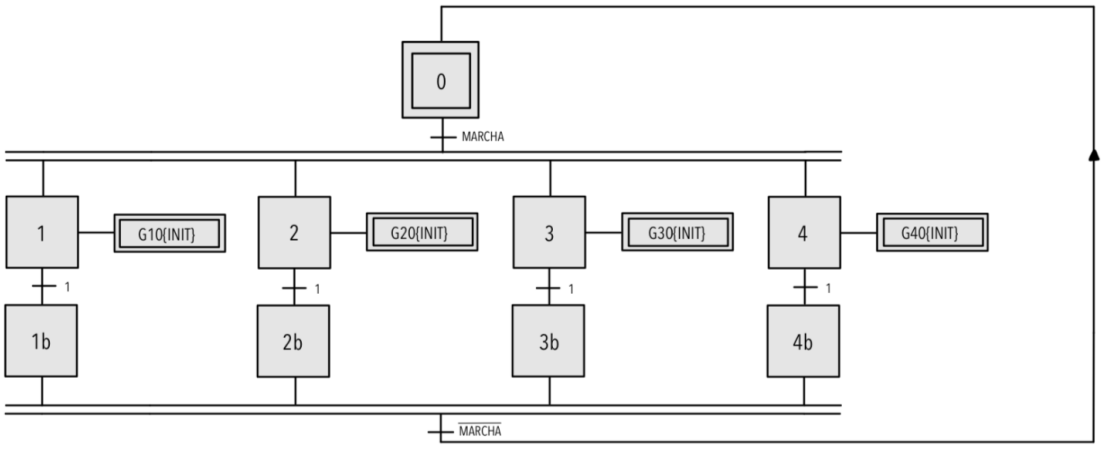


Figura 22. Diagrama grafcet para el diseño en paralelo de la línea de procesado

En el primer módulo, distribución, se incrementará el tiempo de distribución de las piezas para que estas salgan más espaciadas, facilitando el proceso de clasificación que se lleva a cabo en el tercer módulo. Además, se condiciona la salida del pistón para que no se distribuyan piezas hasta que la cinta haya estado suficiente tiempo activa como para evitar la colisión entre la pieza expulsada y la anterior. Se pueden ver las modificaciones resaltadas en la Figura 23.

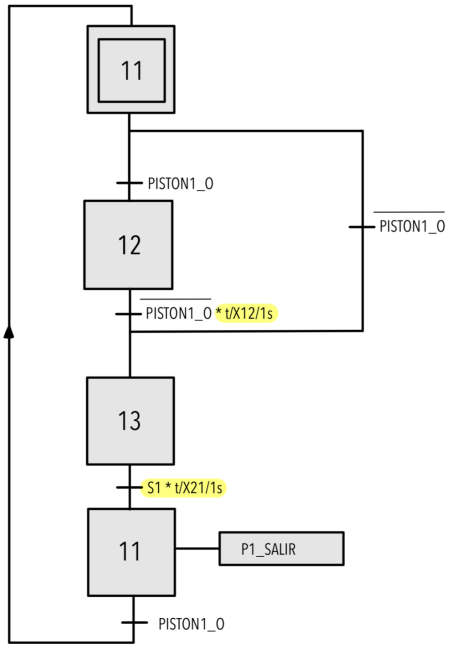


Figura 23. Diagrama grafcet para el diseño en paralelo del módulo de distribución

Como se ha mencionado anteriormente, se le asigna al módulo de estampado el control de la cinta transportadora, con lo que se añadirán nuevos condicionantes teniendo en cuenta las necesidades de los demás módulos, como se observa en la Figura 24.

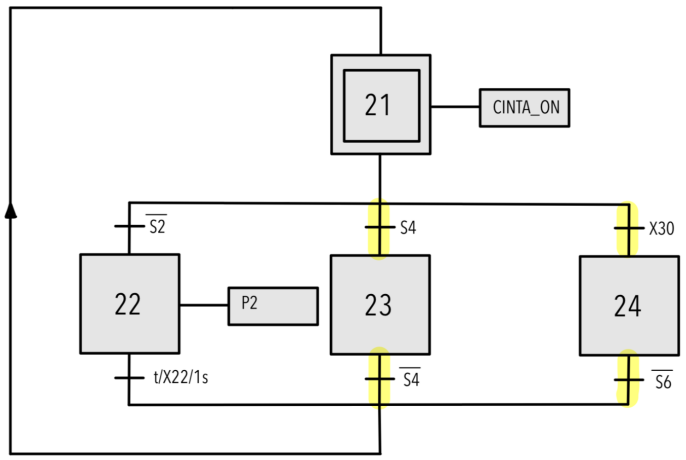


Figura 24. Diagrama grafcet para el diseño en paralelo del módulo de estampado

En el módulo de clasificación (Figura 25) se añade una rama que evite el descarte de una pieza cuando aún no se haya retirado la pieza anterior, ya que esta bloquea el sensor al final de la rampa, que no detectará la caída de una nueva pieza.

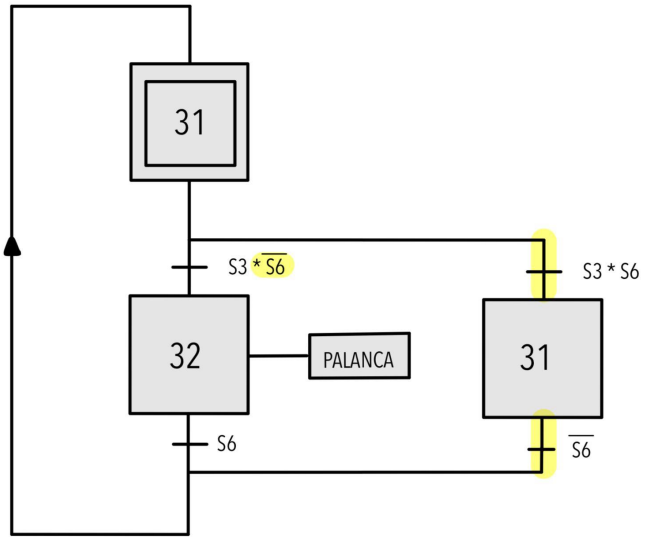


Figura 25. Diagrama grafcet para el diseño en paralelo del módulo de clasificación

El último módulo, procesado manual, no requiere de modificaciones, siendo igual que el de la Figura 20.

1.7.3. Análisis de soluciones de programación

1.7.3.1.Selección del lenguaje de programación

En la Figura 26 se muestra de forma gráfica la funcionalidad de cada uno de los lenguajes, según su nivel y las fases de la programación que pueden cubrir extraída del libro *IEC-61131-3 Programming methodology (1997)* [6]. Se puede apreciar como el SFC es el único que cubre desde el análisis hasta la implementación, pasando por el diseño. Esto, junto a que se trata de un lenguaje de programación de alto nivel, lo hace apto para implementar proyectos donde se usan secuencias simples.

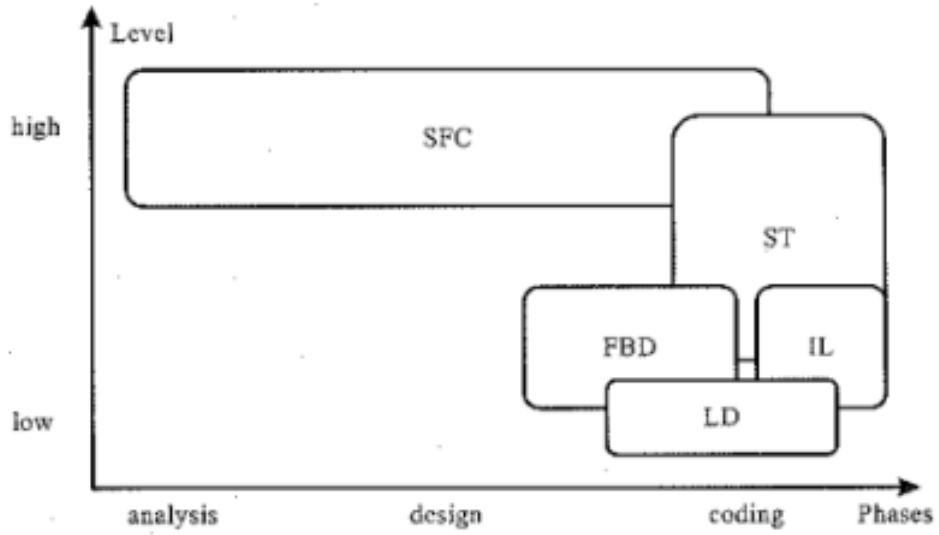


Figura 26. Representación gráfica de los lenguajes de programación bajo el estándar IEC-61131 en función del nivel y las fases en las que se puede emplear. Extracto del libro IEC-61131-3 Programming methodology.

Dentro de los lenguajes representados en el diagrama (e incluidos en la norma IEC-61131-3), se encuentran el Ladder Diagram (LD), el Structured Text (ST), el Function Block Diagram (FBD), Instruction List (IL) y Sequential Function Chart (SFC).

El lenguaje LD es de los lenguajes más comunes, destacando por su facilidad de aprendizaje ya que, como se muestra en la Figura 26, se trata de un lenguaje de bajo nivel. Sin embargo, puede resultar en una implementación más complicada en tareas complejas o programación matemática avanzada.

Para un nivel de programación un poco más elevado, se encuentran el FBD e IL. El FBD es ideal para sistemas complejos y facilita la reutilización de bloques de función, pero puede volverse complicado en proyectos grandes, mientras que el IL es eficiente en términos de recursos del PLC y es útil para tareas precisas, pero puede ser difícil de depurar y menos legible.

Por otro lado, como lenguajes de alto nivel, se encuentran el ST y SFC. El lenguaje ST permite la programación matemática avanzada y es altamente legible, aunque puede requerir un mayor nivel de habilidad de programación. Finalmente, el SFC es excelente para controlar secuencias de eventos y estados, al derivar del lenguaje de modelado GRAFCET, para programas sencillos la implementación es casi directa.

El lenguaje SFC presenta ventajas clave que lo destacan como una elección preferida sobre los otros lenguajes para el control de sistemas. Ofrece un alto poder expresivo, similar a los diagramas de estados y redes de Petri, lo que lo hace ideal para abordar problemas concurrentes y modelar la dinámica del sistema. Su sintaxis gráfica facilita su aprendizaje y uso, siendo útil tanto en el diseño preliminar como en el detallado, adaptándose a la evolución de la información disponible. Además, se integra fácilmente con otros lenguajes del estándar, lo que mejora la eficiencia del desarrollo de software y permite la fragmentación del código para una mejor organización.

Por lo tanto para este proyecto, se escoge como lenguajes de programación de preferencia el lenguaje SFC, acompañado del lenguaje ST para ciertas tareas específicas, como son las funciones.

1.7.3.2.Entorno de ejecución de las aplicaciones

El proceso de programación se ha llevado a cabo mediante el software CODESYS®, el cual está disponible de forma gratuita en la página web de la compañía. Este programa ha sido desarrollado integralmente siguiendo las pautas del estándar IEC-61131, garantizando así la conformidad con los requisitos previamente mencionados y asegurando la compatibilidad con la mayoría de los controladores lógicos programables (PLCs) comerciales. Es de relevancia resaltar que las ediciones más actuales de CODESYS® incorporan un protocolo de conexión OPC, cumpliendo así los requisitos de programación.

Para la programación se usa la versión V3.5 SP17.30 Patch 3 de CODESYS®. Se observa que ninguna de las opciones de dispositivo ofrecidas por defecto corresponde al PLC Wago 750-8212, por lo que se ha comprado el paquete CODESYS® Control for PFC200 SL 4.8.0.0, que soporta la comunicación con los dispositivos de la serie 750 de WAGO lo que incluye el controlador a utilizar (750-8212).

1.7.3.3.Implementación en SFC

A partir del modelado en lenguaje de diseño GRAFCET definido en el apartado 1.7.2.2, se implementan los programas que controlarán el funcionamiento tanto del gemelo digital como de la maqueta. Como ambos sistemas son controlados mediante programas similares, aunque cada uno cuenta con una etiqueta para distinguir las variables (FIO para el gemelo digital y GVL para la maqueta), se incluyen a continuación solo las aplicaciones de control del sistema real. Para una comparación entre los programas para ambos sistemas, así como un listado de las variables utilizadas, consultar el Anexo 2.3.1 y Anexo 2.4 respectivamente.

1.7.3.3.1. Gemelo digital/Sistema físico modular

El comportamiento descrito por los grafcets presentados en las Figuras 23, 24, 25 y 20 se han implementado en el lenguaje SFC, obteniendo así los diagramas presentados en las Figuras 27-30. Estos programas son los encargados del control individual de los módulos que componen la línea de procesado, por los motivos ya mencionados, se incluyen a continuación únicamente los que controlan el sistema físico.

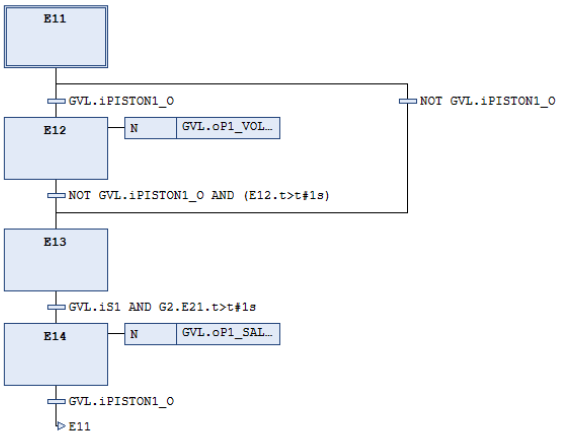


Figura 27. Modelado del proceso de distribución para el sistema físico

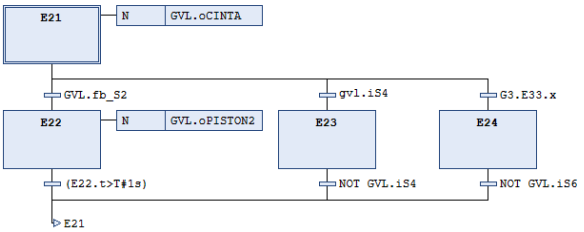


Figura 28. Modelado del proceso de estampado para el sistema físico

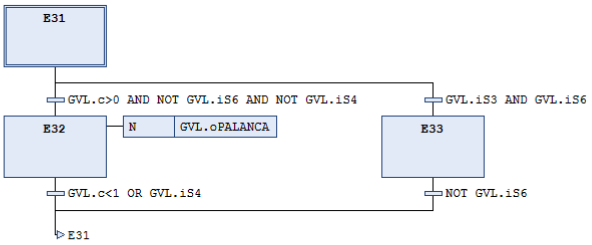


Figura 29. Modelado del proceso de clasificación para el sistema físico

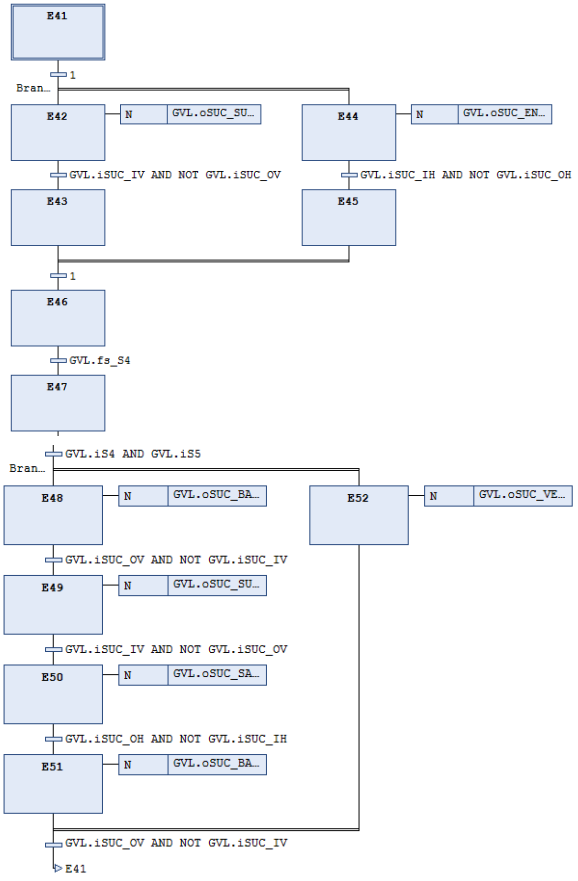


Figura 30. Modelado del módulo de procesado manual para el sistema físico

1.7.3.3.2. Gemelo digital/Sistema físico completo

La implementación del grafcet para la ejecución de los módulos en paralelo es una traducción directa del diseño descrito en la Figura 22, sin necesidad de modificaciones.

1.7.3.3.3. Control simultáneo de ambos sistemas

Para la ejecución simultánea de las aplicaciones del sistema real y digital, se siguen los modos de funcionamiento descritos en el apartado 1.6.2.3. Con esto se obtiene el SFC maestro GEMMA representado en la Figura 31. De este modo, se definen los programas FISICO y FACTORY (Figura 32 y 33, respectivamente) que controlan el forzado de los SFC modulares, ejecutándolos en paralelo en función de los estados activos en GEMMA.

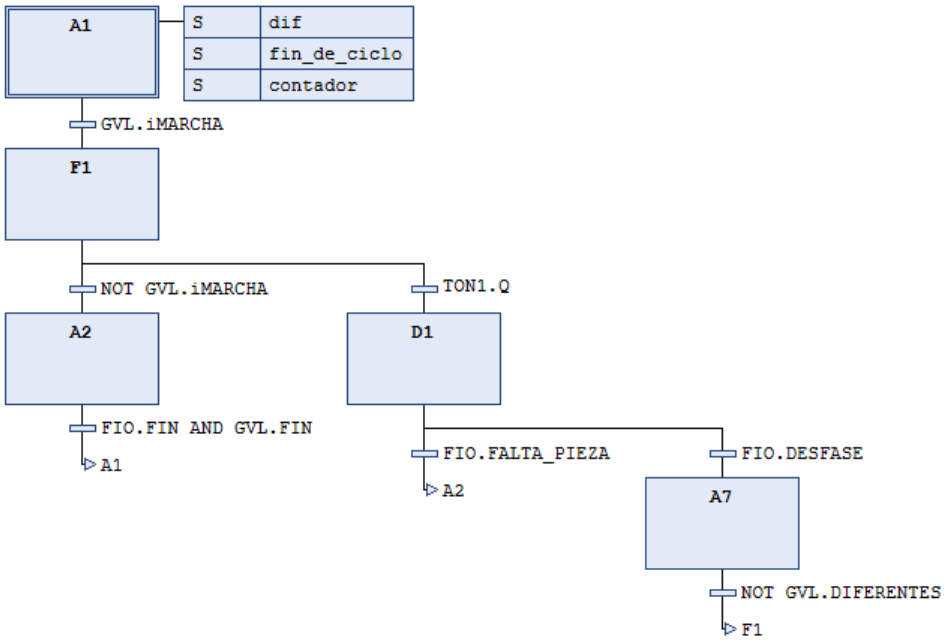


Figura 31. SFC GEMMA

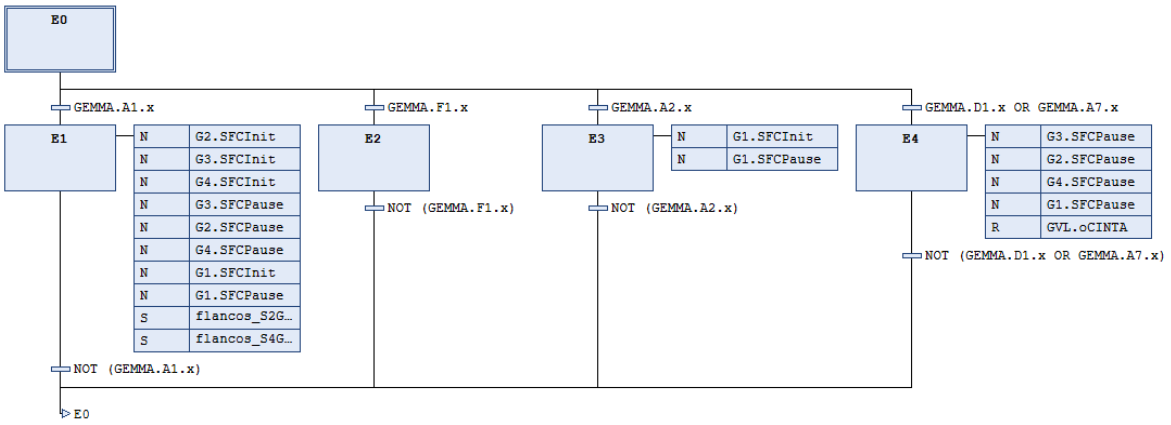


Figura 32. SFC FISICO

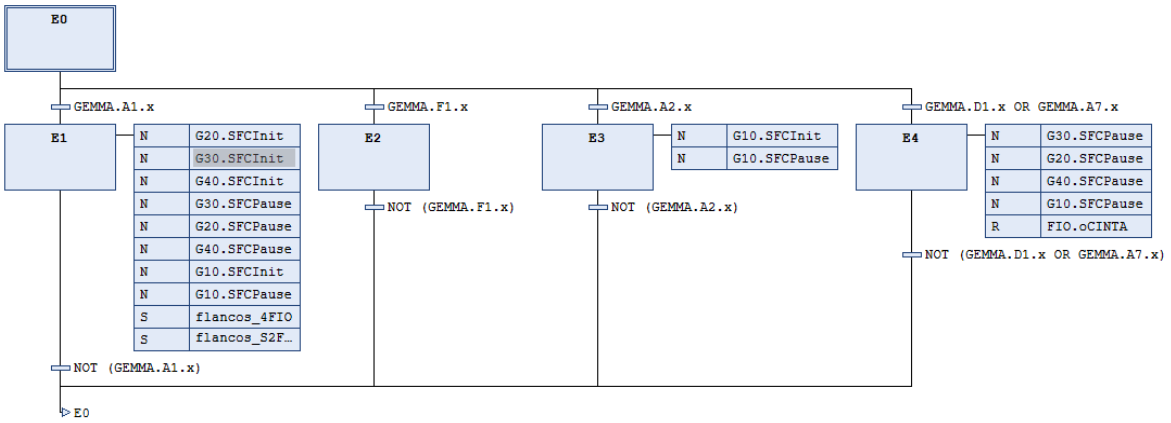


Figura 33. SFC FACTORY

Para el forzado de los SFCs modulares debe ser posible la interrupción del funcionamiento de los distintos módulos, así como el reinicio de cada uno de ellos, de manera independiente a los demás.

Es por esto que en este proyecto se incorporan *SFCInit* y *SFCPause*, variables predeterminadas que permiten el reinicio y la pausa del programa respectivamente. Sin embargo, estas variables se encuentran desactivadas por defecto, por lo que se deben activar específicamente como se muestra en la Figura 34 (desde su menú, accesible desde Ver/Propiedades/Configuración SFC).

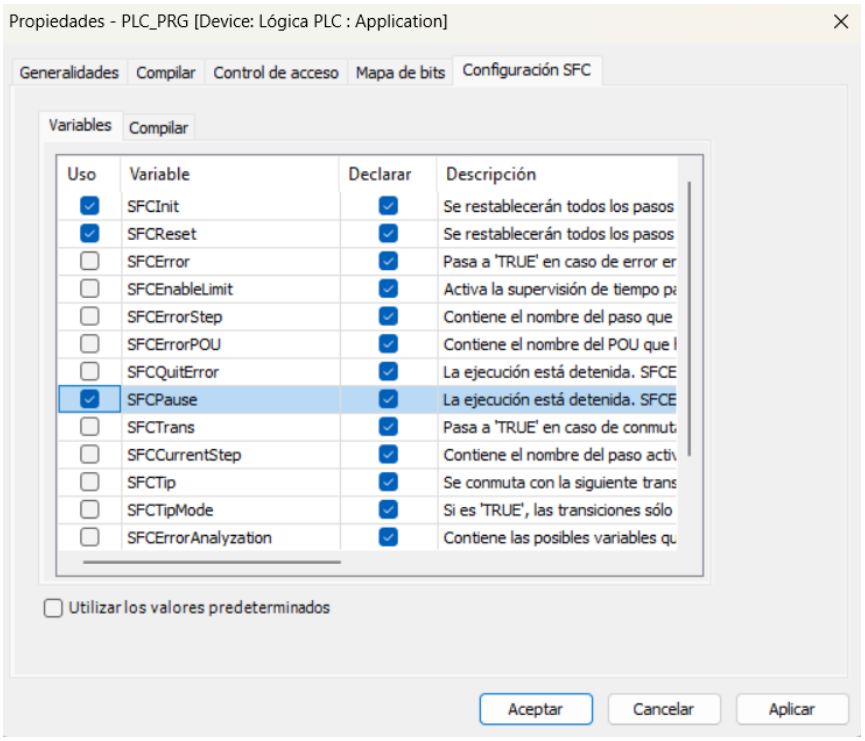


Figura 34. Ventana de propiedades del programa

1.7.4.Análisis de soluciones de comunicación

El proceso de establecer una conexión entre el PLC y Factory I/O se inicia con la consideración de una conexión OPC desde el principio. Esta elección se basa en las ventajas que el protocolo OPC presenta, citadas en la sección anterior, que se puede resumir en una configuración más sencilla y eficiente.

El proceso de garantizar una comunicación fluida y precisa entre CODESYS® y Factory I/O se convierte en una etapa crítica en el desarrollo de este proyecto. Diversos métodos disponibles para este propósito son detallados en la sección 1.6.4 del documento, planteando un abanico de opciones para considerar. En esta línea, se lleva a cabo un proceso de selección que permita identificar el método más adecuado para las necesidades específicas del proyecto. A continuación, se expone en detalle el proceso de evaluación de los protocolos Modbus TCP/IP, OPC DA y OPC UA, y la elección final de este último como la opción más idónea.

En un principio se dispone de la versión de CODESYS® 2.3, dado que es la versión compatible con el firmware del controlador original (Controlador ETHERNET 750-841 de WAGO), pero este carece del soporte necesario para soportar la comunicación mediante OPC UA/DA. En vista de esta limitación, surge la consideración de recurrir al protocolo Modbus TCP/IP como alternativa viable. No obstante, se descubre que esta versión del programa presenta problemas en otros aspectos del proyecto, con lo que se decide actualizar la versión de CODESYS® a la V3.5 Patch17.30. Esta nueva versión permite regresar al plan original de emplear la comunicación a través de OPC. Esta configuración es más sencilla y permite una implementación en un rango más amplio de equipos, así como la comunicación con otros protocolos, lo que no sucede con Modbus.

Una vez descartado el protocolo Modbus, queda decidir entre el protocolo OPC UA y OPC DA. La decisión es indiferente para este proyecto en concreto ya que el Factory I/O sitúa ambas configuraciones en una misma categoría. Sin embargo, a pesar de que ambas opciones brindan funcionalidades similares, se escoge OPC UA ya que este destaca por su mayor rapidez en la puesta en marcha de la comunicación. Además, este protocolo asegura un nivel adicional de seguridad al cifrar la información transmitida.

Ante esta situación, se pasa a analizar y considerar las posibles alternativas. Se evalúan dos opciones principalmente: la primera implica actualizar el firmware del PLC original, mientras que la segunda consiste en la selección de un PLC alternativo que cumpliera con todos los criterios establecidos. Al valorar los costes de cada una de las opciones, se prefiere optar por un nuevo PLC dado que ofrece una mayor versatilidad para futuros proyectos. Puesto que los módulos disponibles corresponden a la serie 750 de la marca WAGO, se ha decidido seleccionar un controlador de la misma marca que sea compatible con dicha serie, siendo este el Controlador PFC200 (PAC) 750-8212.

Este controlador no solo es compatible con los módulos ya incorporados en la maqueta, sino que también es configurable mediante CODESYS® y proporciona un soporte integral para el estándar OPC. En el Anexo 2.5 se pueden observar y comparar las características de cada uno de los controladores.

Teniendo en cuenta además que el Controlador ETHERNET 750-841 se califica como “obsoleto” en la página web de WAGO, se concluye que la elección del Controlador PFC200 (PAC) 750-8212 asegura una base sólida y confiable para la implementación del proyecto en cuestión y posibles extensiones del mismo.

1.8.Resultados finales

1.8.1. Rehabilitación

Como solución a los problemas explicados en el apartado 1.6.1, se reemplaza la junta tórica y se incorpora un amortiguador al final de la rampa de descartes, que amortiguará la caída de las piezas tal y como se muestra en la Figura 35.

En lo referente a la incorporación de un nuevo sensor, este se sitúa al final de la rampa. Entre las opciones valoradas, se descarta el sensor inductivo, ya que tiene un alcance muy limitado y basta con detectar la presencia de una pieza, no es necesario distinguir si es metálica o no. De este modo se sopesa añadir un detector o un sensor de barrera, aunque ambos desempeñan la misma función, en este caso se opta por el detector Telemecanique XUD-J003937/ de SCHNEIDER, ya que dada la disposición de la maqueta resulta más sencillo de incorporar.

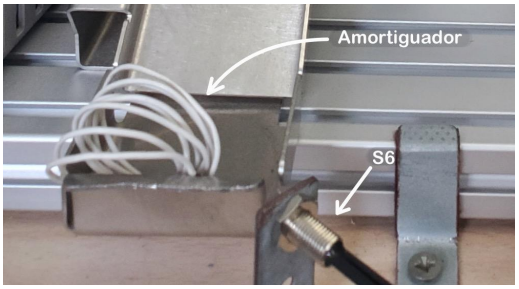


Figura 35. Componentes incorporados en el módulo de clasificación

1.8.2.Modelado

1.8.2.1.Modelado del gemelo digital

Los gemelos digitales que se usan a lo largo del proyecto posibilitan la evaluación de software de forma previa a la implementación en el sistema físico. Con este fin, se deberán adaptar para que sean una réplica lo más parecida posible a la maqueta. Comienza así el proceso de modelado del entorno virtual, el cual consiste en disponer los elementos ofrecidos en la librería de Factory I/O de forma que se crea una estación de procesado que sigue el mismo comportamiento que el sistema físico. Las escenas descritas a continuación conforman los resultados del modelado del gemelo digital.

Como se ha visto con antelación, se diseñan y programan individualmente los distintos módulos involucrados, incorporando (en aquellos módulos donde es necesario) la cinta transportadora. También se incorpora un emisor en cada escena de simulación, encargado del suministro de las piezas. Cada módulo debe demostrar su correcto funcionamiento de manera individual antes de avanzar hacia la prueba del programa en su conjunto.

Se diseña primero el módulo de distribución, encargado de la distribución de las piezas. Como se aprecia en la Figura 36, este consta de un pistón de doble efecto y un sensor detector de presencia, además de un emisor que otorga al usuario la capacidad de seleccionar el material a distribuir.

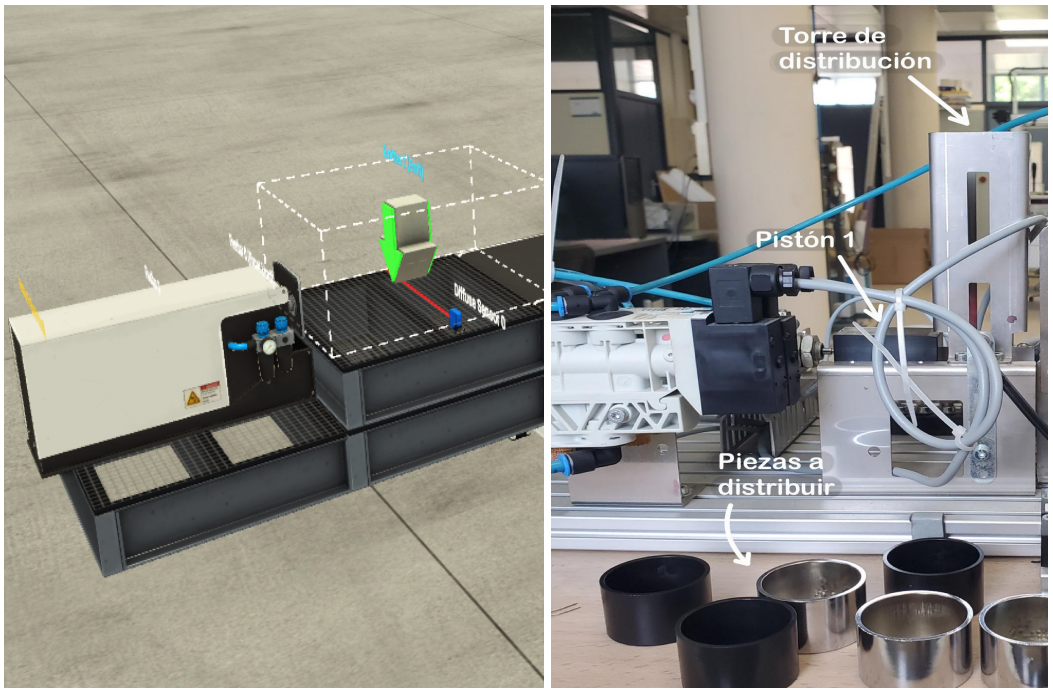


Figura 36. Estación de distribución a la izquierda en el gemelo digital y a la derecha en la maqueta.

La estación de estampado, mostrada en la Figura 37, está compuesta por un sensor de barrera y un pistón que simula el estampado de las piezas. Como se ha visto previamente en la sección 1.7.4.2, el pistón requiere de ajustes específicos en su programación para replicar el comportamiento de la maqueta los cuales son implementados en el dispositivo de control.

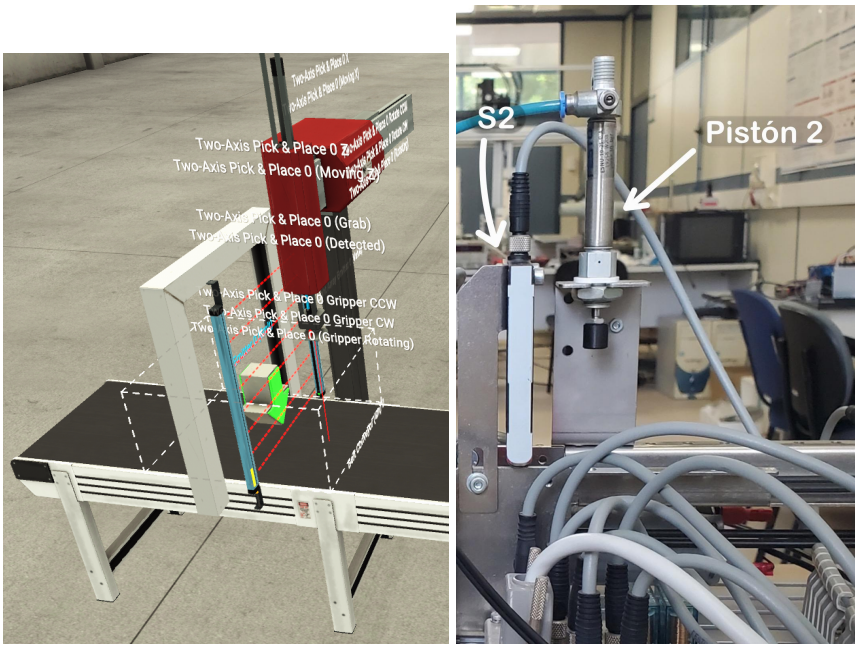


Figura 37. Estación de estampado a la izquierda en el gemelo digital y a la derecha en la maqueta.

Como se observa en la Figura 38, la estación de clasificación consta de un sensor inductivo y una palanca, a los que más tarde se les añadirá un sensor detector de presencia.

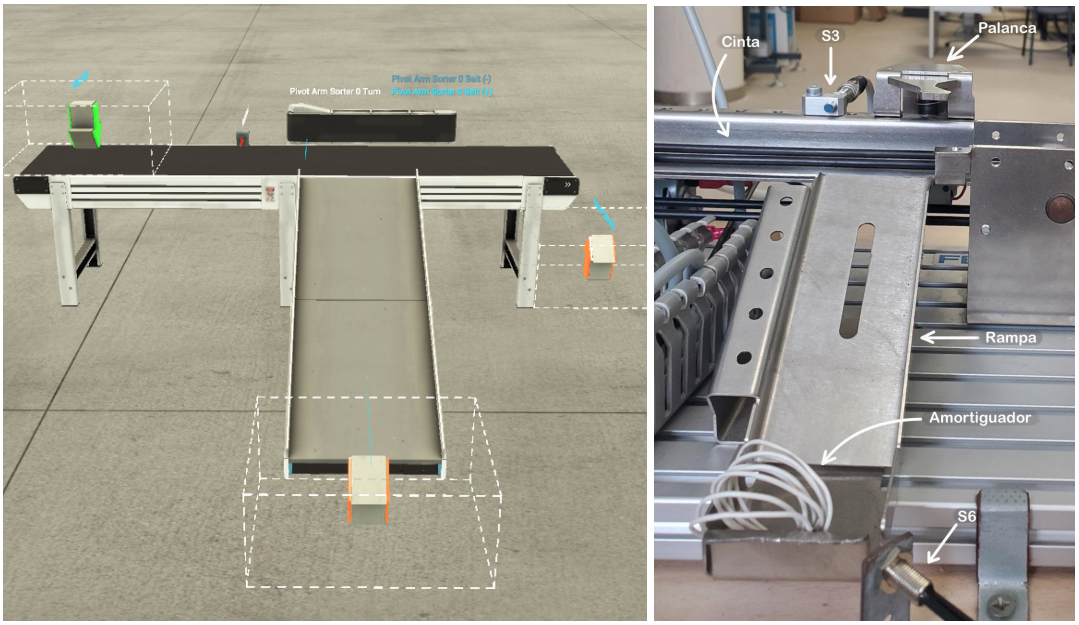


Figura 38. Estación de clasificación a la izquierda en el gemelo digital y a la derecha en la maqueta.

Se modela a continuación la estación de procesado manual (Figura 39) compuesta por dos detectores de presencia, un robot *pick & place* y un emisor que simula el dispensador de bolas de la maqueta.

Al haber verificado el código de adaptación para el pistón de estampado, se ha adoptado la misma solución. Al igual que el resto de módulos se comprueba el correcto funcionamiento del programa sobre el gemelo digital, para después descargarlo en la maqueta y por último ejecutarlo simultáneamente en ambos sistemas.

Con la confirmación de que cada uno de los módulos opera de manera correcta y sin fallos, se avanza hacia la evaluación de la línea de procesado en su conjunto.

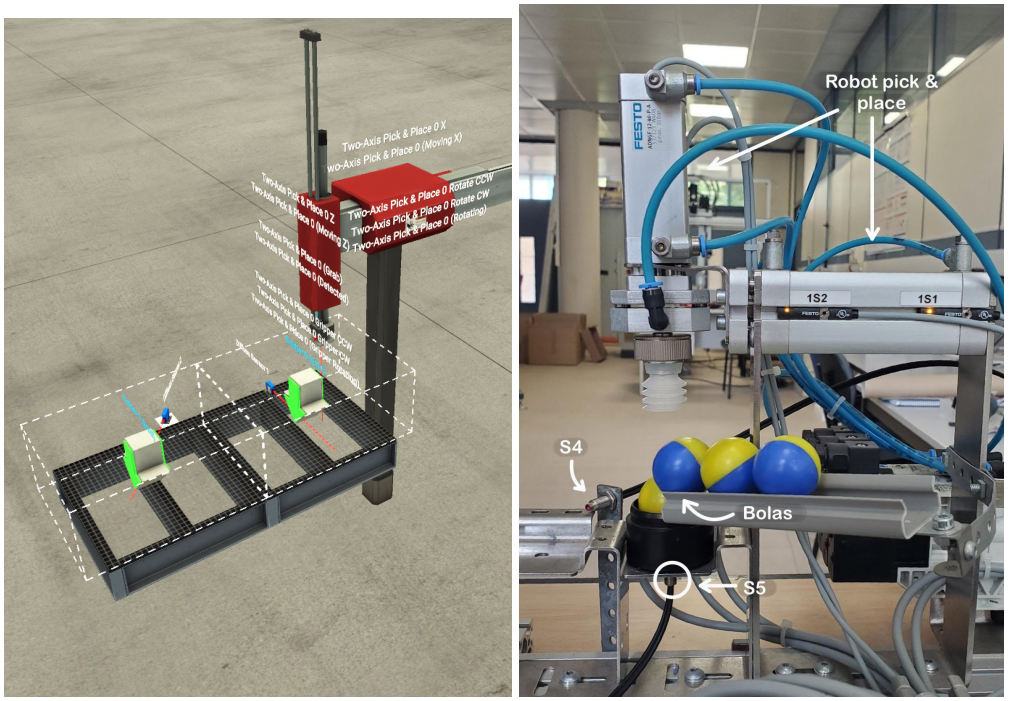


Figura 39. Estación de procesado manual a la izquierda en el gemelo digital y a la derecha en la maqueta.

Tras lograr resultados satisfactorios en el programa sobre el modelo virtual, se estudia su comportamiento en el sistema físico, manteniendo los mismos requisitos. Es entonces cuando se concluye que para lograr una sincronización de tiempos se debe acortar la longitud de la cinta, lo que resulta imposible dadas las restricciones del Factory I/O. En su defecto, se sitúa la estación de distribución sobre la cinta transportadora, y para evitar el transporte temprano de las piezas, se incorpora una plataforma sobre la que se emitirán las piezas. Esto implica que las piezas serán dispensadas desde cierta altura, al caer desde la plataforma las piezas "base del producto" podrían llegar a la cinta ligeramente desviadas. Esta desviación podría ocasionar que, al llegar al módulo de procesado manual, la "tapa del producto" no encaje de manera óptima. De todos modos, se escoge esta opción al no ser un problema crítico ya que el proceso físico no tiene como foco la colocación de la pieza y ser la que mejor permite sincronizar los tiempos de ejecución de los sistemas físico y virtual.

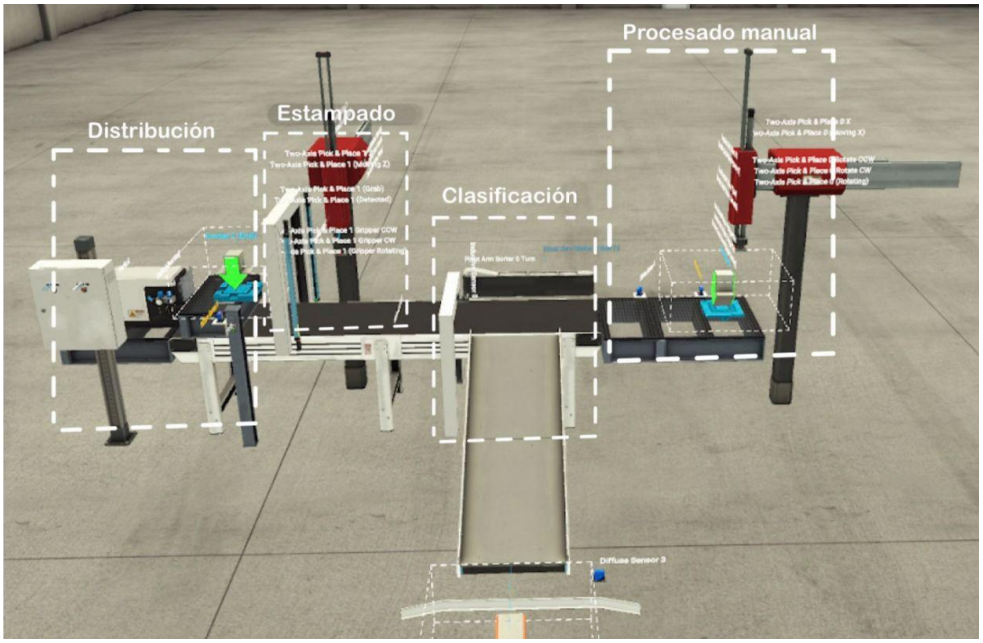


Figura 40. Línea de procesado completa (gemelo digital)

1.8.2.2.Modelado en GRAFCET

Como resultado del modelado en GRAFCET se obtiene la documentación descrita en el apartado 1.7.2.2.2, conformada por los grafcets de la Figura 22 a la Figura 25, además de la Figura 20.

1.8.3.Programación

El primer paso para comenzar a programar consiste en crear dos listas de variables globales, de forma que una de ellas recoja los inputs y outputs para el sistema físico (GVL, que hace referencia a la Lista de Variables Globales) y la otra para el sistema virtual (FIO, que hace referencia a Factory I/O). Por motivos de organización se pone una “i” a las variables que representan inputs y una “o” a aquellas que representan outputs. También se añaden variables internas, que no llevarán ningún identificador delante, como por ejemplo contadores. Para acceder a un listado detallado de variables se puede consultar el Anexo 2.4.

Los SFCs implementados a partir de los grafcets descritos anteriormente conforman el resultado que se pretendía alcanzar. Se incluye a continuación la estructura de la aplicación final, describiendo sus componentes de arriba a abajo según la Figura 41. De este modo, el programa llamado “GEMMA”, controla las aplicaciones “FÍSICO” y “FIO”, las cuales operan de paralelamente y gestionan de forma independiente los cuatro bloques en sus respectivos sistemas.

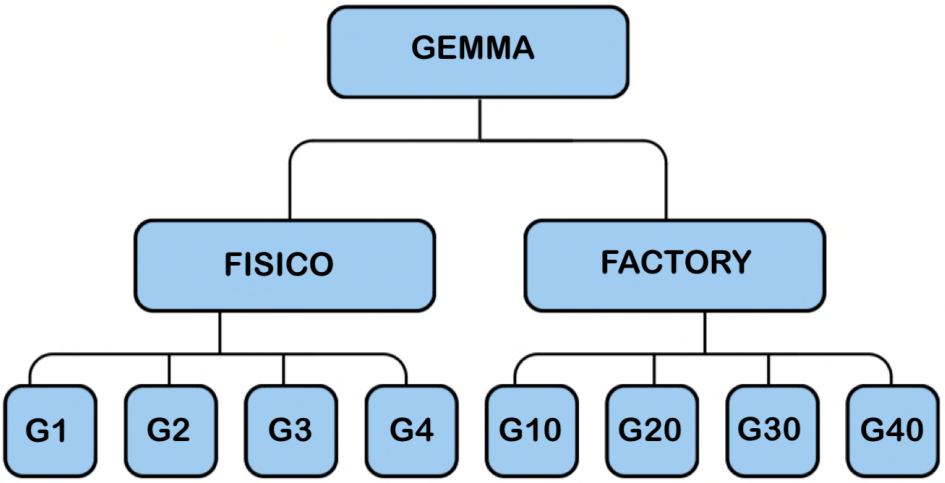


Figura 41. Jerarquía de los Grafcets diseñados

SFC GEMMA

Define las etapas en función a los estados de funcionamiento definidos en la guía gemma. Este SFC opera como SFC maestro, ejerciendo control sobre la activación y desactivación de los restantes programas. Se establece como un componente fundamental en la gestión y control del proceso. Se estructura en cinco etapas basadas en los modos de funcionamiento de la guía GEMMA que se usarán como condiciones de activación/desactivación de los grafcet FISICO y FACTORY. Para la sincronización entre los sistemas físico y digital se evaluarán los estados de los sensores S3, S4 y S5 mediante la función dif, descrita más adelante. La representación completa de este programa se encuentra detallada en la Figura 31.

SFC FISICO y **SFC FACTORY**

Como se puede ver en las Figuras 32 y 33, ambos SFCs comparten la misma estructura, con la diferencia de que cada cual usa sus respectivas variables. Además, se observa una diferencia en la activación/desactivación de los pistones: para el sistema físico, se procede a activar su salida o retorno (doble efecto), mientras que para el sistema digital se activa o desactiva únicamente su salida (simple efecto). Ambos programas constan de una etapa inicial seguida por cuatro etapas mutuamente excluyentes. Todas sus etapas se activan o desactivan en función del estado del SFC GEMMA. Estos programas fuerzan el estado de los SFCs esclavos mediante las variables SFCInit y SFCPause, que permiten reiniciar y pausar el SFC deseado, respectivamente.

SFC G1/G10

Estos SFCs (Anexo 2.3.1) gestionan el módulo de distribución. Se comienza en un estado de reposo, luego verifican si el pistón 1 está extendido y lo retraen si es necesario. Luego, se comprueba si hay una pieza en espera y si ha pasado un segundo desde que el pistón del módulo de estampado comenzó a descender. Finalmente, se activa la salida del pistón 1 para distribuir las piezas.

SFC G2/G20.

Ambos SFCs (Anexo 2.3.1) controlan los módulos de estampado y la cinta transportadora en el sistema. Se inicia activando la cinta transportadora de modo que se detenga en cualquiera de los siguientes casos: durante el estampado de una pieza, durante el procesado manual o cuando se detectan piezas de metal en S3 y S6 simultáneamente.

SFC G3/G30

Estos SFCs controlan el módulo de clasificación, que descarta piezas metálicas. La activación de la palanca de descartes no depende únicamente de la presencia de una pieza metálica si no que se deben cumplir otros requisitos, tal y como se muestra en el Anexo 2.3.1.

SFC G4/G40

Estos SFCs (Anexo 2.3.1) se encargan del control de la estación de procesado manual, compuesta principalmente por el *pick & place*. En este módulo tiene lugar la secuencia de movimientos que junto con la activación de la ventosa permiten recoger la bola y colocarla en el interior de la pieza o, en su defecto, recoger la tapa del producto y colocarla sobre la base del producto. La secuencia de operaciones comienza asumiendo que el sistema pick and place está completamente extendido en ambos ejes.

Se describen a continuación las funciones auxiliares incorporadas mediante ST, cuyo código se puede consultar en el Anexo 2.1:

- **Funciones de detección de flancos.** Durante la programación será necesario detectar los flancos de subida y/o bajada de los sensores S2, S3, S4 y S6. Para esto, por cada flanco a detectar se crea una nueva variable que almacena el valor de dicho sensor, de modo que al cambiar el estado del sensor, se detecta la diferencia entre el estado del sensor y la nueva variable. Por ejemplo, para el flanco de bajada GVL.fb_S2 la variable nueva es “m”, mientras que el sensor que se estudia es S2.

- **Función contador**. Se crea un contador que realiza un seguimiento de piezas plateadas, con el fin de asegurar que estas sean descartadas correctamente. Se programa una función de modo que por cada flanco de subida de S3 (detección de pieza metálica) se suma uno al contador, mientras que por cada flanco de subida de S6 (detección de presencia al final de la rampa) se resta uno. De este modo si dos piezas plateadas circulan muy seguidas la una de la otra no habrá errores. Cabe añadir que como medida preventiva, si se detecta una pieza metálica en S3 cuando aún no se ha retirado la pieza que se encuentra en S6, se parará la cinta, ya que no se detectaría su caída a la pila de descartes.

- **Función de fin de ciclo.** Esta función permanece activa en segundo plano desde el momento en que se activa la línea, haciendo un seguimiento de las piezas que se encuentran en cada línea de procesado (física y digital), almacenando este número en la variable cont, que suma uno por cada flanco de subida a la entrada de la línea (S2), y resta uno por cada flanco de bajada de cada una de las posibles salidas (S6 y S4). De este modo cuando “cont” baja a cero se detecta el fin del ciclo.

- **Función dif:** sincronización sistema físico y virtual. Se encarga de comprobar que ambos sistemas funcionen simultáneamente, para lo que comprueba y compara el estado de las variables cont, S3, S4 y S5. Si cualquiera de estas variables

proporciona valores distintos para cada sistema, se activa la variable GVL.DIFERENTES, que se encuentra por defecto desactivada, y que se volverá a desactivar en caso de dejar de percibir una diferencia entre las señales recibidas por el sistema físico y las señales recibidas por el sistema virtual.

- **Temporizador**. Este código cuenta con un único temporizador programado como: *GEMMA.TON1(In:= GVL.DIFERENTES ,PT:= T#1500MS)*. Esto nos indica que es un temporizador definido en el SFC maestro GEMMA, que depende de la variable GVL.DIFERENTES, y cuyo valor pasará de 0 a 1 cuando hayan transcurrido 1500 milisegundos desde la activación de dicha variable. Se utiliza para complementar a la función dif, dejando un margen de error de sincronización de un segundo y medio.

1.8.4.Comunicación

Como se menciona previamente, el proceso de establecer una conexión entre el PLC y Factory I/O se realiza mediante una conexión OPC UA.

Este protocolo proporciona una conexión segura, sin dar lugar a problemas en ningún momento del proyecto. Permite abordar distintos modos de funcionamiento, mostrados en la Figura 42, permitiendo la comunicación bidireccional. De este modo, se parte de un programa que controla únicamente el gemelo digital o el sistema real. Una vez la comunicación unidireccional es satisfactoria, se pasa a un nuevo modelo que permita el intercambio de información entre ambos sistemas (físico y virtual), este nuevo enfoque no supone ningún inconveniente, ya que únicamente requiere de la definición de nuevas variables que permitan almacenar y trabajar con la información de Factory IO.

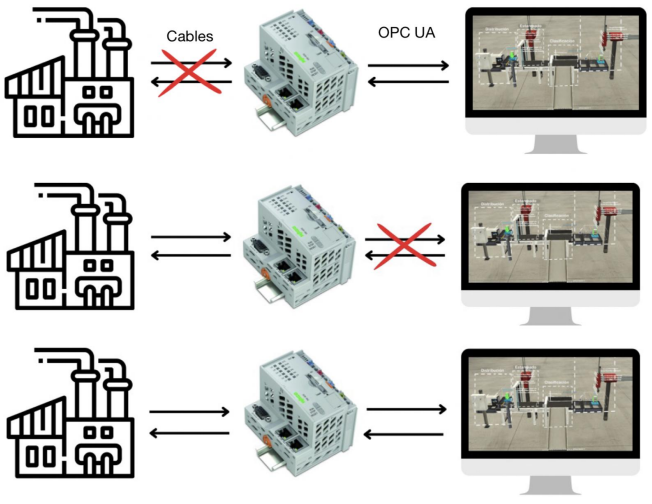


Figura 42. Modos de comunicación maqueta-PLC-gemelo digital.

Para una explicación de la configuración necesaria para establecer la comunicación entre el sistema real con el gemelo digital, consultar el Anexo 2.6.

Como resultado del trabajo realizado, se ha actualizado el PLC a uno más versátil, y por lo tanto más útil para futuros proyectos, instalando además el paquete necesario para conectar el entorno de programación CODESYS®.

1.8.5.Discusión de resultados

Se plantea, en un principio, un programa que controle tanto la maqueta como el gemelo digital de manera secuencial. El principal problema de este código es el procesado de una única pieza a la vez. Debido a la estructura de programación implementada en el grafcet utilizado para esta fase inicial, para que una nueva pieza sea distribuida, la pieza anterior debe haber abandonado la línea de procesado. Es por esto que se opta por una ejecución de los programas de los módulos en paralelo

Al realizar la validación de este nuevo código sobre el gemelo digital se ha detectado que son necesarias distintas modificaciones, tanto sobre el gemelo digital, como sobre el modelado en GRAFCET ya que se pueden dar ciertas discrepancias entre el comportamiento del sistema virtual y el físico.

No obstante, se identifican ciertas limitaciones en la convergencia del comportamiento físico con el virtual. Por ejemplo, a consecuencia de tener que ajustar el gemelo digital para obtener periodos de ejecución similares, se introduce un margen mínimo de error que podría pasar inadvertido. Aunque este error aparentemente insignificante no supone un problema en las fases iniciales de prueba del software en el gemelo digital, se convierte en un desafío cuando ambos sistemas se ejecutan en paralelo durante períodos extensos, ya que el error acumulado, en términos de desfases temporales, se magnifica y genera una asincronía.

También se menciona en la sección 1.7.1 la incorporación de un sensor detector de presencia (S6), como solución a un problema al ejecutar el programa en la maqueta. Este fallo no se pudo prever durante el análisis y estudio en el gemelo digital. Esta circunstancia resalta la dificultad de anticipar todas las posibles complicaciones en un entorno virtual, especialmente cuando se trata de sistemas altamente complejos. En el contexto de una fábrica real, enfrentar situaciones similares podría llevar a tiempos de espera no planificados y, consecuentemente, a costos materiales adicionales.

Por el contrario, también se ha dado el caso de tener que proponer soluciones a la falta de flexibilidad del entorno de simulación. En el caso de la palanca, debido a la falta de similaridad entre la palanca ofrecida en la librería del Factory I/O y la incluída en la maqueta, surgían problemas que no existían en la realidad. Aunque esta situación puede ser resuelta mediante ajustes en los componentes del gemelo digital, también se requiere de la edición del programa en sí. Este esfuerzo adicional de edición sería prescindible en caso de trabajar exclusivamente con el sistema físico, lo que muestra una de las desventajas que pueden derivar la gestión de sistemas híbridos que incorporan tanto aspectos virtuales como físicos.

A pesar de estas limitaciones, se ha confirmado que el gemelo digital es una herramienta valiosa para prevenir errores de programación, ahorrar tiempo y costes, al permitir el desarrollo y prueba de las aplicaciones de control al tiempo que se piden o reemplazan piezas físicas.

De este mismo modo, la programación modular previa a la implementación en la maqueta ayuda a desarrollar y estructurar el código de forma sencilla. Permite su reusabilidad, al ser luego usado en conjunto para controlar el sistema completo.

En conclusión, se puede decir que los gemelos digitales han demostrado ser una herramienta útil para replicar de forma generalista el comportamiento de sistemas reales. Existen, sin embargo, ciertas limitaciones en el uso del software Factory I/O, a saber, falta de flexibilidad en el escalado temporal e incorporación de nuevos componentes.

Debido a que sí que pueden emular el funcionamiento de un sistema real en un entorno visual, constituyen una herramienta útil para la validación del software.

1.9. Plan de trabajo

Al tratarse de un proyecto de investigación, han ido surgiendo cuestiones a medida que se avanzaba en la implementación, lo que requiere una adaptación y resolución constante. La planificación seguida durante la realización de este proyecto se representa de forma aproximada en la Figura 43.

En este diagrama se puede ver el proceso de validación del programa, que se realizó primero de manera modular para después estudiar el comportamiento de la línea de producción al comunicar el sistema físico y el digital entre sí.

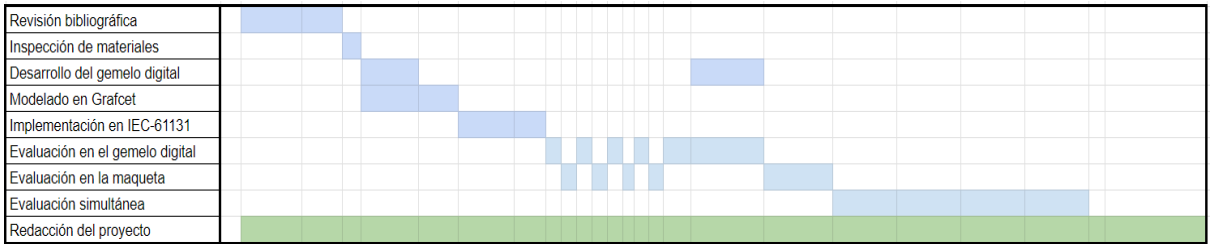


Figura 43. Diagrama de Gantt del proyecto

El modelado y la implementación de este proyecto se han realizado siguiendo el método V, planteado como se muestra en la Figura 44. En esencia, este modelo divide el proyecto en dos vertientes, una que va de izquierda a derecha y otra que corresponde a su contraparte de derecha a izquierda, creando una estructura en forma de "V" que engloba las fases de desarrollo, de validación y de verificación. La metodología en forma de "V" sugiere una conexión directa entre los pasos de definición de requisitos y validación, lo que asegura que cada etapa esté alineada con su equivalente en la fase de validación. Esto favorece la detección temprana de problemas y la corrección de posibles desviaciones.

Como se aprecia en la imagen, la organización del proyecto se corresponde con los niveles de definición de la aplicación de control: En el más alto nivel, encontramos las tareas relativas al desarrollo y diseño de programas en función a los modos de marcha y parada del proceso, identificados como "GEMMA"; a continuación, se trabajan las tareas de coordinación de módulos, representadas por "FISICO" y "FACTORY"; y en el nivel más bajo, se encuentran las tareas relacionadas con el modelado y la programación de los módulos en los que se divide la línea de procesado, que incluyen "G1/10", "G2/20", "G3/30" y "G4/40".

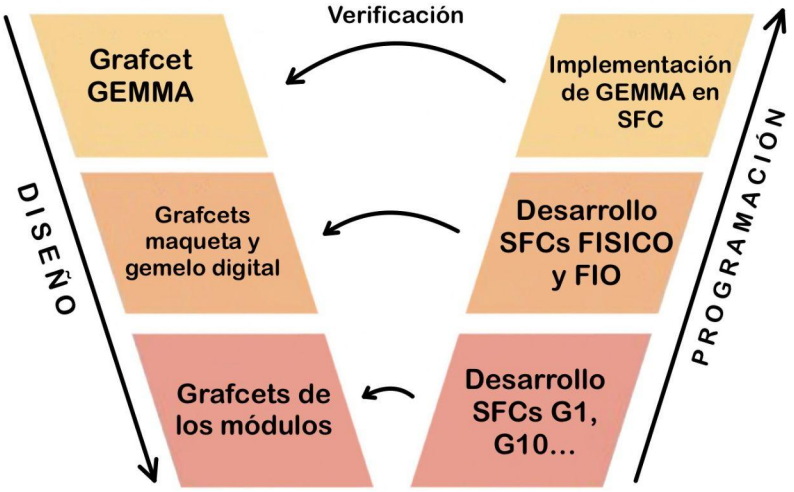


Figura 44. Modelo en V del proyecto

2.Anexos

2.1.Programación en ST

```
        IF (GVL.oP1_SALIR <> GVL.oP1_VOLVER) THEN
        FIO.oP1_SALIR := GVL.oP1_SALIR * (NOT GVL.oP1_VOLVER);
    END_IF
```

Algoritmo 1. Programación del pistón 1 digital en función de la información del sistema real

```
    IF (GVL.oSUC_BAJAR <> GVL.oSUC_SUBIR) THEN
    FIO.oSUC_BAJAR := GVL.oSUC_BAJAR * (NOT GVL.oSUC_SUBIR);
END_IF
    IF (GVL.oSUC_SALIR <> GVL.oSUC_ENTRAR) THEN
    FIO.oSUC_SALIR := GVL.oSUC_SALIR * (NOT GVL.oSUC_ENTRAR);
END_IF
```

Algoritmo 2. Programación del pick & place digital en función de la información del sistema real

```
1   //bajada
2   IF (GVL.iS2=0 AND GVL.m) THEN
3       GVL.m:=0;
4       GVL.fb_S2:=1;
5   ELSIF (NOT(GVL.iS2=0) AND (NOT GVL.m)) THEN
6       GVL.m:=1;
7       GVL.fb_S2:=0;
8   ELSE
9        GVL.fb_S2:=0;
10  END_IF
```

Algoritmo 3. Código para la detección de flancos de bajada del detector 2 (S2)

```
//FISICO

IF (NOT(GVL.iS3=0) AND NOT GVL.p) THEN
    GVL.p:=1;
    GVL.fs_S3:=1;
ELSIF (GVL.iS3=0 AND GVL.p) THEN
    GVL.p:=0;
    GVL.fs_S3:=0;
ELSE
     GVL.fs_S3:=0;
END_IF
IF (NOT(GVL.iS6=0) AND NOT GVL.q) THEN
    GVL.q:=1;
    GVL.fs_S6:=1;
ELSIF (GVL.iS6=0 AND GVL.q) THEN
    GVL.q:=0;
    GVL.fs_S6:=0;
ELSE
     GVL.fs_S6:=0;
END_IF
IF (GVL.fs_S3) THEN
    GVL.c:=GVL.c + 1;
END_IF
IF (GVL.fs_S6) THEN
    GVL.c:=GVL.c - 1;
END_IF
```

```
//FIO

IF (NOT(FIO.iS3=0) AND NOT FIO.p) THEN
    FIO.p:=1;
    FIO.fs_S3:=1;
ELSIF (FIO.iS3=0 AND FIO.p) THEN
    FIO.p:=0;
    FIO.fs_S3:=0;
ELSE
     FIO.fs_S3:=0;
END_IF
IF (NOT(FIO.iS6=0) AND NOT FIO.q) THEN
    FIO.q:=1;
    FIO.fs_S6:=1;
ELSIF (FIO.iS6=0 AND FIO.q) THEN
    FIO.q:=0;
    FIO.fs_S6:=0;
ELSE
     FIO.fs_S6:=0;
END_IF
IF (FIO.fs_S3) THEN
    FIO.c:=FIO.c + 1;
END_IF
IF (FIO.fs_S6) THEN
    FIO.c:=FIO.c - 1;
END_IF
```

Algoritmo 4. Código de la función contador. A la izquierda aplicado al sistema físico, y a la derecha aplicado al gemelo digital.

```
//fisico
IF (GVL.fs_S2=TRUE) THEN
    GVL.cont:=GVL.cont+1;
END_IF
IF (GVL.fs_S6=TRUE OR GVL.fb_S4=TRUE) THEN
    GVL.cont:=GVL.cont-1;
END_IF
IF (GVL.cont>0)THEN
    GVL.FIN:=FALSE;
ELSE
    GVL.FIN:=TRUE;
END_IF

//factory
IF (FIO.fs_S2=TRUE) THEN
    FIO.cont:=FIO.cont+1;
END_IF
IF (FIO.fs_S6=TRUE OR FIO.fb_S4=TRUE) THEN
    FIO.cont:=FIO.cont-1;
END_IF
IF (FIO.cont>0)THEN
    FIO.FIN:=FALSE;
ELSE
    FIO.FIN:=TRUE;
END_IF
```

Algoritmo 5. Código de la función fin de ciclo.

```
IF( (GVL.cont <> FIO.cont) OR (FIO.iS3 <> GVL.iS3) OR (FIO.iS4 <> GVL.iS4) OR (FIO.iS5 <> GVL.iS5) )THEN
    GVL.DIFERENTES:=1;
ELSE
    GVL.DIFERENTES:=0;
END_IF
```

Algoritmo 6. Código de la función dif.

2.2.Modelado en GRAFCET

2.2.1.Modelado de las macroetapas

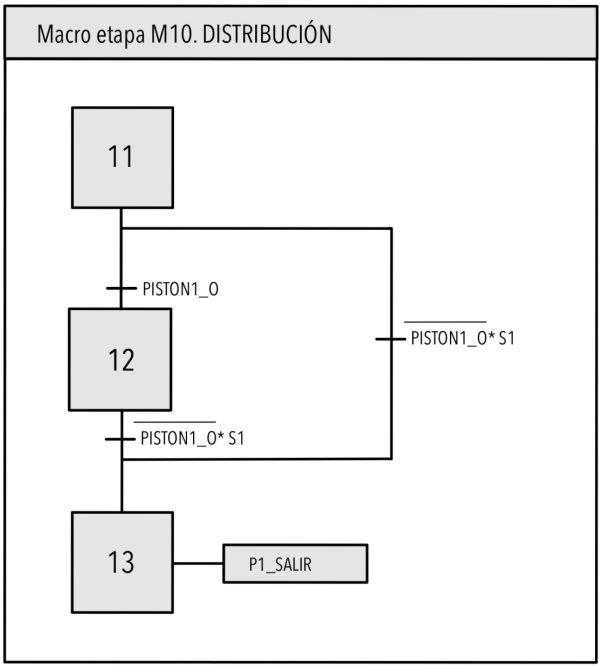


Figura 45. Modelado de la macroetapa del proceso de distribución

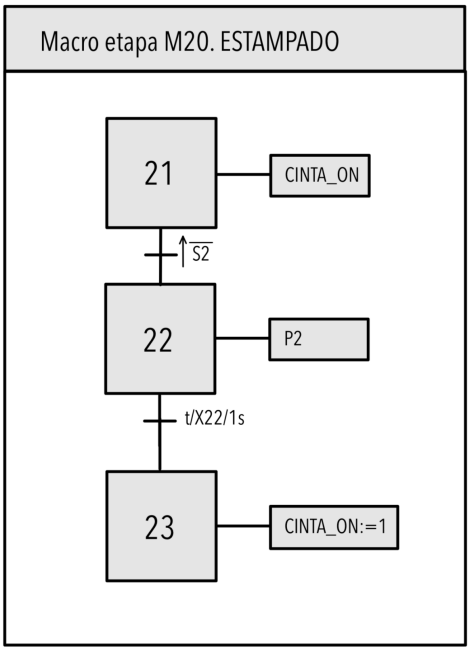


Figura 46. Modelado de la macroetapa del proceso de estampado

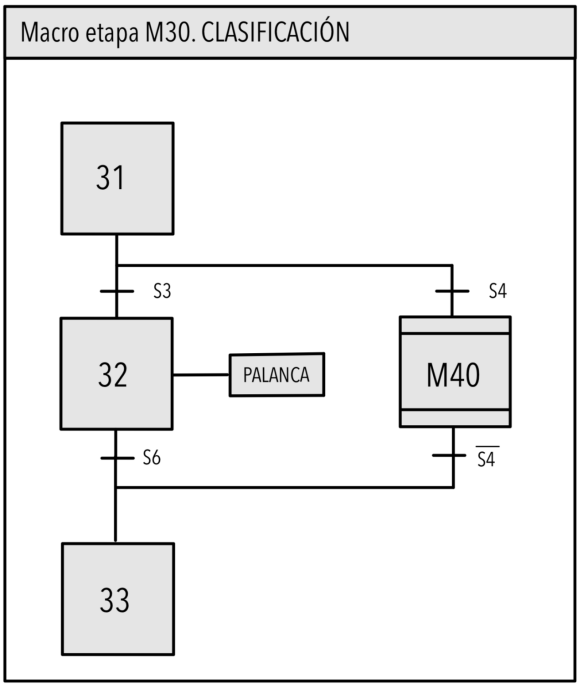


Figura 47. Modelado de la macroetapa del proceso de clasificación

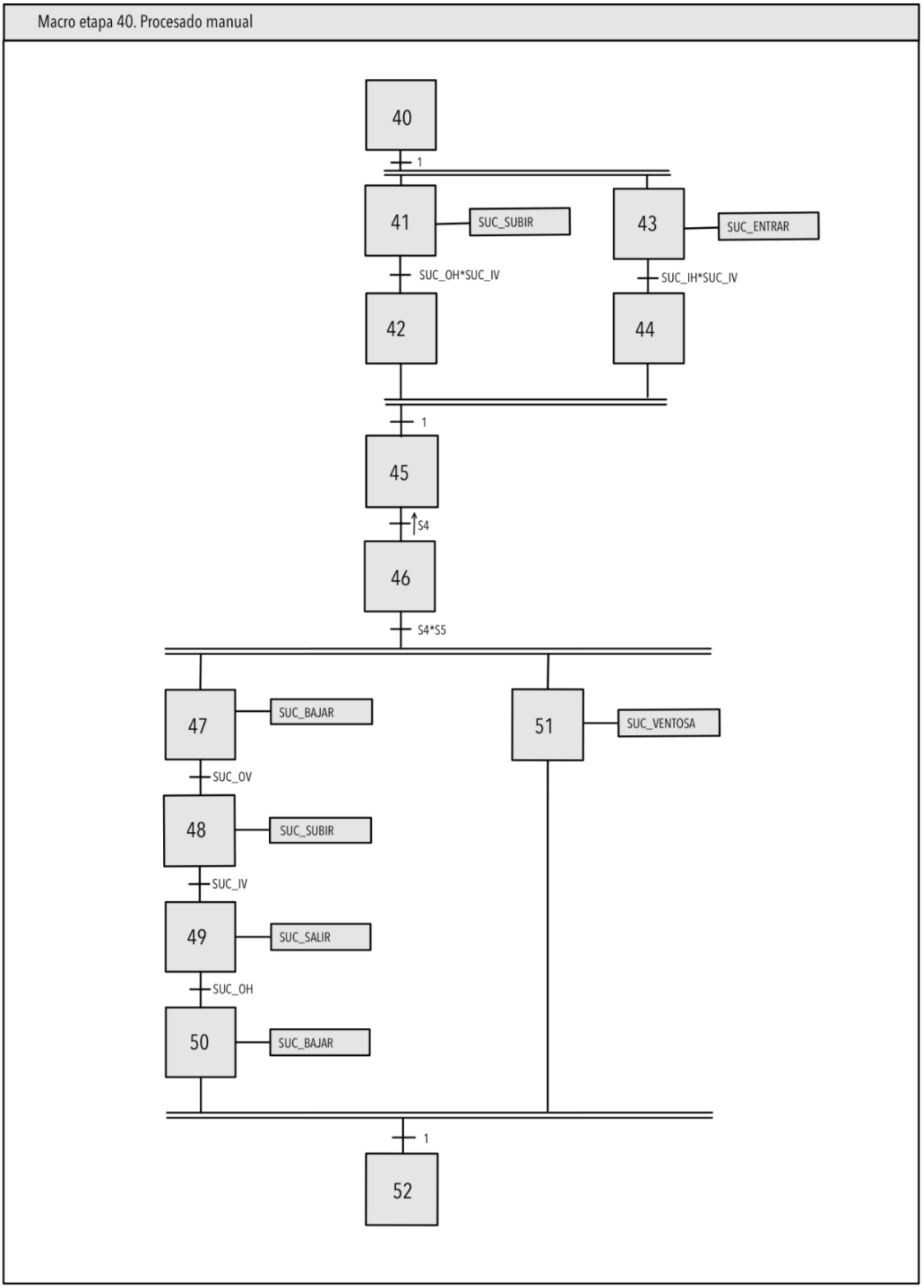


Figura 48. Modelado de la macroetapa del procesado manual

2.3.Implementación en SFC

2.3.1.Implementación de los módulos en paralelo

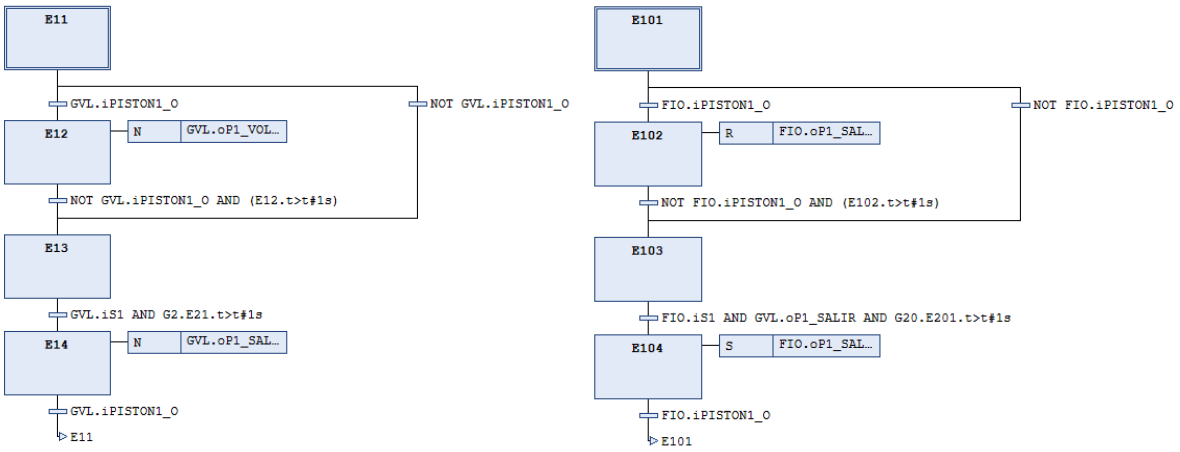


Figura 49. Modelado del proceso de distribución para: (izquierda) sistema físico y (derecha) gemelo digital

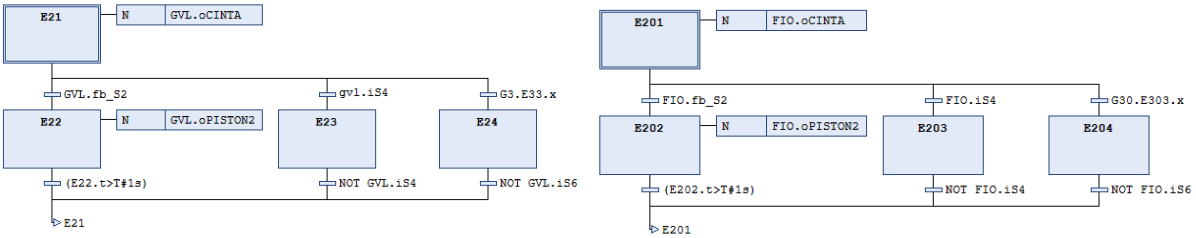


Figura 50. Modelado del proceso de estampado para: (izquierda) sistema físico y (derecha) gemelo digital

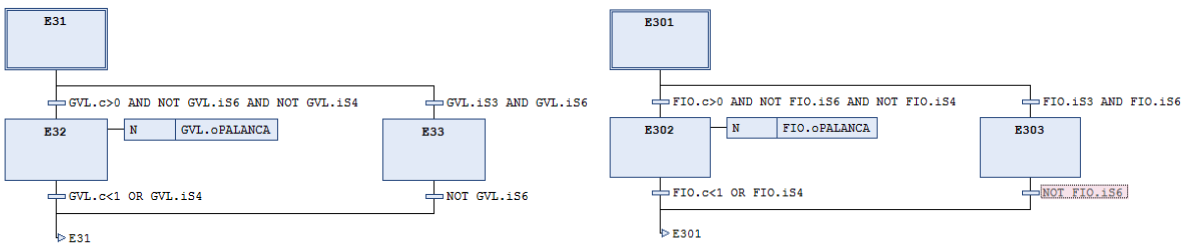


Figura 51. Modelado del proceso de clasificación para: (izquierda) sistema físico y (derecha) gemelo digital

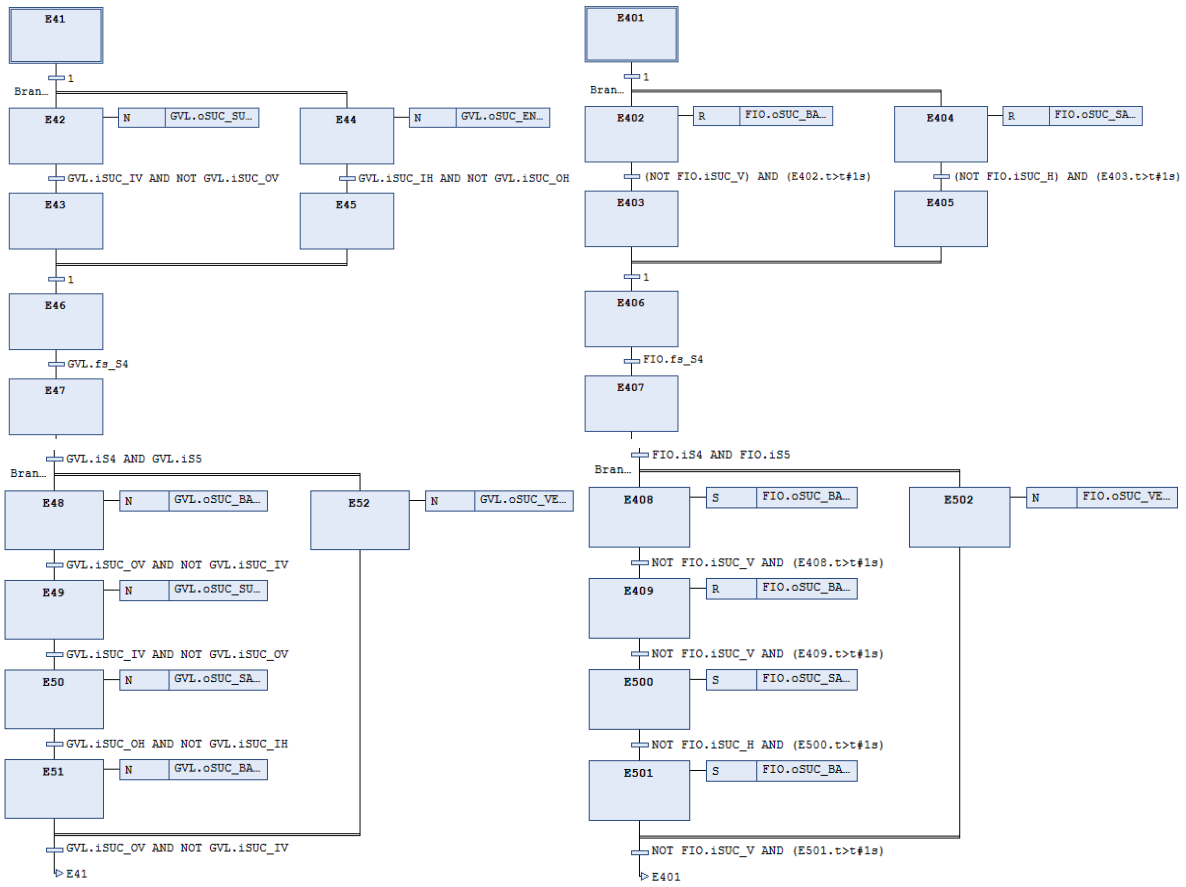


Figura 52. Modelado del módulo de procesado manual para: (izquierda) sistema físico y (derecha) gemelo digital

2.4.Listados de variables

Módulo	Cable	Dirección	Tipo	Variable
I0	4	%IX0.0	BOOL	iPISTON1_O
	---	%IX0.1	BOOL	---
I1	1	%IX1.0	BOOL	iS3
	7	%IX1.1	BOOL	iSUC_IV
	5	%IX1.2	BOOL	iSUC_OV
	3	%IX1.3	BOOL	iS2
I2	3	%IX2.0	BOOL	iSUC_IH
	1	%IX2.1	BOOL	iSUC_OH
	*	%IX2.2	BOOL	iS4

	*	%IX2.3	BOOL	iS5
I3	*	%IX3.0	BOOL	iS1
	*	%IX3.1	BOOL	iMARCHA
	---	%IX3.2	BOOL	---
	*	%IX3.3	BOOL	iS6
Q0	3	%QX0.0	BOOL	oP1_SALIR
	1	%QX0.1	BOOL	oP1_VOLVER
Q1	2	%QX1.0	BOOL	oSUC_SALIR
	4	%QX1.1	BOOL	oSUC_ENTRAR
	6	%QX1.2	BOOL	oSUC_BAJAR
	8	%QX1.3	BOOL	oSUC_SUBIR
Q2	2	%QX2.0	BOOL	oPISTON2
	2	%QX2.1	BOOL	oPALANCA
	2	%QX2.2	BOOL	oSUC_VENTOSA
	4	%QX2.3	BOOL	oCINTA
Variables internas	*	*	BOOL	FALTA_PIEZA
	*	*	BOOL	DESFASE
	*	*	BOOL	DIFERENTES
	*	*	BOOL	fb_S2
	*	*	BOOL	fs_S2
	*	*	BOOL	m
	*	*	BOOL	m2
	*	*	BOOL	fb_S4
	*	*	BOOL	fs_S4
	*	*	BOOL	n
	*	*	BOOL	n2
	*	*	INT	c
	*	*	BOOL	fs_S3
	*	*	BOOL	p

	*	*	BOOL	fs_S6
	*	*	BOOL	q
	*	*	BOOL	FIN
	*	*	INT	cont

Tabla 2. Listado de variables con asignación de entradas (%IX) y salidas (%QX) en el PLC

** No aplicable*

2.5.Comparación PLCs

	Controlador ETHERNET 750-841	Controlador PFC200 (PAC) 750-8212
Comunicación	EtherNet/IPTM Modbus (TCP, UDP) ETHERNET	Modbus TCP maestro/esclavo Modbus (UDP), WagoAppPlcModbus Library Modbus (RTU), WagoAppPlcModbus Library ETHERNET EtherNet/IPTM Adapter (slave) EtherNet/IPTM Scanner EtherCAT® Master OPC UA Server/Client OPC UA Pub/Sub (can be installed later) MQTT Interfaz serie RS-232 Interfaz serie RS-485 BACnet/IP, requiere una licencia adicional Protocolos de telecontrol (requiere una licencia adicional en el dispositivo)
Protocolos ETHERNET	HTTP BootP DHCP DNS SNTP FTP SNMP SMTP	DHCP DNS NTP FTP FTPS SNMP HTTP HTTPS SSH

Entorno de programación	WAGO-I/O-PRO V2.3 (basado en CODESYS® V2.3)	CODESYS® V3.5, versión de firmware 23 o superior e!COCKPIT (basado en CODESYS® V3) hasta la versión de firmware 22 WAGO-I/O-PRO V2.3 (basado en CODESYS® V2.3), hasta la versión de firmware 22
Indicadores		LED (SYS, RUN, I/O, U1 … U7) rojo/verde/naranja: estado de sistema, programa, bus de datos local, estado programable por usuario (puede utilizarse a través de librería CODESYS®); LED (A, B) verde: estado de fuente de alimentación de sistema, alimentación de campo

Tabla 3. Comparación de los controladores 750-841 y 750-8212

2.6.Configuración de comunicación

A continuación se presentan las configuraciones necesarias en Factory I/O para establecer una comunicación efectiva con el PLC físico, al que se descargará el programa desarrollado en CODESYS®. Una vez seleccionado el dispositivo PLC con el que se comunica CODESYS®, será este el encargado de intercambiar información tanto con la maqueta física como con su contraparte virtual.

El primer paso consiste en seleccionar el dispositivo al que se desea conectar el programa, pudiendo ser necesaria la creación de usuario y contraseña en el proceso, en función de la versión de CODESYS® utilizada.

Tras esto se ajusta la configuración de comunicación, de modo que se permita el inicio de sesión anónimo y la gestión de usuario, tal y como se muestra en la Figura 57.

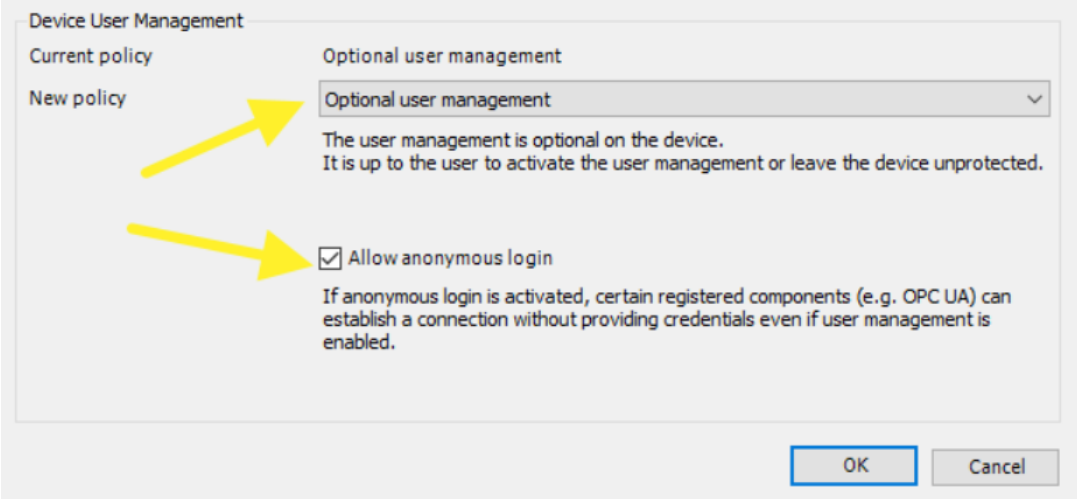


Figura 53. Ventana de cambio de la política de comunicación

A continuación se comprueban los derechos de acceso para el servidor OPC UA, cerciorarse de que este tiene permiso para modificar, ver y ejecutar el programa. De este mismo modo, al crear la ventana de “Configuración de símbolos” (Figura 58) se debe tener en cuenta que debe soportar las características para OPC UA, lo que permite acceder a un listado de variables y seleccionar las que se utilizarán en el gemelo virtual, en este caso las etiquetadas como “FIO”. Tras esto se procede a compilar y descargar el proyecto al PLC.

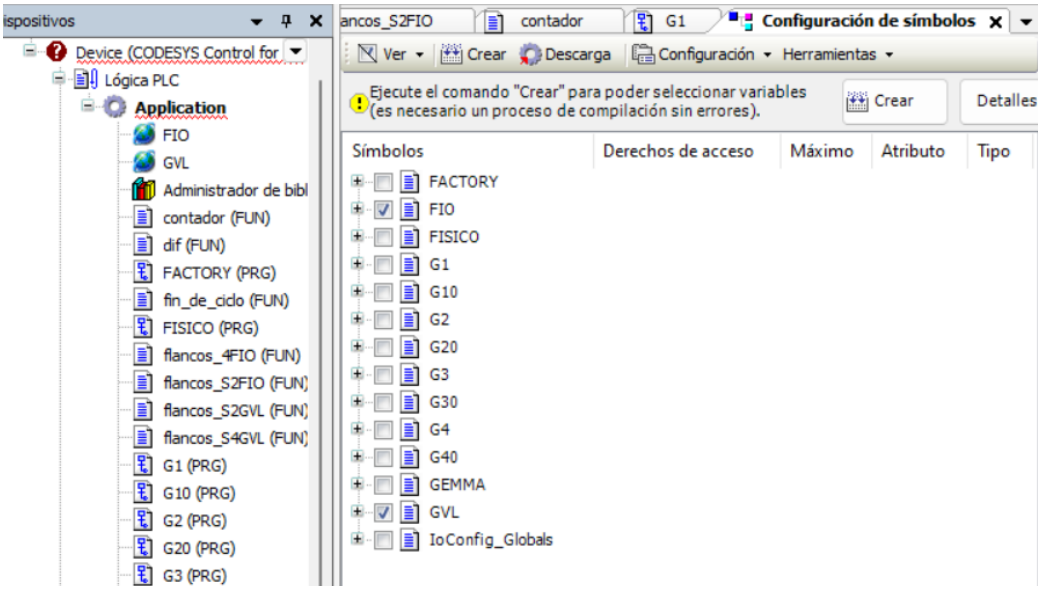


Figura 54. Ventana de configuración de símbolos

Una vez el programa ha sido descargado al PLC, éste establecerá una comunicación directa con el sistema físico mediante los módulos incorporados, mientras que para la comunicación con el sistema virtual se recurre a una conexión vía Ethernet.

Para realizar la conexión mediante Ethernet, es necesario descargar la aplicación BootP-DHCP Tool, la cual permite redefinir la dirección IP del dispositivo (en este caso el PLC) mientras la aplicación esté en funcionamiento. Para esto, primero se selecciona el

adaptador de Ethernet mediante el cual se conecta el PC al PLC o dispositivo deseado, reconocible gracias a la dirección MAC, que en este caso es 00:30:DE:49:FE:C4. Una vez seleccionado, el dispositivo debería comenzar a enviar señales de ping a la estación de trabajo. La aplicación mostrará la cantidad de dichos pings y el tipo, así como la dirección MAC del dispositivo que los envió. Se debe buscar la dirección MAC que coincida con el dispositivo objetivo y el tipo debe estar configurado como BOOTP.

Una vez se ha localizado el dispositivo, se hace doble clic sobre éste para configurar la dirección IP temporal, que debe estar en la misma red que el ordenador.

Se pasa ahora a la configuración del software de simulación Factory I/O. Partiendo de la escena ya diseñada previamente se escoge el protocolo “OPC Client DA/UA” entre las diversas opciones de protocolos de comunicación disponibles. Una vez elegido el protocolo de comunicación se busca el servidor donde están las variables del sistema. La dirección que se utiliza por defecto es “opc.tcp://localhost:4840”, pero se adaptará para reflejar la dirección IP del dispositivo objetivo de modo que, para este proyecto, en el campo OPC Server se escribe “opc.tcp://localhost:(192.168.1.3)”.

Por último, para concluir con la configuración del servidor se buscan y establecen los nodos o variables a utilizar. Mediante las opciones de filtrado que ofrece la aplicación, se filtran las variables para mostrar únicamente aquellas que comienzan con la etiqueta "FIO", que previamente se ha asignado a todas las variables destinadas a la simulación virtual. Al regresar a la pantalla anterior, estas variables pueden asignarse a los sensores y actuadores correspondientes (Figura 59).

Figura 55. Asignación de entradas y salidas en Factory I/O

Una vez se ha concluido la asignación de variables se comprueba que haya una conexión satisfactoria, si no hay errores se vuelve a la escena del proyecto, donde se dará comienzo a la simulación.

2.7.Materiales utilizados

Se listan a continuación las piezas incluidas en el MecLab® Mechatronics Training System de FESTO, junto con sus características. Se ha de tener en cuenta que algunas de las piezas originales han sido reemplazadas a lo largo de los años, con lo que el material listado puede no ser de FESTO, pero cumple la misma función.

DSNU-10-50-P-A/19186 W908/pmax 10 bar FESTO - PISTÓN 1

Pistón de doble efecto con una carrera de 50 mm, y un diámetro de émbolo de 10 mm. Rosca del vástago M4. Presión de funcionamiento entre 1.5 bar y 10 bar.

Figura 56. Cilindro normalizado DSNU-10-50-P-A

Telemecanique XUD-J003937/ SCHNEIDER - S1, S4, S5, S6

Detector fotoeléctrico con fibra óptica (difuso), NPN. Actualmente descontinuado por el fabricante.

Figura 57. Detector fotoeléctrico XUD-J003937

SOOF-P-FL-ST-C50-P/553563WS/100mA-pnp FESTO - S2

Detector de posición optoelectrónico mediante una barrera fotoeléctrica ahorquillada. Tipo de luz: rojo. Diámetro mínimo del objeto:0.3 mm. Precisión de 0.03 mm.

Figura 58. Barrera fotoeléctrica ahorquillada SOOF-P-FL-ST-C50-P

Nidec 404 603/entstort/08035G/24V

Motor con dirección de rotación bidireccional. Voltaje nominal: 24V. Torque nominal: 1 Nm.

Figura 59. Motor Nidec 404 603

ESNU-10-25-P-A/19258 W208/pmax 10 bar FESTO - PISTÓN 2

Pistón de simple efecto con una carrera de 25 mm, y un diámetro de émbolo de 10 mm. Rosca del vástago M4. Presión de funcionamiento entre 1.5 bar y 10 bar.

Figura 60. Cilindro normalizado ESNU-10-25-P-A

SIEN-M5B-PS-S-L/PNP/UO W7B/150371 FESTO - S3

Sensor inductivo con una distancia de conmutación nominal de 0.8 mm. Precisión de 0.01 mm.

Figura 61. Sensor de proximidad SIEN-M5B-PS-S-L

ADNGF-12-40-P-A/537123 W408/pmax 10 bar FESTO - PICK & PLACE

Cilindro compacto con una carrera de 40 mm, y un diámetro de émbolo de 12 mm. Presión de funcionamiento entre 1.5 bar y 10 bar.

Figura 62. Cilindro compacto ADNGF-12-40-P-A

ADNGF-20-60-P-A/554228 W408/pmax 10 bar

Cilindro compacto con una carrera a elegir desde 3 mm a 200 mm, y un diámetro de émbolo de 20 mm. Presión de funcionamiento entre 1.5 bar y 10 bar.

Figura 63. Cilindro compacto ADNGF-20-?-P-A

VUVB-S-M42-AZD-QX-1C1/ 537534

Electroválvula, no se encuentra información ya que el producto ha sido descatalogado. Se recomienda reemplazarlo por la electroválvula VUVS-L20-M52-AZD-G18-F7-1C1/ 575676 de FESTO.

VUVB-S-B42-ZD-QX-1C1/ 537535 W408/pmax 10 bar FESTO

Electroválvula, no se encuentra información ya que el producto ha sido descatalogado. Se recomienda reemplazarlo por la electroválvula VUVS-L20-B52-ZD-G18-F7-1C1/ 575683 de FESTO.

G(A)36, Pressure Gauge for General Purpose (O.D. 37)

Manómetro de uso general con indicador de límite. Rango de presión de 0 a 1.5 MPa, precisión de indicación del 3% FS.

A continuación, se detallan los módulos que componen el PLC empleado para este Trabajo de Fin de Grado.

Controlador PFC200 (PAC); 2ª Generación; 2 x ETHERNET, RS-232/-485

Programación según IEC-61131-3. Conexión directa de módulos de E/S de WAGO. 2 x ETHERNET (configurable), RS-232/485. Sistema operativo Linux con RT-Preempt Patch. Configuración a través de CODESYS®, e!COCKPIT o la interfaz de Web-Based Management. Sin mantenimiento.

Figura 64. Controlador PFC200 (PAC)

***Módulos de entrada**: Item no. 750-400 / Entrada digital, 2 canales / WAGO, Item no. 750-403 / Entrada digital, 4 canales / WAGO y Item no. 750-409 / Entrada digital, 4 canales / WAGO*

Estos módulos de entrada digital reciben señales de control de los dispositivos de campo (p. ej., sensores). Cada módulo de entrada cuenta con un filtro de eliminación de ruido. Los niveles de campo y sistema están separados galvánicamente.

Figura 65. Módulos 750-400, 750-403 y 750-409 respectivamente

***Módulos de salida**: Item no. 750-501 / Salida digital, 2 canales / WAGO y Item no. 750-504 / Salida digital, 4 canales / WAGO*

Este módulo de salida digital transmite señales de control desde el componente de automatización a los actuadores conectados. Todas las salidas están protegidas contra cortocircuito. Los niveles de campo y sistema están separados galvánicamente.

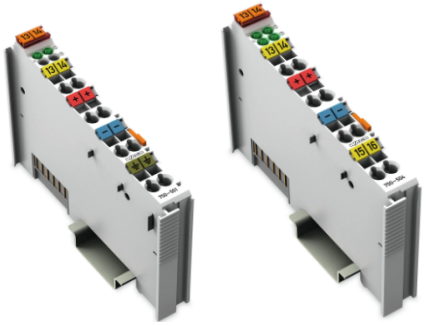


Figura 66. Módulos 750-501 y 750-504 respectivamente

Item no. 750-600 / Módulo final / WAGO

Una vez montados el nodo de bus de campo con el acoplador de bus y los módulos de E/S correctos, se acopla el módulo final.

Figura 67. Módulo 750-600

El resultado de la unión de los módulos es el mostrado en la Figura 72. De esta forma, se completa el circuito de datos interno y garantiza la correcta transmisión de datos.

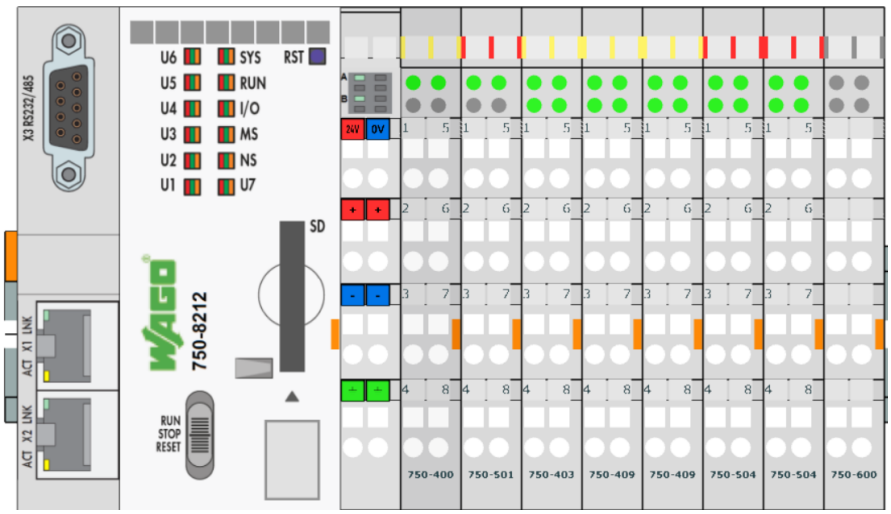
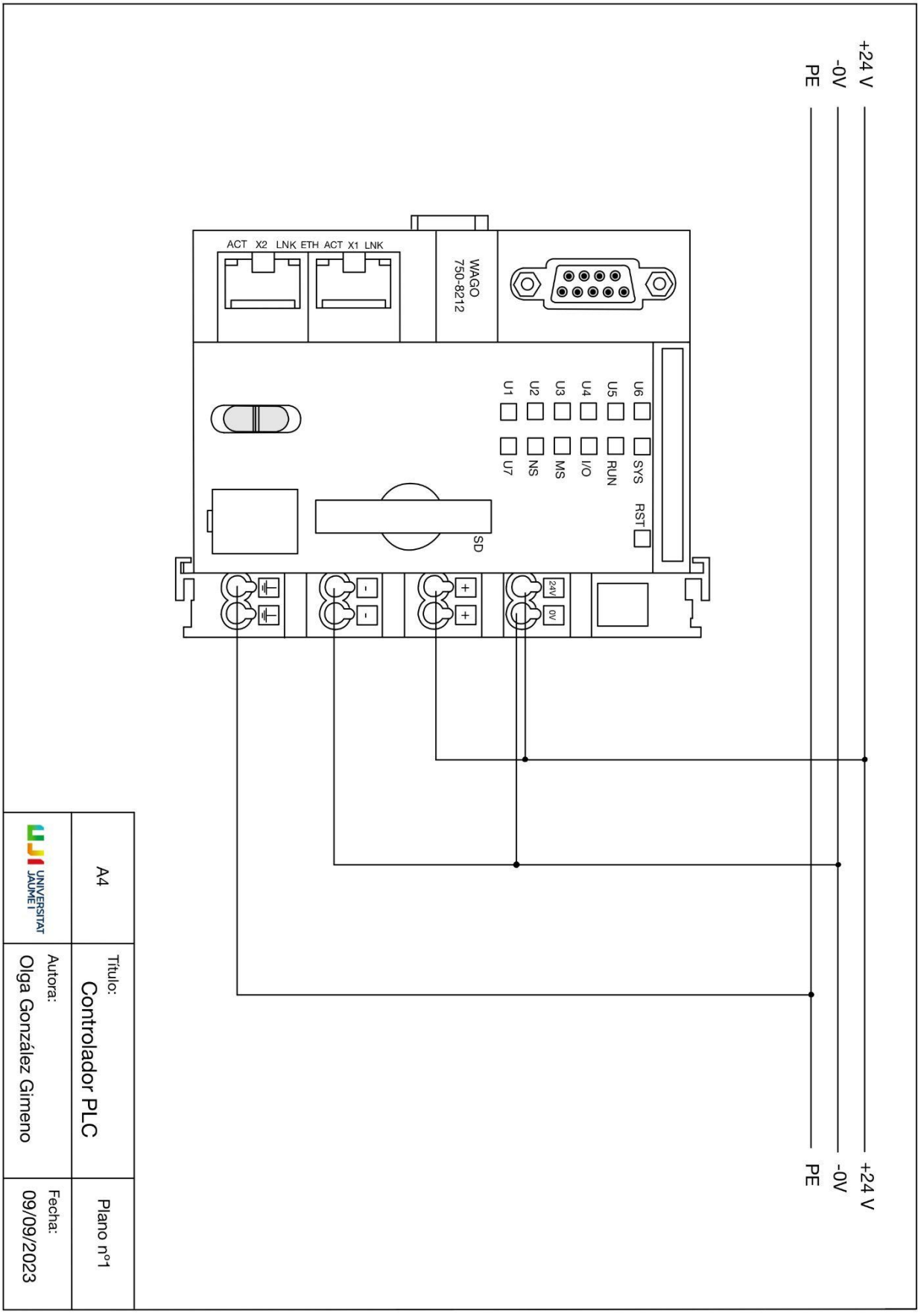
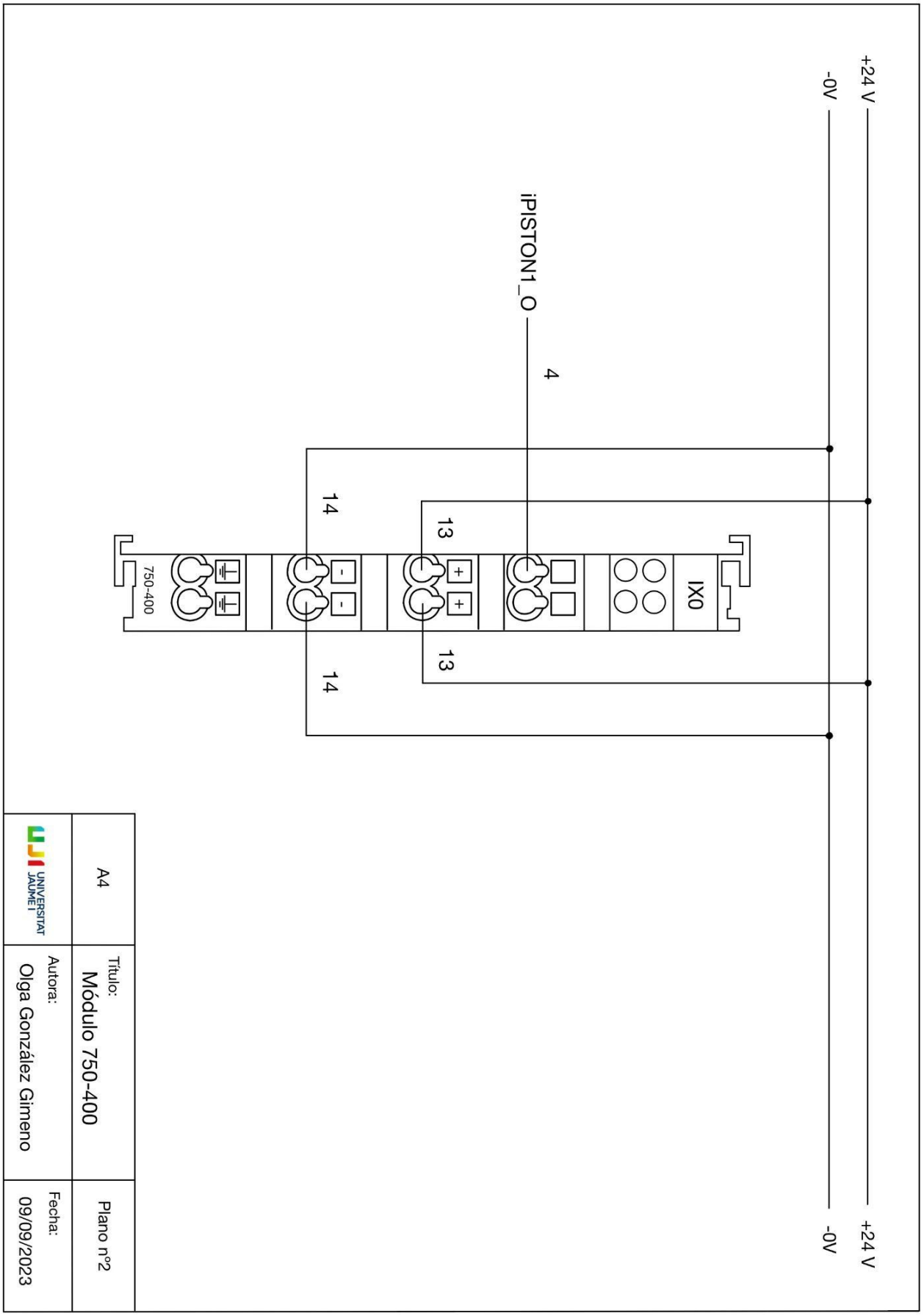


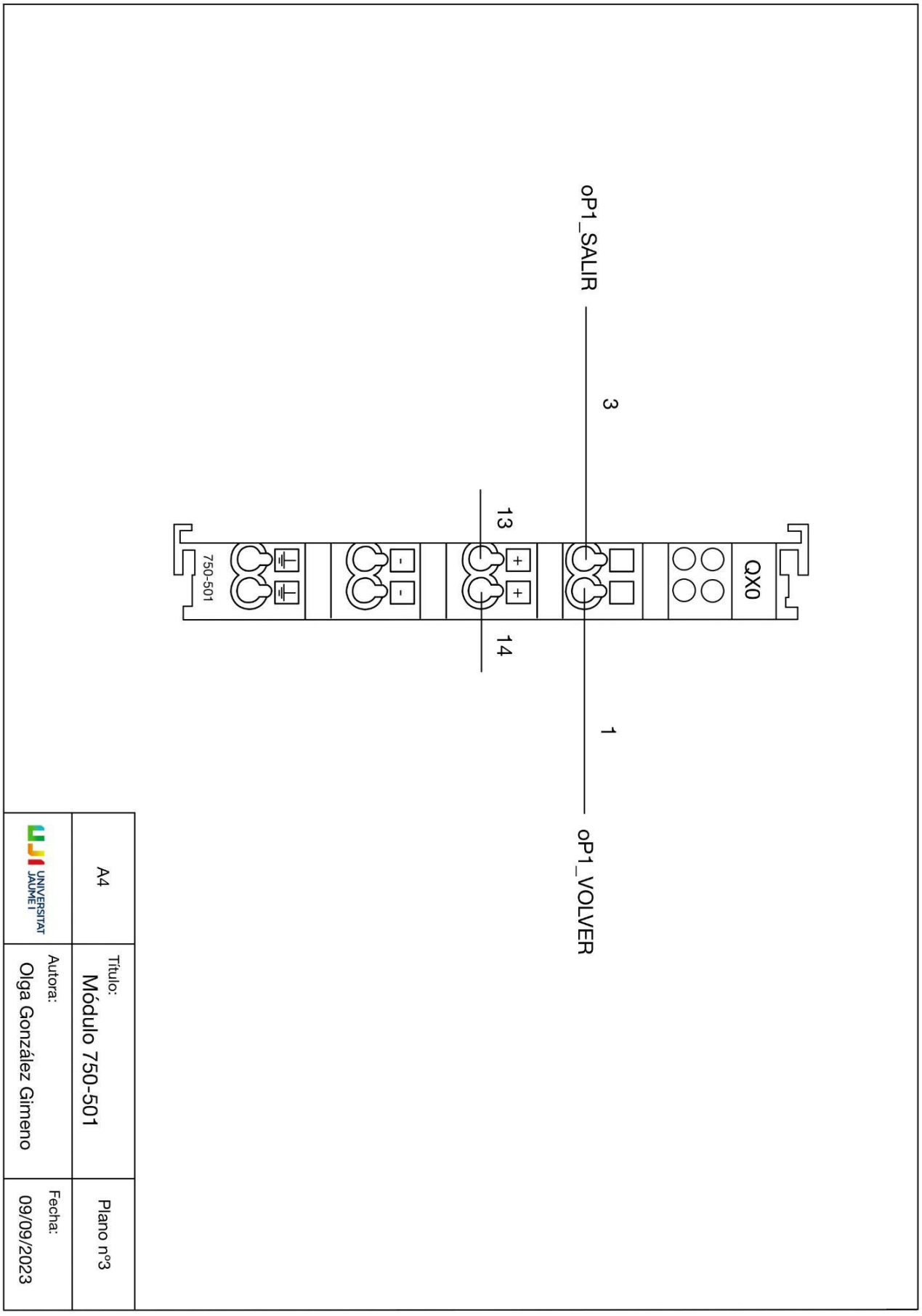
Figura 68. Imagen completa del PLC

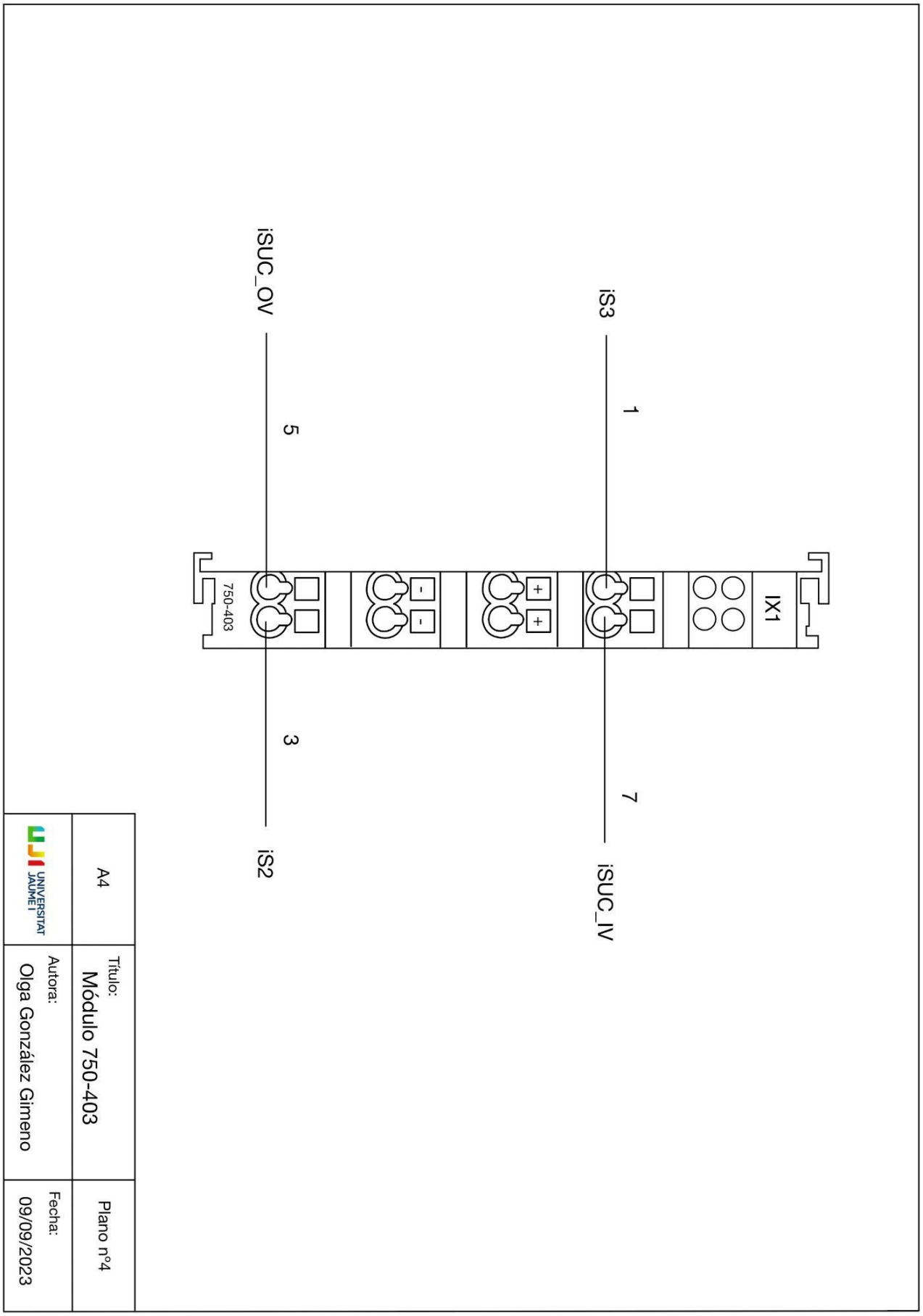
3.Planos

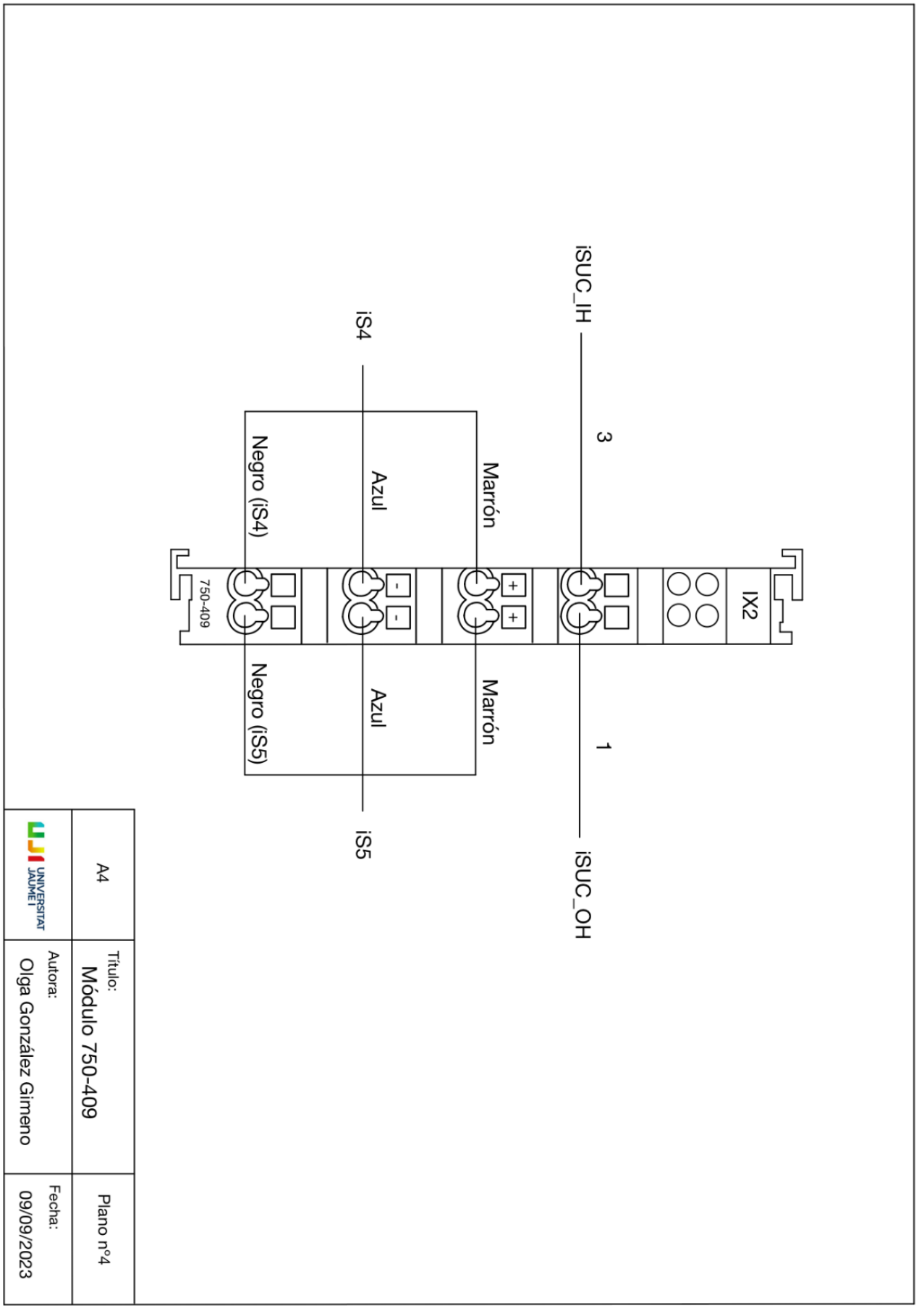


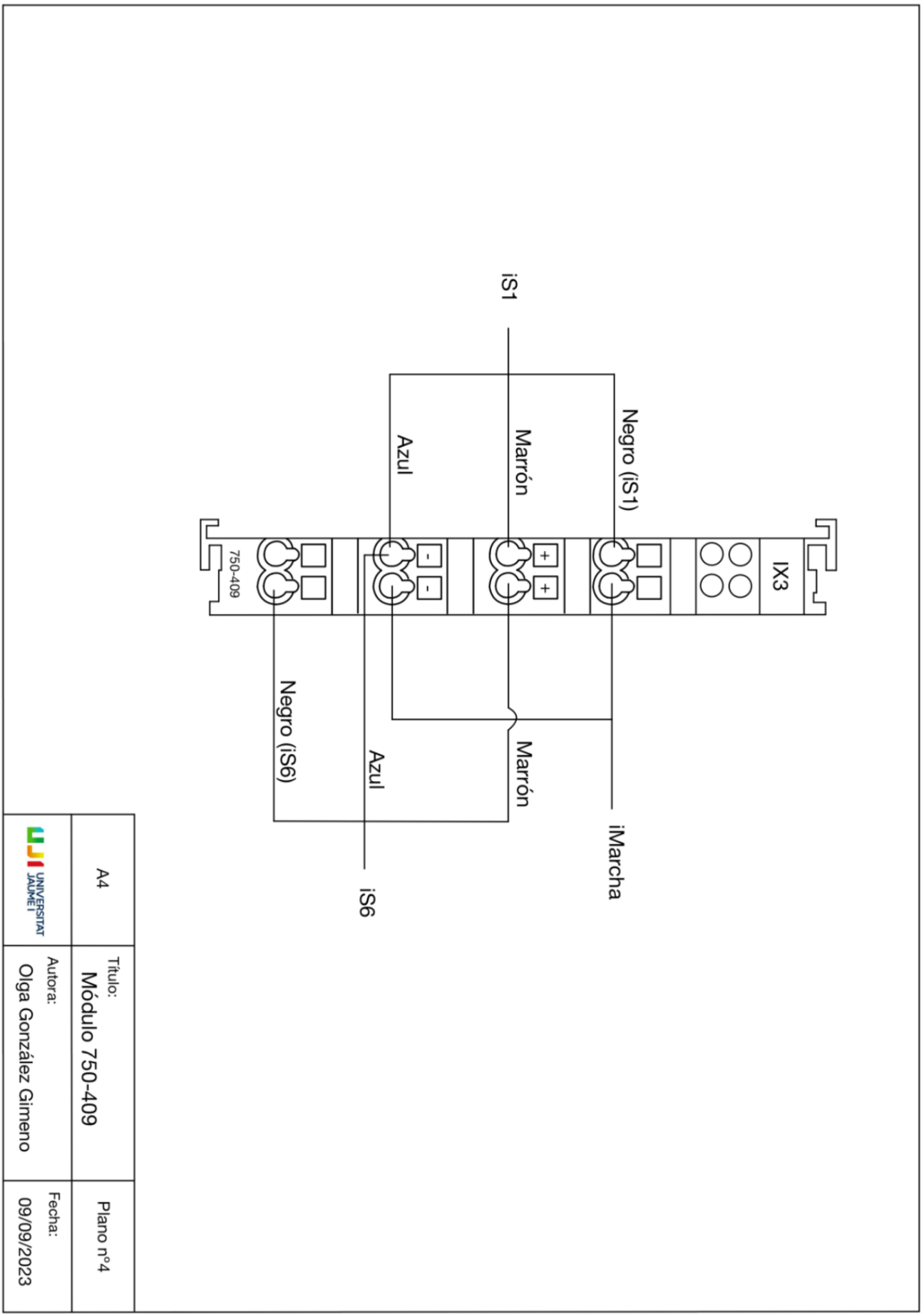
A4	
Título: Controlador PLC	
Autora: Olga González Gimeno	
Fecha: 09/09/2023	Plano nº1


+24 V
-0V
iPISTON1_O
4
14
13
750-400
IX0
13
14
-0V
+24 V
A4
Título:
Módulo 750-400
Plano nº2
Autora:
Olga González Gimeno
Fecha:
09/09/2023
UNIVERSITAT JAUME I

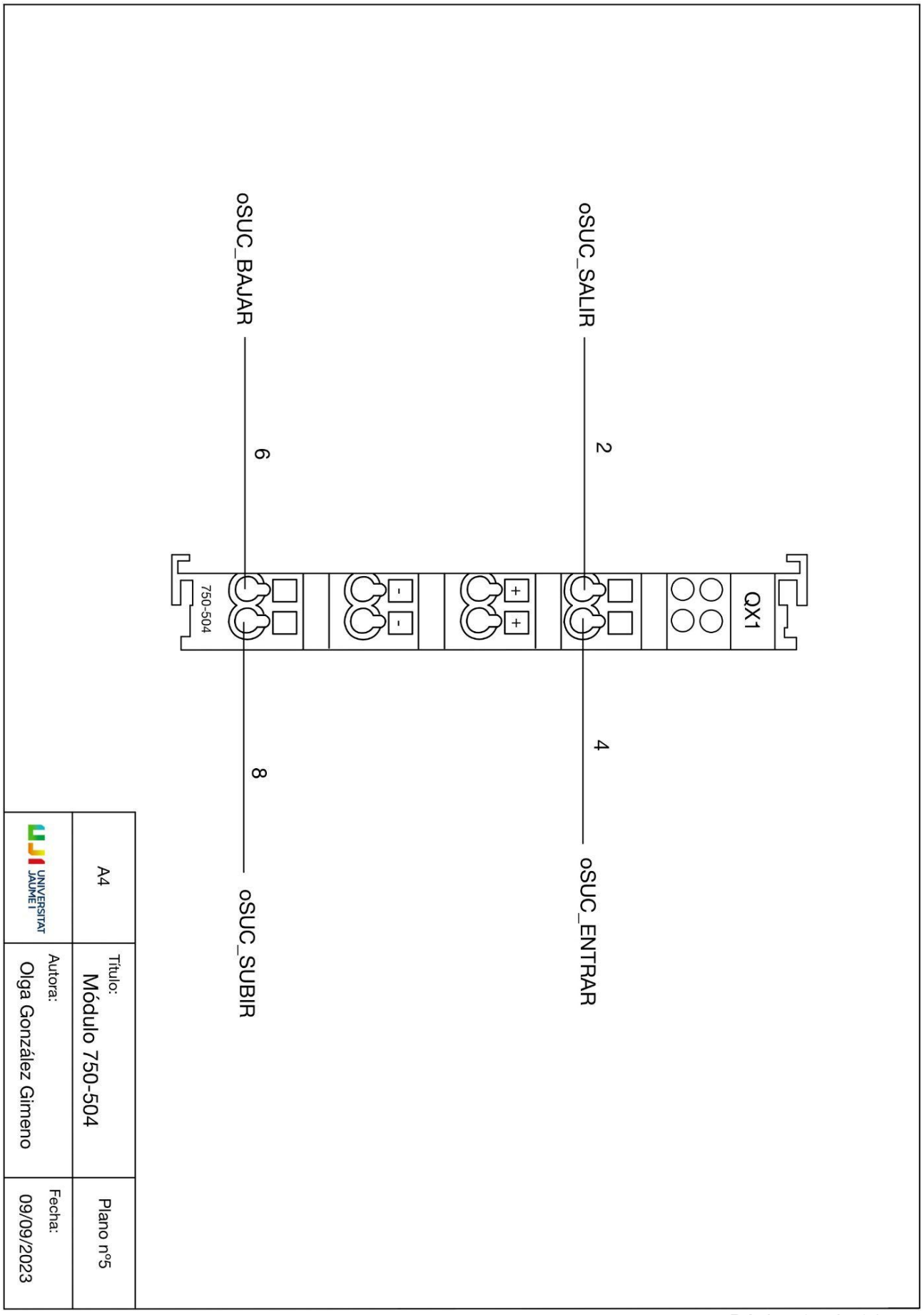

oP1_SALIR
3
13
14
1
oP1_VOLVER
QX0
750-501
A4
Título:
Módulo 750-501
Plano nº3
Autora:
Olga González Gimeno
Fecha:
09/09/2023


iSUC_OV
iS3
1
5
3
7
iS2
iSUC_IV
750-403
IX1
A4
Título:
Módulo 750-403
Plano nº4
Autora:
Olga González Gimeno
Fecha:
09/09/2023

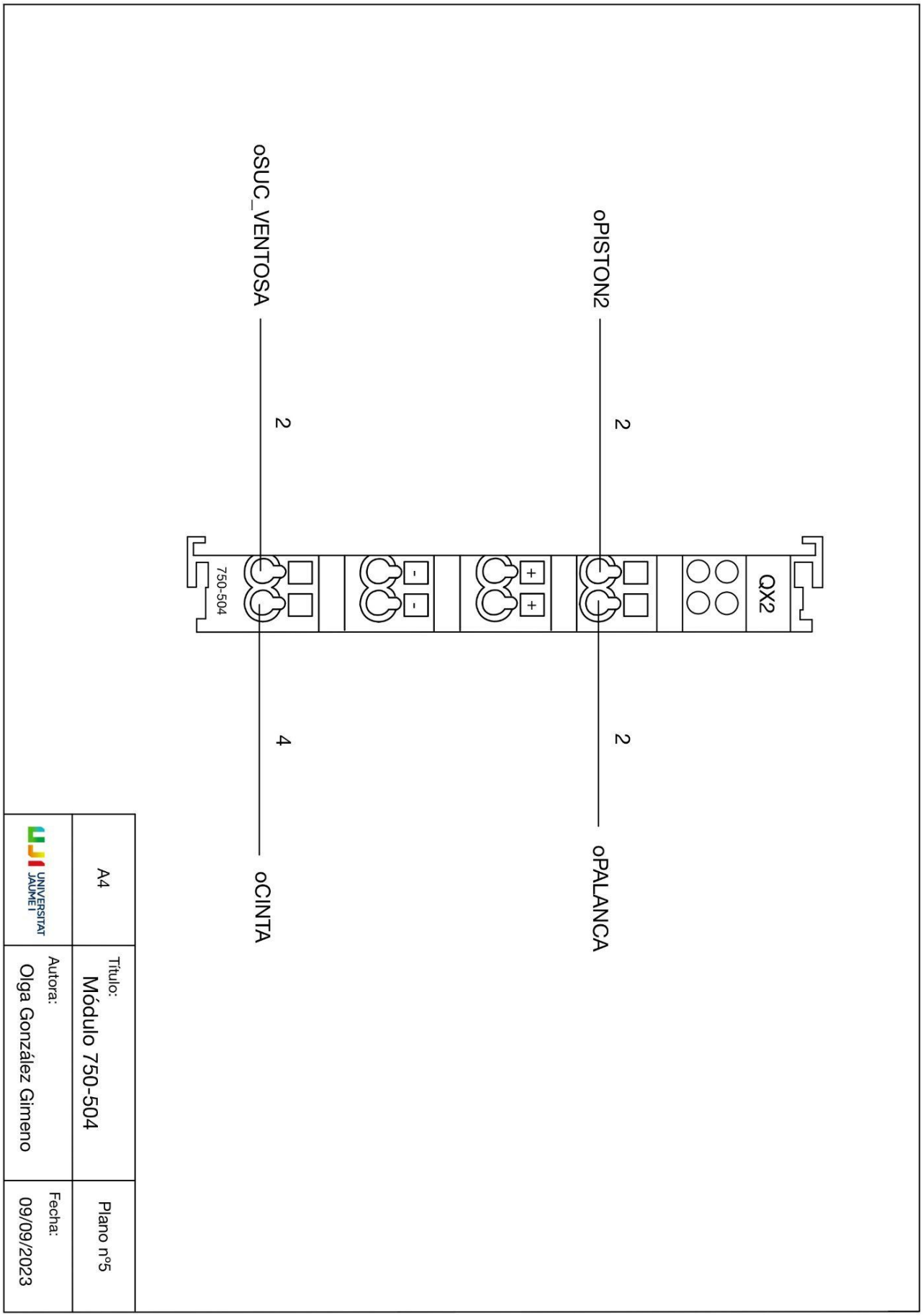

iSUC_IH
3
iS4
Negro (iS4)
Azul
Marrón
IX2
750-409
Negro (iS5)
Azul
Marrón
1
iS5
iSUC_OH
A4
Título:
Módulo 750-409
Plano nº4
Autora:
Olga González Gimeno
Fecha:
09/09/2023



A4	Título: Módulo 750-409	Plano nº4
UNIVERSITAT JAUME I	Autora: Olga González Gimeno	Fecha: 09/09/2023



UNIVERSITAT JAUME I	A4	Plano nº5
Autora: Olga González Gimeno	Título: Módulo 750-504	Fecha: 09/09/2023



A4	Título: Módulo 750-504	Plano nº5
UNIVERSITAT JAUME I	Autora: Olga González Gimeno	Fecha: 09/09/2023

4.Pliego de condiciones

4.1.Especificaciones de hardware

A continuación se listan todos los dispositivos de hardware utilizados. Comenzando por las características de ordenador:

- El ordenador está equipado con un procesador Intel(R) Core(TM) i5-8400. Este procesador ofrece un rendimiento sólido con una velocidad base de 2.80 GHz y una velocidad turbo de hasta 2.81 GHz.
- La memoria RAM instalada en el sistema es de 8,00 GB, de los cuales 7,88 GB son utilizables.
- El sistema operativo se ejecuta en una arquitectura de 64 bits, con procesador basado en x64.
- El sistema operativo preinstalado en el ordenador es Windows 10 Home. La versión específica es la 22H2.

Seguidamente se nombran los elementos integrados en la maqueta, para una explicación más detallada de los componentes se puede consultar el Anexo 2.7 o MecLab® Mechatronics Training System.

Elementos comunes a todos los módulos:

— Distribuidor de enchufe multipolo

— Manómetro

— Piezas metálicas y de plástico

— Placa de perfil de aluminio

Módulos de distribución y estampado:

— Módulo de almacén en pila

— 2 válvulas solenoides

— 2 cilindros

— 1 interruptor de láminas magnético

Módulo de clasificación:

— Cinta transportadora con motor CC

— Solenoide como tope/deflector

— Sensor inductivo

— Sensor óptico (barrera de luz)

Módulo de procesado manual:

— 3 válvulas solenoides

— 4 finales de carrera magnéticos

— 2 cilindros neumáticos con guía de rodamiento simple

— 1 pinza neumática

En lo que al PLC se refiere, tal y como se ha visto previamente, será necesario un controlador, y un conjunto de módulos de modo que haya un total de 12 canales de entrada y 10 canales de salida. Se incluye además un “módulo final", que no debe tener ninguna conexión.

4.2.Especificaciones de software

Se enumeran a continuación las versiones de los programas informáticos empleados en el presente proyecto::

- **Factory I/O Modbus & OPC Edition.** Cualquier versión que contenga el protocolo OPC UA/DA resultará adecuada para este proyecto.
- **CODESYS® V3.5 SP17.30**. También se consideran aptas las variantes más recientes. Requiere la instalación de paquetes adicionales.
    - **CODESYS® Control for PFC200 SL 4.8.0.0**. Este componente permite al usuario programar en diversos modelos de PFC200 de WAGO. Es compatible con CODESYS® V3 o sus versiones subsiguientes.
- **BootP-DHCP Tool.**

4.3.Especificaciones de ejecución

Para llevar a cabo la ejecución simultánea de la aplicación desarrollada en los sistemas virtual y digital, se deben seguir los siguientes procedimientos:

1. Ajustar la dirección IP del dispositivo PLC conectado mediante la interfaz Ethernet, de manera que esté alineada con una dirección que pertenezca a la misma subred que la del ordenador. Para lograr esto, se emplea la herramienta BootP-DHCP Tool como referencia, siguiendo las directrices del apartado 1.7.3.2 del documento.

2. Iniciar la plataforma CODESYS®, llevar a cabo un escaneo de los dispositivos disponibles y acceder al dispositivo PLC seleccionado mediante el ingreso de las credenciales de usuario y contraseña previamente establecidas.

3. Si se han realizado modificaciones al programa que involucren la eliminación o adición de variables, se debe actualizar la lista de "symbol configuration". Si no se han efectuado modificaciones, este paso no será necesario.

4. Transferir el programa diseñado al PLC y activar su ejecución.

5. Configurar los parámetros de comunicación en el entorno Factory I/O, asegurándose de que se seleccione el protocolo OPC AU/DA. Es importante ingresar la dirección del dispositivo PLC y establecer la correspondencia entre cada variable de entrada/salida y su contraparte virtual.

6. Dar inicio a la simulación en Factory I/O clicando el triángulo ubicado en la barra de herramientas.

5.Mediciones

Se definen a continuación las distintas partes o unidades de obra en las que se divide el proyecto realizado.

5.1.Reemplazos y actualizaciones

Para poder llevar a cabo la implementación del programa de control tanto sobre la maqueta como sobre el gemelo digital, se debe evaluar el estado de los componentes disponibles, y en caso de ser necesario reemplazar componentes de la maqueta y/o actualizar elementos del software.

5.2.Trabajos electrónicos

En esta parte del trabajo se incluyen los trabajos relativos al sistema físico, desde el conexionado de la maqueta con el PLC, hasta la comunicación de dicho PLC con el ordenador en que se programa el software de control. De modo que se realiza el conexionado de los actuadores y sensores de la maqueta con los módulos de entrada y salida del controlador. Si bien los módulos y la maqueta no suponen costes adicionales para esto proyecto (ya habían sido comprados para proyectos anteriores), se incluirán en el presupuesto para dar una idea clara del coste que supondría recrear este trabajo.

5.3.Trabajos de programación

Esta sección cubre todo el proceso de elaboración del código de programación, desde el modelado hasta la implementación de los distintos programas. Se realizarán varias pruebas sobre el gemelo digital y la maqueta, que permitan validar el software, analizando los resultados obtenidos y modificando el programa en función a estos, con el fin de alcanzar un código mejorado.

6.Presupuesto

6.1.Costes derivados de reparaciones/actualizaciones

Componente	Unidades	Precio/unidad (€)	Precio total (€)
Controlador PFC200 (PAC) 750-8212; 2ª Generación; 2 x ETHERNET, RS-232/-485	1	294	294
Junta tórica de nitrilo 182x190x4	2	14,5	29
TOTAL:			**323**

Tabla 4. Costes derivados de reparaciones/actualizaciones

6.2.Costes derivados de los trabajos electrónicos

Componente	Descripción	Unidades	Precio/unidad (€)	Precio total (€)
Pack MecLab FESTO	Maqueta	1	2480	2480
PC de sobremesa		1	1000	1000
Item no. 750-400 / Entrada digital, 2 canales	Módulo PLC	1	32	32
Item no. 750-501 / Salida digital, 2 canales	Módulo PLC	1	32	32
Item no. 750-403 / Entrada digital, 4 canales	Módulo PLC	1	32	32
Item no. 750-409 / Entrada digital, 4 canales	Módulo PLC	2	32	64
Item no. 750-504 / Salida digital, 4 canales	Módulo PLC	2	32	64
Item no. 750-600 / Módulo final	Módulo PLC	1	55	55
Otros (bornera, cables, etc.)		-	-	300
Trabajos de electrónica		50 (horas)	30 (€/hora)	1500
TOTAL:				**5559**

Tabla 5. Costes derivados de trabajos electrónicos

6.3.Costes derivados de los trabajos de programación

Componente	Unidades	Precio/unidad (€)	Precio total (€)
CODESYS® Control for PFC200 SL 4.8.0.0	1	110	110
Licencia Factory I/O Modbus & OPC Edition	1	158	158
Trabajos relativos a la programación, tests y validación	250 (horas)	30 (€/hora)	7500
TOTAL:			**7768**

Tabla 6. Costes derivados de trabajos de programación

6.4.Presupuesto final

Concepto	Coste (€)
Costes derivados de reemplazos/actualizaciones	296,96
Costes derivados de los trabajos electrónicos	5559
Costes derivados de los trabajos de programación	7768
TOTAL:	13649

Tabla 7. Presupuesto final