

## **TRABAJO DE FIN DE GRADO**

**“Integración del Factory IO y el PLC-Lab con el  
Grafcet Studio para el diseño y la  
implementación de la Guía GEMMA en plantas  
simuladas”**

**Grado de Ingeniería Electrónica Industrial y Automática.**

**Alumna: Nuria Villaverde Blanchard.**

**Tutora: Marta Sigut Saavedra**

**Curso 2023/2024**

# ÍNDICE

<b>Resumen.....</b>	<b>4</b>
<b>Abstract.....</b>	<b>5</b>
<b>1. Introducción.....</b>	<b>6</b>
1.1. Objetivos del TFG.....	6
1.2. Antecedentes.....	6
1.3. Justificación.....	7
<b>2. Descripción de la planta Festo.....</b>	<b>8</b>
2.1. Estación cero: almacén y distribución de piezas.....	8
2.2. Estación uno: test.....	10
2.3. Estación dos: procesado de piezas.....	13
2.4. Estación tres: acarreo de piezas.....	15
2.5. Estación cuatro: clasificación de piezas.....	17
<b>3. Herramientas de diseño y programación.....</b>	<b>20</b>
3.1. Guía GEMMA.....	20
3.2. Grafcet.....	23
<b>4. Herramientas software empleadas para el modelado de la planta Festo y su programación en GRAFCET.....</b>	<b>34</b>
4.1. Modelado de plantas.....	34
4.1.1. Factory I/O.....	34
4.1.2. PLC-Lab.....	44
4.1.3. PLC-Lab 3D Player.....	82
4.2. Programación en GRAFCET: Grafcet Studio.....	84
<b>5. Integración del Factory I/O con el Grafcet Studio para la implementación de la Guía GEMMA en las estaciones 0-2 de la planta Festo.....</b>	<b>89</b>
5.1. Diseño de la Guía GEMMA de la estación cero.....	91
5.1.1 Sensores y Actuadores definidos en la estación cero.....	91
5.1.2 Estados implementados de la Guía GEMMA.....	96
5.1.3 Estados NO implementados de la Guía GEMMA.....	99
5.1.4 Pupitre de control de la estación 0.....	100
5.1.5 Guía GEMMA de la estación cero.....	102
5.2. Diseño de la Guía GEMMA de la estación uno.....	102
5.2.1 Sensores y Actuadores definidos en la estación uno.....	102
5.2.2 Estados implementados de la Guía GEMMA.....	108
5.2.3 Estados NO implementados de la Guía GEMMA.....	112
5.2.4 Pupitre de control de la estación 1.....	113
5.2.5 Guía GEMMA de la estación uno.....	114
5.3. Diseño de la Guía GEMMA de la estación dos.....	115
5.3.1 Sensores y Actuadores definidos en la estación dos.....	115
5.3.2 Estados implementados de la Guía GEMMA.....	121
5.3.3 Estados NO implementados de la Guía GEMMA.....	125
5.3.4 Pupitre de control de la estación dos.....	126
5.3.5 Guía GEMMA de la estación dos.....	127

5.4. Programación de la Guía GEMMA con Grafcet Studio.....	127
<b>6. Modelado de las estaciones de la planta Festo con el PLC-Lab.....</b>	<b>129</b>
6.1. Estación cero.....	129
6.2. Estación uno.....	132
6.3. Estación dos.....	136
6.4. Estación tres.....	143
6.5. Estación cuatro.....	147
6.6. Comparativa entre el Factory I/O y el PLC-Lab.....	151
<b>7. Integración del PLC-Lab con el Grafcet Studio para la programación de las estaciones de la planta Festo.....</b>	<b>152</b>
7.1. Estación cero (funcionamiento normal y emergencia).....	153
7.2. Estación uno (funcionamiento normal y emergencia).....	154
7.3. Estación dos (Guía GEMMA).....	155
7.4. Estación tres (funcionamiento normal y emergencia).....	156
7.5. Estación cuatro (funcionamiento normal y emergencia).....	158
<b>8. Presupuesto.....</b>	<b>160</b>
<b>9. Conclusiones y propuestas de mejora.....</b>	<b>161</b>
<b>Referencias.....</b>	<b>164</b>

## Resumen

En el presente Trabajo de Fin de Grado (TFG) se propone la integración del Grafcet Studio, un programa para la programación en Grafcet, con varias herramientas de modelado de plantas industriales.

La primera parte de este proyecto se ha centrado en el diseño y la programación de la Guía GEMMA para las estaciones cero, uno y dos de la planta Festo de procesado de piezas que se encuentra disponible en el laboratorio Lorenzo Moreno, del departamento de Ingeniería Informática y de Sistemas. En lugar de con las estaciones físicas, se ha trabajado con los modelos de las mismas desarrollados en un TFG previo, aunque se han introducido algunas modificaciones que se describen en este documento. La programación de la Guía GEMMA se ha realizado con el programa Grafcet Studio.

En la segunda parte se ha analizado con profundidad otra herramienta para el modelado y la creación de escenas: el software PLC-Lab. Con este programa se han creado los modelos de las cinco estaciones de la planta Festo, lo que ha permitido comparar el Factory IO y el PLC-Lab, identificando las ventajas y desventajas de cada herramienta y su adecuación para distintas aplicaciones industriales. Los modelos creados con PLC-Lab también se han programado con Grafcet Studio. Para las estaciones cero, uno, tres y cuatro, únicamente se ha programado el funcionamiento normal y la parada de emergencia, mientras que para la estación dos se ha programado la Guía GEMMA.

Este proyecto no solo busca explorar las capacidades de los programas utilizados, sino también evaluar su potencial educativo para su implementación en las asignaturas de automatización, tanto de grado como de máster, adscritas al departamento de Ingeniería Informática y de Sistemas.

## **Abstract**

In this Final Degree Project (TFG), the integration of Grafcet Studio, a program for programming in Grafcet, with various industrial plant modeling tools, is proposed.

The first part of this project focused on the design and programming of the GEMMA Guide for stations zero, one, and two of the Festo part processing plant, available in the Lorenzo Moreno laboratory of the Department of Computer Engineering and Systems. Instead of using the physical stations, work was carried out with models of these stations developed in a previous TFG, although some modifications were introduced, which are described in this document. The GEMMA Guide programming was done with the Grafcet Studio program.

In the second part, another tool for modeling and creating scenes, the PLC-Lab software, was thoroughly analyzed. With this program, models of the five stations of the Festo plant were created, allowing a comparison between Factory IO and PLC-Lab, identifying the advantages and disadvantages of each tool and their suitability for different industrial applications. The models created with PLC-Lab were also programmed with Grafcet Studio. For stations zero, one, three, and four, only normal operation and emergency stop were programmed, while for station two, the GEMMA Guide was programmed.

This project aims not only to explore the capabilities of the programs used, but also to evaluate their educational potential for implementation in automation courses, both at the undergraduate and master's levels, within the Department of Computer Engineering and Systems.

# 1. Introducción

## 1.1. Objetivos del TFG

En este Trabajo de Fin de Grado se han planteado dos grandes objetivos:

1. El uso conjunto del Grafcet Studio (software para la programación en Grafcet) y el Factory IO (software para el modelado de plantas industriales) para el diseño y la programación de la Guía GEMMA en tres estaciones de procesamiento de piezas.

2. Proponer alternativas al Factory IO de cara a obtener modelos simulados más parecidos a las estaciones de la planta Festo, concretamente los programas PLC-Lab y PLC-Lab 3D Player. También se propone la integración del primero con el Grafcet Studio.

## 1.2. Antecedentes

Para el desarrollo de este Trabajo de Fin de Grado se ha utilizado como referencia el trabajo realizado en dos TFGs previos desarrollados por dos alumnos del grado de Ingeniería Electrónica Industrial y Automática. Estos trabajos han sido fundamentales para la comprensión y aplicación de los programas Factory IO y Grafcet Studio.

El primer trabajo de referencia es el titulado “Diseño y programación de plantas virtuales de laboratorio para asignaturas de Automatización”, desarrollado por Cristian José Luis González [1]. Este trabajo ha proporcionado conocimientos sobre el uso de Factory IO, así como de los modelos de las estaciones de la planta Festo que sirvieron como base para realizar este trabajo.

El segundo trabajo de referencia es el titulado “ Grafcet-Studio : Una herramienta para la Implementación de la Guía GEMMA en la Planta Festo”, realizado por Luis Gerardo Rodríguez Pérez. Este trabajo fue crucial para entender el diseño y la implementación de la Guía GEMMA, así como el uso de la herramienta Grafcet-Studio. No se incluye la referencia debido a que este TFG no se ha encontrado en el repositorio institucional de la ULL.

Estos antecedentes han enriquecido el marco teórico y práctico de este proyecto, proporcionando orientación y metodologías que se han adaptado y ampliado en el presente trabajo de fin de grado.

### **1.3. Justificación**

En la actualidad, la automatización industrial es un sector con un gran crecimiento y relevancia, ya que permite incrementar la productividad y flexibilizar las herramientas, producir con calidad constante, reducir costos y aumentar la eficiencia y seguridad en entornos de producción [2].

La automatización industrial con simuladores como Factory IO , PLC-Lab u otras alternativas es un enfoque efectivo para diseñar y probar sistemas de automatización industrial de manera virtual. Estos simuladores ofrecen beneficios como reducir costos y mejorar la eficiencia, así como facilitar la capacitación de ingenieros y técnicos en el diseño e implementación de proyectos de automatización [3].

Tanto el Factory IO, u otras alternativas como las presentadas en este trabajo, como el Grafcet Studio son herramientas software de gran utilidad para la docencia en el ámbito de la automatización industrial, lo que garantiza que los estudiantes aprendan a utilizar herramientas que encontrarán en el ámbito laboral. Ambas herramientas son altamente flexibles y pueden adaptarse a una amplia variedad de escenarios industriales. Permiten una reducción de los errores ya que, al utilizar simulaciones realistas lo más precisas posibles, se pueden identificar y corregir errores antes de la implementación real, lo que podría reducir los costos asociados con fallos en el sistema.

Este Trabajo de Fin de Grado propone integrar tanto el Factory IO como el PLC-Lab con el Grafcet Studio para el diseño y la implementación de la Guía GEMMA en plantas simuladas. La justificación de esta elección radica en el potencial del uso conjunto de estas herramientas para mejorar la formación de los estudiantes en el ámbito de la automatización industrial.

## 2. Descripción de la planta Festo

La planta Festo es una planta a pequeña escala dedicada al procesamiento de piezas, que abarca desde la extracción de las piezas de un almacén hasta su clasificación.

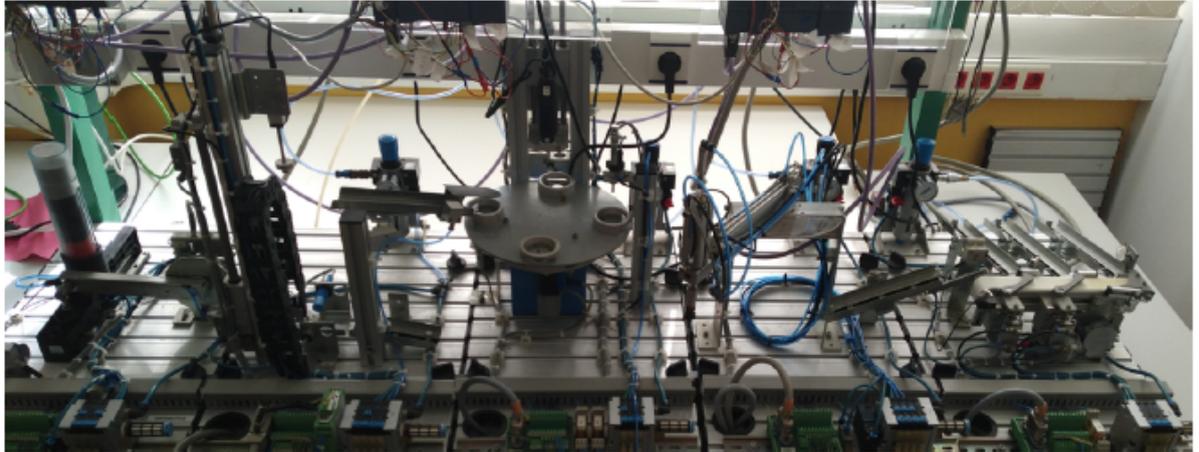


Figura 1. Planta Festo del Laboratorio Lorenzo Moreno Ruíz en la Escuela Superior de Ingeniería y Tecnología de la Universidad de La Laguna.

Cada una de las cinco estaciones que la componen cumple una función específica que se describe a continuación. En este TFG se ha diseñado e implementado la Guía GEMMA de las tres primeras estaciones (estaciones cero, uno y dos). Además, se han realizado los modelos de las cinco estaciones con el PLC-Lab. Las estaciones de la planta real cuentan con autómatas programables S7-1200 o S7-1500, sin embargo, en nuestro caso, no son utilizados ya que se ha trabajado exclusivamente con los modelos simulados de la planta.

### 2.1. Estación cero: almacén y distribución de piezas

Esta estación es la encargada de extraer las piezas del almacén donde se encuentran apiladas y transferirlas mediante un brazo rotativo que está equipado con una ventosa. Además, dispone de un pistón que se encarga de la extracción de las piezas.

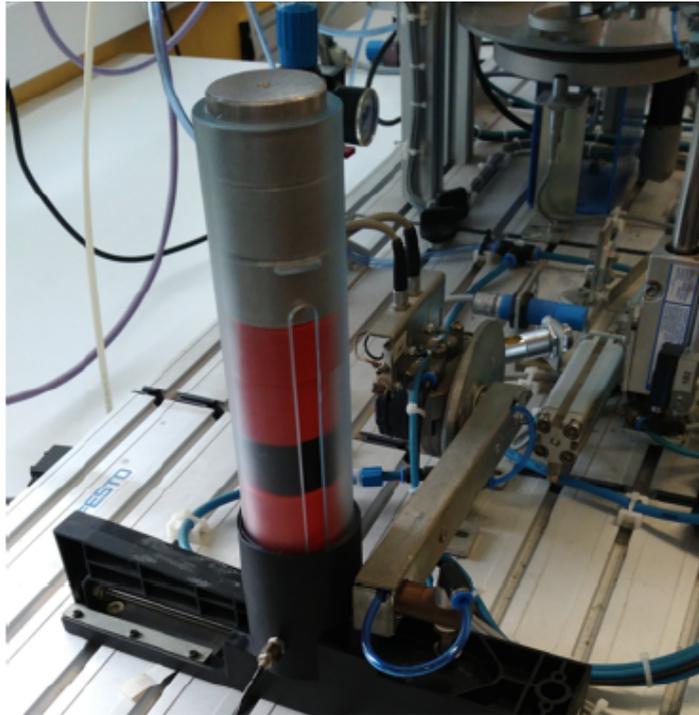


Figura 2. Estación Cero de la Planta Festo

En este caso la estación se controla con un autómatas S7-1200. Las condiciones iniciales que deben cumplirse para que esta estación comience a operar cuando el operario accione el botón START son las siguientes:

- ❖ Almacén de piezas lleno ( tiene una capacidad para 8 piezas).
- ❖ Pistón de expulsión extendido.
- ❖ Ventosa apagada (OFF).
- ❖ Brazo intercambiador en la posición del almacén.

El funcionamiento de la estación cero consiste en la realización secuencial de las siguientes operaciones:

1. El brazo intercambiador gira hacia la estación 1.
2. El pistón de expulsión empuja una pieza fuera del almacén (para esto debe retraerse).
3. El brazo intercambiador gira hacia la estación 0 (posición del almacén).
4. Se activa la ventosa.
5. Se espera un segundo para que la pieza se sujete.
6. El pistón de expulsión vuelve a su posición inicial (la cual es extendida).
7. El brazo intercambiador gira llevando la pieza a la estación 1.

8. Una vez llega a dicha estación se ‘sopla’ la pieza para depositarla en la posición 1 de la estación 1. De forma simultánea, el pistón de expulsión debe estar retrayendo para empujar una nueva pieza fuera del almacén.

A partir de aquí, se repite la secuencia desde el paso 3 hasta que el almacén se vacíe, para lo cual se debe llevar la cuenta del número de piezas que han salido del almacén.

## 2.2. Estación uno: test

La función de la estación uno es medir el tamaño de cada pieza. Si la pieza que se mide tiene un tamaño considerado ‘normal’, se envía a la siguiente estación, que es la estación 2. Sin embargo, si la pieza es más grande o más pequeña de lo ‘normal’, y por lo tanto, se descarta. En este caso la estación está controlada por un autómeta S7-1500.

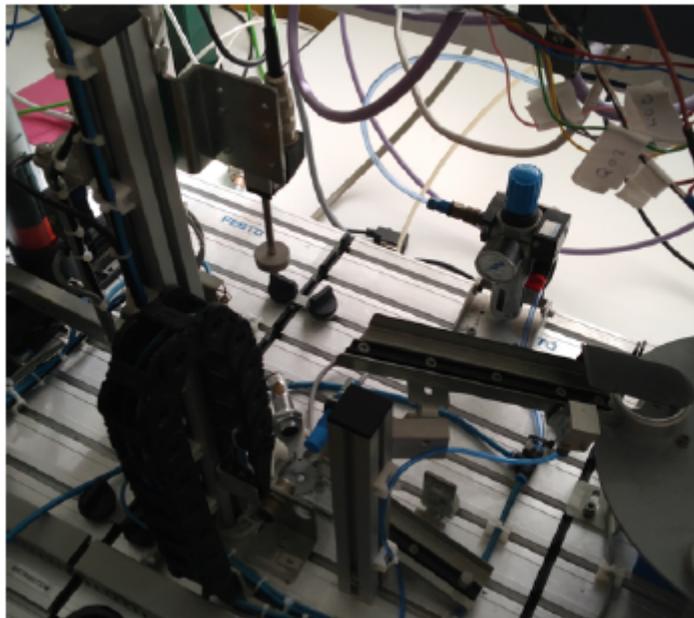


Figura 3. Estación 1 de la Planta Festo.

Esta estación está formada por los siguientes elementos:

- ❖ Una plataforma elevadora.
- ❖ Dos rampas:
  - Una que lleva las piezas a la estación 2.
  - Otra que se encarga de desechar las piezas defectuosas.

- ❖ Sensor que detecta la presencia de un objeto en la posición 1.
- ❖ Sensor que mide la longitud de las piezas.
- ❖ Un motor de doble sentido para subir y bajar la plataforma.
- ❖ Otro motor de doble sentido para subir y bajar el sensor que mide la longitud de las piezas.
- ❖ Un pistón que empuja las piezas de la plataforma a las rampas.
- ❖ Una pestaña que retiene las piezas de tamaño normal en la rampa que lleva las piezas hacia la estación 2, hasta que ésta esté preparada para recibir las.
- ❖ Dos finales de carrera en las posiciones 1 y 2, correspondientes a la posición del elevador cuando se encuentra abajo o arriba, respectivamente.

Al igual que en la estación anterior, se cuenta con una serie de condiciones iniciales que se exigen para que esta estación comience a funcionar, son las siguientes:

- ❖ La plataforma elevadora debe estar abajo ( lo que corresponde con la posición 1).
- ❖ El pistón debe estar retraído.
- ❖ La pestaña de sujeción en la rampa 2 está retraída.
- ❖ Sensor de longitud retraído.

Una vez se cumplen las condiciones iniciales, se comienza la secuencia de acciones correspondiente:

1. Se sube la plataforma hasta alcanzar la posición 2 (arriba).
2. Cuando llega a esta posición, se detiene.
3. Baja el sensor que mide la longitud de la pieza.
4. Se mide la longitud de la pieza y la medida es guardada.
5. Una vez medida y guardada la longitud, sube el sensor de longitud.

Según la medida que se haya obtenido de la pieza se deben seguir distintos procedimientos:

- ❖ Si la longitud de la pieza es la correcta:
  1. Con la plataforma en la posición 2 (arriba), el pistón se extiende para empujar la pieza hacia la rampa 2.
  2. La pestaña de sujeción es extendida para que retenga la pieza al final de la rampa.
  3. Una vez el pistón se ha extendido, se retrae.

4. La plataforma, que ahora se encuentra vacía, baja hasta la posición 1, donde se detiene a la espera de una nueva pieza.

A partir de este la secuencia de acciones comienza desde el principio.

Cuando la estación dos indique que está preparada para recibir una pieza, se retrae la pestaña de sujeción, permitiendo que la pieza pase de la rampa 2 a la estación 2.

- ❖ Si la longitud de la pieza no es la correcta, considerando entonces la pieza como defectuosa:
  1. Con la plataforma en la posición 2 (arriba), se baja hasta la posición 1.
  2. Cuando la plataforma alcanza esta posición, se detiene.
  3. Se extiende el pistón para empujar la pieza por la rampa 1.
  4. Una vez que el pistón se ha extendido por completo, se retrae.
  5. Se espera a que llegue una nueva pieza desde la estación cero para repetir el proceso.

Es importante considerar que puede darse la situación de que ya haya una pieza esperando a que la estación 2 esté lista para recibirla. Si llega una nueva pieza desde la estación cero, se procede con normalidad a subir el elevador hasta la posición 2 y medir la longitud de la pieza. Si su longitud es normal, como ya se contaba con una pieza esperando a ir a la estación 2, el pistón no empujará la pieza hacia la rampa 2 hasta que la pieza en espera se haya retirado. Una vez esto ocurra se continúa el ciclo con normalidad. Sin embargo, si en la misma situación llega una pieza defectuosa desde la estación cero, esto no representa un obstáculo para realizar el proceso con normalidad, ya que la pieza en cuestión es desechada.

### 2.3. Estación dos: procesamiento de piezas

La estación dos se encarga de las operaciones de transformación sobre una pieza y está controlada por un autómata S7-1200.

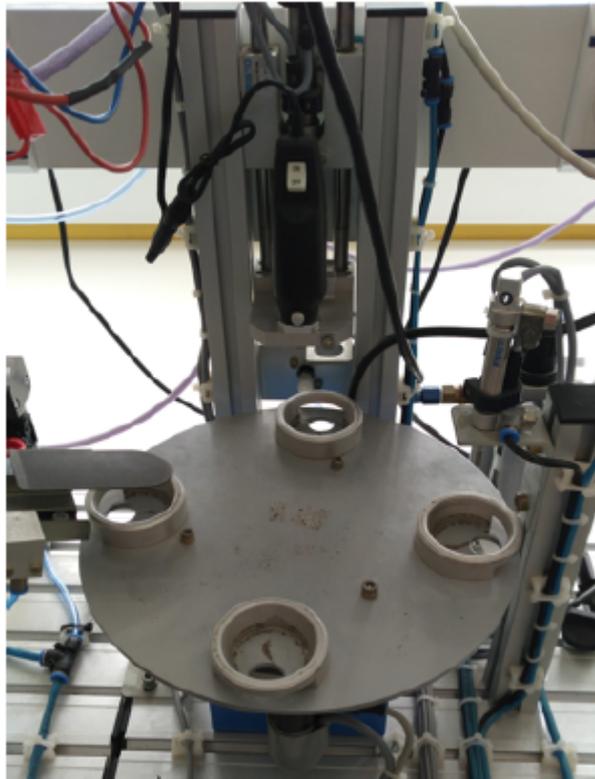


Figura 4. Estación 2 de la Planta Festo

En esta estación contamos con los siguientes componentes:

- ❖ Una mesa giratoria.
- ❖ Un taladro.
- ❖ Un cilindro de sujeción.
- ❖ Un cilindro de comprobación, que se encarga de verificar si la pieza ha sido taladrada correctamente.

Como en todas las estaciones se cuenta con unas condiciones que se deben cumplir:

- ❖ La mesa debe estar en una posición correcta.
- ❖ El cilindro de sujeción debe estar retraído.
- ❖ El motor del taladro apagado (OFF).
- ❖ Y el taladro retraído (en alto).

Para comprender el funcionamiento de la estación 2 se debe saber que:

- ❖ La pieza que llega desde la estación 1 se sitúa sobre la mesa giratoria en la posición 1.
- ❖ Cada giro de la mesa coloca la pieza en una nueva posición.
- ❖ En la posición 2, la pieza se sujeta por el cilindro de sujeción y el taladro realiza un orificio en la parte superior de la misma.
- ❖ En la posición 3, el cilindro de comprobación realiza la verificación del taladrado.
- ❖ Finalmente en la posición 4 la pieza es recogida por la estación 3.

Una vez se cumplan las condiciones iniciales se comienza con la secuencia de acciones que consta de los siguientes pasos:

1. Preparación de la mesa: con la mesa en una posición correcta y cuando todas las posiciones (1,2,3 y 4) están listas para cambiar, la mesa gira 90°.
2. Recepción de la pieza: si la posición está vacía se notifica a la estación anterior (la estación 1) que se está listo para recibir una pieza. Una vez llega (es detectada en la posición 1) y si las posiciones 2,3 y 4 están listas, se gira la mesa 90°.
3. Taladrado: Con la pieza en la posición 2, se sujeta con el cilindro de sujeción, se activa el motor del taladro (ON) y el taladro baja. Cuando esté extendido por completo, el motor del taladro se apaga(OFF) , sube el taladro y el cilindro de sujeción se retrae. Una vez el taladro y el cilindro estén completamente retraídos (arriba) ha terminado el proceso de la posición 2 y si las posiciones 1, 3 y 4 están listas, la mesa podrá girar de nuevo.
4. Comprobación: Cuando la mesa ha girado y se cuenta con una pieza en la posición 3, el cilindro de comprobación baja para verificar si la pieza ha sido taladrada correctamente. Si el cilindro se encuentra completamente extendido, se considera que la pieza está bien taladrada. En caso contrario, se considera que la pieza está mal taladrada. En ambos casos, una vez completada la comprobación, el cilindro sube de nuevo. Con el cilindro retraído, la mesa estará lista para girar de nuevo, siempre y cuando las posiciones 1, 2 y 4 estén listas.
5. Retirada: Cuando se cuenta con una pieza en la posición 4, se le comunica a la estación 3 que hay una pieza lista para ser recogida, especificando si es una pieza bien o mal taladrada, para que la estación 3 se encargue de ella adecuadamente. Se espera a que se retire la pieza y una vez retirada, se le

comunica a las posiciones 1,2 y 3 que se está listo para que la mesa gire de nuevo.

Los pasos de recepción de las piezas, taladrado, comprobación y retirada se deben realizar simultáneamente.

## **2.4. Estación tres: acarreo de piezas**

Esta estación permite transferir una pieza desde la estación 2 (que es la estación anterior) hacia el almacén de piezas defectuosas o a la siguiente estación (la estación 4). La estación 2 debe comunicar que hay una pieza lista para ser recogida, mientras que la estación 4 debe indicar que está lista para recibir la pieza. Esta estación es controlada por medio de un autómata S7-1200.



Figura 5. Estación tres de la Planta Festo.

La estación cuenta con:

- ❖ Un brazo de transferencia con capacidad de rotación horizontal de 180°, lo que le permite moverse hacia la estación 2 o 4. Además, cuenta con un movimiento horizontal y vertical.
- ❖ Un almacén de piezas defectuosas que se encuentra situado al lado del brazo de transferencia (el cual es como una rampa).

- ❖ Un sistema de vacío que se usa para succionar las piezas para su traslado.

Las condiciones iniciales necesarias son:

- ❖ El brazo debe estar girado hacia la estación 2.
- ❖ Los elementos horizontal y vertical deben estar retraídos.
- ❖ El sistema de vacío debe estar apagado (OFF).

Una vez se cumplen las condiciones iniciales se comienza con la secuencia de acciones:

1. Se extiende el elemento horizontal del brazo.
2. Una vez extendido, se extiende el vertical y se activa el vacío (ON).
3. A continuación, se retraen los elementos vertical y horizontal.
4. Se gira el brazo hacia la estación 4.

Puede ocurrir que durante el traslado se pierda la pieza por el camino, en cuyo caso se debe volver a la posición inicial (brazo girado hacia la estación 2) y comunicar a las estaciones 2 y 4 que se está listo para comenzar de nuevo.

Si la pieza no se ha perdido y el brazo se encuentra en la posición posterior (hacia la estación 4), se pueden tener dos opciones:

- La pieza recogida en la estación 2 es defectuosa:
  1. Se extiende el elemento vertical.
  2. Se desactiva el vacío (OFF) para soltar la pieza.
  3. Se retrae el elemento vertical.
- Si la pieza recogida en la estación 2 no es defectuosa:
  1. Se extiende el elemento horizontal.
  2. Se extiende el elemento vertical.
  3. Una vez ambos están extendidos, se desactiva el vacío (OFF) para soltar la pieza.
  4. Una vez terminado, se retraen los elementos vertical y horizontal.

Después de completar cualquiera de estas dos situaciones, el brazo gira de nuevo hacia la estación 2 y comienza de nuevo el ciclo.

## 2.5. Estación cuatro: clasificación de piezas

La estación de clasificación de piezas se controla mediante un autómatas S7-1500 y utiliza actuadores electromecánicos y neumáticos al igual que las demás estaciones. Este autómatas actúa como máster en una red MPI, comunicándose con los otros autómatas responsables de las demás estaciones. Su tarea principal es coordinar la automatización de las cinco estaciones, gestionando la lectura y escritura de datos en la memoria de los otros autómatas. Además, la automatización de esta estación está condicionada por el estado de la estación previa (estación 3).

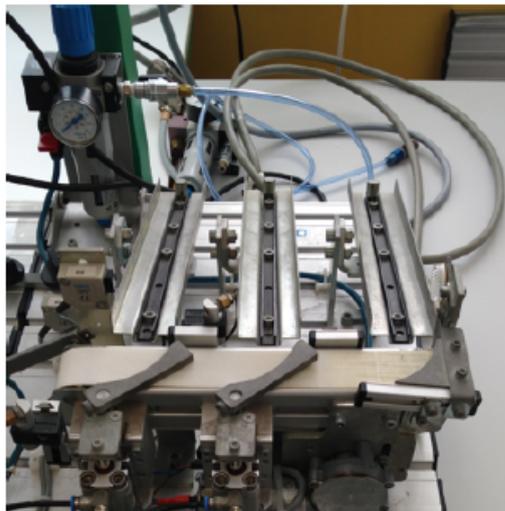


Figura 6. Estación 4 de la Planta Festo

Los componentes de esta estación son:

- ❖ Una cinta transportadora accionada mediante un motor eléctrico unidireccional.
- ❖ Dos desviadores neumáticos que llevan las piezas al almacén.
- ❖ Tres rampas por las cuales se deslizan las piezas hacia el almacén.
- ❖ Dos sensores ópticos
  - Uno para detectar piezas al inicio de la cinta transportadora.
  - Otro para detectar piezas entrando a los almacenes.
- ❖ Una botonera con los siguientes botones, cada uno con su sensor correspondiente:
  - Botón Stop: para detener la cinta transportadora.
  - Botón Start: Para iniciar la cinta transportadora.

- Conmutador Auto/Manual: para seleccionar el modo de operación en el que se quiera trabajar.

En esta estación hay dos modos de funcionamiento, a continuación se ven las secuencias en cada caso:

- ❖ Para el funcionamiento automático:
  - La cinta transportadora comienza a moverse cuando detecta una pieza al inicio y se detiene cuando la pieza haya entrado a alguno de los almacenes o cuando se pulse el botón Stop.
  - Los desviadores tienen dos posiciones (extendido o retraído) y son controlados por dos señales CLAS 1 (para el desviador 1) y CLAS 2 (para el desviador 2). Estas señales se controlan mediante dos interruptores del autómeta. Cuando se activa el interruptor de CLAS 1, la pieza que esté sobre la cinta será empujada hacia el almacén por el desviador 1. Ocurre lo mismo con el interruptor CLAS 2 Y el desviador 2. Si ninguna de estas señales están activas (CLAS 1 y CLAS 2) , la pieza se dirige al último almacén.

Se debe tener en cuenta que si se está en el modo automático y se desea cambiar al manual, pero la cinta se encuentra en movimiento a la hora de escoger el modo manual, esta deberá seguir moviéndose hasta que la pieza que está sobre ella entre en alguno de los almacenes. Solo entonces se producirá el cambio de modo.

- ❖ En el modo manual:
  - Los desviadores cuentan con el mismo funcionamiento descrito anteriormente.
  - Sin embargo, la cinta comienza a moverse cuando se pulsa el botón Start y se detiene cuando una pieza ha entrado en uno de los almacenes o se pulsa el botón Stop.

Para el correcto funcionamiento de esta estación se debe tener en cuenta ciertas normas:

- ❖ Los desviadores no pueden cambiar de extendido a retraído o viceversa, mientras la cinta está en movimiento.

- ❖ La cinta no debe moverse hasta que los desviadores no estén o extendidos o retraídos por completo, es decir, la cinta y los desviadores no pueden moverse al mismo tiempo.

### **3. Herramientas de diseño y programación**

#### **3.1. Guía GEMMA**

La Guía GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts) desarrollada por ADEPA (Association pour le Développement des Équipements Pédagogiques en Automatique) es una metodología muy utilizada en el ámbito de la automatización industrial. Su objetivo es ayudar a gestionar y estandarizar los modos de operación de sistemas de fabricación.

Esta guía es una herramienta gráfica que facilita la representación clara y ordenada de los distintos modos o estados de marcha y paro en un proceso productivo automatizado, además de describir las condiciones y transiciones necesarias para pasar de un estado a otro [4].

Representa los estados mediante rectángulos que contienen una serie de situaciones estándar presentes en cualquier sistema automatizado. Las transiciones entre estados se representan con líneas orientadas, y las condiciones de evolución, en caso de existir, se deben colocar en las transiciones. Estas categorías abarcan los estados de parada, producción y defecto, contando en total con 17 estados de funcionamiento posible, como se muestra en la figura 7, y que se explicarán a continuación.

Es importante destacar que la figura 7 ilustra todos los estados tipificados así como las transiciones más habituales entre ellos (líneas orientadas que conectan unos estados con otros) más habituales. Sin embargo, y debido a la gran variedad de procesos que existen en la industria, la Guía GEMMA tiene que ser lo suficientemente flexible como para poder adaptarse. Así pues, por un lado, no es necesario que la Guía GEMMA incluya siempre todos los estados tipificados. Es perfectamente admisible que alguno o algunos de ellos no se incluyan porque no se consideren de interés o porque no se puedan implementar con el hardware del que se dispone. Por otro lado, también puede ocurrir que para algún proceso en particular se requiera algún estado que no coincida exactamente con ninguno de los tipificados. En este caso, y gracias a la flexibilidad de la herramienta, la recomendación es hacerlo corresponder con alguno de los estados de la figura 7, en lugar de crear un estado nuevo. Asimismo, las transiciones y condiciones de evolución se deben adaptar en cada diseño a las características específicas del proyecto [5].

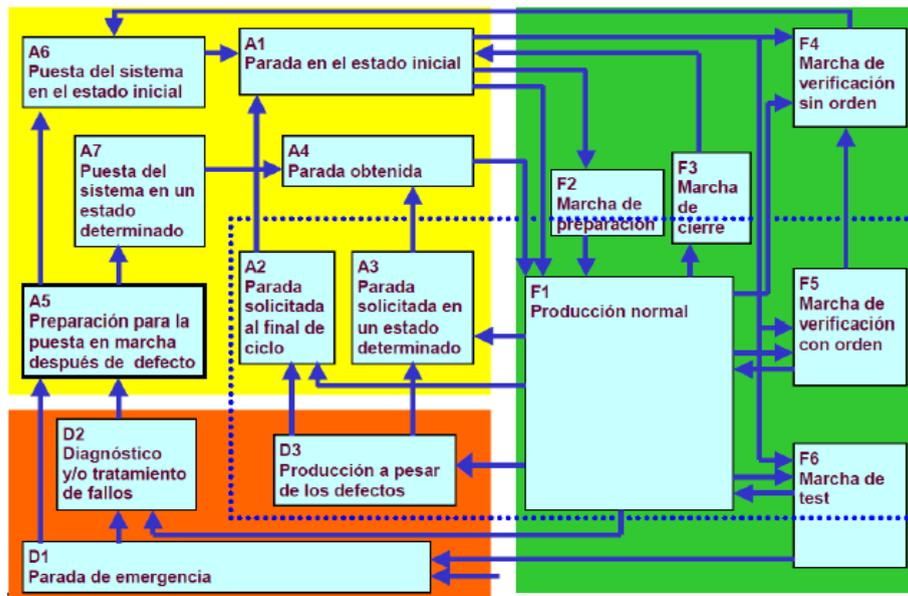


Figura 7. Familias de estados de la Guía GEMMA

- ❖ Estados de seguridad: se encarga de las condiciones de seguridad del sistema, abarcando los estados:
  - D1- Parada de emergencia: Este estado se activa cuando se necesita detener la máquina de inmediato debido a una situación de emergencia.
  - D2- Diagnóstico y/o tratamiento de fallos: Estado en el que la máquina debe ser examinada y reparada después de un defecto o fallo.
  - D3- Producción a pesar de los defectos: Estado en el que pese a los defectos se debe/puede continuar produciendo, por ejemplo, en situaciones en las que se pueda sustituir de manera transitoria el trabajo de la máquina por la de un operario hasta que se repare la avería.
- ❖ Estados de modos de conducción: se gestionan los diferentes estados operativos del sistema, incluyendo:
  - A1- Parada en el estado inicial: Estado de reposo de la máquina antes de que comience cualquier otra operación. Estado en el que la máquina se encuentra lista para comenzar el ciclo de producción.
  - A2- Parada solicitada al final del ciclo: Es un estado transitorio en que la máquina, que hasta el momento estaba produciendo normalmente, debe producir solo hasta acabar el ciclo y pasar a estar parada en el estado inicial [5].

- A3- Parada solicitada en un estado determinado: Estado en el que la máquina se detiene en un estado determinado, que no necesariamente es el final del ciclo.
  - A4- Parada obtenida: Estado de reposo de la máquina distinto al estado inicial. Puede emplearse para una pausa breve o cuando la máquina necesita estar en una condición específica antes de reanudar la producción.
  - A5- Preparación para la puesta en marcha después de defecto: En este estado se procede a todas las operaciones, como puede ser el vaciado o la reposición de piezas, necesarias para la puesta de nuevo en funcionamiento de la máquina después de un defecto.
  - A6- Puesta del sistema en el estado inicial : Se realiza el retorno del sistema al estado inicial, preparándolo para comenzar de nuevo el ciclo de producción.
  - A7- Puesta del sistema en un estado determinado: Se retorna el sistema a una posición distinta de la inicial para su puesta en marcha. Puede ser manual o automático.
- ❖ Estados de producción: en él se controlan las operaciones productivas del sistema, cuyos estados son:
- F1- Producción normal: Estado en el que la máquina o sistema produce normalmente.
  - F2- Marcha de preparación: Se llevan a cabo las acciones necesarias para preparar la máquina antes de entrar en producción.
  - F3- Marcha de cierre: Se corresponde a las acciones que se realizan antes de la parada como, por ejemplo, el vaciado de la planta antes de la parada o antes de un cierre.
  - F4- Marcha de verificación sin orden: En este caso la máquina, normalmente por orden del operario, puede realizar cualquier movimiento o conjunto de movimientos preestablecidos. Es el denominado control manual y se utiliza para funciones de mantenimiento y verificación.
  - F5- Marcha de verificación con orden : La máquina realiza el ciclo completo de funcionamiento en orden pero al ritmo fijado por el operador. En este estado la máquina puede estar en producción, por lo que se asocia al control semiautomático.

- F6- Marcha de test: Empleada para realizar operaciones de ajuste y mantenimiento preventivo, asegurando que la máquina funcione correctamente antes de entrar en producción.

Para aplicar la Guía GEMMA a una automatización se debe seguir los siguientes pasos [4].

- ❖ Identificar los aspectos generales del proceso.
- ❖ Seleccionar los sensores y actuadores.
- ❖ Desarrollar el Grafcet tecnológico del estado de producción normal.
- ❖ Seleccionar qué estados de la guía son necesarios.
- ❖ Definir las condiciones de evolución entre los diferentes estados.
- ❖ Elaborar el grafcet final.
- ❖ Diseñar el pupitre de control, para el cual hay que tener en cuenta las siguientes especificaciones para la utilización de los pulsadores y los pilotos:
  - Pulsador blanco: para la puesta en marcha o puesta en tensión.
  - Pulsador negro: para la parada o la puesta fuera de tensión.
  - Pulsador rojo sobre fondo amarillo: para la parada de emergencia o indicación de una situación de emergencia.
  - Pulsador amarillo para las condiciones anormales o el restablecimiento de un ciclo interrumpido.
  - Pulsador azul: para la reinicialización del sistema.
  - Piloto rojo: para indicar emergencia.
  - Piloto amarillo: indica anomalía.
  - Piloto verde: para indicar que la máquina está preparada para entrar en servicio o que está en funcionamiento normal.

### **3.2. Grafcet**

*“Un grafcet es un modelo gráfico que resume en forma de estados y transiciones conectados mediante líneas de enlace, el comportamiento de un automatismo de una forma clara y exacta que no deja lugar a errores de implementación.” [Guerrero Saiz, J. M. (2019). Programación Estructurada de Autómatas Programables con GRAFCET. Ediciones Paraninfo.]* Por ser la nomenclatura más habitual y para evitar confusiones, en adelante se hablará de ‘etapas’ de Grafcet y ‘estados’ de la Guía GEMMA.

En sí mismo, Grafcet no es un lenguaje de programación sino más bien una herramienta que describe el funcionamiento de un sistema. Sus principios están definidos por la norma UNE-EN 60848:2022, la cual establece los principios y directrices para la representación de sistemas de control mediante Grafcet, detallando cómo deben definirse y ejecutarse los estados y las transiciones en procesos automatizados.

Para la programación en Grafcet de los estados de la Guía GEMMA, se sigue el modelo jerárquico de Grafcet, que consiste en:

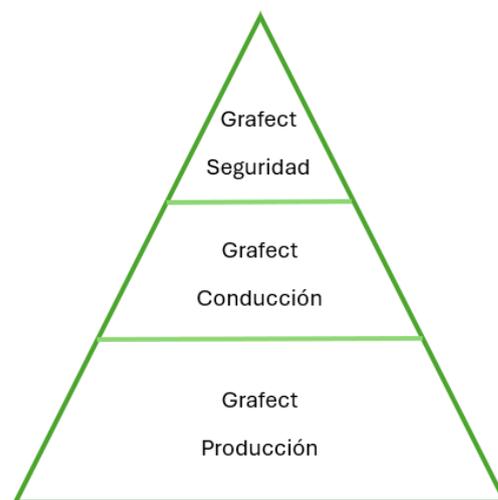


Figura 8. Estructura de la jerarquía del Grafcet.

En la parte superior de la pirámide se encuentra el Grafcet de seguridad. Después se cuenta con el Grafcet de conducción, en el cual se recogen todos los modos de conducción, como pueden ser los estados A6, A1, A2, A3 y A4 de la Guía GEMMA. Por último, el Grafcet de producción en el cual se contemplan los modos de funcionamiento.

Los elementos principales que componen Grafcet son [6]:

- ❖ Etapas: Representa la situación en las que se encuentra el sistema o el proceso en un momento dado. Son representadas por un rectángulo simple, excepto las etapas iniciales que se representan con un doble rectángulo. Una etapa puede estar activa o inactiva, el

conjunto de las etapas activas de un grafcet en un instante determinado, representa la situación actual del grafcet.

Como norma general, nunca debe haber más de una etapa activa simultáneamente, a excepción de de las divergencias en Y que se verán más adelante.



Figura 9. Símbolo etapa y etapa inicial.

- ❖ Transiciones o condiciones de transición: Representan las condiciones que permiten la evolución de una etapa a la siguiente, que pueden ser sensores, temporizadores, contadores, marcas, variables de datos, entre otros. Las transiciones pueden tener flancos de subida y bajada y también pueden incluir operaciones lógicas AND, OR y NOT, y comparaciones entre variables, así como temporizadores de retardo a la conexión. Al disparar una transición, simultáneamente se activan todas las etapas posteriores y se desactivan todas las inmediatas anteriores. Nunca debe haber dos condiciones de transición activas consecutivas, según las reglas establecidas en la norma UNE-EN 60848:2022 [6].

Se representan mediante una línea perpendicular que une dos etapas.



Figura 10. Ejemplo diagrama de Grafcet con transición de flanco de subida.

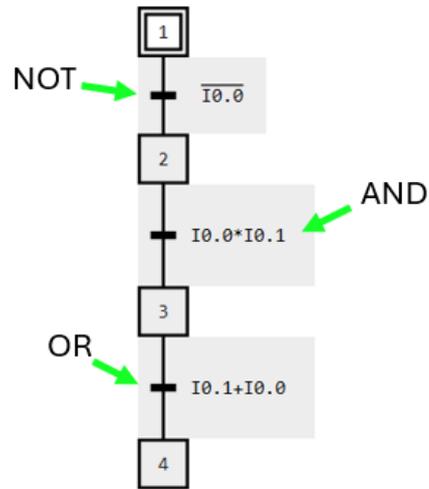


Figura 11. Ejemplo diagrama Grafcet con transición NOT, AND y OR..

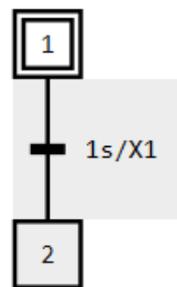


Figura 12. Ejemplo diagrama Grafcet con temporizador de retardo a la conexión (TON).

Los operadores más habituales, que además son los que se han utilizado en este Trabajo de Fin de Grado, son los que se muestran en la tabla 1.

Tabla 1. Algunos operadores de Grafcet.

OPERADOR	SÍMBOLO
MAYOR QUE	>
MENOR QUE	<
IGUAL	=

MAYOR O IGUAL QUE	$\geq$
MENOR O IGUAL QUE	$\leq$
DISTINTO DE	$\neq$

❖ Acciones:

Especifica que se debe ejecutar cuando se activa su etapa correspondiente. La norma UNE-EN 60848:2022 distingue entre acciones continuas y acciones con memorización [6]. Ambas se han empleado en este trabajo de fin de grado. Estas acciones pueden ser de activación, ejecutadas al activar la etapa, o de desactivación, ejecutadas al desactivar la etapa.

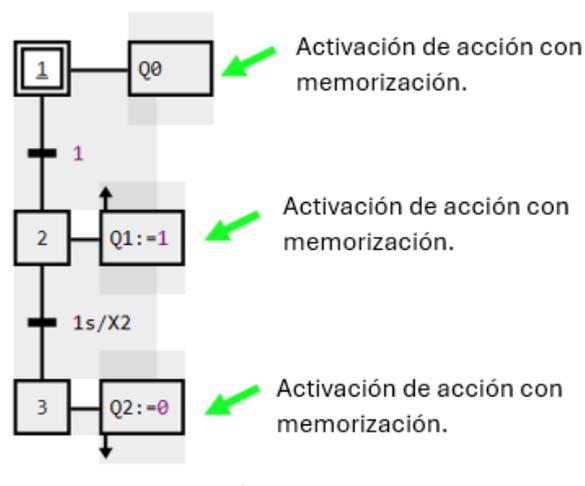


Figura 13. Ejemplo de diferentes tipos de acciones..

Asimismo, para la programación se dispone de los siguientes elementos:

- ❖ Repetición de secuencias: Permite repetir una serie de estados hasta que se satisfaga una condición que haga abandonar este bucle, tal y como se muestra en la figura 14.

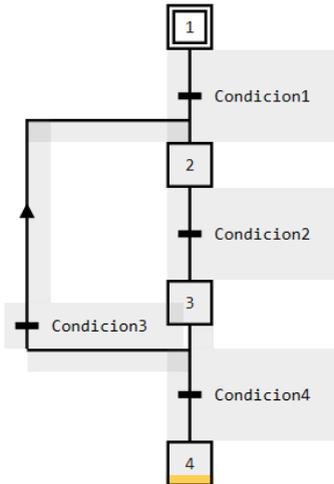


Figura 14. Ejemplo de repetición de secuencias en Grafcet. Se repetirán los estados 2 y 3 hasta que se cumpla la condición 4.

- ❖ **Divergencia en Y:** Estructura en la que una condición lleva a múltiples caminos (estados) paralelos que se ejecutan de forma simultánea. Se utiliza para iniciar varias secuencias al mismo tiempo desde un único estado inicial. Esto se ilustra en la figura 15, donde a partir de que se cumpla la condición 1 se activan las etapas 2 y 3, lo que implica que se ejecutan las acciones 1 y 2 de manera simultánea. Para salir de esta divergencia la condición de transición de salida (condición 4) es común para todas las secuencias de la divergencia. Así pues, para poder salir de una divergencia en Y deben estar activas la última etapa de todas las secuencias y activarse la condición de transición de salida.

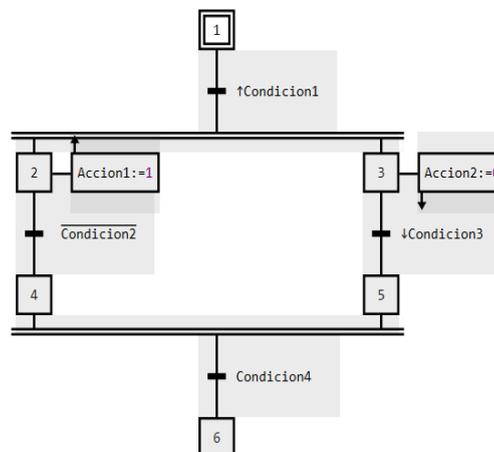


Figura 15. Ejemplo de Divergencia en Y.

❖ Divergencia en O:

En esta divergencia, desde una misma etapa, varias transiciones pueden ser activadas según las condiciones que cumplan. Esto significa que el sistema puede tomar diferentes caminos, cada uno asociado a una condición de transición. Esto se ve a continuación en la figura 16, donde en el momento de encontrarse en la etapa 1, si se cumple la condición 2, se evoluciona a la etapa 3, pero si se cumplía la condición se habría evolucionado a la etapa 2.

Es importante destacar que en las divergencias en O, hay que asegurarse de que nunca se pueda activar más de una condición de transición al mismo tiempo.

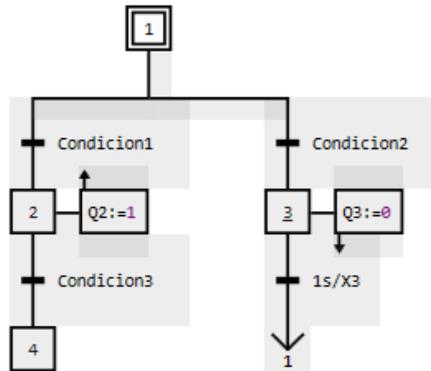


Figura 16. Divergencia en O.

❖ Macroetapas:

Se utilizan para simplificar los diagramas y mejorar su legibilidad. Las macroetapas agrupan varias etapas y transiciones en un Grafcet-Parcial [6]. Puede tener cualquier transición de salida pero normalmente será una transición que sea siempre válida ( igual 1 ). En la figura 17 se ve que al activarse la macroetapa 2 se activa la etapa E2.

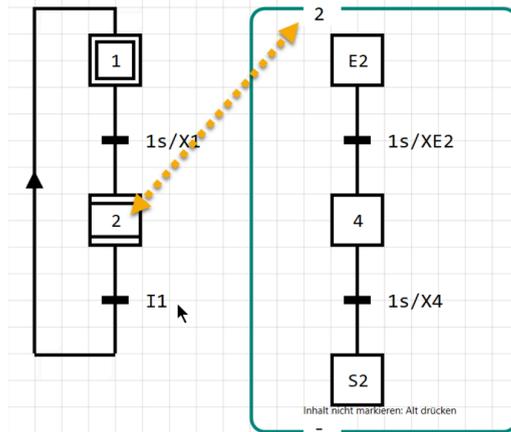


Figura 17. Ejemplo macroetapa [6]

❖ Saltos condicionales

Permiten que el sistema pueda evolucionar de una etapa a otra no contigua. Pueden ser saltos hacia adelante como el que se muestra en la figura 18, o hacia atrás como se muestra en la figura 19

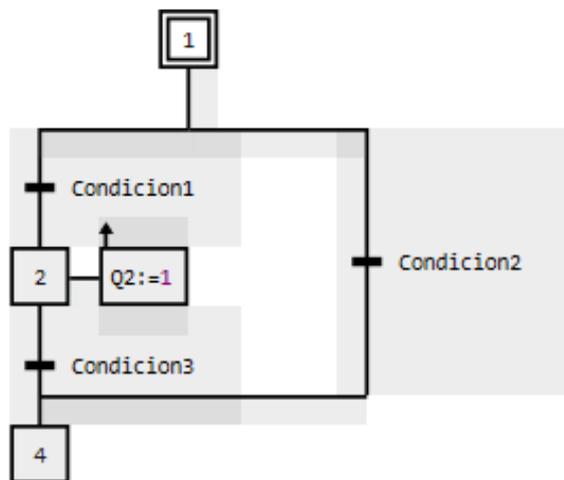


Figura 18. Ejemplo de salto condicional hacia adelante. Si desde la etapa 1 se activa la condición 2 antes que la condición 1, la etapa 2 no se activará.

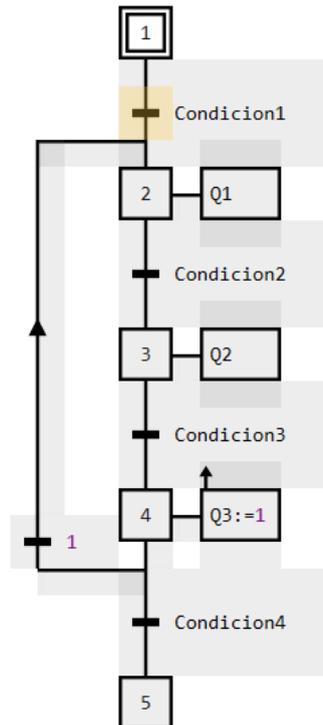


Figura 19. Ejemplo de salto condicional hacia atrás. La secuencia de acciones 2→3→4→2 se ejecuta repetidamente hasta que se active la condición 4.

- ❖ **Grafcet Parcial:** se corresponden con una parte o sección de un diagrama de Grafcet completo. Permiten enfocarse en partes específicas del proceso y dividir procesos complejos en partes manejables, facilitando la comprensión del sistema.

*“Se etiquetan con la letra G seguida de un nombre o de un número [5]”*

Eso se ve en la Figura 20, donde el grafcet parcial G0 muestra las acciones y transiciones que se deben ejecutar durante la etapa 1 del proceso. Una vez que la condición 3 se cumple, el proceso deja de ejecutar el grafcet parcial G0 y avanza directamente a la etapa 5.

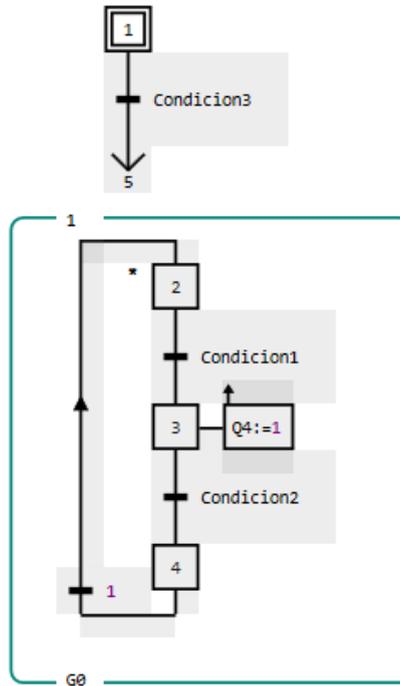


Figura 20. Ejemplo Grafcet Parcial.

❖ Forzados:

Cuando un sistema está constituido por varios GRAFCETs parciales, es posible que un GRAFCET fuerce el estado de otro. El forzado de GRAFCETs es útil para el tratamiento de defectos de funcionamiento y emergencias. El forzado implica una jerarquía entre GRAFCETs parciales. La jerarquía la fija el diseñador del sistema cuando hace que un GRAFCET fuerce o no a otro [7].

Representación del forzado: Se hace con la letra F seguida de una barra, y a continuación se indica el nombre del GRAFCET que se desea forzar.

Los comando de forzado están vinculados a una etapa y hay cuatro tipos:

- ❖ Forzado sostenido a la situación actual [F/GX {Y}]: Este tipo de comando tiene como objetivo mantener o congelar el sistema en su situación actual.

- ❖ Forzado a una situación específica [F/GX {\*}]: Este comando se utiliza para activar una etapa específica mientras se desactivan los demás pasos.
- ❖ Forzado a la situación vacía [F/GX {}]: En este caso, el comando desactiva todos los pasos del sistema
- ❖ Forzado a la situación inicial [F/GX {INiT}]: Este tipo de comando devuelve el sistema a su etapa inicial.

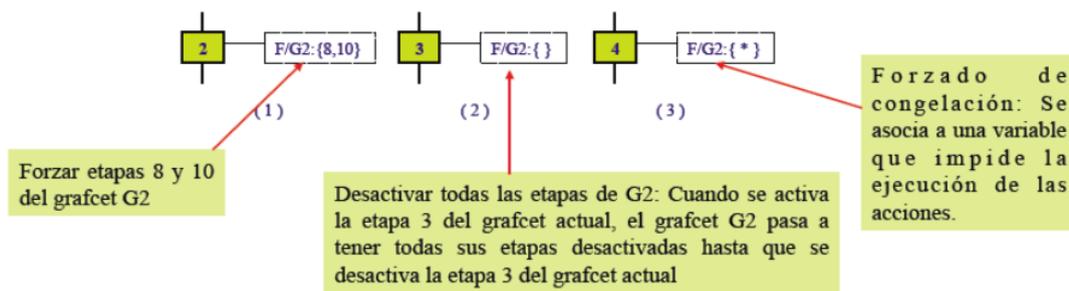


Figura 21. Ejemplo tipo de forzados en Grafcet [8].

## **4. Herramientas software empleadas para el modelado de la planta Festo y su programación en GRAFCET**

En este apartado se describirán las distintas herramientas de software utilizadas para crear los modelos simulados de las distintas estaciones de la planta Festo. Se abordará el uso de Factory IO y Grafcet-Studio. Además posteriormente se detallarán las otras opciones de software para la simulación descubiertas posteriormente como PLC-Lab y PLC-Lab 3D Player.

### **4.1. Modelado de plantas**

El modelado de plantas industriales permite simular y analizar procesos complejos en un entorno virtual, lo cual reduce los riesgos y los costos que se pueden tener en una planta real. Además, facilita la identificación de posibles mejoras.

En el mercado actual, hay varias herramientas de software disponibles para el modelado de plantas industriales, como PC-SIMU, Factory IO, SolidWorks, Game4Automation y Autodesk Inventor, PLC-Lab y PLC-Lab 3D Player. Estas herramientas son capaces de crear representaciones detalladas y precisas de los procesos industriales, lo cual es fundamental para mejorar el diseño [1].

Como se mencionó anteriormente, en el Trabajo de fin de Grado del alumno Cristian José Luis González [1], el software elegido para modelar las estaciones de la planta Festo fue Factory IO. Por lo tanto, en este proyecto se ha trabajado directamente con estos modelos, utilizando esta base para continuar con el desarrollo y las mejoras necesarias.

#### **4.1.1. Factory I/O.**

Es un software de simulación 3D para PLC diseñado y creado por la empresa Real Games, para facilitar el aprendizaje de tecnologías de automatización.

Para su utilización, el sistema debe cumplir los siguientes requisitos [9]:

- ❖ Sistema operativo: Windows 7 SP1+ o superior.
- ❖ CPU: Procesador con soporte para conjunto de instrucciones SSE2.
- ❖ API de gráficos: Compatible con DX10, DX11, DX12.
- ❖ GPU:

- NVIDIA desde 2006 (GeForce 8).
- AMD desde 2006 (Radeon HD 2000).
- Intel desde 2012 (HD 4000 / IvyBridge).

La versión que se ha empleado para la realización de este proyecto ha sido la v2.2.3.

Según el fabricante, *“Factory I/O es una aplicación de simulación de fábrica en 3D diseñada. El escenario más común es utilizar Factory I/O como plataforma de capacitación de PLC, ya que los PLC son los controladores más comunes que se encuentran en aplicaciones industriales. Sin embargo, también se puede utilizar con microcontroladores, SoftPLC, Modbus, entre muchas otras tecnologías”* [9].

Su interfaz intuitiva permite construir rápidamente una fábrica virtual utilizando una amplia gama de piezas industriales como piezas, cintas transportadoras, posicionadores, sensores de diferentes tipos, pulsadores, setas de emergencia, almacenes, elevadores, rampas, brazos, creadores de piezas, removedores de piezas, entre muchos otros. Además, ofrece una variedad de escenarios inspirados en aplicaciones industriales típicas, que van desde niveles de dificultad principiante hasta avanzados.



Figura 22. Algunos de los elementos disponibles en Factory IO [9].

Sin embargo presenta algunas limitaciones en la creación de las plantas industriales. Algunas de las que se han encontrado al realizar este proyecto han sido:

- ❖ **Selectores limitados:** Los selectores solo permiten dos posiciones, sin la opción de ampliarlos a más.
- ❖ **El movimiento de los brazos:** Los brazos disponibles en Factory IO no permiten la rotación sobre el eje Z, limitándose los movimientos en los ejes vertical y horizontal.
- ❖ **Modificación de objetos o su colocación:** No es posible modificar el tamaño de los objetos utilizados ni colocarlos libremente en cualquier posición. Las posiciones vienen preestablecidas en una cuadrícula, como se ilustra en la Figura 24. Todo ello limita la capacidad de personalizar libremente el entorno de simulación.



Figura 23. Selector disponible en Factory IO.

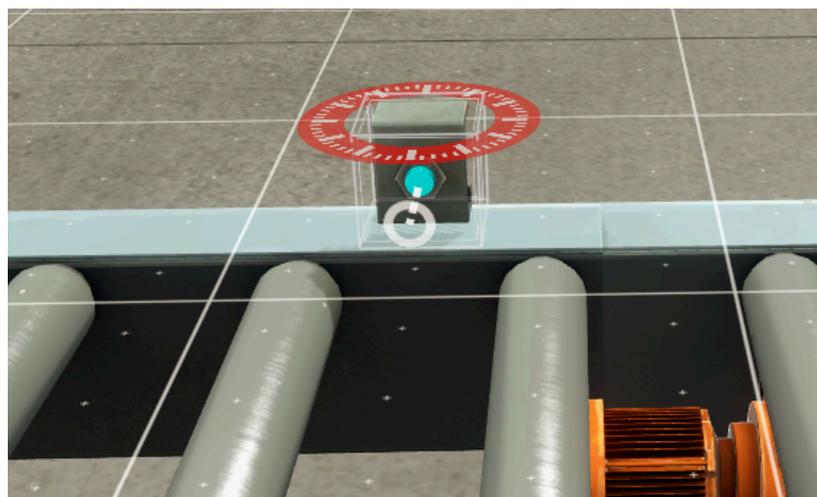


Figura 24. Colocación de objetos en Factory IO.

Por último, Factory IO cuenta con una característica incorporada en la versión Ultimate. Se trata de un controlador de E/S responsable de comunicarse con un

controlador externo. Factory I/O incluye muchos controladores de E/S, cada uno de los cuales está diseñado para usarse con una tecnología específica. Se puede seleccionar el controlador según el dispositivo que se desee utilizar [10].

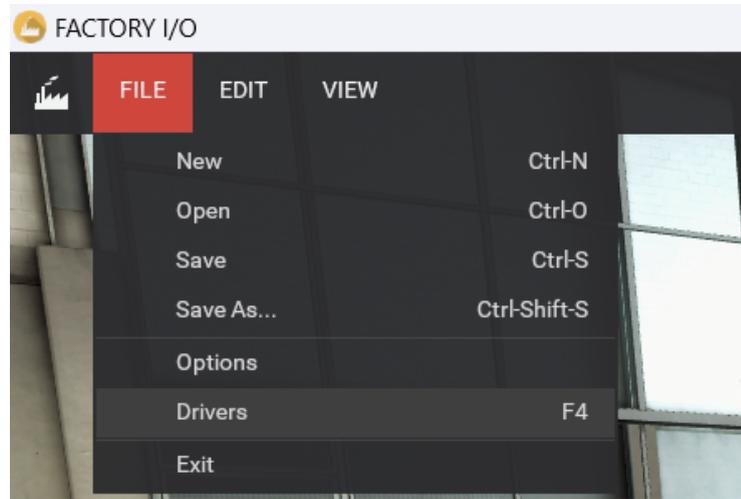


Figura 25. Ruta para acceder a los controladores de E/S disponibles .

Una vez dentro, se encuentra la siguiente lista de controladores:

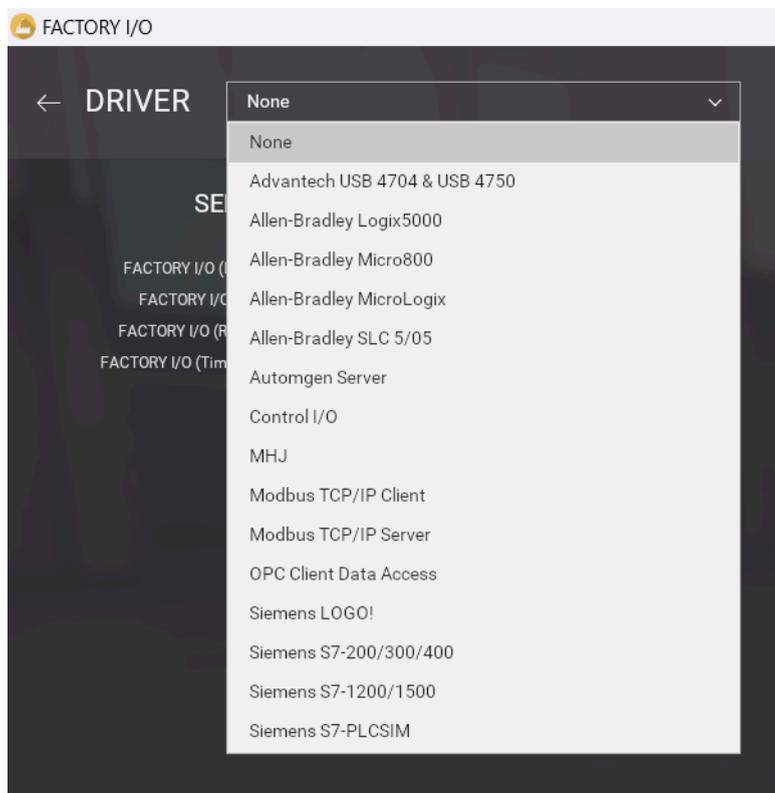


Figura 26. Listado de los controladores disponibles en la versión Ultimate de Factory IO.

Una vez que el controlador elegido esté configurado correctamente, se puede asignar etiquetas de sensores y actuadores a los puntos de E/S del controlador, [10].

Como ya se comentó, para la realización de este trabajo se utilizaron los modelos de las estaciones de la planta Festo ya existentes en Factory IO, creados por el alumno Cristian José Luis González en su Trabajo de Fin de Grado titulado “Diseño y programación de plantas virtuales de laboratorio para asignaturas de Automatización [1]. A la hora de trabajar con estos modelos para el diseño y la implementación de la Guía GEMMA en las estaciones cero, uno y dos de la planta Festo, ha sido necesario realizar ciertas modificaciones, que en la mayoría de los casos han consistido en añadir algunos sensores y modificar el pupitre de control. A continuación, se verá la comparativa entre el modelo ya existente para cada estación y el modelo modificado, así como el detalle de los cambios introducidos.

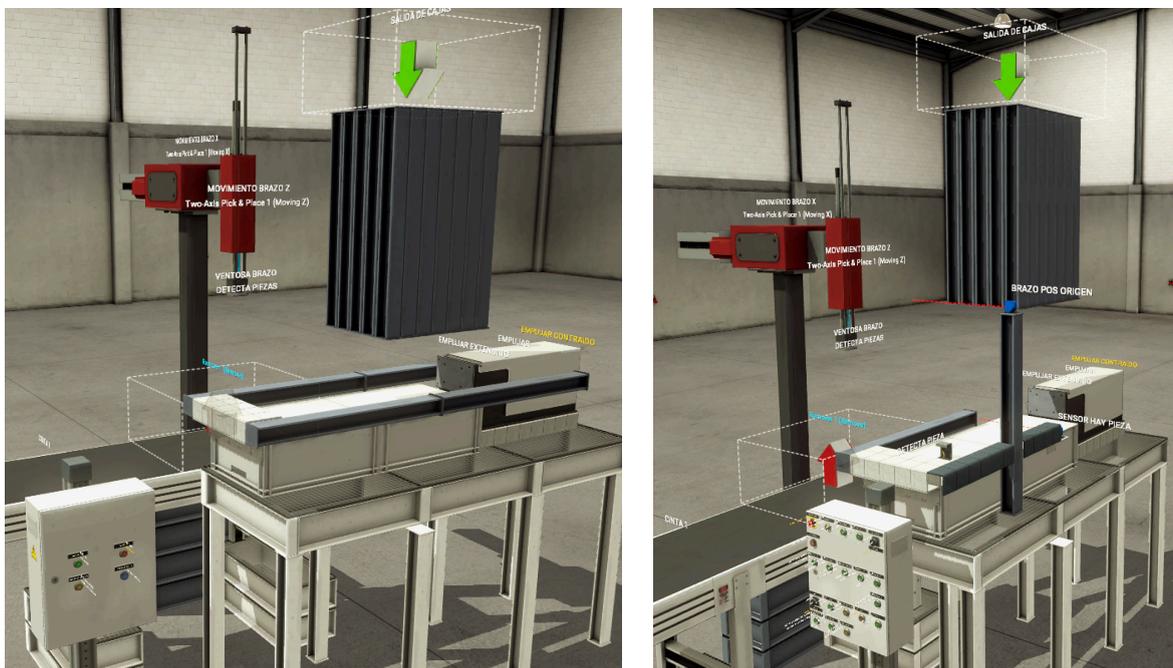


Figura 27. A la izquierda, el modelo existente creado en Factory IO para la estación 0. A la derecha, el modelo con las modificaciones introducidas.

Los elementos añadidos han sido:

- ❖ Un sensor “Brazo Pos Origen” que detecta cuando el brazo está extendido en el eje X, que corresponde a su posición de origen.
- ❖ Un sensor “Detecta Pieza” que indica la presencia de una pieza al final de la plataforma.
- ❖ Un sensor “Hay Pieza” que indica la presencia de una pieza al inicio de la plataforma.
- ❖ En el panel de control, todos los botones necesarios para que el operario pueda forzar la entrada en los distintos estados de la Guía GEMMA que se han incluido en el diseño de esta estación, tal y como se explicará en el apartado 5.1.

Modelo existente para la estación 1.



Figura 28. Modelo existente creado en Factory IO para la estación 1.

Modelo con las modificaciones introducidas.



Figura 29. Modelo modificado para la estación 1.

Los cambios realizados en esta estación han sido los siguientes:

- ❖ En el panel de control, todos los botones necesarios para que el operario pueda forzar la entrada en los distintos estados de la Guía GEMMA que se han incluido en el diseño de esta estación, tal y como se explicará en el apartado 5.1.
- ❖ Un sensor en la rampa 2 que detecta si se cuenta con la presencia de una pieza sobre ella esperando a ser solicitada por la estación 2.

Modelo existente para la estación 2.

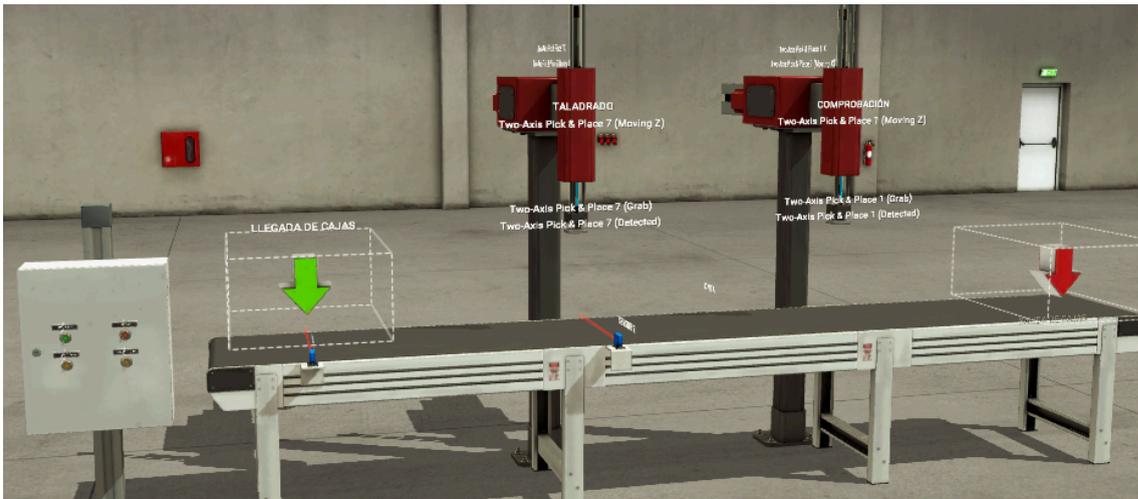


Figura 30. Modelo existente creado en Factory IO para la estación 2.

Modelo empleado.

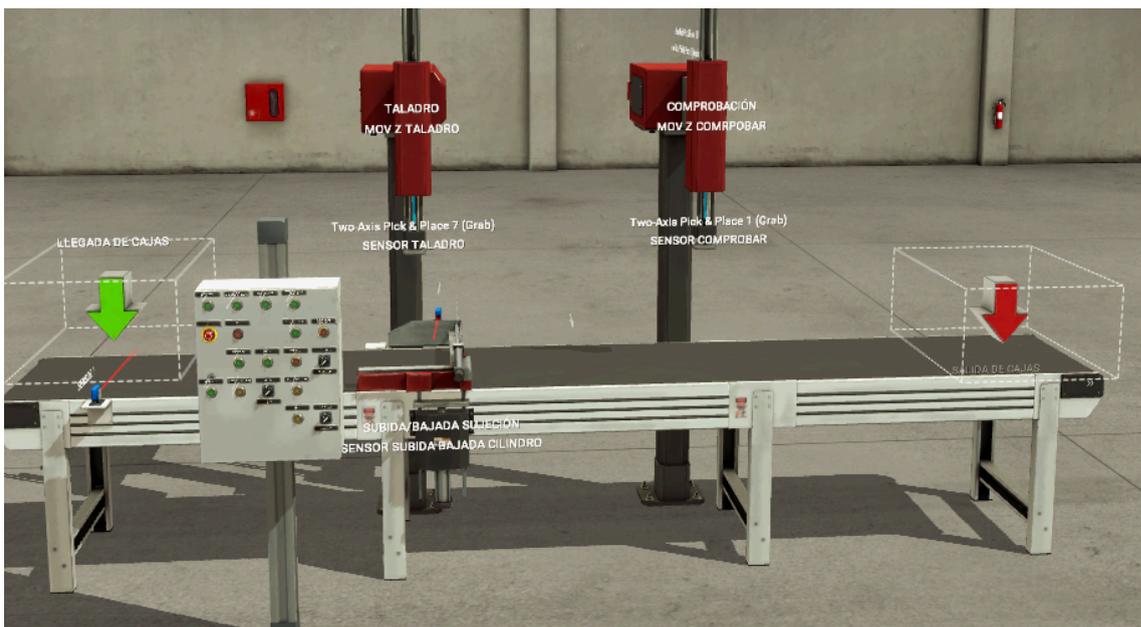


Figura 31. Modelo modificado para la estación 2.

Los cambios se han realizado en el modelo de esta estación han sido los siguientes:

- ❖ Se ha añadido una “abrazadera” que simula ser el cilindro de sujeción con el que se cuenta en la estación 2 de la planta Festo real.

- ❖ Se ha añadido en el panel de control, todos los botones necesarios para que el operario pueda forzar la entrada en los distintos estados de la Guía GEMMA que se han incluido en el diseño de esta estación, tal y como se explicará en el apartado 5.1.
- ❖ Se ha intercambiado de lado el sensor que se encuentra a la altura del primer brazo.

A pesar de que no se han empleado en este trabajo, en las figuras 32 y 33 se muestran los modelos de Factory IO creados por el alumno Cristian José Luis González correspondientes a las estaciones 3 y 4 [1].

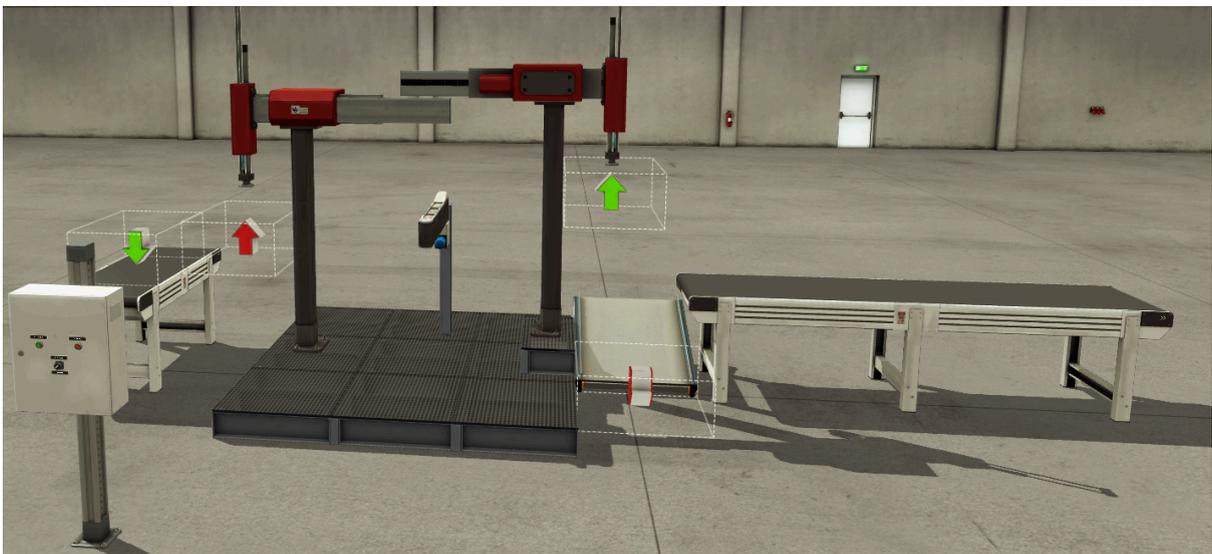


Figura 32. Modelo existente creado en Factory IO para la estación 3.

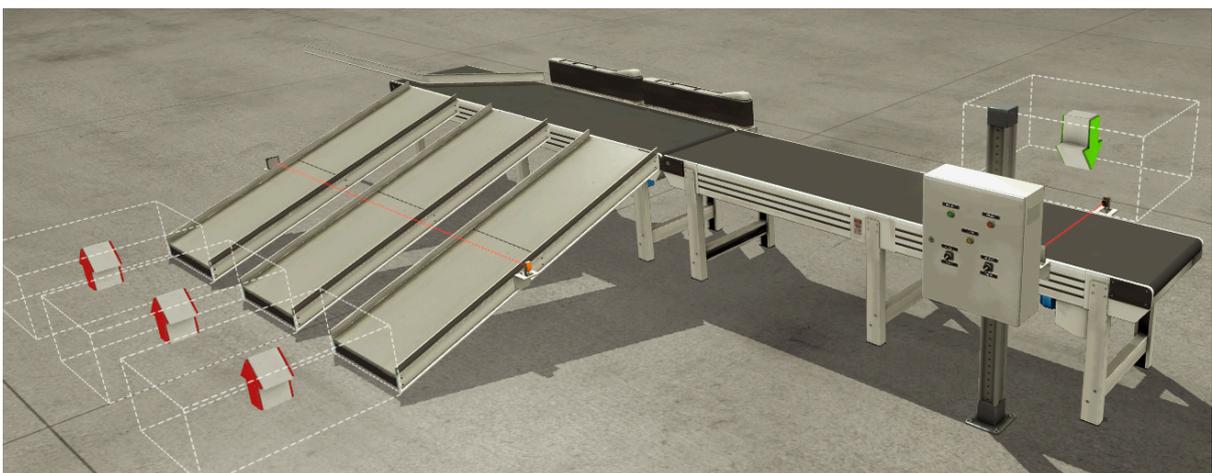


Figura 33. Modelo existente creado en Factory IO para la estación 4.

Si se comparan las estaciones de la planta real (ver figuras 1-6) con los modelos creados en Factory IO (ver figuras 27-33), se ve claramente que existen diferencias apreciables en el aspecto de unos y otros. Esto resulta especialmente significativo en estaciones como la dos, donde la mesa circular giratoria de la estación real se ha tenido que modelar con una cinta transportadora. Además, del cilindro de sujeción y de los brazos del taladro y de comprobación, así como en la estación 3 donde el brazo intercambiador está sustituido por dos brazos, etc. Estas diferencias se deben a que la versión que se dispone del Factory IO no proporciona una gran variedad de elementos a la hora de construir los modelos. Es por esta razón por lo que se decidió buscar alguna otra herramienta de simulación de plantas industriales con la que se pudieran construir unos modelos más parecidos a las estaciones reales. Hay que destacar que de cara al uso docente de estos modelos, es importante que los estudiantes los identifiquen claramente con las estaciones con las que trabajan en el laboratorio. A continuación, se describen dos herramientas alternativas al Factory IO que podrían resultar de interés.

#### **4.1.2. PLC-Lab**

Es un software creado por MHJ-Software. Con PLC-Lab se puede crear una réplica digital simplificada en 2D que imita un sistema real controlado por PLC. Esto permite identificar y corregir errores en los programas de PLC, así como practicar y aprender sobre el funcionamiento de los PLCs en un entorno virtual antes de aplicarlo a situaciones reales. Configurar un sistema o entorno de prueba es sencillo y el uso de PLC-Lab es similar a programas como Microsoft PowerPoint [11].

Los requisitos con los que debe contar el sistema para su uso son :

- ❖ Sistema operativo: Windows7 SP1 + o superior.
- ❖ Para su utilización con TIA PORTAL: Mínimo 16 GB de RAM, CPU Core™ i5 de 3,4 GHz o superior.
- ❖ Cuando se usa PLC-Lab con WinSPS-S7 o Grafcet-Studio: Mínimo 8GB de RAM o mejor. CPU: Core™ i3 3,2 GHz o mejor.
- ❖ .NET Framework: 4.5.2 y 4.6.2.
- ❖ Idiomas aportados: Interfaz de usuario: inglés y alemán.  
Documentación: inglés, alemán.

PLC-Lab es compatible con muchos sistemas de destino diferentes como::

**Sistemas de software:**

- ❖ PLCSIM-S7300 (Administrador Simatic).
- ❖ PLCSIM-S7300 (PORTAL TIA).
- ❖ PLCSIM-1200 (PORTAL TIA).
- ❖ PLCSIM-1500 (PORTAL TIA).
- ❖ PLCSIM-Avanzado.
- ❖ Memoria interna (IM, Banderas).
- ❖ WinSPS-S7.
- ❖ Estudio Grafcet.

**Sistemas hardware:**

- ❖ S7-300 (Siemens AG)
- ❖ S7-400 (Siemens AG)
- ❖ S7-1200 (Siemens AG)
- ❖ S7-1500 (Siemens AG)
- ❖ ¡LOGO! (Siemens AG)

Además, también se pueden utilizar sistemas que admitan **OPC/UA** , por ejemplo , PLC de software y hardware basados en **CODESYS V3**.

El espacio de trabajo en PLC-Lab, que es donde se encuentran los elementos de dibujo para la creación de escenas simuladas, tiene el aspecto que se muestra en la figura 34.

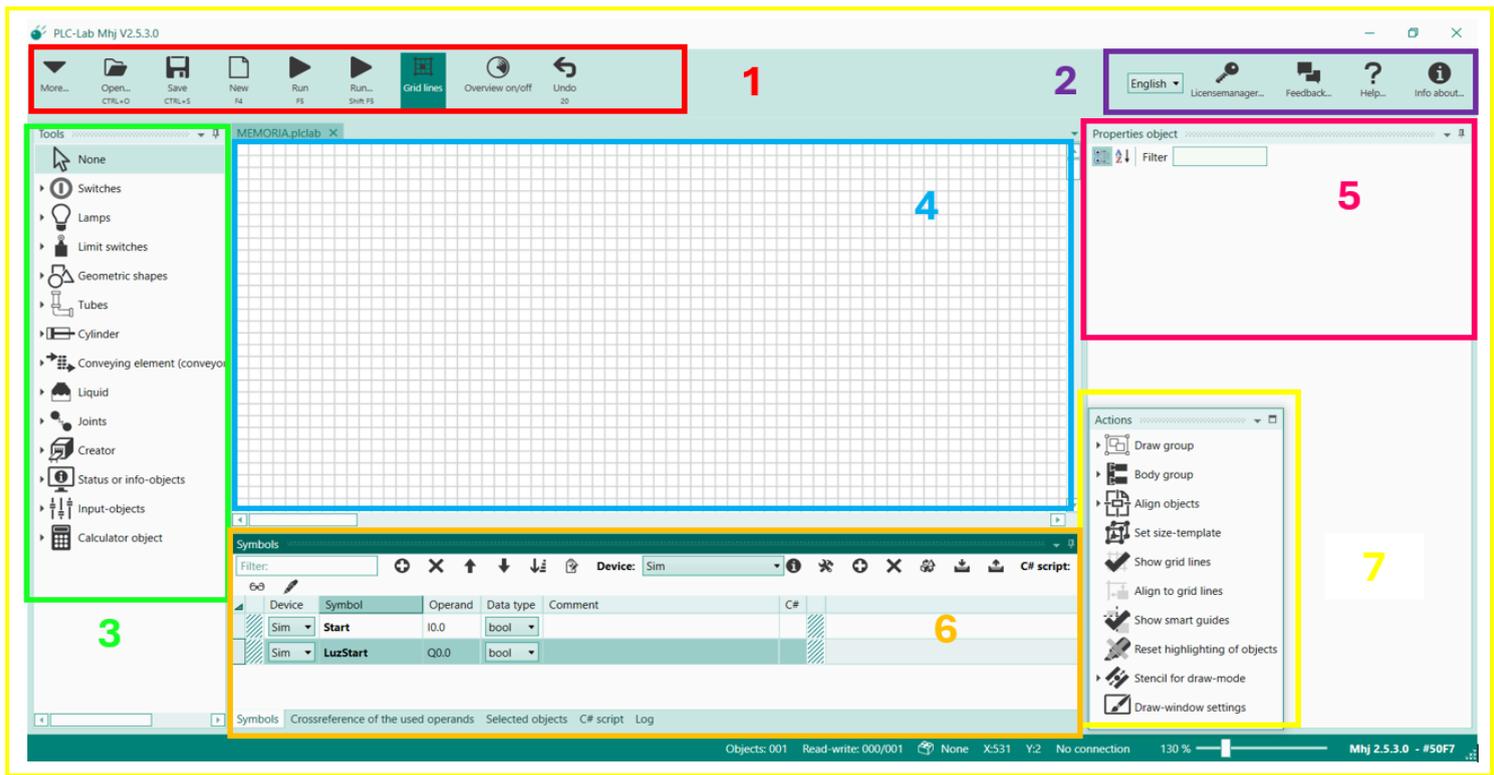


Figura 34. Diseño de pantalla en PLC-Lab.

PLC-Lab dispone de una amplia variedad de objetos, tal y como se observa en la figura 36, que, a su vez, ofrecen grandes posibilidades de diseño y configuración, lo que permite implementar una gran variedad de proyectos.

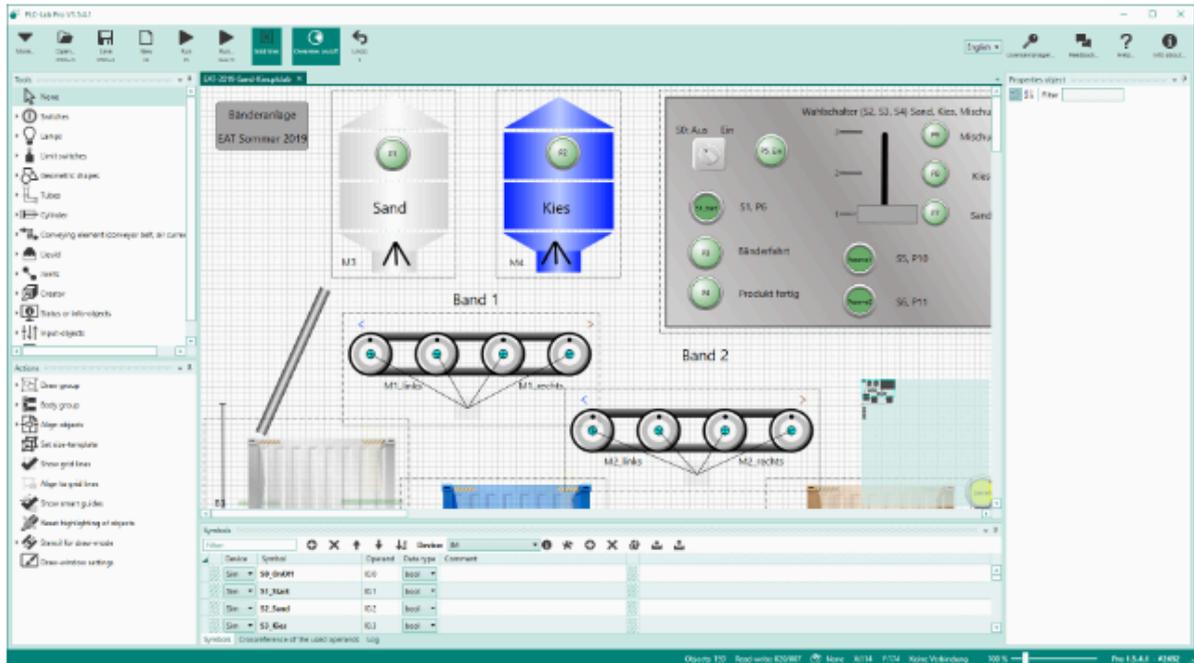


Figura 35. Ejemplo de escena creada en PLC-Lab [12].

En la figura 35 se observa un ejemplo de una escena creada con PLC-Lab en la que se hace uso de algunos de los elementos mostrados en la figura 36.

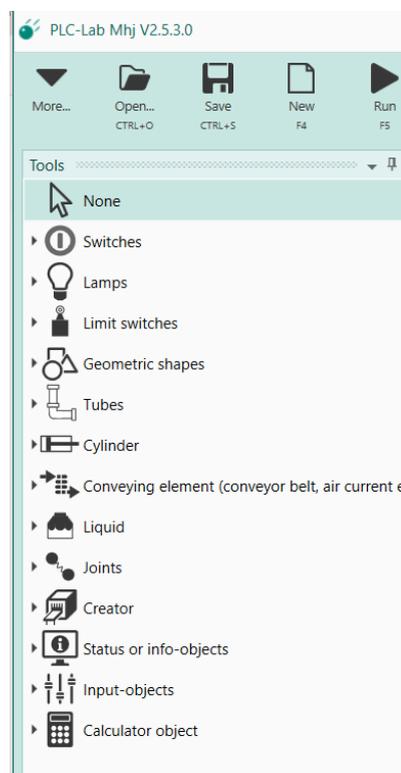


Figura 36. Elementos de dibujo disponibles para la creación de escenas en PLC-Lab.

A estos elementos se les pueden asignar propiedades físicas que influyen en su comportamiento durante la simulación. Por ejemplo, los objetos pueden estar sujetos a la gravedad y ser magnéticos. También se pueden configurar parámetros como la densidad, la fricción superficial y la elasticidad. La interfaz de usuario y el área de dibujo están completamente orientados a vectores, lo que implica que los objetos pueden ampliarse o reducirse sin perder calidad.

Se puede entender mejor todo lo mencionado anteriormente con la ayuda de un ejemplo concreto. Cuando se crea un interruptor no iluminado y se selecciona, el menú de ajustes principales aparece a su derecha, los cuales se ven en detalle a continuación.

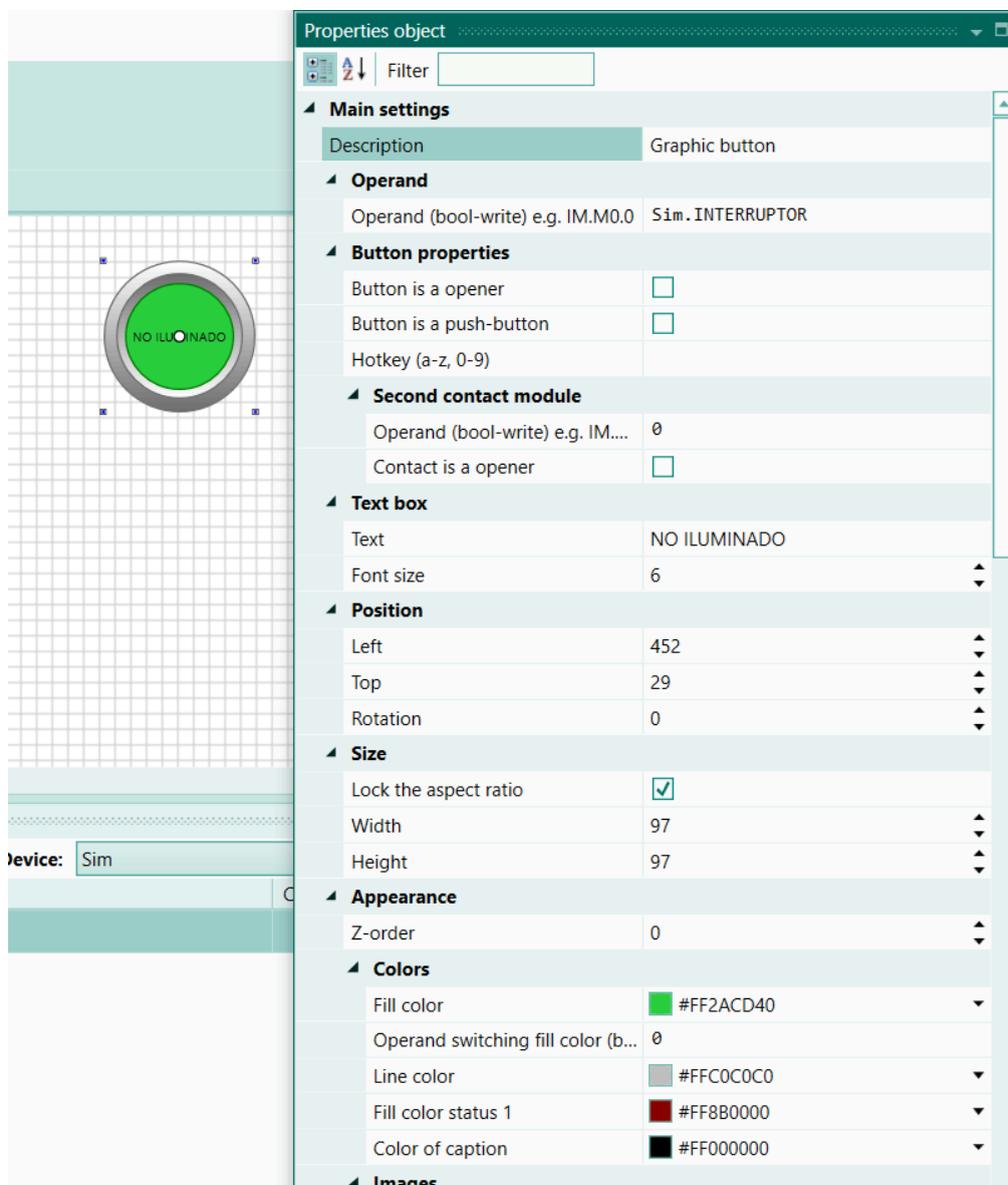


Figura 37. Ajustes principales de los elementos de PLC-Lab (1ª Parte).

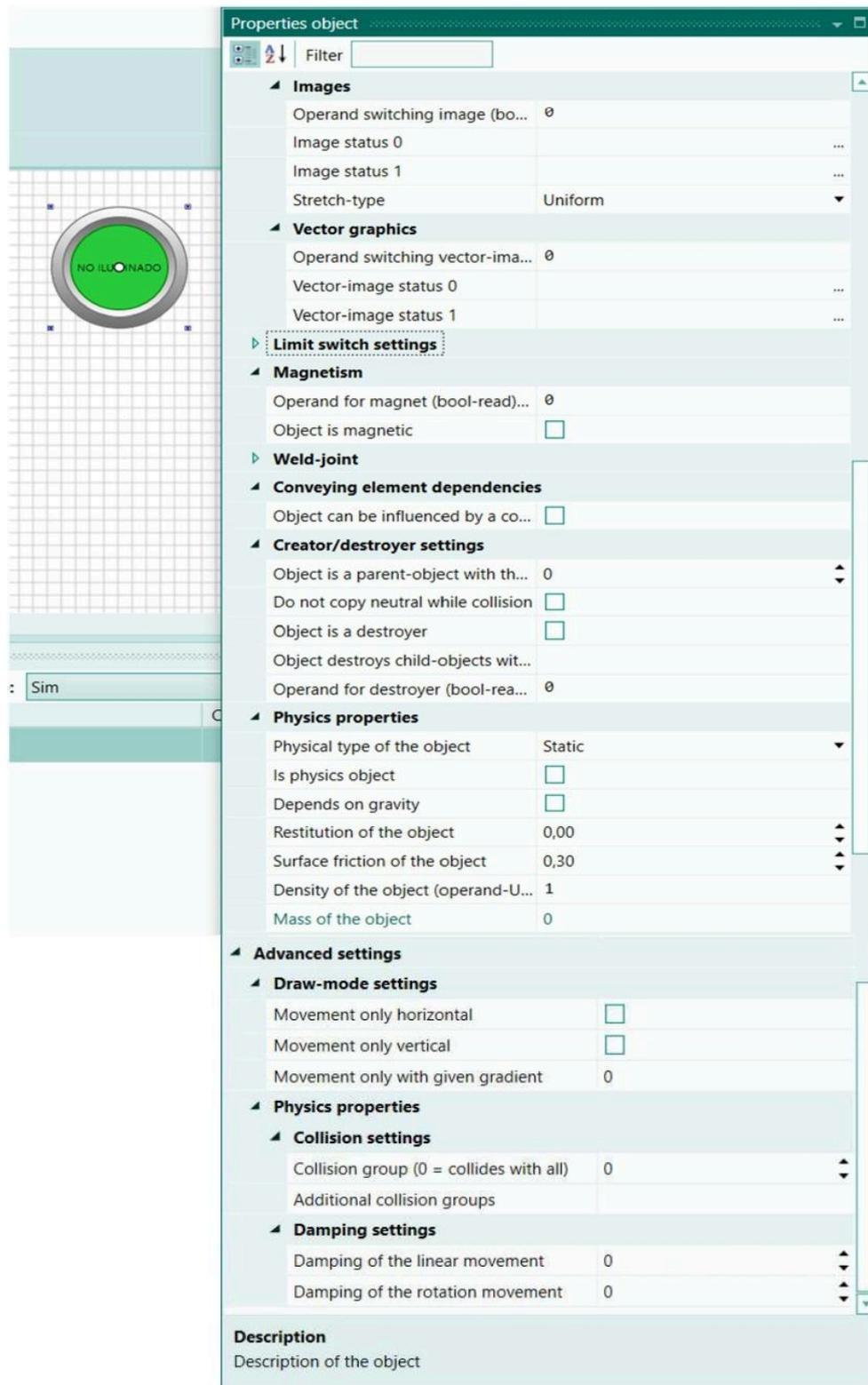


Figura 38. Ajustes principales de los elementos de PLC-Lab (2ª Parte).

Debido al gran número de parámetros que se pueden configurar en los objetos de PLC-Lab (ver figuras 37 y 38), en este trabajo únicamente se describirán los que se han empleado para crear los modelos de la

En la Figura 37 se tiene:

- ❖ **Description** : Descripción del elemento. Esta es la que aparece por defecto para este interruptor, pero se puede modificar según se desee.
- ❖ **Operand**: Es el operando que se encarga del funcionamiento del interruptor. En este caso aparece como Sim.INTERRUPTOR, que en la tabla de símbolos corresponde al sensor I0.0, al que se ha llamado "interruptor". Se puede ver en la siguiente figura.

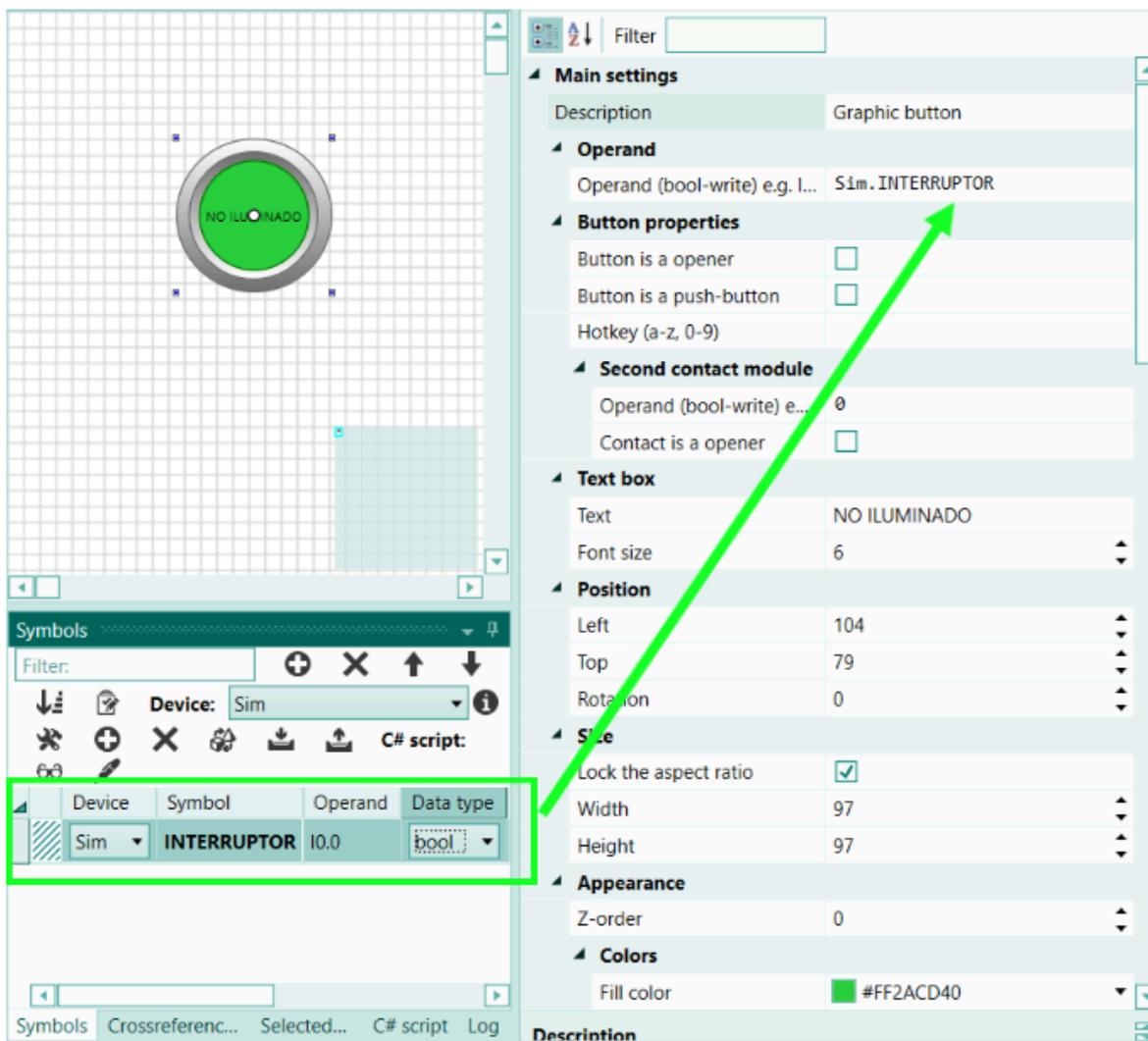


Figura 39. Ejemplo de asignación de un operando.

- ❖ **Button properties:**
  - **Button is a opener:** Si está marcado, funciona de la siguiente manera: cuando se pulsa, se mantiene en la posición presionada sin necesidad de mantener el “dedo” sobre él. Permanecerá en esta posición hasta que se vuelva a presionar, momento en el que regresará a su posición inicial. Por ejemplo, como una lámpara de escritorio con un botón de clic.
  - **Button is a push-button:** Cuando se presiona, activa una función pero no se mantiene presionado. En cuanto se suelta, regresa a su posición original. Por ejemplo, como el funcionamiento de un timbre.
- ❖ **Text Box:** Se configura el texto y su tamaño que aparece en el interior del botón, en este caso “NO ILUMINADO”. (No todos los objetos cuentan con texto en su interior).
- ❖ **Position:** Permite colocar el elemento en la posición deseada. Se puede desplazar hacia la izquierda, hacia arriba o girarlo.
- ❖ **Size:** Configurar el tamaño del objeto (en este caso es de 97x97).
- ❖ **Appearance:** Se configuran los aspectos relacionados con la apariencia del elemento.
  - **Z-order:** Define la capa en la que se encuentra un objeto. Los objetos con un orden Z más alto se superponen a aquellos con un orden Z más bajo. Sus límites son  $-20 < Z\text{-order} < 20$ .
  - **Colors:** Se definen los colores de relleno, la línea que rodea al objeto y el color del texto que aparece en su interior. Además se puede añadir una nueva característica, la cual se muestra en la figura 40 y que es el operando que cambia el color de relleno del objeto. En este caso no se utiliza, pero si se quisiera tener un botón “START” con una “LUZ START” asignada, se agregaría el actuador “LUZ START” a esta casilla. Esto se explicará con más detalle cuando se definan las lámparas.

Operand switching fill color (bool-read) e.g. IM.M0.0	0
---	---

Figura 40. Casilla destinada a asignar un operando para que se produzca un cambio de color.

- ❖ **Images:** La forma del elemento que se quiere emplear se puede sustituir por una imagen, la cual se carga pulsando sobre los tres puntos al final de la casilla “Image status 0” (o “Image status 1” si se quieren dos imágenes).

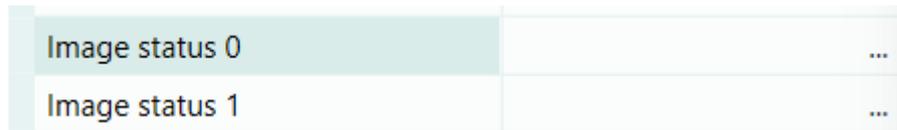


Figura 41. Casilla para cargar una imagen que reemplaza el elemento de dibujo.

- ❖ **Limit switch settings:** En el ejemplo planteado, no se han configurado. Sin embargo, en el caso de un cilindro que se desplaza verticalmente, en este apartado se pueden agregar los sensores correspondientes al cilindro extendido o retraído. Esto se explorará con más detalle más adelante al definir los cilindros.
- ❖ **Magnetism:** Permite que el elemento sea magnético, marcando la casilla “Object is magnetic”. Como se ve en la figura 41, se cuenta con otra posibilidad para configurar el magnetismo y es la casilla “Operand for magnet (bool-read) e.g. IM.M0.0” se utiliza para controlar el magnetismo mediante un sensor que activa o desactiva el mismo. Por ejemplo, se puede tener un interruptor para activar el magnetismo de una pieza hasta que no se presione el interruptor, la pieza no será magnética.

Ya que, tal y como se comentó, el PLC-Lab permite construir escenas bidimensionales, la propiedad del magnetismo es muy interesante. De hecho, es una forma sencilla de evitar que en ciertas situaciones determinados elementos de la escena 'caigan' por el efecto de la gravedad. Por ejemplo, en el caso de la estación 2 de la planta Festo, las piezas se colocan en los huecos de la mesa pero, debido a su poca profundidad, no quedan completamente encajadas en ellos. Esto que en el Factory IO, donde con se construye escenas tridimensionales, no tiene ninguna consecuencia, en el PLC-Lab plantea algunos problemas. Lo que ocurre es que, en ausencia de magnetismo, las piezas se 'caen' cuando la mesa gira. Para que se entienda mejor, se ha grabado un pequeño vídeo [27].

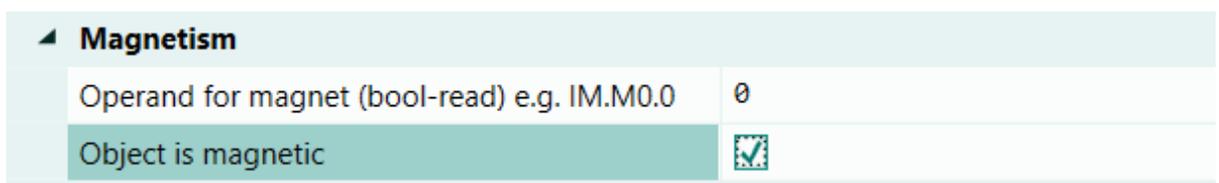


Figura 41. Propiedades para dotar de magnetismo a un objeto en PLC-Lab.

❖ **Weld-joint:**

Se configura cuando se desea que el objeto esté influenciado por una junta. Este concepto se explorará en mayor profundidad más adelante, al examinar los tipos de juntas disponibles en PLC-Lab.

❖ **Creator/destroyer settings:**

En este apartado se pueden configurar los objetos para que dependan de un creador. Aunque el funcionamiento del objeto creador se explicará más adelante, es importante saber que estos creadores son los responsables de generar objetos secundarios asignándoles un ID específico.

En esta casilla se asigna el número de ID correspondiente al objeto que se quiera que dependa de un creador, permitiendo así que este gestione su generación.

❖ **Physics properties:**

En este apartado se pueden asignar las propiedades físicas al objeto.

➤ **Physical type:** Tal y como se observa en la figura 42, pueden ser de tres tipos: estáticos, dinámicos o cinemáticos.

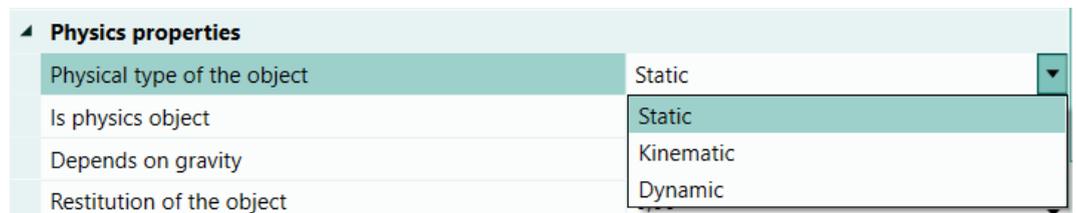


Figura 42. Propiedades físicas disponibles para los objetos en PLC-Lab.

La elección depende de la escena deseada. Se debe tener en cuenta que cuando se activa el modo de ejecución (RUN), los objetos dinámicos y cinemáticos caen hacia abajo mientras que los estáticos permanecen fijos en el plano de dibujo. Para que se entienda mejor este efecto, se ha grabado un pequeño vídeo [27].

➤ **Is physics object:** Se trata de una casilla que indica si se trata de un objeto físico. Si no se selecciona, el objeto es invisible para el motor-físico.

Los elementos de este tipo generalmente vienen con esta casilla marcada por defecto. Sin embargo, en el caso del ejemplo que se ha planteado de interruptor no iluminado, esta propiedad no viene seleccionada.

- **Depends on gravity:** Esta opción permite dotar al objeto de gravedad o no. Por ejemplo, los objetos del tipo físico dinámico caen al activar el modo de ejecución (RUN) ya que esta opción viene activada por defecto. Si se desmarca, el objeto ya no caerá, sino que se quedará “flotando”. Para que se entienda mejor este efecto, se ha grabado un pequeño vídeo [27].

❖ **Advanced settings:**

- **Draw-mode settings:** Se puede configurar el objeto para que su movimiento sea solo horizontal o vertical.
- **Physics properties:**
  - **Collision settings [13].**

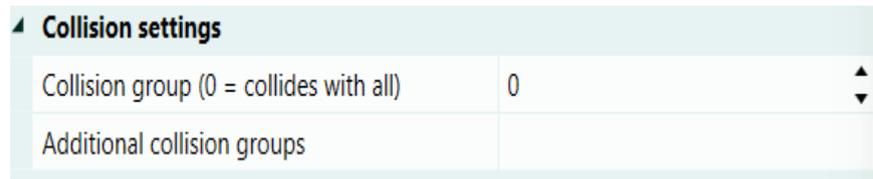


Figura 43. Propiedad grupos de colisión en PLC-Lab.

En este apartado se pueden configurar los grupos de colisión

Para entender a qué se refiere, se comienza definiendo qué son los grupos de colisión:

Los objetos de física dinámica y estática se pueden asignar a grupos de colisión específicos. Por defecto, todos los objetos están en el grupo de colisión 0, lo que implica que chocan con todos los demás objetos físicos que se encuentren en el plano de dibujo.

Los grupos de colisión pueden ser empleados de múltiples formas, por ejemplo, como se muestra en la figura 44, para diseños con separadores.

La siguiente imagen muestra dicho diseño.

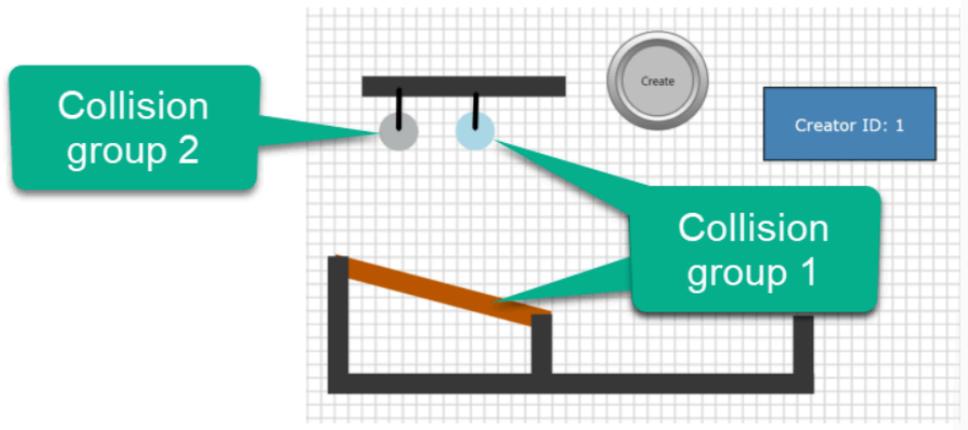


Figura 44. Ejemplo de escena con grupos de colisión en PLC-Lab [14].

El objeto circular azul y el rectángulo marrón en la parte inferior pertenecen al grupo de colisión 1, lo que significa que chocan entre sí. El objeto circular gris pertenece al grupo de colisión 2, por lo que no choca con el rectángulo marrón. Todos los rectángulos negros de los contenedores están configurados en el grupo de colisión 0, por lo que chocan con todos los objetos. Por tanto, mientras que cuando los objetos grises caen atraviesan el plano inclinado, los objetos azules chocan con él y se deslizan pendiente abajo. Así, los primeros quedan en el contenedor de la izquierda y los azules en el de la derecha. Para que se entienda mejor este concepto, se ha grabado un pequeño vídeo [27].

Por último, en este menú de ajustes principales siempre aparece en la parte inferior un apartado "Description", en el cual aparece una pequeña descripción del ajuste que se quiera configurar.



Figura 45. Apartado "Description" del menú de ajustes principales.

Por ejemplo, si se pulsa sobre “Additional collision groups” aparece:

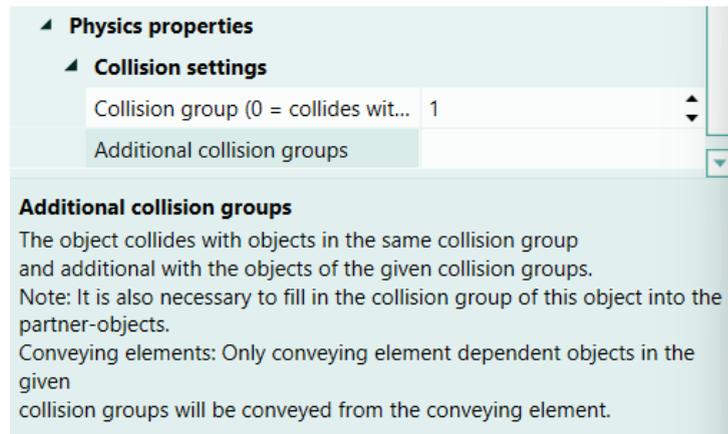


Figura 46. Ejemplo de información en el apartado “Description” para la propiedad “Additional collision groups” en PLC-Lab.

Esto ofrece una pequeña descripción sobre para que se emplea la propiedad de grupos de colisión adicionales e información para su configuración.

A pesar de que PLC-Lab ofrece una amplia gama de elementos de dibujo para la creación de proyectos, solo se verán en profundidad aquellos que se han empleado en este trabajo de fin de grado. No obstante, si se desea explorar el resto de elementos disponibles, se puede hacer a través del siguiente enlace: [14] en el apartado “Objetos en PLC-Lab”.

Los elementos de dibujo empleados para la creación de las distintas estaciones que componen la Planta Festo han sido los siguientes:

❖ **Switches (Interruptores):**

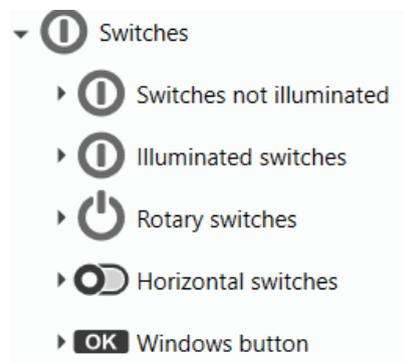


Figura 47. Interruptores disponibles en PLC-Lab.

- Switches not illuminated: Interruptor que no se ilumina.
- Illuminated switches: interruptores que cuando son pulsados se iluminan.

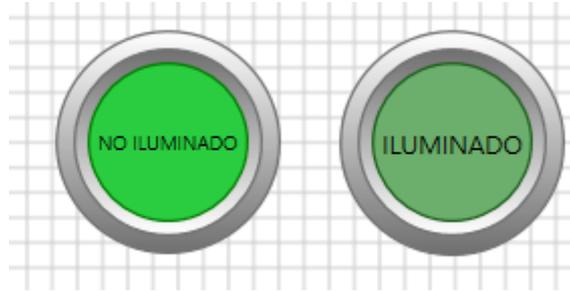


Figura 48. Símbolos interruptor iluminado y no iluminado.

- Rotary Switches: Se pueden tener dos opciones para este tipo de interruptores, como se ve en la figura 49.
  - ON-OFF: Únicamente rotan hacia dos posiciones ON u OFF.
  - Multiple Switch: Se pueden configurar con hasta 4 posiciones.

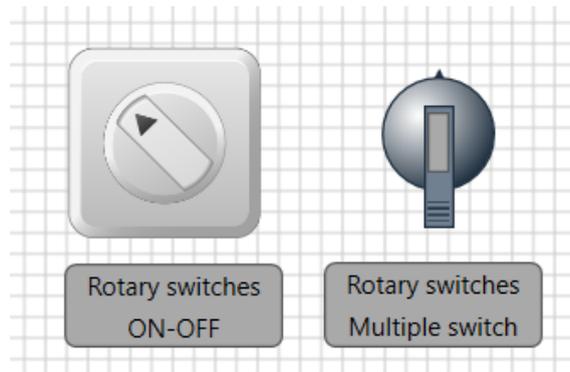


Figura 49. Símbolos de interruptores rotatorios disponibles en PLC-Lab.

- Horizontal switches:
  - ON/OFF: Pueden estar únicamente a ON u OFF.
  - Multiple Switch: Permite tener varias posiciones. Se puede configurar su número de posiciones tanto a la izquierda como a la derecha y su posición inicial, con un máximo de cuatro posiciones a la derecha y tres a la izquierda. Por ejemplo, el interruptor mostrado en el lado derecho de la figura 50, está configurado con dos posiciones a la derecha y dos a la izquierda, con la posición inicial ubicada en el 0.

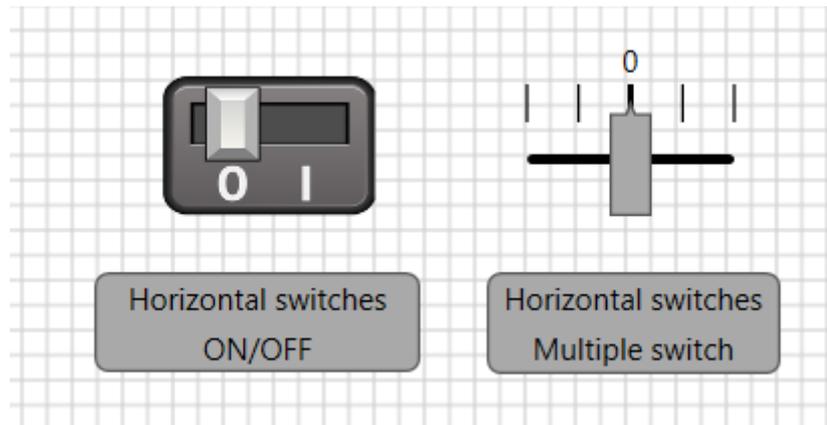


Figura 50. Símbolos de los interruptores horizontales disponibles en PLC-Lab.

❖ **Lámparas:**

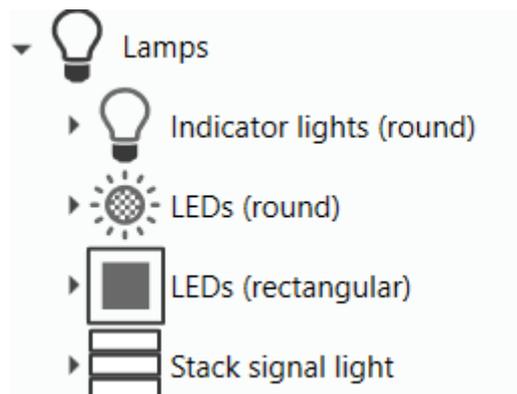


Figura 51. Símbolos de las lámparas disponibles en PLC-Lab.

- Indicator lights (round): Luces que son empleadas como indicador. Por ejemplo si se quiere que cuando esté produciendo de forma normal haya un indicador de luz verde encendido.
- LEDs (round) : Indicador LED de forma redonda.

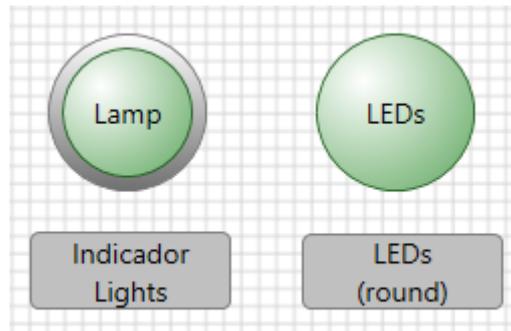


Figura 52 Símbolo de los indicadores luminosos y LEDs redondos de PLC-Lab.

Si se desea conocer los demás tipos de lámparas disponibles en PLC-Lab, puede hacerlo a través del siguiente enlace [15].

### ❖ Limit Switches



Figura 53. Finales de carrera en PLC-Lab.

En esta sección, se encontrará un objeto rectangular preconfigurado como tipo físico estático.

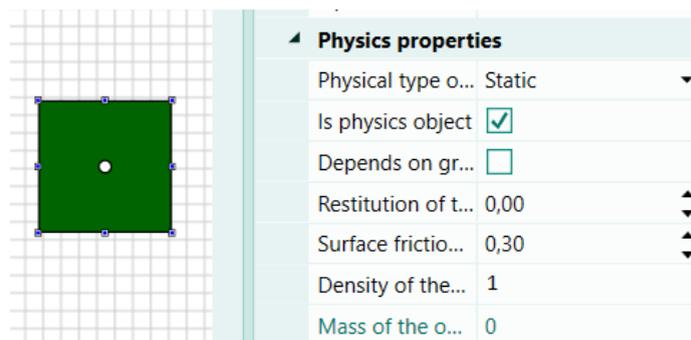


Figura 54. Símbolo de los finales de carrera en PLC-Lab.

Además, se establece la propiedad “Neutral durante la colisión”, tal y como se muestra en la figura 55.

Limit switch settings	
Limit switch operand (bool-write) e.g. IM.M0.0	0
Is opener	<input type="checkbox"/>
Neutral while collision	<input checked="" type="checkbox"/>
Do not influence limit switches	<input type="checkbox"/>
Limit switch changes collision group	<input type="checkbox"/>
New collision-group when collide (operand-UI...	0
Distance-sensor: distance in pixels (operand-U...	0
Distance-sensor: Origin to the right	<input type="checkbox"/>
Distance-sensor: Invert signal	<input type="checkbox"/>

Figura 55. Propiedad “Neutral durante colisión” de los finales de carrera en PLC-Lab.

La configuración que se muestra en la figura 55 indica que el objeto detecta la colisión con el objeto físico, pero no aplica fuerza sobre él. En esta sección también se encuentra la propiedad “Limit switch operand (bool-write) e.g. IM.M0.0”, donde se debe especificar el operando que se cambiará en caso de una colisión [16].

Configurando la propiedad de grupos de colisión se puede realizar una identificación de material o una identificación de color con estos interruptores. Para que se entienda mejor este concepto, se ha grabado un pequeño vídeo [27].

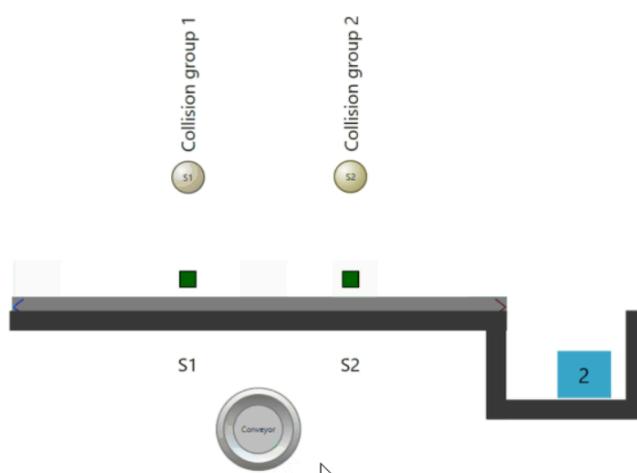


Figura 56. Ejemplo de escena con finales de carrera como detectores en PLC-Lab.

Estos "Limit Switches" se pueden configurar también como sensores de distancia y sensores para detectar la altura/anchura de los objetos. Esto se encuentra en la configuración de la figura 55, en las casillas mostradas en la siguiente figura 57:

Distance-sensor: distance in pixels (operand-U...	0
Distance-sensor: Origin to the right	<input type="checkbox"/>
Distance-sensor: Invert signal	<input type="checkbox"/>

Figura 57. Propiedades de los finales de carrera.

Si el objeto ha sido definido como sensor de distancia tendrá la siguiente apariencia en el modo de dibujo:

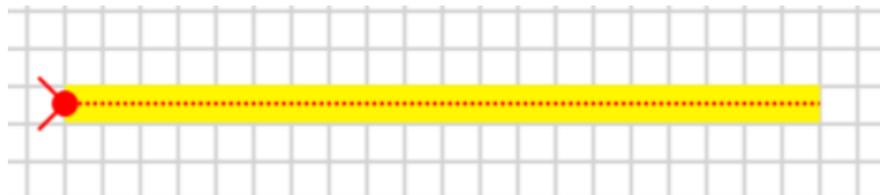


Figura 58. Símbolo del sensor de distancia en PLC-Lab.

En el lado izquierdo del sensor hay una flecha roja y un círculo rojo que representan el origen o punto cero del sensor. Este punto puede modificarse usando la propiedad "Origen a la derecha", lo cual hace que los símbolos se muestren en el lado derecho del sensor. En el centro del sensor, se encuentra una línea discontinua que simboliza el láser utilizado para la medición, indicando así que el punto de medición se encuentra en el centro del objeto. Cuando se inicia la simulación, estos símbolos se eliminan y solo queda visible el rectángulo [16].

## ❖ Geometric shapes

Son las formas que se pueden emplear para la creación de las escenas y, tal y como se muestra en la figura 59, pueden ser de forma rectangular, elíptica o triangular.

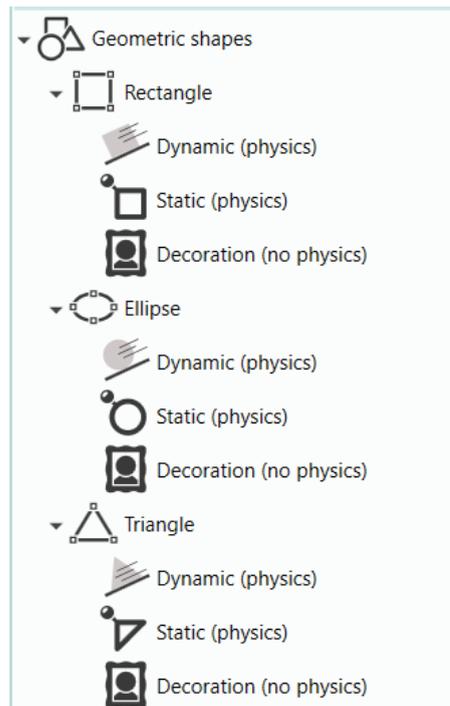


Figura 59. Formas geométricas disponibles en PLC-Lab.

En los tres casos se puede elegir entre el tipo físico dinámico, estático o tipo decoración. Ese último se utiliza, tal y como su nombre indica, como decoración de la escena. Por ejemplo, si se desea tapar una determinada parte del dibujo con un rectángulo, se debe elegir la forma rectangular y, dentro de ésta, el tipo decoración.

## ❖ Cylinder

En el PLC-Lab se dispone de tres tipos de cilindros, que se muestran en la figura 60.



Figura 60. Cilindros disponibles en PLC-Lab.

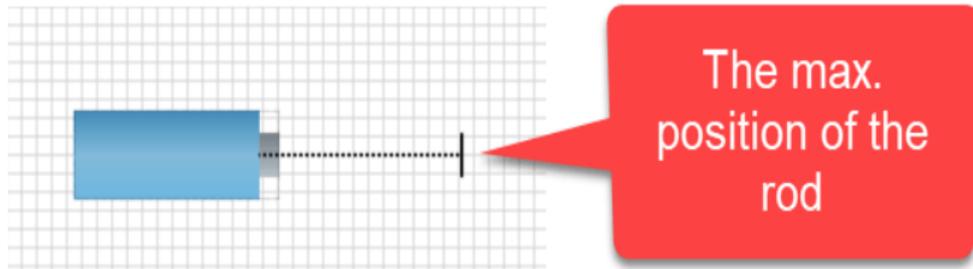


Figura 61. Símbolo de los cilindros disponibles en PLC-Lab.[14]

En este Trabajo de Fin de Grado se han empleado cilindros que actúan con resorte (Acting with spring). Se trata de un cilindro de simple acción que funciona con resorte. Por tanto, si el operando tiene el estado '1', el vástago del pistón se moverá en esa dirección. En cambio, cuando el operando tiene el estado '0', el vástago del pistón retrocede, [17].

A los cilindros se les puede configurar sus propiedades, tales como cambios de color, configuración de su posición, su tipo físico y asignación de un grupo de colisión. Además, cuentan con sus propias propiedades específicas, que son las que se muestran en la figura 62.:

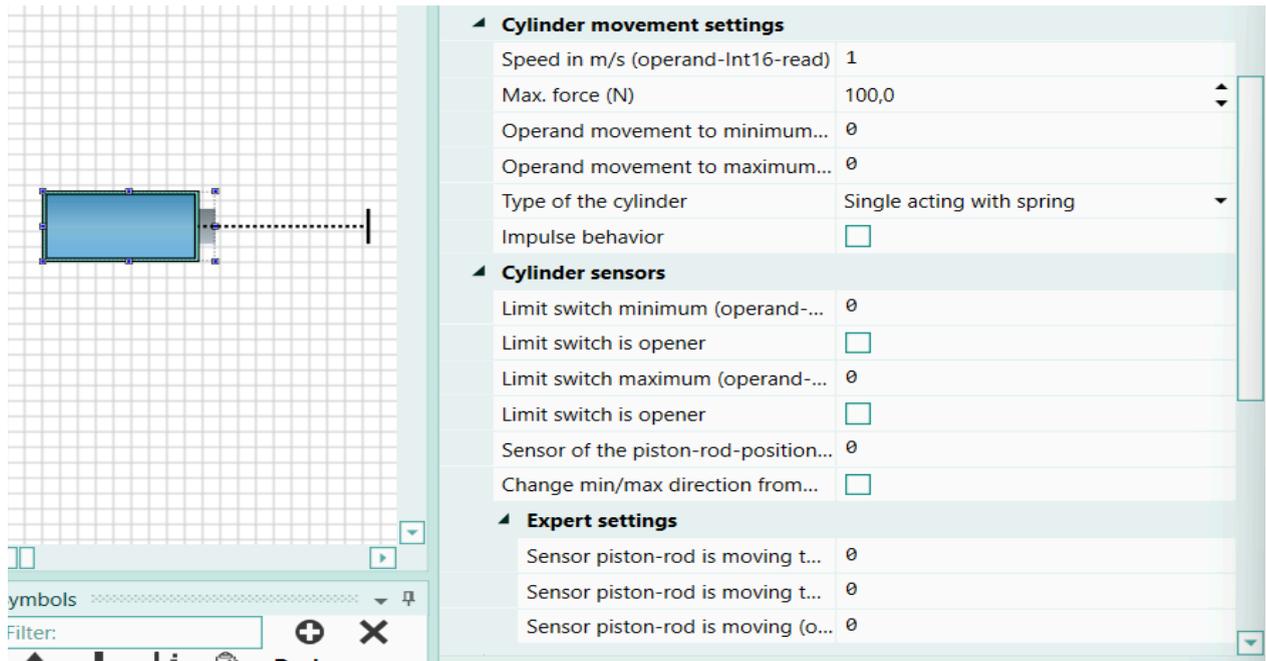


Figura 62. Propiedades de los cilindros disponibles en PLC-Lab.

Se puede configurar la fuerza del cilindro y asignarle un operando para que realice el movimiento hasta su máximo y mínimo. Además, se pueden agregar sensores (Cylinder sensors) que indican si el cilindro se encuentra en su posición máxima o mínima.

Si se desea conocer el funcionamiento de los otros dos tipos de cilindro disponibles, se puede hacer a través del siguiente enlace [17].

#### ❖ Conveying element [18].

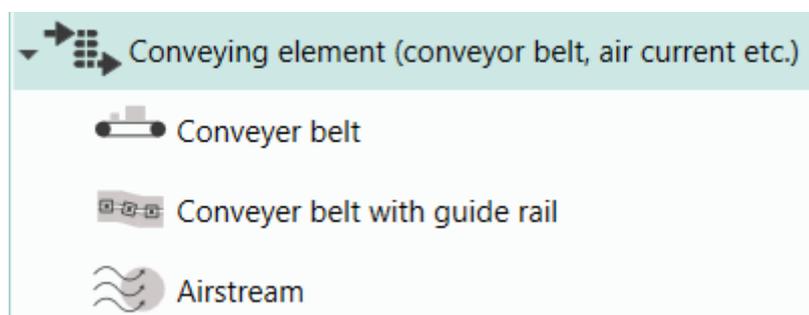


Figura 63. Elementos transportadores disponibles en PLC-Lab.

Con este objeto se pueden crear cintas transportadoras, corrientes de aire, etc. Un elemento transportador afecta a los objetos que dependen de él, siempre que se encuentren dentro de la zona del elemento transportador.

Si un objeto se encuentra en la zona de influencia del elemento transportador, el objeto se desviará.

Los elementos transportadores se pueden girar, lo cual también gira la dirección del movimiento.

A partir de la versión 1.5.0.9 de PLC-Lab se han introducido diferentes tipos de elementos transportadores. Aunque se trata del mismo objeto, se han realizado ajustes que lo hacen ideal para determinadas aplicaciones.

Los tipos disponibles son:

- Cinta transportadora.
- Corriente de aire.
- Cinta transportadora con carril guía (disponible a partir de V1.5.2).

En este trabajo se han empleado las cintas transportadoras, pero si se desea conocer obtener más información sobre los otros dos tipos se, puede consultar el siguiente enlace.

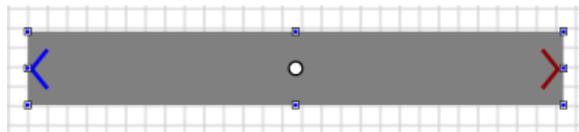


Figura 64. Símbolo de la cinta transportadora en PLC-Lab.

Se pueden representar en vista superior o lateral. Cuando se transportan objetos a través de una cinta transportadora su dinámica se ve restringida por la propia dinámica del transportador, lo que es similar a la fricción de la cinta con los objetos. Dos propiedades "amortiguación del movimiento lineal de los objetos transportados" y "amortiguación del movimiento giratorio de los objetos transportados" determinan este comportamiento. Aumentar estos valores asegura que los objetos transportados se detengan de inmediato cuando se apaga el motor de la cinta transportadora.

Esto se configura en el apartado “configuraciones expertas” (expert settings) de las propiedades de la cinta transportadora.

Expert settings	
Velocity set direction	<input type="checkbox"/>
Compensate the mass influence	<input checked="" type="checkbox"/>
Damping of the linear movement of the tra...	10
Damping of the rotation movement of the t...	10
Is exclusive conv. belt	<input type="checkbox"/>
Change rotation of conveyed objects. Oper...	0
Rotation offset on collision (operand-Int16-...	0
Amount of incremental rotation sequences	10
With guide rail in the middle of the object	<input type="checkbox"/>
Tolerance of the guide rail in the middle of...	8

Figura 65. Propiedades de la cinta transportadora en PLC-Lab.

### ❖ Joints

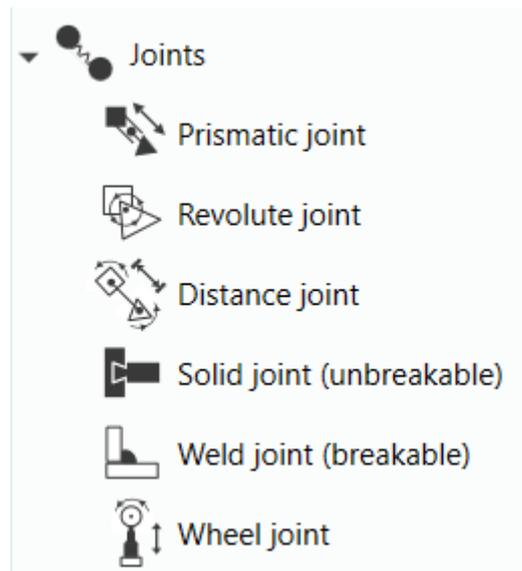


Figura 66. Símbolo de las juntas disponibles en PLC-Lab.

Las juntas permiten crear movimientos lineales similares a los de cilindros, movimientos giratorios y conexiones de ruedas, entre otros. Siempre definen la relación entre dos objetos físicos. Estos dos objetos son los elementos de anclaje para la articulación.

- **Junta prismática (Prismatic joint) [19].**

Las juntas prismáticas se emplean, por ejemplo, en cilindros, ascensores, plataformas elevadoras, etc. A continuación se muestra el símbolo que corresponde a este elemento:

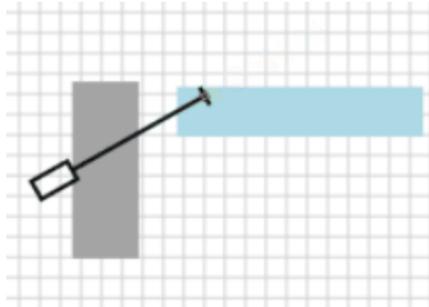


Figura 67. Ejemplo junta prismática en PLC-Lab.

En la figura 67 se observan dos objetos conectados por una junta prismática. El objeto de la izquierda es el objeto maestro y el segundo objeto se desplaza a lo largo de la línea del símbolo de unión.

La longitud de la junta se puede ajustar para definir la duración del movimiento, esto puede hacerse de tres formas:

- Seleccionando y moviendo el objeto esclavo.
- Seleccionando el símbolo de la junta y moviendo el recorrido de desplazamiento.
- A partir de las propiedades, se selecciona la junta y se cambia la propiedad "longitud del movimiento lineal" (Length linear movement (pixel))

▲ General	
Anchor objects collide	<input type="checkbox"/>
Start-point	165; 200
End-point	168; 117
Length linear movement (pixel)	83

Figura 68. Propiedad para cambiar la longitud en una junta prismática en PLC-Lab..

Además de la longitud, también se puede configurar su rotación a través de tres opciones:

- Seleccionando y moviendo el objeto esclavo.
- Haciendo doble clic sobre el símbolo de la junta, mediante lo cual se entra en el modo rotación, que se muestra en las figuras 69 y 70.

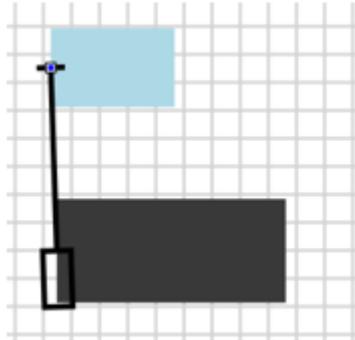


Figura 69. Apariencia de la junta sin estar en modo rotación (el punto de su final aparece en color azul)

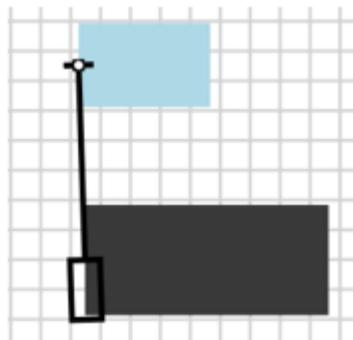


Figura 70. Apariencia de la junta al estar en modo rotación (el punto de su final aparece en color blanco)

- Cambiándolo en el apartado de propiedades Size>Rotation



Figura 71. Propiedad para cambiar la rotación en una junta prismática en PLC-Lab..

La junta cuenta con un motor que se puede configurar mediante la propiedad “Motor” en la sección “Physics properties” > “General > “Configuración del motor”.

Physics properties	
General	
Anchor objects collide	<input type="checkbox"/>
Start-point	165; 200
End-point	163; 135
Length linear movement (pixel)	65
Motor settings	
Motor is active	<input checked="" type="checkbox"/>
Speed in m/s (operand-Int16-read)	1
Speed divisor	1,0
Max. force (N)	100,0
Operand movement to minimum (b...	0
Operand movement to maximum (b...	1
Motor force depends on the status...	<input type="checkbox"/>
Reverse velocity if status of operand...	<input type="checkbox"/>
Impulse behavior	<input type="checkbox"/>
Velocity set direction	<input type="checkbox"/>
Set motor speed to zero when locks...	<input type="checkbox"/>

Figura 72. Propiedades para configurar el motor en una junta prismática en PLC-Lab.

En la figura anterior se puede ver que hay varias propiedades que se pueden configurar con lo respectivo al uso de la junta como un motor, sin embargo en este trabajo se verán las que han sido empleadas. Si se quiere conocer más acerca de estas opciones y conocer ejemplos sobre el uso de esta junta, se puede hacer a través del siguiente enlace.

“Anchor objects collide” : Como sugiere el nombre de esta propiedad, es utilizada para lograr que el objeto maestro y el esclavo colisionan si esta se activa.

“Operand movement to minimum (bool-read) e.g. IM.M0.0” y “Operand movement to maximum (bool-read) e.g. IM.M0.0” : El movimiento del objeto esclavo está controlado por operandos, los cuales se añadirán en estas casillas. Dado que el valor se lee del operando, también se puede especificar la constante ‘1’ para un movimiento continuo.

Por último la junta cuenta con unos finales de carrera y sensores. Se configuran en la propiedad “Motor” en la sección “Physics properties” > “General > “Finales de carrera”. Cuenta con numerosos sensores que permiten evaluar las posiciones finales, medir la posición del objeto esclavo como un valor analógico y realizar muchas otras funciones.

▲ Limit switches	
Limit switch minimum (operand...	0
Limit switch is opener	<input type="checkbox"/>
Limit switch maximum (operan...	0
Limit switch is opener	<input type="checkbox"/>
Sensor position in pixels (opera...	0
Change min/max direction fro...	<input type="checkbox"/>
Sensor speed m/s (operand-Int...	0
Gain factor for rotating velocity	1
Sensor force N (operand-Int16-...	0
Sensor moving to maximum (o...	0
Sensor moving to minimum (op...	0
Sensor a moving is happened (...)	0

Figura 73. Propiedades para configurar los finales de carrera en una junta prismática en PLC-Lab.

- **Junta de revolución (Revolute joint) [19].**

Se utiliza para implementar movimientos giratorios, puertas giratorias, mesas giratorias, barreras, rampas variables entre muchos otros.

La siguiente figura 74 muestra dos objetos conectados por una "junta de revolución". El objeto inferior actúa como el objeto maestro, que generalmente será un objeto fijo. Una línea continua va desde el objeto maestro hasta el punto de pivote, representado por un círculo de color con una cruz en su interior. Las líneas de conexión con el objeto esclavo son discontinuas. El objeto esclavo es el objeto que gira alrededor del punto de pivote.

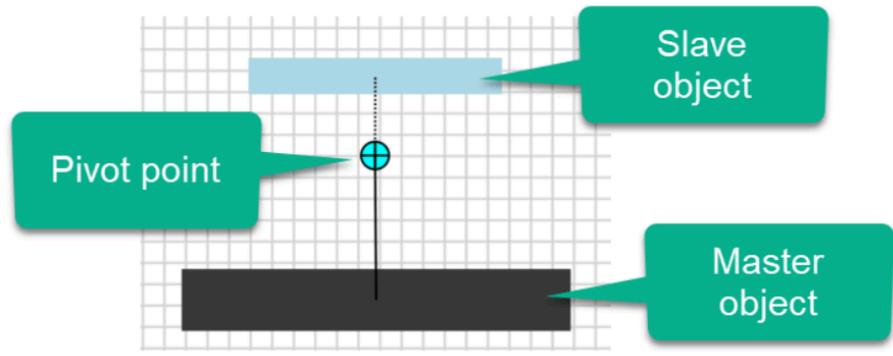


Figura 74. Ejemplo junta de revolución en PLC-Lab [19].

Esta junta incluye un motor que se puede configurar a través de la propiedad "Motor" en el apartado "Propiedades físicas" > "General" > "Configuración del motor". Como en la junta anterior, se puede configurar la velocidad del motor (en rpm en este caso) especificando un operando, del cual se lee el valor. También es posible especificar una constante si no es necesario que la velocidad sea variable.

La junta permite la rotación hacia la derecha o la izquierda del objeto esclavo, la cual se controla mediante operandos, y también se puede especificar la constante '1' para una rotación continua. Sin embargo, es importante tener en cuenta que solo puede estar activo un sentido de rotación. Si ambos sentidos de rotación están activos, no se producirá el movimiento giratorio.

Además, se cuenta con finales de carrera y sensores, configurados en la propiedad "Motor" en la sección "Physics properties" > "General" > "Finales de carrera". Si el movimiento giratorio es limitado, también se pueden especificar sensores de posición final.

Dentro de la sección de propiedades "Propiedades físicas->General->"Finales de carrera" está disponible la propiedad "Habilitar límites". Se puede utilizar esta propiedad si se desea limitar el movimiento giratorio, antes de activar esta opción se debe establecer un límite superior e inferior en grados. Los valores permitidos para cada caso son:

- Límite inferior: -180° a 0°.
- Límite superior: 0° a 180°.

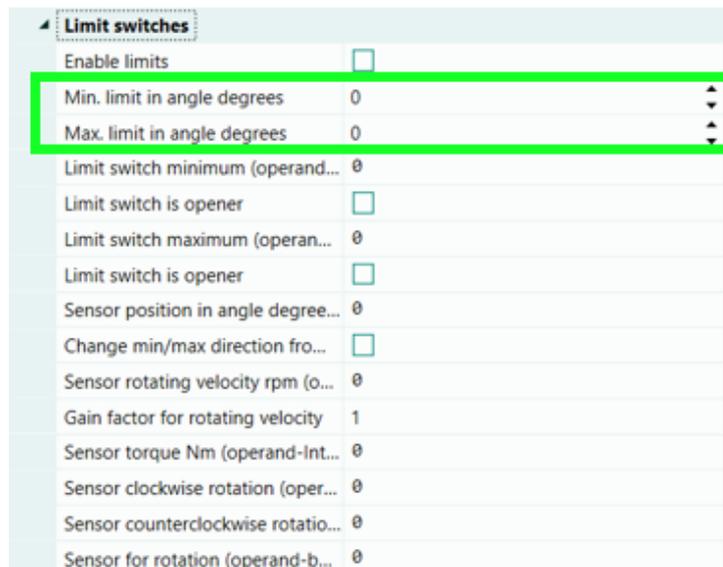


Figura 75. Propiedades para configurar los ángulos de giro en una junta de revolución en PLC-Lab.

Para obtener más información sobre el resto de propiedades y ver algunos ejemplos de este tipo de junta, puede consultar el siguiente enlace en el apartado “Revolute joint”.

- **Junta distanciada (Distance joint) [19].**

La junta de distancia constante es empleada cuando se necesita conectar dos objetos mediante una varilla. Este tipo de unión se utiliza para mantener objetos principales en una ubicación específica cuando un creador los multiplica. También puede ser empleada para implementar un péndulo.

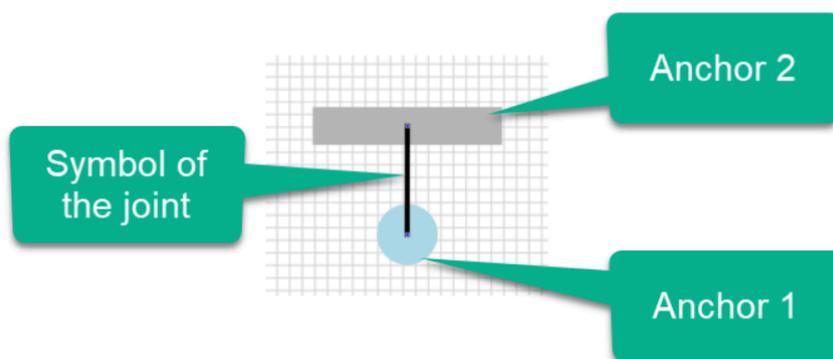


Figura 76 . Objetos conectados por una junta de distancia. La línea negra simboliza la conexión en sí [19].

- **Junta sólida (Solid joint) [19].**

Se emplea cuando dos objetos deben conectarse de manera inseparable, o cuando se quiere unir uno o más objetos al vástago de un cilindro.

En la siguiente imagen, dos objetos están conectados por una unión sólida. El objeto de la izquierda es el objeto maestro. Con este objeto se inició el dibujo de la articulación. El objeto esclavo (círculo) se encuentra a la derecha”.

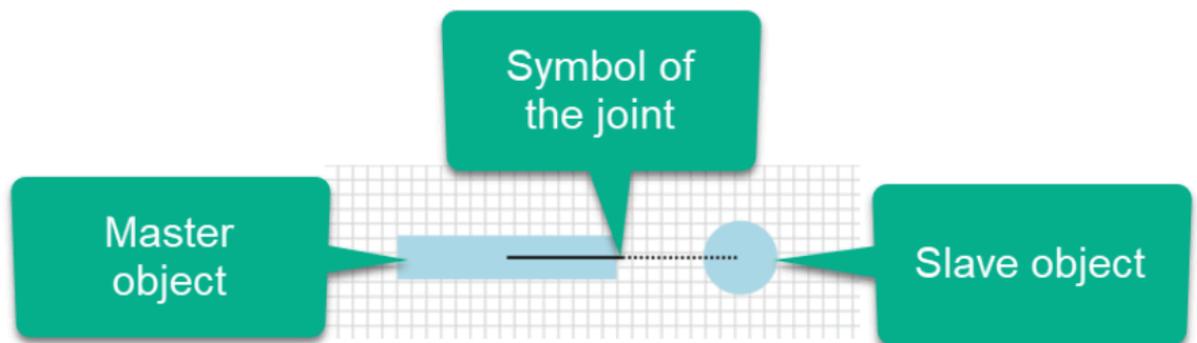


Figura 77. Objeto conectado por una junta sólida. La línea negra simboliza la conexión en sí [19].

Cuando se crea esta unión, el objeto esclavo adopta algunas propiedades del objeto maestro. Por esto, es importante asignarle el rol de objeto maestro al objeto que sea “más dominante”.

- **Junta de rueda (Wheel joint) [19].**

Se utiliza principalmente cuando se quiere una conexión de rueda con amortiguación. Puede ser empleado para, por ejemplo, crear la mesa giratoria con la que se cuenta en la estación 2.

Al crear la articulación, se debe comenzar con el objeto que representa la parte fija de la articulación de la rueda. El segundo objeto de anclaje para la articulación será el objeto esclavo que girará.

“La siguiente imagen muestra dos objetos que están conectados por una "junta de rueda". El objeto inferior es el objeto maestro, que en muchos casos será un objeto fijo. Desde el objeto maestro, una línea sólida ligeramente más ancha conduce al centro del símbolo de unión. También desde el centro del símbolo de la articulación parte una línea ligeramente más fina que conduce al objeto esclavo”.

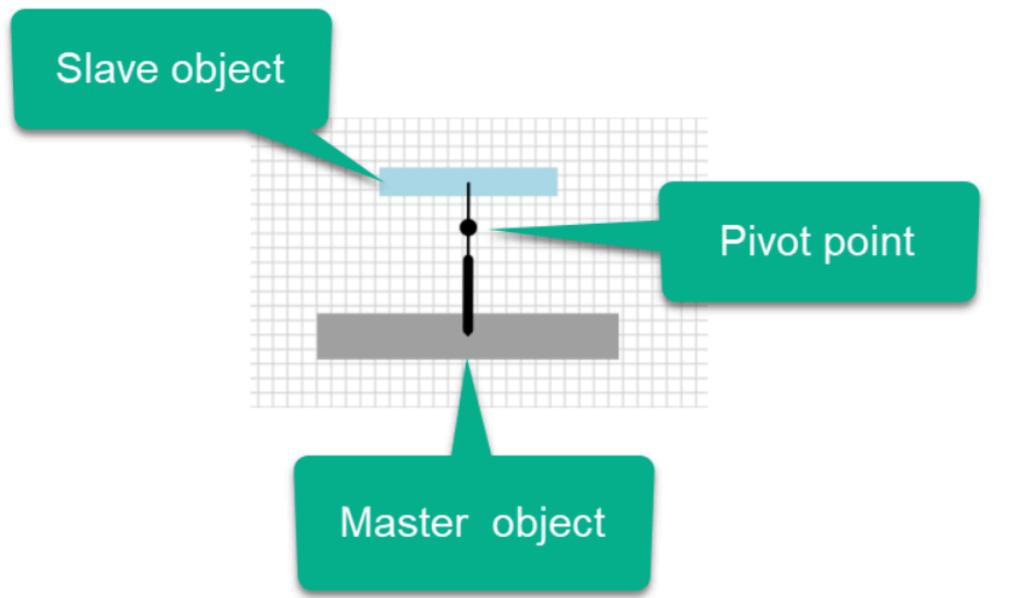


Figura 78. Objetos conectados por una junta de rueda. La línea negra simboliza la conexión en sí. El círculo indica el punto de pivote. El objeto esclavo es el que gira alrededor de este punto de pivote [19].

La articulación incluye un motor que se puede configurar a través de la propiedad "Motor" en la sección "Propiedades físicas > General > Configuración del motor". La velocidad del motor (en rpm) se puede definir usando un operando. También es posible especificar una constante si no se necesita que la velocidad sea variable.”

Si se desea obtener más información sobre las demás características disponibles de las juntas, o conocer los otros tipos, se puede hacer a través del siguiente enlace [19].

❖ **Creador (Creator) [20].**

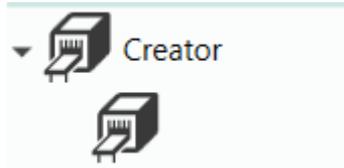


Figura 79. Objeto “Creador” en PLC-Lab.

Los creadores generan objetos secundarios con un ID específico. Cuando se cumple la condición para crear un nuevo objeto secundario, estos se crean como copias de sus objetos principales en la posición de aquellos

El creador de objetos permite generar fuentes de objetos de un sistema, por ejemplo, para suministrar piezas a elementos de transporte, que es para lo que ha sido empleado en este trabajo de fin de grado.

Los pasos para crear un objeto dinámico a partir de un creador son los siguientes:

1. Se crea el objeto dinámico en el espacio de dibujo (por ejemplo una elipse dinámica).
2. En las propiedades de esa elipse dinámica, en la sección “Configuración principal” > “Configuración del creador/destructor”, se asigna una ID única (entre 1 y 30) en la propiedad “El objeto es un objeto principal con la ID”, como se muestra en la siguiente figura.

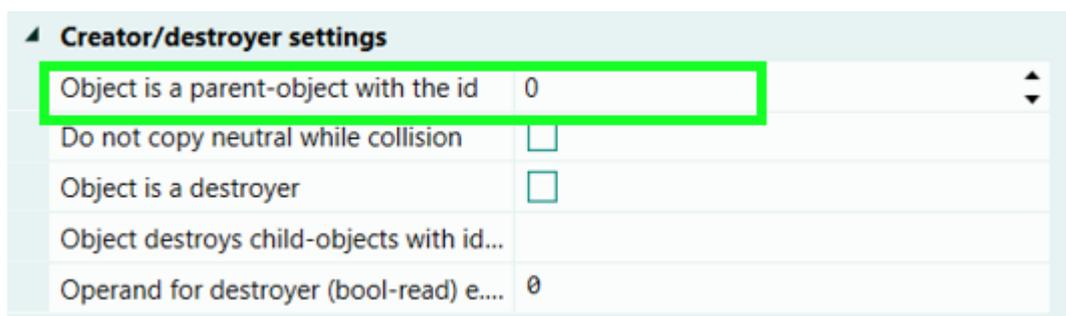


Figura 80. Propiedades del objeto dinámico creado en PLC-Lab.

3. Se fija el objeto principal en la posición donde se crearán los secundarios. Para ello se suele emplear un objeto fijo (estático), al

que se le conecta el objeto principal mediante una junta de distancia constante, como se muestra en la siguiente figura:

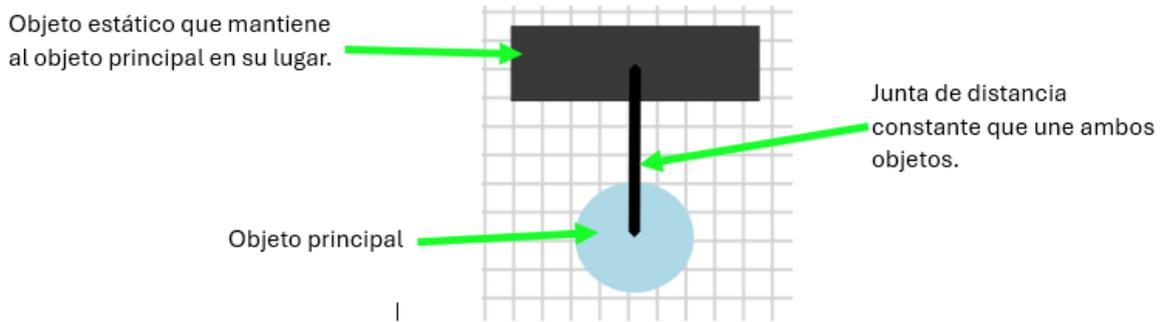


Figura 81. Ejemplo objeto dinámico y estático unidos por una junta de distancia en PLC-Lab.

- Se crea el objeto “Creador” en el espacio de dibujo y se le asigna la ID que se haya establecido en el objeto principal en sus propiedades, en el apartado “Creator properties” > “ Creator has the id”

▲ Creator properties	
Creator has the id	0
Create trigger operand (bool-read) e.g...	0
Trigger-type	Positive edge ▼
Trigger-pause-time (milliseconds)	100
Max. create cycles (0 = no limit)	10000

Figura 82. Propiedades del objeto “Creador” en PLC-Lab.

- Se especifica un operando ( o una constante) en las propiedades del creador, en el apartado “Creator properties”, en la casilla “ Create trigger operand (bool-read) e.g. IM.M0.0” que se muestra en la figura 82, para controlar la creación de los objetos secundarios.

Se puede dotar al creador con un máximo de ciclos de creación, propiedad que se ve en la figura figura 82. Si se alcanza este número, no se crearán más objetos secundarios.

El siguiente ejemplo muestra el diseño de una escena con un creador:

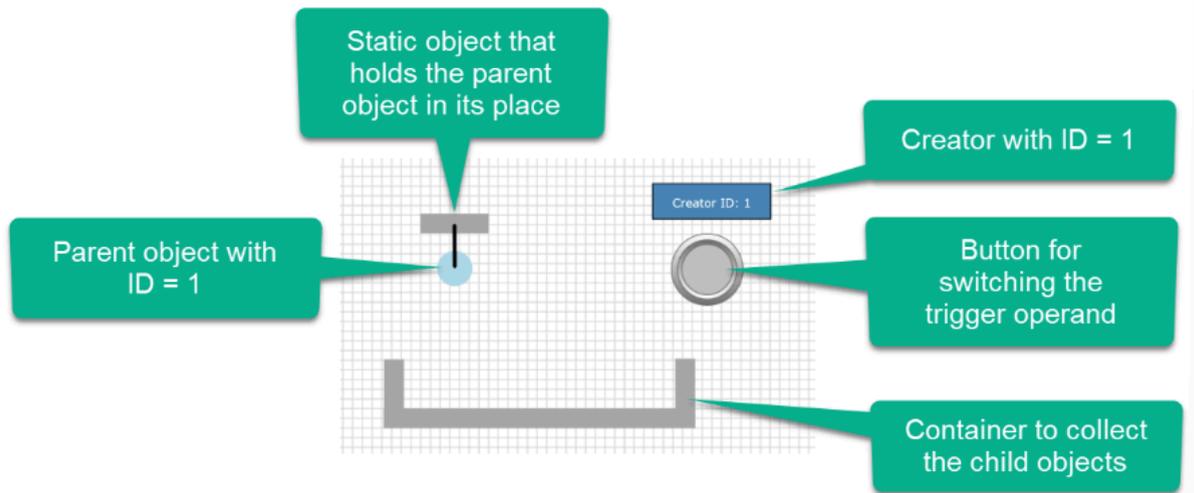


Figura 83. Ejemplo escena que incluye un “Creador” en PLC-Lab [20].

“El objeto principal que creará el creador es un círculo. Este objeto principal tiene el ID = 1, lo que significa que la propiedad "El objeto es un objeto principal con el ID" se establece en el valor "1".

El círculo está unido a un objeto fijo mediante una junta de distancia constante. De esta manera, el objeto principal siempre permanece en la misma posición y los objetos secundarios se insertan en el diseño desde allí.

El creador está simbolizado por un rectángulo azul. La propiedad "El creador tiene el ID" se establece en "1", por lo que el Creador es "responsable" de este ID. El bit operando IM.M0.0 se ha especificado como "Crear operando de activación"; esto debería desencadenar la creación La conmutación del operando de activación IM.M0.0 se realiza mediante un pulsador al que se le ha asignado este operando.

Los objetos secundarios generados se recopilan en un contenedor ubicado debajo del objeto principal y se ensamblan mediante rectángulos fijos.

Si se desean ver más características u otros usos que se le puede dar al objeto “Creador” visite el siguiente enlace [20].

❖ **Objetos indicadores (Status or info-objects)**

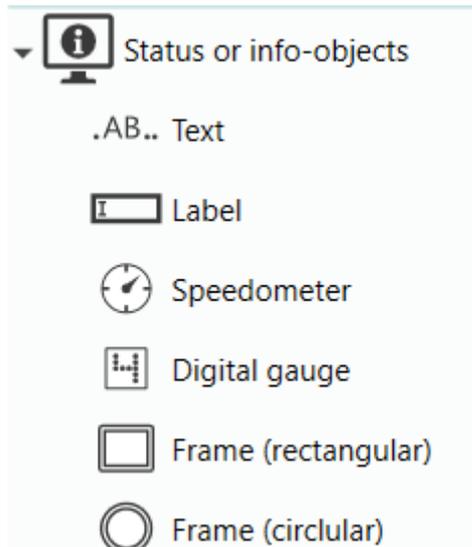


Figura 84. Objetos indicadores disponibles en PLC-Lab.

Aquí se incluyen los objetos que se pueden utilizar para mostrar valores o texto en una escena. Algunos también pueden utilizarse como efectos visuales para resaltar situaciones especiales en el diseño.

En este proyecto solamente se han empleado los dos primeros: “.AB..Text” y Label. Ambos empleados para información de texto o de un operando en las escenas. Esto se configura a través de sus propiedades. Si lo que se quiere es añadir un texto, se configura en el apartado “Text box”. Si por el contrario se quiere que muestre el valor de un operando, por ejemplo en la estación 1 se quiere que el sensor que mide la longitud de las piezas muestre el valor de la medida en la pantalla, solo hay que definir el operando correspondiente en sus propiedades en el apartado “Operand” como se ilustra en la figura 86.

Text box	
Header text	--Text--
Text second line	
Font size	16

Figura 85. Propiedad para configurar el texto del objeto indicador “.AB..Text” en PLC-Lab.

Operand	
Operand (Any64-read)	0
Show the value binary	<input type="checkbox"/>
Operand switching fill color (bool-re...	0
Operand flashing (bool-read) e.g. IM...	0
Operand pulsing (bool-read) e.g. IM...	0

Figura 86. Apartado para dotar al objeto indicador “.AB..Text” de un operando en PLC-Lab.

Si se desea más información sobre el resto de propiedades o sobre el resto de elementos disponibles para este apartado, se puede consultar el siguiente enlace [21].

En este apartado de la memoria se han descrito todos los elementos de dibujo que han sido empleados en este Trabajo de Fin de Grado. No obstante, en el PLC-Lab hay muchos otros disponibles, que se pueden consultar en el siguiente enlace [22]. “Manual-Online-PLC-Lab > Objetos en PLC-Lab (Disponibles en el menú desplegable de la izquierda)”

#### ❖ Acciones (Actions)

Recordando la figura 34, en PLC-Lab hay una serie de acciones que se pueden realizar sobre el espacio de dibujo. Se hablará a continuación sobre las que han sido empleadas en este trabajo de fin de grado.

- Grupos corporales (Body Group) [23].

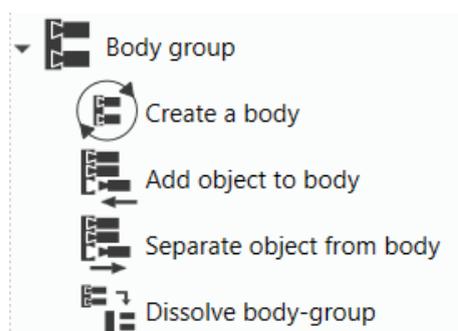


Figura 87. Acciones disponibles en PLC-Lab.

En PLC-Lab es posible unir formas geométricas dinámicas para la formación de cuerpos. Estas formas se convierten en una unidad en la simulación y están conectadas de forma inseparable.

La creación de un grupo corporal tiene el mismo efecto que la de las juntas sólidas (Solid joint) anteriormente descritas. Esto plantea la cuestión de cuándo se debería usar uno u otro. La respuesta es sencilla: cuando se quiere combinar más de dos objetos en un solo cuerpo, se utiliza la acción “Crear un grupo corporal” y en caso de que sean solamente dos elementos los que se quieran unir, se usa la junta.

Los grupos corporales se identifican en el espacio de dibujo porque están rodeados por un marco:

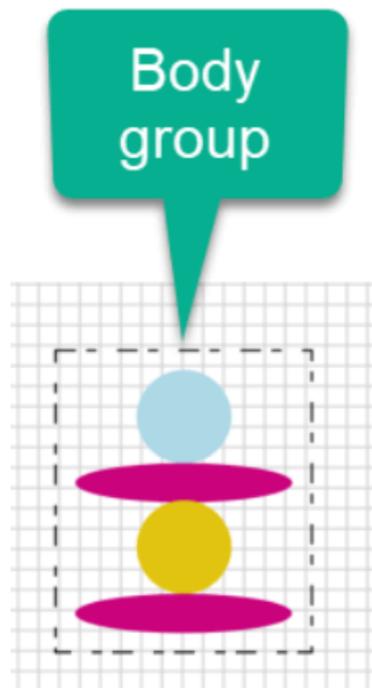


Figura 88. Ejemplo de grupo corporal en PLC-Lab [23].

Los pasos que deben seguirse para su creación son:

1. Se seleccionan los objetos que se desean combinar en un grupo corporal.
2. Se elige la acción “Crear un cuerpo” (Create a Body) de la figura 87.

Si se desea mover el grupo corporal, solamente hay que seleccionarlo y moverlo a donde se desee. También se puede seleccionar o mover un objeto que forme parte del grupo corporal, para lo cual primero se selecciona el grupo y, a continuación, se selecciona el objeto que se quiera haciendo clic sobre él.

En la figura 87 hay disponibles otras acciones para los grupos corporales, pero las empleadas en este trabajo han sido.

- Separate object from body: Utilizado para separar objetos individuales de un grupo corporal existente. Se procede de la siguiente manera:
  1. Se selecciona el grupo corporal.
  2. Se selecciona el objeto que se quiera separar del grupo.
  3. Se elige la acción “Separate object from body”.
- Dissolve body- group: Acción utilizada para disolver el grupo corporal. Se seleccionan (uno por uno) los elementos del grupo corporal, se elige esta acción y ya se tendría el grupo corporal disuelto.
- Alinear objetos (Align objects)

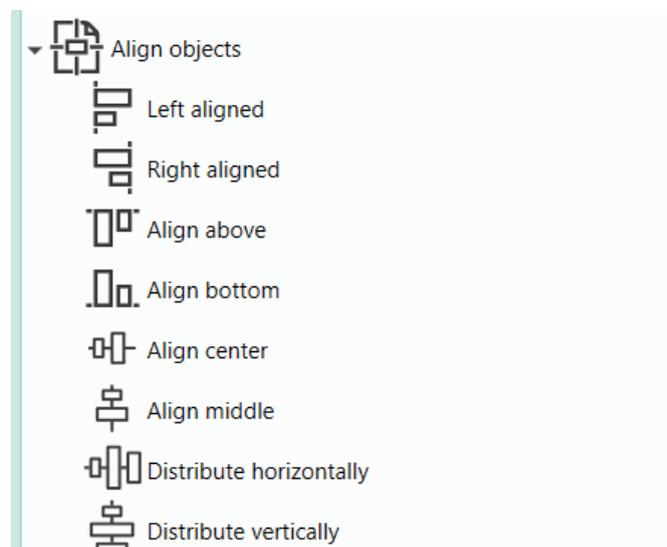


Figura 89. Opciones de alineación para los objetos en PLC-Lab.

Las acciones de esta sección permiten alinear los objetos actualmente seleccionados en el espacio de dibujo de una manera específica. Hay varias opciones disponibles. Al hacer clic en la acción que se desea, esta se aplica a los objetos seleccionados.

Si se desea obtener más información sobre el resto de acciones disponibles en PLC-Lab, consulte el siguiente enlace.

#### 4.1.3. PLC-Lab 3D Player

Si bien PLC-Lab crea escenas 2D, PLC-Lab 3D Player es una plataforma de simulación 3D diseñada para la educación y formación en sistemas automatizados dinámicos, ofreciendo una experiencia interactiva y práctica. Proporciona simulaciones realistas de estos sistemas, ofreciendo una interfaz interactiva y práctica. Este software es intuitivo y es compatible con diversos controladores y PLCs utilizados en el campo educativo y profesional [24].

En concreto, es compatible con los siguientes PLCs:

- ❖ Real Siemens-PLCs (S7-300/400/1200/1500/LOGO!).
- ❖ Real CODESYS V3 based PLCs (WAGO, Schneider, EATON, ...).
- ❖ Compatible with the built-in PLCSIM (S7-1200,S7-1500) of the TIA PORTAL V13-V18.
- ❖ PLCSIM of the Simatic-Manager V5.x.
- ❖ Soft PLCs PLCSIM-Advanced, CODESYS Runtime.
- ❖ Any PLC that supports OPC/UA.
- ❖ Allen Bradley PLCs Micro800, Micro800 Simulator, CompactLogix, ControlLogix.
- ❖ WinSPS-S7 V6.
- ❖ Grafcet-Studio

Requiere las siguientes especificaciones para el sistema:

- Sistema operativo: Windows 7 SP1 + o superior.
- Hardware: CPU: arquitectura X64 compatible con el conjunto de instrucciones SSE2.  
GPU: GPU compatible con DX10, DX11 y DX12.
- .NET Framework : 4.5.2 y 4.6.2.
- Idiomas soportados: Interfaz de usuario: alemán, inglés.  
Documentación: inglés, alemán.

PLC-Lab incluye algunos módulos que no incluye PLC-Lab y que se han considerado interesantes; Usando el Módulo de Etiquetas, se pueden monitorear todas las entradas y salidas. Además, es posible configurar las entradas y salidas con valores fijos, también conocido como forzado. Esta herramienta es muy útil a la hora de la resolución de problemas. El módulo piloto automático, que actúa como sistema de control para el modelo 3D, simula el comportamiento de un programa PLC. Esto permite comprender mejor la funcionalidad del modelo. El módulo de exportación de E/S permite exportar todas las variables de entrada y salida en varios formatos de archivo. Estos archivos pueden ser importados por el sistema de programación (por ejemplo, TIA PORTAL), eliminando la necesidad de introducir las variables manualmente.

Pese a todas las ventajas que ofrece PLC-Lab 3D Player sobre PLC-Lab y Factory IO, actualmente, la versión que hay disponible para este software en el mercado solamente permite trabajar con los modelos que ya vienen contruídos, por lo que no permite construir modelos desde cero como PLC-Lab.

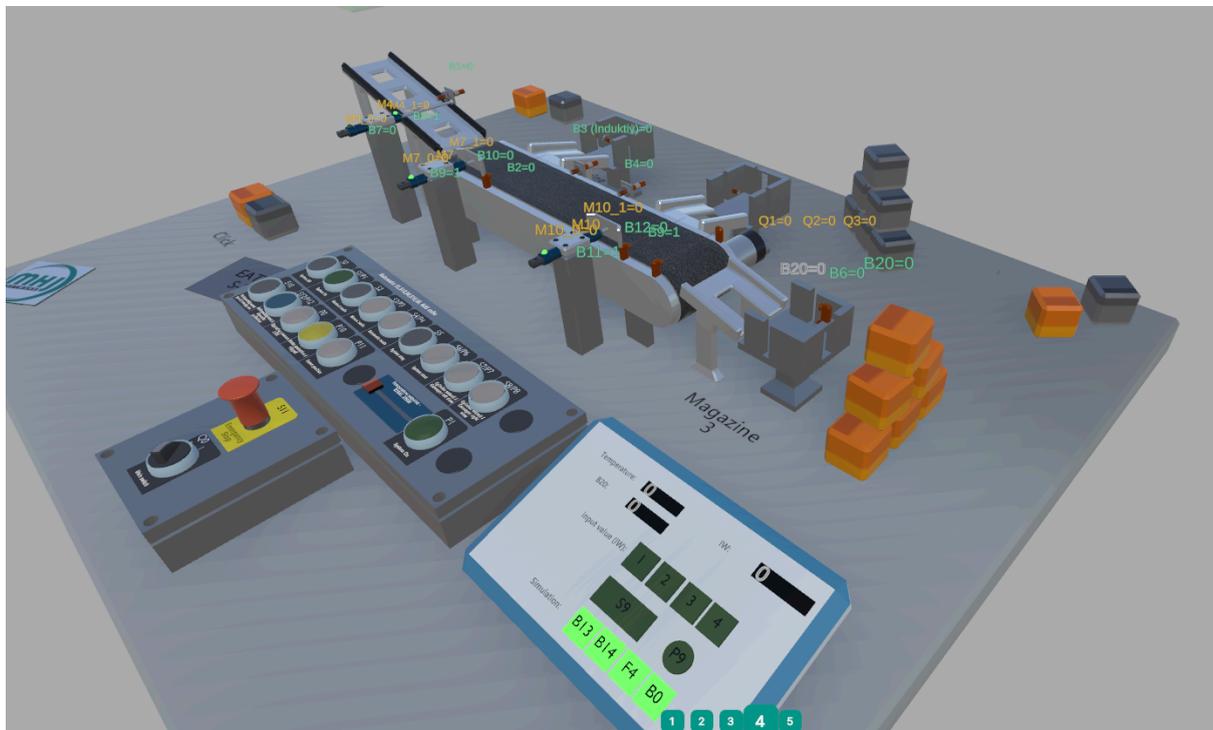


Figura 90. Ejemplo disponible de un modelo para una planta simulada en PLC-Lab 3D Player.

## 4.2. Programación en GRAFCET: Grafcet Studio

Grafcet-Studio es una herramienta de software empleada para la programación en Grafcet. Permite diseñar diagramas de forma eficiente para posteriormente probarlos utilizando un simulador integrado. Este simulador verifica el comportamiento esperado del diagrama, asegurando que las condiciones y secuencias definidas funcionen correctamente.

Para utilizar Grafcet Studio se necesita un sistema que cuente con alguno de los siguientes sistemas operativos instalados [25].

- Windows 7.
- Windows 8.1.
- Windows 10.
- Windows 11.

Requiere Microsoft .NET Framework versión 4.5.2 y 4.6.2 y unos 25 MB de espacio en el disco duro.

Se puede disponer de distintas licencias para el uso de este Software. En el caso de este trabajo se ha utilizado una licencia Grafcet Studio Pro Edition (100), que posee ciertas limitaciones que se muestran en la tabla 2.

**Tabla 2. Limitaciones de Grafcet- Studio Pro Edition (100).**

<b>Elementos de Grafcet</b>	<b>Límite</b>
Etapas	100
Transiciones	100
Acciones	110
Ramificación para cada etapa	10
Acciones para cada etapa	16
Etapas macro de salida	2

Además de las limitaciones mencionadas, se cuenta con otra limitación importante a la hora de dibujar el diagrama de Grafcet, y es que el espacio de

dibujo está restringido a una sola página. Cuando se agregan elementos que se encuentran fuera de esta área, Grafcet-Studio muestra una advertencia y dichos elementos no se cargan al PLC.

Los elementos de pantalla en Grafcet Studio son los siguientes:

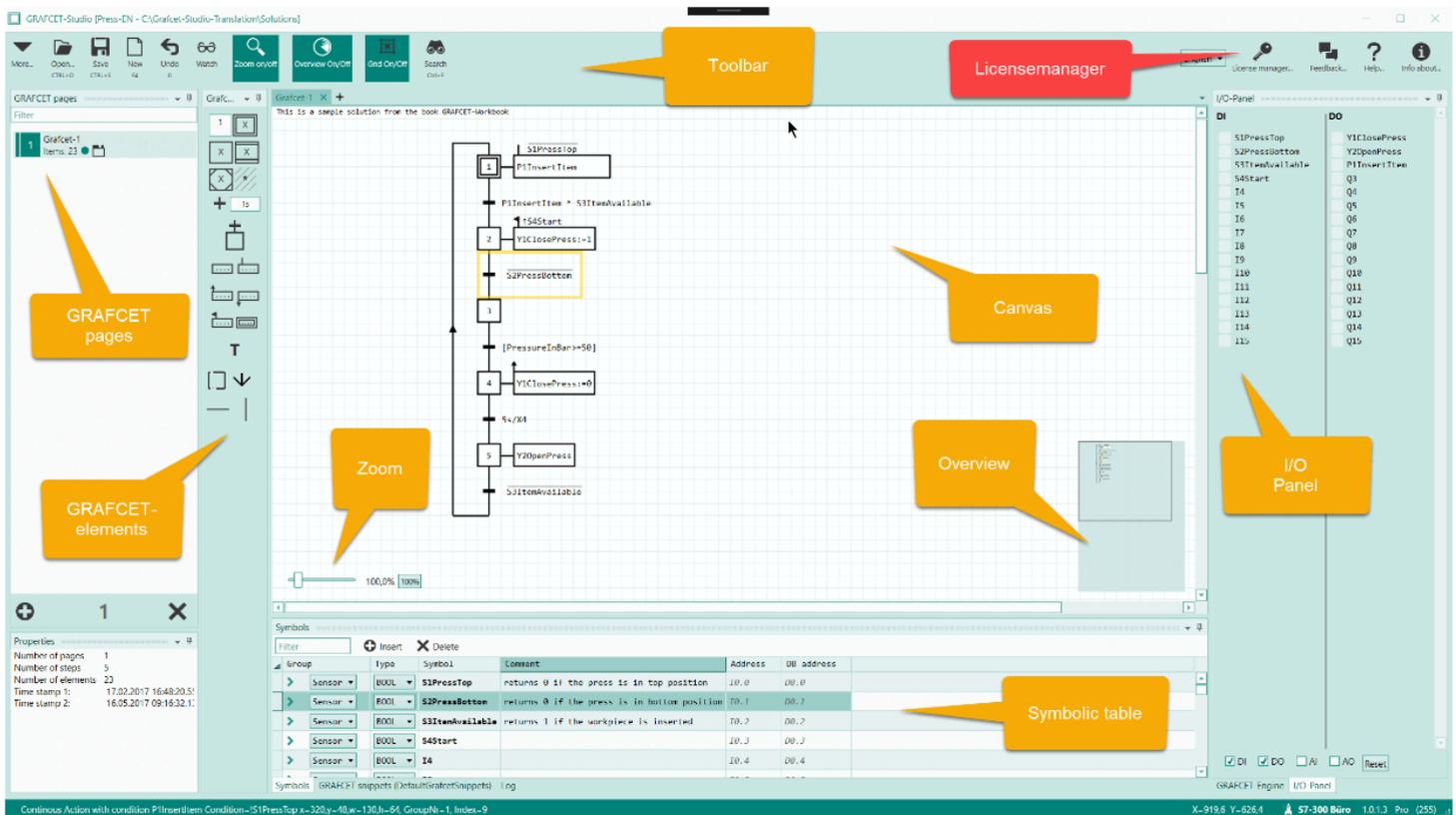


Figura 91. Diseño de pantalla de Grafcet-Studio [25].

- Barra de herramientas : Contiene los comandos clave de acceso rápido y se puede acceder a todos los elementos del menú haciendo clic en el icono (la flecha hacia abajo).
- Lienzo: Espacio destinado para el dibujo de los diagramas de Grafcet. En la parte inferior derecha se cuenta con una miniatura, por la que se puede navegar.
- Zoom: Control deslizante que permite acercar o alejar o el diagrama.
- Páginas de Grafcet: Se insertan o eliminan las páginas de Grafcet necesarias para su proyecto.
- Panel de E/S: Se muestran las entradas y salidas de la tabla de símbolos. Se puede emplear el ratón para manipular las entradas.

- Elementos Grafcet: Se encuentran los elementos de dibujo disponibles en Grafcet-Studio.
- Tabla simbólica: Se definen todos los operandos (entradas, salidas, datos) que se quieran utilizar en el Grafcet.
- Administrador de licencias.

Para la programación de la Guía GEMMA de las estaciones cero, uno y dos de la planta Festo con el Grafcet Studio se ha hecho uso de los elementos que se explicaron en el apartado 3.2: acciones, transiciones y acciones.

En Grafcet Studio se dispone de un tipo particular de etapas que se han empleado en este trabajo y que se describen a continuación:

- **Etapas [26].:**

Además de las etapas definidas en el apartado 3.2 de esta memoria, se debe conocer que hay disponibles otros tipos de etapas en Grafcet-Studio. A continuación se describen las que son empleadas en este proyecto.

- **Etapas incluyentes:**

Para utilizarlas, se debe iniciar la etapa inicial de cada Grafcet parcial utilizando un asterisco para activarla. Además, para asociar un grafcet parcial a una etapa incluyente, se debe colocar el número de la etapa incluyente en la parte superior del Grafcet parcial, como se muestra en la siguiente figura 92.

Para salir de una etapa incluyente basta que se active su transición de salida, sin importar en qué etapa se encuentre el Grafcet parcial.

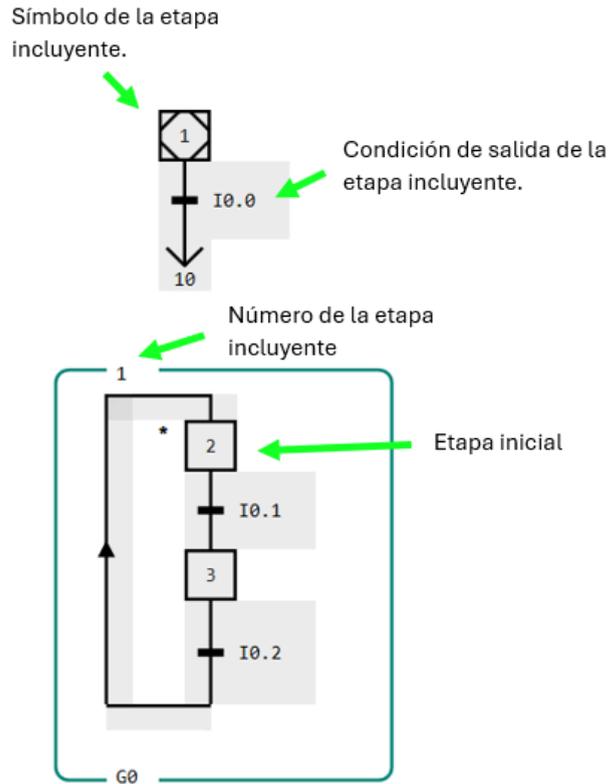


Figura 92. Ejemplo etapa incluyente en Grafcet-Studio.

- **Initial enclosing step:**

Se refiere a un paso que está asociado con uno o más subgrafcets, estos subgrafcets contienen al menos, un estado inicial. Esencialmente, el estado configurado como paso inicial de cierre, marca el comienzo de una secuencia de pasos dentro de un área definida del Grafcet, asegurando que las operaciones dentro de ese recinto se inicien adecuadamente en el orden especificado.

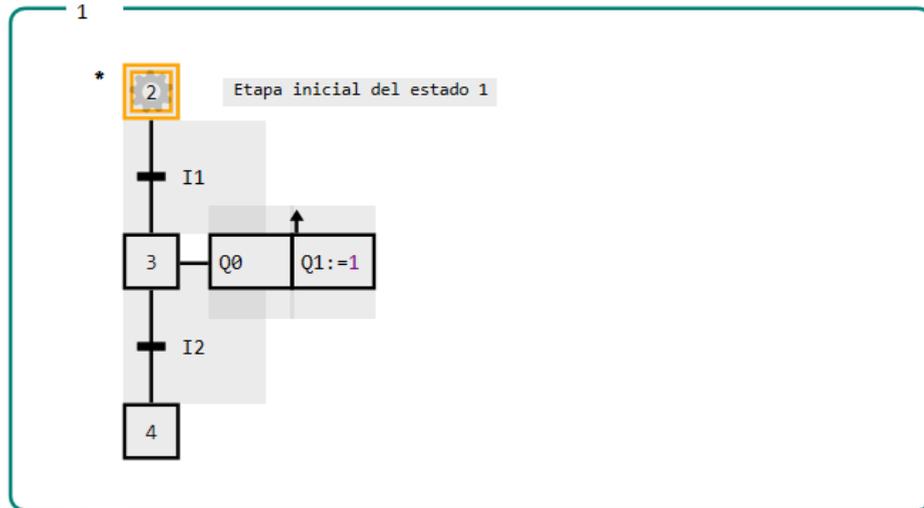
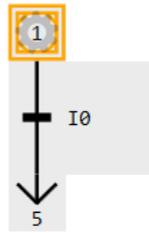


Figura 93. Ejemplo diagrama con un paso inicial de cierre en Grafcet-Studio.

Antes de comenzar con una descripción detallada sobre la integración de Factory IO con el Grafcet Estudio, es importante destacar que, hay algunas características de Grafcet las cuales aún no son compatibles con Grafcet-Studio.

- Forzado de un Grafcet parcial a la situación actual (comando de congelación).
- Borde de un término entre paréntesis. Ejemplo:  $\uparrow(I0*11)$ . Los bordes de los operandos son posibles:  $\uparrow I0$ .
- Acción limitada por tiempo: Esto no está soportado  $!(6s/X28)$  en una acción continua con condición.

## 5. Integración del Factory I/O con el Grafcet Studio para la implementación de la Guía GEMMA en las estaciones 0-2 de la planta Festo.

La combinación de Grafcet-Studio y Factory IO para la implementación de la Guía GEMMA brinda la ventaja de poder trabajar con los modelos simulados de la planta FESTO sin tener que interactuar con la planta real. Esto significa que se puede evitar la necesidad de conectar Grafcet Studio con el PLC que se fuese a emplear.

Grafcet-Studio y Factory IO deben comunicarse entre sí de alguna manera. Como se explicó en el apartado 4.1.1 de esta memoria, en Factory IO se puede elegir el controlador para las entradas y salidas de la planta creada. En este caso, para que se logre esta comunicación se debe elegir el controlador MHJ.

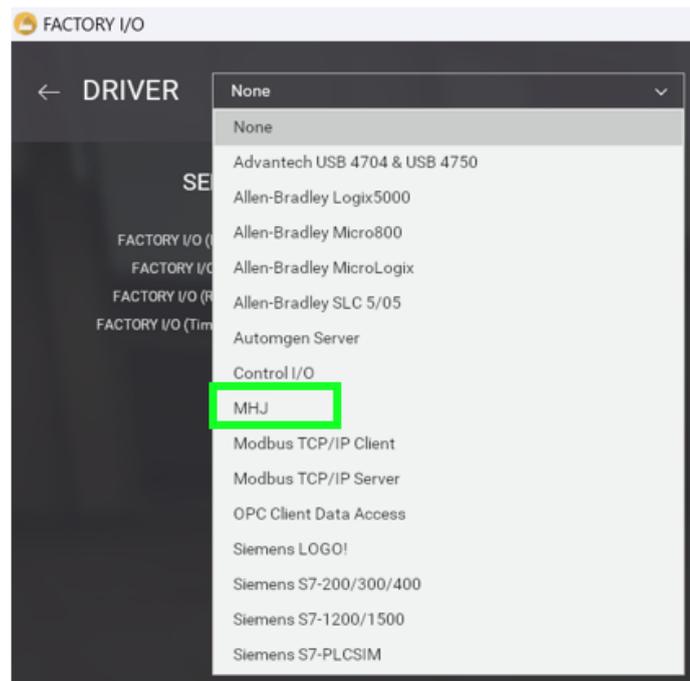


Figura 94. Listado de los controladores disponibles en la versión Ultimate de Factory IO.

Y asignar las entradas y salidas con las mismas direcciones con las que figuran en Grafcet-Studio en la tabla de símbolos. Esto se ilustra a continuación en la siguiente figura:

The image displays two software interfaces side-by-side. The left interface, Grafset-Studio, shows a table of symbols with the following columns: Grupo, Tipo, Icono, Coment, and Direcc. The right interface, Factory IO, shows a configuration window for 'DRIVER' with sections for SENSORS and ACTUATORS, listing various inputs and outputs with their addresses.

Grupo	Tipo	Icono	Coment	Direcc
Sensor	BOOL	BotonVentosa		I2.2
Sensor	BOOL	PE		I2.3
Sensor	BOOL	BHayPieza	Botón	I2.4
Sensor	BOOL	AlmacenVacio		I2.5
Sensor	BOOL	Fallo		I2.6
Sensor	BOOL	HayPieza		I2.7
Sensor	BOOL	BrazoPosOrigen		I3.0
Sensor	BOOL	BotonPiezaProcesada		I3.1
Sensor	BOOL	BotonTest		I3.2
Accionador	BOOL	SalidaCL	Salida	Q0.0
Accionador	BOOL	SalidaDeCajas		Q0.1
Accionador	BOOL	StartL		Q0.2
Accionador	BOOL	StopL		Q0.3
Accionador	BOOL	MovX		Q0.4
Accionador	BOOL	MovZ		Q0.5
Accionador	BOOL	VentosaBrazo		Q0.6
Accionador	BOOL	Empujar		Q0.7
Accionador	BOOL	LPiezaCorrecta		Q1.0
Accionador	BOOL	LEmpujar		Q1.1
Accionador	BOOL	LFallo		Q1.2

Figura 95. Ejemplo de entradas y salidas asignadas POS con el mismo operando en Grafset-Studio y Factory IO.

Por lo tanto, una vez el controlador está configurado en Factory IO, lo único que es necesario para que se produzca la comunicación es ejecutar ambas simulaciones al mismo tiempo.

Una vez se ha comprendido el funcionamiento Grafset Studio y de Factory IO, tanto por separado como en conjunto, se puede proceder al diseño e implementación de la Guía GEMMA para las estaciones cero, uno y dos de la planta Festo. Esto nos permitirá establecer un esquema detallado para la automatización de los procesos en estas estaciones, garantizando que cada una de ellas opere de manera eficiente. El diseño de la Guía GEMMA incluye la definición de los estados necesarios para cada estación, así como la definición de los sensores y actuadores previamente configurados en el entorno de simulación.

## 5.1. Diseño de la Guía GEMMA de la estación cero

En este apartado, se describe el proceso de diseño e implementación de la Guía GEMMA en la estación Cero de la Planta Festo.

### 5.1.1 Sensores y Actuadores definidos en la estación cero

En las tablas 3 y 4 se presentan, simultáneamente, el listado de sensores y actuadores de la estación cero, incluyendo los necesarios para poder implementar los estados de la Guía GEMMA que se ha decidido incluir en el diseño.

**Tabla 3. Sensores de la estación 0.**

Icono	Tipo	Comentario	Dirección	Descripción
Start	BOOL	-	I0.0	Botón de Start.
Stop	BOOL	-	I0.1	Botón de Stop.
BotonSalidaCajas	BOOL	-	I0.2	Botón para que se produzca la salida de cajas.
EmpujarContraido	BOOL	-	I0.3	Sensor que indica que el pistón se encuentra contraído.
EmpujarExtendido	BOOL	-	I0.4	Sensor que indica que el pistón se encuentra retraído.
MovingX	BOOL	-	I0.5	Sensor que indica que se está ejecutando el desplazamiento en el eje X.
MovingZ	BOOL	-	I0.6	Sensor que indica que

				se está ejecutando el desplazamiento en el eje Z.
DetectaPiezas	BOOL	-	I0.7	Sensor incorporado en el brazo que detecta una pieza.
Auto	BOOL	-	I1.0	Modo del selector para el funcionamiento automático.
Manual	BOOL	-	I1.1	Modo del selector para el funcionamiento Manual .
DetectaPieza	BOOL	-	I1.2	Sensor que detecta una pieza al final de la plataforma.
FinCiclo	BOOL	-	I1.3	Botón utilizado para indicar que es el fin del ciclo.
FalloVB	BOOL	Fallo Ventosa Brazo	I1.4	Modo del selector para indicar que el fallo se encuentra en la ventosa o el brazo.
FalloPiston	BOOL	-	I1.5	Modo del selector para indicar que el fallo se encuentra en el pistón.
MoverBrazoX	BOOL	-	I1.6	Botón para realizar el movimiento del brazo en el eje X de forma manual.

MoverBrazoZ	BOOL	Mover Brazo eje Z.	I1.7	Botón para realizar el movimiento del brazo en el eje Z de forma manual.
BotonRetraer	BOOL	-	I2.0	Botón para retraer el pistón de forma manual
BotonEmpujar	BOOL	-	I2.1	Botón para empujar el pistón de forma manual.
BotonVentosa	BOOL	-	I2.2	Botón para activar/desactivar la ventosa del brazo de forma manual.
PE	BOOL	Parada de Emergencia.	I2.3	Botón para la Parada de Emergencia.
BHayPieza	BOOL	Botón Hay Pieza.	I2.4	Botón para indicar que hay ( o no) una pieza en la plataforma. 1: Hay pieza en la plataforma que debe retirarse. 0: No hay ninguna pieza en la plataforma.
AlmacenVacio	BOOL	-	I2.5	Botón que indica que se ha vaciado el almacén .
Fallo	BOOL	-	I2.6	Botón para indicar que se tiene un fallo.
HayPieza	BOOL	-	I2.7	Sensor que indica la presencia de una pieza en la bandeja.
BrazoPosOrigen	BOOL	Botón	I3.0	Sensor que indica que

		Posición Origen.		el brazo se encuentra en la posición origen (extendido en el eje X).
BotonPiezaProcesada	BOOL	-	I3.1	Botón que utiliza el operario, para indicar que la pieza ha sido procesada.
BotonTest	BOOL	-	I3.2	Botón empleado para realizar la marcha de test.

**Tabla 4. Actuadores de la estación 0.**

<b>Icono</b>	<b>Tipo</b>	<b>Comentario</b>	<b>Dirección</b>	<b>Descripción</b>
Salida Cajas Luz	BOOL	-	Q0.0	Luz del botón para la salida de cajas.
Salida de Cajas	BOOL	-	Q0.1	Actuador que provoca la salida de las cajas.
StartL	BOOL	Start Luz	Q0.2	Luz del botón Start.
StopL	BOOL	Stop Luz	Q0.	Luz del botón Stop.
MovX	BOOL	Movimiento en X	Q0.4	Actuador que provoca el movimiento del brazo en el eje X.
MovZ	BOOL	Movimiento en Z	Q0.5	Actuador que provoca el movimiento del brazo en el eje X.

Ventosa Brazo	BOOL	-	Q0.6	Actuador que acciona la ventosa del brazo.
Empujar	BOOL	-	Q0.7	Actuador para empujar/retraer el pistón. 1: Empujar 0:Retraer
LPiezaCorrecta	BOOL	Luz Pieza Correcta	Q1.0	Luz que indica que se tiene una pieza correcta.
LEmpujar	BOOL	Luz Empujar	Q1.1	Luz que indica que se debe pulsar el botón empujar.
LFallo	BOOL	Luz Fallo	Q1.2	Luz que indica que hay un fallo.
LFinCiclo	BOOL	Luz Fin Ciclo	Q1.3	Luz que indica que se está en el fin de ciclo.
LVentosa	BOOL	Luz Ventosa	Q1.4	Luz que indica que se debe pulsar el botón ventosa.
MoverBrazoXL	BOOL	Mover Brazo en X Luz	Q1.5	Luz que indica que se debe pulsar el botón para el movimiento del brazo en X.
MoverBrazoZL	BOOL	Mover Brazo en Z Luz	Q1.6	Luz que indica que se debe pulsar el botón para el movimiento del brazo en X.

RetraerL	BOOL	Retraer Luz	Q1.7	Luz que indica que se debe pulsar el botón para retraer el pistón.
LE	BOOL	Luz Emergencia	Q2.0	Luz que indica que se encuentra en la parada de emergencia.
LHayPieza	BOOL	Luz Hay Pieza	Q2.1	Luz que indica que hay una pieza en la plataforma.
LAlmacenVacio	BOOL	Luz Almacén Vacío	Q2.2	Luz para indicar que se ha vaciado el almacén.
LFalloVB	BOOL	Luz Fallo Ventosa/Brazo	Q2.3	Luz para indicar que se tiene un fallo en la Ventosa o el Brazo
LFalloPiston	BOOL	Luz Fallo Pistón	Q2.4	Luz para indicar que se tiene un fallo en el Pistón
LBRetraer	BOOL	Luz Botón Retraer	Q2.5	Luz para indicar que se debe pulsar el botón "Retraer".
BotonTestL	BOOL	Botón Test Luz	Q2.6	Luz para indicar que se encuentra en el estado "Marcha de Test"

### 5.1.2 Estados implementados de la Guía GEMMA.

A continuación se indican los estados de la guía que se ha decidido implementar para la estación cero.

**A1- Parada en el estado inicial:** Se accede una vez se haya pasado por el estado A6 o A2. Desde este estado se espera a que el operario escoja un modo de operación: si se escoge en modo manual se accede al estado F4, si escoge el modo automático se accede a F1, si pulsa el botón de test se accede al estado F6, y en caso de que pulse la parada de emergencia irá al estado D1.

**A2- Parada solicitada al final del ciclo:** Este estado es empleado para parar todo lo que sea necesario una vez se quiera finalizar el ciclo.

**A6- Puesta del sistema en el estado inicial:** Implementado para cumplir con las condiciones iniciales de la estación 0. En este estado, se apagan las luces necesarias, se desactiva la ventosa del brazo y se verifica que el brazo esté en la posición origen (extendido en el eje X). En caso de que no lo esté deberá colocarse. Una vez el brazo esté en la posición de origen, se verifica que no se cuente con la presencia de una pieza sobre la plataforma antes de ir al estado A1. En caso de que la hubiera, se activará la “luz Hay Pieza” y será el operario quien la retire. Una vez retirada la pieza, el operario deberá pulsar el botón “Hay pieza” para indicar que se ha retirado, y se podrá acceder al estado A1.

**F1- Producción normal:** Se llevan a cabo las acciones correspondientes a la producción normal en la estación 0. Comienza al pulsar el botón “Start” y seleccionando el “Modo auto”. La “Luz Salida de Cajas” ( ‘Cajas’ representan las piezas que son procesadas en la planta Festo) se encenderá, indicando que se debe pulsar el “Botón Salida de Cajas” para que salga la primera pieza. Una vez fuera, el pistón la empuja por la plataforma hasta que se active el sensor “Detecta pieza” (I1.2) ( el cual indica que hay una pieza al final de la misma). El brazo bajará y cuando el sensor del brazo “Detecta Piezas” detecte la pieza, se activará la ventosa, el brazo sube, se retrae en el eje X , bajará de nuevo y se desactivará la ventosa para llevar la pieza a la estación 1. Una vez terminado, este proceso se repite automáticamente sin necesidad de volver a pulsar el “Botón Salida de Cajas”.

**F3- Marcha de cierre:** Permite al operario vaciar el almacén en cualquier momento. Esto puede ser útil si, por ejemplo, se detecta que las piezas que están saliendo no son del tipo correcto. Para ello, se utilizará el botón “Stop”

para entrar en este estado y el botón “Salida de Cajas” para ir sacándolas una a una. Será el operario quien se encargue de retirarlas de la plataforma. Este proceso se repite hasta que se pulse el botón “Almacén Vacío” momento en el cual se evolucionará al estado A6.

**F6- Marcha de test:** Se accede pulsando “Botón Test”. En él se realiza un ciclo de producción para verificar que todo funciona correctamente. Se utilizan temporizadores para evaluar el funcionamiento. Por ejemplo, si una acción específica que suele realizarse en 1 segundo pero transcurren 5 segundos y la condición de transición esperada no se cumple, se identifica un fallo. Si algún elemento de la planta falla, se encenderá la luz de fallo correspondiente, pero se continuarán comprobando los demás elementos. Si la luz de fallo no se activa implica que todo funciona correctamente, por tanto, al finalizar la comprobación, se evoluciona al estado A6. Sin embargo, en el caso de un fallo, se deja la decisión al operario para decidir a qué estado quiere ir (D3 o F4).

**F4- Marcha de verificación sin orden:** En este estado se evalúan el comportamiento del brazo y del pistón. Para seleccionar cual de ellos se desea mover, se utilizan los botones “MoverBrazoX”, “MoverBrazoZ”, “BotonVentosa”, “BotonRetraer” y “BotonEmpujar”.

Si se elige probar el brazo, se debe realizar una comprobación completa de su funcionamiento. Para ello debe haber una pieza presente en la plataforma y el operario se encargará de mover el brazo y activar la ventosa.

Si se desea evaluar el pistón, primero se debe retirar cualquier pieza que haya en la posición inicial de la plataforma (esto lo indica el sensor “HayPieza”). Una vez no se cuente con pieza en la plataforma, se emplean los botones “BotonRetraer” y “BotonEMpujar” para realizar los movimientos de extensión y retracción del cilindro.

Después de comprobar uno de estos de los actuadores, se puede optar por comprobar el otro. Si todo ha funcionado correctamente, se puede regresar a las condiciones iniciales. En caso de que hubiese sucedido algún fallo, se puede evolucionar al estado D3 y continuar la producción a pesar de los defectos.

**D1- Parada de Emergencia:** La parada de emergencia detiene todas las operaciones de la estación. Al desenclavar la seta de emergencia, se regresa al estado A6.

**D3- Producción a pesar de los efectos:** Este estado contempla posibles fallos en el pistón y en el brazo o ventosa y permite que la estación siga funcionando en presencia de dichos fallos. Los dos últimos se han agrupado ya que, en ambos casos, el operario deberá llevar manualmente la pieza a la estación 1. Como el simulador no falla por sí solo, se debe “provocar” el fallo usando “Botón Fallo” del panel de control, conjuntamente con un selector que permite al operario elegir el tipo de fallo que se desea simular.

- Fallo en el pistón: La pieza sale del almacén, y deberá ser el operario quien la coloque manualmente al final de la plataforma. Cuando se activa el sensor "Detecta pieza" , el ciclo continúa normalmente.
- Fallo en la ventosa o el brazo: La pieza sale del almacén y el pistón la empuja hasta que se active el sensor "Detecta pieza" . Será el operario quien se encargue de llevarla manualmente a la estación 1.

### 5.1.3 Estados NO implementados de la Guía GEMMA

En este apartado se reflejan los estados que no han sido incluidos en el diseño de la Guía GEMMA de la estación 0, los cuales han sido excluidos por la falta de necesidad o de interés para esta estación.

**A3- Parada en un estado determinado:** Debido a la poca complejidad de la estación no se ha implementado.

**A4- Parada en un estado intermedio:** Consiste en detener el proceso en un instante diferente al de inicio o final del ciclo. Al igual que en el estado anterior no se ha implementado debido a la poca envergadura del proceso.

**A5- Preparación para la puesta en marcha después de un defecto :** No se ha implementado ya que no se especifican acciones específicas de preparación después de un fallo en la estación.

**A7- Puesta del sistema en un estado determinado:** No es necesario ya que no se utiliza el estado A4.

**F2- Marcha de preparación:** En esta estación no es necesario preparar la máquina antes de entrar a producción.

**F5- Marcha de verificación con orden:** Debido a que ya se realiza la verificación sin orden en el estado F4, no se ha visto necesario implementar este estado.

**D2- Diagnóstico y/o tratamiento de fallos:** No se implementa debido a la poca complejidad del proceso.

#### **5.1.4 Pupitre de control de la estación 0**

En la figura 96 se muestra el aspecto del pupitre de control que se ha diseñado para la estación cero.



Figura 96. Pupitre de control diseñado en Factory IO para la estación 0.

### 5.1.5 Guía GEMMA de la estación cero

En la figura 97 se representa la guía GEMMA implementada para la estación cero. En ella se observan los estados incluidos, las transiciones permitidas entre ellos así como las condiciones de evolución, cuando las hay.

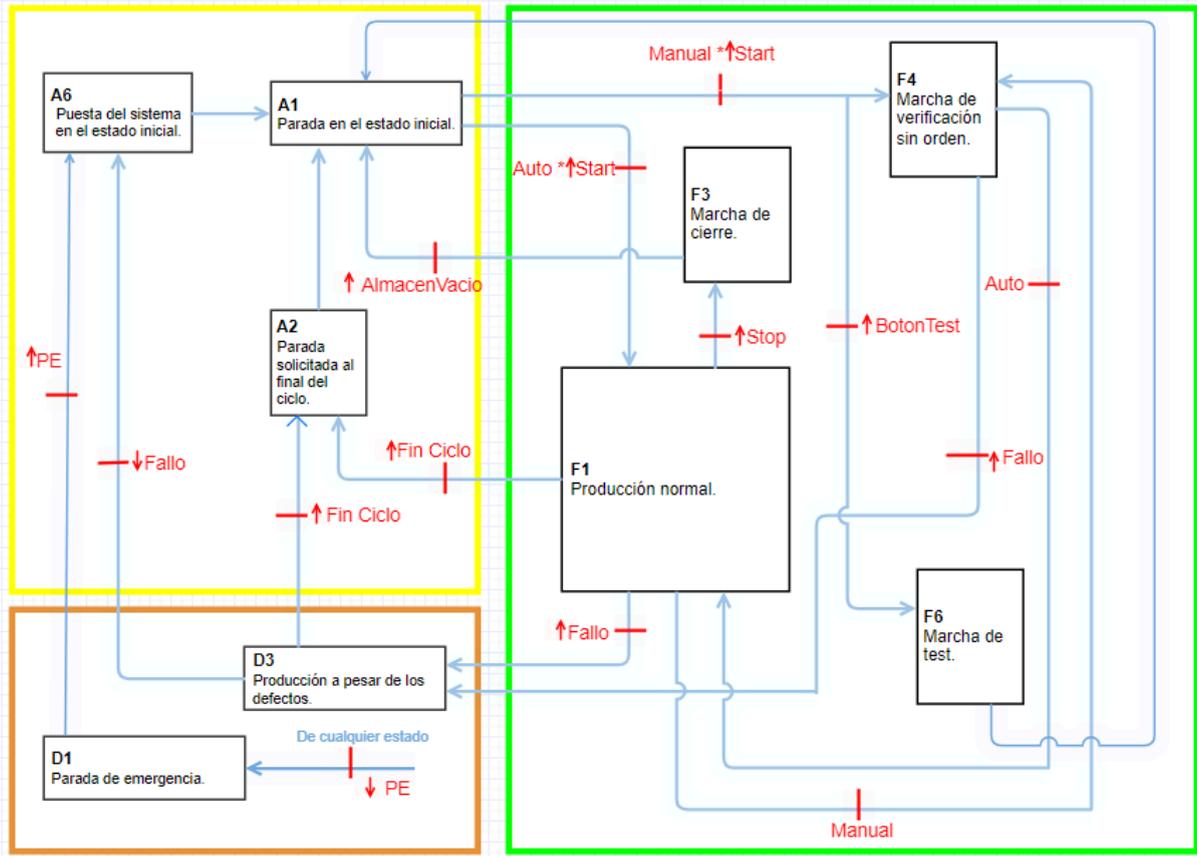


Figura 97. Guía GEMMA diseñada para la estación cero de la planta Festo.

## 5.2. Diseño de la Guía GEMMA de la estación uno

En este apartado, se describe el proceso de diseño e implementación de la Guía GEMMA en la estación Uno de la planta Festo.

### 5.2.1 Sensores y Actuadores definidos en la estación uno

En las tablas 5, 6 y 7 se presentan, simultáneamente, el listado de sensores, actuadores y variables de datos de la estación uno, incluyendo los necesarios para poder implementar los estados de la Guía GEMMA que se ha decidido incluir en el diseño.

**Tabla 5. Sensores de la estación 1.**

<b>Icono</b>	<b>Tipo</b>	<b>Comentario</b>	<b>Dirección</b>	<b>Descripción</b>
Start	BOOL	-	I0.0	Botón de Start.
Stop	BOOL	-	I0.1	Botón de Stop.
1Contraído	BOOL	-	I0.2	Sensor que indica que el pistón 1 se encuentra contraído.
1Extendido	BOOL	-	I0.3	Sensor que indica que el pistón 1 se encuentra extendido.
2Contraído	BOOL	-	I0.4	Sensor que indica que el pistón 2 se encuentra contraído.
2Extendido	BOOL	-	I0.5	Sensor que indica que el pistón 2 se encuentra extendido.
Final1	BOOL	-	I0.6	Sensor que indica que la plataforma se encuentra abajo.
Final2	BOOL	-	I0.7	Sensor que indica que la plataforma se encuentra arriba..
LlegaCaja	BOOL	-	I1.0	Botón para que llegue una pieza.

PidenCaja	BOOL	-	I1.1	Botón para simular que la pieza puede ir a la estación 2.
Auto	BOOL	-	I1.4	Modo del selector para el funcionamiento automático.
Manual	BOOL	-	I1.5	Modo del selector para el funcionamiento Manual .
PiezaBuenaBoton	BOOL	-	I1.6	Botón para indicar que se trata de una pieza buena.
PiezaMalaBoton	BOOL	-	I1.7	Botón para indicar que se trata de una pieza mala.
BotonSubir	BOOL	-	I2.0	Botón para subir la plataforma elevadora.
BotonBajar	BOOL	-	I2.1	Botón para bajar la plataforma elevadora.
BotonER1	BOOL	Botón Empujar Retraer 1.	I2.2	Botón para empujar y retraer el pistón 1.
BotonER2	BOOL	Botón Empujar Retraer 1.	I2.3	Botón para empujar y retraer el pistón 2.
PE	BOOL	Parada de	I2.4	Botón para la Parada

		Emergencia.		de Emergencia.
Fallo	BOOL	-	I2.5	Botón para indicar que se tiene un fallo.
FalloSensor	BOOL	-	I2.6	Modo del selector para indicar fallo en el sensor de medida.
FalloElevador	BOOL	-	I2.7	Modo del selector para indicar fallo en el elevador.
FalloPiston1	BOOL	-	I3.0	Modo del selector para indicar fallo en el pistón 1.
FalloPiston2	BOOL	-	I3.1	Modo del selector para indicar fallo en el pistón 2.
PiezaProcesada	BOOL	-	I3.2	Botón para indicar que se ha procesado la pieza.
BotonTest	BOOL	-	I3.3	Botón empleado para realizar la marcha de test.
PiezaBuena	BOOL	-	I3.4	Indica la presencia de una pieza buena esperando en la rampa 2 a ser pedida por la estación 2.
NoHayPieza	BOOL	-	I3.5	Botón para indicar que no hay pieza esperando a ser llevada a la estación

				2.
SensorMedida	UNIT16	-	I256-I257	Indica el valor medido de la pieza.

**Tabla 6. Actuadores de la estación 1.**

Icono	Tipo	Comentario	Dirección	Descripción
LuzStart	BOOL	-	Q0.0	Luz del botón Start.
LuzStop	BOOL	-	Q0.1	Luz del botón Stop.
Subir	BOOL	-	Q0.2	Botón para subir la plataforma elevadora.
Bajar	BOOL	-	Q0.	Botón para bajar la plataforma elevadora.
SalidaCajas	BOOL	-	Q0.4	Elemento creador en Factory IO para la simulación de salida de las "piezas".
Empujar1	BOOL	-	Q0.5	Pistón 1.
Empujar2	BOOL	-	Q0.6	Pistón 2.
LuzPidenCaja	BOOL	-	Q0.7	Luz del botón "Piden Caja".
LuzLlegaCaja	BOOL	-	Q1.0	Luz del botón "Llega Caja".
ParadaCajas	BOOL	-	Q1.1	Pestaña de sujeción

				para que las piezas no vayan a la estación 2.
LPBuena	BOOL	Luz Pieza Buena.	Q1.2	Luz que indica que se trata de una pieza buena.
LPMala	BOOL	Luz Pieza Mala.	Q1.3	Luz que indica que se trata de una pieza defectuosa.
LuzSubir	BOOL	-	Q1.4	Luz del botón "Subir".
LuzBajar	BOOL	-	Q1.5	Luz del botón "Bajar".
LuzBER1	BOOL	Luz Botón Empujar Retraer 1.	Q1.6	Luz del botón "Empujar Retraer 1".
LuzBER2	BOOL	Luz Botón Empujar Retraer 2.	Q1.7	Luz del botón "Empujar Retraer 2".
LBFallo	BOOL	Luz Botón Fallo	Q2.0	Luz del botón "Fallo".
LBTest	BOOL	Luz Botón Test	Q2.1	Luz del "Botón Test".
LE	BOOL	Luz Emergencia	Q2.2	Luz de emergencia..
LuzPiezaProcesada	BOOL	-	Q2.3	Luz del botón "Pieza procesada".
MedirAnalogica	UNIT16	-	Q256	Actuador necesario para que el sensor de

				medida baja.
NumeroPB	INT16	Número Piezas buenas	Q257-Q258	Contador utilizado para saber el número de piezas buenas que se tiene en ese momento en la estación 1.

**Tabla 7. Variables de datos de la estación 1.**

Icono	Tipo	Dirección	Descripción
MedidaGuardada	UNIT16	-	Variable utilizada para guardar la medida que se tome.
MedidaTomada	BOOL	-	1: Indica que se ha tomado la medida. 0: Significa que no se ha medido.
PiezaNormal	BOOL	-	1: Se ha medido una pieza buena. 0: No se ha medido una pieza buena.
PiezaDefectuosa	BOOL	-	1: Se ha medido una pieza defectuosa. 0: No se ha medido una pieza defectuosa.

### 5.2.2 Estados implementados de la Guía GEMMA.

A continuación se indican los estados de la guía que se han decidido implementar para la estación Uno.

**A1- Parada en el estado inicial:** Se accede una vez se haya pasado por el estado A6 o A2. Desde este estado se espera a que el operario escoja un modo de operación: si se escoge en modo manual y se pulsa Start se accede al estado F4, si se escoge el modo automático y se pulsa Start se accede a F1, si se pulsa el botón de test se accede al estado F6, y en caso de que se pulse la parada de emergencia se irá al estado D1.

**A2- Parada solicitada al final del ciclo:** Este estado es empleado para parar todo lo que sea necesario una vez se quiera finalizar el ciclo.

**A6- Puesta del sistema en el estado inicial:** Implementado para cumplir con las condiciones iniciales de la estación 1. Esto implica verificar que la plataforma elevadora esté abajo (en caso de no estarlo se debe bajar) y que los pistones, la pestaña de sujeción en la rampa 2 y el sensor de longitud estén retraídos. Además, se restablece a 0 el valor almacenado en la variable de datos “medida tomada”.

**F1- Producción normal:** Se llevan a cabo las acciones correspondientes a la producción normal en la estación 1. Se comienza al pulsar el botón “Llega Caja” mediante el cual llega la primera pieza a la plataforma. Se debe tener en cuenta que en el Factory IO se puede hacer que lleguen piezas de distintos tamaños de forma aleatoria. Una vez la pieza llega, la plataforma elevadora sube y se detiene cuando se active el sensor “Final 2”. A continuación, el sensor de medida baja, mide la pieza, y el valor de esta medida se guarda en la variable “Medida Guardada”. Finalmente se sube el sensor de medida. Se pueden dar dos casos:

- Es una pieza buena, es decir, tiene la altura correcta:  
Para ello la variable de datos “Medida Guardada” debe ser  $< 25000$ . El pistón 2 se encarga de empujar la pieza hacia la rampa 2 y luego se retrae. La pestaña de sujeción de la rampa 2 impide que la pieza avance hasta la estación 2 hasta que se pulse el botón “Piden Caja”, que simula la petición de pieza por parte de la estación 2. La plataforma bajará y se seguirá produciendo con normalidad hasta que llegue de nuevo una pieza buena. Si se da esta situación, habrá que esperar a que se pulse el botón “Piden Caja” para que la pieza que estaba en espera se deslice hacia la estación 2. Al mismo tiempo, la pieza que se encuentra en la plataforma es empujada por el

pistón 2 hacia la rampa 2, donde la pestaña de sujeción se extenderá para retenerla hasta que sea solicitada por la estación 2, siguiendo el procedimiento anterior.

- La pieza es defectuosa, es decir, no tiene la altura correcta:  
En este caso la variable de datos “Medida Guardada” debe tener un valor >25000. La plataforma elevadora bajará hasta que se active “Final 1”, seguidamente el pistón 1 empujará la pieza hacia la rampa 1 para “desecharla” y se retrae.

En ambos casos, una vez se termine el ciclo, la producción continua de forma automática sin necesidad de pulsar el botón “Llega Caja”.

**F6- Marcha de test:** Se accede pulsando “Botón Test”. En él se realiza un ciclo de producción para verificar que todo funciona correctamente. Como ya se comentó en la estación 0, se utilizan temporizadores para evaluar el funcionamiento. Si algún elemento de la planta falla, se encenderá la luz de fallo correspondiente, pero se continuará comprobando los demás elementos. Si no se enciende la luz de fallo esto indica que todo funciona correctamente, por tanto, al finalizar la comprobación, se evolucionará al estado A6. Sin embargo, en el caso de un fallo, se deja la decisión al operario para decidir a qué estado quiere ir.

**F4- Marcha de verificación sin orden:** Este estado corresponde al modo manual. Para su implementación se han añadido los botones correspondientes al movimiento de la plataforma elevadora (I2.0 e I2.1) y los pistones: pistón 1 (I2.2) y pistón 2 (I2.3). Esto permite al operario realizar pruebas y verificar individualmente el funcionamiento de cada actuador. Si el operario verifica que todo funciona correctamente, puede optar por seguir realizando pruebas o avanzar al estado A6 para continuar con el proceso normal de producción.

En caso de que el operario detecte fallos en alguna parte del proceso por inspección visual, puede evolucionar al estado D3 pulsando el botón “Fallo” para poder seguir produciendo a pesar de los defectos.

**D1- Parada de Emergencia:** La parada de emergencia detiene todas las operaciones de la estación. Al quitar el pulsador de emergencia, se regresa al estado A6.

**D3- Producción a pesar de los efectos:** En esta estación se contempla la posibilidad de que pueda fallar el sensor de medida o el elevador, y que puedan fallar los pistones. Este estado permite que la estación siga funcionando en presencia de fallos. A la hora de provocar dichos fallos, lo más conveniente sería incorporar un selector con cuatro posiciones para poder escoger el tipo de fallo, sin embargo, como se mencionó anteriormente, debido a las limitaciones de Factory IO esto no es posible, ya que los selectores cuentan solo con dos posiciones. Por tanto, en un selector se tienen los fallos del sensor y el elevador y en otro los del pistón 1 o 2, lo que significa que siempre va a estar implementado el fallo en alguno de los pistones.



Figura 98. Selectores creados en Factory IO para escoger el tipo de fallo en la estación 1.

Por lo tanto, hay varias opciones de fallo:

- Fallo del sensor de medición y el pistón 1: Cuando la pieza llegue a la plataforma y esta se eleve hasta activar el sensor “final 1”, el sensor de medida no se activará (no bajará a medir) por lo que será el operario, mediante la botonera diseñada, quien determine si se trata de una pieza buena o defectuosa. Si se trata de una pieza buena, el pistón 2 empujará la pieza hacia la rampa 2 y se retraerá. Sin embargo, si se trata de una pieza defectuosa, la plataforma bajará, pero el pistón 1 no funciona, por lo que será el operario de nuevo quien se encargue de desechar la pieza manualmente.

- Fallo del sensor de medición y el pistón 2: En este caso, se sigue un procedimiento similar al caso anterior. Sin embargo, si la pieza es designada como buena por el operario, éste deberá llevarla a la rampa 2, donde esperará a ser solicitada por la estación 2. Si la pieza es defectuosa la plataforma bajará y el pistón 1 se encargará de desecharla.
  
- Fallo del elevador y el pistón 1: En caso de un fallo en el elevador, la pieza no podrá ser medida por el sensor de medida, lo que requerirá nuevamente la intervención del operario para determinar si es buena o defectuosa utilizando la botonera. Si la pieza es considerada buena, deberá colocarla manualmente en la rampa 2. En el caso de una defectuosa, dado que el pistón 1 no funciona, deberá desecharla el mismo.
  
- Fallo del elevador y el pistón 2: Que falle el pistón 2 en esta situación es irrelevante, ya que la plataforma nunca se elevará y, por lo tanto, siempre se necesitará que el operario coloque la pieza en la rampa 2. Sin embargo, si el operario indica que se trata de una pieza defectuosa, el pistón 1 se encargará de desecharla.

### **5.2.3 Estados NO implementados de la Guía GEMMA.**

En este apartado se reflejan los estados que no han sido incluidos en el diseño de la Guía GEMMA de la estación 1, los cuales han sido excluidos por la falta de necesidad o de interés para esta estación.

**A3- Parada en un estado determinado:** Debido a la poca complejidad de la estación no se ha implementado.

**A4- Parada en un estado intermedio:** Consiste en detener el proceso en un instante diferente al de inicio o final del ciclo. Al igual que en el estado anterior no se ha implementado debido a la poca envergadura del proceso.

**A5- Preparación para la puesta en marcha después de un defecto :** No es implementado ya que no se especifican acciones específicas de preparación después de un fallo en la estación.

**A7- Puesta del sistema en un estado determinado:** No es necesario ya que no se utiliza el estado A4.

**F2- Marcha de preparación:** En esta estación no es necesario preparar la máquina antes de entrar a producción.

**F5- Marcha de verificación con orden:** Debido a que ya se realiza la verificación sin orden en el estado F4, no se ha visto necesario implementar este estado.

**D2- Diagnóstico y/o tratamiento de fallos:** No se implementa debido a la poca complejidad del proceso.

#### **5.2.4 Pupitre de control de la estación 1**

En la figura 99 se muestra el aspecto del pupitre de control que se ha diseñado para la estación uno.



Figura 99. Pupitre de control creado en Factory IO para la estación uno.

### 5.2.5 Guía GEMMA de la estación uno

En la figura 100 se representa la guía GEMMA implementada para la estación uno. En ella se observan los estados incluidos, las transiciones permitidas entre ellos así como las condiciones de evolución, cuando las hay.

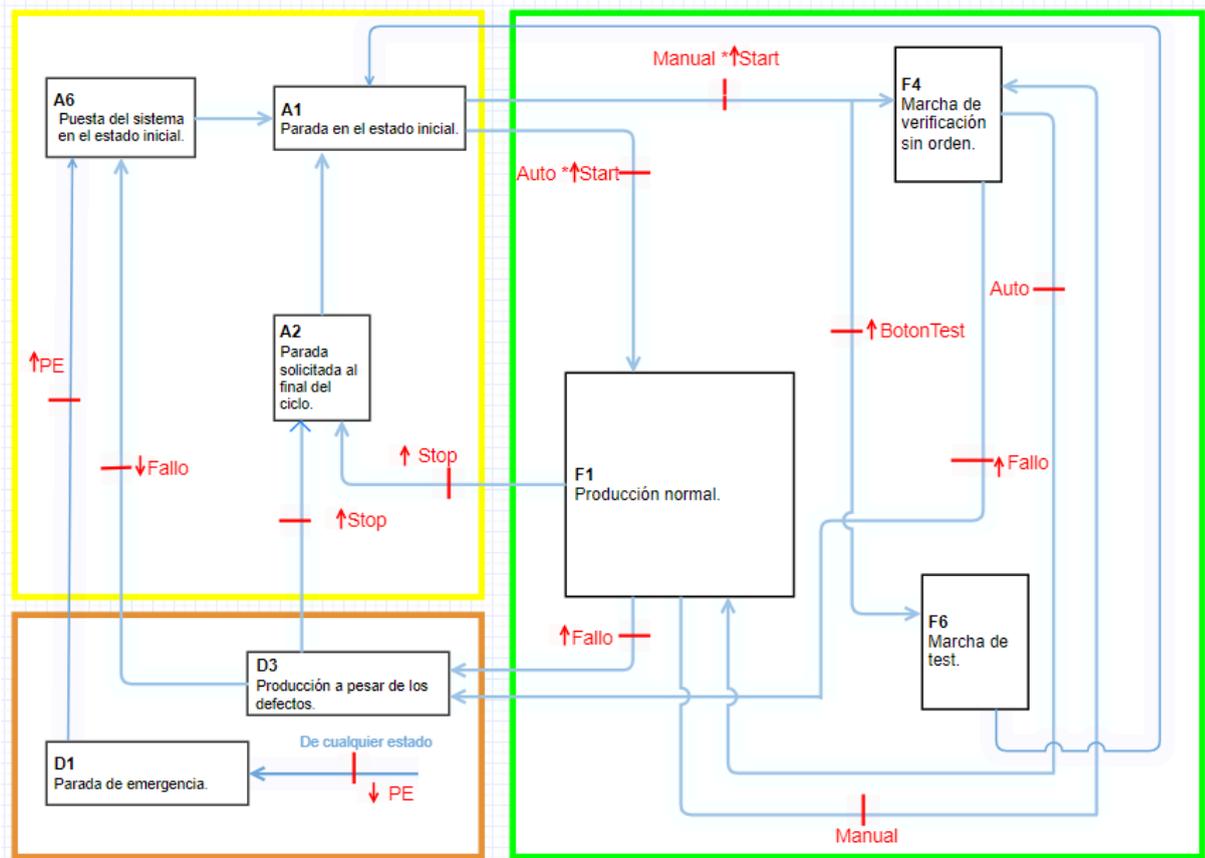


Figura 100. Guía GEMMA diseñada para la estación uno de la planta Festo.

### 5.3. Diseño de la Guía GEMMA de la estación dos

En este apartado, se describe el proceso de diseño e implementación de la Guía GEMMA en la estación dos de la Planta Festo.

#### 5.3.1 Sensores y Actuadores definidos en la estación dos

En las tablas 8 y 9 se presentan, simultáneamente, el listado de sensores y actuadores de la estación dos, incluyendo los necesarios para poder implementar los estados de la Guía GEMMA que se ha decidido incluir en el diseño.

**Tabla 8. Sensores de la estación 2.**

<b>Icono</b>	<b>Tipo</b>	<b>Comentario</b>	<b>Dirección</b>	<b>Descripción</b>
BotonLlegadaCajas	BOOL	-	I0.0	Botón empleado para simular la llegada de piezas desde la estación1.
BotonSalidaCajas	BOOL	-	I0.1	Botón empleado para simular la retirada de piezas por la estación 3.
Sensor1	BOOL	-	I0.2	Sensor que detecta la presencia de una pieza en la posición 1.
Sensor2	BOOL	-	I0.3	Sensor que detecta la presencia de una pieza en la posición 2.
Start	BOOL	-	I0.4	Botón de Start.
Stop	BOOL	-	I0.5	Botón de Stop.
MANUAL	BOOL	-	I0.6	Modo del selector para el funcionamiento automático.
AUTO	BOOL	-	I0.7	Modo del selector para el funcionamiento Manual .
BotonCinta	BOOL	-	I1.0	Botón para activar el

				movimiento de la cinta transportadora.
BotonTaladro	BOOL	-	I1.1	Botón para mover en el eje X el brazo que se corresponde con el "Taladro".
SensorCilindro	BOOL	-	I1.4	Sensor que indica que el cilindro de sujeción está retraído.
SensorSBCilindro	BOOL	-	I1.5	Sensor que indica que el cilindro de sujeción se encuentra abajo o arriba: 1: Cuando el cilindro alcanza una de las dos posiciones. 0: Mientras el cilindro está en movimiento.
SinProcesar	BOOL	-	I1.6	Modo del selector para elegir que se encuentra sin procesar piezas.
Procesando	BOOL	-	I1.7	Modo del selector para elegir que se encuentra procesando piezas.
BotonSBCilindro	BOOL	Botón Subir Bajar Cilindro.	I2.0	Botón para subir/bajar el cilindro de sujeción.

BotonCilindro	BOOL	-	I2.1	Botón para extender/retraer el cilindro de sujeción.
FinCiclo	BOOL	Botón Empujar Retraer 1.	I2.2	Botón para indicar que es el fin del ciclo.
BTest	BOOL	Botón Empujar Retraer 1.	I2.3	Botón empleado para realizar la marcha de test.
PE	BOOL	-	I2.4	Botón para la Parada de Emergencia.
Fallo	BOOL	-	I2.5	Botón para indicar que se tiene un fallo.
FalloCinta	BOOL	-	I2.6	Selector para indicar que se tiene un fallo en la cinta.
FalloTaladro	BOOL	-	I2.7	Selector para indicar que se tiene un fallo en el taladro.
MovZTaladro	BOOL	-	I3.0	Sensor que indica que el brazo del "taladro" se está moviendo. 1: Se mueve el brazo del taladro. 0: No se está moviendo el brazo del taladro.
MovZComprobar	BOOL	-	I3.1	Sensor que indica que el brazo del "cilindro de comprobación" se está moviendo.

				1: Se mueve el brazo del cilindro de comprobación. 0: No se está moviendo el cilindro de comprobación.
SensorTaladro	BOOL	-	I3.2	Sensor del brazo del taladro que detecta la pieza.
SensorComprobar	BOOL	-	I3.3	Sensor del brazo de comprobación que detecta la pieza.
BotonPiezaC	BOOL	Botón Pieza Correcta	I3.4	Botón para indicar que se tiene una pieza correcta.
BotonPiezaI	BOOL	Botón Pieza Incorrecta.	I3.5	Botón para indicar que se tiene una pieza incorrecta.

**Tabla 9. Actuadores de la estación 2.**

Icono	Tipo	Comentario	Dirección	Descripción
Cinta	BOOL	-	Q0.0	Actuador que activa el movimiento de la cinta.
Comprobacion	BOOL	-	Q0.1	Actuador que activa el movimiento del brazo de comprobación.
LlegadaCajas	BOOL	-	Q0.2	Actuador para activar el objeto "creador" de Factory IO

LuzLlegadaCajas	BOOL	-	Q0.	Luz del botón "Llegada Cajas".
SalidaCajas	BOOL	-	Q0.4	Actuador que activa el objeto "removedor" de Factory IO
LuzSalidaCajas	BOOL	-	Q0.5	Luz del botón "Salida Cajas".
LuzStart	BOOL	-	Q0.6	Luz del botón Start.
LuzStop	BOOL	-	Q0.7	Luz del botón Stop.
Taladro	BOOL	-	Q1.0	Actuador para producir el movimiento del taladro (bajar o subir el brazo)
LuzBotonCinta	BOOL	-	Q1.1	Luz del "Botón Cinta".
LuzBotonTaladro	BOOL	Luz Pieza Buena.	Q1.2	Luz del "Botón Taladro".
CilindroSuj	BOOL	Luz Pieza Mala.	Q1.3	Actuador para activar/desactivar el cilindro de sujeción.
SubirBajarCilindro	BOOL	-	Q1.4	Actuador para subir/bajar el cilindro de sujeción.
LuzPiezasC	BOOL	Luz Piezas Correctas.	Q1.5	Luz del "Botón Pieza Correcta".
LuzPiezasI	BOOL	Luz Piezas Incorrectas.	Q1.6	Luz del "Botón Pieza Incorrecta".

LuzBotonCilindro	BOOL	-	Q1.7	Luz del "Botón Pieza Cilindro".
LuzBSBCilindro	BOOL	Luz Botón Subir Bajar Cilindro.	Q2.0	Luz del "Botón Subir Bajar Cilindro".
LuzFinCiclo	BOOL	-	Q2.1	Luz del botón Fin Ciclo..
LuzBotonTest	BOOL	-	Q2.2	Luz del "Botón Test".
LE	BOOL	Luz Emergencia	Q2.3	Luz de emergencia.
LuzFallo	UNIT16	Luz Botón Fallo	Q256	Luz del botón "Fallo".
ContadorP	UNIT16	Contador Piezas	Q256-Q257	Contador para conocer el número de piezas que se tiene sobre la cinta. ContadorP=4 : Hay piezas en todas las posiciones de la cinta.

### 5.3.2 Estados implementados de la Guía GEMMA

A continuación se indican los estados de la guía que se han decidido implementar para la estación dos.

**A1- Parada en el estado inicial:** Se accede una vez se haya pasado por el estado A6 o A2, se espera a que el operario escoja un modo de operación a través de la botonera del panel de control. Las opciones son las siguientes:

- Si selecciona el modo manual y pulsa Start accede al estado F5.
- Si escoge el modo automático, pulsa Start y el contador de piezas es igual a cero, accede al estado F2.
- Pulsando el botón de test accede al estado F6.

- Al pulsar la parada de emergencia evoluciona al estado D1.
- Si se escoge el modo automático, pulsa Start y el valor del contador de piezas es igual a 4, se evoluciona al estado F1.

**A2- Parada solicitada al final del ciclo:** Este estado es empleado para parar todo lo que sea necesario una vez se quiera finalizar el ciclo.

**A6- Puesta del sistema en el estado inicial:** Implementado para cumplir con las condiciones iniciales de la estación 2. El cilindro de sujeción debe estar arriba, mientras que los brazos del taladro y el cilindro de comprobación deben estar retraídos (arriba). En este estado también se apagan todas las luces que sean necesarias y se enciende la luz del botón "Start" para indicar que debe ser pulsado.

**F2- Marcha de preparación:** Se llevan a cabo las acciones necesarias para preparar la "mesa" (que en el modelo de Factory IO ha sido sustituido por una cinta transportadora) antes de entrar al estado "Producción normal" (F1). El proceso comienza simulando la llegada de una pieza desde la estación 1, pulsando el "Botón Llegada Cajas". Cuando el sensor 1 la detecta, se activa la cinta transportadora hasta que el sensor 2 detecta la pieza, momento en el que la cinta se detiene. El cilindro de sujeción baja y se retrae para sujetar la pieza, seguidamente bajará el taladro. Dado que en Factory IO no se puede simular el taladrado de una pieza en sí, se utiliza un temporizador de 1 segundo para realizar esta operación. Transcurrido este tiempo el taladro sube y el cilindro de sujeción se extiende y sube.

A continuación, se simula la llegada de una nueva pieza desde la estación 1. La cinta se mueve y se detiene nuevamente cuando se active el sensor 2. La pieza previamente taladrada debe ser comprobada bajando el "cilindro" de comprobación (que en el modelo de Factory IO ha sido sustituido por un brazo) y la nueva debe ser sujeta por el cilindro de sujeción y taladrada. Se pulsa nuevamente el botón para la llegada de una nueva pieza, repitiendo el procedimiento anterior. Finalmente, se pulsa una última vez el botón y llega la última pieza (el contador de piezas tendrá un valor de 4). Cuando esto ocurre, y con una pieza en las cuatro posiciones, se evoluciona al estado F1.

**F1- Producción normal:** Se llevan a cabo las acciones correspondientes a la producción normal en la estación 2. Con la cinta llena de piezas (contador piezas igual a 4) se comienza pulsando el “Botón Salida Cajas” simulando que la estación 3 se lleva la pieza y se decrementa el contador de piezas en 1. A continuación, se mueve la cinta, de manera que la pieza que se encontraba en la posición 1, ahora debe ser sujeta y taladrada, y la que se encontraba en la posición 2 (taladro) debe ser comprobada, mientras que la que ya ha sido taladrada y comprobada espera a ser retirada.

Como el contador de piezas ahora tiene un valor de 3, se simula la llegada de una nueva pieza desde la estación 1 y se repite el ciclo de forma indefinida mientras el operario no fuerce la evolución a algún otro estado de la Guía GEMMA.

**F3- Marcha de cierre:** Este estado se ha incluido para que el operario pueda vaciar la "mesa" cuando lo considere oportuno. Además, tiene la posibilidad de ir sacando las piezas sin procesar (por ejemplo, si en un momento dado se diera cuenta de que, debido a un fallo en el sensor de la estación 1, se están taladrando piezas que no tienen la altura correcta) o bien procesándolas. En ambos casos se impide la llegada de nuevas piezas procedentes de la estación 1. Para elegir una opción u otra dispone de un conmutador que puede estar en dos posiciones: "Procesando" y "Sin Procesar". Dependiendo de la que elija se procederá de una manera u otra, tal y como se explica a continuación

- Sin procesar: solo será necesario vaciar la mesa de las piezas presentes sobre ella en ese momento. No se realizarán los procesos de taladrado ni comprobación, la cinta irá avanzando hasta colocar las piezas (una a una) en la posición 4, donde el operario deberá pulsar el “Botón Salida Cajas” para simular que las cajas son retiradas, ya sea por él o por la estación 3.
- Procesando: Se deben realizar las operaciones de taladrado y comprobación de las piezas, igual que en el estado F1. A medida que realicen la cinta irá avanzando para que las piezas lleguen a la siguiente posición que les corresponda. Cuando llegue una pieza a la última posición será retirada por el operario del mismo modo que en el caso anterior.

En ambos modos se llevará la cuenta del número de piezas que se encuentran sobre la cinta. Se saldrá de este estado cuando el contador llegue a 0 y no esté pulsado el “BotonTest”. Sin embargo, si se encuentra pulsado este botón se evoluciona al estado F6, para realizar una comprobación antes de entrar a producir de nuevo.

**F6- Marcha de test:** Se accede pulsando “Botón Test”. En él se realiza un ciclo de producción para verificar que todo funciona correctamente. Como ya se comentó en la estación 0, se utilizan temporizadores para evaluar el funcionamiento. En el caso de este modelo ha sido necesario que haya una pieza sobre la cinta para poder realizar todas las comprobaciones necesarias, ya que sin ella alguno de los procesos indicaba fallo pese a que estuviese funcionando correctamente. Si algún elemento de la planta falla, se encenderá la luz de fallo correspondiente, pero se continuará comprobando los demás elementos. Si todo funciona correctamente y no se cuenta con ninguna pieza sobre la cinta, al finalizar la comprobación, se evoluciona al estado A6. Sin embargo, en el caso de un fallo, se deja la decisión al operario para decidir a qué estado quiere ir. Y en el caso de que si queden piezas sobre la mesa, se evoluciona al estado F3 para retirarlas.

**F5- Marcha de verificación con orden:** Este estado corresponde al modo manual, donde se han configurado los botones correspondientes para controlar los movimientos del motor de la mesa, el cilindro de sujeción, el taladro y el cilindro de comprobación. De este modo, el operario tiene la capacidad de realizar la producción totalmente de manera manual.

**D1- Parada de Emergencia:** La parada de emergencia detiene todas las operaciones de la estación. Al quitar el pulsador de emergencia, se regresa al estado A6.

**D3- Producción a pesar de los efectos:** Este estado permite que la estación siga funcionando en presencia de dichos fallos. En él se han considerado dos tipos de fallos. El primer paso en ambos es verificar si hay una pieza en la cinta. Si no hay ninguna, se debe pulsar el “Botón Llegada Cajas” para simular la llegada de una pieza desde la estación 1.

Los fallos considerados son:

- Fallo de la cinta transportadora: el operario debe encargarse de mover la cinta manualmente (lo que equivaldría a girar la mesa en la planta real) hasta que se alcance una nueva posición correcta.
- Fallo en el cilindro de comprobación: Si el fallo se encuentra en el cilindro de comprobación, lo que ocurrirá es que cuando una pieza se situó en esta posición para ser comprobada, será el operario quien tenga que indicar mediante la botonera correspondiente si se trata de una pieza bien o mal taladrada, ya que la estación 3 necesita conocer esta información.

### 5.3.3 Estados NO implementados de la Guía GEMMA

En este apartado se reflejan los estados que no han sido incluidos en el diseño de la Guía GEMMA de la estación 2, los cuales han sido excluidos por la falta de necesidad o de interés para esta estación.

**A3- Parada en un estado determinado:** Debido a la poca complejidad de la estación no se ha implementado.

**A4- Parada en un estado intermedio:** Consiste en detener el proceso en un instante diferente al de inicio o final del ciclo. Al igual que en el estado anterior no se ha implementado debido a la poca envergadura del proceso.

**A5- Preparación para la puesta en marcha después de un defecto :** No es implementado ya que no se especifican acciones específicas de preparación después de un fallo en la estación.

**A7- Puesta del sistema en un estado determinado:** No es necesario ya que no se utiliza el estado A4.

**F4- Marcha de verificación sin orden:** Debido a que ya se realiza la verificación con orden en el estado F5, no se ha visto necesario implementar este estado.

**D2- Diagnóstico y/o tratamiento de fallos:** No se implementa debido a la poca complejidad del proceso.

### 5.3.4 Pupitre de control de la estación dos

En la figura 101 se muestra el aspecto del pupitre de control que se ha diseñado para la estación dos.



Figura 101. Pupitre de control creado en Factory IO para la estación 2.

### 5.3.5 Guía GEMMA de la estación dos

En la figura 102 se representa la guía GEMMA implementada para la estación dos. En ella se observan los estados incluidos, las transiciones permitidas entre ellos así como las condiciones de evolución, cuando las hay.

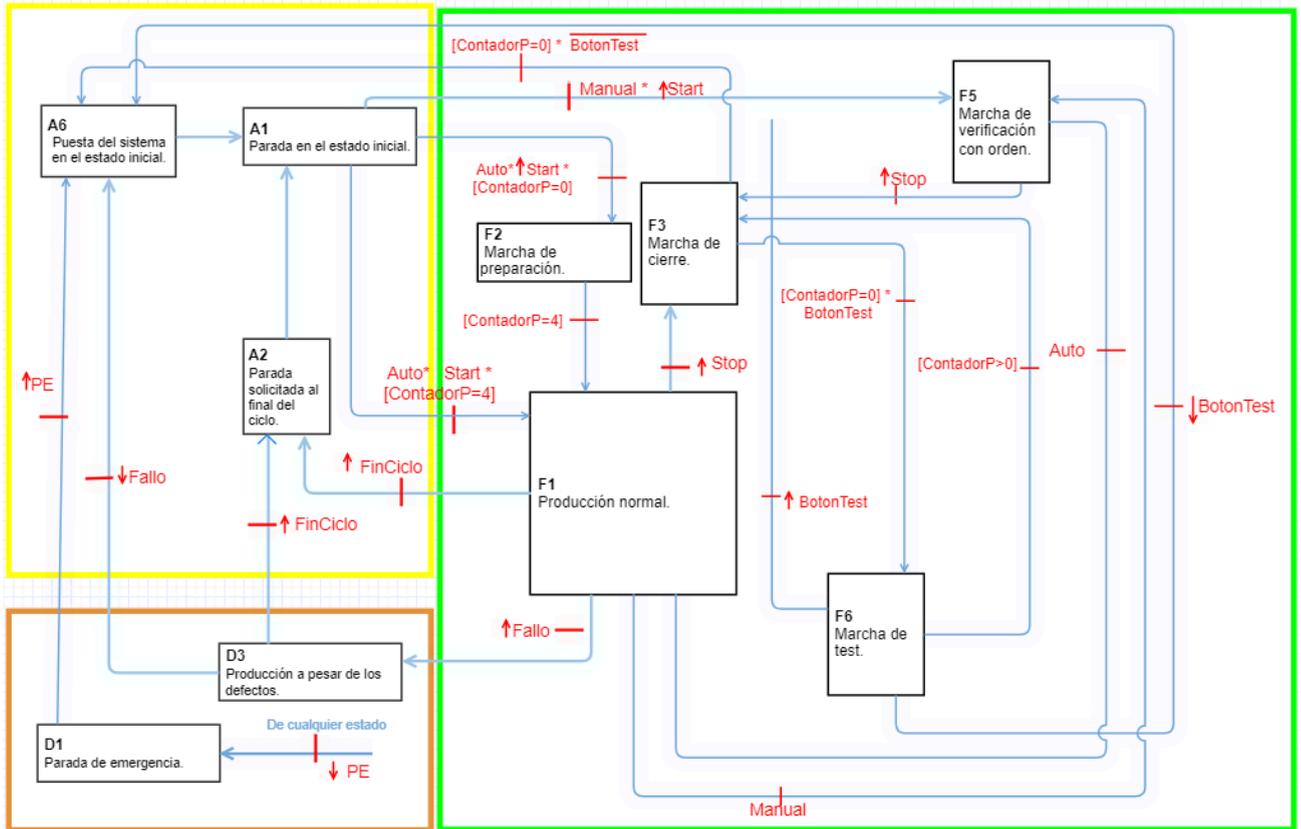


Figura 102. Guía GEMMA diseñada para la estación dos de la planta Festo.

### 5.4. Programación de la Guía GEMMA con Grafcet Studio

La Guía GEMMA para las estaciones cero, uno y dos de la planta Festo, cuyo diseño se ha descrito en los apartados previos, se ha programado en Grafcet de nivel 2 utilizando el software Grafcet-Studio. En este apartado se proporcionan algunos detalles relacionados con dicha programación. Los códigos están disponibles en [28]. Para ilustrar el funcionamiento conjunto de Grafcet-Studio con los modelos de Factory IO [29], así como los estados de la Guía GEMMA implementados, se han grabado una serie de vídeos que están disponibles en [30].

A continuación se describen los aspectos más destacados de la programación que se ha llevado a cabo:

- Se ha diseñado un diagrama de Grafcet para cada estado implementado de la guía GEMMA para cada estación. En estos diagramas se representan claramente las diferentes etapas y transiciones, lo que proporciona una visión detallada del proceso.
- Para cada estación se ha creado un Grafcet de conducción, lo que permite visualizar en qué estado se encuentra el proceso y sus posibles evoluciones. Se ha intentado optimizar el proceso para minimizar el número de etapas. Sin embargo, en algunos casos ha sido más complicado debido a que algunas acciones requieren un tiempo determinado para completarse (mediante el uso de un temporizador) o necesitan cumplir una condición de transición para poder ejecutarse.

## 6. Modelado de las estaciones de la planta Festo con el PLC-Lab.

En este apartado se presentan los diseños creados en PLC-Lab para todas las estaciones de la planta Festo. Se destacarán las similitudes y diferencias con la planta real. Se verá cómo, en términos generales, los modelos creados con PLC-Lab se parecen más a las estaciones de la planta real que los de Factory IO, lo cual se hará especialmente evidente en estaciones como la dos o la tres.

### 6.1. Estación cero

Como se vió anteriormente, la estación 0 de la planta real cuenta con un brazo intercambiador, una ventosa, un almacén de piezas y un pistón. El modelo que se ha creado en PLC-Lab, que se muestra en la figura 103, incluye los siguientes elementos:

- Brazo intercambiado con capacidad de giro de 180°.
- Pupitre de control con los botones:
  - Start: Pulsador de Start.
  - Magnet: En la planta real, se utiliza una ventosa para transportar las piezas desde la estación 0 hasta la estación 1. Sin embargo, ese no es un elemento que se pueda implementar en PLC-Lab, por lo ha sido sustituido por el botón "Magnet". Cuando este botón está activo (en estado 1), se activa el magnetismo del brazo intercambiador y de la pieza, permitiendo así que el brazo transporte la pieza a la siguiente estación. Una vez llegue allí, solo es necesario pulsar de nuevo el botón para desactivar el magnetismo y soltar la pieza.
  - Emergencia: Pulsador de emergencia.
- Pistón.
- Almacén de piezas: Se ha utilizado un elemento creador para simular la llegada de las piezas desde el almacén, como se ve en la figura 83.
- "estación 1" : Lugar donde se depositan las piezas.

La diferencia de funcionamiento entre este modelo y la planta real es el siguiente:

En la planta real, el pistón es el encargado de sacar las piezas. Sin embargo, esto no ha sido posible implementarlo de la misma manera en el modelo de PLC-Lab. En

este caso el pistón se extiende para empujar las piezas que van saliendo del almacén, permitiendo que el brazo intercambiador las recoja.

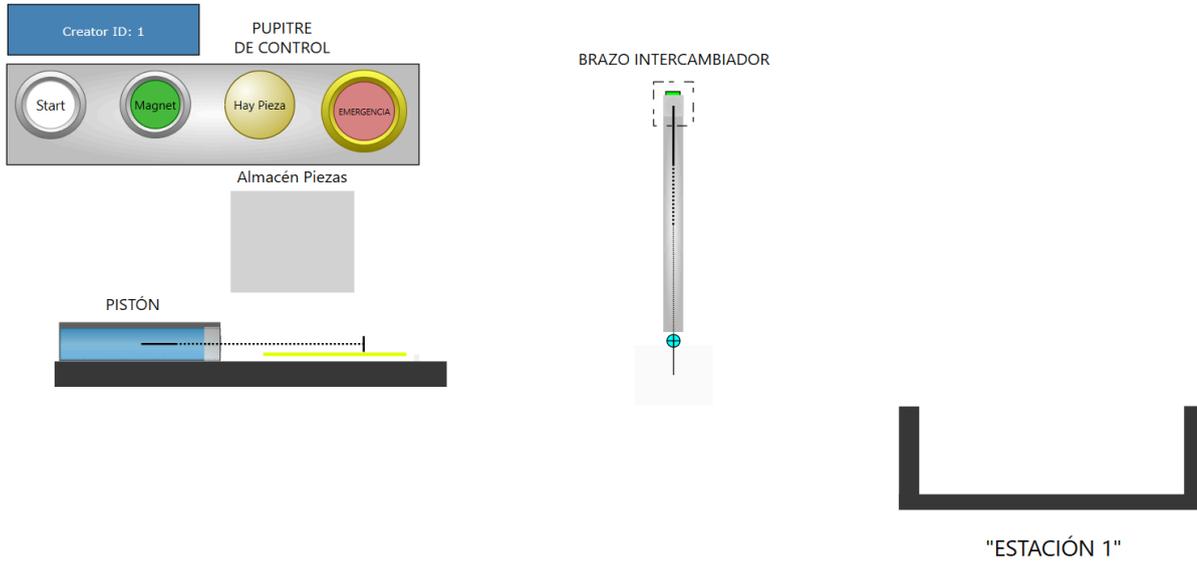


Figura 103. Modelo creado en PLC-Lab para la estación 0.

Los actuadores y sensores que ha sido necesario emplear para la creación del modelo de la figura 103 son los que se incluyen en las tablas 10 y 11.

Tabla 10. Sensores creados en PLC-Lab para la estación 0.

Símbolo	Operando	Tipo de dato	Descripción
Start	I0.0	BOOL	Activa el start.
PistonExtendido	I0.1	BOOL	Indica que el pistón se encuentra extendido.
PistonRetraido	I0.2	BOOL	Indica que el pistón se encuentra retraído.
PiezaSujetaVentosa	I0.3	BOOL	Indica que el brazo detecta una pieza.

BrazoOrigen	I0.4	BOOL	Indica si el brazo se encuentra en la posición origen.
BrazoDestino	I0.5	BOOL	Indica si el brazo se encuentra en la posición destino.
PiezaEnInicio	I0.6	BOOL	Indica la presencia de una pieza en el inicio de la plataforma.
LlegaPieza	I0.7	BOOL	
Magnet	I1.0	BOOL	Activa el magnetismo de los elementos.
EMERGENCIA	I1.1	BOOL	Indica la emergencia. 1: Se está en la situación de emergencia 0: No se está en situación de emergencia.

**Tabla 11. Actuadores creados en PLC-Lab para la estación 0.**

<b>Símbolo</b>	<b>Operando</b>	<b>Tipo de dato</b>	<b>Descripción</b>
RetraerPiston	Q0.0	BOOL	Activa el movimiento del pistón. 1: El pistón se extiende. 0: El pistón se retrae.
MoverBrazoPosPrevia	Q0.3	BOOL	Activa el movimiento del brazo hacia la posición previa (almacén). 1: El brazo se mueve hacia el almacén. 0: El brazo no se mueve hacia el almacén.
MoverBrazoPosDestino	Q0.4	BOOL	Activa el movimiento del brazo hacia la posición destino (estación 1). 1: El brazo se mueve hacia la estación 1.

			0: El brazo no se mueve hacia la estación 1.
LuzMagnet	Q0.5	BOOL	Activa la luz del botón Magnet.
SalePieza	Q0.6	BOOL	Actuador empleado para la creación de piezas mediante el objeto creador
LuzPE	Q0.7	BOOL	Activa la luz del botón Emergencia.
LuzHayPieza	Q1.0	BOOL	Luz que indica la presencia de una pieza.
LuzStart	Q1.1	BOOL	Luz del botón "Start".

## 6.2. Estación uno.

El modelo creado en PLC-Lab para la estación 1, que se muestra en la figura 104, cuenta con los elementos que se tienen en la planta real, que son: una plataforma elevadora, un pistón, dos rampas, una pestaña de sujeción, dos finales de carrera (plataforma arriba y plataforma abajo), un sensor que detecta la presencia de un objeto en la plataforma y el sensor que mide la longitud de las piezas, además de otros elementos extra que se necesitó configurar y que se explicarán a continuación.

El modelo creado en PLC-Lab es el siguiente:

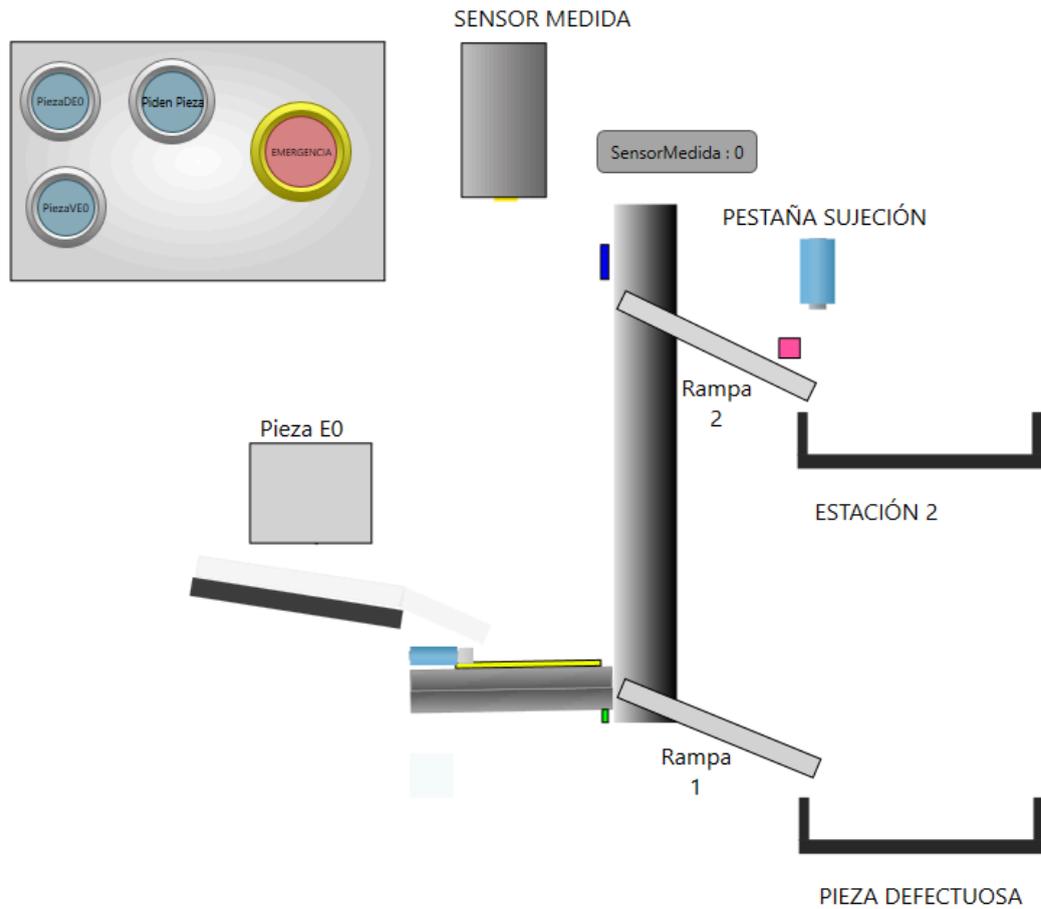


Figura 104. Modelo creado en PLC-Lab para la estación 1.

Los actuadores y sensores , y variables de datos que han sido necesarios emplear para la creación del modelo de la figura 104 son los que se incluyen en las tablas 12, 13 y 14.

Tabla 12. Sensores creados en PLC-Lab para la estación 0.

Símbolo	Operando	Comentario	Tipo de dato	Descripción
ObjetoPos1	I0.1	Objeto Posición 1.	BOOL	Detecta la presencia de una pieza sobre la plataforma.
FinalCarrera1	I0.3	-	BOOL	Final de carrera que indica que la plataforma

				se encuentra abajo.
FinalCarrera2	I0.4	-	BOOL	Final de carrera que indica que la plataforma se encuentra arriba.
PistonER	I0.5	Pistón Extendido Retraído.	BOOL	Indica si el pistón se encuentra extendido o retraído.
SensorLongitudER	I0.6	Sensor Longitud Extendido Retraído.	BOOL	Indica si el sensor de longitud se encuentra extendido o retraído.
SujecionRetraido	I0.7	-	BOOL	Indica si el pistón de sujeción de la rampa 2 (pestaña de sujeción) se encuentra retraído.
SujecionE	I1.0	Sujeción Extendido.	BOOL	Indica si el pistón de sujeción de la rampa 2 (pestaña de sujeción) se encuentra extendido.
PiezaVE0	I1.1	Pieza Válida Estación 0.	BOOL	Botón para simular la llegada de una pieza válida desde la estación 0.
PE	I1.2	Parada Emergencia.	BOOL	Botón para indicar la emergencia.
PiezaRampa2	I1.3	-	BOOL	Detecta la presencia de una pieza en la rampa 2.
PidenPieza	I1.4	-	BOOL	Botón para simular que la estación 3 solicita una pieza.
PiezaDE0	I1.5	Pieza Defectuosa Estación 0.	BOOL	Botón para simular la llegada de una pieza defectuosa desde la

				estación 0
--	--	--	--	------------

**Tabla 13. Actuadores creados en PLC-Lab para la estación 0.**

<b>Símbolo</b>	<b>Operando</b>	<b>Comentario</b>	<b>Tipo de dato</b>	<b>Descripción</b>
BajarPlataforma	Q0.0	-	BOOL	Activa el movimiento de bajada de la plataforma. 1: La plataforma baja.
SubirPlataforma	Q0.3	-	BOOL	Activa el movimiento de subida de la plataforma. 1: La plataforma sube.
ExtenderPiston1	Q0.4	-	BOOL	Activa el movimiento del pistón. 1: El pistón se extiende. 0: El pistón se retrae.
BajarSubirSensorLongitud	Q0.5	-	BOOL	Activa el movimiento de bajada y subida del sensor de longitud. 1: El sensor de longitud baja. 0: El sensor de longitud sube.
PistonSuj2	Q0.6	Pistón Sujeción 2.	BOOL	Activa el movimiento de bajada y subida del pistón de sujeción en la rampa 2. 1: El pistón de sujeción baja. 0: El pistón de sujeción sube.

LuzPE	Q0.7	Luz Parada Emergencia.	BOOL	Activa la luz del botón Emergencia.
LuzPiezaV	Q1.0	Luz Pieza Válida.	BOOL	Luz del botón PiezaVE0.
LuzPiezaD	Q1.1	Luz Pieza Defectuosa.	BOOL	Luz del botón PiezaDE0.
LuzPidenPieza	Q1.2	-	BOOL	Luz del botón Piden Pieza.

**Tabla 14. Variables de datos de la estación 1.**

Icono	Tipo	Dirección	Descripción
MedidaGuardada	UNIT16	-	Variable utilizada para guardar la medida que se tome.
MedidaTomada	BOOL	-	1: Indica que se ha tomado la medida. 0: Significa que no se ha medido.

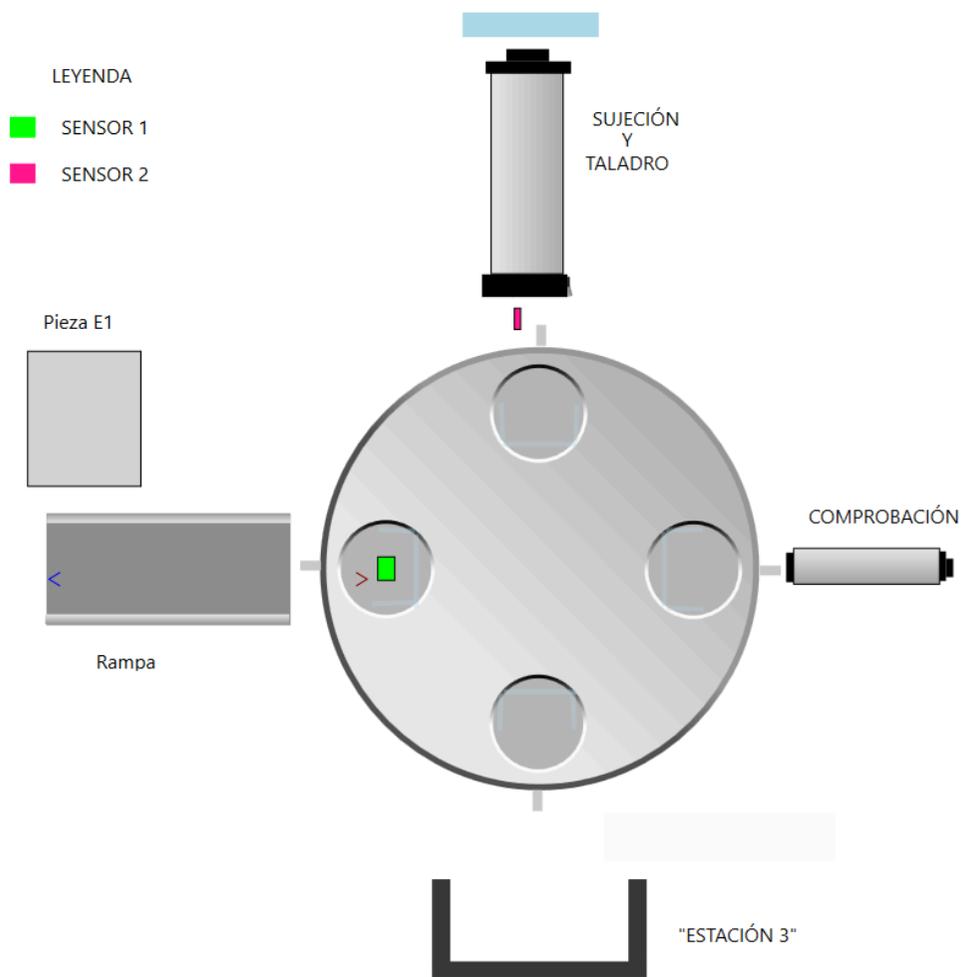
### 6.3. Estación dos

Tal y como se describe en el apartado 2.3, la estación dos de la planta real está equipada con una mesa giratoria, un taladro, un cilindro de sujeción y un cilindro de comprobación. El modelo creado en PLC-Lab para esta estación se muestra en la figura 105.

El modelo incluye los siguientes elementos:

- Mesa giratoria.
- Cilindro de sujeción.

- Cilindro que simula el taladrado.
- Cilindro que simula la comprobación.
- Pupitre de control.
- Sensor 1: Detecta cuando hay una pieza en la posición 1.
- Sensor 2: Detecta cuando la mesa ha girado 90°.
- Rampa: Llegan las piezas desde la estación 1.
- Pieza E1: Se ha utilizado un elemento creador para simular la llegada de las piezas desde la estación 1, como se ve en la figura 83, la pieza saldrá al pulsar este botón.
- "estación 3": Se "simula" que las piezas son recogidas por la estación 3.



### PUPITRE DE CONTROL



Figura 105. Modelo creado en PLC-Lab para la estación 2.

En este modelo, tanto las piezas como los objetos que se han creado y colocado en los “huecos” de la mesa para que la pieza encaje en su interior (figura 106) han sido dotados de magnetismo. Esto es necesario porque, al trabajar en 2D, no hay profundidad, y el magnetismo asegura que cuando la mesa gira, la pieza gira junto a ella.

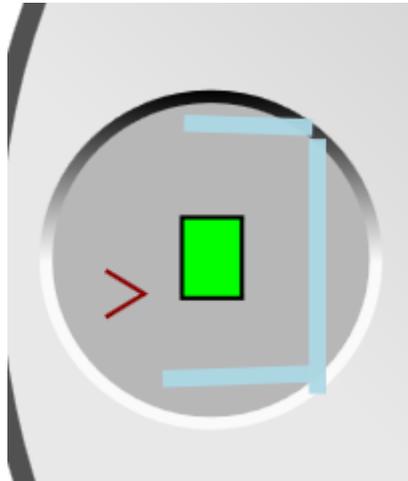


Figura 106. En color azul, los objetos creados en PLC-Lab para que las piezas encajen en la mesa de la estación 2.

Los actuadores y sensores que ha sido necesario emplear para la creación del modelo de la figura 105 son los que se incluyen en las tablas 15 y 16.

**Tabla 15. Sensores creados en PLC-Lab para la estación 2.**

<b>Símbolo</b>	<b>Operando</b>	<b>Comentario</b>	<b>Tipo de dato</b>	<b>Descripción</b>
PiezaPos1	I0.0	Pieza Posición 1.	BOOL	Indica la presencia de una pieza en la posición 1 de la mesa.
Sensor2	I0.1	-	BOOL	Indica que la mesa ha girado 90°.
CilindroSujR	I0.2	Cilindro Sujeción Retraído.	BOOL	Indica que el cilindro de sujeción se encuentra retraído.

CilindroSujE	I0.3	Cilindro Sujeción Extendido.	BOOL	Indica que el cilindro de sujeción se encuentra extendido.
TaladroR	I0.4	Taladro Retraído.	BOOL	Indica que el taladro se encuentra retraído
TaladroE	I0.5	Taladro Extendido.	BOOL	Indica que el taladro se encuentra extendido.
CilindroPR	I0.6	Cilindro Prueba Retraído.	BOOL	Indica que el cilindro de prueba se encuentra retraído.
CilindroPE	I0.7	Cilindro Prueba Extendido.	BOOL	Indica que el cilindro de prueba se encuentra extendido.
LlegaPieza	I1.0	-	BOOL	Utilizado para solicitar la llegada de piezas.
Start	I1.1	-	BOOL	Botón de Start.
BotonTaladro	I1.2	-	BOOL	Botón para activar el taladro.
BotonPrueba	I1.3	-	BOOL	Botón para activar el cilindro de prueba.
BotonSujecion	I1.4	-	BOOL	Botón para activar el cilindro de sujeción.
Magnet	I1.5	-	BOOL	Botón para activar el magnetismo.
AUTO	I1.6	-	BOOL	Modo automático del selector.
MANUAL	I1.7	-	BOOL	Modo manual del selector.
RetirarPieza	I2.0	-	BOOL	Botón para simular que la estación 3 solicita una

				pieza.
BotonPiezaC	I2.1	Botón Pieza Correcta.	BOOL	Botón para indicar que se trata de una pieza bien taladrada.
BotonPiezaI	I2.2	Botón Pieza Incorrecta.	BOOL	Botón para indicar que se trata de una pieza mal taladrada.
Procesando	I2.3	-	BOOL	Modo del selector para indicar que la estación se encuentra procesando piezas.
SinProcesar	I2.4	-	BOOL	Modo del selector para indicar que la estación no se encuentra procesando piezas.
Stop	I2.5	-	BOOL	Botón de Stop.
BotonMotorMesa	I2.6	-	BOOL	Botón para activar el movimiento de la mesa.
BotonFinCiclo	I2.7	-	BOOL	Botón para indicar el final del ciclo.
PE	I3.0	Parada de Emergencia.	BOOL	Botón para la parada de emergencia.
BotonTest	I3.1	-	BOOL	Botón para activar la Marcha de Test.
Fallo	I3.2	-	BOOL	Botón para indicar que hay un fallo.
FalloMotorM	I3.3	Fallo Motor Mesa.	BOOL	Modo del selector para indicar que se tiene un fallo en el motor de la mesa.

FalloPrueba	I3.4	-	BOOL	Modo del selector para indicar que se tiene un fallo en el cilindro de Prueba.
-------------	------	---	------	--

**Tabla 16. Actuadores creados en PLC-Lab para la estación 2.**

<b>Símbolo</b>	<b>Operando</b>	<b>Tipo de dato</b>	<b>Descripción</b>
Sujecion	Q0.0	BOOL	Activa el cilindro de sujeción.
MotorTaladro	I0.1	BOOL	Activa el motor del taladro.
MotorMesa	I0.2	BOOL	Activa el motor de la mesa.
BajarTaladro	Q0.3	BOOL	Activa el movimiento para bajar el taladro.
SubirTaladro	Q0.4	BOOL	Activa el movimiento para subir el taladro.
CilindroPrueba	Q0.5	BOOL	Activa el movimiento del cilindro de prueba. 1: El cilindro baja. 0: El cilindro sube.
LuzLlegadaPieza	Q0.6	BOOL	Luz del botón "Llega Pieza".
LuzRetiradaPieza	Q0.7	BOOL	Luz del botón "Retirar Pieza".
LuzPiezaC	Q1.0	BOOL	Luz que indica que se trata de una pieza correcta.
LuzPiezaI	Q1.1	BOOL	Luz que indica que se trata de una pieza incorrecta.
LuzStop	Q1.2	BOOL	Luz del botón Stop.
LuzMotorMesa	Q1.3	BOOL	Luz del "Botón Motor Mesa".
LuzBSujecion	Q1.4	BOOL	Luz del "Botón Sujeción".

LuzBTaladro	Q1.5	BOOL	Luz del "Botón Taladro".
LuzBPrueba	Q1.6	BOOL	Luz del "Botón Prueba".
LuzMagnet	Q1.7	BOOL	Luz del botón "Magnet".
LuzFinCiclo	Q2.0	BOOL	Luz del botón "Fin Ciclo".
LuzEmergencia	Q2.1	BOOL	Luz de emergencia.
LuzStart	Q2.2	BOOL	Luz del botón "Start".
LuzBotonTest	Q2.3	BOOL	Luz del "Botón Test".
LuzFallo	Q2.4	BOOL	Luz del botón "Fallo".
LuzFalloPrueba	Q2.5	BOOL	Luz que indica un fallo en el cilindro de prueba.
LuzFalloMotorMesa	Q2.6	BOOL	Luz que indica un fallo en el motor de la mesa.

#### 6.4. Estación tres

La estación 3 en la planta real está equipada con un un brazo de transferencia, un almacén de piezas defectuosas y un sistema de vacío para la sujeción de las piezas. El modelo correspondiente creado en PLC-Lab se muestra en la figura 107 e incluye:

- Un pupitre de control con la siguiente botonera:
  - Emergencia : Pulsador de Emergencia.
  - Pieza D: Botón para indicar que la pieza proveniente de la estación 2 es defectuosa.
  - Pieza V: Botón para indicar que la pieza proveniente de la estación 2 es válida.
  - Pieza E2: Se ha utilizado un elemento creador para simular la llegada de las piezas desde la estación 1, tal como se mostró en la figura 83. La pieza "sale" al pulsar este botón.

- Magnet: Botón para activar/desactivar el magnetismo en el brazo y en las piezas.
- Almacén para las piezas defectuosas.
- “Mesa Estación 2” : Lugar donde se simula que la pieza está en la última posición de la mesa de la estación 2, lista para ser recogida por la estación 3.
- Estación 4: Lugar donde se depositan las piezas válidas.
- Brazo intercambiador: En la estación real realiza movimientos en horizontal y vertical. Sin embargo, debido a que el simulador trabaja en 2D, no será posible visualizar el movimiento vertical, por lo que se decidió a implementar dos cilindros distintos para simular estos movimientos, uno realiza el movimiento horizontal del brazo y otro el vertical. En la figura 107 se detalla qué cilindro se encarga de cada movimiento.

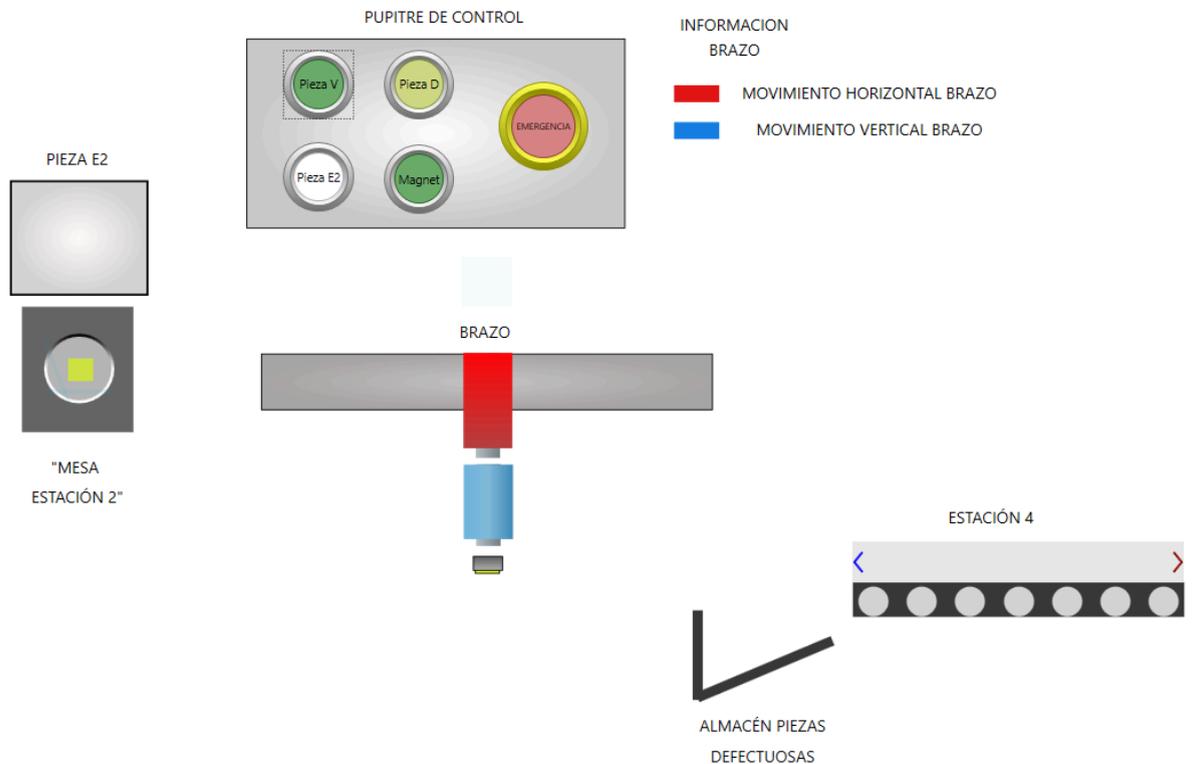


Figura 107. Modelo creado en PLC-Lab para la estación 3.

En las tablas 17 y 18 se incluyen los actuadores y sensores que se han creado para que el modelo de la figura 107 se comporte como se indica en las especificaciones de la estación 3 de la planta Festo.

**Tabla 17. Sensores creados en PLC-Lab para la estación 3.**

<b>Símbolo</b>	<b>Operando</b>	<b>Comentario</b>	<b>Tipo de dato</b>	<b>Descripción</b>
PiezaRecogida	I0.0	-	BOOL	Indica que se ha recogido una pieza de la estación 2.
BrazoEPosterior	I0.1	Brazo Estación Posterior.	BOOL	Indica que el brazo se encuentra en la estación 4.
BrazoEPrevia	I0.2	Brazo Estación Previa.	BOOL	Indica que el brazo se encuentra hacia la estación 2.
ElementoHExtendido	I0.3	Elemento Horizontal Extendido.	BOOL	Indica que el elemento horizontal está extendido.
ElementoHRetraido	I0.4	Elemento Horizontal Retraído.	BOOL	Indica que el elemento horizontal está retraído.
ElementoVExtendido	I0.5	Elemento Vertical Extendido.	BOOL	Indica que el elemento vertical está extendido.
ElementoVRetraido	I0.6	Elemento Vertical Retraído.	BOOL	Indica que el elemento vertical está retraído.
PiezaValida	I0.7	-	BOOL	Botón para indicar que se trata de una pieza válida.
PiezaDefectuosa	I1.0	-	BOOL	Botón para indicar que se trata de una pieza defectuosa.
Start	I1.1	-	BOOL	Botón de Start.
Magnet	I1.2	-	BOOL	Botón de Stop.

PiezaE2	I1.3	-	BOOL	Botón para activar el magnetismo.
EMERGENCIA	I1.4	-	BOOL	Botón para indicar una emergencia.
PiezaPos4	I1.5	Pieza Posición 4.	BOOL.	Indica la presencia de una pieza en la "posición 4" de la mesa de la Estación 2.

**Tabla 18. Actuadores creados en PLC-Lab para la estación 3.**

<b>Símbolo</b>	<b>Operando</b>	<b>Comentario</b>	<b>Tipo de dato</b>	<b>Descripción</b>
RetraerH	Q0.1	Retraer Horizontal.	BOOL	Se retrae el elemento horizontal.
ExtenderH	Q0.2	Extender Horizontal.	BOOL	Se extiende el elemento horizontal.
GirarEPosterior	Q0.3	Girar Estación Posterior.	BOOL	Activa el giro del brazo hacia la estación 4.
GirarEPrevia	Q0.4	Girar Estación Previa.	BOOL	Activa el giro del brazo hacia la estación 2.
BajarSubirV	Q0.7	Bajar Subir Vertical.	BOOL	Activa la bajada o subida del elemento vertical. 1: Baja el elemento vertical. 0: Sube el elemento vertical.
LuzPiezaV	Q1.1	Luz Pieza	BOOL	Luz del botón. "Pieza"

		Válida.		Válida”
LuzPiezaD	Q1.2	Luz Pieza Defectuosa.	BOOL	Luz del botón. “Pieza Defectuosa”.
LuzMagnet	Q1.3	-	BOOL	Luz del botón.”Magnet”.
LuzPE	Q1.4	Luz Parada de Emergencia.	BOOL	Luz del botón .”Emergencia”.

## 6.5. Estación cuatro

El modelo creado para esta estación cuenta con todos los elementos presentes en la planta real, además de algunos adicionales que fueron necesarios crear. Todos ellos pueden verse en la figura 108, mostrada a continuación. Los elementos que componen este modelo son:

- Una cinta transportadora.
- Dos desviadores que dirigen las piezas al almacén.
- Un sensor que detecta piezas al inicio de la cinta.
- Otro sensor que detecta piezas entrando a los almacenes.
- Un pupitre de control con los siguientes elementos:
  - Start: Botón de Start.
  - Stop: Botón de Stop.
  - Selector para elegir modo automático o manual.
  - Emergencia: Botón de emergencia.
  - Pieza E3: Se ha utilizado un elemento creador para simular la llegada de las piezas desde la estación 3, tal como se mostró en la figura 83. La pieza “sale” al pulsar este botón.
  - Desviador 1: Se utiliza para seleccionar el movimiento del desviador 1.
  - Desviador 2: Se utiliza para seleccionar el movimiento del desviador 2.

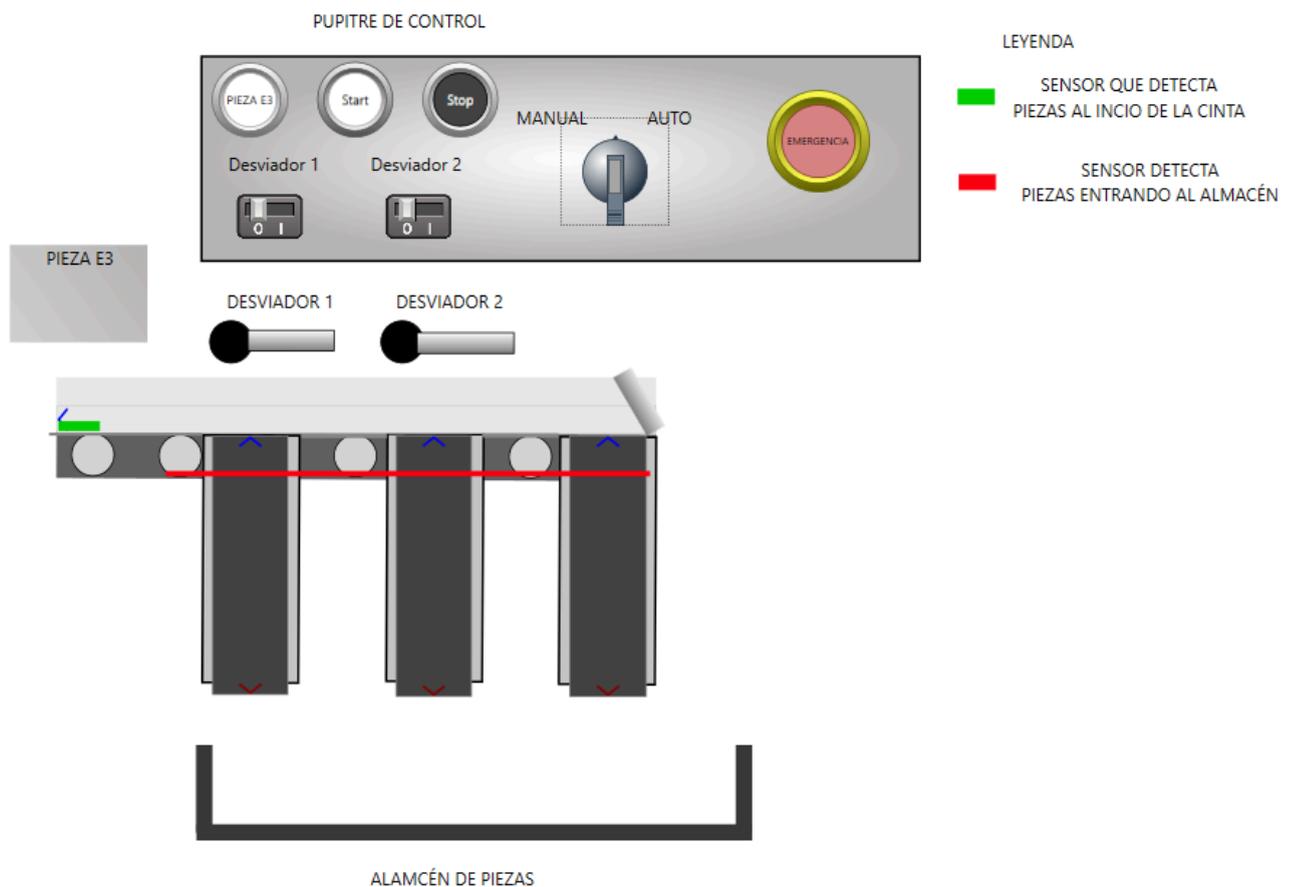


Figura 108. Modelo creado en PLC-Lab para la estación 4.

En las tablas 19 y 20 se incluyen los actuadores y sensores que se han creado para que el modelo de la figura 108 se comporte como se indica en las especificaciones de la estación 4 de la planta Festo.

Tabla 19. Sensores creados en PLC-Lab para la estación 4.

Símbolo	Operando	Comentario	Tipo de dato	Descripción
Desviador1Origen	I0.0	-	BOOL	Indica que el desviador 1 está retraído.
Desviador1Extendido	I0.1	-	BOOL	Indica que el desviador 1 está extendido.

Desviador2Origen	I0.2	-	BOOL	Indica que el desviador 2 está retraído.
Desviador2Extendido	I0.3	-	BOOL	Indica que el desviador 2 está extendido.
DetectaPiezaPrincipioCinta	I0.4	-	BOOL	Indica la presencia de una pieza al inicio de la cinta.
SensorPiezaDentroAlmacen	I0.5	-	BOOL	Indica que ha entrado una pieza al almacén.
Start	I1.0	-	BOOL	Botón Start.
AUTO	I1.1	-	BOOL	Modo del selector para escoger el modo automático.
Stop	I1.2	-	BOOL	Botón Stop.
PiezaE3.	I1.3	-	BOOL	Botón para solicitar botón pieza a la estación 3.
EMERGENCIA	I1.4	-	BOOL	Botón Emergencia.
CLAS1	I1.6	-	BOOL	Señal para indicar que se activa el desviador 1.
CLAS2	I1.7	-	BOOL	Señal para indicar que se activa el desviador 2.
Manual	I2.0	-	BOOL	Modo del selector para escoger el modo manual.

**Tabla 20. Actuadores creados en PLC-Lab para la estación 4.**

<b>Símbolo</b>	<b>Operando</b>	<b>Comentario</b>	<b>Tipo de dato</b>	<b>Descripción</b>
MoverCinta	Q0.0	-	BOOL	Activa el movimiento de la cinta.
ExtenderDesviador 1	Q0.1	-	BOOL	Extiende el desviador 1.
ExtenderDesviador 2	Q0.2	-	BOOL	Extiende el desviador 2.
RetraerDesviador1	Q0.3	-	BOOL	Retrae el desviador 1.
RetraerDesviador2	Q0.4	-	BOOL	Retrae el desviador 2.
LuzPE	Q0.5	Luz Parada Emergencia.	BOOL	Luz del botón "Emergencia".
LuzPiezaE3	Q0.6	-	BOOL	Luz del botón "Pieza E3".
LuzStart	Q0.7	-	BOOL	Luz del botón "Start".
LuzStop	Q1.0	-	BOOL	Luz del botón "Stop".

Una vez creados todos los modelos y asignados sus sensores y actuadores, se puede proceder a desarrollar la programación correspondiente para su funcionamiento. Dado que el PLC-Lab se ha planteado como una posible alternativa al Factory IO para crear los modelos, para la programación de éstos se continuará utilizando Grafcet-Studio.

## **6.6. Comparativa entre el Factory I/O y el PLC-Lab**

Con ambos software ya definidos, es posible realizar una comparativa para evaluar las ventajas que puede ofrecer uno sobre el otro.

Factory IO es un software con una interfaz intuitiva y fácil de usar, lo que permite crear modelos de estaciones industriales de forma rápida y sencilla. Sin embargo, los elementos de dibujo disponibles pueden ser limitados, al igual que las características configurables. Además, al modificar elementos existentes, las configuraciones previas se borran, teniendo que ser definidas de nuevo. Esto en PLC-Lab no ocurre, lo que hace que sea más ágil la creación del modelo. Es importante mencionar que, a partir de la versión 2.3 de Factory IO, se han añadido mejoras significativas al software y a los elementos de dibujo, lo que aumenta su utilidad.

Por otra parte, PLC-Lab también cuenta con una interfaz fácil de usar, aunque es menos intuitiva que la de Factory IO, requiriendo un mayor tiempo de aprendizaje. Este software incluye una amplia variedad de elementos de dibujo y ofrece muchas opciones de configuración, cosa con la que no se cuenta en Factory IO. Todo ello permite construir cualquier escena que se imagine.

Por tanto, se puede afirmar que, por su facilidad de uso y su capacidad para crear simulaciones rápidas, Factory IO es ideal para la educación y la formación. mientras que PLC-Lab ofrece herramientas muy precisas para la simulación de modelos de estaciones industriales, lo que lo hace adecuado para aplicaciones más técnicas.

## **7. Integración del PLC-Lab con el Grafcet Studio para la programación de las estaciones de la planta Festo**

En este apartado se detalla la programación en Grafcet-Studio para los modelos creados en PLC-Lab para todas las estaciones de la planta Festo. La primera cuestión que se planteó fue ver hasta qué punto los diagramas de Grafcet creados para la programación de la Guía GEMMA en los modelos de Factory IO seguían siendo válidos para los modelos desarrollados con PLC-Lab. Afortunadamente, se pudo reutilizar la programación existente, siendo necesario únicamente realizar algunas modificaciones.

En un primer momento, en este Trabajo de Fin de Grado se planteó usar el PLC-Lab únicamente como una herramienta de simulación alternativa al Factory IO para crear un modelo de la estación 2 de la planta Festo más parecida a la real. Por ello, y como se verá a continuación, es la única estación que tiene implementada la Guía GEMMA también para el modelo de PLC-Lab. Además, debido a la incertidumbre sobre si daría tiempo a implementar todas las estaciones, se priorizó la estación 2 por ser la que menos se parecía al modelo real. Sin embargo, con la experiencia adquirida con la creación del modelo de la estación 2, se decidió modelar las otras cuatro.

Para visualizar la simulación ambos programas deben comunicarse entre sí de alguna manera. Para lograrlo, el espacio destinado para definir los símbolos cuenta con una casilla llamada "Device", donde se puede elegir entre los distintos dispositivos PLC o sistemas de programación.

En el caso de Grafcet-Studio, se debe utilizar el dispositivo con la denominación "Sim". Es esencial asignar la misma dirección (en PLC-Lab las denominan 'operandos') a las entradas, salidas, indicadores o bloques de datos en PLC-Lab como en Grafcet Studio para lograr un funcionamiento correcto.

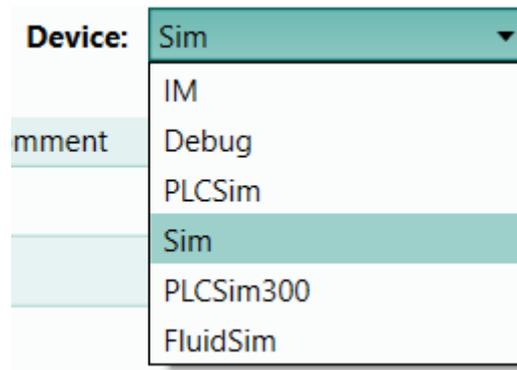


Figura 109. Dispositivos PLC o sistemas de programación en PLC-Lab.

	Device	Symbol	Operand	Data type
	Sim	<b>Start</b>	I0.0	bool

Figura 110. Ejemplo de variable creada con Device Sim en PLC-Lab.

Una vez creadas las entradas, salidas, indicadores o bloques de datos, la simulación y la programación en Grafcet-Studio, para ejecutar la simulación solamente será necesario poner en modo 'Run' ambos programas.

### 7.1. Estación cero (funcionamiento normal y emergencia)

En esta estación se han creado el estado de funcionamiento normal (F1) y la parada de emergencia (D1). A continuación se describe el funcionamiento de cada uno de ellos.

**F1- Funcionamiento normal:** Se comienza definiendo al principio sus condiciones iniciales las cuales son las que se definieron en el apartado 5.1.1 de esta memoria. Los cambios que sufre este estado respecto al estado F1 ya definido para el modelo de Factory IO, es que ahora el brazo si puede girar desde el almacén hacia la estación 1. Además, el magnetismo de la pieza y el brazo deben estar activados para transportar las piezas.

**D1 -Parada de Emergencia:** Se accede a este estado activando el pulsador de "EMERGENCIA". En él se detiene todo el proceso de la estación 0 y se

enciende la luz de emergencia. Para salir de él, se debe desactivar el pulsador de emergencia, lo que llevará de nuevo al estado F1. Sin embargo, existen las siguientes opciones para salir de este estado:

- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encuentra girado hacia la estación 2 y está activo el sensor “Pieza Posición 4”: En este caso ya hay una pieza esperando a ser procesada por a estación 3, por lo que se evoluciona al paso 2 de la secuencia de acciones de F1.
- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo no está girado hacia ninguna estación: se comienza desde el inicio del ciclo.
- Se desactiva el pulsador de “EMERGENCIA” y el brazo se encuentra girado hacia la estación 2 pero no se cuenta con una pieza en “Pieza Posición 4”: En este caso, se debe solicitar una pieza mediante el botón “Pieza E2” y continuar el proceso desde ahí.
- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encontraba girando con una pieza recogida: El brazo continuará girando hacia la estación 4 y se retomará el proceso desde el paso 5 (incluido) de la secuencia de acciones de F1.
- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encontraba girado hacia la estación 4 y no había soltado la pieza: se continuará desde el paso en el que indica con la botonera de que tipo de pieza se trata (válida o defectuosa).

## 7.2. Estación uno (funcionamiento normal y emergencia)

En esta estación se han creado el estado de funcionamiento normal (F1) y la parada de emergencia (D1). A continuación se describe el funcionamiento de cada uno de ellos.

**F1- Funcionamiento normal:** Se comienza definiendo al principio sus condiciones iniciales las cuales son las que se definieron en el apartado 5.2.1 de esta memoria, Comparado con el Estado F1 definido en el modelo de Factory IO, este modelo experimenta los siguientes cambios: ahora se dispone de un único pistón para empujar las piezas hacia las rampas, y se han incorporado dos botones para simular la llegada de piezas válidas defectuosas desde la estación 1.

**D1 -Parada de Emergencia:** Al igual que en el apartado 5.2.1 se accede a este estado activando el pulsador de “EMERGENCIA” y para salir de él, se debe desactivar. Lo que llevará de nuevo al estado F1. Sin embargo, existen las siguientes opciones para salir de este estado:

- No se contaba con ninguna pieza sobre la plataforma. Cuando se desactiva la emergencia se debe comenzar el ciclo solicitando una pieza a la estación cero.
- Se contaba con una pieza en la plataforma cuando se pulsó la emergencia, pero todavía no se había tomado su medida. Cuando se desactiva la emergencia se subirá la plataforma, se medirá y continuará el ciclo.
- Se contaba con una pieza en la plataforma, ya había sido medida, y se trataba de una pieza válida. Cuando se desactiva la emergencia se realizan las acciones necesarias en el caso de una pieza válida.
- Por último, se contaba con una pieza en la plataforma que había sido medida y se trataba de una pieza defectuosa. Cuando se desactiva la emergencia se realizan las acciones necesarias en el caso de una pieza defectuosa.

### **7.3. Estación dos (Guía GEMMA)**

Como ya se mencionó anteriormente, para esta estación se ha implementado la Guía GEMMA por completo.

Los estados que han sido implementados son los mismos que se detallaron para Factory IO en el apartado 5.3.2 de esta memoria, con algunas modificaciones en ciertas partes del proceso.

Los estados que han sufrido modificaciones respecto a los ya definidos en esta memoria, han sido los siguientes:

**A6- Puesta del sistema en el estado inicial:** Implementado para cumplir con las condiciones iniciales de la estación 2. El cilindro de sujeción, el de comprobación deben y el taladro deben estar retraídos, y el motor del taladro apagado (OFF). En este estado también se apagan todas las luces que sean

necesarias y se enciende la luz del botón “Start” para indicar que debe ser pulsado.

**F2- Marcha de preparación:** Los cambios que presenta estado F2 para el modelo creado en PLC-Lab frente al modelo de Factory IO son, que el cilindro de sujeción en este caso si se extiende y se retrae como el cilindro de la planta real, y que como se ha tenido que dotar de magnetismo a algunos elementos, este debe ser activado cuando se mueva la mesa y desactivado cuando se quiera simular que la estación 3 retira una pieza.

**F1- Producción normal:** Cuenta con los mismos cambios realizados en el estado anterior.

**F3- Marcha de cierre:** Cuenta con los mismos cambios realizados en el estado F2.

**F6- Marcha de test:** Todo se realiza de manera similar que en el apartado 5.3.1, el único cambio de este estado para el caso de este modelo ha sido que, a diferencia del creado en Factory IO, no ha sido necesaria la presencia de una pieza sobre la mesa para la correcta verificación.

#### **7.4. Estación tres (funcionamiento normal y emergencia)**

Para esta estación se han implementado los estados de funcionamiento normal (F1) y parada de emergencia (D1). A continuación se describe el funcionamiento de cada uno de ellos.

##### **F1- Funcionamiento normal.**

Se comienza estableciendo las condiciones iniciales que se definieron en el apartado 2.4 de esta memoria.

Una vez cumplidas, se solicita una pieza a la estación 2 mediante el botón “Pieza E2” y se gira el brazo hacia esta estación. A continuación, la secuencia de pasos que se sigue es la siguiente:

1. Se extiende el elemento horizontal del brazo.

2. Una vez extendido, se “baja” el elemento vertical y se activa la “Luz Magnet” para indicar que se debe pulsar el botón “Magnet”.
3. El operario pulsa dicho botón para activar el magnetismo en la pieza y en el brazo, permitiendo que el brazo pueda llevarse la pieza.
4. “Sube” el elemento vertical y se retrae el horizontal.
5. Con la pieza recogida el brazo gira hacia la estación 4.

Como ya se ha mencionado, dado que desde la estación 2 pueden llegar piezas que estén bien o mal taladradas, se debe especificar el estado de la pieza mediante la botonera creada.:

- Si se pulsa el botón “Pieza Válida”, se trata de una pieza bien taladrada. El elemento horizontal se extiende y el vertical “baja”. A su vez, se activa de nuevo la “Luz Magnet”. Cuando se pulse el botón “Magnet” se desactiva el magnetismo y la pieza se deposita sobre la estación 4. Una vez depositada, se retrae y se sube el elemento horizontal y vertical respectivamente.
- Si se pulsa el botón “Pieza Defectuosa” se trata de una pieza mal taladrada. Solamente baja el elemento vertical y se activa la “Luz Magnet” para desactivar el magnetismo y desechar la pieza por la rampa.

En ambos casos, una vez finalizado se repite la secuencia desde el paso 2.

Es importante tener en cuenta que, en caso de que la pieza se pierda por el camino (se desactive el magnetismo), se coloca de nuevo el brazo hacia la estación 4 y se comienza de nuevo el ciclo.

**D1 -Parada de Emergencia:** Se accede a este estado activando el pulsador de “EMERGENCIA”. En él se detiene todo el proceso de la estación 3 y se enciende la luz de emergencia. Para salir de él, se debe desactivar el pulsador de emergencia, lo que llevará de nuevo al estado F1. Sin embargo, existen las siguientes opciones para salir de este estado:

- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encuentra girado hacia la estación 2 y está activo el sensor “Pieza Posición 4”: En este caso ya hay una pieza esperando a ser procesada por a estación 3, por lo que se evoluciona al paso 2 de la secuencia de acciones de F1.

- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo no está girado hacia ninguna estación: se comienza desde el inicio del ciclo.
- Se desactiva el pulsador de “EMERGENCIA” y el brazo se encuentra girado hacia la estación 2 pero no se cuenta con una pieza en “Pieza Posición 4”: En este caso, se debe solicitar una pieza mediante el botón “Pieza E2” y continuar el proceso desde ahí.
- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encontraba girando con una pieza recogida: El brazo continuará girando hacia la estación 4 y se retomará el proceso desde el paso 5 (incluido) de la secuencia de acciones de F1.
- Se desactiva el pulsador de “EMERGENCIA”, pero el brazo se encontraba girado hacia la estación 4 y no había soltado la pieza: se continuará desde el paso en el que indica con la botonera de que tipo de pieza se trata (válida o defectuosa).

## **7.5. Estación cuatro (funcionamiento normal y emergencia)**

El funcionamiento de los estados F1 y D1 implementados para esta estación es el siguiente:

### **F1 -Funcionamiento normal :**

Se comienza cumpliendo con las condiciones iniciales definidas en el apartado 2.5.

Una vez cumplidas, se comienza la secuencia de acciones detallada a continuación, Cabe destacar que como ya se ha visto anteriormente, en esta estación hay un selector para elegir entre procesar piezas de forma manual o automática.

- En modo automático: .
  1. Lo primero debe ser simular la llegada de una pieza desde la estación 3, utilizando el botón “Sale Pieza Botón”.
  2. Cuando se activa el sensor “Detecta Pieza Principio Cinta” la cinta comienza a moverse.
  3. Si la señal “Clas 1” está activa, el desviador 1 se extenderá para que la pieza vaya hasta el almacén por la primera rampa. Si por el contrario la señal “CLAS 2” es la activa, el desviador 2 se extenderá y la pieza irá hasta el almacén por la segunda

rampa. Y en caso de que ninguno esté activo, la pieza entrará al almacén por la tercera rampa.

4. En todos los casos anteriores, cuando se activa el “Sensor Pieza Dentro Almacén”, se detendrá la cinta y comenzará el ciclo de nuevo desde el primer paso.
- En modo manual se procede de una forma similar en algunos aspectos, lo único que cambiará es que en este modo la cinta se mueve al pulsar botón “Start” y se detiene cuando se pulsa el botón “Stop”.

**D1 -Parada de Emergencia:** Se accede a este estado activando el pulsador de “EMERGENCIA”. En él se detiene todo el proceso de la estación 4 y se enciende la luz de emergencia. Para salir de él, se debe desactivar el pulsador de emergencia, lo que llevará de nuevo al estado F1.

Los modelos creados en PLC-Lab para las estaciones cero, uno y dos de la planta Festo, cuyo diseño se ha descrito en los apartados previos, se ha programado en Grafset de nivel 2 (creo) utilizando el software Grafset-Studio. Los códigos están disponibles en [31] y los modelos en [33]. Para ilustrar el funcionamiento conjunto de Grafset-Studio con los modelos de PLC-Lab, así como los estados de la Guía GEMMA implementados, se han grabado una serie de vídeos que están disponibles en [33].

## 8. Presupuesto

A continuación, se indica el presupuesto necesario para diseñar e implementar la Guía GEMMA utilizando las herramientas Grafcet-Studio, Factory IO y PLC-Lab.

**Tabla 21. Presupuesto del Trabajo de Fin de Grado incluyendo las licencias de software, la mano de obra y los impuestos y gastos que son de aplicación.**

PRESUPUESTO DE EJECUCIÓN DE MATERIAL (PEM)					
1, MATERIAL					
SOFTWARE					
DESCRIPCIÓN	ELEMENTO	CANTIDAD	COSTE UNITARIO	SUBTOTAL	IMPORTE TOTAL
Licencia de Grafcet-Studio	Grafcet-Studio Pro Edition (255)	20	72,50 €	72,50 €	953,85 €
Licencia de PLC-Lab	PLC-Lab MHJ-Edition	1	99,00 €	99,00 €	
Licencia de FactoryIO	Factory I/O MHJ Edition	1	695,00 €	695,00 €	
Licencia de PLC-Lab 3D	WinSPS-S7	1	87,35 €	87,35 €	
2, MANO DE OBRA					
HORAS		COSTE POR HORA		TOTAL	
300		48,00 €		14.400,00 €	
PRESUPUESTO TOTAL					
DESCRIPCIÓN				IMPORTE	
SOFTWARE				953,85 €	
MANO DE OBRA				14.400,00 €	
PEM				15.353,85 €	
COSTES INDIRECTOS (13%)				1.996,00 €	
BENEFICIO ONDUSTRIAL (6%)				921,23 €	
SUBTOTAL				18.271,08 €	
IGIC(7%)				1.278,98 €	
TOTAL				19.550,06 €	

## 9. Conclusiones y propuestas de mejora.

En este apartado se presentan las principales conclusiones que se extraen de la realización de este Trabajo de Fin de Grado, así como algunas propuestas de mejora que se plantean como una forma de ampliar o mejorar los resultados obtenidos.

En cuanto a las conclusiones, lo más destacado es que este trabajo me ha permitido familiarizarme tanto con el manejo de varias herramientas software para la construcción de modelos de plantas industriales, como Factory IO, PLC-Lab y en menor medida, PLC-Lab 3D Player como para la programación en Grafset.

Factory IO es un software de simulación en 3D que es bastante sencillo e intuitivo de utilizar y que permite un alto grado de realismo. Sin embargo, el principal inconveniente encontrado es el número limitado de elementos para la creación de escenas en la versión de la que se dispone. Por ejemplo, tal y como se ha visto, no fue posible simular la mesa giratoria de la estación dos o el brazo intercambiador de la estación cero, entre otras.

Sin embargo, el PLC-Lab, aunque sólo permite diseñar escenas en 2D, dispone de una amplia variedad de elementos con los cuales Factory IO no cuenta. Además estos elementos son altamente configurables y permiten crear modelos más cercanos a la realidad. Es un programa sencillo de manejar aunque, debido a la gran cantidad de opciones de configuración disponibles para los elementos de dibujo, es algo menos intuitivo que el Factory IO. Por último, el PLC-Lab 3D Player es el programa con el que menos he trabajado, ya que aunque posee un gran potencial al ser un programa en tres dimensiones y compartir la versatilidad de PLC-Lab, además de muchas otras funciones nuevas interesantes, actualmente sólo permite trabajar con escenas ya diseñadas y no permite crear nuevas.

La comparación entre el Factory IO y el PLC-Lab me permite concluir que, pese a los avances en las nuevas licencias disponibles de Factory IO, cuando se trata de crear modelos para, por ejemplo, ser empleado en la docencia práctica de asignaturas de automatización, sin que deban ajustarse a una planta física, el Factory IO es una herramienta interesante. Esto es lo que realizó el alumno Cristian José Luis González en su trabajo de fin de Grado [1]. En cambio, cuando se quiere crear modelos que repliquen una planta física, el PLC-Lab es la mejor opción, a

pesar de ser en 2D, ya que permite crear la escena que tu quieras sin tantas limitaciones como en Factory IO.

Una vez terminado el trabajo, los resultados reflejan que se alcanzaron los objetivos propuestos. Por un lado, se diseñó e implementó la guía GEMMA para las estaciones cero, uno y dos utilizando Grafcet Studio y los modelos en Factory IO siguiendo una estructura jerárquica en Grafcet. Además, se diseñaron los modelos de la planta Festo en PLC-Lab y se implementó la guía GEMMA en la estación dos. Para el resto de estaciones, se programó el funcionamiento normal y la parada de emergencia.

En cuanto al posible uso docente del trabajo que se ha realizado, y centrándonos en la asignatura Automatización Industrial Avanzada, de segundo curso del grado de Ingeniería Electrónica Industrial y Automática, idealmente los modelos simulados de la planta real podrían ser utilizados por los estudiantes desde sus domicilios. Esto les permitiría probar el código sin necesidad de estar físicamente en el laboratorio, lo cual promueve un aprendizaje más flexible y accesible. Puesto que para ello los estudiantes tendrían que disponer en sus domicilios de las licencias tanto de Grafcet Studio como de Factory IO o PLC-Lab, es posible que esta solución, aunque muy interesante, no resulte viable. Aun en ese caso, seguiría siendo útil tener el Factory IO o PLC-Lab y el Grafcet Studio instalados en todos los ordenadores del laboratorio, con sus correspondientes licencias. El hecho de disponer de una sola planta física de procesamiento de piezas (planta Festo), hace muy tedioso el proceso de testeo de los códigos desarrollados por todos los estudiantes de la asignatura. Con la solución que se propone, cada estudiante o grupo de estudiantes podría trabajar en su puesto programando con el Grafcet Studio los modelos de Factory IO de las estaciones y sólo cuando los códigos estuvieran ya depurados harían las pruebas finales sobre la planta física. De esta manera, los estudiantes aprovecharían mucho mejor el tiempo en el laboratorio.

Finalmente, en lo que se refiere a las propuestas de mejora, se sugiere la adquisición de la licencia Grafcet-Studio Pro (255), la cual proporciona una mayor flexibilidad en términos del número de etapas y transiciones disponibles para la programación. Esto facilitará la creación de la Guía GEMMA sin necesidad de tener que prescindir de ciertos estados que se puedan considerar interesantes porque no se disponga de etapas suficientes como para ser programadas. El mayor inconveniente es que el coste de esta licencia es de 499,00€, una cantidad

significativamente mayor que la de la versión con la que se ha trabajado en este TFG.

Además, también sería interesante adquirir la licencia de Factory IO a partir de la versión 2.3, ya que se han incorporado transportadores curvos, cintas inclinadas, piezas metálicas, selectores con 3 posiciones, brazos que pueden girar alrededor del eje Z y posiciones más precisas de los objetos. Estos son algunos de los inconvenientes que se han enfrentado en este proyecto y que, por lo tanto, quedarían resueltos.

La última mejora recomendada sería adquirir la licencia de PLC-Lab 3D Player. Aunque actualmente solo permite trabajar con escenas predefinidas en lugar de poder crear proyectos desde cero, creemos que es probable que esto cambie a corto plazo, permitiendo así la creación completa de proyectos, como ocurre con PLC-Lab.

## Referencias

- [1] Universidad de La Laguna. *Repositorio Institucional de la Universidad de La Laguna*. Disponible en <https://riull.ull.es/xmlui/handle/915/24767>
- [2] A. Brunete, "Automatización industrial," Bookdown, 2023. [En línea]. Disponible en [https://bookdown.org/alberto\\_brunete/intro\\_automatica/automatizacionindustrial.html](https://bookdown.org/alberto_brunete/intro_automatica/automatizacionindustrial.html)
- [3] Recursos CITCEA-UPC, [En línea]. Disponible en: <https://recursos.citcea.upc.edu/grafcet/avan/>
- [4] "Integración de alimentador de bandejas a célula de fabricación flexible controlada por PLC," [En línea]. Disponible en : <https://idus.us.es/handle/11441/38845>.
- [5] V. Suárez, "GemmaTelemecanique," Universidad de Oviedo, [En línea]. Disponible en: <https://isa.uniovi.es/~vsuarez/Download/GemmaTelemecanique.PDF>
- [6] UNE. "Norma Española UNE-EN 60848," mayo 2022. [En línea]. Disponible en: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0069711>
- [7] R. Villa, "Tema 9," Universidad Nacional de San Luis, [En línea]. Disponible en: <http://linux0.unsl.edu.ar/~rvilla/c3m10/tema9.pdf>
- [8] AcofiPapers, [En línea]. Disponible en: <https://acofipapers.org/index.php/eiei/article/view/1333/1337>
- [9] Factory IO, [En línea]. Disponible en: <https://docs.factoryio.com/>
- [10] Factory IO, "Factory IO Manual: Drivers," [En línea]. Disponible en: <https://docs.factoryio.com/manual/drivers/>
- [11] MHJ Tools, [En línea]. Disponible en: <https://www.mhj-tools.com/?page=plc-lab>
- [12] MHJ Tools, [En línea]. Disponible en: <https://www.mhj-tools.com/?page=plc-lab>
- [13] MHJ Wiki, "Using Collision Groups," [En línea]. Available: <https://www.mhj-wiki.de/en/plc-lab/using-collision-groups/>
- [14] MHJ Wiki, "Introduction to PLC-Lab," [En línea]. Disponible en: <https://www.mhj-wiki.de/en/plc-lab/Introduction/>

- [15] MHJ Wiki, "Lamps in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/lamps/>
- [16] MHJ Wiki, "Limit Switches in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/limit-switches/>
- [17] MHJ Wiki, "Cylinder in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/cylinder/>
- [18] MHJ Wiki, "Conveying Element in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/conveying-element/>
- [19] MHJ Wiki, "Joints in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/joints/>
- [20] MHJ Wiki, "Creator in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/creator/>
- [21] MHJ Wiki, "Indicator Objects in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/indicator-objecs/>
- [22] MHJ Wiki, [En línea]. Disponible en: <https://www.mhj-wiki.de/en/mhj/manuals/>
- [23] MHJ Wiki, "Action List in PLC-Lab," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab/action-list/>
- [24] MHJ Wiki, "Introduction to PLC-Lab 3D Player," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/plc-lab-3d-player/introduction/>
- [25] MHJ Wiki, "Installation of Grafcet-Studio," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/grafcet-studio/installation/>
- [26] MHJ Wiki, "Grafcet Workbook 3," [En línea]. Disponible en:  
<https://www.mhj-wiki.de/en/grafcet-workbook/grafcet-workbook-3/>
- [27] *V-deos-explicativos-PLC-Lab*, "GitHub Repository," Nuria Villaverde. Disponible en: <https://github.com/NuriaVillaverde/V-deos-explicativos-PLC-Lab>.

[28] *Archivos-Grafcet-Studio-TFG-Nuria-Villaverde-Blanchard*, "GitHub Repository," Nuria Villaverde. Disponible en:  
<https://github.com/NuriaVillaverde/Archivos-Grafcet-Studio-TFG-Nuria-Villaverde-Blanchard>.

[29] *Modelos-en-Factory-IO-TFG-Nuria-Villaverde-Blanchard*, "GitHub Repository," Nuria Villaverde. Disponible en:  
<https://github.com/NuriaVillaverde/Modelos-en-Factory-IO-TFG-Nuria-Villaverde-Blanchard>.

[30] *Vdeos-Guia-GEMMA-Factory-IO-Grafcet-Studio-Estaciones-0-1-y-2.*, "GitHub Repository," Nuria Villaverde. Disponible en:  
<https://github.com/NuriaVillaverde/Vdeos-Guia-GEMMA-Factory-IO-Grafcet-Studio-Estaciones-0-1-y-2.-TFG-Nuria-Villaverde-Blanchard>.

[31] "Archivos-Grafcet-Studio-para-los-modelos-de-PLC-Lab", Repositorio GitHub, Nuria Villaverde Blanchard. Disponible en:  
<https://github.com/NuriaVillaverde/Archivos-Grafcet-Studio-para-los-modelos-de-PLC-Lab-TFG-Nuria-Villaverde-Blanchard> .

[32] "Modelos-creados-en-PLC-Lab", Repositorio GitHub, Nuria Villaverde Blanchard. Disponible en:  
<https://github.com/NuriaVillaverde/Modelos-creados-en-PLC-Lab-TFG-Nuria-Villaverde-Blanchard> .

[33] "Videos-Grafcet-Studio-y-PLC-Lab", Repositorio GitHub, Nuria Villaverde Blanchard. Disponible en:  
<https://github.com/NuriaVillaverde/Videos-Grafcet-Studio-y-PLC-Lab-TFG-Nuria-Villaverde-Blanchard> .