

Máster Universitario en Ingeniería Industrial

Simulación de sistemas de automatización

Jorge Esteban Pérez

Silvia Alayón Miranda

Septiembre 2022

La publicación de este Trabajo Fin de Máster solo implica que el estudiante ha obtenido al menos la nota mínima exigida para superar la asignatura correspondiente, no presupone que su contenido sea correcto, aunque si aplicable. En este sentido, la ULL no posee ningún tipo de responsabilidad hacia terceros por la aplicación total o parcial de los resultados obtenidos en este trabajo. También pone en conocimiento del lector que, según la ley de protección intelectual, los resultados son propiedad intelectual del alumno, siempre y cuando se haya procedido a los registros de propiedad intelectual o solicitud de patentes correspondientes con fecha anterior a su publicación.

ÍNDICE

1.- Resumen.	6
1.- Abstract.	6
2.- Objetivo.	6
2.- Objective.	6
3. Introducción.	6
3. Introduction.	8
4. Herramientas de simulación en automatización.	9
4.1.- Simulación de PLC.....	9
4.1.1.- Codesys.	9
4.1.2.- S7-PLCSIM.	10
4.1.3.- OpenPLC.....	11
4.2.- Simulación de automatización de plantas industriales.	12
4.2.1.- Realvirtual.io.....	12
4.2.2.- Factory I/O.	14
4.2.3.- Emulate3D.	16
4.2.4.- CIROS.....	17
4.3.- Selección.....	18
5.- Factory I/O.	18
5.1.- Entorno de trabajo.	19
5.2.- Paleta.	21
5.2.1.- Elementos del modelo.	21
5.3.- Navegación.	40
5.4.- Etiquetas de sensores y actuadores.	41
5.5.- Escenas predeterminadas.....	41
5.5.1.- Cinta transportadora.....	42

5.5.2.- Doble cinta transportadora.	43
5.5.3.-Tanque de llenado.	44
5.5.4.- Entrada de artículos.	44
5.5.5.- Ensamblador.	45
5.5.6.- Ensamblador analógico.	46
5.5.7.- Almacén automatizado.	46
5.5.8.- Estación de almacenaje.	47
5.5.9.- Estación convergente.	48
5.5.10.- Elevador básico.	48
5.5.11.- Elevador avanzado.	49
5.5.12.- Control de nivel.	50
5.5.13.- Paletizado.	50
5.5.14.- Recoger y colocar (Básico).	51
5.5.15.- Recoger y colocar (XYZ).	52
5.5.16.- Línea de producción.	52
5.5.17.- Estación separadora.	53
5.5.18.- Clasificación por altura (Básico).	54
5.5.19.- Clasificación por altura (Avanzado).	54
5.5.20.- Clasificación por peso.	55
5.5.21.- Estación de clasificación.	56
5.6.- Controladores de E/S.	56
6.- OpenPLC.	60
6.1 Editor.	60
6.1.1.- Entorno de trabajo.	60
6.1.2.- Elementos.	62
6.2.- Runtime.	64
7.- Conexión OpenPLC – Factory I/O.	68

8.- Análisis de usabilidad.....	76
8.1.- Análisis OpenPLC.....	76
8.2.- Análisis Factory I/O.	78
9.- Resultados: prácticas probadas.	79
9.1.- Práctica 1: Automatización de una cinta transportadora.	80
9.2.- Práctica 2: Automatización de un montacargas.....	81
9.3.- Práctica 3: Automatización de un semáforo.	83
9.4.- Práctica 4: Automatización de un sistema clasificador de paquetes.	84
9.5.- Práctica 5: Automatización de un sistema clasificador de paquetes según peso.....	86
10.- Conclusiones.....	88
10.- Conclusions	89
11.- Referencias.	90
ANEXO I	91
Práctica 1: Automatización de una cinta transportadora.....	91
ANEXO II.....	92
Práctica 2: Automatización de un montacargas.	92
ANEXO III.	93
Práctica 3: Automatización de un semáforo.	93
ANEXO IV.	95
Práctica 4: Automatización de un sistema clasificador de paquetes.....	95
ANEXO V.....	97
Práctica 5: Automatización de un sistema clasificador de paquetes según peso. ...	97

1.- Resumen.

En este trabajo fin de máster se realiza un estudio de diferentes paquetes de simulación 3D de automatización, con el objetivo de encontrar alguno de código abierto. Tras realizar este estudio se comprueba la viabilidad de incorporar los softwares escogidos a los programas de prácticas de las asignaturas de automatización de la Escuela.

1.- Abstract.

In this master's thesis, a study of different 3D automation simulation packages is carried out, with the aim of finding an open source one. After carrying out this study, the feasibility of incorporating the chosen software into the practice programs of the automation subjects of the University is verified.

2.- Objetivo.

El objetivo de este trabajo fin de máster (TFM), del Máster en Ingeniería Industrial de la Universidad de La Laguna, es:

- Estudiar diferentes herramientas software de simulación en el campo de la Automatización.
- Estudiar en más profundidad las herramientas disponibles de software libre, y analizar la viabilidad de su inclusión en los programas de prácticas.

2.- Objective.

The objective of this Master's Thesis (TFM), of the Master's Degree in Industrial Engineering at the University of La Laguna, is:

- Study different simulation software tools in the field of Automation.
- Study in greater depth the available free software tools, and analyze the feasibility of their inclusion in practical sessions.

3. Introducción.

Las prácticas de en ingeniería son indispensables y necesarias para la adquisición de las competencias de la disciplina. Por este motivo, todos los planes educativos en las escuelas universitarias las incluyen en múltiples asignaturas.

La Automatización no es una excepción, las prácticas en esta área son también necesarias. Pero planear laboratorios de Automatización no es una tarea trivial, ya que normalmente los centros no cuentan con el espacio ni el presupuesto suficiente para incorporar plantas industriales reales a sus programas de prácticas. Es aquí cuando la Simulación entra en juego, como una ventaja desde el punto de vista educativo.

Existen muchos trabajos que evidencian los beneficios de la simulación en el campo de la Educación. En la última década se ha producido un incremento notable de los simuladores comerciales en la docencia [10]. Por ejemplo, para el control de procesos químicos, los trabajos [10,11] proponen simuladores cuyo objetivo es modelar el comportamiento en estado estacionario del proceso, teniendo en cuenta las variables de presión, temperatura y velocidad de flujo. Estos simuladores de procesos permiten prever el comportamiento de un proceso, analizar simultáneamente diferentes casos mediante la modificación de las variables en tiempo real, optimizar las condiciones de operación en plantas existentes o nuevas y hacer un seguimiento de una planta a lo largo de su vida útil.

En el ámbito de la automatización de plantas industriales también hay muchas propuestas de simulación. Por ejemplo, en [7] se muestra el diseño y desarrollo de un sistema de clasificación y montaje de piezas, y su posterior almacenamiento dependiendo del tipo de pieza. Este es un ejemplo de simulación en automatización con la herramienta CODESYS, que simula el comportamiento de un PLC, y con el software Factory I/O, con el que se modela la planta en 3D.

También existe la posibilidad de controlar el modelo 3D de la planta directamente con un autómatas. Es el caso presentado en [8], donde la automatización se realiza con un autómatas M241 de Schneider Electric y la simulación de la planta mediante el software Factory I/O.

Con el objetivo de mostrar las numerosas posibilidades que existen en este sector, se puede ver cómo en [9] se realiza el mismo proyecto anteriormente descrito [8], pero utilizando para la simulación de la planta el software CIROS.

Como se puede observar, las herramientas de simulación en la enseñanza de automatización son claves para que el estudiante aprenda cómo automatizar plantas industriales. El

inconveniente de estas herramientas suele ser su coste, en algunos casos, excesivo, para los centros educativos. Por este motivo, este TFM intentará buscar soluciones de código abierto.

3. Introduction.

Practical sessions are essential and necessary for the acquisition of the skills of the discipline. For this reason, all educational plans in university schools include them in multiple subjects.

Automation is not an exception, practices in this area are also necessary. But planning Automation labs is not a trivial task, since centers usually do not have the space or the budget to incorporate real industrial plants into their internship programs. This is when Simulation comes into play, as an advantage from an educational point of view.

There are many works that show the benefits of simulation in the field of Education. In the last decade there has been a notable increase in commercial simulators in teaching [10]. For example, for the control of chemical processes, the works [10,11] propose simulators whose objective is to model the steady-state behavior of the process, taking into account the variables of pressure, temperature and flow rate. These process simulators make it possible to predict the behavior of a process, simultaneously analyze different cases by modifying variables in real time, optimize operating conditions in existing or new plants, and monitor a plant throughout its useful life.

In the field of industrial plant automation there are also many simulation proposals. For example, [7] shows the design and development of a parts classification and assembly system, and its subsequent storage depending on the type of part. This is an example of simulation in automation with the CODESYS tool, which simulates the behavior of a PLC, and with the Factory I/O software, with which the plant is modeled in 3D.

There is also the possibility of controlling the 3D model of the plant directly with an automaton. This is the case presented in [8], where the automation is carried out with a Schneider Electric M241 automaton and the simulation of the plant using the Factory I/O software.

In order to show the numerous possibilities that exist in this sector, it can be seen how in [9] the same project described above [8] is carried out, but using the CIROS software for the plant simulation.

As can be seen, simulation tools in automation teaching are key for students to learn how to automate industrial plants. The drawback of these tools is usually their cost, in some cases excessive, for schools. For this reason, this TFM will try to find open source solutions.

4. Herramientas de simulación en automatización.

En este apartado se estudian las principales herramientas de simulación en automatización que existen en el mercado actual, tanto para simular autómatas programables (Programmable Logic Controller – PLC) como para simular plantas industriales.

4.1.- Simulación de PLC.

Los distintos softwares de simulación 3D estudiados son los siguientes:

- Codesys.
- PLCSIM.
- OpenPLC.

4.1.1.- Codesys.

CoDeSy (Controller Development System) [5] es una plataforma de software orientada a las tecnologías de automatización industrial, aplicable en cualquier hardware, que ofrece un sistema de desarrollo de controladores de acuerdo con la norma IEC 61131-3 con todos los lenguajes de programación definidos independientemente del fabricante del hardware.

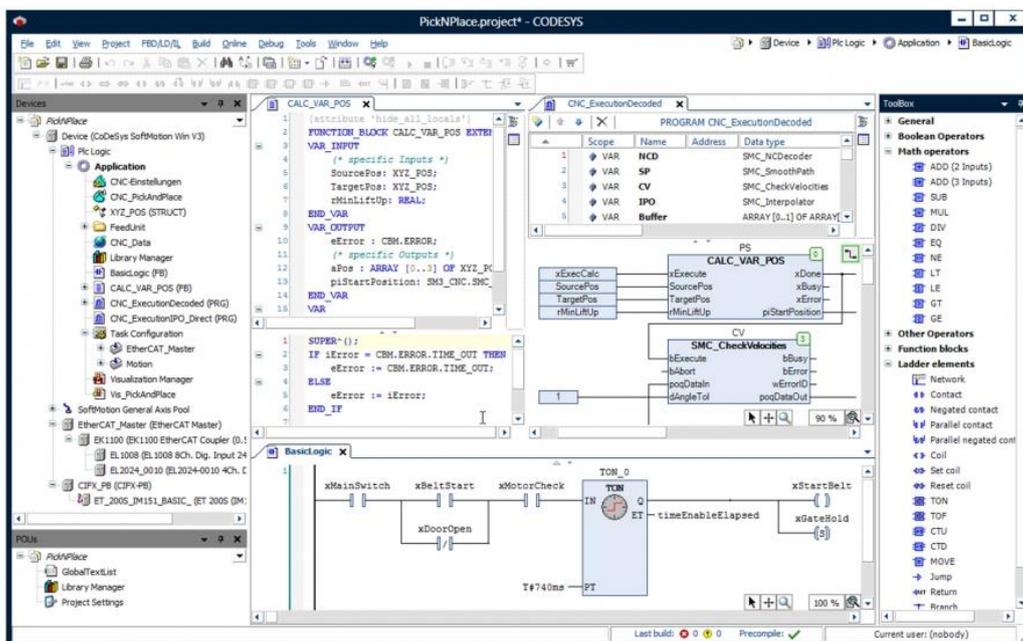


Figura 1.- Codesys [5].

El objetivo principal de Codesys es proporcionar a los usuarios un soporte práctico en la implementación de sus tareas de programación de entornos de automatización. Una de sus mayores particularidades es que es independiente al hardware utilizado, lo que posibilita que se puedan programar muchos controladores.

Estos son algunos de sus beneficios:

- Producto disponible como un runtime modular de código único para diferentes plataformas
- Soporte para todos los procesadores y sistemas operativos comunes para sistemas industriales.
- Fácil implementación de propiedades específicas del dispositivo mediante librerías (por ejemplo, para funciones de E/S propias)
- Adaptación y capacidad de ampliación de la interfaz de usuario mediante plugins específicos del fabricante (por ejemplo, plantillas, editores, configuradores y asistentes)
- Integración en arquitecturas de sistemas existentes mediante interfaces o estándares patentados, como OPC/OPC UA

Se trata de un software de simulación de PLC que cuesta en torno a unos 1.400€ por licencia.

4.1.2.- S7-PLCSIM.

La aplicación S7-PLCSIM [12], creada por Siemens, permite ejecutar y comprobar el programa de usuario en un sistema de automatización (PLC) simulado en un PC o en una unidad de programación. Es un simulador de autómatas Siemens, que incorpora un sencillo interface de usuario para visualizar y modificar diversos parámetros utilizados por el programa. En la actualidad viene integrado en el software TiaPortal de Siemens.

Las principales ventajas de este software son:

- Emulación elevada de una gran cantidad de funciones: permite visualizar las funciones tal como si se estuviese llevando a cabo físicamente en una instalación.
- Generación de varios controladores virtuales a la vez: se pueden desarrollar distintos controladores con funciones independientes entre cada uno.
- Desarrollo de trabajos para la industria 4.0: en la digitalización, favorece el mantener una buena capacidad de comunicación entre diferentes servidores.

- Compatibilidad con otros productos de Siemens: este programa ofrece distintas conexiones sin que se presenten fallos.

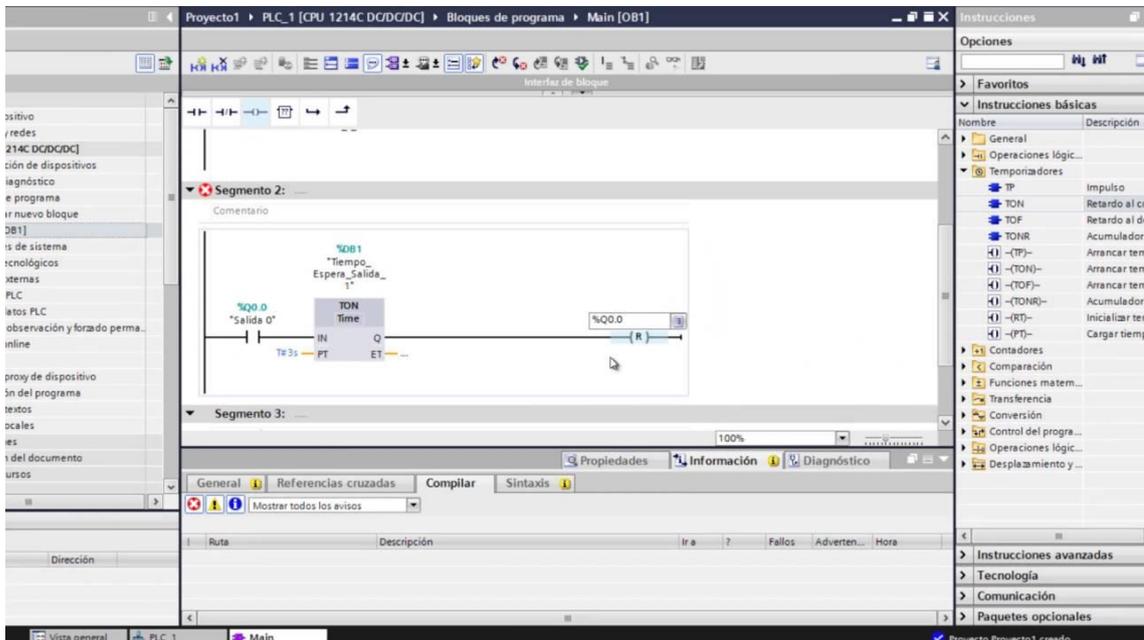


Figura 2.- S7-PLCSIM [12].

Este software de simulación de PLC tiene un coste aproximado de 2.000€ por licencia.

4.1.3.- OpenPLC.

OpenPLC es un simulador de PLC cuya característica principal es que es de código abierto, estandarizado y funcional. OpenPLC se creó siguiendo las indicaciones del estándar “IEC 61131-3” que define los lenguajes básicos de programación.

El software tiene dos partes: OpenPLC Editor y OpenPLC Runtime. La primera permite la edición del código, la segunda es la que ejecuta el simulador de PLC. Los códigos generados con el editor luego pueden ser cargados en el PLC simulado del Runtime.

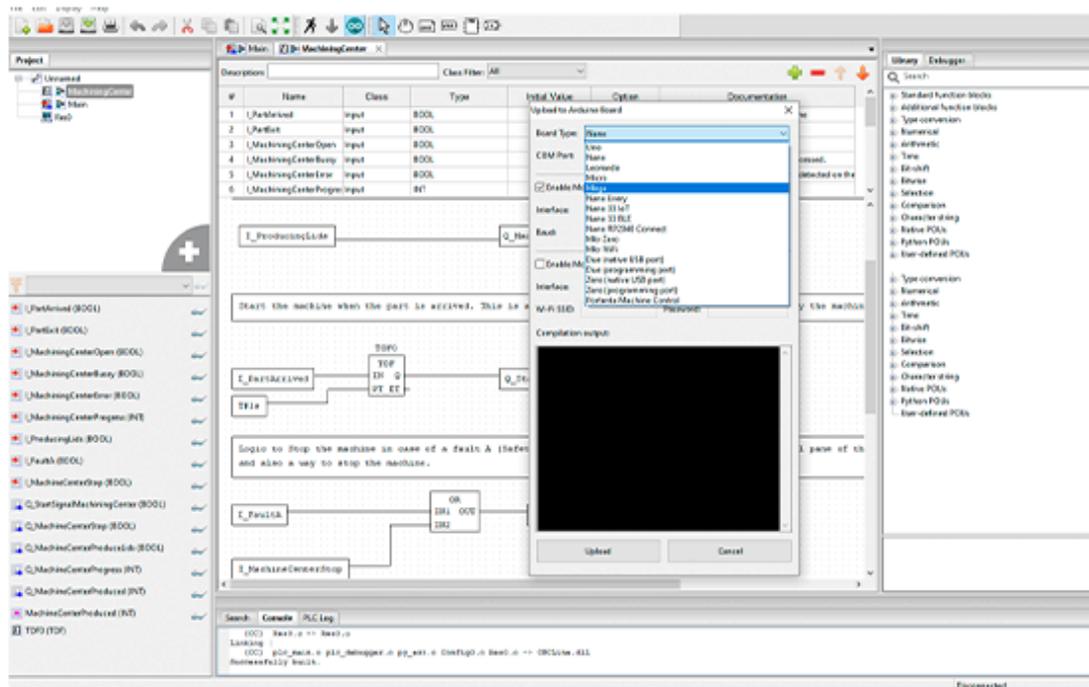


Figura 3.- OpenPLC [2].

OpenPLC es totalmente gratuito y de código abierto y es compatible para diferentes sistemas operativos.

4.2.- Simulación de automatización de plantas industriales.

Los softwares de modelado y simulación 3D estudiados son los siguientes:

- Realvirtual.io.
- Factory I/O.
- Emulate3D.
- CIROS.

4.2.1.- Realvirtual.io.

Realvirtual.io [3] es un marco abierto para visualización, simulación, puesta en marcha virtual, VR y realidad aumentada, basados en el motor de juego Unity en el entorno de automatización. El objetivo es permitir al usuario la creación de gemelos digitales para la experimentación.

Las interfaces de automatización para controladores industriales (ABB Robotstudio, OPCUA, Siemens S7, TwinCAT...) permiten la integración con el controlador real. Se puede integrar en sus propias aplicaciones en la plataforma Windows, IOS o Android como reproductor Unity.

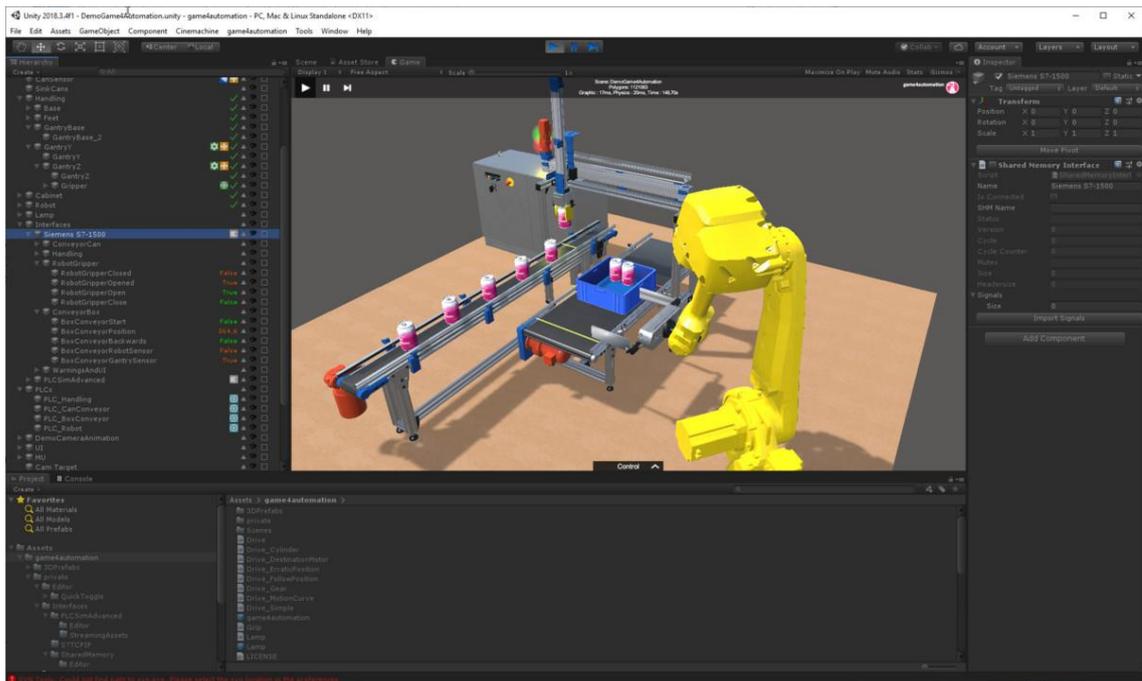


Figura 4.- realvirtual.io [3].

Realvirtual.io contiene interfaces para TwinCAT, Modbus, Simit, PLCSim Advanced y vía TCP-IP para hardware de Siemens como S7-300, S7-1500, Sinumerik y Simotion. Muchos otros controladores se pueden conectar con OPC UA.

La simulación realvirtual.io se basa en Realvirtual.io Starter o Professional. La biblioteca permite la simulación de procesos de flujo de materiales basados en rutas, como sistemas de transporte. Una simulación de flujo de materiales continuo y de alto rendimiento permite obtener imágenes detalladas de todos los sensores industriales, como escáneres láser o sistemas de cámaras.

La versión de inicio, compuesta por el marco base de realvirtual.io completo, la interfaz Siemens S7 TCP-IP incluida y acceso a los controles de hardware de Siemens (S7-300, S7-400, S7-1200, S7-1500, Sinumerik y Simotion), cuesta tan sólo 8.93€. Pero uno de los principales inconvenientes de este software es que para manejarlo es necesario tener nociones de programación de Unity.



Figura 5.- Simulador realvirtual.io [3].

Uno de los inconvenientes de este software de simulación es que es necesario tener nociones de programación en el software Unity.

La versión de inicio está compuesta por el marco base de realvirtual.io completo, la interfaz Siemens S7 TCP-IP incluida y acceso a los controles de hardware de Siemens (S7-300, S7-400, S7-1200, S7-1500, Sinumerik y Simotion).

4.2.2.- Factory I/O.

El Factory IO [1] es un software de simulación 3D de una fábrica, que permite aplicar técnicas de automatización sobre plantas simuladas. Ha sido diseñado para poder trabajar de una manera fácil sobre algunos de los escenarios más comunes que se pueden encontrar en las instalaciones de una industria.

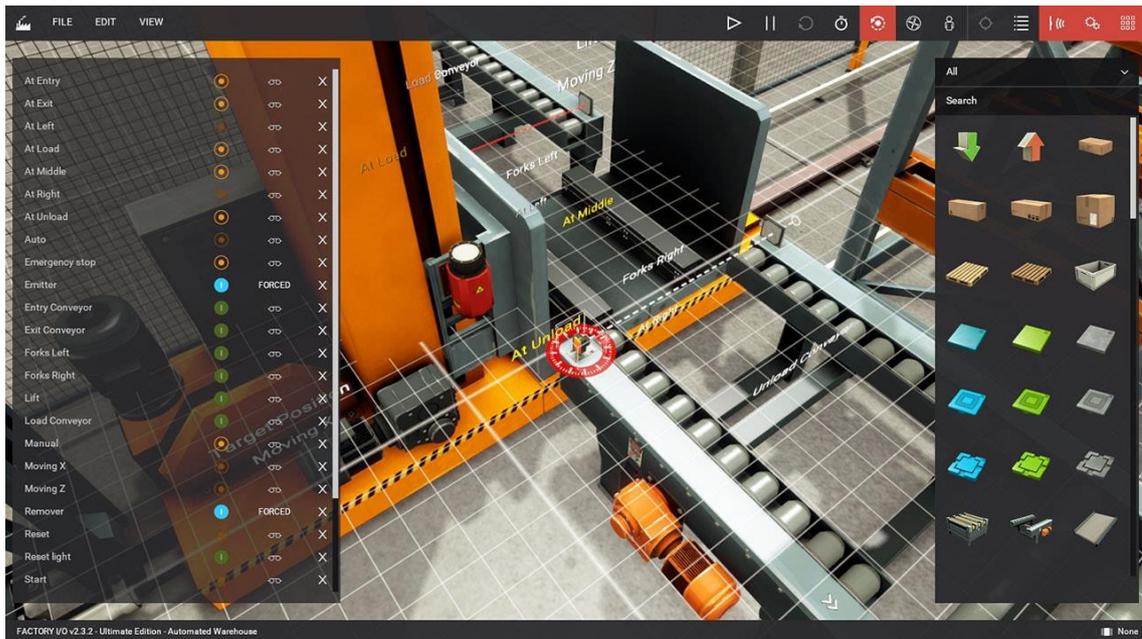


Figura 6.- Factory I/O [1].

Las herramientas de edición inteligente Factory I/O hacen que la creación de una escena 3D sea fácil y cómoda. Se utiliza la biblioteca de piezas industriales y se personaliza las entradas y salidas de la fábrica creando escenarios personalizados.

Factory I/O utiliza controladores para interactuar con PLC, simuladores PLC, Modbus... Cada edición incluye un paquete de controladores para distintas tecnologías (Allen-Bradley, Siemens...).

Las distintas versiones de este software de simulación de PLC tienen un coste que incrementa dependiendo de las funciones que se requieran. La versión que es capaz de conectarse a los simuladores de PLC de Siemens cuesta 399€.

Las principales ventajas de este software son:

- Su manejo es muy sencillo.
- No es necesario saber programar en ningún lenguaje de programación de escenas de video.
- Permite diseñar instalaciones industriales personalizadas, además de las predeterminadas.

4.2.3.- Emulate3D.

Emulate3D Controls Testing [6], propiedad de Rockwell Automation, permite probar y depurar su sistema de control fuera de línea, antes de que se construya el equipo real y sin interrumpir la producción existente. Comenzó a operar en 2005 y es reconocido como líder del mercado de software de visualización, simulación y emulación de clase mundial en la industria de la automatización.

La tecnología de software tiene un historial comprobado en una amplia gama de sectores industriales, incluidos el automotriz, la logística y el almacenamiento, los aeropuertos y el manejo de equipaje, la robótica, la automatización y el manejo de materiales, así como diversos procesos de fabricación.

Emulate3D ha desarrollado una relación exitosa con muchos integradores de automatización y desarrolladores de PLC, así como con proveedores de software CAD.

Los modelos de prueba de controles se conectan directamente al PLC; los sensores fotoeléctricos del modelo bloqueados por las cargas escriben en las entradas del PLC, y las salidas del PLC pueden cambiar los estados del equipo modelado.

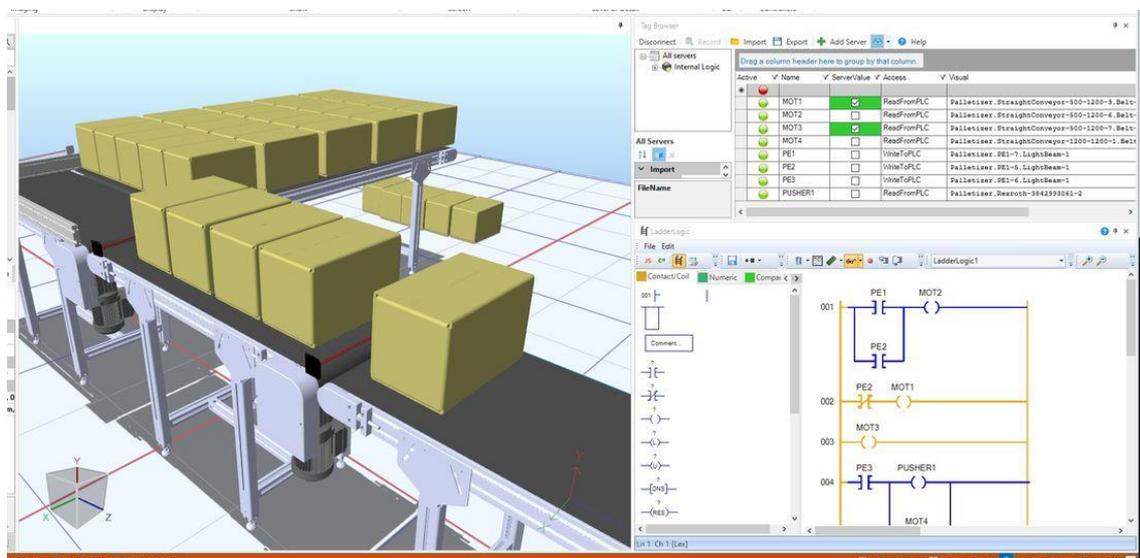


Figura 7.- Emulate3D [4].

Los modelos de prueba de controles Emulate3D se conectan de forma nativa a una amplia gama de PLC y se pueden conectar a productos de otros fabricantes a través de una amplia gama de protocolos industriales estándar.

Cabe destacar que se trata de un software de simulación 3D que no dispone de licencia gratuita para su uso, aunque se puede solicitar una versión demo gratuita.

4.2.4.- CIROS.

CIROS [6] es un software desarrollado por la empresa Festo que es compatible con diversos escenarios de simulación de PLC y programación fuera de línea, en instalaciones de tecnología de la automatización controladas por PLC. Se puede acoplar a CIROS un PLC real mediante EasyPort. En esta situación, CIROS recibe los valores de salida del PLC, simula los procesos controlados y transmite los valores actuales de los sensores mediante EasyPort de nuevo a las entradas del PLC. Otra opción consiste en utilizar diferentes controles de software, como S7-PLCSIM o Codesys para controlar la ejecución sin el hardware correspondiente. CIROS también admite la conexión del control mediante un servidor OPC.

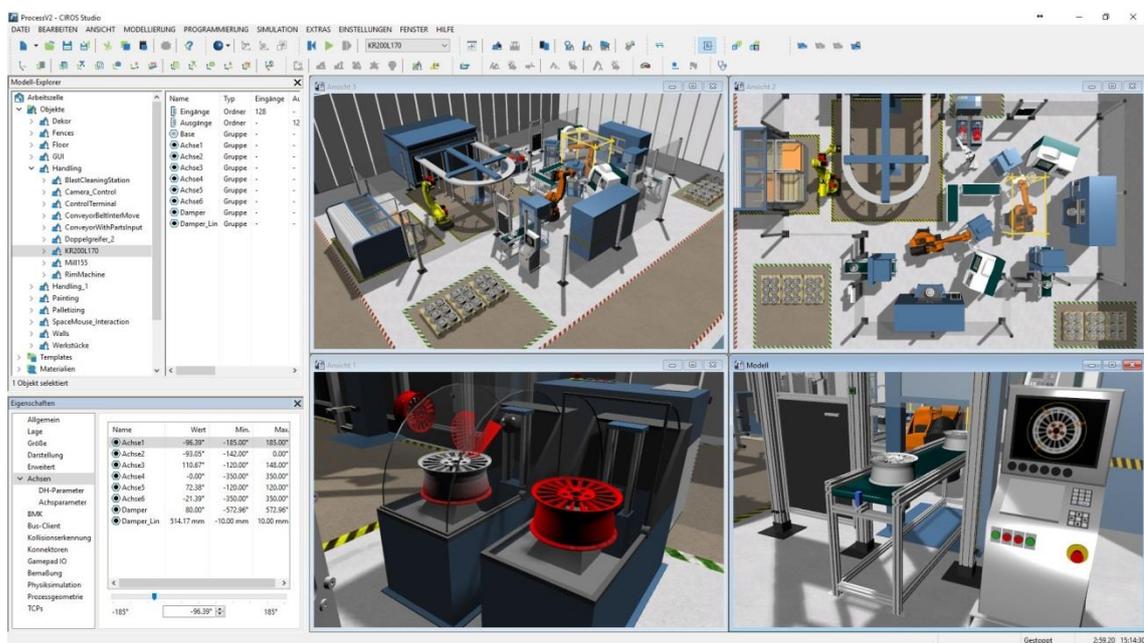


Figura 8.- Ciros [6].

Se puede utilizar en términos generales cualquier PLC para el control del modelo simulado. Por lo tanto, puede elegir el fabricante y el tipo de PLC y nosotros le recomendaremos el mejor medio para conectarlo al entorno virtual de aprendizaje.

Se trata de un software de simulación de PLC con un precio dependiente de la versión que se desee (hay versiones de diversa complejidad). La ULL posee una licencia de la versión CIROS MECHATRONICS, que en su momento costó 1800€. Esta versión permite solo controlar modelos 3D de las estaciones educativas Festo, pero no ofrece herramientas para la creación de plantas industriales propias. Además, es un software que consume muchísima memoria y es inestable.

4.3.- Selección.

Tras analizar las ventajas e inconvenientes de cada una de las opciones anteriormente descritas, los softwares elegidos son los siguientes:

- OpenPLC: software de simulación de PLC abierto, gratuito y sencillo.
- Factory I/O: software de simulación 3D de plantas industriales económico, del que la ULL ya dispone de licencias.

El objetivo es comprobar si la unión de estos programas puede ser utilizada en las prácticas de las asignaturas de Automatización de la Escuela. En los siguientes apartados se describirán estos programas en más detalle.

5.- Factory I/O.

El Factory IO [1] es un software de simulación 3D de una fábrica, que permite aplicar técnicas de automatización sobre plantas simuladas. Ha sido diseñado para poder trabajar de una manera fácil sobre algunos de los escenarios más comunes que se pueden encontrar en las instalaciones de una industria.



Figura 9.- Escena de Factory I/O [1].

El Factory IO permite al usuario diseñar sus propios modelos 3D y guardarlos en “escenas”. También ofrece algunas escenas prediseñadas. A continuación, se describirán los aspectos más importantes del programa.

5.1.- Entorno de trabajo.

El entorno que permite el diseño de escenas y su simulación se muestra en la figura 10. El primer elemento de la barra de herramientas es el menú de “bienvenida” (con el icono de una fábrica), en el que se despliega una lista de opciones que da acceso a documentación, tutoriales, escenas predeterminadas o personalizadas grabadas previamente, etc. El menú de “archivo” permite crear, abrir y guardar escenas. El menú de “edición” es el que se usa para editar escenas (permite seleccionar, deshacer, rehacer, copiar, pegar...). Por último, el menú de “vista” se utiliza para mostrar diferentes ventanas en el entorno: lista de elementos que se puede poner en la escena (que se denomina “paleta”), el tipo de cámara para ver la escena, visualización de las etiquetas de sensores y actuadores, etc.

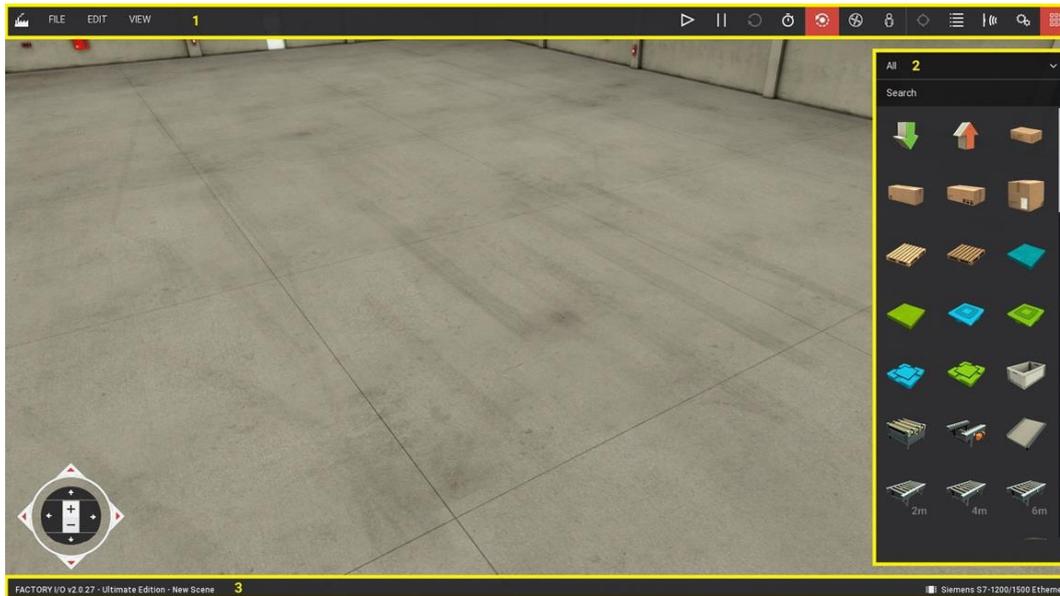


Figura 10.- Entorno de trabajo [1].

En la parte superior derecha del entorno se muestran los botones enumerados en la figura XI. Estos botones se usan para controlar la simulación del modelo.

Botón	Acción
	Ejecutar/Editar Cambia el modo de simulación (Editar/Ejecutar).
	Pausa Pausa la simulación.
	Restablecer Restablece la simulación.
	Seguir una Parte Permite seleccionar una parte a seguir.
	Escala de tiempo Establece la escala de tiempo de la simulación (cámara lenta, velocidad normal o avance rápido). Cuando está activo, muestra un control deslizante para cambiar el valor de la escala de tiempo.
	Cámara orbital Selecciona la cámara orbital.
	Fly Camera Selecciona la cámara Fly.
	Cámara en primera persona Selecciona la cámara en primera persona.
	Ventana de cámaras Abre la ventana de cámaras.
	Etiquetas del sensor Muestra/oculta las etiquetas del sensor.
	Etiquetas del actuador Muestra/oculta las etiquetas del actuador.
	Paleta Muestra/oculta la paleta.

Figura 11.- Botones del entorno y acciones asociadas [1].

5.2.- Paleta.

La paleta es una parte importante del programa, ya que contiene todos los elementos disponibles para crear la simulación. La única acción que se debe realizar es arrastrar la pieza elegida al espacio de simulación para crear una escena en 3D e ir construyendo la planta industrial que se pretende simular. Se puede desplegar la paleta en su totalidad o se puede filtrar por categorías.

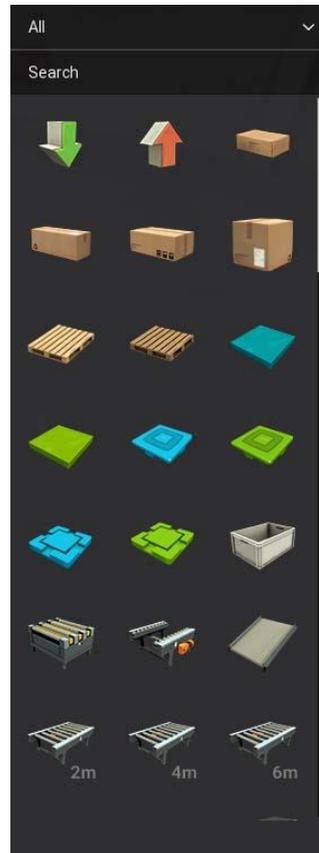


Figura 12.- Paleta [1].

A continuación, se muestran las diferentes piezas que existen y se pueden emplear para conformar la simulación en 3D de la fábrica que se desea automatizar.

5.2.1.- Elementos del modelo.

Las piezas están diseñadas acorde a los equipos más empleados a nivel industrial y se divide en las siguientes categorías:

- Artículos.
- Carga pesada.
- Carga ligera.
- Sensores
- Pulsadores y botones.

- Estaciones.
- Dispositivos de advertencia.
- Pasarelas.
- Emisor y removedor.



Imagen 13.- Piezas de Factory I/O [1].

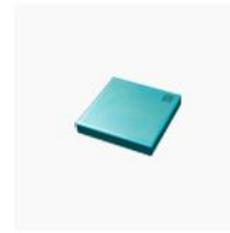
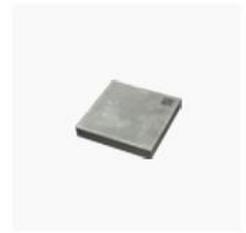
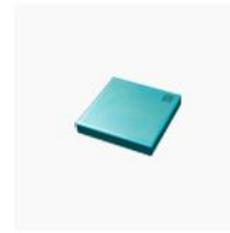
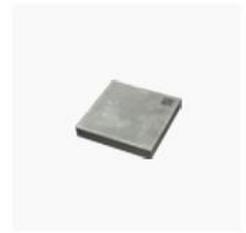
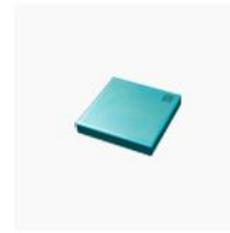
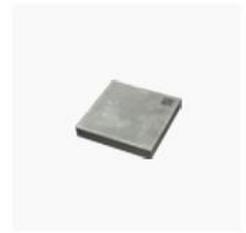
5.2.1.1.- Artículos.

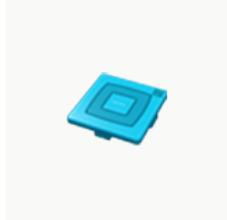
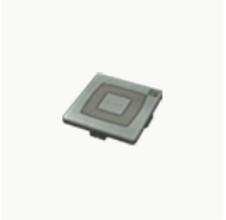
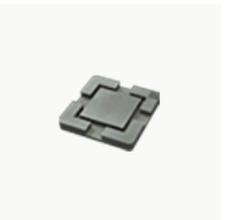
En este apartado se pueden distinguir las piezas que se pretende transportar o fabricar a lo largo de la simulación definida por el usuario.



Figura 14.- Artículos [1].

En esta categoría se pueden observar los siguientes elementos: cajas, pallets, cajas apilables que pueden transportar en su interior distintos artículos, materias primas, tapas, bases y productos finales (tabla 1).

Artículo	Imagen								
Cajas	<table border="1"> <tr> <td data-bbox="504 259 740 322">Pequeño 8Kg</td> <td data-bbox="746 259 983 322">Medio 10 Kg</td> <td data-bbox="989 259 1225 322">Grande 15Kg</td> <td data-bbox="1232 259 1394 322">Caja Paletizador 3 Kg</td> </tr> <tr> <td data-bbox="504 331 740 555"></td> <td data-bbox="746 331 983 555"></td> <td data-bbox="989 331 1225 555"></td> <td data-bbox="1232 331 1394 555"></td> </tr> </table>	Pequeño 8Kg	Medio 10 Kg	Grande 15Kg	Caja Paletizador 3 Kg				
Pequeño 8Kg	Medio 10 Kg	Grande 15Kg	Caja Paletizador 3 Kg						
									
Pallets	<table border="1"> <tr> <td data-bbox="504 586 772 649">Palet Cuadrado 20 Kg</td> <td data-bbox="778 586 1046 649">Palet 20 Kg</td> </tr> <tr> <td data-bbox="504 658 772 936"></td> <td data-bbox="778 658 1046 936"></td> </tr> </table>	Palet Cuadrado 20 Kg	Palet 20 Kg						
Palet Cuadrado 20 Kg	Palet 20 Kg								
									
Caja apilable	<table border="1"> <tr> <td data-bbox="504 976 772 1039">Caja Apilable 15 Kg</td> </tr> <tr> <td data-bbox="504 1048 772 1326"></td> </tr> </table>	Caja Apilable 15 Kg							
Caja Apilable 15 Kg									
									
Materia prima	<table border="1"> <tr> <td data-bbox="504 1366 772 1429">Materia Prima Azul 8 Kg</td> <td data-bbox="778 1366 1046 1429">Materia Prima Verde 8 Kg</td> <td data-bbox="1053 1366 1337 1429">Materia Prima Metálica 10 Kg</td> </tr> <tr> <td data-bbox="504 1438 772 1715"></td> <td data-bbox="778 1438 1046 1715"></td> <td data-bbox="1053 1438 1337 1715"></td> </tr> </table>	Materia Prima Azul 8 Kg	Materia Prima Verde 8 Kg	Materia Prima Metálica 10 Kg					
Materia Prima Azul 8 Kg	Materia Prima Verde 8 Kg	Materia Prima Metálica 10 Kg							
									

<p>Producto tapa</p>	<p>Producto Tapa Azul 5 Kg</p> 	<p>Producto Tapa Verde 5 Kg</p> 	<p>Tapa Metálica Producto 7 Kg</p> 
<p>Producto base</p>	<p>Producto Base Azul 7 Kg</p> 	<p>Producto Verde Base 7 Kg</p> 	<p>Producto Base Metalica 9 Kg</p> 

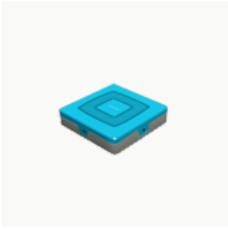
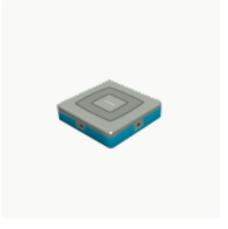
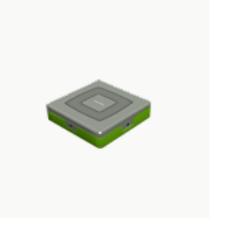
Producto final	Tapa Azul - Base Azul 12 Kg	Tapa Azul - Base Verde 12 Kg	Tapa Azul - Base Metálica 14 Kg
			
	Tapa Verde - Base Azul 12 Kg	Tapa Verde - Base Verde 12 Kg	Tapa Verde - Base Metálica 14 Kg
			
	Tapa Metálica - Base Azul 14 Kg	Tapa Metálica - Base Verde 14 Kg	Tapa Metálica - Base Metálica 16 Kg
			

Tabla 1.- Artículos. Figuras extraídas de [1].

5.2.1.2.- Carga pesada.

En esta categoría están contempladas las piezas necesarias para el manejo de cargas pesadas: transportador de rodillos, transportador de rodillos curvo, transportador de carga, tope de rodillos, rodillo libre, transferencia de cadena, placa giratoria, y transportador de canal bajo. Se pueden observar estos elementos en la tabla 2.

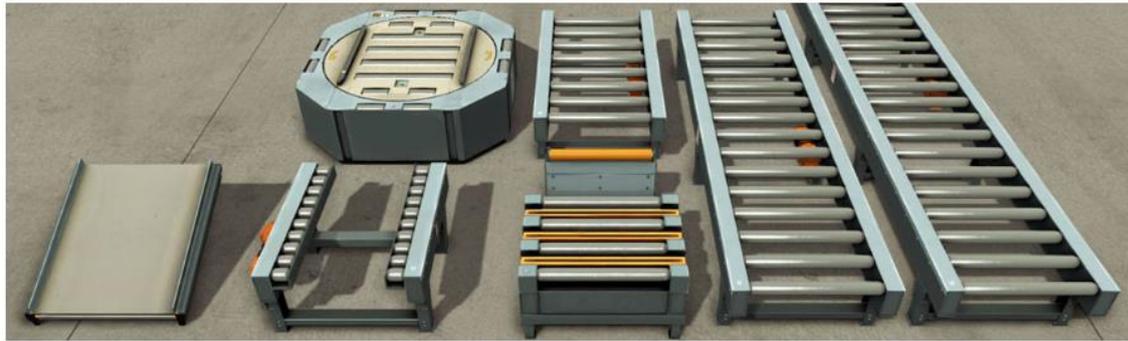
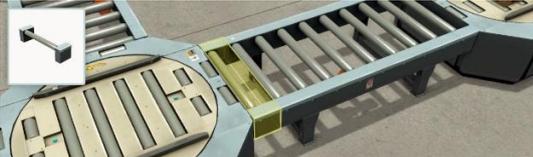


Figura 15.- Carga pesada [1].

Elemento	Descripción	Imagen
Transportador de rodillos	Este elemento puede ser tanto analógico como digital, dependiendo de la configuración establecida por el usuario.	
Transportador de rodillos curvos	Este elemento puede ser tanto analógico como digital, dependiendo de la configuración establecida por el usuario.	
Transportador de carga	Este elemento puede ser tanto analógico como digital, dependiendo de la configuración establecida por el usuario y se emplea para cargar o descargar en una grúa apiladora.	
Tope de rodillos	Este dispositivo se acciona de forma neumática con el objetivo de detener, acumular o evitar colisiones de materiales a lo largo de la simulación.	
Rodillo libre	Se trata de un rodillo no motorizado que se emplea para ajustar los transportadores de rodillos.	

<p>Transferencia de cadena</p>	<p>Este dispositivo se emplea para realizar la transferencia de la carga, desde un transportador a otro, de forma longitudinal mediante rodillos o de forma transversal mediante tres pistas de cadena.</p>	
<p>Placa giratoria</p>	<p>Este dispositivo se emplea para realizar la transferencia de la carga, desde un transportador a otro, de forma longitudinal o transversal, mediante rodillos en una plataforma giratoria.</p>	
<p>Transportador de canal bajo</p>	<p>Se trata de una rampa por la que se desliza la carga desde un transportador desde un nivel superior a un nivel inferior.</p>	

Tabla 2.- Carga pesada. Figuras extraídas de [1].

5.2.1.3.- Carga ligera.

En esta categoría están contempladas las piezas necesarias para el manejo de cargas ligeras: transportador de banda, transportador de banda curva, compuerta de cinta transportadora, transportador de cinta inclinada, báscula transportadora, transportador recto-recto, clasificador de brazo pivotante, clasificador de ruedas desplegable, alineadores, clasificador de empujador neumático, transportador de tolva, detección de cuchilla, soporte y esquina de metal y barras de posicionamiento. Se pueden observar estos elementos en la tabla 3.



Figura 16.- Carga ligera [1].

Elemento	Descripción	Imagen
Transportador de banda	Son transportadores de banda encargados de transportar cargas ligeras, y puede ser controlado de forma digital o analógica.	
Transportador de banda curva	Son transportadores de banda encargados de transportar cargas ligeras, y puede ser controlado de forma digital o analógica.	
Compuerta de cinta transportadora	Se trata de un elemento motorizado que se eleva para proporcionar paso al personal, y puede ser controlado de forma digital o analógica.	
Transportador de cinta inclinada	Se trata de un elemento que transporta la carga desde un nivel superior a un nivel inferior, y puede ser controlado de forma digital o analógica.	
Báscula transportadora	Se trata de un elemento que dispone de un sensor de peso, midiendo distintos rangos según la configuración del usuario.	

<p>Transportador recto-recto</p>	<p>Compuesto por múltiples correas, su propósito es combinarse entre transportadores de cinta. Puede ser controlado de forma digital o analógica.</p>	
<p>Clasificador de brazo pivotante</p>	<p>Se trata de un elemento que se acciona a 45° y su fin es desviar los artículos transportados por la cinta. Puede girar a la derecha y a la izquierda según la configuración del usuario.</p>	
<p>Clasificador de ruedas desplegable</p>	<p>Se trata de un elemento formado por ruedas emergentes de doble cara de 45° y su fin es desviar los artículos transportados por la cinta a tres direcciones diferentes.</p>	
<p>Alineadores</p>	<p>Se trata de estructuras metálicas que se acoplan a un transportador para evita la caída de piezas. Existen 4 tipos de alineadores diferentes.</p>	
<p>Clasificador de empujador neumático</p>	<p>Se trata de un elemento equipado con dos sensores que indican los límites delantero y trasero.</p>	
<p>Transportador de tolva</p>	<p>Se trata de una rampa por la que se desliza la carga desde un transportador desde un nivel superior a un nivel inferior.</p>	

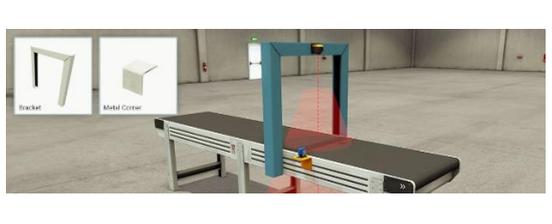
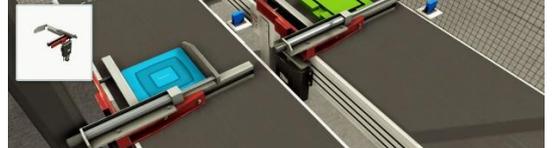
<p>Detección de cuchilla</p>	<p>Dispositivo que se acciona neumáticamente para detener o acumular artículos.</p>	
<p>Soporte y esquina de metal</p>	<p>Estructuras que se emplean para fijar la altura máxima de un artículo y para fijar sensores.</p>	
<p>Barras de posicionamiento</p>	<p>Dispositivo que se utiliza para posicionar artículos en el mismo lugar.</p>	

Tabla 3.- Carga pesada. Figuras extraídas de [1].

5.2.1.4.- Sensores.

En esta categoría están contemplados los sensores para la detección, medición e identificación de piezas: sensor capacitivo, difuso, inductivo, matriz de luces, sensor retrorreflectante y de visión. Se pueden observar estos elementos en la tabla 4.

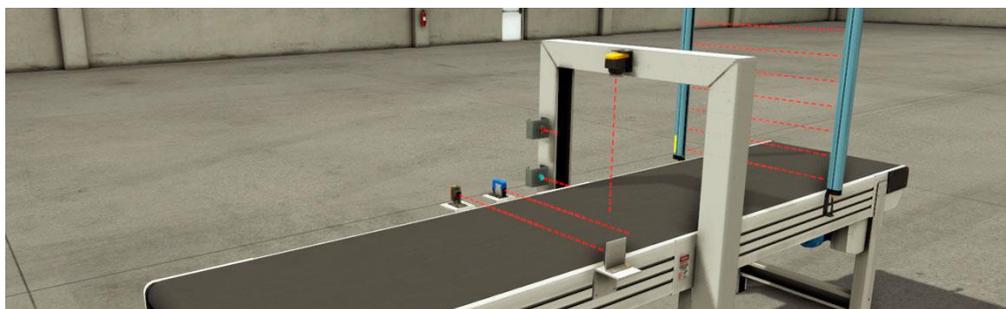


Figura 17.- Sensores [1].

Elemento	Descripción	Imagen
Sensor capacitivo	Sensor de proximidad que es utilizado para la detección de material a corta distancia. El valor puede ser digital o analógico dependiendo de la configuración del usuario.	
Sensor difuso	Sensor fotoeléctrico cuyo objetivo es detectar cualquier material sólido.	
Sensor inductivo	Sensor de proximidad para detectar materiales conductores en un rango corto de distancia. El valor puede ser digital o analógico dependiendo de la configuración del usuario.	
Matriz de luces	Formado por un conjunto de haces de luz paralelos que crean un campo de detección. Puede trabajar en modo numérico, digital o analógico, según la configuración del usuario.	
Sensor retroreflectante	Se trata de un sensor que necesita la presencia de un reflector alineado con dicho sensor, para que funcione correctamente.	
Sensor de visión	Se trata de un sensor que reconoce los artículos de materias primas, bases y tapas de productos, y sus colores correspondientes.	

Tabla 4.- Sensores. Figuras extraídas de [1].

5.2.1.5.- Pulsadores y botones.

En esta categoría están contemplados los distintos elementos para operar en la simulación que se ha configurado: parada de emergencia, pulsadores, indicadores luminosos, selector, potenciómetro, pantalla digital y cuadro eléctrico. Se pueden observar estos elementos en la tabla 5.



Figura 18.- Operadores [1].

Elemento	Descripción	Imagen
Parada de emergencia	Se trata de un pulsador de dos posiciones, no iluminado, cuyo uso normal es para condiciones de emergencia.	
Pulsadores	Existen tres tipos de pulsadores luminosos: verde para la marcha, rojo para la parada y amarillo para el reset.	
Indicadores luminosos	Indicadores de luz para montaje en el panel de control. Su fin es la indicación de estado del proceso productivo y están disponibles en cuatro colores.	
Selector	Se utiliza principalmente para establecer el estado del proceso.	

<p>Potenciómetro</p>	<p>Se utiliza para poder generar un valor analógico dependiendo del giro que se le establezca.</p>	
<p>Pantalla digital</p>	<p>Este dispositivo permite la visualización de los valores numéricos establecidos por el usuario para la simulación. correspondientes.</p>	
<p>Cuadro eléctrico</p>	<p>Se emplea como panel de control para colocar los operadores.</p>	

Tabla 5.- Pulsadores y botones. Figuras extraídas de [1].

5.2.1.6.- Dispositivos de advertencia.

En esta categoría están contemplados los distintos dispositivos de advertencias que se pueden utilizar para operar en la simulación que se ha configurado: sirena de alarma, stack light y luz de alerta. Se pueden observar estos elementos en la tabla 6.

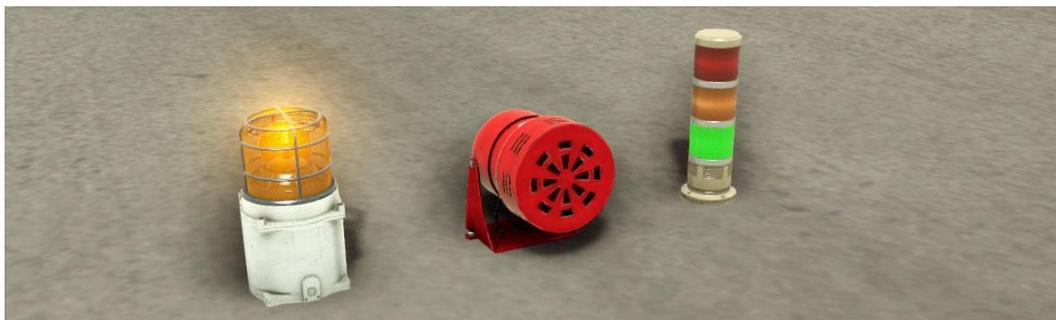


Figura 19.- Dispositivos de advertencia [1].

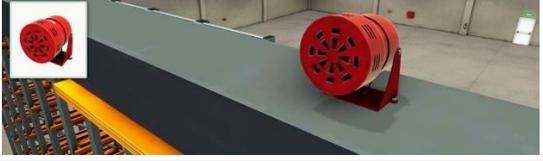
Elemento	Descripción	Imagen
Sistema de alarma	Este dispositivo es audible debido a su alto volumen.	
Stack light	Este dispositivo se emplea como un indicador visual de los estados del proceso que se ha simulado, mediante tres colores distintos: verde, amarillo y rojo.	
Indicadores luminosos	Se trata de una lámpara giratoria que se enciende según la configuración del usuario.	

Tabla 6.- Dispositivos de advertencia. Figuras extraídas de [1].

5.2.1.7.- Pasarelas.

En esta categoría están contempladas las distintas piezas que se pueden utilizar para delimitar o dar acceso a los operadores a las distintas zonas de la simulación que se ha configurado: pasamanos, plataformas, rejas, escaleras, y puertas de seguridad. Se pueden observar estos elementos en la tabla 7.



Figura 20.- Pasarelas [1].

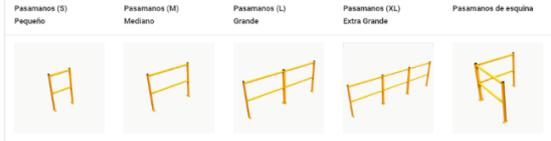
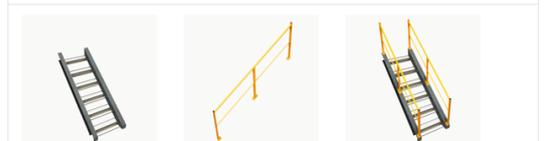
Elemento	Descripción	Imagen
Pasamanos	Estructuras de metal utilizadas para imponer seguridad a los trabajadores.	
Pulsadores	Estructuras metálicas utilizadas para crear suelos a la altura estipulada por el usuario.	
Rejas	Estructuras metálicas de rejilla utilizadas para delimitar zonas de trabajo.	
Escaleras	Estructuras metálicas para acceder a distintos niveles de altura.	
Puertas de seguridad	Se trata de una puerta de seguridad que posee un interruptor de enclavamiento incorporado.	

Tabla 7.- Pasarelas. Figuras extraídas de [1].

5.2.1.8.- Emisor y removedor.

En esta categoría se encuentra el emisor y removedor, en los cuales se comienza o se termina una escena. En ellos se introducen o se expulsan los diferentes artículos configurados para la simulación del proceso industrial. Se puede elegir el artículo, el tiempo, la cantidad, posición u orientación.



Figura 21.- Emisor y removedor [1].

5.2.1.9.- Estaciones.

En esta categoría están contempladas las distintas estaciones que se pueden utilizar para operar en la simulación que se ha configurado.

Se pueden observar los siguientes elementos:

- Centro de mecanizado.

Esta estación se utiliza para la fabricación de tapas y bases de producto mediante la introducción de materias primas. El robot recoge la materia prima de la bandeja de entrada y lo deposita en el centro de mecanizado, donde se producen las tapas (6 segundos) o las bases (3 segundos) y finalmente el robot traslada el producto a la bandeja de salida.



Figura 22.- Estaciones – Centro de mecanizado [1].

- Ascensor.

Este dispositivo se emplea para el transporte de cualquier carga a cualquier nivel de altura necesario por el usuario, y está equipado con dos sensores retrorreflectantes en cada uno de los extremos del transportador de rodillos. Puede ser controlado mediante valores digitales o analógicos, dependiendo de la configuración del usuario.

- Radio de rodillo: 45 mm.

- Carrera de la plataforma: 7 m.
- Velocidad de la plataforma: 0,68 m/s.
- Velocidad máxima de transporte: 0,45 m/s (digital); 0,8 m/s (analógico).
- Sensor retrorreflectante LED: verde (reflector de detección), amarillo (haz no interrumpido).



Figura 23.- Estaciones – Ascensor [1].

- Pick & Place.

Esta estación posee tres ejes controlados por servomotores y se emplea normalmente para desplazar cargas ligeras desde un transportador a otro. Tiene cuatro grados de libertad correspondientes al movimiento lineal de xyz y a la rotación de las pinzas. Dichas pinzas poseen ventosas y un sensor de proximidad, pudiendo ser controlado de forma digital o analógica.

- Carrera del eje Y: 1,25 m.
- Carrera del eje X: 2,125 m.
- Carrera del eje Z: 0,5 m.
- Paso: 0,125 m
- Velocidad del haz: 1,5 m/s.
- Velocidad angular de la pinza: 4,6 rad/s.

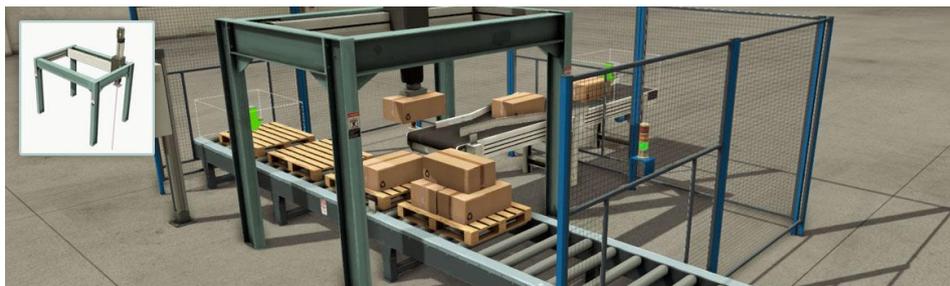


Figura 24.- Estaciones – Pick & Place [1].

- Grúa apiladora y estantes.

Esta estación se utilizar para almacenar carga pesada en las estanterías. Está formado por un carro, una plataforma vertical y dos cadenas que pueden desplazarse en ambos sentidos.

La posición horizontal y vertical de la plataforma es medida por dos telémetros láser y las cargas se pueden almacenar desde ambos lados de la estantería. Puede ser controlado mediante valores digitales o analógicos, dependiendo de la configuración del usuario.

- Carrera de cadenas: 1,3 m.
- Carrera del carro: 10,5 m.
- Carrera de la plataforma: 6,625 m.
- Velocidad del carro: 1,4 m/s.
- Velocidad cadenas: 0,5 m/s.
- Velocidad de la plataforma: 1,7 m/s.
- Número de celdas: 18.



Figura 25.- Estaciones – Grúa apiladora y estantes [1].

- Paletizador.

Se trata de un paletizador automático empleado para apilar cajas de cartón sobre pallets.

- Carrera del empujador: 0,88 m.
- Carrera del elevador: 1,75 m.
- Velocidad del ascensor: 2 m/s.



Figura 26.- Estaciones – Paletizador [1].

▪ Pick & Place de dos ejes.

Este dispositivo se emplea para el ensamblaje de tapas en bases de producto, recogiendo en un transportador y colocándolo en otro transportador.

- Carrera del eje X: 1,125 m.
- Carrera del eje Z: 0,625 m.
- Velocidad del brazo y recogedor: 2 m/s.
- Rotación del eje Z en incrementos de 90° (brazo y pinza)



Figura 27.- Estaciones – Pick & Place de dos ejes [1].

▪ Tanque.

Se trata de un tanque de líquido en el que se incluyen dos válvulas de control y un sensor de nivel capacitivo, para controlar el nivel de líquido del interior del tanque.

- Altura: 3 m.
- Diámetro: 2 m.
- Radio de la tubería de descarga: 0,125 m.
- Caudal máximo de entrada: 0,25 m³/s.
- Caudal máximo de salida: 0,3543 m³/s.
- Los sensores capacitivos pueden detectar líquidos.



Figura 28.- Estaciones – Tanque [1].

5.3.- Navegación.

Para desplazarse y navegar por la simulación en el área 3D, se requiere conocer cómo manejar correctamente los distintos tipos de cámaras que existen: orbital, aérea y en primera persona. Como vimos anteriormente, estas cámaras se seleccionan con los botones que están en la parte superior derecha del entorno. También cabe destacar que existe una “ventana de cámaras”, que permite al usuario guardar sus propias posiciones de cámaras.

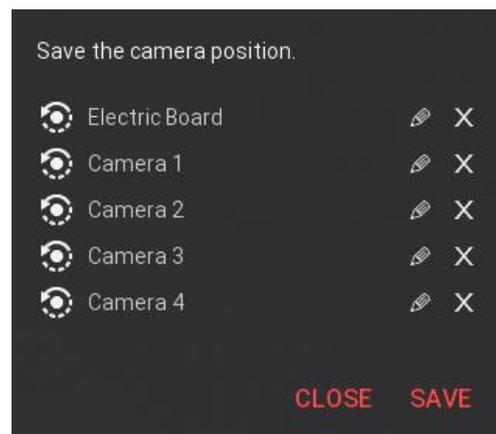


Figura 29.- Ventana de cámaras [1].

La cámara orbital ha sido diseñada para las acciones de edición, para la creación de escenas por parte del usuario. Proporciona una vista alrededor de un punto establecido por el usuario, indicado con un punto blanco en la pantalla. La cámara aérea se emplea para desplazarse libremente por el espacio 3D creado por el usuario. La cámara en primera persona representa a una persona de 1,8 metros de altura y su objetivo es simular la presencia de un operario en el proceso industrial que ha sido simulado por el usuario.

5.4.- Etiquetas de sensores y actuadores.

Cada uno de los sensores o actuadores introducidos en el modelo 3D está identificado por una etiqueta compuesta por un nombre y un valor. Al introducir una pieza en la escena, esta etiqueta se crea automáticamente, pero se puede cambiar de forma sencilla pinchando sobre ella en el entorno. Dichas etiquetas se pueden mostrar u ocultar con unos botones que están en la parte superior derecha, como se indicó en la figura 30.

Existen tres tipos de etiquetas: Bool, Float e Int, que indica el tipo de dato que maneja ese sensor o actuador. Se pueden forzar valores en los sensores y en los actuadores por medio de estas etiquetas.

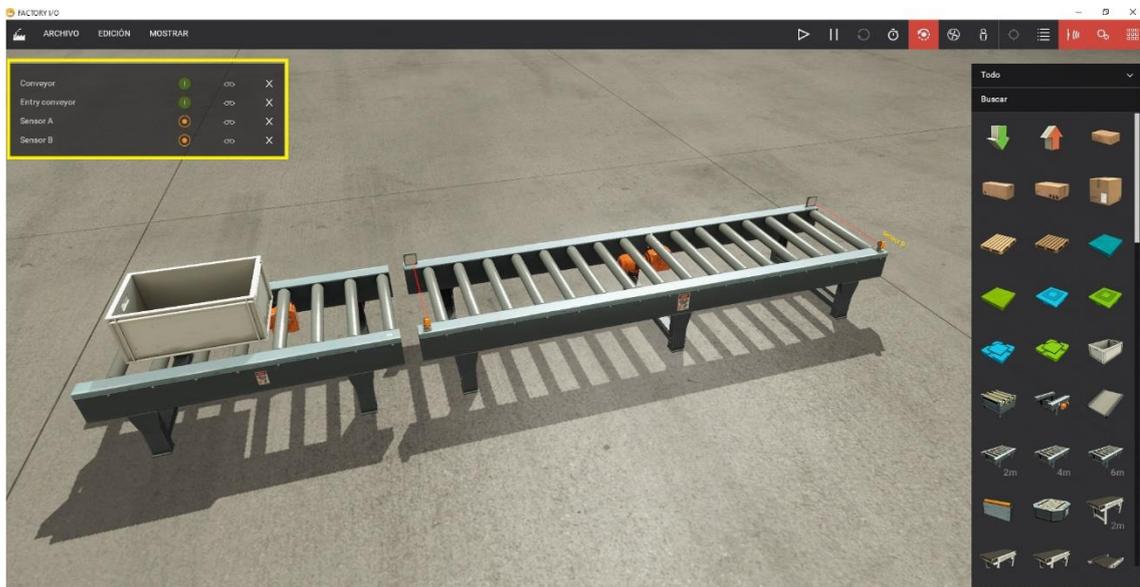


Figura 30.- Etiquetas de sensores y actuadores [1].

5.5.- Escenas predeterminadas.

En Factory I/O se han generado escenas predeterminadas para que el usuario pueda practicar e incluso tomarlas como comienzo para generar una escena personalizada añadiendo o eliminando elementos de dicha escena.

Las escenas que existen por defecto en Factory I/O son las siguientes:

- Cinta transportadora.
- Doble cinta transportadora.
- Tanque de llenado.
- Entrada de artículos.

- Ensamblador.
- Ensamblador analógico.
- Almacén automatizado.
- Estación de almacenaje.
- Estación convergente.
- Elevador avanzado.
- Elevador básico.
- Control de nivel.
- Paletizado.
- Recoger y colocar (Básico).
- Recoger y colocar (XYZ).
- Línea de producción.
- Estación separadora.
- Clasificación por altura (Básico).
- Clasificación por altura (Avanzado).
- Clasificación por peso.
- Estación de clasificación.

A continuación, se expone un breve resumen de cada una de las escenas predeterminadas.

5.5.1.- Cinta transportadora.

Transporte de una caja apilable, a través de una cinta transportadora de rodillos, hasta que es detenida al ser detectada por un sensor situado al final de dicha cinta transportadora.



Figura 31.- Cinta transportadora [1].

5.5.2.- Doble cinta transportadora.

Transporte de una caja apilable, a través de una cinta transportadora de rodillos, hasta que sea detectada por un sensor situado al comienzo de otra cinta transportadora de rodillos. Entonces se activará con dicho sensor, el motor de la cinta transportadora hasta que dicha caja sea detectada por otro sensor situado al final de dicha cinta transportadora.



Figura 32.- Doble cinta transportadora [1].

5.5.3.-Tanque de llenado.

Llenado y vaciado de un tanque mediante la utilización de temporizadores, que activarán las válvulas necesarias para que el líquido que se encuentra en el interior sea introducido o sacado de dicho tanque.



Figura 33.- Tanque de llenado [1].

5.5.4.- Entrada de artículos.

Se activa la entrada de artículos a una cinta transportadora de rodillos. Dicho artículo avanza a lo largo de la cinta, hasta llegar al final donde el artículo se desplaza por una rampa y se activa la salida del artículo.



Figura 34.- Entrada de artículos [1].

5.5.5.- Ensamblador.

En una cinta transportadora se activa la salida de bases mientras que en la otra cinta transportadora se activa la salida de tapas. Cuando ambas llegan a la barrera establecida, son detectados por un robot que recoge la tapa y la ensambla en la base de la cinta contigua.

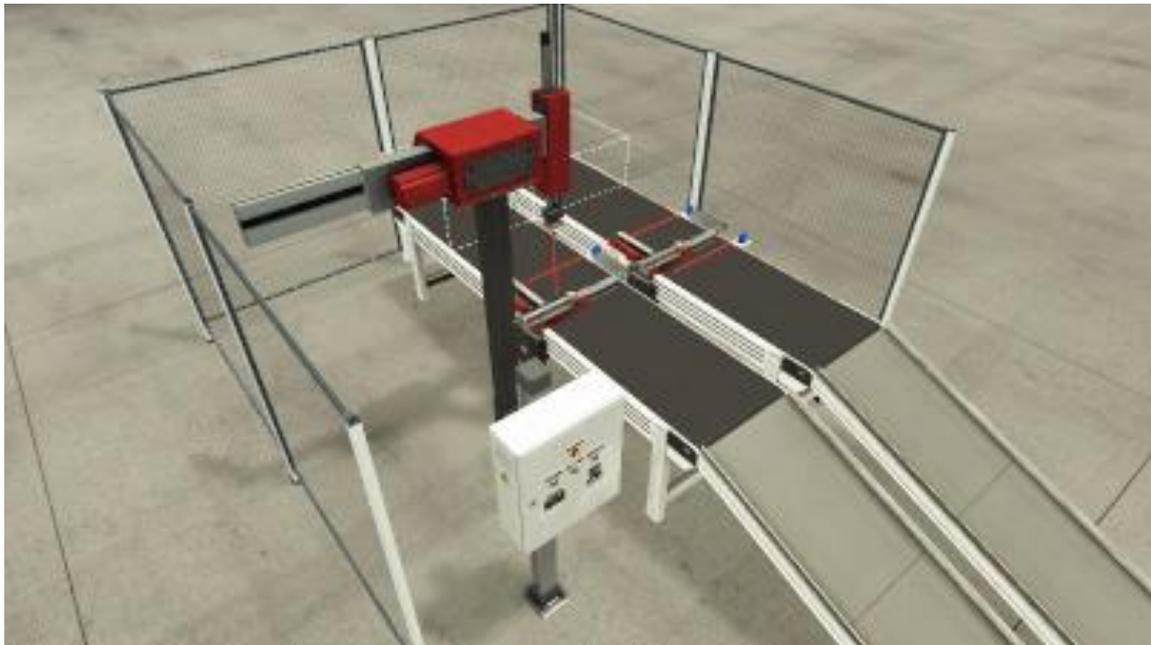


Figura 35.- Ensamblador [1].

5.5.6.- Ensamblador analógico.

En una cinta transportadora se activa la salida de bases mientras que en la otra cinta transportadora se activa la salida de tapas. Cuando ambas llegan a una zona determinada de las cintas, se activa una barrera para detener, las bases y las tapas, para ensamblarlos.



Figura 36.- Ensamblador analógico [1].

5.5.7.- Almacén automatizado.

Se activa la entrada de un artículo, para posteriormente ser archivado en el almacén. También se puede recuperar un artículo almacenado para su salida de dicho almacén.



Figura 37.- Almacén automatizado [1].

5.5.8.- Estación de almacenaje.

Se activa la salida de artículos y se van acumulando en la estación de almacenaje mediante la subida de una barrera. Posteriormente, la barrera se abre para dejar pasar de uno en uno, un artículo a la siguiente cinta transportadora y cuando dicho artículo llega al final de dicha cinta transportadora, se activa la salida y se abre la barrera para dejar pasar otro artículo.



Figura 38.- Estación de almacenaje [1].

5.5.9.- Estación convergente.

Se activan dos salidas de artículos en dos cintas transportadoras de rodillos. Cuando llegan a la zona de convergencia a otra cinta transportadora de rodillos, se establece la prioridad para el paso de los artículos.



Figura 39.- Estación convergente [1].

5.5.10.- Elevador básico.

Se activa la salida de los artículos, los cuáles, mediante dos elevadores, van ascendiendo a los demás niveles en el primer elevador, y descendiendo en el segundo elevador.



Figura 40.- Elevador básico [1].

5.5.11.- Elevador avanzado.

Se activa la salida de una caja apilable en el nivel inferior. Cuando llega al elevador, comienza a subir, deteniéndose en cada nivel. En cada uno de los niveles existe una cinta transportadora que desplaza una pieza que se deposita en la caja apilable. Cuando es depositada la pieza, la caja asciende al siguiente nivel. Finalmente desciende hasta el nivel inferior para activarse la salida al final de la cinta.



Figura 41.- Elevador avanzado [1].

5.5.12.- Control de nivel.

Se realiza el control del nivel de líquido de un tanque, teniendo en cuenta la consigna establecida por el usuario. Dependiendo de dicha consigna las válvulas se abren para introducir o sacar líquido del tanque.



Figura 42.- Control de nivel [1].

5.5.13.- Paletizado.

Se desplazan a lo largo de una cinta transportadora cada artículo. Son acumulados tres artículos y son desplazados mediante un empujador neumático hacia el paletizado, donde se abre una compuerta para que se coloquen en el pallet. Entonces se vuelven a colocar otros tres artículos para ser empujados hacia el pallet. Se repiten estas acciones hasta completar el pallet y finalmente dicho pallet desciende hasta una cinta transportadora de rodillos donde al final se activará la salida.



Figura 43.- Paletizado [1].

5.5.14.- Recoger y colocar (Básico).

Se activa una salida de artículos en una cinta transportadora. Cuando llega al sensor, se detiene y un robot transporta el artículo de dicha cinta transportadora a otra cinta transportadora situada paralelamente a la primera. Se activa el motor de la segunda cinta transportadora y el artículo es desplazado hasta llegar a una rampa donde es deslizado hasta que se activa la salida.



Figura 44.- Recoger y colocar (Básico) [1].

5.5.15.- Recoger y colocar (XYZ).

Se activa la salida de los artículos en una cinta transportadora, y una salida de pallets en una cinta transversal de rodillos. Cuando llegan a la zona establecida, un robot apila los artículos en el pallet, para su posterior salida.



Figura 45.- Recoger y colocar (XYZ) [1].

5.5.16.- Línea de producción.

Se activan dos salidas en dos cintas transportadoras. Posteriormente, dos brazos robots recogen los artículos y los introducen en un centro de maquinado (fabricación de tapas y bases de productos mediante materia prima). Finalmente, los brazos robots trasladan dicho artículo a otras dos cintas transportadoras hasta la salida.



Figura 46.- Línea de producción [1].

5.5.17.- Estación separadora.

Se activan dos salidas de piezas en dos cintas transportadoras distintas. El objetivo es que en una cinta se transporte las piezas azules y en otra las piezas verdes. Cuando las piezas se desplazan por las cintas transportadoras, el sensor de visión, situado en cada una de las cintas transportadoras, detecta el color de las piezas. Si la pieza está situada en la cinta transportadora correcta, sigue desplazándose por la misma cinta transportadora. En caso de que esté en la cinta de las piezas de otro color, un empujador neumático lo traslada hacia la otra cinta transportadora para que se desplace por la cinta transportadora correcta.



Figura 47.- Estación separadora [1].

5.5.18.- Clasificación por altura (Básico).

Se activa una salida de pallets con una caja encima y dependiendo de la altura de dicha caja, el pallet se traslada hacia la derecha o hacia la izquierda en la siguiente cinta transportadora, mediante transferencia de cadena.



Figura 48.- Clasificación por altura (Básico) [1].

5.5.19.- Clasificación por altura (Avanzado).

Se activa una salida de pallets con una caja encima y dependiendo de la altura de dicha caja, el pallet se traslada hacia la derecha o hacia la izquierda en la siguiente cinta transportadora mediante una placa giratoria.



Figura 49.- Clasificación por altura (Avanzado) [1].

5.5.20.- Clasificación por peso.

Se activa la salida de las diferentes cajas posibles. Dependiendo del peso de la caja, se traslada hacia una de las tres cintas transportadoras.



Figura 50.- Clasificación por peso [1].

5.5.21.- Estación de clasificación.

Se activa la salida de piezas en una cinta transportadora y, mediante un sensor se detecta el color de la pieza y se envía a las diferentes rampas de salida.



Figura 51.- Estación de clasificación [1].

5.6.- Controladores de E/S.

El controlador de E/S permite la comunicación del modelo 3D con un controlador externo. El Factory I/O permite la comunicación con autómatas de diferentes fabricantes y con varios simuladores de autómatas. Para poder elegir un controlador se debe acudir al apartado “Drivers” del menú “Archivo”.

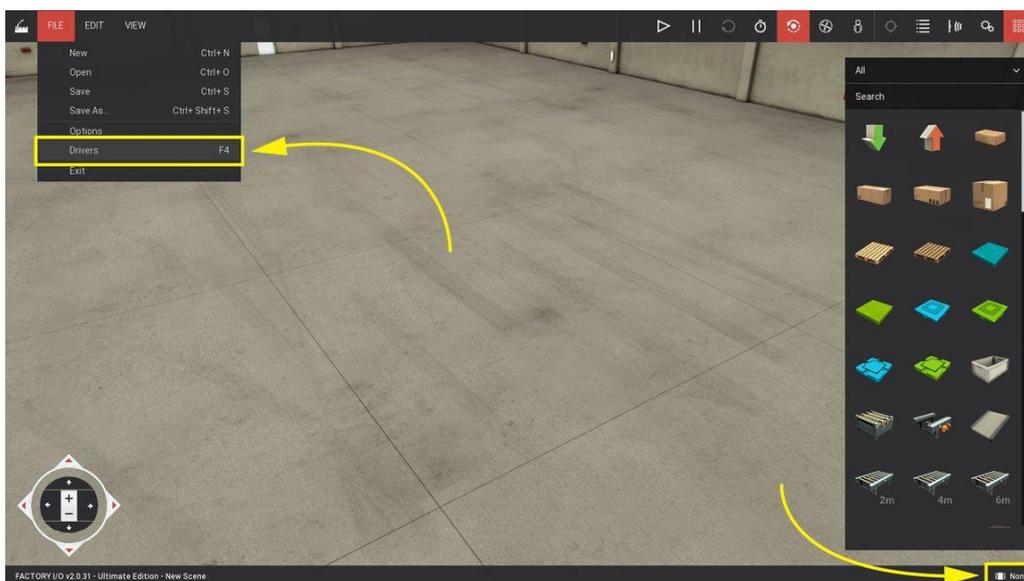


Figura 52.- Ajuste de drivers [1].

En ese apartado se despliega una lista (mostrada en la tabla VIII) donde se puede escoger entre diferentes controladores, y, posteriormente, ser configurados por el usuario.

Controlador	Descripción
Advantech USB 4750 y USB 4704	Interfaz para tarjetas de interfaz Advantech USB 4750 y 4704.
Allen Bradley Logix 5000	Conexión Ethernet a Allen-Bradley ControlLogix, CompactLogix o SoftLogix PAC.
Allen Bradley Micro800	Conexión Ethernet a PLC Allen-Bradley Micro800.
MicroLogix de Allen-Bradley	Conexión Ethernet a PLC MicroLogix de Allen-Bradley.
Allen Bradley SLC 5/05	Conexión Ethernet a PLC Allen-Bradley SLC-5/05.
Servidor automático	Interfaz para Autogen a través de un servidor TCP/IP.
E/S de control	Interfaz para Control I/O: un SoftPLC independiente de la marca, diseñado desde cero para Factory I/O.
MHJ	Interfaz para el software WinPLC-Engine y WinSPS-S7.
Cliente Modbus TCP/IP	Implementa un cliente Modbus TCP/IP.
Servidor Modbus TCP/IP	Implementa un servidor Modbus TCP/IP.
Cliente OPC DA/UA	Implementa un cliente OPC DA/UA.
LOGOTIPO Siemens!	Conexión Ethernet a Siemens LOGO! Módulo Lógico.
Siemens S7-200/300/400	Conexión Ethernet a PLC Siemens S7-200/S7-200 SMART/300/400.
Siemens S7-1200/1500	Conexión Ethernet a PLC Siemens S7-1200/1500.
Siemens S7-PLCSIM	Interfaz para Siemens S7-PLCSIM.

Tabla 8.- Controladores E/S que se pueden conectar al Factory I/O [1].

Tras la elección del controlador, debemos realizar su configuración en el apartado destinado para este efecto en la ventana de “Drivers”. Cuando especificamos el controlador, en la parte derecha se deben introducir los parámetros específicos que se solicitan para dicho controlador.

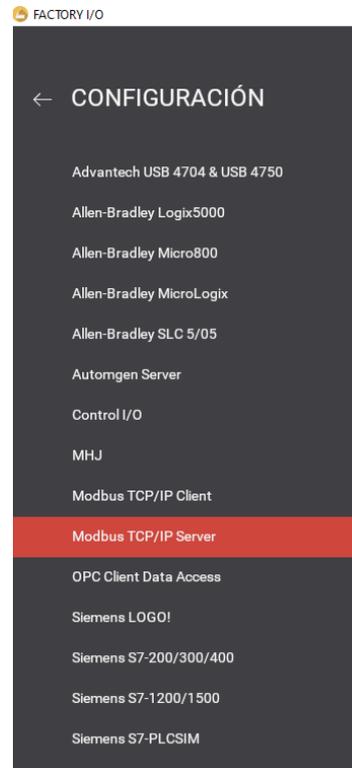


Figura 53.- Configuración de controlador [2].

En el ejemplo de la figura LIV se ha escogido como controlador el Modbus TCP/IP Server:

	Offset	Cuenta
Salidas Bool	0	3
Entradas Bool	0	1
Entradas Registro	0	0
Salidas Registro	0	0

Figura 54.- Configuración de controlador - Parámetros [2].

Se deben introducir los parámetros correspondientes a número de entradas y salidas, número de entradas y salidas de registro, identificación del esclavo, el puerto y la IP del host...

Tras realizar la configuración del controlador, ya se puede establecer la distribución de las distintas entrada y salidas como se observa en la figura 55.



Figura 55.- Configuración de drivers [2].

6.- OpenPLC.

Se trata de un simulador de controlador lógico programable (PLC) cuya característica principal es que es de código abierto, estandarizado y funcional. Cabe destacar que OpenPLC se creó siguiendo las indicaciones del estándar “IEC 61131-3” que define los lenguajes básicos de programación.

Este programa es utilizado por empresas y estudiantes a lo largo de todo el mundo para diseñar y realizar procesos de automatización industrial y del hogar, y SCADA para controlar y supervisar procesos industriales a distancia. Fue creado por Thiago Rodrigues Alves, con el objetivo de ser un vínculo entre la industria automatizada y diseñado para ser portable, y poder ser utilizado por distintas plataformas como Raspberrypi y Arduino [2].

OpenPLC es totalmente gratuito y de código abierto y es compatible para los siguientes sistemas operativos:

- Windows.
- MasOs.
- Linux.

El software OpenPLC se divide en dos partes:

- Editor.
- Runtime.

Se puede descargar en el link: <https://openplcproject.com/download/>

6.1 Editor.

Es la parte del software que permite escribir y editar los programas de PLC. Es compatible, de acuerdo con el estándar IEC 61131-3, con los siguientes lenguajes de programación:

- Ladder Logic (LD).
- Function Block Diagram (FBD).
- Instruction List (IL).
- Structured Text (ST).
- Sequential Function Chart (SFC).

6.1.1.- Entorno de trabajo.

El entorno de trabajo del OpenPLC Editor, tal y como se muestra en la figura LVI, se compone de las siguientes partes:

1. Barra de herramientas: en este espacio se pueden realizar las siguientes acciones:
 - Abrir un proyecto nuevo.

- Abrir un proyecto existente.
 - Guardar el proyecto.
 - Deshacer / Rehacer.
 - Cortar / Copiar / Pegar.
 - Comenzar la simulación.
 - Generar el programa para el Runtime.
 - Seleccionar un objeto.
 - Mover la vista.
 - Crear comentario, línea de potencia, bobina, contacto, variable, bloque o conexión (En el caso del lenguaje Ladder Logic).
2. Proyecto: se puede ver la estructura del proyecto.
 3. Variables: se dispone una lista de las variables que se utilizan en el código. La información que se debe introducir es la siguiente:
 - Nombre.
 - Clase.
 - Tipo.
 - Ubicación.
 - Valor inicial.
 - Opción.
 - Documentación.
 4. Código: este espacio está destinado a contener el código del programa que se diseña.
 5. Consola: se puede observar los fallos de compilación del código a la hora de generar el programa para el Runtime, que se deben solucionar.
 6. Librería: este apartado contiene todos los elementos que se pueden emplear para realizar el código del programa.

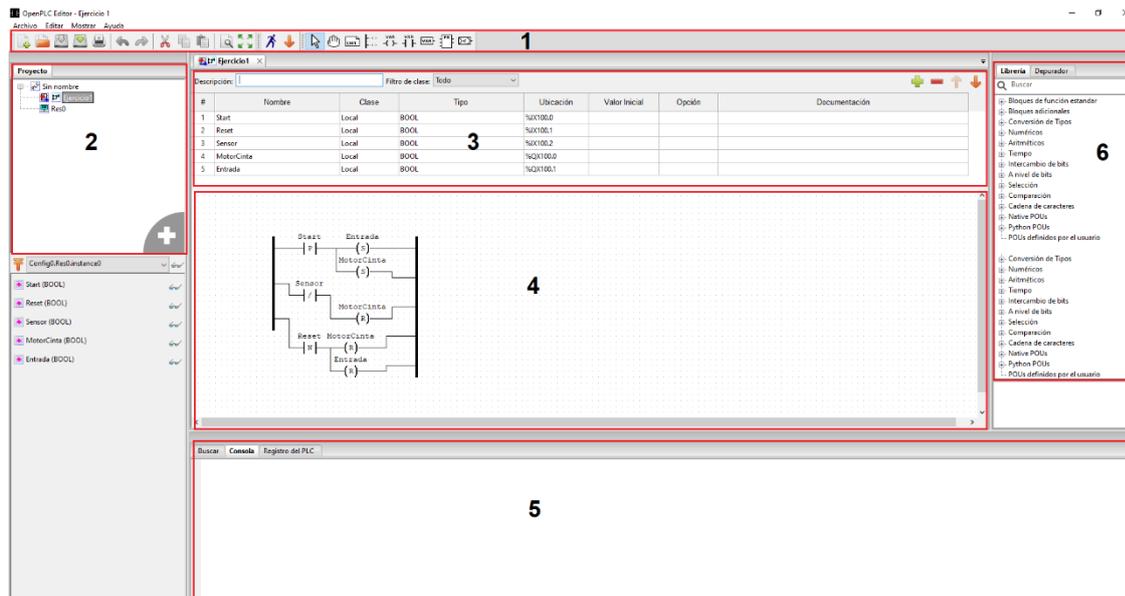


Figura 56.- Entorno de trabajo OpenPLC Editor [3].

6.1.2.- Elementos.

Los elementos disponibles en la librería, al utilizar el lenguaje Ladder Logic, son los que se nombran a continuación:

- Bloques de función estándar: enclavamientos, contadores, temporizadores...
- Bloques adicionales: PID, derivativo, integral...
- Conversión de tipos: referentes a los distintos tipos de variables (real, int, bool, time...).
- Numéricos: seno, coseno, tangente, logaritmos, raíz cuadrada...
- Aritméticos: sumar, restar, multiplicar, dividir...
- Tiempo: sumar, restar, multiplicar o dividir distintos tiempos.
- Intercambio de bits: desplazamientos y giros.
- A nivel de bits: Y, O, NO...
- Selección: máximo, mínimo, límite...
- Comparación: mayor que, menor que, igual que...
- Cadena de caracteres: insertar, eliminar, reemplazar, longitud...

Para insertar los elementos que sean necesarios para el diseño del código en lenguaje Ladder, existen varias maneras:

- **Librería:** se arrastra el elemento desde la librería hasta el espacio de trabajo.

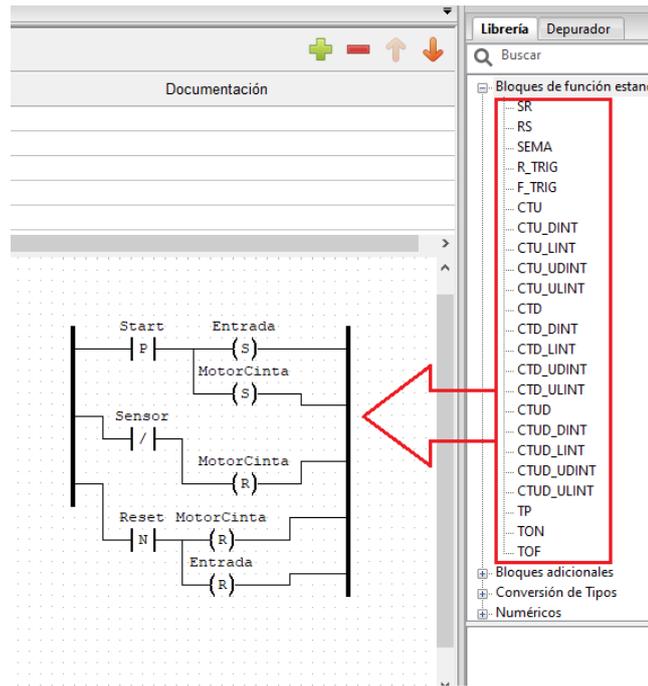


Figura 57.- Elementos – Librería [3].

- **Espacio de trabajo:** mediante el menú generado con el botón derecho del ratón se añaden los elementos necesarios.

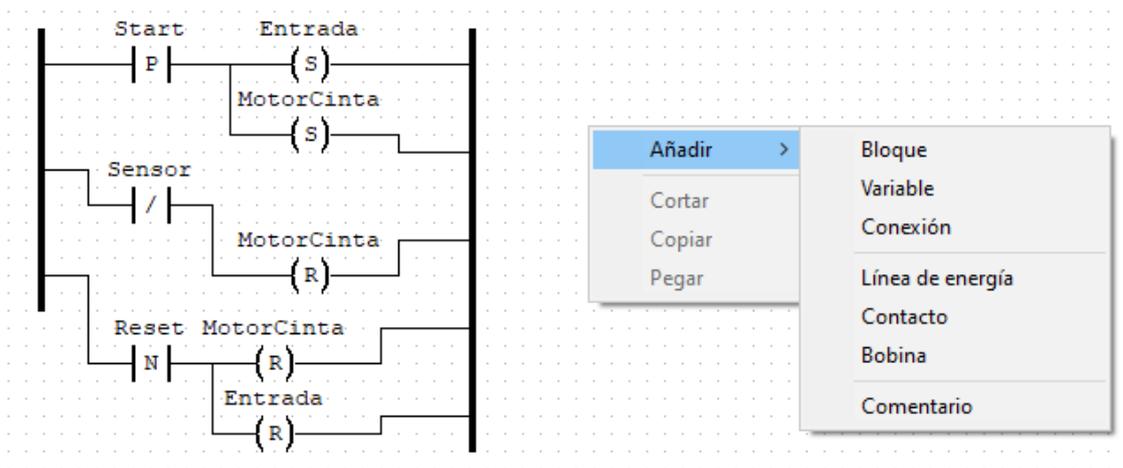


Figura 58.- Elementos – Entorno de trabajo [3].

- **Barra de herramientas:** en la parte superior del espacio de trabajo, se encuentran distintos elementos que se pueden activar para que se generen en el espacio de trabajo al pulsar el botón izquierdo del ratón.

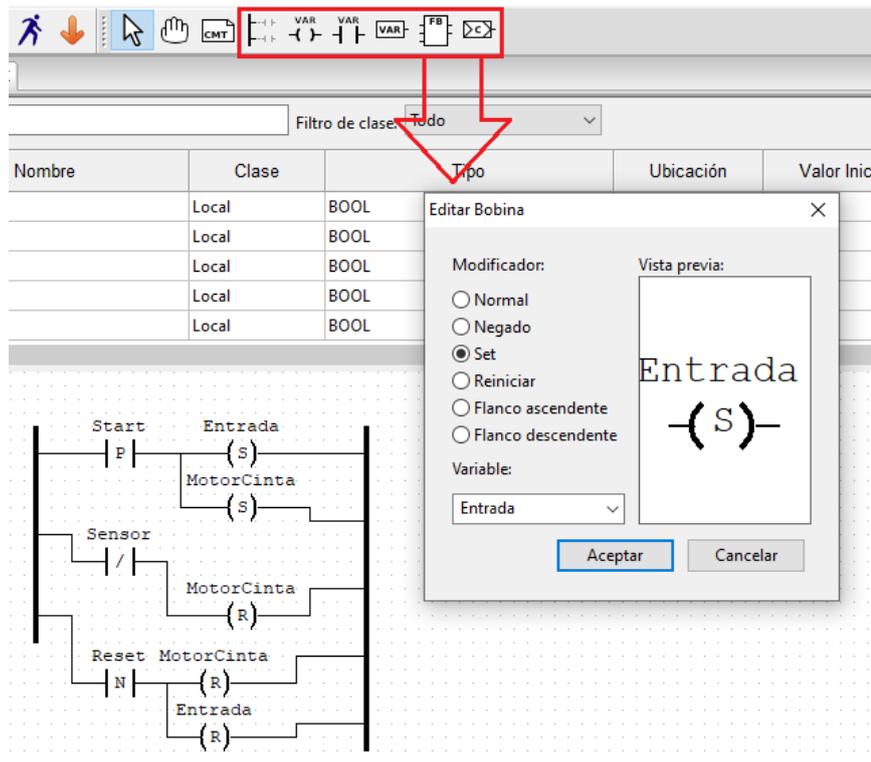


Figura 59.- Elementos – Barra de herramientas [3].

6.2.- Runtime.

Se trata de un software portátil que puede ser ejecutado en diferentes entornos, desde microcontroladores hasta potentes servidores. Es la parte responsable de simular el comportamiento de un PLC y ejecutar de manera simulada los programas de PLC que se crean en el Editor.

OpenPLC Runtime es compatible con:

- Arduino Uno/Nano/Leonardo/Micro.
- Arduino Mega / Due.
- Arduino Nano Every / IoT / BLE.
- Arduino RB2040 Connect.
- Arduino Mkr / Zero / WiFi.
- Arduino Pro (Machine Control and EDGE).
- Productivity Open P1AM)
- ESP8266 (nodemcu)
- ESP32.
- Raspberry Pi 2 / 3 / 4.
- PiXtend.

- UniPi Industrial Platform.
- Neuron PLC.
- FreeWave Zumlink.
- FreeWave ZumIQ.
- Windows (generic target as a soft-PLC).
- Linux (generic target as a soft-PLC).

En nuestro caso, empleamos este Runtime para la conexión de OpenPLC con el software de simulación de plantas industriales Factory I/O.

En su menú principal aparecen las funciones que puede realizar (figura ¿?).

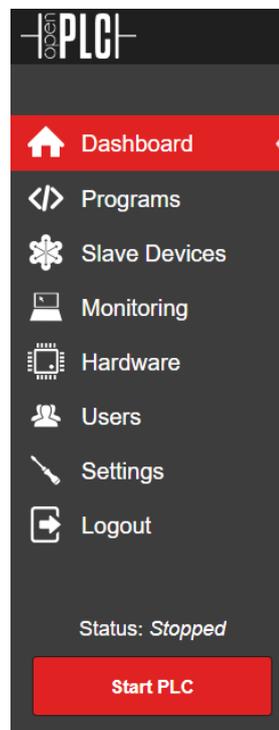


Figura 60.- OpenPLC Runtime [2].

En primer lugar, “Dashboard”, recopila los datos correspondientes al programa que se ha generado.

Dashboard

Status: Stopped

Program: Definitivo

Description:

File: 503711.st

Runtime: N/A

Figura 61.- OpenPLC Runtime - Dashboard [2].

La opción “Programs” es la que permite cargar en el simulador de PLC el programa que se ha generado en el OpenPLC Editor.

Programs

Here you can upload a new program to OpenPLC or revert back to a previous uploaded program shown on the table.

Program Name	File	Date Uploaded
Definitivo	503711.st	Jun 29, 2022 - 08:09PM

Figura 62.- OpenPLC Runtime – Programs [2].

Posteriormente, se encuentra el apartado de dispositivos conectados.

Slave Devices

List of Slave devices attached to OpenPLC.

Attention: Slave devices are attached to address 100 onward (i.e. %IX100.0, %IW100, %QX100.0, and %QW100)

Device Name	Device Type	DI	DO	AI	AO
Definitivo	TCP	%IX100.0 to %IX101.2	%QX100.0 to %QX102.3	%IW100 to %IW103	%QW100 to %QW101

Figura 63.- OpenPLC Runtime – Slave Devices [2].

Se introduce en el apartado de Slave Devices la configuración de los dispositivos conectados.

Para ello se realiza la configuración, estableciendo los siguientes datos:

- Nombre del dispositivo.
- Tipo del dispositivo.
- Identificativo del dispositivo.
- Dirección IP.
- Puerto IP.
- Número de entradas y salidas (Digitales y analógicas).

Edit slave device

<p>Device Name</p> <input type="text" value="Definitivo"/>	<p>Discrete Inputs (%IX100.0)</p> <p>Start Address: <input type="text" value="0"/> Size: <input type="text" value="11"/></p>
<p>Device Type</p> <input type="text" value="Generic Modbus TCP Device"/>	<p>Coils (%QX100.0)</p> <p>Start Address: <input type="text" value="0"/> Size: <input type="text" value="20"/></p>
<p>Slave ID</p> <input type="text" value="1"/>	<p>Input Registers (%IW100)</p> <p>Start Address: <input type="text" value="0"/> Size: <input type="text" value="2"/></p>
<p>IP Address</p> <input type="text" value="192.168.0.16"/>	<p>Holding Registers - Read (%RW100)</p> <p>Start Address: <input type="text" value="0"/> Size: <input type="text" value="2"/></p>
<p>IP Port</p> <input type="text" value="502"/>	<p>Holding Registers - Write (%QW100)</p> <p>Start Address: <input type="text" value="0"/> Size: <input type="text" value="2"/></p>

Figura 64.- OpenPLC Runtime – Edit Slave Device [2].

Finalmente, cuando se pone en funcionamiento, se pueden monitorizar las diferentes variables asociadas a la simulación en la opción “Monitoring”.

Monitoring

Refresh Rate (ms): Update

Point Name	Type	Location	Forced	Value
Start	BOOL	%IX100.0	No	FALSE
Stop	BOOL	%IX100.1	No	TRUE
Encendido	BOOL	%IX100.2	No	FALSE
Emergencia	BOOL	%IX100.3	No	TRUE
Sensor1	BOOL	%IX100.4	No	TRUE
Sensor2	BOOL	%IX100.5	No	TRUE
FC1	BOOL	%IX100.6	No	FALSE
FC2	BOOL	%IX100.7	No	FALSE
SensorAscensor1	BOOL	%IX100.8	No	TRUE
SensorAscensor2	BOOL	%IX100.9	No	TRUE

Figura 65.- OpenPLC Runtime - Monitoring [13].

7.- Conexión OpenPLC – Factory I/O.

En este apartado se van a explicar los pasos a seguir para realizar la conexión entre OpenPLC y Factory I/O:

1. En primer lugar, debemos abrir Factory I/O y cargar la escena que contenga el modelo 3D de la planta industrial que se desea controlar.

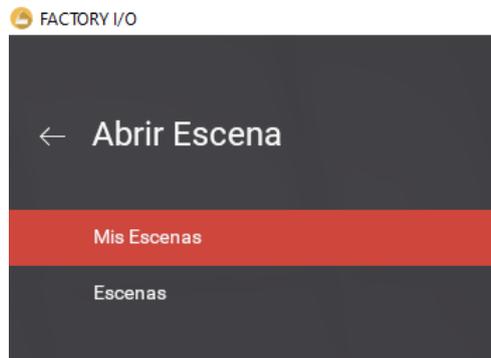


Figura 66.- Mis escenas – Factory I/O [1].

2. Posteriormente se abre la ventana de “Drivers” del menú desplegable “Archivo”.
3. Se escoge el controlador que se va a utilizar. Para conectarnos con el OpenPLC es necesario seleccionar “Modbus TCP/IP Server”.

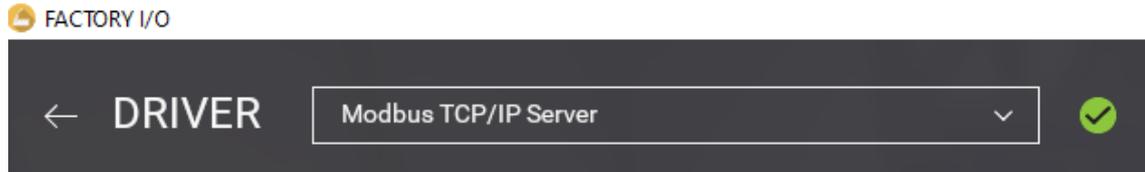


Figura 67.- Drivers – Factory I/O [1].

4. Se procede a realizar la configuración de dicho controlador entrando en la pestaña de “Configuración”, escogemos el controlador (Modbus TCP/IP Server).

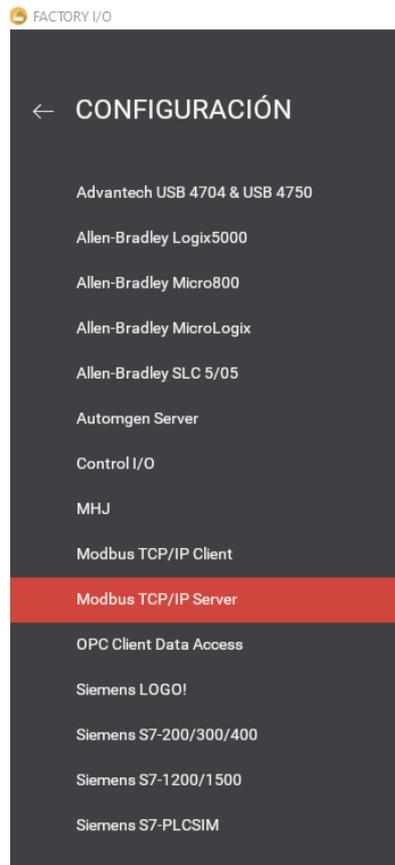


Figura 68.- Configuración – Factory I/O [1].

5. Se rellena la información solicitada:

Servidor:

- IP del Host.
- Número del puerto.
- Identificativo del esclavo.
- Adaptador de red.

Servidor

Autoarranque

Host
192.168.0.16

Puerto
502

Esclavo ID
1

Adaptador de RED
Intel(R) Dual Band Wireless-AC 3165

Figura 69.- Configuración – Servidor [12].

I/O Configuración:

- Escritura digital.
- Lectura digital.
- Escritura de registro.
- Lectura de registro.
- Escala.

I/O Configuración

Escritura Digital
Inputs

Lectura Digital
Coils

Escribir Registro
Input Registers

Leer Registro
Holding Registers

Escala
100

Figura 70.- Configuración – I/O [12].

I/O Terminales:

- Número de salidas BOOL.
- Número de entradas BOOL.
- Número de salidas de registro.

I/O Terminales		
	Offset	Cuenta
Salidas Bool	0	11
Entradas Bool	0	20
Entradas Registro	0	2
Salidas Registro	0	2

Figura 71.- Configuración – I/O Terminales [12].

6. Número de entradas de registro. A continuación, conectamos el controlador.

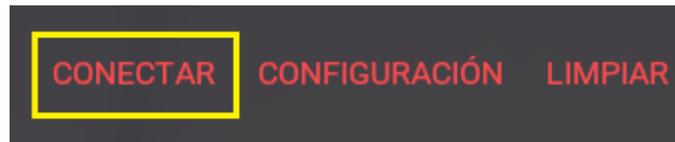


Figura 72.- Conectar controlador – Factory I/O [1].

7. Ahora se abre, en el OpenPLC Editor, el proyecto correspondiente a la simulación 3D que hemos realizado en Factory I/O.

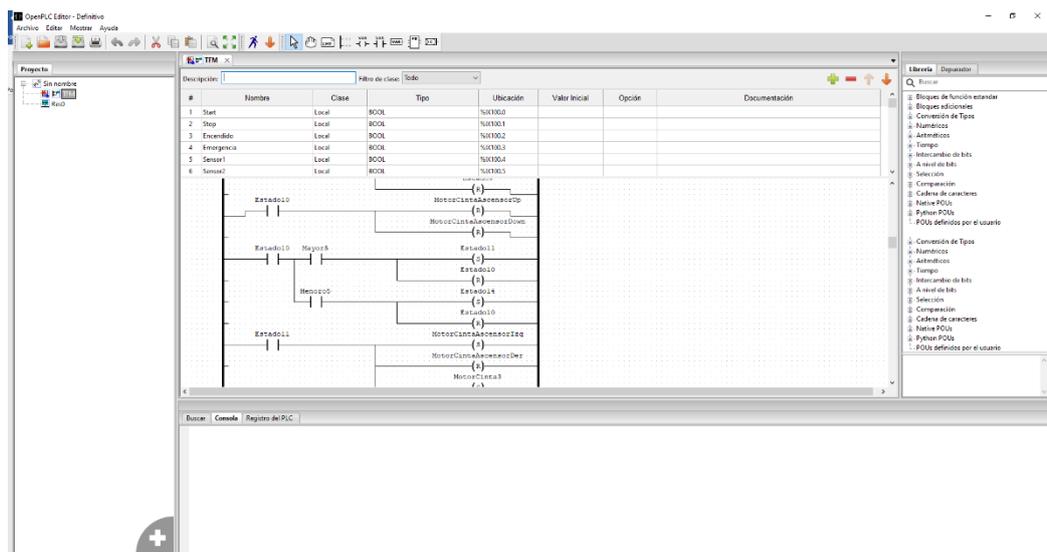


Figura 73.- Proyecto - OpenPLC Editor [2].

8. Se genera el programa para el OpenPLC Runtime.

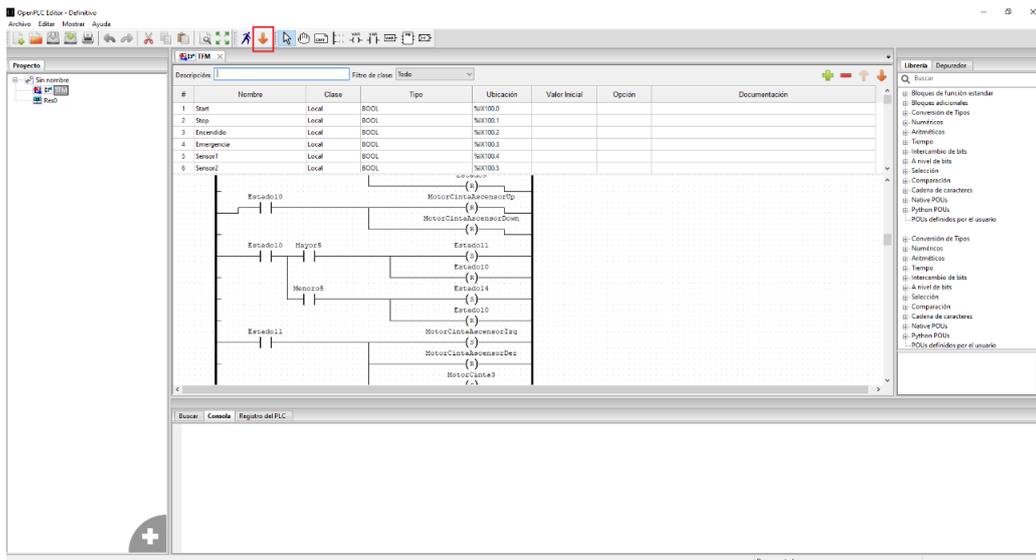


Figura 74.- Generación de programa - OpenPLC Editor [2].

9. Se revisa en la consola, que la generación del programa haya concluido con éxito, y se guarda el archivo generado.

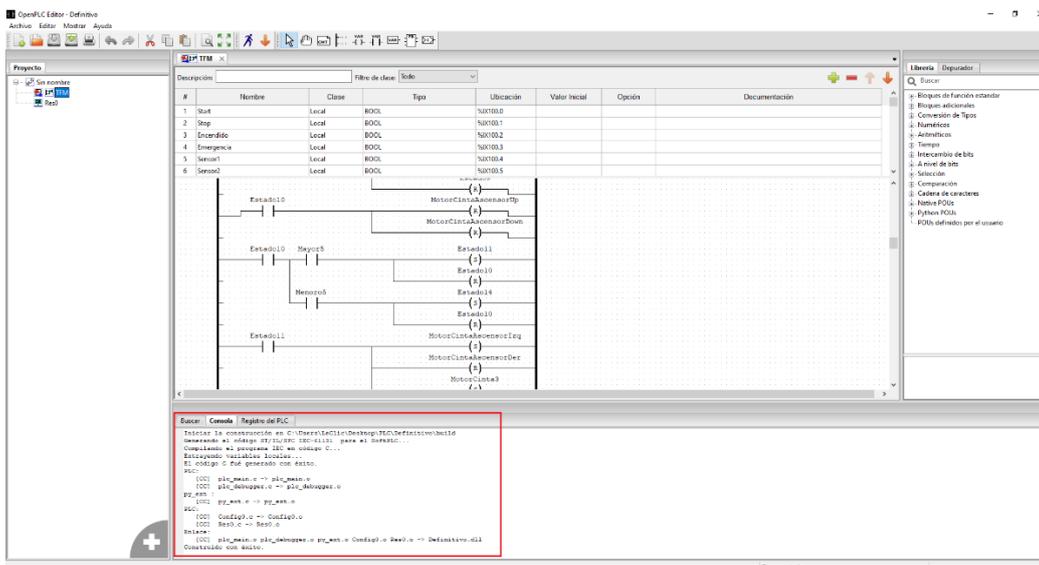
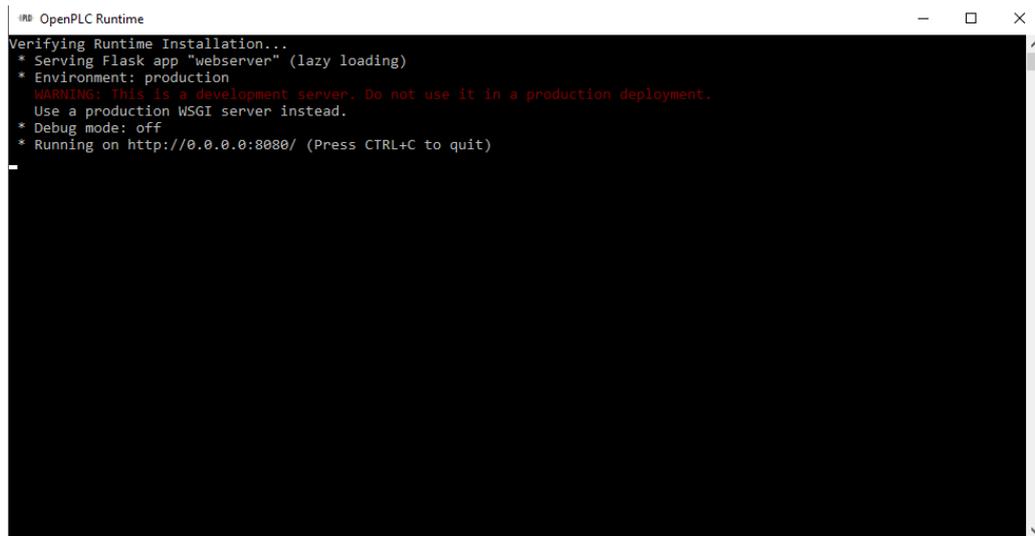


Figura 75.- Consola - OpenPLC Editor [2].

10. Después, se procede a abrir el OpenPLC Runtime, verificando que funcione correctamente.

A terminal window titled "OpenPLC Runtime" showing the following output:

```
Verifying Runtime Installation...
* Serving Flask app "webservice" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Figura 76.- OpenPLC Runtime [2].

11. En el siguiente paso abrimos el navegador y accedemos al servidor de OpenPLC (OpenPLC Webservice): <http://localhost:8080/login>

El usuario debe crear una cuenta con un nombre de usuario y una clave.

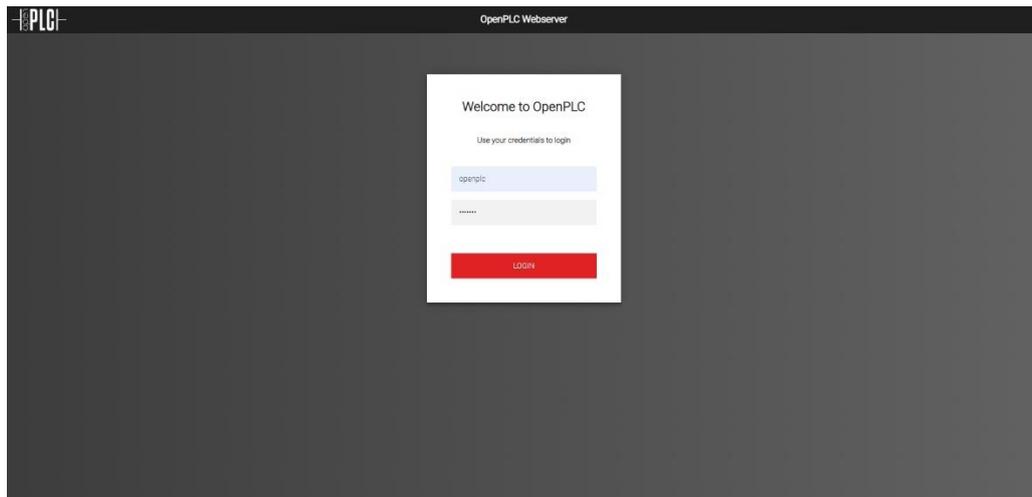


Figura 77.- OpenPLC Webservice [2].

12. Se carga el programa que se ha generado en la pestaña "Programs".

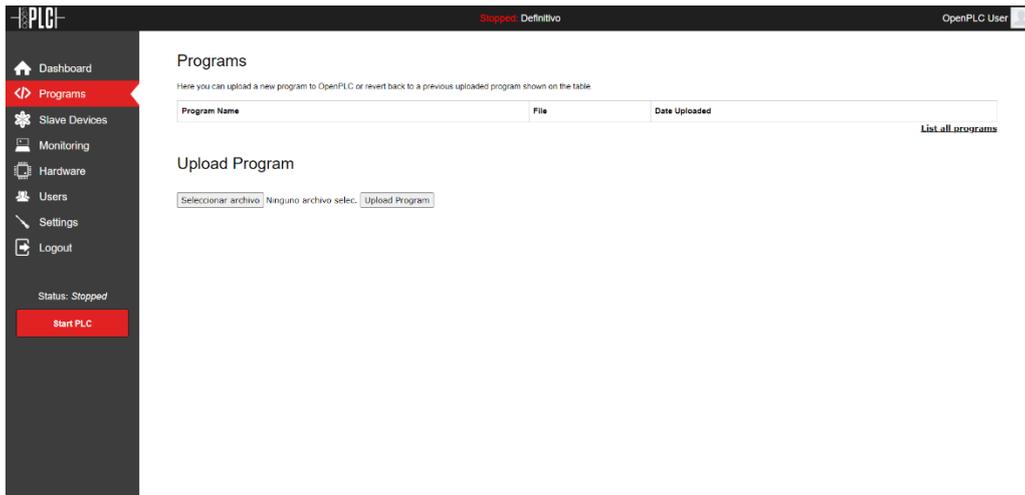


Figura 78.- Cargar programa [2].

13. Accedemos al dashboard donde se recogen los datos del programa que se ha generado en el OpenPLC Editor y se ha cargado anteriormente.

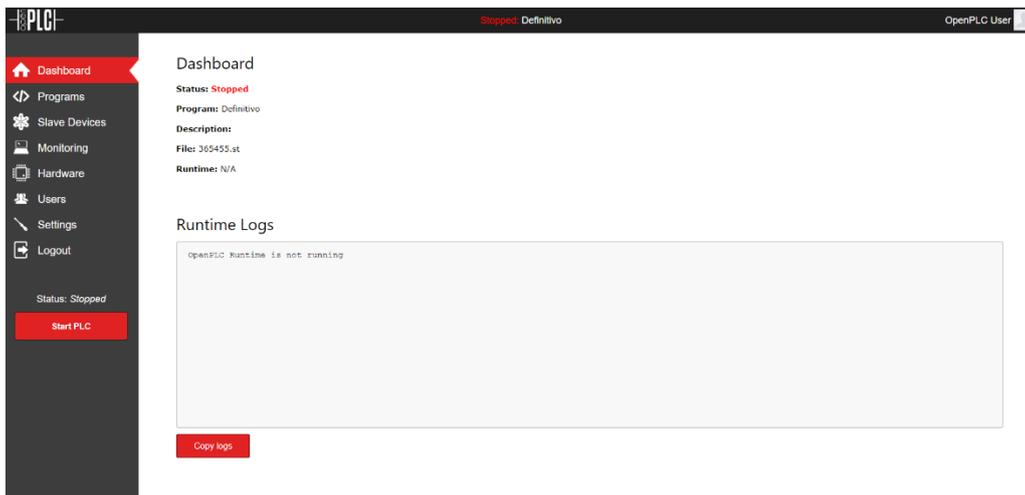


Figura 79.- Dashboard - OpenPLC [2].

14. Luego se añade la información solicitada dispositivo (Generic Modbus TCP Device), que hemos establecido anteriormente en la configuración del controlador en el Factory I/O.

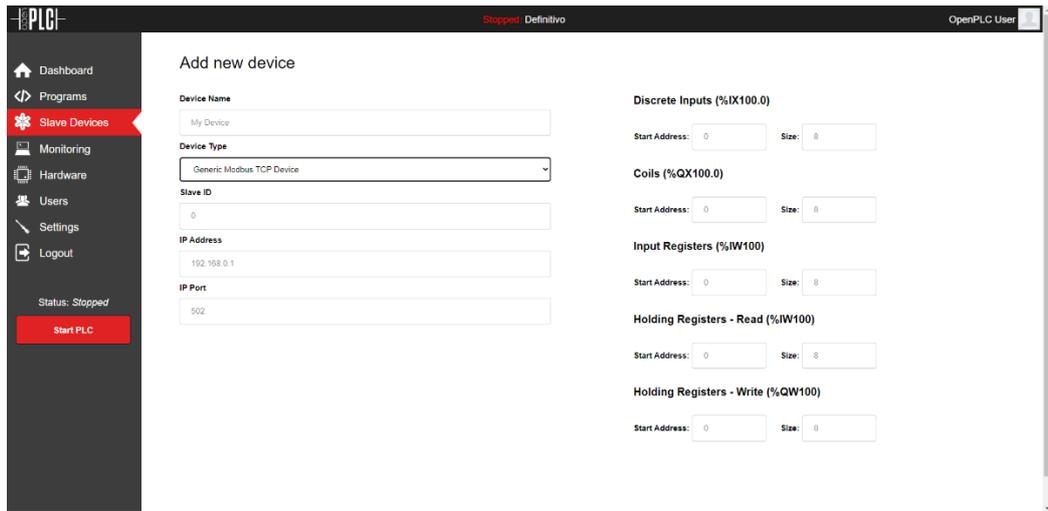


Figura 80.- Slave Devices [2].

15. Se activa el funcionamiento del PLC mediante el botón “Start PLC”.

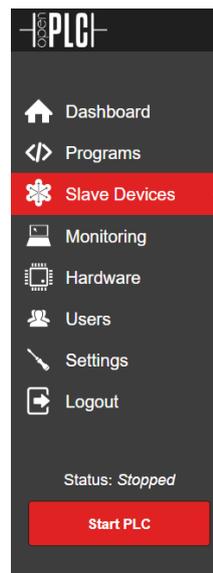


Figura 81.- Start PLC [2].

16. Finalmente, en el Factory I/O se activa el funcionamiento de la simulación mediante el botón “Play”, y se puede determinar si la conexión se ha realizado correctamente y si el programa funciona como se ha diseñado.

8.- Análisis de usabilidad.

En este apartado se realiza el análisis de los problemas, dificultades, elementos favorables y sensaciones que se han encontrado a la hora de trabajar con los siguientes softwares:

- OpenPLC.
- Factory I/O.

8.1.- Análisis OpenPLC.

En el OpenPLC Editor resulta un poco incómodo añadir líneas de código a un programa ya editado. Esto es debido a que, para añadir nuevas líneas de código, hay que añadir conexiones a la línea de potencia, y esto implica desconectar y volver a conectar todas las entradas y salidas. En programas largos resulta engorroso. Este problema se ha solucionado añadiendo más conexiones a la línea de potencia de las necesarias, por si en el futuro se quiere modificar el código.

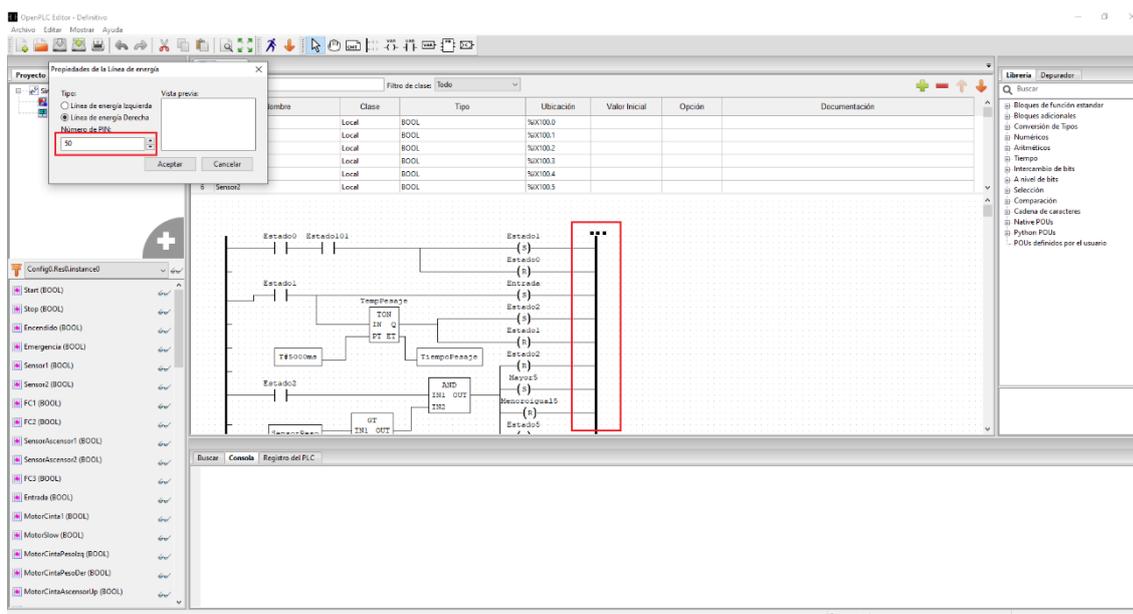


Figura 82.- Línea de potencia [2].

Por otra parte, cabe destacar que las entradas y salidas digitales se nombran desde la 0 hasta la 99 en este editor. Hay que tenerlo en cuenta para que dichas entradas y salidas coincidan con las del modelo 3D de Factory I/O (esto se ajusta en el apartado “drivers” del programa Factory I/O).

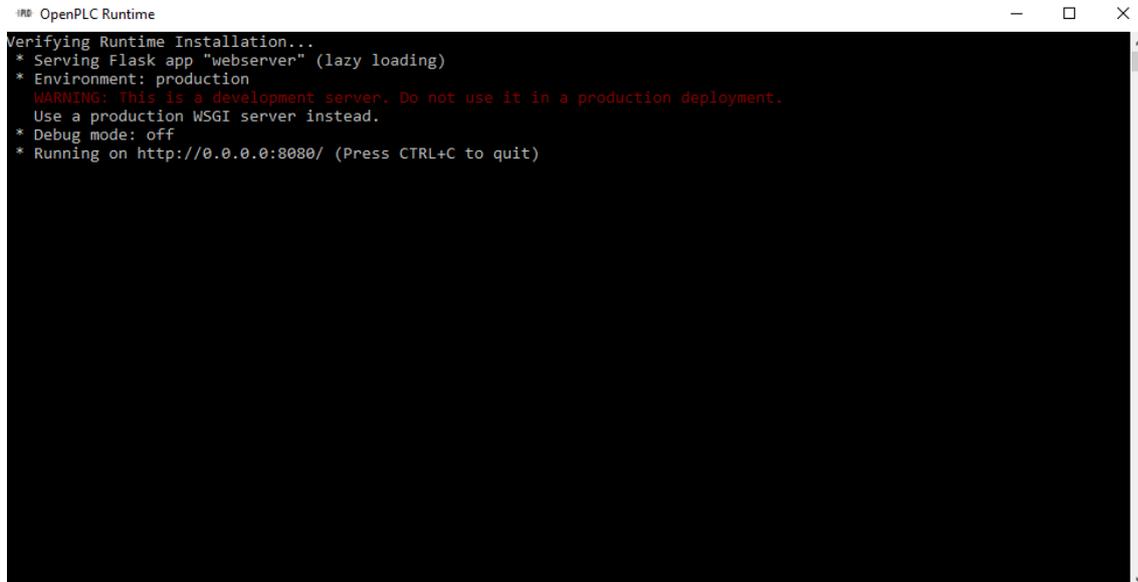
#	Nombre	Clase	Tipo	Ubicación
13	MotorCinta1	Local	BOOL	%QX100.1
14	MotorSlow	Local	BOOL	%QX100.2
15	MotorCintaPesolzq	Local	BOOL	%QX100.3
16	MotorCintaPesoDer	Local	BOOL	%QX100.4
17	MotorCintaAscensorUp	Local	BOOL	%QX100.5
18	MotorCintaAscensorDown	Local	BOOL	%QX100.6
19	MotorCinta2	Local	BOOL	%QX100.7
20	MotorCinta3	Local	BOOL	%QX100.8
21	MotorCintaAscensorIzq	Local	BOOL	%QX100.9
22	MotorCintaAscensorDer	Local	BOOL	%QX100.10
23	LEDStart	Local	BOOL	%QX100.11
24	LEDStop	Local	BOOL	%QX100.12
25	LEDEmergencia	Local	BOOL	%QX100.13
26	LEDContenedor1Full	Local	BOOL	%QX100.14
27	LEDContenedor2Full	Local	BOOL	%QX100.15
28	Output1	Local	BOOL	%QX100.16
29	Output2	Local	BOOL	%QX100.17
30	Menoroigual5	Local	BOOL	%QX100.18
31	Mayor5	Local	BOOL	%QX100.19
32	Estado1	Local	BOOL	%MX100.0
33	Estado2	Local	BOOL	%MX100.1

Figura 83.- Variables – OpenPLC [2].

(192.168.0.16:502) Esclavo ID:1			
Botón Start 1	Input 0	Coil 0	Entrada
Botón STOP 1	Input 1	Coil 1	Motor Cinta1
Encendido	Input 2	Coil 2	MotorSlow
Botón Emergencia 1	Input 3	Coil 3	Motor Cinta Peso Izq
Sensor 1	Input 4	Coil 4	Motor Cinta Peso Der
Sensor 2	Input 5	Coil 5	Motor Ascensor Ascenso
FC1	Input 6	Coil 6	Motor Ascensor Descenso
FC2	Input 7	Coil 7	MotorCinta2
Sensor Ascensor 1	Input 8	Coil 8	MotorCinta3
Sensor Ascensor 2	Input 9	Coil 9	Motor Cinta Ascensor Izq
FC3	Input 10	Coil 10	Motor Cinta Ascensor Der
Sensor de Peso	Input Reg 0	Coil 11	LED Start 1
FACTORY I/O (Time Scale)	Input Reg 1	Coil 12	LED STOP 1
		Coil 13	LED Emergencia 1
		Coil 14	LED Contenedor 1 Lleno
		Coil 15	LED Contenedor 2 Lleno
		Coil 16	Output 1
		Coil 17	Output 2
		Coil 18	Menor o igual 5
		Coil 19	Mayor 5
		Holding Reg 0	Contenedor 2
		Holding Reg 1	Contenedor 1

Figura 84.- Variables – Factory I/O [12].

Con respecto al uso del OpenPLC Runtime, es importante tener en cuenta que se ejecuta en segundo plano sobre una máquina virtual. Para cerrarlo se debe presionar CTRL+C. Si no se detiene el programa de esta manera, se queda el puerto ocupado y da error al volverlo a reiniciar.

The image shows a terminal window titled "OpenPLC Runtime". The text inside the terminal is as follows:

```
Verifying Runtime Installation...
* Serving Flask app "webserver" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Figura 85.- OpenPLC Runtime [2].

Por lo demás, el funcionamiento del software OpenPLC Runtime es muy sencillo y fácil de entender. Todas sus funcionalidades están bien descritas. La carga del programa desde el OpenPLC Editor es intuitiva y el procedimiento de conexión que se realiza entre el OpenPLC Runtime y el Factory I/O es efectivo.

Por último, comentar que hemos encontrado un problema en la transmisión de los valores de las entradas analógicas entre el Factory I/O y el OpenPLC. Las señales se comunican en escalas diferentes. Por ejemplo, el sensor de peso de Factory I/O ofrece la medida en Kg, pero el OpenPLC la registra en gramos. Esto es importante tenerlo en cuenta a la hora de realizar el programa.

8.2.- Análisis Factory I/O.

La mayor dificultad de usabilidad que se ha encontrado en el manejo de este programa ha sido en la parte de edición de escenas. Por un lado, hay que aprender a manejar las cámaras que permiten visualizar la escena, porque al principio es difícil moverse por ella, o enfocar con detalle alguna parte de esta. Por otro, nos ha costado colocar adecuadamente los sensores de fin de carrera en algunos modelos 3D, sobre todo en las escenas que incluyen el ascensor. Era

necesario ajustar su posición teniendo en cuenta la velocidad de los motores de subida y bajada del ascensor, para que se pudiera detener en el punto deseado. Por lo demás, se trata de un software sencillo y hasta divertido de usar.

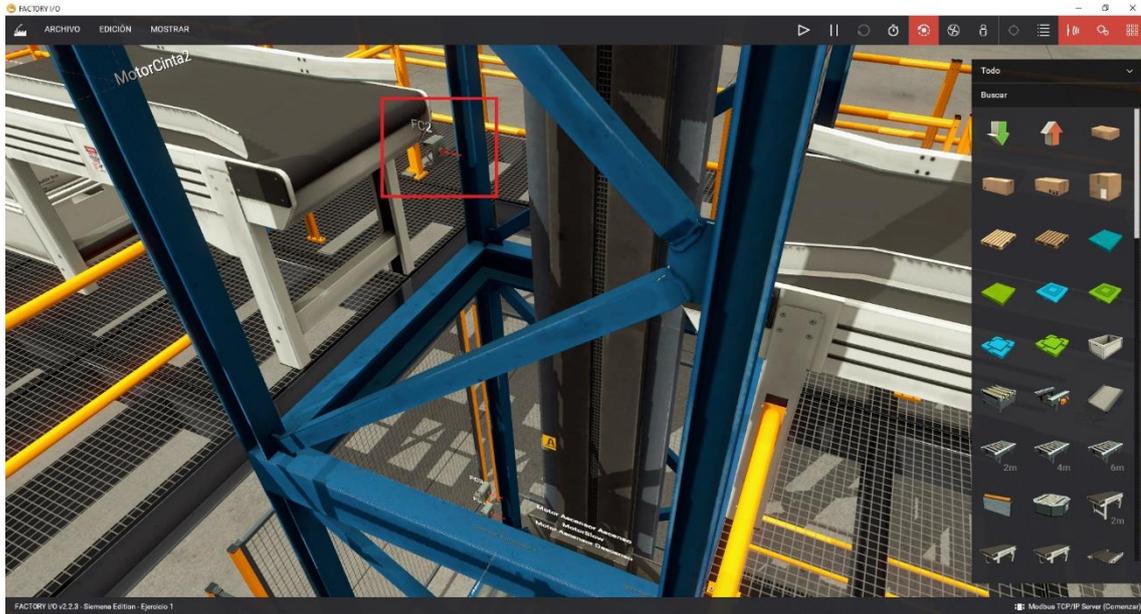


Figura 86.- Colocación de sensores [1].

9.- Resultados: prácticas probadas.

En este apartado se muestran las prácticas de automatización que se han probado para comprobar que se pueden hacer utilizando el software de OpenPLC:

- Práctica 1: Automatización de una cinta transportadora.
- Práctica 2: Automatización de un montacargas.
- Práctica 3: Automatización de un semáforo.
- Práctica 4: Automatización de un sistema clasificador de paquetes.
- Práctica 5: Automatización de un sistema clasificador de paquetes según peso.

9.1.- Práctica 1: Automatización de una cinta transportadora.

Se pretende automatizar la cinta transportadora de la figura 87.

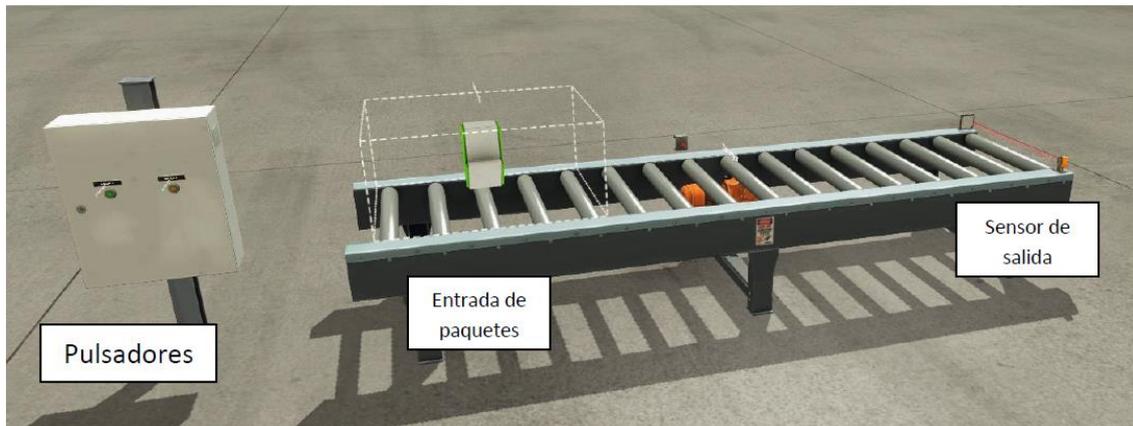


Figura 87.- Práctica 1.

El funcionamiento de la planta es el siguiente:

1. Comienza cuando el operario pulsa el botón Start, teniendo en cuenta que el valor de inicialización de este botón es 0, se produce un flanco de subida, y se activa la entrada de paquetes y el motor de la cinta transportadora.
2. Finalmente, cuando el paquete alcanza el final de la cinta transportadora, el sensor, cuyo valor de inicialización es 1, detecta dicho paquete y se desactiva el motor de la cinta transportadora.
3. Se añade un botón de Reset, cuyo valor de inicialización es 1, que, al ser presionado por el operario, se desactiva tanto la entrada de paquetes, como el motor de la cinta transportadora.

Las entradas y salidas del autómata están conectadas a los siguientes sensores y actuadores:

Entradas	Salidas
Start - I100.0	Motor Cinta - Q100.0
Reset - I100.1	Entrada de paquetes - Q100.1
Sensor al final de la cinta - I100.2*	

Tabla 9.- Entradas y salidas - Práctica 1. (*) Este sensor es de lógica inversa: está a 1 cuando no detecta nada, y a 0 cuando detecta un paquete.

El código generado con OpenPLC se muestra en el Anexo I. El funcionamiento de este modelo 3D de Factory controlado por el OpenPLC Runtime se muestra en el vídeo <https://youtu.be/uDE1AZrKfII>

9.2.- Práctica 2: Automatización de un montacargas.

Se pretende automatizar el montacargas de la figura 88.

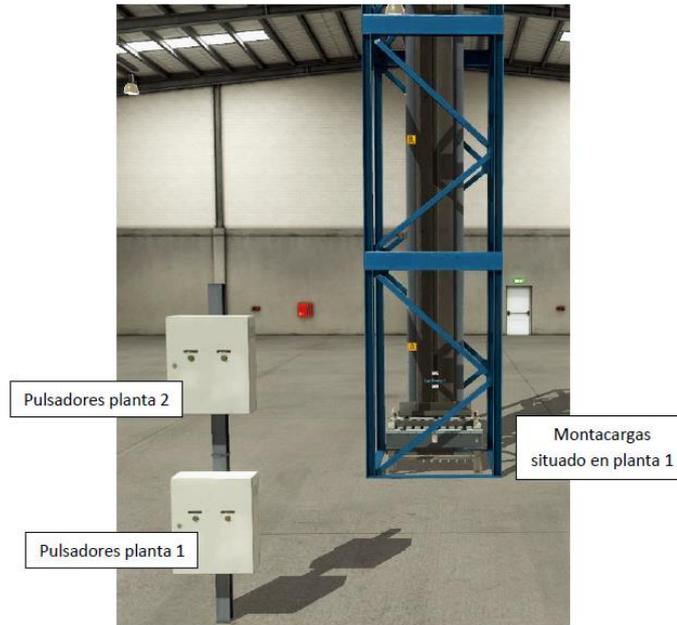


Figura 88.- Práctica 2.

En primer lugar, cabe destacar que en cada una de las dos plantas existe un sensor fin de carrera (sensor inductivo) que nos dará la información de la situación del montacargas. También se debe conocer que el valor inicial de todos los botones es 0, por lo que al pulsaron se produciría un flanco positivo.

El funcionamiento de la planta es el siguiente:

- Si el montacargas se encuentra en la planta 1 y, recibe la petición desde la planta 2 o recibe la orden de envío a la planta 2 (en ambos casos se produciría un flanco positivo), se activa el motor de subida del montacargas y se resetea el motor de bajada de dicho montacargas.
- Si el montacargas se encuentra en la planta 2 y, recibe la petición desde la planta 1 o recibe la orden de envío a la planta 1 (en ambos casos se produciría un flanco positivo),

se activa el motor de bajada del montacargas y se resetea el motor de subida de dicho montacargas.

- Cuando el montacargas sube y es detectado por el sensor fin de carrera de la planta 2 o cuando el montacargas baja y es detectado por el sensor fin de carrera de la planta 1, se resetea el motor de bajada y el motor de subida.

Las entradas y salidas del autómatas están conectadas a los siguientes sensores y actuadores:

Entradas	Salidas
Pet1 - I100.0	Motor Subida (MS) - Q100.0
Pet2 - I100.1	Motor Bajada (MB) - Q100.1
Env1 - I100.2	
Env2 - I100.3	
FC1 - I100.4	
FC2 - I100.5	

Tabla 10.- Entradas y salidas - Práctica 2.

El código generado con OpenPLC se muestra en el Anexo II. El funcionamiento de este modelo 3D de Factory controlado por el OpenPLC Runtime se muestra en el vídeo <https://youtu.be/3Cpb57-gHg>

9.3.- Práctica 3: Automatización de un semáforo.

Se pretende automatizar el semáforo de la figura 89.

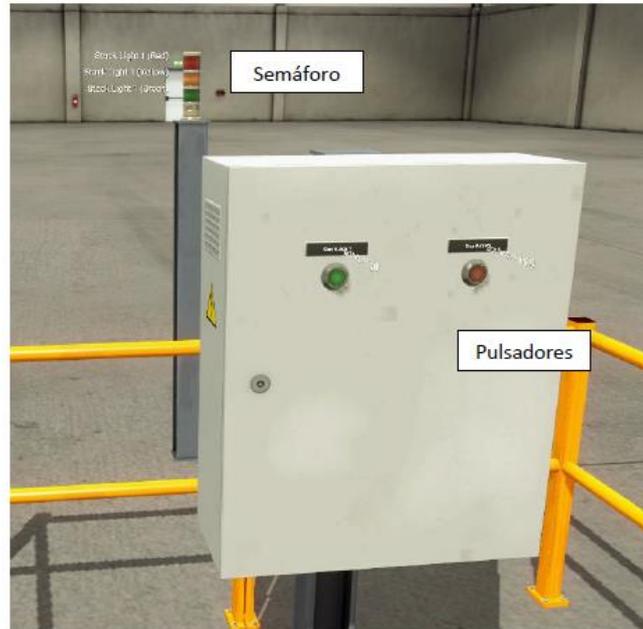


Figura 89.- Práctica 3.

El funcionamiento de la planta es el siguiente:

1. Cuando el trabajador pulsa el botón de Start, teniendo en cuenta que el valor de inicialización de este botón es 0, se produce un flanco de subida y se activa el estado 1.
2. Al activarse el estado 1, se enciende la luz roja y se activa un temporizador TON que comienza a contar el tiempo asignado para la luz roja, en este caso 5 segundos.
3. Al transcurrir los 5 segundos designados para la luz roja, se resetea el estado 1, el botón Start y dicha luz roja, y se enciende la luz amarilla.
4. Cuando se enciende la luz amarilla, se activa un temporizador TON que comienza a contar el tiempo asignado para la luz amarilla, en este caso 6 segundos.
5. Al transcurrir los 6 segundos designados para la luz amarilla, se apaga la luz amarilla y se enciende la luz verde.
6. Cuando se enciende la luz verde, se activa un temporizador TON que comienza a contar el tiempo asignado para la luz verde, en este caso 8 segundos.
7. Al transcurrir los 8 segundos designados para la luz verde, se apaga la luz verde y se activa el estado 1, comenzando de nuevo el ciclo de en el punto 2.

8. En caso de que el trabajador pulse el botón Stop, teniendo en cuenta que el valor de inicialización de este botón es 1, se resetea el estado 1, la luz roja, luz amarilla, y luz verde.

Las entradas y salidas del autómatas están conectadas a los siguientes sensores y actuadores:

Entradas	Salidas
Start - I100.0	Luz Roja - Q100.0
Stop - I100.1*	Luz Amarilla - Q100.1
	Luz Verde - Q100.2

Tabla 11.- Entradas y salidas - Práctica 3. (*) **Este pulsador es de lógica inversa:** está a 1 cuando no está pulsado, y pasa a 0 cuando se pulsa.

El código generado con OpenPLC se muestra en el anexo III. El funcionamiento de este modelo 3D de Factory controlado por el OpenPLC Runtime se muestra en el vídeo <https://youtu.be/r8KEw16iOxE>

9.4.- Práctica 4: Automatización de un sistema clasificador de paquetes.

Se pretende automatizar el sistema clasificador de paquetes de la figura 90.

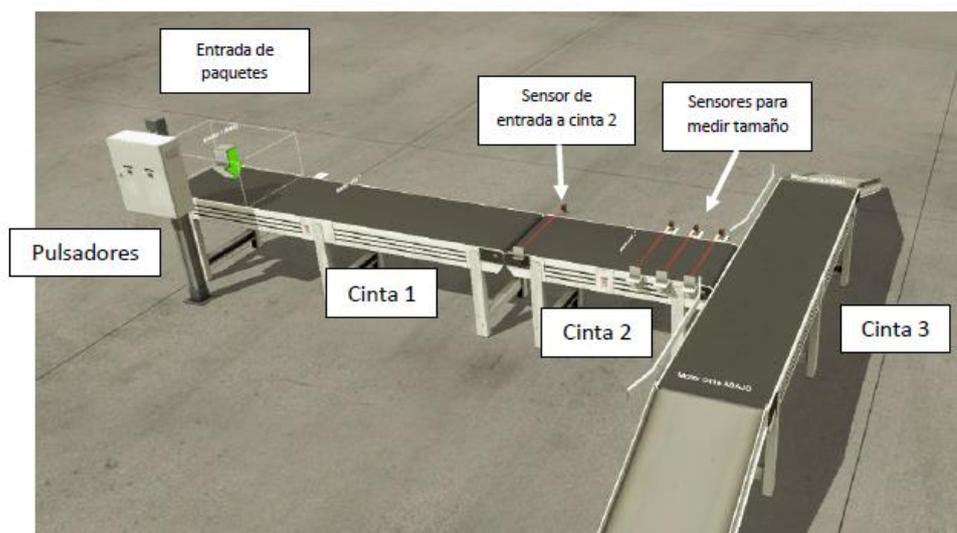


Figura 90.- Práctica 4.

El funcionamiento de la planta es el siguiente:

1. Cuando el trabajador pulsa el botón de Start, teniendo en cuenta que el valor de inicialización de este botón es 0, se produce un flanco de subida y se activa el estado 1.
2. Al activarse el estado 1, se activa la entrada de paquetes y el motor correspondiente a la cinta transportadora 1.
3. Cuando el paquete alcanza el final de la cinta transportadora 1, el sensor de la cinta 2, cuyo valor de inicialización es 1, detecta dicho paquete, se desactiva la entrada, el estado 1 y el motor de la cinta transportadora 1 y se activa el motor de la cinta transportadora 2.
4. En caso de que el paquete sea grande, es detectado por los sensores 1, 2 y 3 simultáneamente y se activa el estado 2 y el motor de la cinta transportadora 3 hacia arriba.
5. Al activarse el estado 2, se activa un temporizador TON que comienza a contar el tiempo asignado en el informe de la práctica, en este caso 10 segundos (se ha aumentado para que el paquete haya salido de la cinta transportadora 3).
6. Al transcurrir los 10 segundos, se resetea el estado 2, el motor de la cinta transportadora 3 hacia arriba y se activa el estado 1, comenzando de nuevo el ciclo.
7. En caso de que el paquete sea pequeño, es detectado por el sensor 2 a su paso por la cinta transportadora 2, y se activa el estado 3 y el motor de la cinta transportadora 3 hacia abajo.
8. Al activarse el estado 3, se activa un temporizador TON que comienza a contar el tiempo asignado en el informe de la práctica, en este caso 10 segundos (se ha aumentado para que el paquete haya salido de la cinta transportadora 3).
9. Al transcurrir los 10 segundos, se resetea el estado 3, el motor de la cinta transportadora 3 hacia abajo y se activa el estado 1, comenzando de nuevo el ciclo.
10. En caso de que el trabajador pulse el botón Stop, teniendo en cuenta que el valor de inicialización de este botón es 1, se resetea el estado 1, la entrada de paquetes, el motor de la cinta transportadora 1, el motor de la cinta transportadora 2, el motor de la cinta transportadora 3 hacia arriba y el motor de la cinta transportadora 3 hacia abajo.

Las entradas y salidas del autómatas están conectadas a los siguientes sensores y actuadores:

Entradas	Salidas
Start - I100.0	Cinta 1 - Q100.0
Stop - I100.1*	Cinta 2 - Q100.1
Sensor medida 1 - I100.2**	Cinta 3 - Arriba - Q100.2
Sensor medida 2 - I100.3**	Cinta 3 - Abajo - Q100.3
Sensor medida 3 - I100.4**	Entrada de paquetes - Q100.4
Sensor entrada cinta 2 - I100.5**	

Tabla 12.- Entradas y salidas - Práctica 4. (*) **Este pulsador es de lógica inversa:** está a 1 cuando no está pulsado, y pasa a 0 cuando se pulsa.

El código generado con OpenPLC se muestra en el anexo IV. El funcionamiento de este modelo 3D de Factory controlado por el OpenPLC Runtime se muestra en el vídeo <https://youtu.be/F8MZN5QMm-M>

9.5.- Práctica 5: Automatización de un sistema clasificador de paquetes según peso.

En esta práctica se pretende realizar la automatización de un clasificador de peso, dependiendo de los paquetes que se coloquen en la cinta transportadora.



Figura 91.- Práctica 5.

La planta debe:

- Clasificar los paquetes dependiendo de los pesos (mayor o igual a 5 kg, o menor a 5 kg).
- El peso se obtendrá mediante una báscula transportadora.
- El paquete se pesa en la planta baja y se clasifica en la planta alta
- Dependiendo del peso, en la planta alta, el paquete se desplazará en un sentido u otro.
- Cuando el paquete haya llegado al contenedor, se pone en marcha el contador del contenedor y se activará la salida de un nuevo paquete en la planta baja.
- Cuando el contenedor de uno de los pesos establecidos, según la configuración del usuario, esté completo, se para el proceso hasta que el operario lo vuelva a activar.
- Existen los siguientes botones y pulsadores:
 - Botón Encendido para el estado inicial del proceso.
 - Botón Start para activar el funcionamiento normal del proceso.
 - Botón Stop para detener el funcionamiento normal del proceso debido a alguna necesidad del operario.
 - Botón de Emergencia para detener el funcionamiento normal del proceso debido a alguna situación de emergencia o debido a un error en dicho proceso.

Las entradas y salidas del autómatas están conectadas a los siguientes sensores y actuadores:

Entradas	Salidas
Start – I100.0	Entrada – Q100.0
Stop – I100.1*	MotorCinta1 – Q100.1
Encendido – I100.2	MotorSlow – Q100.2
Emergencia – I100.3*	MotorCintaPesoIzq – Q100.3
Sensor 1 – I100.4	MotorCintaPesoDer – Q100.4
Sensor 2 – I100.5	MotorCintaAscensorUp – Q100.5
FC1 – I100.6	MotorCintaAscensorDown – Q100.6
FC2 – I100.7	MotorCinta2 – Q100.7
SensorAscensor1 – I100.8	MotorCinta3 – Q100.8
SensorAscensor2 – I100.9	MotorCintaAscensorIzq – Q100.9
FC3 – I100.10	MotorCintaAscensorDer – Q100.10
	LEDStart – Q100.11
	LEDStop – Q100.12

	LEDEmergencia – Q100.13
	LEDContenedor1Full – Q100.14
	LEDContenedor2Full – Q100.15
	Output1 – Q100.16
	Output2 – Q100.17
	Menoroigual5 – Q100.18
	Mayor5 – Q100.19

Tabla 13.- Entradas y salidas - Práctica 5. (*) **Este pulsador es de lógica inversa:** está a 1 cuando no está pulsado, y pasa a 0 cuando se pulsa.

El código generado con OpenPLC se muestra en el anexo V. El funcionamiento de este modelo 3D de Factory controlado por el OpenPLC Runtime se muestra en el vídeo <https://youtu.be/pivCtr5Lr28>

10.- Conclusiones.

En este Trabajo Fin de Máster (TFM) se ha realizado un estudio de los principales softwares de simulación de automatización que hay en el mercado, y se han escogidos los softwares con las características que más se aproximan a las necesidades formativas de las asignaturas de automatización de la Escuela, priorizando la facilidad de uso y el coste.

La principal conclusión de este TFM es que la utilización combinada de OpenPLC y Factory I/O es viable para la realización de prácticas simuladas de automatización. Para comprobarlo, hemos realizado la simulación de algunas de las prácticas de las asignaturas de automatización en la Escuela.

El software OpenPLC, que es gratuito y abierto, es una herramienta muy interesante. Su parte de edición de código ofrece una gran variedad de elementos y posibilidades, para diferentes lenguajes de programación estándar, aunque requiere de un tiempo de aprendizaje para su uso eficiente. La parte Runtime es más fácil de manejar.

Por otra parte, el software Factory I/O posee una amplia gama de elementos para generar modelos 3D de plantas industriales, lo que permite la simulación de muchos tipos de procesos

diferentes. El diseño de estos modelos 3D es sencillo, ya que no implica la necesidad de saber programar en Unity, por ejemplo, como ocurre con otros paquetes software similares. Ciertamente es que no se trata de software libre, pero su precio no es abusivo, y el producto es robusto.

La conexión entre ambos programas es sencilla, aspecto muy importante, ya que esto no ocurre siempre en este tipo de conexiones.

Como se ha mencionado ya, en este TFM se han probado algunas de las prácticas que se proponen en las asignaturas de automatización de la Escuela. Como trabajo futuro se podrían diseñar nuevas prácticas, e incluso, ofrecer a los alumnos la posibilidad de proponer prácticas personalizadas.

10.- Conclusions

In this Master's Thesis (TFM) a study of the main automation simulation software on the market has been carried out, and the software with the characteristics that are closest to the training needs of automation subjects has been chosen of the university, prioritizing ease of use and cost.

The main conclusion of this TFM is that the combined use of OpenPLC and Factory I/O is viable for carrying out simulated automation practices. To verify this, we have simulated some of the practices of the automation subjects at the university.

OpenPLC software, which is free and open, is a very interesting tool. Its code editing part offers a great variety of elements and possibilities, for different standard programming languages, although it requires learning time for its efficient use. The Runtime part is easier to handle.

On the other hand, the Factory I/O software has a wide range of elements to generate 3D models of industrial plants, which allows the simulation of many different types of processes. The design of these 3D models is simple, since it does not imply the need to know how to program in Unity, for example, as is the case with other similar software packages. It is true that it is not free software, but its price is not unreasonable, and the product is robust.

The connection between both programs is simple, a very important aspect, since this does not always happen in this type of connection.

As has already been mentioned, in this TFM some of the practices that are proposed in the automation subjects of the University have been tested. As future work, new practices could be designed, and even offer students the possibility of proposing personalized practices.

11.- Referencias.

- [1] Página web del software Factory I/O: <https://docs.factoryio.com/>
- [2] Página web del software OpenPLC: <https://openplcproject.com/docs/openplc-overview/>
- [3] Página web del software realvirtual.io: <https://realvirtual.io/>
- [4] Página web del software Emulate3D: <https://www.demo3d.com/>
- [5] Página web del software Codesys: <https://www.codesys.com/>
- [6] Página web del software CIROS: <https://www.festo-didactic.com/es-es/productos/software-e-learning/ciros/ciros-aplicaciones-en-la-tecnologia-de-la-automatizacion.htm?fbid=ZXMuZXMuNTQ3LjE0LjE4LjExMTAuODE4OQ>
- [7] Garrido R. Desarrollo de la automatización de un proceso de clasificación, montaje y almacenamiento de piezas con implementación mediante la herramienta de programación IEC 61131-3 CODESYS y con simulación de la planta a través del software Factory I/O. TFG del Grado en Ingeniería Electrónica Industrial y Automática. Universitat Politècnica de València, 2021.
- [8] Lopez D. Desarrollo de la automatización de un sistema de montaje de piezas bajo demanda con almacenamiento automático mediante PLC M241 de Schneider y simulación del proceso con el software Factory I/O. TFG del Grado en Ingeniería en Tecnologías Industriales. Universitat Politècnica de València, 2021.
- [9] Huesca A. Proyecto de automatización de un sistema de almacenamiento mediante PLC M241 de Schneider con monitorización HMI y simulación del proceso con el software CIROS. TFG del Grado en Ingeniería Electrónica Industrial y Automática. Universitat Politècnica de València, 2021.
- [10] Cuevas M. Aplicación del simulador HYSYS. Plant para la resolución de ejercicios y el desarrollo de prácticas de laboratorio. Ed. Universidad de Jaén, 2011.
- [11] Gil I, Guevara J, García J., Leguizamon A. Análisis y simulación de procesos en Ingeniería Química. Ed. Universidad Nacional de Colombia, 2011.
- [12] Página web de Siemens automation:
<https://new.siemens.com/global/en/products/automation/systems.html>

ANEXO I

Práctica 1: Automatización de una cinta transportadora.

En este anexo se muestran las variables y el código OpenPLC que se ha programado para realizar la práctica 1.

#	Nombre	Clase	Tipo	Ubicación
1	Start	Local	BOOL	%IX100.0
2	Reset	Local	BOOL	%IX100.1
3	Sensor	Local	BOOL	%IX100.2
4	MotorCinta	Local	BOOL	%QX100.0
5	Entrada	Local	BOOL	%QX100.1

Tabla 14.- Variables - Práctica 1.

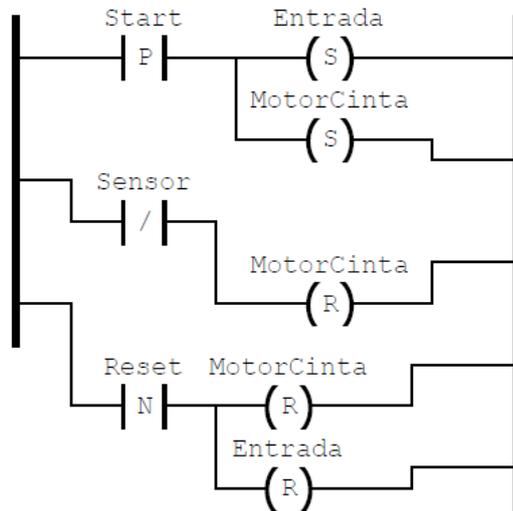


Figura 92.- Código - Práctica 1.

ANEXO II.

Práctica 2: Automatización de un montacargas.

En este anexo se muestran las variables y el código OpenPLC que se ha programado para realizar la práctica 2.

#	Nombre	Clase	Tipo	Ubicación
1	PET1	Local	BOOL	%IX100.0
2	PET2	Local	BOOL	%IX100.1
3	ENV1	Local	BOOL	%IX100.2
4	ENV2	Local	BOOL	%IX100.3
5	FC1	Local	BOOL	%IX100.4
6	FC2	Local	BOOL	%IX100.5
7	MS	Local	BOOL	%QX100.0
8	MB	Local	BOOL	%QX100.1

Tabla 15.- Variables - Práctica 2.

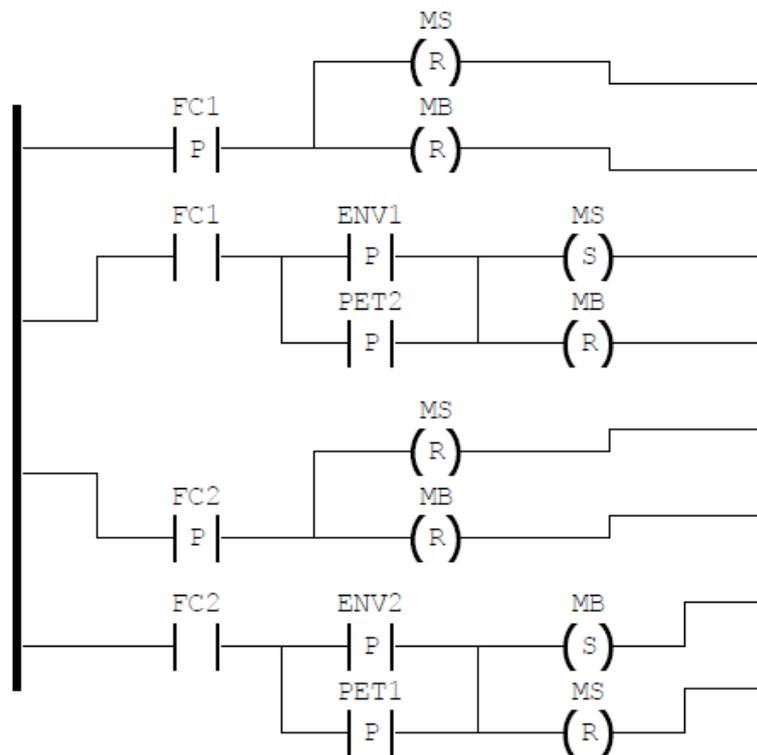


Figura 93.- Código - Práctica 2.

ANEXO III.**Práctica 3: Automatización de un semáforo.**

En este anexo se muestran las variables y el código OpenPLC que se ha programado para realizar la práctica 3.

#	Nombre	Clase	Tipo	Ubicación
1	Start	Local	BOOL	%IX100.0
2	Stop	Local	BOOL	%IX100.1
3	LuzRoja	Local	BOOL	%QX100.0
4	LuzAmarilla	Local	BOOL	%QX100.1
5	LuzVerde	Local	BOOL	%QX100.2
6	TiempoRojo	Local	TIME	
7	TiempoAmarillo	Local	TIME	
8	TiempoVerde	Local	TIME	
9	TON0	Local	TON	
10	TON1	Local	TON	
11	TON2	Local	TON	
12	Estado1	Local	BOOL	%MX100.0

Tabla 16.- Variables - Práctica 3.

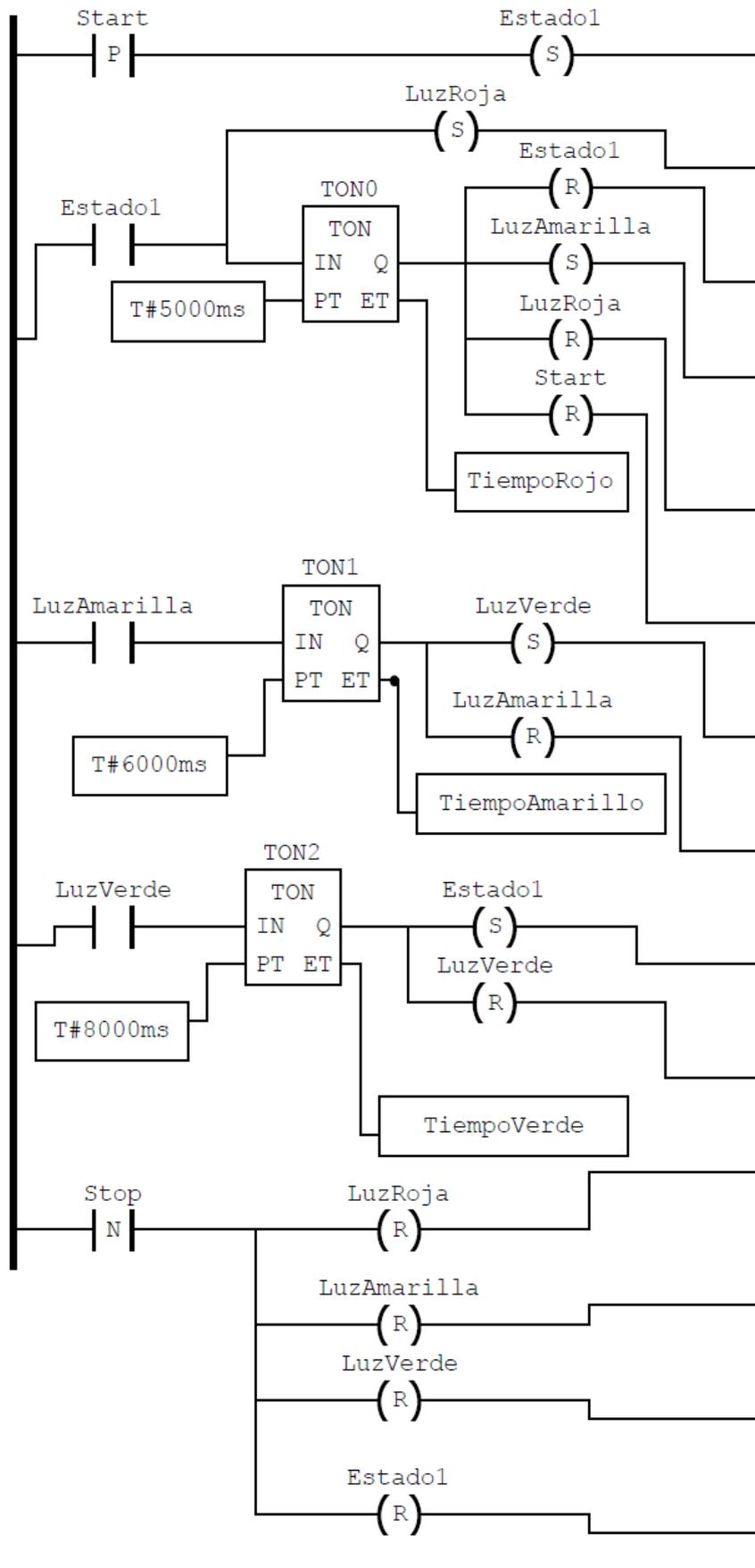


Figura 94.- Código - Práctica 3.

ANEXO IV.**Práctica 4: Automatización de un sistema clasificador de paquetes.**

En este anexo se muestran las variables y el código OpenPLC que se ha programado para realizar la práctica 4.

#	Nombre	Clase	Tipo	Ubicación
1	START	Local	BOOL	%IX100.0
2	STOP	Local	BOOL	%IX100.1
3	Sensor1	Local	BOOL	%IX100.2
4	Sensor2	Local	BOOL	%IX100.3
5	Sensor3	Local	BOOL	%IX100.4
6	SensorCinta2	Local	BOOL	%IX100.5
7	Cinta1	Local	BOOL	%QX100.0
8	Cinta2	Local	BOOL	%QX100.1
9	Cinta3Arriba	Local	BOOL	%QX100.2
10	Cinta3Abajo	Local	BOOL	%QX100.3
11	Entrada	Local	BOOL	%QX100.4
12	Estado1	Local	BOOL	%MX100.0
13	Estado2	Local	BOOL	%MX100.1
14	Estado3	Local	BOOL	%MX100.2
15	TON0	Local	TON	
16	TON1	Local	TON	
17	TiempoArriba	Local	TIME	
18	TiempoAbajo	Local	TIME	

Tabla 17.- Variables - Práctica 4.

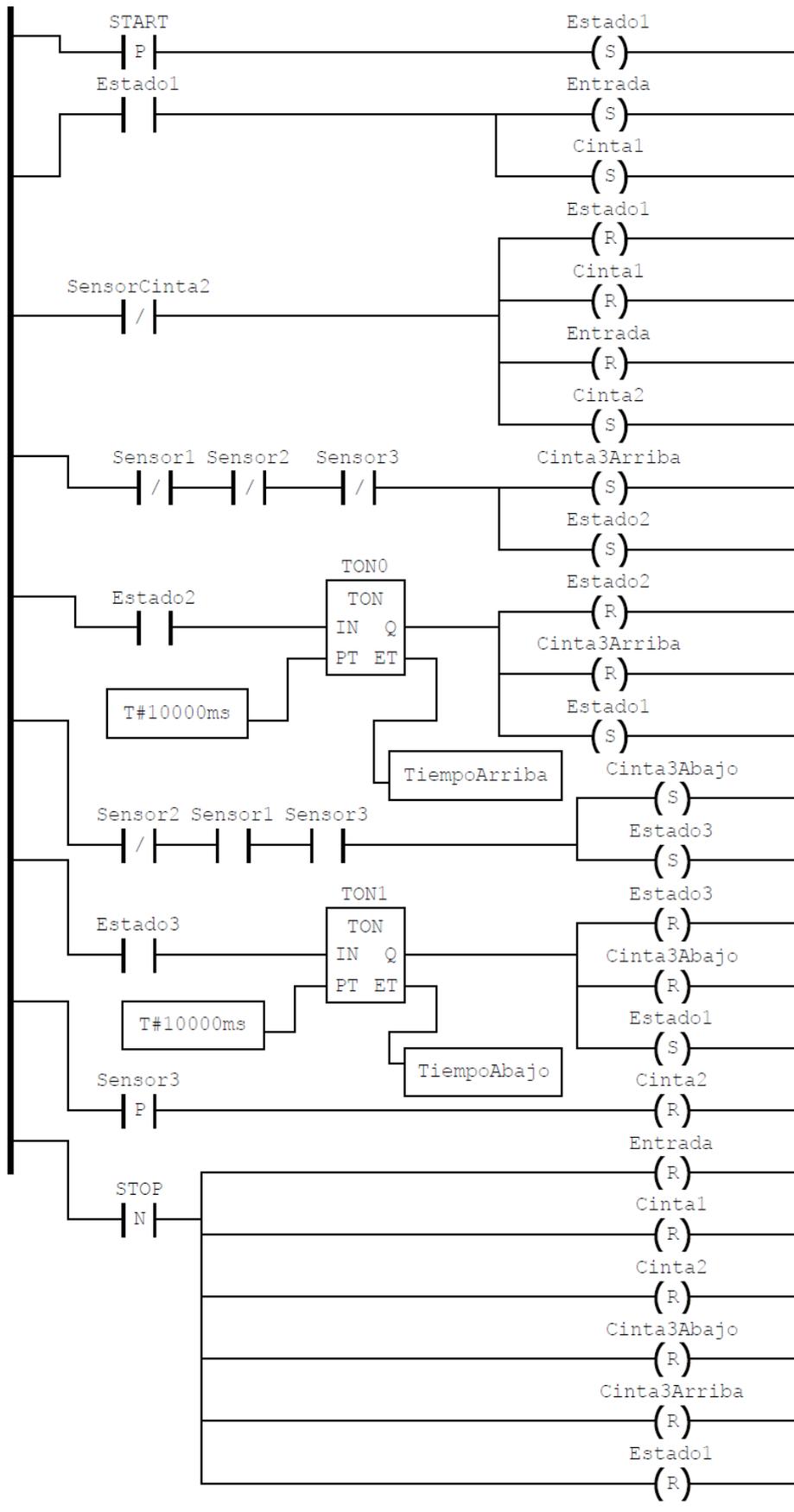


Figura 95.- Código - Práctica 4.

ANEXO V.**Práctica 5: Automatización de un sistema clasificador de paquetes según peso.**

En este anexo se muestran las variables y el código OpenPLC que se ha programado para realizar la práctica 5.

#	Nombre	Clase	Tipo	Ubicación
1	Start	Local	BOOL	%IX100.0
2	Stop	Local	BOOL	%IX100.1
3	Encendido	Local	BOOL	%IX100.2
4	Emergencia	Local	BOOL	%IX100.3
5	Sensor1	Local	BOOL	%IX100.4
6	Sensor2	Local	BOOL	%IX100.5
7	FC1	Local	BOOL	%IX100.6
8	FC2	Local	BOOL	%IX100.7
9	SensorAscensor1	Local	BOOL	%IX100.8
10	SensorAscensor2	Local	BOOL	%IX100.9
11	FC3	Local	BOOL	%IX100.10
12	Entrada	Local	BOOL	%QX100.0
13	MotorCinta1	Local	BOOL	%QX100.1
14	MotorSlow	Local	BOOL	%QX100.2
15	MotorCintaPesolzq	Local	BOOL	%QX100.3
16	MotorCintaPesoDer	Local	BOOL	%QX100.4
17	MotorCintaAscensorUp	Local	BOOL	%QX100.5
18	MotorCintaAscensorDown	Local	BOOL	%QX100.6
19	MotorCinta2	Local	BOOL	%QX100.7
20	MotorCinta3	Local	BOOL	%QX100.8
21	MotorCintaAscensorIzq	Local	BOOL	%QX100.9
22	MotorCintaAscensorDer	Local	BOOL	%QX100.10
23	LEDStart	Local	BOOL	%QX100.11
24	LEDStop	Local	BOOL	%QX100.12
25	LEDEmergencia	Local	BOOL	%QX100.13
26	LEDContenedor1Full	Local	BOOL	%QX100.14
27	LEDContenedor2Full	Local	BOOL	%QX100.15
28	Output1	Local	BOOL	%QX100.16
29	Output2	Local	BOOL	%QX100.17
30	Menorigual5	Local	BOOL	%QX100.18
31	Mayor5	Local	BOOL	%QX100.19
32	Estado1	Local	BOOL	%MX100.0
33	Estado2	Local	BOOL	%MX100.1
34	Estado3	Local	BOOL	%MX100.2
35	Estado4	Local	BOOL	%MX100.3
36	Estado5	Local	BOOL	%MX100.4
37	Estado6	Local	BOOL	%MX100.5
38	Estado7	Local	BOOL	%MX100.6
39	Estado8	Local	BOOL	%MX100.7
40	Estado9	Local	BOOL	%MX100.8
41	Estado10	Local	BOOL	%MX100.9
42	Estado11	Local	BOOL	%MX101.0
43	Estado12	Local	BOOL	%MX101.1
44	Estado13	Local	BOOL	%MX101.2
45	Estado14	Local	BOOL	%MX101.3
46	Estado20	Local	BOOL	%MX101.9
47	Estado21	Local	BOOL	%MX102.0
48	Estado30	Local	BOOL	%MX102.1
49	Estado31	Local	BOOL	%MX102.2
50	Estado32	Local	BOOL	%MX102.3
51	Estado33	Local	BOOL	%MX103.0
52	Estado100	Local	BOOL	%MX103.1
53	Estado101	Local	BOOL	%MX103.2
54	ContenedorLleno	Local	BOOL	%MX103.3
55	Estado0	Local	BOOL	%MX103.4
56	TiempoPesaje	Local	TIME	
57	TiempoAscensor	Local	TIME	
58	TiempoAscensor1	Local	TIME	
59	TiempoAscensor2	Local	TIME	
60	TiempoAscensor3	Local	TIME	
61	SensorPeso	Local	INT	%IW100
62	Contenedor1	Local	CTU	
63	Contenedor2	Local	CTU	
64	Contador1	Local	INT	%QW100
65	Contador2	Local	INT	%QW101
66	TempPesaje	Local	TON	
67	TempAscensor	Local	TON	
68	TempAscensor0	Local	TON	
69	TempAscensor3	Local	TON	
70	TempAscensor2	Local	TON	
71	TempAscensor1	Local	TON	

Tabla 18.- Variables - Práctica 5.

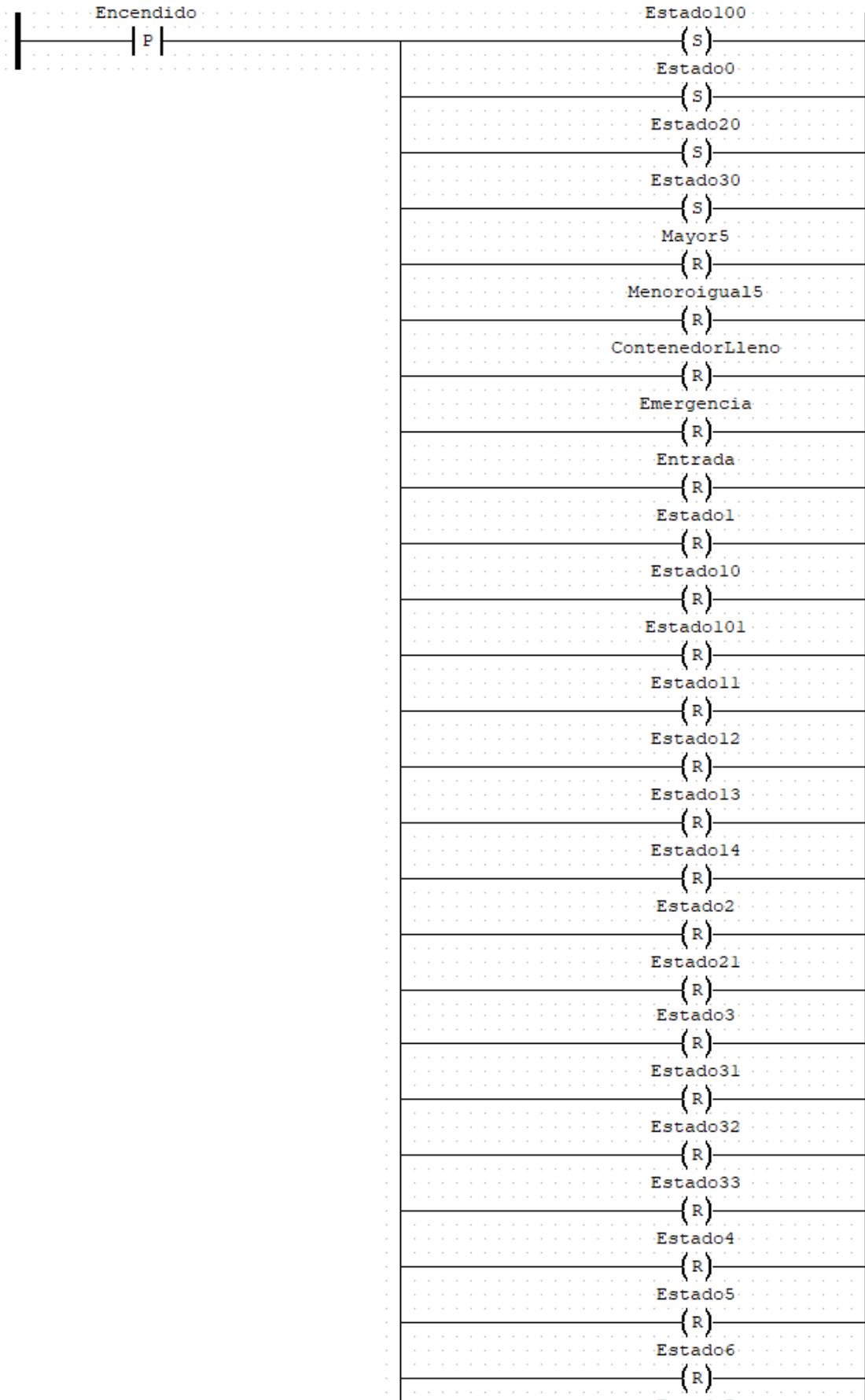


Figura 96.- Código puesta en marcha 1 - Práctica 5.

Estado7
(R)
Estado8
(R)
Estado9
(R)
LEDContenedor1Full
(R)
LEDContenedor2Full
(R)
LEDEmergencia
(R)
LEDStart
(R)
LEDStop
(R)
MotorSlow
(R)
MotorCintal
(R)
MotorCinta2
(R)
MotorCinta3
(R)
MotorCintaAscensorDer
(R)
MotorCintaAscensorDown
(R)
MotorCintaAscensorIzq
(R)
MotorCintaAscensorUp
(R)
MotorCintaPesoDer
(R)
MotorCintaPesoIzq
(R)
Output1
(R)
Output2
(R)
Start
(R)
Stop
(R)

Figura 97.- Código puesta en marcha 2 - Práctica 5.

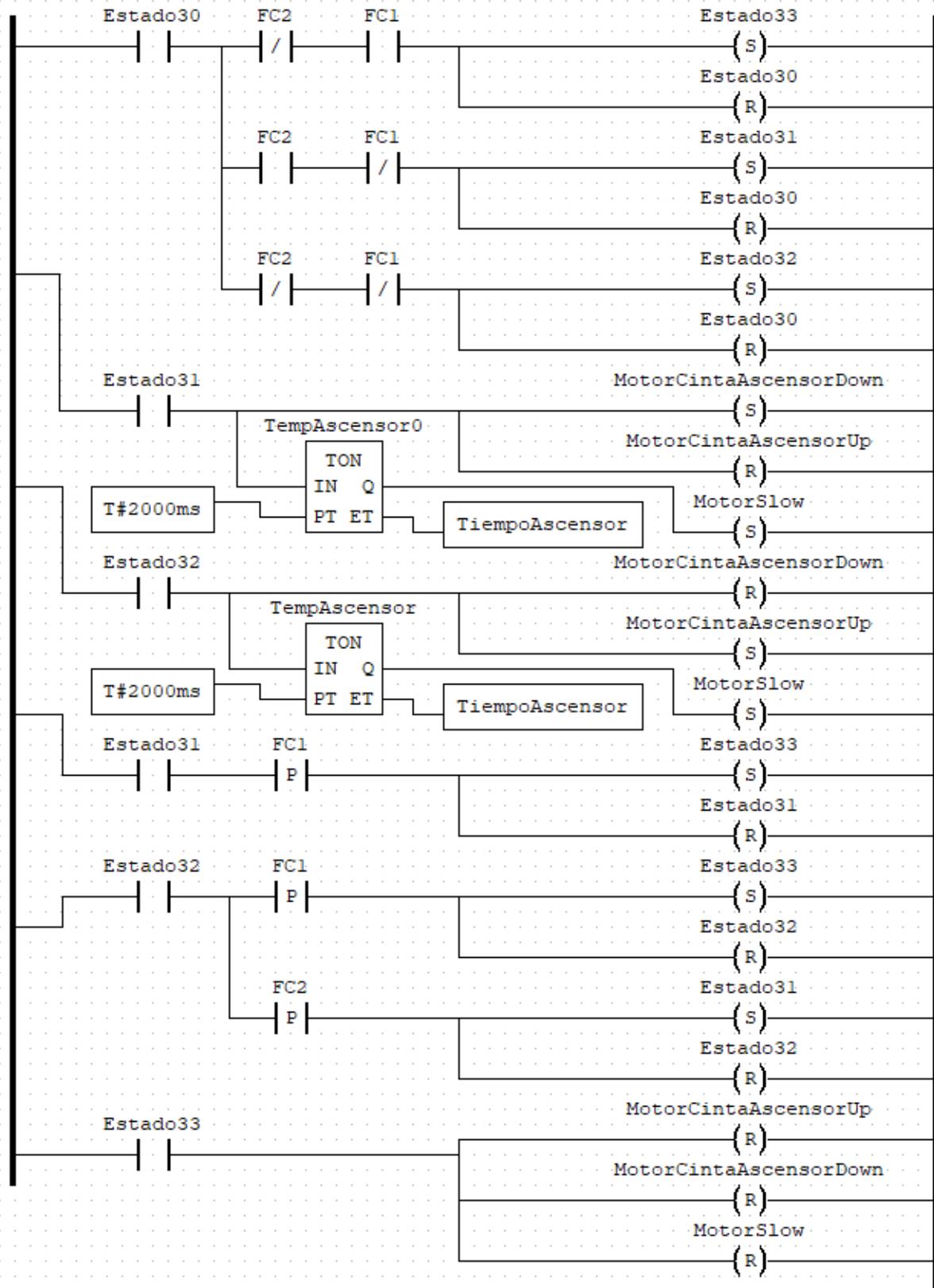


Figura 98.- Código estado inicial - Práctica 5.

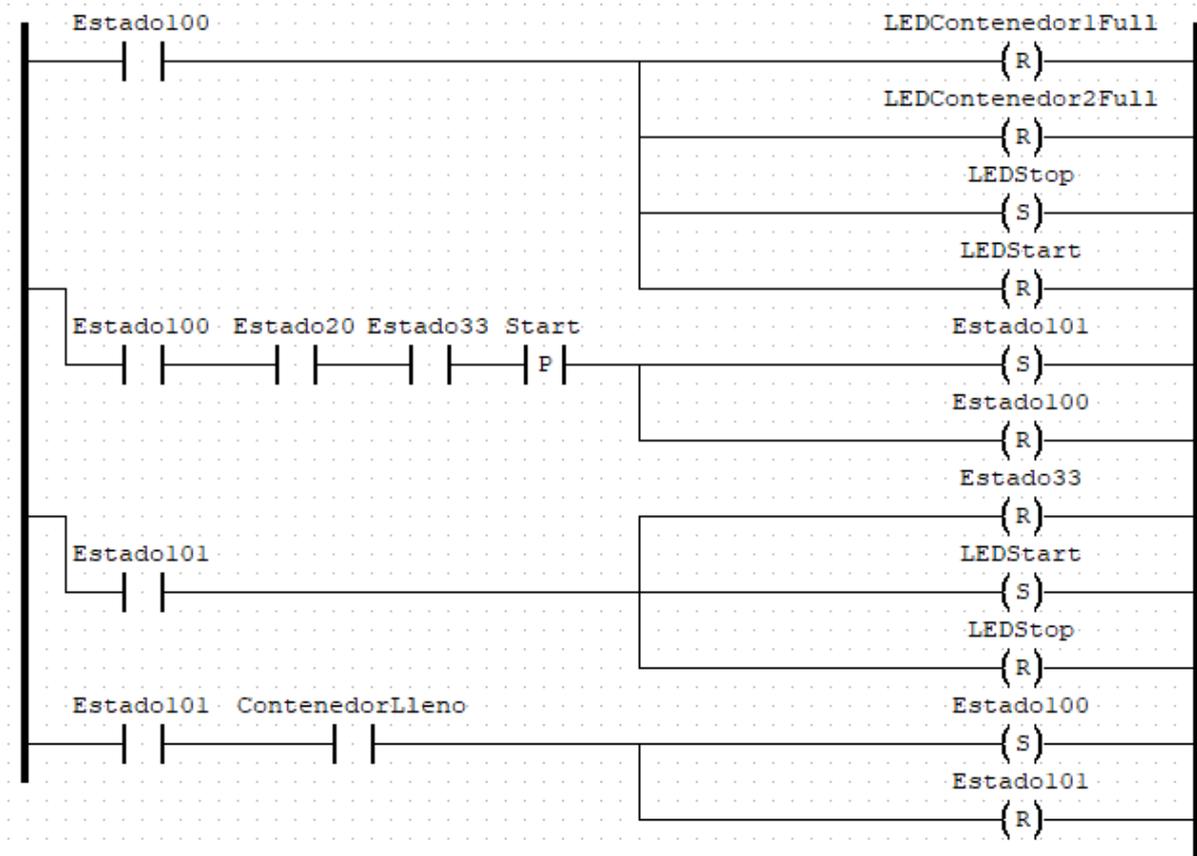


Figura 99.- Código operaciones - Práctica 5.

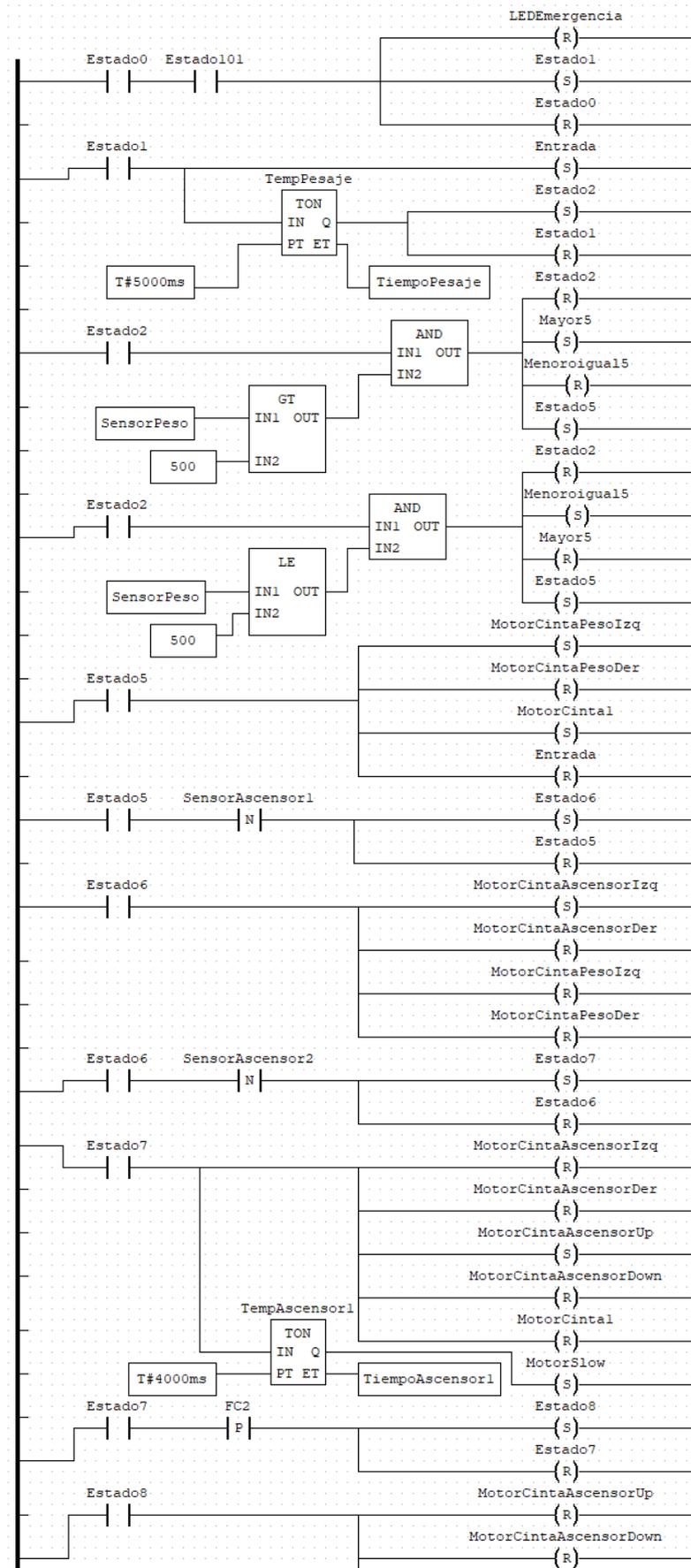


Figura 100.- Código funcionamiento normal 1 - Práctica 5.

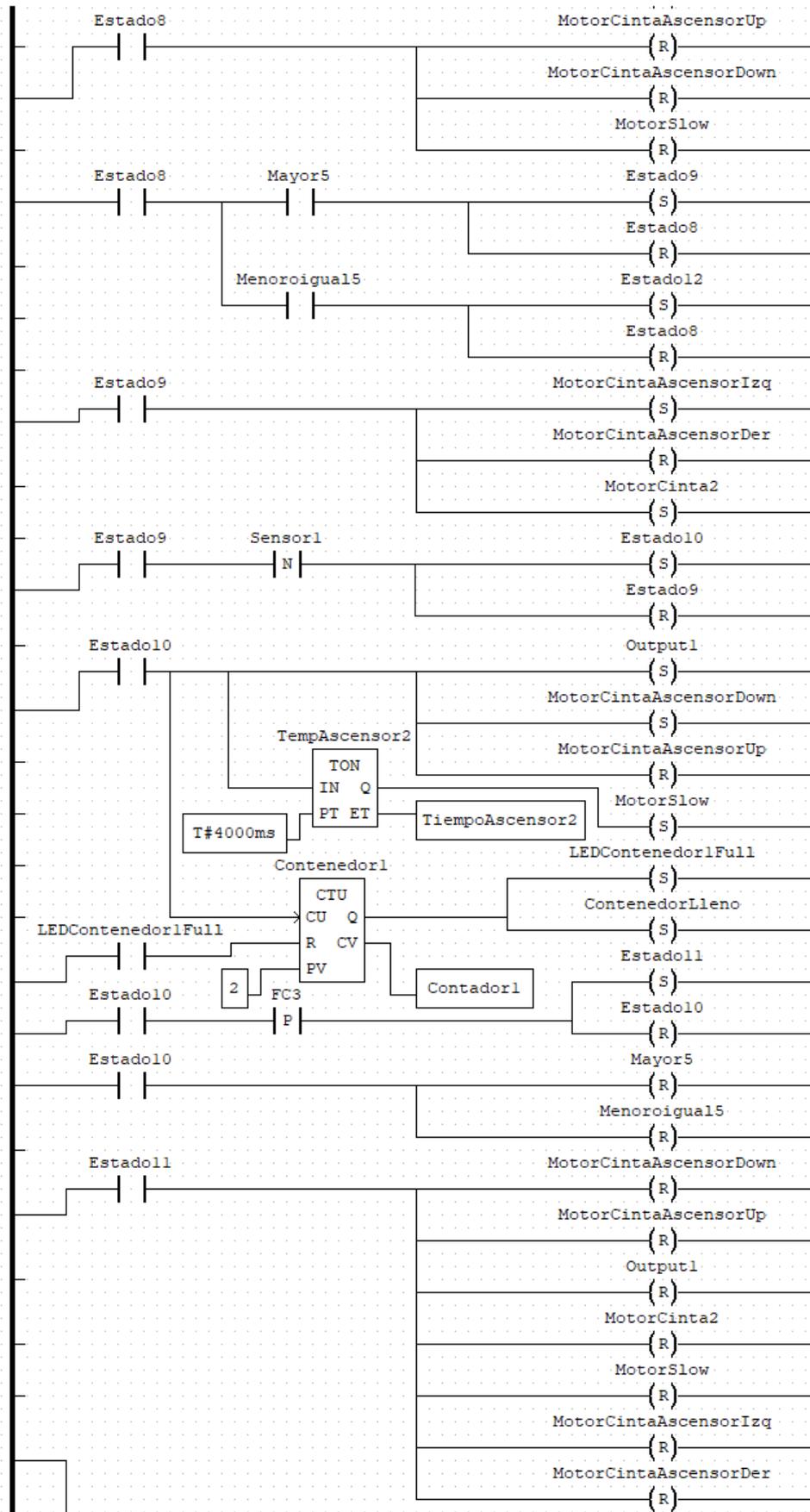


Figura 101.- Código funcionamiento normal 2 - Práctica 5.

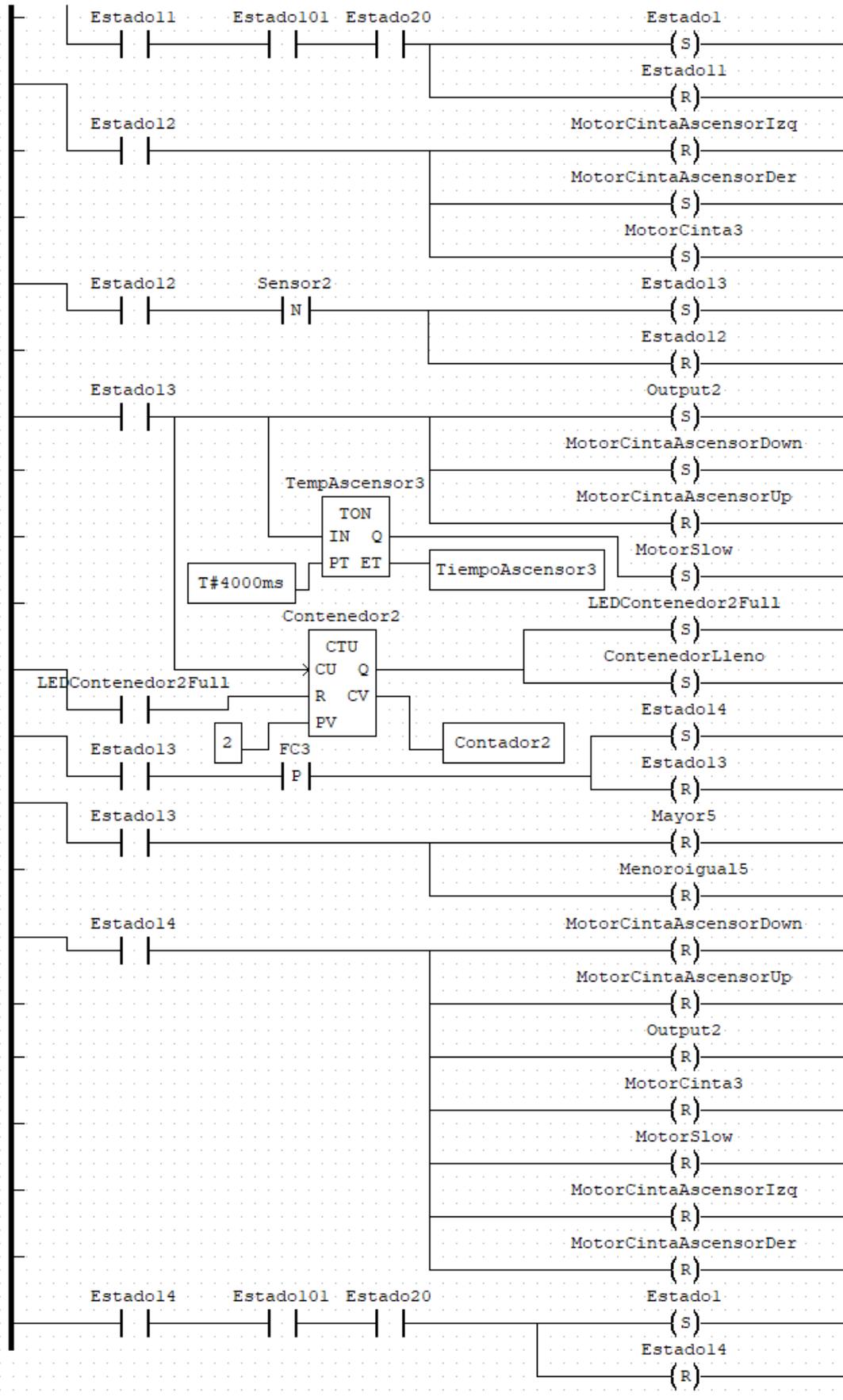


Figura 102.- Código funcionamiento normal 3 - Práctica 5.

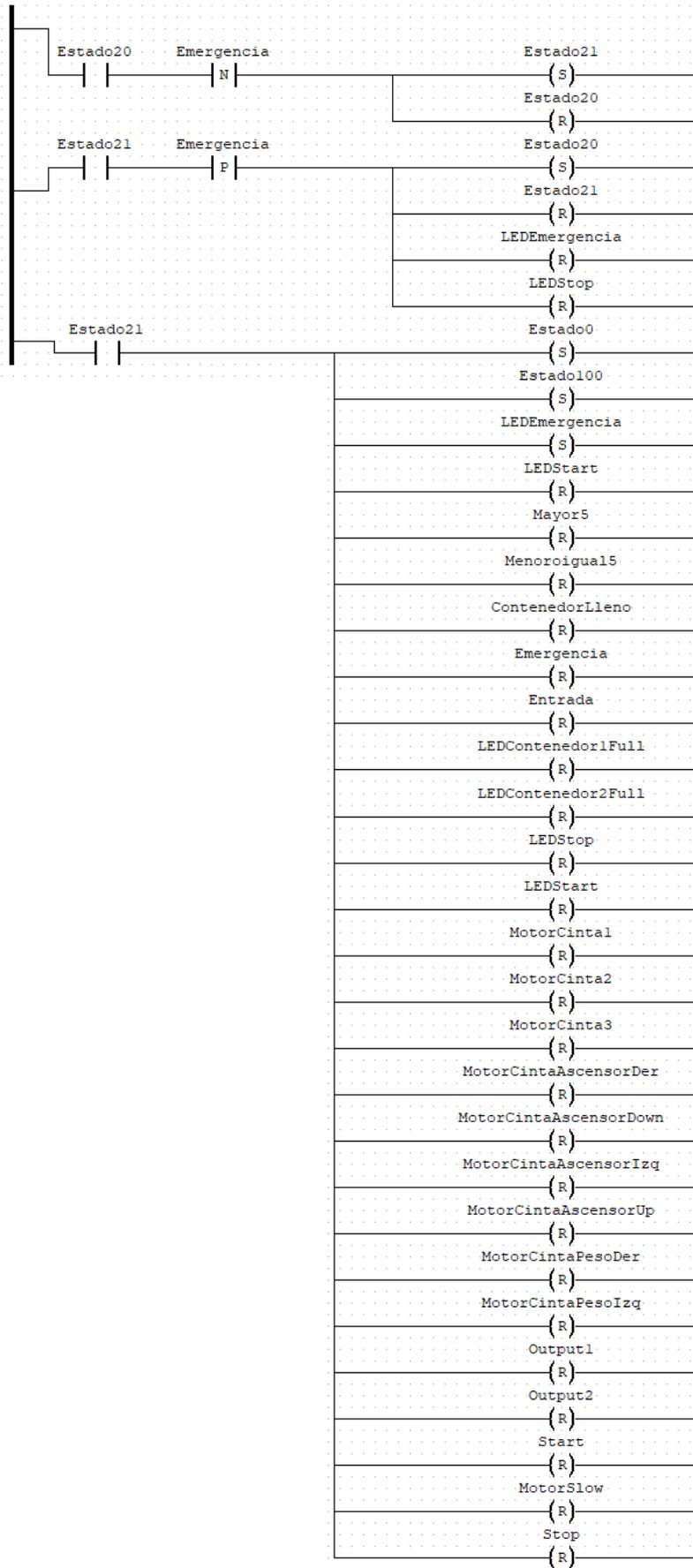


Figura 103.- Código parada de emergencia - Práctica 5.