



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Control distribuido de un sistema de clasificación,  
ensamblado y almacenaje de carcasas mediante Codesys  
y Factory IO

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Bou Grau, Sergio

Tutor/a: Correcher Salvador, Antonio

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería del Diseño

---

**Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcassas mediante Codesys y Factory IO**

### **Memoria**

**TRABAJO FINAL DEL**

Grado en Ingeniería Electrónica Industrial y Automática

**REALIZADO POR**

Sergio Bou Grau

**TUTORIZADO POR**

Antonio Correcher Salvador

**CURSO ACADÉMICO: 2021/2022**



## Introducción

Desde un punto de vista exógeno al tejido industrial podemos visualizar cada centro de trabajo como una caja en la que entran materias primas y estas salen revalorizadas después de ser procesadas en su interior por máquinas y personas, los procesos llevados a cabo en cada centro de trabajo pueden ser muy diversos según el sector en el que se esté trabajando, pero en general la gran mayoría puede automatizar parte de sus procesos o su totalidad, dejando a los mecanismos que interactúen con el producto y pueda ser manipulado o simplemente desplazado para alcanzar el objetivo final en la elaboración, estos procesos son la clave para que las empresas sean competentes en un mercado muy masificado, por eso es importante para las empresas alcanzar el nivel actual industrial 4.0 para estar a la altura de sus competidores o superarlos.

Mediante la automatización industrial se consigue controlar, supervisar y ajustar una gran variedad de procesos industriales: cadenas de producción, sistemas de riego, señales de tráfico, etc. Con la finalidad de realizar tareas en las que no hay diferencia entre el humano y la máquina, este tipo de tareas generalmente son repetitivas o demasiado pesadas para el humano. También se opta por automatizar procesos cuando se desea mantener una producción continua en la que no se para la producción por horarios o relevos de personal, permitiendo aumentar la producción. Otro beneficio que aporta la producción automatizada es que el producto no sufre variaciones en serie permitiendo uniformidad en la calidad del producto.

Para la automatización de procesos existen hoy en día múltiples soluciones que dividiremos en dos partes el software y el hardware. El hardware está compuesto principalmente por los PLC (Programmable Logic Controller), los actuadores y los sensores, estos dos últimos serán conectados al PLC con el que interactuarán según el estado del proceso. Por otra parte, tenemos el software, que es empleado por una parte para la realización del programa que controla el PLC y este a su vez controla el proceso a automatizar y por otra parte se puede emplear para la realización de pruebas con el objetivo de visualizar posibles errores en el control, simulando el proceso a controlar de forma virtual, destacando el beneficio económico y material que esto supone.

Este tipo de control mediante la automatización industrial muchas veces es incorporado a la industria de forma pausada lo que implica que en un mismo centro de trabajo coexistan diferentes elementos de control de diferentes fabricantes controlando diferentes o el mismo proceso y quepa la posibilidad de que haya que comunicarlos entre ellos para una finalidad determinada, por esto es necesario conocer las herramientas que nos ofrece el mercado para controlar y programar los PLC's que a su vez nos permitan interactuar con distintos dispositivos de igual forma.

Una mejora que han adquirido los controladores lógicos programables es poseer la capacidad de conectarse a internet, permitiendo su control, supervisión y programación en largas distancias.

No cabe duda de que en los tiempos que corren evitar desplazamientos ofrece un servicio de atención más rápido, reduce costes y evita el contacto humano con el consiguiente beneficio que aporta a escenarios pandémicos.

## Contenido

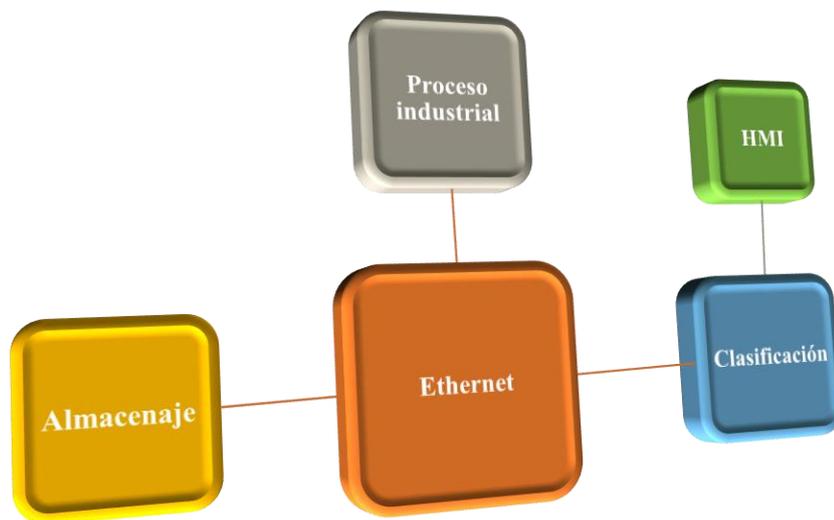
<b>1. Memoria</b> .....	6
1.1. Objeto.....	6
1.2. Antecedentes .....	7
1.2.1. Exigencias del mercado .....	7
1.2.2. Necesidades de ampliación.....	7
1.2.3. Necesidades de ahorro.....	8
1.3. Factores a considerar: limitaciones y condicionantes .....	8
1.3.1. Especificaciones.....	8
1.3.2. Normativa.....	9
1.4. Planteamiento de soluciones alternativas y justificación de la solución adoptada....	10
1.4.1. Estudio de alternativas.....	10
1.4.2. Argumentación de la solución adoptada .....	13
1.5. Descripción detallada de la solución adoptada .....	23
1.5.1. Descripción del sistema a controlar en Factory IO.....	23
1.5.2. Descripción sistema controlador de Codesys .....	28
1.5.2.1. Descripción del programa “clasificación” .....	29
1.5.2.2. Descripción del programa “almacenaje” .....	32
1.6. Manual de usuario .....	33
1.6.1. Instalación .....	34
1.6.2. Funcionamiento y uso .....	41
1.6.2.1. Modo automático.....	41
1.6.2.2. Modo manual .....	41
1.6.3. Sistemas de emergencia.....	42
1.7. Conclusión/Prólogo .....	44
1.8. Bibliografía.....	45
1.9. Tabla de ilustraciones .....	46
Anexo 1: Manual del programador .....	48

Programa “clasificación” .....	48
Programa “almacenaje” .....	60
Anexo 2: Listas: actuadores, sensores, variables, entradas y salidas. ....	67

# 1. Memoria

## 1.1. Objeto

En el presente proyecto se va a implementar el gemelo digital de un proceso industrial y se va a diseñar y programar un sistema de control que permita la gestión y control del gemelo de forma totalmente remota. Para ello por una parte se utilizará un software donde se diseñará y simulará el proceso industrial virtual y, con otro software, se implementará la programación necesaria para el control y automatización del proceso industrial simulado. El control se dividirá en dos partes, por una parte se hará una clasificación y ensamblaje de productos denominados carcasas y, por otra parte, se gestionará la demanda de pedidos y almacenaje de las mismas, empleando las herramientas necesarias para interconectar los tres sistemas a distancia a través de conexión ethernet y el protocolo Modbus TCP/IP.



*Ilustración 1. Representación de todos los sistemas.*

Este documento se centra en procesos que pueden ser automatizados con los materiales y herramientas más conocidas y que llevan un largo periodo de asentamiento en el mercado y en la industria. Estos materiales pueden ser sistemas automatizados para el transporte de mercancías, PLC's, sensores, actuadores, robots, etc. y las herramientas en este caso son programas de software que nos permiten realizar simulaciones previas al montaje real y realizar pruebas de funcionamiento previas para detectar posibles errores en el sistema. Estas herramientas se han ido actualizando y adaptando a las necesidades ofreciendo soluciones óptimas a las necesidades planteadas por las industrias como pueden ser los programas de software personalizados que piden las empresas a terceros para controlar únicamente sus equipos.

Teniendo en cuenta la época que nos ha tocado vivir: crisis económicas, discrepancias políticas y pandemias, cabe destacar que parte de este proyecto consta del control a distancia de procesos, que dejando aparte la instalación física, el mantenimiento y sus correspondientes reparaciones, toda la parte de control supervisión y ajustes se desarrollara a distancia, disminuyendo desplazamientos con lo que se evita contacto personal y se reduce el gasto en desplazamientos.

La redacción del presente proyecto se realiza en cumplimiento del artículo 18.1.a del Reglamento Electrotécnico de Baja Tensión y para que sirva como base para ejecución y la puesta en marcha de la instalación del sistema.

## 1.2. Antecedentes

Uno de los aspectos más importantes en cualquier proyecto que hay que tener en cuenta antes de su realización es que la razón de su propósito sea de utilidad de forma que resuelva un problema existente o modifique otra solución ya existente para mejorar el resultado que ya obtuviere esta.

### 1.2.1. Exigencias del mercado

En la actualidad y durante la creación de este documento existe una creciente inversión en el desarrollo industrial y la automatización de procesos, y con la esperanza de que en un futuro no muy lejano esto se mantenga y se vuelva al estado de crecimiento donde se incrementa el número de empresas en funcionamiento y sean solicitados más servicios como el que este documento ofrece con mayor frecuencia. Se prevé, en ese futuro, que serán exigidos unos protocolos a los que habrá que adaptarse y seguramente en lo que se refiere al contacto humano seguro que se deseará el mínimo, para evitar pérdidas personales y económicas. Es aquí donde el control de sistemas industriales a distancia puede ofrecer una solución debido a su gran dinamismo en lo que se refiere a comunicaciones telemáticas que en la actualidad consigue conectar la oficina técnica y el mundo de la automatización a través de la red ethernet y protocolos de comunicación TCP/IP, permitiendo conectar diferentes dispositivos como: ordenadores, autómatas, sistemas de control numérico, impresoras, etc. dejando de lado el uso de buses de campo para interconectar sistemas próximos entre sí.

### 1.2.2. Necesidades de ampliación

No cabe duda de que las empresas, para adaptarse al mercado, tienen tres opciones: ofrecer un producto único, ofrecer un producto de calidad diferenciada o crecer siendo más

competitivas aumentando la producción para abarcar más mercado. Para realizar un crecimiento óptimo donde se aumente la producción, una de las mejores opciones es automatizar los procesos donde se aumenta la producción, se disminuye la acción humana y se transforma a los empleados otorgándoles puestos de trabajo más seguros y saludables.

### 1.2.3. Necesidades de ahorro

Ya sea para ser más competitivos o verse forzados a disminuir la producción en periodos de crisis o por otros motivos, es una realidad que los empleados son los primeros en ser afectados por los recortes que deben realizar las empresas para subsistir y evitar el cierre, disponer de un sistema automatizado permite a las empresas variar la producción para adaptarse a la demanda instantáneamente ya sea disminuyendo velocidades o suspendiendo procesos por completo, evitando gastos de despidos y poder ofrecer un producto más competitivo económicamente. Es aquí donde juega un gran papel disponer de procesos automatizados donde desde una oficina de control mediante un ordenador o una Human Machine Interface (HMI) se puede adaptar la producción sin interferir en los costes de producción.

## 1.3. Factores a considerar: limitaciones y condicionantes

Para la creación de cualquier proyecto se han de definir unas especificaciones como base que serán el objetivo que alcanzar o problema a resolver, estas especificaciones serán propuestas principalmente por el demandante del proyecto y podrán ofrecerse varias soluciones para un mismo problema con el fin de adaptar idóneamente la solución a las características específicas del problema a resolver.

### 1.3.1. Especificaciones

Se plantea la creación de un sistema automático que clasifique piezas según sean bases (partes inferiores) o tapas (partes superiores) y según el color que tengan, una vez sean clasificadas las piezas serán ensambladas las partes superiores sobre las partes inferiores. A continuación se dispondrán sobre palés y se ubicarán de forma automática en una única estantería en la que se dispondrán de forma ordenada quedando claramente diferenciadas por color. El sistema comenzará cuando se indique mediante un botón de inicio desde una interfaz manejada a través de ethernet y finalizará cuando se complete la cantidad demandada desde la misma HMI. El sistema además dispondrá de un segundo subsistema que permita la clasificación

manual determinada por un usuario que permita seleccionar que piezas serán ensambladas y cuales descartadas. A continuación, se listan otras especificaciones más detalladas a cumplir:

- Únicamente se considerarán piezas válidas las de color verde y azul, considerando las de color metal defectuosas, como si no hubiesen sido pintadas por lo que serán descartadas del sistema para su posterior reutilización.
- El sistema dispondrá al usuario de una interfaz para la realización de pedidos basándose en el número de unidades de los colores disponibles. Es decir, el usuario podrá elegir cuantas unidades desea elaborar de cada color disponible, hasta cubrir el número total de espacios disponibles en el almacén.
- Las materias primas con las que el sistema se alimenta son piezas aleatorias tanto en número, color y forma. Diferenciadas por tres colores: verde, azul y gris; y por su forma: parte inferior o parte superior.
- En el modo automático el proceso de ensamblaje tendrá previamente un clasificador por colores y forma en el que se clasificarán las piezas y se ensamblarán las del mismo color, descartando las que no cumplan con las condiciones establecidas.
- Las piezas ensambladas serán paletizadas para facilitar su almacenaje en una estantería donde quedarán a disposición de un posterior sistema logístico.
- Solo se usará un apilador para el almacenaje, por lo que los transportadores de pales deberán de configurarse y controlarse para unificar la línea transportadora de pales en una.
- Los estantes de almacenaje se dividirán del 1 al 27 para las piezas azules y del 28 al 54 para las verdes, por lo que el sistema detectara la capacidad de la estantería para no ocasionar accidentes.
- Se incorporará un sistema que solucione a distancia mediante la HMI posibles fallos en el manejo de las piezas tanto por parte de las cintas como por parte de los robots ensambladores, así como de los transportadores de palés, con el fin de que la intervención humana no sea necesaria al menos para la finalización de pedidos.

### 1.3.2. Normativa

Para la redacción del presente proyecto, su puesta en marcha e instalación hay que tener en cuenta los siguientes reglamentos y normativas listados a continuación:

- **RD 842/2002** Reglamento electrotécnico para baja tensión.
- **UNE-EN 50174-2** Tecnología de la Información. Instalación del cableado. Parte 2: Métodos y planificación de la instalación en el interior de los edificios.
- **UNE 20 514 92 1M** Reglas de seguridad para los aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uso general análogo conectados a una red de energía.

Así como las siguientes instrucciones técnicas complementarias del reglamento electrotécnico de baja tensión.

- **ITC BT 19** Instalaciones interiores o receptoras. Prescripciones generales.
- **ITC BT 20** Instalaciones interiores o receptoras. Sistemas de instalación.
- **ITC BT 23** Protección de instalaciones interiores. Protección contra sobretensiones.
- **ITC BT 43** Instalación de receptores. Prescripciones generales.
- **ITC BT 51** Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios

#### 1.4. Planteamiento de soluciones alternativas y justificación de la solución adoptada

A continuación veremos que no solo tenemos una única opción para realizar el proyecto, sino que el resultado no solo depende de elegir la opción más actual con la última versión, ya que se debe tener en cuenta que se puedan cumplir el máximo de especificaciones que demanda el proyecto y para ello se analizarán diversas opciones. De ese modo podrá asegurarse que no se descartan otras opciones que en un momento dado se puedan implementar en el proyecto si fuese necesario para cumplir especificaciones más exigentes como disminuir el coste o simplificar el proceso que se desea controlar.

##### 1.4.1. Estudio de alternativas

Las necesidades sobre software que requiere este proyecto son básicamente dos, un simulador donde implementar el proceso y un software para controlar el proceso simulado. Además, si fuese necesario, debe poder implementarse en un proceso real con las modificaciones pertinentes.

En lo que respecta al simulador del proceso se han examinado las siguientes tres opciones:

1. **LogixPro:** este software está orientado a la composición de actuadores y elementos habituales en los procesos industriales en formato 2D, el programa cuenta con varios procesos ya definidos y solo permite la programación mediante el lenguaje Ladder.

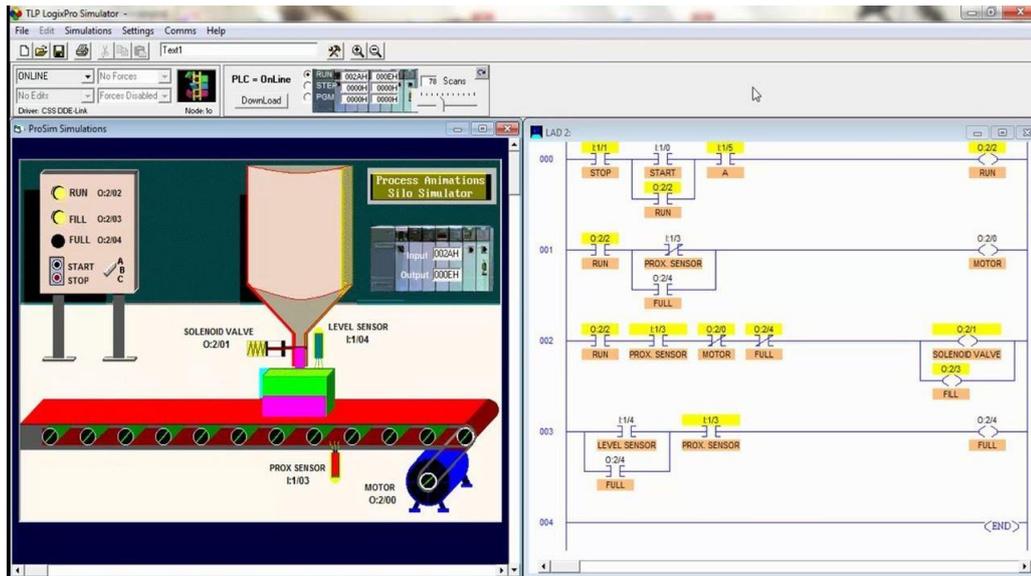


Ilustración 2. Ejemplo de proceso en LogixPro.

2. **EasyPLC Machines Simulator:** es un software que permite la creación de un escenario en 3D mediante elementos habituales en la industria como cintas transportadoras, sensores, cuadros de mando, etc. También permite simular un PLC y realizar la programación para controlar el sistema simulado lo que permite la comprobación del buen funcionamiento del sistema. Otra característica de este software es la posibilidad de conectarse a dispositivos físicos mediante los puertos del PC y el fabricante proporciona las herramientas para interactuar con el proceso simulado mediante gafas virtuales que permiten al usuario usar botones de paneles o hasta coger elementos del escenario.



Ilustración 3. Representación de realidad virtual en EasyPLC Machines Simulator.



Ilustración 4. Ejemplo de proceso en EasyPLC Machines Simulator.

3. **Factory IO:** se trata de un software que permite la configuración de un escenario en 3D de un proceso industrial mediante cintas de transportación de productos, actuadores, sensores y sistemas de almacenaje entre otros, con alta calidad gráfica. También dispone de controladores para interactuar con PLCs, simuladores de PLCs, Modbus y algunas otras tecnologías.

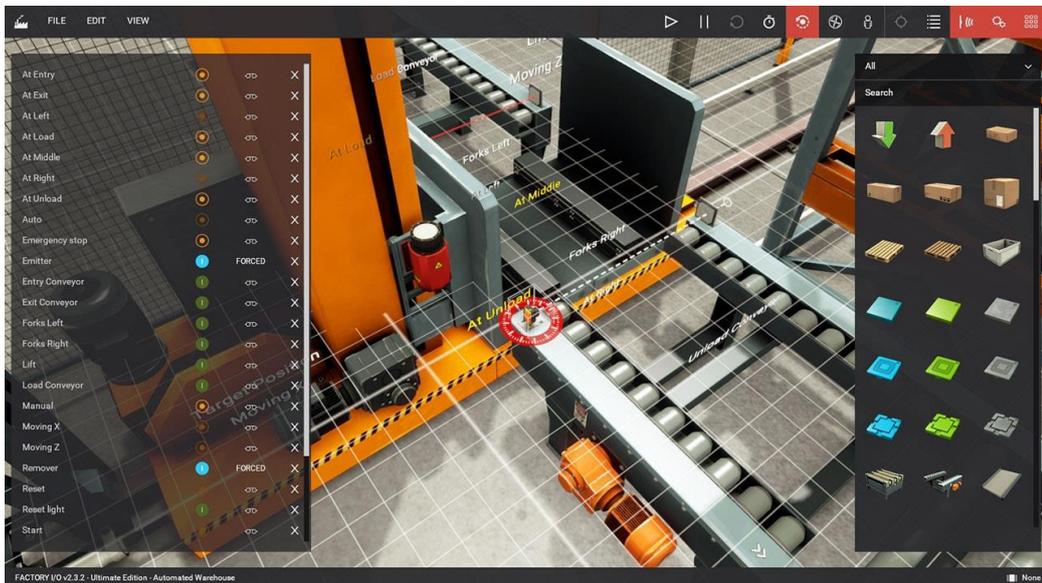


Ilustración 5. Ejemplo de proceso en Factory IO.

En lo que concierne al software para controlar el proceso este depende directamente del simulador elegido ya que no todos los simuladores ofrecen las mismas características.

En el caso del simulador LogixPro este mismo permite realizar la programación de escenarios ya diseñados en dos dimensiones, pero únicamente permite la programación mediante lenguaje Ladder.

Sobre el caso del simulador EasyPLC Machines Simulator el mismo desarrollador dispone del software para implementar la programación para controlar el proceso y es compatible con otros sistemas de programación, pero si quisiéramos conectar un PLC físico debemos de adquirir una interfaz externa para comunicar el PLC con el PC.

En cuanto A Factory IO, el desarrollador no ofrece ninguna herramienta para desarrollar el control del proceso mediante los lenguajes normalmente usados para programar PLCs, pero si dispone de del método de control mediante diagrama de bloques. Este simulador lo que si ofrece es una alta calidad gráfica y la compatibilidad con las principales marcas de fabricantes de PLCs y protocolos de comunicación. Además, proporciona tutoriales bien confeccionados para conectar las principales marcas de PLCs o herramientas de software como Codesys mediante diversos protocolos de comunicación.

#### 1.4.2. Argumentación de la solución adoptada

Tras estudiar las diversas opciones que se han descrito, se ha optado por el uso de un software para simular el proceso a controlar y otro para controlar el sistema simulado, ambos comunicados mediante la arquitectura maestro/esclavo y el protocolo de comunicación estándar Modbus.

En lo que respecta al simulador se ha optado por realizar el sistema a controlar mediante el software Factory IO por su calidad de gráficos que permite una aproximación a la realidad muy optima y la versatilidad de movimientos con la cámara permite asimilar con gran detalle todo el escenario, otra cualidad que ha atraído el uso de este software son los diversos elementos para configurar el sistema a controlar permitiendo configurar múltiples escenarios que se asemejan a situaciones posibles que pueden darse en la realidad en múltiples factorías como las cintas transportadoras o brazos robóticos, además este software da la posibilidad de simular diferentes marcas de PLCs y utilizar protocolos de comunicación para controlar el proceso.

Para el control del proceso simulado se ha elegido usar el software Codesys que nos da la posibilidad de simular PLCs con configuración a medida, incluir módulos de comunicación Ethernet y Modbus, y además permite diseñar interfaces humano máquina (HMI en adelante por sus siglas en inglés). Principalmente estas características y otras como la posibilidad de usar los principales lenguajes de programación usados en el control de sistemas industriales mediante

PLCs, más la posibilidad de usar el programa con total disponibilidad de forma gratuita, durante dos horas ininterrumpidas, han sido clave para optar por este software.

### 1.4.2.1. Factory IO

Este software desarrollado por la compañía Real Games está basado en un simulador en tres dimensiones de una factoría, en él es posible diseñar una gran cantidad de escenarios posibles de ver en la realidad de forma sencilla y con gran capacidad de configuración para conseguir una óptima compatibilidad a la hora de llevar el diseño a la realidad. En Factory IO inicialmente disponemos de escenarios ya creados de aplicaciones típicas en la industria como cintas de transporte para objetos o rodillos para transportar palés, hasta escenarios más complejos con robots, depósitos de líquidos o apiladores automáticos. Si deseamos crear nuestro propio escenario tenemos a nuestra disposición una gran cantidad de elementos que nos permiten generar casi cualquier situación dada en la vida real, para ello disponemos de gran cantidad de sensores, transportadores, y estructuras entre otros, que permiten hallar una solución muy similar, si no exacta, a nuestro objetivo. Además, este software permite ser manejado simulando poseer un PLC de las principales marcas conocidas o mediante sistemas de comunicación muy usados en la industria como el protocolo Modbus TCP/IP. Lo que permite ser controlado mediante una amplia gama de sistemas usados habitualmente en la industria para la automatización de procesos industriales.

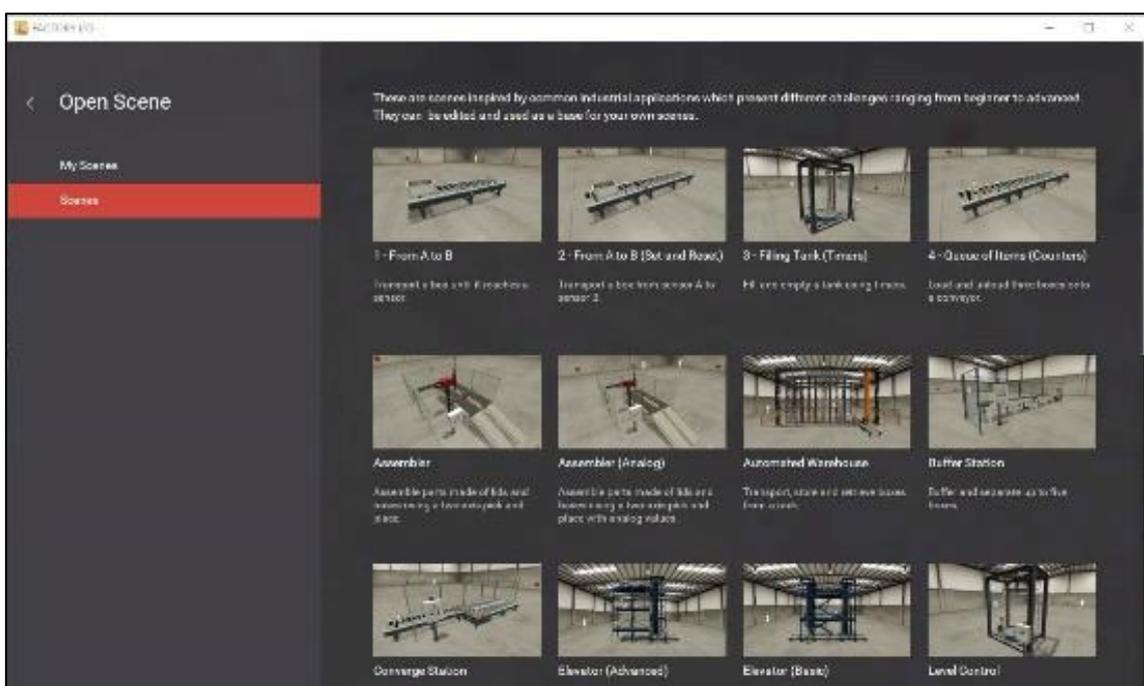


Ilustración 6. Ejemplo de escenas.



Ilustración 7. Escenario configurable.

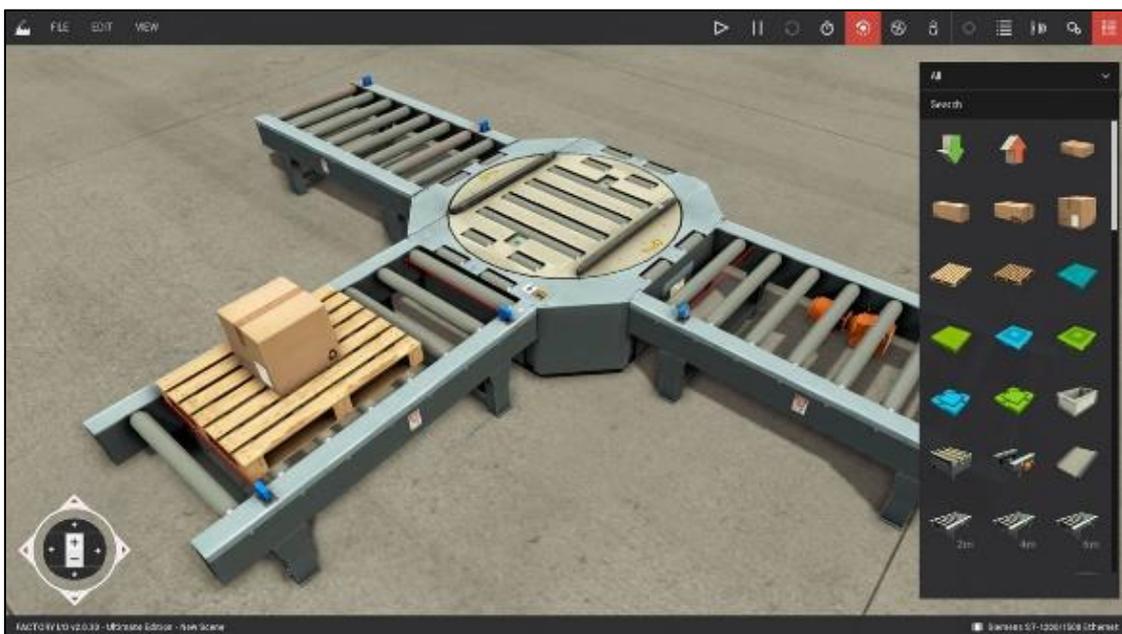


Ilustración 8. Multitud de objetos disponibles.



Ilustración 9. Amplia conectividad

Otra particularidad de este software es la realización de pruebas para evitar posibles errores ya sea por la configuración del escenario o la programación del controlador que lo maneje, con el fin de reducir costes a la hora de implementarlo en la realidad al no sacrificar materiales o evitar posibles daños personales al no poner en riesgo a posibles operarios que intervienen en el proceso.



Ilustración 10. Ejemplo de fallo en el proceso.

Una de las características que ofrece este software que lo hacen especialmente particular es la posibilidad de manejar valores tanto digitales como analógicos en la mayoría de las partes que componen el sistema, ya sea para manejar la velocidad de cintas transportadoras, comprobar el peso de materiales o medir niveles de líquidos.



*Ilustración 11. Control analógico y digital.*

#### 1.4.2.2. Codesys

Codesys es producto de la compañía 3S-Smart Software Solutions GmbH, este software es ofrecido en versión demo gratuita y permite la programación de PLCs en diversos lenguajes de programación. Las características que posee este software por las que ha sido seleccionado han sido su compatibilidad con el control de PLCs de los principales fabricantes, la posibilidad de conexión a través de ethernet mediante las conexiones más utilizadas en la industria de forma fácil y económica, la total disponibilidad del programa a la hora de programar cualquier sistema y de cualquier magnitud y la posibilidad de crear HMIs muy versátiles y completas.

Otras características de la que dispone este software es la posibilidad de simular el proyecto con la finalidad de realizar comprobaciones rápidas y que en el entorno real supondrían inconvenientes como el deterioro de la maquinaria o el producto manipulado, en este modo podemos realizar lo que es conocido en el entorno informático como “debug” realizando un seguimiento de variables, direcciones de entrada o salida y valores tanto analógicos como

digitales así como forzar estos valores para realizar un control manual del sistema para hallar fallos en la codificación.

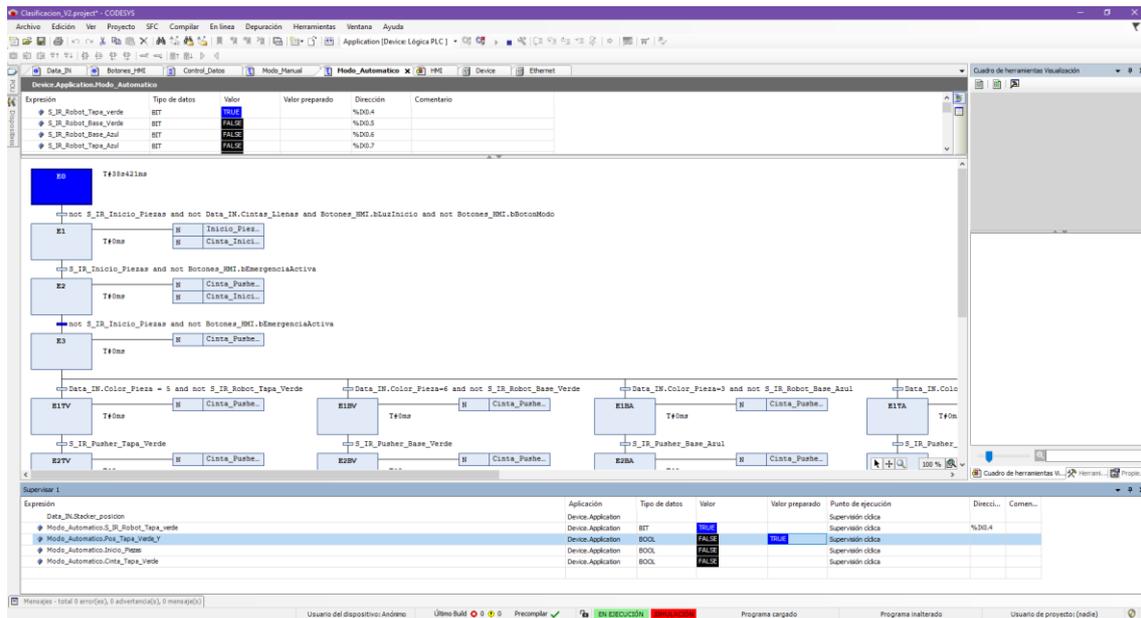


Ilustración 12. Ejemplo de sistema simulado y supervisión de variables.

Como ya he mencionado una de las características que hace este software muy útil es su versatilidad a la hora de configurar la conexión vía ethernet ya que nos ofrece una amplia gama de tipos de comunicación entre los que se encuentran los más utilizados en la industria para la comunicación de dispositivos automatizados de control para procesos como Profinet, Profibus, Modbus, etc.

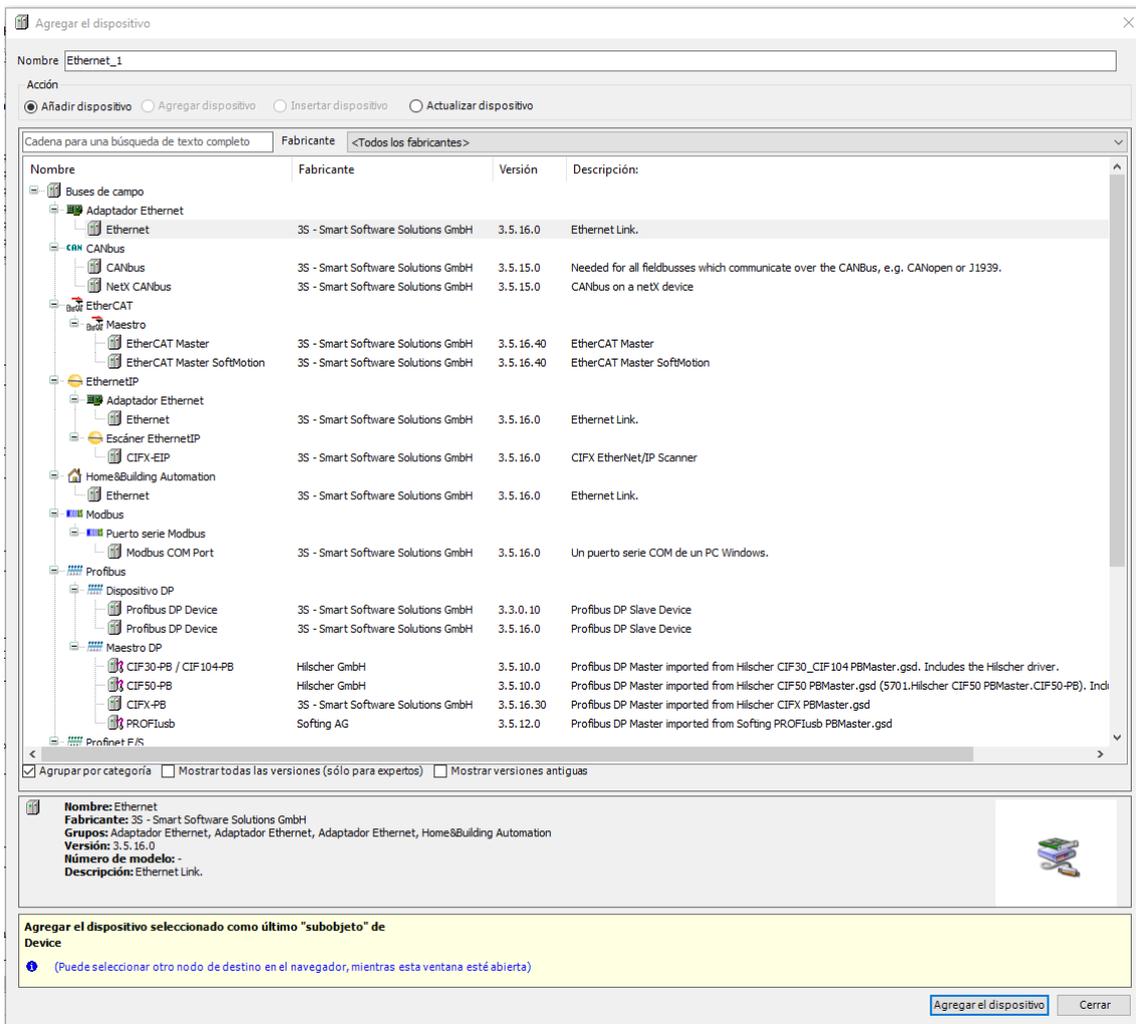


Ilustración 13. Tipos de conexión seleccionables en Codesys vía ethernet.

Otra cualidad que se agradece en este tipo de productos es la posibilidad de crear nuestro proyecto mediante varios lenguajes de programación para dispositivos pensados para el control industrial y usarlos todos a la vez sin que haya incompatibilidades entre ficheros a la hora de compartir variables. Sobre estos, en este software, cabe destacar que el manejo de los diferentes objetos a la hora de programar mediante SFC, *sequential function chart* por sus siglas en inglés, es muy cómodo, intuitivo y fácil de realizar debido a que la interfaz del programa permite arrastrar los objetos y mediante unos movimientos de ratón podemos realizar rápidamente una secuencia.

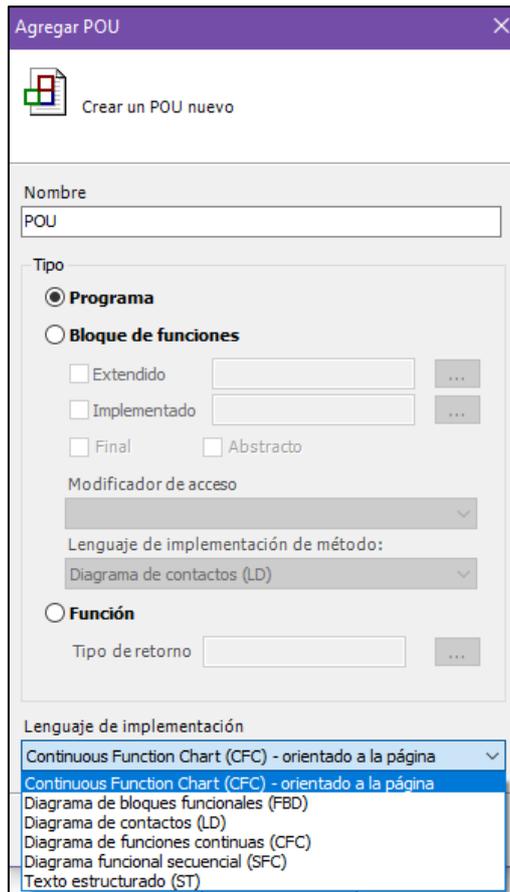


Ilustración 14. Lenguajes seleccionables en Codesys.

La posibilidad de poder configurar el número y tipo de entradas y salidas de los módulos de comunicación son otra gran ventaja que nos ofrece este producto ya que esto nos permite adaptar el software a nuestra necesidad con lo que es una característica muy útil a la hora de acoplarlo a diversos proyectos. A su vez el mismo software nos permite acceder a cada canal bit a bit permitiéndonos observar su tipo, valor y dirección identificativa.

Variable	Asignación	Canal	Dirección	Tipo	Unidad	Descripción
		Single_Inputs	%IB0	ARRAY [0..5] OF BYTE		Read Discrete Inputs
		Single_Inputs[0]	%IB0	BYTE		Read Discrete Inputs
		Bit0	%IX0.0	BOOL		0x0000
		Bit1	%IX0.1	BOOL		0x0001
		Bit2	%IX0.2	BOOL		0x0002
		Bit3	%IX0.3	BOOL		0x0003
		Bit4	%IX0.4	BOOL		0x0004
		Bit5	%IX0.5	BOOL		0x0005
		Bit6	%IX0.6	BOOL		0x0006
		Bit7	%IX0.7	BOOL		0x0007
		Single_Inputs[1]	%IB1	BYTE		Read Discrete Inputs
		Single_Inputs[2]	%IB2	BYTE		Read Discrete Inputs
		Single_Inputs[3]	%IB3	BYTE		Read Discrete Inputs
		Single_Inputs[4]	%IB4	BYTE		Read Discrete Inputs
		Single_Inputs[5]	%IB5	BYTE		Read Discrete Inputs
		Single_Outputs	%QB0	ARRAY [0..3] OF BYTE		Write Multiple Coils
		Single_Outputs[0]	%QB0	BYTE		Write Multiple Coils
		Bit0	%QX0.0	BOOL		0x0000
		Bit1	%QX0.1	BOOL		0x0001
		Bit2	%QX0.2	BOOL		0x0002
		Bit3	%QX0.3	BOOL		0x0003
		Bit4	%QX0.4	BOOL		0x0004
		Bit5	%QX0.5	BOOL		0x0005
		Bit6	%QX0.6	BOOL		0x0006
		Bit7	%QX0.7	BOOL		0x0007
		Single_Outputs[1]	%QB1	BYTE		Write Multiple Coils
		Single_Outputs[2]	%QB2	BYTE		Write Multiple Coils
		Single_Outputs[3]	%QB3	BYTE		Write Multiple Coils
		Color_Inicio_Piezas	%IW3	ARRAY [0..0] OF WORD		Read Input Registers
		Color_Inicio_Piezas[0]	%IW3	WORD		0x0003

Ilustración 15. Visualización de direcciones configuradas.

Una de las habilidades que se ha de poseer a la hora de utilizar este software es la imaginación ya que va a ser fundamental a la hora de crear una interfaz debido a su amplitud de posibilidades a la hora de crear cualquier elemento común usado en cualquier HMI. Incluir un simple pulsador en nuestra HMI puede ser simple o extremadamente pesado, ya que las posibilidades son tan amplias que deberemos determinar el grado de tiempo invertido en cada elemento para no invertir demasiados recursos en algo que no sea fundamental en nuestro diseño. En este software además de los elementos que nos ofrece por defecto podremos incluir nuestras imágenes haciendo nuestro diseño único y dirigido a un cliente específico. Además mediante el uso de elementos en la HMI no solo podremos manejar variables o entradas y salidas, también se pueden realizar ejecuciones de código o interactuar con otras visualizaciones.

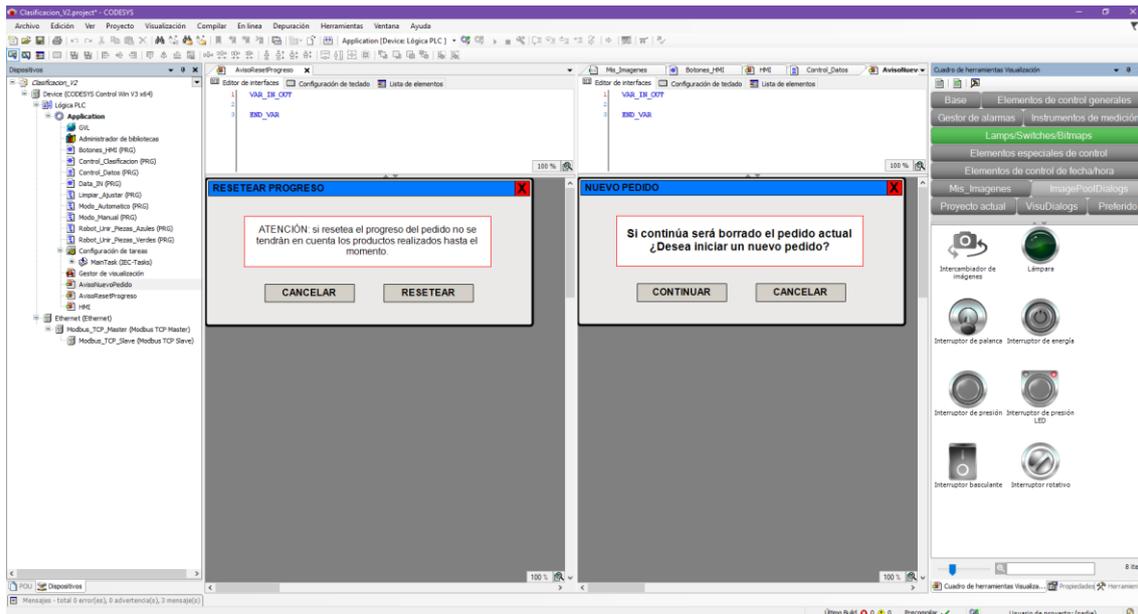


Ilustración 16. Ejemplo de interfaz gráfica en modo edición.

### 1.4.2.3. Lenguajes de programación

En el mundo de los autómatas programables dedicados a la industria son diversos los lenguajes de programación utilizados para transferir los deseos humanos a acciones que deben realizar las máquinas, el uso de cada uno de estos lenguajes viene determinado por el objetivo final, que dependiendo de su complejidad es conveniente usar uno determinado o cabe la posibilidad de realizar las mismas acciones con diferentes lenguajes. También se pueden diferenciar al usarlos por ser de tipología gráfica o basados en texto. Los basados en texto son lenguajes de bajo nivel, es decir más próximos al lenguaje máquina, como por ejemplo el denominado lista de instrucciones (IL o STL) o el denominado texto estructurado (ST), los lenguajes gráficos son de alto nivel por lo que son fáciles de comprender visualmente ya que utilizan elementos gráficos para componer las instrucciones, entre estos últimos destacaremos el diagrama de contactos o Ladder (LD), el diagrama de bloques (SFD) y el diagrama funcional secuencial (SFC). En este proyecto se han usado principalmente tres LD, SFC y ST, por eso a continuación detallaremos un poco más estos tres.

**Diagrama de contactos (LD):** este tipo de lenguaje es de tipología interfaz gráfica y el nombre de Ladder (escalera) se debe a que su forma estructural se asemeja a una escalera en la que los peldaños son la unión de la parte izquierda donde tendríamos la fuente de energía y la parte derecha que representa la tierra, a su vez los peldaños están compuestos por contactos. La finalidad de este lenguaje es ser simple de usar para lógicas sencillas y no invertir tiempo en su uso.

**Diagrama funcional secuencial (SFC):** mediante este lenguaje de representación diagramática podemos controlar un proceso en el cual las acciones transcurran de forma secuencial, estas secuencias son representadas por cajas rectangulares y entre estas se hallan las condiciones, representadas por líneas horizontales, que determinan si la energía transcurre de una etapa a la siguiente. Mediante este lenguaje se pueden realizar controles complejos con varias ramificaciones o una simple sucesión de secuencias, por este motivo es un lenguaje utilizado en ciertos procesos de complejidad media en los que existan acciones secuenciales.

**Texto estructurado (ST):** de sintaxis similar a Pascal este lenguaje es utilizado para manejar expresiones aritméticas con valores analógicos o digitales, también ofrece la posibilidad de usar estructuras como funciones, bucles y ciclos de iteración. Principalmente usado cuando mediante los lenguajes gráficos no es fácil de realizar operaciones complejas o controlar procesos no secuenciales.

## 1.5. Descripción detallada de la solución adoptada

Como objetivo final, este proyecto tiene la intención de demostrar una forma de realizar el control de un proceso industrial de complejidad alta y dimensión media, donde intervienen elementos destinados a la clasificación y almacenaje de un producto dado como ejemplo. La solución adoptada se puede dividir en dos partes, por un lado se ha diseñado un sistema para el procesamiento del producto dado como ejemplo implementada en Factory IO que permite la clasificación por color y forma del producto dado como ejemplo y seguido su almacenaje de forma ordenada en una única estantería, por otro lado tenemos el sistema distribuido que controla de forma remota el sistema de procesamiento permitiendo mediante una interfaz gráfica la solicitud de pedidos que el sistema procesará de forma automática, además el sistema de control ofrece el modo manual que permite la clasificación del producto de forma unitaria y a elección del usuario en tiempo real.

En este proyecto se propone una solución determinada para el procesamiento del producto, basada en la utilización de los recursos justos para que funcione el sistema de forma eficiente y a una velocidad de procesamiento del producto ajustada a unas circunstancias donde los resultados son en base a la eficacia del sistema y no a su velocidad, además el sistema de clasificación se ha diseñado de forma que puedan realizarse acciones a distancia para solucionar posibles fallos mecánicos que impidan la continuación del sistema.

### 1.5.1. Descripción del sistema a controlar en Factory IO

Para la descripción del sistema diseñado en Factory IO lo dividiremos de dos partes, la de clasificación y la de almacenaje. La parte de clasificación está constituida por cintas

transportadoras, empujadores mecánicos y robots ensambladores, que mediante diversos sensores ubicará las distintas partes en una posición determinada permitiendo a los robots su ensamblaje de forma ordenada en el caso de que el usuario haya elegido el modo automático de clasificación.



*Ilustración 17. Disposición de las diferentes piezas.*

El proceso, cuando está seleccionado el modo automático, comienza con la inserción de piezas con forma tapa o base aleatorias de color verde, azul y gris en la primera cinta transportadora, hasta que llegan a un sensor basado en tecnología optoelectrónica mediante luz infrarroja, sensor IR en adelante, que comunica al sistema que hay una pieza dispuesta para su clasificación, si el sistema permite su clasificación la cinta sitúa a la pieza en posición debajo de un arco que posee un sensor capaz de detectar el color y forma de la pieza otorgándole un código numérico y comunica este dato al sistema de control que lo almacena en memoria. Los códigos obtenidos por los sensores correspondientes a cada pieza son los siguientes:

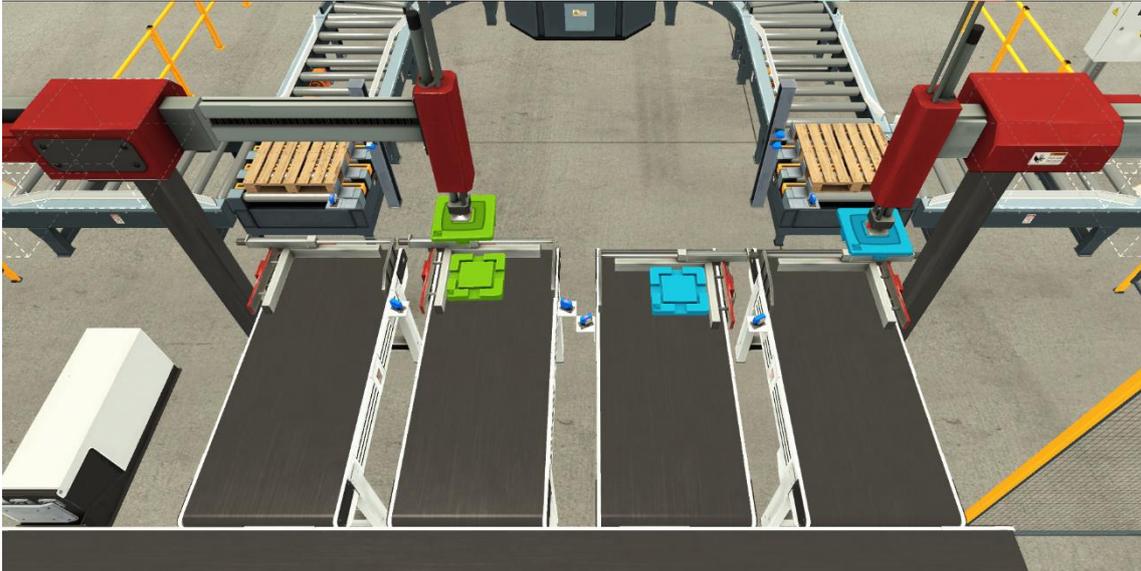
Tipo/color	Verde	Azul	Gris
Tapa	5	2	8
Base	6	3	9

Tabla 1. Códigos de piezas obtenidos de los sensores.



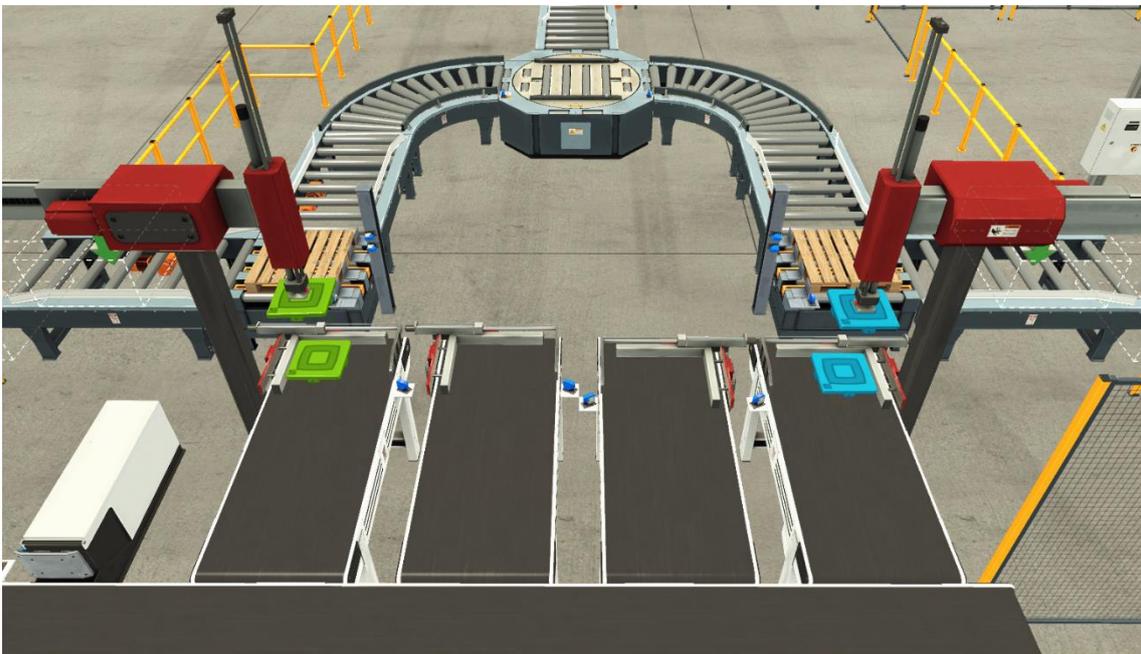
*Ilustración 18. Lectura de piezas por forma y color.*

Cuando ha quedado registrado el código de la pieza se procede a transportar la pieza hasta el primer empujador que descartará la pieza si es tapa o base de color gris ya que se ha considerado que estas piezas carecen de pintura y han de ser descartadas por defectuosas para un posterior tratamiento y su vuelta al sistema. Si la pieza es una tapa o base de color verde o azul el sistema considerará si está disponible su ubicación al final de las cintas para el ensamblaje, si no está disponible descartará la pieza para permitir el paso a la siguiente pieza debido a que la inserción de piezas es aleatoria y se considera que hay una recirculación manual de piezas. Si la ubicación de la pieza está disponible la segunda cinta transportadora desplazará la pieza hasta el empujador correspondiente, cuando actúe el sensor IR perteneciente a el empujador indicado se detendrá la segunda cinta y actuará el empujador sobre la pieza situándola en la cinta que le corresponda ya sea tapa verde, base verde, base azul o tapa azul, esta última cinta transportara la pieza hasta su posición final donde un mecanismo ajustará la posición de la pieza para que el robot la coja por un punto exacto permitiendo su ensamblaje de forma precisa.



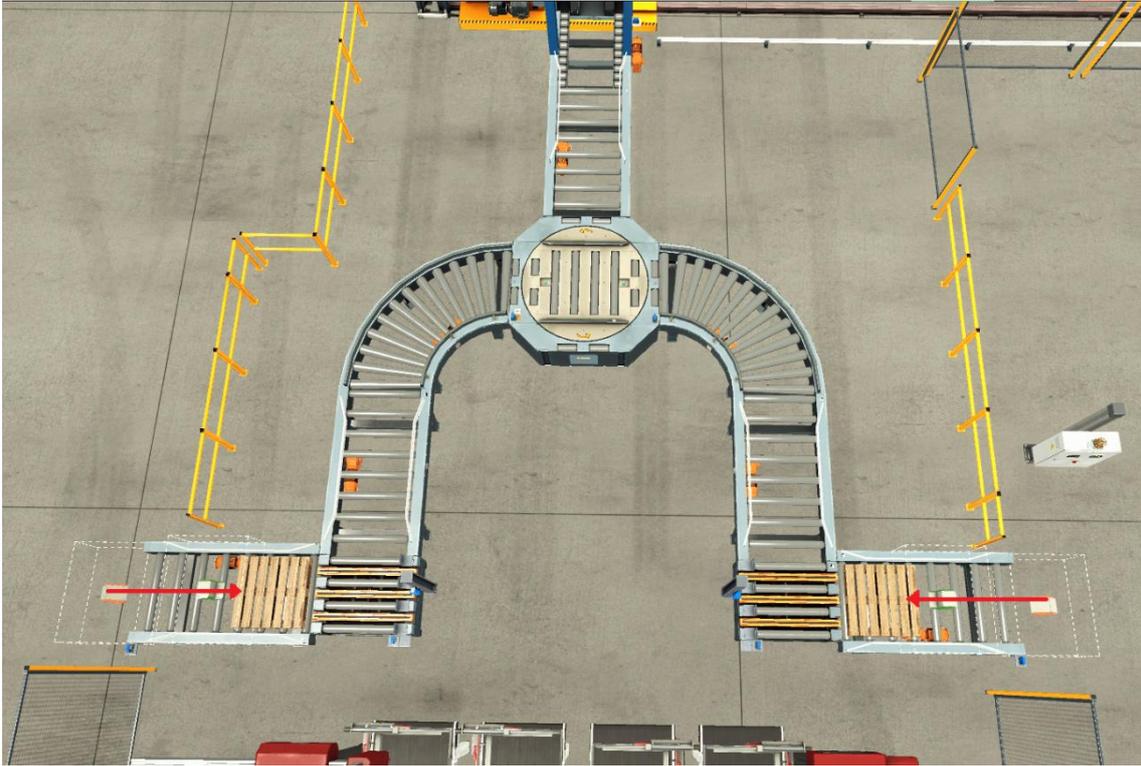
*Ilustración 19. Robots ensamblando las piezas.*

Cuando el sistema detecte que hay una tapa disponible ordenará al robot correspondiente, ya sea el lado verde o el lado azul, que comience el proceso de ensamblaje, de este modo, aunque no haya una pieza base disponible para terminar el proceso de ensamblaje quedarán disponibles las ubicaciones de las tapas permitiendo procesarlas y que no sean descartadas.



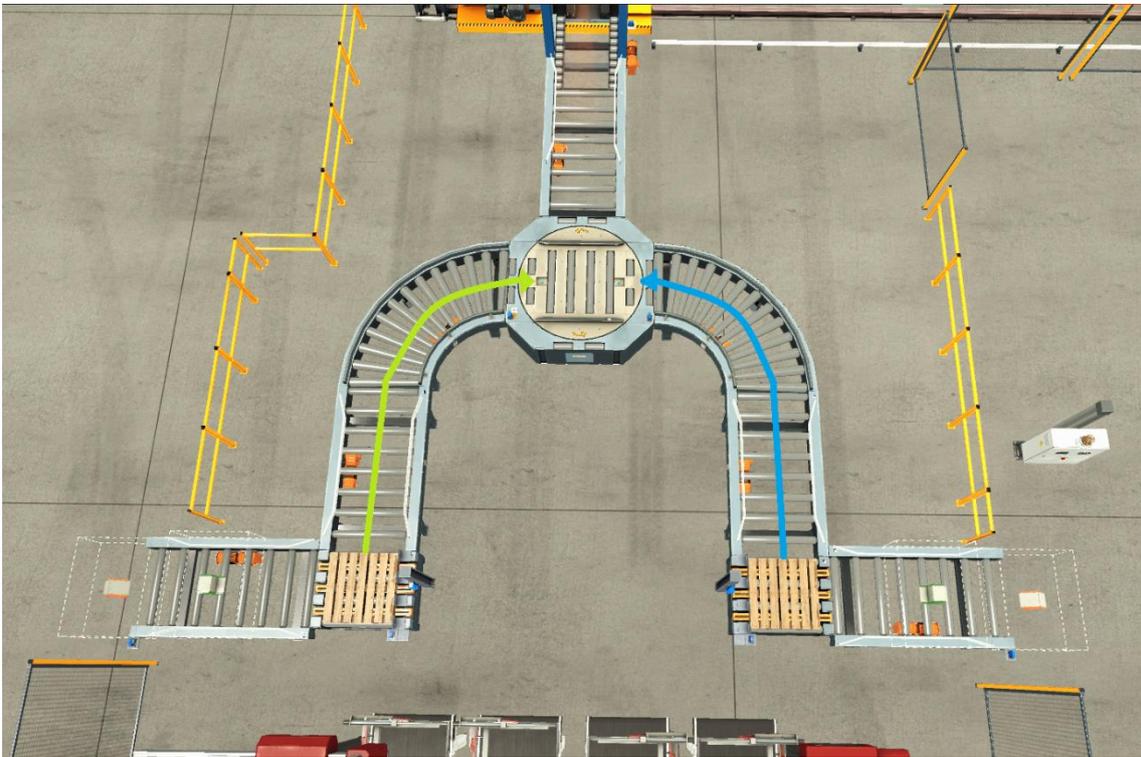
*Ilustración 20. Robots con tapas cogidas esperando bases.*

Una vez algún robot complete el ensamblaje de una pieza comenzará la segunda parte del proceso, la de almacenaje, este segundo proceso parte con dos caminos de transportadores, uno para la parte de las piezas verdes y otro para la parte de las piezas azules.



*Ilustración 21. Palés posicionándose para recibir las piezas ensambladas.*

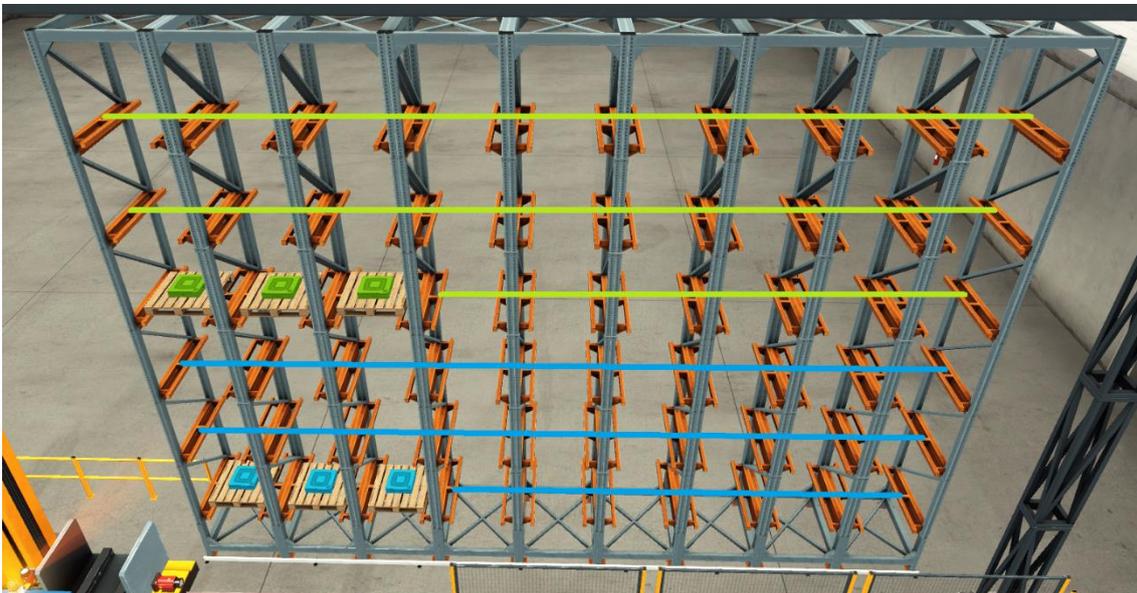
Tanto el camino que ha de seguir el palé azul como el verde comienza con la inserción del propio palé en el sistema, el cual se realizará de forma automática suponiendo que un operario realiza las tareas de abastecimiento de este material.



*Ilustración 22. Recorrido de los palés según el color de pieza que transportan.*

El sistema, mediante sensores, es capaz de detectar la presencia del palé en posición para recibir la pieza ensamblada y si esta ha sido ensamblada correctamente, de no ser así se procede a desechar el palé y su contenido devolviéndolo al punto de partida considerando que la pieza depositada en él es defectuosa. Si la pieza depositada sobre el palé es considerada correcta el sistema autoriza que el palé avance si no está en movimiento otro palé ya que los dos caminos convergen en el girador de palés, una vez se haya girado el palé para encararlo al apilador esta pasará por un sensor para determinar de que color es la pieza que se transporta y que sea ubicada en su correspondiente zona por el apilador. El sistema autoriza el avance de los palés cuando el girador esta libre, es decir el girador es el recurso compartido por esta parte del sistema.

Se ha establecido que en el modo automático el sistema apile las piezas diferenciadas por color situando unas en las filas superiores y las otras en las inferiores como se muestra en la imagen.



*Ilustración 23. Estantería donde se muestra la ubicación de los palés según el color de pieza que contienen.*

### 1.5.2. Descripción sistema controlador de Codesys

La particularidad de este proyecto es que se realiza el control mediante un sistema gemelo, es decir, se realiza el control mediante dos PLCs virtuales implementadas en dos unidades distintas que controlan un único sistema. Para lograr esto se han creado dos proyectos en Codesys, uno para realizar el control de la parte de ensamblado y clasificación y otro para el paletizado y almacenaje, a continuación se procede a describir estos proyectos.

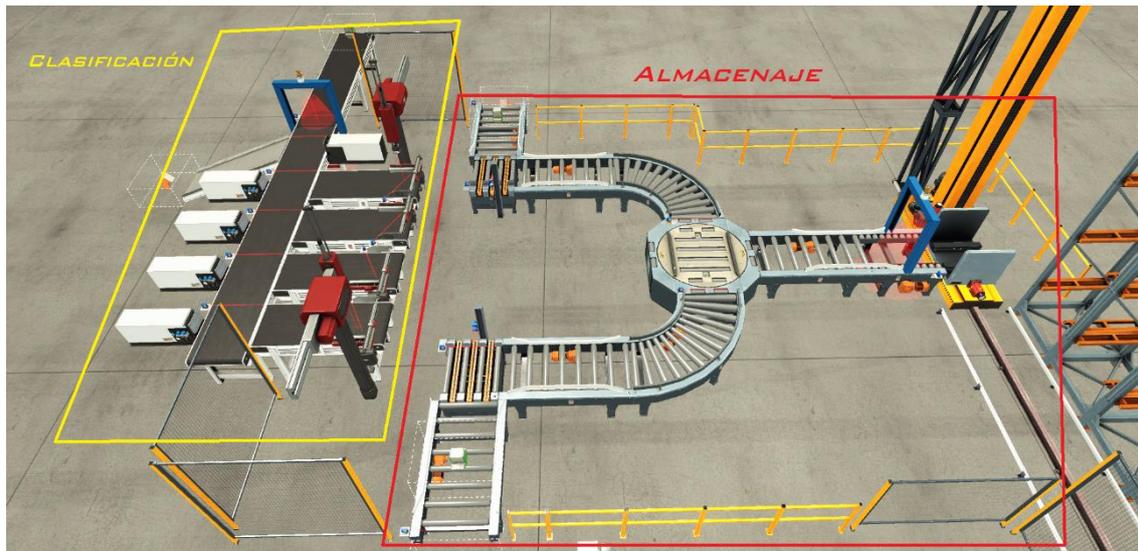


Ilustración 24. Delimitación controlada por cada programa.

### 1.5.2.1. Descripción del programa “clasificación”

El programa “clasificación” se encarga de gestionar las piezas que van entrando al sistema y mediante sensores, empujadores y las cintas las clasifica por su color y su forma, ubicándolas finalmente para que puedan ser cogidas por los robots ensambladores.

El programa que controla la parte de la clasificación y del ensamblaje consta de los siguientes ficheros:

- **Botones HMI**

Este fichero está creado mediante el lenguaje Ladder y su propósito es gestionar casi todos los inputs realizados desde la interfaz de usuario.

- **Control Clasificación**

Escrito mediante el lenguaje de programación Ladder, la finalidad de este fichero es enlazar las variables locales desde distintos ficheros con las entradas del sistema a controlar.

- **Control Datos**

Fichero creado mediante el lenguaje Texto estructurado, donde se realizan operaciones aritméticas o booleanas que dependen de condicionales.

- **Data IN**

En este fichero se gestionan los valores obtenidos desde los diferentes sensores que leen el color o forma de las piezas o mediante sensores basados en luz infrarroja se determina la presencia de las piezas cuando son movidas por las cintas o llegan a una ubicación en concreto, este fichero está creado en lenguaje Ladder.

- **Limpiar Ajustar**

Este fichero forma parte del sistema de control de errores a distancia, cuya finalidad es poder continuar con la producción en el caso de que ocurra un error en la posición de las piezas para su ensamblaje o que los robots no cojan bien las piezas superiores, cuyo acto provoca que no sea posible su ensamblaje. La finalidad es poder dar el poder de realizar dos acciones cuando ocurra un error, una es limpiar las cintas y la otra es reposicionar las piezas nuevamente. Escrito mediante diagrama funcional secuencial.

- **Modo Automático**

En este fichero, escrito en diagrama funcional secuencial, se controlan las diferentes cintas y empujadores en función del color o forma de la pieza leída por el sensor.

- **Modo Manual**

Mediante este fichero es posible controlar el sistema de forma manual desde la HMI, una vez seleccionado este modo es posible seleccionar las piezas una a una del color que se desee pudiendo realizar ensamblajes con distinto color. Este fichero está escrito en diagrama funcional secuencial.

- **Robot Unir Piezas Azules**

Este fichero, creado mediante diagrama funcional secuencial, controla los movimientos del robot encargado de ensamblar y transportar las piezas azules.

- **Robot Unir Piezas Verdes**

Este fichero, creado mediante diagrama funcional secuencial, controla los movimientos del robot encargado de ensamblar y transportar las piezas verdes.

Y las siguientes visualizaciones, que son elementos creados para realizar la interacción entre el humano y la maquina:

- **Aviso Nuevo Pedido**

Esta visualización surge cuando se pulsa el botón “Nuevo pedido” en la HMI, su finalidad es advertir al usuario de que se borrara el pedido en curso.

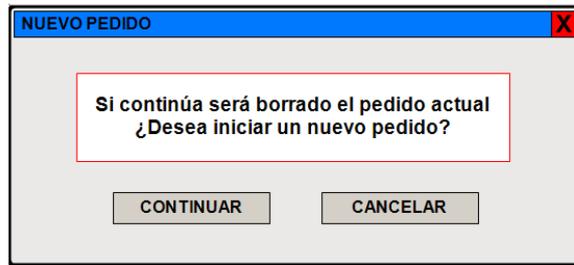


Ilustración 25. Ventana de aviso al pulsar el botón "Nuevo pedido."

#### - **Aviso Reset Progreso**

Esta visualización surge cuando se pulsa el botón "Borrar progreso", su finalidad es avisar al usuario de que se reiniciarán los contadores y no se tendrá en cuenta las piezas realizadas hasta el momento.

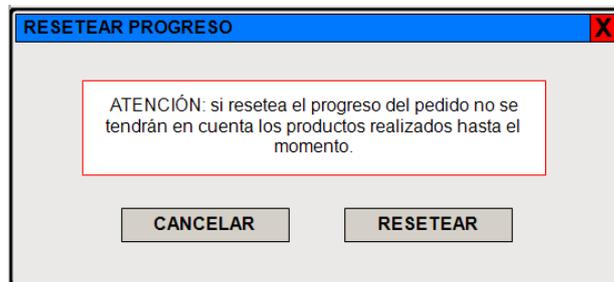


Ilustración 26. Ventana de aviso que surge al pulsar el botón "Borrar progreso".

#### - **HMI**

En esta visualización están definidos todos los botones, luces y contadores que permiten transmitir los deseos del usuario al sistema, entre otras cosas se puede realizar pedidos, visualizar el progreso en tiempo real, cambiar de modo de funcionamiento, resolver atascos en la producción a distancia o realizar paradas de emergencia.

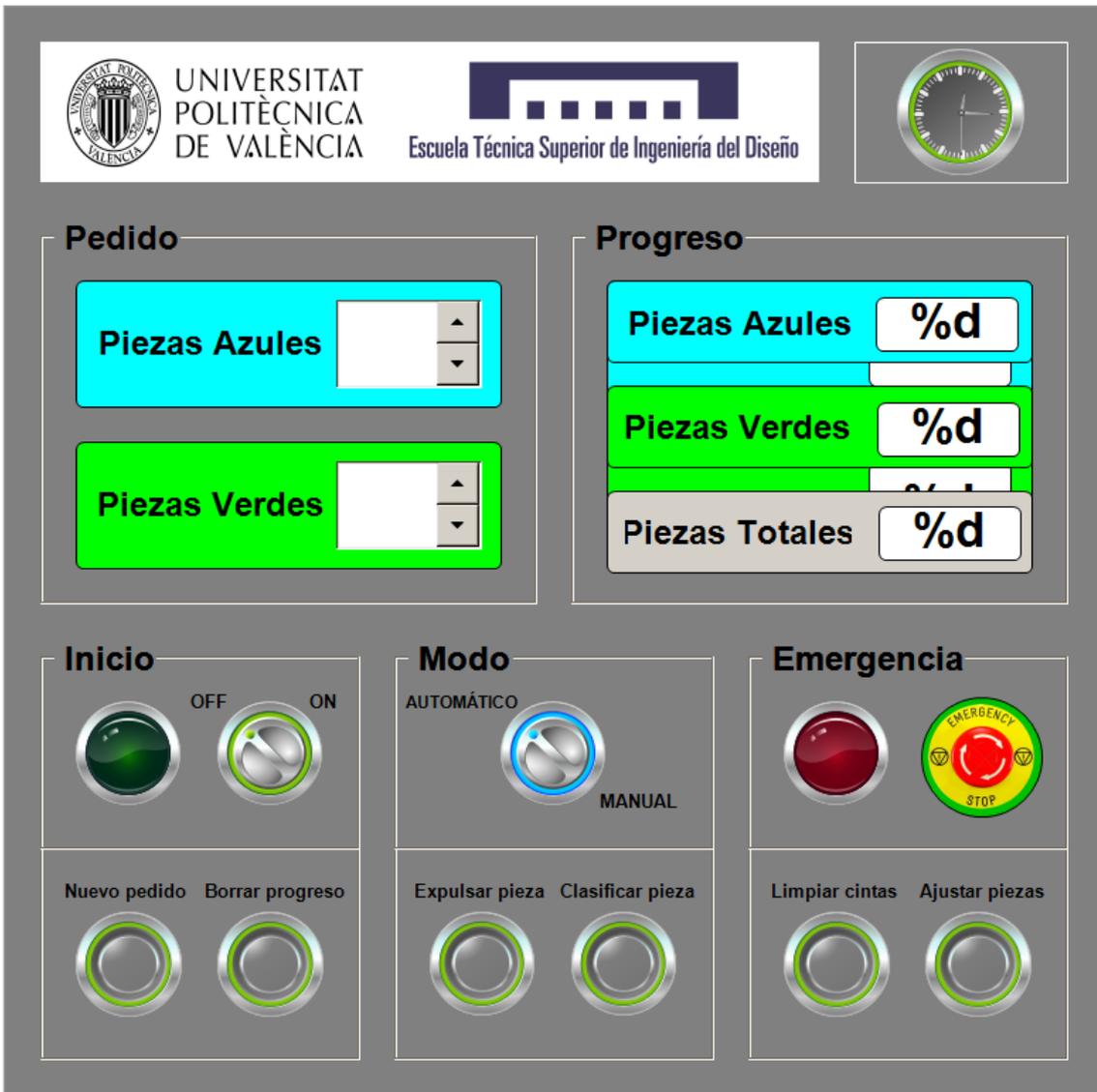


Ilustración 27. HMI en modo edición.

### 1.5.2.2. Descripción del programa “almacenaje”

El programa “almacenaje” controla la parte paletizada que se encarga de recibir las piezas ensambladas y mediante transportadores de palés mover estos hasta el apilador que ubicará cada palé en una posición predeterminada.

Los ficheros que componen este programa son los siguientes.

- **Almacenamiento**

Este fichero, escrito mediante diagrama funcional secuencial, controla el apilador que ubica cada palé en una posición determinada según el color que de la pieza que transporta.

- **Control Almacenaje**

Escrito con lenguaje texto estructurado, en este fichero se realizan operaciones aritméticas y booleanas que dependen de condicionales. En él se realizan los incrementos al contabilizar las piezas o se determina la posición de los palés.

- **Gestión Almacenaje**

Este fichero comprende las variables internas que dependen de una o varias señales para gestionar los movimientos de los palés si según las respuestas de los diferentes sensores. Está escrito mediante lenguaje Ladder.

- **Girar Palé**

Escrito mediante diagrama funcional secuencial, este fichero controla el girador de palés que dependiendo del lado por el que venga, el azul o el verde, hace girar en un sentido u otro su base.

- **Inicio Palé Azul**

Para el control del suministro de palés, su traslación hasta el girador o su retirada en caso de que se deposite una pieza defectuosa sobre el palé en el lado de las piezas azules se ha creado este fichero mediante diagrama funcional secuencial.

- **Inicio Palé Verde**

Para el control del suministro de palés, su traslación hasta el girador o su retirada en caso de que se deposite una pieza defectuosa sobre el palé en el lado de las piezas verdes se ha creado este fichero mediante diagrama funcional secuencial.

- **Palé Girado**

Este fichero tiene la finalidad de comunicar al sistema la llegada del palé al final de los transportadores de palés, está escrito en diagrama funcional secuencial.

## 1.6. Manual de usuario

La puesta en marcha de este sistema es altamente compleja si se desconoce el funcionamiento y como realizar la configuración para adaptarlo a las conexiones remotas de la ubicación final, por este motivo se va a detallar como realizar estas configuraciones para que quede constancia de como proceder en el caso de implementar este sistema en una nueva ubicación. Primeramente, deberemos de abstraernos hasta un plano en el que veamos todo el sistema con los elementos principales que en definitiva son tres PLCs virtuales interconectadas, dos que controlan y una que gobierna el proceso industrial.

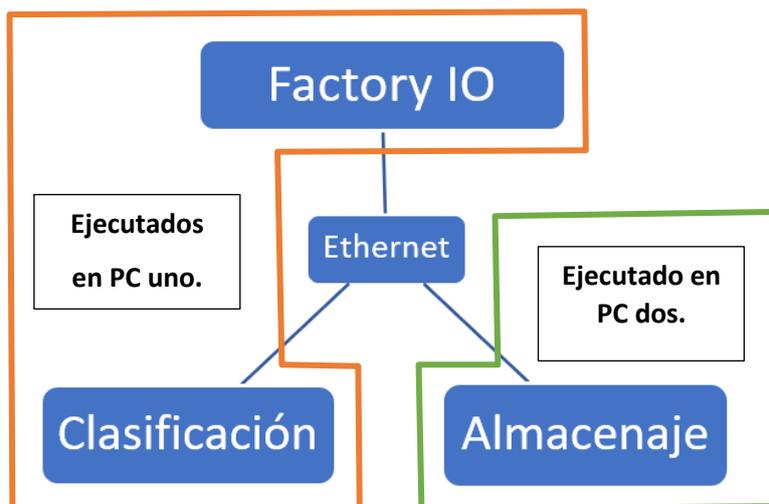


Ilustración 28. Representación del conjunto del sistema.

Para hacer funcionar todos los programas de Codesys como el de Factory IO necesitaremos por lo menos dos PCs, uno donde ejecutaremos Factory IO y uno de los dos programas que lo controlan y en el otro PC ejecutaremos el programa de Codesys restante.

### 1.6.1. Instalación

Las dos PLCs que controlan el sistema en Factory IO son simuladas mediante aplicaciones proporcionadas por el software Codesys, para activarlas iremos al menú del sistema operativo y en primer lugar ejecutaremos el *Gateway* y seguido el *Control Win V3 – x64 SysTray*, una vez ejecutados los dos nos aseguraremos de que están en activo dirigiéndonos al menú de la barra de tareas en el caso de Windows donde veremos los iconos ocultos, si alguno de los dos está en blanco y negro significa que esta desactivado y tendremos que hacer click derecho sobre el e iniciarlo, en el caso del *Control Win V3 – x64 SysTray*, *si no tenemos licencia se desconectará cada dos horas, por lo que si se nos desconecta el sistema lo primero que deberemos de sospechar que lo pudo causar será esto.*

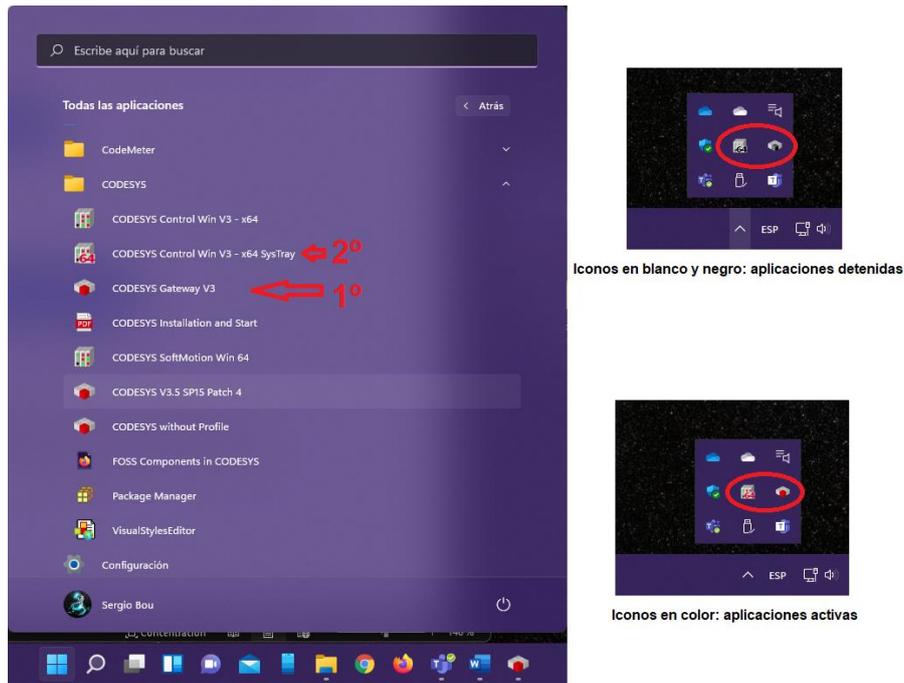


Ilustración 29. Menú del sistema operativo y aplicaciones de Codesys.

Una vez activados el Gateway y el Control WIN V3 ya podemos abrir el programa Codesys y haciendo doble click sobre “Device” en el gestor de archivos veremos una representación del sistema de conexiones entre nuestro PC el Gateway y el PLC virtual, debajo de este mismo hay un desplegable donde deberemos de seleccionar el PLC activo en nuestro PC y veremos como debe pasar a color verde la luz indicadora de estado en la esquina inferior derecha de la figura.

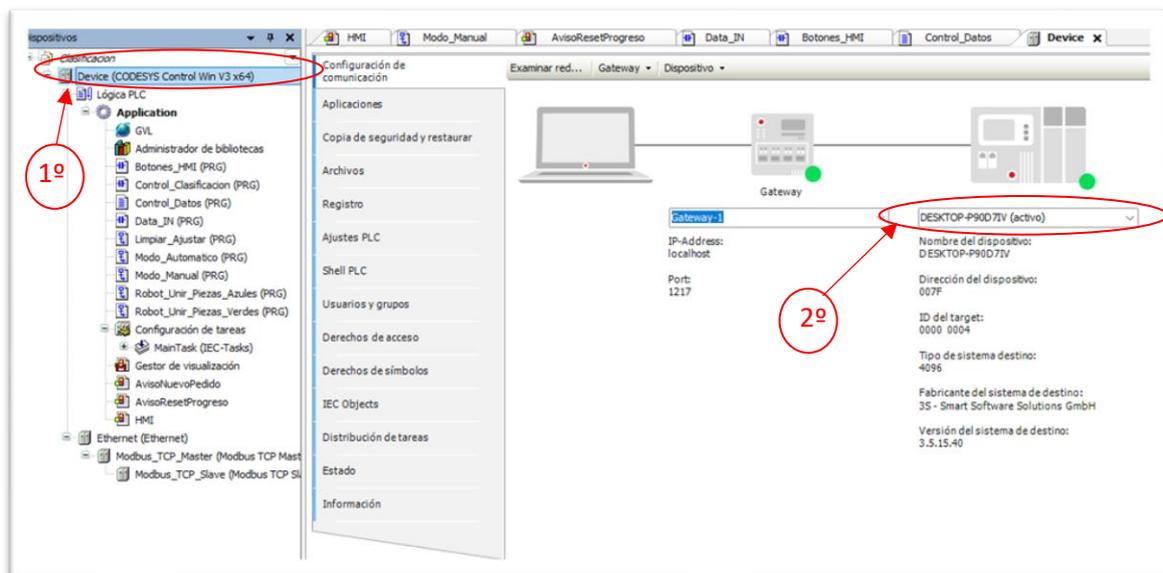


Ilustración 30. Conexión del programa Codesys con el PLCs virtual.

Una vez hayamos configurado la conexión entre el PC y el PLC virtual a través del Gateway adecuadamente pasaremos a configurar el módulo Ethernet para establecer los parámetros de conexión de internet y veremos la configuración de las entradas y salidas configuradas para conectarse adecuadamente al sistema que controlar.

Podemos observar en el gestor de archivo como está añadido el módulo Ethernet y cuelga de este el módulo Master y el módulo Slave para la comunicación Modbus TCP, primeramente accederemos a el módulo Ethernet e introduciremos nuestra dirección IP, si no la conocemos podemos pulsar sobre los tres puntos y se abrirá una ventana donde podremos observar las conexiones existentes en nuestro entorno, en esta ventana seleccionaremos la que nos corresponde y después de pulsar “Aceptar” se auto rellenarán los espacios correspondientes a la dirección IP, la máscara subred y la puerta de enlace, por lo que corresponde a esta parte ya estaría.

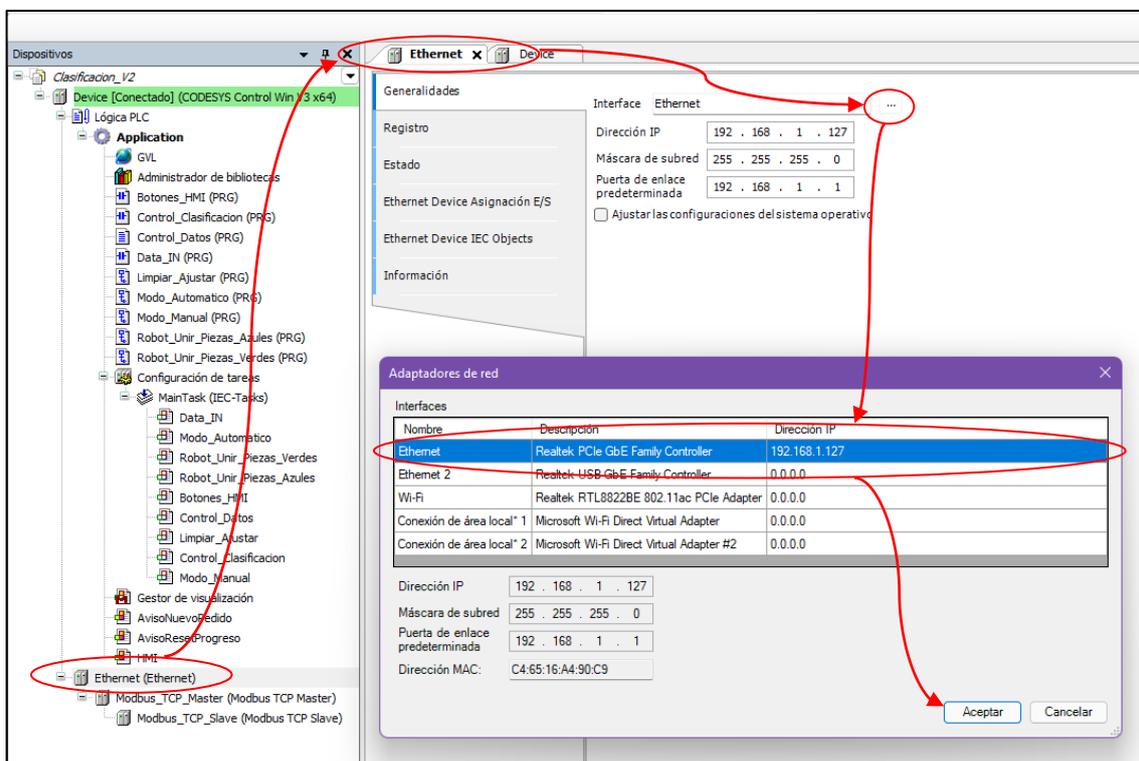


Ilustración 31. Configuración del modulo Ethernet.

Sobre el módulo Master en principio no tendremos que realizar ningún cambio, pero podemos observar en la pestaña de configuración las diferentes opciones que se pueden ajustar como habilitar la optimización, el tiempo de respuesta o el auto reconectado.

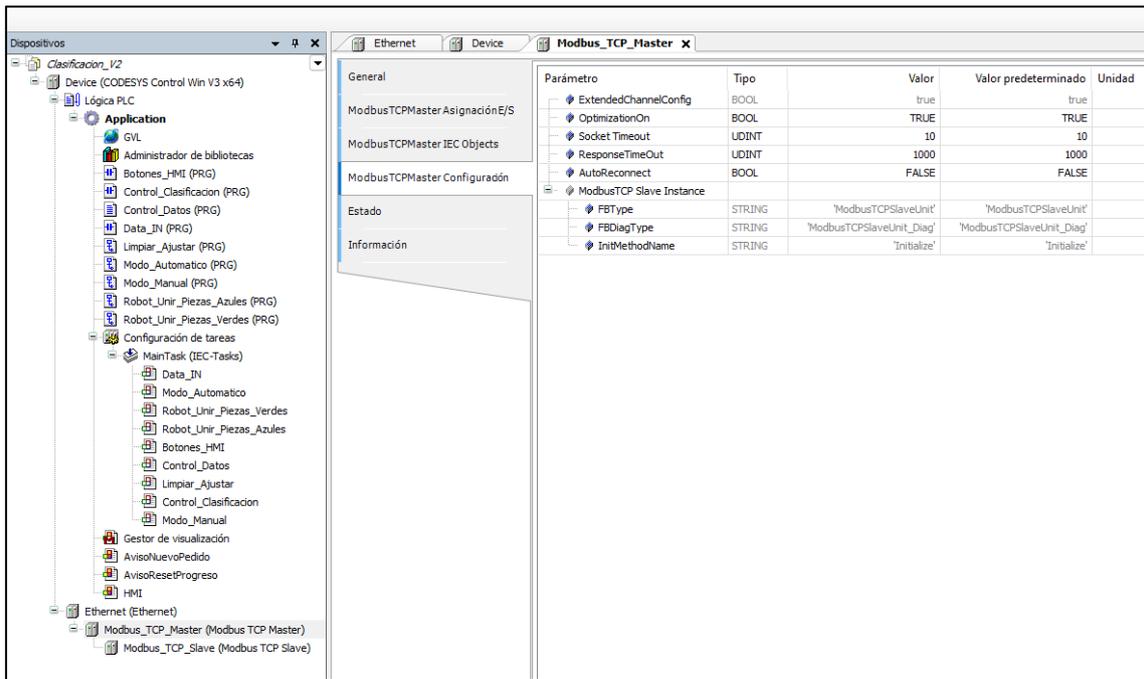


Ilustración 32. Módulo Master.

En el módulo Slave veremos en la pestaña general la dirección IP, el número de identificación el tiempo de espera de respuesta y el puerto de conexión, en esta parte nos aseguraremos de que coinciden con la configuración de Factory IO ya que deben de reconocerse, en la siguiente ilustración podemos ver estos parámetros. En la pestaña ModbusTCPSlaveAsignacionE/S podremos consultar la numeración asignada a cada entrada o salida y el valor que tienen, en la ilustración 15 se puede observar un ejemplo de esta pestaña desplegada.

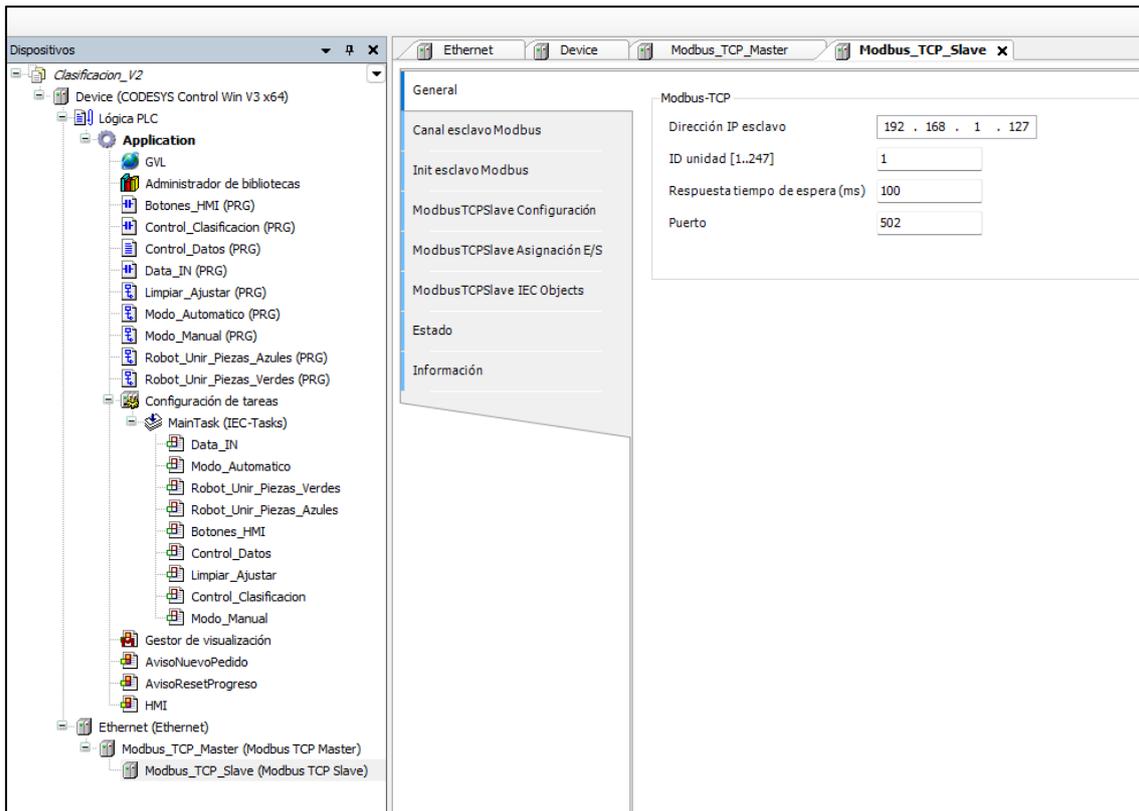


Ilustración 33. Módulo Slave

Lo que si deberemos de hacer, en el caso de que se quiera modificar el proyecto, es configurar los canales en el módulo Slave para que coincidan con la configuración del PLC destino que en este caso es la del Factory IO. En la siguiente imagen podemos observar que hay tres canales configurados con los siguientes parámetros: el canal 0 está configurado para leer entradas discretas con una longitud de 48; el canal 1 está configurado para escribir múltiples salidas con una longitud de 32; y el canal 2 está configurado para leer un registro completo como entrada.

Nombre	Tipo de acceso	Trigger	READ Offset	Longitud	Tratamiento de errores	WRITE Offset	Longitud
0 Single_Inputs	Read Discrete Inputs (Código funcional 02)	Cíclico, t#100ms	16#0000	48	Conservar el último valor		
1 Single_Outputs	Write Multiple Coils (Código funcional 15)	Cíclico, t#100ms				16#0000	32
2 Color_Inicio_Piezas	Read Input Registers (Código funcional 04)	Cíclico, t#100ms	16#0003	1	Conservar el último valor		

Ilustración 34. Entradas y salidas en el módulo Slave del programa clasificación.

En el módulo Slave del programa almacenaje también podemos observar esta disposición y la configuración de los canales: el canal 0 está configurado para leer entradas discretas con una longitud de 48; el canal 1 está configurado para escribir múltiples salidas con una longitud de 32; el canal 2 está configurado para leer un registro completo como entrada; el canal 3 está configurado para leer un registro completo como entrada; los canales 4, 5, 6 están configurados para escribir un registro completo. Hay que tener en cuenta los offsets para no pisar las salidas del módulo Master con las de el módulo Slave.

Nombre	Tipo de acceso	Trigger	READ Offset	Longitud	Tratamiento de errores	WRITE Offset	Longitud
0 Single_Inputs	Read Discrete Inputs (Código funcional 02)	Cíclico, t#100ms	16#0000	48	Conservar el último valor		
1 Single_Outputs	Write Multiple Coils (Código funcional 15)	Cíclico, t#100ms				16#0020	32
2 Color_Inicio_Piezas	Read Input Registers (Código funcional 04)	Cíclico, t#100ms	16#0003	1	Conservar el último valor		
3 Color_FIN	Read Input Registers (Código funcional 04)	Cíclico, t#100ms	16#0004	1	Conservar el último valor		
4 Stackler_Posicion	Write Multiple Registers (Código funcional 16)	Cíclico, t#100ms				16#0004	1
5 Progreso_Azules_Cuadro	Write Multiple Registers (Código funcional 16)	Cíclico, t#100ms				16#0005	1
6 Progreso_Verdes_Cuadro	Write Multiple Registers (Código funcional 16)	Cíclico, t#100ms				16#0006	1

Ilustración 35. Entradas y salidas en el módulo Slave del programa almacenaje.

Para ver las conexiones virtuales de los programas Codesys en Factory IO tenemos la siguiente imagen en la que podemos observar la división de cada parte que controlan los programas. Como se observa se han dispuesta tanto las entradas como las salidas de forma ordenada y consecutiva, es por esto que hay que indicar los offsets adecuadamente para que las direcciones de cada programa coincidan.



## 1.6.2. Funcionamiento y uso

El interfaz usuario máquina está diseñada para que sea fácilmente comprensible a simple vista y no se requieran de altos conocimientos para su uso, en particular este sistema consta de dos modos, manual y automático, fácilmente seleccionables con el mando indicado en la HMI, a continuación se va a proceder a describir el funcionamiento y uso de cada uno, así como los sistemas de seguridad.

### 1.6.2.1. Modo automático

Por defecto cuando se inicia el sistema el selector de modo está situado en modo automático, en este modo, una vez realizado un pedido e iniciado con el interruptor de inicio, clasificará las piezas que entran al sistema separándolas por colores y forma, es decir clasificará las bases y tapas verdes por una parte y las bases y tapas azules por otra, situándolas en el lado especificado para ser ensambladas y posteriormente paletizadas y almacenadas en la estantería en su posición correspondiente, además descartará las piezas grises ya que se considera que este color representan piezas sin pintura, por lo que son descartadas como defectuosas.

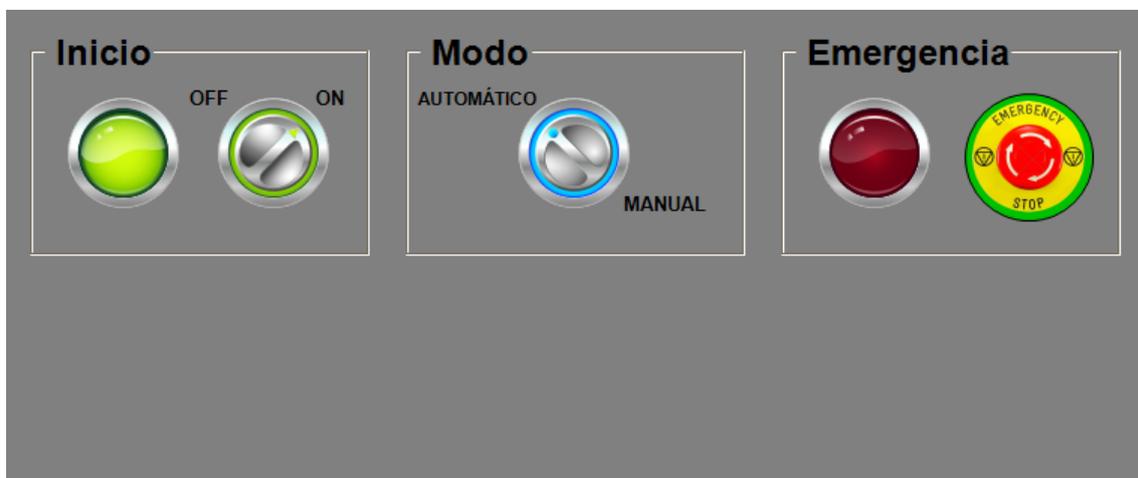


Ilustración 37. HMI activada en modo automático.

### 1.6.2.2. Modo manual

Si es seleccionado el modo manual se mostrarán dos botones adicionales, como se muestra en la siguiente imagen, que nos permitirán elegir que hacer con cada pieza que entre al

sistema, cuando entre una pieza al sistema se detendrá delante del empujador que desplaza las piezas a la rampa de descartes y según el botón que pulsemos la pieza será descartada o clasificada para ser ensamblada. Con este método podremos ensamblar cualquier pieza de un color determinado con cualquier otra, es decir podremos clasificar piezas grises y mezclar colores. Para ser ensambladas en este modo únicamente se usará un robot, el correspondiente a las piezas verdes en el modo automático para ser más exactos.



Ilustración 38. HMI activa en modo manual.

Además como se puede observar en la imagen vemos que los cuadros de inserción de pedido pasan a estar deshabilitados y se añade un tercer cuadro en la parte de progreso para contar las piezas totales que se ensamblan, que serán las mismas que ensamble el robot del lado de las piezas verdes.

### 1.6.3. Sistemas de emergencia

Como en todos los procesos industriales se pueden producir fallos en el manejo de las piezas ya sea por imperfecciones de estas o desgastes en las máquinas, en Factory IO, pese a ser un sistema simulado, también ocurren este tipo de fallos, lo que lo acerca aún más si cabe a la realidad. Para solventar estos fallos en el sistema, que no se detenga la producción y no interactúe nadie sobre el proceso en la planta, se han diseñado dos acciones accesibles cuando se pulsa la seta de emergencia o cuando el sistema detecte que se ha producido un error en el ensamblaje de una pieza, lo que detendrá el proceso y deberemos de realizar acciones para solventarlo previamente a reactivar el sistema y continuar el pedido. Como se observa en la

siguiente ilustración cuando está activada una emergencia se despliegan dos botones debajo de la seta de parada de emergencia



Ilustración 39. HMI con emergencia activa.

El método limpiar cintas está pensado para cuando tras un fallo en el transporte de las piezas se quede alguna atascada y se impida la circulación del resto, lo que ocasiona una acumulación de piezas si no hay nadie supervisando el proceso. Una vez se halla activado la emergencia y se pulse el botón “Limpiar cintas” se elevarán los posicionadores de piezas de las cuatro cintas de clasificación y se pondrán en marcha las cintas para expulsar del sistema todas las piezas que se hallan desviado de su recorrido.



Ilustración 40. Piezas atascadas.



Ilustración 41. Posicionadores elevados.

Con el botón “Ajustar piezas” lo que se pretende es que si un robot no ha cogido bien una pieza para ensamblarla y ha depositado solo una parte en el palé quiere decir que una pieza no se ha posicionado bien para ser ensamblada por lo que al pulsar este botón se activarán de nuevo todos los posicionadores para solventar este fallo.



*Ilustración 42. Pieza desajustada.*



*Ilustración 43. Ajustadores activados manualmente.*

## 1.7. Conclusión/Prólogo

Con este proyecto se ha deseado ejemplarizar como realizar el control de un proceso industrial de complejidad media, es por esto por lo que se ha creado un sistema virtual en el que se ha tenido que resolver como clasificar y ensamblar las piezas con los recursos justos que permitan una velocidad de procesado aceptable, al igual que en la parte de clasificación se ha tenido que resolver como hacer el uso de una sola estantería para dos productos distintos elaborados a la vez de forma aleatoria y todo controlado por un sistema gemelo que permite controlar los procesos separados con la dificultad de que no se pueden comunicar directamente.

También cabe destacar que el proceso es totalmente adaptable a otras situaciones como clasificar más colores incorporando más robots y cintas, añadir preproceso a las piezas, añadir más estanterías o subprocesos como paletizar varias piezas en el mismo palé o disponer pedidos de varios colores en cada palé. Y controlar cada subproceso con un PLC distinto.

Como conclusión podemos afirmar que se han conseguido los objetivos iniciales ya que los resultados obtenidos son satisfactorios pese a que se han tenido que resolver previamente problemas como la falta de conocimientos sobre las herramientas de software utilizadas, los tipos de comunicaciones entre dispositivos industriales y la resolución de problemas particulares como los fallos del sistema.

## 1.8. Bibliografía

3S-Smart Software Solutions GmbH. (s.f.). *Codesys*. Recuperado el 5 de Septiembre de 2021, de <https://www.codesys.com/>

EasyPLC Software. (s.f.). *Nirtec*. Recuperado el 9 de Junio de 2021, de <https://www.nirtec.com/index.php/easyplc-machines-simulator/>

Modbus Organization. (11 de 1 de 2021). *Modbus*. Obtenido de <https://modbus.org/>

Real games. (s.f.). *Factory I/O*. Recuperado el 14 de Enero de 2021, de <https://docs.factoryio.com/>

SMC Corporation. (s.f.). *SMC International training*. Recuperado el 8 de Agosto de 2021, de <https://www.smctraining.com/es>

Wikimedia. (s.f.). *Wikipedia*. Recuperado el 21 de Febrero de 2021, de [https://es.wikipedia.org/wiki/Sequential\\_Function\\_Chart](https://es.wikipedia.org/wiki/Sequential_Function_Chart)

## 1.9. Tabla de ilustraciones

Ilustración 1. Representación de todos los sistemas. ....	6
Ilustración 2. Ejemplo de proceso en LogixPro. ....	11
Ilustración 3. Representación de realidad virtual en EasyPLC Machines Simulator. ....	11
Ilustración 4. Ejemplo de proceso en EasyPLC Machines Simulator. ....	12
Ilustración 5. Ejemplo de proceso en Factory IO. ....	12
Ilustración 6. Ejemplo de escenas. ....	14
Ilustración 7. Escenario configurable. ....	15
Ilustración 8. Multitud de objetos disponibles. ....	15
Ilustración 9. Amplia conectividad ....	16
Ilustración 10. Ejemplo de fallo en el proceso. ....	16
Ilustración 11. Control analógico y digital. ....	17
Ilustración 12. Ejemplo de sistema simulado y supervisión de variables. ....	18
Ilustración 13. Tipos de conexión seleccionables en Codesys vía ethernet. ....	19
Ilustración 14. Lenguajes seleccionables en Codesys. ....	20
Ilustración 15. Visualización de direcciones configuradas. ....	21
Ilustración 16. Ejemplo de interfaz gráfica en modo edición. ....	22
Ilustración 17. Disposición de las diferentes piezas. ....	24
Ilustración 18. Lectura de piezas por forma y color. ....	25
Ilustración 19. Robots ensamblando las piezas. ....	26
Ilustración 20. Robots con tapas cogidas esperando bases. ....	26
Ilustración 21. Palés posicionándose para recibir las piezas ensambladas. ....	27
Ilustración 22. Recorrido de los palés según el color de pieza que transportan. ....	27
Ilustración 23. Estantería donde se muestra la ubicación de los palés según el color de pieza que contienen. ....	28
Ilustración 24. Delimitación controlada por cada programa. ....	29
Ilustración 25. Ventana de aviso al pulsar el botón "Nuevo pedido." ....	31
Ilustración 26. Ventana de aviso que surge al pulsar el botón "Borrar progreso". ....	31
Ilustración 27. HMI en modo edición. ....	32
Ilustración 28. Representación del conjunto del sistema. ....	34
Ilustración 29. Menú del sistema operativo y aplicaciones de Codesys. ....	35
Ilustración 30. Conexión del programa Codesys con el PLCs virtual. ....	35
Ilustración 31. Configuración del modulo Ethernet. ....	36
Ilustración 32. Módulo Master. ....	37

Ilustración 33. Módulo Slave.....	38
Ilustración 34. Entradas y salidas en el módulo Slave del programa clasificación.....	38
Ilustración 35. Entradas y salidas en el módulo Slave del programa almacenaje.....	39
Ilustración 36. Representación de entradas y salidas en Factory IO. ....	41
Ilustración 37. HMI activada en modo automático.....	41
Ilustración 38. HMI activa en modo manual. ....	42
Ilustración 39. HMI con emergencia activa.....	43
Ilustración 40. Piezas atascadas. ....	43
Ilustración 41. Posicionadores elevados. ....	43
Ilustración 42. Pieza desajustada. ....	44
Ilustración 43. Ajustadores activados manualmente.....	44

## Anexo 1: Manual del programador

Con la finalidad de que este documento ofrezca a usuarios expertos más información y les facilite la comprensión y la posibilidad de realizar modificaciones se procederá a documentar la parte de programación de los distintos ficheros creados en Codesys.

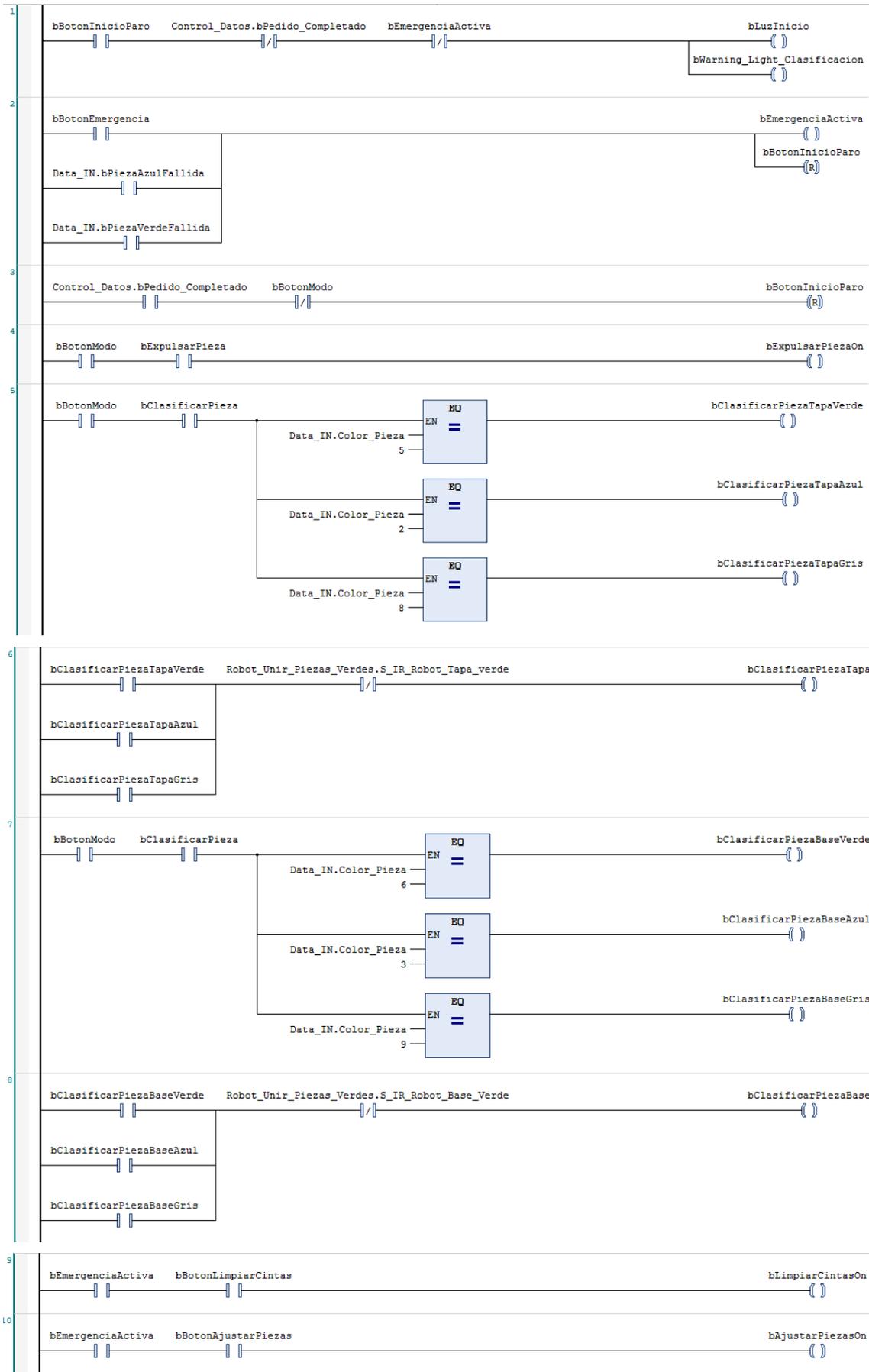
### Programa “clasificación”

#### - Botones HMI

Variables:

S_IR_FalloPiezaAzul	AT %IX2.7	:BOOL;
S_IR_FalloPiezaVerde	AT %IX3.0	:BOOL;
bWarning_Light_Clasificacion	AT %QX3.7	:BOOL;
bBotonInicioParo		:BOOL;
bLuzInicio		:BOOL;
bBotonEmergencia		:BOOL;
bEmergenciaActiva		:BOOL;
bBotonModo		:BOOL;
bResetProgreso		:BOOL;
bResetPedido		:BOOL;
bExpulsarPieza		:BOOL;
bExpulsarPiezaOn		:BOOL;
bClasificarPieza		:BOOL;
bClasificarPiezaTapaVerde		:BOOL;
bClasificarPiezaTapaAzul		:BOOL;
bClasificarPiezaTapaGris		:BOOL;
bClasificarPiezaTapa		:BOOL;
bClasificarPiezaBaseVerde		:BOOL;
bClasificarPiezaBaseAzul		:BOOL;
bClasificarPiezaBaseGris		:BOOL;
bClasificarPiezaBase		:BOOL;
bBotonAjustarPiezas		:BOOL;
bAjustarPiezasOn		:BOOL;
bBotonLimpiarCintas		:BOOL;
bLimpiarCintasOn		:BOOL;

Código:



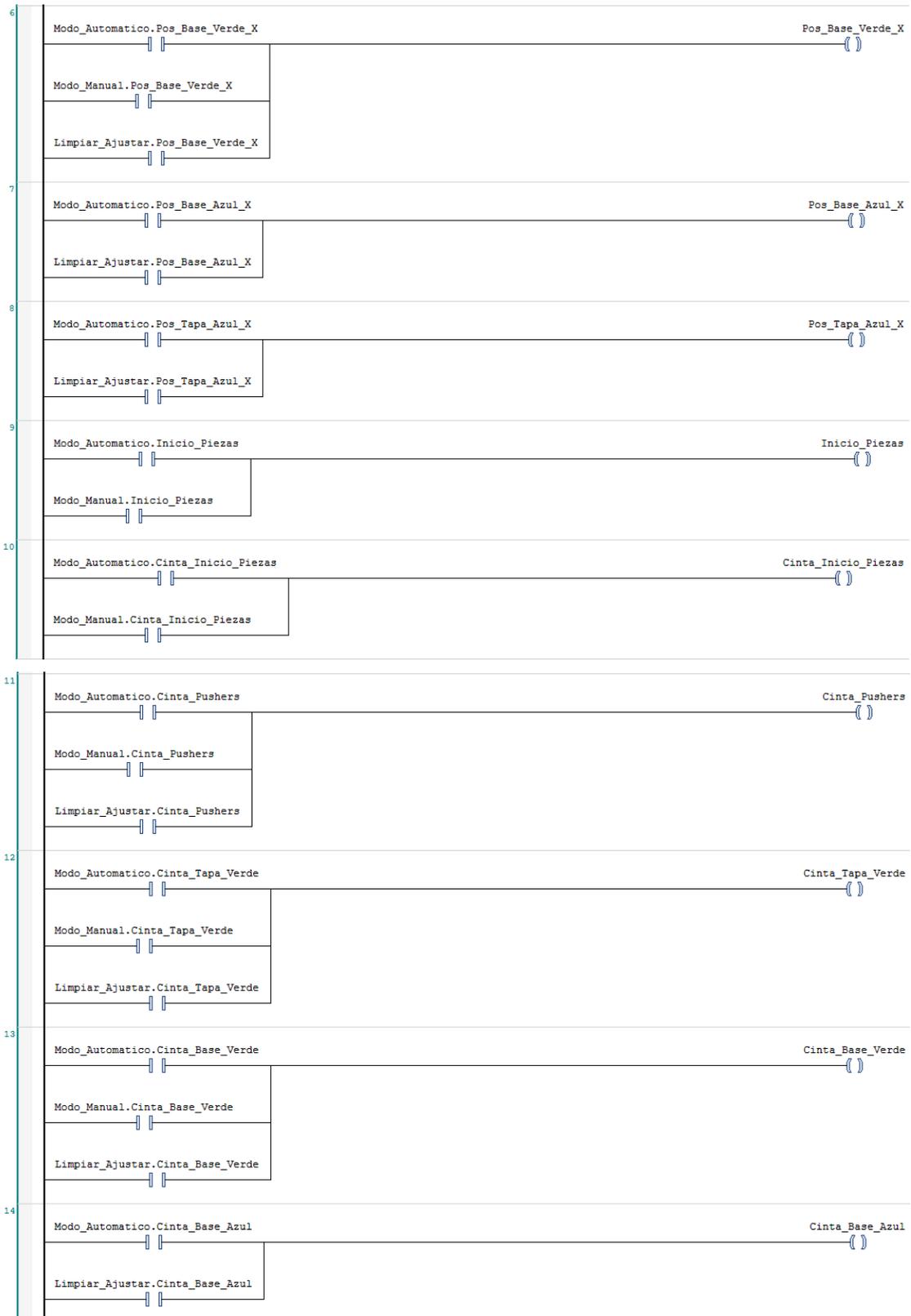
## - Control Clasificación

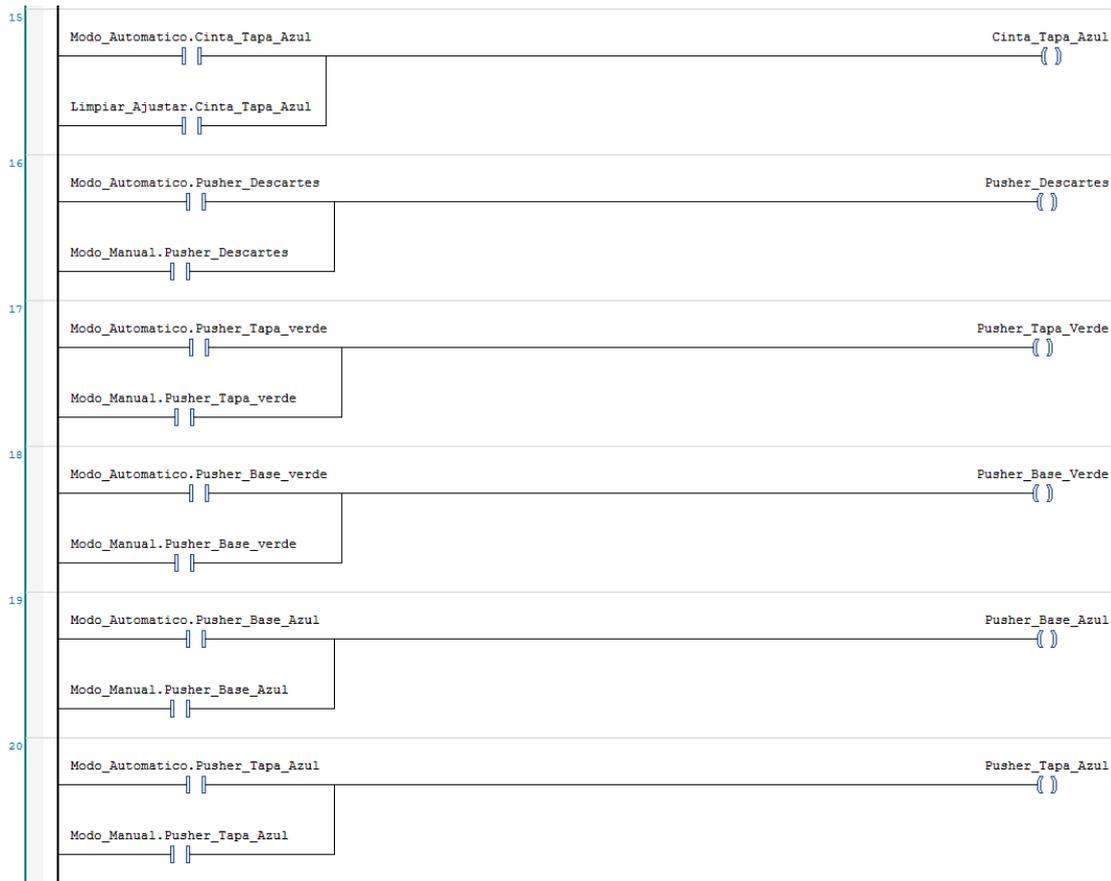
Variables:

Pos_Tapa_Verde_Y	AT %QX0.0: BOOL;
Pos_Base_Verde_Y	AT %QX0.1: BOOL;
Pos_Base_Azul_Y	AT %QX0.2: BOOL;
Pos_Tapa_Azul_Y	AT %QX0.3: BOOL;
Pos_Tapa_Verde_X	AT %QX0.4: BOOL;
Pos_Base_Verde_X	AT %QX0.5: BOOL;
Pos_Base_Azul_X	AT %QX0.6: BOOL;
Pos_Tapa_Azul_X	AT %QX0.7: BOOL;
Inicio_Piezas	AT %QX1.0: BOOL;
Cinta_Inicio_Piezas	AT %QX1.1: BOOL;
Cinta_Pushers	AT %QX1.2: BOOL;
Cinta_Tapa_Verde	AT %QX1.3: BOOL;
Cinta_Base_Verde	AT %QX1.4: BOOL;
Cinta_Base_Azul	AT %QX1.5: BOOL;
Cinta_Tapa_Azul	AT %QX1.6: BOOL;
Pusher_Descartes	AT %QX2.0: BOOL;
Pusher_Tapa_Verde	AT %QX2.1: BOOL;
Pusher_Base_Verde	AT %QX2.2: BOOL;
Pusher_Base_Azul	AT %QX2.3: BOOL;
Pusher_Tapa_Azul	AT %QX2.4: BOOL;

Código:







- **Control Datos**

Variables:

```

S_IR_Pale_Azul_Inicio    AT %IX3.4 :BOOL;
S_IR_Pale_Verde_Inicio  AT %IX3.7 :BOOL;
S_Turntable_DE          AT %IX4.1 :BOOL;
S_Turntable_IZ          AT %IX4.2 :BOOL;
S_Turntable_EXIT        AT %IX4.3 :BOOL;
S_Turntable_IN          AT %IX4.4 :BOOL;
S_IR_Pale_En_Posicion   AT %IX4.5 :BOOL;
S_Stacker_Centrado      AT %IX4.7 :BOOL;
S_Stacker_Moviendose_X  AT %IX5.0 :BOOL;
S_Stacker_Moviendose_Z  AT %IX5.1 :BOOL;
S_IR_Pale_Azul_Fin      AT %IX5.3 :BOOL;
S_IR_Pale_Verde_Fin     AT %IX5.4 :BOOL;
S_Color_FIN             AT %IW4   :WORD;
bActivo_Lado_Azul       :BOOL;
bActivo_Lado_Verde      :BOOL;
Color_En_Pale           :WORD;
bPedido_Azul_Completado :BOOL;
bPedido_Verde_Completado :BOOL;
bPedido_Completado      :BOOL;
Pedido_Azules           :UINT:=0;
Pedido_Verdes           :UINT:=0;
Pedido_Azules_terminadas :UINT:=0;
Pedido_Verdes_Terminadas :UINT:=0;
Azules_Terminadas       :UINT:=0;
Verdes_Terminadas       :UINT:=0;

```

Código:

```

IF Data_IN.bSumarPiezaAzul THEN
    Pedido_Azules_Terminadas := Pedido_Azules_Terminadas + 1;
    Azules_Terminadas := Azules_Terminadas + 1;
END_IF

IF Data_IN.bSumarPiezaVerde THEN
    Pedido_Verdes_Terminadas := Pedido_verdes_Terminadas + 1;
    Verdes_Terminadas := Verdes_Terminadas + 1;
END_IF

IF Pedido_Azules = Pedido_Azules_Terminadas AND Botones_HMI.bBotonInicioParo THEN
    bPedido_Azul_Completado := TRUE;
    Pedido_Azules_Terminadas := 0;
END_IF

IF Pedido_Verdes = Pedido_Verdes_Terminadas AND Botones_HMI.bBotonInicioParo THEN
    bPedido_Verde_Completado := TRUE;
    Pedido_Verdes_Terminadas := 0;
END_IF

IF bPedido_Azul_Completado AND bPedido_Verde_Completado THEN
    bPedido_Completado := TRUE;
END_IF

IF Botones_HMI.bBotonModo THEN
    Control_Datos.bPedido_Azul_Completado := FALSE;
    Control_Datos.bPedido_Verde_Completado := FALSE;
    Control_Datos.bPedido_Completado := FALSE;
END_IF

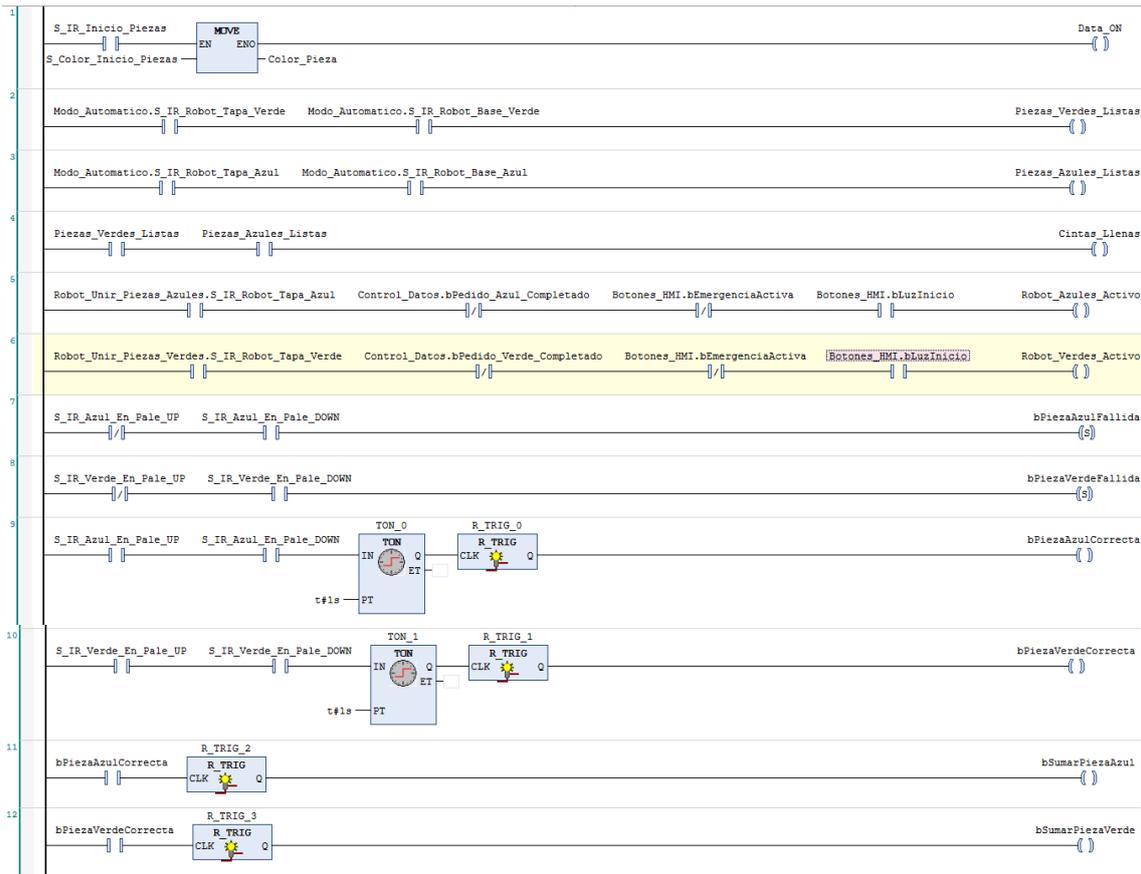
```

## - Data IN

Variables:

S_Color_Inicio_Piezas	AT %IW3 :WORD;
S_IR_Inicio_Piezas	AT %IX1.0 :BOOL;
S_IR_FalloPiezaAzul	AT %IX2.7 :BOOL;
S_IR_FalloPiezaVerde	AT %IX3.0 :BOOL;
S_IR_Pale_Azul_Inicio	AT %IX3.4 :BOOL;
S_IR_Pale_Verde_Inicio	AT %IX3.7 :BOOL;
S_IR_Azul_En_Pale_UP	AT %IX3.2 : BOOL;
S_IR_Azul_En_Pale_DOWN	AT %IX3.3 : BOOL;
S_IR_Verde_En_Pale_UP	AT %IX3.5 : BOOL;
S_IR_Verde_En_Pale_DOWN	AT %IX3.6 : BOOL;
S_IR_Pale_Azul_Fin	AT %IX5.3 : BOOL;
S_IR_Pale_verde_Fin	AT %IX5.4 : BOOL;
bPiezaAzulFallida	: BOOL;
bPiezaVerdeFallida	: BOOL;
bPiezaAzulCorrecta	: BOOL;
bPiezaVerdeCorrecta	: BOOL;
bSumarPiezaAzul	: BOOL;
bSumarPiezaVerde	: BOOL;
Piezas_Verdes_Listas	: BOOL;
Piezas_Azules_Listas	: BOOL;
Cintas_Llenas	: BOOL;
Color_Pieza	: WORD;
Data_ON	: BOOL;
Robot_Azules_Activo	: BOOL;
Robot_Verdes_Activo	: BOOL;
R_TRIG_0	: R_TRIG;
R_TRIG_1	: R_TRIG;
R_TRIG_2	: R_TRIG;
R_TRIG_3	: R_TRIG;
R_TRIG_4	: R_TRIG;
R_TRIG_5	: R_TRIG;
TON_0	: TON;
TON_1	: TON;

Código:

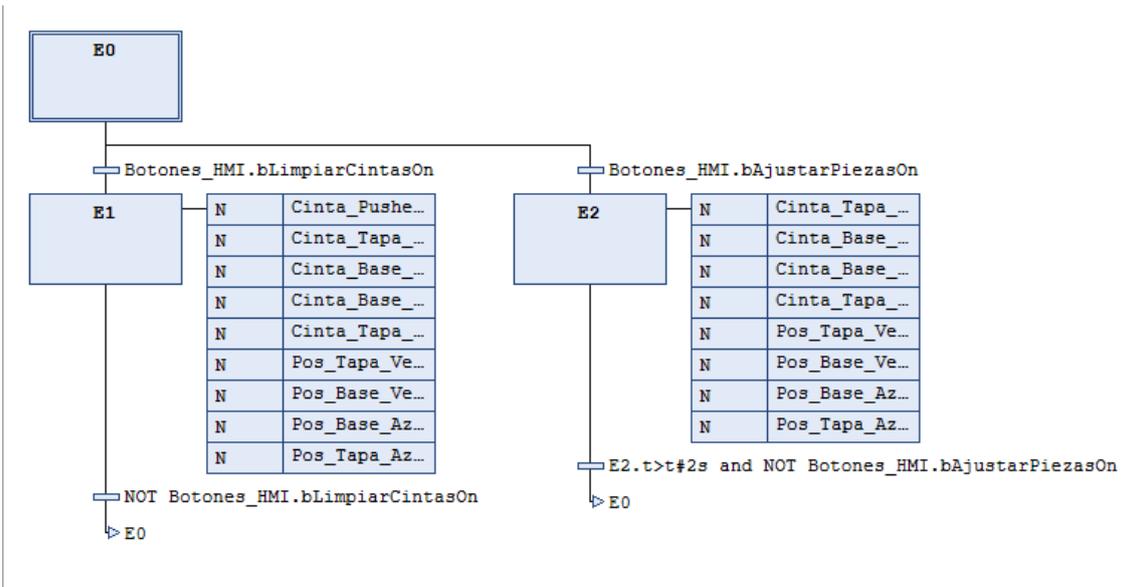


- **Limpiar Ajustar**

Variables:

Pos_Tapa_Verde_Y	: BOOL;
Pos_Base_Verde_Y	: BOOL;
Pos_Base_Azul_Y	: BOOL;
Pos_Tapa_Azul_Y	: BOOL;
Pos_Tapa_Verde_X	: BOOL;
Pos_Base_Verde_X	: BOOL;
Pos_Base_Azul_X	: BOOL;
Pos_Tapa_Azul_X	: BOOL;
Inicio_Piezas	: BOOL;
Cinta_Inicio_Piezas	: BOOL;
Cinta_Pushers	: BOOL;
Cinta_Tapa_Verde	: BOOL;
Cinta_Base_Verde	: BOOL;
Cinta_Base_Azul	: BOOL;
Cinta_Tapa_Azul	: BOOL;

Código:

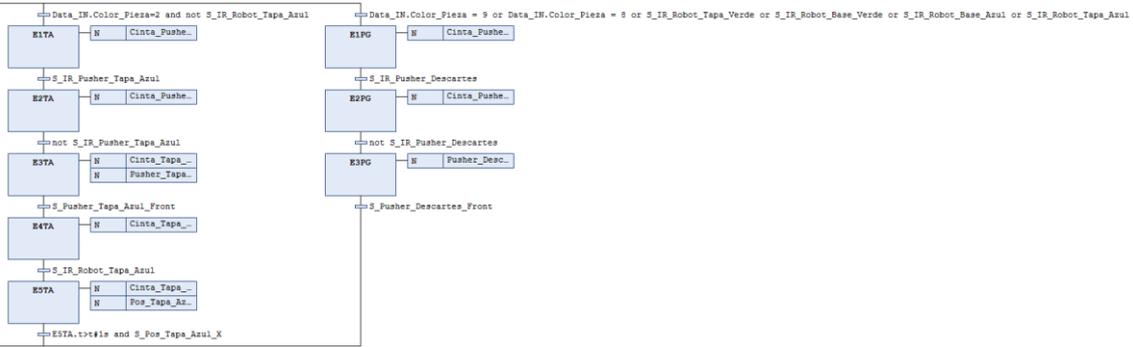
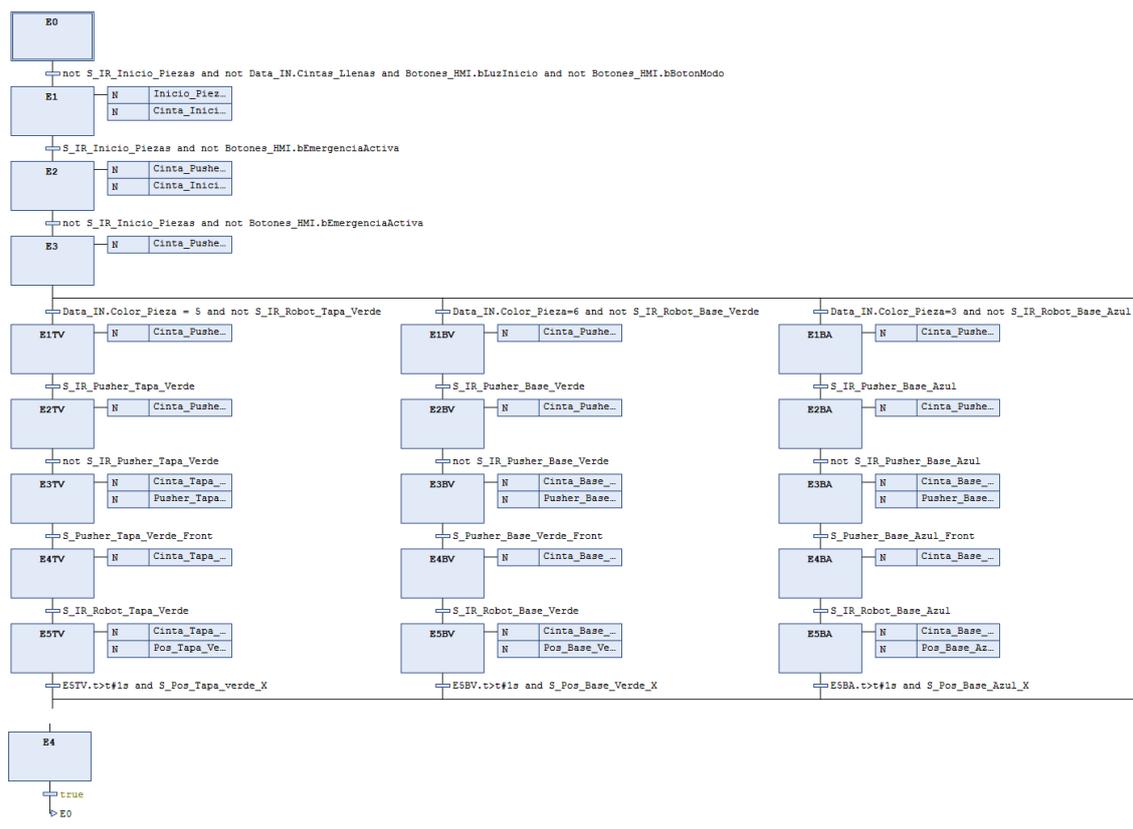


- **Modo Automático**

Variables:

S_IR_Robot_Tapa_verde	AT %IX0.4: BOOL;
S_IR_Robot_Base_Verde	AT %IX0.5: BOOL;
S_IR_Robot_Base_Azul	AT %IX0.6: BOOL;
S_IR_Robot_Tapa_Azul	AT %IX0.7: BOOL;
S_IR_Inicio_Piezas	AT %IX1.0: BOOL;
S_IR_Pusher_Descartes	AT %IX1.1: BOOL;
S_IR_Pusher_Tapa_Verde	AT %IX1.2: BOOL;
S_IR_Pusher_Base_Verde	AT %IX1.3: BOOL;
S_IR_Pusher_Base_Azul	AT %IX1.4: BOOL;
S_IR_Pusher_Tapa_Azul	AT %IX1.5: BOOL;
S_Pusher_Descartes_Front	AT %IX1.6: BOOL;
S_Pusher_Tapa_Verde_Front	AT %IX1.7: BOOL;
S_Pusher_Base_Verde_Front	AT %IX2.0: BOOL;
S_Pusher_Base_Azul_Front	AT %IX2.1: BOOL;
S_Pusher_Tapa_Azul_Front	AT %IX2.2: BOOL;
S_Pos_Base_Azul_X	AT %IX2.3: BOOL;
S_Pos_Tapa_Azul_X	AT %IX2.4: BOOL;
S_Pos_Base_Verde_X	AT %IX2.5: BOOL;
S_Pos_Tapa_verde_X	AT %IX2.6: BOOL;
Pos_Tapa_Verde_Y	: BOOL;
Pos_Base_Verde_Y	: BOOL;
Pos_Base_Azul_Y	: BOOL;
Pos_Tapa_Azul_Y	: BOOL;
Pos_Tapa_Verde_X	: BOOL;
Pos_Base_Verde_X	: BOOL;
Pos_Base_Azul_X	: BOOL;
Pos_Tapa_Azul_X	: BOOL;
Inicio_Piezas	: BOOL;
Cinta_Inicio_Piezas	: BOOL;
Cinta_Pushers	: BOOL;
Cinta_Tapa_Verde	: BOOL;
Cinta_Base_Verde	: BOOL;
Cinta_Base_Azul	: BOOL;
Cinta_Tapa_Azul	: BOOL;
Pusher_Descartes	: BOOL;
Pusher_Tapa_Verde	: BOOL;
Pusher_Base_Verde	: BOOL;
Pusher_Base_Azul	: BOOL;
Pusher_Tapa_Azul	: BOOL;
Tapa_Verde_Lista	: BOOL;
Base_Verde_Lista	: BOOL;

Código:



- **Modo Manual**

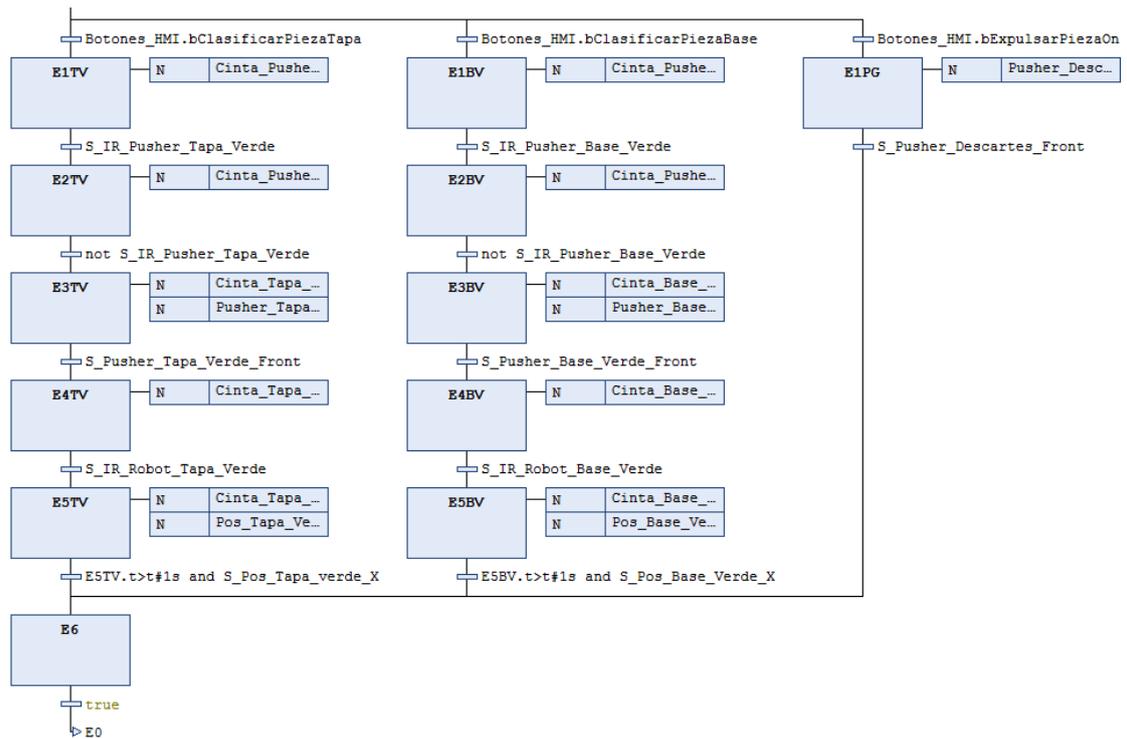
Variables:

- |                           |                  |
|---------------------------|------------------|
| S_IR_Robot_Tapa_verde     | AT %IX0.4: BOOL; |
| S_IR_Robot_Base_Verde     | AT %IX0.5: BOOL; |
| S_IR_Robot_Base_Azul      | AT %IX0.6: BOOL; |
| S_IR_Robot_Tapa_Azul      | AT %IX0.7: BOOL; |
| S_IR_Inicio_Piezas        | AT %IX1.0: BOOL; |
| S_IR_Pusher_Descartes     | AT %IX1.1: BOOL; |
| S_IR_Pusher_Tapa_Verde    | AT %IX1.2: BOOL; |
| S_IR_Pusher_Base_Verde    | AT %IX1.3: BOOL; |
| S_IR_Pusher_Base_Azul     | AT %IX1.4: BOOL; |
| S_IR_Pusher_Tapa_Azul     | AT %IX1.5: BOOL; |
| S_Pusher_Descartes_Front  | AT %IX1.6: BOOL; |
| S_Pusher_Tapa_Verde_Front | AT %IX1.7: BOOL; |
| S_Pusher_Base_Verde_Front | AT %IX2.0: BOOL; |
| S_Pusher_Base_Azul_Front  | AT %IX2.1: BOOL; |
| S_Pusher_Tapa_Azul_Front  | AT %IX2.2: BOOL; |

S_Pos_Base_Azul_X	AT %IX2.3: BOOL;
S_Pos_Tapa_Azul_X	AT %IX2.4: BOOL;
S_Pos_Base_Verde_X	AT %IX2.5: BOOL;
S_Pos_Tapa_verde_X	AT %IX2.6: BOOL;
Pos_Tapa_Verde_Y	: BOOL;
Pos_Base_Verde_Y	: BOOL;
Pos_Base_Azul_Y	: BOOL;
Pos_Tapa_Azul_Y	: BOOL;
Pos_Tapa_Verde_X	: BOOL;
Pos_Base_Verde_X	: BOOL;
Pos_Base_Azul_X	: BOOL;
Pos_Tapa_Azul_X	: BOOL;
Inicio_Piezas	: BOOL;
Cinta_Inicio_Piezas	: BOOL;
Cinta_Pushers	: BOOL;
Cinta_Tapa_Verde	: BOOL;
Cinta_Base_Verde	: BOOL;
Cinta_Base_Azul	: BOOL;
Cinta_Tapa_Azul	: BOOL;
Pusher_Descartes	: BOOL;
Pusher_Tapa_verde	: BOOL;
Pusher_Base_verde	: BOOL;
Pusher_Base_Azul	: BOOL;
Pusher_Tapa_Azul	: BOOL;
Tapa_Verde_Lista	: BOOL;
Base_Verde_Lista	: BOOL;

Código:



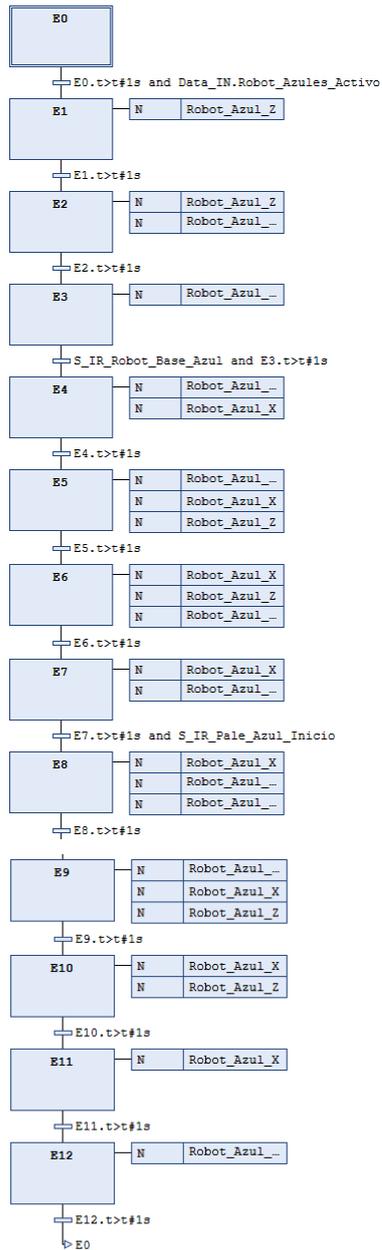


- **Robot Unir Piezas Azules**

Variables:

S_IR_Robot_Base_Azul	AT %IX0.6:BOOL;
S_IR_Robot_Tapa_Azul	AT %IX0.7:BOOL;
S_IR_Pale_Azul_Inicio	AT %IX3.4:BOOL;
Robot_Azul_X	AT %QX3.2:BOOL;
Robot_Azul_Z	AT %QX3.3:BOOL;
Robot_Azul_Pinza	AT %QX3.4:BOOL;
Robot_Azul_Horario	AT %QX3.5:BOOL;
Robot_Azul_Antihorario	AT %QX3.6:BOOL;

Código:

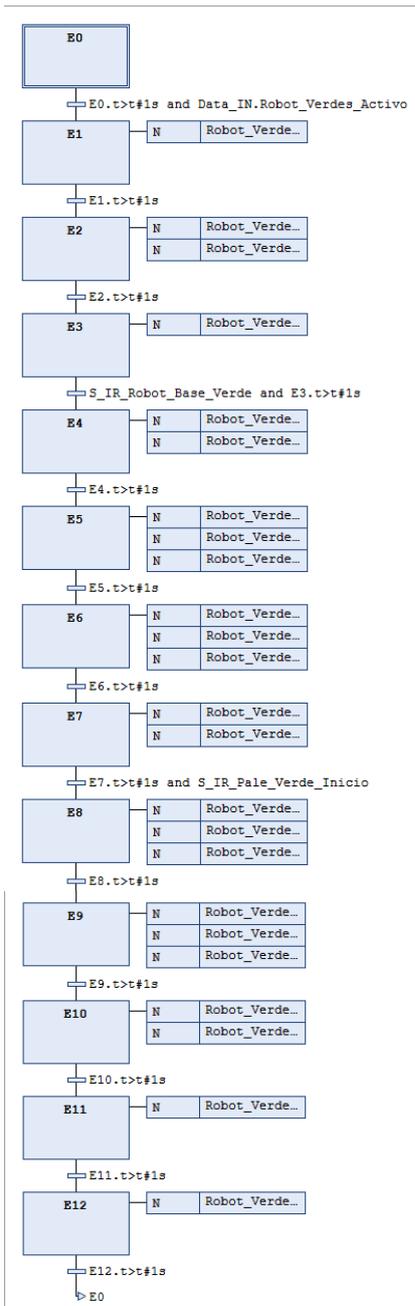


**- Robot Unir Piezas Verdes**

Variables:

S_IR_Robot_Tapa_verde	AT %IX0.4:BOOL;
S_IR_Robot_Base_Verde	AT %IX0.5:BOOL;
S_IR_Pale_Verde_Inicio	AT %IX3.7:BOOL;
Robot_Verde_X	AT %QX2.5:BOOL;
Robot_Verde_Z	AT %QX2.6:BOOL;
Robot_Verde_Pinza	AT %QX2.7:BOOL;
Robot_Verde_Horario	AT %QX3.0:BOOL;
Robot_Verde_Antihorario	AT %QX3.1:BOOL;

Código:



## Programa “almacenaje”

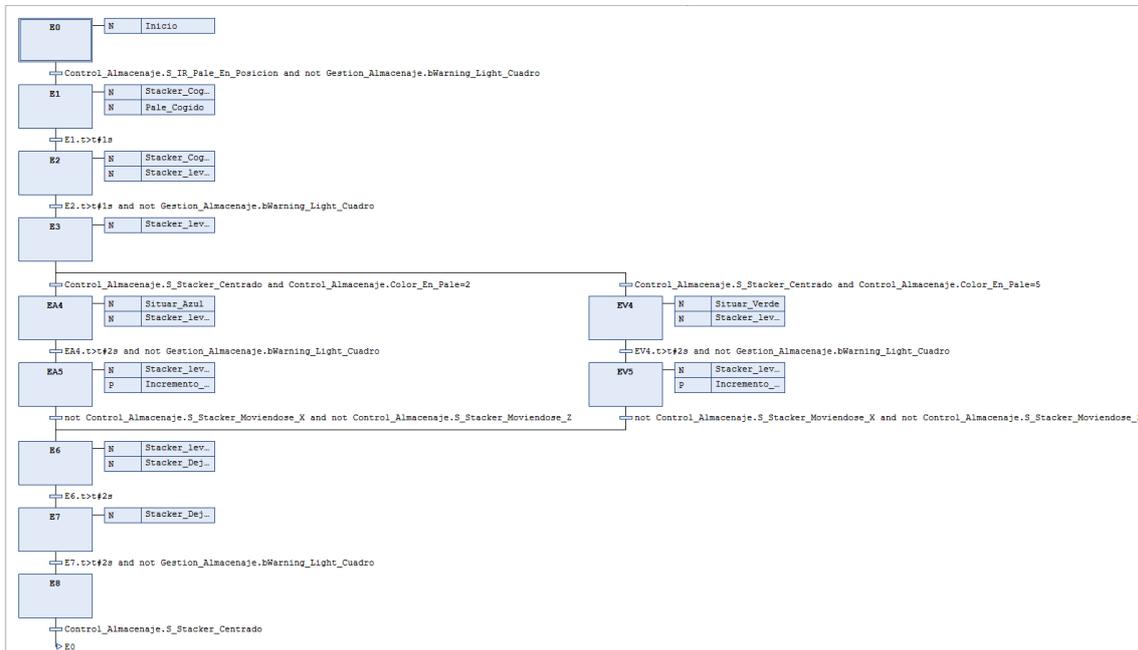
### - Almacenamiento

Variables:

Stacker_Coger	AT %QX2.7: BOOL;
Stacker_Dejar	AT %QX3.0: BOOL;
Stacker_levantar	AT %QX3.1: BOOL;
Pale_Cogido	: BOOL:=FALSE;
Situar_Azul	: BOOL;
Situar_Verde	: BOOL;
Inicio	: BOOL;
Incremento_Posicion_Azul	: BOOL:=FALSE;

Incremento\_Posicion\_verde : BOOL:=FALSE;

Código:



- **Control Almacenaje**

Variables:

S_IR_Azul_En_Pale_UP	AT %IX3.2 :BOOL;
S_IR_Azul_En_Pale_DOWN	AT %IX3.3 :BOOL;
S_IR_Pale_Azul_Inicio	AT %IX3.4 :BOOL;
S_IR_Verde_En_Pale_UP	AT %IX3.5 :BOOL;
S_IR_Verde_En_Pale_DOWN	AT %IX3.6 :BOOL;
S_IR_Pale_Verde_Inicio	AT %IX3.7 :BOOL;
S_Turtable_DE	AT %IX4.1 :BOOL;
S_Turtable_IZ	AT %IX4.2 :BOOL;
S_Turtable_EXIT	AT %IX4.3 :BOOL;
S_Turtable_IN	AT %IX4.4 :BOOL;
S_IR_Pale_En_Posicion	AT %IX4.5 :BOOL;
S_Stacker_Centrado	AT %IX4.7 :BOOL;
S_Stacker_Moviendose_X	AT %IX5.0 :BOOL;
S_Stacker_Moviendose_Z	AT %IX5.1 :BOOL;
S_IR_Pale_Azul_Fin	AT %IX5.3 :BOOL;
S_IR_Pale_Verde_Fin	AT %IX5.4 :BOOL;
S_Color_FIN	AT %IW4 :WORD;
Stacker_Posicion	AT %QW2 :WORD:=55;
Progreso_Azules_Cuadro	AT %QW3 :WORD:=0;
Progreso_Verdes_Cuadro	AT %QW4 :WORD:=0;
Activo_Lado_Azul	:BOOL;
Activo_Lado_Verde	:BOOL;
Color_En_Pale	:WORD;
Posicion_Verde	:UINT:=28;
Posicion_Azul	:UINT:=1;
Posicion_Inicial	:UINT:=55;
Almacen_Azul_Lleno	:BOOL:=FALSE;
Almacen_Verde_Lleno	:BOOL:=FALSE;

Código:

IF Almacenamiento.Pale\_Cogido THEN

```

        Color_En_Pale := S_Color_FIN;
    END_IF

    IF Almacenamiento.inicio THEN
        Stacker_Posicion := Posicion_Inicial;
    END_IF

    IF Almacenamiento.Situar_Azul THEN
        Stacker_Posicion := Posicion_Azul;
    END_IF

    IF Almacenamiento.Situar_Verde THEN
        Stacker_Posicion := Posicion_Verde;
    END_IF

    IF Gestion_Almacenaje.bSuma_Azul THEN
        Progreso_Azules_Cuadro := Progreso_Azules_Cuadro + 1;
    END_IF

    IF Gestion_Almacenaje.bSuma_Verde THEN
        Progreso_Verdes_Cuadro := Progreso_Verdes_Cuadro + 1;
    END_IF

    IF Almacenamiento.Incremento_Posicion_Azul THEN
        Posicion_Azul := Posicion_Azul + 1;
    END_IF

    IF Almacenamiento.Incremento_Posicion_Verde THEN
        Posicion_Verde := Posicion_Verde + 1;
    END_IF

    IF Posicion_Azul=28 THEN
        Almacen_Azul_Lleno:=TRUE;
    END_IF

    IF Posicion_Verde=55 THEN
        Almacen_Verde_Lleno:=TRUE;
    END_IF

```

## - **Gestión Almacenaje**

Variables:

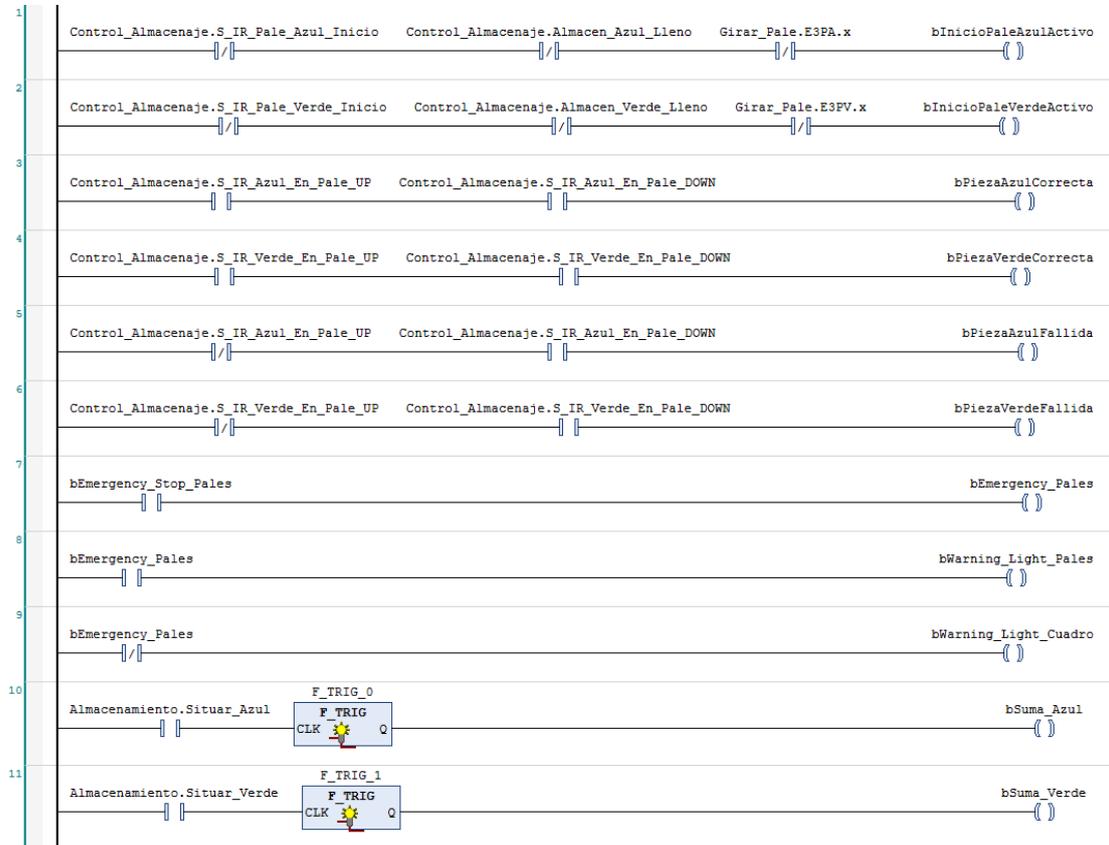
bEmergency_Stop_Pales	AT %IX5.5 :BOOL;
bWarning_Light_Pales	AT %QX3.4 :BOOL;
bWarning_Light_Cuadro	AT %QX3.5 :BOOL;
bEmergency_Pales	:BOOL;
bInicioPaleAzulActivo	:BOOL;
bInicioPaleVerdeActivo	:BOOL;
bPiezaAzulCorrecta	:BOOL;
bPiezaVerdeCorrecta	:BOOL;
bPiezaAzulFallida	:BOOL;
bPiezaVerdeFallida	:BOOL;

```

bSuma_Azul           :BOOL;
bSuma_Verde         :BOOL;
F_TRIG_0             : F_TRIG;
F_TRIG_1             : F_TRIG;

```

Código:



### - Girar Palé

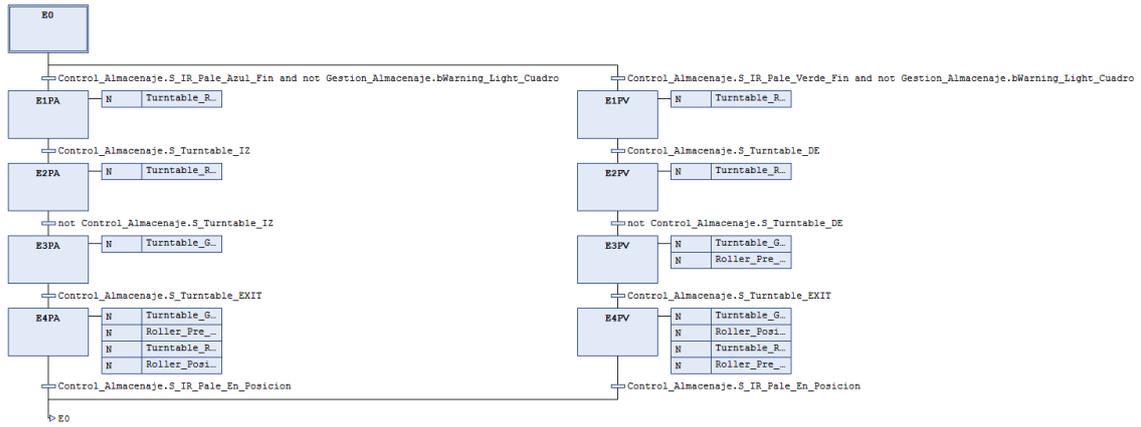
Variables:

```

Turntable_Giro      AT %QX2.1: BOOL;
Turntable_Roll_IZ  AT %QX2.2: BOOL;
Turntable_Roll_DE  AT %QX2.3: BOOL;
Roller_Pre_Posicion AT %QX2.4: BOOL;
Roller_Posicion_FIN AT %QX2.5: BOOL;

```

Código:



- **Inicio Palé Azul**

Variables:

Emisor_Pales_Azul	AT %QX0.0: BOOL;
Roller_Inicio_Front_Azul	AT %QX0.1: BOOL;
Roller_Inicio_Back_Azul	AT %QX0.2: BOOL;
Chain_Transfer_Front_Azul	AT %QX0.3: BOOL;
Chain_Transfer_Back_Azul	AT %QX0.4: BOOL;
Roller_Transfer_Azul	AT %QX0.5: BOOL;
Roller_Pale_Azul	AT %QX0.6: BOOL;
Roller_Curva_Azul	AT %QX0.7: BOOL;

Código:



- **Inicio Palé Verde**

Variables:

Emisor_Pales_Verde	AT %QX1.0: BOOL;
Roller_Inicio_Front_Verde	AT %QX1.1: BOOL;
Roller_Inicio_Back_Verde	AT %QX1.2: BOOL;
Chain_Transfer_Front_Verde	AT %QX1.3: BOOL;
Chain_Transfer_Back_Verde	AT %QX1.4: BOOL;
Roller_Transfer_Verde	AT %QX1.5: BOOL;
Roller_Pale_Verde	AT %QX1.6: BOOL;
Roller_Curva_Verde	AT %QX1.7: BOOL;

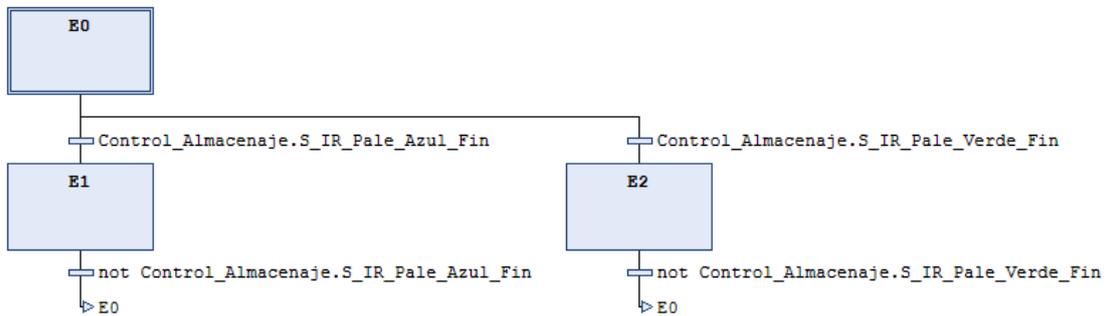
Código:



- **Palé Girado**

Variables:

Código:



## Anexo 2: Listas: actuadores, sensores, variables, entradas y salidas.

Con el objetivo de tener un acceso rápido a la hora de consultar cualquier conexión se han creado las siguientes tablas donde podremos consultar fácilmente a que dirección apunta cualquier elemento y su correspondencia entre el PLC y Factory IO, así como a que programa pertenece. En la primera tabla tenemos las salidas de los sensores y en la segunda las entradas para manejar los actuadores.

Nombre Inputs	Pertenece al programa	Dirección en CODESYS	Dirección en FACTORY
	-	%IX0.0	Input 0
	-	%IX0.1	Input 1
	-	%IX0.2	Input 2
	-	%IX0.3	Input 3
S_IR_Robot_Tapa_verde	Clasificación	%IX0.4	Input 4
S_IR_Robot_Base_verde	Clasificación	%IX0.5	Input 5
S_IR_Robot_Base_Azul	Clasificación	%IX0.6	Input 6
S_IR_Robot_Tapa_Azul	Clasificación	%IX0.7	Input 7
S_IR_Inicio_Piezas	Clasificación	%IX1.0	Input 8
S_IR_Pusher_Descartes	Clasificación	%IX1.1	Input 9
S_IR_Pusher_Tapa_Verde	Clasificación	%IX1.2	Input 10
S_IR_Pusher_Base_Verde	Clasificación	%IX1.3	Input 11
S_IR_Pusher_Base_Azul	Clasificación	%IX1.4	Input 12
S_IR_Pusher_Tapa_Azul	Clasificación	%IX1.5	Input 13
S_Pusher_Descartes_Front	Clasificación	%IX1.6	Input 14
S_Pusher_Tapa_Verde_Front	Clasificación	%IX1.7	Input 15
S_Pusher_Base_Verde_Front	Clasificación	%IX2.0	Input 16
S_Pusher_Base_Azul_Front	Clasificación	%IX2.1	Input 17
S_Pusher_Tapa_Azul_Front	Clasificación	%IX2.2	Input 18
S_Pos_Base_Azul_X	Clasificación	%IX2.3	Input 19
S_Pos_Tapa_Azul_X	Clasificación	%IX2.4	Input 20
S_Pos_Base_Verde_X	Clasificación	%IX2.5	Input 21
S_Pos_Tapa_Verde_X	Clasificación	%IX2.6	Input 22
S_IR_FalloPiezaAzul	Clasificación	%IX2.7	Input 23
S_IR_FalloPiezaVerde	Clasificación	%IX3.0	Input 24
	-	%IX3.1	Input 25
S_IR_Azul_En_Pale_UP	Almacenaje	%IX3.2	Input 26
S_IR_Azul_En_Pale_DOWN	Almacenaje	%IX3.3	Input 27
S_IR_Pale_Azul_Inicio	Almacenaje	%IX3.4	Input 28
S_IR_Verde_En_Pale_UP	Almacenaje	%IX3.5	Input 29
S_IR_Verde_En_Pale_DOWN	Almacenaje	%IX3.6	Input 30

S_IR_Pale_Verde_Inicio	Almacenaje	%IX3.7	Input 31
	-	%IX4.0	Input 32
S_Turntable_DE	Almacenaje	%IX4.1	Input 33
S_Turntable_IZ	Almacenaje	%IX4.2	Input 34
S_Turntable_EXIT	Almacenaje	%IX4.3	Input 35
S_Turntable_IN	Almacenaje	%IX4.4	Input 36
S_IR_Pale_En_Posicion	Almacenaje	%IX4.5	Input 37
	-	%IX4.6	Input 38
S_Stacker_Centrado	Almacenaje	%IX4.7	Input 39
S_Stacker_Moviendose_X	Almacenaje	%IX5.0	Input 40
S_Stacker_Moviendose_Z	Almacenaje	%IX5.1	Input 41
	-	%IX5.2	Input 42
S_IR_Pale_Azul_Fin	Almacenaje	%IX5.3	Input 43
S_IR_Pale_Verde_Fin	Almacenaje	%IX5.4	Input 44
Emergency_Stop_pales	Almacenaje	%IX5.5	Input 45
	-	%IX5.6	Input 46
	-	%IX5.7	Input 47
S_Color_Inicio_Piezas	Clasificación	%IW3	I.R. 3
S_Color_FIN	Almacenaje	%IW4	I.R. 4

Nombre Outputs	Pertenece al programa	Dirección en CODESYS	Dirección en FACTORY
Pos_Tapa_verde_Y	Clasificación	%QX0.0	Coil 0
Pos_Base_verde_Y	Clasificación	%QX0.1	Coil 1
Pos_Base_Azul_Y	Clasificación	%QX0.2	Coil 2
Pos_Tapa_Azul_Y	Clasificación	%QX0.3	Coil 3
Pos_Tapa_verde_X	Clasificación	%QX0.4	Coil 4
Pos_Base_verde_X	Clasificación	%QX0.5	Coil 5
Pos_Base_Azul_X	Clasificación	%QX0.6	Coil 6
Pos_Tapa_Azul_X	Clasificación	%QX0.7	Coil 7
Inicio_Piezas	Clasificación	%QX1.0	Coil 8
Cinta_Inicio_Piezas	Clasificación	%QX1.1	Coil 9
Cinta_Pushers	Clasificación	%QX1.2	Coil 10
Cinta_Tapa_Verde	Clasificación	%QX1.3	Coil 11
Cinta_Base_Verde	Clasificación	%QX1.4	Coil 12
Cinta_Base_Azul	Clasificación	%QX1.5	Coil 13
Cinta_Tapa_Azul	Clasificación	%QX1.6	Coil 14
	-	%QX1.7	Coil 15
Pusher_Descartes	Clasificación	%QX2.0	Coil 16
Pusher_Tapa_Verde	Clasificación	%QX2.1	Coil 17

Pusher_Base_verde	Clasificación	%QX2.2	Coil 18
Pusher_Base_Azul	Clasificación	%QX2.3	Coil 19
Pusher_Tapa_Azul	Clasificación	%QX2.4	Coil 20
Robot_verde_X	Clasificación	%QX2.5	Coil 21
Robot_Verde_Z	Clasificación	%QX2.6	Coil 22
Robot_Verde_Pinza	Clasificación	%QX2.7	Coil 23
Robot_Verde_Horario	Clasificación	%QX3.0	Coil 24
Robot_Verde_Antihorario	Clasificación	%QX3.1	Coil 25
Robot_Azul_X	Clasificación	%QX3.2	Coil 26
Robot_Azul_Z	Clasificación	%QX3.3	Coil 27
Robot_Azul_Pinza	Clasificación	%QX3.4	Coil 28
Robot_Azul_Horario	Clasificación	%QX3.5	Coil 29
Robot_Azul_Antihorario	Clasificación	%QX3.6	Coil 30
Warning_Light_Clasificacion	Clasificación	%QX3.7	Coil 31
Emisor_Pales_Azul	Almacenaje	%QX0.0	Coil 32
Roller_Inicio_Front_Azul	Almacenaje	%QX0.1	Coil 33
Roller_Inicio_Back_Azul	Almacenaje	%QX0.2	Coil 34
Chain_Transfer_Front_Azul	Almacenaje	%QX0.3	Coil 35
Chain_Transfer_Back_Azul	Almacenaje	%QX0.4	Coil 36
Roller_Transfer_Azul	Almacenaje	%QX0.5	Coil 37
Roller_Pale_Azul	Almacenaje	%QX0.6	Coil 38
Roller_Curva_Azul	Almacenaje	%QX0.7	Coil 39
Emisor_Pales_Verde	Almacenaje	%QX1.0	Coil 40
Roller_Inicio_Front_Verde	Almacenaje	%QX1.1	Coil 41
Roller_Inicio_Back_Verde	Almacenaje	%QX1.2	Coil 42
Chain_Transfer_Front_Verde	Almacenaje	%QX1.3	Coil 43
Chain_Transfer_Back_Verde	Almacenaje	%QX1.4	Coil 44
Roller_Transfer_Verde	Almacenaje	%QX1.5	Coil 45
Roller_Pale_Verde	Almacenaje	%QX1.6	Coil 46
Roller_Curva_Verde	Almacenaje	%QX1.7	Coil 47
	-	%QX2.0	Coil 48
Turntable_Giro	Almacenaje	%QX2.1	Coil 49
Turntable_Roll_IZ	Almacenaje	%QX2.2	Coil 50
Turntable_Roll_DE	Almacenaje	%QX2.3	Coil 51
Roller_Pre_Posicion	Almacenaje	%QX2.4	Coil 52
Roller_Posicion_FIN	Almacenaje	%QX2.5	Coil 53
	-	%QX2.6	Coil 54
Stacker_Coger	Almacenaje	%QX2.7	Coil 55
Stacker_Dejar	Almacenaje	%QX3.0	Coil 56
Stacker_Levantar	Almacenaje	%QX3.1	Coil 57
	-	%QX3.2	Coil 58
	-	%QX3.3	Coil 59
Warning_Light_Pales	Almacenaje	%QX3.4	Coil 60
Warning_Light_Cuadro	Almacenaje	%QX3.5	Coil 61
	-	%QX3.6	Coil 62

	-	%QX3.7	Coil 63
Stacker_Posicion	Almacenaje	%QW2	H.R. 4
Progreso_Azules	Clasificación	%QW3	H.R. 5
Prograso_Verdes	Clasificación	%QW4	H.R. 6





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería del Diseño

---

### **Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO**

#### **Pliego de condiciones**

##### **TRABAJO FINAL DEL**

Grado en Ingeniería Electrónica Industrial y Automática

##### **REALIZADO POR**

Sergio Bou Grau

##### **TUTORIZADO POR**

Antonio Correcher Salvador

**CURSO ACADÉMICO: 2021/2022**



## **Contenido**

1.	Definición y alcance del pliego .....	4
2.	Condiciones y normas de carácter general.....	4
3.	Condiciones particulares .....	5
1.1.	Condiciones técnicas .....	5
1.1.1.	Materiales .....	5
1.1.2.	Ejecución del proyecto .....	5
1.2.	Condiciones facultativas.....	6
1.3.	Condiciones económicas .....	6
1.4.	Condiciones legales .....	7



## **1. Definición y alcance del pliego**

Este documento tiene como objeto establecer las condiciones técnicas mínimas que debe cumplir la instalación de un sistema de control y automatización sobre un proceso industrial basado en el manipulado del producto objeto, su transportación sobre medios electromecánicos y su almacenaje, especificando los requisitos de durabilidad, fiabilidad y seguridad.

Se extiende el ámbito de aplicación de este documento a todos los componentes y sistemas mecánicos, hidráulicos, eléctricos y electrónicos que forman parte de la instalación.

Debido al actual y creciente desarrollo tecnológico, en determinados supuestos se podrán adoptar soluciones alternativas a las exigidas en este documento, siempre y cuando queden expresamente justificadas o aporten una mejora significativa en el ámbito económico o sobre la productividad y no suponga una disminución de las exigencias mínimas especificadas o de calidad.

## **2. Condiciones y normas de carácter general**

Todos los componentes que componen este proyecto han de ser conectados a la red eléctrica de corriente alterna de 220V y 50Hz, así que están bajo el régimen del Reglamento Electrotécnico de Baja Tensión (REBT). También deberemos de tener en cuenta las siguientes instrucciones técnicas complementarias:

- ITC-BT-19: Instalaciones interiores o receptoras. Prescripciones generales.
- ITC-BT-20: Instalaciones interiores o receptoras. Sistemas de instalación.
- ITC-BT-23: Instalaciones interiores o receptoras. Protección contra sobretensiones.
- ITC-BT-43 Instalación de receptores. Prescripciones generales.
- ITC-BT-51: Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.

Así como la norma:

- UNE 20 514 92 1M: Reglas de seguridad para los aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uso general análogo conectados a una red de energía.

### **3. Condiciones particulares**

A continuación, se detallarán las condiciones particulares técnicas, facultativas, económicas y legales aplicables a este proyecto y de obligado cumplimiento para la realización de este proyecto.

#### **1.1. Condiciones técnicas**

##### **1.1.1. Materiales**

La empresa destinada a realizar la instalación del sistema de control será la encargada de elegir los materiales y mostrará documentalmente con detalle y brevedad suficiente para su análisis al cliente y al director del proyecto los componentes elegidos antes del inicio de la instalación de los mismos.

Los materiales destinados al control de la instalación, así como los componentes para su instalación, deberán de constar de la característica de poder ser empleados no solo para abarcar este proyecto si no para que si en un futuro se desea ampliar la instalación deberán de poder ser válidos para cubrir una ampliación de al menos un 25% más del presente proyecto.

La adquisición de licencias para obtener la funcionalidad total de los productos de software necesarios para el control, manejo y seguimiento del proyecto deberán ser de una duración mínima de un año.

El mantenimiento de las licencias necesarias para el continuado uso del software de control pasará a ser responsabilidad del cliente pasado el primer año después de la instalación.

##### **1.1.2. Ejecución del proyecto**

Los elementos para el control de la instalación serán instalados por una empresa con capacidad y recursos suficientes para abordar la totalidad del proyecto en el plazo estipulado y con los materiales adecuados para que el sistema funcione correctamente.

La empresa instaladora deberá de ofrecer al menos un año de supervisión y garantía de funcionamiento del sistema en el funcionamiento de las piezas de nueva instalación.

La empresa instaladora no podrá paralizar la instalación del proyecto una vez iniciada hasta que no tenga un mínimo de funcionamiento.

## 1.2. Condiciones facultativas

La empresa encargada de la instalación de la planta deberá de ofrecer al cliente al menos dos alternativas que cubran todos los requisitos principales del proyecto, y ofrecerle alternativas que satisfagan la calidad de la instalación, así como su coste.

La empresa encargada de la instalación de la planta será la encargada de diseñar y crear los planos necesarios previos a la instalación y deberán de aprobarse por parte del cliente y la parte de diseño de software si es otra distinta a la instaladora.

La empresa instaladora de la planta deberá certificar que conoce y mantiene actualizado el conocimiento de toda la normativa que regula este tipo de instalaciones tanto de seguridad de los materiales como de salud e higiene en el trabajo, para lo cual deberá de presentar un plan que acompañe a la documentación de la instalación.

La empresa instaladora de la planta deberá de disponer de los seguros, permisos y certificados necesarios que cubran toda la obra y no impidan el progreso normal de instalación de la planta en caso de accidente.

Los encargados de supervisar la instalación de la planta deberán de certificar su labor de seguimiento y comprobación del estadio del proyecto firmando cada día laboral en el libro de órdenes presente en la zona de instalación de la planta.

## 1.3. Condiciones económicas

Antes del inicio del proyecto se pactarán entre las partes de instalación de la planta, el desarrollador del sistema de control de la misma y el cliente tres porcentajes de pago que se realizarán al inicio del proyecto en un momento intermedio y al final del mismo.

La totalidad del pago del proyecto no se efectuará hasta que no se certifique por todas las partes que la planta es totalmente funcional y no se requieren más cambios ni modificaciones.

En el caso de no haber terminado el proyecto en el plazo estipulado por cualquier parte participante en el proyecto habrá una penalización económica del 2% por cada semana de retraso.

## 1.4. Condiciones legales

La empresa instaladora de la planta deberá de acreditar que su actividad principal está relacionada con la solicitada para la realización de este proyecto.

Para la adjudicación de la empresa instaladora deberá de existir consenso entre la parte de implementación del control de la planta y el cliente una vez revisados los candidatos y sus presupuestos.

Una vez decididas las partes y/o empresas participantes en el proyecto se formalizará un contrato de obra con fecha de finalización acordada entre todas las partes.

La responsabilidad de cada parte y/o empresa que participe en el proyecto se hará cargo de los daños ocasionados por sus empleados y por sus negligencias sobre los materiales y las instalaciones donde se instalarán.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería del Diseño

---

### **Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO**

## **Presupuesto**

### **TRABAJO FINAL DEL**

Grado en Ingeniería Electrónica Industrial y Automática

### **REALIZADO POR**

Sergio Bou Grau

### **TUTORIZADO POR**

Antonio Correcher Salvador

**CURSO ACADÉMICO: 2020/20**



## Contenido

1.	Precios unitarios.....	5
1.1.	<b>Cuadro de mano de obra</b> .....	5
1.2.	<b>Cuadro de materiales</b> .....	8
2.	Precios descompuestos.....	9
2.1.	<b>Anejo de justificación de precios</b> .....	9
2.2.	<b>Cuadro de precios</b> .....	13
3.	Resumen de presupuesto.....	14



## 1. Precios unitarios

### 1.1. Cuadro de mano de obra

Cuadro de mano de obra

Num. Código	Denominación de la mano de obra	Precio	Horas	Total
1 ALM001	Este fichero, escrito mediante diagrama funcional secuencial, controla el apilador que ubica cada palé en una posición determinada según el color que de la pieza que transporta.	35,000	4,000 h	140,00
2 ALM002	Escrito con lenguaje texto estructurado, en este fichero se realizan operaciones aritméticas y booleanas que dependen de condicionales. En él se realizan los incrementos al contabilizar las piezas o se determina la posición de los palés.	35,000	4,000 h	140,00
3 ALM003	Este fichero comprende las variables internas que dependen de una o varias señales para gestionar los movimientos de los palés si según las respuestas de los diferentes sensores. Está escrito mediante lenguaje Ladder.	35,000	3,500 h	122,50
4 ALM004	Escrito mediante diagrama funcional secuencial, este fichero controla el girador de palés que dependiendo del lado por el que venga, el azul o el verde, hace girar en un sentido u otro su base.	35,000	5,500 h	192,50
5 ALM005	Para el control del suministro de palés, su traslación hasta el girador o su retirada en caso de que se deposite una pieza defectuosa sobre el palé en el lado de las piezas azules se ha creado este fichero mediante diagrama funcional secuencial.	35,000	3,000 h	105,00
6 ALM006	Para el control del suministro de palés, su traslación hasta el girador o su retirada en caso de que se deposite una pieza defectuosa sobre el palé en el lado de las piezas verdes se ha creado este fichero mediante diagrama funcional secuencial.	35,000	3,000 h	105,00
7 ALM007	Este fichero tiene la finalidad de comunicar al sistema la llegada del palé al final de los transportadores de palés, está escrito en diagrama funcional secuencial.	35,000	4,000 h	140,00
8 CLA001	Este fichero está creado mediante el lenguaje Ladder y su propósito es gestionar casi todos los inputs realizados desde la interfaz de usuario.	35,000	3,000 h	105,00
9 CLA002	Escrito mediante el lenguaje de programación Ladder, la finalidad de este fichero es enlazar las variables locales desde distintos ficheros con las entradas del sistema a controlar.	35,000	2,500 h	87,50
10 CLA003	Fichero creado mediante el lenguaje Texto estructurado, donde se realizan operaciones aritméticas o booleanas que dependen de condicionales.	35,000	2,000 h	70,00

## Cuadro de mano de obra

Num. Código	Denominación de la mano de obra	Precio	Horas	Total	
11	CLA004	En este fichero se gestionan los valores obtenidos desde los diferentes sensores que leen el color o forma de las piezas o mediante sensores basados en luz infrarroja se determina la presencia de las piezas cuando son movidas por las cintas o llegan a una ubicación en concreto, este fichero está creado en lenguaje Ladder.	35,000	2,500 h	87,50
12	CLA005	Este fichero forma parte del sistema de control de errores a distancia, cuya finalidad es poder continuar con la producción en el caso de que ocurra un error en la posición de las piezas para su ensamblaje o que los robots no cojan bien las piezas superiores, cuyo acto provoca que no sea posible su ensamblaje. La finalidad es poder dar el poder de realizar dos acciones cuando ocurra un error, una es limpiar las cintas y la otra es reposicionar las piezas nuevamente. Escrito mediante diagrama funcional secuencial.	35,000	4,000 h	140,00
13	CLA006	En este fichero, escrito en diagrama funcional secuencial, se controlan las diferentes cintas y empujadores en función del color o forma de la pieza leída por el sensor.	35,000	3,000 h	105,00
14	CLA007	Mediante este fichero es posible controlar el sistema de forma manual desde la HMI, una vez seleccionado este modo es posible seleccionar las piezas una a una del color que se desee pudiendo realizar ensamblajes con distinto color. Este fichero está escrito en diagrama funcional secuencial.	35,000	2,500 h	87,50
15	CLA008	Este fichero, creado mediante diagrama funcional secuencial, controla los movimientos del robot encargado de ensamblar y transportar las piezas azules.	35,000	3,500 h	122,50
16	CLA009	Este fichero, creado mediante diagrama funcional secuencial, controla los movimientos del robot encargado de ensamblar y transportar las piezas verdes.	35,000	3,000 h	105,00
17	CLA010	Esta visualización surge cuando se pulsa el botón "Nuevo pedido" en la HMI, su finalidad es advertir al usuario de que se borrara el pedido en curso.	35,000	2,500 h	87,50
18	CLA011	Esta visualización surge cuando se pulsa el botón "Borrar progreso", su finalidad es avisar al usuario de que se reiniciarán los contadores y no se tendrá en cuenta las piezas realizadas hasta el momento.	35,000	2,000 h	70,00
19	CLA012	En esta visualización están definidos todos los botones, luces y contadores que permiten transmitir los deseos del usuario al sistema, entre otras cosas se puede realizar pedidos, visualizar el progreso en tiempo real, cambiar de modo de funcionamiento, resolver atascos en la producción a distancia o realizar paradas de emergencia.	35,000	4,000 h	140,00
20	FAC002	Configuración de entradas y salidas del PLC simulado en Factory IO y la conexión Modbus.	30,000	3,000 h	90,00
21	FAC001	Montaje de proceso industrial a controlar: posicionamiento de los elementos que conforman el proceso, ajuste de sensores y posicionamiento de elementos de seguridad.	20,000	8,000 h	160,00

Cuadro de mano de obra

Num. Código	Denominación de la mano de obra	Precio	Horas	Total
22 FAC003	Pruebas de funcionamiento de los distintos ficheros y partes de la parte de control para corroborar el funcionamiento del sistema.	20,000	29,000 h	580,00
23 MEM001	Redacción de memoria	15,000	34,000 h	510,00
24 PLA001	Redacción de planos.	15,000	3,000 h	45,00
25 PLI001	Pliego de condiciones	15,000	2,000 h	30,00
26 PRE001	Presupuesto	15,000	2,000 h	30,00
			Total mano de obra:	3.597,50

## 1.2. Cuadro de materiales

Cuadro de materiales

Num.	Código	Denominación del material	Precio	Cantidad	Total
1	INS001	Controlador M241 40 ES Transistor NPN Ethernet	452,110	1,000 ud	452,11
2	INS003	Armario eléctrico compacto Rittal AX 1045.000 - 400 x 500 x 210	85,790	1,000 ud	85,79
3	INS002	Fuente de alimentación conmutada PHOENIX 2902993 - UNO-PS/1AC/24DC/100W	55,800	1,000 ud	55,80
4	INS004	Carriles para montaje interior (4 unidades) Rittal AX 2394.210	29,040	1,000 ud	29,04
5	INS005	Soportes para pared (4 unidades) Rittal SZ 2508.020	14,280	1,000 ud	14,28
6	INS006	Latiguillo de red RJ45 LAPP ETHERLINE LAN Cat.6A RJ45 5,0 GY - 24441205	12,460	1,000 ud	12,46
7	COD001	Licencia para Codesys programación subsistema almacenaje.	0,278	27,000 h	7,51
8	COD002	Licencia para Codesys programación subsistema clasificación	0,278	34,500 h	9,59
9	COD003	Licencia para Codesys simulación en Factory IO	0,278	29,000 h	8,06
10	CON001	Portátil - Asus F415EA-EK1005W, 14" Full-HD, Intel® Core™ i5-1135G7, 8 GB, 512 GB, Iris® X?, Windows 11 H	0,250	95,000 h	23,75
11	CON002	Portátil - Asus F415EA-EK1005W, 14" Full-HD, Intel® Core™ i5-1135G7, 8 GB, 512 GB, Iris® X?, Windows 11 H	0,250	95,000 h	23,75
12	FACT03	Licencia para Factory IO proceso de simulación.	0,141	29,000 h	4,09
13	FACT02	Licencia para Factory IO configuración del sistema simulado.	0,141	3,000 h	0,42
14	FACT01	Licencia para Factory IO montaje del sistema simulado.	0,141	8,000 h	1,13
				Total materiales:	727,78

## 2. Precios descompuestos

### 2.1. Anejo de justificación de precios

Anejo de justificación de precios

Num.	Código	Ud	Descripción	Precio/ud	Total
<b>1</b>	<b>DOC001</b>		Memoria		
	MEM001	34,000 h	Redacción de memoria	15,000	510,00
				Total:	510,00
			Son QUINIENTOS DIEZ EUROS.		
<b>2</b>	<b>DOC002</b>		Redacción del pliego de condiciones del proyecto.		
	PLI001	2,000 h	Pliego de condiciones	15,000	30,00
				Total:	30,00
			Son TREINTA EUROS.		
<b>3</b>	<b>DOC003</b>		Redacción del presupuesto del proyecto.		
	PRE001	2,000 h	Presupuesto	15,000	30,00
				Total:	30,00
			Son TREINTA EUROS.		
<b>4</b>	<b>DOC004</b>		Redacción de planos.		
	PLA001	3,000 h	Planos	15,000	45,00
				Total:	45,00
			Son CUARENTA Y CINCO EUROS.		

Anejo de justificación de precios

Num.	Código	Ud	Descripción	Precio/ud	Total
------	--------	----	-------------	-----------	-------

**5 LIC001** El coste de la licencia del software Codesys Professional Developer Edition es de 500€/año, estimando que en un año se puede usar durante 1800 horas laborales se obtiene un coste por hora de 0,278€ Precio/ud.

COD002		34,500 h	Programación Clasificación	0,278	9,59
COD001		27,000 h	Programación Almacenaje	0,278	7,51
COD003		29,000 h	Simulación en Factory IO	0,278	8,06
Total:					25,16

Son VEINTICINCO EUROS CON DIECISEIS CÉNTIMOS.

**6 LIC002** El coste de la licencia del software Factory IO es de 253€/año, estimando que durante un año puede utilizarse durante unas 1800 horas laborales se obtiene un coste por hora de 0,141€ Precio/ud.

FACT01		8,000 h	Montaje de proceso	0,141	1,13
FACT02		3,000 h	Configuración de conexiones	0,141	0,42
FACT03		29,000 h	Pruebas de funcionamiento	0,141	4,09
Total:					5,64

Son CINCO EUROS CON SESENTA Y CUATRO CÉNTIMOS.

**7 MAT001** Para el cálculo de los siguientes precios se ha estimado el uso de dos computadoras durante 95 horas cada una. Y se ha estimado un precio de 1500€ por computadora y una vida útil de 10.000 horas. Por lo que se ha obtenido un precio/ud de 0,150 €/h.

CON001		95,000 h	Computadora 1	0,250	23,75
CON002		95,000 h	Computadora 2	0,250	23,75
Total:					47,50

Son CUARENTA Y SIETE EUROS CON CINCUENTA CÉNTIMOS.

Anejo de justificación de precios

Num.	Código	Ud	Descripción	Precio/ud	Total
<b>8</b>	<b>MAT002</b>		Materiales necesarios para la instalación del PLC		
	INS001	1,000 ud	PLC	452,110	452,11
	INS002	1,000 ud	Fuente de alimentación	55,800	55,80
	INS003	1,000 ud	Cuadro electrico	85,790	85,79
	INS004	1,000 ud	Carriles para montaje	29,040	29,04
	INS005	1,000 ud	Soportes para pared	14,280	14,28
	INS006	1,000 ud	Latiguillo de red	12,460	12,46
			<b>Total:</b>		<b>649,48</b>

Son SEISCIENTOS CUARENTA Y NUEVE EUROS CON CUARENTA Y OCHO CÉNTIMOS.

**9 PRO001** Programación clasificación

	CLA001	3,000 h	Botones HMI	35,000	105,00
	CLA002	2,500 h	Control clasificación	35,000	87,50
	CLA003	2,000 h	Control datos	35,000	70,00
	CLA004	2,500 h	Data IN	35,000	87,50
	CLA005	4,000 h	Limpiar ajustar	35,000	140,00
	CLA006	3,000 h	Modo automático	35,000	105,00
	CLA007	2,500 h	Modo manual	35,000	87,50
	CLA008	3,500 h	Robot unir piezas azules	35,000	122,50
	CLA009	3,000 h	Robot unir piezas verdes	35,000	105,00
	CLA010	2,500 h	Aviso nuevo pedido	35,000	87,50
	CLA011	2,000 h	Aviso reset progreso	35,000	70,00
	CLA012	4,000 h	HMI	35,000	140,00
			<b>Total:</b>		<b>1.207,50</b>

Son MIL DOSCIENTOS SIETE EUROS CON CINCUENTA CÉNTIMOS.

Anejo de justificación de precios

Num.	Código	Ud	Descripción	Precio/ud	Total
------	--------	----	-------------	-----------	-------

**10 PRO002** Programación almacenaje

ALM001		4,000 h	Almacenamiento	35,000	140,00
ALM002		4,000 h	Control almacenaje	35,000	140,00
ALM003		3,500 h	Gestión almacenaje	35,000	122,50
ALM004		5,500 h	Girar palé	35,000	192,50
ALM005		3,000 h	Inicio palé azul	35,000	105,00
ALM006		3,000 h	Inicio palé verde	35,000	105,00
ALM007		4,000 h	Palé girado	35,000	140,00
				<b>Total:</b>	<b>945,00</b>

Son NOVECIENTOS CUARENTA Y CINCO EUROS.

**11 PRO003** Generar proyecto simulado en Factory IO

FAC001		8,000 h	Montaje de proceso	20,000	160,00
FAC002		3,000 h	Configuración conexiones	30,000	90,00
FAC003		29,000 h	Pruebas de funcionamiento	20,000	580,00
				<b>Total:</b>	<b>830,00</b>

Son OCHOCIENTOS TREINTA EUROS.

## 2.2. Cuadro de precios

Cuadro de precios			
Nº	Designación	Importe	
		En cifra (Euros)	En letra (Euros)
1	Memoria	510,00	QUINIENTOS DIEZ EUROS
2	Redacción del pliego de condiciones del proyecto.	30,00	TREINTA EUROS
3	Redacción del presupuesto del proyecto.	30,00	TREINTA EUROS
4	Redacción de planos.	45,00	CUARENTA Y CINCO EUROS
5	El coste de la licencia del software Codesys Professional Developer Edition es de 500€/año, estimando que en un año se puede usar durante 1800 horas laborales se obtiene un coste por hora de 0,278€ Precio/ud.	25,16	VEINTICINCO EUROS CON DIECISEIS CÉNTIMOS
6	El coste de la licencia del software Factory IO es de 253€/año, estimando que durante un año puede utilizarse durante unas 1800 horas laborales se obtiene un coste por hora de 0,141€ Precio/ud.	5,64	CINCO EUROS CON SESENTA Y CUATRO CÉNTIMOS
7	Para el cálculo de los siguientes precios se ha estimado el uso de dos computadoras durante 95 horas cada una. Y se ha estimado un precio de 1500€ por computadora y una vida útil de 10.000 horas. Por lo que se ha obtenido un precio/ud de 0,150 €/h.	47,50	CUARENTA Y SIETE EUROS CON CINCUENTA CÉNTIMOS
8	Materiales necesarios para la instalación del PLC	649,48	SEISCIENTOS CUARENTA Y NUEVE EUROS CON CUARENTA Y OCHO CÉNTIMOS
9	Programación clasificación	1.207,50	MIL DOSCIENTOS SIETE EUROS CON CINCUENTA CÉNTIMOS
10	Programación almacenaje	945,00	NOVECIENTOS CUARENTA Y CINCO EUROS
11	Generar proyecto simulado en Factory IO	830,00	OCHOCIENTOS TREINTA EUROS

### 3. Resumen del presupuesto

Proyecto: Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO

<b>Capítulo</b>	<b>Importe</b>
Capítulo 1 Partes de la memoria	615,00
Capítulo 2 Software utilizado	30,80
Capítulo 3 Instalación y manejo	696,98
Capítulo 4 Programación	2.982,50
Presupuesto de ejecución material	4.325,28
10% de gastos generales	432,53
5% de beneficio industrial	216,26
Suma	4.974,07
21% IVA	1.044,55
Presupuesto de ejecución por contrata	<b>6.018,62</b>

Asciende el presupuesto de ejecución por contrata a la expresada cantidad de SEIS MIL DIECIOCHO EUROS CON SESENTA Y DOS CÉNTIMOS.

En \_\_\_\_\_ a \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

Firma cliente:

Firma ingeniero:





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería del Diseño

---

### **Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO**

## **Planos**

#### **TRABAJO FINAL DEL**

Grado en Ingeniería Electrónica Industrial y Automática

#### **REALIZADO POR**

Sergio Bou Grau

#### **TUTORIZADO POR**

Antonio Correcher Salvador

**CURSO ACADÉMICO: 2021/2022**



# Contenido

<b>1. Planos</b> .....	5
1.1. Plano posicionamiento de sensores en la parte de clasificación .....	5
1.1.1. Plano posicionamiento de sensores en la parte de clasificación .....	6
1.2. Plano posicionamiento de sensores en la parte de almacenamiento .....	7
1.2.1. Plano posicionamiento de sensores en la parte de almacenaje .....	8
1.3. Plano posicionamiento de actuadores en la parte de clasificación .....	9
1.3.1. Plano posicionamiento de actuadores en la parte de clasificación .....	10
1.4. Plano posicionamiento de actuadores en la parte de almacenaje .....	11
1.4.1. Plano posicionamiento de actuadores en la parte de almacenaje .....	12



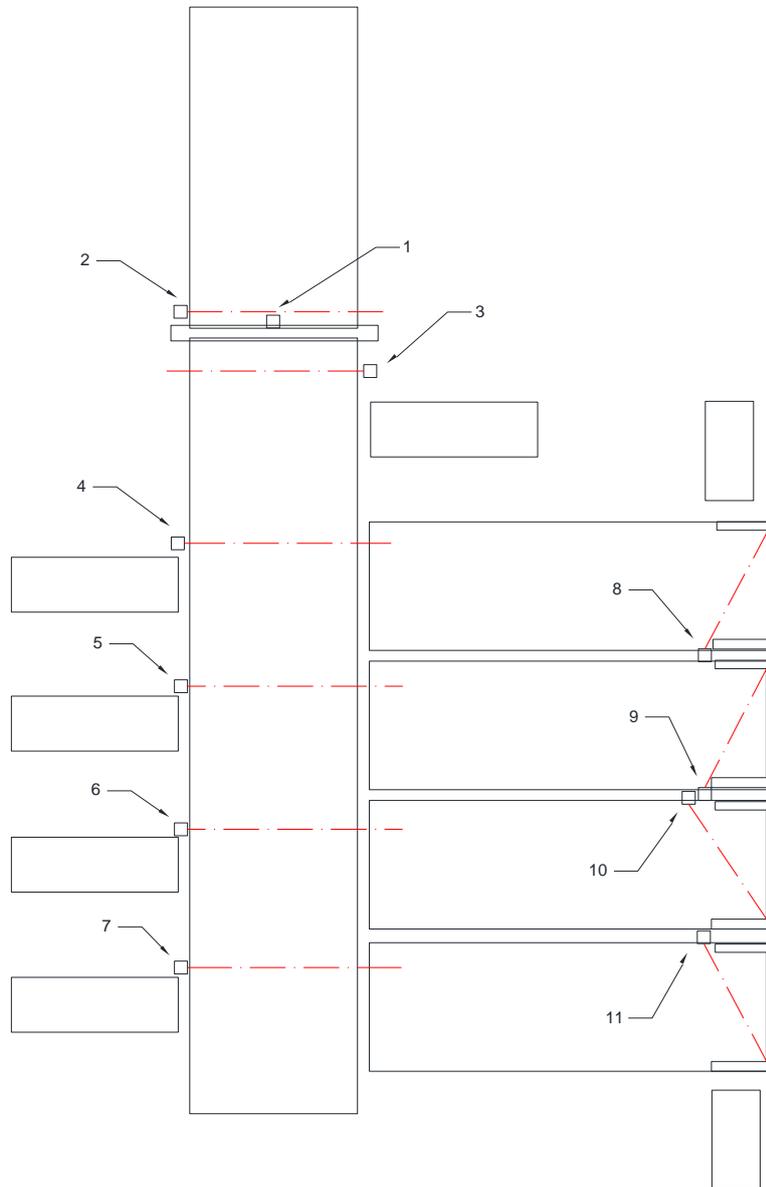
## 1. Planos

Para tener una mejor perspectiva de la distribución de la planta simulada y una lista de los elementos como sensores, actuadores y demás partes que componen el proceso se muestran a continuación planos y listas donde es más rápida su consulta que directamente en el programa y puede ayudar a realizar un cableado óptimo.

### 1.1. Plano posicionamiento de sensores en la parte de clasificación

El siguiente plano contiene todos los sensores infrarrojos y de captación de piezas usados en la parte de clasificación, también contiene una lista para poder localizarlos y conocer su denominación en la programación y su dirección usada en el PLC.

### 1.1.1. Plano posicionamiento de sensores en la parte de clasificación



Numeración de sensores

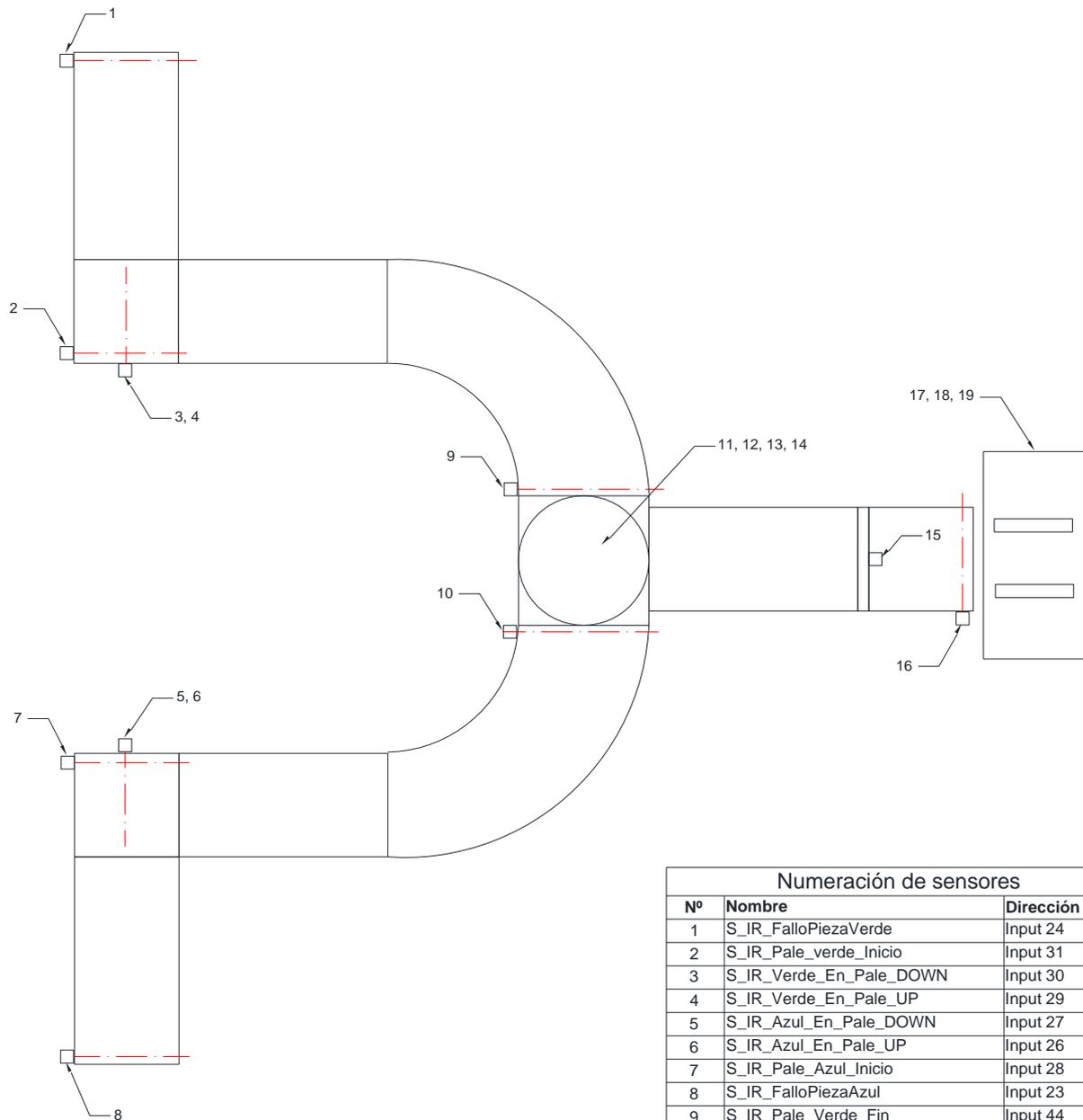
Nº	Nombre	Dirección PLC
1	S_Color_Inicio_Piezas	Input Reg 3
2	S_IR_Inicio_Piezas	Input 8
3	S_IR_Pusher_Descartes	Input 9
4	S_IR_Pusher_Tapa_Verde	Input 10
5	S_IR_Pusher_Base_Verde	Input 11
6	S_IR_Pusher_Base_Azul	Input 12
7	S_IR_Pusher_Tapa_Azul	Input 13
8	S_IR_Robot_Tapa_Verde	Input 4
9	S_IR_Robot_Base_Verde	Input 5
10	S_IR_Robot_Base_Azul	Input 6
11	S_IR_Robot_Tapa_Azul	Input 7

PROYECTO: Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcassas mediante Codesys y Factory IO		Fecha: 19/3/2022
TITULAR: Sergio Bou Grau Referencia: 0001		Escala 1:1
Autor: Sergio Bou Grau	Plano:	Plano Nº
I.T.I. Pre-colegiado nº: 50217	SENSORES CLASIFICACIÓN	01

## 1.2. Plano posicionamiento de sensores en la parte de almacenamiento

El siguiente plano contiene todos los sensores infrarrojos y de captación de piezas usados en la parte de almacenaje, también contiene una lista para poder localizarlos y conocer su denominación en la programación y su dirección usada en el PLC.

## 1.2.1. Plano posicionamiento de sensores en la parte de almacenaje



Numeración de sensores

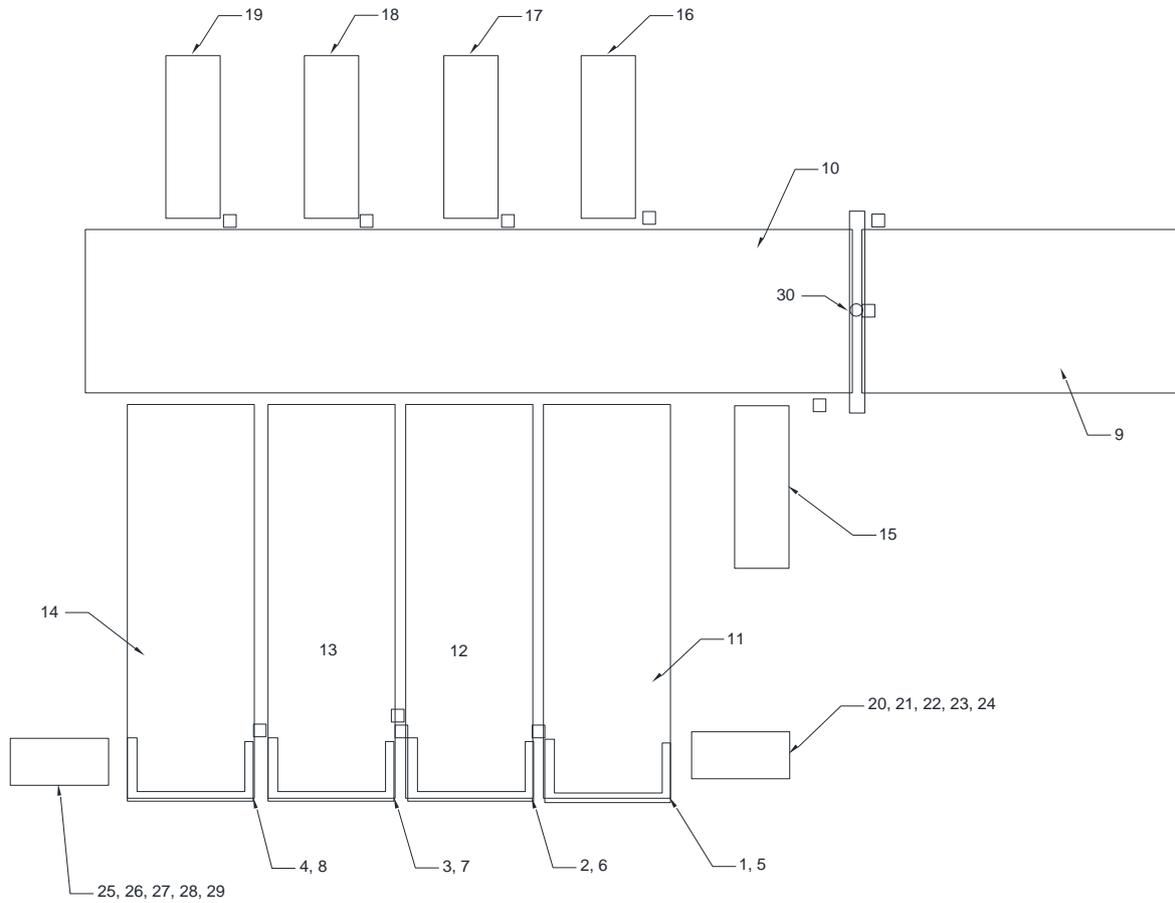
Nº	Nombre	Dirección PLC
1	S_IR_FalloPiezaVerde	Input 24
2	S_IR_Pale_verde_Inicio	Input 31
3	S_IR_Verde_En_Pale_DOWN	Input 30
4	S_IR_Verde_En_Pale_UP	Input 29
5	S_IR_Azul_En_Pale_DOWN	Input 27
6	S_IR_Azul_En_Pale_UP	Input 26
7	S_IR_Pale_Azul_Inicio	Input 28
8	S_IR_FalloPiezaAzul	Input 23
9	S_IR_Pale_Verde_Fin	Input 44
10	S_IR_Pale_Azul_Fin	Input 43
11	S_Turtable_DE	Input 33
12	S_Turtable_IJ	Input 34
13	S_Turtable_EXIT	Input 35
14	S_Turtable_IN	Input 36
15	S_Color_FIN	Input Reg 4
16	S_IR_Pale_En_posicion	Input 37
17	S_Stacker_Centrado	Input 39
18	S_Stacker_moviendose_X	Input 40
19	S_Stacker_Moviendose_Z	Input 41

PROYECTO: Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO		Fecha: 19/3/2022
TITULAR: Sergio Bou Grau Referencia: 0002		Escala 1:1
Autor: Sergio Bou Grau	Plano: SENSORES ALMACENAJE	Plano Nº 02
I.T.I. Pre-colegiado nº: 50217		

### 1.3. Plano posicionamiento de actuadores en la parte de clasificación

El siguiente plano contiene todos los actuadores usados en la parte de clasificación, también contiene una lista para poder localizarlos y conocer su denominación en la programación y su dirección usada en el PLC.

### 1.3.1. Plano posicionamiento de actuadores en la parte de clasificación



Numeración de actuadores

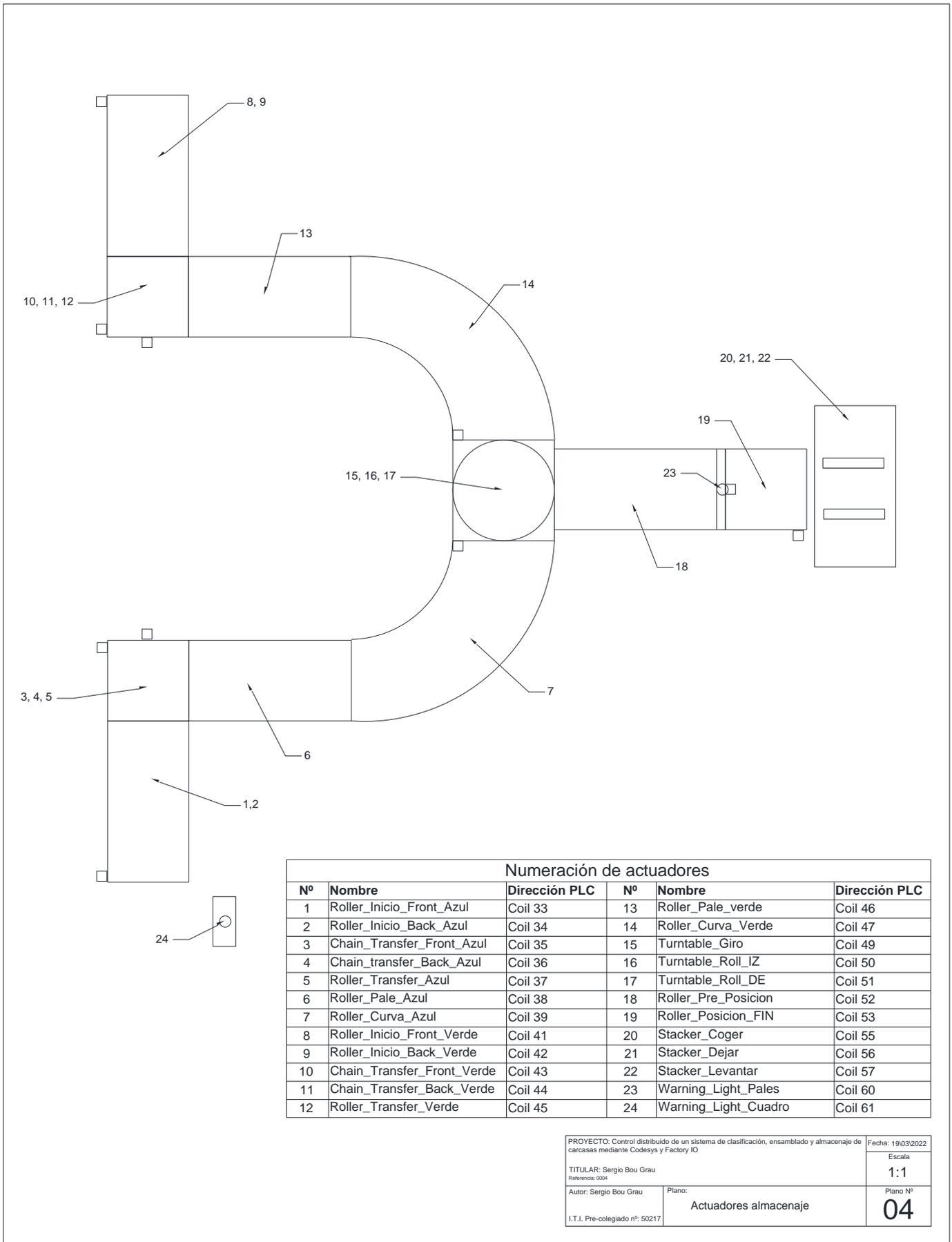
Nº	Nombre	Dirección PLC	Nº	Nombre	Dirección PLC
1	Pos_Tapa_verde_Y	Coil 0	16	Pusher_Tapa_Verde	Coil 17
2	Pos_Base_Verde_Y	Coil 1	17	Pusher_Base_Verde	Coil 18
3	Pos_Base_Azul_Y	Coil 2	18	Pusher_Base_Azul	Coil 19
4	Pos_Tapa_Azul_Y	Coil 3	19	Pusher_Tapa_Azul	Coil 20
5	Pos_Tapa_verde_X	Coil 4	20	Robot_Verde_X	Coil 21
6	Pos_Base_Verde_X	Coil 5	21	Robot_Verde_Z	Coil 22
7	Pos_Base_Azul_X	Coil 6	22	Robot_Verde_Pinza	Coil 23
8	Pos_Tapa_Azul_X	Coil 7	23	Robot_Verde_Horario	Coil 24
9	Cinta_Inicio_Piezas	Coil 9	24	Robot_Verde_Antihorario	Coil 25
10	Cinta_Pushers	Coil 10	25	Robot_Azul_X	Coil 26
11	Cinta_Tapa_verde	Coil 11	26	Robot_Azul_Z	Coil 27
12	Cinta_Base_Verde	Coil 12	27	Robot_Azul_Pinza	Coil 28
13	Cinta_Base_Azul	Coil 13	28	Robot_Azul_Horario	Coil 29
14	Cinta_Tapa_Azul	Coil 14	29	Robot_Azul_Antihorario	Coil 30
15	Pusher_Descartes	Coil 16	30	Warning_Light_Clasificacion	Coil 31

PROYECTO: Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO		Fecha: 19/3/2022
TITULAR: Sergio Bou Grau Referencia: 003		Escala 1:1
Autor: Sergio Bou Grau	Plano: ACTUADORES CLASIFICACIÓN	Plano Nº 03
I.T.I. Pre-colegiado nº: 50217		

#### 1.4. Plano posicionamiento de actuadores en la parte de almacenaje

El siguiente plano contiene todos los actuadores usados en la parte de almacenaje, también contiene una lista para poder localizarlos y conocer su denominación en la programación y su dirección usada en el PLC.

### 1.4.1. Plano posicionamiento de actuadores en la parte de almacenaje



PROYECTO: Control distribuido de un sistema de clasificación, ensamblado y almacenaje de carcasas mediante Codesys y Factory IO		Fecha: 19/03/2022
TITULAR: Sergio Bou Grau Referencia: 0004		Escala 1:1
Autor: Sergio Bou Grau	Plano: Actuadores almacenaje	Plano Nº 04
I.T.I. Pre-colegiado nº: 50217		

