

Trabajo Fin de Grado  
Ingeniería Electrónica, Robótica y Mecatrónica

Diseño de sistemas para control mediante autómatas programables en Factory I/O y comunicación con Codesys

Autor: Abraham Cano Muñoz

Tutor: Antonio Javier Gallego Len

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2023





Trabajo Fin de Grado  
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

# **Diseño de sistemas para control mediante autómatas programables en Factory I/O y comunicación con Codesys**

Autor:

Abraham Cano Muñoz

Tutor:

Antonio Javier Gallego Len

Profesor Sustituto Interino

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023



Trabajo Fin de Grado: Diseño de sistemas para control mediante autómatas programables en Factory I/O y comunicación con Codesys

Autor: Abraham Cano Muñoz

Tutor: Antonio Javier Gallego Len

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



*A mi familia*

*A mis amigos*

*A mi equipo médico*





# Agradecimientos

---

Desde mi infancia, mis padres me han transmitido el refrán “*Es de bien nacido ser agradecido*”, así que quiero empezar expresando mi gratitud hacia ellos, quienes han sido y siempre serán los pilares fundamentales de mi vida. Su apoyo incondicional desde el inicio ha sido crucial, destacando la constancia, la lucha y la superación de mi madre, así como la fortaleza de mi padre en los momentos difíciles.

Me gustaría hacer una mención especial al equipo médico del Hospital Reina Sofía de Córdoba que me ha tratado. Gracias a su dedicación y esfuerzo, hoy en día disfruto de una salud que me permite seguir adelante con mi vida.

Asimismo, quiero agradecer a mi familia y mis amigos por su apoyo fundamental, los ánimos que me han brindado cuando la situación ha sido complicada, y la felicidad que me han transmitido.

Por último, pero no menos importante, quería expresar mi agradecimiento a mi tutor, por su comprensión y disponibilidad, pero sobre todo, por el entusiasmo puesto en este trabajo.

Para concluir, me gustaría hacer mención a otro refrán: “*Felicidad no es hacer lo que uno quiere, sino querer lo que uno hace*”.

*Abraham Cano Muñoz*

*Estudiante autor*

*Sevilla, 2023*



**E**ste proyecto está enfocado en el uso de tecnologías avanzadas de automatización industrial, concretamente en el uso de Controladores Lógicos Programables (PLCs) de forma virtual.

Por un lado, se trabajará con FACTORY IO, software donde se diseñarán los sistemas que se pretenden controlar. Este programa permite crear sistemas virtuales de automatización industrial donde, entre otros componentes, se puede implementar un panel de control que brinde al usuario la posibilidad de interactuar con el entorno. Además, ofrece una simulación realista de los sistemas que facilita la interpretación para determinar si el funcionamiento está siendo el deseado o no.

Por otro lado, se trabajará con CODESYS V3.5, desde donde se realizará el control. Este software permite programar las distintas prestaciones y funcionalidades de las que dispondrán los sistemas y se encargará de gestionar las respuestas correspondientes a la interacción del usuario.



# Abstract

---

In the context of advanced industrial automation technologies, this project focuses on the application of Programmable Logic Controllers (PLCs) in a virtual environment. The project is made up of two parts: the simulation of the system in a virtual environment in FACTORY IO, and the implementation of the control algorithms for each of the systems in CODESYS V3.5.

The systems were designed in the software FACTORY IO. This program allows you to create virtual industrial automation systems where, among other components, a control panel can be implemented that gives the user the possibility to interact with the environment. In addition, it offers a realistic simulation of the systems, facilitating an interpretation to determine whether or not the desired operation is being executed.

The control is carried out in the software CODESYS V3.5. This software allows you to program the different features and functionalities that the systems will have, and will be in charge of managing the responses corresponding to the user interaction.



# Índice abreviado

---

<b>Agradecimientos</b>	<b>9</b>
<b>Resumen</b>	<b>11</b>
<b>Abstract</b>	<b>13</b>
<b>Índice abreviado</b>	<b>15</b>
<b>Índice</b>	<b>17</b>
<b>Índice de Figuras</b>	<b>19</b>
<b>Índice de Tablas</b>	<b>23</b>
<b>1 Introducción</b>	<b>1</b>
1.1 <i>Motivación</i>	1
1.2 <i>Propósito</i>	1
1.3 <i>Objetivos</i>	1
<b>2 Software implementado</b>	<b>3</b>
2.1 <i>CODESYS V3.5</i>	3
2.2 <i>FACTORY IO</i>	3
<b>3 Comunicación software</b>	<b>5</b>
3.1 <i>CODESYS - OPC</i>	5
3.2 <i>FACTORY IO – OPC</i>	10
<b>4 Clasificador y paletizador de cajas</b>	<b>11</b>
4.1 <i>DESCRIPCIÓN</i>	11
4.2 <i>PRESTACIONES</i>	12
4.3 <i>DESARROLLO SOFTWARE</i>	15
4.4 <i>RESULTADO</i>	26
<b>5 Formador y Clasificador de Conjuntos</b>	<b>27</b>
5.1 <i>DESCRIPCIÓN</i>	27
5.2 <i>PRESTACIONES</i>	28
5.3 <i>DESARROLLO SOFTWARE</i>	32
5.4 <i>RESULTADO</i>	41
<b>6 CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>42</b>
<b>Referencias</b>	<b>43</b>
<b>Anexo A: Modularización</b>	<b>45</b>
<b>Anexo B: Guía Gemma</b>	<b>47</b>
1. <i>Introducción</i>	47
2. <i>Aplicación</i>	48
2.1 <i>MODOS</i>	48
2.2 <i>CAMINOS</i>	48





<b>Agradecimientos</b>	<b>9</b>
<b>Resumen</b>	<b>11</b>
<b>Abstract</b>	<b>13</b>
<b>Índice abreviado</b>	<b>15</b>
<b>Índice</b>	<b>17</b>
<b>Índice de Figuras</b>	<b>19</b>
<b>Índice de Tablas</b>	<b>23</b>
<b>1 Introducción</b>	<b>1</b>
1.1 <i>Motivación</i>	1
1.2 <i>Propósito</i>	1
1.3 <i>Objetivos</i>	1
<b>2 Software implementado</b>	<b>3</b>
2.1 <i>CODESYS V3.5</i>	3
2.1.1 Descripción	3
2.1.2 Uso	3
2.2 <i>FACTORY IO</i>	3
2.2.1 Descripción	3
2.2.1 Uso	3
<b>3 Comunicación software</b>	<b>5</b>
3.1 <i>CODESYS - OPC</i>	5
3.1.1 Crear un nuevo proyecto	5
3.1.2 Definir las variables globales	5
3.1.3 Iniciar controlador	6
3.1.4 Conexión CODESYS – OPC	7
3.1.5 Registro de Usuario	8
3.1.6 Habilitar comunicación CODESYS – FACTORY IO	8
3.1.7 Conceder permisos al servidor	8
3.1.8 Simular las funcionalidades de un PLC	9
3.1.9 Configuración de símbolos	9
3.1.10 Crear las variables globales en el servidor	10
3.1.11 Ejecución del programa	10
3.2 <i>FACTORY IO – OPC</i>	10
3.2.1 Seleccionar driver de conexión	10
3.2.2 Mostrar variables cargadas en el servidor	11
<b>4 Clasificador y paletizador de cajas</b>	<b>11</b>
4.1 <i>DESCRIPCIÓN</i>	11
4.2 <i>PRESTACIONES</i>	12
4.2.1 Paneles de control	12
4.2.2 Modos de funcionamiento	14
4.2.3 Estado de emergencia	15
4.3 <i>DESARROLLO SOFTWARE</i>	15

4.3.1	Introducción	15
4.3.2	FACTORY IO	16
4.3.3	CODESYS [5] [6] [7] [8]	21
4.4	<i>RESULTADO</i>	26
<b>5</b>	<b>Formador y Clasificador de Conjuntos</b>	<b>27</b>
5.1	<i>DESCRIPCIÓN</i>	27
5.2	<i>PRESTACIONES</i>	28
5.2.1	Panel de control	28
5.2.2	Modos de funcionamiento	30
5.2.3	Estado de emergencia	31
5.3	<i>DESARROLLO SOFTWARE</i>	32
5.3.1	FACTORY IO	32
5.3.2	CODESYS	35
5.4	<i>RESULTADO</i>	41
<b>6</b>	<b>CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>42</b>
	<b>Referencias</b>	<b>43</b>
	<b>Anexo A: Modularización</b>	<b>45</b>
	<b>Anexo B: Guía Gemma</b>	<b>47</b>
	1. <i>Introducción</i>	47
	2. <i>Aplicación</i>	48
	2.1 MODOS	48
	2.2 CAMINOS	48

# ÍNDICE DE FIGURAS

---

<b>Figura 3.1</b> Creación de un nuevo proyecto	5
<b>Figura 3.2</b> Añadir lista de variables globales	6
<b>Figura 3.3</b> Configurar lista de variables globales	6
<b>Figura 3.4</b> Servidor OPC	7
<b>Figura 3.5</b> Conexión CODESYS - OPC	7
<b>Figura 3.6</b> Registro de usuario/Inicio de sesión	8
<b>Figura 3.7</b> Administrar usuarios del dispositivo	8
<b>Figura 3.8</b> Permisos para el servidor	9
<b>Figura 3.9</b> Simulación del PLC	9
<b>Figura 3.10</b> Configuración de símbolos	9
<b>Figura 3.11</b> Cargar variables globales en el servidor	10
<b>Figura 3.12</b> Ejecución del programa	10
<b>Figura 3.13</b> Driver de conexión	10
<b>Figura 3.14</b> Visualización de las variables	11
<b>Figura 4.1</b> Vista general del sistema	11
<b>Figura 4.2</b> Cajas de FACTORY IO: paletizadora, pequeña, mediana y grande	11
<b>Figura 4.3</b> Paneles de control	12
<b>Figura 4.4</b> Cintas clasificadoras	13
<b>Figura 4.5</b> Cintas cajas pequeñas	13
<b>Figura 4.6</b> Cintas zona de descarga	13
<b>Figura 4.7</b> Cinta paletizador	14
<b>Figura 4.8</b> Configuración del emisor	16
<b>Figura 4.9</b> Elementos de FACTORY IO	16
<b>Figura 4.10</b> Emisor como <i>'Controller-defined'</i>	17
<b>Figura 4.11</b> Eliminador	18
<b>Figure 4.12</b> Sensor de presencia	18
<b>Figura 4.13</b> Emisor y receptor de matriz luminosa	19
<b>Figura 4.14</b> Transferencia de cadena	19
<b>Figura 4.15</b> Pick & Place	20
<b>Figura 4.16</b> Paletizador	20
<b>Figure 4.17</b> Botones normalmente cerrados	21
<b>Figure 4.18</b> Cintas transportadoras	21
<b>Figura 4.19</b> Organización del proyecto	21
<b>Figura 4.20</b> Carpeta AUTOMATICO	22

<b>Figura 4.21</b>	<b>Programación en LD</b>	<b>22</b>
<b>Figura 4.22</b>	<b>Modos de funcionamiento</b>	<b>23</b>
<b>Figura 4.23</b>	<b>Programación ST</b>	<b>23</b>
<b>Figura 4.24</b>	<b>Carpeta EMERGENCIA</b>	<b>23</b>
<b>Figura 4.25</b>	<b>Detención de la estación 'Pick &amp; Place'</b>	<b>23</b>
<b>Figura 4.26</b>	<b>Carpeta FACTORY</b>	<b>24</b>
<b>Figura 4.27</b>	<b>Contador para generar cajas</b>	<b>24</b>
<b>Figura 4.28</b>	<b>Aproximación de los puntos del eje x del 'Pick &amp; Place'</b>	<b>25</b>
<b>Figura 4.29</b>	<b>Carpeta MANUAL</b>	<b>25</b>
<b>Figura 4.30</b>	<b>Carpeta TEMPORIZADORES Y CONTADORES</b>	<b>25</b>
<b>Figura 4.31</b>	<b>CTUD asociado a las cintas dirigidas al paletizador</b>	<b>26</b>
<b>Figura 5.1</b>	<b>Vista general del sistema</b>	<b>27</b>
<b>Figura 5.2</b>	<b>Elementos de FACTOY IO: materias primas, tapas, bases y conjuntos</b>	<b>27</b>
<b>Figure 5.3</b>	<b>Panel de control</b>	<b>28</b>
<b>Figura 5.4</b>	<b>Cintas transportadoras iniciales</b>	<b>29</b>
<b>Figura 5.5</b>	<b>Cintas transportadoras centrales</b>	<b>29</b>
<b>Figura 5.6</b>	<b>Puntos de saturación</b>	<b>30</b>
<b>Figure 5.7</b>	<b>Centro de mecanizado</b>	<b>33</b>
<b>Figure 5.8</b>	<b>Sensor de visión</b>	<b>33</b>
<b>Figure 5.9</b>	<b>Brazo robótico</b>	<b>34</b>
<b>Figure 5.10</b>	<b>Posicionador</b>	<b>34</b>
<b>Figure 5.11</b>	<b>Cinta transportadora y brazo pivotante</b>	<b>35</b>
<b>Figura 5.12</b>	<b>Organización del proyecto</b>	<b>35</b>
<b>Figura 5.13</b>	<b>Carpeta AUTOMATICO</b>	<b>36</b>
<b>Figura 5.14</b>	<b>Programación en SFC</b>	<b>36</b>
<b>Figura 5.15</b>	<b>Contador CTUD</b>	<b>37</b>
<b>Figura 5.16</b>	<b>Habilitar SFCReset</b>	<b>37</b>
<b>Figura 5.17</b>	<b>Accion RESET del POU ModosFunc</b>	<b>37</b>
<b>Figura 5.18</b>	<b>Carpeta EMERGENCIA</b>	<b>38</b>
<b>Figura 5.19</b>	<b>Comienzo del reinicio</b>	<b>39</b>
<b>Figura 5.20</b>	<b>Carpeta FACTORY</b>	<b>39</b>
<b>Figura 5.21</b>	<b>Materia prima aleatoria para bases</b>	<b>39</b>
<b>Figura 5.22</b>	<b>Materia prima aleatoria para tapas</b>	<b>40</b>
<b>Figura 5.23</b>	<b>Carpeta MANUAL</b>	<b>40</b>
<b>Figura 5.24</b>	<b>Fragmento del modo manual</b>	<b>40</b>
<b>Figura 5.25</b>	<b>Carpeta TEMPORIZADORES Y CONTADORES</b>	<b>40</b>
<b>Figura 5.26</b>	<b>Fragmento de los temporizadores</b>	<b>41</b>
<b>Figura 5.27</b>	<b>Temporizador en SFC</b>	<b>41</b>

<b>Figura A.1</b> Modularización	<b>45</b>
<b>Figura B.1</b> Guía GEMMA	<b>47</b>
<b>Figura B.1</b> Parada a fin de ciclo	<b>48</b>
<b>Figura B.2</b> Funcionamiento manual	<b>49</b>
<b>Figura B.3</b> Pulsador de emergencia	<b>49</b>
<b>Figura B.4</b> Parada y reanudación en una etapa determinada	<b>49</b>



# ÍNDICE DE TABLAS

---

<b>Tabla 4–1.</b> Identificadores ' <i>Part</i> '	17
<b>Tabla 4–2.</b> Identificadores ' <i>Base</i> '	17
<b>Tabla 4–3.</b> Cajas generadas	24
<b>Tabla 5–1.</b> Identificadores ' <i>Part</i> '	32





# 1 INTRODUCCIÓN

---

## 1.1 Motivación

La automatización industrial está avanzando significativamente en las últimas décadas. La necesidad de una mayor eficiencia en la producción y reducción de los costos de producción, está llevando a muchas empresas a adoptar tecnologías avanzadas de automatización, sistemas de control avanzados y flexibles como los PLC.

El uso de los Controladores Lógicos Programables (PLCs), permite controlar una gran variedad de procesos industriales, desde simples líneas de producción, hasta sistemas de alta complejidad. Estos dispositivos pueden operar de forma continua y son capaces de ejecutar tareas repetitivas con alta precisión y velocidad, ofreciendo una gran confiabilidad, lo que aumenta la eficiencia del proceso y reduce los errores humanos.

Existe una gran variedad de programas para controlar los PLCs, algunos ejemplos son CODESYS, Unity Pro y TIA Portal. La mayoría de fabricantes de PLCs proporcionan su propio software de programación y control.

Para el diseño de los sistemas también se cuenta con un elevado número de programas que se centran en la simulación y visualización de sistemas de automatización, como por ejemplo FACTORY IO, Siemens PLCSIM Advanced, 3D Factory Simulation y PLC-Lab.

## 1.2 Propósito

El propósito de este proyecto es realizar simulaciones de sistemas automatizados industriales antes de su implementación en un sistema real, lo que permite comprobar su correcto funcionamiento y minimizar el riesgo y el costo asociado a posibles errores durante el proceso de producción.

De esta forma se demostrará la capacidad y la versatilidad de las que disponen los sistemas controlados por PLCs, y la gran ventaja que supone realizar este control en una simulación antes de su implementación en una industria real.

## 1.3 Objetivos

El proyecto tiene como finalidad lograr los siguientes objetivos:

- Investigar los distintos softwares disponibles y seleccionar los que mejor se adapten a las necesidades del proyecto.
- Investigar cómo establecer conexión entre los softwares escogidos.
- Diseñar sistemas que sean prácticos y factibles de ser implementados por una empresa.
- Llevar a cabo un control siguiendo la norma IEC 61131<sup>1</sup> y tomando como referencia la guía Gemma.

---

<sup>1</sup> La norma IEC (International Electrotechnical Commission) 61131 es un conjunto de normas e informes técnicos publicados con el objetivo de definir los estándares de programación de los autómatas programables



# 2 SOFTWARE IMPLEMENTADO

---

## 2.1 CODESYS V3.5

### 2.1.1 Descripción

CODESYS V3.5 es una herramienta software, concretamente un entorno de desarrollo integrado (IDE)<sup>2</sup>, utilizado en la industria para la automatización de procesos y sistemas, que permite la programación y configuración de diferentes tipos de controladores.

Dispone de una interfaz de usuario que permite la configuración de distintos dispositivos de automatización, como controladores de lógica programable (PLCs), sistemas de control distribuido (DCS) y sistemas de supervisión y control de procesos (SCADA).

Para comunicarse con estos controladores, admite distintos protocolos, como son: OPC DC/UA, Ethernet/IP, Modbus, etc., y para la programación de los mismos dispone de una amplia variedad de lenguajes de programación, entre los que destacar: diagrama de contactos (LD), texto estructurado (ST) y diagrama de funciones secuenciales (SFC), también llamado GRAFCET (Grafos de Control Etapa-Transición). [2]

### 2.1.2 Uso

CODESYS V3.5, a diferencia de la versión anterior, CODESYS V2.3, dispone del CODESYS Control Win V3. Esta herramienta se trata de un servidor OPC que se ejecuta en el sistema operativo Windows, y que permite establecer comunicación con otros programas en los que desarrollar sistemas que pueden ser controlados desde CODESYS.

## 2.2 FACTORY IO

### 2.2.1 Descripción

FACTORY IO es un software de simulación 3D que permite crear entornos virtuales para simular procesos industriales automatizados. Incluye una amplia gama de componentes, entre los que destacar sensores, actuadores, cintas transportadoras, centros de mecanizado y brazos robóticos.

Este programa puede conectarse a un PLC virtual y es compatible con dispositivos Siemens, Schneider y Allen-Bradley, entre otros. Dado que estas marcas son las principales fabricantes de PLC, el control realizado puede ser directamente implementado en un sistema real.

Para controlar el sistema, se dispone de distintos softwares de programación de PLC: CODESYS V3.5, Siemens TIA Portal, Allen-Bradley Studio 5000, Schneider Unity Pro, etc. La conexión entre ambos programas se puede realizar a través de OPC DC/UA<sup>3</sup> o importando el archivo de programa a Factory IO. [3]

### 2.2.1 Uso

La variedad, el realismo y la versatilidad de sus componentes, sumado a la compatibilidad del programa

---

<sup>2</sup> Los entornos IDE proporcionan un conjunto de herramientas, como editor de código, compilador, depurador, linker y empaquetador, de forma que permiten programar y desarrollar software de una forma más eficiente y ágil. [1]

<sup>3</sup> OPC UA (Open Platform Communications Unified Architecture) es una tecnología de comunicación utilizada en la industria, que define un conjunto de protocolos y especificaciones para la comunicación de datos entre dispositivos y aplicaciones de diferentes fabricantes. Es una versión avanzada del modelo de comunicación OPC DA. [4]

con numerosas plataformas de automatización, entre ellas CODESYS V3.5, y contando con la comunicación OPC UA, hacen de FACTORY IO una herramienta muy eficiente y atractiva para el diseño de los sistemas industriales que se abarcan en este proyecto.

# 3 COMUNICACIÓN SOFTWARE

Este proyecto se enfoca en el uso de CODESYS V3.5 y FACTORY IO, los cuales se comunican entre sí a través del protocolo OPC UA mediante el servidor OPC proporcionado por CODESYS. Esta conexión se divide en dos partes: en primer lugar, la conexión de CODESYS V3.5 con el servidor OPC, CODESYS Control Win V3, mencionado previamente, y en segundo lugar, la conexión de FACTORY IO con dicho controlador.

Es importante tener en cuenta que las versiones utilizadas son CODESYS V3.5 SP18 Patch 2 y FACTORY IO V2.4.3. Dependiendo de las versiones instaladas, es posible que los pasos a seguir varíen.

## 3.1 CODESYS - OPC

### 3.1.1 Crear un nuevo proyecto

Seleccionar un 'Proyecto Standard' y el dispositivo que se encargará de controlar el sistema. Hay dos opciones, 'CODESYS Control Win V3 (3S – Smart Software Solutions GmbH)' y su versión 'x64'.

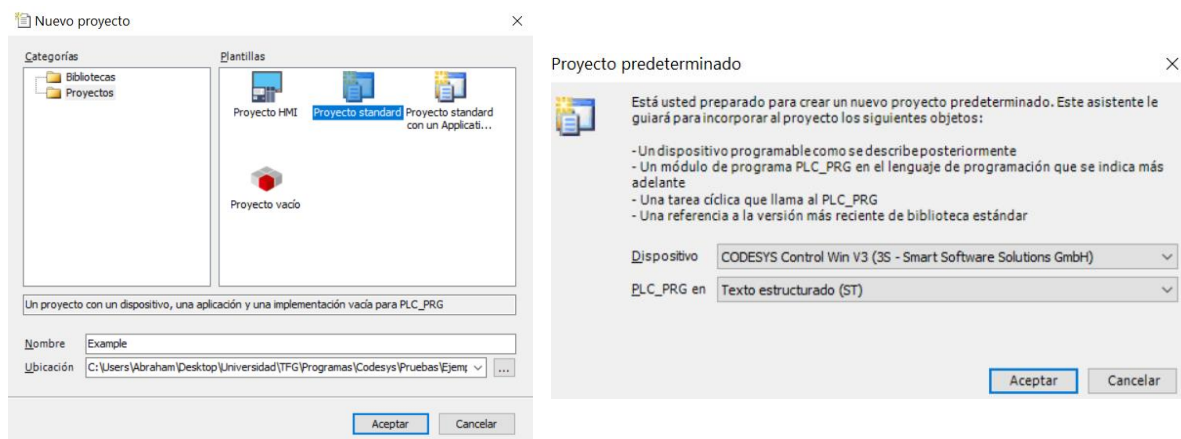


Figura 3.1 Creación de un nuevo proyecto

### 3.1.2 Definir las variables globales

Las variables globales serán las variables que estarán accesibles desde ambos programas, aquellas que no estén declaradas en CODESYS como globales, solo estarán accesibles internamente en este programa.

Para agregar la lista de variables globales, hacer click derecho en 'Application' → 'Agregar objeto' → 'Lista de variables globales...' y se le asignará un nombre. A modo de ejemplo se definen 3 variables:

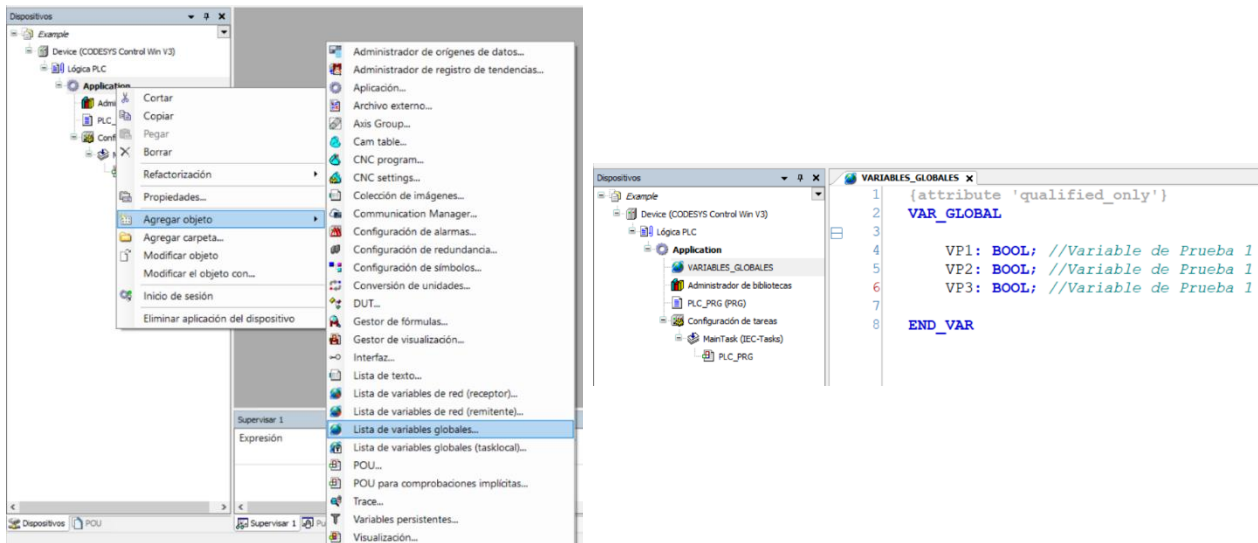


Figura 3.2 Añadir lista de variables globales

Click derecho sobre la lista recién creada, acceder a 'Propiedades' y en la sección 'Compilar' marcar 'Ligar siempre'. Esto le indicará al compilador que las variables globales creadas se asignarán a una dirección de memoria fija durante la ejecución, independientemente si la variable se está utilizando o no, de esta forma se descartarán problemas de lectura/escritura.

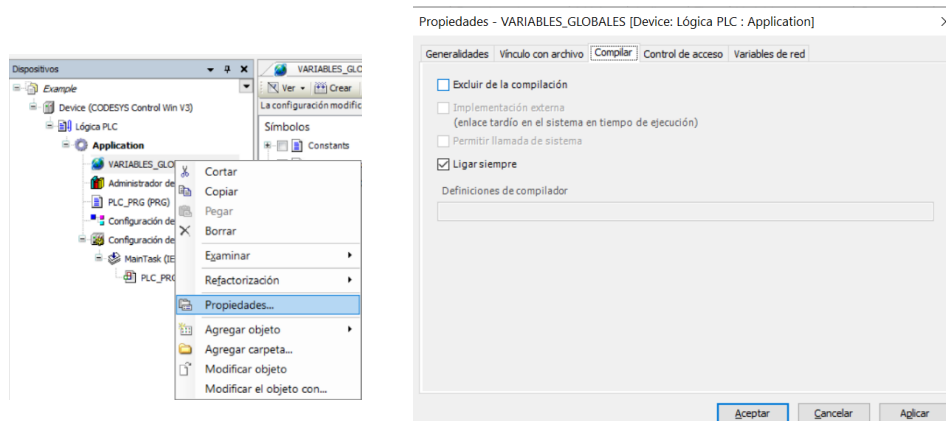


Figura 3.3 Configurar lista de variables globales

### 3.1.3 Iniciar controlador

Abrir el buscador de Windows, escribir el nombre del controlador y hacer click en él. En unos segundos habrá cargado y ya estará listo. Si se dispone de la versión demo, tras tenerlo abierto 2 horas dejará de funcionar, bastará con cerrarlo y abrirlo de nuevo para disponer de 2 horas adicionales.

En la información que ha ido mostrando el controlador, se puede observar la inicialización del servidor OPC UA y el nombre del Host en el que está siendo ejecutado<sup>4</sup>.

<sup>4</sup> Por motivos de privacidad, se han tapado los datos correspondientes al ordenador donde se ejecuta del servidor

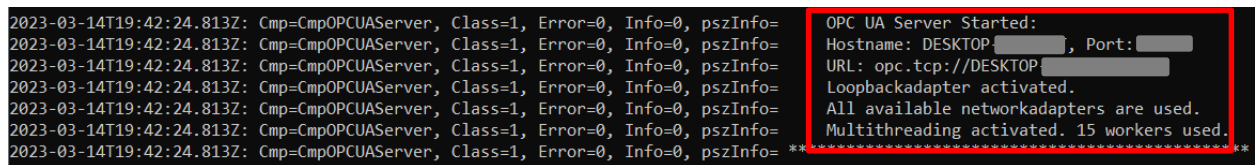
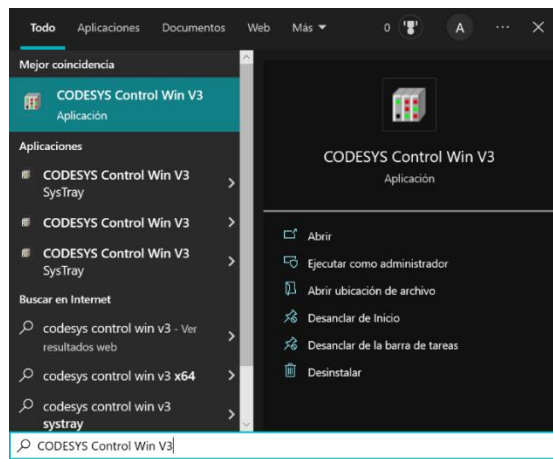


Figura 3.4 Servidor OPC

### 3.1.4 Conexión CODESYS – OPC

Hacer doble click en ‘Device (CODESYS Control Win V3)’ → ‘Examinar red’, seleccionar la ruta de red para el controlador, que se corresponde con el nombre del host obtenido en la imagen anterior, y se pulsa el botón de ‘Aceptar’. Si emerge una nueva pestaña, se debe pulsar ‘Si’.

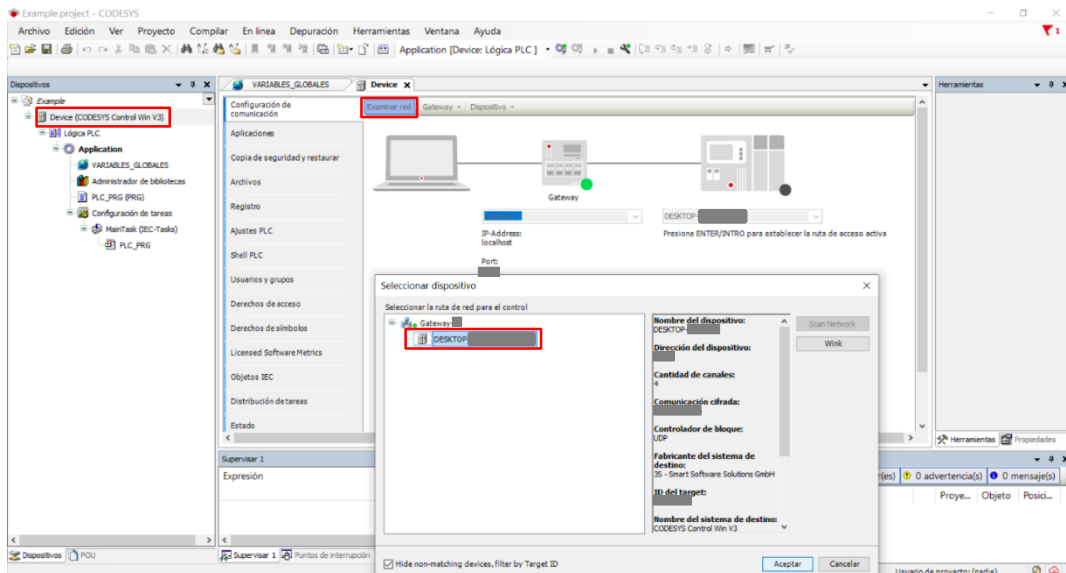


Figura 3.5 Conexión CODESYS - OPC

### 3.1.5 Registro de Usuario

Al establecer conexión por primera vez, deberá registrarse. Acto seguido, el programa pedirá iniciar sesión con el usuario y la contraseña recién registrada.

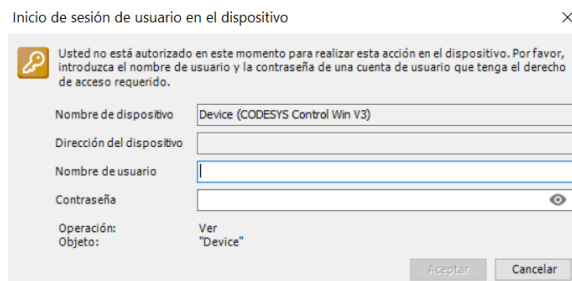


Figura 3.6 Registro de usuario/Inicio de sesión

### 3.1.6 Habilitar comunicación CODESYS – FACTORY IO

Continuando en la sección ‘Device (CODESYS Control Win V3)’, acceder a ‘Dispositivo’ → ‘Cambiar política de comunicación...’ y en el apartado ‘Administración de usuarios de dispositivos’ seleccionar en nueva política ‘Administración de usuarios opcional’. Marcar también la casilla ‘Permitir inicio de sesión anónimo’ y pulsar ‘Aceptar’.

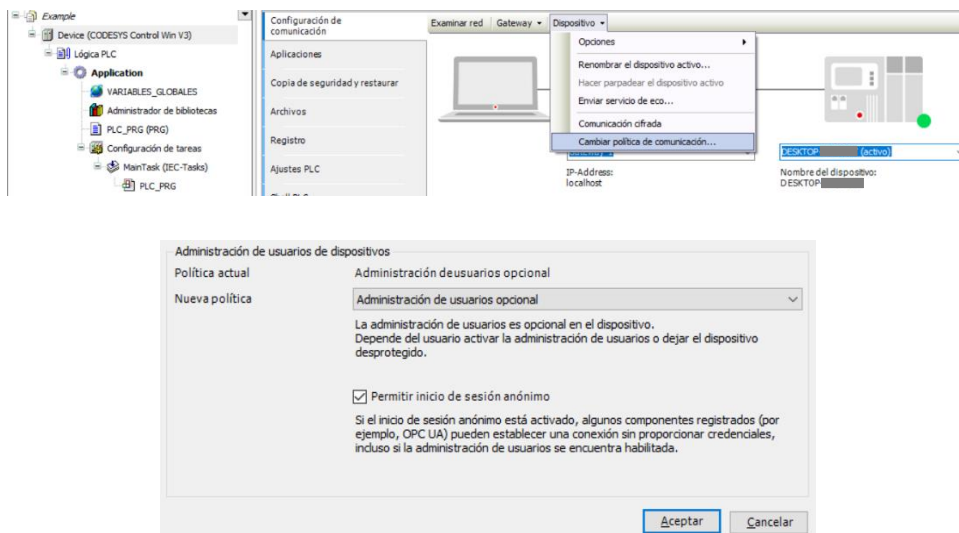


Figura 3.7 Administrar usuarios del dispositivo

### 3.1.7 Conceder permisos al servidor

Siguiendo en la misma sección, hacer click en ‘Derechos de acceso’ → sincronizar (fechas verdes) → ‘OPCUAServer’ y deben de estar permitidos los accesos ‘Cambiar’, ‘Vista’ y ‘Ejecutar’, de lo contrario, hacer doble click en la casilla correspondiente.



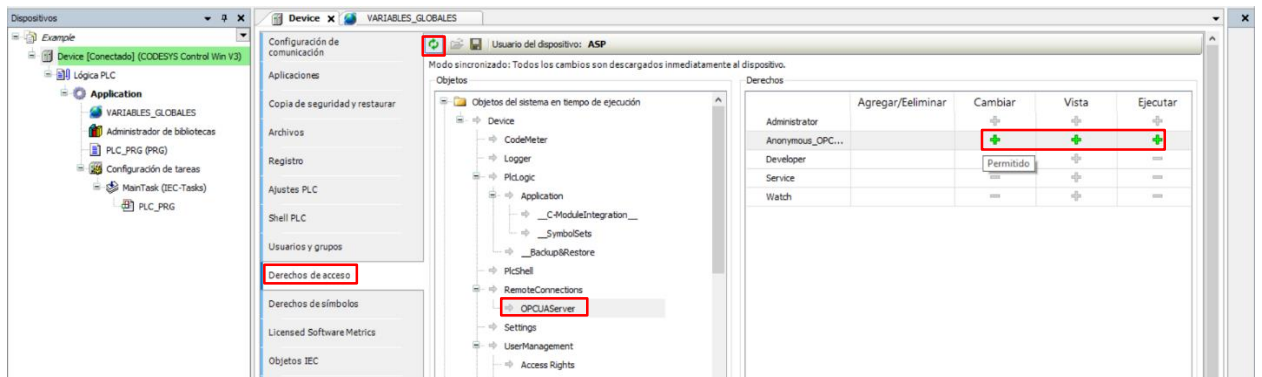


Figura 3.8 Permisos para el servidor

### 3.1.8 Simular las funcionalidades de un PLC

En la barra de herramientas pulsar ‘*Crear código*’ para descargar el programa al Software PLC.

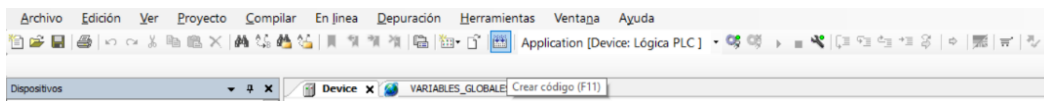


Figura 3.9 Simulación del PLC

### 3.1.9 Configuración de símbolos

Click derecho en ‘*Application*’ → ‘*Agregar objeto*’ → ‘*Configuración de símbolos...*’, marcar ‘*Compatible con características OPC UA*’ y pulsar ‘*Agregar*’.

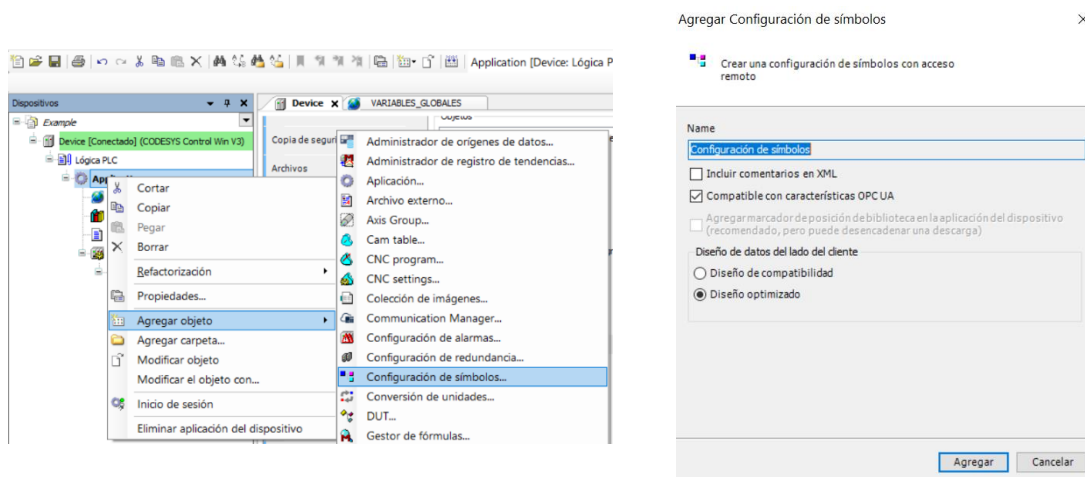


Figura 3.10 Configuración de símbolos

### 3.1.10 Crear las variables globales en el servidor

Ejecutar el comando ‘*Crear*’ para que aparezcan las listas de símbolos que se pueden seleccionar, marcar la casilla correspondiente a la lista de variables globales declara con anterioridad y de nuevo pulsar ‘*Crear*’.

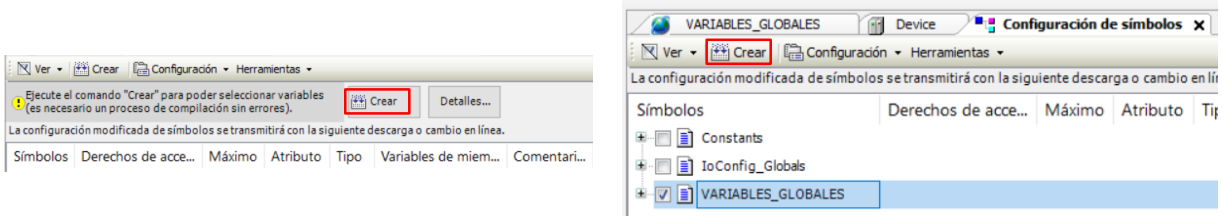


Figura 3.11 Cargar variables globales en el servidor

### 3.1.11 Ejecución del programa

En la barra de herramientas pulsar en el icono ‘*Iniciar Sesión*’ y en la pantalla emergente, marcar la primera opción y pulsar ‘*Aceptar*’. Por último, pulsar el botón de ‘*Inicio*’ de la barra de herramientas para iniciar el programa.

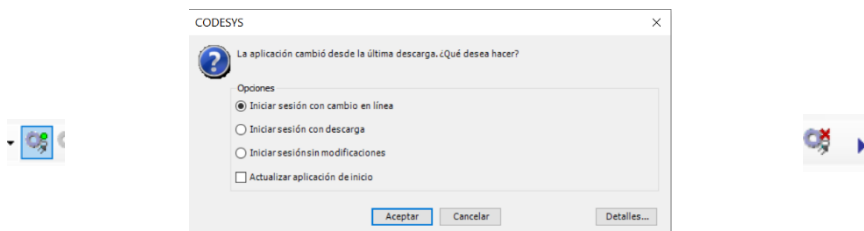


Figura 3.12 Ejecución del programa

## 3.2 FACTORY IO – OPC

### 3.2.1 Seleccionar driver de conexión

Abrir una escena y acceder a ‘*Archivo*’ → ‘*Drivers*’, seleccionar ‘*OPC Client DA/UA*’ y pulsar ‘*Configuración*’ en la parte superior derecha.

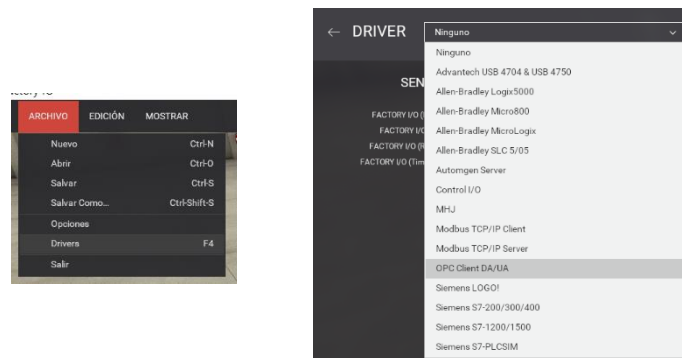
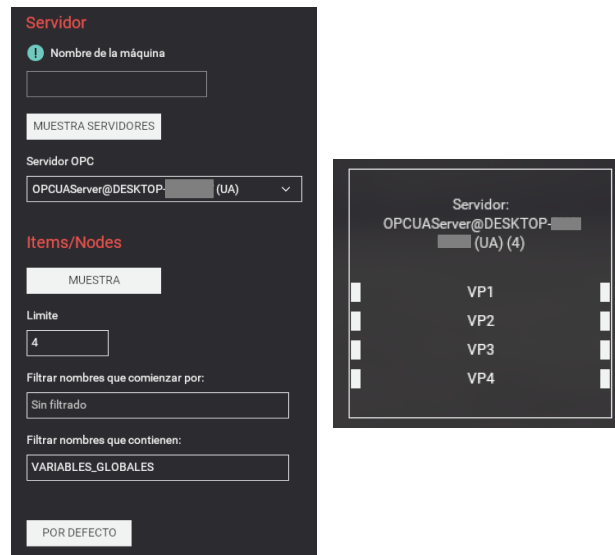


Figura 3.13 Driver de conexión

### 3.2.2 Mostrar variables cargadas en el servidor

En ‘*Servidor OPC*’, seleccionar la opción correspondiente a OPC UA y presionar la tecla intro del teclado del ordenador para que FACTORY IO se conecte con el servidor. Cuando se haya establecido conexión, el apartado ‘*Items/Nodes*’ estará accesible y se podrá escribir en los campos correspondientes. En el filtro introducir el nombre de la lista que contiene las variables globales y adaptar el límite a la cantidad de variables creadas, por último, pulsar en ‘*MUESTRA*’ y las variables ya estarán cargadas y accesibles.



**Figura 3.14** Visualización de las variables



# 4 CLASIFICADOR Y PALETIZADOR DE CAJAS

## 4.1 DESCRIPCIÓN

Se propone un sistema de clasificación de cajas que opera mediante un proceso de recepción y distribución. El sistema recibe cajas de diferentes tamaños en palets a través de una cinta transportadora, las clasifica y las distribuye a distintas cintas transportadoras en función de su altura: altas, medianas o bajas.

Una estación 'Pick & Place' se encarga de llevar las cajas paletizadas hacia una cinta transportadora específica que se dirige a la paletizadora, donde estas cajas se agrupan en palets de 18 unidades.

Dado que aproximadamente la mitad de las cajas que ingresan al sistema son paletizadas, la paletizadora se vuelve esencial para la eficiencia del proceso. Teniendo en cuenta la frecuencia a la que llegan estas cajas, para asegurar una operación fluida, las cintas transportadoras que llevan las cajas hacia la paletizadora deben estar en constante movimiento, evitando así tener que encender y apagar los motores de las cintas continuamente.



**Figura 4.1** Vista general del sistema



**Figura 4.2** Cajas de FACTORY IO: paletizadora, pequeña, mediana y grande

## 4.2 PRESTACIONES

### 4.2.1 Paneles de control

El sistema cuenta con dos paneles de control que permiten su supervisión y gestión.

El panel principal o primario permite gestionar el modo de funcionamiento, ofreciendo opciones tanto para el modo manual como para el automático. Además, en caso de emergencia, cuenta con elementos específicos como la seta de emergencia, el botón de rearme y la luz de emergencia.

Por otro lado, el panel de control secundario, diseñado para controlar el sistema en modo manual, cuenta con un indicador verde ubicado en la esquina superior izquierda, que se encenderá cuando el sistema esté funcionando en este modo para indicar que el panel está operativo.

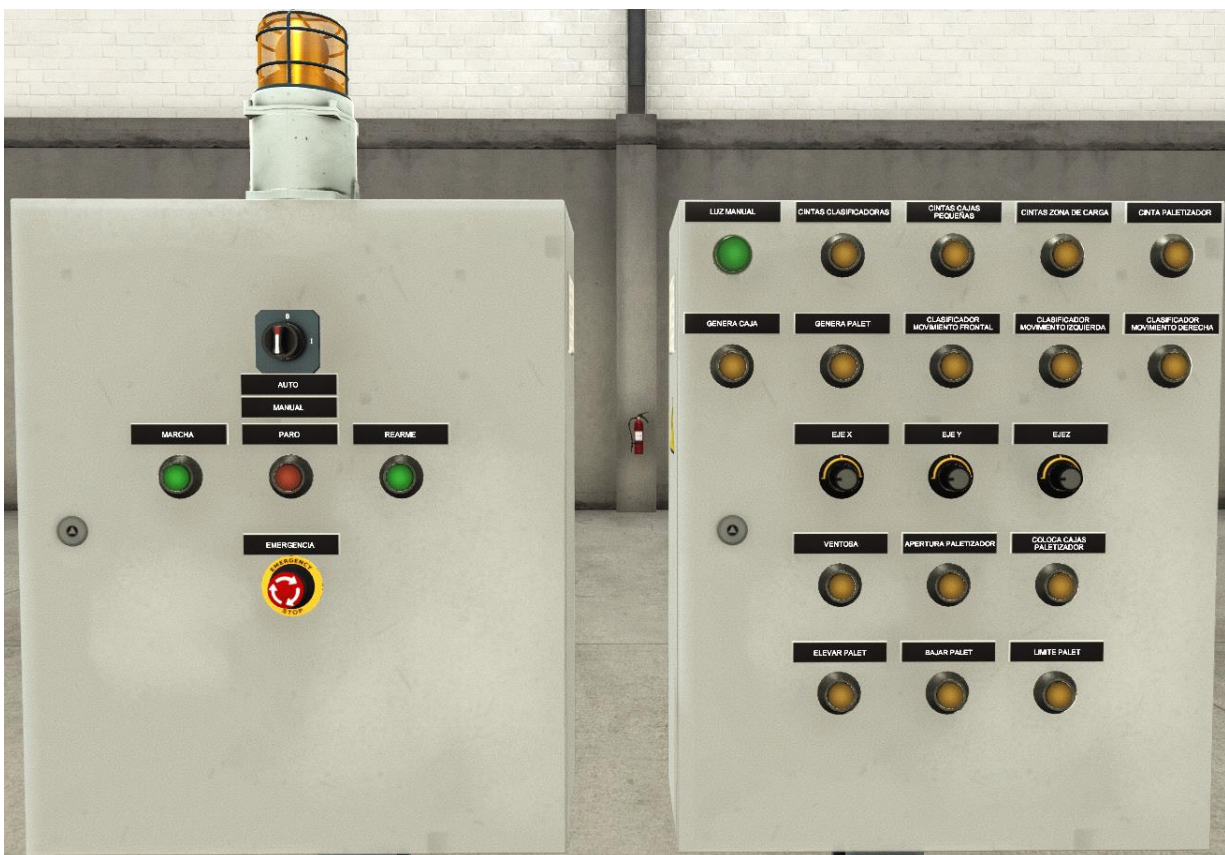


Figura 4.3 Paneles de control

La funcionalidad de cada uno de estos componentes es la siguiente:

#### 4.2.1.1 Panel principal

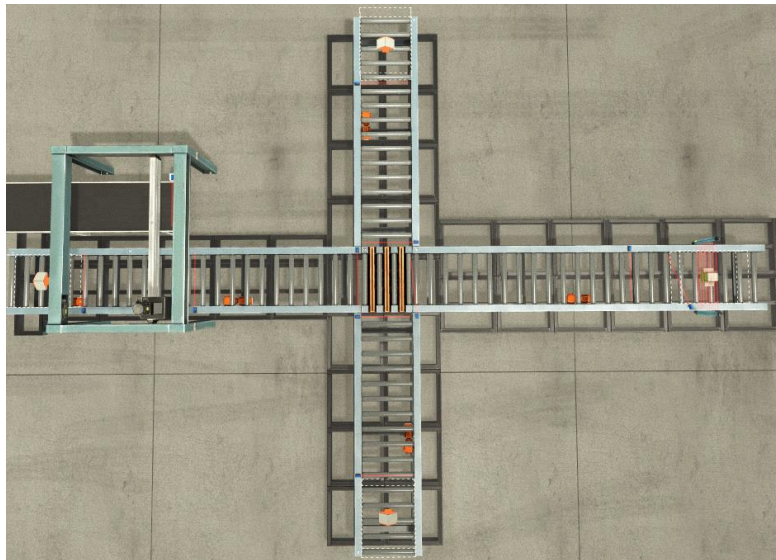
- **Luz de emergencia:** se activa al pulsar la seta de emergencia y se desactiva cuando se haya salido del estado de emergencia.
- **Seleccionador de modo:** permite seleccionar el modo de funcionamiento del sistema, automático o manual.
- **Marcha:** inicia el funcionamiento del sistema en modo automático y reanuda el funcionamiento después de haber solicitado una parada.
- **Paro:** solicita una parada del funcionamiento de las máquinas a final de ciclo.



- **Rearme:** botón que permite salir del estado emergencia.
- **Emergencia:** seta de emergencia que detiene instantáneamente todo el sistema.

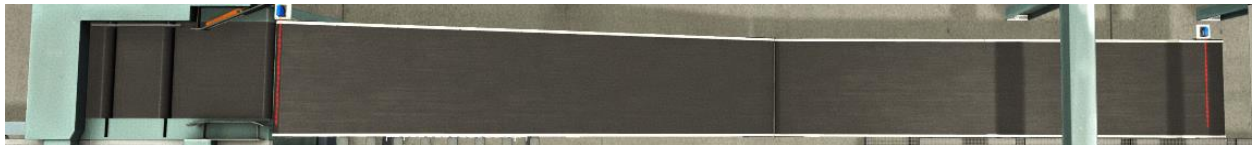
#### 4.2.1.2 Panel secundario

- **Cintas clasificadoras:** se corresponde con cintas transportadoras del comienzo del sistema.



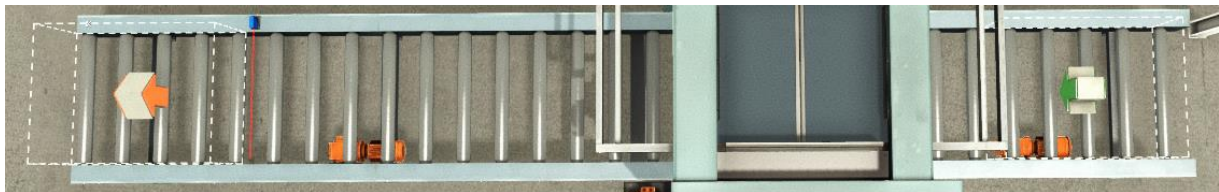
**Figura 4.4** Cintas clasificadoras

- **Cintas cajas pequeñas:** cintas que transportan las cajas pequeñas a la paletizadora.



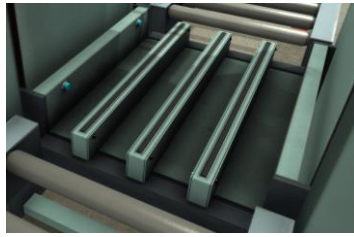
**Figura 4.5** Cintas cajas pequeñas

- **Cintas zona de descarga:** cintas situadas al final del sistema.



**Figura 4.6** Cintas zona de descarga

- **Cinta paletizador:** cinta inferior del paletizador.



**Figura 4.7** Cinta paletizador

- **Genera caja:** genera en la cinta inicial una caja sobre un palet.
- **Genera palet:** genera un palet vacío para el paletizador.
- **Clasificador movimiento frontal/izquierda/derecha:** activa las distintas cintas del clasificador.
- **Eje x, eje y, eje z:** potenciómetros que permiten modificar la posición del brazo del 'Pick & Place'.
- **Ventosa:** gira y activa la ventosa.
- **Apertura paletizador:** abre la compuerta del paletizador para colocar las cajas sobre el palet.
- **Coloca cajas paletizador:** activa dos actuadores del paletizador: un empujador que desplaza las cajas paletizadoras hasta la compuerta, y un colocador que posiciona las cajas en el lugar adecuado para cuando se abra la compuerta.
- **Elevar palet:** permite subir el palet del paletizador progresivamente.
- **Bajar palet:** permite bajar el palet del paletizador progresivamente
- **Límite palet:** si se pulsa este botón junto a alguno de los dos botones anteriores, el desplazamiento será directamente hasta el extremo.

## 4.2.2 Modos de funcionamiento

Como se ha podido comprobar en el panel de control principal, se dispone de dos modos de funcionamiento, manual y automático.

### 4.2.2.1 Automático

Si al iniciar el sistema está seleccionado el modo auto y se pulsa el botón marcha, el sistema comienza a funcionar automáticamente, clasificando las cajas según se vaya liberando espacio. El diseño se ha realizado para que trabaje con el máximo número de cajas posibles.

Se ha creado un vector de 4 componentes donde se guarda la información asociada a la altura de las cajas entrantes, y que permite llevar a cabo la clasificación de las mismas. El valor de las componentes se actualiza cuando se activa el sensor de entrada de alguna de las cintas de clasificación.

Por este motivo, es importante que no haya más de 4 cajas sin clasificar, ya que esto supondría que no se guarda información de la altura de las últimas cajas y por lo tanto la clasificación comenzaría a ser incorrecta. Para evitar que esto ocurra, se ha colocado un sensor al comienzo de la cinta de inicio que controla la generación de material.

Se ha definido un *tiempo de espera* para cuando lleguen 2 o más cajas paletizadoras de forma consecutiva. Este tiempo comienza cuando el palet de la primera caja se encuentra en su totalidad en la cinta de clasificación, y termina cuando el brazo se eleva verticalmente teniendo sujeta la caja. Durante este tiempo, el segundo conjunto de palet y caja paletizadora estará esperando en el clasificador.

Esta espera es importante para asegurar que las cajas paletizadoras son tratadas correctamente



independientemente del orden de llegada.

#### 4.2.2.2 Manual

Si el modo manual está seleccionado al iniciar el sistema, el indicador del panel secundario se encenderá y el sistema quedará a la espera de que el operario pulse cualquier botón.

#### 4.2.2.3 Transición

El paso del modo automático a modo manual es directo, no requiere de ninguna transición. Con lo cual, es importante tener en cuenta que al pasar a modo manual, el brazo robótico de la estación 'Pick & Place' desactivará la ventosa y se moverá a la posición indicada en el panel de control.

Para pasar de modo manual a modo automático el sistema no puede hacerlo directamente, ya que el modo automático debe de iniciarse en unas condiciones conocidas, en este caso, el estado inicial.

El operario debe limpiar el sistema manualmente, retirando las cajas y los palets presentes en él. Una vez terminado, debe pulsar el botón de rearme, que se iluminó al cambiar de modo, indicando al sistema que la limpieza ha sido realizada.

Sin embargo, es posible que el sistema esté en su estado inicial (sin elementos), pero no en su posición inicial. Debido a las limitaciones del operario en esta situación, antes de comenzar con la limpieza, el sistema bajará automáticamente el elevador del paletizador (si está elevado) y moverá el brazo de la estación 'Pick & Place' a la posición  $x=0$ ,  $y=0$ ,  $z=0$  para colocarse en la posición inicial. De esta manera, el sistema estará listo para la limpieza en la posición inicial, y el operario podrá llevarlo al estado inicial.

### 4.2.3 Estado de emergencia

Al pulsar la seta de emergencia se pasará al estado de emergencia, en él se detendrá todo el sistema y se bloquearán los paneles de control a la espera de que se suelte la seta de emergencia. Cuando esto ocurra, al igual que en la transición del modo manual al modo automático, el sistema reiniciará automáticamente la posición del paletizador y de la estación y quedará a la espera de que el operario limpie el sistema y pulse el botón de rearme.

## 4.3 DESARROLLO SOFTWARE

### 4.3.1 Introducción

La programación principal se ha realizado en lenguaje SFC, que está basado en las Redes de Petri (RDP) y es un grafo orientado, que se utiliza para modelar el comportamiento dinámico de sistemas de eventos discretos. Una de las ventajas de SFC sobre otros lenguajes de programación es su capacidad para describir gráficamente, lo que permite trabajar con automatismos complejos y procesos en paralelo, así como facilitar la realización de modificaciones.

Por otro lado, también se ha hecho uso del lenguaje ST, que es un lenguaje de alto nivel similar a C, Matlab o Java. ST es adecuado para gestionar información o datos que no son booleanos, como números enteros (para contar), números reales, vectores y matrices (para gestión de stock, por ejemplo) y cadenas de caracteres.

Se utilizan sentencias de ST en RDP para la gestión de variables no booleanas y expresiones booleanas para la condición lógica de transición. En SFC, las transiciones permiten la introducción directa de expresiones en ST, y las acciones se pueden agregar a los lugares para ejecutar otros programas.

Por último, para realizar funciones más directas que requieren menor complejidad que las mencionadas con anterioridad, se ha utilizado el lenguaje LD, que permite activar directamente las bobinas correspondientes a los actuadores del sistema. También es un lenguaje basado en la representación gráfica, lo que facilita su entendimiento y la detección de errores.

## 4.3.2 FACTORY IO

Para comenzar el desarrollo software hay que entender las siguientes funcionalidades de FACTORY.

### 4.3.2.1 Emisor

El sistema comienza por los ‘*Emmitter*’, componente que se encarga de generar elementos en el sistema.

Algunos componentes pueden ser configurados para que tengan un funcionamiento automático, controlado por FACTORY y con ciertas características, ‘*User-defined*’, o para que sea controlado desde un programa externo, ‘*Controller-defined*’.

Resulta de mayor interés controlar su funcionalidad desde CODESYS, indicando el momento y el elemento que se desea generar, por ello se selecciona ‘*Controller-defined*’.

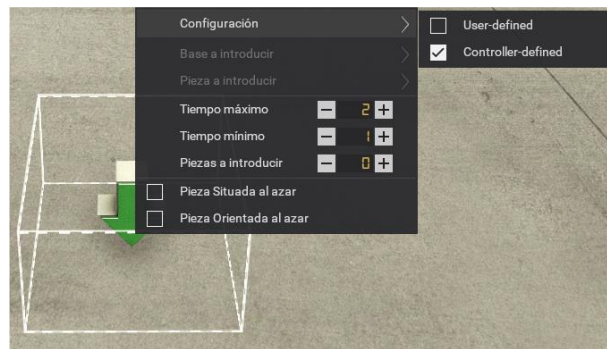


Figura 4.8 Configuración del emisor

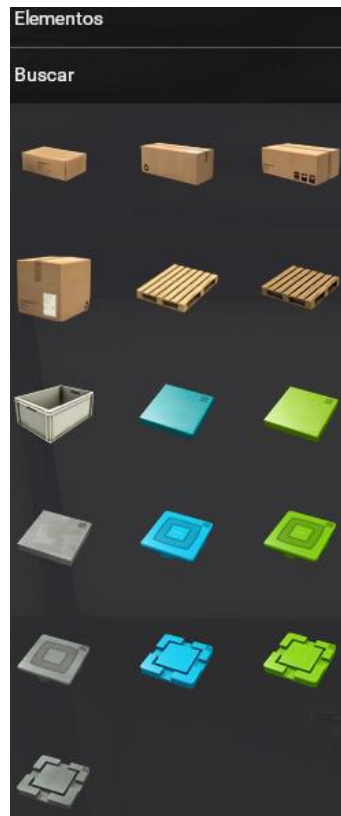
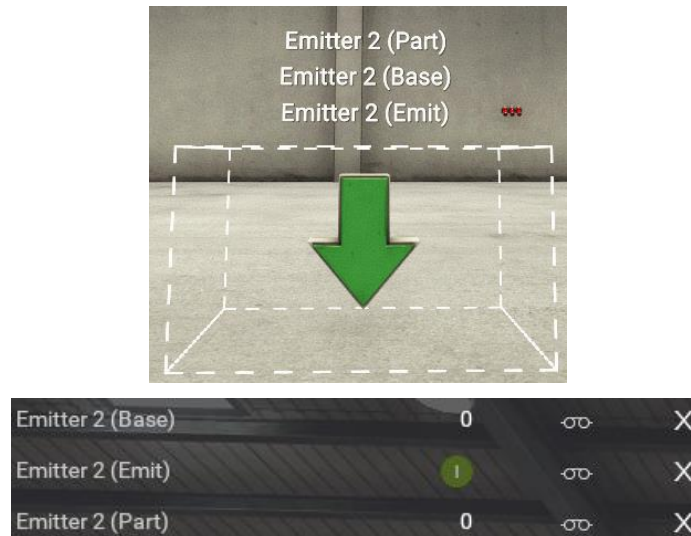


Figura 4.9 Elementos de FACTORY IO

Con esta configuración seleccionada, estarán accesibles 3 nuevas señales, que, al comprobar su naturaleza, se puede observar como las 2 primeras se corresponden con números enteros y la última con una variable booleana.



**Figura 4.10** Emisor como ‘*Controller-defined*’

Estas señales permiten generar cualquiera de los elementos disponibles en FACTORY. Para ello se introduce el identificador característico del elemento que se desea generar y se activa la variable booleana que actúa como emisor. Los números enteros que se pueden introducir en ‘*Base*’ o en ‘*Part*’ correspondientes a los elementos con los que se trabaja en este sistema son los siguientes:

**Tabla 4–1.** Identificadores ‘*Part*’

Elemento	Identificador
Caja (S)	1
Caja (M)	2
Caja (L)	4
Caja Paletizadora	8

**Tabla 4–2.** Identificadores ‘*Base*’

Elemento	Identificador
Palet	1
Palet cuadrado	2

Se puede comprobar como los identificadores de cada elemento son potencias de 2. El resto de valores numéricos generan elementos con una aleatoriedad que el programador no puede controlar.

### 4.3.2.2 Eliminador

El ‘*Remover*’ sigue la misma dinámica que el ‘*Emmitter*’, la diferencia es que estos componentes se van a configurar como ‘*User-defined*’ para que elimine todo lo que caiga en él. Si se optara por controlar desde CODESYS, habría que tener en cuenta el tipo de conjunto que cae por cada rampa en modo automático, cuando el sistema se esté limpiando, y por último, el caso que añade más complejidad, en modo manual.

Por lo tanto, no resulta de gran interés añadir dicha complejidad a estos componentes teniendo en cuenta su funcionalidad en el sistema.



Figura 4.11 Eliminador

### 4.3.2.3 Sensores

En el sistema se implementan sensores de presencia, que detectan cuando algún elemento se ha colocado en su rango de lectura.

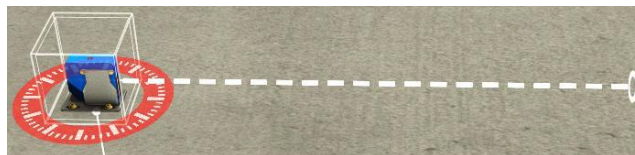


Figure 4.12 Sensor de presencia

### 4.3.2.4 Emisor y receptor de matriz luminosa

El emisor consta de una serie de sensores que permiten determinar la altura de uno o varios objetos. El receptor permite identificar dicha altura en valor digital, tratando con cada uno de los sensores por separado, o en valor analógico, obteniendo un único valor.

Como el resto de sensores y señales son mayoritariamente booleanas, el receptor se configura en analógico.

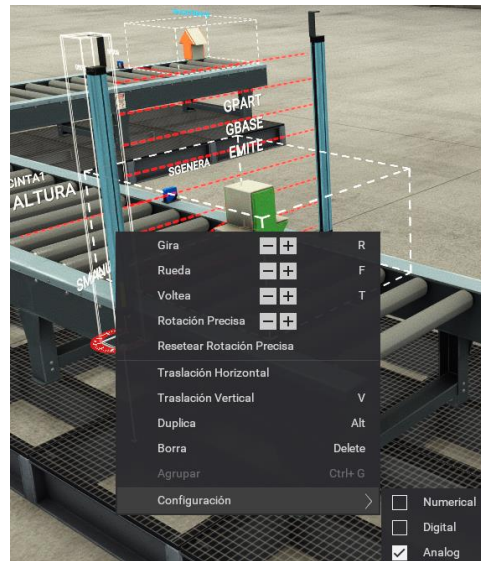


Figura 4.13 Emisor y receptor de matriz luminosa

#### 4.3.2.5 Transferencia de cadena

Para la clasificación de las cajas se dispone de este dispositivo que está formado por 4 actuadores que permiten mover el objeto colocado encima en 4 direcciones, facilitando así la distribución de los palets. Dos de ellos están asociados a una cinta transportadora y otros dos a unas cadenas de transporte.

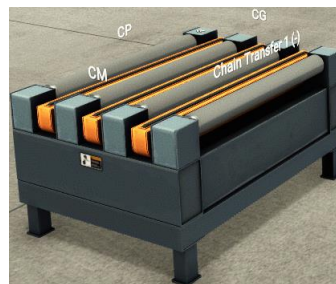


Figura 4.14 Transferencia de cadena

#### 4.3.2.6 Pick & Place

Este componente, que cuenta con un gran número de señales, dispone de un brazo robótico móvil en los 3 ejes de coordenadas. Para realizar movimientos precisos, cada eje de coordenadas tiene asociado 2 señales: la posición a la que se desea mover el brazo y la posición actual del mismo en todo momento.

Los ejes pueden ser configurados para trabajar con valores analógicos o digitales. La principal ventaja del valor analógico frente al valor digital en el movimiento del brazo es la precisión, ya que permite indicar el punto exacto al que se desea mover.

El extremo del brazo robótico está equipado con 4 señales adicionales: un detector de proximidad, que indica cuando se encuentra en cercanía a un objeto, una señal de activación de la ventosa, otra señal que permite la rotación de la misma y una cuarta señal que no se utilizará.

Debido a la posición relativa entre la ventosa y las cajas paletizadoras, la rotación de la ventosa es muy importante en este sistema, ya que logrará un agarre más seguro y firme al cubrir una mayor área de superficie de la caja.



Figura 4.15 Pick & Place

**4.3.2.7 Paletizador**

Es el componente que dispone del mayor número de señales, teniendo todas una configuración fija de valores digitales. Empezando por la parte inferior, cuenta con unas cadenas de transporte que permiten el movimiento en ambas direcciones, y sensores en los extremos de las mismas para controlar la posición del palet que estén desplazando.

Además, dispone de 4 señales adicionales relacionadas con la posición del palet: la primera se activa mientras el elevador esté en movimiento, la segunda permite realizar movimientos del elevador al extremo (superior o inferior) y las dos últimas realizan pequeños incrementos de la posición. Combinando la segunda con las dos últimas, el elevador puede realizar un movimiento completo de extremo a extremo.

En la parte superior, el paletizador cuenta con una cinta transportadora, un colocador que fija la posición de las cajas y una compuerta que, al abrirse, permite depositar las cajas en el palet. También dispone de una aleta para girar las cajas y un empujador que las desplaza hasta el colocador.



Figura 4.16 Paletizador



#### 4.3.2.8 Botones

Los botones de FACTORY son en su mayoría normalmente abiertos y pasan al estado normalmente cerrados cuando se pulsan, sin embargo, con el botón de paro y la seta de emergencia ocurre al contrario, en estado de reposo se encuentran normalmente cerrados y cuando se pulsan pasan a normalmente abiertos.



Figure 4.17 Botones normalmente cerrados

#### 4.3.2.9 Cintas transportadoras

Las cintas transportadoras se pueden configurar para que sean analógicas o digitales y pueden girar en ambos sentidos. En los sistemas desarrollados en este proyecto estarán todas configuradas como digitales y sus movimientos serán unidireccionales.



Figure 4.18 Cintas transportadoras

### 4.3.3 CODESYS [5] [6] [7] [8]

Siguiendo el principio de modularización, se ha dividido el control en distintos subsistemas que se han agrupado en 5 carpetas que se explicarán por separado.

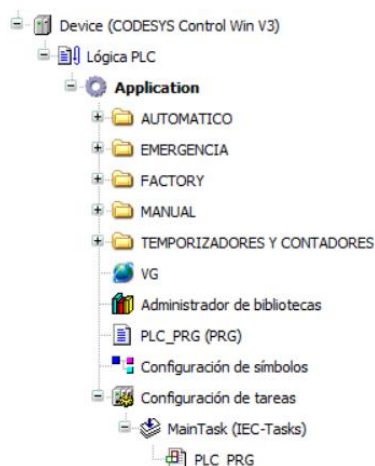


Figura 4.19 Organización del proyecto

### 4.3.3.1 AUTOMATICO

Contiene la mayor parte de la programación, ya que no solo incluye el modo automático, sino también la gestión de los modos de funcionamiento. A su vez está dividida en subcarpetas que organizan el contenido y lo hacen más accesible:

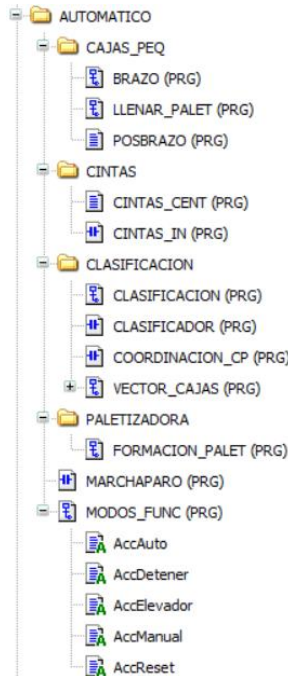


Figura 4.20 Carpeta AUTOMATICO

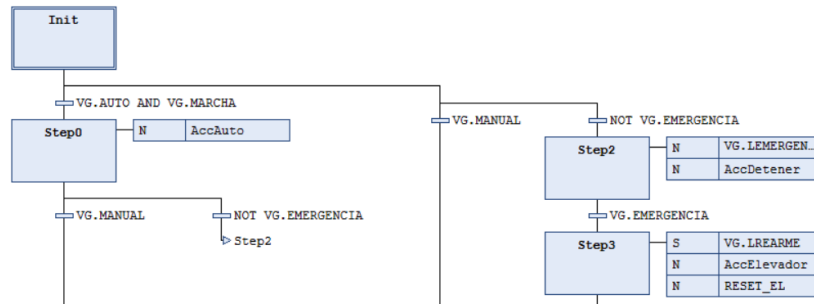
Entre los distintos programas destacar MarchaParo, que se encarga de gestionar la interacción del usuario con los botones indicados en su propio nombre. Asocia las pulsaciones a unas variables internas que se activan o desactivan y que permiten identificar en todo momento el estado en el que se encuentra el sistema. Por ejemplo, el estado de marcha se configura de la siguiente forma:



Figura 4.21 Programación en LD

También destacar ModosFunc, que llama a las acciones correspondientes al modo de funcionamiento presente en el sistema. Aquí se puede ver parte del código donde se aprecia el trato diferente con el botón (la seta) de emergencia, que a diferencia de los botones AUTO y MANUAL, está precedido por un NOT:





**Figura 4.22** Modos de funcionamiento

Por último, hacer mención al programa POSBRAZO donde se definen los puntos a los que se desplazará el extremo del robot de la estación ‘Pick & Place’.

```

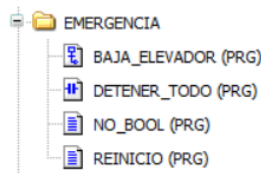
1  IF BRAZO.PINICIAL THEN
2      VG.POSDX:=REAL#2.3;
3      VG.POSDY:=REAL#1.9;
4  END_IF
5
6  IF BRAZO.MOVZIN THEN
7      VG.POSDZ:=REAL#7.6;
8  END_IF
9
10 IF BRAZO.RESETZ THEN
11     VG.POSDZ:=REAL#0;
12
13 END_IF
14
15 IF BRAZO.PFINAL THEN
16     VG.POSDX:=REAL#9;
17 END_IF
18
19 IF BRAZO.MOVZFIN THEN
20     VG.POSDZ:=REAL#7.3;
21 END_IF

```

**Figura 4.23** Programación ST

### 4.3.3.2 EMERGENCIA

Programas encargados de detener el sistema cuando se pulse la seta de emergencia, y de dejar el sistema en la posición inicial cuando se suelte la seta de emergencia, faltando la limpieza por parte del operario para dejar al sistema en estado inicial.



**Figura 4.24** Carpeta EMERGENCIA

Para detener la estación ‘Pick & Place’, se han asignado a las señales de posición deseada del robot, la posición actual de los ejes del brazo, de esta forma se le estará indicando que se desea mover a su posición actual y se detendrá. Esto se hace en el archivo NO\_BOOL.

```

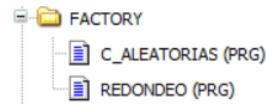
VG.POSDX:=VG.POSRX;
VG.POSDY:=VG.POSRY;
VG.POSDZ:=VG.POSRZ;

```

**Figura 4.25** Detención de la estación ‘Pick & Place’

### 4.3.3.3 FACTORY

Contiene la parte de la programación relacionada directamente con FACTORY.



**Figura 4.26** Carpeta FACTORY

Dado que CODESYS V3.5 no tiene ninguna función que genere números aleatorios, para generar las cajas se ha creado un contador de 15 unidades, donde a cada unidad se le ha asignado una caja teniendo en cuenta distintas consideraciones.

```

IF (CONT MOD 2 = 0) THEN
  VG.GPART:=8;
ELSIF (CONT MOD 3 = 0) THEN
  VG.GPART:=4;
ELSIF CONT<8 THEN
  VG.GPART:=1;
ELSE
  VG.GPART:=2;
END_IF
  
```

**Figura 4.27** Contador para generar cajas

Concretamente, el tamaño de las cajas asociado a cada valor del contador son los siguientes:

**Tabla 4-3.** Cajas generadas

Valor del contador	Tamaño
1	S
2	Paletizadora
3	L
4	Paletizadora
5	S
6	Paletizadora
7	S
8	Paletizadora
9	L
10	Paletizadora
11	M
12	Paletizadora
13	M
14	Paletizadora
15	L

Es importante tener en cuenta que la altura de las cajas de tamaño S y M es la misma, por lo tanto, ambas serán clasificadas por la misma cinta transportadora.

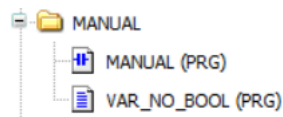
Otro aspecto peculiar de FACTORY es que cuando el brazo robótico alcanza un punto, el valor real de este punto difiere en las milésimas respecto al punto deseado. Esto no supone ningún problema en la simulación, pero sí en la programación, ya que hay transiciones en los programas con la condición posición real = posición deseada, y al no ser exactamente iguales, no cumple la condición. Por lo tanto, se ha creado un programa llamado REDONDEO donde a esa igualdad se incluye una tolerancia de  $\pm 0.1$ , considerando así que el punto deseado y el real es el mismo.

```
IF (VG.POSRX>VG.POSDX-REAL#0.1) AND (VG.POSRX<VG.POSDX+REAL#0.1) THEN
  XIGUAL:=TRUE;
ELSE
  XIGUAL:=FALSE;
END_IF
```

**Figura 4.28** Aproximación de los puntos del eje x del 'Pick & Place'

#### 4.3.3.4 MANUAL

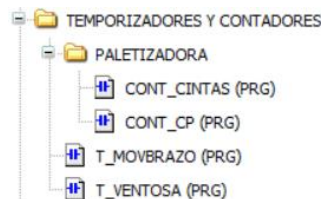
Contiene los programas que tienen como entrada los botones y actúa directamente sobre los actuadores. Para programar la funcionalidad de los botones es muy útil el lenguaje ST, pero para los potenciómetros es más recomendable ST.



**Figura 4.29** Carpeta MANUAL

#### 4.3.3.5 TEMPORIZADORES Y CONTADORES

En esta última carpeta se encuentra el código correspondiente al conteo de palets y cajas que permite reanudar el funcionamiento de las cintas transportadoras tras un estado de paro, y los temporizadores asociados al brazo robótico.



**Figura 4.30** Carpeta TEMPORIZADORES Y CONTADORES

Si un palet con una caja se encuentra en mitad de una cinta, no hay sensor que pueda detectar su presencia. La solución planteada es llevar la cuenta de los elementos que entran y salen de las cintas usando contadores ascendentes y descendentes, CTUD. Cuando su valor sea mayor de 1, activarán su salida que estarán directamente conectadas a los actuadores de las cintas. El código CONT\_CINTAS se corresponde a las cintas donde circulan las cajas pequeñas camino al paletizador:

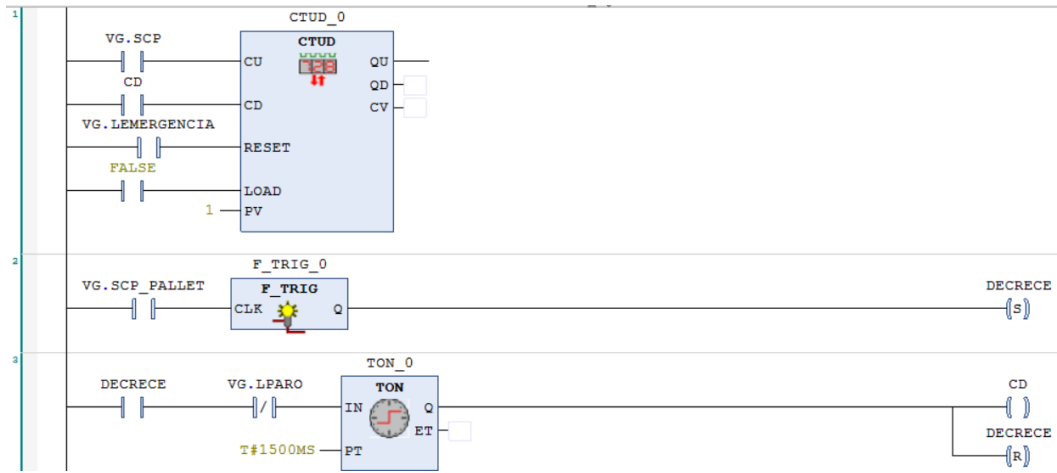


Figura 4.31 CTUD asociado a las cintas dirigidas al paletizador

## 4.4 RESULTADO

Para demostrar el correcto funcionamiento del sistema, se ha grabado una demostración que se encuentra disponible en el siguiente [enlace](#).

# 5 FORMADOR Y CLASIFICADOR DE CONJUNTOS

## 5.1 DESCRIPCIÓN

Este sistema consiste en un proceso de mecanizado y ensamblaje automatizado para el tratamiento de piezas, que garantiza la precisión y la velocidad, lo que se traduce en una mayor productividad y calidad en la producción de conjuntos.

La materia prima es la entrada del sistema, y se transporta a unos centros de mecanizado donde se procesa para crear bases y tapas. Las piezas resultantes se trasladan a unos posicionadores donde un brazo robótico las ensambla en conjuntos, colocando la tapa sobre la base. Una vez que el conjunto se ha formado, se transporta y clasifica según su color mediante brazos pivotantes. El desplazamiento de las piezas se realiza a través de cintas transportadoras.

La principal diferencia respecto al sistema anterior, es que ahora se cuenta con un modo de reinicio que limpiará automáticamente el sistema al salir del estado de emergencia o para pasar de modo manual a modo automático. Así se elimina la responsabilidad del operario de retirar manualmente las piezas y se dejará directamente el sistema en su posición y estado inicial.



Figura 5.1 Vista general del sistema

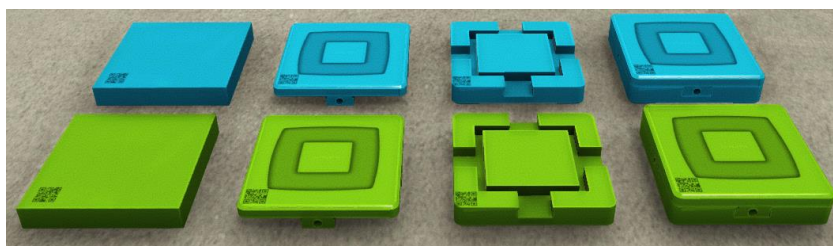


Figura 5.2 Elementos de FACTOY IO: materias primas, tapas, bases y conjuntos

## 5.2 PRESTACIONES

### 5.2.1 Panel de control

El sistema dispone de un panel de control desde el cual monitorizar su funcionamiento.



Figure 5.3 Panel de control

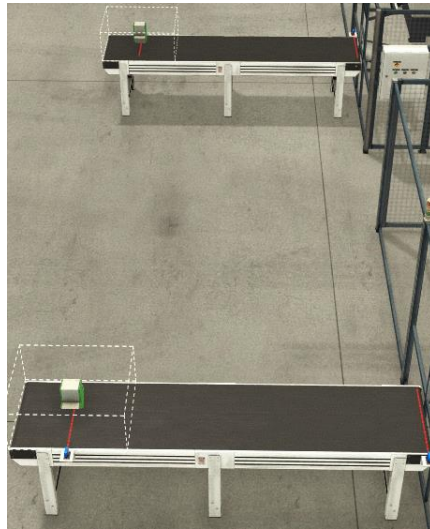
Se procede a explicar las funcionalidades de las componentes nuevas respecto al sistema anterior:

- **Contador de conjuntos:** indica el número de conjuntos que han sido clasificados en modo automático. Se reinicia después de un estado de emergencia.

Los siguientes botones se corresponden con el modo manual:

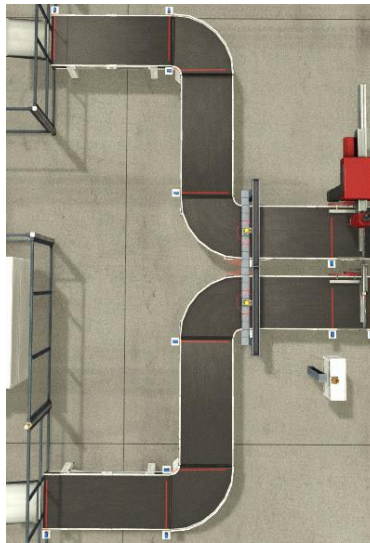
- **Genera material base/tapa:** genera materia prima para base o para tapa respectivamente.
- **Cintas transportadoras 1:** activa las cintas transportadoras iniciales, encargadas de llevar las materias primas a los centros de mecanizado.





**Figura 5.4** Cintas transportadoras iniciales

- **Cintas transportadoras 2:** activa las cintas transportadoras centrales, encargadas de llevar las piezas procesada hasta los posicionadores.



**Figura 5.5** Cintas transportadoras centrales

- **Cierre posicionadores:** cierra los posicionadores, colocando las piezas en la posición correcta para el brazo robótico.
- **Elevador posicionadores:** levanta los posicionadores para desechar piezas o para clasificar conjuntos.
- **Extensión horizontal/vertical brazo robótico:** movimientos asociados al brazo robótico.
- **Activa ventosa:** activa la ventosa del brazo robótico.
- **Cintas conjuntos:** activa las cintas finales del sistema, encargadas de transportar los conjuntos.
- **Brazos pivotantes:** activa los brazos pivotantes y sus respectivas cintas transportadoras.
- **Reinicio contadores:** da por finalizado el modo de limpieza automático. Se entenderá mejor su funcionamiento en la explicación de software (apartado 5.2.2.2).

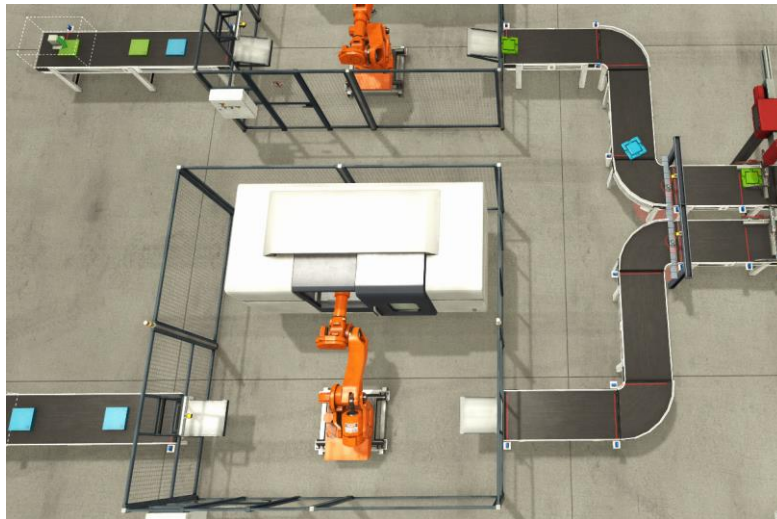
## 5.2.2 Modos de funcionamiento

Al igual que en el sistema anterior y como se ha podido comprobar en el panel de control principal, se dispone de dos modos de funcionamiento, manual y automático.

### 5.2.2.1 Automático

Si al iniciar el sistema está seleccionado el modo auto y se pulsa el botón marcha, el sistema comienza a funcionar automáticamente, procesando las piezas según se vaya liberando espacio. El diseño se ha realizado para que trabaje con el máximo número de piezas posibles.

Se han definido tanto para las tapas como para las bases los llamados '*puntos de saturación*', que se corresponden con 3 puntos de las cintas centrales que cuando están ocupados detienen la parte del sistema saturada. En la siguiente imagen se puede observar cómo la parte superior del sistema, correspondiente a las bases, está detenida a la espera de la liberación del espacio situado debajo del brazo robótico:



**Figura 5.6** Puntos de saturación

Como el tiempo de procesamiento para ambas piezas es muy similar, si el tiempo de simulación es reducido, esos puntos de saturación no serán necesarios. Sin embargo, para tiempos de simulación elevados, esa pequeña diferencia temporal (mayor tiempo de procesamiento para las tapas), puede dar lugar a una situación en la que sí sean necesarios.

Los conjuntos completamente azules serán clasificados por la primera rampa, los verdes por la segunda rampa y los que sean una mezcla de colores se deslizarán por la última rampa. Cada vez que esto ocurra, el contador de conjuntos aumentará su valor en una unidad.

### 5.2.2.2 Manual

Si al iniciar el sistema está seleccionado el modo manual, el sistema comienza a funcionar directamente en este modo, quedándose a la espera de que el operario pulse algún botón.

Desde el panel de control se podrá comprobar el correcto funcionamiento de cada parte del sistema. Cabe destacar que los centros de mecanizado tienen su propio panel de control con botones de start, stop y reset, de colores verde, rojo y naranja respectivamente. También disponen de seta de emergencia.

Dada la libertad de la que dispone el operario en este modo de funcionamiento, el contador de conjuntos no verá modificado su valor.



**Importante:** el modo manual está destinado a hacer pruebas y comprobar el correcto funcionamiento de los distintos componentes del sistema. El modo de reinicio se ha optimizado teniendo en consideración que el operario no forma conjuntos, no retira piezas del sistema manualmente, y no mueve tapas a las cintas de las bases o viceversa (lo mismo con la materia prima).

No obstante, si por error humano o por necesidad del operario se incumplen algunas de estas restricciones, se dispone del botón Reinicio Contadores, que dará por finalizada la limpieza automática. Es **importante** después de que se realice esta acción, asegurarse de que el sistema quede limpio. En caso de no estarlo, antes de proceder, será responsabilidad del operario quitar manualmente las piezas restantes.

Para que el operario corrobore que el sistema está limpio deberá pulsar el botón de rearme (que se iluminará a modo de indicación) y una vez pulsado, el sistema ya estará disponible para funcionar con normalidad.

El botón Reinicio Contadores, dando por hecho que se va a incumplir alguna de las restricciones, puede ser pulsado en las siguientes situaciones:

- Tras haber terminado de realizar las comprobaciones pertinentes en modo manual.
- Tras haber pasado de modo manual a modo automático. Implica un error humano, ya que al incumplir las restricciones, antes de pasar a modo automático debería de haber pulsado Reinicio Contadores.
- Tras haber pasado de modo manual a estado de emergencia.

En los dos últimos casos se detectará fácilmente la necesidad de pulsar el botón, ya que, el sistema dará por finalizada la limpieza, pero no la habrá realizado correctamente, o el modo de limpieza no se detendrá a la espera de que salgan del sistema piezas que han sido quitadas previamente al incumplir las restricciones.

### 5.2.2.3 Transición

El paso del modo automático a modo manual es directo, no requiere de ninguna transición. El único aspecto a tener en cuenta es que, si los centros de mecanizado estaban procesando materia prima, aunque se haya pasado a modo manual, terminará de procesarse.

Sin embargo, para pasar de modo manual a modo automático el sistema no puede hacerlo directamente, ya que el modo automático debe de iniciarse en unas condiciones conocidas. Por lo tanto, hay que llevar el sistema al estado de inicial.

Para alcanzar esas condiciones de inicio del modo automático, se hace uso del reinicio implementado en el estado de emergencia. De esta forma, el sistema vuelve al estado original y se queda a la espera de que se pulse el botón de marcha para empezar a funcionar en modo automático.

## 5.2.3 Estado de emergencia

Cuando se pulse la seta de emergencia se pasará al estado de emergencia, en él se detendrá todo el sistema inmediatamente a excepción del brazo robótico, que si estaba realizando un movimiento lo terminará antes de detenerse. Se trata de una limitación de FACTORY.

Mientras que los brazos robóticos de los centros de mecanizado si están preparados para detenerse inmediatamente, este brazo robótico está diseñado para extenderse o retraerse, así que no es posible detenerlo en una posición intermedia.

En este estado, quedará bloqueado todo el panel de control a la espera de que se suelte la seta de emergencia. Cuando esto ocurra, se ejecutará un modo de limpieza automática que se encargará de ir limpiando el sistema desde el final, en caso de que haya algún conjunto sin clasificar, hasta el comienzo.

El modo de limpieza desechará todas las piezas que haya en el sistema. Los conjuntos y las bases se enviarán por la última rampa y las tapas se enviarán por la rampa situada tras el posicionador de tapas.

Cuando ya no haya piezas en el sistema, se apagará la luz de emergencia, indicando que el sistema está en

reposito, y se encenderá la luz correspondiente al botón rearme, indicando que el sistema está limpio y recuperado del sistema de emergencia. El operario deberá corroborar que el sistema está limpio y pulsar el botón de rearme, en ese momento el sistema estará listo para reanudar su funcionamiento con normalidad.

## 5.3 DESARROLLO SOFTWARE

### 5.3.1 FACTORY IO

A continuación, se explicarán los componentes usados en la visualización que no han sido implementados en el sistema anterior.

#### 5.3.1.1 Emisor

En este sistema se hace uso de nuevos elementos a los que les corresponden los siguientes identificadores:

**Tabla 5-1.** Identificadores ‘Part’

Elemento	Identificador
Materia prima azul	16
Materia prima verde	32

#### 5.3.1.2 Centros de mecanizado

En los paneles de control de los centros de mecanizado, se pueden controlar los botones de start, stop y reset pero no las setas de emergencias. Funcionan, pero no tienen una señal asociada que pueda ser controlada externamente.

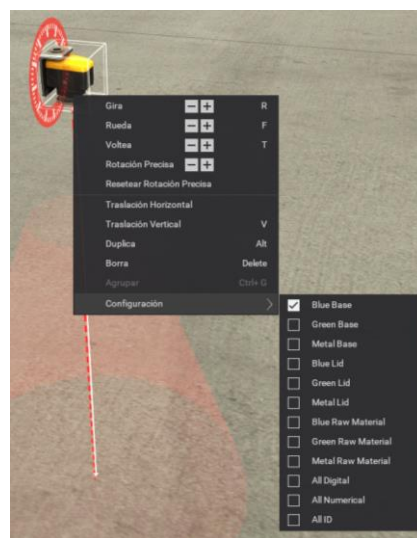
Disponen de 5 variables, de las cuales 2 resultan de interés: la señal que indica que el centro de mecanizado está operando, y la que es necesario activar si se desea producir una tapa. Por defecto, el centro de mecanizado procesa la materia prima convirtiéndola en una base.



**Figure 5.7** Centro de mecanizado

### 5.3.1.3 Sensores

En el sistema se implementan unos sensores de visión que permiten identificar un tipo de pieza concreta, y serán usados para detectar el color de las piezas para su posterior clasificación.



**Figure 5.8** Sensor de visión

### 5.3.1.4 Brazo robótico

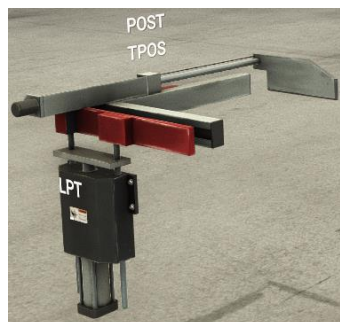
Los movimientos deseados de este componente son extensión horizontal (eje x) y extensión vertical (eje z). En cuanto al resto de señales, será de utilidad el sensor colocado en su extremo, que indicará si la ventosa se encuentra encima de una pieza, lo que permitirá controlar la ventosa para activarla o desactivarla.



**Figure 5.9** Brazo robótico

### 5.3.1.5 Posicionadores

Permiten colocar las piezas en una posición adecuada para que el brazo robótico las agarre en el centro. Para ello dispone de dos señales, una de ellas se encarga de accionar el posicionador para posicionar la pieza y la otra detecta cuando la pieza ha sido posicionada. Dispone de otra señal que eleva el posicionador, permitiendo el paso de las piezas por debajo de él.



**Figure 5.10** Posicionador

### 5.3.1.6 Brazos pivotantes

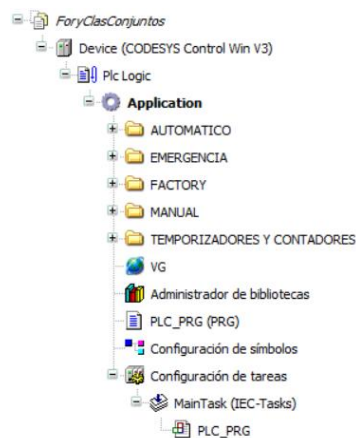
Los brazos pivotantes tienen una señal que les permite girar (ser accionados), y la cinta transportadora de la que disponen, puede girar en ambos sentidos, dependerá de la variable activada.



**Figure 5.11** Cinta transportadora y brazo pivotante

### 5.3.2 CODESYS

Siguiendo la misma dinámica que la programación del sistema anterior, se ha dividido el control en distintos subsistemas que se han agrupado en 5 carpetas:



**Figura 5.12** Organización del proyecto

Se procede a explicar cada una de ellas por separado.

5.3.2.1 AUTOMATICO

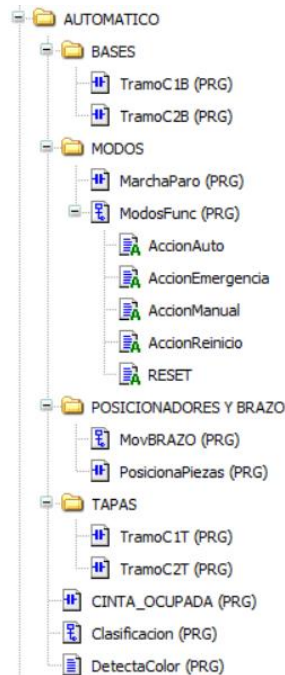


Figura 5.13 Carpeta AUTOMATICO

Entre los distintos programas destacar ModosFunc, donde se aprecia el trato diferente no solo con el con la seta de emergencia sino también con el botón que reinicia los contadores (BRC):

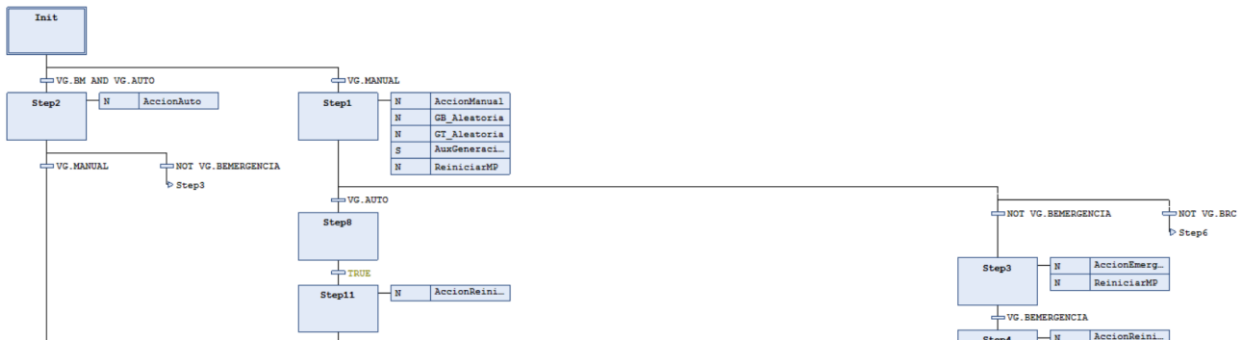
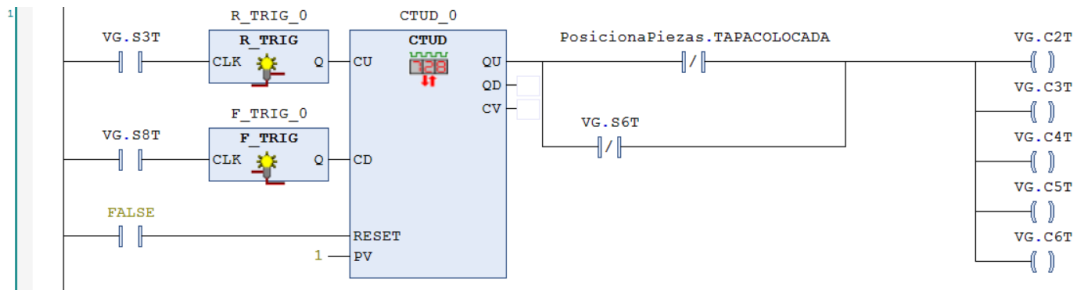


Figura 5.14 Programación en SFC

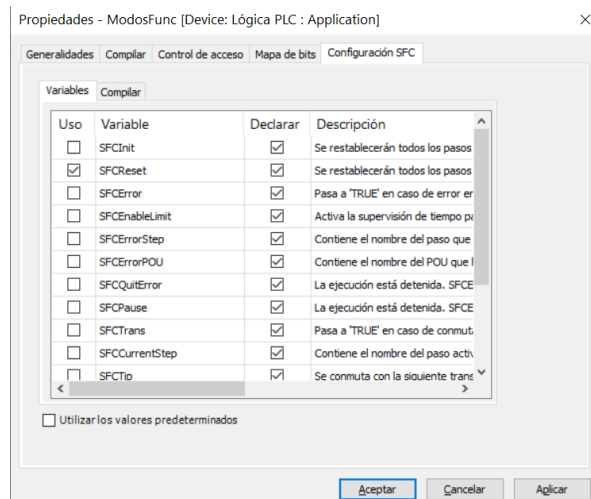
Por último, hacer mención a los programas TramoC2B y TramoC2T, en ellos se ha configurado un contador creciente y decreciente, que llevará la cuenta de las piezas que se encuentran en el tramo central de cintas transportadoras. Mientras su valor sea mayor de 0, se encargará de activar las cintas correspondientes a este tramo. Lo que detiene el funcionamiento de estas cintas son los puntos de saturación explicados con anterioridad.



**Figura 5.15** Contador CTUD

Para llevar a cabo el reset del estado lógico de las variables, se utilizan diferentes métodos dependiendo del lenguaje de programación implementado. En el caso de SFC, se puede utilizar el parámetro de entrada SFCReset, el cual debe ser declarado como VAR\_INPUT en cada POU a resetear.

En CODESYS V3.5 es necesario habilitar en cada Grafcet esta variable, para ello click derecho sobre el código SFC → ‘Propiedades’ → ‘Configuración SFC’, desmarcar la casilla ‘Utilizar los valores predeterminados’ y marcar la casilla de la variable que se desea utilizar.



**Figura 5.16** Habilitar SFCReset

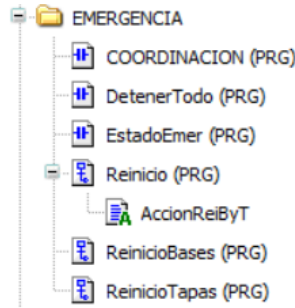
```
MovBRAZO (SFCReset:=TRUE);
Clasificacion(SFCReset:=TRUE);
Reinicio(SFCReset:=TRUE);
ReinicioBases(SFCReset:=TRUE);
ReinicioTapas(SFCReset:=TRUE);
```

**Figura 5.17** Accion RESET del POU ModosFunc

Si el POU está programado en LD, para reiniciar las variables se activan bobinas de reset. Por otro lado, si el POU está programado en ST, se asignan directamente a las variables el valor FALSE.

### 5.3.2.2 EMERGENCIA

Programas encargados de detener el sistema cuando se pulse la seta de emergencia y de realizar la limpieza automática, dejando el sistema en estado y posición inicial listo para comenzar a funcionar con normalidad.



**Figura 5.18** Carpeta EMERGENCIA

La limpieza automática se ha programado llevando la cuenta en todo momento de las piezas que hay en el sistema. El contador correspondiente a las tapas se incrementa cada vez que se genera una materia prima nueva en la cinta de las tapas y se decrementa cuando se ha formado un conjunto. Lo mismo con el contador de bases.

El contador de conjuntos aumenta cuando la tapa se coloca sobre la base y se suelta, y se decrementa cuando el conjunto ha sido clasificado. Por lo tanto, este contador oscila entre 0 y 1, ya que antes de que haya dos conjuntos formados, el primero ya ha sido clasificado.

La limpieza automática se realizará desde el final del sistema hasta el inicio, para ello primero se comprueba el contador de conjuntos, si no hay conjuntos formados la parte final está limpia, se procede a limpiar las partes correspondientes a las bases y las tapas hasta que los contadores correspondientes valgan 0. Si se pulsa el botón Reinicio Contadores, no se realizará ninguna de estas comprobaciones, sino que directamente se reiniciarán los contadores a 0 y es responsabilidad del operario limpiar el sistema.

Para realizar la limpieza automática, el componente del sistema que tiene más grados de libertad es el brazo robótico, así que se llevará a una posición conocida, la de reposo. Para ello lo primero que hará será retraerse verticalmente y después horizontalmente. Cuando queden solo 2 tapas en el sistema, se comprobará si una de ellas se encuentra en la ventosa del brazo robótico, en caso de ser afirmativo, cuando solo quede esta última, el robot la colocará en la cinta de las tapas y será desechada, de lo contrario se continuará con la limpieza normal.

Una consideración importante que se ha tenido en cuenta es que, si en el momento que se pulsó el botón de emergencia el robot iba justo a soltar la tapa sobre la base, al llevar el robot a la posición de reposo, el brazo levantará no solo la tapa, sino también la base, ya que ambas habrán encajado. El procedimiento del sistema es el mismo que se ha explicado en el párrafo anterior, pero hay que tenerlo en cuenta en el contador correspondiente a las bases, ya que el brazo desechará tanto la tapa como la base por la cinta de las tapas.

El primer modo de reinicio implementado era más robusto, pero implicaba una mayor duración de tiempo porque se encargaba de limpiar todo el sistema parte por parte, esta versión se encarga de limpiar únicamente las partes donde hay piezas. Por este motivo, se ha hecho tanto énfasis en la explicación del modo manual en las restricciones y en el botón Reinicio Contadores.



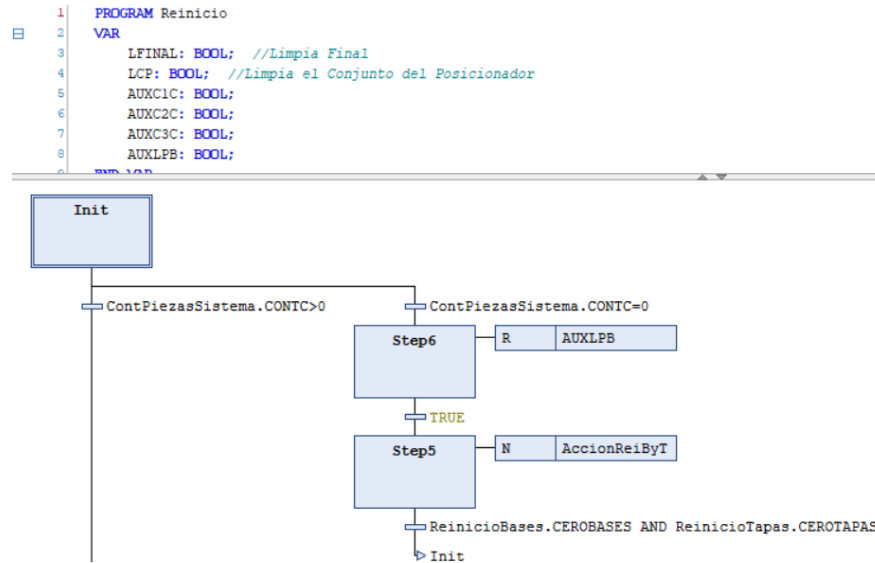


Figura 5.19 Comienzo del reinicio

### 5.3.2.3 FACTORY

Contiene la parte de la programación relacionada directamente con FACTORY, en este sistema el control de las señales de los emisores de material prima. Si los eliminadores estuvieran configurados para ser controlados desde Codesys, se incluirían también en esta carpeta.

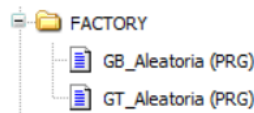


Figura 5.20 Carpeta FACTORY

Dado que CODESYS V3.5 no tiene ninguna función que genere números aleatorios, para generar combinaciones de colores diferentes se han creado dos contadores con distintas saturaciones, uno para cada emisor. De esta forma se crea una cierta aleatoriedad con el paso del tiempo de simulación.

```

IF CONT < 2 THEN
    VG.GBASE:=16;
ELSIF CONT<5 THEN
    VG.GBASE:=32;
END_IF

IF CONT=5 THEN
    CONT := 0;
END_IF

```

Figura 5.21 Materia prima aleatoria para bases

```

IF CONT < 5 THEN
    VG.GTAPA:=16;
ELSIF CONT<8 THEN
    VG.GTAPA:=32;
END_IF

IF CONT=8 THEN
    CONT := 0;
END_IF
    
```

**Figura 5.22** Materia prima aleatoria para tapas

### 5.3.2.4 MANUAL

Contiene un solo programa que tiene como entrada los botones y actúa directamente sobre los actuadores.



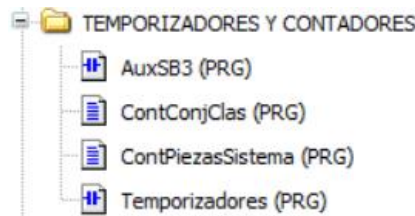
**Figura 5.23** Carpeta MANUAL



**Figura 5.24** Fragmento del modo manual

### 5.3.2.5 TEMPORIZADORES Y CONTADORES

En esta última carpeta se encuentra el código correspondiente al conteo de piezas y conjuntos en el sistema y los temporizadores.



**Figura 5.25** Carpeta TEMPORIZADORES Y CONTADORES

Los temporizadores son los asociados a la limpieza del sistema y a los movimientos del brazo robótico. Se ha asignado un tiempo de 1500 ms al movimiento horizontal del brazo, 1000ms al movimiento vertical y 500ms al agarre de la ventosa.

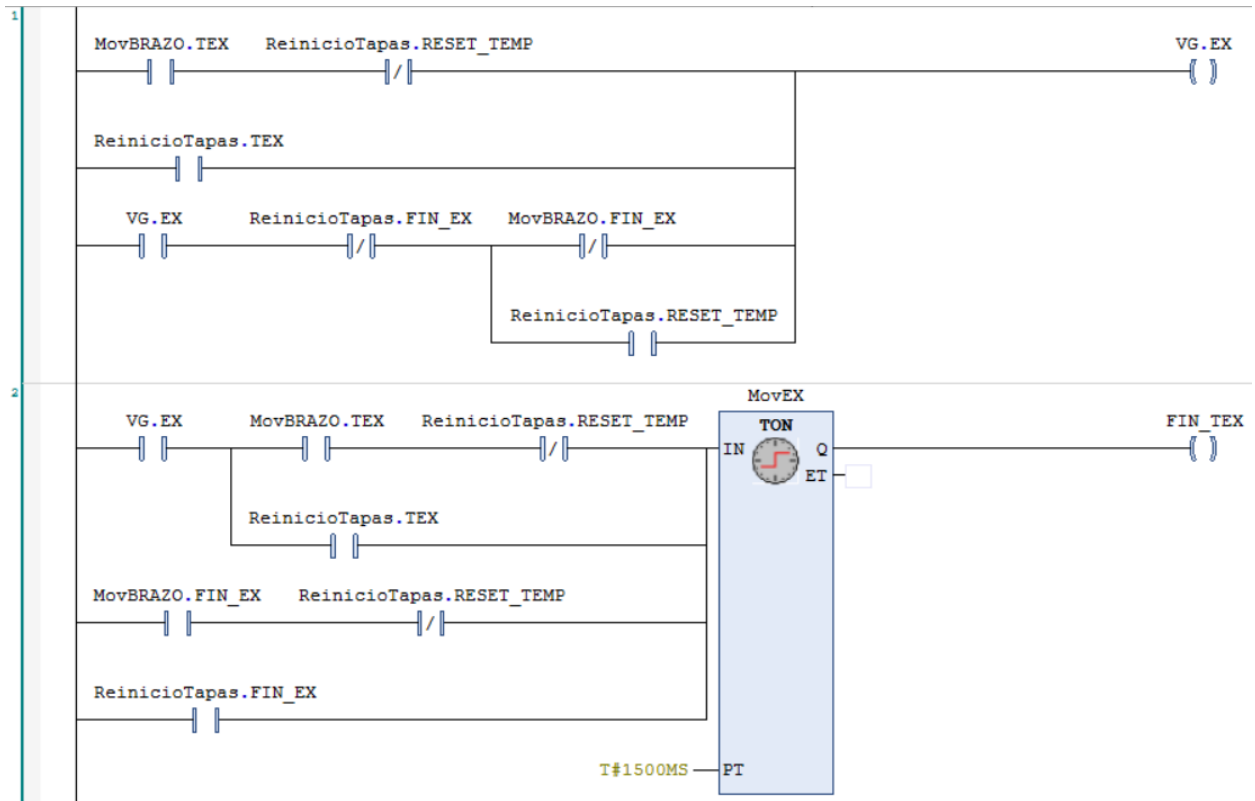


Figura 5.26 Fragmento de los temporizadores

CODESYS permite implementar temporizadores en lenguaje SFC de forma más directa, pero al tratarse de algo particular de CODESYS que no puede ser aplicado en otros softwares, se ha optado por programar en LD los temporizadores más significativos del sistema. No obstante, aquí se muestra un ejemplo:

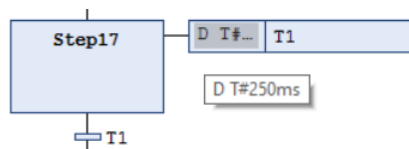


Figura 5.27 Temporizador en SFC

## 5.4 RESULTADO

Para demostrar el correcto funcionamiento del sistema, se ha grabado una demostración que se encuentra disponible en el siguiente [enlace](#).

## 6 CONCLUSIONES Y TRABAJOS FUTUROS

---

En un principio consideré centrar el proyecto únicamente en CODESYS, tanto el control como la visualización. Sin embargo, trabajar con sistemas virtuales que simulen el comportamiento real de los dispositivos y maquinarias, permitiría probar y validar los sistemas desarrollados antes de implementarlos en el mundo real, lo que no sería posible con las visualizaciones en CODESYS, ya que solo permiten ver el comportamiento teórico del sistema. Fue en este punto cuando surgió la idea de implementar los sistemas en una interfaz de usuario más realista y detallada como la que ofrece FACTORY IO, lo que facilitaría la interpretación de los datos de los dispositivos y la interacción con el entorno.

Además, FACTORY IO es compatible con múltiples plataformas PLC, lo que significa que los usuarios pueden trabajar con diferentes tipos de PLCs y dispositivos de automatización en una plataforma común. En cambio, las visualizaciones en CODESYS solo pueden utilizarse en la plataforma CODESYS.

Al intentar establecer la conexión entre ambos programas fue cuando surgió la principal dificultad de este proyecto. La información disponible en línea no era adecuada para la versión de CODESYS V3.5 que tenía instalada, y la mayoría de ella se refería a CODESYS V2.3. A pesar de buscar ayuda en videotutoriales y foros, no pude resolver diferentes errores que me surgieron. Finalmente, decidí descargar una versión de CODESYS más reciente y basarme en la documentación proporcionada por la página web de FACTORY IO, lo que me permitió conectar ambos programas satisfactoriamente.

En cuanto a las proyecciones futuras del proyecto, una posible línea de investigación sería el estudio de los distintos softwares disponibles para el control de sistemas automatizados industriales, lo que permitirá una selección más precisa de los programas que mejor se adapten a las necesidades específicas del proyecto.

Además, se podría explorar la integración de tecnologías emergentes, como la inteligencia artificial y el aprendizaje automático, que permitirían el desarrollo de sistemas más complejos y adaptativos, capaces de tomar decisiones en tiempo real y optimizar el proceso de producción de manera autónoma.

Una vez que se hayan diseñado y probado los sistemas en la simulación, se podría buscar la colaboración con empresas para la implementación de los mismos en un entorno real. Esto permitiría medir la efectividad de los sistemas en un entorno real y ajustarlos en consecuencia. Además, la colaboración con empresas podría ayudar a identificar necesidades y oportunidades específicas en el mercado, lo cual permitiría desarrollar sistemas más enfocados en las necesidades reales de la industria.

En definitiva, este proyecto de simulación de sistemas automatizados industriales tiene un gran potencial para contribuir al avance de la eficiencia y la calidad en la producción industrial, y sus proyecciones futuras podrían tener un impacto significativo en la industria y abrir nuevas oportunidades para la investigación y el desarrollo en este campo.

## REFERENCIAS

---

- [1] Pérez, J. (2017). "IDE Programación". Revista de UNIR. [En línea]. 3(1), 12-18. Recuperado el 31 de marzo de 2023, de <https://www.unir.net/ingenieria/revista/ide-programacion/>
- [2] Contaval. ¿Qué es CODESYS y por qué es importante en la industria de la automatización? (s.f.). Recuperado el 31 de marzo de 2023, de <https://www.contaval.es/que-es-codesys-y-por-que-es-importante-la-industria-de-la-automatizacion/>
- [3] Real Games. Factory IO. (s.f.). Recuperado el 31 de marzo de 2023, de <https://factoryio.com/>
- [4] OPIRON. ¿Qué es OPC UA? (s.f.). Recuperado el 31 de marzo de 2023, de <https://www.opiron.com/que-es-opc-ua/>
- [5] RIDAO CARLINI. M.Á.. 2018. Introducción a la programación de autómatas programables usando CoDeSys. Sevilla: Editorial Universidad de Sevilla. Escuela Técnica Superior de Ingeniería. ISBN 9788447221356.
- [6] Universidad de Sevilla. (s.f.). Automatización y Control de Sistemas de Producción [transparencias]. Máster en Ingeniería Industrial. Sevilla: Universidad de Sevilla. Recuperado el 31 de marzo de 2023
- [7] CODESYS GmbH. CODESYS - the IEC 61131-3 automation software. (s.f.). Recuperado el 31 de marzo <https://www.codesys.com/>
- [8] CODESYS GmbH. CODESYS Help Online. (s.f.). Recuperado el 31 de marzo de 2023, de <https://help.codesys.com/>



# ANEXO A: MODULARIZACIÓN

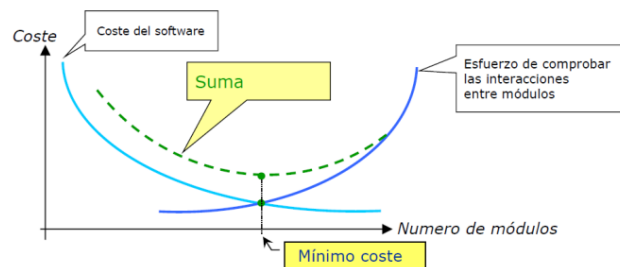
En la implementación de automatismos complejos, es útil dividir el sistema en subsistemas para facilitar la programación, depuración y mantenimiento. Este concepto se llama **Modularización**, donde cada subsistema se corresponde con una POU (Programa de Organización de la Unidad) de tipo programa que se ejecuta en un orden determinado.

Los subsistemas sólo intercambian información a través de variables internas definidas por el usuario como variables globales. Para cada subsistema, es necesario definir por un lado las entradas, que incluyen entradas del autómatas, salidas del autómatas gestionadas por otros subsistemas y variables internas gestionadas por otros subsistemas; y por otro lado, las salidas, que incluyen salidas del autómatas y variables internas.

Es importante tener en cuenta que, en caso de haber varios subsistemas ejecutándose simultáneamente, no pueden compartir señales de salida. Esto puede provocar problemas de asignación múltiple y las señales en cuestión no tendrían el funcionamiento deseado.

Por lo tanto, se debe establecer una jerarquía entre estas POUs que incluya la noción de inhibición de código, y que permita la ejecución de solo uno de los programas en cada ciclo del autómatas. Otra opción es establecer una coordinación mediante el uso de variables internas de estos subprogramas.

Aunque se promueve la idea de subdividir el problema, no se recomienda dividirlo indefinidamente, ya que esto aumentaría el esfuerzo para solucionar problemas en los submódulos y se tendría que comprobar numerosas interacciones entre ellos. La representación gráfica sería la siguiente curva:



**Figura A.1** Modularización

El objetivo es crear bloques (módulos) que automaticen partes diferenciadas del sistema y se puedan reutilizar. Hay dos características principales a tener en cuenta para estos módulos:

- Grado de cohesión: se refiere a los objetivos o funciones que un módulo intenta cumplir.
- Grado de acoplamiento: se refiere al número de módulos con los que interactúa el módulo.

La modularidad del software es buena si los módulos presentan una gran cohesión intrínseca, es decir, si un módulo tiene varias entradas pero solo una función a cumplir. Además, los módulos deben tener un acoplamiento débil entre ellos, lo que significa que cada módulo debe interactuar con el menor número de nodos posible de módulos.





# ANEXO B: GUÍA GEMMA

## 1. Introducción

La ADEPA (Agence nationale pour le Développement de la Productique Appliquée à l'industrie) ha presentado la guía GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts) como un sistema universal para definir los modos de funcionamiento, arranque y parada en la industria.

GEMMA es una guía de estudio gráfica que permite definir los diferentes modos en los que puede operar un sistema y especifica los caminos y condiciones de cambio entre modos que atraviesa un sistema. Se complementa con las Redes de Petri y GRAFCET para una descripción más detallada del automatismo.

Esta guía cuenta con tres modos principales: producción normal, parada y fallo. Cabe destacar que la producción puede ocurrir en cualquier modo, aunque en algunos casos las unidades producidas pueden no ser válidas.

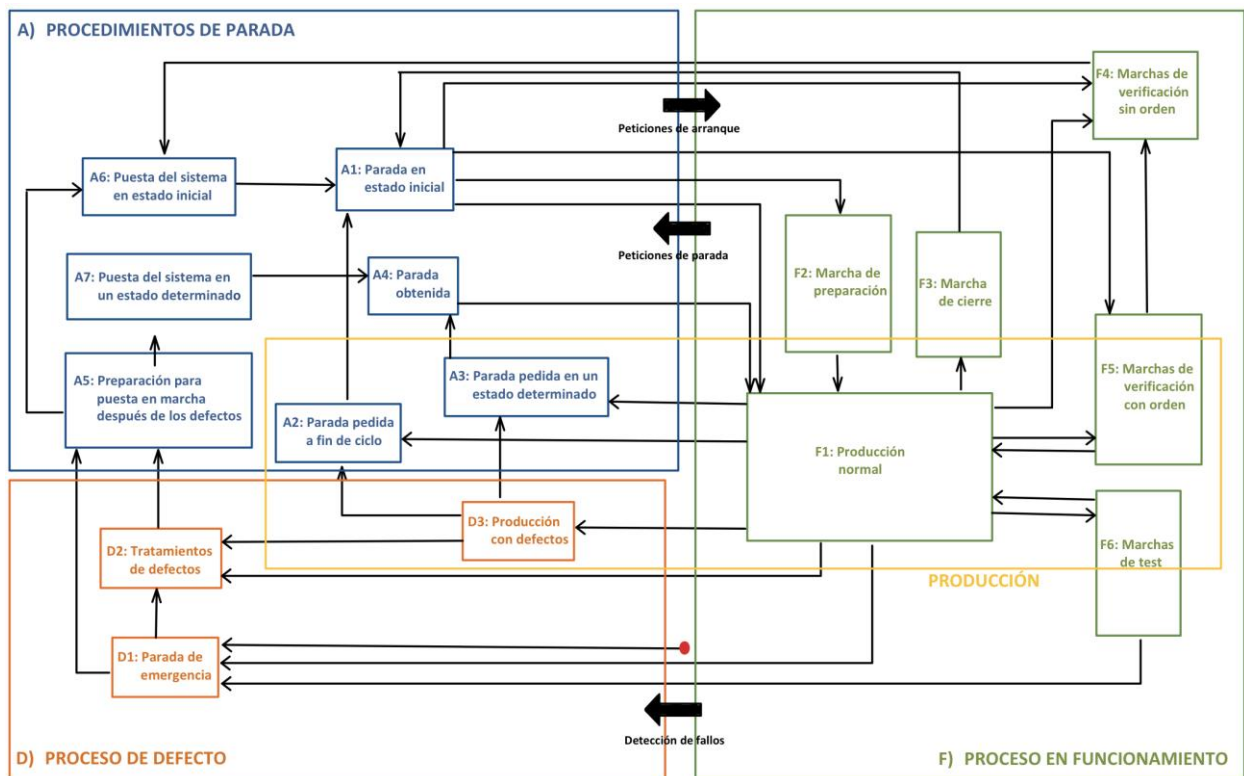


Figura B.1 Guía GEMMA

Es importante aclarar que, tal como su nombre lo indica, la guía GEMMA es simplemente una guía. Dependiendo de los requisitos del sistema y/o los criterios del programador, es posible proponer caminos y cambios de modos alternativos o adicionales.

En definitiva, para elaborar un automatismo completo, es recomendable enriquecer el programa base mediante la adición de condiciones de maniobra y descomponer el programa en procesos, modos o sistemas sencillos, que puedan ser probados por separado, y disminuyan la complejidad que presente el sistema inicialmente.

La coordinación entre sistemas puede ser compleja y puede realizarse de manera vertical o jerarquizada, en la que un sistema maestro gobierna a los sistemas esclavos, o de manera horizontal, en la que no hay sistema

maestro y los programas operan en paralelo. Se recomienda esta última coordinación para pocas tareas y con relaciones limitadas.

## 2. Aplicación

A continuación, se indican los modos y los caminos que presentan los sistemas desarrollados.

### 2.1 MODOS

- **Grupo F. Procedimientos de funcionamiento**

- **F1: Producción normal**. El estado de producción normal del sistema es el estado más importante, ya que en él se deben realizar las tareas por las cuales el sistema ha sido desarrollado.
- **F4: Marchas de verificación sin orden**. En este caso, la máquina puede realizar cualquier movimiento, ya sea según las órdenes del operario o siguiendo una serie de movimientos preestablecidos. Esta función se conoce como control manual, y se utiliza principalmente para tareas de mantenimiento y verificación.

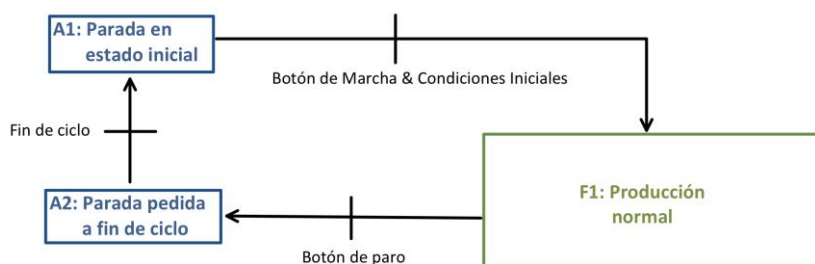
- **Grupo A. Procedimiento de paradas y puestas en marcha.**

- **A1: Parada en el estado inicial**. Estado de reposo del sistema.
- **A2: Parada solicitada al final del ciclo**. Se trata de un estado transitorio en el cual el sistema, que previamente se encontraba produciendo con normalidad, debe continuar produciendo solo hasta finalizar el ciclo y luego detenerse en su estado inicial.
- **A4: Parada obtenida**. Es un estado de reposo del sistema distinto al estado inicial.
- **A6: Puesta del sistema en el estado inicial**. Estado de reinicio donde se realiza el retorno, manual o automático, del sistema al estado inicial.

- **Grupo D. Procedimientos de defecto**

- **D1: Parada de emergencia**. Estado alcanzado después de una parada de emergencia donde se consideran tanto las paradas como los procedimientos y precauciones necesarias para evitar o limitar las consecuencias de los defectos.

### 2.2 CAMINOS



**Figura B.1** Parada a fin de ciclo

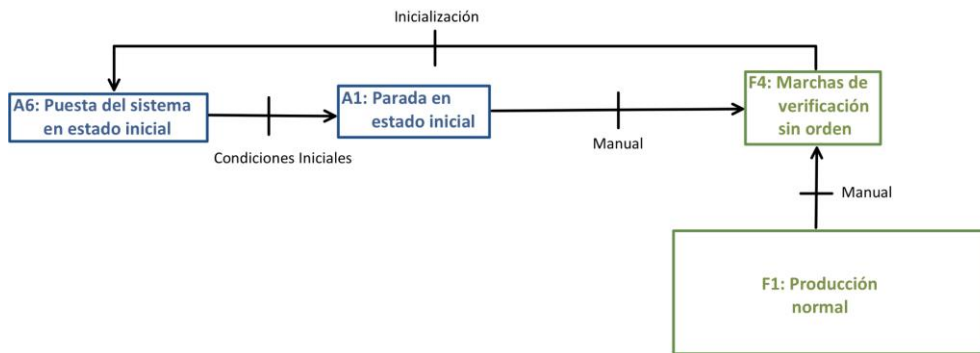


Figura B.2 Funcionamiento manual

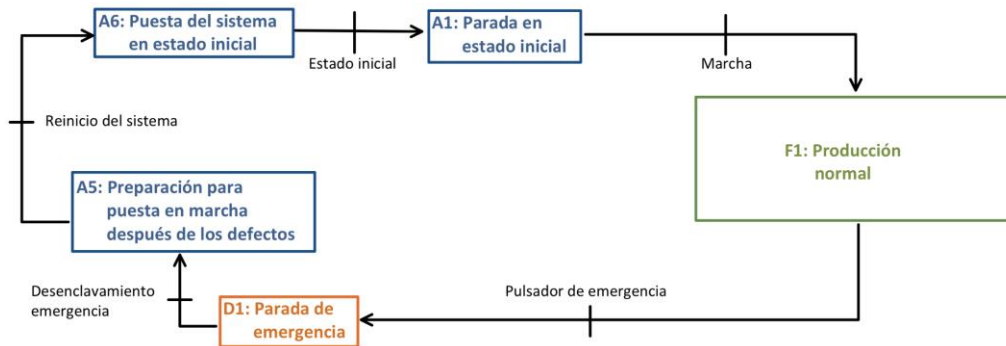


Figura B.3 Pulsador de emergencia

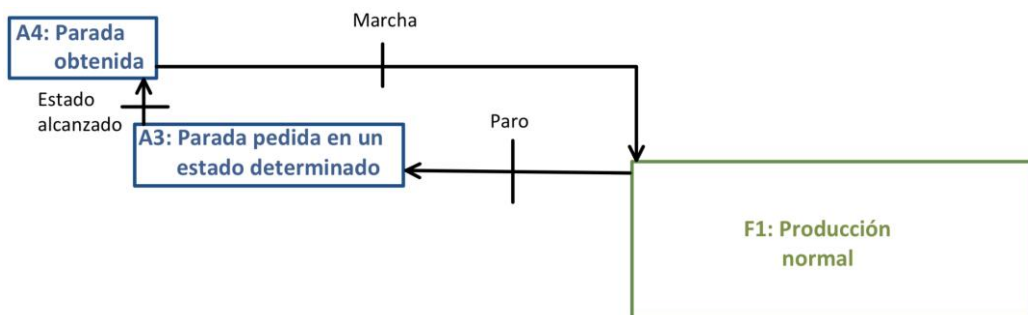


Figura B.4 Parada y reanudación en una etapa determinada