

Trabajo Fin de Grado de Ingeniería de Tecnologías Industriales

Automatización y picking de un almacén logístico

Autor: Ramón Enrique Jiménez Bellido

Tutor: Francisco Salas Gómez

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

Sevilla, 2021



Trabajo de Fin de Grado de
Ingeniería en Tecnologías Industriales

Automatización y picking de un almacén logístico

Autor:

Ramón Enrique Jiménez Bellido

Tutor:

Francisco Salas Gómez

Profesor titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Autor: Ramón Enrique Jiménez Bellido

Tutor: Francisco Salas Gómez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Gracias a mis padres, por permitirme haber estado estos cuatro años estudiando esta carrera y apoyarme a lo largo de ella. Gracias por recomendarme venirme el segundo año a Sevilla desde Madrid porque veían que era lo mejor para mí, estaban en lo correcto.

Gracias a todos los profesores que me han formado durante este camino, que me han aportado tanto la curiosidad como el deseo por aprender. En especial agradecer a mi tutor Francisco Salas Gómez que, de todos, logró ser el que más me enseñó y el más cercano en tiempos de COVID y que confió en mí para darme la oportunidad de realizar este TFG dándome la libertad para poder llevar a cabo lo que quería.

La pandemia ha sido algo duro de llevar para toda España, por eso agradecer a todos los trabajadores que han conseguido realizar sus trabajos incluso cuando debían hacerlo desde sus casas y en malas circunstancias, a los sanitarios por ser el muro que nos ha salvado durante este tiempo, a los profesores y alumnos, por esforzarse en conseguir que el online se pareciera lo más posible a la presencialidad aunque fuera complicado, y para terminar, y porque me toca de cerca, a todos los agricultores y ganaderos por haber seguido dándonos de comer llenando nuestros supermercados.

Ramón Enrique Jiménez Bellido

Sevilla, 2021

Resumen

Hoy en día y gracias a las nuevas tecnologías, un almacén se ha convertido en algo más que un sitio donde almacenar mercancía por operarios a pie en espera de que sea transportada a otro lugar. Ahora los almacenes son instalaciones complejas, con maquinaria inteligente que deben llevar a cabo funciones programadas y que necesitan formar parte del flujo del producto llevando a cabo desde la recepción de productos hasta la entrega a los clientes finales, todo ello realizando controles avanzados de diferentes tipos.

Esto junto con el deseo de las empresas actuales las cuales requieren gran precisión en todo el manejo del producto, eliminando en lo posible los errores y con controles avanzados de tiempo y de nivel de stock, se ha conseguido que la única forma de llevarlo a cabo fuera la implantación de almacenes automatizados. Estos almacenes consiguen el movimiento y almacenaje de los productos de manera programada y automatizada, con la ayuda de maquinaria compleja, con la necesidad de pocos operarios y consiguiendo un nivel de errores muy inferior y por lo tanto una seguridad de la mercancía excepcional.

La economía es un ámbito de la empresa que mejora después de la implantación de un almacén automatizado debido entre otros motivos al aprovechamiento de espacios, consiguiendo optimizar el uso de la superficie disponible adaptándose a cada lugar y cada tipo de producto que se desea guardar. Se aprovecha también la productividad ya que estos almacenes pueden trabajar en cualquier horario y a una velocidad superior al resto. Por último, se consigue una mejor economía de costes laborales debido a que los operarios que antes trabajaban en el almacenaje de productos ahora puedan dedicarse a otras tareas.

En este proyecto se podrá observar cómo se ha implantado un almacén automatizado que consigue una recepción de productos, un almacenaje de los mismos y un envío al cliente final de forma completamente automatizada. Se ha conseguido realizar un sistema de picking que cumple con los requisitos de cada cliente pudiendo enviar diferentes tipos de productos a diferentes localizaciones llevando el control sobre ellos y sobre la planta en su totalidad.

Abstract

Nowday and thanks to new technologies, a warehouse has become more than just a place to store merchandise by workers on foot waiting for it to be transported to another place. Now warehouses are complex facilities, with intelligent machinery that must carry out programmed functions and that need to be part of the product flow, carrying out from the receipt of products to delivery to the end customers, all of this by performing advanced controls of different type.

This together with the desire of current companies which require great precision in all product handling, eliminating errors as far as possible and with advanced time and stock level controls, it has been achieved that the only way to carry it out outside the implantation of automated warehouses. These warehouses manage the movement and storage of products in a programmed and automated way, with the need for few operators and achieving a much lower level of errors and therefore exceptional merchandise security.

The economy is an area of the company that improves after the introduction of an automated warehouse due to, among other reasons, the use of space, optimizing the use of the available surface, adapting to each place and each type of product to be stored. Productivity is also taken advantage of since these warehouses can work at any time and at a higher speed than the rest. Finally, a better labor cost economy is achieved because operators who previously worked in the storage of products can now dedicate themselves to other tasks.

In this project it will be possible to observe how an automated warehouse has been implemented that achieves a receipt of products, a storage of them and a shipment to the end customers in a completely automated way. It has been possible to create a picking system that meets the requirements of each client, being able to send different types of products to different locations, taking control of them and the plants as a whole.

Índice

Agradecimientos	ixx
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvi
Índice de Figuras	xviii
1 Instrucción y objetivos	1
2 Requisitos de diseño	3
2.1. <i>Softwares</i>	3
2.1.1 Codesys	3
2.1.2 Factory i/o	3
2.2. <i>Conexiones entre softwares</i>	4
3 Diseño de la planta	13
3.1. <i>Descripción de la planta</i>	13
3.2. <i>Componentes de la planta</i>	14
3.2.1. Maquinaria	14
3.2.2. Sensores	23
3.3. <i>Requerimiento de espacio y ubicaciones físicas</i>	24
3.3.1. <i>Zona de recepción de productos</i>	25
3.3.2. <i>Zona de almacenaje</i>	27
3.3.3 <i>Zona de envío de pedidos</i>	30
3.4. <i>Limitaciones y posibles mejoras</i>	31
4 Descripción de la programación del controlador	33
4.1. <i>Descripción de variables</i>	33
4.1.1. Variables de salida	33
4.1.1.1. <i>Zona de recepción de productos</i>	39
4.1.1.2. <i>Zona de almacenaje</i>	40
4.1.1.3. <i>Zona de envío de pedidos</i>	45
4.1.2. Variables internas	48
4.2. <i>Programación del controlador</i>	50
4.2.1. Descripción de los POU's	51
5 SCADA y modos de control	70
5.1. <i>Diseño del SCADA</i>	70
5.2. <i>Variables y funciones del SCADA</i>	72

5.2.1. Pedir un producto	73
5.2.2. Comprobación de Stocks	75
5.2.3. Comprobación de pedidos	76
5.2.4. Resolución de problemas	77
5.2.3. Comprobación de pedidos	77
5.3. Modos de control	80
5.3.1. Automático	81
5.3.2. Parada de emergencia y parada	82
5.3.3. Rearme	83
6 Problemas observados, antecedentes y mejoras	87
7 Resultados finales y conclusiones	90
Referencias	91
Anexos	92
Anexo-A. Vídeos	92
Anexo-B Archivos Codesys V3.5	92
Anexo-C Archivos Factory IO	92

ÍNDICE DE TABLAS

Tabla 4-1: Sensores de las cintas ubicadas en la zona de recepción	39
Tabla 4-2: Actuadores de las cintas ubicadas en la zona de recepción	39
Tabla 4-3: Actuadores y sensores de la base giratoria 1	40
Tabla 4-4: Actuadores cintas generales ubicadas en la zona de almacenaje	41
Tabla 4-5: Sensores cintas generales ubicadas en la zona de almacenaje	41
Tabla 4-6: Actuadores y sensores de las cintas específicas de productos verdes	42
Tabla 4-7: Actuadores y sensores del apilador de productos verdes	43
Tabla 4-8: Actuadores y sensores de las cintas específicas de productos plateados	43
Tabla 4-9: Actuadores y sensores de las cintas específicas de productos azules	44
Tabla 4-10: Actuadores y sensores de las cintas generales ubicadas en la zona de envío	45
Tabla 4-11: Actuadores y sensores de la grúa recogedora de productos y de los brazos giratorios	46
Tabla 4-12: Contadores de las cintas	48
Tabla 4-13: Contadores de los productos recibidos, almacenados y pedidos	48
Tabla 4-14: Matrices del programa	49

ÍNDICE DE FIGURAS

Figura 1-1: Perspectiva planta FACTORY IO	2
Figura 2-1: Logo CODESYS	4
Figura 2-2: Logo FACTORY IO	4
Figura 2-3: Creación de un nuevo proyecto 1 CODESYS	5
Figura 2-4: Creación de un nuevo proyecto 2 CODESYS	5
Figura 2-5: Creación de una lista de variables 1 CODESYS	6
Figura 2-6: Creación de una lista de variables 2 CODESYS	6
Figura 2-7: Ligado de variables CODESYS	7
Figura 2-8: Acceder a objeto “Configuración de símbolos” CODESYS	7
Figura 2-9: Configuración variables CODESYS	8
Figura 2-10: Apertura “CODESYS Control Win V3” CODESYS	8
Figura 2-11: Dirección de la aplicación “OPCConfig”	9
Figura 2-12: Apertura aplicación “OPCConfig”	9
Figura 2-13: Configuración “OPCConfig”	9
Figura 2-14: Conexión red CODESYS	10
Figura 2-15: Conexión vía OPC FACTORY IO	11
Figura 2-16: Conexión variables FACTORY IO	12
Figura 3-1: Vista de la planta FACTORY IO	14
Figura 3-2: Cintas de rodillos FACTORY IO	15
Figura 3-3: Cintas transportadoras FACTORY IO	15
Figura 3-4: Cintas de carga o descarga FACTORY IO	15
Figura 3-5: Cinta inclinada FACTORY IO	16
Figura 3-6: Cintas de rodillos FACTORY IO	17
Figura 3-7: Base giratoria FACTORY IO	17
Figura 3-8: Brazo giratorio FACTORY IO	18
Figura 3-9: Grúa recogedora de productos FACTORY IO	18
Figura 3-10: Estantería FACTORY IO	19
Figura 3-11: Apiladora de productos FACTORY IO	19
Figura 3-12: Elevador FACTORY IO	20
Figura 3-13: Elementos de separación FACTORY IO	21

Figura 3-14: Componentes de control FACTORY IO	22
Figura 3-15: Sensores FACTORY IO	23
Figura 3-16: Sensor de visión FACTORY IO	23
Figura 3-17: Mapa de nuestra planta VISIO	24
Figura 3-18: Zona de recibo de productos VISIO	25
Figura 3-19: Zona de recibo de productos FACTORY IO	26
Figura 3-20: Zona de recibo de almacenaje VISIO	27
Figura 3-21: Zona de almacenaje de productos verdes FACTORY IO	28
Figura 3-22: Zona de almacenaje de productos azules FACTORY IO	28
Figura 3-23: Zona de almacenaje de productos plateados FACTORY IO	29
Figura 3-24: Zona de envío de productos VISIO	30
Figura 3-25: Zona de envío de productos FACTORY IO	30
Figura 3-26: Apiladora de productos FACTORY IO	29
Figura 3-27: Robot automatizado FARMACIA REAL	30
Figura 3-28: Cinta automática FACTORY IO	30
Figura 3-29: Cinta automatizada FARMACIA REAL	31
Figura 3-30: Problemas de cintas FACTORY IO	32
Figura 4-1: Variables cintas FACTORY IO	34
Figura 4-2: Variables bases giratorias FACTORY IO	35
Figura 4-3: Variables apiladoras de productos FACTORY IO	36
Figura 4-4: Variables elevador FACTORY IO	36
Figura 4-5: Variables grúa recogedora de productos FACTORY IO	37
Figura 4-6: Variables emisor y eliminador de productos FACTORY IO	38
Figura 4-7: Sensores P1, P2, P3 y P4 FACTORY IO	47
Figura 4-8: Lista de POU CODESYS	51
Figura 4-9: POU CONTADORESPROVEEDORES CODESYS	52
Figura 4-10: POU CONTADORESPEDIDOS CODESYS	53
Figura 4-11: POU CONTADORESALMACENES CODESYS	54
Figura 4-12: POU CONTADORES CINTAS CODESYS	55
Figura 4-13: POU C1 CODESYS	56
Figura 4-14: POU GIR1 CODESYS	57
Figura 4-15: POU GIR1 ampliada CODESYS	58
Figura 4-16: POU GIR2 CODESYS	59
Figura 4-17: POU AlmacenA FACTORY IO	60
Figura 4-18: POU ALMACENES CODESYS	61
Figura 4-19: POU ELEVADOR CODESYS	62
Figura 4-20: POU ALMACENES CODESYS	62
Figura 4-21: POU EXTPRODUCTO CODESYS	63

Figura 4-22: POU C8 CODESYS	63
Figura 4-23: POU ALMACENES CODESYS	64
Figura 4-24: POU C10 CODESYS	65
Figura 4-25: POU C10 ampliada CODESYS	66
Figura 4-26: POU ALMACENES CODESYS	67
Figura 4-27: POU ALMACENES CODESYS	68
Figura 4-28: POU ALMACENES CODESYS	69
Figura 5-1: Visualización principal CODESYS	70
Figura 5-2: Lista visualizaciones CODESYS	71
Figura 5-3: Configuración visualización web CODESYS	71
Figura 5-4: Visualización web CODESYS	72
Figura 5-5: Menú principal CODESYS	73
Figura 5-6: Elección de producto CODESYS	73
Figura 5-7: Error disponibilidad de stock CODESYS	74
Figura 5-8: Elección de sala CODESYS	74
Figura 5-9: Número de pedido CODESYS	75
Figura 5-10: Nivel stock CODESYS	75
Figura 5-11: Comprobación pedidos CODESYS	76
Figura 5-12: Teclado comprobación pedidos CODESYS	76
Figura 5-13: Comprobación estado planta CODESYS	77
Figura 5-14: Error fallo en la planta CODESYS	78
Figura 5-15: Elección zona del error fallo en la planta CODESYS	78
Figura 5-16: Elección tipo de producto recibido origen error CODESYS	79
Figura 5-17: Elección tipo de producto enviado origen error CODESYS	80
Figura 5-18: POU MODOFUNCIONAMIENTO 1 CODESYS	81
Figura 5-19: POU MODOFUNCIONAMIENTO 2 CODESYS	81
Figura 5-20: POU AUTOMÁTICO	82
Figura 5-21: POU PARADAEMERGENCIA CODESYS	83
Figura 5-22: POU REARME CODESYS	85
Figura 5-23: POU AlmacenAREARME CODESYS	85
Figura 6-1: Barreras protectoras 1 FACTORY IO	87
Figura 6-2: Barreras protectoras 2 FACTORY IO	87
Figura 6-3: Intento planta 2 y 3 FACTORY IO	88

1 INTRODUCCIÓN Y OBJETIVOS

La automatización de almacenes es una de las claves más importante para cualquier empresa o fábrica cuya cadena logística tenga un peso en la misma importante. Los almacenes han pasado de ser una zona sin ningún valor, aparte del de guardar mercancías, a formar parte importante de la cadena de suministro de un producto. Este cambio se nota de manera inmediata en la mejora del flujo logístico, en un aumento de la productividad, en la mejora de la eficiencia a la hora de eliminar tareas de bajo valor que antes realizaba cualquier operario y en la minimización de errores.

De este objetivo surge este proyecto. En un inicio, la idea era la automatización de un almacén de medicamentos farmacéuticos de un hospital, pero este es un proyecto que se puede asociar a multitud de empresas, donde tanto la zona de recibo de pedidos de proveedores, la zona de almacenaje y la zona de envío de pedidos a clientes puedan ser ubicadas en el mismo edificio, o en diferentes pero cercanos, completando así el ciclo de cualquier producto.

Algunos de los beneficios que aporta un almacén totalmente automatizado son la mejora en los tiempos de envío y almacenaje de los productos con un movimiento fluido y programado de los mismos, se eliminan errores humanos, se aprovecha una superficie de forma óptima al no necesitar por ejemplo de grandes zonas para el recibo de productos, no se necesitan zonas por las que tuviera que ir el operario cargando con los productos o por ejemplo, se consigue transportar mercancía entre diferentes zonas o edificios solo necesitando cavidades con la simple forma de una cinta transportadora. Esto consigue un excelente control de los pedidos y de las ubicaciones de los productos en cada momento.

El mejor ejemplo de lugar donde implantar este proyecto sería, como se ha dicho antes, en un hospital, donde los medicamentos llegaran de forma directa desde el almacén hasta las zonas de tratamientos con la única intervención de máquinas, eliminando así tiempos perdidos de enfermeros o médicos acercándose a por los medicamentos cuando podrían estar tratando a los pacientes.

Para realizar este proyecto, se ha necesitado principalmente de dos softwares: Factory IO, para la simulación en 3D de nuestra planta, y Codesys, para la programación del controlador de la misma.

Los objetivos que se han fijado a la hora de realizar este proyecto han sido:

- Conseguir, en primer lugar, una conexión entre softwares exitosa.
- También, poder utilizar los conocimientos aprendidos en la asignatura automatización industrial y ampliarlos.
- Realizar un diseño del almacén distribuyendo diferentes zonas para cada momento de la vida de un producto, desde la recepción del mismo, siguiendo con su almacenaje hasta el envío al cliente incluyendo en cada una de ellas puntos de control.
- Conseguir un aprovechamiento máximo del espacio con zonas para el movimiento de los operarios.
- Realizar un sistema de interacción del cliente o del operador con la planta sencilla e informativa.
- Conseguir un sistema de picking preciso.
- Incluir modos de funcionamiento para poder reiniciar la planta siempre que se quiera.

Esta automatización se ha conseguido mediante la unión de diferentes softwares. El primero, Codesys que permite realizar la programación del almacén y de las zonas de recibo y entrega de pedidos y la creación de pantallas y Scadas que permiten controlar los pedidos y el funcionamiento de las máquinas. Y de Factory I/O, que permite la simulación del almacén en 3D utilizando las variables asociadas a las de Codesys.

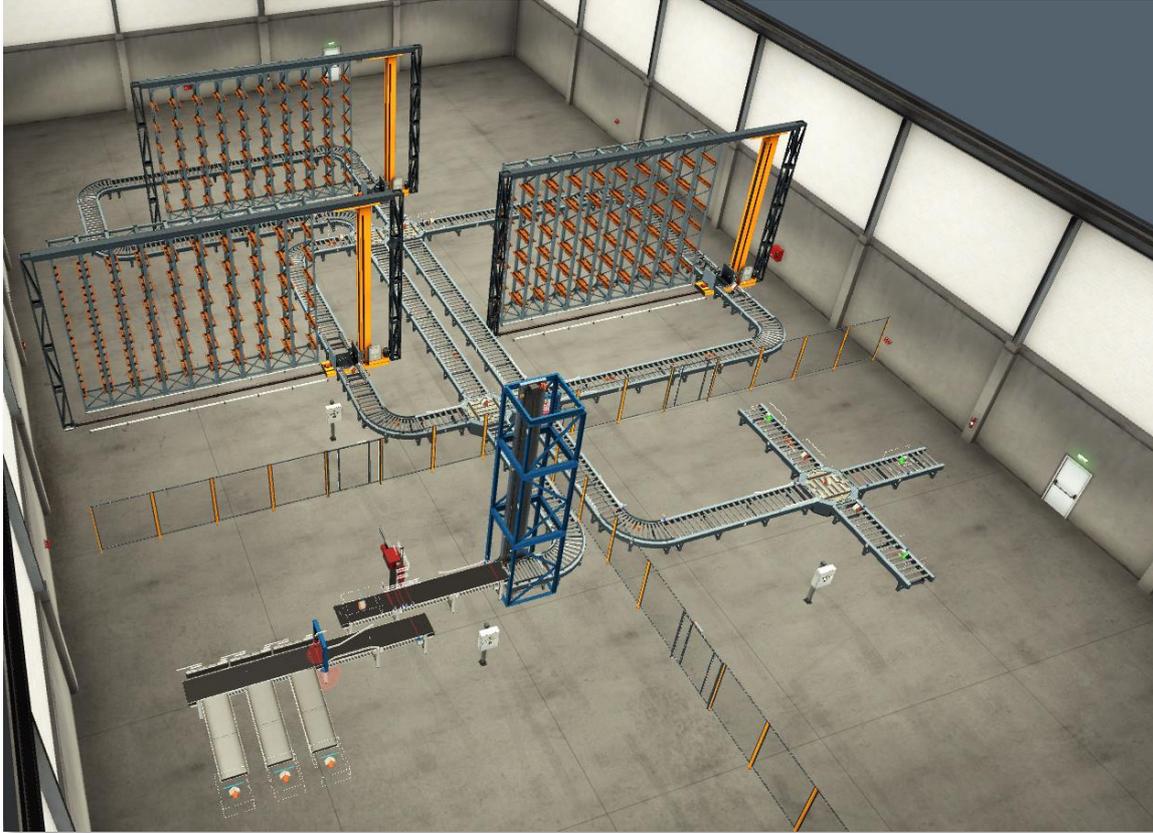


Figura 1-1: Perspectiva planta FACTORY IO

2 REQUISITOS DE DISEÑO

2.1 Softwares

La implementación de la automática en un almacén se puede conseguir mediante diferentes softwares. Nosotros hemos elegido para la parte de programación el software Codesys y algunos que no conocíamos como Factory I/O o Waremare para la parte de simulación o para el control del flujo de productos con un SCADA aunque este último no ha sido posible implementarlo al final debido a problemas con la licencia.

2.1.1 Codesys

Codesys es un ambiente de desarrollo software creada para el desarrollo de proyectos de automatización industrial con el objetivo de configurar, programación, puesta en funcionamiento de controladores y su respectivo mantenimiento. Se eligió este software debido a que era el utilizado durante la carrera y gracias a su facilidad de uso. La versión con la que se aprendió a usar fue la 2.3 pero la última versión de este y la que utilizaremos para este proyecto es la 3.5, con algunas diferencias con respecto a su predecesora, pero con bastantes mejoras.

CODESYS permite trabajar con diferentes lenguajes de programación. En el que nos centraremos en este proyecto será el “Diagrama de funciones secuenciales”, que se le llama de forma abreviada programación “GRAFCET” o “SFC” aunque deberemos utilizar otros como el lenguaje de “Texto estructurado” o “ST” o el lenguaje de “Diagrama Ladder” o “LD”.

Otra función de CODESYS que es de gran utilidad es la de crear visualizaciones que permiten desde crear simulaciones mediante coordenadas y la programación de los movimientos por “ST”, hasta la creación de SCADAS.

Esta última versión de CODESYS la hemos descargado del portal de soporte técnico FESTO. Este portal permitía la descarga de CODESYS de forma gratuita y facilitaba la conexión con el otro software utilizado vía OPC.

La descripción de la plataforma CODESYS según la compañía traducida al castellano es:

“CODESYS es una plataforma de software para la tecnología de automatización industrial. El núcleo de la plataforma es la herramienta de programación IEC-61131-3 o “Sistema de desarrollo CODESYS”. Ofrece a los usuarios soluciones integradas orientadas a la práctica para la configuración conveniente de aplicaciones de automatización” [1]

2.1.2 Factory I/O

Factory IO es un software utilizado para la creación de simulaciones 3D en las cuales podemos ejecutar programas de automatización “PLC” o otros creado por microprocesadores. Permiten una descarga gratuita por 30 días de la última actualización por lo que pudimos utilizarlo.

La utilización de este software ha sido algo nuevo, pero se eligió debido a su rápida comunicación con CODESYS mediante “OPC Data Access” que es la interfaz estándar que permite el acceso a datos de un proceso de automatización, en este caso, a los datos creados en nuestro proyecto de CODESYS. Otro motivo por el que se eligió fue debido a su gran cantidad de componentes a utilizar, aunque no se adaptaran en su totalidad a nuestro proyecto.

La compañía describe Factory I/O traducido al castellano como:

“Factory I/O es una simulación de fábrica en 3D para el aprendizaje de tecnologías de automatización. Diseñado para ser fácil de usar, permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes. Factory I / O también incluye muchas escenas inspiradas en aplicaciones industriales típicas, que van desde niveles de dificultad principiantes hasta avanzados. El escenario más común es utilizar Factory I / O como una plataforma de capacitación de PLC, ya que los PLC son los controladores más comunes que se encuentran en aplicaciones industriales. Sin embargo, también se puede utilizar con microcontroladores, SoftPLC, Modbus, entre muchas otras tecnologías.”. [2]



Figura 2-1: Logo CODESYS



Figura 2-2: Logo Factory IO

2.2 Conexión entre softwares

Como información y para entender mejor el funcionamiento de este proyecto explicaré de forma rápida y sencilla como realizar la comunicación de los softwares utilizados para el desarrollo de este trabajo.

Esta comunicación será a través de “opc Data Access”, interfaz estándar que nos permitirá que Factory I/O pueda acceder a los datos que habremos desarrollado con anterioridad en CODESYS, como a las variables o a los controladores desarrollados para cada componente ubicado en la simulación.

Para la comunicación entre el software Codesys V3.5 y el software de simulación Factory I/O se procede de la siguiente manera:

1º Paso. Iniciamos CODESYS V3.5 y creamos un nuevo proyecto.

2º Paso. Seleccionamos “Proyecto estándar”, le damos un nombre al proyecto y le asignamos una ubicación.

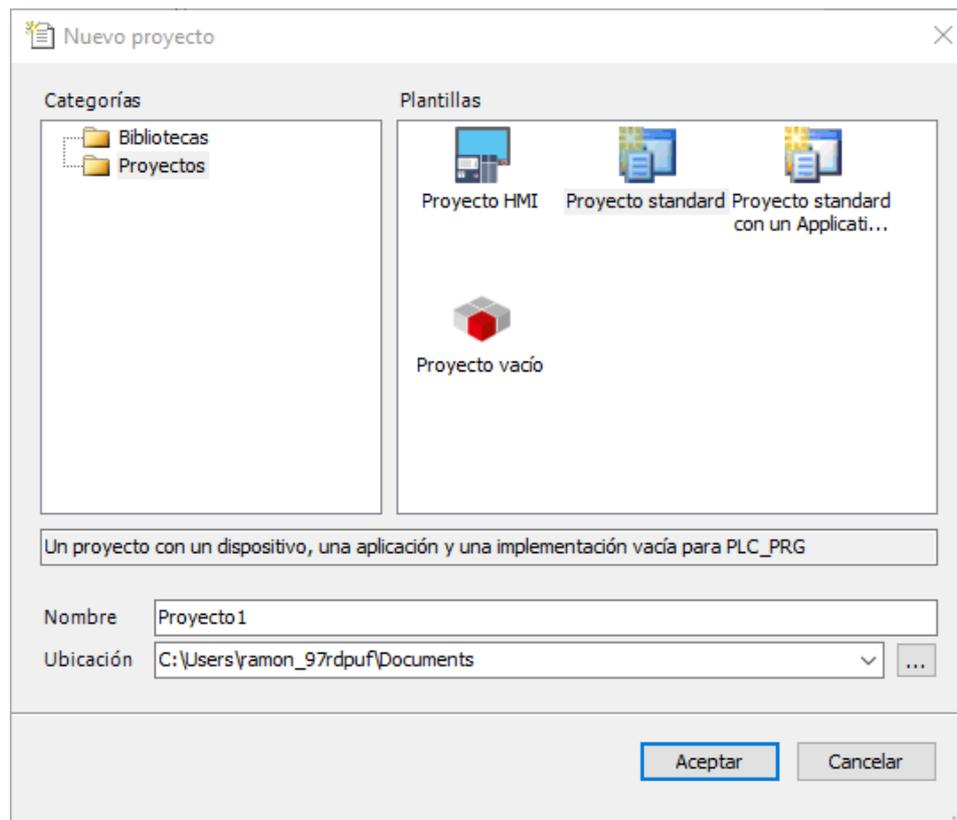


Figura 2-3: Creación de un nuevo proyecto CODESYS

3º Paso. Dejamos el dispositivo predeterminado y seleccionamos un lenguaje para nuestro PLC_PRG que será nuestro POU principal. Se puede elegir el lenguaje que más se prefiera, aunque luego no se tenga que utilizar.

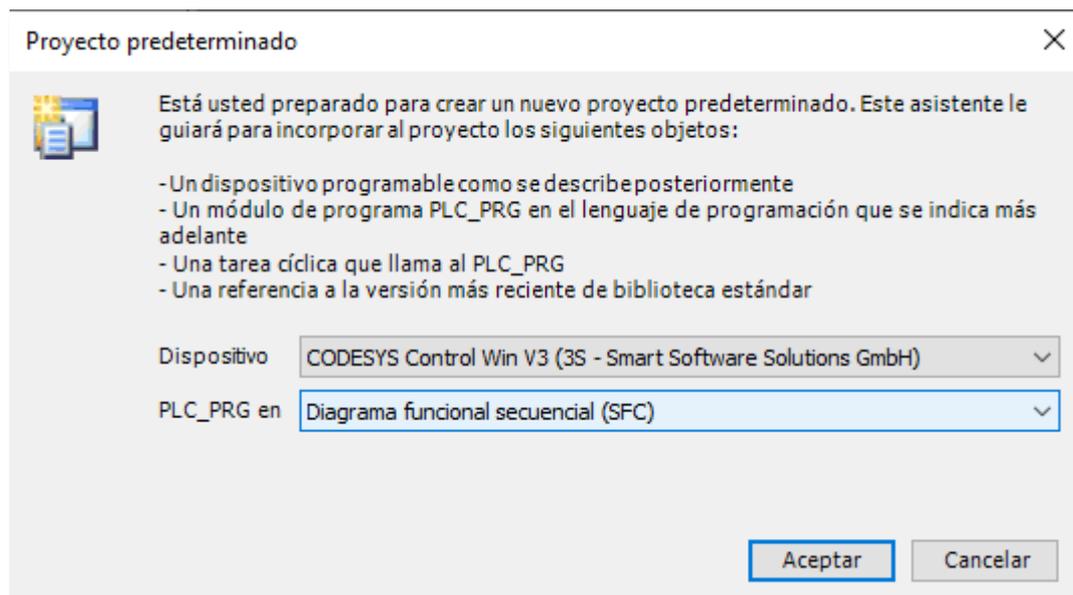


Figura 2-4: Creación de un nuevo proyecto CODESYS

4º Paso. Una vez ya tengamos creado nuestro proyecto y tengamos inicializada la lógica del PLC Y el POU creamos una lista de variables globales, donde podremos crear las variables que utilizaremos en un futuro y que podremos utilizar en cada uno de los diferentes POU's que creamos.

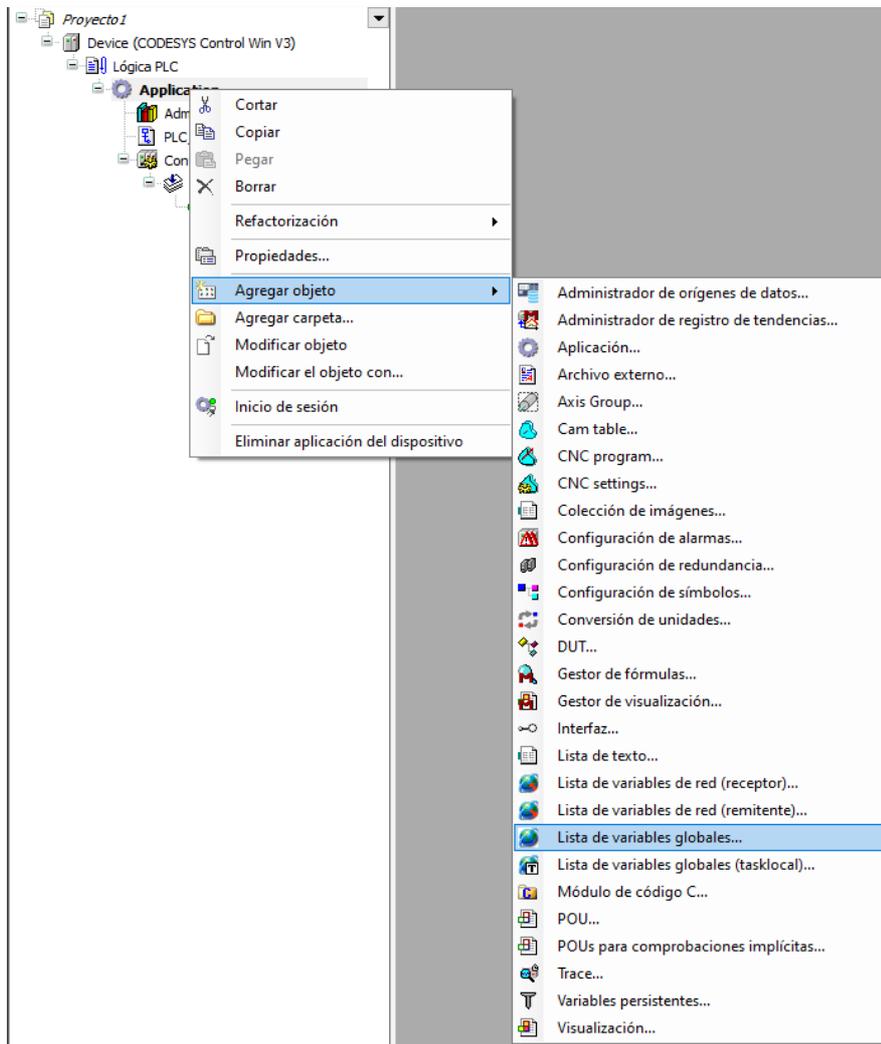


Figura 2-5: Creación de una lista de variables CODESYS

Nosotros para este ejemplo hemos creado cuatro variables de tipo BOOL para una rápida explicación.

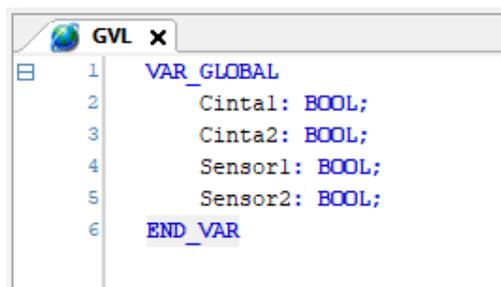


Figura 2-6: Creación de una lista de variables CODESYS

5º Paso. Accedemos a las propiedades de la lista de variables globales mediante botón derecho. En la pestaña Compilar seleccionamos la opción Ligar siempre con la cual permitimos que el Factory IO pueda acceder a estas variables.

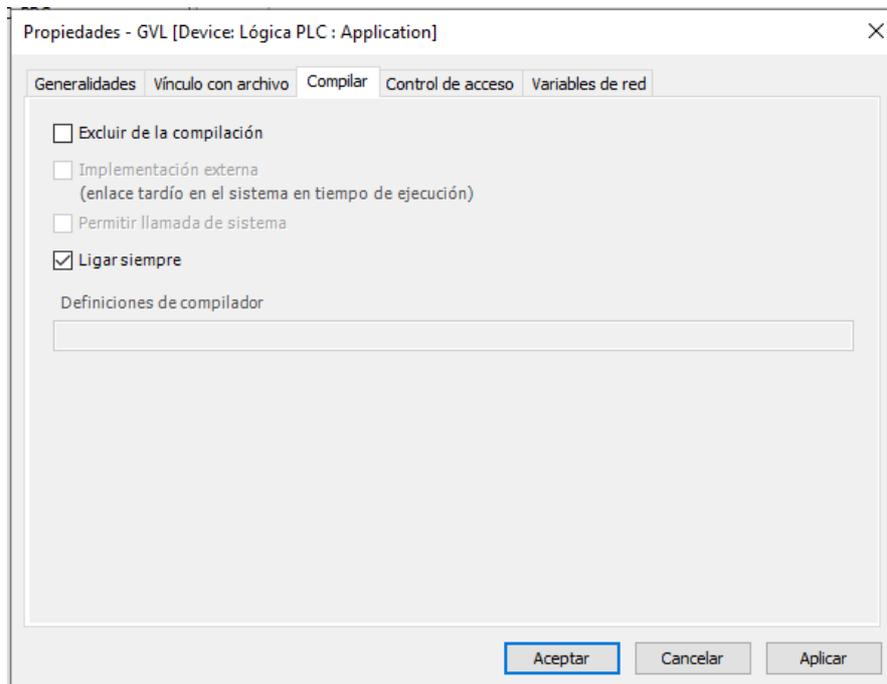


Figura 2-7: Ligado de variables CODESYS

6º Paso. Agregamos un objeto denominado Configuración de símbolos.

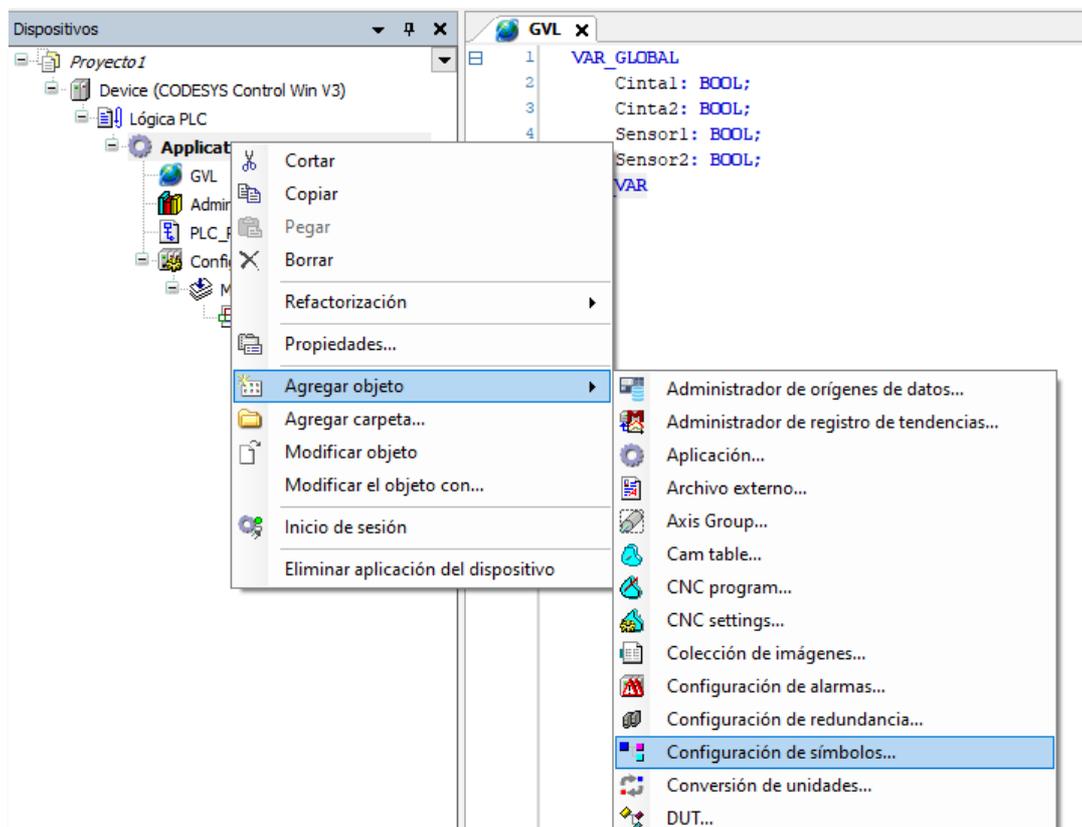


Figura 2-8: Acceder a objeto “Configuración de símbolos” CODESYS

Compilamos para que nos aparezcan nuestras variables y seleccionamos todas ellas. Este paso se debe hacer cada vez que se añadan nuevas variables debido a que siempre que se crean, aparecen sin señalar.

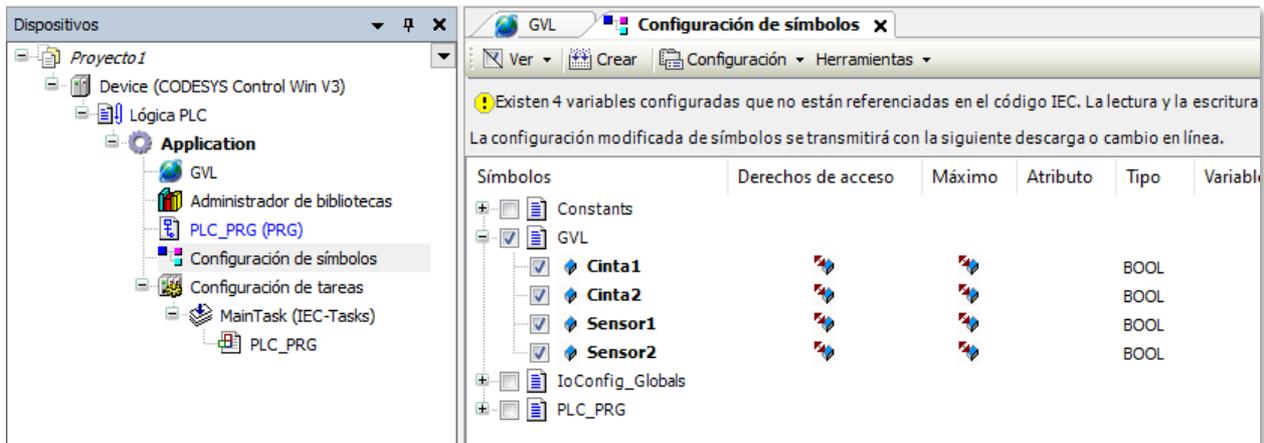


Figura 2-9: Configuración variables CODESYS

7º Paso. Abrimos desde el cuadro de búsqueda de Windows el archivo “CODESYS Control Win V3” y cuando termine de cargar lo disminuimos. Esto permite la conexión entre softwares.

```

CODESYS Control Win V3
01630798136244: Cmp=CmpOPCUA, Class=1, Error=0, Info=0, pszInfo= URL: opc.tcp://192.168.56.1:4840
01630798136244: Cmp=CmpOPCUA, Class=1, Error=0, Info=0, pszInfo= URL: opc.tcp://192.168.1.134:4840
01630798136244: Cmp=CmpOPCUA, Class=1, Error=0, Info=0, pszInfo= URL: opc.tcp://127.0.0.1:4840
01630798136244: Cmp=CmpOPCUA, Class=1, Error=0, Info=0, pszInfo= *****
*****
01630798136245: Cmp=CmpApp, Class=1, Error=0, Info=10, pszInfo= Application [<app>Application</app>] started
01630798136245: Cmp=CM, Class=1, Error=0, Info=34, pszInfo= CODESYS Control ready
01630798136267: Cmp=IECVisualization, Class=1, Error=0, Info=0, pszInfo= Visuinitialization starting.
01630798136268: Cmp=IECVisualization, Class=1, Error=0, Info=0, pszInfo= Visuinitialization done.
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= *****
*****
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Web Server
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Root directory : visu
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Host : DESKTOP-C4FCBUL
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Port : 8080
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Secure port : 443
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= Connection type : HTTP_ONLY
01630798136269: Cmp=CmpWebServer, Class=1, Error=0, Info=0, pszInfo= *****
*****
01630798136343: Cmp=CM, Class=2, Error=0, Info=0, pszInfo=!!!! CODESYS Control Service: DEMO mode activated, terminating
in approx. 120 minutes.
01630798136461: Cmp=CmpRouter, Class=1, Error=0, Info=1, pszInfo= Setting router <instance>3</instance> address to <addr
ess>(0301:0001)</address>
01630798136768: Cmp=IECVisualization, Class=1, Error=0, Info=0, pszInfo= Visu_PRG: Creating Client for Extern-ID: 805927
01630798136768: Cmp=OnlineLicenseManager, Class=4, Error=0, Info=0, pszInfo=**** License for TargetVisualization not ins
talled.
01630798136769: Cmp=IECVisualization, Class=1, Error=0, Info=0, pszInfo= Visu_PRG: Creating Client successful for Extern
-ID: 805927 Returned IEC-ID: 0

```

Figura 2-10: Apertura “CODESYS Control Win V3” CODESYS

8º Paso. Iniciamos el configurador OPC iniciando la aplicación OPCConfig.

Esta aplicación se encuentra normalmente en la siguiente dirección.



Figura 2-11: Dirección de la aplicación “OPCConfig”

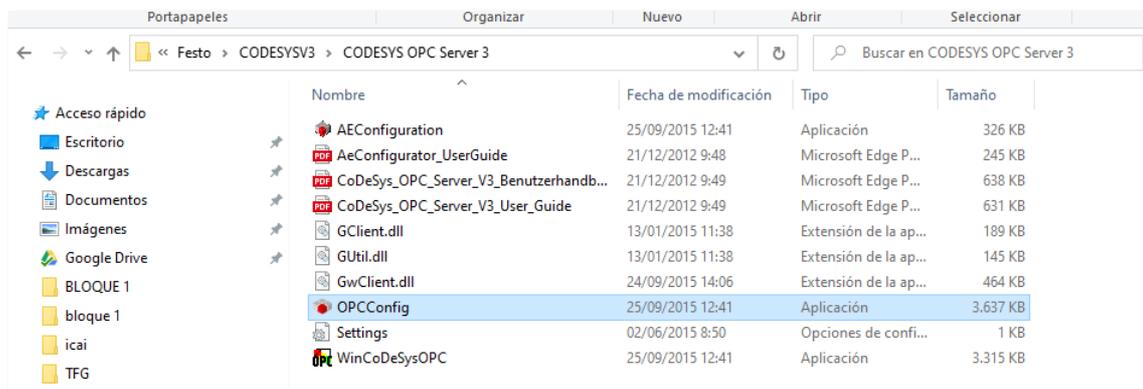


Figura 2-12: Apertura aplicación “OPCConfig”

Al iniciarla, cambiamos el parámetro “Update Rate” del valor inicial a un valor de 10ms. Este parámetro indica como de rápido se actualizan las variables en la base de datos del OPC. Si no se tiene un ordenador potente se puede optar por valores más grandes.

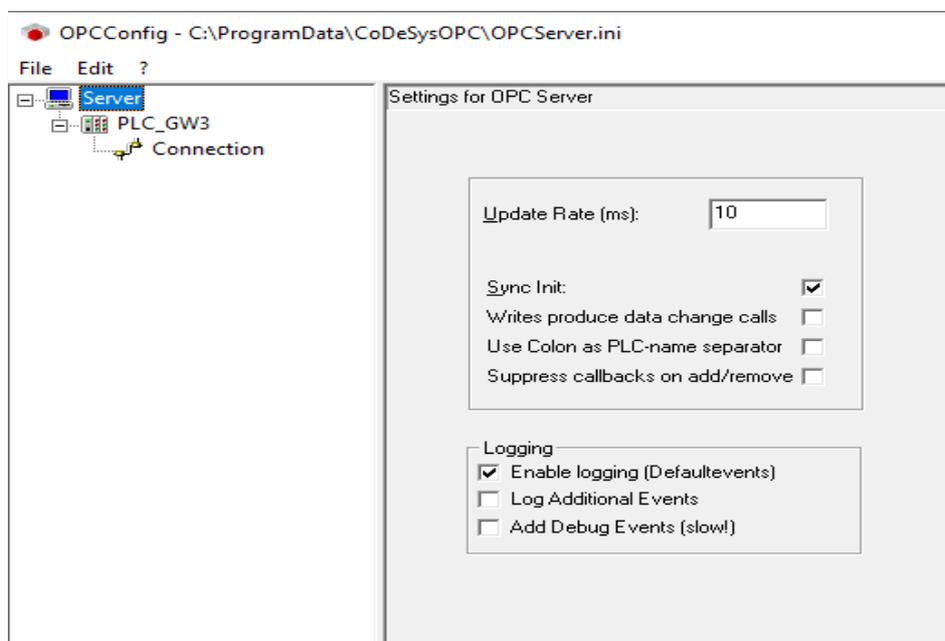


Figura 2-13: Configuración “OPCConfig”

9ºPaso. Creamos un controlador con el lenguaje que se haya elegido con anterioridad para el “PLC_PRG” o se utiliza otro desde otro POU creado siempre que a este se le haya llamado antes. Una vez creado, compilamos iniciando sesión y si no nos da errores buscamos una red accesible y establecemos una ruta activa. Después compilamos de nuevo.

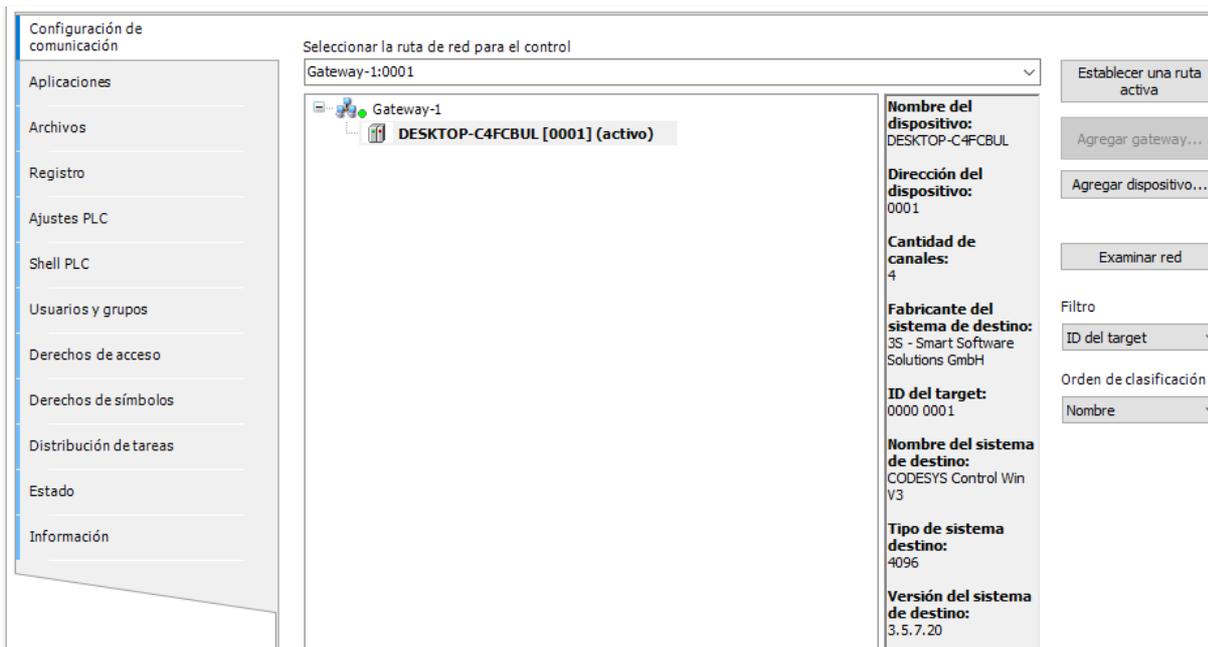


Figura 2-13: Conexión red CODESYS

10º Paso. Iniciamos Factory IO y abrimos una escena o creamos una con los componentes pensados para tu proyecto. Lo más sencillo es llamarlos de igual forma que en CODESYS ya que será más fácil identificarlos cuando haya muchos y ponemos nuestros componentes en modo “No failure” con lo que podremos cambiar sus valores desde el controlador.

11º Paso. Dentro del proyecto de Factory IO seleccionamos FILE y después DRIVERS. Seleccionaremos OPC Client DA/UA que es la conexión elegida para este proyecto. Después seleccionamos configuración y hacemos clic sobre “BROWSE SERVERS” en el que nos aparecerán diferentes servidores. Nosotros seleccionaremos Codesys.OPC.DA y seleccionaremos el límite de variables que se podrán leer del controlador. Después pulsamos sobre “BROWSE” para actualizar los valores y volvemos atrás a “DRIVERS”.

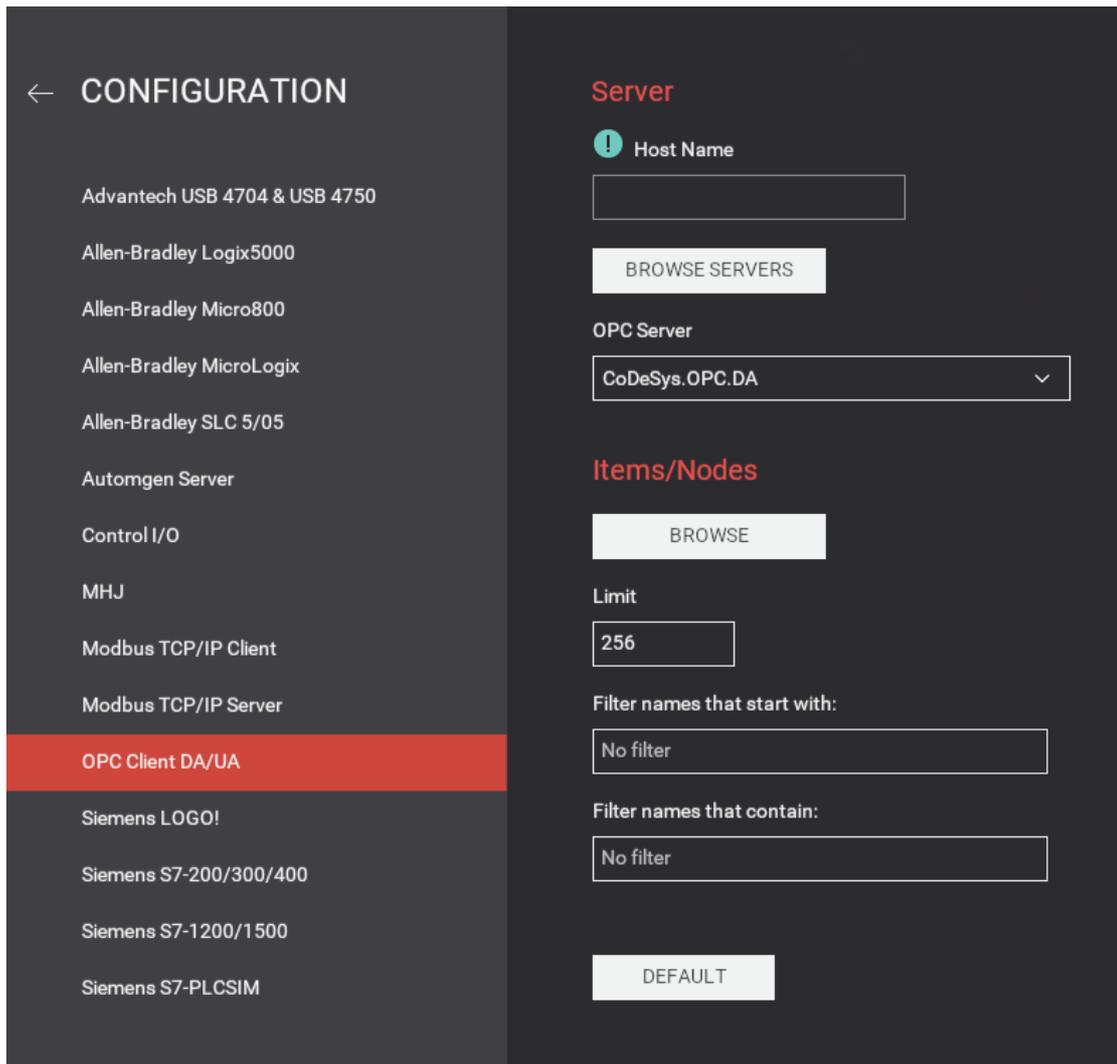


Figura 2-14: Conexión vía OPC FACTORY IO

11º Paso. Nos aparecerán en el medio, las variables creadas en Codesys y que hayan sido ligadas. A la izquierda los sensores creados en Factory IO y a la derecha los actuadores creados en Factory IO. Uniremos las variables de Factory IO arrastrándolas a su correspondiente variable en CODESYS.

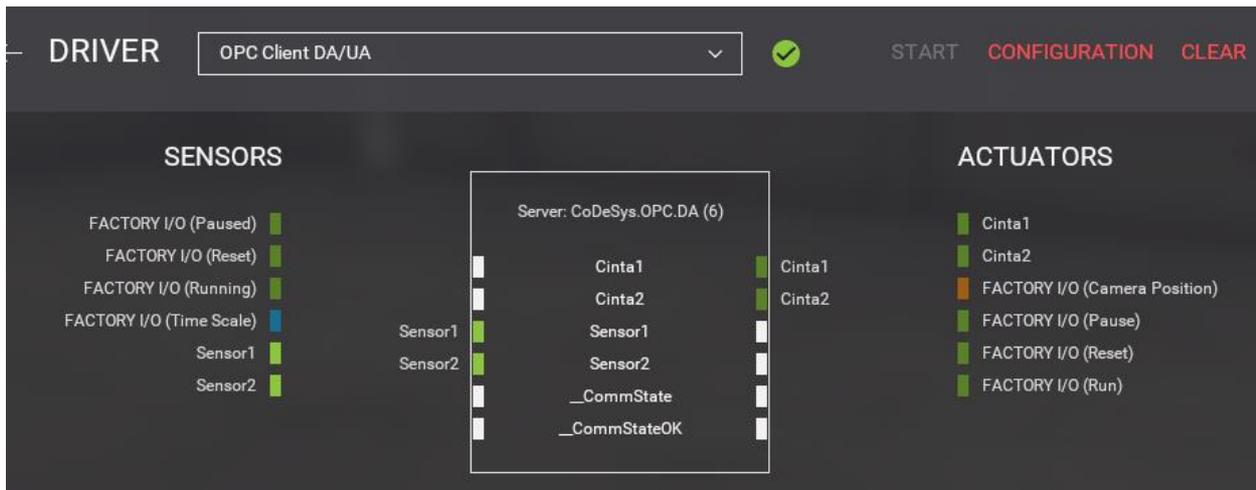


Figura 2-15: Conexión variables FACTORY IO

12º Paso. Compilar el programa en CODESYS y darle a "RUN" en Factory IO, comprobar si funciona y si no lo hace, reparar errores.

3 DISEÑO DE LA PLANTA

En este apartado explicaremos el número de componentes que forman nuestro almacén en las zonas de entrega y recibo, así como su ubicación en estas. En este proyecto una de las cosas más importantes que se han tenido en mente era el uso mínimo de cintas, ya que al no haber cintas largas para utilizar y de la forma requerida, se han necesitado de muchas otras para el correcto funcionamiento y la colocación de estas ha sido importante.

3.1 Descripción de la planta

La idea principal del proyecto ha sido la de diseñar una planta logística que se encargara de recibir mercancía paletizada, almacenarla y suministrarla, todo esto de forma automática y sin necesidad de ningún operario. Cuando comenzamos a realizar el diseño, pudimos ver algunos problemas que teníamos con el software Factory IO. Estos eran debido a que este software está destinado a simulaciones de procesos industriales, líneas de montaje o para el diseño de almacenes de gran tamaño por lo que no se podría comparar con la idea de un almacén automatizado de hospital donde los componentes son mucho más pequeños y donde no se dispone de tanto espacio. Aún así, decidimos utilizarlo para nuestro proyecto utilizando los componentes de los que dispoíamos para intentar diseñar de la mejor manera posible lo que teníamos pensado.

Se diseñó una planta con tres zonas diferenciadas, la primera zona, la encargada de recibir los tres tipos de “medicamentos” paletizados y de distribuirlos a la siguiente zona. La segunda zona, encargada de almacenar los productos, recibe los medicamentos mezclados a través de la misma cinta proveniente de la primera zona, después, mediante un sensor de visión, los distribuye a los diferentes almacenes según el color del producto. Una vez almacenados, se encarga de recibir las notificaciones de pedidos y de enviar los medicamentos a la siguiente zona mediante otro conjunto de cintas de rodillos. En la tercera zona, la zona de envío, se reciben los medicamentos de nuevo mezclados y se encargan de distribuirlos a cada una de las plantas según la cantidad requerida de cada tipo de producto.

Como se ha dicho antes, este es un proyecto escalable a cada tipo de empresa o industria. La separación entre zonas es simbólica, intentando representar tres zonas completamente independientes y que en la realidad cada una de ellas estaría en diferentes salas.

Todo el diseño de la planta se ha intentado hacer de la mejor forma posible para que, aunque pudiera llegar a ser más compleja, en lo que a programación se refiera, se disminuyeran los costes y se aprovechara mejor el espacio de la planta.

En la imagen siguiente se puede observar una perspectiva de la planta:



Figura 3-1: Vista de la planta FACTORY IO

3.2 Componentes

En esta sección se darán a conocer cada uno de los componentes que forman nuestra simulación, su función, su número dentro de esta y el nombre de estos componentes junto con el apodo que se les ha dado para la sencillez del programa.

3.2.1 Maquinaria

•Cintas automáticas:

Habrán tres tipos de cintas automáticas:

-Cinta de rodillos.

Nombre técnico en Factory IO: Roller Conveyor

Función: Son cintas que generan movimiento lineal a través del giro de sus múltiples rodillos. Están hechas para mover sobre todo palés o cajas de transporte ya que otros objetos más pequeños pueden caerse entre los rodillos.

Tipos: Puede haber tanto de 6 metros, de 4 metros, de 2 metros y una curva. En nuestra simulación utilizaremos todas.

Cantidad: 12 cintas de rodillos de 6 metros, 9 de 4 metros, 8 de 2 metros y 10 curvas

-Cinta transportadora:

Nombre técnico en Factory IO: Belt Conveyor

Función: Son cintas que generan movimiento lineal a través del movimiento de una cinta que avanza gracias al giro de un motor. Permiten el movimiento de cualquier tipo de objeto ya que no tienen zonas donde se puedan caer y a diferencia de las cintas de rodillo pueden desplazar objetos, aunque estos sobresalgan por los laterales.

Tipos: Puede haber tanto de 6 metros, de 4 metros, de 2 metros y curvas. En nuestra simulación utilizaremos de 6 metros y de 4.

Cantidad: 2 cintas transportadoras de 6 metros y 1 de 4 metros.

-Cinta de carga o descarga:

Nombre técnico en Factory IO: Loading Conveyor

Función: Son cintas que generan movimiento lineal de la misma forma que las cintas de rodillos, pero en este caso están hechas para ser cintas de carga y descarga de palés ya que tiene una abertura en el interior donde pueden entrar brazos mecánicos para recogerlos.

Tipos: Solo hay uno.

Cantidad: 6 Cintas de carga o descarga



Figura 3-2: Cintas de rodillos FACTORY IO

Figura 3-3: Cintas transportadoras FACTORY IO



Figura 3-4: Cintas de carga o descarga FACTORY IO

•**Cinta inclinada:**

Nombre técnico en Factory IO: Chute Conveyor

Función: Se encarga de enviar productos desde las cintas transportadoras. Solo tiene una inclinación y una longitud por lo que no se puede utilizar con otro tipo de cintas.

Tipos: Solo hay uno

Cantidad: 3 cintas inclinadas

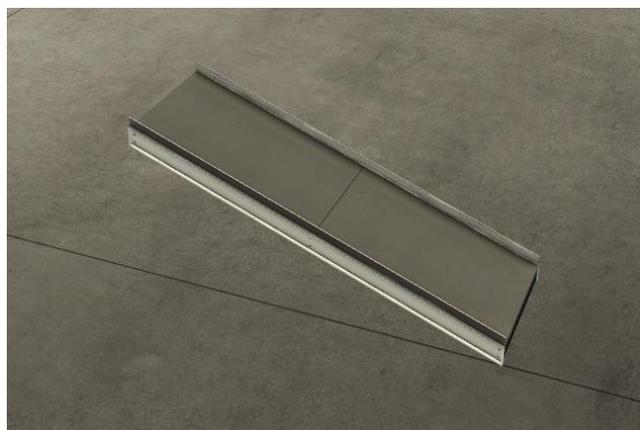


Figura 3-5: Cinta inclinada FACTORY IO

•**Objetos de transporte:**

Habrán dos tipos:

-Pallet:

Nombre técnico en Factory IO: Pallet

Función: Sirven para transportar los medicamentos durante su recogida hasta la zona de entrega siendo eliminado antes de la entrega final.

Tipos: Puede haber tanto pallets normales como pallets cuadrados. En nuestra simulación utilizaremos

los normales debido a que los cuadrados tenían peor equilibrio en diferentes zonas de la planta por lo que decidimos no utilizarlos. Ambos pesan 20 kg.

Cantidad: No definida

-Materiales crudos:

Nombre técnico en Factory IO: Raw Material

Función: Por ejemplo, representarán los diferentes medicamentos que serían el objeto de entrega en las diferentes salas del hospital.

Tipos: Podemos encontrarlos en color azul de un peso de 8 kg, en color verde de un peso de 8 kg y en color plateado de un peso de 10 kg.

Cantidad: No definida

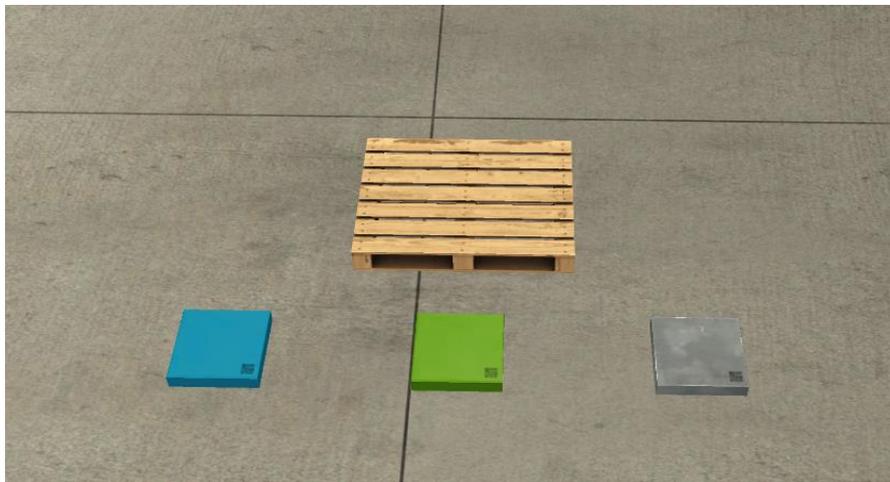


Figura 3-6: Cintas de rodillos FACTORY IO

•Base giratoria:

Nombre técnico en Factory IO: Turntable

Función: Se encargan de recibir los pallets con productos y de girarlos para enviarlos en diferentes direcciones Pueden hacer un giro de 90° y tienen movimiento lineal en dos sentidos.

Tipos: Solo hay uno

Cantidad: 5 Bases giratorias



Figura 3-7: Base giratoria FACTORY IO

•**Brazo giratorio:**

Nombre técnico en Factory IO: Pivot Arm Sorter

Función: Sirven para desplazar cualquier tipo de objeto en otra dirección. Pueden accionar unas cintas que hacen que el objeto se desplace de forma más rápida.

Tipos: Solo hay un tipo.

Cantidad: 3 Brazos giratorios



Figura 3-8: Brazo giratorio FACTORY IO

•**Grúa recogedora de productos:**

Nombre técnico en Factory IO: Two-Axis Pick & Place

Función: Sirven para desplazar objetos de una zona a otra. En este proyecto se encarga de separar los “medicamentos” y los pallets para una entrega más sencilla.

Tipos: Solo hay un tipo.

Cantidad: 1 Grúa recogedora de productos

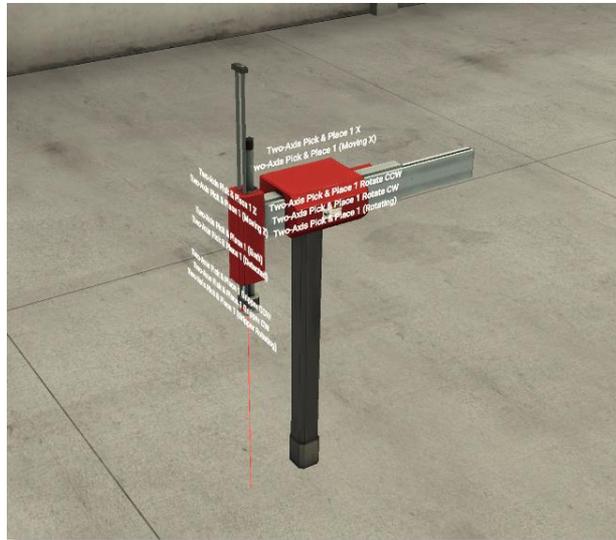


Figura 3-9: Grúa recogedora de productos FACTORY IO

•**Estantería:**

Nombre técnico en Factory IO: Rack

Función: Sirve para almacenar cualquier objeto que esté sobre un pallet. Tiene 6 filas y 3 columnas.

Tipos: Solo hay un tipo.

Cantidad: 9 Estanterías

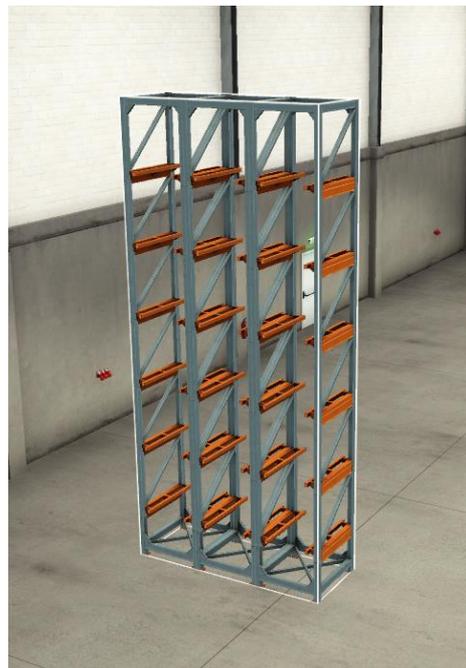


Figura 3-10: Estantería FACTORY IO

•**Apiladora de productos:**

Nombre técnico en Factory IO: Stacker Crane

Función: Se encarga de transportar productos sobre pallets para almacenarlos en las estanterías o para extraerlos de las mismas. Cada apiladora puede almacenar productos en 3 estanterías juntas ya que tienen 56 movimientos diferentes para cada una de las cavidades de las 3 estanterías y otro más para el de recogida.

Tipos: Solo hay un tipo.

Cantidad: 3 Apiladoras de productos



Figura 3-11: Apiladora de productos FACTORY IO

•Elevador:

Nombre técnico en Factory IO: Elevator

Función: Se encarga de transportar verticalmente objetos. Dispone de una cinta de rodillos que se encarga de conectar con otras cintas a diferente altura.

Tipos: Solo hay un tipo.

Cantidad: 1 elevador.



Figura 3-12: Elevador FACTORY IO

•Elementos de separación:

Hay tres tipos

-Valla metálica de seguridad

Nombre técnico en Factory IO: Safeguard (L)

Función: Sirven para separar cada una de las zonas que hemos creado para nuestra planta. En este caso separan la zona de recepción de productos, la zona del almacén y la zona de entrega.

Tipos: Solo hay un tipo.

Cantidad: 22 vallas metálicas de seguridad

-Valla metálica de seguridad con abertura

Nombre técnico en Factory IO: Safeguard (I)

Función: Sirven también para separar zonas, pero permiten pasar productos a través de una abertura que se encuentra en el centro de la misma.

Tipos: Solo hay un tipo.

Cantidad: 2 vallas de seguridad con abertura

-Puerta de seguridad

Nombre técnico en Factory IO: Safety Door

Función: Sirven para conectar las diferentes zonas de nuestra planta para que los operarios puedan acceder a cada una de ellas

Tipos: Solo hay un tipo.

Cantidad: 1 Grúa recogedora de productos



Figura 3-13: Elementos de separación FACTORY IO

•Elementos de control:

Como elementos físicos del proyecto, hemos utilizado estos componentes para el control de los modos de

funcionamiento del programa. De esta forma hemos utilizado tanto componentes estructurales como componentes de control. Estos serán:

-Cuadro de distribución eléctrico

Nombre técnico en Factory IO: Electric Switchboard.

Función: Se encarga de ser la zona donde ubicar los diferentes componentes de control.

Tipos: Solo hay uno.

Cantidad: 1 Cuadro de distribución eléctrico

-Columna

Nombre técnico en Factory IO: Column.

Función: Se encarga de ser la estructura la cual sujeta el cuadro de distribución eléctrico.

Tipos: Solo hay uno.

Cantidad: 1 Columna

-Parada de emergencia

Nombre técnico en Factory IO: Emergency Stop.

Función: Pulsador encargado de informar de una parada de emergencia.

Tipos: Solo hay uno

Cantidad: 1 Parada de emergencia

-Botón

Nombre técnico en Factory IO: Button.

Función: Se encarga de informar del diferente modo de funcionamiento del programa.

Tipos: Hay tres tipos, el Start Button o como lo hemos nombrado Botón Automático, el cual es verde. El Stop Button o Botón de parada, el cual es rojo. Y el Reset Button o Botón de rearme, el cual es amarillo

Cantidad: 3 Botones

-Pantalla digital

Nombre técnico en Factory IO: Digital Display.

Función: Se encarga de mostrar un contador por pantalla. En el proyecto no se utilizará, pero pienso que podría ser útil mantenerlo.

Tipos: Solo hay una

Cantidad: 1 Pantalla digital



Figura 3-14: Componentes de control FACTORY IO

3.2.2 Sensores

Factory IO permite utilizar diferentes sensores para el control de los materiales. En este proyecto utilizaremos dos de ellos:

-Sensor láser

Nombre técnico en Factory IO: Sensor

Función: Detectan objetos que se encuentran en su rango. Se utilizan para saber cuándo un objeto de transporte pasa por la zona donde está colocado.

Tipos: Podemos encontrar diferentes tipos de sensores láseres. Los dos que hemos utilizado en este proyecto son los señores fotoeléctricos que detectan un objeto gracias a que frenan el láser del que disponen el cual tiene un alcance de 0 a 1,6 metros, y los sensores retro reflectantes que detectan cuando un objeto pasa por su zona de alcance, la cual es de 0 a 6 metros, debido a que su haz de luz no llega a su reflector.

Cantidad: 2 sensores retro reflectantes y 67 fotoeléctricos.

-Sensor de visión:

Nombre técnico en Factory IO: Vision Sensor

Función: El sensor de visión reconoce diferentes colores y las formas de los productos dentro de su rango de detección que es de 0,3 a 2 metros. Este sensor crea un valor para cada tipo de producto.

Nosotros trabajaremos con los valores 1, 4 y 7 que se relacionan con los tres tipos de materiales crudos vistos con anterioridad. 1 para el azul, 4 para el verde y 7 para el plateado. Estos sensores los colocaremos en una estructura hecha para ellos denominada en Factory IO como Bracket, la cual se puede ver en la IMAGEN.

Tipos: Solo hay un tipo

Cantidad: 2 sensores de visión con su respectiva estructura

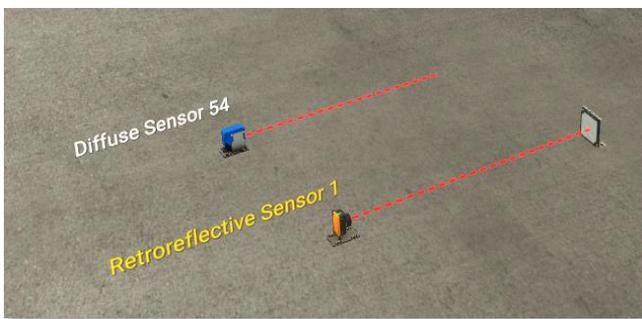


Figura 3-15: Sensores FACTORY IO

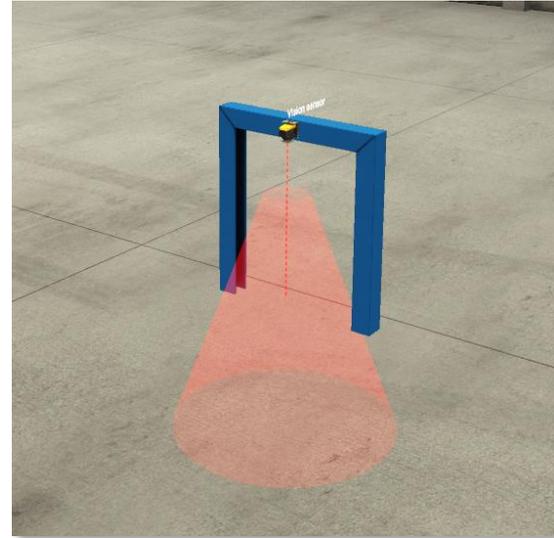


Figura 3-16: Sensor de visión FACTORY IO

3.3 Requerimiento de espacio y ubicaciones físicas

Uno de los problemas con los que nos topamos nada más empezar el proyecto fue la escasez de material con el que contábamos para hacer la simulación 3D.

Para observar la planta de manera óptima, se ha creado una perspectiva en planta por completo y de cada una de las diferentes zonas en las que hemos dividido ésta misma. La primera zona, la zona de recepción de productos se representa con la letra “R”. La segunda zona, la zona de almacenaje, se representa con la letra “A”. Y, por último, la tercera zona o zona de envío de productos, la cual se representa con la letra “E”.

Esta distribución por zonas de la planta se debe a que, en la realidad, para el mejor funcionamiento del sistema logístico, se repartirían estas zonas en diferentes salas físicas. Por ejemplo, la zona de almacenaje en un hospital debe estar aislada del resto de zonas ya que es un lugar donde debe haber una higiene óptima y donde no puede entrar cualquiera, como sería el caso de la zona de recepción de productos donde se encontrarían diferentes operarios para la carga en cintas de los productos o la zona de envío de pedidos donde además de médicos podrían encontrarse pacientes.

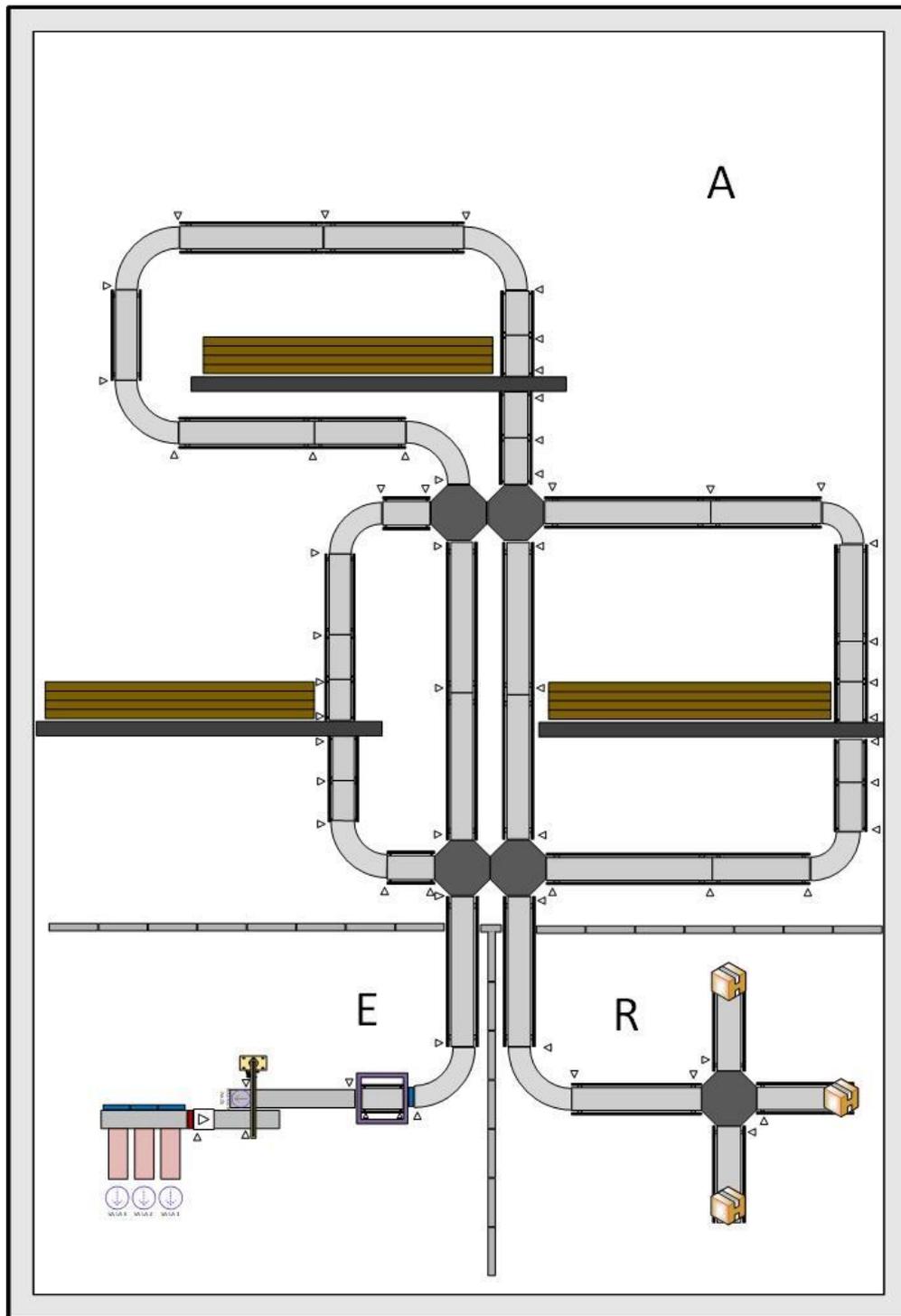


Figura 3-17: Mapa de nuestra planta VISIO

3.3.1 Zona de recepción de productos

La zona de recepción de productos, en la simulación, se representa de forma más sencilla a como sería en la realidad. En la imagen se muestran mediante cajas de diferente color, la zona donde se cargarían los productos paletizados, en las cuales debería de haber maquinaria de carga y descarga así como una zona amplia de llegada de camiones con los diferentes medicamentos. Esta sería sin lugar a duda la zona más amplia de la planta si estuviera a escala real.

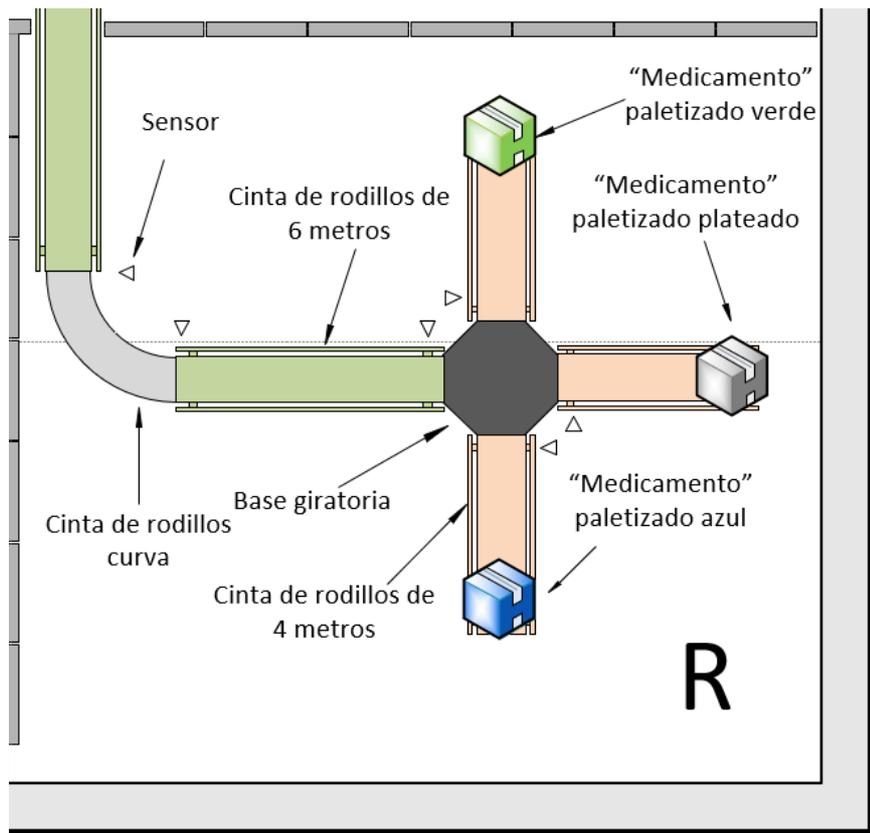


Figura 3-18: Zona de recibo de productos VISIO

Para entender mejor la ubicación de los componentes físicos dentro de la zona de Recibo de Productos, mostramos a continuación una imagen en perspectiva donde encontramos las diferentes cintas y componentes que la componen y con la cual se es más fácil el entendimiento de la distribución.

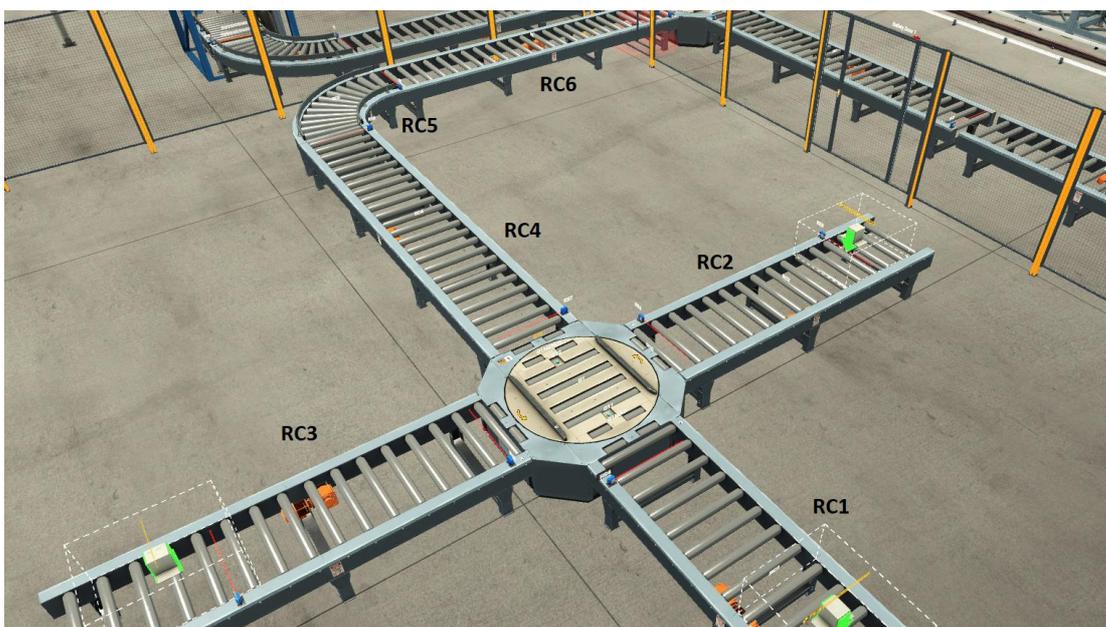


Figura 3-19: Zona de recibo de productos FACTORY IO

3.3.2 Zona de almacenaje

Esta zona es la dedicada a el almacenamiento por minutos horas o días de los productos que han sido recibidos de los proveedores y todavía no han sido enviados al cliente.

Si se enfocara por ejemplo en el almacén de un hospital, la zona de almacenaje de productos sería, al contrario que la zona de recepción de pedidos, la zona más pequeña de todas, al contrario de lo que se muestra en la simulación, donde es la zona más amplia. Esta zona en la realidad sería pequeña debido a que los productos que se guardan en ella son bastantes pequeños y se apilan de forma óptima en cada cajón de las diferentes estanterías dispuestas para el almacenaje de los medicamentos.

A continuación, se muestra la planta de la zona de almacenaje donde se pueden observar la ubicación de las diferentes estanterías junto con la forma de sus cintas de entrada y salida.

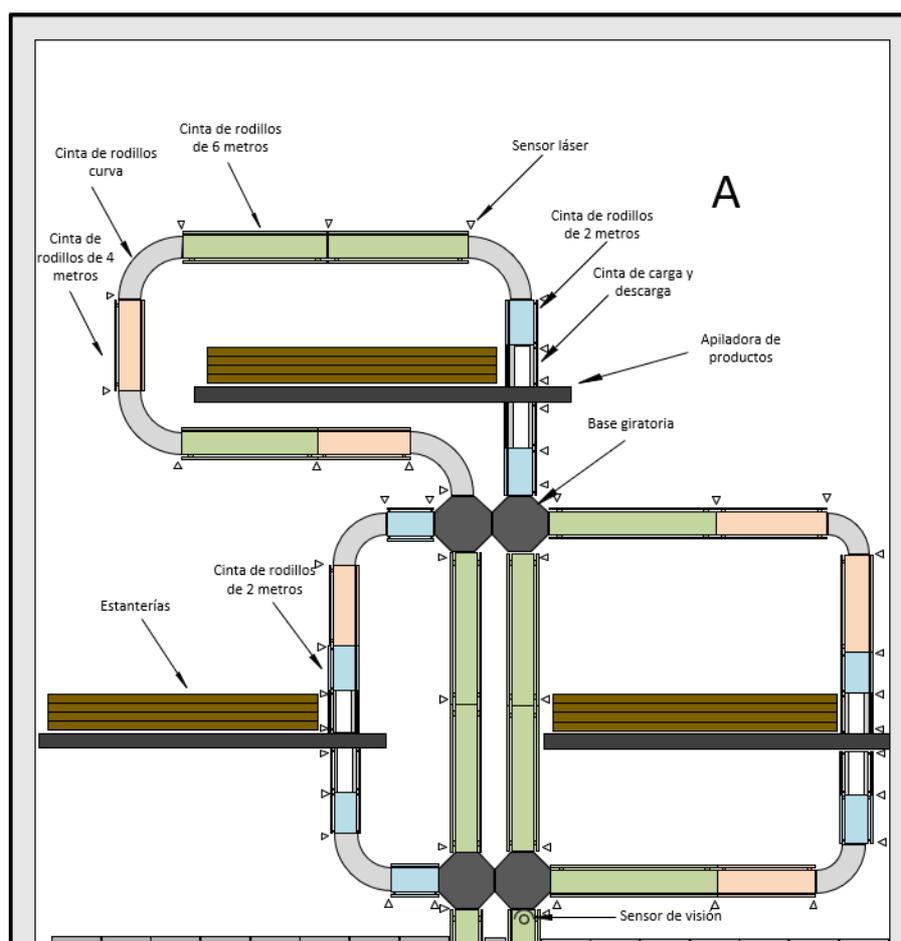


Figura 3-20: Zona de recibo de almacenaje VISIO

Las imágenes siguientes muestran, en perspectiva, las diferentes partes de la zona de almacenaje. Estas se diferencian según el tipo de producto que almacena.

La primera imagen sería la correspondiente a la zona de almacenaje del producto verde, la siguiente sería la correspondiente a la zona de almacenaje del producto azul y la último sería la correspondiente a la zona de almacenaje del producto plateado.



Figura 3-21: Zona de almacenaje de productos verdes FACTORY IO

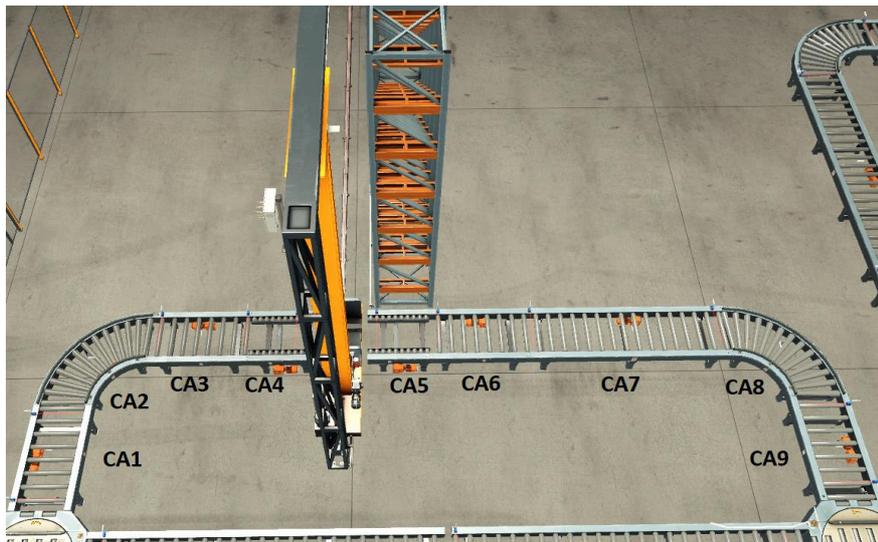


Figura 3-22: Zona de almacenaje de productos azules FACTORY IO

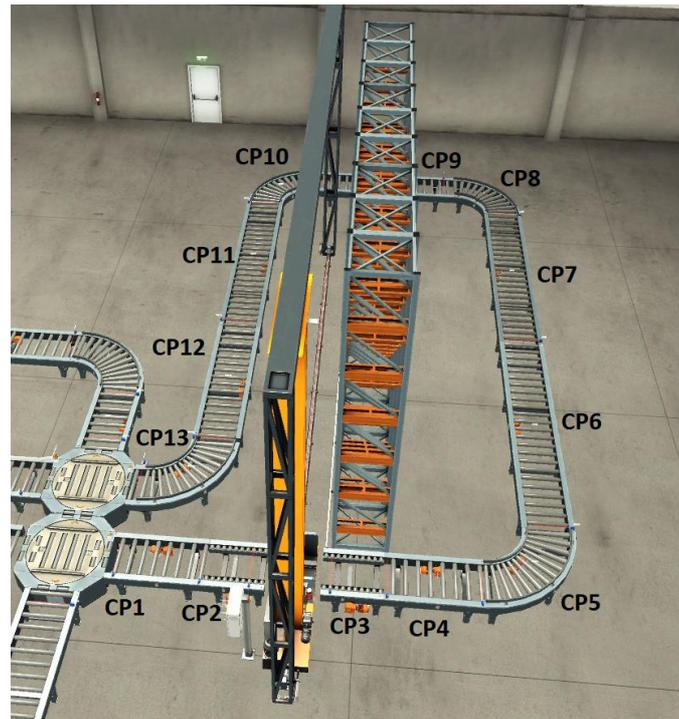


Figura 3-23: Zona de almacenaje de productos plateados FACTORY IO

3.3.3 Zona de envío de productos

La última zona, o zona de envío de productos, sería otra zona que en la realidad aumentaría bastante su tamaño. Esto es debido a que las diferentes salas dentro de un edificio no se situarían tan cercanas la una a la otra como se muestra en el proyecto. Por ejemplo, en un hospital, las diferentes salas podrían estar unas en zonas de atención de urgencias, otra en zonas de quirófanos o en zonas de laboratorios.

Otra de las características que la haría aumentar de tamaño sería la utilización del elevador, el cual permite la posibilidad de poder distribuir productos a otras plantas. En este proyecto no se han colocado más plantas debido a que se superaban el número de componentes que dejaba enlazar Factory IO.

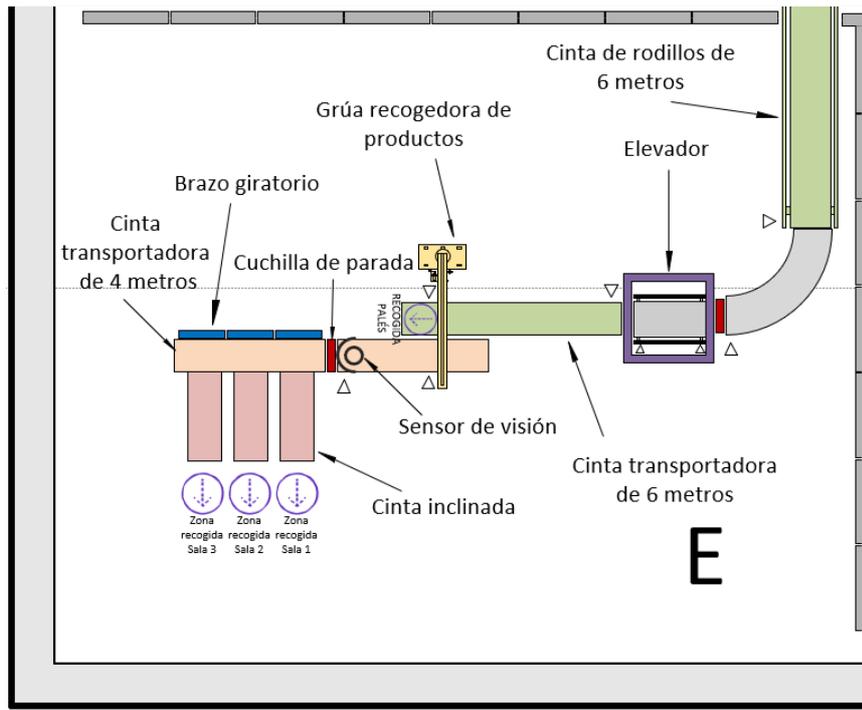


Figura 3-24: Zona de envío de productos VISIO

En la siguiente imagen en perspectiva se pueden observar las diferentes posiciones de los componentes destinados al envío de los productos y su localización.

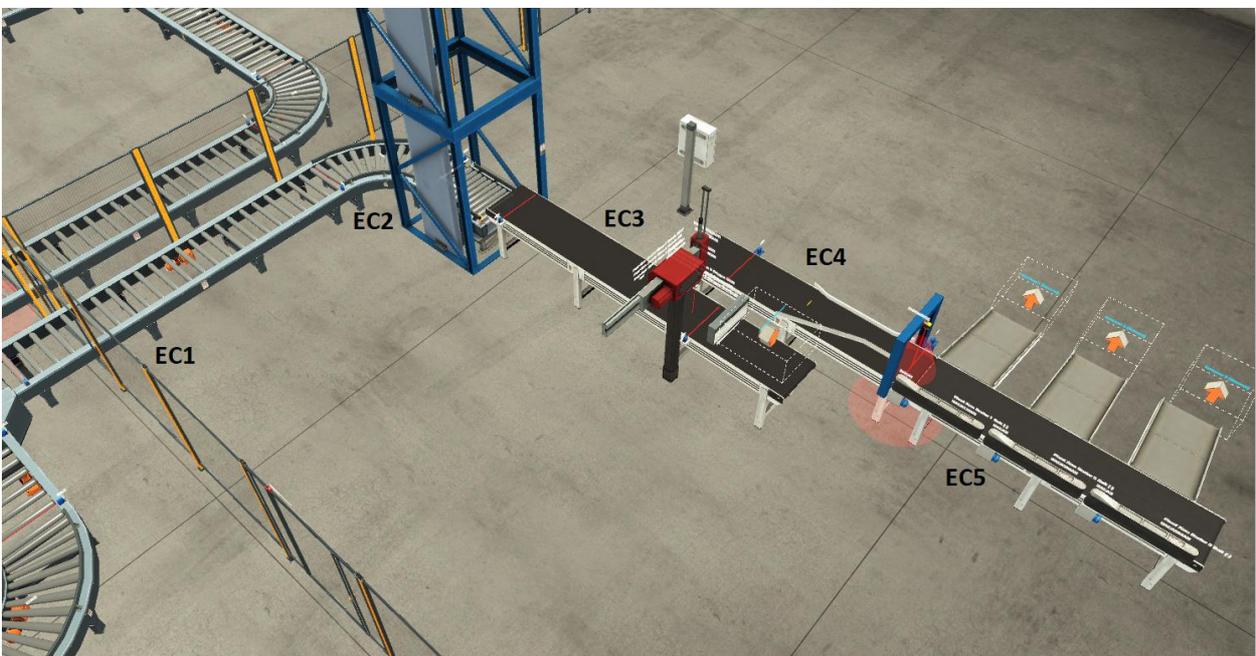


Figura 3-25: Zona de envío de productos FACTORY IO

3.4 Limitaciones y mejoras

Por supuesto, las principales limitaciones que hemos tenido en las ubicaciones físicas de este proyecto han sido las dimensiones de la diferente maquinaria cuyo uso está permitido en FACTORY IO. Como se ha dicho con anterioridad, el objetivo principal del proyecto era el de realizar la automatización de un almacén de una farmacia hospitalaria, pero para ello se hubiera necesitado tanto cintas transportadoras como estanterías especializadas, de un tamaño más pequeño y con mayor precisión.

Para realizar esta comparativa se muestran a continuación diferente maquinaria utilizada en este proyecto y la usada en una farmacia automatizada.



Figura 3-26: Apiladora de productos
FACTORY IO



Figura 3-27: Robot automatizado
FARMACIA REAL

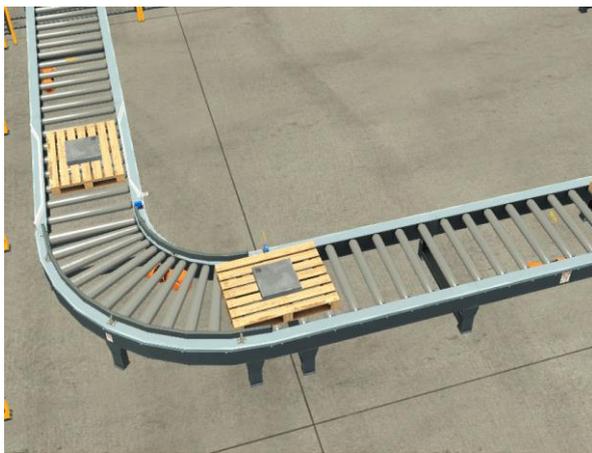


Figura 3-28: Cinta automática
FACTORY IO



Figura 3-29: Cinta automatizada
FARMACIA REAL

Otra limitación en el proyecto ha sido la de no poder ubicar cada zona de este en diferentes edificios o salas como sería lo normal si se implantara en alguna empresa. Esto es debido al tamaño de los productos y a la necesidad de tener zonas especializadas para cada momento de la cadena logística. El problema con esta limitación se ha intentado disminuir utilizando las vallas explicadas con anterioridad para la distinción de las zonas.

También se ha encontrado un problema con la concordancia de tamaños de las cintas y el resto de maquinaria ya que estas tenían tamaños específicos de 2, 4 y 6 metros que no podían variarse. Debido a esto surgían problemas en algunas zonas con pallets que se desequilibraban por no encontrar puntos de apoyo en el transporte. Debido a esto se decidió por utilizar los “FREE ROLLER” los cuales son rodillos individuales que pueden completar estos huecos

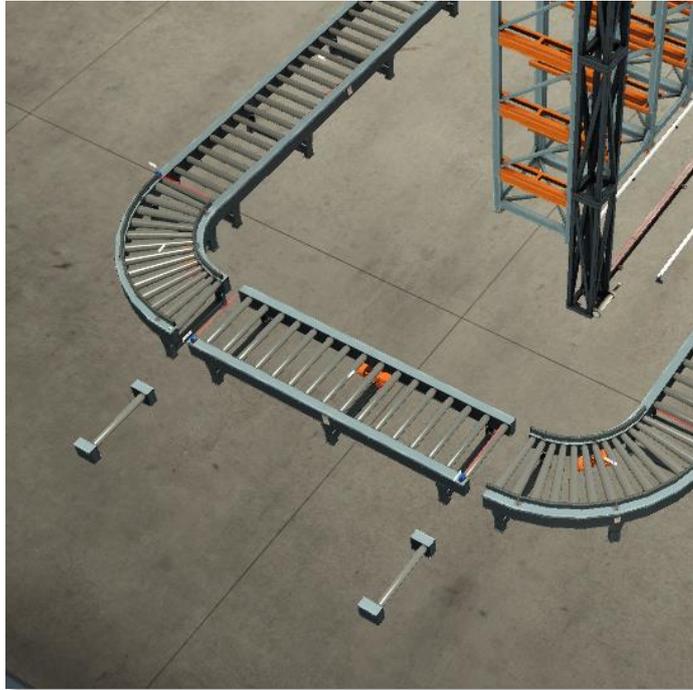


Figura 3-30: Problemas de cintas FACTORY IO

4 DESCRIPCIÓN DE LAS VARIABLES Y DE LA PROGRAMACIÓN DEL CONTROLADOR

En este capítulo trataremos de explicar el diseño de la programación del controlador, así como las diferentes variables creadas para el control de la maquinaria y para el seguimiento del número de productos tanto almacenados en sus correspondientes estanterías como los pedidos que se realicen por el personal médico.

4.1 Descripción de variables

En este apartado explicaremos tanto el porqué del nombre de las variables como la función que tienen dentro del programa. Para ellos primero distinguiremos dos tipos de variables, las externas y las internas.

4.1.1 Variables externas

Las variables externas serán aquellas las cuales tendrán una salida o entrada del almacén logístico físico. Es decir, estarán encargadas de un sensor o actuador dentro del proyecto.

En primer lugar, y para entender mejor como han sido llamados los actuadores y sensores de cada zona logística, explicaremos la forma en la que han sido llamados. De esta forma cualquier operario que notara un problema en alguna cinta podría, de forma rápida, comunicarla al encargado del almacén con el nombre de la variable que muestra el problema para mejorar el tiempo de resolución.

Cintas

Para todo el conjunto de cintas durante todo el proyecto, se han establecido nombres que las identifica según la zona en las que se ubican.

De esta forma los nombres distinguiremos tres zonas para las cintas generales, las cuales son aquellas en las que puede haber cualquiera de los tres tipos de productos, y también distinguiremos tres zonas según los tres tipos de productos que tenemos para su zona de almacenaje.

Con esto, los actuadores que mueven las cintas generales comenzarán con la letra “R”, “A” o “E”, según si está ubicada en la zona de “Recepción de productos”, en la de “Almacenaje de productos” o en la de “envío de productos”. A esta letra le seguirá la letra “C”, la cual distinguirá que se trata de una cinta. Y a esta le seguirá un número que identificará el orden en las que están colocadas dentro de la zona.

Ej: “EC1” será el actuador asignado a la cinta general que se encuentra en la zona de “Envío de productos” en primer lugar.

Para las cintas ubicadas en las zonas de almacenaje de cada tipo de producto el nombre del actuador comenzará por la letra “C”, identificativa de que se tratará de una cinta. A esta le seguirá una letra según el tipo de productos que se encuentra en esta zona de almacenaje, por ejemplo, para la zona de productos verdes le seguirá la letra “V”, para la zona de productos azules le seguirá la letra “A” y para la zona de productos plateados, le seguirá la letra “P”. El nombre del actuador continuaría con un número según el orden en el que se mueven los productos por ellas.

Ej: “CV2” es la segunda cinta que se encuentra el producto al entrar en la zona de almacenaje verde.

Los sensores se colocan al inicio y al fin de cada cinta para saber la ubicación de las diferentes cajas y para establecer contadores en cada una de ellas que nos permita conocer la cantidad de productos que hay dentro de una cinta para poder saber cuándo deben funcionar y cuando pueden estar apagadas. A los sensores ubicados a través de las diferentes cintas, se les denomina de igual forma que a los actuadores. Aunque hayamos ubicado sensores en todas las uniones de cada cinta, solo utilizaremos los sensores iniciales y finales de los diferentes tramos para que funcionen las diferentes cintas unidas como una sola.

Ej: “AS2” es el segundo sensor encontrado en una cinta general de la zona de almacenaje y el “SV6” es el sexto sensor que nos encontramos en la zona de almacenaje verde y se utilizan en nuestra programación ya que son el inicio de un tramo de cintas. Tanto el sensor “SV3”, el “SV4” o el sensor “SV5” mostrados en la siguiente imagen, aunque los hayamos colocado en la simulación, no los utilizamos en el controlador debido a que se ubican entre medias de un tramo de cintas.

Todas estas variables de tipo serán de tipo BOOL.



Figura 4-1: Variables cintas FACTORY IO

Bases giratorias

Estas bases giratorias están formadas por tres actuadores y cuatro sensores. Tanto los nombres de los actuadores como de los sensores comienzan con la letra “G”, seguida de un número que diferencia cada una de las cinco bases giratorias. La terminación será la que diferencia cada una de las funciones. Para los actuadores, si no tiene terminación, será el encargado del giro de 90°, si el nombre termina en “MAS”, será el encargado del movimiento de los rodillos en un sentido y si la terminación es “MENOS”, será el encargado del movimiento en el otro sentido.

Para los sensores, si termina en la terminación “1”, será el sensor encargado de determinar la posición en un extremo de la base, si la terminación es “2”, será el encargado de determinar la ubicación en el otro extremo de la base, si es “R”, el sensor determina que la base no está girada y si es “D”, determina que está girada.

Todas estas variables de tipo BOOL.



Figura 4-2: Variables bases giratorias FACTORY IO

Apiladora de productos.

La apiladora de productos dispone de cuatro actuadores y de cinco sensores. Estos se llamarán de igual forma en las diferentes apiladoras exceptuando la letra que diferencia a cada producto.

Se explicarán a partir de aquí el funcionamiento de cada uno de ellos para la apiladora de productos verdes.

Los actuadores de tipo BOOL son, AVMENOS, el cual se encarga de desplazar los brazos de la apiladora en un sentido, AVMAS, que se encarga de desplazarlas en el otro sentido y AVSUBEBAJA, el cual se encarga de subir los brazos para extraer los palés de las diferentes zonas o bajarlos para almacenarlo o enviarlos. El otro actuador ALMACENVCD es un actuador de tipo INT el cual tiene un rango desde 1 hasta el 54 para cada una de las baldas de cada estantería y el número 21474 que se encarga de regresar la apiladora a su posición inicial.

Los sensores serán todos de tipo BOOL. De esta forma, SAVX se encargará de mostrar si la apiladora se está moviendo en el sentido de la x, SAVZ si se está moviendo en el sentido de la z, AVEXTMAS dirá si los brazos están extendidos completamente en el sentido del actuador AVMAS y el sensor AVEXTMENOS se encargará de decir si están extendidos hacia el sentido del actuador AVMENOS. El último sensor será AVCENTRADO que se encargará de decir si los brazos están en su situación inicial que es la de no extendidos.



Figura 4-3: Variables apiladoras de productos FACTORY IO

Elevador

El elevador dispone de tres actuadores, dos de tipo BOOL y uno de tipo REAL. ALTURAELEVADOR será el de tipo REAL y variará de 0.0 a 10.0 según la altura que se desea. En nuestro caso se desplazará desde la altura 0.0 a la altura 0.6. De los otros dos actuadores solo utilizaremos ELEVADORMENOS que se encargará de desplazar los productos hacia el interior del elevador y más tarde hacia la zona de envío de productos.

Consta de dos sensores retro reflectivos de tipo BOOL pero solo utilizaremos el primo de ellos el cual lo hemos llamado ES3.



Figura 4-4: Variables elevador FACTORY IO

Grúa recogedora de productos

Esta grúa consta de diferentes sensores y actuadores, aunque no todos ellos nos han sido útiles para este caso.

En el caso de los actuadores, los cuales son todos de tipo BOOL, hemos utilizado tres de ellos, BRAZOEXP que se encarga de extender el brazo o contraerlo, BRAZOBJADO que se encarga de bajar o subir el brazo, y BRAZOCOG que se encarga de agarrar la pieza.

Para los sensores, contamos con tres de ellos. BRAZOMOVX que nos dice si el brazo se está expandiendo o contrayendo, BRAZOMOVZ que se encarga de decir si el brazo está bajando o subiendo y BRAZOPIEZA que se encarga de decirnos si se detecta un producto en el extremo del brazo.

En la grúa se encontró un problema con el delay que se originaba en el valor de las variables desde que se cambiaban en el programa de "CODESYS" hasta que llegaban al programa de "FACTORY IO". A este problema se le sumaba también otro con el movimiento del producto encima del pallet. En otras zonas se notaba de forma más insignificante, pero en este apartado, que el producto se encontrara debajo de la grúa a la hora de la recogida era indispensable. Para solucionar esto se colocaron los sensores "P1", "P2", "P3" Y "P4" y se programó para que al final del movimiento de la cinta, si el producto final se encontraba en un lugar diferente a la zona situada justo debajo de la grúa, la cinta "EC3", la cual tiene dos direcciones, cambiaba el sentido hasta que el producto se encontrara justo en el centro.

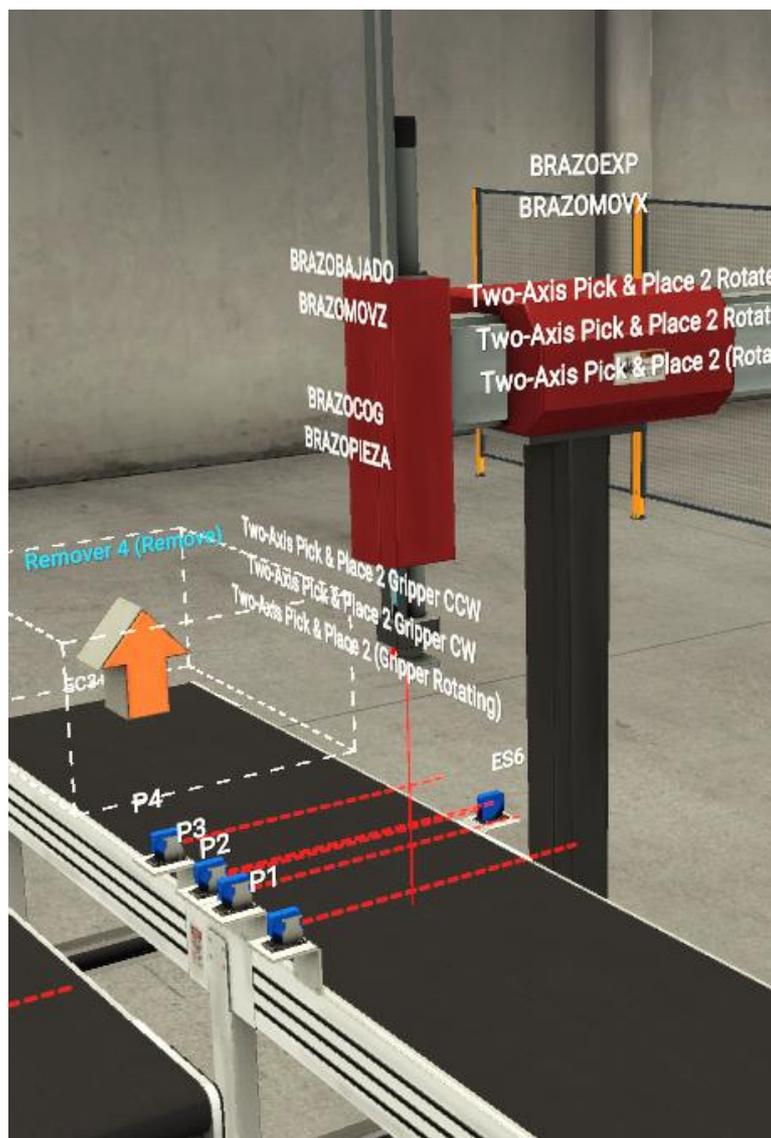


Figura 4-5: Variables grúa recogedora de productos FACTORY IO

Emisor y eliminador de productos

Estos son los encargados de hacer aparecer y desaparecer los productos paletizados y no paletizados.

Dentro de los emisores, tendremos tres variables asociadas a cada uno de ellos: una denominada “PLATEADORECIBIDO” la cual se encarga de hacer aparecer los productos plateados, “VERDERECIBIDO” la cual se encarga de los productos verdes y “AZULRECIBIDO” la cual se encarga de los productos azules. Estas estarán activas solo cuando se requiera.

Luego, para hacer desaparecer los productos ya entregados tendremos tres eliminadores de productos denominados “SALA1ENTREGADO”, “SALA2ENTREGADO” Y “SALA3ENTREGADO” según el tipo de sala a los que han sido enviados los productos. También estará una variable asociada a un eliminador de productos el cual se encarga de eliminar los pallets de los productos entregados. Estas estarán activas siempre.

Luego tendremos otros eliminadores de productos ubicados tanto en la zona de almacenaje como en la zona de recepción de productos y se denominarán. Estos se utilizarán para el modo de rearme donde se deberán eliminar todos los productos que no hayan sido enviados todavía para dejar la zona vacía al completo. En la zona de entrega de productos no se ubicará ninguno ya que se supone que en esta zona solo se encontrarán productos que ya han sido enviados a los clientes. De igual forma pasará con las zonas posteriores a las estantería donde se encontrarán productos enviados pero esto se controlará de forma que solo se pueda pulsar la opción de “REARME” cuando ya hayan sido entregados todos los productos.



Figura 4-6: Variables emisor y eliminador de productos FACTORY IO

A continuación, se han construido tablas en la cuales se muestran todas las variables externas que se han utilizado en el proyecto según su localización en el mismo, con la información del tipo de variable y la descripción de la misma.

4.1.1.1 Zona de recepción de productos

En primer lugar, se muestran las variables usadas para las cintas de la zona de recepción de productos.

RECEPCIÓN: SENSORES CINTAS		
VARIABLES	TIPO	DESCRIPCIÓN
RS1	BOOL	Sensor que muestra cuando se ha recibido del proveedor un producto plateado
RS2	BOOL	Sensor que muestra cuando se ha recibido del proveedor un producto verde
RS3	BOOL	Sensor que muestra cuando se ha recibido del proveedor un producto azul
RS4	BOOL	Sensor que muestra cuando llega un producto plateado a la base giratoria número 1
RS5	BOOL	Sensor que muestra cuando llega un producto verde a la base giratoria número 1
RS6	BOOL	Sensor que muestra cuando llega un producto azul a la base giratoria número 1
RS7	BOOL	Sensor que muestra cuando ha salido un producto de la base giratoria número 1 hacia la cinta general número 4
RS8	BOOL	Sensor que muestra cuando un producto pasa de la cinta general número 4 al número 5
RS9	BOOL	Sensor que muestra cuando un producto pasa de la cinta general número 5 al número 6

Tabla 4-1: Sensores de las cintas ubicadas en la zona de recepción

RECEPCIÓN: ACTUADORES CINTAS		
VARIABLES	TIPO	DESCRIPCIÓN
RC1	BOOL	Actuador de la cinta 1 que recibe producto plateado del proveedor
RC2	BOOL	Actuador de la cinta 2 que recibe producto verde del proveedor
RC3	BOOL	Actuador de la cinta 3 que recibe producto azul del proveedor
RC4	BOOL	Actuador de la cinta general 4

RC5	BOOL	Actuador de la cinta general 5
RC6	BOOL	Actuador de la cinta general 6
RC7	BOOL	Actuador de la cinta general 7
RC8	BOOL	Actuador de la cinta general 8
RC9	BOOL	Actuador de la cinta general 9

Tabla 4-2: Actuadores de las cintas ubicadas en la zona de recepción

Puesto que a las bases giratorias les corresponden unas variables que coinciden en cada una de ellas excepto en el segundo dígito que muestra el número de la base giratoria a la que corresponde. G1, G2, G3, G4, G5 estando las últimas cuatro en la zona de almacenaje

RECEPCIÓN: BASE GIRATORIA 1		
VARIABLES	TIPO	DESCRIPCIÓN
G1	BOOL	Actuador de la base giratoria 1 en la cual se encarga de hacerla girar 90°
G1MENOS	BOOL	Actuador de la base giratoria 1 en la cual se encarga de transportar los productos en un sentido de la base
G1MAS	BOOL	Actuador de la base giratoria 1 en la cual se encarga de transportar los productos en el otro sentido de la base
G1R	BOOL	Sensor de la base giratoria 1 en la cual se encarga de mostrar que la base giratoria no está girada
G1D	BOOL	Sensor de la base giratoria 1 en la cual se encarga de mostrar que la base giratoria está girada
G11	BOOL	Sensor de la base giratoria 1 en la cual se encarga de mostrar que hay un producto sobre ella en un lado de la misma
G12	BOOL	Sensor de la base giratoria 1 en la cual se encarga de mostrar que hay un producto sobre ella en el otro lado de la misma

Tabla 4-3: Actuadores y sensores de la base giratoria 1

4.1.1.2 Zona de almacenaje

Para la zona de almacenaje se muestran, en primer lugar, las variables correspondientes a las cintas generales y después se muestran las variables específicas para las cintas dedicadas a cada tipo de producto y para las estanterías.

ALMACENAJE: SENSORES CINTAS GENERALES		
VARIABLES	TIPO	DESCRIPCIÓN
AC1	BOOL	Actuador de la cinta general AC1
AC2	BOOL	Actuador de la cinta general AC2
AC3	BOOL	Actuador de la cinta general AC3
AC4	BOOL	Actuador de la cinta general AC4

Tabla 4-4: Actuadores cintas generales ubicadas en la zona de almacenaje

ALMACENAJE: SENSORES CINTAS GENERALES		
VARIABLES	TIPO	DESCRIPCIÓN
TIPOMED	INT	Sensor que nos dice el tipo de producto que llega a la zona de almacenaje
AS1	BOOL	Sensor que muestra cuando un producto llega a la zona de almacenaje antes de la base giratoria 2
AS2	BOOL	Sensor que muestra cuando un producto sale de la base giratoria 2 hacia la cinta general AC1
AS3	BOOL	Sensor que muestra cuando un producto está entre la cinta general AC1 y AC2
AS4	BOOL	Sensor que muestra cuando un producto llega a la base giratoria 4 desde la cinta general AC2
AS5	BOOL	Sensor que muestra cuando un producto sale de la base giratoria 5 hacia la cinta general AC3
AS6	BOOL	Sensor que muestra cuando un producto está entre la cinta general AC3 y AC4
AS7	BOOL	Sensor que muestra cuando un producto llega a la base giratoria 3 desde la cinta general AC4
AS8	BOOL	Sensor que muestra cuando un producto sale de la cinta general 3 hacia la zona de entrega

Tabla 4-5: Sensores cintas generales ubicadas en la zona de almacenaje

ALMACENAJE: VARIABLES CINTAS ESTANTERÍA VERDE		
VARIABLES	TIPO	DESCRIPCIÓN
CV1	BOOL	Actuador encargado de la cinta de productos verdes CV1 a su salida de la base giratoria 2
CV2...CV4	BOOL	Actuador encargado de la cinta de productos verdes CV1...CV4
CV5	BOOL	Actuador encargado de la cinta de productos verdes CV5 la cual conecta con la estantería de productos verdes
CV6	BOOL	Actuador encargado de la cinta de productos verdes CV6 a su salida de la estantería de productos verdes
CV7...CV10	BOOL	Actuador encargado de la cinta de productos verdes CV7...CV10
CV11	BOOL	Actuador encargado de la cinta de productos verdes CV6 que conecta con la base giratoria 4
SV1	BOOL	Sensor que muestra cuando un producto verde ha salido de la base giratoria 2 en su camino hacia la estantería verde
SV2...SV5	BOOL	Sensores que muestran cuando un producto verde pasa entre las cintas de ingreso a la estantería de productos verdes
SV6	BOOL	Sensor que muestra cuando un producto verde llega a la estantería de productos verdes desde la cinta CV5
SV7	BOOL	Sensor que muestra cuando un producto verde sale de la estantería verde hacia la cinta CV6
SV8...SV12	BOOL	Sensores que muestran cuando un producto verde pasa entre las cintas de salida de la estantería de productos verdes
SV13	BOOL	Sensor que muestra cuando un producto verde llega a la base giratoria 4 desde la cinta CV11

Tabla 4-6: Actuadores y sensores de las cintas específicas de productos verdes

Ahora se mostrarán las variables usadas por la apiladora de productos. Debido a que los nombres de las variables de estas apiladoras son similares para los tres tipos de productos excepto por la letra característica,

mostraremos solo la tabla de las variables de la apiladora de productos verdes.

ALMACENAJE: VARIABLES ESTANTERÍA VERDE		
VARIABLES	TIPO	DESCRIPCIÓN
ALMACENVCD	INT	Actuador encargado de ubicar la posición de la apiladora según el valor de este. Esta variable tendrá 54 valores diferentes, una para cada una de las diferentes celdas de la estantería y el valor de 21474 para la ubicación inicial de la apiladora.
AVSUBEBAJA	BOOL	Actuador encargado de subir y bajar las pinzas de la apiladora de productos verdes
AVCENTRADO	BOOL	Sensor encargado de mostrar cuando las pinzas están centradas.
AVMENOS	BOOL	Actuador encargado de mover las pinzas en uno de los dos sentidos que tiene.
AVMAS	BOOL	Actuador encargado de mover las pinzas en el otro de los sentidos que tiene.
AVEXTMENOS	BOOL	Sensor encargado de mostrar cuando las pinzas han sido extendidas en el sentido en el cual las movía el actuador AVMENOS
AVEXTMAS	BOOL	Sensor encargado de mostrar cuando las pinzas han sido extendidas en el sentido en el cual las movía el actuador AVMAS
SAVX	BOOL	Sensor que muestra cuando la apiladora se está moviendo en el eje de la X o paralela al suelo.
SAVZ	BOOL	Sensor que muestra cuando la apiladora se está moviendo en el eje de la Z o perpendicular al suelo.

Tabla 4-7: Actuadores y sensores del apilador de productos verdes

ALMACENAJE: VARIABLES CINTAS ESTANTERÍA PLATEADA		
VARIABLES	TIPO	DESCRIPCIÓN
CP1	BOOL	Actuador encargado de la cinta de productos plateados CP1 a su salida de la base giratoria 4

CP2	BOOL	Actuador encargado de la cinta de productos verdes CP2 la cual conecta con la estantería de productos plateados
CP3	BOOL	Actuador encargado de la cinta de productos verdes CP3 a su salida de la estantería de productos plateados
CP4...CP12	BOOL	Actuador encargado de la cinta de productos verdes CP4...CP12
CP13	BOOL	Actuador encargado de la cinta de productos plateados que conecta con la base giratoria 5
SP1	BOOL	Sensor que muestra cuando un producto plateado ha salido de la base giratoria 4 en su camino hacia la estantería plateado
SP2	BOOL	Sensor que muestran cuando un producto plateado pasa entre las cintas de ingreso a la estantería de productos plateados
SP3	BOOL	Sensor que muestra cuando un producto verde llega a la estantería de productos plateados desde la cinta CP2
SP4	BOOL	Sensor que muestra cuando un producto plateado sale de la estantería verde hacia la cinta CP3
SP5...SP14	BOOL	Sensores que muestran cuando un producto plateado pasa entre las cintas de salida de la estantería de productos plateados
SP15	BOOL	Sensor que muestra cuando un producto verde llega a la base giratoria 5 desde la cinta CP13

Tabla 4-8: Actuadores y sensores de las cintas específicas de productos plateados

ALMACENAJE: VARIABLES CINTAS ESTANTERÍA AZUL		
VARIABLES	TIPO	DESCRIPCIÓN
CA1	BOOL	Actuador encargado de la cinta de productos azules CA1 a su salida de la base giratoria 3
CA2...CA3	BOOL	Actuador encargado de la cinta de productos azules CA2...CA3
CA4	BOOL	Actuador encargado de la cinta de productos azules CA4 la cual conecta con la estantería de productos azules
CA5	BOOL	Actuador encargado de la cinta de productos azules CA5 a su salida de la estantería de productos azules

CA6...CA8	BOOL	Actuador encargado de la cinta de productos azules CA6...CA8
CA9	BOOL	Actuador encargado de la cinta de productos azules CA9 que conecta con la base giratoria 5
SA1	BOOL	Sensor que muestra cuando un producto azul ha salido de la base giratoria 3 en su camino hacia la estantería azules
SA2...SA4	BOOL	Sensores que muestran cuando un producto azul pasa entre las cintas de ingreso a la estantería de productos azules
SA5	BOOL	Sensor que muestra cuando un producto verde llega a la estantería de productos verdes desde la cinta CA4
SA6	BOOL	Sensor que muestra cuando un producto azul sale de la estantería verde hacia la cinta CA5
SA7...SA10	BOOL	Sensores que muestran cuando un producto azul pasa entre las cintas de salida de la estantería de productos azules
SA11	BOOL	Sensor que muestra cuando un producto verde llega a la base giratoria 5 desde la cinta CA9

Tabla 4-9: Actuadores y sensores de las cintas específicas de productos azules

4.1.1.3 Zona de envío de productos

Para esta zona mostraremos en primer lugar las variables de las cintas generales ubicadas en la misma como de los sensores ubicados en las mismas. También se mostrarán las variables usadas en el elevador, en el brazo robótico o en las manos de envío.

ENVÍO: VARIABLES GENERALES		
VARIABLES	TIPO	DESCRIPCIÓN
EC1	BOOL	Actuador de la cinta general EC1
EC2	BOOL	Actuador de la cinta general EC2
EC3±	BOOL	Actuadores de la cinta general AC3
EC4	BOOL	Actuador de la cinta general EC4
EC5	BOOL	Actuador de la cinta general EC5
ELEVADORMENOS	BOOL	Actuador que mueve la cinta del elevador en el sentido del envío de pedidos
ELEVADORMAS	BOOL	Actuador que mueve la cinta del elevador en el sentido contrario del envío de pedidos

ALTURAELEVADOR	INT	Actuador que marca la altura a la que debe estar el elevador
ES1	BOOL	Sensor que muestra cuando un producto llega a la zona de envío al llegar desde la cinta general EC1 hasta la cinta general EC2
ES2	BOOL	Sensor que muestra cuando un producto llega desde la cinta EC2 hasta el elevador
ES3	BOOL	Sensor que muestra cuando un producto entra en la cinta del elevador
ES4	BOOL	Sensor que muestra cuando un producto llega al final de la cinta del elevador
ES5	BOOL	Sensor que muestra cuando un producto sale de la cinta del elevador
ES6	BOOL	Sensor que muestra cuando un producto llega a la zona del brazo mecánico
ES7	BOOL	Sensor que muestra cuando un producto sin paletizar es dejado en la cinta general EC4
ES8	BOOL	Sensor que muestra cuando un producto llega a la cinta de envío de pedidos donde se mandan a cada una de las salas correspondientes
TIPOMED2	INT	Sensor que muestra el tipo de producto que llega a la zona de envío de pedidos

Tabla 4-10: Actuadores y sensores de las cintas generales ubicadas en la zona de envío

Para la grúa recogedora de productos se añadieron, en último lugar, los sensores P1, P2 P3 y P4 debido a un problema surgido con la cogida del producto final debido a que algunos venían desplazados en el pallet y este no era buena referencia de parada.

ENVÍO: GRÚA RECOGEDORA DE PRODUCTOS Y MANOS DE ENVÍO		
VARIABLES	TIPO	DESCRIPCIÓN
BRAZOBJADO	BOOL	Actuador encargado de bajar y subir el brazo robótico
BRAZOMOVZ	BOOL	Sensor que muestra si el brazo robótico está bajando o subiendo
BRAZOEXP	BOOL	Actuador encargado de mover el brazo robótico hacia adelante y atrás

BRAZOMOVX	BOOL	Sensor que muestra si el brazo robótico está moviéndose hacia adelante o atrás
P1...P4	BOOL	Sensores que muestran cuando está situado el producto final en la zona perfecta para la recogida de este por el brazo robótico.
SALA1	BOOL	Actuador encargado de mover el brazo de envío 1 hacia la zona de envío 1
SALA1MAS	BOOL	Actuador encargado de mover la cinta del brazo de envío 1
SALA2	BOOL	Actuador encargado de mover el brazo de envío 2 hacia la zona de envío 2
SALA2MAS	BOOL	Actuador encargado de mover la cinta del brazo de envío 2
SALA3	BOOL	Actuador encargado de mover el brazo de envío 2 hacia la zona de envío 2
SALA3MAS	BOOL	Actuador encargado de mover la cinta del brazo de envío 2

Tabla 4-11: Actuadores y sensores de la grúa recogedora de productos y de los brazos giratorios

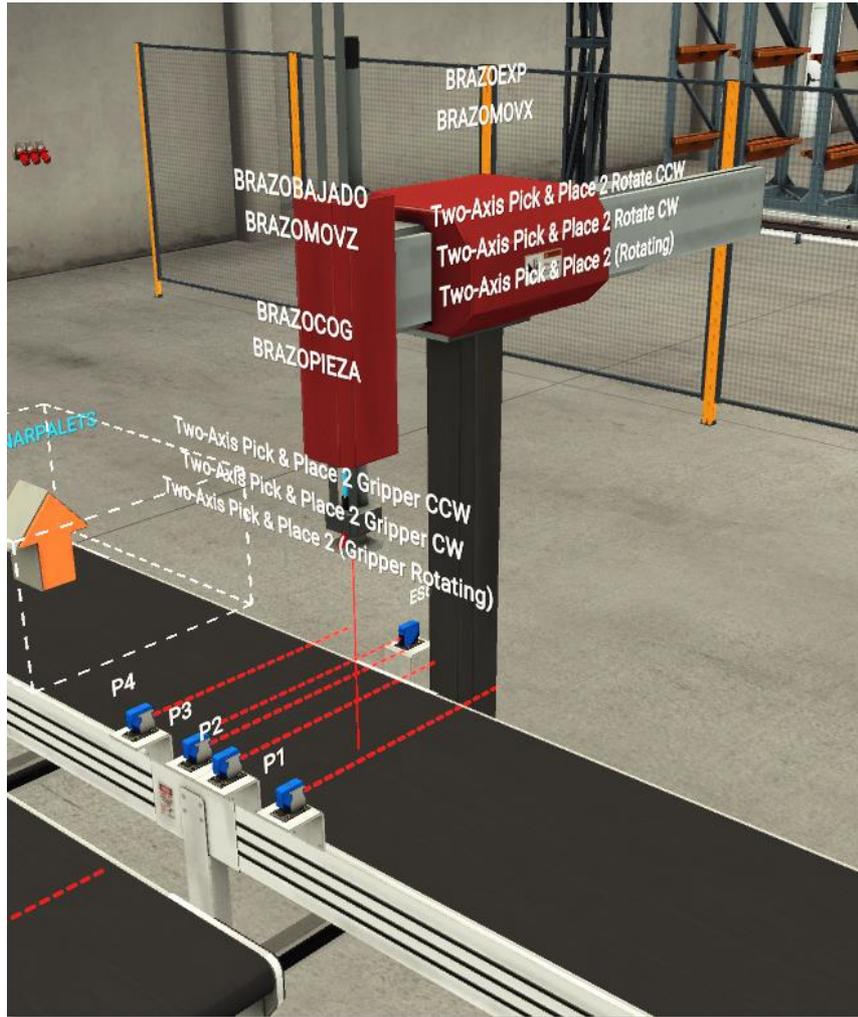


Figura 4-7: Sensores P1, P2, P3 y P4 FACTORY IO

4.1.2 Variables internas

Las variables internas son aquellas que han sido creadas para el correcto funcionamiento del programa y no porque tuvieran que enlazarse con algún sensor o actuador del programa. Estas pueden ser tanto contadores encargados de llevar el número de productos en cada zona del almacén, como variables para llevar la cuenta de los pedidos, o simplemente variables BOOL encargadas de enlazar las diferentes POU's.

En la siguiente tabla se muestran aquellos contadores utilizados para saber el número de productos ubicados en la cinta correspondiente.

CONTADORES CINTAS		
VARIABLES	TIPO	DESCRIPCIÓN
CONC1...CONC9	BOOL	Contadores encargados de llevar la cuenta del número de productos que discurren por las cintas generales a la cual están adjudicados.
CONCV1...CONCV2	BOOL	Contador encargado de llevar la cuenta de productos verdes que pasan por las cintas a las que están adjudicados.

CONCA1...CONCA2	BOOL	Contador encargado de llevar la cuenta de productos azules que pasan por las cintas a las que están adjudicados.
CONCP1...CONCP2	BOOL	Contador encargado de llevar la cuenta de productos plateados que pasan por las cintas a las que están adjudicados.

Tabla 4-12: Contadores de las cintas

En la siguiente tabla se muestran los contadores encargados de llevar la cuenta de aquellos productos que o han sido recibidos del proveedor o han sido almacenados en las estanterías correspondientes o han sido pedidos por los clientes. Hay otros contadores como el contador "CONPEDIDOSVERDESALA1" el cual muestra los productos verdes pedidos para la sala 1, pero no los introduciremos en la tabla ya que son contadores solo informativos sin ninguna función hacia el programa.

CONTADORES PRODUCTOS RECIBIDOS, ALMACENADOS Y PEDIDOS		
VARIABLES	TIPO	DESCRIPCIÓN
CONRA	CTUD	Contador encargado de llevar la cuenta de productos azules que han sido recibidos del proveedor. Si este tiene un valor superior a 54 hacen que se pare de recibir productos azules del proveedor.
CONRV	CTUD	Contador encargado de llevar la cuenta de productos verdes que han sido recibidos del proveedor. Si este tiene un valor superior a 54 hacen que se pare de recibir productos verdes del proveedor.
CONRP	CTUD	Contador encargado de llevar la cuenta de productos plateados que han sido recibidos del proveedor. Si este tiene un valor superior a 54 hacen que se pare de recibir productos plateados del proveedor.
CONAA	CTUD	Contador encargado de llevar la cuenta de los productos azules almacenados en la estantería de productos azules
CONAV	CTUD	Contador encargado de llevar la cuenta de los productos azules almacenados en la estantería de productos verdes
CONAP	CTUD	Contador encargado de llevar la cuenta de los productos azules almacenados en la estantería de productos plateados
CONPA	CTUD	Contador encargado de llevar la cuenta de productos azules que han sido pedidos.
CONPV	CTUD	Contador encargado de llevar la cuenta de productos verdes que han sido pedidos.

CONPP	CTUD	Contador encargado de llevar la cuenta de productos plateados que han sido pedidos
-------	------	--

Tabla 4-13: Contadores de los productos recibidos, almacenados y pedidos

También se ha necesitado crear unas variables de tipo ARRAY las cuales se han utilizado para saber tanto las celdas ocupadas en las estanterías, como para almacenar los datos de los diferentes pedidos o el número de los pedidos ya entregados.

MATRICES		
VARIABLES	TIPO	DESCRIPCIÓN
MATRIZAA	ARRAY	Matriz encargada de almacenar y mostrar cuales celdas de la estantería de productos azules están ocupadas.
MATRIZAV	ARRAY	Matriz encargada de almacenar y mostrar cuales celdas de la estantería de productos verdes están ocupadas.
MATRIZAP	ARRAY	Matriz encargada de almacenar y mostrar cuales celdas de la estantería de productos plateados están ocupadas.
MATRIZPEDIDOS	ARRAY	Matriz encargada de almacenar y mostrar tanto el tipo de producto que se ha pedido, como la sala donde la cual se ha pedido y el número de pedido.
PEDIDOSENTREGADOS	ARRAY	Matriz encargada de almacenar y mostrar los pedidos que ya han sido entregados.

Tabla 4-14: Matrices del programa

Algunas variables también han sido creadas debido a su utilidad en algunas zonas del proyecto como en las bases giratorias. Un ejemplo de estas pueden ser “TURNOVERDE” o “TURNOAZUL” o “TURNOPLATEADO” las cuales han ayudado a saber cuándo podía una variables de cada tipo entrar en la base giratoria número 1. Estas se crearon ya que había tres zonas por las que se podían entrar a la base giratoria y se necesitaba un orden.

Otras variables internas que se han creado han sido las correspondientes a las SCADAS o a los modos de funcionamiento, las cuales serán explicadas aunque en menor profundidad en el apartado dedicado a esto.

Para finalizar este apartado, explicar que todas estas variables enseñadas son variables globales. Esto se debe a que era necesario que pudieran enlazarse entre diferentes POU. Aunque también se han utilizado otras variables locales que solo se han necesitado en POU específicas y se ha querido disminuir en todo lo posible el tamaño de la lista de variables globales.

4.2 Programación del controlador

Para la hora de la programación, hemos utilizado en su mayoría grafcet para todas las partes del proyecto.

“CONTADORESPROVEEDORES”.

En el POU CONTADORESPROVEEDORES podemos encontrar los contadores CONRA, CONRV y CONRP. Estos contadores tienen la función de llevar el recuento de los productos que se han recibido de los proveedores para que, teniendo en cuenta los pedidos ya enviados desde el almacén, nunca se superen de 56 pedidos de cada tipo enviados a cada estantería ya que es el máximo número de productos que caben en cada estantería. En el primer contador podemos observar que con la variable RS3, la cual es la que nos dice cuando hemos recibido un producto azul de los proveedores, se aumenta en uno el valor del contador. Y con la variable AASOLTAR, la cual es la utilizada para decir que hemos enviado en pedido azul a un cliente, la disminuimos en uno. La variable PALETAZUL es la variable que se activa cuando el operario aparta un producto que ha ocasionado una parada de emergencia.

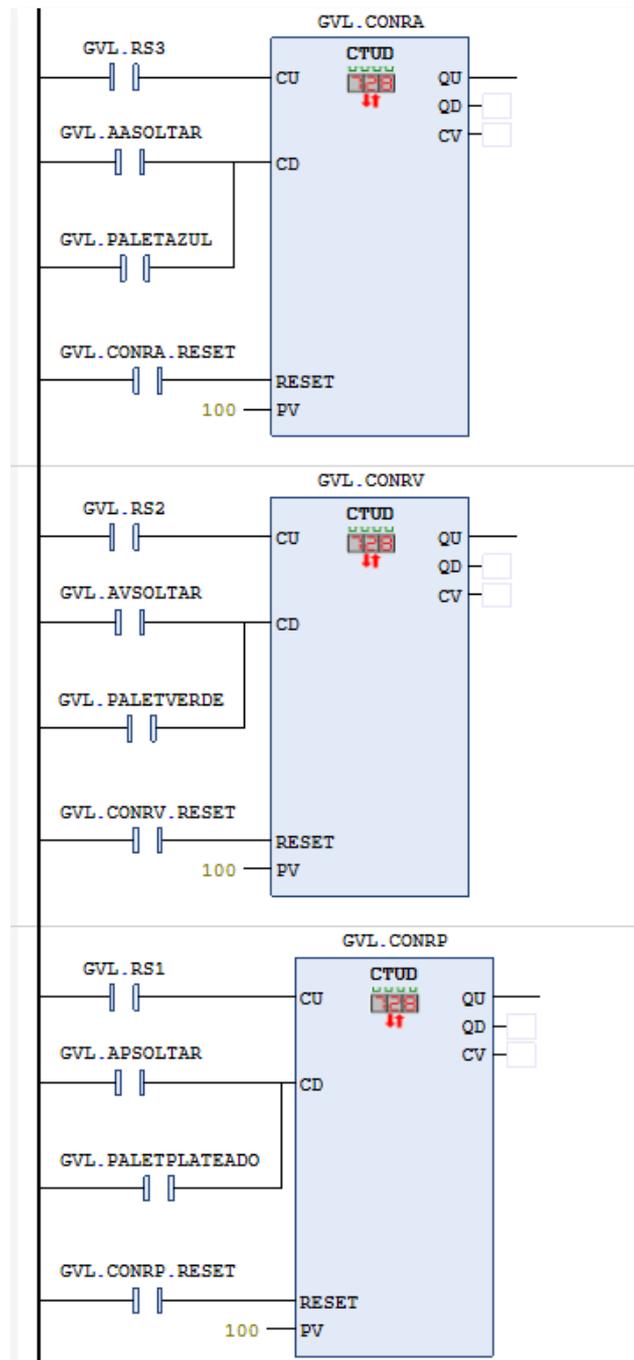


Figura 4-9: POU CONTADORESPROVEEDORES CODESYS

La siguiente POU que comentaremos es CONTADORESPEDIDOS. En esta POU, aparte de otros contadores que no se utilizan en el programa y que solo se utilizan para llevar un recuento específico de cada tipo de pedido, encontramos los contadores CONPA, CONPV y CONPP. Estos contadores serán utilizados para llevar el recuento de aquellos productos que han sido pedidos y se requieren del almacén. Por ejemplo, en el contador CONPA, las variables PEDIDOAZUL1, PEDIDOAZUL2 Y PEDIDOAZUL3 son las variables que nos dicen cuándo se ha pedido un producto azul a cada una de las salas y estas aumentan en uno el valor del contador. Luego la variable SA6, que es la que muestra cuando un producto azul sale de su estantería lo disminuye en uno. La variable PEDIDOAZUL es la correspondiente a los pedidos ya enviados que han sido apartados ya que han ocasionado una parada de emergencia.

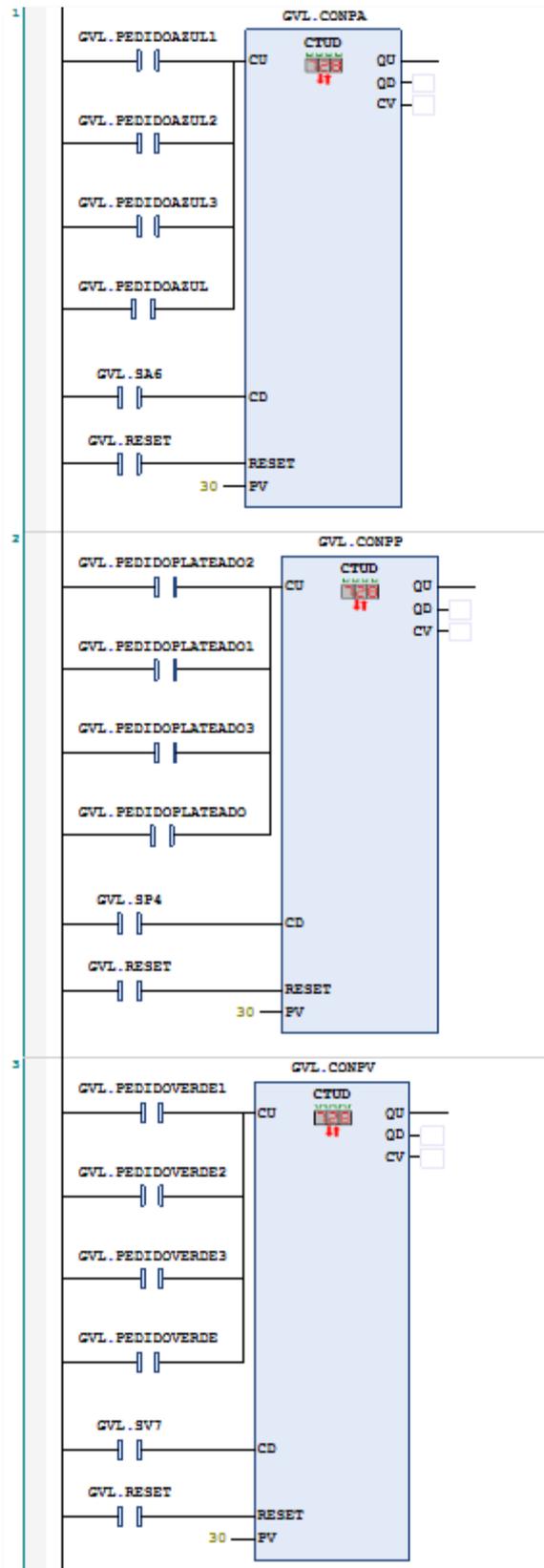


Figura 4-10: POU CONTADORES PEDIDOS CODESYS

Ahora comentaremos la POU CONTADORESALMACENES la cual contiene los contadores dedicados a llevar la cuenta del número de productos almacenados en cada una de las estanterías. Por ejemplo, en el contador CONAV, la variable AVCOLOCAR que es la que nos dice cuándo se ha colocado un producto verde en la estantería, aumenta el contador en uno. Y la variable AVSOLTAR que es la que nos dice cuándo se ha enviado un producto verde a un cliente, lo disminuye en uno.

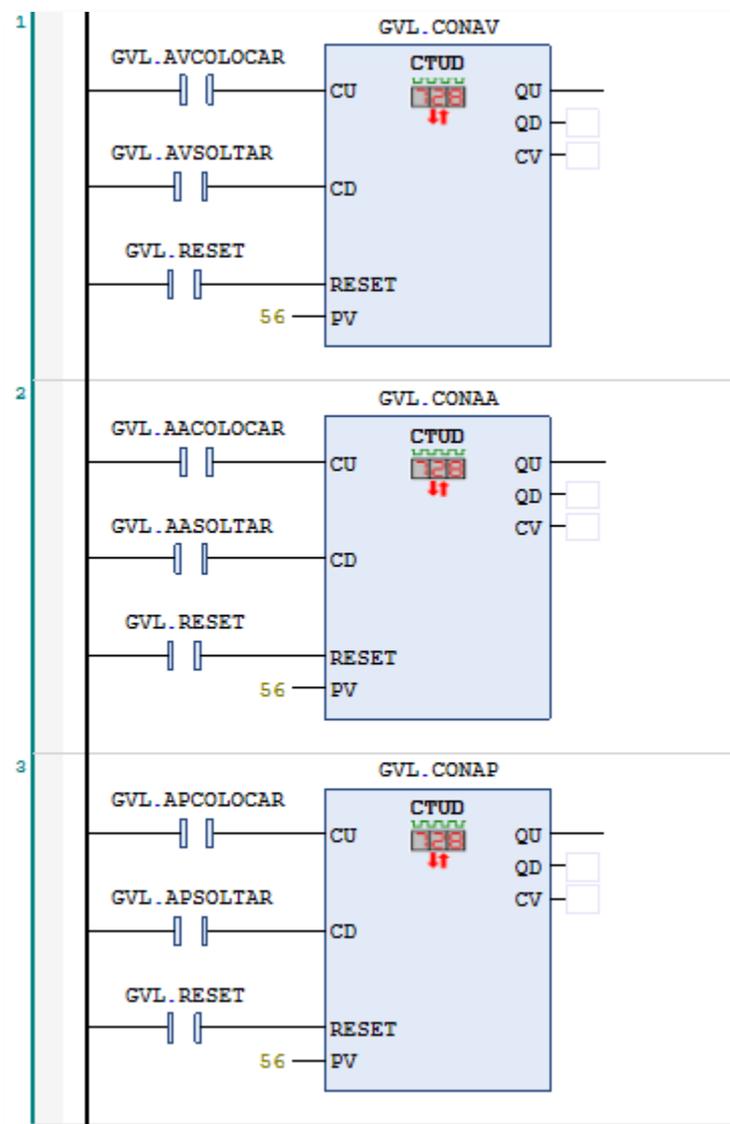


Figura 4-11: POU CONTADORESALMACENES CODESYS

El último de los POU's dedicados a contadores es el POU CONTADORESCINTAS, el cual contiene aquellos contadores dedicados a llevar el recuento de aquellos productos que están pasando en ese momento por la cinta a la que está dedicada ese contador. Por ejemplo, el contador CONC1 lleva la cuenta de los productos que pasan en la cinta 1. La variable RS1 es la encargada de decirnos cuando entra un producto a la cinta y aumenta

el valor del contador en 1. Y la variable RS4 es la que se encarga de decirnos cuando un producto sale de la cinta.

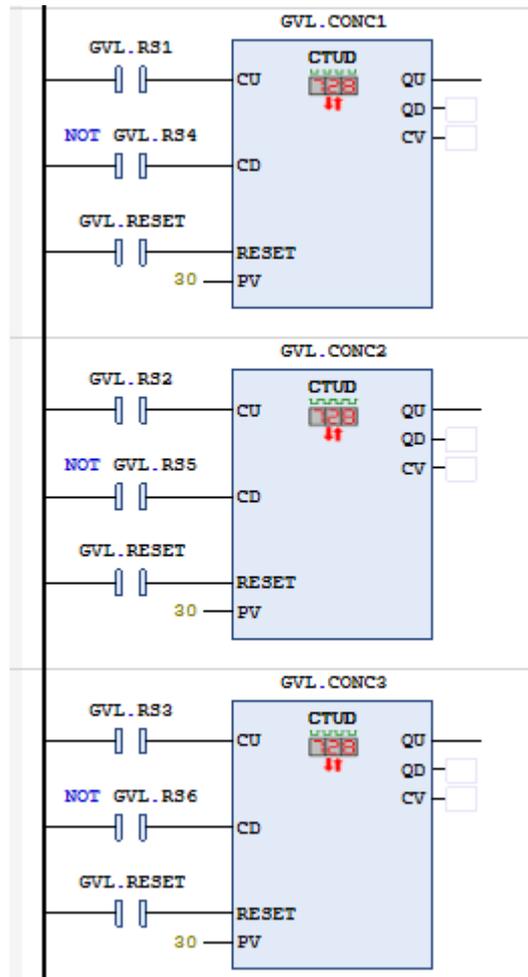


Figura 4-12: POU CONTADORES CINTAS CODESYS

A continuación, comentaremos cada una de las POU's dedicadas a funciones físicas dentro de nuestra planta. No se mostrarán POU's que tengan funciones similares ya que, con la explicación de una, se entenderán el resto.

En primer lugar, explicaremos las POU's dedicadas a hacer funcionar cada una de las cintas que conforma el proyecto. Estas POU's serán: C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, CV1, CV2, CA1, CA2, CP1 y CP2.

Las POU's pueden ser dedicadas a una sola cinta o a un conjunto de Cintas. Por ejemplo, la POU C1 está dedicada a la cinta RC1 solamente, pero la POU C4 está dedicada a las cintas RC4, RC5 y RC6 o la POU CV1 está dedicada a las cintas CV1, CV2, CV3 y CV4. Para saber las cintas de las que se ocupa cada POU, se recomienda visitar el ANEXO 1.

Para poner un ejemplo mostraremos la POU C1. En la POU observamos los pasos que se dan para obtener el ciclo completo de la cinta RC1. Primero se observa si entra un producto a la cinta con la variable RS1, y si es así, se pone en marcha la cinta RC1. Si el producto llega al final de la cinta, se activa la variable RS4, y se para la cinta. Después, la variable TURNOPLATEADO nos dice cuando el producto de la cinta, el cual es plateado, puede pasar a la base giratoria 1. Cuando se activa, se vuelve a accionar la cinta RC1 hasta que sale de ella cuando el valor de RS4 pasa a FALSE. Al final, por si un producto se encuentra en medio de la cinta y ya no

puede activar la variable RS1, se comprueban el número de producto que hay en la cinta, y si es mayor que 0 se vuelve a poner en marcha.

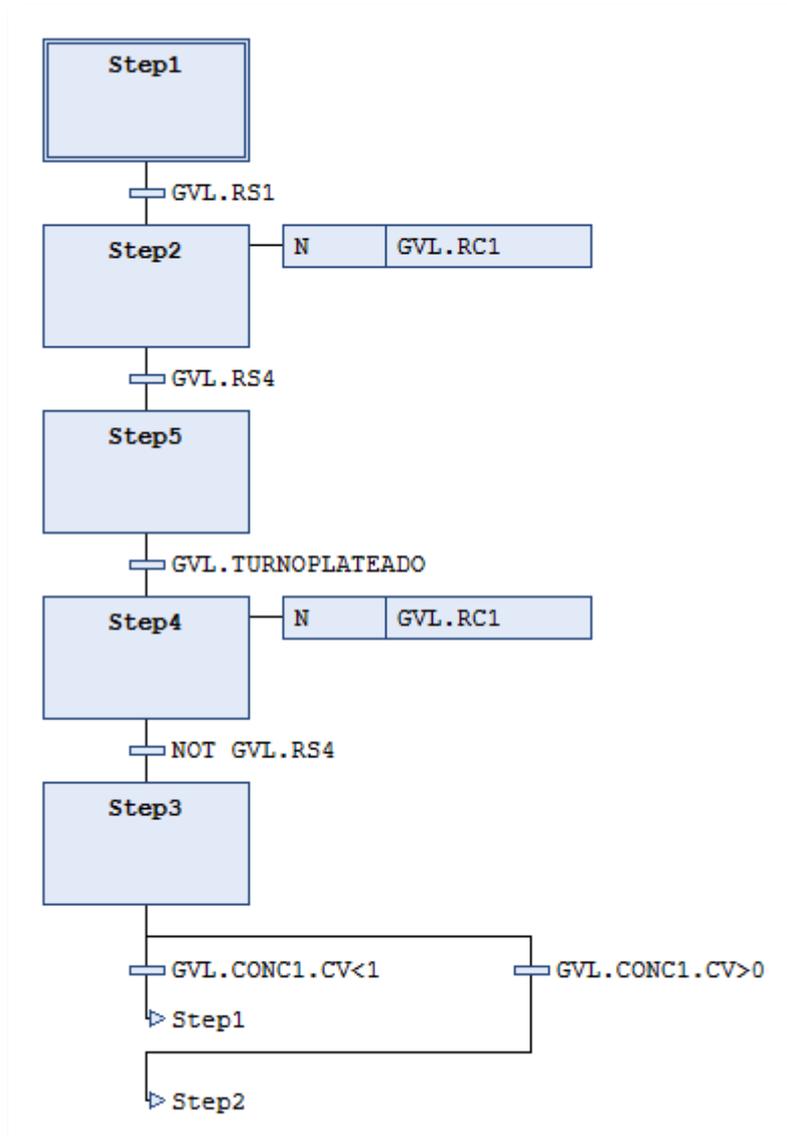


Figura 4-13: POU C1 CODESYS

Ahora comentaremos una de las POU's dedicadas a las bases giratorias. Las POU's de las bases giratorias son muy parecidas entre ellas, solo variando la complejidad según el número de entradas que tienen. De ejemplo se mostrará la POU GIR1 dedicada a la base giratoria 1, la cual, se puede observar a continuación.

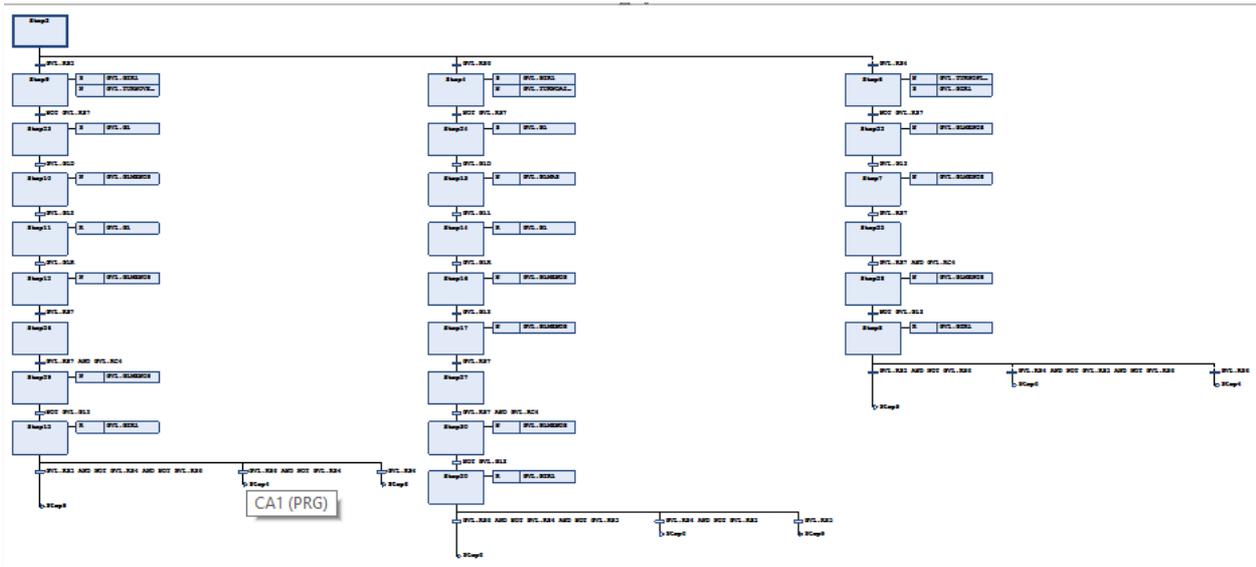


Figura 4-14: POU GIR1 CODESYS

A esta base giratoria se tiene acceso por tres zonas diferentes, desde la cinta RS1, desde la cinta RS2 y desde la cinta RS3. Debido a esto se establecen tres ciclos diferentes de maniobrabilidad de la base giratoria, según de donde venga el producto y se debe establecer un orden de entrada por si coinciden delante de esta dos o más productos.

En primer lugar, se detecta desde donde llega un producto a una de las tres entradas. Cuando llegan un producto o más, el programa detecta el primero y se completa el ciclo de maniobrabilidad del mismo. Nada más empezar este ciclo y para resolver problemas de simultaneidad entre entradas, sobre todo a la hora de activar bases giratorias que están a continuación de otra, se pone según corresponda, una de las variables creadas para ello TURNOVERDE, TURNOAZUL o TURNOPLATEADO en TRUE. Esto es así para que la cinta que envía el producto sepa cuando debe ponerse en marcha y avisa a las otras dos cintas de las otras dos entradas a la base giratoria de que esa base giratoria en ese momento está ocupada. De esta forma se consigue que ningún pedido se moleste a la vez en la cinta. Una vez acabado el ciclo se pone la misma variable a FALSE y se observa si hay uno o más productos esperando a la base giratoria. Si es solo uno, este completa su ciclo. Si son dos, se le da preferencia a uno de ellos para que el envío de productos esté equilibrado.

Para completar este ciclo se utilizan unas variables que se han creado para resolver los problemas de simultaneidad entre entradas. En el caso de la base giratoria 1 estos serían: G1, G1MENOS, G1MAS, G11, G12, G1D y G1R cuya función se ha comentado anteriormente.

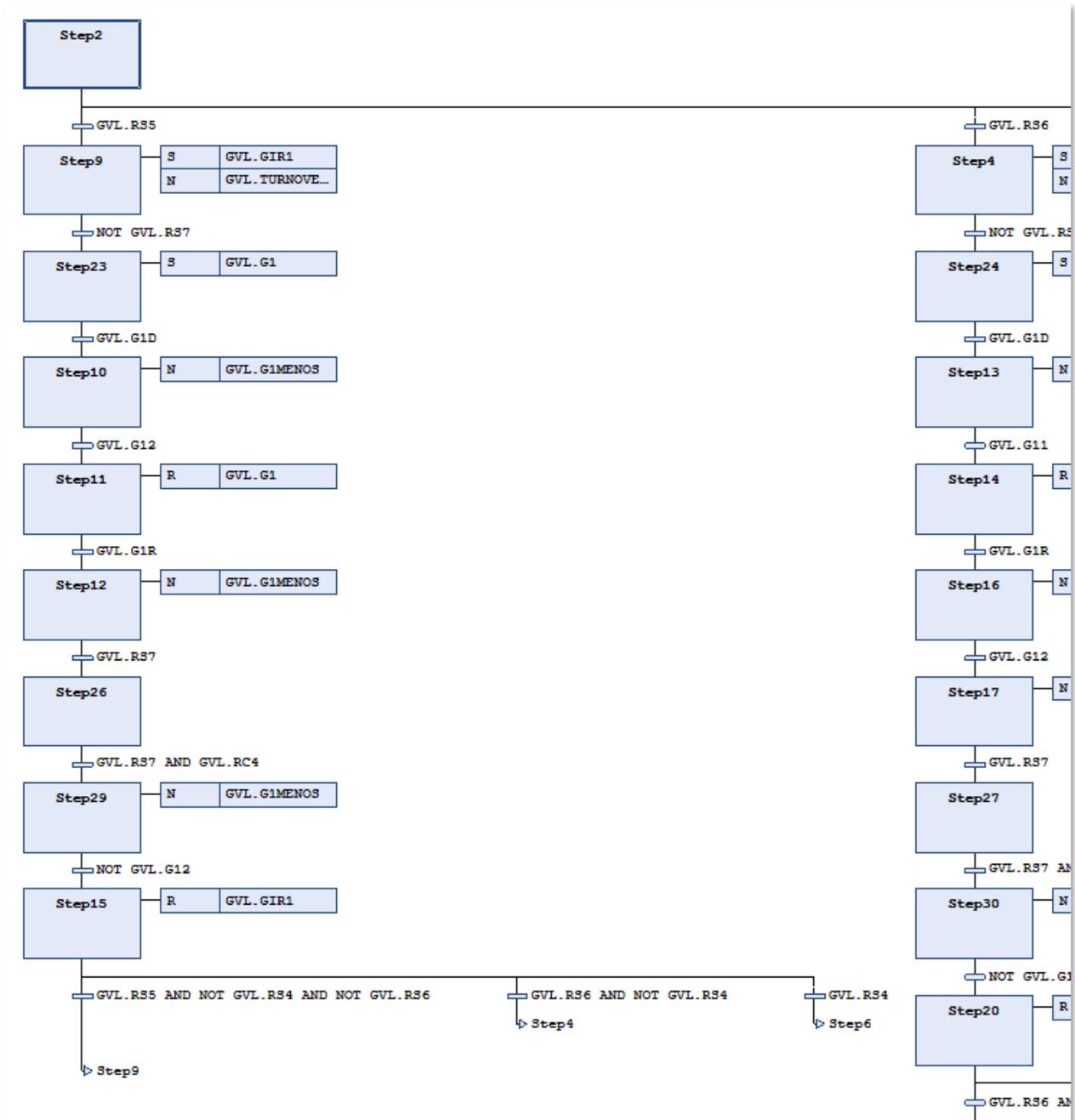


Figura 4-15: POU GIR1 ampliada CODESYS

Otro ejemplo de bases giratorias es el de la base giratoria 2. A esta solo llegan productos de una entrada, pero según el tipo de producto que sea, se envía a una de las otras tres salidas. Esto se consigue con la variable TIPOMED cuya función se ha comentado anteriormente.

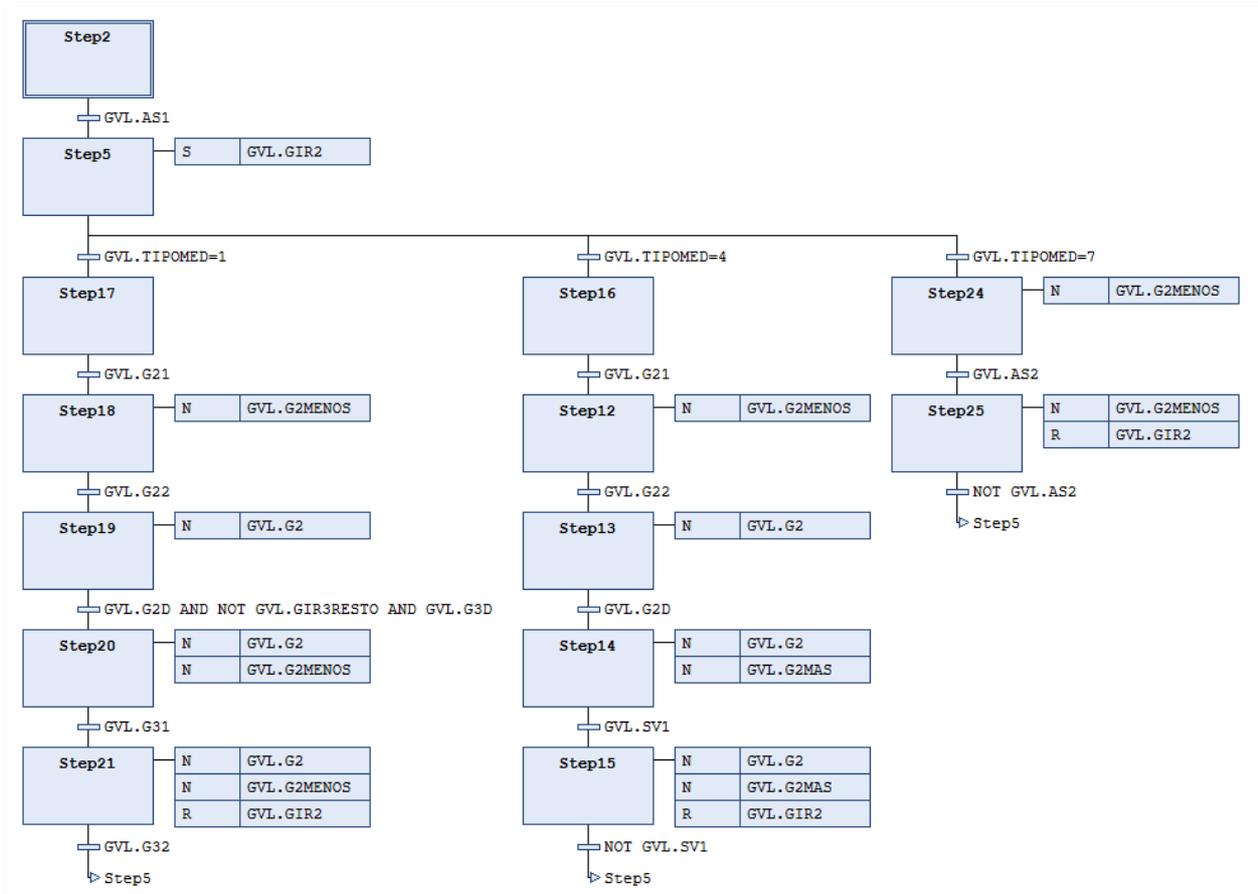


Figura 4-16: POU GIR2 CODESYS

Ahora comentaremos la función de las POUs dedicadas a el almacenaje de los productos en las estanterías gracias a la apiladora de productos. Para cada una de las POUs dedicadas a cada una de las estanterías, se han establecido dos ciclos diferentes, uno dedicado a colocar los productos en las estanterías y otro dedicado a sacarlos de ella. De ejemplo comentaremos la POU AlmacenaA.

Al iniciar la POU, en primer lugar se comprueba el valor del número de pedidos que se ha realizado. Si el valor es mayor que cero, se completa el ciclo dedicado a soltar un producto de la estantería. Explicar que este contador solo será mayor que cero si se ha comprobado antes que hay productos en la estantería, de no ser así, no se podría haber pedido ningún producto de este tipo.

Si el contador vale cero y se detecta un producto en la entrada de la estantería se procede a completar el ciclo de colocarlo en la estantería. Para cada uno de estos ciclos, se habrá establecido antes cuales cavidades de la estantería están llenas y cuales vacías, con la matriz rellena con anterioridad MATRIZAA, para saber el orden de recogida y suelta de productos.

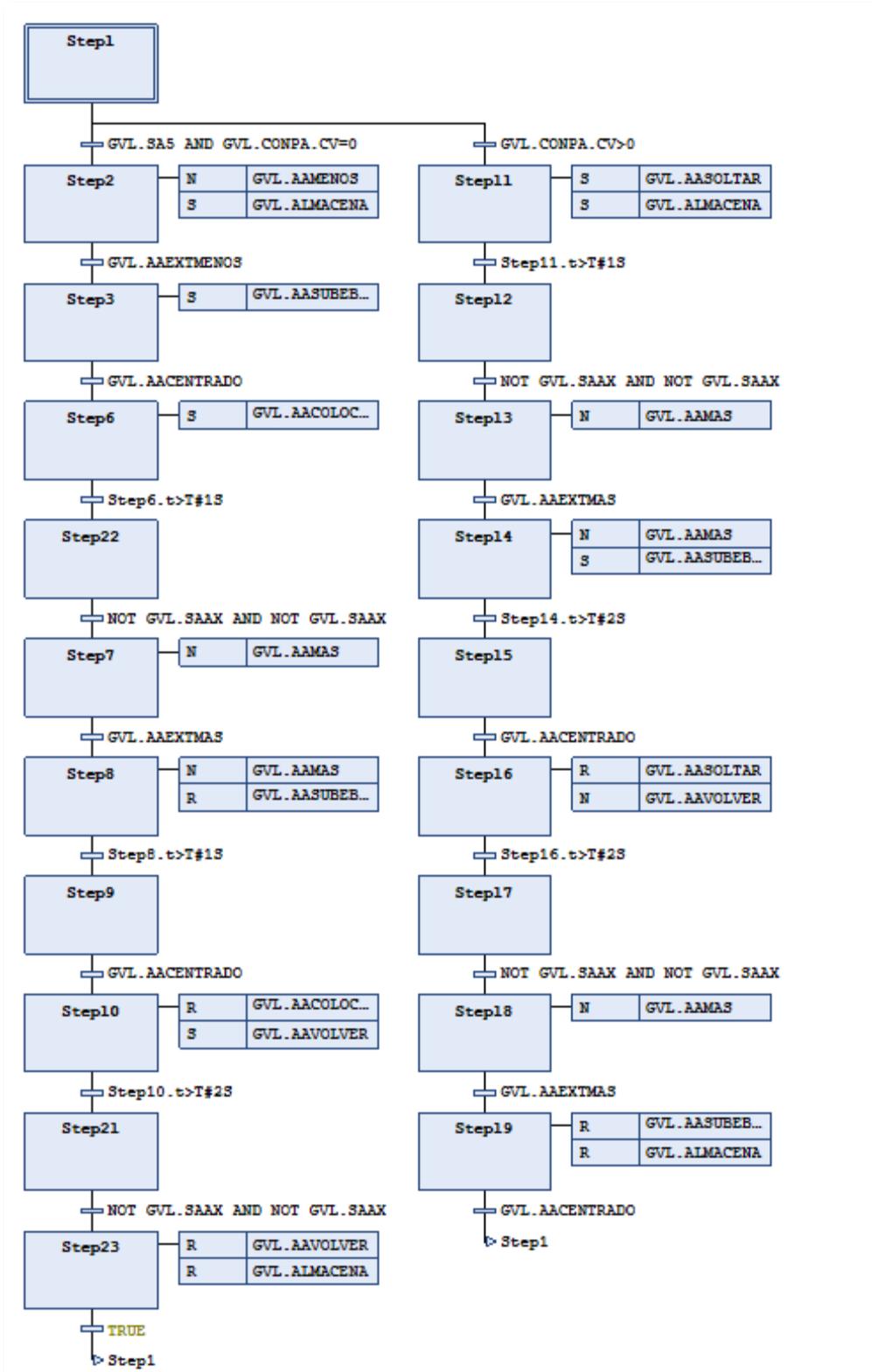


Figura 4-17: POU AlmacenA FACTORY IO

A la hora de rellenar o de vaciar cada una de las estanterías con productos, se comprueban cuáles son las baldas que están ocupadas para seguir un orden. A continuación, se muestra una parte del POU ALMACENES dedicado a conseguir este control para la estantería de productos azules. Para ello se ha creado la matriz MATRIZAA. Esta matriz, de dimensión 6x9, será la encargada de almacenar cuales son las baldas que están ocupadas. En la POU ALMACENES, cuando se quiera almacenar un producto en la estantería de

productos azules que será cuando se active la variable AACOLOCAR, se iniciará un FOR el cual leerá cual es la primera balda vacía para introducir el producto en la misma. Si se inicia el proceso de sacar un producto, el cual comenzará cuando se active la variable AASOLTAR, se volverá a leer la matriz para saber cuál es la primera balda que está llena para vaciarla.

```

(*ALMACEN AZUL*)
AA(CLK:= GVL.AACOLOCAR);
IF AA.Q THEN
  FOR i:=1 TO 6 BY 1 DO
    FOR j:=1 TO 9 BY 1 DO

      l:=(i-1)*9+j;
      IF GVL.MATRIZAA[i,j]=FALSE THEN

          MENORALLENARAA:=1;
          GVL.MATRIZAA[i,j]:=TRUE;
          i:=7;
          j:=10;

      END_IF
    END_FOR
  END_FOR
END_IF

AAE(CLK:= GVL.AASOLTAR);
IF AAE.Q THEN
  FOR i:=1 TO 6 BY 1 DO
    FOR j:=1 TO 9 BY 1 DO

      l:=(i-1)*9+j;
      IF GVL.MATRIZAA[i,j]=TRUE THEN

          MENORAVACIARAA:=1;
          GVL.MATRIZAA[i,j]:=FALSE;
          i:=7;
          j:=10;

      END_IF
    END_FOR
  END_FOR
END_IF

IF GVL.AACOLOCAR=TRUE AND GVL.AAVOLVER=FALSE THEN
  GVL.ALMACENACD := MENORALLENARAA;
END_IF
IF GVL.AAVOLVER=TRUE THEN
  GVL.ALMACENACD:=21474;
END_IF
IF GVL.AASOLTAR=TRUE THEN
  GVL.ALMACENACD:=MENORAVACIARAA;
END_IF

```

Figura 4-18: POU ALMACENES CODESYS

Ahora comentaremos, la POU dedicada a el elevador de productos denominada “ELEVADOR”. Esta POU al solo tener una entrada, solo necesitará un ciclo de maniobrabilidad. En primer lugar, detecta cuando un producto llega a la entrada de la apiladora con la variable ES2 y pone en marcha la cinta del elevador para subir el producto. Después, la variable ALTURAELEVADOR de tipo INT se pone al valor correspondiente al piso al cual debe ir el pedido. Al llegar, se pone en marcha de nuevo la cinta y espera a detectar que ya ha salido con la variable ES6. Una vez salido, el elevador vuelve a la planta baja a recoger el siguiente producto.

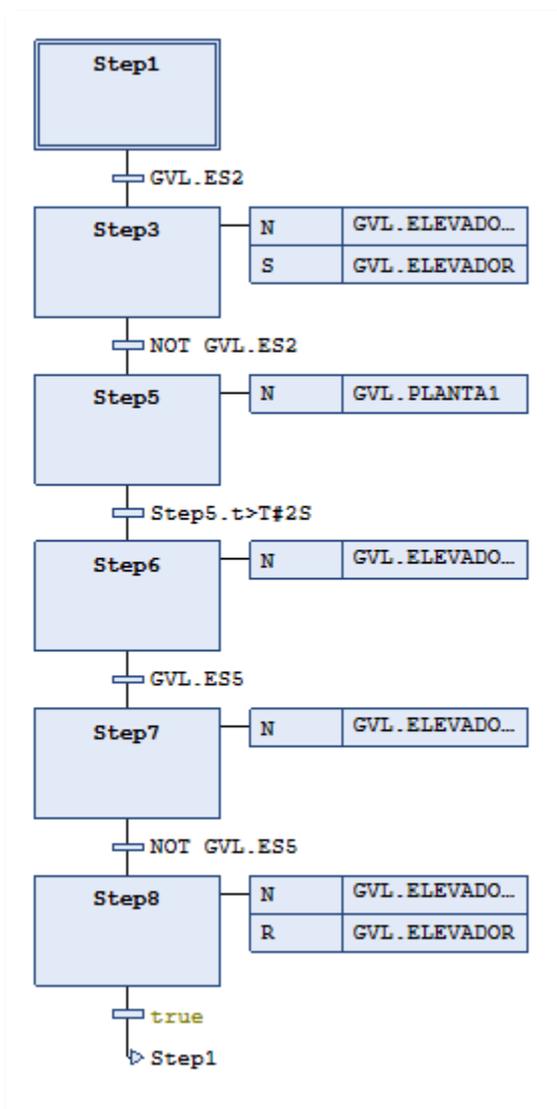


Figura 4-19: POU ELEVADOR CODESYS

```
(*ELEVADOR*)
IF GVL.ELEVADORLIBRE=TRUE THEN
    GVL.ALTURAELEVADOR:=0.0;
END_IF
IF GVL.PLANTA1=TRUE THEN
    GVL.ALTURAELEVADOR:=0.6;
END_IF
```

Figura 4-20: POU ALMACENES CODESYS

Otra POU utilizada es la POU “EXTPRODUCTO” la cual se encarga del funcionamiento del brazo robótico. Para explicar esta POU se mostrará otra POU denominada “C8” la cual es la dedicada al movimiento de la cinta “EC3”. En esta última POU se ha aplicado una solución para el problema del movimiento del producto final en la superficie superior de los pallets. Esto se ha conseguido con la implantación de 4 sensores “P1”, “P2”, “P3” y “P4” junto con una variable interna denominada “P” de tipo INT la cual hace que valiendo 50 sea cuando el producto final se encuentre justo debajo del brazo robótico. Una vez esta bien situado se procede a desarrollarse el POU “EXTMEDICAMENTO” consiguiendo así desplazar de forma correcta el producto final del pallet.

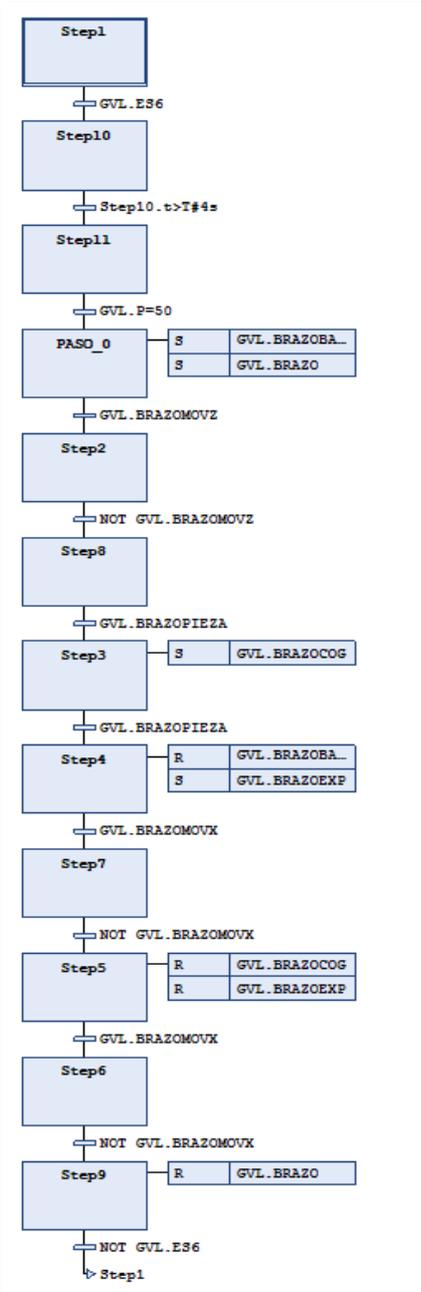


Figura 4-21: POU EXTPRODUCTO CODESYS

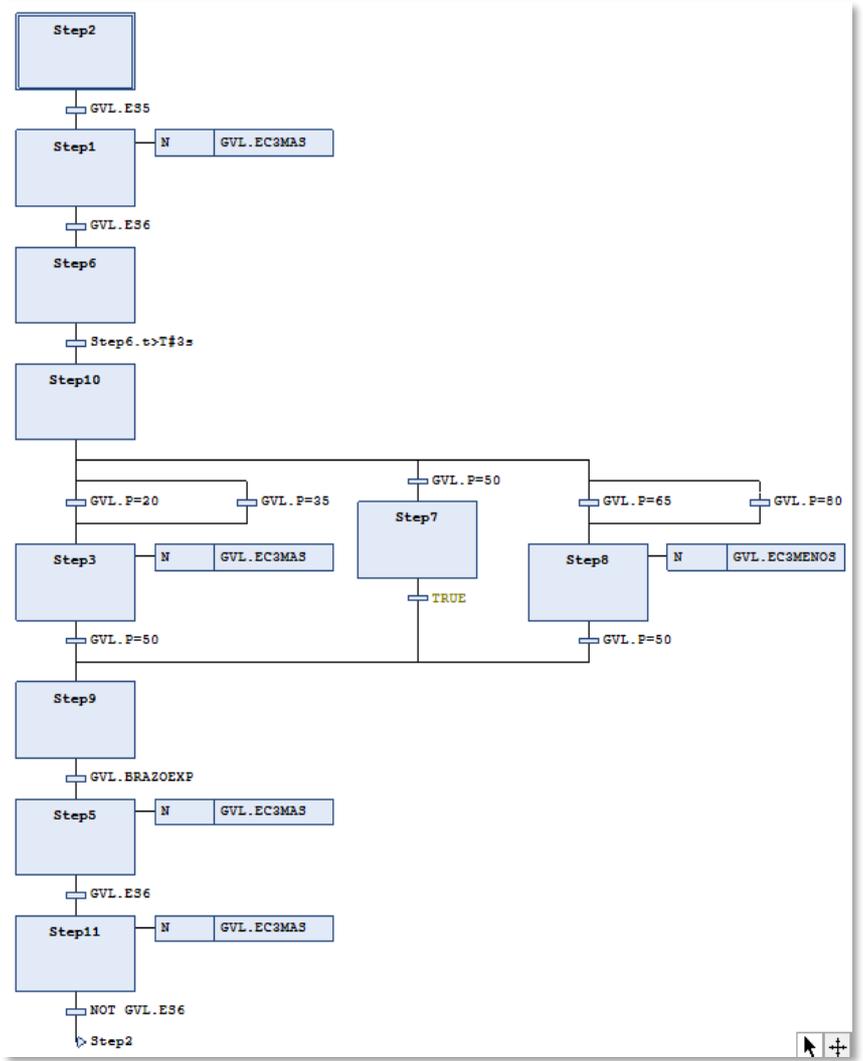


Figura 4-22: POU C8 CODESYS

En la siguiente imagen se muestra una parte de la POU ALMACENES, donde se consigue que el producto acabe justo debajo de la grúa de productos, consiguiendo así que se pueda extraer con facilidad con el uso de los sensores P1, P2 P3 y P4. La diferente activación de estos sensores hace que varía el valor de la variable P haciendo que, cuando esta variable valga 50, lo cual sería cuando el producto esté centrado, se active la recogida de productos.

```
IF GVL.P1=TRUE AND GVL.P2=FALSE AND GVL.P3=FALSE AND GVL.P4=FALSE THEN
    GVL.P:=20;
END_IF
IF GVL.P1=TRUE AND GVL.P2=TRUE AND GVL.P3=FALSE AND GVL.P4=FALSE THEN
    GVL.P:=35;
END_IF
IF GVL.P1=FALSE AND GVL.P2=TRUE AND GVL.P3=TRUE AND GVL.P4=FALSE THEN
    GVL.P:=50;
END_IF
IF GVL.P1=FALSE AND GVL.P2=FALSE AND GVL.P3=TRUE AND GVL.P4=TRUE THEN
    GVL.P:=65;
END_IF
IF GVL.P1=FALSE AND GVL.P2=FALSE AND GVL.P3=FALSE AND GVL.P4=TRUE THEN
    GVL.P:=80;
END_IF
IF GVL.P1=FALSE AND GVL.P2=FALSE AND GVL.P3=FALSE AND GVL.P4=FALSE THEN
    GVL.P:=0;
END_IF
```

Figura 4-23: POU ALMACENES CODESYS

Una vez que el producto final ya es separado del pallet se le envía a través de la cinta EC4 y al final de esta se encuentra con el sensor TIPOMED2. Este sensor da un valor de tipo INT dependiendo del tipo de producto que ha llegado y según este producto y gracias a la POU denominada “C10”, se envía el producto a la sala que le corresponda. En las imágenes a continuación se observa esta POU C10.

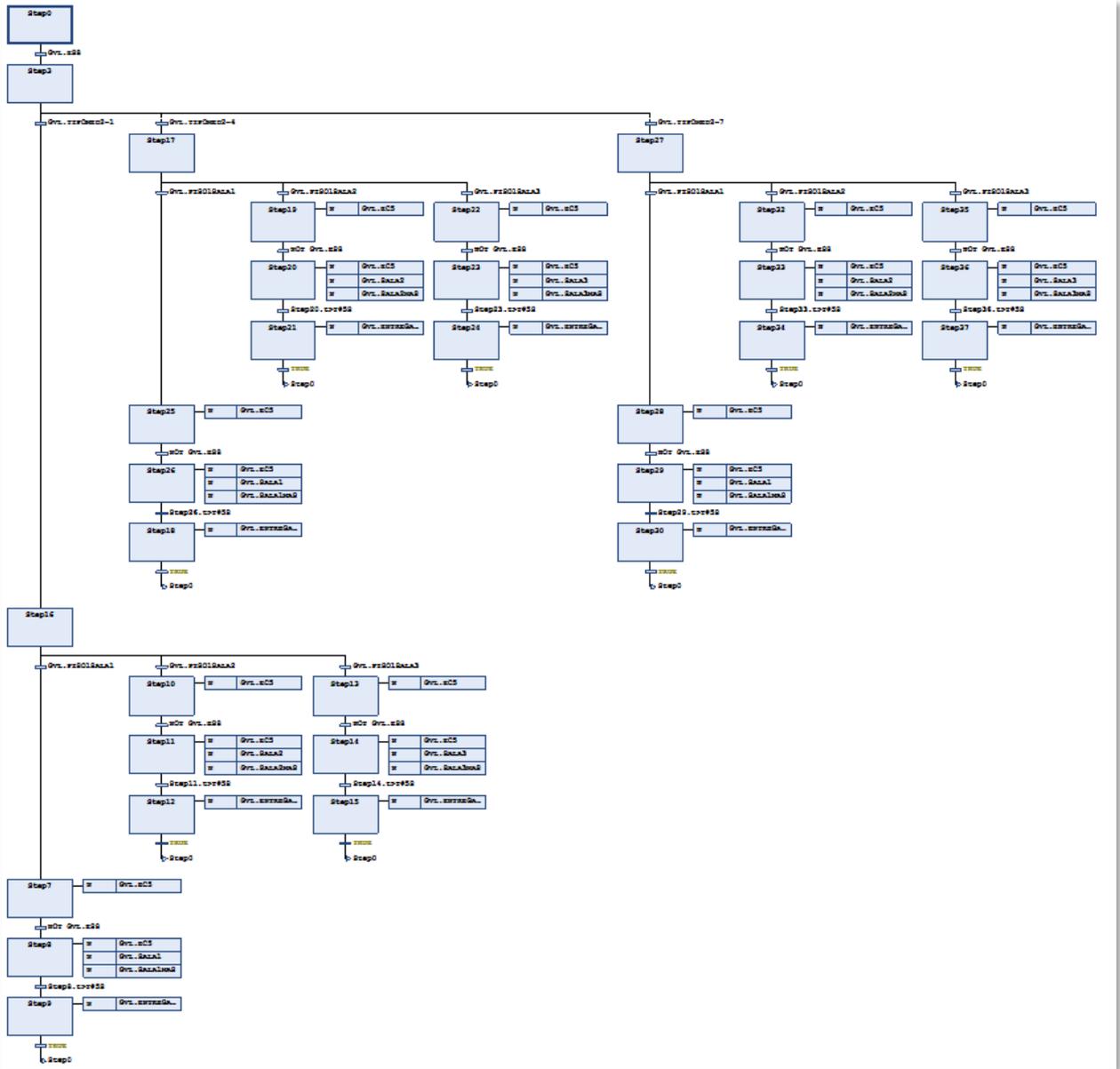


Figura 4-24: POU C10 CODESYS

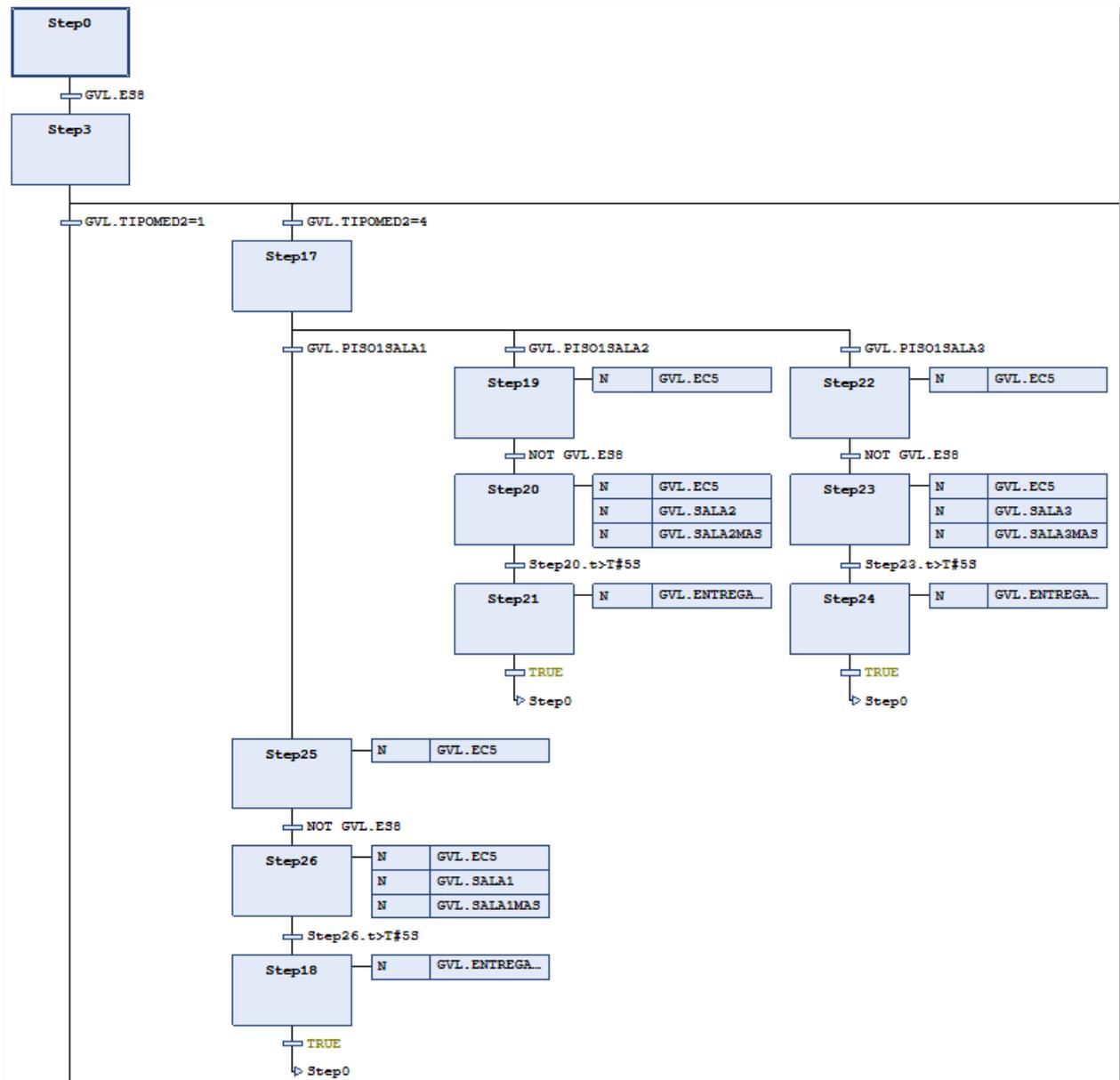


Figura 4-25: POU C10 ampliada CODESYS

Gracias a la variables “PISO1SALA1”, “PISO1SALA2” y “PISO1SALA3” podemos saber la sala a la cual esta destinado el producto. Estas variables se activan en la POU ALMACENES. En esta POU, al llegar un producto a la zona de envío y conocer el tipo de producto que es, se realiza una lectura de la matriz “MATRIZPEDIDOS”, buscando el primer pedido de este tipo de producto, para conocer la sala a la que está enviado y activando después la variable correspondiente a cada sala. A continuación se muestra una parte de la POU ALMACENES, donde se realiza esta función.

```

AZUL(CLK:= GVL.TIPOMED2=1);
IF AZUL.Q THEN
  FOR i:=1 TO 30 BY 1 DO

    IF GVL.MATRIZPEDIDOS[i,1]=1 THEN
      IF GVL.MATRIZPEDIDOS[i,2]=1 THEN
        GVL.PISO1SALA1:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      IF GVL.MATRIZPEDIDOS[i,2]=2 THEN
        GVL.PISO1SALA2:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      IF GVL.MATRIZPEDIDOS[i,2]=3 THEN
        GVL.PISO1SALA3:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      FOR p:=i TO 30 BY 1 DO
        GVL.MATRIZPEDIDOS[p,1]:=GVL.MATRIZPEDIDOS[p+1,1];
        GVL.MATRIZPEDIDOS[p,2]:=GVL.MATRIZPEDIDOS[p+1,2];
        GVL.MATRIZPEDIDOS[p,3]:=GVL.MATRIZPEDIDOS[p+1,3];
      END_FOR
    END_IF
  END_FOR
  i:=30;
  j:=2;
END_IF
END_FOR
END_IF

```

Figura 4-26: POU ALMACENES CODESYS

La matriz MATRIZ PEDIDOS ha sido rellena antes de realizar esta función. Esta matriz, de dimensión 30x3, recoge los datos del tipo de producto pedido, la sala a la cual se ha pedido y el número de pedido del mismo.

A continuación se muestra como se introducen estos datos.

```

('matriz_pedidos')
IF GVL.PEDIDOAZUL1 OR GVL.PEDIDOAZUL2 OR GVL.PEDIDOAZUL3 OR GVL.PEDIDOPLATEADO1 OR GVL.PEDIDOPLATEADO2 O
GVL.NUMEROPEDIDO:=GVL.NUMEROPEDIDO+1;
END_IF
IF GVL.PEDIDOAZUL1=TRUE THEN
FOR i:=1 TO 30 BY 1 DO
FOR j:=1 TO 2 BY 1 DO

IF GVL.MATRIZPEDIDOS[i,j]=0 THEN
GVL.MATRIZPEDIDOS[i,j]:=1;
GVL.MATRIZPEDIDOS[i,j+1]:=1;
GVL.MATRIZPEDIDOS[i,j+2]:=GVL.NUMEROPEDIDO;
i:=30;
j:=2;
END_IF
END_FOR
END_FOR
END_IF
IF GVL.PEDIDOAZUL2=TRUE THEN
FOR i:=1 TO 30 BY 1 DO
FOR j:=1 TO 2 BY 1 DO

IF GVL.MATRIZPEDIDOS[i,j]=0 THEN
GVL.MATRIZPEDIDOS[i,j]:=1;
GVL.MATRIZPEDIDOS[i,j+1]:=2;
GVL.MATRIZPEDIDOS[i,j+2]:=GVL.NUMEROPEDIDO;
i:=30;
j:=2;
END_IF

END_FOR
END_FOR
END_IF
IF GVL.PEDIDOAZUL3=TRUE THEN
FOR i:=1 TO 30 BY 1 DO
FOR j:=1 TO 2 BY 1 DO

IF GVL.MATRIZPEDIDOS[i,j]=0 THEN
GVL.MATRIZPEDIDOS[i,j]:=1;
GVL.MATRIZPEDIDOS[i,j+1]:=3;
GVL.MATRIZPEDIDOS[i,j+2]:=GVL.NUMEROPEDIDO;

```

Figura 4-27: POU ALMACENES CODESYS

Para llevar la cuenta de los pedidos entregados, se ha creado un vector denominado "PEDIDOSENTREGADOS", en el cual introducimos el número de los pedidos que han sido entregados. A continuación se muestra como se ha realizado dentro de la POU ALMACENES.

```

(*ENVIO PEDIDOS*)
AZUL(CLK:= GVL.TIPOMED2=1);
IF AZUL.Q THEN
  FOR i:=1 TO 30 BY 1 DO

    IF GVL.MATRIZPEDIDOS[i,1]=1 THEN
      IF GVL.MATRIZPEDIDOS[i,2]=1 THEN
        GVL.PISO1SALA1:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      IF GVL.MATRIZPEDIDOS[i,2]=2 THEN
        GVL.PISO1SALA2:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      IF GVL.MATRIZPEDIDOS[i,2]=3 THEN
        GVL.PISO1SALA3:=TRUE;
        FOR j:=1 TO 2000 BY 1DO
          IF GVL.PEDIDOSENTREGADOS[j]=0 THEN
            GVL.PEDIDOSENTREGADOS[j]:=GVL.MATRIZPEDIDOS[i,3];
            j:=2001;
          END_IF
        END_FOR
      END_IF
      FOR p:=i TO 30 BY 1 DO
        GVL.MATRIZPEDIDOS[p,1]:=GVL.MATRIZPEDIDOS[p+1,1];
        GVL.MATRIZPEDIDOS[p,2]:=GVL.MATRIZPEDIDOS[p+1,2];
        GVL.MATRIZPEDIDOS[p,3]:=GVL.MATRIZPEDIDOS[p+1,3];
      END_FOR

      i:=30;
      j:=2;
    END_IF
  END_FOR
END_IF

```

Figura 4-28: ALMACENES CODESYS

5 SCADA

Para la creación de una SCADA, hemos utilizado la visualización que nos ofrece CODESYS. La versión 3.5 ofrece gran variedad de funciones y componentes que nos han permitido realizar satisfactoriamente un sistema de solicitud de pedidos junto con otra de visualización del nivel de stock de los diferentes productos en sus respectivos almacenes.

5.1 Diseño del SCADA

La SCADA ha sido diseñada de forma que, tanto el cliente como el operador de la planta, puedan utilizarla.

En un inicio, se creó una sola visualización y se introdujeron en la misma el menú principal por completo, la función completa de petición de productos y la comprobación del stock. Esto se consiguió utilizando variables destinadas a la invisibilidad de los diferentes botones consiguiendo que solo pudiéramos observar los botones requeridos en cada momento. En la siguiente imagen se puede observar esta primera visualización.

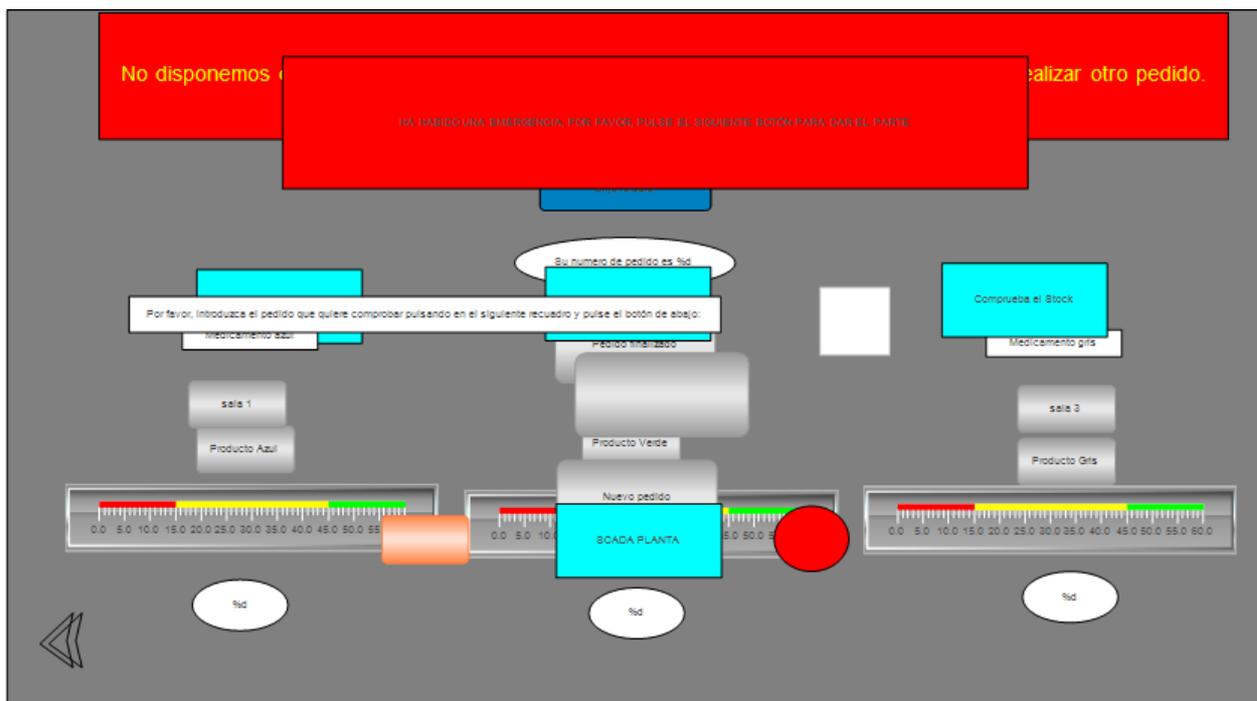


Figura 5-1: Visualización principal CODESYS

De la misma forma, para no necesitar demasiadas variables de invisibilidad y no saturar la visualización, se han creado visualizaciones adicionales a las cuales se pueden acceder con un click sobre unos botones en específico. Un ejemplo de esto es al acceder a la opción SCADA PLANTA en la cual se puede observar una planta de la zona lógica con las partes en funcionamiento. Esto hace que nos aparezca una visualización diferente a la inicial.

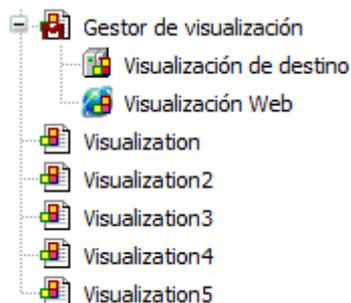


Figura 5-2: Lista visualizaciones CODESYS

Una de las posibilidades que nos permite el software Codesys es la opción de poder tener acceso a la SCADA del programa y por lo tanto ver el estado de la planta y poder realizar pedidos desde el móvil o el PC con conexión vía wifi o USB. Aquí comprobaremos la posibilidad de ver el SCADA desde el localhost del PC. Para hacer esto solo necesitamos cambiar los parámetros como se puede observar en la siguiente imagen

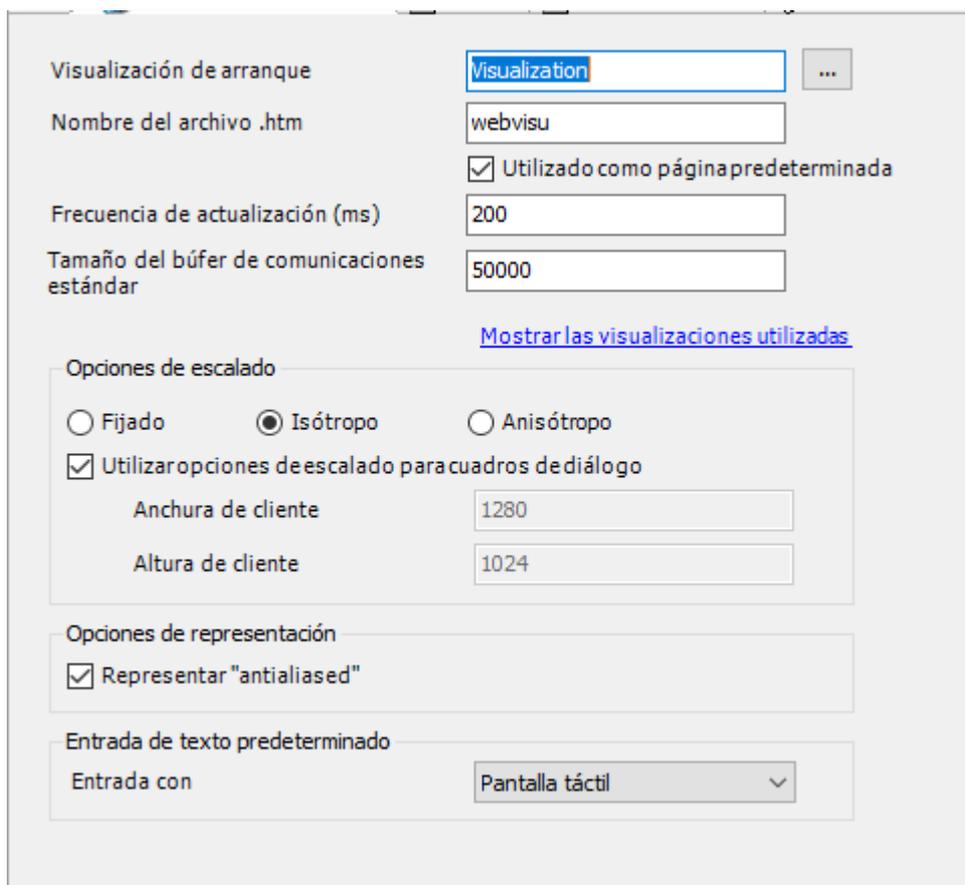


Figura 5-3: Configuración visualización web CODESYS

Una vez cambiados estos parámetros, en el buscador introducimos <http://localhost:8080/webvisu.htm>.

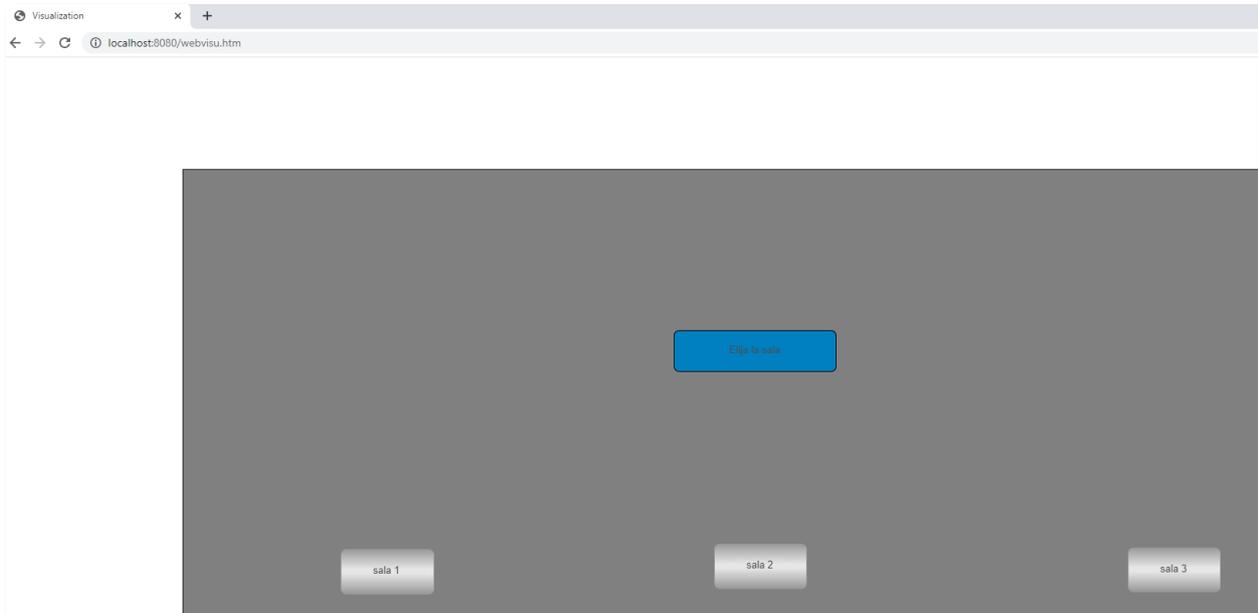


Figura 5-4: Visualización web CODESYS

5.2 Variables y funciones del SCADA

Para conseguir nuestro SCADA, se han utilizado una gran cantidad de variables para diferentes funciones.

Algunas de ellas como la variable “inv1” se han utilizado para volver invisible algunas de las partes que no se quisieran ver en algunos momentos en una misma visualización. Otras se han utilizado como botones para realizar los pedidos de los productos o para informar de la ubicación desde la que han sido pedidos, este es el caso de las variables “PEDIDOAZUL”, la cual quiere decir que se desea un producto azul o “h2” la cual informa de que el producto se quiere recibir en la sala 2.

También ha habido variables que han sido creadas solo para saber en la función “SCADA PLANTA”, cuando se está produciendo un movimiento en una maquinaria, este es el caso de variables como “GIR1”, la cual quiere decir que si se activa es que la base giratoria 1 está en movimiento o, “ELEVADOR”, con la que sabemos cuándo el elevador está en movimiento y cuando no.

Luego también encontraremos otras variables como “VOLVERPAGINAINICIAL” O “PEDIDOFINALIZADO” que se utilizan para la mejor funcionalidad entre visualizaciones del SCADA.

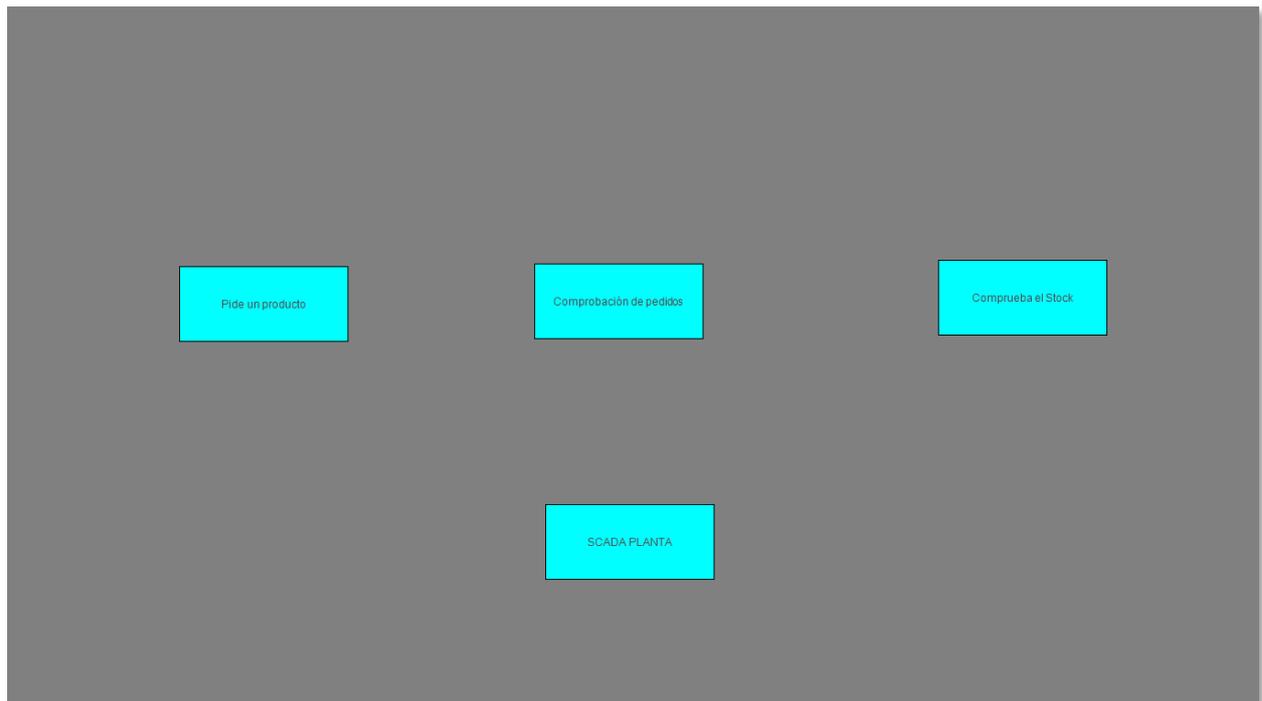


Figura 5-5: Menú principal CODESYS

5.2.1 Pedir un producto

El principal objetivo del proyecto era el poder suministrar diferentes tipos de productos automáticamente a diferentes zonas según un orden y con un control de los pedidos.

Para realizar estos pedidos hemos establecido una función asociada a un botón en el menú principal la cual hemos denominado “Pide un producto”. Al seleccionar esta función, nos saldrá otra pantalla donde podremos seleccionar el tipo de producto que deseamos solicitar.

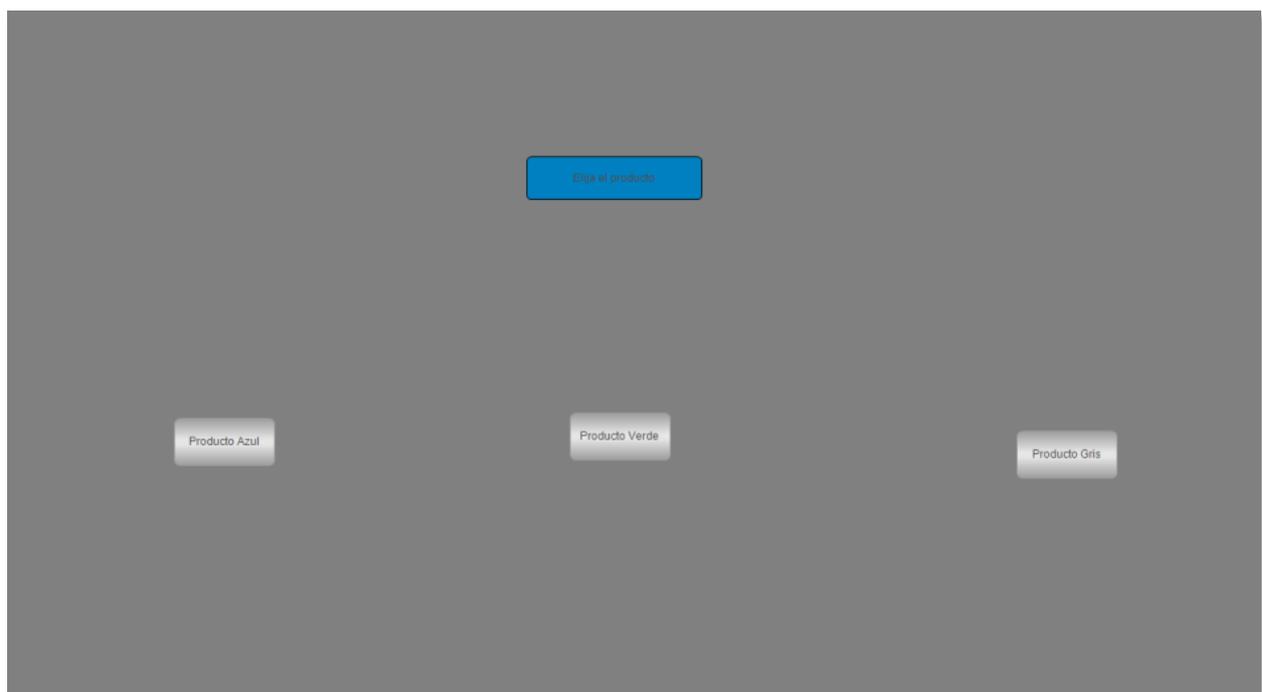


Figura 5-6: Elección de producto CODESYS

Al seleccionarlo, el programa comprobará si hay stock de ese producto y si no lo hay nos devolverá un mensaje de error, el cual aparece en la siguiente imagen. Después de esto, podremos seleccionar un nuevo botón denominado “Nuevo pedido”, el cual nos devolverá al menú principal.

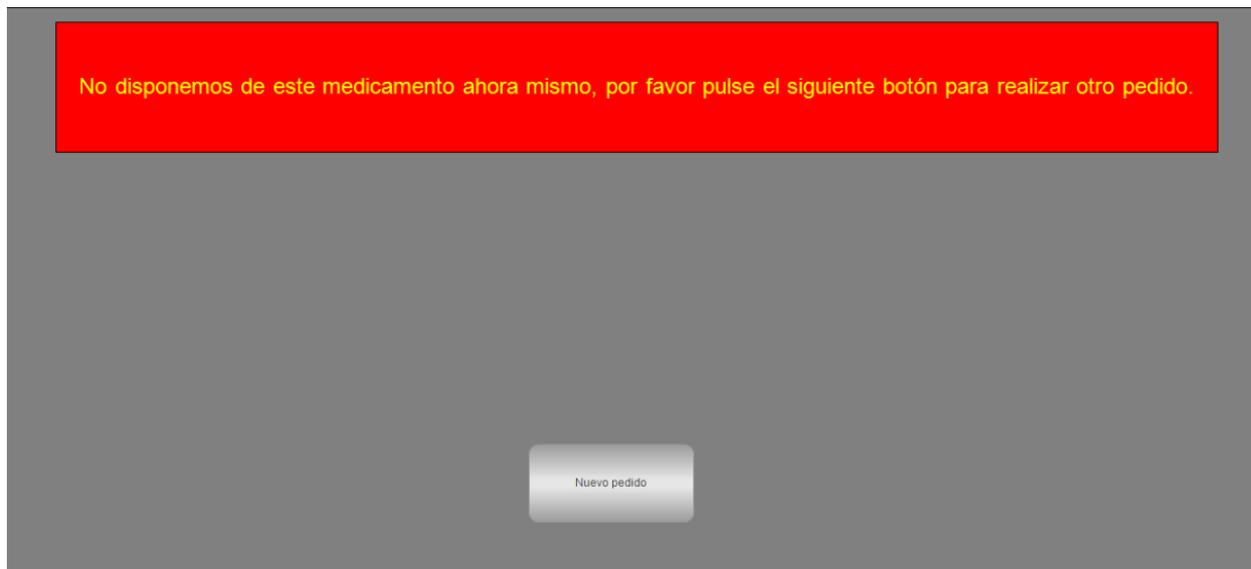


Figura 5-7: Error disponibilidad de stock CODESYS

Si hay stock, nos devolverá otra pantalla para elegir la sala a la que queremos que nos mandé el producto. Al seleccionarla nos aparecerá otra pantalla donde Después de esto el programa guardará

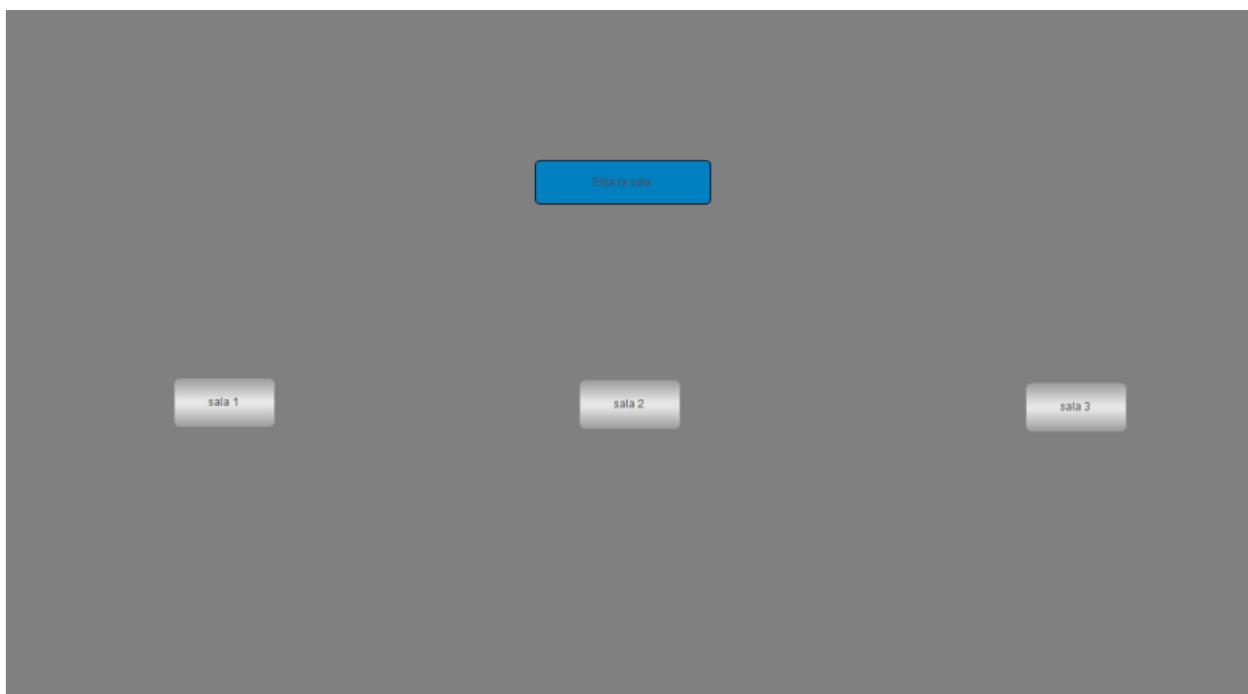


Figura 5-8: Elección de sala CODESYS

Al seleccionar la ubicación del pedido, el programa guardará la información del mismo junto con un número de pedido para rastrearlo. Este número nos saldrá al terminar el pedido junto a un botón denominado “Pedido finalizado”, el cual nos devolverá al menú principal.



Figura 5-9: Número de pedido CODESYS

5.2.2 Comprobación del Stock

Otra función importante que ha sido integrada es la denominada “Comprobación de stock”. Esta nos permitirá ver el nivel de stock de cada producto, información importante tanto para el que quiere realizar un pedido como para los encargados de la planta. Esta información aparecerá enumerada por productos junto con una gráfica que, aunque aparezca como número máximo el 60, en realidad será el 54, el cual es la capacidad máxima de cada estantería.

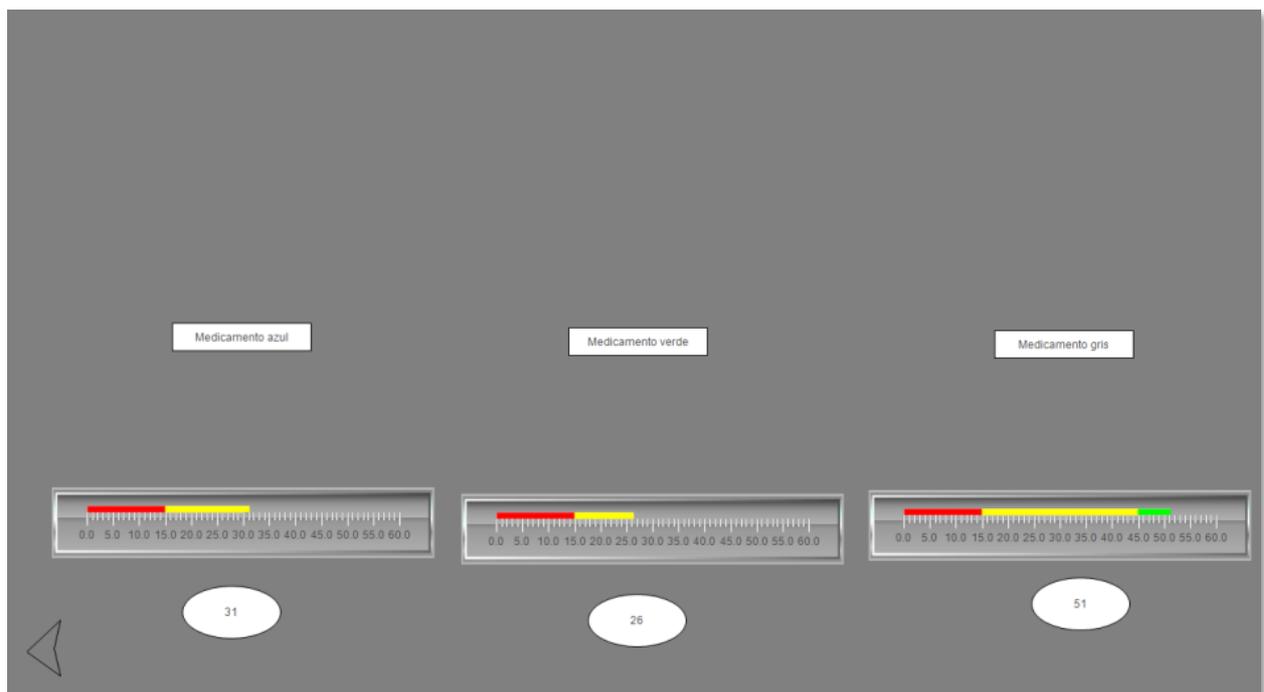


Figura 5-10: Nivel stock CODESYS

5.2.3 Comprobación de pedidos

Otra función importante para el cliente es la denominada “Comprobación de pedidos”. Esta nos permitirá, gracias al número del pedido que nos ha dado antes el programa, informarnos sobre el estado de nuestro pedido, entregado o no entregado. Se introducirá el número del pedido pulsando sobre el recuadro blanco de la siguiente imagen y después se pulsará el botón naranja situado en la parte inferior. Si la elipse se pone de color verde, el producto habrá sido entregado, si se pone rojo, el producto no ha sido entregado.



Figura 5-11: Comprobación pedidos CODESYS

En la siguiente imagen podemos ver el teclado emergente que nos saldría al pulsar sobre el recuadro blanco. En este se introduciría el número del pedido y se pulsaría el botón de “OK”.

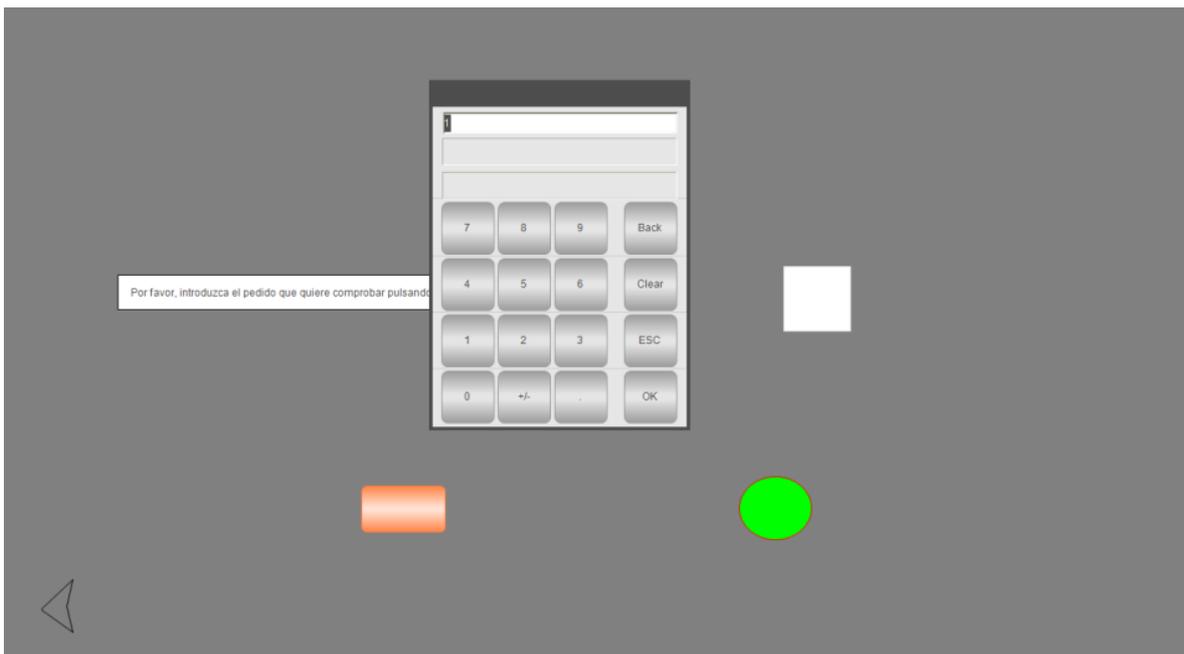


Figura 5-12: Teclado comprobación pedidos CODESYS

5.2.4 Observación de la SCADA de la planta

Una de las mayores necesidades para un operario de la planta es la de comprobar el estado de la misma. Esto se ha conseguido introduciendo otra función la cual se ha denominado “SCADA PLANTA” y con a que podremos ver que partes de la planta están en movimiento y el número de productos en cada cinta y en cada estantería. También se ha introducido un medidor situado en la parte superior izquierda el cual nos dice la situación del producto final debajo de la mano de la grúa. Cuando la aguja apunta hacia la zona verde, significa que el producto final está bien situado.

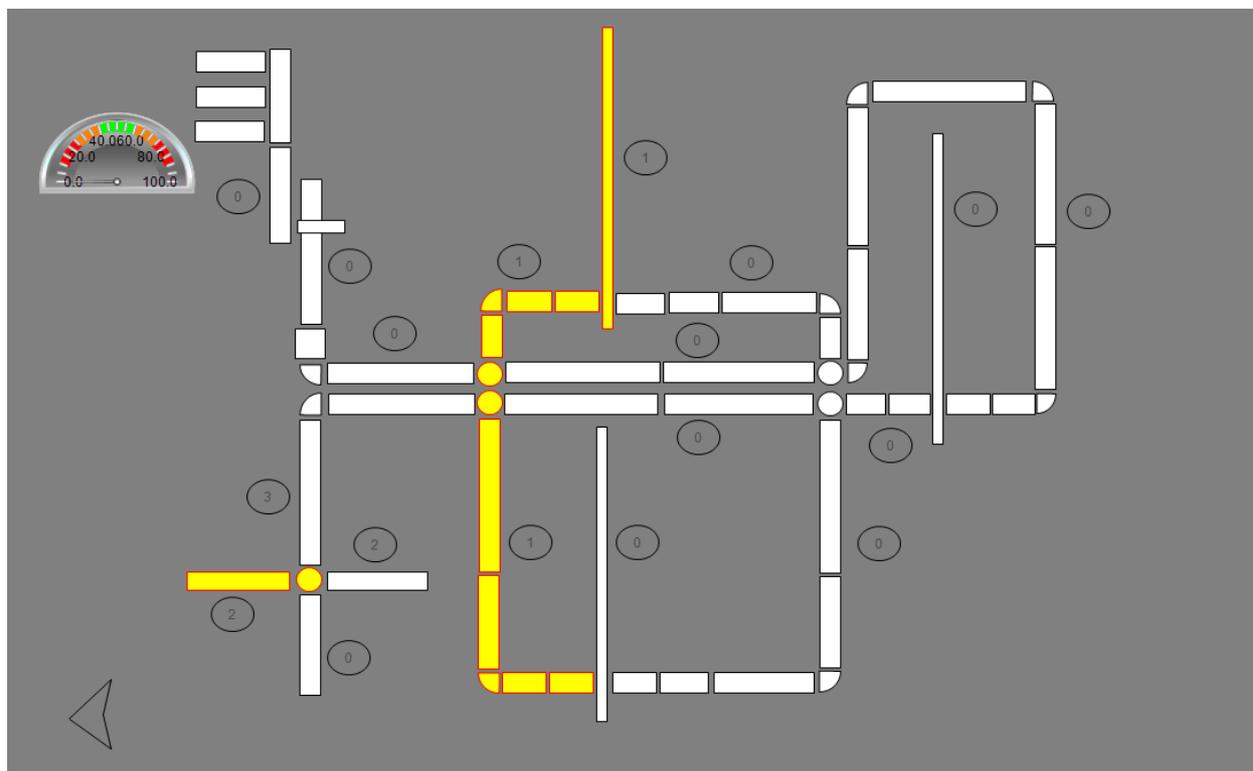


Figura 5-13: Comprobación estado planta CODESYS

5.2.5 Resolución de errores

La última de las funciones se ha implementado debido de algunos fallos que surgían durante la simulación del proyecto como desvíos de productos en las cintas o golpes debidos al delay entre los programas.

Esta función tiene dos objetivos. Uno es el de hacer que cuando surge un error el cliente no tenga la opción de pedir productos hasta que se ha arreglado. El otro es el dar parte del error y llevar la cuenta de los productos tanto de los enviados como de los recibidos.

En la imagen siguiente se puede observar la pantalla que emerge cuando se pulsa el botón de parada de emergencia debido a un error.

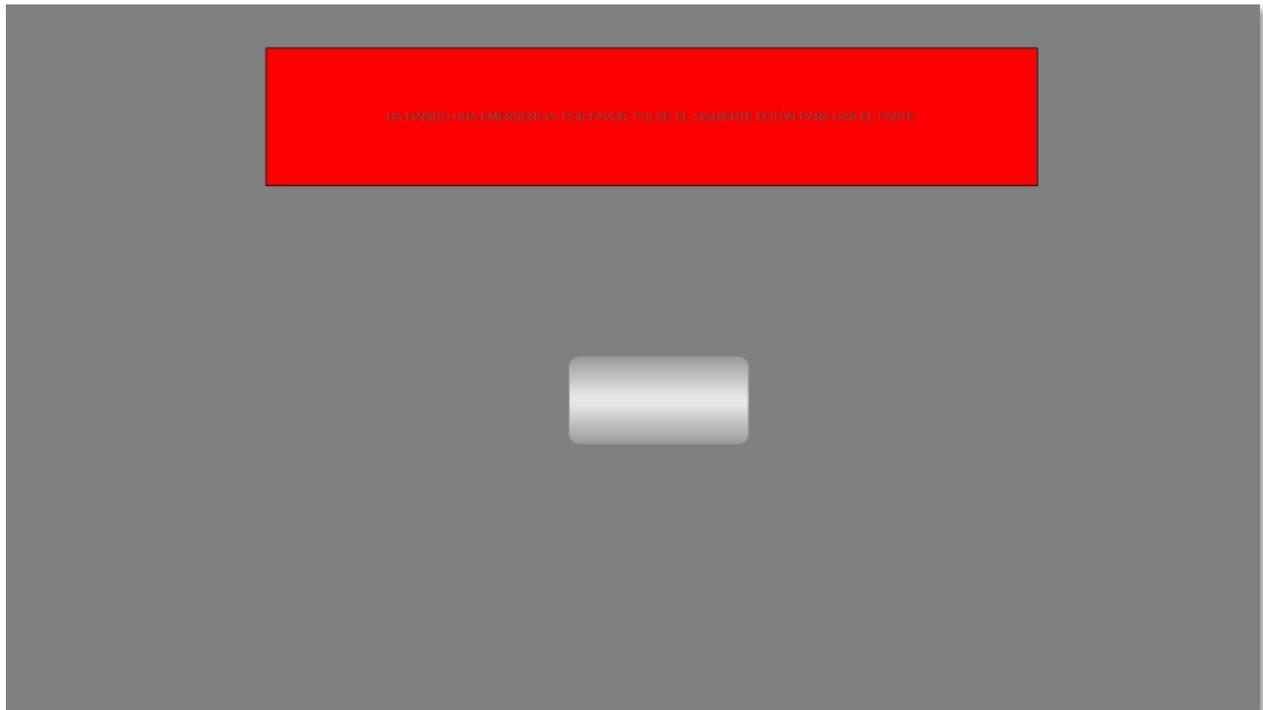


Figura 5-14: Error fallo en la planta CODESYS

Cuando se pulse el botón de la imagen anterior, el programa nos enseñará una gráfica de la planta donde distinguiremos una zona roja y otra blanca. El operario encargado del error deberá seleccionar la zona donde ha ocurrido el error.

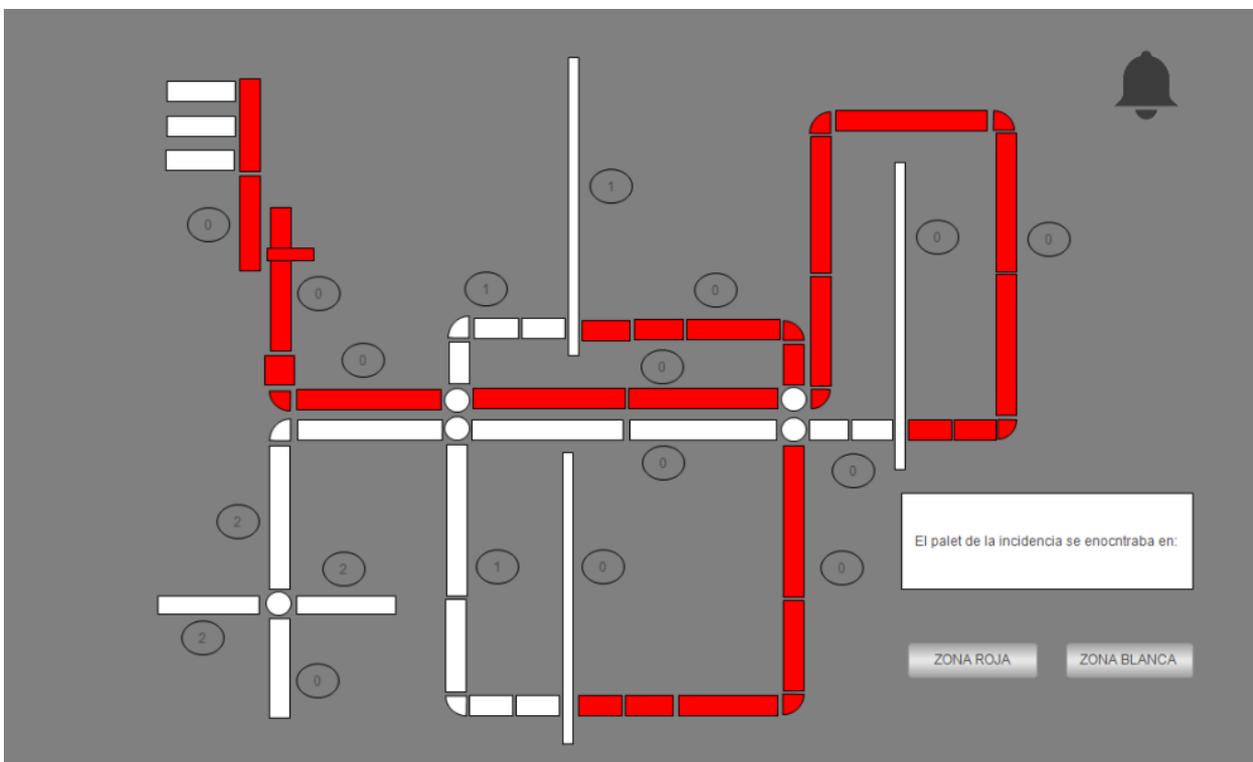


Figura 5-15: Elección zona del error fallo en la planta CODESYS

Para que no haya exceso de productos tanto recibidos por los proveedores o que no haya falta de productos

enviados a los clientes, se lleva una cuenta exhaustiva de estos productos. Debido a esto es por lo que se han diferenciado dos zonas.

Si un producto se encontraba en la zona blanca es debido a que era un producto que había sido recibido del proveedor y se estaba enviando a las estanterías para almacenarlo. Este producto estaba dentro de las cuantas para no exceder el número de productos de ese tipo que cabían en su estantería por lo que, si el error había causado que un producto en esta zona fuera retirado, este debía descontarse de estas cuentas para pedir más al proveedor y volver a ocupar el total de los espacios de las estanterías. Debido a esto se nos da la opción de decir el tipo de producto que era y cuantos productos han debido de ser retirados. Si no hubiera hecho falta retirar ningún producto se pulsa el botón que se ve en la parte inferior de la siguiente imagen y nos devolvería al menú principal.

SE TRATA DE UN PRODUCTO PROCEDENTE DE LA ZONA DE PRODUCTOS RECIBIDOS

POR FAVOR SEÑALE EL TIPO DE PRODUCTO QUE ERA Y PÚLSELO UN NÚMERO DE VECES EQUIVALENTE AL NÚMERO DE PALETS APARTADOS

PRODUCTO AZUL

PRODUCTO VERDE

PRODUCTO PLATEADO

CUANDO HAYA TERMINADO PULSE EL SIGUIENTE BOTÓN

[Botón gris]

Figura 5-16: Elección tipo de producto recibido origen error CODESYS

La zona roja es la zona de aquellos productos que ya han sido enviados a los clientes por lo que todo producto retirado de esta zona debe tenerse en cuenta en el número de productos pedidos para que no hay un cliente que se quede sin su producto. Debido a esto se nos presentaría otra pantalla como la anterior donde podremos seleccionar el tipo del producto y el número de productos retirados para sumarlos al contador encargado de solicitar los productos a cada almacén. Esta pantalla se puede observar en la siguiente imagen.

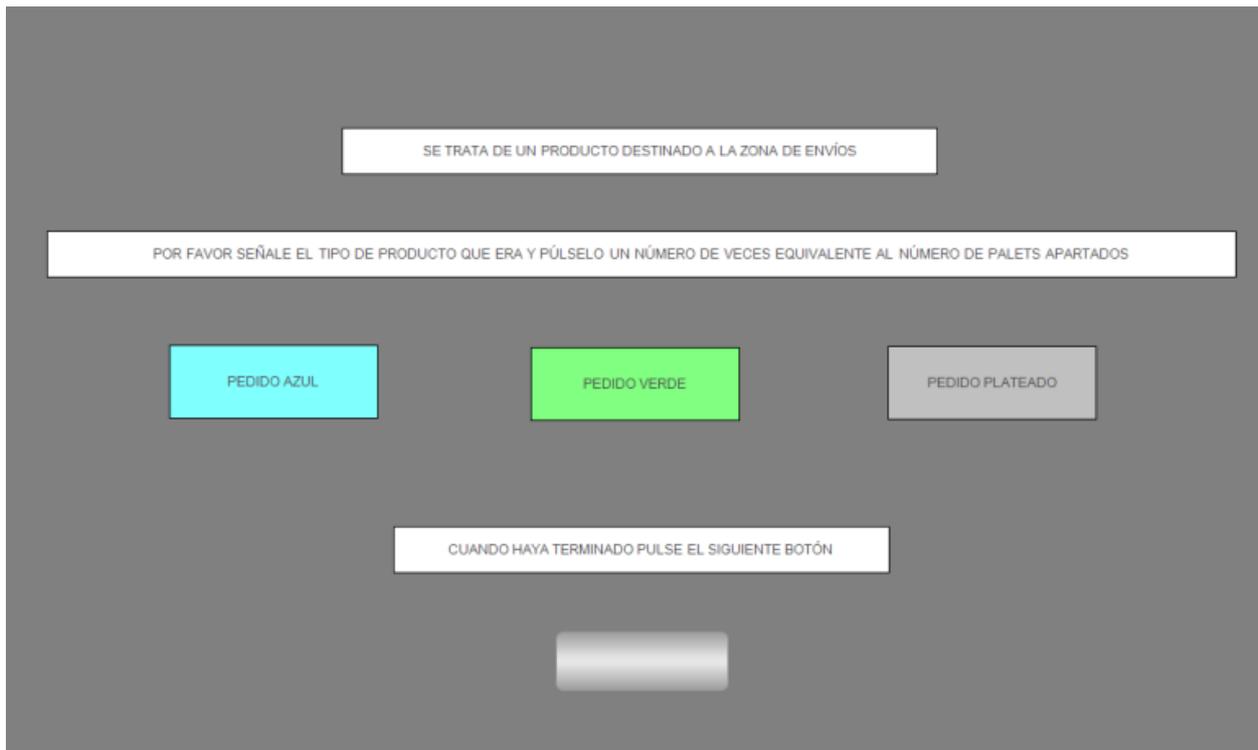


Figura 5-17: Elección tipo de producto enviado origen error CODESYS

5.3 Modos de control

Cualquier Sistema o Proyecto automatizado debe tener siempre unos modos de control establecidos ya que son necesarios para cualquier tipo de situación. Por ejemplo, si durante un proceso automático surgiera un error, la intención sería no tener que apagar el sistema y que, pulsando un botón situado cerca de la zona de error, se pueda pausar el programa y solucionar el problema para volver a ponerse en marcha en cualquier momento sin perder las situaciones antes del fallo.

Se han establecido diferentes modos de control para cada momento logístico

Para empezar un día cualquiera en el que se necesite el almacén automatizado se inicia el programa y se pulsa cualquier botón de automático de cualquiera de las tres zonas. A partir de ese momento el programa se encuentra automatizado y todo sigue un orden establecido. Si se necesita una parada, tanto de emergencia como una parada normal se pulsa o el botón de parada de emergencia o el de parada normal pero las dos funcionan similares menos por un detalle que veremos a continuación.

Luego si se quiere salir de la parada de emergencia, se deberá pulsar de nuevo el botón de emergencia de la misma zona de la que se haya pulsado antes y luego se deberá pulsar el botón de automático de la misma zona que se haya pulsado antes y se volvería a la situación inicial. De la parada normal se podrán tomar dos caminos: uno, que el botón de parada se hubiera pulsado para comprobar algo y se pueda seguir trabajando solamente pulsando el botón de automático de la misma zona del botón de parada. O dos, que se haya parado la producción por el final de un día laboral y se quiera limpiar el almacén por completo. Entonces se pulsaría el botón de rearme para poner todos los actuadores de cintas en activo junto con un POU dedicado a la extracción de productos de las estanterías y los eliminadores de productos. También se quedarían en inactivo los emisores de productos.

En la imagen siguiente se observan las fases de automatización y de parada de emergencia con la intención de volver al estado automático.

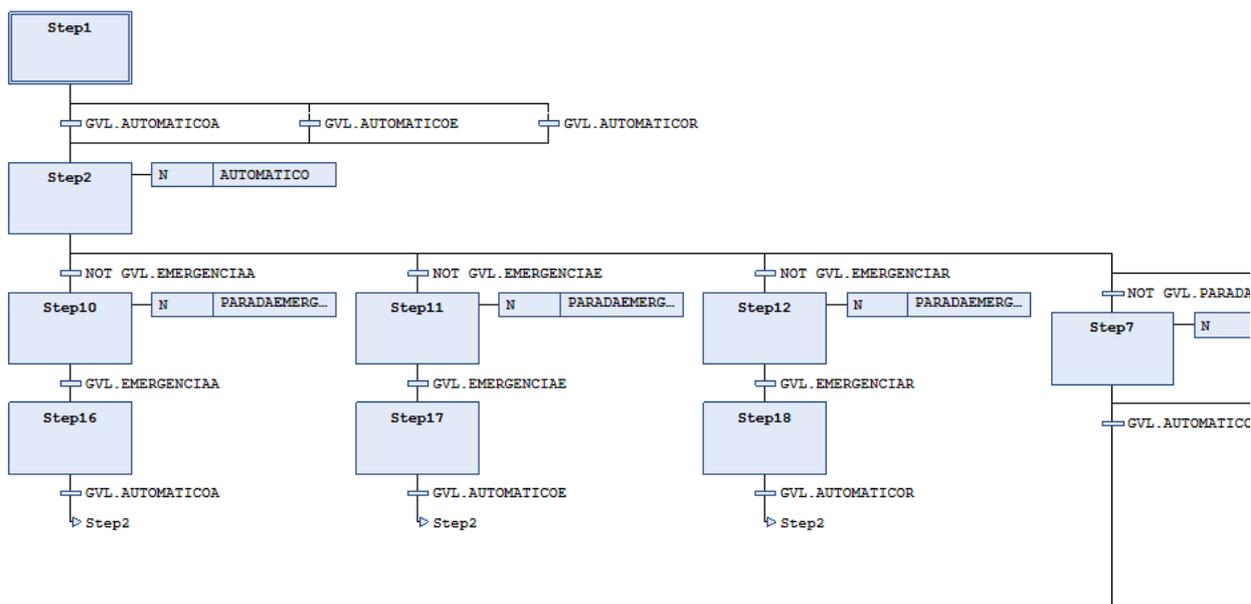


Figura 5-18: POU MODOFUNCIONAMIENTO 1 CODESYS

Y en la de ahora, se observan las fases de parada para la próxima elección de volver a la automatización o terminar la misma con un rearme del almacén

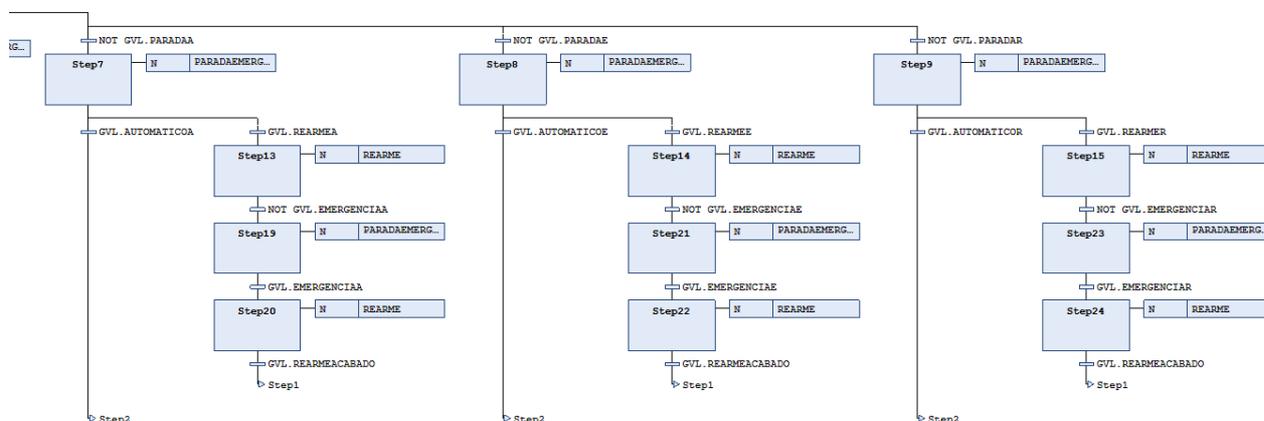


Figura 5-19: POU MODOFUNCIONAMIENTO 2 CODESYS

5.3.1 Automático

Este apartado es el principal del proyecto ya que cuando nos situamos en él, el programa es capaz de funcionar de forma independiente para resolver la función para la cual ha sido programado. Durante este modo de control, se llamarán al resto de POU's menos a "ALMACENESREARME", "AlmacenAREARME", "AlmacenPREARME", "AlmacenVREARME", "REARME", "PARADAEMERGENCIA" y "AUTOMATICO". En la siguiente imagen se pueden observar el resto de POU's.

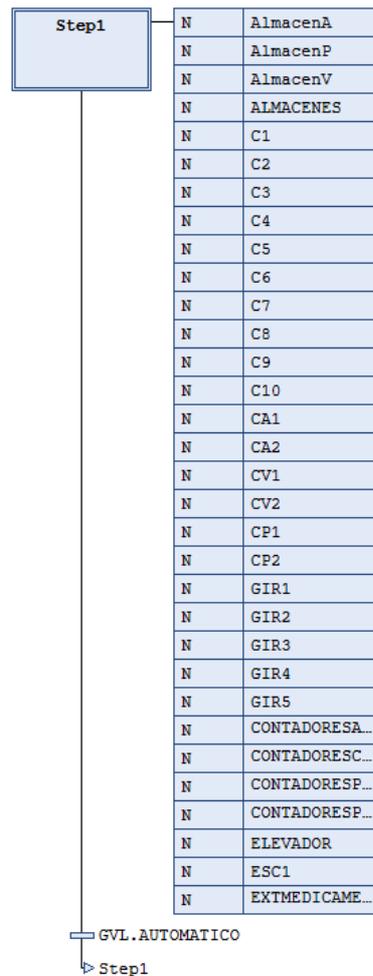


Figura 5-20: POU AUTOMÁTICO CODESYS

5.3.2 Parada de emergencia y parada

Este modo de control tiene dos posibilidades, que haya sido llegado después de que se haya observado un error y se haya pulsado el botón de emergencia que se haya realizado una parada para o, volver al automatismo o, realizar un rearme en el almacén.

De cualquiera de los dos casos las POUS del automatismo se desactivan y se apagan todos los actuadores para conseguir un estado de cero movimientos. Esto se observa en la siguiente imagen.

```
GVL.RC1:=FALSE;  
GVL.RC2:=FALSE;  
GVL.RC3:=FALSE;  
GVL.RC4:=FALSE;  
GVL.RC5:=FALSE;  
GVL.RC6:=FALSE;  
GVL.AC1:=FALSE;  
GVL.AC2:=FALSE;  
GVL.AC3:=FALSE;  
GVL.AC4:=FALSE;  
GVL.CV1:=FALSE;  
GVL.CV2:=FALSE;  
GVL.CV3:=FALSE;  
GVL.CV4:=FALSE;  
GVL.CV5:=FALSE;  
GVL.CV6:=FALSE;  
GVL.CV7:=FALSE;  
GVL.CV8:=FALSE;  
GVL.CV9:=FALSE;  
GVL.CV10:=FALSE;  
GVL.CV11:=FALSE;  
GVL.CP1:=FALSE;  
GVL.CP2:=FALSE;  
GVL.CP3:=FALSE;  
GVL.CP4:=FALSE;  
GVL.CP5:=FALSE;  
GVL.CP6:=FALSE;  
GVL.CP7:=FALSE;  
GVL.CP8:=FALSE;  
GVL.CP9:=FALSE;  
GVL.CP10:=FALSE;  
GVL.CP11:=FALSE;  
GVL.CP12:=FALSE;  
GVL.CP13:=FALSE;  
GVL.CA1:=FALSE;  
GVL.CA2:=FALSE;  
GVL.CA3:=FALSE;  
GVL.CA4:=FALSE;  
GVL.CA5:=FALSE;  
GVL.CA6:=FALSE;  
GVL.CA7:=FALSE;  
GVL.CA8:=FALSE;  
GVL.CA9:=FALSE;  
GVL.EC1:=FALSE;  
GVL.EC2:=FALSE;  
GVL.EC3MAS:=FALSE;  
GVL.EC3MENOS:=FALSE;
```

Figura 5-21: POU PARADAEMERGENCIA CODESYS

Si se ha parado de emergencia se deberá hacer un formulario sobre el fallo por el SCADA como se ha comentado anteriormente y si se ha parado de forma normal, se puede continuar como se quiera.

5.3.3 Rearme

El rearme es el modo de control que se necesita para volver al almacén a su estado normal e inicial. Con este modo, se consigue vaciarlo completamente, volviendo las variables a su estado inicial y reseteando todos los posibles contadores.

Este modo tiene en cuenta que los POU's anteriormente utilizados deben volver a su estado inicial. Esto se consigue con la una variable de estado específica para cada POU. Se denomina SFCInit y hace que ese POU vuelva a su primera acción, aunque a la hora de la parada para el rearme estuviera en medio de un ciclo de acción. Esta variable es llamada en el POU denominado "REARME" y debe ser llamada con el POU al que está destinada ya que sino solo variaría su valor global pero no le afectaría.

Una vez desactivadas estas POU's se procede a activar tanto las cintas del almacén como los eliminadores de productos, los cuales se han situado al final de cada zona hasta llegar a la zona de envío de productos, como las POU's destinadas al rearme de toda la zona logística las cuales serían las POU's "AlmacenAREARME", "AlmacenVREARME" y "AlmacenPREARME" las cuales consiguen vaciar al completo las estanterías.

La zona de productos se queda libre del rearme ya que en esta zona solo se encontrarían productos que están a punto de llegar al cliente por lo que se les permitiría llegar a la ubicación deseada. Aun así para mejorar todo el sistema, lo mejor sería establecer horarios para poder pedir productos y gracias a los contadores el encargado de planta sabría cuando ya no hay más productos para enviar y podrá llamar al rearme sin preocupación ya que este si afecta a productos que han sido enviados pero todavía están en la zona de almacenaje. También se desactivarían los pedidos a los proveedores parando así el flujo de productos.

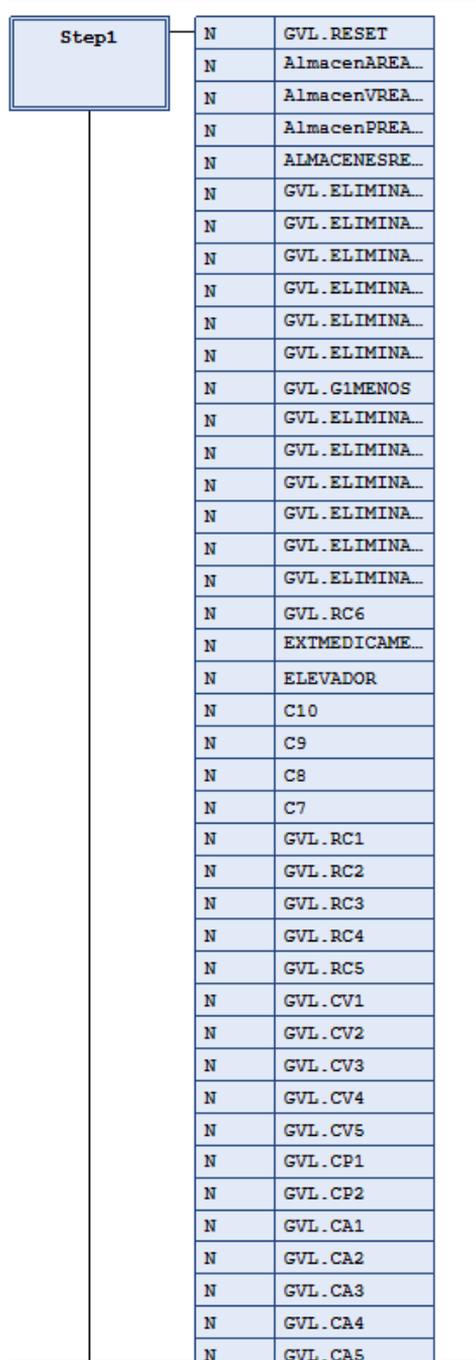


Figura 5-22: POU REARME CODESYS

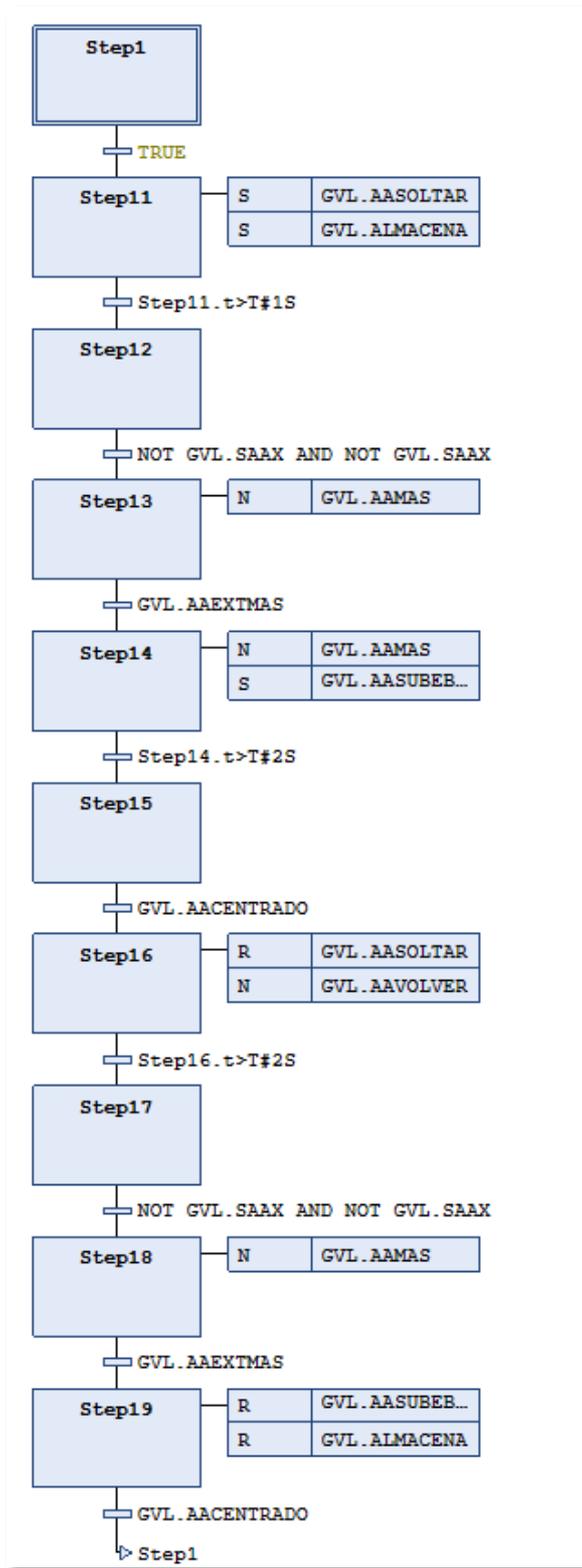


Figura 5-23: POU AlmacenAREARME CODESYS

Una vez ha pasado el tiempo que se ha establecido para completar el rearme , se supone que los almacenes están vaciados y no queda ningún producto en ninguna de las cintas y en ninguna de las zonas. Por lo que al terminar se llaman a las POUs las cuales se han desactivado con anterioridad y se les vuelve a llamar junto con la desactivación de las variables SFCInit para que vuelvan a poder evolucionar con normalidad cuando se

inicie de nuevo el programa.

Este sistema permite dos formas de volver a iniciar el programa cuando se retomara la actividad. Una sería habiendo parado el programa al final de la actividad anterior y simplemente reanudándola ya que todo se habría quedado en orden. Otra sería apagar y iniciar el programa desde cero. Entre las dos se considera más útil la primera opción ya que conseguimos almacenar datos que de la otra forma se borrarían.

6 PROBLEMAS OBSERVADOS, ANTECEDENTES Y MEJORAS

A la hora de realizar el proyecto han surgido diferentes problemas, tanto de diseño como de funcionalidad que han influido a la hora de realizar el mismo. Se comentarán tanto problemas que han surgido al ser una simulación como problemas que podrían surgir aun siendo en la realidad surgir en la realidad.

Por lo tanto, comentaremos en primer lugar aquellos que surgieron por problemas con los softwares y después el resto:

Salida de las bases giratorias

Uno de los problemas que más han repercutido en el proyecto ha sido un fallo en las físicas del software Factory IO. Este problema hacía que un pallet, al salir de una base giratoria o de una cinta automática en curva, se girara levemente hacia alguno de los lados. En principio este movimiento no perjudica a la simulación, pero al sumar varios fallos similares, el pallet puede llegar a chocar con sensores, con los laterales de las cintas o salirse de las mismas. Este sería un fallo que se podría considerar normal en la vida real, pero el problema surge cuando se convierte en algo cíclico más que en algo aleatorio como es en la realidad.

Debido a esto se han considerado diferentes soluciones. La primera que se consideró inicialmente, la cual fue una medida preventiva, fue la de colocar barreras a los laterales de las cintas para que los pallets no tuvieran opción de girarse. El problema de esta solución es que hay pocos tipos de barreras y de pocos tamaños diferentes y muchas no han servido dependiendo de la zona del fallo. Al ver que el error no se corregía, se planteó la implantación de medidas correctivas. Una solución, y la cual hubiera se hubiera implantado en el proyecto al margen de este error, es la de agregar un modo de funcionamiento llamado “PARADA DE EMERGENCIA”. Gracias a este modo, podemos parar el funcionamiento de nuestra planta a la vista de cualquier error, comunicarlo y corregirlo, consiguiendo así tener un control a la hora de estos errores.



Figura 6-1: Barreras protectoras 1
FACTORY IO

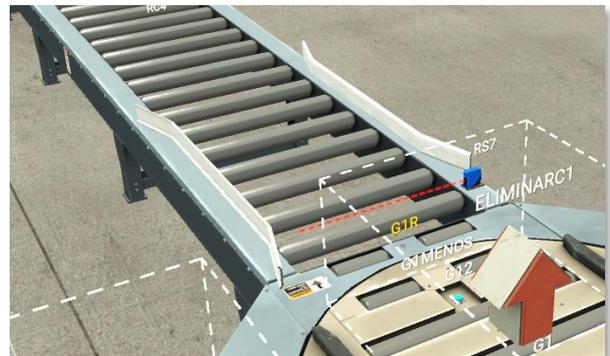


Figura 6-2: Barreras protectoras 2
FACTORY IO

Limitación del número de variables

Un problema de limitación fue comprobar que hay un número máximo de variables que se pueden enlazar desde Factory IO. Este era de 256 y aunque sean bastantes, eliminaba posibilidades de implantación de otras características a nuestra planta. Una de ellas y la cual se consideró antes de conocer este límite, fue la de añadir otro piso a la zona de “Entrega de productos”. En la siguiente imagen se puede ver cómo podría haber acabado la planta sin esta limitación.

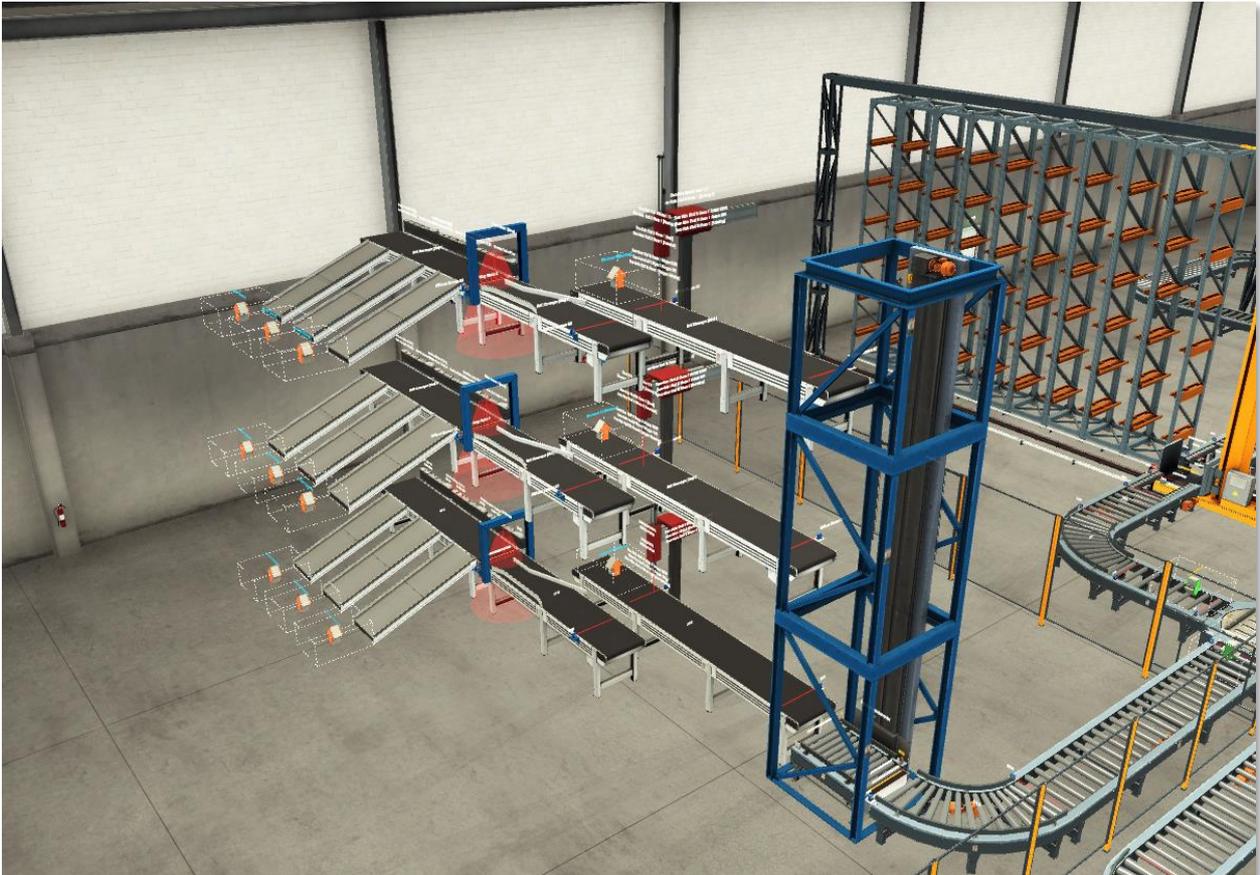


Figura 6-3: Intento planta 2 y 3 FACTORY IO

Botones pillados del scada

Un de los principales fallos encontrados durante la realización del proyecto ha sido un problema con los botones del scada. Este problema consistía y consiste en que, al pulsar un botón de nuestra visualización, los cuales están programados para realizar un flanco de salida instantáneo, se quedaban pillados en la posición de activos y esto hacía que la visualización entera quedara pillada. Durante la prueba con diferentes botones, se ha llegado a la conclusión de que uno de los orígenes de estos problemas era la pantalla de visualización que se nos crea al activar la aplicación CODESYS Control Win V3 necesaria para la conexión entre softwares. Debido a esto se recomienda eliminar esta pestaña al iniciar el programa. Aun así, hay veces que se vuelven a quedar pillados, aunque con muy poca frecuencia, pero no se ha llegado a encontrar el origen del error.

Conseguir flancos de salida instantáneos

Uno de los problemas encontrados fue el de conseguir flancos de salida instantáneos. Esto era necesario para momentos en los que, con la activación de una variable, solo se quería realizar un bucle una sola vez. Por ejemplo, con la activación de variables como “AASOLTAR”, la cual se activaba cuando se quería mandar un producto a los clientes desde la estantería azul, se quería iniciar un bucle que viera cual era la primera balda libre de la estantería. Este bucle solo se quería iniciar una sola vez por lo que al final se optó por utilizar unas variables del tipo R_STRIG con las cuales se conseguía este propósito.

Poca funcionalidad de la cinta sin rodillos para los pallets

Otro problema que se tuvo con las limitaciones de la maquinaria fue el uso de cintas sin rodillo ya que éstas solo permiten el transporte de productos finales sin pallets. Además de esto, también causó problemas que estas cintas fueran levemente más altas que las de rodillos por lo que no se podía conectar excepto con el elevador. Debido a esto solo se han utilizado en la zona de envíos de productos para conseguir solo utilizarlas en zonas donde solo tuvieran que transportar productos finales.

Unión de variables

La unión de variables entre CODESYS y FACTORY IO, se realiza al iniciar el programa de CODESYS y FACTORY mediante la explicación realizada anteriormente. Un problema que surgió inicialmente fue la constante conexión entre variables que hacía que se perdiera bastante tiempo. Trabajando con ello se observó que las conexiones se podían guardar en el programa y no se tenían que enlazar en cada conexión. Esto era así hasta que se actualizaba el número de variables en CODESYS que se querían enlazar con FACTORY IO, ya que estas no aparecían hasta que se reiniciaba la búsqueda de esta y se perdían las conexiones.

Modos de funcionamiento

Para la realización de los modos de funcionamiento se han observado diferentes problemas. Uno de ellos fue el inicio de POU's cuando se rearmaba el mecanismo. Al realizar este modo de funcionamiento se observaban, al poner en marcha el modo automático de nuevo, que había POU's que se habían quedado con el ciclo iniciado de la vez anterior. Esto se resolvió utilizando las variables asociadas a cada POU con las cuales se reiniciaba el ciclo. Estas variables se completan con el nombre de la POU y la terminación SFCReset o SFCInit según cual sea más útil para cada caso. Otro problema observado fue cuando se realizaba un rearme con un objeto enviado, el cual a no ser que estuviera en la zona de envío de productos, no llegaba a su zona de destino. Debido a esto, se recomienda la inclusión de horarios de forma que cuando se acabe este, no se puedan realizar más pedidos y que pasado un tiempo, se sepa con certeza que no haya ningún producto pedido en la planta.

Delay entre softwares

La simulación en FACTORY IO tiene la posibilidad de cambiar la velocidad de las físicas del proyecto. Esto ayuda a la hora de realizar pruebas del mismo, pero también ocurría que se creaba delay entre softwares con los cuales se causaban fallos en la planta como golpes entre productos o bases giratorias separadas. Debido a esto, la máxima velocidad recomendada para el proyecto es la "x2" dentro de FACTORY IO.

7 RESULTADOS FINALES Y CONCLUSIONES

Cada vez son más las empresas e industrias que buscan la innovación y el perfecto control en sus sistemas de almacenaje. Algunas las cuales basan su sistema de picking en satisfacer pedidos en zonas cercanas al almacén de forma rápida y eficiente son las que más pueden llegar a necesitar formas de automatización como la referida en este proyecto.

Este ha sido un proyecto basado en satisfacer estas necesidades de una forma sencilla, con un objetivo y unas características adaptadas al mismo. Para cada tipo de proyecto se usan máquinas diferentes, se procesan pedidos diferentes y se establece un control diferente por lo que es importante ser preciso en cada uno y hacer un estudio previo del mismo.

Realizar este proyecto con la simulación 3D ha hecho que vea como de útil puede llegar a ser la implementación de programación en almacenes y como de autónomo pueda llegar a serlo consiguiendo así que una persona detrás de un ordenador, después de haber realizado el trabajo de implantación de sistemas físicos, pueda llegar a encargarse del trabajo que realizarían un grupo de operarios ubicados físicamente en el almacén. Y no solo una implantación inicial, sino ser capaz de variarla de forma que, en cualquier momento de la vida útil del almacén, este se pueda volver a programar de forma diferente según las peticiones de la empresa encargada.

Para terminar, comentar que se ha intentado explicar lo máximo posible el funcionamiento del proyecto, pero sin lugar a dudas no tiene comparación a ver el funcionamiento del programa hecho en CODESYS y el hecho en FACTORY IO. Debido a esto, incluiré unos enlaces en los anexos finales donde se podrán descargar los archivos de CODESYS Y FACTORY IO junto con videos del funcionamiento para mejorar el entendimiento del proyecto.

REFERENCIAS

- [1] Codesys, «Definición Codesys» [En línea]. Available: <https://www.codesys.com/the-system.html#:~:text=CODESYS%20is%20a%20software%20platform,convenient%20configuration%20of%20automation%20applications>.
- [2] Factory IO, «Definición Factory IO» [En línea]: <https://docs.factoryio.com/>

ANEXO A: video

<https://drive.google.com/file/d/1LbjICX2b5K8VjIpzTLIDHOItUYRRWTA5/view?usp=sharing>

ANEXO B: Archivo CODESYS

<https://drive.google.com/file/d/1U70cHcYVlnTxzwBrJnDWn69zzCqVcDva/view?usp=sharing>

ANEXO C: Archivo FACTORY IO

<https://drive.google.com/file/d/1nsWS9IgL2cuuh3IcsgzTMjj6kAcV2paz/view?usp=sharing>