



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Development of the automation of an order system  
preparation and its digital twin using the Factory IO  
software

End of Degree Project

Bachelor's Degree in Industrial Engineering

AUTHOR: Torner Cantó, Ángela Alicia

Tutor: Vallés Miquel, Marina

ACADEMIC YEAR: 2023/2024

## AGRADECIMIENTOS:

En primer lugar, expresar un profundo agradecimiento a mi tutora Marina Vallés Miquel por su ayuda y dedicación a lo largo de este trabajo. Su orientación y compromiso han sido fundamentales para el desarrollo y culminación de este proyecto.

Asimismo, quiero agradecer sinceramente a mi tutor Marco Pellegrini, que durante mi estancia en Italia me ha brindado su apoyo y ayuda en todo momento.

No puedo dejar de mencionar el apoyo incondicional de mi familia y amigos. Han estado siempre a mi lado apoyándome y animándome a dar siempre lo mejor de mí misma.

## RESUMEN:

En este proyecto se realiza la automatización mediante PLC de una planta de preparación de pedidos de diferentes productos. Estos productos se empaquetan dependiendo del pedido deseado y un elevador los almacena en una estantería. Después, el elevador puede recogerlos de la estantería y llevarlos a la línea de envío que lo requiera. La planta se ha simulado con el programa Factory IO y la automatización se realiza en lenguaje SFC y Ladder mediante el software llamado CODESYS.

**Palabras clave:** Automatización; PLC; Factory IO; CODESYS

## ABSTRACT:

In this project, the automation of an order preparation plant for different products is carried out by means of PLC. These products are packaged depending on the desired order and an elevator stores them on a shelf. The elevator can then pick them up from the shelf and take them to the sending line that requires it. The plant has been simulated with the Factory IO program and the automation is carried out in SFC and Ladder languages using the software called CODESYS.

**Keywords:** Automation; PLC; Factory IO; CODESYS

## DOCUMENTOS CONTENIDOS EN EL TFG:

1. Memoria
2. Presupuesto
3. Pliego de condiciones

## ÍNDICE:

### DOCUMENTO N°1: MEMORIA

1. RESUMEN EJECUTIVO .....	1
2. OBJETIVO DEL PROYECTO .....	2
3. ANTECEDENTES, MOTIVACIÓN Y JUSTIFICACIÓN ....	3
4. NORMATIVA.....	3
5. ÁMBITO DE APLICACIÓN Y RANGO DE SOLUCIONES	3
6. AUTOMATIZACIÓN.....	3
7. SOFTWARE DE MODELADO .....	4
8. SOFTWARE DE PROGRAMACIÓN.....	5
9. LENGUAJES DE PROGRAMACIÓN .....	5
10. DESCRIPCIÓN DEL PROCESO .....	6
11. ESQUEMA DE FUNCIONAMIENTO DE LA PLANTA ....	8
12. OPERACIÓN Y CONTROL DE LA INSTALACIÓN .....	8
13. LIMITACIONES DEL PROYECTO.....	9
13.1. Limitaciones de los productos .....	9
13.2. Limitación almacenamiento .....	10
13.3. Limitación cierre de paquetes.....	10
13.4. Limitaciones tránsitos de entrada y salida.....	11
14. DEFINICIÓN DE LAS VARIABLES.....	11
14.1. Sensores .....	11
14.2. Actuadores .....	13
15. DISEÑO DE LA PLANTA .....	14
15.1. Elementos línea de almacenaje.....	14

15.2. Descripción línea de almacenaje .....	17
15.3. Elementos línea de recogida y envío .....	19
15.4. Descripción línea de recogida .....	21
16. GRAFCETS Y SFC .....	22
17. MANUAL DEL PROGRAMADOR .....	40
18. CONCLUSIONES .....	48
19. BIBLIOGRAFÍA .....	50

## DOCUMENTO N°2: PRESUPUESTO

1. COSTES DIRECTOS .....	1
1.1. Mano de obra .....	1
1.2. Material .....	1
1.3. Amortización .....	1
2. COSTES INDIRECTOS .....	2
2.1. Electricidad .....	2
2.2. Internet .....	2
3. UNIDADES DE OBRA .....	2
4. PRECIOS DESCOMPUESTOS .....	3
5. PRESUPUESTO FINAL .....	5

## DOCUMENTO N°3: PLIEGO DE CONDICIONES

1. OBJETIVO Y ALCANCE DEL PLIEGO .....	1
2. DOCUMENTOS QUE DEFINEN EL PROYECTO .....	1
3. CONDICIONES GENERALES .....	1
4. CONDICIONES PARTICULARES .....	2
4.1. Técnicas .....	2
4.1.1. Especificaciones de Hardware .....	2
4.1.2. Especificaciones de Software .....	2
4.1.3. Instalación .....	2
4.1.4. Garantía .....	2

4.2.	Facultativas .....	3
4.2.1.	Modificaciones y defectos en el proyecto .....	3
4.2.2.	Plazo de ejecución.....	3
4.2.3.	Recepción provisional .....	3
4.2.4.	Plazo de garantía.....	3
4.2.5.	Recepción definitiva.....	3
4.3.	Económicas.....	3
4.3.1.	Garantías.....	3
4.3.2.	Fianza.....	3
4.3.3.	Presupuesto del proyecto.....	4
4.3.4.	Modificaciones .....	4
4.3.5.	Revisión de precios .....	4
4.3.6.	Pagos.....	4
4.3.7.	Penalizaciones .....	4
4.4.	Legales.....	4
4.4.1.	Responsabilidad del proyectista .....	4
4.4.2.	Responsabilidad del programador .....	4
4.4.3.	Causas de rescisión del contrato.....	4

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

DESARROLLO DE LA AUTOMATIZACIÓN DE UN  
SISTEMA DE PREPARACIÓN DE PEDIDOS Y DEL  
GEMELO DIGITAL DE ESTE MEDIANTE EL  
SOFTWARE FACTORY IO

# MEMORIA

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales



## ÍNDICE MEMORIA:

1. RESUMEN EJECUTIVO .....	1
2. OBJETIVO DEL PROYECTO .....	2
3. ANTECEDENTES, MOTIVACIÓN Y JUSTIFICACIÓN ....	3
4. NORMATIVA.....	3
5. ÁMBITO DE APLICACIÓN Y RANGO DE SOLUCIONES	3
6. AUTOMATIZACIÓN.....	3
7. SOFTWARE DE MODELADO .....	4
8. SOFTWARE DE PROGRAMACIÓN.....	5
9. LENGUAJES DE PROGRAMACIÓN .....	5
10. DESCRIPCIÓN DEL PROCESO .....	6
11. ESQUEMA DE FUNCIONAMIENTO DE LA PLANTA ....	8
12. OPERACIÓN Y CONTROL DE LA INSTALACIÓN .....	8
13. LIMITACIONES DEL PROYECTO.....	9
13.1. Limitaciones de los productos .....	9
13.2. Limitación almacenamiento .....	10
13.3. Limitación cierre de paquetes.....	10
13.4. Limitaciones tránsitos de entrada y salida.....	11
14. DEFINICIÓN DE LAS VARIABLES.....	11
14.1. Sensores.....	11
14.2. Actuadores .....	13
15. DISEÑO DE LA PLANTA .....	14
15.1. Elementos línea de almacenaje.....	14
15.2. Descripción línea de almacenaje .....	17
15.3. Elementos línea de recogida y envío.....	19
15.4. Descripción línea de recogida .....	21
16. GRAFCETS Y SFC.....	22
17. MANUAL DEL PROGRAMADOR .....	40
18. CONCLUSIONES.....	48

19. BIBLIOGRAFÍA .....	50
------------------------	----

# 1. RESUMEN EJECUTIVO

En el mundo de la industria, cada vez más se busca rapidez y eficiencia, así como facilitar en la medida de lo posible el trabajo de las personas. Tareas repetitivas que pueden resultar aburridas o exigentes físicamente para las personas, pueden ser realizadas gracias a la automatización. Concretamente, en este trabajo se ha planteado automatizar una planta de empaquetado y almacenado de pedidos mediante gemelos digitales, técnica que actualmente avanza a un ritmo acelerado. La resolución del problema se alcanza de manera virtual, viendo así desde el ordenador cómo funcionaría la planta, sin poner en riesgo ni la maquinaria ni los empleados.

Antes de realizar la programación de la planta industrial con el software CODESYS se ha trabajado primero manualmente, realizando los diagramas grafcet que describen los procesos a automatizar. Por tanto, hay que recalcar que se sigue la normativa vigente UNE-EN 60848:2022 que define el lenguaje de especificación GRAFCET para diagramas funcionales en secuencia.

La finalidad de este proyecto es facilitar las tareas de los empleados a la hora de empaquetar productos y almacenar paquetes, además de aumentar la rapidez en la que se realizan estas tareas y reducir la cantidad de errores que pueden surgir.

Desde el inicio, se planteó la idea de los gemelos digitales como una posible solución al problema planteado, al ser una técnica compatible con el programa Erasmus del que ha formado parte la autora del proyecto durante la realización. Se estuvo barajando qué tipo de proceso realizar mediante estos softwares, ya que el software de modelado es limitado y se buscaba un proyecto que planteara un proceso lo más ajustado a la realidad posible. Inicialmente se planteó un proceso únicamente de almacenamiento. Sin embargo, esto resultó ser demasiado simple, por lo que se amplió para incluir primero una fase de preparación, en la que se empaquetan una serie de productos. Después del almacenamiento de estos paquetes, se añadió la opción de envío de estos paquetes hacia 2 líneas de envío, donde se encontrarán los camiones de reparto, que se encargarán de realizar el envío. En un principio se quiso también plantear unos modos de funcionamiento que permitieran al operario manejar de forma manual la planta, pero se descartó esta opción por falta de desarrollo de la idea. No se tenía claro en qué parte de la planta sería útil un modo manual.

Los objetivos establecidos inicialmente se han cumplido en su mayoría. Se ha conseguido un prototipo de planta industrial virtual que automatiza un proceso de empaquetado y almacenaje de forma rápida y eficaz, y que permite que las personas se encarguen de realizar tareas más complejas acordes con sus capacidades humanas de razonamiento

Este proyecto se asemeja a instalaciones que ya existen en empresas de la actualidad y que se están implementando cada vez más. Permite observar cómo es posible acelerar los procesos, reducir costes y errores y ayudar a que los trabajadores realicen tareas menos físicas y repetitivas, en un amplio abanico de sectores.

## 2. OBJETIVO DEL PROYECTO

El objetivo del presente trabajo de fin de grado es diseñar y automatizar una planta de preparación de pedidos, empleando la avanzada tecnología de gemelos digitales mediante el software Factory IO, y la automatización a través del software CODESYS.

El sistema diseñado incluirá un proceso de empaquetado de los productos deseados y su almacenaje mediante un elevador. Después, estos pedidos podrán ser solicitados por los operarios para las 2 líneas de envío existentes en la planta, que dirigen los paquetes hacia los camiones de reparto. Este mecanismo automatizado no solo mejorará la rapidez en la preparación de pedidos, sino que también garantizará una mayor precisión en el manejo de productos, reduciendo así errores y tiempos de espera. Gracias a esta tecnología podemos ver cómo funcionaría la planta antes de realizar el proyecto en la vida real, sin comprometer la maquinaria y proporcionando mayor seguridad a los trabajadores, ya que los errores que se pudiesen cometer serían de forma virtual.

Este proyecto busca poner en práctica los conocimientos teóricos y prácticos adquiridos en asignaturas como Tecnología Automática. Además, representa una oportunidad única para aprender a automatizar, programar y controlar procesos industriales mediante el uso de softwares avanzados que no se han utilizado durante el grado.

En cuanto a los Objetivos de Desarrollo Sostenible (ODS), el proyecto está relacionado con 3 de los 17 objetivos:

- ODS 8: Trabajo decente y crecimiento económico. La tecnología de gemelos digitales que se utiliza en este proyecto permite aumentar la seguridad en el lugar de trabajo de los empleados. Además, la automatización de procesos ayuda a mejorar la eficiencia y la productividad, lo que implica crecimiento económico.
- ODS 9: Industria, innovación e infraestructura. En el proyecto se promueven el uso de nuevas tecnologías en la industria y la modernización de las infraestructuras actuales, mediante la implementación de automatismos, el uso de PLCs y de softwares como Factory IO.
- ODS 12: Producción y consumo responsables. Al conseguir un proceso industrial más eficiente, se reduce la cantidad de errores que se cometen, evitando así daños en la maquinaria y el desperdicio de recursos, lo que conduce a una producción más ética y sostenible.

(Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura, 2024)



Ilustración 1: Imagen de los ODS de la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.

### 3. ANTECEDENTES, MOTIVACIÓN Y JUSTIFICACIÓN

En este proyecto se ha profundizado en la automatización industrial basándonos en los conocimientos adquiridos durante el grado. Se ha escogido este tema al ser una de las ramas de la ingeniería que más interesa a la autora.

La mayoría de las industrias en la actualidad hace uso de la automatización por su eficiencia, rapidez y precisión. También el uso de esta tecnología ayuda a evitar que los trabajadores realicen trabajos peligrosos o forzosos y facilitar sus labores. Para su implementación es necesario el testeo, que conlleva la posibilidad de cometer errores a la hora de su realización (Totvs, 2024).

El software Factory IO es una herramienta muy útil que nos permite experimentar con una planta similar a la que se tendría en la realidad sin comprometer la planta real. Al testear de forma digital, los errores no suponen un problema grave. El uso de gemelos digitales está en expansión debido a los continuos avances de las tecnologías y a las necesidades de las industrias, permitiendo al usuario simular la realidad asegurando la seguridad y el ahorro.

Cabe resaltar que se han utilizado estos softwares concretos por el hecho de hacer el proyecto a distancia. Esto ha sido una limitación al no poder hacer uso de las licencias y las instalaciones de la Universidad Politécnica de Valencia. Por ello, se ha investigado cuales eran los softwares que resultaban aptos para la realización del proyecto y cuya licencia era gratis o se podía utilizar en remoto.

### 4. NORMATIVA

A continuación, se menciona la normativa vigente que se ha considerado a la hora de realizar este proyecto:

- UNE-EN 60848:2022: Lenguaje de especificación GRAFCET para diagramas funcionales en secuencia.

### 5. ÁMBITO DE APLICACIÓN Y RANGO DE SOLUCIONES

Este proyecto de automatización de pedidos puede implementarse en numerosos sectores industriales, desde la industria automotriz hasta en la industria alimentaria. Por eso en el proyecto no se especifica qué tipo de productos contiene el paquete para así ser adaptable a su implementación en múltiples sectores.

En los primeros planteamientos del trabajo, la idea de los gemelos digitales resultó interesante al ser una tecnología en continuo avance y desarrollo. Se buscaba un proceso que se ajustara lo máximo a la realidad, lo que resultaba complicado por las limitaciones del software Factory IO. Primero, el proyecto iba encaminado hacia un único proceso de almacenamiento de paquetes, pero su programación resultaba muy básica. Se decidió ampliar esta idea, colocando previamente una etapa de llenado de paquetes con distintos productos, añadiendo para conseguirlo dos Pick and Place que se mueven en los ejes X y Z. Después, se incorporó una última fase de selección de paquetes para su envío al cliente.

### 6. AUTOMATIZACIÓN

La automatización industrial es el uso de la tecnología para la realización de procesos, minimizando así la intervención humana. Esta tecnología es capaz de modificar la forma de realizar los procesos y tareas, optimizándolos y alcanzando una mayor eficiencia. Hoy en día, la

automatización está presente en nuestra vida diaria de muchas maneras, facilitando nuestro día a día.

El tema del proyecto se centra en la automatización industrial, tecnología que avanza cada vez más gracias a tecnologías como la Inteligencia Artificial o la robótica. Las empresas buscan constantemente optimizar y reducir costes, objetivos que la automatización permite alcanzar, al aumentar la eficiencia de los procesos y reducir los errores que se cometen. Es posible ya optimizar procesos en tiempo real y predecir fallos antes de que ocurran. Supone también facilitar y reducir los riesgos de algunas de las tareas que realizan los empleados proporcionando una mayor seguridad en la planta acompañada de una disminución de los accidentes laborales.

También cabe destacar que muchas empresas estos últimos años han acelerado la adopción de la automatización debido a la pandemia del COVID-19. Ha supuesto una inversión en tecnologías automatizadas para así poder mantener los niveles de producción con menos personal presente físicamente y la reconfiguración de las líneas de producción para cumplir con nuevos estándares de salud y seguridad (Pictet Asset Management, 2020).

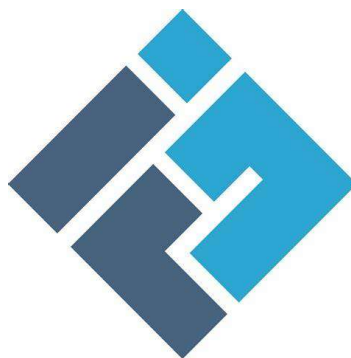
## 7. SOFTWARE DE MODELADO

El software utilizado para el modelado del proceso es Factory IO. Este software permite realizar simulaciones 3D de plantas industriales y su control mediante un PLC (real o simulado) en tiempo real. Ofrece a los ingenieros y técnicos no solo la posibilidad de diseñar nuevos proyectos sino de simular procesos industriales ya existentes con el objetivo de realizar cambios y corregir problemas antes de su implementación en el entorno real. Esto supone una reducción de costes, tiempos de inactividad en la producción y mayor seguridad para los operarios.

Existen varias opciones a la hora de crear un proyecto en Factory IO, ya que puedes empezar uno desde cero o hacer uso de una serie de escenas de las que dispone el programa. Dispone de ciertos elementos industriales desde cintas transportadoras y sensores hasta paletizadoras y elevadores. También dispone de una serie de materiales, cajas y pallets a utilizar en el proceso que se esté diseñando. Cuenta con una interfaz de usuario sencilla e intuitiva que facilita su uso y aprendizaje.

Factory IO también consta de ciertas limitaciones como la escasez de elementos de los que dispone, además de no poder modificar su dinámica de comportamiento ni tamaño. Por eso, es necesario a la hora de empezar a trabajar con el software tener en cuenta que condiciona mucho lo que puedes o no programar (Dominguez, A).

Se ha escogido este software para el modelado del proceso por ser sencillo de utilizar y visual para el usuario. Además, al estar realizando el proyecto a distancia y no disponer de las instalaciones de la UPV para trabajar, Factory IO era una opción gratuita y accesible con el equipo informático del que se disponía.



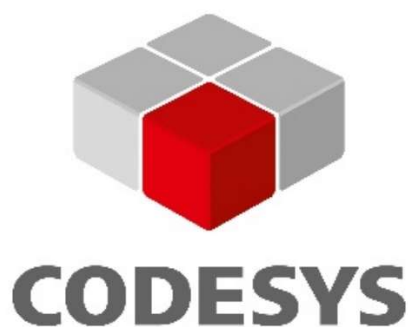
*Ilustración 2: Logo Factory IO de Real Games*

## 8. SOFTWARE DE PROGRAMACIÓN

El software en el que se ha realizado toda la programación del proyecto es CODESYS. Es un software que permite programar todo tipo de PLC's de forma eficiente y en distintos lenguajes de programación conforme con el estándar industrial internacional IEC 61131-3.

El software facilita el monitoreo de los procesos en tiempo real y proporciona herramientas de simulación y depuración que permiten probar y resolver problemas en los programas de control sin comprometer el hardware físico. Todo ello integrado en una única interfaz de usuario. Además, presenta una gran compatibilidad con buses de campo, como el Modbus TCP utilizado en este proyecto (Larraioz Group).

Se ha trabajado con este software al ser utilizado por cantidad de empresas en el mundo de la automatización industrial y no tener coste de instalación.



*Ilustración 3: Logo CODESYS de CODESYS Group*

## 9. LENGUAJES DE PROGRAMACIÓN

En CODESYS existe la posibilidad de utilizar un amplio abanico de lenguajes de programación lo que proporciona flexibilidad al programador y hace que su uso sea tan amplio y versátil.

El Diagrama de Flujo Secuencial o SFC, un lenguaje gráfico que representa el proceso secuencial mediante estados y transiciones. Permite una visualización clara de los estados y de la estructura del programa. Es uno de los que se ha utilizado en el proyecto debido a que es un lenguaje muy visual y se inspira en el método Grafcet que se utilizaba en las asignaturas de automatización de la universidad.

Después, el lenguaje de Bloques de Función o FBD es un lenguaje gráfico que permite controlar sistemas utilizando bloques de función. Cada bloque representa una función o operación lógica y se pueden conectar para formar una lógica de control. Es un lenguaje intuitivo ya que las operaciones y acciones se representan con símbolos gráficos e ideal para realizar programas sencillos.

El Texto estructurado o ST es un lenguaje textual que se asemeja al lenguaje de programación en C y permite el uso de estructuras de control como bucles, condiciones y variables. Es útil para la realización de algoritmos complejos.

El Diagrama de Contactos o LD es un lenguaje gráfico que se basa en la representación de circuitos eléctricos, utilizando contactos y bobinas para formar la lógica de control. Los contactos representan las entradas y las bobinas las salidas del sistema de control. Es un lenguaje de programación muy habitual en automatización de PLCs y comprensible para personas con conocimientos eléctricos. Es el otro lenguaje que se ha utilizado en el proyecto, para poder realizar la implementación de los elementos de parada y parada de emergencia de la planta industrial.

El Diagrama de Funciones Continuas o CFC es un lenguaje gráfico similar al FBD pero permite organizar los bloques de función de forma más libre, sin una red predefinida.

La Lista de Instrucciones o IL es un lenguaje textual formado por una serie de instrucciones que se ejecutan de manera consecutiva. Es el lenguaje apropiado si se buscan funciones que se ejecuten lo más rápido posible. Sin embargo, no es muy habitual en la automatización.

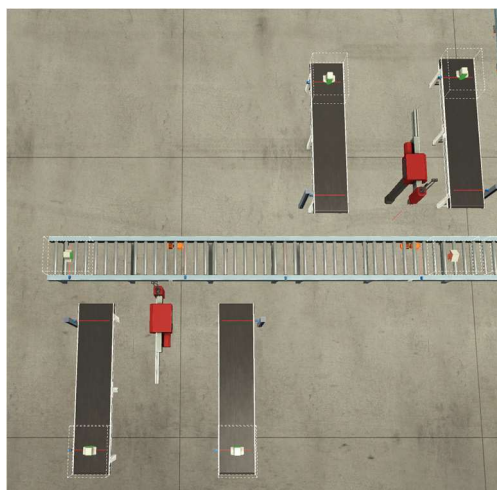
En definitiva, CODESYS al proporcionar tantas opciones de lenguajes de programación, permite escoger al usuario el que más se adapte a sus necesidades, además de ofrecer la oportunidad de combinar varios de estos lenguajes (SATOSHI, 2017).

## 10. DESCRIPCIÓN DEL PROCESO

El proceso productivo consta de dos partes, la primera se realiza la preparación de los pedidos y su clasificación en una estantería, y la segunda, la solicitud de esos pedidos preparados para las 3 líneas de producción en las que se requieran. El proceso hace uso de una serie de cintas, sensores, brazos mecánicos, un elevador y una estantería.

A continuación, se describen las cuatro etapas del proceso con mayor detalle:

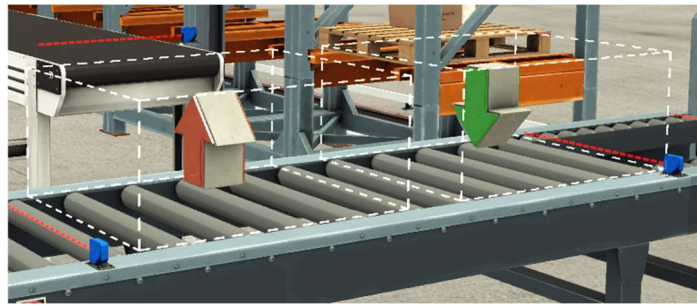
- 1. Preparación de pedidos:** la cinta principal transporta el conjunto pallet y caja abierta, donde dentro irán colocados los productos deseados. Después, se dispone de 4 tipos de productos que se desplazan por cintas transportadoras distintas hasta llegar a unos sensores donde los productos esperaran a ser recogidos. La recogida de estos es llevada a cabo por unos brazos mecánicos que se mueven en los ejes XZ y que cuando detecten que la caja está en la posición idónea, se encargará de coger los productos de sus respectivas cintas e introducirlos en la caja.



*Ilustración 4: Preparación de pedidos*



- 2. Cierre de los paquetes:** una vez el paquete contiene los productos deseados se cierra para su posterior almacenamiento. Esta parte del proceso se simula de manera irreal y se explicará detalladamente en el apartado Limitaciones del proyecto.



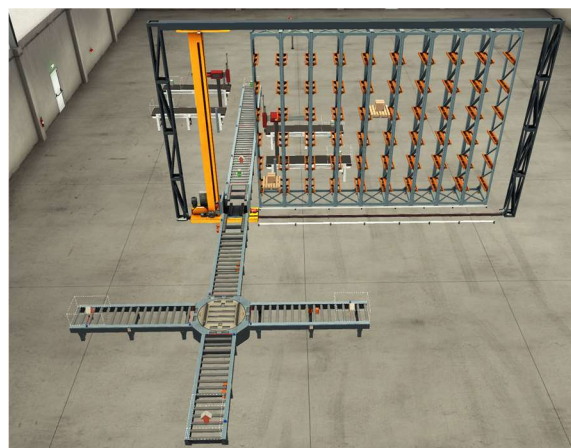
*Ilustración 5: Cierre de los paquetes*

- 3. Almacenamiento:** Una vez el pedido está preparado, se dirige por la cinta principal hacia la estantería. El pedido llega a unos sensores que detectan su posición para que así el elevador, que se desplaza vertical y horizontalmente, lo recoja y lo lleve a la posición donde se almacenará.



*Ilustración 6: Almacenamiento*

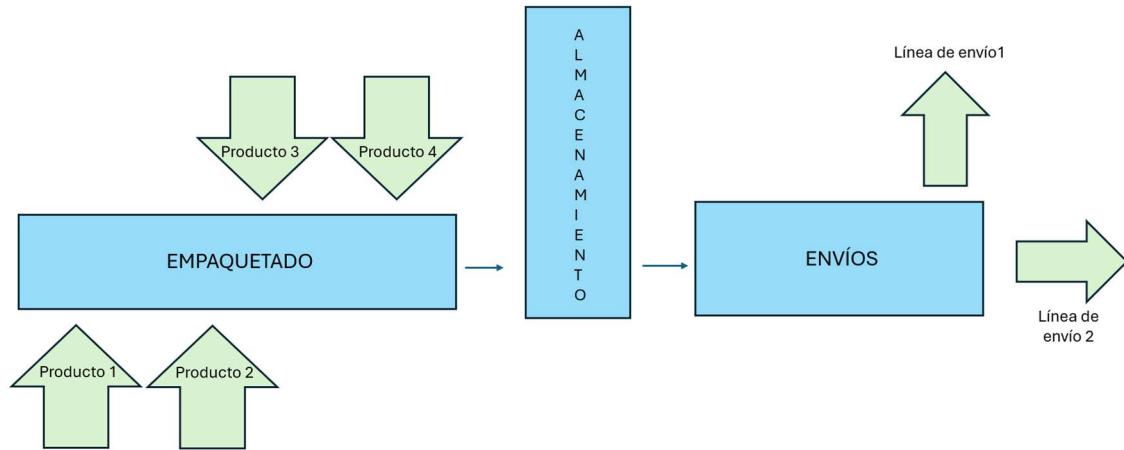
- 4. Solicitud de pedidos para su envío:** en esta parte, los operarios son los que introducirán el código de los pedidos que se van a enviar. Tras esto, serán recogidos de la estantería por el elevador y después, a través de unas cintas, se dirigirán a la línea de envío correspondiente.



*Ilustración 7: Envío de los pedidos*

## 11. ESQUEMA DE FUNCIONAMIENTO DE LA PLANTA

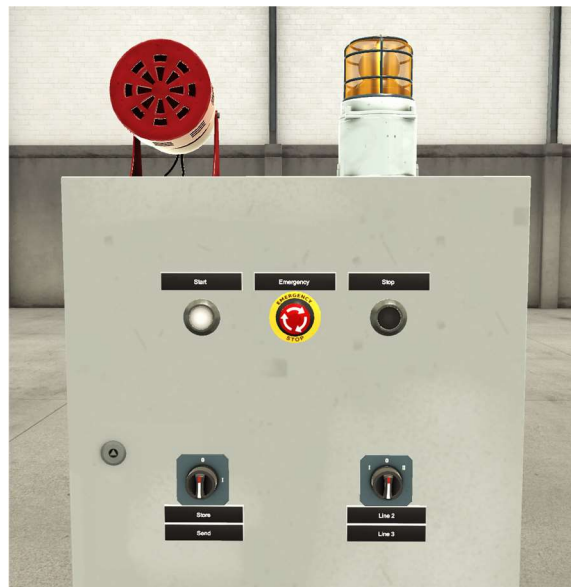
A continuación, se muestra la distribución de la planta industrial a automatizar de forma esquemática con el flujo de productos y salida de los pedidos.



*Ilustración 8: Esquema distribución de la planta*

## 12. OPERACIÓN Y CONTROL DE LA INSTALACIÓN

Se ha diseñado un panel de control gracias al cual los operarios pueden controlar el proceso de la planta. A continuación, se muestra como es el panel:



*Ilustración 9: Panel de control de la planta industrial*

El panel consta de tres botones: Start, Emergency y Stop. El botón Start se utiliza para iniciar la planta. El botón Emergency será pulsado por el operario en caso de emergencia y mientras se mantenga activado, se parará la planta y sonará una sirena de emergencia y una alarma sonora.

Para desactivarlo basta con volverlo a pulsar. En cambio, el botón stop simplemente parará la planta y para reanudar el proceso es necesario pulsar el botón start.

Por otro lado, en el panel se dispone de dos contactores que permiten escoger la operación a realizar y la línea de envío a la que se dirige paquete. Con el primer contactor el operario decide qué operación realizar, si quiere almacenar un paquete en la estantería (store) o cogerlo de la misma para realizar su envío (send). Mediante el otro contactor se decide a cuál de las 2 líneas de envío se dirigirá el paquete (line 1 or line 2).

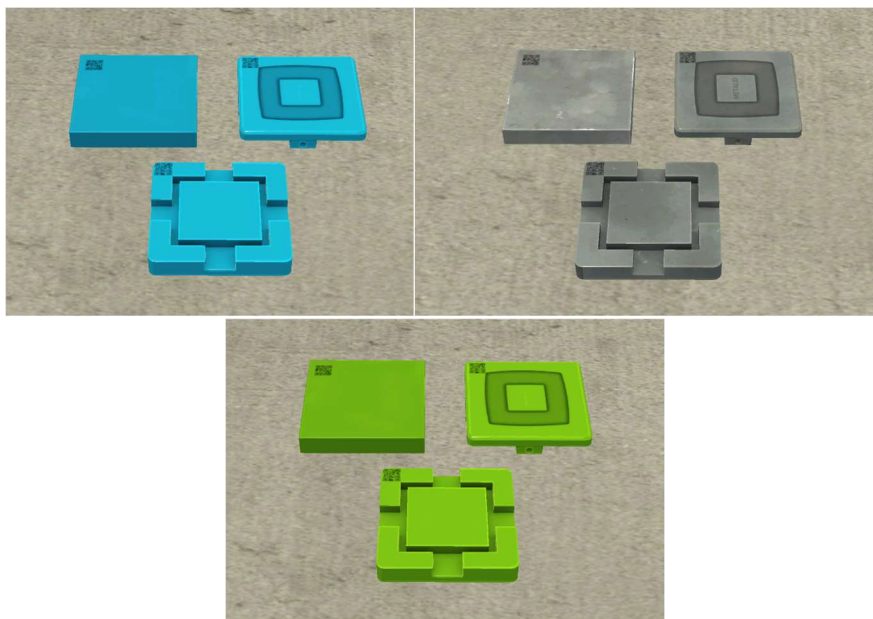
Cuando el operario decide cambiar los contactores de posición mientras la planta está en funcionamiento, el cambio comenzará tras la finalización del almacenamiento o del envío de los paquetes que estén circulando en la planta.

## 13. LIMITACIONES DEL PROYECTO

El software Factory IO es una herramienta actual y versátil para la simulación de procesos industriales, sin embargo, no consta de todo lo necesario para simular cualquier proceso industrial con exactitud. Para realizar este proyecto se han debido tener en cuenta las limitaciones que había por parte del software y se ha intentado simular aquello que más se asemejara a la realidad.

### 13.1. Limitaciones de los productos

La primera limitación ha sido la poca variedad de productos de la que se dispone. Hay materias primas, bases y tapas cuadradas de tres colores, pero no encontramos otro tipo de productos ni la posibilidad de cargarlos en el programa.



*Ilustración 10: Tipos de productos disponibles en Factory IO*

### 13.2. Limitación almacenamiento

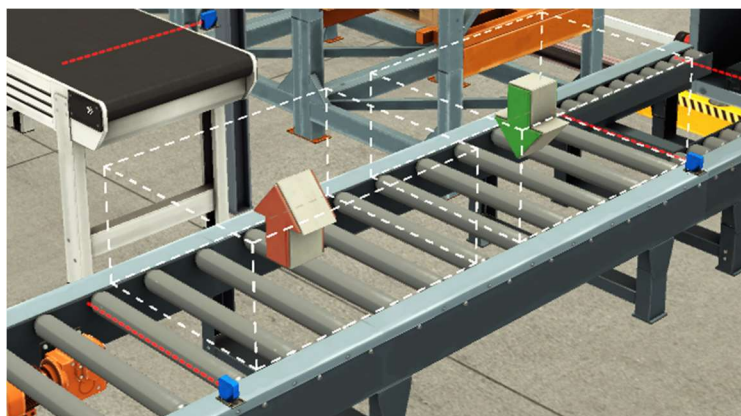
Otra limitación ha sido la parte de almacenamiento, ya que la estantería solo permite almacenar productos si debajo hay un pallet, por tanto, era necesario colocar un pallet debajo de la caja para poder colocarlos en la estantería.



*Ilustración 11: Limitación de la caja en la estantería*

### 13.3. Limitación cierre de paquetes

El proceso de cierre de los paquetes una vez metidos los productos es otra de las limitaciones que se ha encontrado. La primera caja que se mueve por la cinta principal tiene que estar abierta y vacía para poder mostrar cómo se introducen los productos. Sin embargo, a la hora de almacenar los paquetes, lo más ajustado a la realidad es almacenarlos en la estantería cerrados y listos para su posterior envío. Esto no se podía realizar en Factory IO, así que la medida que se ha tomado es mediante un elemento del que dispone el software llamado remove (flecha roja), eliminar la caja abierta en la que se han introducido los productos. Seguidamente usando otro elemento llamado emitter (flecha verde), hacer aparecer la caja cerrada, intentando simular el proceso de cierre de los paquetes.



*Ilustración 12: Limitación proceso de cierre de los paquetes*

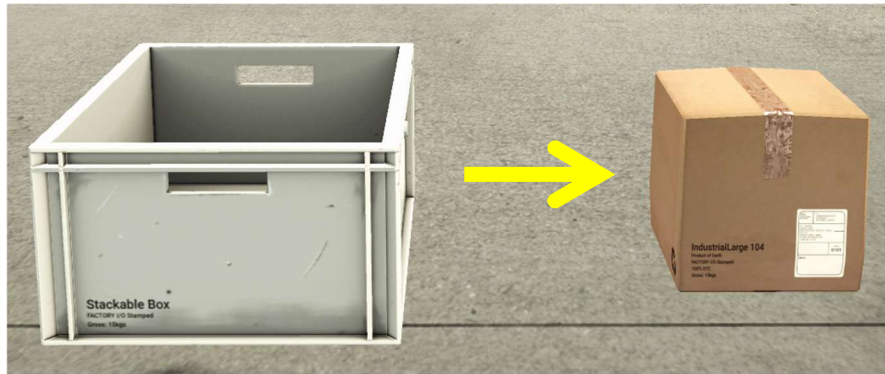


Ilustración 13: Limitación cajas

### 13.4. Limitaciones tr nsitos de entrada y salida

Otra limitaci3n que se ha encontrado trabajando con este software es no poder simular procesos manuales como los procesos de entrada de salida de los productos y pedidos. Estos procesos se realizan por los operarios de la planta y se han simulado de la siguiente forma:

- El primer proceso no autom tico que se debe llevar a cabo en esta planta es la colocaci3n de los 4 diferentes tipos de productos en las distintas citas transportadoras. Esto en la vida real ser  trabajo de los operarios de la planta o realizado por otro robot y en el software esto se ha simulado con los elementos “emitter” que hacen que aparezcan los productos en las cintas.
- El segundo proceso no autom tico que se ha realizado es el env o de los pedidos, que se ha resuelto dirigi ndolos mediante cintas hacia 2 l neas de env o donde despu s desaparecen simulando su recogida. Esto se ha simulado con los elementos “remover” que dispone el software y que hacen desaparecer el pedido.

## 14. DEFINICI3N DE LAS VARIABLES

### 14.1. Sensores

NOMBRE	FUNCI3N
Start	Bot3n de inicio
Stop	Bot3n de paro
Emergency	Bot3n de paro de emergencia
Store	Selector en modo almac�n
Send	Selector en modo env�o
Line1	Selector l�nea de env�o 1
Line2	Selector l�nea de env�o 2
Sensorinitial	Conjunto pallet caja al inicio del proceso
Sensoremmitter1	Detecta la emisi3n del producto 1
Sensoremmitter2	Detecta la emisi3n del producto 2
Sensoremmitter3	Detecta la emisi3n del producto 3
Sensoremmitter4	Detecta la emisi3n del producto 4
Sensor1	Producto 1 en posici3n para ser empaquetado
Sensor2	Producto 2 en posici3n para ser empaquetado
Sensor3	Producto 3 en posici3n para ser empaquetado
Sensor4	Producto 4 en posici3n para ser empaquetado
SensorPP	Paquete en posici3n para meter los productos
Sensorchangeroller	Detecta el paso a la siguiente cinta de rodillos

Sensorcierrrecaja	Detecta el comienzo del proceso de cierre de la caja
Sensorroller2	Detecta el paso a la siguiente cinta de rodillos
Sensorroller3	Paquete en posición para ser almacenado por el elevador
PP0_movingX	Pick and Place 0 se mueve en el eje X
PP0_movingZ	Pick and Place 0 se mueve en el eje Z
PP0_rotating	Pick and Place 0 rotando
PP0_detected	Pick and Place 0 detecta producto
PP1_movingX	Pick and Place 1 se mueve en el eje X
PP1_movingZ	Pick and Place 1 se mueve en el eje Z
PP1_rotating	Pick and Place 0 rota
PP1_detected	Pick and Place 1 detecta producto
Elev_atmiddle	Forks del elevador en posición central (inicial)
Elev_atright	Forks del elevador completamente hacia la derecha
Elev_atleft	Forks del elevador completamente hacia la izquierda
Elev_movingX	
Elev_movingZ	
Sensorexit	Paquete en posición de salida hacia las líneas de producción
Sensorarrivalturtable	Paquete colocado justo antes de la mesa giratoria
Turntable_frontlim	Límite frontal que alcanza el paquete en la mesa giratoria
Turntable_backlim	Límite trasero que alcanza el paquete en la Mesa giratoria
Turntable_lim0	Mesa giratoria en posición inicial
Turntable_lim90	Mesa giratoria girada 90°
Sensorfinalconveyor1	Entrada del paquete en la cinta que se dirige a la línea de envío 1
Sensorfinalconveyor2	Entrada del paquete en la cinta que se dirige a la línea de envío 2
Sensorremove1	Detecta los paquetes que se dirigen a la línea de envío 1
Sensorremove2	Detecta los paquetes que se dirigen a la línea de envío 2
Message1	Variable de tipo string para mostrar texto en la visualización
Message2	Variable de tipo entero para introducir el número del producto que se desea empaquetar
Message3	Variable de tipo string para mostrar texto en la visualización
Message4	Variable para tipo string para introducir el código del paquete que se desea enviar
Botonfin	Indica cuando finaliza el proceso de introducir los productos en el paquete.

## 14.2. Actuadores

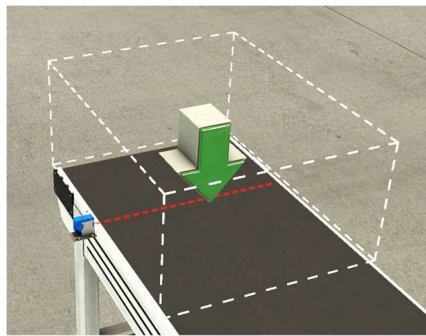
<b>NOMBRE</b>	<b>FUNCIÓN</b>
Siren	Sirena de emergencia
Lightalarm	Luz de emergencia
Emitterinitial	Emite conjunto pallet y caja
Emitter1	Emite producto 1
Emitter2	Emite producto 2
Emitter3	Emite producto 3
Emitter4	Emite producto 4
Conveyor1	Cinta que transporta los productos 1
Conveyor2	Cinta que transporta los productos 2
Conveyor3	Cinta que transporta los productos 3
Conveyor4	Cinta que transporta los productos 4
Conveyorroller1	Cinta de rodillos que transporta el paquete
Conveyorroller2	Cinta de rodillos que transporta el paquete
Conveyorroller3	Cinta de rodillos que transporta el paquete
PP0_X	Desplazamiento del Pick and Place 0 en el eje X
PP0_Z	Desplazamiento del Pick and Place 0 en el eje Z
PP0_grab	Pick and Place 0 recoge el producto
PP0_left	Rotación del Pick and Place 0 hacia la izquierda
PP0_right	Rotación del Pick and Place 0 hacia la derecha
PP1_X	Desplazamiento del Pick and Place 1 en el eje X
PP1_Z	Desplazamiento del Pick and Place 1 en el eje Z
PP1_grab	Pick and Place 1 recoge el producto
PP1_left	Rotación del Pick and Place 1 hacia la izquierda
PP1_right	Rotación del Pick and Place 1 hacia la derecha
Removeropen	Elimina la caja abierta
Emitterclosed	Crea una caja cerrada
Forksleft	Forks se desplazan a la izquierda
Forksright	Forks se desplazan a la derecha
Lift	Elevador asciende
Pos 1	Posición 1 del elevador
Pos 2	Posición 2 del elevador
Pos 3	Posición 3 del elevador
Pos 4	Posición 4 del elevador
Pos 5	Posición 5 del elevador
Pos 6	Posición 6 del elevador
Conveyorroller4	Cinta de rodillos por la que se mueve el paquete tras su recogida de la estantería
Conveyorroller5	Cinta de rodillos por la que se mueve el paquete tras su recogida de la estantería
Turntable_turn	Mesa giratoria gira
Turntable_rollpos	Rodillos de la mesa giratoria ruedan en sentido positivo
Turntable_rollneg	Rodillos de la mesa giratoria ruedan en sentido negativo

Conveyorfinal1	Cinta que dirige el paquete a la línea de envío 1
Conveyorfinal2	Cinta que dirige el paquete a la línea de envío 2
Remover1	Simula la continuación del paquete hacia la línea de envío 1
Remover2	Simula la continuación del paquete hacia la línea de envío 2

## 15. DISEÑO DE LA PLANTA

### 15.1. Elementos línea de almacenaje

- **Entrada productos:**



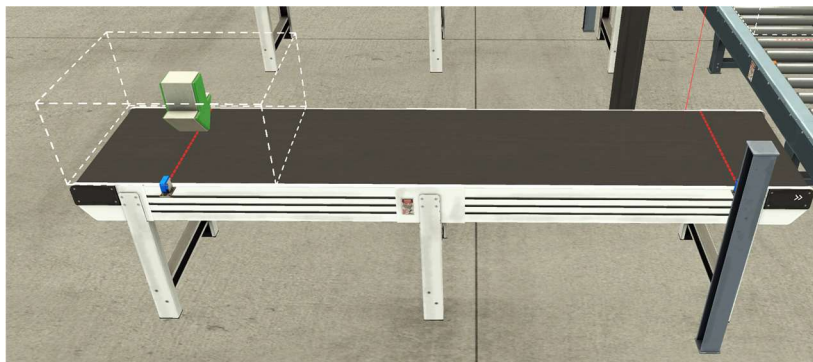
*Ilustración 14: Emitter y sensor entrada producto*

Función: Emite un tipo de producto de los 4 disponibles en la planta (creación de los productos con el actuador emitter)

Actuadores digitales: “Emitter1”, “Emiter2”, “Emitter3” y “Emitter4”.

Sensores digitales: “Sensoremitter1”, “Sensoremitter2”, “Sensoremitter3” y “Sensoremitter4”

- **Cintas transportadoras productos:**



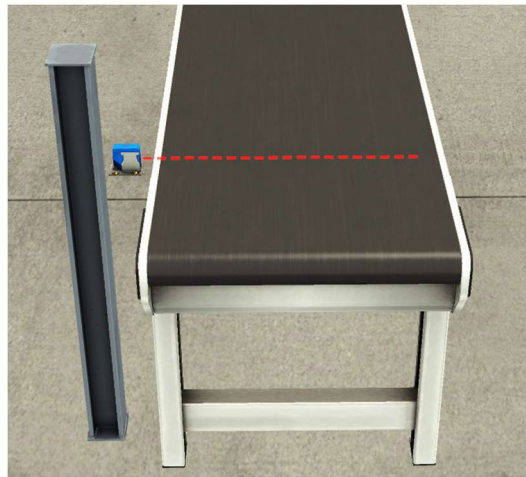
*Ilustración 15: Cinta productos*

Función: Transportar los diferentes tipos de productos hasta que sean recogidos por el Pick and Place XZ.

Actuadores digitales: “Conveyor1”, “Conveyor2”, “Conveyor3” y “Conveyor4”.



- **Sensores productos:**

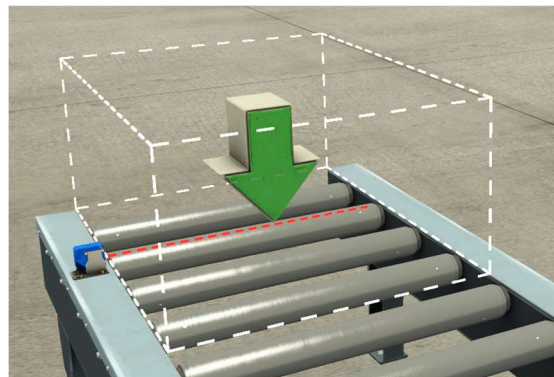


*Ilustración 16: Sensor fin cinta producto*

Función: Detectar la llegada de los productos al final de la cinta que los transporta.

Sensores digitales: “Sensor1”, “Sensor2”, “Sensor3” y “Sensor4”.

- **Entrada paquete:**



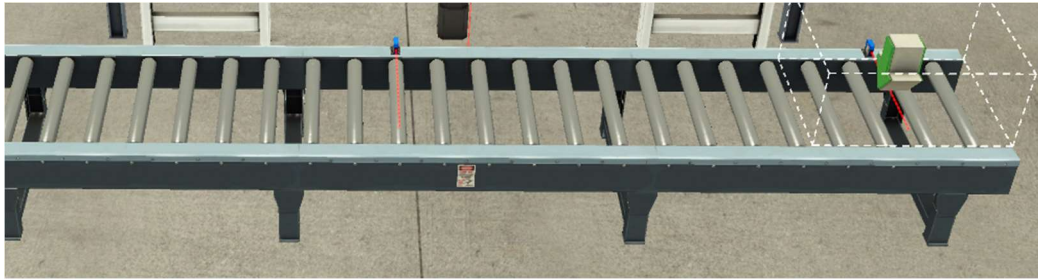
*Ilustración 17: Emitter y sensor entrada pallet-caja*

Función: Simula la entrada del conjunto pallet-caja en el que se introducirán los 4 tipos de productos (creación del paquete con el actuador emitter)

Actuador digital: “Emitterinitial”

Sensor digital: “Sensorinitial” que detecta la emisión de un pallet-caja.

- **Cintas transportadoras paquete:**

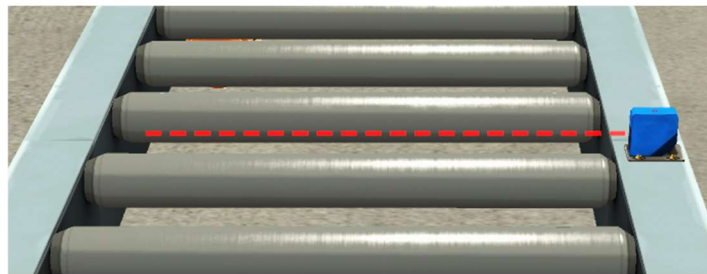


*Ilustración 18: Cinta transportadora pallet-caja*

Función: Cinta de rodillos por la que se mueve el conjunto pallet-caja.

Actuadores digitales: “Conveyorroller1”, “Conveyorroller2” y “Conveyorroller3”.

- **Sensores:**



*Ilustración 19: Sensores cintas rodillos*

Función: Detectar el paso del conjunto pallet-caja por las cintas de rodillos.

Sensores digitales: “SensorPP”, “Sensorchangeroller”, “Sensorcierrecaja”, “Sensor roller 2” y “Sensor roller 3”.

- **Pick and Place XZ:**



*Ilustración 20: Pick and Place XZ*

Función: Recoger los diferentes productos de sus respectivas cintas y colocarlos dentro de la caja que circula sobre el pallet por las cintas de rodillos.

Actuadores digitales: “PP0\_X”, “PP0\_Z”, “PP0\_left”, “PP0\_right”, “PP0\_grab”, “PP1\_X”, “PP1\_Z”, “PP1\_left”, “PP1\_right” y “PP1\_grab”

Sensores digitales: “PP0\_movingX”, “PP0\_movingZ”, “PP0\_rotating”, “PP0\_detected”, “PP1\_movingX”, “PP1\_movingZ”, “PP1\_rotating” y “PP1\_detected”.

- **Elevador:**



*Ilustración 21: Elevador de paquetes*

Función: Almacenar el conjunto pallet-caja en la estantería.

Actuadores digitales: “Forksleft”, “Forksright”, “Lift”, “Pos1”, “Pos2”, “Pos3”, “Pos4”, “Pos5” y “Pos6”.

Sensores digitales: “Elev\_atleft”, “Elev\_atright”, “Elev\_atmiddle”, “Elev\_movingX” y “Elev\_movingZ”.

## 15.2. Descripción línea de almacenaje

El proceso de preparación de los pedidos comienza con la entrada del conjunto pallet-caja mediante el actuador “Emitterinitial”. Cuando el “Sensorinitial” detecta un paquete sobre la cinta de rodillos, se enciende el actuador de dicha cinta que es “Conveyorroller1” hasta que el paquete es detectado por “Sensor PP” que indica que la caja está en la posición correcta para ser rellena con los productos.

De forma simultánea a lo descrito anteriormente, los 4 productos de los que disponemos han aparecido mediante el actuador “EmitterX” y al detectar su aparición con el “Sensoremitter1” se activan los actuadores del tipo “ConveyorX” que encienden las cintas que transportan los productos hasta que llegan a “SensorX” que detecta que están en la posición correcta para ser empaquetados (siendo X cualquiera un número del 1 al 4 haciendo referencia a los 4 productos disponibles para empaquetar).

Estando en posición la caja para la introducción de los productos entran en acción los Pick and Place. El pick and place 0 se encarga de recoger los productos 1 y 2 y el Pick and Place 1 se encarga de recoger los productos 3 y 4.

La posición inicial del Pick and Place 0 es con el actuador “PP0\_X” activado, que se mantiene así durante todo el proceso. Estando ya el pick and place sobre el paquete, el operario pulsa el botón del producto 1 o 2. Se desplazará hacia la izquierda con “PP0\_left” o hacia la derecha con “PP0\_right” dependiendo del producto, este movimiento será detectado por el sensor “PP0\_rotating”. Una vez ha rotado del todo, se enciende el actuador “PP0\_Z” que baja el eje Z. Cuando el eje Z está completamente bajo y toca el producto, es cuando se activa el sensor “PP0\_detected” y se puede activar el “PP0\_grab” para coger el producto. Una vez se coge el producto (este actuador se mantiene hasta que el Pick and Place se encuentre sobre la caja), el actuador del eje Z se desactiva y el Pick and Place vuelve a rotar a la posición inicial a la posición inicial (encima del pallet-caja).

Sobre el Pick and Place 1, su posición inicial es solo con el actuador “PP1\_X” activado, que se mantiene así durante todo el proceso. Estando el pick and place sobre el paquete, el operario pulsa el botón del producto 3 o 4. El pick and place se desplazará hacia la izquierda con “PP1\_left” o hacia la derecha con “PP1\_right”, este movimiento será detectado por el sensor “PP1\_rotating”. Una vez ha rotado del todo, se enciende el actuador “PP1\_Z” que baja el eje Z. Cuando el eje Z está completamente bajo y toca el producto, es cuando se activa el sensor “PP1\_detected” y se puede activar el “PP1\_Grab” para coger el producto. Una vez se coge el producto (este actuador se mantiene hasta que el Pick and Place se encuentre sobre la caja), el actuador del eje Z se desactiva para dirigirse a la posición inicial (encima del pallet-caja).

Una vez ya se han introducido las cantidades de productos deseadas, se ha de pulsar el interruptor “FIN EMPAQUETADO” que se encuentra en la visualización del programa y que activará “Conveyorroller1” hasta pasado el “Sensorchangeroller”. Al pasar este sensor se activará la siguiente cinta mediante el actuador “Conveyorroller2”. Después, la caja pasa el “Sensorcierrecaja” que tras dejarlo atrás provoca la activación del actuador “Removertransform” que elimina la caja abierta, y junto a este, se encuentra el “Emitertransform” que crea una caja cerrada, simulando el proceso de cierre del paquete. El “Sensorroller2” indica que se activa también la “Conveyorroller3” hasta que el paquete llega al “Sensorroller3” donde espera a ser recogido por el ascensor.

Tras la activación del “sensorroller3” el ascensor se pone en marcha. Se encuentra en su posición inicial, y activa primero el actuador “Forksleft” hasta la activación del sensor “Elev\_atleft”, indicando que se encuentra totalmente desplazado hacia la izquierda. Después se enciende “Lift” que lo eleva hacia arriba cuyo movimiento es detectado por el sensor “Elev\_movingZ”. Ahora el paquete esta sobre el ascensor y tras desactivar el actuador de “Forksleft” que hace que el ascensor se encuentre en su posición central (detectado por “Elev\_atmiddle”), el ascensor se mueve a la posición correspondiente mediante los actuadores “Pos X” (siendo X un número del 1 al 6, que son los actuadores del ascensor). El ascensor se moverá a la posición más cercana que no esté ocupada por otro paquete. Por tanto, comenzará colocando el paquete en la primera fila y primera columna del ascensor, si esta resultase ocupada, pasaría a la segunda columna de la primera fila y así sucesivamente.



*Ilustración 22: Visualización donde introducir el producto deseado y botón fin empaquetado*

### 15.3. Elementos línea de recogida y envío

- **Cintas transportadoras:**

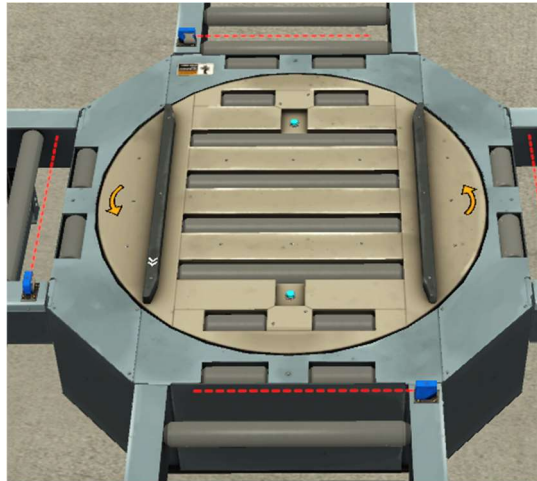


*Ilustración 23: Cinta transportadora*

Función: Cintas de rodillos por la que se mueve el conjunto pallet-caja tras ser recogido de la estantería por el elevador.

Actuadores digitales: “Conveyorroller4”, “Conveyorroller5”, “Conveyorfinal1” y “Conveyorfinal2”.

- **Mesa giratoria:**



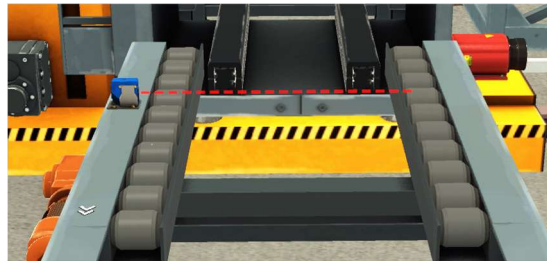
*Ilustración 24: Mesa giratoria*

Función: Mesa de rodillos que gira dependiendo de la línea de producción a la que se dirija el paquete.

Actuadores digitales: “Turntable\_rollpos”, “Turntable\_rollneg” y “Turntable\_turn”

Sensores digitales: “Turntable\_lim0”, “Turntable\_lim90”, “Turntable\_frontlim” y “Turntable\_backlim”

- **Sensores:**



*Ilustración 25: Sensores*

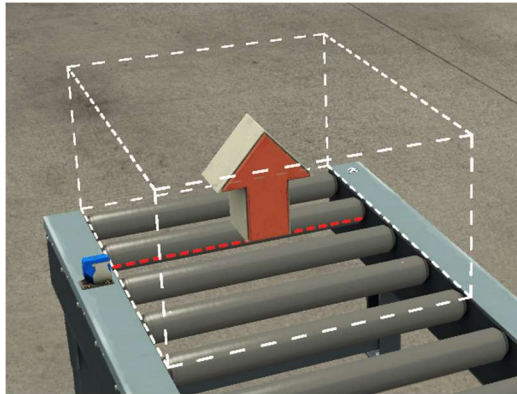


*Ilustración 26: Sensores*

Función: Detectar la llegada del paquete desde la estantería a las cintas de rodillos, detectar la llegada del paquete a la mesa giratoria y la salida del paquete de la mesa giratoria.

Sensores digitales: “Sensorexit”, “Sensorarrivalturntable”, “Sensorfinalconveyor1” y “Sensorfinalconveyor2”.

- **Salida paquete:**



*Ilustración 27: Remover y sensor salida paquete*

**Función:** Simula la salida del paquete a la línea de producción correspondiente (eliminación del paquete con el actuador remover).

**Actuadores digitales:** “Remover1” y “Remover2”.

**Sensores digitales:** “Sensorremover1” y “Sensorremover2”.

#### 15.4. Descripción línea de recogida

La línea de recogida de los pedidos comienza recogiendo el pedido de la estantería mediante el elevador. Se le pide al usuario que introduzca el código del producto que se quiere recoger de la estantería, habiendo solo dos opciones por simplificación del proyecto (el cuadro de escritura del código aparece en una visualización en el software CODESYS). Los códigos de los dos productos disponibles en la estantería son: “UNO” o “DOS”. Una vez el usuario escribe uno de estos dos códigos, el elevador se dirigirá a su posición. Después, tras haber recogido el pedido de la estantería de la posición indicada por el usuario, el elevador retorna a la posición inicial para mandar el paquete a las líneas de envío, activando el actuador “Lift” y después “Forksright” para elevar y desplazar el paquete a la cinta correspondiente. Esta cinta se activa con “Conveyorroller4” cuando “Sensorexit” detecta el paquete sobre la cinta. El paquete se desplaza por la cinta hasta que es detectado por “Sensorarriwalturtable” donde el paquete espera para moverse a la línea de envío a la que vaya destinado.

Una vez la tabla giratoria está en la posición correcta, que se sabe gracias a la activación del sensor “Turntable\_lim0”, se vuelve a activar la cinta y se activan también los rodillos de la tabla giratoria en sentido positivo mediante el actuador “Turntable\_rollpos” hasta que se activa el sensor “Turntable\_frontlim”. Llegado este punto, dependiendo de la línea a la que se dirija el paquete la tabla giratoria rotará mediante “Turntable\_turn” activándose para detectar que está completamente girada 90° el sensor “Turntable\_lim90” o se mantendrá en esa posición en el caso de dirigirse hacia la línea de envío 2. Una vez la tabla giratoria está orientada correctamente, se activa “Turntable\_rollpos” y al activarse “Sensorfinalconveyor”, las cintas finales se activan mediante “Conveyorfinal”.

Por último, existe un sensor llamado “Sensorremover” que detecta cuando puede activarse el actuador “Remover” que elimina el paquete simulando que se dirige a la línea de envío correspondiente.

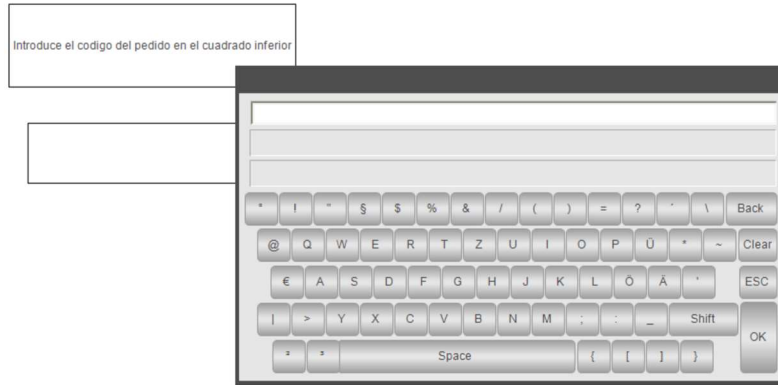


Ilustración 28: Visualización donde introducir el código del paquete a enviar

## 16. GRAFCETS Y SFC

En este apartado, se muestran los grafkets que se han realizado antes de realizar la programación en el software CODESYS. Han sido extremadamente útiles ya que se asemejan al lenguaje SFC que se ha empleado en el automatismo realizado. Se trata de un proceso secuencial y todos los grafkets comparten variables, por lo que están entre ellos relacionados.

Se muestran también los programas desarrollados en el software CODESYS, que están asociados a estos grafkets. Se han utilizado los lenguajes SFC, al ser más visual y estar relacionado con los grafkets, y el lenguaje Ladder, ya que era necesario para realizar el paro y el paro de emergencia de la planta.

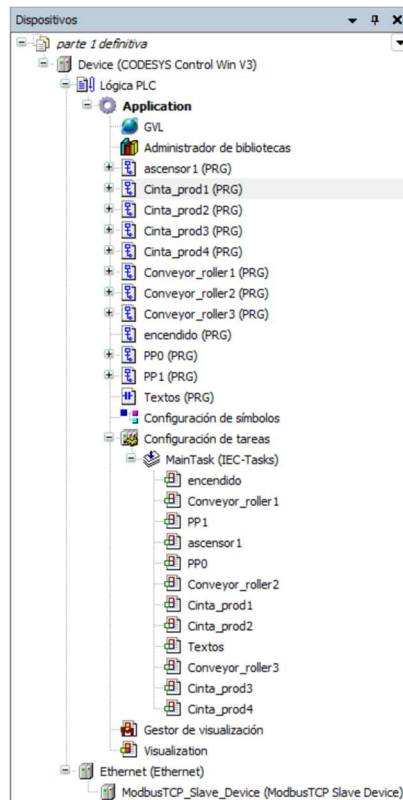


Ilustración 29: Distribución SFCs en el programa



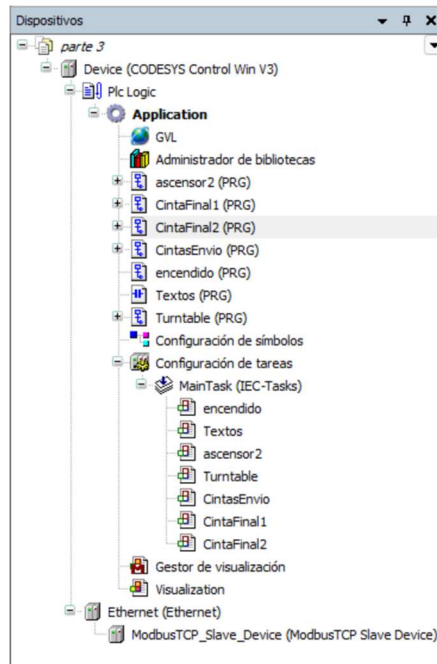


Ilustración 30: Distribución SFCs en el programa II

En las ilustraciones 29 y 30 se muestran los POU que se han creado para cada elemento de la planta, escritos todos en lenguaje SFC, menos los llamados “Textos” que están escritos en lenguaje Ladder. Dentro de cada POU se encuentran las acciones que realiza cada etapa, escritas en lenguaje Ladder por lo mencionado anteriormente de los paros de la planta.

Se da una breve explicación del primer grafcet que es el que determina el inicio, el paro y el paro de emergencia del proceso ya que no ha sido descrito como funciona con anterioridad el encapsulado que aparece. Una vez se inicia el proceso, empieza la etapa 1 (encapsulado), donde se encuentran todos los demás grafcets del proceso. Por tanto, una vez empieza esta etapa, comienzan todos los demás grafcets. Como se observa en la Ilustración 26, al pulsar el botón Stop o Emergency, se sale del encapsulado, parándose todos los grafcets que se estaban llevando a cabo.

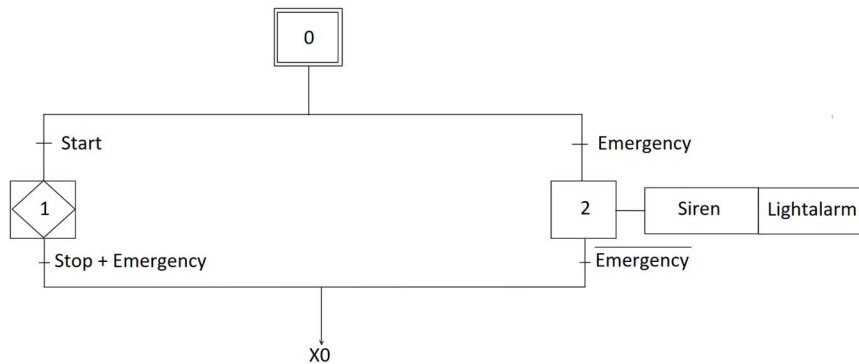


Ilustración 31: Grafcet arranque y paro

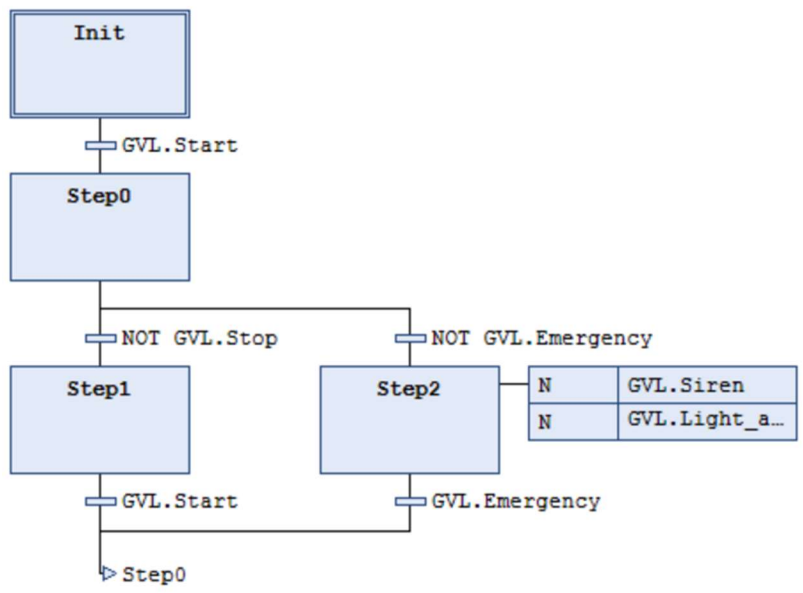


Ilustración 32: SFC arranque y paro

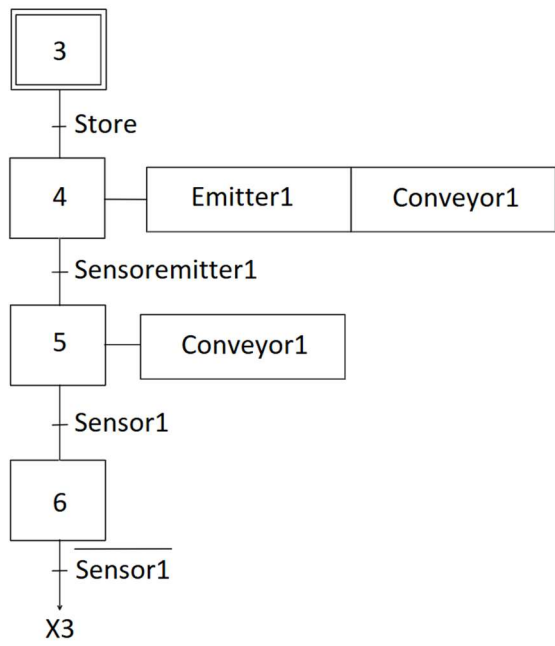


Ilustración 33: Graficet cinta producto 1

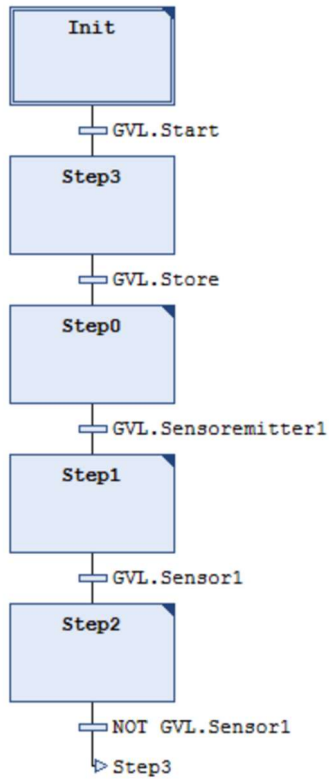


Ilustración 34: SFC cinta producto 1

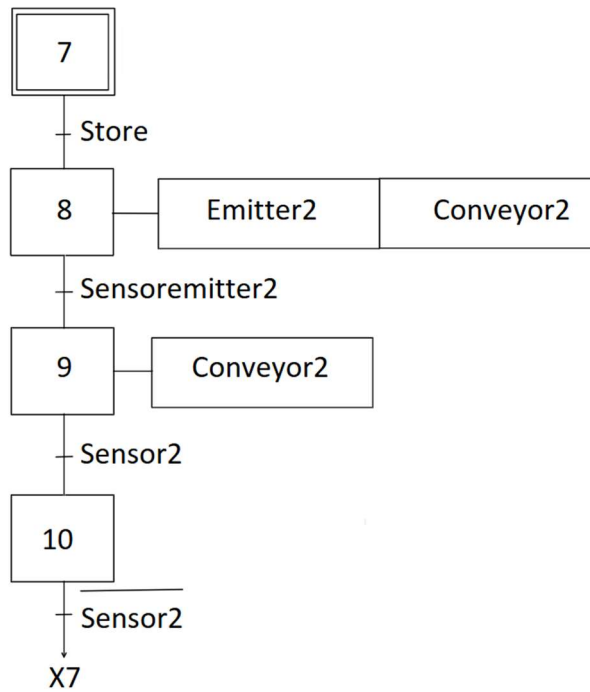


Ilustración 35: Grafset cinta producto 2

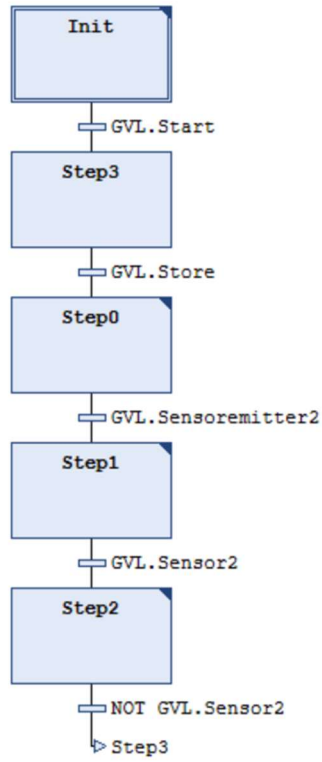


Ilustración 36: SFC cinta producto 2

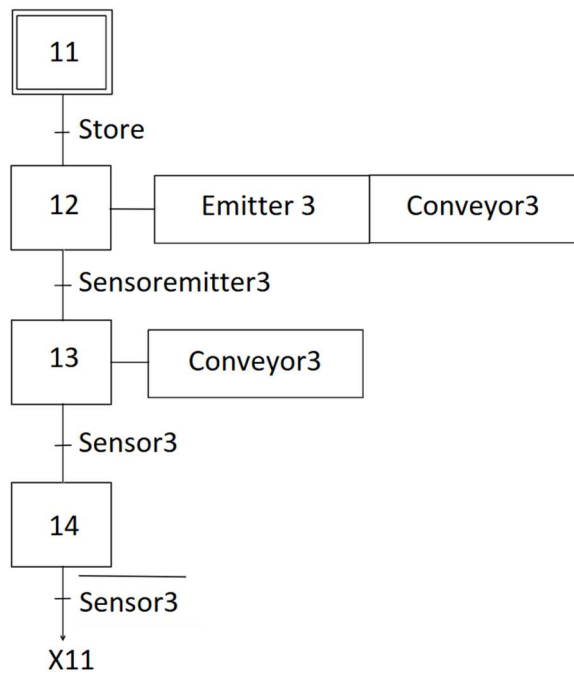


Ilustración 37: Grafset cinta producto 3

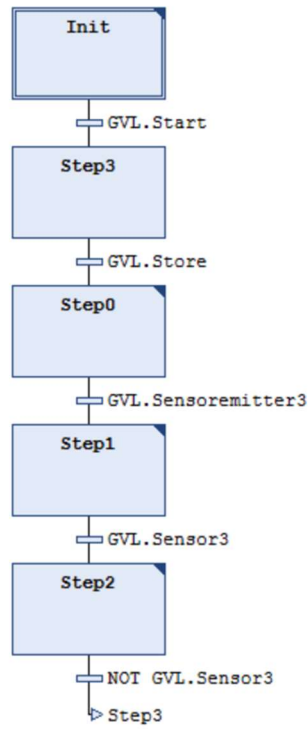


Ilustración 38: SFC cinta producto 3

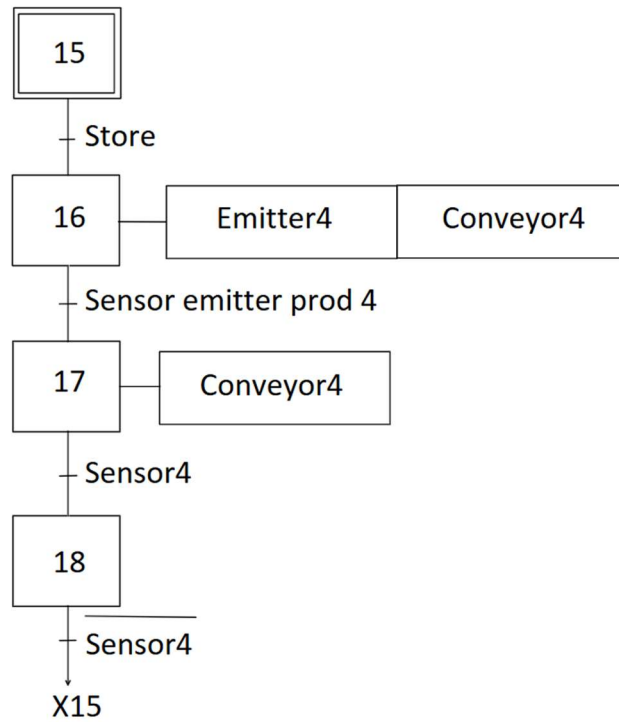


Ilustración 39: Grafset cinta producto 4

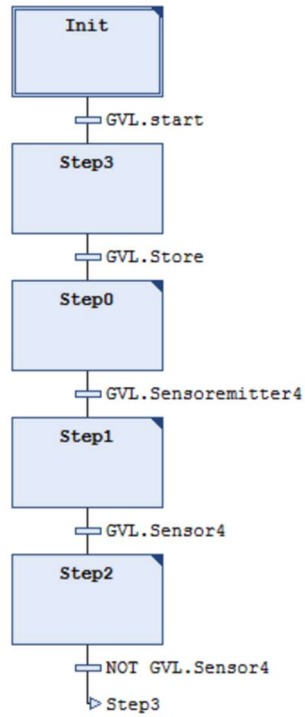


Ilustración 40: SFC cinta producto 4

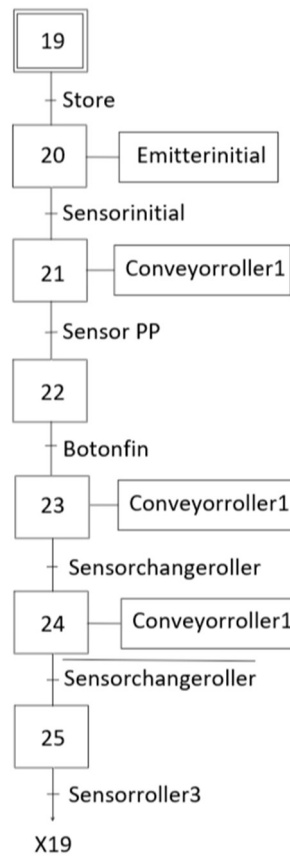


Ilustración 41: Grafcet cinta de rodillos 1

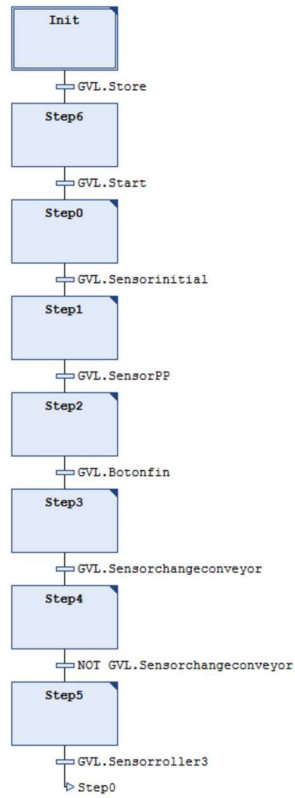


Ilustración 42: SFC cinta de rodillos 1

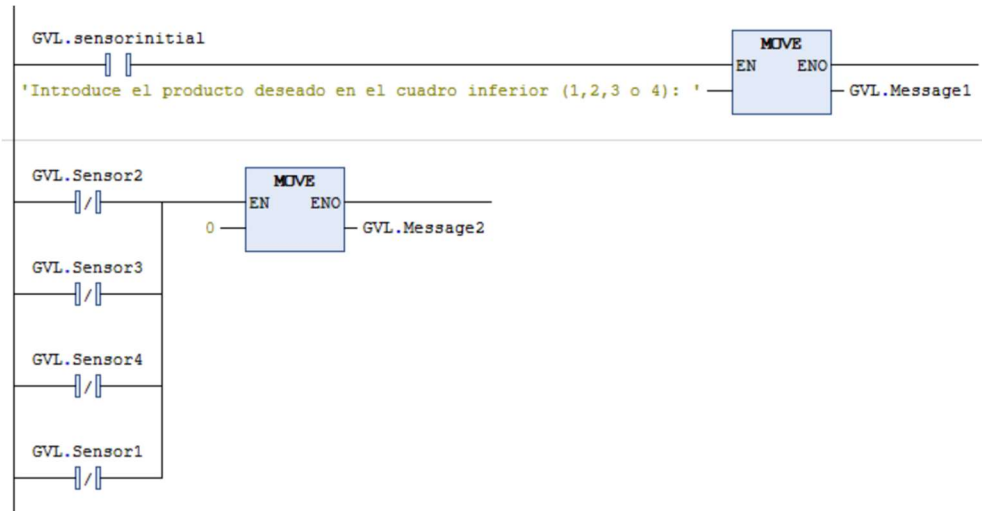


Ilustración 43: Ladder de textos

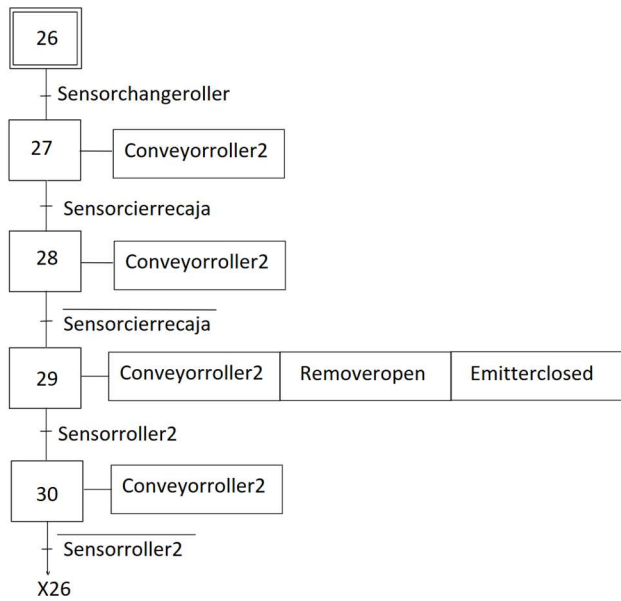


Ilustración 44: Grafcet cinta roller 2

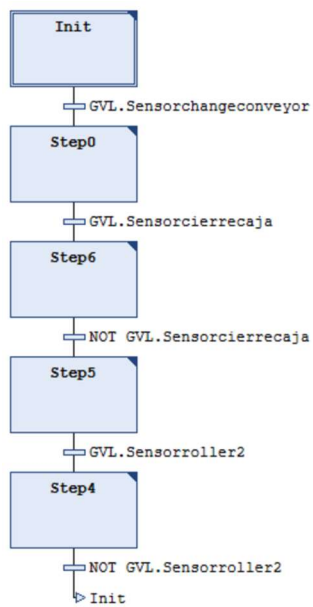


Ilustración 45: SFC cinta roller 2

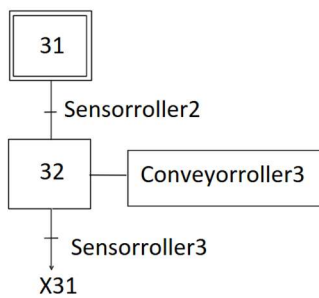


Ilustración 46: Grafcet cinta roller 3



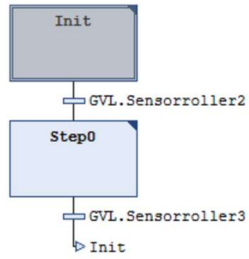


Ilustración 47: SFC cinta roller 3

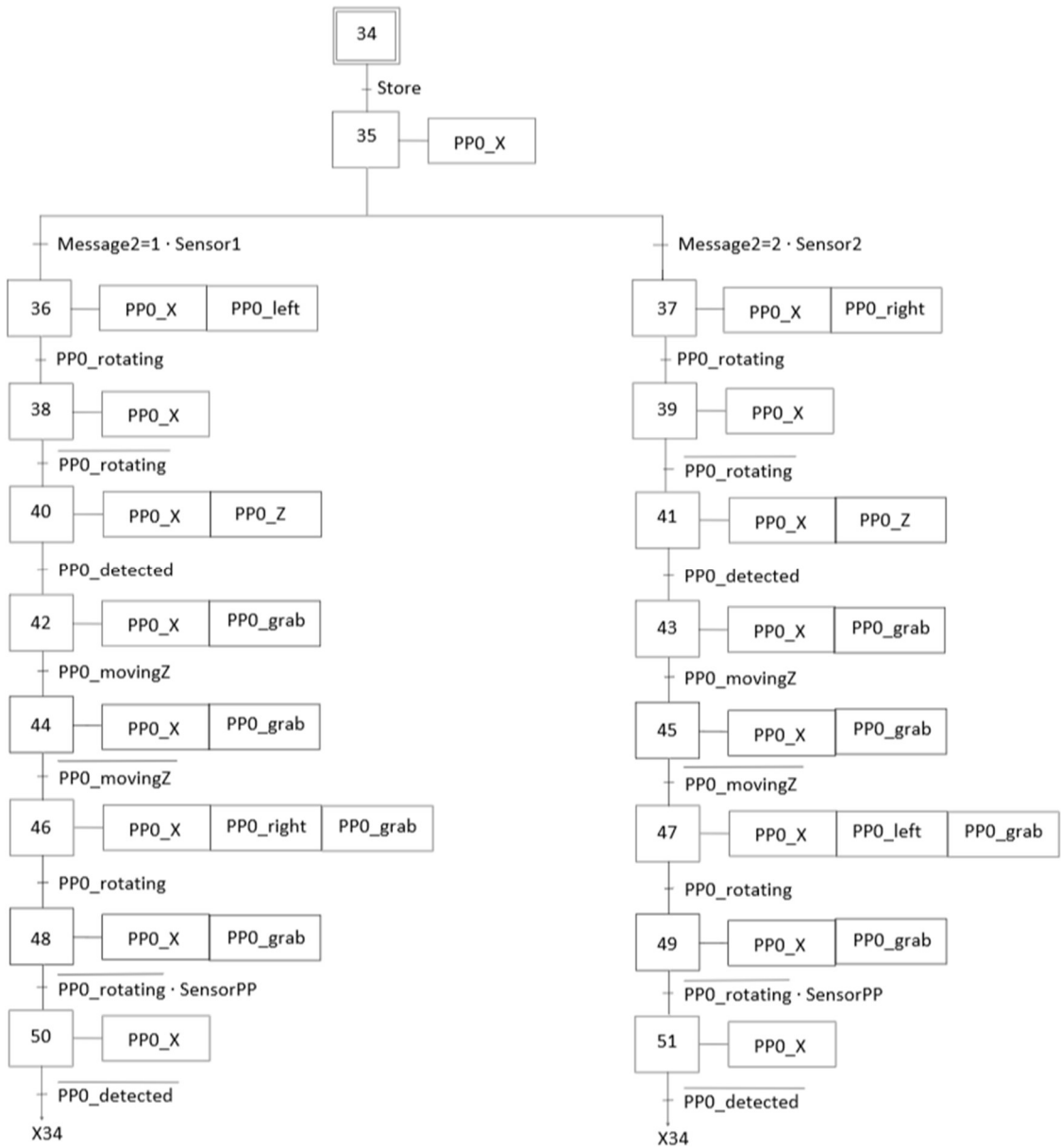


Ilustración 48: Gracet Pick and Place 0

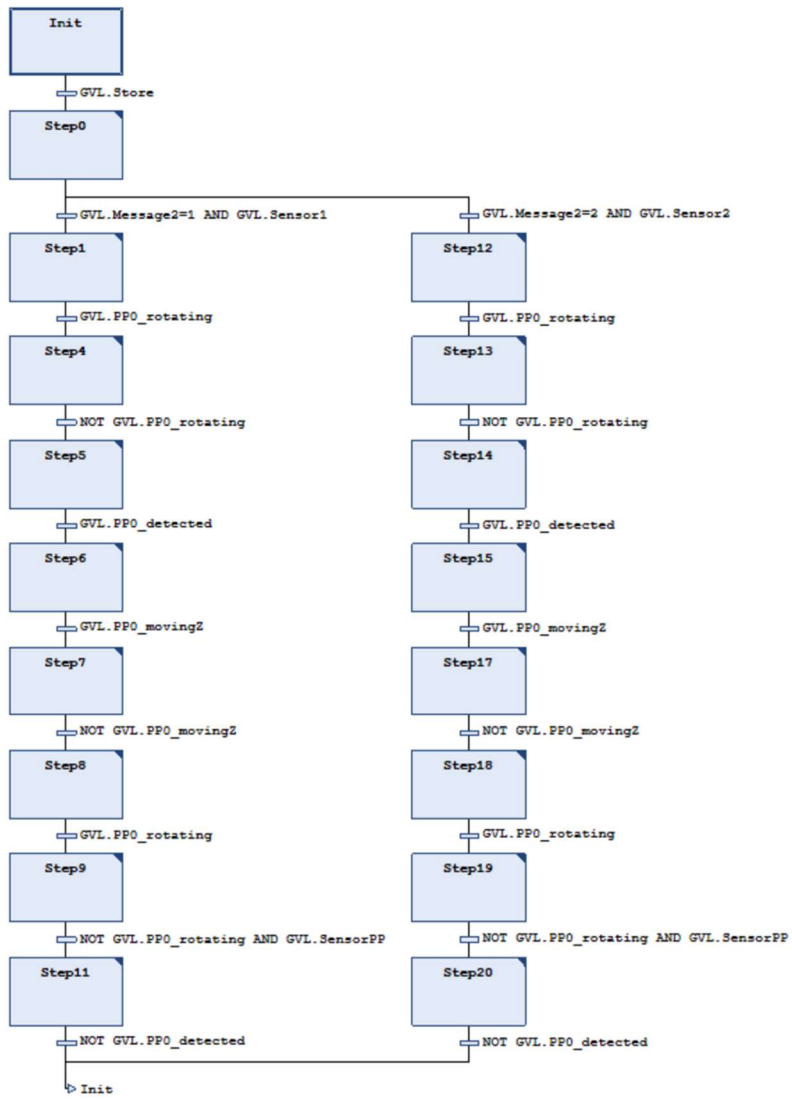


Ilustración 49: SFC Pick and Place 0

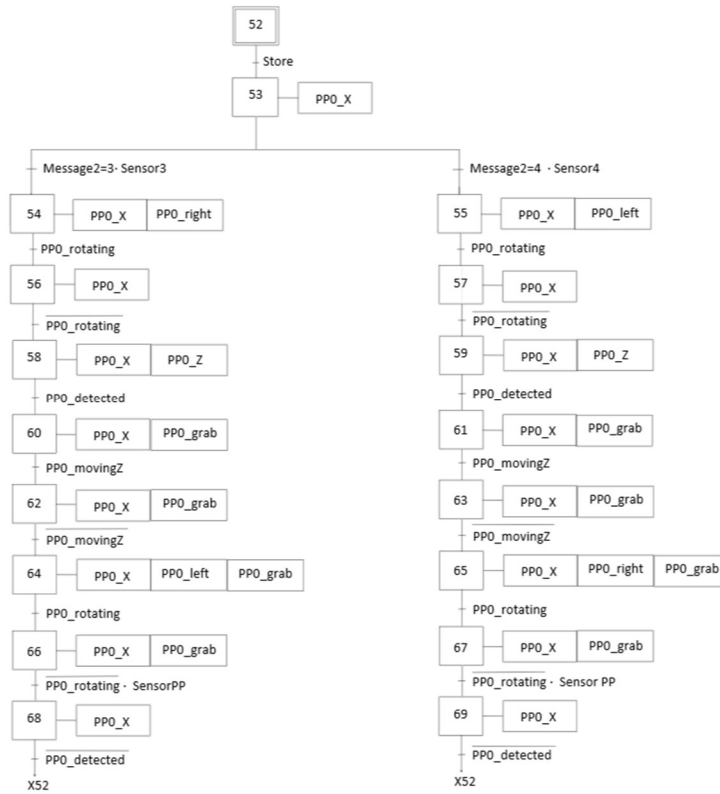


Ilustración 50: Grafset Pick and Place 1

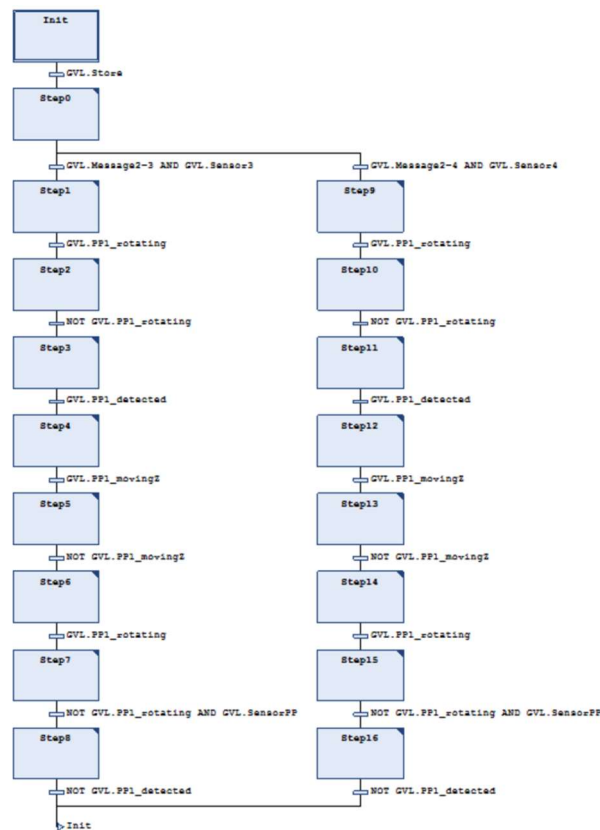


Ilustración 51: SFC Pick and Place 1

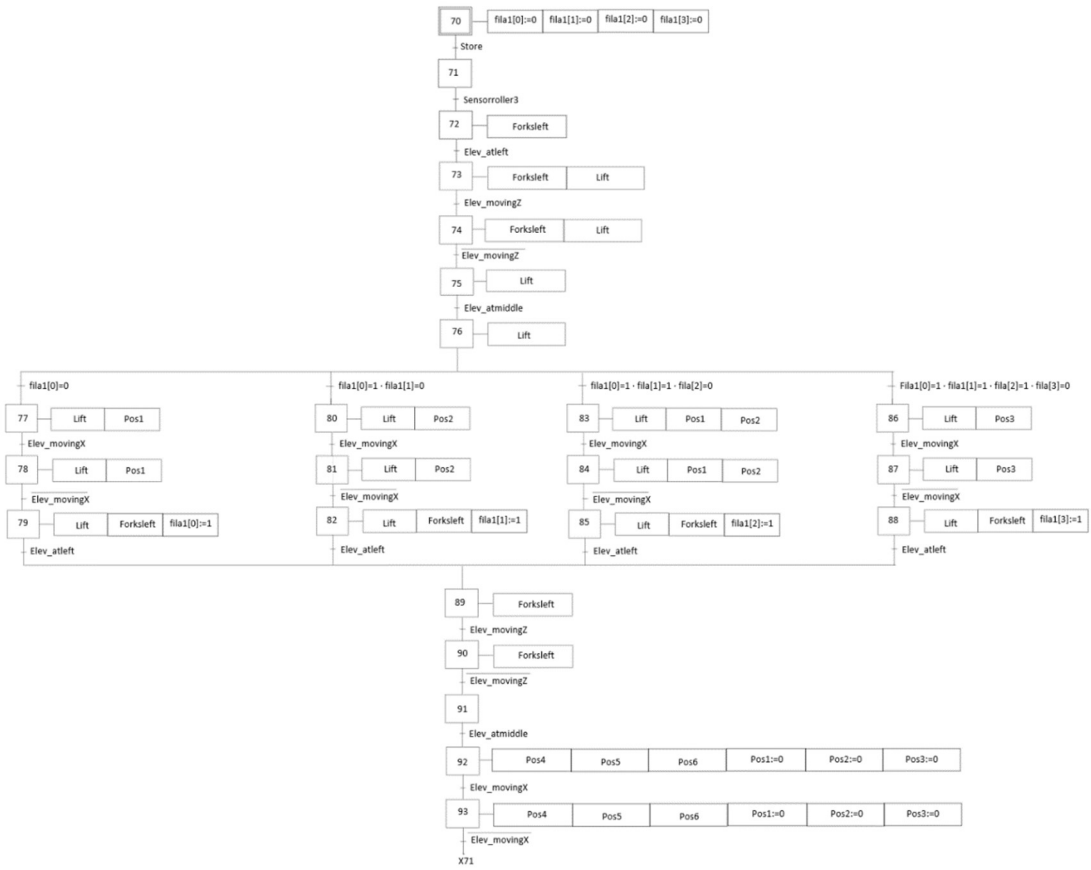


Ilustración 52: Grafset elevador modo store

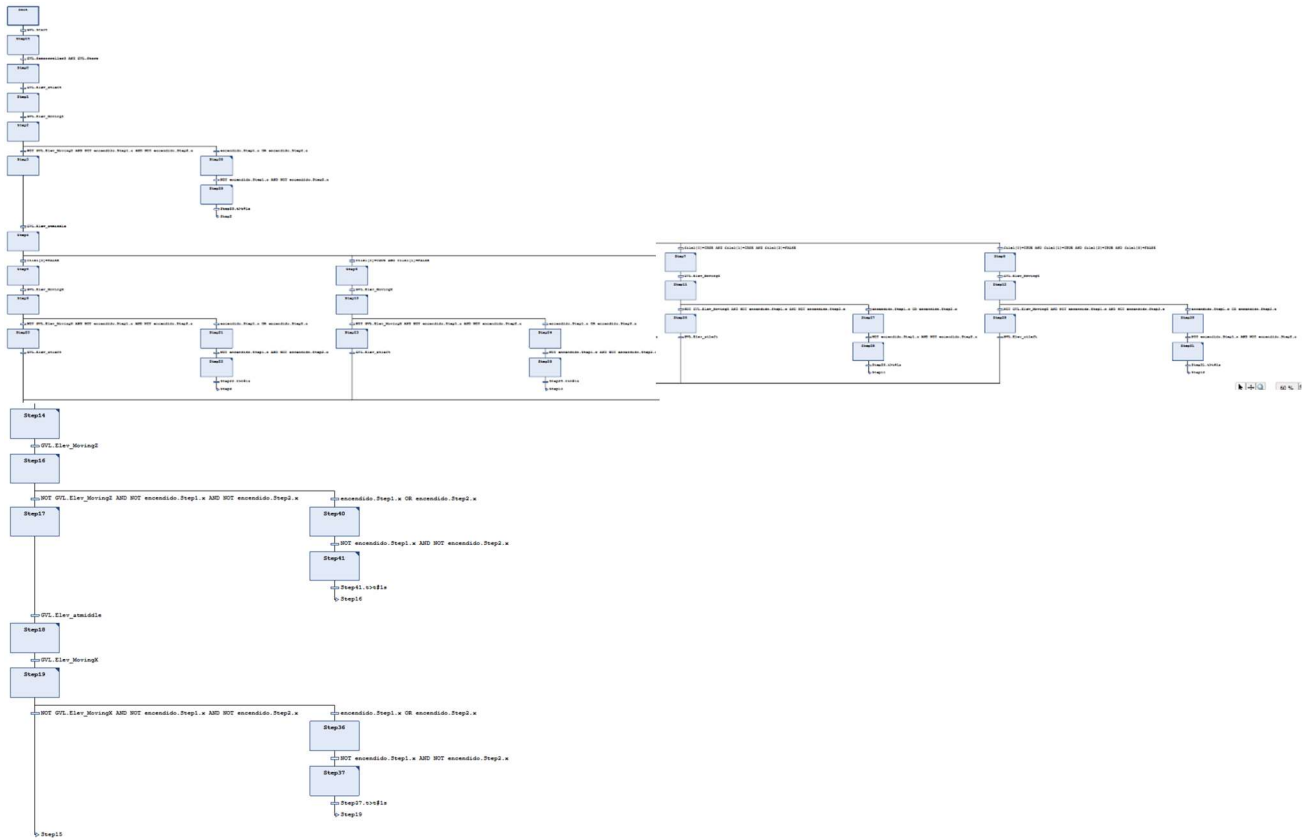


Ilustración 53: SFC elevador modo store

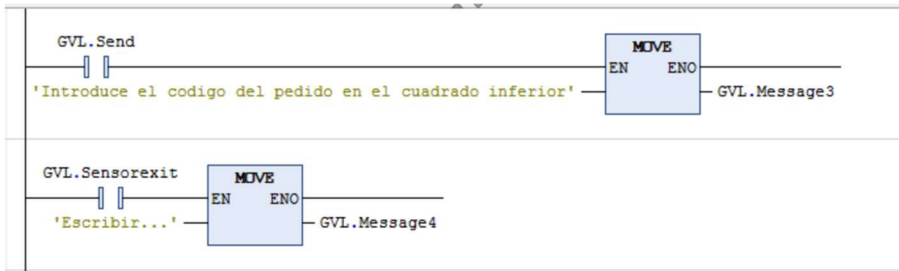


Ilustración 54: Ladder de textos

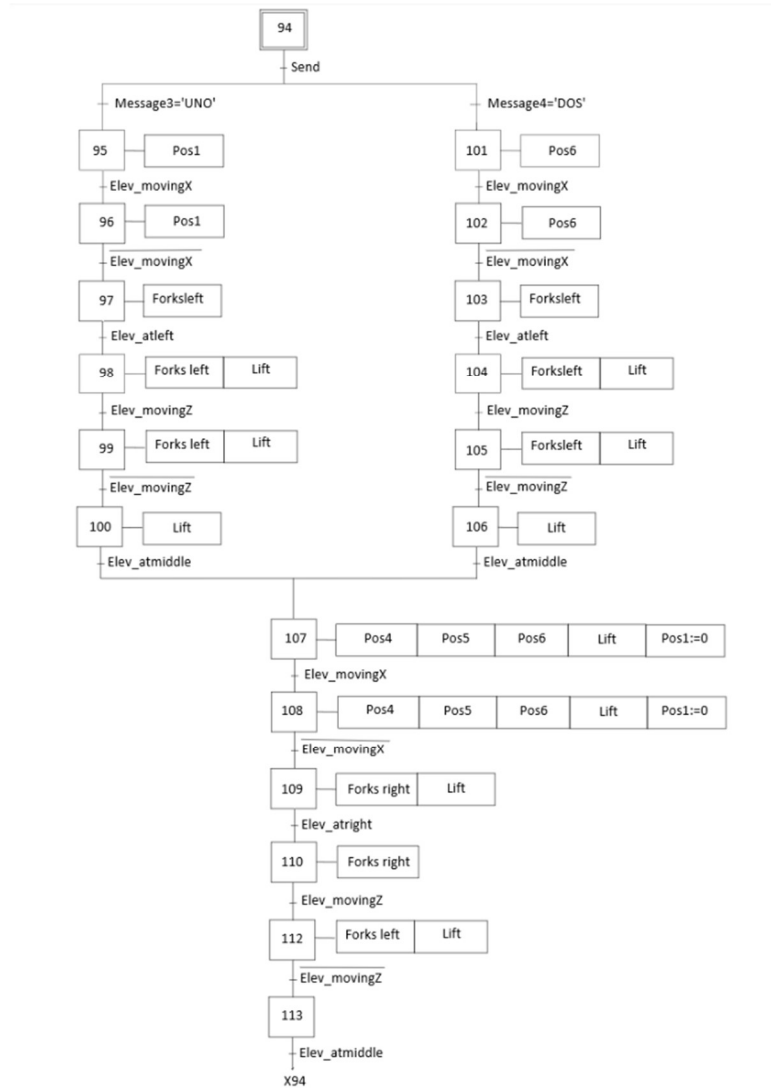


Ilustración 55: Grafset elevador modo send

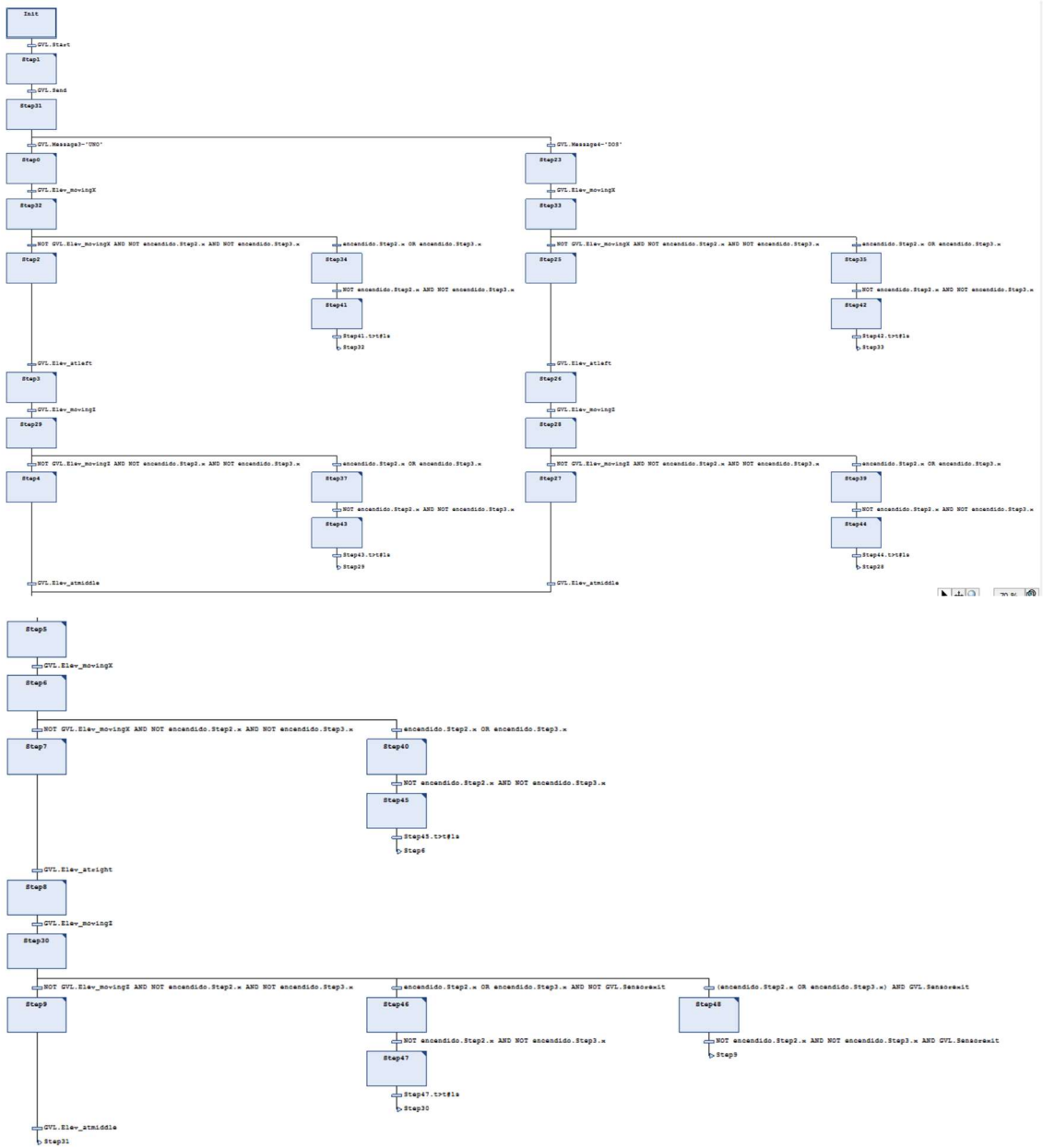


Ilustración 56: SFC elevador modo send

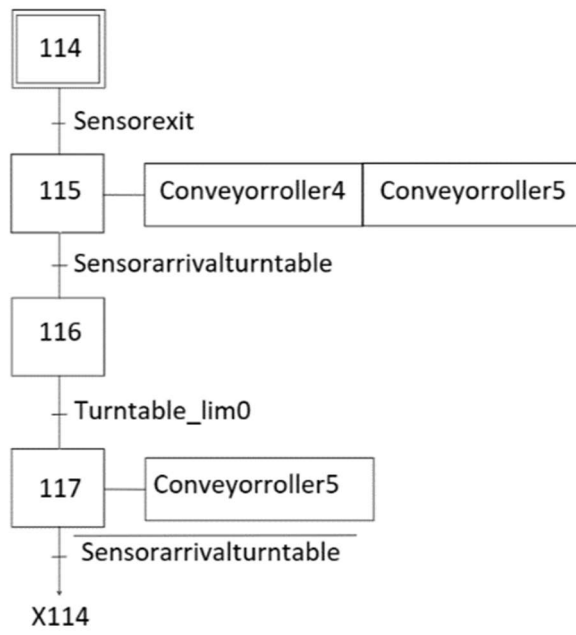


Ilustración 57: Grafcets cintas roller 4 y 5

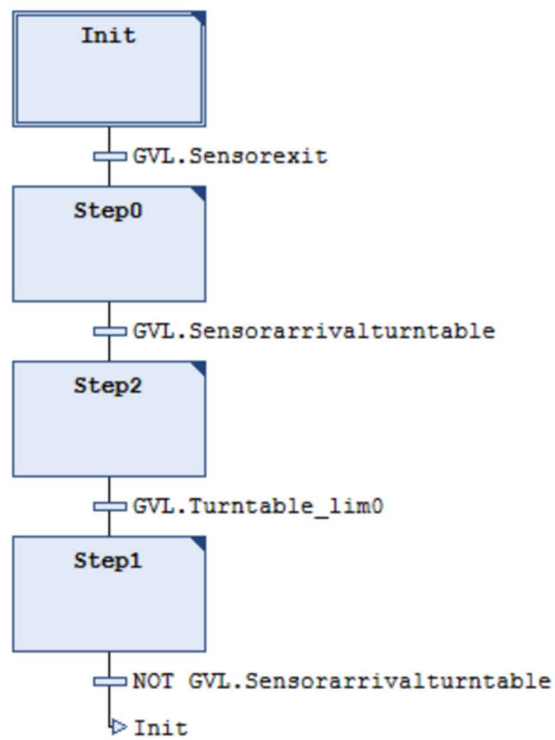


Ilustración 58: SFC cintas roller 4 y 5

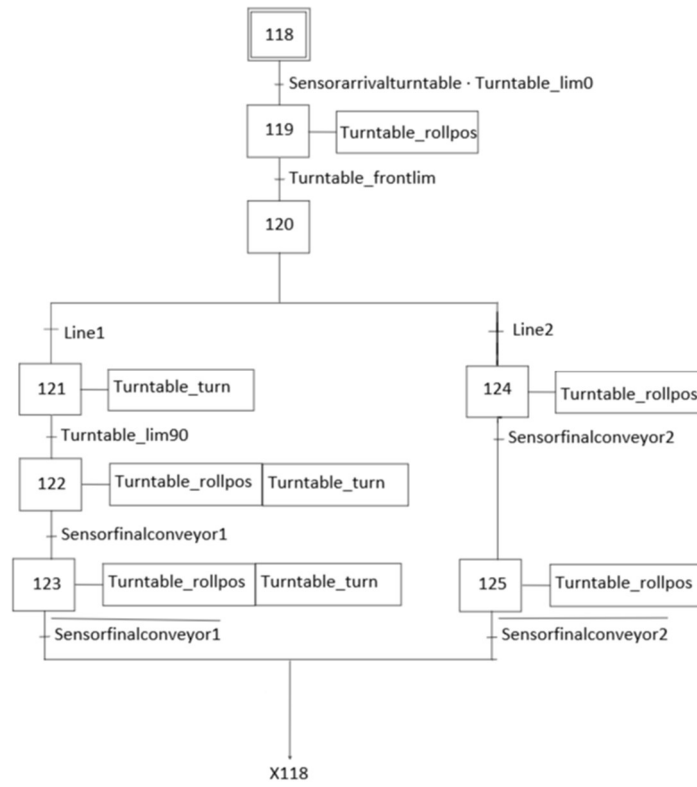


Ilustración 59: Grafcet turntable

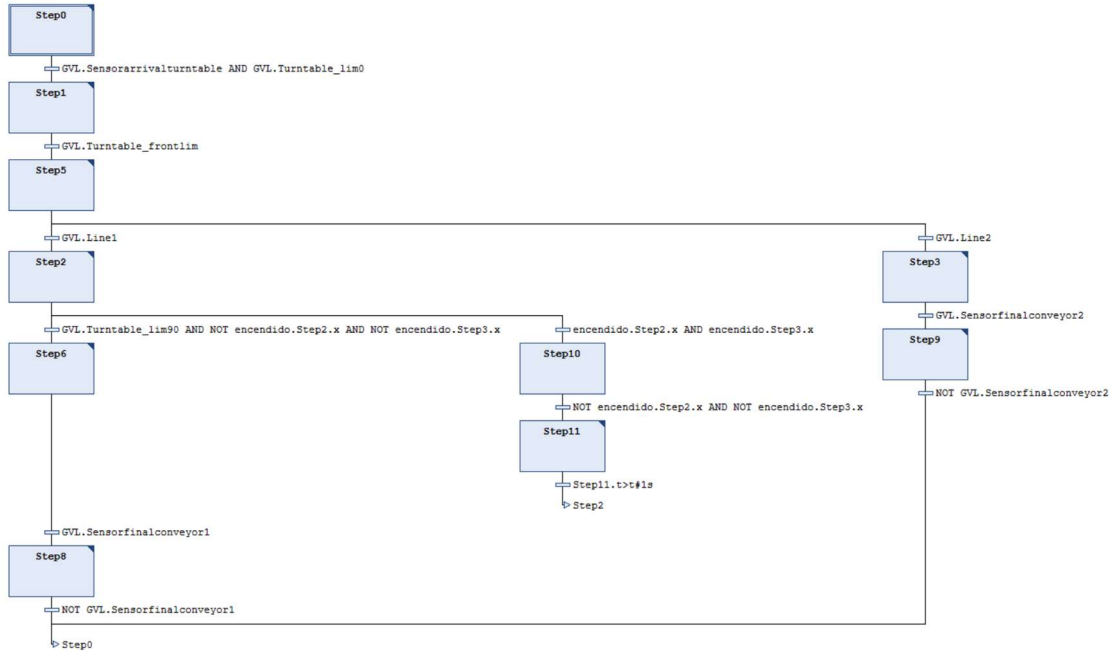


Ilustración 60: SFC turntable



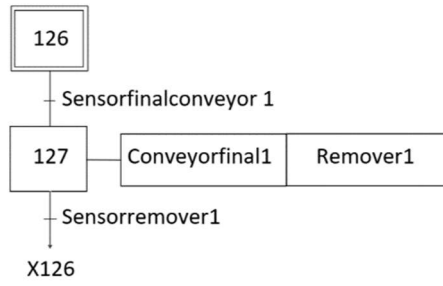


Ilustración 61: Grafcet cinta final 1

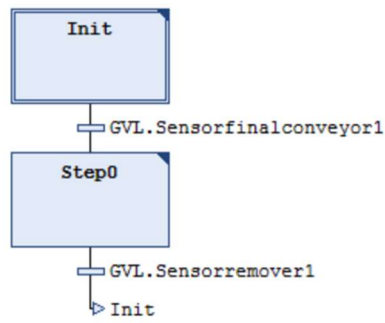


Ilustración 62: SFC cinta final 1

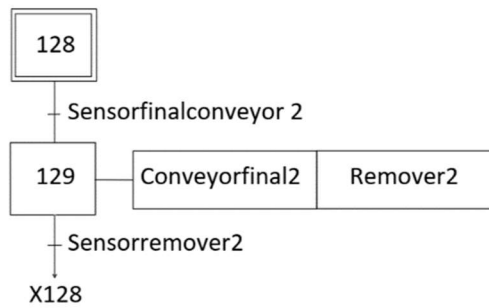


Ilustración 63: Grafcet cinta final 2

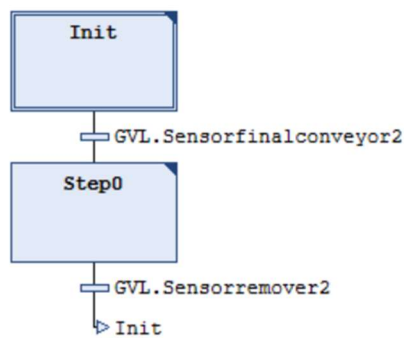
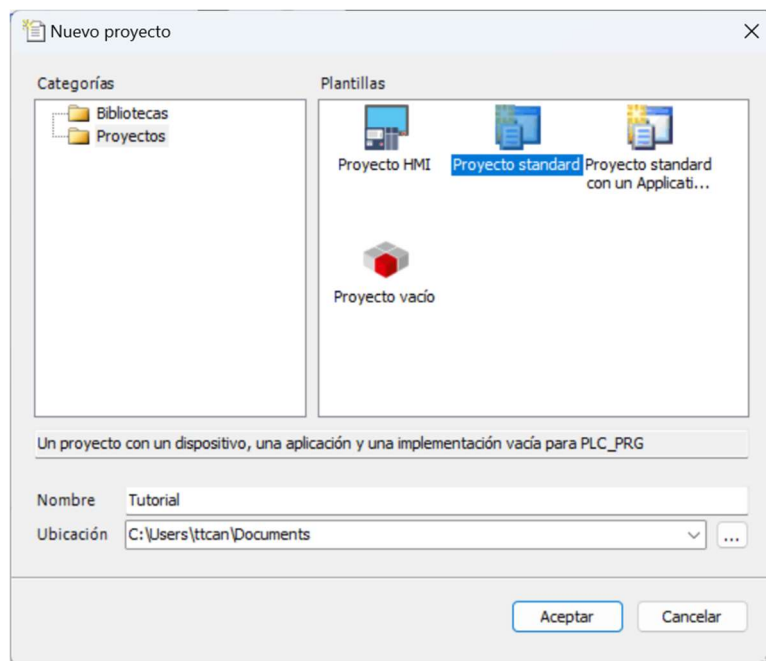


Ilustración 64: SFC cinta final 2

## 17. MANUAL DEL PROGRAMADOR

A continuación, se detallan los pasos a seguir para crear un proyecto con el software CODESYS y como realizar su comunicación con el software FACTORY IO donde se simulará lo programado a través del Modbus TCP Slave device. Debido a su diseño intuitivo y facilidad de uso no se considera necesario incluir una descripción detallada del software Factory IO en este manual, el usuario podrá utilizarlo eficazmente sin necesidad de instrucciones adicionales.

Primero de todo, se crea un nuevo proyecto en CODESYS donde se realizará la programación del proceso. Para ello seleccionamos la opción de Standard Project y se nombra el proyecto como se desee.



*Ilustración 43: Manual del programador 1*

Después se escoge el dispositivo que se quiera utilizar, en este caso es CODESYS Control Win V3 (3S-Smart Software Solutions GmbH) y el lenguaje de programación en el que se quiera trabajar. Los utilizados en este proyecto son Grafcet, Ladder y Structured text.

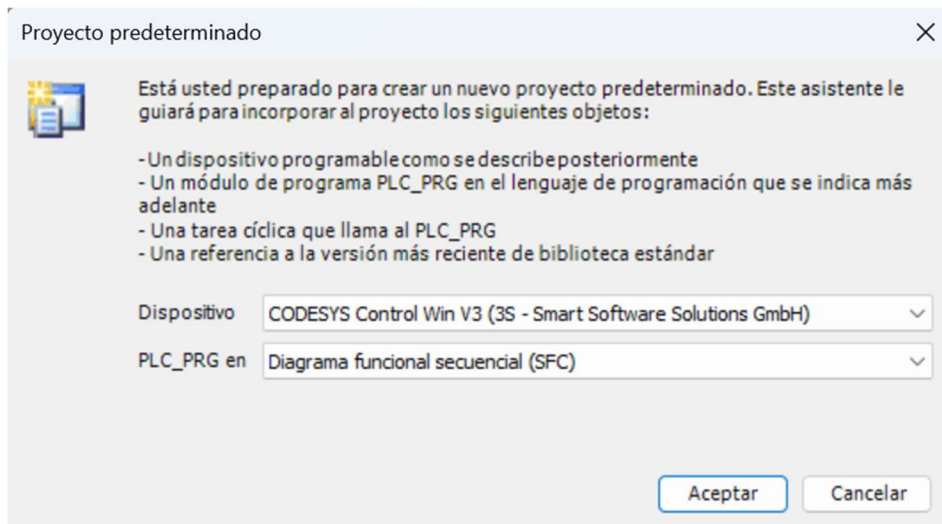


Ilustración 44: Manual del programador 2

Una vez creado el proyecto, el primer paso es crear una lista de variables globales para almacenar todas las variables que serán transferidas al Factory IO. Para ello, habrá que hacer click derecho sobre la pestaña “Application” y seleccionar “Add object” y ahí se encuentra la opción “Lista de variables globales”. A la hora de utilizar las variables en el código será necesario llamarlas de la siguiente forma: GVL.nombredelavariablen haciendo referencia que se trata de una variable global.

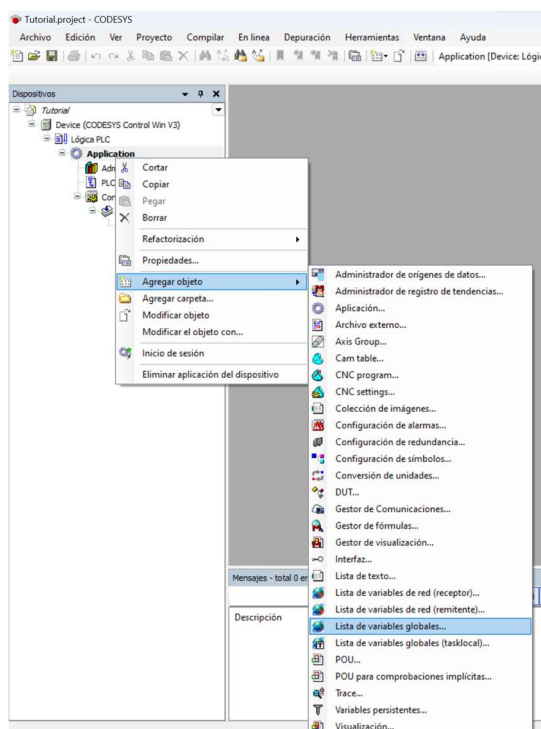


Ilustración 45: Manual del programador 3

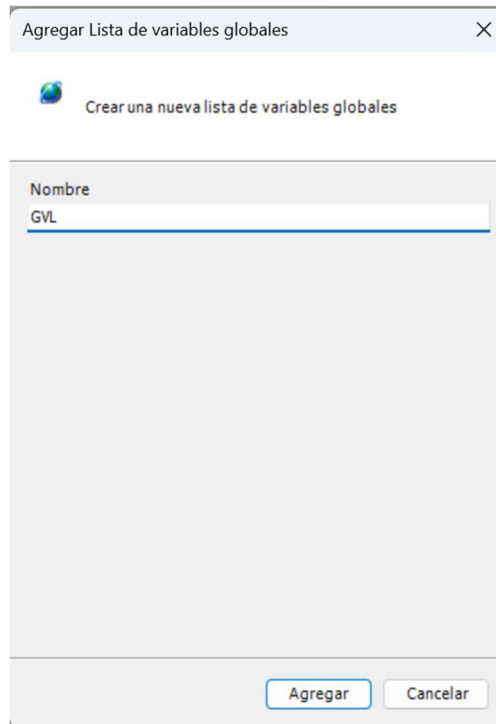


Ilustración 46: Manual del programador 4

Además de estas variables globales, se tendrán una serie de variables locales que se nombran al inicio del propio código, ya que estas no tienen que ser transferidas al Factory IO.

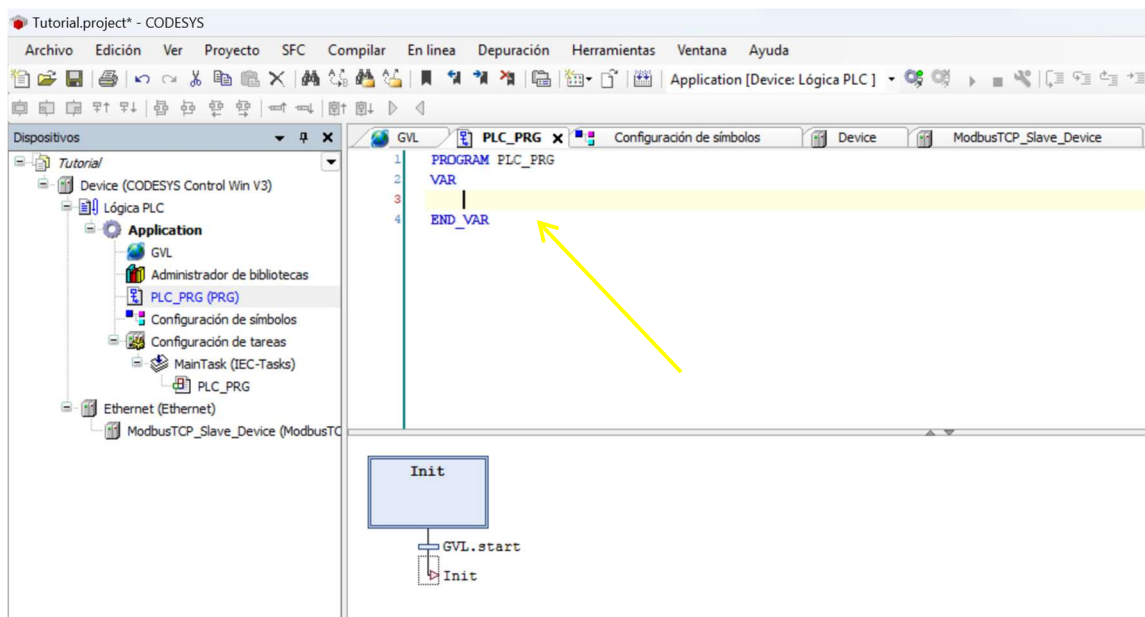


Ilustración 47: Manual del programador 5

Con ambas listas de variables creadas, se procede a realizar la programación del proceso, en su mayoría en lenguaje SFC. La sección de herramientas de la derecha nos permite crear pasos, transiciones, acciones, saltos, macros (conjuntos de pasos y transiciones) y ramas. Estos se añaden arrastrándolos desde la pestaña de la derecha al graficet que se está creando en el centro. En la barra superior encontramos las opciones para crear conjuntos de paso transición, acciones, ramas paralelas, similar a lo que se tiene en la pestaña derecha.

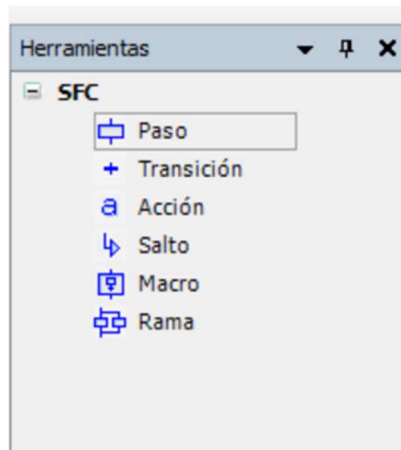


Ilustración 48: Manual del programador 6

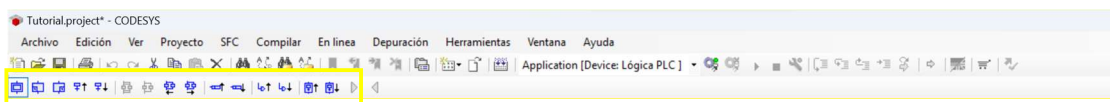


Ilustración 49: Manual del programador 7

Una vez creado el código, hacer click derecho en “Application”, “Agregar objeto” y seleccionar la opción “Configuración de símbolos”. Dentro de esta, clicar en crear y marcar con un tick la carpeta de GVL.

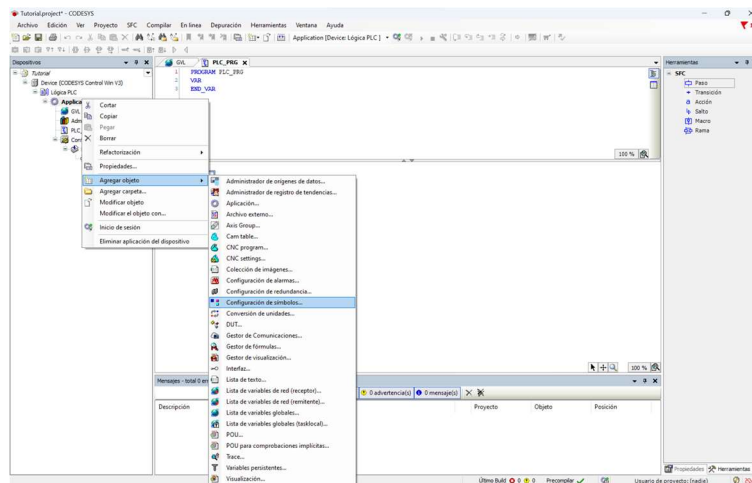


Ilustración 50: Manual del programador 8

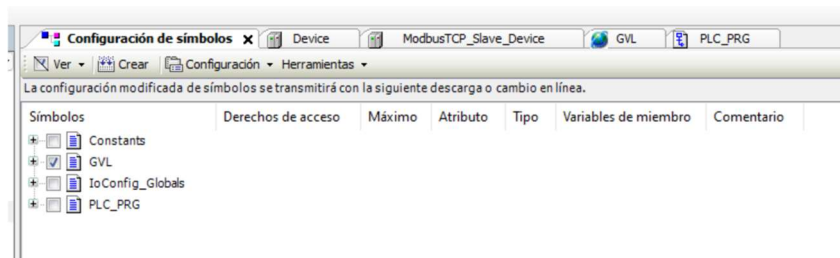


Ilustración 51: Manual del programador 9

Para inicializar el PLC, dirigirse a la parte inferior derecha de la barra de tareas del ordenador y clicando en el icono CODESYS Control Win Systray se escoge la opción “Start

PLC”. Hay que asegurarse también de que esté en modo running el icono CODESYS Gateway SysTray, situado en la misma zona inferior derecha de la barra de tareas.

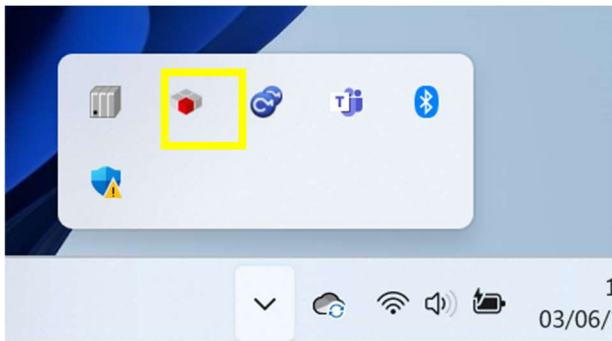


Ilustración 52: Manual del programador 10

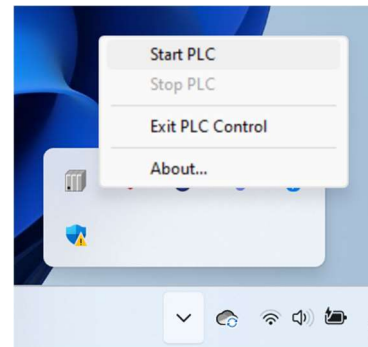


Ilustración 53: Manual del programador 11



Ilustración 54: Manual del programador 12

Después, en CODESYS hay que clicar en la pestaña Device (CODESYS Control Win V3), seleccionar Configuración de comunicación y escanear red. Se abre una pestaña en la que se debe seleccionar el dispositivo que aparecerá con el nombre DESKTOP-...

Para realizar la conexión, es necesario añadir un adaptador ethernet. Dirigirse a la pestaña de la izquierda llamada dispositivos, hacer click derecho sobre “device” y escoger la opción “agregar dispositivo”. Dentro de adaptador ethernet escoger el dispositivo Ethernet y agregar el dispositivo.

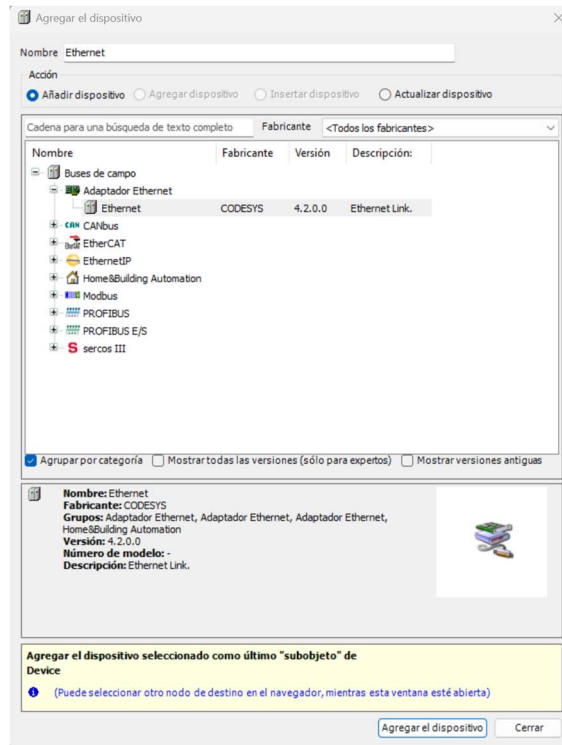


Ilustración 55: Manual del programador 13

Después hacer click derecho sobre el dispositivo ethernet y repetir el proceso para escoger el Modbus TCP Slave Device.

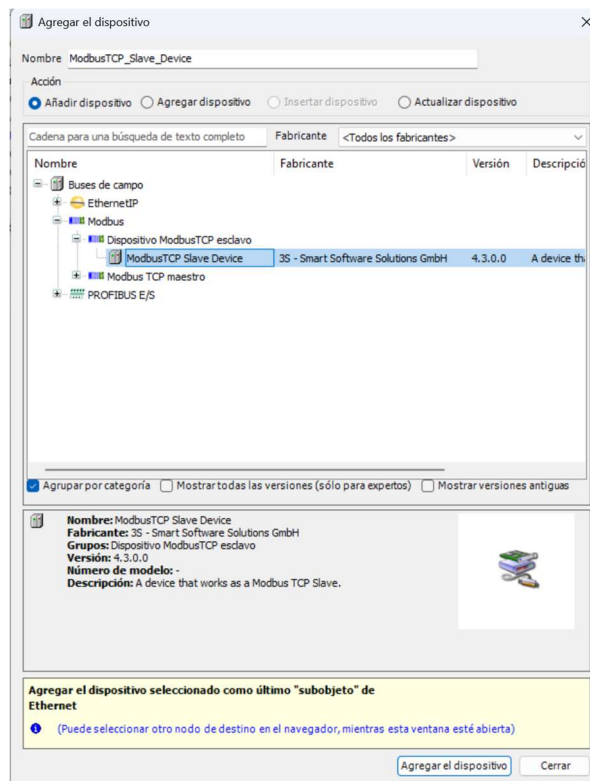


Ilustración 56: Manual del programador 14

Hacer click en el dispositivo Modbus y en el apartado “General” hay que añadir las variables globales que van a transferirse al Factory IO. Para añadir las entradas y salidas de tipo

booleano hay que seleccionar la opción Áreas de Bits discretos, donde podemos añadir el número de bobinas, que serán las entradas y entradas discretas que serán las salidas. Las variables enteras son registros holding para las entradas y registros input para las salidas.

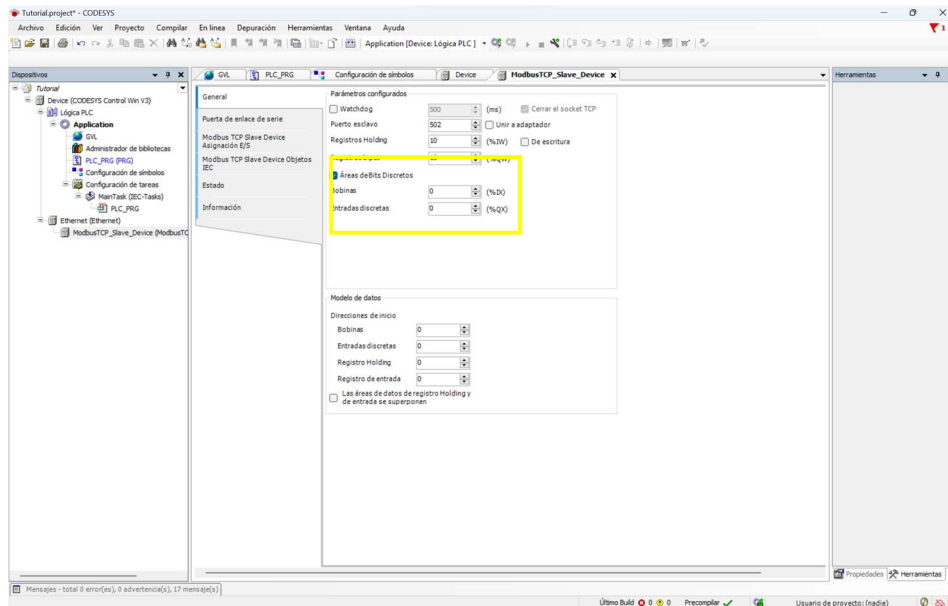


Ilustración 57: Manual del programador 15

Después, clicar en la pestaña Modbus TCP Slave Device Asignación E/S para realizar la asignación.



Ilustración 58: Manual del programador 16

Por último, habría que compilar el programa para ver que no existen errores. Se hace click en “En línea” y en “Iniciar la sesión” y en el apartado depuración clicamos en “start”.



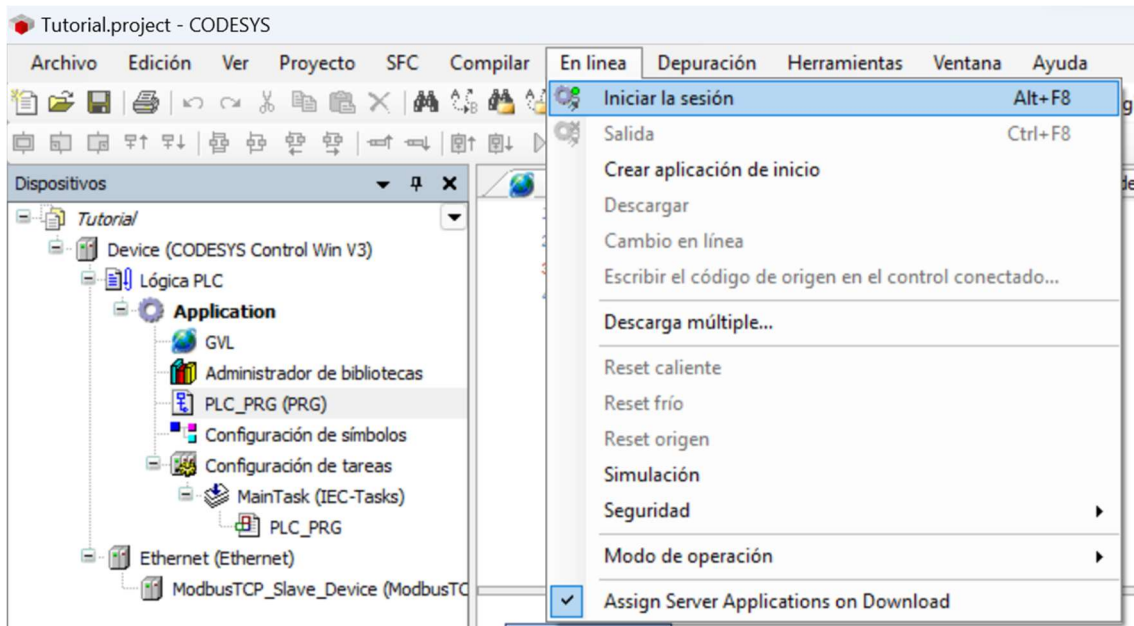


Ilustración 59: Manual del programador 17

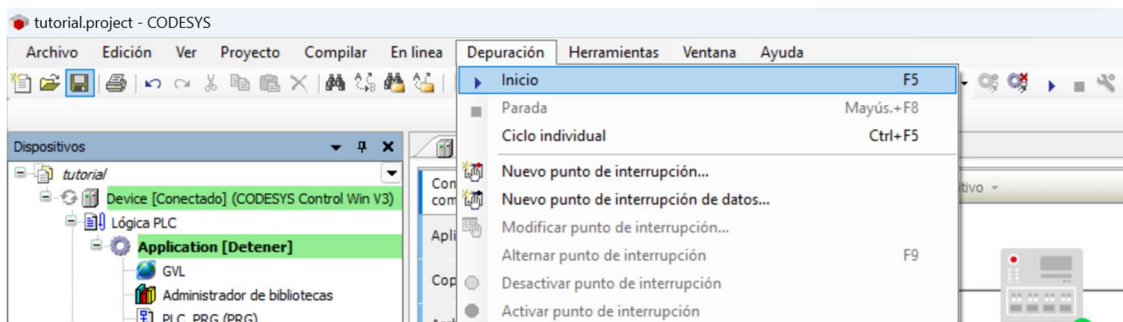


Ilustración 60: Manual del programador 18

A partir de este punto se comienza con el Factory IO. Primero se abre el proyecto y dentro de la ventana file, apartado drivers, se elige la opción Modbus TCP/IP Client. Una vez seleccionado hacer click en el botón configuración.

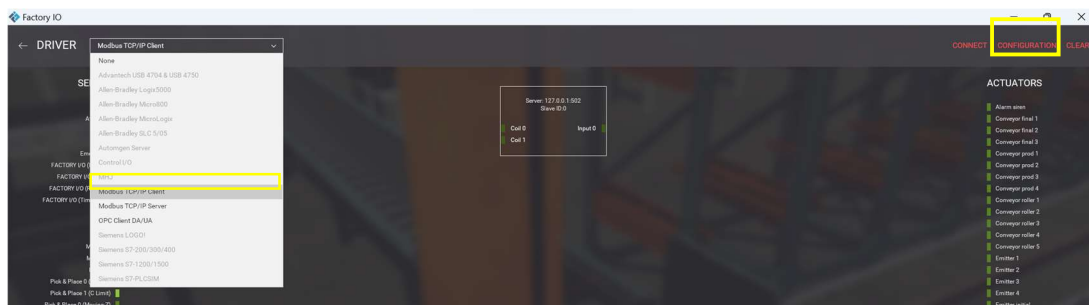


Ilustración 61: Manual del programador 19

En configuración hay que escribir un 0 o un 255 donde pone Slave ID y en el apartado I/O Points seleccionar en la columna count el número de entradas booleanas (digital inputs), salidas booleanas (digital outputs), entradas enteras (register inputs) y salidas enteras (register outputs).

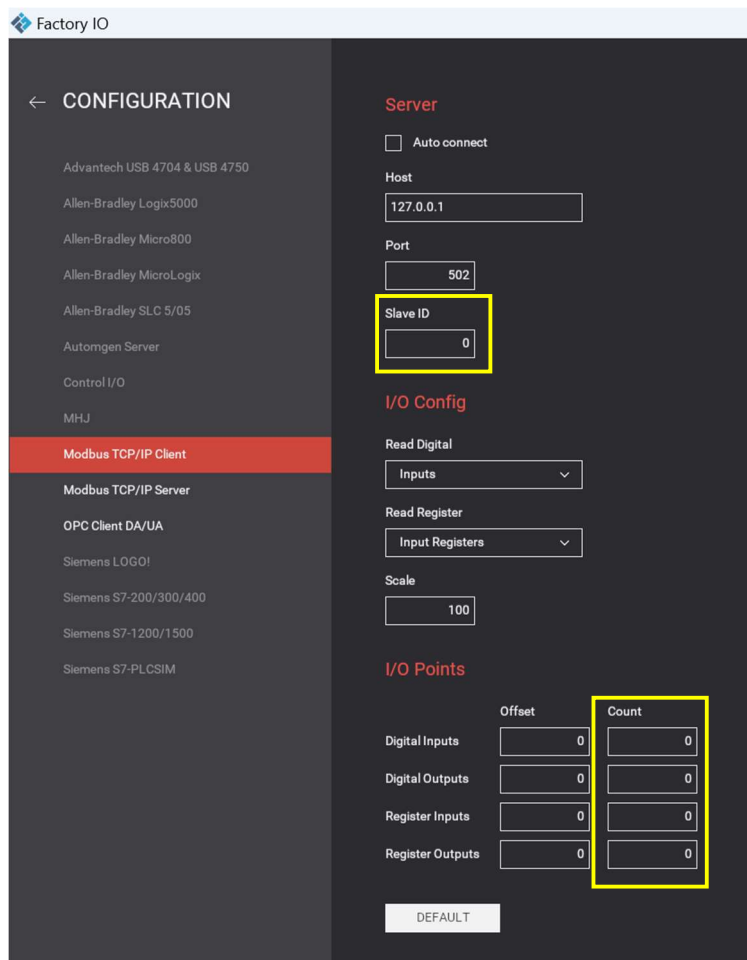


Ilustración 62: Manual del programador 20

Por último, al volver a la pantalla drivers y colocar todas las variables en su correspondiente posición, seleccionamos la opción conectar (Maibela, 2021).

Para finalizar la simulación, el orden correcto es primero parar la simulación del Factory IO, ir a la ventana drivers y seleccionar desconectar. Luego habría que ir a CODESYS y darle al botón Stop y después en la pestaña en línea seleccionar “log out”. Siguiendo esta serie de pasos se pueden evitar problemas de funcionamiento.

## 18. CONCLUSIONES

Con este trabajo se quiere poner énfasis en la importancia que la tecnología de los gemelos digitales está adquiriendo con el paso del tiempo en el ámbito de la automatización industrial. Una herramienta que resulta ser extremadamente eficiente y de utilidad para la creación y mejora de todo tipo de procesos industriales.

Mediante estos dos softwares se ha conseguido los objetivos planteados inicialmente, diseñar y automatizar un proceso de preparación de pedidos buscando una mayor eficiencia, reducción de costes y una mayor seguridad en la planta.

Recalcar también cómo el software Factory IO proporciona un entorno didáctico que permite al usuario aprender cómo funciona la automatización en el ámbito industrial. Está diseñado de forma que resulte fácil de utilizar y proporciona un entorno muy visual, que incluso

los errores que se puedan cometer se observan en la planta. Además, este software no solo permite realizar proyectos desde el principio, sino puede ser también útil para la mejora de procesos ya implementados en una planta industrial. Sin embargo, como se menciona a lo largo del proyecto, es un software que tiene ciertas limitaciones, por lo que aún necesita desarrollo para poder dar mayor flexibilidad al usuario.

Por otro lado, el software CODESYS, utilizado por muchas empresas en la actualidad, es potente y proporciona libertad al usuario al haber tantos lenguajes con los que poder programar. Concretamente en este proyecto se han combinado dos tipos de lenguajes, el SFC y Ladder.

Gracias a la realización de este trabajo se ha profundizado en el mundo de la automatización, que hoy en día avanza cada vez más, y sobre todo en dos softwares muy útiles con los que no se había trabajado hasta el momento. Además, ha proporcionado la oportunidad de tener una aplicación práctica de conocimientos adquiridos durante el grado como puede ser el diseño de graficets o el uso de PLCs.

## 19. BIBLIOGRAFÍA

### ➤ **Diseño de graficets:**

UPV (2024). Apuntes de Tecnología Automática.

### ➤ **Automatización en la industrial:**

Totvs. (25 de enero de 2024). Automatización industrial: una guía completa y actualizada sobre el tema. <https://es.totvs.com/blog/gestion-industrial/automatizacion-industrial-una-guia-completa-y-actualizada-sobre-el-tema/>

### ➤ **Aceleración automatización con la COVID-19:**

Pictet Asset Managment. (Noviembre 2020). La digitalización y la era de la pandemia. <https://am.pictet/es/spain/mega/impacto-de-la-covid-19-en-la-automatizacion>

### ➤ **Características Factory IO:**

Dominguez, A. Factory IO- Simulador 3D para PLC. Programación multidisciplinar. <https://www.programacionmultidisciplinar.com/factory-io/>

### ➤ **Página de ayuda CODESYS:**

CODESYS Group. CODESYS Online Help. <https://www.helpme-codesys.com/>

### ➤ **Características CODESYS:**

CODESYS Group. Why CODESYS. <https://www.codesys.com/the-system/why-codesys.html>

Larraioz Group. Codesys, convierte cualquier dispositivo en un PLC. <https://larraioz.com/blog/articulos-es/codesys-convierte-cualquier-dispositivo-en-un-plc>

### ➤ **Manual Factory IO:**

Real Games. Manual Factory IO. <https://docs.factoryio.com/manual/>

### ➤ **Conexión Factory IO y CODESYS mediante Modbus TCP Slave:**

Maibela, N. (25 de diciembre de 2021). *CONNECTING FACTORY IO WITH CODESYS MODBUS* [Video]. Youtube. [https://youtu.be/T3wnmmU6q9c?si=Cesz5DTwXHm-c\\_ra](https://youtu.be/T3wnmmU6q9c?si=Cesz5DTwXHm-c_ra)

### ➤ **Bloques de funciones CODESYS:**

Abner, C. (16 de junio de 2020). *CODESYS 3.5- CREAR BLOQUES DE FUNCIONES* [Video]. Youtube. <https://youtu.be/8G0RSDbKyXQ?si=385C-oKE6l0sGyNL>

### ➤ **Campos numéricos en visualizaciones CODESYS:**

Abner, C. (23 de junio de 2020). *CODESYS 3.5- AGREGAR CAMPOS NUMÉRICOS EN VISUALIZACIONES*[Video]. Youtube.

<https://youtu.be/0CBRaYLhcSc?si=tAv5k3D3Fv4iIDiu>

➤ **Campos de texto en visualizaciones CODESYS:**

Abner, C. (24 de junio de 2020). *CODESYS 3.5- AGREGAR CAMPO DE TEXTO EN VISUALIZACIONES* [Video]. Youtube.

[https://youtu.be/NqkMC71De0Q?si=Vhbr7q0HM\\_fsWdfG](https://youtu.be/NqkMC71De0Q?si=Vhbr7q0HM_fsWdfG)

➤ **Tipos de lenguajes:**

SATOSHI. (12 de marzo de 2017). *Codesys ®: IL, SFC, LD, FBD o ST, ¿Cuál elijo?*. Opcion. <https://www.opiron.com/codesys-mejor-lenguaje-programacion/>

➤ **Objetivos de desarrollo sostenible:**

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
Escuela Técnica Superior de Ingeniería Industrial

DESARROLLO DE LA AUTOMATIZACIÓN DE UN  
SISTEMA DE PREPARACIÓN DE PEDIDOS Y DEL  
GEMELO DIGITAL DE ESTE MEDIANTE EL  
SOFTWARE FACTORY IO

# PRESUPUESTO

Trabajo Fin de Grado  
Grado en Ingeniería en Tecnologías Industriales

## ÍNDICE PRESUPUESTO:

1. COSTES DIRECTOS.....	1
1.1. Mano de obra.....	1
1.2. Material.....	1
1.3. Amortización .....	1
2. COSTES INDIRECTOS .....	2
2.1. Electricidad.....	2
2.2. Internet.....	2
3. UNIDADES DE OBRA .....	2
4. PRECIOS DESCOMPUESTOS .....	3
5. PRESUPUESTO FINAL.....	5

# 1. COSTES DIRECTOS

Son aquellos costes que se asignan directamente al proyecto en cuestión. A continuación, se describen los costes directos que se han tenido en cuenta en este proyecto.

## 1.1. Mano de obra

El proyecto ha sido realizado por un Ingeniero Industrial cuyo sueldo mensual se puede estimar en unos 4500 euros brutos. Teniendo en cuenta la jornada laboral en España de 8 horas al día, el ingeniero trabaja 160 horas al mes. Con todo ello se concluye que el precio unitario de la mano de obra es de 28 euros/hora.

## 1.2. Material

El material que se ha utilizado en el proyecto es un ordenador de gama alta, por la necesidad de una potencia considerable al utilizarse para programar, que se estima en unos 1500 euros y además se hace uso del paquete Microsoft Office 365 para la redacción del proyecto que supone un coste de 70 euros/año.

El software Factory IO Modbus & OPC Edition tiene un coste de 435 euros y el software CODESYS es gratuito.

## 1.3. Amortización

Para calcular el coste unitario, se ha considera que el ordenador portátil y los softwares tienen un periodo de amortización de 5 años, funcionando 230 días al año (teniendo en cuenta solo los días laborables y considerando 30 días de vacaciones al año) y 8 horas al día. Para el paquete Microsoft 365 se ha tenido en cuenta el mismo número de días y horas, pero su amortización es respecto a 1 año que es el tiempo de la licencia. A continuación, se muestra una tabla en la que se indica toda la información que se ha descrito:

<b>Elementos</b>	<b>Coste inicial (€)</b>	<b>Vida útil (años)</b>	<b>Vida útil (horas)</b>	<b>Coste unitario (€/h)</b>	<b>Amortización Anual (€)</b>
Portátil	1500	5	9200	0,163	300
Software Factory IO	435	5	9200	0,047	87
Paquete Microsoft 365	70	1	1840	0,038	70
<b>TOTAL</b>					<b>457</b>

*Tabla 1: Amortización*



## 2. COSTES INDIRECTOS

Son aquellos costes que no están asociados directamente al proyecto, pero son necesarios para que se lleve a cabo. A continuación, se describen los costes indirectos que se han tenido en cuenta en este proyecto.

### 2.1. Electricidad

Para el cálculo de la electricidad que se consume para la realización del proyecto se tiene en cuenta el consumo de los dispositivos utilizados y el coste por kilovatio-hora. Se ha calculado un coste de 185 euros al año aproximadamente.

### 2.2. Internet

En cuanto a la conexión a Internet, es necesaria para la licencia online de Factory IO, y se ha optado por la opción de la fibra óptica. Se calcula que su precio se encuentra alrededor de los 25 euros al mes.

## 3. UNIDADES DE OBRA

UD01: Análisis de alternativas: posibles soluciones al problema planteado.

UD02: Modelado: diseño del automatismo mediante el software Factory IO.

UD03: Programación: Programación del automatismo mediante el software CODESYS.

UD04: Conexión de los softwares: Conexión de los softwares mediante Modbus TCP.

UD05: Correcciones del proyecto: ajustes y comprobaciones de funcionamiento.

UD06: Redacción de los documentos del proyecto: redacción de la memoria, presupuesto y pliego de condiciones.

## 4. PRECIOS DESCOMPUESTOS

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD01	u	Análisis de alternativas			
MO01	h	Ingeniero Industrial	15	28	420
MA01	u	Ordenador portátil	15	0,163	2,445
CI01	h	Electricidad	15	0,1	1,5
CI02	h	Internet	15	0,163	2,445
				<b>TOTAL</b>	<b>426,39 €</b>

Tabla 2: Cuadro de precios del análisis de alternativas

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD02	u	Modelado			
MO01	h	Ingeniero Industrial	20	28	560
MA01	u	Ordenador portátil	20	0,163	3,26
MA03	u	Software Factory IO	20	0,047	0,94
CI01	h	Electricidad	20	0,1	2
CI02	h	Internet	20	0,163	3,26
				<b>TOTAL</b>	<b>569,46 €</b>

Tabla 3: Cuadro de precios del modelado

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD03	u	Programación			
MO01	h	Ingeniero Industrial	70	28	1960
MA01	u	Ordenador portátil	70	0,163	11,41
MA02	u	Software CODESYS	70	0	0
MA03	u	Software Factory IO	25	0,047	1,175
CI01	h	Electricidad	70	0,1	7
CI02	h	Internet	25	0,163	4,075
				<b>TOTAL</b>	<b>1.983,66 €</b>

Tabla 4: Cuadro de precios de la programación

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD04	u	Conexión de los softwares			
MO01	h	Ingeniero Industrial	5	28	140
MA01	u	Ordenador portátil	5	0,163	0,815
MA02	u	Software CODESYS	5	0	0
MA03	u	Software Factory IO	5	0,047	0,235
CI01	h	Electricidad	5	0,1	0,5
CI02	h	Internet	5	0,163	0,815
				<b>TOTAL</b>	<b>142,37 €</b>

Tabla 5: Cuadro de precios de la conexión de los softwares

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD05	u	Correcciones			
MO01	h	Ingeniero Industrial	20	28	560
MA01	u	Ordenador portátil	20	0,163	3,26
MA02	u	Software CODESYS	20	0	0
MA03	u	Software Factory IO	10	0,047	0,47
CI01	h	Electricidad	20	0,1	2
CI02	h	Internet	20	0,163	3,26
				<b>TOTAL</b>	<b>568,99 €</b>

*Tabla 6: Cuadro de precios de las correcciones*

Código	Unidades	Descripción	Rendimiento	Precio (€)	Importe (€)
UD06	u	Redacción documentos			
MO01	h	Ingeniero Industrial	60	28	1680
MA01	u	Ordenador portátil	60	0,163	9,78
MA04	u	Microsoft Office 365	60	0,038	2,28
CI01	h	Electricidad	60	0,1	6
				<b>TOTAL</b>	<b>1.698,06 €</b>

*Tabla 7: Cuadro de precios de la redacción de documento*

## 5. PRESUPUESTO FINAL

A partir de los cuadros de precios descompuestos que se muestran anteriormente, se calcula el presupuesto de ejecución material como la suma del total de cada unidad de obra. A partir de este, se calculan los gastos generales, que son los gastos que se han tenido durante el proyecto, y el beneficio industrial, que es el beneficio que obtiene la empresa que lleva a cabo el proyecto. La suma de todo permite obtener el presupuesto de ejecución por contrata y sumando el IVA es como se obtiene el presupuesto base de licitación.

Unidad de obra	Importe (€)
Análisis de alternativas	426,39
Modelado	569,46
Programación	1.983,66
Conexión de los softwares	142,37
Correcciones del proyecto	568,99
Redacción de documentos	1.698,06
<b>PRESUPUESTO DE EJECUCIÓN MATERIAL</b>	<b>5.388,93</b>
Gastos generales (13%)	700,56
Beneficio Industrial (6%)	323,34
<b>PRESUPUESTO DE EJECUCIÓN POR CONTRATA</b>	<b>6.412,82</b>
I.V.A. (21%)	1.131,67
<b>PRESUPUESTO BASE DE LICITACIÓN</b>	<b>7.544,50</b>

*Tabla 8: Cuadro del presupuesto final*

Asciende el presente presupuesto a la expresada cantidad de:

SIETE MIL QUINIENTOS CUARENTA Y CUATRO EUROS CON CINCUENTA CÉNTIMOS.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

DESARROLLO DE LA AUTOMATIZACIÓN DE UN  
SISTEMA DE PREPARACIÓN DE PEDIDOS Y DEL  
GEMELO DIGITAL DE ESTE MEDIANTE EL  
SOFTWARE FACTORY IO

# PLIEGO DE CONDICIONES

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

## ÍNDICE PLIEGO DE CONDICIONES:

1. OBJETIVO Y ALCANCE DEL PLIEGO .....	1
2. DOCUMENTOS QUE DEFINEN EL PROYECTO.....	1
3. CONDICIONES GENERALES.....	1
4. CONDICIONES PARTICULARES .....	2
4.1. Técnicas .....	2
4.1.1. Especificaciones de Hardware .....	2
4.1.2. Especificaciones de Software.....	2
4.1.3. Instalación.....	2
4.1.4. Garantía .....	2
4.2. Facultativas .....	3
4.2.1. Modificaciones y defectos en el proyecto .....	3
4.2.2. Plazo de ejecución.....	3
4.2.3. Recepción provisional .....	3
4.2.4. Plazo de garantía.....	3
4.2.5. Recepción definitiva.....	3
4.3. Económicas.....	3
4.3.1. Garantías.....	3
4.3.2. Fianza.....	3
4.3.3. Presupuesto del proyecto.....	4
4.3.4. Modificaciones .....	4
4.3.5. Revisión de precios .....	4
4.3.6. Pagos.....	4
4.3.7. Penalizaciones .....	4
4.4. Legales.....	4
4.4.1. Responsabilidad del proyectista .....	4
4.4.2. Responsabilidad del programador .....	4
4.4.3. Causas de rescisión del contrato.....	4

## 1. OBJETIVO Y ALCANCE DEL PLIEGO

El objetivo del pliego de condiciones es fijar las condiciones mínimas que debe cumplir el desarrollo e implementación de la automatización mediante PLC de una planta virtual de preparación de pedidos. En el proyecto se utiliza el software Factory IO para el modelado de la planta y se programa con el software CODESYS.

El alcance del proyecto abarca la simulación del proceso de empaquetado de productos, almacenaje temporal en estantería y recogida y entrega de paquetes a las distintas líneas de envío de las que dispone la planta, todo ello desde una implementación virtual y no física.

Al tratarse de un proyecto informático solo se abarcan los aspectos referidos a la programación de la planta industrial y a su puesta en marcha virtual.

## 2. DOCUMENTOS QUE DEFINEN EL PROYECTO

Los documentos que definen el proyecto pueden ser de índole contractual o meramente informativa. Son documentos contractuales el Pliego de Condiciones y Presupuesto Total, que se incluyen en el presente proyecto. Cualquier cambio de planteamiento que implique un cambio sustancial respecto de lo proyectado, deberá ponerse en conocimiento de la Dirección Técnica para que lo apruebe, si procede, y redacte el oportuno proyecto reformado.

## 3. CONDICIONES GENERALES

A continuación, se detallan las normas de carácter general y que han de cumplirse obligatoriamente para la realización del proyecto:

- IEC 61131-3: Norma estándar internacional industrial para los lenguajes de programación de PLCs.
- IEC 61158: Protocolo de comunicación industrial basado en Ethernet.
- IEC 61158: Norma estándar para el protocolo de comunicación Modbus TCP.
- IEC 61508: Normativa para la seguridad funcional de sistemas eléctricos, electrónicos y programables electrónicos relacionados con la seguridad (E/E/PE).
- IEC 61499: Esta norma se centra en la arquitectura para sistemas de control distribuidos.
- IEC 61800: Conjuntos de normas para sistemas de accionamiento de velocidad variable.
- IEC 61850: Norma para la comunicación en subestaciones eléctricas y sistemas de automatización relacionados.
- IEC 62061: Norma para la seguridad funcional de sistemas de control relacionados con la maquinaria.
- ISO 13849-1: Seguridad de las máquinas. Partes de sistemas de control relacionadas con la seguridad. Parte 1: Principios generales de diseño.
- EN 60204-1: Seguridad de las máquinas. Equipos eléctricos de las máquinas. Parte 1: Requisitos generales.
- ISO 12100: Seguridad de las máquinas. Principios generales de diseño. Evaluación de riesgos y reducción del riesgo.
- Reglamento de Instalaciones Térmicas en los Edificios (RITE): Reglamento español que establece las condiciones que deben cumplir las instalaciones térmicas de los edificios para atender la demanda de bienestar e higiene de las personas.
- Reglamento Electrotécnico para Baja Tensión (REBT): Reglamento español que establece las condiciones técnicas y garantías que deben reunir las instalaciones eléctricas conectadas a una fuente de suministro en baja tensión.

- Directiva de Máquinas 2006/42/CE: Directiva europea que establece los requisitos esenciales de seguridad y salud para la comercialización de máquinas dentro del Espacio Económico Europeo.

## 4. CONDICIONES PARTICULARES

### 4.1. Técnicas

#### 4.1.1. Especificaciones de Hardware

**Memoria RAM:** como mínimo 4GB, aunque se recomiendan 8GB. Si se van a utilizar ambos softwares y otras aplicaciones a la vez se recomienda una memoria de 16 GB para un funcionamiento rápido y eficiente.

**Almacenamiento:** que se disponga como mínimo de 10 GB para la instalación de los dos softwares.

**Pantalla:** resolución mínima de Mínimo 1366x768 píxeles y al menos 14 pulgadas para una experiencia visual cómoda. Como recomendación, pantalla Full HD (1920x1080 píxeles) o superior y de 15.6 pulgadas o más.

**Tarjeta gráfica:** con soporte DirectX 11 y memoria VRAM de 2 GB (recomendado 4 GB o más para evitar problemas al manejar gráficos complejos en Factory IO).

**Ratón:** no es imprescindible, pero es altamente recomendable.

**Conexión:** puertos USB, Ethernet (RJ-45) recomendado para una conexión de red estable y rápida.

#### 4.1.2. Especificaciones de Software

**Sistema operativo:** Se admite Windows 7, 8, 10 y 11. Se recomienda Windows 10 (64 bits) o superior al tener mejor rendimiento, más actualizaciones de seguridad y mayor soporte a largo plazo que sistemas operativos anteriores.

**Factory IO:** Factory IO requiere de internet al ser la licencia online

#### 4.1.3. Instalación

El encargado de la instalación de los dos softwares será siempre el programador, ya que tiene conocimientos sobre los softwares que el cliente puede desconocer.

#### 4.1.4. Garantía

En caso de pérdida o eliminación por error de algún fichero, el programador se compromete a facilitar una copia de dichos ficheros o del programa completo al cliente en un tiempo no superior a 5 días laborables. El cliente deberá demostrar la pérdida de estos ficheros al programador para que surja efecto esta garantía, que tiene una duración de 3 años desde la entrega definitiva del proyecto. Si tras este periodo se requiere de una nueva copia, se deberá llegar a un pacto que defina un nuevo precio por suministrar una nueva copia del programa.

Cualquier cambio o modificación del programa, instalación no autorizada o copia supondrá la anulación de todas las condiciones de garantía mencionadas con anterioridad.

Ni el programador ni el proyectista se hacen responsables de los posibles daños que ocurran debidos a una mala utilización del programa.



## 4.2.Facultativas

### 4.2.1. Modificaciones y defectos en el proyecto

El cliente es el responsable de analizar el funcionamiento del programa e indicar aquellos aspectos que bajo su juicio no sean correctos. Dicha responsabilidad también engloba aquellos defectos presentes en el programa, pero no detectados.

El cliente podrá introducir en el proyecto modificaciones, aunque no se hayan previsto en el proyecto, y siempre que el Proyectista considere que no suponen una alteración de la función que tiene el proyecto y del espíritu del proyecto.

Cualquier desperfecto que englobe los softwares que se utilizan en el proyecto, no quedan bajo responsabilidad del proyectista, siendo el único causante de estos desperfectos el proveedor de estos.

En el supuesto de que los defectos se deban a fallos de programación el proyectista se compromete a solventarlos sin costes adicionales.

### 4.2.2. Plazo de ejecución

El proyecto será realizado según lo estipulado en la memoria y el programador deberá cumplir con su entrega en el plazo de tiempo establecido con el cliente.

### 4.2.3. Recepción provisional

Una vez el proyecto se finaliza ejecutado siguiendo las condiciones establecidas, se procede con la recepción provisional, cuyo plazo de garantía desde esa fecha exacta será de dos meses.

### 4.2.4. Plazo de garantía

Desde el plazo de la entrega del proyecto al cliente, se empieza a contar el plazo de garantía que será de 3 años. Durante este periodo, el programador se hará cargo de todas aquellas reparaciones requeridas por el programa, exceptuando aquellas que se deban a un mal uso del mismo que quedarán fuera de esta garantía.

### 4.2.5. Recepción definitiva

Terminado el plazo de recepción provisional, se verificará la recepción definitiva con las mismas condiciones que la provisional, y si el proyecto no requiere de ninguna modificación o arreglo, termina la responsabilidad del proyectista, en caso contrario se retrasará la recepción definitiva hasta que, a juicio del cliente, el proyecto quede del modo y forma que se determinan en este Pliego.

## 4.3.Económicas

### 4.3.1. Garantías

Se podrá exigir al cliente la presentación de referencias bancarias u otras garantías, con el fin de asegurar que reúne todas las condiciones requeridas para el cumplimiento del contrato.

### 4.3.2. Fianza

Una vez asignado el proyecto y antes de su inicio, el comprador entregará una fianza del 5% del presupuesto del proyecto, que en caso de demora aumentará un 2% semanalmente.

El resto del coste del proyecto se abonará cuando finalice el proyecto y el cliente compruebe su correcto funcionamiento.

#### 4.3.3. Presupuesto del proyecto

Los precios correspondientes a los distintos elementos de las unidades de obra están actualizados. Todo lo referido a los costes del proyecto está explicado en extenso en el documento “Presupuesto”.

#### 4.3.4. Modificaciones

Las modificaciones que el cliente quiera realizar supondrán un incremento del Presupuesto fijado, ya que no son fundamentales para el funcionamiento del programa. Las tarifas que se aplican tras modificaciones serán las mismas que las que se han tenido en cuenta para el Proyecto.

#### 4.3.5. Revisión de precios

Si durante el tiempo de realización del proyecto se producen cambios en los precios de la mano de obra y los materiales, se admite la revisión de precios únicamente al alza.

#### 4.3.6. Pagos

El cliente realizará los pagos en los plazos acordados, y su importe estará claramente explicado y seguirá lo establecido en el presupuesto del proyecto.

#### 4.3.7. Penalizaciones

En caso de demora de los pagos, se añadirá un incremento del 1% del importe total del proyecto por cada semana de retraso en los pagos.

Si el programador sobrepasa el plazo de entrega acordado con el cliente, cada día de demora supondrá un coste al proyectista del 0,5% del presupuesto total del proyecto.

### 4.4. Legales

#### 4.4.1. Responsabilidad del proyectista

El proyectista es el responsable de la ejecución del proyecto siguiendo las condiciones establecidas en el contrato y en los documentos que componen el proyecto (Memoria, Presupuesto y Pliego de condiciones).

#### 4.4.2. Responsabilidad del programador

El programador es el responsable de toda la programación del proyecto y de lo referido a ésta en el contrato y en los documentos del proyecto (Memoria, Presupuesto y Pliego de condiciones).

#### 4.4.3. Causas de rescisión del contrato

Serán causas de rescisión del contrato las siguientes: la muerte o incapacitación del proyectista, la quiebra del proyectista y las alteraciones del contrato por las causas siguientes:

- La modificación del Proyecto en tal forma que represente alteraciones fundamentales a juicio del proyectista y del programador.
- Las modificaciones de unidades de obras siempre que éstas representen variaciones en más o menos del 40% como mínimo de algunas de las unidades del proyecto, o en más de un 50% de unidades del proyecto.
- El incumplimiento de las condiciones del contrato.
- La terminación del plazo de la ejecución del proyecto.
- La cancelación del proyecto por alguna de las partes sin causa justificada.