



UNIVERSITAT JAUME I

**ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES
EXPERIMENTALS**

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

***DISEÑO DE UN GEMELO DIGITAL DE
UN PROCESO DE PALETIZADO Y
DESARROLLO DEL SISTEMA DE
CONTROL AUTOMÁTICO DEL MISMO.***

TRABAJO FIN DE GRADO

AUTOR/A

Carlos Ávila Foix

DIRECTOR/A

Roberto Sanchís Llopis

Castellón, Julio de 2022

Agradecimientos:

A mi tutor de TFG Roberto Sanchís por facilitarme la idea inicial y por sus consejos durante el proceso de elaboración del proyecto.

A la empresa System España, donde pude realizar las prácticas curriculares de la UJI, que me permitieron conocer en profundidad un mundo nuevo para mí, pese a la cercanía del mismo como es la industria cerámica.

A mis compañeros de grado que en algún momento u otro nos hemos ayudado en diferentes tareas y con los que he compartido grupo de trabajo.

Al resto de profesores que he tenido durante el grado, por su profesionalidad y estar siempre disponibles en caso de dudas.

ÍNDICE GENERAL

MEMORIA.....	1
ANEXOS.....	69
PLIEGO DE CONDICIONES.....	133
PRESUPUESTO.....	141

Índice de Figuras

Figura 1: Logotipo de Factory I/O.....	8
Figura 2: Logotipo de la IEC.....	9
Figura 3: Espacio de trabajo en Factory I/O (vista general).....	12
Figura 4: Recorte de la pestaña Drivers de Factory I/O.....	13
Figura 5: Pieza de plástico (derecha) y metálica (izquierda).....	14
Figura 6: Entrada de Piezas y Empujador.....	14
Figura 7: Paletizador (vista general).....	15
Figura 8: Máquina Rotativa.....	15
Figura 9: Tramo de salida.....	16
Figura 10: Zona de control del proceso en el centro de la línea.....	16
Figura 11: Vista general de la zona de producción con perímetro de seguridad.....	17
Figura 12: Paso de Escalera montado.....	22
Figura 13: Grafcet 0 - Control del Cuadro General.....	24
Figura 14: Grafcet 1 - Control de Cintas y Rodillos.....	25
Figura 15: Grafcet 2 - Control de la Máquina Rotativa.....	27
Figura 16: Grafcet 3 - Control del Empujador de Piezas.....	28
Figura 17: Grafcets 4 y 5 - Control de los Paletizadores M y N.....	29
Figura 18: Grafcets 5 y 6 - Control del Cuadro de los Paletizadores M y N.....	31
Figura 19: Ventana de creación de Nuevo Proyecto en CODESYS.....	33
Figura 20: Pantalla de configuración del Proyecto Estándar.....	33
Figura 21: Árbol del proyecto en CODESYS tras la configuración inicial.....	33
Figura 22: Creación de la lista de Variables Global.....	34
Figura 23: Variables Globales 1/5.....	35
Figura 24: Variables Globales 2/5.....	35
Figura 25: Variables Globales 3/5.....	35
Figura 26: Variables Globales 4/5.....	35
Figura 27: Variables Globales 5/5.....	35

Figura 28: Etapas de un programa en LD.....	36
Figura 29: Variables locales de Etapas, Copia de Etapas e Inicialización.....	36
Figura 30: Otras Variables Locales que se utilizarán en el código.....	37
Figura 31: Reseteo de Etapas del Grafcet 0.....	37
Figura 32: Activación de la Etapa 0 del Grafcet 0.....	37
Figura 33: Barra de herramientas superior.....	38
Figura 34: Menú a la derecha del espacio de trabajo.....	38
Figura 35: Copia de valores de la variable E1 a la variable CE1.....	38
Figura 36: Detección de flanco de subida de la Etapa 1 del Grafcet 0.....	38
Figura 37: Desactivación Etapa 0 del Grafcet 0 y activación Etapa 1 del Grafcet 0....	39
Figura 38: Desactivación Etapa 0 del Grafcet 1 y activación de las etapas 1 – 6.....	39
Figura 39: Definición de las variables PosMOK y PosNOK.....	40
Figura 40: Uso de los operadores EQ y NE.....	40
Figura 41: Forzado de reseteo de etapas (en el Grafcet 0 - Etapa 0).....	41
Figura 42: Forzado de Grafcets a cierta etapa combinando AND y OR.....	41
Figura 43: Acción impulsional con caja de tipo EXECUTE.....	42
Figura 44: Comprobación de desigualdad entre E0 y CE0.....	42
Figura 45: Actuador que se activa bajo distintas Etapas y condiciones.....	43
Figura 46: Programación del descenso progresivo del brazo del Paletizador.....	43
Figura 47: Temporizador de espera en final de línea de 0,25s.....	43
Figura 48: Icono de CODESYS Control Win SysTray (parado).....	45
Figura 49: Icono de CODESYS Control Win SysTray (en marcha).....	45
Figura 50: Icono de CODESYS Gateway SysTray (en marcha).....	45
Figura 51: Ventana principal de Device en CODESYS.....	46
Figura 52: Ventana que se abre al presionar Scan Network.....	46
Figura 53: Desplegable de menú Device. Opción que se debe seleccionar.....	46
Figura 54: Ventana de Change Runtime Security Policy.....	47
Figura 55: Configuración en la pestaña de Access Rights.....	47
Figura 56: Icono de la opción Generate Code.....	47

Figura 57: Pestaña de Symbol Configuration.....	48
Figura 58: Botón de Login.....	48
Figura 59: Botón de Start.....	48
Figura 60: Pantalla de confirmación de cambios en la aplicación.....	48
Figura 61: Desplegable de la pestaña Drivers en Factory I/O.....	49
Figura 62: Página de Configuración en Factory I/O.....	49
Figura 63: Mapeo de Sensores y Actuadores 1/3.....	50
Figura 64: Mapeo de Sensores y Actuadores 2/3.....	50
Figura 65: Mapeo de Sensores y Actuadores 3/3.....	51
Figura 66: Captura del proceso tras 15 minutos de marcha.....	52
Figura 67: Vista zenital Paletizador N (salida de pallet).....	53
Figura 68: Vista zenital Paletizador M (salida de pallet).....	53
Figura 69: Espera Pallet piezas metálicas.....	53
Figura 70: Espera pallet piezas normales.....	53
Figura 71: El proceso se para al pulsar el botón de emergencia general.....	54
Figura 72: Se activan la luz y la alarma de emergencia.....	54
Figura 73: Pulsador de Silencio de Alarma activo (apaga alarma, mantiene luz).....	54
Figura 74: Emergencia en Paletizador M. Proceso parado y luces de advertencia.....	55
Figura 75: El otro paletizador sigue su marcha (luces verdes).....	56
Figura 76: Paro de la línea de paletizado al abrir una puerta.....	56
Figura 77: Botón de Solicitud de Paro pulsado (indicador luminoso).....	57
Figura 78: Ambos paletizadores terminan los últimos pallets. Entran en reposo.....	57
Figura 79: Paro completo tras la extracción de todos los pallets de la línea.....	58
Figura 80: Definición del contador de piezas.....	59
Figura 81: Cambio en la transición entre E3.0 y E3.1.....	59

Índice de Tablas

Tabla 1: Resumen coste temporal actividades.....	63
Tabla 2: Coste económico método tradicional.....	64
Tabla 3: Coste económico método propuesto.....	64
Tabla 4: Elementos del proceso: Características y Configuración.....	80
Tabla 5: Listado de Variables Globales - Sensores.....	83
Tabla 6: Listado de Variables Globales - Actuadores.....	87
Tabla 7: Coste mano de obra.....	144
Tabla 8: Presupuesto de Ejecución de Material.....	145
Tabla 9: Presupuesto de Ejecución por Contrata.....	145
Tabla 10: Presupuesto Total.....	145



MEMORIA



Índice de la Memoria

1.1. Antecedentes.....	5
1.2. Objetivos.....	6
1.3. Alcance.....	7
1.4. Normas y Referencias.....	8
1.4.1. Disposiciones legales y normas aplicadas.....	8
1.4.2. Programas utilizados.....	8
1.5. Definiciones y abreviaturas.....	9
1.6. Problemática.....	12
1.7. Análisis de soluciones.....	14
1.7.1. Diseño del proceso.....	14
1.7.1.1. Descripción detallada del proceso.....	14
1.7.1.2. Proceso de ensamblado de las partes.....	17
1.7.2. Variables Globales.....	23
1.7.3. Método de programación.....	24
1.7.4. Programación en CODESYS.....	32
1.7.5. Conexión CODESYS – Factory I/O.....	44
1.7.5.1. Configuración de CODESYS.....	44
1.7.5.2. Configuración de Factory I/O.....	49
1.8. Resultados finales.....	52
1.8.1. Comprobación del correcto funcionamiento de la línea.....	52
1.8.2. Posibles cambios en la línea y su implementación.....	58
1.9. Viabilidad técnica.....	62
1.10. Viabilidad económica.....	63
1.10.1. Ahorro temporal.....	63
1.10.2. Ahorro económico.....	64
1.11. Conclusiones.....	66
1.12. Bibliografía.....	67



1.1. Antecedentes

Tradicionalmente, el desarrollo de sistemas automatizados se ha fundamentado en el diseño de los componentes del proceso, su montaje en planta, la programación del sistema de control y la prueba y puesta a punto del mismo. En los últimos años, gracias al gran avance tecnológico por el que ha pasado la sociedad, han surgido alternativas que tratan de optimizar el proceso anterior, tanto en términos de reducción de tiempo como económicos.

Factory I/O es un programa que permite simular desde la más sencilla línea de producción, hasta montajes mucho más elaborados. Esto permite crear gemelos digitales de procesos que se van a llevar a cabo en la realidad y probar el control automático de los mismos, lo cual hace posible detectar y corregir errores previamente a la puesta en marcha de dicha máquina o proceso en la realidad, de forma que se reducen los riesgos para la seguridad de los trabajadores y se evitan accidentes con la propia maquinaria. De esta forma, se reducen costes adicionales de tiempo y dinero a la hora de la puesta a punto en la realidad.

Este proyecto se realizará íntegramente a través de programas informáticos y va a estar dividido en dos partes: diseño del proceso y desarrollo del sistema de control.

En primer lugar, se utilizará el ya mencionado software Factory I/O, herramienta que permite el montaje y representación en 3D de diferentes procesos industriales, así como la posterior simulación de su funcionamiento tras su programación. Factory I/O es un programa desarrollado por la empresa *'realgames'* y que fue lanzado en su versión inicial (1.0.0) en Julio de 2014. Desde entonces ha ido recibiendo progresivas mejoras, a la par que recibía premios como el otorgado por la prestigiosa compañía *'Unity Technologies'* en la categoría de Mejor Simulación en el año 2016; hasta llegar a la última versión lanzada, la 2.5.1 en octubre de 2021, que es sobre la que se realizará el presente trabajo.

Por otro lado, CODESYS Development System (de ahora en adelante CODESYS) será el programa utilizado para el desarrollo del sistema de control del proceso del mismo modo que permitirá utilizar el ordenador como PLC (*Programmable Logic Controller*) Virtual, lo cual permitirá, conectado a Factory I/O, controlar la simulación de la línea de producción diseñada por ordenador. CODESYS Group (anteriormente CODESYS GmbH o 3S-Smart Software Solutions GmbH) es una empresa alemana fundada en 1994 y líder en el sector de la programación de PLC. Siendo CODESYS el programa más popular de la compañía, no es ni mucho menos el único ya que la empresa ofrece gran cantidad de herramientas adicionales (gratuitas o de pago) orientadas al área de la automatización.

1.2. Objetivos

El objetivo principal de este Trabajo Final de Grado es diseñar de forma virtual y en 3D un proceso de paletizado similar al que podría ocurrir en una industria, realizar la programación del control automático del proceso de forma que funcione autónomamente, así como verificar que la simulación se lleva a cabo correctamente, satisfaciendo los requisitos que tendría el proceso real simulado.

El proceso escogido va a consistir en una línea de producción a la cuál llegan piezas metálicas y de plástico, las cuales se desea depositar en pallets separados. Para ello serán clasificadas previamente, y dos paletizadores independientes serán los encargados de colocarlas ordenadamente en un pallet. Posteriormente a su paletizado, ambas líneas volverán a converger en una sola, llevando los pallets completos hacia una zona de salida.

Para que el trabajo se lleve a cabo de forma satisfactoria, el proceso debe garantizar la seguridad de los trabajadores y trabajar a prueba de errores, lo cual implicará el uso de una serie de sensores y elementos especiales colocados estratégicamente a lo largo de la línea. La simulación deberá funcionar correctamente tanto en régimen estacionario (durante un periodo largo de tiempo) como al producirse algún evento de emergencia, siendo capaz de retomar la marcha por donde se encontraba, una vez resuelto el problema.

Sumado a lo anterior, se debe estudiar la viabilidad tanto técnica (en términos de hardware y software necesario) como económica del proyecto, justificando que el coste de su desarrollo virtual supone un ahorro posterior a la hora de la puesta en marcha en planta y que, por tanto, podría ser rentable el uso de este sistema por parte de empresas de este sector.

1.3. Alcance

Este proyecto abarca tanto el diseño del gemelo digital de un proceso de paletizado, como el desarrollo y la programación del sistema de control en CODESYS, así como la posterior verificación del correcto funcionamiento de dicho proceso a través de una simulación que utilizará el PC como PLC Virtual. Asimismo, se realizará un estudio acerca de la viabilidad técnica y económica del proyecto en comparación con el método tradicional de puesta a punto en fábrica.

Quedan fuera del alcance del trabajo el diseño del proceso real, en ninguno de sus ámbitos, ni mecánico ni eléctrico.

Este trabajo, desarrollado de forma independiente, puede servir como modelo para que las empresas puedan utilizar este software para testear los controles automáticos de sus líneas de producción sin necesidad de tener la máquina o línea en cuestión montada en planta, de modo que se puede trabajar de forma paralela para cuando esté montada el trabajo sea mucho menor.

Este tipo de herramientas, no existían o eran muy básicas hasta hace muy poco, pero con Factory I/O la creación de gemelos digitales de procesos industriales pretende volverse viable y efectiva en términos de comparativa con la realidad.

A menor escala, el uso de este software puede aplicarse también al ámbito didáctico, de forma que los alumnos de automatización puedan trabajar con gran variedad de procesos sin necesidad de maquetas y acercándose a su funcionamiento real gracias a la simulación en 3D de los mismos.

1.4. Normas y Referencias

1.4.1. Disposiciones legales y normas aplicadas

- **IEC 61131-3:** Parte 3 del estándar de programación para PLC, en el cual se basa CODESYS, aceptado a nivel internacional. Su edición actual data del 2013 y define los estándares de los siguientes lenguajes de programación: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), Sequential Function Chart (SFC).
- **UNE-EN IEC 62541:2020:** Norma que determina las condiciones en que se debe dar la comunicación del tipo OPC UA. Consta de 15 partes.
- **UNE-EN 60848:2022:** Lenguaje de especificación GRAFCET para diagramas funcionales en secuencia.

1.4.2. Programas utilizados

- **Factory I/O:** Diseño y simulación de la línea de paletizado.
- **CODESYS V3.5 SP16 Patch 2:** Programación del control automático de la línea.
- **CODESYS Control Win SysTray:** Uso de ordenador como PLC Virtual.
- **OFT2 Omegon Fluid Technology Grafcet:** Dibujo de Grafkets que describen el proceso.
- **Microsoft Office Word:** Procesador de Textos, utilizado para la redacción del presente documento.
- **GIMP 2:** Editor de imágenes de licencia libre, utilizado para retocar algunas figuras presentes en este TFG.



Figura 1: Logotipo de Factory I/O

1.5. Definiciones y abreviaturas

- **IEC:** *International Electrotechnical Commission* (Comisión Electrotécnica Internacional en español) es una organización encargada de la normalización en los campos: eléctrico, electrónico y otras tecnologías relacionadas.



Figura 2: Logotipo de la IEC

- **PLC:** *Programmable Logic Controller* (Controlador Lógico Programable o, simplemente, Autómata Programable, en español) es un ordenador industrial diseñado para trabajar en condiciones difíciles (vibración, temperatura, suciedad...) y adaptado para el control de procesos de producción, como líneas de ensamblaje, máquinas, robots u otras actividades que requieran alta fiabilidad.
- **POU:** *Program Organization Units* (Unidades de Organización de Programa en español) es un acrónimo que sirve para englobar programas (PRG), funciones (FUN) y bloques de funciones (FB) siguiendo el estándar IEC 61131-3, es decir, los módulos de los que puede componerse un programa completo.
- **Ladder Diagram (LD):** Diagrama en escalera o de contactos. Es un lenguaje de programación gráfico muy utilizado en la programación de PLC incluido en el estándar IEC 61131-3 y que emula los diagramas de circuitos basados en bobinas.
- **Grafcet:** *Graphe Fonctionnel de Commande Etape Transition* (Gráfico Funcional de Control de Etapas y Transiciones en español) es un modelo de representación gráfica y visual que describe la evolución de un sistema lógico, es decir, de un proceso automatizado.
- **Etapas:** Definen cada uno de los estados que puede adoptar un proceso a lo largo del tiempo
- **Transición:** Condición que debe cumplirse para avanzar entre etapas.

- **Inputs:** Señales de entrada que recibe el PLC desde el proceso a controlar. Suelen provenir de sensores de distintos tipos y determinan las transiciones de una etapa a otra o la activación de señales.
- **Outputs:** Señales de salida, enviadas desde el PLC a los distintos actuadores del proceso. Suelen ir asociados a etapas y pueden
- **Acción Impulsional:** Es aquella que se realiza una única vez tras la activación de una etapa. Suele asociarse a contadores o asignación de valores.
- **Acción de Nivel:** Es aquella que se mantiene activa mientras la etapa que la acciona también lo está.
- **Forzados entre Graficets:** Mecanismo que permite controlar la evolución de un Graficet esclavo a partir del estado de un Graficet maestro.
- **TIPOS DE DATOS:** Las variables utilizadas en la programación del control automático pueden ser de diferente tipo, según el tipo de valores que pueden tomar.
 - **BOOL:** Variables booleanas (1 bit). Pueden tomar el valor TRUE (verdadero) o FALSE (falso). Representan una amplia mayoría ya que incluyen los sensores y actuadores del sistema a controlar.
 - **INTEGRER:** Variables enteras (8 bits). Pueden tomar valores enteros positivos y negativos. Se utilizarán para realizar convertir variables reales en la realización de cálculos que el software no permite hacer con este tipo de variables.
 - **REAL:** Variables reales (32 bits). Pueden tomar valores decimales, su rango se determina por el estándar IEC 559. En otras aplicaciones se les llama *float* o números en coma flotante. Su utilidad en este proyecto será muy grande, puesto que de este tipo serán los sensores y actuadores de posición de los paletizadores.
 - **WORD:** Cadena de variables booleanas de 16 bits que puede tomar valores desde 0 hasta FFFF en hexadecimal. Se utiliza para la activación y desactivación de etapas puesto que se puede acceder a cada bit individualmente mediante la expresión *variable.bit* .
- De la misma forma, hay ciertas variables que no representan un tipo de dato como tal, sino que expresan la función que realizan en el programa y vienen determinadas por el bloque de función (FB) utilizado.



- **R_TRIG y F_TRIG:** Respectivamente, recibirán esta clasificación las variables asignadas a los bloques de función encargados de detectar flancos de subida o de bajada de ciertas variables, es decir, los momentos de activación o desactivación de las mismas. Para posteriormente utilizar estos flancos en el código, se deberá llamar como *variable.Q* .
- **TON:** Reciben esta nomenclatura las variables asignadas a los bloques de que hacen la función de temporizadores de activación. De forma similar, para llamar a la variable temporizador, se puede hacer con *variable.Q* .

En muchos casos, aparecerán secuencias redundantes relativas a ambos paletizadores, para evitar una repetición excesiva, se llamará PM y PN a los paletizadores (de piezas metálicas y normales o de plástico, respectivamente) así como a las líneas tanto de cintas como de rodillos (LM y LN, respectivamente). Del mismo modo, la línea de rodillos de salida podrá ser abreviada como Línea S.

Las abreviaturas anteriores, aparecerán sobre todo a la hora de asignar nombre a diferentes sensores y actuadores relativos a cada una de estas líneas, para evitar nombres excesivamente largos, y se hará referencia a ellos en este documento.

De la misma forma, como se verá más adelante, en los Graficets se utilizará la nomenclatura GX.EY para referirse a la Etapa Y del Graficet X.

1.6. Problemática

En primer lugar, se debe determinar el proceso a diseñar, puesto que no se trata de un caso concreto o de alguna petición de una empresa, pero se pretende que éste represente fielmente un proceso que podría darse en una industria real. Además, el diseño debe adaptarse a las posibilidades que ofrece el programa Factory I/O que, si bien son muy amplias, cuentan con alguna que otra limitación. En la Figura 3, se puede observar desde una vista alejada, es espacio de trabajo que ofrece Factory I/O.



Figura 3: Espacio de trabajo en Factory I/O (vista general)

Para ello, se ha contado con la experiencia obtenida durante la estancia en prácticas de la universidad, que en mi caso se ha dado en la empresa System España, empresa dedicada a la maquinaria cerámica (diseño en su sede en Italia, instalación, mantenimiento y reparación), máquinas entre las cuáles se encuentran paletizadores de funcionamiento similar al que ofrece el software Factory I/O. Asimismo, durante la estancia en la empresa, se han podido visitar diferentes plantas reales para conocer más a fondo la maquinaria y vivir de primera mano diferentes tipos de procesos industriales.

Tras esto, el siguiente paso es escoger el método de programación, entre los que ofrece el software CODESYS, y que esté recogido en el estándar IEC 61131-3. Las posibilidades incluyen lenguajes de programación gráficos (caso del *Ladder Diagram*, el *Function Block Diagram* y el *Sequential Function Chart*) y lenguajes de texto (*Structured Text* y *Instruction List*), aunque varios de ellos pueden ser combinados dentro de un mismo programa.

Relacionado con lo anterior, la tercera tarea consiste en exportar las variables de entrada o inputs (sensores) y las variables de salida o outputs (actuadores) del diseño realizado al proyecto de CODESYS. Se trata de un trabajo algo tedioso, sobre todo cuando se trabaja con gran cantidad de señales, puesto que se debe hacer de forma manual.

Por suerte, Factory I/O en su pestaña Drivers, ofrece una lista de los sensores y actuadores presentes en el proyecto (se puede ver un recorte de la misma en la Figura 4), por lo que se facilita la introducción de estos en CODESYS sin dejarse ninguno. Se debe tomar en cuenta que estas variables se deben incluir en una lista de variables global, para que más adelante se puedan compartir de nuevo con Factory I/O a la hora de realizar la simulación.

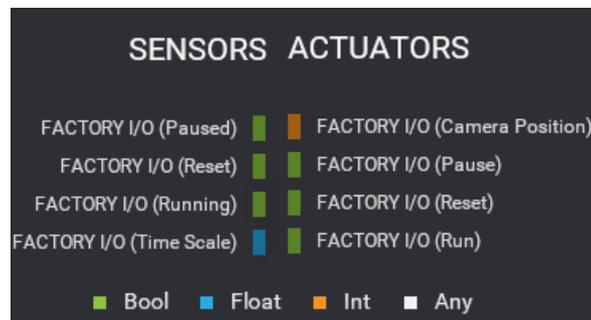


Figura 4: Recorte de la pestaña Drivers de Factory I/O

Una vez completadas las tareas anteriores, lo siguiente es escribir el código de control. Para ello, puede ser necesario definir nuevas variables locales dentro del POU de programa, ya sea de un tipo de dato concreto (BOOL, INT, REAL) o instancias de Bloques de Función (R_TRIG, F_TRIG, TON).

Este paso representará una porción importante del trabajo ya que, una vez terminada la programación, siempre pueden aparecer situaciones durante la simulación que precisen añadir o modificar parte del programa. Por lo tanto, esta etapa no es de paso único, sino que por ella se pasará cada vez que se quiera realizar una modificación para corregir un error visto en la simulación.

No obstante, hay un proceso imprescindible que se debe realizar antes de poder realizar la simulación y que consiste en conectar CODESYS con Factory I/O. Es un proceso que requiere de varios pasos y que se puede realizar de varias formas, según facilita la documentación de Factory I/O en su web:

- Conexión mediante protocolo OPC DA (*Data Access*)
- Conexión mediante protocolo OPC UA (*Unified Architecture*)
- Conexión mediante protocolo Modbus TCP

Más adelante, se debatirá sobre el protocolo escogido, así como sus ventajas y desventajas respecto a los demás.

1.7. Análisis de soluciones

1.7.1. Diseño del proceso

1.7.1.1. Descripción detallada del proceso

El proceso va a consistir en una línea de producción a la que llegan, de forma azarosa y en un intervalo de entre 1 y 2 segundos (también aleatorio), piezas de dos tipos: plásticas y metálicas, como se puede apreciar en la Figura 5.

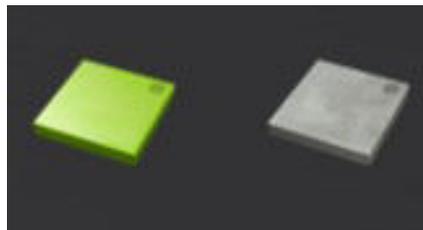


Figura 5: Pieza de plástico (derecha) y metálica (izquierda)

Dicha línea se subdivide un poco más adelante en dos tramos, uno recto por el que, en funcionamiento normal, seguirán las piezas de plástico, y un desvío a 90° al que serán empujadas las piezas metálicas. La Figura 6 muestra el proceso en cuestión.



Figura 6: Entrada de Piezas y Empujador.

A partir de aquí, las piezas pasan a ser tratadas individualmente, las de cada tipo en un paletizador (Figura 7) que las colocará de forma ordenada hasta formar un total de 10 piezas por pallet (este número puede modificarse con una ligera variación del código de programación que se indicará en su momento).



Figura 7: Paletizador (vista general)

Terminado el proceso de paletizado, los pallets avanzan hasta el punto mostrado en la Figura 8, que será llamado Máquina Rotativa, a modo descriptivo de su funcionamiento. Allí, si la llegada de pallets se produce simultáneamente, tendrán preferencia los que vengan de derecha a izquierda según la Figura 7, puesto que ese es el estado de reposo de la máquina. En caso de llegar un pallet procedente de la línea frontal a la imagen, la máquina entrará en funcionamiento para dar paso al pallet y redirigirlo hacia la salida.



Figura 8: Máquina Rotativa

Pasado este punto, los pallets se dirigen al punto de salida (Figura 9), donde se sitúa un elemento propio de Factory I/O cuya función es eliminarlos, que en la realidad podría corresponder a un operario con un toro mecánico que los sitúe en un almacén o bien a continuar la producción en otra máquina (flejadora, embaladora, etc.) o proceso de la fábrica.



Figura 9: Tramo de salida

Una vez definido la base del proceso en sí, cabe añadir un par de detalles al funcionamiento global del mismo.

En primer lugar, la marcha, paro y parada de emergencia del proceso (con su correspondiente alarma y luz de emergencia) y de ambos paletizadores individualmente se controla desde la zona central del mismo (Figura 10), aprovechando la geometría que se da al realizar el paletizado de piezas de plástico y metálicas por separado y que esa zona permite al operario tener una vista general del proceso con fácil control visual de ambos paletizadores, la entrada de piezas, y la Máquina Rotativa.

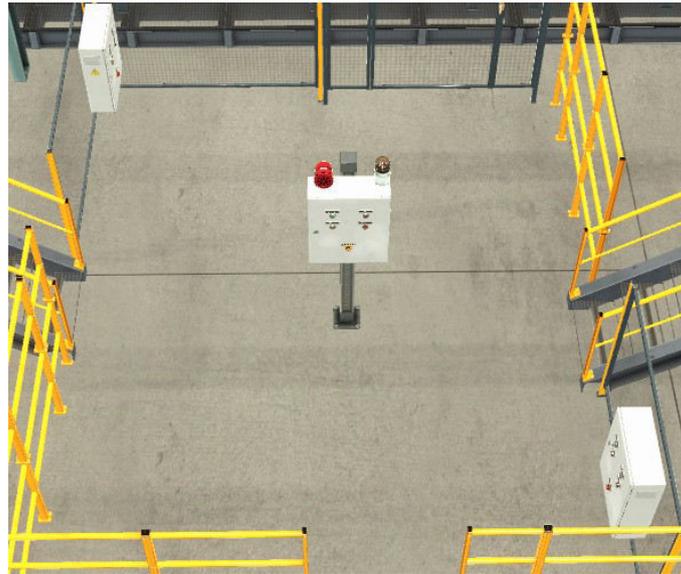


Figura 10: Zona de control del proceso en el centro de la línea

Por último, es importante remarcar que toda la zona de producción (Figura 11) está rodeada de un perímetro de seguridad formado por vallas que impide el acceso del personal, garantizando su seguridad. Se ha optado por esta opción y no por otra como podrían ser barreras láser (para las máquinas) o líneas de vida (caso de cintas y rodillos) puesto que las barreras láser de las que dispone el software son de corto alcance y se requeriría de muchas de ellas para cubrir la zona necesaria. Respecto a la segunda opción, el programa no dispone de la opción de añadir líneas de vida a las cintas y rodillos.



Figura 11: Vista general de la zona de producción con perímetro de seguridad

No obstante, se han habilitado un total de 5 puertas de seguridad, para que los operarios puedan acceder a cualquier parte de la línea en caso de emergencia o para realizar labores de mantenimiento, reparación o limpieza.

De esta forma, queda completamente definido el proceso sobre el que se va a trabajar, las características y partes del mismo, así como los elementos de seguridad que lo componen.

1.7.1.2. Proceso de ensamblado de las partes

Factory I/O clasifica los elementos que se pueden añadir a una escena en las categorías de Emisor, Extractor, Elementos de Carga Ligera, Elementos de Carga Pesada, Sensores, Operadores, Estaciones, Dispositivos de Advertencia y Pasarelas. En los párrafos siguientes se va a describir los elementos utilizados de cada tipo, así como su función específica, no obstante, en el ANEXO I se puede encontrar información más detallada acerca de cada uno de los elementos utilizados, así como la simbología empleada por el programa para representar cada uno de ellos.

- **Emisores:** Se trata de partes que introducen algunos tipos concretos de elementos a la escena durante la simulación, y se pueden configurar según convenga. Los utilizados en este proyecto son:
 - 1 emisor de piezas, colocado al inicio de la cinta de entrada y configurado para que emita aleatoriamente piezas verdes (plástico) y piezas metálicas, en un intervalo entre 1 y 2 segundos entre ellas.
 - 2 emisores de pallets, colocados en el inicio de los rodillos que llevan a cada uno de los paletizadores. Emiten pallets cuadrados a una frecuencia de, al menos, 10 segundos (puesto que el tiempo que tarda en llenar el pallet el paletizador es mayor, esta frecuencia no es relevante).
- **Extractor:** Se trata de una parte que se encarga, como su propio nombre indica, de eliminar las piezas de la escena. En este proyecto solamente se utilizará uno de estos elementos, colocado al final de los rodillos de salida.
- **Elementos de carga ligera:** Por ellos transcurrirán las piezas previamente a su paletizado. Factory I/O dispone de cintas de 2, 4 y 6m así como diversos accesorios para poder realizar los caminos deseados. Para este proyecto se han utilizado las siguientes partes:
 - 1 cinta de 6m será utilizada para la sección de entrada de piezas.
 - 4 cintas de 2m serán empleadas en la entrada de la zona para piezas de plástico/metálicas y en la aproximación a los paletizadores.

- 2 cintas de 4m se utilizarán en la parte intermedia de cada una de las zonas (plástico/metálicas)
- 13 alineadores de tipo 3, colocados 6 de ellos en cada una de las aproximaciones a los paletizadores y uno extra en la cinta de entrada. Su función principal es asegurar la correcta posición de llegada de las piezas a los paletizadores.
- 1 alineador de tipo 2, colocado junto al de tipo 3 en la cinta de entrada para orientar las piezas hacia un lateral de la cinta.
- 1 empujador, colocado a continuación de los alineadores de la entrada, encargado de enviar las piezas metálicas hacia el desvío correspondiente.
- 4 hojas de parada, encargadas de impedir el avance de las piezas hacia ciertos puntos. 2 de ellas irán colocadas justo al final de cada una de las cintas de entrada al paletizador y las otras dos entre este tramo y el anterior.
- 7 esquinas de metal, colocadas en un lateral de las cintas y cuya función es la de colocación de sensores sobre las mismas. En ambos paletizadores, una estará colocada justo al final de la línea, la segunda a la entrada de la cinta de aproximación y la tercera a la entrada de la cinta intermedia. La última irá colocada junto al empujador.
- **Elementos de Carga Pesada:** Por ellos discurrirán los pallets, tanto vacíos como completos. De nuevo, Factory I/O dispone de rodillos de 2, 4 y 6m así como de diversos elementos extra que aumentan las posibilidades de ejecución. Estas son las partes utilizadas en el proyecto:
 - 4 transportadores de rodillos de 2m, utilizados dos de ellos en las aproximaciones a la Máquina Rotativa (uno en el tramo de cada paletizador) y los otros dos en la zona de salida (uno el último tramo y otro en la aproximación a la curva).
 - 7 transportadores de rodillos de 6m, utilizados 4 de ellos en la entrada y salida de cada paletizador (2 en cada uno) y los otros tres en la salida de la Máquina Rotativa y en el resto de tramo de salida.
 - 1 transportador de rodillos curvo, utilizado en el tramo de salida.
 - 2 dispositivos de paro de rodillos, colocados en ambas entradas a la Máquina Rotativa, que se encargarán de parar los pallets procedentes

en caso de que lleguen simultáneamente o que se requiera un giro de la máquina para su entrada.

- 1 máquina giratoria, situada en el punto intermedio de las salidas de ambos paletizadores. Su función será redirigir los pallets hacia los transportadores de salida.
- **Sensores:** Son los encargados de detectar la presencia de piezas, pallets, etc. No se incluyen en esta lista aquellos sensores propios de alguna máquina en concreto, puesto que serán analizados más tarde, solamente aquellos introducidos manualmente. Factory I/O dispone de muchos tipos de sensores, los utilizados en este montaje han sido:
 - 4 sensores capacitivos, cuya función es la de detectar pallets en dos puntos específicos y simétricos en ambos paletizadores (2 sensores para cada línea). El primero se colocará en la posición en que se desea que el pallet pare al llegar al paletizador, de forma que quede lo más centrado posible respecto a la cinta de llegada de piezas. El segundo irá colocado poco antes de la máquina rotativa, y su función será la de detectar la llegada de los pallets a dicho elemento y dar prioridad en función de la llegada de estos.
 - 6 sensores difusos, utilizados en la detección de piezas de ambos tipos y preferidos sobre los capacitivos, en este caso, debido a su mayor rango. Serán colocados en la entrada de la cinta intermedia de cada línea (para detectar si dicha línea se llena), al inicio de la cinta de aproximación al paletizador (para evitar un exceso de piezas acumuladas en ese tramo) y en la posición de llegada al punto de agarre de las piezas por parte del paletizador (para que el proceso de paletizado de comienzo).
 - 1 sensor inductivo, colocado justo antes del empujador y cuya función es detectar exclusivamente las piezas metálicas para poner en marcha el pistón que las lleva hacia su línea.
- **Operadores:** Botones, pulsadores, interruptores e indicadores luminosos que se utilizan para controlar el estado del proceso. En el presente proyecto existen 3 cuadros eléctricos de control, siendo uno general y los otros dos dedicados a cada uno de los paletizadores. Por lo tanto, los elementos utilizados han sido los siguientes:
 - 1 columna colocada en el centro de la zona de control.

- 3 cuadros eléctricos, colocado el general apoyado sobre la columna en el centro de la zona de control y los correspondientes a cada paletizador en la valla de seguridad más próxima a cada uno de ellos.
 - 3 interruptores de parada de emergencia (con enclavamiento), uno en cada cuadro. Se encargan de parar de inmediato ya sea el proceso general o el trabajo de uno de los paletizadores.
 - 3 pulsadores verdes, uno en cada cuadro, que se utilizarán para el arranque de los paletizadores o del resto de elementos del proceso.
 - 1 pulsador amarillo, colocado en el cuadro general y que permite silenciar la alarma de emergencia.
 - 3 pulsadores rojos, uno en cada cuadro, encargados de solicitar la parada normal de los paletizadores o del proceso en general.
 - 2 indicadores luminosos amarillos, cada uno en el cuadro de un paletizador, que se utilizarán para notificar el estado de emergencia de cada uno de los paletizadores.
 - 3 indicadores luminosos rojos, uno en cada cuadro, que determinarán el estado completamente parado del proceso general o de cada uno de los paletizadores.
- **Estaciones:** Factory I/O dispone de ciertas máquinas predefinidas (tanques, soluciones de almacenamiento...) que permiten realizar operaciones sobre las piezas u otros elementos como cajas o pallets. En este caso, tratándose de un proceso de paletizado, el software ofrece dos opciones, una máquina de paletizado de cajas y un paletizador de tipo *Pick & Place* (Coge y Coloca). Siendo esta última la que ofrece una mayor versatilidad, y que por lo tanto puede adaptarse más fácilmente a ciertos cambio, ha sido la escogida para realizar este proyecto:
 - 2 Paletizadores Pick & Place, colocados al final de las cintas y a través de los cuales pasan los transportadores de rodillos que portan los pallets.
- **Dispositivos de Advertencia:** Elementos encargados de notificar de ciertos eventos (usualmente marcha, emergencia o parado) a través de señales visuales o sonoras. En el proyecto se han utilizado:
 - 1 sirena de alarma colocada sobre el cuadro general, que se activa en caso de emergencia.

- 1 luz de advertencia colocada junto a la sirena de alarma y que se activa también en caso de emergencia o al abrir una de las puertas de seguridad.
- 2 *Stack Light* (luces apiladas) colocadas sobre el marco de ambos paletizadores para indicar su estado (parado, marcha, emergencia...)
- **Pasarelas:** Se incluyen elementos que permiten crear lugares de paso para los operarios así como otros para crear perímetros de seguridad.
 - Pasamanos: Su uso principal es el de proporcionar seguridad al operario en cuando se encuentra en plataformas en altura. No obstante, en este proyecto se han utilizado para cerrar ciertas partes del perímetro de seguridad ya que las vallas metálicas de protección son de tamaño único y no lo permitían (esto no sucedería en la realidad, puesto que se pueden obtener diferente tamaño de vallas, pero se trata de un aspecto secundario de la simulación, por lo que no se le da mayor importancia. Los pasamanos están disponibles en varios tamaños y pueden combinarse para obtener estructuras más complejas. Se han utilizado un total de 38 pasamanos de tamaño S, y 28 de tamaño M.
 - Plataformas: Permiten la creación de suelos a la altura deseada. El uso de estas en este caso ha sido reutilizado para aumentar la altura de los transportadores de rodillos creando dos carriles uniendo plataformas de tamaño S (unas 110 en total) sobre los que se apoyan las patas de dichos elementos, con el fin de acercar los rodillos, y por tanto los pallets al brazo del paletizador. Adicionalmente, se han utilizado 4 plataformas de tamaño M para crear la zona llana en las escaleras que permiten el paso sobre las líneas (ver Figura 12).
 - Escaleras: Compuestas de los propios escalones más la barandilla, en este caso se añade una plataforma intermedia que permite comunicar dos zonas de la línea de producción, como puede observarse en la Figura 11. Se incluyen un total de 4 conjuntos de escalera (8 escalones y 8 barandillas).



Figura 12: Paso de Escalera montado

- Vallas de Seguridad: Utilizadas para crear perímetros de seguridad alrededor de líneas y máquinas. Factory I/O dispone de 3 tipos pequeña (S), grande (L) y con hueco para línea (I). Ubicadas alrededor de toda la línea de producción, se han utilizado un total de 32 de tamaño S, 5 de tamaño L (en la zona cercana a los paletizadores) y 4 de tipo I, en las dos entradas de pallets y en la de piezas.
- Puertas de Seguridad: Se reparten un total de 5 a lo largo de la línea de producción, permitiendo así el acceso seguro a las diferentes partes del proceso en caso de necesidad de realizar mantenimiento o reparación. Su ubicación se puede ver en la Figura 11 (página 17).

Tras las indicaciones realizadas, el montaje del proceso quedaría finalizado y la planta lista para funcionar a falta de realizar la programación de sistema de control.

1.7.2. Variables Globales

Las variables globales son aquellas que aparecen de forma directa en los elementos del proyecto, es decir, los sensores y actuadores de todos los componentes del proceso. Deben definirse en una lista global (que permita el acceso externo) en el programa donde se va a realizar la programación del control automático, puesto que no solo deben ser leídas y/o escritas por el software de control, sino que también por el de simulación. En este caso, en CODESYS se ha llamado FIO a dicha lista como abreviatura de Factory I/O.

Puesto que se trata de una lista muy extensa, en el ANEXO II se proveen todo tipo de detalles (tipo de variable, nombre asignado, descripción, etc.) acerca de las mismas, ya sea en la Tabla 5 si estamos hablando de sensores o en la Tabla 6 para los actuadores. No obstante, seguidamente se ofrece un listado con los grupos que forman estas variables:

- **SENSORES – INPUTS:**

- Cuadro general (pulsadores marcha, paro, emergencia...)
- Empujador (sensor piezas metálicas y finales de carrera)
- Cuadro de los Paletizadores (pulsadores marcha, paro...)
- Paletizadores (sensores de posición, detección de pieza...)
- Máquina rotativa (sensores a la entrada, posición correcta...)
- Otros sensores (puertas de seguridad y sensores de línea llena)

- **ACTUADORES – OUTPUTS:**

- Cuadro general (luces indicadores marcha, paro, emergencia...)
- Empujador (pistón de empuje)
- Cuadro de los Paletizadores (luces de marcha, paro y emergencia)
- Motores Cintas (entrada, línea piezas metálicas, línea piezas plástico...)
- Máquina rotativa (giro, rodillos de espera, rodillos rotativa)
- Paletizadores (agarre de piezas, luces estado, motores de posición...)
- Motores Rodillos (línea piezas metálicas, línea piezas plástico, salida)
- Emisores y Extractores de piezas

En total, se dispone de 105 variables globales, siendo 42 de ellas inputs y 63 outputs, a las cuales habrá que añadir, las variables internas del código de control, que serán mencionadas unos puntos más adelante.

1.7.3. Método de programación

El primer paso a la hora de decidir el método de programación es tener claro cómo han de funcionar por separado cada una de las partes del proceso, así como la interacción que pueda haber entre las mismas, para ello, es muy útil la elaboración de Graficets, puesto que son herramientas que permiten tener de forma rápida una concepción muy visual del funcionamiento del proceso.

El control de un proceso simple puede plasmarse en un solo Graficet, pero un proceso con mayor complejidad como con el que se va a trabajar requiere una

división por partes y Grafquets que se relacionen entre sí. Para ello, se ha separado en 5 subprocesos (con sus correspondientes Grafquets) el funcionamiento de toda la línea de producción. A continuación, se explica cada uno de ellos individualmente:

NOTA: De ahora en adelante, tanto en los Grafquet como posteriormente en el código, para diferenciar entre los dos paletizadores, se les llamará Paletizador de piezas metálicas (PM) y Paletizador de piezas normales (PN) así como a las líneas (línea M y línea N).

- Grafquet 0 – Control del Cuadro General:** Define los eventos de marcha, paro y emergencia del proceso en su conjunto. Se trata del Grafquet maestro, a través del cual se controla la activación y desactivación de los demás. El ANEXO III recoge la imagen completa de este Grafquet, junto con otros detalles añadidos sobre el mismo a la imagen mostrada en la Figura 13. Se compone de 5 etapas que funcionan de la manera siguiente:

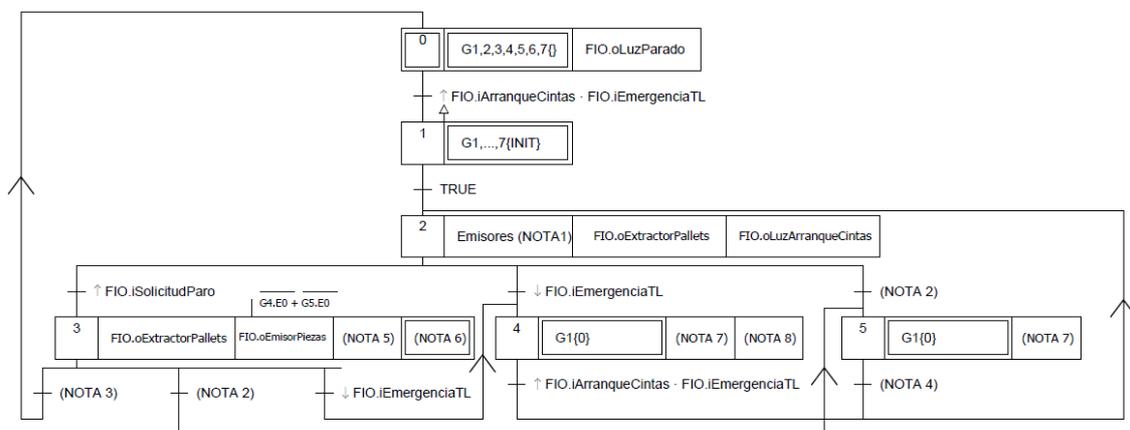


Figura 13: Grafquet 0 - Control del Cuadro General

- Etapa 0:** Representa la etapa de parado del proceso. Todos los Grafquets son forzados a desactivarse y la luz de parado se encuentra encendida. Se activa en la inicialización del control, o bien cuando se completa el paro tras esperar un tiempo específico a que todos los pallets se hayan eliminado.
- Etapa 1:** Se trata de una etapa inestable en la que se inician el resto de Grafquets. Se accede a ella con un pulso en el botón de marcha siempre que no haya una emergencia activa.
- Etapa 2:** Funcionamiento normal del proceso. Activa emisores de piezas y pallets y el extractor, así como la luz de arranque. Se puede llegar a ella desde las Etapas 4 y 5, siempre que la emergencia se haya resuelto y se dé un nuevo pulso de marcha.
- Etapa 3:** Etapa que se activa al solicitar un paro general del proceso estando en la Etapa 2. Se fuerzan los Grafquets de control de los

paletizadores (6 y 7) a su etapa de parada también y se mantienen activos el extractor y los emisores (estos últimos mientras ambos paletizadores no estén ya parados).

- **Etapa 4:** Se llega a esta etapa al presionar la seta de emergencia, ya sea desde la Etapa 2 o 3. Esta etapa desactiva la marcha de todos los transportes y mantiene el proceso en el estado en que estaba cuando se produjo la emergencia, así como activa la alarma y luz de emergencia. Para salir de la etapa, se debe restablecer el pulsador de emergencia y presionar de nuevo el botón de marcha.
- **Etapa 5:** A esta etapa se accede cuando se abre cualquiera de las puertas de seguridad. Actúa como la Etapa 4 con la excepción de que no se activa la alarma, solamente la luz de emergencia. Para volver a la marcha, todas las puertas deben estar cerradas y presionar el botón de marcha en el cuadro general.
- **Grafcet 1 – Control de Cintas y Rodillos:** Controla el funcionamiento de todas las cintas y rodillos de transporte de piezas y pallets. El Grafcet (Figura 14), se recoge también en el ANEXO IV con mayor detalle, se describen a continuación sus 11 etapas:

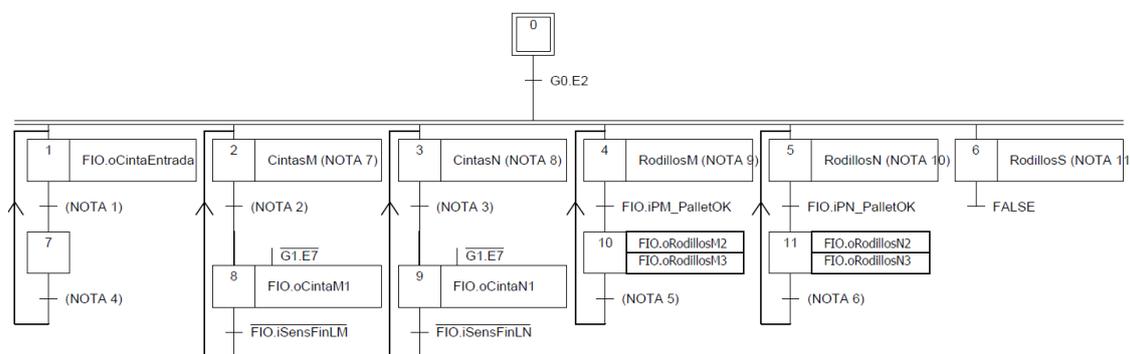


Figura 14: Grafcet 1 - Control de Cintas y Rodillos

- **Etapa 0:** Etapa de inicio del Grafcet. Todos los transportadores se encuentran parados. Solo se puede activar mediante forzado a través del Grafcet 0. Se desactiva cuando la Etapa 2 del Grafcet 0 se encuentra activa, y da paso a la activación de una secuencia de etapas 1—6.
- **Etapa 1:** Mientras esta etapa se encuentre activa, la cinta de entrada de piezas estará en movimiento.
- **Etapa 2:** Esta etapa activa los tres tramos de cintas de la línea M, condicionando el tercer tramo (el más próximo al paletizador) a que el Grafcet 4 no se encuentre en Etapa 0.

- **Etapa 3:** Mismo funcionamiento que la anterior, pero con la línea N, estando el tercer tramo condicionado a que no esté activa la Etapa 0 del Grafcet 5.
- **Etapa 4:** Durante esta etapa, los tres tramos de rodillos correspondientes a la línea M se encuentran en marcha.
- **Etapa 5:** Mismo funcionamiento que la Etapa 4 pero en la línea N.
- **Etapa 6:** Esta etapa activa los 5 tramos rectos de la salida de pallets, así como el tramo curvo. La transición para su desactivación está en falso, puesto que dichos tramos solo pararán en caso de forzado de este Grafcet, de lo contrario, siempre estarán en marcha.
- **Etapa 7:** Se activa a partir de la Etapa 1, cuando alguna de las dos líneas detecta que se encuentra llena y pasa un cierto tiempo. Es una etapa de espera, en la que la cinta de entrada se encuentra parada. Se vuelve a la Etapa 1 cuando ambas líneas ya no se encuentran llenas.
- **Etapa 8:** Se activa cuando en la Etapa 2 se produce una detección de pieza en el final de la línea M (con un pequeño temporizador de retardo). En esta etapa solo está activa la Cinta M1 (el primer tramo) y la misma está condicionada a que no se encuentre activa la línea 7. Se vuelve a la Etapa 2 cuando el sensor de final de línea deja de detectar.
- **Etapa 9:** Mismo funcionamiento que la Etapa 8 pero con los elementos de la línea N y sucesiva a la Etapa 3.
- **Etapa 10:** Esta etapa se activa cuando, estando activa la Etapa 4 el pallet llega a la posición correcta de paletizado en el Paletizador M. Se desactiva el tramo de rodillos inicial (M1), pero siguen en marcha los siguientes (M2 y M3). Se vuelve a la Etapa 3 cuando el pallet deja de estar en posición correcta y si, a la vez, el Grafcet 0 no se encuentra en la Etapa 3 (solicitud de paro).
- **Etapa 11:** Mismo funcionamiento que la Etapa 10 pero con los elementos de la línea N y sucesiva a la Etapa 5.
- **Grafcet 2 – Control de la Máquina Rotativa:** Este Grafcet describe el modus operandi de la máquina encargada de redirigir los pallets según vengan de la línea M o de la línea N hacia la línea de salida. El Grafcet y sus seis etapas (Figura 15), se puede consultar en mayor detalle en el ANEXO V, y su descripción se realiza seguidamente:

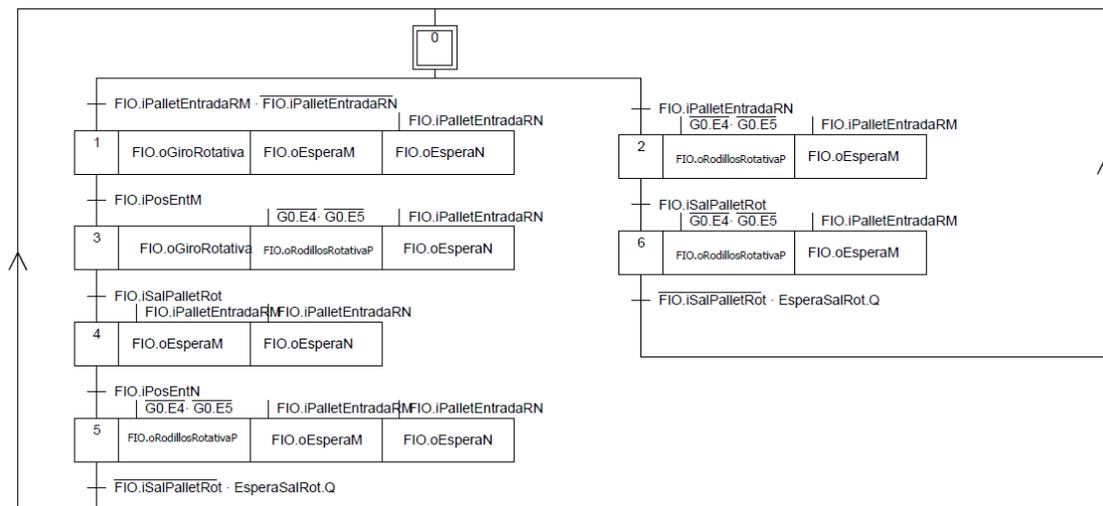


Figura 15: Grafcet 2 - Control de la Máquina Rotativa

- **Etapa 0:** Etapa de reposo, ningún actuador se encuentra activo. Se accede a ella en el forzado de inicio o tras las Etapas 5 y 6, en ambos casos cuando el sensor de Salida de la máquina deja de detectar y ha pasado un tiempo establecido.
- **Etapa 1:** Se accede a ella desde la Etapa 0, cuando se detecta un pallet en la entrada de la línea M y no se ha detectado en la línea N. Se produce el giro de la rotativa y se activa el rodillo de espera de la línea M y N, este último condicionado a la detección de un pallet proveniente de esta línea.
- **Etapa 2:** También accesible desde la Etapa 0, en este caso cuando se produce detección de pallet en la línea N. Durante esta etapa se activan los rodillos de la máquina en sentido positivo y, en caso de detectar un pallet en la línea M, se activa también el rodillo de espera de dicha línea.
- **Etapa 3:** Cuando estando en la Etapa 1, se alcanza la posición correcta para la entrada del pallet proveniente de PM, se activa esta etapa. En ella se mantiene activa la señal de giro de la máquina rotativa (de forma que no vuelva a su posición original) y se activan los rodillos de la máquina en sentido positivo. También se mantiene o se activa el rodillo de espera en LN si se detecta un pallet.
- **Etapa 4:** Esta etapa se activa a partir de la Etapa 3, cuando el pallet está correctamente colocado en la máquina y activa el sensor de Salida. Se desactiva la señal de giro, por lo que se vuelve a la posición de reposo y se activan o mantienen activos los rodillos de espera de detectar un pallet llegando de cualquiera de las dos líneas.

- **Etapa 5:** Alcanzada la posición correcta, se desactiva la Etapa 4 para dar paso a ésta. Los rodillos se activan de nuevo en sentido positivo y los rodillos de espera funcionan de igual manera.
- **Etapa 6:** En esta etapa, consecutiva a la Etapa 2, se mantienen los mismos actuadores, pero se accede a ella cuando el pallet proveniente de la Línea N activa el sensor de salida. Esta etapa tiene sentido para resetear el temporizador de espera de salida, que es el que permite volver a Etapa 0 desde 5 y 6.

NOTA: Los actuadores de los rodillos están condicionados a que el Grafcet 0 no se encuentre ni en la Etapa 4 ni 5 (etapas de emergencia).

- **Grafcet 3 – Control del Empujador de Piezas:** Con solo 3 etapas (Figura 16), es el Grafcet más sencillo de todos. Se puede ver en detalle en el ANEXO VI y su funcionamiento es el siguiente:

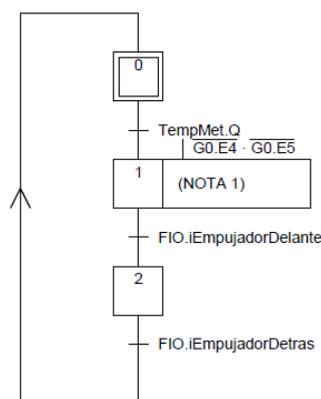


Figura 16: Grafcet 3 - Control del Empujador de Piezas

- **Etapa 0:** Posición de reposo. Se activa cuando se da la marcha general o cuando el empujador vuelve a su posición de reposo tras la Etapa 2.
- **Etapa 1:** Se activa con un temporizador que introduce un pequeño retraso desde la detección de la pieza por parte del sensor de piezas metálicas. Durante el transcurso de la misma se activa el pistón que empuja las piezas hacia la Línea de Metálicas. El funcionamiento del pistón está condicionado a que el Grafcet 0 no se encuentre en las etapas de emergencia 4 o 5.
- **Etapa 2:** Se activa cuando la etapa 1 hace llegar al pistón al final de carrera delantero, entonces, se desactiva el pistón hasta su vuelta a su posición de reposo.

- Grafkets 4 y 5 – Control de los Paletizadores M y N:** Dado que el funcionamiento de ambos paletizadores es idéntico, un solo Grafket (que posteriormente se deberá duplicar a la hora de la programación) basta para describir ambos procesos. En el ANEXO VII se muestra la imagen del Grafket, junto con varias notas aclarativas para su mejor comprensión. El comportamiento del Grafket (vista resumida en la Figura 17) en sus 5 etapas, es el siguiente:

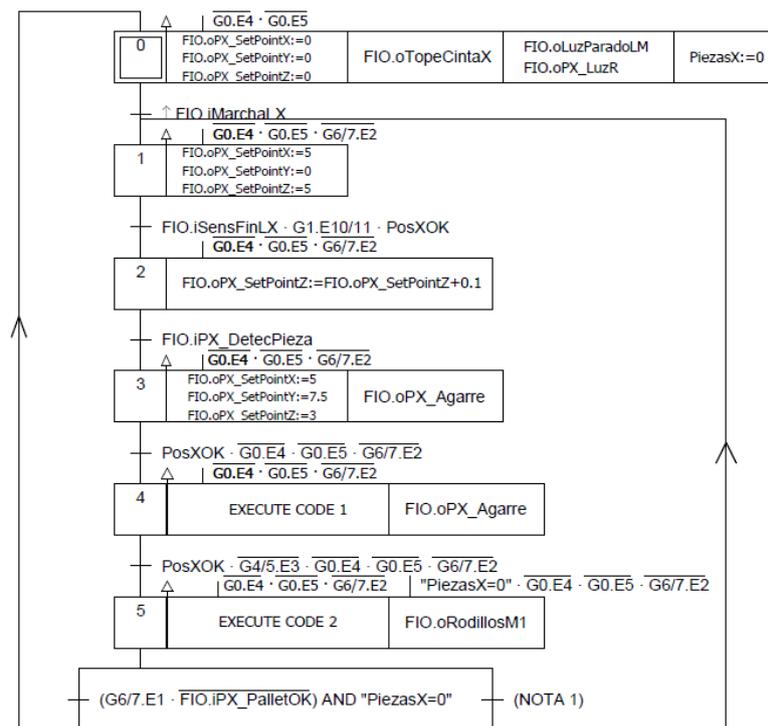


Figura 17: Grafkets 4 y 5 - Control de los Paletizadores M y N

- Etapa 0:** Se inicia al dar la marcha general y se puede volver a ella si se solicita un paro y este se termina correctamente tras la Etapa 5. En este estado, el paletizador se encuentra parado, con la pinza en posición de reposo XYZ (0,0,0). La hoja que impide el acceso de piezas a la zona de aproximación al paletizador está activo, el número de piezas se establece a 0 y se activan la luz de parado y la luz roja de estado de la máquina.
- Etapa 1:** Se activa tras un flanco en el pulsador de marcha y se puede volver a ella si, tras la Etapa 5, no se ha solicitado paro o si el pallet está incompleto. En esta etapa se lleva la pinza a posición de espera XYZ (5,0,5)
- Etapa 2:** Cuando la posición de espera es alcanzada, hay un pallet colocado, y una pieza es detectada por el detector de final de línea, se

activa la Etapa 2. En ella la pinza desciende progresivamente hasta que se realiza la detección de la pieza mientras que las ventosas de agarre empiezan a succionar.

- **Etapa 3:** Cuando se da la detección, la pieza es tomada por la pinza, que, manteniendo la succión activa, se mueve verticalmente hasta el punto XYZ (5,7.5,5).
- **Etapa 4:** Cuando se alcanza el punto mencionado, manteniendo la succión, se escoge un punto de destinación según el número de piezas que se hayan depositado ya en el pallet. Esto se realiza gracias a un código que deberá escribirse en lenguaje ST y que se puede consultar tanto junto al Grafcet en el ANEXO VII como al final del código de programación, situado en el ANEXO IX.
- **Etapa 5:** Llegada la pieza al punto de destino, se desactiva la succión, depositándose la pieza en el pallet. Si el pallet se ha completado con esta pieza, se activan los rodillos M1 para llevarlo hacia la salida y, simultáneamente, se resetea el contador de piezas a 0. Caso contrario, se suma 1 al contador de piezas. Tanto el reset a 0 como la suma se realiza con un pequeño código que aparece junto al mencionado en la etapa anterior.

NOTA: Varios actuadores y algunas transiciones están condicionadas a que el proceso no se encuentre en estado de emergencia. Se recomienda consultar el Grafcet en el ANEXO VII o el código de programación en el ANEXO IX para conocer los casos concretos.

- **Grafcet 6 y 7 – Control del Cuadro del Paletizador M y N:** Del mismo modo que con el funcionamiento del paletizador, el control del cuadro de mandos del paletizador es igual para ambos. Consta de tres etapas (Figura 18), y se puede consultar con mayor detalle en el ANEXO VIII. Funciona de la siguiente manera:

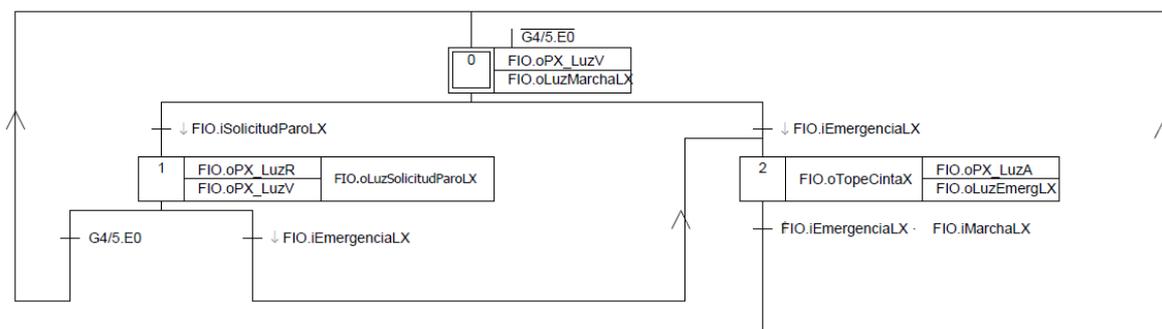


Figura 18: Grafcets 5 y 6 - Control del Cuadro de los Paletizadores M y N

- **Etapa 0:** Corresponde a la etapa de funcionamiento normal, que se activa con la marcha del proceso general. Con ella activa, y siempre que el paletizador no se encuentre en Etapa 0 también (parado), la luz de marcha en el cuadro y la verde en la lámpara de estado del paletizador se activan. Se vuelve a esta etapa tras un paro completo o tras resolver una emergencia y pulsar de nuevo el botón de marcha.
- **Etapa 1:** Si estando en la Etapa 0 se solicita la parada individual de un paletizador, se activa esta etapa. Se mantiene activa la luz verde de estado, puesto que el proceso sigue en marcha, pero a la vez se activa la roja indicando que se va a parar al terminar la producción actual. También se ilumina el botón de solicitud de paro.
- **Etapa 2:** Etapa de emergencia individual del paletizador, se accede a ella tras pulsar el botón de emergencia ya sea en funcionamiento normal (Etapa 0) o durante una solicitud de paro (Etapa 1). Esta etapa activa la hoja que impide acceder nuevas piezas a la zona de aproximación del paletizador y activa la luz amarilla de estado, así como el indicador también amarillo de emergencia en el cuadro.

De esta forma, quedan todos los subprocesos definidos. Y se puede escoger un método de programación de entre las alternativas posibles.

Inicialmente se descarta Instruction List por su bajo uso en la industria. De la misma manera, Structured Text, si bien será necesario en algún momento puntual, requiere demasiada transformación respecto a los Grafcets creados, por lo que también se desecha la opción de programar completamente en este lenguaje. De las opciones restantes, Function Block Diagram requeriría definir un gran número de funciones y el trabajo sería excesivo y muy costoso, por lo que las dos alternativas finales son Sequential Function Chart y Ladder Diagram.

Se ha optado por Ladder Diagram como la opción a usar, si bien Sequential Function Chart es un paso mucho más directo desde los Grafcet al software de programación ya que, si en un futuro se requiere modificar algún aspecto, o añadir alguna funcionalidad, LD es una mejor opción, puesto que añadir, eliminar o cambiar alguna de las líneas de código es mucho más sencillo. Además, es un lenguaje que muchos más software soportan, ya que se encuentra globalmente extendido, mientras que, aun siendo igualmente popular, SFC todavía está abriéndose paso. Finalmente, y esto puede considerarse una apreciación algo más personal, visualmente LD es mucho más intuitivo y ordenado que SFC, en el cual los elementos se encuentran mucho más dispersos.

1.7.4. Programación en CODESYS

En esta sección se van a explicar los puntos clave del código de programación en Ladder Diagram para el control automático de la línea de producción. El código completo puede consultarse en el ANEXO IX, donde se muestra de forma continua y ordenada, dividido en las diferentes secciones que lo conforman.

Antes de comenzar, es necesario remarcar que este proceso se ha llevado a cabo con el software CODESYS, que se puede descargar de forma gratuita desde la web de la compañía. Se ha optado por el uso de este software a diferencia de otras alternativas por tres principales motivos, el primero de los cuales ya ha sido mencionado, y es que se trata de un software de descarga gratuita. El segundo motivo radica en que CODESYS está desarrollado de forma completa bajo el estándar IEC 61131-3, lo cual lo hace compatible con la mayoría de PLC comerciales. Y la tercera y última razón es la posibilidad que ofrece CODESYS de utilizar el ordenador como PLC Virtual para comprobar el funcionamiento del código mediante la simulación en Factory I/O.

Ahora sí, para comenzar con la programación, en primer lugar, se deberá abrir el programa, en este caso la versión “V3.5 SP18 Patch 2”. A continuación, se deberá crear un nuevo proyecto, lo cual se consigue presionando el título *New Project...* que aparece en la pantalla de bienvenida o desplegando el menú *File* y haciendo clic en *New Project...*

Una vez realizado esto, se abrirá una ventana como la que se muestra en la Figura 19. Se deberá seleccionar la categoría de *Projects* en la columna de la izquierda, *Standard Project* en la derecha y, posteriormente, introducir un nombre y una ubicación en el disco duro del ordenador para el proyecto. Se recomienda crear una carpeta específica para el proyecto, puesto que CODESYS genera múltiples archivos sobre el mismo proyecto. Finalmente se presiona *OK*.

Tras esto, una nueva ventana se mostrará (Figura 20). En ella, debemos seleccionar la opción *CODESYS Control Win V3 (3S – Smart Software Solutions GmbH)* en el desplegable junto a *Device* así como *Ladder Logic Diagram (LD)* en el desplegable junto a *PLC_PRG in*. Importante en el primero de los dos desplegables **no** seleccionar la opción de igual nombre que la mencionada pero que incluye *x64* en su nombre, puesto que esto deshabilita una funcionalidad que se requiere más adelante. Una vez verificado esto, se pulsa *OK*.

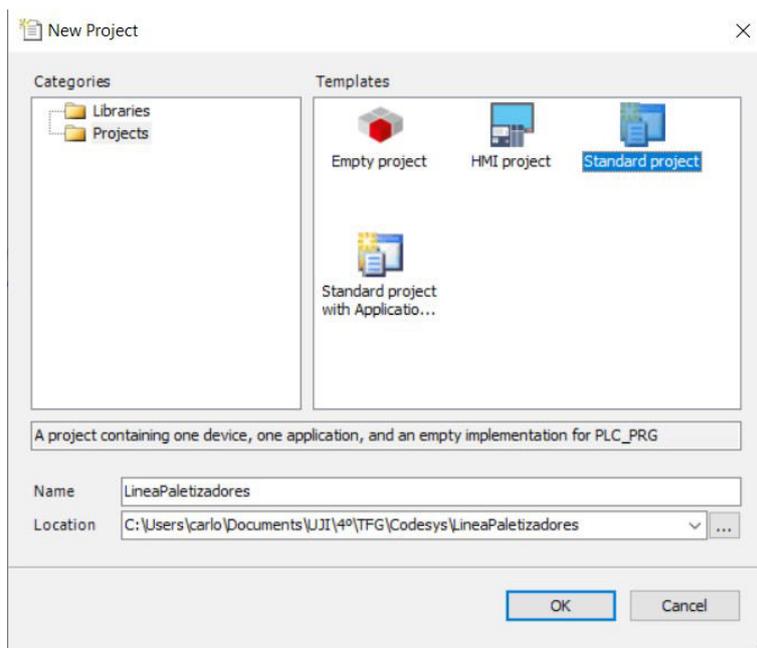


Figura 19: Ventana de creación de Nuevo Proyecto en CODESYS

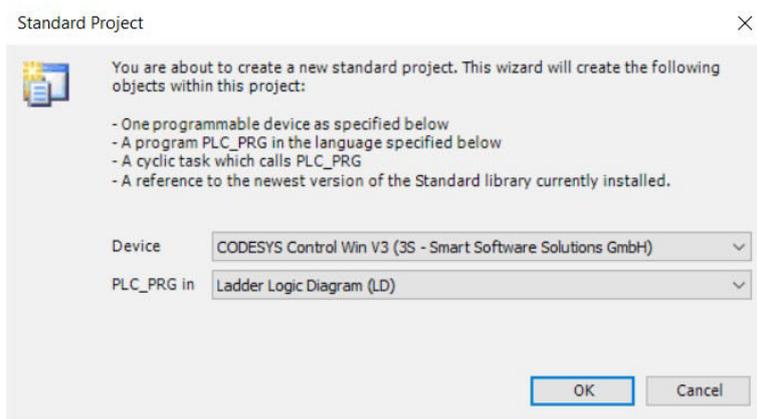


Figura 20: Pantalla de configuración del Proyecto Estándar

Con esto finaliza la configuración inicial del proyecto, y puede pasarse a realizar la programación. Tras cargar, CODESYS mostrará en la izquierda de la pantalla un árbol del proyecto similar al de la Figura 21.

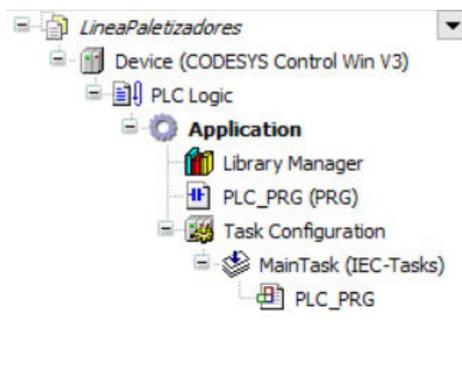


Figura 21: Árbol del proyecto en CODESYS tras la configuración inicial

El primer paso que se debe llevar a cabo es crear la lista de variables globales, que será llenada con todas las variables explicadas en el punto 1.7.2 y enunciadas en el ANEXO II, tanto inputs como outputs. Para ello, se realizará un clic derecho sobre el rótulo *Application* en el árbol del proyecto y se seleccionará *Add Object* → *Global Variable List* tal y como se muestra en la Figura 22. El programa pedirá nombrar la lista, en nombre escogido en este proyecto es FIO, pero puede usarse otro. Se recomienda que el nombre sea corto, puesto que será utilizado con asiduidad durante la programación.

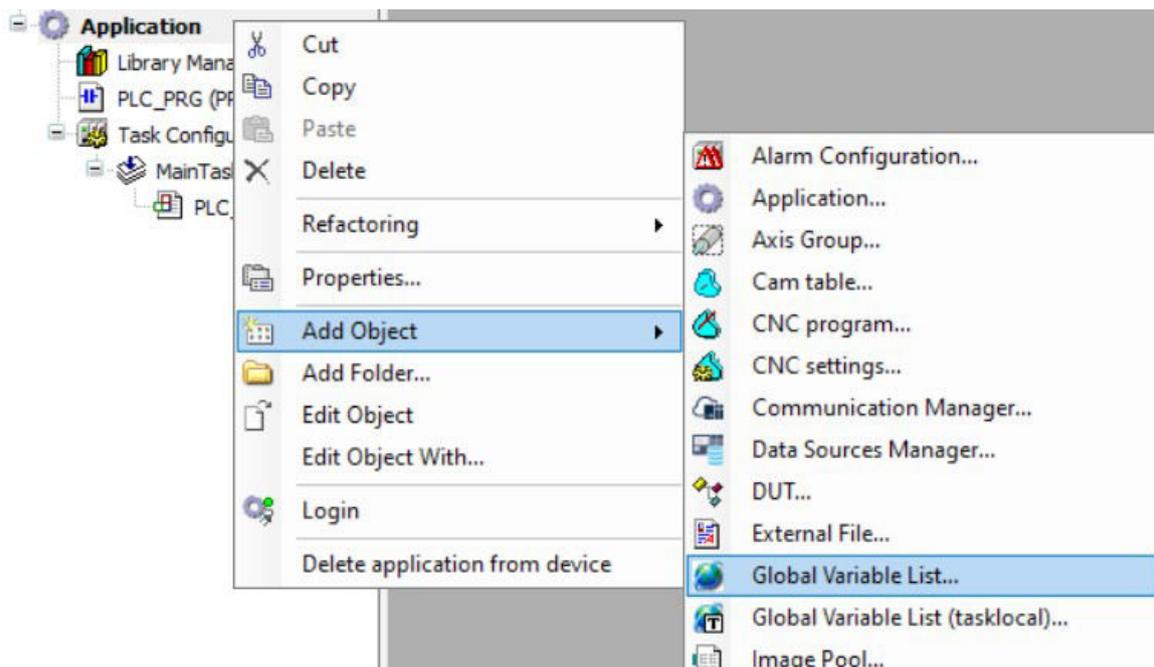


Figura 22: Creación de la lista de Variables Global

Al crear la lista, esta se abre automáticamente, de no ser así, puede abrirse con un doble clic en el nombre de la misma situado en el árbol del proyecto. A continuación, se deberán introducir una a una las variables indicadas anteriormente entre los rótulos *VAR_GLOBAL* y *END_VAR*. El resultado final debería asemejarse a las Figuras 23, 24, 25, 26 y 27. En ellas se puede observar que se han añadido comentarios y ordenado las variables en la lista por funcionalidad, así como diferenciando entre inputs y outputs.

```

1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3  //INPUTS - Sensores/Botones
4  //Pulsadores Cuadro General
5  iArranqueCintas      :BOOL;
6  iEmergenciaTL       :BOOL;
7  iSilencioAlarma     :BOOL;
8  iSolicitudParo      :BOOL;
9  //Empujador piezas metálicas
10 iSensorMetalicas    :BOOL;
11 iEmpujadorDelante   :BOOL;
12 iEmpujadorDetras    :BOOL;
13 //Paneles Paletizadoras
14 iMarchaLM           :BOOL;
15 iMarchaLN           :BOOL;
16 iEmergenciaLM       :BOOL;
17 iEmergenciaLN       :BOOL;
18 iSolicitudParoLM    :BOOL;
19 iSolicitudParoLN    :BOOL;
20 //Paletizador Metálicas
21 iPM_DetecPieza      :BOOL;
22 iPM_LimGiroPin      :BOOL;
23 iSensFinLM          :BOOL;
24 iSensEntLM          :BOOL;
25 iPM_PalletOK        :BOOL;
26 iPM_PositionX       :REAL; // En V
27 iPM_PositionY       :REAL; // En V
28 iPM_PositionZ       :REAL; // En V
    
```

Figura 23: Variables Globales 1/5

```

29 //Paletizador Normales
30 iPN_DetecPieza      :BOOL;
31 iPN_LimGiroPin      :BOOL;
32 iSensFinLN          :BOOL;
33 iSensEntLN          :BOOL;
34 iPN_PalletOK        :BOOL;
35 iPN_PositionX       :REAL; // En V
36 iPN_PositionY       :REAL; // En V
37 iPN_PositionZ       :REAL; // En V
38 //Rotativa
39 iPalletEntradaRM    :BOOL;
40 iPalletEntradaRN    :BOOL;
41 iPosEntM            :BOOL;
42 iPosEntN            :BOOL;
43 iEntPalletRot       :BOOL;
44 iSalPalletRot       :BOOL;
45 //Otros Sensores
46 iLN_Llena           :BOOL;
47 iLM_Llena           :BOOL;
48 iPuertaEyPM         :BOOL;
49 iPuertaEyPN         :BOOL;
50 iPuertaPMYS         :BOOL;
51 iPuertaPNYS         :BOOL;
52 iPuertaInt          :BOOL;
    
```

Figura 24: Variables Globales 2/5

```

53 //OUTPUTS - ACTUADORES
54 //Panel General
55 oLuzArranqueCintas  :BOOL;
56 oLuzParado           :BOOL;
57 oLuzSolicitudParo   :BOOL;
58 oLuzEmergenciaGeneral :BOOL;
59 oLuzSilencioEmerg   :BOOL;
60 oAlarmaEmergGeneral :BOOL;
61 //Paneles Paletizadoras
62 oLuzMarchaLM        :BOOL;
63 oLuzMarchaLN        :BOOL;
64 oLuzParadoLM        :BOOL;
65 oLuzParadoLN        :BOOL;
66 oLuzSolicitudParoLM :BOOL;
67 oLuzSolicitudParoLN :BOOL;
68 oLuzEmergLM         :BOOL;
69 oLuzEmergLN         :BOOL;
70 //Motores Cintas
71 oCintaEntrada       :BOOL;
72 oCintaM1             :BOOL;
73 oCintaM2             :BOOL;
74 oCintaM1             :BOOL;
75 oCintaN1             :BOOL;
76 oCintaN1             :BOOL;
77 oCintaN2             :BOOL;
78 oTopeCintaM         :BOOL;
79 oTopeCintaN         :BOOL;
80 //Empujador
81 oEmpujadorPiston    :BOOL;
    
```

Figura 25: Variables Globales 3/5

```

82 //Rotativa
83 oGiroRotativa        :BOOL;
84 oRodillosRotativaP   :BOOL;
85 oRodillosRotativaN   :BOOL;
86 oEsperaM             :BOOL;
87 oEsperaN             :BOOL;
88 //Paletizador Metálicas
89 oPM_Agarre           :BOOL;
90 oPM_GiroPinza        :BOOL;
91 oPM_LuzA             :BOOL;
92 oPM_LuzR             :BOOL;
93 oPM_LuzV             :BOOL;
94 oPM_StopEntrada     :BOOL;
95 oPM_SetPointX        :REAL; //En V
96 oPM_SetPointY        :REAL; //En V
97 oPM_SetPointZ        :REAL; //En V
98 //Paletizador Normales
99 oPN_Agarre           :BOOL;
100 oPN_GiroPinza        :BOOL;
101 oPN_LuzA             :BOOL;
102 oPN_LuzR             :BOOL;
103 oPN_LuzV             :BOOL;
104 oPN_StopEntrada     :BOOL;
105 oPN_SetPointX        :REAL; //En V
106 oPN_SetPointY        :REAL; //En V
107 oPN_SetPointZ        :REAL; //En V
    
```

Figura 26: Variables Globales 4/5

```

108 //Motores Rodillos
109 oRodillosM1          :BOOL;
110 oRodillosM2          :BOOL;
111 oRodillosM3          :BOOL;
112 oRodillosN1          :BOOL;
113 oRodillosN2          :BOOL;
114 oRodillosN3          :BOOL;
115 oRodillosS1          :BOOL;
116 oRodillosS2          :BOOL;
117 oRodillosS3          :BOOL;
118 oRodillosSC          :BOOL;
119 oRodillosS4          :BOOL;
120 oRodillosS5          :BOOL;
121 //Emisores-Extractores
122 oEmisorPiezas        :BOOL;
123 oEmisorPallets1     :BOOL;
124 oEmisorPallets2     :BOOL;
125 oExtractorPallets    :BOOL;
126 END_VAR
    
```

Figura 27: Variables Globales 5/5

Una vez definidas las variables globales, es hora de abrir la pestaña del programa. Esto se hace mediante un doble clic en el ítem *PLC_PRG (PRG)* del árbol del proyecto.

Llega el momento de realizar la programación en LD, para lo cual conviene recordar la estructura que debe tener un programa de estas características, si el proceso para programar está basado en la implementación desde un Grafcet, como es el caso que se está estudiando. La Figura 28 muestra esquemáticamente la mencionada estructura.

NOTA: Los pasos de desactivación y activación de etapas pueden combinarse a la hora de realizar la programación. Se ha optado por realizar ambos pasos conjuntamente, pero el resultado sería el mismo de realizarse por separado.

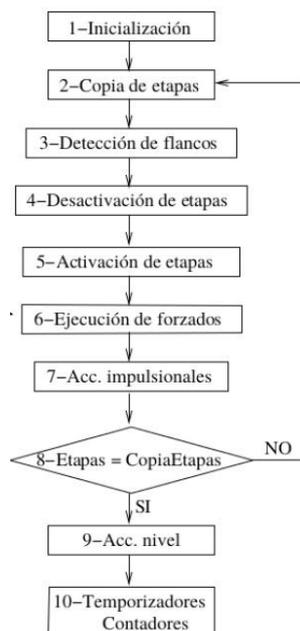


Figura 28: Etapas de un programa en LD

Antes de comenzar, se observa que la ventana central se divide en dos espacios. El primero de ellos será dónde definiremos las variables locales del proyecto, es decir, aquellas que serán utilizadas por el programa, pero que no corresponden a ningún input/output. En este proyecto, además de las variables correspondientes a las etapas, copia de etapas e inicialización (Figura 29), se incluyen otras variables adicionales que se utilizarán como soporte a lo largo del código (Figura 30). A esta lista se añadirán automáticamente en el momento de su definición, las variables de flancos de subida y bajada y los temporizadores.

```

    PLC_PRG x
    1 PROGRAM PLC_PRG
    2 VAR
    3 //Grupo de Etapas Grafcet Cuadro General
    4 E0:WORD;
    5 CE0:WORD;
    6 //Grupo de Etapas Grafcet Cintas y Rodillos
    7 E1:WORD;
    8 CE1:WORD;
    9 //Grupo de Etapas Grafcet Rotativa
    10 E2:WORD;
    11 CE2:WORD;
    12 //Grupo de Etapas Grafcet Rotativa
    13 E3:WORD;
    14 CE3:WORD;
    15 //Grupo de Etapas Grafcet Paletizador M
    16 E4:WORD;
    17 CE4:WORD;
    18 //Grupo de Etapas Grafcet Paletizador N
    19 E5:WORD;
    20 CE5:WORD;
    21 //Grupo de Etapas Grafcet Cuadro Paletizador M
    22 E6:WORD;
    23 CE6:WORD;
    24 //Grupo de Etapas Grafcet Cuadro Paletizador N
    25 E7:WORD;
    26 CE7:WORD;
    27 //Primer loop - INICIALIZACIÓN
    28 scan1: BOOL := 1;
    29 FirstScan: BOOL;
    
```

Figura 29: Variables locales de Etapas, Copia de Etapas e Inicialización

```

30 //Otras Variables
31 TempSensFinLM: BOOL;
32 TempSensFinLN: BOOL;
33 PiezasPalletM: INT:=9; //Piezas máximas por pallet M (empieza en 0)
34 PiezasPalletN: INT:=9; //Piezas máximas por pallet N (empieza en 0)
35 PiezasM: INT; //Conteo de piezas en Paletizador M
36 PiezasN: INT; //Conteo de piezas en Paletizador N
37 PosMOK: BOOL; //Determina cuando la posición del Paletizador M es correcta
38 PosNOK: BOOL; //Determina cuando la posición del Paletizador N es correcta
39 //Variables que convierten a enteros (INT) los valores REAL de SetPoint y Position de los Paletizadores
40 SetMX: INT;
41 SetMY: INT;
42 SetMZ: INT;
43 PosMX: INT;
44 PosMY: INT;
45 PosMZ: INT;
46 SetNX: INT;
47 PosNX: INT;
48 SetNY: INT;
49 PosNY: INT;
50 SetNZ: INT;
51 PosNZ: INT;
    
```

Figura 30: Otras Variables Locales que se utilizarán en el código

Definidas todas las variables (tanto globales como locales) que intervienen en el código, ya es momento de comenzar con la programación. Como se mencionó al inicio de este punto, debido a su extensión el código completo se incluye seccionado en el ANEXO IX y a continuación se comentarán los aspectos más relevantes de cada sección:

- **Inicialización:** Corresponde a las acciones que se llevan a cabo únicamente durante el primer loop (la primera lectura) de ejecución del programa. Todas las etapas se resetean como se muestra en la Figura 31 (esto para cada una de las variables EX), es decir, se les da un valor FALSE o se pone el bit correspondiente a 0. No obstante, se activa (valor TRUE o bit a 1) la etapa 0 del Grafcet 0 (Figura 32), que corresponde al Grafcet maestro, encargado de coordinar los demás Grafcets.

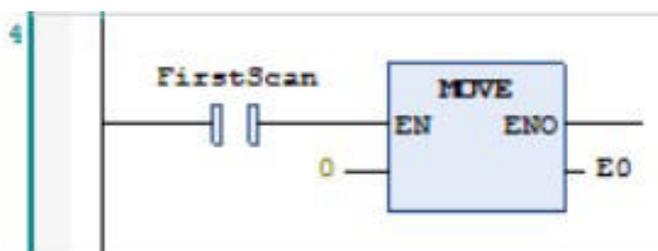


Figura 31: Reseteo de Etapas del Grafcet 0



Figura 32: Activación de la Etapa 0 del Grafcet 0

Para introducir el contacto normalmente abierto (NO), como el de FirstScan, se puede pulsar la combinación *Ctrl+K* o bien hacer clic en el icono de contacto abierto en la barra de herramientas superior (ver Figura 33) donde también se encuentra la bobina de SET (S). La caja de operación MOVE puede encontrarse en el menú que se muestra a la derecha de la pantalla (Figura 34) en el apartado *Other Operators*.



Figura 33: Barra de herramientas superior

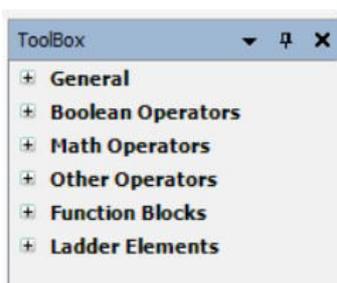


Figura 34: Menú a la derecha del espacio de trabajo

- Copia de Etapas: En la primera línea de esta sección se debe insertar una etiqueta (*label*) con el nombre de Salto (para ello, basta con hacer clic derecho en la línea y presionar *Insert Label*), puesto que esta es la destinación del salto que se realiza tras la comprobación del paso 7 (8 en la Figura 28). En esta sección se realiza la copia del estado de las variables EX a las variables CEX, para ello, se opera como en el ejemplo de la Figura 35 para todas las variables Etapa y Copia de Etapa.

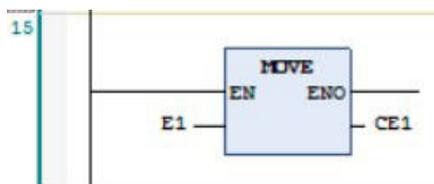


Figura 35: Copia de valores de la variable E1 a la variable CE1

- Detección de Flancos: En esta sección se deben incluir todas las variables para las cuales es necesario detectar flancos de subida (R_TRIG) o de bajada (F_TRIG). Para conseguir la detección se debe colocar en la línea un contacto NO con la variable a detectar y seguidamente el bloque de función correspondiente. Dichos bloques se encuentran en el menú de la derecha, mostrado anteriormente (Figura 34) dentro del apartado *Function Blocks*. Un ejemplo de detección de flanco se puede observar en la Figura 36.

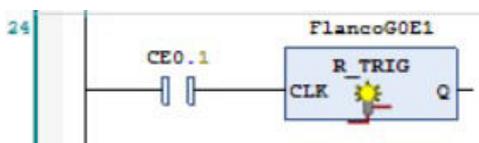


Figura 36: Detección de flanco de subida de la Etapa 1 del Grafset 0

- Desactivación y activación de etapas: Es una de las secciones más densas del código de programación. En ella, se debe introducir, una a una, las condiciones bajo las que se desactiva una etapa y se activa una o varias etapas, según el caso. Para ello se introduce un contacto abierto (NO) con el nombre de la Copia de Etapa que se va a desactivar. Junto a él se colocan otros contactos NO o Normalmente Cerrados (NC) con las condiciones de transición de cada etapa. Tras ello se introduce una Bobina de Reset con el nombre de la Etapa (no Copia de Etapa) que se desactiva, y una o más Bobinas de Set con el nombre de la o las etapas que se activan. En las Figuras 37 y 38 se muestra un ejemplo de desactivación y activación de una etapa y de múltiples etapas, respectivamente.

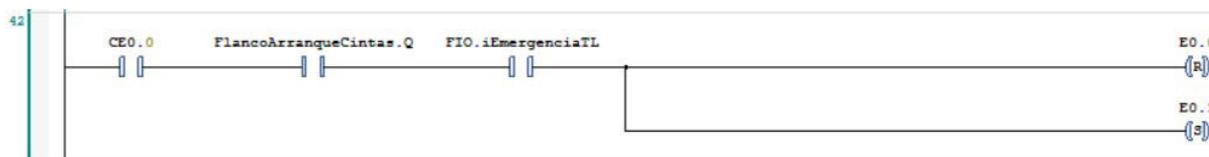


Figura 37: Desactivación Etapa 0 del Grafcet 0 y activación Etapa 1 del Grafcet 0



Figura 38: Desactivación Etapa 0 del Grafcet 1 y activación de las etapas 1 – 6

Durante este paso, es importante remarcar dos casos especiales, que se da a la hora de definir las transiciones del Grafcet de control de ambos paletizadores.

El primero de ellos es la definición de la variable PosXOK (con X siendo M o N según el paletizador. Para que esta variable se active, se debe comprobar que la posición del paletizador es igual al *set point* dado en cada momento. Esto no se puede realizar directamente con las variables provenientes del paletizador, ya que al tratarse de números en formato REAL, la comparación entre los mismos puede dar lugar a errores. Para ello se introduce el código que se observa en la Figura 39 donde se crean variables adicionales de *Set Point* y Posición transformando a enteros los valores de las variables de los paletizadores multiplicadas por 10 (ya que dichos valores se dan con precisión de 1 decimal). A continuación, se comparan las nuevas variables y se determina si la posición es correcta.



Figura 39: Definición de las variables PosMOK y PosNOK

El segundo aspecto ligeramente especial (y que también toma parte en las líneas de código anteriores) se da a la hora de querer comprobar si el número de piezas en el paletizador es igual o diferente a 0 (Figura 40). Para ello se usan unos operadores que pueden encontrarse en el menú mostrado anteriormente en la Figura 34, concretamente en el apartado de *Math Operators*, llamados *Equal (EQ)* y *Not Equal (NE)*.

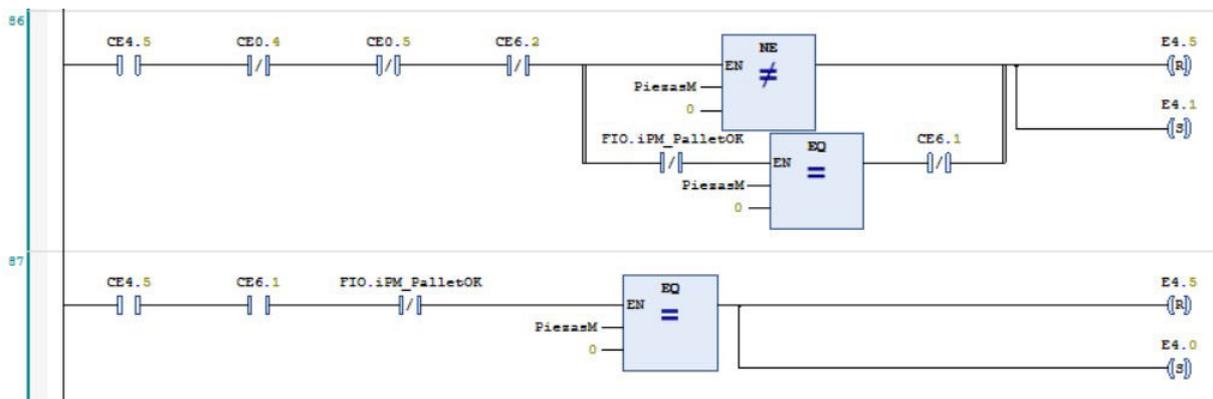


Figura 40: Uso de los operadores EQ y NE

- Ejecución de Forzados: En esta sección, como su nombre indica, se incluyen las órdenes de forzado del Grafcet 0 (maestro) a los esclavos. Para ello, se utilizan los operadores AND y OR, que se pueden encontrar en el apartado *Boolean Operators* del menú de la Figura 34.

Los operadores AND se utilizan en operaciones de reseteo (Figura 41), y se combinan con los OR para establecer los Grafcets en una etapa en concreto (Figura 42). Se introduce el valor en hexadecimal de la etapa o etapas afectadas (la etapa 0 es el valor $16\#00001$, la etapa 1 es $16\#00002$ y ambas sería $16\#00003$). Se debe tener en cuenta si los forzados son durante todo el tiempo en que una etapa está activa o si son forzados por flanco (en los que, tras el forzado, el Grafcet esclavo sigue evolucionando libremente).

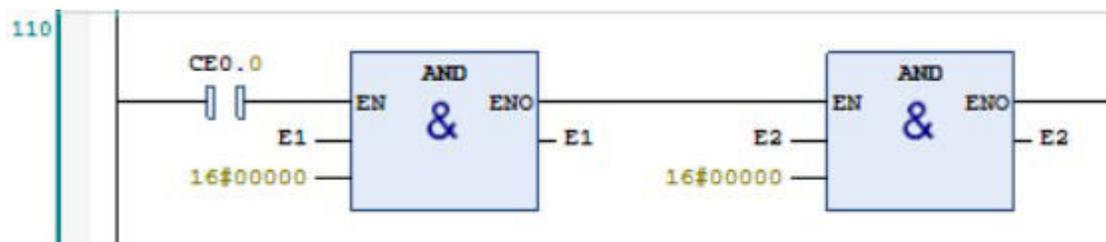


Figura 41: Forzado de reseteo de etapas (en el Grafcet 0 - Etapa 0)

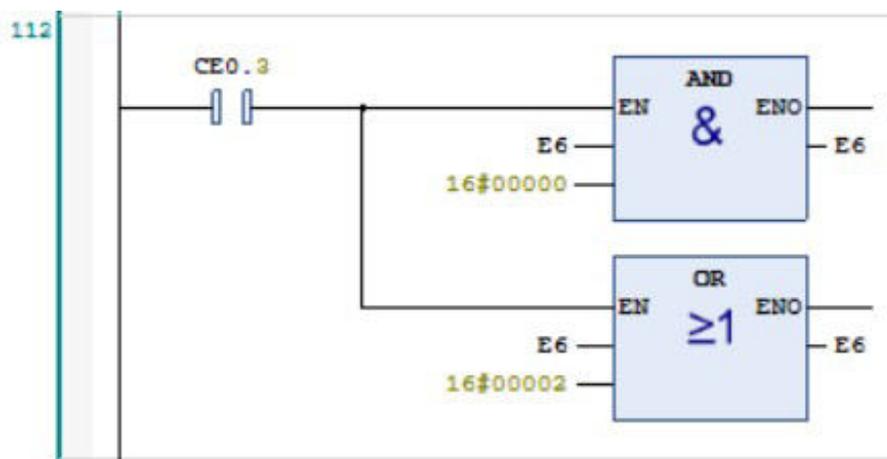


Figura 42: Forzado de Grafcets a cierta etapa combinando AND y OR

- Acciones impulsionales: Este paso recoge las acciones que se activan por flanco, es decir, aquellas que deben realizarse siempre que se active una etapa (aunque ésta sea inestable) y una sola vez (suele tratarse de asignaciones o contadores). Para detectar el momento de la activación se coloca un contacto NO con el nombre de la Etapa, seguido de uno NC con el nombre de la Copia de Etapa.

En este caso en particular, se añade a las condiciones de ejecución de las acciones impulsionales el hecho de **no** encontrarse en ninguna etapa de emergencia, ya que todas ellas son acciones relacionadas con el movimiento o la configuración de los paletizadores, que queremos que, dada una emergencia se queden congelados en el punto en que se encuentren.

En la Figura 43 se muestra una de las acciones impulsionales de este proceso, correspondiente a la orden de movimiento del brazo del paletizador hacia un punto en concreto según la pieza a depositar en el pallet. Para introducir el código, se necesita una caja de tipo EXECUTE, que se puede insertar o bien desde el menú de la derecha, en el apartado *General* o haciendo clic derecho en la línea de programación y seguidamente en *Insert Execute Box*. El código en detalle se encuentra al final del ANEXO IX.

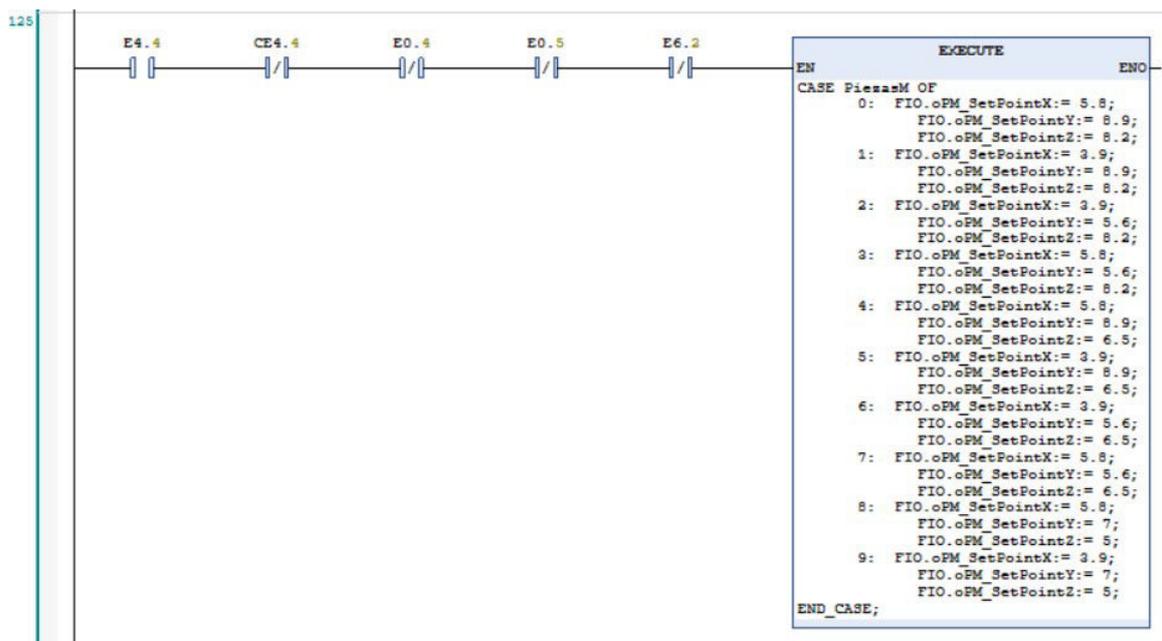


Figura 43: Acción impulsional con caja de tipo EXECUTE

- Comprobación Igualdad ETAPAS – COPIA DE ETAPAS: Este paso es muy importante, ya que hace posible que tanto los forzados, como los flancos y las acciones impulsionales se ejecuten correctamente y de forma separada a las acciones de nivel. Se debe comprobar, individualmente y en cada ocasión que se llegue a este punto, que todas las EX son iguales a su homólogo CEX (siendo X el número de Grafcet de 0 hasta 7). De no ser así se realiza un salto a la etiqueta Salto (podría tener cualquier nombre siempre que coincida con la introducida en este punto) insertada en la sección de Copia de Etapas. Para ello, cada comprobación en una línea, se introduce un contacto NO con el valor TRUE, para que siempre se evalúe esta condición y un operador NE a cuya salida se colocará un elemento de tipo *Jump* con la etiqueta Salto. *Jump* puede encontrarse en el menú de la derecha, apartado *General* y en la barra de herramientas superior. La Figura 44 muestra el ejemplo para uno de los Grafcets.



Figura 44: Comprobación de desigualdad entre E0 y CE0.

- Acciones de nivel: Penúltimo bloque y otro muy denso, puesto que en él, se deben incluir todas las acciones de nivel, es decir, aquellas que tienen lugar cuando una etapa se encuentra activa (aunque pueden tener otras condiciones adicionales). Su programación se realiza de forma similar a la desactivación y activación de etapas, utilizando contactos NO y NC, con la diferencia de que se utilizan bobinas normales y no de tipo Set o Reset. Es importante, de lo contrario se producen errores de funcionamiento, que cada

actuador (en forma de bobina) aparezca en una sola y única línea, y no se repita en otras. Es decir, si un actuador debe funcionar en varias etapas, se introducirán las condiciones en paralelo, pero la bobina de dicho actuador sólo aparecerá una vez (ver Figura 45)

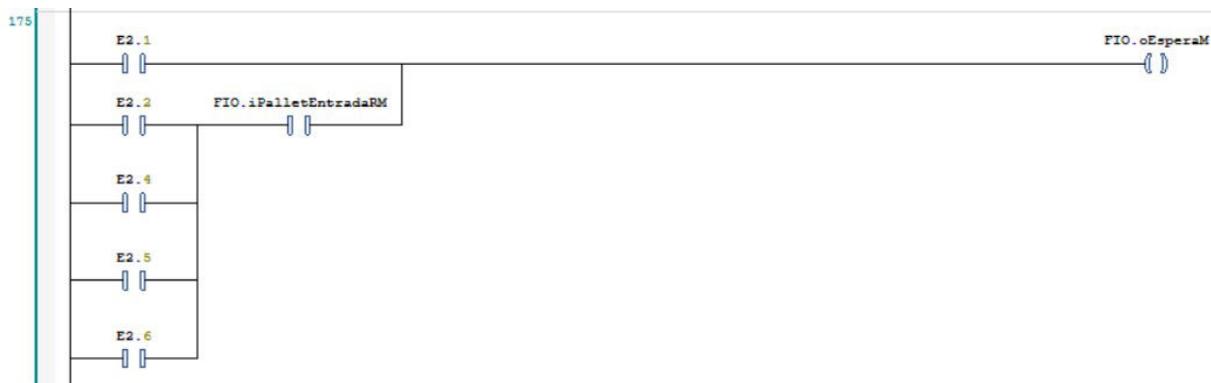


Figura 45: Actuador que se activa bajo distintas Etapas y condiciones

Un caso peculiar ocurre a la hora de programar la aproximación vertical a la pieza por parte del brazo del paletizador. Para realizar dicho movimiento, se introduce una línea como la que se muestra en la Figura 46, en la que, más allá de las condiciones básicas, se añade un operador tipo ADD (añadido desde el menú de la derecha, categoría *Math Operators*, del tipo 2-Inputs) que se encarga de añadir a la variable del SetPoint vertical del brazo un valor de 0.1 cada escaneo del PLC.

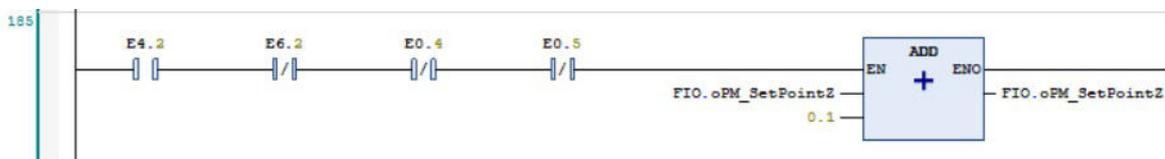


Figura 46: Programación del descenso progresivo del brazo del Paletizador

- Temporizadores: Para este proyecto, se han considerado necesarios un total de 8 temporizadores, todos ellos de retraso en la activación (tipo TON). Se añaden colocando un contacto con la variable con cuya activación comenzará la cuenta y una caja de temporizador TON, que se puede encontrar en el apartado *Function Blocks*. Dicho bloque de función tiene dos entradas y dos salidas. Las salidas las dejaremos en blanco, la entrada IN irá conectada a el o los contactos de entrada y en la entrada PT (*preset time*, tiempo establecido) colocaremos el tiempo que queremos que controle el temporizador con la estructura *T#XXs* sustituyendo XX por el número de segundos a temporizar. Un ejemplo de temporizador se muestra en la Figura 47.

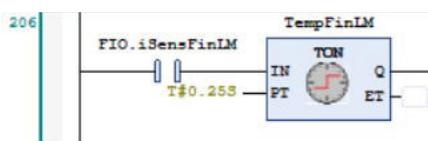


Figura 47: Temporizador de espera en final de línea de 0,25s

De esta forma queda completamente entendido el procedimiento para la programación en CODESYS de la línea de paletizado diseñada, dejando claro que los diferentes pasos explicados se deben colocar de forma ordenada para un correcto funcionamiento del programa de control.

1.7.5. Conexión CODESYS – Factory I/O

Una vez se dispone del programa completo, el siguiente paso es realizar la configuración, tanto desde CODESYS como de Factory I/O para que la comunicación entre ambos sea correcta y se pueda utilizar el ordenador como PLC Virtual.

Esto puede realizarse a través de varios métodos, como se mencionó en el punto 1.6, aunque el escogido finalmente ha sido el protocolo OPC UA. Se ha preferido éste en defecto de las otras opciones debido a dos principales ventajas:

- OPC UA es un protocolo incluido en el estándar IEC62541 para intercambio de datos entre sensores y aplicaciones y, a su vez, dicho estándar, está regularizado por la norma *UNE-EN IEC 62541:2020*: la cual determina en qué condiciones se debe dar dicha comunicación
- OPC UA representa una evolución respecto a OPC DA, ya que permite las mismas funcionalidades mejorando la rapidez la puesta a punto de la comunicación, además de un cifrado de la información transmitida.
- En comparación con el protocolo Modbus TCP/IP, OPC UA ofrece una implementación en un rango más amplio de equipos, además de que permite comunicarse con otros protocolos, cosa que no sucede con el Modbus. Enfocándolo en el proyecto, la configuración de OPC UA es más sencilla que la de Modbus, por lo que, siendo ambas funcionales, se ha optado por la primera.

Vistas las razones de la elección de OPC UA respecto a sus alternativas, es momento de conocer cómo preparar tanto CODESYS como Factory I/O para lograr una comunicación satisfactoria y poder realizar la simulación correctamente.

1.7.5.1. Configuración de CODESYS

Antes de comenzar, será necesario comprobar que dos elementos necesarios se encuentran instalados en el ordenador donde se va a realizar la simulación. El primero de ellos es *CODESYS Control Win SysTray* (no es válida la versión con el mismo nombre que incluye x64). Normalmente se instala junto con CODESYS, pero puede darse el caso en el que su instalación deba hacerse de forma manual.

Para ello, se deberá acceder a la tienda de CODESYS (store.codesys.com) y descargar el paquete llamado *CODESYS Control Win SL*. La versión completa es de

pago, pero se puede descargar y ejecutar de forma gratuita (modo demo) para realizar simulaciones cortas (de una duración máxima de 2 horas, tras las cuales se requiere un reinicio manual). De requerir el uso de este mismo protocolo en una línea real, que deberá estar trabajando durante muchas horas seguidas, la versión completa deberá ser adquirida.

Tras la descarga, se deberá descomprimir la carpeta *.zip* e instalar el complemento con el ejecutable *CODESYS Control Win 3.5.18.20* (de nuevo tener cuidado porque la versión 64 no es válida). Tras una instalación correcta (puede ser necesario un reinicio del ordenador), aparecerá un icono como el de la Figura 48 en la barra de tareas de Windows. De no aparecer, debería poderse abrir desde el menú de inicio de Windows, sino, comprobar de nuevo la instalación. Una vez instalado, haciendo clic derecho en el icono y en *Start PLC*, el icono debería mostrarse como en la Figura 49.



Figura 48: Icono de CODESYS Control Win SysTray (parado)



Figura 49: Icono de CODESYS Control Win SysTray (en marcha)

El segundo elemento, que si suele quedar instalado por defecto con CODESYS es *CODESYS Gateway SysTray*. Si por algún motivo no aparece iniciado con el icono en la barra de tareas (Figura 50), verificar que se encuentra instalado (buscar en el menú inicio). En este caso es indiferente que se trate de la versión “normal” o x64.



Figura 50: Icono de CODESYS Gateway SysTray (en marcha)

Si se verifica que no está instalado, se puede descargar de forma gratuita en la tienda de CODESYS. Aparece como *CODESYS Edge Gateway for Windows* y su instalación es sencilla.

Con todo esto verificado, es momento de pasar al programa. En el árbol del proyecto, se hará doble clic en el elemento *Device*. Esto abrirá una ventana como la que se muestra en la Figura 51. Allí, se presionará el rótulo *Scan Network*, y aparecerá una ventana como la de la Figura 52, con la diferencia que el nombre del equipo señalado en azul en la imagen será diferente (puede ser cambiado posteriormente al que se desee). Se debe seleccionar dicho dispositivo y presionar *OK*.

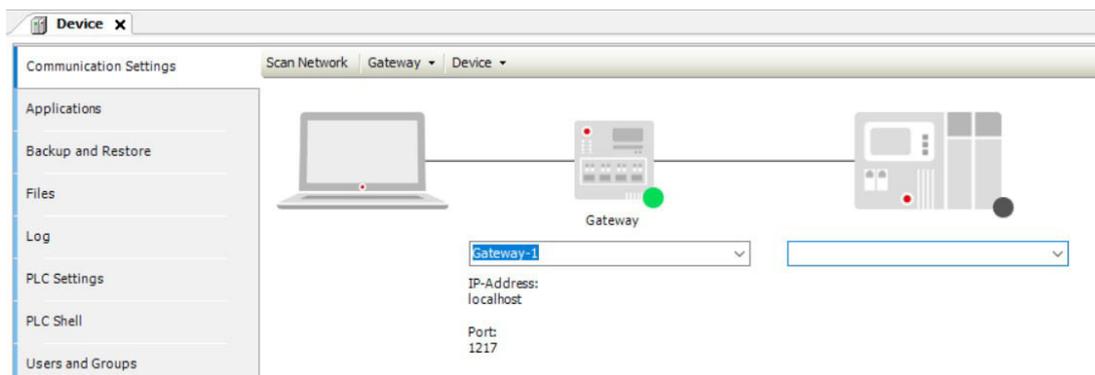


Figura 51: Ventana principal de Device en CODESYS

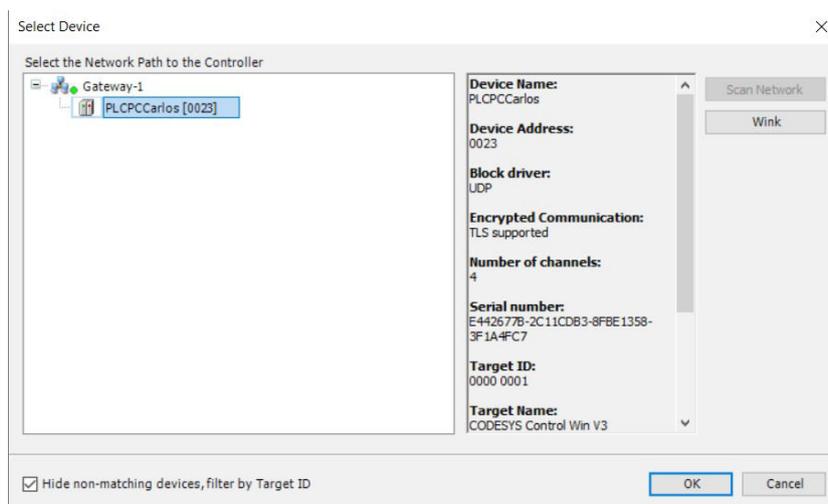


Figura 52: Ventana que se abre al presionar Scan Network

Tras esto, puede aparecer una ventana de advertencia (no apareció en nuestro caso) que pregunta sobre la activación de la gestión de usuario con tal de conectar con el dispositivo. Simplemente se debe pulsar en *Yes*, lo cual nos llevará a una nueva ventana que pedirá crear un usuario del dispositivo (*Device User*). Simplemente llenar el formulario con un nombre de usuario y contraseña (recordar para más adelante) y presionar en *OK*. Una nueva ventana pedirá realizar un *Log In* con los datos recién introducidos.

El siguiente paso dentro de la ventana principal de *Device*, fundamental para que la comunicación con *Factory I/O* sea correcta y el que no era posible realizar con la versión *x64* de *Control Win SysTray*, se realiza en el desplegable del rótulo *Device*, la opción de *Change Communication Policy...* (ver Figura 53)

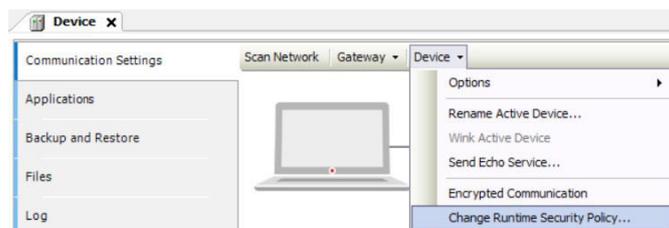


Figura 53: Desplegable de menú Device. Opción que se debe seleccionar.

Aparecerá una ventana (Figura 54) donde se deberá modificar el apartado de *Device User Management* de forma que *New Policy* quede con la opción de *Optional user management* y marcando la casilla *Allow anonymous login*.

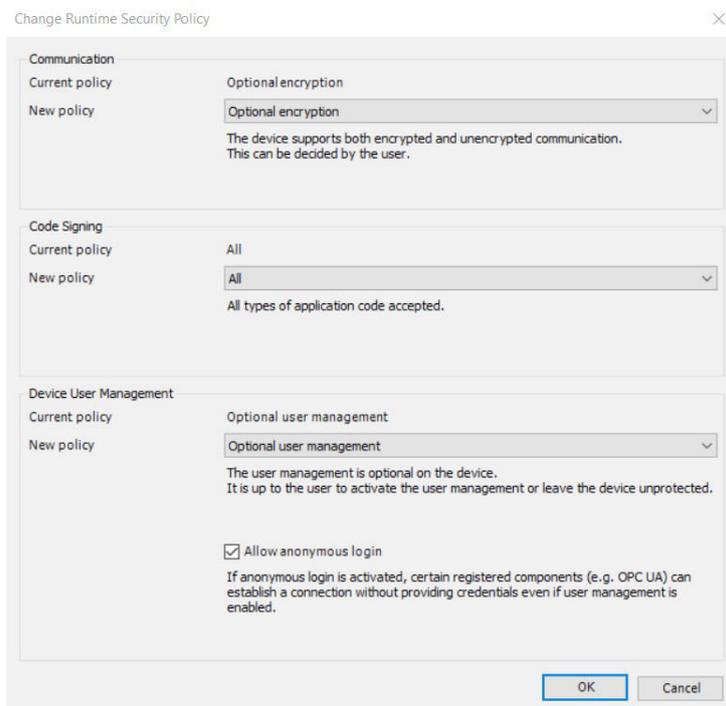


Figura 54: Ventana de Change Runtime Security Policy

Sin salir de la pestaña *Device* en el submenú de la izquierda (donde hasta ahora nos encontramos en *Communication Settings*) accederemos a la pestaña *Access Rights*. Presionaremos el botón de *Synchronization* (flechas verdes) para que carguen las opciones de la ventana. En ellas, nos dirigimos a *Device* → *RemoteConnections* → *OPCUAServer* y se debe verificar que en la fila *Anonymous_OPCUAServer* las opciones *Modify*, *View* y *Execute* estén con un símbolo + en verde (Figura 55).

Rights	Add/R...	Modify	View	Execute
Administrator		+	+	+
Anonymous_OPCUAServer		+	+	+
Developer		-	+	-
Service		-	+	-
Watch		-	+	-

Figura 55: Configuración en la pestaña de Access Rights

Con todo esto configurado, se debe hacer clic en el icono *Generate Code* de la barra de tareas superior (Figura 56).



Figura 56: Icono de la opción Generate Code

El siguiente paso consiste en hacer clic derecho en *Application*, seguidamente en el desplegable *Add Object* se selecciona *Symbol Configuration*. Se le puede dar un nombre (opcional) y se debe verificar que la casilla *Support OPC UA features* está marcada antes de hacer clic en *Add*.

En la ventana que se abre automáticamente, hacer clic en *Build* y seleccionar FIO (la lista de variables globales del programa). Tras esto, hacer clic de nuevo en *Build*. La pantalla debe quedar como se muestra en la Figura 57.

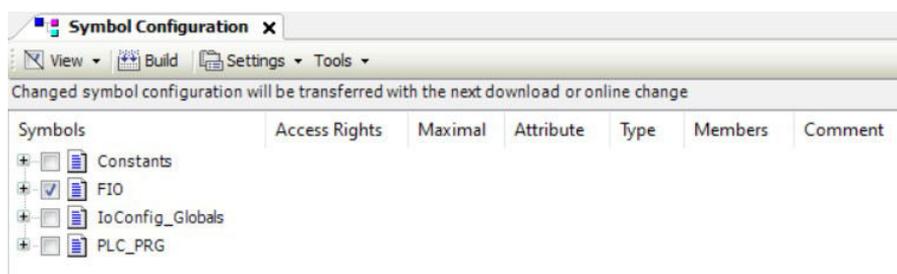


Figura 57: Pestaña de Symbol Configuration

Por último, hacer clic en el botón *Login* (Figura 58) de la barra de tareas superior (volver a introducir usuario y contraseña si se solicita) y, posteriormente, en el botón de *Start* (Figura 59). Si tras pulsar en *Login* (e introducir la contraseña) se muestra una pantalla donde se informa de que ha habido cambios en la aplicación (Figura 60), seleccionar la opción *Login with download* y marcar *Update boot application*, seguidamente presionar *OK*.



Figura 58: Botón de Login



Figura 59: Botón de Start

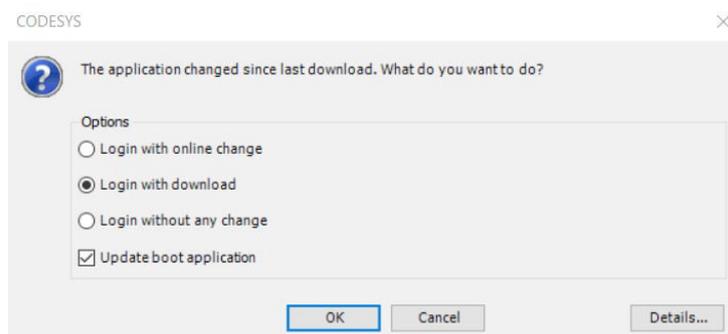


Figura 60: Pantalla de confirmación de cambios en la aplicación

Con esto queda completamente configurado CODESYS y listo para la simulación. El siguiente paso radica en preparar Factory I/O para la comunicación, como veremos en el siguiente punto.

1.7.5.2. Configuración de Factory I/O

Para la configuración de la comunicación de CODESYS con Factory I/O el primer paso es abrir el software y cargar el proyecto. Tras esto, nos dirigimos al menú *FILE* → *Drivers*. Allí, en el desplegable de la parte superior de la pantalla (Figura 61), se selecciona *OPC Client DA/UA*.

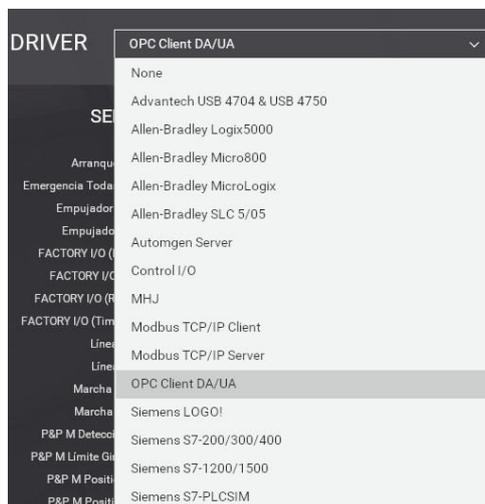


Figura 61: Desplegable de la pestaña Drivers en Factory I/O

Tras esto, nos dirigimos a *CONFIGURATION*, a la derecha de la barra superior. Allí deberemos escribir ***opc.tcp://localhost:4840*** en el campo *OPC Server* y presionar la tecla *Enter* de nuestro teclado. En el apartado *Items/Nodes* escribiremos *FIO* (o el nombre que hayamos otorgado a la lista de variables global) en *Filter names that contain:* incrementaremos el valor límite a 150 (ya que el proyecto tiene más de 100 inputs y outputs) y seguidamente haremos clic en *BROWSE*. La Figura 62 muestra cómo debería quedar configurada esta página.

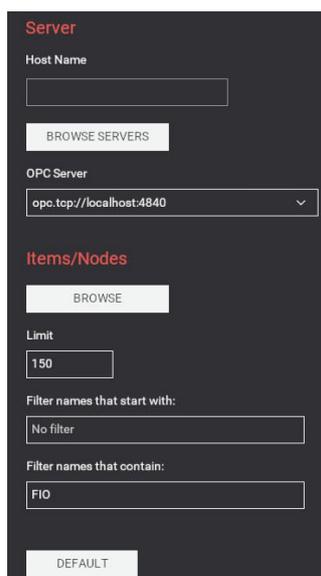


Figura 62: Página de Configuración en Factory I/O

Volviendo a la ventana anterior, donde se encuentran a la izquierda los sensores del proyecto y a la derecha los actuadores, habrá aparecido en el centro una columna con las variables que se han definido en la lista FIO. El último paso será mapear, es decir colocar en su lugar, cada uno de los sensores y actuadores del proyecto con su homólogo en CODESYS. El resultado debería ser como el que se muestra en las Figuras 63, 64 y 65.

Server: opc.tcp://localhost:4840 (105)	
Arranque Cintas	iArranqueCintas
Parada Emergencia PM	iEmergenciaLM
Parada Emergencia PN	iEmergenciaLN
Emergencia Todas Líneas	iEmergenciaTL
Empujador Delante	iEmpujadorDelante
Empujador Detrás	iEmpujadorDetras
SensorEntradaPalletRot	iEntPalletRot
LíneaMLlena	iLM_Llena
LíneaNLlena	iLN_Llena
Marcha Línea M	iMarchaLM
Marcha Línea N	iMarchaLN
PalletEntradaRotativaM	iPalletEntradaRM
PalletEntradaRotativaN	iPalletEntradaRN
P&P M Detección Caja	iPM_DetecPieza
P&P M Límite Giro Pinza	iPM_LimGiroPin
Pos. Pallet OK M	iPM_PalletOK
P&P M Position X (V)	iPM_PositionX
P&P M Position Y (V)	iPM_PositionY
P&P M Position Z (V)	iPM_PositionZ
P&P N Detección Pieza	iPN_DetecPieza
P&P N Límite Giro Pinza	iPN_LimGiroPin
Pos. Pallet OK N	iPN_PalletOK
P&P N Position X (V)	iPN_PositionX
P&P N Position Y (V)	iPN_PositionY
P&P N Position Z (V)	iPN_PositionZ
PosEntradaRotativaM	iPosEntM
PosEntradaRotativaN	iPosEntN
Puerta Cinta E y PM	iPuertaEyPM
Puerta Cinta E y PN	iPuertaEyPN
Puerta Interior	iPuertaInt
Puerta PM y Salida	iPuertaPMyS
Puerta PN y Salida	iPuertaPNyS
SensorSalidaPalletRot	iSalPalletRot

Figura 63: Mapeo de Sensores y Actuadores 1/3

SensorEntradaM	iSensEntLM	
SensorEntradaN	iSensEntLN	
Sensor Pieza Final Línea M	iSensFinLM	
Sensor Pieza Final Línea N	iSensFinLN	
Sensor Metálicas	iSensorMetalicas	
Silenciar Alarma	iSilencioAlarma	
Solicitud Paro	iSolicitudParo	
Solicitud Paro Línea M	iSolicitudParoLM	
Solicitud Paro Línea N	iSolicitudParoLN	
	...armaEmergGeneral	Alarma Emergencia General
	oCintaEntrada	Cinta E
	oCintaM1	Cinta M1
	oCintaM2	Cinta M2
	oCintaMi	Cinta Mi
	oCintaN1	Cinta N1
	oCintaN2	Cinta N2
	oCintaNi	Cinta Ni
	oEmisorPallets1	Emisor Pallets 1
	oEmisorPallets2	Emisor Pallets 2
	oEmisorPiezas	Emisor Piezas
	oEmpujadorPiston	Empujador Pistón
	oEsperaM	EsperaRotativaM
	oEsperaN	EsperaRotativaN
	oExtractorPallets	Salida Pallets
	oGiroRotativa	GiroRotativa
	oLuzArranqueCintas	Arranque Cintas (Luz)
	...mergenciaGeneral	Luz Emergencia General
	oLuzEmergLM	Emergencia LM
	oLuzEmergLN	EmergenciaLN
	oLuzMarchaLM	Marcha Paletizador M (Luz)
	oLuzMarchaLN	Marcha Paletizador N (Luz)
	oLuzParado	Paro completado
	oLuzParadoLM	Paro Completado PM
	oLuzParadoLN	Paro Completado PN
	oLuzSilencioEmerg	Silenciar Alarma (Luz)
	oLuzSolicitudParo	Solicitud Paro (Luz)
	oLuzSolicitudParoLM	Solicitud Paro PM (Luz)
	oLuzSolicitudParoLN	Solicitud Paro PN (Luz)

Figura 64: Mapeo de Sensores y Actuadores 2/3

oPM_Agarre	P&P M Agarre Pieza
oPM_GiroPinza	P&P M Giro Pinza (+)
oPM_LuzA	P&P M Luz Amarilla
oPM_LuzR	P&P M Luz Roja
oPM_LuzV	P&P M Luz Verde
oPM_SetPointX	P&P M Set Point X (V)
oPM_SetPointY	P&P M Set Point Y (V)
oPM_SetPointZ	P&P M Set Point Z (V)
oPM_StopEntrada	FinalCarreraM
oPN_Agarre	P&P N Agarre Pieza
oPN_GiroPinza	P&P N Giro Pinza (+)
oPN_LuzA	P&P N Luz Amarilla
oPN_LuzR	P&P N Luz Roja
oPN_LuzV	P&P N Luz Verde
oPN_SetPointX	P&P N Set Point X (V)
oPN_SetPointY	P&P N Set Point Y (V)
oPN_SetPointZ	P&P N Set Point Z (V)
oPN_StopEntrada	FinalCarreraN
oRodillosM1	Rodillos M1
oRodillosM2	Rodillos M2
oRodillosM3	Rodillos M3
oRodillosN1	Rodillos N1
oRodillosN2	Rodillos N2
oRodillosN3	Rodillos N3
oRodillosRotativaN	RodillosRotativa-
oRodillosRotativaP	RodillosRotativa+
oRodillosS1	Rodillos S1
oRodillosS2	Rodillos S2
oRodillosS3	Rodillos S3
oRodillosS4	Rodillos S4
oRodillosS5	Rodillos S5
oRodillosSC	Rodillos Salida Curva
oTopeCintaM	HojaParoM
oTopeCintaN	HojaParoN

Figura 65: Mapeo de Sensores y Actuadores 3/3

Tras esto, finaliza la configuración de Factory I/O. Tan solo queda hacer clic en el rótulo *CONNECT*, se indicará la conexión satisfactoria con un símbolo de *check* verde junto al desplegable de la barra superior y, volviendo a la escena del proyecto, puede darse comienzo a la simulación. Esto se realiza de forma simple, presionando el botón de *Play* en la barra superior y, tras ello, acercarse con la cámara al cuadro principal para dar la marcha a la línea y comprobar su correcto funcionamiento.

1.8. Resultados finales

1.8.1. Comprobación del correcto funcionamiento de la línea

Realizados los pasos anteriores, estaremos en situación de realizar una simulación, en la que se pueda comprobar que todo funciona correctamente. Hay diversos escenarios a analizar, el primero de ellos, el más obvio, es el funcionamiento en régimen estacionario.

Se ha dejado la línea en simulación durante un periodo de 15 minutos, durante los cuales no se ha producido ninguna incidencia, y se ha tomado una captura de pantalla del estado del proceso en dicho momento, que puede observarse en la Figura 66.



Figura 66: Captura del proceso tras 15 minutos de marcha

Entrando un poco más en detalle, hay varios elementos que resulta importante comprobar su correcto comportamiento. El primero de ellos, como no puede ser de otra manera, es la colocación de piezas en los paletizadores, siendo este uno de los objetivos principales de la línea. En las Figuras 67 y 68, se muestra una vista zenital de los paletizadores, que enseña cómo la colocación de las piezas se lleva a cabo de forma correcta.



Figura 67: Vista zenital Paletizador N (salida de pallet)



Figura 68: Vista zenital Paletizador M (salida de pallet)

Otro elemento que se debe controlar es el funcionamiento correcto de la máquina rotativa, en caso de que dos pallets lleguen al mismo tiempo, vigilando que la espera se realice de la forma esperada. Debido a que el hecho de que dos pallets lleguen a la vez a la rotativa es algo que no tiene por qué suceder, se ha recreado la situación con pallets vacíos para poder controlar el correcto funcionamiento del elemento. Las Figuras 69 y 70 muestran las esperas cuando llega primero un pallet de la línea de piezas normales y cuando el que llega primero es el de la línea de metálicas.



Figura 69: Espera Pallet piezas metálicas



Figura 70: Espera pallet piezas normales

Finalmente, se debe comprobar la efectividad de los métodos de seguridad activa empleados. Pueden dividirse en 3 elementos y cada uno tiene sus funcionalidades concretas, por lo que se van a estudiar por separado.

- Seta de emergencia general: Trabaja como un contacto NC y pulsarla provoca la apertura del circuito, por lo que todos los procesos activos se pausan. Al tratarse de una emergencia general, se activa la alarma, que puede silenciarse con el botón del cuadro “Silenciar Alarma”, y también se enciende la luz de emergencia, la cual no se apaga hasta haber resuelto la emergencia.

Para volver al trabajo, se debe desbloquear la seta ya que es un pulsado con enclavamiento, y pulsar el botón de marcha “Arranque Cintas”. Todo este proceso se ha plasmado en las Figuras 71, 72 y 73.



Figura 71: El proceso se para al pulsar el botón de emergencia general



Figura 72: Se activan la luz y la alarma de emergencia



Figura 73: Pulsador de Silencio de Alarma activo (apaga alarma, mantiene luz)

Es importante remarcar, que el botón de emergencia puede ser pulsado en cualquier momento, incluso cuando se ha solicitado una parada de la línea, y el efecto será el mismo en todos los casos. No obstante, en el caso de ser pulsado durante el proceso de parada, tras reestablecer la marcha, será necesario solicitar la parada de nuevo.

- Setas de emergencia de los paletizadores: Su funcionamiento es similar a de la seta de emergencia general, con la diferencia de que sólo se para el trabajo del paletizador cuya seta de emergencia sea pulsada (el trabajo en el otro continuará hasta que se llene la línea de piezas). En este caso, no se activa ni la alarma ni la luz de emergencia general, sino que se enciende un led de advertencia en el panel del paletizador, y la torre de luces colocada en el almacén del paletizador en cuestión cambia a color ámbar. Se reanuda el trabajo desanclando el pulsador de emergencia y dando un pulso de marcha en el paletizador en cuestión. Las Figuras 74 y 75 muestran los detalles del proceso.



Figura 74: Emergencia en Paletizador M. Proceso parado y luces de advertencia



Figura 75: El otro paletizador sigue su marcha (luces verdes)

- Puertas de Seguridad: Un total de 5 en toda la línea de producción, colocadas estratégicamente para permitir el acceso a todas las zonas de la misma. Su funcionamiento recae en la parada automática de la línea en el momento de apertura de cualquiera de las 5, puesto que están destinadas a permitir a los operarios realizar labores de mantenimiento, reparación y limpieza, para lo cual las máquinas deben estar paradas. Para reanudar la marcha, se comprueba que todas las puertas están cerradas y se requiere un pulso en el botón de marcha “Arranque Cintas”. Durante el tiempo en que alguna puerta está abierta, la luz de emergencia está activa, pero no así la alarma. En la Figura 76 se muestra la parada del proceso con una puerta abierta.

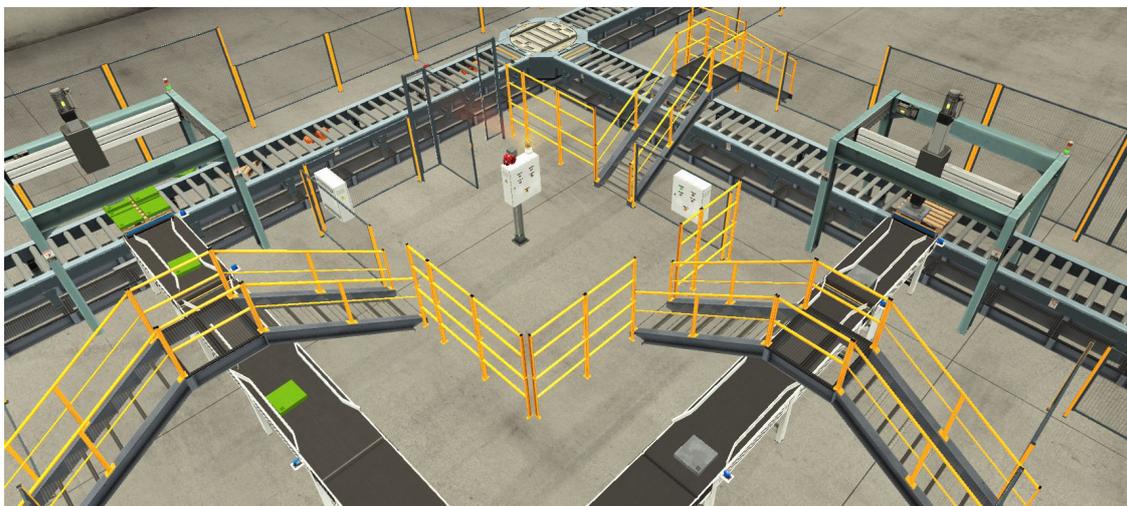


Figura 76: Paro de la línea de paletizado al abrir una puerta.

De esta forma queda completamente comprobado el correcto funcionamiento de la línea de paletizado, siendo el último paso verificar que se realiza correctamente el proceso de paro de la línea.

Aunque puede realizarse el paro de uno de los paletizadores en exclusiva (lo cual dejará el otro en funcionamiento hasta que se llene la línea de piezas del otro tipo) lo más habitual será realizar un paro general, mediante el pulsador “Solicitud Paro” del cuadro general. Este botón fuerza a ambos paletizadores a entrar en la etapa de paro, que consiste en terminar el pallet actual (llenarlo con 10 piezas) para luego entrar en la etapa de reposo. Cuando el último paletizador entra en reposo, se activa un temporizador que permite a los pallets abandonar la línea por la línea de salida y, tras esto, todos los rodillos transportadores se paran, logrando un paro completo.

Las Figuras 77, 78 y 79 muestran el proceso de paro.



Figura 77: Botón de Solicitud de Paro pulsado (indicador luminoso)



Figura 78: Ambos paletizadores terminan los últimos pallets. Entran en reposo.



Figura 79: Paro completo tras la extracción de todos los pallets de la línea

NOTA GENERAL: Factory I/O dispone de un botón de *Pause* y otra funcionalidad para acelerar o decelerar el ritmo de la simulación. Debido a que se trabaja con temporizadores, y que estos no se ven afectados por estas funciones de Factory I/O, utilizarlas puede llevar a un funcionamiento incorrecto de la línea.

1.8.2. Posibles cambios en la línea y su implementación

Llegado al punto en que la línea funciona correctamente, se han planteado diferentes escenarios que podrían darse en una industria real y a los que puede adaptarse la programación realizada con ligeros cambios.

- **Solo llegan piezas de un tipo, trabajar con un paletizador:** Puede darse el caso que, durante un día, semana, o similar, la empresa sólo produzca piezas de un tipo. Lo ideal en este caso sería utilizar el paletizador asignado a cada tipo de pieza, dejando el otro en estado de paro.

No obstante, sería posible utilizar el Paletizador M para piezas normales, lo cual requeriría un cambio del sensor inductivo previo al empujador por uno capacitivo (que sí sería capaz de detectar las piezas de plástico).

Del mismo modo, también se podría utilizar el Paletizador N para piezas metálicas. Bastaría con desconectar el empujador o, de no ser posible, retirar o desconectar temporalmente el sensor inductivo, de forma que las piezas metálicas pasarían de largo hacia el Paletizador N.

- **Solo llegan piezas de un tipo, trabajar con ambos paletizadores:** Si la producción es de piezas de un solo tipo, pero muy alta, puede ser interesante trabajar con ambos paletizadores a la vez. Esto implica un cambio mayor, afectando ya al código de programación, puesto que lo lógico sería que el empujador actúe de forma que deje pasar una pieza y empuje la siguiente.

Si las piezas son de plástico, el detector previo al empujador deberá ser capacitivo, como se ha explicado antes, mientras que si la producción es de piezas metálicas, el detector por inductivo es válido.

Para modificar el código de forma que el empujador actúe en 1 de cada 2 piezas, se ha pensado la siguiente opción: crear un contador CTU, que sume 1 a una variable auxiliar (ContPiezas, de tipo INT) cada vez que se detecte un flanco en el sensor previo al empujador (Figura 80). En la transición entre la Etapa 0 y la Etapa 1 del Grafset 3 (Empujador) añadir, sin eliminar las condiciones de transición por defecto, un operador AND entre el valor 16#00001 y ContPiezas que devuelva el resultado en una nueva variable auxiliar (Resultado, de tipo INT) y seguidamente colocar un operador *Equals* (EQ) con Resultado y 16#00001 como entradas (Figura 81).

De esta forma, el empujador actuará una vez sí y otra no, pudiendo ambos paletizadores trabajar con el mismo tipo de pieza. Importante en el caso de cambiar a un sensor capacitivo, cambiar las variables y temporizadores asociados en el código.

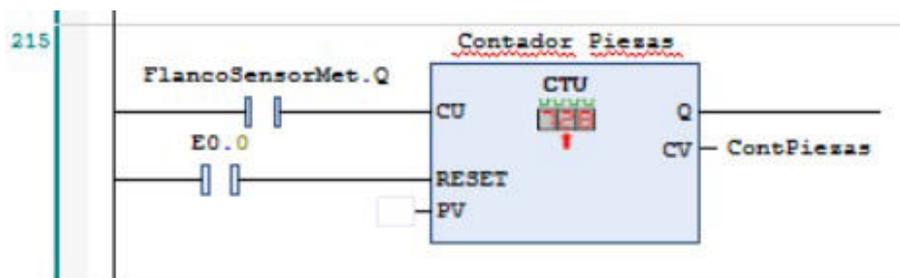


Figura 80: Definición del contador de piezas

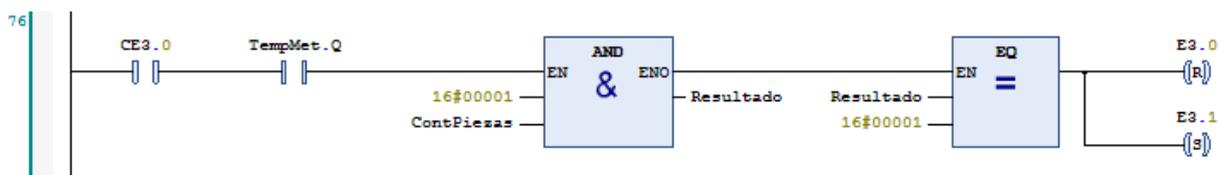


Figura 81: Cambio en la transición entre E3.0 y E3.1

- **Cambiar el número de piezas por pallet:** puede surgir la necesidad de aumentar o reducir el número de piezas que son depositadas en cada pallet. Este trabajo puede ser algo laborioso, pero si se trata de un cambio que va a persistir en el tiempo, se puede realizar sin problema.
 - **Menos de 10 piezas por pallet:** El primer paso sería cambiar al valor deseado las variables internas PiezasPalletM y PiezasPalletN, teniendo en cuenta que se debe introducir el número de piezas deseado -1.

El siguiente paso es opcional y consiste en editar los puntos en que se depositan las piezas para que dicha colocación se realice de forma uniforme. Para ello, se debe modificar el código mostrado al final del ANEXO IX (página 127-128) indicando los nuevos puntos en los que emplazar cada pieza (se pueden dejar los puntos que sobren puesto que al modificar las variables el número de piezas total en el pallet, sólo se leerá hasta el valor señalado)

- **Más de 10 piezas por pallet:** El procedimiento es muy similar, pero se ha separado puesto que requiere una modificación obligatoria del código de colocación de las piezas. El primer paso es idéntico, es decir, se deben modificar las variables internas de número de piezas por pallet al valor deseado (número de piezas – 1).

La diferencia radica en que, al tratarse de un número de piezas mayor al configurado por defecto, el código del ANEXO IX (página 100-101) debe modificarse y ampliarse, para que la colocación de las piezas sea la correcta. Con tal de lograrlo, se deben añadir tantos bloques de coordenadas (*SetPoint X, Y y Z*) como piezas quieran añadirse al pallet.

Las coordenadas se deberán obtener mediante un procedimiento de prueba en Factory I/O, que permite forzar los valores de las variables al deseado por el usuario, hasta alcanzar la posición deseada. Cuando esto se logra, se anotan las coordenadas y se introducen en el código en un nuevo bloque (con el número de la pieza que irá en dicha posición menos uno).

- **Espera de los pallets en la salida:** El proceso ha sido pensado de forma que el pallet continuaría hacia otra fase de producción tras atravesar el tramo de salida, que está en constante movimiento. Se podría dar el caso de que esta sea la última fase de la línea de producción, y que los pallets necesiten ser transportados a otro lugar (ya sea otra línea o al almacén) con una carretilla elevadora o similar.

Para ello, la cinta de salida deberá esperar a que se retire el pallet. Esto se podría lograr instalando sensores (que podrían ser de tipo capacitivo, difusos o de barrera) en el final de cada tramo de la línea de salida, de forma que los actuadores de dichos transportadores estén condicionados a la detección de estos.

De esta forma podrían acumularse algunos pallets completos hasta que sean extraídos de forma manual, incluso se podría considerar dividir la línea de salida en tramos más cortos para permitir la acumulación de más pallets completos, si se considera necesario por el tipo de producción a realizar.



Todo esto no supone un trabajo excesivo en la modificación del código, puesto que solamente bastaría con definir las variables de los nuevos sensores y añadirlos como contacto NC a las condiciones de los actuadores de la línea de salida. En el caso de querer introducir tramos más cortos simplemente sería necesario sustituir las variables antiguas por las nuevas y añadir las líneas de código según los nuevos tramos colocados.

De esta forma se han definido y planteado posibles escenarios alternativos para la línea de producción, que aportan una mayor flexibilidad al proyecto de cara a poder implementarse a futuro en una línea real, así como realizar su implementación sin necesidad de modificar en exceso el código de programación.

1.9. Viabilidad técnica

La viabilidad técnica del presente proyecto está sujeta a la disponibilidad de los programas y complementos utilizados en el desarrollo del trabajo. Estos son:

- **Factory I/O:** Diseño del proceso y simulación del funcionamiento del mismo tras la programación del control automático. Para uso exclusivo enfocado en simulaciones mediante el protocolo explicado en el presente proyecto, la versión *Factory I/O Modbus & OPC Edition* será suficiente.
- **CODESYS V3.5 SP18 Patch 2:** Utilizado en la programación del control automático del proceso diseñado.
 - **CODESYS Control Win SysTray:** Complemento necesario para utilizar el ordenador como PLC Virtual.
 - **CODESYS Gateway SysTray:** Complemento necesario para la comunicación entre CODESYS y Factory I/O.

Estos programas se encuentran disponibles para su descarga en sus respectivas páginas web (consultar bibliografía), y en los casos específicos de los complementos, se ha hecho hincapié en cómo instalarlos correctamente a lo largo del trabajo. Así mismo, más allá de su instalación, se debe disponer de un ordenador que permita ejecutar ambos programas. Los requisitos son:

- **Factory I/O:** Para que el programa funcione correctamente se necesita una versión de Windows 7 SP1 o superior (compatible con Windows 8 y 10), la CPU debe soportar la instrucción SSE2, y se debe disponer de una tarjeta gráfica NVIDIA o AMD posterior a 2006, o bien, Intel posterior a 2012.
- **CODESYS:** Para un correcto funcionamiento se recomienda un procesador a una velocidad de reloj de 2.5 GHz, dispositivo con 8GB de RAM, 12GB libres en el disco duro y la versión de 64 bit de Windows 7/8/10.

Cumpliendo todos los requisitos mencionados, y con la disponibilidad del software en el mercado, el proyecto se considera viable técnicamente.

1.10. Viabilidad económica

La viabilidad económica de este proyecto viene determinada por el ahorro que introduce la realización de un proyecto en gemelo digital, tanto en coste económico como temporal, respecto a un proceso tradicional, en el que la puesta a punto de las máquinas se realiza posteriormente a su montaje en planta y los tiempos de puesta en marcha pueden llegar muy prolongados.

En primer lugar, será necesario conocer el coste económico y temporal que puede tener el proceso de puesta en marcha en planta utilizando el método tradicional, es decir, realizando el ajuste del control automático mediante ensayos una vez la máquina está correctamente montada.

Posteriormente, se comparará dicho coste con el que supondría realizar el diseño virtual y la programación del control automático de la línea en paralelo al montaje de la misma en planta, lo cual permite que el tiempo posterior de ensayo in situ sea mucho menor.

1.10.1. Ahorro temporal

Durante la estancia en prácticas en empresa ofrecida por la UJI se ha podido asistir a varios montajes de nuevas líneas, así como a puestas en marcha de maquinaria similar a la simulada en el presente proyecto. Siendo variable el tiempo requerido para el montaje y la puesta en marcha, se estima que, en promedio, se requieren unas 3 semanas (15 días laborables) para el montaje y unos 8 días (semana y media laborable) para la puesta a punto previa al arranque de la línea.

En caso de optar por la alternativa planteada, el grueso del trabajo de la puesta a punto se podría realizar durante el tiempo que los técnicos se encargan del montaje de la línea, siendo solamente un pequeño ajuste y revisión final necesarios previamente a la puesta en marcha definitiva.

Se estima que el proceso de diseño virtual con Factory I/O y programación del control automático con CODESYS por un solo ingeniero, así como las correspondientes simulaciones de prueba, costará alrededor de 10 días laborables (2 semanas) mientras que el tiempo requerido de ajuste en planta (donde serán necesarios el ingeniero y un técnico para posibles ajustes) se vería reducido a 3 días.

La Tabla 1 muestra un resumen de los párrafos anteriores.

Actividad	Método Tradicional	Alternativa Propuesta
Montaje línea	15 días	15 días
Diseño y programación virtual	-	10 días (paralelo al montaje)
Puesta en Marcha en planta	10 días	4 días

Tabla 1: Resumen coste temporal actividades

En resumen, el método tradicional tendrá una duración de alrededor de 25 días laborables (5 semanas) mientras que la alternativa propuesta en el presente proyecto reduce unos 6 días el tiempo de trabajo (siendo el cuello de botella el tiempo de montaje de la línea), hasta algo menos de 4 semanas de trabajo (19 días).

De esta forma se concluye que el proyecto es viable en términos de ahorro de tiempo de trabajo.

1.10.2. Ahorro económico

El método tradicional ocupa un técnico y un ingeniero durante el proceso de la puesta en marcha de la máquina por un tiempo mayor que la alternativa del gemelo virtual. No obstante, se debe tener en cuenta los costes de software y el trabajo en solitario del ingeniero durante el diseño y la programación del control automático, para determinar si el proyecto es viable en términos económicos.

Para el método tradicional, se tienen los costes que se muestran en la Tabla 2.

Actividad	Coste/h (€)	Horas
Técnico electromecánico	15	80
Ingeniero (programación PLC)	20	80
TOTAL	2.800€/montaje	

Tabla 2: Coste económico método tradicional

No obstante, al realizar las pruebas in situ, el método tradicional supone un mayor riesgo de accidentes con la maquinaria, que pueden derivar en costes adicionales de varios miles de euros (ya que la maquinaria como la utilizada en este proyecto tiene valor de decenas de miles de euros) y mayores retrasos en las puestas en marcha, puesto que muchas piezas suelen ser fabricadas a medida para determinados procesos y, de sufrir algún tipo de rotura, la obtención de un recambio puede ser difícil. El riesgo de accidente es mucho menor (lo cual también es positivo para la seguridad de los trabajadores) con la alternativa planteada.

La alternativa propuesta requiere, eso sí, el pago de las licencias de uso de Factory I/O y del complemento CODESYS Control Win SL (aunque podría considerarse trabajar con la versión demo si no se requieren simulaciones de prueba de tiempo mayor a 2 horas), pero por el contrario reduce el tiempo que los operarios pasarán en planta. La Tabla 3 muestra los costes de este proceso.

Actividad	Coste/h (€)	Horas
Factory I/O versión OPC	Pago único: 395€	
CODESYS Control Win SL (opcional)	Pago único: 420€	
Técnico electromecánico	15	32
Ingeniero (diseño y programación)	20	112
TOTAL	2.720€/montaje + 815€ inversión inicial	

Tabla 3: Coste económico método propuesto



Como se puede observar, el coste entre ambos métodos es muy similar, siendo ligeramente superior el coste de la alternativa en el inicio, pero ofreciendo una mayor rentabilidad a partir de los 10 montajes, momento en el que ambas opciones se igualan y comienza a ser más económico optar por la alternativa planteada. Por ello, se deduce que, a medio-largo plazo, el proyecto es viable económicamente.

No obstante, hay que tener en cuenta que la opción del desarrollo del gemelo digital reduce la probabilidad de accidentes durante la puesta a punto final y, consecuentemente, también mitiga los riesgos para la seguridad de los trabajadores, así como el hecho de la reducción de tiempo de alrededor de una semana de trabajo mencionado anteriormente. Por ello, se llega a la conclusión de que la alternativa planteada ofrece una buena viabilidad económica en su conjunto.

1.11. Conclusiones

En el presente proyecto se ha presentado una alternativa a la forma de trabajo tradicional en cuanto a la programación y puesta a punto de maquinaria industrial mediante el diseño de un gemelo virtual de un proceso de paletizado y la programación de su control automático. Se ha evaluado las mejoras en tiempo que se consiguen en la puesta en marcha de una máquina respecto al procedimiento usual, así como el ahorro económico que puede suponer a largo plazo utilizar este sistema.

Este trabajo puede abrir la puerta a que empresas se interesen en software como Factory I/O para realizar simulaciones en gemelos virtuales 3D de sus máquinas de forma anticipada a realizar dichas pruebas en la maquinaria real, con la consecuente reducción de riesgos de accidentes (tanto con la máquina como hacia los técnicos) y los altos costes derivados de los mismos.

Personalmente, este trabajo me ha permitido consolidar y ampliar mis conocimientos en el ámbito de la Ingeniería en Sistemas Automáticos y descubrir una útil y potente herramienta como es Factory I/O. Desde un inicio estuve encantado de dirigir el trabajo hacia esta área de la Ingeniería y el hecho de poder enfocarlo a esta temática final, que en mi opinión es algo original sobre lo que se ha tratado poco en este tipo de trabajos, como son los gemelos digitales de procesos industriales, me ha hecho llevar el trabajo con mucho interés hacia el tema.

1.12. Bibliografía

El software necesario para llevar a cabo el proyecto puede encontrarse en las siguientes direcciones web:

CODESYS Group. CODESYS Development System V3. Versión 3.5.18.20.

Disponible en: <https://store.codesys.com/en/engineering/codesys.html>

CODESYS Group. CODESYS Control Win SL. Versión 3.5.18.20.

Disponible en: <https://store.codesys.com/en/codesys-control-win-sl-bundle.html>

CODESYS Group. CODESYS Edge Gateway for Windows. Versión 3.5.18.20.

Disponible en: <https://store.codesys.com/en/codesys-edge-gateway-for-windows.html>

Realgames. Factory I/O Modbus & OPC Edition. Versión 2.5.1.

Disponible en: <https://factoryio.com/buy>

Omegon Teachware. OFT2 Omegon Fluid Technology Grafcet. Versión 2.0.9.2.

Disponible en: <https://en.omesim.com/download-2/>

Asimismo, las siguientes fuentes han sido consultadas en algún momento, como apoyo para la realización de este trabajo:

CODESYS. CODESYS Online Help.

Disponible en: <https://www.helpme-codesys.com/>

Factory I/O. Setting up CODESYS OPC UA (SP17 or higher).

Disponible en: <https://docs.factoryio.com/tutorials/codesys/setting-up/codesys-opc-ua-sp17/>

Factory I/O. Manual – Parts. <https://docs.factoryio.com/manual/parts/>

Peterson, Jon. Control Automation. Understanding PLC Program Commands: Comparison instructions.

Disponible en: <https://control.com/technical-articles/understanding-plc-program-commands-comparison-instructions/>

Logitek Team. KEPServerEX. 5 ventajas de OPC UA respecto a OPC DA.

Disponible en: [https://www.kepserverexopc.com/5-ventajas-de-opc-ua-respecto-a-opc-da/...](https://www.kepserverexopc.com/5-ventajas-de-opc-ua-respecto-a-opc-da/)

Satoshi. Opiron. De Modbus a OPC UA.

Disponible en: <https://www.opiron.com/de-modbus-a-opc-ua/>



Wikipedia. OPC Unified Architecture.

Disponible en: https://en.wikipedia.org/wiki/OPC_Unified_Architecture

Wikipedia. OPC Data Access.

Disponible en: https://en.wikipedia.org/wiki/OPC_Data_Access

Wikipedia. Modbus.

Disponible en: <https://es.wikipedia.org/wiki/Modbus>

Wikipedia. IEC 61131-3.

Disponible en: https://es.wikipedia.org/wiki/IEC_61131-3

Romero Pérez, Julio Ariel. Apuntes de la asignatura ET1030 – Automatización Industrial. *Introducción a la Norma IEC-61131-3 para la Programación de Sistemas de Automatización Industrial.*

Romero Pérez, Julio Ariel. Apuntes de la asignatura ET1030 – Automatización Industrial. *Modelado mediante GRAFCET.*

Sanchís Llopis, Roberto. Apuntes de la asignatura ET1023 – Sistemas Automáticos. *Modelado de sistemas de eventos discretos.*



ANEXOS



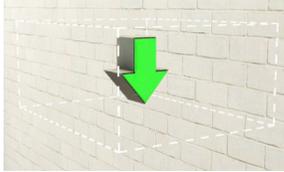
Índice de Anexos:

ANEXO I.	Elementos del proceso y sus características.....	73
ANEXO II.	Listado de variables globales.....	81
ANEXO III.	Grafcet 0 – Cuadro General.....	89
ANEXO IV.	Grafcet 1 – Control de Cintas y Rodillos.....	91
ANEXO V.	Grafcet 2 – Control de la Máquina Rotativa.....	93
ANEXO VI.	Grafcet 3 – Control del Empujador de Piezas.....	95
ANEXO VII.	Grafcet 4 y 5 – Control de los Paletizadores.....	97
ANEXO VIII.	Grafcet 6 y 7 – Cuadro de los Paletizadores.....	99
ANEXO IX.	Código en Ladder Diagram.....	101



ANEXO I. Elementos del proceso y sus características

En la Tabla 4 se incluyen los elementos utilizados en el diseño virtual de la línea de paletizado, así como la configuración que se debe asignar a cada uno de ellos, para realizar la programación como se indica en este proyecto. Todas las señales de sensores y actuadores son booleanas (*TRUE/FALSE*) salvo que se indique lo contrario

ELEMENTO	SÍMBOLO	Ud.	Configuración
Emisor		3	<p><i>User-Defined</i>: El emisor puede ser activado y desactivado por el programa de control.</p> <p>1 Emisor configurado para emitir <i>Green raw material</i> y <i>Metal raw material</i>.</p> <p>2 emisores configurados para emitir <i>Square Pallet</i>.</p>
Extractor		1	<p><i>User-Defined</i>: El extractor puede ser activado y desactivado por el programa de control.</p> <p>Configurado para eliminar cualquier elemento.</p>
Cintas Transportadoras		1+2+4	<p><i>Digital</i>: El motor puede ser activado y desactivado por el programa de control (sentido y velocidad únicos)</p> <p>1 cinta de 6m de longitud</p> <p>2 cintas de 4m de longitud</p>

ELEMENTO	SÍMBOLO	Ud.	Configuración
			4 cintas de 2m de longitud
Alineadores tipo 2		1	Sin configuración
Alineadores tipo 3		13	Sin configuración
Empujador		1	Monoestable: Pistón de simple efecto y velocidad fija, con detector de final de carrera delante y detrás.

ELEMENTO	SÍMBOLO	Ud.	Configuración
Hojas de Parada		4	Configuración única: dispone de un actuador que al activarse levanta la hoja, impidiendo el paso del material.
Esquinas de metal		7	Sin configuración
Transportadores de Rodillos		7+4	Digital: El motor puede ser activado y desactivado por el programa de control (sentido y velocidad únicos) 7 transportadores de rodillos de 6m de longitud 4 transportadores de rodillos de 2m de longitud
Transportador de Rodillos Curvo		1	<i>Counterclockwise</i> : El motor puede ser activado y desactivado por el programa de control (en sentido contrario a las agujas del reloj y velocidad única)

ELEMENTO	SÍMBOLO	Ud.	Configuración
Rodillos de parada		2	Configuración única:
Máquina Rotativa		1	Monoestable: Actuador con giro en un único sentido (determinado por flechas) y que vuelve a su posición de reposo si se desactiva. Cuenta con rodillos que pueden girar en ambos sentidos, y sensores en ambas posiciones límite de giro así como a la entrada y salida de la plataforma.
Sensor Capacitivo		4	Digital: Indica la presencia de elementos, sin especificar distancia.
Sensor Difuso		6	Configuración única: Indica la presencia de sólidos

ELEMENTO	SÍMBOLO	Ud.	Configuración
Sensor Inductivo		1	Digital: Indica la presencia de elementos
Columna		1	Sin Configuración
Cuadro Eléctrico		3	Sin Configuración
Setas Emergencia		3	Configuración única: Normalmente Cerrados, pulsador con enclavamiento (no vuelve a su posición inicial tras ser pulsado)

ELEMENTO	SÍMBOLO	Ud.	Configuración
Pulsador Verde		3	<i>Momentary action:</i> Sensor que detecta mientras el botón se encuentra pulsado. Luz que puede estar encendida o apagada según necesidad.
Interruptor Amarillo		1	<i>Alternate Action:</i> Al pulsarlo se alterna entre ON / OFF. También dispone de luz que puede encenderse o apagarse según preferencia
Pulsador Rojo		3	<i>Momentary action:</i> Sensor que detecta mientras el botón se encuentra pulsado. Luz que puede estar encendida o apagada según necesidad.
Indicador Luminoso Amarillo		2	Configuración única: Luz encendida o apagada
Indicador Luminoso Rojo		3	Configuración única: Luz encendida o apagada

ELEMENTO	SÍMBOLO	Ud.	Configuración
Alarma de Emergencia		1	Configuración única: Sonido encendido o apagado
Pila de Luces		2	Configuración única: Luces Roja, Amarilla y Verde que se pueden encender y apagar individualmente.
Luz de Advertencia		1	Configuración única: Luz y lámpara rotatoria encendidas o apagadas.
Paletizador Pick & Place		2	<p><i>Analog:</i></p> <p>Sensores analógicos de posición en los 3 ejes (XYZ, variables tipo REAL). Rango de 0 a 10 V</p> <p>Sensores de detección de pieza bajo la pinza y de límite de giro de la pinza.</p> <p>Actuadores analógicos de posicionamiento XYZ (variables tipo REAL) Rango de 0 a 10 V</p>

ELEMENTO	SÍMBOLO	Ud.	Configuración
			Actuadores digitales de giro de la pinza y de succión de pieza.
Pasamanos Tamaño S		38	Color por defecto
Pasamanos Tamaño M		28	Color por defecto
Plataforma Tamaño S		110	Sin configuración
Plataforma Tamaño M		4	Sin configuración
Vallas de Seguridad Tipo S		32	Color por defecto
Vallas de Seguridad Tipo L		5	Color por defecto

ELEMENTO	SÍMBOLO	Ud.	Configuración
Vallas de Seguridad Tipo I		4	Color por defecto
Escaleras		8	Sin Configuración
Barandilla		16	Color por defecto
Puerta de Seguridad		5	Configuración única: sensor NC que se desactiva al abrir la puerta.

Tabla 4: Elementos del proceso: Características y Configuración

ANEXO II. Listado de variables globales

NOTA: Todos los nombres en CODESYS llevan el encabezado *FIO.* al ser llamados en el código de programación.

INPUTS - Sensores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
Grupo: Pulsadores Cuadro General			
Arranque Cintas	iArranqueCintas	BOOL	Pulsador de arranque del proceso en el cuadro general
Emergencia Todas Líneas	iEmergenciaTL	BOOL	Seta de emergencia en el cuadro general
Silencio Alarma	iSilencioAlarma	BOOL	Interruptor que activa/desactiva el sonido de la alarma
Solicitud Paro	iSolicitudParo	BOOL	Pulsador utilizado para solicitar el paro general del proceso
Grupo: Empujador Piezas Metálicas			
Sensor Metálicas	iSensorMetálicas	BOOL	Sensor inductivo encargado de detectar piezas metálicas
Empujador Delante	iEmpujadorDelante	BOOL	Informa que el pistón ha llegado a su máximo despliegue
Empujador Detrás	iEmpujadorDetras	BOOL	Informa que el pistón se encuentra completamente replegado
Grupo: Cuadro Paletizadores			
Marcha Línea M	iMarchaLM	BOOL	Pulsador de marcha del Paletizador M
Marcha Línea N	iMarchaLN	BOOL	Pulsador de marcha del Paletizador N
Parada Emergencia LM	iEmergenciaLM	BOOL	Seta de emergencia de la Línea M
Parada Emergencia LN	iEmergenciaLN	BOOL	Seta de emergencia de la Línea N



INPUTS - Sensores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
Solicitud Paro Línea M	iSolicitudParoLM	BOOL	Pulsador para solicitar el paro de la Línea M
Solicitud Paro Línea N	iSolicitudParoLN	BOOL	Pulsador para solicitar el paro de la Línea N
Grupo: Paletizador Piezas Metálicas			
P&P M Detección Pieza	iPM_DetecPieza	BOOL	Sensor integrado brazo del Palet.M. Detecta piezas próximas
P&P M Límite Giro Pinza	iPM_LimGiroPin	BOOL	Indica el límite del giro del brazo del Paletizador M
Sensor Pieza Final Línea M	iSensFinLM	BOOL	Indica que hay una pieza al final de la Línea M
Sensor Entrada M	iSensEntLM	BOOL	Indica pieza entrando a la zona de aproximación al PM
Pos. Pallet OK M	iPM_PalletOK	BOOL	Indica que el pallet ha llegado a la posición de paletizado
P&P M Position X (V)	iPM_PositionX	REAL	Indica la posición en eje X del brazo del PM (Rango 0-10 V)
P&P M Position Y (V)	iPM_PositionY	REAL	Indica la posición en eje Y del brazo del PM (Rango 0-10 V)
P&P M Position Z (V)	iPM_PositionZ	REAL	Indica la posición en eje Z del brazo del PM (Rango 0-10 V)
Grupo: Paletizador Piezas Normales			
P&P N Detección Pieza	iPN_DetecPieza	BOOL	Sensor del brazo del Paletizador N. Detecta piezas próximas
P&P N Límite Giro Pinza	iPN_LimGiroPin	BOOL	Indica el límite del giro del brazo del Paletizador N
Sensor Pieza Final Línea N	iSensFinLN	BOOL	Indica que hay una pieza al final de la Línea N
Sensor Entrada N	iSensEntLN	BOOL	Indica pieza entrando a la zona de aproximación al PN
Pos. Pallet OK N	iPN_PalletOK	BOOL	Indica que el pallet ha llegado a la posición de paletizado
P&P N Position X (V)	iPN_PositionX	REAL	Indica la posición en eje X del brazo del PN (Rango 0-10 V)

INPUTS - Sensores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
P&P N Position Y (V)	iPN_PositionY	REAL	Indica la posición en eje Y del brazo del PN (Rango 0-10 V)
P&P N Position Z (V)	iPN_PositionZ	REAL	Indica la posición en eje Z del brazo del PN (Rango 0-10 V)
Grupo: Máquina Rotativa			
PalletEntradaRotativaM	iPalletEntradaRM	BOOL	Indica pallet procedente de Línea M llegando a la Rotativa
PalletEntradaRotativaN	iPalletEntradaRN	BOOL	Indica pallet procedente de Línea N llegando a la Rotativa
PosEntradaRotativaM	iPosEntM	BOOL	Indica que la Rotativa se encuentra en posición de entrada LM
PosEntradaRotativaN	iPosEntN	BOOL	Indica que la Rotativa se encuentra en posición de entrada LN
SensorEntradaPalletRot	iEntPalletRot	BOOL	Indica pallet entrando a la Rotativa
SensorSalidaPalletRot	iSalPalletRot	BOOL	Indica pallet saliendo de la Rotativa
Grupo: Otros Sensores			
LíneaN Llena	iLN_Llena	BOOL	Indica que la Línea N está llena de piezas
LíneaM Llena	iLM_Llena	BOOL	Indica que la Línea M está llena de piezas
Puerta Cinta E y PM	iPuertaEyPM	BOOL	Indica puerta de seguridad con acceso a Cinta E y PM cerrada
Puerta Cinta E y PN	iPuertaEyPN	BOOL	Indica puerta de seguridad con acceso a Cinta E y PN cerrada
Puerta PM y Salida	iPuertaPM y S	BOOL	Indica puerta de seguridad con acceso a PM y salida cerrada
Puerta PN y Salida	iPuertaPN y S	BOOL	Indica puerta de seguridad con acceso a PN y salida cerrada
Puerta Interior	iPuertaInt	BOOL	Indica puerta de seguridad de acceso interior cerrada

Tabla 5: Listado de Variables Globales - Sensores



OUTPUTS - Actuadores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
Grupo: Cuadro General			
Arranque Cintas (Luz)	oLuzArranqueCintas	BOOL	Luz que indica marcha general (cintas y rodillos)
Paro Completado	oLuzParado	BOOL	Luz que indica paro completo de la línea (paletizadores incl.)
Solicitud Paro (Luz)	oLuzSolicitudParo	BOOL	Luz que indica que se ha solicitado un paro completo
Luz Emergencia General	oLuzEmergenciaGeneral	BOOL	Luz de emergencia tras activar la seta del cuadro general
Luz Silencio Alarma	oLuzSilencioEmerg	BOOL	Luz que indica que la alarma está silenciada
Alarma Emergencia General	oAlarmaEmergGeneral	BOOL	Alarma de emergencia tras activar la seta del cuadro general
Grupo: Cuadro Paletizadores			
Marcha Paletizador M (Luz)	oLuzMarchaLM	BOOL	Luz que indica marcha en el Paletizador M
Marcha Paletizador N (Luz)	oLuzMarchaLN	BOOL	Luz que indica marcha en el Paletizador N
Paro Completado PM	oLuzParadoLM	BOOL	Luz que indica paro completado del Paletizador M
Paro Completado PN	oLuzParadoLN	BOOL	Luz que indica paro completado del Paletizador N
Solicitud Paro PM (Luz)	oLuzSolicitudParoLM	BOOL	Luz que indica paro solicitado del Paletizador M
Solicitud Paro PN (Luz)	oLuzSolicitudParoLN	BOOL	Luz que indica paro solicitado del Paletizador N
Emergencia LM	oLuzEmergLM	BOOL	Luz que indica que se ha activado la seta de emergencia LM
Emergencia LN	oLuzEmergLN	BOOL	Luz que indica que se ha activado la seta de emergencia LN
Grupo: Cintas Transportadoras			
Cinta E	oCintaEntrada	BOOL	Cinta transportadora en la Entrada de piezas
Cinta M1	oCintaM1	BOOL	Cinta transportadora primer tramo Línea M
Cinta Mi	oCintaMi	BOOL	Cinta transportadora tramo intermedio Línea M
Cinta M2	oCintaM2	BOOL	Cinta transportadora tramo aproximación al Paletizador M

OUTPUTS - Actuadores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
Cinta N1	oCintaN1	BOOL	Cinta transportadora primer tramo Línea N
Cinta Ni	oCintaNi	BOOL	Cinta transportadora tramo intermedio Línea N
Cinta N2	oCintaN2	BOOL	Cinta transportadora tramo aproximación al Paletizador N
HojaParoM	oTopeCintaM	BOOL	Hoja de paro situada en la Línea M (entre Mi y M2)
HolaParoN	oTopeCintaN	BOOL	Hoja de paro situada en la Línea N (entre Ni y N2)
Grupo: Empujador			
Empujador Pistón	oEmpujadorPiston	BOOL	Actuador que ejecuta la salida del pistón del empujador
Grupo: Máquina Rotativa			
GiroRotativa	oGiroRotativa	BOOL	Motor de giro de la Máquina Rotativa
RodillosRotativa+	oRodillosRotativaP	BOOL	Motor de giro en sentido positivo rodillos rotativa
RodillosRotativa-	oRodillosRotativaN	BOOL	Motor de giro en sentido negativo rodillos rotativa
EsperaRotativaM	oEsperaM	BOOL	Rodillo de espera en la entrada a la rotativa desde Línea M
EsperaRotativaN	oEsperaN	BOOL	Rodillo de espera en la entrada a la rotativa desde Línea N
Grupo: Paletizador Piezas Metálicas			
P&P M Agarre Pieza	oPM_Agarre	BOOL	Activa la succión de la pieza por parte del brazo del PM
P&P M Giro Pinza (+)	oPM_GiroPinza	BOOL	Motor de giro de 90° de la pinza del PM
P&P M Luz Amarilla	oPM_LuzA	BOOL	Luz amarilla en la torre de luces de estado del PM
P&P M Luz Roja	oPM_LuzR	BOOL	Luz roja en la torre de luces de estado del PM
P&P M Luz Verde	oPM_LuzV	BOOL	Luz verde en la torre de luces de estado del PM
FinalCarreraM	oPM_StopEntrada	BOOL	Hoja de paro en el final de Línea M
P&P M Set Point X (V)	oPM_SetPointX	REAL	Coord. X de destino del brazo del PM (Rango 0-10V)



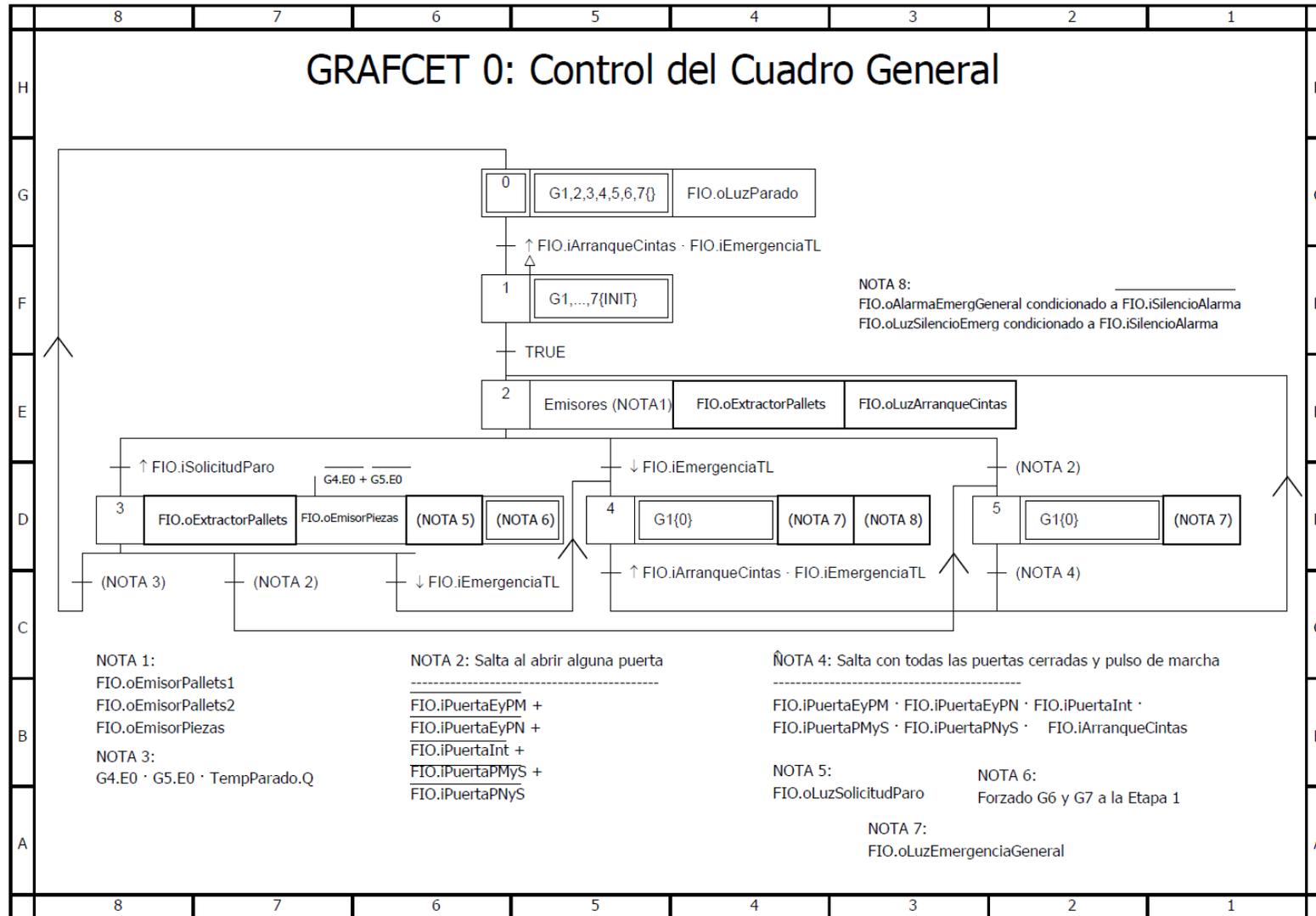
OUTPUTS - Actuadores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
P&P M Set Point Y (V)	oPM_SetPointY	REAL	Coord. Y de destino del brazo del PM (Rango 0-10V)
P&P M Set Point Z (V)	oPM_SetPointZ	REAL	Coord. Z de destino del brazo del PM (Rango 0-10V)
Grupo: Paletizador Piezas Normales			
P&P N Agarre Pieza	oPN_Agarre	BOOL	Activa la succión de la pieza por parte del brazo del PN
P&P N Giro Pinza (+)	oPN_GiroPinza	BOOL	Motor de giro de 90° de la pinza del PN
P&P N Luz Amarilla	oPN_LuzA	BOOL	Luz amarilla en la torre de luces de estado del PN
P&P N Luz Roja	oPN_LuzR	BOOL	Luz roja en la torre de luces de estado del PN
P&P N Luz Verde	oPN_LuzV	BOOL	Luz verde en la torre de luces de estado del PN
FinalCarreraN	oPN_StopEntrada	BOOL	Hoja de paro en el final de Línea N
P&P N Set Point X (V)	oPN_SetPointX	REAL	Coord. X de destino del brazo del PN (Rango 0-10V)
P&P N Set Point Y (V)	oPN_SetPointY	REAL	Coord. Y de destino del brazo del PN (Rango 0-10V)
P&P N Set Point Z (V)	oPN_SetPointZ	REAL	Coord. Z de destino del brazo del PN (Rango 0-10V)
Grupo: Rodillos Transportadores			
Rodillos M1	oRodillosM1	BOOL	Motor del Transportador de Rodillos inicio Línea M
Rodillos M2	oRodillosM2	BOOL	Motor del Transportador de Rodillos intermedio Línea M
Rodillos M3	oRodillosM3	BOOL	Motor del Transportador de Rodillos final Línea M
Rodillos N1	oRodillosN1	BOOL	Motor del Transportador de Rodillos inicio Línea N
Rodillos N2	oRodillosN2	BOOL	Motor del Transportador de Rodillos intermedio Línea N
Rodillos N3	oRodillosN3	BOOL	Motor del Transportador de Rodillos final Línea N
Rodillos S1	oRodillosS1	BOOL	Motor del Transportador de Rodillos Línea de Salida 1
Rodillos S2	oRodillosS2	BOOL	Motor del Transportador de Rodillos Línea de Salida 2



OUTPUTS - Actuadores			
Nombre en Factory I/O	Nombre en CODESYS	TIPO	Descripción
Rodillos S3	oRodillosS3	BOOL	Motor del Transportador de Rodillos Línea de Salida 3
Rodillos SC	oRodillosSC	BOOL	Motor del Transportador de Rodillos Línea de Salida Curva
Rodillos S4	oRodillosS4	BOOL	Motor del Transportador de Rodillos Línea de Salida 4
Rodillos S5	oRodillosS5	BOOL	Motor del Transportador de Rodillos Línea de Salida 5
Grupo: Emisores - Extractores			
Emisor Piezas	oEmisorPiezas	BOOL	Activación de la llegada de piezas a la Cinta de Entrada
Emisor Pallets 1	oEmisorPallets1	BOOL	Activación de la llegada de pallets a la Línea M
Emisor Pallets 2	oEmisorPallets2	BOOL	Activación de la llegada de pallets a la Línea N
Salida Pallets	oExtractorPallets	BOOL	Activación del extractor de pallets completos.

Tabla 6: Listado de Variables Globales - Actuadores

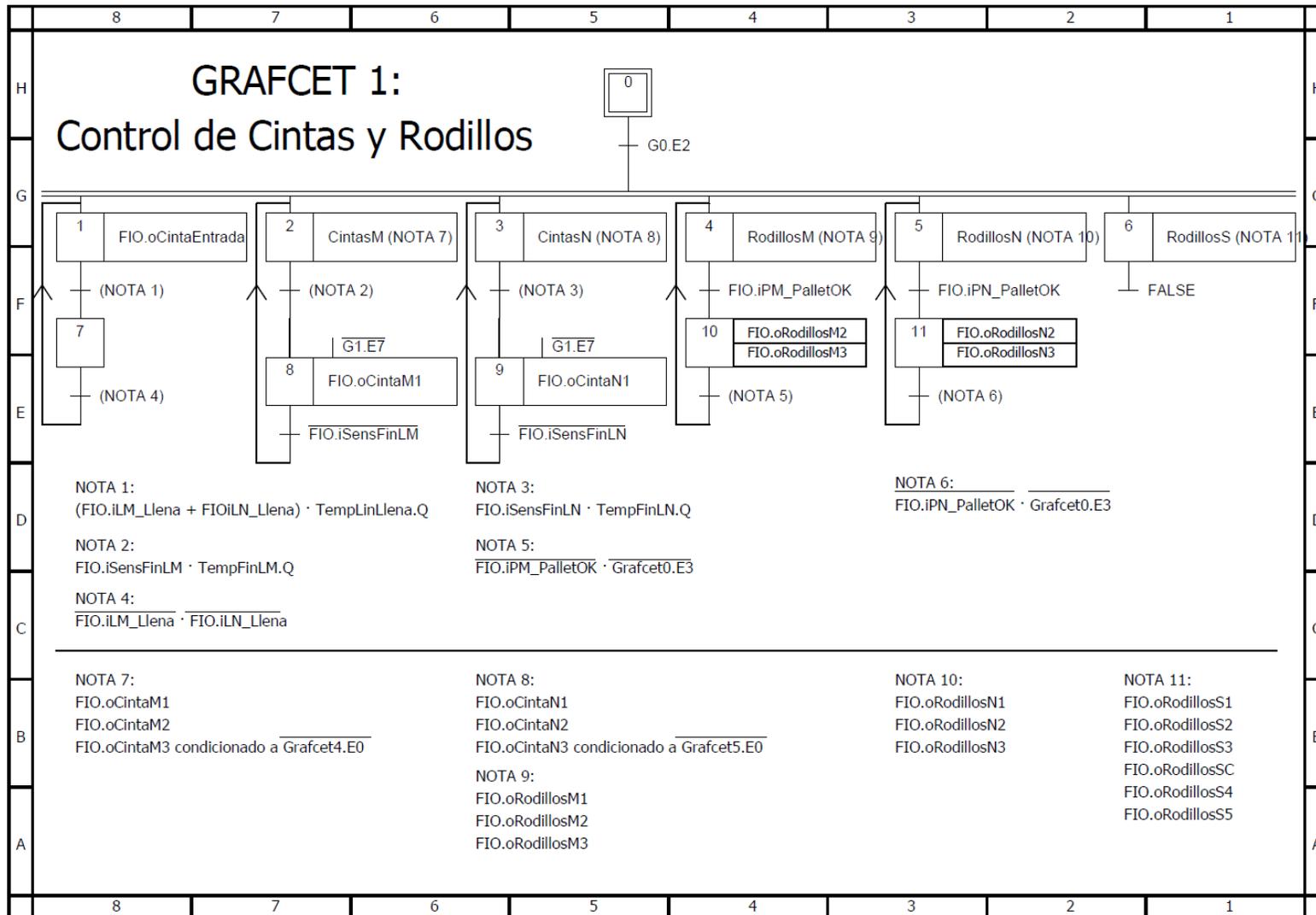
ANEXO III. Grafcet 0 – Cuadro General







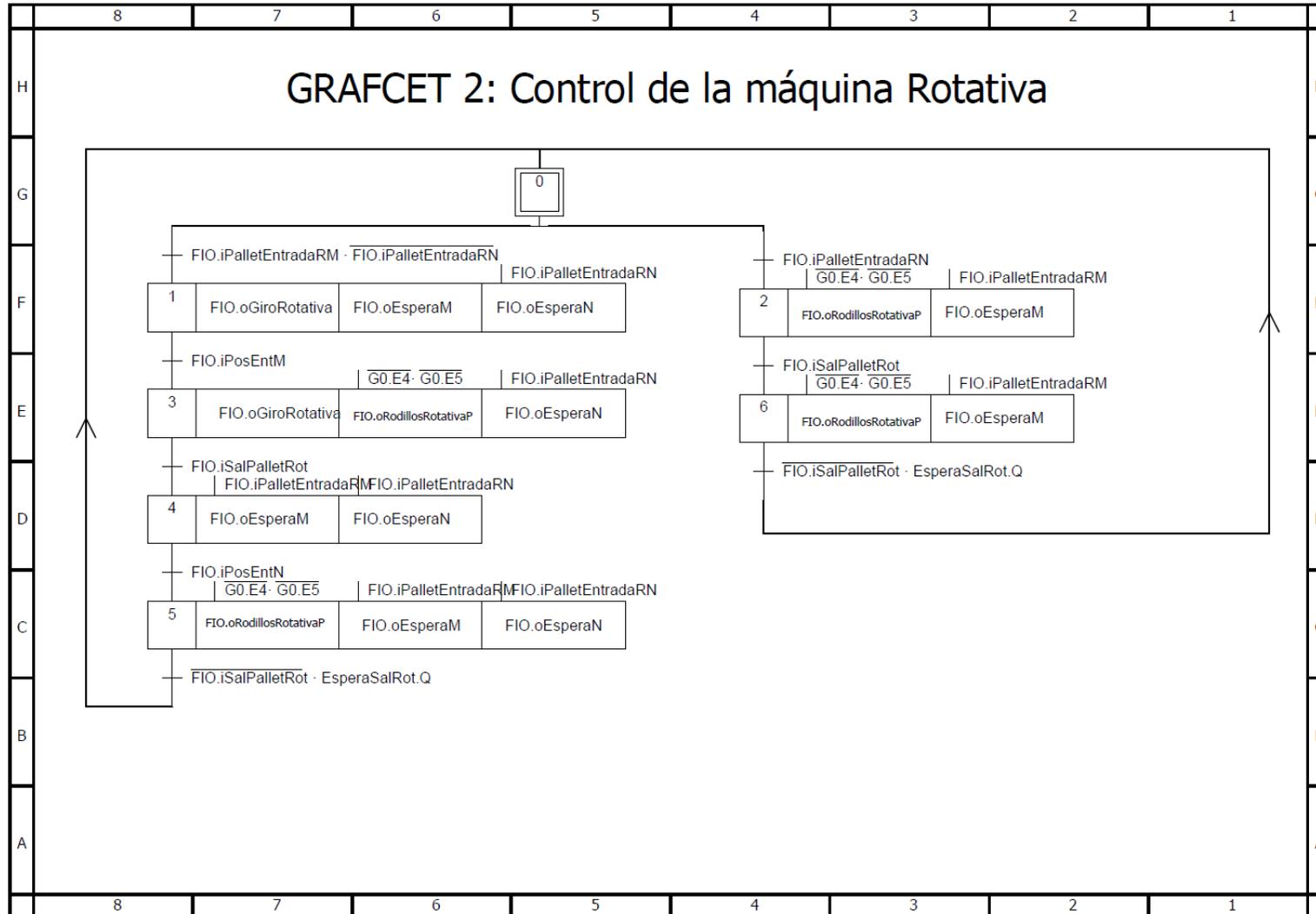
ANEXO IV. Grafcet 1 – Control de Cintas y Rodillos







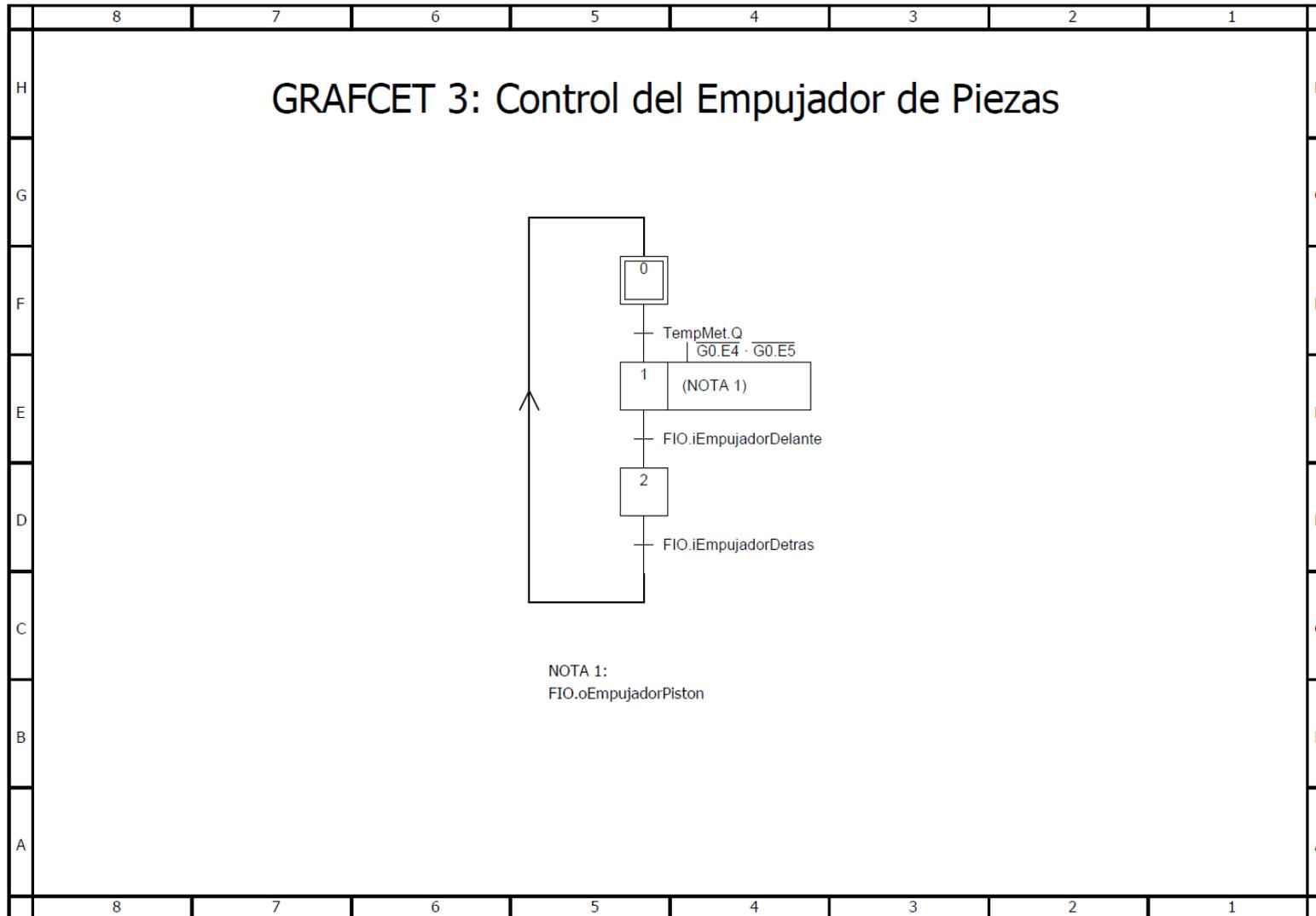
ANEXO V. Grafcet 2 – Control de la Máquina Rotativa







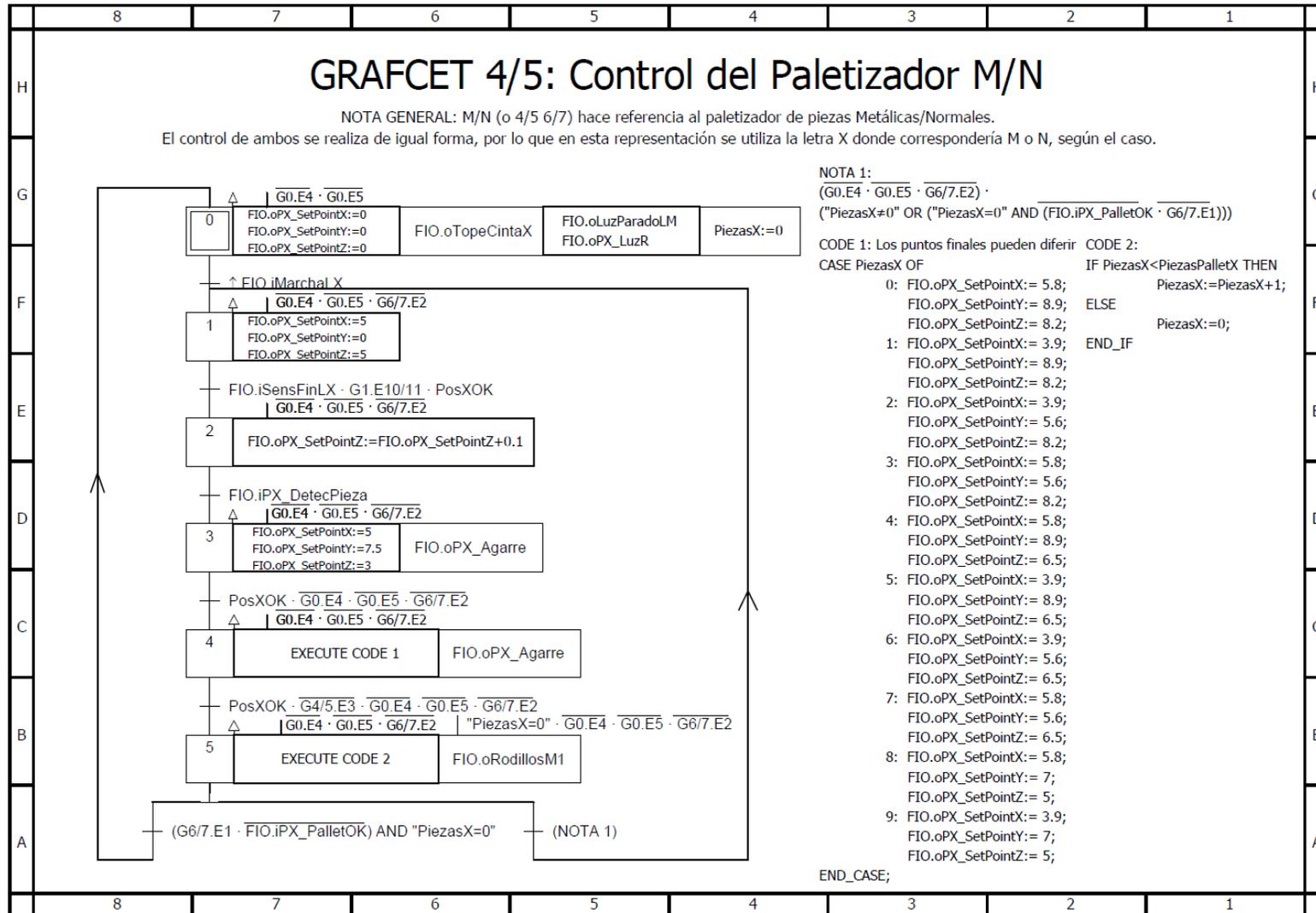
ANEXO VI. Grafcet 3 – Control del Empujador de Piezas







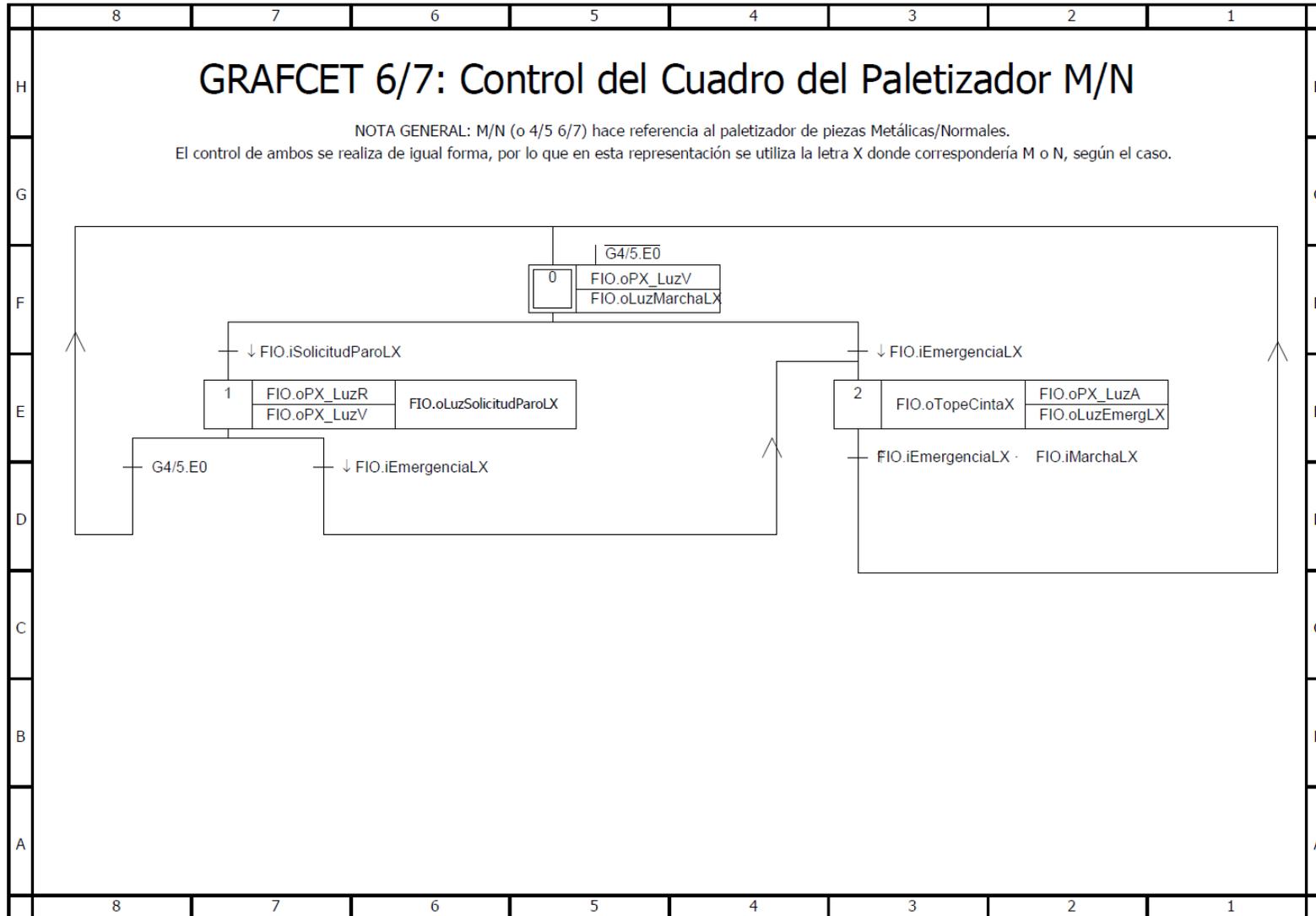
ANEXO VII. Grafcet 4 y 5 – Control de los Paletizadores







ANEXO VIII. Grafcet 6 y 7 – Cuadro de los Paletizadores







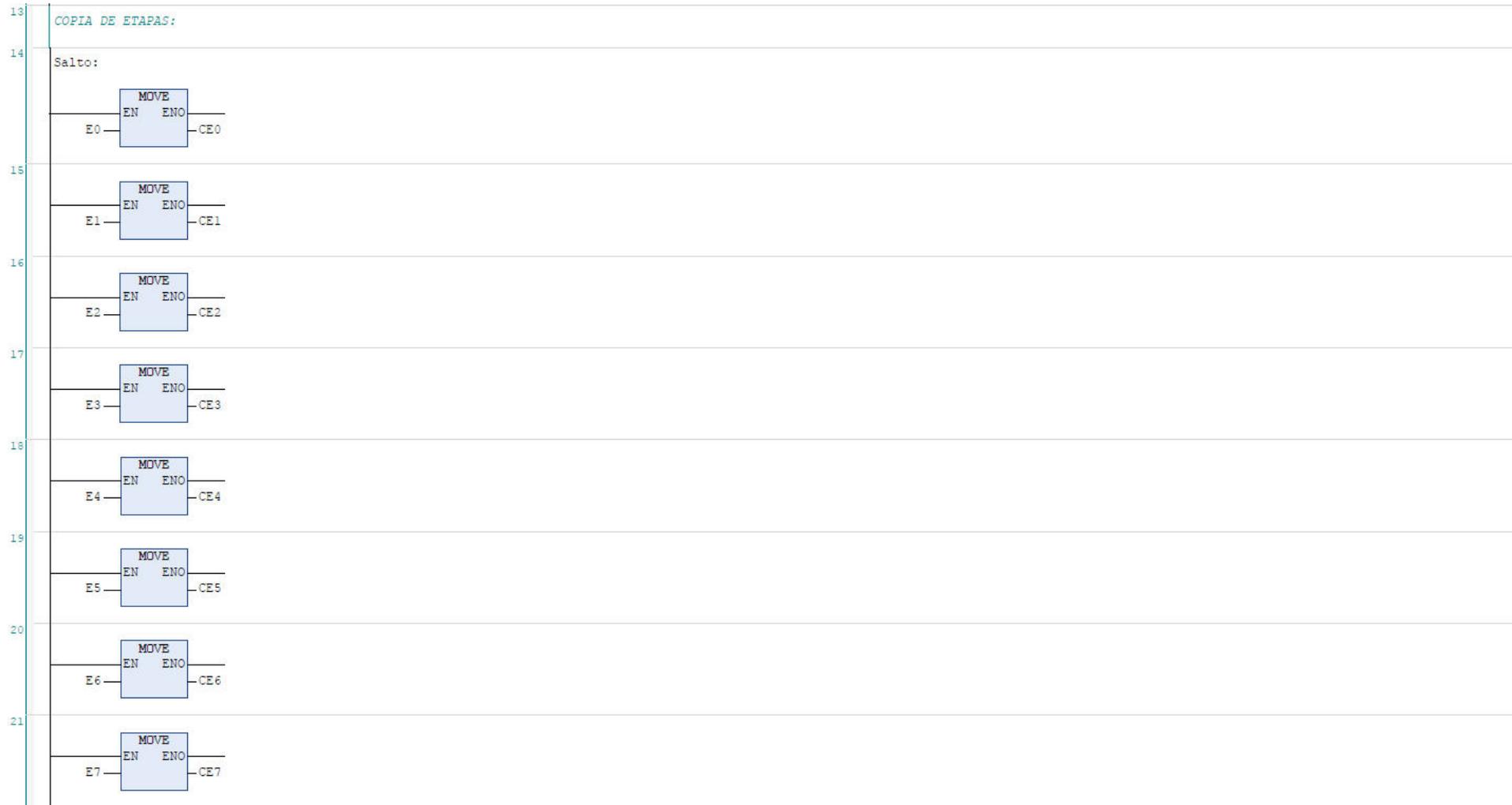
ANEXO IX. Código en Ladder Diagram

Líneas de 1 a 12: Inicialización y Primer escaneo





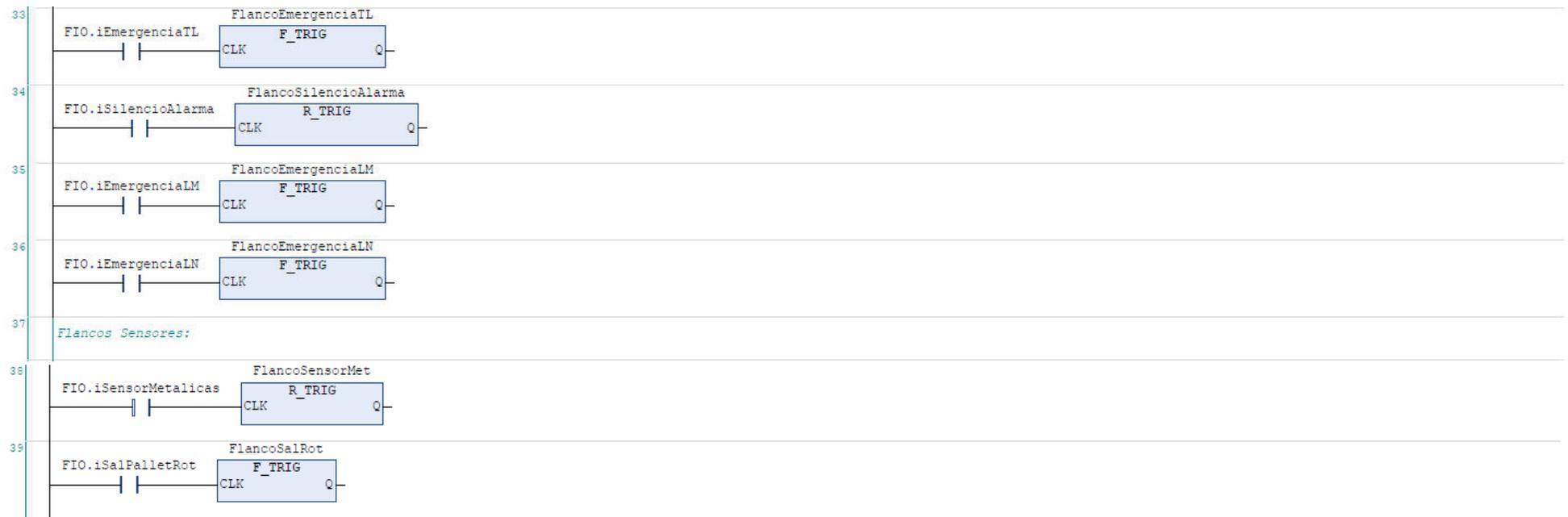
Líneas 13 a 21: Copia de etapas





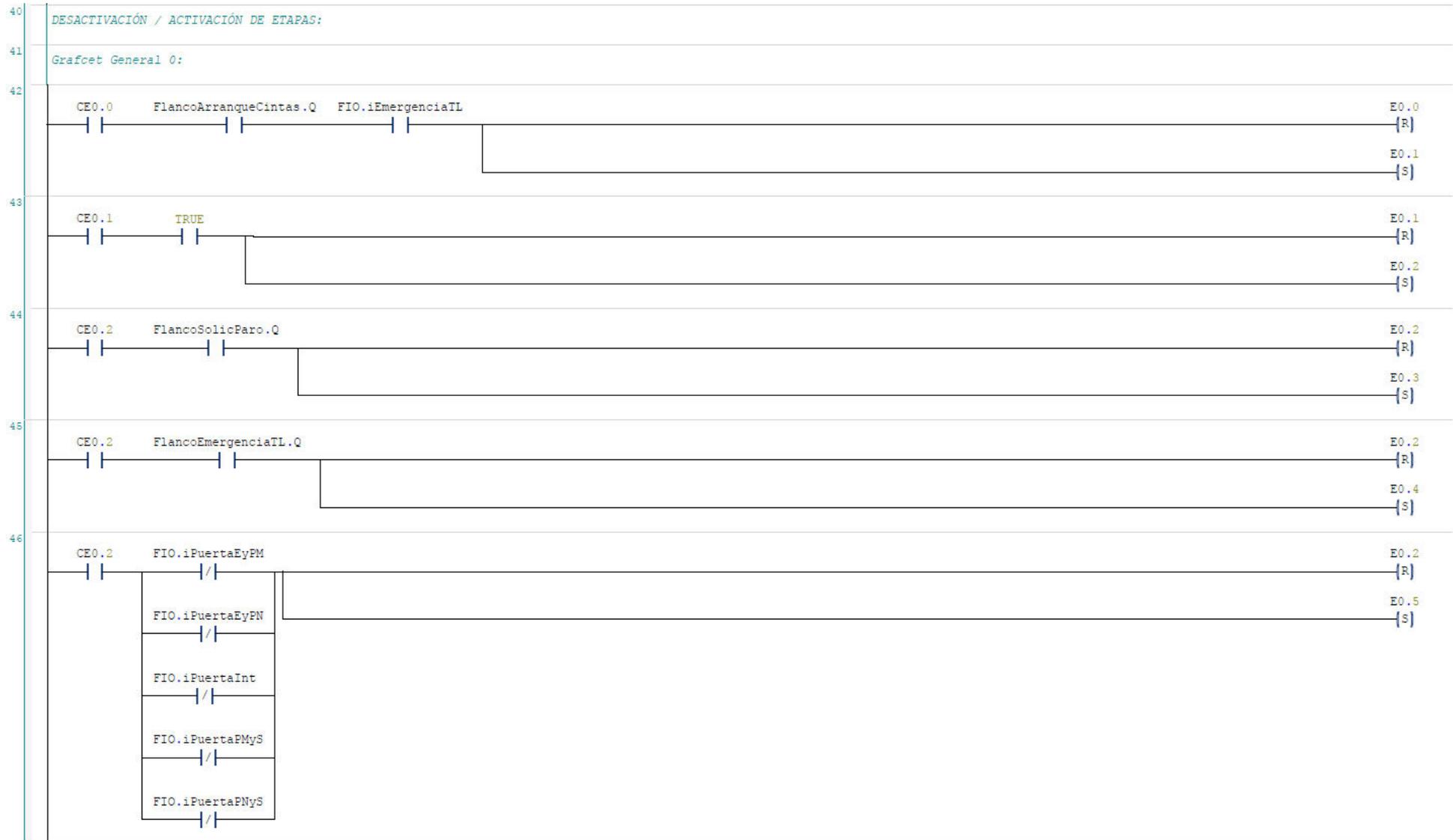
Líneas 22 a 39: Detección de Flancos

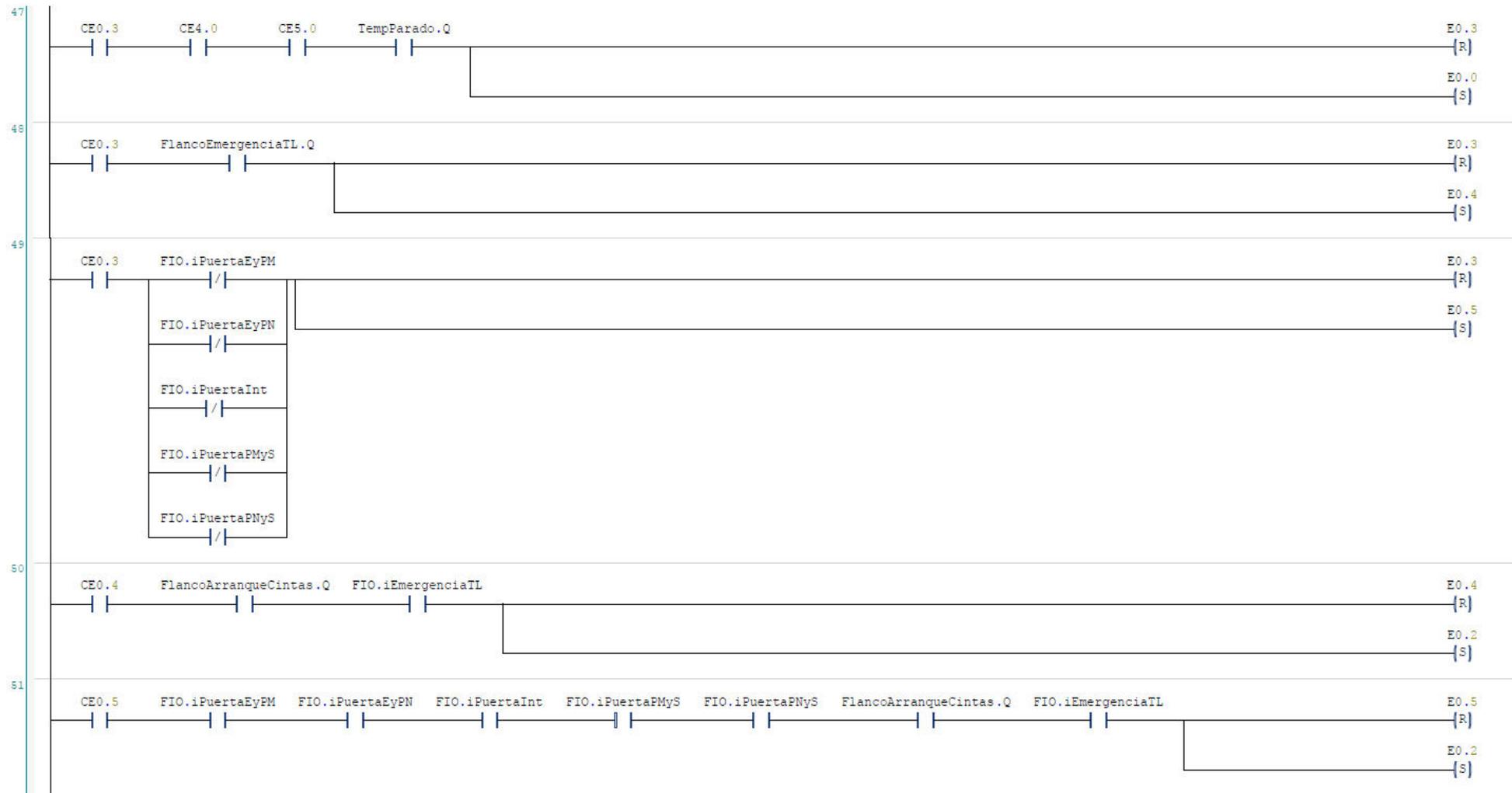


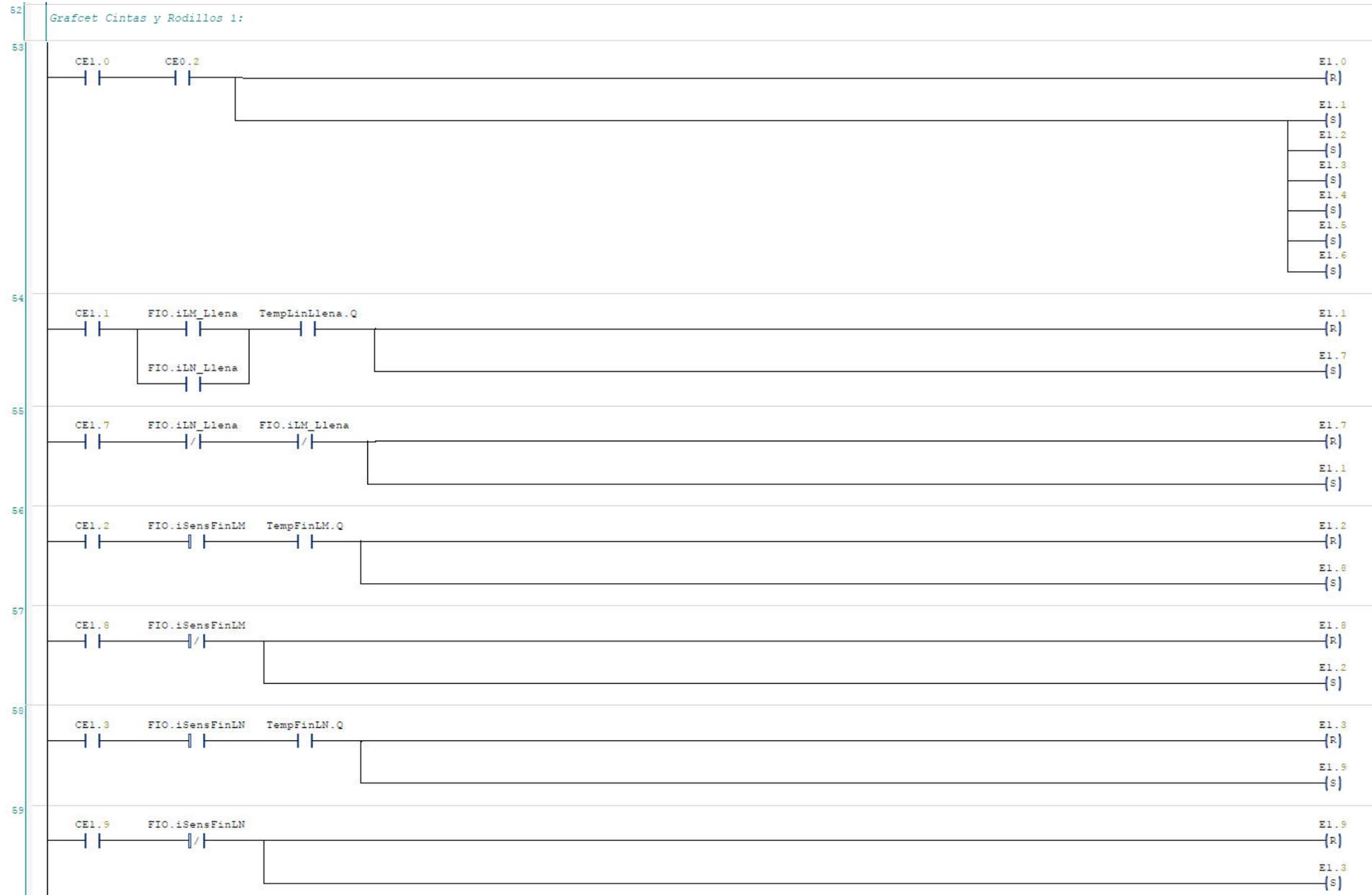




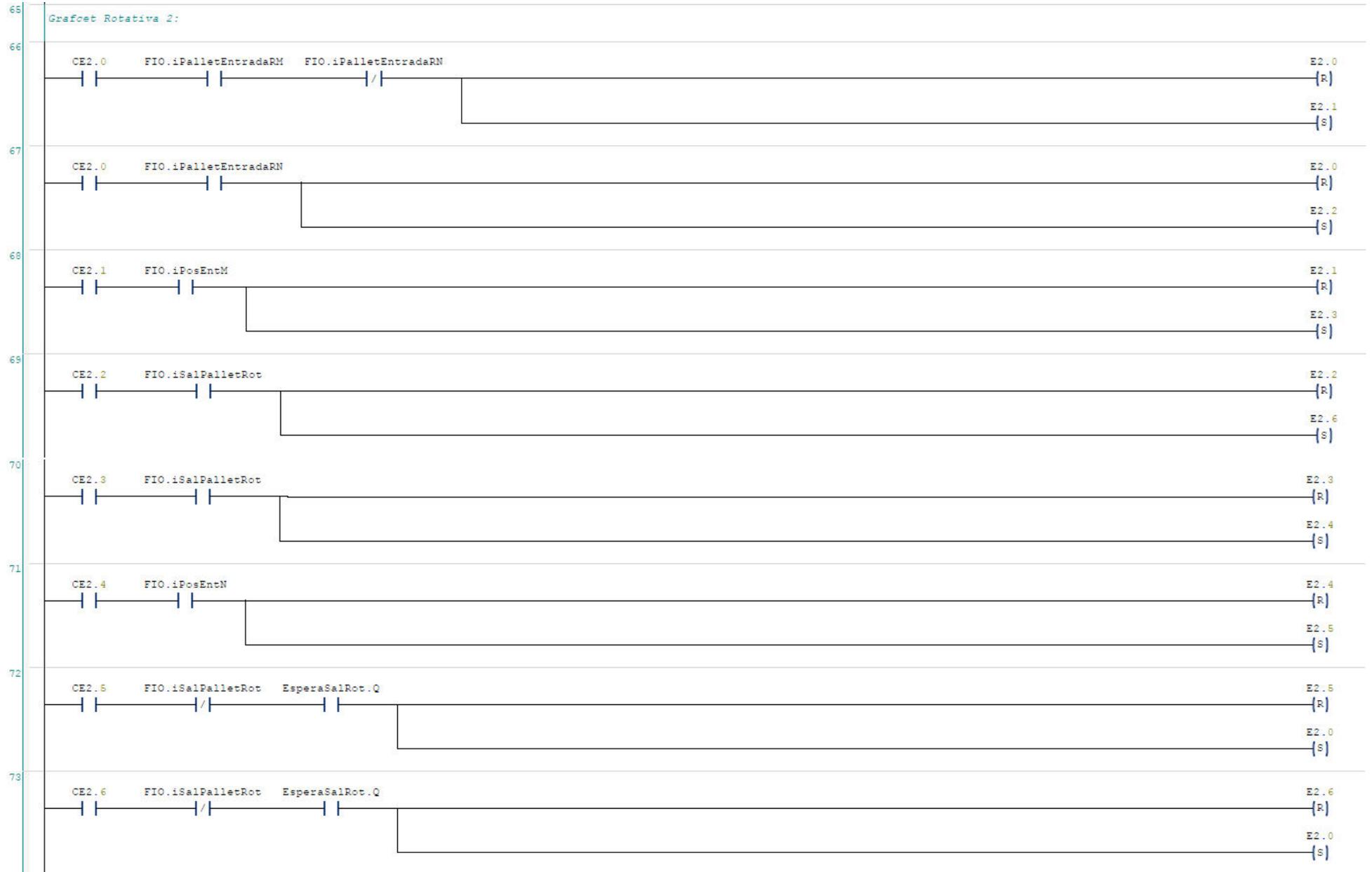
Líneas 40 a 107: Activación y Desactivación de Etapas

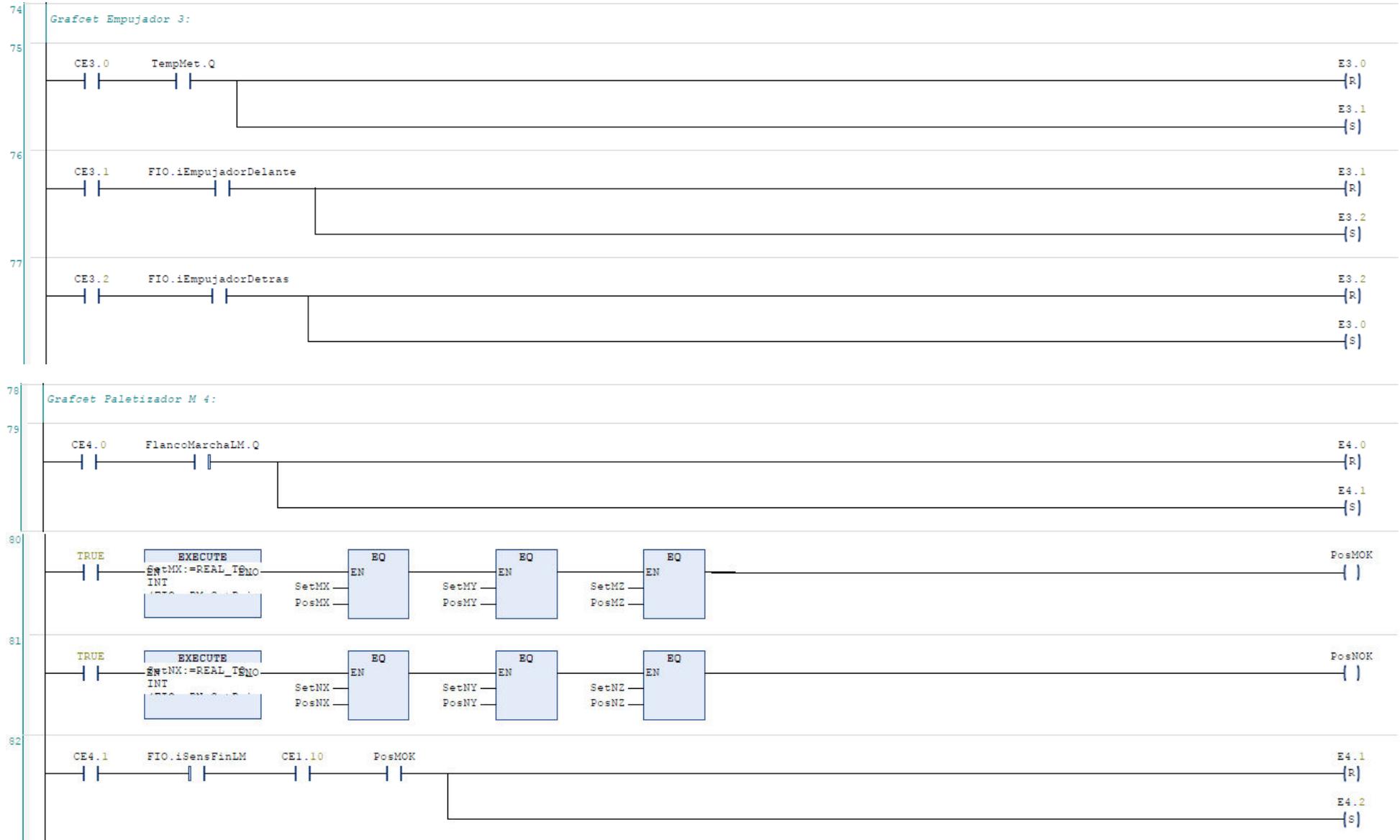


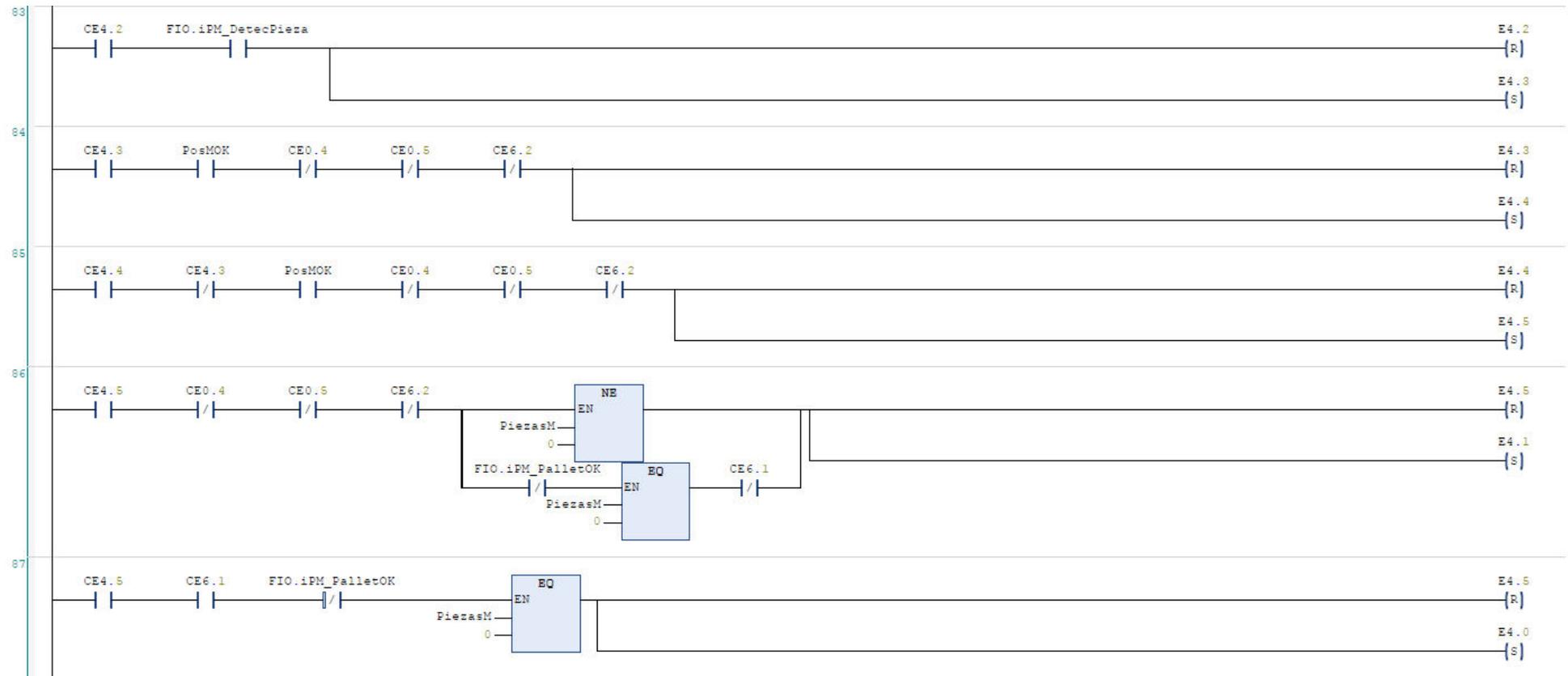


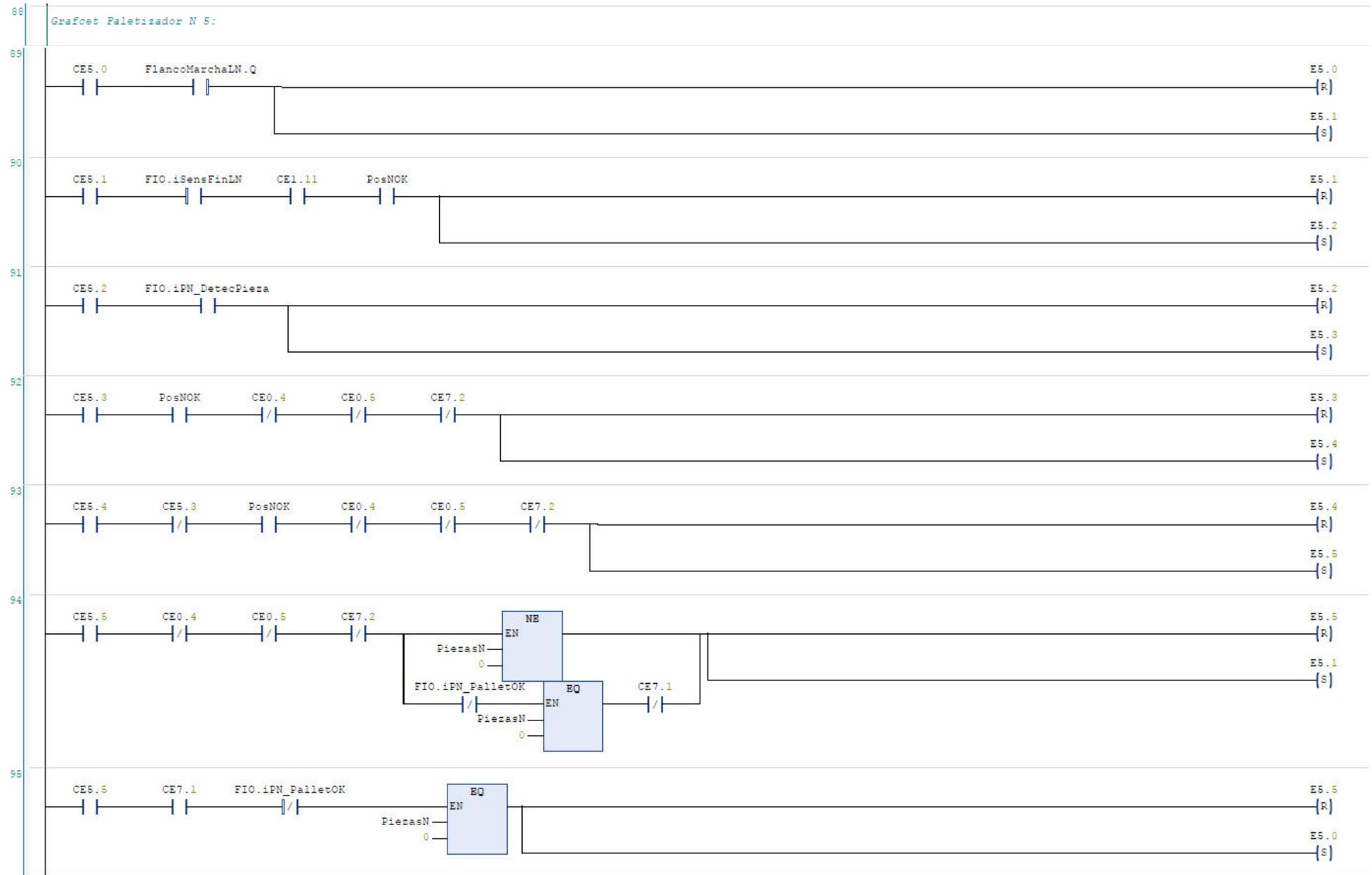










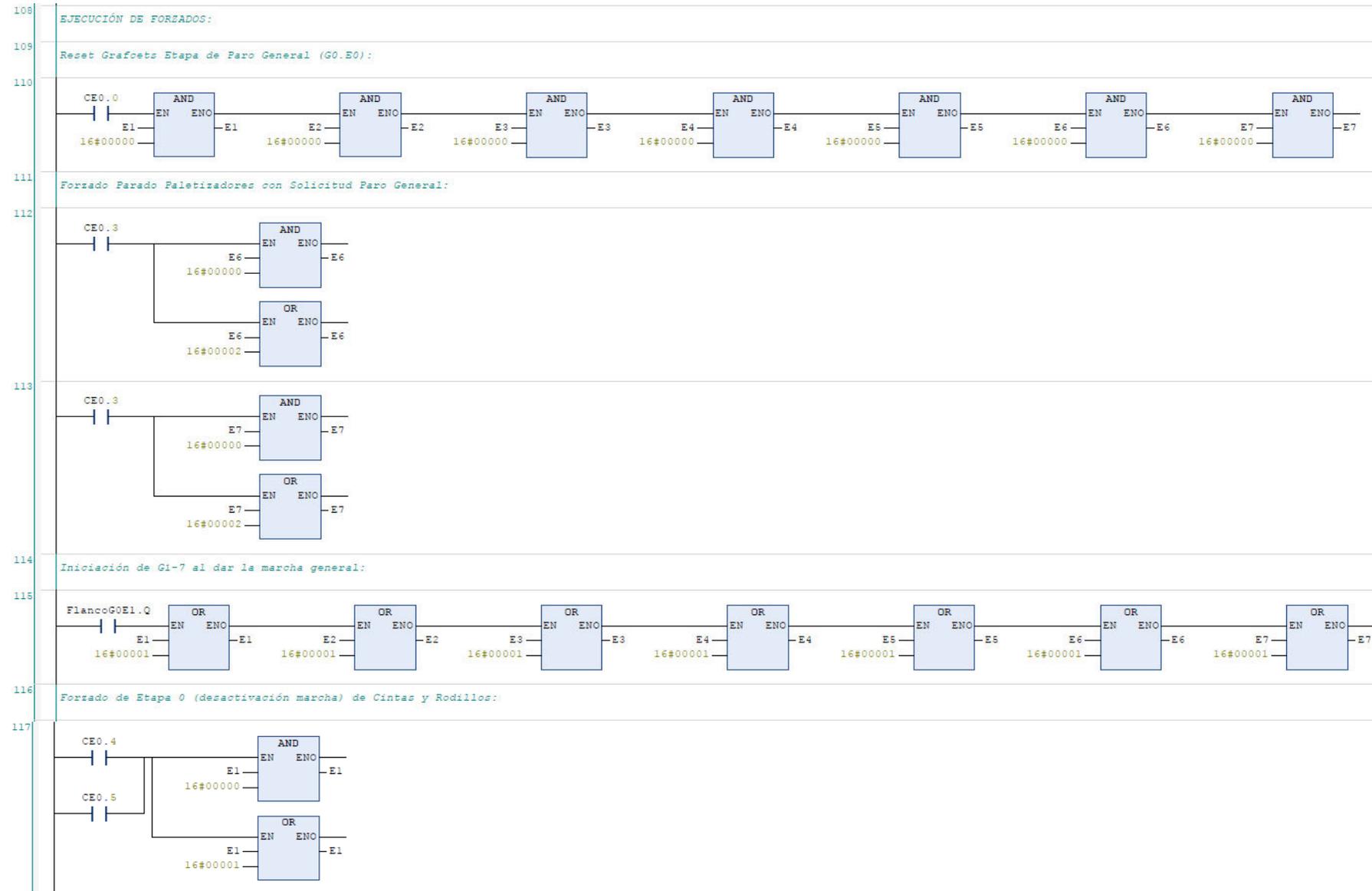






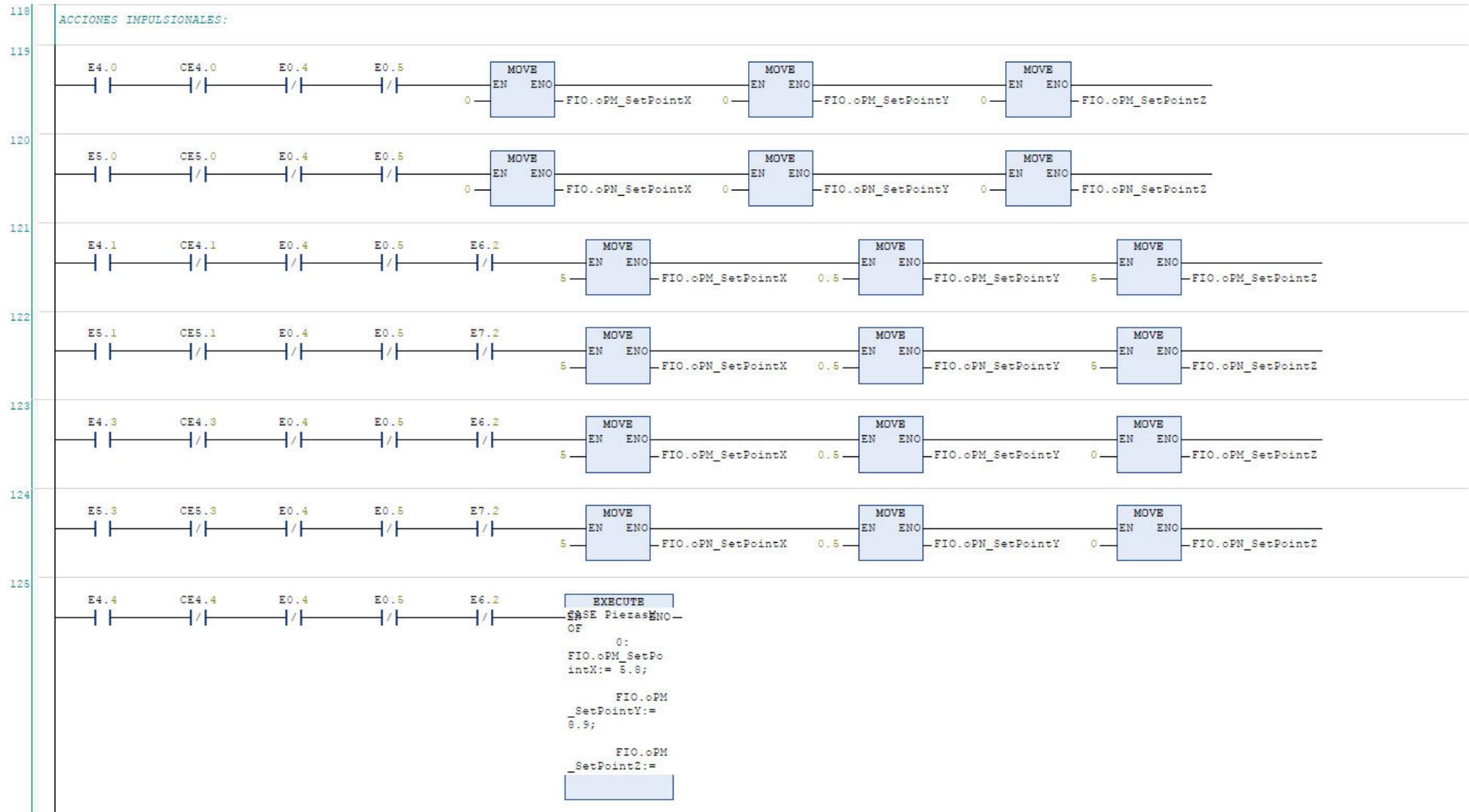


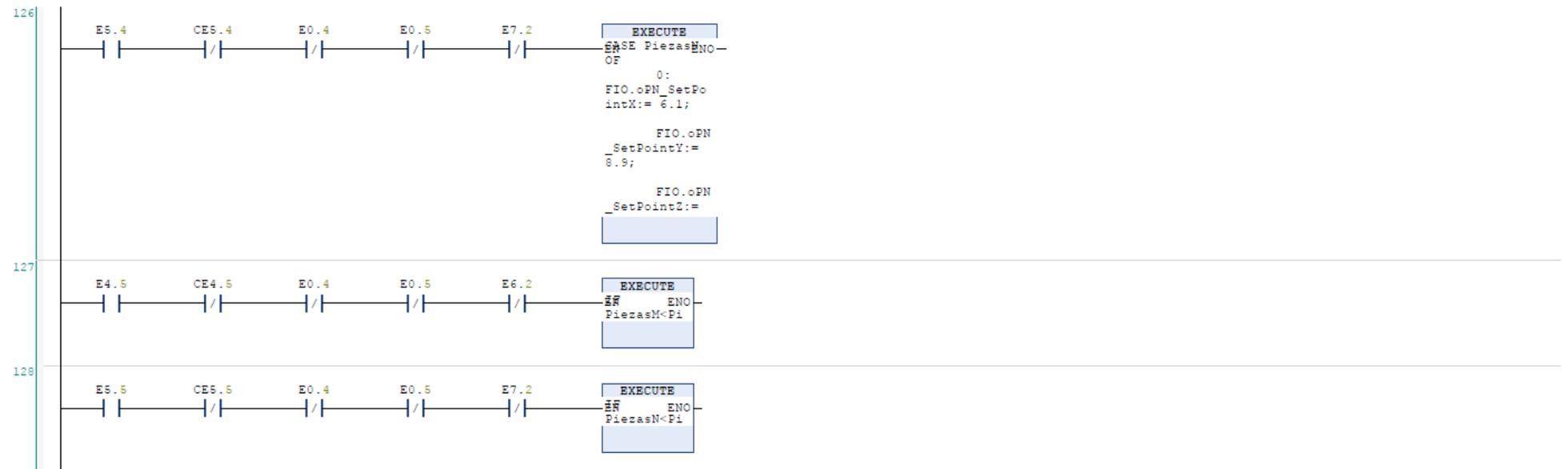
Líneas 108 a 117:





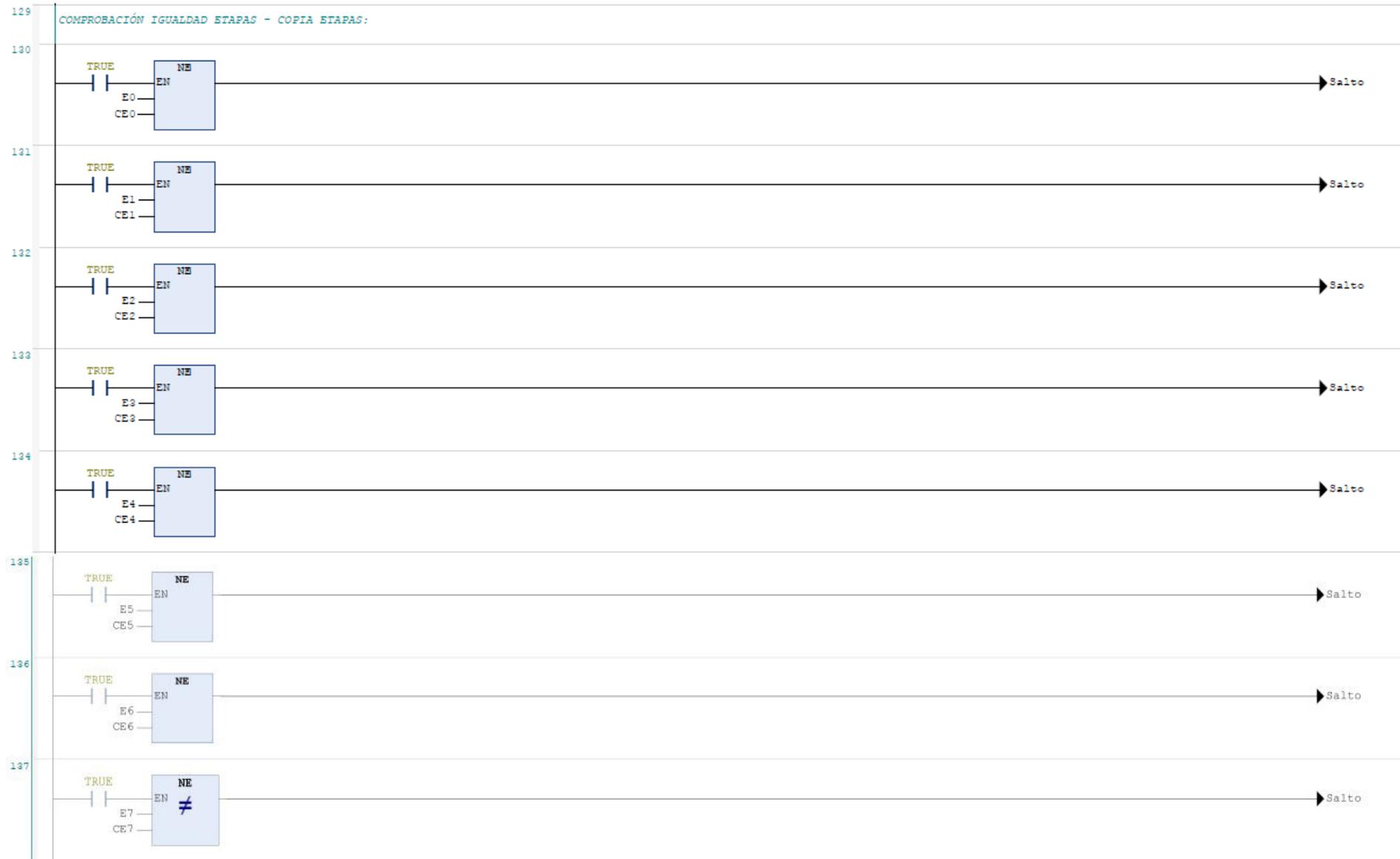
Líneas 118 a 128: Acciones Impulsionales







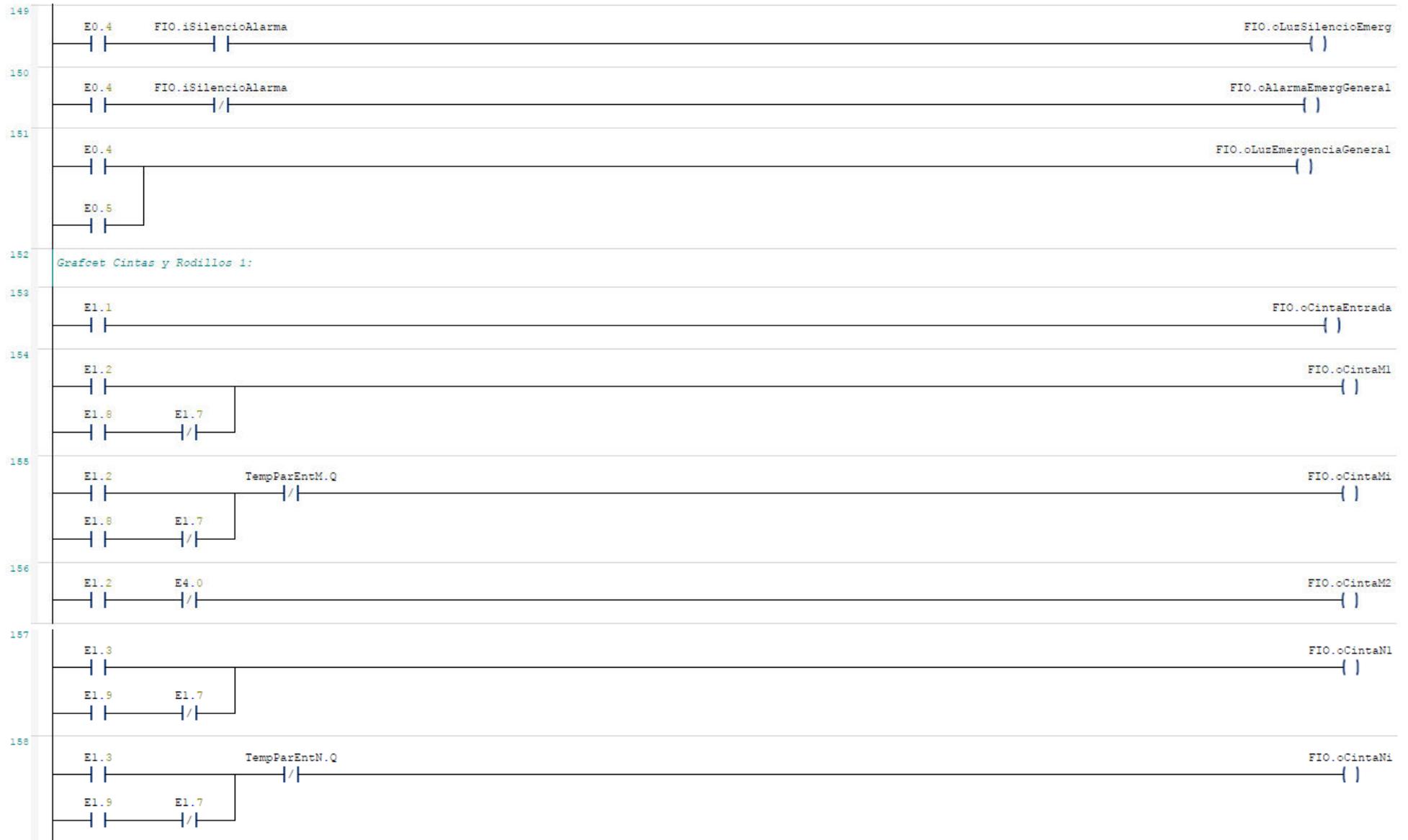
Líneas 129 a 137: Comprobación igualdad Etapas – Copia Etapas

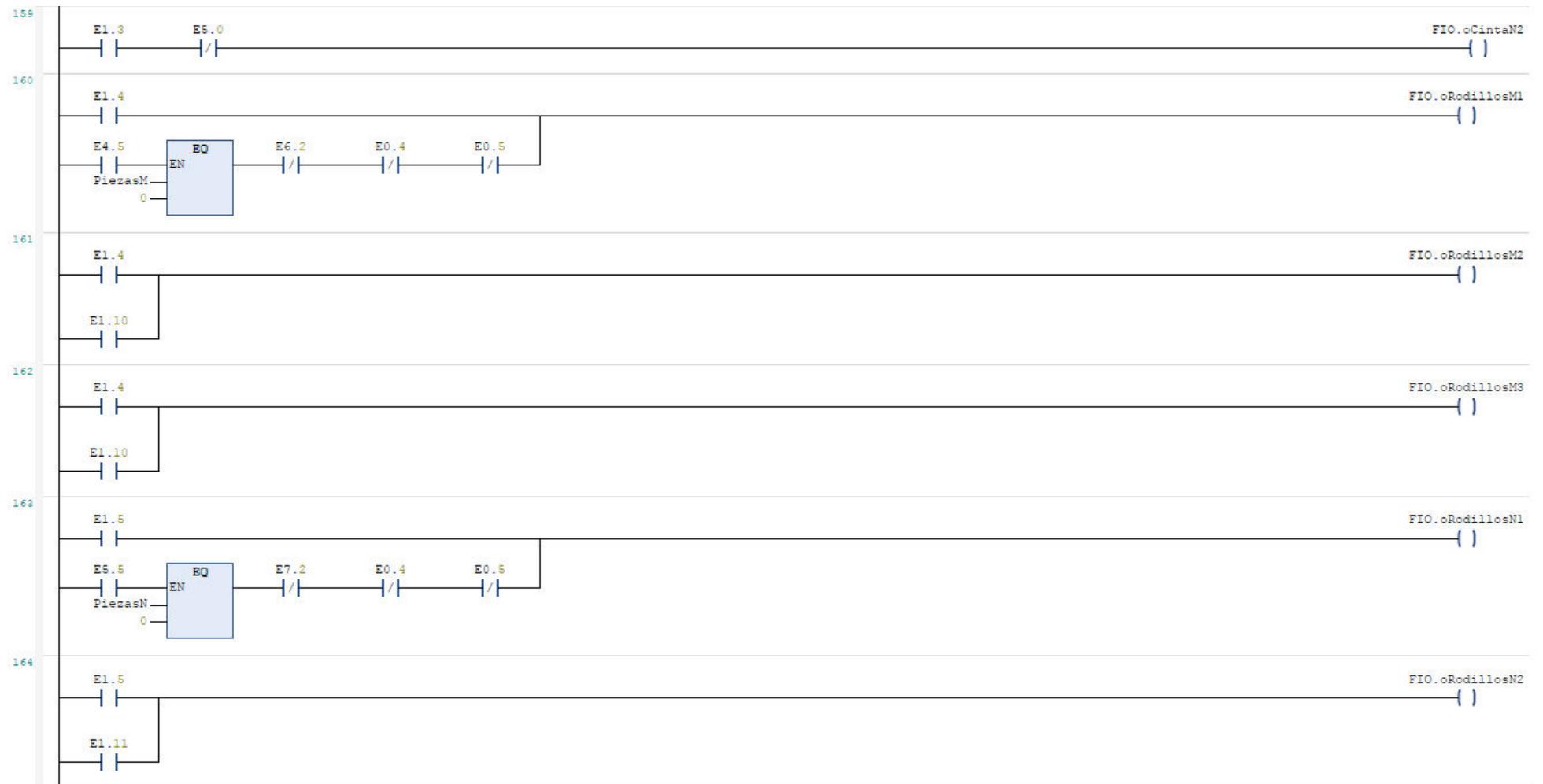




Líneas 138 a 204: Acciones de Nivel

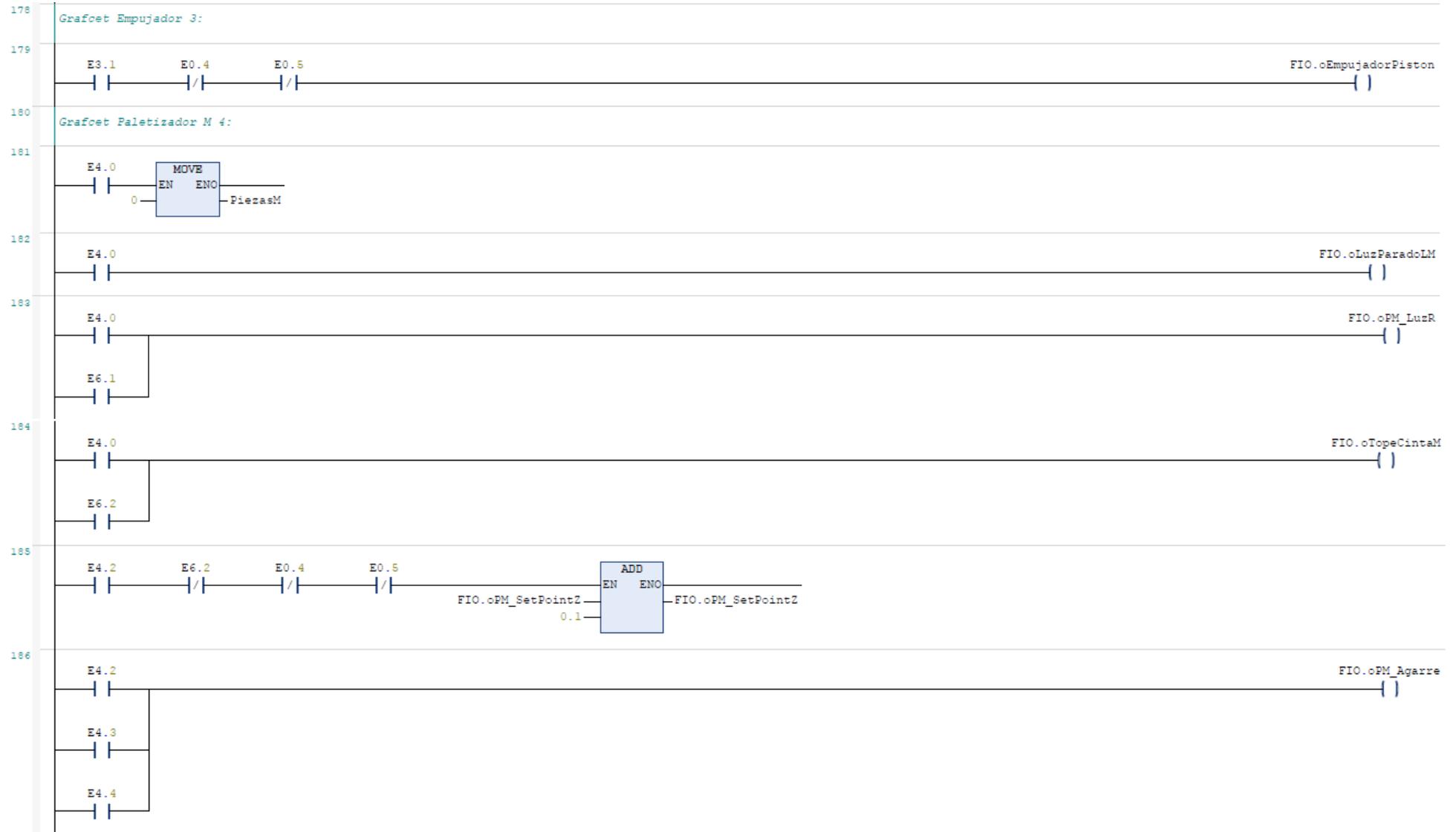


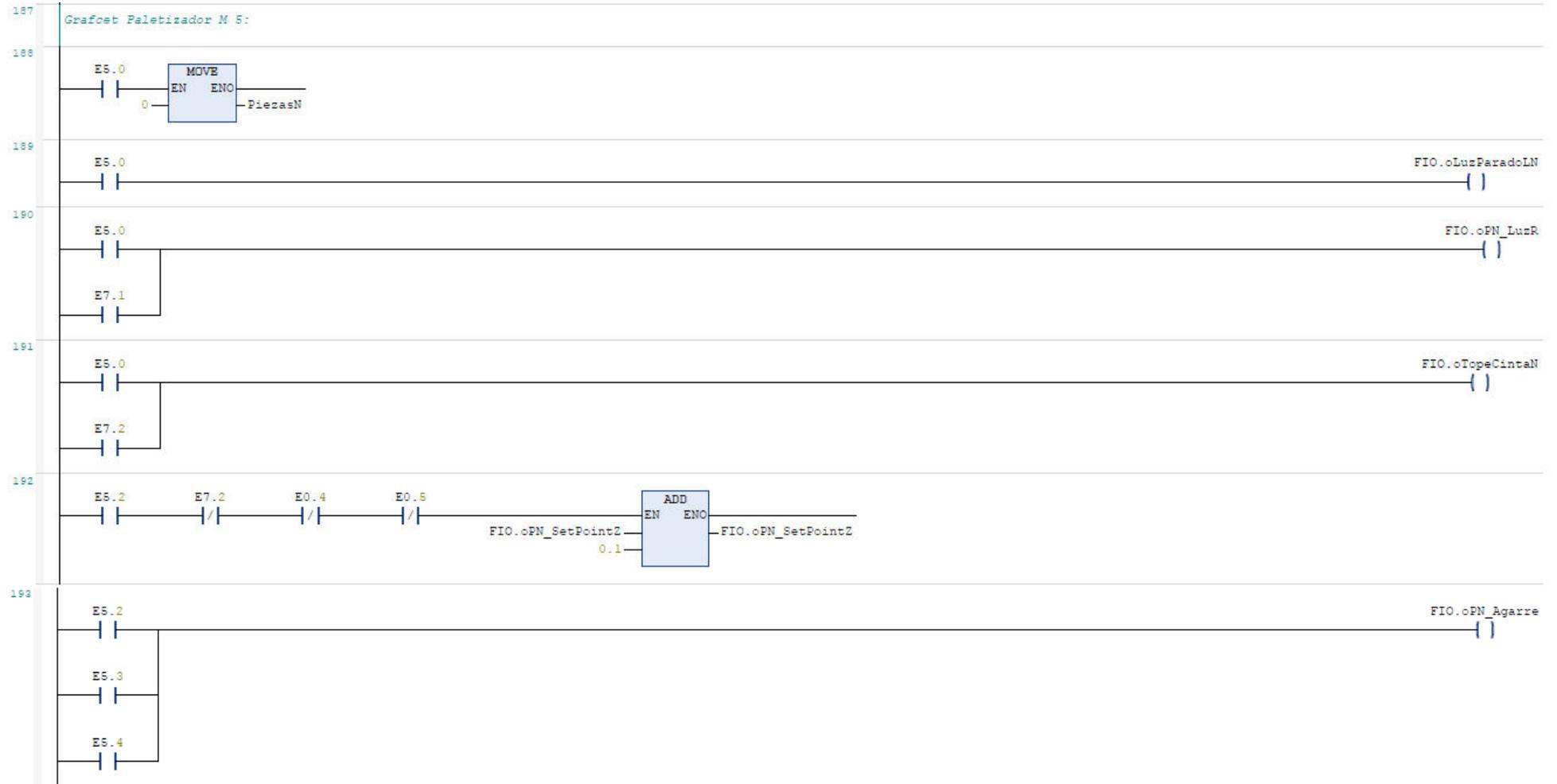
















Línea 205 a 213: Temporizadores



En algunas líneas se han introducido fragmentos de código en ST dentro de cajas de tipo EXECUTE que no se visualizan correctamente en la representación anterior. Se muestran de forma completa en los párrafos siguientes:

- Línea 80: El siguiente código convierte la señal REAL producida por los sensores de posición y la enviada a los actuadores del Paletizador M, en valores INTEGRER para poder realizar comparaciones, ya que realizarlas de forma directa con la variable tipo REAL causaba anomalías (que se pueden ver en la misma línea y activan si se cumplen la señal de posición correcta).

```

1  SetMX:=REAL_TO_INT (FIO.oPM_SetPointX*10);
2  SetMY:=REAL_TO_INT (FIO.oPM_SetPointY*10);
3  SetMZ:=REAL_TO_INT (FIO.oPM_SetPointZ*10);
4  PosMX:=REAL_TO_INT (FIO.iPM_PositionX*10);
5  PosMY:=REAL_TO_INT (FIO.iPM_PositionY*10);
6  PosMZ:=REAL_TO_INT (FIO.iPM_PositionZ*10);
    
```

- Línea 81: Se trata del mismo código que en la Línea 80, pero enfocado en el Paletizador N.

```

1  SetNX:=REAL_TO_INT (FIO.oPN_SetPointX*10);
2  SetNY:=REAL_TO_INT (FIO.oPN_SetPointY*10);
3  SetNZ:=REAL_TO_INT (FIO.oPN_SetPointZ*10);
4  PosNX:=REAL_TO_INT (FIO.iPN_PositionX*10);
5  PosNY:=REAL_TO_INT (FIO.iPN_PositionY*10);
6  PosNZ:=REAL_TO_INT (FIO.iPN_PositionZ*10);
    
```



- Línea 125: Este código determina los Set Points (puntos a los que se dirigirá la pinza) que se asignan para depositar cada pieza en su sitio. En este caso, estos puntos corresponden al Paletizador M.

```
1 CASE PiezasM OF
2     0: FIO.oPM_SetPointX:= 5.8;
3         FIO.oPM_SetPointY:= 8.9;
4         FIO.oPM_SetPointZ:= 8.2;
5     1: FIO.oPM_SetPointX:= 3.9;
6         FIO.oPM_SetPointY:= 8.9;
7         FIO.oPM_SetPointZ:= 8.2;
8     2: FIO.oPM_SetPointX:= 3.9;
9         FIO.oPM_SetPointY:= 5.6;
10        FIO.oPM_SetPointZ:= 8.2;
11    3: FIO.oPM_SetPointX:= 5.8;
12        FIO.oPM_SetPointY:= 5.6;
13        FIO.oPM_SetPointZ:= 8.2;
14    4: FIO.oPM_SetPointX:= 5.8;
15        FIO.oPM_SetPointY:= 8.9;
16        FIO.oPM_SetPointZ:= 6.5;
17    5: FIO.oPM_SetPointX:= 3.9;
18        FIO.oPM_SetPointY:= 8.9;
19        FIO.oPM_SetPointZ:= 6.5;
20    6: FIO.oPM_SetPointX:= 3.9;
21        FIO.oPM_SetPointY:= 5.6;
22        FIO.oPM_SetPointZ:= 6.5;
23    7: FIO.oPM_SetPointX:= 5.8;
24        FIO.oPM_SetPointY:= 5.6;
25        FIO.oPM_SetPointZ:= 6.5;
26    8: FIO.oPM_SetPointX:= 5.8;
27        FIO.oPM_SetPointY:= 7;
28        FIO.oPM_SetPointZ:= 5;
29    9: FIO.oPM_SetPointX:= 3.9;
30        FIO.oPM_SetPointY:= 7;
31        FIO.oPM_SetPointZ:= 5;
32 END_CASE ;
```

- Línea 126: Mismo código que en la línea 125, adaptado al Paletizador N

```

1  CASE PiezasN OF
2      0: FIO.oPN_SetPointX:= 6.1;
3          FIO.oPN_SetPointY:= 8.9;
4          FIO.oPN_SetPointZ:= 8.2;
5      1: FIO.oPN_SetPointX:= 4.1;
6          FIO.oPN_SetPointY:= 8.9;
7          FIO.oPN_SetPointZ:= 8.2;
8      2: FIO.oPN_SetPointX:= 4.1;
9          FIO.oPN_SetPointY:= 5.6;
10         FIO.oPN_SetPointZ:= 8.2;
11     3: FIO.oPN_SetPointX:= 6.1;
12         FIO.oPN_SetPointY:= 5.6;
13         FIO.oPN_SetPointZ:= 8.2;
14     4: FIO.oPN_SetPointX:= 6.1;
15         FIO.oPN_SetPointY:= 8.9;
16         FIO.oPN_SetPointZ:= 6.5;
17     5: FIO.oPN_SetPointX:= 4.1;
18         FIO.oPN_SetPointY:= 8.9;
19         FIO.oPN_SetPointZ:= 6.5;
20     6: FIO.oPN_SetPointX:= 4.1;
21         FIO.oPN_SetPointY:= 5.6;
22         FIO.oPN_SetPointZ:= 6.5;
23     7: FIO.oPN_SetPointX:= 6;
24         FIO.oPN_SetPointY:= 5.6;
25         FIO.oPN_SetPointZ:= 6.5;
26     8: FIO.oPN_SetPointX:= 6.1;
27         FIO.oPN_SetPointY:= 7.1;
28         FIO.oPN_SetPointZ:= 4.8;
29     9: FIO.oPN_SetPointX:= 4.1;
30         FIO.oPN_SetPointY:= 7.1;
31         FIO.oPN_SetPointZ:= 4.8;
32 END_CASE;
```

- Línea 127: Código encargado de sumar 1 al número de piezas colocadas en el pallet actual en el Paletizador M o de resetear el contador cuando el pallet está completo.

```
1  IF PiezasM<PiezasPalletM THEN
2      PiezasM:=PiezasM+1;
3  ELSE
4      PiezasM:=0;
5  END_IF
```

- Línea 128: Mismo código que en la Línea 127, adaptado a las variables del Paletizador N.

```
1  IF PiezasN<PiezasPalletN THEN
2      PiezasN:=PiezasN+1;
3  ELSE
4      PiezasN:=0;
5  END IF
```



PLIEGO DE CONDICIONES



Índice Pliego de Condiciones

3.1. Condiciones técnicas.....	137
3.2. Condiciones económicas.....	138
3.3. Condiciones legales.....	139
3.4. Condiciones facultativas.....	140



3.1. Condiciones técnicas

Al tratarse de un proyecto llevado a cabo en ordenador, el requisito técnico principal es disponer de un dispositivo que cumpla las siguientes especificaciones, de forma que sea posible trabajar con los programas con normalidad:

- 8GB de memoria RAM.
- 15GB de espacio libre en el disco duro.
- Procesador con velocidad de reloj de 2.5GHz o superior con soporte de instrucciones SSE2.
- Sistema Operativo: Windows 7 SP1 o más reciente, versión de 64 bits.
- Tarjeta Gráfica NVIDIA o AMD posterior a 2006, o bien, Intel posterior a 2012.

Del mismo modo, se deberá disponer de los programas necesarios instalados en la versión mencionada a continuación o más reciente:

- CODESYS V3.5 SP18 Patch 2 (versión 3.5.18.20).
- CODESYS Control Win SysTray (versión de 32-bit).
- CODESYS Gateway SysTray (versión 3.5.18.20).
- Factory I/O versión 2.5.1 – Modbus & OPC Edition.
- OFT2 Omegon Fluid Technology Grafcet (versión 2.0.9.2).

3.2. Condiciones económicas

La materialización del presente proyecto está condicionado a la adquisición de una licencia de los programas utilizados en el desarrollo del mismo, siempre que no sean programas *freeware* (licencia libre).

Los programas a obtener licencia, así como el coste de esta son:

- CODESYS Control Win SL: Licencia para un dispositivo. Coste de 420€ más impuestos, pago único.
- Factory I/O edición Modbus & OPC: Licencia para un dispositivo. Coste de 395€ más impuestos, pago único.

Asimismo, se deberá adquirir un ordenador que cumpla los requisitos mencionados en el apartado anterior, si no se dispone de uno. El coste del mismo puede ser variable en función de las características finales y el momento de mercado.

3.3. Condiciones legales

La correcta realización de este proyecto está sujeta al seguimiento de las normas y estándares que han sido explicados en el apartado de Disposiciones legales y normas aplicadas. Se enumeran a continuación:

- IEC 61131-3: Estándar de lenguajes de programación de PLC
- UNE-EN IEC 62541:2020: Norma que determina el protocolo de comunicación OPC UA. Consta de 15 partes.
- UNE-EN 60848:2022: Norma que fija el lenguaje GRAFCET.

3.4. Condiciones facultativas

El ingeniero encargado de realizar el diseño y la programación del gemelo virtual del proceso debe ajustarse con la mayor fidelidad posible al proceso que se va a dar en la realidad. No obstante, se debe comprobar el correcto funcionamiento del programa en la línea real antes de comenzar la producción en masa, puesto que puede haber ligeras variaciones entre el proyecto virtual y la puesta en marcha final que merezcan una corrección.

Esta herramienta está diseñada para reducir el tiempo necesario de ajuste en la puesta en marcha de la maquinaria, y en ningún caso su deber es sustituir dicho ajuste final que debe ser realizado con las precauciones habituales.

El proyectista no se hace responsable de posibles daños a personas o máquinas derivados de una utilización irresponsable o incorrecta de la herramienta presentada.

PRESUPUESTO



Índice Presupuesto

4.1. Cuadro de precios descompuestos.....	144
4.2. Cálculo PEM, PEC y Presupuesto Total.....	145

4.1. Cuadro de precios descompuestos

- Mano de Obra: La realización del proyecto requiere un ingeniero industrial o similar, con formación en Automatización Industrial.

Actividad	€/h	Nº horas
Diseño del proceso	20	40
Programación del control automático	20	40
TOTAL		1600€

Tabla 7: Coste mano de obra

- Software: Licencias para 1 dispositivo
 - Factory I/O Modbus & OPC Edition: 395€, pago único
 - CODESYS Control Win SL: 420€, pago único

4.2. Cálculo PEM, PEC y Presupuesto Total

A continuación, se muestran en detalle, y por separado los diferentes constituyentes del presupuesto total del proyecto.

PRESUPUESTO DE EJECUCIÓN DE MATERIAL (PEM)

Concepto	Coste (€)
Ingeniero (diseño y programación)	1600,00
Software	815,00
TOTAL PEM	2415,00

Tabla 8: Presupuesto de Ejecución de Material

PRESUPUESTO DE EJECUCIÓN POR CONTRATA (PEC)

Concepto	Coste (€)
PEM	2415,00
20% gastos generales y cargas fiscales	483,00
6% beneficio industrial	144,90
TOTAL PEC	3042,90

Tabla 9: Presupuesto de Ejecución por Contrata

PRESUPUESTO TOTAL

Concepto	Coste (€)
PEC	3042,90
7% Honorarios Proyectista	213,00
<i>TOTAL (sin iva)</i>	<i>3255,90</i>
21% IVA	683,74
PRESUPUESTO TOTAL	3939,64

Tabla 10: Presupuesto Total

El presupuesto total asciende a tres mil novecientos treinta y nueve euros con sesenta y cuatro céntimos.