



**Universidad
Europea**

**RETROFIT DEL SISTEMA DE
AUTOMATIZACIÓN DE UNA
MÁQUINA DESHUESADORA
DE PRODUCTOS DE LA
INDUSTRIA PORCINA**

TRABAJO FIN DE MÁSTER

**UNIVERSIDAD EUROPEA DE MADRID
ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO
ÁREA INGENIERÍA INDUSTRIAL**

CURSO ACADÉMICO 2023-2024

Quiero dedicar este trabajo a Ragnar, mi fiel compañero durante tantos años. Su lealtad, cariño y alegría fueron un apoyo constante en los momentos más difíciles. Aunque ya no está conmigo, su recuerdo siempre me acompañará, y este logro también es en parte gracias a la energía y felicidad que me brindó. Este trabajo está dedicado a él, que con su amor incondicional hizo más especial cada día de esta etapa de mi vida.



**Universidad
Europea**

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

ÁREA INGENIERÍA INDUSTRIAL

MÁSTER UNIVERSITARIO EN INGENIERÍA

INDUSTRIAL HCAP

TRABAJO FIN DE MÁSTER

RETROFIT DEL SISTEMA DE

AUTOMATIZACIÓN DE UNA MÁQUINA

DESHUESADORA DE PRODUCTOS DE LA

INDUSTRIA PORCINA

Alumno: D. PABLO TORRES MORENO

Director: D. ÁLVARO ROMERAL GALLEGO

SEPTIEMBRE 2024

TÍTULO: RETROFIT DEL SISTEMA DE AUTOMATIZACIÓN DE UNA MÁQUINA
DESHUESADORA DE PRODUCTOS DE LA INDUSTRIA PORCINA

AUTOR: PABLO TORRES MORENO

DIRECTOR DEL PROYECTO UEM: ÁLVARO ROMERAL GALLEGO

DIRECTOR DEL PROYECTO EMPRESA: MIGUEL ÁNGEL ROSADO RODRIGUEZ

FECHA: 20 de septiembre de 2024

JUSTIFICACIÓN

Este proyecto consiste en un trabajo real al cual nos hemos enfrentado en mi lugar de trabajo. Se trata del Retrofit de una máquina para deshuesar productos de la industria porcina, para ser más exactos de los huesos de la pata del jamón.

Se ha realizado tanto el cálculo de los motores y de los nuevos ejes, como la programación del PLC, a través del programa Codesys en lenguaje de texto estructurado, y de las controladoras; el diseño del nuevo cuadro eléctrico y la programación del HMI que lleva la máquina (PLC y HMI integrados en un solo componente).

La máquina tenía funcionamiento/automatismo prácticamente neumático, salvo un motor asíncrono trifásico para girar una sierra mediante una polea que se encarga de cortar la pezuña del jamón. Por lo que se decidió que se iba a cambiar toda esta tecnología de finales de los 90 por una tecnología basada y preparada para la industria actual, manteniendo los movimientos más simples con actuadores neumáticos y los movimientos en lo que el control tiene que ser más exhaustivo, por actuadores eléctricos.

Esta, llegó a la empresa operativa, pero como gran objetivo reducir los tiempos de funcionamiento, hacerla más funcional gracias al PLC y al HMI incorporado con el que este cuenta; más segura y sobre todo, tener un control total de ciertos movimientos, como los movimientos de corte (mediante variador de frecuencia y eje eléctrico que es el que mueve la sierra hacia la pata del jamón) y el movimiento de desgarrar el hueso de la parte de la carne (la cual se ha realizado mediante un control de par, al igual que el cilindro eléctrico del corte de la pata).

La parte mecánica de la máquina estaba en perfecto estado, ya que se trataba de una máquina de mecánica muy robusta, por lo que se decidió realizar un retrofit de la misma, en lugar de cambiar la máquina completa. A continuación, se hizo un estudio previo para entender el funcionamiento de esta, para proceder más adelante, a realizar los cálculos pertinentes para la elección de los nuevos accionamientos (tanto eléctricos como neumáticos). Posteriormente se decidió añadir un variador de frecuencia, para tener un mayor control sobre el motor asíncrono para el corte de la pata del jamón. Y para finalizar, se comenzó a realizar la programación conjunta del PLC y del HMI. Realizando simulaciones para comprobar el buen funcionamiento de esta. Como último paso se implementó en la máquina y se realizó un seguimiento del funcionamiento.

Actualmente, se encuentra en proceso de mejora, implementando mejoras en la programación y últimas necesidades del cliente.

Palabras clave: Retrofit, automatización, cálculo, electrónica, simulación y programación.

ABSTRACT

This project consists of a real task we encountered at my workplace. It involves the retrofit of a machine designed to debone pork industry products, specifically the bones from ham legs.

We performed both the calculation of the motors and the new shafts, as well as the programming of the PLC, using Codesys in structured text language, and the controllers; the design of the new electrical panel, and the programming of the HMI that the machine carries (PLC and HMI integrated into a single component).

The machine's operation/automation was almost entirely pneumatic, except for a three-phase asynchronous motor that rotated a saw via a pulley to cut the ham's hoof. Therefore, it was decided that all this technology from the late 90s would be replaced with technology suitable for today's industry, keeping the simpler movements with pneumatic actuators and the movements where more precise control is needed with electric actuators.

The machine arrived at the company operational, but the main goal was to reduce operating times, make it more functional thanks to the incorporated PLC and HMI, make it safer, and above all, have total control over certain movements, such as cutting movements (using a frequency inverter and an electric axis that moves the saw towards the ham leg) and the movement to tear the bone from the meat (which was done using torque control, just like the electric cylinder for cutting the leg).

The mechanical part of the machine was in perfect condition, as it was a very robust mechanical machine, so it was decided to retrofit it rather than replace the entire machine. A preliminary study was carried out to understand its operation, and later, the necessary calculations were made for the selection of the new actuators (both electric and pneumatic). A frequency inverter was subsequently added to have better control over the asynchronous motor for cutting the ham leg. Finally, joint programming of the PLC and HMI was started, and simulations were conducted to verify its proper functioning. As the final step, it was implemented into the machine, and the operation was monitored.

Currently, it is in the process of improvement, with enhancements being made to the programming and final adjustments based on the client's needs.

DEFINICIONES Y ABREVIATURAS [1]

- **PD:** Positioning Drives.
- **ST:** Structured Text (Texto Estructurado).
- **Retrofit:** modernización de alguna de las partes de una máquina, normalmente suelen ser los elementos de control. Suele realizarse cuando las piezas de fábrica no sirven o ya no realizan los procesos de la manera deseada.
- **PLC:** Programmable Logic Controller, se trata de un ordenador que se utiliza para aplicaciones de automatización, para automatizar procesos, control de maquinaria., procesos de producción...
- **HMI:** Human Machine Interface, es la interfaz entre el proceso y los operarios, es una pantalla que nos muestra información en tiempo real del estado de la máquina y de sus variables. De la misma manera, que muestra información nos puede servir para introducir información al PLC.
- **Ethercat:** protocolo de comunicación en código abierto de alto rendimiento, actualmente es el sistema más rápido de procesamiento en marcha. Busca utilizar el protocolo de Ethernet en el entorno de la industria.
- **Servomotor:** se trata de un motor que contiene un encoder, el cual nos permite tener un control de precisión de la velocidad, aceleración, par y posición del motor.
- **Encoder:** se trata de un decodificador que convierte el movimiento giratorio de un eje en pulsos digitales, los cuales interpreta un driver.
- **Driver:** se encarga de interpretar los pulsos del encoder y ejecutar órdenes sobre este; y de alimentar a los servomotores.
- **Variador de frecuencia:** regulador industrial que se encuentra entre la alimentación energética y el motor. La energía de la red pasa por el variador y regula la energía antes de que ésta llegue al motor para luego ajustar la frecuencia y la tensión en función de los requisitos del procedimiento.

Índice

JUSTIFICACIÓN	7
ABSTRACT	8
DEFINICIONES Y ABREVIATURAS [1]	9
Índice de Ilustraciones	13
Índice de Tablas	14
1. INTRODUCCIÓN	15
1.1. PLANTEAMIENTO DEL PROBLEMA	15
1.2. OBJETIVOS DEL PROYECTO	15
2. PROGRAMAS UTILIZADOS	17
3. OBJETIVOS	18
4. ANÁLISIS DE SOLUCIONES	19
4.1. ESTUDIO DE LOS MOVIMIENTOS DE LA MÁQUINA	19
4.1.1. <i>SECTOR 1 – BOTONERÍA</i>	19
4.1.2. <i>SECTOR 2 – AGARRE Y CORTE</i>	19
4.1.3. <i>SECTOR 3 - PRIMER DESHUESADO</i>	20
4.1.4. <i>SECTOR 4 – DESHUESADO PRINCIPAL</i>	20
4.1.5. <i>SECTOR 5 – ELEVACIÓN MÁQUINA</i>	21
4.3. PASOS REALIZADOS	22
4.3.1. <i>ANÁLISIS Y DESMONTAJE</i>	22
4.3.2. <i>CÁLCULO DE CILINDROS</i>	22
4.3.3. <i>MECANIZADO DE ACTUADORES</i>	23
4.3.4. <i>MECANIZADO EN MÁQUINA</i>	23
4.3.5. <i>CABLEADO INTERIOR DEL ARMARIO, MOTORES Y PULSATERIA Y SEÑALIZACIÓN</i>	23
4.3.6. <i>MONTAJE DEL ARMARIO DE PVC</i>	23
4.3.7. <i>PROGRAMACIÓN</i>	23
4.4. DIAGRAMAS DE PROGRAMACIÓN	24
4.4.1. <i>DIAGRAMA PRINCIPAL DE EJECUCIÓN</i>	24
4.4.2. <i>DIAGRAMA PARO</i>	25
4.4.2.1. <i>Enable y condiciones iniciales de funcionamiento</i>	26
4.4.2.2. <i>Gestión movimientos neumáticos y eléctricos</i>	27

4.4.2.3.	<i>Gestión alarmas</i>	29
4.4.2.4.	<i>Gestión HMI</i>	29
4.4.2.5.	<i>Gestión paro</i>	30
4.5.	ELEMENTOS DE CONTROL UTILIZADOS	31
4.6.	ELEMENTOS DE ACTUACIÓN ELÉCTRICA UTILIZADOS	32
4.7.	ELEMENTOS DE ACTUACIÓN PNEUMÁTICA UTILIZADOS	33
4.8.	CUADRO ELÉCTRICO FINAL	34
4.8.1.	<i>PARTES CUADRO ELÉCTRICO</i>	34
4.8.2.	<i>ESQUEMA CUADRO ELÉCTRICO</i>	35
5.	CÁLCULO DE LOS EJES Y MOTORES ELÉCTRICOS	39
5.1.	CÁLCULO CON EL PROGRAMA POSITIONING DRIVES	39
5.1.1.	<i>SELECCIÓN DEL TIPO DE ACTUADOR</i>	39
5.1.2.	<i>PARÁMETROS DEL SISTEMA</i>	39
5.1.3.	<i>SELECCIÓN DE EJES, MOTORES Y CONTROLADORAS</i>	40
5.1.4.	<i>RESUMEN EJES</i>	43
5.2.	CÁLCULO MANUAL	45
6.	CÁLCULO DE LOS EJES PNEUMÁTICOS	46
7.	COMPARATIVA EJES PNEUMÁTICOS Y ELÉCTRICOS	47
7.1.	<i>EJES PNEUMÁTICOS VIEJOS</i>	47
7.2.	<i>EJES PNEUMÁTICOS NUEVOS</i>	47
7.3.	<i>EJES ELÉCTRICOS Y PNEUMÁTICOS NUEVOS</i>	48
8.	COMPARATIVA Y CONSUMOS	49
8.1.	<i>CONSUMO AIRE CILINDROS PNEUMÁTICOS</i>	49
8.2.	<i>CONSUMO ENERGÉTICO EJES ELÉCTRICOS</i>	50
8.3.	<i>SOSTENIBILIDAD DEL RETROFIT</i>	51
9.	PRESUPUESTO	52
10.	CONCLUSIONES	53
11.	BIBLIOGRAFÍA	54
12.	ANEXO	55
12.1.	<i>ORGANIZACIÓN PLC</i>	55
12.2.	<i>LISTAS DE VARIABLES GLOBALES</i>	58
12.2.1.	<i>GVL</i>	58
12.2.2.	<i>GVL_HMI</i>	61
12.2.3.	<i>PERSISTENTVARS</i>	62
12.3.	<i>FUNCIONES</i>	63

12.4.	PROGRAMA PRINCIPAL (PLC_PRG) Y SUBPROGRAMAS	65
12.4.1.	<i>CILINDROAGARRE_E</i>	69
12.4.2.	<i>CILINDROSIERRA</i>	70
12.4.3.	<i>CONDICIONES_INICIALES</i>	71
12.4.4.	<i>GESTION_ALARMAS</i>	72
12.4.5.	<i>GESTION_MOVIMIENTOS</i>	73
12.4.6.	<i>GESTION_MOVIMIENTOS ELÉCTRICOS</i>	75
12.4.7.	<i>GESTION_VARIADOR</i>	76
12.5.	PROGRAMA PRINCIPAL (HMI)	77
12.6.	PROGRAMA PRINCIPAL (MOV_SIM_HMI)	79
12.7.1.	<i>ENTRADAS DIGITALES</i>	79
12.7.2.	<i>SALIDAS DIGITALES</i>	80
12.7.3.	<i>ENTRADA ANALÓGICA</i>	80
12.8.	ALARMAS	81
12.9.	HMI	82
12.9.1.	<i>PANTALLA PRINCIPAL</i>	82
12.9.2.	<i>ÁREAS PRINCIPALES</i>	82
12.9.2.1.	<i>A_SMenu_MarchaParo</i>	83
12.9.2.2.	<i>B_SMenu_Ajustes</i>	83
12.9.2.3.	<i>C_SMenu_EstadoES</i>	83
12.9.2.4.	<i>D_SMenu_Grafico</i>	84
12.9.2.5.	<i>E_SMenu_Alarmas</i>	84
12.9.2.6.	<i>F_SMenu_Usuarios</i>	84

Índice de Ilustraciones

Ilustración 1. Máquina final	15
Ilustración 2. Máquina Inicial	15
Ilustración 3. Descripción general de la máquina	21
Ilustración 4. Posición de trabajo del operador	22
Ilustración 5. Programa principal y subprogramas	26
Ilustración 6. CilindroN_Pata	27
Ilustración 7. Funcionamiento sierra y cilindro.....	28
Ilustración 8. Ejecución diagrama paro.....	30
Ilustración 9. Cuadro eléctrico final	34
Ilustración 10. Esquemas cuadro eléctrico	38
Ilustración 11. Selección del tio de actuador.....	39
Ilustración 12. Parámetros del sistema.....	40
Ilustración 133. Selección ejes	40
Ilustración 14. Selección motores.....	41
Ilustración 15. Selección engranajes.....	41
Ilustración 16. Selección de drivers	42
Ilustración 17. Resultados cilindro agarre.....	42
Ilustración 18. Listado de piezas	43
Ilustración 19. Resultado final cilindro agarre	43
Ilustración 20. Resultado cilindro sierra	44
Ilustración 21. Consumo de aire de los cilindros neumáticos	49
Ilustración 22. Consumo aire todos cilindros neumáticos.....	49
Ilustración 23. Consumo aire algunos cilindros neumáticos	50
Ilustración 24. Variables Globales.....	55
Ilustración 25. Funciones	55
Ilustración 26. Programación	55
Ilustración 27. Gestión HMI	56
Ilustración 28. Gestión alarmas.....	56
Ilustración 29. Gestión de tareas	57
Ilustración 30. Comunicaciones del PLC.....	57

Índice de Tablas

Tabla 1. Elementos de control utilizados	31
Tabla 2. Elementos de actuación eléctrica utilizados	32
Tabla 3. Elementos de actuación neumática utilizados	33
Tabla 4. Cálculo manual	45
Tabla 5. Tabla fuerza ejes neumáticos	46
Tabla 6. Características cilindros neumáticos viejos	47
Tabla 7. Características todos cilindros neumáticos nuevos	47
Tabla 8. Características cilindros neumáticos y eléctricos nuevos	48
Tabla 9. Cálculo energético ejes eléctricos	51
Tabla 10. Presupuesto	52

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Nos encontramos ante una máquina antigua, con un gran potencial de mejora y que por su construcción robusta y posibilidades nos planteamos actualizarla, la cual necesitamos que, con la misma estructura mecánica, consigamos actualizarla a la tecnología actual.

Los componentes de la máquina eran antiguos, por lo que aparte de hacer un estudio sobre los pasos de funcionamiento de la máquina, se ha realizado el cálculo de los nuevos accionamientos que se han usado en la máquina. Estos cálculos se han realizado a través del programa Positioning Drives de FESTO, ya que los motores y ejes serán de esta marca. A continuación, se procedió a realizar la programación del PLC y del HMI (ambos son el mismo dispositivo de la marca Berghof), esto a través del programa CODESYS.



Ilustración 2. Máquina Inicial



Ilustración 1. Máquina final

1.2. OBJETIVOS DEL PROYECTO

Podemos resumir el objetivo principal como la actualización de una máquina antigua mejorando sus tiempos de respuesta, haciéndola más intuitiva en su uso, más autónoma, más segura, aumentando su producción, siendo más rentable económicamente y más amigable con el medioambiente.

A demás, se buscó como objetivo hacer la máquina lo más sencilla posible, por lo que se optó por usar un PLC con HMI integrado para hacer más sencilla la programación de ambos.; también se buscó que esta fuese configurable en las partes eléctricas, por lo que los operarios pueden cambiar multitud de parámetros (velocidades, aceleraciones, torque, altura de la máquina...) para que la máquina funcione de la mejor manera posible en función de cada pata de jamón.

2.PROGRAMAS UTILIZADOS

- a) Microsoft Office Word.
- b) CODESYS V3.5 SP19 Patch 2: Programación del PLC y del HMI.
- c) CODESYS SoftMotion Win V3: Simulación de la programación.
- d) Positioning Drives: Cálculo de ejes y servomotores eléctricos.
- e) Excel: Cálculo de ejes y servomotores eléctricos; y presupuesto de los materiales.
- f) Air Consumption: cálculo del consumo de aire de los cilindros neumáticos.
- g) EPLAN: Diseño cuadro eléctrico.

3. OBJETIVOS

Para la elaboración del proyecto se han tenido en cuenta una serie de requisitos que se han tenido que llevar a cabo a lo largo del proceso. Los puntos más importantes son:

- **Mejorar el proceso de producción:** Se ha buscado a través de la implementación de servomotores y ejes eléctricos, dejando, en la medida de lo posible, atrás la neumática, que los procesos sean lo más rápidos, controlados y precisos posible.
- **Reducir el consumo energético:** al quitar elementos neumáticos de la máquina, se reduce considerablemente el consumo eléctrico de esta, ya que el consumo eléctrico que provoca el uso del compresor es muy superior al consumo que se genera al tener más elementos eléctricos.
- **Requisito económico:** la solución final de la elección de todo el material (ejes, servos y controladoras) no podía superar los 10.000€.
- **Requisito logístico:** el material tiene que entrar en el mismo espacio en el que estaba el material antiguo, por lo que los tamaños deben ser prácticamente iguales.
- **Requisitos de programación:** en la empresa llevamos como estándar que la comunicación sea en Ethercat y la programación en entorno Codesys.
- **Realizar un control de la producción de cada operario.**

4. ANÁLISIS DE SOLUCIONES

En este capítulo se van a tratar todas las opciones que se plantearon durante el proyecto para la resolución de los diferentes problemas que han ido apareciendo a lo largo del mismo.

4.1. ESTUDIO DE LOS MOVIMIENTOS DE LA MÁQUINA

La máquina llegó a las instalaciones de la empresa estando en completo funcionamiento, por lo que los movimientos de esta se pudieron probar y estudiar de una manera más exhaustiva.

Para realizar este estudio, se va a dividir la máquina en los siguientes sectores:

4.1.1. SECTOR 1 – BOTONERÍA

Los botones/pulsadores que se usaban en la máquina eran los siguientes:

- Dos pulsadores para realizar la función de marcha (hombre muerto).
- Un selector de si hay o no hay que realizar la función del corte de la pezuña del jamón.
- Un selector para un primer agarre del jamón para un deshuese básico del hueso del jamón.
- Un botón para la colocación de la cadena tirada por un cilindro neumático para un deshuesado en profundidad.
- Un pulsador de “patada” para la despresurización total de la máquina.
- Dos pulsadores de subida y bajada de la altura de la máquina.

4.1.2. SECTOR 2 – AGARRE Y CORTE

Teniendo claro el uso de la botonería, se puede empezar a hablar del funcionamiento de la máquina.

Lo primero que realiza es el agarre de la pata mediante un cilindro neumático, y para ello hay que usar los dos pulsadores que realizan la función de marcha, hay que pulsarlos al mismo tiempo por seguridad y evitar riesgos de atrapamiento; si baja el cilindro y este detecta que hay pata mediante un sensor, se activa una electroválvula que saca unos pinchos (mediante otro cilindro neumático) de debajo de la pata y del cilindro neumático de agarre, para tener un mejor agarre de esta, en caso contrario, el cilindro neumático deja de actuar y los pinchos no llegan a actuar en ningún momento.

A partir de aquí, el funcionamiento toma dos caminos:

a. Con corte de pezuña

Si mediante el selector de corte de pezuña, se ha seleccionado que, sí que se quiere cortar la pezuña del jamón, actúa otro cilindro neumático que lo que realiza es un movimiento de aproximación de una sierra, movida por un sistema de una polea, que es girada por un motor asíncrono trifásico. Este movimiento provoca que la sierra corte la pezuña del jamón. Una vez ha cortado la pezuña, la sierra se para y el cilindro vuelve a su posición inicial.

b. Sin corte de pezuña

En el caso contrario, el corte de la pezuña del jamón, no se realizaría y este paso se lo saltaría.

4.1.3. *SECTOR 3 - PRIMER DESHUESADO*

Para realizar un primer deshuese del jamón, este necesita tener una presión hacia abajo para que el corte sea más preciso, por lo que mediante un sistema de un cilindro neumático con una pinza con pinchos en su base, se tira del jamón hacia abajo para que el operario pueda realizar el primer deshuese. Este movimiento es accionado mediante el otro selector.

La función de este primer deshuese es facilitar la introducción de la siguiente pinza, la cual es explicada en el siguiente sector.

4.1.4. *SECTOR 4 – DESHUESADO PRINCIPAL*

Antes de desactivar el sector tres quitando el selector, el operario introduce una pinza en el jamón que ayudará al operario a seguir deshuesando, esta pinza está unida a una cadena que a su vez está unida a un cilindro neumático que tirará de ambas. Para ello, una vez ha introducido esta pinza, quitará el selector del sector tres y quitará la pinza de este; y pulsará el botón de tiro de cadena, el cilindro comenzará a tirar de la cadena y de la pinza y el operario podrá terminar de deshuesar el jamón por completo.

Una vez haya terminado de deshuesar el jamón, pulsará el botón de despresurización para que pueda volver a repetirse todo el ciclo.

Fuera de estos sectores hay un movimiento que solo se realiza cuando se enciende por primera vez la máquina y es realizado por un pequeño cilindro neumático, este lo que realiza es generar tensión en la cadena del agarre del jamón del sector cuatro.

4.1.5. SECTOR 5 – ELEVACIÓN MÁQUINA

La máquina tiene que poder estar en diferentes alturas, ya que cada operario mide diferente y debe tener la máquina ajustada a su altura, por lo que la altura de la máquina es regulada por otro cilindro neumático y ajustada por dos botones.

4.2. DESCRIPCIÓN GENERAL DE LA MÁQUINA

El esquema general de la máquina antigua sería el siguiente:

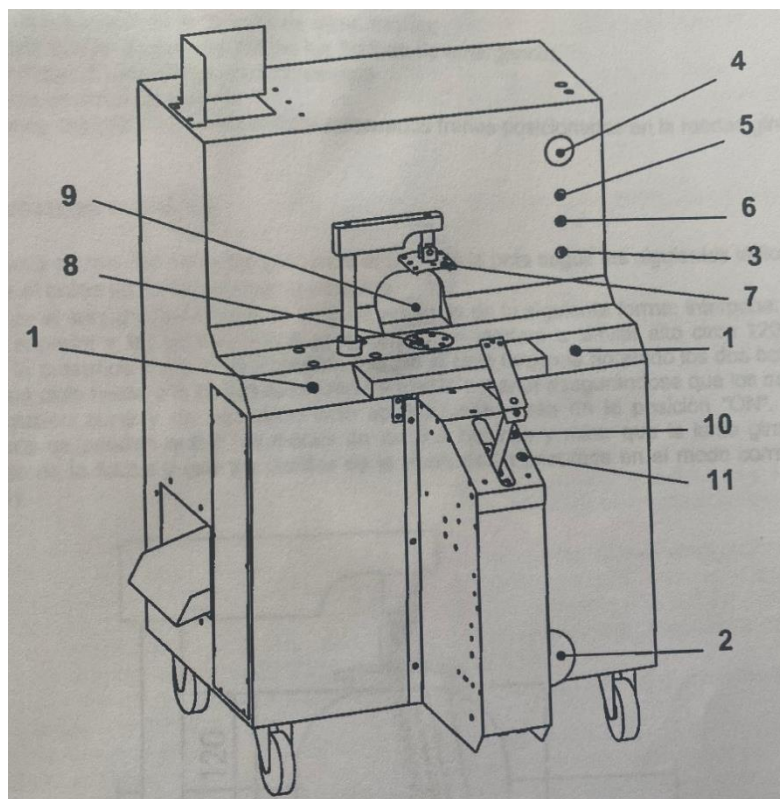


Ilustración 3. Descripción general de la máquina

1. Botón doble de start ciclo.
2. Botón de reset de ciclo.
3. Selector exclusión corte.
4. Manómetro.
5. Botón de restablecimiento de la máquina.
6. Seta de emergencia.
7. Platillo superior.
8. Puntas inferiores.
9. Túnel entrada para jamón.
10. Botón tracción pinza.
11. Selector exclusión ciclo aplastamiento.

La posición de trabajo del operador con la pata de jamón introducida sería la siguiente:

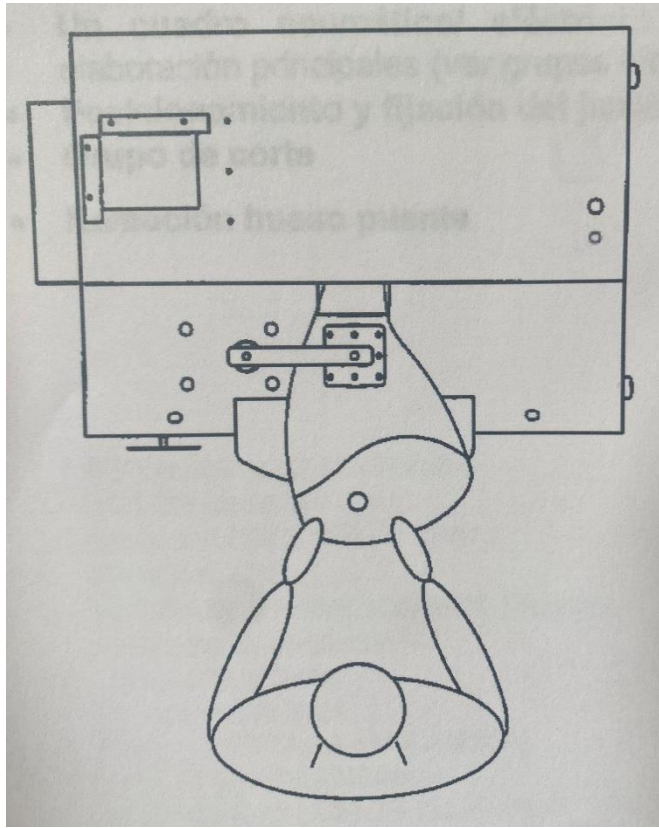


Ilustración 4. Posición de trabajo del operador

4.3.PASOS REALIZADOS

4.3.1. ANÁLISIS Y DESMONTAJE

Se llevó a cabo el desmontaje de las piezas neumáticas y los cilindros hidroneumáticos del sistema anterior. Además, se realizó un análisis detallado del esquema neumático anterior para comprender su funcionamiento y la interacción entre los componentes. Esto proporcionó una base sólida para el diseño e implementación de las mejoras en el sistema, asegurando una transición fluida y eficiente hacia el nuevo diseño y funcionamiento del sistema.

4.3.2. CÁLCULO DE CILINDROS

Se realizó un análisis detallado para determinar la fuerza requerida por los cilindros eléctricos para garantizar un funcionamiento óptimo del sistema. Esto implica consideraciones como la carga a mover, la velocidad deseada y otros factores relevantes para el rendimiento del sistema.

4.3.3. *MECANIZADO DE ACTUADORES*

Se llevaron a cabo procesos de mecanizado para adaptar y preparar los actuadores según las especificaciones requeridas para su instalación y funcionamiento dentro del sistema. Esto puede incluir la fabricación de componentes adicionales o modificaciones en los actuadores existentes para garantizar su correcto funcionamiento.

4.3.4. *MECANIZADO EN MÁQUINA*

Para el acoplamiento del armario de acero inoxidable: Se realizaron operaciones de mecanizado en una máquina específica, en este caso la TGS-150, para preparar el acoplamiento del armario de acero inoxidable al sistema. Esto puede incluir perforaciones, roscado u otros procesos de mecanizado necesarios para la integración adecuada del armario.

4.3.5. *CABLEADO INTERIOR DEL ARMARIO, MOTORES Y PULSATERIA Y SEÑALIZACIÓN*

Se llevó a cabo la instalación y conexión de cables eléctricos dentro del armario, así como la conexión de los motores y dispositivos de control. Además, se realizaron labores de señalización para indicar el estado y funcionamiento del sistema.

4.3.6. *MONTAJE DEL ARMARIO DE PVC*

Para la unidad de mantenimiento combinada y el bloque de electroválvulas: Se ensambló el armario de PVC que alberga la unidad de mantenimiento combinada y el bloque de electroválvulas, asegurando su correcta ubicación y fijación dentro del sistema.

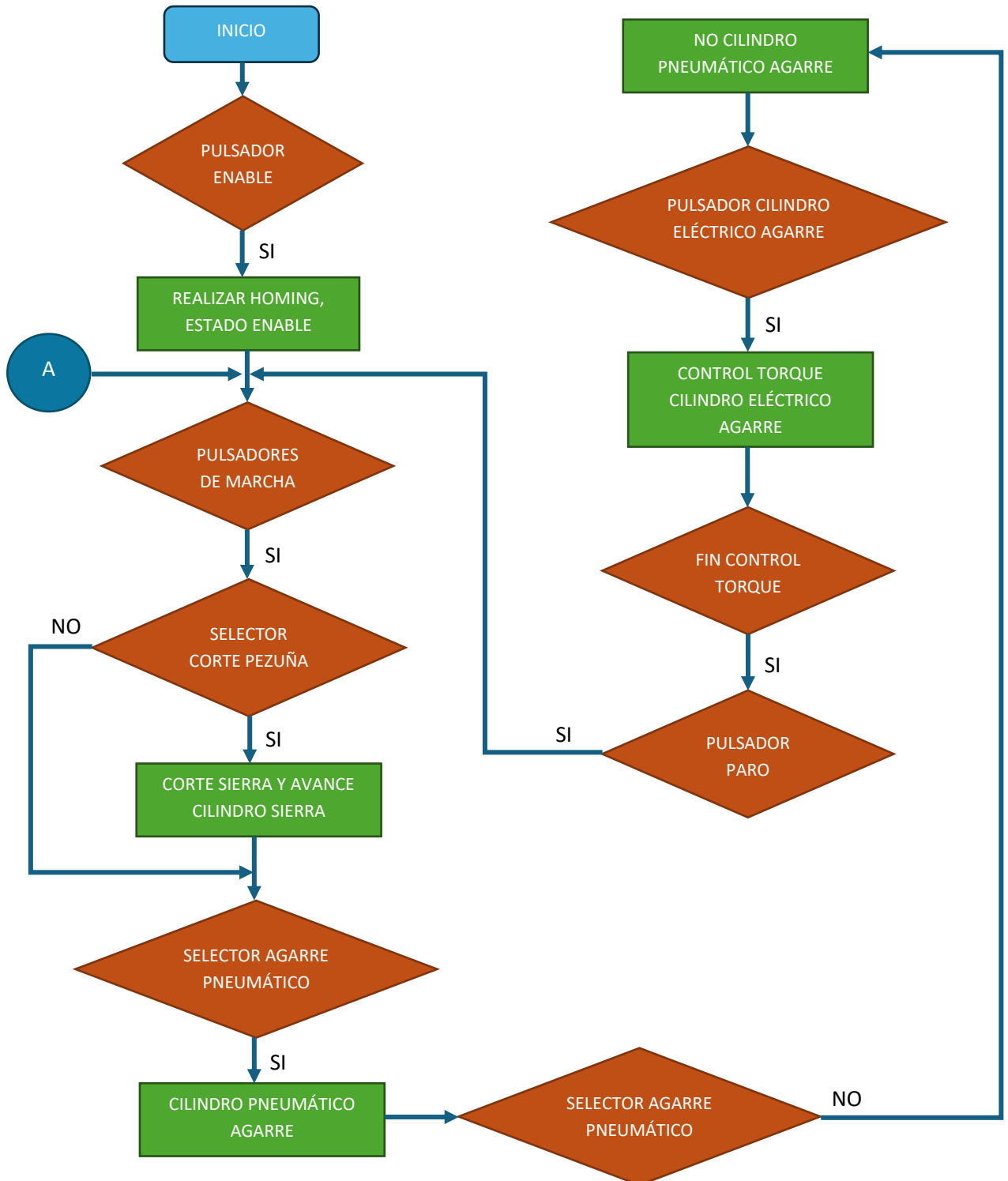
4.3.7. *PROGRAMACIÓN*

De drivers, variador de frecuencia, PLC y diseño de HMI: Se llevó a cabo la programación de los controladores de motor (drivers), el variador de frecuencia, así como del controlador lógico programable (PLC) y la interfaz hombre-máquina (HMI). Esto implica la configuración de parámetros, lógica de control y diseño de la interfaz de usuario para el monitoreo y control del sistema.

4.4. DIAGRAMAS DE PROGRAMACIÓN

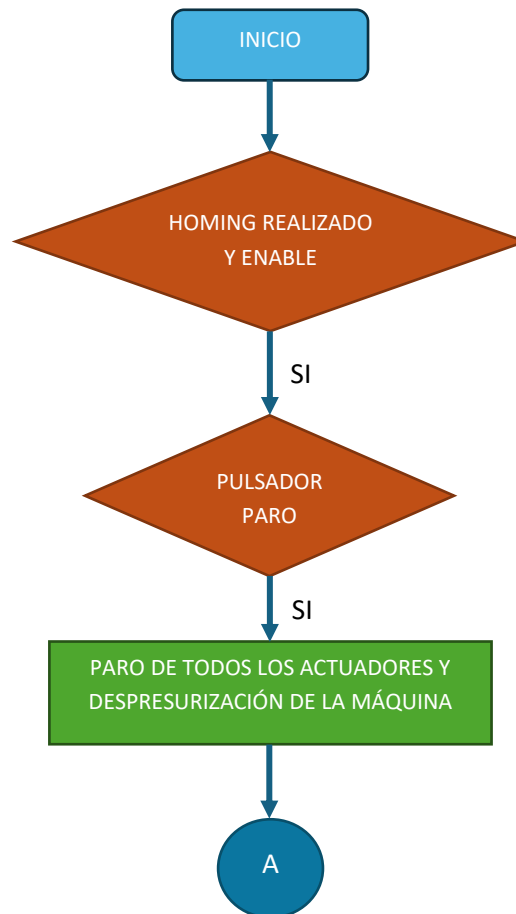
4.4.1. DIAGRAMA PRINCIPAL DE EJECUCIÓN

La secuencia de los procesos de la máquina se ve reflejada en el siguiente diagrama:



4.4.2. *DIAGRAMA PARO*

Para parar el proceso en cualquier momento habría que seguir el siguiente diagrama:



Para la realización de este diagrama se ha programado en el entorno de programación de CODESYS. Se ha hecho uso de un programa principal y varios subprogramas, se muestran en la siguiente imagen:

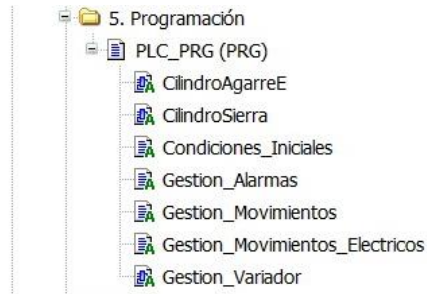


Ilustración 5. Programa principal y subprogramas

Los pasos explicados en la programación son los siguientes:

4.4.2.1. Enable y condiciones iniciales de funcionamiento

Una vez se le da al Pulsador de Enable el PLC ejecuta el subprograma de [Condiciones Iniciales](#). En este subprograma se ejecuta la secuencia que pone la máquina lista para empezar a ejecutarse:

1. Los cilindros neumáticos se mueven a sus posiciones iniciales.
2. Se ponen los cilindros eléctricos en Enable para ello se ejecutarán continuamente dos subprogramas que ejecutan las comunicaciones entre el PLC y las controladoras de los dos cilindros eléctricos ([CilindroAgarre E](#) y [CilindroSierra](#)).
3. Una vez están los cilindros en Power, se realiza el Homing de los dos cilindros eléctricos.
4. A continuación, el cilindro agarre se mueve hasta su posición inicial (no la del homing, si no, la que se introduce a través del HMI para empezar siempre desde ahí el movimiento).
5. Para terminar, se comprueba que la presión de aire en la máquina es la correcta. Para comprobarla, se mide la presión a través del presostato, que es la variable analógica de la que disponemos. El valor de la presión es una interpolación del valor que se lee, para realizar la interpolación se hace en el [PLC PRG](#).
6. Si se realiza toda la secuencia de manera correcta, las condiciones iniciales se consideran correctas y se puede pasar a la siguiente fase de la programación.

4.4.2.2. Gestión movimientos neumáticos y eléctricos

Si se cumple el subprograma de condiciones iniciales y se pulsa a los dos pulsadores de marcha (por seguridad se dispone de dos pulsadores que tienen que estar pulsados simultáneamente para que el operario no introduzca las manos en la sierra por equivocación), comienza el subprograma de [Gestion Movimientos](#). La secuencia sería la siguiente:

1. El cilindro neumático CilindroN_Pata, agarra la para para que esta no se mueva durante todos los procesos. El esquema del cilindro es el siguiente:

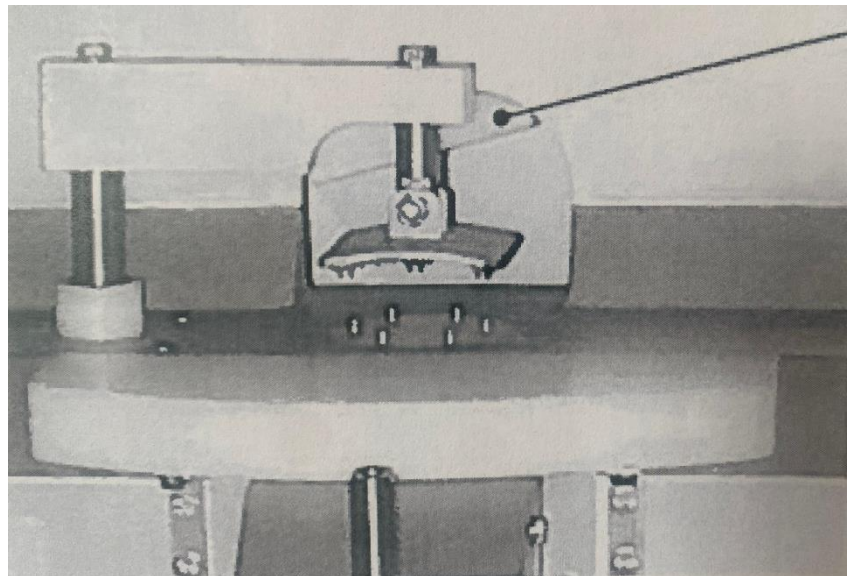


Ilustración 6. CilindroN_Pata

2. Una vez ha llegado a sujetar la pata, se confirma que hay un jamón sujeto, para ello hay un sensor que indica que el cilindro neumático no ha llegado al final de su recorrido. De hacerlo no continuaría el proceso y saltaría una alarma.
Si sí que se ha detectado que hay un jamón sujetado, salen unos pinchos que agarran la pata desde la base, estos son ejecutados por un pequeño cilindro neumático.
3. Si está el selector de corte de jamón (Selector_Discriminacion_Corte), se ejecuta el subprograma [Gestion Movimientos Electricos](#). La secuencia sería la siguiente:

1. La sierra comienza a girar, para ello se ejecuta el subprograma Gestion_Variador y la función fb_Control_Variador, una vez se ha dado un tiempo para que la sierra alcance la velocidad requerida, el cilindro eléctrico avanza en modo de control de par. El esquema es el siguiente:

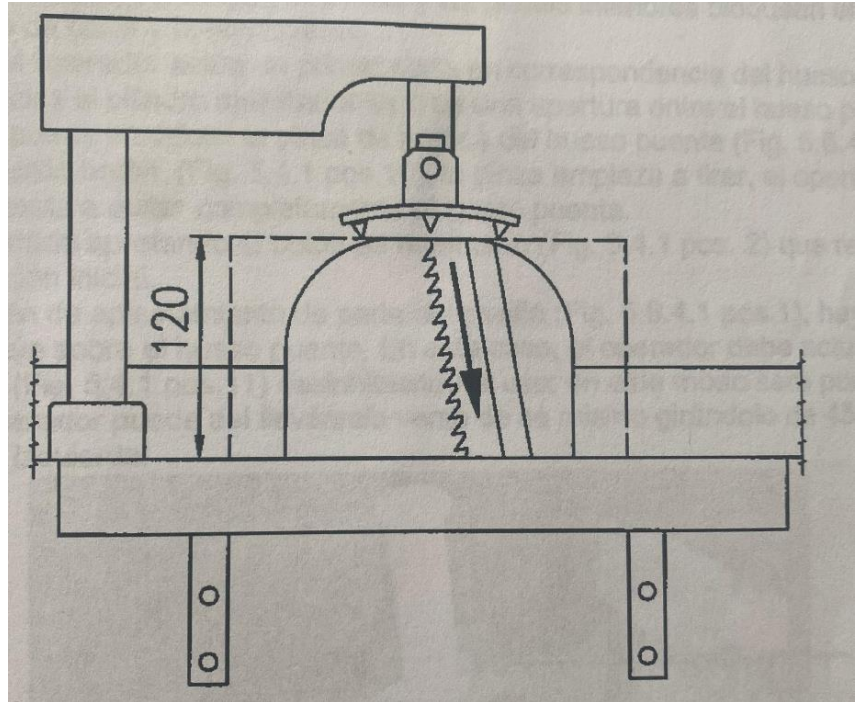


Ilustración 7. Funcionamiento sierra y cilindro

2. Cuando el cilindro ha llegado a la posición final se paran el cilindro y la sierra.
3. Una vez se ha parado, se devuelve el cilindro a su posición inicial.
4. Se vuelve al subprograma Gestion_Movimientos.

Si no está seleccionado no se ejecutaría el subprograma Gestion_Movimientos_Electricos y continuaríamos el proceso del subprograma Gestion_Movimientos.

4. Si se activa el SelectorExtremoJamonOk, este actúa para ayudar al operario a empezar a deshuesar el jamón.
5. Una vez se ha terminado de hacer un primer corte en la pata del jamón, el operario desactiva el SelectorExtremoJamonOk para que el cilindro deje de actuar.
6. Si se activa el pulsador del Boton_Cadena, se vuelve al subprograma de Gestion_Movimientos_Electricos para ejecutar el Cilindro_Agarre_Electrico. La secuencia sería la siguiente:
 1. Comienza el control de par del Cilindro_Agarre_Electrico. Una vez ha llegado a la posición final o si se vuelve a pulsar el Boton_Cadena, se para el cilindro.
 2. Cuando se ha parado, el cilindro vuelve a su posición inicial.

3. Se vuelve al subprograma `Gestion_Movimientos`.
7. Una vez se pulse al `Boton_Despresurizacion` (Pulsador de paro), este suelta el `CilindroN_Pata`, los pinchos y nos suma uno en el contador de jamones.
8. Con esto el proceso habría terminado y para volver a ejecutarlo habría que volver a activar los pulsadores de marcha.

4.4.2.3. Gestión alarmas

Si en algún momento del proceso hubiese alguna alarma se ejecutaría el subprograma de [Gestion Alarmas](#), además de ejecutarse la ejecución de las alarmas del [Alarm Configuration](#) propias del PLC.

4.4.2.4. Gestión HMI

La ejecución del HMI se hace mediante dos programas [HMI](#) y [Mov Sim HMI](#), los cuales ejecutan los diferentes pulsadores, potenciómetro, contadores, simulación de movimientos en el HMI y gestión de usuarios del HMI.

4.4.2.5. Gestión paro

El diagrama del paro de la maquina se ejecuta en el PLC_PRG, como se muestra en la siguiente imagen:

```
IF NOT Sensor_Boton_Despresurizacion OR Stop_HMI THEN
  Marcha_OK := FALSE;
  NoRepetir := FALSE;
  FinCorte := FALSE;
  Pinchos := FALSE;
  CilindroN_Pata := FALSE;
  Enable_HMI := FALSE;
  Marca_CilindroAgarreE := FALSE;
  Marca_CilindroSierra := FALSE;
  Marca_CilindroAgarreE_Fin := FALSE;
  Marca_CilindroSierra_Fin := FALSE;
  Motor_Sierra_Marcha := FALSE;
  CilindroN_Agarre := FALSE;
  Torque_Execute_Sierra := FALSE;
  Torque_Execute_Agarre := FALSE;
  Reset_Execute := FALSE;
  Temporizador_Corte.IN := FALSE;
  Temporizador_Marca_Sierra.IN := FALSE;
  Stop_Execute_Agarre := FALSE;
  Stop_Execute_Sierra := FALSE;
  i := 0;
  j := 0;
  k := 0;
  n := 0;
  Marca_Despresurizacion := TRUE;
  IF EtherCAT_Master.xDistributedClockInSync AND EtherCAT_Master.xConfigFinished THEN
    Temporizador_Operario.IN := TRUE;
  END_IF
END_IF
```

Ilustración 8. Ejecución diagrama paro

4.5. ELEMENTOS DE CONTROL UTILIZADOS



DISPOSITIVO	MODELO	IMAGEN	DESCRIPCIÓN
PLC	B-PRIMIS- ET- PRO-7		PLC con HMI integrado de 7 pulgadas; permite la comunicación en multitud de buses, en este caso se ha utilizado el Ethercat
DRIVERS	CMMT-AS-MP- S1-EC		Controladora de servomotores con comunicación en Ethercat. En el proyecto se ha hecho uso de dos de ellas.
MÓDULO ENTRADAS SALIDAS	CPX-E-EC		Cabecera más un módulo de entradas más un módulo de salidas con comunicación en Ethercat
VARIADOR DE FRECUENCIA	COMANDER- C300		Variador de frecuencia de motores trifásicos asíncronos. Se le ha añadido una cabecera para la comunicación con el PLC en el entorno de Ethercat

Tabla 1. Elementos de control utilizados

4.6. ELEMENTOS DE ACTUACIÓN ELÉCTRICA UTILIZADOS







DISPOSITIVO	MODELO	IMAGEN
CILINDRO ELÉCTRICO AGARRE	ESBF-BS-40-300-5P	
MOTOR CILINDRO ELÉCTRICO AGARRE	EMMT-AS-60-M-LS- RS	
KIT DE MONTAJE CILINDRO ELÉCTRICO AGARRE	EAMM-U-70-D40- 60P-96-S1	
CILINDRO ELÉCTRICO SIERRA	ESBF-BS-40-100-5P	
MOTOR CILINDRO ELÉCTRICO SIERRA	EMMT-AS-60-M-LS- RS	
KIT DE MONTAJE CILINDRO ELÉCTRICO SIERRA	EAMM-U-70-D40- 60P-96-S1	

Tabla 2. Elementos de actuación eléctrica utilizados

4.7.ELEMENTOS DE ACTUACIÓN PNEUMÁTICA UTILIZADOS





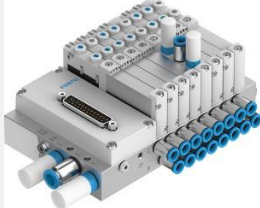
DISPOSITIVO	MODELO	IMAGEN
CILINDRO_N_PINCHOS	ADN-S-50-15-I-P-A	
CILINDRO_N_TENSOR	DSNU-25-50-P-A	
CILINDRO_N_AGARRE	DSBC-50-150-PPSA-N3	
CILINDRO_N_PATA	DSBC-50-150-PPSA-N3	
TERMINAL VÁLVULAS	VTUG	

Tabla 3. Elementos de actuación neumática utilizados

4.8. CUADRO ELÉCTRICO FINAL

4.8.1. PARTES CUADRO ELÉCTRICO

El cuadro eléctrico de control es el siguiente:

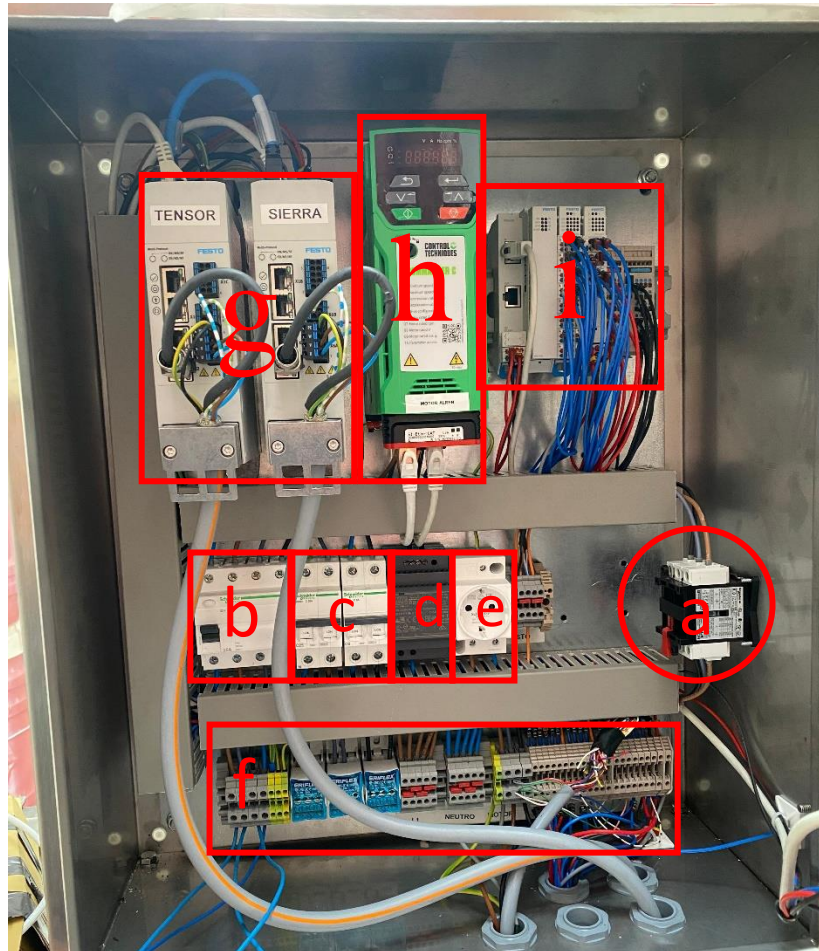
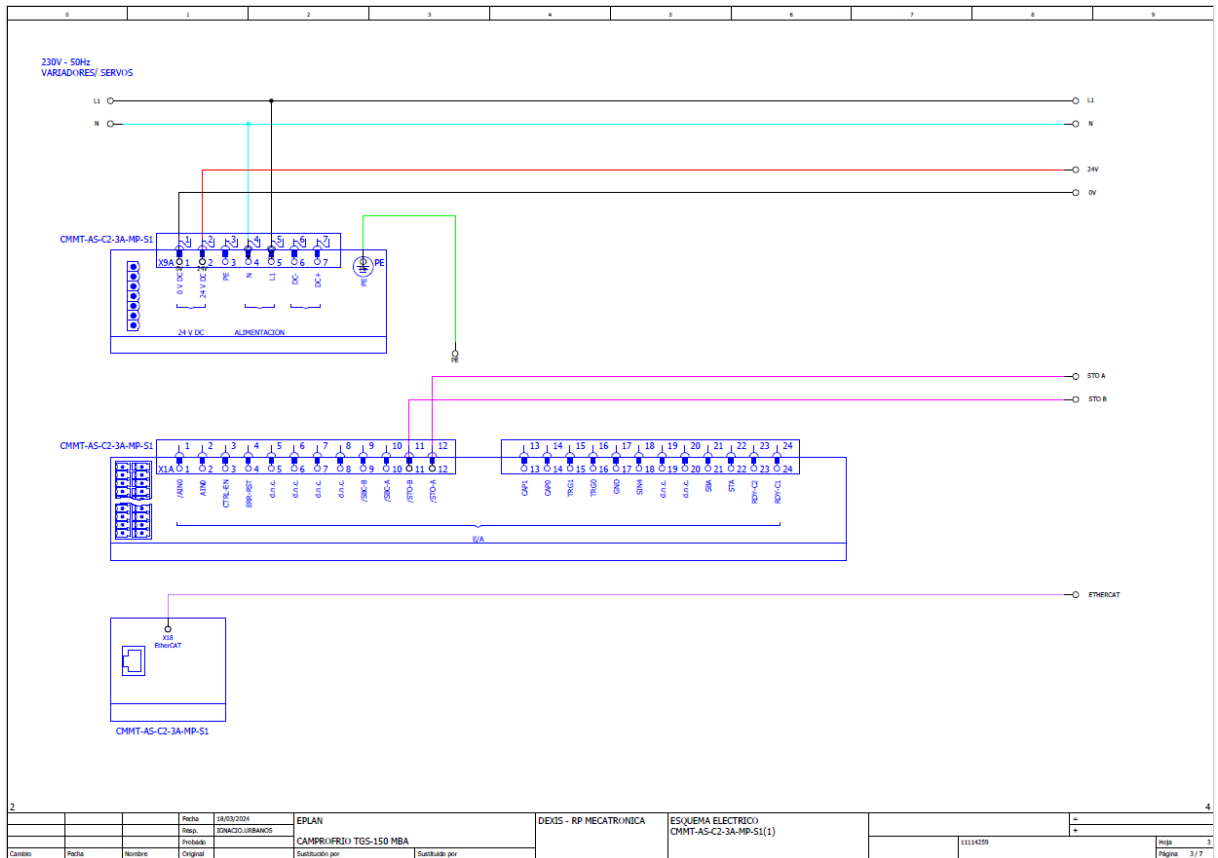
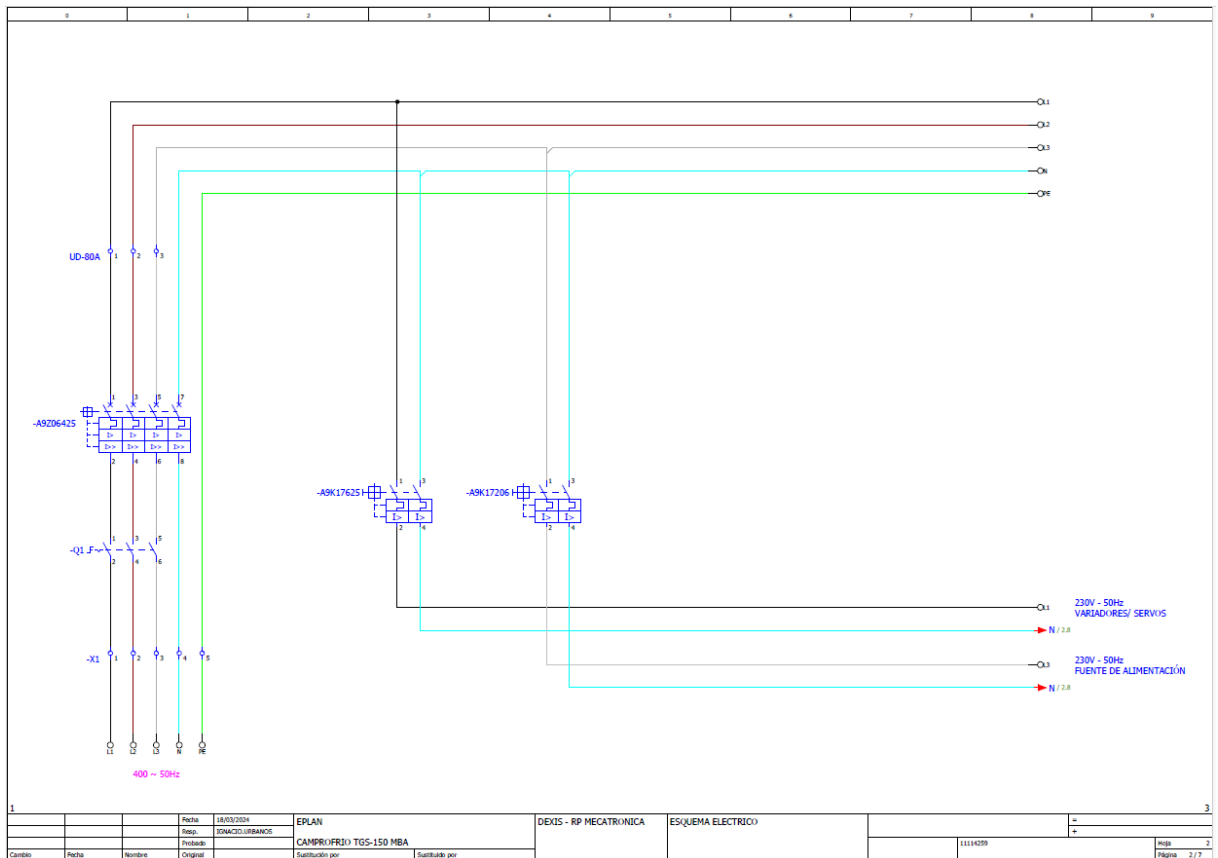
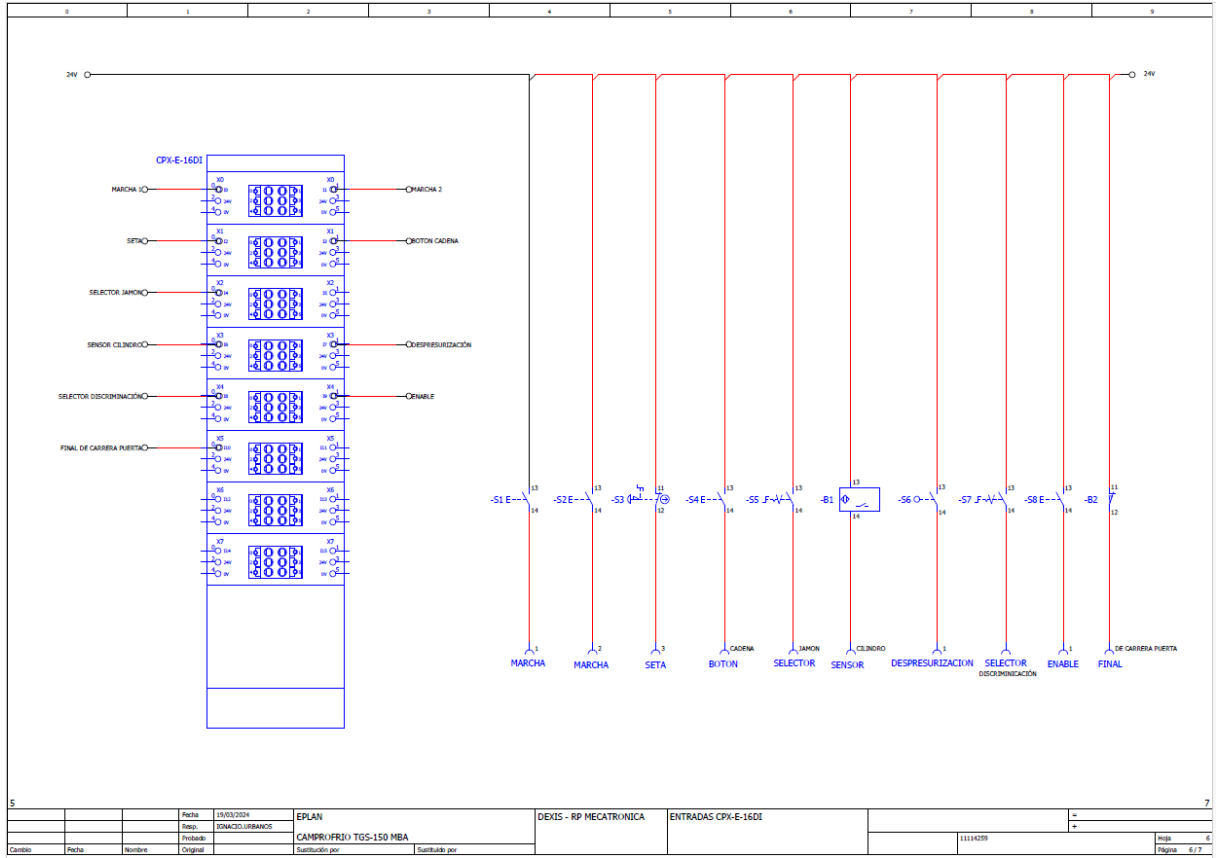
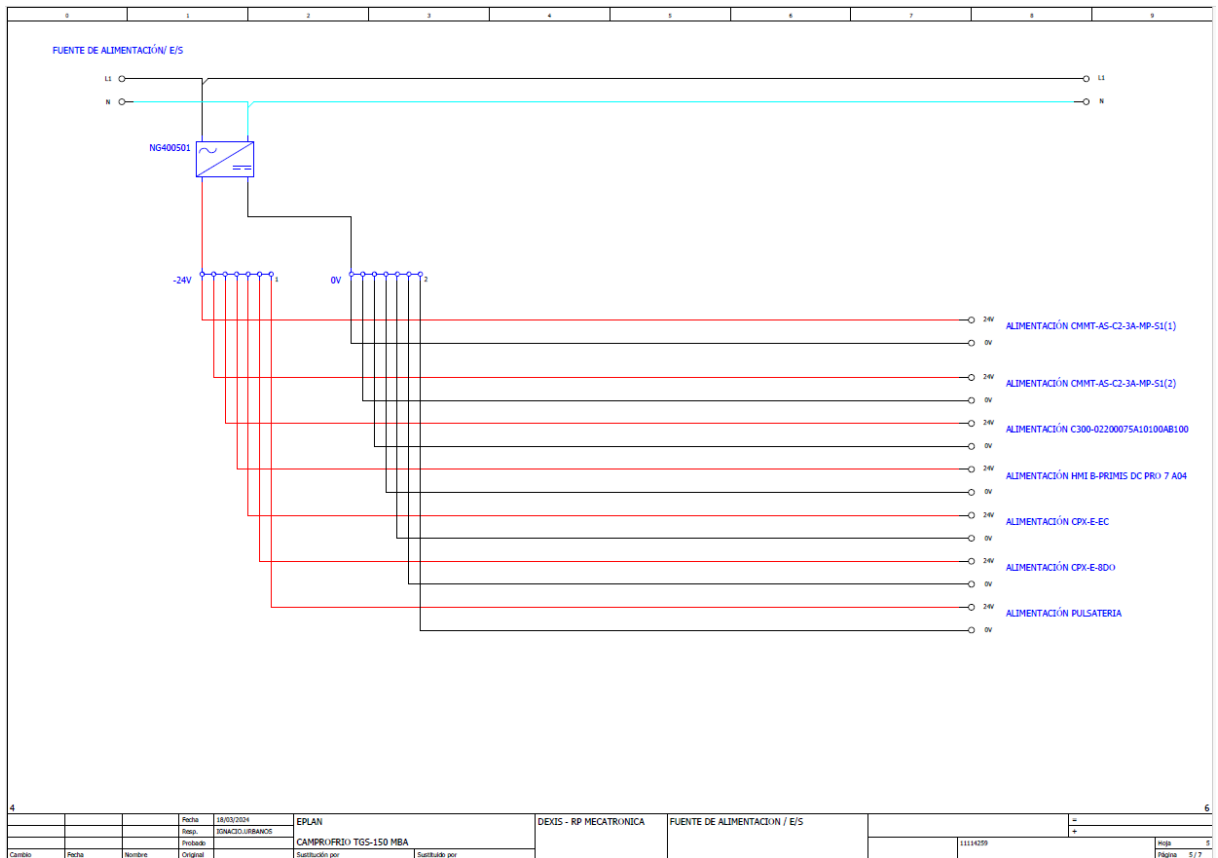


Ilustración 9. Cuadro eléctrico final

Los elementos eléctricos utilizados en el cuadro son:

- a) Un seccionador para cortar al entrada de potencia al cuadro.
- b) Un diferencial para proteger la instalación frente a cortocircuitos y a las personas de sufrir descargas eléctricas.
- c) Dos magnetotérmicos para proteger la instalación frente a sobrecargas o cortocircuitos.
- d) Fuente de 230 a 24V, ya que los drivers, el PLC y el módulo de E/S funcionan a 24V.
- e) Un tomacorriente para conexiones futuras.
- f) Bornas.
- g) Controladoras ejes eléctricos.
- h) Variador de frecuencia para el control de la sierra.
- i) Módulo de entradas/salidas.





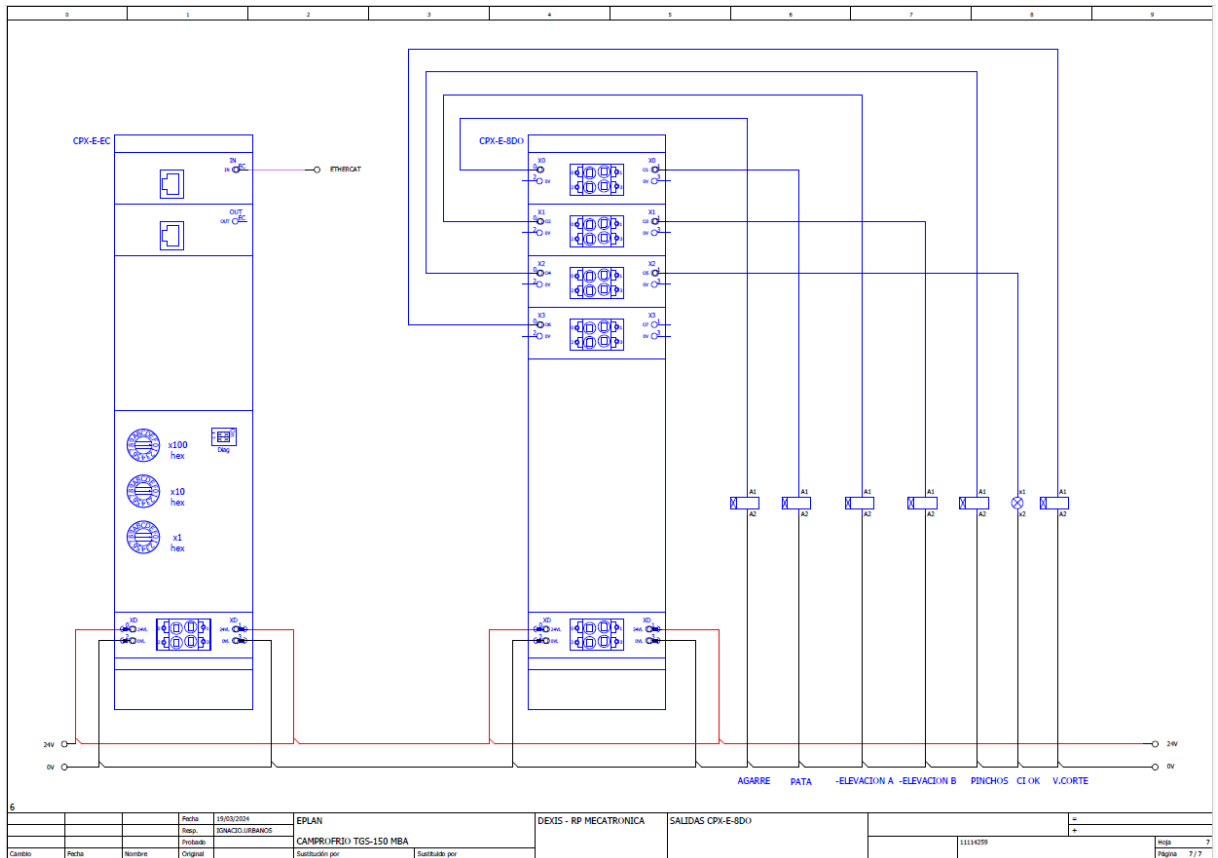


Ilustración 10. Esquemas cuadro eléctrico

5. CÁLCULO DE LOS EJES Y MOTORES ELÉCTRICOS

5.1. CÁLCULO CON EL PROGRAMA POSITIONING DRIVES

En este apartado se muestra mediante el cálculo del sistema del cilindro de agarre, para realizarlo a modo de ejemplo para el sistema del cilindro de agarre, cómo se ha usado el programa Positioning Drives, el cual nos da los ejes y motores eléctricos que podemos usar en el proyecto con la información que le hemos introducido.

5.1.1. SELECCIÓN DEL TIPO DE ACTUADOR

Lo primero que se ha realizado es seleccionar si se trata de un actuador lineal o giratorio.

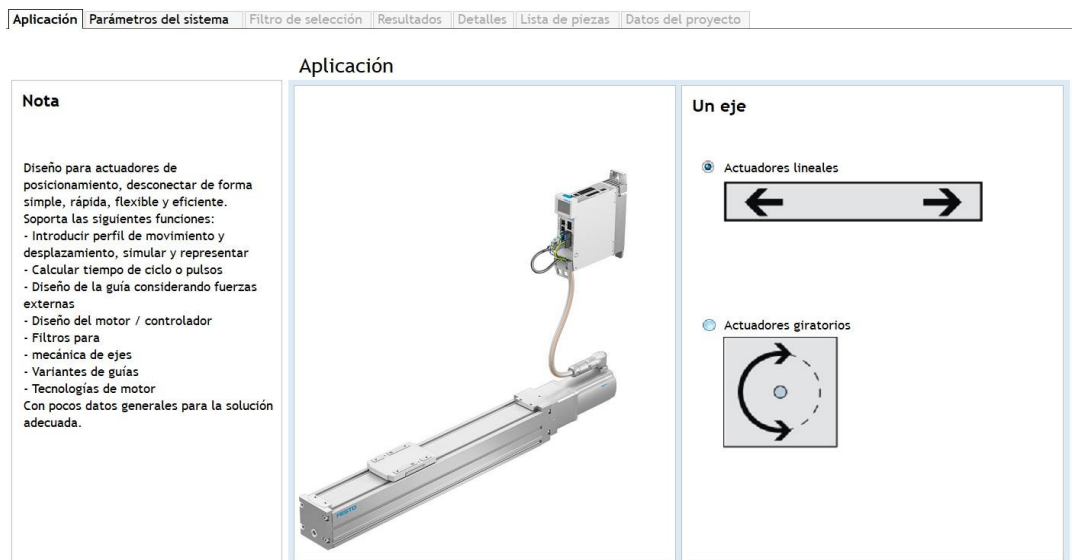


Ilustración 11. Selección del tipo de actuador

5.1.2. PARÁMETROS DEL SISTEMA

En este apartado se deben seleccionar en primer lugar si el eje viene con guía incluida y si se trata de un eje de pórtico o en voladizo. En este caso no es preciso disponer de una guía y es un eje en voladizo, ya que el eje neumático anterior disponía de una serie de características que hay que seguir cumpliendo.

En segundo lugar, se ha seleccionado la posición de montaje que va a soportar la carga. En este mismo apartado se introduce la carga máxima que se espera, la carrera útil del eje, la precisión que debe tener y la fuerza adicional externa que va a soportar.

En tercer lugar, se han introducido los tiempos de ciclo máximos, las condiciones térmicas a las que normalmente va a estar funcionando y si el movimiento del eje es a través de una correa dentada o de un usillo.

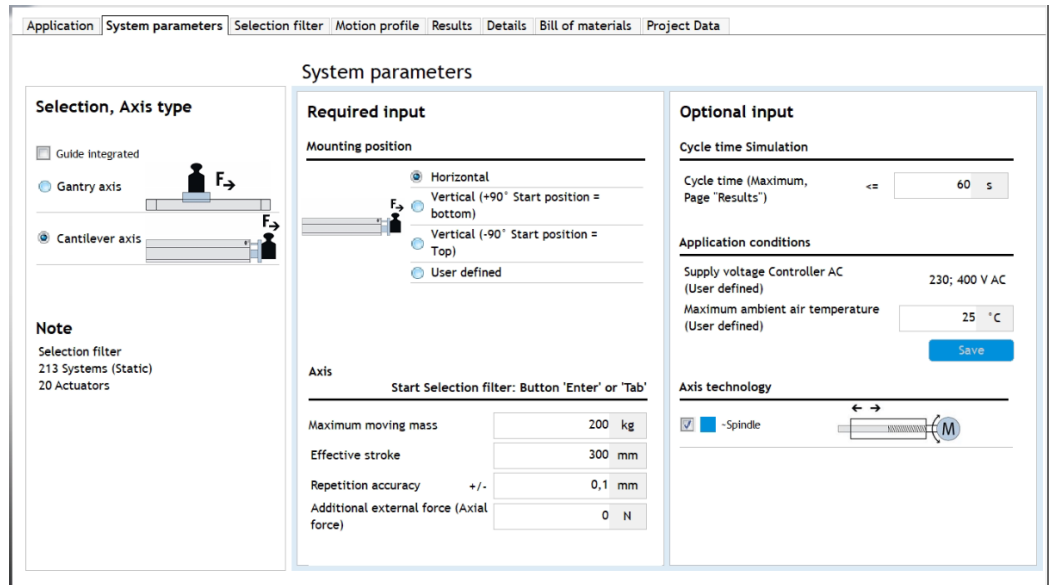


Ilustración 12. Parámetros del sistema

5.1.3. SELECCIÓN DE EJES, MOTORES Y CONTROLADORAS

En este apartado se seleccionan los ejes que vemos que podemos necesitar, con su tipo de guía, en la siguiente ilustración.

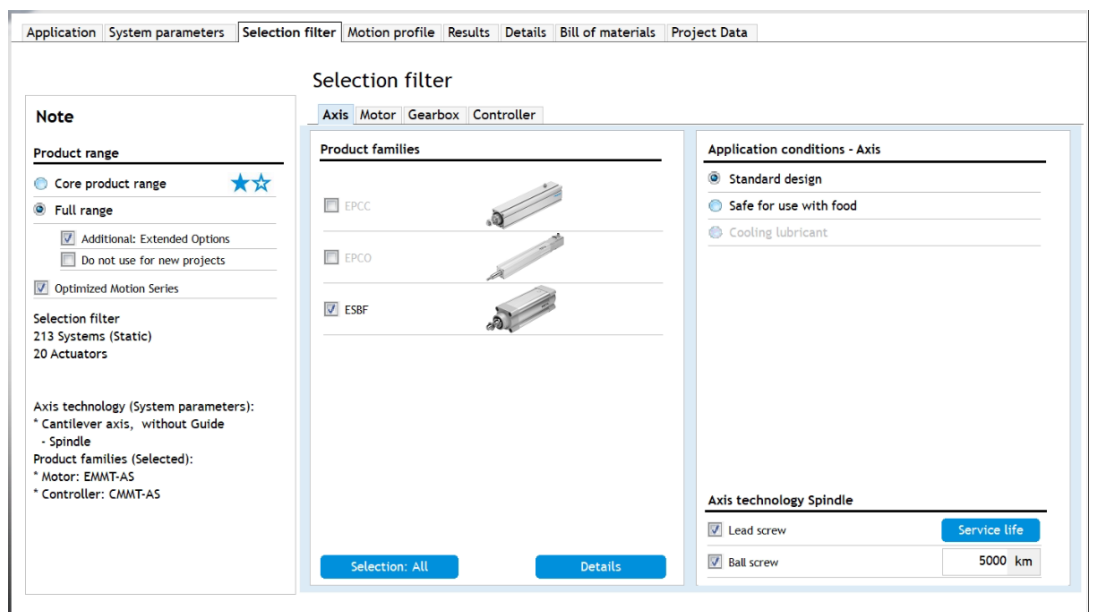


Ilustración 133. Selección ejes

Los motores, que podemos seleccionar si queremos que sean servomotores, motores paso a paso o con controladora integrada (Servomotor EC); si lleva o no freno o el tipo de montaje, lo vemos en la siguiente ilustración.

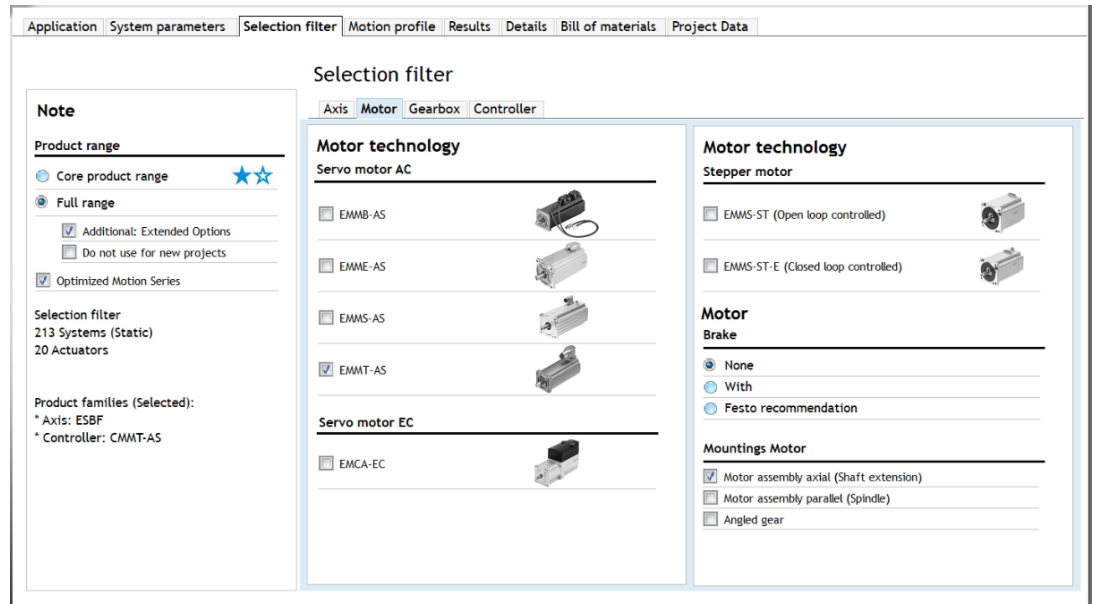


Ilustración 14. Selección motores

Los tipos de engranajes y si lleva o no reductora, lo vemos en la imagen siguiente.

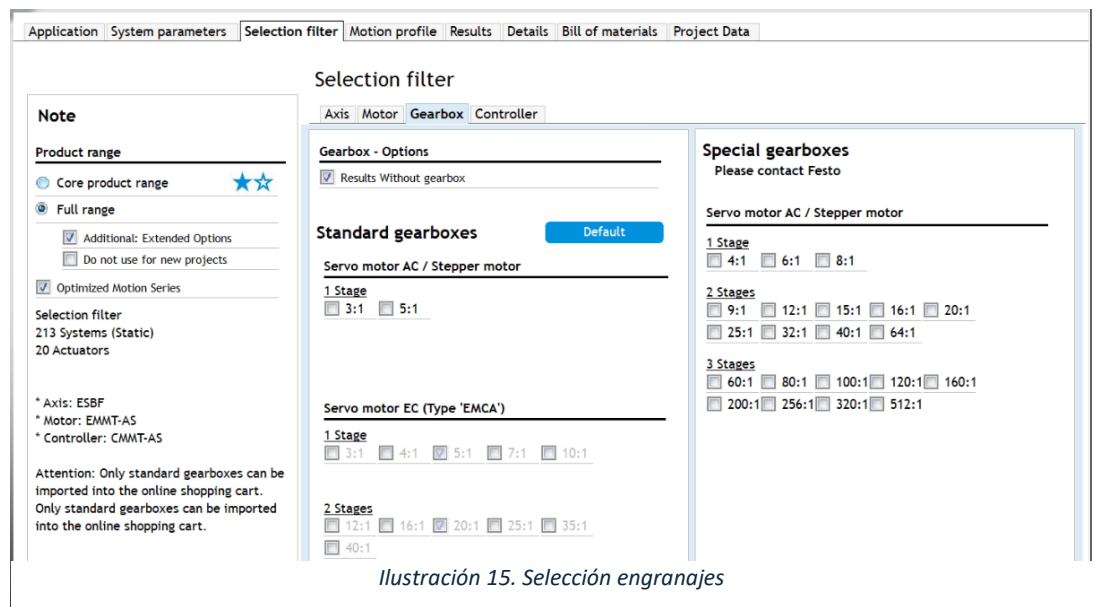


Ilustración 15. Selección engranajes

Y la controladora que puede llevar, si es para servomotores o si es para motores paso a paso, el tipo de tensión de alimentación, el bus de campo, en nuestro caso Ethercat y la longitud del cable, lo vemos en la siguiente ilustración.

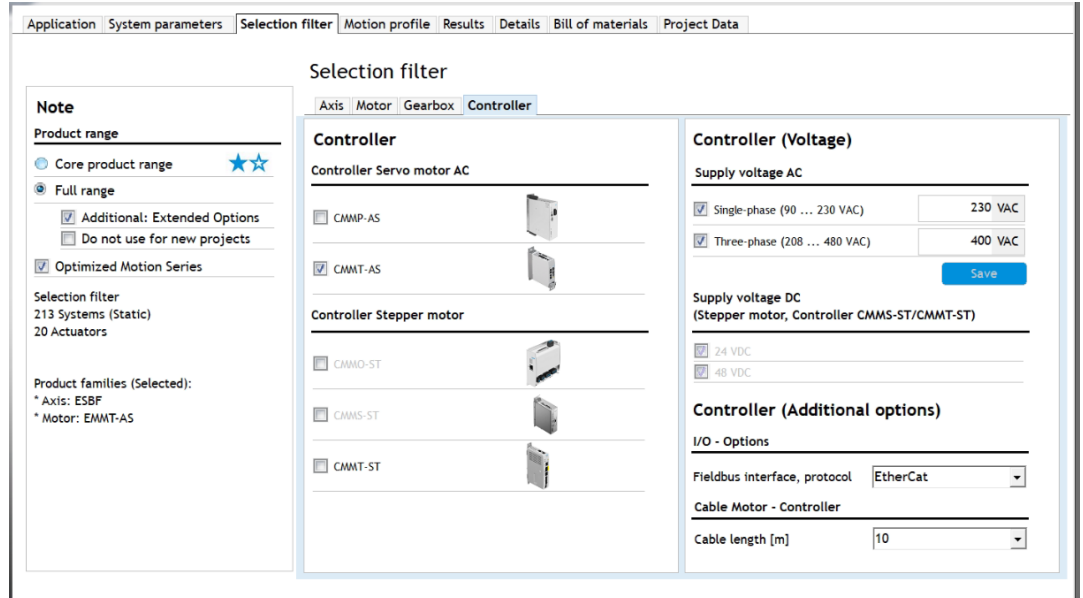


Ilustración 16. Selección de drivers

En la siguiente ilustración aparecen todos los resultados compatibles con la selección de valores que se han introducido. El que más se ajusta a las necesidades del proyecto es el siguiente:

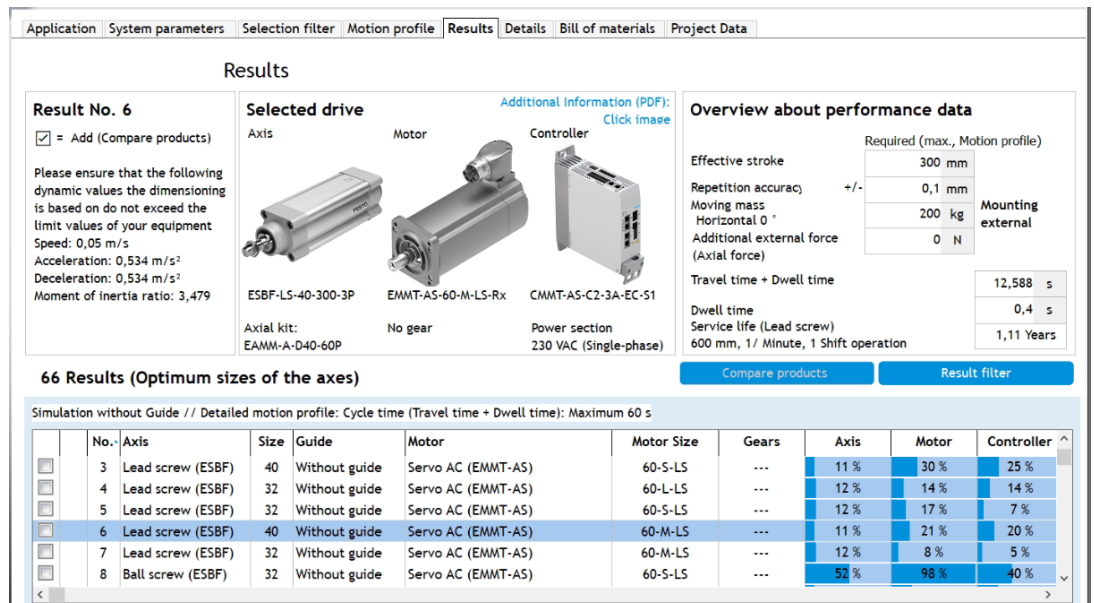
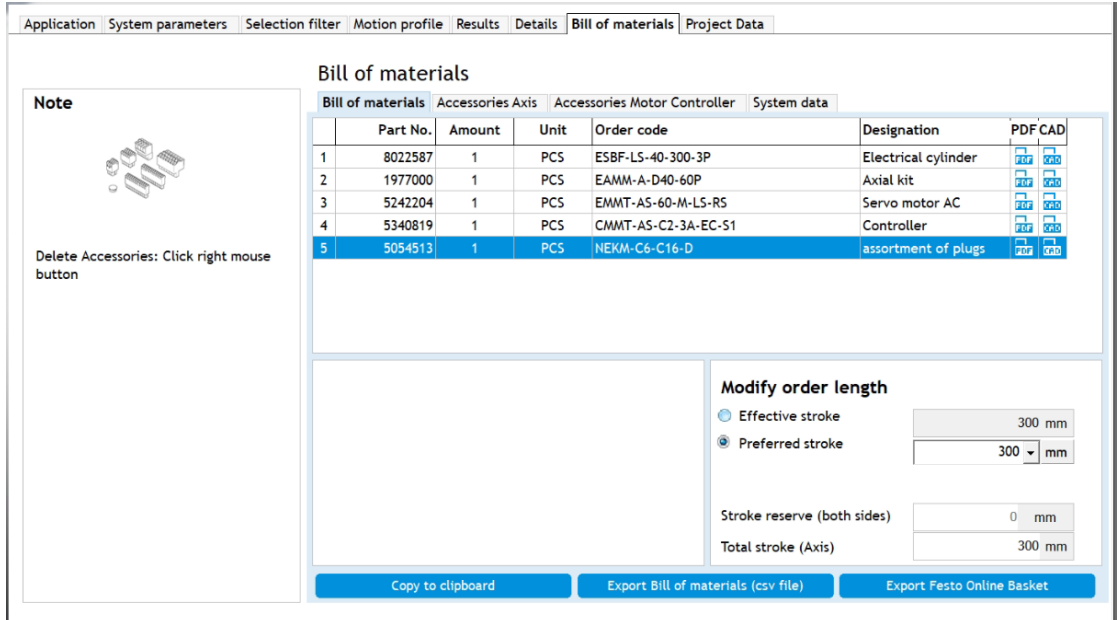


Ilustración 17. Resultados cilindro agarre

El programa da un listado de componentes y la opción de llevarlo directamente a la cesta de la compra de la página web de FESTO, se muestra en la siguiente ilustración.



Bill of materials

Part No.	Amount	Unit	Order code	Designation	PDF CAD
1	1	PCS	ESBF-LS-40-300-3P	Electrical cylinder	PDF CAD
2	1	PCS	EAMM-A-D40-60P	Axial kit	PDF CAD
3	1	PCS	EMMT-AS-60-M-LS-R5	Servo motor AC	PDF CAD
4	1	PCS	CMMT-AS-C2-3A-EC-S1	Controller	PDF CAD
5	1	PCS	NEKM-C6-C16-D	assortment of plugs	PDF CAD

Modify order length

Effective stroke: 300 mm
 Preferred stroke: 300 mm
 Stroke reserve (both sides): 0 mm
 Total stroke (Axis): 300 mm

[Copy to clipboard](#) [Export Bill of materials \(csv file\)](#) [Export Festo Online Basket](#)

Ilustración 18. Listado de piezas

5.1.4. RESUMEN EJES

El eje eléctrico de agarre sería el siguiente:



Selected drive

Axis
 ESBF-LS-40-300-3P
 Axial kit:
 EAMM-A-D40-60P

Motor
 EMMT-AS-60-M-LS-Rx
 No gear

Controller
 CMMT-AS-C2-3A-EC-S1
 Power section
 230 VAC (Single-phase)

[Additional Information \(PDF\): Click image](#)

Ilustración 19. Resultado final cilindro agarre

Realizando el mismo proceso para el otro eje eléctrico, nos daría el siguiente resultado:



Ilustración 20. Resultado cilindro sierra

5.2. CÁLCULO MANUAL

Para asegurarse de que los cálculos realizados son correctos, se ha realizado una tabla de Excel en la que se calculan los valores más importantes para la selección del material necesario. La tabla quedaría de la siguiente manera:

Cilindro	t(s)	D(mm)	V(m/s)	A(m/s ²)	Paso	rpm	F(N)	Rmto	F(Nm)	Red F (Nm)
CilindroSierraE	0,5	100	0,4	1,6	40	600	50	0,81	0,39	0,4
CilindroAgarreE	0,5	300	1,2	4,8	32	2250	200	0,81	1,26	1,3

Tabla 4. Cálculo manual

Los valores que salen en gris en la gráfica son los introducidos manualmente y los amarillos son los calculados mediante las siguientes fórmulas en el Excel. Estas fórmulas se han usado de los apuntes del curso de cálculo de motores y ejes eléctricos del curso impartido por la empresa Intra Automation SL. Las fórmulas que se han utilizado son las siguientes:

- V(m/s):

$$V = \frac{D}{1000} \cdot 2$$

- A(m/s²):

$$A = \frac{V}{\frac{T}{2}}$$

- Rpm:

$$rpm = \frac{V \cdot 60.000}{Paso}$$

- F(Nm):

$$F(Nm) = \frac{F(N) \cdot Paso}{2.000 \cdot \pi \cdot rmto}$$

Para seleccionar la F en Nm, se ha realizado un redondeo de Red F(Nm)

6. CÁLCULO DE LOS EJES PNEUMÁTICOS

En la neumática únicamente se han cambiado dos de los cilindros por ejes eléctricos para tener un mayor control de sus variables (posición, velocidad, torque, aceleración...). Para los cilindros neumáticos se ha hecho un cálculo para elegir los nuevos de la marca FESTO. Lo primero que se ha realizado fue el cálculo de la fuerza que ejercían los cilindros neumáticos viejos. Este cálculo se realizó a través de una página de Excel, en la cual, mediante una fórmula de los apuntes del curso de cálculo de motores y ejes eléctricos de la empresa Intra Automation SL, se ha calculado la fuerza de cada uno de estos (siendo P, presión y S, superficie).

$$F(N) = P \cdot S$$

La superficie, al tratarse de cilindros neumáticos es circular en todo su interior, por lo que hay que poner el área del círculo.

$$\text{Área del círculo} = \pi \cdot r^2$$

Se dispone del diámetro del émbolo como dato inicial. En la fórmula hay que dividir el diámetro entre dos para tener el radio. A continuación, se realizan las conversiones de las unidades para que el resultado sea en Newtons.

$$F(N) = \left(\frac{\text{Diámetro(mm)}}{2} \right)^2 \cdot \pi \cdot \frac{1(\text{cm}^2)}{1000(\text{mm}^2)} \cdot 6 \left(\frac{\text{Kg}}{\text{cm}^2} \right) \cdot 9,8 \left(\frac{\text{N}}{\text{Kg}} \right)$$

Cilindro	Diámetro (mm)	Fuerza (N)
CilindroN_Pinchos	50	1154,5
CilindroN_Tensor	25	288,6
CilindroN_Agarre	50	1154,5
CilindroN_Pata	50	1154,5

Tabla 5. Tabla fuerza ejes neumáticos

Una vez se realizó este cálculo, mirando los catálogos y los manuales de los cilindros neumáticos de FESTO, se hizo una selección manual de los cilindros nuevos que serían sustituidos por los viejos.

El resto de los cilindros neumáticos se han dejado como estaban ya que los acababan de cambiar hace poco y estaban prácticamente nuevos.

7.COMPARATIVA EJES PNEUMÁTICOS Y ELÉCTRICOS

Al terminar con la selección de los cilindros neumáticos nuevos se hizo una tabla de Excel en la que se ve una comparativa entre los cilindros viejos, los cilindros neumáticos nuevos y los ejes eléctricos nuevos, además, se ha añadido el presupuesto de cada uno de ellos.

7.1.EJES PNEUMÁTICOS VIEJOS

Los ejes neumáticos que estaban colocados anteriormente eran los siguientes:

Cilindro	Fuerza (N)	Carrera (mm)
CilindroAgarre	1869,9	300
CilindroSierra	1154,5	100
CilindroN_Pinchos	1154,5	15
CilindroN_Tensor	288,6	50
CilindroN_Agarre	1154,5	150
CilindroN_Pata	1154,5	150
CilindroN_Elevar	1154,5	100

Tabla 6. Características cilindros neumáticos viejos

7.2.EJES PNEUMÁTICOS NUEVOS

Si todos los ejes se cambiasen por cilindros neumáticos, pero nuevos de la marca Festo, se haría la misma selección de cilindros, por lo que la tabla quedaría de la misma manera. Lo que habría que hacer es añadir el precio de cada uno de los cilindros.

Cilindro	Fuerza (N)	Carrera (mm)	Precio
CilindroAgarre	1869,9	300	215,10 €
CilindroSierra	1154,5	100	163,23 €
CilindroN_Pinchos	1154,5	15	83,93 €
CilindroN_Tensor	288,6	50	58,98 €
CilindroN_Agarre	1154,5	150	168,40 €
CilindroN_Pata	1154,5	150	168,40 €
CilindroN_Elevar	1154,5	100	Se mantiene
TOTAL			858,04 €

Tabla 7. Características todos cilindros neumáticos nuevos

7.3.EJES ELÉCTRICOS Y PNEUMÁTICOS NUEVOS

Cambiando algunos ejes neumáticos por eléctricos (solución final) la tabla quedaría de la siguiente manera (los precios incluyen las controladoras, los motores, los ejes y los cables):

EXPLICAR MEJOR

Cilindro	Fuerza (N)	Carrera (mm)	Precio
CilindroE_Agarre	200	300	1.544,18 €
CilindroE_Sierra	50	100	1.569,12 €
CilindroN_Pinchos	1154,5	15	83,93 €
CilindroN_Tensor	288,6	50	58,98 €
CilindroN_Agarre	1154,5	150	168,40 €
CilindroN_Pata	1154,5	150	168,40 €
CilindroN_Elevar	1154,5	100	Se mantiene
TOTAL			3.593,01 €

Tabla 8. Características cilindros neumáticos y eléctricos nuevos

8. COMPARATIVA Y CONSUMOS

8.1. CONSUMO AIRE CILINDROS PNEUMÁTICOS

Para realizar el cálculo del consumo de aire de los cilindros neumáticos se ha realizado gracias a la página web de Festo, en su soporte técnico. La forma de hacerlo es la siguiente:

Cylinder Air Consumption

Operating mode
single acting, pulling


Size *
50 mm

Stroke
15 mm

Inside tubing diameter
8 mm

Length
100 mm

Number of cycles
1 1/min



Add

* NOTE: Using estimated piston rod diameter of 20 mm.

Air Consumption

per Minute per day per Year

0 m³ 0 m³ 0 m³

Energy Costs Energy Saving

Price per day

0.02 € /m³ 0€

Utilization

Working pressure Working hours per day Working days per year

6 bar 8 h 200 d

Selected cylinders and their air consumption Print Copy Clear table

Op. Mode	Size [mm]	Stroke/Length [mm]	Pressure [bar]	Number of cycles [1/min]	Air Consumption per Cycle [l]	Air Consumption per Minute [l]

Ilustración 21. Consumo de aire de los cilindros neumáticos

Introduciendo los datos de todos los cilindros neumáticos la página web nos aporta el siguiente resultado del consumo de los cilindros:

Si se cambian todos los cilindros antiguos por nuevos neumáticos:

Operating mode
single acting, pulling


Size *
50 mm

Stroke
15 mm

Inside tubing diameter
8 mm

Length
100 mm

Number of cycles
1 1/min



Add

* NOTE: Using estimated piston rod diameter of 20 mm.

Air Consumption

per Minute per day per Year

0.02 m³ 19.07 m³ 3814.34 m³

Energy Costs Energy Saving

Price per Year

0.02 € /m³ 76.29€

Utilization

Working pressure Working hours per day Working days per year

6 bar 16 h 200 d

Ilustración 22. Consumo aire todos cilindros neumáticos

Si se cambian algunos de los cilindros neumáticos por eléctricos, el consumo de los neumáticos sería:

The screenshot shows a software interface with three main sections:

- Air Consumption:** Displays consumption rates for 'per Minute' (0.01 m³), 'per day' (5.44 m³), and 'per Year' (1087.43 m³).
- Energy Costs:** Shows a price of 0.02 €/m³ and a calculated cost of 21.75€ per Year. An 'Energy Saving' toggle is visible.
- Utilization:** Lists parameters: Working pressure (6 bar), Working hours per day (16 h), and Working days per year (200 d).

Ilustración 23. Consumo aire algunos cilindros neumáticos

8.2. CONSUMO ENERGÉTICO EJES ELÉCTRICOS

Para realizar el cálculo del consumo eléctrico de los ejes eléctricos se ha realizado usando el programa informático Excel.

Para saber el tiempo que va a estar funcionando cada eje al día se ha usado la siguiente fórmula:

$$T_{Func \text{ al día}} = \frac{\text{Ciclo}}{\text{min}} \cdot \frac{T_{Ciclo}(\text{segundos}) \cdot 60(\text{min}) \cdot 12(\text{meses})}{3600(\text{segundos})}$$

A continuación, se ha calculado el consumo en kW que tiene cada eje al año haciendo uso de la siguiente fórmula:

$$kW(\text{anuales}) = W \cdot T_{Func \text{ al día}} \cdot 22(\text{días}) \cdot 12(\text{meses})$$

Para finalizar, sabiendo cual es el precio del kWh actualmente, se ha realizado el siguiente cálculo:

$$\frac{\text{€}}{\text{año}} = \frac{\text{€}}{kWh_{actual}} \cdot kW(\text{anual})$$

La tabla final con el uso de las fórmulas anteriores es la siguiente:

Cilindro	kW	ciclos/min	T. Ciclo(s)	T. Func al día (horas)	kW(anual)	€/año
CilindroSierraE	0,35	2	2	0,8	73,92	13,31 €
CilindroAgarreE	0,35	2	5	2	184,8	33,26 €
					TOTAL	46,57 €

€/kWh actual	0,18
---------------------	-------------

Tabla 9. Cálculo energético ejes eléctricos

8.3. SOSTENIBILIDAD DEL RETROFIT

La realización de este retrofit contribuye a la sostenibilidad, ya que, la realización de retrofit impulsa la generación de empleo (se necesitan puestos de trabajo ya sea para la implementación como para el mantenimiento de la máquina), mejora las condiciones laborales (mejora de los equipos, de la eficiencia, haciendo un entorno de trabajo más seguro y cómodo para los trabajadores); se reduce el consumo energético (al hacer uso de ejes eléctricos en lugar de neumáticos), disminuyendo así las emisiones contaminantes y haciendo más eficiente el uso de los recursos; se ahorra en costos operativos (modernizando los equipos se consigue una mayor eficiencia, reduciendo costos de operación y mantenimiento y consumo de energía), se prolonga la vida útil de las máquinas (ya que no se cambia completamente, si no que se reemplazan algunos de sus equipos).

9. PRESUPUESTO

El presupuesto en material, en montaje y en programación es el siguiente:

Presupuesto		
Material	Nº	Precio
CilindroE_Agarre	1	1.544,18 €
CilindroE_Sierra	1	1.569,12 €
CilindroN_Pinchos	1	83,93 €
CilindroN_Tensor	1	58,98 €
CilindroN_Agarre	1	168,40 €
CilindroN_Pata	1	168,40 €
CilindroN_Elevar	1	- €
PLC	1	562,00 €
Módulo Entradas/Salidas	1	240,32 €
Variador de frecuencia	1	413,25 €
Terminal de Válvulas	1	298,32 €
Protecciones Eléctricas	1	514,13 €
Armario	1	456,21 €
Cables Ethernet	7	25,99 €
Programación	25	1.375,00 €
Montaje	7	490,00 €
	TOTAL	7.968,23 €

Tabla 10. Presupuesto

10. CONCLUSIONES

Como conclusión de este proyecto, se ha terminado por completo la programación del PLC y del HMI, el montaje de todas las partes de la máquina y se ha realizado la puesta en marcha de la máquina en las instalaciones del cliente.

Habría que añadir que, como mejora de la máquina se le ha ofrecido al cliente un Switch de comunicaciones con red wifi, por el que podrá acceder tanto a la programación del PLC y del HMI; a la visualización de este, datos de la máquina en tiempo real (registro de datos), desde cualquier lugar que disponga de red wifi. Esto se le ha ofrecido al cliente para que esta deje de ser una máquina sin ningún tipo de comunicación con el exterior a ser una máquina que podría incluirse en la tecnología de la industria 4.0.

La realización de este retrofit contribuye a las tres dimensiones de la sostenibilidad, ya que, aporta en el tema social, económico y ambiental.

En cuanto a los objetivos cumplidos, tal y como se ha planteado en el proyecto, se ha realizado un retrofit seleccionado los ejes principales de la máquina como ejes eléctricos y no neumáticos, se ha comprobado que la máquina es más rápida, más precisa, más segura y que se dispone de un mayor control de esta que con la máquina predecesora. Los ejes eléctricos (con todos sus componentes) cumplen con las necesidades de diseño de la máquina, ya que cumplen con el espacio logístico que se requería y se ha reducido el consumo eléctrico con el cambio; también se ha cumplido con los requisitos de programación, ya que las comunicaciones del PLC cumplen con los estándares internos de la empresa (programación en el entorno Codesys y red de comunicación por Ethercat) y con el requisito externo de realizar un control de la producción de cada operario; por último, cumplimos con el requisito económico, ya que el precio de la máquina sería entorno a los 8000 €, menos de lo que se puso como objetivo.

11. BIBLIOGRAFÍA

- Catálogo y manuales de los motores y ejes eléctricos de FESTO.
- Catálogo y manuales del variador de frecuencia de CONTROL TECHNIQUES.
- Apuntes Curso Calculo de motores eléctricos empresa Intra Automation Solutions SL.
- Portal de soporte y documentación técnica de FESTO.
- Apuntes de Automatización Industrial UEM.
- [1] TFG – Retrofit de sistema de automatización de una máquina de encajado industrial (Elaboración propia).

12. ANEXO

12.1. ORGANIZACIÓN PLC

La programación se ha estructurado de la siguiente manera:

- **Listas de variables globales:** son las listas que componen todas las variables globales que se han usado en el proyecto. Se incluyen las variables globales, las variables globales del HMI y las variables persistentes (variables globales que no son temporales, aunque se apague el PLC el valor de las variables se queda guardado en el PLC).



Ilustración 24. Variables Globales

- **Bloques de funciones:** en esta programación, se incluye la única función que se ha usado en el programa, que gestiona el funcionamiento del variador de frecuencia.



Ilustración 25. Funciones

- **Programación:** se encuentra el programa principal y los subprogramas que lo componen.

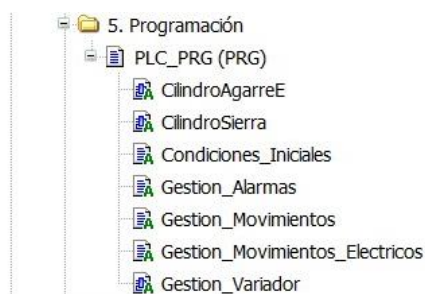


Ilustración 26. Programación

- Visualizaciones: la visualización del HMI se ha realizado mediante marcos. Existe una pantalla principal compuesta por diferentes marcos, estos marcos van cambiando en vez de cambiar toda la pantalla, esto agiliza la programación y hace el HMI más esquemático. En la parte de visualización se encuentra además dos programas para la gestión del HMI, el archivo donde se encuentran las imágenes y por último la gestión del WebVisu.

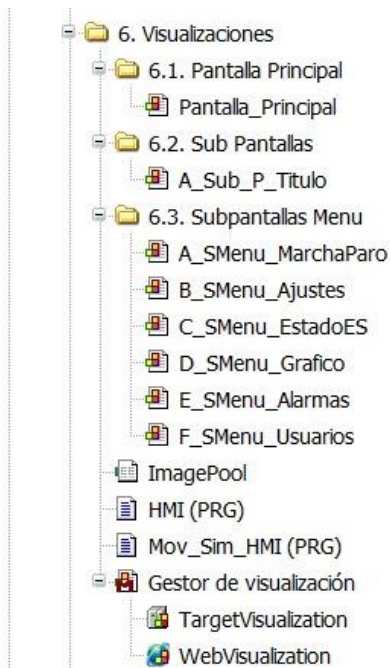


Ilustración 27. Gestión HMI

- Alarmas: se encuentra toda la gestión de las alarmas del programa.

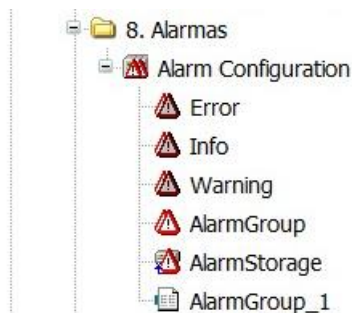


Ilustración 28. Gestión alarmas

- Configuración de tareas.

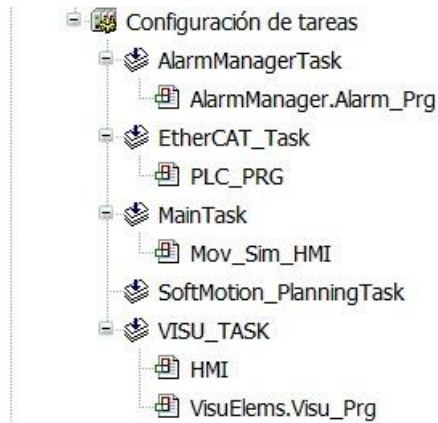


Ilustración 29. Gestión de tareas

- Comunicaciones: la comunicación de todos los componentes electrónicos de la máquina se ha realizado a través de Ethercat.

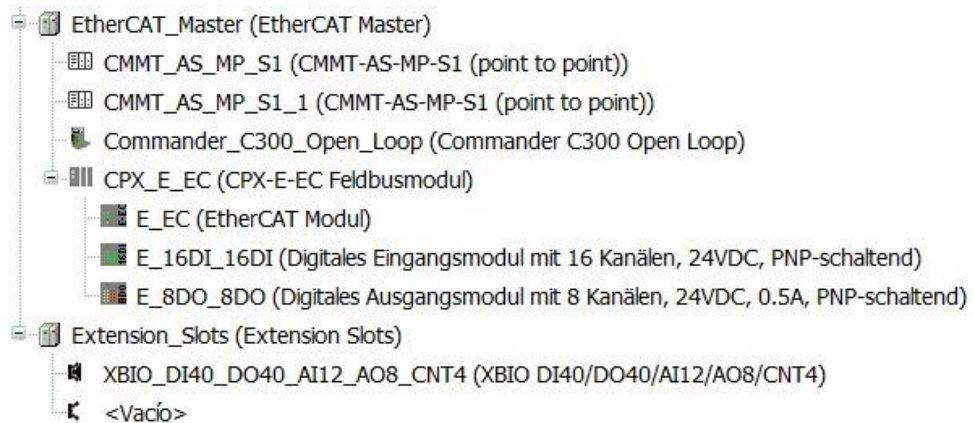


Ilustración 30. Comunicaciones del PLC

12.2. LISTAS DE VARIABLES GLOBALES

12.2.1. GVL

```
1  VAR_GLOBAL
2  //Pulsadores Físicos:
3  Marcha_Fis_1 : BOOL ;
4  Marcha_Fis_2 : BOOL ;
5  Selector_Discriminacion_Corte : BOOL ;
6  Seta : BOOL ;
7  Boton_Cadena : BOOL ;
8  SelectorExtremoJamonOk : BOOL ;
9  Enable_Fis : BOOL ;
10
11 // Sensores:
12 Sensor_Puerta : BOOL ;
13 Sensor_CilindroN_Pata_Fin : BOOL ;
14 Sensor_Boton_Despresurizacion : BOOL ;
15 PresionAire : DINT ;
16
17 // Generales:
18 FinCorte : BOOL ;
19 Corte : BOOL ;
20 Marcha_OK : BOOL ;
21 NoRepetir : BOOL ;
22 CondicionesInicialesOK : BOOL ;
23 NoRepetir_CondicionesIniciales : BOOL ;
24 Marca_Stop : BOOL ;
25 Marca_STO : BOOL ;
26 ValvulaCorte : BOOL ;
27 PresionInterpolada : REAL ;
28 Temporizador_CilindroN_Pata : TON ;
29 Existe_Alarma : BOOL ;
30 Marca_Seta : BOOL ;
31 Marca_Puerta : BOOL ;
32 Error_Presion : BOOL ;
33 Error_Comunicaciones : BOOL ;
34 Temporizador_Error_Ethercat : TON ;
35 Info_Falta_Jamon : BOOL ;
36 Marca_Despresurizacion : BOOL ;
37 CondicionesIniciales_Run : BOOL ;
38 CilindroAgarreE : INT ;
39
40 // Cilindros:
41 CilindroN_Pata : BOOL ;
42 CilindroN_Agarre : BOOL ;
43 CilindroN_Elevacion_A : BOOL ;
44 CilindroN_Elevacion_B : BOOL ;
45 Pinchos : BOOL ;
46 Marca_CilindroSierra_Fin : BOOL ;
47 Marca_CilindroAgarreE_Fin : BOOL ;
48 Marca_CilindroSierra : BOOL ;
49 Motor_Sierra_Marcha : BOOL ;
50 Marca_CilindroAgarreE : BOOL ;
51
52 // Gestión Cilindros:
53 MC_Power_Festo_0 : MC_Power_Festo ;
54 MC_TorqueControl_Festo_0 : MC_TorqueControl_Festo ;
55 MC_ReadActualPosition_Festo_0 : MC_ReadActualPosition_Festo ;
56 MC_ReadActualTorque_Festo_0 : MC_ReadActualTorque_Festo ;
57 MC_Home_Festo_0 : MC_Home_Festo ;
58 MC_Reset_Festo_0 : MC_Reset_Festo ;
59 MC_MoveAbsolute_Festo_0 : MC_MoveAbsolute_Festo ;
60 MC_Stop_Festo_0 : MC_Stop_Festo ;
61 MC_Power_Festo_1 : MC_Power_Festo ;
62 MC_TorqueControl_Festo_1 : MC_TorqueControl_Festo ;
63 MC_Home_Festo_1 : MC_Home_Festo ;
```

```
64      MC_MoveAbsolute_Festo_1 : MC_MoveAbsolute_Festo ;
65      MC_ReadActualPosition_Festo_1 : MC_ReadActualPosition_Festo ;
66      MC_ReadActualTorque_Festo_1 : MC_ReadActualTorque_Festo ;
67      MC_Reset_Festo_1 : MC_Reset_Festo ;
68      MC_Stop_Festo_1 : MC_Stop_Festo ;
69
70      // Índices:
71      i : INT ;
72      j : INT ;
73      k : INT ;
74      n : INT ;
75      p : INT ;
76
77      // MarcasAjustes:
78      // CilindroSierra:
79      MarcaPosicionLimite_CilindroSierra : REAL ;
80      MarcaTorqueLimite_CilindroSierra : LREAL ;
81      MarcaPosicionVuelta_CilindroSierra : LREAL ;
82      MarcaVelocidadVuelta_CilindroSierra : LREAL ;
83      MarcaAceleracionVuelta_CilindroSierra : LREAL ;
84      MarcaDeceleracionVuelta_CilindroSierra : LREAL ;
85      MarcaJerkVuelta_CilindroSierra : LREAL ;
86
87      // CilindroAgarre:
88      MarcaPosicionLimite_CilindroAgarreE : REAL ;
89      MarcaTorqueLimite_CilindroAgarreE : LREAL ;
90      MarcaPosicionVuelta_CilindroAgarreE : LREAL ;
91      MarcaVelocidadVuelta_CilindroAgarreE : LREAL ;
92      MarcaAceleracionVuelta_CilindroAgarreE : LREAL ;
93      MarcaDeceleracionVuelta_CilindroAgarreE : LREAL ;
94      MarcaJerkVuelta_CilindroAgarreE : LREAL ;
95
96      // Power:
97      Power_Enable : BOOL ;
98      Power_Status_Sierra : BOOL ;
99      Power_Error_Sierra : BOOL ;
100
101      Power_Status_Agarre : BOOL ;
102      Power_Error_Agarre : BOOL ;
103
104      // Torque:
105      Torque_Execute_Sierra : BOOL ;
106      Torque_Cilindro_Sierra : REAL ;
107      Torque_Rampa_Sierra : REAL ;
108      Torque_Velocidad_Sierra : REAL ;
109      Torque_Error_Sierra : BOOL ;
110      ReadTorque_Actual_Sierra : REAL ;
111      ReadTorque_Error_Sierra : BOOL ;
112      Torque_Execute_Agarre : BOOL ;
113      Torque_CilindroTorque_Agarre : REAL ;
114      Torque_Rampa_Agarre : REAL ;
115      Torque_Velocidad_Agarre : REAL ;
116      Torque_Error_Agarre : BOOL ;
117      ReadTorque_Actual_Agarre : REAL ;
118      ReadTorque_Error_Agarre : BOOL ;
119
120      // Home:
121      Home_Execute : BOOL ;
122      Home_Position : REAL ;
123      Home_Done_Sierra : BOOL ;
124      Home_Error_Sierra : BOOL ;
125      Home_Done_Agarre : BOOL ;
126      Home_Error_Agarre : BOOL ;
127      Home_Execute_Sierra : BOOL ;
128      Home_Execute_Agarre : BOOL ;
129
```

```
130 // Reset:
131 Reset_Execute : BOOL ;
132 Reset_Done_Sierra : BOOL ;
133 Reset_Error_Sierra : BOOL ;
134
135 Reset_Done_Agarre : BOOL ;
136 Reset_Error_Agarre : BOOL ;
137
138 // Stop:
139 Stop_Execute_Sierra : BOOL ;
140 Stop_Done_Sierra : BOOL ;
141 Stop_Error_Sierra : BOOL ;
142 Stop_Done_Agarre : BOOL ;
143 Stop_Error_Agarre : BOOL ;
144 Stop_Execute_Agarre : BOOL ;
145
146
147 // MoveAbsolute:
148 MoveAbsolute_Execute_Sierra : BOOL ;
149 MoveAbsolute_Posicion_Sierra : REAL ;
150 MoveAbsolute_Velocidad_Sierra : REAL ;
151 MoveAbsolute_Aceleracion_Sierra : REAL ;
152 MoveAbsolute_Deceleracion_Sierra : REAL ;
153 MoveAbsolute_Jerk_Sierra : REAL ;
154 MoveAbsolute_Done_Sierra : BOOL ;
155
156 MoveAbsolute_Error_Sierra : BOOL ;
157 MoveAbsolute_Execute_Agarre : BOOL ;
158 MoveAbsolute_Posicion_Agarre : REAL ;
159 MoveAbsolute_Velocidad_Agarre : REAL ;
160 MoveAbsolute_Aceleracion_Agarre : REAL ;
161 MoveAbsolute_Deceleracion_Agarre : REAL ;
162 MoveAbsolute_Jerk_Agarre : REAL ;
163 MoveAbsolute_Done_Agarre : BOOL ;
164 MoveAbsolute_Direccion_Sierra : Festo_PtP_Base . MC_DIRECTION ;
165 MoveAbsolute_Direccion_Agarre : Festo_PtP_Base . MC_DIRECTION ;
166
167 // Reads:
168 ReadPos_Actual_Sierra : REAL ;
169 ReadPos_Error_Sierra : BOOL ;
170 ReadPos_Actual_Agarre : REAL ;
171 ReadPos_Error_Agarre : BOOL ;
172
173 // Variador:
174 Control_Word_Variador : UINT ;
175 Velocidad_Variador_Giro : INT ;
176 Status_Word_Variador : UINT ;
177 Read_Velocidad_Variador_Giro : INT ;
178
179 END_VAR
180
```

12.2.2. GVL_HMI

```
1  VAR_GLOBAL
2  // Identificador de pantallas principales:
3  giFramePrincipal : INT ;
4
5  // Botones HMI:
6  Enable_HMI : BOOL ;
7  Stop_HMI : BOOL ;
8  Reset_HMI : BOOL ;
9  BotonMenu_HMI : BOOL ;
10 Homing_HMI : BOOL ;
11 Reset_Contador : BOOL ;
12 AjustesPredeterminados : BOOL ;
13 PosicionSierra_HMI : REAL ;
14 PosicionAgarre_HMI : REAL ;
15 PosicionPata_HMI : INT ;
16 PosicionAgarreN_HMI : INT ;
17 Velocidad_Variador : INT ;
18 Subir_Maquina_HMI : BOOL ;
19 MensajeIntroducirOperario : BOOL ;
20 Boton_SalirOperario : BOOL ;
21
22 // Recetas:
23 Recetas_On : BOOL ;
24 MensajeRecetasOff : BOOL ;
25
26 // Generales:
27 MovimientoMenu : INT ;
28 NoReturn_BM : BOOL ;
29 Inicio_NoReturn_BM : BOOL ;
30 Marcha_BM : BOOL ;
31 VueltaMovMenu : BOOL ;
32 ColorMenu : BOOL ;
33
34 // Índices:
35 l : INT ;
36 m : INT ;
37 w : INT ;
38
39 // Temporizadores:
40 TemporizadorMovMenu : TON ;
41 TemporizadorMenuCerrar : TON ;
42 TemporizadorAjPred : TON ;
43 Temporizador_Led_Alarma : TON ;
44 Temporizador_Operario : TON ;
45
46 // Gestión potenciómetro HMI:
47 Potenciómetro_Mas_HMI : BOOL ;
48 Potenciómetro_Menos_HMI : BOOL ;
49 Potenciómetro_Torque_CilindroAgarreE_HMI : REAL ;
50 Regla3_Torque_CilindroAgrreE_HMI : REAL ;
51 // Flancos:
52 Marca_1 : INT ;
53 Marca_2 : INT ;
54
55 // Simulaciones:
56 Giro_HMI : INT ;
57
58 // Gestión contador jamones manual:
59 Jamon_Mas_HMI : BOOL ;
60 Jamon_Menos_HMI : BOOL ;
61 // Flancos:
62 Marca_3 : INT ;
63 Marca_4 : INT ;
64
65 END_VAR
66
```

12.2.3. PERSISTENTVARS

```
1  VAR_GLOBAL PERSISTENT RETAIN
2      // Contadores:
3      ContadorJamones : INT ;
4
5      // Ajustes HMI cilindros eléctricos:
6      // CilindroSierra:
7      HMI_Torque_CilindroSierra : REAL ;
8      HMI_TorqueRampa_CilindroSierra : REAL ;
9      HMI_TorqueVelocidad_CilindroSierra : REAL ;
10     HMI_PosicionLimite_CilindroSierra : REAL ;
11     HMI_TorqueLimite_CilindroSierra : LREAL ;
12     HMI_PosicionVuelta_CilindroSierra : LREAL ;
13     HMI_VelocidadVuelta_CilindroSierra : LREAL ;
14     HMI_AceleracionVuelta_CilindroSierra : LREAL ;
15     HMI_Deceleracion_CilindroSierra : LREAL ;
16     HMI_JerkVuelta_CilindroSierra : LREAL ;
17     HMI_Velocidad_Variador : INT ;
18
19     // CilindroAgarreE:
20     HMI_TorqueRampa_CilindroAgarreE : REAL ;
21     HMI_TorqueVelocidad_CilindroAgarreE : REAL ;
22     HMI_PosicionLimite_CilindroAgarreE : REAL ;
23     HMI_TorqueLimite_CilindroAgarreE : LREAL ;
24     HMI_PosicionVuelta_CilindroAgarreE : LREAL ;
25     HMI_VelocidadVuelta_CilindroAgarreE : LREAL ;
26     HMI_AceleracionVuelta_CilindroAgarreE : LREAL ;
27     HMI_Deceleracion_CilindroAgarreE : LREAL ;
28     HMI_JerkVuelta_CilindroAgarreE : LREAL ;
29
30     Potenciometro_HMI : INT ;
31     Bajar_Maquina_HMI : BOOL ;
32     nombreOperario : INT ;
33     TiempoDescansoOperario : INT ;
34 END_VAR
35
```

12.3. FUNCIONES

```
1  FUNCTION_BLOCK fb_Control_Variador
2  VAR_INPUT
3      Marcha_Variador : BOOL ;
4      Reset_Variador : BOOL ;
5      Velocidad_Actual_Variador : INT ;
6      Condiciones_Iniciales_OK : BOOL ;
7      Status_Word_Variador : UINT ;
8  END_VAR
9  VAR_OUTPUT
10     Velocidad_Variador_fb : INT ;
11     Control_Word_Variador : UINT ;
12 END_VAR
13 VAR
14     y : INT ;
15     z : INT ;
16 END_VAR
17
```

```
1  IF Condiciones_Iniciales_OK THEN
2      z := 0 ;
3      IF Marcha_Variador THEN
4          y := 0 ;
5          CASE w OF
6              0 :
7                  IF Status_Word_Variador = 563 THEN
8                      Velocidad_Variador_fb := Velocidad_Variador ;
9                      w := w + 10 ;
10                 END_IF
11
12             10 :
13                 Control_Word_Variador := 31 ;
14                 w := w + 10 ;
15             END_CASE
16         ELSE
17             w := 0 ;
18             CASE y OF
19                 0 :
20                     Velocidad_Variador_fb := 0 ;
21                     IF Velocidad_Actual_Variador = 0 THEN
22                         Control_Word_Variador := 6 ;
23                         y := y + 10 ;
24                     END_IF
25
26                 10 :
27                     IF Status_Word_Variador = 561 THEN
28                         Control_Word_Variador := 7 ;
29                         y := y + 10 ;
30                     END_IF
31             END_CASE
32         END_IF

```

```
33     ELSE
34         CASE z OF
35             0 :
36                 Control_Word_Variador := 6 ;
37                 z := z + 10 ;
38
39             10 :
40                 IF Status_Word_Variador = 561 THEN
41                     z := z + 10 ;
42                 END_IF
43
44             20 :
45                 Control_Word_Variador := 7 ;
46                 z := z + 10 ;
47
48             30 :
49                 IF Status_Word_Variador = 563 THEN
50                     z := z + 10 ;
51                 END_IF
52         END_CASE
53     END_IF
54
```

12.4. PROGRAMA PRINCIPAL (PLC_PRG) Y SUBPROGRAMAS

El programa principal PLC_PRG incluye las variables locales, las llamadas a los subprogramas, a las funciones, los temporizadores, el marcha/paro, la seta de emergencia o el sensor de apertura de la puerta, los valores de los ajustes predeterminados y las conversiones.

```
1 PROGRAM PLC_PRG
2 VAR
3     PruebaCilindroN_Pata_Fin : BOOL ;
4     Temporizador_Marca_Sierra : TON ;
5     Temporizador_Marca_Agarre : TON ;
6     PruebaSelectorExtremoJamonOK : BOOL ;
7     PruebaBotonCadena : BOOL ;
8     PruebaSensorBotonDespresurizado : BOOL ;
9     fb_Control_Variador_0 : fb_Control_Variador ;
10    Presion : DINT ;
11    PresionAireReal : REAL ;
12    PresionR : REAL ;
13    Temporizador_Prueba : TON ;
14    Temporizador_Prueba_2 : TON ;
15    Temporizador_Corte : TON ;
16    o : INT ;
17    Home_Realizado_Despresurizacion_Sierra : BOOL ;
18    Home_Realizado_Despresurizacion_AgarreE : BOOL ;
19 END_VAR
20
```

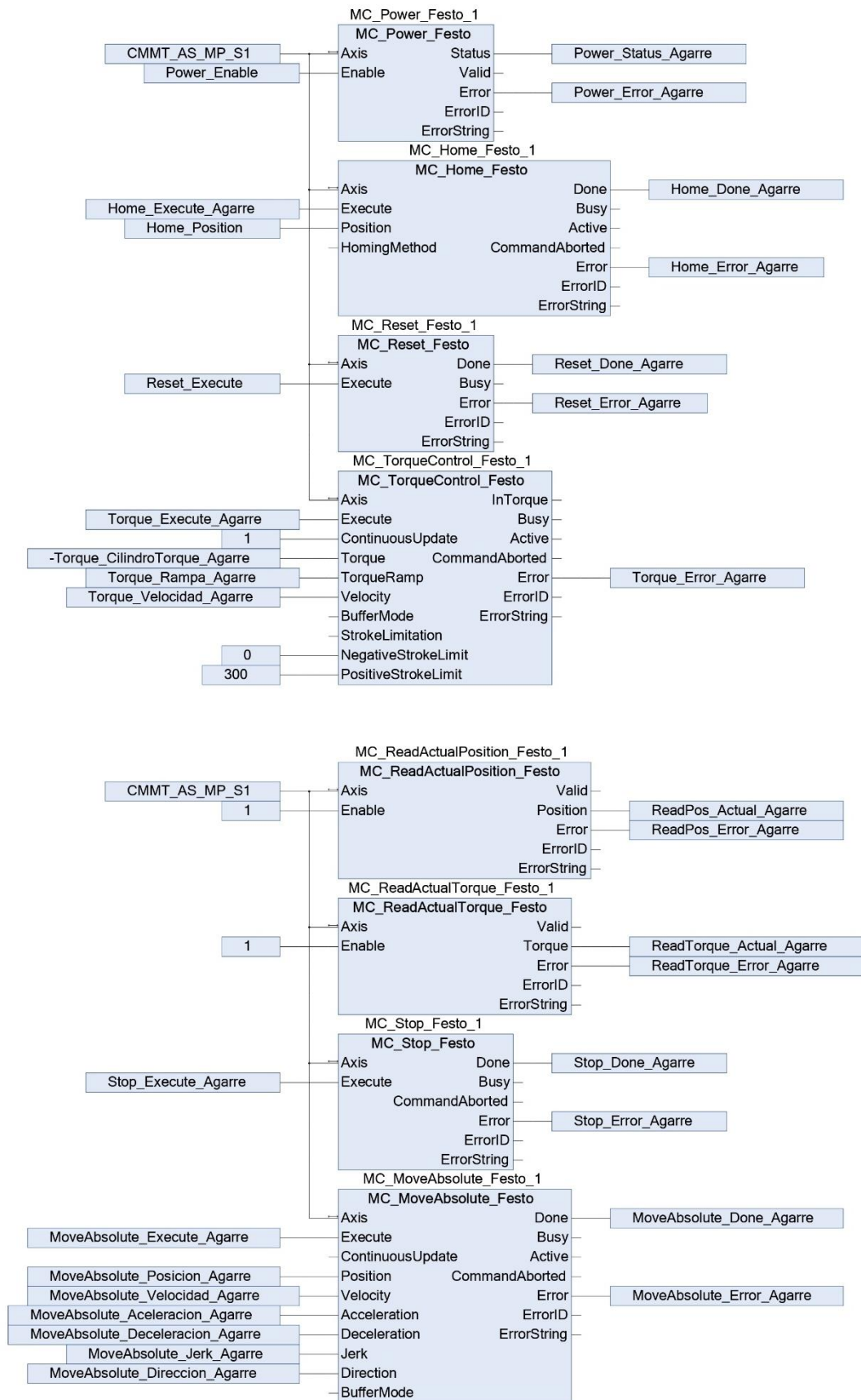
```
1 (* ***** *)
2 (* ***** Programa Principal ***** *)
3 (* ***** *)
4
5 // Para usar el webvisu:
6 // http://192.168.1.2:8080/webvisu.htm
7
8 // Llamada a los subprogramas:
9 Condiciones_Iniciales ();
10 Gestion_Movimientos ();
11 Gestion_Movimientos_Electricos ();
12 CilindroAgarreE ();
13 CilindroSierra ();
14 Gestion_Variador ();
15 Gestion_Alarmas ();
16
17 // Llamada a las funciones:
18 fb_Control_Variador_0 ();
19
20 // Temporizadores:
21 Temporizador_Marca_Sierra ();
22 Temporizador_Marca_Agarre ();
23 Temporizador_Prueba ();
24 Temporizador_Prueba_2 ();
25 Temporizador_Corte ();
26 Temporizador_CilindroN_Pata ();
27 Temporizador_Led_Alarma ();
28 Temporizador_Error_Ethercat ();
29
30 // Marcha/Paro:
31 IF CondicionesInicialesOK THEN
32     IF (Marcha_Fis_1 AND Marcha_Fis_2) AND NOT Marca_STO AND NOT MensajeIntroducirOperario THEN
33         Marcha_OK := TRUE ;
34         Reset_Execute := FALSE ;
35         Temporizador_Operario.IN := FALSE ;
36         Home_Realizado_Despresurizacion_Sierra := FALSE ;
37         Home_Realizado_Despresurizacion_AgarreE := FALSE ;
38     END_IF
39 END_IF
40
```

```
41     IF NOT Sensor_Boton_Despresurizacion OR Stop_HMI THEN
42     Marcha_OK := FALSE ;
43     NoRepetir := FALSE ;
44     FinCorte := FALSE ;
45     Pinchos := FALSE ;
46     CilindroN_Pata := FALSE ;
47     Enable_HMI := FALSE ;
48     Marca_CilindroAgarreE := FALSE ;
49     Marca_CilindroSierra := FALSE ;
50     Marca_CilindroAgarreE_Fin := FALSE ;
51     Marca_CilindroSierra_Fin := FALSE ;
52     Motor_Sierra_Marcha := FALSE ;
53     CilindroN_Agarre := FALSE ;
54     Torque_Execute_Sierra := FALSE ;
55     Torque_Execute_Agarre := FALSE ;
56     Reset_Execute := FALSE ;
57     Temporizador_Corte.IN := FALSE ;
58     Temporizador_Marca_Sierra.IN := FALSE ;
59     Stop_Execute_Agarre := FALSE ;
60     Stop_Execute_Sierra := FALSE ;
61     i := 0 ;
62     j := 0 ;
63     k := 0 ;
64     n := 0 ;
65     Marca_Despresurizacion := TRUE ;
66     IF EtherCAT_Master.xDistributedClockInSync AND EtherCAT_Master.xConfigFinished THEN
67     END_IF
68 END_IF
69
70
71 IF Marca_Despresurizacion THEN
72     IF NOT (ReadPos_Actual_Sierra >= HMI_PosicionVuelta_CilindroSierra - 10 AND
73     ReadPos_Actual_Sierra <= HMI_PosicionVuelta_CilindroSierra + 10) AND NOT
74     Home_Realizado_Despresurizacion_Sierra THEN
75     MoveAbsolute_Execute_Sierra := TRUE ;
76     Home_Realizado_Despresurizacion_Sierra := TRUE ;
77     END_IF
78
79     IF MoveAbsolute_Done_Sierra THEN
80     MoveAbsolute_Execute_Sierra := FALSE ;
81     Home_Realizado_Despresurizacion_Sierra := FALSE ;
82     END_IF
83
84     IF NOT (ReadPos_Actual_Agarre >= HMI_PosicionVuelta_CilindroAgarreE - 10 AND
85     ReadPos_Actual_Agarre <= HMI_PosicionVuelta_CilindroAgarreE + 10) AND NOT
86     Home_Realizado_Despresurizacion_AgarreE THEN
87     MoveAbsolute_Execute_Agarre := TRUE ;
88     Home_Realizado_Despresurizacion_AgarreE := TRUE ;
89     END_IF
90
91     IF MoveAbsolute_Done_Agarre THEN
92     MoveAbsolute_Execute_Agarre := FALSE ;
93     Home_Realizado_Despresurizacion_AgarreE := FALSE ;
94     END_IF
95     IF NOT MoveAbsolute_Execute_Sierra AND NOT MoveAbsolute_Execute_Agarre THEN
96     Stop_HMI := FALSE ;
97     Marca_Despresurizacion := FALSE ;
98     END_IF
99 END_IF
100
101 IF NOT Seta OR NOT Sensor_Puerta THEN
102     ValvulaCorte := FALSE ;
103     Pinchos := FALSE ;
104     Motor_Sierra_Marcha := FALSE ;
105     CilindroN_Agarre := FALSE ;
106     CilindroN_Pata := FALSE ;
107     Torque_Execute_Sierra := FALSE ;
108     MoveAbsolute_Execute_Sierra := FALSE ;
109     Torque_Execute_Agarre := FALSE ;
110     MoveAbsolute_Execute_Agarre := FALSE ;
111     Home_Execute := FALSE ;
112     CondicionesInicialesOK := FALSE ;
113     NoRepetir_CondicionesIniciales := FALSE ;
114     Marca_STO := TRUE ;
115     Power_Enable := FALSE ;
116     Marcha_OK := FALSE ;
117     Marca_CilindroSierra := FALSE ;
118     Marca_CilindroAgarreE := FALSE ;
119     Marca_CilindroAgarreE_Fin := FALSE ;
120     Marca_CilindroSierra_Fin := FALSE ;
121     NoRepetir := FALSE ;
122     Reset_Execute := FALSE ;
123     Temporizador_Corte.IN := FALSE ;
124     Temporizador_Marca_Sierra.IN := FALSE ;
```

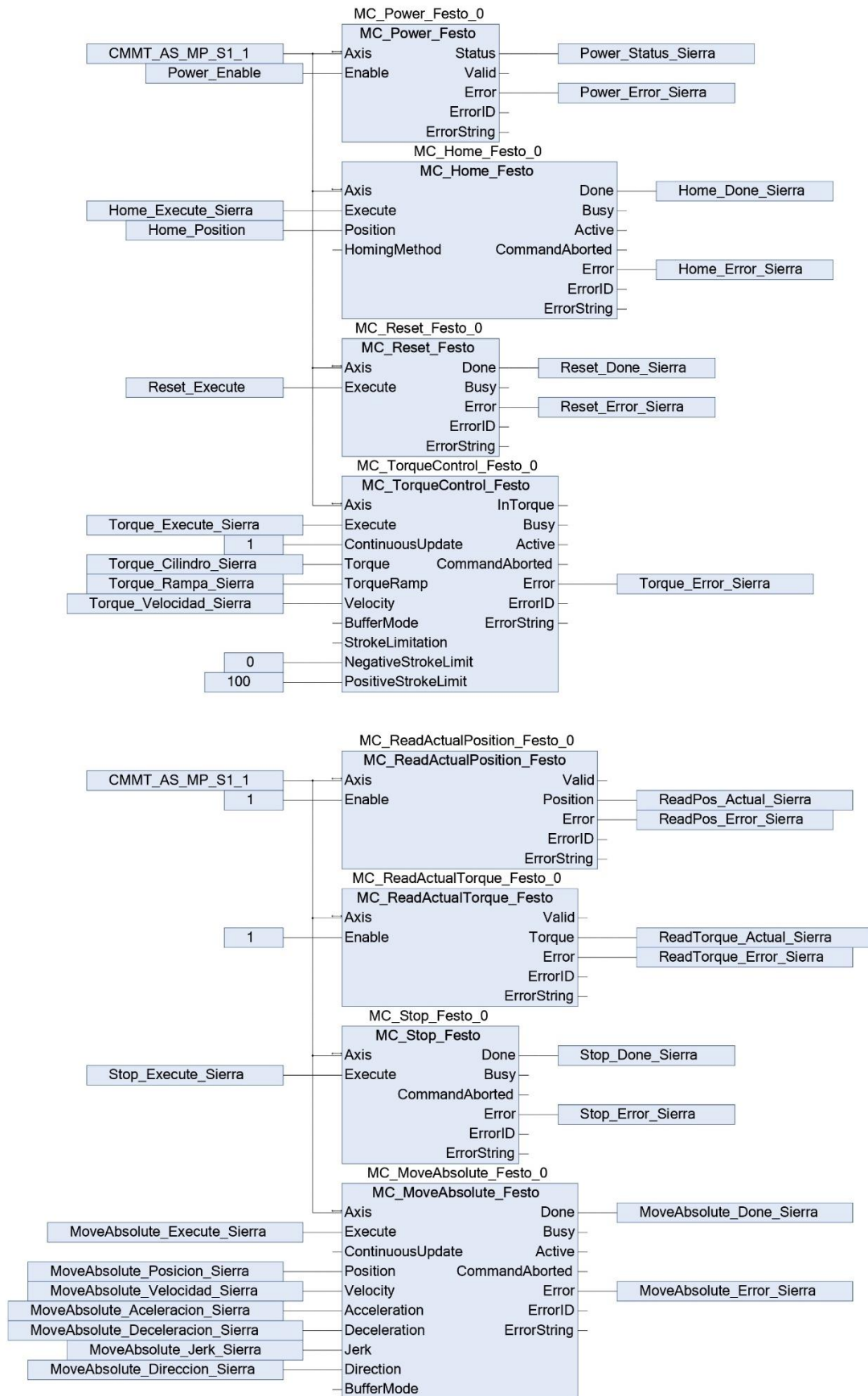
```
125         Stop_Execute_Agarre := FALSE ;
126         Home_Execute_Sierra := FALSE ;
127         Home_Execute_Agarre := FALSE ;
128         Enable_HMI := FALSE ;
129         i := 0 ;
130         j := 0 ;
131         k := 0 ;
132         n := 0 ;
133         IF NOT Seta THEN
134             Marca_Seta := TRUE ;
135         ELSIF NOT Sensor_Puerta THEN
136             Marca_Puerta := TRUE ;
137         END_IF
138
139         IF CondicionesInicialesOK THEN
140             Stop_Execute_Sierra := TRUE ;
141             Stop_Execute_Agarre := TRUE ;
142         END_IF
143     ELSE
144         Marca_STO := FALSE ;
145         Marca_Seta := FALSE ;
146         Marca_Puerta := FALSE ;
147     END_IF
148
149     IF Reset_HMI AND NOT Marca_STO THEN
150         Reset_Execute := TRUE ;
151         Reset_HMI := FALSE ;
152     END_IF
153
154     // Ajustes cilindros por HMI:
155     IF AjustesPredeterminados THEN // Hay que configurar los ajustes predeterminados reales,
156                                     // estos son números de otro programa (TensorTricomposites).
157
158                                     // Además, hay que poner los máximos y mínimos en el HMI.
159
160         // CilindroSierra:
161         Torque_Cilindro_Sierra := 2.62 ; //1.24;
162         Torque_Rampa_Sierra := 1 ;
163         Torque_Velocidad_Sierra := 3 ;
164         MarcaPosicionLimite_CilindroSierra := 89 ;
165         MarcaTorqueLimite_CilindroSierra := 0.2 ;
166         MarcaPosicionVuelta_CilindroSierra := 0 ;
167         MarcaVelocidadVuelta_CilindroSierra := 100 ;
168         MarcaAceleracionVuelta_CilindroSierra := 1000 ;
169         MarcaDeceleracionVuelta_CilindroSierra := 1000 ;
170         MarcaJerkVuelta_CilindroSierra := 1000 ;
171
172         // CilindroAgarreE:
173         Torque_CilindroTorque_Agarre := Potenciometro_Torque_CilindroAgarreE_HMI ; // Según potenciometro.
174         Torque_Rampa_Agarre := 1 ;
175         Torque_Velocidad_Agarre := 3 ;
176         MarcaPosicionLimite_CilindroAgarreE := 0 ;
177         MarcaTorqueLimite_CilindroAgarreE := 0.2 ;
178         MarcaPosicionVuelta_CilindroAgarreE := 290 ;
179         MarcaVelocidadVuelta_CilindroAgarreE := 100 ;
180         MarcaAceleracionVuelta_CilindroAgarreE := 1000 ;
181         MarcaDeceleracionVuelta_CilindroAgarreE := 1000 ;
182         MarcaJerkVuelta_CilindroAgarreE := 1000 ;
183         Velocidad_Variador := 1300 ;
184
185     ELSE
186
187         // CilindroSierra:
188         Torque_Cilindro_Sierra := HMI_Torque_CilindroSierra ;
189         Torque_Rampa_Sierra := HMI_TorqueRampa_CilindroSierra ;
190         Torque_Velocidad_Sierra := HMI_TorqueVelocidad_CilindroSierra ;
191         MarcaPosicionLimite_CilindroSierra := HMI_PosicionLimite_CilindroSierra ;
192         MarcaTorqueLimite_CilindroSierra := HMI_TorqueLimite_CilindroSierra ;
193         MarcaPosicionVuelta_CilindroSierra := HMI_PosicionVuelta_CilindroSierra ;
194         MarcaVelocidadVuelta_CilindroSierra := HMI_VelocidadVuelta_CilindroSierra ;
195         MarcaAceleracionVuelta_CilindroSierra := HMI_AceleracionVuelta_CilindroSierra ;
196         MarcaDeceleracionVuelta_CilindroSierra := MarcaAceleracionVuelta_CilindroSierra ;
197         MarcaJerkVuelta_CilindroSierra := 1000 ;
198
199         // CilindroAgarreE:
200         Torque_CilindroTorque_Agarre := Potenciometro_Torque_CilindroAgarreE_HMI ; // Según potenciometro.
201         Torque_Rampa_Agarre := HMI_TorqueRampa_CilindroAgarreE ;
202         Torque_Velocidad_Agarre := HMI_TorqueVelocidad_CilindroAgarreE ;
203         MarcaPosicionLimite_CilindroAgarreE := - HMI_PosicionLimite_CilindroAgarreE ;
204         MarcaTorqueLimite_CilindroAgarreE := HMI_TorqueLimite_CilindroAgarreE ;
205         MarcaPosicionVuelta_CilindroAgarreE := HMI_PosicionVuelta_CilindroAgarreE ;
206         MarcaVelocidadVuelta_CilindroAgarreE := HMI_VelocidadVuelta_CilindroAgarreE ;
207         MarcaAceleracionVuelta_CilindroAgarreE := HMI_AceleracionVuelta_CilindroAgarreE ;
208         MarcaDeceleracionVuelta_CilindroAgarreE := MarcaAceleracionVuelta_CilindroAgarreE ;
209         MarcaJerkVuelta_CilindroAgarreE := 1000 ; // 100.
210         Velocidad_Variador := HMI_Velocidad_Variador ;
211     END_IF
212
213 END
```

```
209 // Conversiones:
210 // CilindroSierra:
211 MoveAbsolute_Posicion_Sierra := LREAL_TO_REAL (MarcaPosicionVuelta_CilindroSierra);
212 MoveAbsolute_Velocidad_Sierra := LREAL_TO_REAL (MarcaVelocidadVuelta_CilindroSierra);
213 MoveAbsolute_Aceleracion_Sierra := LREAL_TO_REAL (MarcaAceleracionVuelta_CilindroSierra);
214 MoveAbsolute_Deceleracion_Sierra := LREAL_TO_REAL (MarcaDeceleracionVuelta_CilindroSierra);
215 MoveAbsolute_Jerk_Sierra := LREAL_TO_REAL (MarcaJerkVuelta_CilindroSierra);
216 // CilindroAgarreE:
217 MoveAbsolute_Posicion_Agarre := - LREAL_TO_REAL (MarcaPosicionVuelta_CilindroAgarreE);
218 MoveAbsolute_Velocidad_Agarre := LREAL_TO_REAL (MarcaVelocidadVuelta_CilindroAgarreE);
219 MoveAbsolute_Aceleracion_Agarre := LREAL_TO_REAL (MarcaAceleracionVuelta_CilindroAgarreE);
220 MoveAbsolute_Deceleracion_Agarre := LREAL_TO_REAL (MarcaDeceleracionVuelta_CilindroAgarreE);
221 MoveAbsolute_Jerk_Agarre := LREAL_TO_REAL (MarcaJerkVuelta_CilindroAgarreE);
222
223 // Interpolacion presion de aire (señal analógica):
224 Presion := PresionAire;
225 PresionR := PresionAireReal / 1000;
226 PresionInterpolada := 10 + ((0 - 10) * (PresionR - 10)) / (2 - 10);
227
228
```

12.4.1. CILINDROAGARRE_E



12.4.2. CILINDROSIERRA



12.4.3. CONDICIONES_INICIALES

```
1  (* ***** *)
2  (* ***** Condiciones Iniciales ***** *)
3  (* ***** *)
4
5  IF ( (Marcha_Fis_1 AND Marcha_Fis_2 ) OR Enable_HMI OR Enable_Fis AND NOT NoRepetir_CondicionesIniciales ) THEN
6  NoRepetir_CondicionesIniciales := TRUE;
7  j := 0;
8  END_IF
9
10 IF Homing_HMI THEN
11 CondicionesInicialesOK := FALSE;
12 j := 0;
13 END_IF
14
15 IF (NoRepetir_CondicionesIniciales AND NOT CondicionesInicialesOK ) THEN
16 CASE j OF
17 0 :
18 ValvulaCorte := TRUE;
19 CilindroN_Agarre := FALSE;
20 CilindroN_Pata := FALSE;
21 Motor_Sierra_Marcha := FALSE;
22 Reset_Execute := TRUE;
23 CondicionesIniciales_Run := TRUE;
24 j := j + 5;
25
26 5 :
27 IF Reset_Done_Agarre AND Reset_Done_Sierra THEN
28 Reset_Execute := FALSE;
29 j := j + 5;
30 END_IF
31
32 10 : // Poner en Enable los cilindros eléctricos.
33 IF NOT Power_Status_Sierra THEN
34 Power_Enable := TRUE;
35 END_IF
36
37 IF NOT Power_Status_Agarre THEN
38 Power_Enable := TRUE;
39 END_IF
40
41 IF Power_Status_Sierra AND Power_Status_Agarre THEN
42 j := j + 10;
43 END_IF
44
45 20 : // Realizar el Homming de los cilindros eléctricos.
46 IF NOT Home_Done_Sierra THEN
47 Home_Execute_Sierra := TRUE;
48 Home_Position := 0;
49 END_IF
50
51 IF NOT Home_Done_Agarre THEN
52 Home_Execute_Agarre := TRUE;
53 Home_Position := 0;
54 END_IF
55
56 IF Home_Done_Sierra AND Home_Done_Agarre THEN
57 Home_Execute_Sierra := FALSE;
58 Home_Execute_Agarre := FALSE;
59 Enable_HMI := FALSE;
60 j := j + 10;
61 END_IF
62
63 30 :
64 MoveAbsolute_Execute_Agarre := TRUE;
65 IF ReadPos_Actual_Agarre >= MoveAbsolute_Posicion_Agarre THEN
66 MoveAbsolute_Execute_Agarre := FALSE;
67 j := j + 10;
68 END_IF
69
70 40 : // Si todo está Ok las condiciones iniciales se consideran correctas.
71 IF PresionInterpolada > 5 THEN
72 CondicionesInicialesOK := TRUE;
73 CondicionesIniciales_Run := FALSE;
74 j := j + 10;
75 END_IF
76 END_CASE
77 END_IF
78
```

12.4.4. GESTION_ALARMAS

```
1 //IF EtherCAT_Master.xDistributedClockInSync THEN
2 IF Marca STO OR Home Error Agarre OR Home Error Sierra OR MoveAbsolute Error Agarre OR MoveAbsolute Error Sierra
3 OR Power Error Agarre OR Power Error Sierra OR Reset Error Agarre OR Reset Error Sierra OR Error Presion
4 OR Torque_Error_Agarre OR Torque_Error_Sierra OR Error_Comunicaciones THEN
5 CASE o OF
6 0 :
7     Existe Alarma := TRUE ;
8     Temporizador_Led_Alarma .TN := TRUE ;
9     Temporizador_Led_Alarma .PT := T#0.5S ;
10    IF Temporizador_Led_Alarma .Q THEN
11        Temporizador_Led_Alarma .IN := FALSE ;
12        o := o + 10 ;
13    END_IF
14
15    10 :
16        Existe Alarma := FALSE ;
17        Temporizador_Led_Alarma .IN := TRUE ;
18        IF Temporizador_Led_Alarma .Q THEN
19            Temporizador_Led_Alarma .TN := FALSE ;
20            o := 0 ;
21        END_IF
22    END_CASE
23 ELSE
24     o := 0 ;
25     Existe_Alarma := FALSE ;
26     Temporizador_Led_Alarma .TN := FALSE ;
27 END_IF
28 //END_IF
29
30 IF PresionInterpolada > 4.5 THEN
31     Error_Presion := FALSE ;
32 ELSE
33     Error_Presion := TRUE ;
34 END_IF
35
36 IF NOT EtherCAT_Master .xDistributedClockInSync AND NOT EtherCAT_Master .xConfigFinished THEN
37     Temporizador_Error_Ethercat .TN := TRUE ;
38     Temporizador_Error_Ethercat .PT := T#10S ;
39     IF Temporizador_Error_Ethercat .Q THEN
40         Error_Comunicaciones := TRUE ;
41     END_IF
42 ELSE
43     Temporizador_Error_Ethercat .TN := FALSE ;
44     Error_Comunicaciones := FALSE ;
45 END_IF
46
```

12.4.5. GESTION_MOVIMIENTOS

```
1  IF CondicionesInicialesOK THEN
2  IF Marcha OK AND NOT NoRepetir THEN
3      i := 0;
4      NoRepetir := TRUE;
5  END_IF
6
7  IF NoRepetir THEN
8      CASE i OF
9
10         0 : // Se agarra el jamón con el cilindro neumático.
11             CilindroN_Pata := TRUE;
12             Temporizador_CilindroN_Pata.IN := TRUE;
13             Temporizador_CilindroN_Pata.PT := T#2S;
14             i := i + 10;
15
16         10 : // Cuando se ha agarrado el jamón, se mueve el cilindro eléctrico por control
17             // de par y se activa la sierra para cortar la pata,
18             IF Temporizador_CilindroN_Pata.Q THEN
19                 IF NOT Sensor_CilindroN_Pata_Fin THEN
20                     Info_Falta_Jamon := FALSE;
21                     Pinchos := TRUE;
22                     IF Selector_Discriminacion_Corte THEN
23                         Marca_CilindroSierra := TRUE;
24                         PruebaCilindroN_Pata_Fin := FALSE;
25                         k := 0;
26                         i := i + 10;
27                     ELSE
28                         i := 30;
29                     END_IF
30                     Temporizador_CilindroN_Pata.IN := FALSE;
31                 ELSE
32                     Pinchos := FALSE;
33                     CilindroN_Pata := FALSE;
34                     Temporizador_CilindroN_Pata.IN := FALSE;
35                     Info_Falta_Jamon := TRUE;
36                     NoRepetir := FALSE;
37                     Marcha_OK := FALSE;
38                 END_IF
39             END_IF
40
41         20 : // Cuando ha terminado de cortar la pata, se para la cinta y de mueve al inicio
42             // el cilindro eléctrico por control de posición.
43             IF Marca_CilindroSierra_Fin THEN
44                 Marca_CilindroSierra := FALSE;
45                 i := i + 10;
46             END_IF
47
48         30 : // Los operarios ponen a mano el agarre del extremo del jamón y una vez pongan el
49             // selector en marcha, el cilindro eléctrico 2 empieza a tirar del jamón por
50             // control de par.
51             IF SelectorExtremoJamonOk THEN
52                 CilindroN_Agarre := TRUE;
53                 Marca_CilindroSierra_Fin := FALSE;
54                 i := i + 10;
55             END_IF
56
57         40 : // Los operarios empiezan a cortar el jamón a mano. Cuando llegan a un punto quitan
58             // el selector y ponen la cadena del cilindro eléctrico.
59             IF NOT SelectorExtremoJamonOk THEN
60                 CilindroN_Agarre := FALSE;
61                 i := i + 10;
62             END_IF
63
64         50 : // Cuando el operario ha quitado el cilindro agarre extremo y ha puesto la cadena
65             // del eléctrico, si se pulsa el botón, actua el cilindro eléctrico por control de
66             // par para que el operario siga cortando el jamón a mano.
67             IF SelectorExtremoJamonOk THEN // Si por lo que sea tienen que volver a oclocar el cilindro neumático.
68                 i := 30;
69             END_IF
70
71             IF Boton_Cadona OR PruebaBotonCadona THEN
72                 Marca_CilindroAgarreE := TRUE;
73                 n := 0;
74                 i := i + 10;
75             END_IF
76
77         60 : // Paso intermedio.
78             IF NOT Boton_Cadena THEN
79                 i := i + 10;
80             END_IF
81
82         70 : // Cuando terminan de cortar el jamón, si vuelven a pulsar el botón el cilindro
83             // vuelve a la posición inicial, por lo que se suelta el jamón.
84             IF (*Boton_Cadena AND*) Marca_CilindroAgarreE_Fin THEN
85                 Marca_CilindroAgarreE := FALSE;
86                 Marca_CilindroAgarreE_Fin := FALSE;
87                 i := i + 5;
88             END_IF
89
90         75 :
91             IF NOT Boton_Cadena THEN
92                 i := i + 5;
93             END_IF
```

```
94         80 :
95         IF Boton_Cadena THEN
96             i := 50 ;
97         END_IF
98
99         IF NOT Sensor_Boton_Despresurizacion THEN
100             Marcha_OK := FALSE ;
101             Pinchos := FALSE ;
102             NoRepetir := FALSE ;
103             FinCorte := FALSE ;
104             CilindroN_Pata := FALSE ;
105             Enable_HMI := FALSE ;
106             ContadorJamones := ContadorJamones + 1 ;
107             i := i + 10 ;
108         END_IF // Fin del proceso.
109     END_CASE
110 END_IF
111 END_IF
112
113 IF CilindroN_Pata THEN // Simulación movimiento HMI.
114     PosicionPata_HMI := 25 ;
115 ELSE
116     PosicionPata_HMI := 0 ;
117 END_IF
118
119 IF CilindroN_Agarre THEN // Simulación movimiento HMI.
120     PosicionAgarreN_HMI := 63 ;
121 ELSE
122     PosicionAgarreN_HMI := 0 ;
123 END_IF
124
125 IF Bajar_Maquina_HMI AND NOT Marca_STO THEN
126     CilindroN_Elevacion_A := TRUE ;
127     CilindroN_Elevacion_B := FALSE ;
128 ELSIF Subir_Maquina_HMI AND NOT Marca_STO THEN
129     CilindroN_Elevacion_A := FALSE ;
130     CilindroN_Elevacion_B := TRUE ;
131 ELSE
132     Bajar_Maquina_HMI := FALSE ;
133     Subir_Maquina_HMI := FALSE ;
134     CilindroN_Elevacion_A := FALSE ;
135     CilindroN_Elevacion_B := FALSE ;
136 END_IF
137
```

12.4.6. GESTION_MOVIMIENTOS ELÉCTRICOS

```

1  (* ***** *)
2  (* ***** Gestión Movimientos ***** *)
3  (* ***** *)
4
5  IF CondicionesInicialesOK THEN
6      IF Marca_CilindroSierra THEN
7          CASE k OF
8              0 : // Empieza a moverse el cilindro, se activa la sierra y cuando ha terminado
9                  // de cortarse paran.
10                 // IF (Marcha_Fis_1 AND Marcha_Fis_2) THEN
11                     Stop_Execute_Sierra := FALSE;
12                     Motor_Sierra_Marcha := TRUE;
13                     Corte := TRUE;
14                     Temporizador_Corte.IN := TRUE;
15                     Temporizador_Corte.PT := T#2S;
16                     IF Temporizador_Corte.Q THEN
17                         Torque_Execute_Sierra := TRUE;
18                         IF ReadPos_Actual_Sierra >= MarcaPosicionLimite_CilindroSierra THEN
19                             Stop_Execute_Sierra := TRUE;
20                             Torque_Execute_Sierra := FALSE;
21                             Motor_Sierra_Marcha := FALSE;
22                             Velocidad_Variador_Giro := 0;
23                             FinCorte := TRUE;
24                             Corte := FALSE;
25                             Temporizador_Corte.IN := FALSE;
26                             k := k + 10;
27                         END_IF
28
29                         IF Read_Velocidad_Variador_Giro < Velocidad_Variador_Giro * 0.5 THEN
30                             Stop_HMI := TRUE;
31                         END_IF
32                     END_IF
33                 (*ELSE
34                     Temporizador_Corte.IN := FALSE;
35                     Torque_Execute_Sierra := FALSE;
36                     Motor_Sierra_Marcha := FALSE;
37                     Velocidad_Variador_Giro := 0;
38                     Stop_Execute_Sierra := TRUE;
39                     Corte := FALSE;
40                     k := k + 5;
41                 END_IF
42
43                 5: // Pasos intermedios repetir el movimiento 0.
44                     Stop_Execute_Sierra := FALSE;
45                     Temporizador_Corte.IN := FALSE;
46                     IF (Marcha_Fis_1 AND Marcha_Fis_2) AND NOT Stop_Execute_Sierra THEN
47                         Torque_Execute_Sierra := TRUE;
48                         Temporizador_Prueba.IN := TRUE;
49                         Temporizador_Prueba.PT := T#0.5S;
50                         k := 7;
51                     END_IF
52
53                 7: // Pasos intermedios repetir el movimiento 0.
54                     IF Temporizador_Prueba.Q THEN
55                         Torque_Execute_Sierra := FALSE;
56                         Temporizador_Prueba.IN := FALSE;
57                         Temporizador_Prueba_2.IN := TRUE;
58                         Temporizador_Prueba_2.PT := T#0.5S;
59                         Stop_Execute_Sierra := TRUE;
60                         k := 8;
61                     END_IF
62
63                 8:
64                     IF Temporizador_Prueba_2.Q THEN
65                         Temporizador_Prueba_2.IN := FALSE;
66                         k := 0;
67                     END_IF*)
68
69                 10 : // Si se ha parado, comienza el movimiento de devolver el cilindro a la posición
70                     // de inicio.
71                     IF Stop_Done_Sierra THEN
72                         Stop_Execute_Sierra := FALSE;
73                         Temporizador_Marca_Sierra.IN := TRUE;
74                         Temporizador_Marca_Sierra.PT := T#0.5S;
75                         k := k + 10;
76                     END_IF
77
78                 20 : // Si ha llegado a la posición inicial, termina el proceso.
79                     IF Temporizador_Marca_Sierra.Q THEN
80                         MoveAbsolute_Execute_Sierra := TRUE;
81                         IF MoveAbsolute_Done_Sierra THEN
82                             MoveAbsolute_Execute_Sierra := FALSE;
83                             Temporizador_Marca_Sierra.IN := FALSE;
84                             Marca_CilindroSierra_Fin := TRUE;
85                             k := k + 10;
86                         END_IF
87                     END_IF
88                 END_CASE
89             END_IF

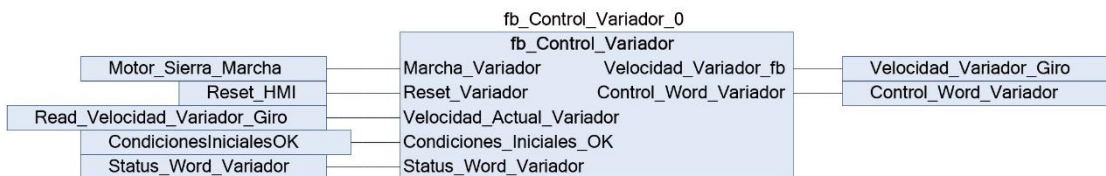
```

```

90
91     IF Marca_CilindroAgarreE THEN
92         CASE n OF
93             0 :
94                 IF NOT Boton_Cadena THEN
95                     n := n + 10 ;
96                 END_IF
97
98             10 : // Empieza a moverse el cilindro, tirando del jamón mediante una cadena. Cuando
99                 // llega a una posición o termina de hacer la fuerza marcada, se para. (HAY QUE HACERLO).
100                Torque_Executo_Agarre := TRUE ;
101                IF ReadPos_Actual_Agarre <= MarcaPosicionLimite_CilindroAgarreE + 30 OR Boton_Cadena THEN
102                    Stop_Executo_Agarre := TRUE ;
103                    Torque_Executo_Agarre := FALSE ;
104                    Motor_Sierra_Marcha := FALSE ;
105                    n := n + 10 ;
106                END_IF
107
108             20 : // Si se ha parado, comienza el movimiento de devolver el cilindro a la posición
109                 // de inicio.
110                IF Stop_Done_Agarre THEN
111                    Stop_Executo_Agarre := FALSE ;
112                    Temporizador_Marca_Agarre . TN := TRUE ;
113                    Temporizador_Marca_Agarre . PT := T#0.5S ;
114                    n := n + 10 ;
115                END_IF
116
117             30 : // Si ha llegado a la posición inicial, termina el proceso.
118                IF Temporizador_Marca_Agarre . Q THEN
119                    MoveAbsolute_Executo_Agarre := TRUE ;
120                    n := n + 10 ;
121                END_IF
122
123             40 :
124                IF MoveAbsolute_Done_Agarre THEN
125                    MoveAbsolute_Executo_Agarre := FALSE ;
126                    Marca_CilindroAgarreE_Fin := TRUE ;
127                    Temporizador_Marca_Agarre . TN := FALSE ;
128                    n := n + 10 ;
129                END_IF
130            END_CASE
131        END_IF
132    END_IF
133
134

```

12.4.7. GESTION_VARIADOR



12.5. PROGRAMA PRINCIPAL (HMI)

```

1      (* ***** *)
2      (* ***** Gestión HMI ***** *)
3      (* ***** *)
4
5      // Temporizadores:
6      TemporizadorMovMenu ( IN := , PT := T#0.000000001S );
7      TemporizadorMenuCerrar ( IN := , PT := T#15S );
8      TemporizadorAJPred ( IN := , PT := T#1S );
9      Temporizador_Operario ( );
10
11     // Gestión movimiento menú en la pantalla:
12     IF BotonMenu_HMI AND NOT NoReturn_BM THEN
13         Marcha_BM := TRUE ;
14         Inicio_NoReturn_BM := TRUE ;
15         NoReturn_BM := TRUE ;
16         MovimientoMenu := 0 ;
17         l := 10 ;
18     ELSIF NOT BotonMenu_HMI THEN
19         Marcha_BM := FALSE ;
20         Inicio_NoReturn_BM := FALSE ;
21         NoReturn_BM := FALSE ;
22         m := 10 ;
23     END_IF
24
25     IF BotonMenu_HMI THEN
26         TemporizadorMenuCerrar.IN := TRUE ;
27         IF TemporizadorMenuCerrar.Q THEN
28             BotonMenu_HMI := FALSE ;
29             TemporizadorMenuCerrar.IN := FALSE ;
30         END_IF
31     ELSE
32         TemporizadorMenuCerrar.IN := FALSE ;
33     END_IF
34
35     IF Marcha_BM AND Inicio_NoReturn_BM THEN
36         CASE l OF
37             10 :
38                 TemporizadorMovMenu.IN := TRUE ;
39                 IF TemporizadorMovMenu.Q THEN
40                     MovimientoMenu := MovimientoMenu + 10 ;
41                     TemporizadorMovMenu.IN := FALSE ;
42                     IF MovimientoMenu >= 190 THEN
43                         l := l + 20 ;
44                     ELSE
45                         l := l + 10 ;
46                     END_IF
47                 END_IF
48             20 :
49                 MovimientoMenu := MovimientoMenu + 10 ;
50                 IF MovimientoMenu >= 190 THEN
51                     l := l + 10 ;
52                 ELSE
53                     l := 10 ;
54                 END_IF
55             END_CASE
56     ELSIF NOT Marcha_BM AND MovimientoMenu >= 190 THEN
57         VueltaMovMenu := TRUE ;
58     END_IF
59
60     // Movimiento Menú:
61     IF VueltaMovMenu THEN
62         CASE m OF
63             0 :
64                 TemporizadorMovMenu.IN := TRUE ;
65                 IF TemporizadorMovMenu.Q THEN
66                     MovimientoMenu := MovimientoMenu - 10 ;
67                     TemporizadorMovMenu.IN := FALSE ;
68                     IF MovimientoMenu = 0 THEN
69                         VueltaMovMenu := FALSE ;
70                         m := m + 20 ;
71                     ELSE
72                         m := m + 10 ;
73                     END_IF
74                 END_IF
75             10 :
76                 MovimientoMenu := MovimientoMenu - 10 ;
77                 IF MovimientoMenu = 0 THEN
78                     VueltaMovMenu := FALSE ;
79                     m := m + 10 ;
80                 ELSE
81                     m := 0 ;
82                 END_IF
83             END_CASE

```

```
84         END_IF
85     END_CASE
86 END_IF
87
88 // Funcionamiento potenciómetro:
89 IF ( Potenciómetro_Mas_HMI ) AND ( Potenciómetro_HMI >= 0 AND Potenciómetro_HMI < 10 ) AND Marca_1 = 0 THEN
90     Potenciómetro_HMI := Potenciómetro_HMI + 1;
91     Marca_1 := 1;
92 ELSIF NOT Potenciómetro_Mas_HMI THEN
93     Marca_1 := 0;
94 END_IF
95
96 IF ( Potenciómetro_Menos_HMI ) AND ( Potenciómetro_HMI > 0 AND Potenciómetro_HMI <= 10 ) AND Marca_2 = 0 THEN
97     Potenciómetro_HMI := Potenciómetro_HMI - 1;
98     Marca_2 := 1;
99 ELSIF NOT Potenciómetro_Menos_HMI THEN
100     Marca_2 := 0;
101 END_IF
102
103 Regla3_Torque_CilindroAgarreE_HMI := 1.24 * 10 / 100;
104 Potenciómetro_Torque_CilindroAgarreE_HMI := Regla3_Torque_CilindroAgarreE_HMI * Potenciómetro_HMI;
105
106 // Funcionamiento contador jamones manual:
107 IF Jamon_Mas_HMI AND Marca_3 = 0 THEN
108     Jamon_Mas_HMI := FALSE;
109     ContadorJamones := ContadorJamones + 1;
110     Marca_3 := 1;
111 ELSIF NOT Jamon_Mas_HMI THEN
112     Marca_3 := 0;
113 END_IF
114
115 IF Jamon_Menos_HMI AND ContadorJamones > 0 AND Marca_4 = 0 THEN
116     Jamon_Menos_HMI := FALSE;
117     ContadorJamones := ContadorJamones - 1;
118     Marca_4 := 1;
119 ELSIF NOT Jamon_Menos_HMI THEN
120     Marca_4 := 0;
121 END_IF
122
123 IF Reset_Contador THEN
124     Reset_Contador := FALSE;
125     ContadorJamones := 0;
126 END_IF
127
128 // Gestión Usuarios (No del PLC, si no operarios que usen la máquina):
129 IF EtherCAT_Master . xDistributedClockInSync AND EtherCAT_Master . xConfigFinished THEN
130     IF nombreOperario = 0 THEN
131         MensajeIntroducirOperario := TRUE;
132         giFramePrincipal := 5;
133     ELSE
134         MensajeIntroducirOperario := FALSE;
135         IF Temporizador Operario . Q THEN
136             nombreOperario := 0;
137             Temporizador Operario . IN := FALSE;
138         END_IF
139
140         IF Boton_SalirOperario THEN
141             nombreOperario := 0;
142             Boton_SalirOperario := FALSE;
143         END_IF
144     END_IF
145     Temporizador Operario . PT := INT_TO_TIME ( TiempoDescansoOperario ) * 60 * 1000;
146 END_IF
147
```

12.6. PROGRAMA PRINCIPAL (MOV_SIM_HMI)

```

1  (* ***** *)
2  (* ***** Simulaciones HMI ***** *)
3  (* ***** *)
4
5  // Simulación giro:
6  IF Corte THEN
7    CASE w OF
8      0 :
9        Giro_HMI := 1500 ;
10       w := w + 10 ;
11
12      10 :
13        Giro_HMI := 3600 ;
14        w := w + 10 ;
15
16      20 :
17        Giro_HMI := 5200 ;
18        w := 0 ;
19
20    END_CASE
21  ELSIF FinCorte THEN
22    Giro_HMI := 0 ;
23  END_IF
24
25
26  //      541 inicio - 461 fin
27  // Movimiento Sierra HMI:
28  PosicionSierra_HMI := ReadPos_Actual_Sierra ;
29  PosicionAgarre_HMI := ReadPos_Actual_Agarre * 100 / 300 ;
30

```

12.7. ENTRADAS/SALIDAS

Las entradas salidas utilizadas son las del módulo de Ethercat, que son entradas/salidas digitales, y la entrada analógica se ha utilizado la que dispone el propio PLC.

12.7.1. ENTRADAS DIGITALES

Variable	Asignación	Canal	Dirección	Tipo
16#1A01 Inputs				
Application.Marcha_Fis_1		Input 0	%IX50.0	BIT
Application.Marcha_Fis_2		Input 1	%IX50.1	BIT
Application.Seta		Input 2	%IX50.2	BIT
Application.Boton_Cadena		Input 3	%IX50.3	BIT
Application.SelectorExtremoJamonOk		Input 4	%IX50.4	BIT
Reserva_I_6		Input 5	%IX50.5	BIT
Application.Sensor_CilindroN_Pata_Fin		Input 6	%IX50.6	BIT
Application.Sensor_Boton_Despresuriz...		Input 7	%IX50.7	BIT
Application.Selector_Discriminacion_C...		Input 8	%IX51.0	BIT
Application.Enable_Fis		Input 9	%IX51.1	BIT
Application.Sensor_Puerta		Input 10	%IX51.2	BIT
Reserva_I_1		Input 11	%IX51.3	BIT
Reserva_I_2		Input 12	%IX51.4	BIT
Reserva_I_3		Input 13	%IX51.5	BIT
Reserva_I_4		Input 14	%IX51.6	BIT
Reserva_I_5		Input 15	%IX51.7	BIT

12.7.2. SALIDAS DIGITALES




























Variable	Asignación	Canal	Dirección	Tipo
16#1602 Outputs				
Application.CilindroN_Agarre		Output 0	%QX60.0	BIT
Application.CilindroN_Pata		Output 1	%QX60.1	BIT
Application.CilindroN_Elevacion_B		Output 2	%QX60.2	BIT
Application.CilindroN_Elevacion_A		Output 3	%QX60.3	BIT
Application.Pinchos		Output 4	%QX60.4	BIT
Application.CondicionesInicialesOK		Output 5	%QX60.5	BIT
Application.ValvulaCorte		Output 6	%QX60.6	BIT
Reserva_Q_4		Output 7	%QX60.7	BIT

12.7.3. ENTRADA ANALÓGICA

Variable	Asignación	Canal	Dirección	Tipo	Valor prede
Digital In					
		Digital Input 0	%IB52	BYTE	
		Digital Input 1	%IB53	BYTE	
		Digital Input 2	%IB54	BYTE	
		Digital Input 3	%IB55	BYTE	
		Digital Input 4	%IB56	BYTE	
Digital Out					
		Digital Output 0	%QB64	BYTE	
		Digital Output 1	%QB65	BYTE	
		Digital Output 2	%QB66	BYTE	
		Digital Output 3	%QB67	BYTE	
		Digital Output 4	%QB68	BYTE	
Analog In					
		Channel 0	%ID15		
Application.PresionAire		Value	%ID15	DINT	
		AI_U	%ID16	REAL	
		AI_I	%ID17	REAL	
		PT	%ID18	REAL	
		Status	%IB76	BYTE	

12.8. ALARMAS

Las alarmas que se han introducido en el programa hasta el momento son las siguientes:

ID	Tipo de supervisión	Detalles		Clase	Mensaje
0	 Digital	Marca_Seta = TRUE	<input type="checkbox"/>	 Error	Seta Pulsada
1	 Digital	Home_Error_Agarre = TRUE	<input type="checkbox"/>	 Error	Error Homing Cilindro Agarre
2	 Digital	Home_Error_Sierra = TRUE	<input type="checkbox"/>	 Error	Error Homing Cilindro Sierra
3	 Digital	MoveAbsolute_Error_Agarre = TRUE	<input type="checkbox"/>	 Error	Error MoveAbsolute Cilindro Agarre
4	 Digital	MoveAbsolute_Error_Sierra = TRUE	<input type="checkbox"/>	 Error	Error MoveAbsolute Cilindro Sierra
5	 Digital	Power_Error_Agarre = TRUE	<input type="checkbox"/>	 Error	Error Enable Cilindro Agarre
6	 Digital	Power_Error_Sierra = TRUE	<input type="checkbox"/>	 Error	Error Enable Cilindro Sierra
7	 Digital	Reset_Error_Agarre = TRUE	<input type="checkbox"/>	 Error	Error Reset Cilindro Agarre
8	 Digital	Reset_Error_Sierra = TRUE	<input type="checkbox"/>	 Error	Error Reset Cilindro Sierra
11	 Digital	Torque_Error_Agarre = TRUE	<input type="checkbox"/>	 Error	Error Torque Cilindro Agarre
12	 Digital	Torque_Error_Sierra = TRUE	<input type="checkbox"/>	 Error	Error Torque Cilindro Sierra
13	 Digital	Marca_Puerta = TRUE	<input type="checkbox"/>	 Error	Puerta Abierta
14	 Digital	Error_Presion = TRUE	<input type="checkbox"/>	 Error	Presión Insuficiente
15	 Digital	Error_Comunicaciones = TRUE	<input type="checkbox"/>	 Error	Error Comunicación Ethercat
16	 Digital	Info_Falta_Jamon = TRUE	<input type="checkbox"/>	 Info	Falta introducir Jamón

12.9. HMI

12.9.1. PANTALLA PRINCIPAL

La pantalla principal sería la siguiente:



Para ir a las diferentes pantallas (cambiará solo el marco) si se pulsa en la parte roja de la derecha se despliega el siguiente menú para la selección:



12.9.2. ÁREAS PRINCIPALES

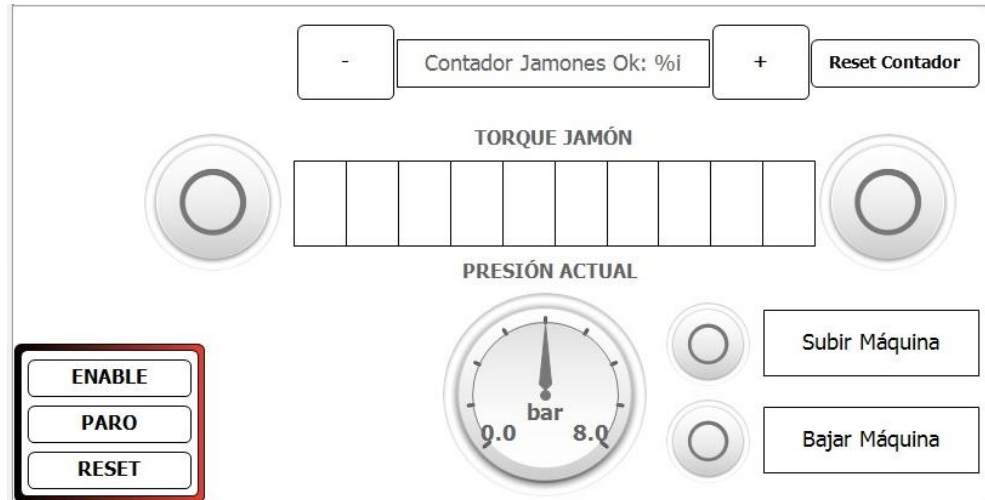
Existen dos áreas principales:

- Área del título y visualización de leds importantes.



- Área de menús: en esta área estarán todas las diferentes pantallas del HMI.

12.9.2.1.A_SMenu_MarchaParo



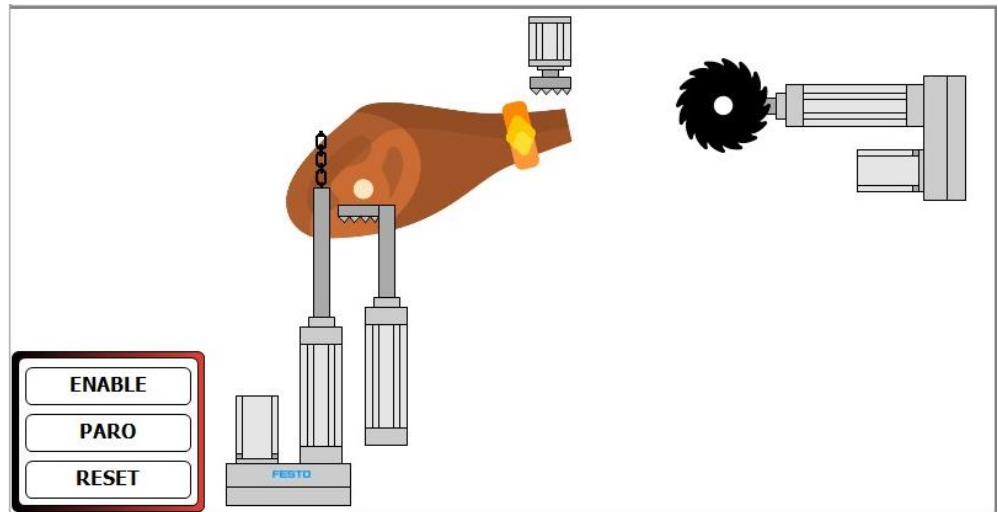
12.9.2.2.B_SMenu_Ajustes



12.9.2.3.C_SMenu_EstadoES



12.9.2.4.D_SMenu_Grafico



12.9.2.5.E_SMenu_Alarmas

Hora	Descripción	Valor
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

12.9.2.6.F_SMenu_Usuarios

