

GRADO EN INGENIERÍA ELÉCTRICA

TRABAJO FIN DE GRADO

**CONTROL Y VISUALIZACIÓN
DEL PROCESO
DE ELABORACIÓN DE LA
SALSA
DE MOSTAZA**

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019

ÍNDICE GENERAL:

1. MEMORIA
2. PLIEGO DE CONDICIONES
3. PRESUPUESTO
4. PESQUEMAS ELÉCTRICOS
5. ANEXOS

GRADO EN INGENIERÍA ELÉCTRICA
TRABAJO FIN DE GRADO

MEMORIA

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019

ÍNDICE

1.- Introducción	1
2.-Antecedentes	2
3.- Consideraciones	4
4.- Objetivos.....	7
5.-Descripción del proceso	10
5.1 Zona de dosificado (Azul):.....	13
5.2 Zona de pesado y mezclado (Amarillo):.....	16
5.3 Zona de triturado y cocido (Naranja):.....	17
6.- Modos de funcionamiento	18
6.1 Modo manual.....	19
6.2 Modo automático	21
7.- Solución adoptada para el control del proceso	23
7.1 Necesidades a cubrir por el hardware	23
7.2 Estudio de alternativas del hardware	25
7.3 Necesidades a cubrir por el software	28
7.4 Programa de control	30
7.4.1 Bloque organización principal MAIN [OB1]	32
7.4.2 Bloque organización “Modo_automat” [OB124].....	36
7.4.3 Bloque organización “Mov_manuales” [OB123]	37
7.5 Desarrollo de una librería de bloques de función	38
7.5.1 Bloques de función generales destinados a la librería	39
7.5.2 Crear una librería.....	50
7.5.3 Bloques de función específicos (únicamente para el proyecto):.....	52
7.6 Entorno HMI, solución adoptada para el SCADA.....	62
7.6.1 Pantalla de inicio	62
7.6.2 Gestión del sistema de almacenaje	63
7.6.3 Histórico de alarmas y control de fallos.....	64
7.6.4 visualización del estado de los silos	70
7.6.4 Interfaz movimientos manuales.....	71
7.6.5 Interfaz modo automático	72

8.- Conclusiones.....	75
-----------------------	----

ÍNDICE DE TABLAS

Tabla 1 Necesidades de hardware para el control	23
Tabla 2 Principales OB's para la gama s7-1200 fuente: http://support.automation.siemens.com/	33
Tabla 3 Variables declaradas FB "Silo"	39
Tabla 4 Variables declaradas bloque de función "Control_Histéresis"	43
Tabla 5 Variables declaradas "Esc_señal_analog"	47
Tabla 6 Fallos y avisos mostrados en el HMI	69

ÍNDICE DE FIGURAS

Figura 1 Esquema Fase 1: disposición del vertido y mezclado de los ingredientes	2
Figura 2 Esquema Fase 2: trasvasado de la mezcla al triturador	3
Figura 3 Esquema control proceso industrial	6
Figura 4 Grafico comparativo activación actuadores vs tiempo de carga	11
Figura 5 Esquema de la instalación	12
Figura 6 Diagrama funcionamiento general	18
Figura 7 Esquema de funcionamiento del modo manual	19
Figura 8 Esquema funcionamiento del modo automático	21
Figura 9 PLC Micro820 Allen Bradley	25
Figura 10 PLC Electric ModiconM421 Schenider Electric	25
Figura 11 PLC Siemens S7-1200	25
Figura 12 PLC Micro820 Allen Bradley	26
Figura 13 Ampliación TMC4HOIS01 1 AI	26
Figura 14 Ampliación pasa S7-1200, SM1223	26
Figura 15 Requisitos hardware para el soporte del WinCC	29
Figura 16 Vista completa árbol del proyecto	30
Figura 17 Vista de los distintos tipos de bloques de organización	32
Figura 18 Funcionamiento básico OB1	34
Figura 19 Ejemplo de estructuración de un programa complejo	34
Figura 20 Cronograma ejecución OB's	35
Figura 21 Diagrama de funcionamiento "Modo_automat"	36
Figura 22 Diagrama de funcionamiento "Mov_manuales"	37
Figura 23 Visualización función SILO	40

Figura 24 Ejemplo conexasión función SILO	42
Figura 25 Ejemplo de un control por histéresis. Fuente: https://zulaco64.updog.co/Material_Didactico/Control-OnOff_Sistemas_Control_Controladores.pdf	44
Figura 26 Vista de la función "Control_histéresis"	45
Figura 27 Ejemplo conexasión función "Control_histéresis"	46
Figura 28 Vista función "Esc_señal_analog"	48
Figura 29 Ejemplo conexasión de la función "Esc_señal_analog"	49
Figura 30 Primer paso para crear una librería de FB's	50
Figura 31 Segundo paso para crear una librería de FB's	51
Figura 32 Vista de la librería creada	51
Figura 33 Ejemplo del tipo de dato "Parámetros_silos"	52
Figura 34 Ejemplo creación de las alarmas a mostrar en el HMI	54
Figura 35 Vista interior del DB "Alarmas_silos"	55
Figura 36 Vista al llamar la función "Alarmas_y_fallos_SILOS"	56
Figura 37 Vista función "Movimientos_manuales"	58
Figura 38 Ejemplo conexasión conjunto "Movimientos_manuales" y "Esc_señal_analog"	59
Figura 39 Ejemplo conexasiones en la función "Activación_actuadores"	60
Figura 40 Vista al realizar la llamada de las funciones "Alarmas_y_fallos_SILO" y "Activación_actuadores"	61
Figura 41 Pantalla de inicio.....	62
Figura 42 Pantalla de parametrización de la planta	63
Figura 43 Pantalla del histórico de alarmas	64
Figura 44 Pantalla monitorización estado de los silos	70
Figura 45 Pantalla movimientos manuales	71
Figura 46 Pantalla selección receta	72
Figura 47 Pantalla modo automático	73

1.- INTRODUCCIÓN

Actualmente en gran cantidad de industrias se hace necesario tener un control sobre las cantidades de materias primas que se emplean en la fabricación de diversos productos. Para el control de estas cantidades se recurre a sistemas de dosificación y pesaje.

Independientemente del tipo de industria que requiera de dosificación y pesaje, la formulación requiere precisión. Una mezcla incorrecta de los distintos componentes empleados en el proceso productivo siempre es motivo de pérdidas económicas para la empresa al tener que desechar materias primas o invertir recursos, ya sean materiales o humanos, en reaprovechar o reciclar los componentes de la mezcla incorrecta. Además, según el campo de aplicación mezclar cantidades erróneas de material puede resultar peligroso, como puede ser en la industria farmacéutica, en la fabricación de productos químicos de limpieza o pinturas, etc.

Por estos motivos, los sistemas de pesaje y dosificación son ampliamente utilizados en un gran variedad de procesos industriales, en una aplicación tan extensa que cubre desde el ensacado de piensos para alimentación animal, hasta la fabricación de medicamentos, con proporciones extraordinariamente precisas.

En función del tipo de industria los sistemas de dosificación y pesaje varían para adaptarse a las necesidades de cada proceso puesto que los requisitos de cada empresa pueden variar desde la necesidad de controlar el peso en gramos a tener que pesar varias toneladas.

Actualmente hay una gran variedad de sistemas que buscan adaptarse del modo más eficaz a cada proceso productivo. En la fabricación de productos relacionados con la cerámica o el cemento, es habitual encontrar sistemas de pesaje continuo mediante cintas transportadoras. En otros procesos como el ensacado de grano, se tiende al uso de básculas gravimétricas que descargan al llegar a un peso determinado.

En este trabajo de fin de grado se busca controlar un sistema de dosificación y pesaje para el proceso productivo de la salsa de mostaza. Para ello se hará uso de varios actuadores tales como electroválvulas, motores y elementos calefactores, y de una única balanza, a través de los cuales se pretende ajustar distintas cantidades de material a mezclar para obtener productos diferentes variando las proporciones de las mismas materias primas.

2.- ANTECEDENTES

Se parte de una hipotética fábrica de salsa que pretende automatizar su producción, aunque no se tenga una instalación física, se va a trabajar en base a cubrir las distintas necesidades que se pudieran derivar de una fabricación real.

En la instalación, para el almacenamiento de las materias primas, es decir, de los distintos ingredientes necesarios en la producción de las distintas variantes de la salsa de mostaza, se emplean distintos tipos de silos.

Conforme a la disposición actual de la fábrica, para iniciar la producción de la salsa es necesario que un operario vierta silo por silo las cantidades necesarias de materia prima en un tanque de mezclado que descansa sobre una báscula, como se muestra en la figura 1. Esta báscula no realiza discriminación, es decir, el operario únicamente es capaz de visualizar la lectura del peso total del material en conjunto que se ha vertido en el tanque, y no el peso de cada materia prima por separado.

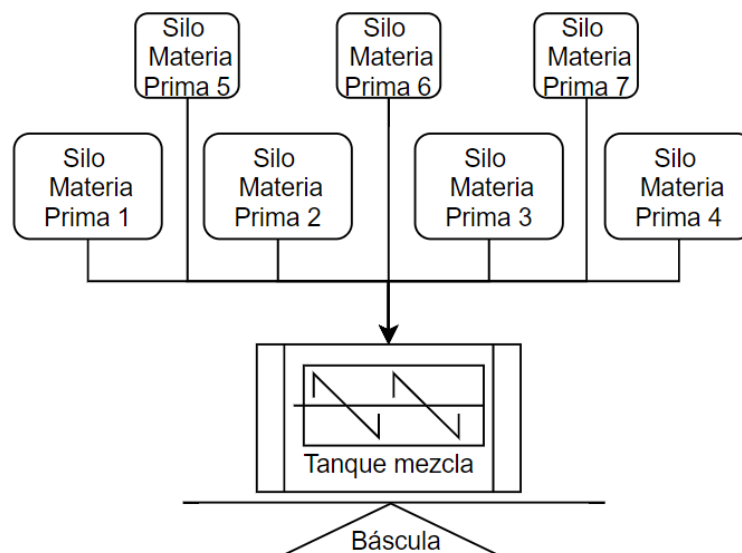


Figura 1 Esquema Fase 1: disposición del vertido y mezclado de los ingredientes

El tanque dispone incorporado un motor que acciona un eje con paletas para realizar la acción del mezclado de las materias primas. El accionamiento de este motor es según criterio del operario, es decir, una vez el operario ha terminado de verter los ingredientes y está conforme con la lectura marcada por la báscula, él mismo se encarga de encender este motor para realizar la fase del mezclado. Esta fase finaliza cuando el operario considera que la mezcla es homogénea y es entonces cuando procede a la desconexión del motor.

Al finalizar esta fase de la producción, la mezcla se considera adecuada tanto en sus proporciones como en la homogeneidad de la distribución de sus componentes. Por lo que, como muestra la figura 2, se procederá a realizar el trasvase del material a otra zona de la planta donde se realizará el cocido y texturizado de la salsa mediante un triturador calefactado.

La figura 2 representa como este trasvase se realiza mediante el empleo de una bomba succionadora, que se encargará de enviar el material desde el tanque de mezclado hasta el triturador.

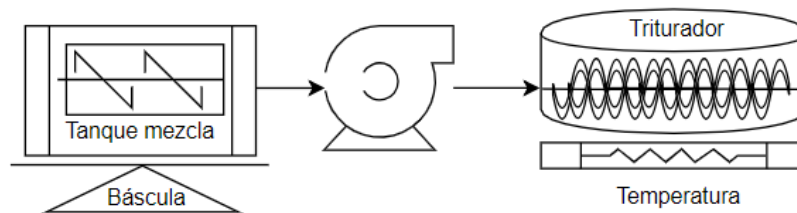


Figura 2 Esquema Fase 2: trasvasado de la mezcla al triturador

Al igual que para el vertido y mezclado, en esta fase del proceso el operario es el responsable asegurar que todo el material que había en el tanque de mezcla ha sido transportado al tanque del triturador. Una vez el propio operario ha determinado que el trasvasado se ha efectuado correctamente, es él de nuevo, el encargado de accionar tanto el motor del triturador como las resistencias de calentamiento.

El triturador permanecerá encendido tanto tiempo como el operario estime oportuno, pues la conexión y desconexión del triturador así como de las resistencias es completamente manual. La temperatura dependerá pues, de si el operario enciende las resistencias y las deja funcionando al 100% durante el tiempo que esté el triturador encendido, o por el contrario las va encendiendo y apagando en función de la temperatura a la que se encuentre el tanque en ese momento, realizando de este modo un control manual, cuya precisión depende del grado de atención que esté prestando el operario.

Dada la descripción de la ejecución actual del proceso, es fácilmente deducible que el proceso productivo descansa de manera demasiado sensible sobre el personal encargado de la fabricación en planta, por lo que el producto resultante, difícilmente tendrá una buena repetibilidad en cuanto a lo que refiere a las características de los lotes de fabricación.

3.- CONSIDERACIONES

Para mejorar la identificación por parte de los clientes potenciales de un producto que la fábrica produce de manera regular, así como a la hora de facilitar su implementación y mantenimiento en el mercado, es importante que si un usuario se ve atraído por su sabor, aroma y textura estos se mantengan a lo largo del tiempo.

Desde el punto de vista industrial, el producto que se está vendiendo deberá de mantener siempre las mismas características, con la intención de que bajo un mismo nombre de una gama de producto, sea prácticamente imposible distinguir si el lote salió hoy de la fábrica o hace un año.

Si se comprueba que esta premisa se cumple, es un indicativo contundente de que el proceso de fabricación implantado es adecuado, pues asegura una repetibilidad que es difícilmente reproducible con una elaboración más manual.

La automatización per se, lleva aparejada varias ventajas:

- Reducción de costes una vez el sistema es productivo.
- Reducción los errores humanos, puesto que los PLC's no se ven afectados por factores psicológicos asociados al comportamiento humano como fatiga, malestar, etc. Ni por factores físicos como contracturas musculares, roturas de huesos, bajas por enfermedad etc. o incapacidad en general.
- Reducción de los tiempos de ciclo de fabricación del producto, al estar la instalación enfocada a ser controlada automáticamente minimiza o elimina la necesidad de intervención humana, con los intervalos no productivos que se puedan asociar a ello.
- Control exhaustivo de los parámetros de fabricación, ya que la ejecución de un programa siempre va a ser la misma. Es mucho más preciso controlar, entre otras magnitudes, tiempos y temperaturas automáticamente que depender del operario que ese día este al cuidado de la máquina; elimina en alto grado el factor error humano.

Como principal desventaja frente a una fabricación artesanal, se encuentra el elevado coste inicial que puede suponer adaptar o construir una instalación para automatizar el proceso productivo en cuestión.

Si una empresa tiene un volumen de negocio suficiente como para plantearse la automatización del proceso, en el mayor número de los casos la eliminación de desventajas justifica la inversión. A continuación se enumeran las principales desventajas asociadas a la fabricación manual que pueden evitarse a través de la automatización:

- Reducción de los tiempos de fabricación.
- Reducción de errores humanos
- Reducción de las necesidades de personal.
- Eliminación de la subjetividad a la hora de evaluar el estado de la salsa en los distintos pasos de fabricación.

Aunque automatizar un proceso resulta gravoso económicamente, el sobrecoste inicial derivado de la modificación y adaptación de la planta al proceso productivo de forma automática se amortiza rápidamente al eliminar la mayor parte de los costes asociados a las desventajas mencionadas.

En todo sistema industrial se tienen tres partes principales y diferenciadas:

- 1.) El proceso productivo, que se pretende controlar.
- 2.) El propio control del sistema, que se encargará de pilotar los diferentes actuadores y de monitorizar el estado de los sensores necesarios para asegurar el funcionamiento dentro de unos parámetros preestablecidos, de modo que el sistema se controle de forma segura y personalizable.
- 3.) El sistema de supervisión por parte del personal a cargo del sistema, el cual servirá para que éste actúe sobre él o pueda visualizar de manera rápida el estado de la máquina. Es habitual encontrar dos denominaciones principales:
 - a. HMI: Human Machine Interface (Interfaz Humano Máquina)
 - b. SCADA: Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos)

Esquemáticamente el control del proceso será:

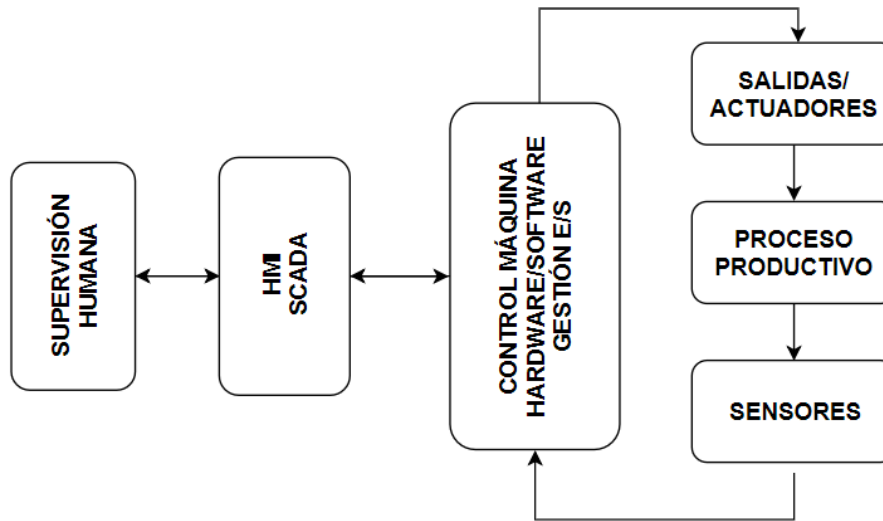


Figura 3 Esquema control proceso industrial

De forma que el personal a cargo de la instalación debe de tener siempre una cantidad de información suficiente para supervisar el correcto desempeño del proceso, y el control necesario para que ante el surgimiento de una disfunción en el sistema o ante una emergencia, se pueda intervenir y detener el proceso de manera segura.

Este proyecto pretende cubrir todos los niveles de automatización y control que se pueden observar en la figura 3.

4.- OBJETIVOS

El objetivo que persigue el desarrollo del presente trabajo de fin de grado es realizar el control y supervisión del proceso de elaboración de la salsa de mostaza.

Para ello se han planteado subobjetivos que en su conjunto permitirán alcanzar la consecución de este objetivo último.

Objetivos Generales:

Uno de los objetivos de este proyecto es el de conseguir que el producto final resultante de la fabricación tenga unas características determinadas que sean reproducibles al detalle lote tras lote, es decir, conseguir repetibilidad en la producción de la salsa a través de la automatización del proceso.

Además, mediante esta automatización, se persigue reducir los tiempos de fabricación, puesto que automatizar implica ahorrar tiempo en la ejecución de tareas repetitivas en las cuales no se aporta un mayor valor añadido si son realizadas manualmente.

Otro objetivo que se pretende conseguir a través de la automatización es el de reducir los errores humanos que se puedan producir en el desempeño de tareas repetitivas que acaban por minar la concentración del operario encargado del proceso productivo, así como evitar los errores que se puedan derivar del estado anímico de este.

También se pretende escoger los equipos necesarios para la gestión de la automatización de la planta, es decir, definir las necesidades del hardware que soportará el programa de control.

Se buscará que el control de esta automatización se realice a través de un sistema SCADA, que contará con un número suficiente de pantallas por las que navegar, para realizar la monitorización de la planta. De este modo el operario a cargo de la fabricación será capaz de supervisar todo el proceso fácilmente, minimizando la posibilidad de errores en la ejecución de la fabricación.

Específicos:

Para conseguir los objetivos generales, se deberán alcanzar unos objetivos más específicos aplicados a la parte del control del proceso.

Uno de los objetivos principales del proyecto será pues, desarrollar un programa de control capaz de automatizar el proceso de la producción de la salsa de mostaza, de modo que la planta sea capaz de trabajar de manera autónoma con la mínima intervención física humana a través de la selección de recetas.

Otro de los objetivos que se persigue, es crear bloques de función que permitan el desarrollo de funciones genéricas aplicables no solo a este proceso concreto, sino que se realicen de manera que posibilite reaprovecharlas en futuros proyectos, reduciendo los tiempos de programación y disminuyendo de este modo los costes.

Para las funciones que se vayan a emplear en más de una ocasión y que necesiten de una cantidad de datos concretos como entradas o salidas, se crearán estructuras de datos específicos que contengan internamente todos los tipos que se necesiten (Bool, Word, Int, etc).

Estas estructuras, formadas por los distintos tipos de datos declarados dentro de ellas, permiten tratarlas como bloques. De este modo solo se declara el tipo de dato una única vez, en lugar de declarar todas las variables internas que contenga la estructura una por una, con el consiguiente ahorro de tiempo.

Igualmente, se perseguirá que el desarrollo de estas funciones se realice de modo que el personal de mantenimiento pueda entender el programa principal de control, pero sin necesidad de acceder a la lógica interna de las funciones. Es decir, crear una caja negra de la cual saldrán las entradas y salidas necesarias para la ejecución del proceso, limitando así la acción del personal de mantenimiento a realizar las conexiones oportunas, en el caso de que fuese necesario.

Para la interfaz programa-operario, se buscará desarrollar un sistema SCADA que sea operable a través de un HMI. Esto permitirá al operario estar informado de los estados de la planta, manipular los actuadores y/o ejecutar las órdenes de producción desde una pantalla.

Desde este HMI se deberá de poder acceder a una pantalla que haga posible la parametrización de la planta, esto permitirá que ante cambios en los componentes que intervienen en el proceso productivo, no sea necesario entrar al programa del PLC para modificar los parámetros afectados por los cambios.

La parametrización de los elementos que conforman la planta, por lo general es fija. Se suele parametrizar una única vez en la puesta en marcha de la instalación y estos parámetros únicamente se cambian ante cambios en los elementos que la componen.

Al no ser una información que requiera modificaciones a diario, se buscará crear un control de usuarios de modo que únicamente el personal autorizado sea capaz de modificar estos valores en caso necesario, imposibilitando así que se modifiquen los parámetros por error o desconocimiento.

Para evitar errores en la parametrización, así como para alertar al operador de planta de los posibles fallos que puedan deberse a malfuncionamientos en los componentes de la planta y para mostrar los avisos que puedan ser necesarios a la hora de gestionar los recursos del sistema de producción se implementará un sistema de gestión de alarmas y además se llevará un registro histórico que permitirá realizar un análisis pormenorizado de las alarmas producidas.

Esto permitirá determinar las averías más comunes, cuantificar los tiempos de parada asociados a cada avería así como poder detectar posibles puntos de mejora con el fin de reducir el número de averías y, en definitiva, el tiempo de parada de producción.

En toda industria manufacturera se realizan ensayos con el objetivo de comprobar si los experimentos diseñados con intención de modificar o alterar, bien la formulación de una receta, bien para observar cómo afecta un tratamiento térmico distinto a las cualidades del producto, en definitiva como afecta la variación de cualquier magnitud monitorizable, suponen una mejora o se alcanza el resultado esperado.

Con el objeto de poder desarrollar nuevas versiones y variedades de la salsa, se dotará al sistema de la posibilidad de realizar un control manual del mismo, que permita ir realizando todos los pasos necesarios para la fabricación de la salsa. En este caso será el operario el encargado de introducir los pesos, proporciones, tiempos o temperaturas y será él el responsable último de los movimientos o acciones que ejecute la planta.

Por el contrario, para mezclas ya definidas y testadas, se buscará desarrollar un modo automático. Para su implementación, se dotará al sistema con una gestión de recetas, en las cuales vendrán predefinidas las cantidades, tiempos, etc. Facilitando así la labor del operario de planta al tener únicamente que seleccionar la receta que desea fabricar, y tras cargarla, esta se ejecutará de modo autónomo a lo largo de todo el proceso de producción hasta dar un aviso al finalizar.

5.-DESCRIPCIÓN DEL PROCESO

El proyecto pretende controlar un sistema de dosificación, pesaje industrial y control de temperatura para la fabricación de la salsa de mostaza.

Para ello, se necesitará:

- Sistema para el almacenaje y conservación de las materias primas.
- Sistema encargado de realizar el pesaje: Un elemento con el que se puedan controlar las cantidades depositadas de cada material, en este caso una báscula.
- Sistema de discriminación: Si la mezcla ha resultado correcta se pasará a otra zona donde se trabajará con ella. Si la mezcla ha resultado defectuosa, se enviará desecho o reciclaje.
- Sistema de mezclado que permita homogeneizar los ingredientes vertidos en el tanque de mezclado.
- Sistema de triturado y cocido, donde el producto adquiere una textura o tamaños característicos a través del triturado a la vez que el producto se lleva a una temperatura determinada durante un tiempo controlado.
- Capacidad de dosificación: se ha planteado que se consiga mediante el empleo de dos actuadores, uno de gran capacidad y otro de capacidad más reducida para tener mayor control sobre la cantidad vertida. De esta forma se cuenta con dos modos dentro del propio dosificado:
 - Dosificado rápido: En esta primera fase, con los dos actuadores en marcha, se vierte gran parte de la materia prima sin un control excesivamente preciso; se hace una dosificación aproximada hasta alcanzar un porcentaje cercano al peso consigna.
 - Dosificado lento: O ajuste fino; Al llegar al porcentaje anterior, por ejemplo un 75%, se desactivará el actuador de mayor tamaño y únicamente quedará vertiendo material el más pequeño. De esta forma se añadirá material hasta que se alcance el peso consigna establecido de manera más lenta, pero más precisa.

Para conseguir una mayor precisión en las cantidades de material vertido, se controlará cuando se desactivan los actuadores. Si se dispusiera de un solo actuador de vertido grande, las velocidades de deposición de material serían elevadas, pero en concordancia a su tamaño, al tener una gran inercia la precisión en el dosificado se reduce notablemente.

Para solucionar este problema, se emplearán dos actuadores, uno de mayor diámetro y por tanto de mayor capacidad, y otro de menores dimensiones, que es el que dotará al sistema de vertido de la precisión requerida para las proporciones de la mezcla.

En la figura 4, se ha representado la evolución que tendrá el vertido de peso en función de qué actuadores se tengan encendidos. Para conseguir el grado de precisión buscado, primero se activarán simultáneamente los dos actuadores, y al estar cerca del peso de consigna únicamente permanecerá encendido el actuador de menor capacidad, permitiendo llegar al peso consigna con un margen de error muy reducido.

La primera pendiente de la línea poligonal "Combinación motores" representa la zona en la que los dos actuadores están encendidos al mismo tiempo, y el último tramo coincidiendo con el cambio de pendiente, representa la zona en la cual únicamente queda activo el actuador de menor tamaño.

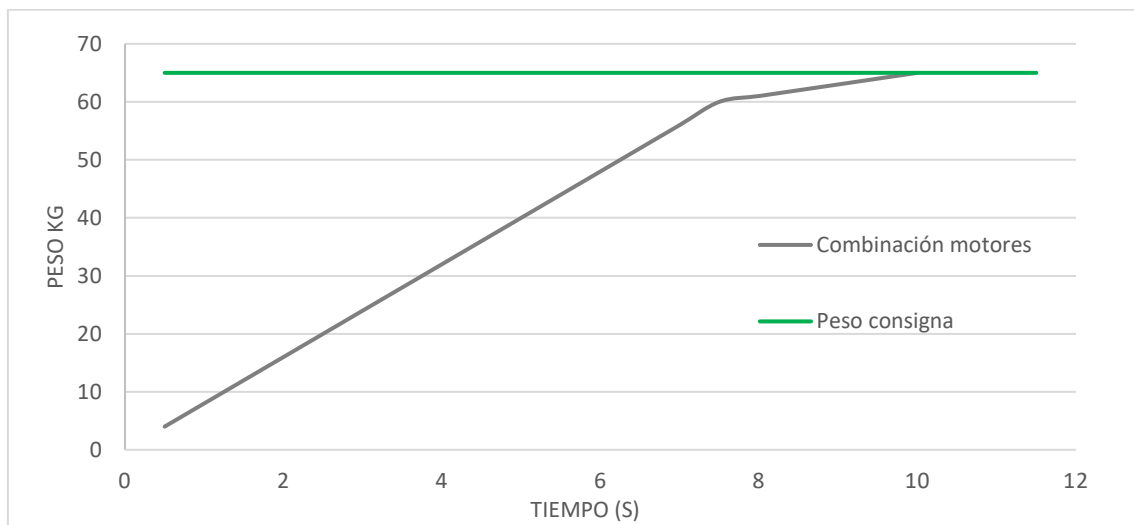


Figura 4 Grafico comparativo activación actuadores vs tiempo de carga

En la figura 5 se ha representado el esquema del proceso completo:

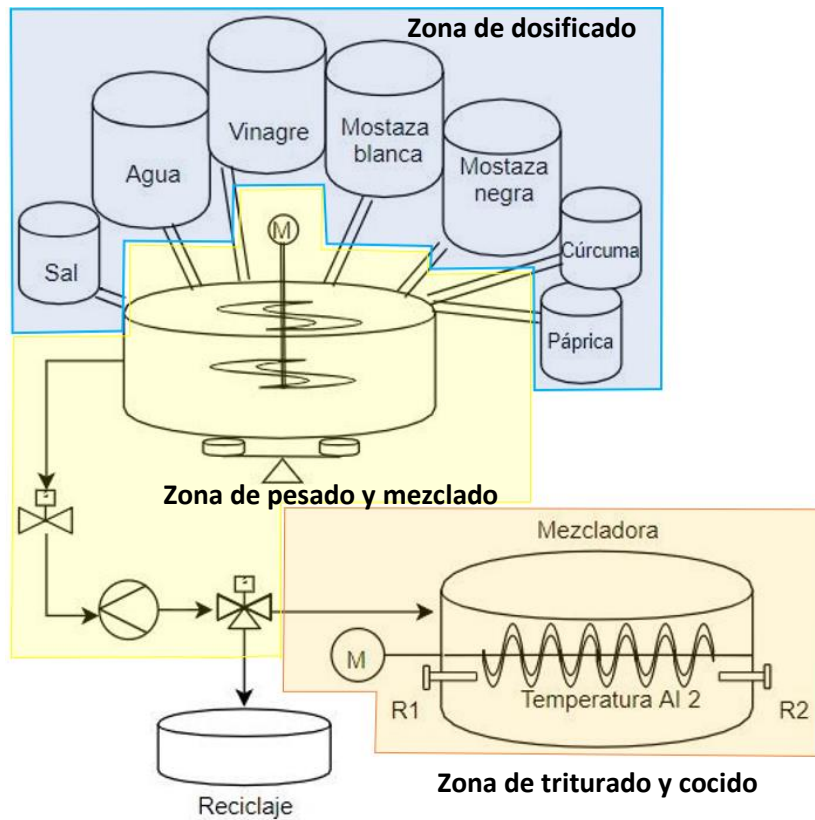


Figura 5 Esquema de la instalación

En el esquema de la figura 5 se puede encontrar:

- Siete silos, para el almacenaje del material.
- Actuadores, tales como motores, resistencias de calentamiento, electroválvulas etc.
- Una báscula.
- Un tanque de mezclado.
- Un tanque de material desechado.
- Un termómetro.
- Un tanque de triturado.

5.1 ZONA DE DOSIFICADO (AZUL):

Dentro de cada silo habrá almacenada una materia prima diferente lista para ser enviada al tanque de mezclado, en el cual se realizará el pesaje. Estos silos contendrán los ingredientes necesarios para la elaboración de la salsa: agua, vinagre, mostaza blanca, mostaza negra, cúrcuma, paprika y sal.

Los porcentajes de la mezcla, aunque se variarán en funcion de la receta, rondan valores cercanos a los mostrados a continuacion:

- 60% agua
- 20% vinagre
- 15% mostaza
- 5% sal y especias.

Como la fabricacion de la salsa requiere de diferentes proporciones de materias primas, para los ingredientes de los que se necesitan mayores cantidades (agua, vinagre y las mostazas) se dispondra de silos de mayor tamano y por el contrario, los materiales que se necesitan en menor proporcion (la sal y las especias) se almacenaran en silos de menores dimensiones, permitiendo evitar tener grandes cantidades en stock.

En base a su tamano, para los silos grandes se emplearan dos actuadores; uno para el ajuste basto y otro para el ajuste fino, que se activaran a la vez, y al acercarse al peso consigna quedara encendido unicamente el de menor tamano ganando precision en el control sobre el peso vertido, conforme a lo expuesto respecto a la figura 4. Con esto se conseguira, ademas de la mencionada mejora de la precision del peso vertido, reducir tiempos de ciclo, lo que permite mayor cantidad de pesajes por hora, que se traducen en una mayor productividad.

Por otra parte, en los silos pequenos solamente se empleara un actuador de ajuste fino, puesto que como las cantidades a verter no son excesivamente grandes, el factor velocidad conseguido en el vertido es suficientemente satisfactorio con un solo actuador.

Para mejorar la precision del vertido se tendra en cuenta la "cola de carga", que sera la cantidad de material que sigue cayendo a la bascula desde que se da la orden de paro a los actuadores, hasta que la parada del vertido de material se hace efectiva. Esta cola de carga sera un dato empirico que formara parte de los parametros de la planta.

A la hora de controlar la cantidad de material vertido por cada silo, y conocer en todo momento el peso de cada ingrediente que se tiene en el tanque, se realizara una comparacion del peso consigna con el peso marcado por la bascula.

La báscula únicamente será capaz de entregar una señal, por lo que para realizar correctamente la comparación el programa deberá de ser capaz de discriminar el material que se está vertiendo, e indicar su peso relativo a través de la diferencia entre la lectura del peso que se tenía al iniciar el vertido con la lectura del peso que se tenga cuando se deje de verter.

Se va a prescindir de los sensores de nivel para el control de la cantidad de material restante en los silos. Con esto se pretende contribuir a la reducción del impacto económico en la implantación de la automatización del proyecto, ya que los sensores de nivel por ultrasonidos o láser tienen unos costes de adquisición elevados.

No se ha optado por montar sensores con un tipo de señal conmutada debido a que se pierde precisión en la medida de la cantidad restante en el silo. Es decir, con estos sensores se podría crear una escala de medida discreta pero para obtener una resolución aceptable se requeriría de una cantidad elevada de sensores.

A mayor cantidad de sensores mayor incremento en el coste, lo que puede derivar en que la diferencia final del precio de todos los sensores conmutados con los sensores analógicos se reduzca hasta el punto de no ser viable. Además del coste de la sensorería, hay que tener en cuenta que se necesitarían tantas entradas al PLC como saltos discretos se pretendiera que tuviera la escala de medida del nivel.

Para controlar la cantidad de material que se encuentra almacenado en los silos se ha partido de las siguientes premisas:

- Se introducirá al sistema cuantos kg es capaz de almacenar el silo a máxima capacidad.
- Cada vez que se rellene el silo se hará hasta su máximo.
- Se programarán dos avisos, aviso de nivel bajo, idealmente cuando el silo este cerca del 25% de su capacidad, y aviso de silo vacío o fallo de nivel, si bien los valores serán personalizables. Estos avisos pretenden evitar que los niveles de materias primas en la planta nunca estén por debajo de valores que puedan suponer una rotura de stock que afecte a la planificación de la producción.
- Aunque las cantidades restantes de material se expresarán de manera gráfica en el HMI, si alguno de los avisos se activa, se informará a través de una ventana emergente al operario, para que no pase por alto que el nivel de material es bajo y se pueda planificar así un contacto con el proveedor de la materia prima en cuestión.
- El operario que se encargue de rellenar el silo deberá de accionar un pulsador que hará de *reset* de las medidas de la carga del silo, lo que hará que el programa cargue de nuevo la capacidad máxima del silo.

5.2 ZONA DE PESADO Y MEZCLADO (AMARILLO):

El propio tanque de mezclado descansa sobre una báscula. Los distintos materiales se irán añadiendo necesariamente por orden de uno en uno, para permitir realizar a través del programa la distinción de los distintos pesos de los productos que componen la salsa.

A través de la señal analógica que ofrece la báscula el sistema ofrecerá la lectura de dos pesos:

- Uno correspondiente al peso individualizado de cada material.
- Uno correspondiente a la lectura del peso total del conjunto de materias primas que se han vertido al tanque de mezclado.

Una vez terminado el vertido de los componentes de la receta, se procederá al mezclado. Para esta fase del proceso, se activará un motor que arrastrará una serie de paletas que se encargarán de homogeneizar la mezcla durante un tiempo determinado.

El tiempo de mezcla dependerá de la receta, y una vez finalizado se procederá al trasvase del producto al tanque de triturado mediante una bomba.

Para controlar el inicio del trasvasado se dispone una electroválvula de dos posiciones pilotada por un solenoide. Una vez se haya terminado el tiempo de mezclado, se abrirá la citada electroválvula, y seguidamente se activará la bomba.

La activación de la bomba tendrá un ligero retraso respecto a la apertura de la electroválvula para permitir que el material llegue hasta la misma y evitar así su funcionamiento en vacío, lo que podría ocasionarle daños o una reducción de la vida útil.

Si la mezcla ha resultado correcta una segunda electroválvula permanecerá en estado de reposo, permitiendo pasar la mezcla hacia el tanque de triturado.

En caso contrario, si el pesado ha resultado incorrecto, esta válvula se activará para desviar el material a un tanque de desecho o reciclado.

Para dar por finalizada la etapa de trasvasado, se procederá a la desconexión de la bomba cuando se detecte que la lectura del peso total marcada por la báscula sea de 0 kg.

5.3 ZONA DE TRITURADO Y COCIDO (NARANJA):

Cuando el trasvase finaliza el tanque de triturado está lleno, con lo que se activarán los motores del triturador.

Estos motores permanecerán un tiempo programado encendidos. El triturado servirá para reducir al tamaño deseado los sólidos incorporados a la mezcla.

Para el cocido, se dispondrá de una serie de resistencias calefactoras a través de las cuales se podrá controlar la temperatura a la que se cocerá la mezcla mientras se encuentra en el triturador.

La temperatura será programable, al sistema se le indicará un valor de consigna y a través de un control por histéresis se mantendrá la mezcla en torno a la temperatura deseada.

Una vez el tiempo de triturado se agote, el proceso habrá llegado a su fin y el producto estará terminado y disponible para el envasado.

6.- MODOS DE FUNCIONAMIENTO

Una vez se tiene conocimiento de las zonas que componen la instalación, a continuación se describen los modos de funcionamiento.

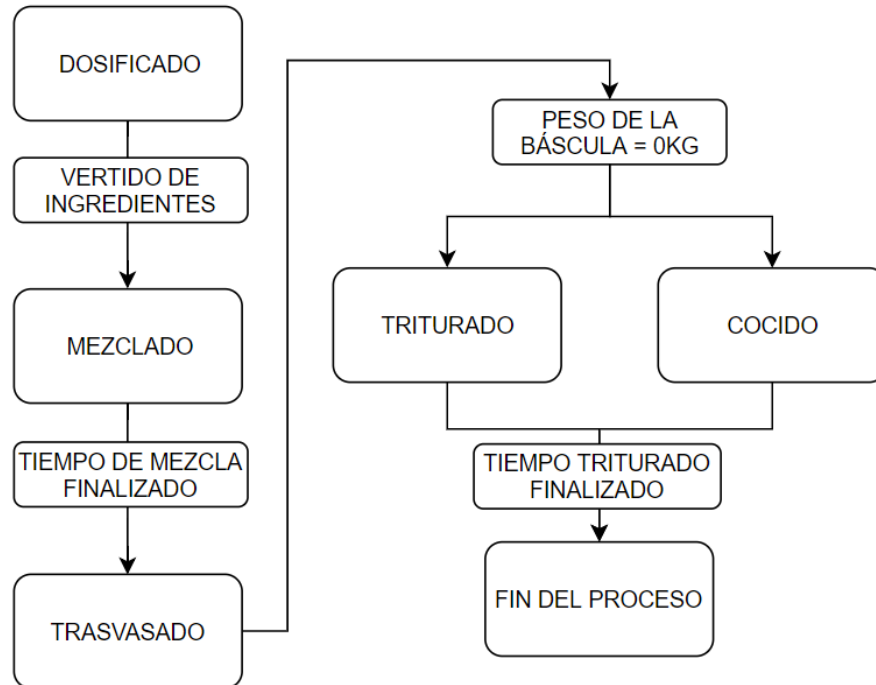


Figura 6 Diagrama funcionamiento general

Siguiendo el flujograma representado en la figura 6 se ha optado por la creación de dos modos de trabajo:

- El **modo manual**, donde el operario será capaz de ir recorriendo todos los pasos mostrados por el flujograma controlando él mismo todas las fases del proceso completo.
- El **modo automático**, que recorrerá todos los bloques del flujograma de forma automática.

6.1 MODO MANUAL

En el diagrama de bloques de la figura 7, se muestra el funcionamiento previsto del modo manual.

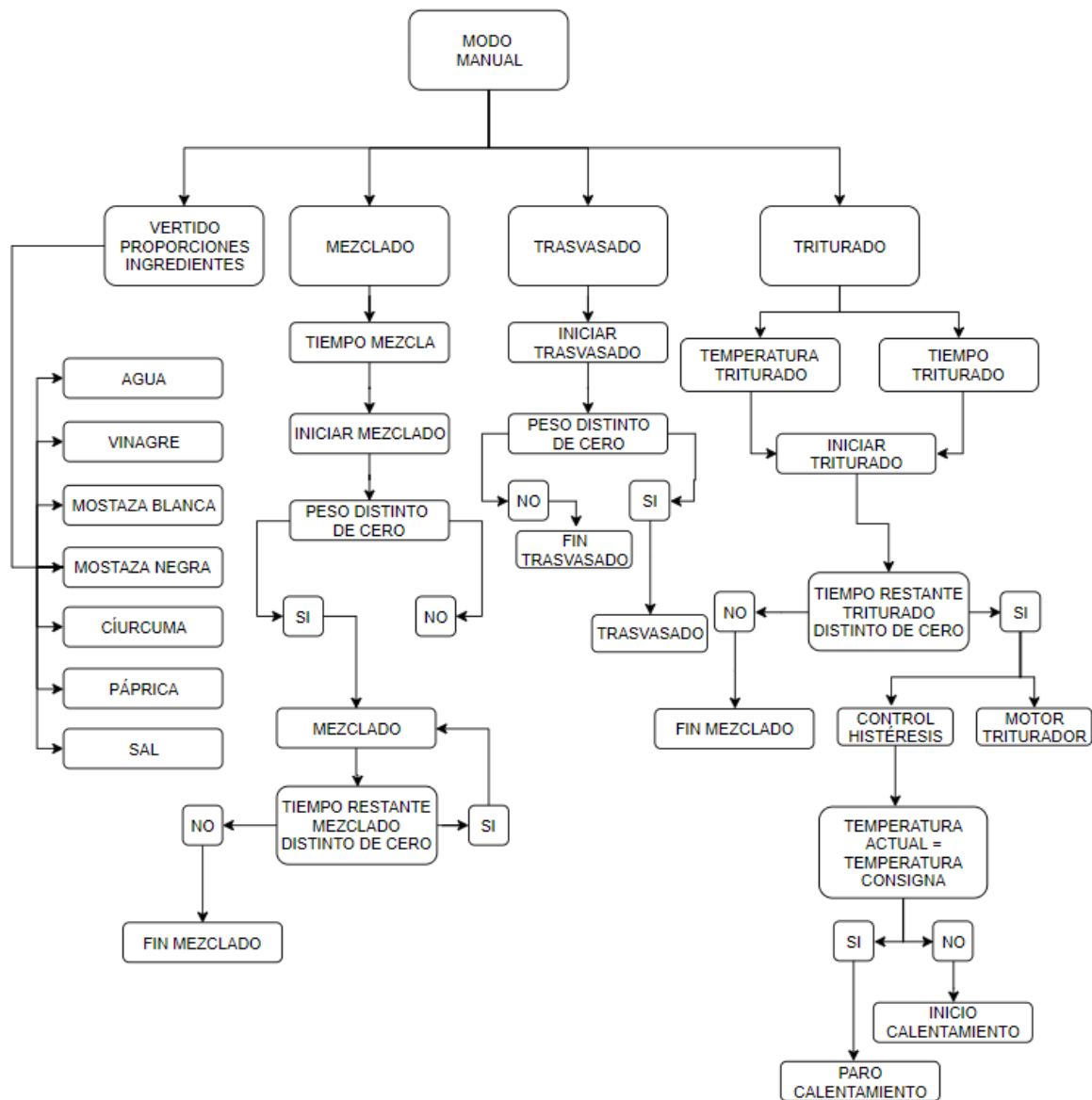


Figura 7 Esquema de funcionamiento del modo manual

Si al encender la máquina se selecciona el modo manual el operario deberá ir agregando los ingredientes uno a uno a través de la activación de los distintos actuadores. Una vez contento con el peso, se podrá encender el mezclador.

Para efectuar la fase de mezclado previamente se habrá introducido el tiempo de mezcla deseado. Al agotarse el tiempo de mezcla, se parará.

Una vez la mezcla preparada y a la espera en el tanque de mezclado, al pulsar un botón de "inicio trasvase", se activará la EV del fondo del mezclador permitiendo el paso hacia la bomba impulsora. La bomba impulsora se conectará unos segundos después de abrir la EV, para que no funcione en vacío y evitar así futuros problemas de funcionamiento. La bomba se detendrá cuando el peso de la báscula esté a cero, y en pantalla un piloto informará de que el proceso de trasvase ha finalizado.

Por último, para la fase de triturado, se deberá de introducir tanto el tiempo como la temperatura a la que debe de cocerse la mezcla. Al darle a la marcha, se activarán las resistencias y el motor del triturador. Las resistencias irán apagándose y encendiéndose en función de las órdenes de un control por histéresis. Al finalizar el tiempo de la fase de triturado se detendrá tanto el motor como las resistencias calefactoras.

6.2 MODO AUTOMÁTICO

En el diagrama de bloques de la figura 8, se muestra el funcionamiento previsto del modo automático:

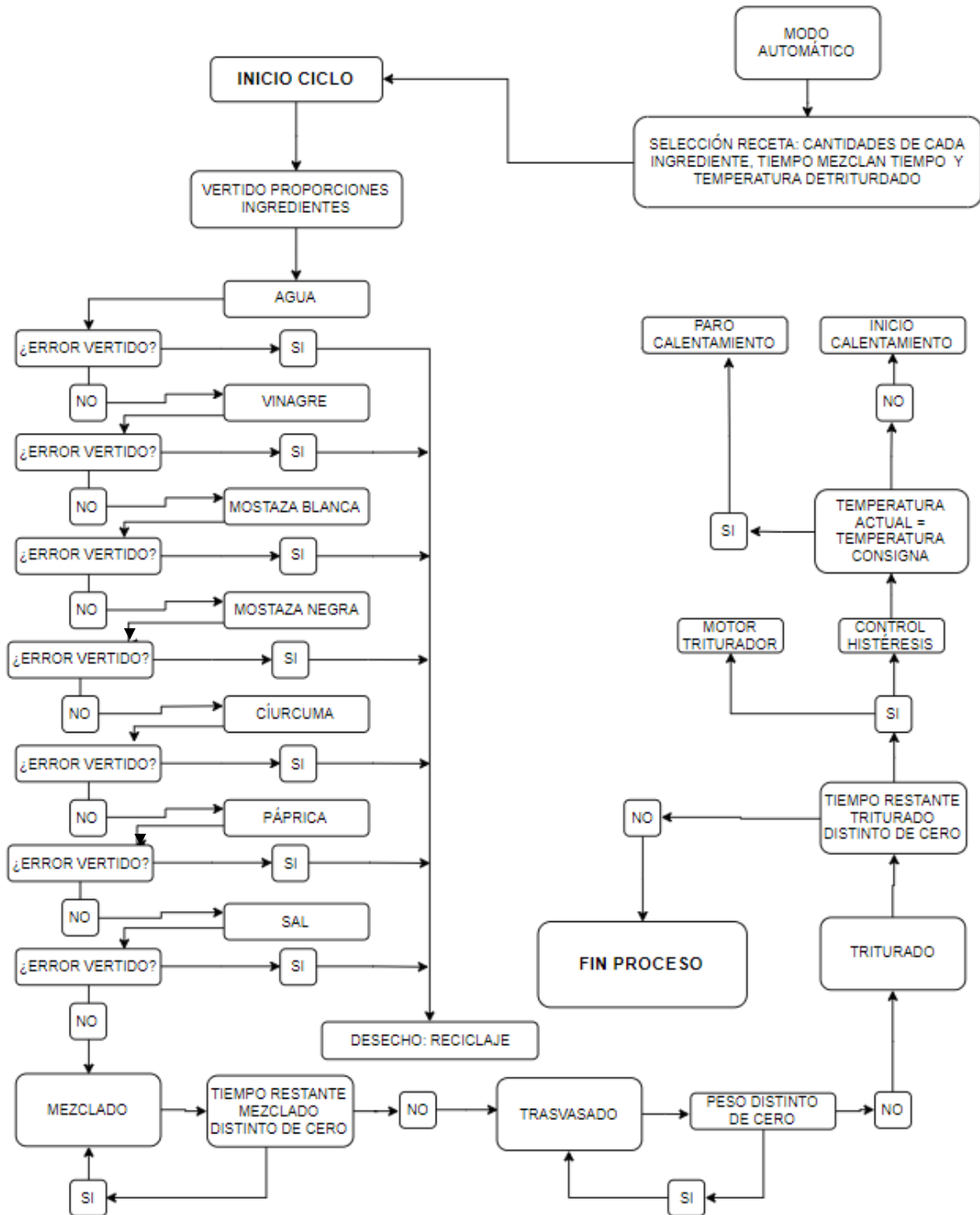


Figura 8 Esquema funcionamiento del modo automático

En el **modo automático** el primer paso será introducir la receta de la salsa, que contendrá las cantidades de cada materia prima, el tiempo de mezcla, el tiempo de triturado y la temperatura de consigna del triturador. Una vez cargada en el sistema, se iniciará necesariamente por orden, el vertido de los distintos materiales uno detrás de otro puesto que solo se dispone de una báscula. Primero el agua, luego el vinagre, las mostazas, las especias y por último la sal.

Una vez todo el material se ha vertido correctamente en el tanque, se activará el motor que accionará el mezclador. Este mezclador permanecerá encendido tanto tiempo como se le haya establecido en la receta.

Transcurrido este tiempo, con la mezcla ya homogeneizada, se abrirá una válvula de vaciado, que permitirá el paso de material hacia una bomba impulsora que se encargará de transportar el material hasta el triturador.

Una vez se tenga todo el material en el triturador, se encenderá el motor que moverá las piedras/aspas de interiores. A su vez, se mantendrá la temperatura constante a la consigna de la receta, a través de unas resistencias de calentamiento o manta calefactora.

El triturado, al igual que el mezclado, durará un tiempo programable en función de la receta. Cuando este finalice, el programa dará una señal para que los operadores de la planta tengan conocimiento de que el producto final está listo para el envasado o distribución a granel.

7.- SOLUCIÓN ADOPTADA PARA EL CONTROL DEL PROCESO

7.1 NECESIDADES A CUBRIR POR EL HARDWARE

Para controlar todos los elementos de la planta, se necesitará:

Electroválvulas pilotadas a 24 VDC, en total seis:

- Cuatro que se encargarán de controlar el vertido de las materias primas líquidas, el agua y el vinagre.
- Una encargada de controlar el trasvasado de material desde la zona de mezclado hacia la zona de triturado.
- Una encargada de desviar el material hacia el tanque de reciclado siempre y cuando se haya producido un error en el vertido.

Contactores pilotados a 24 VDC encargados del accionamiento de los actuadores, en total once:

- Siete contactores que se encargarán de controlar la activación de los motores encargados de mover los tornillos sinfín para el vertido de las materias primas sólidas; Dos para la mostaza negra (finos y gruesos), dos para la mostaza blanca uno para la cúrcuma, uno para la páprika y finalmente uno para la sal.
- Un contactor para el control del inicio del movimiento del motor del mezclador.
- Un contactor encargado del control de la activación de la bomba de trasvasado.
- Un contactor para el control del inicio del movimiento del motor del triturador.
- Un contactor para activar las resistencias de calentamiento.

Un termómetro para el control de la temperatura de cocido en la estación del triturado.

Una báscula para el control del peso vertido en el tanque de mezclado.

Descripción	Tipo	Cantidad
Contactores	DO	11
Electroválvulas	DO	6
Báscula para el pesaje	AI	1
Termómetro para el control del cocido	AI	1
Pulsador de marcha	DI	1
Pulsador de paro	DI	1

Tabla 1 Necesidades de hardware para el control

Para la parte del control del proceso se empleará un PLC (autómata programable o Programmable Logic Controller) para el control de las entradas y salidas. Para la monitorización del estado, el PLC se encontrará conectado a un ordenador en el cual se implementará un sistema SCADA desde el que el usuario podrá visualizar y ajustar las variables del sistema.

La conexión se realizará por Profinet que es la que se tiene disponible en el aula de accionamientos, y además es ampliamente utilizada a escala industrial.

Para el desarrollo del programa, se necesitará un entorno de programación que vendrá definido por el hardware que finalmente se opte por emplear en el proceso.

7.2 ESTUDIO DE ALTERNATIVAS DEL HARDWARE

Lo primero llegado este punto, es definir que hardware se va a emplear para la automatización del proceso. Se van a comparar tres autómatas programables, uno de la marca Allen Bradley, otro de la marca Schneider y otro de la marca Siemens.

Allen Bradley Micro820:

- Entradas integradas: 12 DI 24 V DC
- Salidas integradas 7 DO, tipo relé hasta 2 A
- 4 entradas analógicas AI, 0-10V DC
- Alimentación: AC 120-240 V
- Memoria de programas/datos 20 KB

Precio aproximado según catálogo web: 375 €



Figura 9 PLC Micro820 Allen Bradley

Schneider Electric ModiconM421:

- Entradas integradas: 14 DI 24 V DC
- Salidas integradas 6 DO, tipo relé hasta 2 A
- Salidas integradas 4 DO, tipo transistor hasta 0.5A
- Alimentación: AC 100-240 V
- Memoria de programas/datos 8 MB

Precio aproximado según catálogo web: 325 €



Figura 10 PLC Electric ModiconM421 Schenider Electric

Siemens S7-1214C:

- Entradas integradas: 14 DI 24 V DC
- Salidas integradas 10 DO, tipo relé hasta 2 A
- 2 entradas analógicas AI, 0-10V DC
- Alimentación: AC 85-264 V AC con 47-63 Hz
- Memoria de programas/datos 100 KB

Precio aproximado según catálogo web: 295 €



Figura 11 PLC Siemens S7-1200

Ninguno de los tres PLC's dispone de un número suficiente de entradas o salidas para gestionar todos los actuadores, por lo que se hace necesario el empleo de módulos de ampliación:

- Módulo de entradas/salida Allen Bradley Guardmaster 2080-IQ4OB4, 4 entradas, 4 salidas, 24 V dc

Precio aproximado según catálogo web: 115 €

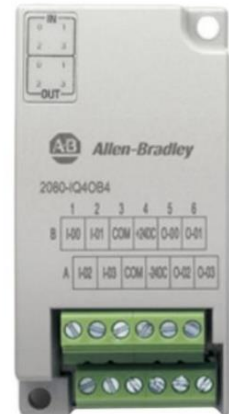


Figura 12 PLC Micro820 Allen Bradley

- Módulo de entrada analógica TMC4HOIS01

Precio aproximado según catálogo web: 135 €

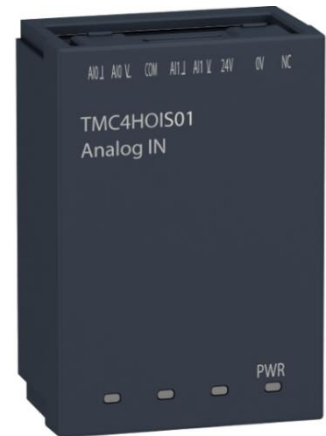


Figura 13 Ampliación TMC4HOIS01 1 AI

- Módulo SM 1223: Entradas adicionales: 16 DI 24 V DC Salidas adicionales 16 DO, tipo transistor hasta 0,5 A Alimentación: DC 24V

Precio aproximado según catálogo web: 215 €



Figura 14 Ampliación pasa S7-1200, SM1223

La combinación de la marca Allen Bradley es la menos económica de las opciones comparadas, pues haría falta comprar al menos dos módulos de ampliación para poder controlar todos los actuadores, con lo que el montante final ascendería a una suma cercana a los 600 €.

La combinación de la marca Schneider Electric tampoco satisface las necesidades del proyecto, puesto que con los módulos de ampliación, no es posible establecer una configuración tal que se puedan controlar todos los actuadores necesarios para la automatización de este proceso, por lo que se descarta este proveedor.

La opción de la marca Siemens, además de ser la que mejor se adapta a las necesidades del proyecto en cuanto a modularidad y necesidades de entradas y salidas, económicamente es la más ventajosa puesto que la valoración del hardware escogido ronda los 510 €.

Por lo expuesto anteriormente, **se ha decidido emplear la opción de la marca Siemens.**

7.3 NECESIDADES A CUBRIR POR EL SOFTWARE

Dado que la marca escogida para el hardware es Siemens, el software empleado va a ser el TIA Portal (Totally Integrated Automation). Este software es el entorno específico de SIEMENS para crear los programas de control que se descargarán a la familia de PLC's S7-1500 y S7-1200, como es el caso que se plantea.

La parte gráfica es el medio interactivo que tendrá el operario para comunicarse con el programa. El desarrollo de una interfaz intuitiva contribuirá a reducir los tiempos de aprendizaje del manejo del programa además de disminuir la probabilidad de errores asociados a un empleo inadecuado.

Siemens ofrece diversidad de pantallas, desde paneles básicos monocromáticos a pantallas táctiles de 19". Para el caso que se está tratando, se ha escogido una tercera vía. Siemens ofrece la posibilidad de controlar todo lo relacionado con sus PLC's a través de un sistema PC genérico. Esto quiere decir que instalando su software en un ordenador común se podrá utilizar un mismo PC para el control HMI y para otras tareas, aunque es preferible que se destine únicamente al control de la planta si el proceso de fabricación está activo.

El software necesario para la programación del SCADA es el Windows Control Center, conocido más comúnmente como WinCC. Este sistema viene integrado dentro del paquete de TIA Portal, permitiendo emplear un único entorno para realizar la programación de los autómatas y la parte de la visualización de las pantallas, a través de las cuales se interactuará con el programa de control.

A través del empleo del WinCC, se pretende:

- Manejar la instalación desde un SCADA personalizado para cada tipo de aplicación.
- Visualizar gráficamente los diferentes estados en los que se encuentra el proceso.
- Señalizar alarmas o eventos a través de tablas de alarmas personalizables.
- Crear y gestionar recetas, sin necesidad de tener que programarlas en la parte del PLC.
- Crear una administración de usuarios para proteger pantallas cuyos valores se consideren críticos, para evitar las modificaciones sin consentimiento.

Para instalar el sistema SCADA se necesitará un ordenador capaz de soportar los requisitos del software expuestos en la figura 15:

		Mínimo	Recomendado
CPU	Windows 7 / Windows 8.1 (32 bits)	CPU Dual Core Cliente / sistema monopuesto: 2,5 GHz	CPU Multi Core Cliente: 3 GHz Sistema monopuesto: 3,5 GHz
	Windows 7 / Windows 8.1 / Windows 10 (64 bits)	CPU Dual Core Cliente / sistema monopuesto: 2,5 GHz	CPU Multi Core Cliente: 3 GHz Sistema monopuesto: 3,5 GHz
	Windows Server 2008 R2 / Windows Server 2012 R2	CPU Dual Core Cliente / sistema monopuesto / servidor: 2,5 GHz	CPU Multi Core Sistema monopuesto: 3,5 GHz Servidor: 3,5 GHz
Memoria de trabajo	Windows 7 / Windows 8.1 (32 bits)	Cliente: 1 GB Sistema monopuesto: 2 GB	Cliente: 2 GB Sistema monopuesto: 3 GB
	Windows 7 / Windows 8.1 / Windows 10 (64 bits)	Cliente: 2 GB Sistema monopuesto: 4 GB	4 GB
	Windows Server 2008 R2 / Windows Server 2012 R2	4 GB	8 GB
Espacio de memoria libre en el disco duro - para la instalación de WinCC - para trabajar con WinCC ^{1) 2)}		Instalación: <ul style="list-style-type: none"> • Cliente: 1,5 GB • Servidor: > 1,5 GB Trabajar con WinCC: <ul style="list-style-type: none"> • Cliente: 1,5 GB • Servidor: 2 GB 	Instalación: <ul style="list-style-type: none"> • Cliente: > 1,5 GB • Servidor: 2 GB Trabajar con WinCC: <ul style="list-style-type: none"> • Cliente: > 1,5 GB • Servidor: 10 GB Las bases de datos de ficheros puede necesitar capacidad de memoria adicional.
Memoria de trabajo virtual ³⁾		Memoria de trabajo 1,5 veces mayor	Memoria de trabajo 1,5 veces mayor
Intensidad cromática/calidad cromática		256	Máxima (32 bits)
Resolución		800 * 600	1920 * 1080 (Full HD)

Figura 15 Requisitos hardware para el soporte del WinCC

El ordenador del que dispondrá el cliente es un HP Pavilion 690-0302ns, cuyas características cumplen sobradamente los requisitos para la instalación y empleo del WinCC.

La ficha técnica de este ordenador se puede consultar en el [anexo VI](#).

7.4 PROGRAMA DE CONTROL

Para el control de la instalación, se ha dividido el programa en varios bloques de organización y funciones de manufactura propia que se explicarán más adelante en este documento. La figura 16 muestra el árbol del proyecto completo con todos los elementos que se han empleado en el desarrollo del programa:



Figura 16 Vista completa árbol del proyecto

El orden de los bloques que se describen en los siguientes apartados, ha pretendido mostrar las funciones programadas, así como los OB's creados para la gestión del programa de control.

Para la elaboración del programa se han empleado distintos lenguajes de programación.

Se ha utilizado el lenguaje Ladder para las funciones y bloques de organización principales.

El personal de mantenimiento de las fábricas suele tener menos problemas para entender un programa siempre que la estructura de este se asemeje a la lógica de contactos que pudiera hacerse con contactores y relés en unos esquemas eléctricos.

Dado que no es habitual que este tipo de personal tenga conocimientos específicos de lenguaje C o similares, se ha optado por el lenguaje en escalera para facilitar el seguimiento del programa en caso de necesidad.

Por otra parte, para las funciones a las que se les requiere cálculos matemáticos o instrucciones con algoritmos complejos, se ha escogido el tipo de lenguaje SCL.

El SCL es un lenguaje de programación de alto nivel, cuyas estructuras son similares al lenguaje C o al Pascal, haciendo más flexible la programación al permitir incluir en la lógica, funciones condicionales entre otras. Además, simplifica la tarea de la creación de algoritmos a través de varias instrucciones, que programadas en Ladder serían muy farragosas.

Este tipo de lenguaje se ha empleado para crear algunas de las funciones de manufactura propia, principalmente las que se iban a necesitar más de una vez. De este modo se consigue que la programación se realice una única vez permitiendo ahorrar tiempo de programación así como mantener la estructura del programa más ordenada.

Estos bloques de función se guardarán dentro de librerías, de modo que queden disponibles para utilizarlas para el desarrollo de futuros proyectos y no tener que volver a generarlas de nuevo. Esto permite emplearlas de manera directa y tantas veces como sea necesario con el consiguiente ahorro de tiempo.

Es decir, al crear una función con el código escrito y luego hacer uso de ella en el bloque de organización principal, aparecerá un recuadro con el nombre que se le haya asignado. Estos bloques no tienen restricciones en el número de veces que se pueden emplear.

De este bloque saldrán una serie puntos de conexión con las entradas y salidas que se le hayan definido en el código interno. Para el personal de mantenimiento de la instalación será un bloque cerrado del que conocerán el número de entradas y salidas, así como el modo adecuado de conexión para conseguir realizar una tarea determinada.

7.4.1 BLOQUE ORGANIZACIÓN PRINCIPAL MAIN [OB1]

Para el desarrollo del programa es necesario crear bloques de organización (OB's) que son bloques que ejecutan de forma cíclica el programa.

Los bloques de organización permiten estructurar el programa. Para ello, hay diferentes tipos de OB's que se ejecutan conforme al nivel de prioridad que SIEMENS les ha establecido, por lo que, primero se ejecutan los de nivel de prioridad más bajo.

Aunque pueda parecer contraproducente, se ejecutan por este orden porque los menos prioritarios pueden ser interrumpidos por otros tipos de OB's necesarios para el procesamiento del programa.

La figura 17 muestra los distintos tipos de bloques de organización que se pueden implementar en un autómatas S7-1200 de SIEMENS:

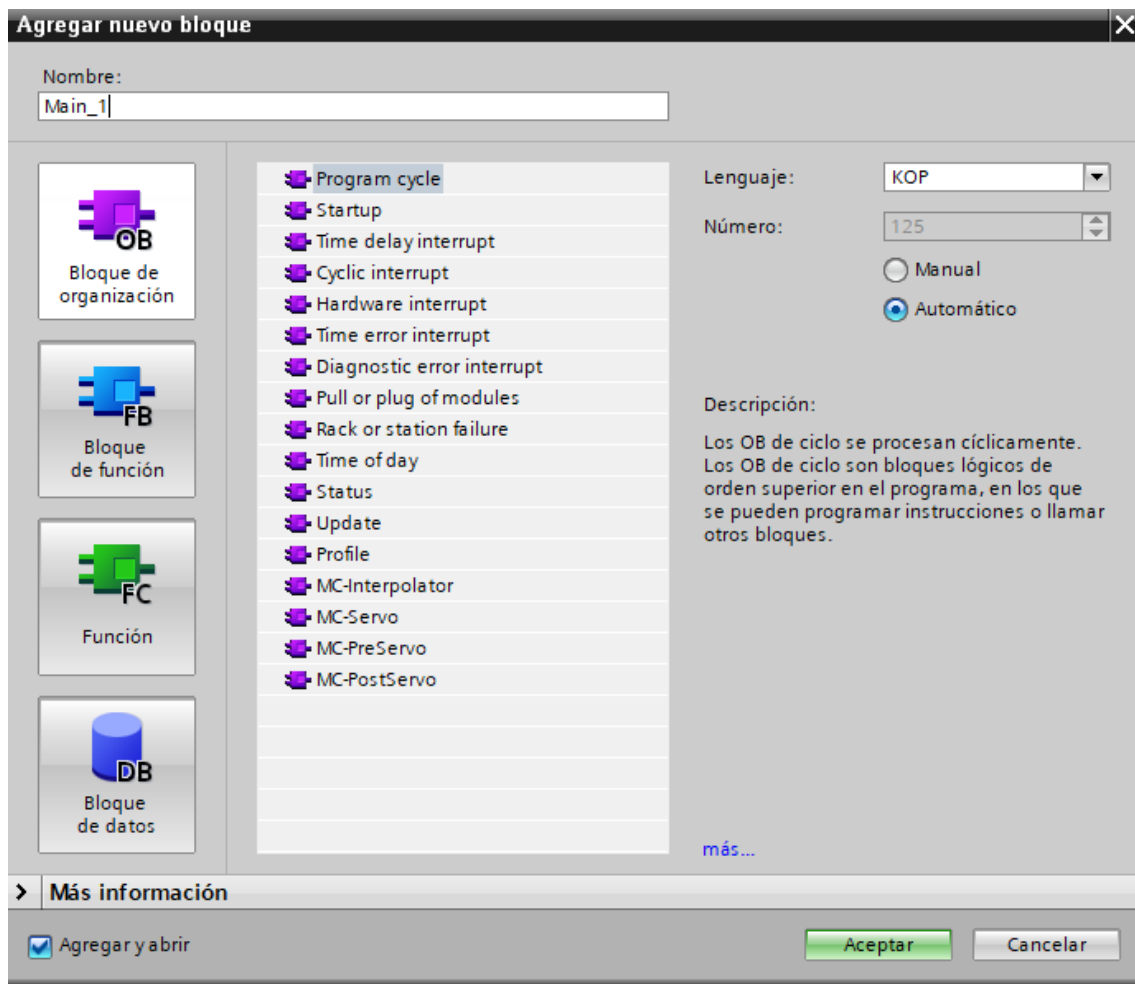


Figura 17 Vista de los distintos tipos de bloques de organización

La tabla 2 se explican los principales tipos de OB's que se puede emplear:

Bloques de organización para el S7-1200:
Bloque de organización para la elaboración cíclica del programa (S7-1200/1500)
Bloque de código superior que se tienen que elaborar cíclicamente en el programa y en los que se programan instrucciones o pueden llamar a otros bloques. El OB1 es el bloque estándar para la ejecución cíclica del programa de usuario.
Clase de evento: Ciclo del programa, nombre: Main (OB1)
Bloque de organización para el arranque (S7-1200)
El OB de arranque sólo se elaboran una vez cuando el modo de servicio de la CPU pasa de STOP a RUN.
Clase de evento: Arranque (OB100)
Bloque de organización para la elaboración de alarmas de retardo (S7-1200)
El OB de alarma de retardo interrumpen la elaboración cíclica del programa tras transcurrir un tiempo determinado
Clase de evento: Interrupción de retardo de tiempo (OB20)
Bloque de organización para la elaboración de alarmas de tiempo (S7-1200)
El OB de alarma de tiempo interrumpen la elaboración cíclica del programa en intervalos de tiempo definidos.
Clase de evento: Interrupción cíclica (OB30)
Bloque de organización para la elaboración de alarmas de proceso HSC (S7-1200)
El OB de alarma de proceso interrumpen la elaboración cíclica del programa debido a eventos hardware. Las interrupciones hardware no sólo se utilizan para las "alarmas del proceso HSC", sino también para las alarmas del proceso de los canales digitales.
Clase de evento: Interrupción hardware (OB40)
Bloque de organización para la elaboración de alarmas de diagnóstico (S7-1200)
Si un módulo con capacidad de diagnóstico, para el cual ha sido liberada la alarma de diagnóstico, reconoce un error, el OB de alarma de diagnóstico interrumpe la elaboración cíclica del programa.
Clase de evento: Interrupción de error de diagnóstico (OB82)
Bloque de organización para la elaboración de errores de tiempo (S7-1200)
Si se sobrepasa el tiempo de ciclo máximo, el OB de error de tiempo interrumpe la elaboración cíclica del programa.
Clase de evento: Interrupción de fallo de tiempo (OB80)

Tabla 2 Principales OB's para la gama s7-1200 fuente: <http://support.automation.siemens.com/>

Para más información sobre los tipos de OB's, así como del funcionamiento general de la programación en TIA Portal consultar el manual del S7-1200 adjunto en el Anexo VIII.

El primer OB que se ha empleado, es el que crea por defecto el TIA Portal al iniciar un nuevo proyecto, el OB1.

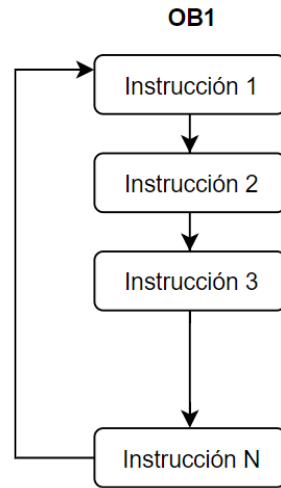


Figura 18 Funcionamiento básico OB1

Para programas cortos y simples se puede implementar todo en el OB1 (como en la imagen superior), pero cuando los programas se requieren para problemas más complejos, la mejor forma de escribir el código es dividiéndolo en bloques más pequeños. Esto permite dividir el programa por partes, facilitando el control de los distintos subprocesos de forma separada, pero que se ejecutarán como un programa único durante el funcionamiento del proceso.

Esquemáticamente se podría representar así:

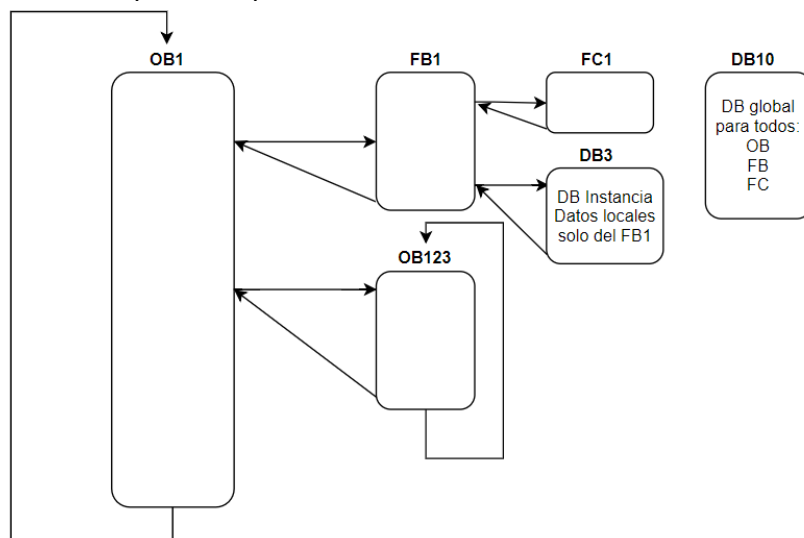


Figura 19 Ejemplo de estructuración de un programa complejo

En el OB1 Main, se han llamado a los bloques de función FB **“Silo”**, y a las FC **“Alarmas_y_fallos_SILOS”** y **“Activación_actuadores”** entre otras líneas de código. El código completo del OB puede consultarse en el **anexo VI**.

Siguiendo las jerarquías a lo largo de este punto, en la figura 20 se muestra el cronograma con el orden de ejecución que realizará el programa, leído de izquierda a derecha y de arriba hacia abajo.

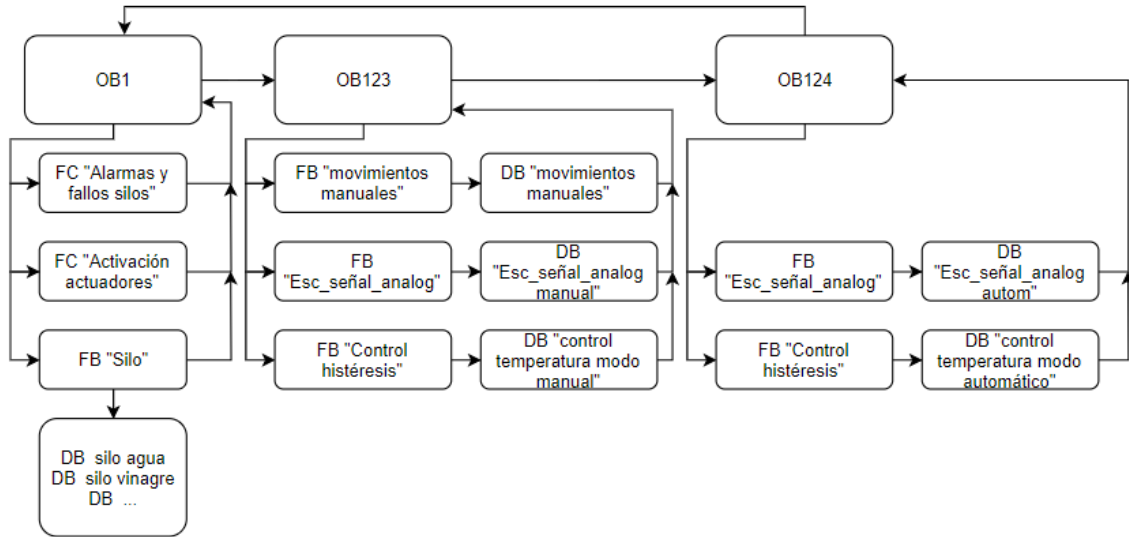


Figura 20 Cronograma ejecución OB's

7.4.2 BLOQUE ORGANIZACIÓN “MODO_AUTOMAT” [OB124]

Tras lo visto en el apartado anterior, en este bloque de organización se ha escrito el código encargado de controlar el modo automático. Para este modo se seguirían los siguientes pasos:

- Vertido de ingredientes
- Mezclado
- Trasvasado
- Triturado y cocido

El diagrama de funcionamiento de la función se muestra en la figura 21:

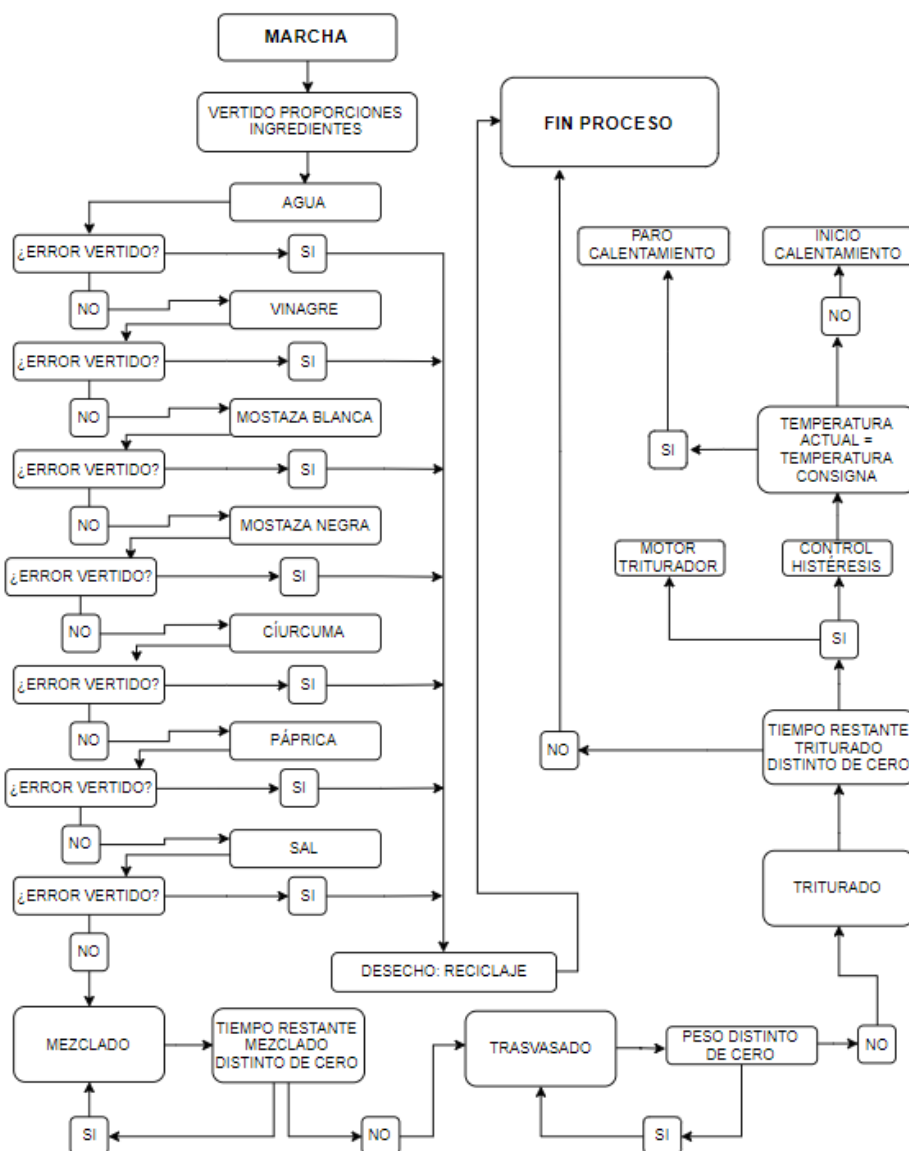


Figura 21 Diagrama de funcionamiento “Modo_automat”

El código completo del bloque puede consultarse en el [anexo VI](#).

7.4.3 BLOQUE ORGANIZACIÓN “MOV_MANUALES” [OB123]

En este bloque de organización se ha escrito el código encargado de controlar el modo Manual. Se siguen los siguientes pasos:

- Vertido de ingredientes manual
- Selección del proceso a ejecutar: mezcla, trasvase o triturado
- Gestión de los procesos mezcla, trasvase y triturado.

El diagrama de funcionamiento de la función se muestra en la figura 22:

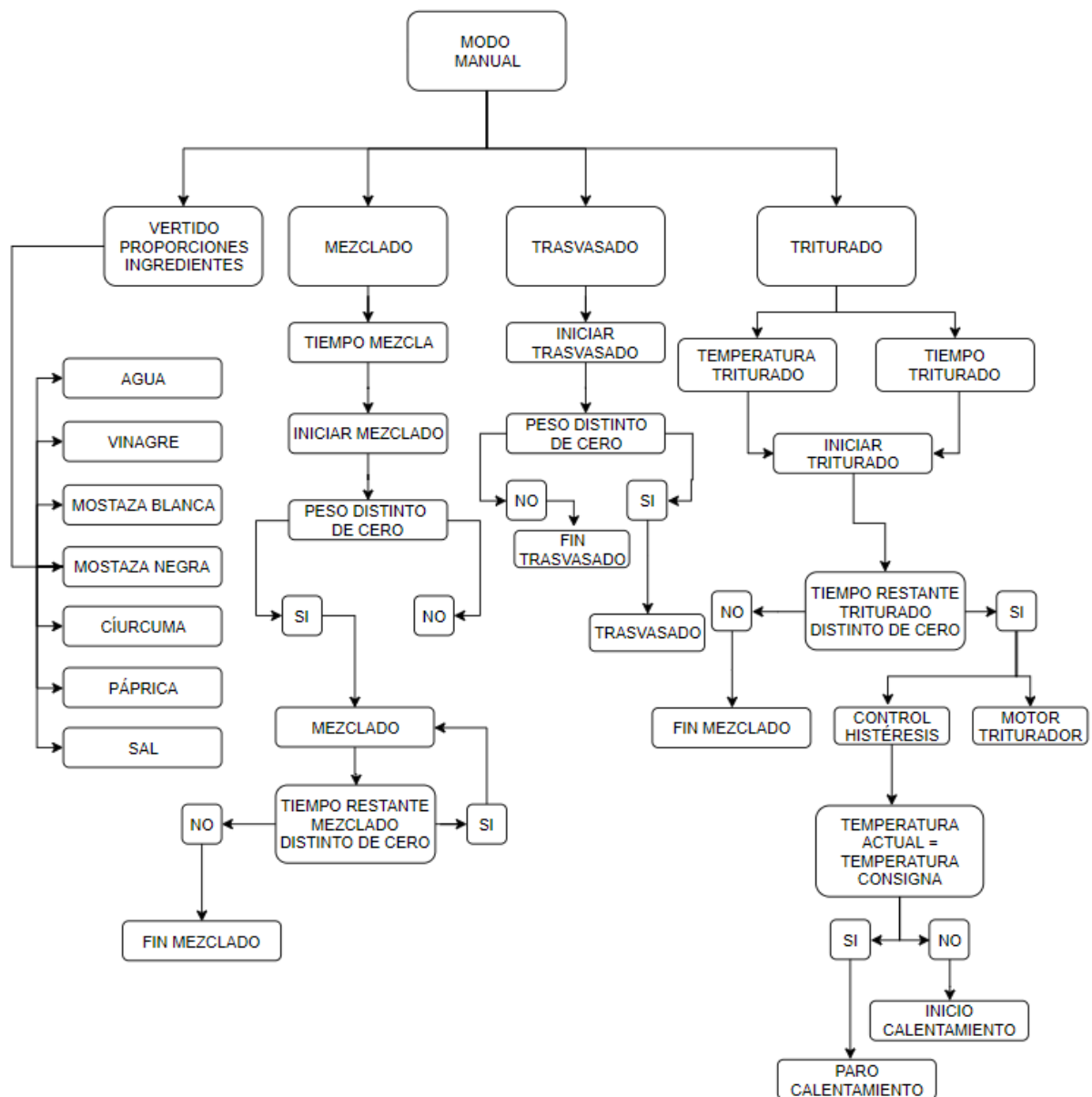


Figura 22 Diagrama de funcionamiento “Mov_manuales”

El código completo del bloque puede consultarse en el [anexo VI](#).

7.5 DESARROLLO DE UNA LIBRERÍA DE BLOQUES DE FUNCIÓN

Para facilitar la programación de futuros proyectos, el software TIA Portal permite guardar bloques de función (FB) o funciones (FC) de manufactura propia para hacer posible su reutilización en otros proyectos sin necesidad de programarlas de nuevo.

Estas librerías almacenan los diferentes bloques de función o funciones creadas por el programador, de modo que en el futuro no tenga que preocuparse de entender cómo consigue sus objetivos el código interno, si no únicamente entender cómo se usan y deben de conectarse las diferentes entradas y salidas asociadas a cada patilla. Estos bloques de función o funciones pueden tener integrada una programación con una funcionalidad compleja.

Las librerías por tanto, permiten la creación de programas de forma modular y reutilizable, reduciendo las líneas de código, así como el tiempo necesario para realizar la programación del proyecto.

Para la programación del presente trabajo de fin de grado, se han creado varias funciones y bloques de función, unos con la finalidad de integrarlos a la librería de uso general y otros para facilitar la tarea del desarrollo del programa. Estas últimas pues, no se incluirán en la librería, ya que tienen una utilidad específica para el caso al que se pretende dar solución, esto es al proceso de fabricación de la salsa de mostaza.

Se tiene en definitiva dos grupos de bloques de función o funciones:

- **Bloques de función genéricos (para la librería):** Estos bloques de función se integrarán posteriormente a la librería de funciones propias. Serán pues bloques de función genéricos que se puedan reutilizar en otros proyectos. Los bloques de función genéricos que se añadirán a la librería serán:
 - Silo
 - Control histéresis
 - Esc_señal_analog
- **Bloques de función específicos (únicamente para el proyecto):** Estos bloques de función y funciones tienen utilidad únicamente en el proyecto, y se desarrollan a parte del bloque de organización principal para estructurar el programa, y facilitar el seguimiento de las tareas que deben ejecutarse por parte del PLC. Los bloques de función y funciones específicas para este proyecto serán:
 - Activación actuadores
 - Alarmas_y_fallos_SILOS
 - Movimientos_manuales

7.5.1 BLOQUES DE FUNCIÓN GENERALES DESTINADOS A LA LIBRERÍA

A continuación se explicarán los bloques de función genéricos que se podrán aprovechar y reutilizar en futuros proyectos.

7.5.1.1 BLOQUE DE FUNCIÓN "SILO"

Las variables declaradas para la creación del bloque de función se muestran en la tabla 3:

Input		Static	
valor_actual_peso	Real	marcha_gruesos	Bool
reset_material	Bool	marcha_finos	Bool
Peso_consigna	Real	puesta_cero	Bool
inicio	Bool	et1	Time
Reset_fallo_motores	Bool	et2	Time
paro	Bool	et3	Time
Parámetros	Parámetros_silo	et4	Time
Output		et5	Time
motor_gruesos	Bool	et6	Time
motor_finos	Bool	et7	Time
nivel_bajo	Bool	aux_1	Bool
fallo_nivel	Bool	aux_2	Bool
Vertido_en_proceso	Bool	aux_3	Bool
Fin_vertido	Bool	aux_4	Bool
Fallo_motor_gruesos	Bool	aux_5	Bool
Fallo_motor_finos	Bool	aux_6	Bool
Error_vertido	Bool	aux_7	Bool
InOut		M_N_fallo	Bool
cantidad_restante	Real	A_B_fallo	Bool
		Fallo_consigna_peso	Bool
		Fallo_capacidad_max	Bool
		IEC_Timer_0_Instance	TON_TIME
		IEC_Timer_0_Instance_1	TON_TIME
		IEC_Timer_0_Instance_2	TON_TIME
		IEC_Timer_0_Instance_3	TON_TIME
		IEC_Timer_0_Instance_4	TON_TIME
		IEC_Timer_0_Instance_5	TON_TIME
		Q	Real
		W	Real
		x	Real
		y	Real
		Peso_objetivo	Real

Tabla 3 Variables declaradas FB "Silo"

Al realizar la llamada al bloque de función creado, aparecerá un recuadro como el mostrado en la figura 23:

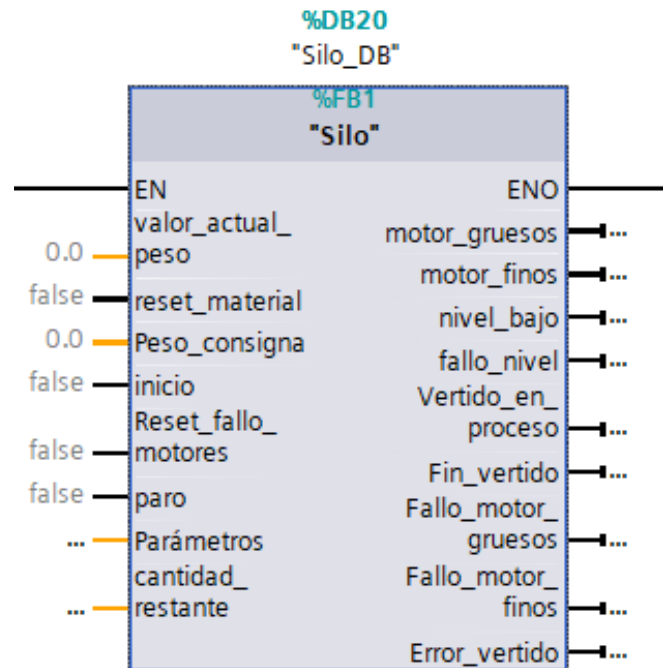


Figura 23 Visualización función SILO

Como se puede apreciar en la figura, se dispone de varias entradas y salidas; El bloque de función lee el estado de las entradas, primeramente leerá los parámetros del silo a la cual está referida, para conocer la cantidad de material restante, los niveles de aviso y fallo, y cuando desconectar los actuadores de vertido.

Seguidamente establecerá los objetivos a alcanzar (**Peso_consigna**) y trabajará en base a ello. Activará los actuadores e irá leyendo la variable **valor_actual_peso** y en función de su valor, desconectará los actuadores que vierten el material hasta alcanzar el peso consigna buscado.

Si la lectura del **valor_actual_peso**, se encuentra dentro del margen de error de la parametrización del silo, entonces activará la salida **Fin_vertido** informando de que el proceso se ha ejecutado satisfactoriamente. Si por cualquier motivo, hubiese habido un error en el vertido, y la lectura es superior al peso consigna fuera de la tolerancia establecida, entonces se activará la variable **Error_vertido**.

Por otra parte, el **fallo_motor_gruesos** y **fallo_motor_finos** se activarán al estar activados demasiado tiempo sin llegar al peso consigna. Es decir, si se activan los actuadores y la lectura del peso rebasa un tiempo parametrizado sin llegar al peso consigna, saltará el fallo indicando que se tiene un problema con el actuador, ya sea que está obstruido, averiado o bloqueado. Para subsanar el error e informar al sistema de que el fallo está subsanado, se deberá de activar la entrada **Reset_fallo_motores** desde el SCADA.

El **reset_fallo_motores** pondrá al sistema en pausa, a la espera de que se repare o elimine el error. Una vez se presione el reset del fallo de los motores, el sistema se pondrá otra vez automáticamente en marcha hasta finalizar el ciclo automático.

La entrada **Reset_material** servirá para restablecer el nivel de material del silo hasta su máximo, por lo que es importante que el operario que se encargue de realizar esta tarea se asegure de que se ha rellenado el silo completamente. Para ayudar en la planificación del rellenado y/o contacto con proveedores, se han programado los avisos de nivel y fallo de nivel.

Las variables **Nivel_bajo** y **Fallo_nivel** se calculan internamente en función de la parametrización establecida en el silo, y se activarán informando de error o aviso, en caso de que se cumpla la condición necesaria para ello, esto es que se lea un nivel de material restante inferior a cualquiera de los dos valores calculados.

Para permitir que la función se inicie, la entrada **Inicio** deberá estar activa, siendo por tanto, necesariamente complementaria con la activación del EN.

La lógica interna de la función puede consultarse en el **anexo VI**. La figura 24 muestra un ejemplo del conexionado de la función en el programa:

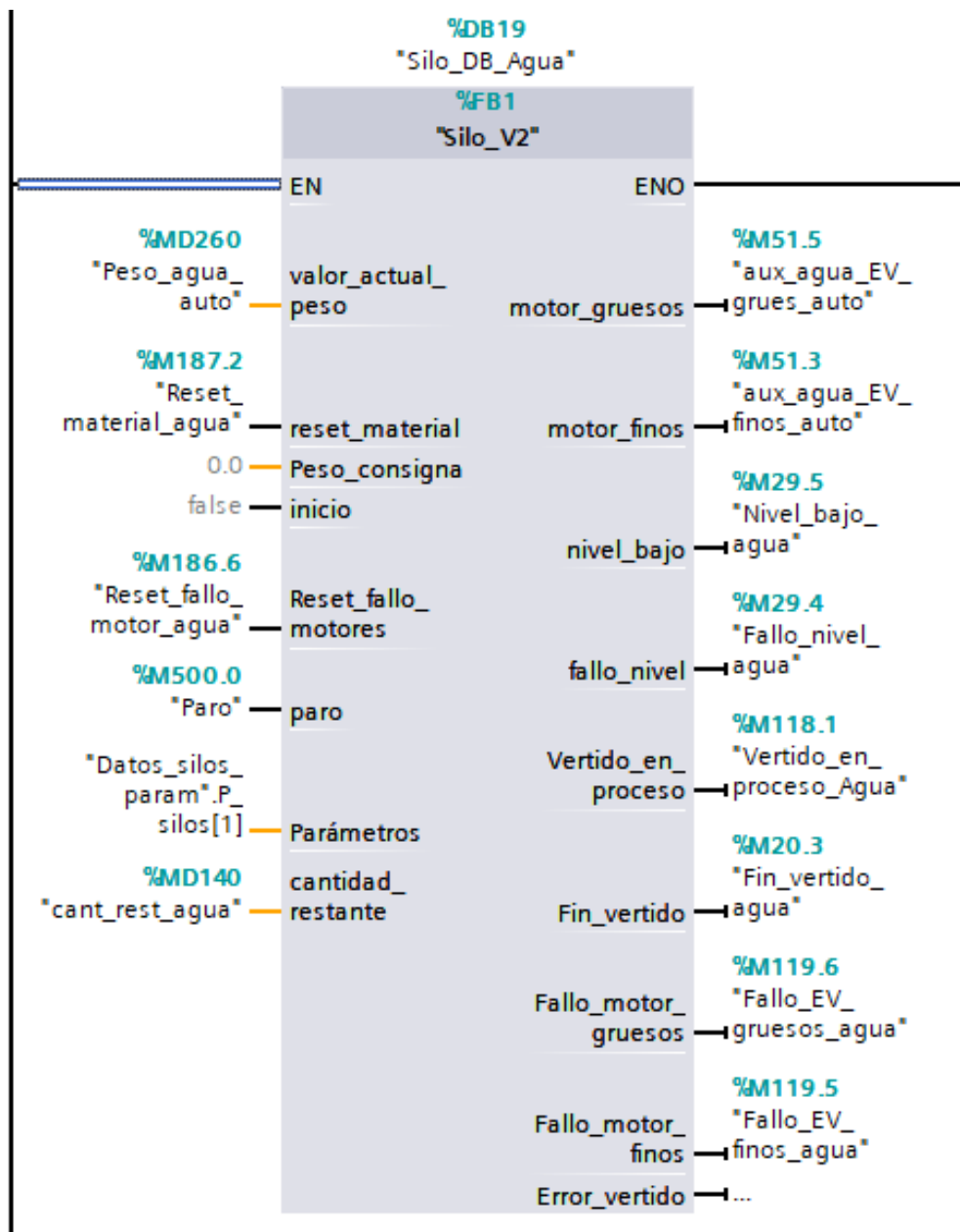


Figura 24 Ejemplo conexionado función SILO

7.5.1.2 BLOQUE DE FUNCIÓN "CONTROL_HISTÉRESIS"

Las variables declaradas para la creación del bloque de función se muestran en la tabla 4:

Input	
Consigna	Real
Banda_control	Real
Sobretemp_alarma	Real
Temperatura_actual	Real
Output	
Calefacción_ON	Bool
Alarma	Bool
Static	
Banda_superior	Real
Banda_inferior	Real
Temp_alarma	Real

Tabla 4 Variables declaradas bloque de función "Control_Histéresis"

La temperatura de este proceso es un parámetro crítico, pero admite un cierto grado de tolerancia. No existe una diferencia significativa en el resultado final del producto en cuanto a que la temperatura de cocido sea 90 que 91.7 °C, por lo que se ha desarrollado un control por histéresis.

En este tipo de control lo que se busca es que se mantengan encendidos los elementos responsables del calentamiento hasta se alcance un límite superior, y se mantengan apagados hasta que al alcanzar un límite inferior se enciendan de nuevo.

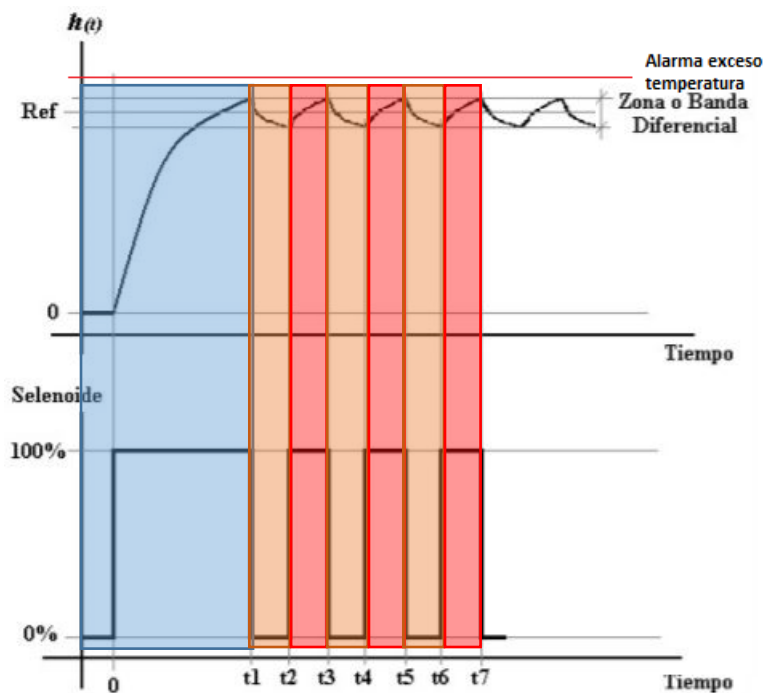


Figura 25 Ejemplo de un control por histéresis. Fuente: https://zulaco64.updog.co/Material_Didactico/Control-OnOff_Sistemas_Control_Controladores.pdf

Como se puede observar en el gráfico, se pueden diferenciar en tres zonas:

- **Zona inicial (azul):** En esta zona se mantendrán los elementos calefactores activados hasta que se toque la banda superior de control.
- **La zona de temperatura decreciente (naranja):** Una vez se ha alcanzado este límite superior se desactivan los elementos calefactores hasta tocar el límite inferior de la banda de histéresis.
- **La zona de temperatura creciente (rojo)** tras por encima del límite inferior de la banda de histéresis: Cuando la temperatura descienda demasiado y la lectura del termómetro toque la banda inferior, se encenderán de nuevo los elementos calefactores hasta tocar de nuevo la banda superior, momento en el que se apagarán.

Una vez se han dado los dos últimos puntos estos se irán repitiendo tantas veces como sea necesario mientras dure el tiempo programado.

Al realizar la llamada de la función en el MAIN aparecerá un bloque como el que se muestra en la figura 26:

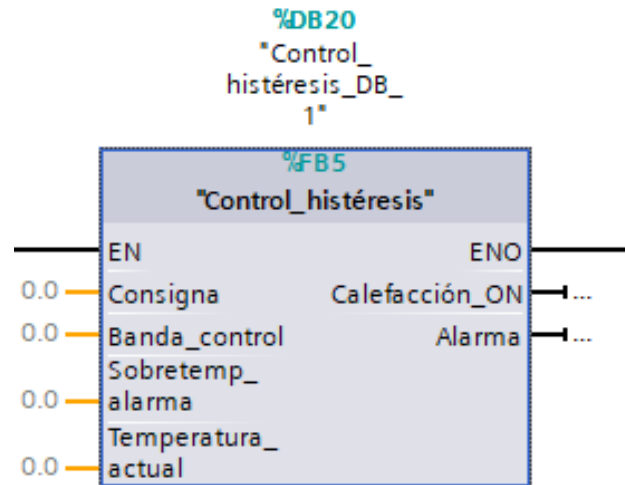


Figura 26 Vista de la función "Control_histéresis"

En esta función como entradas se tiene la **Consigna** de temperatura, que será la temperatura objetivo a alcanzar. La **banda_control** deberá ser un valor entre cero y cien, y marcará la tolerancia en porcentaje de la banda de histéresis. Es decir, cuanto mayor sea este número mayor será la amplitud de la banda, por lo que es recomendable mantenerlo en valores relativamente bajos.

Por ejemplo, si la temperatura consiga es de 95°C y el valor de Banda_control es de 5, significa que el límite superior de la banda de control a partir del cual se desactivaran los elementos calefactores se fijará en 99.75°C, y el valor del límite inferior a partir del cual se activarán de nuevo los actuadores será de 90.25°C.

Por lo que es interesante y recomendable que cuanto mayor sea la consigna de temperatura menor sea el valor de esta variable.

La **Sobretemp_alarma**, al igual que la banda de control, deberá de ser un valor entre el valor de la Banda_control y cien, pues es un valor que servirá para calcular a partir de que temperatura saltará el fallo de alarma por exceso de temperatura.

Siguiendo con el ejemplo anterior, si la banda de control es un cinco por ciento del valor de la consigna, el valor que se podrá asignar a esta entrada irá de seis a cien, por lo que si se le asigna un siete, en el caso de que se produjese un fallo, el valor a partir del cual se activaría la salida **alarma** sería de 101.65°C.

La salida **Calefacción_ON** irá poniéndose a uno o a cero en función de la lectura de la temperatura del termómetro, así como de la parametrización de la banda de histéresis.

La figura 27 muestra un ejemplo de la conexión de la función en el OB 124 *Modo_automat*:

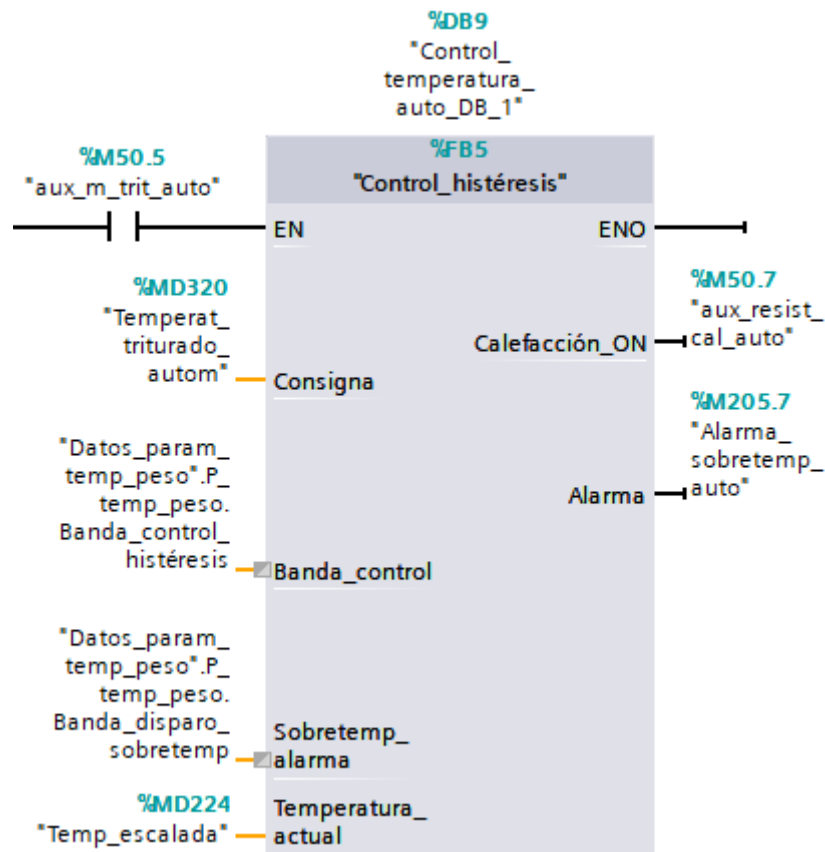


Figura 27 Ejemplo conexionado función "Control_histéresis"

La lógica interna de la función puede consultarse en el [anexo VI](#).

7.5.1.3 BLOQUE DEBFUNCIÓN “ESC_SEÑAL_ANALOG”

Únicamente se dispone de una báscula para todo el proceso, por lo que se hace necesario trabajar y adaptar la señal del sensor de presión a los distintos pesos del vertido.

Las variables necesarias para el desarrollo del bloque de función son las mostradas en la tabla 5:

Input	
Señal_analógica_sensor	Int
Proceso_1	Bool
Proceso_2	Bool
Proceso_3	Bool
Proceso_4	Bool
Proceso_5	Bool
Proceso_6	Bool
Proceso_7	Bool
Proceso_8	Bool
MAX	Real
MIN	Real
Reset_memo	Bool
Output	
señal_escalada_S1	Real
señal_escalada_S2	Real
señal_escalada_S3	Real
señal_escalada_S4	Real
señal_escalada_S5	Real
señal_escalada_S6	Real
señal_escalada_S7	Real
señal_escalada_S8	Real

Tabla 5 Variables declaradas "Esc_señal_analog"

El tener una única báscula obliga a tener que depositar los materiales de uno en uno para controlar el peso de cada materia prima, de lo contrario no habría forma de discernir que peso corresponde a qué material.

Así pues, al igual que para la función "Silo", al llamarla en el MAIN aparecerá un bloque como el mostrado en la figura 28:

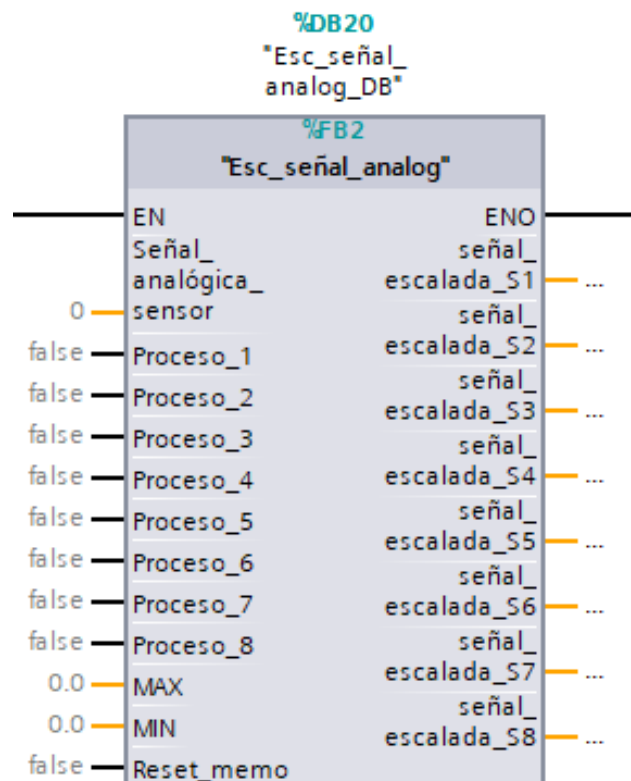


Figura 28 Vista función "Esc_señal_analog"

Como se puede observar se dispone de varias entradas:

- A la entrada **Señal_analógica_sensor** se conectará directamente la lectura de la señal. La función internamente escala y adecúa la señal para transformarla en un número entre el rango máximo y el mínimo del peso que soporta la báscula.
- **MAX** y **MIN** son los valores numéricos del máximo y mínimo peso que leerá la báscula.
- Al activar la entrada **Reset_memo** se pondrán los valores de salida de las todas las señales a cero.
- Al activar cualquiera de las variables de proceso (por **ejemplo proceso_1**) se permite que la variación de la lectura de la señal se asocie al peso de la materia prima que se haya asignada a dicha entrada (en este caso, el proceso_1 correspondería al peso del agua). Es decir, mediante la activación de las entradas de proceso, se estará informando a la función de la báscula de qué material se está vertiendo. El número resultante de cada salida ya es el correspondiente a la señal escalada del peso en kg.

La lógica interna de la función puede consultarse en el **anexo VI**.

La figura 29 muestra un ejemplo de conexionado de la función en el programa:

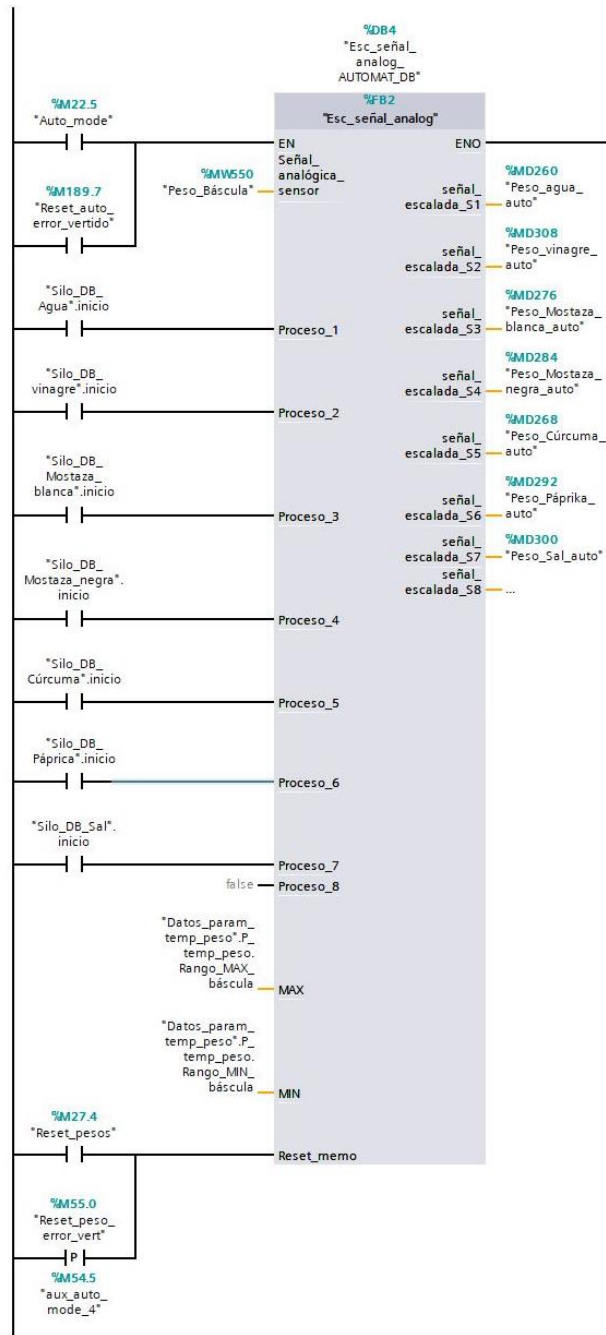


Figura 29 Ejemplo conexionado de la función "Esc_senál_analog"

7.5.2 CREAR UNA LIBRERÍA

A continuación se muestran los pasos necesarios para guardar las funciones genéricas creadas en una librería para disponer de ellas en otros proyectos:

En el margen derecho de la pantalla, se encontrará la vista mostrada en la figura 30.

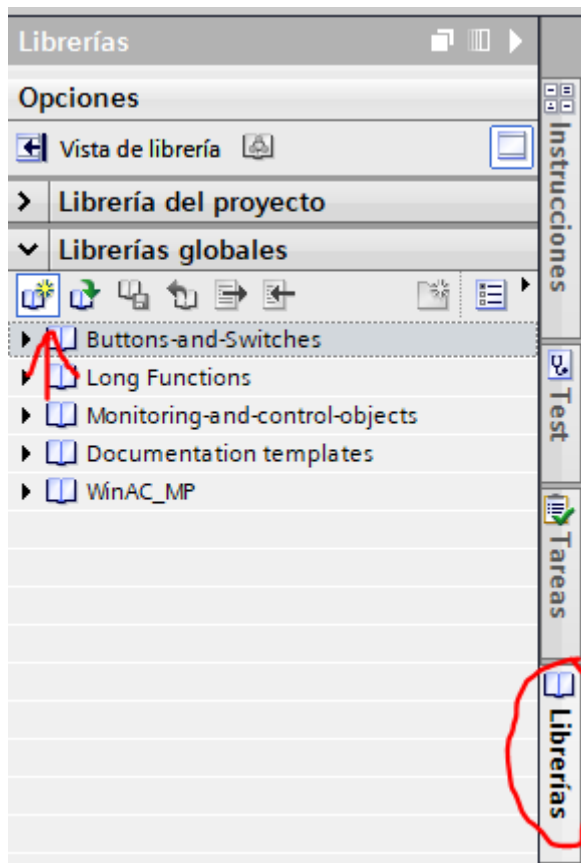


Figura 30 Primer paso para crear una librería de FB's

Haciendo click sobre la opción “Librerías” y luego sobre “crear librería global” aparecerá la ventana emergente mostrada en la figura 31:

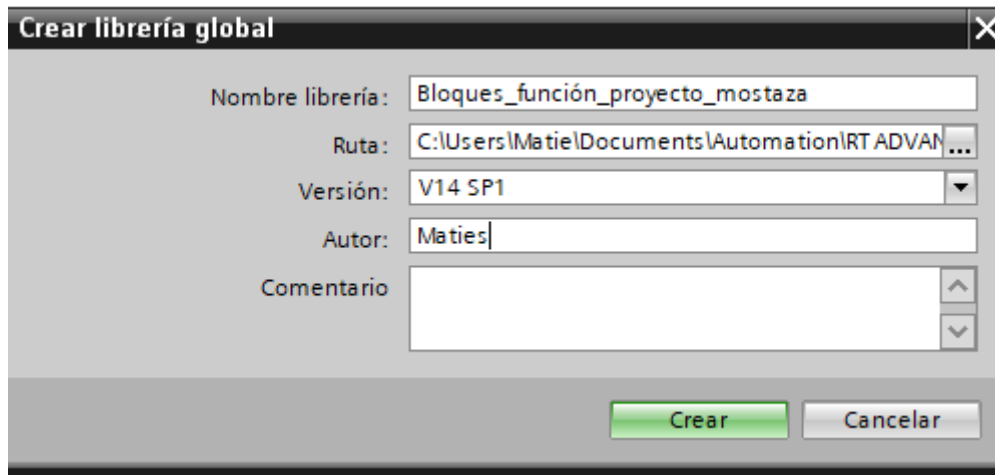


Figura 31 Segundo paso para crear una librería de FB's

Se le asigna el nombre con que se quiere guardar y se pulsa “Crear”.

Una vez creada la librería, para introducir los FB's creados, se deberá de hacer click sobre los propios FB's y sin soltar, arrastarlos a las plantillas maestras.

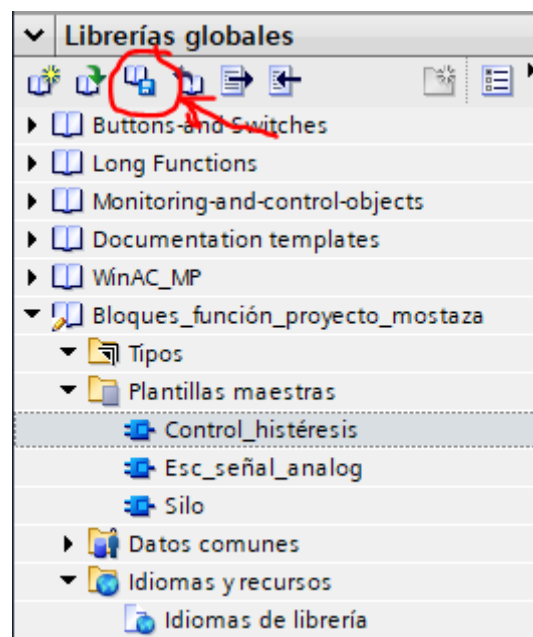


Figura 32 Vista de la librería creada

Una vez se hayan añadido todas las funciones se guardan los cambios y ya se está en disposición de utilizar estos bloques de función en todos los proyectos en los que se tenga necesidad de ellos.

7.5.3 BLOQUES DE FUNCIÓN ESPECÍFICOS (ÚNICAMENTE PARA EL PROYECTO):

Una vez descritos los bloques de función genéricos, se expondrán los específicos para el proyecto.

7.5.3.1 GESTIÓN DE LOS PARÁMETROS DE LOS SILOS

Los sistemas de almacenamiento o silos son elementos clave en los sistemas de dosificación y pesaje industrial. Sus características principales (dimensiones, capacidad, etc.) pueden variar en función del ingrediente, del proceso o incluso del volumen de producción que tenga la industria.

Además, a lo largo de la vida útil de la instalación industrial estos pueden ampliarse o incluso ser sustituidos por otros para adaptarse a las futuras necesidades de la planta.

Para mejorar la gestión de los silos se ha optado por **crear un tipo de dato** y utilizar un DB donde almacenar las características de estos, evitando que el usuario tenga que acceder al programa principal y localizar donde debe realizar las modificaciones.

El TIA Portal, al igual que otros software de programación de PLC's, permite al usuario declarar tipos de datos en función de las necesidades del mismo.

En este caso, como los parámetros de los silos son fijos, se ha creado un tipo de dato con todas las variables referentes a la parametrización de los silos para una mayor claridad tanto en el programa como a la hora de la gestión de los datos desde el HMI, como muestra la figura 33.

8	Parámetros	*Parámetros_silo*			
9	capacidad_max	Real	0.0		
10	A	Real	0.0		P/ calcular el nivel aviso
11	B	Real	0.0		P/ calcular el nivel de fallo
12	M	Real	0.0		P/ tener en cuenta C.D.C y desconectar grueso
13	N	Real	0.0		P/ tener en cuenta C.D.C y desconectar finos
14	tiempo_gruesos	Tíme	T#0ms		
15	tiempo_finos	Tíme	T#0ms		
16	Tolerancia_pesado	Real	0.0		
17	Cola_de_carga	Real	0.0		

Figura 33 Ejemplo del tipo de dato "Parámetros_silos"

Los parámetros de los silos deben de ser un tipo de dato remanente, es decir, que deberán de permanecer grabados en el autómata tras apagarlo, por lo que se hace imperativo marcar la casilla de remanencia. De esta forma se evita que ante una desconexión de la planta, ya sea intencionada o anómala, los datos de la parametrización de los silos y demás componentes de la planta se pierdan.

Y no solo eso, además, se ha creado una pantalla que apunta directamente a estas direcciones de memoria, por lo que el administrador de la planta podrá cambiar la parametrización de los silos desde el SCADA.

De esta forma se evita tener que entrar al entorno de programación y modificar el programa de control, evadiendo los problemas asociados que se puedan derivar de una mala manipulación del programa. Desde la pantalla del HMI únicamente el administrador tendrá acceso a la modificación de los parámetros de los silos, la báscula y la gestión del control de temperatura puesto que se ha implantado un control de usuarios.

7.5.3.2 GESTIÓN DE LAS ALARMAS GENERADAS POR EL PROCESO

Para la gestión de las alarmas, se ha creado una función tipo FC.

La diferencia entre un FB (bloque de función) y un FC (función) es que los bloques de función son bloques lógicos que depositan sus valores de forma permanente en bloques de datos de instancia, de modo que estos datos siguen estando disponibles después de procesar el bloque, mientras que las funciones son bloques lógicos sin memoria.

En el FC **Alarmas_y_fallos_SILOS** se va a leer el estado de algunas entradas y salidas de la función silo, así como de otras funciones que se explicaran posteriormente en este documento.

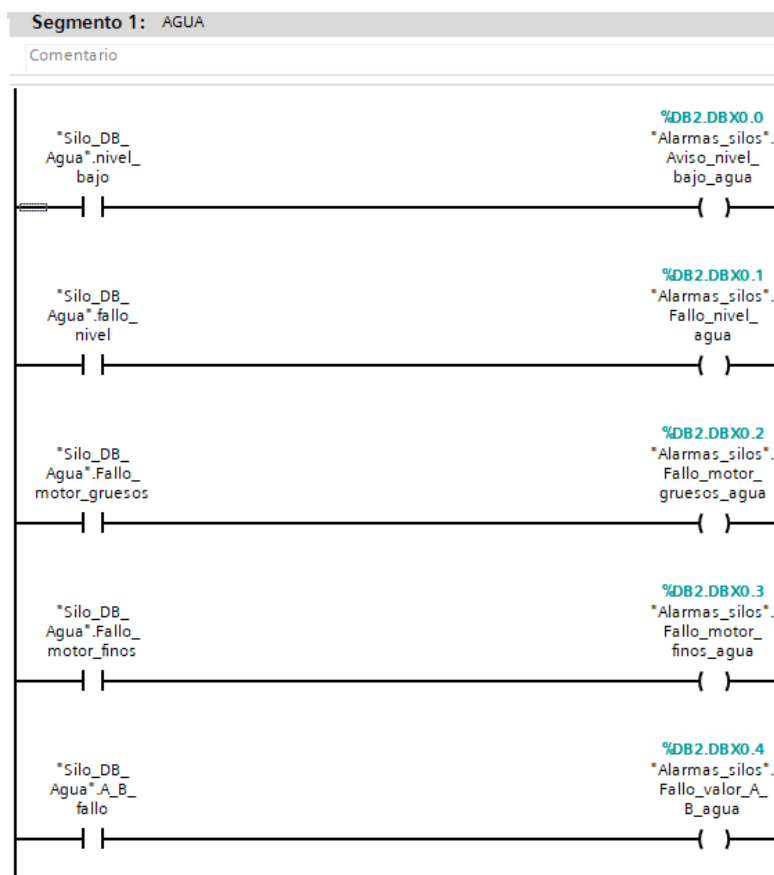


Figura 34 Ejemplo creación de las alarmas a mostrar en el HMI

En la figura 34, se muestra como la función lee los estados de las variables asignadas al bloque de función "silo" y según sea su estado activa o desactiva las alarmas correspondientes, en este caso, al silo que almacena el agua. Se tendrán pues, tantos segmentos como silos además de otras alarmas, como las de temperatura o error de vertido.

Se han organizado las alarmas en una FC por dotar de mayor claridad al programa. De esta forma se tienen todas las alarmas en una sola función, a la que se asociará un DB para realizar la lectura desde el HMI.

En el DB asociado a esta FC se dispondrá de todas las alarmas juntas. En este caso se ha optado por crear un DB compuesto por bits, ya que su tratamiento posterior en el HMI para organizar las tablas de fallos permite numerar los fallos, de modo que cada error o aviso, se asocie con un número que se pueda consultar en una tabla de fallos donde aparezca el número de fallo, la descripción de la causa del fallo, las posibles soluciones, etc.

	Nombre	Tipo de datos	Offset	Valor ...	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Aviso_nivel_bajo_agua	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Fallo_nivel_agua	Bool	0.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Fallo_motor_gruesos_agua	Bool	0.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Fallo_motor_finos_agua	Bool	0.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Fallo_valor_A_B_agua	Bool	0.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Fallo_valor_M_N_agua	Bool	0.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Fallo_peso_consigna_agua	Bool	0.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Fallo_valor_capac_max_agua	Bool	0.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Aviso_error_vertido_agua	Bool	1.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Aviso_nivel_bajo_vinagre	Bool	1.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Fallo_nivel_vinagre	Bool	1.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Fallo_motor_gruesos_vinagre	Bool	1.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Fallo_motor_finos_vinagre	Bool	1.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Fallo_valor_A_B_vinagre	Bool	1.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Fallo_valor_M_N_vinagre	Bool	1.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Fallo_peso_consigna_vinagre	Bool	1.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Fallo_valor_capac_max_vinagre	Bool	2.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Aviso_error_vertido_vinagre	Bool	2.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 35 Vista interior del DB "Alarmas_silos"

La figura 35 muestra el DB de alarmas se puede observar cómo se van repitiendo los fallos para cada silo y cómo se van asignando a un bit (observable en el offset).

Para que la función se active, se deberá de llamar desde el MAIN, en este caso no se tienen entradas y salidas como en la función silo, únicamente la activación, ya que funciona como intermediaria en el tratamiento de datos de otras funciones.

La función lee el estado de las variables asociadas a otras funciones y permite tratar el estado en que se encuentran:

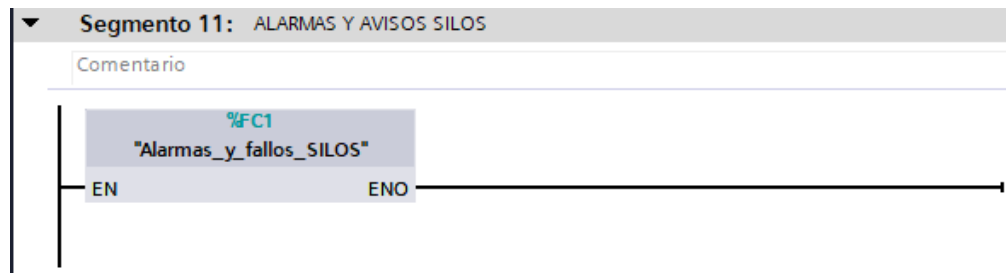


Figura 36 Vista al llamar la función "Alarmas_y_fallos_SILOS"

La totalidad del código de la función "Alarmas_y_fallos_SILOS" puede consultarse en el **anexo VI**.

La tabla completa de los fallos y avisos considerados puede consultarse en el **anexo VII**.

7.5.3.3 FUNCIÓN “MOVIMIENTOS_MANUALES”

En cualquier proceso productivo es habitual que se hagan pruebas y ajustes de las mezclas con cantidades reducidas, por ello, se ha elaborado una función que permita hacer estos experimentos sin tener que reproducir todo un ciclo completo de proceso, dotando de mayor versatilidad a la fábrica.

Al igual que el caso del escalado de la señal analógica. Se ha elaborado para un proceso genérico que permita reutilizar la función en futuros proyectos con las ventajas asociadas.

Se ha dotado a la función de la capacidad de controlar hasta 8 materiales diferentes con dos actuadores por material. En el caso del proceso al que se aplica, cabe recordar que solo se tienen siete materias primas y funciona perfectamente, demostrando así la modularidad del sistema.

Para activar el actuador deben darse dos condiciones:

- Que se haya activado previamente el material que se quiere verter, representado en esta función por las entradas **Silo_(1-8)**
- Que las entradas **marcha_finos_S(1-8)** y **marcha_gruesos_S(1-8)** se pongan a uno, encendiendo los actuadores del silo correspondiente.

Respecto a las salidas de esta función se tiene:

- **Material_(1-8)**: esta salida se pondrá a uno en función del material que se esté vertiendo. De modo que si se tiene la entrada Silo_3 activada, consecuentemente la salida Material_3 estará encendida.
- **Motor_finos_S(1-8)** y **motor_gruesos_S(1-8)** serán las salidas que piloten la orden de marcha de los actuadores.

Lo que se pretende con esta función es asegurar que se ha seleccionado el tipo de material que se pretende pesar, y que los actuadores y procesos del resto de materiales permanecerán desactivados mientras tanto.

Por lo tanto, si se activa el proceso_1, equivaldrá a decir que se quiere verter agua, y podrá controlarse la apertura y cierre de las electroválvulas que regulan el caudal vertido.

Al llamar la función, aparecerá un bloque como el que se muestra en la figura 37:

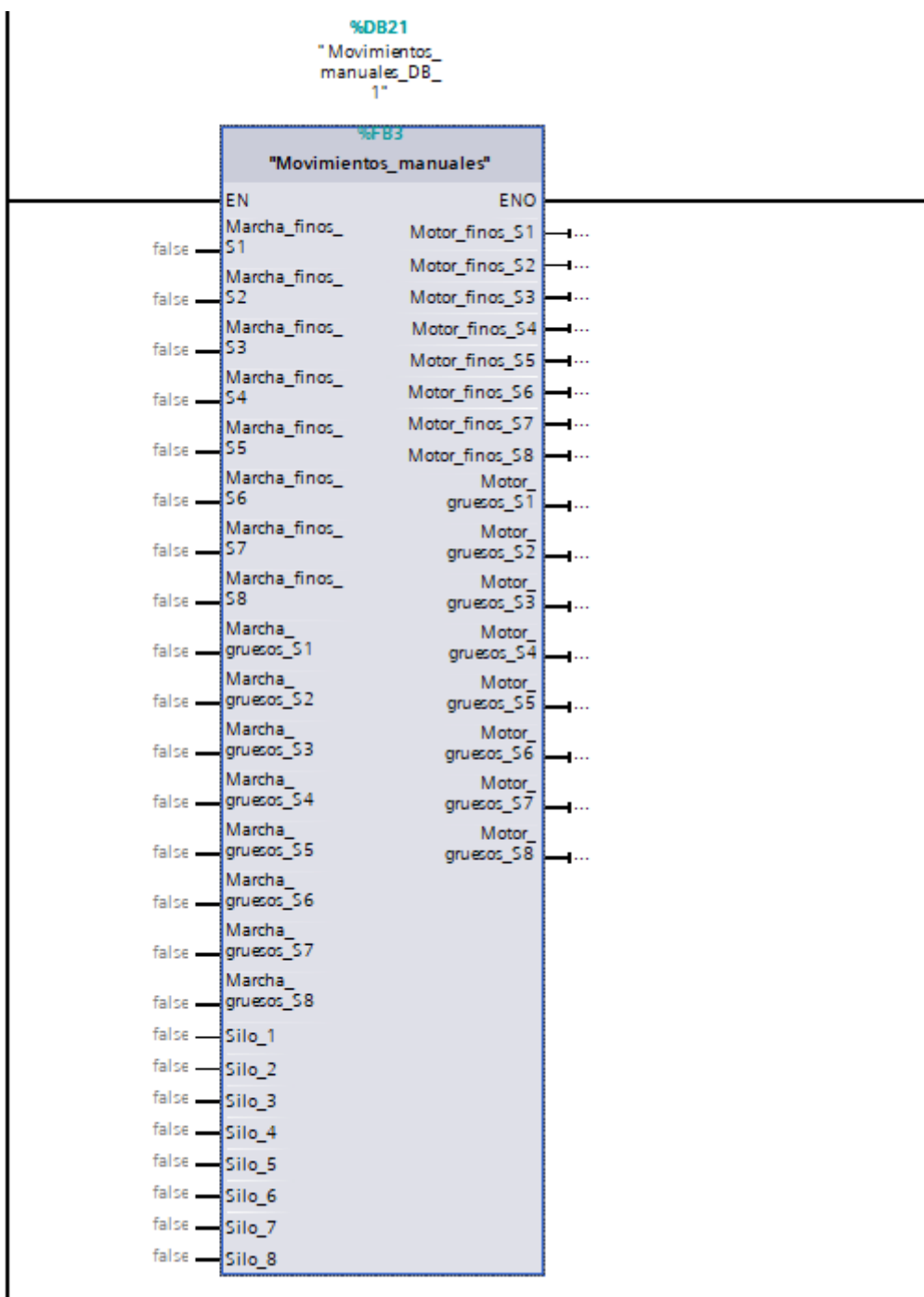


Figura 37 Vista función "Movimientos_manuales"

La lógica interna de la función puede consultarse en el anexo VI.

Esta función puede trabajar de manera complementaria a la función **Esc_señal_analog** para controlar en un mismo segmento el material escogido para el vertido y el peso del mismo que se tiene en la báscula.

Al realizar la llamada de la función en el MAIN se pueden conectar los bloques de función de manufactura propia tal como se muestra en la figura 36:

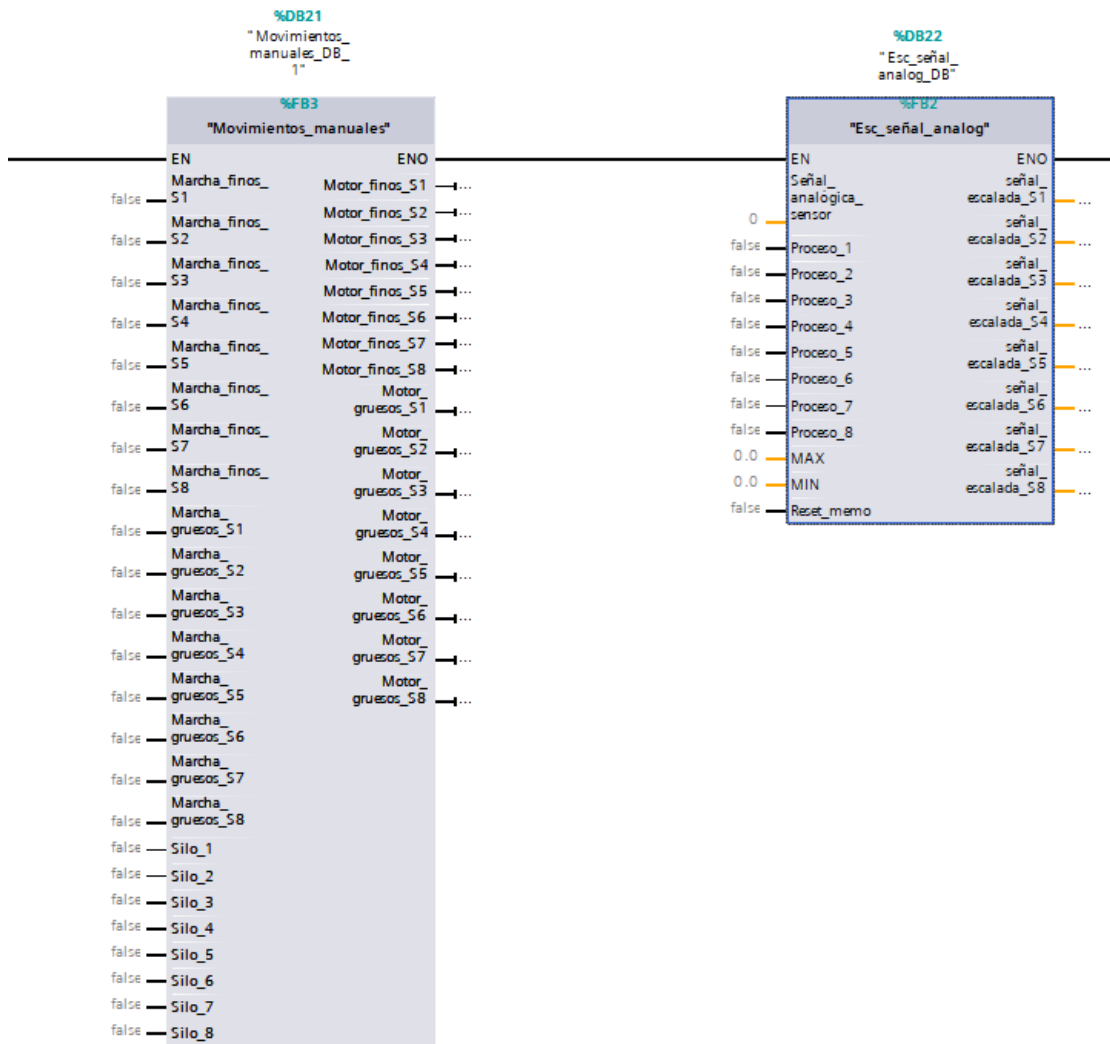


Figura 38 Ejemplo conexionado conjunto "Movimientos_manuales" y "Esc_señal_analog"

7.5.3.4 Función "Activación_ACTUADORES"

Como el programa es largo y tiene varias partes, se llama o activan los mismos actuadores a lo largo de diferentes líneas y partes del código.

La ejecución del programa se realiza de forma cíclica, lee el estado de las entradas, ejecuta el programa, actualiza el estado de las salidas, y repite de nuevo el mismo ciclo mientras esté encendido. Por lo tanto, si se intenta activar directamente las salidas a lo largo del programa con diferentes funciones y condiciones, es probable que el estado último que el autómata escribe en las salidas no sea el esperado.

Para evitar estos solapamientos, se han asociado memorias a cada salida, de modo que sean estas las que activen indirectamente los actuadores. Dentro de esta FC "Activación_actuadores" se encontrarán todas las memorias asociadas a los actuadores conectadas en paralelo con el mismo. La figura 39 muestra un extracto de la función a modo de ejemplo:

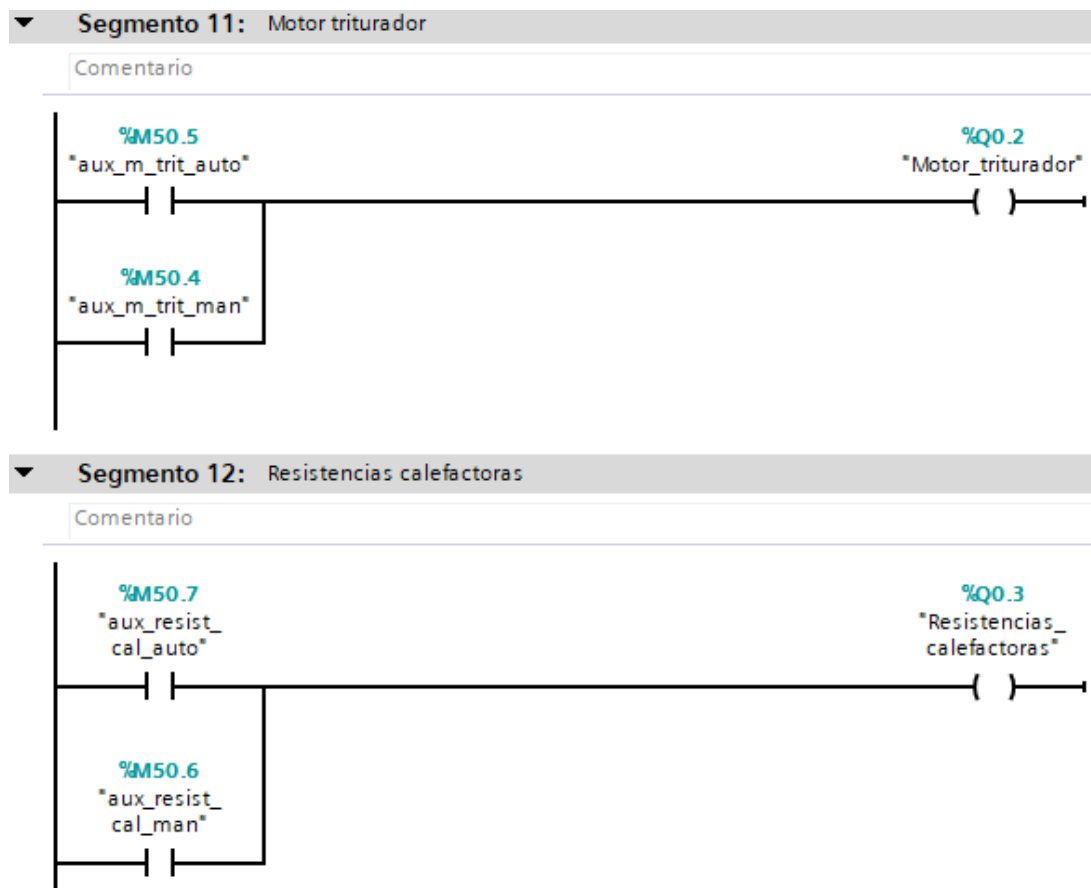


Figura 39 Ejemplo conexiones en la función "Activación_actuadores"

Esta función, al igual que la de "Alarmas_y_fallos_SILOS", actúan como pasarela y permiten mantener una estructura clara en el programa, es decir, no tienen entradas ni salidas, se han de llamar desde el MAIN y ejecutan el código que se les ha programado:



Figura 40 Vista al realizar la llamada de las funciones "Alarmas_y_fallos_SILO" y "Activación_actuadores"

La lógica interna de la función puede consultarse en el [anexo VI](#).

7.6 ENTORNO HMI, SOLUCIÓN ADOPTADA PARA EL SCADA

Para que la implementación de la automatización sea completa se debe de hacer uso de las herramientas que el WinCC dispone, no basta únicamente con hacer un programa que haga todas las funciones que necesita el proceso, hay otra etapa igual de importante que es que la pantalla que se muestre al operario, que será quién pase la mayor parte del tiempo con la máquina, sea fácil de comprender y que se pueda encontrar en ella toda la información necesaria de una manera sencilla y fácilmente entendible.

Para interactuar con el programa se ha desarrollado un sistema SCADA cuyo HMI será la pantalla del sistema PC que se destine para esta tarea en la planta. Desde un ordenador se manejará todo lo relacionado con el estado del proceso, la producción, las alarmas etc.

Persiguiendo estos objetivos, se han creado un total de siete pantallas:

7.6.1 PANTALLA DE INICIO

En la figura 41 se muestra la pantalla de inicio:

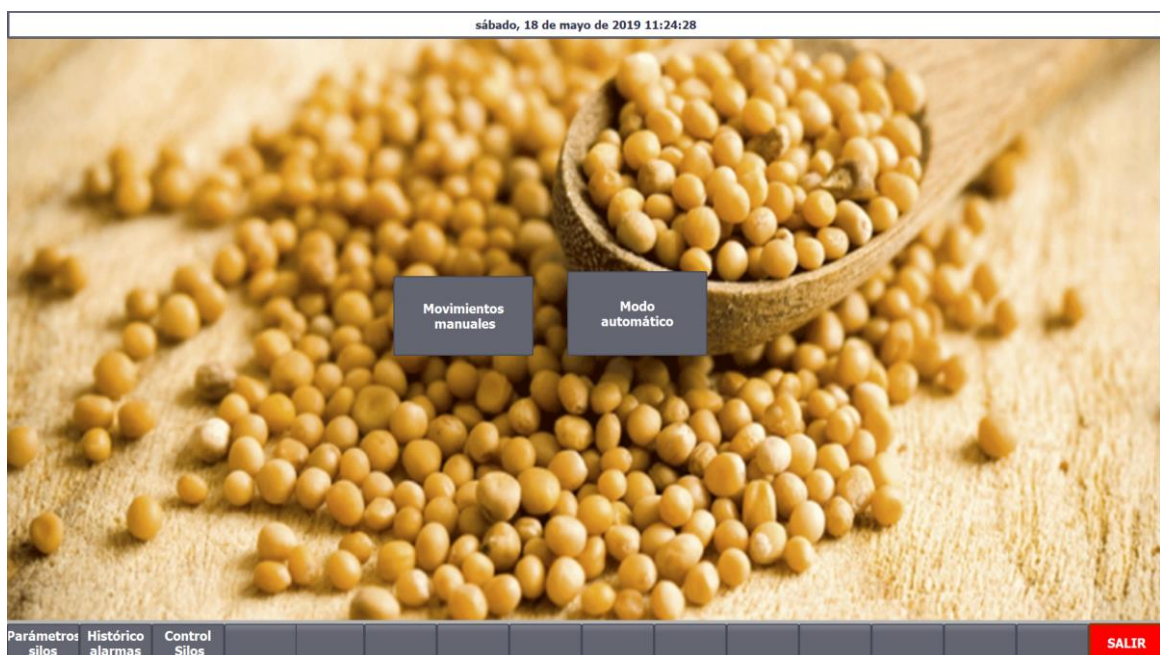


Figura 41 Pantalla de inicio

Esta pantalla será la que se muestre al acceder al HMI, y desde aquí se navegará a las diferentes pantallas que configuran el SCADA.

7.6.2 GESTIÓN DEL SISTEMA DE ALMACENAJE

Para tener un control de los silos, parametrizar y visualizar su estado en tiempo real se ha optado por diseñar una pantalla “Parámetros silos”, como la mostrada en la figura 42:

PARAMETRIZACIÓN SILOS		SILO 1	SILO 2	SILO 3	SILO 4	SILO 5	SILO 6	SILO 7
Cola de carga		5	5	5	5	5	5	5
Capacidad máxima		5000	3000	1500	1500	1000	1000	1000
Porcentaje sobre capacidad máxima para aviso nivel		15	20	20	20	20	20	20
Porcentaje sobre capacidad máxima para fallo nivel		8	5	5	5	3	3	3
Porcentaje sobre peso consigna para desconexión motor gruesos		85	85	85	85	95	95	95
Porcentaje sobre peso consigna para desconexión motor finos		95	95	95	95			
Tiempo a exceder para fallo motor gruesos		0	0	0	0	0	0	0
Tiempo a exceder para fallo motor finos		45	45	45	45			
Porcentaje de tolerancia sobre el peso consi		5	5	5	5	40	40	40

PARAMETRIZACIÓN BÁSCULA	
Peso máximo báscula	1200
Peso mínimo báscula	0

PARAMETRIZACIÓN CONTROL DE TEMPERATURA			
Temperatura máxima lectura	150	Banda histéresis	3
Temperatura mínima lectura	-20	Fallo exceso temperatur	5

Figura 42 Pantalla de parametrización de la planta

En esta pantalla se insertarán los parámetros de la planta.

Al ser datos de partes de la instalación que o bien no se cambian con asiduidad, como los silos, o que solamente se cambian cuando hay necesidad, de nuevo, los silos por reformas, o por ejemplo si la báscula se avería y ya no se fabrica una igual a la que había montada, debe de poder reparametrizarse la planta de un modo cómodo para el usuario final.

Como estos datos no son de trasiego habitual y son importantes para el desarrollo del proceso, pues la automatización depende en buena medida de ellos, se ha implantado un control de usuarios de modo que solo el personal a cargo con permiso y conocimientos específicos pueda modificar estos valores.

En esta pantalla se encontrarán los parámetros necesarios para el correcto funcionamiento de los silos, los parámetros de la báscula y los parámetros necesarios para la gestión de la temperatura en la fase del cocido y triturado.

En el Anexo I se puede encontrar más información referida la navegación y empleo de la pantalla “Parámetros silos”.

7.6.3 HISTÓRICO DE ALARMAS Y CONTROL DE FALLOS

La figura 43 muestra la pantalla “Histórico de alarmas”:

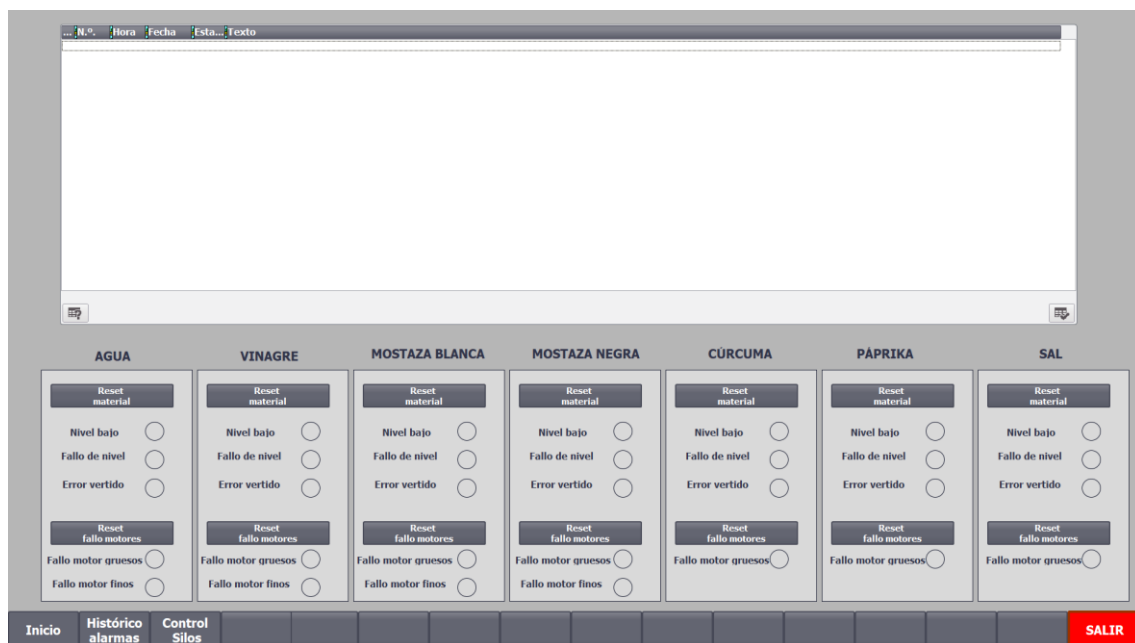


Figura 43 Pantalla del histórico de alarmas

En esta pantalla se mostrarán las alarmas o avisos que estén actualmente activos, así como las alarmas o avisos que se hayan acusado, permitiendo consultar si ha habido algún error y tener un control de cuando se produjo.

Además, permitirá un control visual del estado de los actuadores de vertido mediante el piloto “Fallo motor gruesos” o “Fallo motor finos” así como hacer el reset del material, devolviendo el valor del nivel de material actual a la capacidad máxima del silo, tal y como se explicó en el punto 5 del presente documento.

Como apoyo visual, algunos de los fallos más habituales durante la fabricación (los fallos o vería de alguno de los actuadores y los aviso de nivel) se han complementado los textos de las alarmas con la iluminación de pilotos para identificar rápidamente la fuente del problema.

En el Anexo I se puede encontrar más información referida la navegación y empleo de la pantalla “Histórico de alarmas”

A continuación se detalla una tabla con todos los avisos programados para que aparezcan en el histórico de alarmas del HMI:

Nº error	Texto de aviso	Causa
1	Aviso_nivel_bajo_agua	Nivel de material bajo, rellenar material y pulsar "Reset material"
2	Fallo_nivel_agua	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
3	Fallo_motor_gruesos_agua	Tiempo de marcha excedido, comprobar el motor y sus protecciones
4	Fallo_motor_finos_agua	Tiempo de marcha excedido, comprobar el motor y sus protecciones
5	Fallo_valor_A_B_agua	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
6	Fallo_valor_M_N_agua	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
7	Fallo_peso_consigna_agua	Error en el valor introducido de la consigna del peso. Revisar el parámetro
8	Fallo_valor_capac_max_agua	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
9	Aviso_error_vertido_agua	Error en el vertido del material. Revisar y limpiar los actuadores
10	Aviso_nivel_bajo_vinagre	Nivel de material bajo, rellenar material y pulsar "Reset material"
11	Fallo_nivel_vinagre	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
12	Fallo_motor_gruesos_vinagre	Tiempo de marcha excedido, comprobar el motor y sus protecciones

13	Fallo_motor_finos_vinagre	Tiempo de marcha excedido, comprobar el motor y sus protecciones
14	Fallo_valor_A_B_vinagre	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
15	Fallo_valor_M_N_vinagre	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
16	Fallo_peso_consigna_vinagre	Error en el valor introducido de la consigna del peso. Revisar el parámetro
17	Fallo_valor_capac_max_vinagre	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
18	Aviso_error_vertido_vinagre	Error en el vertido del material. Revisar y limpiar los actuadores
19	Aviso_nivel_bajo_mostaza_blanca	Nivel de material bajo, rellenar material y pulsar "Reset material"
20	Fallo_nivel_mostaza_blanca	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
21	Fallo_motor_gruesos_mostaza_blanca	Tiempo de marcha excedido, comprobar el motor y sus protecciones
22	Fallo_motor_finos_mostaza_blanca	Tiempo de marcha excedido, comprobar el motor y sus protecciones
23	Fallo_valor_A_B_mostaza_blanca	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
24	Fallo_valor_M_N_mostaza_blanca	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
25	Fallo_peso_consigna_mostaza_blanca	Error en el valor introducido de la consigna del peso. Revisar el parámetro
26	Fallo_valor_capac_max_mostaza_blanca	Error en el valor introducido de la capacidad máxima. Revisar el parámetro

27	Aviso_error_vertido_mostaza_blanca	Error en el vertido del material. Revisar y limpiar los actuadores
28	Aviso_nivel_bajo_mostaza_negra	Nivel de material bajo, rellenar material y pulsar "Reset material"
29	Fallo_nivel_mostaza_negra	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
30	Fallo_motor_gruesos_mostaza_negra	Tiempo de marcha excedido, comprobar el motor y sus protecciones
31	Fallo_motor_finos_mostaza_negra	Tiempo de marcha excedido, comprobar el motor y sus protecciones
32	Fallo_valor_A_B_mostaza_negra	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
33	Fallo_valor_M_N_mostaza_negra	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
34	Fallo_peso_consigna_mostaza_negra	Error en el valor introducido de la consigna del peso. Revisar el parámetro
35	Fallo_valor_capac_max_mostaza_negra	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
36	Aviso_error_vertido_mostaza_negra	Error en el vertido del material. Revisar y limpiar los actuadores
37	Aviso_nivel_bajo_cúrcuma	Nivel de material bajo, rellenar material y pulsar "Reset material"
38	Fallo_nivel_cúrcuma	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
39	Fallo_motor_gruesos_cúrcuma	Tiempo de marcha excedido, comprobar el motor y sus protecciones
40	Fallo_motor_finos_cúrcuma	Tiempo de marcha excedido, comprobar el motor y sus protecciones

41	Fallo_valor_A_B_cúrcuma	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
42	Fallo_valor_M_N_cúrcuma	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
43	Fallo_peso_consigna_cúrcuma	Error en el valor introducido de la consigna del peso. Revisar el parámetro
44	Fallo_valor_capac_max_cúrcuma	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
45	Aviso_error_vertido_cúrcuma	Error en el vertido del material. Revisar y limpiar los actuadores
46	Aviso_nivel_bajo_páprika	Nivel de material bajo, rellenar material y pulsar "Reset material"
47	Fallo_nivel_páprika	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
48	Fallo_motor_gruesos_páprika	Tiempo de marcha excedido, comprobar el motor y sus protecciones
49	Fallo_motor_finos_páprika	Tiempo de marcha excedido, comprobar el motor y sus protecciones
50	Fallo_valor_A_B_páprika	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
51	Fallo_valor_M_N_páprika	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
52	Fallo_peso_consigna_páprika	Error en el valor introducido de la consigna del peso. Revisar el parámetro
53	Fallo_valor_capac_max_páprika	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
54	Aviso_error_vertido_páprika	Error en el vertido del material. Revisar y limpiar los actuadores

55	Aviso_nivel_bajo_sal	Nivel de material bajo, rellenar material y pulsar "Reset material"
56	Fallo_nivel_sal	Nivel de material cercano a cero, rellenar el material y pulsar "Reset material"
57	Fallo_motor_gruesos_sal	Tiempo de marcha excedido, comprobar el motor y sus protecciones
58	Fallo_motor_finos_sal	Tiempo de marcha excedido, comprobar el motor y sus protecciones
59	Fallo_valor_A_B_sal	Valores de A o B incorrectos, A debe ser mayor que B y estar entre cero y cien
60	Fallo_valor_M_N_sal	Valores de M o N incorrectos, A debe ser mayor que B y estar entre cero y cien
61	Fallo_peso_consigna_sal	Error en el valor introducido de la consigna del peso. Revisar el parámetro
62	Fallo_valor_capac_max_sal	Error en el valor introducido de la capacidad máxima. Revisar el parámetro
63	Aviso_error_vertido_sal	Error en el vertido del material. Revisar y limpiar los actuadores
64	Fallo valores iniciales control temperatura	Error. Sobretemperatura de alarma debe ser mayor que la banda de control
65	Alerta! exceso de temperatura en el triturador	Temperatura máxima rebasada. Revisar los componentes de la instalación
66	Error vertido conjunto	Se ha producido error de vertido en alguno de los materiales

Tabla 6 Fallos y avisos mostrados en el HMI

7.6.4 visualización DEL ESTADO DE LOS SILOS

La figura 44 muestra la pantalla “Control silos”:

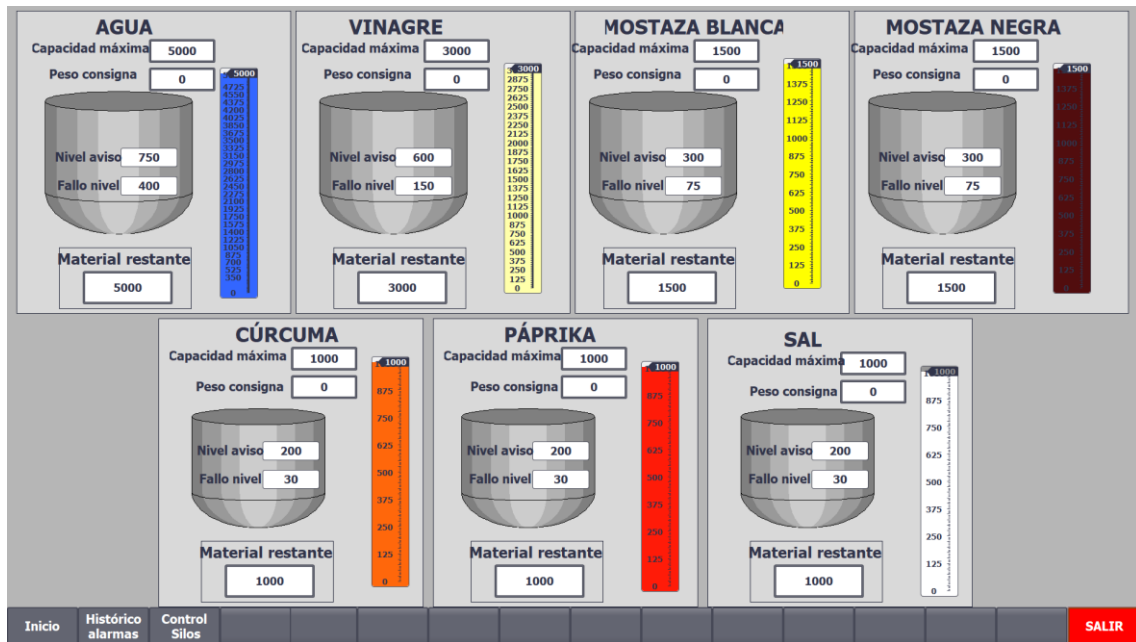


Figura 44 Pantalla monitorización estado de los silos

Esta pantalla es puramente informativa, desde aquí se podrá consultar el nivel de material restante, la capacidad máxima de los silos, además de visualizar si hay fallo o aviso de nivel, y a partir de que valores se producirán estos. Si se ha iniciado el modo automático, permitirá consultar el peso consigna asociado a cada material que se ha cargado de la receta.

En el Anexo I se puede encontrar más información referida la navegación y empleo de la pantalla “Control silos”

7.6.4 INTERFAZ MOVIMIENTOS MANUALES

Una vez realizada la correcta parametrización, eliminados los fallos y teniendo material en los silos, al escoger “Movimientos manuales” se accederá a la pantalla mostrada en la figura 45:

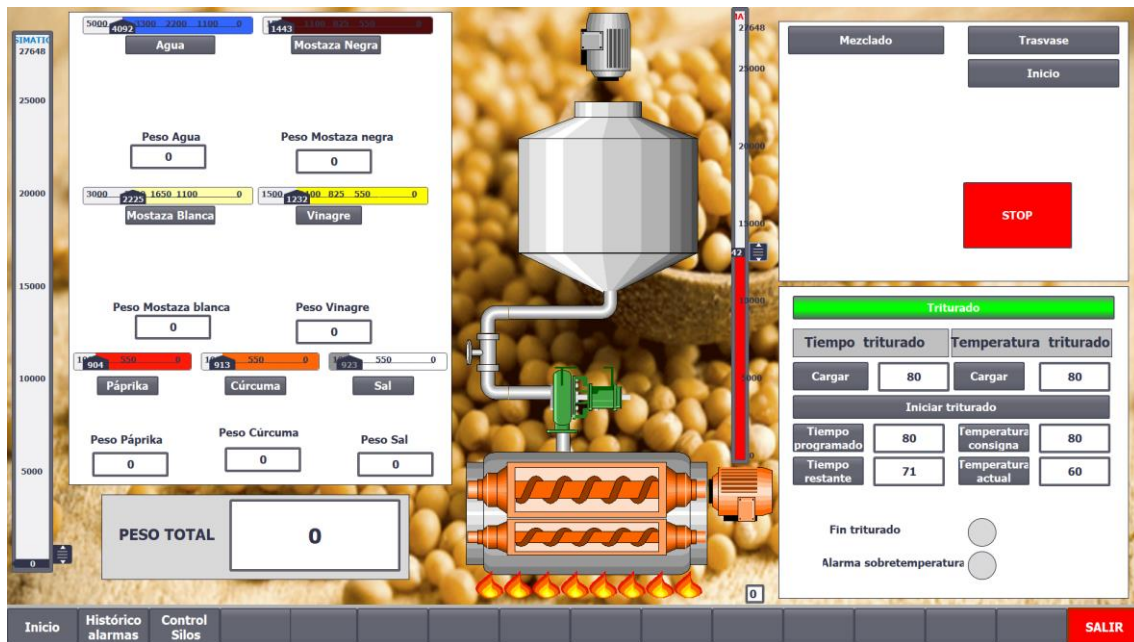


Figura 45 Pantalla movimientos manuales

En esta pantalla se tiene a disposición todos los controles necesarios para ejecutar un ciclo de fabricación completo activando manualmente cada etapa.

La mitad derecha de la figura pertenece al vertido y pesado de los ingredientes. En esta mitad se seleccionará primeramente la materia prima que se desea verter, y aparecerán los botones que permitirán encender los actuadores destinados a tal fin. En la parte izquierda se podrán activar las etapas de mezclado, trasvase y triturado.

En el Anexo I.I se puede encontrar más información referida la navegación y empleo de la pantalla “Modo manual”

7.6.5 INTERFAZ MODO AUTOMÁTICO

La figura 46 muestra la pantalla de selección de la receta a fabricar:

Nombre de entrada	Valor
AGUA	400
VINAGRE	200
MOSTAZA BLANCA	100
MOSTAZA NEGRA	0
CÚRCUMA	10
PÁPRKA	10
SAL	15
TIEMPO MEZCLADO	6
TIEMPO TRITURADO	20
TEMPERATURA CONSIGNA	95

Transferencia concluida

Ir a modo automático

Inicio Histórico alarmas Control Silos SALIR

Figura 46 Pantalla selección receta

Al pulsar sobre el botón de la figura 46 “Modo automático”, se dirige al operario a la selección de la receta, por lo que obligatoriamente antes de pasar a la pantalla de visualización del propio modo automático, se deberá de seleccionar una receta y cargarla al autómata. Una vez transferida, aparecerá un botón “Ir a modo automático”, que dirigirá al operario a la pantalla de control del estado del proceso en modo automático.

Por último, la figura 47 muestra la pantalla de la visualización del modo automático:

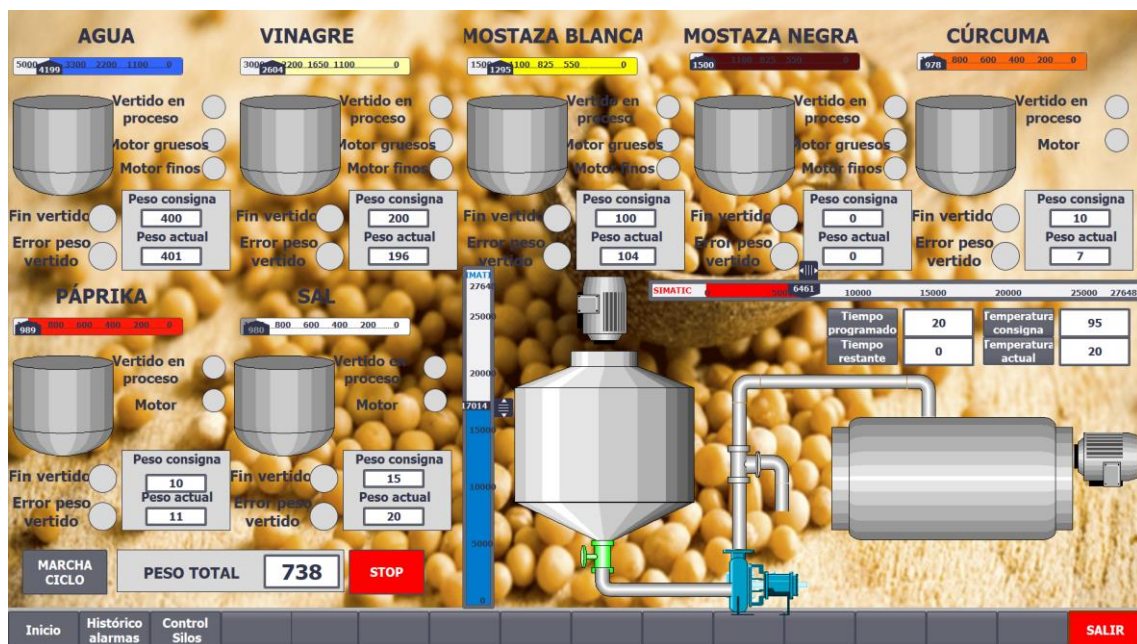


Figura 47 Pantalla modo automático

En esta pantalla, se podrán monitorizar todos los datos importantes del proceso en tiempo real, tales como el peso consigna de cada material y el peso vertido real, el peso total del lote, las cantidades restantes de material en cada silo, si el vertido se está produciendo o ha finalizado o el actuador que se encuentra encendido mediante la activación de diferentes pilotos y visualizaciones de actuadores para que el proceso se pueda seguir de un golpe de vista. También será observable la temperatura del cocido en el triturado así como el tiempo restante de triturado para finalizar el proceso.

En el Anexo I.II se puede encontrar más información referida la navegación y empleo de la pantalla “Modo automático”

8.- CONCLUSIONES

A través de la realización del proyecto se ha conseguido automatizar la producción de la salsa de mostaza. Esta automatización ha permitido conseguir la repetibilidad que se andaba buscando para el posicionamiento de la salsa en el mercado.

Se han escogido el software y materiales necesarios de entre un amplio abanico de posibilidades, que permitieran automatizar el proceso de fabricación manteniendo los costes en unos valores razonables, y que a la vez que han permitido reducir tiempos de fabricación han conseguido reducir los errores atribuidos a la parte humana de la producción.

Se ha desarrollado e implementado un programa de control mediante el empleo combinado de funciones genéricas que vienen integradas dentro del TIA Portal, con funciones complejas desarrolladas específicamente para este proyecto en lenguaje de alto nivel, cuya estructura y diseño permitirá reutilizarlas en futuros proyectos similares, reduciendo los tiempos de diseño y programación.

La parte correspondiente al programa principal, susceptible a ser revisada o consultada en alguna ocasión por el personal de mantenimiento, se ha desarrollado en lenguaje Ladder o en escalera, que se asemeja a la lógica de contactos para facilitar la comprensión por parte del personal de la planta, al que se le han añadido las funciones propias creadas en SCL, las cuales se visualizarán como una caja cerrada con una serie de entradas y salidas que permitirá la comprensión al operario, si tener que estudiar el funcionamiento de la lógica interna de estas.

Se ha creado un tipo de dato para la parametrización de los silos, permitiendo gestionar de manera más eficiente la parametrización de los silos, así como reducir el tiempo de programación al declarar las variables de este tipo de dato una única vez, y poder ir empleándolo cuando ha sido necesario.

Se ha desarrollado un SCADA que permite al operario interactuar con el programa de control, y además supervisar todo el proceso productivo. Se ha incluido una pantalla que permita la parametrización, previo control de usuarios de personal autorizado, de los elementos físicos que componen la planta, de modo que estos se puedan modificar en caso necesario sin necesidad de acceder al programa del PLC.

Se ha implementado un histórico de alarmas en el SCADA, desde el que poder observar si hay algún componente en fallo, errores de parametrización y avisos por niveles de material bajo.

Se ha dotado a la planta con un modo manual desde el que hacer pruebas para nuevas formulaciones de producto, y de un modo automático, desde el que se selecciona la receta y al iniciar el proceso se ejecuta completamente de manera automática.

Bibliografía:

<https://unicrom.com/historia-del-plc-modicon-modbus/>

<https://www.automation.com/automation-news/article/the-dawn-of-the-programmable-logic-controller-plc>

<http://www.aldakin.com/automatizacion-industrial-robotica-claves-exito/>

[https://zulaco64.updog.co/Material Didactico/Control-OnOff Sistemas Control Controladores.pdf](https://zulaco64.updog.co/Material_Didactico/Control-OnOff_Sistemas_Control_Controladores.pdf)

<http://www.udb.edu.sv/udb/archivo/guia/electronica-ingenieria/automatas-programables/2015/i/guia9.pdf>

<https://www.agenciatributaria.es/AEAT.internet/Inicio/ Segmentos /Empresas y profesionales/Empresas/Impuesto sobre Sociedades/Periodos impositivos a partir de 1 1 2015/Base imponible/Amortizacion/Tabla de coeficientes de amortizacion lineal .shtmlç>

https://cache.industry.siemens.com/dl/files/216/109736216/att_879792/v1/WinCC_GeneralInfo_Installation_Readme_es-ES_es-ES.pdf

<https://www.infoplç.net/descargas/106-siemens/software-step7-tiaportal/2049-bloques-organización-ob-tia-portal-automatas-s7-1200-s7-1500>

<https://www.infoplç.net/descargas/106-siemens/software-step7-tiaportal/2287-tia-portal-estructura-datos>

<https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/advanced-programming-s7-1200/sce-051-201-scl-s7-1200-r1709-es.pdf>

[https://support.industry.siemens.com/cs/document/109479728/librería-con-funciones-generales-\(lgf\)-para-simatic-step-7-\(tia-portal\)-y-s7-1200-s7-1500?dti=0&lc=es-WW](https://support.industry.siemens.com/cs/document/109479728/librería-con-funciones-generales-(lgf)-para-simatic-step-7-(tia-portal)-y-s7-1200-s7-1500?dti=0&lc=es-WW)

<https://www.youtube.com/watch?v=BDKWCvQZebo>

<https://www.youtube.com/watch?v=wUL-ALcTOA>

PLIEGO DE CODICIONES

CONTROL Y VISUALIZACIÓN DEL PROCESO DE ELABORACIÓN DE LA SALSA DE MOSTAZA

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019

ÍNDICE

1.- GENERALIDADES.....	1
1.1 DESARROLLO DEL PROGRAMA.....	1
1.2 plazo de ejecución	1
1.3 necesidades a cubrir por el programa	1
1.4 trabajos adicionales	2
1.5 modificaciones sobre el programa acordado originalmente.....	2
1.6 conservación del programa	2
2.- recepción del programa	3
2.1 derechos y licencias	3
2.2 garantía.....	3
3.- condiciones económicas	4
3.1 revisión de costes	4
3.2 penalizaciones	4
3.3 responsabilidad adquirida	4
4.- Rescisión del contrato	5
4.1 liquidación por rescisión.....	5

1.- GENERALIDADES

El pliego desarrollado a lo largo de las siguientes páginas tiene como finalidad establecer a la empresa responsable del proyecto la ejecución y alcance del trabajo.

Este trabajo consistirá en desarrollar un programa capaz de automatizar el proceso productivo de la salsa de mostaza dotando a la planta de capacidad de fabricación de forma autónoma.

Para ello, la empresa responsable del proyecto se encargará también de la elección del hardware y software necesario, así como la puesta en marcha de la producción.

1.1 DESARROLLO DEL PROGRAMA

El desarrollo del programa se iniciará en la fecha establecida entre contratista y contratante, especificada el contrato realizado para la ejecución del proyecto.

El contratista notificará el inicio del desarrollo con fines informativos para el seguimiento de los plazos a la parte contratante.

1.2 PLAZO DE EJECUCIÓN

El contratista prestará el servicio definido por el contrato, suscrito por las partes intervinientes, en el plazo que en este se haya estipulado, con un margen de una semana ante posibles imprevistos por parte de la empresa contratista.

La empresa contratante se compromete a notificar por escrito la detección de posibles alteraciones en los plazos originadas por factores externos a la empresa contratista, que pudiesen alterar las fechas estimadas inicialmente, descargando así de responsabilidad ante posibles retrasos a la empresa contratista.

1.3 NECESIDADES A CUBRIR POR EL PROGRAMA

Las necesidades a cubrir por el programa se definirán previa firma del contrato por parte del contratante y contratista.

El contratista se ceñirá a los puntos desarrollados en el contrato y cada modificación requerida por el contratante deberá de dejarse en constancia por escrito con la ampliación de los plazos acordes a la gravedad de las modificaciones, alterando por tanto los plazos originales, siendo esto motivo inhabilitante de futuras reclamaciones por parte del contratante ante el retraso de la ejecución del proyecto frente a las fechas originales.

1.4 TRABAJOS ADICIONALES

La empresa contratista se compromete a realizar los trabajos que no viniesen específicamente definidos en el contrato siempre y cuando el motivo de la necesidad de los mismos venga dado por un mal funcionamiento en el programa detectado durante el empleo intensivo del mismo durante el proceso productivo de la empresa contratante, así como por unas definiciones de hardware que no se ajusten a las necesidades de la planta.

1.5 MODIFICACIONES SOBRE EL PROGRAMA ACORDADO ORIGINALMENTE

La empresa contratista se compromete a realizar en la medida de lo posible los cambios requeridos por el contratante, siempre previo acuerdo de términos y plazos de ejecución. Para estas modificaciones se deberá llegar a un acuerdo entre partes de modo que se fijen las cantidades adicionales a remunerar en el contrato, dejando por escrito constancia y siendo necesariamente firmadas por las partes intervinientes.

1.6 CONSERVACIÓN DEL PROGRAMA

La empresa contratista se compromete a conservar el programa descargado en el PLC de la planta durante al menos dos años posteriores a la fecha de ejecución del proyecto.

La empresa contratante se compromete a no realizar modificaciones en el programa sin consultar previamente a la empresa contratista. Si esto no se cumpliera, se descarga de responsabilidades a la empresa contratista por los posibles daños, materiales o humanos, que pudiesen ocasionarse.

2.- RECEPCIÓN DEL PROGRAMA

Una vez realizada la fase de diseño, se acordará una fecha entre contratante y contratista para realizar la carga del programa en planta, y acordar las posibles modificaciones requeridas por parte del contratante, a las cuales se les aplicarán las condiciones definidas en los puntos previos del presente pliego de condiciones.

2.1 DERECHOS Y LICENCIAS

Una vez realizada la recepción del programa, se otorga licencia para el uso del mismo a la empresa contratante, únicamente para la instalación acordada. El contratante no podrá vender, copiar o descargar el programa en otras máquinas, propias o de terceros, ya sea con o sin ánimo de lucro.

La empresa contratista se reserva el derecho de tomar acciones legales contra el contratante si se comprobara la infracción de este punto.

El presente pliego de condiciones se considerará parte indispensable de la documentación necesaria para los derechos de explotación del programa desarrollado.

2.2 GARANTÍA

El plazo de garantía se establece en un periodo de dos años desde la puesta en marcha en las instalaciones de la empresa contratante. Durante este periodo, será la empresa contratista la encargada de resolver posibles defectos de programación o manipulaciones del programa. No se cubrirá por parte de la garantía peticiones de la empresa contratante que modifiquen el proceso productivo tras la aprobación del programa en su recepción.

Si ante cualquier anomalía de funcionamiento se detecta la modificación del programa sin consulta a la empresa contratista, se invalidará la garantía.

3.- CONDICIONES ECONÓMICAS

La cuantía a abonar por el contratante al contratista vendrá especificada en el contrato.

En dicho documento se detallarán las cantidades y plazos en los que se realizarán los abonos.

3.1 REVISIÓN DE COSTES

La empresa contratista se reserva el derecho de modificar las tarifas horarias en cuanto a lo que las modificaciones posteriores a la recepción del programa se refieren, estando obligada a presentar un nuevo presupuesto detallando las modificaciones y los costes asociados a ellas, y que deberá de ser aprobado por el contratante para que la empresa contratista inicie los trabajos requeridos.

3.2 PENALIZACIONES

Se podrán establecer penalizaciones tabuladas y aceptadas por las partes intervinientes a la hora de firmar el contrato, a la empresa contratista ante retrasos no justificados o acordados.

3.3 RESPONSABILIDAD ADQUIRIDA

La empresa contratista se responsabiliza de la ejecución del programa en las condiciones definidas en el contrato por las partes intervinientes. Ante una mala ejecución del programa, el contratista se hará cargo de las reparaciones y modificaciones oportunas en caso de necesitarlas.

El contratista se hace responsable de los posibles errores de programación cometidos por el personal a cargo de la misma, comprometiéndose a subsanar los fallos que pudieran encontrarse por este motivo, sin que suponga un coste añadido a la empresa contratante.

La empresa contratista será la encargada de proporcionar los EPI's específicos que pudieran necesitarse en planta, que no sean los de uso habitual (botas, casco o protecciones auditivas), tales como mascarillas, ropa protectora específica etc.

4.- RESCISIÓN DEL CONTRATO

La empresa contratista podrá rescindir el contrato cuando se den alguna de las siguientes condiciones y no se haya llegado a un acuerdo con la empresa contratante:

- Modificación sustancial del proyecto inicial sin renegociación de los términos, tanto económicos como temporales, que pueda suponer un sobrecoste importante con el perjuicio asociado a la empresa contratista.
- Suspensión de los pagos acordados en los plazos de pago por parte de la empresa contratante, reservándose el contratista el derecho de emprender acciones legales.
- Subcontratar bien una parte, bien la totalidad del desarrollo del programa a terceros sin consulta ni autorización expresa por parte del contratista.
- Quiebra del contratista.

4.1 LIQUIDACIÓN POR RESCISIÓN

Si por alguna razón tuviese lugar una de las causas de rescisión anteriormente mencionadas, la empresa contratante tendrá la obligación de abonar las cuantías establecidas acorde a lo avanzado de los plazos de entrega, debiendo abonar por tanto, hasta el último plazo entregado por la empresa contratista.

PRESUPUESTO

CONTROL Y VISUALIZACIÓN DEL PROCESO DE ELABORACIÓN DE LA SALSA DE MOSTAZA

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019

ÍNDICE

1.- Presupuesto	1
1.1 Hardware.....	2
1.2 Software.....	3
1.3 Mano de obra.....	3
1.4 Montante total.....	4

ÍNDICE DE TABLAS

Tabla 1 Presupuesto Hardware	2
Tabla 2 Presupuesto Software.....	3
Tabla 3 Presupuesto Mano de Obra	3
Tabla 4 Montante total.....	4
Tabla 5 Montante total, IVA incluido	4

ÍNDICE DE FIGURAS

Figura 1 Extracto tabla amortización Agencia Tributaria. Fuente:

https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/

.....	2
-------	---

1.- PRESUPUESTO

Para el desarrollo del presupuesto del proyecto, se ha dividido en tres partidas, puesto que se pueden separar los gastos en tres grupos diferenciados:

- **Hardware:** En este apartado se incluirán los precios del hardware escogido en el punto 6 del presente documento, necesario para la implementación de la automatización en el proyecto, así como la amortización de los equipos físicos necesarios para la instalación del hardware de programación.
- **Software:** En esta partida se incluirá el coste proporcional para la amortización de las licencias necesarias para realizar la programación del proyecto.
- **Mano de obra:** En este apartado se incluirán los costes asociados a la parte de la intervención humana a la hora de realizar el proyecto, es decir, la imputación de horas necesarias para la fase de diseño, puesta en marcha y generación de la documentación necesaria.

1.1 HARDWARE

Para la amortización de equipos para procesos de información, la Agencia Tributaria estima coeficiente lineal máximo de amortización del 25%, lo que se traduce en que el ordenador empleado se amortizará a lo largo de cuatro años.

Tipo de elemento	Coefficiente lineal máximo	Período de años máximo
Equipos electrónicos e informáticos. Sistemas y programas		
Equipos electrónicos	20%	10
Equipos para procesos de información	25%	8
Sistemas y programas informáticos	33%	6

Figura 1 Extracto tabla amortización Agencia Tributaria. Fuente:

https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/

Por lo que con un PC Lenovo ThinkPad E580 valorado en 1209,99 € y teniendo en cuenta que se trabajará 1840 horas laborales por año, se obtiene un coste imputable al proyecto de 49,32€.

$$\text{Coste imputable} = \frac{\text{Coste total ordenador}}{n^{\circ} \text{ de horas trabajadas al año}} \times n^{\circ} \text{ horas proyecto} \times 0,25$$

$$\text{Coste imputable} = \frac{1209,99}{1840} \times 300 \times 0,25 = 49,32\text{€}$$

PRESUPUESTO HARDWARE				
Concepto	Referencia	Cantidad	Coste unitario	Coste
Amortización PC: Lenovo ThinkPad E580	-	1,00	49,32	49,32
PC para el cliente: HP Pavilion	690-0302ns	1,00	699,00	699,00
Siemens S7-1200, CPU 1214C, COMPACT CPU, AC/DC/RLY	6ES7 214-1BG40-0XB0	1,00	293,46	293,46
Siemens S7-1200, DIGITAL I/O SM 1223, 16 DI / 16 DO RLY	6ES7223-1PL32-0XB0	1,00	213,16	213,16
Total				1254,94

Tabla 1 Presupuesto Hardware

1.2 SOFTWARE

Para el software se ha procedido del mismo modo. En este caso hay que amortizar la licencia del software de programación SIMATIC S7, valorada en 1417.49 € así como la licencia del sistema operativo del ordenador, en este caso el Windows 10 PRO valorada en 259 €. Al igual que para el hardware, se ha tenido en cuenta que se trabajará 1840 horas laborales por año, por lo que se obtiene un coste imputable al proyecto de 79.17€ para la licencia y de 13.94 € para la amortización del sistema operativo.

$$\text{Coste imputable} = \frac{\text{Coste licencia}}{n^{\circ} \text{ de horas trabajadas al año}} \times n^{\circ} \text{ horas proyecto} \times 0,33$$

$$\text{Coste imputable licencia WinCC} = \frac{1417.49}{1840} \times 300 \times 0,33 = 79.17 \text{ €}$$

$$\text{Coste imputable Licencia Windows 10 PRO} = \frac{259}{1840} \times 300 \times 0,33 = 13.94 \text{ €}$$

PRESUPUESTO SOFTWARE				
Concepto	Referencia	Cantidad	Coste unitario	Coste
WinCC Basic a instalar en PC del cliente	6AV2100-0AA05-0AA5	1,00	103,00	103,00
Amortización Licencia Windows 10 PRO	-	1,00	13,94	13,94
Amortización licencia SIMATIC Siemens	6ES7822-1AE04-0YA5	1,00	79,17	79,17
Total				196,11

Tabla 2 Presupuesto Software

1.3 MANO DE OBRA

Para el presupuesto de la mano de obra, se han considerado meses de 4 semanas laborales, con 40 horas semanales:

PRESUPUESTO MANO DE OBRA				
Concepto	Duración (meses)	Cantidad horas	Coste unitario	Coste
Desarrollo programa	1,1	176,00	30,00	5280,00
Puesta en marcha	0,50	80,00	30,00	2400,00
Documentación generada: Manual de uso	0,2	32,00	30,00	960,00
Formación operario planta	0,08	12,00	30,00	360,00
Total				9000,00

Tabla 3 Presupuesto Mano de Obra

1.4 MONTANTE TOTAL

El coste del proyecto será pues la suma de los totales de las tablas 2, 3 y 4. Esta suma será la cantidad total del presupuesto de ejecución material (PEM), a lo que hay que añadir los gastos generales (GG) y el beneficio industrial (BI). Con la suma de estas tres partidas se ha obtenido el presupuesto de ejecución del contrato (PEC).

$$PEC = PEM + 13\% PEM + 6\% PEM \text{ antes del IVA}$$

PRESUPUESTO HARDWARE	1254,94
PRESUPUESTO SOFTWARE	196,11
PRESUPUESTO MANO DE OBRA	9000,00
GASTOS GENERALES (13%)	1358,64
BENEFICIO INDUSTRIAL (6%)	627,06
TOTAL PRESUPUESTADO	12436,75

Tabla 4 Montante total

El presupuesto para la automatización del proceso asciende a doce mil cuatrocientos treinta y seis euros con setenta y cinco céntimos sin IVA.

La tabla 6 muestra el montante total, impuestos incluidos:

PRESUPUESTO HARDWARE	1254,94
PRESUPUESTO SOFTWARE	196,11
PRESUPUESTO MANO DE OBRA	9000,00
GASTOS GENERALES (13%)	1358,64
BENEFICIO INDUSTRIAL (6%)	627,06
IVA (21%)	2611,72
TOTAL PRESUPUESTADO	15048,47

Tabla 5 Montante total, IVA incluido

Por lo que el presupuesto final para la automatización del proceso asciende a quince mil cuarenta y ocho euros con cuarenta y siete céntimos IVA incluido.

ESQUEMAS ELÉCTRICOS

CONTROL Y VISUALIZACIÓN DEL PROCESO DE ELABORACIÓN DE LA SALSA DE MOSTAZA

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019



Tel.

Empresa/cliente

Descripción de proyecto

ESQUEMAS CABLEADO PLC

Número de diseño

CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN SALSA DE MOSTAZA

Comisión

Fabricante (empresa)

Circuito

Nombre de proyecto

TFG

Producto

IR_CONTROL_2019

Tipo

SIEMENS CPU 1214

Lugar de instalación

Responsable del proyecto

Particularidad de pieza

Creado 08/05/2019

Modificado 12/07/2019

de (abreviatura) W7_64

Número de páginas 23

		Fecha	12/07/2019			Portada				= CTRL
		Resp.	W7_64							+ GENERAL
		Probado		ESQUEMAS CABLEADO PLC						
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por			CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN SALSA DE MOSTAZA		Página 1 / 23

MODIFICACIONES

--	--	--	--	--	--	--	--	--	--

		Fecha		12/07/2019				HOJA DE MODIFICACIONES		CGB_7000080-1		= CTRL	
		Resp.		W7_64								+ GENERAL	
		Probado				ESQUEMAS CABLEADO PLC						CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por							Página 3 / 23	

COLORES DE CONDUCCION

POTENCIA		
FASE R/L1:	BN	NEGRO
FASE S/L2:	GY	NEGRO
FASE T/L3:	BK	NEGRO
NEUTRO N:	TQ	AZUL CLARO
CONTROL		
C.Alterna 230V:		
ACTIVO	RD	ROJO
COMUNES	WH	BLANCO
C.Continua 24V:		
POSITIVO	BU	AZUL
NEGATIVO	BU	AZUL
TENSION DE ALIMENTACION EXTERNA/PERMANENTE	BU	AZUL
CONTACTOS LIBRE DE POTENCIAL:	OG	NARANJA
PE	GNYE	VERDE/AMARILLO

MEDIDA DEL CABLE

Sección mínima cable de potencia	2.5 mm ²
Sección mínima cable de mando/control	1.5 mm ²
Sección mínima cable de entradas/salidas	0.75 mm ²

OBSERVACIONES

NC = NO CABLEADO

COLORES DE CONDUCTORES

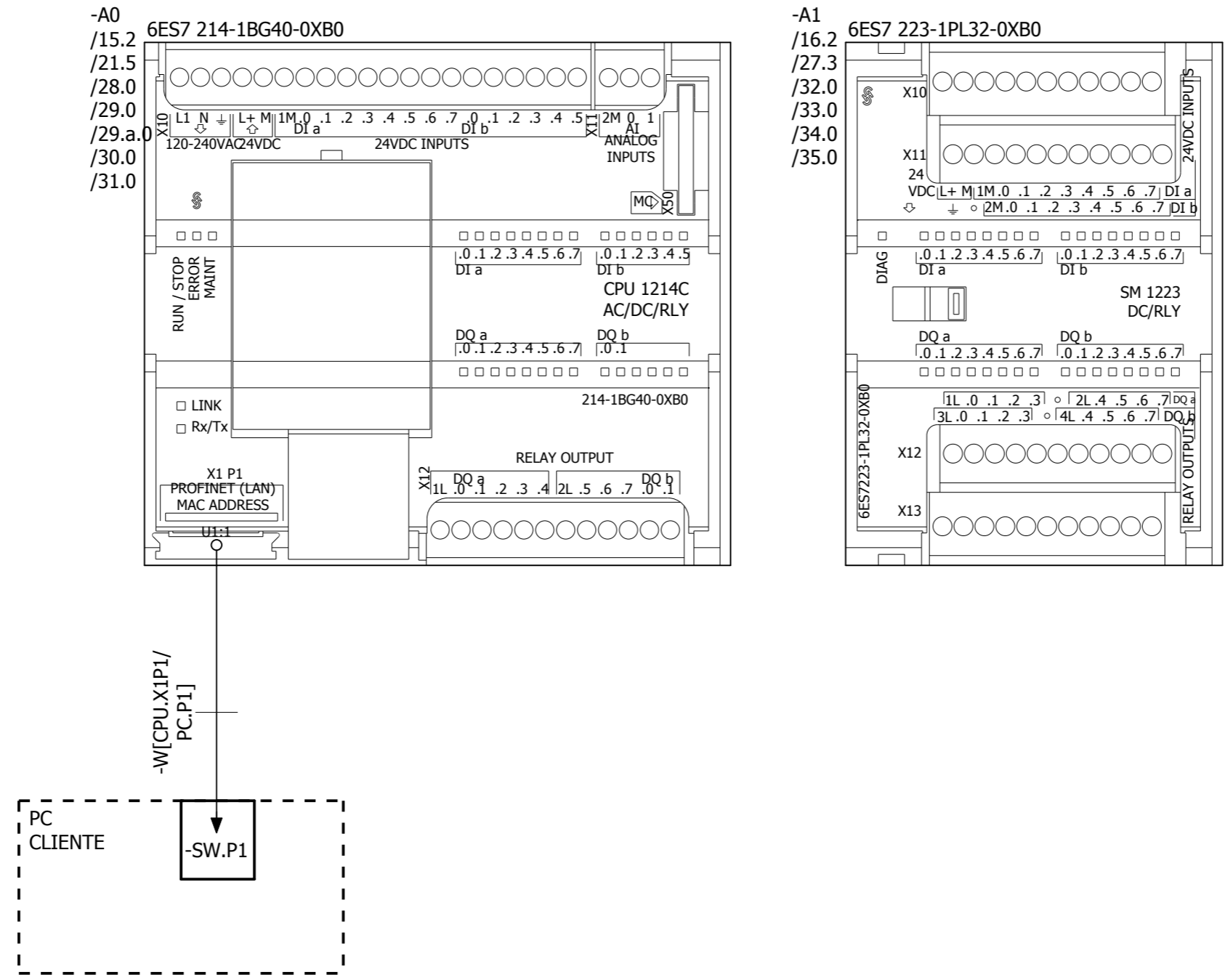
	IEC 757
NEGRO	BK
MARRON	BN
ROJA	RD
NARANJADO	OG
ARMARILLO	YE
VERDE	GN
AZUL	BU
VIOLETA	VT
GRIS	GY
BLANCO	WH
ROSADO	PK
ORO	GD
TURQUÖ	TQ
PLATA	SR
VERDE/AMARILLO	GNYE

ABREVIADO DE TIPO DE CABLE

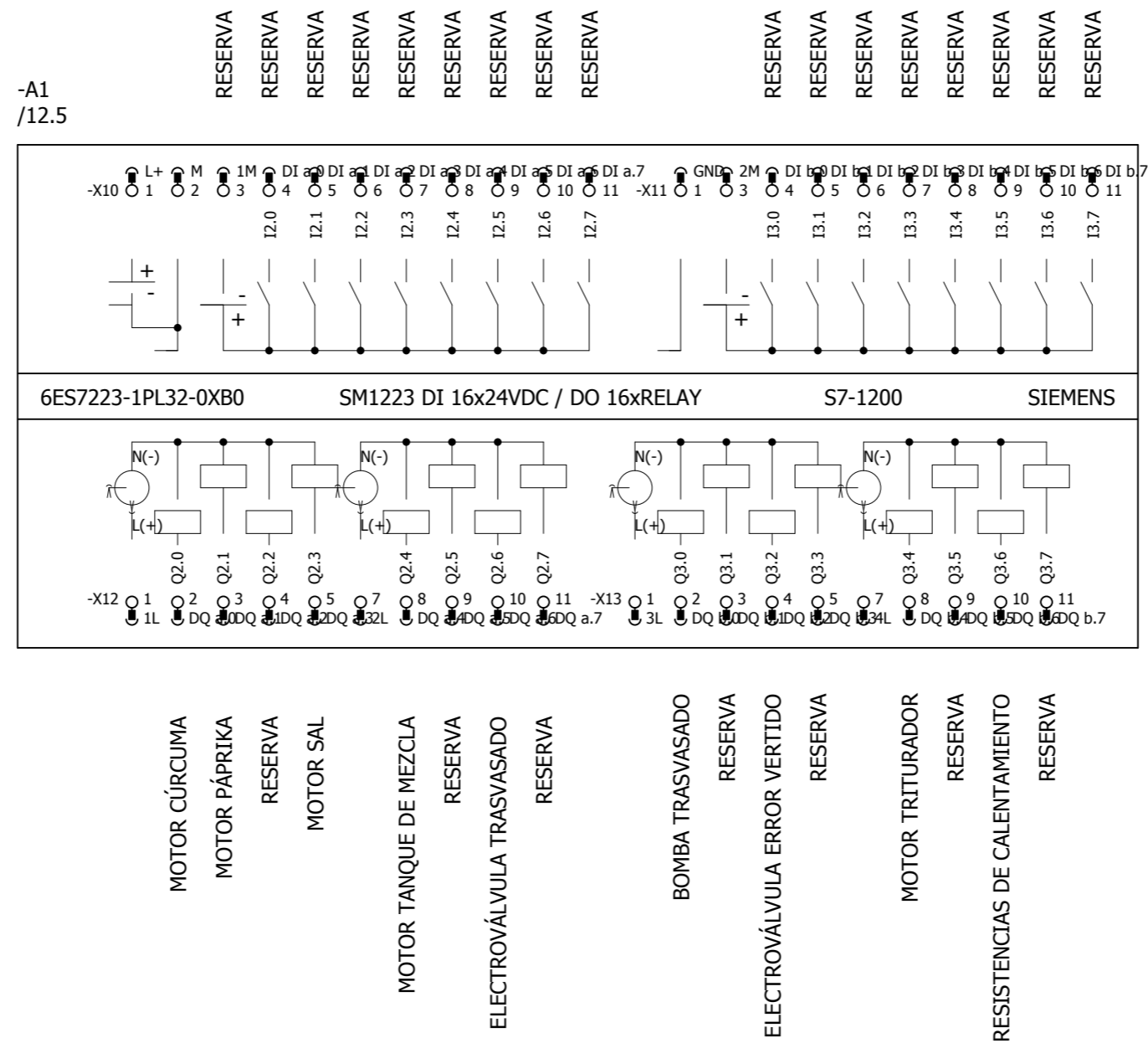
PVC CABLES DE CONTROL	
(N)YSLY	CONDUCTORES COLORES SIN VERDE/AMARILLO
(N)YSLY-J	CONDUCTORES COLORES CON VERDE/AMARILLO
(N)YSLY-Z	CONDUCTORES NUMERICO SIN VERDE/AMARILLO
(N)YSLY-JZ	CONDUCTORES NUMERICO CON VERDE/AMARILLO
PVC CABLES DE CONTROL CON Y BLINDAJE	
(N)YSLYCY	CONDUCTORES COLORES SIN VERDE/AMARILLO
(N)YSLYCY-J	CONDUCTORES COLORES CON VERDE/AMARILLO
(N)YSLYCY-Z	CONDUCTORES NUMERICO SIN VERDE/AMARILLO
(N)YSLYCY-JZ	CONDUCTORES NUMERICO CON VERDE/AMARILLO
PVC CABLES DE CONTROL CON Y BLINDAJE DE ACERO	
(N)YSLYSY	CONDUCTORES COLORES SIN VERDE/AMARILLO
(N)YSLYSY-J	CONDUCTORES COLORES CON VERDE/AMARILLO
(N)YSLYSY-Z	CONDUCTORES NUMERICO SIN VERDE/AMARILLO
(N)YSLYSY-JZ	CONDUCTORES NUMERICO CON VERDE/AMARILLO
CABLE DE ALTO FLEXIBLE	
(N)*-F	CONDUCTORES COLORES SIN VERDE/AMARILLO
(N)*-J-F	CONDUCTORES COLORES CON VERDE/AMARILLO
(N)*-Z-F	CONDUCTORES NUMERICO SIN VERDE/AMARILLO
(N)*-JZ-F	CONDUCTORES NUMERICO CON VERDE/AMARILLO
OTROS	
H07RN-F	CABLE MUY RESISTENTE, CON CUBIERTA DE GOMA
LiYY	CONDUCTOR FLEXIBLE CON PVC ISOLATION
LiYY (TP)	ditto CON RETORICDO
LiYCY	CONDUCTOR FLEXIBLE CON PVC ISOLATION Y BLINDAJE
LiYCY (TP)	ditto CON RETORICDO

				Fecha	12/07/2019			INFORMACIONES DE INSTALACIÓN	CRGA_600199_0816	= CTRL
				Resp.	W7_64			CABLE Y CONDUCCIÓN		+ GENERAL
				Probado		ESQUEMAS CABLEADO PLC		CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN		
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por			Página 4 / 23		

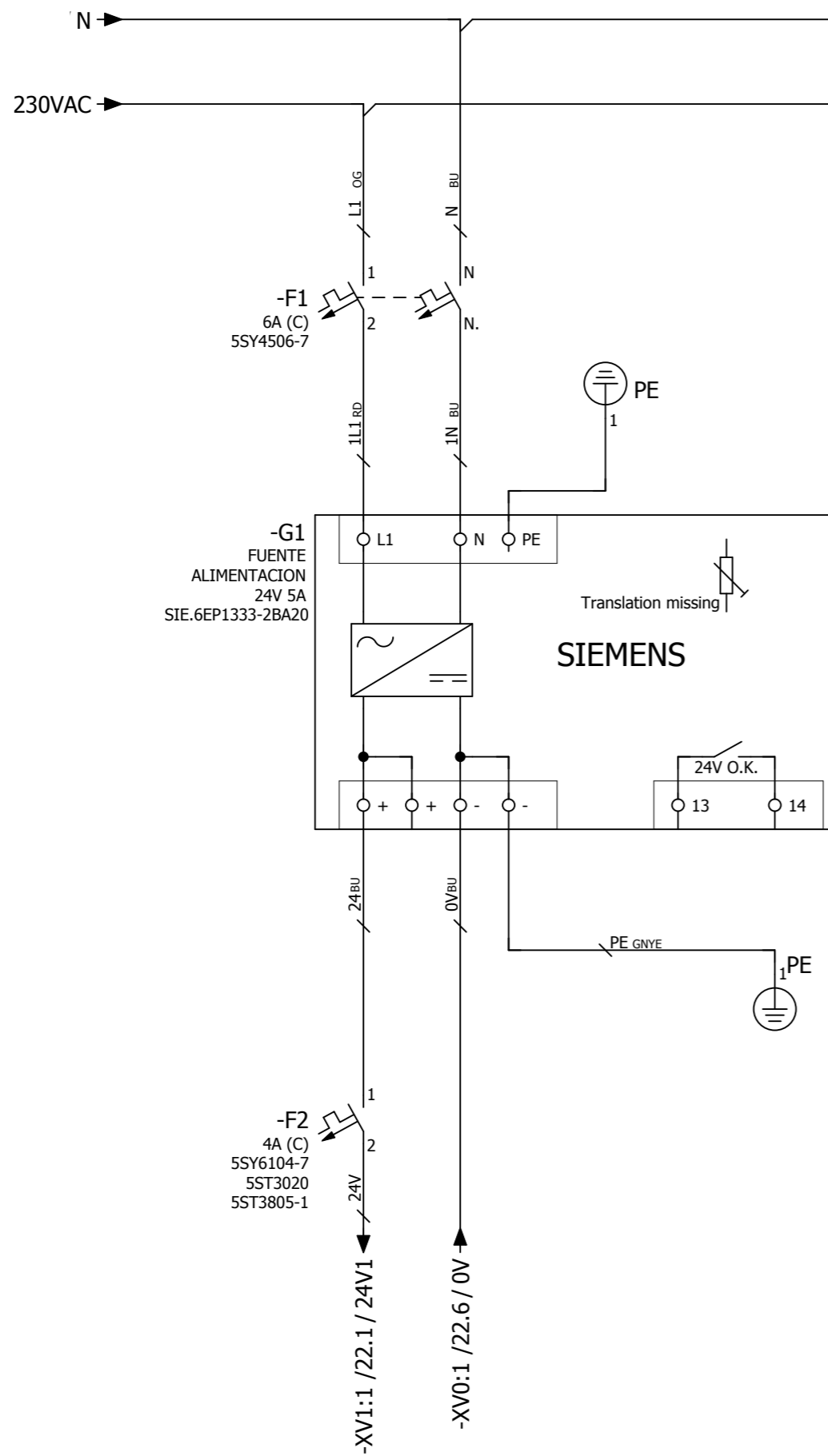
LAYOUT PLC



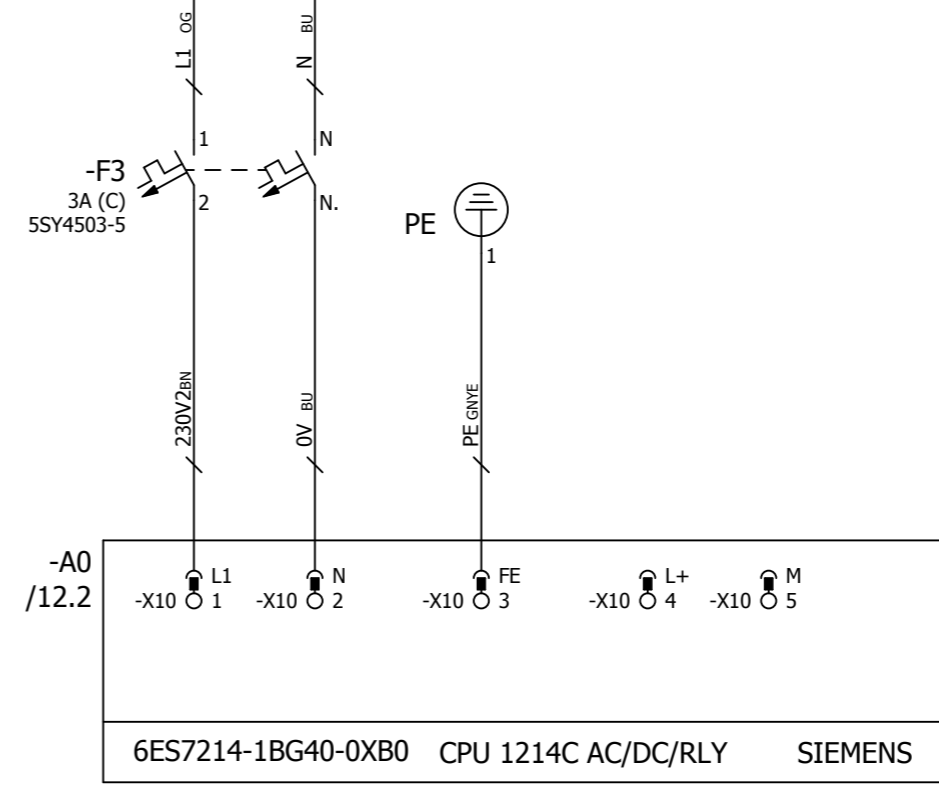
RESUMEN ENTRADAS/SALIDAS DIGITALES AMPLIACIÓN 1



		Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	RESUMEN ENTRADAS/SALIDAS AMP PLC	DIC-PAL-2016_v2	= CTRL
		Resp.	W7_64				+ ARM
		Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN SAUSA DE MOSTRA	
						Página 7 / 23	

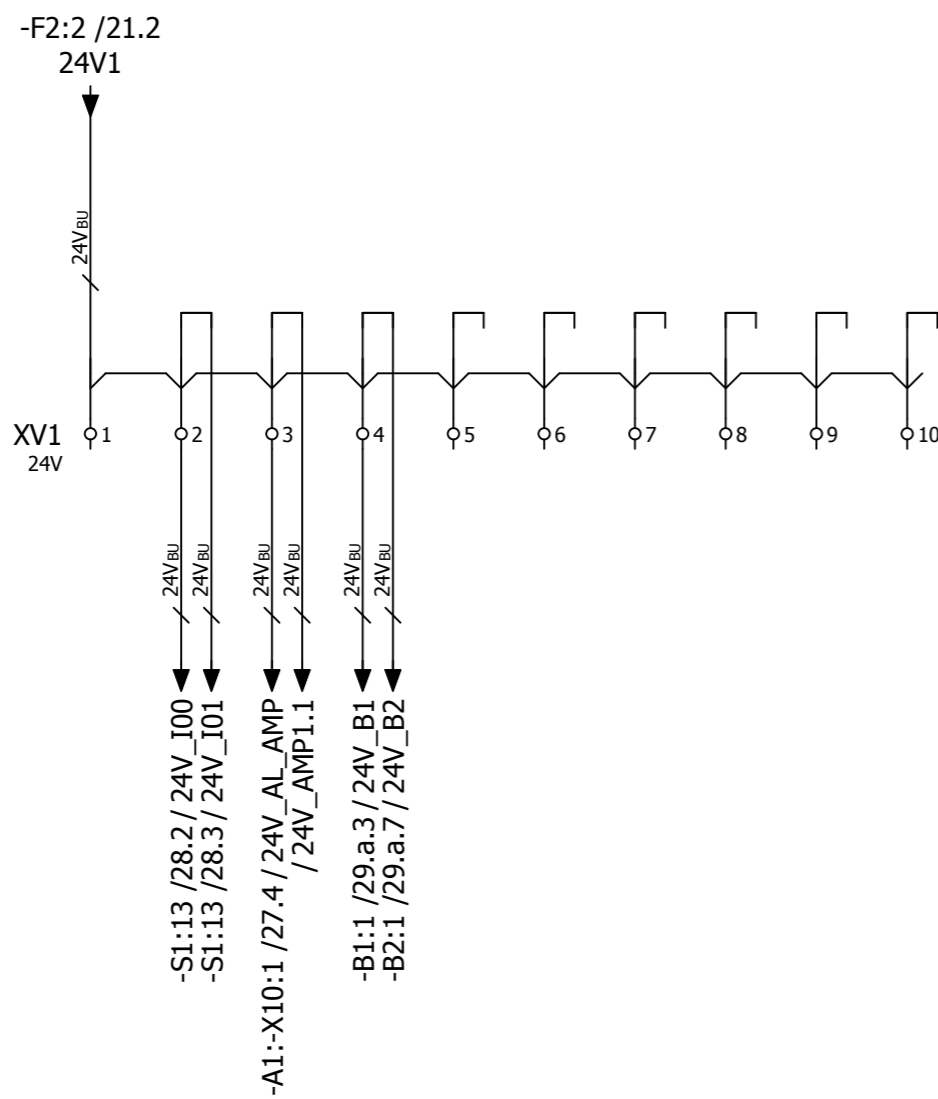


Distribución 24V DC

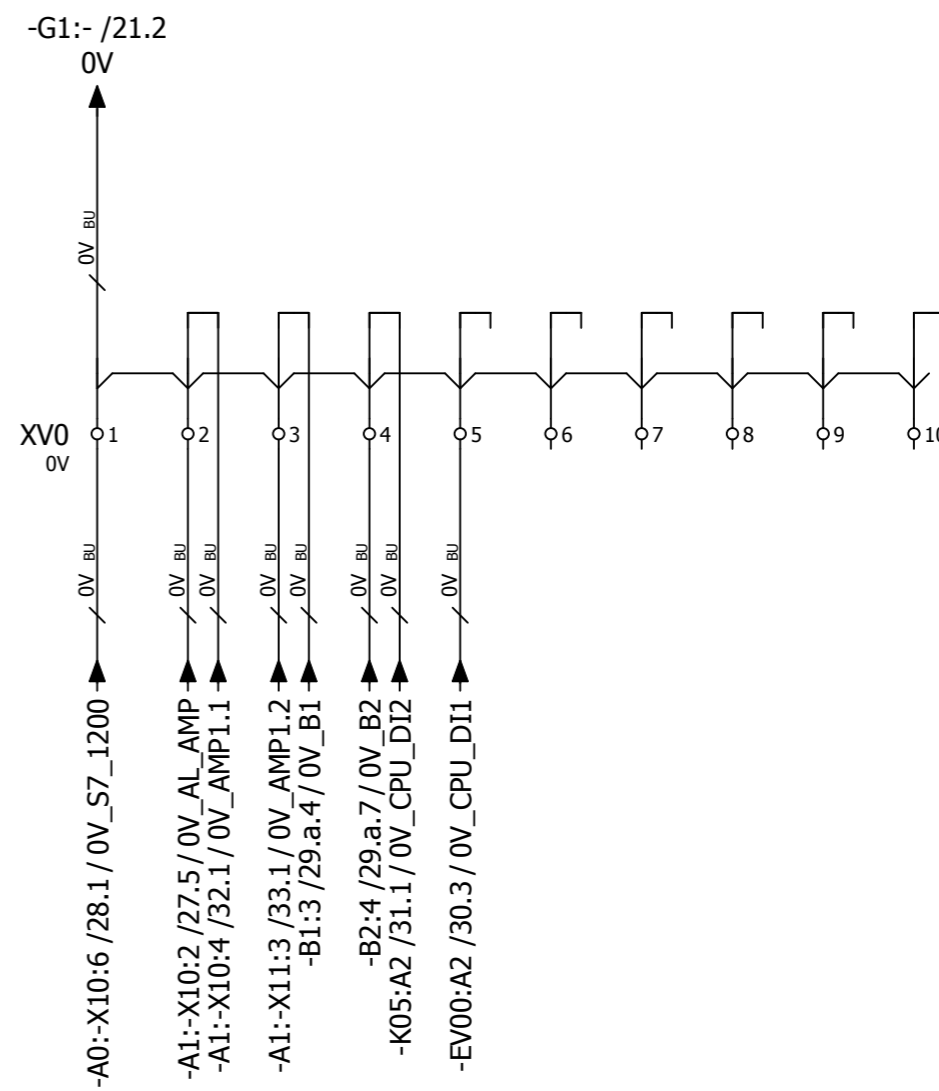


Alimentación PLC

Fecha		12/07/2019		FUENTE ALIMENTACION		CGB_7000080-1		= CTRL	
Resp.		W7_64						+ ARM	
Probado				ESQUEMAS CABLEADO PLC				CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por			Página 8 / 23	

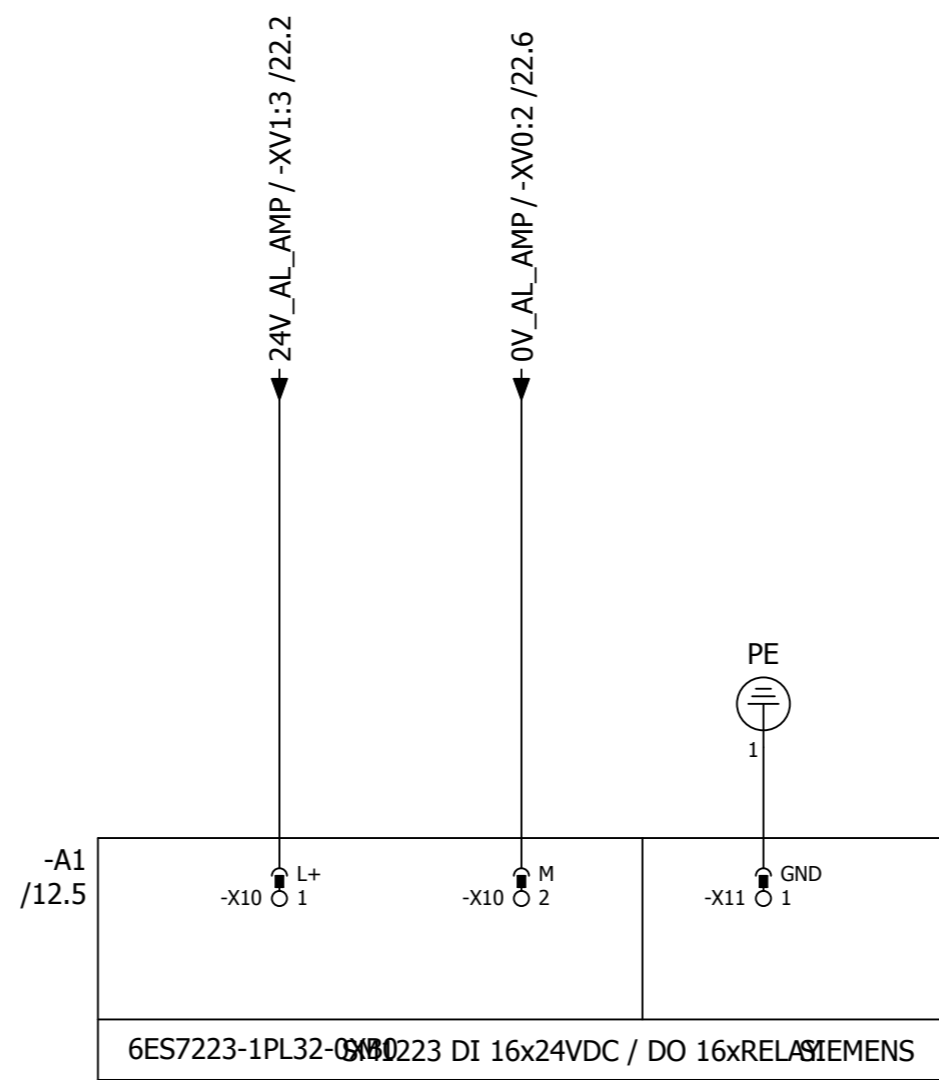


DISTRIBUCIÓN 24 VDC

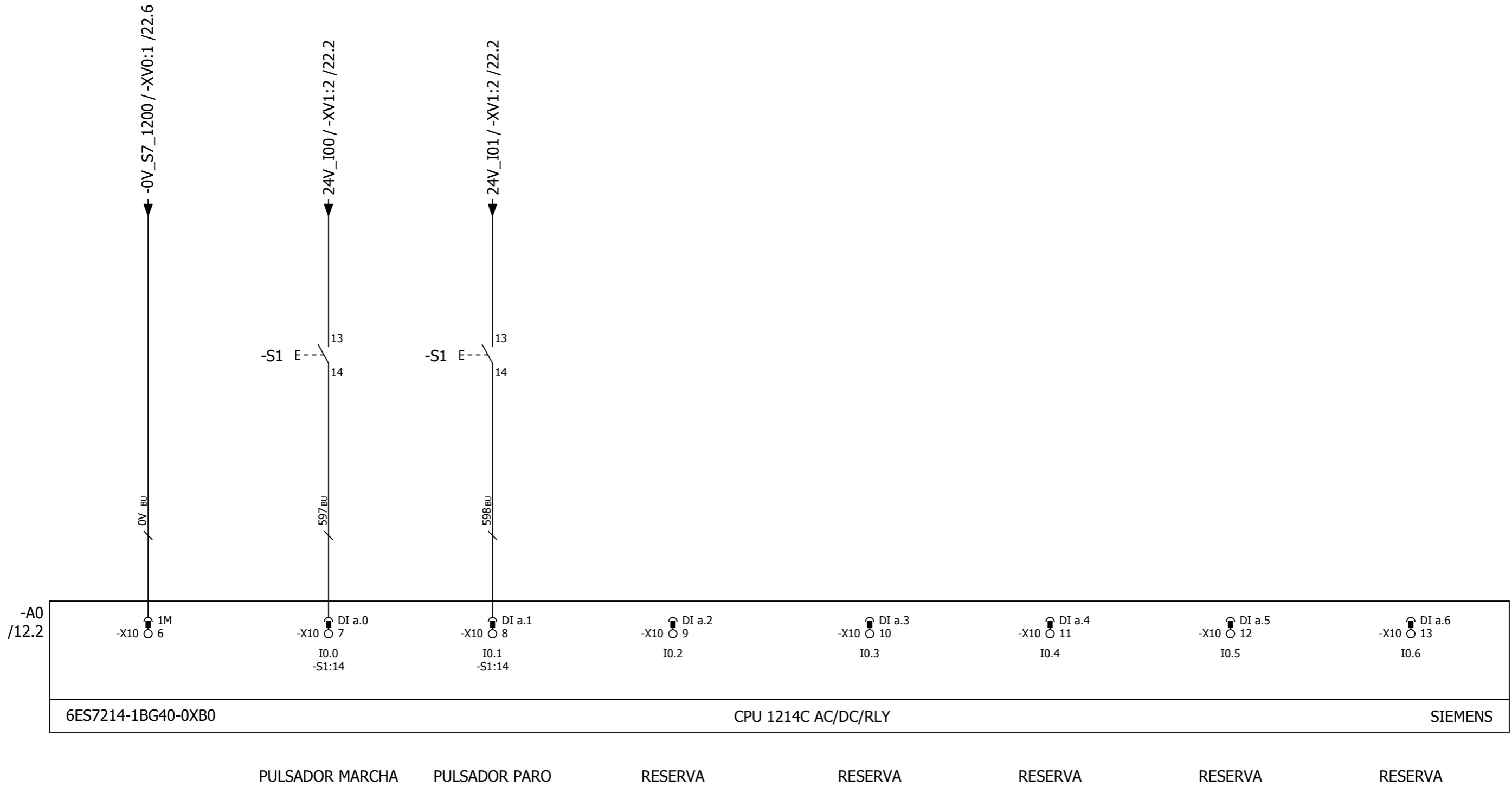


DISTRIBUCIÓN 0 VDC

			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	DISTRIBUCION VDC	CGB_7000080-1	= CTRL
			Resp.	W7_64				+ ARM
			Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN SADE DE MOSA		Página 9/23

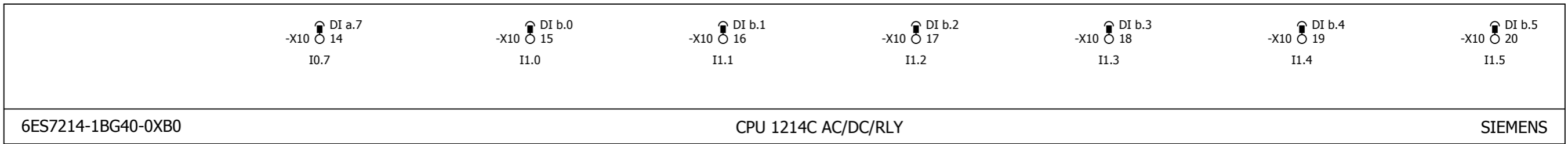


			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	ALIMENTACIÓN S7-1200	DIC-RPAL-2017_v1	= CTRL
			Resp.	W7_64				+ ARM
			Probado					
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S7-1200	Página 10 / 23



			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	PLC DI (I)	DIC-RPAL-2017_v1	= CTRL
			Resp.	W7_64				+ ARM
			Probado					
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S...	...

-A0
/12.2



RESERVA

RESERVA

RESERVA

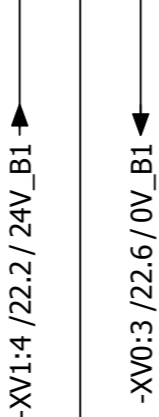
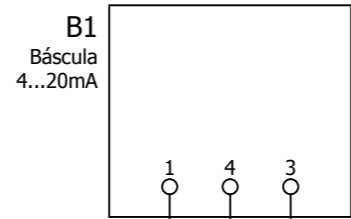
RESERVA

RESERVA

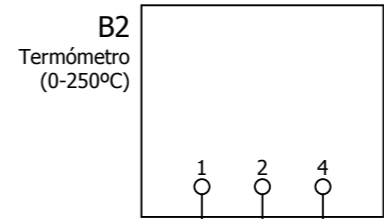
RESERVA

RESERVA

			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC		PLC DI (II)	DIC-RPAL-2017_v1	= CTRL	
			Resp.	W7_64					+ ARM	
			Probado						CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	PÁGINA DE MOSTRA
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por					Página 12 / 23

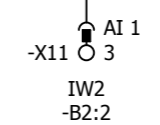
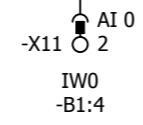
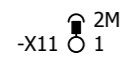


599 BU



600 BU

-A0 / 12.2



6ES7214-1BG40-0XB0

CPU 1214C AC/DC/RLY

SIEMENS

BÁSCULA

TERMÓMETRO

		Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	PLC AI	DIC-RPAL-2017_v1	= CTRL
		Resp.	W7_64				+ ARM
		Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S	PLC DE MONTAJE
							Página 13 / 23

RESERVA

MOTOR GRUESOS
MOSTAZA BLANCA

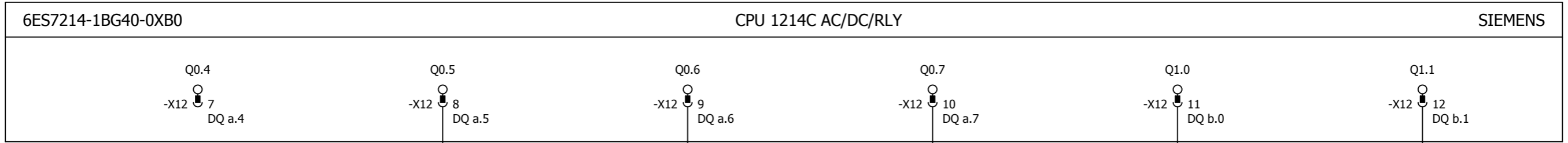
MOTOR FINOS
MOSTAZA BLANCA

MOTOR GRUESOS
MOSTAZA NEGRA

MOTOR FINOS
MOSTAZA NEGRA

MOTOR CÚRCUMA

-A0
/12.2



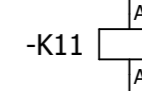
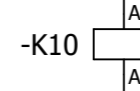
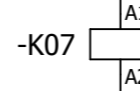
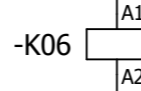
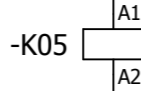
605_{BU}

606_{BU}

610₆₀₇

608_{BU}

609_{BU}



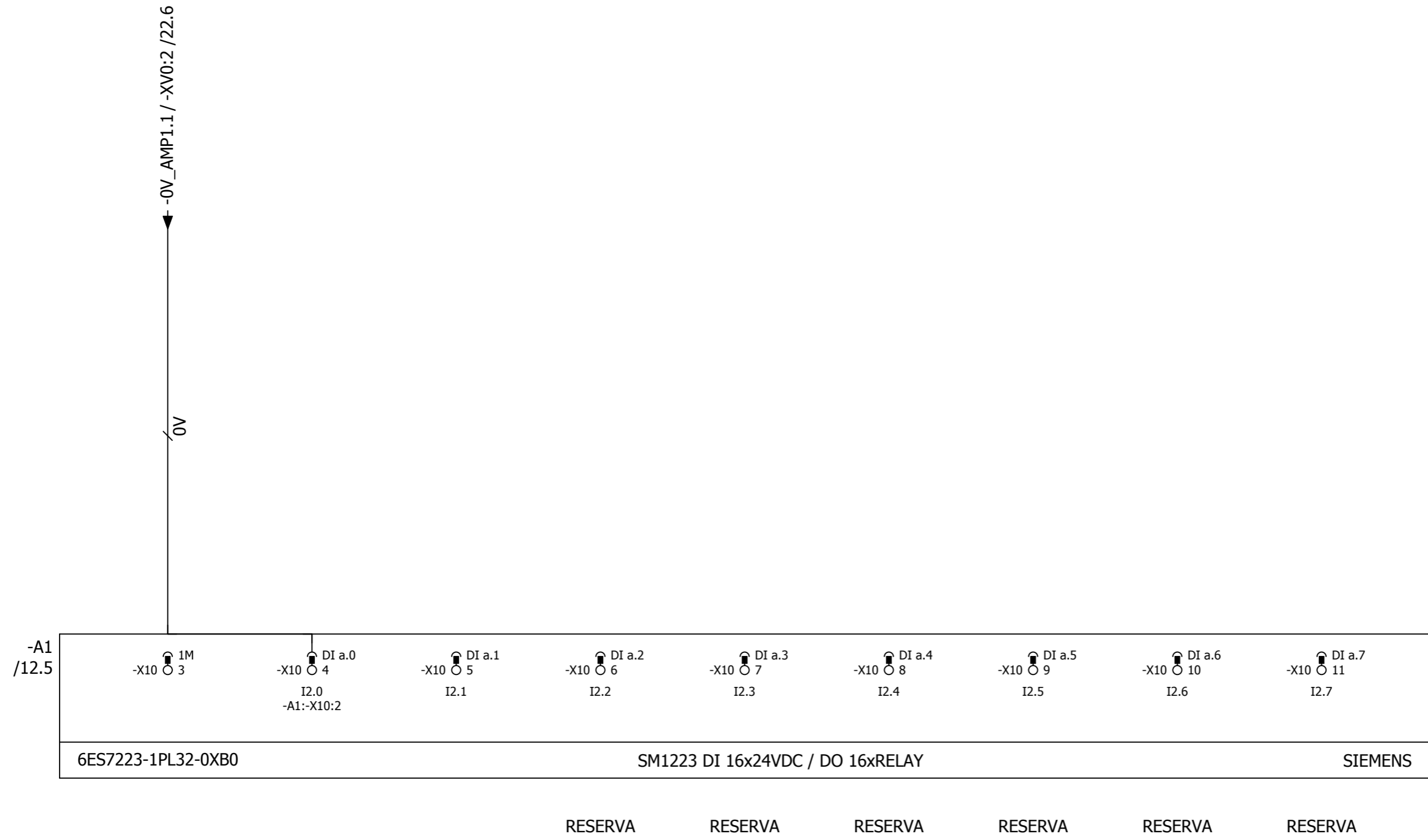
-XV0:4 /22.7 / 0V_CPU_DI2 ←

0V_{BU}

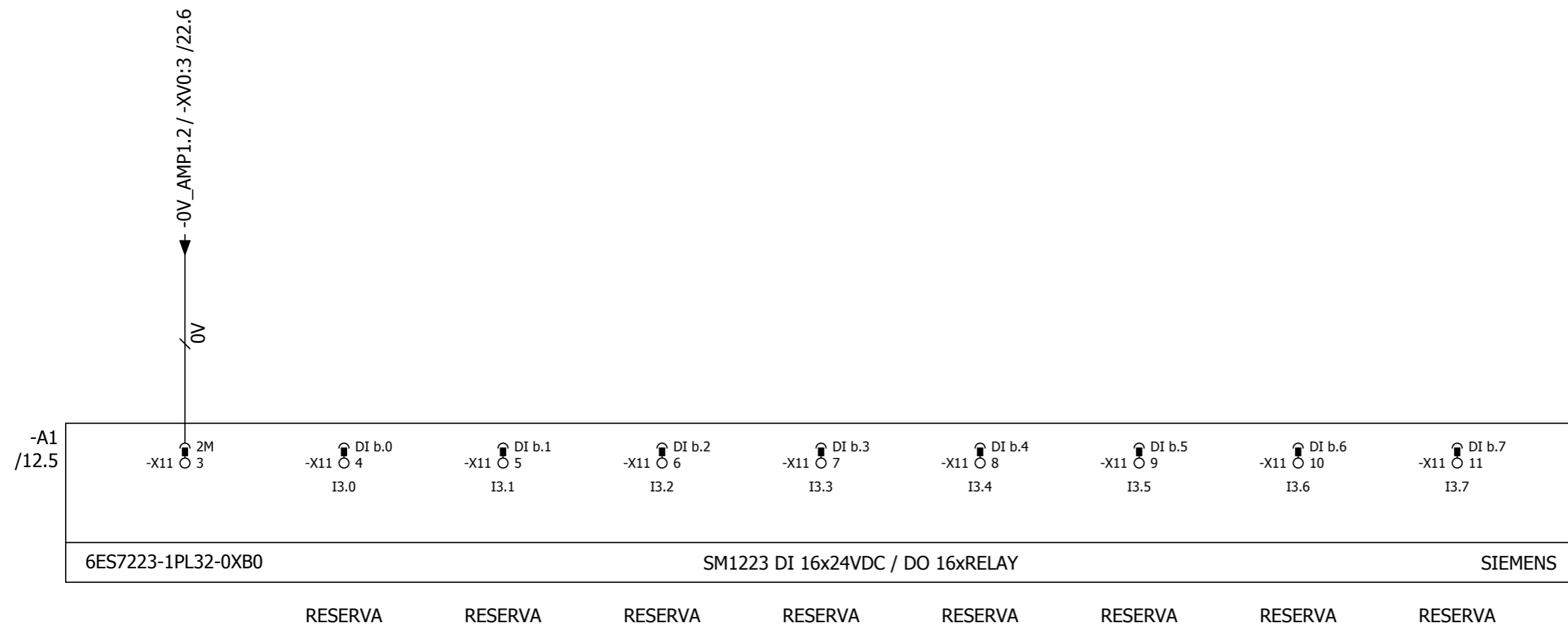
0V_{BU}

0V_{BU}

		Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	PLC DO (II)	DIC-RPAL-2017_v1	= CTRL
		Resp.	W7_64				+ ARM
		Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S	PLC DE MOSTAZA
							Página 15 / 23



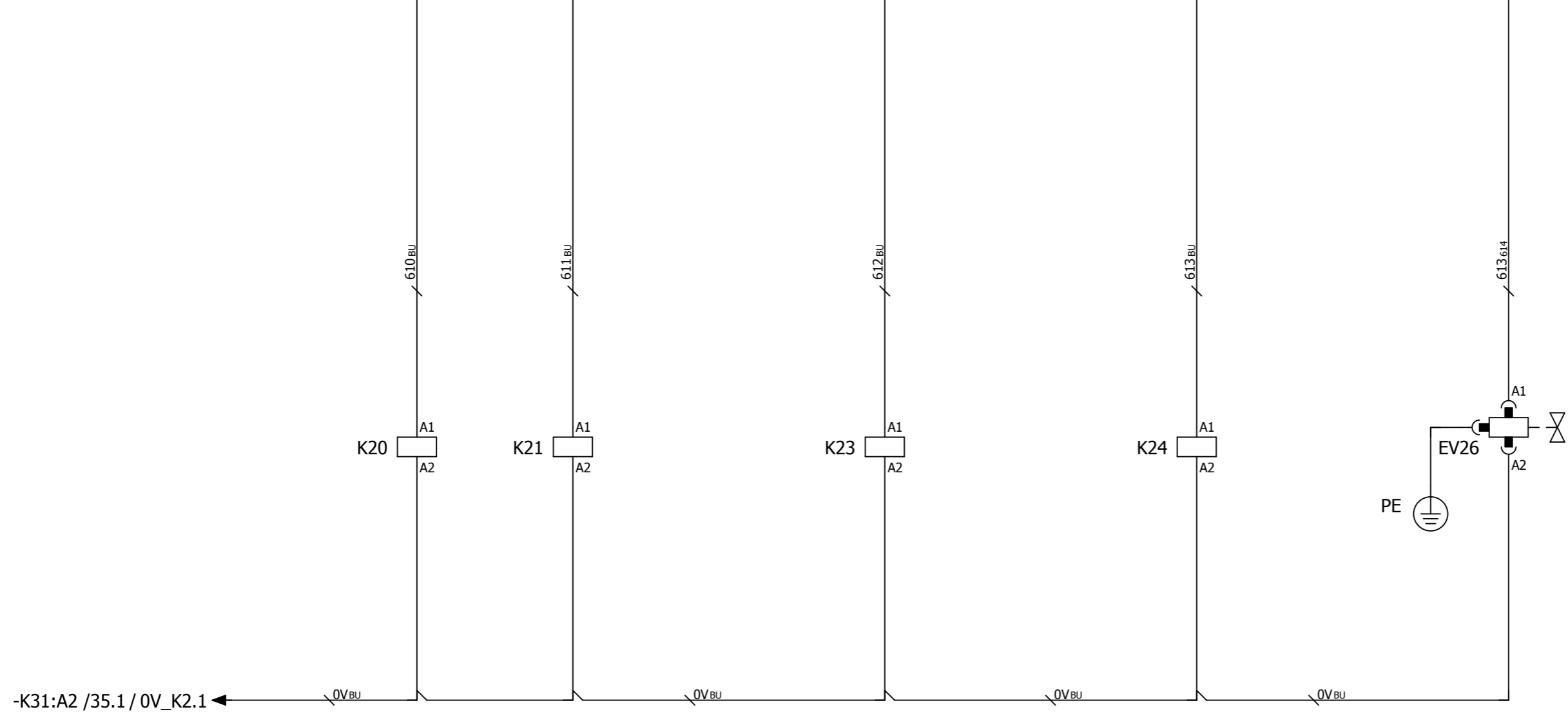
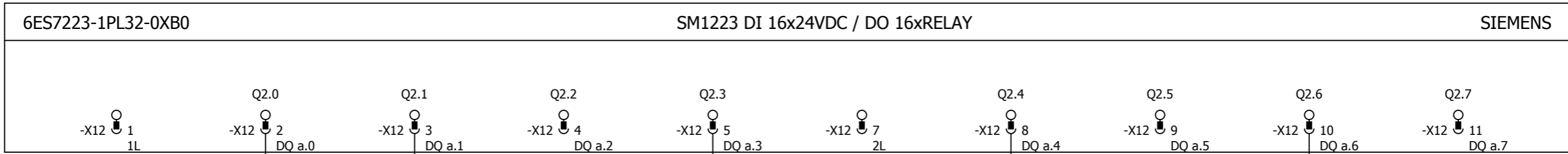
			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	AMP1 DI (I)	DIC-RPAL-2017_v1	= CTRL
			Resp.	W7_64				+ ARM
			Probado					
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S...	...



			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	AMP1 DI (II)	DIC-RPAL-2017_v1	= CTRL
			Resp.	W7_64				+ ARM
			Probado					
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por		CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S...

MOTOR CÚRCUMA MOTOR PÁPRIKA RESERVA MOTOR SAL RESERVA MOTOR TANQUE DE MEZCLA RESERVA ELECTROVÁLVULA TRASVASADO RESERVA

-A1 /12.5



				Fecha	12/07/2019			AMP1 DO (I)	DIC-RPAL-2017_v1	= CTRL
				Resp.	W7_64					+ ARM
				Probado		ESQUEMAS CABLEADO PLC				
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por				CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	ÁREA DE MOSTRA
										Página 18 / 23

BOMBA
TRASVASADO

RESERVA

ELECTROVÁLVULA
ERROR VERTIDO

RESERVA

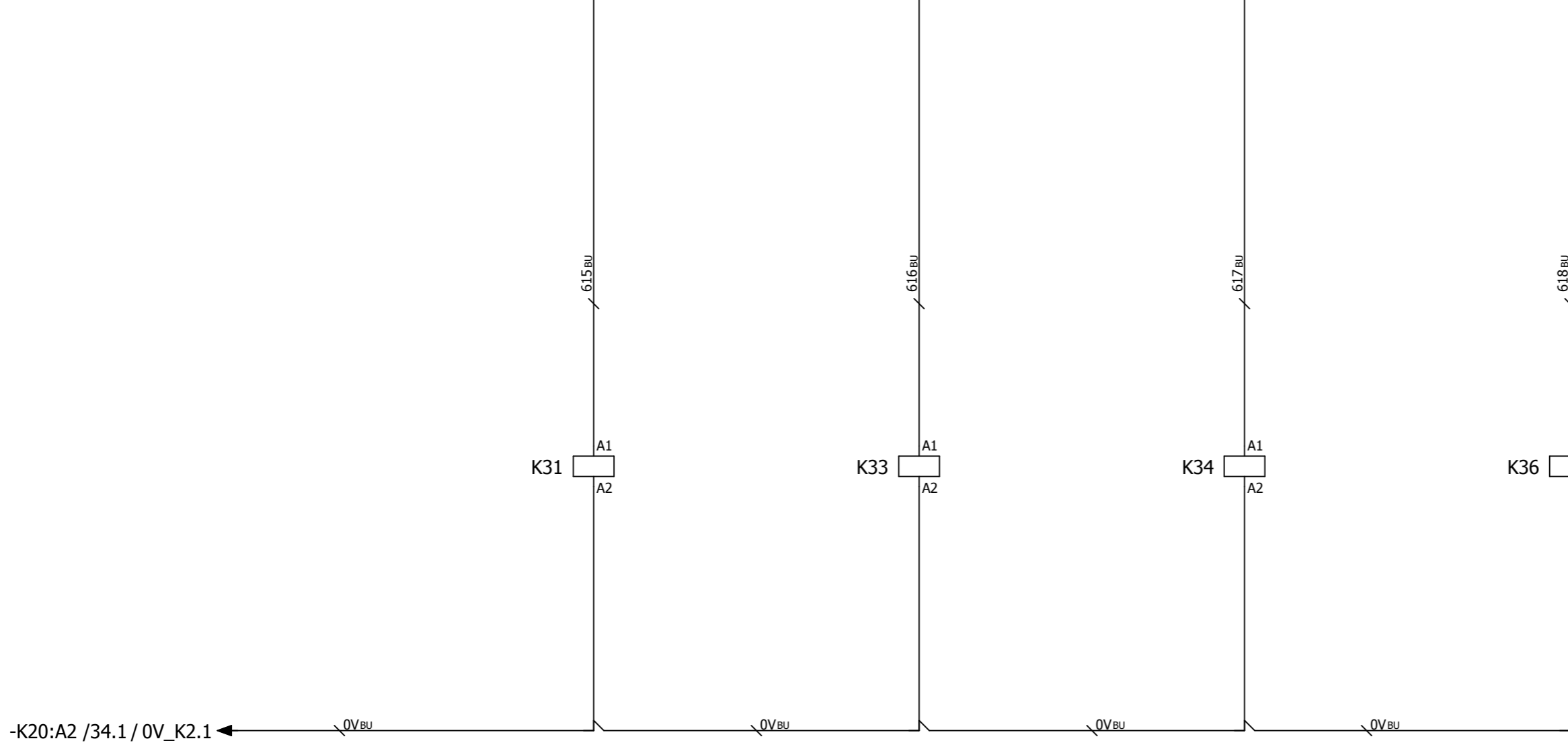
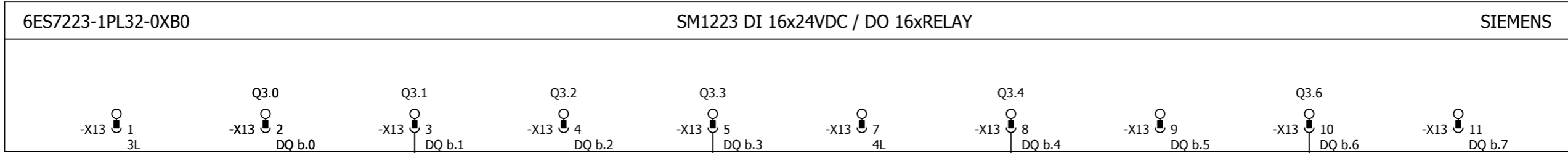
MOTOR
TRITURADOR

RESERVA

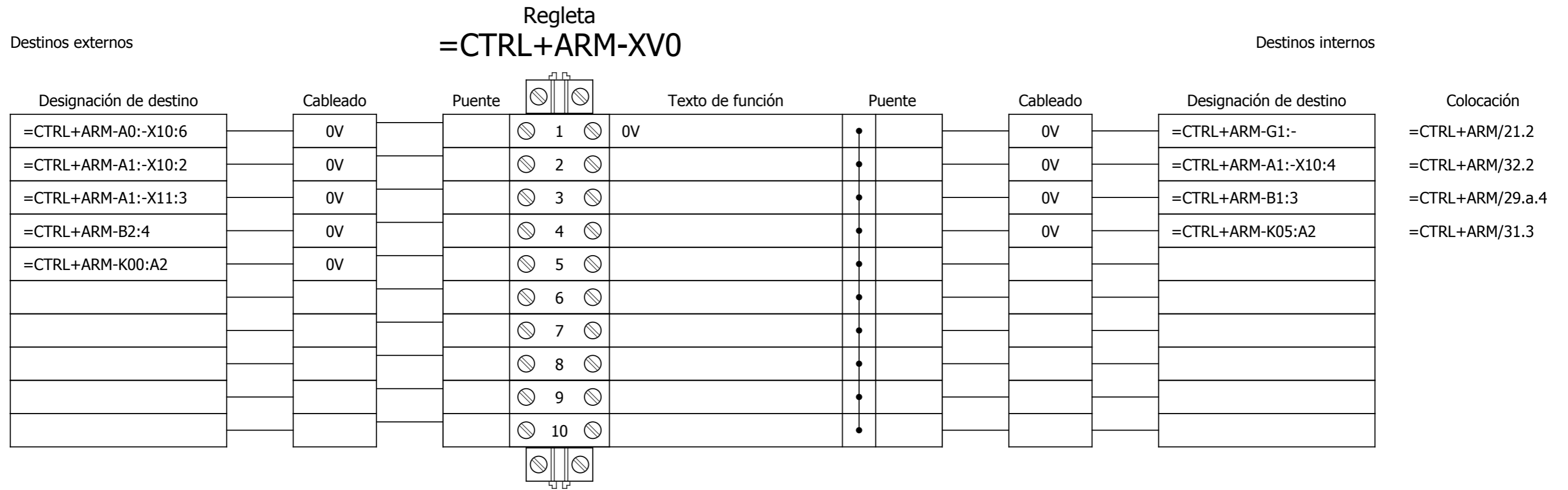
RESISTENCIAS DE
CALENTAMIENTO

RESERVA

-A1
/12.5



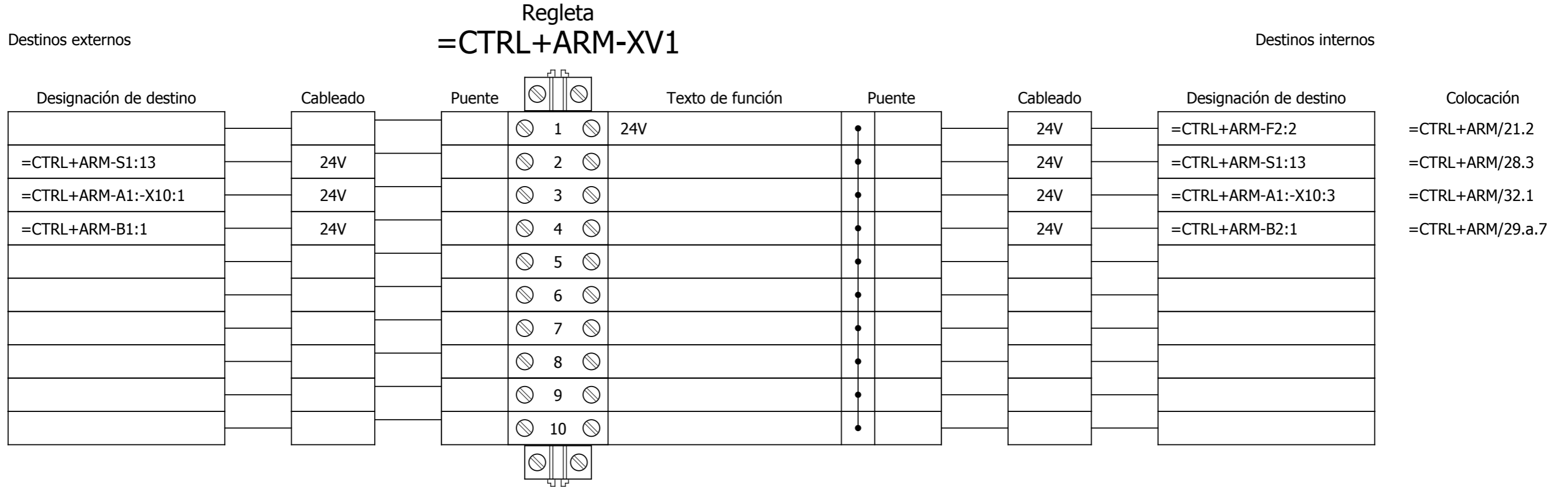
				Fecha	12/07/2019			AMP1 DO (II)	DIC-RPAL-2017_v1	= CTRL
				Resp.	W7_64					+ ARM
				Probado		ESQUEMAS CABLEADO PLC				
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por				CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	ANS DE MOS
										Página 19 / 23



			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	Plano de bornes =CTRL+ARM-XV0		= CTRL
			Resp.	W7_64				+ XL
			Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por		CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	USUARIO DE MOSTRA
							Página 20 / 23	

Plano de bornes

F13_003



1

+SL/1

			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	Plano de bornes =CTRL+ARM-XV1		= CTRL
			Resp.	W7_64				+ XL
			Probado					
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN S. DE MOSTRA	
								Página 21 / 23

Lista de suma de artículos

F02_002

Número de pedido	Cantidad	Designación	Número de tipo	Proveedor	Precio por unidad	Precio total
6ES7214-1BG40-0XB0	1	CPU 1214C, AC/DC/RELES, 14DI/10DO/2AI	6ES7214-1BG40-0XB0	SIE	0,00	0,00
6ES7223-1PL32-0XB0	1	E/S DIGITAL SM 1223, 16DI/16DO	6ES7223-1PL32-0XB0	SIE	0,00	0,00
1608540000	23	3-CONDUCTOR TERMINAL BLOCK ZDU 2.5	1608540000	WEI	0,00	0,00
5SY4506-7	1	INT.AUT 1P+1N 6A (C) 10KA	5SY4506-7	SIE	0,00	0,00
5SY6104-7	1	INT.AUT 1P 4A(C)	5SY6104-7	SIE	0,00	0,00
5ST3020	1	Bloque contactos auxiliares 1NA+1NC	5ST3020	SIE	7,78	0,00
5ST3805-1	1	UNIÓN P/CONT.AUX INT.AUTO	SIE.5ST3805-1	SIE	17,00	0,00
5SY4503-5	1	INTERR. PROTEC. SELEC. 230V 10KA, 1+N-POLOS, A, 3A, P=70MM	5SY4503-5	SIE	0,00	0,00
6EP1333-2BA20	1	SITOP PSU100S 230V/24V 5 A	6EP1333-2BA20	SIE	0,00	0,00

+SL/1

			Fecha	12/07/2019	ESQUEMAS CABLEADO PLC	Lista de suma de artículos : SIE.6ES7214-1BG40-0XB0 - SIE.6EP1333-2BA20	CONTROL Y VISUALIZACIÓN PROCESO ELABORACIÓN	= CTRL
			Resp.	W7_64				+ SL_R
			Probado					
Cambio	Fecha	Nombre	Original	Sustitución por	Sustituido por			Página 23 / 23

ANEXOS

CONTROL Y VISUALIZACIÓN DEL PROCESO DE ELABORACIÓN DE LA SALSA DE MOSTAZA

AUTOR: MATIAS PILES BOLUDA

TUTOR: ÁNGEL SAPENA BAÑO

COTUTOR: RUBÉN PUCHE PANADERO

Valencia, Septiembre 2019

ÍNDICE

ANEXO I: Navegación y descripción de las pantallas	1
I.I Movimientos manuales desde el HMI.....	7
I.II Modo automático desde el HMI	13
ANEXO II: Código de las funciones creadas y del programa de control	19
II.I) Función “Silo”	19
II.II) Gestión de las alarmas generadas por el proceso	29
II.III) Función “Esc_señal_analog”	32
II.IV) Función “Movimientos_manuales”	38
II.V Función “Control_histéresis”	44
II.VI Función “Activación_actuadores”	46
II.VII Bloque organización principal MAIN [OB1]	51
II. VIII) Bloque organización “Modo_automat” [OB124].....	62
II.IX) Bloque organización “Mov_manuales” [OB123]	78
ANEXO III: Visualización de los distintos lenguajes de programación	90
ANEXO IV: Creación de un tipo de dato	93
ANEXO V: Añadir la pantalla para la interfaz HMI	95
ANEXO VI Características Hardware	99
ANEXO VIII manual general PLC s7-1200.....	100
ANEXO IX programa integro sin comentarios.....	100

ÍNDICE DE FIGURAS

Figura 1 Pantalla de inicio HMI.....	1
Figura 2 Vista de la pantalla de inicio tras el primer arranque del proceso	2
Figura 3 Vista del control de usuarios en el HMI.....	3
Figura 4 Vista de la pantalla de parametrización de la fábrica (bis).....	3
Figura 5 Vista de la pantalla del histórico de alarmas	4
Figura 6 Vista del acuse de los fallos	5
Figura 7 Vista de la pantalla "Controls Silos".....	5
Figura 8 Visualización fallo de nivel y nivel aviso 1	6
Figura 9 Visualización fallo de nivel y nivel aviso 2	6
Figura 10 Vista de la pantalla "Movimientos manuales".....	7
Figura 11 Vista activación actuadores en la pantalla "Movimientos manuales".....	7
Figura 12 Vista selección mezclado en la pantalla "Movimientos manuales".....	8
Figura 13 Vista del mezclado en la pantalla "Movimientos manuales"	8
Figura 14 Vista finalización del mezclado en la pantalla "Movimientos manuales"	9
Figura 15 Vista selección del transvasado en la pantalla "Movimientos manuales"	9
Figura 16 Vista transvase en curso en la pantalla "Movimientos manuales".....	10
Figura 17 Vista finalización trasvase en la pantalla "Movimientos manuales"	10
Figura 18 Vista selección del triturado en la pantalla "Movimientos manuales".....	11
Figura 19 Vista triturado y cocido en funcionamiento en la pantalla "Movimientos manuales".....	11
Figura 20 Vista desconexión elementos calefactores durante el triturado en la pantalla "Movimientos manuales".....	12
Figura 21 Vista finalización del triturado en la pantalla "Movimientos manuales"	12
Figura 22 Vista pantalla recetas al seleccionar el "Modo automático"	13
Figura 23 Seleccionar y cargar receta en el "Modo automático".....	13
Figura 24 Vista de la receta cargada correctamente en el "Modo automático"	14
Figura 25 Vista del proceso en el "Modo automático".....	14
Figura 26 Ejemplo vertido materiales en el "Modo automático".....	15
Figura 27 Activación mezclador en el "Modo automático".....	15
Figura 28 Vista del trasvasado en el "Modo automático"	16
Figura 29 Vista del triturado en el "Modo automático".....	16
Figura 30 Vista tras la finalización del proceso en el "Modo automático".....	17
Figura 31 Vista tras producirse un error de vertido en el "Modo automático"	17
Figura 32 Vista tras acusar el error de vertido en el "Modo automático"	18
Figura 33 Variables Función silo	19

Figura 34 Vista función Silo II	27
Figura 35 Vista de los DB's de las funciones Silo en el árbol del proyecto	28
Figura 36 Vista del interior de un DB de la función Silo	28
Figura 37 Programación segmento 1 "Alarmas_y_fallos_SILOS"	29
Figura 38 Vista interior del DB "Alarmas_silos" (bis).....	30
Figura 39 Programación segmento 8 y 9 "Alarmas_y_fallos_SILOS"	31
Figura 40 Programación segmento 10 "Alarmas_y_fallos_SILOS"	31
Figura 41 Variables declaradas en la función "Esc_señal_analog"	32
Figura 42 Vista función "Esc_señal_analog" (bis).....	37
Figura 43 Variables declaradas en la función "Movimientos_manuales"	38
Figura 44 Variables declaradas en la función "Control_histéresis"	44
Figura 45 Vista de la función "Control_histéresis" (bis)	45
Figura 46 Programación segmento 1 "Activación_actuadores"	46
Figura 47 Programación segmento 2 "Activación_actuadores"	47
Figura 48 Programación segmento 3 "Activación_actuadores"	47
Figura 49 Programación segmento 4 "Activación_actuadores"	48
Figura 50 Programación segmentos 5 y 6 "Activación_actuadores"	48
Figura 51 Programación segmentos 7 y 8 "Activación_actuadores"	49
Figura 52 Programación segmento 9 "Activación_actuadores"	49
Figura 53 Programación segmento 10 "Activación_actuadores"	50
Figura 54 Programación segmentos 11 y 12 "Activación_actuadores"	50
Figura 55 Programación segmento 1 MAIN [OB1]	51
Figura 56 Programación segmento 2 MAIN [OB1]	52
Figura 57 Programación segmento 3 MAIN [OB1]	53
Figura 58 Programación segmento 4 MAIN [OB1]	54
Figura 59 Detalle programación segmento 5 MAIN [OB1]	55
Figura 60 Programación segmento 5 MAIN [OB1]	55
Figura 61 Programación segmento 6 MAIN [OB1]	56
Figura 62 Programación segmento 7 MAIN [OB1]	57
Figura 63 Programación segmento 8 MAIN [OB1]	58
Figura 64 Programación segmento 9 MAIN [OB1]	59
Figura 65 Ampliación detalle 1 programación segmento 1 MAIN [OB1].....	59
Figura 66 Ampliación detalle 2 programación segmento 1 MAIN [OB1].....	60
Figura 67 Programación segmento 10, 11 y 12 MAIN [OB1]	61
Figura 68 Programación segmento 1 "Modo_automat" [OB124]	62
Figura 69 Programación segmento 2.a "Modo_automat" [OB124]	63
Figura 70 Programación segmento 2.b "Modo_automat" [OB124].....	63

Figura 71 Programación segmento 2.c “Modo_automat” [OB124]	64
Figura 72 Programación segmento 2.d “Modo_automat” [OB124]	65
Figura 73 Programación segmento 2.e “Modo_automat” [OB124]	66
Figura 74 Programación segmento 2.f “Modo_automat” [OB124]	67
Figura 75 Programación segmento 3 “Modo_automat” [OB124]	68
Figura 76 Programación segmento 4 “Modo_automat” [OB124]	69
Figura 77 Programación segmento 5 “Modo_automat” [OB124]	69
Figura 78 Programación segmento 6 “Modo_automat” [OB124]	70
Figura 79 Programación segmento 7.a “Modo_automat” [OB124]	70
Figura 80 Programación segmento 7.b “Modo_automat” [OB124]	71
Figura 81 Programación segmento 8 “Modo_automat” [OB124]	72
Figura 82 Programación segmento 9 “Modo_automat” [OB124]	73
Figura 83 Programación segmento 10.a “Modo_automat” [OB124]	74
Figura 84 Programación segmento 10.b “Modo_automat” [OB124]	74
Figura 85 Programación segmento 10.c “Modo_automat” [OB124]	75
Figura 86 Programación segmento 11.a “Modo_automat” [OB124]	76
Figura 87 Programación segmento 11.b “Modo_automat” [OB124]	77
Figura 88 Figura 138 Programación segmento 12 “Modo_automat” [OB124]	77
Figura 89 Programación segmento 1.a “Mov_manuales” [OB123]	78
Figura 90 Programación segmento 1.b “Mov_manuales” [OB123]	78
Figura 91 Programación segmento 2.a “Mov_manuales” [OB123]	79
Figura 92 Programación segmento 2.b “Mov_manuales” [OB123]	79
Figura 93 Programación segmento 2.c “Mov_manuales” [OB123]	80
Figura 94 Programación segmento 3 “Mov_manuales” [OB123]	80
Figura 95 Programación segmento 4.a “Mov_manuales” [OB123]	81
Figura 96 Programación segmento 4.b “Mov_manuales” [OB123]	81
Figura 97 Programación segmento 5.a “Mov_manuales” [OB123]	82
Figura 98 Programación segmento 5.b “Mov_manuales” [OB123]	82
Figura 99 Programación segmento 6.a “Mov_manuales” [OB123]	83
Figura 100 Programación segmento 6.b “Mov_manuales” [OB123]	83
Figura 101 Programación segmento 7.a “Mov_manuales” [OB123]	84
Figura 102 Programación segmento 7.b “Mov_manuales” [OB123]	85
Figura 103 Programación segmento 7.c “Mov_manuales” [OB123]	85
Figura 104 Programación segmento 8.a “Mov_manuales” [OB123]	86
Figura 105 Programación segmento 8.b “Mov_manuales” [OB123]	87
Figura 106 Programación segmento 8.c “Mov_manuales” [OB123]	87
Figura 107 Programación segmento 9 “Mov_manuales” [OB123]	88

Figura 108 Programación segmento 10.a “Mov_manuales” [OB123]	89
Figura 109 Programación segmento 10.b “Mov_manuales” [OB123]	89
Figura 110 Ejemplo de creación FB TIA Portal	90
Figura 111 Declaración de variables en un FB	90
Figura 112 Ejemplo escritura código SCL.....	91
Figura 113 Ejemplo vista de una función creada al ser llamada	91
Figura 114 Ejemplo creación automática de un DB.....	91
<i>Figura 115 Declaración de variables principales en el árbol del proyecto</i>	92
Figura 116 Ejemplo conexión entradas/salidas a una función creada	92
Figura 117 Ejemplo maniobra marcha paro en Ladder	92
Figura 118 Ejemplo maniobra marcha paro en SCL.....	93
Figura 119 Vista del árbol del proyecto.....	93
Figura 120 Declaración de variables del tipo de dato "Parámetros_silos"	94
Figura 121 Vista en el árbol del proyecto del DB creado para almacenar el tipo de dato "Parámetros_silos"	94
Figura 122 Vista en el interior del DB creado para almacenar el tipo de dato "Parámetros_silos"	94
Figura 123 Añadir un sistema PC para el control HMI.....	95
Figura 124 Vista de los dispositivos a comunicar	96
Figura 125 Configuración hardware necesario para la comunicación profinet.....	96
Figura 126 Vista del hardware necesario para la comunicación comunicado	96
Figura 127 Vista conexión PN/IE	97
Figura 128 Vista conexión PN/IE establecida	97
Figura 129 Vista de las IP de los dispositivos configurados	97
Figura 130 Vista del nuevo sistema PC para el HMI en el árbol del proyecto	98

ANEXO I: NAVEGACIÓN Y DESCRIPCIÓN DE LAS PANTALLAS

Pantalla de inicio:

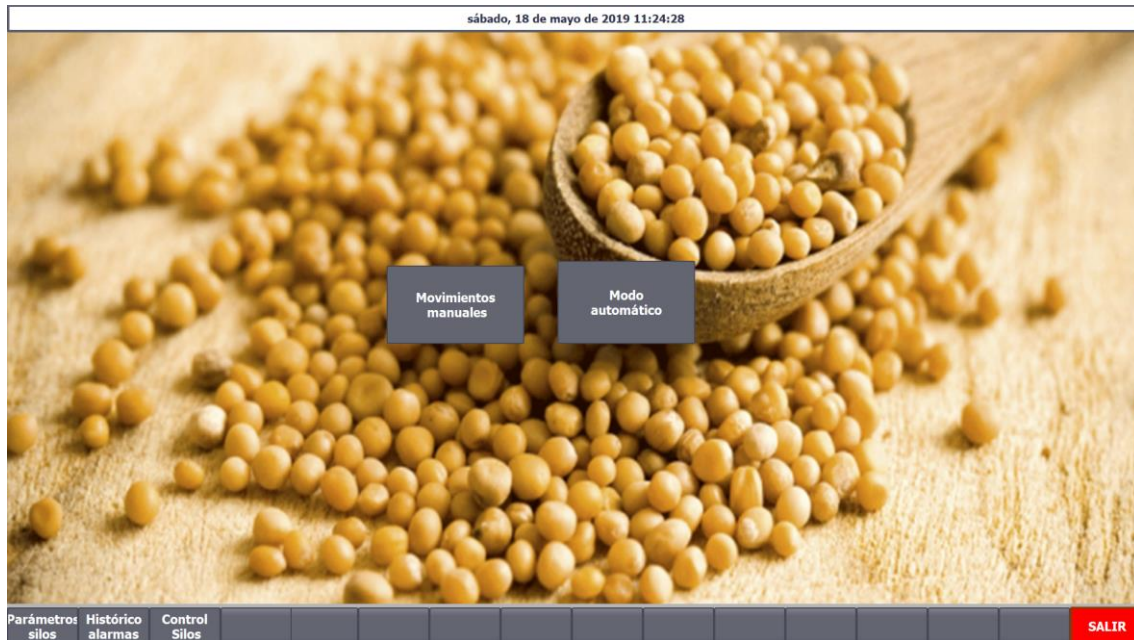


Figura 1 Pantalla de inicio HMI

Esta será la primera pantalla que aparezca al encender el autómata. Desde aquí se tiene acceso al resto de pantallas que forman el entorno de control.

<p>Movimientos manuales</p>	<p>Al pulsar este botón se accede a la pantalla de movimientos manuales.</p>
<p>Modo automático</p>	<p>Al pulsar este botón, se accede a la pantalla del modo automático, en la cual se deberá de seleccionar la receta que se quiere fabricar.</p>
<p>Parámetros silos</p>	<p>Al pulsar este botón, se aparece en la pantalla para la parametrización de la instalación. Esta parametrización debe de ser hecha por personal autorizado, por ello se ha dotado de control de usuarios a la pantalla. Únicamente podrán modificar los valores asociados a la instalación las personas que conozcan el usuario y contraseña de acceso.</p>
<p>Histórico alarmas</p>	<p>Al pulsar este botón se accede a la tabla del histórico de alarmas.</p>
<p>Control Silos</p>	<p>Al pulsar este botón, se accede a la visualización del estado de los silos.</p>

A continuación se van a exponer por orden lógico, la configuración del sistema, y la navegación entre pantallas:

Al arrancar por primera vez aparecerá lo siguiente:

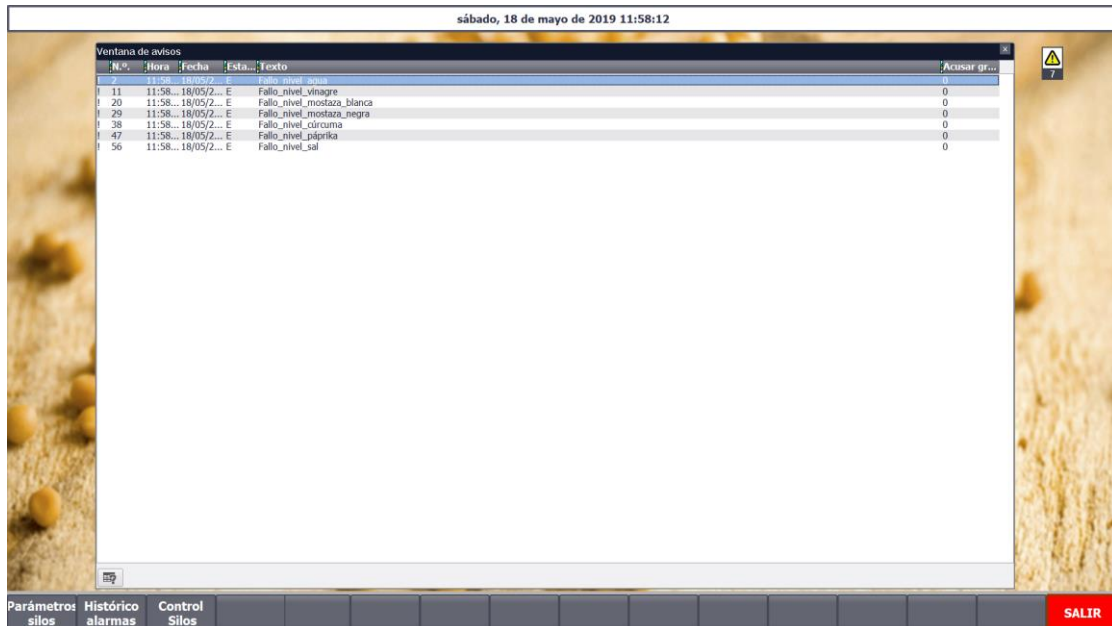


Figura 2 Vista de la pantalla de inicio tras el primer arranque del proceso

Lo primero que se muestra al acceder a la pantalla de inicio con el autómata en RUN, es el visor de avisos, con todos los errores y alarmas que actualmente hay activos, puesto que se ha configurado este visor como ventana emergente para que obligatoriamente se deba de cerrar manualmente.

Se pretende así evitar que el operario no sea consciente de los errores o fallos del sistema. Además, aun habiendo cerrado esta ventana, si hay avisos que permanecen activos, saldrá un símbolo de atención con un número abajo que reflejará la cantidad de fallos activos actualmente.

Al cerrar esta ventana, se puede pasar a la parametrización de los silos:

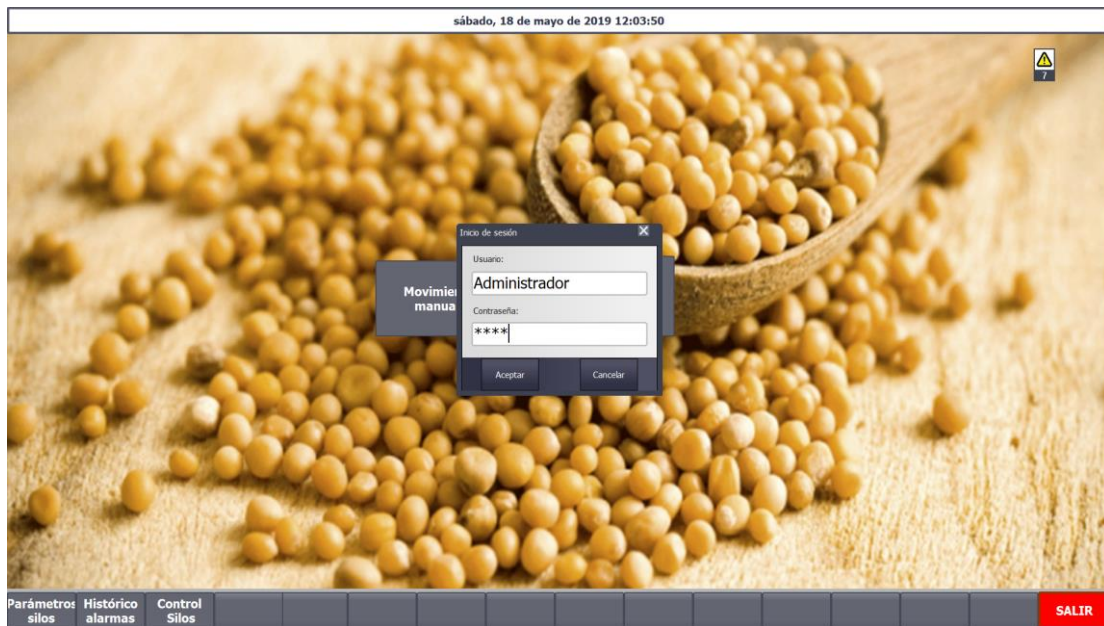


Figura 3 Vista del control de usuarios en el HMI

Como se ha comentado anteriormente, al pulsar el botón “Parámetros silos” e intentar acceder a esta pantalla salta el **control de usuarios**.

Se ha considerado que los únicos con potestad para la parametrización de la planta serán los administradores. Por ello se ha creado un control de accesos:

Usuario: Administrador

Contraseña: 1234

Se deberá de hacer click en aceptar, y volver a pulsar “**Parámetros silos**” para acceder a la siguiente pantalla:

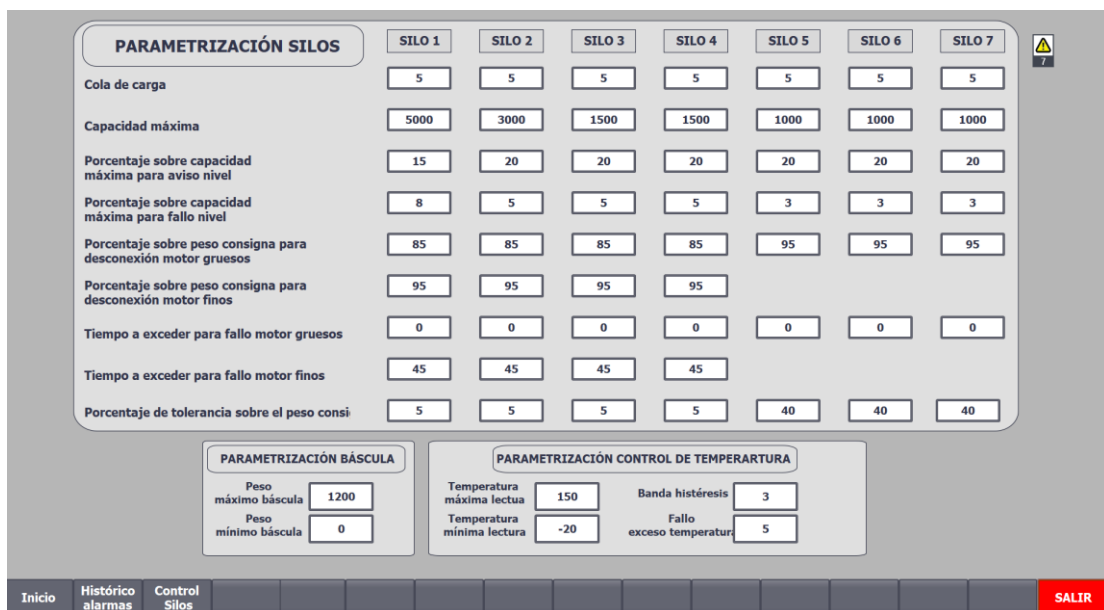


Figura 4 Vista de la pantalla de parametrización de la fábrica (bis)

Dentro de esta pantalla se encuentran todos los parámetros a ajustar del proceso: capacidad de los silos, porcentaje para el aviso de nivel bajo y fallo de nivel, cola de carga etc. Además se parametrizará la báscula, el sensado de temperatura y su banda de histéresis para el control de la misma durante el cocido de la salsa.

Al salir de esta pantalla, por seguridad, automáticamente el sistema cierra sesión, lo que en la práctica significa que para volver a acceder a la pantalla de parametrización de los silos, se deberá de introducir el usuario y la contraseña de nuevo.

Una vez parametrizados los silos, si los parámetros se han establecido correctamente, la mayoría de los fallos y errores desaparecerán. Aún quedarán los relacionados con la cantidad de material disponible en los silos.

Al pulsar en el botón “**Histórico de alarmas**” se accede a una pantalla en la cual se encuentra una tabla de alarmas, donde se podrán ver las alarmas actuales, las producidas y acusadas anteriormente, etc.

Además de poder borrar o acusar las alarmas, se pueden observar unos indicadores del estado de los actuadores de cada silo. También se dispone de una serie de botones desde los que realizar el reseteo del fallo de material, o de los fallos de los motores/actuadores.

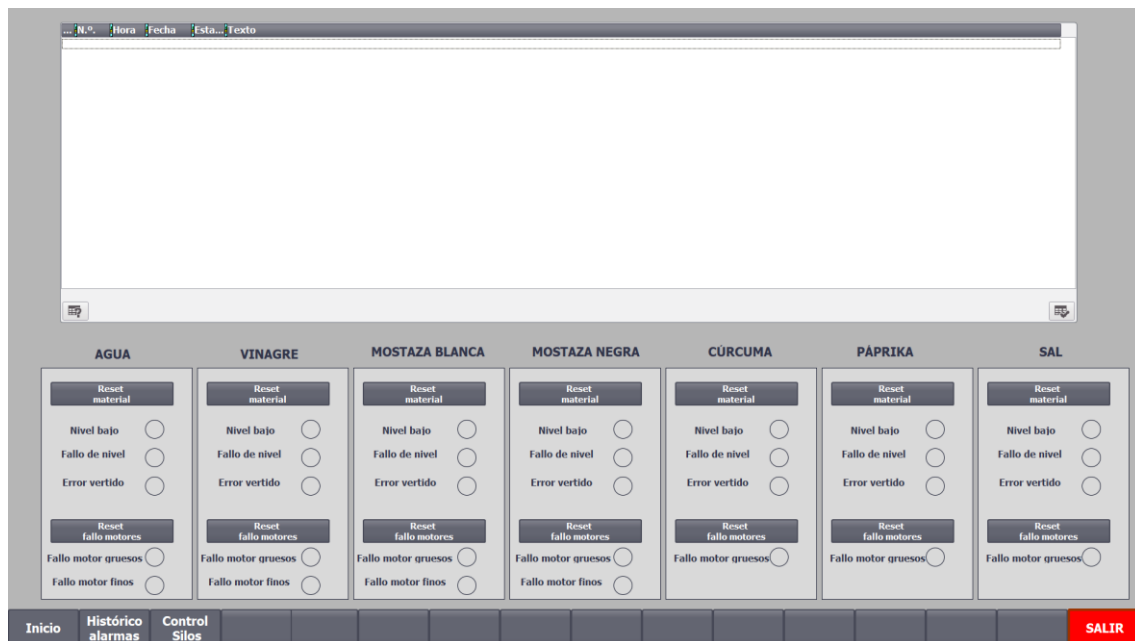


Figura 5 Vista de la pantalla del histórico de alarmas

Una vez pulsado el botón “Reset material” correspondiente a cada silo, el fallo de nivel quedará desactivado (estado del fallo ES):

N.º	Hora	Fecha	Esta...	Texto
2	11:58...	18/05/2...	ES	Fallo_nivel_agua
11	11:58...	18/05/2...	ES	Fallo_nivel_vinagre
20	11:58...	18/05/2...	ES	Fallo_nivel_mostaza_blanca
29	11:58...	18/05/2...	ES	Fallo_nivel_mostaza_negra
38	11:58...	18/05/2...	ES	Fallo_nivel_cúrcuma
47	11:58...	18/05/2...	ES	Fallo_nivel_páprika
56	11:58...	18/05/2...	ES	Fallo_nivel_sal

Figura 6 Vista del acuse de los fallos

Si se pulsa ahora el botón “Control silos” se accede a la visualización del estado de cada silo:

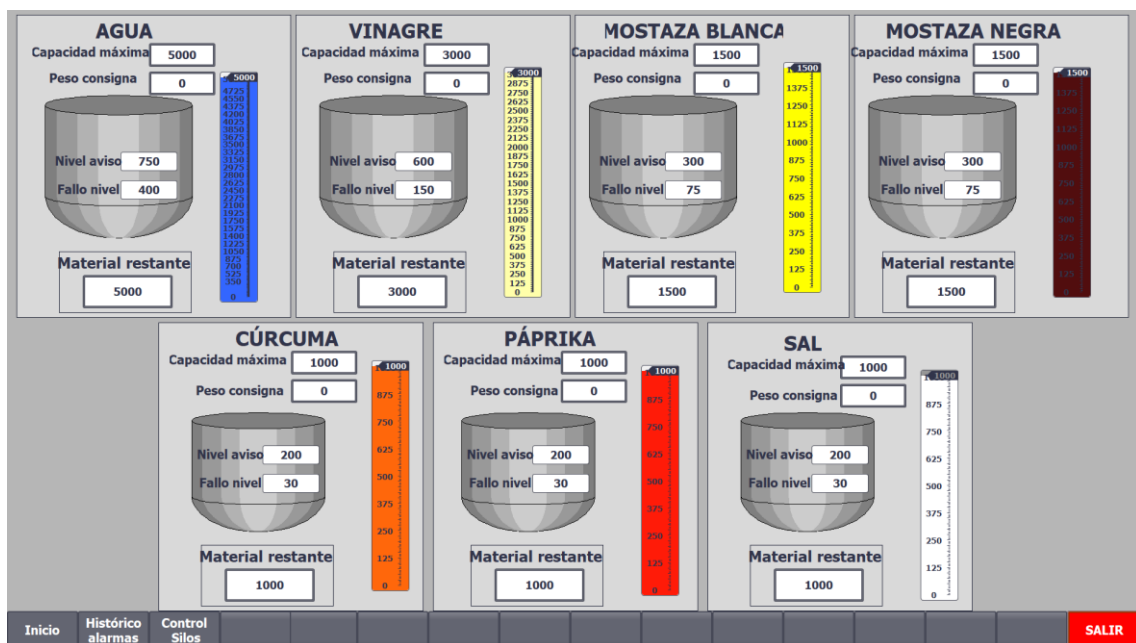


Figura 7 Vista de la pantalla "Controls Silos"

En esta pantalla, se visualiza todo lo relacionado con el material de cada silo, desde la capacidad máxima a los niveles de aviso y fallo a los que saltarán. Los recuadros correspondientes a los niveles de aviso y fallo de nivel aparecerán resaltados si se dan las condiciones para su activación.

Por ejemplo:

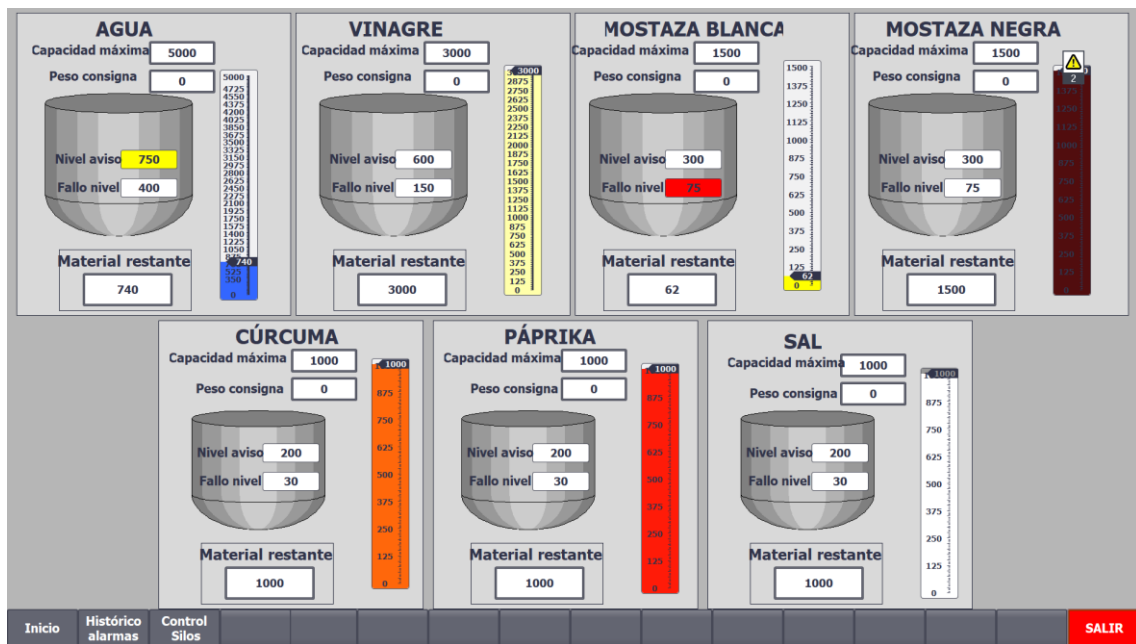


Figura 8 Visualización fallo de nivel y nivel aviso 1

Como se puede apreciar en la imagen, el silo encargado del almacenaje del agua tiene actualmente un valor inferior al valor del nivel de aviso, por lo que el recuadro que indica a partir de qué nivel se da el “Aviso nivel bajo” comienza a parpadear y pasa del fondo negro al amarillo. Lo mismo ocurre con el fallo de nivel, pero en este caso pasará a color rojo.

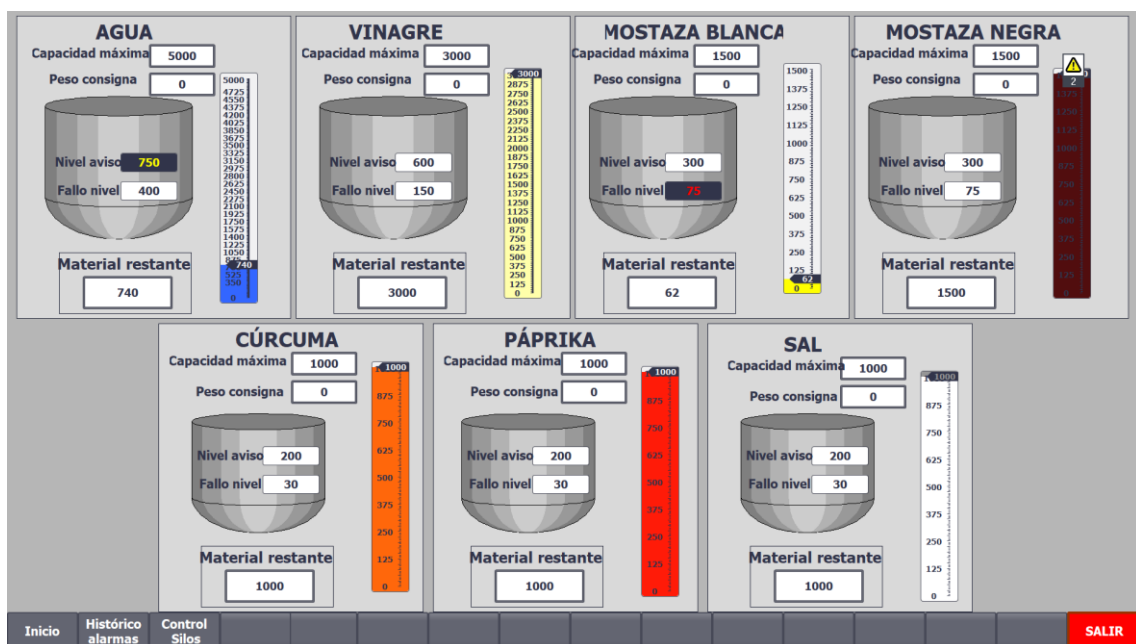


Figura 9 Visualización fallo de nivel y nivel aviso 2

I.1 MOVIMIENTOS MANUALES DESDE EL HMI

Una vez realizada la correcta parametrización, eliminados los fallos y teniendo material en los silos, se puede escoger entre movimientos manuales y el modo automático. Si se pulsa el botón **“Movimientos manuales”** se accederá a la siguiente pantalla:

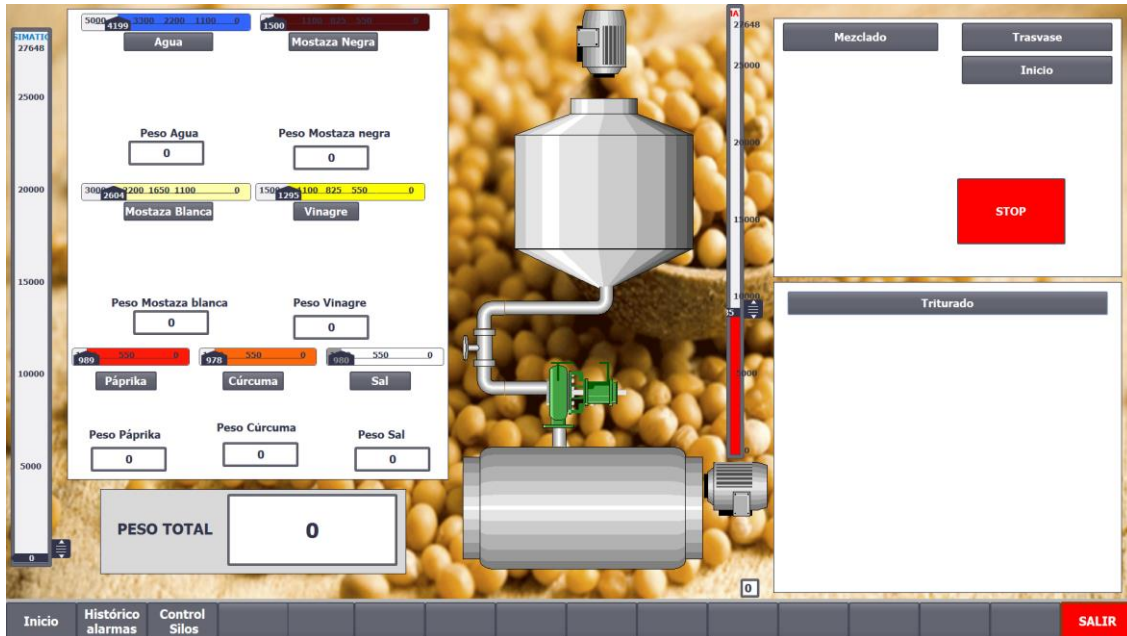


Figura 10 Vista de la pantalla "Movimientos manuales"

Desde aquí, además de controlar los movimientos manuales, se visualiza también que cantidad restante en cada silo.

Para iniciar el vertido se deberá de pulsar el botón correspondiente al material que se pretende pesar, y seleccionar los actuadores a activar:

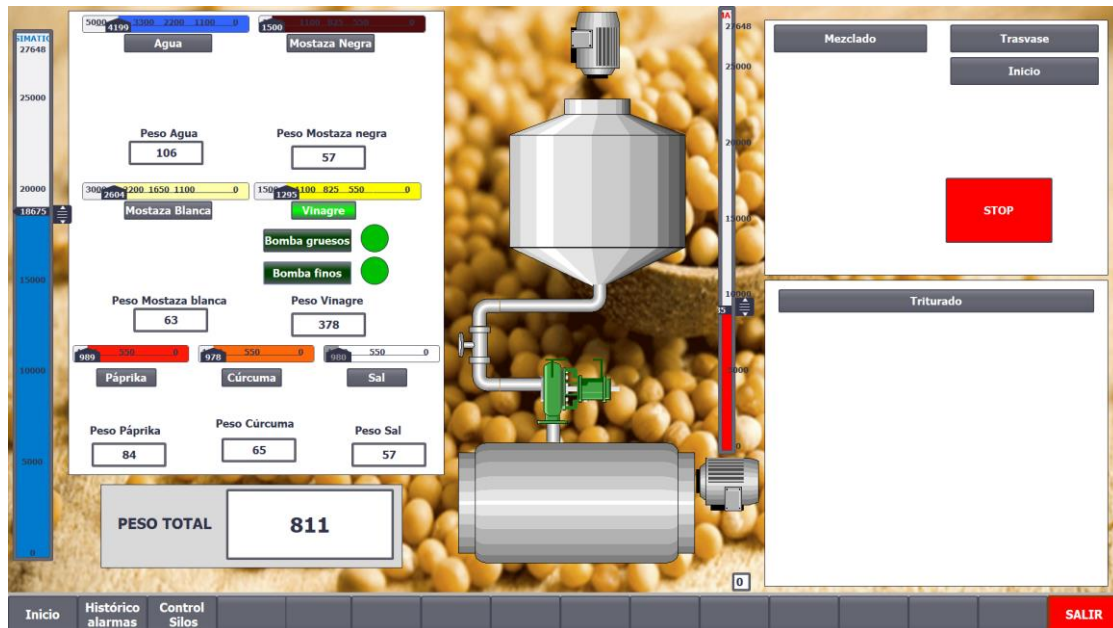


Figura 11 Vista activación actuadores en la pantalla "Movimientos manuales"

Una vez el operario considere adecuada la cantidad de cada material, el siguiente paso es realizar el mezclado, para ello hay que pulsar el botón "Mezclado", y obligatoriamente se deberá de programar un tiempo de mezcla (en segundos):

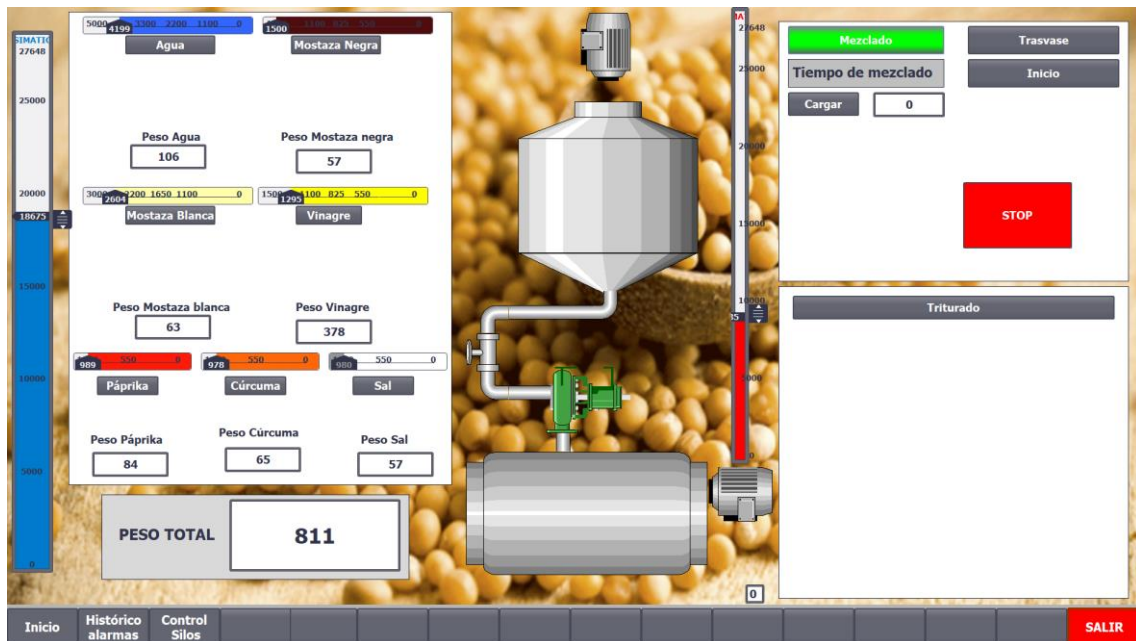


Figura 12 Vista selección mezclado en la pantalla "Movimientos manuales"

Tras pulsar "Cargar" y seguidamente "Iniciar mezclado" la pantalla mostrará una imagen similar a la siguiente:

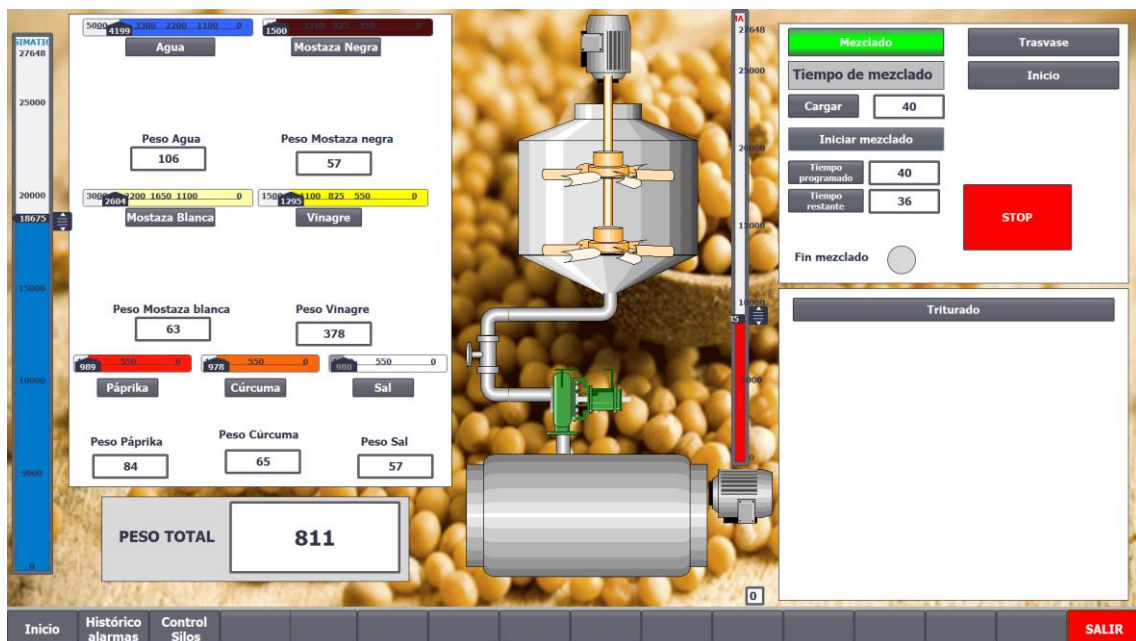


Figura 13 Vista del mezclado en la pantalla "Movimientos manuales"

Durante el tiempo que dura el mezclado se mantendrá parpadeando la imagen del mezclador, indicando que la salida está activa.

Al finalizar el tiempo, se iluminará el piloto “Fin mezclado” y se podrá pasar al trasvasado.

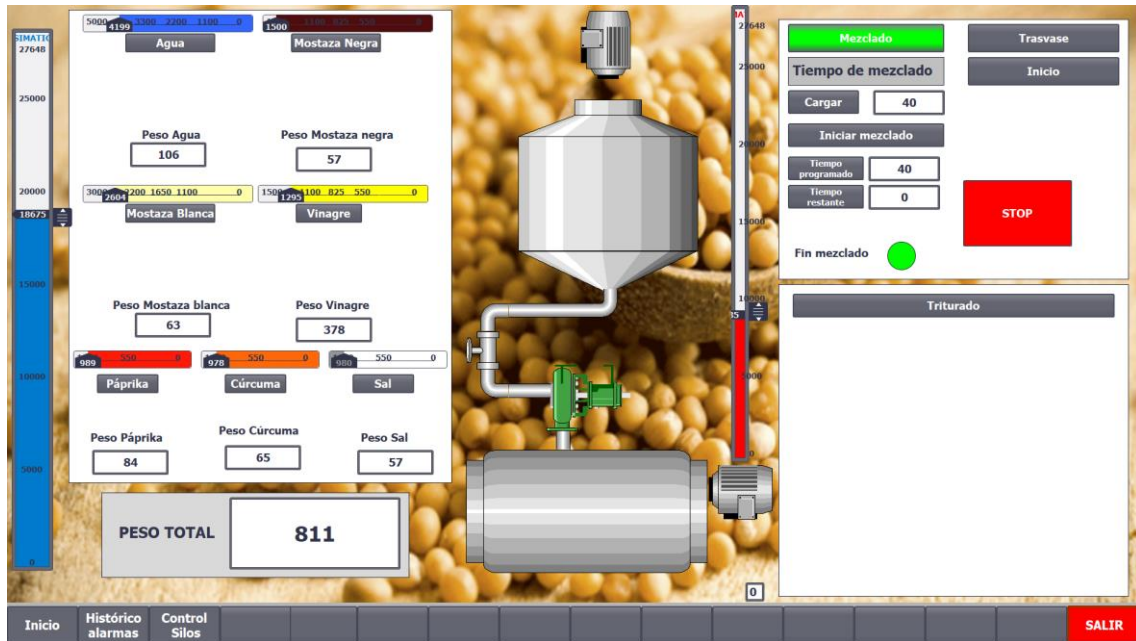


Figura 14 Vista finalización del mezclado en la pantalla "Movimientos manuales"

Para el trasvasado, se selecciona este paso, y se pulsa el botón “Inicio”:

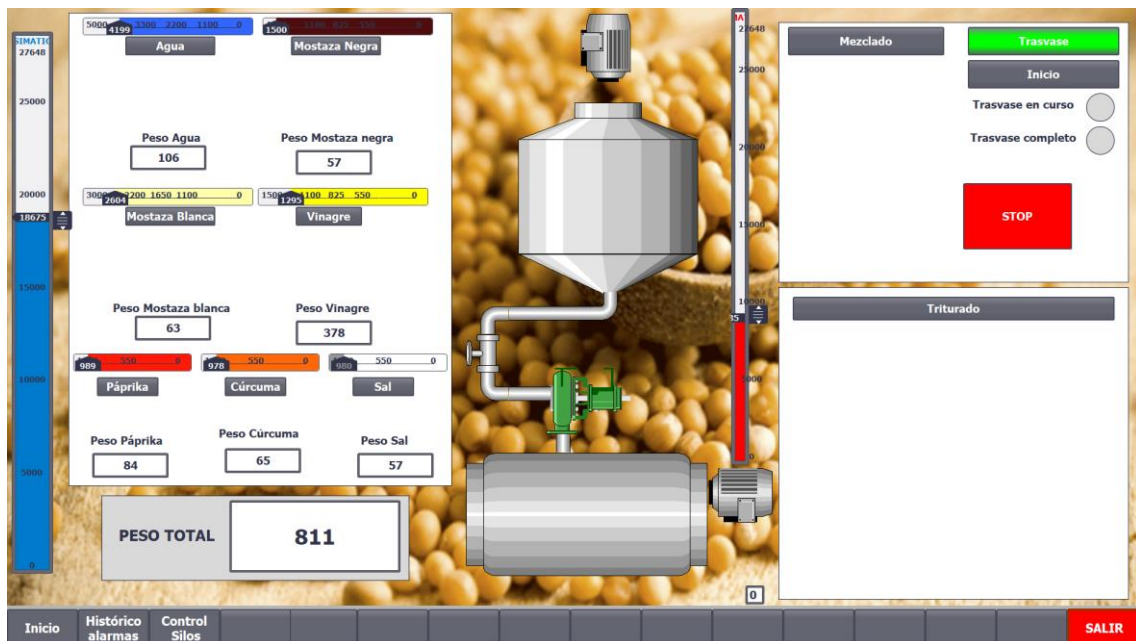


Figura 15 Vista selección del trasvasado en la pantalla "Movimientos manuales"

Al pulsar inicio, se verá como primero se activa la válvula de descarga del silo, y tras el tiempo que tardará en llenarse el tubo se activará la bomba.

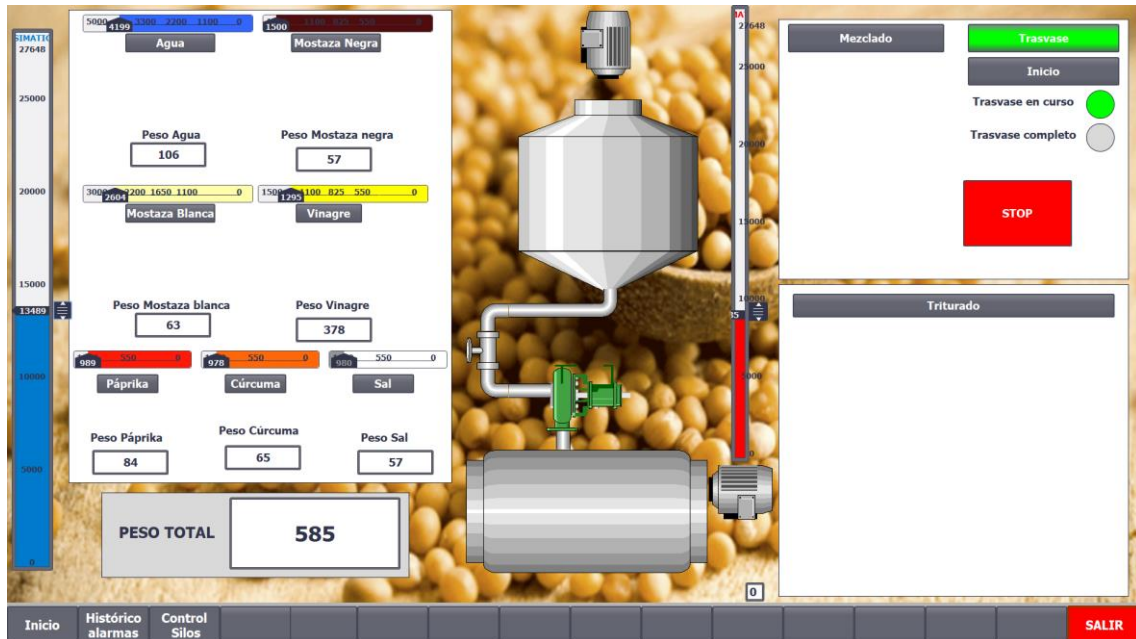


Figura 16 Vista trasvase en curso en la pantalla "Movimientos manuales"

Cuando el peso del material sea cero, se desactivarán los actuadores, y se iluminará el piloto de "Trasvase completado".

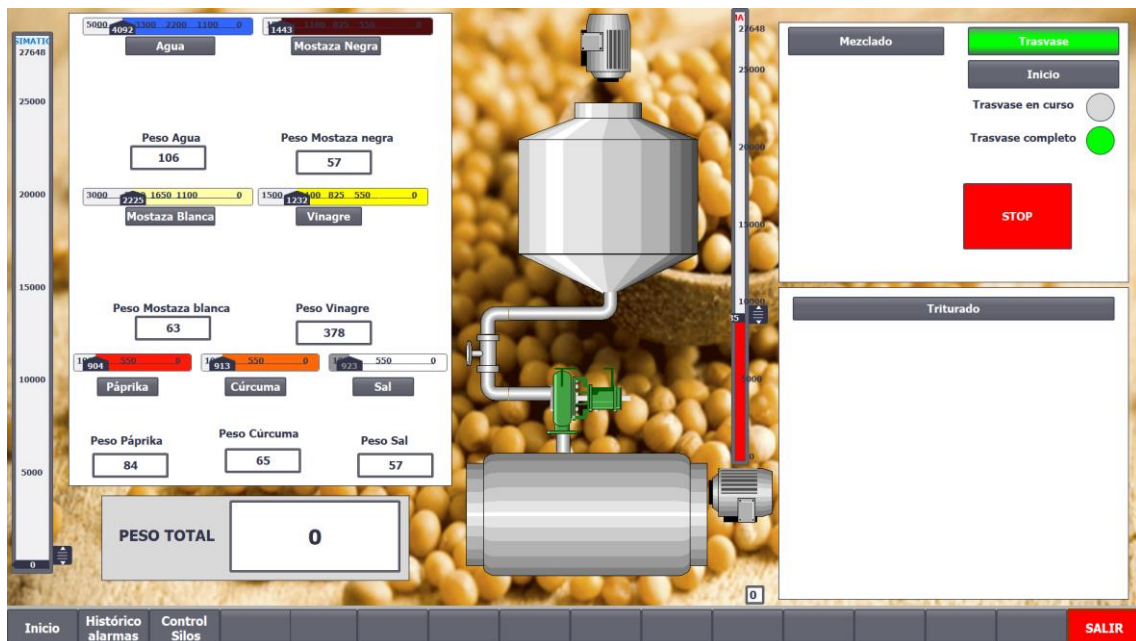


Figura 17 Vista finalización trasvase en la pantalla "Movimientos manuales"

Por último al pulsar el botón “triturado” se deberá de introducir el tiempo de triturado así como la temperatura de consigna:

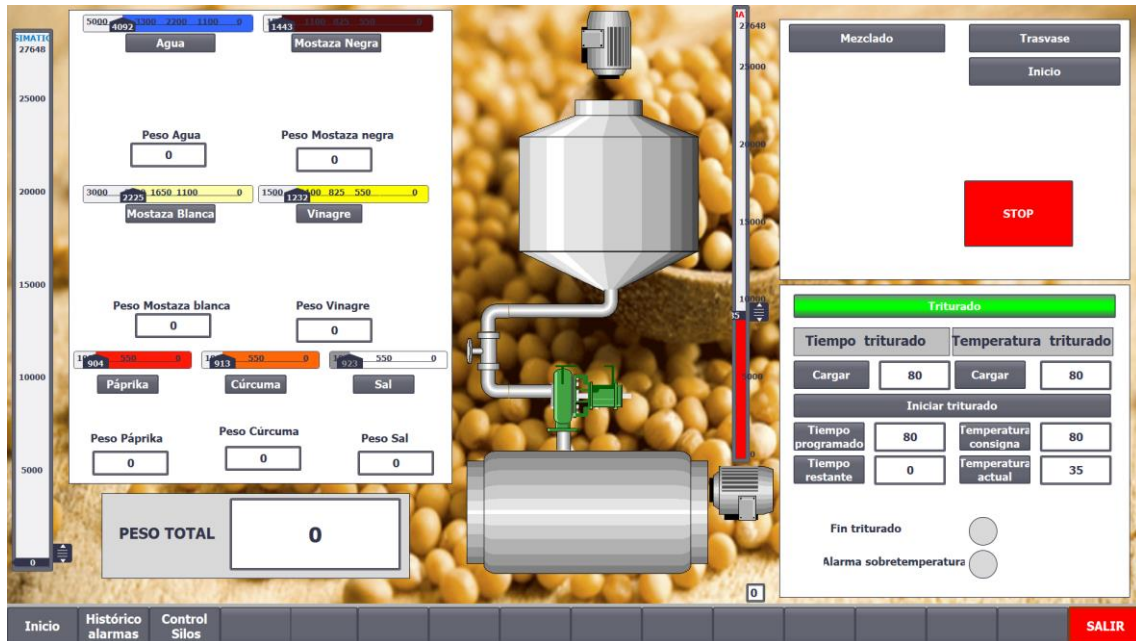


Figura 18 Vista selección del triturado en la pantalla "Movimientos manuales"

Y al igual que con el mezclado, darle a “Cargar” e “iniciar triturado”:

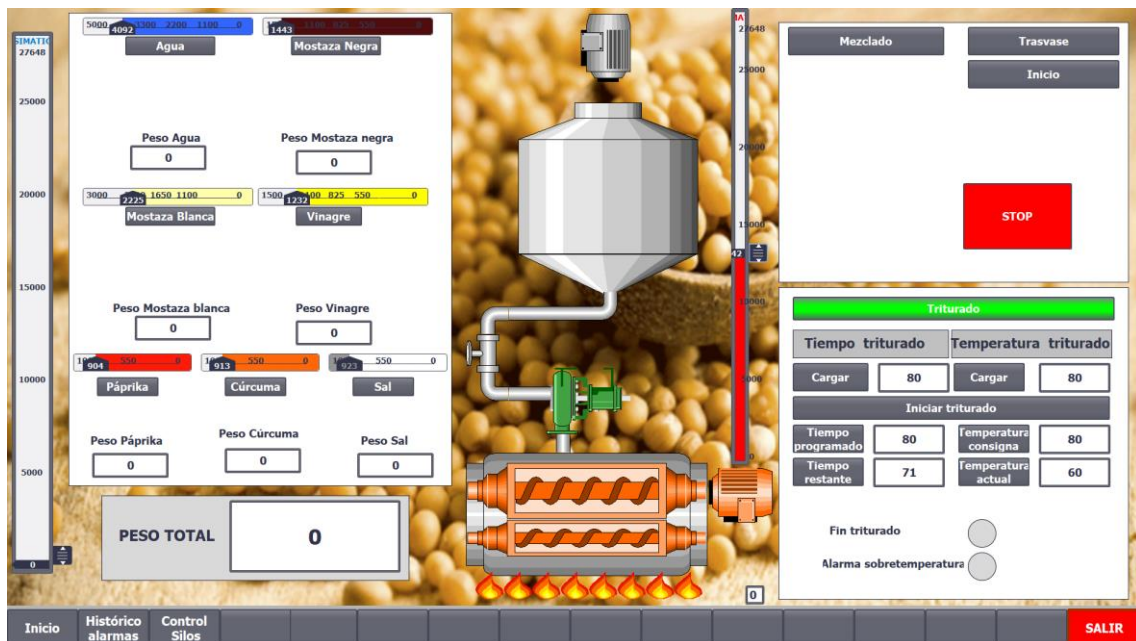


Figura 19 Vista triturado y cocido en funcionamiento en la pantalla "Movimientos manuales"

Al iniciar al triturado, se activarán los elementos de calentamiento y los motores del triturador, que estarán encendidos tanto tiempo como se haya programado.

El control de temperatura será mediante un control por histéresis. Si se compara la figura 19 con la figura 20, se puede observar cómo al sobrepasar ligeramente la temperatura de consigna se han parado los elementos de la calefacción:

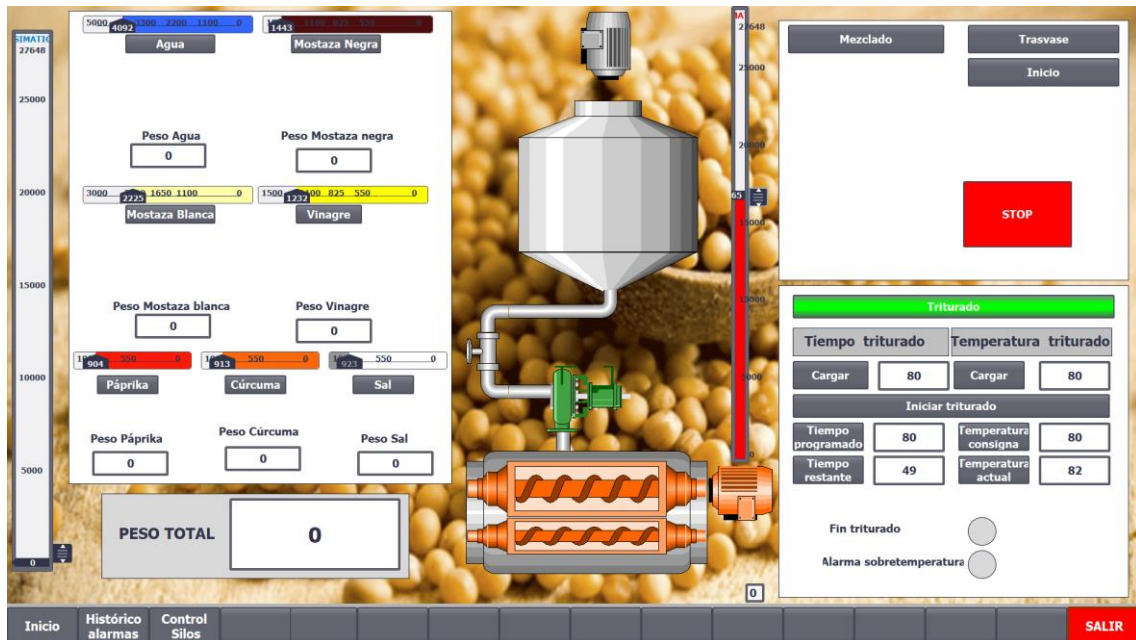


Figura 20 Vista desconexión elementos calefactores durante el triturado en la pantalla "Movimientos manuales"

Al finalizar el tiempo, se desactivará el triturador y los elementos calefactores, y se iluminará el piloto "Fin triturado" informando de que el ciclo está completo:

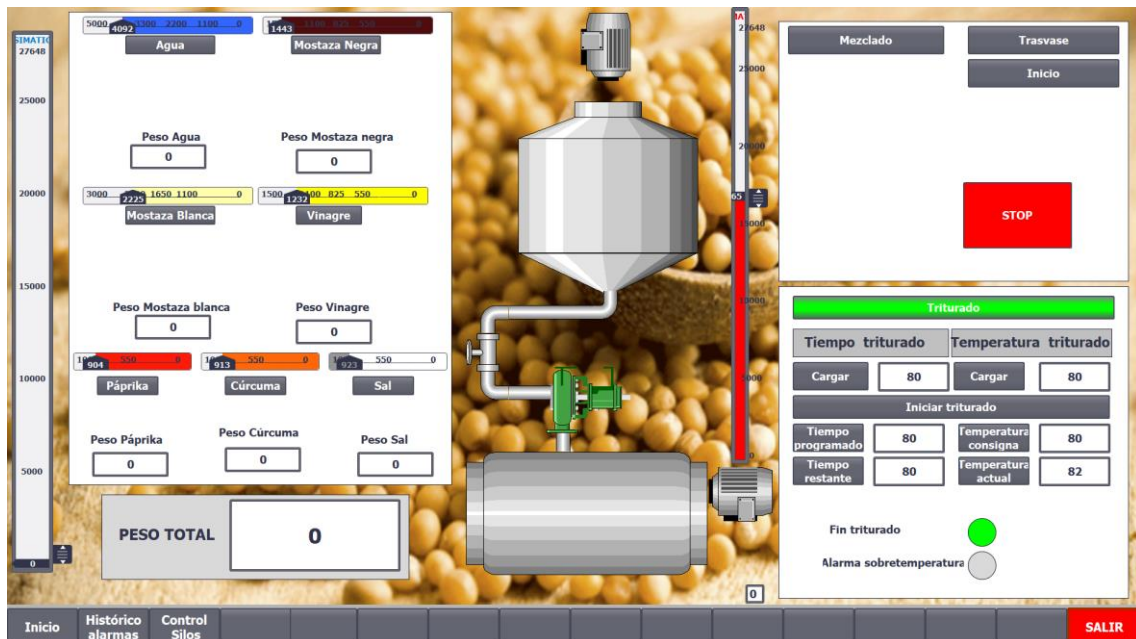


Figura 21 Vista finalización del triturado en la pantalla "Movimientos manuales"

I.II MODO AUTOMÁTICO DESDE EL HMI

Si en el menú principal “Inicio”, se pulsa el botón de “Modo automático” se accede a la siguiente pantalla:

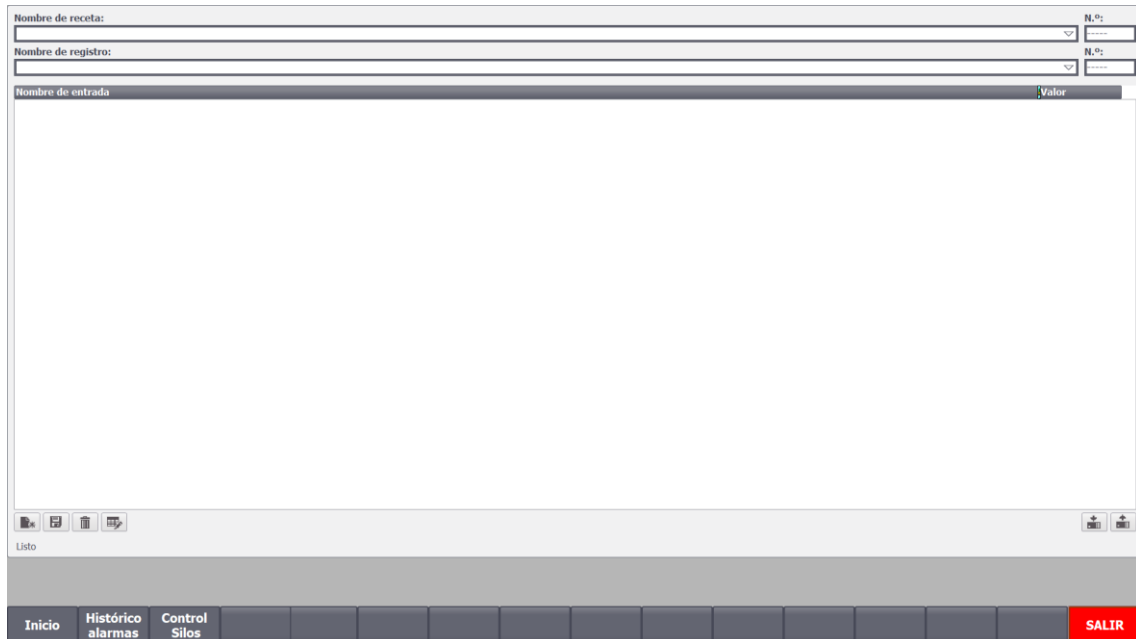


Figura 22 Vista pantalla recetas al seleccionar el "Modo automático"

Para seguir avanzando, se deberá de seleccionar una receta, y cargarla en el PLC.

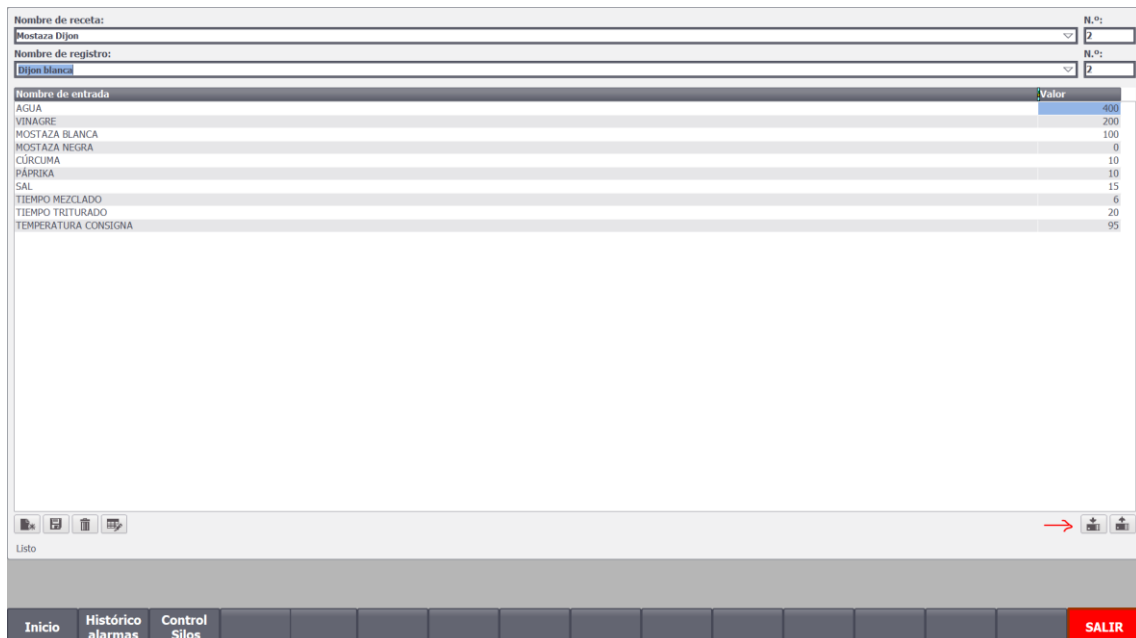


Figura 23 Seleccionar y cargar receta en el "Modo automático"

Una vez el PLC tiene una receta cargada, aparecerá un botón “Ir al modo automático”:

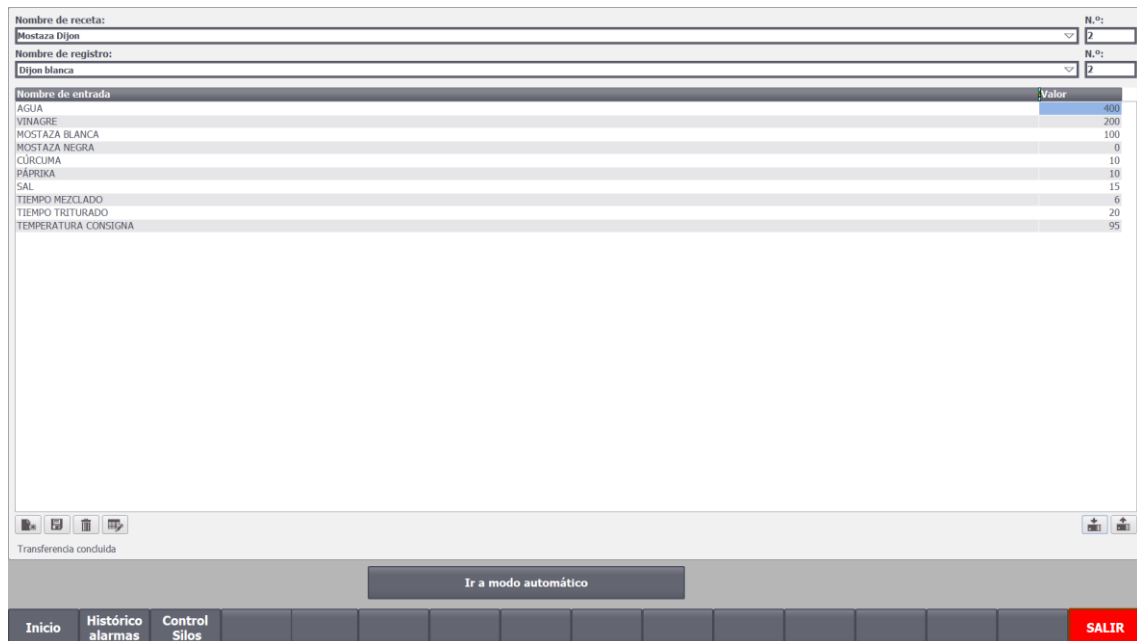


Figura 24 Vista de la receta cargada correctamente en el "Modo automático"

Una vez pulsado este botón aparecerá la siguiente pantalla:

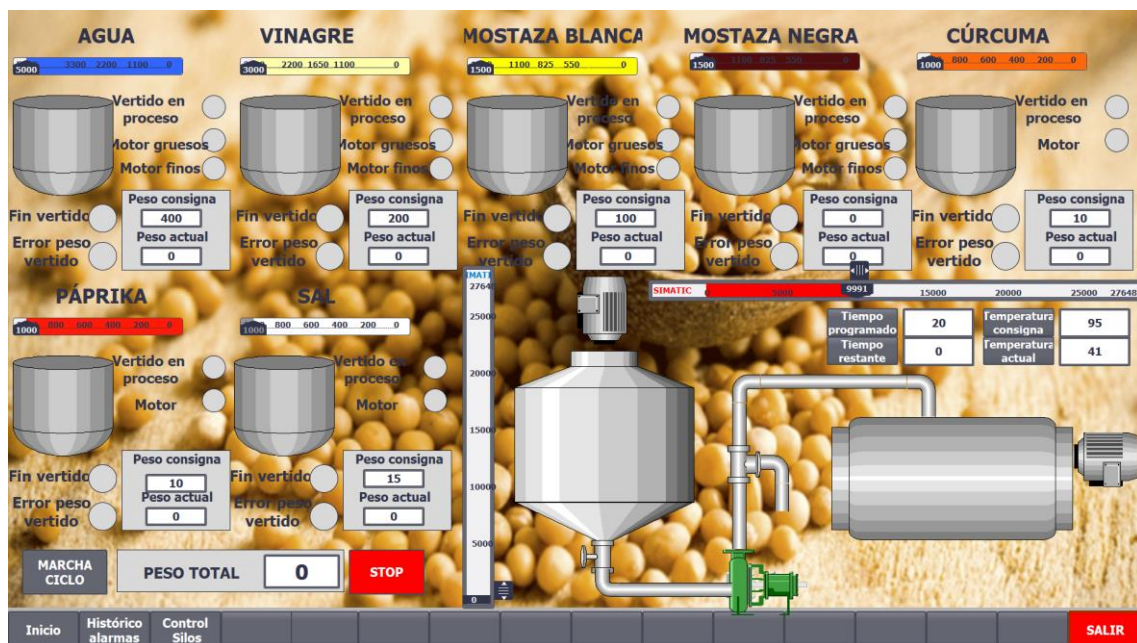


Figura 25 Vista del proceso en el "Modo automático"

Una vez aquí, habrá de pulsarse el botón “Marcha ciclo” para que el ciclo se complete automáticamente. Los pilotos de estado del proceso (como vertido en proceso, o los de los motores) se irán iluminando conforme se vayan activando al avanzar el ciclo. Cuando finalice, aparecerá un aviso en pantalla para informar de que el proceso se ha completado correctamente. Si por el contrario, ha habido un error durante el vertido de algún material, se vaciará el material cargado en el tanque de mezclado tras acusar el fallo.

La figura 26 muestra un ejemplo de como el sistema va vertiendo los materiales uno a uno:

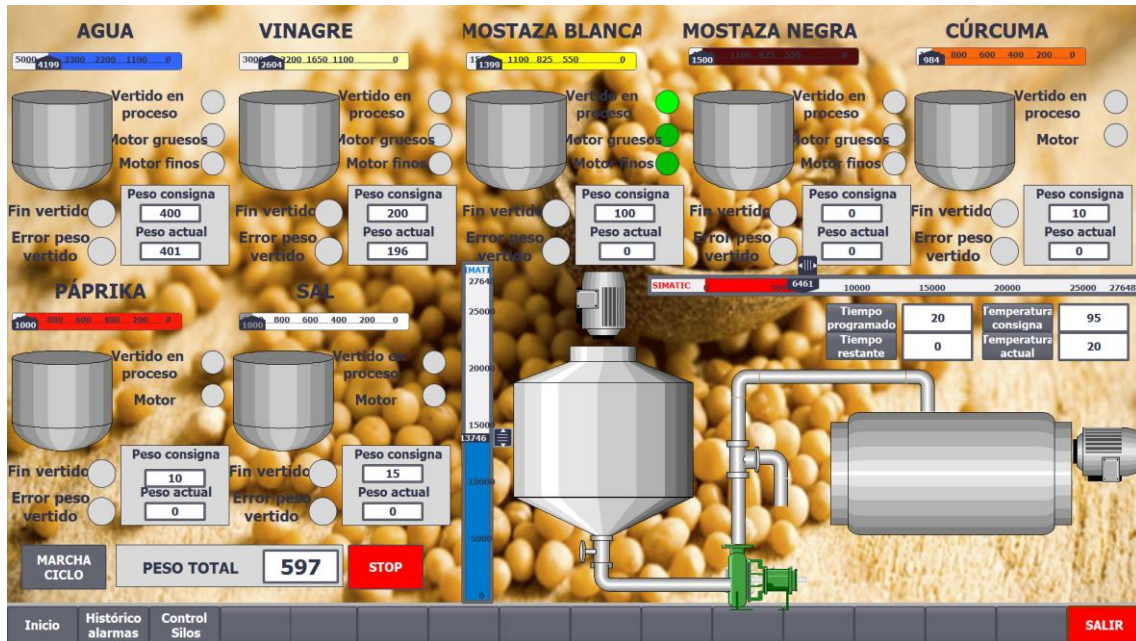


Figura 26 Ejemplo vertido materiales en el "Modo automático"

Una vez se han vertido peso todos los materiales, se activará el mezclador:

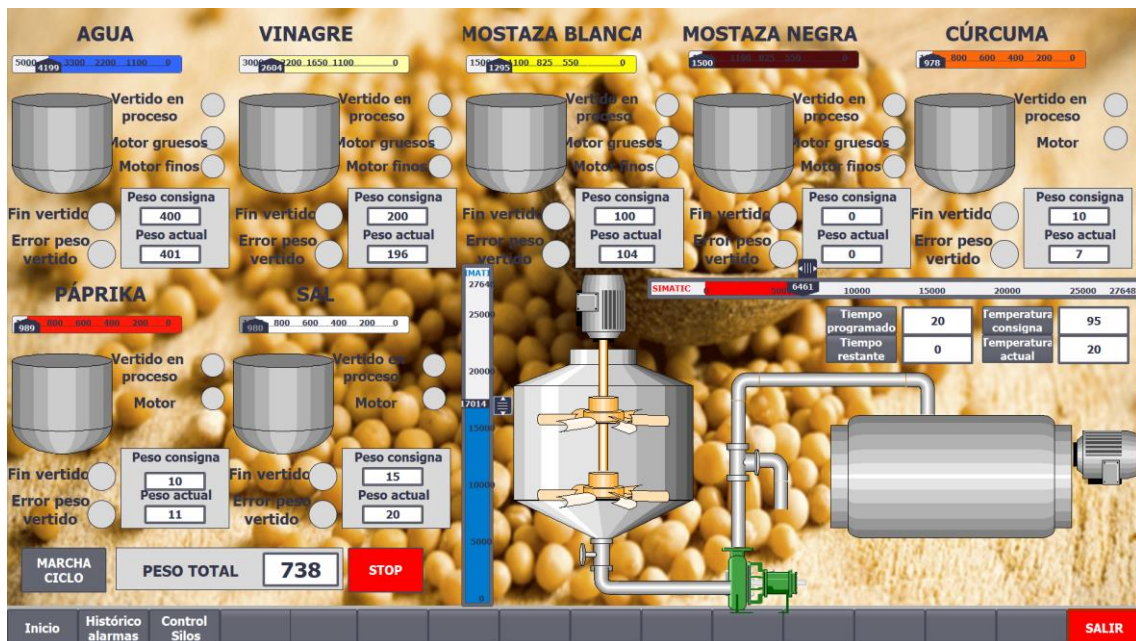


Figura 27 Activación mezclador en el "Modo automático"

Transcurrido el tiempo de mezclado, se abrirá primero la electroválvula de trasvasado e instantes después se arranca la bomba de trasvase:

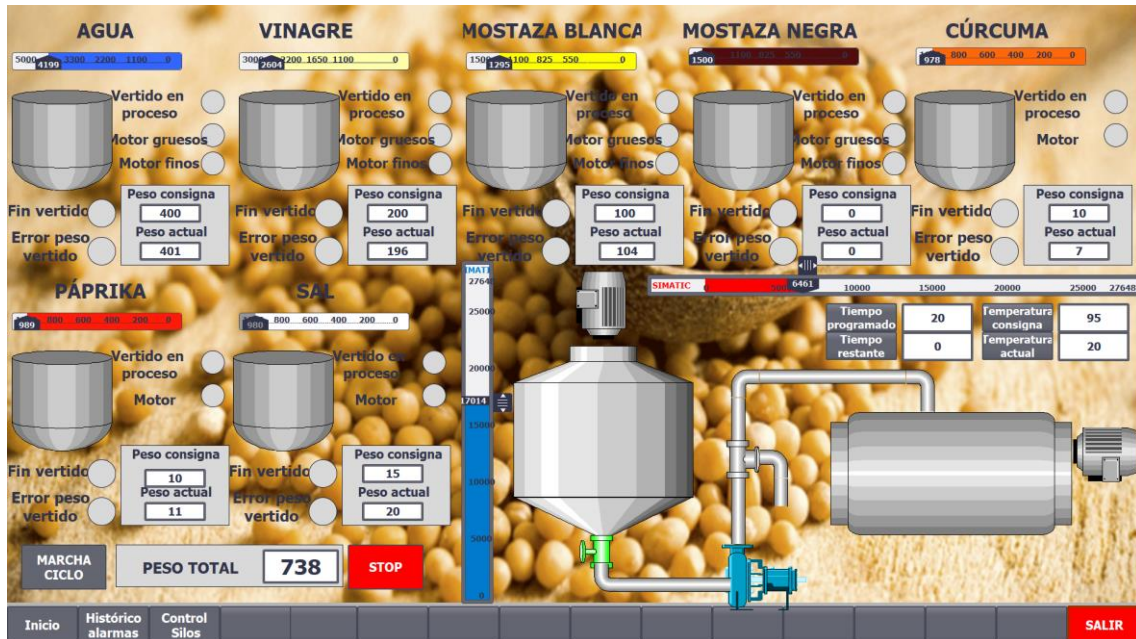


Figura 28 Vista del trasvasado en el "Modo automático"

Cuando el peso de la báscula marque cero, se desactivarán la bomba y la electroválvula y se activará la fase de triturado: el motor del triturador y las resistencias calefactoras.

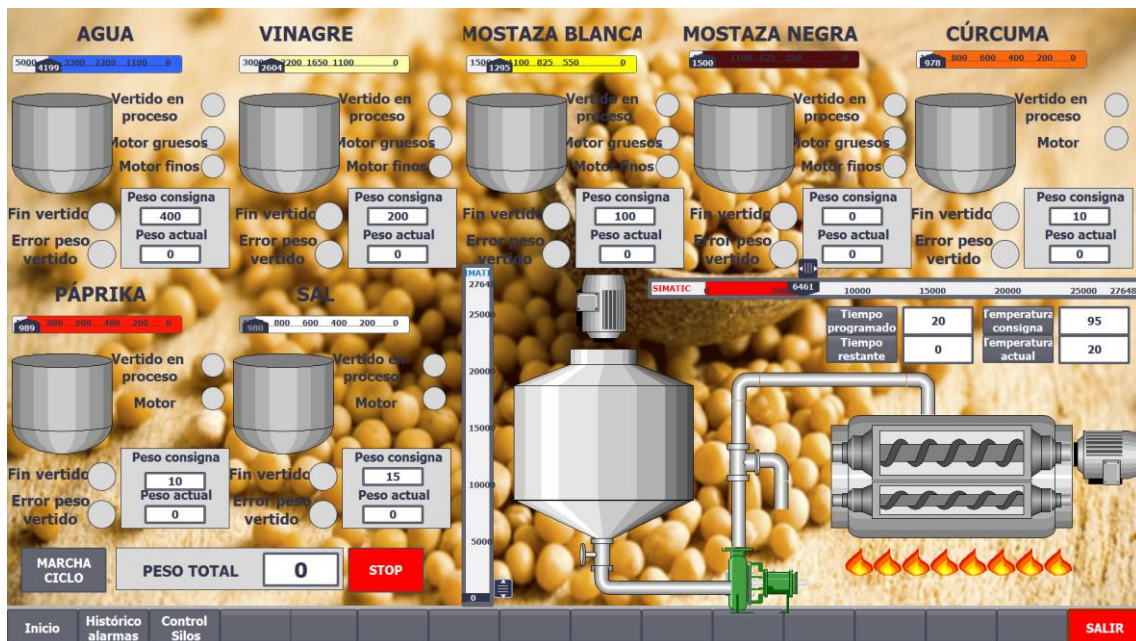


Figura 29 Vista del triturado en el "Modo automático"

Al terminar el tiempo de triturado, es notable que la temperatura está rondando la de consigna (como de cerca dependerá de la parametrización de los silos que se haya establecido en la pantalla correspondiente), y que el motor se ha parado, dando además un aviso de fin de ciclo.



Figura 30 Vista tras la finalización del proceso en el "Modo automático"

Si durante el proceso ha habido un error de vertido, saltará en pantalla el siguiente aviso:

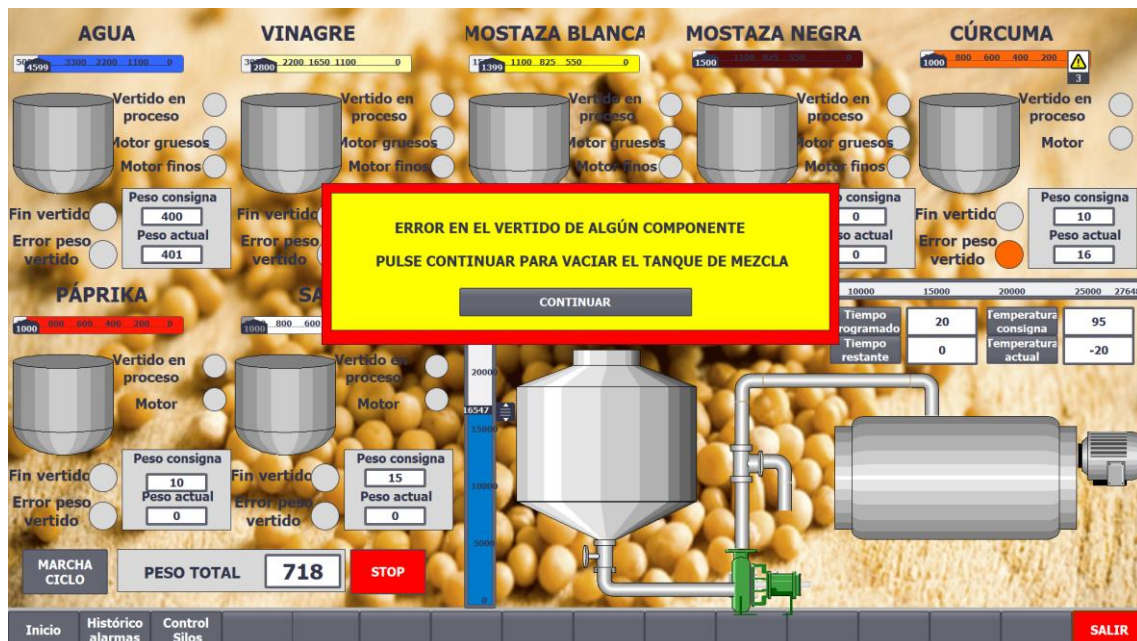


Figura 31 Vista tras producirse un error de vertido en el "Modo automático"

Al pulsar continuar, automáticamente se abrirá la electroválvula de trasvasado, y una segunda electroválvula que normalmente no tiene operación, se activará para desviar el flujo de material hacia el tanque de reciclado:



Figura 32 Vista tras acusar el error de vertido en el "Modo automático"

Una vez el peso de la báscula es cero, el sistema resta las cantidades de material que se habían leído, actualiza las cantidades restantes y desactiva los motores y resetea el fallo de vertido, dejando el sistema listo para el inicio de un nuevo ciclo al pulsar "Marcha inicio".

ANEXO II: CÓDIGO DE LAS FUNCIONES CREADAS Y DEL PROGRAMA DE CONTROL

II.1) FUNCIÓN "SILO"

A continuación se describirá el bloque de función que se ha creado para el control de los actuadores de los silos encargados del vertido del material. Dicho bloque se ha realizado en el lenguaje de programación ST, puesto que dadas la características de la lógica interna del bloque, además de dotar de una mayor claridad al programa a la hora de escribir el algoritmo de control, permite la programación de funciones condicionales de manera más rápida.

Como se ha visto previamente, la ventaja de realizar esta función en un FB, es que al llamarla en el Main, aparecerá una caja cerrada con una serie de conexiones a modo de entradas y salidas. Estas conexiones vendrán definidas por las variables que se declaren dentro del FB.

En este caso se han declarado:

1	▼ Input					
2	valor_actual_peso	Real		30	et6	Time
3	reset_material	Bool		31	et7	Time
4	Peso_consigna	Real		32	aux_1	Bool
5	inicio	Bool		33	aux_2	Bool
6	Reset_fallo_motores	Bool		34	aux_3	Bool
7	paro	Bool		35	aux_4	Bool
8	▶ Parámetros	"Parámetros_silo"		36	aux_5	Bool
9	▼ Output			37	aux_6	Bool
10	motor_gruesos	Bool		38	aux_7	Bool
11	motor_finos	Bool		39	M_N_fallo	Bool
12	nivel_bajo	Bool		40	A_B_fallo	Bool
13	fallo_nivel	Bool		41	Fallo_consigna_peso	Bool
14	Vertido_en_proceso	Bool		42	Fallo_capacidad_max	Bool
15	Fin_vertido	Bool		43	▶ IEC_Tímer_0_Instance	TON_TIME
16	Fallo_motor_gruesos	Bool		44	▶ IEC_Tímer_0_Instance...	TON_TIME
17	Fallo_motor_finos	Bool		45	▶ IEC_Tímer_0_Instance...	TON_TIME
18	Error_vertido	Bool		46	▶ IEC_Tímer_0_Instance...	TON_TIME
19	▼ InOut			47	▶ IEC_Tímer_0_Instance...	TON_TIME
20	cantidad_restante	Real		48	▶ IEC_Tímer_0_Instance...	TON_TIME
21	▼ Static			49	Q	Real
22	marcha_gruesos	Bool		50	W	Real
23	marcha_finos	Bool		51	x	Real
24	puesta_cero	Bool		52	y	Real
25	et1	Time		53	Peso_objetivo	Real
26	et2	Time		54	▼ Temp	
27	et3	Time		55	trig	Bool
28	et4	Time		56	trig_2	Bool
29	et5	Time		57	trig_3	Bool

Figura 33 Variables Función silo

Donde las entradas Input serán datos del sistema de los que se necesita saber el estado, para que en función de este los actuadores se comporten de una forma u otra. Son siempre variables que el bloque solo lee.

Las salidas Output son las relacionadas con los actuadores. Son variables en las que se actúa sobre su estado. Son variables cuyos valores escribe el bloque.

Las InOut son variables en las que el bloque al efectuar la llamada a la función lee, ejecuta el programa en función de los valores leídos, y vuelve a escribir en ellas el resultado obtenido tras la ejecución, por ello se les llama variables de entrada/salida.

Los datos locales tipo Static son variables que sirven para almacenar resultados intermedios estáticos en los bloques de datos de instancia. Estos datos estáticos se conservan a lo largo de varios ciclos de programa, hasta que se vuelven a escribir.

Los datos locales Temp son variables que sirven para almacenar resultados intermedios temporales, y hay que tener cuidado con ellos, puesto que estos datos se conservan únicamente durante un ciclo.

Una vez se ha hecho una breve descripción de los diferentes tipos de variables y de la declaración de las variables, se pasa a exponer y comentar el programa de la función "Silo":

```
#x := #Parámetros.capacidad_max * (#Parámetros.A /100); (*indica que el nivel de material del silo es bajo*)  
#y := #Parámetros.capacidad_max * (#Parámetros.B /100); (*inidica que el silo está a punto de quedar sin material*)
```

#X es un parámetro que calcula y la función para determinar cuál va a ser el valor numérico a partir del cual se considera que hay que dar un aviso de nivel, puesto que el silo está próximo a quedarse sin material, un aviso de nivel bajo. **#Y** será el valor, a partir del cual si se ha ignorado el aviso de nivel bajo, se bloquearán los actuadores del silo en cuestión y se dará un aviso en el HMI, al no disponer de material que verter.

```
#Peso_objetivo := #Peso_consigna - #Parámetros.Cola_de_carga;
```

El **peso objetivo** será el peso al que debe de llegar el sistema para desconectar los actuadores del vertido de material. Esto se ha considerado así para tener en cuenta la cola de carga, que es una cantidad de material determinada que se sabe que cae una vez se da la orden de parada al actuador.

La cola de carga variará en función del material y del tipo de actuador que lleve montado el silo, por lo que es una variable de entrada externa, que será parametrizable desde una ventana de Parámetros en el HMI.

```
IF #paro = 1 THEN
    #motor_finos := 0;
    #motor_gruesos := 0;

END_IF;
IF #inicio = 0 THEN
    #aux_1 := 0;
    #aux_4 := 0;
END_IF;
```

Las variables M y N forman parte de la parametrización del silo comentada anteriormente. En este caso las variables **#M** y **#N** son un número de cero a cien que se empleará para calcular el porcentaje del peso a partir de los cuales se desactivarán los actuadores de vertido. #M para el actuador de mayor capacidad y #N para el más pequeño, por lo que lógicamente el valor de N deberá de ser superior al de M.

```
IF #Parámetros.M >= #Parámetros.N OR #Parámetros.M < 0 OR #Parámetros.N <
0 OR #Parámetros.N>100 THEN (*Asegurar que los valores introducidos de M y N
son correctos*)
    #M_N_fallo := 1;
ELSE
    #M_N_fallo := 0;
END_IF;
```

Como es necesario que el valor sea entre cero y cien, se limitarán las posibilidades de que el usuario introduzca una valor que el programa no entienda. En caso de una introducción errónea de los datos, aparecerá un aviso en el HMI informando de que el valor no es correcto y hasta que esté solventado, no podrá iniciarse el modo automático.

```
IF #Parámetros.capacidad_max <0 THEN (*Asegurar que el valor de la capacidad
máxima del silo es correcto*)
    #Fallo_capacidad_max := 1;
ELSE
    #Fallo_capacidad_max := 0;
END_IF;
```

```
IF #Peso_consigna >#Parámetros.capacidad_max OR #Peso_consigna < 0 THEN
(*Asegurar que el valor del peso consigna es correcto*)
  #Fallo_consigna_peso := 1;
ELSE
  #Fallo_consigna_peso := 0;
END_IF;
```

A continuación aparecen los parámetros #A y #B que son los responsables del control del nivel de material en los silos. El parámetro #A deberá de tener un valor entre cero y cien para calcular el porcentaje a partir del cual se da el **aviso de nivel bajo**. Y el parámetro #B se utilizará para el cálculo del **nivel de fallo de material**, por lo que deberá de tener un valor superior al de #A. Si se llega a una situación en la que se detecta que el sistema está a punto de quedarse sin material dado que el fallo de nivel ha saltado, se impedirá la activación de los actuadores, y se mostrará un aviso en pantalla hasta que el problema quede solventado.

```
IF #Parámetros.B >= #Parámetros.A OR #Parámetros.A < 0 OR #Parámetros.B < 0
OR #Parámetros.A > 100 THEN (*Asegurar que los valores introducidos de A y B
son correctos*)
  #A_B_fallo := 1;
ELSE
  #A_B_fallo := 0;
END_IF;
```

```
IF #cantidad_restante <= #x AND #cantidad_restante > #y THEN (*Aviso nivel bajo*)
  #nivel_bajo := 1;
ELSE
  #nivel_bajo := 0;
END_IF;
```

```
IF #cantidad_restante <= #y OR #cantidad_restante < #Peso_consigna THEN
(*Alarma fallo nivel*)
  #fallo_nivel := 1;
ELSE
  #fallo_nivel := 0;
END_IF;
```

```
IF #fallo_nivel = 1 THEN
  #motor_gruesos := 0;
  #motor_finos := 0;
  #Vertido_en_proceso := 0;
```

```
#Error_vertido := 0;
#Fin_vertido := 0;
ELSE
  #paro := 0;
END_IF;
```

Cuando el nivel del material sea bajo, al rellenar los silos, se deberá de informar al sistema pulsando **#reset_material**. Esto hará que el valor que tome el programa sea el valor de la capacidad máxima del silo. Lo que obliga a que cada vez que se rellene el material se haga de manera completa.

```
IF #reset_material = 1 THEN
  #cantidad_restante := #Parámetros.capacidad_max;
END_IF;
```

(*M es un parámetro para tener en cuenta y ajustar la desconexión relativa al motor de gruesos N es lo mismo pero para el motor de finos*)

La variable #W se explicará más adelante, a grandes rasgos es un valor límite superior por debajo del cual se considera la pesada correcta.

```
IF (#valor_actual_peso <= #W) THEN
```

Por lo que la activación de los actuadores (llamados motores en la función) se dará cuando el valor leído del peso actual sea menor que el límite superior #W, no se tengan fallos en la parametrización de los silos, y a la función se le haya requerido el inicio.

En el caso del actuador me mayor tamaño, se activará hasta que el peso supere el valor del porcentaje del peso consigna calculado anteriormente por la variable M, y lo mismo ocurrirá con el actuador de menor tamaño y la variable N.

Por ejemplo, si se tiene un peso consigna de 100 kg, y el valor de M = 80 y el valor de N = 90, significa que el motor de gruesos permanecerá activo hasta que el peso del material alcance el 80% del peso consigna y el motor de finos hasta el 90% del peso consigna. Estos valores deberán de ajustarse teniendo en cuenta la cola de carga.

```
IF #valor_actual_peso <= ((#Parámetros.M / 100) * #Peso_objetivo) AND
#Peso_objetivo>0 AND #fallo_nivel = 0 AND #M_N_fallo = 0 AND #A_B_fallo = 0
AND #Fallo_consigna_peso = 0 AND #Fallo_capacidad_max = 0 AND #inicio = 1
AND #Reset_fallo_motores=0 THEN
```

```
    #motor_gruesos := 1;
    #aux_2 := 1;
```

```
ELSE
```

```
    #motor_gruesos := 0;
    #aux_2 := 0;
```

```
END_IF;
```

```
IF #valor_actual_peso <= ((#Parámetros.N / 100) * #Peso_objetivo) AND
#Peso_objetivo>0 AND #fallo_nivel = 0 AND #M_N_fallo = 0 AND #A_B_fallo = 0
AND #Fallo_consigna_peso = 0 AND #Fallo_capacidad_max = 0 AND #inicio = 1
AND #Reset_fallo_motores=0 THEN
```

```
    #motor_finos := 1;
    #aux_3 := 1;
```

```
ELSE
```

```
    #motor_finos := 0;
    #aux_3 := 0;
```

```
END_IF;
```

```
END_IF;
```

Las variables auxiliares #aux_2 y #aux_3 se activan y desactivan conjuntamente con los actuadores, por lo que si estos permanecen encendidos más tiempo del estimado, se considerará que ha habido algún problema con estos actuadores y saltará un aviso a través de la pantalla. Si por algún motivo se llegara a esta situación, al saltar el aviso se desactivarán los motores.

(*TEMPORIZACIÓN PARA FALLO DE LOS MOTORES Y RESET FALLO MOTORES*)
(*Temporizadores para el control de los motores, si pasan demasiado tiempo encendidos y no llegan al peso, salta la alarma*)

```
#IEC_Timer_0_Instance(IN:=#aux_2,
    PT:=#Parámetros.tiempo_gruesos,
    Q=>#Fallo_motor_gruesos,
    ET=>#et3);
```

```
#IEC_Timer_0_Instance_1(IN:=#aux_3,
    PT:=#Parámetros.tiempo_finos,
    Q=>#Fallo_motor_finos,
    ET=>#et4);
```

```
IF #Fallo_motor_gruesos=1 THEN
    #motor_gruesos := 0;
END_IF;
IF #Fallo_motor_finos = 1 THEN
    #motor_finos := 0;
END_IF;
```

```
IF #Reset_fallo_motores = 1 THEN
    #Fallo_motor_finos := 0;
    #Fallo_motor_gruesos := 0;
    #aux_2 := 0;
    #aux_3 := 0;
    #paro := 1;
END_IF;
```

En la pantalla, para informar del estado del vertido, aparecerá un piloto que se mantendrá activo siempre que alguno de los dos actuadores esté encendido.

```
IF (#motor_finos = 1 OR #motor_gruesos = 1) THEN
    #Vertido_en_proceso := 1;
ELSE
    #Vertido_en_proceso := 0;
END_IF;
```

Para la activación de la retroalimentación que señala que el vertido se ha realizado correctamente se han declarado dos variables. La #W será el valor del límite superior por debajo del cual se tomará el pesaje como correcto y la #Q será el valor del límite inferior por encima del cual se tomará el pesaje como válido.

```
(*Q y W son parámetros para controlar la activación de la salida "fin vertido"*)
#Q := (1-#Parámetros.Tolerancia_pesado/100) * #Peso_consigna;
#W := (1+#Parámetros.Tolerancia_pesado/100) * #Peso_consigna;
```

Una vez se ha realizado el vertido, si se tiene material en la báscula dentro del rango admitido para dar el pesaje por bueno, y los actuadores están detenidos, se desactivará la retroalimentación de “vertido en proceso” y se activará otra con el texto “fin de

vertido". Esta última permanecerá activa un tiempo programado, y al desactivarse permitirá resetear los estados de la función y realizará la resta del material que haya actualmente en la báscula a la cantidad de material que se tenía en el silo al iniciar el vertido.

```
IF #aux_1 = 1 THEN
  #Fin_vertido := 0;
  #trig := 0;
  #cantidad_restante := #cantidad_restante - #valor_actual_peso;
  #valor_actual_peso := 0;
END_IF;
```

```
IF (#inicio = 1 AND #valor_actual_peso > #Q AND #valor_actual_peso < #W AND
  #Vertido_en_proceso = 0 AND #aux_1 = 0 AND #Error_vertido=0) THEN
  #Fin_vertido := 1;
ELSE
  #Fin_vertido := 0;
  IF (#inicio = 1 AND #Peso_consigna = 0 AND
    #Vertido_en_proceso = 0 AND #aux_1 = 0) THEN
    #Fin_vertido := 1;
  END_IF;
END_IF;
#IEC_Timer_0_Instance_2(IN := #Fin_vertido AND NOT #aux_1,
  PT := t#1s,
  Q => #trig,
  ET => #et1);

#IEC_Timer_0_Instance_3(IN := #trig,
  PT := t#4s,
  Q => #aux_1,
  ET => #et2);
```

Si por cualquier motivo se supera el límite superior del peso admitido como correcto #W, se dará un fallo de vertido, para informar de que la cantidad volcada en el tanque no es la adecuada. La gestión del error del vertido se hará externamente a la función "Silo".

```

IF (#inicio = 1 AND #valor_actual_peso > #W AND
    #Vertido_en_proceso = 0 AND #aux_4 = 0 AND #Fin_vertido = 0) THEN
    #Error_vertido := 1;

ELSE
    #Error_vertido := 0;

END_IF;
    
```

Vista la lógica interna del programa, se muestra cómo queda el bloque al realizar la llamada de la función:

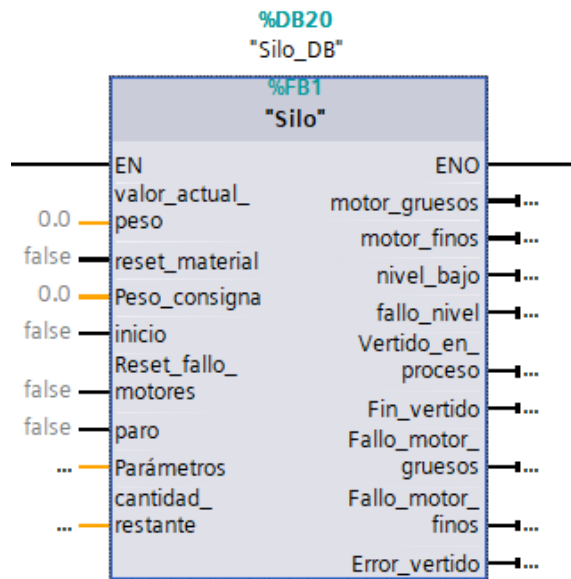


Figura 34 Vista función Silo II

Los datos asociados a estas entradas y salidas deberán de guardarse en FB's, tantos como llamadas se realicen. En este caso se tendrán siete, pues tal es la cantidad de materias primas que se quiere controlar:

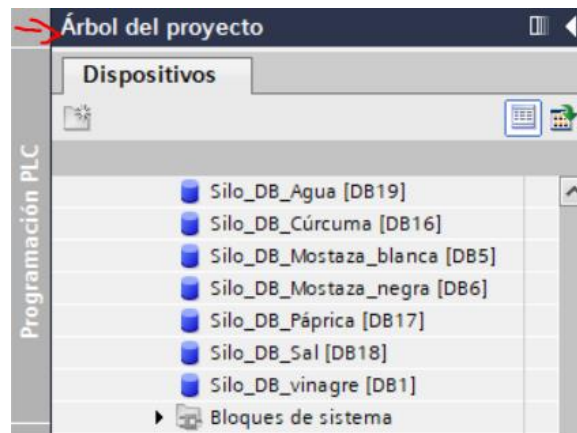


Figura 35 Vista de los DB's de las funciones Silo en el árbol del proyecto

El DB asociado a la función será un espejo de las variables que se han declarado en la FB. Estos DB's los crea el TIA Portal automáticamente cada vez que se introduce la función en el MAIN.

Esta es una de las grandes ventajas que aporta el empleo de los FB's. Se declaran las variables y se escribe la lógica una vez. Luego se podrá llamar a la función tantas veces como sea necesario y además se le especificará dónde se quieren guardar los datos asociados a cada función.

	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de a...
1	Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	valor_actual_peso	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	reset_material	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Peso_consigna	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	inicio	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Reset_fallo_motores	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	paro	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Parámetros	*Parámetros_silo*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	motor_gruesos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	motor_finos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	nivel_bajo	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	fallo_nivel	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Vertido_en_proceso	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Fin_vertido	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Fallo_motor_gruesos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Fallo_motor_finos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Error_vertido	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	InOut			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20	cantidad_restante	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	marcha_gruesos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	marcha_finos	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 36 Vista del interior de un DB de la función Silo

II.II) GESTIÓN DE LAS ALARMAS GENERADAS POR EL PROCESO

Para la gestión de las alarmas, se ha creado una función tipo FC.

En el FC “Alarmas_y_fallos_SILOS” se va a leer el estado de algunas entradas y salidas de la función silo, así como de otras funciones que se han explicado en este documento, de las que se tiene necesidad de controlar algunas de las variables.

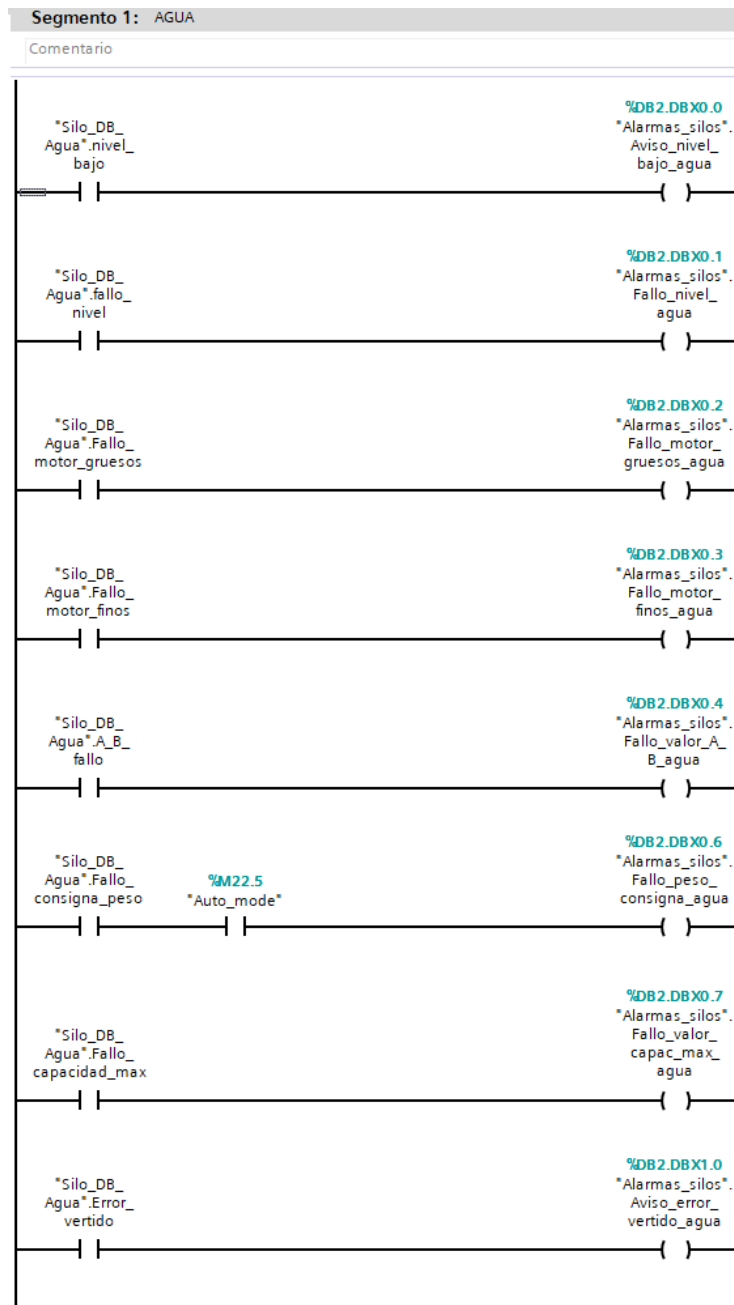


Figura 37 Programación segmento 1 “Alarmas_y_fallos_SILOS”

En la figura 37 lo que se ve es como la función lee los estados de las variables del bloque de función “silo” y según sea su estado activa o desactiva las alarmas, en este caso del silo que almacena el agua. Se tiene pues tantos segmentos como el anterior como silos, además de las alarmas de temperatura o error de vertido.

Se han organizado las alarmas en una FC para dotar de mayor claridad del programa. De esta forma se tienen todas las alarmas en una sola función a la que se ha asociado un DB para realizar la lectura desde el HMI. Se tienen todas las alarmas juntas en el DB asociado a esta FC.

	Nombre	Tipo de datos	Offset	Valor ...	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...
1	Static								
2	Aviso_nivel_bajo_agua	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Fallo_nivel_agua	Bool	0.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Fallo_motor_gruesos_agua	Bool	0.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Fallo_motor_finos_agua	Bool	0.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Fallo_valor_A_B_agua	Bool	0.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Fallo_valor_M_N_agua	Bool	0.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Fallo_peso_consigna_agua	Bool	0.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Fallo_valor_capac_max_agua	Bool	0.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Aviso_error_vertido_agua	Bool	1.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Aviso_nivel_bajo_vinagre	Bool	1.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Fallo_nivel_vinagre	Bool	1.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Fallo_motor_gruesos_vinagre	Bool	1.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Fallo_motor_finos_vinagre	Bool	1.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Fallo_valor_A_B_vinagre	Bool	1.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Fallo_valor_M_N_vinagre	Bool	1.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Fallo_peso_consigna_vinagre	Bool	1.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Fallo_valor_capac_max_vinagre	Bool	2.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Aviso_error_vertido_vinagre	Bool	2.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Aviso_nivel_bajo_mostaza_blanca	Bool	2.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	Fallo_nivel_mostaza_blanca	Bool	2.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	Fallo_motor_gruesos_mostaza_blanca	Bool	2.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	Fallo_motor_finos_mostaza_blanca	Bool	2.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	Fallo_valor_A_B_mostaza_blanca	Bool	2.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25	Fallo_valor_M_N_mostaza_blanca	Bool	2.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26	Fallo_peso_consigna_mostaza_blanca	Bool	3.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
27	Fallo_valor_capac_max_mostaza_blanca	Bool	3.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
28	Aviso_error_vertido_mostaza_blanca	Bool	3.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 38 Vista interior del DB "Alarmas_silos" (bis)

Además de las alarmas relacionadas con la función silo, se tendrán las relacionadas con el control de temperatura y la parametrización de los valores iniciales:

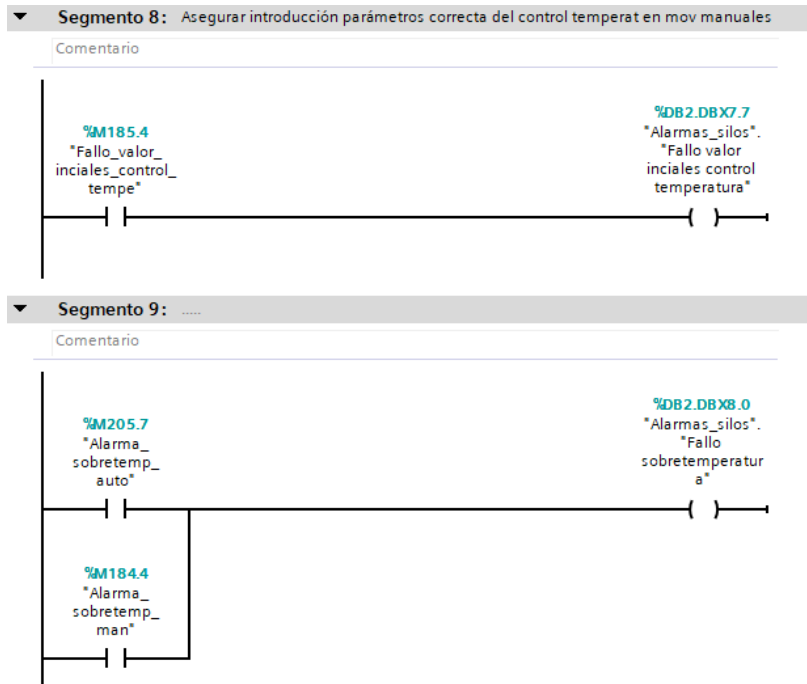


Figura 39 Programación segmento 8 y 9 “Alarmas_y_fallos_SILOS”

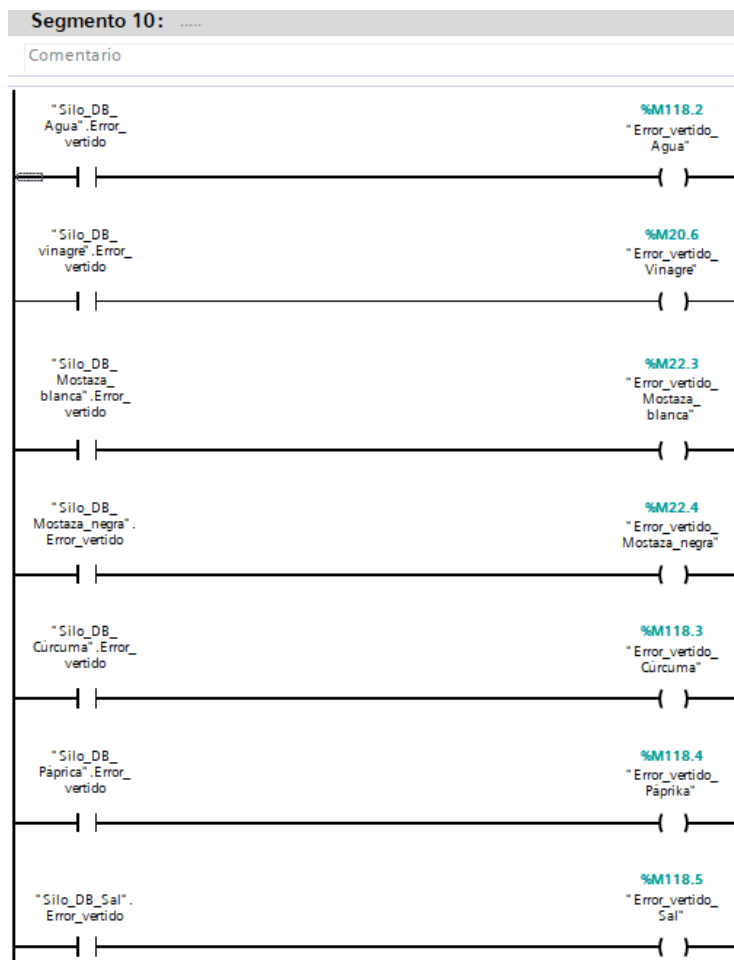


Figura 40 Programación segmento 10 “Alarmas_y_fallos_SILOS”

II.III) FUNCIÓN “ESC_SEÑAL_ANALOG”

Como únicamente se dispone de una báscula para todo el proceso, por lo que se hace necesario trabajar y adaptar la señal del sensor de presión a los distintos pasos del vertido además, de tener que depositar los materiales de uno en uno para controlar el peso.

Con estas dos condiciones (una báscula y añadir las materias primas de una en una) se ha realizado un FB para el escalado de la señal analógica. Las variables que se han declarado son:

Esc_señal_analog							
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Señal_analógica_sensc	Int	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Proceso_1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Proceso_2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Proceso_3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Proceso_4	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Proceso_5	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Proceso_6	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Proceso_7	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Proceso_8	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	MAX	Real	0.0	Remanente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	MIN	Real	0.0	Remanente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Reset_memo	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	señal_escalada_S1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	señal_escalada_S2	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	señal_escalada_S3	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	señal_escalada_S4	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	señal_escalada_S5	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	señal_escalada_S6	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	señal_escalada_S7	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	señal_escalada_S8	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	<Agregar>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	señal_normalizada_S1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	señal_normalizada_S2	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	señal_normalizada_S3	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	señal_normalizada_S4	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	señal_normalizada_S5	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
31	señal_normalizada_S6	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	señal_normalizada_S7	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	señal_normalizada_S8	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	Señal_peso_total_esc...	Real	0.0	No rem...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	Señal_peso_total_nor...	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	Señal_analógica_total	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 41 Variables declaradas en la función “Esc_señal_analog”

Al igual que para el FB de la función “Silo”, esta se ha realizado en ST. Se pasa a exponer y comentar las líneas del programa:

Por jerarquía, cuando se dé a la activación del reset de los pesos, se pondrá un cero en todas las señales, lo que se traducirá en un forzado a cero del valor mostrado por el programa. Esto se empleará cuando se ha terminado el ciclo del vertido y se ha realizado la resta del material pesado al material restante que queda en el silo.

```
IF #Reset_memo=1 THEN  
  
    #señal_escalada_S1 := 0;  
  
    #señal_escalada_S2 := 0;  
  
    #señal_escalada_S3 := 0;  
  
    #señal_escalada_S4 := 0;  
  
    #señal_escalada_S5 := 0;  
  
    #señal_escalada_S6 := 0;  
  
    #señal_escalada_S7 := 0;  
  
    #señal_escalada_S8 := 0;  
  
END_IF;
```

En esta función cuando se activa el proceso que se quiere pesar, se permiten las lecturas y escrituras en las variables que intervienen en el IF. Lo primero que se hace es convertir la señal analógica del sensor (de valor 0...27648) a una señal normalizada entre cero y uno. Luego, esta señal la se escalará entre el mínimo y el máximo peso que soporte la báscula, y para discriminar las señales de otros procesos, se deberá de restar el valor que tengan a la señal que se está midiendo.

A la función se le ha dado un tratamiento de genérica. Es decir, el objetivo es que se pueda reutilizar en otros proyectos, cogiéndola desde una librería. Como función genérica, permite que con una sola señal se pueda distinguir el peso de hasta 8 materiales distintos siempre y cuando se conecte correctamente el bloque en el MAIN.

```
IF #Proceso_1=1 THEN
```

```
    #señal_normalizada_S1 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S1 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S1, MAX := #MAX)-  
    #señal_escalada_S8 - #señal_escalada_S7 -#señal_escalada_S6 -#señal_escalada_S5 -  
    #señal_escalada_S4 - #señal_escalada_S3 - #señal_escalada_S2;
```

```
ELSE
```

```
    #Proceso_1 := 0;
```

```
END_IF;
```

```
IF #Proceso_2=1 THEN
```

```
    #señal_normalizada_S2 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S2 :=SCALE_X(MIN := 0, VALUE := #señal_normalizada_S2, MAX := #MAX)-  
    #señal_escalada_S8 - #señal_escalada_S7 -#señal_escalada_S6 -#señal_escalada_S5 -  
    #señal_escalada_S4 - #señal_escalada_S3 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_2 := 0;
```

```
END_IF;
```

```
IF #Proceso_3 =1 THEN
```

```
    #señal_normalizada_S3 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S3 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S3, MAX := #MAX)-  
    #señal_escalada_S8 - #señal_escalada_S7 -#señal_escalada_S6 -#señal_escalada_S5 -  
    #señal_escalada_S4 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_3 := 0;
```

```
END_IF;
```

```
IF #Proceso_4 = 1 THEN
```

```
    #señal_normalizada_S4 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S4 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S4, MAX := #MAX) -  
    #señal_escalada_S8 - #señal_escalada_S7 - #señal_escalada_S6 - #señal_escalada_S5 -  
    #señal_escalada_S3 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_4 := 0;
```

```
END_IF;
```

```
IF #Proceso_5 = 1 THEN
```

```
    #señal_normalizada_S5 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S5 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S5, MAX := #MAX) -  
    #señal_escalada_S8 - #señal_escalada_S7 - #señal_escalada_S6 - #señal_escalada_S4 -  
    #señal_escalada_S3 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_5 := 0;
```

```
END_IF;
```

```
IF #Proceso_6 = 1 THEN
```

```
    #señal_normalizada_S6 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S6 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S6, MAX := #MAX) -  
    #señal_escalada_S8 - #señal_escalada_S7 - #señal_escalada_S5 - #señal_escalada_S4 -  
    #señal_escalada_S3 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_6 := 0;
```

```
END_IF;
```

```
IF #Proceso_7 = 1 THEN
```

```
    #señal_normalizada_S7 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S7 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S7, MAX := #MAX) -  
    #señal_escalada_S8 - #señal_escalada_S6 - #señal_escalada_S5 - #señal_escalada_S4 -  
    #señal_escalada_S3 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_7 := 0;
```

```
END_IF;
```

```
IF #Proceso_8 = 1 THEN
```

```
    #señal_normalizada_S8 := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
    #señal_escalada_S8 := SCALE_X(MIN := 0, VALUE := #señal_normalizada_S8, MAX := #MAX) -  
    #señal_escalada_S7 - #señal_escalada_S6 - #señal_escalada_S5 - #señal_escalada_S4 -  
    #señal_escalada_S3 - #señal_escalada_S2 - #señal_escalada_S1;
```

```
ELSE
```

```
    #Proceso_8 := 0;
```

```
END_IF;
```

En esta parte, el programa se asegura de que si por error se ha quedado algún proceso activado y la señal escalada en ese momento baja de cero, que no muestre por pantalla un número negativo.

```
IF #señal_escalada_S1 < 0 THEN  
    #señal_escalada_S1 := 0;  
END_IF;
```

```
IF #señal_escalada_S2 < 0 THEN  
    #señal_escalada_S2 := 0;  
END_IF;
```

```
IF #señal_escalada_S3 < 0 THEN  
    #señal_escalada_S3 := 0;  
END_IF;
```

```
IF #señal_escalada_S4 < 0 THEN  
    #señal_escalada_S4 := 0;  
END_IF;
```

```
IF #señal_escalada_S5 < 0 THEN  
    #señal_escalada_S5 := 0;  
END_IF;
```

```
IF #señal_escalada_S6 < 0 THEN  
    #señal_escalada_S6 := 0;  
END_IF;
```

```
IF #señal_escalada_S7 < 0 THEN  
    #señal_escalada_S7 := 0;  
END_IF;
```



```
IF #señal_escalada_S8 < 0 THEN
```

```
    #señal_escalada_S8 := 0;
```

```
END_IF;
```

Por último, se tiene acceso a la señal de peso total, que da la suma de todos los pesos de los procesos juntos. Esto es muy útil a la hora de mostrar información por pantalla, pues permite saber qué cantidad total de producto se va a obtener.

```
#Señal_peso_total_normalizada := NORM_X(MIN := 0, VALUE := #Señal_analógica_sensor, MAX := 27648);
```

```
#Señal_peso_total_escalada:= SCALE_X(MIN := 0, VALUE := #Señal_peso_total_normalizada, MAX := #MAX);
```

Así pues, al igual que para la función "Silo", al llamarla en el MAIN aparecerá el FB mostrado en la figura 42:

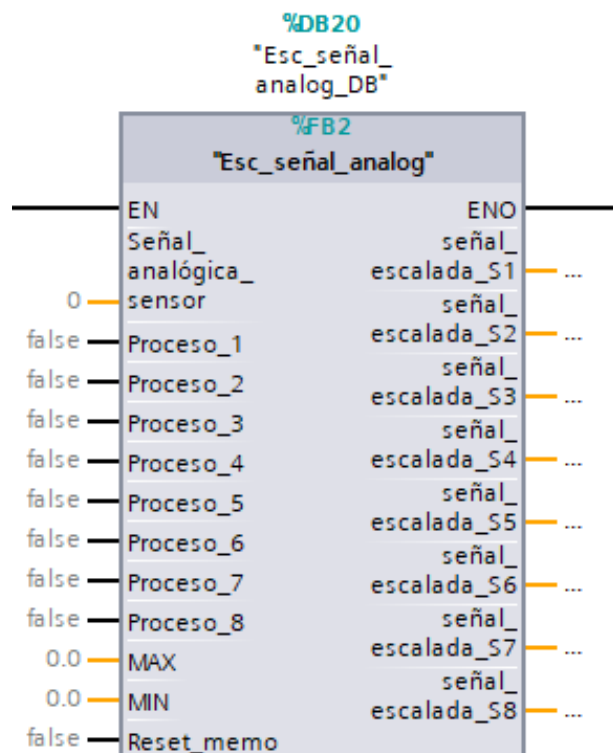


Figura 42 Vista función "Esc_señal_analog" (bis)

II.IV) FUNCIÓN "MOVIMIENTOS_MANUALES"

A continuación se procederá a detallar el funcionamiento de la función "Movimientos_manuales":

Movimientos_manuales							
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..
1	[-] Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	[-] [] Marcha_finos_S1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	[-] [] Marcha_finos_S2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	[-] [] Marcha_finos_S3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	[-] [] Marcha_gruesos_S1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	[-] [] Marcha_gruesos_S2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	[-] [] Marcha_gruesos_S3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	[-] [] Silo_1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	[-] [] Silo_2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	[-] [] Silo_3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	[-] Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	[-] [] Material_1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	[-] [] Material_2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	[-] [] Material_3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	[-] [] Motor_finos_S1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	[-] [] Motor_finos_S2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	[-] [] Motor_finos_S3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	[-] [] Motor_gruesos_S1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	[-] [] Motor_gruesos_S2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	[-] [] Motor_gruesos_S3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	[-] InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	[-] [] <Agrega>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23	[-] Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	[-] [] proceso_1	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	[-] [] proceso_2	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	[-] [] proceso_3	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 43 Variables declaradas en la función "Movimientos_manuales"

Lo que se pretende con esta función es asegurar que se ha seleccionado el tipo de material que se quiere pesar, y que los actuadores y procesos del resto de materiales permanecerán desactivados mientras tanto.

Se repetirá esto para los 8 materiales:

```
(*Obligación a que los silos trabajen de uno en uno*)
(*PROCESO 1*)
IF #Silo_1 = 1 THEN
  #Silo_2 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_7 := 0; #Silo_8 := 0;
  #proceso_1 := 1;

  IF #Marcha_finos_S1 = 1 THEN
    #Motor_finos_S1 := 1;
  ELSE
    #Motor_finos_S1 := 0;
  END_IF;

  IF #Marcha_gruesos_S1 = 1 THEN
    #Motor_gruesos_S1 := 1;
  ELSE
    #Motor_gruesos_S1 := 0;
  END_IF;

ELSE
  #proceso_1 := 0;
  #Motor_finos_S1 := 0;
  #Motor_gruesos_S1 := 0;
  #Marcha_finos_S1 := 0;
  #Marcha_gruesos_S1 := 0;
END_IF;
(*PROCESO 2*)
IF #Silo_2 = 1 THEN
  #Silo_1 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_7 := 0; #Silo_8 := 0;
  #proceso_2 := 1;
  IF #Marcha_finos_S2 = 1 THEN
    #Motor_finos_S2 := 1;
  ELSE
    #Motor_finos_S2 := 0;
  END_IF;
  IF #Marcha_gruesos_S2 = 1 THEN
    #Motor_gruesos_S2 := 1;
  ELSE
    #Motor_gruesos_S2 := 0;
  END_IF;
ELSE
  #proceso_2 := 0;
  #Motor_finos_S2 := 0;
  #Motor_gruesos_S2 := 0;
  #Marcha_finos_S2 := 0;
  #Marcha_gruesos_S2 := 0;
END_IF;
```

(*PROCESO 3*)

IF #Silo_3 = 1 THEN

#Silo_1 := 0; #Silo_2 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_7 := 0; #Silo_8 := 0;
#proceso_3 := 1;

IF #Marcha_finos_S3 = 1 THEN

#Motor_finos_S3 := 1;

ELSE

#Motor_finos_S3 := 0;

END_IF;

IF #Marcha_gruesos_S3 = 1 THEN

#Motor_gruesos_S3 := 1;

ELSE

#Motor_gruesos_S3 := 0;

END_IF;

ELSE

#proceso_3 := 0;

#Motor_finos_S3 := 0;

#Motor_gruesos_S3 := 0;

#Marcha_finos_S3 := 0;

#Marcha_gruesos_S3 := 0;

END_IF;

(*PROCESO 4*)

IF #Silo_4 = 1 THEN

#Silo_1 := 0; #Silo_2 := 0; #Silo_3 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_7 := 0; #Silo_8 := 0;
#proceso_4 := 1;

IF #Marcha_finos_S4 = 1 THEN

#Motor_finos_S4 := 1;

ELSE

#Motor_finos_S4 := 0;

END_IF;

IF #Marcha_gruesos_S4 = 1 THEN

#Motor_gruesos_S4 := 1;

ELSE

#Motor_gruesos_S4 := 0;

END_IF;

ELSE

#proceso_4 := 0;

#Motor_finos_S4 := 0;

#Motor_gruesos_S4 := 0;

#Marcha_finos_S4 := 0;

#Marcha_gruesos_S4 := 0;

END_IF;

(*PROCESO 5*)

IF #Silo_5 = 1 THEN

#Silo_1 := 0; #Silo_2 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_6 := 0; #Silo_7 := 0; #Silo_8 := 0;
#proceso_5 := 1;

IF #Marcha_finos_S5 = 1 THEN

#Motor_finos_S5 := 1;

ELSE

#Motor_finos_S5 := 0;

END_IF;

IF #Marcha_gruesos_S5 = 1 THEN

#Motor_gruesos_S5 := 1;

ELSE

#Motor_gruesos_S5 := 0;

END_IF;

ELSE

#proceso_5 := 0;

#Motor_finos_S5 := 0;

#Motor_gruesos_S5 := 0;

#Marcha_finos_S5 := 0;

#Marcha_gruesos_S5 := 0;

END_IF;

(*PROCESO 6*)

IF #Silo_6 = 1 THEN

#Silo_1 := 0; #Silo_2 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_7 := 0; #Silo_8 := 0; #proceso_6 :=
1;

IF #Marcha_finos_S6 = 1 THEN

#Motor_finos_S6 := 1;

ELSE

#Motor_finos_S6 := 0;

END_IF;

IF #Marcha_gruesos_S6 = 1 THEN

#Motor_gruesos_S6 := 1;

ELSE

#Motor_gruesos_S6 := 0;

END_IF;

ELSE

#proceso_6 := 0;

#Motor_finos_S6 := 0;

#Motor_gruesos_S6 := 0;

#Marcha_finos_S6 := 0;

#Marcha_gruesos_S6 := 0;

END_IF;

(*PROCESO 7*)

```
IF #Silo_7 = 1 THEN
    #Silo_1 := 0; #Silo_2 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_8 := 0;
#proceso_7 := 1;
```

```
IF #Marcha_finos_S7 = 1 THEN
    #Motor_finos_S7 := 1;
ELSE
    #Motor_finos_S7 := 0;
END_IF;
```

```
IF #Marcha_gruesos_S7 = 1 THEN
    #Motor_gruesos_S7 := 1;
ELSE
    #Motor_gruesos_S7 := 0;
END_IF;
```

```
ELSE
    #proceso_7 := 0;
    #Motor_finos_S7 := 0;
    #Motor_gruesos_S7 := 0;
    #Marcha_finos_S7 := 0;
    #Marcha_gruesos_S7 := 0;
END_IF;
```

(*PROCESO 8*)

```
IF #Silo_8 = 1 THEN
#Silo_1 := 0; #Silo_2 := 0; #Silo_3 := 0; #Silo_4 := 0; #Silo_5 := 0; #Silo_6 := 0; #Silo_7 := 0; #proceso_8 :=
1;
```

```
IF #Marcha_finos_S8 = 1 THEN
    #Motor_finos_S8 := 1;
ELSE
    #Motor_finos_S8 := 0;
END_IF;
```

```
IF #Marcha_gruesos_S8 = 1 THEN
    #Motor_gruesos_S8 := 1;
ELSE
    #Motor_gruesos_S8 := 0;
END_IF;
```

```
ELSE
    #proceso_8 := 0;
    #Motor_finos_S8 := 0;
    #Motor_gruesos_S8 := 0;
    #Marcha_finos_S8 := 0;
    #Marcha_gruesos_S8 := 0;
END_IF;
```

Esta función es complementaria a la función “Esc_señal_analog” por lo que se debe de informar a la báscula qué material está pesando. Se ha asociado el proceso 1 al material 1, el proceso 2 al material dos y sucesivos. Esto se consigue mediante:

(*información para la báscula*)

```
IF #proceso_1 THEN
  #Material_1 := 1;
ELSE
  #Material_1 := 0;
END_IF;
IF #proceso_2 THEN
  #Material_2 := 1;
ELSE
  #Material_2 := 0;
END_IF;
IF #proceso_3 THEN
  #Material_3 := 1;
ELSE
  #Material_3 := 0;
END_IF;
IF #proceso_4 THEN
  #Material_4 := 1;
ELSE
  #Material_4 := 0;
END_IF;
IF #proceso_5 THEN
  #Material_5 := 1;
ELSE
  #Material_5 := 0;
END_IF;
IF #proceso_6 THEN
  #Material_6 := 1;
ELSE
  #Material_6 := 0;
END_IF;
IF #proceso_7 THEN
  #Material_7 := 1;
ELSE
  #Material_7 := 0;
END_IF;
IF #proceso_8 THEN
  #Material_8 := 1;
ELSE
  #Material_8 := 0;
END_IF;
```

II.V FUNCIÓN "CONTROL_HISTÉRESIS"

Lo que se pretende con esta función es mantener una temperatura media tan cercana como sea posible a la consigna.

Las variables que se necesitan para este tipo de control son:

Control_histéresis							
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ...
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Consigna	Real	0.0	No rem...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Banda_control	Real	0.0	Remanente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Sobretemp_alarma	Real	0.0	Remanente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Temperatura_actual	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Calefacción_ON	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Alarma	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<Agregar>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12	Banda_superior	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Banda_inferior	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Temp_alarma	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 44 Variables declaradas en la función "Control_histéresis"

El programa se muestra a continuación:

```
#Banda_superior := #Consigna + (#Banda_control / 100*#Consigna);
#Banda_inferior := #Consigna - (#Banda_control / 100*#Consigna);
#Temp_alarma := (#Sobretemp_alarma / 100* #Consigna) + #Consigna;
```

Primero se calculan las bandas superior e inferior y la temperatura a partir de la cual se busca que el sistema dé un aviso de alarma.

Una vez definidas las bandas entre las que se moverá el control, se dará la orden de conexión o desconexión, en función de la zona de la banda en la que se encuentre en ese momento:

```
IF #Temperatura_actual < #Banda_superior AND #Temperatura_actual < #Banda_inferior THEN
    #Calefacción_ON := 1;
END_IF;
IF #Temperatura_actual >= #Banda_superior THEN
    #Calefacción_ON := 0;
END_IF;
IF #Temperatura_actual = #Banda_inferior THEN
    #Calefacción_ON := 1;
END_IF;
```



```

IF #Temperatura_actual >= #Temp_alarma THEN
    #Alarma := 1;
ELSE
    #Alarma := 0;
END_IF;
    
```

Al realizar la llamada de la función en el MAIN aparecerá un bloque como el mostrado en la figura 45:

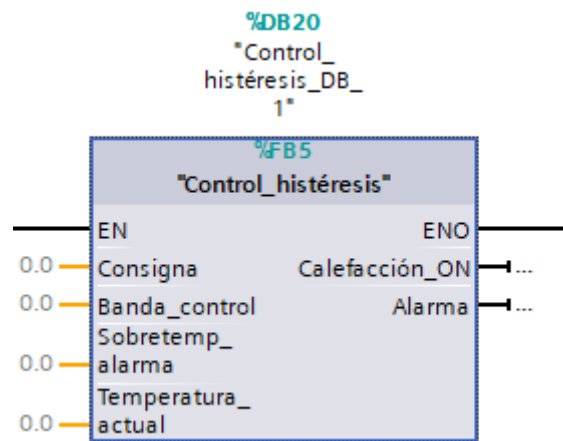


Figura 45 Vista de la función "Control_histéresis" (bis)

II.VI FUNCIÓN “ACTIVACIÓN_ACTUADORES”

A continuación se muestran todos los actuadores llamados a través de distintas memorias auxiliares que se han empleado en la función “Activación_actuadores”:

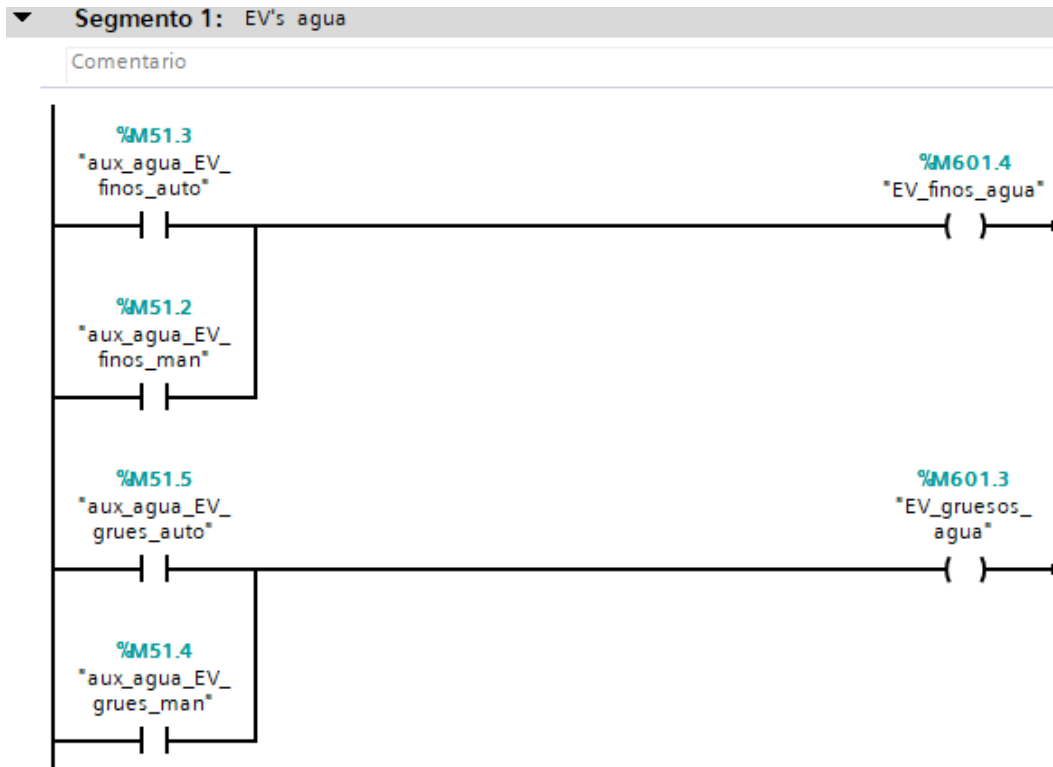


Figura 46 Programación segmento 1 “Activación_actuadores”

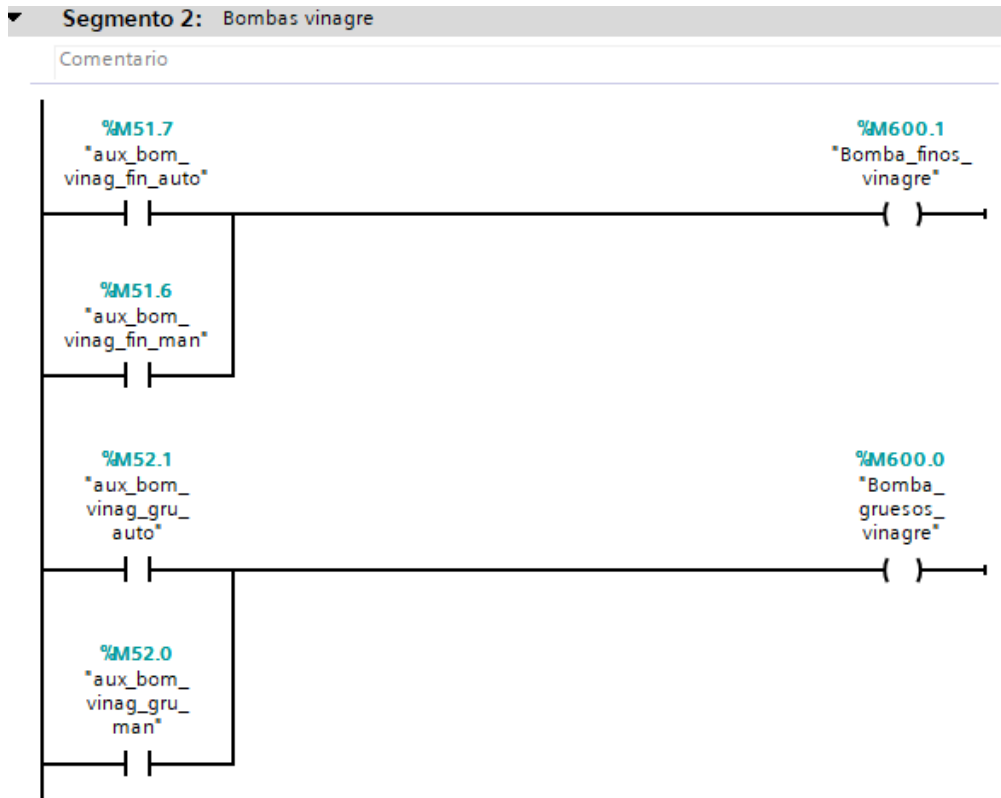


Figura 47 Programación segmento 2 “Activación_actuadores”

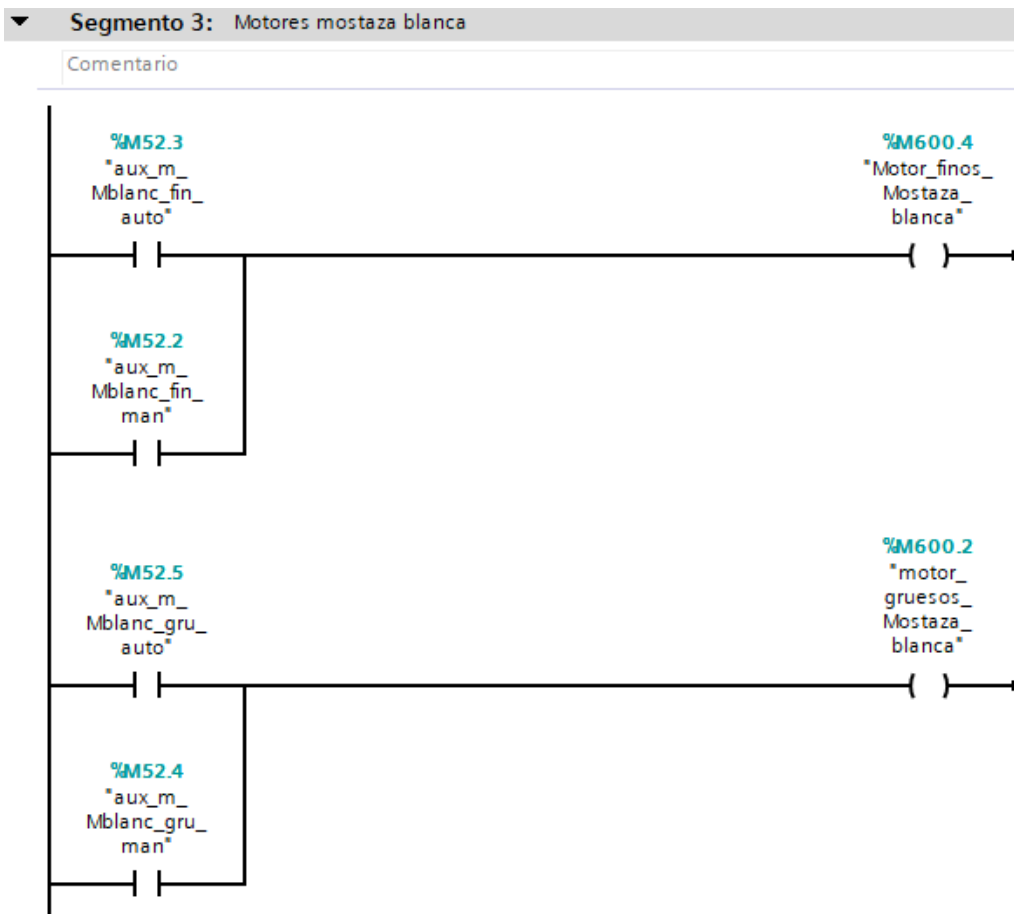


Figura 48 Programación segmento 3 “Activación_actuadores”

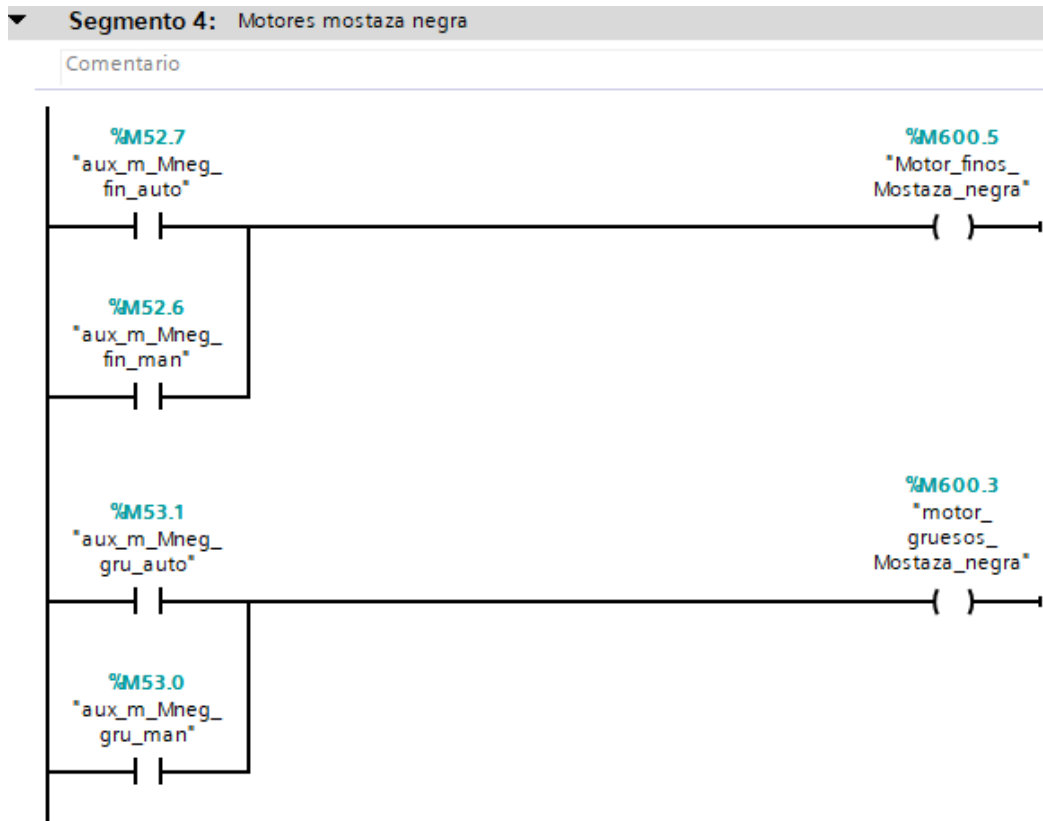


Figura 49 Programación segmento 4 “Activación_actuadores”

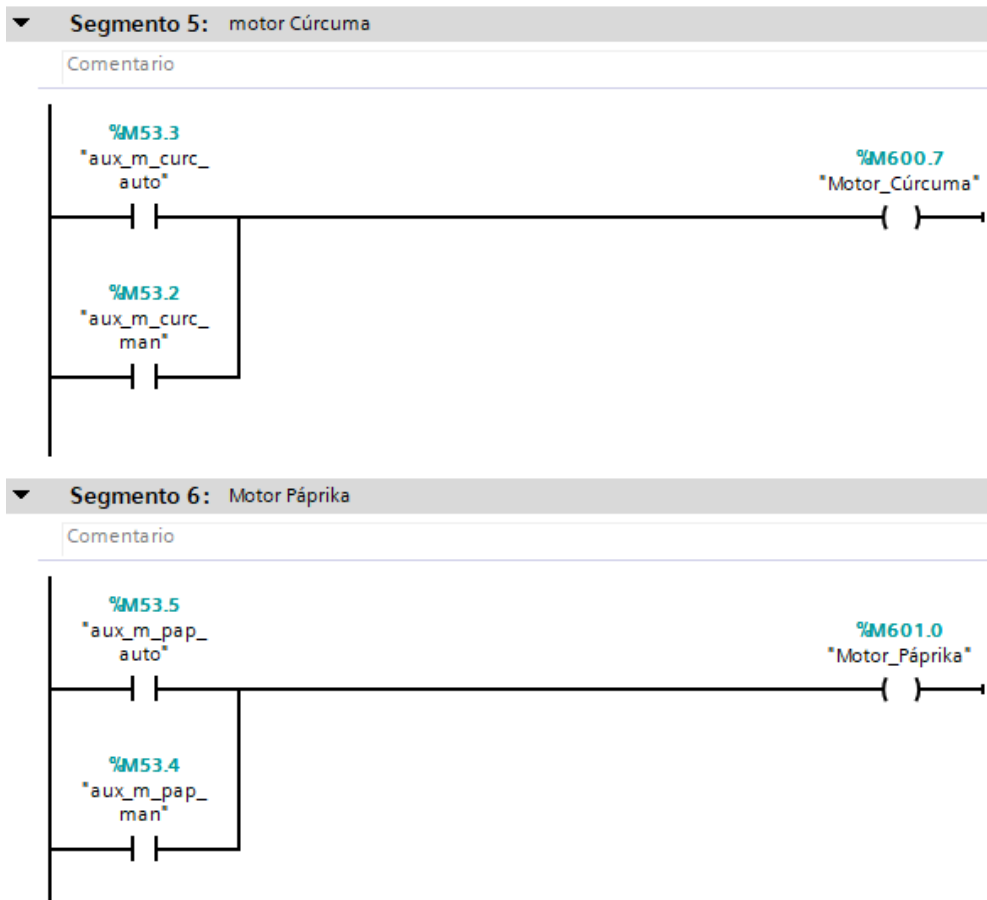


Figura 50 Programación segmentos 5 y 6 “Activación_actuadores”

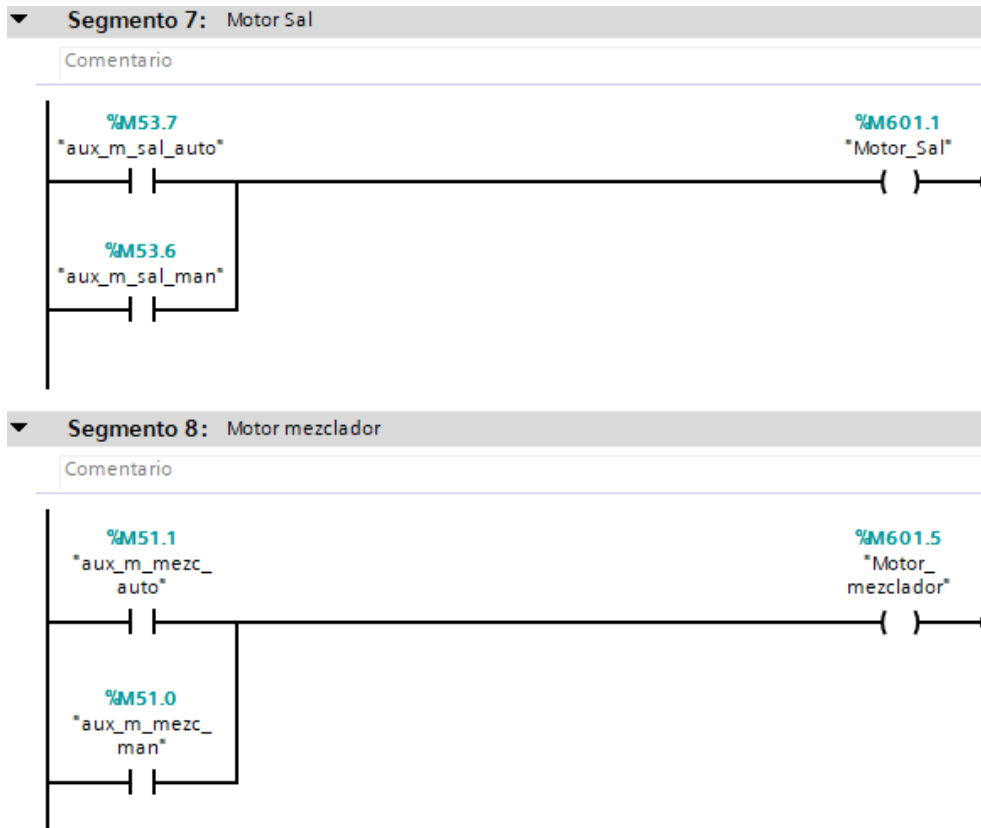


Figura 51 Programación segmentos 7 y 8 "Activación_actuadores"

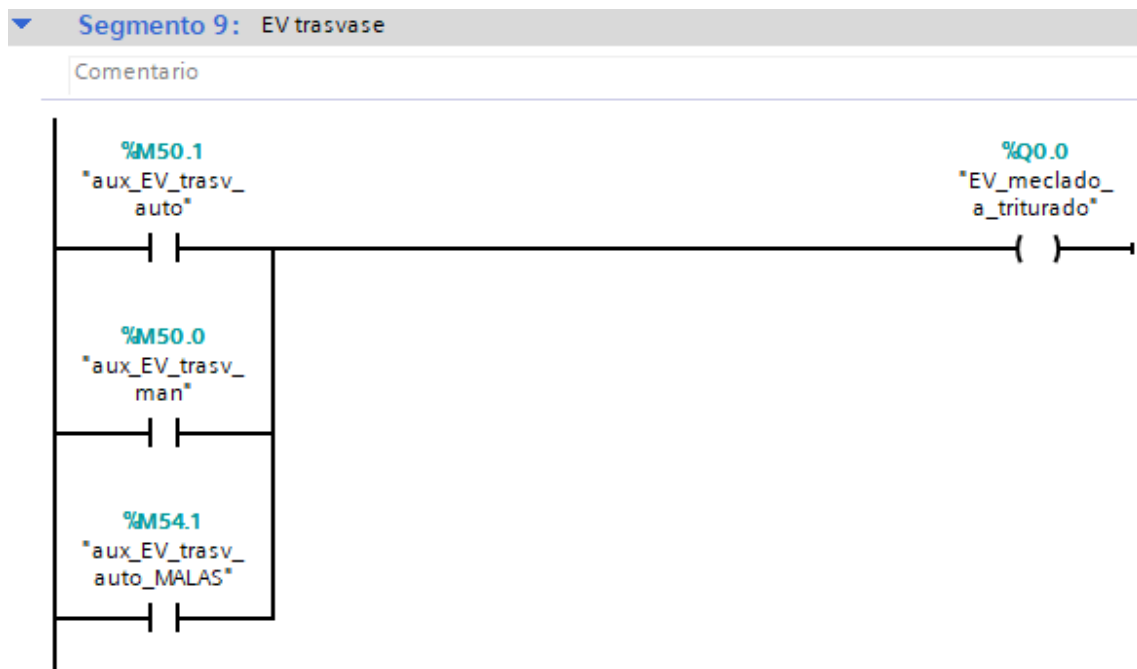


Figura 52 Programación segmento 9 "Activación_actuadores"

▼ **Segmento 10:** bomba trasvase

Comentario



Figura 53 Programación segmento 10 "Activación_actuadores"

▼ **Segmento 11:** Motor trituradora

Comentario



▼ **Segmento 12:** Resistencias calefactoras

Comentario



Figura 54 Programación segmentos 11 y 12 "Activación_actuadores"

II.VII BLOQUE ORGANIZACIÓN PRINCIPAL MAIN [OB1]

La función MAIN se ha empleado para llamar al resto de funciones:

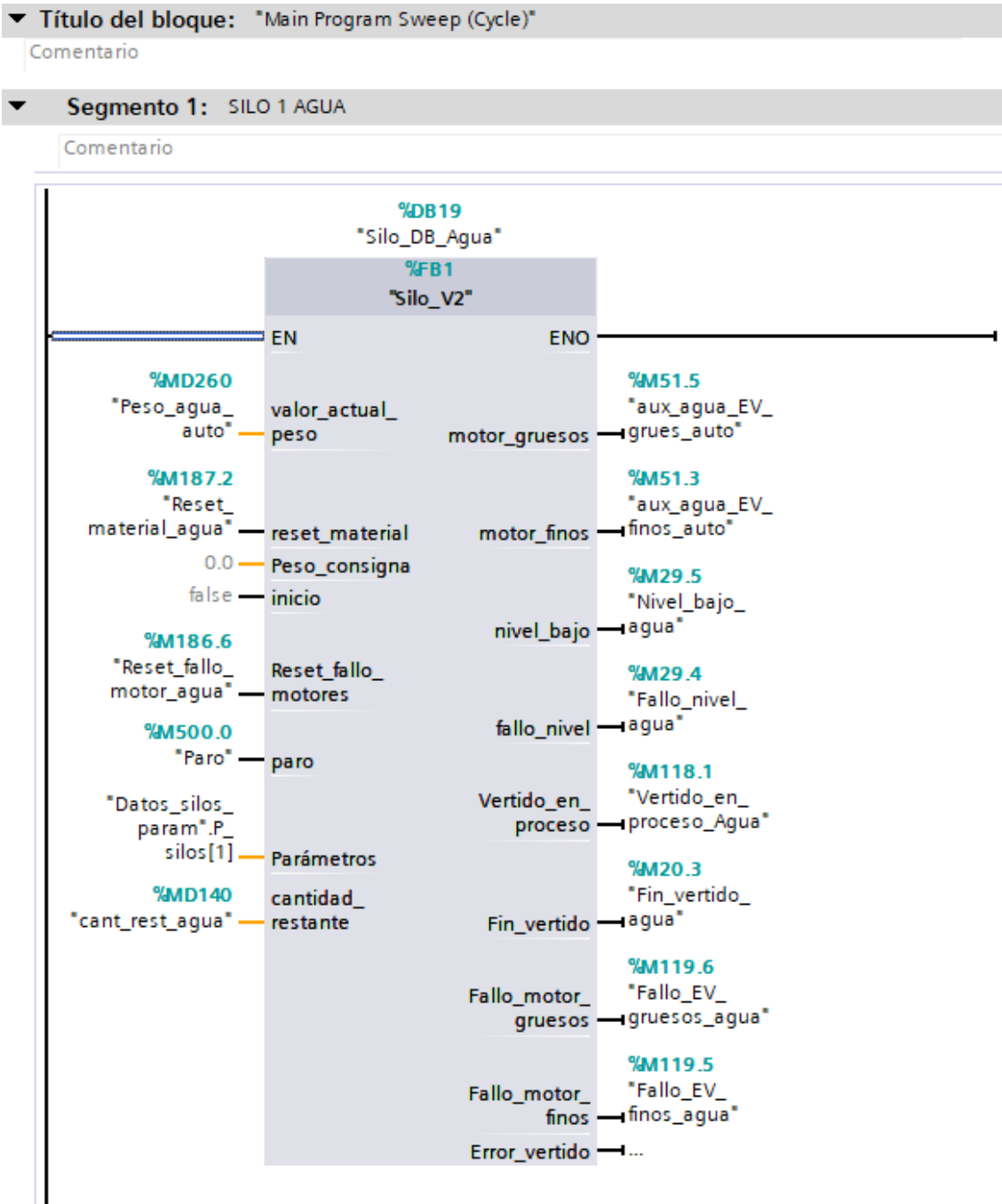


Figura 55 Programación segmento 1 MAIN [OB1]

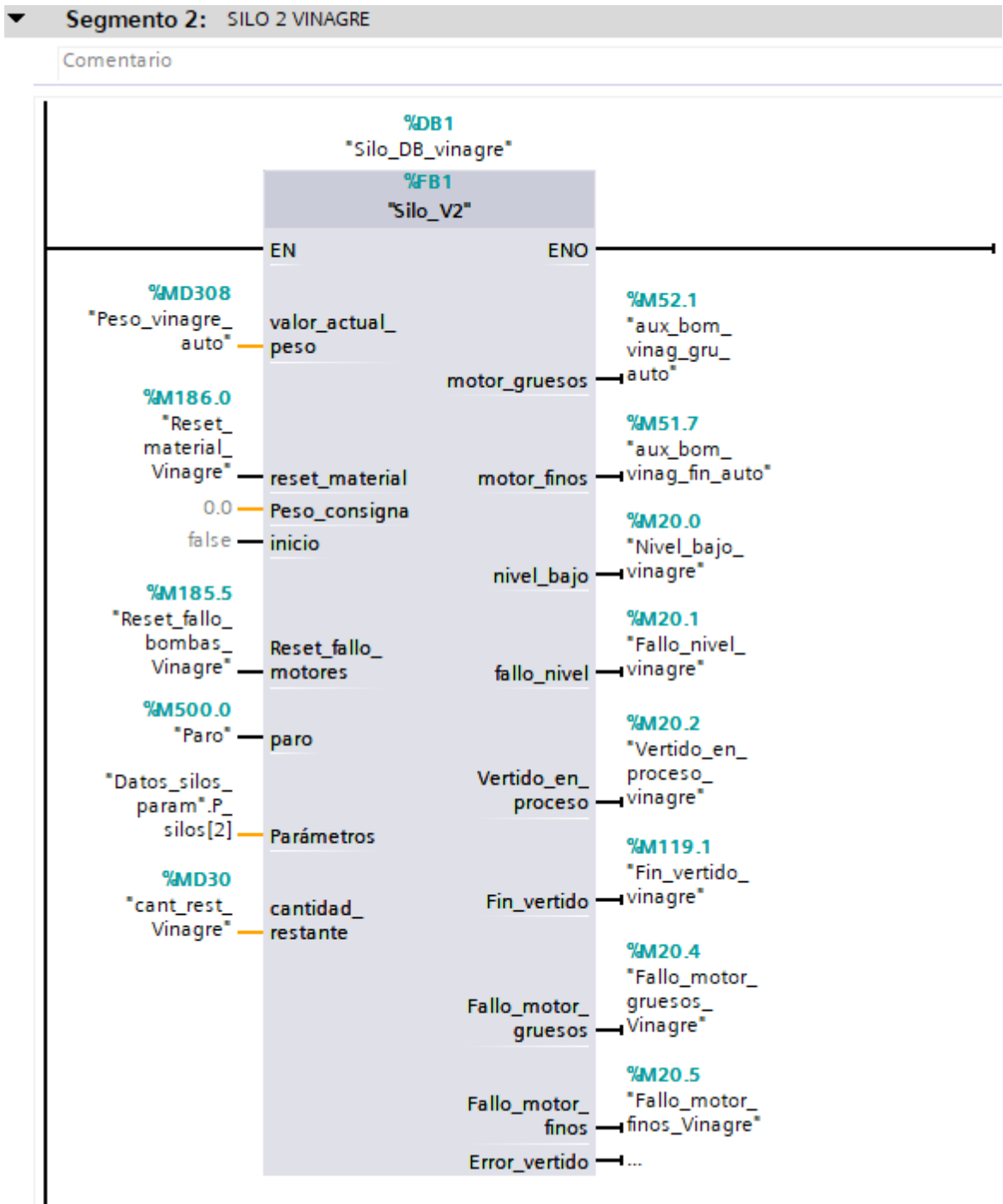


Figura 56 Programación segmento 2 MAIN [OB1]

Segmento 3: SILO 3 MOSTAZA BLANCA

Comentario

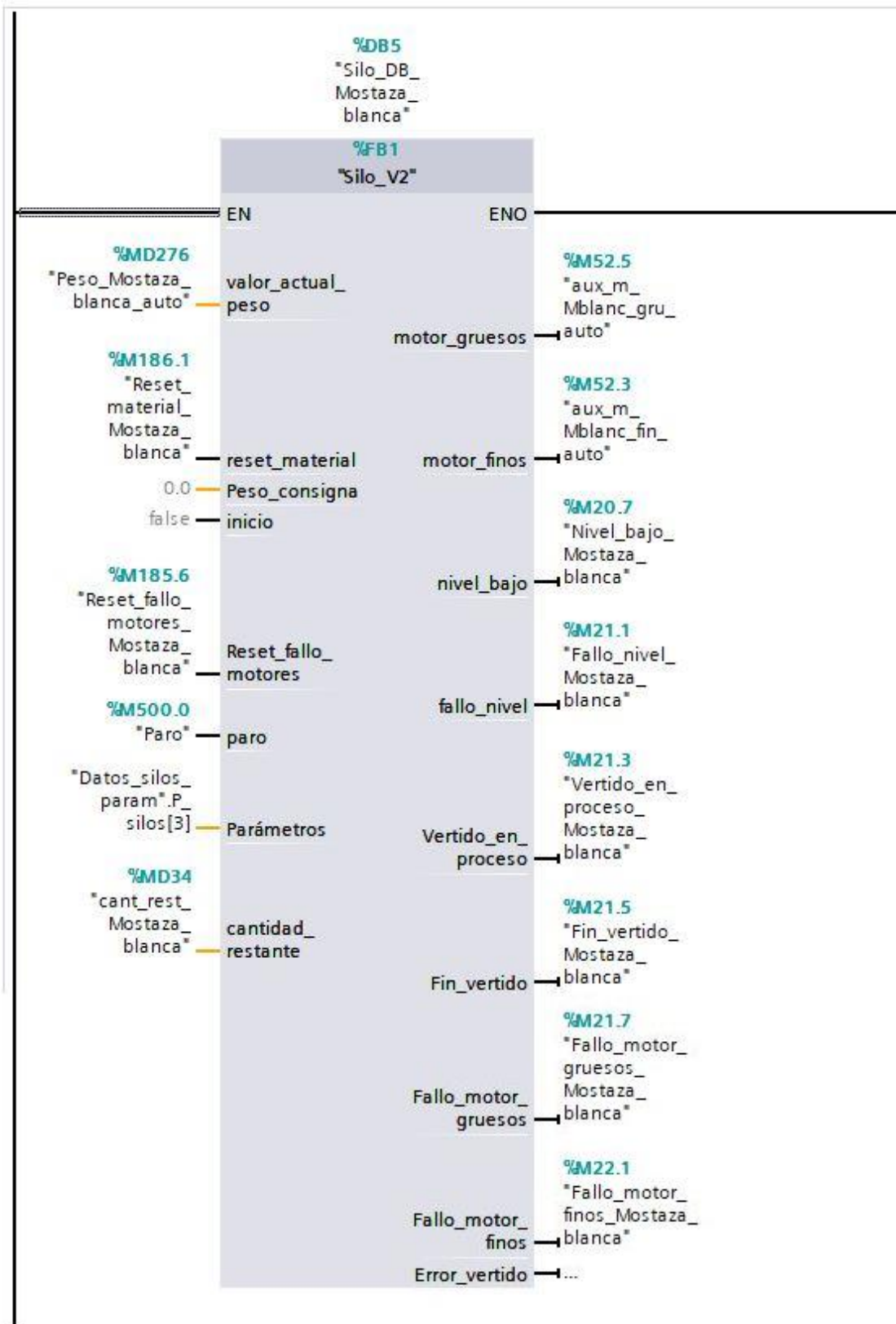


Figura 57 Programación segmento 3 MAIN [OB1]

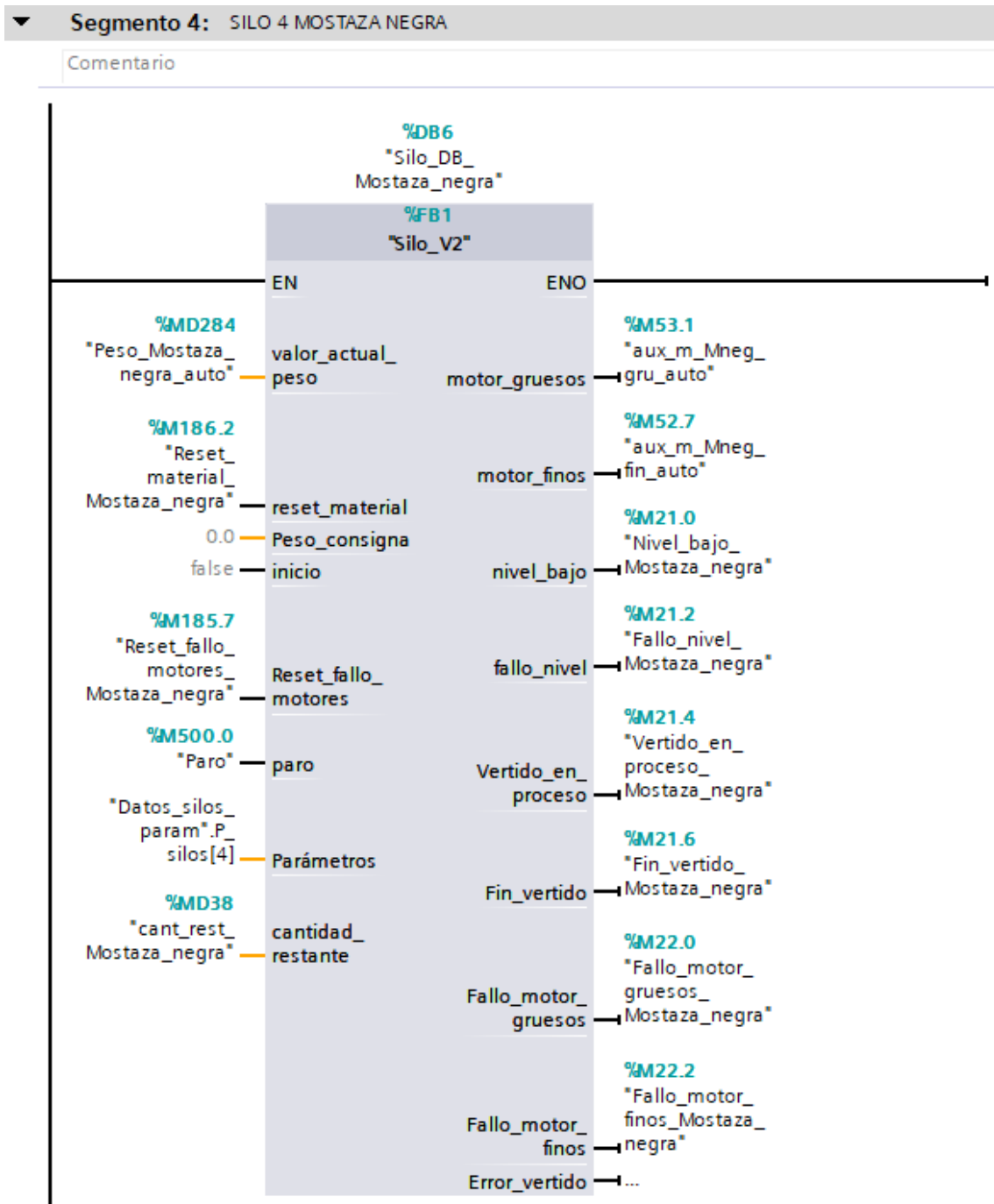


Figura 58 Programación segmento 4 MAIN [OB1]

Como la función está diseñada para activar y leer dos actuadores, a continuación se realiza un reajuste automático para los silos de menor tamaño, que únicamente disponen de un actuador para el vertido de material. De este modo, el usuario que maneje el HMI no tendrá que preocuparse de realizar los cambios necesarios si se modificara alguna vez la planta, y se consigue que no salten los fallos de valor incorrecto en #M y #N.

Se ha sumado al parámetro M y “tiempo_grosos” un valor, paracumplir con las condiciones de una correcta parametrización que se describieron en el punto “Función SILO”.

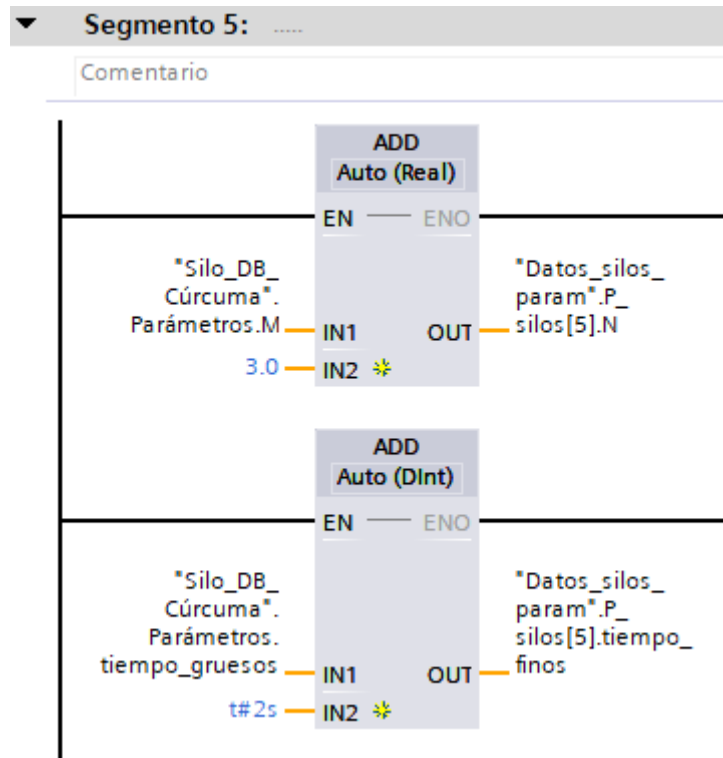


Figura 59 Detalle programación segmento 5 MAIN [OB1]

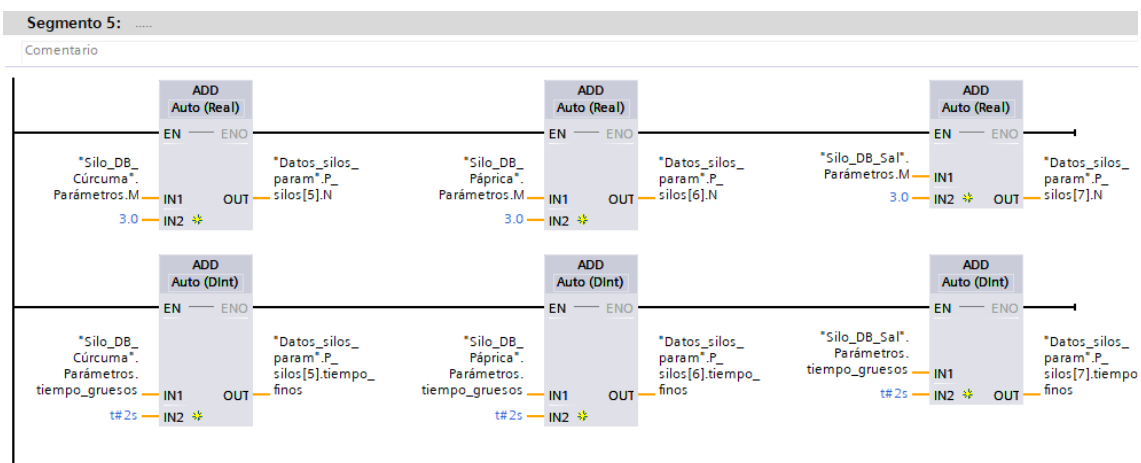


Figura 60 Programación segmento 5 MAIN [OB1]

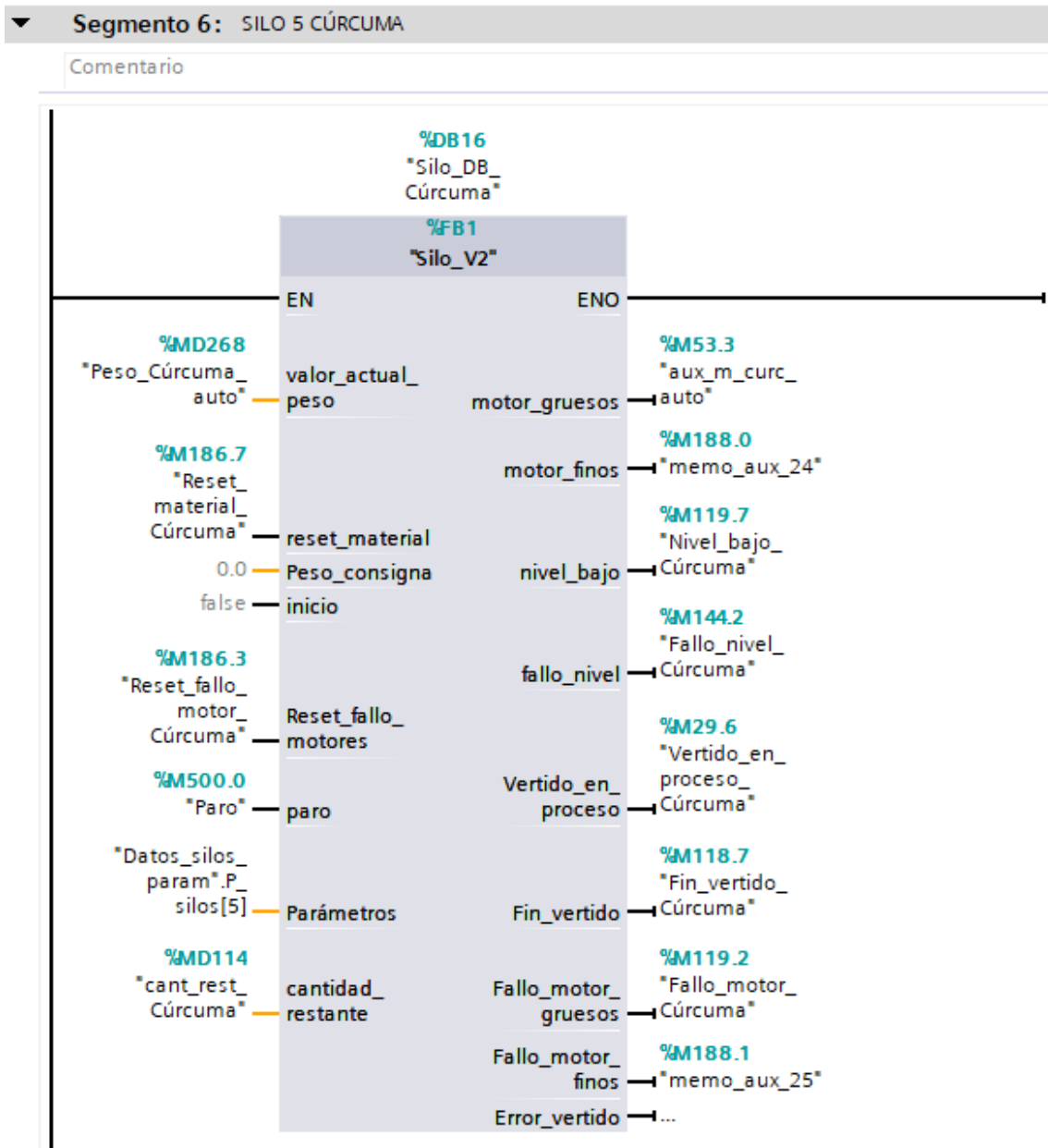


Figura 61 Programación segmento 6 MAIN [OB1]

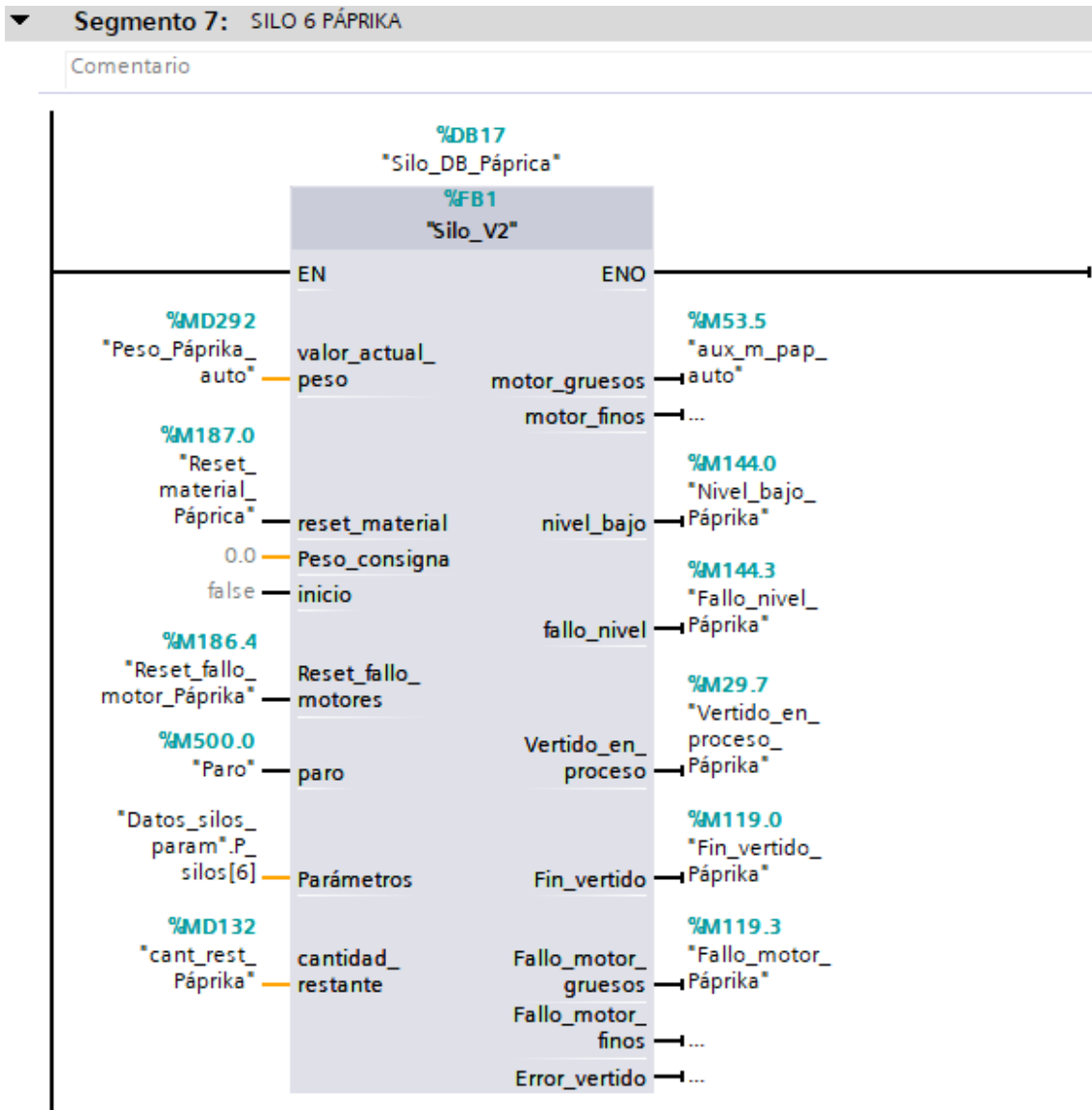


Figura 62 Programación segmento 7 MAIN [OB1]

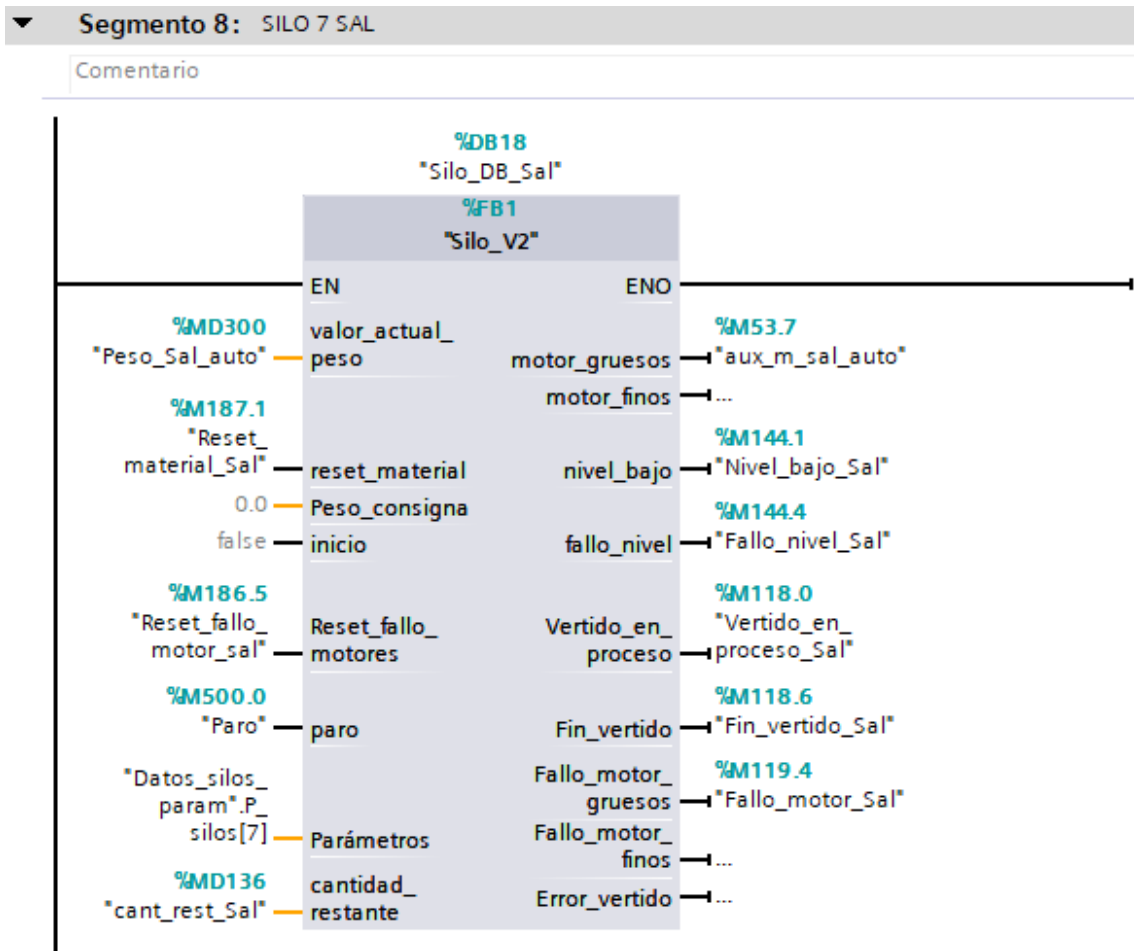


Figura 63 Programación segmento 8 MAIN [OB1]

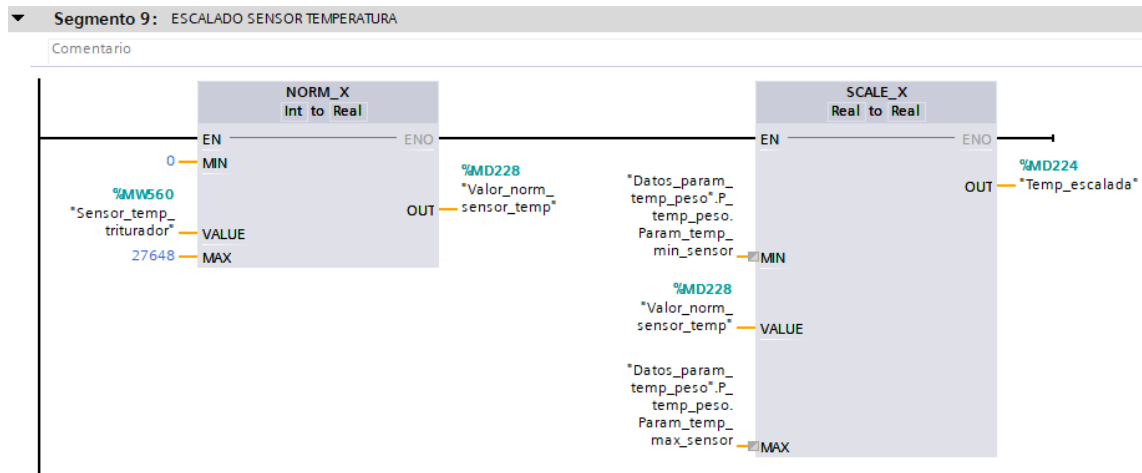


Figura 64 Programación segmento 9 MAIN [OB1]

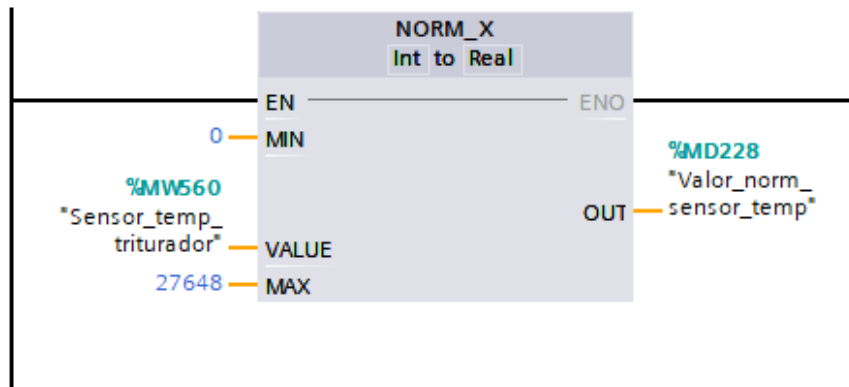


Figura 65 Ampliación detalle 1 programación segmento 1 MAIN [OB1]

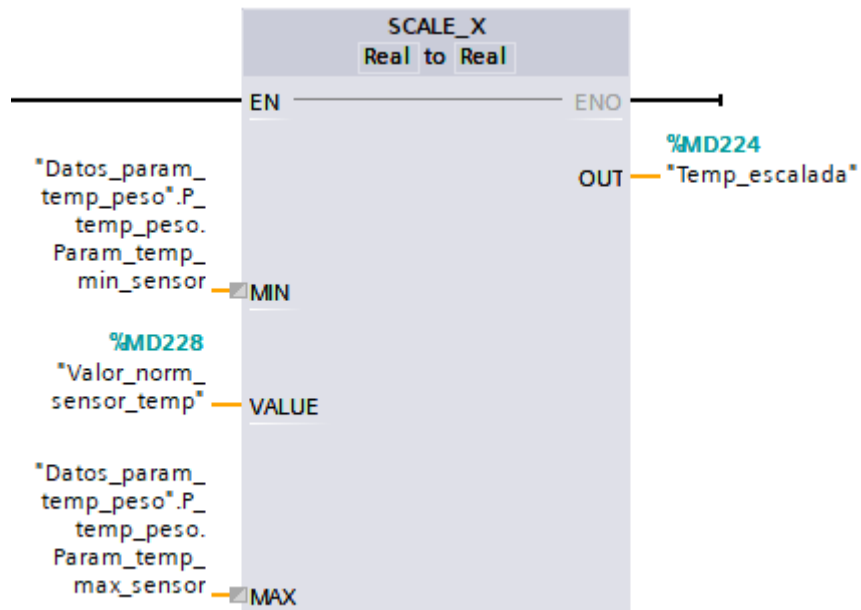


Figura 66 Ampliación detalle 2 programación segmento 1 MAIN [OB1]

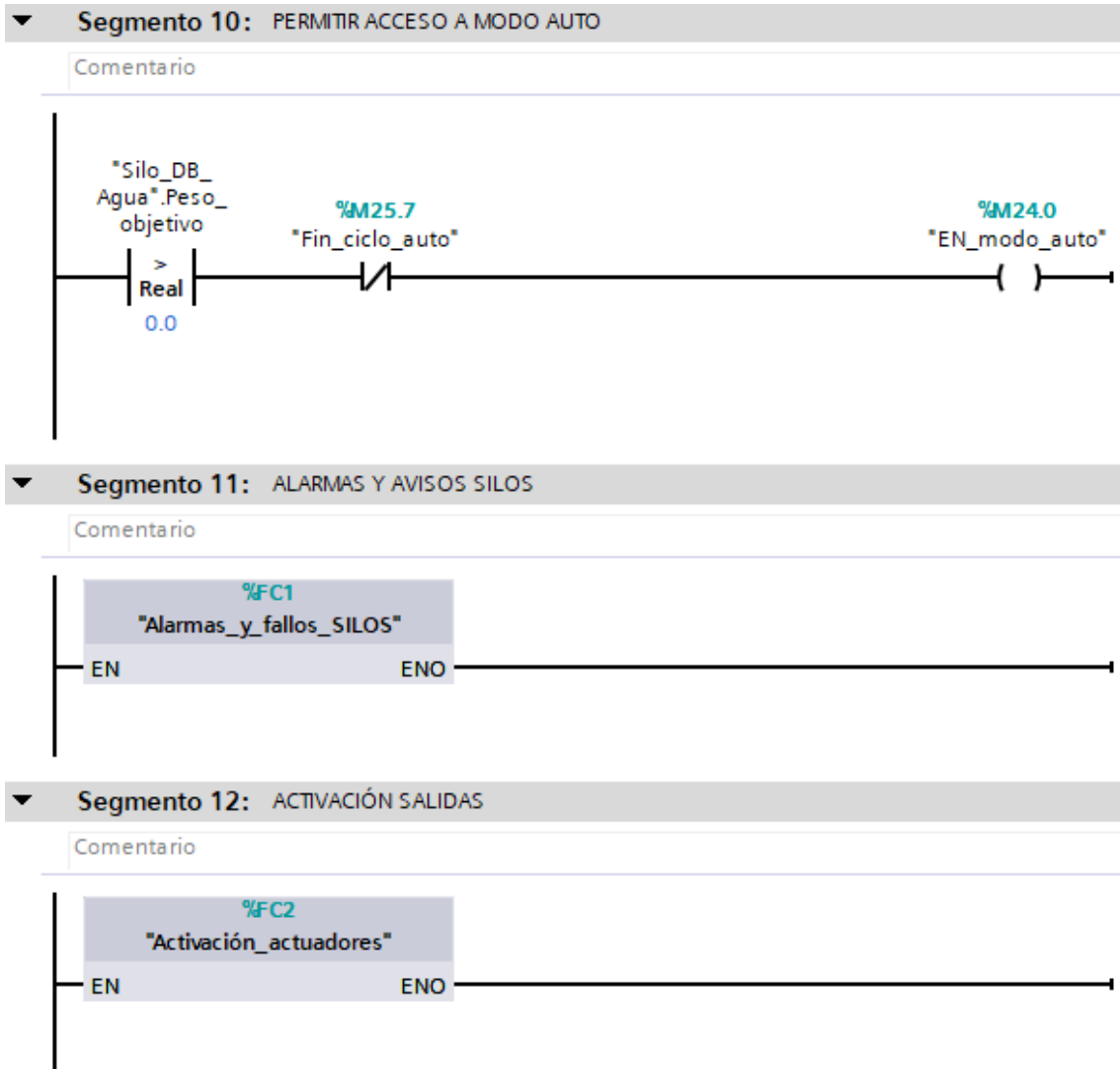


Figura 67 Programación segmento 10, 11 y 12 MAIN [OB1]

II. VIII) BLOQUE ORGANIZACIÓN "MODO_AUTOMAT" [OB124]

En este apartado se muestra el código empleado en el OB 124 para el control del modo automático:

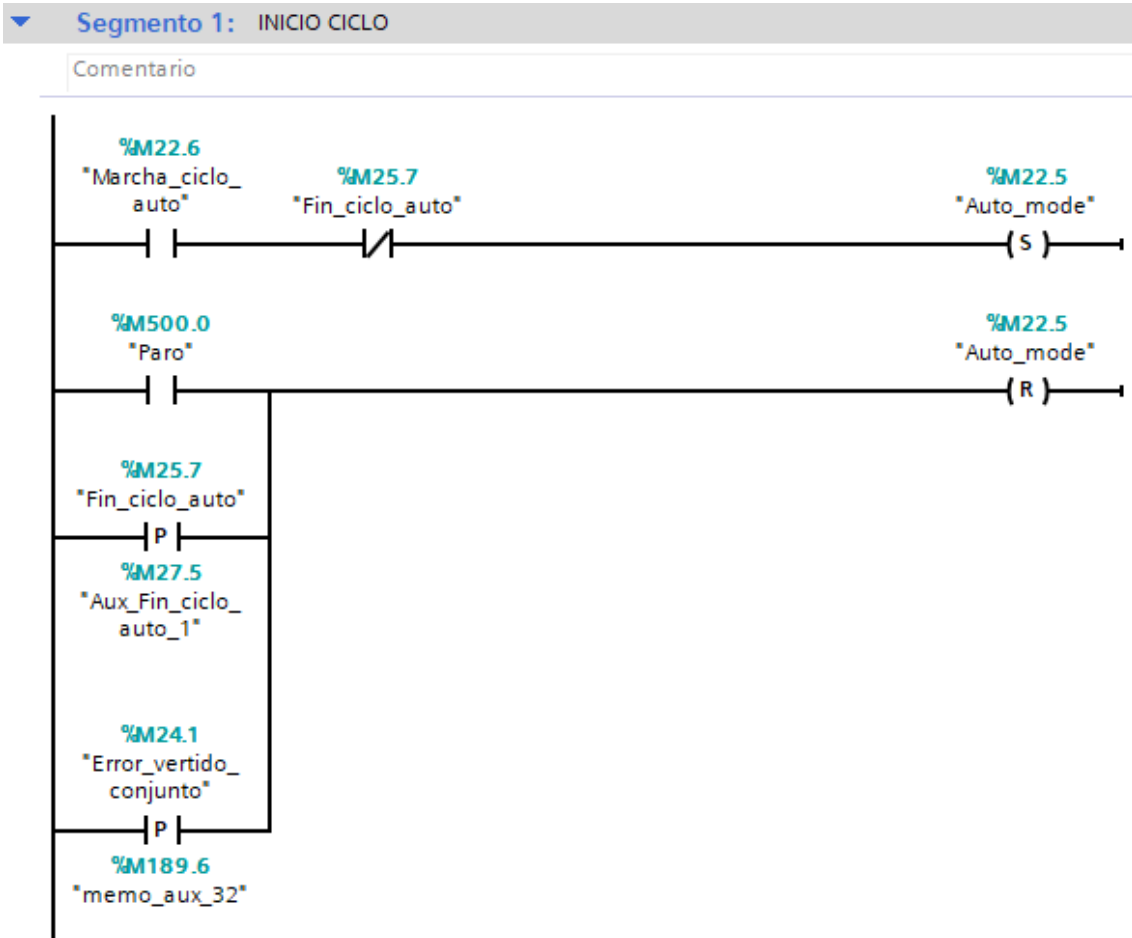


Figura 68 Programación segmento 1 "Modo_automat" [OB124]

▼ Segmento 2: VERTIDO ESCALONADO DE INGREDIENTES

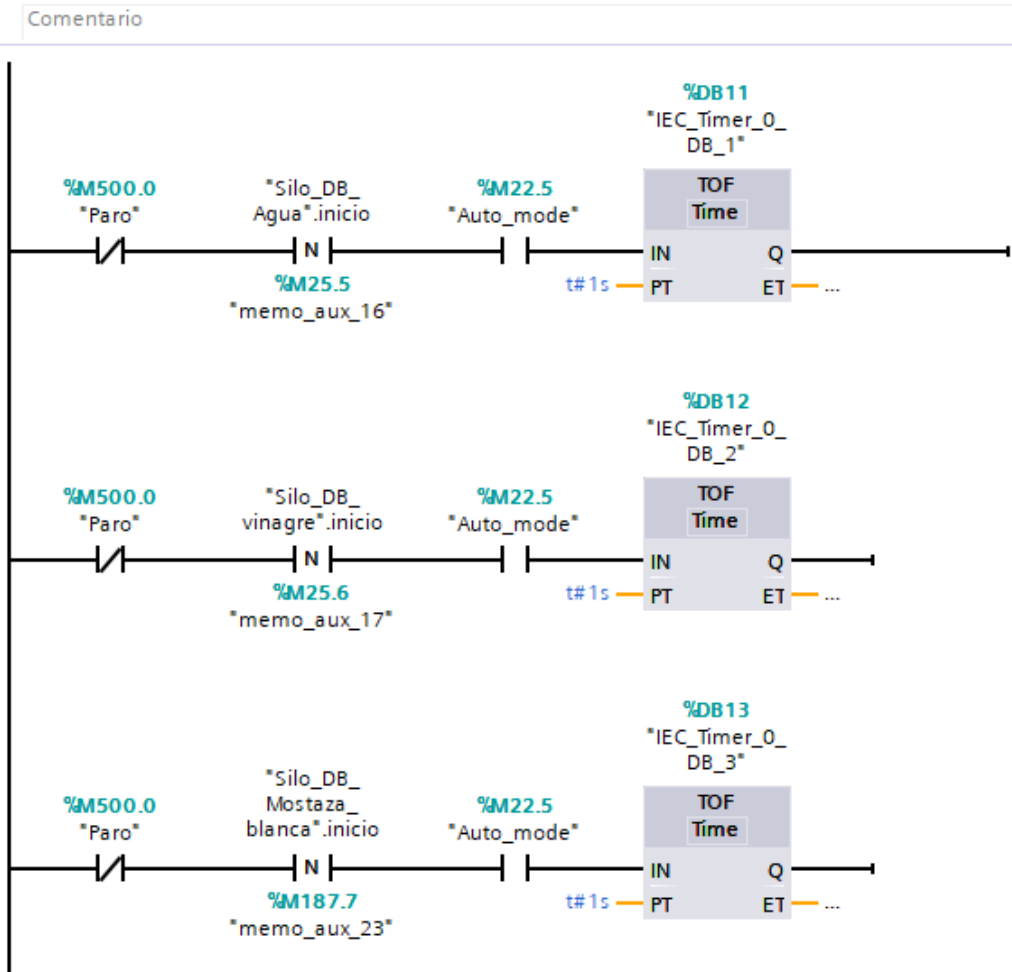


Figura 69 Programación segmento 2.a "Modo_automat" [OB124]

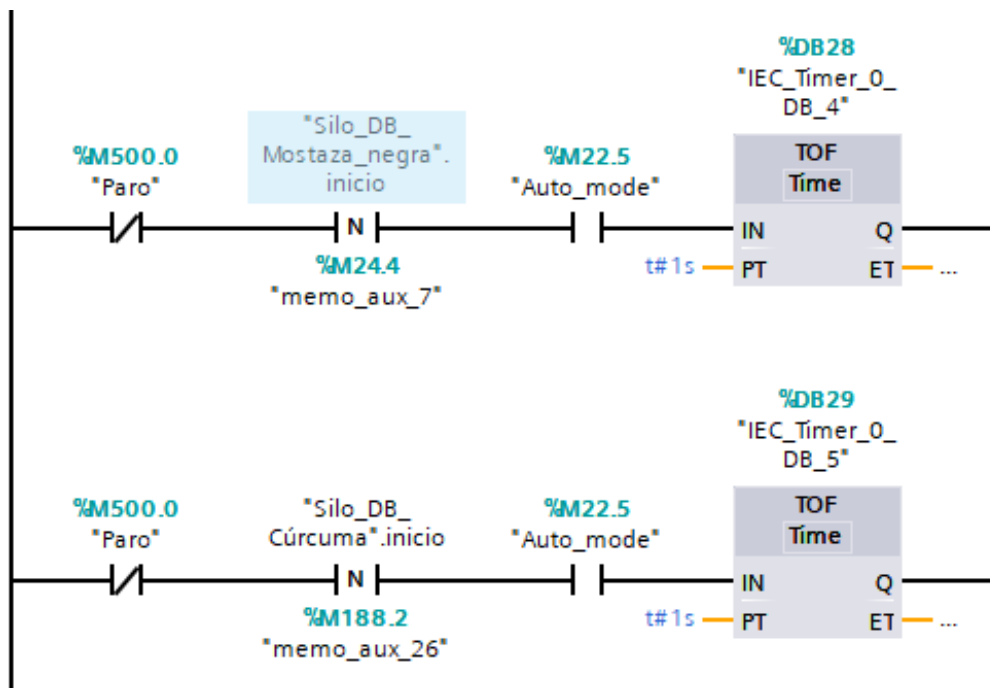


Figura 70 Programación segmento 2.b "Modo_automat" [OB124]

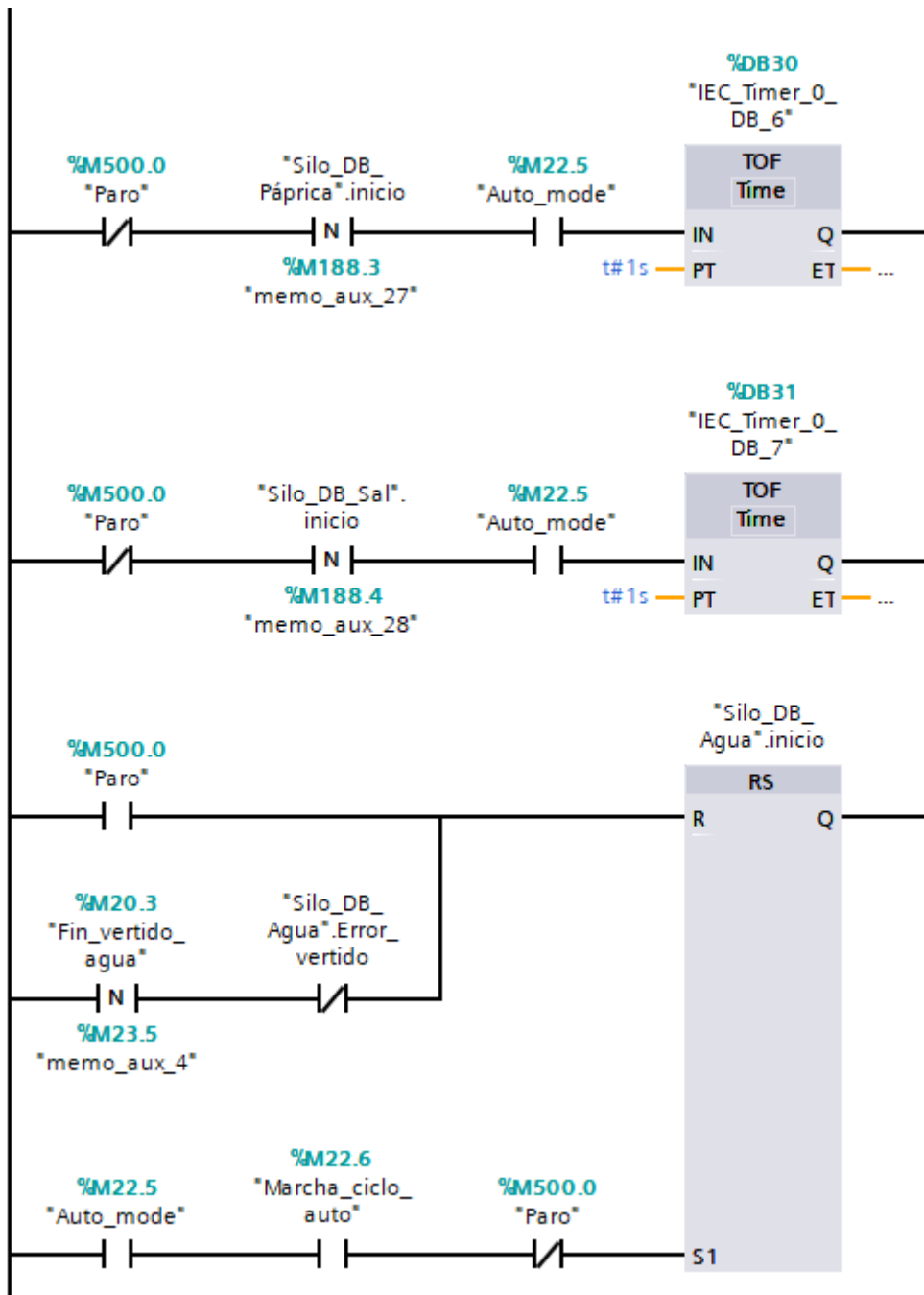


Figura 71 Programación segmento 2.c "Modo_automat" [OB124]

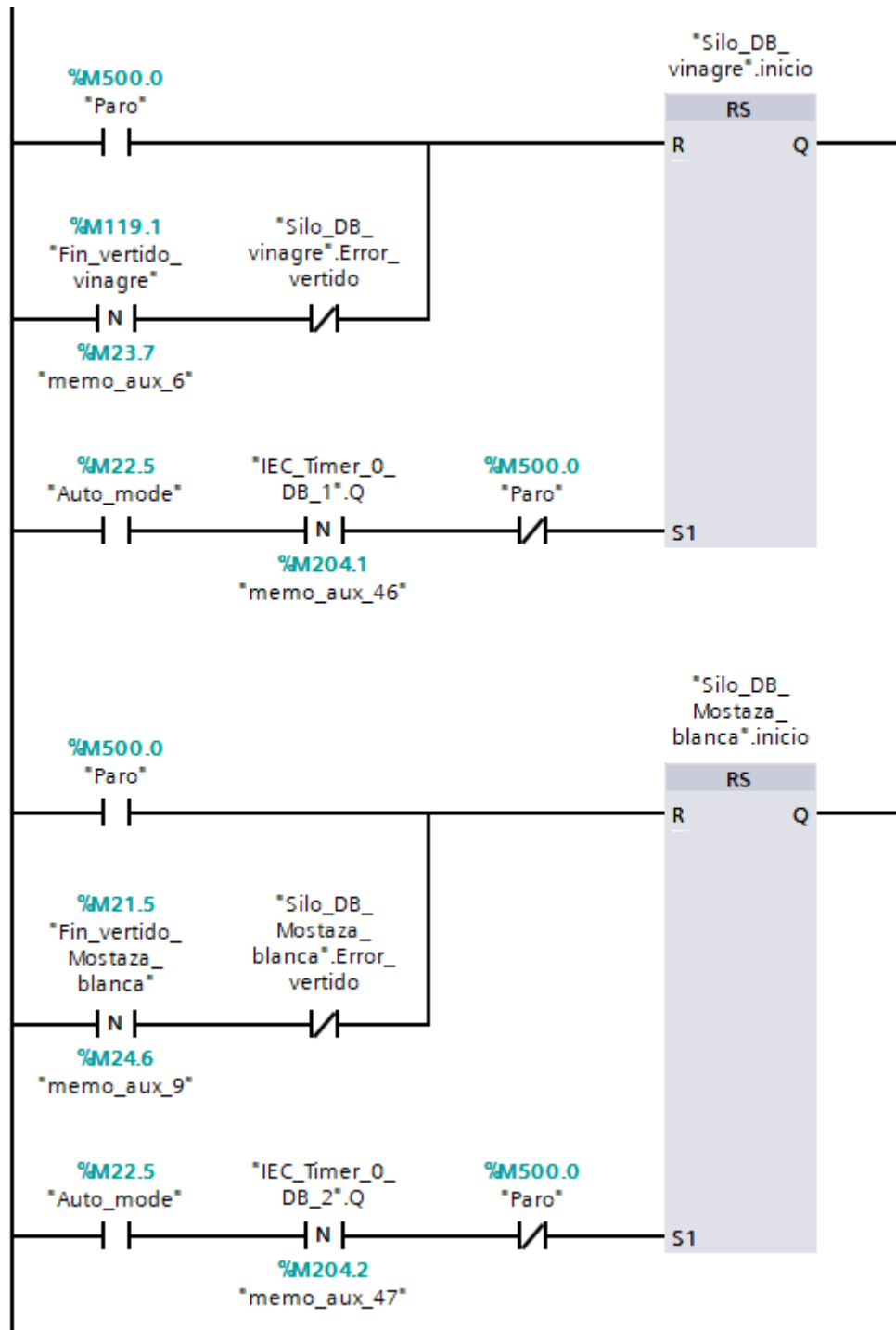


Figura 72 Programación segmento 2.d "Modo_automat" [OB124]

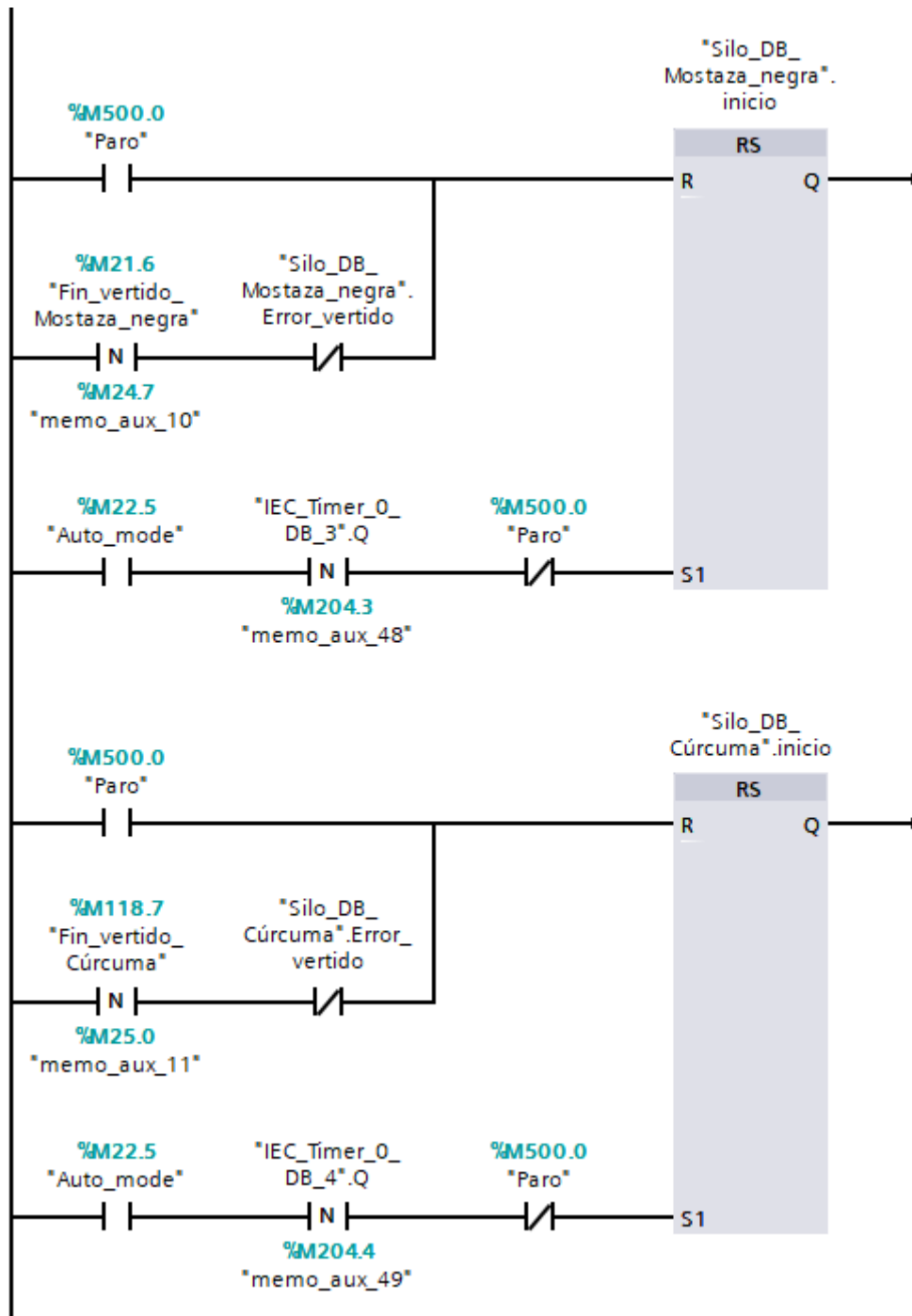


Figura 73 Programación segmento 2.e "Modo_automat" [OB124]

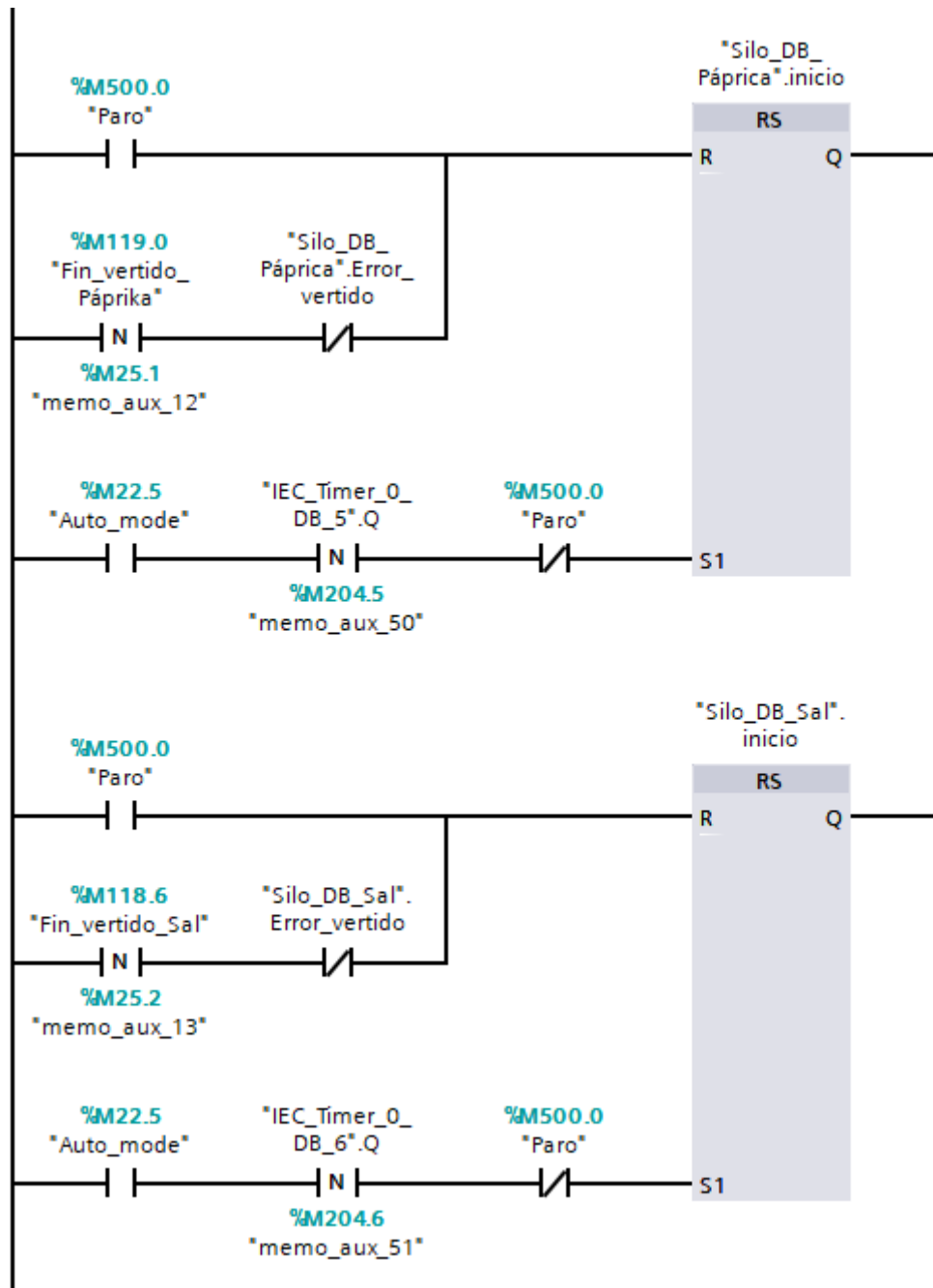


Figura 74 Programación segmento 2.f "Modo_automat" [OB124]

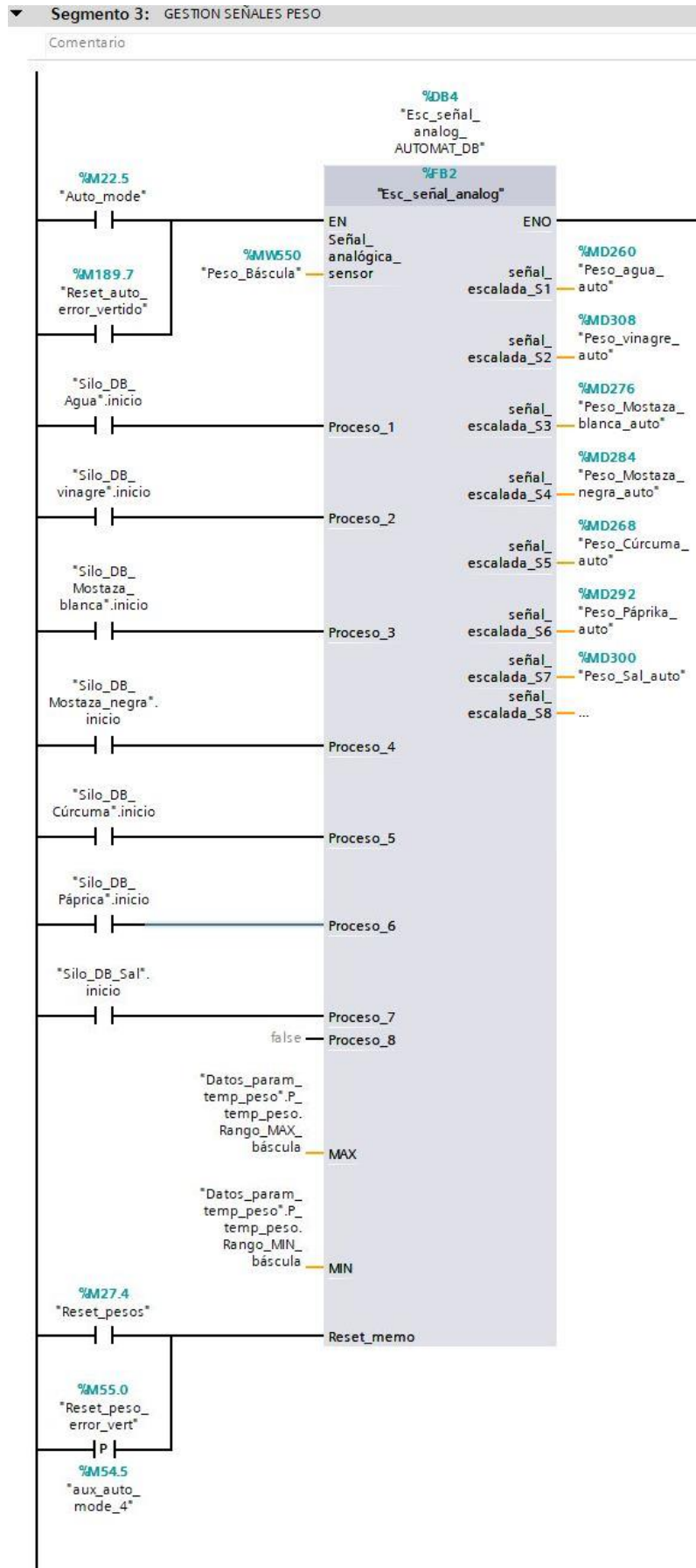


Figura 75 Programación segmento 3 "Modo_automat" [OB124]

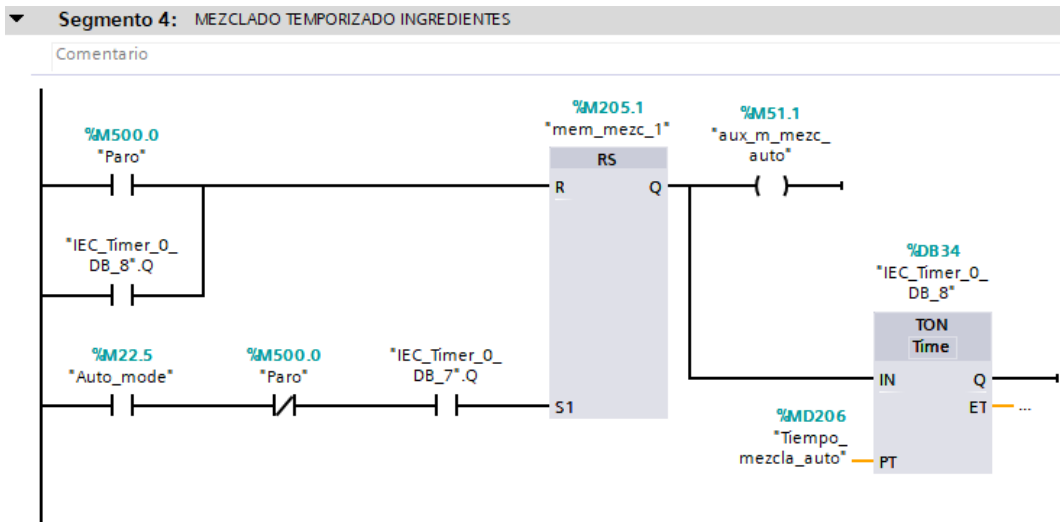


Figura 76 Programación segmento 4 "Modo_automat" [OB124]

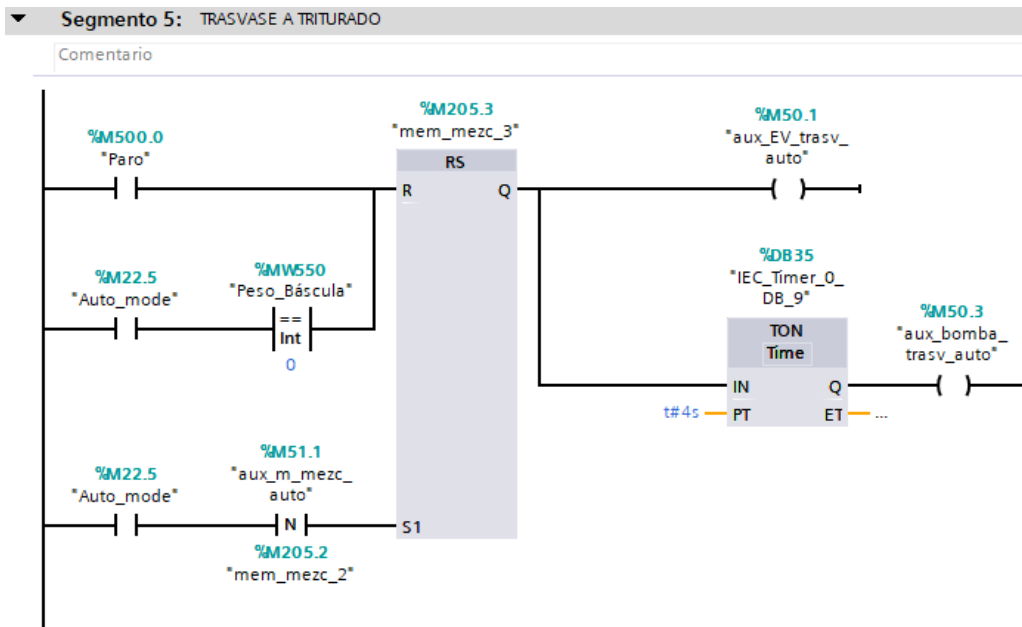


Figura 77 Programación segmento 5 "Modo_automat" [OB124]

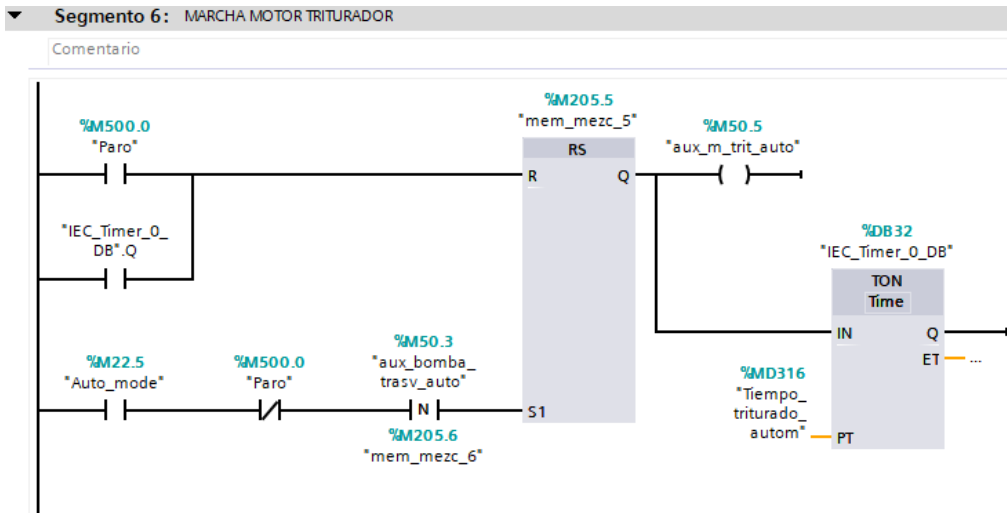


Figura 78 Programación segmento 6 "Modo_automat" [OB124]

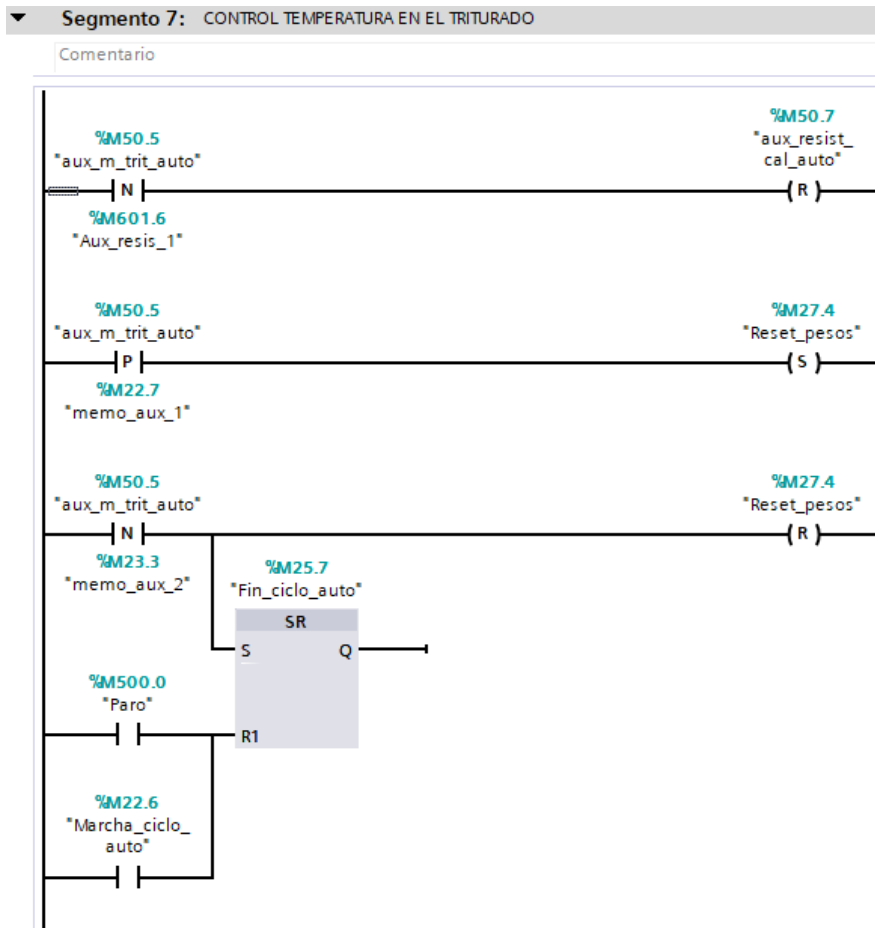


Figura 79 Programación segmento 7.a "Modo_automat" [OB124]

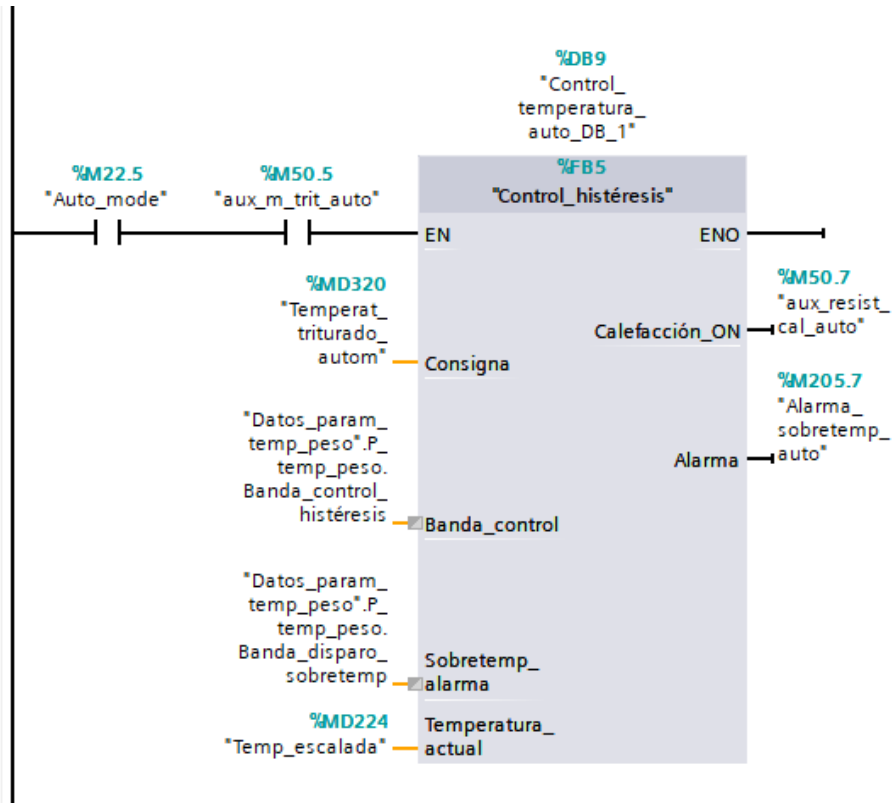


Figura 80 Programación segmento 7.b "Modo_automat" [OB124]

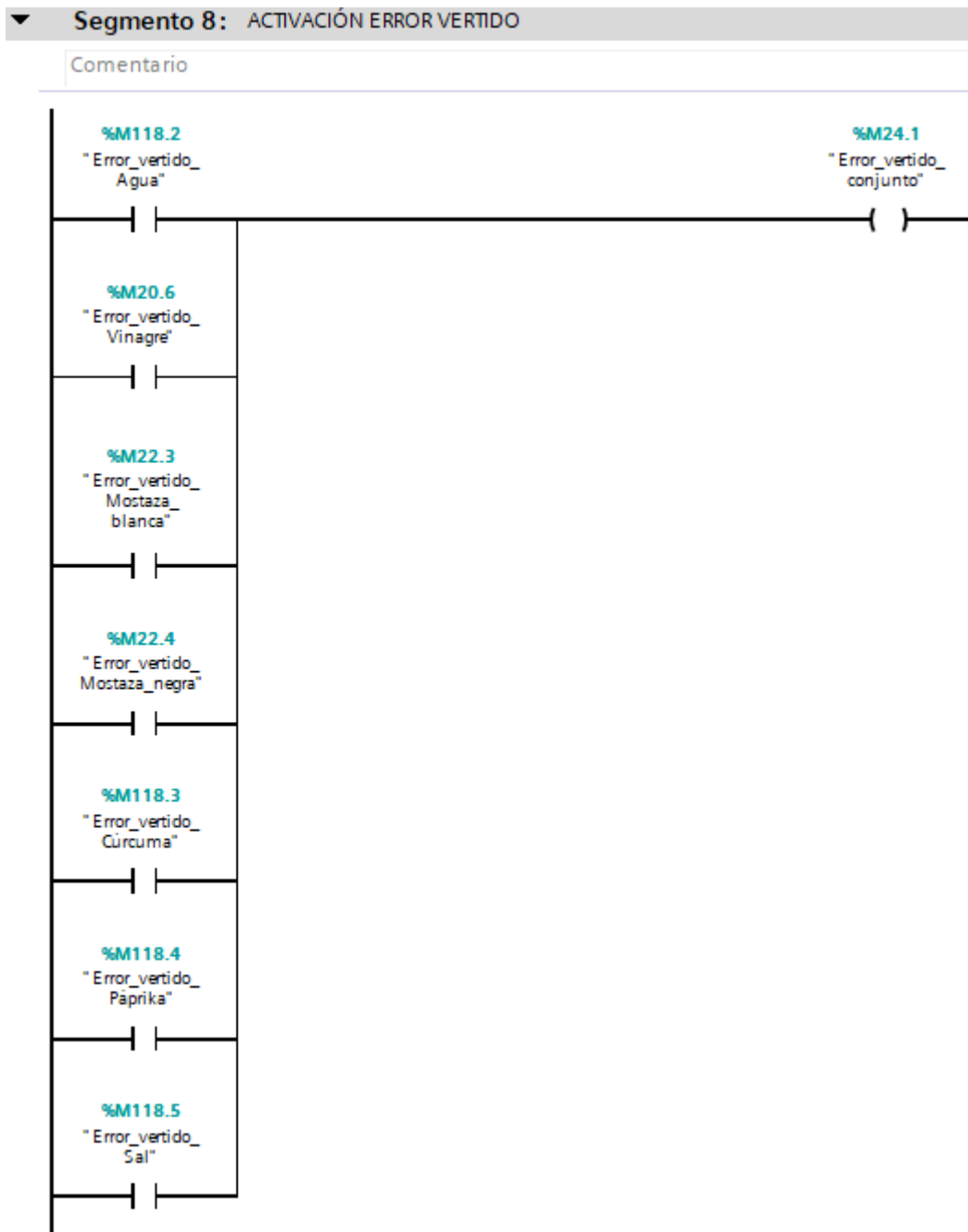


Figura 81 Programación segmento 8 "Modo_automat" [OB124]

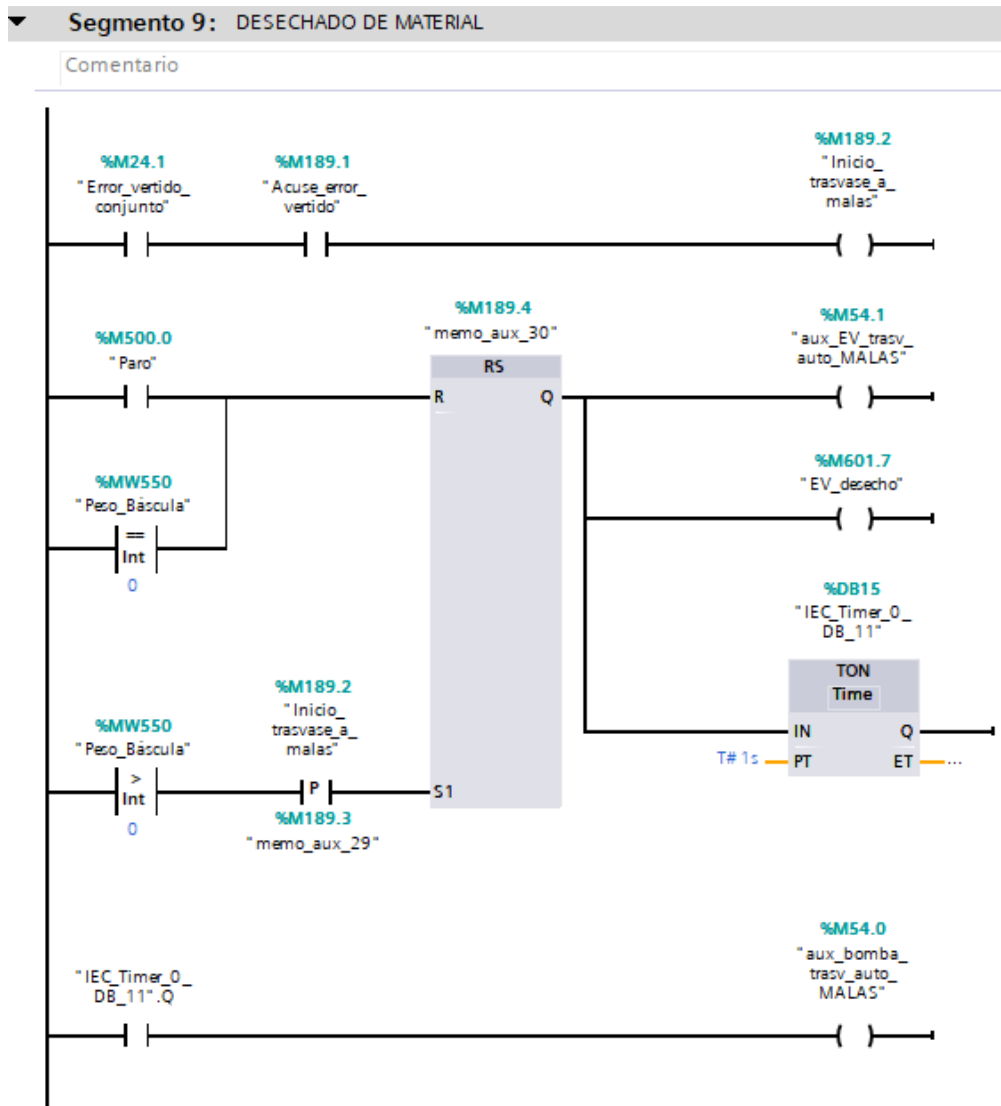


Figura 82 Programación segmento 9 "Modo_automat" [OB124]

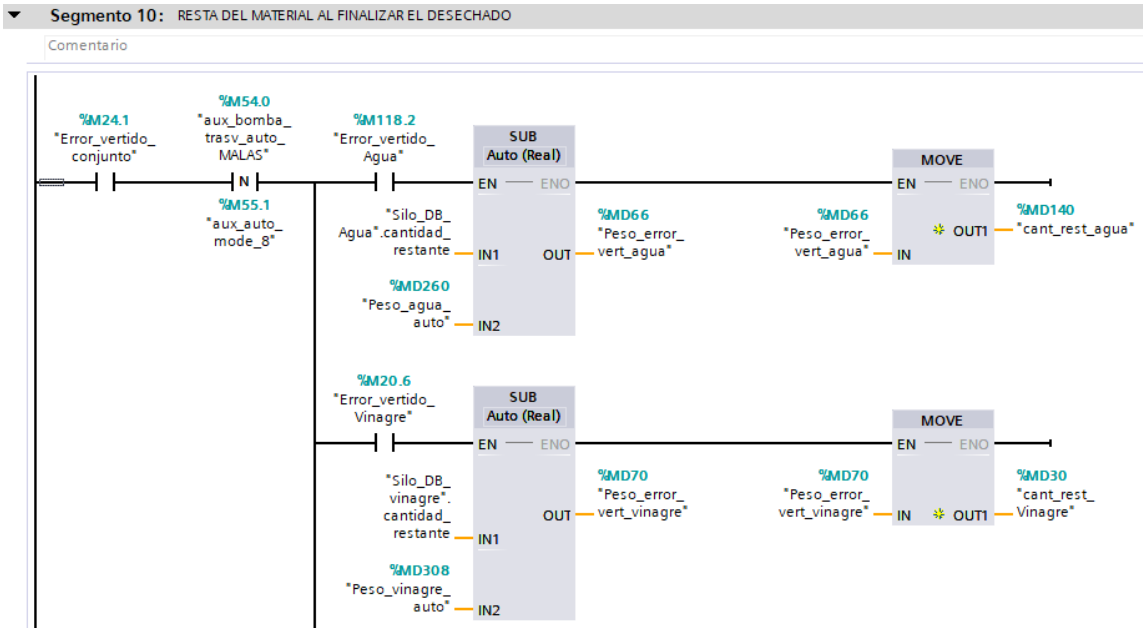


Figura 83 Programación segmento 10.a "Modo_automat" [OB124]

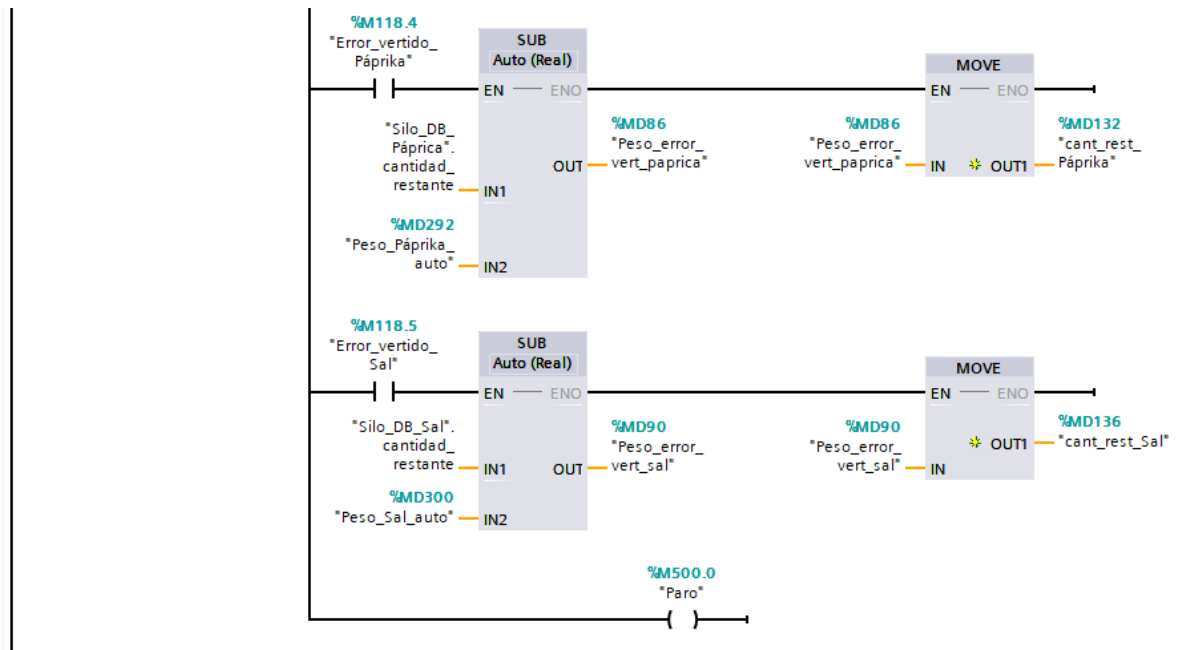


Figura 84 Programación segmento 10.b "Modo_automat" [OB124]

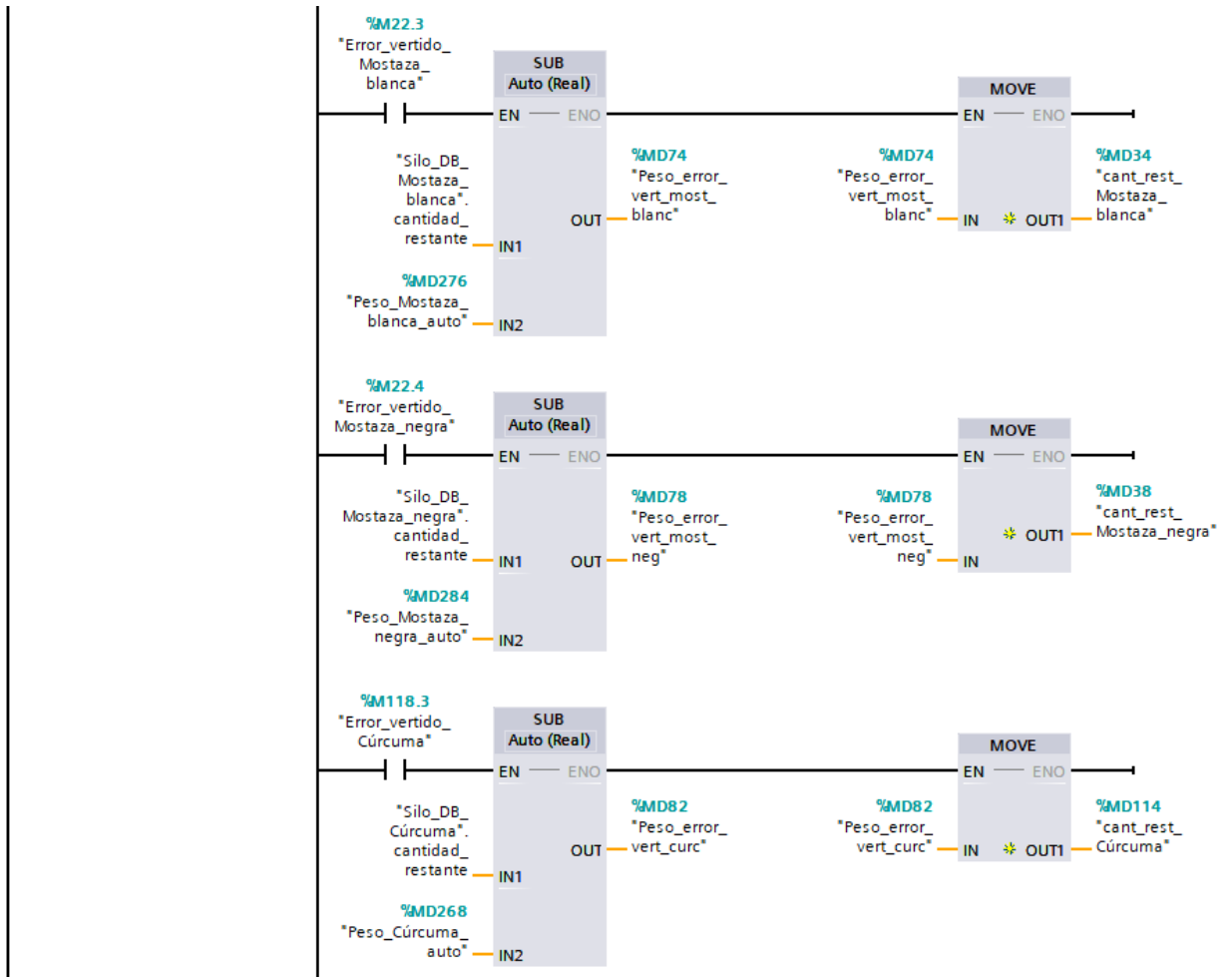


Figura 85 Programación segmento 10.c "Modo_automat" [OB124]

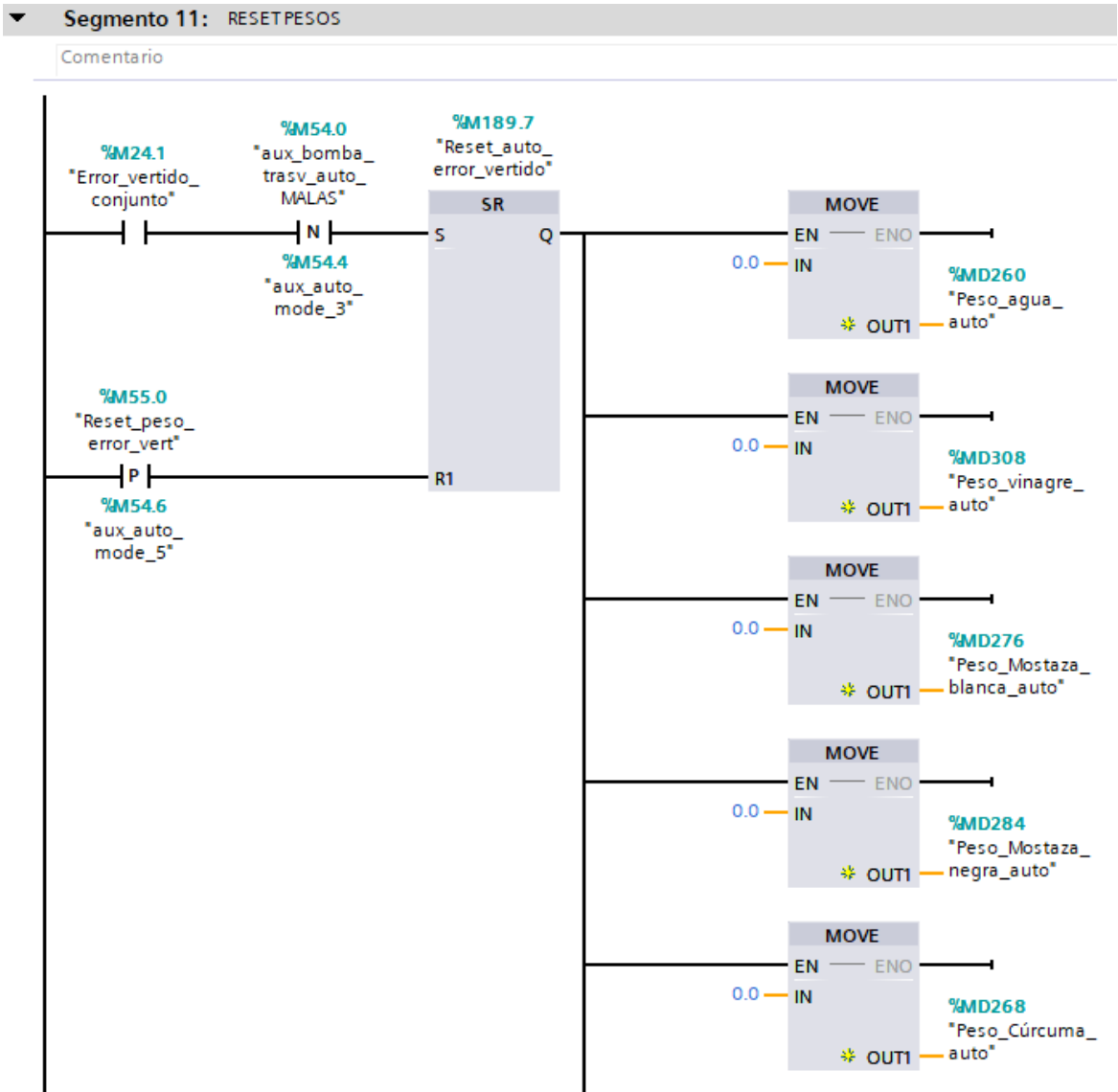


Figura 86 Programación segmento 11.a "Modo_automat" [OB124]

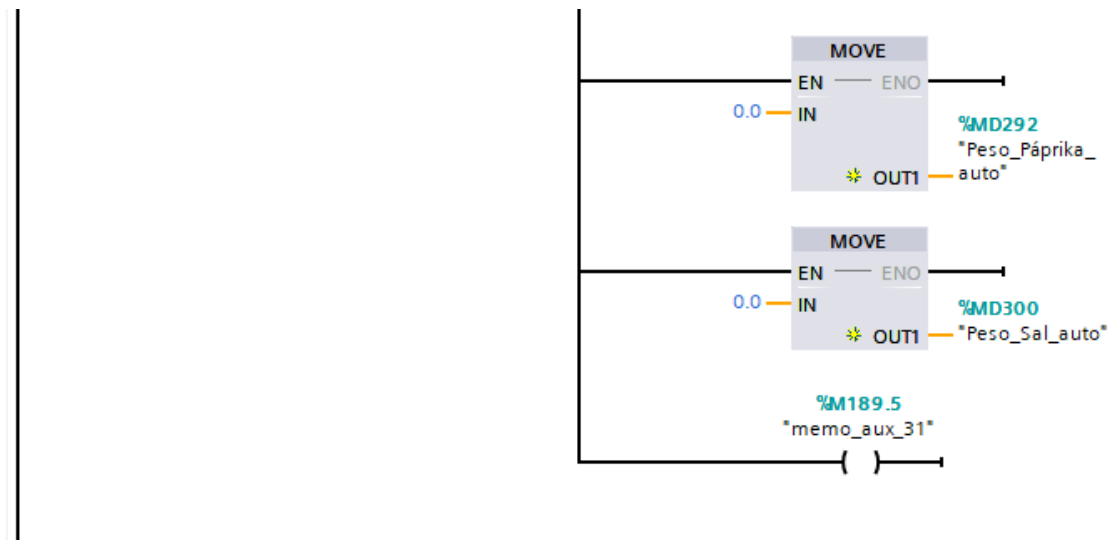


Figura 87 Programación segmento 11.b "Modo_automat" [OB124]

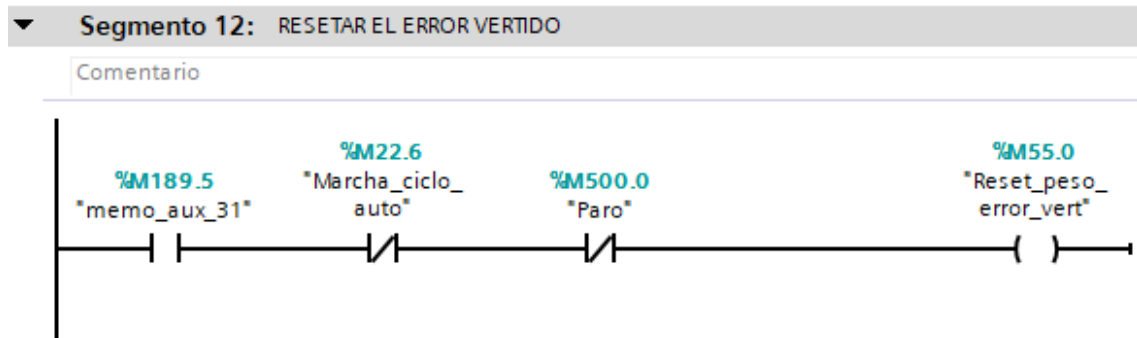


Figura 88 Figura 138 Programación segmento 12 "Modo_automat" [OB124]

II.IX) BLOQUE ORGANIZACIÓN “MOV_MANUALES” [OB123]

En este bloque de organización se ha escrito el código encargado de controlar el modo Manual. A continuación se muestra el código desarrollado:

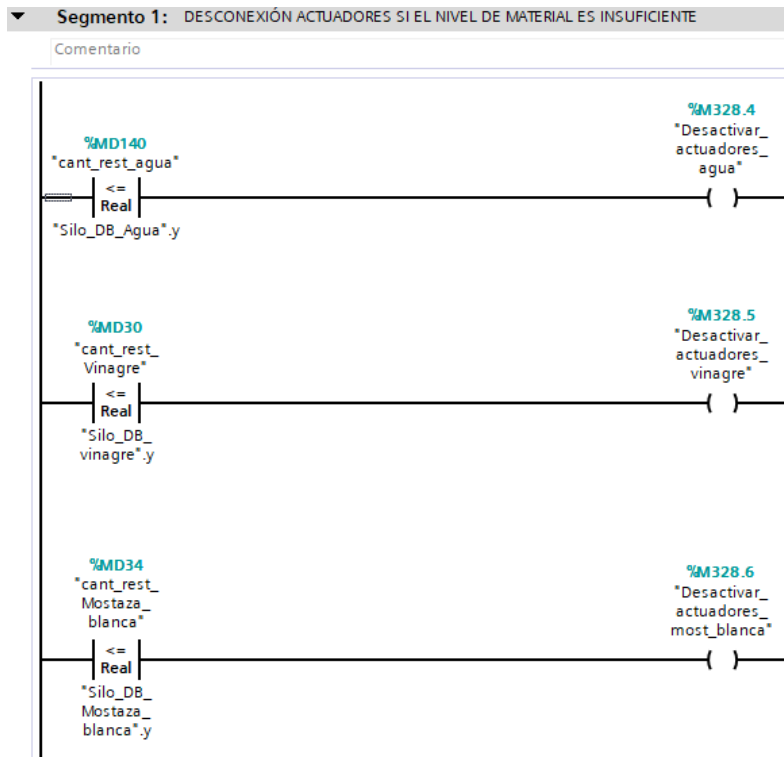


Figura 89 Programación segmento 1.a “Mov_manuales” [OB123]

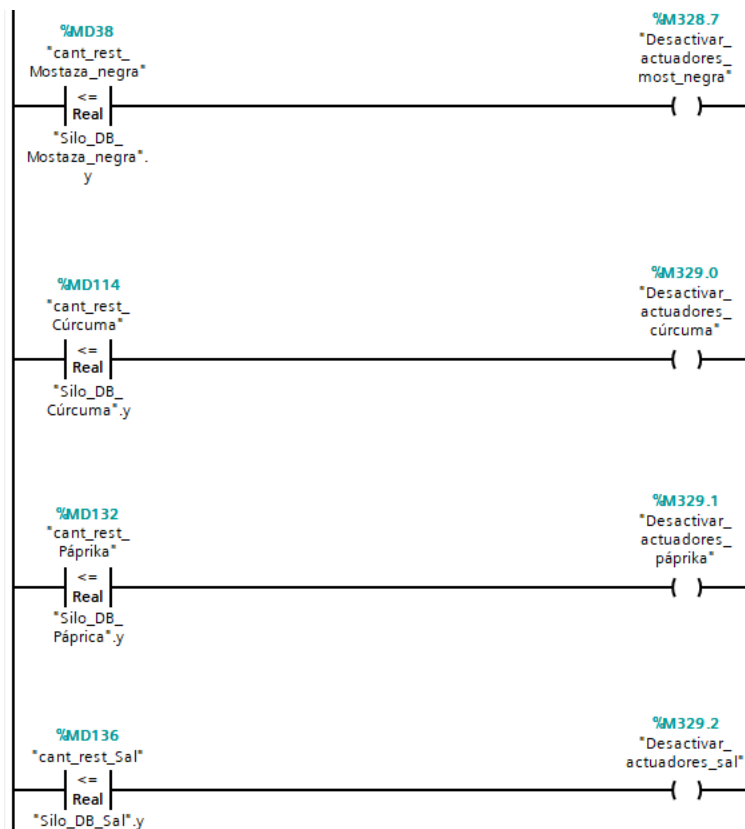


Figura 90 Programación segmento 1.b “Mov_manuales” [OB123]

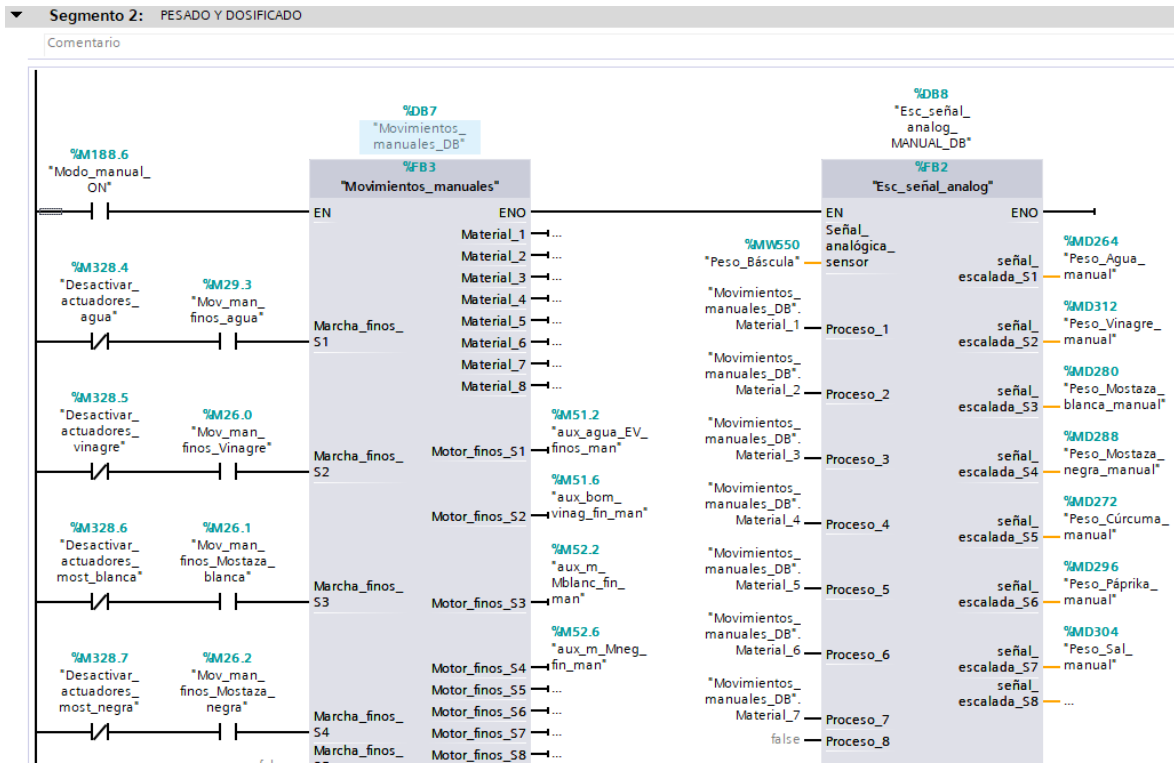


Figura 91 Programación segmento 2.a “Mov_manuales” [OB123]

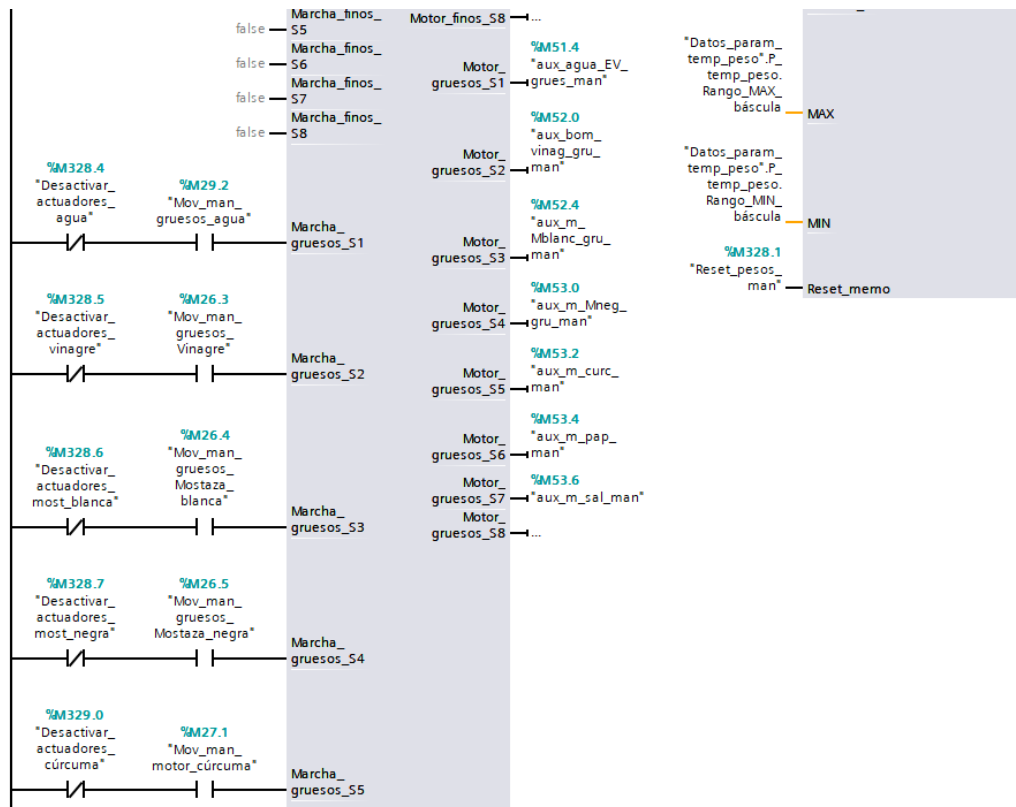


Figura 92 Programación segmento 2.b “Mov_manuales” [OB123]

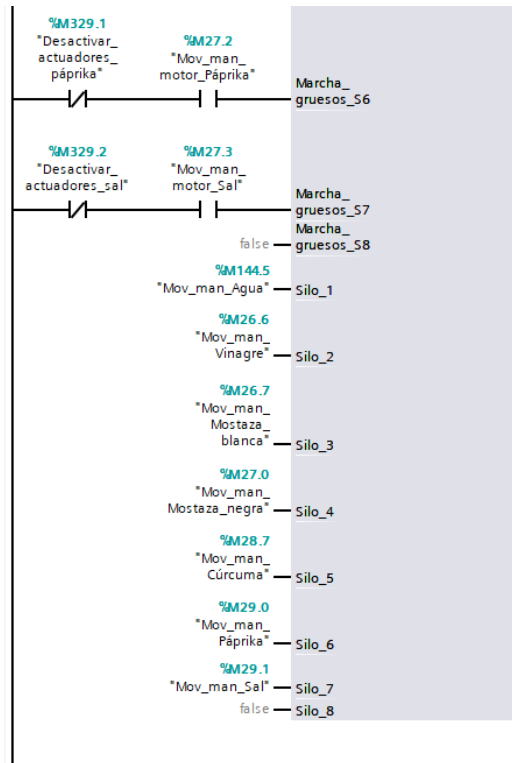


Figura 93 Programación segmento 2.c “Mov_manuales” [OB123]

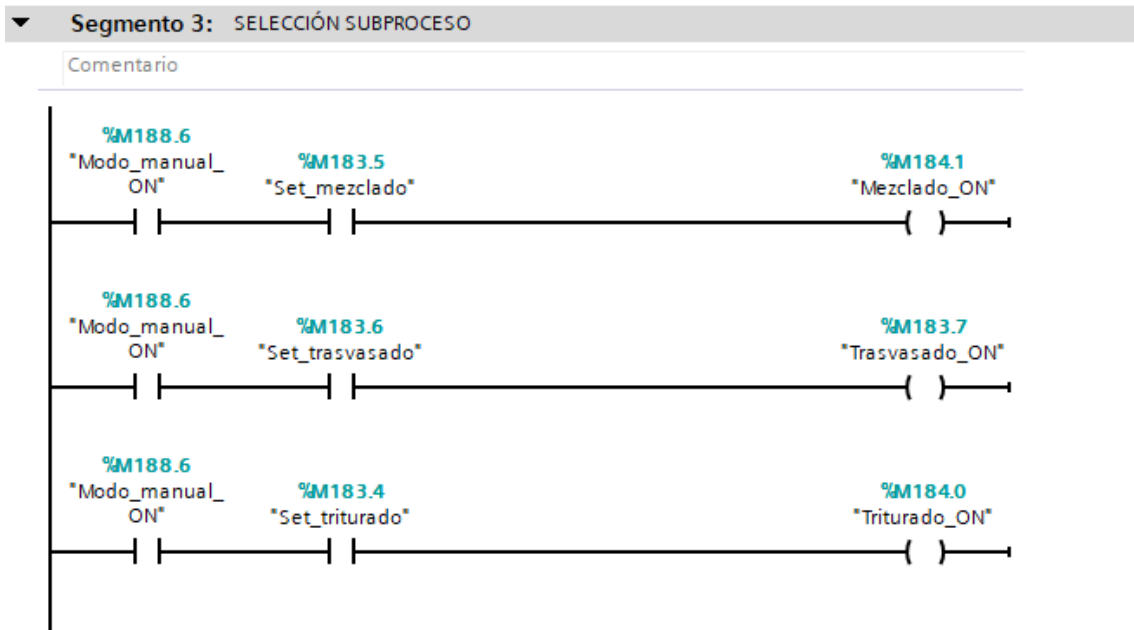


Figura 94 Programación segmento 3 “Mov_manuales” [OB123]

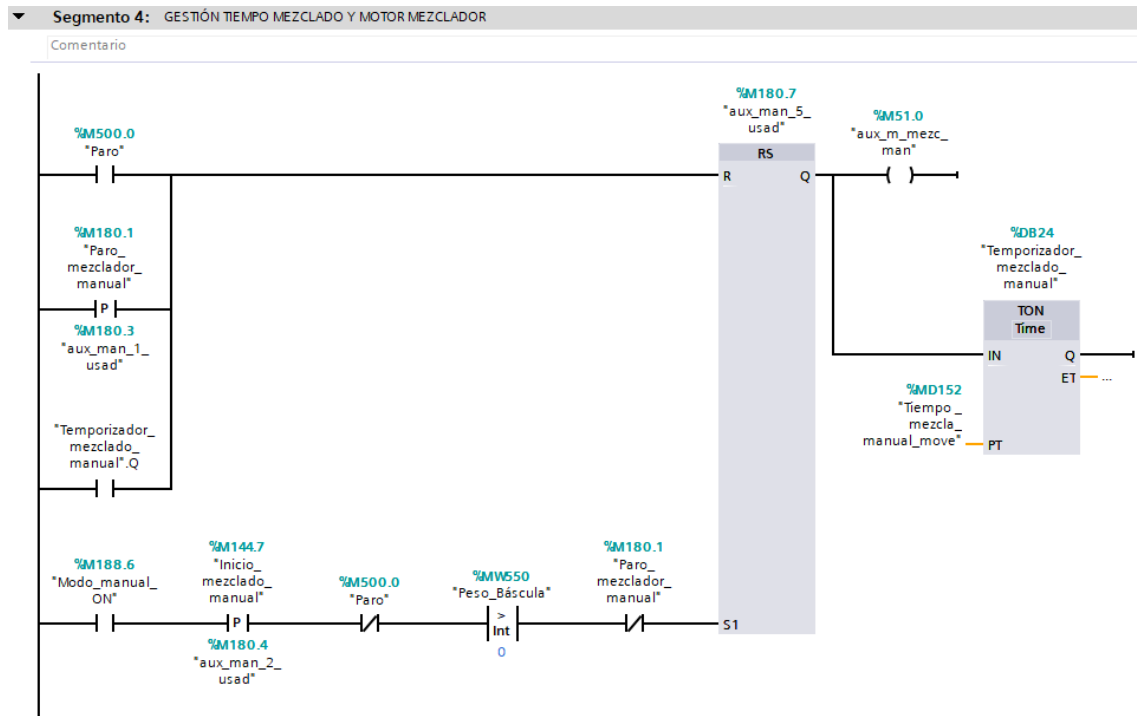


Figura 95 Programación segmento 4.a “Mov_manuales” [OB123]

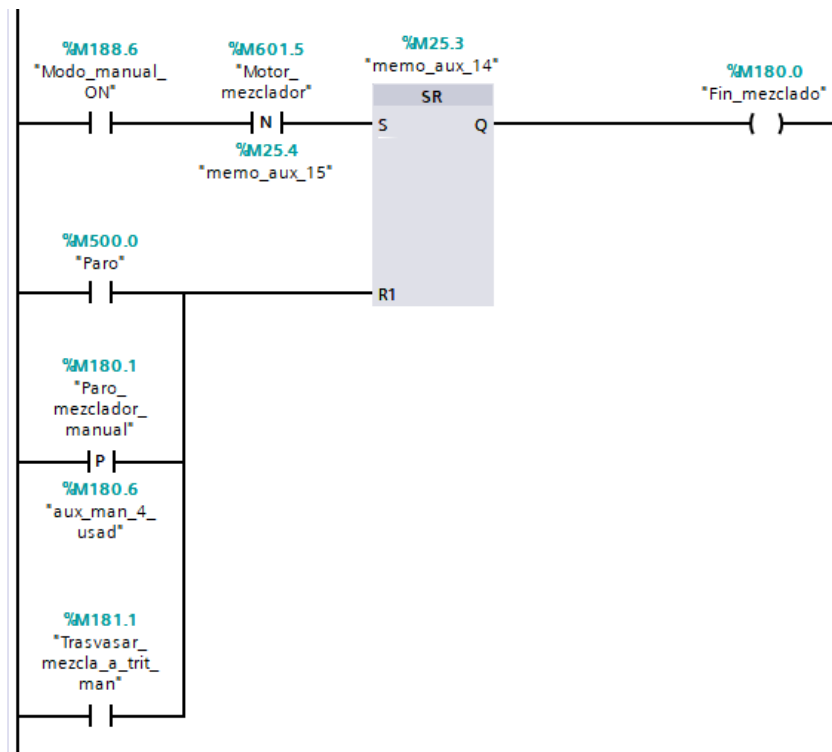


Figura 96 Programación segmento 4.b “Mov_manuales” [OB123]

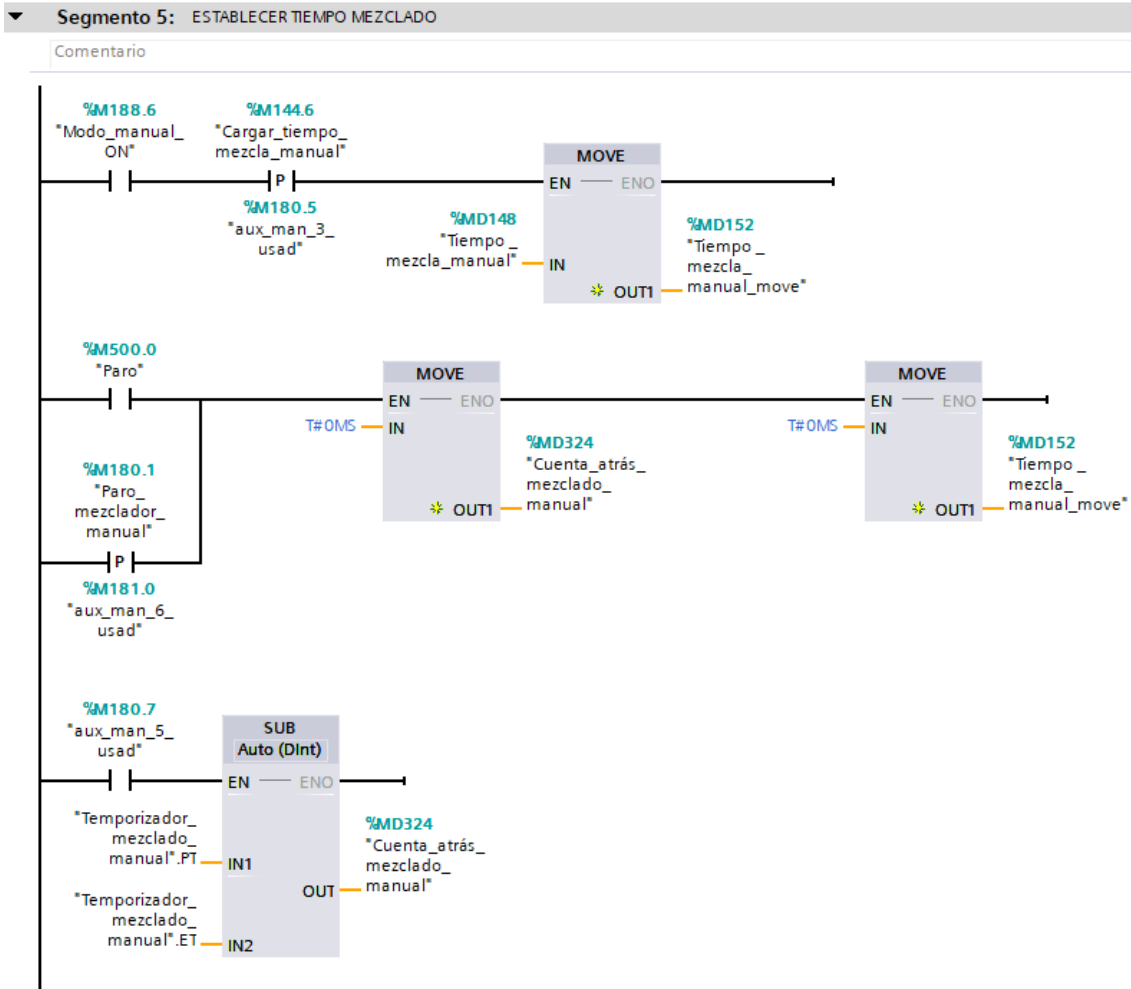


Figura 97 Programación segmento 5.a "Mov_manuales" [OB123]

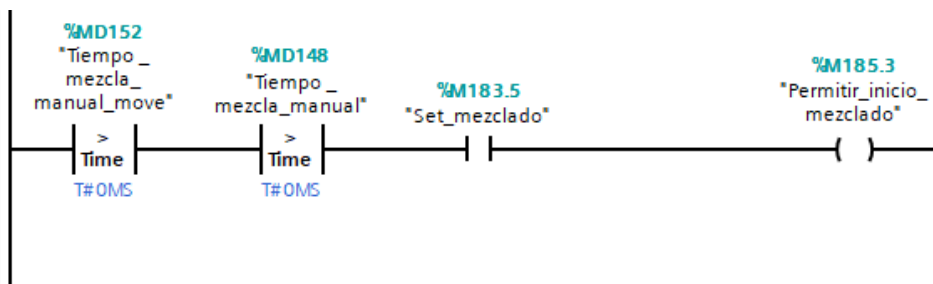


Figura 98 Programación segmento 5.b "Mov_manuales" [OB123]

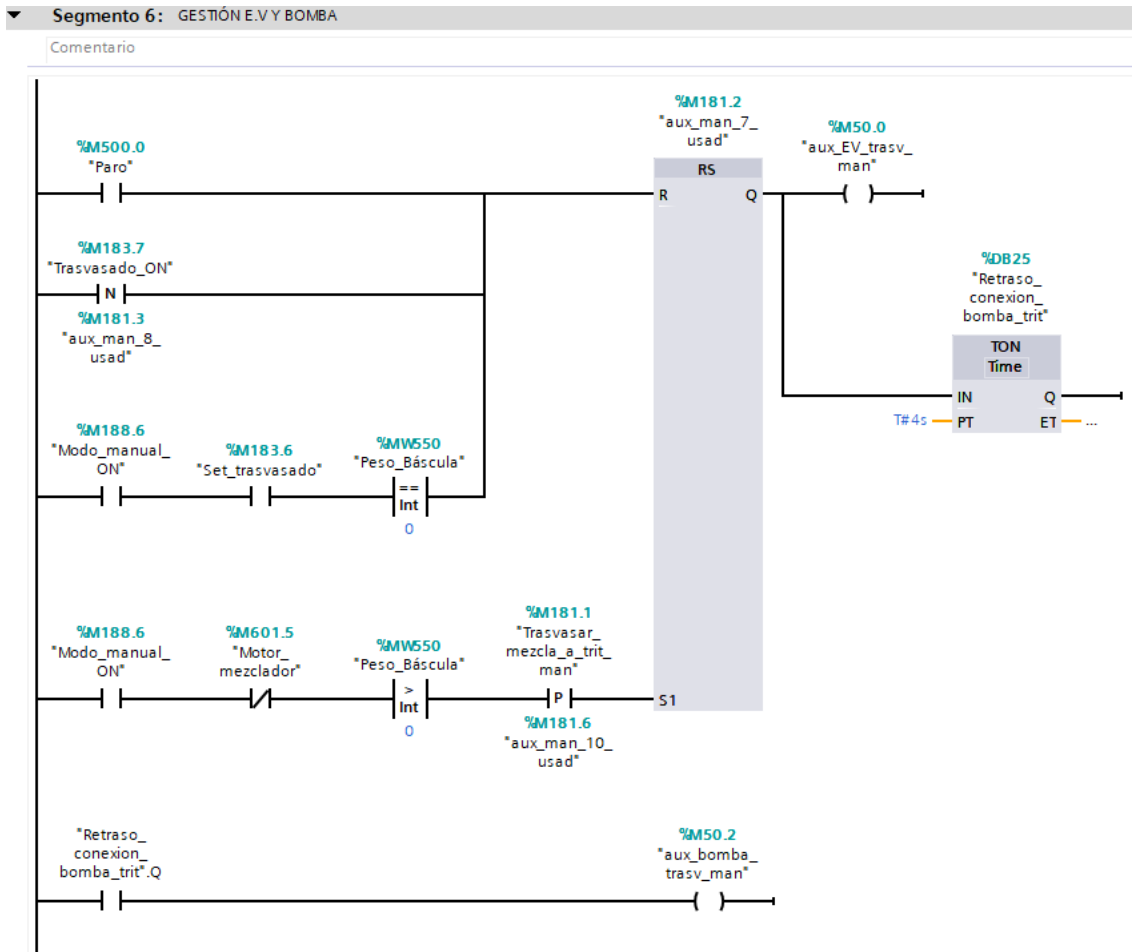


Figura 99 Programación segmento 6.a “Mov_manuales” [OB123]

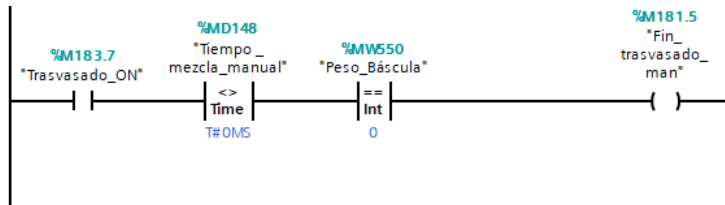


Figura 100 Programación segmento 6.b “Mov_manuales” [OB123]

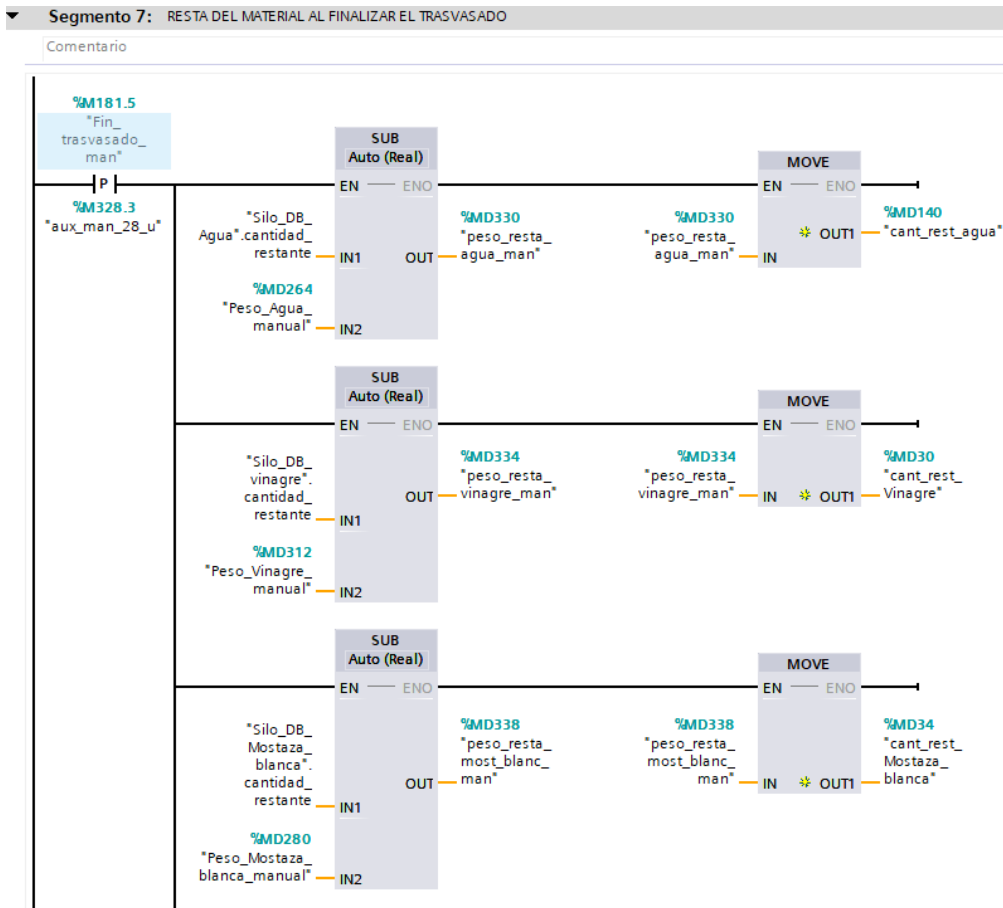


Figura 101 Programación segmento 7.a "Mov_manuales" [OB123]

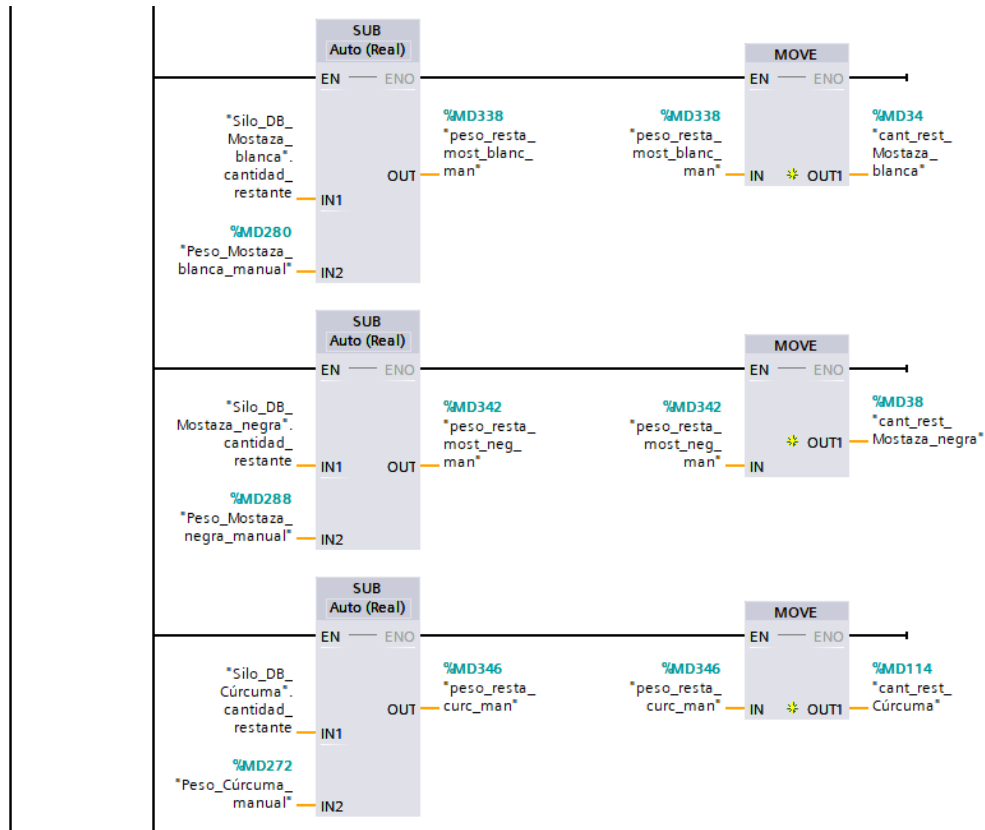


Figura 102 Programación segmento 7.b "Mov_manuales" [OB123]

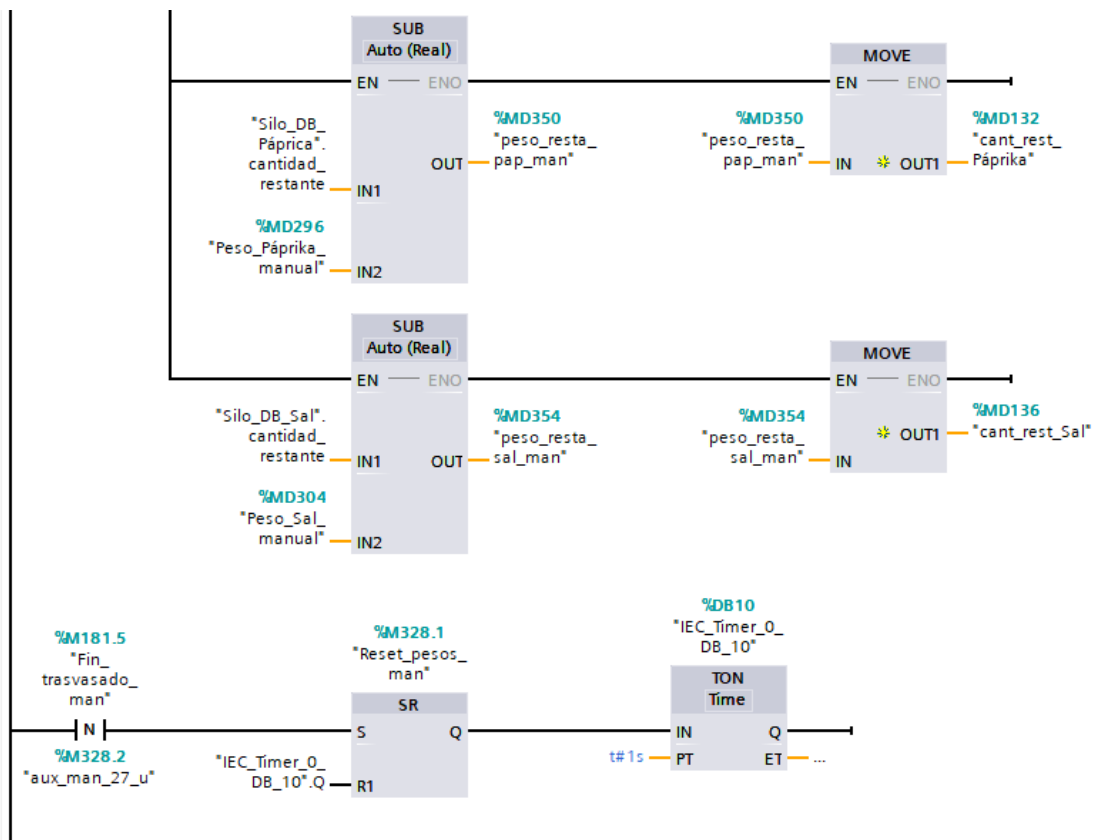


Figura 103 Programación segmento 7.c "Mov_manuales" [OB123]

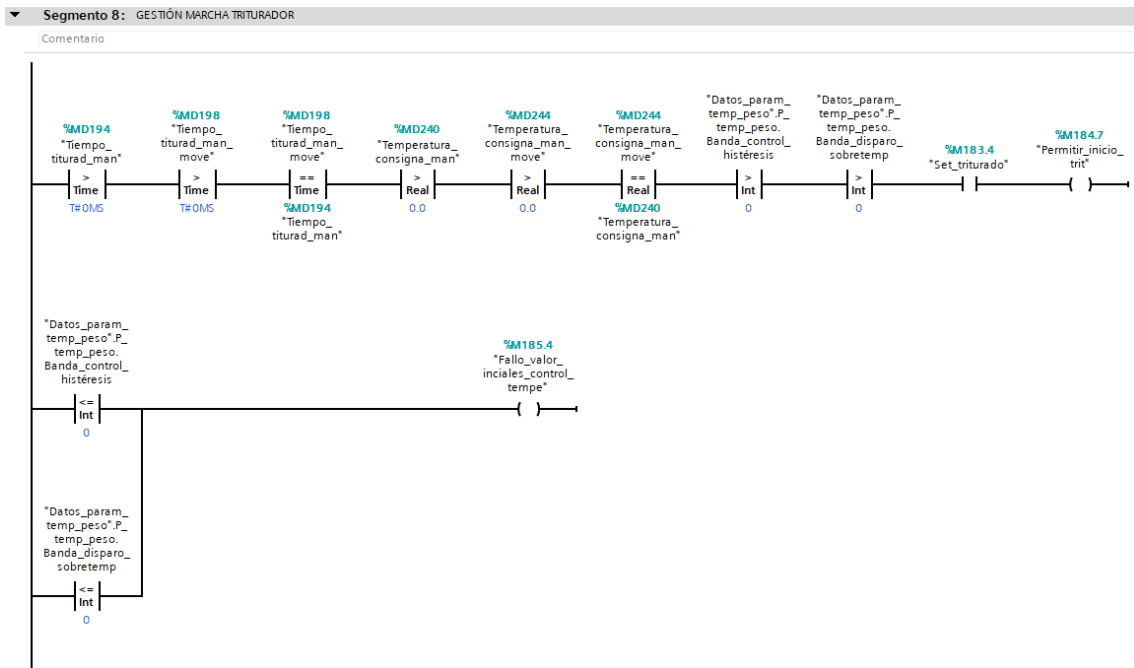


Figura 104 Programación segmento 8.a "Mov_manuales" [OB123]

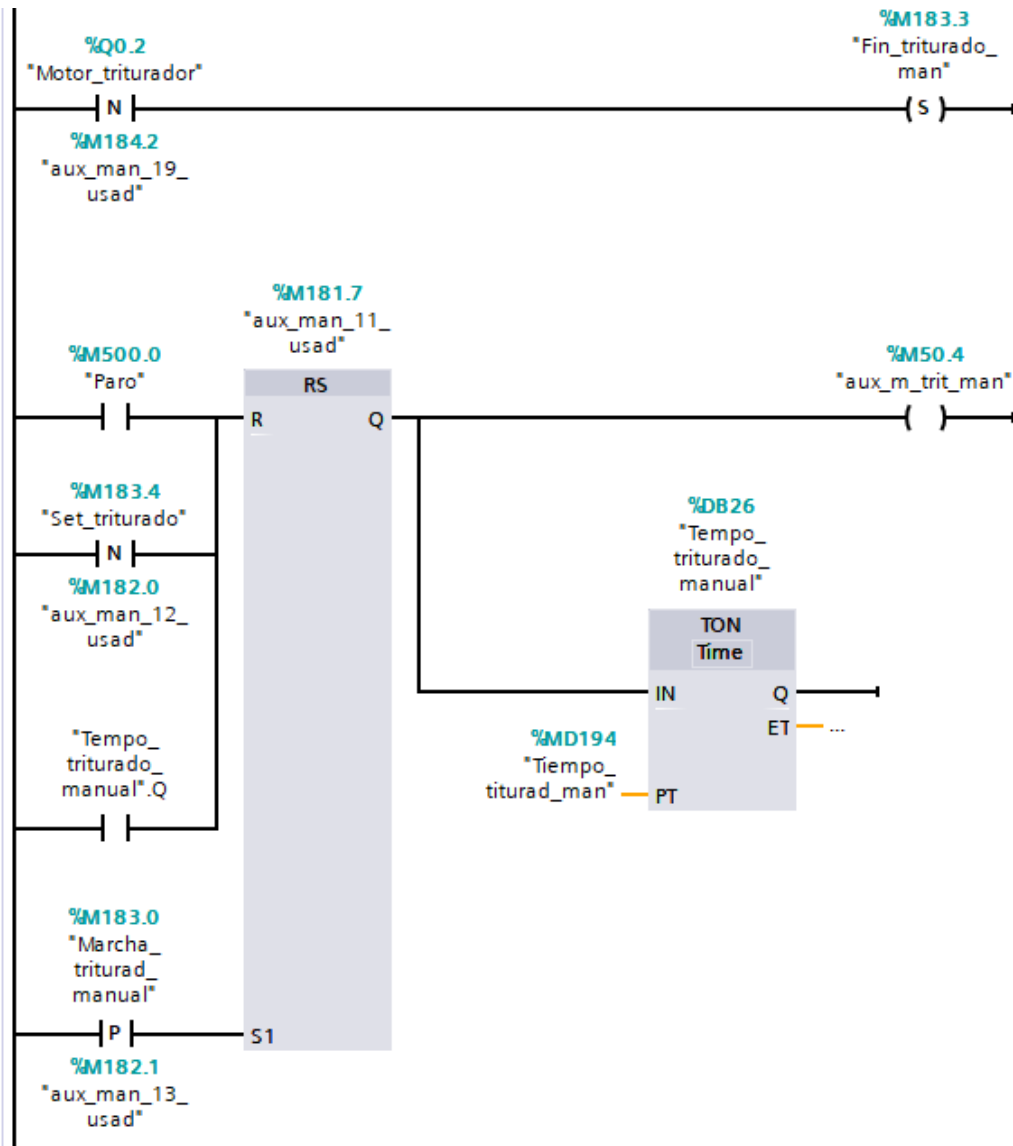


Figura 105 Programación segmento 8.b "Mov_manuales" [OB123]

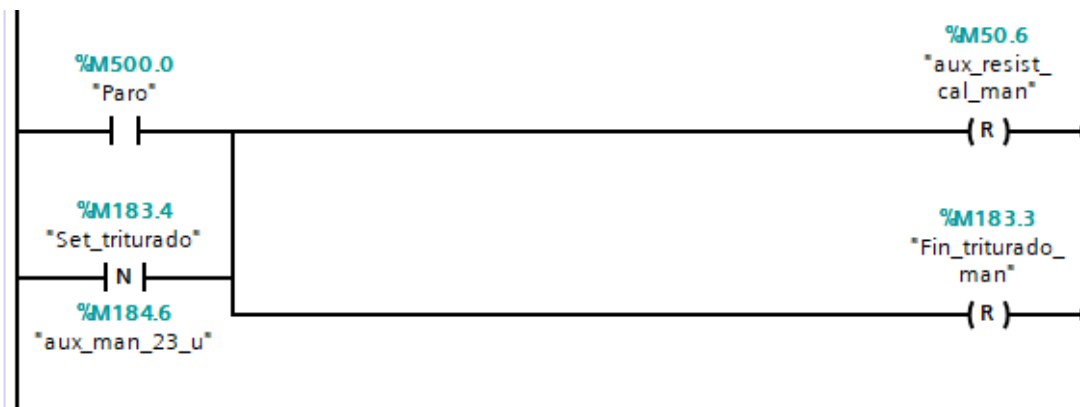


Figura 106 Programación segmento 8.c "Mov_manuales" [OB123]

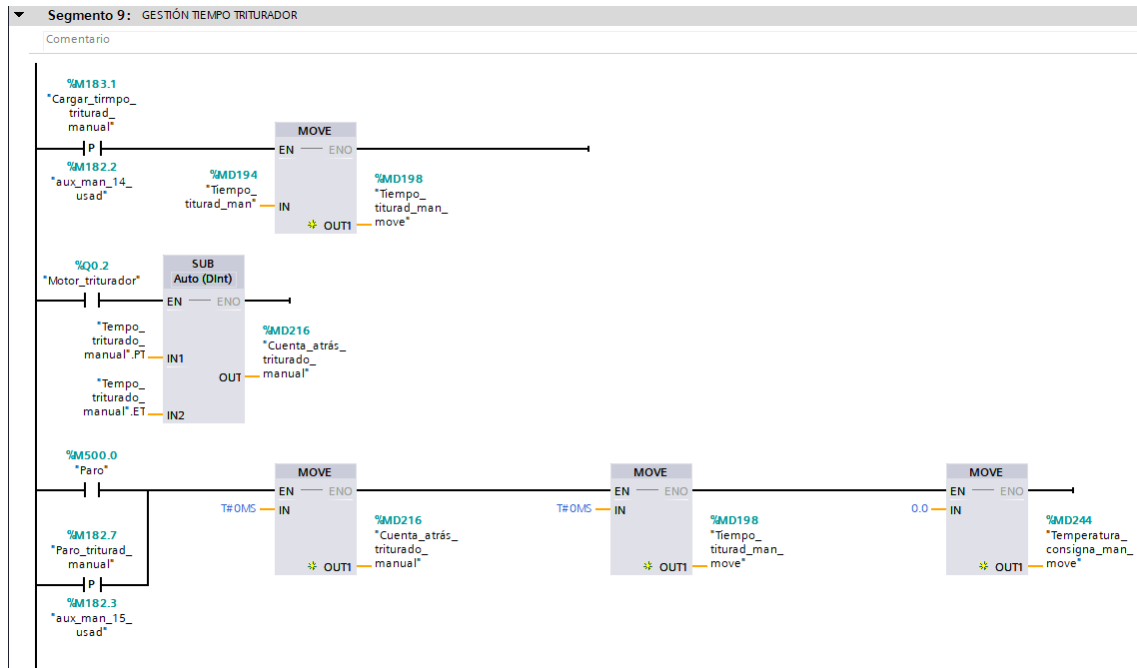


Figura 107 Programación segmento 9 “Mov_manuales” [OB123]

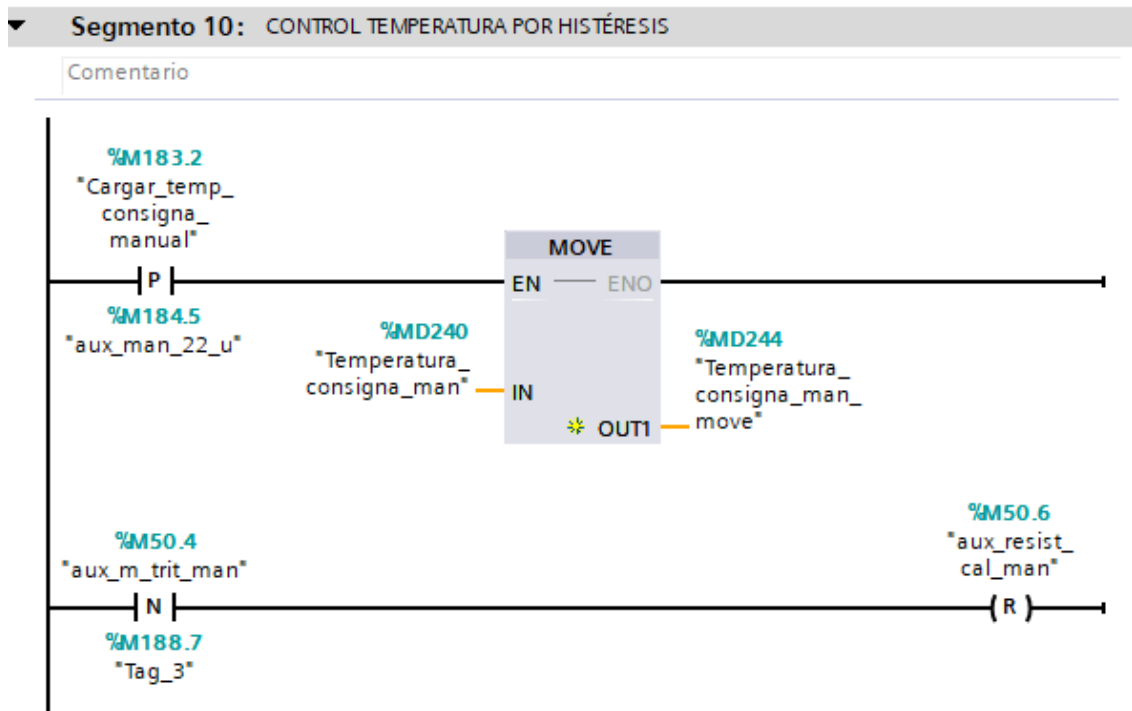


Figura 108 Programación segmento 10.a "Mov_manuales" [OB123]

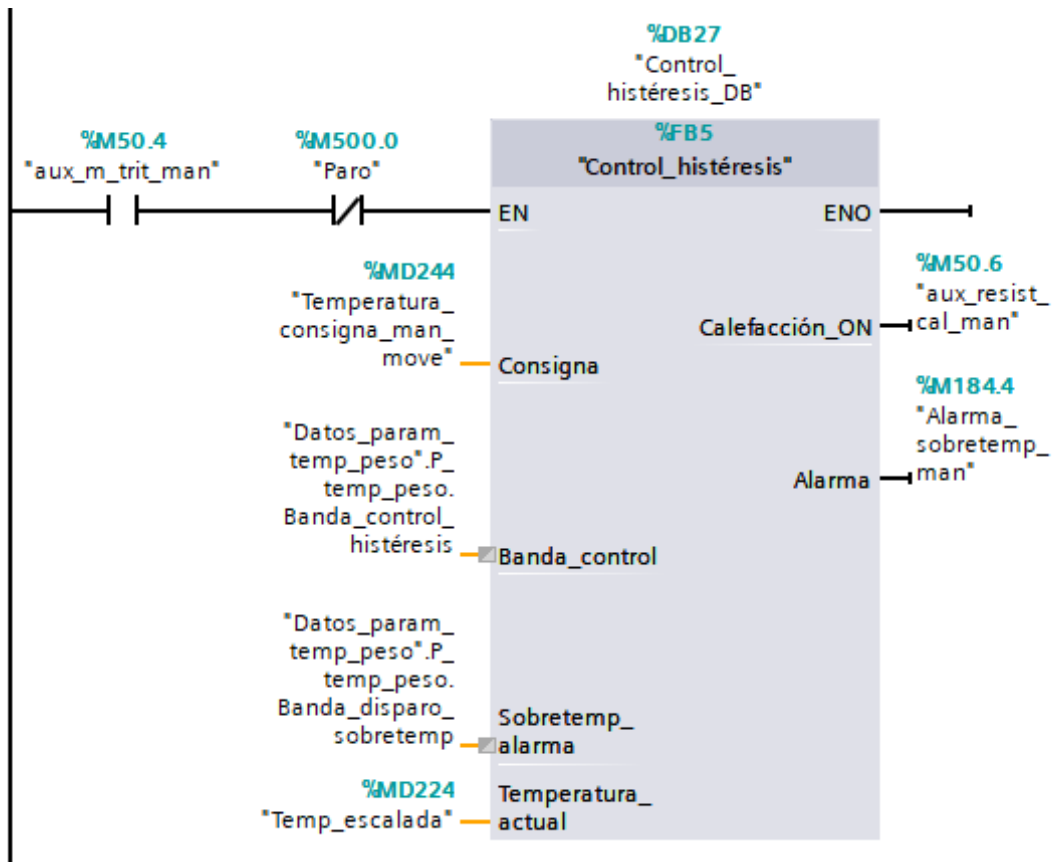


Figura 109 Programación segmento 10.b "Mov_manuales" [OB123]

ANEXO III: VISUALIZACIÓN DE LOS DISTINTOS LENGUAJES DE PROGRAMACIÓN

Para el proyecto han empleado distintos lenguajes de programación, orientados a la estandarización del proceso.

Para visualizar qué es un bloque de función se propone como ejemplo uno de los casos más sencillos, una maniobra de marcha paro de un motor:

- Se crea un FB y se elige el lenguaje de programación, en este caso SCL (que es como llama SIEMENS al lenguaje en texto estructurado):

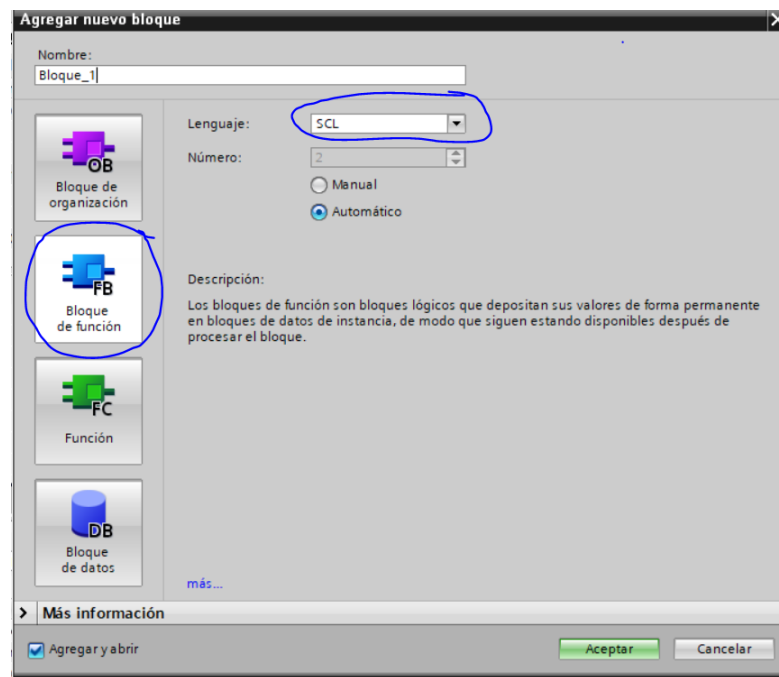


Figura 110 Ejemplo de creación FB TIA Portal

- Se declaran las variables dentro del bloque:

	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..	Valor de a..
1	Input							
2	entrada_B	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<Agregar>							
4	Output							
5	salida_B	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<Agregar>							

Figura 111 Declaración de variables en un FB

- Se escribe el código:

```

1 IF (#entrada_B = 1) THEN
2     #salida_B := 1;
3     // Statement section IF//
4 ELSE
5     #salida_B := 0;
6 END_IF;
7

```

Figura 112 Ejemplo escritura código SCL

Una vez creado el bloque, se debe de llamar en la función Main del programa, y se puede observar que aparece una caja cerrada únicamente con las variables que se le han declarado en el FB como acceso al interior del bloque:

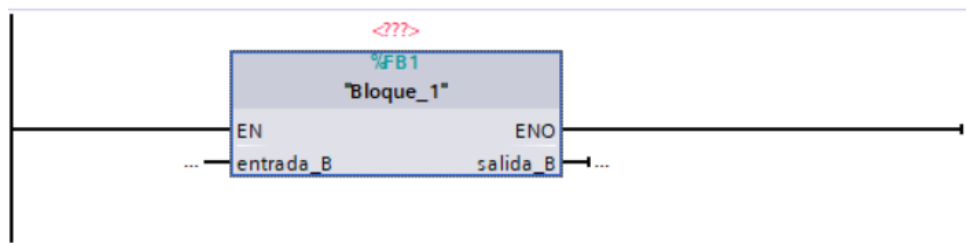


Figura 113 Ejemplo vista de una función creada al ser llamada

Hay que darle un nombre, y automáticamente se creará un DB donde almacenarlos estados de las variables que se le hayan asignado a las diferentes entradas o salidas del bloque:

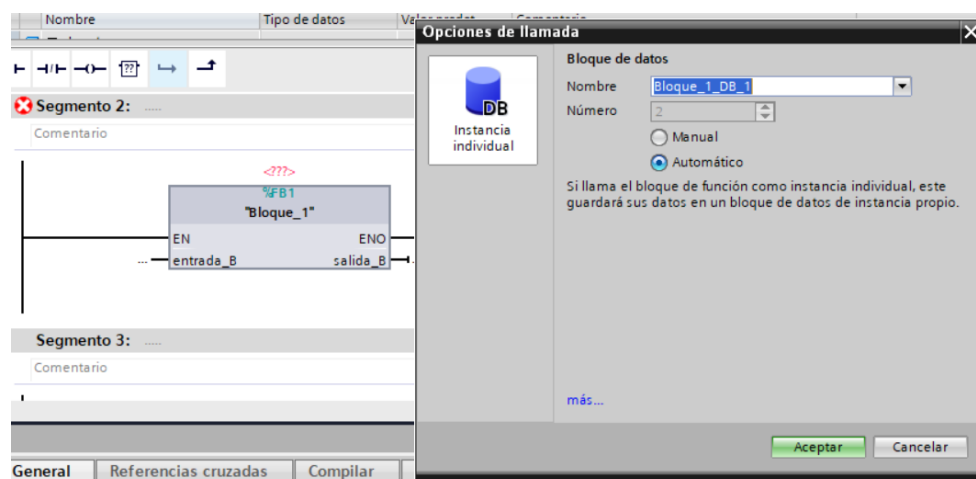


Figura 114 Ejemplo creación automática de un DB

Por último, se declaran las variables de las entradas/salidas del PLC, y se conectan al bloque de función:

	entrada	Tabla de variables e..	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	salida	Tabla de variables e..	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 115 Declaración de variables principales en el árbol del proyecto

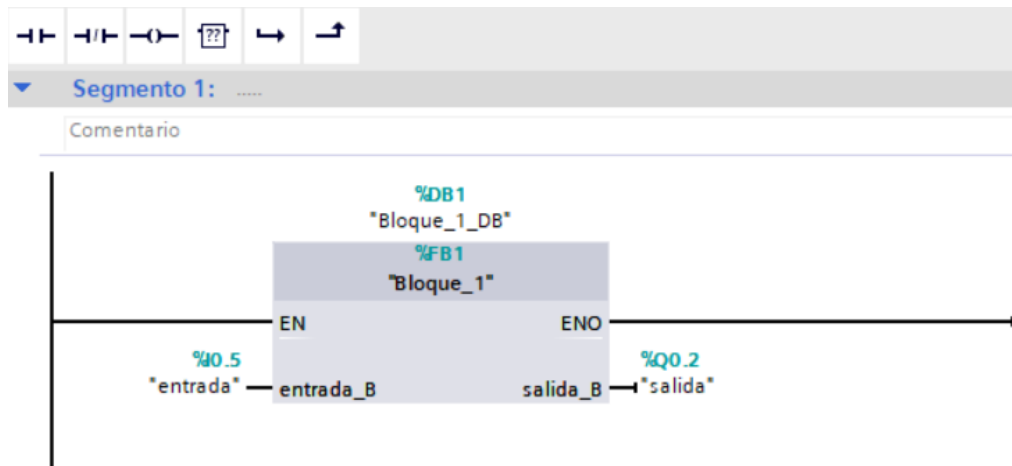


Figura 116 Ejemplo conexión entradas/salidas a una función creada

El **leguaje Ladder** es un lenguaje gráfico basado en los esquemas de la lógica cableada tradicional. Análogamente, serían diagramas de contactos que permiten la activación de bobinas, pero con todas las ventajas que permite la programación y los PLC's modernos. Tales como la incorporación de DFB, funciones personalizables, memorias internas etc.

Siguiendo con el ejemplo del marcha paro del motor, en Ladder quedaría tal que:



Figura 117 Ejemplo maniobra marcha paro en Ladder

Por último, el **ST (structured text)**¹ es un lenguaje de programación que se basa en el empleo de texto estructurado. La principal ventaja es la versatilidad que aporta a la hora de realizar operaciones matemáticas complejas, así como a la hora de implantar

operaciones o funciones lógicas. Siguiendo con el ejemplo, y como se ha expuesto más arriba:

```
1 IF (#entrada_B = 1) THEN
2     #salida_B := 1;
3     // Statement section IF//
4 ELSE
5     #salida_B := 0;
6 END_IF;
7
```

Figura 118 Ejemplo maniobra marcha paro en SCL

En este caso tan sencillo, el empleo de los bloques de función y del lenguaje SCL es innecesario, pero se han expuesto a título ilustrativo. Más adelante en este proyecto se verán las ventajas reales y el porqué del empleo de estas herramientas que ofrece el entorno de programación de TIA Portal.

ANEXO IV: CREACIÓN DE UN TIPO DE DATO

La creación de este tipo de datos se realiza desde el árbol del proyecto, en la carpeta “Tipo de datos PLC”, “Agregar nuevo tipo de datos”.

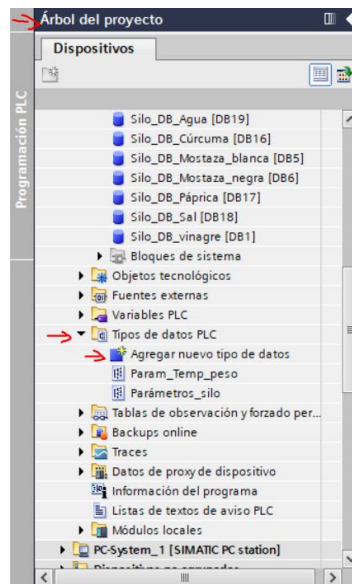


Figura 119 Vista del árbol del proyecto

Una vez dentro de la interfaz, se declaran los nombres y tipo de las variables que se quiere que contenga este tipo de dato:

Nombre	Tipo de datos	Valor predet.	Accesible d...	Escrib...	Visible en ..	Valor de a..	Comentario
capacidad_max	Real	0.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
A	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ calcular el nivel aviso
B	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ calcular el nivel de fallo
M	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ tener en cuenta C.D.C y desconectar gruesos
N	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ tener en cuenta C.D.C y desconectar finos
tiempo_gruesos	Time	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
tiempo_finos	Time	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Tolerancia_pesado	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Cola_de_carga	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figura 120 Declaración de variables del tipo de dato "Parámetros_silos"

Y una vez creado, al seleccionar el tipo de datos de las variables en otros lugares del proyecto (como se ha visto en las entradas de la función "Silo"), permitirá seleccionar "Parámetros_silo" como un tipo de dato más dentro de las posibilidades genéricas como son los Bool, Byte, Word etc. En lo concerniente a la aplicación planteada, se ha creado un DB **Datos_silos_param** para concentrar la gestión de la parametrización de los silos:

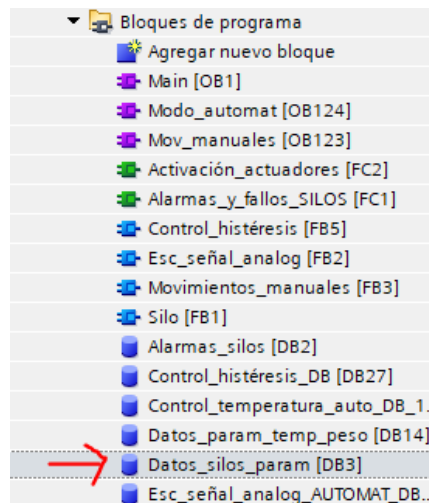


Figura 121 Vista en el árbol del proyecto del DB creado para almacenar el tipo de dato "Parámetros_silos"

Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de a..	Comentario
Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
P_silos	Array[1..8] o...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
P_silos[1]	"Parámetros_silo"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
capacidad_max	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
A	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ calcular el niv
B	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ calcular el niv
M	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ tener en cuer
N	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	P/ tener en cuer
tiempo_gruesos	Time	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
tiempo_finos	Time	T#0ms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Tolerancia_pes...	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Cola_de_carga	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
P_silos[2]	"Parámetros_silo"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
P_silos[3]	"Parámetros_silo"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figura 122 Vista en el interior del DB creado para almacenar el tipo de dato "Parámetros_silos"

ANEXO V: AÑADIR LA PANTALLA PARA LA INTERFAZ HMI

Para escoger la pantalla con la que se va a trabar hay que ir a “Agregar dispositivo” → “sistemas PC” → “SIMATIC HMI Application” → “WinCC RT Profesional”

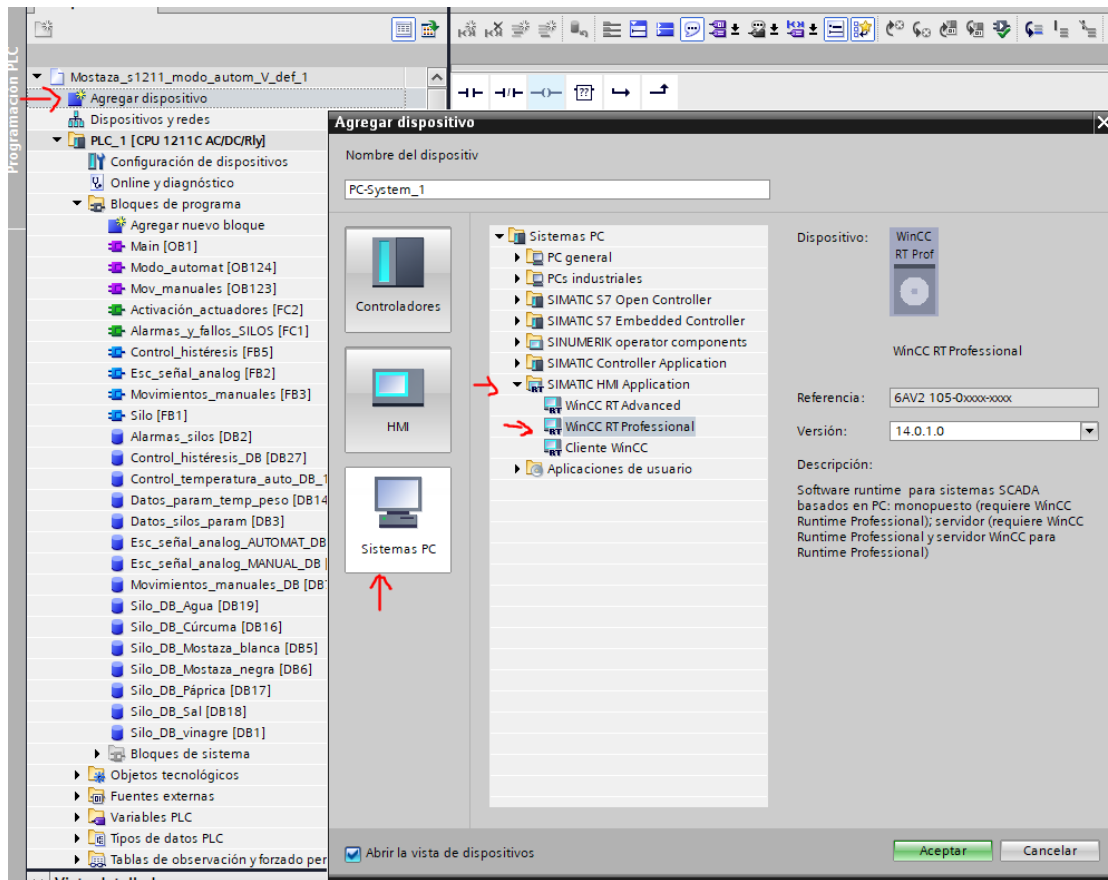


Figura 123 Añadir un sistema PC para el control HMI

Como se puede observar en la figura 123, Siemens ofrece gran variedad de pantallas HMI, en el caso del proyecto, se ha escogido un sistema PC, por lo el SCADA se controlará desde la pantalla de un ordenador normal, el cual estará conectado al autómata.

Al darle a aceptar se crea automáticamente el nuevo sistema PC. El siguiente paso es establecer la comunicación, para ello se deberá de abrir la pestaña “dispositivos y redes”. Lo primero que se ve es que está por un lado el PLC y por otro el PC con el que se pretende establecer la comunicación.

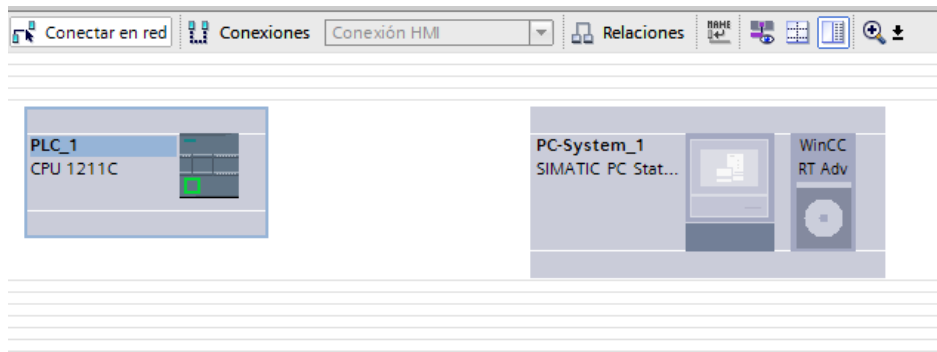


Figura 124 Vista de los dispositivos a comunicar

Es necesario configurar cómo se va a establecer la comunicación, en este caso será Profinet, por lo que hay que ir al catálogo de hardware y seleccionar un módulo de comunicación apropiado, para este caso el genérico, pues la comunicación se realizará a través de un puerto tipo RJ45 del PC:

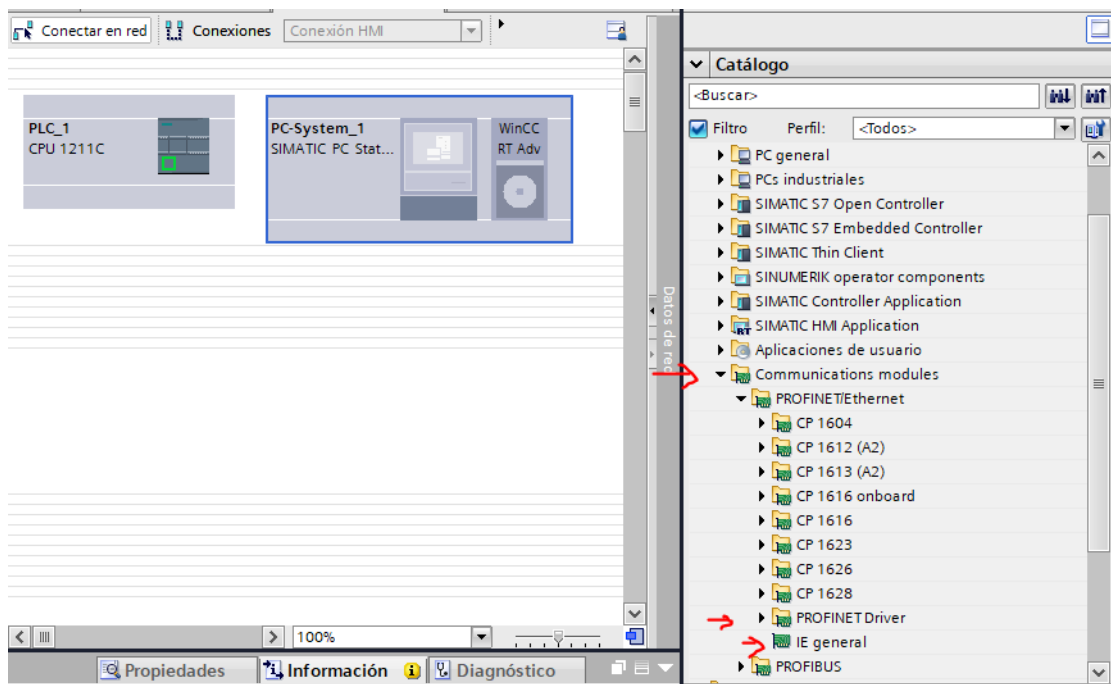


Figura 125 Configuración hardware necesario para la comunicación profinet

Hay que pinchar, y sin soltar arrastrarlo al PC System:

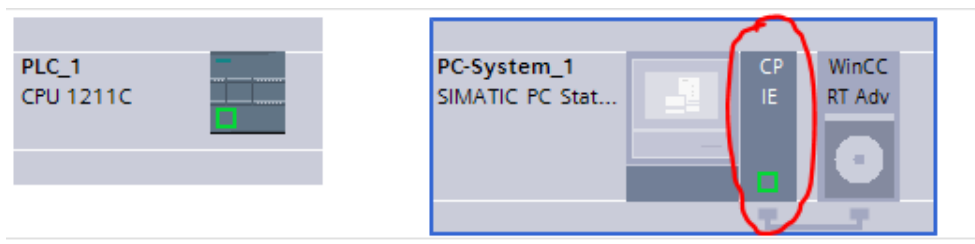


Figura 126 Vista del hardware necesario para la comunicación comunicado

Y ahora faltará establecer la comunicación entre ambos. Para ello, hay que pinchar en el recuadro resaltado en verde del PLC y unirlo con el recuadro resaltado en verde del PC System:

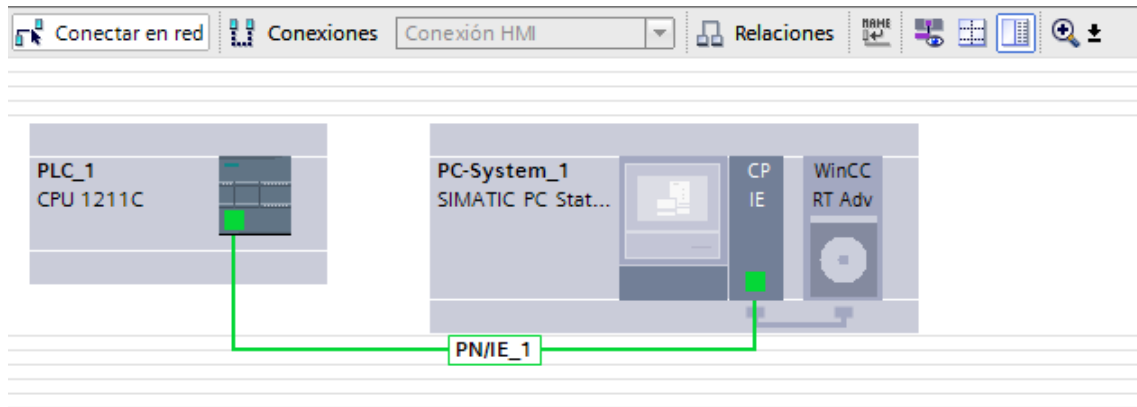


Figura 127 Vista conexión PN/IE

Y para comprobar que la conexión HMI se ha realizado correctamente, se puede pulsar en “conexiones” y aparecerán resaltados en azul los elementos conectados

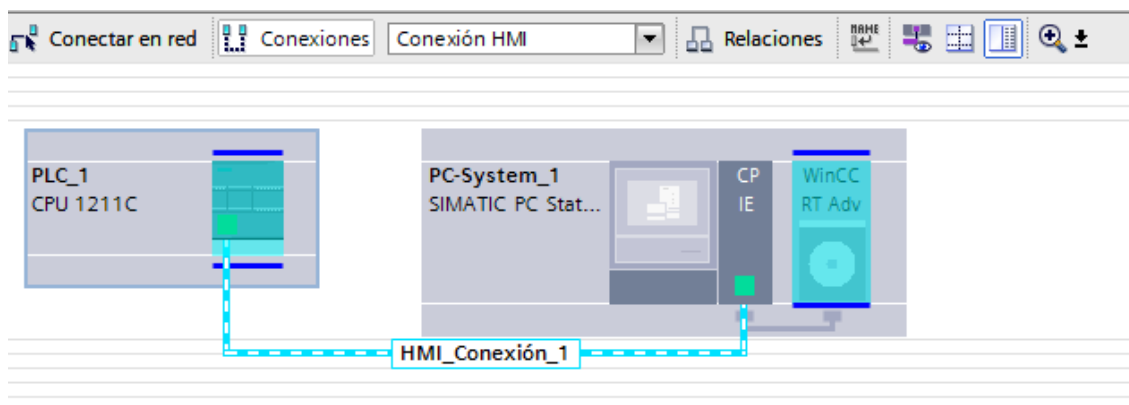


Figura 128 Vista conexión PN/IE establecida

Las IP de los dispositivos deben de estar en rango, por lo que lo más cómodo es determinar la IP del PC con el que se esté trabajando y cambiar la del autómeta. Por ejemplo, en este caso se han definido las siguientes direcciones IP para la comunicación PLC-PC:

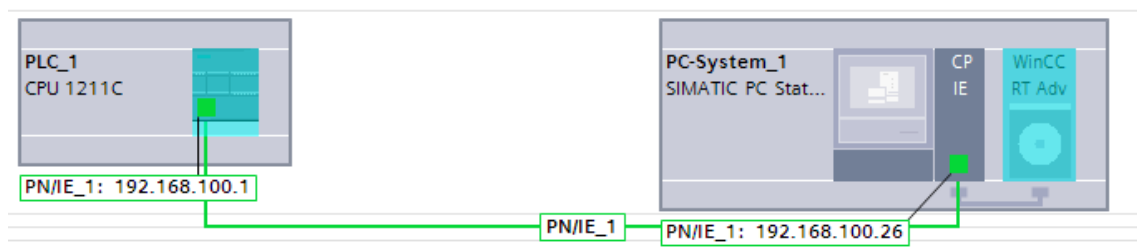


Figura 129 Vista de las IP de los dispositivos configurados

Una vez realizada la conexión, en el árbol del proyecto aparecerá el nuevo sistema PC:

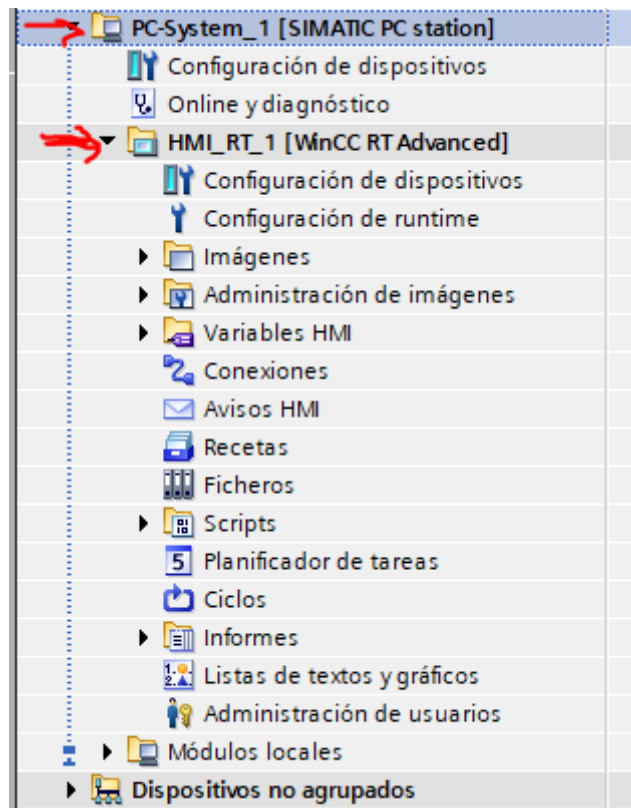


Figura 130 Vista del nuevo sistema PC para el HMI en el árbol del proyecto

Y ahora ya se puede de crear la interfaz gráfica que encontrará el usuario al manejar el programa.

Una interfaz intuitiva siempre es de gran ayuda para el personal que va a manejar la instalación, por lo que se ha perseguido principalmente este propósito.

ANEXO VI CARACTERÍSTICAS HARDWARE

En el Anexo VI se incluyen las hojas de características del PLC y de la ampliación para este:

Las hojas de características se pueden descargar de la página web de SIEMENS.

- Enlace para la CPU:

<https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7214-1BG40-0XBO>

- Enlace para la ampliación:

<https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7223-1PL32-0XBO>

ANEXO VII MANUAL GENERAL PLC S7-1200

Consultar CD

ANEXO VIII PROGRAMA INTEGRO SIN COMENTARIOS

Consultar CD

