

# Práctica 1: Introducción a la programación del autómata S7-1200



## Automatización y Robótica

Francisco Javier Pérez Martínez

Francisco Joaquín Murcia Gómez

10 de abril de 2022



## Índice

<b>1. Descripción</b>	<b>3</b>
<b>2. Ejercicio 1: Control de un motor desde un panel de mando</b>	<b>3</b>
2.1. Tabla de variables . . . . .	3
2.2. Programa en KOP . . . . .	4
2.3. Ejecución . . . . .	5
2.4. Interfaz HMI . . . . .	10
<b>3. Ejercicio 2: Control de un sistema de alarma de una vivienda</b>	<b>11</b>
3.1. Tabla de variables . . . . .	12
3.2. Programa en KOP . . . . .	12
3.3. Ejecución . . . . .	15
3.4. Interfaz HMI . . . . .	19
<b>4. Ejercicio 3: Control de un desapilador automático.</b>	<b>20</b>
4.1. Tabla de variables . . . . .	22
4.2. Programa en KOP y Ejecución . . . . .	23
4.3. Interfaz HMI . . . . .	35



## 1. Descripción

Para el presente documento, correspondiente a la práctica 1 de la asignatura, se trabajará la programación básica del S7-1200 con el software TIA PORTAL. Durante el resto del documento, por cada ejercicio realizado, se mostrarán la organización de las variables y programas en su correspondiente PLC programado mediante el lenguaje KOP. Además, para comprobar el funcionamiento del programa, se mostrarán los diferentes pasos de ejecución asimismo como la simulación del programa mediante una interfaz HMI.

## 2. Ejercicio 1: Control de un motor desde un panel de mando

El funcionamiento del sistema debe ser el siguiente:

- El interruptor ON/OFF pone en marcha o para el sistema y enciende o apaga la lámpara de funcionamiento.
- Si el interruptor está en ON y se actúa sobre la palanca de giro positivo, el motor gira a derechas y se activa la lámpara indicativa de ese sentido de giro.
- Si el interruptor está en ON y se actúa sobre la palanca de giro negativo, el motor gira a izquierdas y se activa la lámpara indicativa de ese sentido de giro.
- Si, con el interruptor en ON, se actúa sobre ambas palancas, el motor no girará en ningún sentido, pero sí se encenderán las dos lámparas indicativas del sentido de giro.

### 2.1. Tabla de variables

En primer lugar, generaremos las variables de entrada y salida siguiendo la tabla del enunciado:

Tabla de variables estándar								
	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	
1	Interruptor On/Off	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Palanca de giro positivo motor	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Palanca de giro negativo motor	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Lámpara funcionamiento	Bool	%M10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Lámpara sentido positivo de giro	Bool	%M10.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Lámpara sentido negativo de g..	Bool	%M10.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Contactor giro positivo motor	Bool	%M10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Contactor giro negativo motor	Bool	%M10.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Sensor temperatura	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Piloto panel de mando	Bool	%M10.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	<Agregar>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 1: Tabla de variables. Ejercicio 1



## 2.2. Programa en KOP

Segmento creado a partir de la especificación dada:

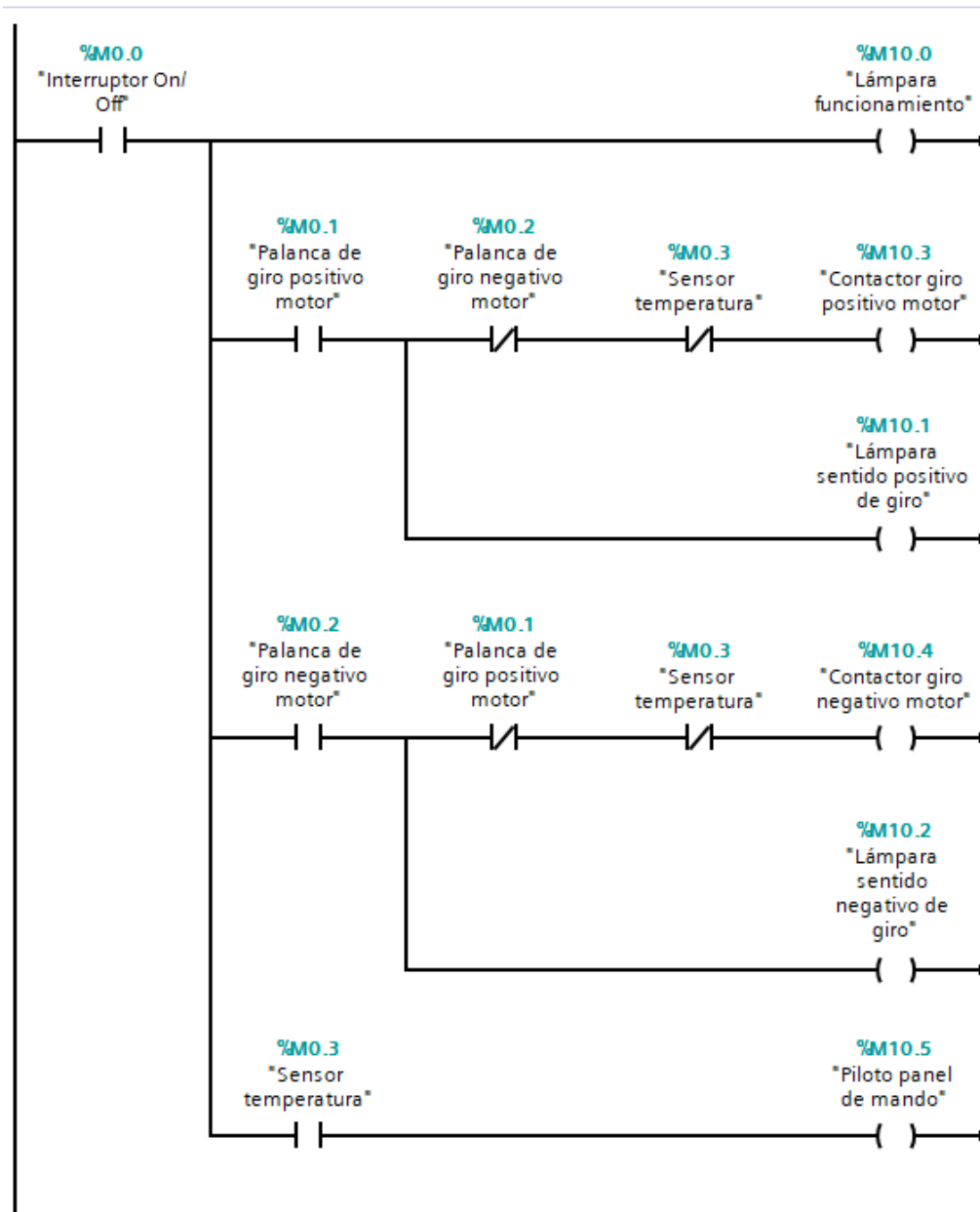


Figura 2: Segmento realizado para el ejercicio 1



## 2.3. Ejecución

Antes de proceder con la ejecución del programa para comprobar su funcionamiento, debemos cargar el bloque de programa en el dispositivo e iniciar la simulación.

Una vez inicializada la simulación, vamos a comprobar los diferentes casos forzando los bits en las entradas correspondientes:

- El interruptor ON pone en marcha el sistema y enciende la lámpara de funcionamiento.

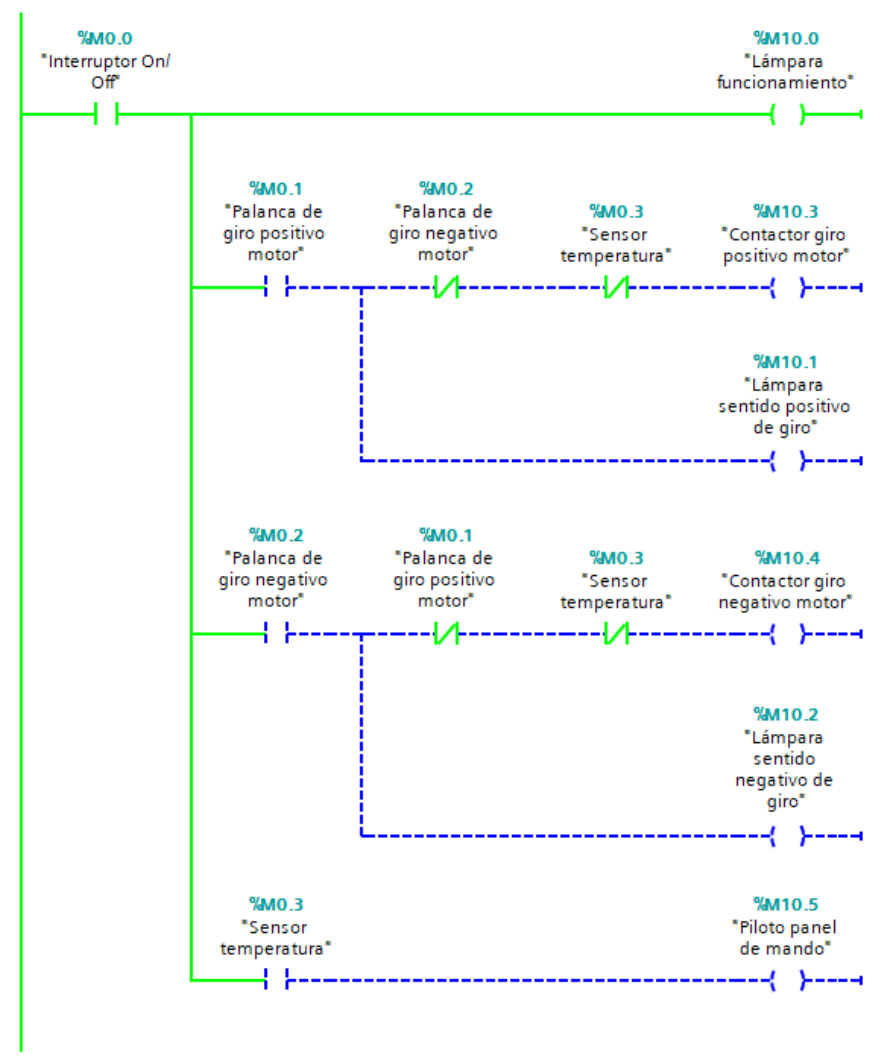


Figura 3: Sistema encendido y lámpara en funcionamiento



- Si el interruptor está en ON y se actúa sobre la palanca de giro positivo, el motor gira a derechas y se activa la lámpara indicativa de ese sentido de giro.

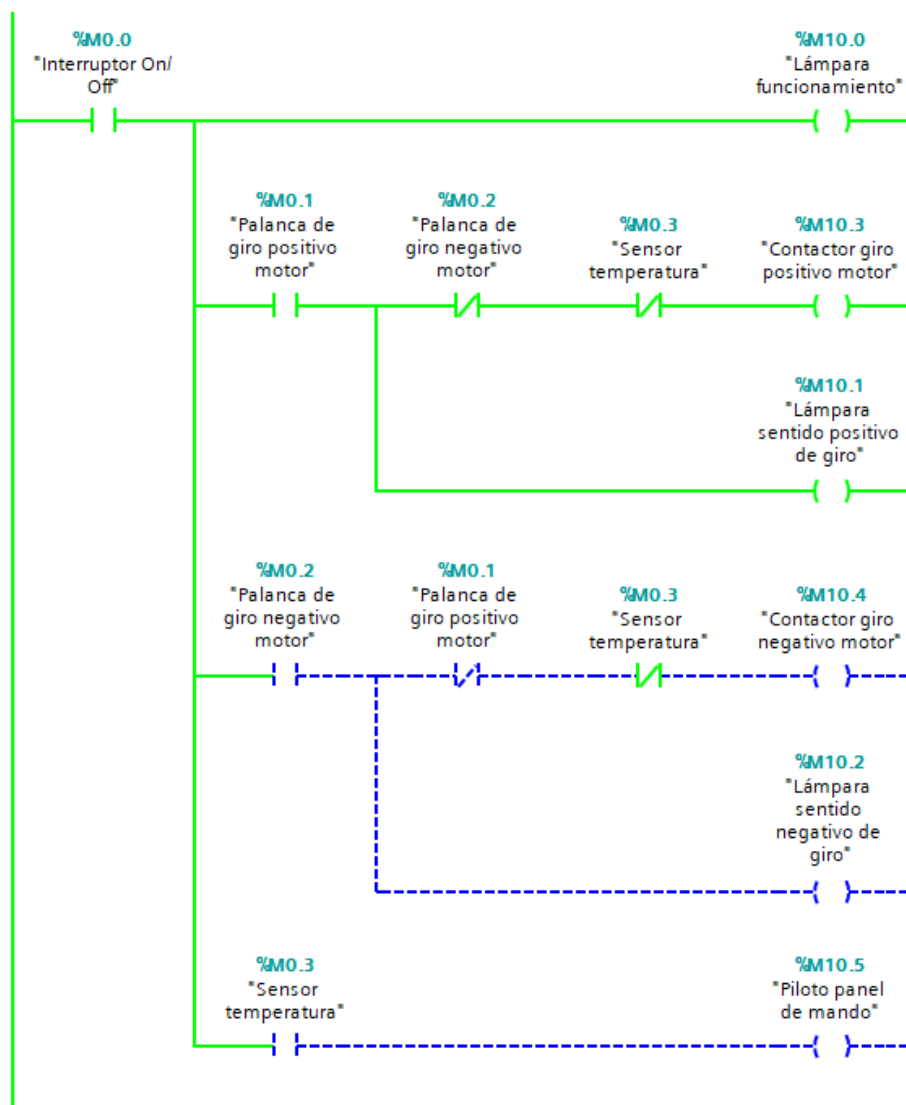


Figura 4: Motor gira a derechas y se activa la lámpara de ese sentido.



- Si el interruptor está en ON y se actúa sobre la palanca de giro negativo, el motor gira a izquierdas y se activa la lámpara indicativa de ese sentido de giro.

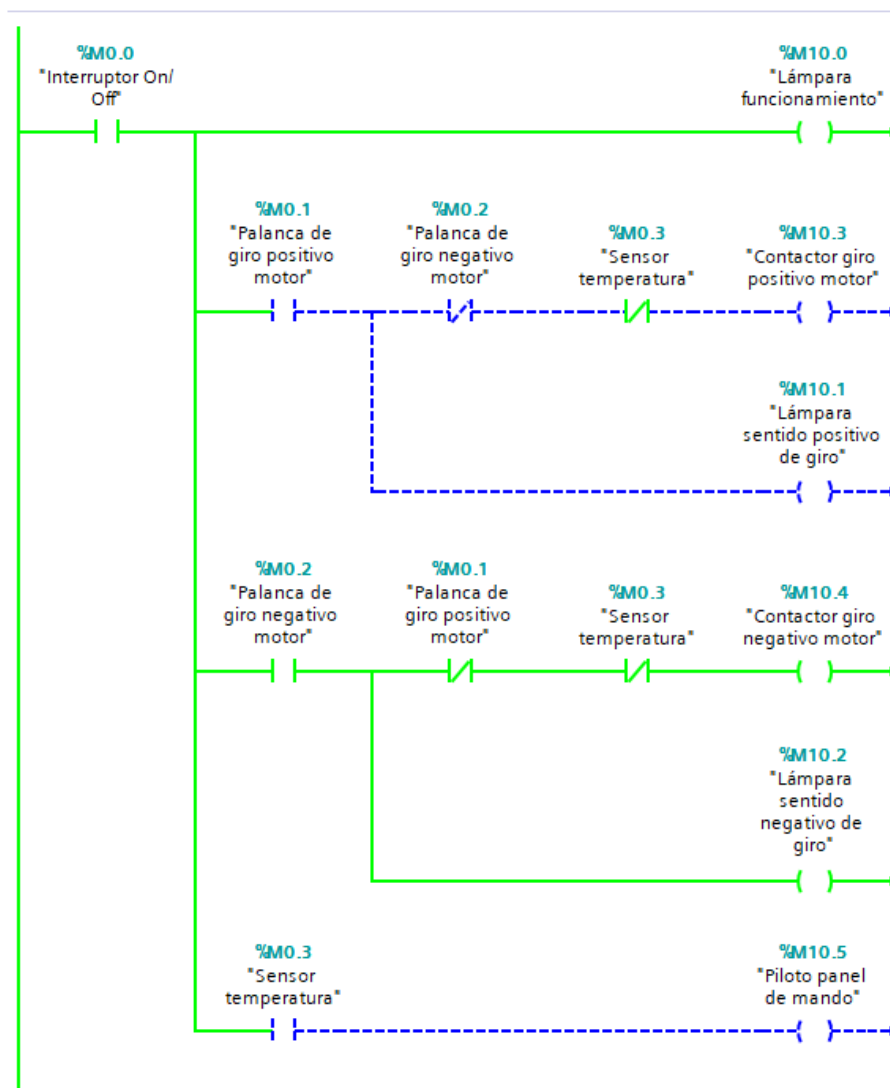


Figura 5: Motor gira a izquierdas y se activa la lámpara de ese sentido.



- Si, con el interruptor en ON, se actúa sobre ambas palancas, el motor no girará en ningún sentido, pero sí se encenderán las dos lámparas indicativas del sentido de giro.

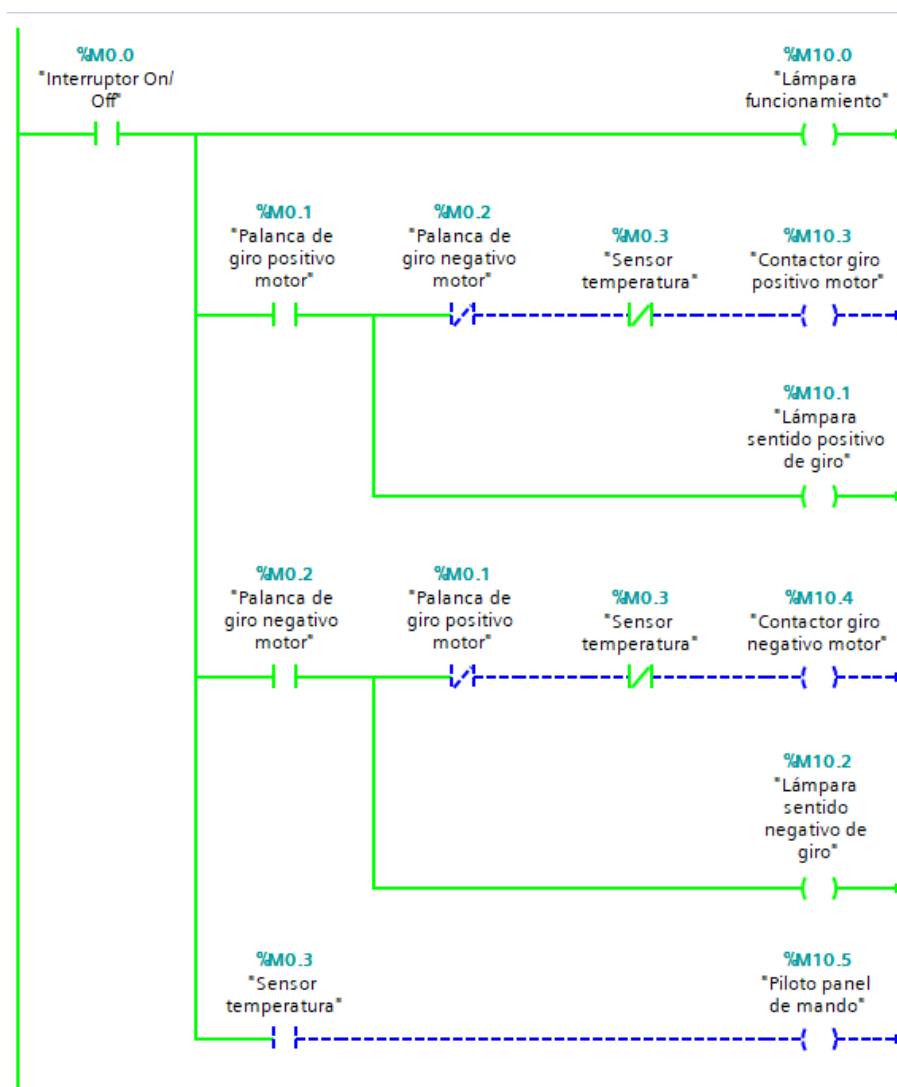
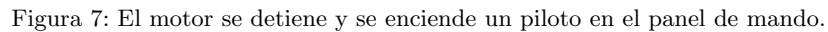


Figura 6: El motor no gira, pero sí se encienden las dos lámparas del sentido de giro.



- Se añade un sensor para medir la temperatura en el bobinado del motor. Cuando la temperatura sea excesiva el motor debe detenerse y se debe encender un piloto en el panel de mando.



- Figura 8: Se apaga el piloto y el motor continua en marcha.



## 2.4. Interfaz HMI

Finalmente, para realizar una monitorización y control del motor crearemos una interfaz HMI.

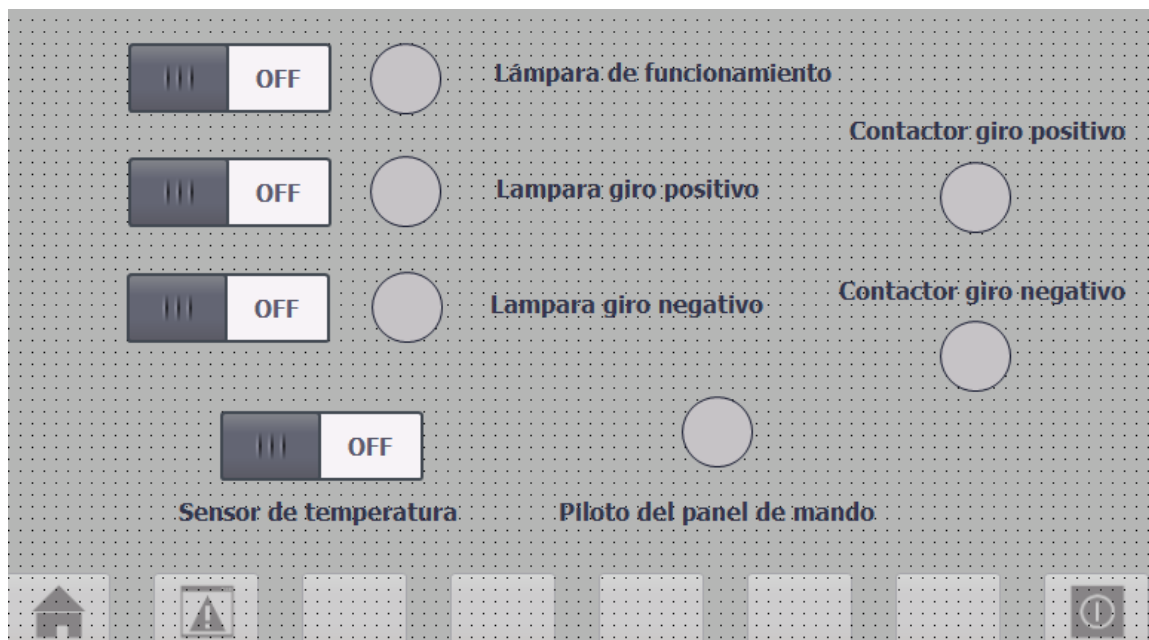


Figura 9: Interfaz HMI

Vídeo ejecución HMI: <https://youtu.be/SzhSmPs3Gnk>



### 3. Ejercicio 2: Control de un sistema de alarma de una vivienda

El funcionamiento del sistema debe ser el siguiente:

- Si el sistema no está activado, el LED (%M10.0) parpadea al estar abiertas la zona 1 (%M0.0) o 2 (%M0.1).
- Si el sistema está activado (girando la llave a la posición “on”, lo que activa la entrada %M0.2), el programa arranca un temporizador de retardo de 90 segundos para que el propietario pueda salir de la vivienda. Durante ese tiempo de retardo, el programa no reacciona si se abre alguna de las zonas (%M0.0 ó %M0.1).
- Si el sistema está activado y ha transcurrido el tiempo de retardo para salir de la vivienda, el programa evalúa el estado de ambas zonas. Si se abre alguna de ellas (%M0.0 ó %M0.1), el programa arranca una secuencia de notificación que activa la señal de alerta baja (%M10.2) y arranca un temporizador. Ello le recuerda al propietario que debe desactivar el sistema de alarma al regresar a casa.
- Una vez arrancada la secuencia de notificación, el programa tiene dos opciones:
  - Si se desactiva el sistema (girando la llave a la posición “off”, lo que desactiva %M0.2), el programa pone a “0” las salidas (%M10.0 y %M10.2) y los temporizadores.
  - Si el sistema no se desactiva al cabo de 60 segundos a más tardar, el programa dispara la alarma y activa la marcación automática del módem (%M10.1 y %M10.3).
- Si se activa la alarma manual (%M0.3), el programa dispara la alarma y activa la marcación automática del módem (%M10.1 y %M10.3). Esta tarea se realiza independientemente de la posición del interruptor que activa o desactiva el sistema de alarma (%M0.2) y no ejecuta la secuencia de notificación que ofrece un tiempo de retardo para desactivar el sistema.
- Si se desactiva el sistema (girando la llave a la posición “off”, lo que desactiva %M0.2) una vez disparada la alarma (%M10.1), el programa pone a “0” las salidas (%M10.1 y %M10.2) y los temporizadores
- Zona 1 (%M0.0): abierto = 0, cerrado = 1.
- Zona 2 (%M0.1): abierto = 0, cerrado = 1.
- Activa o desactiva el sistema de alarma (%M0.2): activa = 1, desactivado = 0.
- Activa manualmente la sirena de alarma (%M0.3): activa = 1, desactivado = 0.



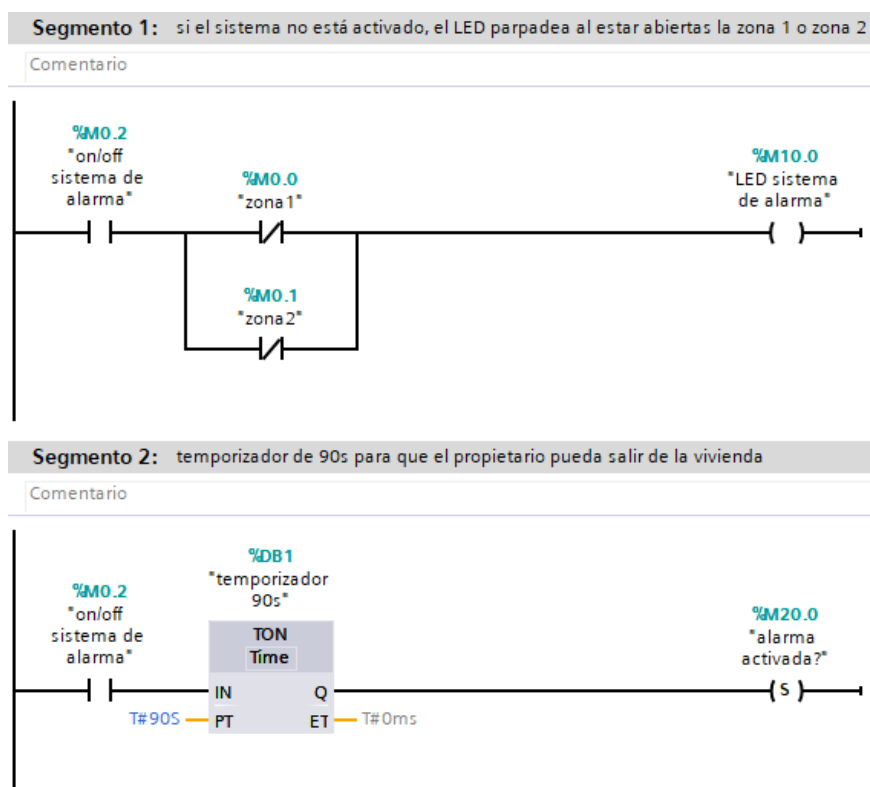
### 3.1. Tabla de variables

En primer lugar, generaremos las variables de entrada y salida siguiendo las indicaciones del enunciado:

Tabla de variables estándar				
	Nombre	Tipo de datos	Dirección	Comentario
1	zona1	Bool	%M0.0	vigila la zona 1 (abierto=0, cerrado=1)
2	zona2	Bool	%M0.1	vigila la zona 2 (abierto=0, cerrado=1)
3	on/off sistema de alarma	Bool	%M0.2	activa o desactiva la alarma
4	alarma manual	Bool	%M0.3	alarma manual
5	LED sistema de alarma	Bool	%M10.0	controla el LED del sistema (encendido=activado, parpadeante=desactivado)
6	Dispara sirena alarma	Bool	%M10.1	dispara la sirena de alarma
7	alerta baja	Bool	%M10.2	activa una señal de alerta baja, se disparará 60 segundos
8	relé activacion automatica	Bool	%M10.3	activa un relé de interface extern
9	alarma activada?	Bool	%M20.0	variable para detectar cuando se ha activado la alarma al salir de casa
10	alerta	Bool	%M20.1	variable para detectar si el usuario ha desactivado la alarma al regresar

Figura 10: Tabla de variables. Ejercicio 2

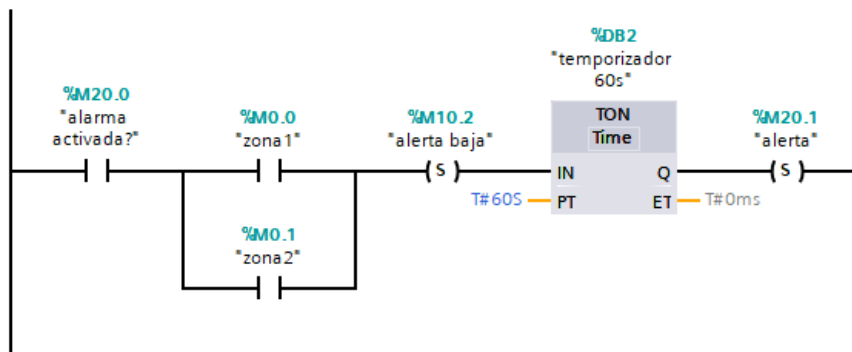
### 3.2. Programa en KOP





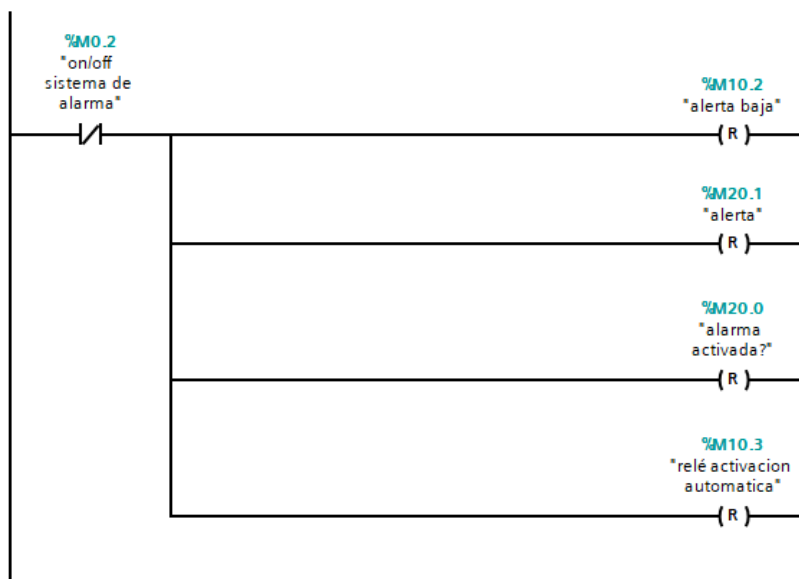
**Segmento 3:** activa una señal de alerta baja, se disparará 60 segundos

Comentario



**Segmento 4:** si se desactiva el sistema, reseteamos los temporizadores y las salidas correspondientes

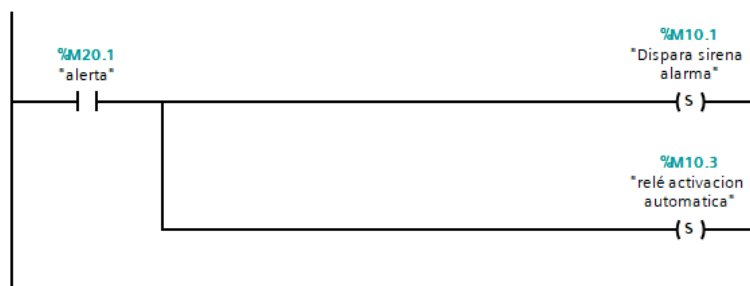
Comentario





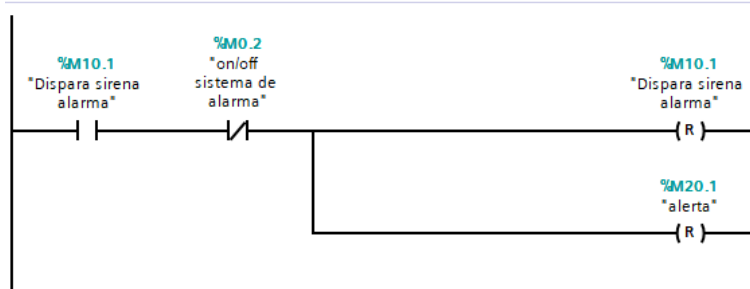
**Segmento 5:** si el sistema no se desactiva al cabo de 60s, el programa dispara la alarma y activa la marcación

Comentario



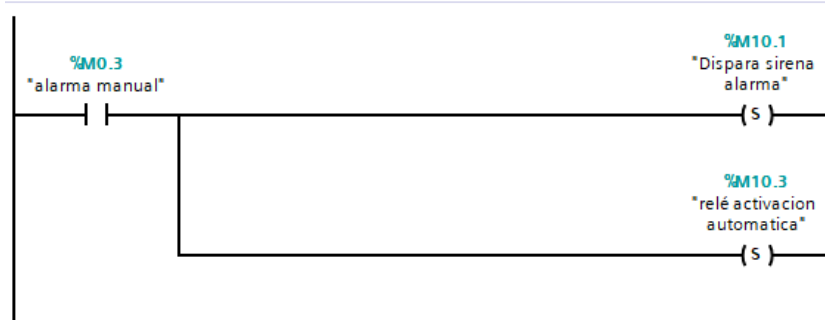
**Segmento 6:** dispara la sirena de alarma

Comentario



**Segmento 7:** si se activa la alarma manual, dispara la alarma y la marcación automática

Comentario





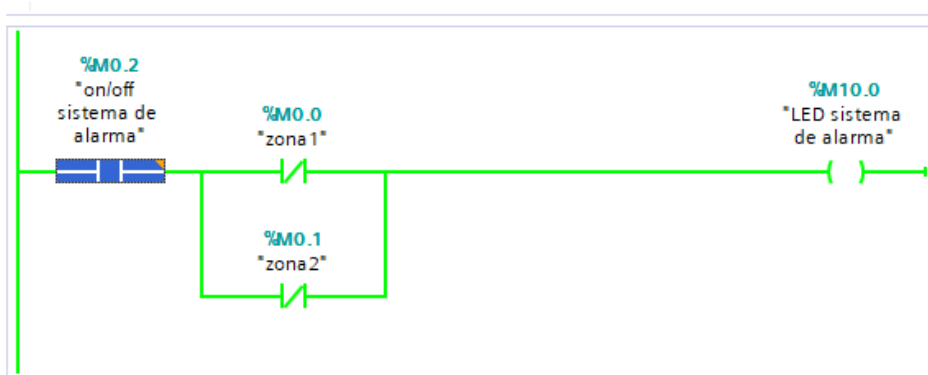
### 3.3. Ejecución

Antes de proceder con la ejecución del programa para comprobar su funcionamiento, debemos cargar el bloque de programa en el dispositivo e iniciar la simulación.

Una vez inicializada la simulación, vamos a comprobar los diferentes casos forzando los bits en las entradas correspondientes:

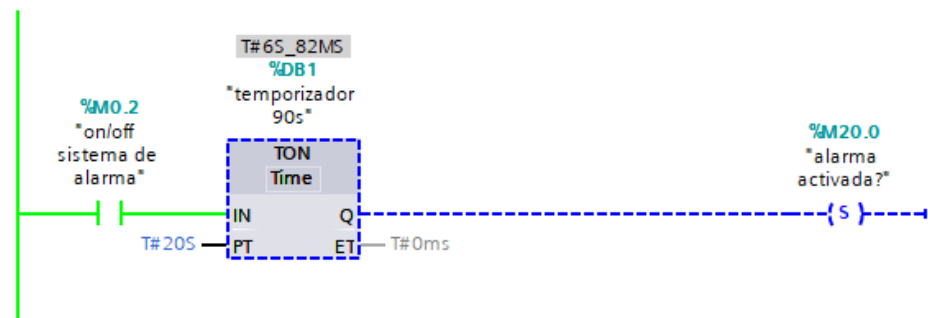
**NOTA:** para realizar la comprobación del funcionamiento del programa se han cambiado los tiempos de los temporizadores a un valor menor para agilizar la ejecución.

- Sistema de alarma activado, LED del sistema de alarma encendido. La alarma no se activará hasta pasar el tiempo del temporizador.



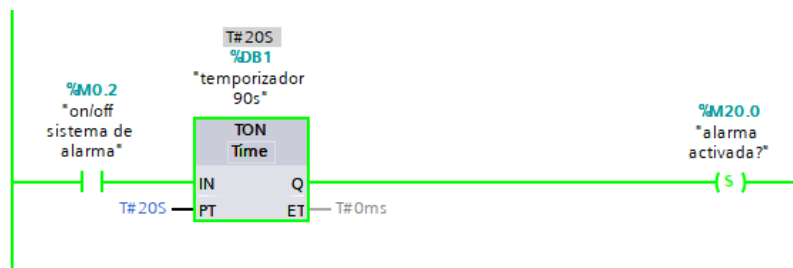
#### ▼ Segmento 2: temporizador de 90s para que el propietario pueda salir de la vivienda

Comentario



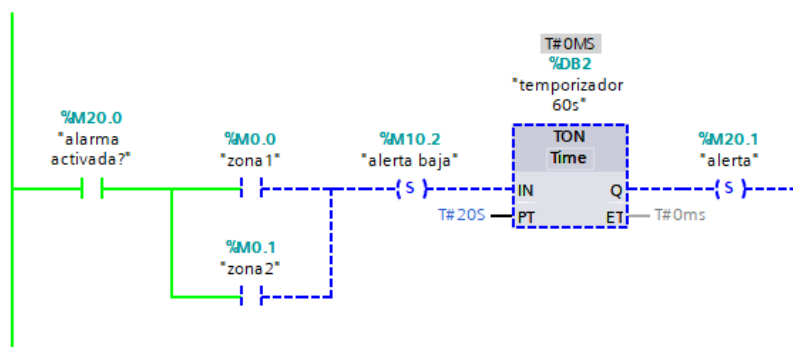


- Una vez pasado el tiempo, la alarma se activa. Para ello, se ha creado una variable auxiliar para saber cuando la alarma se encuentra activada y así poder proceder con las siguientes funcionalidades. En este caso como se puede observar, las zonas están cerradas y por tanto, la alerta baja no se activa.

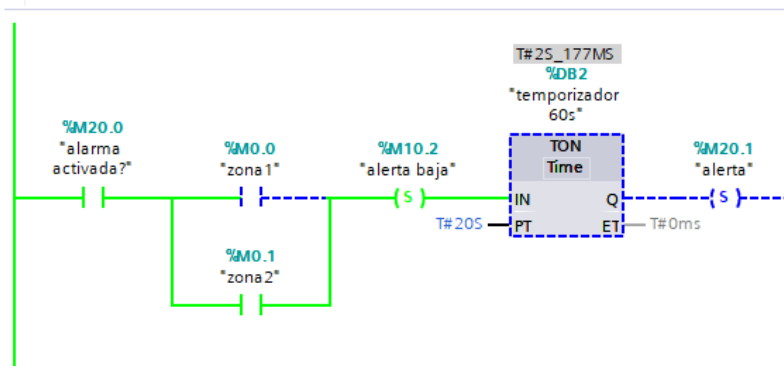


**Segmento 3:** activa una señal de alerta baja, se disparará 60 segundos

Comentario



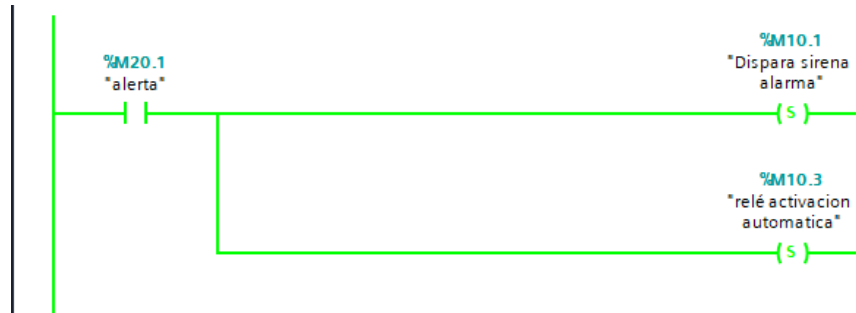
- Al abrirse una de las zonas, el programa arranca una secuencia de notificación y activa la señal de alerta baja con un temporizador de 60s y una vez finalizado se activa la alerta baja.



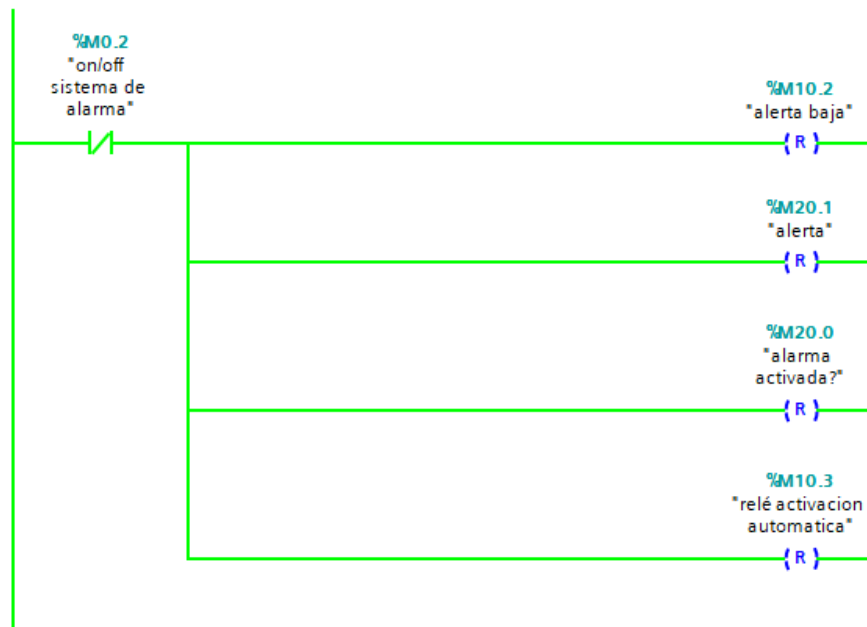




- Si el sistema no se desactiva al cabo de 60 segundos, el programa dispara la alarma y activa la marcación automática del módem. Para ello, se ha utilizado una variable auxiliar alerta para indicar cuando ocurre tal hecho.



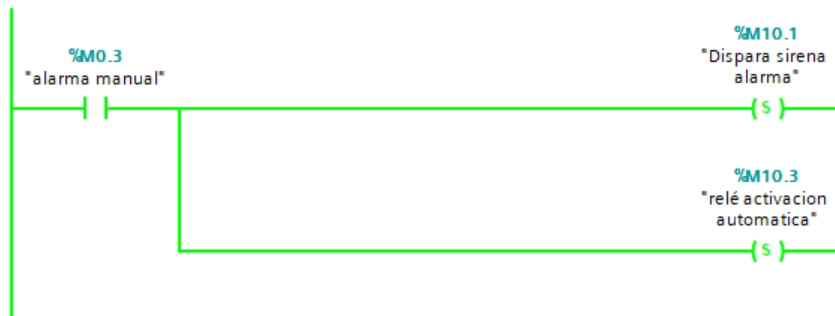
- Si se desactiva el sistema, el programa apaga la señal de alerta baja, alarma y los temporizadores.





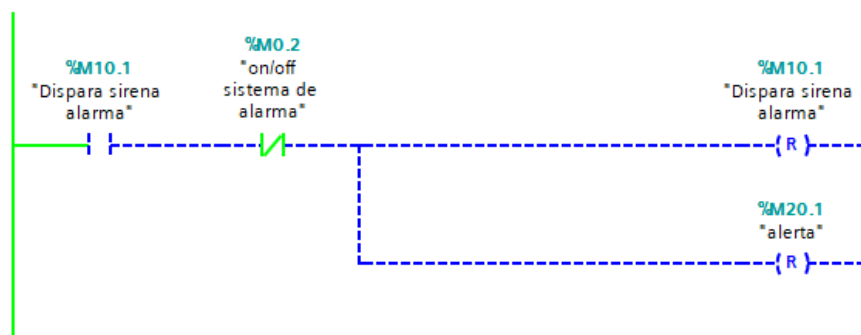
- Si se activa la alarma manual, el programa dispara la alarma y activa la marcación automática del módem independientemente de la posición del interruptor M0.2.

Comentario



Segmento 8: .....

- Si se desactiva el sistema (girando la llave a la posición “off” una vez disparada la alarma el programa pone a 0 la alarma disparada y la alerta.





### 3.4. Interfaz HMI

Finalmente, para realizar una monitorización y control del sistema crearemos una interfaz HMI.

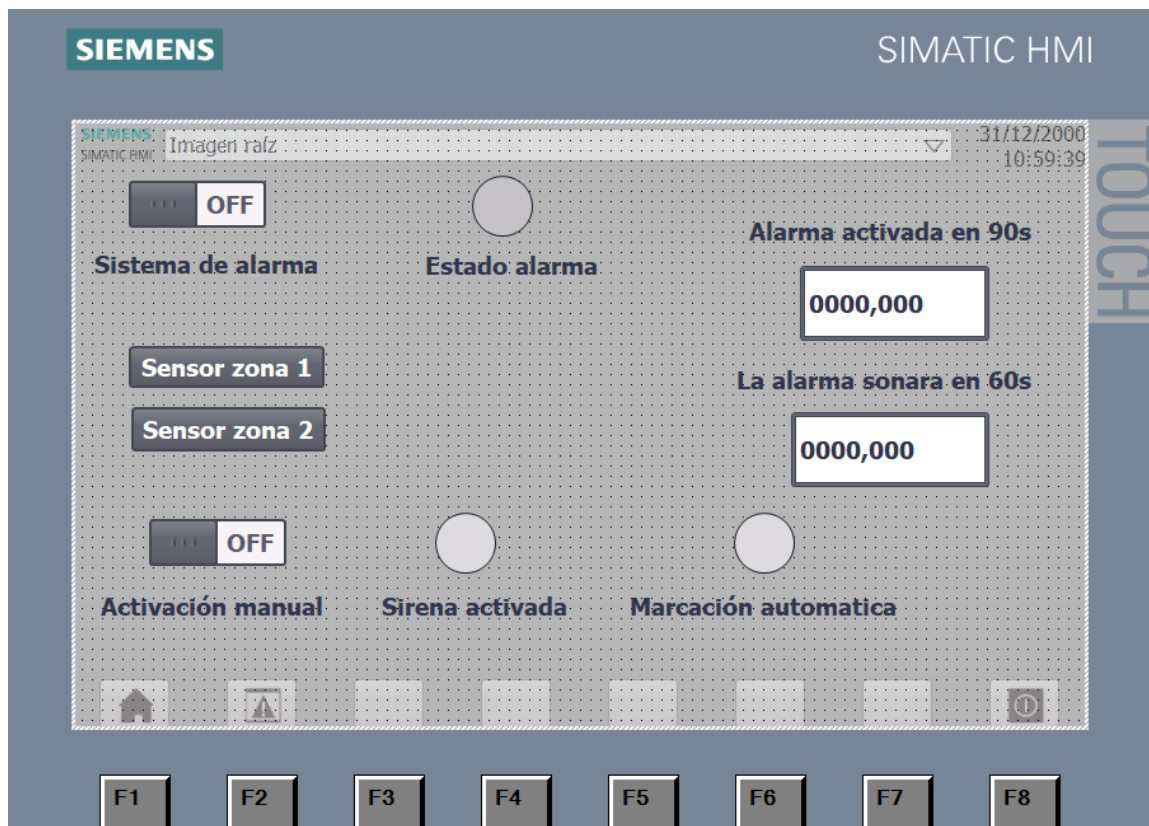


Figura 11: Interfaz HMI - Ejercicio 2

Vídeo ejecución HMI: <https://youtu.be/pcrfcUT39i0>



#### 4. Ejercicio 3: Control de un desapilador automático.

El funcionamiento del desapilador automático debe ser el siguiente:

A la maquina entran pilas de pales y los desapila de forma automática entregándolos de uno en uno. Tenemos un pulsador de marcha uno de paro para el sistema. Si se pulsa el paro la secuencia se detiene continuando en el mismo punto cuando se pulse marcha de nuevo.

El motor M1 está en marcha hasta que detectamos que la pila alcanza la fotocélula 1. La pila permanecerá en esta posición con el motor M1 parado hasta que el desapilador quede vacío, dejando pasar la pila.

La secuencia del desapilador es la siguiente. Entra una pila con el freno C3 arriba y el motor M2 activo hasta que es detectada en 2. En ese momento se para el motor M2. Activamos el cilindro C2 que sujeta las paletas que se encuentran sobre la que queremos extraer. Una vez extendido C2, C1 sube generando el hueco necesario en la pila para que solo salga un palé. En este punto el desapilador espera a tener permiso para extraer palé.

El permiso para extraer palé lo da el no detectar palé en 3 durante más de tres segundos. M3 está en marcha siempre que la maquina lo esté (marcha/paro).

Cuando el desapilador tiene permiso para extraer el palé baja el freno C3 activa el motor M2 hasta que 2 no detecte paleta mas de 3 segundos. En ese momento entendemos que el palé ha salido y bajamos C1 hasta depositar la pila en el camino de rodillos (4) y entonces desactivamos C2. Si tras desactivar C2 no detectamos paleta en 2 entendemos que el desapilador está vacío por lo que se reinicia el ciclo con una nueva entrada de pales.

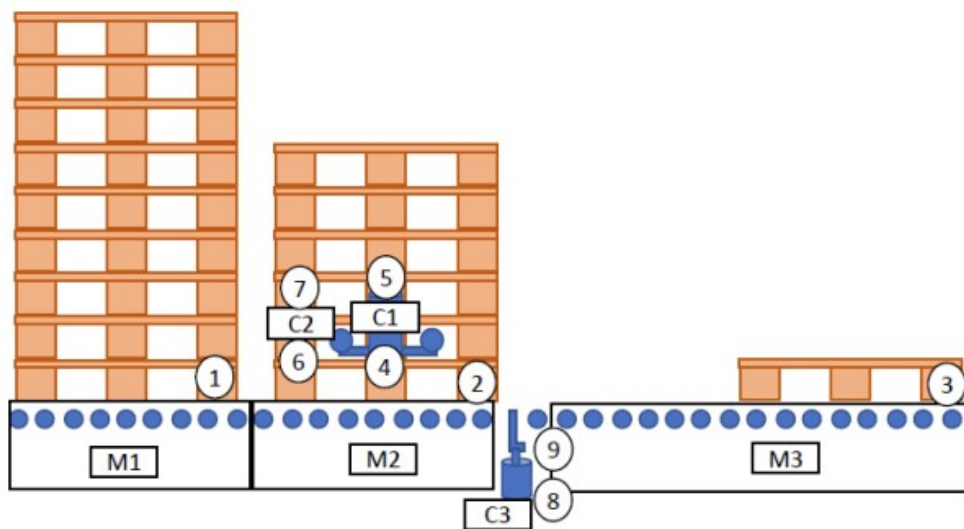


Figura 12: Imagen desapilador automático



Para la realización del ejercicio, se ha creado un grafo de estados:

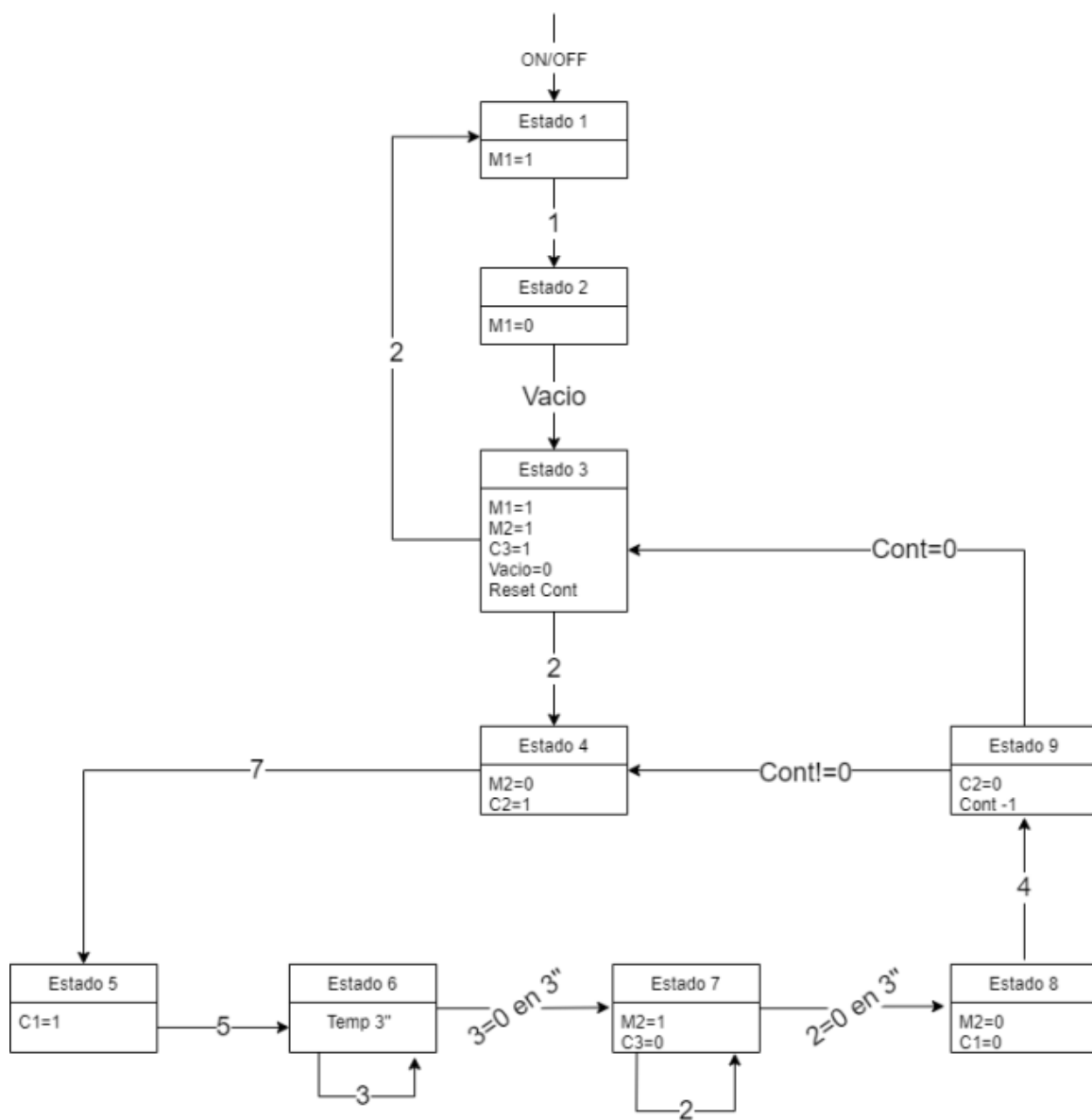


Figura 13: Grafo de estados - Ejercicio 3



#### 4.1. Tabla de variables

En primer lugar, generaremos las variables de entrada y salida siguiendo la tabla del enunciado:





















Tabla de variables estándar					
		Nombre	Tipo de datos	Comentario	Dirección ▲
1		Pulsador de marcha	Bool		%M0.0
2		Pulsador de paro	Bool		%M0.1
3		1.- Pila de pales en la entrada	Bool		%M0.2
4		2.- Pila de pales en desapilador	Bool		%M0.3
5		3.- Palé en rodillos de salida	Bool		%M0.4
6		4.- Cilindro subir pila pales reposo	Bool		%M0.5
7		5.- Cilindro subir pila pales arriba	Bool		%M0.6
8		6.- Cilindro uña sujetar pila reposo	Bool		%M0.7
9		7.- Cilindro uña sujetar pila extendida	Bool		%M1.0
10		8.- Freno pila recogido	Bool		%M1.1
11		9.- Freno pila extendido	Bool		%M1.2
12		M1.- Motor rodillos entrada pilas	Bool		%M10.0
13		M2.- Motor rodillos desapilador	Bool		%M10.1
14		M3.- Motor rodillos salida de palé	Bool		%M10.2
15		C1.- Cilindro subir pila	Bool		%M10.3
16		C2.- Cilindro uñas sujetar pila	Bool		%M10.4
17		C3.- Cilindro freno	Bool		%M10.5
18		Desapilador vacío	Bool	Controla si el desapilador esta vacío	%M20.0
19		on/off	Bool	Variable de encendido	%M20.1
20		Tag_1	DWord	Almacenamos el numero de palets	%MD100

Figura 14: Tabla de variables. Ejercicio 3

Además de las entradas y salidas del enunciado, se ha creado otra tabla correspondiente a los estados que han sido obtenidos a partir del grafo.

Tabla de variables estados					
		Nombre	Tipo de datos	Comentario	Dirección
1		q1	Bool	LLevar palets	%M20.2
2		q2	Bool	Esperar pila	%M20.3
3		q3	Bool	Rellenar desapilador	%M20.4
4		q4	Bool	Inicio de la desapilacion	%M20.5
5		q5	Bool	Levantamos pila	%M20.6
6		q6	Bool	Esperamos a desapilar	%M20.7
7		q7	Bool	Sacamos palet	%M21.0
8		q8	Bool	Bajamos pila	%M21.1
9		q9	Bool	Miramos si quedan palets	%M21.2
10		resetCount	Bool	Resetea el contador	%M21.3
11		SinPalets	Bool	Indica que ya no quedan palets	%M21.4

Figura 15: Tabla de variables de estado. Ejercicio 3



## 4.2. Programa en KOP y Ejecución

Antes de proceder con la ejecución del programa para comprobar su funcionamiento, debemos cargar el bloque de programa en el dispositivo e iniciar la simulación.

Una vez inicializada la simulación, para comprobar los diferentes casos, se ha creado un proyecto del PLC para ir cambiando los bits de una manera más cómoda ya que este ejercicio supone un esfuerzo mayor. Para ello, se han cargado las variables del proyecto en la tabla del PLCSIM y se ha activado el botón marcado en la imagen siguiente para poder forzar las variables ya que éstas corresponden a áreas de memoria.

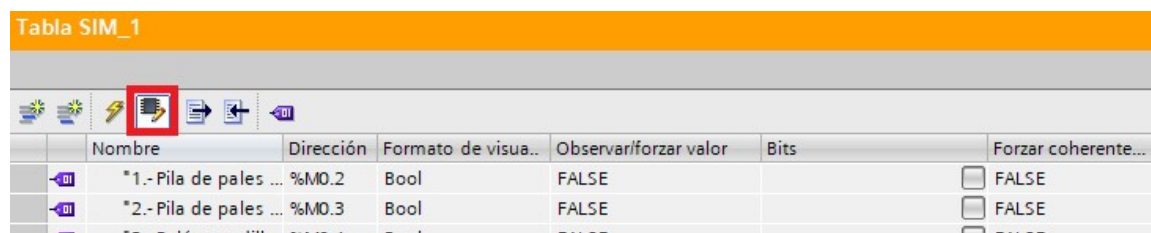


Tabla SIM_1						
	Nombre	Dirección	Formato de visua..	Observar/forzar valor	Bits	Forzar coherente...
	*1.- Pila de pales ...	%M0.2	Bool	FALSE		<input type="checkbox"/> FALSE
	*2.- Pila de pales ...	%M0.3	Bool	FALSE		<input type="checkbox"/> FALSE
	*3.- Pila de pales ...	%M0.4	Bool	FALSE		<input type="checkbox"/> FALSE

Figura 16: Tabla SIM, activar botón forzado en áreas de memoria



- El primer segmento controla la puesta en marcha y paro del sistema de tal forma que cuando se active la marcha se encienda el sistema y cuando se pulse el botón de paro se reinicie el pulsador de marcha apagando el sistema. El motor 3 siempre está activado al activarse el sistema.

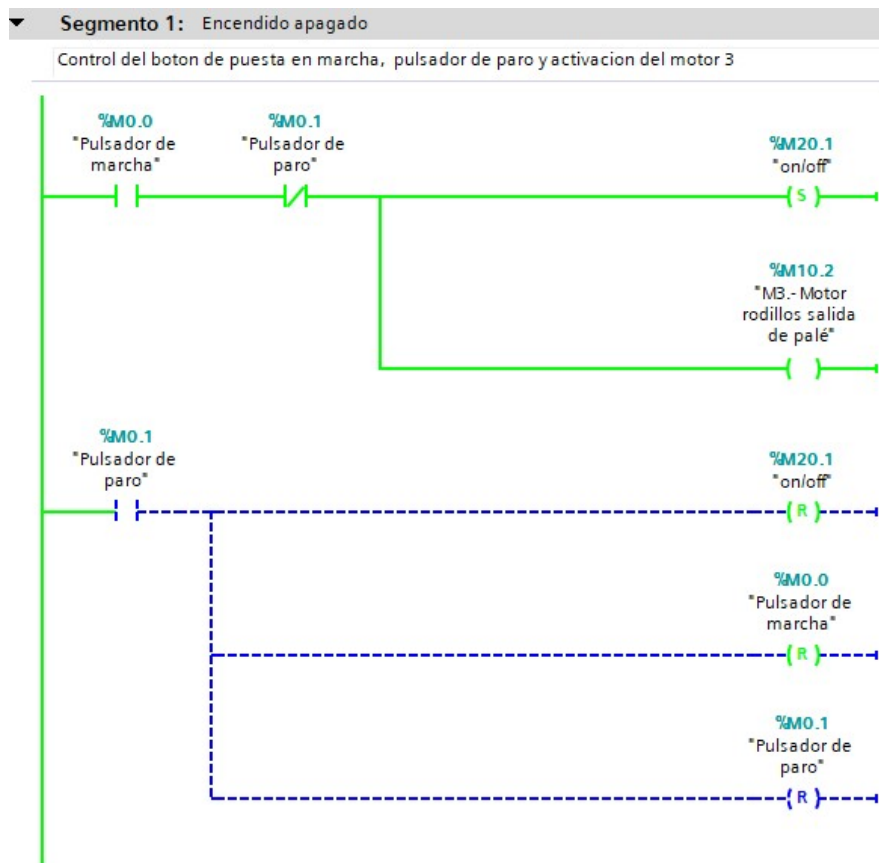


Figura 17: Segmento 1





- Este segmento controla el cambio del estado 0 al estado 1 al pulsar el botón de encendido. También, se controla el estado 1 que representaría cuando el desapilador llena la cola de alimentación. Finalmente cuando se activa la entrada 1, es decir, se ha detectado presencia en la fotocélula 1, cambia al estado 2.

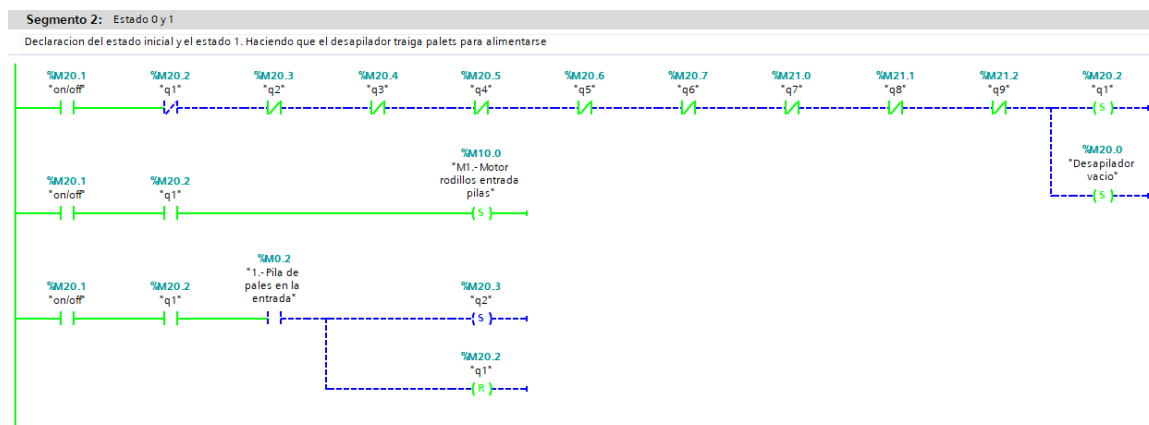


Figura 18: Segmento 2



- El segmento 3 controla la programación del estado 2 la cual se queda a la espera a que el desapilador indique que esta vacío cambiando al estado 3.

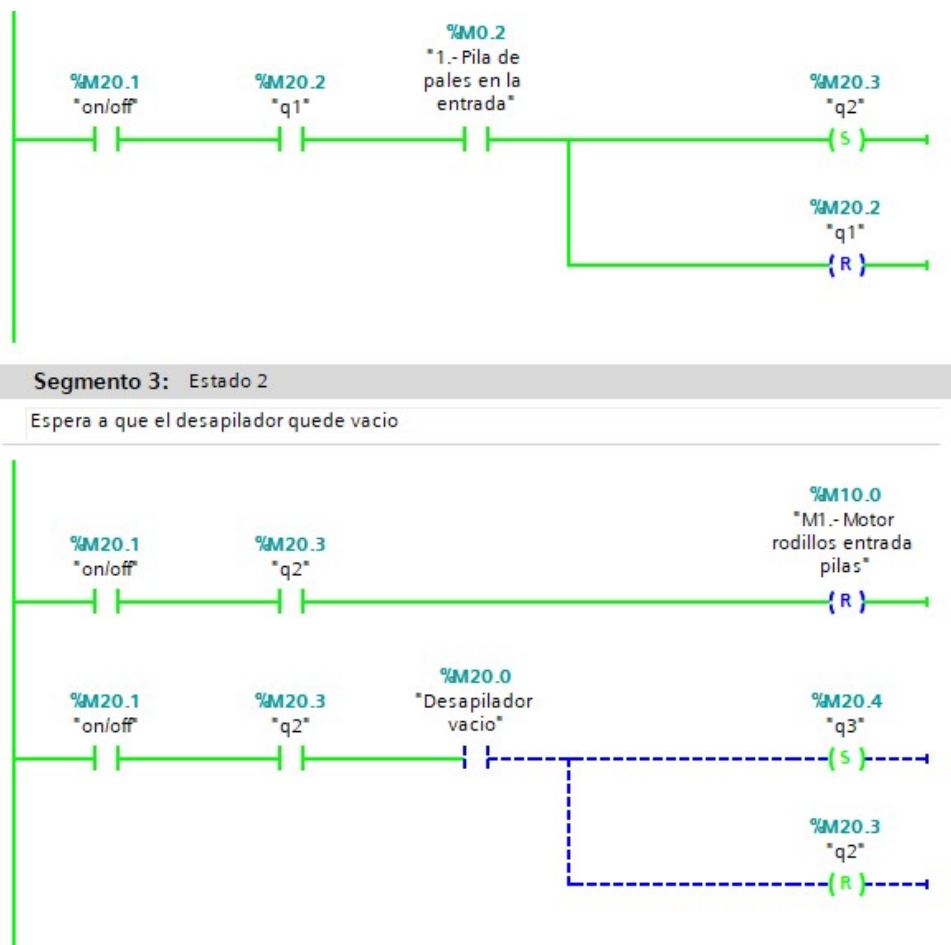


Figura 19: Segmento 3



- Controlamos el estado 3 donde representamos el llenando de palets y cuando detectamos la fotocélula 2 reseteamos el contador de palets y llamamos al estado 1 para que siga a la espera para seguir alimentando los palets y pasamos al estado 4 donde comenzaremos a desapilar.

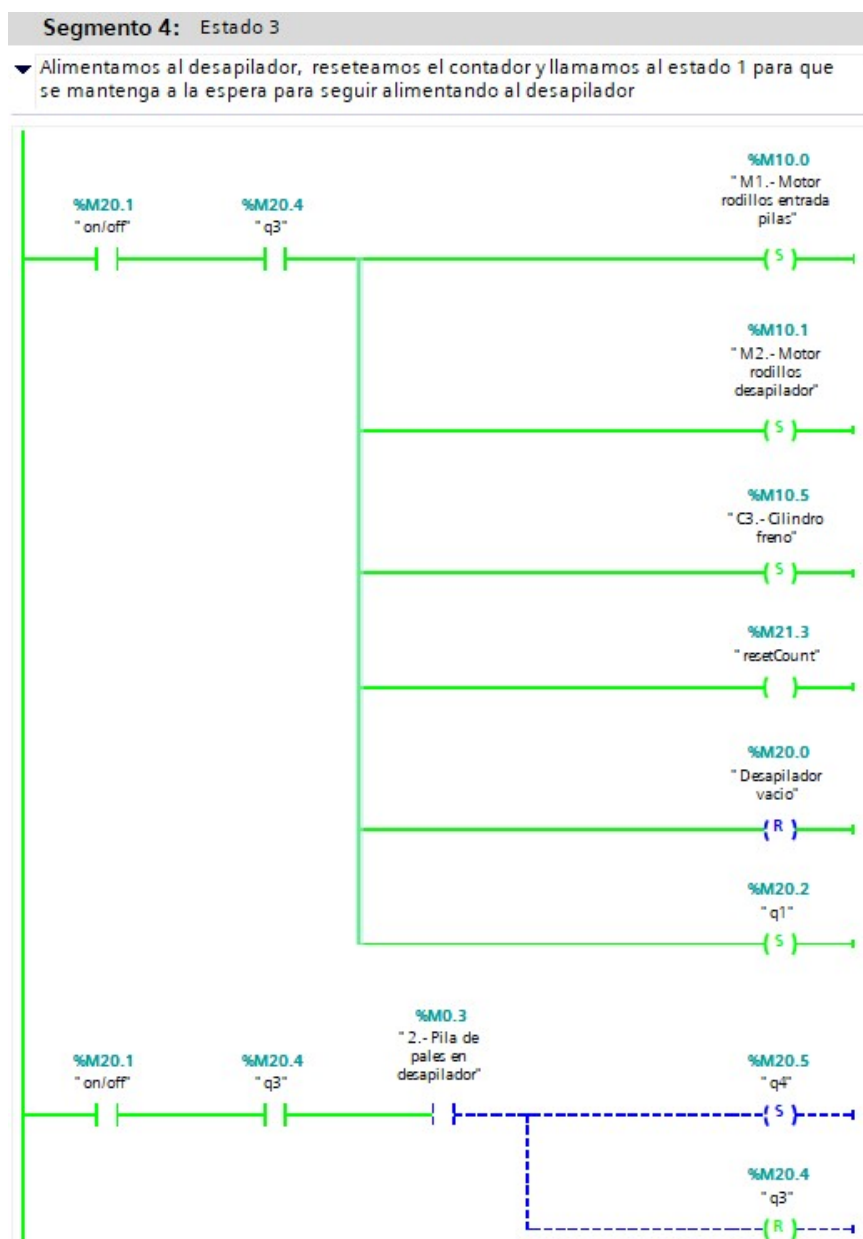
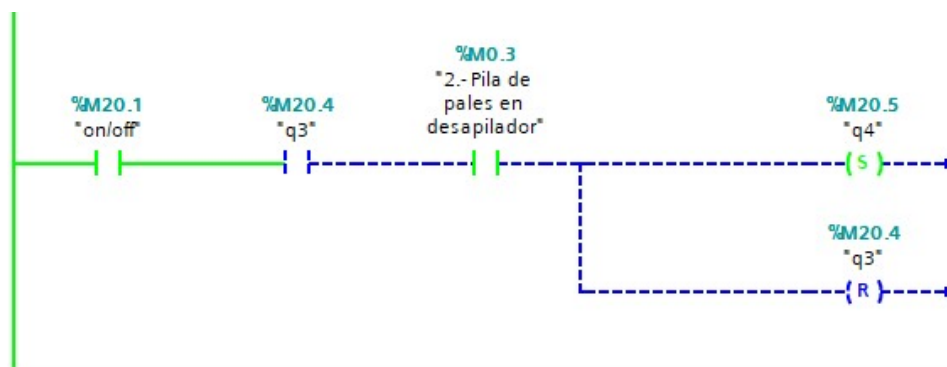


Figura 20: Segmento 4



- En el estado 4 accionamos la uña hasta que detectamos que esta extendida.



#### Segmento 5: Estado 4

- ▼ Comenzamos con la secuencia de desapilamiento.  
Paso 1: Accionamos la uña

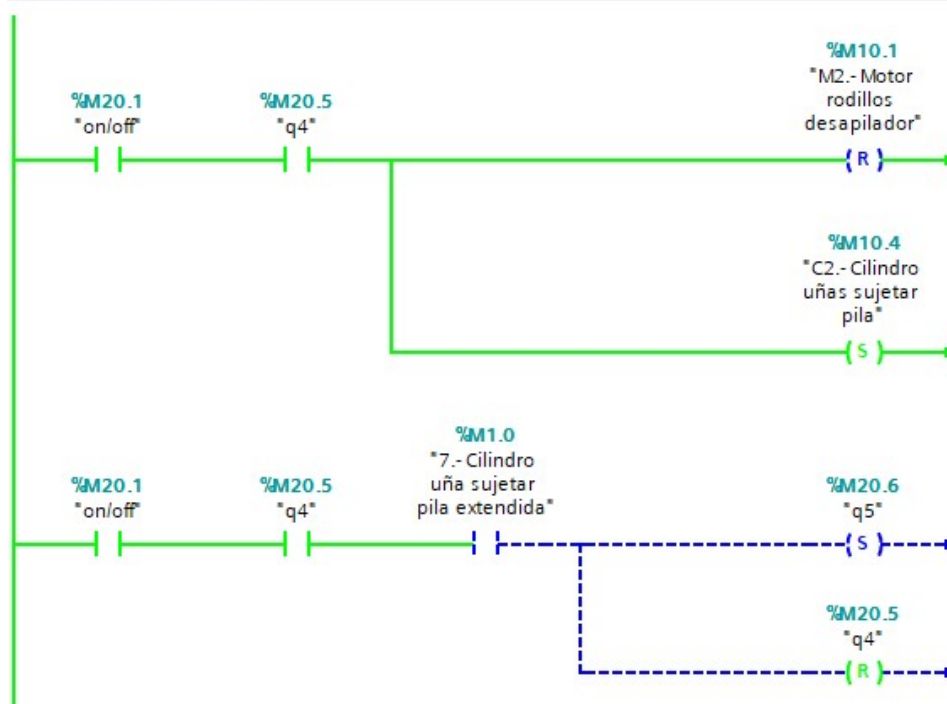
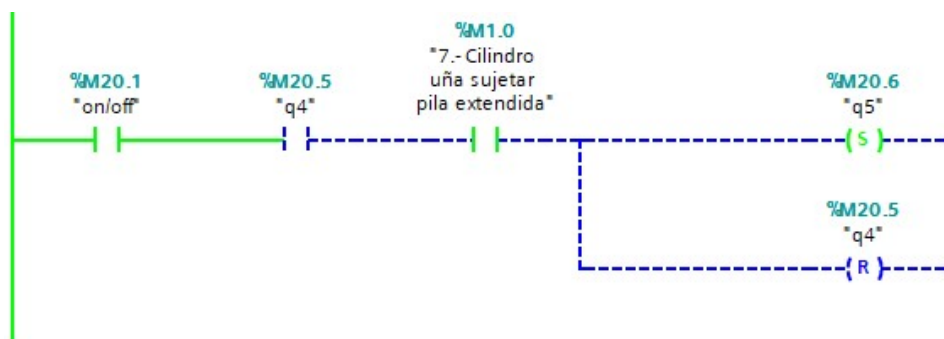


Figura 21: Segmento 5



- En el estado 6 levantamos los palets.



#### Segmento 6: Estado 5

▼ Desapilamiento  
Paso 2: levantamos la pila

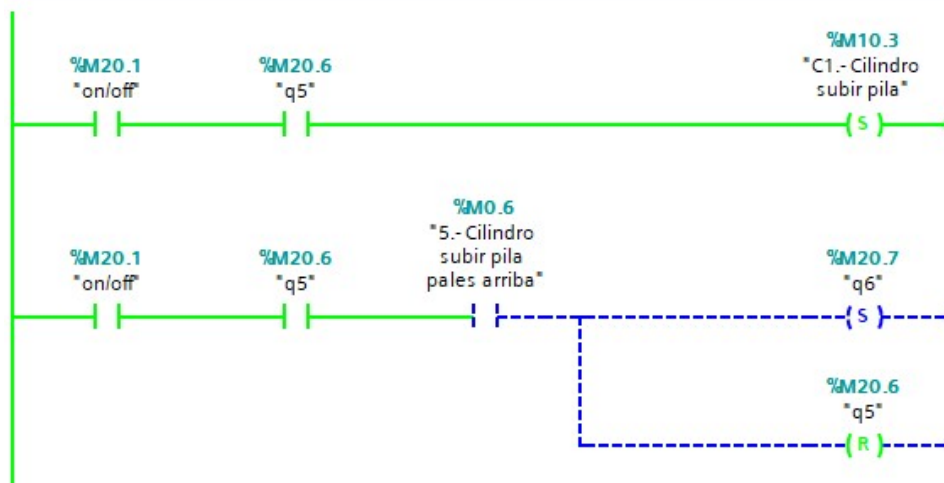


Figura 22: Segmento 6



- En el estado 7 esperamos 3 segundos a no detectar ningún palet en la fotocélula 3 cambiando al estado 8 donde se retira el palet.

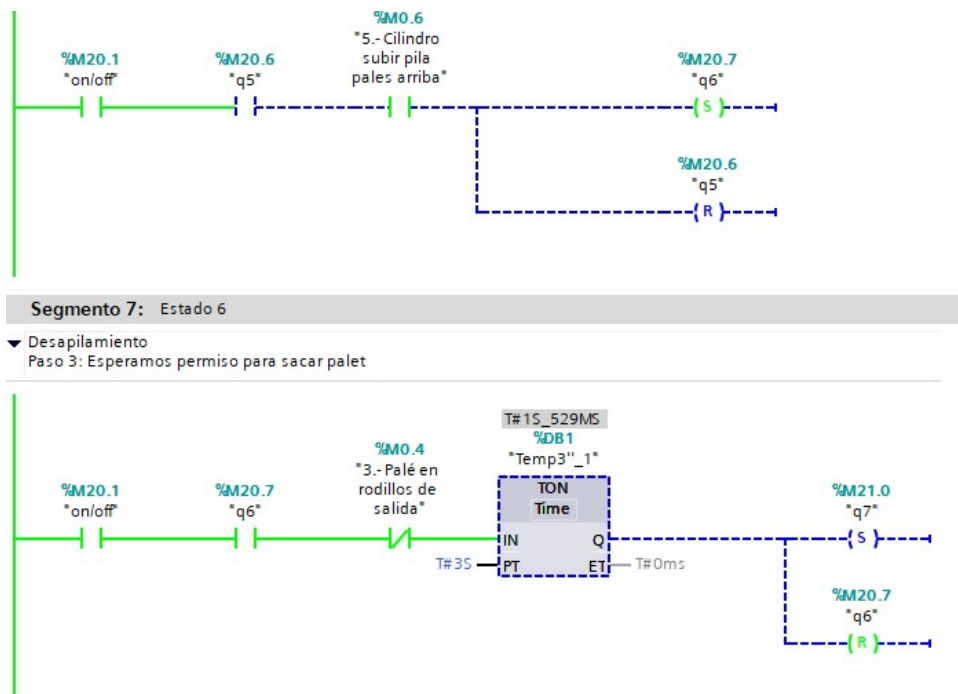
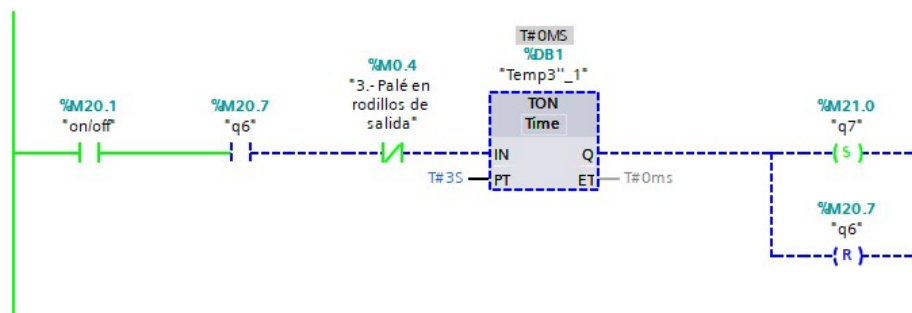


Figura 23: Segmento 7



- En el estado 8 retiramos el palet y esperamos 3 segundos sin detectar palet para asegurarnos que se ha retirado.



#### Segmento 8: Estado 7

▼ Desapilamiento  
Paso 4: Extraemos palet

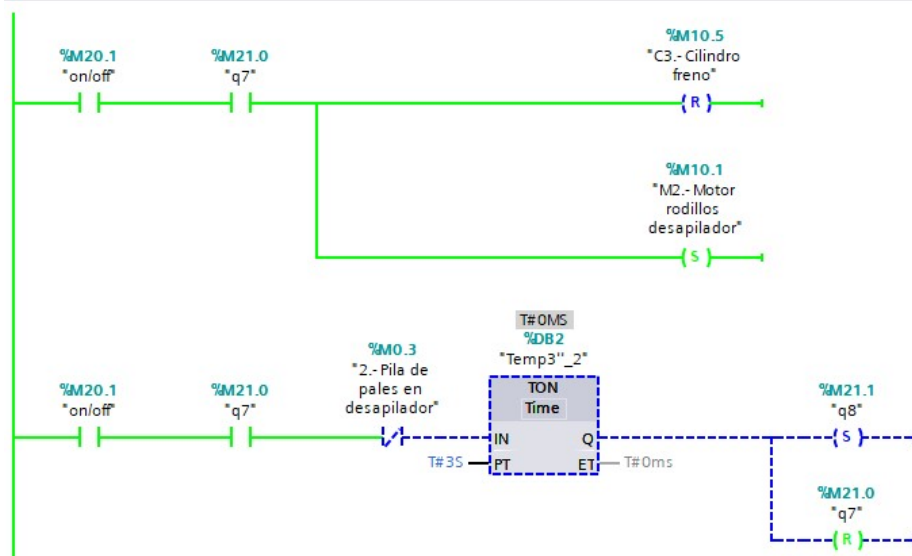


Figura 24: Segmento 8



- En el estado 9 bajamos la pila.

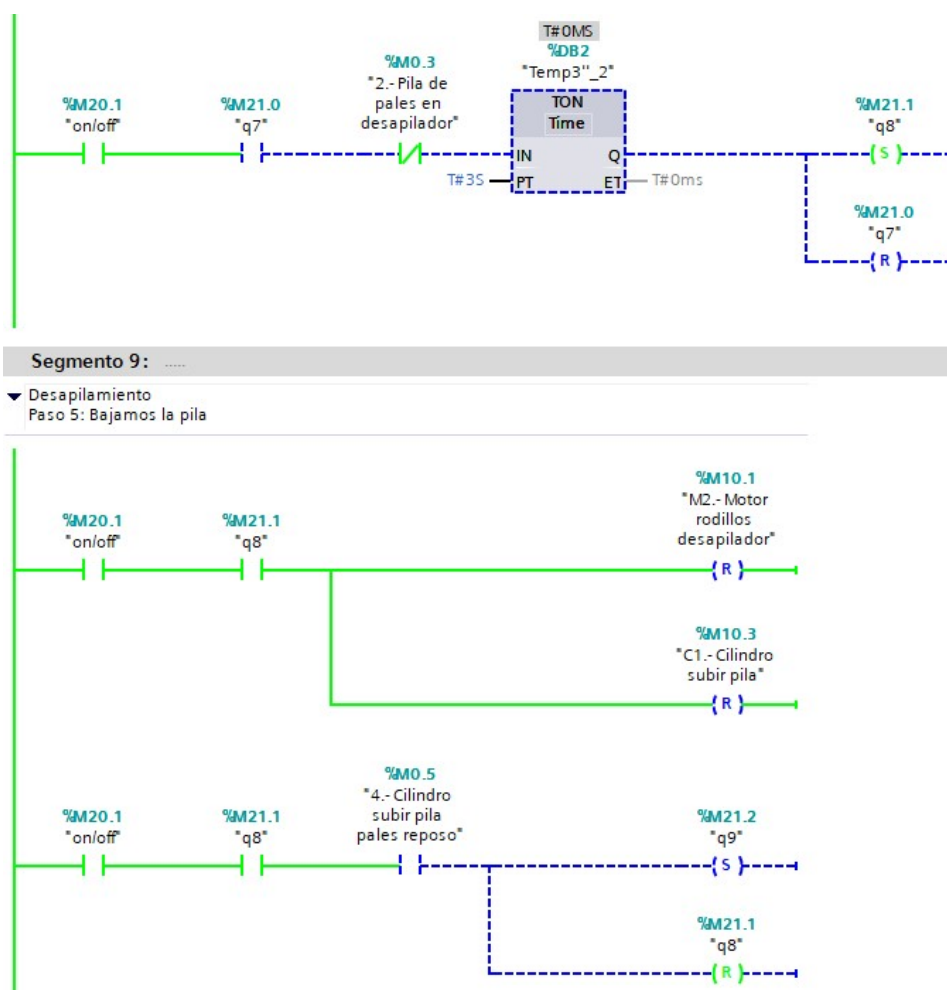
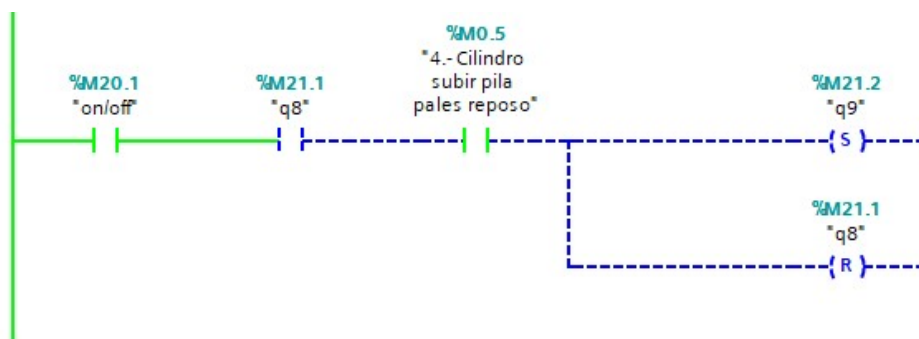


Figura 25: Segmento 9





- En el segmento 10 el estado 9 comprueba si nos quedan mas palets, si quedan mas resta 1 al contador y volvemos al estado 4 para seguir desapilando. Si el contador llega a 0 entonces se reinicia todo para volver al estado 3.

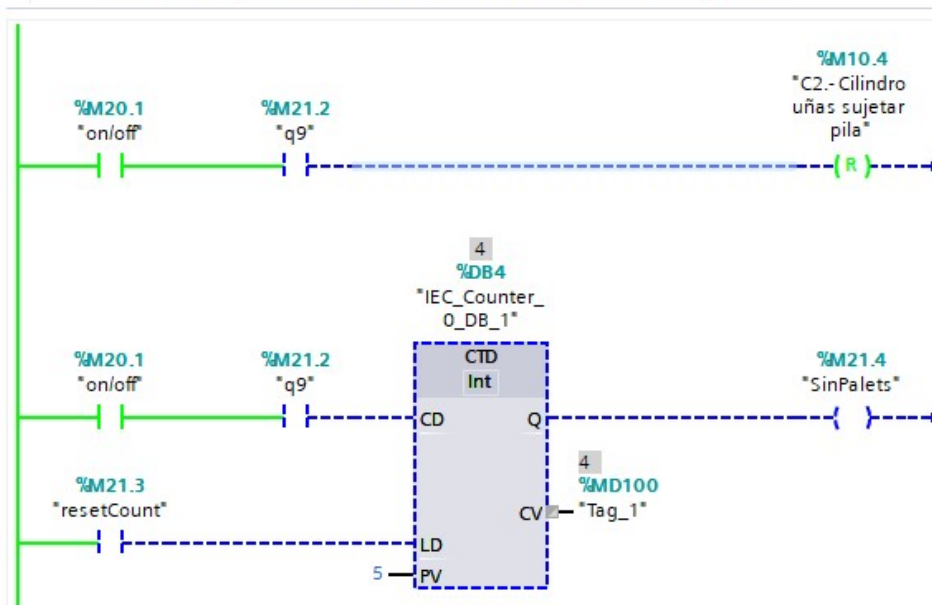


#### Segmento 10: Estado 9

##### ▼ Ultimo paso del desapilamiento

Paso 6: Si no quedan palets solicitamos mas palets llamando al estado 3

De lo contrario, volveriamos al estado 4 comenzando de nuevo el proceso.



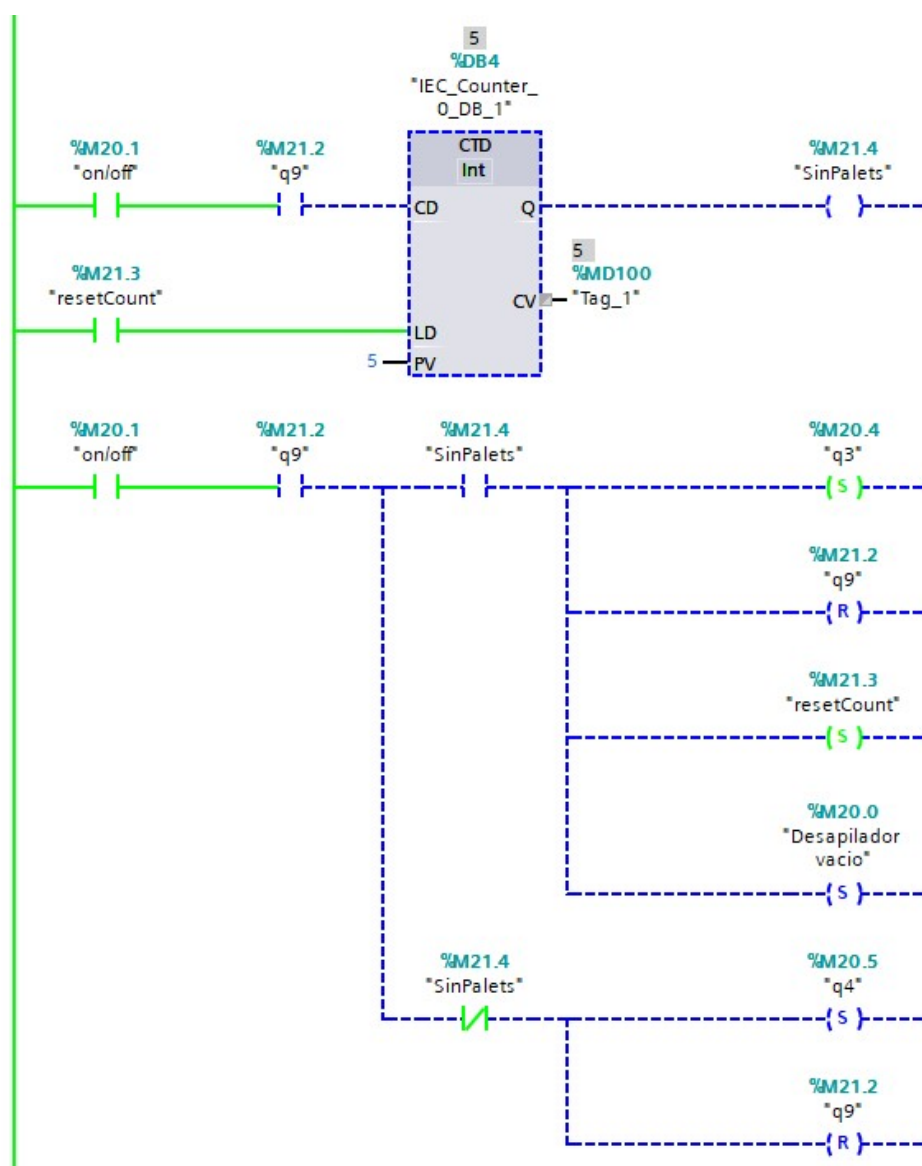


Figura 26: Segmento 10



### 4.3. Interfaz HMI

Finalmente, para realizar una monitorización y control del desapilador crearemos una interfaz HMI:

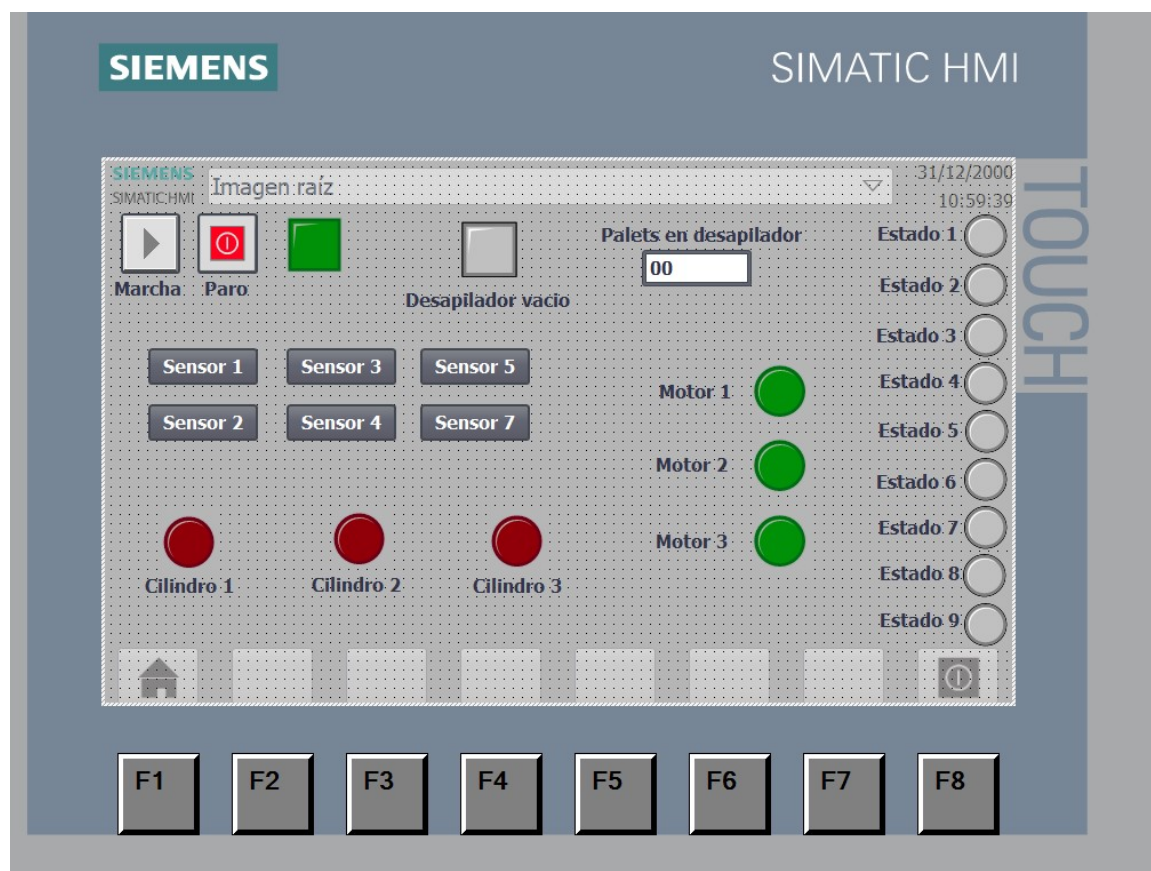


Figura 27: Interfaz HMI - Ejercicio 3

Vídeo ejecución HMI: <https://youtu.be/igrACj0JX7M>