



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Automatización de un sistema de llenado de barriles

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jacobo Martínez Crespín

**Tutor:** Enrique Jorge Bernabeu Soler

2013/2014



Me gustaría agradecer a todas las personas que han contribuido directa o indirectamente a la realización de este trabajo.

Sobre todo agradecer el apoyo por parte de mi familia tanto la que reside en Valencia como la que se encuentra en Sevilla, en especial a mi madre y a mi pareja.

También quiero agradecer a mi director del proyecto Enrique Jorge Bernabeu los consejos y ayudas.

Gracias a todos.



# Resumen

---

El proyecto nace de la idea de integrar el conocimiento adquirido como ingeniero informático, en el sector de la industria, concretamente en el campo de la automática.

El proyecto consiste en automatizar una pequeña planta industrial de embotellado, mediante *PLC (Programmable Logic Controller)*. Para llevarlo a cabo, hemos utilizado una herramienta software, llamada *PL7 Micro*, centrada en el diseño lógico de autómatas, en nuestro caso el diseño se ha realizado mediante un *GRAFCET (GRAPhe Fonctionel de Commande Etape Transition)*.

El documento no solo abarca la programación en todo detalle sino también como la planta está estructurada y sus características. Además se ha realizado un análisis de sus componentes a nivel eléctrico y mecánico.

Por último, se han añadido unos esquemas diseñados en 3D de la planta en cuestión.

**Palabras clave:** GRAFCET, PLC, autómata, sensor, actuador.

# Abstract

---

The project was born from the idea of integrating the acquired knowledge as a computer engineer, in the industrial field, specifically in the automatic field.

This project consist of automate a small bottled industrial plant through *PLC (Programmable Logic Controller)*. For performance it, we have used a software tool, named *PL7 Micro*, centered in automaton logic design, in this case, our design has been done through a *GRAFCET*.

The document does not just embrace all of programming in detail, also it embrace as the plant is structured and his features. As well a review of his components has been done in electric and mechanical terms.

Finally, a 3D designed schema of plant has been added.

**Keywords:** *GRAFCET*, PLC, automaton, sensor, actuator.



# Tabla de contenidos

---

1	Introducción .....	15
1.1	Motivación y alcance.....	15
1.2	Objetivos .....	15
1.3	Metodología .....	16
1.4	Estructura .....	16
2	Definiciones .....	17
3	Sistema IPC-200 .....	18
3.1	Características IPC-200 .....	19
3.2	Estaciones del sistema .....	19
3.2.1	Estación de producción .....	19
3.2.2	Estación de llenado.....	20
3.2.3	Estación de paletizado .....	21
3.3	Requerimientos.....	22
3.3.1	Requerimientos espaciales .....	23
3.3.2	Requerimientos eléctricos .....	23
3.3.3	Requerimientos de aire .....	24
3.4	Instalación .....	25
3.5	Mantenimiento .....	26
3.5.1	Inspección diaria .....	26
3.5.2	Inspección periódica.....	26
3.5.3	Elementos neumáticos .....	27
3.6	Manual de contingencias .....	28
3.7	Normativa .....	29
3.8	Comunicación PC.....	30
3.8.1	Elementos de la comunicación .....	30
3.8.2	Configuración del TSXETZ410.....	31
3.8.3	PLC .....	32
3.8.4	Puesta en marcha .....	34
4	Sensores.....	37
4.1	Sensores magnéticos.....	37
4.1.1	Sensores inductivos .....	37



4.1.2	Sensores capacitivos .....	38
4.1.3	Sensor magnético reed .....	38
4.1.4	Sensor magnético de estado sólido PNP .....	40
4.2	Sensores fotoeléctricos .....	41
4.2.1	Modos de funcionamiento .....	42
5	Actuadores .....	45
5.1	Análisis neumático.....	46
6	Diseño e implementación .....	48
6.1	Estructuración y diseño .....	49
6.1.1	Análisis funcional .....	51
6.2	Implementación.....	61
6.2.1	Etapas inicial.....	63
6.2.2	Etapas 1-14.....	65
6.2.3	Etapas 20-22 .....	76
6.2.4	Etapas 30-35.....	79
6.2.5	Etapas 40-43 .....	85
6.2.6	Etapas 50-59.....	88
6.2.7	Etapas 16-19.....	94
6.2.8	Convergencia Y Divergencia.....	98
6.2.9	Objetos predefinidos, variables internas.....	101
6.2.10	Funcionalidad de la botonera.....	112
6.2.11	Futuras mejoras.....	117
7	Conclusión.....	120
8	Bibliografía.....	121
9	Anexo .....	122
9.1	Anexo A: Esquemas electro-neumáticos .....	122
9.2	Anexo B: Esquemas mecánicos .....	124

# Tabla de Ilustraciones

---

Ilustración 1: Maqueta IPC-200 .....	18
Ilustración 2: Estación de producción.....	20
Ilustración 3: Estación de llenado .....	21
Ilustración 4: Estación de paletizado .....	22
Ilustración 5: Requerimientos de espacio .....	23
Ilustración 6: Esquema de comunicación PLC-PC.....	30
Ilustración 7: Módulo TSXETZ410.....	31
Ilustración 8: Menú Setup del TSXETZ410 .....	32
Ilustración 9: PLC modelo TSX 3720.....	33
Ilustración 10: Interfaz Pl7 pro .....	35
Ilustración 11: Sensores magnéticos reed.....	39
Ilustración 12: Sensores magnéticos de estado sólido .....	41
Ilustración 13: Sensores fotoeléctricos .....	42
Ilustración 14: Funcionamiento por palpación directa .....	43
Ilustración 15: Funcionamiento por sistema de barrera .....	43
Ilustración 16: Funcionamiento por sistema emisor-receptor.....	44
Ilustración 17: Funcionamiento por sistema emisor-receptor.....	44
Ilustración 18: Cilindro neumático .....	45
Ilustración 19: Electroválvula de 3/2 .....	46
Ilustración 20: Electroválvula de 5/2.....	47
Ilustración 21: Bloques funcionales.....	49
Ilustración 22: Alimentador de botes completo.....	52
Ilustración 23: Alimentador de botes (vista inferior) .....	53
Ilustración 24: Alimentador de botes (Vista trasera).....	54
Ilustración 25: Depósito de bote en mesa giratoria .....	55
Ilustración 26: Estación de llenado .....	56
Ilustración 27: Alimentador de tapas.....	57
Ilustración 28: Prensado de tapas.....	58
Ilustración 29: Transporte de botes .....	59
Ilustración 30: Giro de mesa .....	60
Ilustración 31: GRAFCET, Etapa inicial .....	63
Ilustración 32: Post, Inicialización del proceso.....	64
Ilustración 33: GRAFCET, Etapa “inicial” .....	64
Ilustración 34: Post, Etapa o .....	64
Ilustración 35: GRAFCET, Alimentador de botes II .....	65
Ilustración 36: GRAFCET, Alimentado de botes I .....	65
Ilustración 37: Post, Avance alimentador botes.....	66
Ilustración 38: Post, Avance control posición (Set) .....	67
Ilustración 39: Post, Avance empuje salida bote.....	68
Ilustración 40: Post, Avance control posición bote (Reset) .....	68
Ilustración 41: Post, Giro bote al revés (Set) y Avance control posición (Reset) .....	69
Ilustración 42: Post, Apertura pinza introducción (Set) y Giro bote al revés (Reset) ...	70
Ilustración 43: Post, Avance lineal introducción bote (Set).....	71

Ilustración 44: Post, Avance giro introducción bote (Set) .....	72
Ilustración 45: Post, Avance lineal introducción bote (Reset) .....	73
Ilustración 46: Post, Apertura pinza introducción (Reset) .....	76
Ilustración 47: Post, Avance giro introducción bote (Reset).....	75
Ilustración 48: GRAFCET, Llenado de botes .....	76
Ilustración 49: Post, Contadores asociados a la mesa giratoria.....	77
Ilustración 50: Post, Llenado bote .....	77
Ilustración 51: GRAFCET, Alimentado de tapas .....	79
Ilustración 52: Post, %C1 + COMPARE .....	80
Ilustración 53: Transición, Etapas 30-31 .....	79
Ilustración 54: Post, Cierre alimentador tapas (Set).....	80
Ilustración 55: Post, Apertura alimentador tapas (Set).....	81
Ilustración 56: Post, Apertura alimentador tapas (Reset).....	82
Ilustración 57: Post, Cierre alimentador tapas (Reset) .....	83
Ilustración 58: GRAFCET, Prensado de tapas .....	84
Ilustración 59: Post, %C2 + Compare .....	84
Ilustración 60: Transición, Etapas 40-41 .....	85
Ilustración 61: Post, Avance prensa (Set) .....	85
Ilustración 62: Post, Avance prensa (Reset) .....	86
Ilustración 63: GRAFCET, Transporte de botes .....	87
Ilustración 64: Post, Avance lineal extracción bote (Set) .....	88
Ilustración 65: Post, Apertura pinza extracción bote (Set) .....	89
Ilustración 66: Post, Avance lineal extracción bote (Reset) .....	89
Ilustración 67: Post, Avance giro extracción bote (Set) .....	90
Ilustración 68: Post, Apertura pinza extracción bote (Reset) .....	91
Ilustración 69: Post, Avance giro extracción bote (Reset) .....	92
Ilustración 70: GRAFCET, Giro mesa .....	93
Ilustración 71: Post, Avance bloqueo y arrastre .....	94
Ilustración 72: Post, Avance giro plato (Set) .....	94
Ilustración 73: Post, Avance bloqueo y arrastre (Reset) .....	95
Ilustración 74: Post, Avance giro plato (Reset) .....	96
Ilustración 75: GRAFCET, Convergencia/Divergencia .....	97
Ilustración 76: GRAFCET, Convergencia/Divergencia en OR .....	98
Ilustración 77: GRAFCET, Divergencia en AND.....	98
Ilustración 78: GRAFCET, Convergencia en AND.....	99
Ilustración 79: Temporizador .....	100
Ilustración 80: Cronograma, Temporizador TON .....	101
Ilustración 81: Cronograma, Temporizador TOF .....	102
Ilustración 82: Cronograma, Temporizador TP .....	103
Ilustración 83: Variables, Temporizadores .....	104
Ilustración 84: Contador .....	105
Ilustración 85: Variables, Contadores .....	107
Ilustración 86: Variables, Variables internas .....	108
Ilustración 87: Comparador .....	109
Ilustración 88: Botonera .....	111
Ilustración 89: Post, Botón Start .....	112
Ilustración 90: GRAFCET, Divergencia/Convergencia (Reset) .....	113
Ilustración 91: Post, Botón Reset .....	114

Ilustración 92: Post, Botón Stop .....	115
Ilustración 93: Mejora I, Alimentador de botes .....	117





# Índice de Tablas

---

Tabla 1: Potencia máxima para la estaciones .....	18
Tabla 2: Consumo de aire para actuadores de la estación de llenado.....	25
Tabla 3: Inspección diaria .....	26
Tabla 4: Inspección periódica I .....	26
Tabla 5: Inspección periódica II.....	26
Tabla 6: Manual de Contingencias I.....	28
Tabla 7: Manual de Contingencias II.....	29
Tabla 8: Características técnicas del sensor D-A7.....	38
Tabla 9: Características técnicas del sensor D-A93 .....	39
Tabla 10: Características técnicas del sensor D-Y7P .....	39
Tabla 11: Entradas del PLC.....	60
Tabla 12: Salidas del PLC .....	60
Tabla 13: Correspondencia Entradas .....	61
Tabla 14: Correspondencia Salidas.....	61
Tabla 15: Parámetros de los temporizadores .....	103
Tabla 16: Parámetros de los contadores.....	105
Tabla 17: Funcionalidad de los contadores.....	106
Tabla 18: Parámetros de los comparadores verticales.....	110





# 1 Introducción

---

## 1.1 Motivación y alcance

Este proyecto es sin duda un trabajo el cual está muy a la orden del día, orientado para cualquier ingeniero, especialmente aquellos que se dediquen al campo de la automática.

Un buen diseño y una buena implementación pueden mejorar la productividad de una empresa considerablemente con quizás unos recursos limitados. Lo más interesante de este trabajo es sin duda la total confianza y robustez del software para que este sea capaz por si solo de llevar un control total sobre toda la planta industrial.

Por otro lado cabe destacar la facilidad de manejo y de control sobre el autómatas con solo una simple botonera, ver Capítulo siete: Diseño e implementación, mediante la cual un operario podría controlar toda la planta.

Por último, el *software* no requiere de mantenimiento ya que una vez diseñado y puesto en marcha. El mantenimiento se reduciría más bien a nivel mecánico y eléctrico.

## 1.2 Objetivos

El proyecto realizado ha consistido en automatizar el proceso industrial de una planta embotelladora.

Los objetivos a cubrir son los siguientes:

- Diseñar e implementar un *GRAFCE*T que controle dicho proceso.
- Diseñar un mecanismo de control para situaciones de emergencia.

El primer objetivo ha consistido en realizar un diseño e implementación no solo correcta si no que teniendo en cuenta los recursos con los que contamos a nivel de sensores y actuadores, conseguir el rendimiento óptimo de la cadena de producción.

Por otro lado como en cualquier planta industrial, pueden darse situaciones inesperadas, anómalas, para las cuales se ha implementado un mecanismo para poder bloquear inmediatamente toda acción para poder resolver esta inestabilidad del sistema y continuar cuanto antes con el proceso.

### 1.3 Metodología

*Pl7 Micro* es un programa creado por *Schneider-electric* y ha sido nuestra herramienta principal de trabajo. Soporta cuatro lenguajes de programación: lenguaje de contactos, lista de instrucciones, literal estructurado y *GRAFSET*, mediante los cuales podemos representar gráficamente y de forma estructurada el funcionamiento de un automatismo secuencial. Contiene una paleta de elementos gráficos accesibles mediante teclado o ratón permitiendo configurar: etapas, transiciones, enlaces, reenvíos, macro-etapas....

En nuestro caso la implementación ha sido realizada mediante el lenguaje *GRAFSET*, subdividiendo el problema en seis bloques programados y depurados de forma individual. Una vez finalizado este proceso, se unificaron todos los bloques en uno único, conformando así el proyecto.

Cabe destacar, que los barriles han sido sustituidos por botes de carrete de fotografía para de esta forma poder simular el funcionamiento de la planta.

### 1.4 Estructura

El proyecto ha sido estructurado en ocho partes principales, donde podemos encontrar todo el trabajo desarrollado.

Las dos primeras partes conforman la introducción y una serie de definiciones básicas que ayudan a comprender mejor el trabajo.

En la tercera parte, analizamos detalladamente las características de la estación de trabajo, requerimientos, mantenimiento y como interactúa con el computador, así como la normativa que cumple.

En la cuarta y quinta parte hemos realizado un análisis de los sensores y actuadores utilizados en un enfoque neumático y eléctrico.

La sexta parte incluye todos los aspectos relacionados con el diseño e implementación, incluyendo un análisis funcional detallado de la estación.

En la séptima parte hemos añadido unos esquemas en 3D, en los cuales podemos ver con cierto nivel de detalle cómo la planta está construida.

Por último se encuentra la conclusión del trabajo, donde una vez madurado y finalizado se han comentado los resultados y se ha realizado un análisis del impacto que este ha tenido.

## 2 Definiciones

---

Automatización industrial: Uso de sistemas o elementos computarizados y electromecánicos para controlar maquinarias o procesos industriales. Como una disciplina de la ingeniería más amplia que un sistema de control, abarca la instrumentación industrial, que incluye los sensores, los transmisores de campo, los sistemas de control y supervisión, los sistemas de transmisión y recolección de datos y las aplicaciones de software en tiempo real para supervisar, controlar las operaciones de plantas o procesos industriales

GRAFCET (GRAphe Fonctionnel de Commande Etape Transition): Es un grafo o diagrama funcional normalizado, que permite hacer un modelo del proceso a automatizar, contemplando entradas, acciones a realizar, y los procesos intermedios que provocan estas acciones, lo que lo ha convertido en un potente lenguaje gráfico de programación para autómatas, adaptado a la resolución de sistemas secuenciales.

PLC (Programmable Logic Controller): Es un equipo electrónico diseñado para controlar en tiempo real y en ambiente de tipo industrial procesos secuenciales. Fueron diseñados para cubrir las necesidades de control de cualquier tipo de maquinaria.

Sensor: Dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación y transformarlas en variables eléctricas. Destacan en áreas de la industria.

Actuador: Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador para finalmente generar una orden de control sobre un elemento final.

## 3 Sistema IPC-200

---

En este capítulo hemos realizado un análisis funcional del sistema de control de procesos industriales IPC-200, analizando las características básicas, estaciones que lo conforman, condiciones de uso y mantenimiento, así como, la comunicación con el pc y la normativa europea vigente.

El sistema IPC-200 se encuentra alojado en la UPV (Universidad Politécnica de Valencia), concretamente en el laboratorio Columbretes de la ETSID (Escuela Técnica Superior de Ingeniería del Diseño), perteneciente al DISA (Departamento de Ingeniería de Sistemas y Automática).

Para la elaboración del proyecto hemos sustituido el agua destilada por aire a presión regulado y hemos utilizado cartuchos de carretes de fotografía como materia prima.

En la ilustración 1, podemos observar el sistema IPC-200 con sus 3 estaciones.



pág. 1)

*Ilustración 1: Maqueta IPC-200 (SMC International training, Manual de Prácticas,*

### 3.1 Características IPC-200

Dicho sistema tiene como objetivo simular una planta de producción y embotellado de bebidas, integrando diversas tecnologías relacionadas los sistemas de control de procesos industriales.

Es una herramienta útil para realizar estudios de tecnologías industriales en campos como: neumática, PLC, control de ejes eléctricos, control de fluidos, etc.

Por otro lado es una herramienta que puede ayudar al ingeniero a desarrollar capacidades tales como: análisis y diseño, programación de PLC e interpretación de y elaboración de documentación técnica.

### 3.2 Estaciones del sistema

IPC-200, lleva integrado tres estaciones: estación de producción, estación de llenado y estación de paletizado.

#### 3.2.1 Estación de producción

La estación de producción simula la etapa de producción y mezclado de bebidas, cuenta con dos variantes, la primera basada en un control digital (todo o nada) y una segunda variante gobernada por un controlador PID (**P**roportional **I**ntegral **D**erivativo).

En la ilustración 2 podemos observar la estación de producción diferenciando cada uno de sus puestos.



*Ilustración 2: Estación de producción (SMC International training, Manual de Usuario, Estaciones del proceso, pág. 8)*

### 3.2.2 Estación de llenado

La estación de llenado reproduce el proceso de embotellado del producto, que es el proceso que ocupa este proyecto y sobre el que hemos analizado con todo detalle ver Capítulo 7: Diseño e implementación.

Dicha estación se caracteriza por la manipulación neumática, electro-neumática y sensorial. Por otro lado cuenta con dos variantes referentes al mecanismo de alimentación de botes. La primera variante, consiste en un alimentador simple, el cual se caracteriza por tener los botes almacenados unos encima de los otros, de forma que al alimentar el bote que se encuentra en la parte inferior, los otros caen por efecto de la gravedad, es importante que el operario introduzca los botes en el sentido correcto en el alimentador para el correcto funcionamiento del sistema.

El segundo modelo, es el que hemos utilizado para el trabajo, basado también en un modelo gravitacional pero más complejo que el anterior modelo, ya que este realiza el control de la orientación del bote, girando el bote si fuera necesario.

En la ilustración 3 podemos observar dicha estación etiquetada con las componentes más relevantes.

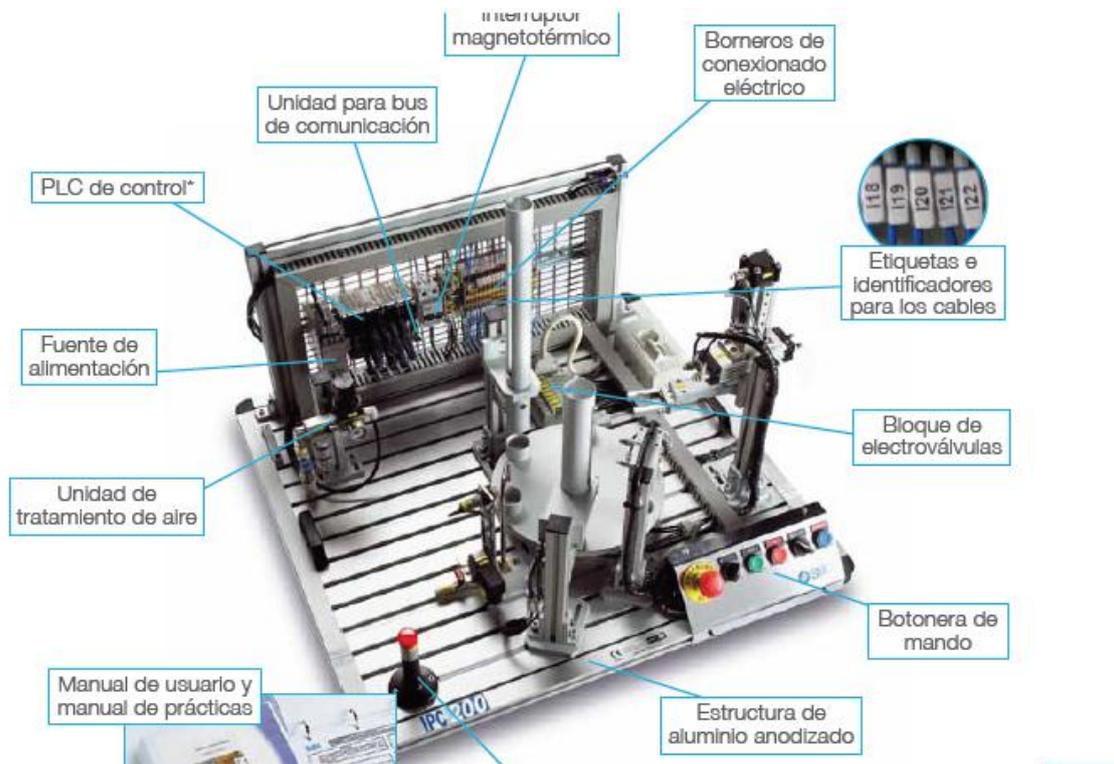


Ilustración 3: Estación de llenado (SMC International training, Control de Procesos Industriales, pág. 215)

### 3.2.3 Estación de paletizado

En la tercera estación se realiza la simulación del paletizado y almacenamiento del producto final. Para ello se utilizan *drivers* posicionadores controlados con entradas y salidas digitales desde el PLC.

En la ilustración siguiente podemos observar la estación correspondiente al paletizado con sus puestos bien diferenciados.



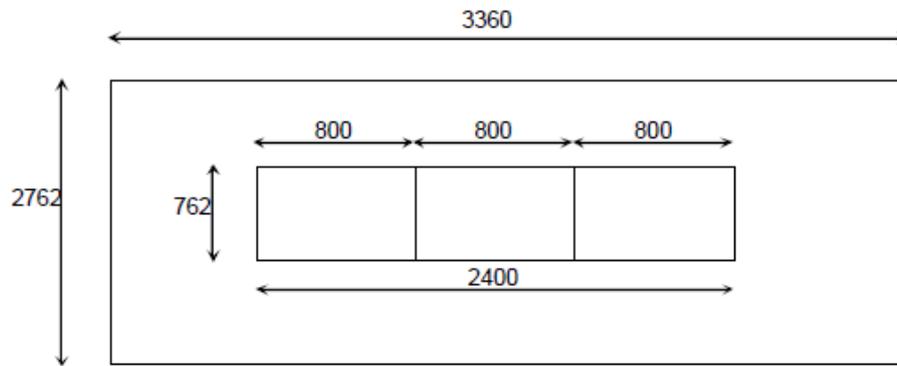
*Ilustración 4: Estación de paletizado (SMC International training , Manual de Usuario, Estaciones del proceso, pág. 23)*

### 3.3 Requerimientos

En esta sección hemos reunido las condiciones necesarias para que el sistema tenga una instalación y puesta en marcha satisfactoria.

### 3.3.1 Requerimientos espaciales

IPC-200 está constituido por tres estaciones, por lo que se precisa de un espacio suficiente para la instalación y un espacio de trabajo para la manipulación de estas. En la ilustración 1 podemos apreciar las dimensiones de cada una de las estaciones que componen la maqueta y el espacio necesario para su utilización.



*Ilustración 5: Requerimientos de espacio*

Por otro lado es recomendable que el habitáculo donde esté instalada la maqueta cumpla los siguientes requisitos:

- La maqueta debe estar situada sobre una superficie totalmente nivelada, rígida, de una altura aproximada de 80 cm.
- El entorno más cercano debe estar exento de perturbaciones y vibraciones.
- Ambiente limpio de polvo y suciedad.
- Alejar lo máximo posible las fuentes de EMI( Incompatibilidad Electro Magnética )
- Verificar la calidad de la toma de tierra.
- Temperatura ambiente moderada.

### 3.3.2 Requerimientos eléctricos

Independientemente de que estación o estaciones se estén utilizando, la maqueta tendrá que estar alimentada por una tensión de 220V.

Por otro lado cada estación requiere de una potencia máxima, reflejada en la tabla 1.

Estación de producción (todo o nada)	85 W
Estación de producción(PID)	500W
Estación de llenado (simple)	15W
Estación de llenado (completa)	20W
Estación de paletizado	110W

*Tabla 1: Potencia máxima para las estaciones*

### 3.3.3 Requerimientos de aire

Dado que cada estación requiere unos mantenimientos de aire distintos y este proyecto está centrado en la estación de llenado, vamos a centrarnos en esta estación para hallar los requisitos.

Los requerimientos de aire como veremos en la tabla 2 está justificados por el consumo generado.

Aplicando una conversión a CV (caballos de vapor), partiendo de que 100l/min equivale a 1CV, obtenemos un total de  $8,5726/100 = 0,086$  CV, posteriormente realizamos la conversión a *Watt* partiendo de que 1 CV equivale a 745,7 W, por lo que se requiere una potencia eléctrica de 64,13 W.

N de Ciclos	Elemento	Carrera	Diámetro	Consumo (l/ciclo)
2	MGPM12-10	10	12	0,0046
2	MGPM12-10	10	12	0,0046
2	CDQ2B12-15D	15	12	0,0136
2	MSQB10A			0,0490
2	CDQSB12-75DCM	75	12	0,0681
6	MRQBS32-50CB	50	32	0,4841
6	MRQBS32-50CB	50	32	0,4841
2	MRHQ16D-90S-N	90	16	0,0097
2	CDQ2B25-50D	50	25	0,1970
2	CQ2B16-10D	10	16	0,0080
2	CQ2B16-10D	10	16	0,0080
2	MHZ2-16D	6	16	0,0097
2	CDQ2B20-30D	30	20	0,0756
2	MHZ2-16DN	6	16	0,0048
2	MHZ2-16DN	6	16	0,0048
			TOTAL:	8,5726 (l/min)

Tabla 2: Consumo de aire para actuadores de la estación de llenado

### 3.4 Instalación

Previamente a la instalación del habitáculo donde se va a realizar esta debe cumplir los requerimientos, ver Requerimientos sección 3.3, en ese caso se procede al conexionado eléctrico y neumático de la estación.

Por motivos de seguridad se recomienda que para la instalación de ambos se corte la tensión de la estación y se cierre la llave de paso de aire.

En el caso de la instalación eléctrica, se realiza mediante la conexión del enchufe de la estación a la toma de corriente.

La instalación neumática se realizara mediante un tubo de diámetro 6 desde la toma del habitáculo a la entrada del grupo de tratamiento de aire de la estación.

### 3.5 Mantenimiento

Para asegurar un buen mantenimiento de la maqueta es recomendable seguir un protocolo de inspección tal como se ha indicado en las tablas 3, 4, 5 y por otro lado la revisión interna de los componentes neumáticos.

#### 3.5.1 Inspección diaria

Punto de inspección	Acción correctora
Comprobar que el nivel de condensados de los filtros no supere las marcas.	Purgar
Comprobar que la caída de presión en el filtro sea inferior a un bar.	Sustituir cartucho filtrante
Mirar que los tornillos y tuercas no estén	Apretar con la herramienta adecuada.

Tabla 3: Inspección diaria

#### 3.5.2 Inspección periódica

Punto de inspección	Acción correctora
Verificar que en la estación no existan obstáculos que puedan distorsionar el funcionamiento del sistema.	Retirar.
Comprobar que no hay ningún componente suelto, desalineado o mal colocado.	Corregir según proceda.
Verificar que suficiente aire a la estación.	Ver localización y corrección de averías.
Verificar que llega tensión a la estación.	Ver localización y corrección de averías.

Tabla 4: Inspección periódica I

flojos o sueltos. (6 meses)	
Verificar que los cilindros funcionan correctamente. (6 meses)	Cambio de juntas si procede
Engrasado de los ejes de la estación de paletizado.	Engrasado
Limpieza de la maqueta. (6 meses)	Limpiar

Tabla 5: Inspección periódica II

### 3.5.3 Elementos neumáticos

Principalmente, los elementos neumáticos de la IPC-200 fueron lubricados en su instalación y no cabe hacer ningún tipo de mantenimiento de forma diaria.

Existe la posibilidad de que tras un uso continuado durante un tiempo prolongado, se produzca un desgaste en las juntas internas de los cilindros. En el caso en el que el cilindro fuera desmontable simplemente bastaría con desmontar estos y cambiar las juntas, en caso contrario, se tendría que sustituir la pieza completa.

Cada ciertos meses o anualmente es conveniente realizar una inspección de los niveles de presión. Si en los filtros se detecta una caída de presión superior a un bar, se debería reemplazar el cartucho filtrante por uno nuevo.

### 3.6 Manual de contingencias

Como hemos observado, ver Estaciones del sistema, sección 3.2, el sistema IPC-200 cuenta con tres estaciones, dado que el plan de contingencia es individual y el proyecto versa sobre la estación de llenado, hemos realizado esta sección. En las tablas 6 y 7 podemos observar cómo el procedimiento que debemos seguir ante una situación de avería de la estación.

Nº	Síntoma	Posible Causa	Remedio
2.1	La estación no funciona después de darle al botón de marcha.	La estación está desenchufada o la aula no tiene tensión. El interruptor magnetotérmico está abierto. La seta de emergencia está enclavada. La seta de emergencia está enclavada. Toma de aire no conectada.	Enchufar la estación o la tensión en el aula. Cerrar interruptor (ON), si resulta que al volver a arrancar se salta es que la potencia instalada es insuficiente. Desenclavar la seta de emergencia. Desenclavar la seta de emergencia. Conectar a toma de aire o cerrar la válvula de aire.
2.2	La estación no se pone en marcha y se activa la baliza.	La estación no se encuentra en las condiciones iniciales.	Se pulsa la Seta de EMERGENCIA para que los elementos neumáticos vuelvan a su origen.
2.3	El plato divisor se ha parado y a pesar de arrancar de nuevo no se mueve.	Cilindros de enclavamiento atascados.	Accionar manualmente la electroválvula correspondiente para que los cilindros vuelvan a su posición.
2.4	La estación arranca pero al cabo del tiempo se para.	No llega suficiente aire a los actuadores. La instalación neumática del aula no cumple los requerimientos mínimos.	Regular la presión del aire de la estación a 3 Kg/cm <sup>2</sup> Poner un compresor más grande o trasladarse a un aula adecuada.
2.5	La estación funciona, pero se para y se enciende la baliza de color rojo.	Se ha enclavado la Seta de EMERGENCIA.	Desenclavar la Seta de EMERGENCIA.
2.6	Un actuador se mueve a velocidad excesiva.	1) Los reguladores de caudal están ajustados al	Regular el caudal hasta alcanzar la velocidad

Tabla 6: Manual de contingencia I (SMC International training, Manual de Usuario, Mantenimiento e inspección, pág. 38)

		mínimo.	deseada.
2.7	Un actuador se mueve con excesiva lentitud.	Los reguladores de velocidad están ajustados al máximo. La regulación de aire de la estación no alcanza el valor mínimo requerido. Las juntas del cilindro están desgastadas.	Regular el caudal hasta alcanzar la velocidad deseada. Regular la presión del aire de la estación a 3 Kg/cm <sup>2</sup> Cambiar juntas o cilindro según proceda.
2.8	Al llegar siempre al mismo punto la estación se para.	Detector averiado. Electroválvula averiada.	Reemplazar detector. Sustituir electroválvula.
2.9	Colocación inadecuada del bote por parte de algún manipulador.	Detector averiado. Medios de sujeción flojos.	Reemplazar detector. Apretar tornillos flojos.
2.10	Se enciende los detectores y las luces del autómeta, pero no funciona ningún pulsador, ni luce la baliza.	El PLC no tiene programa. El programa del PLC está en modo STOP.	Introducir programa al PLC. Cambiar del modo STOP al modo RUN.

Tabla 7: Manual de contingencia II (SMC international training , Manual de Usuario, Mantenimiento e inspección, pág. 38)

### 3.7 Normativa

En esta sección se ha recogido la normativa europea referente a los requerimientos, ver Requerimientos, sección 3.3.

- **UNE-EN-294: 1993** Seguridad de las máquinas. Distancia de seguridad para impedir que se alcancen zonas peligrosas con los miembros superiores.
- **UNE-EN-811: 1997** Seguridad de las máquinas. Distancias de seguridad para impedir que se alcancen zonas peligrosas con los miembros inferiores.
- **UNE-EN-349: 1994** Seguridad de las máquinas. Distancias mínimas para evitar el aplastamiento de partes del cuerpo.
- **UNE-EN-292-1: 1993** Seguridad de las máquinas. Conceptos básicos y principios generales para el diseño. Tecnología básica.
- **UNE-EN-292-2: 1993 / UNE-EN-292-2/A1: 1996** Seguridad de las máquinas. Conceptos básicos y principios generales para el diseño. Principios y especificaciones técnicas.
- **UNE-EN 60204-1:** Seguridad de las máquinas. Equipo eléctrico de las máquinas.

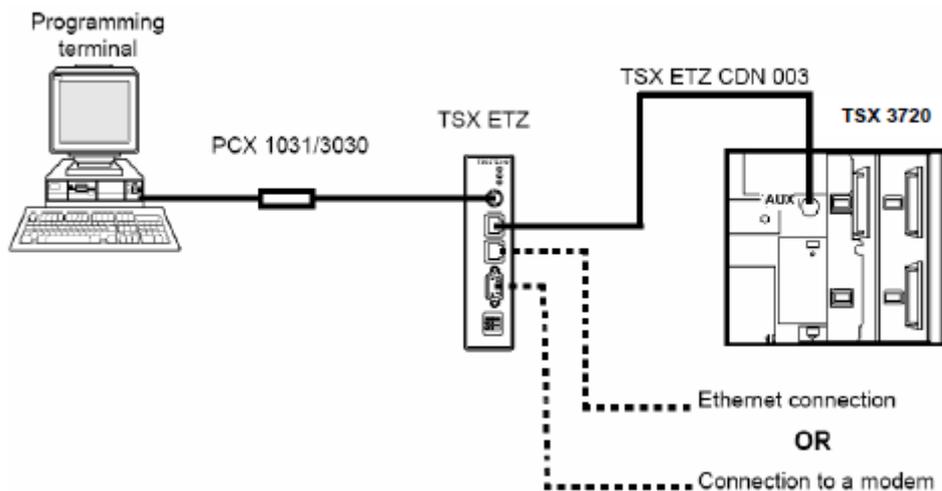
(SMC International training, Manual de Prácticas, Análisis funcional, pág. 95)

### 3.8 Comunicación PC

En esta sección hemos analizado y estudiado como el PLC se comunica con el PC, partiendo de un esquema, ver ilustración 6, veremos cada uno de los elementos necesarios, entrando en detalle en aquellos aspectos relevantes para que la comunicación pueda establecerse con éxito.

#### 3.8.1 Elementos de la comunicación

Para que la conexión pueda llevarse a cabo con éxito se precisa de tres elementos clave mostrados a continuación.



1. Computador (*programming terminal*)
2. Módulo *Ethernet* (TSXETZ410)
3. PLC (TSX 3720)

Ilustración 6: Esquema de comunicación PLC-PC (SMC Internacional training, Manual de Usuario - Anexo D, pág. 20)

La comunicación entre el PC y el PLC es realizada a través del puerto TER del módulo TSXETZ410, y de ahí, al puerto serie del PC. El módulo se conecta con el PLC a través del puerto RS485 que va al puerto AUX del PC. El puerto TER del PLC queda libre para las comunicaciones.

### 3.8.2 Configuración del TSXETZ410

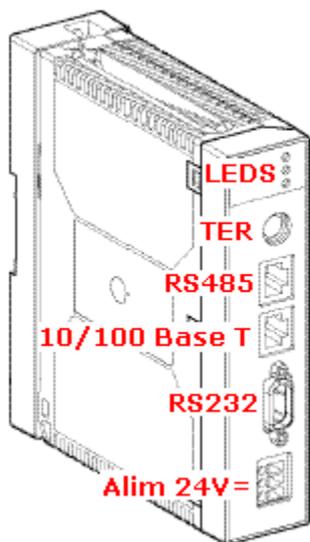


Ilustración 7: Módulo TSXETZ410

Para ver más claro lo explicado referente a este módulo, ver Elementos de la comunicación, sección 3.7.2, cabe observar la ilustración 7, donde podemos ver señalados las distintas conexiones de este módulo.

Para poder conectarnos con este módulo vía *Ethernet* es totalmente necesario su dirección IP. Schneider-electric fija una dirección IP por defecto, aunque esta puede ser gestionada posteriormente por el usuario.

Para identificar esta dirección IP basta con mirar la parte frontal del módulo ETZ410, donde podemos encontrar la dirección MAC de este y a partir de esta, obtener la dirección IP.

En este caso la dirección IP es la siguiente:

085.016.xxx.yyy siendo xxx e yyy los dos últimos números de la dirección MAC: 00-80-F4-06-CE-24.

Si el usuario desea cambiar el dominio del módulo, es necesario cambiar la dirección IP desde la misma página del servidor donde entrando en el menú *Setup* podrá realizar los cambios oportunos.

En la ilustración 8 queda reflejada la interfaz *Setup* y remarcada la zona *IP Address* la cual debe ser modificada.

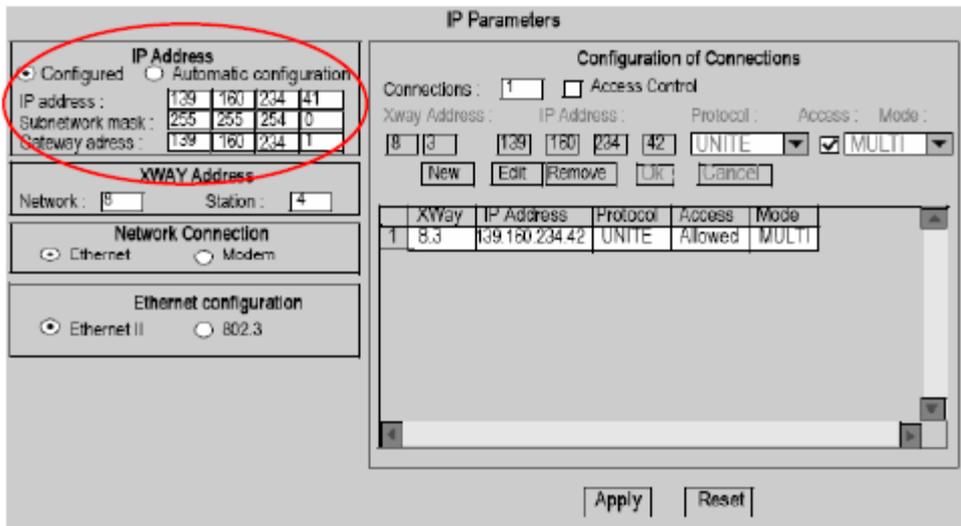


Ilustración 8: Menú Setup del TSXETZ410 (SMC International training. , Manual de Usuario - Anexo D)

### 3.8.3 PLC

En esta sección hemos analizado la estructura de un PLC identificando sus componentes con el fin de entender el funcionamiento interno de este elemento.

En la ilustración 9, podemos observar como es físicamente un PLC, concretamente el que ha sido utilizado para la realización del proyecto, el modelo TSX 3720



Ilustración 9: PLC modelo TSX 3720

Un PLC está compuesto por los siguientes bloques:

- CPU (Central Processing Unit)
- Memoria
- Interfaz de entradas y salidas
- Fuente de alimentación
- Bus interno

La **CPU** es la encargada de consultar el estado de las entradas y recoger de memoria la secuencia de instrucciones a ejecutar para generar las órdenes que enviará al proceso.

La **memoria** contiene todos los datos e instrucciones que se necesita para ejecutar la tarea de control. Hay dos tipos de memoria: interna y programada.

- La memoria interna es la encargada de almacenar datos de cálculo intermedios al igual que las variables internas.
- La memoria programada contiene la secuencia de operaciones que deben realizarse sobre las señales de entrada para obtener las señales de salida.

La **interfaz de entradas y salidas**, establece la comunicación del autómata con la planta, es decir, se encarga de adaptar las señales que se manejan en el proceso a las utilizadas internamente por la máquina.

La **fuerza de alimentación**, dota a los circuitos electrónicos del sistema las tensiones necesarias. Por motivos de seguridad los PLC pueden incorporar una batería adicional conectada a la fuente, por si se diera una interrupción de la tensión externa, asegurar el funcionamiento de la planta.

El **bus interno** está compuesto por una serie de conexiones que realizan la unión eléctrica entre la CPU, la interfaz y las memorias.

### 3.8.4 Puesta en marcha

#### 3.8.4.1 Preparación del sistema

Inicialmente debemos realizar ciertas comprobaciones para asegurar que el sistema está en las condiciones de uso.

Debemos verificar con ayuda de un polímetro que el enchufe al que conectaremos el sistema proporciona una tensión de 220v a 50hz.

Por otro lado debemos verificar que en el manómetro de la unidad de tratamiento indica una presión aproximada de 3kg/cm<sup>2</sup>.

Una vez hemos comprobado que las condiciones eléctricas y de presión son las correctas debemos verificar que el interruptor magnetotérmico se encuentra en la posición *ON* para suministrar energía eléctrica al sistema.

Por último no hay que olvidar realizar las inspecciones oportunas, ver Mantenimiento sección 3.5.

### 3.8.4.2 Transferencia y ejecución del programa

Para realizar correctamente la transferencia debemos seguir los siguientes pasos descritos a continuación, no obstante en la ilustración 10 podemos observar la interfaz del *PL7 Micro*, para facilitar el seguimiento de los pasos a realizar.

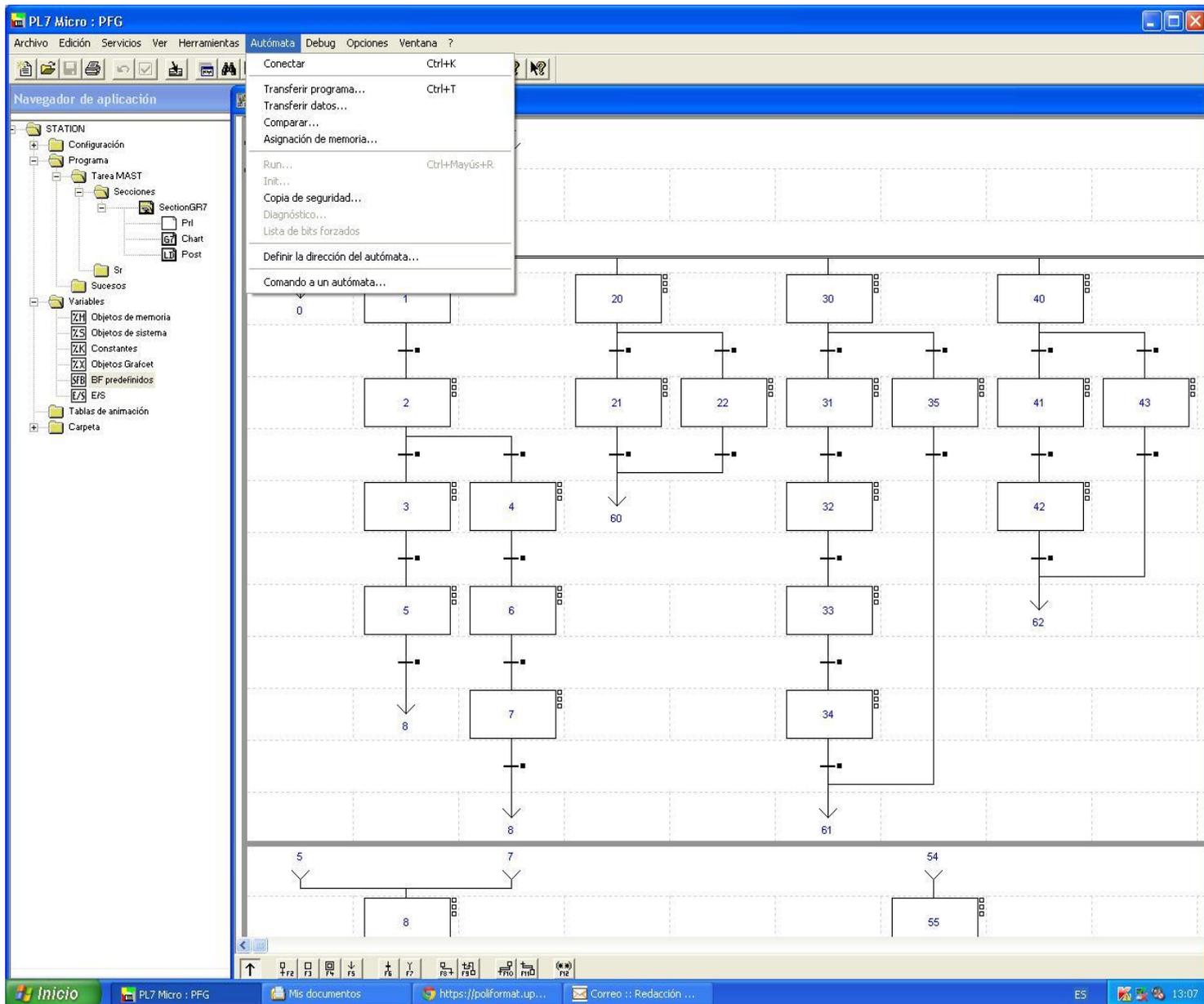


Ilustración 10: Interfaz Pl7 Micro

Para realizar la carga, transferencia y ejecución de nuestro programa con éxito debemos seguir los pasos citados a continuación:

1. **Abrir** el *PL7 Micro*.
2. Seleccionar la pestaña **Archivo** y la opción **Abrir** para seleccionar nuestro proyecto.
3. Seleccionar la pestaña **Autómata** y la opción **Transferir programa** escogiendo la modalidad **PC → Autómata**.
4. Seleccionar la pestaña **Autómata** y la opción **Conectar**.
5. Seleccionar la pestaña **Autómata** y la opción **Init** para resetear el *GRAF CET*.
6. Seleccionar la pestaña **Autómata** y la opción **Run** para iniciar la ejecución.
7. Seleccionar la pestaña **Autómata** y la opción **Stop** para detener la ejecución.

Cabe destacar que a partir del paso cuatro podemos realizar las acciones restantes desde la barra de herramientas que observamos en la parte superior de la pantalla.

# 4 Sensores

---

En este apartado se ha analizado el funcionamiento de los sensores, los cuales son un elemento principal para poder abordar este proyecto. El PLC es el encargado de leer la información proporcionada por estos y tomar las decisiones oportunas a nivel lógico según lo haya especificado el ingeniero programador. Es importante tener claro el concepto de sensor antes de entrar a analizar las características de estos, por lo que se recomienda leer su definición, ver, Definiciones, sección 2.

Haciendo un análisis visual sobre la estación de llenado, hemos podido observar y localizar tres tipos de sensores:

1. Sensores magnético *reed*
2. Sensores magnético estado sólido PNP
3. Sensor fotoeléctrico 3 hilos PNP

## 4.1 Sensores magnéticos

Tenemos dos sensores de este tipo, los cuales tienen en común el funcionamiento a grandes rasgos, ambos provocan un cierre en el circuito cuando se genera un campo magnético. Este campo magnético puede ser generado de dos maneras, de forma inductiva y de forma capacitiva.

Este tipo de sensores surgieron a razón de sustituir aquellos sensores que funcionaban de forma mecánica, mejorando la respuesta de estos ante situaciones de altas velocidades de arranque o frecuencias de conmutación elevadas.

Generalmente son utilizados como interruptores de fin de carrera, proporcionando una fidelidad elevada.

### 4.1.1 Sensores inductivos

Estos sensores constan de un oscilador cuyos bobinados constituyen la cara sensible. En la cara frontal de esta última es generado un campo magnético alterno. Cuando se coloca una pantalla metálica en este campo, las corrientes inducidas provocan una carga adicional consiguiendo que las oscilaciones cesen.

### 4.1.2 Sensores capacitivos

Estos sensores constan de un oscilador cuyos condensadores constituyen la cara sensible. Cuando un material cuya permitividad  $> 1$ , se introduce en este campo, generando variaciones de acoplamiento, desembocando en oscilaciones.

### 4.1.3 Sensor magnético reed

Podemos distinguir dos partes fundamentales: el interruptor y el imán accionador ambos relacionados entre sí.

Los interruptores se accionan mediante imanes y dependiendo del tipo de interruptor el imán será de acción frontal o acción lateral, en función de este parámetro también variarán las distancias de accionamiento.

El funcionamiento es sencillo, en el interior de estos sensores encontramos dos láminas metálicas de materiales ferromagnéticos, en el momento que se genera un campo magnético, se cierra el circuito activando el sensor.

En nuestro proyecto tenemos dos modelos distintos: D-A73 y D-A93

Para ver las características técnicas se ha adjuntado las tablas 8 y 9, donde podemos observar los valores que tienen los parámetros significativos de estos.

<b>Características técnicas</b>		PLC: Controlador Lógico Programable	
<b>D-A7 (con LED indicador)</b>			
Referencia del detector	<b>D-A72</b>	<b>D-A73</b>	
Aplicaciones	Relé/PLC	Relé/PLC	
Tensión de alimentación	200V AC	24V DC	100V AC
Rango de corriente	5 a 10mA	5 a 40mA	5 a 20mA
Circuito protección contactos	Ninguno		
Caída de tensión interna	$\leq 2.4V$		
LED indicador	LED rojo iluminado cuando el detector está en ON		

Tabla 8: Características técnicas del sensor D-A7 (SMC International training, Guía de detectores magnéticos, pág. 13)

<b>D-A93/D-A96 (con LED indicador)</b>			
Referencia del detector	<b>D-A93</b>		<b>D-A96</b>
Aplicaciones	Relé/PLC		Circuito CI
Tensión de carga	24V DC	100V AC	4 a 8V DC
Corriente de carga máx. y rango de corriente	5 a 40mA	5 a 20mA	20mA
Circuito protección contactos	Ninguno		
Caída de tensión interna	2.4V (hasta 20mA)/ 3V (hasta 40mA)		0.8V
LED indicador	LED rojo iluminado cuando el detector está en ON		

Tabla 9: Características técnicas del sensor D-A93 (SMC International training, Guía de detectores magnéticos, pág. 18)

Dado que no precisa de un contacto directo con el elemento móvil hace muy útiles estos sensores en campos como puede ser la ingeniería hidráulica donde si liberaría todo el riesgo de cortocircuitos.

Una vez entendido la constitución y funcionamiento de estos sensores es importante saber reconocerlos, por lo que en la ilustración 11 se puede apreciar los distintos tipos de sensores que hay basados en esta tecnología

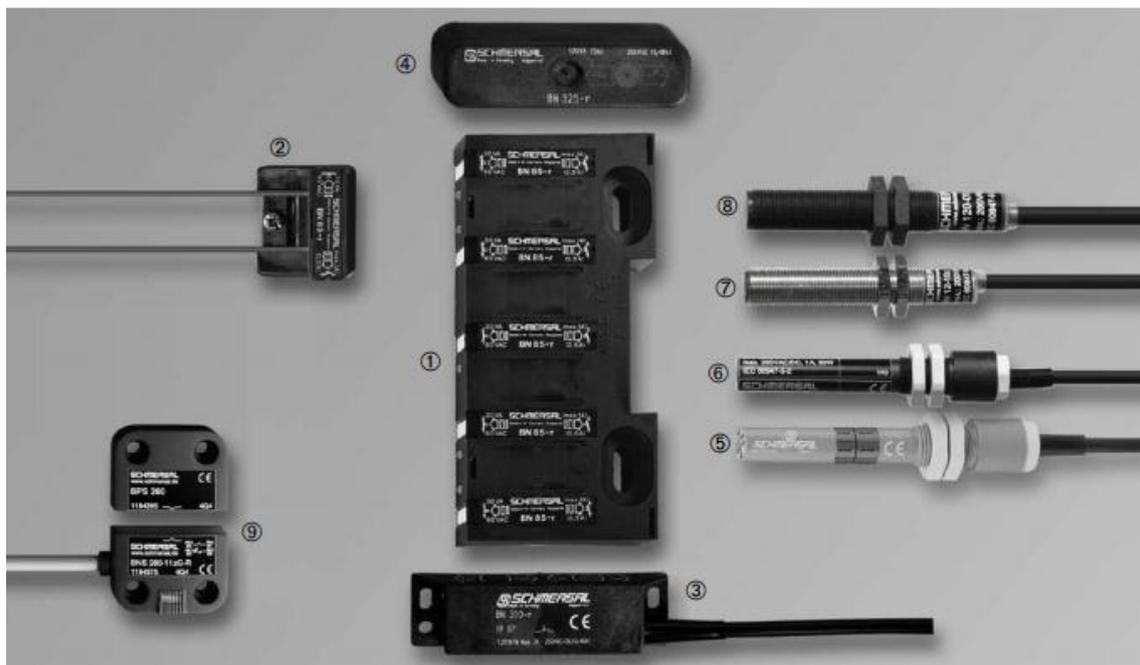


Ilustración 11: Sensores magnéticos reed

#### 4.1.4 Sensor magnético de estado sólido PNP

El elemento clave en este tipo de sensores es la resistencia magneto-sensible que incorpora. Esta resistencia reacciona ante un campo magnético de igual forma que un elemento fotosensible cuando es incidido por la luz. Estos sensores tienen implementados internamente un circuito de conversión Analógica-Digital de forma que los valores de resistencia son traducidos a *ON/OFF*. Estos detectores pueden ser de dos o tres hilos, clasificándose estos en tipo *PNP* y *NPN* en función de la forma de conexión de la carga de los mismos. En nuestro caso tenemos de tipo *PNP*.

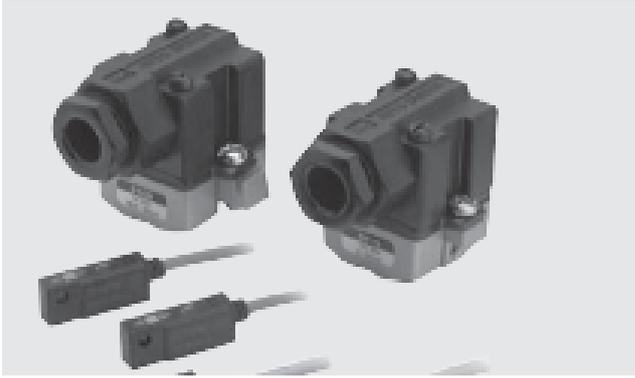
El modelo concreto de este tipo de sensores que tenemos incorporados en el proyecto es el D-F9P.

Al igual que vimos en el caso anterior, en la tabla 10, podemos observar las características técnicas de este sensor.

<b>D-Y5/D-Y6/D-Y7P/D-Y7PV (con LED indicador)</b>						
Referencia del detector	<b>D-Y59A</b>	<b>D-Y69A</b>	<b>D-Y7P</b>	<b>D-Y7PV</b>	<b>D-Y59B</b>	<b>D-Y69B</b>
Entrada eléctrica	En línea	Perpendicular	En línea	Perpendicular	En línea	Perpendicular
Cableado	3 hilos			2 hilos		
Tipo de salida	NPN		PNP		—	
Aplicaciones	Circuito C/Relé/PLC			24V DC Relé/PLC		
Tensión de alimentación	5/12/24V DC (4.5 a 28V DC)			—		
Consumo de corriente	10mA			—		
Tensión de carga	28V DC		—		24V DC (10 a 28V DC)	
Corriente de carga	40mA		80mA o menos		5 a 40mA	
Caída de tensión interna	1.5V o menos 0.8V o menos en 10mA corriente de carga		0.8V o menos		4V o menos	
Corriente de fuga	100µA o menos a 24V DC			0.8mA o menos a 24V DC		
LED indicador	LED rojo iluminado cuando el detector está en ON					

Tabla 10: Características técnicas de sensor D-Y7P (SMC International training, Guía de detectores magnéticos, pág. 42)

Por otro lado es interesante saber cómo son estos elementos físicamente por lo cual hemos adjuntado la ilustración 12



*Ilustración 12: Sensores magnéticos de estado sólido*

## 4.2 Sensores fotoeléctricos

Los sensores fotoeléctricos constan de un emisor de luz y un receptor., el receptor reacciona a las variaciones del rayo luminoso reflejado por el objeto. Están orientados a la detección, clasificación y posicionado de objetos.

Estos sensores incluyen un transductor fotoeléctrico encargado de convertir la luz en una señal eléctrica, la cual a través de un procesamiento digital se acondicionará, compensará y adoptará el formato que se requiera.

Al igual que en los otros sensores, hemos adjuntado la ilustración, en este caso número 13, donde podemos observar la gran variedad de sensores fotoeléctricos que podemos encontrar en el mercado.

En la maqueta IPC-200 solo contamos con un sensor de este tipo, situado en la estación de llenado, concretamente en el puesto de llenado, ocupándose de la detección de bote. El modelo en cuestión es E3F2-DS30B4.



*Ilustración 13: Sensores fotoeléctricos*

#### 4.2.1 Modos de funcionamiento

En la sección anterior, ver Sensores fotoeléctricos sección 4.2, analizamos el comportamiento, la esencia de este tipo de sensores, no obstante, tenemos que tener en cuenta el modo de funcionamiento que buscamos en función de la demanda.

Por lo que podemos diferenciar cuatro modos de funcionamiento:

1. Palpación directa
2. Sistema de barrera
3. Sistema enfocado
4. Sistema emisor-receptor

En el método de **palpación directa** el emisor y receptor se encuentran en la misma unidad, la luz emitida es reflejada en el objeto y es tratada por el receptor.

A continuación a través de la ilustración 13, podemos observar el escenario que representa este modo de funcionamiento.



Ilustración 14: Funcionamiento por palpación directa

El método de **sistema de barrera** cuenta con un elemento más, el reflector. Tanto el emisor como receptor se encuentran en la misma unidad. La reflexión del rayo se efectúa en el reflector, identificando la presencia de un objeto cuando esta es interrumpida.

En la ilustración 15, podemos ver el escenario correspondiente.

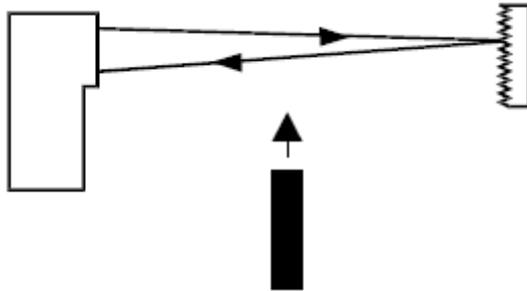
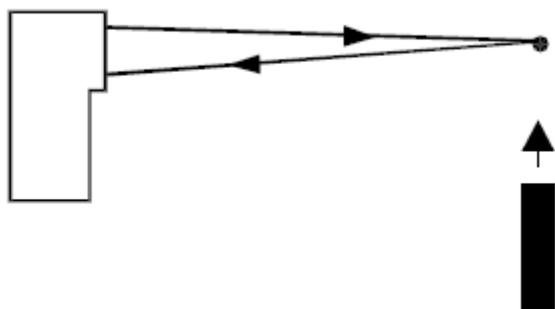


Ilustración 15: Funcionamiento por sistema de barrera

El método de **sistema enfocado** mantiene la esencia del método de sistema de barrera, ver Modos de funcionamiento, sección 4.2.1 ya que la disposición de la lente tiene un punto focal fijo a una distancia determinada. En este caso la reflexión en el objeto provoca la detección del mismo.

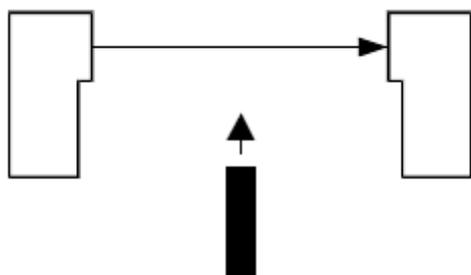
En la ilustración 16 queda puesto en escena este modo de funcionamiento.



*Ilustración 16: Funcionamiento por sistema emisor-receptor*

El método de **sistema emisor-receptor** cambia la dinámica expuesta en los otros modos, ver Modos de funcionamiento, sección 4.2.1. En este caso contamos con dos dispositivos situados en lados opuestos. La detección de un objeto se produce cuando este interrumpe la comunicación entre estos dos.

La ilustración 17 representa el escenario en el cual podemos apreciar el modo de funcionamiento en cuestión.



*Ilustración 17: Funcionamiento por sistema emisor-receptor*

# 5 Actuadores

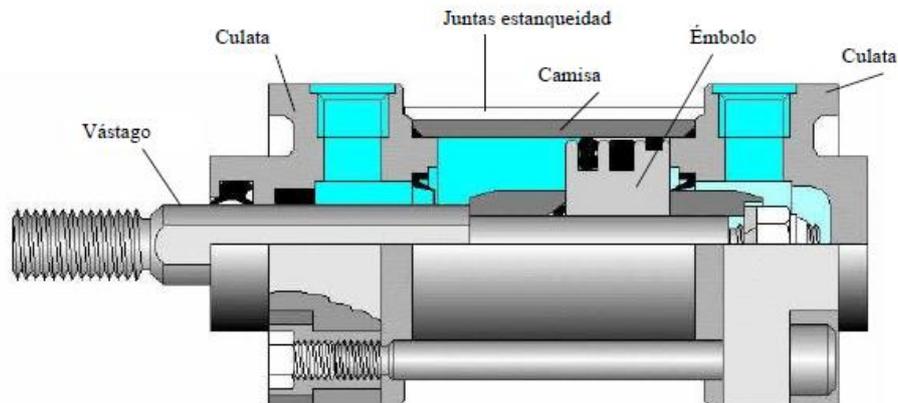
---

Los actuadores juntos a los sensores conforman los dos elementos más importantes de los cuales precisa un PLC. Los actuadores son activados a través de las salidas. La programación asociada a estos es a nivel lógico mediante un control Todo-nada y viene representada en la carpeta *Post* del *GRAFSET*. Es recomendable tener claro el concepto de este antes de explicar funcionamiento y características interna, ver Definiciones, sección 2.

Haciendo una inspección por la maqueta IPC-200, concretamente por la estación de llenado, identificamos dos tipos de actuadores:

1. Electroválvula neumática 5/2 monoestable.
2. Electroválvula fluidos 3/2 monoestable.

La mayoría de los actuadores se corresponden con cilindros neumáticos por lo que hemos visto oportuno adjuntar la ilustración 18, donde podemos ver el interior de estos para conocer la estructura y componentes básicos que tiene.



*Ilustración 18: Cilindro neumático (SMC International training , Manual de Prácticas, Análisis neumático, pág. 188)*

## 5.1 Análisis neumático

Comenzaremos analizando cada una de las características de los actuadores vistos en la inspección de la estación.

Ambos actuadores se corresponden con electroválvulas. Las electroválvulas resultan del acoplamiento de un sistema electromecánico (solenoido-electroimán de accionamiento) a una válvula de distribución neumática elemental, transformándola a una de accionamiento eléctrico.

Las válvulas distribuidoras son las encargadas de distribuir el aire comprimido para que tenga lugar el avance y retroceso de los cilindros.

Cabe destacar que aunque trabajemos con electroválvulas las cuales producen un accionamiento eléctrico, en el mercado contamos con válvulas de accionamiento manual, mecánico, neumático y por supuesto eléctrico.

Por otro lado podemos observar que tenemos electroválvulas de 5/2 y 3/2. La principal diferencia entre ambas es que la primera suele emplearse en cilindros de doble efecto, en cambio las electroválvulas de 3/2 suelen utilizarse en cilindros de simple efecto.

Para ver la composición de estos dos tipos de electroválvula hemos adjuntado dos ilustraciones. La ilustración 19 muestra una electroválvula de 3/2 o también conocidas como “de asiento” y es accionada unilateralmente con reposición por muelle. La válvula permanece abierta cuando se encuentra en reposo.

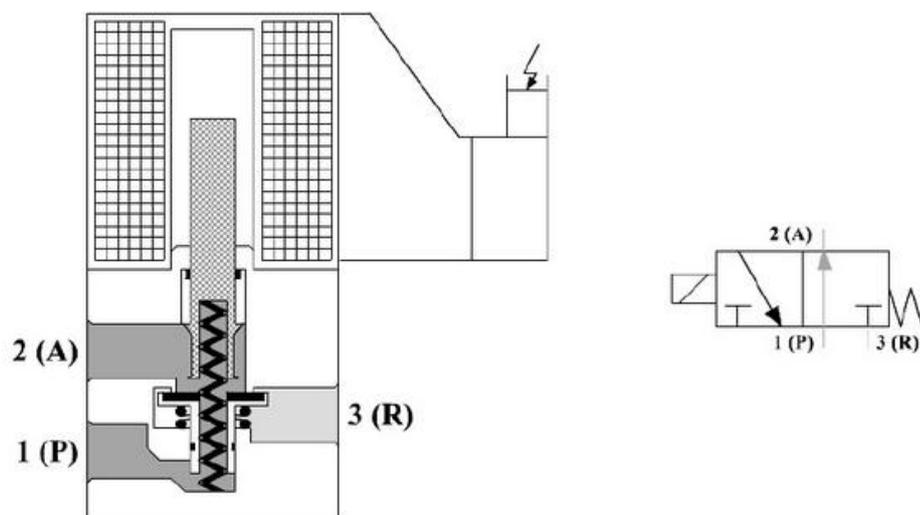


Ilustración 19: Electroválvula de 3/2

Por otro lado tenemos las electroválvulas de 5/2, que a diferencia de las electroválvulas con retorno por muelle, estas poseen un retorno por accionamiento eléctrico.

La ilustración 20 muestra una electroválvula de 5/2 por la parte interior para poder apreciar las distintas componentes.

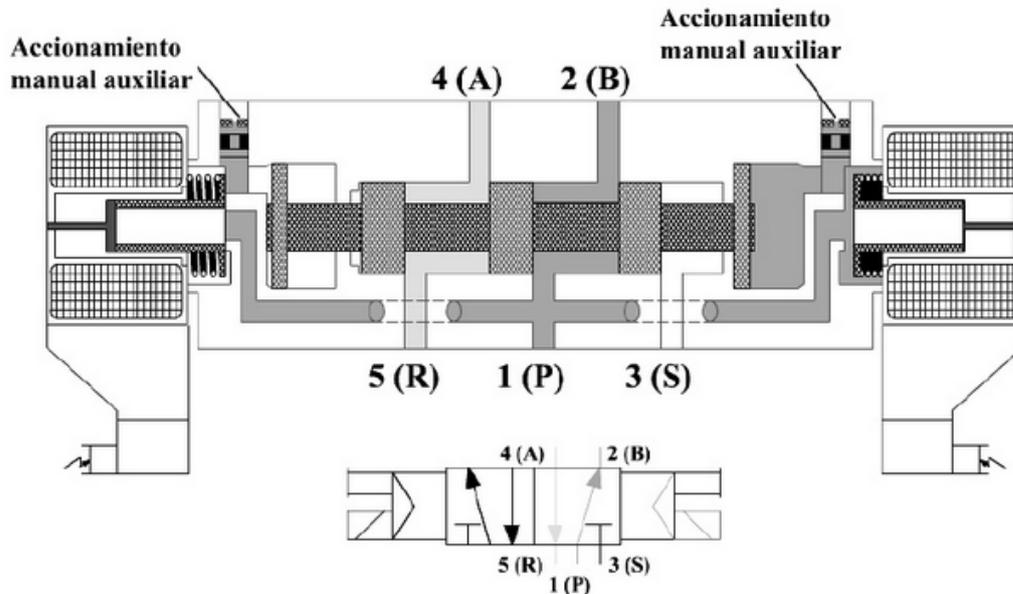


Ilustración 20: Electroválvula de 5/2

Aunque en nuestro sistema tengamos solamente electroválvulas de 3/2 y 5/2, en el mercado también podemos encontrar estas de 2/2 y 4/2 vías.

También cabe analizar el hecho de utilizar electroválvulas monoestables. Estas válvulas solo poseen una bobina para poder cambiar de estado, es decir, para que estas se desplacen y vuelvan a su origen, solo tenemos que activar y desactivar la señal de activación.

En contraposición nos encontramos con las válvulas biestables, las cuales poseen dos bobinas, una para cada cambio de estado. Por lo que la electroválvula conmuta al aplicar tensión en una bobina y al retirar la señal mantiene su estado de conmutación hasta la siguiente señal.

Unas de las principales ventajas de contar con válvulas monoestables es la simplificación a la hora de programar ya que a la hora controlar las acciones es más intuitivo contar con electroválvulas de este tipo, no obstante, podemos simular el comportamiento de electroválvulas biestables vía *software*.

## 6 Diseño e implementación

---

En esta sección hemos comenzado dando un enfoque computacional, donde se expone que modelo de diseño hemos escogido y que consecuencias ha tenido en el resultado final.

Seguidamente se ha realizado un análisis funcional de toda la estación diferenciando cada uno de los seis bloques funcionales:

1. Alimentador de botes
2. Llenador de botes
3. Alimentador de tapas
4. Prensador de tapas
5. Transportador de botes

También hemos analizado con todo detalle cómo ha sido programado nuestro *GRAF CET*, describiendo etapa a etapa cada elemento asociado al *Post*, así como, cada una de las transiciones de la aplicación.

Por otro lado hemos analizado los objetos predefinidos utilizados, exponiendo previamente las características esenciales de estos, como también las variables internas.

Hemos adaptado la botonera que llevaba incorporada la estación para dotarla de funcionalidad, con acciones como: *stop* (señal de emergencia), *Reset* (reseteado del proceso) o *Start* (ejecución del proceso).

Detalladamente hemos explicado cómo se ha implementado el inicio del proceso, el reseteado del mismo y sobre todo en que ha consistido el mecanismo de emergencia.

Una vez finalizada toda la implementación, observando los resultados se ha propuesto una mejora a nivel teórico.

## 6.1 Estructuración y diseño

En este apartado hemos estudiado y planteado el diseño que íbamos a adoptar en nuestro *GRAF CET*. Observando la estación y considerando que el proyecto es de carácter industrial, un factor muy importante era la productividad, por lo que se introdujo el concepto de paralelización. Partiendo de conocimientos obtenidos en asignaturas como *Computación paralela*, se procedió a realizar una descomposición de tareas, identificando los patrones de concurrencia.

Partiendo de este modelo de programación surgieron los distintos bloques funcionales, es decir, tareas totalmente independientes las cuales podían ejecutarse de forma paralela.

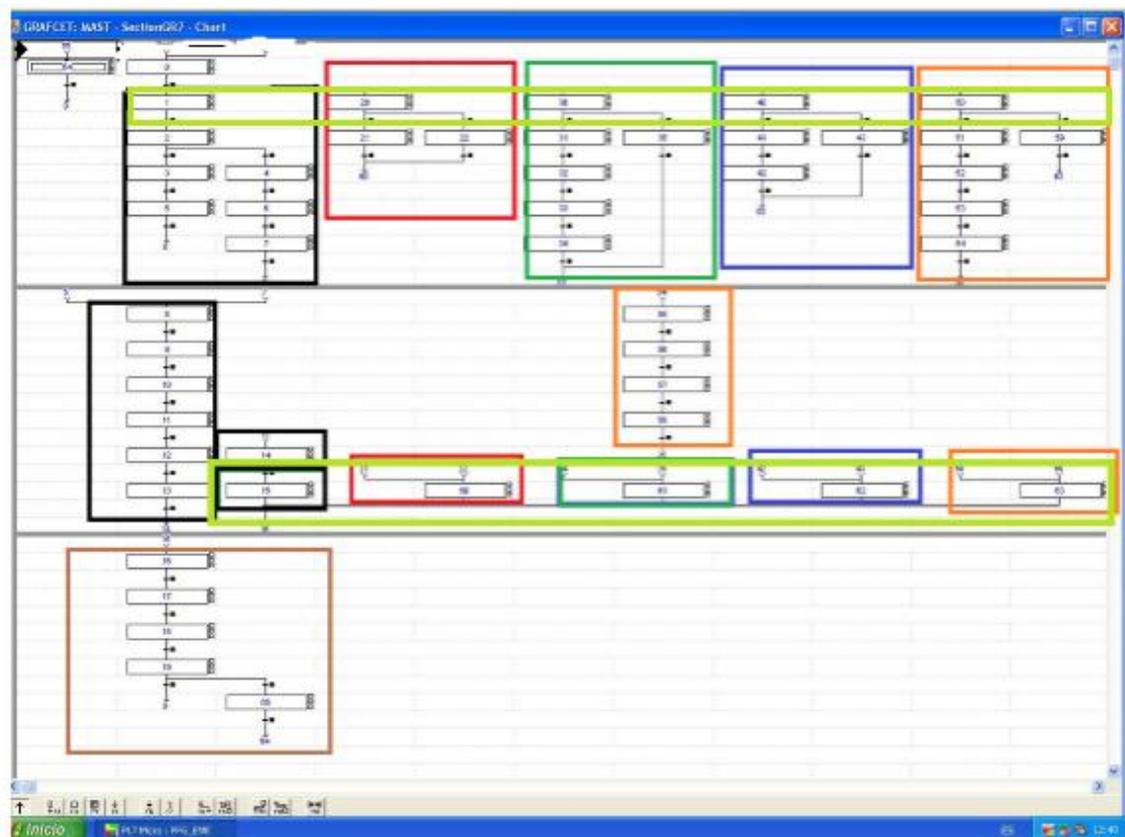


Ilustración 21: Bloques funcionales

La ilustración 21, muestra el *GRAF CET*, con los distintos bloques diferenciados e identificados. Cada uno de estos bloques se corresponde con un bloque funcional en la cadena de producción. La correspondencia es la siguiente:

- El bloque **negro** se corresponde con el proceso de alimentación de botes.
- El bloque **rojo** se corresponde con el proceso de llenado de botes.
- El bloque **verde** se corresponde con el proceso de alimentación de tapas.
- El bloque **azul** se corresponde con el proceso de prensado de tapas.
- El bloque **naranja** se corresponde con el proceso de transporte de botes.
- El bloque **marrón** se corresponde con el proceso de rotación de la mesa.

Los **rectángulos** de color verde representan la divergencia y convergencia en *AND* respectivamente.

Una vez bien diferenciados los distintos bloques, podemos afirmar que tanto el proceso correspondiente a la alimentación de botes como el de transporte de botes parecen ser los más costosos de efectuar. Sin embargo, el proceso del transportado de botes no genera un coste adicional en comparación a los otros bloques.

No ocurre lo mismo con el bloque correspondiente a la alimentación de botes. En este caso, esta tarea es bastante costosa en comparación a las otras y no ha podido dividirse más, por lo que el coste total por iteración viene dado por el tiempo que tarda esta tarea en completarse más el coste de la última tarea, ya que el tiempo que tardan las otras tareas en realizar una iteración se solapa con el tiempo que tarda esta.

También podemos observar en la ilustración 21 que hay una banda color verde claro. Esta banda se corresponde con un punto de sincronización entre los cinco primeros bloques, debido a que la última tarea correspondiente con el proceso de rotación de la mesa es totalmente dependiente de las otras cinco tareas, y solo se llevará a cabo una vez los otros bloques hayan finalizado su iteración.

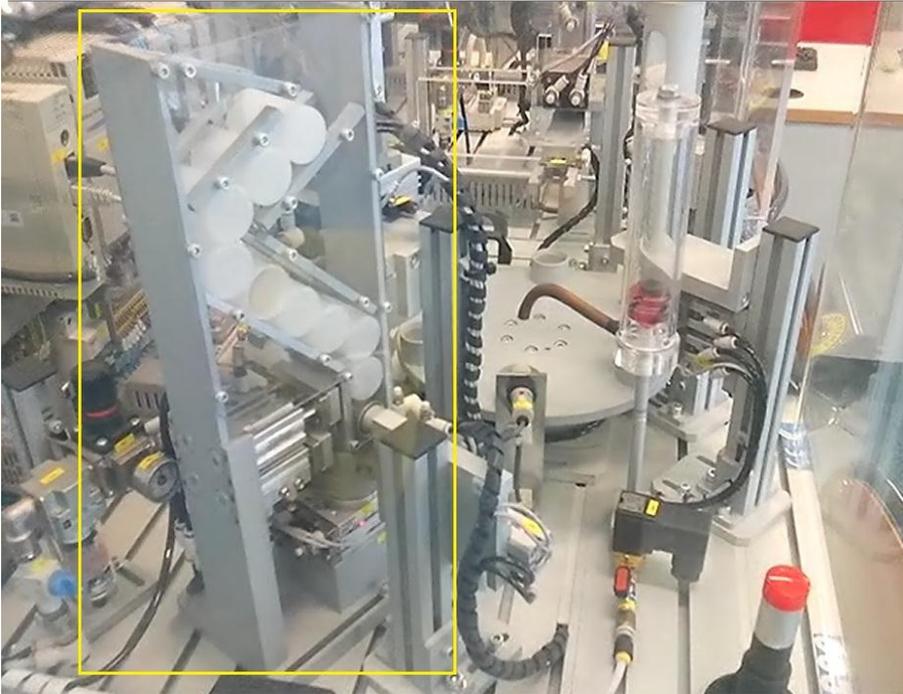
### 6.1.1 Análisis funcional

En esta sección vamos a analizar con detalle el funcionamiento de la estación de llenado, observando a través de ilustraciones cada uno de los bloques funcionales y explicando para cada uno de ellos las acciones que se llevan a cabo, sería recomendable revisar todo lo relacionado con sensores y actuadores para una mejor comprensión ver Sensores y Actuadores, secciones 4 y 5 respectivamente.

#### 6.1.1.1 Alimentado de botes

En esta sección se ha analizado minuciosamente el proceso de alimentación de botes. Comenzaremos observando la ilustración 22, que muestra el alimentador de botes completo. Como podemos observar los botes están dispuestos en *Zig-Zag*, como sabemos, ver Estación de llenado, sección 3.2.2, se trata de un alimentador gravitacional controlado por el alimentador de botes.

El alimentador de botes consta de dos cilindros compactos con guías Ø12, C: 10 mm (MGPM12-10), con reguladores de caudal y controlados por una única electroválvula 5/2 monoestable. Ambos cilindros se sitúan siempre en posiciones contrarias, de modo que cuando el cilindro inferior libera el último bote del cargador, el resto son retenidos por el cilindro superior.



*Ilustración 22: Alimentador de botes completo*

En la ilustración 23 podemos observar el cilindro que controla la orientación del bote y el cilindro empujador de botes.

Una vez el bote ha sido suministrado, debemos comprobar la orientación del bote. Para ello contamos con un cilindro compacto de doble efecto Ø12, C: 10 mm (CDQ2B12-15D), con reguladores de caudal y detección de posición final, controlado por una electroválvula 5/2 monoestable.

Para realizar la comprobación, el cilindro avanza introduciendo una pieza cilíndrica en el bote, de forma que si el bote se encuentra invertido, esta pieza topará con la base de este impidiendo que el cilindro complete su carrera y se active el detector magnético colocado en el cilindro.

En el caso de que el cilindro haya completado su carrera, significará que el bote está al revés y precisa de un giro de 180° antes de su empuje.

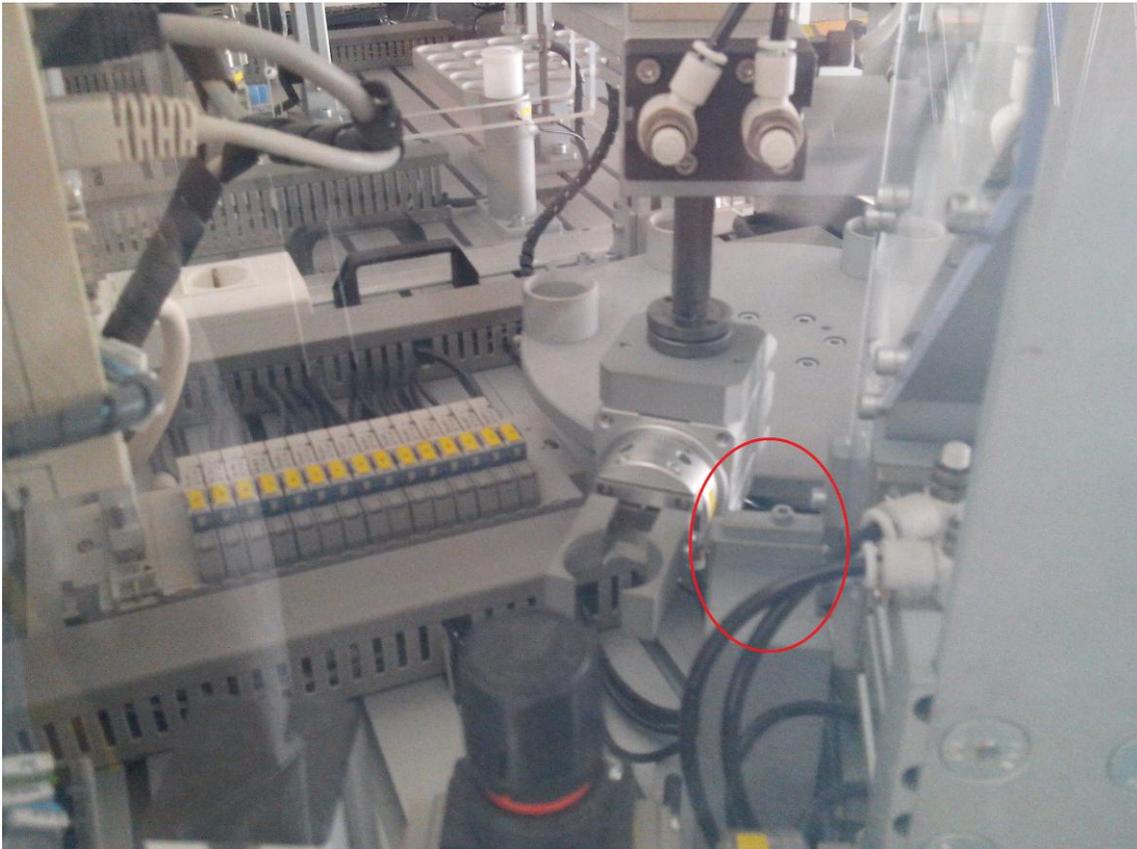


*Ilustración 23: Alimentador de botes (vista inferior)*

En el caso en el que tengamos que rotar el bote contamos con un actuador de giro tipo piñón-doble cremallera  $\text{Ø}15$ ,  $\dot{\alpha}_{\text{max}}$ :  $180^\circ$  (MSQB10A), con reguladores de caudal y detección de las posiciones  $0^\circ$  y  $180^\circ$ , controlado por una electroválvula 5/2 monoestable. Así que simplemente habría que activar este actuador hasta alcanzar una posición de  $180^\circ$ .

A continuación el bote deber ser desplazado para depositarlo en la zona de carga para ser puesto en la mesa giratoria. Para efectuar el desplazamiento contamos con un cilindro compacto de doble efecto  $\text{Ø}12$ , c: 75 mm (CDQSB12-75CM), con reguladores de caudal y detección de las posiciones inicial y final, controlado por una electroválvula 5/2 monoestable. Este deberá únicamente deberá empujar el bote hasta el punto de carga que se corresponde con la posición final del cilindro.

Para no perder la referencia la ilustración 24 nos ubica en el punto del proceso en el que nos encontramos. Como podemos observar la imagen se corresponde con la parte trasera del alimentador de botes y el círculo nos indica el punto de carga.



*Ilustración 24: Alimentador de botes (Vista trasera)*

Una vez el bote se encuentra en el punto de carga, este debe ser insertado en la mesa giratoria. Para ello contamos con un cilindro compacto de movimiento lineal y rotativo  $\text{Ø}32$ , c: 50 mm (MRQBS32-50CB), con reguladores de caudal y detección de posición inicial y final en el movimiento lineal y de  $0^\circ$  a  $180^\circ$  en el rotativo. En este caso contamos con dos electroválvulas 5/2 monoestables para su actuación.

Para la sujeción y giro de los botes, contamos con una pinza neumática giratoria de doble efecto con los dedos de apertura paralela (MRHQ16D90S-N) y un acoplamiento rotolineal (ETP-12-LENZE).

Para realizar el transporte la pinza debe rotar el bote  $90^\circ$  y posteriormente realizar un giro de  $180^\circ$  para depositar el bote en el primer puesto de la mesa giratoria.

Este último paso podemos comprenderlo mejor observando la ilustración 25 donde vemos la pinza depositando el bote en su lugar correspondiente.



*Ilustración 25: Depósito de bote en mesa giratoria*

#### **6.1.1.2 Llenado de botes**

En este puesto se procede al llenado y detección del bote. Para ello es necesario permitir el paso de fluido a esta estación, para ello contamos con una electroválvula de fluido monoestable (VCW21-5D-3-01F-Q) que controla el flujo de fluido, mientras que la presencia de botes en este puesto es controlada por un sensor fotoeléctrico (E3F2-DS30B4).

Vía software podremos controlar el tiempo y nivel de apertura de la válvula de entrada de fluido a la estación.

En la ilustración 26 podemos observar el puesto de llenado de botes, hemos remarcado con un círculo rojo el sensor de detección para que no pase desapercibido.



*Ilustración 26: Estación de llenado*

### **6.1.1.3 Alimentado de tapas**

En este puesto al bote le será suministrada una tapa que se le dejará caer encima. Para ello contamos con dos pinzas neumáticas con guías integradas de doble efecto con dos dedos de apertura paralela  $\varnothing 16$ ,  $\dot{\iota}$  (MHZ2-16D), con detección de las posiciones de abierta y cerrada, controladas por una única electroválvula 5/2 monoestable. Las pinzas están siempre en posiciones contrarias, de forma que solo pueda ser suministrada una tapa por ciclo.

El modo de trabajo es prácticamente igual que en el caso del alimentado de botes, en este caso, la pinza superior se cerrará sujetando todas las tapas menos la que se encuentra inmediatamente en la parte inferior. Esta caerá cuando la pinza inferior sea abierta.

En la ilustración 27 podemos observar detalladamente el puesto alimentador de tapas, identificando el alimentador propiamente dicho y las pinzas superior e inferior.



*Ilustración 27: Alimentador de tapas*

#### **6.1.1.4 Prensado de tapas**

En este puesto se efectuará presión sobre la tapa depositada previamente sobre un bote determinado. Para ello contamos con un cilindro compacto de doble efecto Ø20, C: 30 mm (CDQ2B20-30D), con reguladores de caudal y detector de posición inicial y final, gobernado por una electroválvula 5/2 monoestable.

Cuando el bote sea depositado bajo este puesto, el cilindro compacto descenderá ejerciendo cierta presión sobre el la tapa provocando el cierre hermético del bote.

La ilustración 28 muestra una imagen donde podemos apreciar los detalles de este puesto.



*Ilustración 28: Prensado de tapas*

### 6.1.1.5 Transporte de botes

Esta es la etapa final del proceso donde el bote es transportado al almacén. Para ello contamos con un cilindro compacto de movimiento lineal y rotativo Ø32, c: 50 mm (MRQBS32-50CB), con reguladores de caudal y detección de posición inicial y final en el movimiento lineal y de 0° a 180° en el rotativo. En este caso contamos con dos electroválvulas 5/2 monoestables para su actuación.

Para la sujeción y giro de los botes, contamos con una pinza neumática giratoria de doble efecto con los dedos de apertura paralela (MRHQ16D9oS-N) y un acoplamiento rotolineal (ETP-12-LENZE).

A diferencia que en el caso del transporte del bote a la mesa giratoria, ver Diseño e implementación, sección 6.1.1.1, no se precisa de ninguna rotación del bote, tan solo que la pinza cierre y efectúe el giro y el depósito.

En la ilustración 29 podemos apreciar este último puesto donde hemos marcado la zona de carga y descarga para una mejor comprensión de lo explicado.

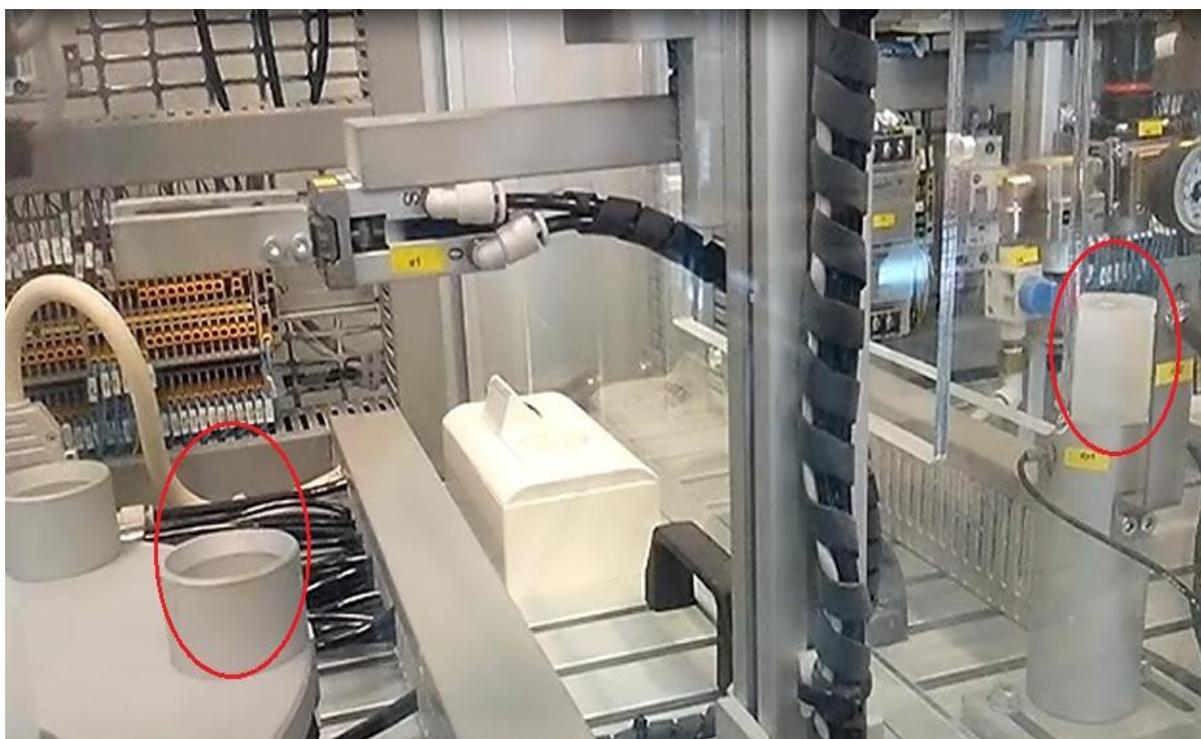


Ilustración 29: Transporte de botes

#### 6.1.1.6 Giro de mesa

Este puesto es el encargado de ir rotando la mesa giratoria en cada iteración del programa. Para ello este cuenta con un cilindro compacto de doble efecto Ø25, c: 50 mm (CDQ2B25-50D) con reguladores de caudal y detección de la posición inicial controlado por una electroválvula 5/2 monoestable y dos cilindros de bloqueo compactos de doble efecto Ø16, c: 10 mm (CQ2B16-10D) controlados por electroválvulas 5/2 monoestables.

La mesa giratoria se utiliza como sistema de movimiento giratorio, de forma que a cada movimiento de giro se produce un avance de un número de grados determinados, resultado de dividir 360° entre el número de puestos definidos. Para llevar a cabo este movimiento se utiliza un cilindro neumático empujador. A su vez disponemos de otros dos cilindros, uno de ellos sujeta el plato mientras está en reposo y el otro que lo sujeta cuando la mesa está en movimiento.

En la ilustración 30 podemos observar cómo está constituida la mesa giratoria con el fin de facilitar la comprensión.



Ilustración 30: Giro de mesa

## 6.2 Implementación

En esta sección hemos analizado cada bloque funcional por separado, pasando por cada una de las etapas y transiciones con el fin de justificar la implementación realizada así como el uso de objetos predefinidos y variables internas.

Antes de comenzar con la explicación es necesario saber que el *PLC* cuenta con canales de 16 entradas y 12 salidas. Las tablas 11 y 12 muestran las correspondencias entre los sensores, actuadores y entradas, salidas del *PLC*.

	ACCION	SIMBOLO	TIPO	I/O
ENTRADAS 24	Pulsador marcha	m	N.A.	0
	Pulsador paro	p	N.C.	1
	Selector Auto/Man	A/M	N.A.	2
	Pulsador reset	r	N.A.	3
	Giro plato atrás	a0	Detector magnético reed	4
	Extracción bote arriba	c0	Detector magnético reed	5
	Extracción bote abajo	c1	Detector magnético reed	6
	Extracción bote origen	d0	Detector magnético reed	7
	Extracción bote fin	d1	Detector magnético reed	8
	Pinza extracción cerrada	e1	Detector magnético estado sólido PNP	9
	Pinza 1 tapa abierta	f0	Detector magnético estado sólido PNP	10
	Pinza 2 tapa cerrada	f1	Detector magnético estado sólido PNP	11
	Prensa arriba	g0	Detector magnético reed	12
	Prensa abajo	g1	Detector magnético reed	13
	Introducción bote arriba	h0	Detector magnético reed	14
	Introducción bote abajo	h1	Detector magnético reed	15
	Introducción bote origen	i0	Detector magnético reed	16
	Introducción bote fin	i1	Detector magnético reed	17
	Control posición bote adelante	l1	Detector magnético reed	18
	Empujador botes adelante	m0	Detector magnético reed	19
	Empujador botes atrás	m1	Detector magnético reed	20
	Giro bote origen	n0	Detector magnético reed	21
	Giro bote final	n1	Detector magnético reed	22
Detector botes	dp1	Sensor fotoeléctrico 3 hilos PNP	23	

Tabla 11: Entradas del PLC (SMC International training, Manual de Prácticas, Análisis por bloques, pág. 153)

	ACCION	SIMBOLO	TIPO	I/O
SALIDAS 17	Baliza roja	BR	Piloto luminoso de color rojo	0
	Avance giro plato	A+	EV neumática 5/2 monoestable	1
	Avance bloqueo y arrastre	B+	EV neumática 5/2 monoestable	2
	Avance lineal extracción bote	C+	EV neumática 5/2 monoestable	3
	Avance giro extracción bote	D+	EV neumática 5/2 monoestable	4
	Apertura pinza extracción bote	E+	EV neumática 5/2 monoestable	5
	Apertura alimentador tapas	F+	EV neumática 5/2 monoestable	6
	Avance prensa	G+	EV neumática 5/2 monoestable	7
	Avance lineal introducción bote	H+	EV neumática 5/2 monoestable	8
	Avance giro introducción bote	I+	EV neumática 5/2 monoestable	9
	Apertura pinza introducción	J+	EV neumática 5/2 monoestable	10
	Avance alimentador botes	K+	EV neumática 5/2 monoestable	11
	Avance control posición bote	L+	EV neumática 5/2 monoestable	12
	Avance empuje salida bote	M+	EV neumática 5/2 monoestable	13
	Giro bote al revés	N+	EV neumática 5/2 monoestable	14
	Llenado bote	A4+	EV fluidos monoestable	15
	Cierre alimentador tapas	O+	EV neumática 5/2 monoestable	16

Tabla 12: Salidas del PLC (SMC International training, Manual de Prácticas, Análisis por bloques pág. 154)

No obstante, con esta información no podemos saber la correspondencia entre las entradas y salidas del PLC y las de nuestro programa, por tanto, hemos adjuntado tablas 13 y 14 para que quede todo bien conectado.

<b>Acción</b>	<b>I/O</b>	<b>Entrada</b>
Pulsador marcha	0	%I1.0
Pulsador paro	1	%I1.1
Selector Auto/ <u>Man</u>	2	%I1.2
Pulsador <u>reset</u>	3	%I1.3
Giro plato atrás	4	%I1.4
Extracción bote arriba	5	%I1.5
Extracción bote abajo	6	%I1.6
Extracción bote origen	7	%I1.7
Extracción bote fin	8	%I1.8
Pinza extracción cerrada	9	%I1.9
Pinza 1 tapa abierta	10	%I1.10
Pinza 1 tapa cerrada	11	%I1.11
Prensa arriba	12	%I1.12
Prensa abajo	13	%I1.13
Introducción bote arriba	14	%I1.14
Introducción bote abajo	15	%I1.15
Introducción bote origen	16	%I3.0
Introducción bote fin	17	%I3.1
Control posición bote adelante	18	%I3.2
Empujador botes adelante	19	%I3.3
Empujador botes atrás	20	%I3.4
Giro bote origen	21	%I3.5
Giro bote final	22	%I3.6
Detector botes	23	%I3.7

Tabla 13: Correspondencia Entradas

<b>Acción</b>	<b>I/O</b>	<b>Salida</b>
Baliza roja	0	%Q2.0
Avance giro plato	1	%Q2.1
Avance bloqueo y arrastre	2	%Q2.2
Avance lineal extracción bote	3	%Q2.3
Avance giro extracción bote	4	%Q2.4
Apertura pinza extracción bote	5	%Q2.5
Apertura alimentador tapas	6	%Q2.6
Avance prensa	7	%Q2.7
Avance lineal introducción bote	8	%Q2.8
Avance giro introducción bote	9	%Q2.9
Apertura pinza introducción	10	%Q2.10
Avance alimentador botes	11	%Q2.11
Avance control posición bote	12	%Q4.0
Avance empuje salida bote	13	%Q4.1
Giro bote al revés	14	%Q4.2
Llenado bote	15	%Q4.3
Cierre alimentador tapas	16	%Q4.4

Tabla 14: Correspondencia Salidas

### 6.2.1 Etapa inicial

El *GRAF CET* presenta una etapa inicial la cual solo es ejecutada una vez en todo el proceso industrial debido a que esta es la que se encarga de poner en marcha todo el mecanismo. A partir de la siguiente iteración tenemos una etapa que actúa de etapa inicial que a diferencia de la original no tiene que arrancar todo el proceso ni realizar todas las operaciones que realiza esta, lo cual libera de carga computacional al PLC y de operaciones innecesarias.

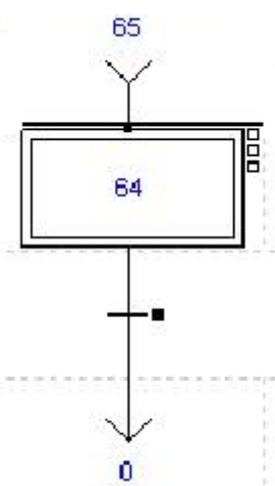


Ilustración 31: *GRAF CET*, Etapa inicial

La ilustración 31 muestra la etapa inicial original, a partir de esta explicaremos su funcionamiento detalladamente.

Al haber declarado esta etapa como inicial, al pulsar el *Run* sobre el *PL7 Micro* automáticamente entra en ejecución.

Como podemos apreciar, le precede la etapa 65. Esta etapa solo es accesible si se ha pulsado en algún momento el botón de *Reset*, ver Funcionalidad de la botonera, sección 6.2.10.2.

Hemos adjuntado la ilustración 32, correspondiente al *Post* de la etapa inicial.

Como vemos hay una entrada adicional correspondiente a la variable interna **%M81**, utilizada para la reanudación del sistema ante un *stop* por emergencia, ver Funcionalidad de la botonera, sección 6.2.10.3.

A la parte derecha se resetean distintas variables del sistema, cabe destacar que aquellas que reciban el nombre **%M3x** están obsoletas, ya que formaban parte de una solución del problema alternativa.

La variable **%M92** es activada cuando el botón de *Reset* se pulsa, por lo que en esta etapa limpiamos su valor, para cuando volvamos a pulsar *Start* esta variable tenga su valor inicial '0'.

Por otro lado realizamos un *Set* a la salida **%Q4.5**, esta salida es la encargada de cerrar el circuito que conecta las distintas entradas y salidas a la corriente eléctrica.

Por último hacemos un *Reset* a la salida **%Q2.0**, esta salida se corresponde con la baliza roja luminosa, la cual participa cuando entramos en situación de emergencia, ver Funcionalidad de la botonera, sección 6.2.10.2 y por tanto al reanudar el funcionamiento habitual esta debe desactivarse.

La transición a la etapa 0 se produce cuando se detecta que el botón de *Start* se ha pulsado.



Ilustración 32: Post, Inicialización del proceso

Hemos hablado de una etapa “inicial”, la cual toma el rol de etapa inicial real a partir de la segunda iteración. En la ilustración 34 podemos ver el aspecto de esta.

Tiene dos entradas procedentes de las etapas 19 y 64. La etapa 19 es la última etapa del proceso, esta se corresponde con la finalización de la rotación de la mesa giratoria.

Por otro lado la etapa 64 hemos visto que se corresponde con la etapa inicial.

La ilustración 33 muestra el fragmento del *POST* donde esta etapa interviene.

Como podemos observar la etapa 0 realiza un *Reset* del contador %C0, el cual está asociado al bloque funcional de alimentado de botes, ver Objetos predefinidos, variables internas del sistema, sección 6.2.9.2.

La transición hacia el resto de etapas está vacía ya que no se requiere ninguna condición.

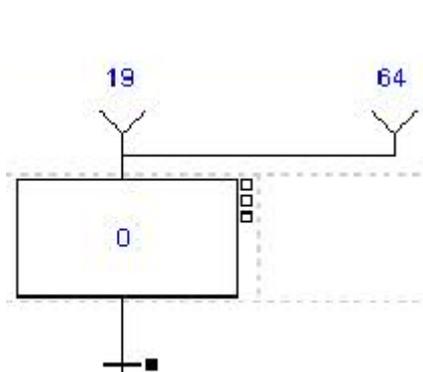


Ilustración 33: GRAFCET, Etapa “inicial”

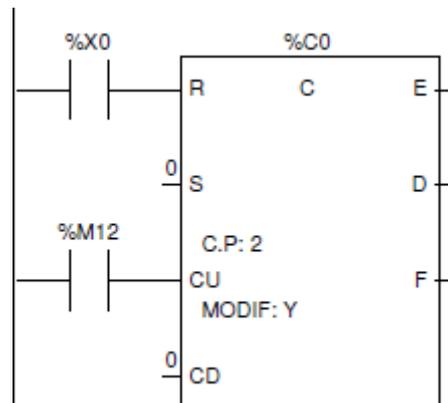


Ilustración 34: Post, Etapa 0

## 6.2.2 Etapas 1-14

En esta sección se ha analizado en bloque funcional correspondiente al alimentado de botes, ver Análisis funcional, sección 6.1.1.1.

En la ilustración 35 y 36 podemos observar el esquema *GRAF CET* de este bloque funcional.

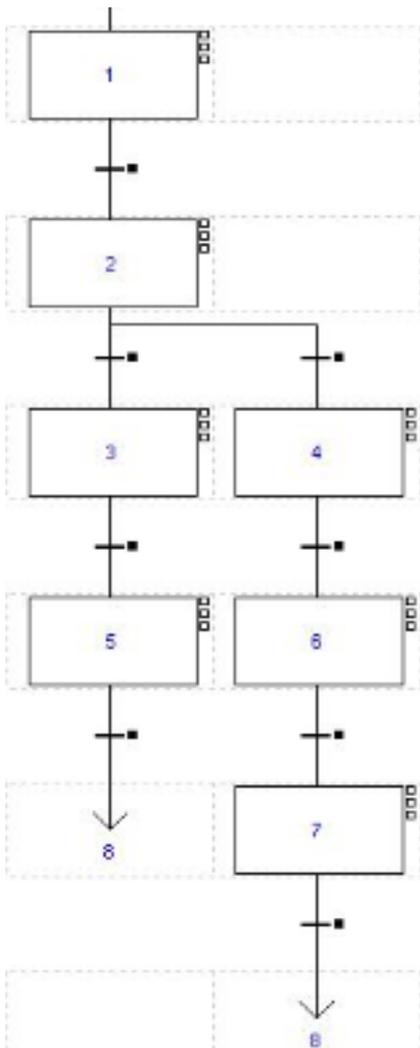


Ilustración 36: *GRAF CET*, Alimentado de botes I

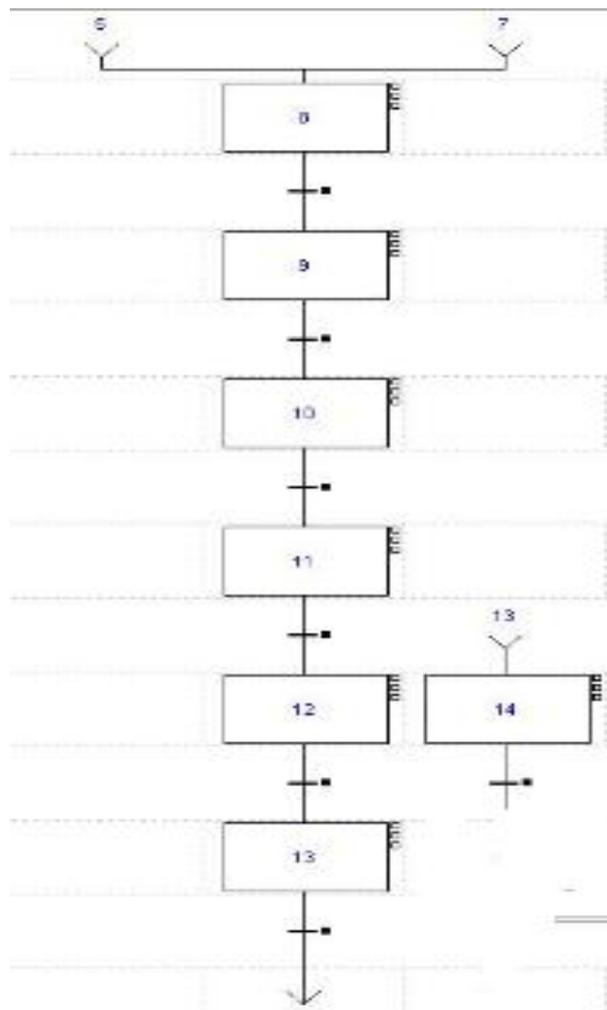


Ilustración 35: *GRAF CET*, Alimentador de botes II

### 6.2.2.1 Etapa 1

En esta etapa hemos activado la salida **%Q2.11**, correspondiente al avance alimentador botes (**K+**), de forma que el bote que estuviese en la parte inferior del alimentador caería.

Debido a que no tenemos ningún sensor asociado a esta salida, decidimos implementar el temporizador **%TMO** para asegurarnos de que la acción se realizase correctamente, de forma que una vez la acción ha comenzado a efectuarse debería transcurrir el tiempo programado en el temporizador para que se efectuase la transición.

En la ilustración 36 podemos observar el fragmento de código del *Post* correspondiente a la etapa 1.



Ilustración 37: Post, Avance alimentador botes

### 6.2.2.2 Etapa 2

En esta etapa hemos activado la salida **%Q4.0**, correspondiente al avance control posición bote (**L+**). Hemos realizado un *Set* sobre la salida de forma que aunque transitemos de un estado a otro la salida sigue estando activa a menos que realicemos un *Reset* sobre esta.

El resultado de aplicar **L+** es provocar que el cilindro avance para realizar la comprobación de orientación del bote.

Aunque tengamos asociado el sensor **I1**, correspondiente al control de posición adelante, consultar el valor antes de que la acción haya finalizado podría proporcionarnos una información inválida, por lo que la decisión que optamos por tomar fue incorporar el temporizador **%TM1** para que una vez empezase el cilindro a avanzar tuviera un margen de tiempo para completar su acción.

Una vez el temporizador venza, se realizará la comprobación del valor que tenga el sensor asociado y realizaremos la transición a la etapa 3 o 4 en función del valor lógico proporcionado.

En la ilustración 38 podemos observar el fragmento de código del *Post* correspondiente a la etapa 2.

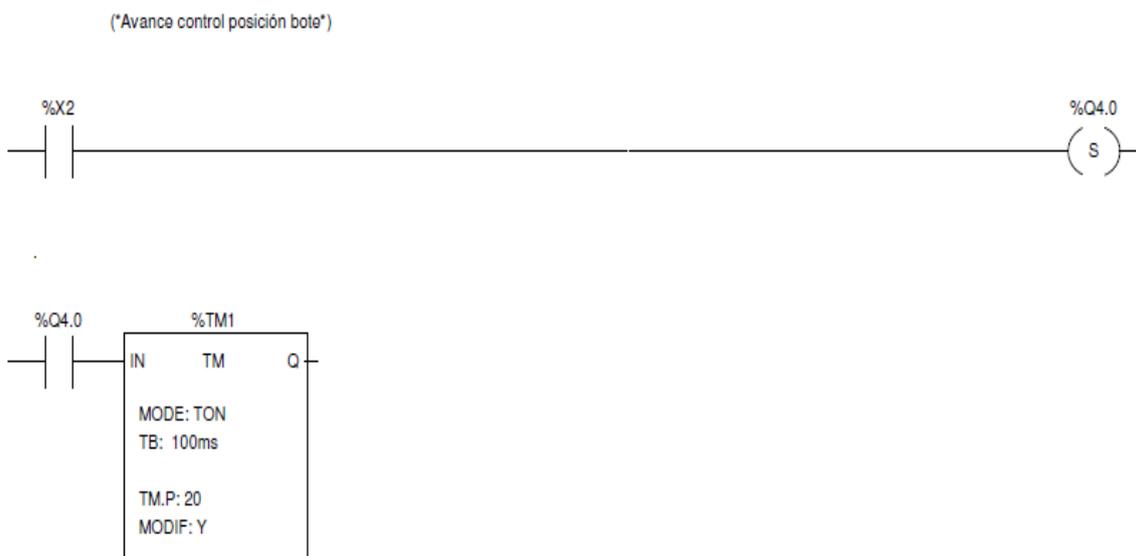


Ilustración 38: Post, Avance control posición (Set)

### 6.2.2.3 Etapa 3 y 6

Ambas etapas tienen el mismo cometido, empujar el bote hacia el punto de carga, a través de la salida **%Q4.1**, Avance empuje salida bote (**M+**). Por otro lado realizar un *Reset* sobre la salida **%Q4.0** a través de la variable interna **%M5** activada tras la activación de la etapa **%X3** para que el cilindro encargado de comprobar la orientación del bote vuelva a su posición de inicio.

La diferencia entre ambas etapas es que la etapa 3, es la etapa que gestiona los botes con la orientación correcta mientras que la etapa 6 gestiona los botes los cuales estaban al revés, por lo que previamente estos han sido rotados 180°, ver Etapas 1-14, sección 6.2.2.4.

De cara a la transición utilizamos la información proporcionada por los sensores **m0** y **m1**, ambos controlan el inicio y fin de carrera del cilindro.

Para asegurar que el bote está en el sentido correcto antes del empuje, hemos añadido el temporizador **%TM2**, que recibe como entrada el valor de la variable interna **%M2**, activada tras la activación de la etapa **%X3**, o la activación de la etapa **%X6**.

En la ilustración 39 y 40, podemos observar el fragmento del *Post* correspondiente a estas dos etapas. Para conocer el funcionamiento de la variable interna **%M6**, ver Implementación, sección 6.2.2.4.

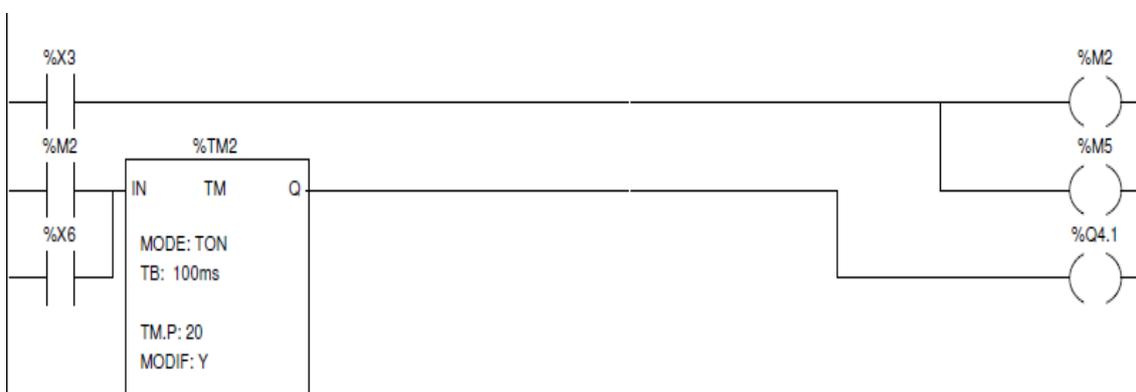


Ilustración 39: Post, Avance empuje salida bote



Ilustración 40: Post, Avance control posición bote (Reset)

#### 6.2.2.4 Etapa 4

Esta etapa se encarga de procesar los botes que están colocados en el sentido inverso, concretamente de rotar estos 180° activando la salida **%Q4.2**, Giro bote al revés (**N+**) y realizar un *Reset* sobre la salida **%Q4.0**, correspondiente al cilindro que verifica la orientación del bote. Para ello hemos hecho uso de dos variables internas: **%M3** y **%M6** y del temporizador **%TM4**.

Inmediatamente tras activarse la etapa **%X4**, activamos las variables **%M3** y **%M6**. La primera de estas se encarga de activar a **%TM4**. El uso de este temporizador se debe a una medida de precaución, de esta forma aseguramos que el bote ha caído correctamente sobre el elemento de giro. Por lo que una vez vencido el tiempo activamos **%Q4.2** para que se efectúe el giro.

La variable **%M6** se encargará de realizar un *Reset* sobre la salida **%Q4.0**, de forma que este volverá a su posición inicial.

La transición se producirá cuando **n1**, sensor que detecta que el elemento giratorio se encuentra en la posición de 180° este activo, es decir, que el bote haya sido rotado correctamente.

En la ilustración 40,41 podemos observar el fragmento de código del *Post* correspondiente a la etapa 4, para más detalles de la variable interna **%M5**, ver Implementación, sección 6.2.2.3.

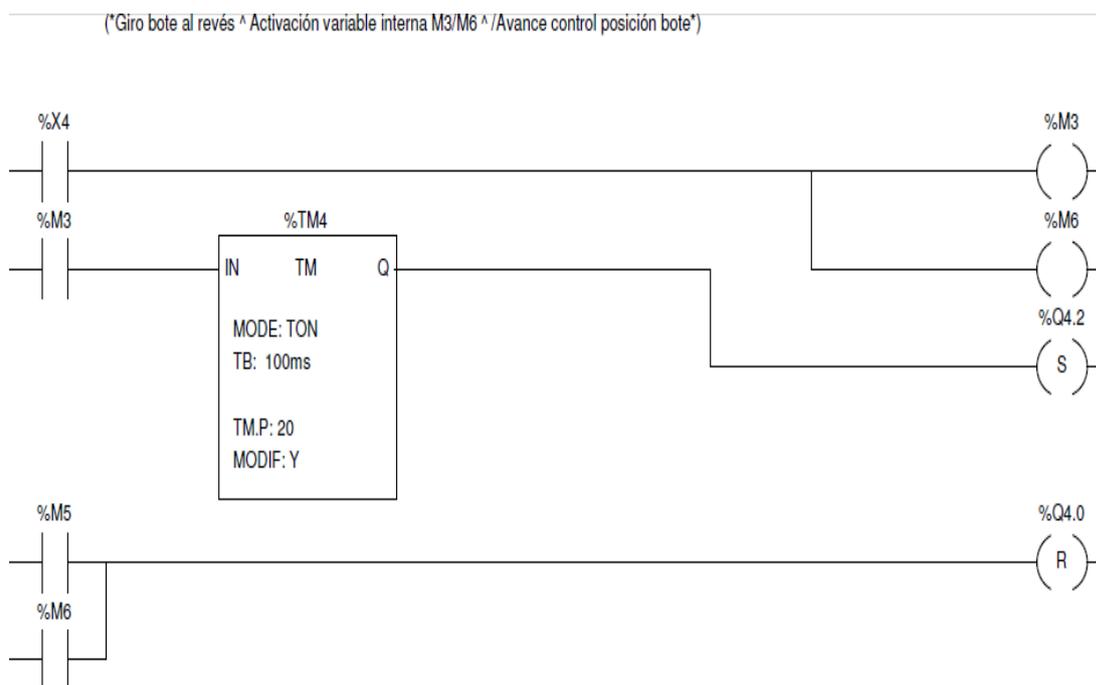


Ilustración 41: Post, Giro bote al revés (Set) y Avance control posición (Reset)

### 6.2.2.5 Etapa 5 y 7

Ambas etapas comparten funcionalidad, pero cada una gestiona un tipo de botes. La etapa 5 trabaja con los botes orientados correctamente, en contraposición la etapa 7 opera con los botes que estaban colocados al revés.

Estas etapas se encargan de cerrar la pinza para dejar bien sujeto el bote, realizando un *Set* sobre la salida **%Q2.10**, correspondiente a apertura pinza introducción (**J+**), para que esta quede cerrada hasta que hagamos un *Reset* sobre ella.

Por otro lado hemos realizado un *Reset* sobre la salida **%Q4.2**, para que el soporte encargado de girar los botes que tenían una orientación incorrecta volviese a su posición de inicio.

En estas etapas hemos utilizado aparte dos temporizadores: **%TM8** y **%TM3** y la variable interna **%M4**.

El **%TM8** lo hemos utilizado para asegurarnos de que el bote había llegado a la zona de carga.

El **%TM3** fue necesario implementarlo debido a que no teníamos un sensor asociado a la pinza, por lo que una forma de asegurar que la pinza había terminado de cerrarse era dándole un tiempo suficiente para que terminase su función.

La transición asociada a ambas etapa venía dada por el vencimiento del temporizador **%TM3**.

La ilustración 42 se corresponde con un fragmento del *Post* correspondiente a las etapas 5 y 7.

### 6.2.2.6 Etapa 8 y 12

(\*Apertura pinza introducción ^ Activación variable interna M4 ^ /Giro bote final\*)

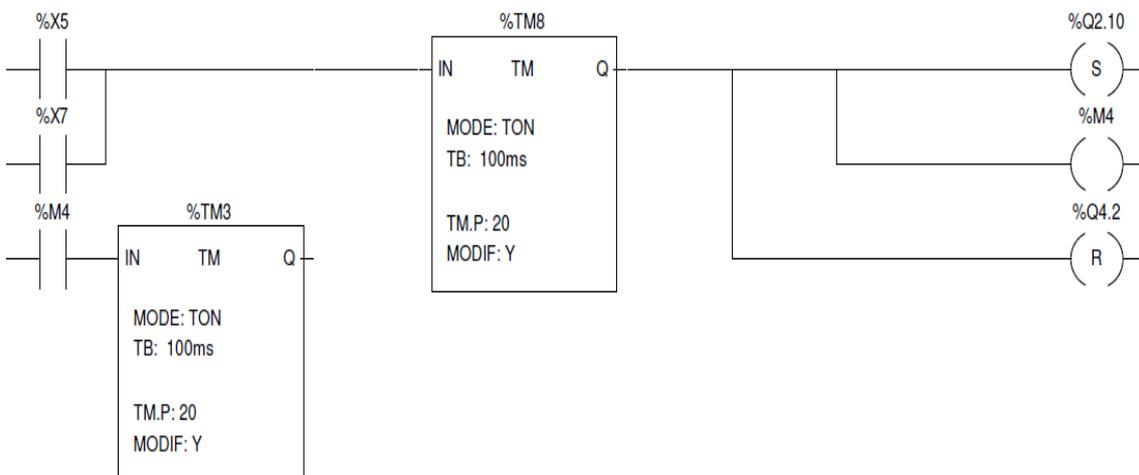


Ilustración 42: Post, Apertura pinza introducción (Set) y Giro bote al revés (Reset)

En la etapa 8 se unifican las dos ramas originadas tras la etapa 2, la que procesaba los botes correctamente orientados y la que procesaba los botes que estaban en sentido opuesto.

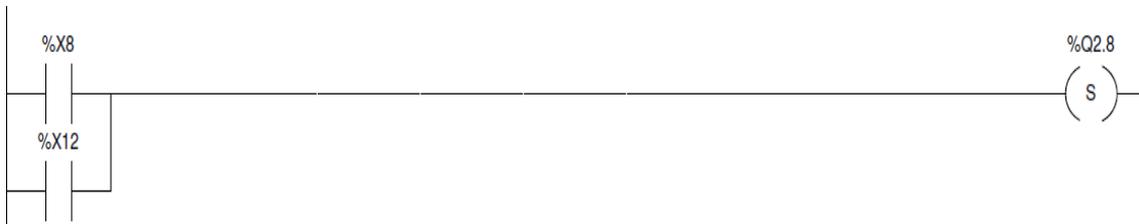
Ambas etapas activan la salida **%Q2.8**, correspondiente con Avance lineal introducción bote (**H+**), pero en situaciones distintas.

En la etapa 8 el accionamiento provoca que el brazo neumático ascienda una vez la pinza se había cerrado para sujetar el bote.

En cambio, en la etapa 12, el bote acaba de ser depositado en la mesa giratoria, por tanto el brazo neumático asciende para realizar la maniobra de vuelta a su posición de inicio.

El brazo neumático cuenta con dos sensores, introducción bote arriba (**h0**) e introducción bote abajo (**h1**). Por tanto la transición de ambas etapas se provocaba cuando el sensor **h0** estaba activado, es decir, cuando el brazo había llegado a su fin de carrera.

En la ilustración 43 podemos observar el fragmento de código del *Post* correspondiente a estas etapas.



*Ilustración 43: Post, Avance lineal introducción bote (Set)*

### 6.2.2.7 Etapa 9

En esta etapa el brazo neumático se dispone a girar a la vez que la pinza rota 90° para que el bote sea depositado correctamente en la mesa. Para ello activamos la etapa **%Q2.9** mediante un *Set*, correspondiente a Avance giro introducción bote (**I+**). Aparte precisamos de una variable interna, **%M7** y de dos temporizadores **%TM5** y **%TM6**.

La etapa 9 está conectada a la entrada del **%TM5** para que antes de activar la salida **%Q2.9** y **%M7** espere una fracción de tiempo para asegurarnos que el brazo ha terminado con éxito el avance lineal.

Por último, la variable **%M7** está conectada al **%TM6** por el cual efectuaremos la transición cuando su tiempo venza ya que no disponemos de ningún sensor que nos aporte información sobre esta salida.

En la ilustración 44 podemos observar el fragmento de código del *Post* donde podemos observar gráficamente lo descrito en esta sección.

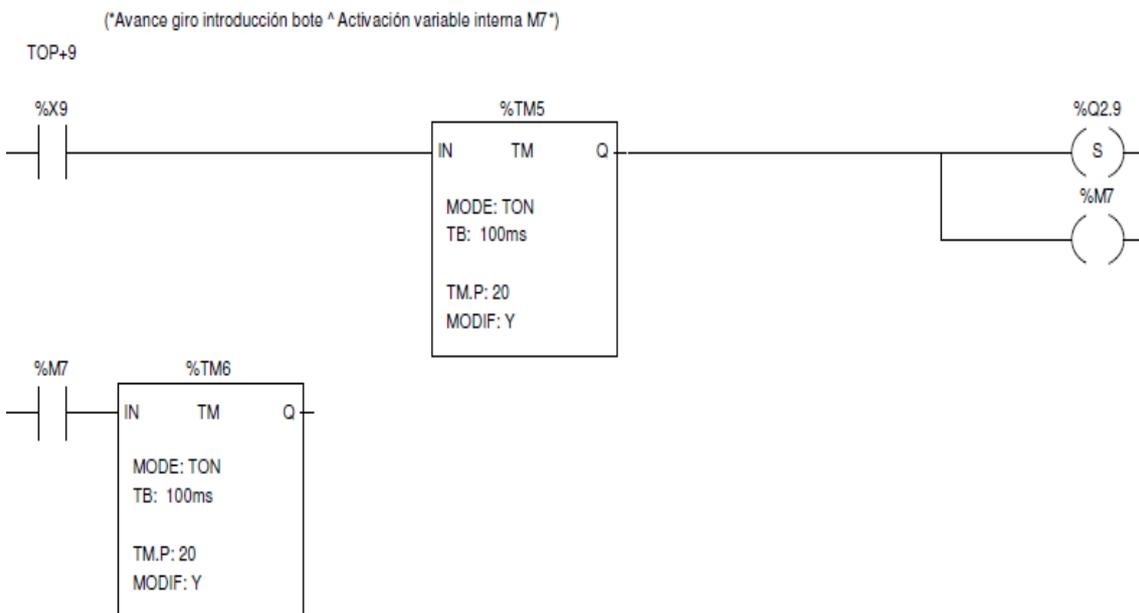


Ilustración 44: Post, Avance giro introducción bote (Set)

### 6.2.2.8 Etapa 10 y 14

Ambas etapas se corresponden con *Reset* de la salida **Q2.8** provocando el descenso del brazo neumático. Estas etapas tiene una particularidad, el final de este bloque funcional viene dado por la finalización de la etapa 14 por lo que para asegurarnos de que este ha finalizado con el *Reset* efectuado en la etapa 14 y no el realizado en la etapa 10, hemos implementado el contador **%C0** y solo cuando este tiene el valor de 2 la etapa 14 transita hacia la 15.

Resumiendo, cuando estas etapas se ejecuten incrementarán en 1 unidad el valor del contador, pero solo cuando la etapa 14 se ejecute se dará como finalizado este bloque.

El reseteado de este contador se realizar en la etapa 0, ver Implementación, sección 6.2.2.1.

En el caso de la etapa 10, el descenso del brazo neumático deja el bote en el primer puesto de la mesa giratoria, en cambio en la etapa 14, el brazo desciende para quedar en su posición inicial.

Las transiciones en cada uno de las etapas difieren levemente. En el caso de la etapa 10 la transición se efectúa cuando el sensor **h1**, correspondiente a Introducción bote abajo se activa. En el caso de la etapa 14 aparte de esta condición debe cumplirse que **%Co.d** sea igual a 2 unidades.

En las ilustraciones 45 podemos observar el fragmento de código del *Post* correspondiente a las etapas 10 y 14.

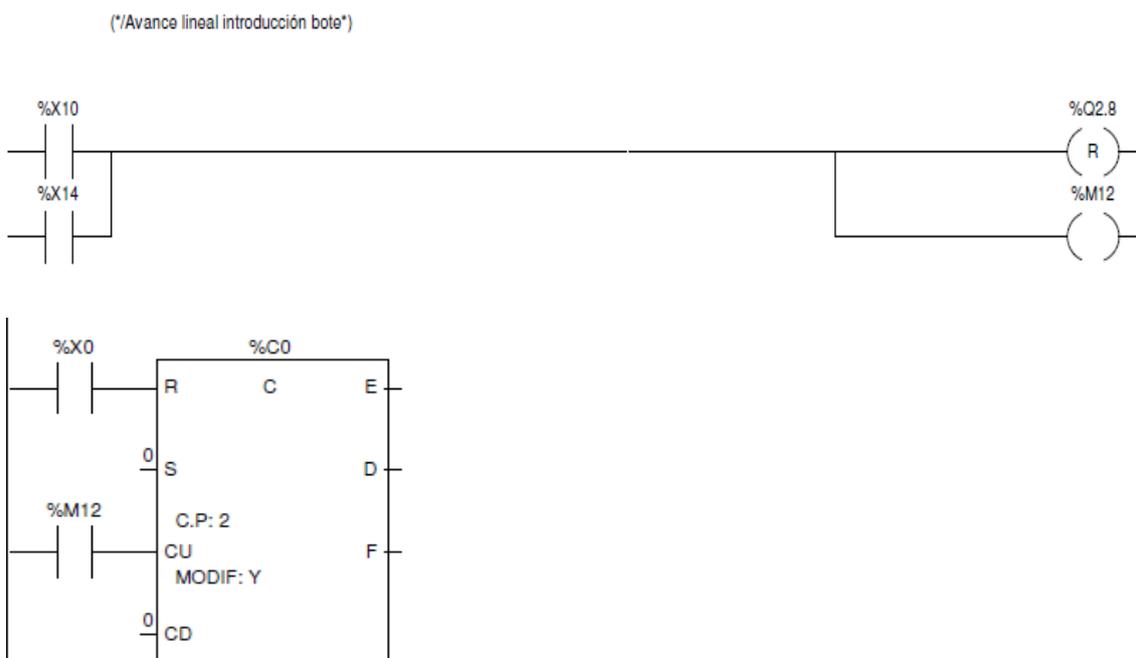


Ilustración 45: Post, Avance lineal introducción bote (Reset)

### 6.2.2.9 Etapa 11

En esta etapa la pinza que incorpora el brazo neumático se abre dejando el bote libre en el primer puesto de la mesa giratoria. Para ello hemos realizado un *Reset* sobre la salida **%Q2.10**.

Por otro lado hemos utilizado la variable interna **%M8** y dos temporizadores, **%TM7** y **%TM9**. El temporizador **%TM9**, se ha utilizado para dar un margen temporal para que el brazo descendiera totalmente y la pinza se abriera sin haber ningún tipo de situación anómala.

Como la pinza no dispone de un sensor asociado hemos introducido el **%TM7** para que esta etapa finalice cuando el tiempo de este venza. Para ello a la vez que realizamos el *Reset* sobre **%Q2.10** activamos **%M8**, que servirá de entrada para el **%TM7**.

En la ilustración 46 podemos observar el fragmento del *Post* asociado a la etapa 11.

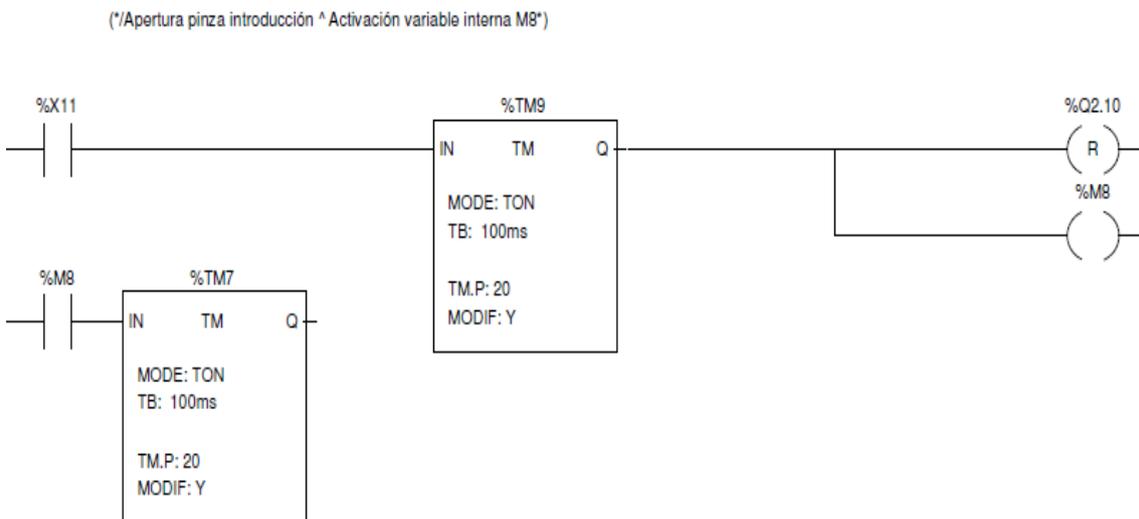


Ilustración 46: Post, Apertura pinza introducción (Reset)

### 6.2.2.10 Etapa 13

En esta etapa el brazo neumático gira para abandonar la mesa giratoria y volver a su posición origen. Para ello realizamos un *Reset* sobre la salida **%Q2.9**.

Por otro lado hemos incluido el temporizador **%TM11** y la variable **%M11**. Cuando la etapa sea activada, esta activará la variable **%M11** que activará al **%TM11** para que empiece a transcurrir el tiempo, de esa forma el brazo podrá realizar todo el recorrido.

Cuando el **%TM11** venza el tiempo establecido, la etapa 13 realizará la transición hacia la etapa 14.

En la ilustración 47 podemos observar el fragmento de código del *Post* asociado a la etapa 13.

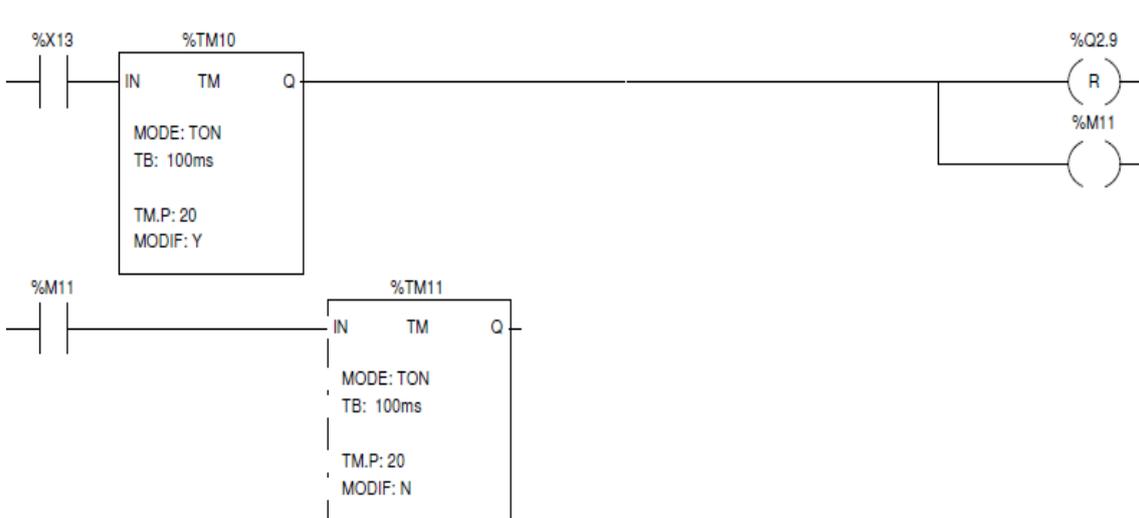


Ilustración 47: Post, Avance giro introducción bote (Reset)

### 6.2.3 Etapas 20-22

En esta sección se ha analizado el rango de etapas correspondiente con el bloque funcional de llenado de botes, ver Análisis funcional, sección 6.1.1.2.

En la ilustración 48 podemos observar el *GRAFSET* correspondiente a este bloque funcional.

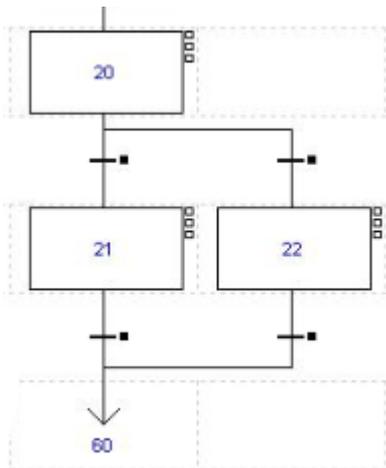


Ilustración 48: *GRAFSET*, Llenado de botes

Cabe destacar que este bloque funcional no estará siempre operativo, concretamente en la primera iteración no realizará ninguna acción, debido a que no habrá ningún bote en el puesto de llenado.

La etapa 20 es una etapa vacía surgida de realizar una divergencia en *AND*, por tanto, la primera etapa funcional es la etapa 21, así que comenzaremos el análisis con ella.

Este bloque funcional es el único que posee un sensor de detección, por tanto, los otros puestos que se encuentran en la mesa giratoria sabrán si deben actuar en función de las iteraciones que hayan pasado desde que el primer bote fue llenado.

Para llevar a cabo esta estrategia, tuvimos que implementar tres contadores: **%C1**, **%C2** y **%C3** y por último tres comparadores.

Los contadores fueron necesarios para ir contabilizando las veces que un bote había sido llenado, por tanto a la entrada de los tres contadores iba asociada la acción de llenado, **%Q4.3**. De esta forma para el puesto de alimentado de tapas **%C1.V** debía valer 1, en el caso del puesto de prensado de tapas **%C2.V** debía valer 2, por último el valor de **%C3.V** para el puesto de transporte de botes debía ser de 3. De esta forma cada puesto sabía cuándo llegaba un bote a su puesto y debían actuar. No obstante los contadores solo emiten un valor lógico de '1' cuando el valor exacto del contador es el valor objetivo, por tanto, cada puesto solo actuaría una iteración, ver Implementación. Sección 6.3.2.

Para poder solucionar este percance, debemos implementar los comparadores, de forma que siempre que el valor del contador sea mayor o igual al valor deseado, el comparador escriba en la salida un '1' lógico.

Resumiendo, **%Q4.3** ira a la entrada de los 3 contadores, por otro lado, cada **%Cx.V** siendo  $x = \{1, 2, 3\}$  ira conectado a la entrada del comparador correspondiente, la salida del comparador será decisiva para cada uno de los puestos para saber si tienen que operar o no.

### 6.2.3.1 Etapa 21

Esta etapa se ejecuta en aquellas iteraciones en las que el sensor **%dp1**, detector de bote, está activo y su funcionalidad es la de llenado del bote. Para ello hemos activado la salida **%Q4.3**, correspondiente a Llenado bote (**A4+**).

Esta etapa no posee un sensor asociado por lo que regularemos la cantidad llenada mediante el temporizador **%TM12** y de esta manera la transición hacia la etapa siguiente.

Por otro lado la activación de esta etapa provoca el incremento en 1 unidad el valor de los contadores: **%C1**, **%C2**, **%C3**.

En las ilustraciones 49 y 50 podemos observar el fragmento del *Post* asociado a la etapa 21, así como los contadores ligados a esta. Sobre la etapa **%X65**, ver Implementación, sección 6.2.1.

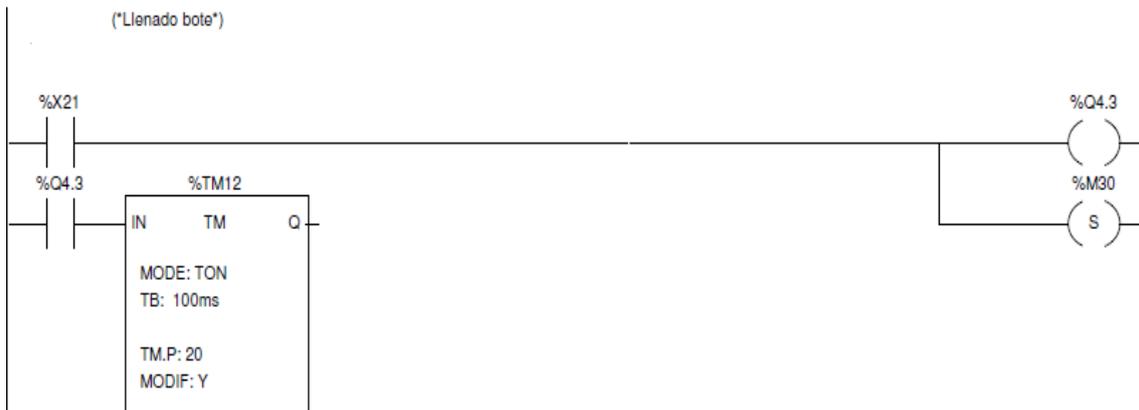


Ilustración 49: Post, Contadores asociados a la mesa giratoria

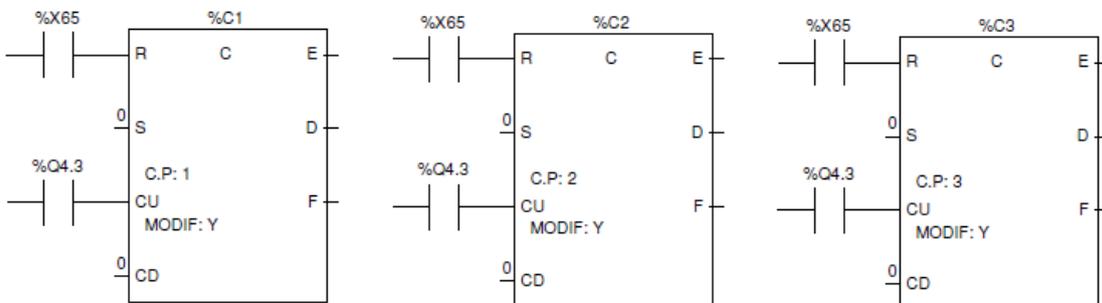


Ilustración 50: Post, Llenado bote

### 6.2.3.2 Etapa 22

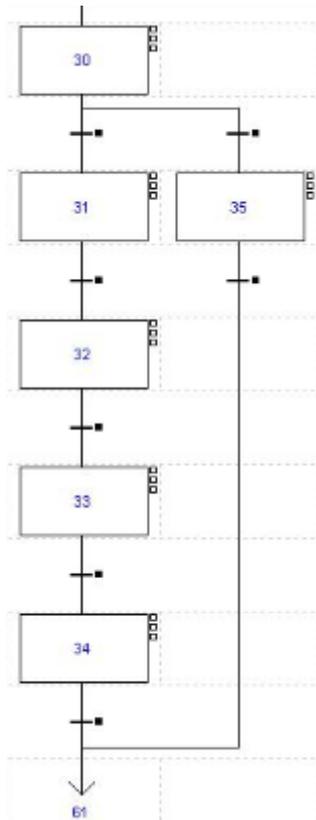
Esta etapa se ejecuta en aquellas iteraciones en las que el sensor **%dp1**, detector de bote, está desactivado, su funcionalidad es simplemente evitar que se active la salida **%Q4.3** cuando no haya un bote en el puesto de llenado.

Por tanto podemos decir que es una etapa vacía, y su condición de transición es simplemente su propia activación.

## 6.2.4 Etapas 30-35

En esta sección hemos analizado el bloque funcional correspondiente al alimentado de tapas, ver Análisis funcional, sección 6.1.1.3.

En la ilustración 51 podemos observar el *GRAF CET* asociado a este bloque funcional.



Cabe destacar que la etapa 30 es una etapa vacía surgida a raíz de la divergencia en *AND* tras la etapa 0, por lo que hemos comenzado el análisis en la etapa 31

Podemos observar que el *GRAF CET* está formado por dos ramas. La rama izquierda se encarga de realizar todo el proceso ligado a la alimentación de tapas.

En contraposición la rama derecha se encarga de evitar que se intente alimentar una tapa cuando no hay un bote en ella.

Para la detección de botes en este puesto es necesario consultar el resultado de realizar la comparación “>=” entre ‘1’ y *%C1.V*. Para revisar el proceso con todo detalle ver Implementación, sección 6.2.3.

Este proceso puede ser algo complejo de comprender y teniendo en cuenta que hemos programado este proyecto en un lenguaje gráfico, hemos adjuntado la ilustración 51 que muestra el fragmento de código del *Post* donde podemos observar paso a paso como efectuamos esta operación.

Ilustración 51: *GRAF CET*, Alimentado de tapas

Por otro lado hemos adjuntado la ilustración 52 que se corresponde con la transición de la etapa 30 hacia la etapa 31, de esta podemos deducir el contenido de la transición de la etapa 30 hacia la etapa 35 que como única diferencia contará con la variable interna **%M120** negada. La variable interna **%M80** forma parte del mecanismo de emergencia, ver Funcionalidad de la botonera, sección 6.2.10.3.

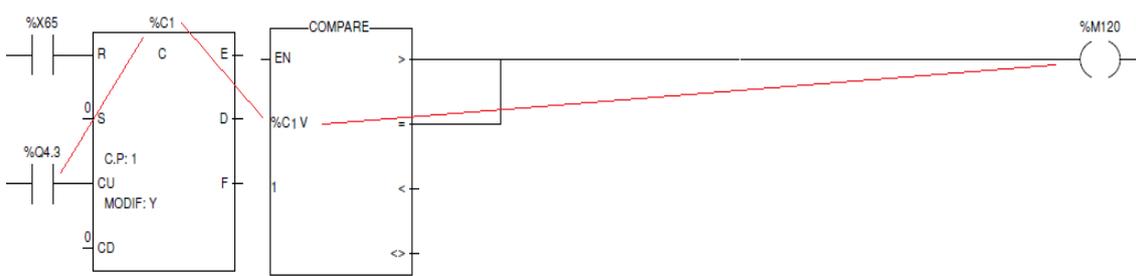


Ilustración 52: Post, %C1 + COMPARE

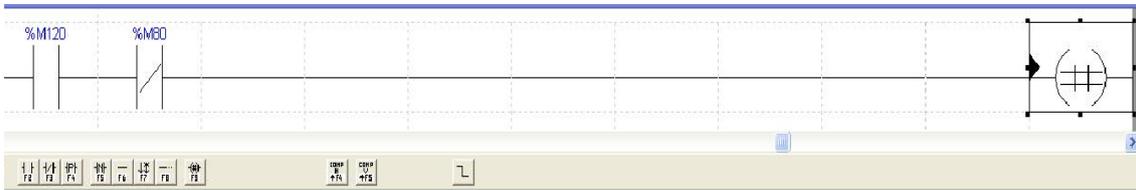


Ilustración 53: Transición Etapas 30-31

### 6.2.4.1 Etapa 31

Esta etapa se encarga de cerrar la pinza superior del alimentador de tapas, de forma que esta sujetará todas las tapas menos la que la estaba debajo de esta que quedará sujeta por la pinza inferior. Para ello realizamos un *Set* sobre la salida **%Q4.4**, correspondiente a Cierre alimentador tapas (**O+**) para que las pinzas queden permanentemente cerradas hasta nueva orden, por otro lado activamos la variable interna **%M13**.

La ejecución de esta etapa activa la salida **%Q4.4** y la variable interna **%M13** conectada a la entrada del temporizador **%TM13**, el cual nos hemos visto obligados a implementar para poder programar la transición a la siguiente etapa ya que no disponemos de ningún sensor asociado esta salida.

En la ilustración 54 podemos observar el fragmento de código del *Post* asociado a la etapa 31.



Ilustración 54: Post, Cierre alimentador tapas (Set)

### 6.2.4.2 Etapa 32

En esta etapa abrimos la pinza inferior de forma que el bote sujeto por esta caerá sobre el bote que se encontrase en este puesto. Para ello hemos realizado un *Set* sobre la salida **%Q2.6**, correspondiente a Apertura alimentador tapas (**F+**) y activado la variable interna **%M82** conectada al temporizador **%TM19**.

En este caso contábamos con dos sensores asociados a esta salida, **f0** y **f1** utilizados como inicio y fin de carrera de la pinza (apertura y cierre).

No obstante la transición en este caso no bastaba con la activación de **f1** ya que en la siguiente etapa la pinza inferior volvía a cerrarse, entonces añadimos un temporizador para que esta se mantuviese abierta durante un instante para que diera tiempo a la tapa a caer. Para ello a la vez que activábamos **%Q2.6**, también activábamos la variable interna **%M82**, de forma que **%TM19** comenzaba a contar.

Por tanto debía darse dos condiciones para que se produjera la transición, que **f1** estuviese activa y **%TM19** hubiera vencido su tiempo.

En la ilustración 55 podemos observar el fragmento de código del *Post* asociado a la etapa 32.

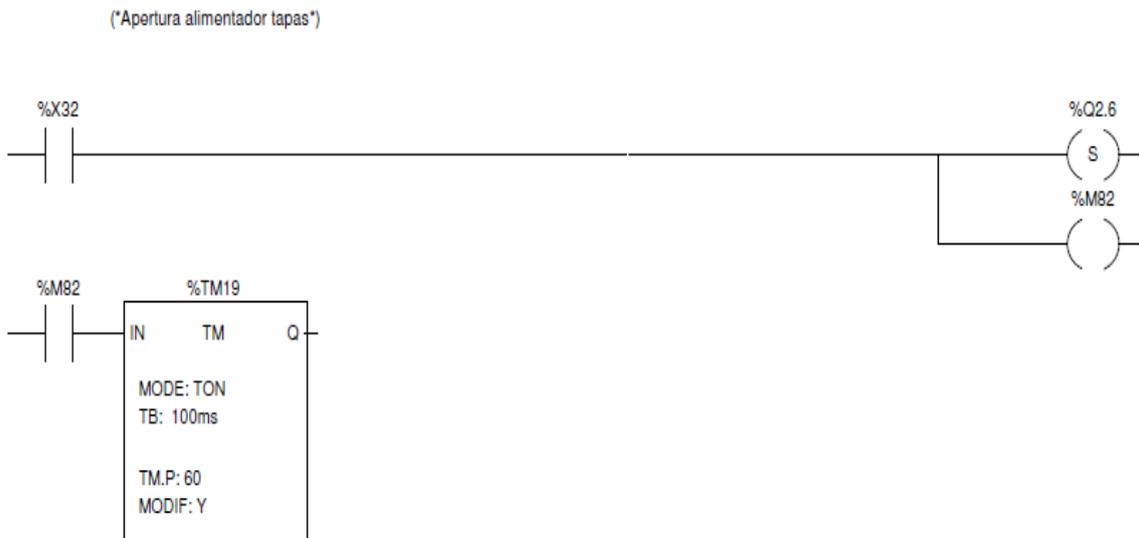


Ilustración 55: Post, Apertura alimentador tapas (Set)

### 6.2.4.3 Etapa 33

En esta etapa cerramos la pinza inferior para evitar que cuando la pinza superior sea abierta los botes caigan al vacío. Para ello realizamos un *Reset* sobre la salida **%Q2.6**.

Por tanto, al final de esta etapa nos encontramos en el puesto de alimentado de tapas con ambas pinzas cerradas.

La salida **%Q2.6**, tiene dos sensores asociados, **f0** y **f1**, utilizados como inicio y fin de carrera de la pinza (apertura y cierre). En este caso la transición a la siguiente etapa vendrá dado por la activación de **f0**, es decir, cuando la pinza haya llegado a su posición de inicio.

En la ilustración 56 podemos observar el fragmento de código del *Post* asociado a esta etapa.



Ilustración 56: Post, Apertura alimentador tapas (Reset)

#### 6.2.4.4 Etapa 34

La etapa 34 se encarga de abrir la pinza superior, de forma que la pinza inferior sujete toda la pila de tapas, tal y como estaba al inicio. Para ello realizamos un *Reset* sobre la salida **%Q4.4**.

En cuanto a la transición, al no contar con un sensor asociado necesitaremos incluir el temporizador **%TM14** y la variable interna **%M14**, activada inmediatamente al ejecutarse la etapa 34 y conectada a la entrada de **%TM14**, de forma que cuando el temporizador venza su tiempo, la etapa 34 realice la transición.

En la ilustración 57 podemos observar el fragmento de código del *Post* asociado a la etapa 34.

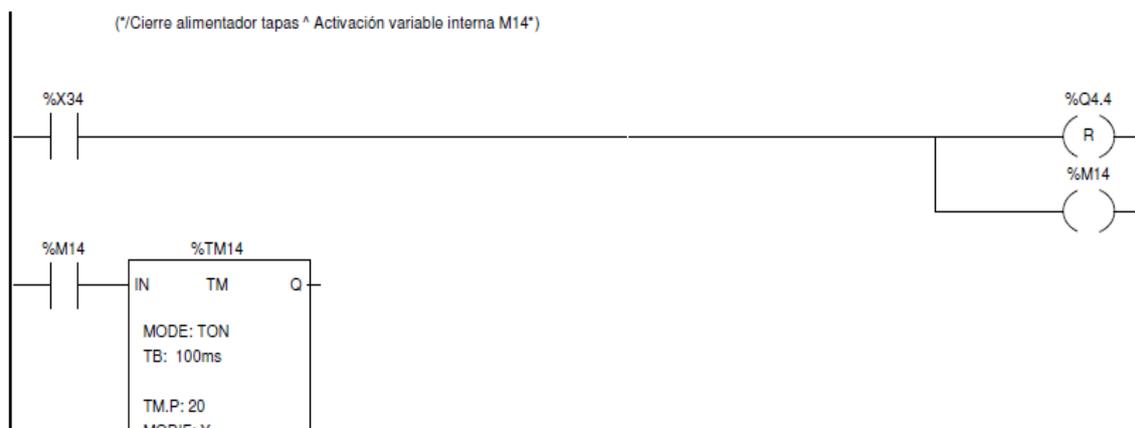


Ilustración 57: Post, Cierre alimentador tapas (Reset)

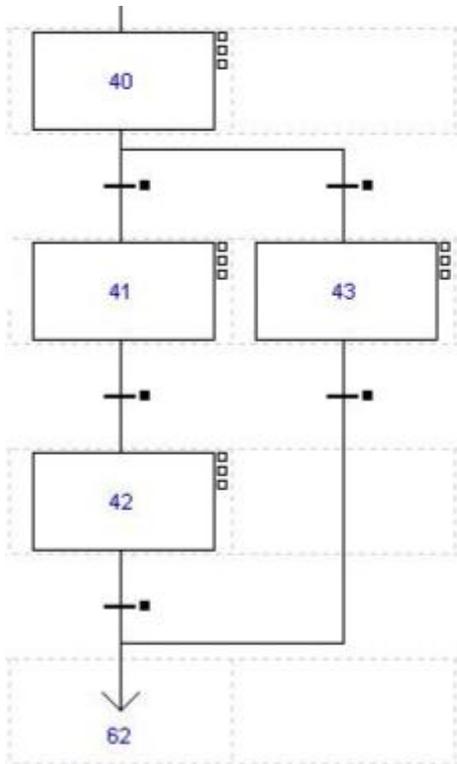
#### 6.2.4.5 Etapa 35

Para llegar a esta etapa, al no disponer de ningún sensor de detección de botes, la variable **%M120** será la que se encargue de indicarnos si hay algún bote en este puesto, en este caso debería estar desactivada. Para más detalles ver Implementación, sección 6.2.4.

Esta etapa tiene como función evitar que se alimente una tapa cuando en este puesto no hay ningún bote. Realmente no tiene que realizar ninguna acción y su transición vendrá dada por su propia activación.

### 6.2.5 Etapas 40-43

En esta sección hemos analizado el bloque funcional asociado al prensado de botes, ver Análisis funcional, sección 6.1.1.1.



En la ilustración 58 podemos observar el *GRAFCET* asociado a este bloque funcional.

Como en los distintos puestos de la mesa giratoria, contamos con una etapa vacía originada de la divergencia en *AND* tras la etapa 0, siendo en este caso la etapa 40.

Contamos con dos ramas, la rama izquierda se encarga de realizar el prensado de tapas

En contraposición la rama derecha se encarga de evitar que se realice el prensado cuando no haya un bote en este puesto.

Para la detección de botes en este puesto es necesario consultar el resultado de realizar la comparación " $\geq$ " entre '2' y  $\%C2.V$ . Para revisar el proceso con todo detalle ver Implementación, sección 6.2.3.

Ilustración 58: *GRAFCET*, Prensado de tapas

Para comprender mejor esta implementación hemos adjuntado la ilustración 59 y 60.

En la ilustración 59 observamos cómo están organizadas las entradas y salidas de los distintos objetos. La etapa  $\%X65$  se corresponde con la etapa inicial, para conocer más detalles, ver Implementación, sección 6.2.1. Por otro lado la salida  $\%Q4.3$ , se corresponde con el llenado de botes, para justificar el uso de esta salida, ver Implementación, sección 6.2.3.

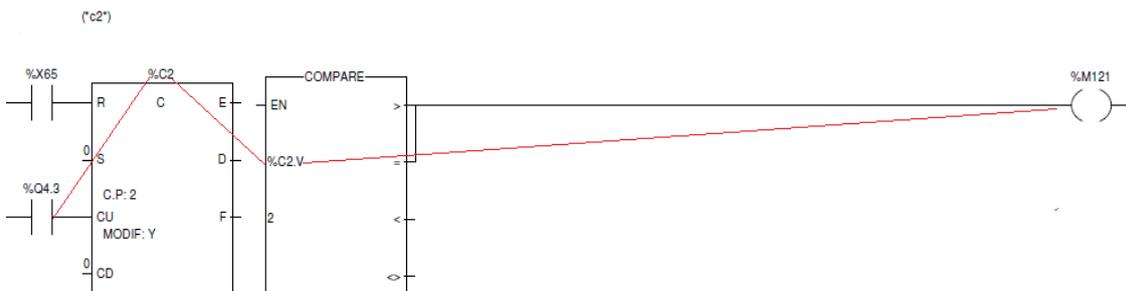


Ilustración 59: Post,  $\%C2$  + COMPARE

En la ilustración 60, podemos observar la transición asociada a la etapa 40 hacia la etapa 41. Una situación análoga ocurrirá en la transición hacia la etapa 43, pero en este caso negando la variable interna **%M121**. Esta variable contendrá el resultado final tras realizar las distintas operaciones.

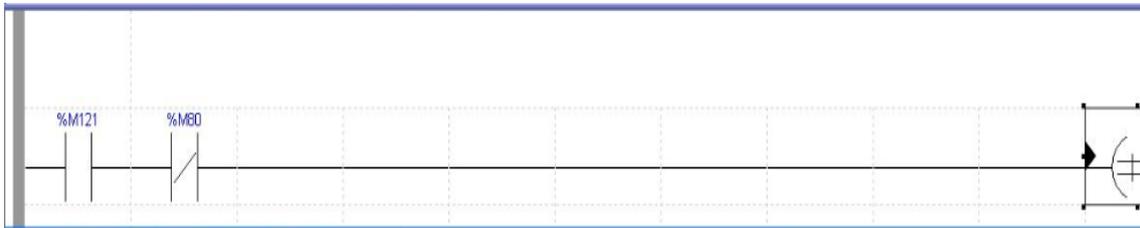


Ilustración 60: Transición, Etapas 40-41

### 6.2.5.1 Etapa 41

En esta etapa el cilindro neumático asociado a este puesto desciende para prensar la tapa previamente dejada caer en el bote. Para ello realizamos un *Set* sobre la salida **%Q2.7**, Avance prensa (**G+**).

Esta salida cuenta con dos sensores asociados, **g0** y **g1**, ambos sensores actúan de principio y fin de carrera. En este caso la transición hacia la siguiente etapa se producirá cuando **g1** se active, es decir, cuando el cilindro haya finalizado su recorrido.

En la ilustración 61 podemos observar el fragmento de código del *Post* asociado a esta etapa.



Ilustración 61: Post, Avance prensa (Set)

### 6.2.5.2 Etapa 42

En esta etapa el cilindro neumático asociado a este puesto asciende tras haber presado la tapa sobre el bote. Para ello realizamos un *Reset* sobre la salida **%Q2.7**, Avance prensa (**G+**).

Esta salida cuenta con dos sensores asociados, **g0** y **g1**, ambos sensores actúan de principio y fin de carrera. En este caso la transición hacia la siguiente etapa se producirá cuando **g0** se active, es decir, cuando el cilindro haya llegado a su posición de inicio.

En la ilustración 62 podemos observar el fragmento de código del *Post* asociado a esta etapa.

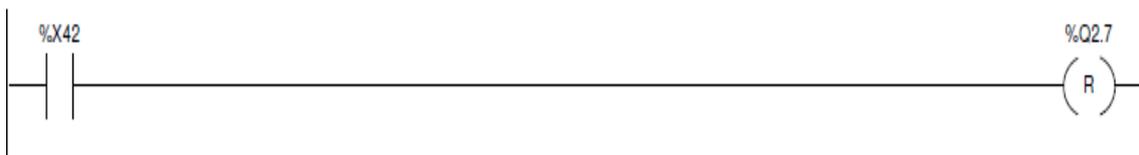


Ilustración 62: Post, Avance prensa (Reset)

### 6.2.5.3 Etapa 43

Esta etapa se ejecuta únicamente cuando no hay botes en el puesto de presando de tapas.

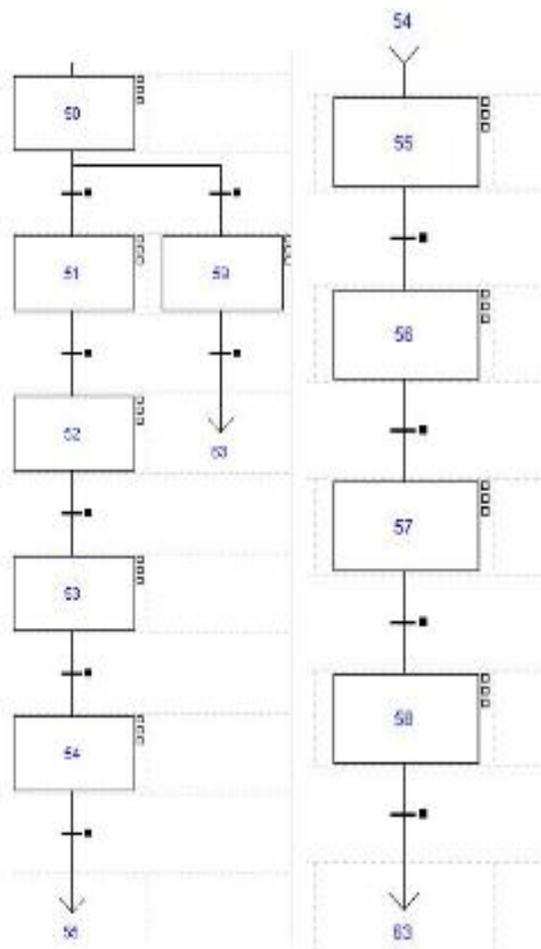
Su función es evitar que se active el proceso de prensado en los casos en los que no se proceda así que realmente no hay ninguna acción asociada a esta etapa.

La transición hacia la siguiente etapa vendrá dada por su propia activación.

### 6.2.6 Etapas 50-59

En esta sección analizaremos el bloque funcional asociado al transporte de botes, ver Análisis funcional, sección 6.1.1.5.

En la ilustración 63 podemos observar el fragmento de *GRAFCET* correspondiente a este puesto.



Como en los distintos puestos de la mesa giratoria, contamos con una etapa vacía originada de la divergencia en *AND* tras la etapa 0, siendo en este caso la etapa 50.

Contamos con dos ramas, la rama izquierda se encarga de realizar el transporte de botes.

En contraposición la rama derecha se encarga de evitar que se realice el transporte cuando no haya un bote en este puesto.

Para la detección de botes en este puesto es necesario consultar el resultado de realizar la comparación “>=” entre ‘3’ y %C3.V. Para revisar el proceso con todo detalle ver Implementación, sección 6.2.3.

Ilustración 63: *GRAFCET*, Transporte de botes

### 6.2.6.1 Etapa 51 y 55

En ambas etapas se realiza la misma acción, accionar el brazo para que este descienda, pero cada etapa en instantes y con objetivos distintos.

La etapa 51 precisa de esta acción para descender el brazo para situarlo de forma que el bote se encuentre entre sus pinzas. En cambio, la etapa 55 precisa de esta para descender el brazo y posteriormente depositar el bote en la estación de paletizado.

Para llevar a cabo esta acción realizamos un *Set* sobre la salida **%Q2.3**, correspondiente a Avance lineal extracción bote (**C+**).

Dado que esta salida tiene dos sensores asociados, **c0** y **c1**, nos hemos apoyado en estos para extraer la información necesaria para elaborar las condiciones de transición. Ambos sensores nos aportan información de fin e inicio de carrera.

En este caso nos hemos apoyado en el sensor **c1**, el cual se activa cuando el brazo ha alcanzado su fin de carrera, para elaborar la condición de transición.

En la ilustración 64 podemos observar el fragmento de código del *Post* asociado a estas etapas.



Ilustración 64: Post, Avance lineal extracción bote (Set)

### 6.2.6.2 Etapa 52

En esta etapa cerramos la pinza del brazo neumático para sujetar el bote que se encuentre en este puesto. Para ello realizamos un *Set* sobre la salida **%Q2.5**, Apertura pinza extracción bote (**E+**).

Dado que esta salida tiene asociado el sensor **e1**, el cual se activa cuando la pinza se ha cerrado, la transición hacia la siguiente etapa se producirá cuando **e1** se active.

En la ilustración 65 podemos observar el fragmento de código del *Post* asociado a esta etapa.



Ilustración 65: Post, Apertura pinza extracción bote (Set)

### 6.2.6.3 Etapa 53 y 57

En ambas etapas se realiza la misma acción, accionar el brazo para que este ascienda, pero cada etapa en instantes y con objetivos distintos.

La etapa 53 precisa de esta acción para ascender el brazo para dejarlo en la posición adecuada para posteriormente realizar el giro hacia la estación de paletizado. En cambio, la etapa 55 precisa de esta para ascender el brazo y posteriormente realizar el giro hacia su posición de inicio.

Para llevar a cabo esta acción realizamos un *Reset* sobre la salida **%Q2.3**, correspondiente a Avance lineal extracción bote (C+).

Dado que esta salida tiene dos sensores asociados, **c0** y **c1**, nos hemos apoyado en estos para extraer la información necesaria para elaborar las condiciones de transición. Ambos sensores nos aportan información de fin e inicio de carrera.

En este caso nos hemos apoyado en el sensor **c0**, el cual se activa cuando el brazo ha alcanzado su inicio de carrera, para elaborar la condición de transición.

En la ilustración 66 podemos observar el fragmento de código del *Post* asociado a estas etapas.



Ilustración 66: Post, Avance lineal extracción bote (Reset)

#### 6.2.6.4 Etapa 54

En esta etapa el brazo neumático realiza el giro para posicionarse en el puesto de descarga. Para ello realizamos un *Set* sobre la salida **%Q2.4**, Avance giro extracción bote (**D+**) de forma que el brazo procederá a realizar el giro hacia el puesto de descarga situado en la estación de paletizado.

Esta etapa tiene dos sensores asociados, **do** y **di** los cuales marcan el inicio y fin de carrera de esta acción (**D+**). En nuestro caso hemos elaborado la condición de transición en base al sensor **di**.

Cuando este se active, el brazo neumático habrá llegado a su fin de carrera, es decir, a su posición de destino.

En la ilustración 67 podemos observar el fragmento de código del *Post* asociado a la etapa 54.



Ilustración 67: Post, Avance giro extracción bote (Set)

#### 6.2.6.5 Etapa 56

En esta etapa, nos encontramos con el bote en el puesto de descarga sujetado por la pinza, por lo que abriremos esta para que el bote quede depositado finalmente en la estación de paletizado. Para ello realizamos un *Reset* sobre la salida **%Q2.5**, Apertura pinza extracción bote (**E+**) de forma que la pinza se abrirá.

Dado que el único sensor asociado a esta etapa, **e1**, se activa cuando la pinza se cierra, no podemos utilizar esta para elaborar la condición de transición hacia la siguiente etapa.

Hemos implementado el temporizador **%TM15**, el cual tiene conectado a su entrada la variable interna **%M15**. Esta variable es activada inmediatamente tras la ejecución de la etapa 56. Por lo que la transición se llevará a cabo cuando el tiempo de **%TM15** haya vencido.

En la ilustración 68 podemos observar el fragmento de código del *Post* asociado a la etapa 56.

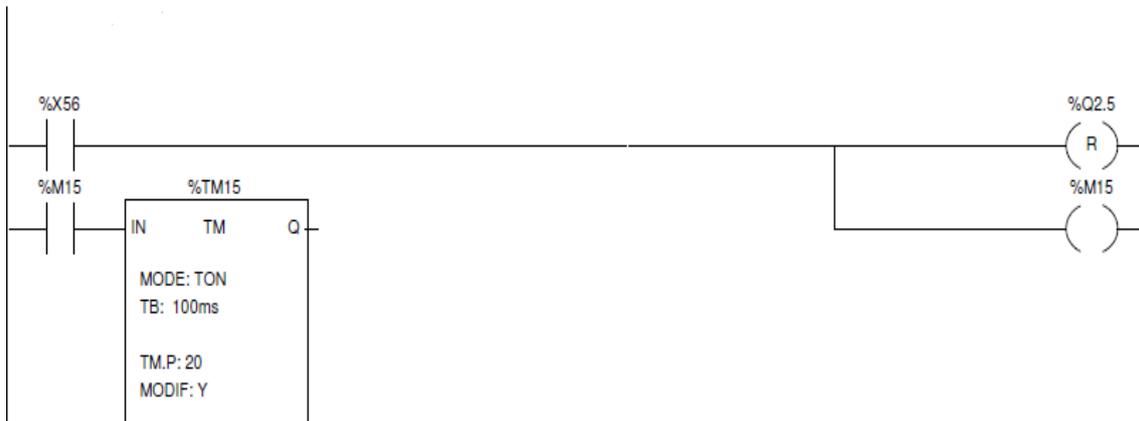


Ilustración 68: Post, Apertura pinza extracción (Reset)

#### 6.2.6.6 Etapa 58

En esta etapa el brazo neumático situado en la estación de paletizado gira para volver a su posición de inicio. Para ello basta con hacer un *Reset* a la salida **%Q2.4** de tal forma que el brazo girará hasta alcanza su inicio de carrera.

Ya que contamos con dos sensores asociados a esta salida, **do** y **d1**, nos apoyaremos en **do** el cual es activado cuando el brazo llega a su posición origen.

Por tanto la transición hacia la siguiente etapa vendrá dada por la activación de este sensor.

En la ilustración 69 podemos observar el fragmento de código del post asociado a la etapa 58.



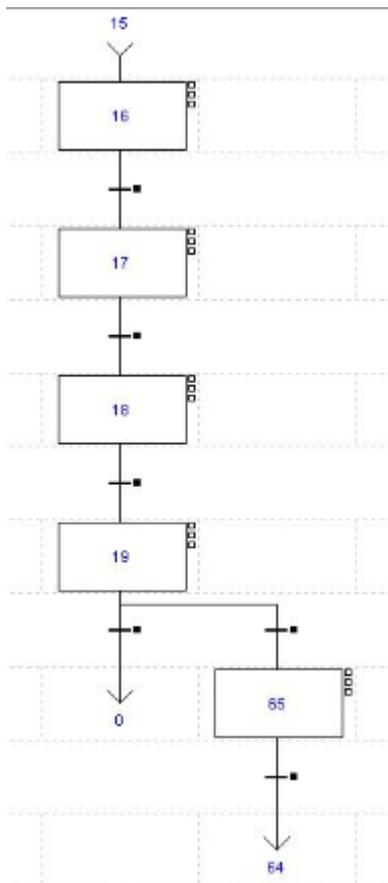
*Ilustración 69: Post, Avance giro extracción bote (Reset)*

#### **6.2.6.7 Etapa 59**

Esta etapa se ejecuta únicamente cuando en una iteración no ha botes que transportar, por tanto su función únicamente es no activar ninguna salida.

La transición hacia la siguiente etapa viene dada por su propia activación

### 6.2.7 Etapas 16-19



En esta sección vamos a analizar la secuencia de etapas: 16, 17, 18 y 19 correspondientes al bloque funcional, Giro mesa, ver Análisis funcional, sección 6.1.1.6.

En la ilustración 70, podemos observar el fragmento de *GRAFCET* asociado a este bloque funcional.

La mesa giratoria debe accionarse una vez todos los bloques funcionales restantes hayan terminado.

Para realizar la sincronización entre los distintos puestos tuvimos que implementar una convergencia/divergencia en AND, para más detalles ver Implementación, sección 6.2.8.

La etapa 19 transita hacia la etapa “inicial” en cambio la etapa 65 transita hacia la etapa inicial original.

La etapa 65 solo se ejecuta cuando el botón de *Reset* haya sido presionado, en cualquier otro caso se ejecutaría la etapa 19.

Para más detalles sobre la etapa 65, ver Implementación, sección 6.2.1.

Ilustración 70: *GRAFCET*, Giro mesa

### 6.2.7.1 Etapa 16

En esta etapa la mesa quedará bloqueada y preparada para el arrastre. Para ello se ha efectuado un *Set* sobre la salida **%Q2.2**, Avance bloque y arrastre (**B+**).

Dado que no contamos con ningún sensor asociado a esta salida, hemos utilizado el temporizador **%TM17**, al cual ha sido conectado a su entrada la variable **%M21**, activada tras la ejecución de la etapa 16. Por tanto la transición hacia la siguiente etapa vendrá dada por el vencimiento de **%TM17**.

En la ilustración 71, podemos observar el fragmento de código del *Post* asociado a la etapa 16.

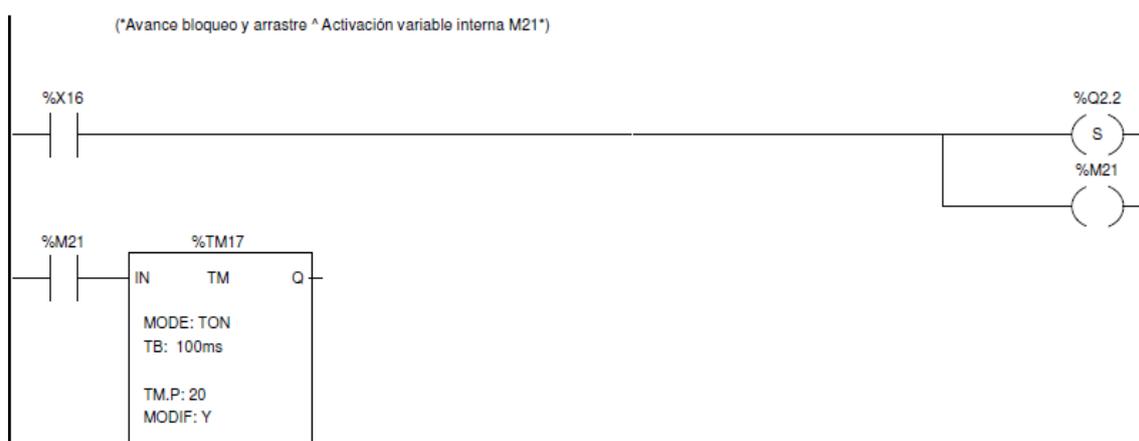


Ilustración 71: Post, Avance bloqueo y arrastre (Set)

### 6.2.7.2 Etapa 17

En esta etapa la mesa procederá a realizar el giro para que los botes implicados pasen al puesto siguiente de cara a la próxima iteración. Para ello realizamos un *Set* sobre la salida **%Q2.1**, Avance giro plato (**A+**).

En este puesto contamos con un sensor asociado, **a0**, Giro plato atrás, que detecta cuando el plato se encuentra en su posición inicial.

En este caso hemos tenido que implementar el temporizador **%TM16**, al cual hemos conectado a su entrada la variable interna **%M20**, activada inmediatamente tras la ejecución de la etapa 17.

Por tanto la transición hacia la siguiente etapa vendrá dada por el vencimiento de **%TM16** siempre que **ao** no esté activado.

En la ilustración 72 podemos observar el fragmentos de código del *Post* asociado a la etapa 17.

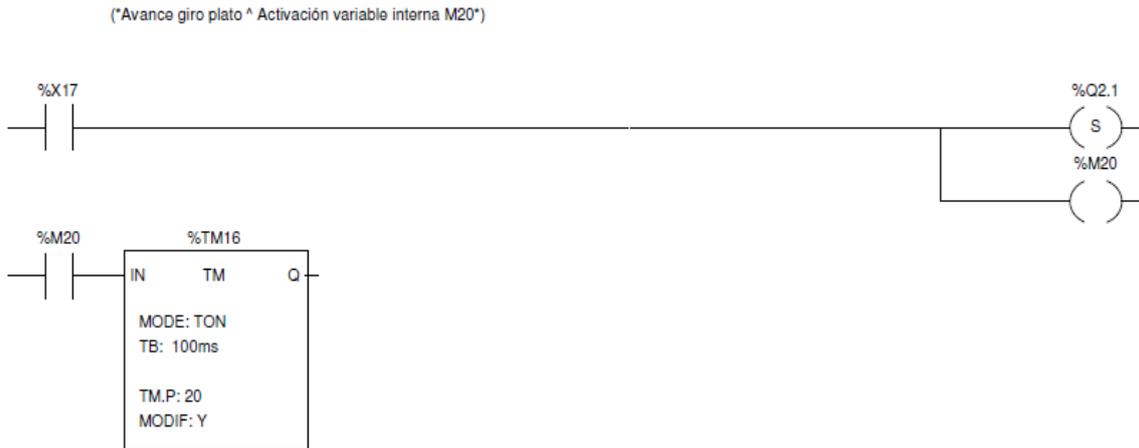


Ilustración 72: *Post*, Avance giro plato (Set)

### 6.2.7.3 Etapa 18

En esta etapa la mesa quedará desbloqueada de nuevo. Para desbloquear esta, hemos realizado un *Reset* sobre la salida **%Q2.2**.

Dado que no contamos con ningún sensor asociado a esta salida, hemos utilizado el temporizador **%TM18**, al cual ha sido conectado a su entrada la variable **%M22**, activada tras la ejecución de la etapa 16. Por tanto la transición hacia la siguiente etapa vendrá dada por el vencimiento de **%TM18**.

En la ilustración 73, podemos observar el fragmento de código del *Post* asociado a la etapa 18.



Ilustración 73: *Post*, Avance bloqueo y arrastre (Reset)

#### 6.2.7.4 Etapa 19

En esta etapa la pieza neumática que se encargaba de girar la mesa volverá a su posición inicial. Para ello realizamos un *Reset* sobre la salida **%Q2.1**.

En este caso la condición de transición vendrá dada por activación del sensor **a0**, este se activa siempre que la pieza neumática alcanza su posición inicial.

En la ilustración 74 podemos observar el fragmento de código del *Post* asociado a la etapa 19.



Ilustración 74: *Post*, Avance giro plato (Reset)

### 6.2.8 Convergencia Y Divergencia

Un factor muy importante que hemos tenido que tener muy presente de cara a la implementación de nuestro proyecto ha sido la incorporación de ramas alternativas y ramas simultáneas.

La ilustración 75 muestra el aspecto que tienen ambos tipos de ramificación.

En presencia de una **ramificación alternativa**, solo una de las ramas se activa de forma que las transiciones iniciales de cada rama deben ser distintas. Estas ramas se abren con transiciones y se cierran de igual forma.

En el caso de una **ramificación simultánea**, todas las ramas se activan a la vez y hasta que la última no finalice, la rama no termina.

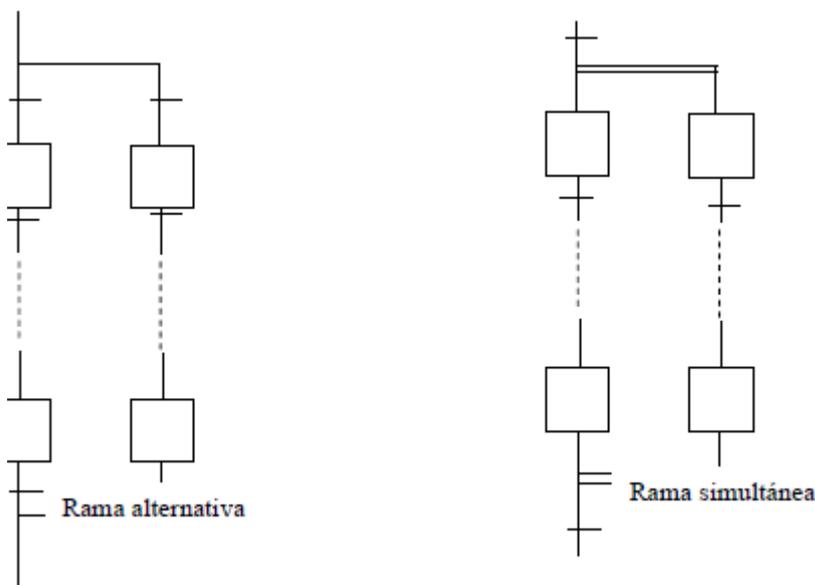


Ilustración 75: GRAFCET, Convergencia/Divergencia

Podemos identificar ambas estructuras en nuestro *GRAFCET*, ver Ilustración 21, sección 6.2.

En cada uno de los bloques funcionales excepto el correspondiente con el giro de mesa, podemos identificar al inicio una divergencia en OR debido a que en cualquiera de los casos se generan dos ramas en función de la identificación de un bote o no. Ambas ramas convergen finalmente a través de una transición.

En la ilustración 76 podemos observar cada una de las ramificaciones en OR.



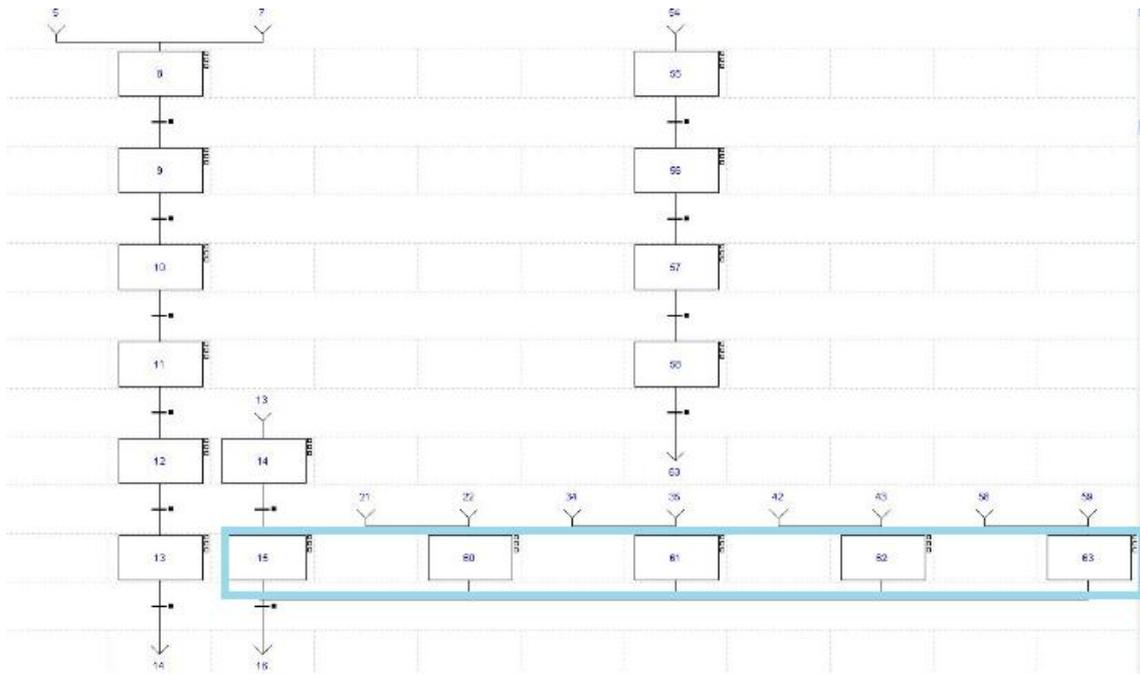


Ilustración 78: GRAFCET, Convergencia en AND

La divergencia abre en las etapas: 1, 20, 30, 40, 50 de forma que todas las etapas descendientes de estas se ejecutarán simultáneamente. Por otro lado estas ramas convergen a través de las etapas: 15, 60, 61, 62, 63 y solo cuando todas finalizan su proceso se produce la transición hacia la siguiente etapa. El siguiente bloque de etapas se corresponde con el bloque funcional de giro de mesa.

## 6.2.9 Objetos predefinidos, variables internas

En esta sección hemos analizado los objetos predefinidos por PL7 Micro y variables internas utilizadas en nuestro proyecto.

Como sabemos, contamos con numerosos temporizadores y algunos contadores y comparadores, ver Implementación, secciones 6.2.1-6.27.

### 6.2.9.1 Temporizadores

Los temporizadores son objetos predefinidos utilizados para introducir retardos en nuestras etapas.

Tienen tres modos de funcionamiento:

1. **TON:** Permite gestionar retardos en la conexión.
2. **TOF:** Permite gestionar retardos en la desconexión.
3. **TP:** Permite elaborar un impulso de duración precisa.

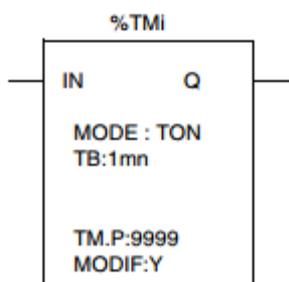


Ilustración 79: Temporizador

En la ilustración 79 podemos observar la representación gráfica de un temporizador.

Por otro lado los parámetros característicos de estos podemos observarlos en la tabla 15.

Característica	Variable	Valor
Número de temporizador	%TMI	0 a 63 para un TSX 37, 0 a 254 para un TSX 57
Modo	TON	• retardo en la conexión (por defecto)
	TOF	• retardo en la desconexión
	TP	• monoestable
Base de tiempo	TB	1mn (por defecto), 1s, 100ms, 10ms (16 temporizadores como máx. a 10ms). Cuanto más baja sea la base de tiempo, más elevada será la precisión del temporizador.
Valor actual	%TMI.V	Palabra que aumenta de 0 a %TMI.P al terminar el temporizador. Puede leerse y probarse, pero no escribirse por programa (%TMI.V puede modificarse mediante terminal).
Valor de preselección	%TMI.P	0-%TMI.P-9999. Palabra que puede leerse, probarse y escribirse por programa. Pasa al valor 9999 por defecto. La duración o el retardo elaborado es igual a %TMI.P x TB.
Ajuste por terminal (MODIF)	Y/N	Y: posibilidad de modificar el valor de preselección %TMI.P en ajuste. N: sin acceso al ajuste.
Entrada (instrucción) *Activación*	IN	En el flanco ascendente (modo TON o TP) o flanco *Activación* descendente (modo TOF), se inicia el temporizador.
Salida *Temporizador*	Q	Bit asociado %TMI.Q, su puesta en 1 depende de la función realizada TON, TOF o TP.

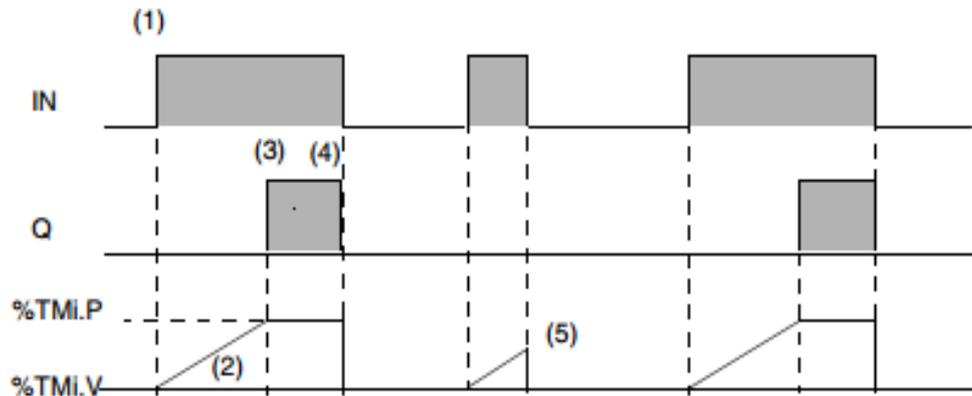
Tabla 15: Parámetros de los temporizadores (Schneider Electric (2005), Instrucciones de base, pág. 35)

6.2.9.1.1 Modos de funcionamiento

En esta sección vamos a entrar en detalle en cómo se comportan y funcionan los temporizadores en función de su modalidad de funcionamiento.

6.2.9.1.1.1 TON

En la ilustración 80 podemos ver el cronograma asociado los temporizadores **TON**.



En la siguiente tabla se describe el funcionamiento del temporizador en modo TON.

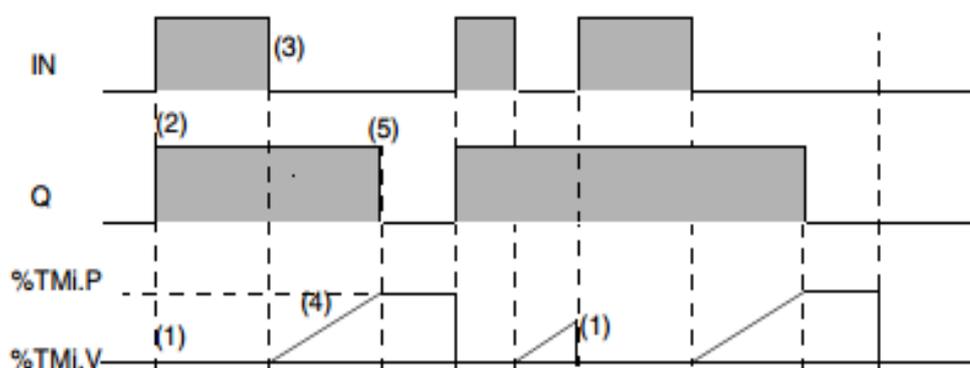
Fase	Descripción
1	En un flanco ascendente en la entrada IN, el temporizador se inicia
2	El valor actual %TMI.V del temporizador aumenta de 0 hacia %TMI.P de una unidad en cada impulso de la base de tiempo TB
3	El bit de salida %TMI.Q pasa a 1 en el momento en el que el valor actual alcanza %TMI.P
4	El bit de salida %TMI.Q permanece en 1 mientras la entrada IN esté en 1.
5	Cuando la entrada IN está en 0, el temporizador se detiene aunque estuviera en curso de evolución: %TMI.V toma el valor 0.

Ilustración 80: Cronograma, Temporizador TON (Schneider Electric ( 2005), Instrucciones de base, pág. 37)

### 6.2.9.1.1.2 TOF

En la ilustración 80 podemos ver el cronograma asociado los temporizadores **TOF**.

El cronograma muestra el funcionamiento del temporizador en modo TOF.



En la siguiente tabla se describe el funcionamiento del temporizador en modo TOF.

Fase	Descripción
1	El valor actual %TMI.V toma el valor 0 en un flanco ascendente de la entrada IN (aunque el temporizador esté en curso de evolución)
2	El bit de salida %TMI.Q pasa a 1.
3	En un flanco descendente en la entrada IN, el temporizador se inicia.
4	El valor actual aumenta hacia %TMI.P de una unidad en cada impulso de la base de tiempo TB.
5	El bit de salida %TMI.Q vuelve a 0 cuando el valor actual alcanza %TMI.P

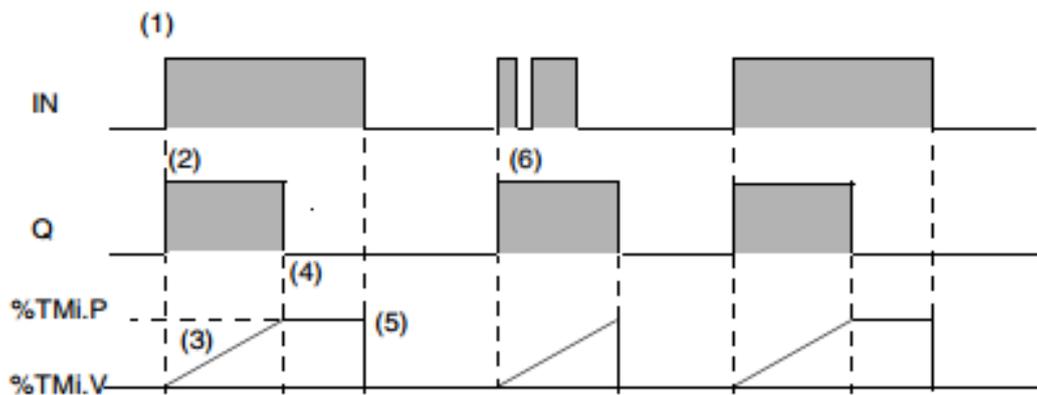
Ilustración 81: Cronograma, Temporizador TOF (Schneider Electric (2005), Instrucciones de base, pág. 38)

6.2.9.1.1.3 TP

En la ilustración 80 podemos ver el cronograma asociado los temporizadores **TP**.

El funcionamiento del temporizador en modo TP permite elaborar un impulso de duración precisa (función monoestable).

El cronograma muestra el funcionamiento del temporizador en modo TP.



En la siguiente tabla se describe el funcionamiento del temporizador en modo TP.

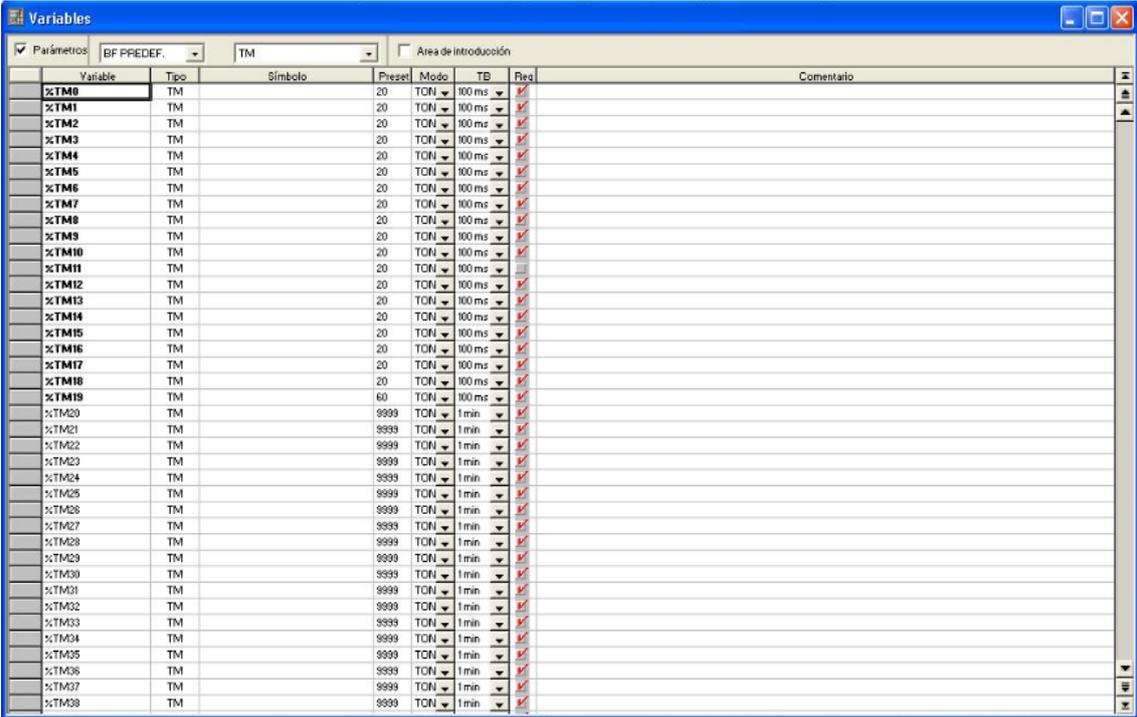
Fase	Descripción
1	En un flanco ascendente en la entrada IN, el temporizador se inicia
2	El bit de salida %TMI.Q pasa a 1.
3	El valor actual %TMI.V del temporizador aumenta de 0 hacia %TMI.P de una unidad en cada impulso de la base de tiempo TB
4	El bit de salida %TMI.Q vuelve a 0 cuando el valor actual alcanza %TMI.P.
5	Cuando la entrada IN y la salida %TMI.Q están en 0, %TMI.V toma el valor 0.
6	Este monoestable no se puede reactivar.

Ilustración 82: Cronograma, Temporizador TP (Schneider Electric (2005), Instrucciones de base, pág. 39)

### 6.2.9.1.2 Variables %TM

Para gestionar los distintos temporizadores, debemos acceder desde el **Navegador de aplicación** → **variables** → **BF predefinidos**. Por último en la segunda pestaña en la parte superior elegir **TM**.

En la ilustración 83 podemos observar el menú donde podemos gestionar este tipo de objetos.



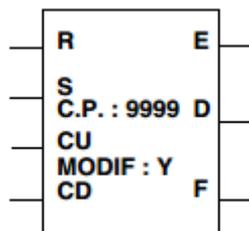
Variable	Tipo	Símbolo	Preset	Modo	TB	Comentario
%TM0	TM		20	TON	100 ms	
%TM1	TM		20	TON	100 ms	
%TM2	TM		20	TON	100 ms	
%TM3	TM		20	TON	100 ms	
%TM4	TM		20	TON	100 ms	
%TM5	TM		20	TON	100 ms	
%TM6	TM		20	TON	100 ms	
%TM7	TM		20	TON	100 ms	
%TM8	TM		20	TON	100 ms	
%TM9	TM		20	TON	100 ms	
%TM10	TM		20	TON	100 ms	
%TM11	TM		20	TON	100 ms	
%TM12	TM		20	TON	100 ms	
%TM13	TM		20	TON	100 ms	
%TM14	TM		20	TON	100 ms	
%TM15	TM		20	TON	100 ms	
%TM16	TM		20	TON	100 ms	
%TM17	TM		20	TON	100 ms	
%TM18	TM		20	TON	100 ms	
%TM19	TM		60	TON	100 ms	
%TM20	TM		9999	TON	1 min	
%TM21	TM		9999	TON	1 min	
%TM22	TM		9999	TON	1 min	
%TM23	TM		9999	TON	1 min	
%TM24	TM		9999	TON	1 min	
%TM25	TM		9999	TON	1 min	
%TM26	TM		9999	TON	1 min	
%TM27	TM		9999	TON	1 min	
%TM28	TM		9999	TON	1 min	
%TM29	TM		9999	TON	1 min	
%TM30	TM		9999	TON	1 min	
%TM31	TM		9999	TON	1 min	
%TM32	TM		9999	TON	1 min	
%TM33	TM		9999	TON	1 min	
%TM34	TM		9999	TON	1 min	
%TM35	TM		9999	TON	1 min	
%TM36	TM		9999	TON	1 min	
%TM37	TM		9999	TON	1 min	
%TM38	TM		9999	TON	1 min	

Ilustración 83: Variables, Temporizadores

Tenemos un total de 20 temporizadores, los cuales han tenido como papel fundamental regular y controlar aquellas salidas que no tenían un sensor asociado. En algunos casos se han utilizado para relajar algún bloque de todo el proceso para que los movimientos fueran más suavizados.

### 6.2.9.2 Contadores

Estos objetos están diseñados para el contado y descontado de sucesos, pudiendo ser ambas acciones simultáneas.



En la ilustración 84 podemos observar la representación gráfica de un contador y en la tabla 16 los parámetros significativos asociados a estos.

Ilustración 84: Contador

Características	Variable	Valor
Número del contador	%Ci	0 a 31 para un TSX 37, 0 a 254 para un TSX 57
Valor actual	%Ci.V	Palabra incrementada o disminuida en función de las entradas CU y CD. Puede leerse y probarse, pero no escribirse mediante programa. Puede modificarse mediante terminal.
Valor de preselección	%Ci.P	$0 \leq \%Ci.P \leq 9999$ . Palabra que puede leerse, probarse y escribirse (valor 9999 por defecto).
Ajuste por terminal (MODIF)	Y/N	<ul style="list-style-type: none"> <li>• Y : posibilidad de modificar el valor de preselección en ajuste.</li> <li>• N : sin acceso al ajuste.</li> </ul>
Entrada (instrucción) Reset	R	En estado 1: %Ci.V = 0
Entrada (instrucción) Preselección	S	En estado 1: %Ci.V = %Ci.P
Entrada (instrucción) Contaje	CU	Incrementa %Ci.V en el flanco ascendente
Entrada (instrucción) Descontaje	CD	Disminuye %Ci.V en el flanco ascendente
Salida Rebasamiento	E (Empty)	El bit asociado %Ci.E=1, cuando %Ci.V pasa de 0 a 9999 (se pone a 1 cuando %Ci.V es igual a 9999) vuelve a 0 si el contador sigue descontando. Cuando se produce un rebasamiento, el bit %S18 pasa a 1
Salida Preselección alcanzada	D (Done)	El bit asociado %Ci.D=1, cuando %Ci.V=%Ci.P.
Salida Rebasamiento	F (Full)	El bit asociado %Ci.F, cuando %Ci.V pasa de 9999 a 0 (se pone a 1 cuando %Ci.V es igual a 0) vuelve a 0 si el contador sigue descontando. Cuando se produce un rebasamiento, el bit %S18 pasa a 1

Tabla 16: Parámetros de los contadores (Schneider Electric ( 2005), Instrucciones de base,pág. 44)

### 6.2.9.2.1 Funcionalidad

La principal función de estos objetos consiste en contar y descontar. En la tabla 17 podemos observar el funcionamiento detallado.

Función Contaje

Acción	Resultado
Aparece un flanco ascendente en la entrada de contaje CU	El valor actual %Cl.V se incrementa de una unidad
El valor actual %Cl.V es igual al valor de preselección %Cl.P	El bit de salida %Cl.D *preselección alcanzada* asociado a la salida D pasa al estado 1
El valor actual %Cl.V pasa de 9999 a 0	El bit de salida %Cl.F (rebasamiento de contaje) pasa al estado 1
El contador sigue contando	El bit de salida %Cl.F (rebasamiento de contaje) vuelve a 0

Función Descontaje

Acción	Resultado
Aparece un flanco ascendente en la entrada de descontaje CD	El valor actual %Cl.V disminuye de una unidad
El valor actual %Cl.V pasa de 0 a 9999	El bit de salida %Cl.E (rebasamiento de descontaje) pasa al estado 1
El contador sigue descontando	El bit de salida %Cl.E (rebasamiento de descontaje) vuelve a 0

Función Contaje/Descontaje

Acción	Resultado
Aparece un flanco ascendente en la entrada de contaje CU	El valor actual %Cl.V se incrementa de una unidad
Aparece un flanco ascendente en la entrada de descontaje CD	El valor actual %Cl.V disminuye de una unidad
Las dos entradas están en 1 simultáneamente	El valor actual no cambia

Reset

Cuando	Resultado
La entrada R se pone a 1 (esta entrada tiene prioridad sobre el resto)	El valor actual %Cl.V se fuerza a 0. Las salidas %Cl.V, %Cl.D y %Cl.F están en 0

Preselección

Acción	Resultado
La entrada S *Preselección* se encuentra en el estado 1 y la entrada R *Reset*	El valor actual %Cl.V toma el valor %Cl.P y la salida %Cl.D pasa a 1

Tabla 17: Funcionalidad de los Contadores (Schneider Electric (2005), Instrucciones de base, págs. 45,46)

## 6.2.9.2.2 Variables %C

Para gestionar los distintos contadores, debemos acceder desde el **Navegador de aplicación** → **variables** → **BF predefinidos**. Por último en la segunda pestaña en la parte superior elegir **C**.

El resultado de esta navegación nos lleva al menú mostrado en la ilustración 85.

Variable	Tipo	Símbolo	Preset	Req	Comentario
%C0	C		2	✓	
%C1	C		1	✓	
%C2	C		2	✓	
%C3	C		3	✓	
%C4	C		4	✓	
%C5	C		9999	✓	
%C6	C		9999	✓	
%C7	C		9999	✓	
%C8	C		9999	✓	
%C9	C		9999	✓	
%C10	C		9999	✓	
%C11	C		9999	✓	
%C12	C		9999	✓	
%C13	C		9999	✓	
%C14	C		9999	✓	
%C15	C		9999	✓	
%C16	C		9999	✓	
%C17	C		9999	✓	
%C18	C		9999	✓	
%C19	C		9999	✓	
%C20	C		9999	✓	
%C21	C		9999	✓	
%C22	C		9999	✓	
%C23	C		9999	✓	
%C24	C		9999	✓	
%C25	C		9999	✓	
%C26	C		9999	✓	
%C27	C		9999	✓	
%C28	C		9999	✓	
%C29	C		9999	✓	
%C30	C		9999	✓	
%C31	C		9999	✓	

Ilustración 85: Variables, Contadores

En nuestro proyecto los contadores han sido utilizados principalmente para llevar un recuento de las veces que un bote había pasado por la estación de llenado y de esa forma saber cuándo debíamos activar los distintos puestos de la mesa giratoria.

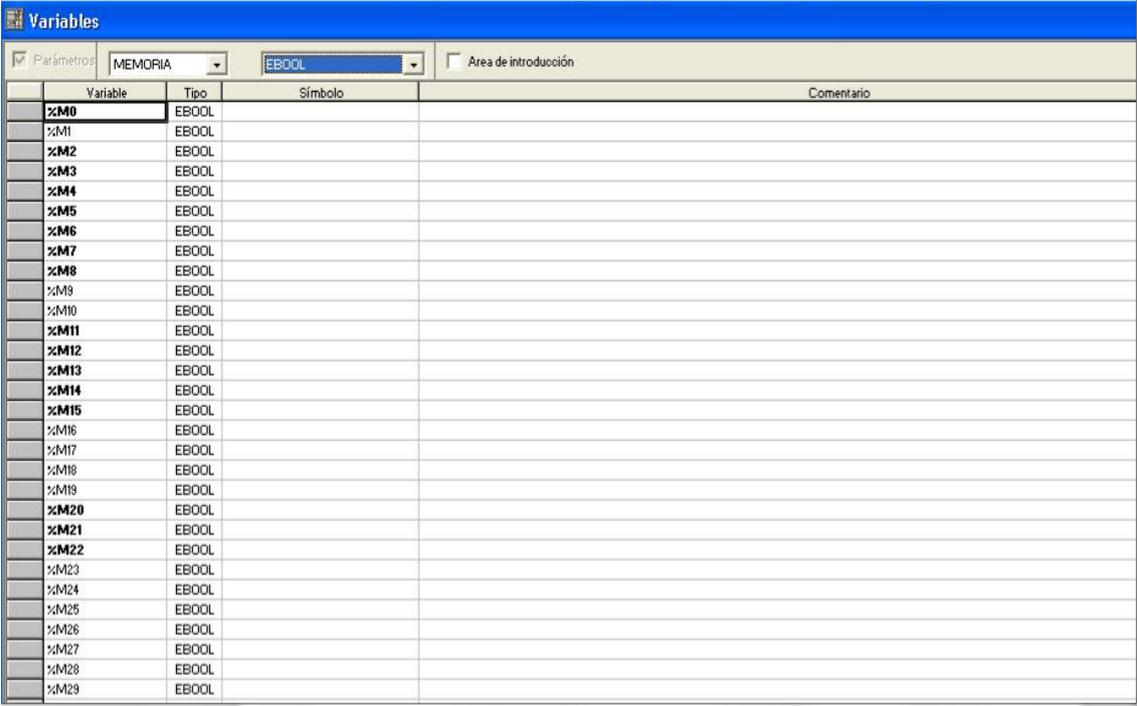
Por otro lado en el bloque funcional correspondiente al alimentado de botes hemos incorporado un contador ya que la última salida activada de este bloque la realiza dos veces y el contador nos permite regular la última transición, de forma que solo pasaremos a la siguiente etapa si la salida en cuestión ha sido activada dos veces en una misma iteración.

### 6.2.9.3 Variables internas

Para gestionar las variables internas, debemos acceder desde el **Navegador de aplicación** → **Variables** → **Memoria**. Por último en la segunda pestaña en la parte superior elegir **EBOOL**.

Las variables internas nos han resultado de gran utilidad como variables auxiliares para implementar los distintos temporizadores o bien para almacenar valores de cálculos intermedios en variables.

En la ilustración 86 podemos observar un fragmento del menú donde las variables internas son administradas.

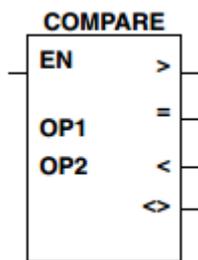


The screenshot shows a software window titled 'Variables'. At the top, there are controls for 'Parámetros' (Parameters), a dropdown menu set to 'MEMORIA', another dropdown set to 'EBOOL', and a checkbox for 'Area de introducción'. Below these is a table with the following columns: 'Variable', 'Tipo', 'Símbolo', and 'Comentario'. The table contains 29 rows of data, all with the type 'EBOOL' and empty symbol and comment fields.

Variable	Tipo	Símbolo	Comentario
%M0	EBOOL		
%M1	EBOOL		
%M2	EBOOL		
%M3	EBOOL		
%M4	EBOOL		
%M5	EBOOL		
%M6	EBOOL		
%M7	EBOOL		
%M8	EBOOL		
%M9	EBOOL		
%M10	EBOOL		
%M11	EBOOL		
%M12	EBOOL		
%M13	EBOOL		
%M14	EBOOL		
%M15	EBOOL		
%M16	EBOOL		
%M17	EBOOL		
%M18	EBOOL		
%M19	EBOOL		
%M20	EBOOL		
%M21	EBOOL		
%M22	EBOOL		
%M23	EBOOL		
%M24	EBOOL		
%M25	EBOOL		
%M26	EBOOL		
%M27	EBOOL		
%M28	EBOOL		
%M29	EBOOL		

Ilustración 86: Variables, Variables internas

#### 6.2.9.4 Otros



A lo largo del proyecto hemos utilizado el operador de comparación vertical, gracias al cual hemos conseguido un correcto funcionamiento de los puestos depositados en la mesa giratoria a lo largo de las iteraciones. Sin este, solo hubiéramos conseguido realizar correctamente la función de cada puesto una vez por ejecución del programa.

La ilustración 87 nos muestra la representación gráfica de este operador.

Ilustración 87:  
Comparador

El bloque comparador vertical nos permite efectuar una comparación entre dos operandos (**OP**). Estos dos operandos son de tipo palabra de 16 bits eventualmente indexados o de valor inmediato. La puesta a '1' de la entrada, provoca la comparación de los dos operandos. Las cuatro salidas se activarán en función del resultado de la comparación.

En la tabla 18 podemos observar la funcionalidad de cada uno de los parámetros del bloque comparador vertical.

<b>Característica</b>	<b>Variable</b>	<b>Valor</b>
Entrada de comando	EN	En el estado 1 provoca la comparación de los dos operandos.
Salida "Superior"	>	Está en el estado 1 si el contenido de OP1 es superior al de OP2.
Salida "Igual"	=	Está en el estado 1 si el contenido de OP1 es igual al de OP2.
Salida "Inferior"	<	Está en el estado 1 si el contenido de OP1 es inferior al de OP2.
Salida "Diferente"	<>	Está en el estado 1 si el contenido de OP1 es diferente del de OP2.
Operando número 1	OP1	Este operando es un objeto palabra de longitud simple (puede indexarse).
Operando número 2	OP2	Este operando es un objeto palabra de longitud simple (puede indexarse).

*Tabla 18: Parámetros de los comparadores verticales (Schneider Electric ( 2005), Instrucciones de base, 2005, pág. 120)*

### 6.2.10 Funcionalidad de la botonera

La estación de llenado contiene una botonera la cual ha sido programada y con la que el usuario final es capaz de controlar la planta industrial sin la necesidad de involucrarse con el PC.

En la ilustración 88 podemos observar el aspecto de la botonera.



*Ilustración 88: Botonera*

Los distintos botones funcionan como el resto de sensores, están conectados a distintas entradas del *PLC*, pudiendo averiguar si están pulsados o no, consultando el valor de la entrada que les corresponda.

Hemos dotado de tres funciones a la botonera:

1. Inicio del proceso industrial
2. Reinicio del proceso industrial
3. Mecanismo de emergencia.

### 6.2.10.1 Inicio del proceso industrial

Una vez el programa ha sido transferido y ejecutado, ver Comunicación PC, sección 3.8.4, la estación ya ha entrado en funcionamiento y se encuentra en la etapa inicial, es decir, etapa 65.

La condición por la cual transitamos a la siguiente etapa es la activación de **%I1.0**, correspondiente a Pulsador marcha (**m**).

De forma que solo comenzará el proceso a funcionar cuando hayamos pulsado este botón.

Mientras no se pulse el botón de *Reset*, ver Funcionalidad de la botonera, sección 6.2.10.2 o el botón de *Stop*, ver Funcionalidad de la botonera, sección 6.2.10.3, la planta funcionará hasta que se detenga el programa.

En la ilustración 89 podemos observar el fragmento de código del *Post* asociado al botón *Start* de la botonera.



Ilustración 89: Post, Botón Start

La variable interna **%M81**, está vinculada con el mecanismo de emergencia, para conocer su funcionalidad, ver Funcionalidad de la botonera sección 6.2.10.3. Por otro lado la variable interna **%M80** ha quedado obsoleta y carece de funcionalidad.

### 6.2.10.2 Reinicio del proceso industrial

Una funcionalidad indispensable es la detención y reinicio del proceso industrial. Para ello hemos utilizado el botón de *Reset* de la botonera, conectado a la entrada **%I1.4**, correspondiente a Pulsador de *Reset* (**r**).

Una vez el botón de *Reset* sea pulsado se activará la variable interna **%M92**, esta variable es consultada al final del *GRAFCET* de forma que si ha sido pulsado el botón de *Reset*, al finalizar la iteración volveremos a la etapa inicial 65 con lo cual habría que retirar los botes que estuviesen en mitad del proceso y ya podríamos volver a empezar el proceso desde cero.

En la ilustración 90 podemos apreciar el fragmento del *GRAFCET* donde podemos observar la divergencia/convergencia en OR causada por **%M92**.

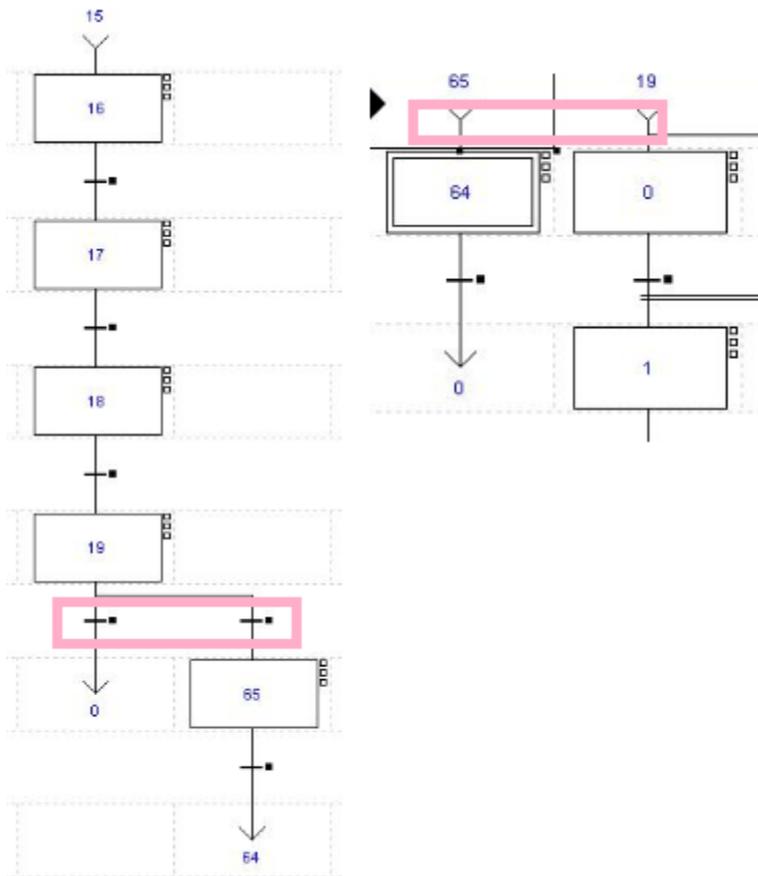


Ilustración 90: *GRAFCET*, Divergencia/Convergencia (*Reset*)

Por otro lado podemos observar en la ilustración 91 el fragmento de código del *Post* asociado al botón *Reset* de la botonera.



Ilustración 91: *Post*, Botón *Reset*

### 6.2.10.3 Mecanismo de Emergencia

En esta sección hemos analizado el funcionamiento y protocolo de actuación asociado al mecanismo de emergencia implementado.

El mecanismo de emergencia solo debe ser activado ante la detección de una situación anómala en el proceso industrial.

En el caso de visualizar una situación de esta índole se debe seguir los siguientes pasos:

1. Pulsar el botón *Stop*.
2. Solucionar el percance.
3. Inspección ocular de la planta.
4. Pulsar el botón *Start*.

Antes de comenzar con la explicación cabe destacar que el botón de *Stop* funciona con lógica inversa, es decir, sacando un 'o' cuando este es activado y viceversa.

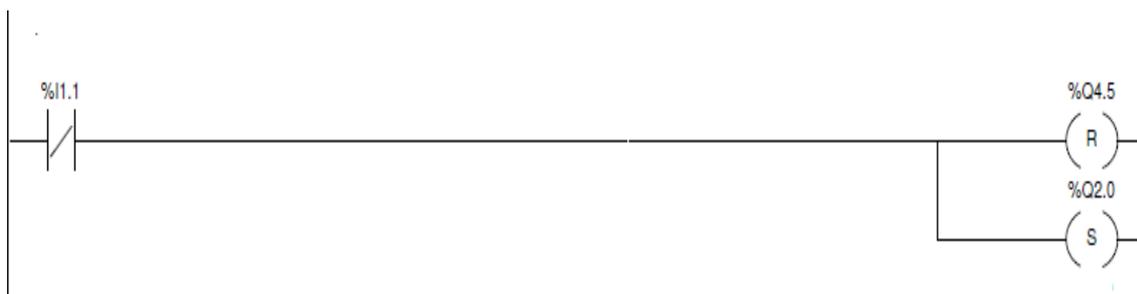
Cuando detectemos una anomalía pulsaremos el botón de *Stop* para detener el proceso. La entrada asociada a este botón es **%I1.0**, Pulsador paro (**p**), la cual se “desactivará” cuando el botón sea pulsado.

Una vez sea pulsado se realizará un Reset sobre la salida **%Q4.5**, correspondiente con el suministro eléctrico a las distintas entradas y salidas del *PLC* y un *Set* sobre la salida **%Q2.0**, correspondiente con Baliza roja (**BR**).

En la ilustración 92 podemos observar el fragmento de código del *Post* asociado al botón de *Stop*.

Una vez los pasos 2 y 3 han sido realizados bastaría con apretar el botón de *Start* para volver al funcionamiento habitual, ver Funcionalidad de la botonera, sección 6.2.10.1.

Tras pulsar el botón de *Start*, la variable **%M81** se activa para restaurar el sistema después de haber sido sometido a una situación de emergencia, apagando la baliza roja (**BR**) y suministrando corriente eléctrica a toda la estación, tal y como si se tratase de la etapa inicial.



*Ilustración 92: Post, Botón Stop*

## 6.2.11 Futuras mejoras

En esta sección hemos aportado una serie de mejoras para la estación de llenado. El principal problema de esta estación es que no ejerce ningún control sobre el número de botes disponibles y otro problema relevante es la total dependencia del sensor **dp1** para controlar los distintos puestos de la mesa giratoria.

### 6.2.11.1 Control del alimentador de botes

Hemos aportado dos posibles soluciones al primero de los problemas:

1. La primera de ellas es una solución *Software* y debemos partir de colocar siempre el mismo número de botes en el alimentador. De esta forma implementar un contador con %Ci.P igual al número de botes, ver Contadores, sección 6.2.9.2.1. La salida %Ci.D estará conectada en la misma línea del post que el botón de *Stop*, de forma que cuando el contador %Ci.V = %Ci.P la estación quede sin suministro eléctrico y se encienda la baliza luminosa roja. Una vez se haya recargado el alimentador, bastaría con pulsar el botón *Start* y el proceso seguiría donde hubiese terminado.

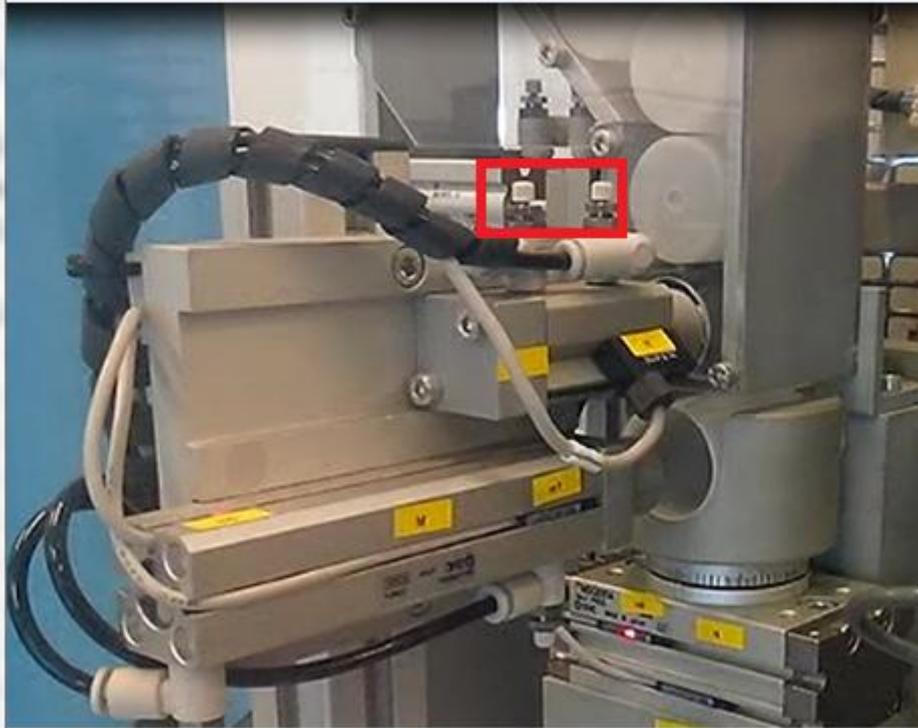
Algo más complejo es decidir que conectamos en %Ci.CU. Lo más adecuado sería crear un %TMi colocado en la etapa %X19 de forma que cuando acabase de realizar las acciones programadas, se lanzara el temporizador y una vez venciera, incrementar el valor del contador. Por otro lado haría falta añadir en la transición de la etapa %X19 la condición de que el temporizador que hemos creado hubiese vencido su tiempo.

2. La segunda es una solución *Hardware*, más sencilla pero implica un suplemento económico debido a que deberíamos incorporar un sensor, en este caso fotoeléctrico 3 hilos PNP al igual que el que tenemos en la estación de llenado. En la ilustración 93 podemos observar donde iría colocado este sensor.

En cuanto a la instalación y puesta en marcha, el sensor debería ser fijado de alguna forma a la plataforma, pero no es un problema que nos ocupe. Por otro lado sería necesario hacer una pequeña incisión en la cristalera donde están depositados los botes, para que el rayo pudiera incidir sobre estos correctamente. Dado que el PLC tiene el canal 3 con varias entradas vacías podríamos conectar este sin problema.

Una vez en funcionamiento, el sensor continuamente debería estar activo ya que detectaría botes en el alimentador. Cuando el último bote entrase en la cadena de producción el sensor no detectaría ningún bote y permanecería a 'o' en esta última iteración. En este caso, una vez la etapa **%X19** hubiese terminado, activaríamos una variable interna creada para este propósito, **%Mi**.

Por último ejecutaríamos el fragmento del *Post* asociado al botón de Stop, si y solo si, la variable interna estuviese activada y el sensor en cuestión desactivado.



*Ilustración 93: Mejora I, Alimentador de botes*

### 6.2.11.2 Mayor robustez

Dado que el sensor **dp1** es el único sensor de detección de botes que disponemos en la mesa giratoria, nos hemos apoyado en este para regular el tránsito de los botes a lo largo de los distintos puestos. De esta forma podemos conseguir un funcionamiento aceptable a bajo coste *Hardware* a cambio un mayor coste *Software*.

Pero podríamos añadir un sensor para cada puesto de forma que la capacidad de control sobre cada uno de los puestos sería mucho mayor y no solo eso, también el sistema ganaría robustez ya que estamos dependiendo de un sensor para controlar gran parte del proceso.

Incluyendo un sensor para cada puesto, cada uno sería responsable del funcionamiento de su puesto, y sabrían cómo responder si eliminásemos parte de los botes incorporados en la mesa ya que eliminaríamos toda dependencia.

Por último la lógica de la programación sería más sencilla y sobre todo podríamos interactuar más con el *Hardware* sin la necesidad de programar bloques de código simulando el control de los puestos, perdiendo la esencia de la programación de *PLC*. La depuración también sería bastante más sencilla ya que mientras menos dependencias tengamos más fácilmente localizaremos la fuente del error.

## 7 Conclusión

---

Una vez terminado el trabajo no solo podemos afirmar haber cumplido con nuestros objetivos satisfactoriamente sino que también nos ha ayudado a comprender y conocer el funcionamiento de distintas tecnologías industriales. Por otro lado el realizar un trabajo de tal envergadura nos ha ayudado a desarrollar nuestra capacidad de: síntesis, montaje y puesta a punto, programación de *PLC*, diagnóstico y reparación de averías y elaboración de documentación técnica y diseño.

El lenguaje *GRAFSET* empleado, es un lenguaje gráfico muy potente de programación de *PLC*, los cuales causaron un impulso muy importante en el sector de la industria, dotando de una gran flexibilidad a cualquier proceso industrial. Uno de los mayores atractivos de esta tecnología es la facilidad de mantenimiento, prácticamente nulo, además de aumentar la fiabilidad del sistema al eliminar contactos móviles y dotando a este la capacidad de detectar e indicar las posibles averías.

La ingeniería en automática es la principalmente encargada del desarrollo de *PLC*, la cual día a día está más involucrada en la sociedad y no solo se ha abierto puertas en sector de la industria. Actualmente la ingeniería automática abarca áreas como: electrónica, domótica, química, mecánica, automovilismo, aeronáutica, robótica, biología, medicina y mecatrónica, etc...

Por último, una vez realizado el proyecto y adquirido las actitudes y conocimientos necesarios, no solo como programador de *PLC* sino como ingeniero, podemos enfrentarnos a un mundo laboral competente donde los sistemas automatizados reinan todo el mercado.

# 8 Bibliografía

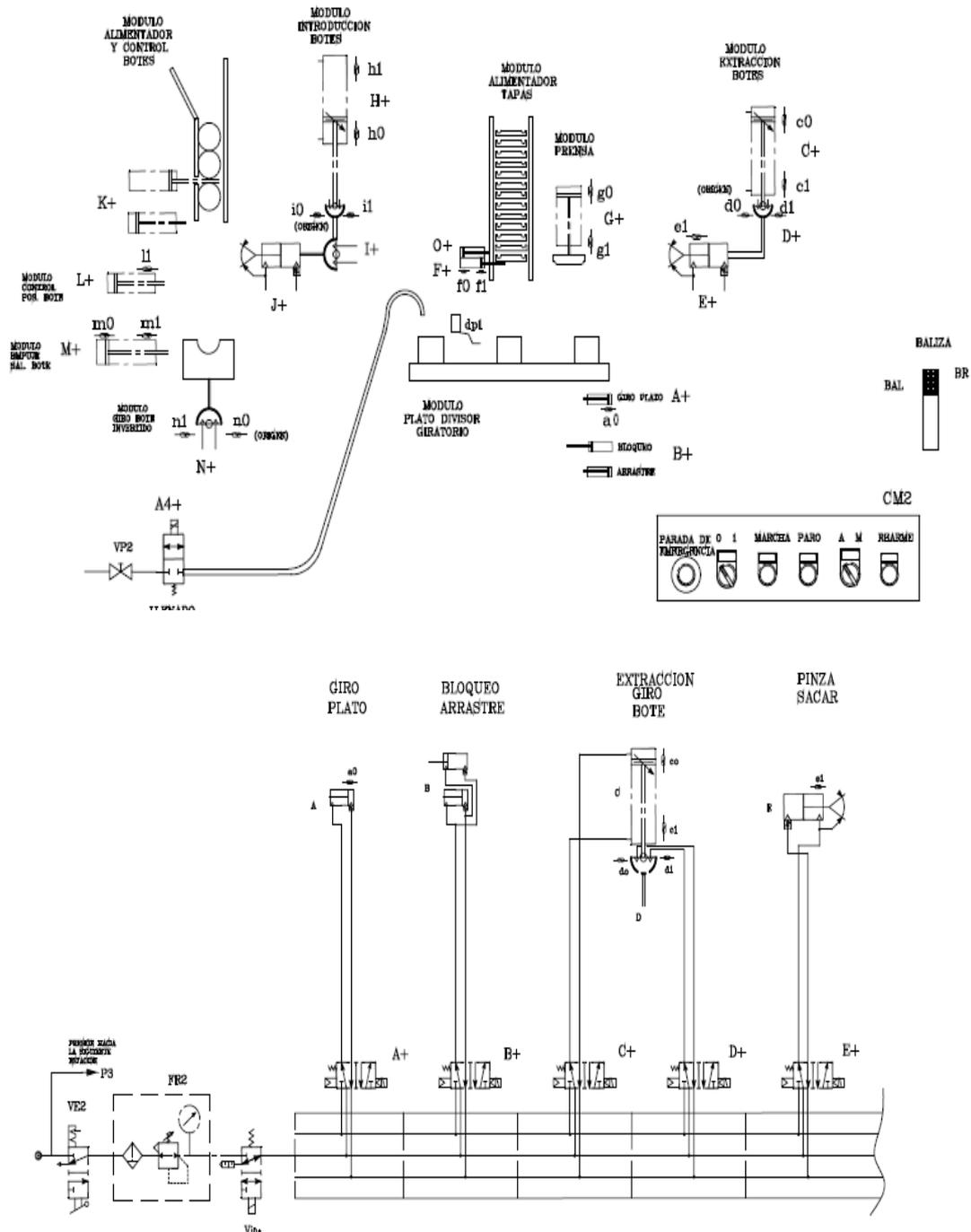
---

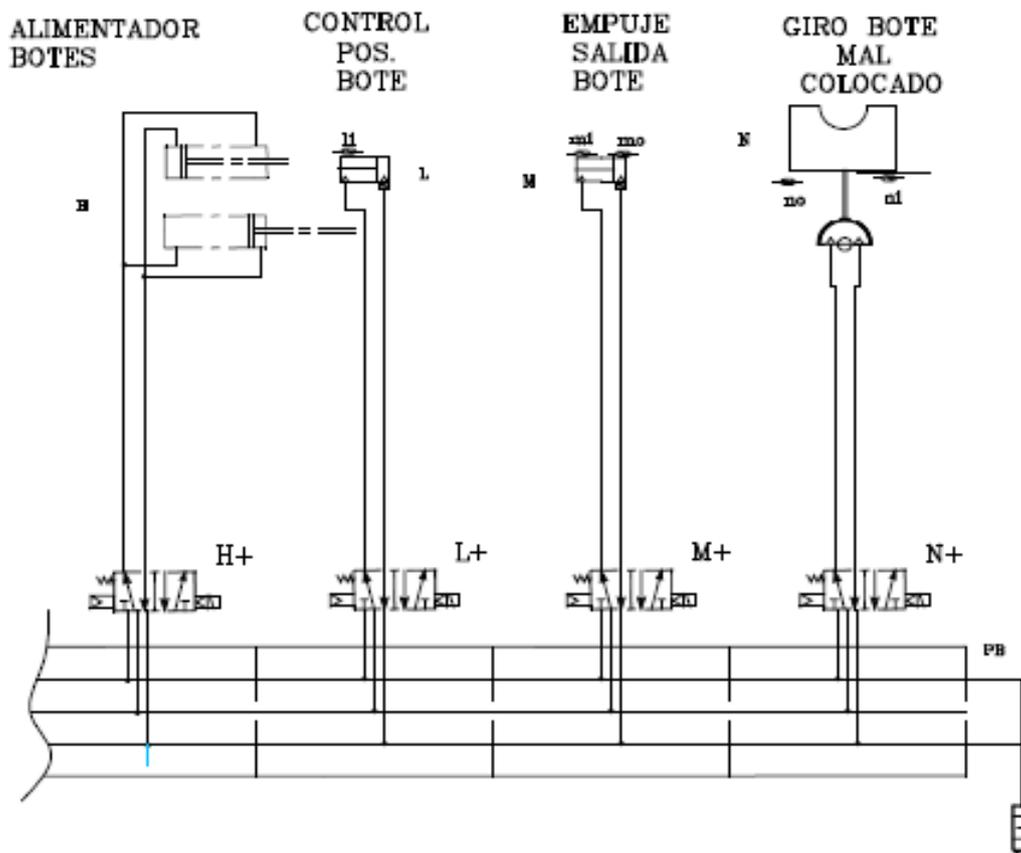
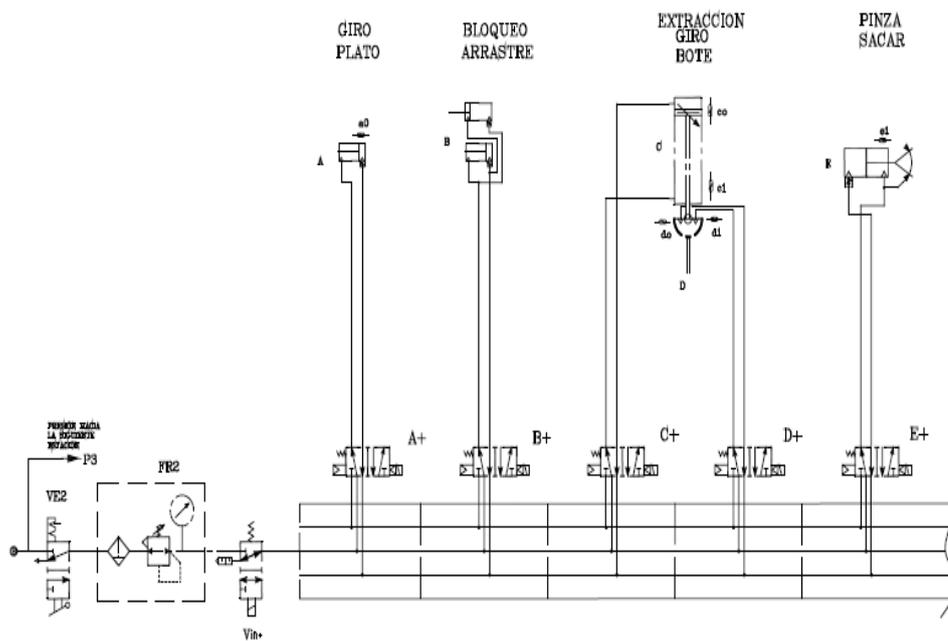
- Schneider Electric. (Marzo de 2005). Autómatas Modicon Micro TSX 3705/ 3708/ 3720. Manual de puesta en marcha.
- Schneider Electric. (Marzo de 2005). Manual de Referencia del software PL7 Micro/Junior/Pro. Descripción detallada de las Instrucciones y Funciones. Instrucciones de base.
- SMC International Training. Manual de Prácticas. Análisis neumático.
- SMC International Training. Manual de Prácticas. Análisis por bloques.
- SMC International Training. Manual de Prácticas. Instalación/montaje y puesta a punto.
- SMC International Training. Manual de Prácticas. Programación.
- SMC International Training. Manual de Usuario. Estaciones de proceso.
- SMC International Training. Manual de Usuario. Localización y corrección de averías.
- SMC International Training. Manual de Usuario. Seguridad.
- SMC International Training. Manual de Usuario - Anexo D.
- SMC International Training & J.R. Barreal. (s.f.). Manual de Usuario - Anexo A.
- Pérez, E. M. (2009). *Autómatas programables y sistemas de automatización*. Barcelona: MARCOMBO, S.A.
- SMC International Training. Manual de Usuario - Anexo E.
- SMC International Training. (2005). *Manual de referencia. PL7 Micro/Junior/pro. Descripción detallada de las instrucciones y funciones*.
- SMC International Training. (s.f.). Control de Procesos Industriales.
- SMC International Training. (s.f.). Guía de detectores magnéticos.



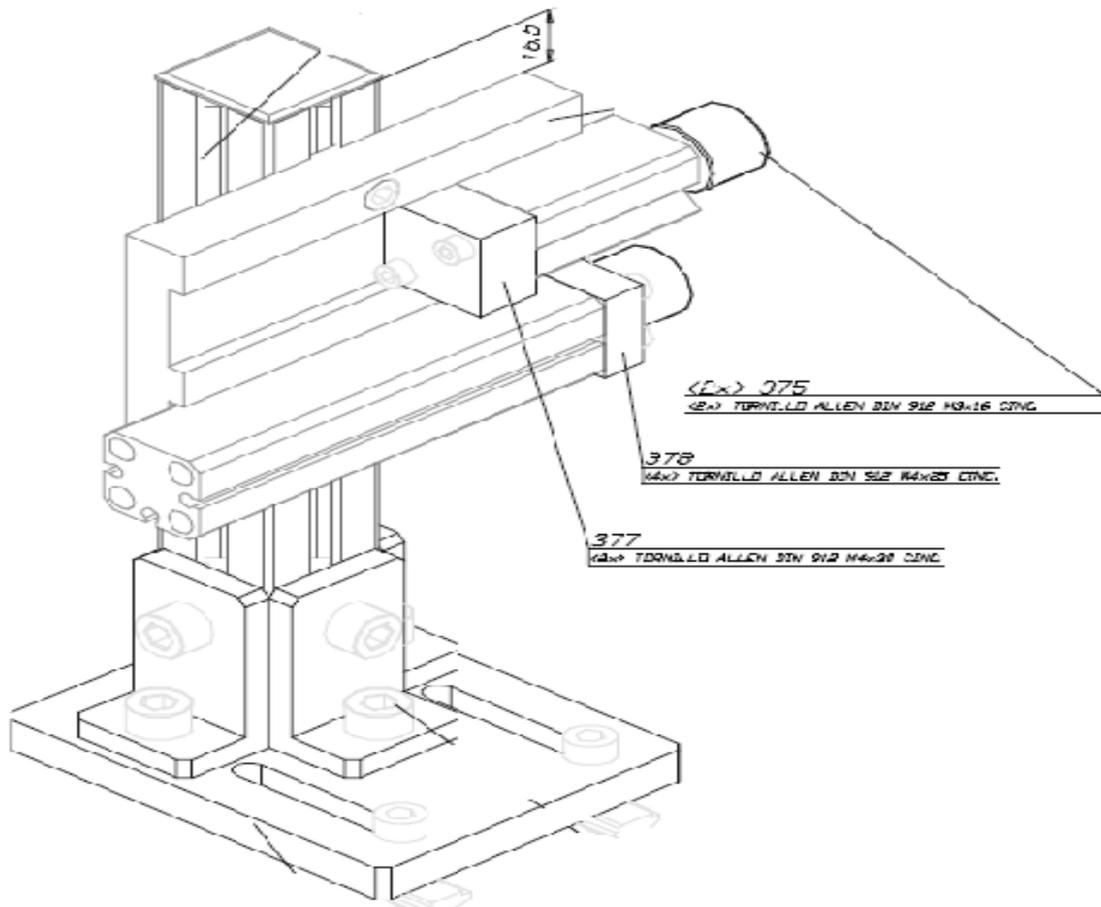
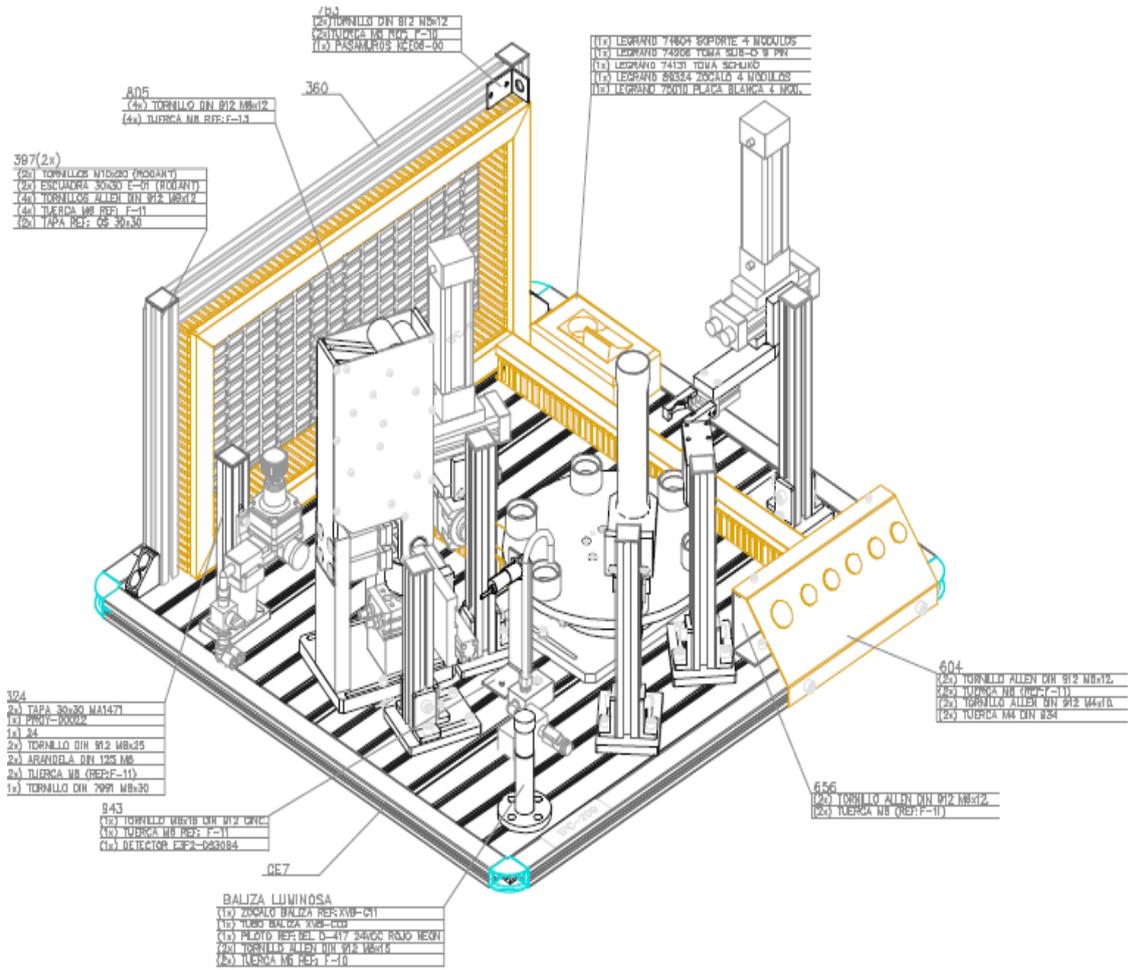
# 9 Anexo

## 9.1 Anexo A: Esquemas electro-neumáticos (SMC International training, Manual de Usuario - Anexo D)









# Automatización de un sistema de llenado de barriles

