



# Trabajo 2 - Línea de producción de vacunas y su embalaje

Carlos Mira López  
Nicolàs Miró Mira  
Vittorio Alessandro Esposito Ceballos

**Automatización Industrial**

**4.º curso - Grado en Informática Industrial y Robótica**

Diciembre del 2025

# Índice general

<b>Resumen</b>	<b>1</b>
<b>1 Metodología</b>	<b>1</b>
<b>2 Estructura TIA Portal</b>	<b>1</b>
2.1 Línea 1 . . . . .	1
2.2 Línea 2 . . . . .	2
2.3 Línea 3 . . . . .	2
2.4 Línea 4 . . . . .	3
2.5 Línea 5 . . . . .	4
2.6 Línea 6 . . . . .	4
2.7 Línea 7 . . . . .	4
<b>3 Grafsets</b>	<b>5</b>
3.1 Grafset Principal . . . . .	5
3.2 Grafset Paro . . . . .	5
3.3 Grafset Máquina 1 . . . . .	6
3.4 Grafset Máquina 2 . . . . .	7
3.5 Grafset Máquina 3 . . . . .	9
3.6 Grafset Máquina 4 . . . . .	10
3.7 Grafset Máquina 5 soporte . . . . .	12
3.8 Grafset máquina 5 vacuna y 6 soporte . . . . .	13
3.9 Grafset Máquina 5 embalaje . . . . .	16
3.10 Grafset Máquina 6 embalaje . . . . .	16
3.11 Grafset Máquina 7 embalaje . . . . .	17
3.12 Grafset de Mantenimiento . . . . .	18
3.13 Grafset Principal de Vaciado . . . . .	19
3.14 Grafset vaciado tipo 1 . . . . .	20
3.14.1 Grafset Vaciado 1 . . . . .	20
3.15 Grafset vaciado tipo 2 . . . . .	21
3.15.1 Grafset Vaciado 2 . . . . .	21
3.16 Gestión de las alarmas . . . . .	22
3.17 Gestión de la seta de emergencia . . . . .	23
<b>4 HMI</b>	<b>24</b>
4.1 Modelo utilizado y conexión con el PLC . . . . .	24
4.2 Pantalla principal . . . . .	25
4.3 Pantalla máquinas . . . . .	26
4.4 Pantalla configuración . . . . .	26
4.5 Usuarios . . . . .	27
4.6 Registro de alarmas . . . . .	27
<b>5 Resultados</b>	<b>28</b>
<b>6 Anexos</b>	<b>28</b>
6.1 Anexo 1: Grafset_0_Principal.scl . . . . .	28
6.2 Anexo 2: Grafset_10.scl . . . . .	31
6.3 Anexo 3: Grafset_10_1.scl . . . . .	33

6.4 Anexo 4: Grafcet_20.scl	35
6.5 Anexo 5: Grafcet_20_1.scl	38
6.6 Anexo 6: Grafcet_30.scl	40
6.7 Anexo 7: Grafcet_40.scl	43
6.8 Anexo 8: Grafcet_50.scl	45
6.9 Anexo 9: Grafcet_70.scl	48
6.10 Anexo 10: Grafcet_80.scl	51
6.11 Anexo 11: Grafcet_90.scl	53
6.12 Anexo 12: Grafcet_100.scl	55
6.13 Anexo 13: Grafcet_110.scl	58
6.14 Anexo 14: Grafcet_120.scl	60
6.15 Anexo 15: Grafcet_125.scl	61
6.16 Anexo 16: Grafcet_130.scl	62
6.17 Anexo 17: Grafcet_145.scl	64

# Índice de figuras

1	Estructura de la Línea 1 . . . . .	2
2	Estructura de la Línea 2 . . . . .	2
3	Estructura de la Línea 3 . . . . .	3
4	Estructura de la Línea 4 . . . . .	3
5	Estructura de la Línea 4 . . . . .	3
6	Estructura de la Línea 5 . . . . .	4
7	Estructura de la Línea 6 . . . . .	4
8	Estructura de la Línea 7 . . . . .	4
9	Grafcet Principal . . . . .	5
10	Grafcet del botón de Paro . . . . .	6
11	Máquina 1 de la planta . . . . .	6
12	Grafcet de la Máquina 1 . . . . .	7
13	Máquina 2 de la planta . . . . .	8
14	Grafcet de la Máquina 2 . . . . .	9
15	Máquina 3 de la planta . . . . .	9
16	Grafcet de la Máquina 3 . . . . .	10
17	Máquina 4 de la planta . . . . .	11
18	Grafcet de la Máquina 4 . . . . .	11
19	Máquina 5 de la línea de soportes . . . . .	12
20	Grafcet de la Máquina 5 Soportes . . . . .	13
21	Máquina 5 de la línea de soportes . . . . .	14
22	Grafcet de la Máquina 5 de vacunas y 6 de soportes . . . . .	15
23	Grafcet de la Máquina 5 Embalaje . . . . .	16
24	Máquina 6 de la línea de embalaje . . . . .	16
25	Grafcet de la Máquina 6 Embalaje . . . . .	17
26	Máquina 7 de la línea de embalaje . . . . .	17
27	Grafcet de la Máquina 7 embalaje . . . . .	18
28	Grafcet de Mantenimiento . . . . .	19
29	Grafcet Principal de Vaciado . . . . .	20
30	Grafcet de vaciado tipo 1 . . . . .	21
31	Grafcet de vaciado tipo 2 . . . . .	22
32	Grafcet de alarmas en guía GEMMA . . . . .	23
33	Grafcet de la gestión de la seta . . . . .	24
34	Grafcet de la seta de emergencia en guía GEMMA . . . . .	24
35	Pantalla principal del HMI . . . . .	25
36	Pantalla de detalle de una máquina . . . . .	26
37	Pantalla de configuración . . . . .	27
38	Registro de alarmas . . . . .	27

## 1 Metodología

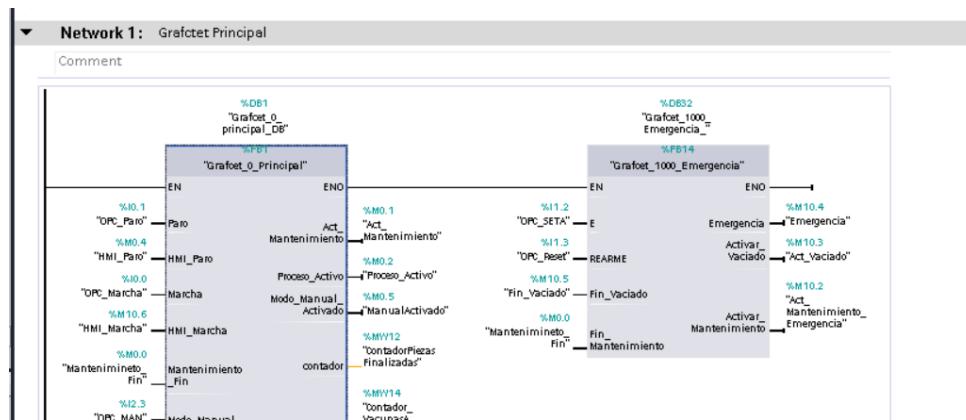
- Análisis del problema
- Crear el entorno virtual
- Comunicar el PLC con el entorno de la máquina en 3D
- Generar los grafsets para el proceso normal
- Implementar y evaluar la secuencia normal en el PLC
- Creación control con Marcha-Paro
- Implementar y evaluar el Marcha-Paro
- Generar el grafset de mantenimiento
- Implementar y evaluar el mantenimiento
- Generar los grafsets de Alarmas
- Implementar y evaluar las Alarmas
- Generar el grafset de Emergencia
- Implementar y evaluar la Emergencia
- Generar el grafset de vaciado
- Implementar y evaluar el vaciado
- Crear el HMI
- Implementar y evaluar el HMI

## 2 Estructura TIA Portal

Respecto a la estructura del bloque main del proyecto en TIA Portal, se ha seguido una metodología de dividir el bloque main en varias ramas, donde cada rama corresponde a una línea o función del sistema.

### 2.1 Línea 1

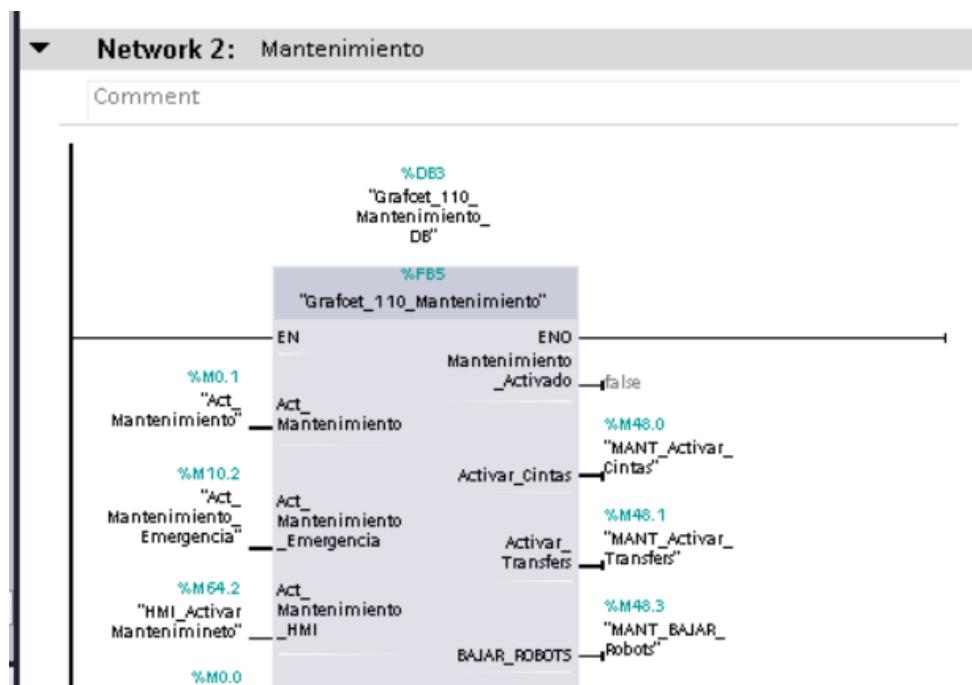
En la línea 0 se encuentra el grafset principal del sistema, que se encarga de coordinar el funcionamiento de todas las máquinas y líneas del sistema.



**Figura 1:** Estructura de la Línea 1

## 2.2 Línea 2

En la línea 2 vemos el bloque que corresponde con el modo de mantenimiento.



*Figura 2: Estructura de la Línea 2*

### 2.3 Línea 3

En esta línea podemos encontrar el modo manual del sistema.

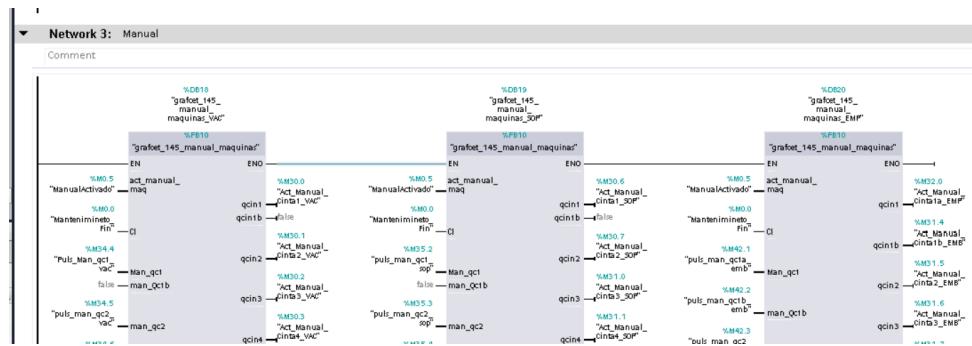


Figura 3: Estructura de la Línea 3

## 2.4 Línea 4

En la siguiente línea encontramos el bloque que gestiona el vaciado general del sistema, y más ramas donde cada una controla el vaciado de cada línea de producción.

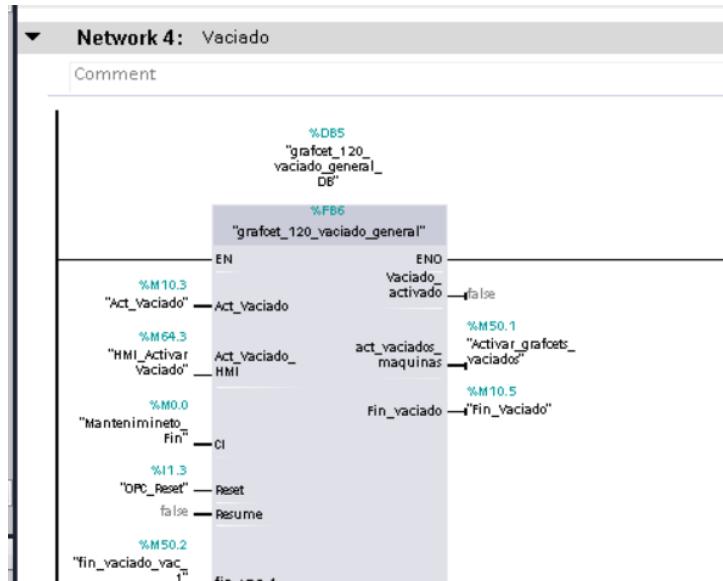


Figura 4: Estructura de la Línea 4

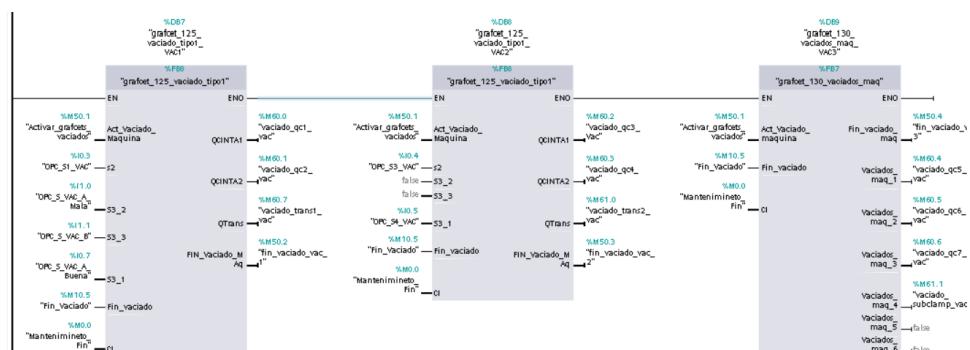
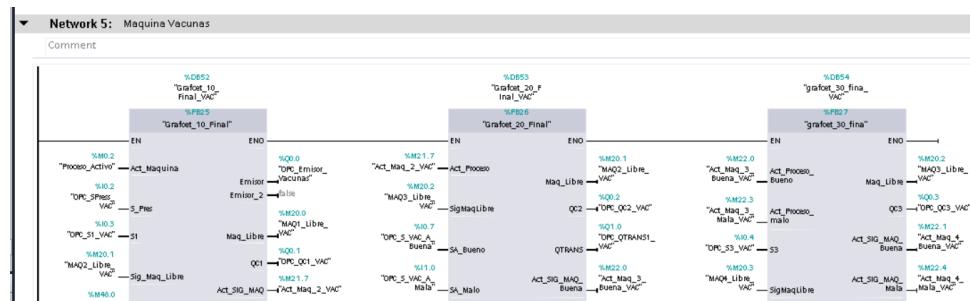


Figura 5: Estructura de la Línea 4

## 2.5 Línea 5

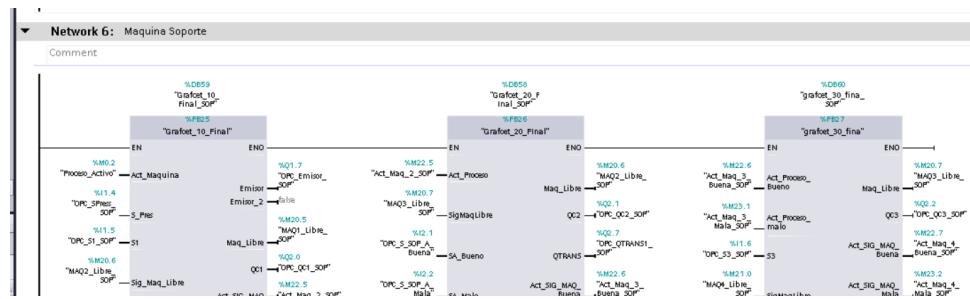
Aquí encontramos todos los bloques que controlan la línea de las vacunas.



*Figura 6: Estructura de la Línea 5*

## 2.6 Línea 6

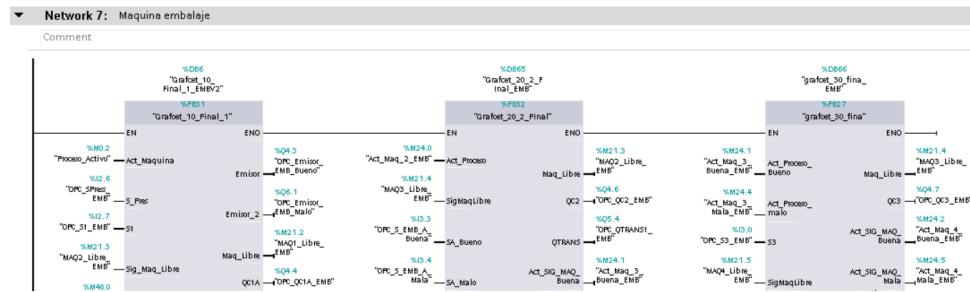
Aquí encontramos todos los bloques que controlan la línea de los soportes.



**Figura 7:** Estructura de la Línea 6

## 2.7 Línea 7

Aquí encontramos todos los bloques que controlan la línea de embalaje.

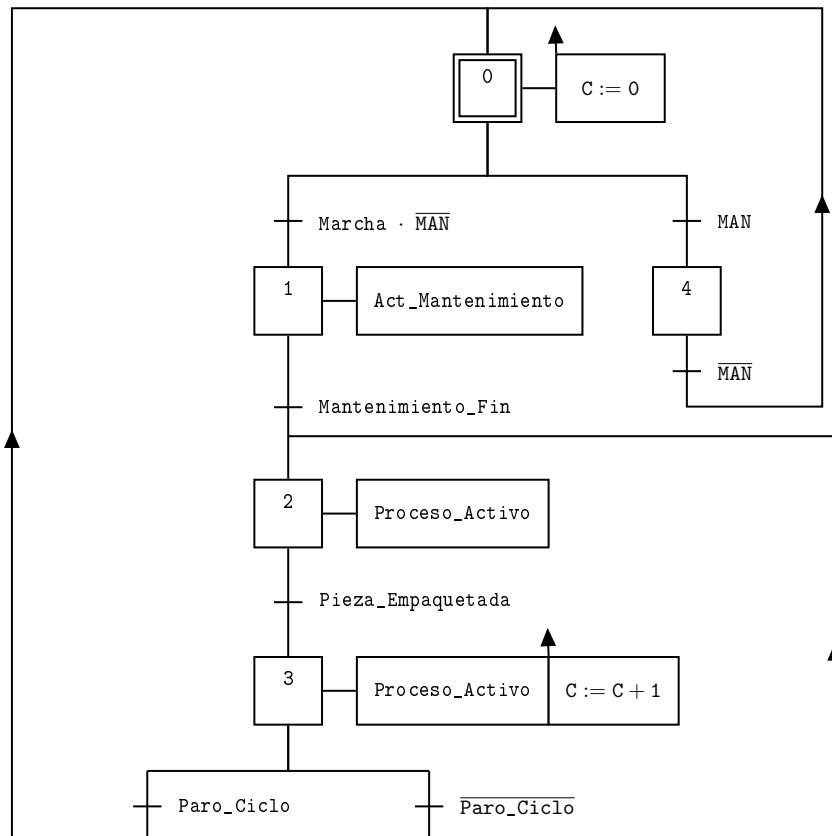


**Figura 8:** Estructura de la Línea 7

## 3 Grafcets

### 3.1 Grafcet Principal

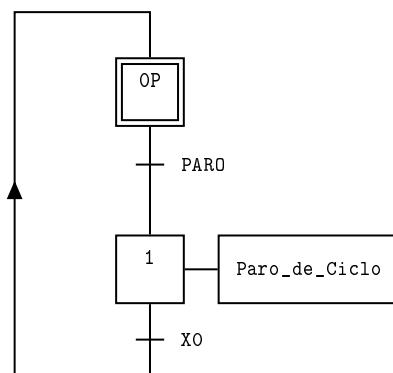
En la siguiente figura ([Grafcet Principal](#)), vemos el grafcet principal, el cual se encarga de poner la máquina en marcha, una vez no hay emergencia, funciona de manera que si modificamos el selector a modo manual, se va a activar el modo manual hasta que lo desactivemos, que entonces ahí, si apretamos marcha, empezará la máquina con el ([Grafcet de mantenimiento](#)), y posteriormente activará ya las primeras máquinas de cada objeto.



*Figura 9: Grafcet Principal*

### 3.2 Grafcet Paro

En la siguiente figura ([Grafcet Paro](#)) observamos el grafcet que trata el botón del paro, cuando apretamos este botón activamos una marca que recibe el grafcet principal para que se deje de emitir piezas y se queden las que están ya en la máquina.

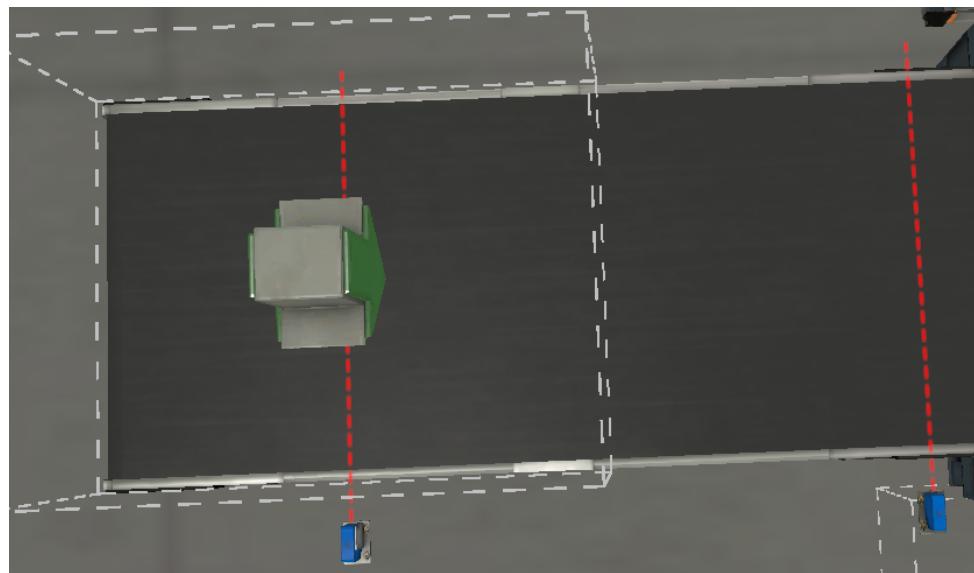


**Figura 10:** Grafcet del botón de Paro

### 3.3 Grafcet Máquina 1

Como vemos en la siguiente figura ([Máquina 1](#)), esta máquina está compuesta por:

- 1 Cinta para Vacunas y Soportes, 2 Cintas para los embalajes.
- 1 Sensor de presencia al principio de la cinta, o al final de la segunda en los embalajes.
- 1 Sensor final de la cinta
- 1 Emisor de objetos.

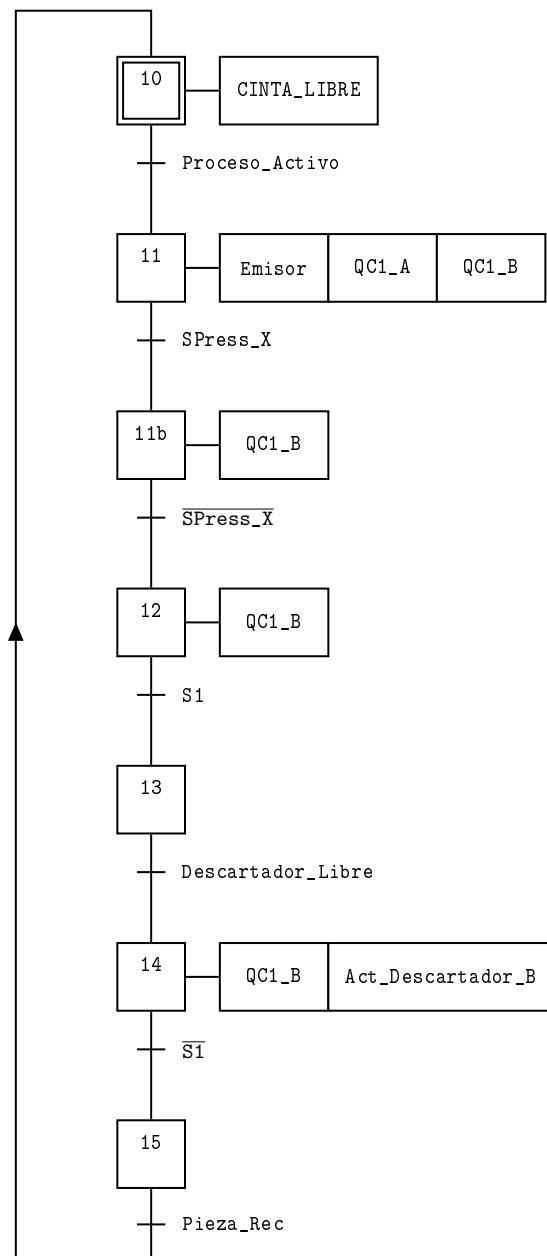


**Figura 11:** Máquina 1 de la planta

Con estos elementos conformamos el siguiente grafcet ([Grafcet Máquina 1](#)), el cual su funcionamiento es generar un objeto enviarlo al final de la cinta, y si la siguiente máquina esta libre entregarselo.

Las alarmas que tenemos en esta máquina que la detendrían son:

- Si cuando el objeto esta de camino al sensor del final de la cinta, se pone otro objeto al principio de la cinta.
- Si el objeto desde que se ha generado hasta el final de la cinta tarda más de 8 segundos.



*Figura 12: Grafset de la Máquina 1*

### 3.4 Grafset Máquina 2

Como vemos en la siguiente figura (Máquina 2), esta máquina esta compuesta por:

- 1 Transfer, que puede desplazar hacia el lado, o ir recto.
- 1 Sensor de visión, o 3 sensores en la parte del embalaje.



**Figura 13:** Máquina 2 de la planta

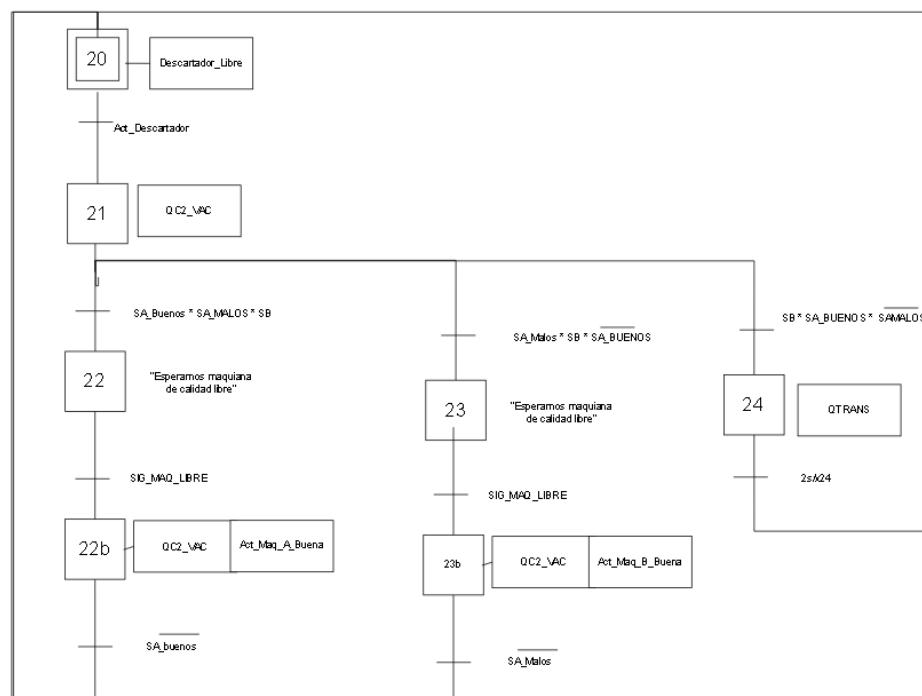
Como vemos en el grafset ([Grafset Máquina 2](#)), en esta máquina cuando nos entrega la pieza la máquina anterior, determinamos 3 caminos:

- Determinamos si la pieza es del tipo A bueno, que nos esperamos a que la siguiente máquina esté disponible.
- Determinamos si la pieza es del tipo A malo, que seguimos a la espera de que la siguiente máquina siga disponible, pero le avisaremos de que la pieza que está de camino es mala .
- Determinamos si es del tipo B, que lo que hacemos es descartarla con un movimiento lateral del transfer.

Respecto a los 3 sensores en la máquina de embalaje, esta hecho por la siguiente razón:

- Si se detectan los 3 sensores al final del transfer, es el tipo A bueno.
- Si al final de la cinta solo se detectan los dos sensores de abajo es el tipo A malo.
- Si solo se detectan los dos sensores finales, que están uno encima del otro determinamos que es tipo B y se debe de descartar.

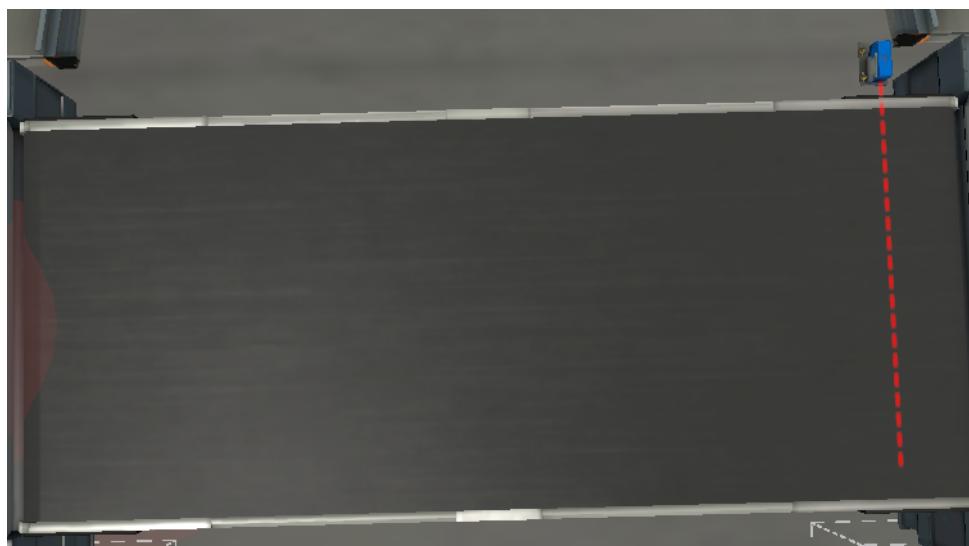
En esta máquina solo se trata una alarma, que es que tardemos más de 5 segundos en recibir la pieza de la máquina anterior.

*Figura 14: Grafcet de la Máquina 2*

### 3.5 Grafcet Máquina 3

Como vemos en la siguiente figura ([Máquina 3](#)), esta máquina esta compuesta por:

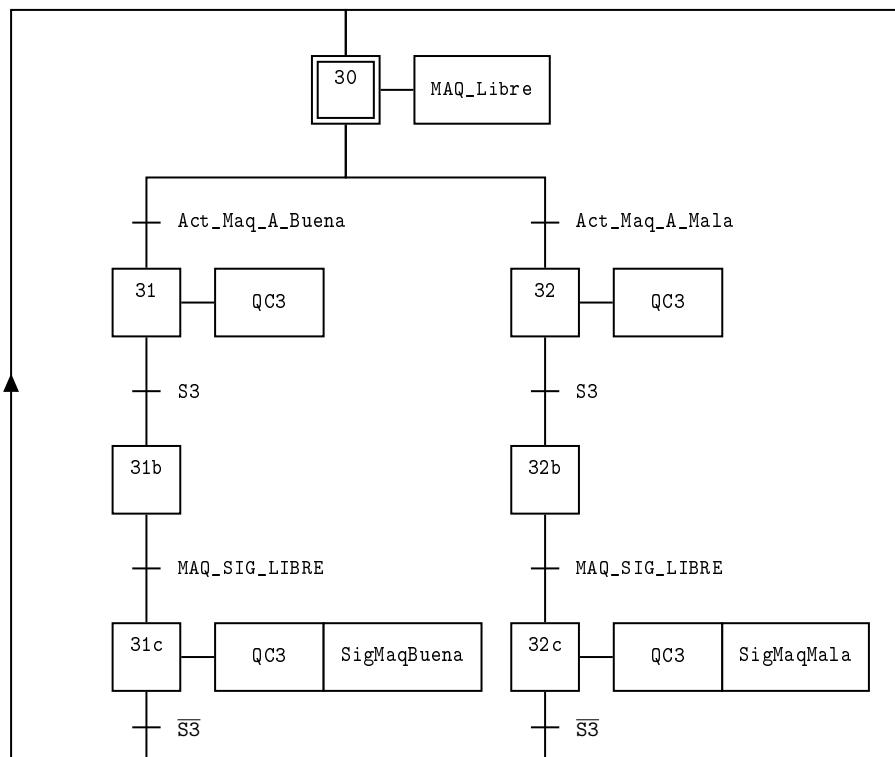
- 1 cinta.
- 1 Sensor al final de la cinta.

*Figura 15: Máquina 3 de la planta*

Como vemos en el grafcet ([Grafcet Máquina 3](#)), en esta máquina cuando nos entrega la pieza la máquina anterior, esta solo será del tipo A, pero nos indicará si es buena o mala, para poder indicarle a la máquina siguiente si la tiene que descartar o no.

El funcionamiento es recibir la pieza indicando si es buena o mala, llevarla hasta el final donde estará el sensor y si la siguiente máquina está libre entregarla.

En esta máquina solo se trata una alarma, que es que tardemos más de 10 segundos en recorrer toda la cinta



*Figura 16: Grafcet de la Máquina 3*

### 3.6 Grafcet Máquina 4

Como vemos en la siguiente figura ([Máquina 4](#)), esta máquina está compuesta por:

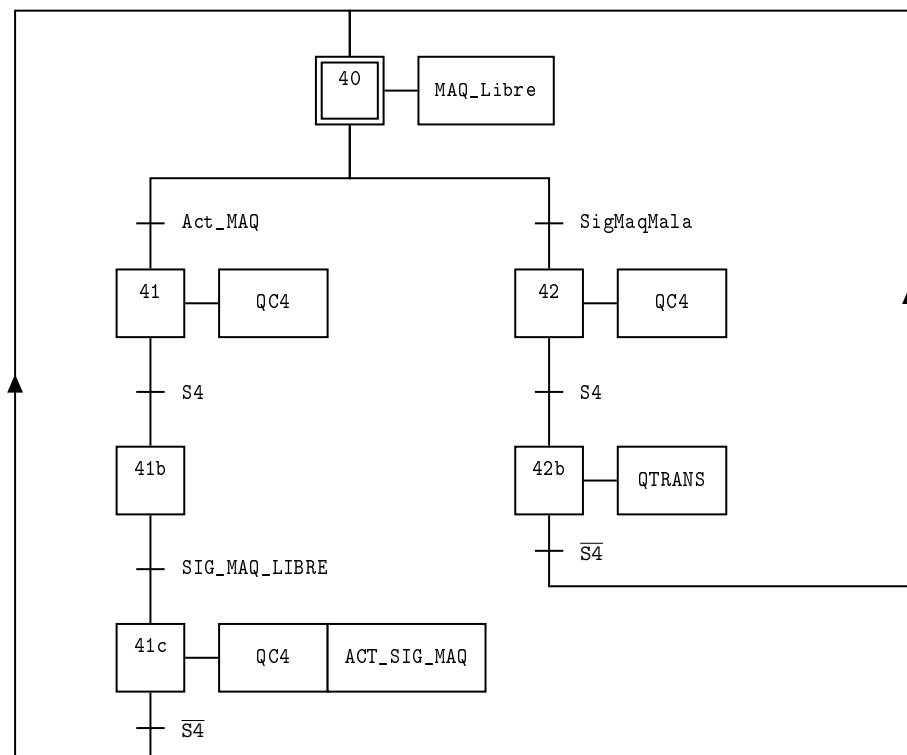
- 1 transfer, que puede mover hacia un lado o ir en linea recta
- 1 Sensor al final de la cinta.



**Figura 17:** Máquina 4 de la planta

Como vemos en el grafcet ([Grafcet Máquina 4](#)), en esta máquina cuando nos entrega la pieza la máquina anterior, esta vendrá indicada si es buena o mala, por lo que al llegar al sensor si es buena se esperará a que la siguiente máquina este libre, y si es mala, la descartaremos.

En esta máquina solo se trata una alarma, que es que tardemos más de 5 segundos en llegar al sensor desde que nos indican que nos activemos

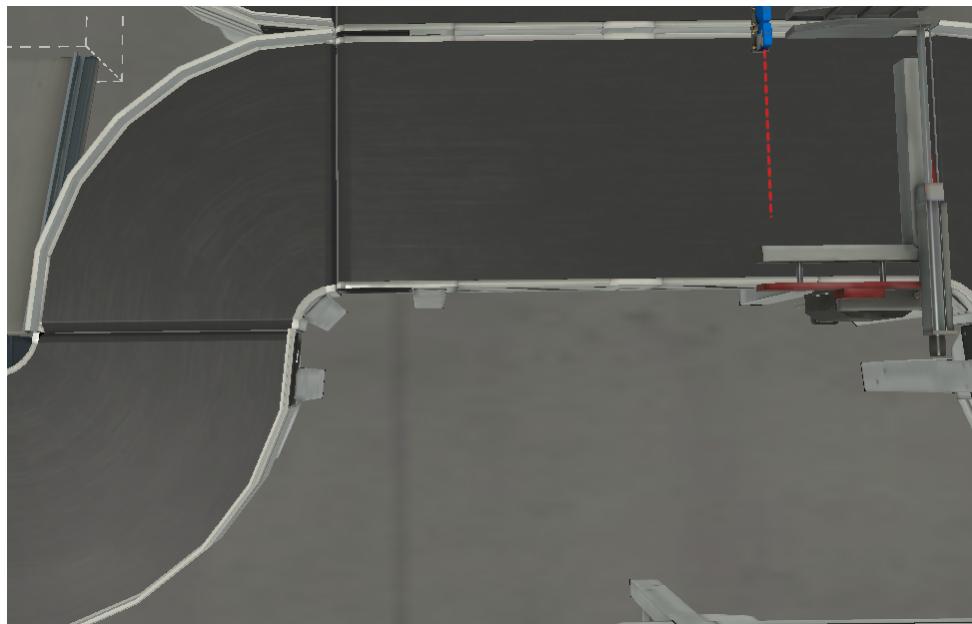


**Figura 18:** Grafcet de la Máquina 4

### 3.7 Grafcet Máquina 5 soporte

En la siguiente figura ([Máquina 5 soporte](#)) vemos los siguientes elementos:

- 3 Cintas
- 1 Clamper
- 1 Sensor

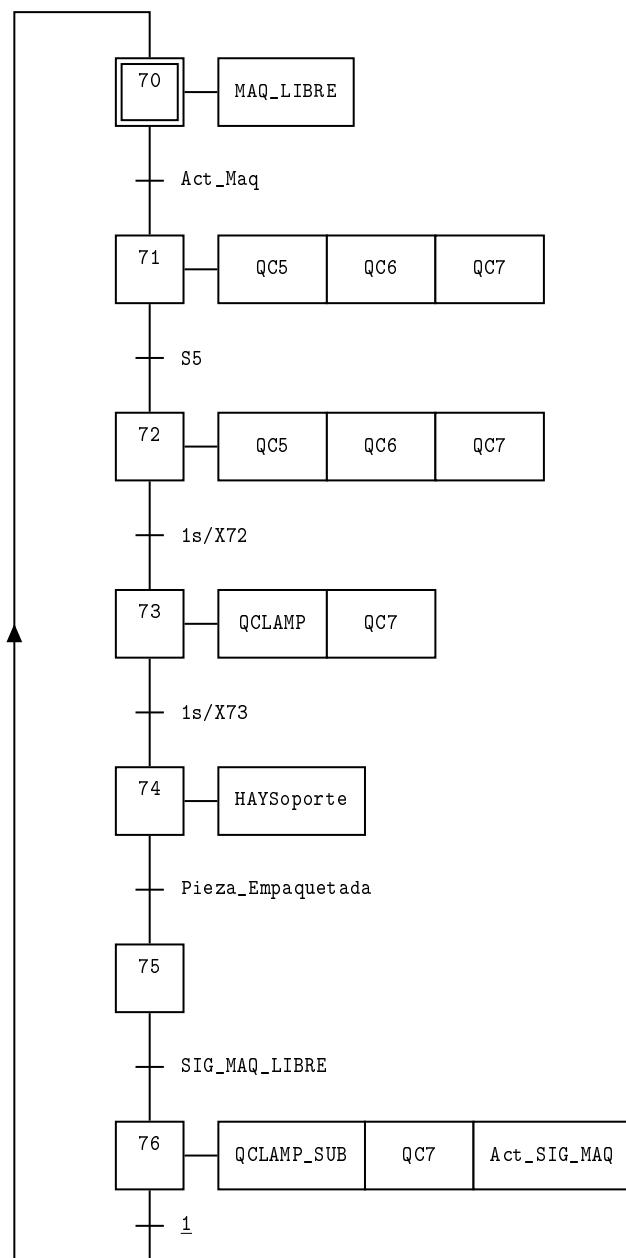


*Figura 19: Máquina 5 de la línea de soportes*

El funcionamiento de la máquina como vemos en el grafcet es el siguiente:

Activamos las 3 cintas, cuando se produce un flanco de bajada en el sensor que hay antes del clamper ponemos un temporizador de 1 segundo para que la pieza entre dentro del clamper. Activamos el clamper, esto lo que nos hace es colocar la pieza en el sitio, para que el robot al colocar la vacuna sobre el soporte, este no falle. Una vez tenemos la pieza colocada avisamos al robot de las vacunas que hay un soporte en el sitio, este aviso lo recibimos en el grafcet que hablaremos más adelante ([Grafcet Máquina 5 soporte](#)), y que nos colocará la pieza en su sitio. Una vez nos diga la otra máquina que tenemos la vacuna en el soporte, nos esperaremos a que la siguiente máquina nos indique que esté libre para poder levantar el clamper y activar las cintas.

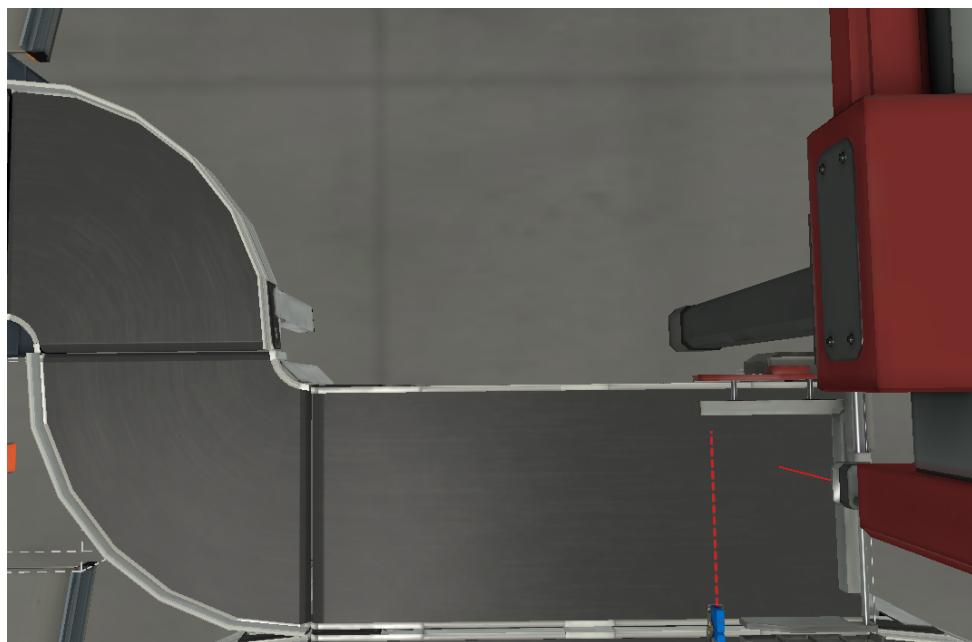
Aquí la alarma que tratamos es que la pieza tarde más de 15 segundos en llegar al sensor.

**Figura 20:** Grafset de la Máquina 5 Soportes

### 3.8 Grafset máquina 5 vacuna y 6 soporte

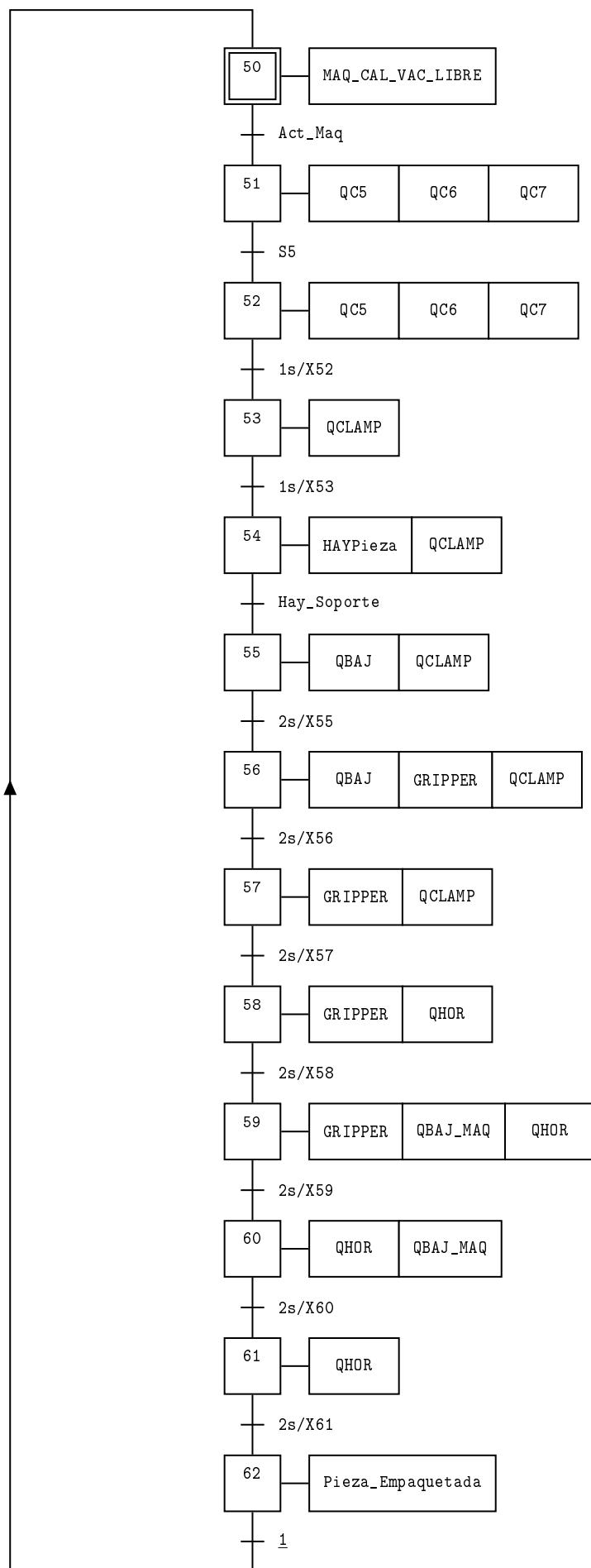
En la siguiente figura ([Máquina 5 vacuna y 6 soporte](#)) observamos la máquina que forman la máquina número 5 de la línea de vacunas, y la número 6 de la línea de soportes, conformada por los siguientes elementos:

- 3 Cintas
- 1 Robot de 2 Ejes (Y-Z), con un gripper



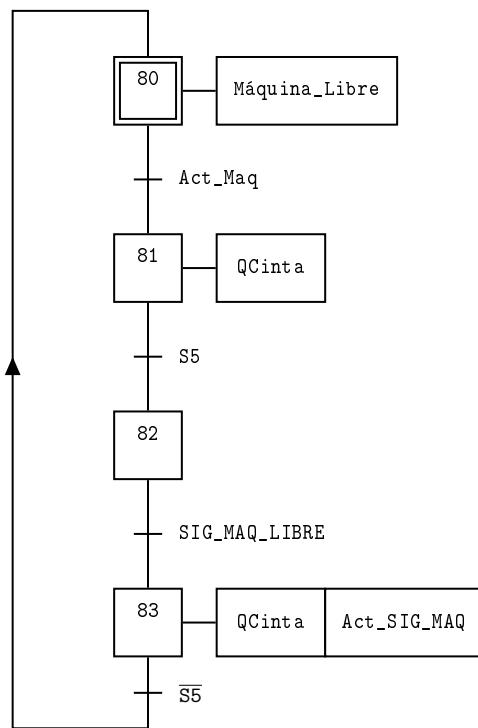
*Figura 21: Máquina 5 de la línea de soportes*

El funcionamiento que describe el grafcet ([Grafcet Máquina 5 vacuna y 6 soporte](#)) es el siguiente, activamos las 3 cintas hasta que se produzca un flanco de bajada del sensor que hay antes del clamper , cuando se produce el flanco nos esperamos 1 segundo, y activamos el clamper para que la pieza se quede colocada correctamente, y nos ponemos a la espera de que en la línea de vacunas nos avisen que tenemos un soporte en el sitio, o en la línea de soportes, tengamos un embalaje en posición. A partir de ahí bajamos el robot durante 2 segundos, activamos el gripper, y con el gripper activado, subimos el robot para después desplazarlo en horizontal y bajarlo, aquí ya soltaremos el gripper y devolveremos el robot a su posición de reposo.

**Figura 22:** Grafcet de la Máquina 5 de vacunas y 6 de soportes

### 3.9 Grafcet Máquina 5 embalaje

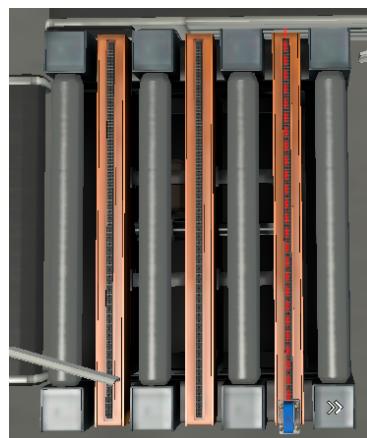
Esta máquina ([Máquina 5 Embalaje](#)) tiene un funcionamiento similar al indicado en la [Subsección de la Máquina 3](#), pero con un único camino, ya que su única función es llevar una pieza buena al final de la cinta, para entregársela al transfer siguiente.



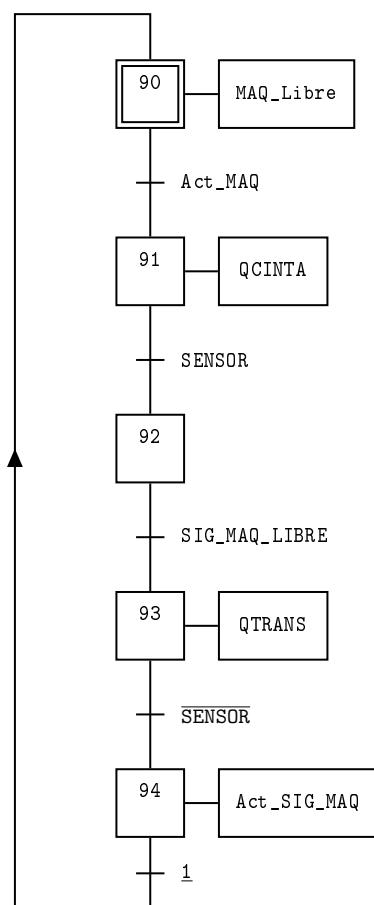
*Figura 23: Grafcet de la Máquina 5 Embalaje*

### 3.10 Grafcet Máquina 6 embalaje

En esta máquina ([Máquina 6 Embalaje](#)) la función es desplazar el embalaje a la cinta final del proceso, que se detalla más adelante.



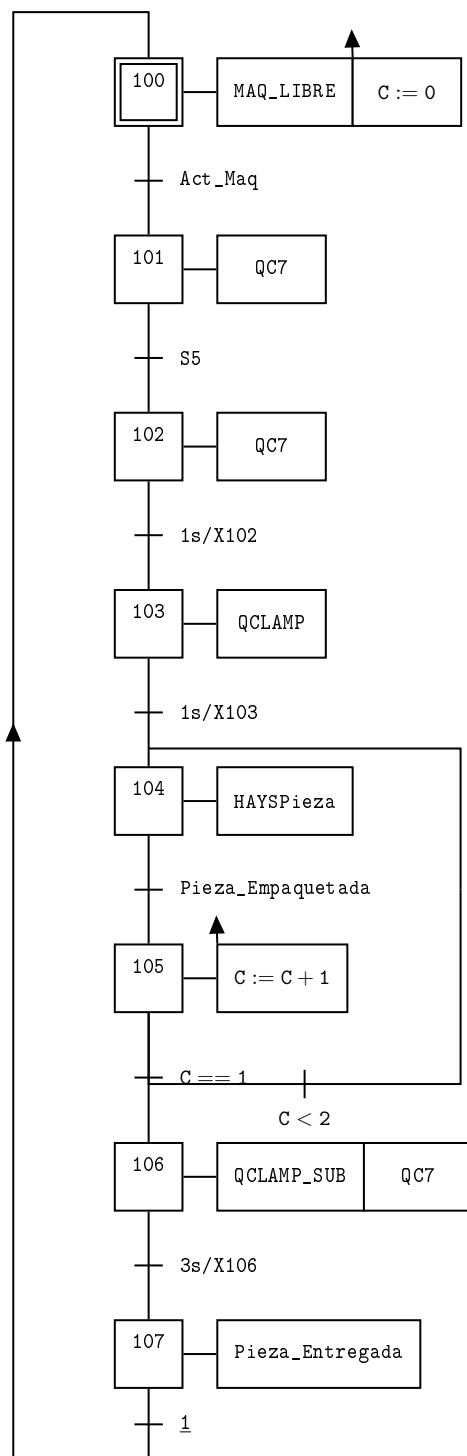
*Figura 24: Máquina 6 de la línea de embalaje*

*Figura 25: Grafcet de la Máquina 6 Embalaje*

### 3.11 Grafcet Máquina 7 embalaje

Esta máquina ([Máquina 7, embalaje](#)) es similar a la máquina 5 de la línea de soportes ([Sección de la Máquina 5 de Soportes](#)). Se cuenta el número de piezas embaladas para, al alcanzar el cupo, levantar el clamper y mover la caja a otra zona.

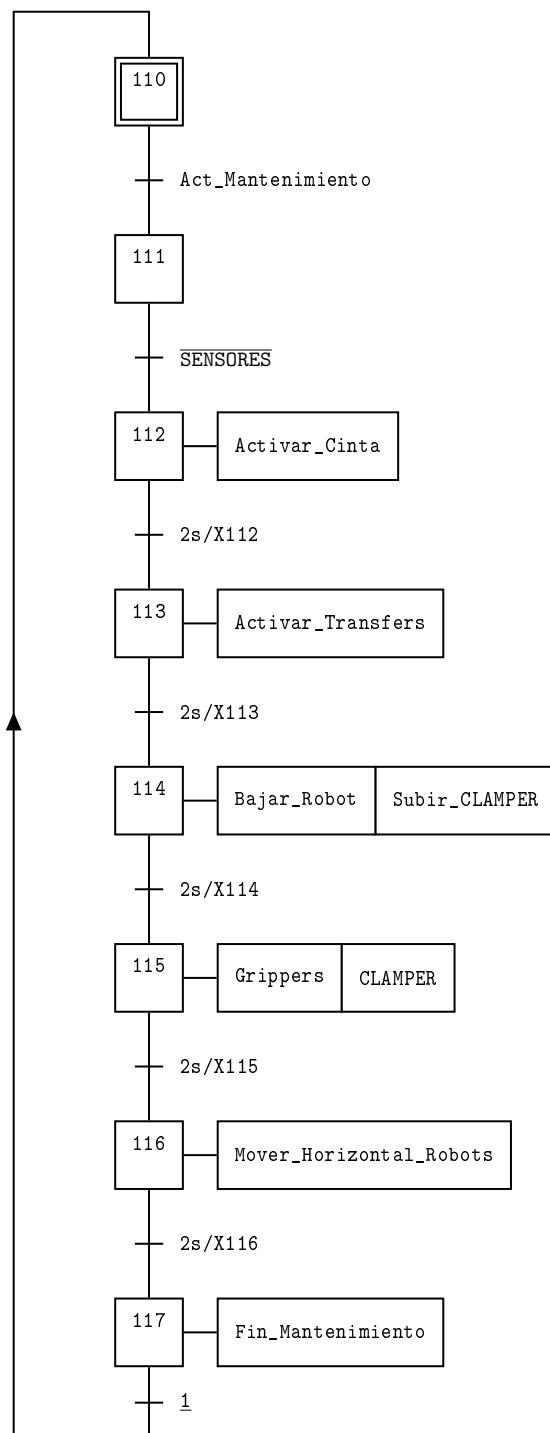
*Figura 26: Máquina 7 de la línea de embalaje*

*Figura 27: Grafcet de la Máquina 7 embalaje*

### 3.12 Grafcet de Mantenimiento

El [Grafcet de Mantenimiento](#) asegura que la máquina funciona correctamente antes de procesar lotes. Se ejecuta al arrancar la máquina y después de la ejecución del [Grafcet Principal de Vaciado](#), garantizando que la puesta en marcha tras soltar la seta sea correcta.

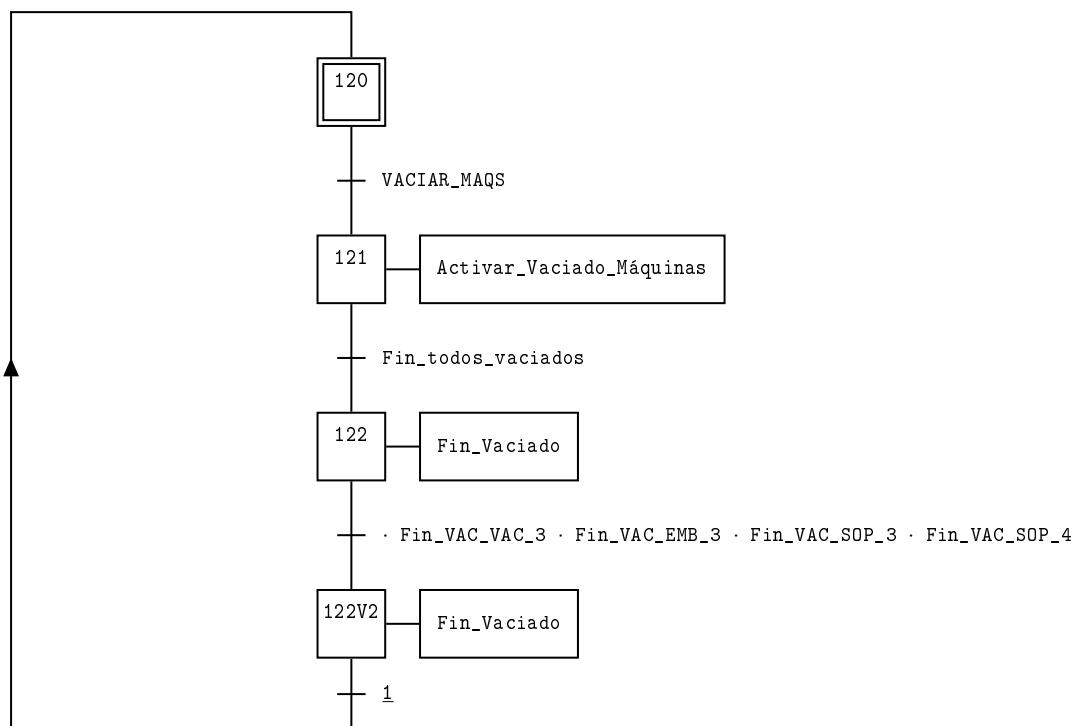
Dispone también de un botón en la pantalla de configuración del HMI (37) para que un usuario con permisos pueda activarlo en cualquier momento.



**Figura 28:** Grafcet de Mantenimiento

### 3.13 Grafcet Principal de Vaciado

El **Grafcet Principal de Vaciado** se activa tras desenclavar la seta de emergencia, ya que no se aceptan las piezas que estuvieran en la máquina durante la emergencia. Las máquinas también se pueden vaciar si un usuario con permisos pulsa el botón de vaciado del HMI (37).



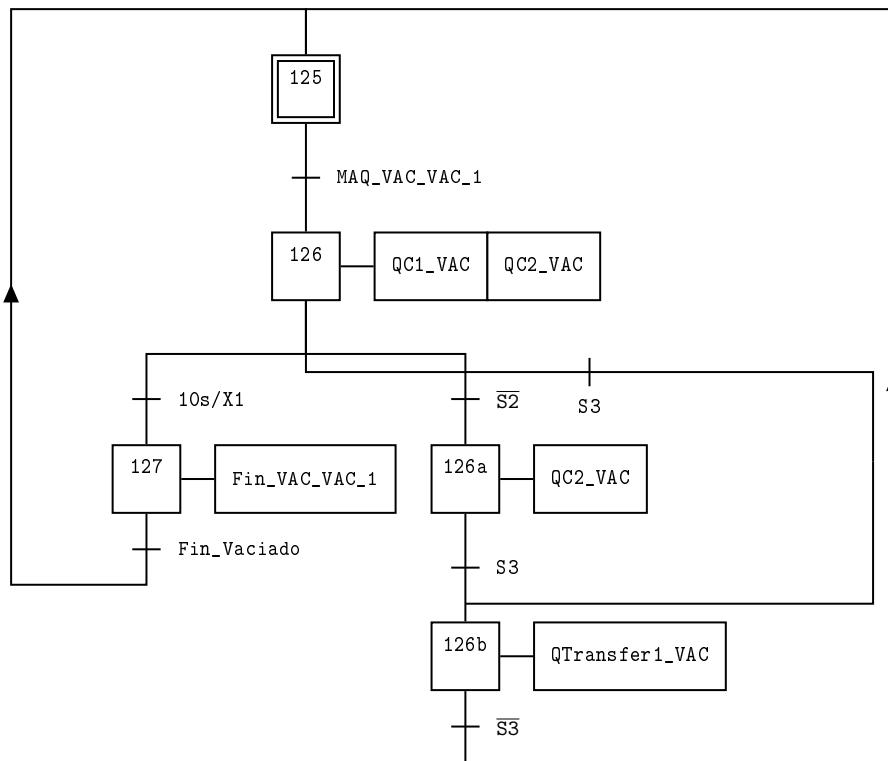
**Figura 29:** Grafcet Principal de Vaciado

### 3.14 Grafcet vaciado tipo 1

#### 3.14.1 Grafcet Vaciado 1

El **Grafcet de Vaciado 1** se activa por medio del [Grafcet Principal de Vaciado](#), y este vacía las siguientes máquinas:

- Máquina 1:
  - Vacunas, Soportes y Embalajes
- Máquina 2:
  - Vacunas, Soportes y Embalajes
- Máquina 3:
  - Vacunas, Soportes y Embalajes
- Máquina 4:
  - Vacunas, Soportes y Embalajes

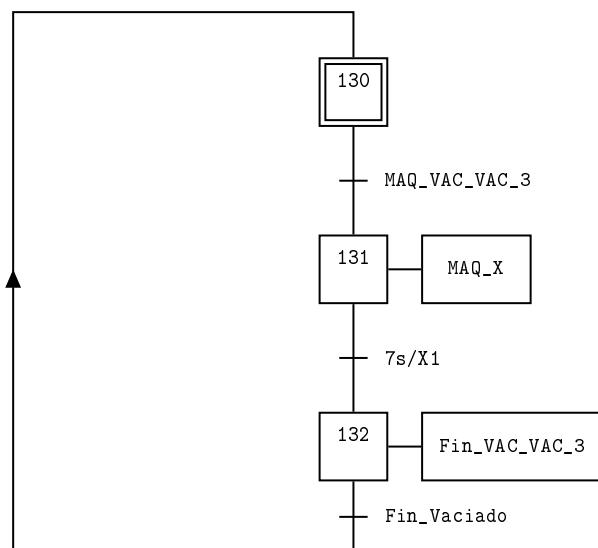
*Figura 30: Grafcet de vaciado tipo 1*

### 3.15 Grafcet vaciado tipo 2

#### 3.15.1 Grafcet Vaciado 2

El [Grafcet de Vaciado 2](#) se activa por medio del [Grafcet Principal de Vaciado](#), y este vacía las siguientes máquinas:

- Máquina 5 de Soportes:
  - [Soporte](#)
- Máquina 5 de vacunas y 6 de soportes:
  - [Vacunas y Soportes](#)
- Máquina 6 y 7 de embalaje:
  - [Embalajes](#)
  - [Embalajes](#)

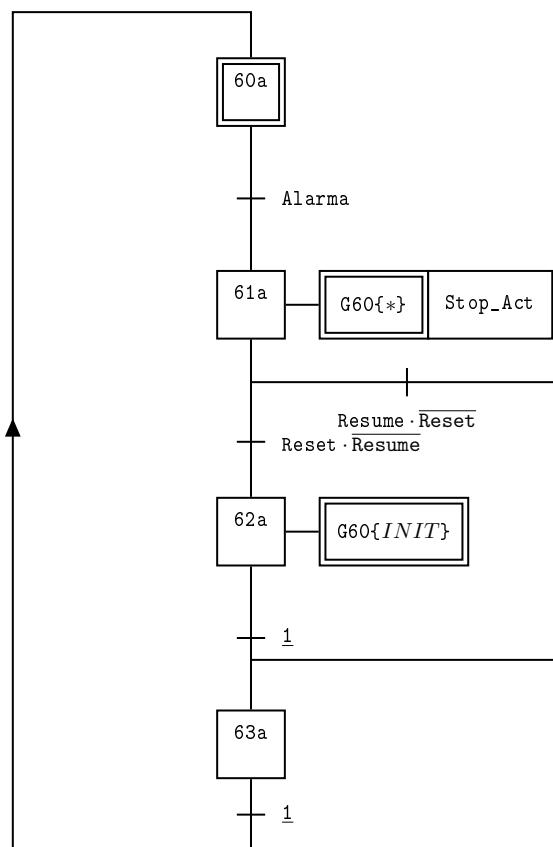


*Figura 31: Grafcet de vaciado tipo 2*

### 3.16 Gestión de las alarmas

El [Grafcet de alarmas en guía GEMMA](#) gestiona las alarmas de cada máquina, asegurándose de que la máquina se detenga cuando haya una alarma activa y permitiendo resetear la alarma cuando se haya solucionado el problema, o reanudarlo desde donde se havia parado si el problema era leve.

Este grafcet se encuentra implementado en todas las máquinas de la línea de producción, siguiendo la guia GEMMA para la estructura del código.

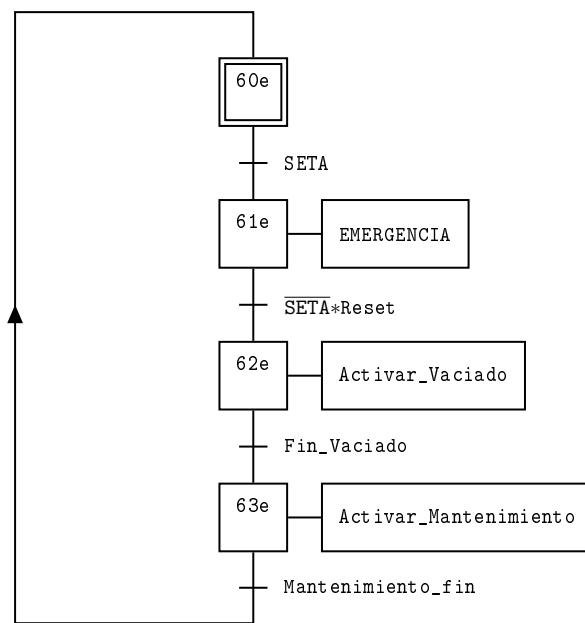


*Figura 32: Grafcet de alarmas en guía GEMMA*

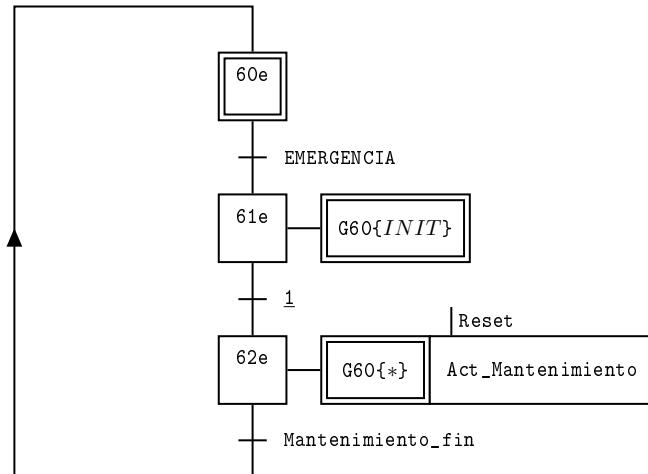
### 3.17 Gestión de la seta de emergencia

El [Grafcet de la seta de emergencia](#) gestiona la señal que produce la seta de emergencia, y todos los pasos que debe de seguir la máquina cuando se desenclava la seta para poder rearmar la máquina y seguir con el proceso de producción.

Lo primero que hacemos es activar la variable "EMERGENCIA" comunicando a todos las partes del programa que la seta se ha enclavado, y por lo tanto la máquina debe de detenerse. Lo siguiente es desenclavar la seta, y una vez desenclavada, pulsar "reset" para activar el vaciado de la máquina, ya que no se aceptan las piezas que estuvieran en la máquina durante la emergencia. Una vez finalizado el vaciado, se activa el mantenimiento para comprobar que todo funciona correctamente, dejando la máquina lista para volver a pulsar marcha o entrar a modo manual.

**Figura 33:** Grafcet de la gestión de la seta

El [Grafcet de la seta de emergencia en guía GEMMA](#), se encuentra implementado en todas las máquinas de la línea de producción, siguiendo la guía GEMMA para la estructura del código. Aunque como hemos hablado con anterioridad, las señales que le llegan a cada bloque de la máquina son que se ha enclavado la seta, y cuando ya se han pasado todas la verificaciones para reanudar el proceso.

**Figura 34:** Grafcet de la seta de emergencia en guía GEMMA

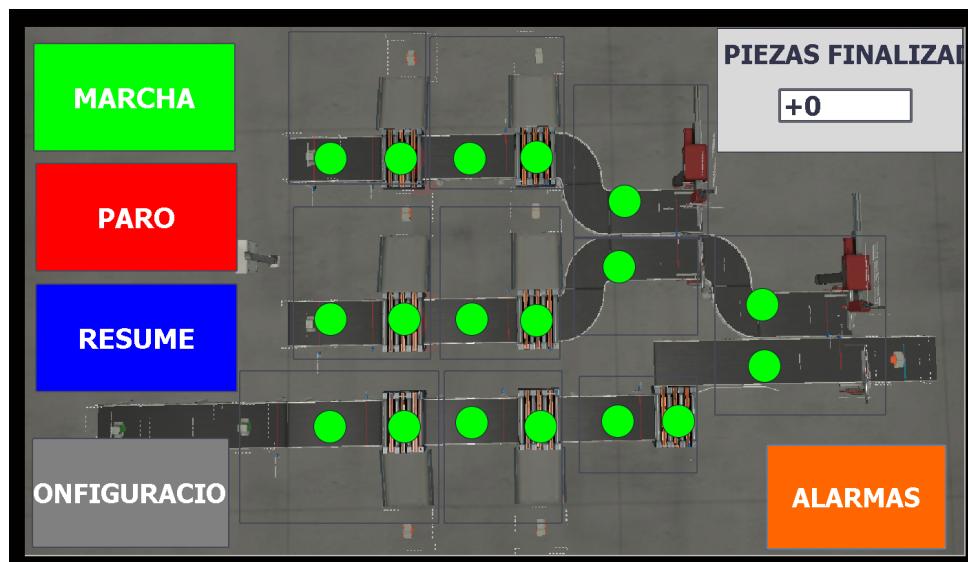
## 4 HMI

### 4.1 Modelo utilizado y conexión con el PLC

Para la realización del proyecto se ha escogido un HMI TP1500 Comfort pro de 15" conectada vía Profinet al PLC, para poder tener acceso a las variables.

## 4.2 Pantalla principal

En la siguiente figura podemos observar la pantalla principal, donde podemos observar distintos elementos.



*Figura 35: Pantalla principal del HMI*

Respecto a los botones tenemos:

- Marcha: Este botón hace la misma función que el botón del panel físico del Factory IO, pone en marcha la máquina si esta no está en alarmas o en emergencia.
- Paro: Al igual que la marcha, es un botón que está duplicado Respecto al cuadro físico que se encarga de parar la producción de piezas.
- Rearme: Este botón solo se encuentra en el HMI, y lo que hace es que cuando está la máquina en alarma, si vemos que con mover un poco la pieza la máquina puede seguir, apretaremos este botón sin necesidad de apretar el "RESET".
- Configuración: Para pulsar este botón se pedirán credenciales; este apartado solo está disponible para administradores y da acceso a la pantalla de configuración (véase figura 37).
- Alarma: Este botón abre la pantalla del registro de alarmas (véase figura 38).

Aparte de los botones, en la parte superior derecha vemos que hay un contador de embalajes completados que ya han acabado el proceso de producción.

En el centro de la pantalla vemos una foto de todo el proceso con luces en cada máquina que forman el proceso completo. Las luces tienen los siguientes estados:

- Verde: Máquina libre y preparada para recibir piezas.
- Amarillo: Máquina funcionando en modo automático.
- Rojo: Máquina en alarma.

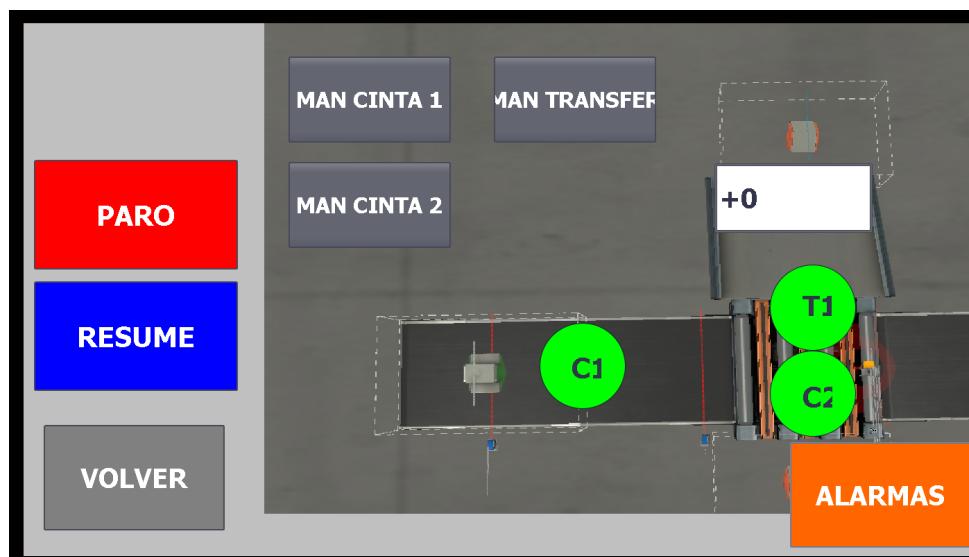
- Rojo parpadeante amarillo : Máquina en emergencia.

A parte de los colores, si pulsamos sobre cualquier parte de la máquina nos llevará a una ventana donde podremos ver qué actuador está en funcionamiento en ese momento, cuál está parado, o el contador de piezas descartadas.

### 4.3 Pantalla máquinas

En la siguiente figura (36) podemos ver como es una pantalla cuando sobre la principal pulsamos sobre una máquina, vemos que podemos ver dos máquinas, unas luces encima de los actuadores que nos indican si están funcionando o no, y un contador donde se puede ver el número de piezas descartadas en esa máquina.

Vemos que también hay unos botones con los nombres de los actuadores, esto es porque si tenemos el modo manual activado, estos botones se harán visibles y podremos accionar los actuadores desde aquí.



*Figura 36: Pantalla de detalle de una máquina*

### 4.4 Pantalla configuración

En la siguiente figura (37) podemos ver la pantalla de configuración a la cual sólo se puede acceder a través de credenciales de administrador. Vemos que hay una columna central donde podemos ver:

- Un campo de entrada, donde pondremos el número de vacunas por cajas que queremos
- Un botón que nos activa el mantenimiento, por si queremos comprobar que todo funciona
- Un botón que activa el vaciado de la máquina, por si en cualquier momento queremos vaciarla sin necesidad de la emergencia.

También disponemos de un botón de cerrar sesión, por si ya hemos configurado la máquina y probado, y ya el próximo que la toque es un trabajador normal.

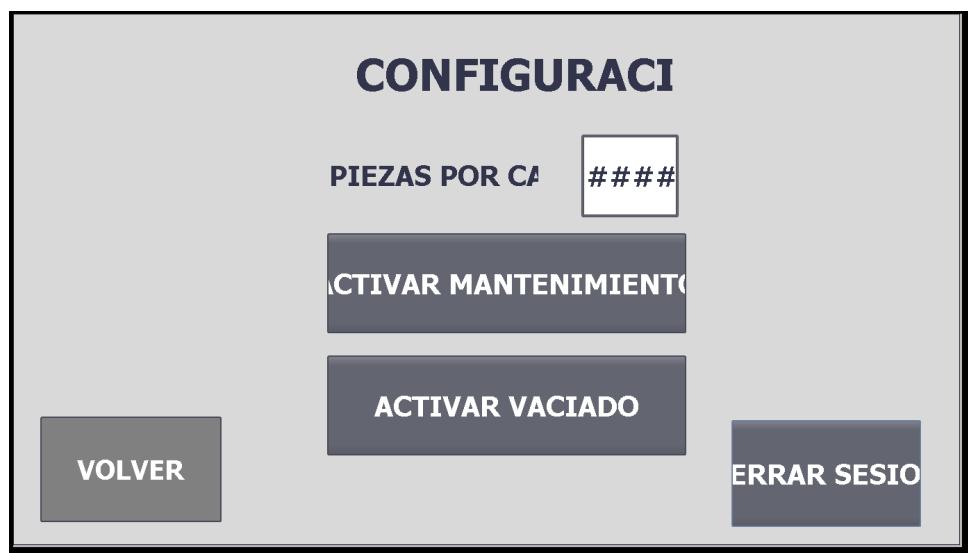


Figura 37: Pantalla de configuración

#### 4.5 Usuarios

Respecto a los usuarios, el HMI dispone de dos grupos:

- Grupo administrador -> Acceso al modo manual, y al modo de configuración de la máquina
- Grupo Usuario -> Solo tiene acceso a ver el estado de la máquina, ponerla en marcha, pararla, reanudarla ,o acceder a los valores de los contadores de piezas

#### 4.6 Registro de alarmas



Figura 38: Registro de alarmas

## 5 Resultados

Aquí adjunto el enlace al vídeo donde se puede ver el funcionamiento del sistema implementado en el PLC y el HMI funcionando en el entorno Factory IO, se enseñan las siguientes funcionalidades:

- Funcionamiento normal del sistema.
- Modo manual.
- Modo mantenimiento.
- Gestión de alarmas.
- Gestión de emergencia.
- Vaciado del sistema.

### [ENLACE VÍDEO](#)

## 6 Anexos

### 6.1 Anexo 1: Grafset\_0\_Principal.scl

*Listing 1: Código del grafset principal*

```
FUNCTION_BLOCK "Grafset_0_Principal"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Paro : Bool;
    HMI_Paro : Bool;
    Marcha : Bool;
    HMI_Marcha : Bool;
    Mantenimiento_Fin : Bool;
    Modo_Manual : Bool;
    Pieza_Empaquetada : Bool;
    CI : Bool;
    VacunaABuena : Bool;
    VacunaAMala : Bool;
    VacunaB : Bool;
    SoporteABueno : Bool;
    SoporteAMalo : Bool;
    SoporteB : Bool;
    EmbalajeABueno : Bool;
    EmbalajeAMalo : Bool;
    EmbalajeB : Bool;
END_VAR

VAR_OUTPUT
    Act_Mantenimiento : Bool;
    Proceso_Activo : Bool;
    Modo_Manual_Activado : Bool;
    contador : Int;
    C_VACAB : Int;
    C_VACAM : Int;
    C_VACB : Int;
    C_SOPAB : Int;
    C_SOPAM : Int;
    C_SOPB : Int;
    C_EMBAB : Int;
    C_EMBAM : Int;
```

```

C_EMBB : Int;
END_VAR

VAR
  S0 : Bool;
  S1 : Bool;
  S2 : Bool;
  X0 : Bool;
  X1 : Bool;
  X2 : Bool;
  X3 : Bool;
  S4 : Bool;
  X4 : Bool;
  STOP_NEG : Bool;
  S3 : Bool;
  Paro_Ciclo : Bool;
  S0p : Bool;
  S10e : Bool;
  S11e : Bool;
  S12e : Bool;
  X10e : Bool;
  S1p : Bool;
  X11e : Bool;
  X0p : Bool;
  FORCE_CURRENT_E : Bool;
  FORCE_INIT_E : Bool;
  X12e : Bool;
  X1p : Bool;
  TrigX3 {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigVACAB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigVACAM {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigVACB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigSOPAB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigSOPAM {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigSOPB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigEMBAB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigEMBAM {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
  TrigEMBB {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
END_VAR
VAR RETAIN
  Contador_G {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_VACAB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_VACAM {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_VACB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_SOPAB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_SOPAM {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_SOPB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_EMBAB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_EMBAM {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
  Contador_EMBB {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
END_VAR

BEGIN
  // EMERGENCIA
  #S10e := (#X12e AND #CI) OR "FirstScan";
  #S11e := #X10e AND "Emergencia";
  #S12e := #X11e;

  #X10e := #S10e OR (#X10e AND NOT #S11e);
  #X11e := #S11e OR (#X11e AND NOT #S12e);
  #X12e := #S12e OR (#X12e AND NOT #S10e);

  #FORCE_INIT_E := #X11e;
  #FORCE_CURRENT_E := #X12e;

  #STOP_NEG := NOT #Paro;

  #S0p := ("FirstScan" OR #X0) AND NOT #FORCE_CURRENT_E;
  #S1p := #X0 AND (#STOP_NEG OR #HMI_Paro) AND NOT #FORCE_CURRENT_E;

  #X0p := (#S0p OR #X0p AND NOT #S1p) AND NOT #FORCE_INIT_E;
  #X1p := (#S1p OR #X1p AND NOT #S0p) AND NOT #FORCE_INIT_E;

```

```

#Paro_Ciclo := #X1p;

//Grafset principal
#S0 := ("FirstScan" OR (#X3 AND #Paro_Ciclo) OR (#X4 AND NOT #Modo_Manual)) AND NOT #FORCE_CURRENT_E;
#S1 := #X0 AND ((#Marcha OR #HMI_Marcha) AND NOT #Modo_Manual) AND NOT #FORCE_CURRENT_E;
#S2 := (#X1 AND #Mantenimiento_Fin) OR (#X3 AND NOT #Paro_Ciclo) AND NOT #FORCE_CURRENT_E;
#S3 := (#X2 AND #Pieza_Empaquetada) AND NOT #FORCE_CURRENT_E;
#S4 := (#X0 AND #Modo_Manual) AND NOT #FORCE_CURRENT_E;

#X0 := (#S0 OR #X0 AND (NOT #S1 AND NOT #S4)) OR #FORCE_INIT_E;
#X1 := (#S1 OR #X1 AND (NOT #S2)) AND NOT #FORCE_INIT_E;
#X2 := (#S2 OR #X2 AND (NOT #S3)) AND NOT #FORCE_INIT_E;
#X3 := (#S3 OR #X3 AND (NOT #S0 AND NOT #S2)) AND NOT #FORCE_INIT_E;
#X4 := (#S4 OR #X4 AND (NOT #S0)) AND NOT #FORCE_INIT_E;

#Act_Mantenimiento := #X1;
#Proceso_Activo := #X2 OR #X3;
#Modo_Manual_Activado := #X4;

#TrigX3(CLK:=#X3);
#TrigVACAB(CLK := (#X3 OR #X2) AND #VacunaABuena);
#TrigVACAM(CLK := (#X3 OR #X2) AND #VacunaAMala);
#TrigVACB(CLK := (#X3 OR #X2) AND #VacunaB);
#TrigSOPAB(CLK := (#X3 OR #X2) AND #SoporteABueno);
#TrigSOPAM(CLK := (#X3 OR #X2) AND #SoporteAMalo);
#TrigSOPB(CLK := (#X3 OR #X2) AND #SoporteB);
#TrigEMBAB(CLK := (#X3 OR #X2) AND #EmbalajeABueno);
#TrigEMBAM(CLK := (#X3 OR #X2) AND #EmbalajeAMalo);
#TrigEMBB(CLK := (#X3 OR #X2) AND #EmbalajeB);

#Contador_G(CU:=#TrigX3.Q,
             R:=#X0,
             PV:=50,
             CV=>#contador);

#Contador_VACAB(CU := #TrigVACAB.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_VACAB);

#Contador_VACAM(CU := #TrigVACAM.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_VACAM);

#Contador_VACB(CU := #TrigVACB.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_VACB);

#Contador_SOPAB(CU := #TrigSOPAB.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_SOPAB);

#Contador_SOPAM(CU := #TrigSOPAM.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_SOPAM);

#Contador_SOPB(CU := #TrigSOPB.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_SOPB);

#Contador_EMBAB(CU := #TrigEMBAB.Q,
                 R := #X0,
                 PV := 50,
                 CV => #C_EMBAB);

```

```

#Contador_EMBAM(CU := #TrigEMBAM.Q,
                  R := #X0,
                  PV := 50,
                  CV => #C_EMBAM);
#Contador_EMBB(CU := #TrigEMBB.Q,
                  R := #X0,
                  PV := 50,
                  CV => #C_EMBB);

END_FUNCTION_BLOCK

```

## 6.2 Anexo 2: Grafset\_10.scl

*Listing 2: Código de la máquina 1: vacuna y soporte*

```

FUNCTION_BLOCK "Grafset_10_Final"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Act_Maquina : Bool;
  S_Pres : Bool;
  S1 : Bool;
  Sig_Maq_Libre : Bool;
  Mantenimiento_Cintas : Bool;
  Vaciado_cinta : Bool;
  Act_man_qc1 : Bool;
  Pieza_Rec : Bool;
  Reset : Bool;
  Resume : Bool;
  CI : Bool;
  Emergencia : Bool;
END_VAR

VAR_OUTPUT
  Emisor : Bool;
  Emisor_2 : Bool;
  Maq_Libre : Bool;
  QC1 : Bool;
  Act_SIG_MAQ : Bool;
  Alarma_Maquina : Bool;
  EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
  Alarma_1 : Bool;
  Alarma_2 : Bool;
END_VAR

VAR
  S10 : Bool;
  S11 : Bool;
  S12 : Bool;
  S13 : Bool;
  S14 : Bool;
  S15 : Bool;
  X10 : Bool;
  X15 : Bool;
  X11 : Bool;
  X12 : Bool;
  X13 : Bool;
  X14 : Bool;
  S10e : Bool;
  S11e : Bool;
  S12e : Bool;
  X10e : Bool;
  X11e : Bool;
  X12e : Bool;
  FORCE_INIT : Bool;

```

```

FORCE_INIT_E : Bool;
FORCE_CURRENT_E : Bool;
S10a : Bool;
S11a : Bool;
S12a : Bool;
S13a : Bool;
X10a : Bool;
X11a : Bool;
X12a : Bool;
X13a : Bool;
FORCE_CURRENT : Bool;
STOP_ACT : Bool;
RESET_ALARMA : Bool;
Alarma : Bool;
S11b : Bool;
X11b : Bool;
T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
trig_PR {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
//Flancos

// EMERGENCIA
#S10e := (#X12e AND #CI) OR "FirstScan";
#S11e := #X10e AND "Emergencia";
#S12e := #X11e;

#X10e := #S10e OR (#X10e AND NOT #S11e);
#X11e := #S11e OR (#X11e AND NOT #S12e);
#X12e := #S12e OR (#X12e AND NOT #S10e);

#FORCE_INIT_E := #X11e;
#FORCE_CURRENT_E := #X12e;

// ALARMAS
#S10a := (#X13a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S11a := (#X10a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S12a := (#X11a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S13a := (#X12a OR (#X11a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X10a := #S10a OR (#X10a AND NOT #S11a) OR #FORCE_INIT_E;
#X11a := (#S11a OR (#X11a AND NOT #S12a AND NOT #S13a)) AND NOT #FORCE_INIT_E;
#X12a := (#S12a OR (#X12a AND NOT #S13a)) AND NOT #FORCE_INIT_E;
#X13a := (#S13a OR (#X13a AND NOT #S10a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X11a;
#FORCE_INIT := #FORCE_INIT_E OR #X12a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X11a;
#RESET_ALARMA := #X13a;

#trig_PR(CLK:=#Pieza_Rec);
#T1(IN := #X15,
    PT := T#2s);

// PRODUCCIÓN
#S10 := "FirstScan" OR( (#X15 AND #T1.Q)) AND NOT #FORCE_CURRENT;
#S11 := (#X10 AND #Act_Maquina) AND NOT #FORCE_CURRENT;
#S11b := (#X11 AND #S_Pres) AND NOT #FORCE_CURRENT;
#S12 := (#X11b AND (NOT #S_Pres)) AND NOT #FORCE_CURRENT;
#S13 := (#X12 AND #S1) AND NOT #FORCE_CURRENT;
#S14 := (#X13 AND #Sig_Maq_Libre) AND NOT #FORCE_CURRENT;
#S15 := (#X14 AND NOT #S1) AND NOT #FORCE_CURRENT;

#X10 := (#S10 OR #X10 AND (NOT #S11)) OR #FORCE_INIT;
#X11 := (#S11 OR #X11 AND (NOT #S11b)) AND NOT #FORCE_INIT;
#X11b := (#S11b OR #X11b AND (NOT #S12)) AND NOT #FORCE_INIT;
#X12 := (#S12 OR #X12 AND (NOT #S13 AND NOT #S14)) AND NOT #FORCE_INIT;
#X13 := (#S13 OR #X13 AND (NOT #S14)) AND NOT #FORCE_INIT;

```

```

#X14 := (#S14 OR #X14 AND (NOT #S15)) AND NOT #FORCE_INIT;
#X15 := (#S15 OR #X15 AND (NOT #S10)) AND NOT #FORCE_INIT;

#Maq_Libre := #X10 AND NOT #STOP_ACT;
#Emisor_2 := (#X11 AND NOT #S_Pres) AND NOT #STOP_ACT;
#Emisor := (#X11 AND NOT #S_Pres) AND NOT #STOP_ACT;
#QC1 := (#X11 OR #X11b OR #X12 OR #X14 OR #Mantenimiento_Cintas OR #Vaciado_cinta OR #Act_man_qci
|AND NOT #STOP_ACT;
#Act_SIG_MAQ := #X14 AND NOT #STOP_ACT;

#T1a(IN:=#X12) AND NOT #RESET_ALARMA ,
PT:=T#5s);
#Alarma_1 := (#Alarma_1 OR (#X12 AND #S_Pres)) AND NOT #RESET_ALARMA;
#Alarma_2 := (#Alarma_2 OR (#T1a.Q)) AND NOT #RESET_ALARMA;
#Alarma := (#Alarma_1 OR #Alarma_2) AND NOT #RESET_ALARMA;
#Alarma_Maquina := #Alarma;

IF #X10 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X10 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

### 6.3 Anexo 3: Grafset\_10\_1.scl

*Listing 3: Código de la máquina 1: embalaje*

```

FUNCTION_BLOCK "Grafset_10_Final_1"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Maquina : Bool;
    S_Pres : Bool;
    S1 : Bool;
    Sig_Maq_Libre : Bool;
    Mantenimiento_Cintas : Bool;
    vaciado_cinta : Bool;
    man_cintaA : Bool;
    man_cintaB : Bool;
    Pieza_Rec : Bool;
    Reset : Bool;
    Resume : Bool;
    CI : Bool;
    Emergencia : Bool;
END_VAR

VAR_OUTPUT
    Emisor : Bool;
    Emisor_2 : Bool;
    Maq_Libre : Bool;
    QC1A : Bool;
    QC1B : Bool;
    Act_SIG_MAQ : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

```

```

VAR_IN_OUT
    Alarma_1 : Bool;
    Alarma_2 : Bool;
END_VAR

VAR
    S10 : Bool;
    S11 : Bool;
    S12 : Bool;
    S13 : Bool;
    S14 : Bool;
    S15 : Bool;
    X10 : Bool;
    X15 : Bool;
    X11 : Bool;
    X12 : Bool;
    X13 : Bool;
    X14 : Bool;
    S10e : Bool;
    S11e : Bool;
    S12e : Bool;
    X10e : Bool;
    X11e : Bool;
    X12e : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    FORCE_CURRENT_E : Bool;
    S10a : Bool;
    S11a : Bool;
    S12a : Bool;
    S13a : Bool;
    X10a : Bool;
    X11a : Bool;
    X12a : Bool;
    X13a : Bool;
    FORCE_CURRENT : Bool;
    STOP_ACT : Bool;
    RESET_ALARMA : Bool;
    Alarma : Bool;
    S11b : Bool;
    X11b : Bool;
    T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    trig_PR {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
END_VAR

BEGIN
    //Flancos

    // EMERGENCIA
    #S10e := (#X12e AND #CI) OR "FirstScan";
    #S11e := #X10e AND "Emergencia";
    #S12e := #X11e;

    #X10e := #S10e OR (#X10e AND NOT #S11e);
    #X11e := #S11e OR (#X11e AND NOT #S12e);
    #X12e := #S12e OR (#X12e AND NOT #S10e);

    #FORCE_INIT_E := #X11e;
    #FORCE_CURRENT_E := #X12e;

    // ALARMAS
    #S10a := (#X13a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
    #S11a := (#X10a AND #Alarma) AND NOT #FORCE_CURRENT_E;
    #S12a := (#X11a AND #Reset) AND NOT #FORCE_CURRENT_E;
    #S13a := (#X12a OR (#X11a AND #Resume)) AND NOT #FORCE_CURRENT_E;

    #X10a := #S10a OR (#X10a AND NOT #S11a) OR #FORCE_INIT_E;
    #X11a := (#S11a OR (#X11a AND NOT #S12a AND NOT #S13a)) AND NOT #FORCE_INIT_E;
    #X12a := (#S12a OR (#X12a AND NOT #S13a)) AND NOT #FORCE_INIT_E;
    #X13a := (#S13a OR (#X13a AND NOT #S10a)) AND NOT #FORCE_INIT_E;

```

```

#STOP_ACT := #X11a;
#FORCE_INIT := #FORCE_INIT_E OR #X12a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X11a;
#RESET_ALARMA := #X13a;

#trig_PR(CLK:=#Pieza_Rec);

// PRODUCCIÓN
#S10 := ("FirstScan" OR (#X15 AND #trig_PR.Q)) AND NOT #FORCE_CURRENT;
#S11 := (#X10 AND #Act_Maquina) AND NOT #FORCE_CURRENT;
#S11b := (#X11 AND (#S_Pres)) AND NOT #FORCE_CURRENT;
#S12 := (#X11b AND (NOT #S_Pres)) AND NOT #FORCE_CURRENT;
#S13 := (#X12 AND #S1) AND NOT #FORCE_CURRENT;
#S14 := (#X13 AND #Sig_Maq_Libre) AND NOT #FORCE_CURRENT;
#S15 := (#X14 AND NOT #S1) AND NOT #FORCE_CURRENT;

#X10 := (#S10 OR #X10 AND (NOT #S11)) OR #FORCE_INIT;
#X11 := (#S11 OR #X11 AND (NOT #S11b)) AND NOT #FORCE_INIT;
#X11b := (#S11b OR #X11b AND (NOT #S12)) AND NOT #FORCE_INIT;
#X12 := (#S12 OR #X12 AND (NOT #S13 AND NOT #S14)) AND NOT #FORCE_INIT;
#X13 := (#S13 OR #X13 AND (NOT #S14)) AND NOT #FORCE_INIT;
#X14 := (#S14 OR #X14 AND (NOT #S15)) AND NOT #FORCE_INIT;
#X15 := (#S15 OR #X15 AND (NOT #S10)) AND NOT #FORCE_INIT;

#Maq_Libre := #X10 AND NOT #STOP_ACT;
#Emisor_2 := ("Proceso_Activo") AND NOT #STOP_ACT;
#Emisor := ("Proceso_Activo") AND NOT #STOP_ACT;
//#Emisor_2 := (#X11 AND NOT #S_Pres) AND NOT #STOP_ACT;
//#Emisor := (#X11 AND NOT #S_Pres) AND NOT #STOP_ACT;
#QC1A := (#X11 OR #X11b OR #Mantenimiento_Cintas OR #vaciado_cinta OR #man_cintaA)
AND NOT #STOP_ACT;
#QC1B := (#X11b OR #X12 OR #X14 OR #Mantenimiento_Cintas OR #vaciado_cinta OR #man_cintaB) AND NOT #STOP_ACT;
#Act_SIG_MAQ := #X14 AND NOT #STOP_ACT;

IF #X10 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X10 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

#T1a(IN:=#X14 AND NOT #RESET_ALARMA,
PT:=T#5s);

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;
#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Maquina := #Alarma;

END_FUNCTION_BLOCK

```

#### 6.4 Anexo 4: Grafset\_20.scl

*Listing 4: Código de la máquina 2: vacuna y soporte*

```
FUNCTION_BLOCK "Grafset_20_FInal"
```

```

{ S7_Optimized_Access := 'TRUE' }

VERSION : 0.1
VAR_INPUT
  Act_Proceso : Bool;
  SigMaqLibre : Bool;
  SA_Bueno : Bool;
  SA_Malo : Bool;
  SB : Bool;
  Mantenimiento_Cintas : Bool;
  Mantenimiento_Transfes : Bool;
  vaciado_cinta : Bool;
  vaciado_Trans : Bool;
  Man_Cinta : Bool;
  Man_Trans : Bool;
  CI : Bool;
  Reset : Bool;
  Resume : Bool;
END_VAR

VAR_OUTPUT
  Maq_Libre : Bool;
  QC2 : Bool;
  QTRANS : Bool;
  Act_SIG_MAQ_Buena : Bool;
  Act_SIG_MAQ_Mala : Bool;
  TipoAbuena : Bool;
  TipoAMala : Bool;
  TipoB : Bool;
  Alarma_Maquina : Bool;
  Pieza_Rec : Bool;
  EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
  Alarma_1 : Bool;
END_VAR

VAR
  S20 : Bool;
  S21 : Bool;
  S22 : Bool;
  S23 : Bool;
  S24 : Bool;
  S22b : Bool;
  S23b : Bool;
  X20 : Bool;
  X21 : Bool;
  X22 : Bool;
  X22b : Bool;
  X23 : Bool;
  X23b : Bool;
  X24 : Bool;
  S20a : Bool;
  S21a : Bool;
  S22a : Bool;
  S23a : Bool;
  X20a : Bool;
  X21a : Bool;
  X22a : Bool;
  X23a : Bool;
  S20e : Bool;
  S21e : Bool;
  S22e : Bool;
  X20e : Bool;
  X21e : Bool;
  X22e : Bool;
  FORCE_INIT : Bool;
  FORCE_INIT_E : Bool;
  FORCE_CURRENT : Bool;
  FORCE_CURRENT_E : Bool;
  STOP_ACT : Bool;
  RESETALARMA : Bool;
  Alarma : Bool;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;

```

```

T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN

// EMERGENCIA
#S20e := (#X22e AND #CI) OR "FirstScan";
#S21e := #X20e AND "Emergencia";
#S22e := #X21e;

#X20e := #S20e OR (#X20e AND NOT #S21e);
#X21e := #S21e OR (#X21e AND NOT #S22e);
#X22e := #S22e OR (#X22e AND NOT #S20e);

#FORCE_INIT_E := #X21e;
#FORCE_CURRENT_E := #X22e;

// ALARMAS
#S20a := (#X23a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S21a := (#X20a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S22a := (#X21a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S23a := (#X22a OR (#X21a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X20a := #S20a OR (#X20a AND NOT #S21a) OR #FORCE_INIT_E;
#X21a := (#S21a OR (#X21a AND NOT #S22a AND NOT #S23a)) AND NOT #FORCE_INIT_E;
#X22a := (#S22a OR (#X22a AND NOT #S23a)) AND NOT #FORCE_INIT_E;
#X23a := (#S23a OR (#X23a AND NOT #S20a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X21a;
#FORCE_INIT := #FORCE_INIT_E OR #X22a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X21a;
#RESET_ALARMA := #X23a;
#T1(IN:=#X24,
    PT:=T#2s);

// PRODUCCIÓN
#S20 := ("FirstScan" OR (#X22b AND NOT #SA_Bueno) OR (#X23b AND NOT #SA_Malo) OR (#X24 AND #T1.Q)
#S21 := (#X20 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S22 := (#X21 AND #SA_Bueno) AND NOT #FORCE_CURRENT;
#S22b := (#X22 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S23 := (#X21 AND #SA_Malo AND NOT #SA_Bueno) AND NOT #FORCE_CURRENT;
#S23b := (#X23 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S24 := (#X21 AND #SB) AND NOT #FORCE_CURRENT;

#X20 := (#S20 OR #X20 AND (NOT #S21)) OR #FORCE_INIT;
#X21 := (#S21 OR #X21 AND (NOT #S22 AND NOT #S23 AND NOT #S24)) AND NOT #FORCE_INIT;
#X22 := (#S22 OR #X22 AND (NOT #S22b)) AND NOT #FORCE_INIT;
#X22b := (#S22b OR #X22b AND (NOT #S20)) AND NOT #FORCE_INIT;
#X23 := (#S23 OR #X23 AND (NOT #S23b)) AND NOT #FORCE_INIT;
#X23b := (#S23b OR #X23b AND (NOT #S20)) AND NOT #FORCE_INIT;
#X24 := (#S24 OR #X24 AND (NOT #S20)) AND NOT #FORCE_INIT;

#Maq_Libre := #X20 AND NOT #STOP_ACT;
#QC2 := (#X21 OR #X22b OR #X23b OR #Mantenimiento_Cintas OR #vaciado_cinta OR #Man_Cinta) AND NOT
#QTRANS := (#X24 OR #Mantenimiento_Transfes OR #vaciado_Trans OR #Man_Trans) AND NOT #STOP_ACT;
#Act_SIG_MAQ_Buena := #X22b AND NOT #STOP_ACT;
#Act_SIG_MAQ_Mala := #X23b AND NOT #STOP_ACT;
#Pieza_Rec := (#X22 OR #X23 OR #X24) AND NOT #STOP_ACT;

#TipoAbuena := #X22b;
#TipoAMala := #X23b;
#Tipob := #X24;

#T1a(IN := #X21 AND NOT #RESET_ALARMA,

```

```

PT := T#5s;

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;

#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Maquina := #Alarma;

IF #X20 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X20 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.5 Anexo 5: Grafset\_20\_1.scl

*Listing 5: Código de la máquina 2: embalaje*

```

FUNCTION_BLOCK "Grafset_20_2_FInal"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    SigMaqLibre : Bool;
    SA_Bueno : Bool;
    SA_Malo : Bool;
    SB : Bool;
    Mantenimiento_Cintas : Bool;
    Mantenimiento_Transfers : Bool;
    vaciado_cinta : Bool;
    vaciado_trans : Bool;
    Man_cinta : Bool;
    Man_trans : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC2 : Bool;
    QTRANS : Bool;
    Act_SIG_MAQ_Buena : Bool;
    Act_SIG_MAQ_Mala : Bool;
    Pieza_Rec : Bool;
    EstadoMaquina : Int;
    Alarma_Maquina : Bool;
    TipoABuena : Bool;
    TipoAMala : Bool;
    TipoB : Bool;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;

```

```

END_VAR

VAR
  S20 : Bool;
  S21 : Bool;
  S22 : Bool;
  S23 : Bool;
  S24 : Bool;
  S22b : Bool;
  S23b : Bool;
  X20 : Bool;
  X21 : Bool;
  X22 : Bool;
  X22b : Bool;
  X23 : Bool;
  X23b : Bool;
  X24 : Bool;
  S20a : Bool;
  S21a : Bool;
  S22a : Bool;
  S23a : Bool;
  X20a : Bool;
  X21a : Bool;
  X22a : Bool;
  X23a : Bool;
  S20e : Bool;
  S21e : Bool;
  S22e : Bool;
  X20e : Bool;
  X21e : Bool;
  X22e : Bool;
  FORCE_INIT : Bool;
  FORCE_INIT_E : Bool;
  FORCE_CURRENT : Bool;
  FORCE_CURRENT_E : Bool;
  STOP_ACT : Bool;
  RESET_ALARMA : Bool;
  Alarma : Bool;
  T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
  T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN

// EMERGENCIA
#S20e := (#X22e AND #CI) OR "FirstScan";
#S21e := #X20e AND "Emergencia";
#S22e := #X21e;

#X20e := #S20e OR (#X20e AND NOT #S21e);
#X21e := #S21e OR (#X21e AND NOT #S22e);
#X22e := #S22e OR (#X22e AND NOT #S20e);

#FORCE_INIT_E := #X21e;
#FORCE_CURRENT_E := #X22e;

// ALARMAS
#S20a := (#X23a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S21a := (#X20a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S22a := (#X21a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S23a := (#X22a OR (#X21a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X20a := #S20a OR (#X20a AND NOT #S21a) OR #FORCE_INIT_E;
#X21a := (#S21a OR (#X21a AND NOT #S22a AND NOT #S23a)) AND NOT #FORCE_INIT_E;
#X22a := (#S22a OR (#X22a AND NOT #S23a)) AND NOT #FORCE_INIT_E;
#X23a := (#S23a OR (#X23a AND NOT #S20a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X21a;
#FORCE_INIT := #FORCE_INIT_E OR #X22a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X21a;
#RESET_ALARMA := #X23a;

```

```

#T1( IN :=#X24 ,
    PT := T#2s );

// PRODUCCIÓN
#S20 := ("FirstScan" OR (#X22b AND NOT #SA_Bueno AND NOT #SA_Malo AND NOT #SB) OR (#X23b AND NOT #SB) OR (#X24b AND NOT #SB);
#S21 := (#X20 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S22 := (#X21 AND #SA_Bueno AND #SA_Malo AND #SB) AND NOT #FORCE_CURRENT;
#S22b := (#X22 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S23 := (#X21 AND #SA_Malo AND NOT #SA_Bueno AND #SB) AND NOT #FORCE_CURRENT;
#S23b := (#X23 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S24 := (#X21 AND #SB AND #SA_Bueno AND NOT #SA_Malo) AND NOT #FORCE_CURRENT;

#X20 := (#S20 OR #X20 AND (NOT #S21)) OR #FORCE_INIT;
#X21 := (#S21 OR #X21 AND (NOT #S22 AND NOT #S23 AND NOT #S24)) AND NOT #FORCE_INIT;
#X22 := (#S22 OR #X22 AND (NOT #S22b)) AND NOT #FORCE_INIT;
#X22b := (#S22b OR #X22b AND (NOT #S20)) AND NOT #FORCE_INIT;
#X23 := (#S23 OR #X23 AND (NOT #S23b)) AND NOT #FORCE_INIT;
#X23b := (#S23b OR #X23b AND (NOT #S20)) AND NOT #FORCE_INIT;
#X24 := (#S24 OR #X24 AND (NOT #S20)) AND NOT #FORCE_INIT;

#Maq_Libre := #X20 AND NOT #STOP_ACT;
#QC2 := (#X21 OR #X22b OR #X23b OR #Mantenimiento_Cintas OR #vaciado_cinta OR #Man_cinta) AND NOT #STOP_ACT;
#QTRANS := (#X24 OR #Mantenimiento_Transfers OR #vaciado_trans OR #Man_trans) AND NOT #STOP_ACT;
#Act_SIG_MAQ_Buena := #X22b AND NOT #STOP_ACT;
#Act_SIG_MAQ_Mala := #X23b AND NOT #STOP_ACT;
#Pieza_Rec := (#X22 OR #X23 OR #X24) AND NOT #STOP_ACT;

#TipoABuena := #X22b;
#TipoAMala := #X23b;
#Tipob := #X24;

#T1a(IN := #X21 AND NOT #RESET_ALARMA ,
     PT := T#5s);

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;

#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Mquina := #Alarma;

IF #X20 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X20 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.6 Anexo 6: Grafset\_30.scl

*Listing 6: Código de la máquina 3*

```

FUNCTION_BLOCK "grafcet_30_fina"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso_Bueno : Bool;
    Act_Proceso_malo : Bool;
    S3 : Bool;
    SigMaqLibre : Bool;
    Mantenimiento_Cinta : Bool;
    vaciado_cinta : Bool;
    man_cinta : Bool;
    Reset : Bool;
    Resume : Bool;
    CI : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC3 : Bool;
    Act_SIG_MAQ_Buena : Bool;
    Act_SIG_MAQ_Mala : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
END_VAR

VAR
    S30 : Bool;
    S31 : Bool;
    S31b : Bool;
    S31c : Bool;
    S32 : Bool;
    S32b : Bool;
    S32c : Bool;
    X30 : Bool;
    X31 : Bool;
    X32 : Bool;
    X32b : Bool;
    X32c : Bool;
    X31b : Bool;
    X31c : Bool;
    S30e : Bool;
    S31e : Bool;
    S32e : Bool;
    X30e : Bool;
    X31e : Bool;
    X32e : Bool;
    X30a : Bool;
    X31a : Bool;
    X32a : Bool;
    X33a : Bool;
    S30a : Bool;
    S31a : Bool;
    S32a : Bool;
    S33a : Bool;
    FORCE_INIT_E : Bool;
    FORCE_INIT : Bool;
    FORCE_CURRENT_e : Bool;
    FORCE_CURRENT : Bool;
    RESET_ALARMA : Bool;
    STOP_ACT : Bool;
    Alarma : Bool;
    T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T2a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
    //Flancos

```

```

// EMERGENCIA
#S30e := (#X32e AND #CI) OR "FirstScan";
#S31e := #X30e AND "Emergencia";
#S32e := #X31e;

#X30e := #S30e OR (#X30e AND NOT #S31e);
#X31e := #S31e OR (#X31e AND NOT #S32e);
#X32e := #S32e OR (#X32e AND NOT #S30e);

#FORCE_INIT_E := #X31e;
#FORCE_CURRENT_e := #X32e;

// ALARMAS
#S30a := (#X33a AND NOT #FORCE_CURRENT_e) OR "FirstScan";
#S31a := (#X30a AND #Alarma) AND NOT #FORCE_CURRENT_e;
#S32a := (#X31a AND #Reset) AND NOT #FORCE_CURRENT_e;
#S33a := (#X32a OR (#X31a AND #Resume)) AND NOT #FORCE_CURRENT_e;

#X30a := #S30a OR (#X30a AND NOT #S31a) OR #FORCE_INIT_E;
#X31a := (#S31a OR (#X31a AND NOT #S32a AND NOT #S33a)) AND NOT #FORCE_INIT_E;
#X32a := (#S32a OR (#X32a AND NOT #S33a)) AND NOT #FORCE_INIT_E;
#X33a := (#S33a OR (#X33a AND NOT #S30a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X31a;
#FORCE_INIT := #FORCE_INIT_E OR #X32a;
#FORCE_CURRENT := #FORCE_CURRENT_e OR #X31a;
#RESET_ALARMA := #X33a;

// PRODUCCIÓN
#S30 := ("FirstScan" OR (#X31c AND NOT #S3) OR (#X32c AND NOT #S3)) AND NOT #FORCE_CURRENT;
#S31 := (#X30 AND #Act_Proceso_Bueno) AND NOT #FORCE_CURRENT;
#S31b := (#X31 AND #S3) AND NOT #FORCE_CURRENT;
#S31c := (#X31b AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S32 := (#X30 AND #Act_Proceso_malo) AND NOT #FORCE_CURRENT;
#S32b := (#X32 AND #S3) AND NOT #FORCE_CURRENT;
#S32c := (#X32b AND #SigMaqLibre) AND NOT #FORCE_CURRENT;

#X30 := (#S30 OR #X30 AND (NOT #S31 AND NOT #S32)) OR #FORCE_INIT;
#X31 := (#S31 OR #X31 AND (NOT #S31b)) AND NOT #FORCE_INIT;
#X31b := (#S31b OR #X31b AND (NOT #S31c)) AND NOT #FORCE_INIT;
#X31c := (#S31c OR #X31c AND (NOT #S30)) AND NOT #FORCE_INIT;
#X32 := (#S32 OR #X32 AND (NOT #S32b)) AND NOT #FORCE_INIT;
#X32b := (#S32b OR #X32b AND (NOT #S32c)) AND NOT #FORCE_INIT;
#X32c := (#S32c OR #X32c AND (NOT #S30)) AND NOT #FORCE_INIT;

#Maq_Libre := #X30 AND NOT #STOP_ACT;
#QC3 := (#X31 OR #X31c OR #X32 OR #X32c OR #Mantenimiento_Cinta OR #vaciado_cinta OR #man_cinta) AND NOT #STOP_ACT;
#Act_SIG_MAQ_Buena := #X31c AND NOT #STOP_ACT;
#Act_SIG_MAQ_Mala := #X32c AND NOT #STOP_ACT;

#T1a(IN:=#X31 OR #X32) AND NOT #RESET_ALARMA,
PT:=T#6s);

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;

#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Mquina := #Alarma;
//  

IF #X30 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X30 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;

```

```

IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.7 Anexo 7: Grafset\_40.scl

*Listing 7: Código de la máquina 4*

```

FUNCTION_BLOCK "grafset_40_final"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso_Bueno : Bool;
    Act_Proceso_Malo : Bool;
    S4 : Bool;
    SigMaqLibre : Bool;
    Mantenimiento_Cinta : Bool;
    Mantenimiento_Trans : Bool;
    vaciado_cinta : Bool;
    vaciado_Trans : Bool;
    man_cinta : Bool;
    man_trans : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC4 : Bool;
    QTRANS : Bool;
    Act_Sig_Maq : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
END_VAR

VAR
    S40 : Bool;
    S41 : Bool;
    S42 : Bool;
    S41b : Bool;
    S41c : Bool;
    S42b : Bool;
    X40 : Bool;
    X41 : Bool;
    X41b : Bool;
    X41c : Bool;
    X42 : Bool;
    X42b : Bool;
    S40e : Bool;
    S41e : Bool;
    S42e : Bool;
    X40e : Bool;
    X41e : Bool;
    X42e : Bool;
    S40a : Bool;
    S41a : Bool;
    S42a : Bool;
    S43a : Bool;
    X40a : Bool;

```

```

X41a : Bool;
X42a : Bool;
X43a : Bool;
FORCE_CURRENT : Bool;
FORCE_CURRENT_E : Bool;
FORCE_INIT : Bool;
FORCE_INIT_E : Bool;
STOP_ACT : Bool;
RESET_ALARMA : Bool;
Alarma : Bool;
T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T3a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN

// EMERGENCIA
#S40e := (#X42e AND #CI) OR "FirstScan";
#S41e := #X40e AND "Emergencia";
#S42e := #X41e;

#X40e := #S40e OR (#X40e AND NOT #S41e);
#X41e := #S41e OR (#X41e AND NOT #S42e);
#X42e := #S42e OR (#X42e AND NOT #S40e);

#FORCE_INIT_E := #X41e;
#FORCE_CURRENT_E := #X42e;

// ALARMAS
#S40a := (#X43a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S41a := (#X40a AND #STOP_ACT) AND NOT #FORCE_CURRENT_E;
#S42a := (#X41a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S43a := (#X42a OR (#X41a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X40a := #S40a OR (#X40a AND NOT #S41a) OR #FORCE_INIT_E;
#X41a := (#S41a OR (#X41a AND NOT #S42a AND NOT #S43a)) AND NOT #FORCE_INIT_E;
#X42a := (#S42a OR (#X42a AND NOT #S43a)) AND NOT #FORCE_INIT_E;
#X43a := (#S43a OR (#X43a AND NOT #S40a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X41a;
#FORCE_INIT := #FORCE_INIT_E OR #X42a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X41a;
#RESET_ALARMA := #X43a;

// PRODUCCIÓN
#S40 := ("FirstScan" OR (#X41c AND NOT #S4) OR (#X42b AND NOT #S4)) AND NOT #FORCE_CURRENT;
#S41 := (#X40 AND #Act_Proceso_Bueno) AND NOT #FORCE_CURRENT;
#S41b := (#X41 AND #S4) AND NOT #FORCE_CURRENT;
#S41c := (#X41b AND #SigMaqLibre) AND NOT #FORCE_CURRENT;
#S42 := (#X40 AND #Act_Proceso_Malo) AND NOT #FORCE_CURRENT;
#S42b := (#X42 AND #S4) AND NOT #FORCE_CURRENT;

#X40 := (#S40 OR #X40 AND (NOT #S41 AND NOT #S42)) OR #FORCE_INIT;
#X41 := (#S41 OR #X41 AND (NOT #S41b)) AND NOT #FORCE_INIT;
#X41b := (#S41b OR #X41b AND (NOT #S41c)) AND NOT #FORCE_INIT;
#X41c := (#S41c OR #X41c AND (NOT #S40)) AND NOT #FORCE_INIT;
#X42 := (#S42 OR #X42 AND (NOT #S42b)) AND NOT #FORCE_INIT;
#X42b := (#S42b OR #X42b AND (NOT #S40)) AND NOT #FORCE_INIT;

#Maq.Libre := #X40 AND NOT #STOP_ACT;
#QC4 := (#X41 OR #X41c OR #X42 OR #Mantenimiento_Cinta OR #vaciado_cinta OR #man_cinta) AND NOT #STOP_ACT;
#Act_Sig_Maq := #X41c AND NOT #STOP_ACT;
#QTRANS := (#X42b OR #Mantenimiento_Trans OR #vaciado_Trans OR #man_trans) AND NOT #STOP_ACT;

```

```

IF #X40 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X40 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

#T1a(IN:=(#X41 OR #X42) AND NOT #RESET_ALARMA ,
PT:=T#4s);

#Alarma_1 := (#Alarma_1 OR #T1a.Q) AND NOT #RESET_ALARMA;

#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
END_FUNCTION_BLOCK

```

## 6.8 Anexo 8: Grafset\_50.scl

*Listing 8: Código de la máquina 5: vacunas y 6 soporte*

```

FUNCTION_BLOCK "grafset_50_Final"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    S5 : Bool;
    HaySoporte : Bool;
    CI : Bool;
    Mantenimiento_Cinta : Bool;
    Mantenimiento_Clamp : Bool;
    Mantenimiento_Gripper : Bool;
    Mantenimiento_BajarGripper : Bool;
    Mantenimiento_MOVHOR : Bool;
    vaciado_cinta : Bool;
    vaciado_subclamp : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC5 : Bool;
    QC6 : Bool;
    QC7 : Bool;
    QCLAMP : Bool;
    qsubclamp : Bool;
    QBAJ : Bool;
    QGRIPPER : Bool;
    QHOR : Bool;
    HayPieza : Bool;
    Pieza_Empaquetada : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
END_VAR

```

```

VAR
    S50 : Bool;
    S51 : Bool;
    S52 : Bool;
    S53 : Bool;
    S54 : Bool;
    S55 : Bool;
    S56 : Bool;
    S57 : Bool;
    S58 : Bool;
    S59 : Bool;
    S60 : Bool;
    S61 : Bool;
    S62 : Bool;
    X50 : Bool;
    X51 : Bool;
    X52 : Bool;
    X53 : Bool;
    X54 : Bool;
    X55 : Bool;
    X56 : Bool;
    X57 : Bool;
    X58 : Bool;
    X59 : Bool;
    X60 : Bool;
    X61 : Bool;
    X62 : Bool;
    S50e : Bool;
    S51e : Bool;
    S52e : Bool;
    X50e : Bool;
    X51e : Bool;
    X52e : Bool;
    S50a : Bool;
    S51a : Bool;
    S52a : Bool;
    S53a : Bool;
    X50a : Bool;
    X51a : Bool;
    X52a : Bool;
    X53a : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    FORCE_CURRENT : Bool;
    FORCE_CURRENT_E : Bool;
    STOP_ACT : Bool;
    RESET_ALARMA : Bool;
    Alarma : Bool;
    T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T2 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T3 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T4 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T5 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T6 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T7 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T8 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T9 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN

// EMERGENCIA
#S50e := (#X52e AND #CI) OR "FirstScan";
#S51e := #X50e AND "Emergencia";
#S52e := #X51e;

#X50e := #S50e OR (#X50e AND NOT #S51e);
#X51e := #S51e OR (#X51e AND NOT #S52e);
#X52e := #S52e OR (#X52e AND NOT #S50e);

```

```

#FORCE_INIT_E := #X51e;
#FORCE_CURRENT_E := #X52e;

// ALARMAS
#S50a := (#X53a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S51a := (#X50a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S52a := (#X51a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S53a := (#X52a OR (#X51a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X50a := #S50a OR (#X50a AND NOT #S51a) OR #FORCE_INIT_E;
#X51a := (#S51a OR (#X51a AND NOT #S52a AND NOT #S53a)) AND NOT #FORCE_INIT_E;
#X52a := (#S52a OR (#X52a AND NOT #S53a)) AND NOT #FORCE_INIT_E;
#X53a := (#S53a OR (#X53a AND NOT #S50a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X51a;
#FORCE_INIT := #FORCE_INIT_E OR #X52a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X51a;
#RESET_ALARMA := #X53a;

#T1( IN:=#X52 ,
    PT:=T#2s );
#T2( IN := #X53 ,
    PT := T#2s );
#T3( IN := #X55 ,
    PT := T#2s );
#T4( IN := #X56 ,
    PT := T#2s );
#T5( IN := #X57 ,
    PT := T#2s );
#T6( IN := #X58 ,
    PT := T#2s );
#T7( IN := #X59 ,
    PT := T#2s );
#T8( IN := #X60 ,
    PT := T#2s );
#T9( IN := #X61 ,
    PT := T#2s );

// PRODUCCIÓN
#S50 := ("FirstScan" OR (#X62)) AND NOT #FORCE_CURRENT;
#S51 := (#X50 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S52 := (#X51 AND #S5) AND NOT #FORCE_CURRENT;
#S53 := (#X52 AND #T1.Q) AND NOT #FORCE_CURRENT;
#S54 := (#X53 AND #T2.Q) AND NOT #FORCE_CURRENT;
#S55 := (#X54 AND #HaySoporte) AND NOT #FORCE_CURRENT;
#S56 := (#X55 AND #T3.Q) AND NOT #FORCE_CURRENT;
#S57 := (#X56 AND #T4.Q) AND NOT #FORCE_CURRENT;
#S58 := (#X57 AND #T5.Q) AND NOT #FORCE_CURRENT;
#S59 := (#X58 AND #T6.Q) AND NOT #FORCE_CURRENT;
#S60 := (#X59 AND #T7.Q) AND NOT #FORCE_CURRENT;
#S61 := (#X60 AND #T8.Q) AND NOT #FORCE_CURRENT;
#S62 := (#X61 AND #T9.Q) AND NOT #FORCE_CURRENT;

#X50 := (#S50 OR #X50 AND (NOT #S51)) OR #FORCE_INIT;
#X51 := (#S51 OR #X51 AND (NOT #S52)) AND NOT #FORCE_INIT;
#X52 := (#S52 OR #X52 AND (NOT #S53)) AND NOT #FORCE_INIT;
#X53 := (#S53 OR #X53 AND (NOT #S54)) AND NOT #FORCE_INIT;
#X54 := (#S54 OR #X54 AND (NOT #S55)) AND NOT #FORCE_INIT;
#X55 := (#S55 OR #X55 AND (NOT #S56)) AND NOT #FORCE_INIT;
#X56 := (#S56 OR #X56 AND (NOT #S57)) AND NOT #FORCE_INIT;
#X57 := (#S57 OR #X57 AND (NOT #S58)) AND NOT #FORCE_INIT;
#X58 := (#S58 OR #X58 AND (NOT #S59)) AND NOT #FORCE_INIT;
#X59 := (#S59 OR #X59 AND (NOT #S60)) AND NOT #FORCE_INIT;
#X60 := (#S60 OR #X60 AND (NOT #S61)) AND NOT #FORCE_INIT;
#X61 := (#S61 OR #X61 AND (NOT #S62)) AND NOT #FORCE_INIT;
#X62 := (#S62 OR #X62 AND (NOT #S50)) AND NOT #FORCE_INIT;

#Maq_Libre := #X50 AND NOT #STOP_ACT;

```

```

#QC5 := (#X51 OR #X52 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QC6 := (#X51 OR #X52 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QC7 := (#X51 OR #X52 OR #X53 OR #X54 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QCLAMP := (#X53 OR #X54 OR #X55 OR #X56 OR #X57 OR #Mantenimiento_Clumper) AND NOT #STOP_ACT;
#QBAJ := (#X55 OR #X56 OR #X59 OR #X60 OR #Mantenimiento_BajarGripper) AND NOT #STOP_ACT;
#QGRIPPER := (#X56 OR #X57 OR #X58 OR #X59 OR #Mantenimiento_Gripper) AND NOT #STOP_ACT;
#QHOR := (#X58 OR #X59 OR #X60 OR #X61 OR #Mantenimiento_MOVHOR) AND NOT #STOP_ACT;
#qsubclamp := #vaciado_subclamp AND NOT #STOP_ACT;
#HayPieza := #X54 AND NOT #STOP_ACT;
#Pieza_Empaquetada := #X62 AND NOT #STOP_ACT;

#T1a(IN:=#X51 AND NOT #RESET_ALARMA,
      PT:=T#12s);

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;

#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Maquina := #Alarma;
//  

IF #X50 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X50 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.9 Anexo 9: Grafset \_70.scl

*Listing 9: Código de la máquina 5: soporte*

```

FUNCTION_BLOCK "Grafset_70_Final"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    S5 : Bool;
    Pieza_Empaquetada : Bool;
    SigMaqLibre : Bool;
    Mantenimiento_Cinta : Bool;
    Mantenimiento_Clumper : Bool;
    Mantenimiento_SubirClamp : Bool;
    vaciado_cinta : Bool;
    vaciado_subclamp : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC5 : Bool;
    QC6 : Bool;
    QC7 : Bool;
    QCLAMP : Bool;
    QCLAMP_SUB : Bool;
    Act_SIG_MAQ : Bool;
    HayPieza : Bool;
    Alarma_Maquina : Bool;

```

```

        EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
END_VAR

VAR
    S70 : Bool;
    S71 : Bool;
    S72 : Bool;
    S73 : Bool;
    S74 : Bool;
    S75 : Bool;
    S76 : Bool;
    X70 : Bool;
    X71 : Bool;
    X72 : Bool;
    X73 : Bool;
    X74 : Bool;
    X75 : Bool;
    X76 : Bool;
    S70e : Bool;
    S71e : Bool;
    S72e : Bool;
    X70e : Bool;
    X71e : Bool;
    X72e : Bool;
    X70a : Bool;
    X71a : Bool;
    X72a : Bool;
    X73a : Bool;
    S70a : Bool;
    S71a : Bool;
    S72a : Bool;
    S73a : Bool;
    STOP_ACT : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    FORCE_CURRENT : Bool;
    FORCE_CURRENT_E : Bool;
    RESET_ALARMA : Bool;
    Alarma : Bool;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T3 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
//Flancos

// EMERGENCIA
#S70e := (#X72e AND #CI) OR "FirstScan";
#S71e := #X70e AND "Emergencia";
#S72e := #X71e;

#X70e := #S70e OR (#X70e AND NOT #S71e);
#X71e := #S71e OR (#X71e AND NOT #S72e);
#X72e := #S72e OR (#X72e AND NOT #S70e);

#FORCE_INIT_E := #X71e;
#FORCE_CURRENT_E := #X72e;

// ALARMAS
#S70a := (#X73a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S71a := (#X70a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S72a := (#X71a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S73a := (#X72a OR (#X71a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X70a := #S70a OR (#X70a AND NOT #S71a) OR #FORCE_INIT_E;

```

```

#X71a := (#S71a OR (#X71a AND NOT #S72a AND NOT #S73a)) AND NOT #FORCE_INIT_E;
#X72a := (#S72a OR (#X72a AND NOT #S73a)) AND NOT #FORCE_INIT_E;
#X73a := (#S73a OR (#X73a AND NOT #S70a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X71a;
#FORCE_INIT := #FORCE_INIT_E OR #X72a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X71a;
#RESET_ALARMA := #X73a;
#T1(IN:=#X72,
    PT:=T#1s
);
#T2(IN := #X73 ,
    PT := T#1s
);
#T3(IN := #X76 ,
    PT := T#3s
);

// PRODUCCIÓN
#S70 := ("FirstScan" OR (#X76 AND #T3.Q)) AND NOT #FORCE_CURRENT;
#S71 := (#X70 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S72 := (#X71 AND #S5) AND NOT #FORCE_CURRENT;
#S73 := (#X72 AND #T1.Q) AND NOT #FORCE_CURRENT;
#S74 := (#X73 AND #T2.Q) AND NOT #FORCE_CURRENT;
#S75 := (#X74 AND #Pieza_Empaquetada) AND NOT #FORCE_CURRENT;
#S76 := (#X75 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;

#X70 := (#S70 OR #X70 AND (NOT #S71)) OR #FORCE_INIT;
#X71 := (#S71 OR #X71 AND (NOT #S72)) AND NOT #FORCE_INIT;
#X72 := (#S72 OR #X72 AND (NOT #S73)) AND NOT #FORCE_INIT;
#X73 := (#S73 OR #X73 AND (NOT #S74)) AND NOT #FORCE_INIT;
#X74 := (#S74 OR #X74 AND (NOT #S75)) AND NOT #FORCE_INIT;
#X75 := (#S75 OR #X75 AND (NOT #S76)) AND NOT #FORCE_INIT;
#X76 := (#S76 OR #X76 AND (NOT #S70)) AND NOT #FORCE_INIT;

#Maq_Libre := #X70 AND NOT #STOP_ACT;
#QC5 := (#X71 OR #X72 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QC6 := (#X71 OR #X72 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QC7 := (#X71 OR #X72 OR #X73 OR #X76 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QCLAMP := (#X73 OR #Mantenimiento_Clamp) AND NOT #STOP_ACT;
#QCLAMP_SUB := (#X76 OR #Mantenimiento_SubirClamp OR #vaciado_subclamp) AND NOT #STOP_ACT;
#Act_SIG_MAQ := #X76 AND NOT #STOP_ACT;
#HayPieza := #X74 AND NOT #STOP_ACT;
//#Pieza_Empaquetada := X62 AND NOT STOP_ACT;

#T1a(IN:=#X71 AND NOT #RESET_ALARMA ,
    PT:=T#12s
);

#Alarma_1 := (#Alarma_1 OR (#T1a.Q)) AND NOT #RESET_ALARMA;
#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Maquina := #Alarma;
// 

IF #X70 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X70 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

```

```
END_FUNCTION_BLOCK
```

## 6.10 Anexo 10: Grafcte\_80.scl

*Listing 10: Código de la máquina 5: embalaje*

```

FUNCTION_BLOCK "grafcte_80_final"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    S5 : Bool;
    SigMaqLibre : Bool;
    Mantenimiento_Cinta : Bool;
    vaciado_cinta : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC3 : Bool;
    Act_Sig_Maq : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
    Alarma_2 : Bool;
END_VAR

VAR
    S80 : Bool;
    S81 : Bool;
    S82 : Bool;
    S83 : Bool;
    X80 : Bool;
    X81 : Bool;
    X82 : Bool;
    X83 : Bool;
    S80e : Bool;
    S81e : Bool;
    S82e : Bool;
    X80e : Bool;
    X81e : Bool;
    X82e : Bool;
    S80a : Bool;
    S81a : Bool;
    S82a : Bool;
    S83a : Bool;
    X80a : Bool;
    X81a : Bool;
    X82a : Bool;
    X83a : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    FORCE_CURRENT_E : Bool;
    FORCE_CURRENT : Bool;
    STOP_ACT : Bool;
    RESET_ALARMA : Bool;
    Alarma : Bool;
    T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
    T2a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN

```

```

//Flancos

// EMERGENCIA
#S80e := (#X82e AND #CI) OR "FirstScan";
#S81e := #X80e AND "Emergencia";
#S82e := #X81e;

#X80e := #S80e OR (#X80e AND NOT #S81e);
#X81e := #S81e OR (#X81e AND NOT #S82e);
#X82e := #S82e OR (#X82e AND NOT #S80e);

#FORCE_INIT_E := #X81e;
#FORCE_CURRENT_E := #X82e;

// ALARMAS
#S80a := (#X83a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S81a := (#X80a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S82a := (#X81a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S83a := (#X82a OR (#X81a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X80a := #S80a OR (#X80a AND NOT #S81a) OR #FORCE_INIT_E;
#X81a := (#S81a OR (#X81a AND NOT #S82a AND NOT #S83a)) AND NOT #FORCE_INIT_E;
#X82a := (#S82a OR (#X82a AND NOT #S83a)) AND NOT #FORCE_INIT_E;
#X83a := (#S83a OR (#X83a AND NOT #S80a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X81a;
#FORCE_INIT := #FORCE_INIT_E OR #X82a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X81a;
#RESET_ALARMA := #X83a;

// PRODUCCIÓN
#S80 := ("FirstScan" OR (#X83 AND NOT #S5)) AND NOT #FORCE_CURRENT;
#S81 := (#X80 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S82 := (#X81 AND #S5) AND NOT #FORCE_CURRENT;
#S83 := (#X82 AND #SigMaqLibre) AND NOT #FORCE_CURRENT;

#X80 := (#S80 OR #X80 AND (NOT #S81)) OR #FORCE_INIT;
#X81 := (#S81 OR #X81 AND (NOT #S82)) AND NOT #FORCE_INIT;
#X82 := (#S82 OR #X82 AND (NOT #S83)) AND NOT #FORCE_INIT;
#X83 := (#S83 OR #X83 AND (NOT #S80)) AND NOT #FORCE_INIT;

#Maq_Libre := #X80 AND NOT #STOP_ACT;
#QC3 := (#X81 OR #X83 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#Act_Sig_Maq := #X83 AND NOT #STOP_ACT;

#T1a(IN:=#X81,
      PT:=T#5s);
#T2a(IN := #X83,
      PT := T#5s);

//#Alarma_1 := (#Alarma_1 OR #T1a.Q) AND NOT #RESET_ALARMA;
//#Alarma_2 := (#Alarma_2 OR #T2a.Q) AND NOT #RESET_ALARMA;
//#Alarma := (#Alarma_1 OR #Alarma_2 ) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
//



IF #X80 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X80 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;

```

```

IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.11 Anexo 11: Grafset\_90.scl

*Listing 11: Código de la máquina 7: embalaje*

```

FUNCTION_BLOCK "grafset_90"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    SENSOR : Bool;
    SigMaqLibre : Bool;
    Mantenimiento_Cinta : Bool;
    Mantenimiento_Trans : Bool;
    vaicado_cinta : Bool;
    vaciado_trnas : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC3 : Bool;
    QTrans : Bool;
    Act_Sig_Maq : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
    Alarma_2 : Bool;
END_VAR

VAR
    S90 : Bool;
    S91 : Bool;
    S92 : Bool;
    S93 : Bool;
    S94 : Bool;
    X90 : Bool;
    X91 : Bool;
    X92 : Bool;
    X93 : Bool;
    X94 : Bool;
    S90e : Bool;
    S91e : Bool;
    S92e : Bool;
    X90e : Bool;
    X91e : Bool;
    X92e : Bool;
    S90a : Bool;
    S91a : Bool;
    S92a : Bool;
    S93a : Bool;
    X90a : Bool;
    X91a : Bool;
    X92a : Bool;
    X93a : Bool;
    Alarma : Bool;

```

```

FORCE_INIT : Bool;
FORCE_INIT_E : Bool;
FORCE_CURRENT : Bool;
FORCE_CURRENT_E : Bool;
STOP_ACT : Bool;
RESET_ALARMA : Bool;
T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
//Flancos

// EMERGENCIA
#S90e := (#X92e AND #CI) OR "FirstScan";
#S91e := #X90e AND "Emergencia";
#S92e := #X91e;

#X90e := #S90e OR (#X90e AND NOT #S91e);
#X91e := #S91e OR (#X91e AND NOT #S92e);
#X92e := #S92e OR (#X92e AND NOT #S90e);

#FORCE_INIT_E := #X91e;
#FORCE_CURRENT_E := #X92e;

// ALARMAS
#S90a := (#X93a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S91a := (#X90a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S92a := (#X91a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S93a := (#X92a OR (#X91a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X90a := #S90a OR (#X90a AND NOT #S91a) OR #FORCE_INIT_E;
#X91a := (#S91a OR (#X91a AND NOT #S92a AND NOT #S93a)) AND NOT #FORCE_INIT_E;
#X92a := (#S92a OR (#X92a AND NOT #S93a)) AND NOT #FORCE_INIT_E;
#X93a := (#S93a OR (#X93a AND NOT #S90a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X91a;
#FORCE_INIT := #FORCE_INIT_E OR #X92a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X91a;
#RESET_ALARMA := #X93a;

// PRODUCCIÓN
#S90 := ("FirstScan" OR (#X94)) AND NOT #FORCE_CURRENT;
#S91 := (#X90 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S92 := (#X91 AND #SENSOR) AND NOT #FORCE_CURRENT;
#S93 := (#X92 AND #SigMaQLibre) AND NOT #FORCE_CURRENT;
#S94 := (#X93 AND NOT #SENSOR) AND NOT #FORCE_CURRENT;

#X90 := (#S90 OR #X90 AND (NOT #S91)) OR #FORCE_INIT;
#X91 := (#S91 OR #X91 AND (NOT #S92)) AND NOT #FORCE_INIT;
#X92 := (#S92 OR #X92 AND (NOT #S93)) AND NOT #FORCE_INIT;
#X93 := (#S93 OR #X93 AND (NOT #S94)) AND NOT #FORCE_INIT;
#X94 := (#S94 OR #X94 AND (NOT #S90)) AND NOT #FORCE_INIT;

#Maq_Libre := #X90 AND NOT #STOP_ACT;
#QC3 := (#X91 OR #Mantenimiento_Cinta OR #vaicado_cinta) AND NOT #STOP_ACT;
#QTrans := (#X93 OR #Mantenimiento_Trans OR #vaciado_trnas) AND NOT #STOP_ACT;
#Act_Sig_Maq := #X94 AND NOT #STOP_ACT;

#T1a(IN:=#X91 AND NOT #RESET_ALARMA,
PT:=T#5s);

```

```

#Alarma_1 := (#Alarma_1 OR #T1a.Q) AND NOT #RESET_ALARMA;
#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);

IF #X90 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X90 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.12 Anexo 12: Grafset\_100.scl

*Listing 12: Código de la máquina 8: embalaje*

```

FUNCTION_BLOCK "grafset_100"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Proceso : Bool;
    S5 : Bool;
    Pieza_Empaquetada : Bool;
    Mantenimiento_Cinta : Bool;
    Mantenimiento_Clamp : Bool;
    Mantenimiento_SubirClamp : Bool;
    vaciado_cinta : Bool;
    vaciado_subclamp : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
    PiezasPalet : Int := 2;
END_VAR

VAR_OUTPUT
    Maq_Libre : Bool;
    QC7 : Bool;
    QCLAMP : Bool;
    QCLAMP_SUB : Bool;
    HayPieza : Bool;
    Pieza_Entregada : Bool;
    Alarma_Maquina : Bool;
    EstadoMaquina : Int;
END_VAR

VAR_IN_OUT
    Alarma_1 : Bool;
END_VAR

VAR
    S100 : Bool;
    S101 : Bool;
    S102 : Bool;
    S103 : Bool;
    S104 : Bool;
    S105 : Bool;
    S106 : Bool;
    X100 : Bool;
    X102 : Bool;

```

```

X101 : Bool;
X103 : Bool;
X104 : Bool;
X105 : Bool;
X106 : Bool;
S100e : Bool;
S101e : Bool;
S102e : Bool;
X100e : Bool;
X101e : Bool;
X102e : Bool;
S100a : Bool;
S101a : Bool;
S102a : Bool;
S103a : Bool;
X100a : Bool;
X101a : Bool;
X102a : Bool;
X103a : Bool;
FORCE_CURRENT_E : Bool;
FORCE_CURRENT : Bool;
FORCE_INIT : Bool;
FORCE_INIT_E : Bool;
STOP_ACT : Bool;
RESET_ALARMA : Bool;
Alarma : Bool;
S107 : Bool;
X107 : Bool;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T3 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
Trig105 {InstructionName := 'R_TRIG'; LibVersion := '1.0'} : R_TRIG;
END_VAR
VAR RETAIN
"Counter" {InstructionName := 'CTU_INT'; LibVersion := '1.0'} : CTU_INT;
END_VAR
VAR
T1a {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
//Flancos

// EMERGENCIA
#S100e := (#X102e AND #CI) OR "FirstScan";
#S101e := #X100e AND "Emergencia";
#S102e := #X101e;

#X100e := #S100e OR (#X100e AND NOT #S101e);
#X101e := #S101e OR (#X101e AND NOT #S102e);
#X102e := #S102e OR (#X102e AND NOT #S100e);

#FORCE_INIT_E := #X101e;
#FORCE_CURRENT_E := #X102e;

// ALARMAS
#S100a := (#X103a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S101a := (#X100a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S102a := (#X101a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S103a := (#X102a OR (#X101a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X100a := #S100a OR (#X100a AND NOT #S101a) OR #FORCE_INIT_E;
#X101a := (#S101a OR (#X101a AND NOT #S102a AND NOT #S103a)) AND NOT #FORCE_INIT_E;
#X102a := (#S102a OR (#X102a AND NOT #S103a)) AND NOT #FORCE_INIT_E;
#X103a := (#S103a OR (#X103a AND NOT #S100a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X101a;
#FORCE_INIT := #FORCE_INIT_E OR #X102a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X101a;
#RESET_ALARMA := #X103a;
#T1(IN:=#X102,

```

```

        PT := T#1s
    );
#T2( IN := #X103 ,
      PT := T#1s
);
#T3( IN := #X106 ,
      PT := T#3s
);

#Trig105(CLK:=#X105);

#Counter(CU:=#Trig105.Q,
          R := #X100,
          PV:=#PiezasPalet);

// PRODUCCIÓN
#S100 := ("FirstScan" OR (#X107)) AND NOT #FORCE_CURRENT;
#S101 := (#X100 AND #Act_Proceso) AND NOT #FORCE_CURRENT;
#S102 := (#X101 AND $S5) AND NOT #FORCE_CURRENT;
#S103 := (#X102 AND #T1.Q) AND NOT #FORCE_CURRENT;
#S104 := (#X103 AND #T2.Q) OR (#X105 AND #Counter.CV < #PiezasPalet) AND NOT #FORCE_CURRENT;
#S105 := (#X104 AND #Pieza_Empaquetada) AND NOT #FORCE_CURRENT;
#S106 := (#X105 AND #Counter.CV = #PiezasPalet) AND NOT #FORCE_CURRENT;
#S107 := (#X106 AND #T3.Q) AND NOT #FORCE_CURRENT;

#X100 := (#S100 OR #X100 AND (NOT #S101)) OR #FORCE_INIT;
#X101 := (#S101 OR #X101 AND (NOT #S102)) AND NOT #FORCE_INIT;
#X102 := (#S102 OR #X102 AND (NOT #S103)) AND NOT #FORCE_INIT;
#X103 := (#S103 OR #X103 AND (NOT #S104)) AND NOT #FORCE_INIT;
#X104 := (#S104 OR #X104 AND (NOT #S105)) AND NOT #FORCE_INIT;
#X105 := (#S105 OR #X105 AND (NOT #S106 AND NOT #S104)) AND NOT #FORCE_INIT;
#X106 := (#S106 OR #X106 AND (NOT #S107)) AND NOT #FORCE_INIT;
#X107 := (#S107 OR #X107 AND (NOT #S100)) AND NOT #FORCE_INIT;

#Maq_Libre := #X100 AND NOT #STOP_ACT;

#QC7 := (#X101 OR #X102 OR #X106 OR #Mantenimiento_Cinta OR #vaciado_cinta) AND NOT #STOP_ACT;
#QCLAMP := (#X103 OR #Mantenimiento_Clumper) AND NOT #STOP_ACT;
#QCLAMP_SUB := (#X106 OR #Mantenimiento_SubirClamp OR #vaciado_subclamp) AND NOT #STOP_ACT;
#HayPieza := #X104 AND NOT #STOP_ACT;
#Pieza_Entregada := #X107;

#T1a(IN := #X101 AND NOT #RESET_ALARMA,
      PT := T#8s);

#Alarma_1 := (#Alarma_1 OR (#X101 AND #T1a.Q)) AND NOT #RESET_ALARMA;
#Alarma := (#Alarma_1) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
#Alarma_Mquina := #Alarma;

IF #X100 = TRUE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 0;
END_IF;
IF #X100 = FALSE AND NOT #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 1;
END_IF;
IF #Alarma AND NOT "Emergencia" THEN
    #EstadoMaquina := 2;
END_IF;
IF "Emergencia" THEN
    #EstadoMaquina := 3;
END_IF;

END_FUNCTION_BLOCK

```

## 6.13 Anexo 13: Grafset\_110.scl

*Listing 13: Funciones de mantenimiento*

```

FUNCTION_BLOCK "Grafset_110_Mantenimiento"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Mantenimiento : Bool;
    Act_Mantenimiento_Emergencia : Bool;
    Act_Mantenimiento_HMI : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
END_VAR

VAR_OUTPUT
    Mantenimiento_Activado : Bool;
    Activar_Cintas : Bool;
    Activar_Transfers : Bool;
    BAJAR_ROBOTS : Bool;
    SUBIR_CLAMPERS : Bool;
    GRIPPERS : Bool;
    CLAMPER : Bool;
    MOVER_HOR : Bool;
    Fin_Mantenimiento : Bool;
END_VAR

VAR
    X110 : Bool;
    X111 : Bool;
    X112 : Bool;
    X113 : Bool;
    X114 : Bool;
    X115 : Bool;
    X116 : Bool;
    X117 : Bool;
    S110 : Bool;
    S111 : Bool;
    S112 : Bool;
    S113 : Bool;
    S114 : Bool;
    S115 : Bool;
    S116 : Bool;
    S117 : Bool;
    Sensores : Bool;
    FORCE_CURRENT : Bool;
    FORCE_CURRENT_E : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    STOP_ACT : Bool;
    S110e : Bool;
    S111e : Bool;
    S112e : Bool;
    X110e : Bool;
    X111e : Bool;
    X112e : Bool;
    S110a : Bool;
    S111a : Bool;
    S112a : Bool;
    S113a : Bool;
    X110a : Bool;
    X111a : Bool;
    X112a : Bool;
    X113a : Bool;
    Alarma : Bool;
    RESET_ALARMA : Bool;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T2 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T3 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T4 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
T5 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;

```

```

END_VAR

BEGIN
//Flancos

// EMERGENCIA
#S110e := (#X112e AND #CI) OR "FirstScan";
#S111e := #X110e AND FALSE;
#S112e := #X111e;

#X110e := #S110e OR (#X110e AND NOT #S111e);
#X111e := #S111e OR (#X111e AND NOT #S112e);
#X112e := #S112e OR (#X112e AND NOT #S110e);

#FORCE_INIT_E := #X111e;
#FORCE_CURRENT_E := #X112e;

// ALARMAS
#S110a := (#X113a AND NOT #FORCE_CURRENT_E) OR "FirstScan";
#S111a := (#X110a AND #Alarma) AND NOT #FORCE_CURRENT_E;
#S112a := (#X111a AND #Reset) AND NOT #FORCE_CURRENT_E;
#S113a := (#X112a OR (#X111a AND #Resume)) AND NOT #FORCE_CURRENT_E;

#X110a := #S110a OR (#X110a AND NOT #S111a) OR #FORCE_INIT_E;
#X111a := (#S111a OR (#X111a AND NOT #S112a AND NOT #S113a)) AND NOT #FORCE_INIT_E;
#X112a := (#S112a OR (#X112a AND NOT #S113a)) AND NOT #FORCE_INIT_E;
#X113a := (#S113a OR (#X113a AND NOT #S110a)) AND NOT #FORCE_INIT_E;

#STOP_ACT := #X111a;
#FORCE_INIT := #FORCE_INIT_E OR #X112a;
#FORCE_CURRENT := #FORCE_CURRENT_E OR #X111a;
#RESET_ALARMA := #X113a;

#T1( IN:=#X112,
    PT:=T#2s
);
#T2( IN := #X113 ,
    PT := T#2s
);
#T3( IN := #X114 ,
    PT := T#2s
);
#T4( IN := #X115 ,
    PT := T#2s
);
#T5( IN := #X116 ,
    PT := T#2s
);

#Sensores := FALSE;

// PRODUCCIÓN
#S110 := ("FirstScan" OR (#X117)) AND NOT #FORCE_CURRENT;
//#S111 := (#X110 AND (#Act_Mantenimiento OR #Act_Mantenimiento_Emergencia)) AND NOT #FORCE_CURRENT;
#S111 := (#X110 AND (#Act_Mantenimiento OR #Act_Mantenimiento_Emergencia OR #Act_Mantenimiento_HMI));
#S112 := (#X111 AND NOT #Sensores) AND NOT #FORCE_CURRENT;
#S113 := (#X112 AND #T1.Q) AND NOT #FORCE_CURRENT;
#S114 := (#X113 AND #T2.Q) AND NOT #FORCE_CURRENT;
#S115 := (#X114 AND #T3.Q) AND NOT #FORCE_CURRENT;
#S116 := (#X115 AND #T4.Q) AND NOT #FORCE_CURRENT;
#S117 := (#X116 AND #T5.Q) AND NOT #FORCE_CURRENT;

#X110 := (#S110 OR #X110 AND (NOT #S111)) OR #FORCE_INIT;
#X111 := (#S111 OR #X111 AND (NOT #S112)) AND NOT #FORCE_INIT;
#X112 := (#S112 OR #X112 AND (NOT #S113)) AND NOT #FORCE_INIT;
#X113 := (#S113 OR #X113 AND (NOT #S114)) AND NOT #FORCE_INIT;
#X114 := (#S114 OR #X114 AND (NOT #S115)) AND NOT #FORCE_INIT;
#X115 := (#S115 OR #X115 AND (NOT #S116)) AND NOT #FORCE_INIT;

```

```

#X116 := (#S116 OR #X116 AND (NOT #S117)) AND NOT #FORCE_INIT;
#X117 := (#S117 OR #X117 AND (NOT #S110)) AND NOT #FORCE_INIT;

#Activar_Cintas := #X112 AND NOT #STOP_ACT;
#Activar_Transfers := #X113 AND NOT #STOP_ACT;
#BAJAR_ROBOTS := #X114 AND NOT #STOP_ACT;
#SUBIR_CLAMPERS := #X114 AND NOT #STOP_ACT;
#GRIPPERS := #X115 AND NOT #STOP_ACT;
#CLAMPER := #X115 AND NOT #STOP_ACT;
#MOVER_HOR := #X116 AND NOT #STOP_ACT;
#Fin_Mantenimiento := #X117 AND NOT #STOP_ACT;

//#Alarma_1 := (#Alarma_1 OR (#X31 AND #T1.Q)) AND NOT RESET_ALARMA;
//#Alarma_2 := (#Alarma_2 OR (#X33 AND #T2.Q)) AND NOT RESET_ALARMA;

//#Alarma := (#Alarma_1 OR #Alarma_2 OR #Alarma_3) AND NOT (#FORCE_INIT OR #RESET_ALARMA);
END_FUNCTION_BLOCK

```

## 6.14 Anexo 14: Grafset\_120.scl

*Listing 14: Código de vaciado general*

```

FUNCTION_BLOCK "grafset_120_vaciado_general"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    Act_Vaciado : Bool;
    Act_Vaciado_HMI : Bool;
    CI : Bool;
    Reset : Bool;
    Resume : Bool;
    fin_vac_1 : Bool;
    fin_vac_2 : Bool;
    fin_vac_3 : Bool;
    fin_sop_1 : Bool;
    fin_sop_2 : Bool;
    fin_sop_3 : Bool;
    fin_emb_1 : Bool;
    fin_emb_2 : Bool;
    fin_emb_3 : Bool;
    fin_emb_4 : Bool;
END_VAR

VAR_OUTPUT
    Vaciado_activado : Bool;
    act_vaciados_maquinas : Bool;
    Fin_vaciado : Bool;
END_VAR

VAR
    vac_finalizados : Bool;
    s120 : Bool;
    s121 : Bool;
    s122 : Bool;
    x120 : Bool;
    x121 : Bool;
    x122 : Bool;
    s120e : Bool;
    s121e : Bool;
    s122e : Bool;
    x120e : Bool;
    x121e : Bool;
    x122e : Bool;
    FORCE_CURRENT : Bool;
    FORCE_CURRENT_E : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
END_VAR

```

```

BEGIN
  //Flancos

  // EMERGENCIA
  #s120e := (#x122e AND #CI) OR "FirstScan";
  #s121e := #x120e AND FALSE;
  #s122e := #x121e;

  #x120e := #s120e OR (#x120e AND NOT #s121e);
  #x121e := #s121e OR (#x121e AND NOT #s122e);
  #x122e := #s122e OR (#x122e AND NOT #s120e);

  #FORCE_INIT_E := #x121e;
  #FORCE_CURRENT_E := #x122e;

  #FORCE_INIT := #FORCE_INIT_E;
  #FORCE_CURRENT := #FORCE_CURRENT_E ;

  #vac_finalizados := #fin_vac_1 AND #fin_vac_2 AND #fin_vac_3 AND #fin_sop_1 AND #fin_sop_2 AND #fin_sop_3 AND #fin_sop_4;

  // PRODUCCIÓN
  #s120 := ("FirstScan" OR (#x122)) AND NOT #FORCE_CURRENT;
  #s121 := (#x120 AND (#Act_Vaciado OR #Act_Vaciado_HMI)) AND NOT #FORCE_CURRENT;
  #s122 := (#x121 AND #vac_finalizados) AND NOT #FORCE_CURRENT;

  #x120 := (#s120 OR #x120 AND (NOT #s121)) OR #FORCE_INIT;
  #x121 := (#s121 OR #x121 AND (NOT #s122)) AND NOT #FORCE_INIT;
  #x122 := (#s122 OR #x122 AND (NOT #s120)) AND NOT #FORCE_INIT;

  #act_vaciados_maquinas := #x121;
  #Fin_vaciado := #x122;

END_FUNCTION_BLOCK

```

## 6.15 Anexo 15: Grafset\_125.scl

*Listing 15: Código de vaciado tipo 1*

```

FUNCTION_BLOCK "grafset_125_vaciado_tipo1"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Act_Vaciado_Maquina : Bool;
  s2 : Bool;
  S3_2 : Bool;
  S3_3 : Bool;
  S3_1 : Bool;
  Fin_vaciado : Bool;
  CI : Bool;
END_VAR

VAR_OUTPUT
  QCINTA1 : Bool;
  QCINTA2 : Bool;
  QTrans : Bool;
  FIN_Vaciado_MAq : Bool;
END_VAR

VAR
  S125 : Bool;
  S126 : Bool;

```

```

S126a : Bool;
S126b : Bool;
S127 : Bool;
X125 : Bool;
X126 : Bool;
x126a : Bool;
x126b : Bool;
x127 : Bool;
s1250e : Bool;
s1251e : Bool;
s1252e : Bool;
x1250e : Bool;
x1251e : Bool;
x1252e : Bool;
FORCE_INIT : Bool;
FORCE_INIT_E : Bool;
FORCE_CURRENT_E : Bool;
S3 : Bool;
T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
F_S2 {InstructionName := 'F_TRIG'; LibVersion := '1.0'} : F_TRIG;
END_VAR

BEGIN
// EMERGENCIA
#s1250e := (#x1252e AND #CI) OR "FirstScan";
#s1251e := #x1250e AND FALSE;
#s1252e := #x1251e;

#x1250e := #s1250e OR (#x1250e AND NOT #s1251e);
#x1251e := #s1251e OR (#x1251e AND NOT #s1252e);
#x1252e := #s1252e OR (#x1252e AND NOT #s1250e);

#FORCE_INIT_E := #x1251e;
#FORCE_CURRENT_E := #x1252e;

#S3 := #S3_1 OR #S3_2 OR #S3_3;

#T1(IN:=#X126,
    PT:=T#10s);
#F_S2(CLK:=#s2);

// PRODUCCIÓN
#S125 := ("FirstScan" OR (#x126b AND NOT #S3_1) OR (#x127 AND #Fin_vaciado)) AND NOT #FORCE_CURRENT_E;
#S126 := ((#X125 AND #Act_Vaciado_Maquina) OR (#x126b AND NOT #S3)) AND NOT #FORCE_CURRENT_E;
#S126a := (#X126 AND #F_S2.Q) AND NOT #FORCE_CURRENT_E;
#S126b := ((#X126 AND #S3) OR (#x126a AND #S3)) AND NOT #FORCE_CURRENT_E;
#S127 := (#X126 AND #T1.Q) AND NOT #FORCE_CURRENT_E;

// hay que revisar lo del FORCE_INIT
#X125 := (#S125 OR #X125 AND NOT #S126) OR #FORCE_INIT_E;
#X126 := (#S126 OR #X126 AND (NOT #S126a AND NOT #S127 AND NOT #S126b)) AND NOT #FORCE_INIT_E;
#x126a := (#S126a OR #x126a AND NOT #S126b) AND NOT #FORCE_INIT_E;
#x126b := (#S126b OR #x126b AND NOT #S125) AND NOT #FORCE_INIT_E;
#x127 := (#S127 OR #x127 AND NOT #S125) AND NOT #FORCE_INIT_E;

#QCINTA1 := #X126;
#QCINTA2 := #X126 OR #x126a;
#FIN_Vaciado_MAq := #x127;
#QTrans := #x126b;
END_FUNCTION_BLOCK

```

## 6.16 Anexo 16: Grafset\_130.scl

*Listing 16: Código de vaciado tipo 2*

```
FUNCTION_BLOCK "grafset_130_vaciados_maq"
```

```

{ S7_Optimized_Access := 'TRUE' }

VERSION : 0.1
VAR_INPUT
    Act_Vaciado_maquina : Bool;
    Fin_vaciado : Bool;
    CI : Bool;
END_VAR

VAR_OUTPUT
    Fin_vaciado_maq : Bool;
    Vaciados_maq_1 : Bool;
    Vaciados_maq_2 : Bool;
    Vaciados_maq_3 : Bool;
    Vaciados_maq_4 : Bool;
    Vaciados_maq_5 : Bool;
    Vaciados_maq_6 : Bool;
    Vaciados_maq_7 : Bool;
    Vaciados_maq_8 : Bool;
END_VAR

VAR
    S130 : Bool;
    S131 : Bool;
    s132 : Bool;
    x130 : Bool;
    x131 : Bool;
    x132 : Bool;
    s130e : Bool;
    s131e : Bool;
    s132e : Bool;
    x130e : Bool;
    x131e : Bool;
    x132e : Bool;
    FORCE_CURRENT : Bool;
    FORCE_CURRENT_E : Bool;
    FORCE_INIT : Bool;
    FORCE_INIT_E : Bool;
    T1 {InstructionName := 'TON_TIME'; LibVersion := '1.0'} : TON_TIME;
END_VAR

BEGIN
    //Flancos

    // EMERGENCIA
    #s130e := (#x132e AND #CI) OR "FirstScan";
    #s131e := #x130e AND FALSE;
    #s132e := #x131e;

    #x130e := #s130e OR (#x130e AND NOT #s131e);
    #x131e := #s131e OR (#x131e AND NOT #s132e);
    #x132e := #s132e OR (#x132e AND NOT #s130e);

    #FORCE_INIT_E := #x131e;
    #FORCE_CURRENT_E := #x132e;

    #FORCE_INIT := #FORCE_INIT_E ;
    #FORCE_CURRENT := #FORCE_CURRENT_E ;
    #T1(IN:=#x131,
        PT:=T#20s);

    // PRODUCCIÓN
    #S130 := ("FirstScan" OR (#x132 AND #Fin_vaciado )) AND NOT #FORCE_CURRENT;
    #S131 := (#x130 AND #Act_Vaciado_maquina) AND NOT #FORCE_CURRENT;
    #s132 := (#x131 AND #T1.Q) AND NOT #FORCE_CURRENT;

    #x130 := (#S130 OR #x130 AND (NOT #S131)) OR #FORCE_INIT;

```

```

#x131 := (#S131 OR #x131 AND (NOT #s132)) AND NOT #FORCE_INIT;
#x132 := (#s132 OR #x132 AND (NOT #S130)) AND NOT #FORCE_INIT;

#Vaciados_maq_1 := #x131;
#Vaciados_maq_2 := #x131;
#Vaciados_maq_3 := #x131;
#Vaciados_maq_4 := #x131;
#Vaciados_maq_5 := #x131;
#Vaciados_maq_6 := #x131;
#Vaciados_maq_7 := #x131;
#Vaciados_maq_8 := #x131;

#Fin_vaciado_maq := #x132;

END_FUNCTION_BLOCK

```

## 6.17 Anexo 17: Grafset\_145.scl

*Listing 17: Modo manual*

```

FUNCTION_BLOCK "grafset_145_manual_maquinas"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    act_manual_maq : Bool;
    CI : Bool;
    Man_qci : Bool;
    man_Qc1b : Bool;
    man_qc2 : Bool;
    man_qc3 : Bool;
    man_qc4 : Bool;
    man_qtrani : Bool;
    man_Qtran2 : Bool;
END_VAR

VAR_OUTPUT
    qcin1 : Bool;
    qcin1b : Bool;
    qcin2 : Bool;
    qcin3 : Bool;
    qcin4 : Bool;
    qtrans1 : Bool;
    qtrans2 : Bool;
END_VAR

VAR
    s145 : Bool;
    s146 : Bool;
    x145 : Bool;
    x146 : Bool;
    s145e : Bool;
    s146e : Bool;
    s147e : Bool;
    x145e : Bool;
    x146e : Bool;
    x147e : Bool;
    force_init : Bool;
    force_current : Bool;
    force_init_e : Bool;
    force_current_e : Bool;
END_VAR

BEGIN
    //Flancos

    // EMERGENCIA

```

```

#s145e := (#x147e AND #CI) OR "FirstScan";
#s146e := #x145e AND "Emergencia";
#s147e := #x146e;

#x145e := #s145e OR (#x145e AND NOT #s146e);
#x146e := #s146e OR (#x146e AND NOT #s147e);
#x147e := #s147e OR (#x147e AND NOT #s145e);

#force_init_e := #x146e;
#force_current_e := #x147e;

#force_init := #force_init_e ;
#force_current := #force_current_e ;

// PRODUCCIÓN
#s145 := ("FirstScan" OR (#x146 AND NOT #act_manual_maq )) AND NOT #force_current;
#s146 := (#x145 AND #act_manual_maq) AND NOT #force_current;

#x145 := (#s145 OR #x145 AND (NOT #s146)) OR #force_init;
#x146 := (#s146 OR #x146 AND (NOT #s145)) AND NOT #force_init;

#qcini := (#x146 AND #Man_qc1) AND NOT "Emergencia";
#qcini1b := (#x146 AND #man_Qc1b) AND NOT "Emergencia";
#qcini2 := (#x146 AND #man_qc2) AND NOT "Emergencia";
#qcini3 := (#x146 AND #man_qc3) AND NOT "Emergencia";
#qcini4 := (#x146 AND #man_qc4) AND NOT "Emergencia";
#qtrans1 := (#x146 AND #man_qtran1) AND NOT "Emergencia";
#qtrans2 := (#x146 AND #man_Qtran2) AND NOT "Emergencia";

END_FUNCTION_BLOCK

```