



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“IMPLEMENTACIÓN DE UN SISTEMA DE  
AUTOMATIZACIÓN PARA UN EQUIPO MECATRÓNICO  
INDUSTRIAL DE ALMACENAMIENTO DE PAQUETES DE  
BOTELLA”**

**INFORME DE MATERIA INTEGRADORA**

Previo a la obtención del Título de:

**INGENIERO EN ELECTRICIDAD, ELECTRÓNICA Y  
AUTOMATIZACIÓN INDUSTRIAL**

MIGUEL ALBERTO DUMANI CARRILLO

STALYN DAVID RODRIGUEZ GOMEZ

GUAYAQUIL – ECUADOR

AÑO: 2017



## AGRADECIMIENTOS

Agradezco a Dios por permitirme llegar a este momento de mi vida, darme fortaleza y ánimos para superar cualquier adversidad que se hacía presente.

A mis padres Miguel Dumani y Patricia Carrillo, por su amor y sus consejos, desde que tengo uso de razón siempre me apoyaron en todo de forma incondicional. A mi hermano Patricio Dumani, por su cariño y su forma única de ayudarme.

A la Mami Albita, Rosa Figueroa por estar dispuesta a ayudarme en cualquier momento. A mi Tesorito, Libia Burgos siento que estás siempre conmigo, faltaron más cosas por vivir, pero sé que este momento sería especial para ambos, A mis tíos, primos y Filito.

A todos mis compañeros por su apoyo a lo largo de toda mi carrera universitaria, la lista es larga...

Al PhD. Douglas Plaza y el Ing. Livingston Miranda por la confianza, apoyo y amistad recibida en el desarrollo de este proyecto, MUCHAS GRACIAS.

*Miguel Alberto Dumani Carrillo*

Agradezco a Dios por ser el tutor de mi vida, la fuerza que me hizo falta, la luz de mi sendero y el guía para cada paso que he dado en el transcurso de mi vida, a tu lado lo he podido todo y sé que siempre caminarás junto a mi sin importar a donde me dirija.

A mis padres, Mónica Gómez y José Rodríguez por ser las personas que me dieron la vida, y la oportunidad de avanzar con mis estudios, a mi tía Mariana Jiménez, la cual considero como mi segunda madre, por su apoyo incondicional en cada etapa de mi vida, a mi abuela, Zulema Jiménez por la crianza que me dio y los valores que me inculco.

A mis hermanos, José Rodríguez, Steven Rodríguez y Marilyn Rodríguez por ser los pilares fundamentales en el transcurso de esta dura etapa, apoyándome y animándome en cada momento de mi vida.

*Stalyn David Rodríguez Gómez*

## DEDICATORIA

Este proyecto se lo dedico a mis padres que lo han dado todo por de mí, cuando necesité de ellos siempre estaban, son la fuerza que me han sostenido en mis peores momentos, gracias por no permitir que renuncie en el camino. A toda mi familia por ayudarme a alcanzar esta meta, a mi hermano, a Bachita y a todos los que hicieron posible este logro.

*Miguel Alberto Dumani Carrillo*

Este proyecto se lo dedico a mi familia entera por ser el apoyo que necesité en esa etapa de mi vida, en especial a mi madre y padre que fueron siempre los que estuvieron a mi lado en todo momento ayudándome a continuar por este camino, a mis amigos y profesores.

*Stalyn David Rodríguez Gómez*

## **TRIBUNAL DE EVALUACIÓN**

---

**Ing. Franklin Kuonquí Gaínza**

PROFESOR DE MATERIA  
INTEGRADORA

---

**PhD. Douglas Antonio Plaza Guingla**

TUTOR ACADÉMICO

## **DECLARACIÓN EXPRESA**

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOI realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Stalyn Rodríguez

Miguel Dumani

## RESUMEN

El presente trabajo consiste en la automatización de una planta mecatrónica real, para lo cual se implementó un algoritmo de control para el almacenamiento y despacho de una bodega, en el sub sistema IMS 8 de LUCAS-NÜLLE.

Inicialmente, se parte de la programación original que poseía el subsistema, esta programación sólo realizaba un solo proceso que involucraba todos los subsistemas de las planta. La programación que poseía permitía almacenar de una manera ordenada. En el momento de despachar, despachaba la última que almaceno, lo que generaba un problema en el cual, no se realizaba un correcto uso de la bodega tanto al momento de almacenar como al despachar debido a que, si se trataba de productos con fecha de expiración estos podrían caducar ya que, la última en despachar sería la primera que ingresó.

En primeras instancias se independiza el subsistema IMS 8 de la planta, para luego, implementar un algoritmo de control que permita que el primer paquete de botella en entrar sea la primera en salir, esto se lo realizó utilizando dos matrices una de valores lógicos y otra de enteros. La matriz de valores lógicos permitió saber si un puesto está ocupado o vacío para poder proceder a almacenar o no. Mientras, la matriz de enteros tomó un rol importante en la etapa de despacho ya que, se iban asignando números enteros mayores a 0 en forma ascendente cada vez que se realizaba un almacenamiento, por medio de una función integrada en la programación se tomaba el menor número de la matriz lo que permitió despachar la primera pieza colocada en la bodega.

Finalmente, se diseñó un modelo en 3D en un software CAD para la visualización de la planta, acompañada de unas prácticas para el uso netamente académica por parte de los estudiantes.

# ÍNDICE GENERAL

RESUMEN .....	i
ÍNDICE GENERAL.....	ii
ABREVIATURAS .....	iv
SIMBOLOGÍA .....	v
ÍNDICE DE FIGURAS .....	vi
CAPÍTULO 1 .....	1
1. DELIMITACIÓN DEL PROBLEMA .....	1
1.1 Planteamiento del Problema.....	1
1.1 Objetivos.....	2
1.1.1 Objetivo General.....	2
1.1.2 Objetivos Específicos .....	2
1.2 Justificación.....	2
1.3 Alcance del proyecto .....	3
CAPÍTULO 2.....	4
2. Metodología de trabajo.....	4
2.1 Planteamiento del proceso. ....	4
2.2 Identificación de las partes del subsistema IMS 8 .....	5
2.3 Diseño del algoritmo de control en Grafcet.....	5
2.4 Implementación del algoritmo.....	13
2.4.1 Descripción del bloque de funciones secuenciales (SFC).....	13
2.4.2 Descripción del bloque de funciones (FBD). ....	14
2.4.3 Descripción del bloque principal MAIN. ....	16
2.5 Comunicación general del subsistema IMS8. ....	17
2.5.1 Comunicación PLC S7-300.....	18
2.6 Elaboración de la planta IMS 8 en CAD. ....	19



2.6.1	Diseño de las piezas de la planta. ....	19
2.6.2	Diseño del ensamble de las piezas de la planta. ....	19
CAPÍTULO 3.....		21
3.	RESULTADOS.....	21
3.1	Comunicación de dispositivos. ....	21
3.2	Control secuencial.....	21
3.3	Sistema Scada del proceso.....	24
3.4	Modelado 3D del subsistema IMS 8. ....	25
CONCLUSIONES Y RECOMENDACIONES.....		27
BIBLIOGRAFÍA.....		28
ANEXOS.....		30

## **ABREVIATURAS**

ESPOL	Escuela Superior Politécnica del Litoral
ASTM	American Society for Testing and Materials
NACE	National Association of Corrosion Engineer
SSC	Electrodo de Plata Cloruro de Plata
CSE	Electrodo de Cobre Sulfato de Cobre
HWL	High Water Level
LWL	Low Water Level
CIS	Inspección pasó a paso, medición de potenciales de encendido
MPY	Milésimas de pulgadas por año

## SIMBOLOGÍA

mil	Milésima de pulgada
mg	Miligramo
pH	Potencial de Hidrógeno
m	Metro
mV	Mili voltio
Cu	Cobre
Ni	Níquel
C	Carbono
Mn	Manganeso
P	Fósforo

# ÍNDICE DE FIGURAS

FIGURA 2.1: Planteamiento del desarrollo de la solución del problema .....	4
FIGURA 2.2: NIVEL DE ESTANTES POR FILAS .....	5
FIGURA 2.3: BLOQUE STAD Y HRSA .....	6
FIGURA 2.4: BOSQUEJO TAREAS PLANTA IMS 8 .....	7
FIGURA 2.5: CONTROL DE POSICIONAMIENTO INICIAL DEL BRAZO.....	8
FIGURA 2.6: CONTROL CONVENCIONAL DE ALMACENAMIENTO. ....	10
FIGURA 2.7: CONTROL CONVENCIONAL DEL DESPACHO .....	12
FIGURA 2.8: CONTROL SECUENCIAL (HRSA-STAD).....	13
FIGURA 2.9: BLOQUE DE FUNCIONES SFC .....	14
FIGURA 2.10: BLOQUES DE FUNCIONES FBD .....	14
FIGURA 2.11: PARTE DEL ALGORITMO DE CONTROL .....	15
FIGURA 2.12: BLOQUE DB DEL FB_IMS8_STAD .....	16
FIGURA 2.13: BLOQUE FB FB_IMS8_X_POSITION .....	16
FIGURA 2.14: SEGMENTOS DEL MAIN PRINCIPAL.....	17
FIGURA 2.15: TIPO DE COMUNICACIÓN ENTRE DISPOSITIVOS .....	17
FIGURA 2.16: COMUNICACIÓN PROFIBUS ENTRE PLC Y ESCLAVO.....	18
FIGURA 2.17: COMUNICACIÓN ETHERNET ENTRE PLC Y HMI.....	18
FIGURA 2.18: MEDICIONES TOMADAS PARA EL DISEÑO DE LA PLANTA.....	19
FIGURA 2.19: CREACIÓN DE PIEZA .....	19
FIGURA 2.20: ENSAMBLES DE LAS PIEZAS DE LA PLANTA.....	20
FIGURA 3.1: COMUNICACIÓN EFECTUADA ENTRE DISPOSITIVOS. ....	21
FIGURA 3.2: COMPILACIÓN EFECTUADA CON ÉXITO.....	22
FIGURA 3.3: CARGA DEL PROGRAMA AL PLC .....	22
FIGURA 3.4: VISUALIZACIÓN DE ENTRADAS/SALIDAS EN LÍNEA .....	22
FIGURA 3.5: MOVIMIENTO INICIAL DEL BRAZO PARA ALMACENAMIENTO .....	23
FIGURA 3.6: MOVIMIENTO FINAL DEL BRAZO PARA ALMACENAMIENTO .....	23
FIGURA 3.7: MOVIMIENTO INICIAL DEL BRAZO PARA DESPACHO .....	23
FIGURA 3.8: MOVIMIENTO FINAL DEL BRAZO PARA DESPACHO .....	24
FIGURA 3.9: MENÚ PRINCIPAL DEL HMI.....	24
FIGURA 3.10: SENSORES Y ACTUADORES DEL IMS 8 .....	25
FIGURA 3.11: BOSQUEJO DEL ESTANTE ELEVADO .....	25
FIGURA 3.12: ENSAMBLES REALIZADOS DETALLADAMENTE .....	25
FIGURA 3.13: DISEÑO TOTAL DEL SUB SISTEMA IMS 8 .....	26

# CAPÍTULO 1

## 1. DELIMITACIÓN DEL PROBLEMA

En este capítulo se desarrollará la delimitación del problema concerniente a la inutilización de la planta Lucas-Nülle IMS 8, los objetivos que se plantearán para el correcto desarrollo del problema, con el fin de automatizar e independizar el proceso de almacenamiento y despacho del IMS 8.

### 1.1 Planteamiento del Problema.

El proceso de almacenamiento y despacho es un punto crítico de control en la industria debido a que, al momento de almacenar la producción muchas veces se lo realiza con un algoritmo de control no tan eficiente lo cual, origina una incorrecta rotación del producto hacia el mercado, provocando que el producto llegue en algunas ocasiones en un mal estado o caducado.

El Laboratorio de Control de Procesos posee en sus instalaciones un sistema moderno compuesto por varias etapas independientes y automáticas para simular procesos industriales reales como: mezclado, llenado, encorchado, descorchado, almacenamiento y despacho. Sin embargo, la estación de almacenamiento y despacho no se ha empleado en el Laboratorio de Control de Procesos debido a que, no se posee un algoritmo de control que permita emular la etapa de almacenamiento y despacho de paquetes de botellas del sistema. Como resultado de esta situación, se está desperdiciando un recurso importante en la formación de los estudiantes de la carrera de Ingeniería en Electrónica y Automatización Industrial.

Adicionalmente, la falta de un algoritmo de control de almacenamiento y despacho genera que no pueda sacar el máximo provecho del sistema, debido a que, el estudiante no puede trabajar con cada etapa independientemente sin completar todo el ciclo de trabajo para el que fue diseñada la planta de manera continua.

## **1.1 Objetivos.**

### **1.1.1 Objetivo General**

Implementar un sistema de automatización industrial de almacenamiento de paquetes de botellas del Laboratorio de Control de Procesos, usando un algoritmo de primero en entrar primero en salir, para el modelado de un manejo mecatrónico de una bodega real.

### **1.1.2 Objetivos Específicos**

- Analizar el manejo de la instrumentación que posee el subsistema de almacenamiento y despacho, para un correcto estudio de la gestión de proceso de la planta.
- Crear un algoritmo de control para almacenamiento y despacho de paquetes de botellas en un PLC.
- Diseñar un protocolo de operación del sistema mecatrónico para su utilización en las prácticas del Laboratorio de Control de Procesos.
- Implementar un SCADA para el manejo de la bodega.
- Elaborar un modelo en 3D de la estación de trabajo en un software de diseño gráfico CAD para que los estudiantes visualicen e identifiquen el sub sistema de la planta en diferentes ángulos previo a la práctica.
- Desarrollar un manual de funcionamiento para el almacenamiento y despacho de paquetes de botellas.
- Redactar un manual del algoritmo de control para un óptimo aprendizaje por parte de los estudiantes del laboratorio.

## **1.2 Justificación**

La mayoría de las industrias poseen bodegas donde almacenan su producción hasta el momento de su despacho, lo cual requiere un adecuado algoritmo de control para no comprometer la integridad de la mercadería y evitar que ésta pueda caducarse o estropearse.

La implementación de un algoritmo de control para el sistema de almacenamiento y despacho de la planta existente en el Laboratorio de Control de Procesos permitirá recuperar un recurso que actualmente está siendo desperdiciado y no cumple con la función académica para la cual fue adquirida.

Adicionalmente, se busca que los estudiantes de la carrera de Electrónica y Automatización puedan complementar sus prácticas de laboratorio con el proceso de almacenado y despacho que se ve actualmente en las industrias.

### **1.3 Alcance del proyecto**

En primer lugar, en este proyecto se realizará el análisis de la instrumentación que posee la planta y la banda transportadora, para lo cual se determinarán las señales entrada o salida utilizadas por los actuadores y sensores incluidos en el sistema.

A continuación, se reconocerán los medios de comunicación existentes entre la planta y el controlador externo PLC que permiten la interacción entre los sensores de la planta la planta y el PLC.

Luego, se procederá al diseño del algoritmo de control primero entrar primero en salir en GRAFCET para el almacenado y despacho de paquetes de botellas, que se implementará utilizando los siguientes lenguajes de programación: Diagrama de funciones secuenciales (SFC) [1], diagrama de bloques de funciones (FBD) [1] y Lenguaje estructurado [2], dependiendo de la complejidad de la función. Además, se realizará un diseño en 3D de la etapa de almacenamiento de la planta en una plataforma de Cad, para poder simular el proceso.

Una vez implementado el algoritmo de control para el almacenamiento y despacho, se realizarán las pruebas necesarias para verificar que el algoritmo no contenga fallas.

Finalmente, se desarrollará un protocolo de operación del sistema mecatrónico para su correcta utilización por parte de los estudiantes también, se desarrollará un manual de funcionamiento para el almacenamiento y despacho de paquetes de botellas en forma de guías de práctica, acompañada con un manual del algoritmo de control.

# CAPÍTULO 2

## 2. Metodología de trabajo

En este capítulo se detallan los métodos utilizados para el desarrollo del proyecto, los cuales implican el análisis y manejo de la instrumentación del subsistema IMS 8, diseño de un sistema SCADA [3], desarrollo de un algoritmo de control para el almacenamiento y despacho de paquetes de botellas; adicionalmente, la elaboración de un modelo 3D de la estación de trabajo.

### 2.1 Planteamiento del proceso.

Para desarrollar el control requerido en la planta de almacenamiento y despacho del IMS 8 en el Laboratorio de Control de Procesos es necesario, el estudio de la gestión de proceso, el análisis la instrumentación y el diseño de un GRAFCET [4] para el sistema usando lenguaje de bloques. A continuación, se realizó un Diagrama de funciones secuenciales (SFC), Diagrama de bloques de funciones y Bloque de Organización (OB), utilizando el software de programación para Plc. La comunicación entre el PLC S7-300 y el subsistema se la realizó a través del módulo esclavo PROFIBUS DP. En la Figura 2.1 se muestra el proceso que se tomó en cuenta para el desarrollo de la solución del problema.

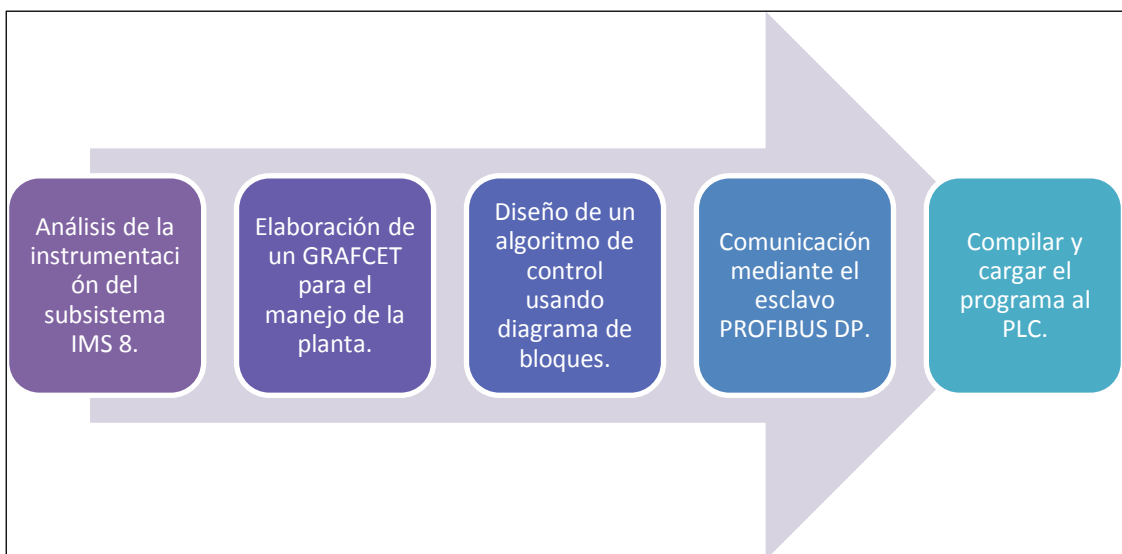


Figura 2.1: Planteamiento del desarrollo de la solución del problema



## 2.2 Identificación de las partes del subsistema IMS 8

La instrumentación del subsistema IMS 8 está compuesto por una gran cantidad de sensores y actuadores tales como: sensores magnéticos, válvulas, motores, relés, etcétera, los cuales como parte de la planta poseen funciones puntuales tales como reconocimiento de los niveles en los ejes XYZ, control del vacío, y movimiento de sus ejes; en el anexo 1 se detallan los sensores y actuadores de la planta con sus respectivas funciones.

## 2.3 Diseño del algoritmo de control en Grafcet

La estrategia de control se basa en segmentar las funciones del sistema en tres bloques fundamentales necesarios para el funcionamiento de la planta, los cuales son: X\_Position, HRSA y STAD. El bloque X\_Position permite determinar las posiciones en las que se colocará o extraerá los paquetes en el eje X. El bloque HRSA (High Rack Store Área) se encarga de la automatización del manejo de paquetes ya sea para despacho o almacenamiento y el bloque STAD (Storage Administrador) se encarga de la gestión de las plazas de almacenamiento y despacho.

El bloque X\_Position cuenta con un sensor de impulsos, el cual trabaja con una relación de 0.5 cm/IMP, con lo que se determinó que cada paquete ocupa un ancho de 5 cm en la base en el eje x (b), se determinó la distancia entre paquetes de 3 cm (d), como se muestra en la Figura 2.2 cuando un nivel está completamente lleno.

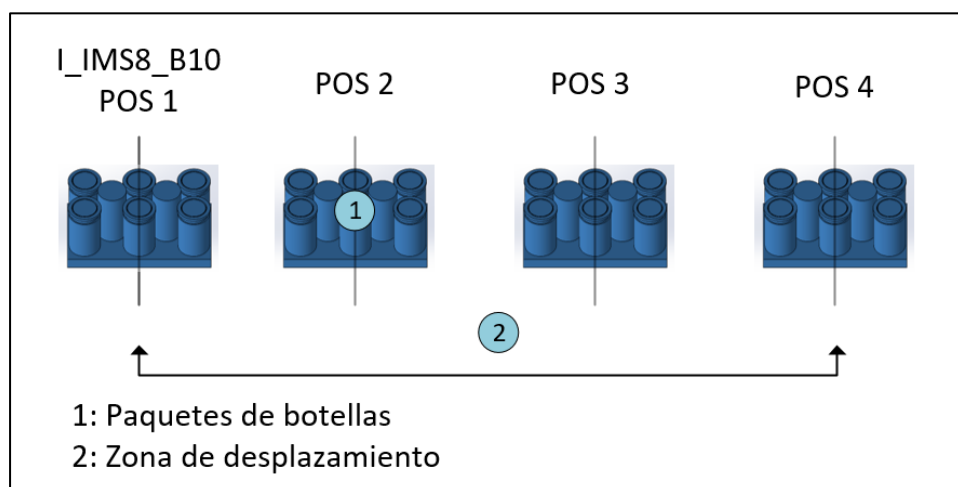
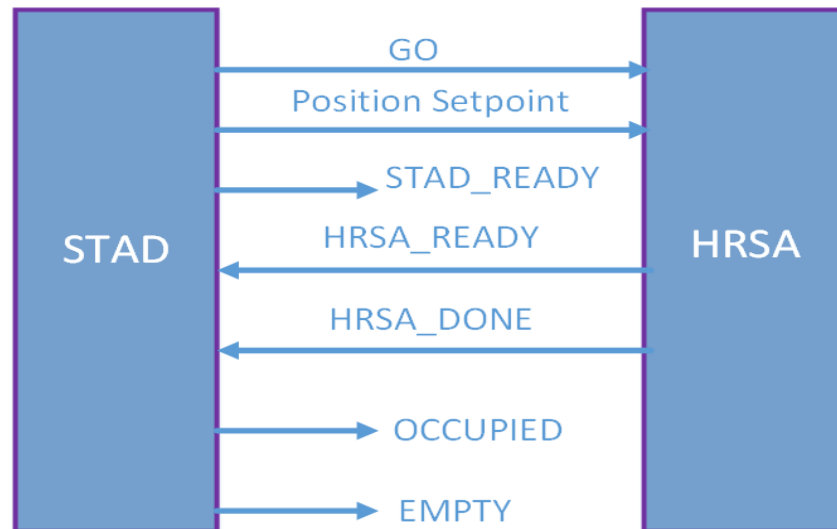


Figura 2.2: Nivel de estantes por filas

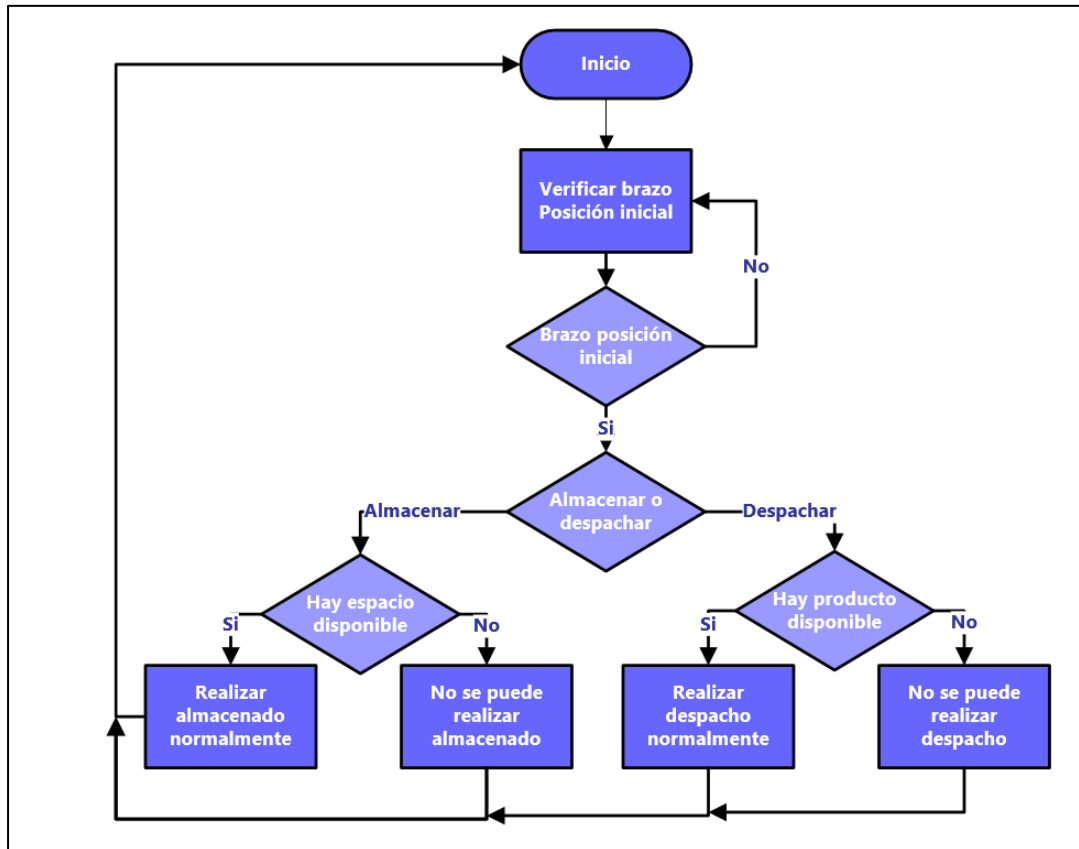
HRSA (High Rack Store Área), contiene los controles de secuencia necesarios para el depósito o retiro de paquetes de botellas. El STAD (Storage Administrador), trabaja en conjunto con el HRSA indicándole las coordenadas de marcha para el manejo de sus actividades. Se mantienen en constante comunicación a lo largo del todo el proceso, uno no puede ejecutar sin la autorización del otro ver Figura 2.3.



**Figura 2.3: Bloque STAD y HRSA**

En la comunicación entre el STAD y el HRSA intervienen una serie de señales previo al proceso, durante la ejecución y posterior a ella, las cuales son: GO, Position Setpoint, STAD\_READY, OCCUPIED, EMPTY, por parte del STAD (encargado de la parte de automatización) y el HRSA\_READY, HRSA\_DONE pertenecientes al HRSA (encargado de la parte manipulación de paquetes).

Para el diseño del sistema de control en lenguaje de bloques se realizó previamente, un bosquejo de las funciones principales del IMS 8, las cuales son despacho y almacenamiento, considerando las condiciones iniciales del brazo mecánico perteneciente a dicha planta ver Figura 2.4.



**Figura 2.4: Bosquejo tareas planta IMS 8**

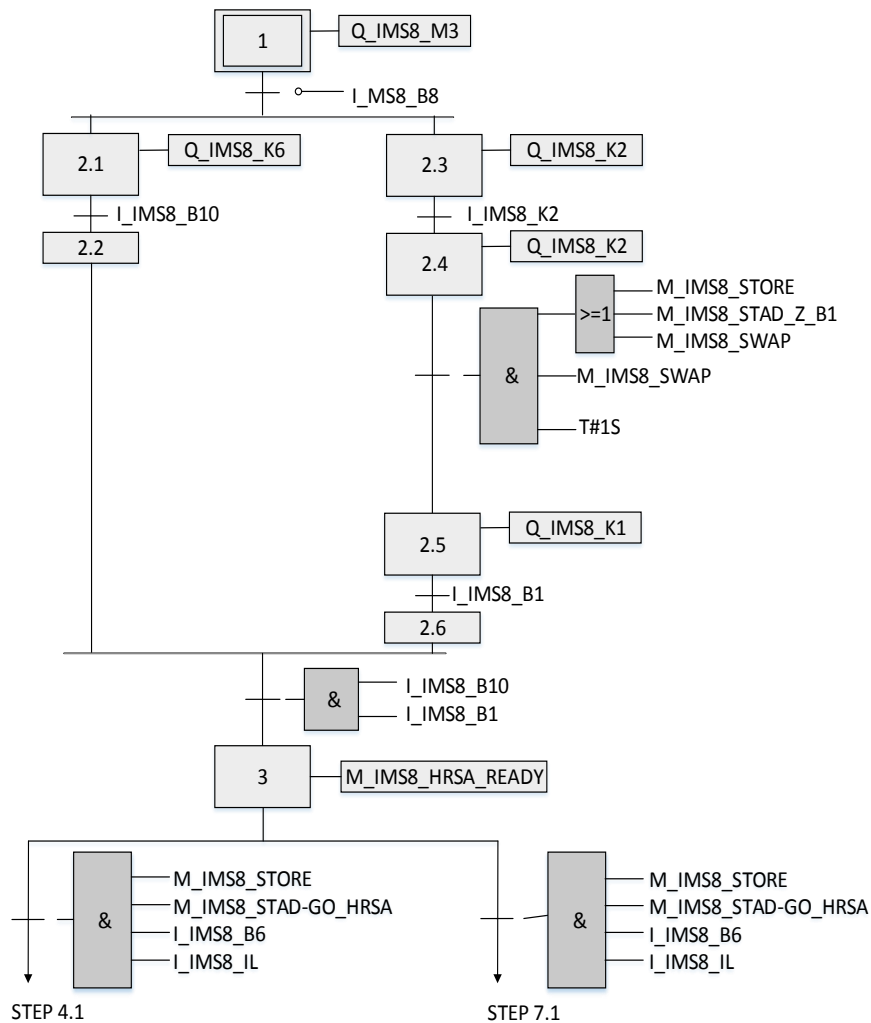
En primera instancia, el brazo mecánico debe estar en posición inicial para dar comienzo al proceso ya sea de despacho o almacenamiento, es necesario tener este punto de referencia porque a partir del mismo se realizarán los movimientos en todos los ejes. Una vez que se tiene el brazo listo en la posición de referencia tenemos dos opciones Almacenar o Despachar, si se almacena se pregunta si hay espacio disponible; si hay, se realiza un almacenado normalmente y se regresa a la condición de inicio, caso contrario no se ejecuta el almacenamiento y se retorna a la posición de inicio, si se elige un despacho se pregunta si hay paquetes disponibles, si hay se realiza un despacho normalmente bajo la condición de que “El primero en entrar sea el primero en salir” y se regresa a la condición de inicio, caso contrario no se efectúa un despacho y se regresa a la posición inicial.

En base a las funciones básicas determinadas anteriormente se realizó el diseño del control secuencial y control convencional secuencial bajo la metodología de GRAFCET ver anexo 2.

En el control convencional secuencial bajo la metodología de GRAFCET se debe establecer todos los estados, resultados y condiciones de transición del proceso

realizado por la IMS 8. A continuación, se detallan las partes de este controlador: Control del posicionamiento inicial del brazo, Control convencional de almacenamiento Figura 2.6, Control convencional del despacho Figura 2.7 y en la Figura 2.8 el control secuencial (HRSA-STAD).

El posicionamiento del brazo es común tanto para la acción de despacho como la de almacenamiento ya que concierne a los pasos puntuales para la localización del brazo mecánico de la planta en el punto de referencia, a partir del cual puede realizar cualquier acción ver Figura 2.5.



**Figura 2.5: Control de posicionamiento inicial del brazo**

Respecto al proceso de almacenamiento, previamente, se debe cumplir las condiciones de transición entre los estados 3 ( ver Figura 2.5 ) y 4.1, para comenzar la inicialización del proceso de almacenado ver Figura 3.6, se parte del estado 4.1 que indica, la activación del motor de la banda a marcha lenta hacia la

derecha (QS y QR), procurando mantener precisión al pasar al siguiente estado, la condición de transición del 4.1 al 4.2 es detectar el pallette por medio del sensor IMS8\_IR, en ese momento se apaga el motor de la banda (QS) y se procede a activar (K2 y K3) los que permiten el movimiento descendente en el eje Z y marcha lenta respectivamente, en el transcurso del movimiento se activa la válvula M2 de 5 a 2 vías que permite el agarre del paquete de botellas, simultáneamente se llega al estado 5.1 y 5.3 en los que se activan las salidas (K1 y K5) los cuales permiten el movimiento ascendente en el eje Z y el movimiento de avance en el eje X correspondientemente. El gestor de almacenamiento STAD proporciona el nivel en el eje Z y el X\_POSITION la ubicación del paquete en los estantes respecto al eje X según la disponibilidad indicada por el STAD. Llegando a los estados 5.2 y 5.4 los cuales cuentan con memoria CACHE1 y CACHE2 respectivamente, en el estado 5.4 se tiene una memoria de retardo debido a la llegada al n nivel en Z y a la posición en X correspondiente, lo que nos lleva al estado 6.1 en el que se activa la señal M1, permitiendo la salida del vástago del cilindro el cual permite el movimiento en el eje Y. Se realizó un retardo de 2 segundos en la posición de llegada del vástago, inmediatamente en el estado 6.2 se produce un ligero movimiento en el eje Z ejecutado por K2, y se activa el cilindro para el retroceso del vástago gracias a M1, la condición de transición al estado 6.3 es un delay de 0.1s, en la salida de este estado se desconecta el tubo de Venturi que permite la generación de vacío gracias a M3, así como el movimiento ascendente del eje Z por activación de K1. En el estado 6.4 se activan dos salidas M3 y HRSA\_DONE lo cual indica que la acción fue ejecutada con éxito, para volver al paso 1 es necesario que el vástago haya retornado indicado por B6 y que el sensor de vacío determinado por B8 esté desactivado.

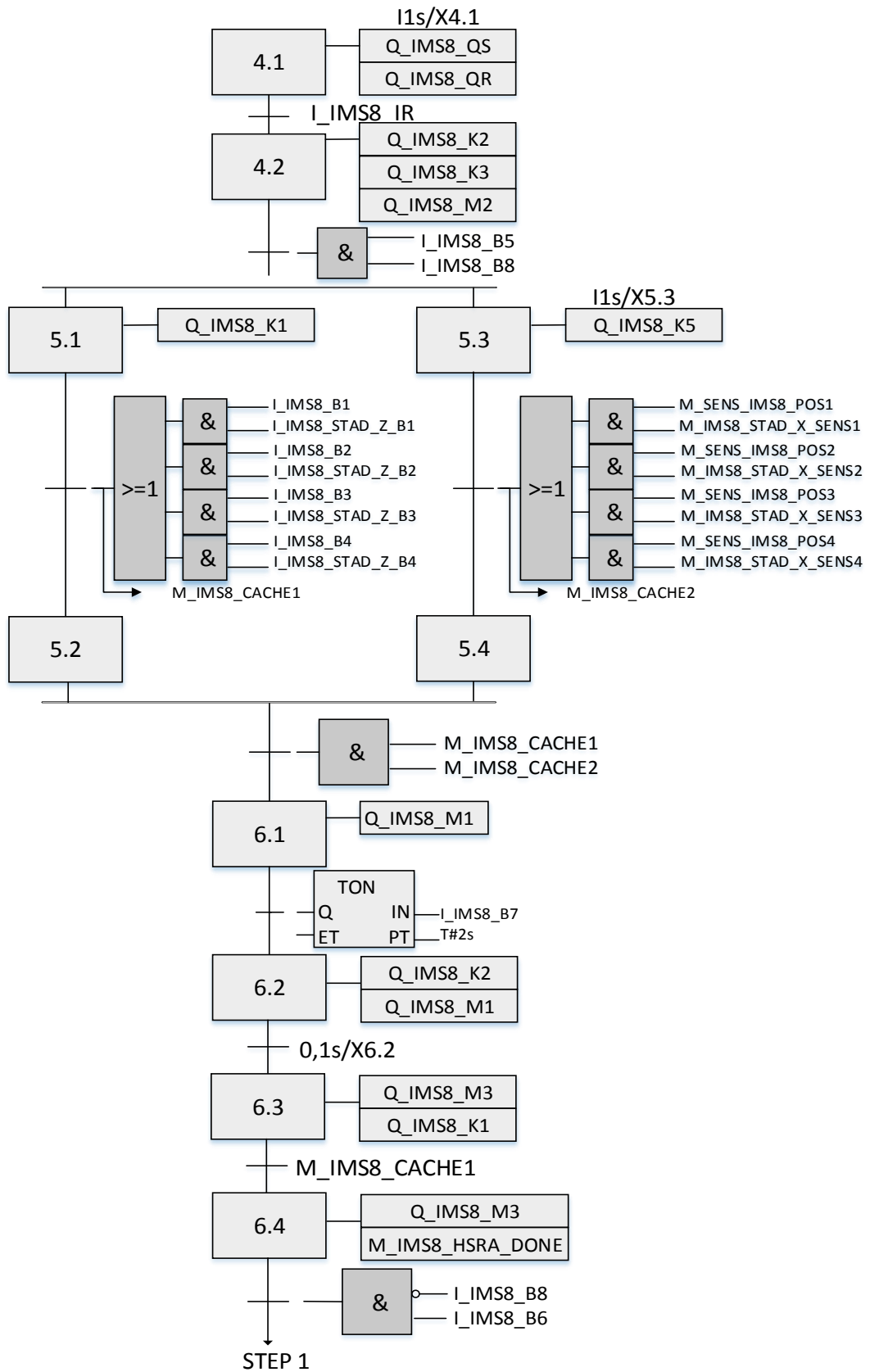
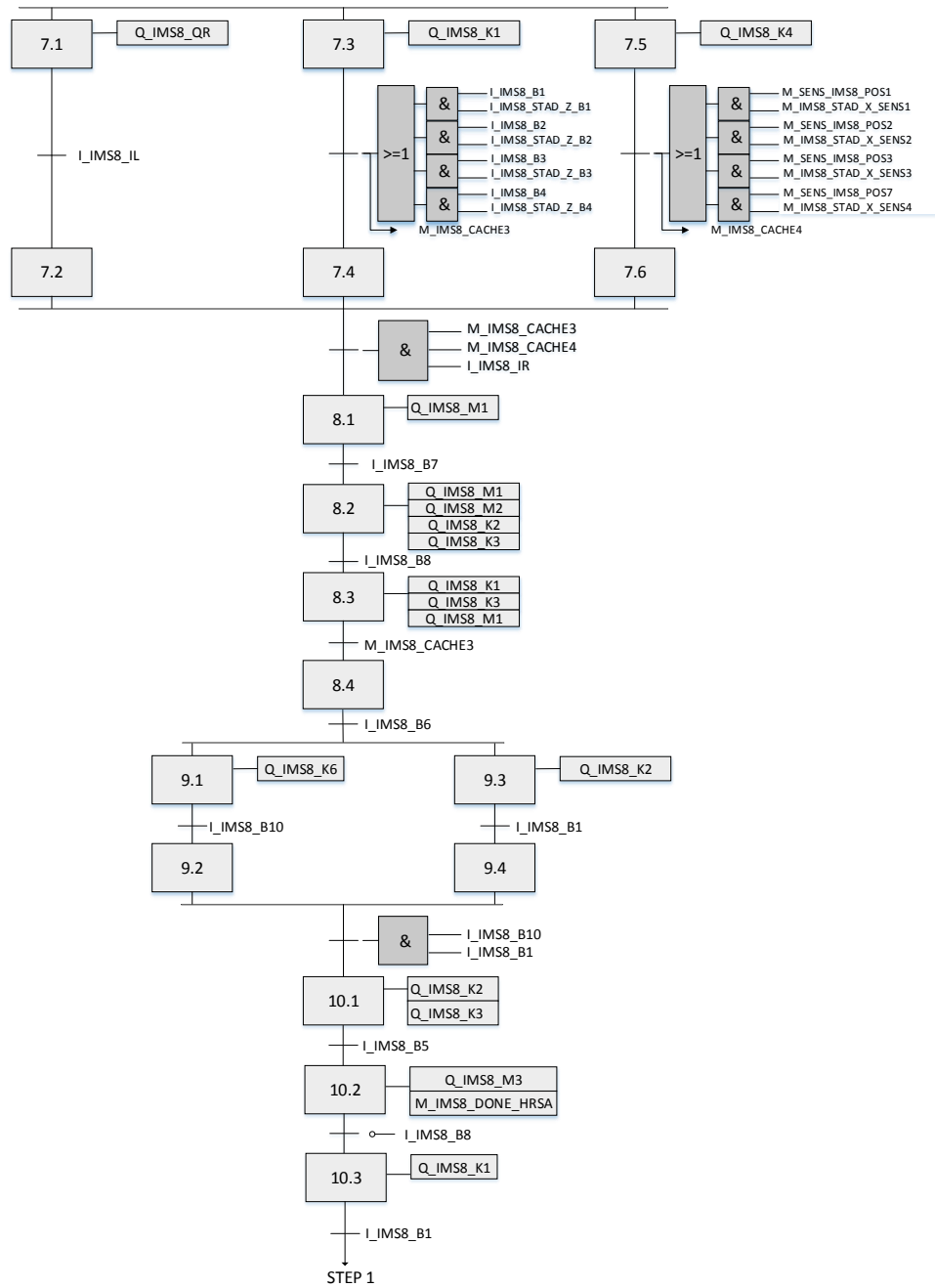


Figura 2.6: Control convencional de almacenamiento

El proceso de despacho es el que se puede ver en la Figura 2.7 se asemeja al proceso de almacenamiento salvo por ciertas variantes tales como: la recepción de un pallet vacío para depositar el despacho, de la misma manera se activa la banda en marcha lenta a la derecha (QS y QR) en el estado 7.1 hasta que el pallet sea detectado por el sensor IR como condición de transición al 7.2, paralelamente tenemos los estados 7.3 y 7.5 los cuales activan K1 (ascenso eje Z) y K5 (avance eje Z), el nivel y posición de donde se extraerá el paquete son determinados por el STAD, el mismo que comprueba que sea el paquete de mayor antigüedad de almacenamiento. Cumpliendo las previas condiciones, se llega a los estados 7.2, 7.4 y 7.6, así como en el almacenamiento se genera una memoria cuando se determina nivel y posición lo mismo pasará para el despacho con CACHE3 y CACHE4, se determina si el pallet está listo para la recepción de la pieza gracias al sensor IR, en el siguiente estado 8.1 se extiende el vástago M1, pasamos al estado 8.2 si el vástago está totalmente extendido determinado por B7, en su salida se activa el sensor de vacío M2 y hay un ligero y lento descenso en el eje Z (K2 y K3). La condición de transición del estado 8.2 a 8.3 es la activación del control de vacío B8, en este estado se tiene un ascenso en marcha lenta en el eje Z (K1 y K3) y un avance en el eje Y (M1). Para pasar del estado 8.3 al 8.4 se tiene como condición la activación de la memoria CACHE3, lo que significa que el eje Z ha vuelto a la posición inicial. La condición para pasar del estado 8.4 al 9.1 es que el eje Y, se haya retraído (B6), una vez retraído se pasa simultáneamente a los estados 9.1 y 9.3 activando consigo las salidas (K2 y K6), que implica el retroceso en el eje X y el descenso en el eje Z respectivamente, hasta llegar a la posición de referencia determinados por los sensores B1 (eje Z) y B10 (eje X), cuando se cumplen las condiciones anteriores se alcanza el estado 10.1 en el que se activa el descenso en el eje Z (K2), con marcha lenta (K3), el sensor B5 establece el fin del descenso en el eje Z condición de transición al estado 10.2, en el cual se desactivan las ventosas (M3) y se tiene la señal de salida DONE\_HRSA que representa el cumplimiento de la orden de despacho, posteriormente tenemos el estado 10.3 el cual verifica que no esté activado el sensor de vacío B8, en este estado se activa el ascenso en el eje Z hasta llegar el nivel inicial determinado por B1, esta condición es necesaria para retornar al estado inicial.



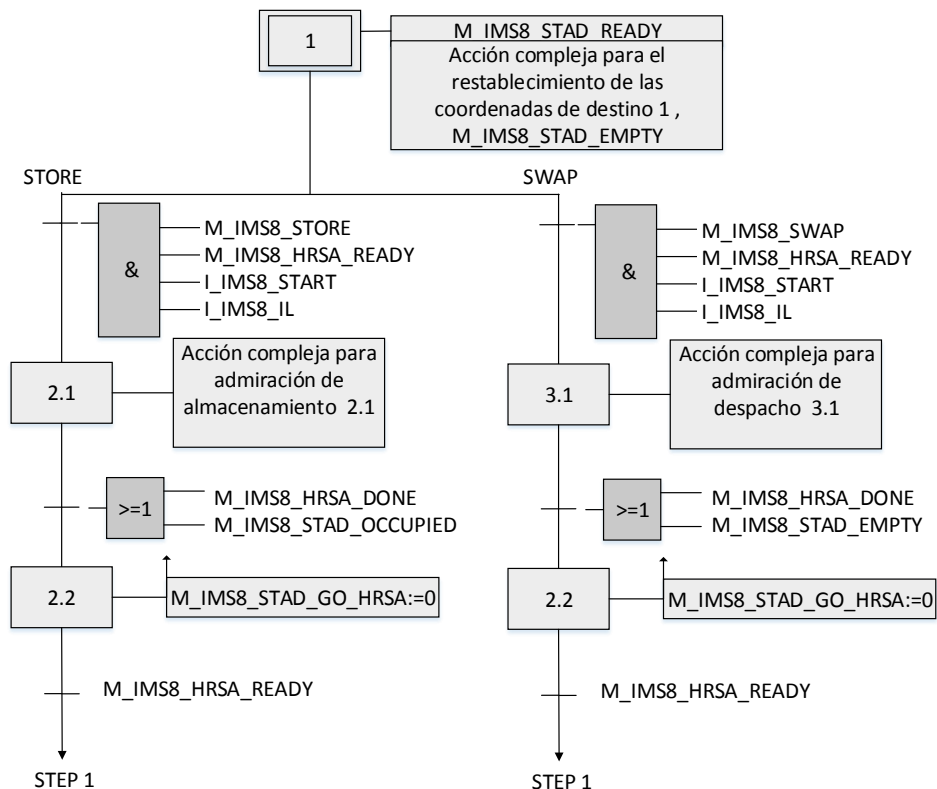
**Figura 2.7: Control convencional del despacho.**

El control secuencial (HRSA-STAD), consta esencialmente de 5 estados de transición ver en la Figura 2.8, el primero tiene como salida de estado la acción de restablecimiento del brazo en las coordenadas de referencia, las tareas de almacenamiento se resumen en los estados 2.1 y 2.2, las condiciones de transición al estado 2.1 son la activación del START, STORE, y la detección del pallette a través del sensor magnético IL, la salida es la ejecución de la acción de



almacenamiento, en el caso del estado 2.2 sus condiciones de transición son la señal HRSA\_DONE la cual indica que el almacenamiento fue exitoso o que los estantes están llenos, la salida de estado es la negación de la señal STAD\_GO\_HRSA lo que significa que se podrá volver a realizar la acción de almacenamiento o despachado según la necesidad.

Las tareas de despacho se resumen en los estados 3.1 y 2.2, las condiciones de transición son similares al de almacenamiento salvo por la acción de SWAP, propios de la acción de despacho.



**Figura 2.8: Control secuencial (HRSA-STAD)**

## 2.4 Implementación del algoritmo

Para la implementación del algoritmo de control se utilizó el lenguaje de programación de bloques entre ellos: El diagrama de funciones secuenciales (SFC), Diagrama de bloques de funciones (FBD) y Bloque de Organización (OB).

### 2.4.1 Descripción del bloque de funciones secuenciales (SFC).

En la Figura 2.9 se muestra el bloque de funciones SFC que se utilizó para la implementación de las etapas de transiciones de los estados (SFC\_IMS8\_HRSA) y las instrucciones de salidas de los estados

(SFC\_INSTRUCTION), este bloque permite crear funciones lógicas sin memoria lo cual permitió realizar correctamente las instrucciones de transiciones y de salidas, ambos bloques fueron programados en lenguaje de bloque FUP.

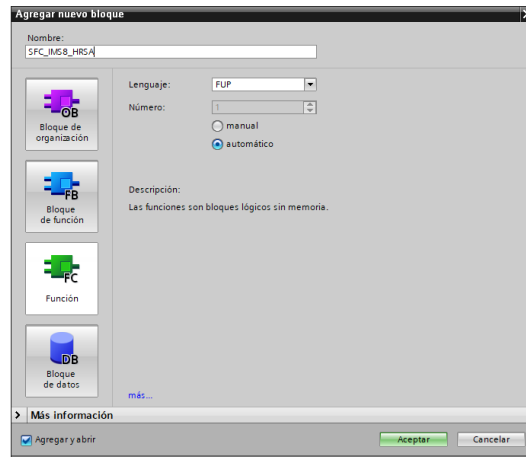


Figura 2.9: Bloque de funciones SFC.

#### 2.4.2 Descripción del bloque de funciones (FBD).

En la Figura 2.10 se observa el bloque de funciones FBD, este bloque pueden almacenar sus valores de forma permanente en los bloques de datos de instancia DB que se crearán después del bloque de funciones FB, de tal manera que siguen estando disponible después de procesar el bloque. Se implementó estos bloques especialmente para la parte del STAD\_HSRA y X\_Position que necesitan retener información para después tomarla en consideración, los bloques fueron programados en lenguaje de bloque FUP y estructurado.

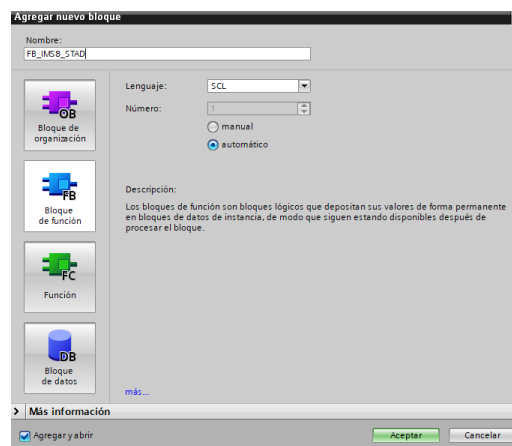
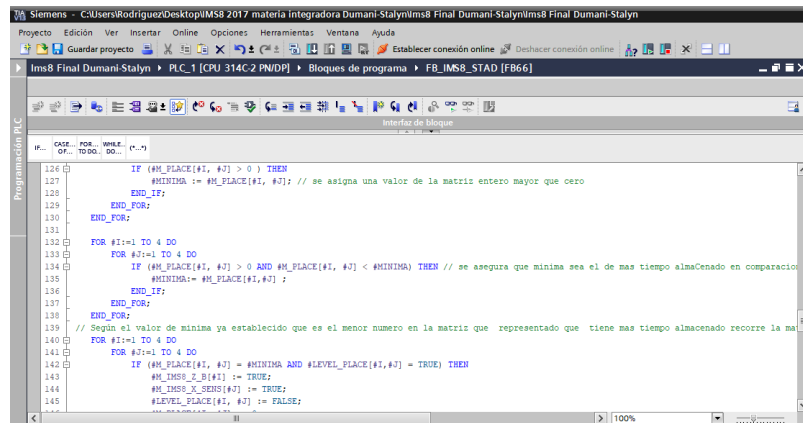


Figura 2.10: Bloques de funciones FBD

El bloque del STAD se programó en lenguaje de bloques estructurado, debido a que facilitaba el manejo de algunas de sus funciones en la programación tales como el manejo de matrices.

Adicionalmente, en este bloque se programó la parte fundamental del algoritmo de control primero entrar primero en salir, esto se implementó usando dos matrices: una con valores booleanos inicialmente FALSO (FALSE) y otra de enteros inicialmente igual a 0. A cada posición de la matriz se le asignaba un número entero mayor que 0 en forma creciente en el momento de almacenar. En la parte de despacho se tomaba el menor valor diferente de 0 mediante una función integrada en la programación; así, se pudo identificar que ubicación de la matriz tenía más tiempo que la otra para luego, proceder a despachar esa posición y asignarle a esa ubicación de la matriz el valor entero de 0.

En la Figura 2.11 se muestra parte de la programación del algoritmo de control primero en entrar primero en salir usando matrices, para ver todo el código de la programación implementada ver en el anexo 3.



```
Siemens - C:\Users\Rodriguez\Desktop\IMS8 2017 materia Integradora Dumani-Stalyn\IMS8 Final Dumani-Stalyn\IMS8 Final Dumani-Stalyn
Proyecto Edición Ver Insertar Online Opciones Herramientas Ventana Ayuda
IMS8 Final Dumani-Stalyn - PLC_1 [CPU 314C-2 PN/DP] Bloques de programa - FB_IMS8_STAD [F866]
Interfaz de bloque
Programación STL
#... CASE_FOR... WHILE... (*...*)
OF... TO DO... DO... (*...*)
126 IF (#M_PLACE[#I, #J] > 0) THEN
127   #MINIMA := #M_PLACE[#I, #J]; // se asigna un valor de la matriz entero mayor que cero
128 END_IF;
129 END_FOR;
130 END_FOR;
131
132 FOR #I:=1 TO 4 DO
133   FOR #J:=1 TO 4 DO
134     IF (#M_PLACE[#I, #J] > 0 AND #M_PLACE[#I, #J] < #MINIMA) THEN // se asegura que minima sea el de mas tiempo almacenado en comparacion
135       #MINIMA:= #M_PLACE[#I,#J];
136     END_IF;
137   END_FOR;
138 END_FOR;
139 // Según el valor de minima ya establecido que es el menor numero en la matriz que representado que tiene mas tiempo almacenado recorre la ma
140 FOR #I:=1 TO 4 DO
141   FOR #J:=1 TO 4 DO
142     IF (#M_PLACE[#I, #J] = #MINIMA AND #LEVEL_PLACE[#I,#J] = TRUE) THEN
143       #M_IMS8_E[#I] := TRUE;
144       #M_IMS8_X_SINIS[#J] := TRUE;
145       #LEVEL_PLACE[#I, #J] := FALSE;
146     END_IF;
147   END_FOR;
148 END_FOR;
```

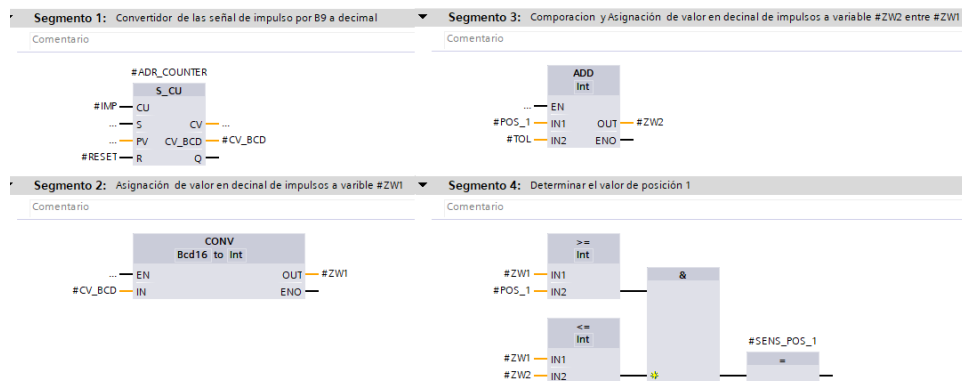
**Figura 2.11: Parte del algoritmo de control**

En la Figura 2.12 se muestran los datos almacenados del bloque de funciones FB\_IMS8\_STAD que estarán disponible y visible para luego, volver a utilizarlos.

Nombre	Tip...	Remanen...	Visible en...	Valor de...	Comentario
Static					
MINIMA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Variable en la que se registra en mínimo valor al recorrer la matriz de enteros
CONT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Contador que empieza en cero
M_PLACE_OLD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de enteros registra lo que esta en los estantes según filas y columnas.
M_PLACE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de enteros registra lo que esta en los estantes según filas y columnas.
LEVEL_PLACE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de booleanos registra lo que esta en los estantes según filas y columnas.
LEVEL_PLACE_OLD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de booleanos registra lo que esta en los estantes según filas y columnas.
M_IMB_X_SENS		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de asignación según el bloque x_position según el sensor a lo largo del eje x para despacho o almac
M_IMB_Z_B		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Matriz de asignación según el nivel a lo largo del eje z; según el bloque STD a lo largo del eje z para un alm
STEP1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Estado 1
STEP2_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Estado 2_1
STEP2_2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Estado 2_2
STEP3_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Estado 3_1
STEP3_2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Estado 3_2

**Figura 2.12: Bloque DB del FB\_IMS8\_STAD**

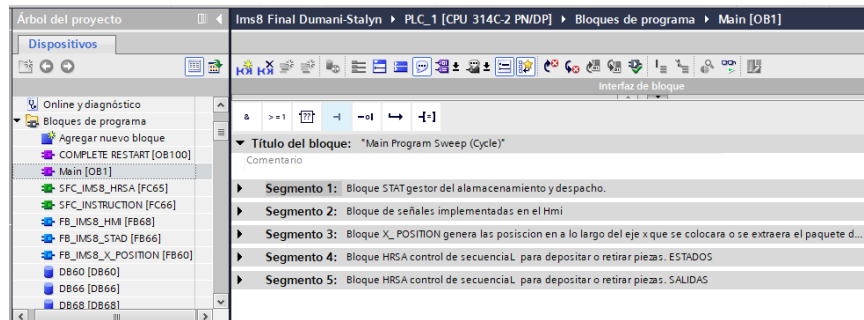
Seguidamente, el bloque X\_Position fue programado en lenguaje de bloques FUP, que permitió dimensionar los espacios en el cual fueron colocados y extraídos los paquetes de botellas en el eje x; adicionalmente, permite la conversión de impulso por el sensor B9 a decimal, en la Figura 2.13 observamos cómo se determinó la posición 1 mediante los bloques.



**Figura 2.13: Bloque FB FB\_IMS8\_X\_POSITION.**

### 2.4.3 Descripción del bloque principal MAIN.

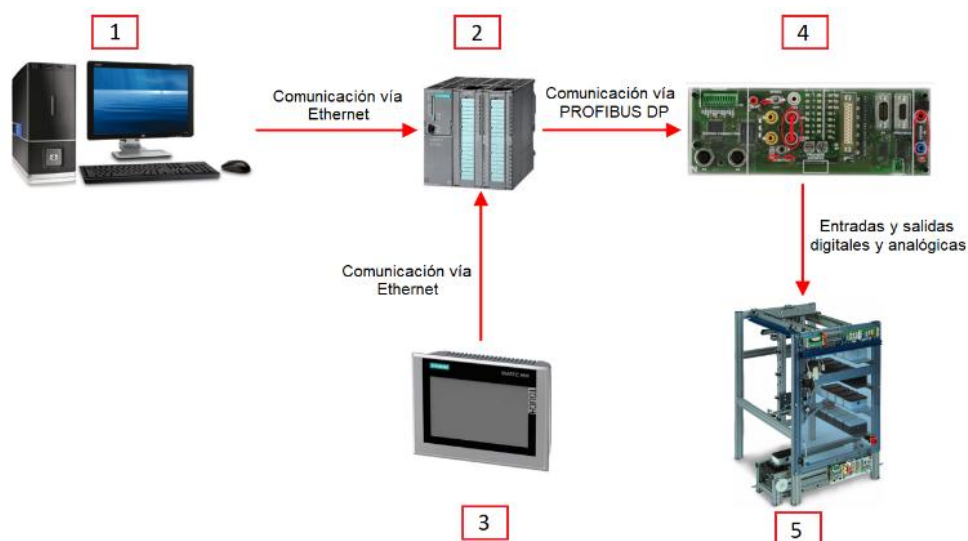
Finalmente, después de programar todos los bloques, se los coloca en el bloque principal MAIN que se encarga de llamar y ejecutar todos los bloques de la programación ejecutando cada etapa de la programación. En la Figura 2.14, se muestra el bloque principal MAIN con los segmentos que contiene todos los bloques de la programación.



**Figura 2.14: Segmentos del MAIN principal**

## 2.5 Comunicación general del subsistema IMS8.

La tarjeta de módulo PROFIBUS DP SLAVE recibe todas las entradas y salidas tanto digitales como analógicas del sub sistema IMS8 y establece la comunicación entre el esclavo PROFIBUS DP, el PLC S7-300 mediante el protocolo de comunicación PROFIBUS DP. Además, la comunicación entre el ordenador PC, el PLC y el Panel táctil HMI TP700 mediante el protocolo de comunicación Ethernet, en la Figura 2.15 se observa una representación de la arquitectura de comunicación detallada anteriormente.

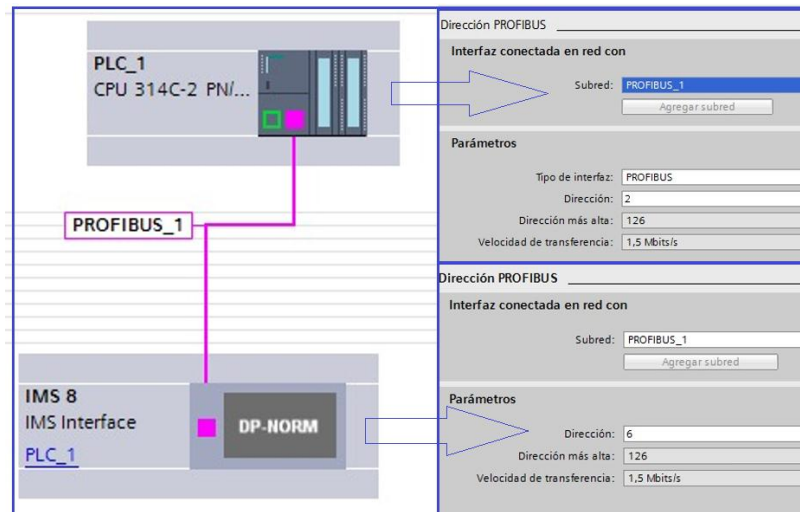


**Figura 2.15: Tipo de comunicación entre dispositivos**

1. Ordenador PC de escritorio.
2. Control lógico programable (PLC) SIMATIC S7-300.
3. Panel táctil HMI TP700.
4. Esclavo PROFIBUS DP SLAVE.
5. Sub sistema IMS 8.

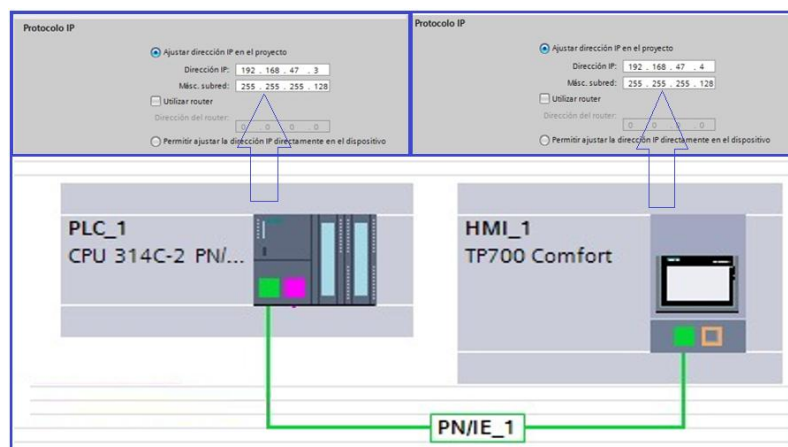
### 2.5.1 Comunicación PLC S7-300

Inicialmente para la comunicación entre el PLC S7-300 y el esclavo PROFIBUS DP SLAVE se realizó una conexión maestro – esclavo, mediante el protocolo de comunicación Profibus DP asignando al maestro con una dirección superior que la del esclavo, en la imagen 2.16 se muestra las direcciones asignadas entre maestro – esclavo.



**Figura 2.16: Comunicación Profibus entre Plc y esclavo**

Luego, se procedió a realizar la comunicación entre el PLC S7-300 y el Panel táctil HMI TP700 mediante el protocolo de comunicación Ethernet, previamente, al establecer la comunicación se asignaron las direcciones IP y máscara de sub redes para los dispositivos como se observa en la Figura 2.17.

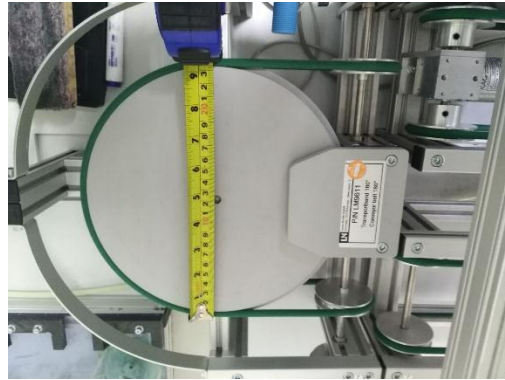


**Figura 2.17: Comunicación Ethernet entre Plc y HMI**

## 2.6 Elaboración de la planta IMS 8 en CAD.

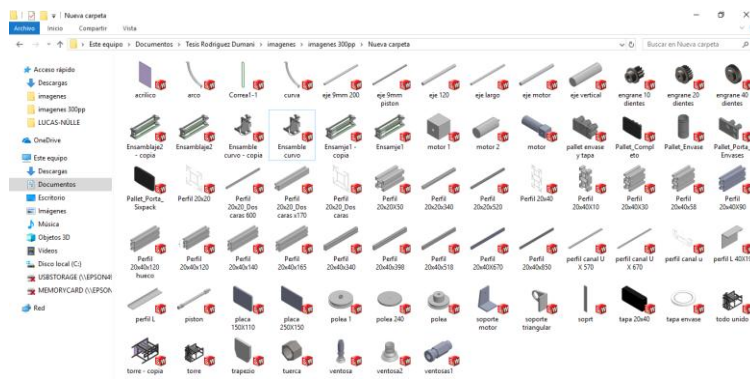
### 2.6.1 Diseño de las piezas de la planta.

Inicialmente para diseñar una pieza se debió realizar mediciones da la planta para obtener medidas reales, en la Figura 2.18 se observa la manera de cómo se procedió a tomar las diferentes medidas por parte de la planta.



**Figura 2.18: Mediciones tomada para el diseño de la planta.**

Luego, se procedió a crear cada pieza de la planta en un archivo diferente para tener un manejo eficaz al momento de armar toda la planta. En la Figura 2.19 se observa todas las piezas creadas de la planta por separadas, tanto perfiles como motores.



**Figura 2.19: Creación de piezas**

### 2.6.2 Diseño del ensamble de las piezas de la planta.

Después de haber diseñado todas las piezas de la planta se creó un nuevo proyecto en el cual se agregó todas las piezas previamente, diseñadas para luego, proceder a realizar el ensamble total de la planta. En la Figura 2.20 se observa el ensamble total de la planta.

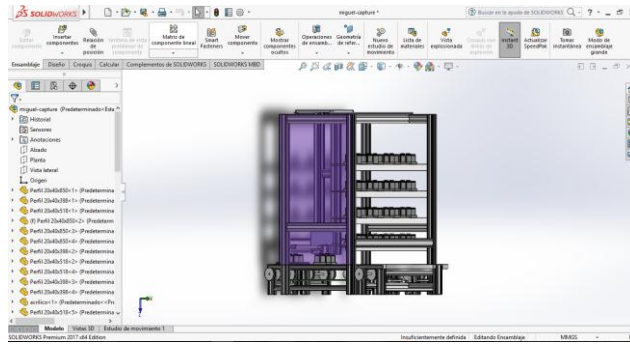


Figura 2.20: Ensamblajes de las piezas de la planta



# CAPÍTULO 3

## 3. RESULTADOS.

En este capítulo se presentan los resultados obtenidos a lo largo del proyecto. De igual manera, la solución de los problemas planteados previamente, en el capítulo 1. Con el propósito de presentar los resultados planteados en el alcance del proyecto de una manera más abreviada, se lo dividió en diferentes puntos: comunicación de los equipos que forman parte del IMS 8, control secuencial del algoritmo de control primero en entrar primero en salir, SCADA del proceso y el diseño en 3D del sub sistema.

### 3.1 Comunicación de dispositivos.

Se logró establecer una comunicación exitosa entre el PLC S7-300 con el Esclavo PROFIBUS DP SLAVE y el HMI TP-700, mediante los protocolos de comunicación Profibus y Ethernet respectivamente, se lo pudo corroborar al momento de visualizarlo en el ordenador pc, como se muestra en la Figura 3.1 donde la planta no mostró errores de comunicación después de haber realizado varias pruebas.

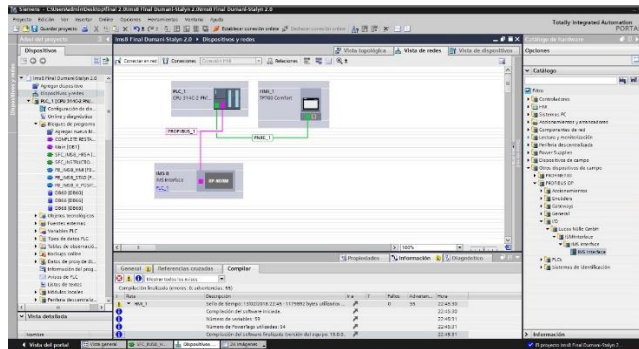


Figura 3.1: Comunicación efectuada entre dispositivos

### 3.2 Control secuencial.

Se logró compilar y cargar exitosamente el algoritmo implementado y se corroboró que no contenía fallas ver Figura 3.2 y 3.3

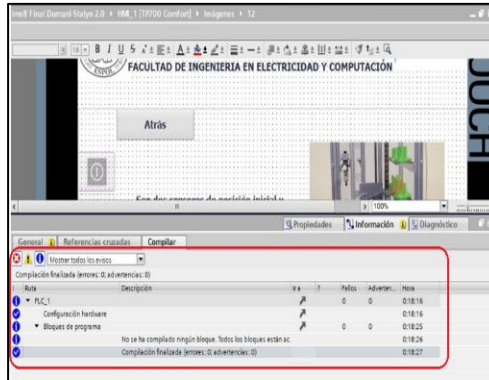


Figura 3.2: Compilación efectuada con éxito

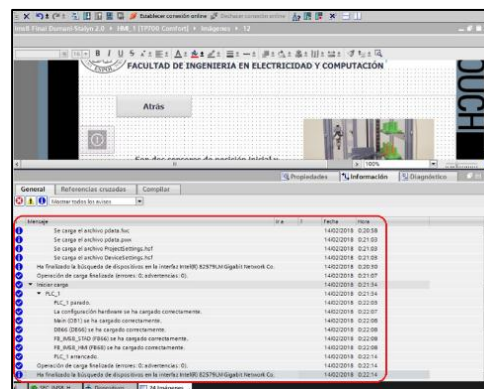


Figura 3.3: Carga del programa al PLC

El IMS 8 respondió correctamente al momento de ejecutar las diversas órdenes previamente, programadas. Al momento de hacer los movimientos los realizó exitosamente logrando el objetivo planteado del algoritmo de control primero en entrar primero en salir.

En la Figura 3.4 se puede observar cómo se realiza el llenado de las 2 matrices para asignar lugares con diferentes antigüedades.

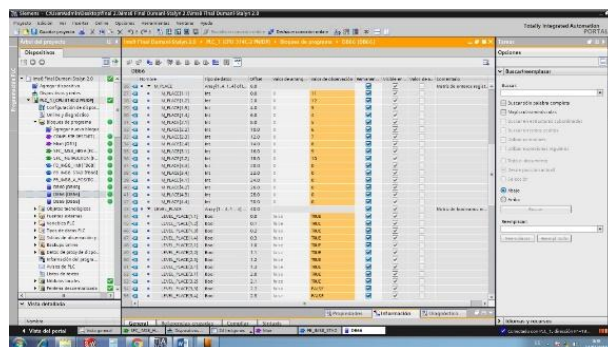
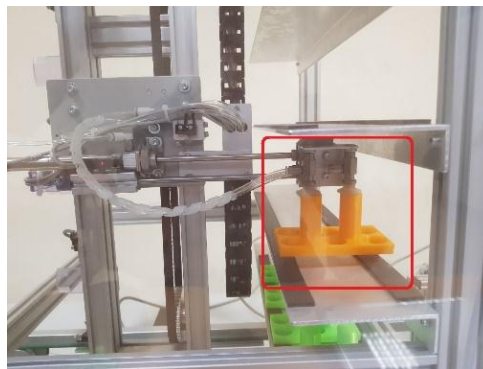


Figura 3.4: Visualización de entradas/salidas en línea

A continuación, en la Figura 3.5 se observa como el brazo neumático ejecuta un movimiento para tomar la pieza luego, en la Figura 3.6 se aprecia cuando se coloca un paquete de botella en unas de los 16 lugares disponibles en el estante elevado.

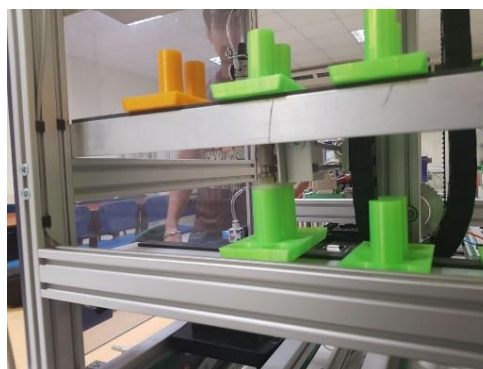


**Figura 3.5: Movimiento inicial del brazo para almacenamiento**

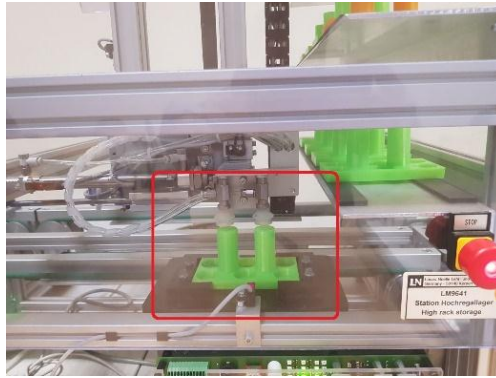


**Figura 3.6: Movimiento final del brazo para almacenamiento**

Posteriormente, en la Figura 3.7 se observa como el brazo neumático ejecuta un movimiento, para retirar uno de los paquetes de botellas en el estante luego, en la Figura 3.8 se muestra cuando depositada un paquete de botella en el pallette para su despacho final.



**Figura 3.7: Movimiento inicial del brazo para despacho**

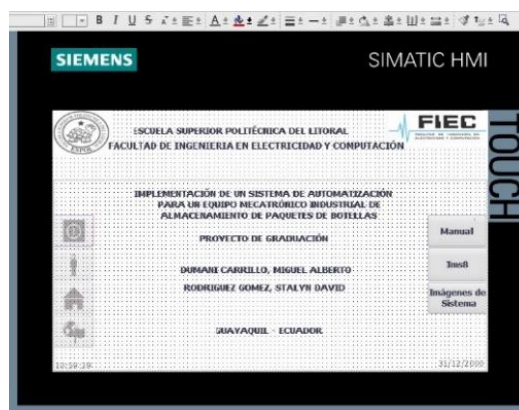


**Figura 3.8: Movimiento final del brazo para despacho**

### 3.3 Sistema Scada del proceso.

Al realizar Scada del proceso se pudo vincular las variables del proceso. Adicionalmente, obtuvimos una mejor interfaz entre el usuario y la planta, ya que se pudo ejecutar ordenes mediante opciones que presento el HMI, también, se observó de una manera detallada resultados en tiempo real de la planta al momento de realizar un almacenado o un despacho.

En la Figura 3.9 se ve como luce el menú de interfaz con el usuario, también, se puede observar en la Figura 3.10 una interfaz que especifica los sensores y actuadores que destacan en la planta. Finalmente, en la Figura 3.11 se visualiza el IMS y 8 todos los respectivos paquetes de botellas en el estante.



**Figura 3.9: Menú principal del HMI**

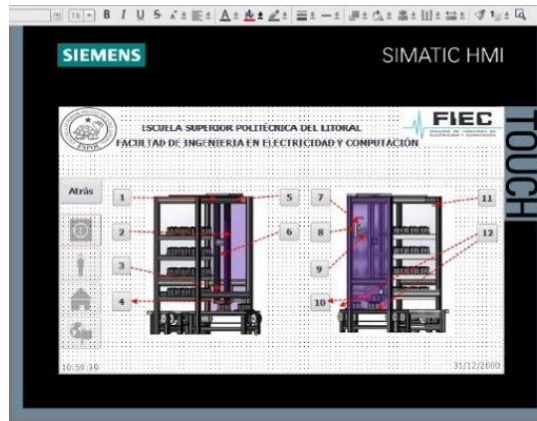


Figura 3.10: Sensores y actuadores del IMS 8

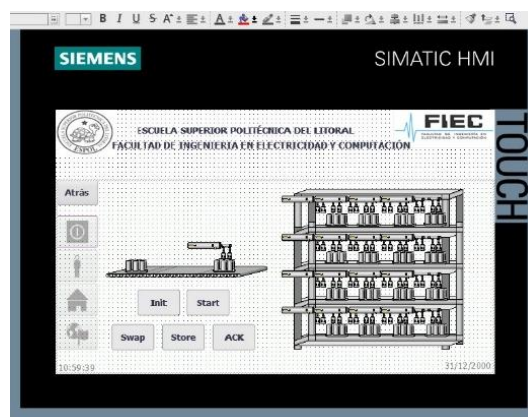


Figura 3.11: Bosquejo del estante elevado.

### 3.4 Modelado 3D del subsistema IMS 8.

El modelado fue realizado exitosamente incluyendo una simulación de demostración, se realizó con medidas reales de la planta para tener un modelado y proceso más real como se aprecia en la Figura 3.12; posteriormente, en la Figura 3.13 se observa el diseño total de la planta.

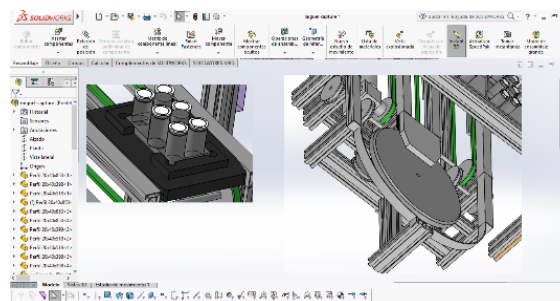
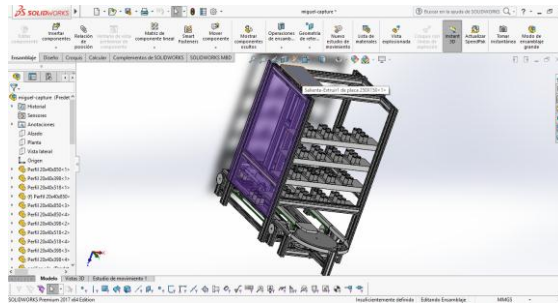


Figura 3.12: Ensamblajes realizados detalladamente.



**Figura 3.13: Diseño total del sub sistema IMS 8**

## CONCLUSIONES Y RECOMENDACIONES

La elaboración de un Graficet el cual detallé de una manera simplificada el proceso de la planta, facilitó la visualización de los procesos, etapas y salidas que poseía la planta, para luego, proceder con la respectiva programación por parte del usuario.

Fue necesario implementar un algoritmo de control de bodega, para realizar almacenamientos y despachos eficientes, debido a que el paquete de botella que ingresaba en primera instancia era el primero que salía al momento de ejecutar un despacho, lo que originaba una correcta rotación de la mercadería por parte de la bodega hacia el mercado.

Al elaborar la programación en diferentes lenguajes de programación como: lenguaje estructurado, KUP y FBD se aprovecha al máximo las fortalezas de cada uno facilitando de manera considerable la programación, ya que en algunos bloques fue necesario implementarlos en un lenguaje diferente para una correcta ejecución del programa.

Al integrar una programación que permita ocupar los espacios dejados inicialmente por los despachos, el uso del brazo mecatrónico de la planta era de un menor tiempo lo que demanda un menor uso de energía por parte de la planta, debido a que se aprovechaba los primeros espacios al momento de realizar un almacenado.

Se recomienda que para futuros proyectos se pueda realizar un estudio, de cómo se podría llegar a realizar un almacenado en una bodega general, el cual receptes diferentes productos de una misma línea para luego, proceder a clasificar el producto de acorde lo requerido, y almacenarlo en diferente área de la bodega. Un ejemplo claro sería el almacenado de colas dependiendo su sabor esto se lograría usando el color de la bebidas mediante una cámara.

Se recomienda implementar un algoritmo que permita limpiar la bodega con los paquetes de botellas con más antigüedad, para prevenir que caduquen en el transcurso del tiempo hasta que lleguen al mercado.

## BIBLIOGRAFÍA

- [1] Automation Academy de B&R Disponible en: <https://www.br-automation.com/es/formacion/automation-academy-de-br/>
- [2] Ecured conocimiento para todos Disponible en: [https://www.ecured.cu/Programacion\\_estructurada](https://www.ecured.cu/Programacion_estructurada)
- [3] Introducción a SCADA por Carlos de Castro Lozano Cristóbal Romero Morales Disponible en: <http://www.uco.es/grupos/eatco/automatica/ihm/descargar/scada.pdf>
- [4] Grafcet uso disponible en: <https://es.slideshare.net/techonmyweb/grafcet-13496276>
- [5] Lucas-Nülle - Excellence in Technology and Training Solution Disponible en: <https://www.lucas-nuelle.es/>
- [6] Lucas-Nülle LabSoft. (2016). IPA 26 IMS 8: Estacion de almacenamietno disponible en: [Software Lucas-Nülle LabSoft](#)
- [7] Lucas-Nülle. Planta de almacenamiento IMS 8. Disponible en : <https://www.lucas-nuelle.es/2273/apg/7561/IMS-8-Almacenamiento-.htm>
- [8] Lucas-Nülle. PROFIBUS DP Slave. Disponible en: <https://www.lucas-nuelle.es/2281/pid/15384/apg/8022/PROFIBUS-DP-Slave-----.htm>
- [9] Lucas-Nülle. Placa portadora de trabajo. Disponible en: <https://www.lucas-nuelle.es/2274/pid/14090/apg/7562/Placa-portadora-de-piezas-de-trabajo----.htm>
- [10] Lucas-Nülle. Segmento de cinta transportadora doble de 24V. Disponible en: <https://www.lucas-nuelle.es/2274/pid/2190/apg/1468/Segmento-de-cinta-transportadora-doble-de-24V---.htm>
- [11] Siemens. SIMATIC S7-300]. Disponible en: [https://w5.siemens.com/spain/web/es/industry/automatizacion/sce\\_educacion/soluciones/Documents/314%20SCE%20-%20CF%20-%20S7300.pdf](https://w5.siemens.com/spain/web/es/industry/automatizacion/sce_educacion/soluciones/Documents/314%20SCE%20-%20CF%20-%20S7300.pdf)



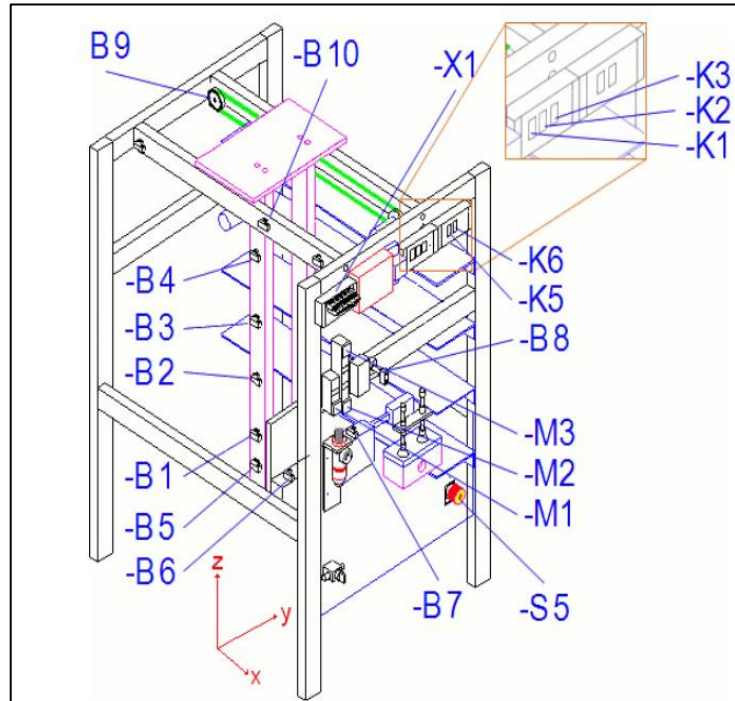
[12] Siemens. TIA PORTAL [Online]. Disponible en:  
[http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Tia-Portal/tia\\_portal/Pages/TIA-Portal.aspx](http://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Tia-Portal/tia_portal/Pages/TIA-Portal.aspx)

[13] 3DCadPortal. SolidWorks [Online]. Disponible en:  
<http://www.3dcadportal.com/solid-works.html>

[14] Método de inventario FIFO por Joannes Vermorel (primero en entrar, primero en salir) Disponible en: <https://www.lokad.com/es/metodo-de-inventario-fifo>

## ANEXOS

### ANEXO 1. Descripción del almacén de estantes elevados



**Figura: Bosquejo del almacén de estantes.**

Sensores	Funciones
<b>-B1, -B2, -B3, -B4</b>	Sensores magnéticos para reconocimiento de los niveles de almacenamiento en el eje Z.
<b>-B5</b>	Sensor magnético para reconocimiento de la posición de retiro y entrega de pieza en el eje Z.
<b>-B6, -B7</b>	Sensores magnético para reconocimiento de la posición de avance o retroceso del eje Y.
<b>-B8</b>	Sensor magnético para control de vacío.
<b>-B9</b>	Sensor magnético para medición de impulsos del eje X.
<b>-B10</b>	Sensor magnético para reconocimiento de la posición referencial del eje X.

**Tabla 1: Sensores de la IMS 8 y sus funciones**

<b>Actuadores</b>	<b>Funciones</b>
<b>-M2</b>	Válvula de 5 a 2 vías. Salida para activar la formación de vacío (tubo de Venturi).
<b>-M3</b>	Válvula de 5 a 2 vías. Salida para desconexión de vacío (tubo de Venturi).
<b>-K1</b>	Motor para el movimiento ascendente del eje Z
<b>-K2</b>	Motor para el movimiento descendente del eje Z.
<b>-K3</b>	Motor para la marcha lenta del eje Z en ambas direcciones.
<b>-K5</b>	Motor para el movimiento de retroceso del eje X.
<b>-K6</b>	Motor para el movimiento de avance (esto es, hacia B10) del eje X.
<b>Tubo de Venturi</b>	Para generación de vacío (activado por medio de la válvula de 5 a 2 vías -- M2,-M3).
<b>2 micro interruptores (-B11, -B12)</b>	Para desconexión de emergencia del eje Z en la posición mecánica final
<b>2 micro interruptores (-B16, -B17)</b>	Para desconexión de emergencia del eje X en la posición mecánica final.
<b>1 micro interruptor - B13 (llegada a un nivel)</b>	Para habilitación de la excitación del eje Z. Éste se conecta en paralelo al micro interruptor -B14 (eje Y en posición de reposo).
<b>1 micro interruptor - B15 (llegada a un nivel)</b>	Para habilitación de la excitación del eje Y
<b>1 relé (-K4)</b>	Para habilitación de la alimentación de corriente del accionamiento del eje Z por medio de los contactos normalmente abiertos -B13, -B14.
<b>-S5</b>	Interruptor de desconexión de emergencia (contacto normalmente abierto).

**Tabla 2: Actuadores de la IMS 8 y sus funciones**

## ANEXO 2. Control secuencial Grafcet.

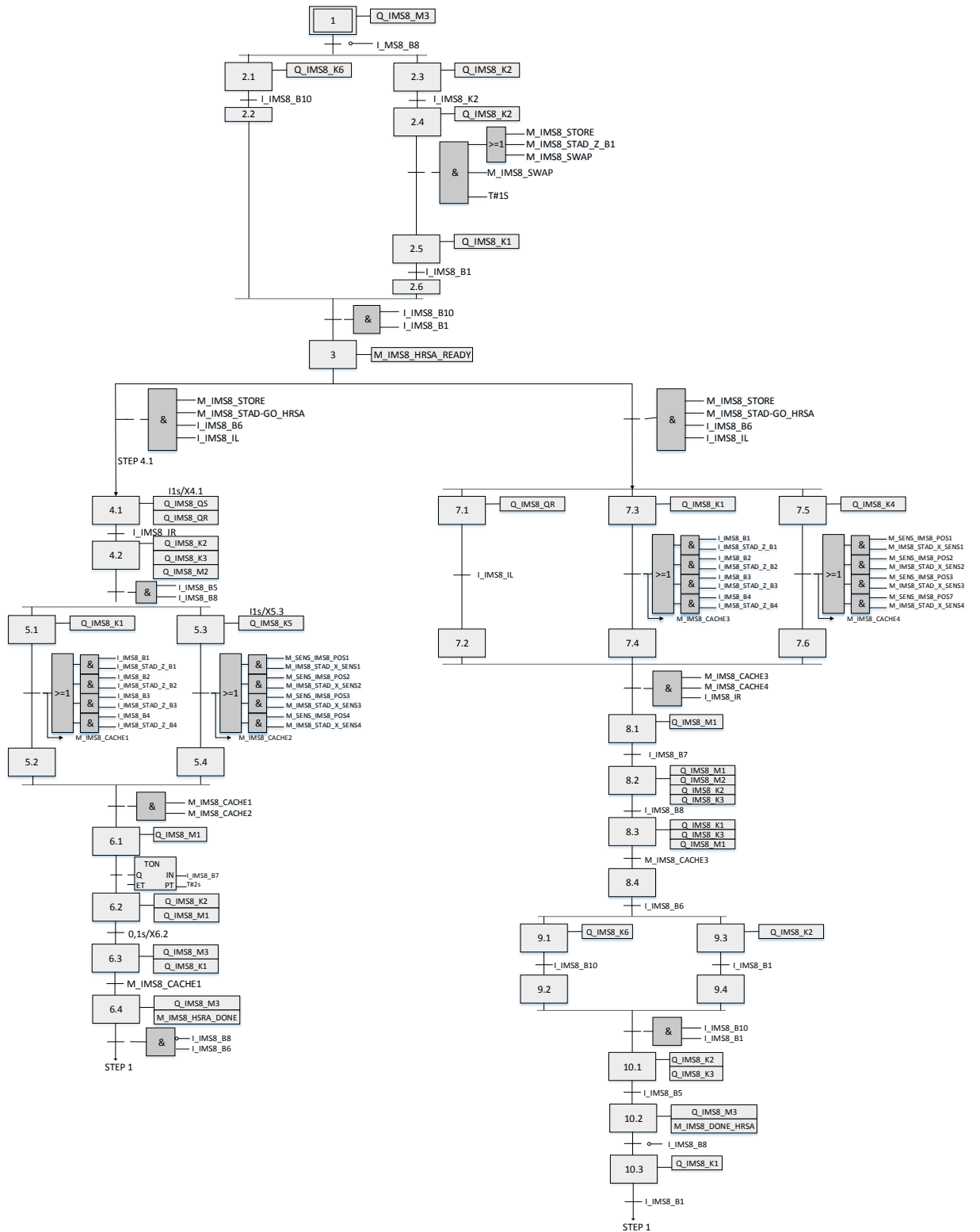
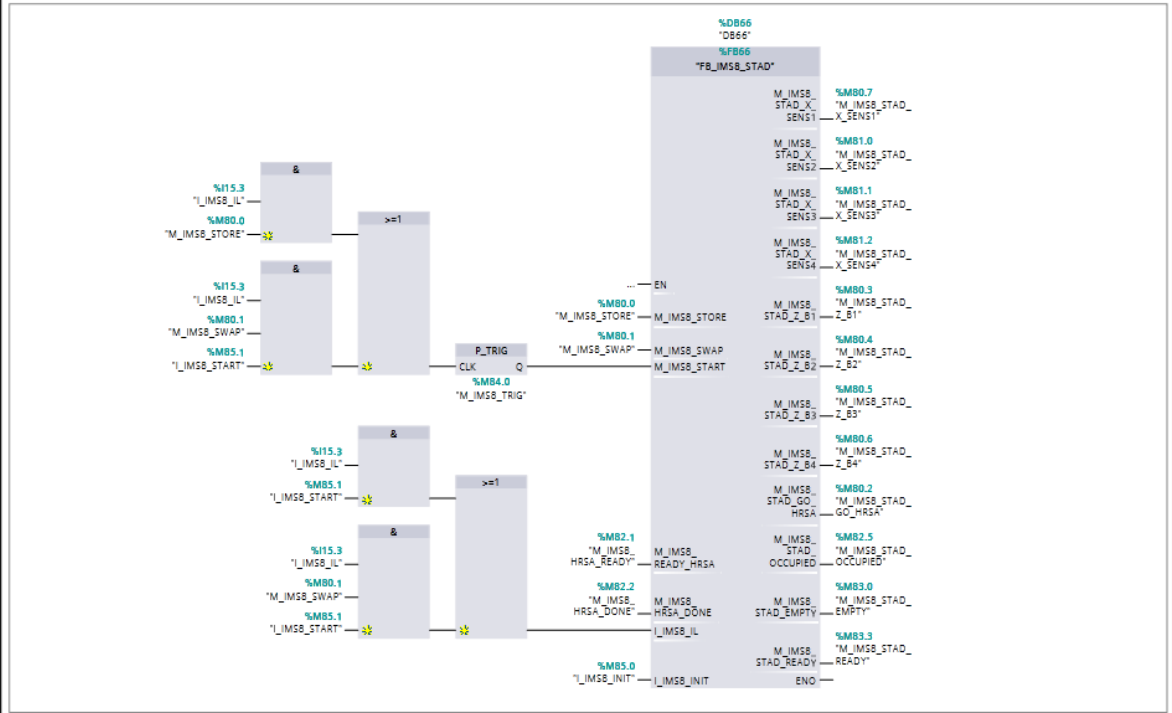


Figura: Control secuencial Grafcet

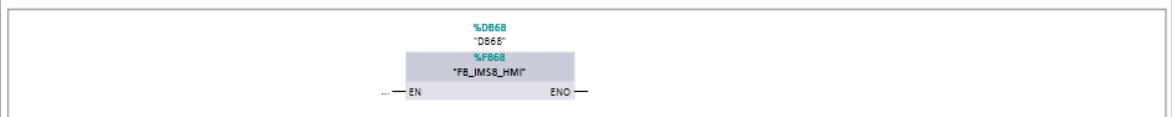
## ANEXO 3. Programación implementada

### a) Main (Organización de Funciones).

Segmento 1: Bloque STAT gestor del almacenamiento y despacho.

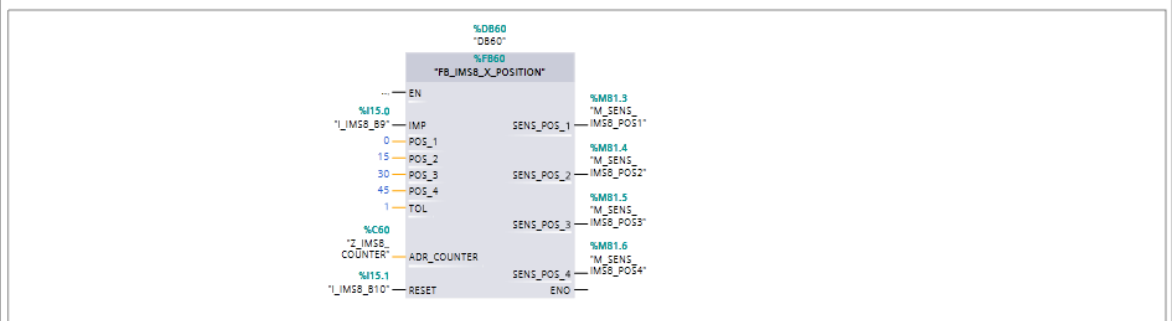


Segmento 2: Bloque de señales implementadas en el Hmi

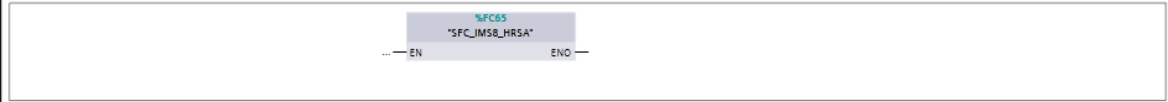


Símbolo	Dirección	Tipo	Comentario
---------	-----------	------	------------

Segmento 3: Bloque X\_POSITION genera las posición en a lo largo del eje x que se colocara o se extraera el paquete de botellas.

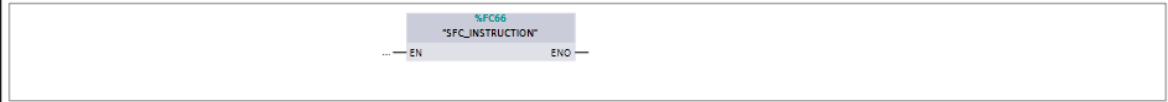


Segmento 4: Bloque HRSA control de secuencial para depositar o retirar piezas. ESTADOS



Símbolo	Dirección	Tipo	Comentario
---------	-----------	------	------------

Segmento 5: Bloque HRSA control de secuencial para depositar o retirar piezas. SALIDAS

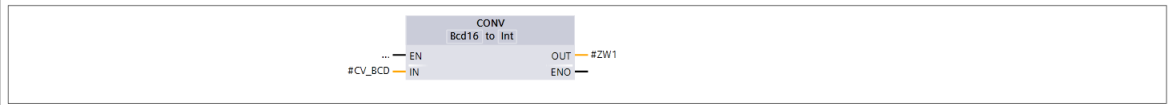


b) FB\_IMS8\_X\_POSITION.

Segmento 1: Convertidor de las señal de impulso por B9 a decimal



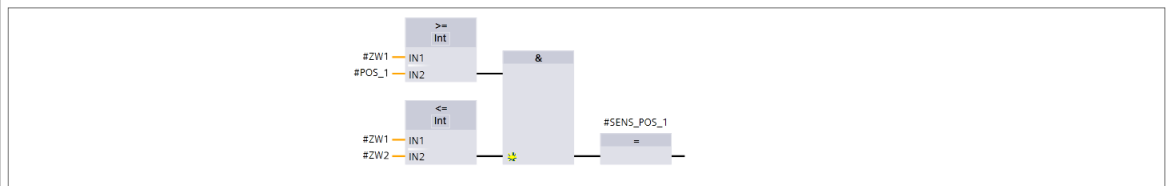
Segmento 2: Asignación de valor en decimal de impulsos a variable #ZW1



Segmento 3: Comparación y Asignación de valor en decimal de impulsos a variable #ZW2 entre #ZW1



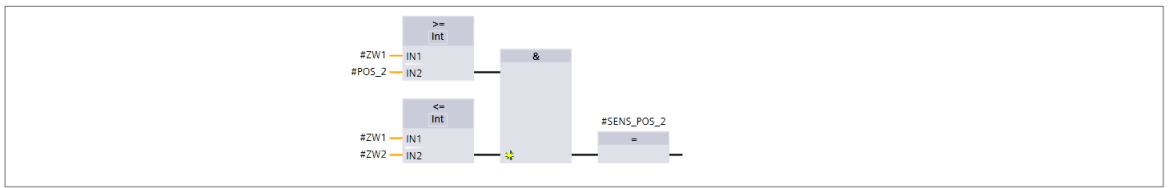
Segmento 4: Determinar el valor de posición 1



Segmento 5: Comparación de #ZW2 Y posición 2



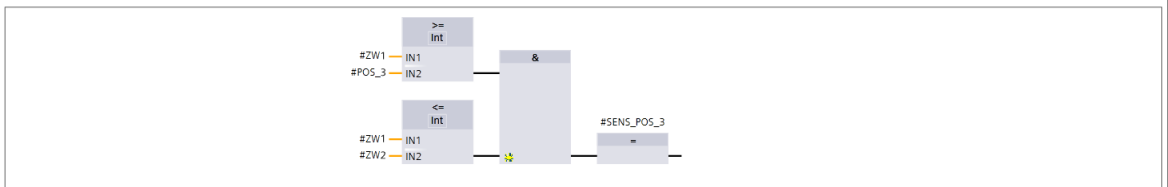
Segmento 6: Determina el valor de posición 2



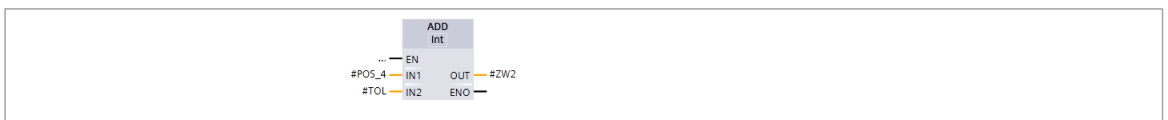
Segmento 7: Comparación de #XW2 Y posición 3



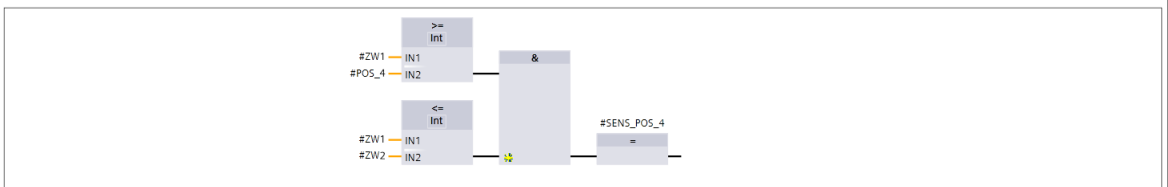
Segmento 8: Determina el valor de posición 3



Segmento 9: Comparación de #XW2 Y posición 4

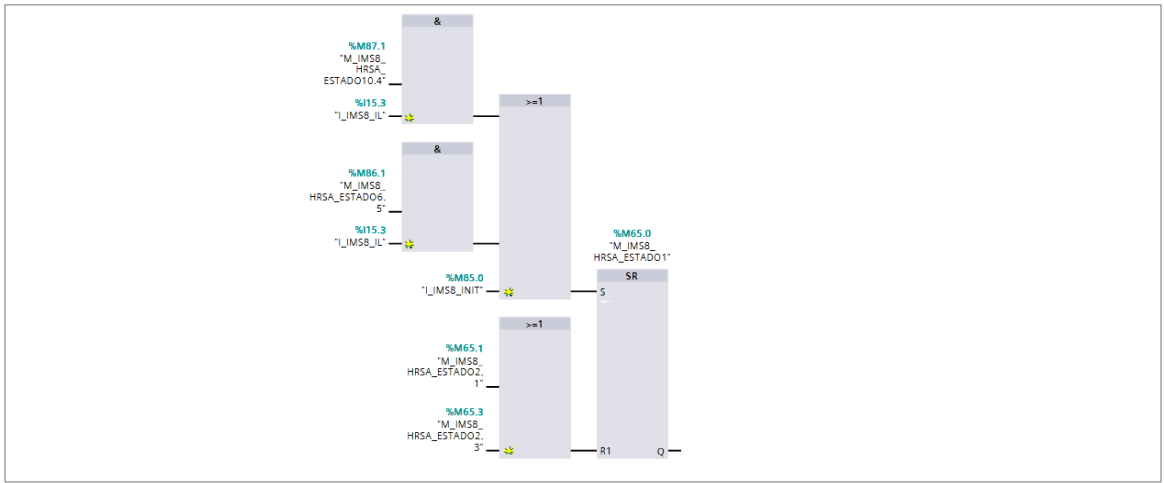


Segmento 10: Determina el valor de posición 4

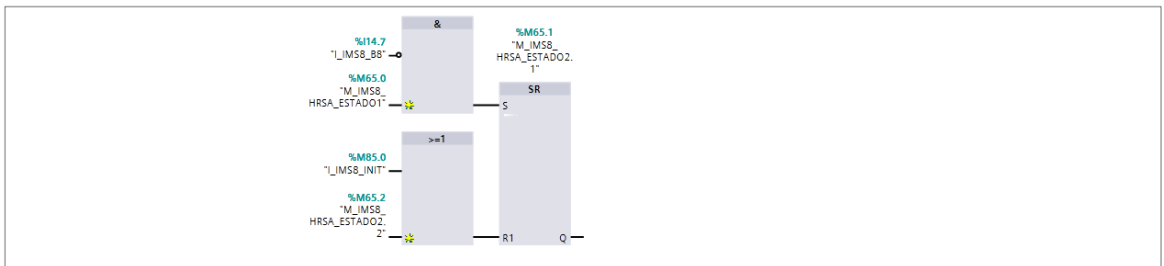


**c) SFC\_IMS8\_HRSA.**

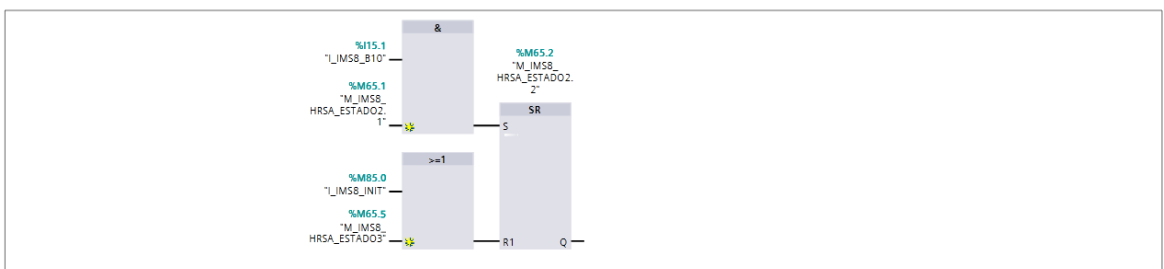
**Segmento 1: Transiciones de los estados 10.4 y 6.5 hacia el estado 1.**



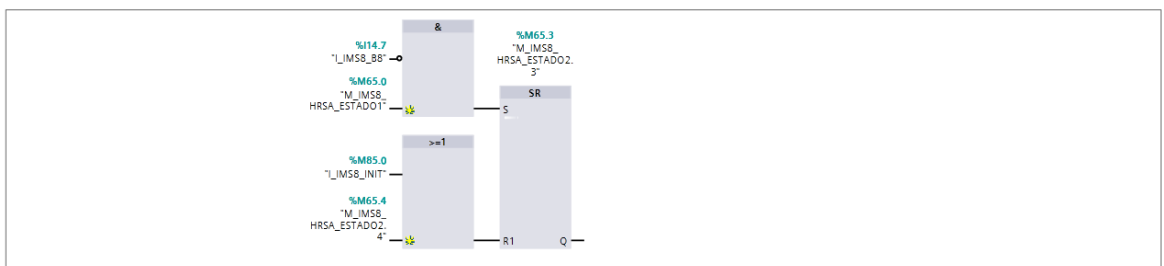
**Segmento 2: Transición del estado 1 hacia el estado 2.1.**



**Segmento 3: Transición del estado 2.1 hacia el estado 2.2.**

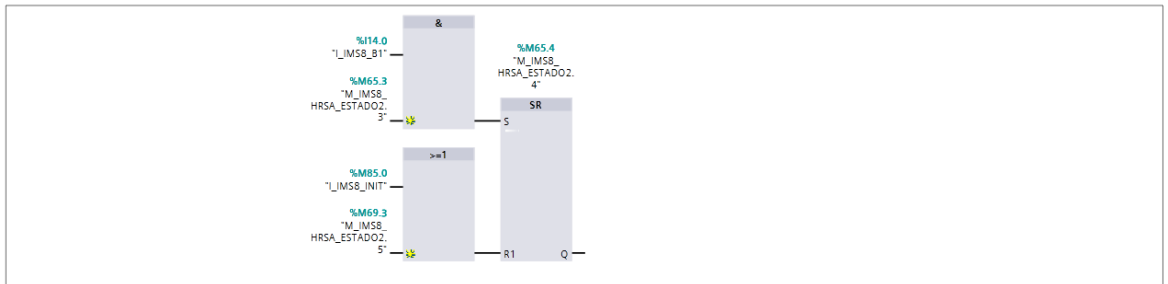


**Segmento 4: Transición del estado 1 hacia el estado 2.3.**

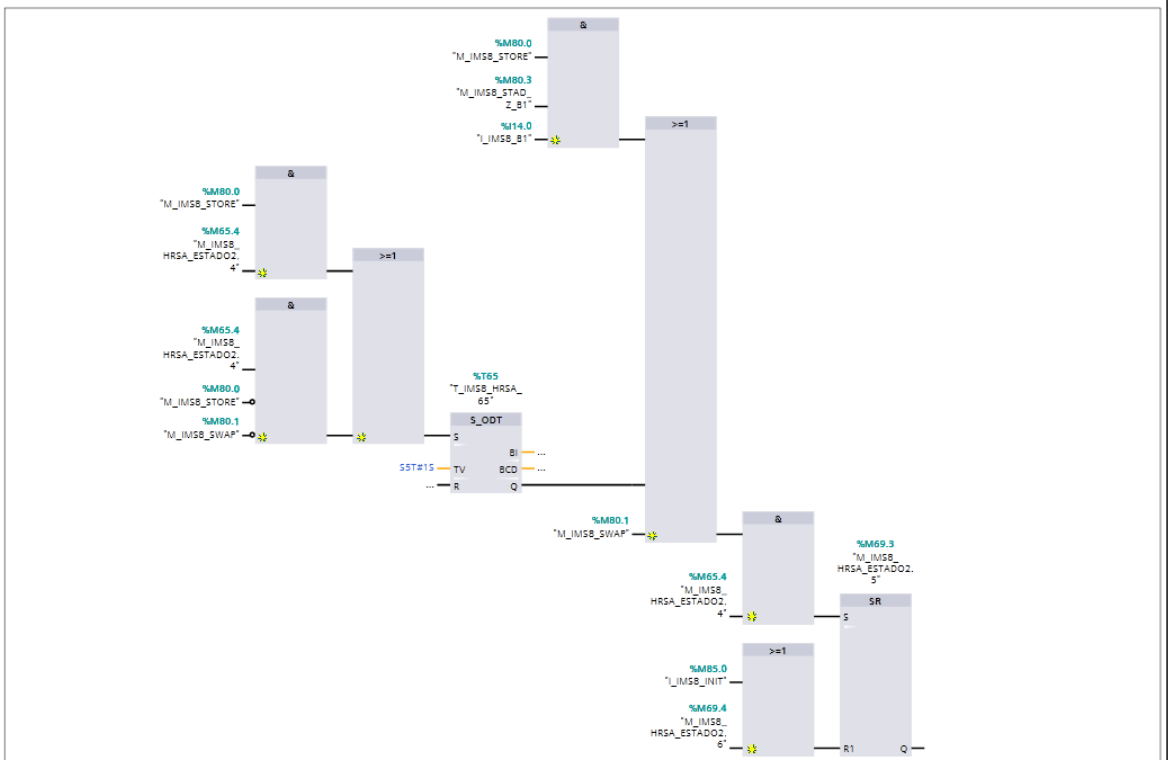




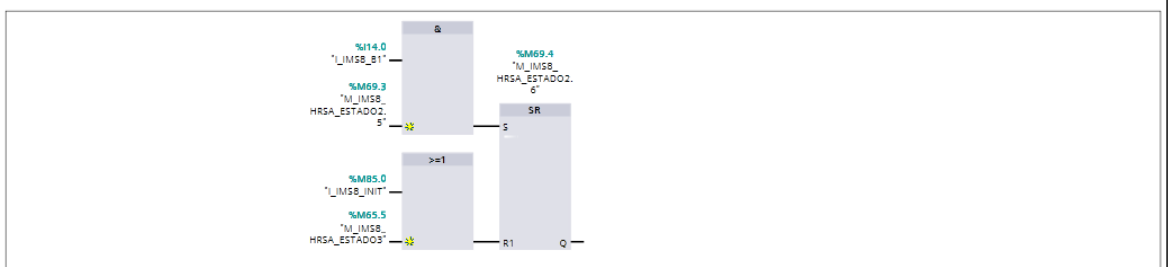
Segmento 5: Transición del estado 2.3 hacia el estado 2.4.



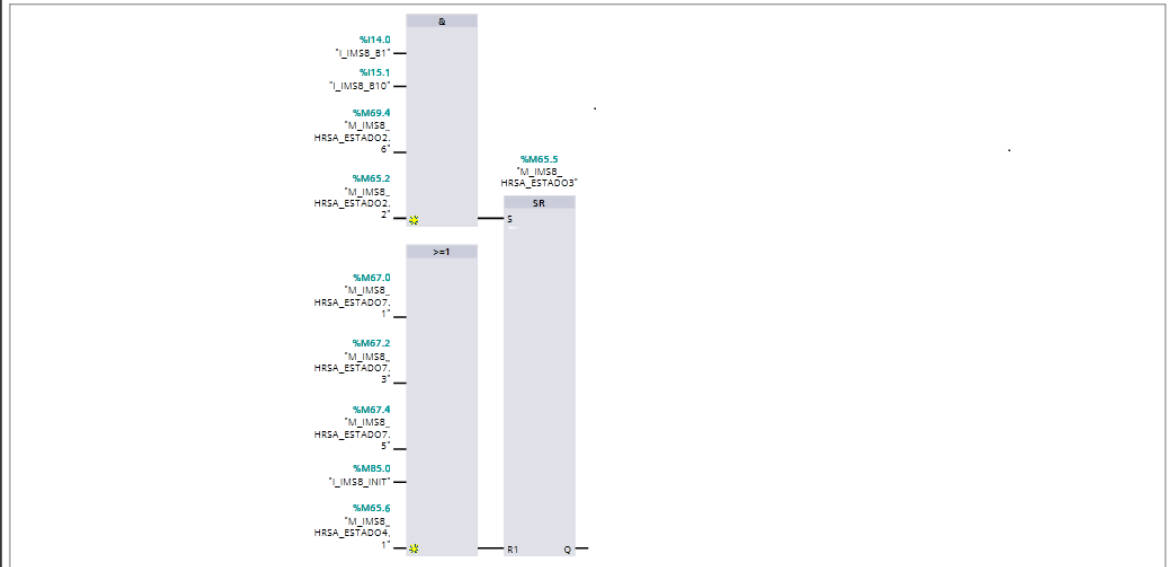
Segmento 6: Transición del estado 2.4 hacia el estado 2.5.



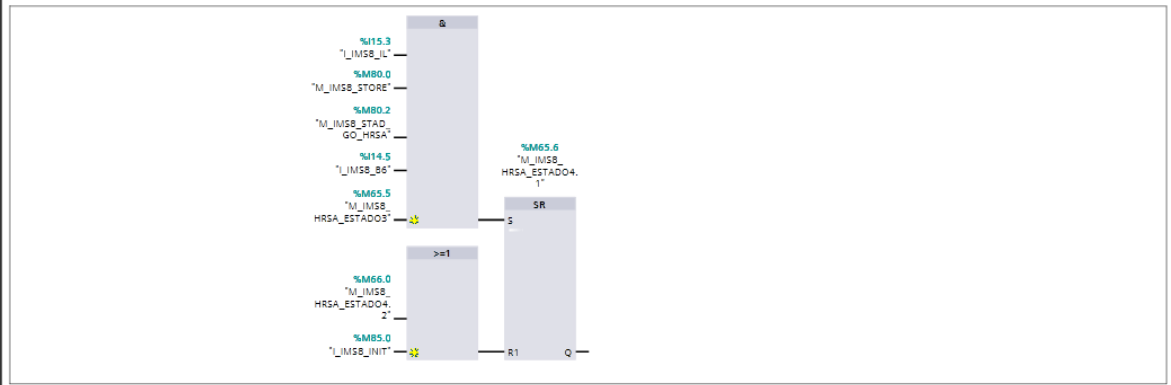
Segmento 7: Transición del estado 2.5 hacia el estado 2.6.



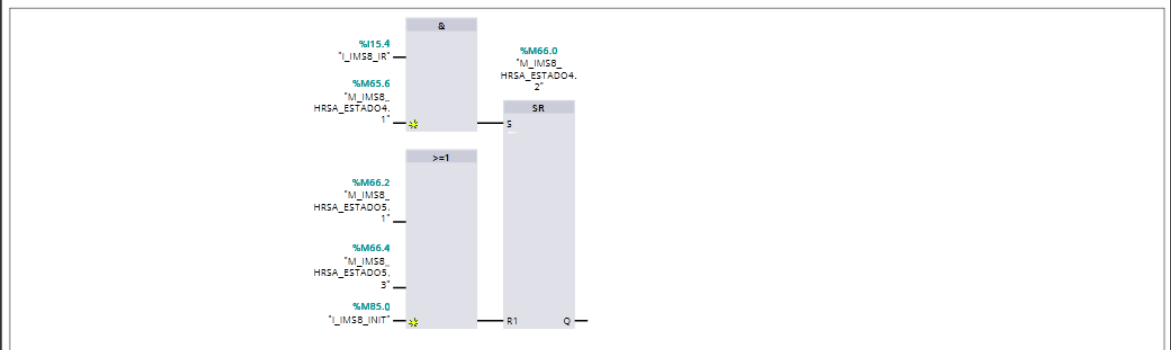
Segmento 8: Transiciones de los estados 2.6 y 2.2 hacia el estado 3.



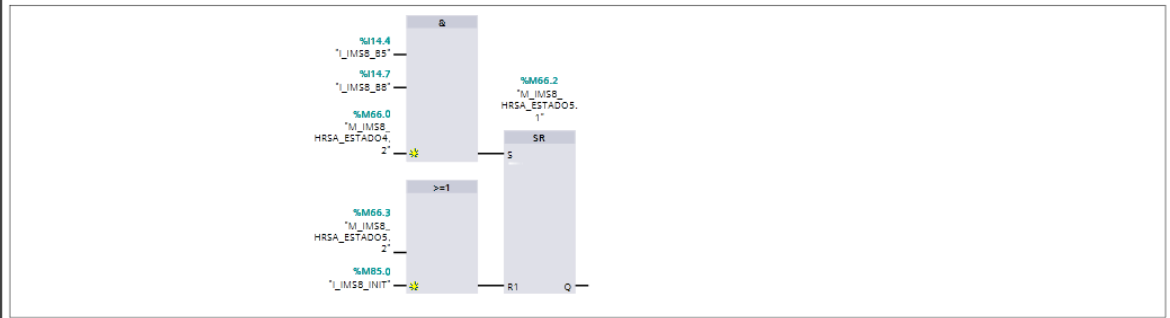
Segmento 9: Transición del estado 3 hacia el estado 4.1.



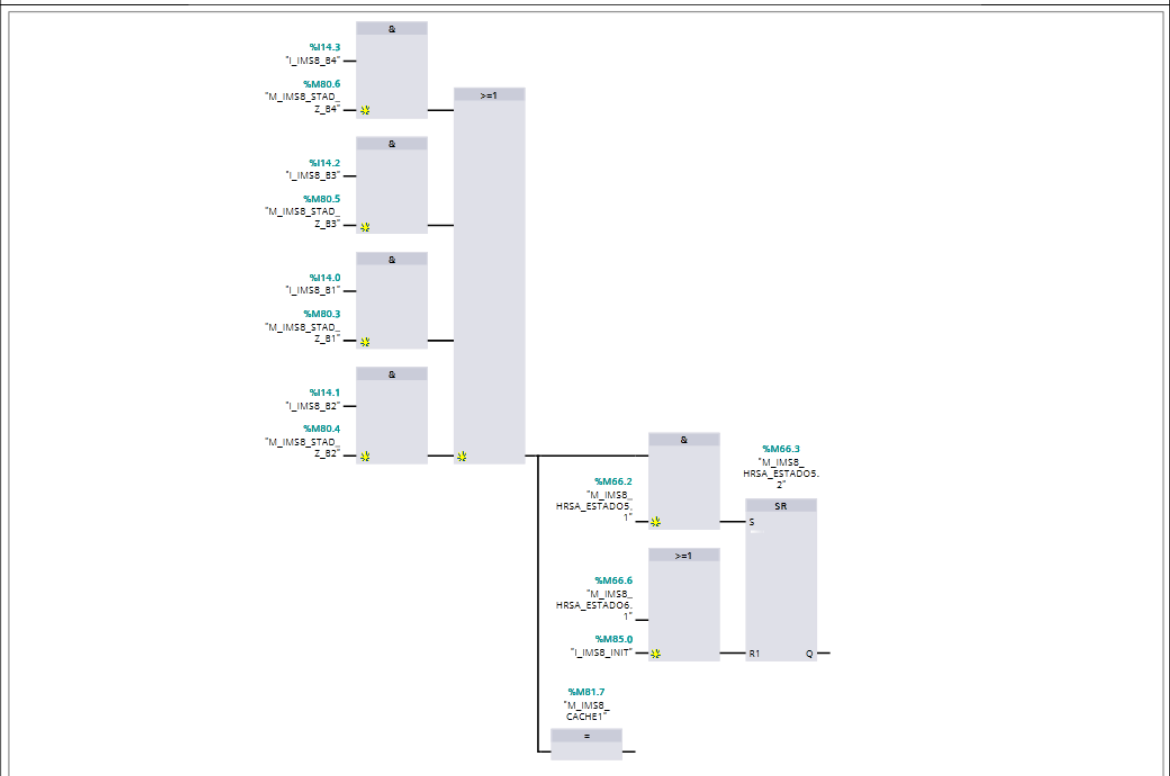
Segmento 10: Transición del estado 4.1 hacia el estado 4.2.



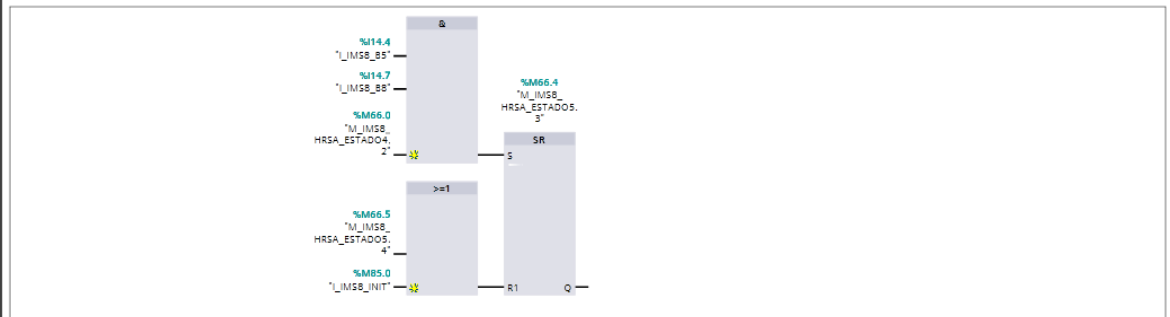
Segmento 11: Transición del estado 4.2 hacia el estado 5.1.



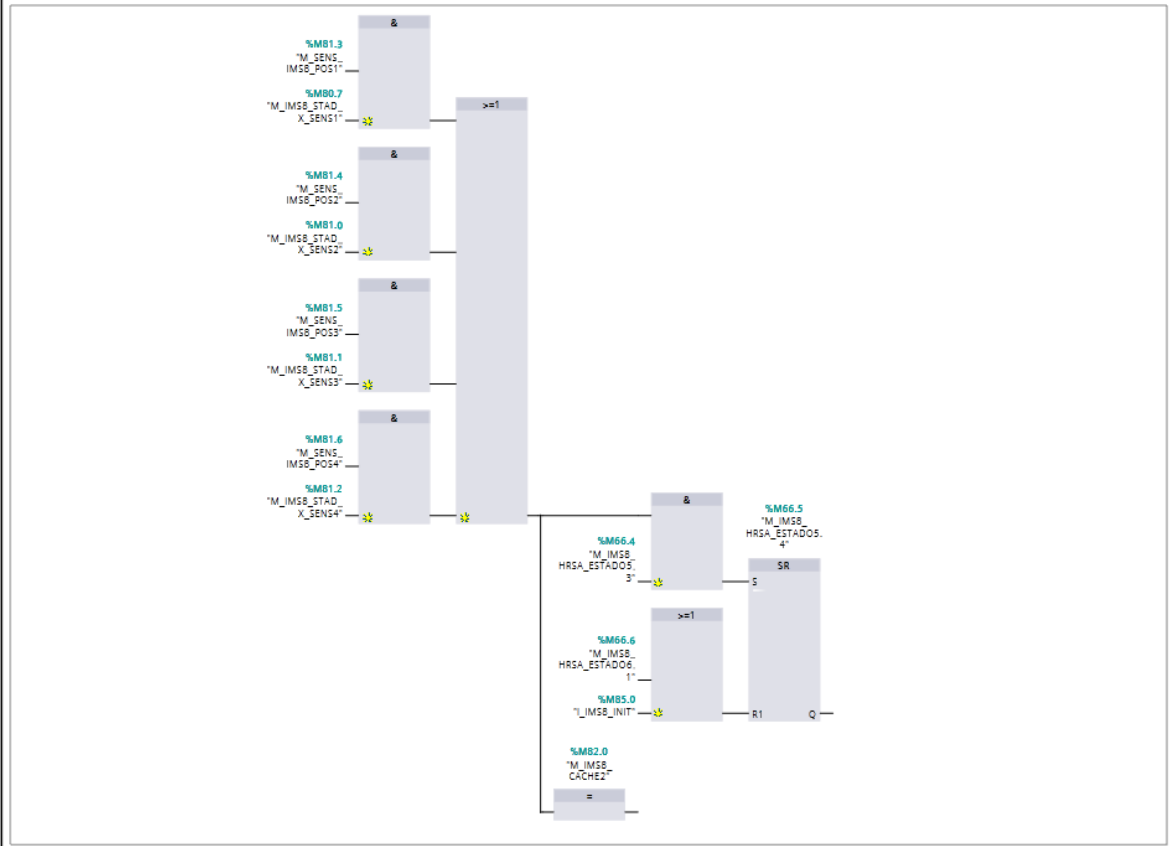
Segmento 12: Transición del estado 5.1 hacia el estado 5.2.



Segmento 13: Transición del estado 4.2 hacia el estado 5.3.



Segmento 14: Transición del estado 5.3 hacia el estado 5.4.

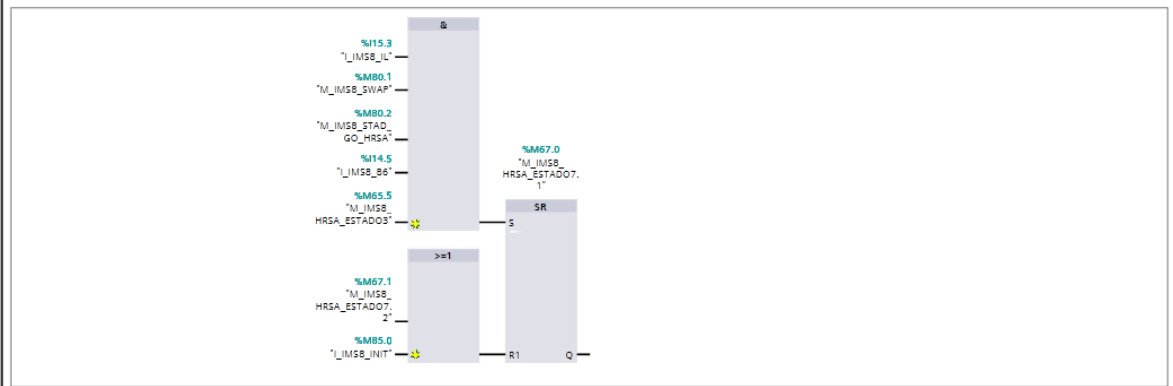


Segmento 15: Transiciones de los estados 5.2 y 5.4 hacia el estado 6.1.

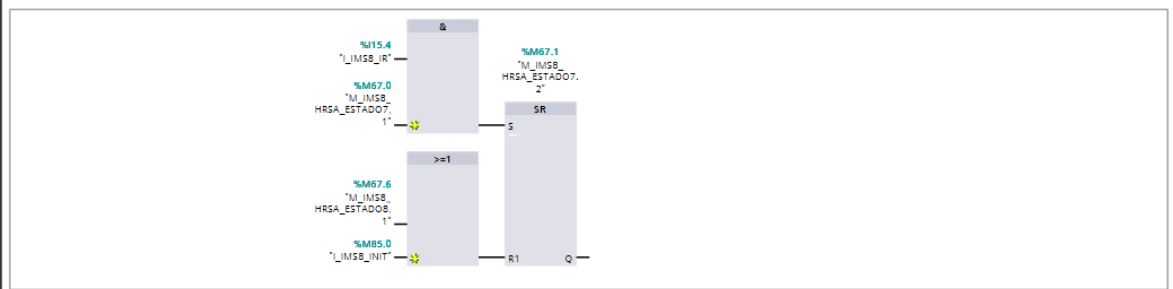




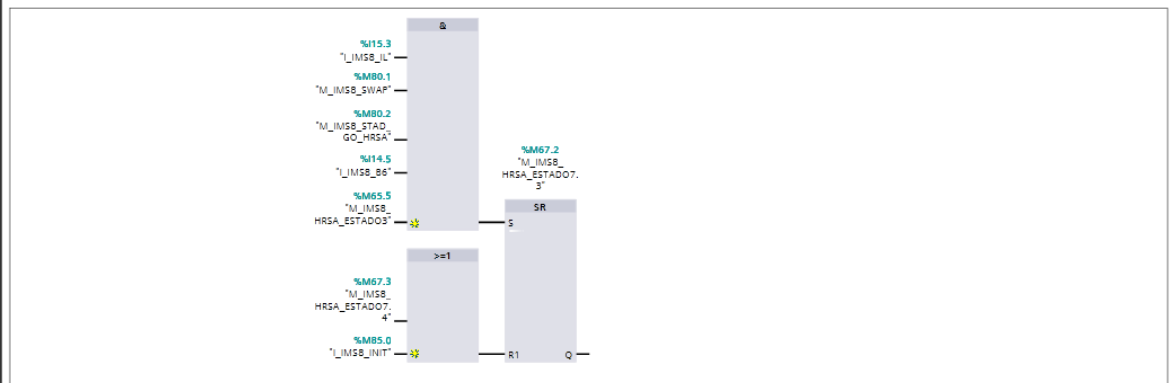
Segmento 19: Transición del estado 3 hacia el estado 7.1.



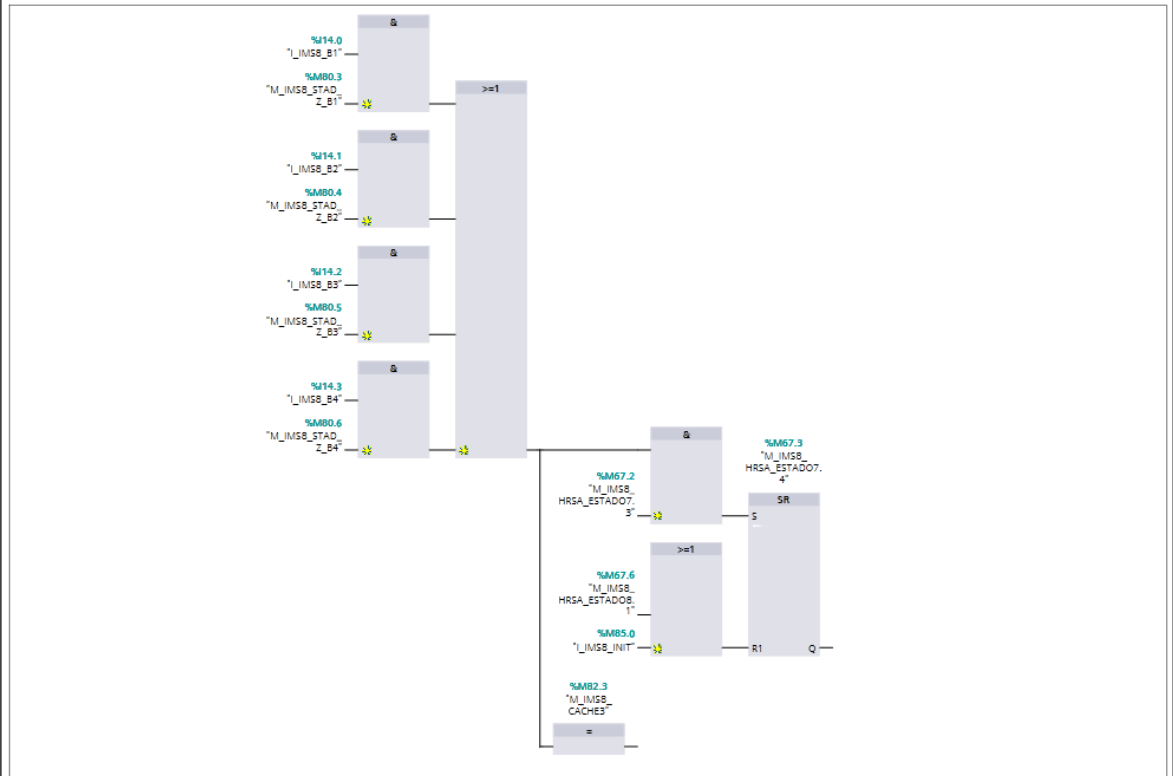
Segmento 20: Transición del estado 7.1 hacia el estado 7.2.



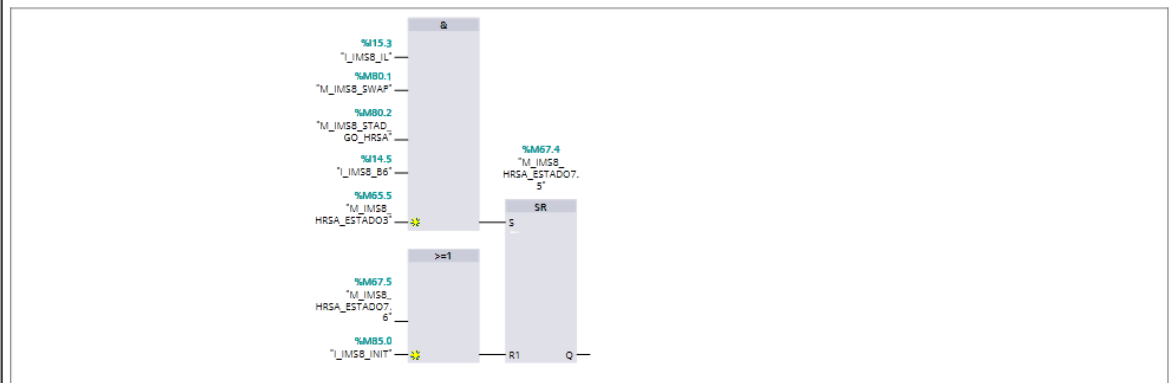
Segmento 21: Transición del estado 3 hacia el estado 7.3.



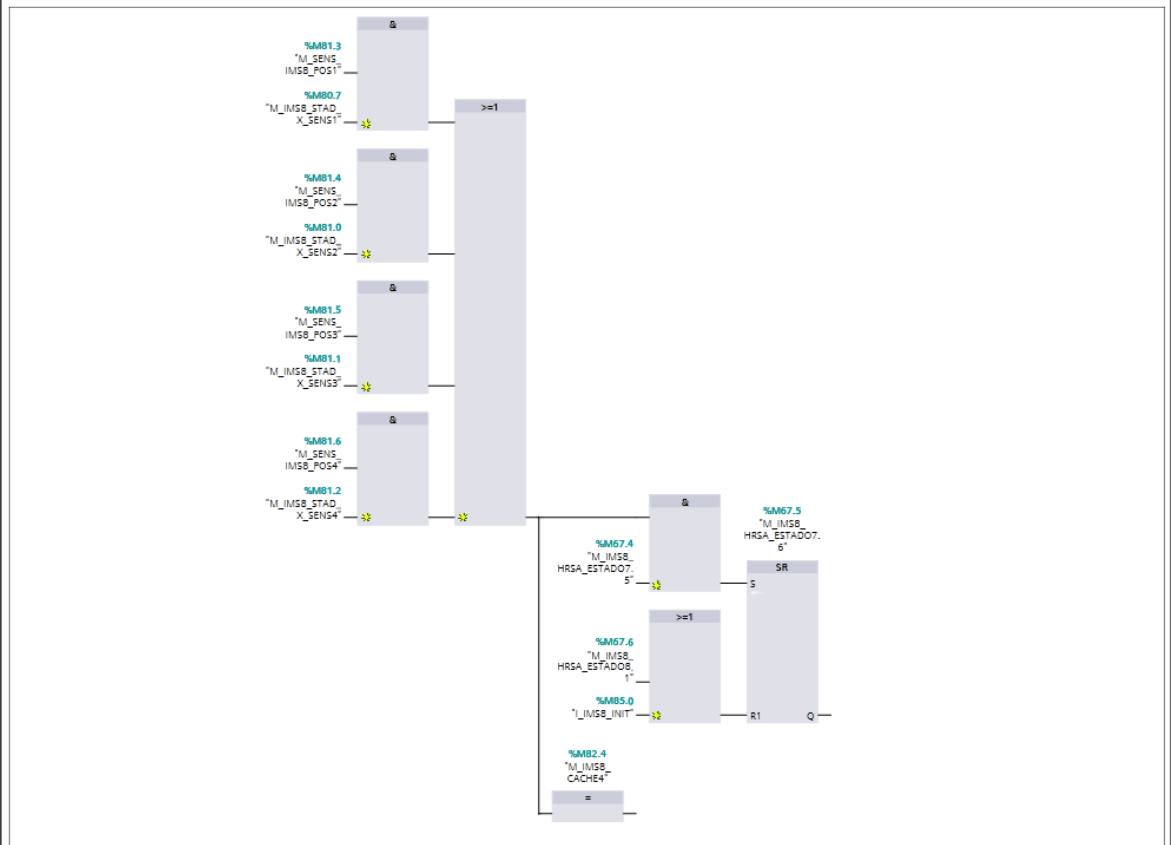
Segmento 22: Transición del estado 7.3 hacia el estado 7.4.



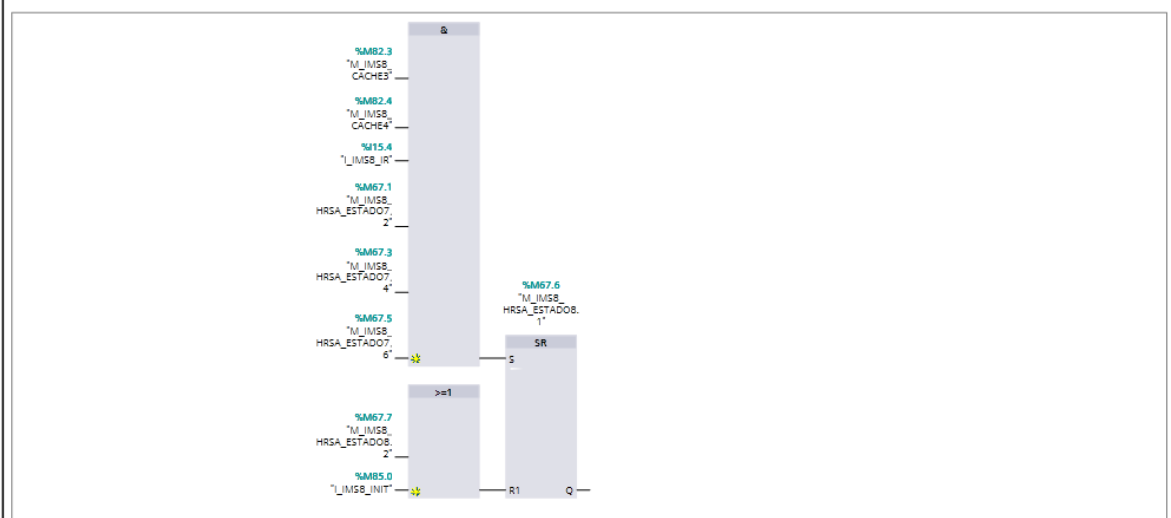
Segmento 23: Transición del estado 3 hacia el estado 7.5.



Segmento 24: Transición del estado 7.5 hacia el estado 7.6.

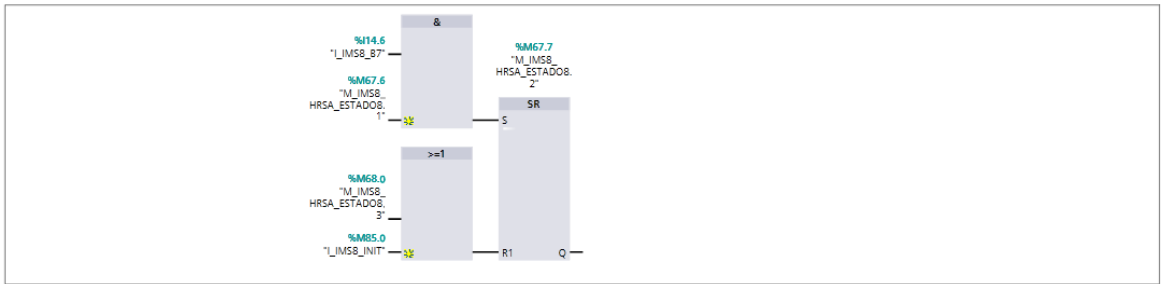


Segmento 25: Transiciones de los estados 7.2 , 7.4 y 7.6 hacia el estado 8.1.

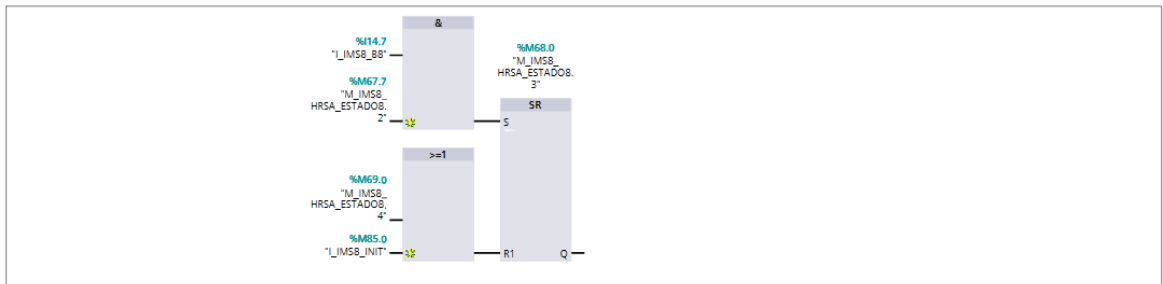




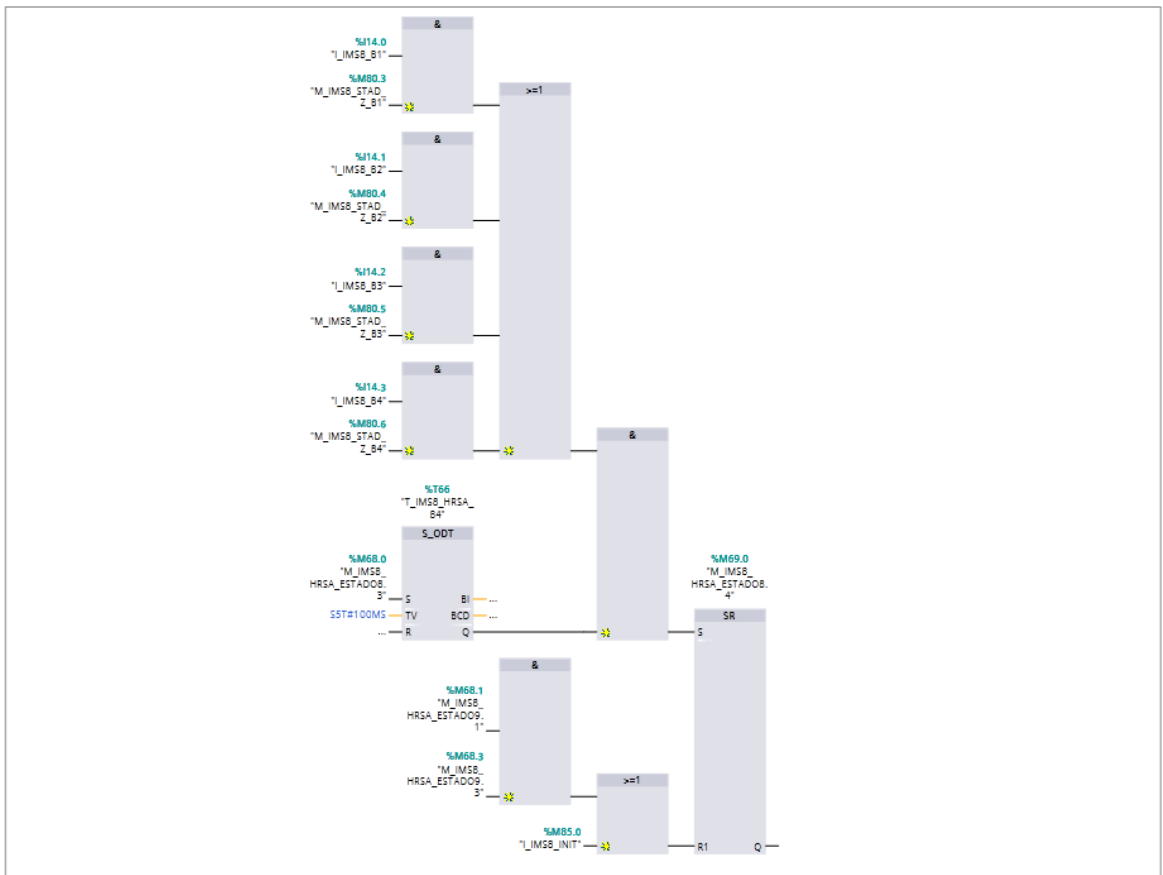
Segmento 26: Transición del estado 8.1 hacia el estado 8.2.



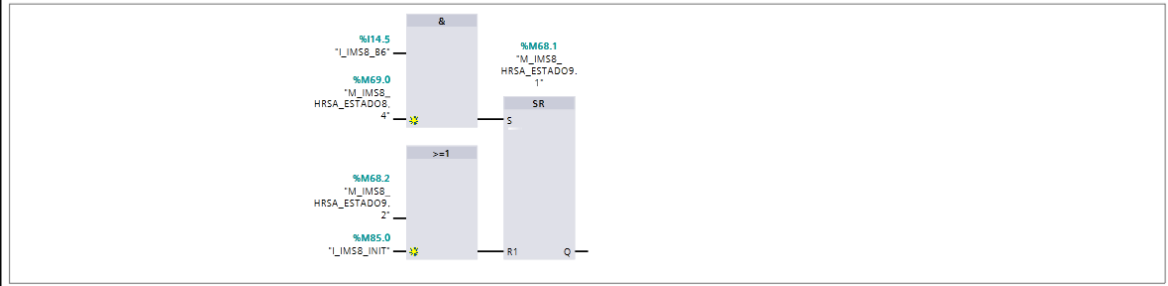
Segmento 27: Transición del estado 8.2 hacia el estado 8.3.



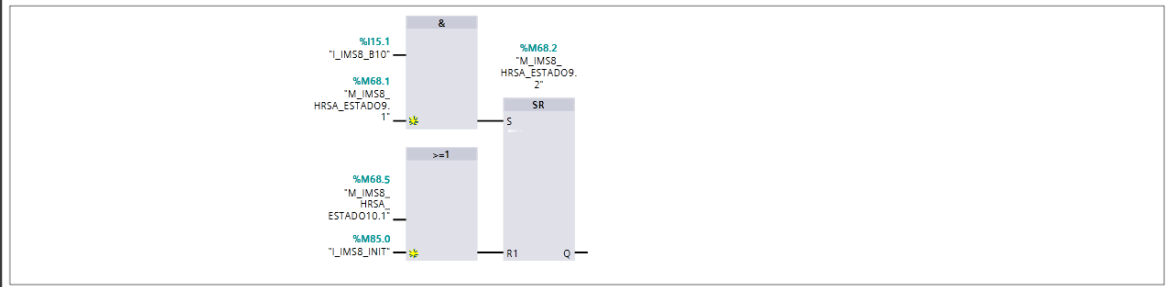
Segmento 28: Transición del estado 8.3 hacia el estado 8.4.



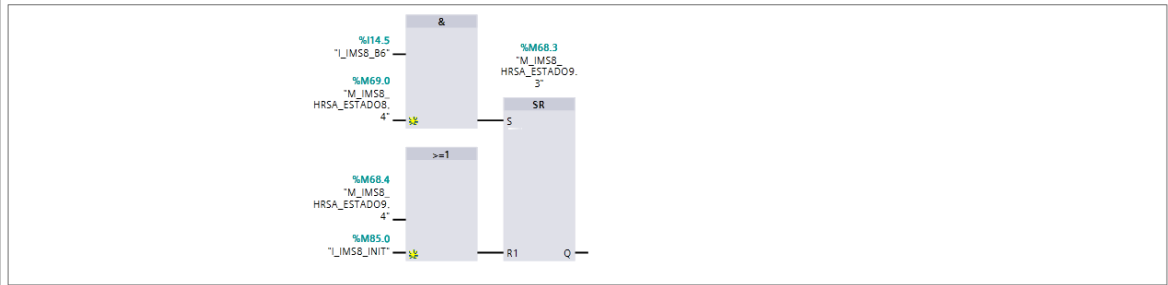
**Segmento 29: Transición del estado 8.4 hacia el estado 9.1.**



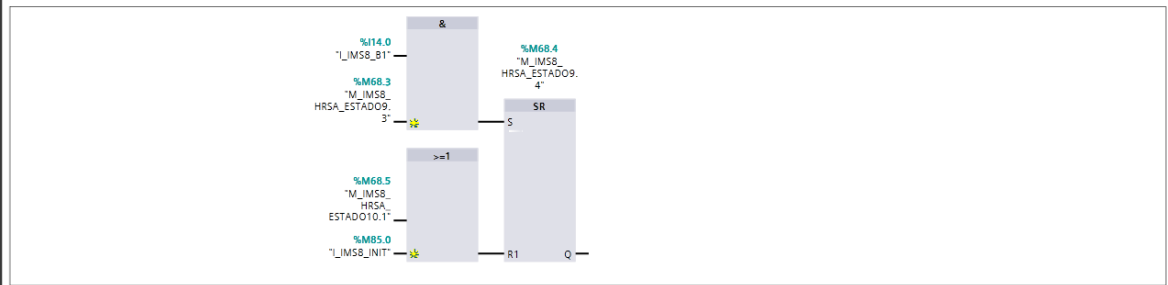
**Segmento 30: Transición del estado 9.1 hacia el estado 9.2.**



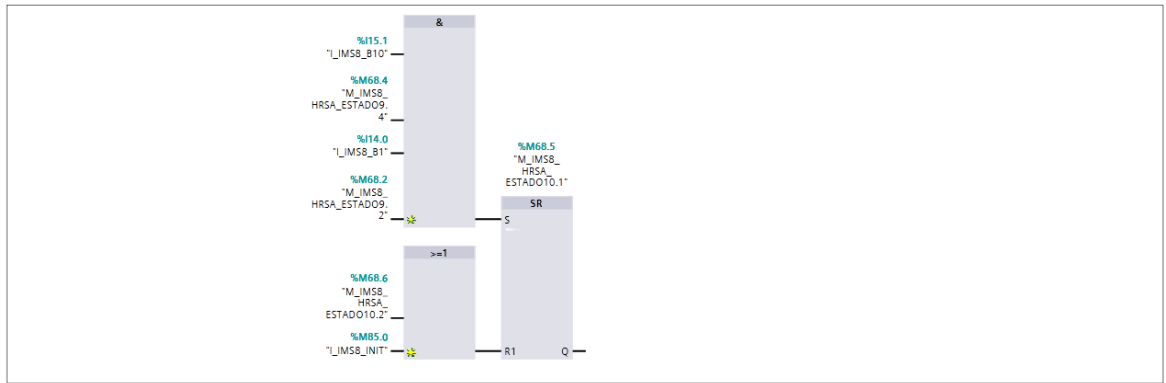
**Segmento 31: Transición del estado 8.4 hacia el estado 9.3.**



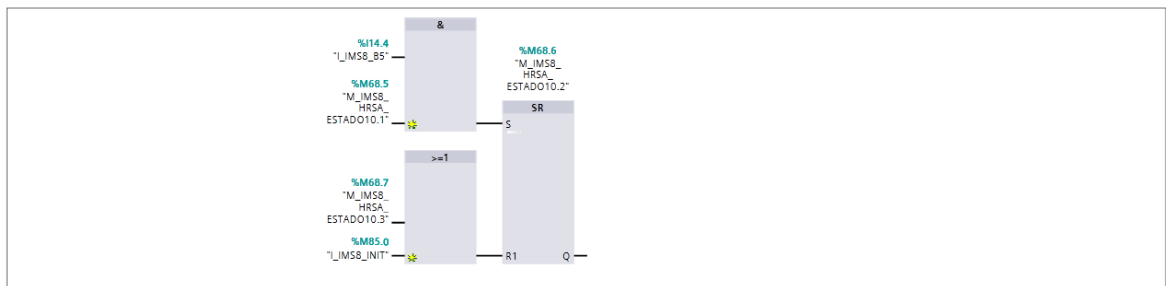
**Segmento 32: Transición del estado 9.3 hacia el estado 9.4.**



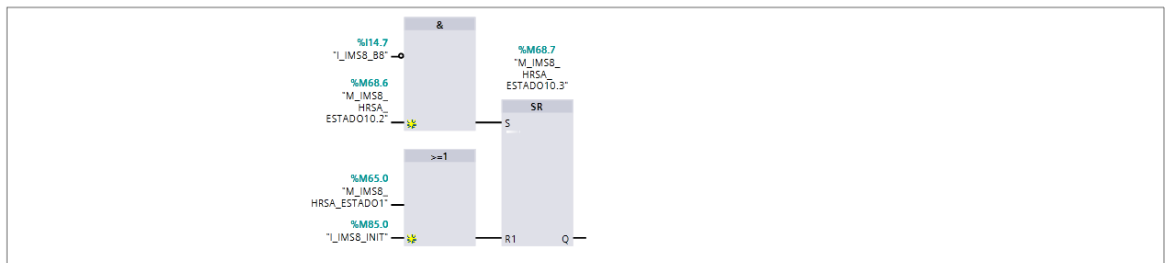
Segmento 33: Transiciones de los estados 9.2 y 9.4 hacia el estado 10.1.



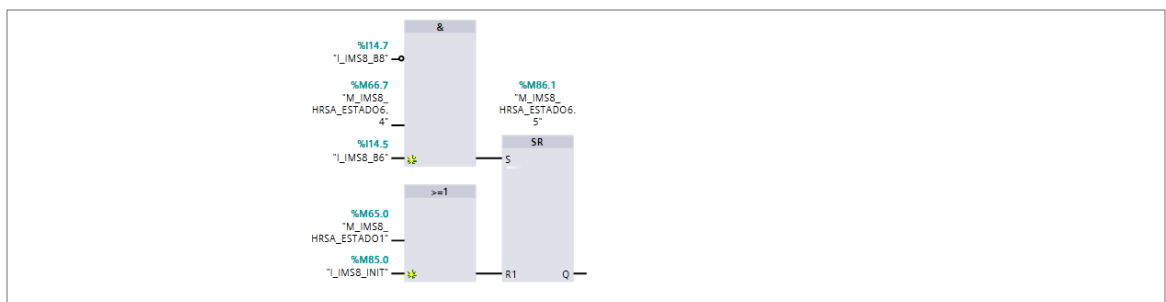
Segmento 34: Transición del estado 10.1 hacia el estado 10.2.



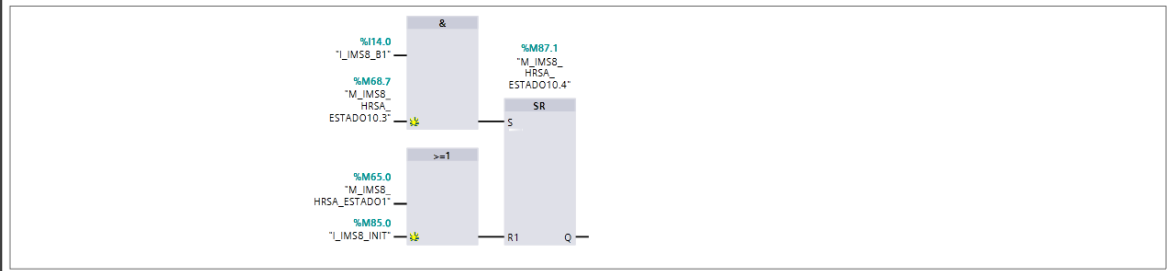
Segmento 35: Transición del estado 10.2 hacia el estado 10.3.



Segmento 36: Transición del estado 6.4 hacia el estado 6.5.

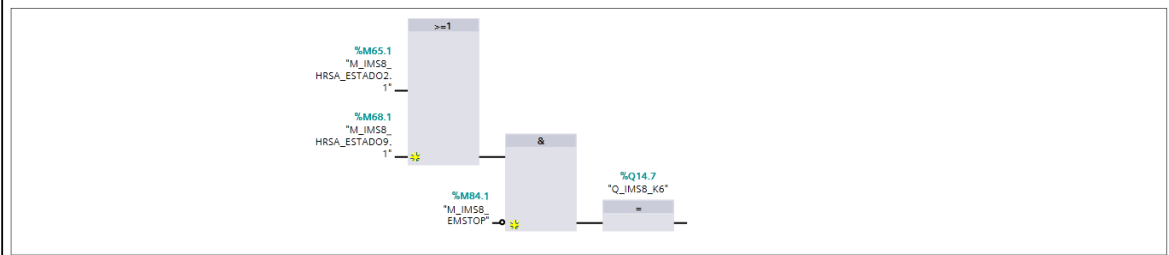


Segmento 37: Transición del estado 10.3 hacia el estado 10.4.

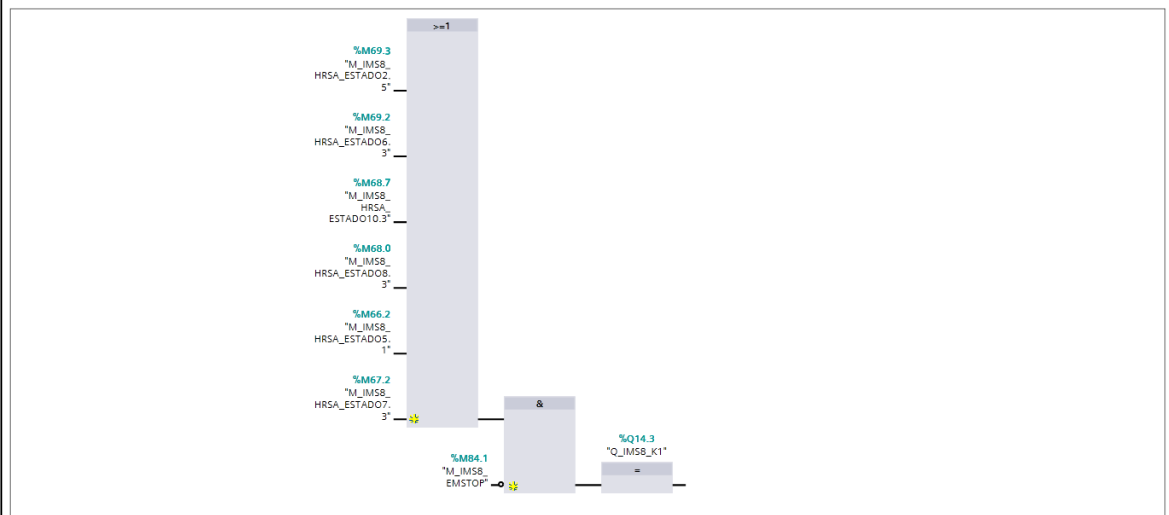


d) SFC\_INSTRUCTION.

Segmento 1: Permite el movimiento de avance del brazo a lo del eje X. en los estados 2.1 y 9.1



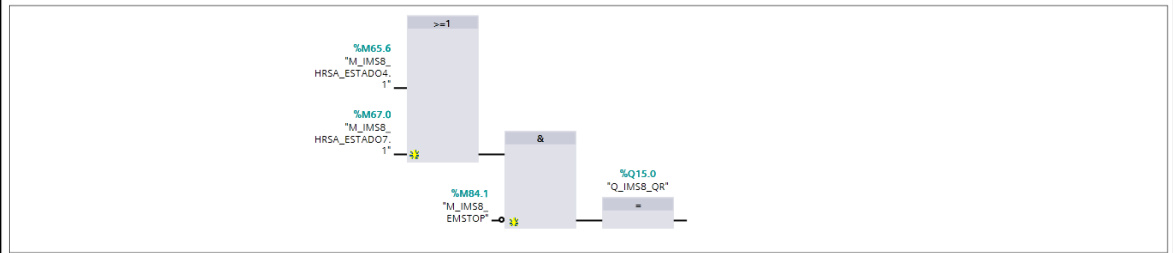
Segmento 2: Permite el movimiento de ascenso del brazo a lo largo del eje z en los estados 2.5 , 6.3, 10.3 , 8.3 , 5.1 y 7.3



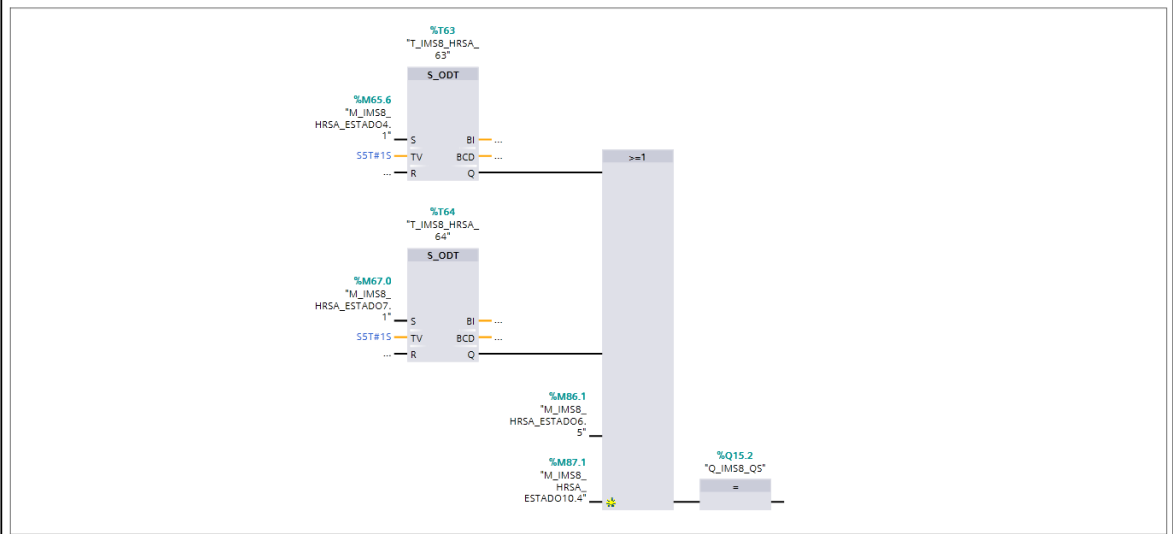
Segmento 3: Señal indicadora que el bloque HRSA esta listo para un despacho o un almacenado en el estado 3



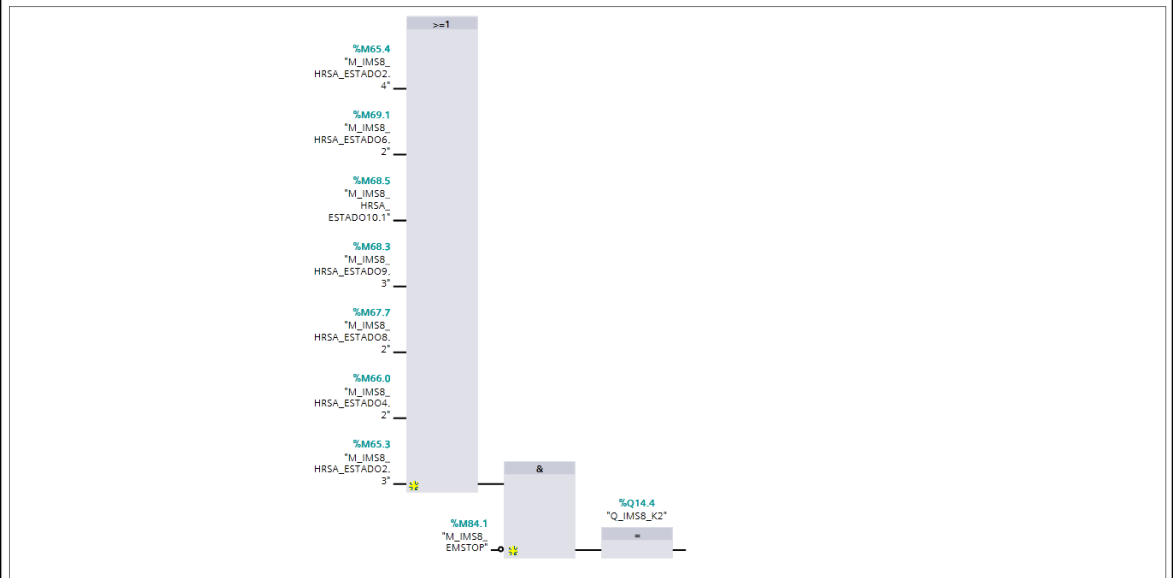
Segmento 4: Activar la banda en en sentido de la derecha en los estados 4.1 y 7.1



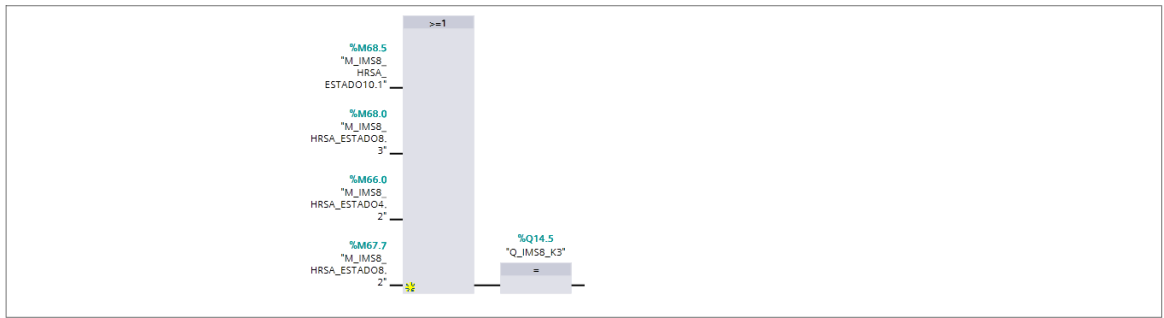
Segmento 5: Activar marcha lenta de la banda en los estados 4.1, 7.1, 6.5 y 10.4



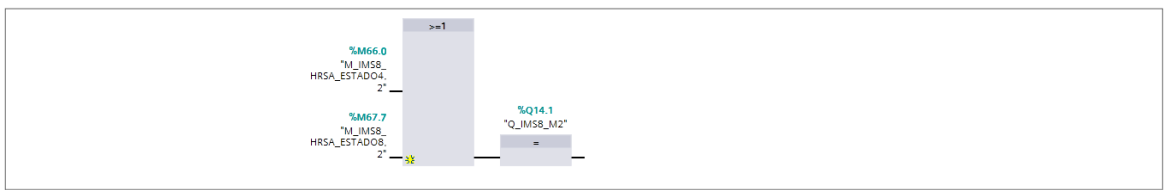
Segmento 6: Permite el movimiento de descenso del brazo a lo largo del eje z en los estados 2.4 , 6.2, 10.1 , 9.3 , 8.2, 4.2 y 2.3.



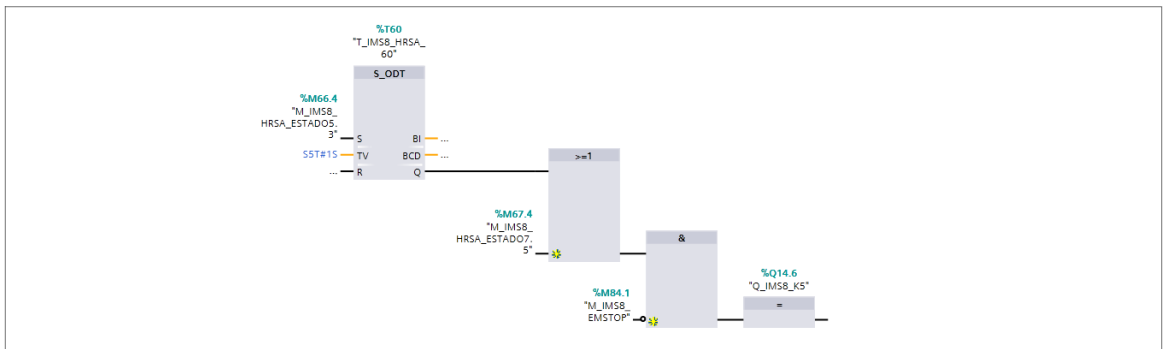
Segmento 7: Activar marcha lenta en el ascenso o descenso del brazo en los estados 10.1, 8.3, 4.2 y 8.2



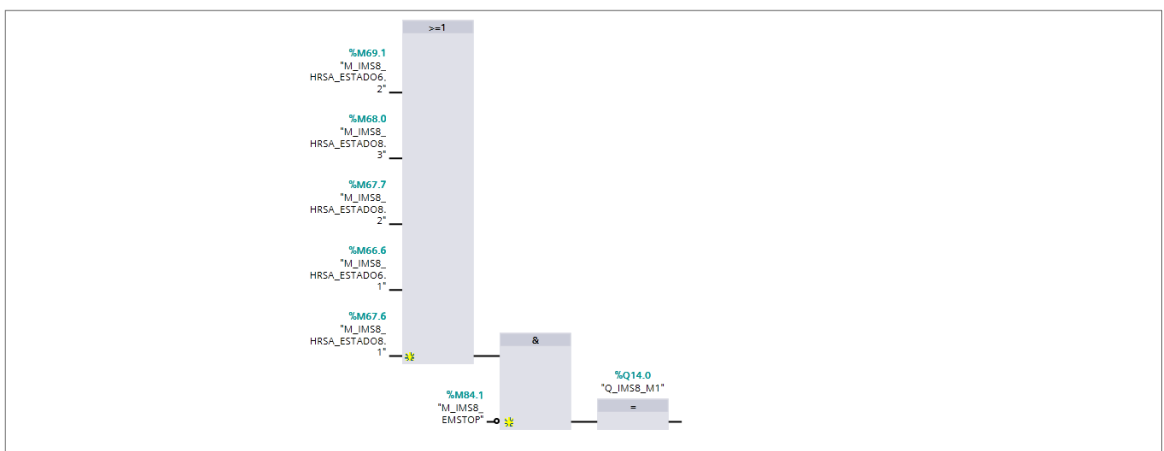
Segmento 8: Activar la formación de vacío (tubo de Venturi) en los estados 4.2 y 8.2.



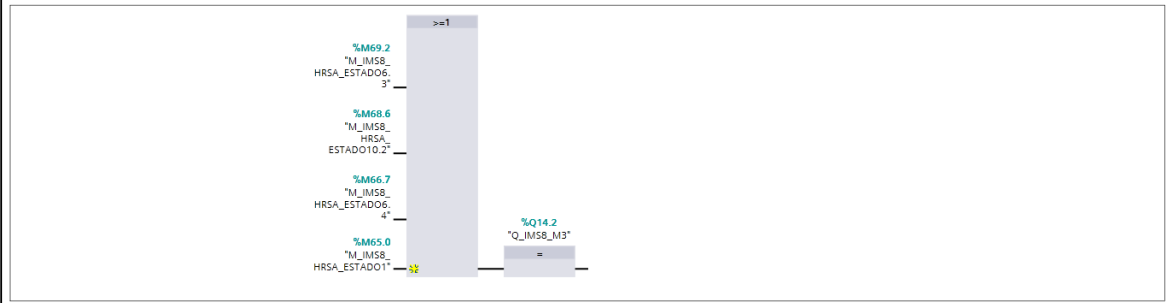
Segmento 9: Permite el movimiento de regreso del brazo a lo del eje X. en los estados 5.3 y 7.5



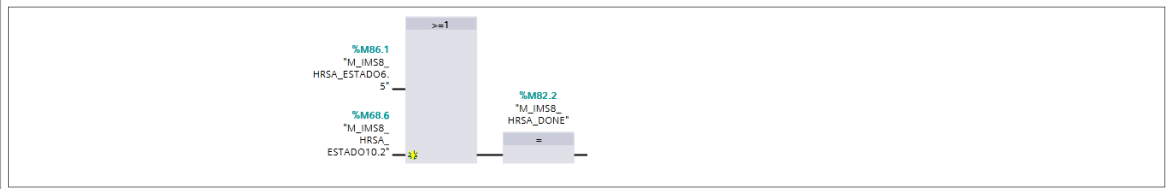
Segmento 10: Activar avance del cilindro neumático de doble acción por la activación de válvula de 4 a 2 vías en los estados 6.2 , 8.3, 8.2, 6.1 y 8.1



Segmento 11: Activar desconexión de vacío (tubo de Venturi) a través de una válvula de 5 a 2 vías en Iso estados 6.3, 10.2 , 6.4 y 1



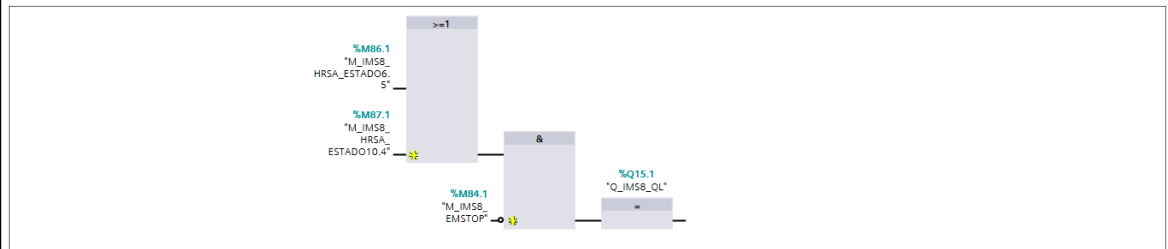
Segmento 12: Señal indicadora que el bloque HRSA termino de hacer un despacho o un almacenamiento en Iso estados 6.5 y 10.2



Segmento 13: Interruptor de emergencia detiene cualquier proceso que ejecute la planta siendo boton de paro d emergencia por alguno inconveniente por el I\_IMS8\_EMSTOP RESET CON I\_IMS8\_ACK



Segmento 14: Activar la banda en en sentido de la IZQUIERDA en los estados 6.5 y 10.4.



**e) FB\_IMS8\_STAD**

```
// Declaración del Grafcet para el bloque STAD para sus estados, transiciones y salidas.
// Iniciación del secuenciador en el estado 1
IF #I_IMS8_INIT=TRUE THEN
  FOR #I:=1 TO 4 DO
    #M_IMS8_X_SENS[#I]:=FALSE;
    #M_IMS8_Z_B[#I]:=FALSE;
  END_FOR;
  FOR #I:=1 TO 4 DO
    FOR #J:=1 TO 4 DO
      #LEVEL_PLACE[#I,#J]:=FALSE;
      #M_PLACE[#I,#J]:=0;
    END_FOR;
  END_FOR;
  #CONT := 0;
  #M_IMS8_STAD_GO_HRSA:=FALSE;// comando GO para el restablecimiento de la
  unidad de almacenamiento y despacho
  #M_IMS8_STAD_EMPTY:=TRUE;// Restablecer VACÍO
  #M_IMS8_STAD_OCCUPIED:=FALSE;// Reset OCUPADO
  (*Iniciación de la secuencia de pasos desde el estado 1 *)

#STEP1:=TRUE;#STEP2_1:=FALSE;#STEP2_2:=FALSE;#STEP3_1:=FALSE;#STEP3_
2:=FALSE;
  END_IF;
//Condiciones de transición sin el estado 1
IF (#STEP2_2 AND #M_IMS8_READY_HRSA) OR (#STEP3_2 AND
#M_IMS8_READY_HRSA)//Estado 1
  THEN #STEP1:=TRUE;
  ELSIF (#STEP2_1 OR #STEP3_1)
  THEN #STEP1:=FALSE;
  END_IF;
//Condiciones de transición sin el estado 2.1
IF (#M_IMS8_STORE AND #M_IMS8_START AND #M_IMS8_READY_HRSA AND
#I_IMS8_IL AND #STEP1)//Estado 2.1
  THEN #STEP2_1:=TRUE;
```



```

ELSIF (#STEP2_2)
    THEN #STEP2_1:=FALSE;
    END_IF;
//Condiciones de transición sin el estado 2.2
IF      (#STEP2_1      AND      (#M_IMS8_HRSA_DONE      OR
#M_IMS8_STAD_OCCUPIED))//Estado 2.2
    THEN #STEP2_2:=TRUE;
    ELSIF (#STEP1)
        THEN #STEP2_2:=FALSE;
        END_IF;
//Condiciones de transición sin el estado 3.1
IF (#M_IMS8_SWAP AND #M_IMS8_START AND #M_IMS8_READY_HRSA AND
#I_IMS8_IL AND #STEP1)//Estado 3.1
    THEN #STEP3_1:=TRUE;
    ELSIF (#STEP3_2)
        THEN #STEP3_1:=FALSE;
        END_IF;
//Condiciones de transición sin el estado 3.2
IF (#STEP3_1 AND (#M_IMS8_HRSA_DONE OR #M_IMS8_STAD_EMPTY))//Estado 3.2
    THEN #STEP3_2:=TRUE;
    ELSIF (#STEP1)
        THEN #STEP3_2:=FALSE;
        END_IF;
// Implementación de salida en los estados con comandos parcialmente complejos para
un almacenado y un despacho.
//Salidas en el estado 1
IF #STEP1 THEN// comando complejo estado 1
    FOR #I:=1 TO 4 DO// Restablecer las coordenadas X y Z de la unidad en el estado 1
        #M_IMS8_X_SENS[#I]:=FALSE;
        #M_IMS8_Z_B[#I]:=FALSE;
    END_FOR;
END_IF;
#M_IMS8_STAD_READY:=#STEP1;// comando sin guardar para
M_IMS8_STAD_READY
//Salidas en el estado 2_1

```

```

IF #STEP2_1 AND #M_IMS8_START THEN // Comando complejo estado 2.1 para
almacenamiento
    // Copie la asignación de espacio de almacenamiento actual, que aún se cambia en el
programa, a la antigua ocupación
    FOR #I:=1 TO 4 DO
        FOR #J:=1 TO 4 DO
            #LEVEL_PLACE_OLD[#I,#J]:=#LEVEL_PLACE[#I,#J]; //Igualar matriz anterior con la
actual
        END_FOR;
    END_FOR;
// Determinar el nivel que se almacenará
    IF (#LEVEL_PLACE_OLD[1,4]=FALSE OR #LEVEL_PLACE_OLD[1,2]=FALSE OR
#LEVEL_PLACE_OLD[1,1]=FALSE OR #LEVEL_PLACE_OLD[1,3]=FALSE )THEN
        #JLEVEL:=1;
        ELIF #LEVEL_PLACE_OLD[1,4]=TRUE AND ( #LEVEL_PLACE_OLD[2,1]=FALSE
OR #LEVEL_PLACE_OLD[2,2]=FALSE OR #LEVEL_PLACE_OLD[2,3]=FALSE OR
#LEVEL_PLACE_OLD[2,4]=FALSE) THEN
            #JLEVEL:=2;
            ELIF #LEVEL_PLACE_OLD[2,4]=TRUE AND ( #LEVEL_PLACE_OLD[3,1]=FALSE
OR #LEVEL_PLACE_OLD[3,2]=FALSE OR #LEVEL_PLACE_OLD[3,3]=FALSE OR
#LEVEL_PLACE_OLD[3,4]=FALSE) THEN
                #JLEVEL:=3;
                ELIF #LEVEL_PLACE_OLD[3,4]=TRUE THEN
                    #JLEVEL:=4;
            END_IF;
        #M_IMS8_Z_B[#JLEVEL]:=TRUE; // Ejecutar la coordenada del eje Z

        IF #LEVEL_PLACE_OLD[#JLEVEL,1]=FALSE // la ubicación de almacenamiento 1 del
nivel está libre
            THEN #LEVEL_PLACE[#JLEVEL,1]:=TRUE; // El contenedor de almacenamiento está
ocupado
                #CONT := #CONT + 1;
                #M_PLACE[#JLEVEL, 1] :=#CONT;
                #M_IMS8_X_SENS[1]:=TRUE; // Coordenada de desplazamiento eje X para el
sensor 1

```

```

END_IF;
FOR #J:=2 TO 4 DO
    IF          (#LEVEL_PLACE_OLD[#JLEVEL,#J]=FALSE          AND
#LEVEL_PLACE_OLD[#JLEVEL,#J-1]=TRUE)
        THEN #LEVEL_PLACE[#JLEVEL,#J]:=TRUE;// El contenedor de almacenamiento
está ocupado
            #CONT := #CONT + 1;
            #M_PLACE[#JLEVEL, #J] := #CONT;
            #M_IMS8_X_SENS[#J]:=TRUE; // Eje X de coordenadas de viaje
        END_IF;
    END_FOR;

//Activar las señales Occupied y Go HRSA cuando termina el almacenado//
IF (#LEVEL_PLACE_OLD[1,1]=TRUE AND #LEVEL_PLACE_OLD[2,1]=TRUE AND
#LEVEL_PLACE_OLD[3,1]=TRUE AND #LEVEL_PLACE_OLD[4,1]=TRUE AND
    #LEVEL_PLACE_OLD[1,2]=TRUE AND #LEVEL_PLACE_OLD[2,2]=TRUE AND
#LEVEL_PLACE_OLD[3,2]=TRUE AND #LEVEL_PLACE_OLD[4,2]=TRUE AND
    #LEVEL_PLACE_OLD[1,3]=TRUE AND #LEVEL_PLACE_OLD[2,3]=TRUE AND
#LEVEL_PLACE_OLD[3,3]=TRUE AND #LEVEL_PLACE_OLD[4,3]=TRUE AND
    #LEVEL_PLACE_OLD[1,4]=TRUE AND #LEVEL_PLACE_OLD[2,4]=TRUE AND
#LEVEL_PLACE_OLD[3,4]=TRUE AND #LEVEL_PLACE_OLD[4,4]=TRUE)
THEN//Poner el campamento ocupado cuando el cuarto nivel esté lleno
    #M_IMS8_STAD_OCCUPIED := TRUE;
ELSE
    #M_IMS8_STAD_OCCUPIED:=FALSE;
END_IF;
IF #M_IMS8_STAD_OCCUPIED =TRUE THEN
    FOR #J:=1 TO 4 DO
        #M_IMS8_X_SENS[#J]:=FALSE; // Cambiar las coordenadas del eje X por
seguridad, a FALSO
        #M_IMS8_Z_B[#J]:=FALSE;//Cambiar las coordenadas del eje Z por seguridad
a FALSO
        #M_IMS8_STAD_GO_HRSA:=FALSE;// comando GO para el restablecimiento
de la unidad de almacenamiento y despacho
    END_FOR;
    ELSIF #M_IMS8_STAD_OCCUPIED=FALSE THEN

```

```

        #M_IMS8_STAD_GO_HRSA:=TRUE; // comando GO para el conjunto de
unidades de almacenamiento y despacho
    END_IF;

END_IF;
////Salidas en el estado 3_1
IF #STEP3_1 AND #M_IMS8_START THEN // Comando complejo en el estado 3.1 para
el despacho
    // Copie la asignación de espacio de almacenamiento actual, que aún se cambia en el
programa, a la antigua ocupación
    FOR #I := 1 TO 4 DO
        FOR #J := 1 TO 4 DO
            #LEVEL_PLACE_OLD[#I,#J] := #LEVEL_PLACE[#I, #J];
        END_FOR;
    END_FOR;
    //Determine el nivel en z y posición en x desde el cual se va despachar según el registro
que determina el que tiene mas tiempo almacenado//
    FOR #I := 1 TO 4 DO
        FOR #J := 1 TO 4 DO
            IF (#M_PLACE[#I, #J] > 0 ) THEN
                #MINIMA := #M_PLACE[#I, #J]; // se asigna una valor de la matriz entero mayor
que cero
            END_IF;
        END_FOR;
    END_FOR;

    FOR #I:=1 TO 4 DO
        FOR #J:=1 TO 4 DO
            IF (#M_PLACE[#I, #J] > 0 AND #M_PLACE[#I, #J] < #MINIMA) THEN // se asegura
que mínima sea el de más tiempo almacenado en comparación con el contado en su valor
mínimo y diferente de cero.
                #MINIMA:= #M_PLACE[#I,#J] ;
            END_IF;
        END_FOR;
    END_FOR;

```

// Según el valor de mínima ya establecido que es el menor número en la matriz que representado que tiene más tiempo almacenado recorre la matriz para apuntar en las coordenadas nivel z y posición x

```
FOR #I:=1 TO 4 DO
  FOR #J:=1 TO 4 DO
    IF (#M_PLACE[#I, #J] = #MINIMA AND #LEVEL_PLACE[#I,#J] = TRUE) THEN
      #M_IMS8_Z_B[#I] := TRUE;
      #M_IMS8_X_SENS[#J] := TRUE;
      #LEVEL_PLACE[#I, #J] := FALSE;
      #M_PLACE[#I, #J] := 0;
    END_IF;
  END_FOR;
END_FOR;
```

//Activar las señales empty y Go HRSA cuando termina el despacho//

```
IF (#LEVEL_PLACE_OLD[1, 1] = TRUE OR #LEVEL_PLACE_OLD[2, 1] = TRUE OR
#LEVEL_PLACE_OLD[3, 1] = TRUE OR #LEVEL_PLACE_OLD[4, 1] = TRUE OR
  #LEVEL_PLACE_OLD[1, 2] = TRUE OR #LEVEL_PLACE_OLD[2, 2] = TRUE OR
#LEVEL_PLACE_OLD[3, 2] = TRUE OR #LEVEL_PLACE_OLD[4, 2] = TRUE OR
  #LEVEL_PLACE_OLD[1, 3] = TRUE OR #LEVEL_PLACE_OLD[2, 3] = TRUE OR
#LEVEL_PLACE_OLD[3, 3] = TRUE OR #LEVEL_PLACE_OLD[4, 3] = TRUE OR
  #LEVEL_PLACE_OLD[1, 4] = TRUE OR #LEVEL_PLACE_OLD[2, 4] = TRUE OR
#LEVEL_PLACE_OLD[3, 4] = TRUE OR #LEVEL_PLACE_OLD[4, 4] = TRUE)
THEN//Poner el campamento ocupado cuando el cuarto nivel esté lleno
  #M_IMS8_STAD_EMPTY := FALSE;
ELSE
  #M_IMS8_STAD_EMPTY := TRUE;
END_IF;
```

```
IF #M_IMS8_STAD_EMPTY=TRUE THEN
  FOR #J:=1 TO 4 DO
    #M_IMS8_X_SENS[#J]:=FALSE; // El viaje coordina el eje X por seguridad,
a FALSO
    #M_IMS8_Z_B[#J]:=FALSE; // Conducir coordenadas del eje Z por
seguridad a FALSO
```

```

        #M_IMS8_STAD_GO_HRSA:=FALSE;// comando GO para el
restablecimiento de la unidad de almacenamiento y despacho a FALSE
        END_FOR;
        ELSIF #M_IMS8_STAD_EMPTY=FALSE THEN
            #M_IMS8_STAD_GO_HRSA:=TRUE;// comando GO para el
restablecimiento de la unidad de almacenamiento y recuperación a TRUE
            END_IF;
        END_IF;
//Salidas en los estados 2_2 y 3_2
// Activa las señales de occupied y empty al recorrer la matriz conjuntamente desactivando
la señal Go HRSA indicando q está listo para un despacho o almacenamiento segun lo
que se requiera
IF #STEP2_2 OR #STEP3_2 THEN // Comandos 2.2 y 3.2: restablecer la señal GO para
la grúa apiladora
    #M_IMS8_STAD_GO_HRSA:=FALSE;

    IF (#LEVEL_PLACE[1, 1] = TRUE AND #LEVEL_PLACE[2, 1] = TRUE AND
#LEVEL_PLACE[3, 1] = TRUE AND #LEVEL_PLACE[4, 1] = TRUE AND
        #LEVEL_PLACE[1, 2] = TRUE AND #LEVEL_PLACE[2, 2] = TRUE AND
#LEVEL_PLACE[3, 2] = TRUE AND #LEVEL_PLACE[4, 2] = TRUE AND
        #LEVEL_PLACE[1, 3] = TRUE AND #LEVEL_PLACE[2, 3] = TRUE AND
#LEVEL_PLACE[3, 3] = TRUE AND #LEVEL_PLACE[4, 3] = TRUE AND
        #LEVEL_PLACE[1, 4] = TRUE AND #LEVEL_PLACE[2, 4] = TRUE AND
#LEVEL_PLACE[3, 4] = TRUE AND #LEVEL_PLACE[4, 4] = TRUE) THEN//Poner el
campamento ocupado cuando el cuarto nivel esté lleno
        #M_IMS8_STAD_OCCUPIED := TRUE;
    ELSE
        #M_IMS8_STAD_OCCUPIED := FALSE;
    END_IF;

    IF (#LEVEL_PLACE[1, 1] = TRUE OR #LEVEL_PLACE[2, 1] = TRUE OR
#LEVEL_PLACE[3, 1] = TRUE OR #LEVEL_PLACE[4, 1] = TRUE OR
        #LEVEL_PLACE[1, 2] = TRUE OR #LEVEL_PLACE[2, 2] = TRUE OR
#LEVEL_PLACE[3, 2] = TRUE OR #LEVEL_PLACE[4, 2] = TRUE OR
        #LEVEL_PLACE[1, 3] = TRUE OR #LEVEL_PLACE[2, 3] = TRUE OR
#LEVEL_PLACE[3, 3] = TRUE OR #LEVEL_PLACE[4, 3] = TRUE OR

```

```
#LEVEL_PLACE[1, 4] = TRUE OR #LEVEL_PLACE[2, 4] = TRUE OR  
#LEVEL_PLACE[3, 4] = TRUE OR #LEVEL_PLACE[4, 4] = TRUE) THEN// Poner el  
campamento ocupado cuando el cuarto nivel esté lleno
```

```
#M_IMS8_STAD_EMPTY := FALSE;  
ELSE  
#M_IMS8_STAD_EMPTY := TRUE;  
END_IF;
```

```
END_IF;
```

```
// Asignación de las variables de salida de las coordenadas según nivel z y posición x en  
el almacenado y despacho según el bloque STAD genere en las condiciones de transición  
y en las salidas de los estados
```

```
#M_IMS8_STAD_X_SENS1:=#M_IMS8_X_SENS[1];  
#M_IMS8_STAD_X_SENS2:=#M_IMS8_X_SENS[2];  
#M_IMS8_STAD_X_SENS3:=#M_IMS8_X_SENS[3];  
#M_IMS8_STAD_X_SENS4:=#M_IMS8_X_SENS[4];  
#M_IMS8_STAD_Z_B1:=#M_IMS8_Z_B[1];  
#M_IMS8_STAD_Z_B2:=#M_IMS8_Z_B[2];  
#M_IMS8_STAD_Z_B3:=#M_IMS8_Z_B[3];  
#M_IMS8_STAD_Z_B4:=#M_IMS8_Z_B[4];
```

**f) FB\_IMS8\_HMI.**

```
#"L1.1" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS1 &  
"I_IMS8_B6";  
#"L1.2" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS2 &  
"I_IMS8_B6";  
#"L1.3" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS3 &  
"I_IMS8_B6";  
#"L1.4" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS4 &  
"I_IMS8_B6";  
#"L2.1" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS1 &  
"I_IMS8_B6";  
#"L2.2" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS2 &  
"I_IMS8_B6";
```

```

#"L2.3" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B6";
#"L2.4" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B6";
#"L3.1" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B6";
#"L3.2" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B6";
#"L3.3" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B6";
#"L3.4" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B6";
#"L4.1" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B6";
#"L4.2" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B6";
#"L4.3" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B6";
#"L4.4" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B6";
#"H1.1" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B7";
#"H1.2" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B7";
#"H1.3" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B7";
#"H1.4" := "DB66".M_IMS8_STAD_Z_B1 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B7";
#"H2.1" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B7";
#"H2.2" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B7";
#"H2.3" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B7";
#"H2.4" := "DB66".M_IMS8_STAD_Z_B2 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B7";

```



```

#"H3.1" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B7";
#"H3.2" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B7";
#"H3.3" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B7";
#"H3.4" := "DB66".M_IMS8_STAD_Z_B3 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B7";
#"H4.1" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS1 &
"I_IMS8_B7";
#"H4.2" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS2 &
"I_IMS8_B7";
#"H4.3" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS3 &
"I_IMS8_B7";
#"H4.4" := "DB66".M_IMS8_STAD_Z_B4 & "DB66".M_IMS8_STAD_X_SENS4 &
"I_IMS8_B7";

```

```

IF ("DB66".I_IMS8_INIT = TRUE) THEN

```

```

    #sixpack1 := FALSE;
    #sixpack2 := FALSE;
    #sixpack3 := FALSE;
    #sixpack4 := FALSE;
    #sixpack5 := FALSE;
    #sixpack6 := FALSE;
    #sixpack7 := FALSE;
    #sixpack8 := FALSE;
    #sixpack9 := FALSE;
    #sixpack10 := FALSE;
    #sixpack11 := FALSE;
    #sixpack12 := FALSE;
    #sixpack13 := FALSE;
    #sixpack14 := FALSE;
    #sixpack15 := FALSE;
    #sixpack16 := FALSE;

```

```

END_IF;

```

```

#sixpack1 := (#"H1.1" OR #sixpack1) AND ( NOT (#"H1.1") OR
"DB66".LEVEL_PLACE[1,1]);
#sixpack2 := (#"H1.2" OR #sixpack2) AND (NOT (#"H1.2") OR "DB66".LEVEL_PLACE[1,
2]);
#sixpack3 := (#"H1.3" OR #sixpack3) AND (NOT (#"H1.3") OR "DB66".LEVEL_PLACE[1,
3]);
#sixpack4 := (#"H1.4" OR #sixpack4) AND (NOT (#"H1.4") OR "DB66".LEVEL_PLACE[1,
4]);
#sixpack5 := (#"H2.1" OR #sixpack5) AND (NOT (#"H2.1") OR "DB66".LEVEL_PLACE[2,
1]);
#sixpack6 := (#"H2.2" OR #sixpack6) AND (NOT (#"H2.2") OR "DB66".LEVEL_PLACE[2,
2]);
#sixpack7 := (#"H2.3" OR #sixpack7) AND (NOT (#"H2.3") OR "DB66".LEVEL_PLACE[2,
3]);
#sixpack8 := (#"H2.4" OR #sixpack8) AND (NOT (#"H2.4") OR "DB66".LEVEL_PLACE[2,
4]);
#sixpack9 := (#"H3.1" OR #sixpack9) AND (NOT (#"H3.1") OR "DB66".LEVEL_PLACE[3,
1]);
#sixpack10 := (#"H3.2" OR #sixpack10) AND (NOT (#"H3.2") OR
"DB66".LEVEL_PLACE[3, 2]);
#sixpack11 := (#"H3.3" OR #sixpack11) AND (NOT (#"H3.3") OR
"DB66".LEVEL_PLACE[3, 3]);
#sixpack12 := (#"H3.4" OR #sixpack12) AND (NOT (#"H3.4") OR
"DB66".LEVEL_PLACE[3, 4]);
#sixpack13 := (#"H4.1" OR #sixpack13) AND (NOT (#"H4.1") OR
"DB66".LEVEL_PLACE[4, 1]);
#sixpack14 := (#"H4.2" OR #sixpack14) AND (NOT (#"H4.2") OR
"DB66".LEVEL_PLACE[4, 2]);
#sixpack15 := (#"H4.3" OR #sixpack15) AND (NOT (#"H4.3") OR
"DB66".LEVEL_PLACE[4, 3]);
#sixpack16 := (#"H4.4" OR #sixpack16) AND (NOT (#"H4.4") OR
"DB66".LEVEL_PLACE[4, 4]);
#inpallets := ("DB66".M_IMS8_STORE OR "DB66".M_IMS8_SWAP);

```

**g) COMPLETE RESTART.**

```
"DB68".sixpack1 := false;
```

```
"DB68".sixpack2 := false;  
"DB68".sixpack3 := false;  
"DB68".sixpack4 := false;  
"DB68".sixpack5 := false;  
"DB68".sixpack6 := false;  
"DB68".sixpack7 := false;  
"DB68".sixpack8 := false;  
"DB68".sixpack9 := false;  
"DB68".sixpack10 := false;  
"DB68".sixpack11 := false;  
"DB68".sixpack12 := false;  
"DB68".sixpack13 := false;  
"DB68".sixpack14 := false;  
"DB68".sixpack15 := false;  
"DB68".sixpack16 := false;
```