



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería Electrónica Industrial y Automática**

**Actualización, diseño y programación de  
varias estaciones de trabajo mediante el uso  
de PLC SIEMENS y softwares específicos**

**Autor:**

**Abril Arroyo, Jorge**

**Tutor:**

**Moya de la Torre, Eduardo Julio  
Departamento de Ingeniería de  
Sistemas y Automática**

**Valladolid, 12 de febrero de 2025.**



Universidad de Valladolid

Actualización, diseño y programación de varias estaciones de trabajo mediante el uso de PLC SIEMENS y softwares específicos



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## Resumen

El objetivo de este proyecto es la actualización y desarrollo de varias prácticas de laboratorio relacionadas con la automatización industrial y enfocadas a fines docentes en grados y másteres industriales de la Universidad de Valladolid.

Las prácticas tienen como finalidad el aprendizaje y aplicación de los conocimientos relacionados con la programación y simulación de controladores lógicos programables (PLC).

Se emplearán varios softwares propios de SIEMENS; TIA Portal, para la programación secuencial y PLCSIM y PLCSIM Advanced, como simuladores del autómatas. Por otra parte, para simular los gemelos digitales de las estaciones a programar, se hará uso de Factory IO y SIMIT.

El conjunto de programas permite reproducir un entorno industrial completo de forma virtual y los softwares para programarlo.

De tal forma, el proyecto dotará a los usuarios de las capacidades necesarias para automatizar diferentes procesos industriales, empleando interfaces hombre máquina (HMI), controladores PID y varios tipos de robots.

### **Palabras clave:**

PLC, PLCSIM, PLCSIM Advanced, TIA Portal, Factory IO.



Universidad de Valladolid

Actualización, diseño y programación de varias estaciones de trabajo mediante el uso de PLC SIEMENS y softwares específicos



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## Abstract

The aim of this project is the updating and development of several laboratory practices related to industrial automation and focused on teaching purposes in industrial degrees and master's at the University of Valladolid.

The practices are meant to learn and apply knowledge related to the programming and simulation of programmable logic controllers (PLC).

Several SIEMENS own softwares will be used; TIA Portal, for sequential programming and PLCSIM and PLCSIM Advanced, as PLC simulators. On the other hand, Factory IO and SIMIT will be used to simulate the digital twins of the stations to be programmed.

The set of programs allows to reproduce a complete industrial environment virtually and the software to program it.

In consequence, the project will provide users with the necessary capabilities to automate different industrial processes, using human-machine interfaces (HMI), PID controllers and several types of robots.

### **Keywords:**

PLC, PLCSIM, PLCSIM Advanced, TIA Portal, Factory IO.



Universidad de Valladolid

Actualización, diseño y programación de varias estaciones de trabajo mediante el uso de PLC SIEMENS y softwares específicos



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## Índice de contenido

<b>1. INTRODUCCIÓN Y OBJETIVOS .....</b>	<b>15</b>
1.1. JUSTIFICACIÓN DEL PROYECTO .....	15
1.2. ALCANCE .....	15
1.3. OBJETIVOS .....	16
1.4. ESTRUCTURA DEL PROYECTO .....	17
<b>2. FUNDAMENTOS TEÓRICOS .....</b>	<b>19</b>
2.1. GRAFCET .....	19
2.1.1 <i>Definición de GRAFCET</i> .....	19
2.1.2 <i>Elementos del GRAFCET</i> .....	19
2.1.3 <i>Reglas del GRAFCET</i> .....	20
2.1.4 <i>Estructuras de secuencias del GRAFCET</i> .....	20
2.1.5 <i>Niveles del GRAFCET</i> .....	21
2.2. DIRECCIONAMIENTO IP .....	22
2.3. PLC .....	23
2.3.1 <i>Descripción del PLC</i> .....	23
2.3.2 <i>Estructura del PLC</i> .....	24
2.3.3 <i>Entradas y salidas</i> .....	26
2.3.4 <i>Programación de los autómatas</i> .....	26
2.3.5 <i>Aplicaciones, ventajas e inconvenientes de los autómatas</i> .....	31
2.3.6 <i>Hardware de SIEMENS</i> .....	32
2.4. PID .....	33
2.4.1 <i>Control en lazo cerrado</i> .....	33
2.4.1 <i>Objetivos del controlador</i> .....	34
2.4.2 <i>Acciones del controlador PID</i> .....	34
2.4.3 <i>Funcionamiento del controlador PID</i> .....	35
<b>3. SOFTWARES DE PROGRAMACIÓN Y SIMULACIÓN .....</b>	<b>37</b>
3.1. SELECCIÓN DEL SOFTWARE .....	37
3.2. TIA PORTAL V17 .....	40
3.2.1 <i>Entorno de la interfaz</i> .....	40
3.2.2 <i>Lenguajes de programación</i> .....	42
3.2.3 <i>Variables y tipos de datos</i> .....	42
3.2.4 <i>Bloques de programa</i> .....	43
3.2.5 <i>Seguridad Integrada</i> .....	44
3.3. PLCSIM Y PLCSIM ADVANCED .....	44
3.4. FACTORY IO Y SIMIT .....	45
3.5. WINCC .....	49
3.6. CONECTIVIDAD ENTRE PROGRAMAS .....	50
<b>4. ACTUALIZACIÓN DE ESTACIONES .....</b>	<b>51</b>
4.1. ACTUALIZACIÓN DE ARCHIVOS .....	51
4.2. DESCARGA DE LA PLANTILLA PARA SIMULAR CON FACTORY IO Y PLCSIM .....	53
4.2.1 <i>Copiar y pegar bloques de programa</i> .....	54
4.3. CONFIGURACIÓN DEL CONTROLADOR DE FACTORY IO .....	54
4.3.1 <i>Cambio de variables en Factory IO y TIA Portal</i> .....	55
4.4. ESTACIÓN 1: ESTACIÓN PALETIZADORA .....	58
4.4.1 <i>Objetivos de aprendizaje</i> .....	58
4.4.2 <i>Descripción de la estación</i> .....	59
4.4.3 <i>Corrección de errores</i> .....	60
4.5. ESTACIÓN 2: ESTACIÓN DISTRIBUIDORA .....	61



4.5.1	Objetivos de aprendizaje .....	61
4.5.2	Descripción de la estación .....	62
4.5.3	Corrección de errores.....	64
4.6.	ESTACIÓN 3: DEPÓSITO DE LÍQUIDO .....	65
4.6.1	Objetivos de aprendizaje .....	65
4.6.2	Descripción de la estación .....	66
4.6.3	Corrección de errores.....	67
4.7.	ESTACIÓN 4: GESTIÓN DE STOCK DE UN ALMACÉN.....	70
4.7.1	Objetivos de aprendizaje .....	70
4.7.2	Descripción de la estación .....	70
<b>5.</b>	<b>ESTACIONES DE SIMULACIÓN DISEÑADAS .....</b>	<b>75</b>
5.1.	ESTACIÓN 5: CINTA TRANSPORTADORA.....	75
5.1.1	Objetivo de la simulación.....	75
5.1.2	Descripción de la escena.....	76
5.1.3	Descripción GRAFCET de la programación.....	78
5.1.4	Programación del autómatas.....	80
5.1.5	Modificación con PLC S7-1500 y Factory IO.....	85
5.1.6	Modificación con PLC S7-1500 y SIMIT .....	88
5.2.	ESTACIÓN 6: CALCULADORA .....	91
5.2.1	Objetivo de la simulación.....	91
5.2.2	Descripción de la escena.....	91
5.2.3	Programación del autómatas.....	94
5.2.4	Programación del HMI.....	98
5.3.	ESTACIÓN 7: CLASIFICADORA .....	100
5.3.1	Objetivo de la simulación.....	100
5.3.2	Descripción de la escena.....	101
5.3.3	Descripción GRAFCET de la programación.....	103
5.3.4	Programación del autómatas.....	105
5.4.	ESTACIÓN 8: COLOCADOR POR PESO .....	111
5.4.1	Objetivo de la simulación.....	111
5.4.2	Descripción de la escena.....	111
5.4.3	Descripción GRAFCET de la programación.....	115
5.4.4	Programación del autómatas.....	118
5.5.	ESTACIÓN 9: PRODUCCIÓN EN LÍNEA CON SEGURIDAD .....	128
5.5.1	Objetivo de la simulación.....	128
5.5.2	Descripción de la escena.....	129
5.5.3	Programación del autómatas.....	136
5.5.4	Programación del HMI.....	141
<b>6.</b>	<b>INTEGRACIÓN DE LAS PRÁCTICAS EN LA DOCENCIA.....</b>	<b>147</b>
<b>7.</b>	<b>ESTUDIO ECONÓMICO .....</b>	<b>149</b>
<b>8.</b>	<b>CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>151</b>
8.1.	CONCLUSIONES .....	151
8.2.	LÍNEAS FUTURAS.....	153
<b>9.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>155</b>
<b>10.</b>	<b>ANEXO 1: CONFIGURACIÓN DEL SOFTWARE</b>	
<b>11.</b>	<b>ANEXO 2: ELEMENTOS BÁSICOS DE FACTORY IO</b>	
<b>12.</b>	<b>ANEXO 3: PROGRAMAS DE TIA PORTAL</b>	



## Índice de figuras

FIGURA 1.	ESTRUCTURA LINEAL DEL GRAFCET .....	20
FIGURA 2.	ESTRUCTURA DE SALTO HACIA ATRÁS DEL GRAFCET .....	21
FIGURA 3.	ESTRUCTURA INTERNA DE UN AUTÓMATA .....	25
FIGURA 4.	PLC MODULAR SIEMENS S7-1200 .....	25
FIGURA 5.	HARDWARE Y SOFTWARE DE PROGRAMACIÓN DE LAS ESTACIONES .....	27
FIGURA 6.	FORMATO ESTANDARIZADO DE VARIABLES .....	28
FIGURA 7.	EJEMPLOS DE DIAGRAMAS LADDER .....	30
FIGURA 8.	HMI TP700 COMFORT DE SIEMENS .....	30
FIGURA 9.	GAMAS DE CONTROLADORES SIMATIC .....	33
FIGURA 10.	CONTROL EN LAZO CERRADO .....	33
FIGURA 11.	SINTONIZACIÓN DE UN PID CON EL MÉTODO DE PRUEBA Y ERROR .....	36
FIGURA 12.	PAQUETES CON TIA PORTAL V17 .....	37
FIGURA 13.	VISTA DE PORTAL DE TIA PORTAL .....	40
FIGURA 14.	VISTA DE PROYECTO DE TIA PORTAL .....	41
FIGURA 15.	DETALLE BARRA DE HERRAMIENTAS DE TIA PORTAL .....	41
FIGURA 16.	EJEMPLO ESTRUCTURA DE BLOQUES DE PROGRAMA EN TIA PORTAL .....	43
FIGURA 17.	SIMULACIÓN DE UN PLC CON PLCSIM.....	44
FIGURA 18.	SIMULACIÓN DE DOS PLC'S CON PLCSIM ADVANCED .....	45
FIGURA 19.	INTERFAZ DE FACTORY IO .....	46
FIGURA 20.	CONFIGURACIÓN DEL CONTROLADOR E/S DE FACTORY IO.....	47
FIGURA 21.	VISTA DEL CONTROLADOR E/S EN FACTORY IO .....	47
FIGURA 22.	INTERFAZ DE SIMIT .....	48
FIGURA 23.	HMI SIMULADO EN WINCC .....	49
FIGURA 24.	ABRIR UN PROYECTO EN TIA PORTAL .....	51
FIGURA 25.	VENTANA DE AVISO DE ACTUALIZACIÓN DE VERSIÓN EN TIA PORTAL .....	52
FIGURA 26.	VENTANA DE FINALIZACIÓN DE LA ACTUALIZACIÓN .....	52
FIGURA 27.	DESCARGA DE LA PLANTILLA PROPORCIONADA POR FACTORY IO.....	53
FIGURA 28.	SELECCIONADOR DEL DRIVER DE FACTORY IO .....	54
FIGURA 29.	CONFIGURACIÓN DEL CONTROLADOR DE FACTORY IO.....	55
FIGURA 30.	PROPIEDADES DEL CONTROLADOR DE TIA PORTAL .....	56
FIGURA 31.	TABLA DE VARIABLES DEL PLC .....	57
FIGURA 32.	DETALLE DE INFORMACIÓN ADICIONAL SOBRE LAS DIRECCIONES.....	57
FIGURA 33.	VISTA GENERAL. ESCENA PALETIZADORA .....	59
FIGURA 34.	DISTRIBUCIÓN DE BLOQUES. ESTACIÓN PALETIZADORA .....	60
FIGURA 35.	MODIFICACIÓN SEGMENTO 4 MAIN. ESTACIÓN PALETIZADORA .....	61



**FIGURA 36. VISTA DESDE ARRIBA. ESCENA DISTRIBUIDORA..... 62**

**FIGURA 37. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN DISTRIBUIDORA ..... 63**

**FIGURA 38. MODIFICACIÓN FC CINTA GENERADORA DE PIEZAS. ESTACIÓN DISTRIBUIDORA ..... 64**

**FIGURA 39. MODIFICACIÓN FC CINTA CAJAS. ESTACIÓN DISTRIBUIDORA ..... 64**

**FIGURA 40. MODIFICACIÓN CONTADOR EN EL MAIN. ESTACIÓN DISTRIBUIDORA ..... 65**

**FIGURA 41. VISTA GENERAL. ESCENA DEPÓSITO DE LÍQUIDO..... 66**

**FIGURA 42. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN DEPÓSITO DE LÍQUIDO ..... 67**

**FIGURA 43. BLOQUE “DEPOSITO” CREADO. ESTACIÓN DEPÓSITO DE LÍQUIDO ..... 68**

**FIGURA 44. COMANDO PID\_COMPACT. ESTACIÓN DEPÓSITO DE LÍQUIDO ..... 68**

**FIGURA 45. CONFIGURACIÓN PID\_COMPACT DE VACIADO. ESCENA DEPÓSITO DE LÍQUIDO ..... 69**

**FIGURA 46. VENTANA DE PUESTA EN SERVICIO DEL PID. ESCENA DEPÓSITO DE LÍQUIDO ..... 69**

**FIGURA 47. VISTA GENERAL. ESCENA GESTIÓN DE STOCK DE UN ALMACÉN ..... 71**

**FIGURA 48. VISTA DISPOSITIVOS Y REDES. ESTACIÓN GESTIÓN DE STOCK DE UN ALMACÉN ..... 72**

**FIGURA 49. DISTRIBUCIÓN DE IMÁGENES. ESTACIÓN GESTIÓN DE STOCK DE UN ALMACÉN ..... 72**

**FIGURA 50. PANTALLAS DEL HMI. ESTACIÓN GESTIÓN DE STOCK DE UN ALMACÉN ..... 73**

**FIGURA 51. PANTALLAS DEL HMI. ESTACIÓN GESTIÓN DE STOCK DE UN ALMACÉN ..... 74**

**FIGURA 52. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN GESTIÓN DE STOCK DE UN ALMACÉN ..... 74**

**FIGURA 53. VISTA GENERAL. ESCENA CINTA TRANSPORTADORA ..... 77**

**FIGURA 54. VISTA CUADRO ELÉCTRICO. ESCENA CINTA TRANSPORTADORA ..... 77**

**FIGURA 55. DISTRIBUCIÓN DE VARIABLES. ESCENA CINTA TRANSPORTADORA ..... 78**

**FIGURA 56. GRAFCET CINTA 1. ESTACIÓN CINTA TRANSPORTADORA ..... 79**

**FIGURA 57. GRAFCET CINTA 2. ESTACIÓN CINTA TRANSPORTADORA ..... 79**

**FIGURA 58. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN CINTA TRANSPORTADORA ..... 81**

**FIGURA 59. DISTRIBUCIÓN DE SEGMENTOS MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 81**

**FIGURA 60. SEGMENTO 1 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 82**

**FIGURA 61. SEGMENTO 2 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 82**

**FIGURA 62. SEGMENTO 3 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 83**

**FIGURA 63. SEGMENTO 4 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 83**

**FIGURA 64. SEGMENTO 5 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 84**

**FIGURA 65. SEGMENTO 6 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 84**

**FIGURA 66. SEGMENTO 7 MAIN. ESTACIÓN CINTA TRANSPORTADORA ..... 85**

**FIGURA 67. SIMULADOR DEL PLC 1200. ESTACIÓN CINTA TRANSPORTADORA ..... 85**

**FIGURA 68. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN CINTA TRANSPORTADORA 1500 ..... 86**

**FIGURA 69. SIMULADOR PLC 1500. ESTACIÓN CINTA TRANSPORTADORA 1500 ..... 86**

**FIGURA 70. CONFIGURACIÓN FACTORY IO. ESCENA CINTA TRANSPORTADORA 1500 ..... 87**

**FIGURA 71. VARIABLES SIMIT. ESTACIÓN CINTA TRANSPORTADORA 1500 ..... 88**



**FIGURA 72. CHART DE ANIMACIONES. ESTACIÓN CINTA TRANSPORTADORA 1500 ..... 89**

**FIGURA 73. CHART DE CINTAS. ESTACIÓN CINTA TRANSPORTADORA 1500 ..... 89**

**FIGURA 74. ANIMACIÓN CAJA CHART DE CINTAS. ESTACIÓN CINTA TRANSPORTADORA 1500 ... 90**

**FIGURA 75. IMAGEN RAÍZ. ESTACIÓN CALCULADORA ..... 92**

**FIGURA 76. IMAGEN MATEMÁTICAS. ESTACIÓN CALCULADORA..... 92**

**FIGURA 77. IMAGEN COMPROBACIÓN. ESTACIÓN CALCULADORA ..... 93**

**FIGURA 78. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN CALCULADORA ..... 95**

**FIGURA 79. VARIABLES DB1 “OPERANDOS Y RESULTADO”. ESTACIÓN CALCULADORA ..... 96**

**FIGURA 80. OB DE ARRANQUE. ESTACIÓN CALCULADORA ..... 96**

**FIGURA 81. BLOQUE MAIN. ESTACIÓN CALCULADORA..... 97**

**FIGURA 82. FC1 “MATEMÁTICAS”. ESTACIÓN CALCULADORA ..... 97**

**FIGURA 83. FC’S CONTENIDAS EN FC2 “COMPROBACIÓN”. ESTACIÓN CALCULADORA..... 98**

**FIGURA 84. ÁRBOL DE PROYECTO DEL HMI. ESTACIÓN CALCULADORA ..... 99**

**FIGURA 85. CONEXIÓN ETHERNET ENTRE EL HMI Y EL PLC. ESTACIÓN CALCULADORA ..... 99**

**FIGURA 86. VISTA GENERAL. ESCENA CLASIFICADORA ..... 102**

**FIGURA 87. VISTA CUADRO ELÉCTRICO. ESCENA CLASIFICADORA ..... 102**

**FIGURA 88. DISTRIBUCIÓN DE VARIABLES. ESCENA CLASIFICADORA..... 103**

**FIGURA 89. GRAFCET CINTA GENERADORA. ESTACIÓN CLASIFICADORA ..... 104**

**FIGURA 90. GRAFCET CINTA DISTRIBUIDORA. ESTACIÓN CLASIFICADORA ..... 104**

**FIGURA 91. GRAFCET BARRERAS DE LA CINTA DISTRIBUIDORA. ESTACIÓN CLASIFICADORA ..... 105**

**FIGURA 92. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN CLASIFICADORA ..... 107**

**FIGURA 93. FC9000 CONEXIÓN CON FACTORY IO. ESTACIÓN CLASIFICADORA ..... 107**

**FIGURA 94. FC10 GESTIÓN DE MARCHA. ESTACIÓN CLASIFICADORA ..... 108**

**FIGURA 95. FC20 GENERAR PIEZA. ESTACIÓN CLASIFICADORA ..... 108**

**FIGURA 96. FC110 CINTA GENERADORA. ESTACIÓN CLASIFICADORA ..... 108**

**FIGURA 97. FC120 CINTA DISTRIBUIDORA. ESTACIÓN CLASIFICADORA ..... 109**

**FIGURA 98. FC130 GESTIÓN BARRERAS DE EMPUJE. ESTACIÓN CLASIFICADORA..... 109**

**FIGURA 99. FB130 BARRERA DISTRIBUIDOR BASES AZULES. ESTACIÓN CLASIFICADORA..... 110**

**FIGURA 100. FC140 CONTADOR DE PIEZAS. ESTACIÓN CLASIFICADORA ..... 110**

**FIGURA 101. VISTA GENERAL. ESCENA COLOCADOR POR PESO..... 113**

**FIGURA 102. VISTA CUADRO ELÉCTRICO. ESCENA COLOCADOR POR PESO..... 113**

**FIGURA 103. DISTRIBUCIÓN DE VARIABLES. ESCENA COLOCADOR POR PESO ..... 114**

**FIGURA 104. GRAFCET CINTA CAJAS MEDIANAS ..... 115**

**FIGURA 105. GRAFCET ROBOT CARTESIANO ..... 116**

**FIGURA 106. GRAFCET ROBOT DE DOS EJES ..... 117**

**FIGURA 107. GRAFCET TRANSPORTADOR DE RODILLOS..... 118**



**FIGURA 108. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN COLOCADOR POR PESO ..... 120**

**FIGURA 109. FB30 0.CINTA ENTRADA Y BÁSCULA. ESTACIÓN COLOCADOR POR PESO ..... 121**

**FIGURA 110. FB40 0.DISTRIBUIDORA. ESTACIÓN COLOCADOR POR PESO ..... 122**

**FIGURA 111. FC110 2.CINTA IZQUIERDA CAJAS MEDIANAS. ESTACIÓN COLOCADOR POR PESO .123**

**FIGURA 112. FB20 0.CINTAS TRANSPORTADORAS. ESTACIÓN COLOCADOR POR PESO .....123**

**FIGURA 113. FB120 0.TRANSPORTADOR RODILLOS. ESTACIÓN COLOCADOR POR PESO ..... 124**

**FIGURA 114. FB130 2.CARTESIANO. ESTACIÓN COLOCADOR POR PESO .....125**

**FIGURA 115. FC130 2.POSICIÓN CARTESIANO. ESTACIÓN COLOCADOR POR PESO ..... 126**

**FIGURA 116. FC131 2.POSICIÓN CAJAS MEDIANAS. ESTACIÓN COLOCADOR POR PESO ..... 126**

**FIGURA 117. FB140 1.ROBOT DOS EJES XZ. ESTACIÓN COLOCADOR POR PESO .....127**

**FIGURA 118. FC142 1.POSICIÓN ROBOTXZ. ESTACIÓN COLOCADOR POR PESO ..... 128**

**FIGURA 119. VISTA GENERAL. ESCENA PRODUCCIÓN EN LÍNEA ..... 130**

**FIGURA 120. VISTA CUADRO ELÉCTRICO CENTRO TAPAS. ESCENA PRODUCCIÓN EN LÍNEA ..... 131**

**FIGURA 121. VISTA CUADRO ELÉCTRICO PRINCIPAL. ESCENA PRODUCCIÓN EN LÍNEA ..... 132**

**FIGURA 122. IMAGEN RAÍZ. ESTACIÓN PRODUCCIÓN EN LÍNEA .....133**

**FIGURA 123. IMAGEN EMERGENTE USUARIOS. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 133**

**FIGURA 124. IMAGEN CENTRO BASES. ESTACIÓN PRODUCCIÓN EN LÍNEA..... 134**

**FIGURA 125. IMAGEN CENTRO TAPAS. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 134**

**FIGURA 126. DISTRIBUCIÓN DE VARIABLES. ESCENA PRODUCCIÓN EN LÍNEA ..... 135**

**FIGURA 127. DISTRIBUCIÓN DE BLOQUES. ESTACIÓN PRODUCCIÓN EN LÍNEA.....137**

**FIGURA 128. PROTECCIÓN KNOW-HOW. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 138**

**FIGURA 129. FC1 0.GESTIÓN SEGURIDAD. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 138**

**FIGURA 130. FC10 0.GESTIÓN MARCHA. ESTACIÓN PRODUCCIÓN EN LÍNEA..... 138**

**FIGURA 131. FB4 GESTIÓN PRIORIDAD SALIDA. ESTACIÓN PRODUCCIÓN EN LÍNEA..... 139**

**FIGURA 132. FB3 CINTAS DE ENTRADA DE LAS BASES. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 139**

**FIGURA 133. FB1 CENTROS DE TRATAMIENTO DE BASES. ESTACIÓN PRODUCCIÓN EN LÍNEA.....140**

**FIGURA 134. FB2 CONTADORES CENTROS MECANIZADO. ESTACIÓN PRODUCCIÓN EN LÍNEA .... 140**

**FIGURA 135. FB5 CINTA DE SALIDA. ESTACIÓN PRODUCCIÓN EN LÍNEA.....141**

**FIGURA 136. FB6 GESTIÓN MOTORES. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 141**

**FIGURA 137. ÁRBOL DE PROYECTO DEL HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA..... 142**

**FIGURA 138. CONEXIÓN ETHERNET ENTRE HMI Y PLC. ESTACIÓN PRODUCCIÓN EN LÍNEA .....143**

**FIGURA 139. ADMINISTRACIÓN DE USUARIOS HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 144**

**FIGURA 140. PERMISOS DE USUARIOS HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 145**

**FIGURA 141. PROPIEDADES DEL BOTÓN “USUARIOS”. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 145**

**FIGURA 142. PLANIFICADOR DE TAREAS HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA ..... 146**

**FIGURA 143. DETALLE DE USUARIO CONECTADO DEL HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA ..146**



## Índice de tablas

<b>TABLA 1.</b>	<b>INSTRUCCIONES BÁSICAS EN LENGUAJE LADDER .....</b>	<b>29</b>
<b>TABLA 2.</b>	<b>COMPARACIÓN DEL SOFTWARE SELECCIONADO .....</b>	<b>39</b>
<b>TABLA 3.</b>	<b>TIPOS DE DATOS TIA PORTAL .....</b>	<b>42</b>
<b>TABLA 4.</b>	<b>VARIABLES DEL PLC. ESTACIÓN CINTA TRANSPORTADORA .....</b>	<b>80</b>
<b>TABLA 5.</b>	<b>VARIABLES DEL PLC. ESTACIÓN CALCULADORA .....</b>	<b>94</b>
<b>TABLA 6.</b>	<b>VARIABLES DEL HMI. ESTACIÓN CALCULADORA.....</b>	<b>100</b>
<b>TABLA 7.</b>	<b>VARIABLES DEL PLC. ESTACIÓN CLASIFICADORA .....</b>	<b>106</b>
<b>TABLA 8.</b>	<b>VARIABLES DEL PLC. ESTACIÓN COLOCADOR POR PESO.....</b>	<b>119</b>
<b>TABLA 9.</b>	<b>VARIABLES DEL PLC. ESTACIÓN PRODUCCIÓN EN LÍNEA .....</b>	<b>136</b>
<b>TABLA 10.</b>	<b>VARIABLES ESTÁNDAR DEL HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA .....</b>	<b>143</b>
<b>TABLA 11.</b>	<b>VARIABLES DE MOTORES DEL HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA .....</b>	<b>143</b>
<b>TABLA 12.</b>	<b>VARIABLES INTERNAS DEL HMI. ESTACIÓN PRODUCCIÓN EN LÍNEA.....</b>	<b>144</b>
<b>TABLA 13.</b>	<b>OBJETIVOS DE APRENDIZAJE CUMPLIDOS SEGÚN LAS GUÍAS DOCENTES .....</b>	<b>147</b>
<b>TABLA 14.</b>	<b>CONVENIENCIA DE LAS SIMULACIONES SEGÚN LOS GRADOS Y MÁSTERES .....</b>	<b>148</b>
<b>TABLA 15.</b>	<b>COSTE DE PERSONAL.....</b>	<b>149</b>
<b>TABLA 16.</b>	<b>COSTES DE MATERIAL .....</b>	<b>150</b>
<b>TABLA 17.</b>	<b>COSTES ENERGÉTICOS Y DE CONEXIÓN .....</b>	<b>150</b>
<b>TABLA 18.</b>	<b>COSTES TOTALES .....</b>	<b>150</b>





# 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1. JUSTIFICACIÓN DEL PROYECTO

El proyecto pretende actualizar y crear varias simulaciones de estaciones industriales controladas por autómatas. Las estaciones están diseñadas para el uso docente en grados y másteres industriales de la Universidad de Valladolid (UVa).

Mediante la programación, simulación y actualización de programas se persigue el aprendizaje y aplicación de conocimientos relacionados con el campo de la automatización industrial.

La simulación aporta la ventaja de no tener que disponer de los elementos físicos de la estación ni del PLC y poder realizar estas simulaciones de forma remota y no presencial mediante el uso de gemelos digitales.

Con el fin de proporcionar el máximo número de conocimientos y aplicaciones relacionadas con el mundo industrial, y en particular con la automatización de procesos, las simulaciones estarán compuestas por todos los elementos que podríamos encontrar en un entorno industrial:

- El **controlador lógico programable** (PLC o autómata), simulado con los softwares PLCSIM y PLCSIM Advanced.
- El **software de programación de autómatas e interfaces hombre-máquina** (HMI o panel de operador), TIA Portal V17.
- El entorno virtual de la **estación real**, que posibilita la simulación e interacción con el proceso industrial a controlar, creado con los programas Factory IO y SIMIT.

Las estaciones industriales contienen robots, HMI's y controladores PID, dispositivos muy empleados en los procesos automatizados, dotando de gran versatilidad al usuario a la hora de aprender la gran mayoría de elementos utilizados en el sector de la automatización de procesos.

Es por ello, que el usuario será capaz, no solo de programar y controlar una estación automatizada, sino también de realizar todo el proceso de conexión entre los diferentes elementos que componen el entorno automatizado.

Puesto que en la industria se emplean gran variedad de softwares de simulación y programación enfocados a la automatización, las simulaciones se han confeccionado para hacer uso de dos programas diferentes de simulación de autómatas y otros dos simuladores de entornos industriales.

## 1.2. ALCANCE

El proyecto se podría dividir en dos grandes bloques. El primero, y menos extenso, se encarga de recuperar unas simulaciones realizadas en la versión TIA Portal V13, creadas por otro autor en su trabajo de fin de grado. Se pretende actualizar y explicar de forma general los programas y simulaciones, de tal forma que se encuentren preparados para su uso, modificando los posibles errores que estos pudieran tener



debido a las actualizaciones, pero no se incidirá la forma específica en la que el autor realiza la programación. Por otra parte, partiendo del carácter docente con el que se enfoca el proyecto, el proceso de actualización también es un objetivo de aprendizaje para el usuario. Por lo tanto, se indica todo el proceso y los campos específicos que el usuario debe revisar para asegurarse de que la actualización de cualquier programa se ha realizado de forma correcta.

La segunda parte del proyecto se centra en la creación de las herramientas necesarias para el aprendizaje de la programación y simulación de autómatas. En este apartado, siendo el más extenso y el objeto principal del proyecto, sí se explicará la programación y simulación de las plantas automatizadas, así como los conceptos que se pretenden transmitir al usuario. Además, se relacionarán los contenidos de aprendizaje con las competencias y objetivos docentes de varios grados y másteres de la Escuela de Ingenierías Industriales (EII) de la Universidad de Valladolid (UVA).

### 1.3. OBJETIVOS

Teniendo en cuenta lo antes mencionado en los apartados de justificación y alcance del proyecto, los objetivos planteados son los siguientes:

- **Adquirir y aplicar conocimientos** relacionados con la programación y simulación de la automatización de procesos industriales.
- **Comprender y programar** interfaces hombre-máquina.
- **Afianzar y emplear** competencias vinculadas con los controladores PID.
- **Comprender** las bases de los **protocolos de comunicación** entre los elementos de las estaciones industriales empleadas.
- **Actualizar cuatro estaciones de simulación**, indicando el proceso de actualización de los programas y dejándolas preparadas para su uso docente.
- **Programar** el proceso automático de varias estaciones industriales enfocadas al aprendizaje de la programación de la automatización de procesos.
- **Crear los entornos de simulación** de las estaciones industriales programadas.
- **Relacionar los conceptos teóricos y prácticos** empleados, con las **competencias y objetivos** de varias asignaturas de grado y másteres donde se podrían implementar.



## 1.4. ESTRUCTURA DEL PROYECTO

La memoria del proyecto comprende los fundamentos teóricos empleados en las estaciones automatizadas, así como la aplicación de estos conocimientos estructurados del siguiente modo:

- **Introducción y objetivos:** especifica la necesidad que cubre el proyecto, así como los objetivos que se pretenden lograr y la estructura de la memoria de este.
- **Fundamentos teóricos:** comprende la base teórica de los conocimientos necesarios para la realización del proyecto.
- **Softwares de programación y simulación:** contiene la información del uso y aplicación de los programas empleados en el trabajo de fin de grado.
- **Actualización de estaciones:** explica el funcionamiento de las estaciones que se recuperan de una versión antigua del programa, así como el propio proceso de actualización de las mismas.
- **Estaciones de simulación diseñadas:** explica el funcionamiento de las estaciones que se crean, junto con los conocimientos empleados en ellas.
- **Integración de las prácticas en la docencia:** relaciona las competencias y objetivos de las asignaturas de grado y másteres donde podrían emplearse las simulaciones.
- **Estudio económico:** analiza el coste que supone el aprendizaje de los softwares de programación y simulación de procesos automáticos.
- **Conclusiones y líneas futuras:** presenta las consideraciones alcanzadas en el proyecto y las posibles continuaciones de este.
- **Bibliografía:** detalla el material empleado como referencia para la elaboración del trabajo.





## 2. FUNDAMENTOS TEÓRICOS

Con el objetivo de clarificar la comprensión del trabajo, se analizan los principales conceptos teóricos empleados en el desarrollo del proyecto. El marco teórico del trabajo es demasiado amplio como para analizar de forma detallada cada elemento, por lo tanto, se definen de forma general los conceptos utilizados y la función que desempeñan.

### 2.1. GRAFCET

Para la programación de la mayoría de las estaciones industriales se toman los diagramas GRAFCET como la estructura que debe seguir el programa para que el proceso industrial se ejecute de forma secuencial. [\[1\]](#)

#### 2.1.1 Definición de GRAFCET

El GRAFCET (o gráfico funcional de control de etapas y transiciones) es un método gráfico de representación de los comportamientos de sistemas secuenciales.

Se encarga de describir de manera gráfica la secuencia de operaciones de un sistema lógico en función de sus entradas y salidas, facilitando la estructuración y descomposición de un problema de control en partes de menor complejidad.

Los GRAFCETS, por norma general, representan la evolución del sistema en sentido descendente y hacia la derecha siempre que se pueda.

#### 2.1.2 Elementos del GRAFCET

Los diagramas GRAFCET están compuestos por tres elementos que representan las funciones de control de un sistema secuencial, siendo estos tres elementos los siguientes:

- **Etapas:** Describen el comportamiento de una parte, o de la totalidad del sistema, en el que se llevan a cabo operaciones concretas. En un instante concreto, una etapa solo puede estar activa o inactiva.
- **Acciones:** Detallan el conjunto de operaciones que se realizan cuando la etapa está activa. Las acciones están asociadas a una o varias etapas concretas.
- **Transiciones:** Conjunto de condiciones que deben cumplirse para permitir la evolución de una etapa a la siguiente. Si las condiciones asociadas a una transición se cumplen, la etapa anterior de la transición pasa de estado activo a inactivo y la etapa posterior de inactiva a activa.

### 2.1.3 Reglas del GRAFCET

Los GRAFCET siguen unas normas sintácticas que permiten describir de forma clara e inequívoca el proceso evolutivo de un sistema secuencial. Estas reglas se pueden resumir de la siguiente forma:

- Una etapa se activa cuando la condición lógica asociada a la transición que la precede es verdadera y la etapa previa está activa.
- Cuando una etapa pasa a estado activo, la primera acción que realiza es la desactivación de la etapa anterior y, por lo tanto, de las acciones que tuviera asociadas.
- Entre dos etapas consecutivas debe haber únicamente una transición.
- El GRAFCET siempre debe estar cerrado. Es decir, sin importar el punto del diagrama en el que se encuentre la secuencia, siempre debe de poder continuar por alguna rama a otra etapa.

### 2.1.4 Estructuras de secuencias del GRAFCET

El modo de distribuir las etapas y las condiciones de transición que se establecen entre ellas permiten definir la secuencia de ejecución de las acciones que determinan la evolución del automatismo. En este sentido, las estructuras por las que puede estar compuesto un GRAFCET son muy variadas, dando lugar a secuencias de ejecución lineal, alternativa, interactiva, concurrente y de saltos. A continuación, se describen las estructuras empleadas en los diagramas GRAFCET de este trabajo.

**Estructura lineal:** La estructura de secuencia lineal es una configuración en la cual una etapa se activa después de la anterior en el instante en el que la condición que la precede es verdadera.

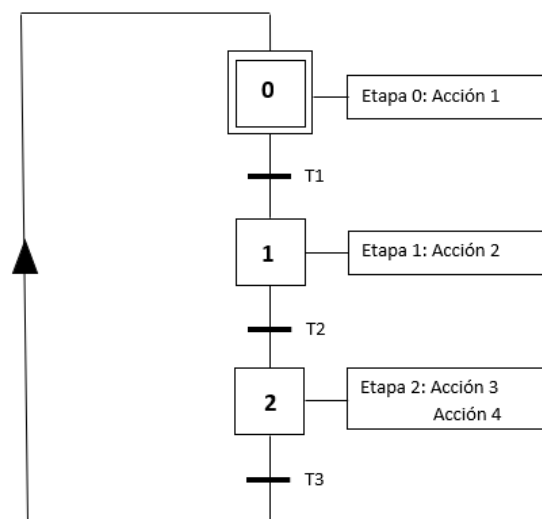


Figura 1. Estructura lineal del GRAFCET

**Estructura de salto hacia atrás:** La estructura de salto hacia atrás permite saltar una o varias etapas siempre que se cumplan las condiciones de la transición asociada a este salto.

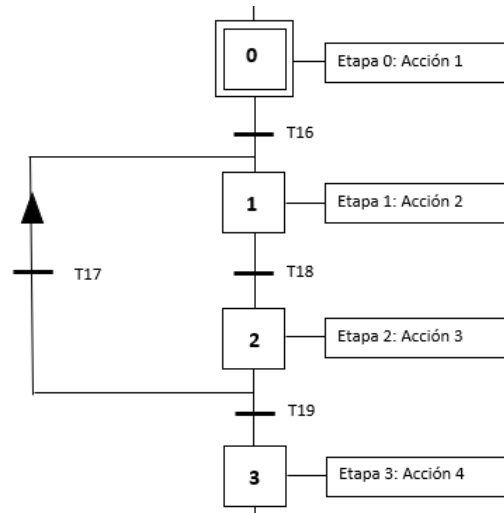


Figura 2. Estructura de salto hacia atrás del GRAFCET

En este caso las transiciones 17 y 19 deben ser mutuamente excluyentes.

### 2.1.5 Niveles del GRAFCET

Según la complejidad y detalle del diagrama se puede clasificar al GRAFCET en tres niveles:

- **Nivel 1 o descripción funcional;** Este tipo de GRAFCET describe de manera genérica el funcionamiento del sistema secuencial. El diagrama no especifica los sensores o actuadores del sistema, ni tampoco las variables que intervienen en el proceso. Su función es permitir al usuario una rápida comprensión del funcionamiento y estructura del mismo.
- **Nivel 2 o descripción tecnológica:** Es un diagrama que contiene toda la información que se podría dar en un GRAFCET de nivel 1 pero describiendo de forma técnica los procesos del sistema. En este tipo de diagramas se especifican las acciones realizadas por sensores y actuadores, así como el uso detallado de cada uno de ellos, pero no se emplean las variables asociadas a ellos.
- **Nivel 3 o descripción operativa:** Es el tipo de GRAFCET más completo, pero también el más complejo de comprender debido a la necesidad de conocer las entradas y salidas asociadas al proceso secuencial. En este diagrama se reflejan los contenidos de un diagrama de nivel 2, con la diferencia de que se emplean las variables de entrada y/o salidas asociadas. De esta forma quedan descritas todas las acciones de las etapas y las condiciones de las transiciones, en función de las variables empleadas en la programación.





- **Clase C:** 3 bits para indicar la clase, 21 bits para la red y 8 bits para el host.

110	Dirección de red (21 bits)	Dirección local (8 bits)
-----	----------------------------	--------------------------

IP's desde 192.X.X.X hasta 223.X.X.X

Máscara: 255.255.255.0

Empleada para redes pequeñas, pudiendo identificar menos de 258 (2<sup>8</sup>) ordenadores en cada red.

- **Clase D:** 4 bits para indicar la clase y 28 bits para la dirección.

1110	Dirección de difusión múltiple (28 bits)
------	--

Las direcciones de clase D se reservan para difusión en un área limitada.

Mediante el uso del protocolo IP, se consigue un modo seguro y fiable de comunicación entre los diferentes elementos de simulación de las estaciones industriales creadas.

A lo largo del proyecto, la dirección IP empleada en los PLC's es 192.168.0.1. En caso de ser necesario un HMI se le asocia la dirección inmediatamente superior.

## 2.3. PLC

Para la programación y control de las estaciones industriales se emplean autómatas o controladores lógicos programables (más conocido por sus siglas en inglés PLC, que provienen de Programmable Logic Controller) de la marca SIEMENS. [\[1\]](#)

### 2.3.1 Descripción del PLC

Un PLC es un dispositivo electrónico con memoria programable para el almacenamiento de instrucciones, permitiendo la implementación de funciones específicas.

El autómata realiza la ejecución cíclica de un programa, permitiendo su interrupción momentánea en función de la prioridad de las tareas. La característica más destacable de esta ejecución de programa es la garantía de ejecución completa del programa principal o main.

Los autómatas son elementos que ofrecen un control en tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada en un tiempo limitado. La capacidad de controlar y variar rápidamente el estado de un gran número de señales de entrada y salida dota al dispositivo de una alta versatilidad en el control de procesos y máquinas.

Además, son elementos que permiten la interconectividad entre ellos y con ordenadores. Esto posibilita la confección de una red de autómatas controladas por uno o varios PC's que permiten acceder al autómata y disponer, e incluso modificar, toda la información que contenga.



Desde su creación en la década de los 70 hasta la actualidad, los autómatas programables desempeñan un papel fundamental en la industria como elementos de control. Hasta el momento, no ha habido ninguna tecnología capaz de competir en este ámbito con los PLC's.

Es por estas razones, por las cuales su uso en la industria, desde su creación hasta la actualidad, está muy extendido, empleándose en procesos y máquinas como elemento de control lógico, secuencial o una combinación de ambos.

De acuerdo con un estudio de Interact Analysis sobre las cuotas de mercado de los fabricantes de PLC en 2017, SIEMENS domina el mercado, seguida por Rockwell Automation, Mitsubishi Electric y Schneider Electric como empresas principales.

En este proyecto, se ha decidido emplear instrumentos de programación del fabricante SIEMENS fundamentalmente por dos motivos. El primero es la importancia y extensión de SIEMENS en el mundo industrial, siendo uno de los principales fabricantes en el mercado. El segundo, está relacionado con el ámbito económico, puesto que la UVA (universidad que contiene a los grados y másteres a los que están enfocadas las prácticas desarrolladas en este trabajo) ya dispone de licencias y componentes de esta marca, reduciendo el coste y proporcionando una mayor utilidad y aplicación a este trabajo.

### 2.3.2 Estructura del PLC

La estructura interna de los autómatas es muy similar, sin embargo, su estructura externa permite clasificarlos en dos grupos, autómatas compactos o modulares.

#### La estructura interna

A nivel de programación se suele ver al autómata como una caja negra con una memoria que almacena el programa, con entradas y salidas que permiten interactuar con el entorno. Sin embargo, todos los PLC's llevan en su interior cinco elementos:

- **Microprocesador:** Puede haber uno o varios, siendo el elemento encargado de leer y ejecutar las secuencias de programa almacenadas en la memoria, calculando los valores de las salidas en función de los valores de las entradas.
- **Memoria:** Es el bloque constituido por un conjunto de elementos que se encargan de almacenar el sistema operativo, el programa, los datos de control y los datos de proceso.
- **Bloque de comunicaciones:** Permite la comunicación del PLC con el entorno.
- **Entradas/salidas:** Es el conjunto de elementos que permiten conectar al autómata con el proceso a controlar.
- **Control de expansión:** Es el bloque encargado de permitir la conexión y comunicación del autómata con otros módulos de ampliación de prestaciones. Algunos autómatas compactos no disponen de este bloque.

Para que todos estos elementos funcionen, es necesario que el controlador esté alimentado con una **fuentes de tensión** (comúnmente de 24 voltios de CC).

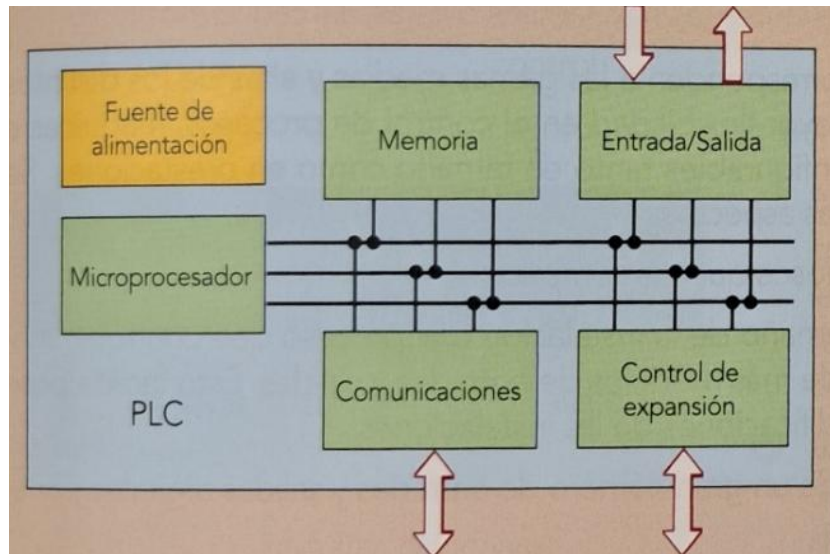


Figura 3. Estructura interna de un autómata

### Estructura externa

Dependiendo del número de módulos empleados para contener a los elementos que conforman la estructura interna del PLC, se puede diferenciar los siguientes tipos de autómatas:

- **Autómatas compactos:** son aquellos autómatas cuya estructura interna está contenida en un solo módulo o caja.

Se caracterizan por tener un coste menor, la fuente de alimentación está integrada y dispone de menos entradas y salidas que los modulares. En ocasiones es posible expandirlos para aumentar sus prestaciones.

- **Autómatas modulares:** son aquellos conformados por varios módulos o cajas en los que se han distribuido los diferentes componentes de la estructura interna del autómata.

Son controladores más versátiles, permitiendo ajustarse a las necesidades del proceso o instalación mediante la anexión y retirada de ciertos módulos. Se caracterizan por ser más caros y voluminosos y al tener una configuración modificable, tienen un número de señales de entrada y salida mucho mayor.



Figura 4. PLC modular SIEMENS S7-1200 [3]



Las estructuras modulares más comunes son la europea y la americana.

La **estructura americana** agrupa procesador, memoria, bloque de expansión y de comunicaciones (conjunto denominado CPU) y la fuente de alimentación en un módulo principal y las entradas y salidas en otros módulos.

En la **estructura europea** se pueden distinguir tres tipos de módulos. El primero alberga la fuente de alimentación, el segundo la CPU y el último tipo, está constituido por las entradas y salidas digitales y analógica.

### 2.3.3 Entradas y salidas

La manera de comunicación del autómatas con sensores y actuadores está limitada al conjunto de entradas y salidas que lo componen.

Las señales de entradas y salidas pueden ser de tres tipos: digitales, analógicas o especiales (señales de alta frecuencia o procedentes de codificadores, entregadas o generadas por el PLC especiales en formato, velocidad o codificación).

En lo que respecta al proyecto, se emplearán únicamente señales digitales y analógicas, siendo las más comunes y permitiendo el control de casi todo el espectro de procesos industriales de complejidad media.

### 2.3.4 Programación de los autómatas

Los instrumentos básicos de programación de un PLC se pueden reducir a dos, el lenguaje de programación o software y un equipo para la programación o hardware.

El hardware esencial para la programación es el propio PLC junto con un ordenador o PC empleado por el programador para la creación, modificación o revisión del programa. En el pasado, no era común el uso de PC's para esta función, se usaban en cambio, terminales o consolas de programación complejas de operar, muy caras y específicas para cada fabricante y/o modelo de PLC.

El software comprende el programa creado y cargado en el autómatas junto con la herramienta para programarlo. Es común que cada fabricante ofrezca un software propio de programación para sus productos.

Dentro de los procesos automatizados los elementos más comunes de la instalación y que intervienen en la programación son los cuatro que se muestran a continuación:

- **PC y software de programación:** En el trabajo, para la realización del programa de control de la estación se emplea un PC como elemento de comunicación con el software, y TIA Portal V17 para la creación, modificación y revisión del programa.
- **PLC:** Como elemento de simulación del autómatas físico, se emplean dos simuladores de SIEMENS en este trabajo, PLCSIM y PLCSIM Advanced.

- **Instalación:** La instalación física compuesta por sensores, actuadores y elementos mecánicos se simula con Factory IO y SIMIT, este último es un programa desarrollado por SIEMENS.
- **HMI:** En el trabajo se emplean dos herramientas de SIEMENS. TIA Portal V17 para la programación y WinCC para la simulación del elemento físico.

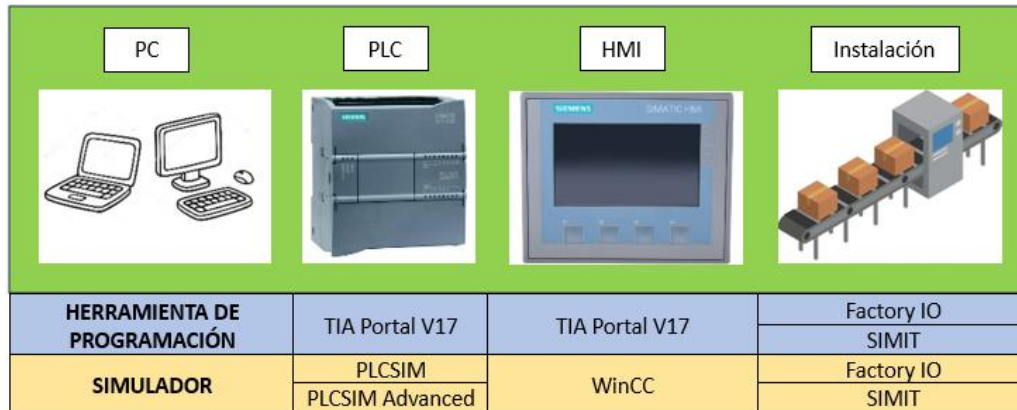


Figura 5. Hardware y software de programación de las estaciones

Como ya se ha mencionado con anterioridad, es muy común que los fabricantes de PLC's desarrollen softwares específicos para su programación, pero también lo hacen para su simulación. De tal forma que, al seleccionar un autómatas para el control de una máquina o instalación, se está eligiendo también los softwares del fabricante de ese controlador para su programación y simulación.

Se puede comprobar que la gran mayoría de softwares de programación y simulación utilizados en el proyecto son propios del mismo fabricante que los PLC's empleados, SIEMENS. Tan solo Factory IO es un programa desarrollado por otra entidad y no tiene la restricción de utilizar softwares específicos del mismo fabricante que el autómatas.

### Modos de funcionamiento

Los autómatas disponen de dos o tres formas diferentes de trabajo dependiendo del fabricante y modelo:

- **Modo RUN:** En este modo el programa se está ejecutando, siendo el modo normal de trabajo.
- **Modo STOP:** La ejecución de programa se detiene, desconectando las salidas y almacenando los valores de las variables internas en memoria interna. De esta forma, se paraliza el autómatas, pero no se pierde la información.
- **Modo ERROR:** El autómatas detiene la ejecución de programa y desconecta las salidas debido a la detección de un fallo de funcionamiento. El PLC se mantendrá en este estado hasta que el fallo se corrija y deje de detectarse.

Dependiendo del fabricante, al modo STOP se le puede llamar PROGRAM o pueden encontrarse con modos de RESET o REMOTE RUN.

## Ciclo de scan

El programa se ejecuta en un tiempo inferior al denominado ciclo de scan. Un ciclo de scan es la secuencia de operaciones que realiza el autómata de manera repetitiva una vez que el PLC está activo. Siendo estas operaciones la ejecución de procesos comunes (comprobaciones y procesos de autodiagnóstico), la lectura de las variables físicas de entrada, la ejecución del programa y la escritura de las variables del programa en las salidas.

El tiempo de ciclo de scan es variable en función de la extensión del programa y de la cantidad y tipo de entradas que se empleen en él. Existen autómatas con ciclos de scan muy variados, pero se podría determinar que la gran mayoría están comprendidos entre cien milisegundos y del orden de microsegundos.

## Norma IEC 61131-3

Actualmente, se ha establecido una estandarización de los lenguajes de programación de autómatas puesto que, en su inicio, cada fabricante establecía los suyos propios. Esta homogenización se recoge en la norma IEC 61131 (que se corresponde con la norma española UNE-EN 61131) definiendo en su apartado 3 los cuatro lenguajes de programación permitidos:

- **Lenguajes textuales:**
  - Lista de instrucciones o AWL.
  - Texto o programación estructurada.
- **Lenguajes gráficos:**
  - Diagramas de contactos o ladder.
  - Diagramas de Bloques Funcionales (FBD).

Para la representación de las entradas y salidas la norma IEC 61131 también especifica un estándar que sigue el siguiente formato:

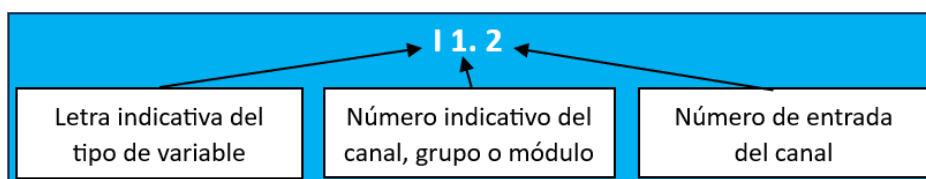


Figura 6. Formato estandarizado de variables

Las entradas se representan con la letra I y las salidas con la letra Q. Es común encontrarse a las entradas caracterizadas mediante la letra E y a las salidas con la A. Para diferenciar un formato de otro, se conoce como nomenclatura inglesa si se emplean I y Q y nomenclatura alemana si se usan E y A como caracteres de variables de entrada y salida respectivamente. Las variables internas del programa, denominadas marcas, se caracterizan con la letra M. A lo largo del trabajo y en el desarrollo de los programas, se ha empleado la nomenclatura inglesa.



## Lenguajes de programación empleados

Como se ha mencionado anteriormente, hay cuatro lenguajes contemplados en la norma IEC 61131, sin embargo, este trabajo se centrará en el aprendizaje y uso del lenguaje ladder. Esto se debe a que es el más empleado en la programación de autómatas, muy polivalente y toda la programación de dificultad baja, media y alta se puede programar en este lenguaje.

Además, el texto estructurado y el AWL son lenguajes complejos y poco visuales, no aportando ninguna funcionalidad añadida, ya que se puede programar las mismas instrucciones y operaciones en un lenguaje más sencillo, como es el lenguaje ladder. Por otra parte, el lenguaje FBD, aunque es más simple, sigue siendo un lenguaje poco visual, con un coste computacional mayor que el ladder y con funcionalidades limitadas. Además, la posibilidad de programar las mismas instrucciones y operaciones en un lenguaje más sencillo inclina a decantarse por una programación más simple como es la de escalera.

Aunque anteriormente no se ha mencionado, existe un lenguaje que no está contemplado en la norma IEC 61131. Se trata de un lenguaje gráfico denominado GRAFCET. Al no estar normalizado, solo algunos autómatas tienen incluida la posibilidad de programar en él, siendo este el caso de la gama S7-1500 a diferencia de los S7-1200.

En el proyecto se emplean diagramas GRAFCET para reflejar el comportamiento del sistema controlado y estructurar la programación del PLC que lo controla, pero no se emplea el lenguaje GRAFCET para programarlo.

- **Diagramas de contactos o ladder**

Denominado también como lenguaje en escalera, representa gráficamente la lógica y diseño de automatismos por relevo de relés. Es el lenguaje más empleado en la programación de PLC's debido a su fácil comprensión y visualización.

Todos los programas de autómatas en este trabajo se han realizado en diagrama de contactos.

La simbología está estandarizada, mostrándose a continuación los símbolos básicos del conjunto de instrucciones y acciones de este lenguaje:

SIMBOLO	DESCRIPCIÓN	SIMBOLO	DESCRIPCIÓN
	Contacto normalmente abierto (NA)		Negación
	Contacto normalmente cerrado (NC)		Bobina directa
	Flanco positivo		Bobina de activación
	Flanco negativo		Bobina de desactivación

Tabla 1. Instrucciones básicas en lenguaje ladder

Los diagramas de contactos se ejecutan de izquierda a derecha y de abajo a arriba. Los símbolos o instrucciones que representan las condiciones de activación se sitúan a la izquierda de las líneas horizontales y las salidas, activaciones de variables o bloques funcionales a la derecha.

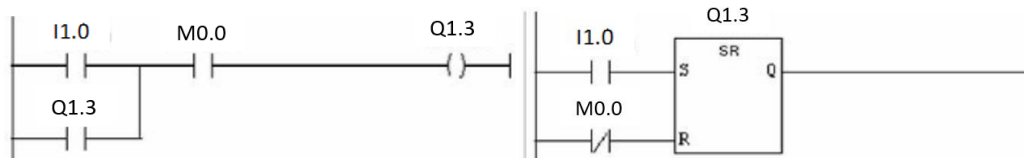


Figura 7. Ejemplos de diagramas ladder

Los bloques funcionales, temporizadores, contadores, etc. se representan con un rectángulo y el acrónimo o nombre correspondiente. Un ejemplo de estas estructuras más complejas es el bloque funcional SR mostrado en la figura anterior.

#### Características:

- Está basado en los esquemas eléctricos.
- Es muy visual y fácil de entender.
- Presenta dificultades al trabajar con bucles y saltos.
- Coste computacional medio.
- Limitado a la hora de mover datos.
- Es el más empleado.

Al ser el lenguaje más empleado en la programación de autómatas, muy polivalente y teniendo en cuenta que toda la programación que se realiza en este trabajo no requiere ni bucles, ni saltos ni mover datos, se ha decidido centrar el aprendizaje de este lenguaje sobre el resto.

#### Interfaces hombre máquina (HMI)

Las interfaces hombre máquina, en inglés Human Machine Interfaces (HMI), son dispositivos electrónicos cuya función es facilitar la interacción del operario con máquinas o procesos, introduciendo parámetros, instrucciones o facilitando la supervisión del sistema.

Los HMI o paneles de operador son pantallas, normalmente táctiles dotadas también de botones físicos, que muestran el estado del proceso, mensajes de alarmas o avisos y permite interactuar con el proceso según su programación.



Figura 8. HMI TP700 Comfort de SIEMENS [3]



Estos dispositivos, al igual que el PLC, deben ser programados con antelación y requieren de un software específico para ello, normalmente facilitado por el fabricante del componente.

La programación de HMI's está enfocada de forma gráfica, permitiendo la creación de pantallas, conocidas también como imágenes, con diferentes elementos visuales e interactivos denominados objetos.

El HMI se conecta con el autómatas, recibiendo y asociando variables entre sí. Estas variables no son únicas para los dos elementos, sino que una variable en uno de los elementos de control tiene asociada una variable análoga en el otro componente. De esta forma, mediante los objetos de cada pantalla se pueden modificar estados o valores de variables empleados en el PLC y viceversa. Sin embargo, la programación de los HMI's es independiente de la de los autómatas. Es decir, las pantallas y los objetos contenidos en ellas se programan y tienen funciones independientes a las del PLC, pero se pueden modificar variables comunes en uno de los componentes lo que condiciona el estado del programa del otro.

Los objetos (botones, pulsadores, displays, gráficas, etc.) suelen llevar asociados una variable. Mediante la modificación del valor de la variable asociada, se puede variar las características de animación, apariencia o visualización de estos objetos.

La combinación del PLC y el HMI proporcionan un control muy completo sobre la instalación. Actualmente muchas de las instalaciones controladas por controladores programables suelen estar dotadas de uno o varios HMI's con el objetivo de favorecer la interacción y supervisión del proceso por el operario o personal de mantenimiento.

### 2.3.5 Aplicaciones, ventajas e inconvenientes de los autómatas

El campo más inmediato de aplicación de los PLC's son los procesos industriales. El desarrollo, evolución y mejora tanto del software como del hardware de programación de autómatas desde su creación en la década de los 70 hasta la actualidad, proporciona a estos elementos de control una variedad de modelos muy extensa con alta consistencia y fiabilidad.

Actualmente todos o casi todos los equipos industriales con cierta complejidad están controlados con un autómatas.

#### **Campos de aplicación principales:**

- Control de máquinas.
- Procesos de fabricación en líneas continuas.
- Procesos en los que se requiera control lógico.
- Control de cultivos, destilerías o refinerías.
- Industrias de automóviles, maderas, muebles, alimentación, etc.
- Domótica y control de sistemas domésticos.
- Instalaciones de procesos complejos.



### **Ventajas de los PLC's:**

- Usan lógica programada en vez de cableada.
- Es un sistema muy fiable.
- Menor tiempo de elaboración de proyectos.
- Menor tiempo de puesta en funcionamiento.
- Mínima ocupación de espacio, reducido peso y tamaño.
- Menor coste de instalación, actualización y mantenimiento.
- Posibilidad de hacer cambios fácilmente con un mínimo coste.
- Posibilidad de gobernar varias máquinas con el mismo autómatas.
- Autómatas reutilizables para distintas máquinas o procesos.
- Es posible su instalación en entornos con condiciones ambientales o climatológicas severas.
- Facilita la implementación de sistemas de control distribuido y el control jerarquizado.
- Permite la simulación de procesos, alarmas y fallos sin influir directamente en la máquina o proceso.

### **Desventajas de los PLC's:**

- Requieren de personal formado específicamente, tanto en su programación como para las tareas de mantenimiento.
- Coste inicial de los componentes de desarrollo elevado.
- Cada fabricante propone unos softwares específicos para sus dispositivos.

## **2.3.6 Hardware de SIEMENS**

SIEMENS AG es uno de los fabricantes principales de controladores programables y dispositivos eléctricos asociados, como pueden ser los HMI. No solo es un fabricante líder en el mercado actual, sino que es una entidad con gran recorrido en el mercado de la automatización. Ofrece una gran variedad productos, ajustándose a la dimensión y complejidad de los procesos o máquinas a controlar y automatizar, asegurando siempre una alta fiabilidad y robustez. [\[4\]](#)

Además de todas estas razones, el hecho de que la UVa posea dispositivos y licencias de software de esta marca hace que la elección de los instrumentos de programación SIEMENS para el desarrollo de este proyecto sea la más acertada.

Como ya se ha comentado anteriormente, en este trabajo se emplean los dispositivos de control de esta entidad, más concretamente los controladores del grupo SIMATIC S7. El grupo SIMATIC lleva desarrollándose y optimizándose desde 1958, con la aparición de la familia SIMATIC G, hasta la actualidad con SIMATIC S7.

La familia SIMATIC S7 ofrece varias gamas de controladores que se adaptan a las necesidades de los diferentes procesos, siendo estas: S7-200, S7-300, S7-400 y las más actuales S7-1200 y S7-1500. Cabe resaltar que la familia S7 no solo está compuesta por controladores, sino que también ofrece diversos modelos de HMI's.

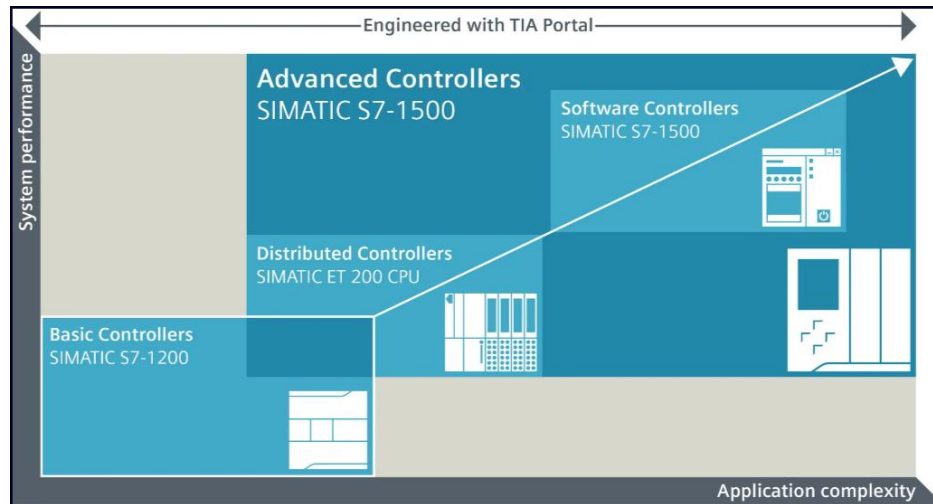


Figura 9. Gamas de controladores SIMATIC [5]

En el gráfico anterior se muestran las gamas más actuales de controladores SIMATIC S7 en función de la complejidad de la aplicación y del rendimiento o actuación del sistema.

## 2.4. PID

El regulador PID (o controlador Proporcional, Integral y Derivativo) se emplea en una de las estaciones industriales de este trabajo como elemento de control del nivel de líquido en un tanque. Con el objetivo de entender el funcionamiento de la estación y el programa que la controla, se explican los conocimientos básicos relacionados con el regulador PID. [6]

### 2.4.1 Control en lazo cerrado

El PID se trata de un controlador de lazo cerrado. Para su mejor comprensión, se define este concepto y se analizan sus diferentes señales.

Un sistema en lazo cerrado es un sistema con realimentación y, por lo tanto, la salida sí que interviene en la acción de control. El valor de salida se compara con el valor deseado y la diferencia entre ambos (error) se emplea sobre la planta para reducir la diferencia de valores y conseguir que la señal de salida alcance el valor deseado. [7]

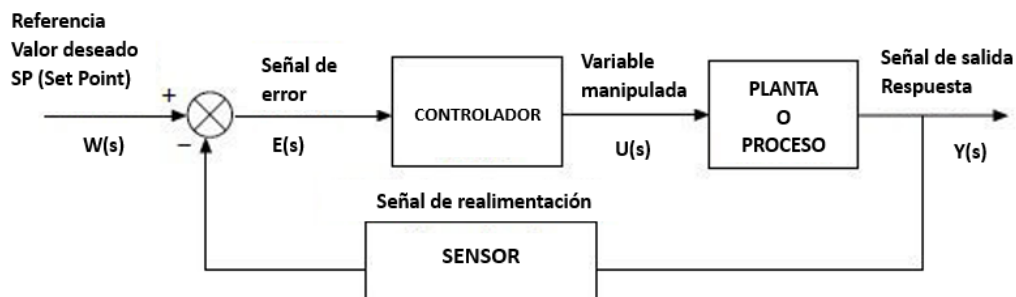


Figura 10. Control en lazo cerrado

Los controles en lazo cerrado se consideran sistemas automáticos, y son el tipo de control que se emplea en un regulador PID.

### 2.4.1 Objetivos del controlador

Con el control en lazo cerrado se pretende:

- Asegurar la estabilidad absoluta de la respuesta y mejorar la relativa.
- Mejorar la respuesta transitoria, consiguiendo un buen seguimiento del valor de referencia.
- Rechazar las perturbaciones.
- Eliminar las oscilaciones de la respuesta.
- Eliminar o reducir el error estacionario.

### 2.4.2 Acciones del controlador PID

El controlador PID es un regulador basado en señal, por lo que no implica un conocimiento explícito del proceso, sino que ofrece una regulación de la señal de salida en función de la señal de entrada.

El controlador consta de 3 parámetros de ajuste:

- $K_p$ : Término proporcional o ganancia.
- $T_i$ : Término integral o tiempo integral.
- $T_d$ : Término derivativo o tiempo derivativo.

Cada uno de los parámetros aporta una acción sobre la señal de salida, que repercutirá sobre la señal del error.

- **Acción proporcional:** La variable manipulada, o acción de control, es proporcional a la señal de error, reduciendo el error estacionario. La representación matemática correspondiente a esta acción, en el tiempo, es la siguiente:

$$u(t) = K_p * e(t)$$

La ecuación indica que el error entre la variable de salida y el valor de referencia ( $e(t)$ ) se multiplica por una constante ( $K_p$ ) para reducir la diferencia entre esos dos valores. Por lo tanto, a mayor  $K_p$ , menor error estacionario. Sin embargo, la acción proporcional no elimina el error estacionario por si sola.

La acción proporcional puede ser directa o inversa. Si la ganancia del controlador y del sistema son de igual signo, la acción es directa. Es decir, si al aumentar el valor de  $K_p$  disminuye el error la acción es directa. En caso contrario, la acción es inversa.

- **Acción integral:** La acción de control depende de la integral del error, afectando a la precisión y a la estabilidad del sistema. Su representación matemática en el tiempo es la siguiente:

$$u(t) = \frac{1}{T_i} * \int_0^t e(\tau) d\tau$$

De la ecuación se deduce que la integral del error crecerá, y con ella la acción de control, hasta que no se alcance el setpoint. Este fenómeno puede eliminar el error estacionario o llevar a la inestabilidad del sistema.

- **Acción derivativa:** La acción derivativa suaviza los cambios de la variable manipulada provocados por cambios bruscos en la señal de error, evitando así los sobrepicos. Su representación matemática en el tiempo es la siguiente:

$$u(t) = T_d * \frac{d}{dt} e(t)$$

La variable manipulada depende de la derivada del error. Si esta es de valor negativo significa que la variable manipulada se acerca al valor de referencia, y la acción de control se modera. En caso contrario, si la derivada del error es positiva, significa que la diferencia entre la variable manipulada y la referencia se hace cada vez mayor, por lo que se incrementa la acción de control. De esta forma se consigue una acción de control de carácter anticipativo, puesto que la variable manipulada crece o decrece en función de los valores inmediatamente anteriores.

Este tipo de acción amplifica el ruido, puede saturar el actuador y no afecta al valor estacionario, solo actúa en el transitorio, puesto que la derivada de un valor constante es cero.

Es habitual la combinación de las acciones de control antes mencionadas para conseguir los objetivos deseados en el control de un sistema según sus requerimientos. La acción proporcional se emplea en la mayoría de los controladores, dando lugar a controladores de tipo P, PI, PD y PID.

Según las necesidades y características del proceso a controlar, se puede recomendar de forma genérica los reguladores más eficaces para su control:

- **P:** Procesos con un integrador integrado o donde no sea importante un error estacionario nulo.
- **PI:** Indicado para la mayor parte de los casos.
- **PID:** Indicado en procesos lentos y sin ruido.

### 2.4.3 Funcionamiento del controlador PID

El Control Proporcional Integral y Derivativo reúne las ventajas de las tres acciones por separado, pero también alguno de sus inconvenientes.

$$u(t) = K_p * [e(t) + \frac{1}{T_i} * \int_0^t e(\tau) d\tau + T_d * \frac{d}{dt} e(t)]$$

Gracias a la acción proporcional, se reduce el error estacionario y el tiempo de respuesta. Debido a la acción integral, se elimina el error estacionario, pero se puede generar inestabilidad en la variable señal de salida. Finalmente, la acción derivativa se anticipa a la tendencia del error, mejora la ganancia y añade amortiguamiento, mejorando así la señal. Por contraposición, amplifica el ruido y puede saturar al actuador.

La sintonía o ajuste del regulador, es el proceso seguido para determinar el valor de los parámetros del PID ( $K_p$ ,  $T_i$  y  $T_d$ ). En el caso de la estación industrial de este trabajo

donde se emplea un PID, se hace uso de la propia herramienta ofrecida por el programa para realizar el ajuste o se emplea un método de prueba y error.

El método de prueba y error consiste en ajustar el valor de los parámetros del PID de forma experimental y centrándose únicamente en la gráfica de la respuesta del sistema. El proceso se podría dividir en cuatro pasos:

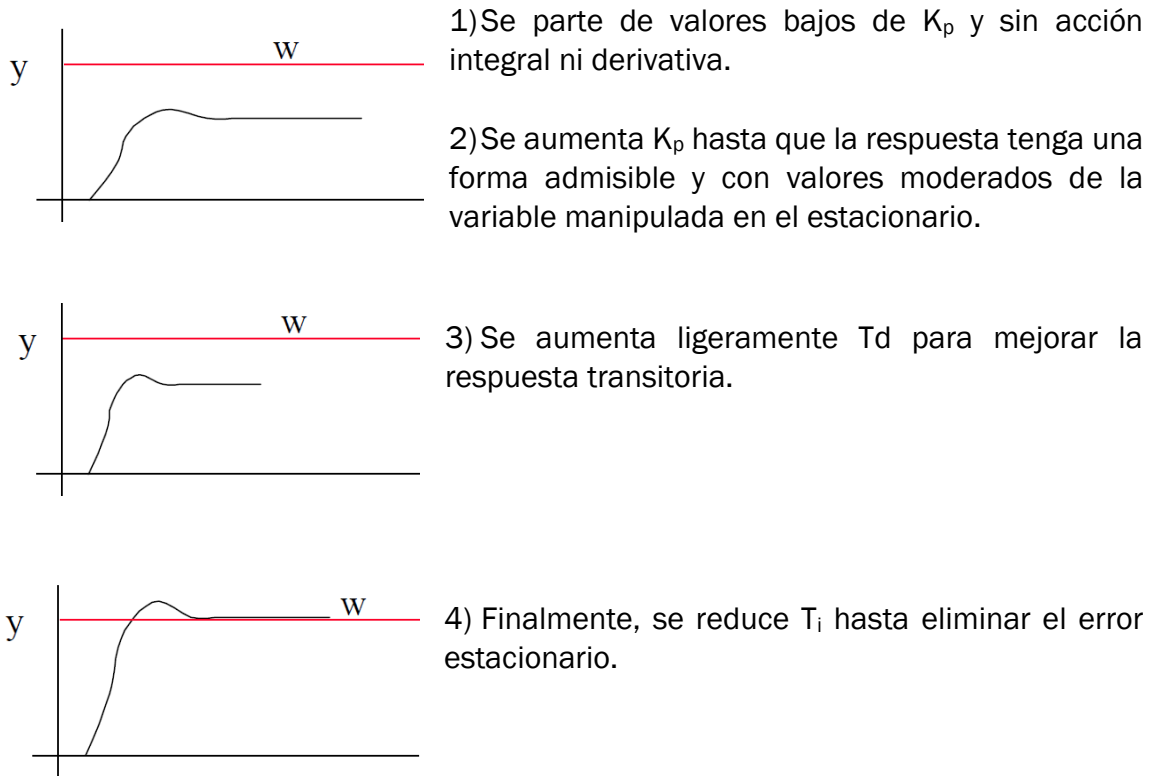


Figura 11. Sintonización de un PID con el método de prueba y error

## 3. SOFTWARES DE PROGRAMACIÓN Y SIMULACIÓN

### 3.1. SELECCIÓN DEL SOFTWARE

La selección del software a emplear a lo largo del proyecto, tanto en el apartado de programación como en la simulación, está fuertemente condicionada al fabricante de los dispositivos de control. Esto se debe a que los fabricantes de controladores programables suelen limitar la programación y simulación de sus PLC's y dispositivos de control, como los HMI's, a softwares propios.

La decisión, ya justificada en el [apartado 2.3](#) del trabajo, de emplear autómatas y HMI's de SIEMENS condiciona el uso de softwares de programación y simulación mayoritariamente del mismo fabricante.

Uno de los principales condicionantes de todo proyecto es el ámbito económico. Como el trabajo tiene como objetivo la creación de prácticas para grados y másteres de la Universidad de Valladolid, se ha considerado las licencias y materiales de los que dispone actualmente la UVA para reducir costes de adquisición de software y material. En lo referente a licencias de software, la UVA cuenta con un paquete de SIEMENS que incluye TIA Portal V17, PLCSIM, PLCSIM Advanced y WinCC. Con estos programas se posibilita la programación y simulación de PLC's y de HMI's.

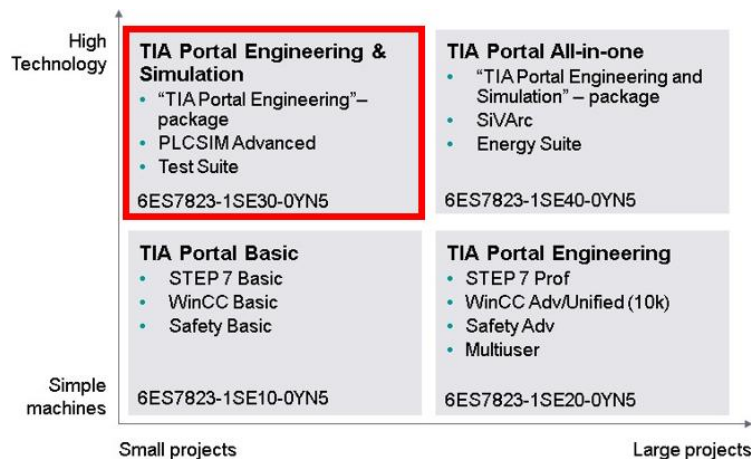


Figura 12. Paquetes con TIA Portal V17 [\[8\]](#)

#### Software de programación de elementos de control

Para la programación de autómatas, y en caso de ser necesario los HMI's, se emplea el software TIA Portal versión 17 (V17) del fabricante SIEMENS. [\[4\]](#)

El software de programación de los dispositivos de control está restringido a los desarrollados por SIEMENS. De todas formas, TIA Portal es una potente herramienta que satisface todas las necesidades de las estaciones programadas y los objetivos docentes de este trabajo.

Además, este software es muy empleado en la industria en la actualidad, por lo tanto, los conocimientos adquiridos con este trabajo tienen una gran aplicación práctica en el mundo laboral relacionado con la industria, domótica o instalaciones con procesos complejos.



### **Software de simulación de elementos de control**

Como simuladores de los autómatas se emplean PLCSIM y PLCSIM Advanced, mientras que como simulador de paneles de operador se emplea WinCC. [4]

PLCSIM está integrado en el software básico de TIA Portal y permite simular autómatas de las gamas S7-1200 y S7-1500 entre otros. Sin embargo, tiene el inconveniente de no poder simular simultáneamente más de un controlador. La elección de este simulador cumple con las necesidades del trabajo y no aumenta el coste económico del mismo.

Al contrario que PLCSIM, PLCSIM Advanced no está integrado en el software básico de TIA Portal, sino que se requiere de una licencia específica para su uso. No obstante, la UVa ya dispone de ella por lo que no supone un coste económico adicional. El uso de otro simulador aumenta las competencias del alumno en el campo de la automatización. En adición, suple la limitación de no poder simular más de un PLC de forma simultánea. Por contraposición no puede simular PLC's de la gama S7-1200.

Para la simulación de HMI's se emplea otro software básico integrado en TIA Portal V17 denominado WinCC con el objetivo de no aumentar costes de adquisición de programas. El uso de paneles de operador complementa y facilita el control de instalaciones automatizadas, por lo que es un dispositivo que vale la pena aprender a programar y simular. Los simuladores de HMI son muy similares, es por esto, que la elección de WinCC frente a otros softwares similares supone una ventaja debido a su coste adicional nulo.

### **Software de simulación de procesos industriales**

La simulación del proceso industrial compuesto por sensores, actuadores y elementos mecánicos se realiza con entornos virtuales creados con Factory IO y SIMIT, lo que permitirá simular el proceso industrial y comprobar el correcto funcionamiento del sistema de automatización y control programado. [4]. [9]

En lo referente a coste de adquisición de los programas, se pretende hacer uso de versiones gratuitas de los mismos. Factory IO ofrece una versión de prueba gratuita de 30 días que permite acceder a todas las funcionalidades del programa. SIMIT opta por una versión demo de uso indefinido con funcionalidades reducidas. La adquisición de licencias para softwares de simulación de entornos industriales repercutiría en un proyecto inaplicable debido a su alto coste económico. La decisión de emplear dos simuladores de entornos virtuales es ofrecer una mayor perspectiva de los diferentes simuladores utilizados en el sector.

La versión demo de SIMIT posee varias limitaciones siendo las más relevantes las mostradas a continuación [3]. [4]:

- Los proyectos, plantillas y componentes de macros pueden guardarse, pero sólo pueden utilizarse en el ordenador en el que se crearon.
- Se puede utilizar el programa durante todo el tiempo que se desee, pero el tiempo de simulación está limitado a 45 minutos.
- No se puede realizar instantáneas ni grabar la simulación.



El software SIMIT obliga a programar toda acción que realice la instalación, desde el movimiento de los elementos mecánicos hasta la activación de sensores, mientras que este aspecto en Factory IO es transparente al programador. Como el objetivo de este proyecto no abarca una especialización en la programación de entornos virtuales, persiguiendo adquirir nociones básicas y el uso de estos entornos, la forma de programar la simulación en SIMIT no es atractiva para este trabajo.

Debido a las limitaciones de SIMIT antes mencionadas, la gran mayoría de escenarios o entornos se realizan con Factory IO. Sin embargo, se han añadido simulaciones muy simples de SIMIT con el objetivo de inculcar a los alumnos una visión general de dos softwares de simulación de instalaciones industriales con métodos de programación muy distintos.

Cabe destacar que cuando Factory IO se simule con PLCSIM se necesita un bloque de programa en TIA Portal (una FC propia) para que la conexión entre la simulación del autómatas y la estación se realice de forma correcta. Más adelante se abordará esta cuestión clarificando la función de este bloque de programa, así como explicando lo que es un bloque de programa FC.

A continuación, se muestra una tabla de comparación entre los softwares seleccionados para el trabajo en función de su conectividad y tiempo de uso:

PROGRAMAS	VENTAJAS (+) / DESVENTAJAS (-)
PLCSIM	+ Puede simular PLC's de las gamas 1200 y 1500. - Es necesaria una FC para conectar con Factory IO.
PLCSIM Advanced	+ No es necesaria la FC para conectar con Factory IO. - Solo puede simular PLC's de la gama 1500.
Factory IO	+ Se puede conectar con PLCSIM y PLCSIM Advanced. + Puede conectar con PLC's de las gamas 1200 y 1500. - Licencia limitada de 30 días (sin versión demo).
SIMIT	+ Se puede conectar con PLCSIM y PLCSIM Advanced. + Licencia demo ilimitada (funcionalidades reducidas). - Solo puede conectar con PLC's de la gama 1500.

Tabla 2. Comparación del software seleccionado

De todos los softwares seleccionados en el trabajo, únicamente Factory IO no está desarrollado por SIEMENS. Este hecho remarca la fuerte restricción que existe entre la elección de los elementos de control seleccionados y la correspondencia con los softwares de programación.

## 3.2. TIA PORTAL V17

Totally Integrated Automation Portal (TIA Portal), o en español, Portal de Automatización Totalmente Integrado, es una aplicación software potente que permite la programación de autómatas e interfaces hombre-máquina, la configuración de objetos tecnológicos como PC's industriales y la parametrización de accionamientos y arrancadores. Todas sus funcionalidades se implementan en una interfaz única, lo que caracteriza a TIA Portal como una herramienta perfecta para la iniciación en la automatización. [3], [4]

Hasta el momento, y en adelante, se ha hecho referencia a todo el software integrado en TIA Portal como un único elemento de programación o simulación, pero cabe remarcar que realmente, este software está compuesto por diferentes herramientas dependiendo de la versión y paquete de TIA Portal. Los componentes de programación (como STEP 7) y visualización (como WinCC) son editores de un sistema que accede a una base de datos común, no programas independientes. De tal forma, permite guardar todos los datos en un proyecto único con todas las funcionalidades de programación y visualización.

### 3.2.1 Entorno de la interfaz

Los programas o proyectos creados en TIA Portal disponen de dos vistas principales, la vista de portal y la vista de proyecto.

Al inicializar TIA Portal se abre una pantalla de inicio que permite acceder a los diferentes campos para crear un proyecto, abrirlo o modificarlo. A esta apariencia se la conoce como vista de portal.

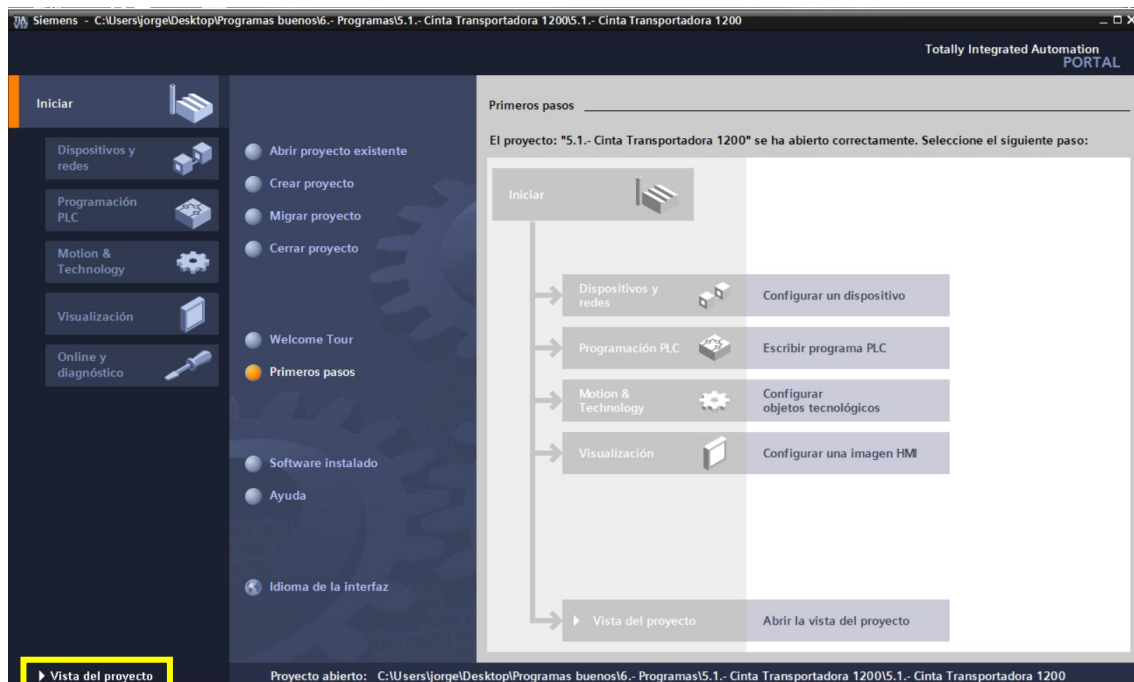


Figura 13. Vista de portal de TIA Portal

Para acceder a la vista de proyecto basta con clicar en la zona marcada con un recuadro amarillo en la figura anterior, desplegándose la siguiente interfaz:

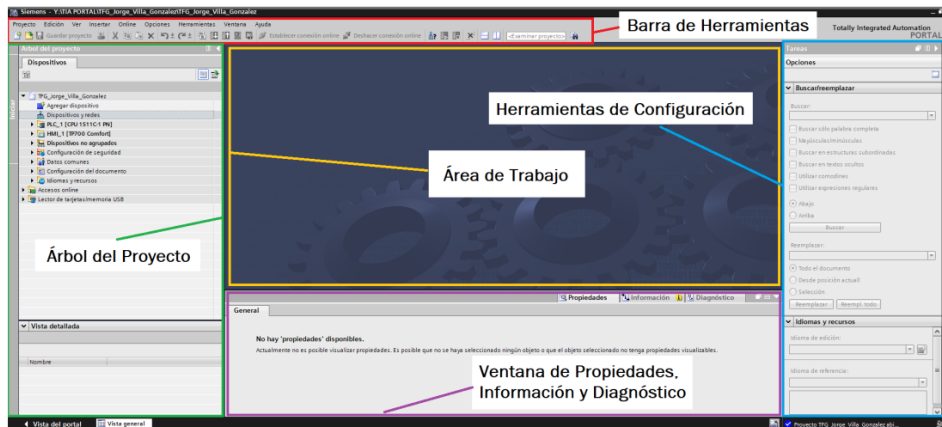


Figura 14. Vista de proyecto de TIA Portal

Las diferentes secciones que conforman la vista de proyectos son:

- **Barra de herramientas:** Contiene las principales funcionalidades relacionadas con el proyecto.

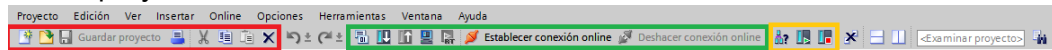


Figura 15. Detalle barra de herramientas de TIA Portal

En la figura anterior, se remarcan en rojo y de izquierda a derecha las acciones de crear, abrir y guardar proyecto, imprimir, cortar, copiar, pegar y borrar. En el recuadro verde, los comandos de compilar, cargar en dispositivo y de dispositivo, iniciar simulación, establecer y deshacer conexión online. De forma análoga, el recuadro amarillo muestra los comandos de visualización de dispositivos accesibles, puesta en marcha y pausa del dispositivo.

- **Árbol del proyecto:** Contiene los dispositivos de control agregados (como pueden ser PLC's y HMI's entre otros), sus programas (compuestos por bloques de programa, tablas de variables, etc.) y las funcionalidades para interconectarlos o configurar su seguridad.
- **Área de trabajo:** Es la zona donde se despliegan las ventanas para la modificación, programación y configuración de los dispositivos de control. Realmente es el área principal de esta vista, puesto que el programa se desarrolla y supervisa desde esta sección, mientras que el resto de las áreas simplemente permiten acceder a opciones de configuración o programación que se emplearán en el área de trabajo.
- **Herramientas de configuración:** Esta sección contiene la familia de instrucciones, test, tareas y librerías empleadas en la programación, modificación y comprobación de los elementos a programar.
- **Ventana de propiedades, información y diagnóstico:** muestra las propiedades, avisos, errores y estado del dispositivo, bloque o función seleccionado.

### 3.2.2 Lenguajes de programación

TIA Portal ofrece los cuatro lenguajes de programación de PLC's contemplados en la norma UNE-EN 61131 y para la gama S7-1500 un lenguaje adicional. Estos lenguajes reciben las siguientes denominaciones:

- **KOP:** Diagrama de contactos, de escalera o ladder (LD).
- **FUP:** Diagrama de bloques de funciones (FBD).
- **AWL:** Lista de instrucciones (AWL).
- **SLC:** Lenguaje estructurado o Structured Control Language (SLC).
- **GRAPH:** Lenguaje GRAFCET.

### 3.2.3 Variables y tipos de datos

Los autómatas programables trabajan con información, en forma digital o analógica, almacenándola en variables mediante agrupaciones binarias. Una variable es un espacio de memoria que almacena un valor, con un tipo de dato concreto, al que se le asigna un nombre y dirección en función del tipo de dato almacenado.

#### Tipos de variables en TIA Portal:

- **Entradas o Inputs (I):** señal de entrada al PLC.
- **Salidas u Outputs (Q):** señal de salida del PLC.
- **Marcas (M):** variables internas de memoria del PLC.

Antes de introducir los principales tipos de datos, necesitamos definir lo que es un bit y un byte. Un bit es la unidad elemental de información que puede tomar los valores de 0 o 1. Un byte es una agrupación de 8 bits, siendo el bit de menor valor el de la derecha y el de mayor valor el de la izquierda. Con un byte se puede representar un número comprendido entre 0 y 255, ambos incluidos.

DATO	DESCRIPCIÓN	BITS	EJEMPLOS
<b>Bool</b>	Booleano (binario)	1 bit	%M6.2
<b>Byte</b>	Byte (8 bits)	8 bits	%MB6
<b>Int/Word</b>	Valores enteros (Integer o entero/Word o palabra)	2 bytes (16 bits)	%MW6
<b>Dint/Dword</b>	Valores enteros (sin decimales) (Double integer o doble entero /Double word o doble palabra)	4 bytes (32 bits)	%MD6
<b>Real</b>	Números en coma flotante (con decimales) (reales)	32 bits	%MD6
<b>Time</b>	Temporizadores	32 bits	%MD6
<b>Char</b>	Cadena de caracteres (Character o carácter)	8 bits	%MB6

Tabla 3. Tipos de datos TIA Portal

Las variables en TIA Portal se pueden agrupar en tablas de observación y forzado para su mejor visualización en el desarrollo o simulación del programa.

### 3.2.4 Bloques de programa

La estructura de programación de autómatas que ofrece TIA Portal está compuesta por cuatro bloques de programación:

- **OB (Organization Blocks o Bloques de organización)**  
Estos bloques son llamados por el sistema operativo (de forma cíclica o en determinadas situaciones como el arranque del PLC) ofreciendo diversas funciones como el bloque principal OB1 o el bloque de ejecución cíclica OB30. Además de poder programar código en ellos, los OB's son los encargados de estructurar el programa llamando, en función de unas condiciones, a los FB's y FC's pero no pueden llamar a otros OB's.
- **FB (Function Blocks o Bloques de funciones)**  
Sirven para compartimentar y reutilizar código. Son bloques con un espacio de memoria asociado a cada llamada, denominado bloque de datos de instancia (DB's), a los que solo se puede acceder desde la FB asociada.
- **FC (Fuctions o Funciones)**  
Sirven para compartimentar y reutilizar código. La diferencia con los FB's es que no poseen una DB propia, no almacenando los valores de las variables internas de la FC tras su ejecución.
- **DB (Data Blocks o Bloques de datos)**  
Son espacios de memoria que almacenan variables. Hay DB's globales, que permiten el acceso a las variables por todos los bloques del programa, y DB's de instancia, estando asociados a un único FB y, por lo tanto, solo se puede leer o escribir variables desde este FB.

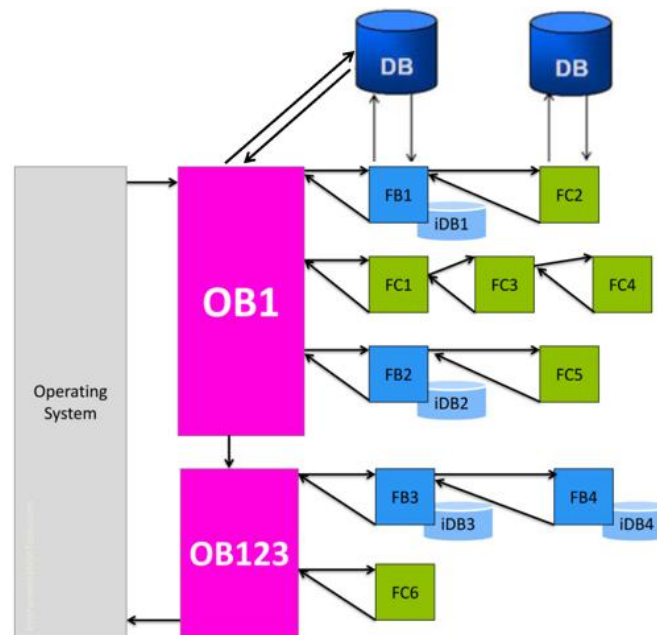


Figura 16. Ejemplo estructura de bloques de programa en TIA Portal [10]

### 3.2.5 Seguridad Integrada

La aplicación software TIA Portal está dotada con un sistema de seguridad denominado “*Know-how*” que impide la copia o modificación, por parte de personal no autorizado, de los bloques de programa.

Después de crear una protección “*Know-how*” para un bloque, ya no se puede acceder al bloque en TIA Portal sin la contraseña. Los bloques de copia de seguridad asociados, creados por el sistema en la carpeta de bloques del sistema, se cifran con la contraseña del usuario y ya no son visibles para el sistema.

Este requisito puede ser muy interesante especialmente para los bloques de seguridad de la sección del programa que contiene la seguridad.

## 3.3. PLCSIM Y PLCSIM ADVANCED

Los sistemas de simulación apoyan el desarrollo de programas y el posterior despliegue en producción. En el mundo de la automatización, un entorno de pruebas simulado acorta los tiempos de puesta en marcha. [3], [4]

Ambos softwares permiten comprobar el correcto funcionamiento del programa del autómatas sin necesidad de disponer del PLC físico mediante la simulación de un controlador virtual. S7-PLCSIM y S7-PLCSIM Advanced funcionan junto con la programación de STEP 7 de TIA Portal, siendo STEP 7 el encargado de programar la lógica de aplicación y descargar la configuración hardware y el programa en los controladores virtuales. Desde allí, se puede ejecutar la lógica del programa, observar los efectos de las entradas y salidas simuladas y adaptar los programas de forma online u offline.



Figura 17. Simulación de un PLC con PLCSIM

Dentro del software básico integrado en TIA Portal se incluye PLCSIM, un programa que permite la simulación de PLC's. S7-PLCSIM permite utilizar todas las herramientas de depuración de TIA Portal (STEP 7), incluida la tabla de observación, el estado del programa, las funciones online y de diagnóstico y otras herramientas. Asimismo, S7-PLCSIM proporciona herramientas exclusivas de su interfaz, como una tabla SIM, un editor de secuencias, un editor de eventos y control del ciclo, aunque estos aspectos no entran en el campo de estudio de este trabajo.

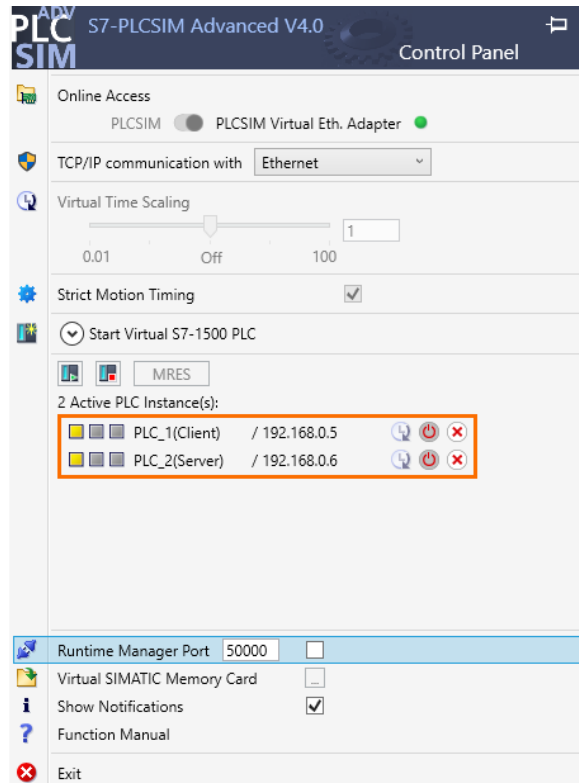


Figura 18. Simulación de dos PLC's con PLCSIM Advanced

S7-PLCSIM Advanced es un producto que proporciona capacidades de simulación avanzadas. A diferencia de S7-PLCSIM, S7-PLCSIM Advanced se puede instalar independientemente de TIA Portal.

La principal diferencia que caracteriza a este simulador es la similitud entre el PLC virtual y el PLC físico, ofreciendo unas capacidades mucho mayores con relación al ámbito de las comunicaciones.

Uno de los factores determinantes para la elección de un simulador u otro, es el controlador que se desee simular. PLCSIM Advanced solo puede simular autómatas de la gama S7-1500, mientras que, con el paquete facilitado por la facultad para el desarrollo de este trabajo, PLCSIM puede simular los de las gamas S7-1200 y S7-1500. Por contraposición, PLCSIM Advanced está mucho mejor dotado en lo referente a comunicación y permite simular varios PLC's y/o HMI's al mismo tiempo y conectarlos con PLC's y HMI's físicos, aspecto que PLCSIM no proporciona.

Es importante saber que es posible instalar S7-PLCSIM y S7-PLCSIM Advanced en la misma máquina, pero no es posible ejecutarlos al mismo tiempo.

### 3.4. FACTORY IO Y SIMIT

Factory IO es una herramienta de simulación de procesos industriales mediante entornos virtuales en tres dimensiones, denominados escenas. Es un software fácil de usar y simular, que brinda la posibilidad de creación de instalaciones o procesos industriales típicos a partir de los elementos industriales ofrecidos por el programa.

[9]

El escenario más común es utilizar Factory IO como plataforma de capacitación de PLC, sin embargo, también se puede utilizar con otros tipos de controladores.

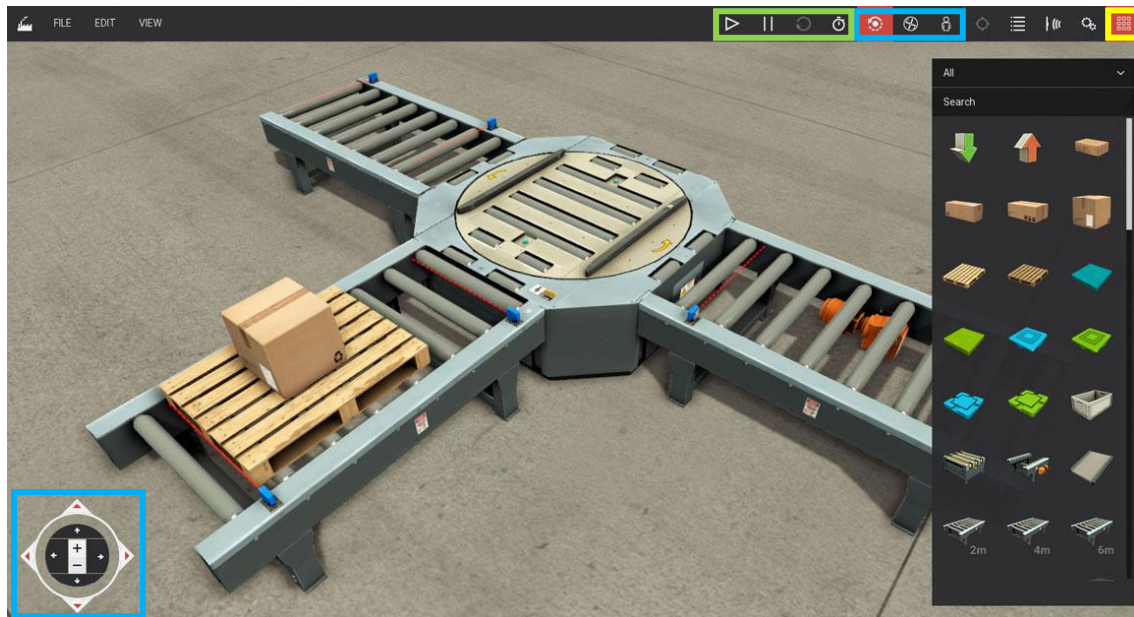


Figura 19. Interfaz de Factory IO

El software dispone de tres tipos de visión (cámara de orbita, cámara voladora y cámara en primera persona) y un panel de navegación que facilitan mucho la interacción con los elementos de la instalación y el desplazamiento por el entorno virtual. En la figura superior se muestran estos elementos en el interior de los cuadriláteros azules.

Para la creación y modificación de escenas, Factory IO incluye un conjunto de elementos industriales típicos estructurados en las siguientes bibliotecas: elementos, carga pesada, carga ligera, sensores, operadores, estaciones, dispositivos de advertencia y pasarelas. Para abrir el desplegable de bibliotecas hay que pulsar en el símbolo contenido en el recuadrado amarillo en la figura anterior. Los principales elementos de este software se han adjuntado en el “Anexo 2”.

La mayoría de los elementos disponen de variables asociadas que se clasifican en dos grupos, sensores (entradas o inputs) o actuadores (salidas u outputs). En Factory IO las variables tienen un nombre que las caracteriza, denominado etiqueta, pudiendo ser datos de tipo bool, int o float (real).

Para verificar el correcto funcionamiento de las escenas sin necesidad de conectarla con un controlador (ya sea simulado o físico) el software ofrece la posibilidad de un control manual. Este consiste en el forzado de etiquetas (modificar su valor en cualquier momento) para suplir la función del controlador.

El control de la instalación en condiciones normales de simulación se realiza por un autómatas (ya sea físico o simulado). Para la comunicación con este, Factory IO dispone de diversos controladores de entrada/salida que se encargan de permitir el intercambio de información con él. Según el tipo de autómatas (marca, modelo, software de simulación, etc.) se selecciona un controlador E/S y se configura con las características del autómatas.

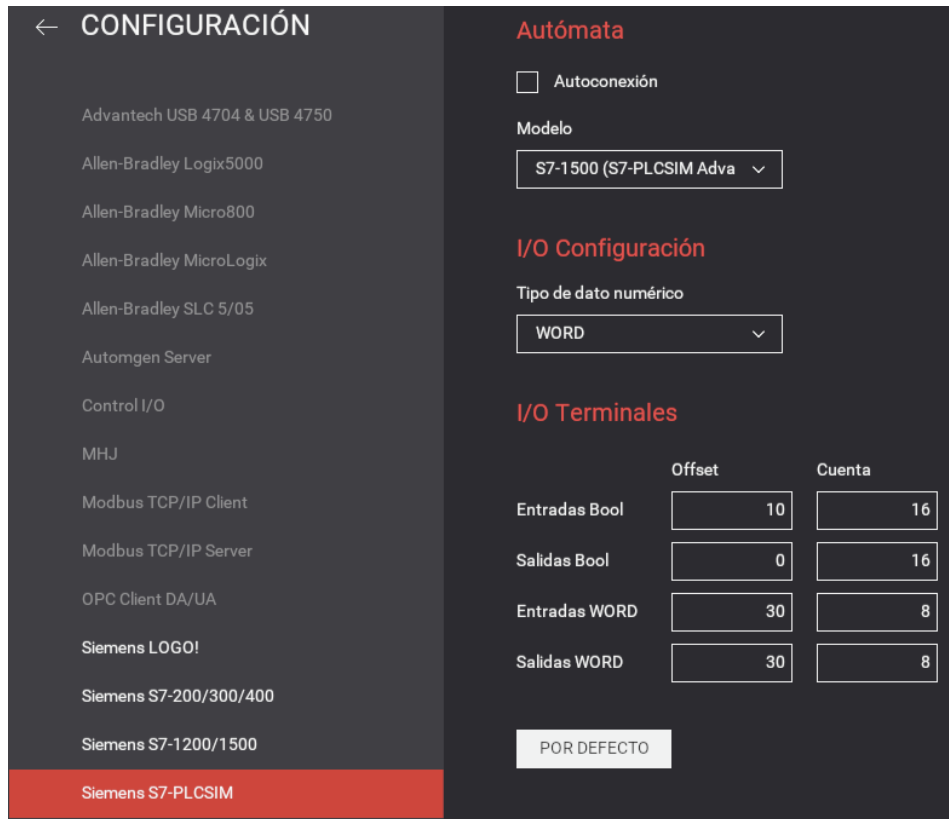


Figura 20. Configuración del controlador E/S de Factory IO

Las etiquetas que deseemos emplear en el PLC deben de estar conectadas a una entrada o salida del controlador E/S de Factory IO. Con el fin de facilitar la simulación, este software ofrece una vista del controlador con todas sus entradas a la izquierda y sus salidas a la derecha.



Figura 21. Vista del controlador E/S en Factory IO

Por otra parte, se empleará otro software de SIEMENS llamado SIMIT. Los sistemas de simulación y entrenamiento SIMIT permiten la formación y la realización de pruebas de proyectos de automatización, al igual que Factory IO. [3], [4]

SIMIT puede usarse como un entorno de simulación en tiempo real de una instalación completa (simulando señales, dispositivos y la respuesta de la instalación ante eventos), como simulador de entradas y salidas de señales de prueba para PLC's y para comprobar el programa del autómatas y modificar el software de automatización. La plataforma o interfaz de SIMIT tiene el siguiente aspecto:

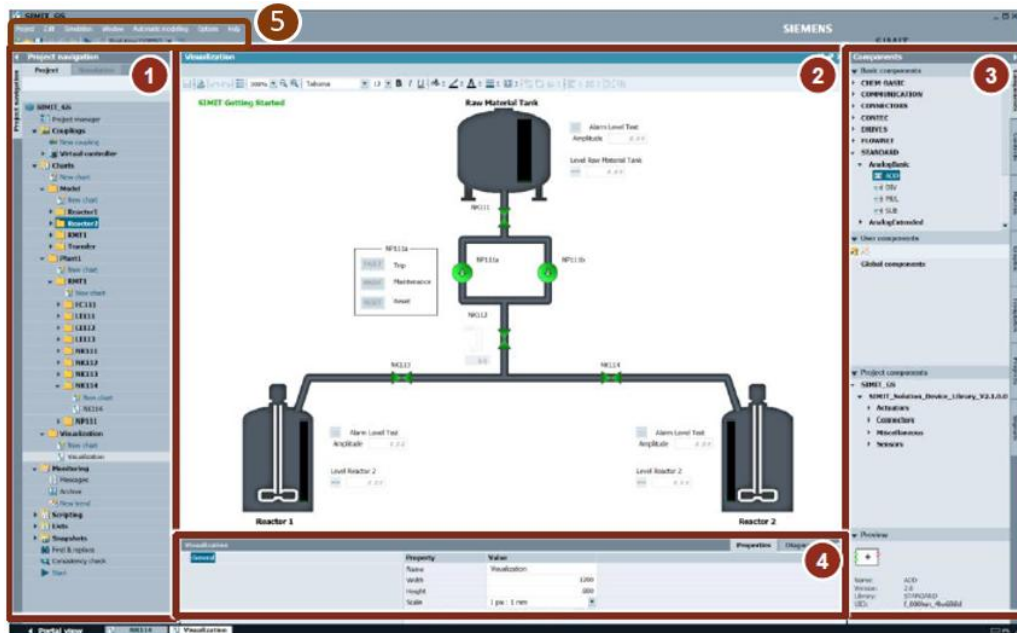


Figura 22. Interfaz de SIMIT [11]

La interfaz está dividida en cinco áreas:

- 1) **Árbol del proyecto:** Muestra el nombre del proyecto en la parte superior y la estructura del proyecto distribuida en carpetas desplegables que contienen el tipo de controlador o las porciones del programa distribuidas en "Charts".
- 2) **Área de trabajo:** Ventana donde se muestran todos los editores de SIMIT, como los gráficos (o charts), scripts o visualizaciones de curvas. Para seleccionar un editor basta con pulsar en un objeto del árbol del proyecto.
- 3) **Herramientas:** Esta sección muestra las familias de componentes, control, macros, gráficos, plantillas, proyectos y señales dependiendo del objeto seleccionado en el área de trabajo.
- 4) **Propiedades:** En esta ventana, el sistema muestra todas las propiedades particulares del objeto seleccionado, como por ejemplo visibilidad, animación, inputs u outputs.
- 5) **Menú y barra de herramientas:** Contiene las principales funciones relacionadas con la simulación y el proyecto, como por ejemplo guardar y abrir un proyecto.

Para crear una simulación se trabaja esencialmente con los siguientes elementos:

- **Gráfico:** Para desarrollar una simulación, se combinan los componentes disponibles en las bibliotecas utilizando el editor de gráficos e introduciendo las claves adecuadas.
- **Visualización:** Ofrece una visión general de las señales de la planta. Las señales se visualizan con controles (objetos de entrada y visualización) y objetos gráficos.
- **Acoplamiento:** Es la interfaz con el sistema de automatización y es necesario para el intercambio de señales. Puede especificar individualmente el ámbito de señales que debe procesar SIMIT.

Este software permite realizar pruebas exhaustivas de proyectos de automatización, así como la puesta en marcha virtual de sistemas, máquinas y procesos en una única plataforma. Además, la plataforma de simulación también se puede utilizar para entornos de formación realistas para instruir a programadores, operarios y personal de mantenimiento.

### 3.5. WINCC

Es una herramienta ofrecida en el paquete de TIA Portal V17, que simula el HMI de forma virtual. Por lo tanto, permite no disponer de un panel de operador físico, siendo sustituido por una simulación virtual idéntica a él en el programa WinCC. La programación del HMI se elabora en TIA Portal, sin embargo, su simulación se realiza con WinCC. [3], [4]

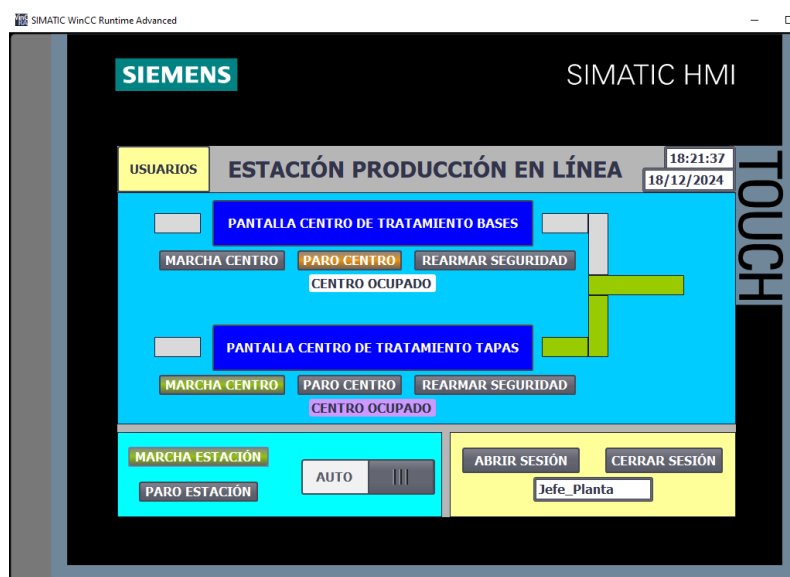


Figura 23. HMI simulado en WinCC

Previamente a la simulación, se debe haber agregado y configurado el panel de operador en el proyecto de TIA Portal, con una dirección IP de la misma red que el autómatas y con una dirección local válida, para su correcta interacción con el PLC.



Para simular el HMI de un proyecto de TIA Portal, se debe seleccionar el HMI en el árbol del proyecto y posteriormente ejecutar el simulador. Al iniciar el simulador, se abre de forma automática WinCC.

WinCC no cuenta con una interfaz propia donde crear un proyecto vinculado al programa de TIA, a diferencia de PLCSIM, sino que ejecuta las funcionalidades ya programadas del HMI.

Para que la interacción entre el HMI y el PLC simulados sea correcta, no solo tienen que estar conectados a la misma red con direcciones IP válidas, sino que además se tienen que estar simulando a la vez. Este dato es importante puesto que, desde un componente se puede variar una variable común a ambos.

### 3.6 CONECTIVIDAD ENTRE PROGRAMAS

Para la simulación de las instalaciones o máquinas automatizadas en este trabajo es necesario conectar varios de los softwares descritos anteriormente.

El programa del autómatas se ejecuta, modifica y revisa desde TIA Portal, software desde el cual se le asigna una dirección IP al PLC. Desde PLCSIM o PLCSIM Advanced se simula el PLC de forma virtual con su dirección IP ya asignada. Desde Factory IO o SIMIT mediante sus controladores de E/S se conectan con el PLC permitiendo el intercambio de información entre el autómatas y el entorno simulado de forma virtual.

En caso de que la instalación esté dotada de un HMI, su programación se realiza en TIA Portal, al igual que su asignación de una dirección IP y la simulación del dispositivo se hace desde WinCC.

Todo el proceso empleado en este trabajo para el correcto intercambio de datos entre los diferentes softwares se ha descrito y adjuntado en el "Anexo 1".

## 4. Actualización de estaciones

Dentro de los objetivos de este trabajo se encuentra la recuperación y actualización de cuatro simulaciones de instalaciones automatizadas realizadas por otro autor. Estas estaciones estaban programadas en TIA Portal V13 y simuladas en Factory IO. La mayoría de las simulaciones (todas menos la última) se habían realizado empleando un PLC físico, por lo que se ha tenido que adaptar el programa para simularse con PLCSIM o PLCSIM Advanced. [\[12\]](#)

### 4.1. ACTUALIZACIÓN DE ARCHIVOS

Para actualizar archivos o proyectos de TIA Portal basta con abrir el archivo, guardado en una versión posterior, con el programa TIA Portal con la versión en la que se quiere trabajar y migrar los proyectos. [\[3\]](#)

En el caso concreto de este proyecto, se tiene que actualizar las estaciones creadas y guardadas en TIA Portal V13 a la versión 17 del mismo software. Para ello se abre el programa TIA Portal V17. En el apartado de “Inicio”, concretamente en “Abrir proyecto existente”, se selecciona el botón “Examinar”, se busca el directorio donde esté guardado el proyecto en versión 13 y se abre. Este proceso se refleja en la siguiente figura:

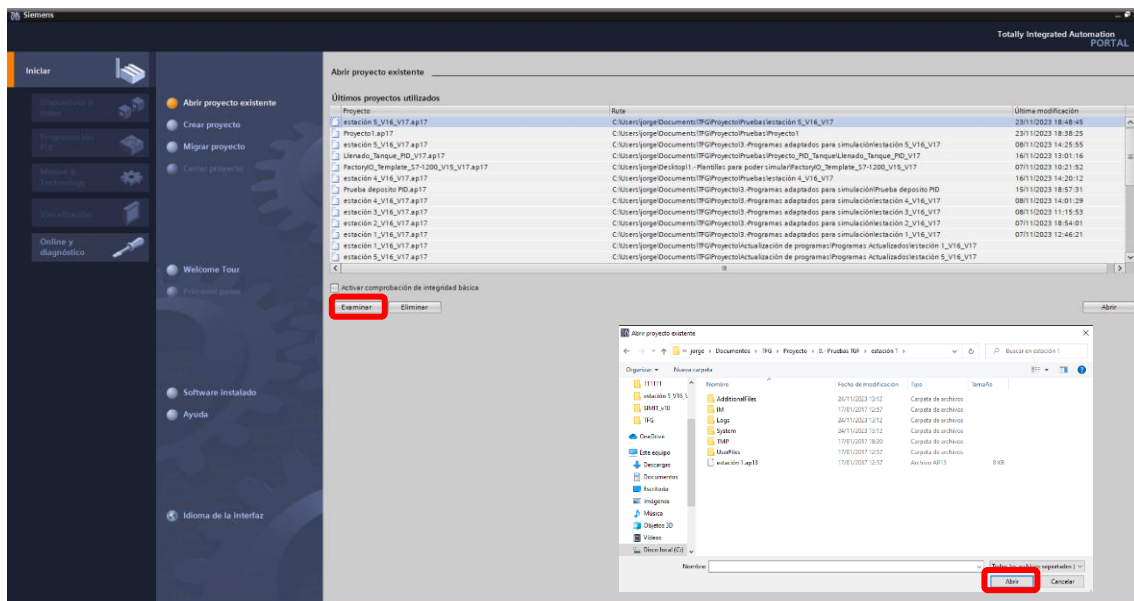


Figura 24. Abrir un proyecto en TIA Portal

Una vez seleccionado el proyecto que se desea actualizar, aparece una ventana de aviso de incompatibilidad de versiones, que indica que el proyecto se actualizará a la versión del programa actual (V17). Se selecciona “Actualizar” y comienza el proceso de actualización de forma automática.

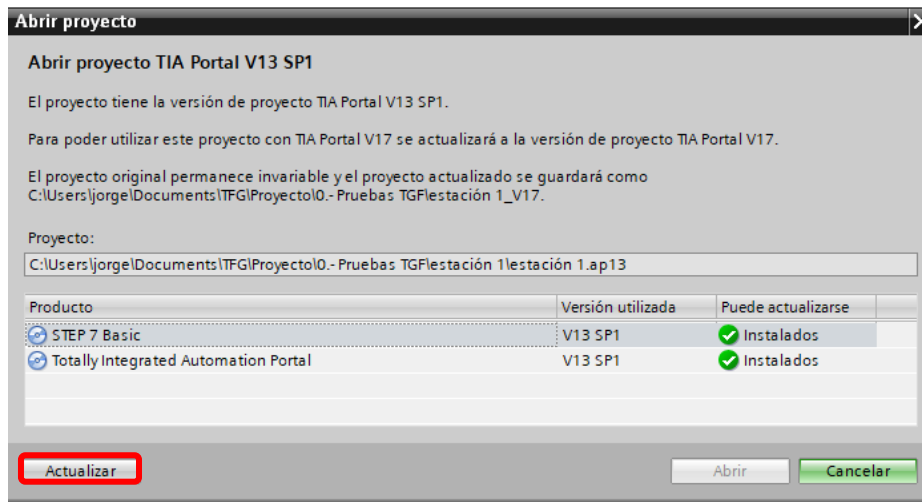


Figura 25. Ventana de aviso de actualización de versión en TIA Portal

Terminado el proceso automático de actualización, salta el siguiente mensaje de aviso:

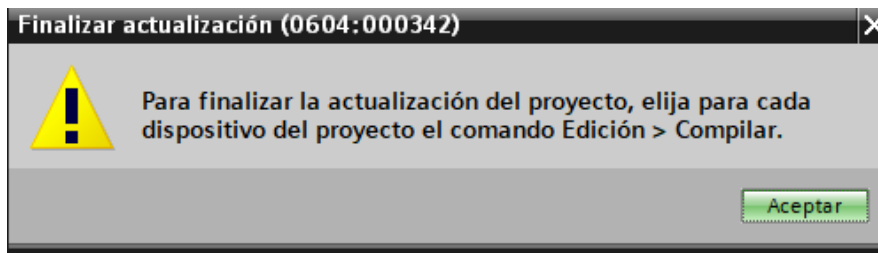


Figura 26. Ventana de finalización de la actualización

En el mismo directorio de la estación original se crea una carpeta con el proyecto actualizado bajo el mismo nombre que el original añadiendo la versión a la que se ha actualizado al final. Además de esta carpeta se crea una copia o backup del proyecto actualizado y un archivo de texto. Estos dos últimos archivos se pueden borrar ya que no proporcionan ningún contenido adicional.

Como se advertía en la figura anterior, es necesario compilar los dispositivos contenidos en el proyecto para que el proceso de actualización se complete. Para ello se abre el proyecto actualizado y se visualiza en pantalla la vista de proyecto.

Se selecciona un dispositivo del proyecto en el árbol del proyecto y se selecciona del desplegable la opción de "Compilar hardware y software". Este proceso se repite para el resto de los dispositivos del proyecto, en el caso de que los hubiera.

## 4.2. DESCARGA DE LA PLANTILLA PARA SIMULAR CON FACTORY IO Y PLCSIM

Las estaciones que se desean actualizar estaban diseñadas para su simulación con un PLC físico. Como el objetivo del proyecto es la simulación de las estaciones sin un PLC físico, sino que se quiere simular este dispositivo con PLCSIM o PLCSIM Advanced, es necesario conocer los requerimientos de conexión entre estos softwares.

Para la correcta comunicación entre PLCSIM y Factory IO es necesario añadir un FC que permita la lectura de señales de entrada de Factory IO y la activación de sus salidas. [9]

Accediendo a la [página oficial de Factory IO](#), en el apartado de “*Tutoriales y ejemplos*” en la categoría de “SIEMENS” se selecciona “S-7 PLCSIM V13-V18” para acceder a un conjunto de plantillas que proporciona esta entidad. Estas plantillas contienen un proyecto de TIA Portal con un PLC, diversas cartas de entradas y salidas y un programa de autómatas compuesto por una única FC. Esta FC está programada en AWL y su función es permitir la lectura y escritura en áreas de memoria mediante los comandos “PEAK” y “POOK”.

Como ya se ha mencionado con anterioridad en la memoria del proyecto, el análisis y comprensión de la programación en AWL no está comprendida en los objetivos de dicho trabajo y, por lo tanto, tampoco se analiza en profundidad en este apartado.

Setting up S7-PLCSIM V13-18

Tutorials & Samples

- Allen-Bradley >
- Schneider >
- Siemens >
- Setting Up >
- Samples >
- S7-1200/1500 >
- S7-PLCSIM V5 >
- S7-PLCSIM V13-18** >
- S7-PLCSIM Advanced >
- S7-200 >
- S7-200 SMART >
- LOGO! >
- MHJ >
- CODESYS >
- Automgen >
- Advantech Wiring >

TIA Portal Template Projects

Note that you must use a TIA Portal template project when connecting to S7-PLCSIM V13-18. **Factory I/O will not be able to communicate with S7-PLCSIM otherwise.**

✓ Downloads for TIA Portal V13

- FactoryIO\_Template\_S7-1200\_V13.zip
- FactoryIO\_Template\_S7-1500\_V13.zip

✓ Downloads for TIA Portal V14

- FactoryIO\_Template\_S7-1200\_V14.zip
- FactoryIO\_Template\_S7-1500\_V14.zip

✓ Downloads for TIA Portal V15 (compatible with V16/17/18)

- FactoryIO\_Template\_S7-1200\_V15.zip**
- FactoryIO\_Template\_S7-1500\_V15.zip**

Setting up S7-PLCSIM with TIA Portal

1. **Download** and **Open** the template project corresponding to the TIA Portal version and PLC family you will be simulating.
2. **Save** the project with a new desired name.

Figura 27. Descarga de la plantilla proporcionada por Factory IO

### 4.2.1 Copiar y pegar bloques de programa

Para incluir la FC obtenida de la página de Factory IO es necesario aprender a copiar y pegar bloques de programa.

Se abre el proyecto descargado de Factory IO que contiene la plantilla que se desea incluir en el programa ya actualizado y se repite el proceso de actualización que ya se ha realizado con la estación para actualizar la plantilla de la versión 15 a la 17. Una vez terminado el proceso de actualización, se abre la vista de proyecto del programa y se despliega el árbol del proyecto. Se selecciona la FC que permite el intercambio de información entre PLCSIM y Factory IO y se clicla la opción de copiar.

Se abre el programa de la estación que se quiere simular en Factory IO, se despliegan los campos del PLC en el árbol del proyecto, se pulsa en bloques de programa y se selecciona la opción de pegar. Es importante resaltar que para que se ejecute la FC se tiene que llamar desde una OB.

Para terminar el proceso, se compila el hardware y el software del PLC del proyecto y guardamos.

## 4.3. CONFIGURACIÓN DEL CONTROLADOR DE FACTORY IO

Como ya se ha comentado con anterioridad, las estaciones actualizadas estaban diseñadas para su simulación con un PLC físico y con el software de Factory IO. Con el fin de permitir el correcto intercambio de información entre Factory IO y PLCSIM es necesario modificar la configuración del controlador de entradas y salidas de Factory IO.

Para ello se abre el controlador de la estación de Factory IO a través del campo de “Archivo” en la barra de herramientas y pulsando en “Drivers”. A continuación, se selecciona el tipo de controlador o de software de simulación del PLC, en este caso “Siemens S7-PLCSIM”.

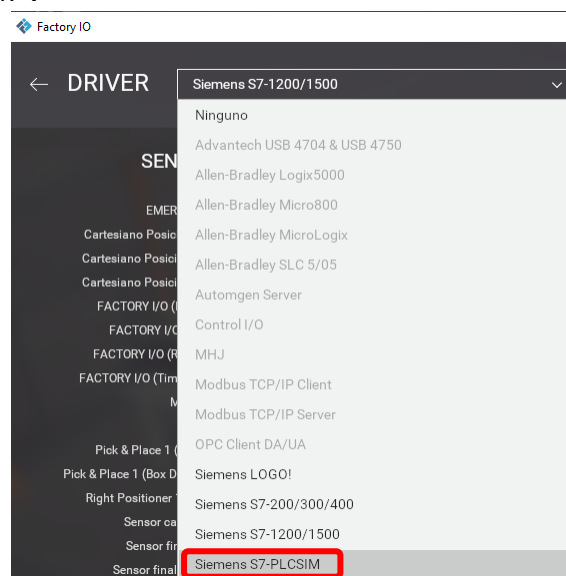


Figura 28. Seleccionador del driver de Factory IO

En la configuración, se selecciona el tipo de controlador, el tipo de dato numérico y el número de entradas y salidas con su offset correspondiente.

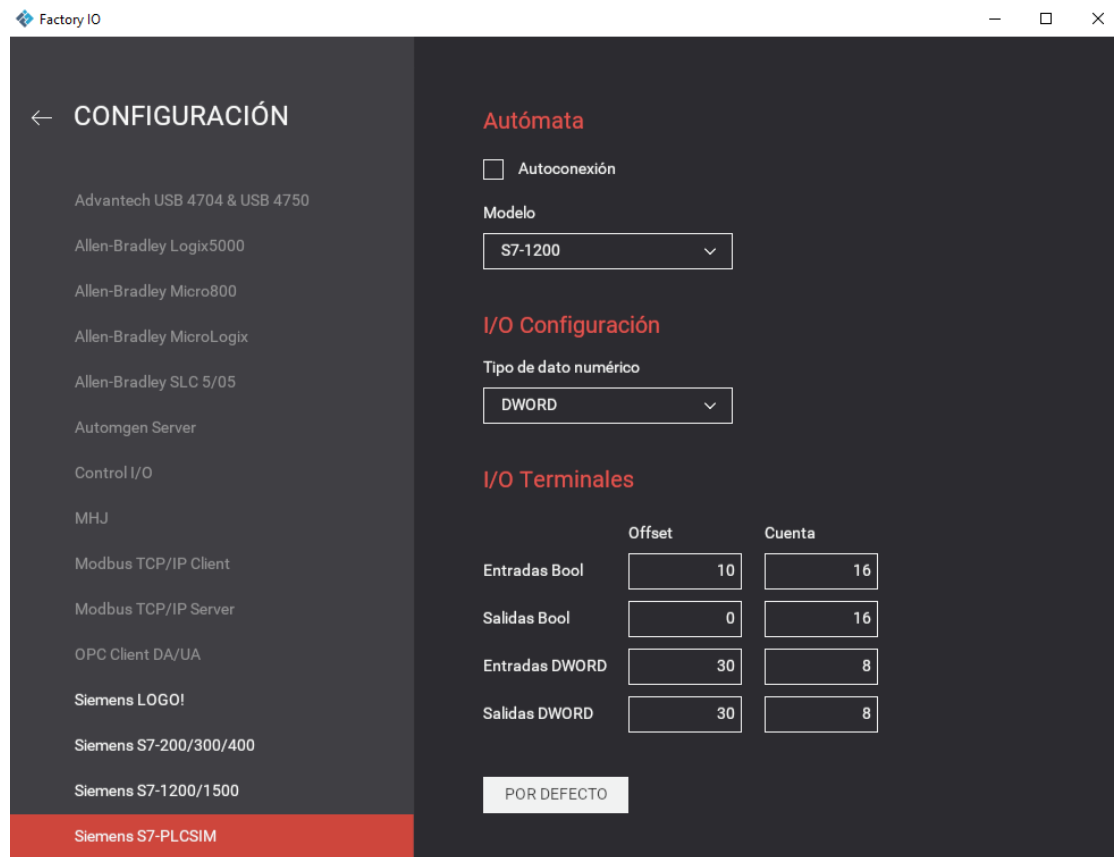


Figura 29. Configuración del controlador de Factory IO

Es necesario remarcar, que para el correcto intercambio de información entre el controlador de E/S del simulador de la estación real y del PLC simulado, es necesario que las direcciones de entradas y salidas de ambos se correspondan. Una de las estaciones recuperadas para la actualización determinaba que las entradas y salidas analógicas tenían un offset de 100, las entradas digitales uno de 10 y las salidas digitales uno de 0. Factory IO no permite seleccionar un offset de 100 con un PLC simulado, por lo tanto, se selecciona uno menor.

Al cambiar las direcciones de las entradas y salidas del controlador de Factory IO, se debe hacer lo correspondiente en las direcciones del programa del autómata en TIA Portal.

#### 4.3.1 Cambio de variables en Factory IO y TIA Portal

En primer lugar, se debe conocer que el autómata posee una cantidad concreta de direcciones para las entradas y salidas en un rango determinado. Esta información es visible en la ventana de propiedades, información y diagnóstico. Para acceder a ella se debe pulsar en el PLC del árbol del proyecto, a continuación, en “Configuración de dispositivos” y, por último, hacer doble click en el PLC del área de trabajo.

Llegados a este punto, en la ventana de propiedades, información y diagnóstico, concretamente en el apartado general del dispositivo, se puede acceder a la información de las variables de entrada y salida entre otras cosas.

El proceso descrito anteriormente se refleja en la siguiente figura:

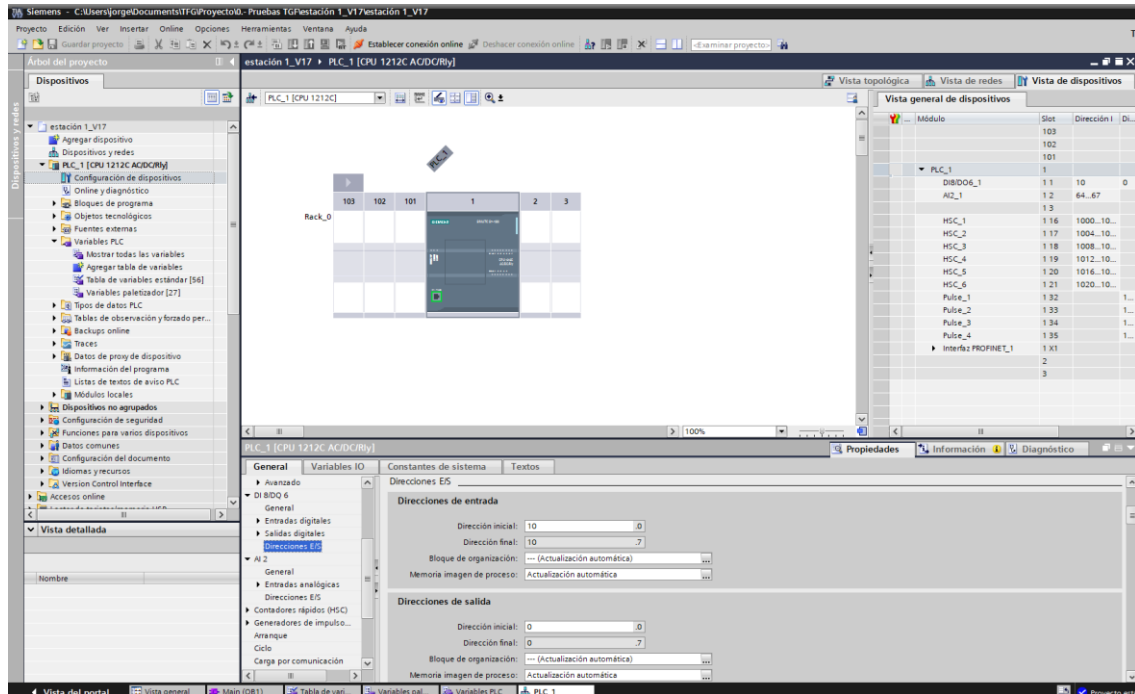


Figura 30. Propiedades del controlador de TIA Portal

Para el correcto funcionamiento de una estación con elementos físicos, es necesario que el número de entradas y salidas se corresponda con la cantidad de variables de E/S que se emplean en el programa, así como su dirección. En caso de no ser suficientes, se deberían de añadir módulos E/S para ampliar la capacidad del PLC. Por lo tanto, se debe realizar la comprobación de las direcciones donde empiezan los canales de entrada y salida del PLC y de sus módulos y cambiarla para que se correspondan con las de las variables.

En el caso que nos compete, al ser una simulación sin ningún elemento físico, el uso de entradas y salidas no comprendidas en la configuración del PLC o de sus módulos de ampliación, simplemente provocará un aviso en el programa de TIA Portal, pero no afectará en ningún aspecto adicional, siempre que sus direcciones si se correspondan.

Al no trabajar con una estación real, sino con un gemelo digital de esta, las lecturas de las señales de entrada del PLC y de Factory IO pueden entrar en conflicto lo que provoca un mal funcionamiento de la estación, aunque el programa sea correcto. Es por ello, que en las prácticas no se añaden módulos de entrada y salida adicionales o no se emplean las direcciones de entrada del PLC para sensores, solo se emplean para botones. Sin embargo, sí que se sabe que, en las estaciones reales, se debe tener en cuenta esta correlación del número y dirección de las E/S y no solo de la dirección.

Una vez se conozcan las direcciones que se desean emplear, se deben de cambiar las direcciones de la tabla de variables oportuna, asignándoles se valor correspondiente.

Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
1	Sensor inicio cinta	Variables paletiz. Bool	%I0.0					
2	Sensor final cinta	Variables paletizador Bool	%I0.1					
3	Sensor pistón	Variables paletizador Bool	%I0.2					
4	Sensor inicio rodillos	Variables paletizador Bool	%I0.3					
5	Sensor final rodillos	Variables paletizador Bool	%I0.4					
6	START	Variables paletizador Bool	%I1.0					
7	STOP	Variables paletizador Bool	%I1.1					
8	EMERGENCIA	Variables paletizador Bool	%I1.2					
9	Cartesiano Posición X	Variables paletizador Real	%ID112					
10	Cartesiano Posición Y	Variables paletizador Real	%ID116					
11	Cartesiano Posición Z	Variables paletizador Real	%ID120					
12	Motor cinta	Variables paletizador Bool	%Q0.0					
13	Neumática pistón	Variables paletizador Bool	%Q0.1					
14	Succion	Variables paletizador Bool	%Q0.2					
15	Giro pinza	Variables paletizador Bool	%Q0.3					
16	Motor rodillos	Variables paletizador Bool	%Q0.4					
17	Rodillo parada	Variables paletizador Bool	%Q0.5					
18	Funcionando	Variables paletizador Bool	%Q0.6					
19	señalización marcha	Tabla de variables e. Bool	%Q1.0					
20	señalización paro	Tabla de variables e. Bool	%Q1.1					
21	Cartesiano Desplazamiento X	Variables paletizador Real	%QD100					

Figura 31. Tabla de variables del PLC

Realizado el cambio de las direcciones en TIA Portal, se vuelve al controlador de Factory IO y se asocian las etiquetas de las variables de entrada y salida empleadas a las mismas direcciones de la tabla de variables.

SENSORES		ACTUADORES	
EMERGENCIA	Sensor inicio cinta %I0.0	%Q0.0	Motor cinta
Cartesiano Posición X(V)	Sensor final cinta %I0.1	%Q0.1	Neumática pistón
Cartesiano Posición Y (V)	Sensor capacitivo %I0.2	%Q0.2	succion
Cartesiano Posición Z (V)	Sensor inicio rodillos %I0.3	%Q0.3	Giro pinza
FACTORY I/O (Paused)	Sensor final cinta %I0.4	%Q0.4	Motor rodillos
FACTORY I/O (Reset)	%I0.5	%Q0.5	Rodillo parada
FACTORY I/O (Running)	%I0.6	%Q0.6	
FACTORY I/O (Time Scale)	%I0.7	%Q0.7	
MARCHA	MARCHA %I1.0	%Q1.0	Señalización marcha
PARO	PARO %I1.1	%Q1.1	Señalización paro
Pick & Place 1 (C Limit)	EMERGENCIA %I1.2	%Q1.2	
Pick & Place 1 (Box Detected)	%I1.3	%Q1.3	
Right Positioner 1 (Limit)	%I1.4	%Q1.4	
Sensor capacitivo	%I1.5	%Q1.5	
Sensor final cinta	%I1.6	%Q1.6	
Sensor final rodillos	%I1.7	%Q1.7	
Sensor inicio cinta	%ID30 (REAL) %QD30	Cartesiano Desplazamiento X(V)	
Sensor inicio rodillos	%ID34 (REAL) %QD34	Cartesiano Desplazamiento Y(V)	
Sensor pistón	%ID38 (REAL) %QD38	Cartesiano Desplazamiento Z(V)	
Cartesiano Posición X(V)	%ID42 (REAL) %QD42		
Cartesiano Posición Y (V)	%ID46 (REAL) %QD46		
Cartesiano Posición Z (V)	%ID50 (REAL) %QD50		
	%ID54 %QD54		
	%ID58 %QD58		

Figura 32. Detalle de información adicional sobre las direcciones

Es importante remarcar que para actualizar la versión de un programa de TIA Portal, no es suficiente con cambiar la versión del programa. En adición, se deben revisar todos los aspectos relacionados con los dispositivos de control (como son el PLC y el HMI, para este trabajo en concreto), las direcciones de las variables o ciertos comandos.



Llegados a este punto, se podría afirmar que se han realizado todos los pasos para el proceso de actualización general de las estaciones. Sin embargo, dependiendo de si el proyecto de TIA Portal contiene determinados elementos o comandos es necesario modificarlos de forma específica.

Este proceso se ha realizado en las cuatro estaciones que se debían actualizar.

Al actualizar el proyecto, se permite que el programa de una versión más moderna sea capaz de ejecutar, modificar o visualizar el proyecto guardado en una versión posterior.

No obstante, los dispositivos como los HMI (empleado en la estación 4) no actualizan su versión de forma automática. Además, al emplear una versión de programa más nueva es posible que algún comando, como es el caso de PID\_Compact (empleado en la estación 3), tenga nuevas restricciones que en la versión del programa original no tenía.

El proceso de actualización específico para estas dos situaciones se describe en los siguientes apartados junto con la estación en la que se emplean.

## 4.4. ESTACIÓN 1: ESTACIÓN PALETIZADORA

### 4.4.1 Objetivos de aprendizaje

Es una estación con un nivel de dificultad medio. Tiene como objetivo el aprendizaje de la compartimentación del programa del autómatas, la utilidad de la OB de arranque (OB100), la utilización de variables de tipo array y un acercamiento al mundo de la robótica mediante el uso y programación de un robot cartesiano encargado del “*Pick and Place*” de las cajas.

- **Conocimientos que adquirir en TIA Portal:**
  - **Variables:** Uso de variables de tipo array.
  - **Programa:** Compartimentación del programa en bloques de programa FC, FB, DB y utilidad de la OB de arranque (OB100).
  - **Conectividad:** Tipo de PLC, direccionamiento IP y conectividad con PLCSIM y Factory IO.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Robot cartesiano y elementos básicos.
  - **Conectividad:** Conectar con PLCSIM.

#### 4.4.2 Descripción de la estación

Se trata de una estación paletizadora automática cuyo objetivo es colocar seis cajas en un palet de una determinada manera mediante el uso de un robot cartesiano con una ventosa de succión en su extremo.

La disposición de la estación es la siguiente:



Figura 33. Vista general. Escena paletizadora

Posee una cinta transportadora, por donde llegan las cajas, dotada con dos sensores difusos y con un posicionador de elementos ubicado al final de ella, para situar la caja en la posición correcta donde lo agarrará el robot mediante una ventosa de succión. De forma perpendicular a ella está emplazado un transportador de rodillos dotado con dos sensores difusos, uno donde entran los pallets y el otro en la posición donde se colocarán las cajas. A la mitad de este transportador se encuentra el robot cartesiano y un rodillo de parada que detecta que el pallet está situado en la posición correcta para la colocación de cajas. Los rodillos se detienen hasta que se colocan todas las cajas sobre el pallet. Posteriormente, el rodillo de parada se retrae, permitiendo el avance del pallet cargado y el transportador de rodillos vuelve a activarse.

Para gobernar el comportamiento de la estación hay un cuadro eléctrico dotado de un botón de marcha, uno de paro y una seta de emergencia con enclavamiento. De tal forma que si se pulsa el botón de marcha el proceso automático comienza y si se pulsa el botón de paro o la seta de emergencia la estación completa se detiene. Para reanudar la marcha tras pulsar el botón de paro basta con pulsar el botón de marcha de nuevo. En caso de que se pulse la seta de emergencia, para reanudar la marcha es necesario girar la seta de emergencia para quitar el enclavamiento de la señal de emergencia y posteriormente pulsar el botón de marcha.

Cabe destacar, que para el manejo del robot cartesiano se emplean arrays para el almacenamiento de las posiciones y variables analógicas, así como su escalado, en el programa del autómatas para el manejo del robot.

El programa del autómatas está dividido en diferentes bloques de programa OB, FC, FB y DB, conteniendo y estructurando la programación del PLC de la siguiente manera:

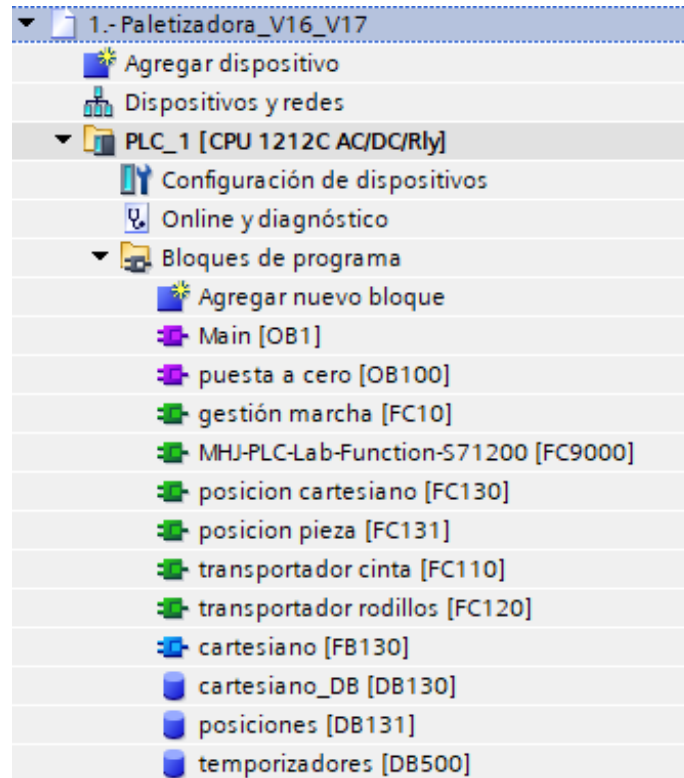


Figura 34. Distribución de bloques. Estación paletizadora

#### 4.4.3 Corrección de errores

Una vez actualizada esta estación, se detectó un fallo en la colocación de la última caja sobre el pallet. Este problema supuso el estudio y comprensión de toda la programación del autómatas para encontrar un remedio a este defecto.

El funcionamiento de los rodillos estaba condicionado a variables temporales y no a señales de sensores. Es por ese motivo, por el cual la única forma de modificar el tiempo entre la colocación de una caja y el avance de los rodillos sin introducir nuevos elementos en la estación industrial simulada en Factory IO era haciendo uso de un temporizador.

Considero que es una solución poco óptima a nivel industrial, pero cumple con su finalidad y a nivel docente, que es el principal interés de este trabajo, no es una mala solución si se pretende conservar lo máximo posible la estructura del programa original.

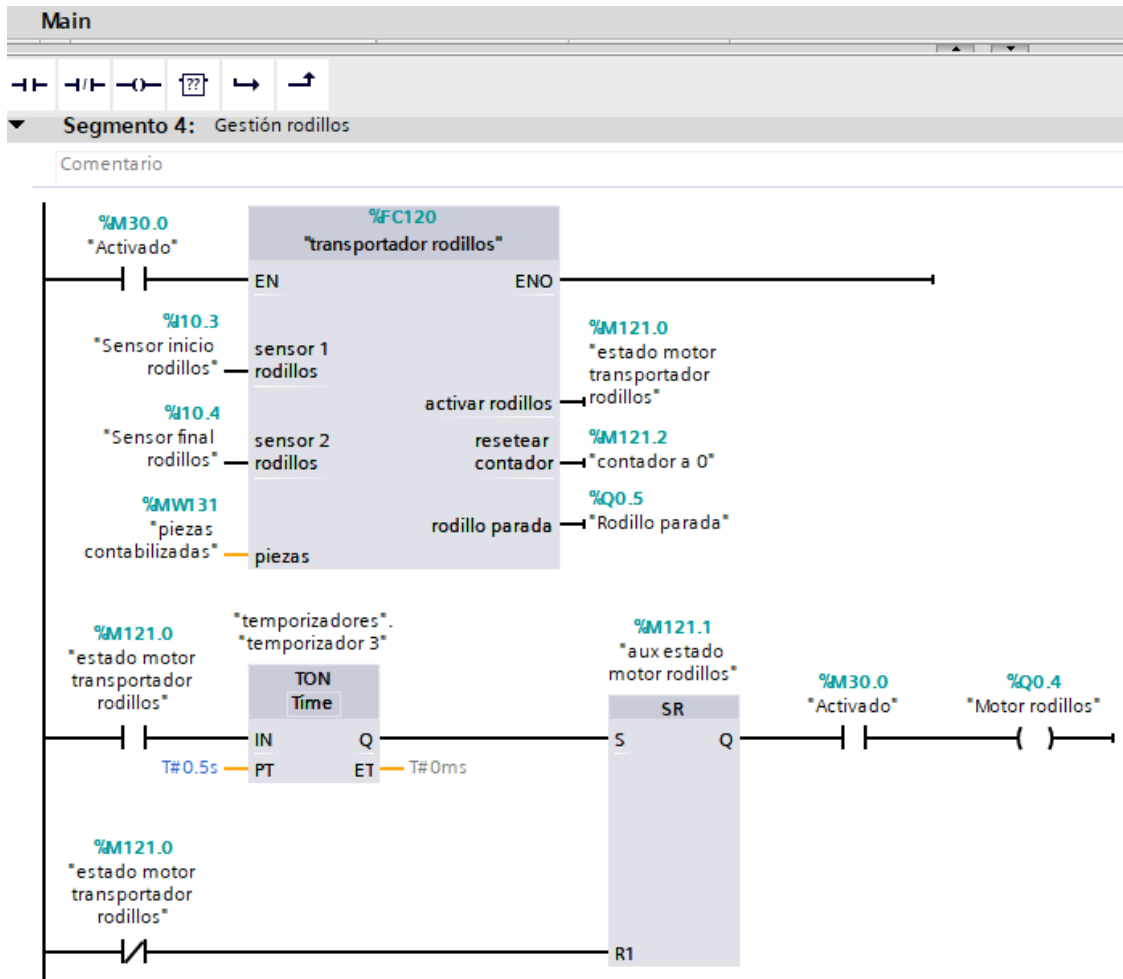


Figura 35. Modificación segmento 4 main. Estación paletizadora

Mediante el uso de un retaso a la conexión en el segmento 4 del main del programa, se consiguió subsanar dicho fallo, modificando lo menos posible el programa original.

## 4.5. ESTACIÓN 2: ESTACIÓN DISTRIBUIDORA

### 4.5.1 Objetivos de aprendizaje

A nivel conceptual es una estación muy similar a la anterior, con un nivel de dificultad medio y con objetivos muy parecidos. Como conceptos a trabajar e interiorizar se pueden destacar la compartimentación de código, la utilidad de la OB de arranque (OB100), el uso de robots de dos ejes encargados del "Pick and Place" de las piezas y el tratamiento de sus variables analógicas con los comandos MUL, DIV y ROUND.

La diferencia más apreciable con la anterior estación, obviando que es una instalación totalmente diferente que requiere de un programa de control específico, es el uso de robots de dos ejes en vez de un robot cartesiano. Por lo tanto, dota al usuario de más herramientas para la interacción entre el ámbito de la automatización y el de la robótica. Además, emplea operaciones matemáticas simples para el escalado de las variables analógicas.

- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** RESET\_BF, MUL, DIV y ROUND.
  - **Variables:** Entradas, salidas y variables internas de tipo analógico y su escalado mediante operaciones matemáticas simples.
  - **Programa:** Compartimentación del programa en bloques de programa FC, FB, DB y utilidad de la OB de arranque (OB100).
  - **Conectividad:** Tipo de PLC, direccionamiento IP y conectividad con PLCSIM y Factory IO.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Robots de dos ejes y elementos básicos.
  - **Conectividad:** Conectar con PLCSIM.

#### 4.5.2 Descripción de la estación

La estación se encarga de forma automática de la separación de piezas de dos colores diferentes, que entran por una misma cinta transportadora, y su colocación en cajas con piezas de un solo color mediante un robot de dos ejes.

La distribución de la estación se muestra a continuación:

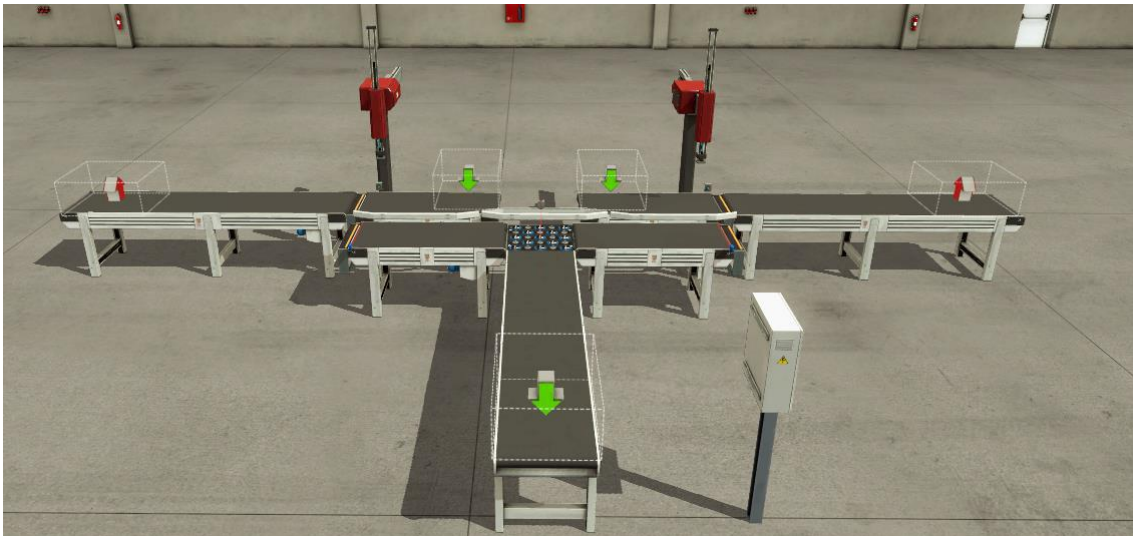


Figura 36. Vista desde arriba. Escena distribuidora

La escena de Factory IO está compuesta por un panel de operador, un conjunto de cintas transportadoras en forma de 'T', un clasificador de ruedas emergentes para modificar el encaminamiento de las piezas en función de su color, cuatro rodillos de parada, dos robots de dos ejes, sensores capacitivos y difusos y un sensor de visión para detectar el color de las piezas para su separación.

Las piezas de diferentes colores entran de forma aleatoria por una cinta que tiene ubicado en su extremo un colocador de ruedas emergentes y un sensor de visión, encargados de distribuir las piezas a izquierda o derecha en función de su color. Una vez separadas las piezas en dos ramas, el comportamiento de ambas es idéntico.

Las cintas con las piezas ya separadas avanzan hasta toparse con un rodillo de parada. Un robot de dos ejes se encarga de agarrar la pieza mediante una ventosa de succión y colocarla en una caja situada en una cinta paralela. Una vez que la caja está llena, el rodillo de parada que impedía su avance se retrae, permitiendo así el avance de la caja llena que sale por el final de la cinta.

Para gobernar el comportamiento de la estación hay un cuadro eléctrico dotado de un botón de marcha, uno de paro, una seta de emergencia con enclavamiento, dos displays que muestran el número de cajas de cada color tratadas en la estación hasta el momento y un botón de reinicio de los contadores asociados a los displays.

De forma análoga a la estación anterior, si se pulsa el botón de marcha el proceso automático comienza y si se pulsa el botón de paro o la seta de emergencia la estación completa se detiene. Para reanudar la marcha tras pulsar el botón de paro basta con pulsar el botón de marcha de nuevo. En caso de que se pulse la seta de emergencia, para reanudar la marcha es necesario quitar el enclavamiento de la señal de emergencia y posteriormente pulsar el botón de marcha.

Para el control del posicionamiento del extremo de los robots de dos ejes es necesario el uso de variables analógicas, así como su escalado para emplear el valor en el programa del autómatas.

El programa del autómatas está dividido en diferentes bloques de programa OB, FC, FB y DB, conteniendo y estructurando la programación del PLC.

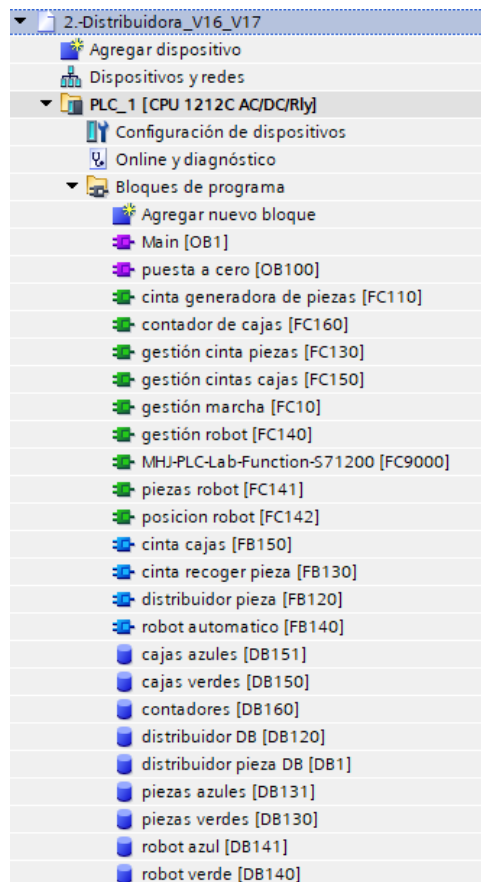


Figura 37. Distribución de bloques. Estación distribuidora

### 4.5.3 Corrección de errores

Al realizar la simulación de la estación, ya actualizada, se hallaron diversos problemas relacionados con los contadores de cajas y la activación de las cintas situadas a izquierda y derecha del colocador de ruedas emergentes. Se cree que estos fallos son debidos a que la versión del programa desde la que se partió a la hora de hacer la actualización no era la final, sino una intermedia sin acabar. Tras la comprensión y estudio del programa del autómatas y del simulador de la estación real, se realizaron diversas modificaciones en el programa del PLC y en Factory IO para solucionar los errores relativos a las cintas transportadoras y los rodillos de parada.

Las modificaciones de Factory IO eran sobre todo configuraciones de los elementos de la estación relativos a tipos de variables o a tiempos de creación de las piezas.

Una vez modificada la configuración de la estación virtual, se hicieron pequeños cambios en el programa para subsanar los problemas con las cintas y los contadores.

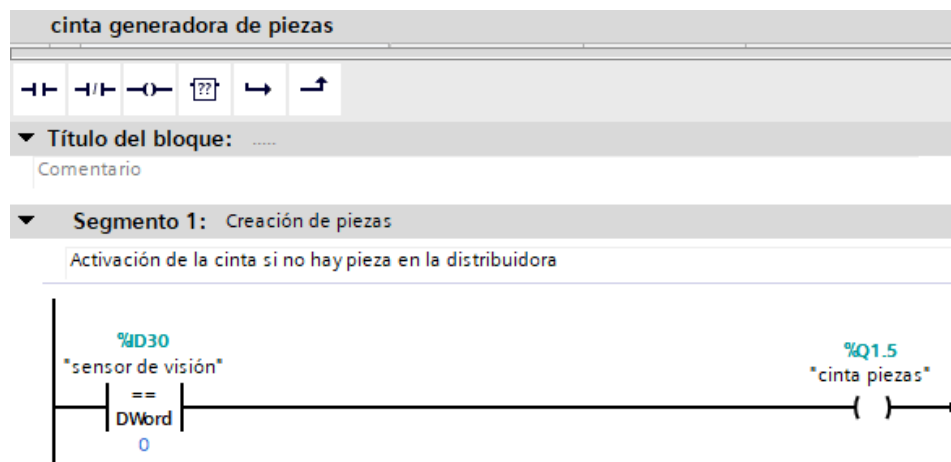


Figura 38. Modificación FC cinta generadora de piezas. Estación distribuidora

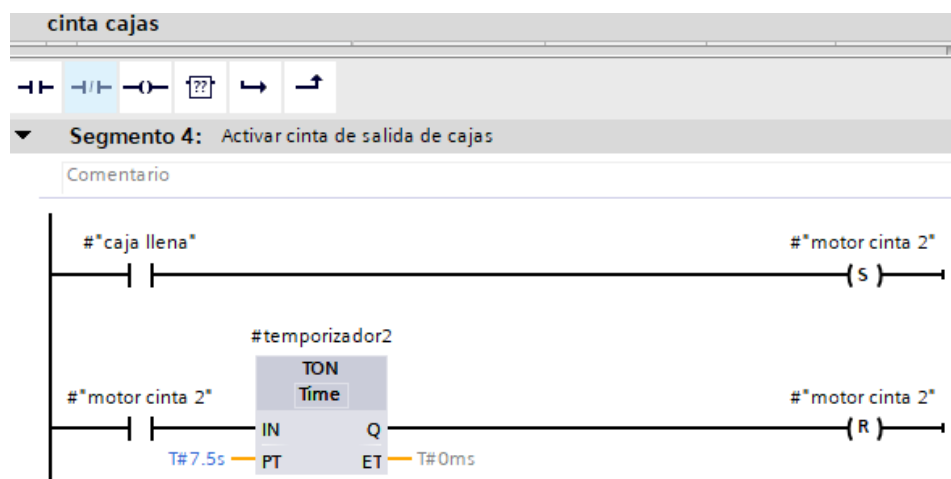


Figura 39. Modificación FC cinta cajas. Estación distribuidora

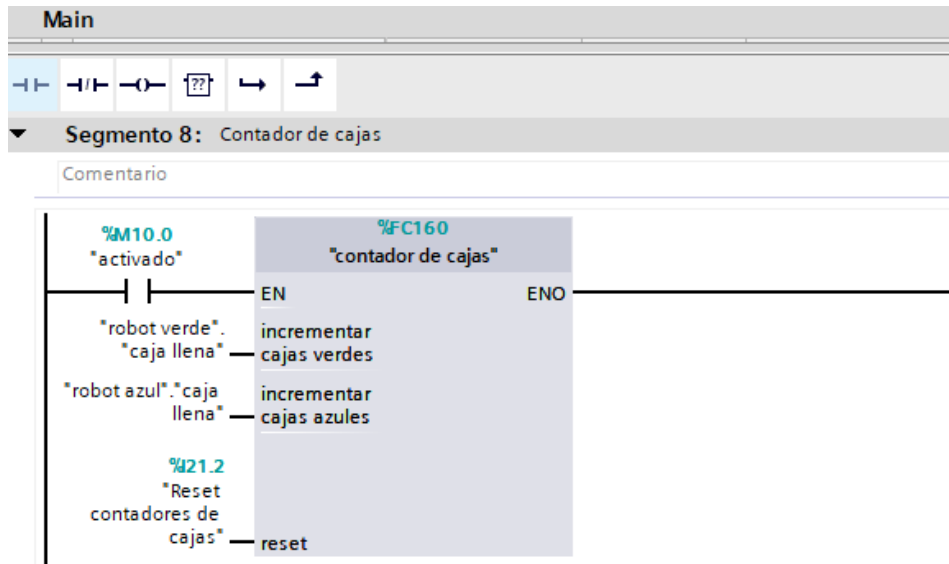


Figura 40. Modificación contador en el main. Estación distribuidora

## 4.6. ESTACIÓN 3: DEPÓSITO DE LÍQUIDO

### 4.6.1 Objetivos de aprendizaje

Es una estación con un nivel de dificultad medio-alto. Tiene como objetivo principal aprender a programar y simular un PID, encargado de controlar el nivel de líquido en un depósito, empleando una OB cíclica (OB30). Como objetivos secundarios, pretende afianzar los conocimientos relacionados con la compartimentación del programa del autómatas, la utilidad de la OB de arranque (OB100), el uso de variables analógicas y su escalado mediante los comandos NORM\_X y SCALE\_X.

- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** PID\_Compact, NORM\_X y SCALE\_X.
  - **Variables:** Entradas, salidas y variables internas de tipo analógico y su escalado.
  - **Programa:** Utilidad de una OB cíclica (OB30), control PID, compartimentación del programa en FC, FB, DB y utilidad de la OB de arranque (OB100).
  - **Conectividad:** Tipo de PLC, direccionamiento IP y conectividad con PLCSIM Advanced y Factory IO.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Estación del tanque y sus válvulas de entra y salida.
  - **Conectividad:** Conectar con PLCSIM Advanced.

## 4.6.2 Descripción de la estación

La estación tiene como finalidad el control de nivel de un tanque mediante la apertura o cierre progresivo de una válvula situada a la entrada y una válvula situada a la salida, controladas cada una de ellas por un regulador PID.

La escena tiene un panel de operador para gobernar la instalación, un depósito, una válvula situada en la parte superior que controla el caudal de líquido que entra al depósito (válvula de entrada) y una válvula situada en la parte inferior que controla la salida del líquido (válvula de salida).

En el cuadro eléctrico se ubica un botón para el llenado, un botón para el vaciado, un display que muestra los litros actuales en el depósito, un potenciómetro para seleccionar los litros deseados en el depósito y un display que muestra este valor.

Por lo tanto, salvo que la válvula de entrada esté totalmente cerrada, siempre entrará líquido por la parte superior y de forma análoga, siempre saldrá algo de líquido salvo que la válvula de salida este totalmente cerrada. De esta manera, controlando el valor de apertura de ambas válvulas se puede controlar el nivel del tanque. Este control se realiza mediante dos PID's (usando dos instrucciones PID\_Compact en el programa del autómatas), uno para cada válvula.

La escena antes descrita es la siguiente:

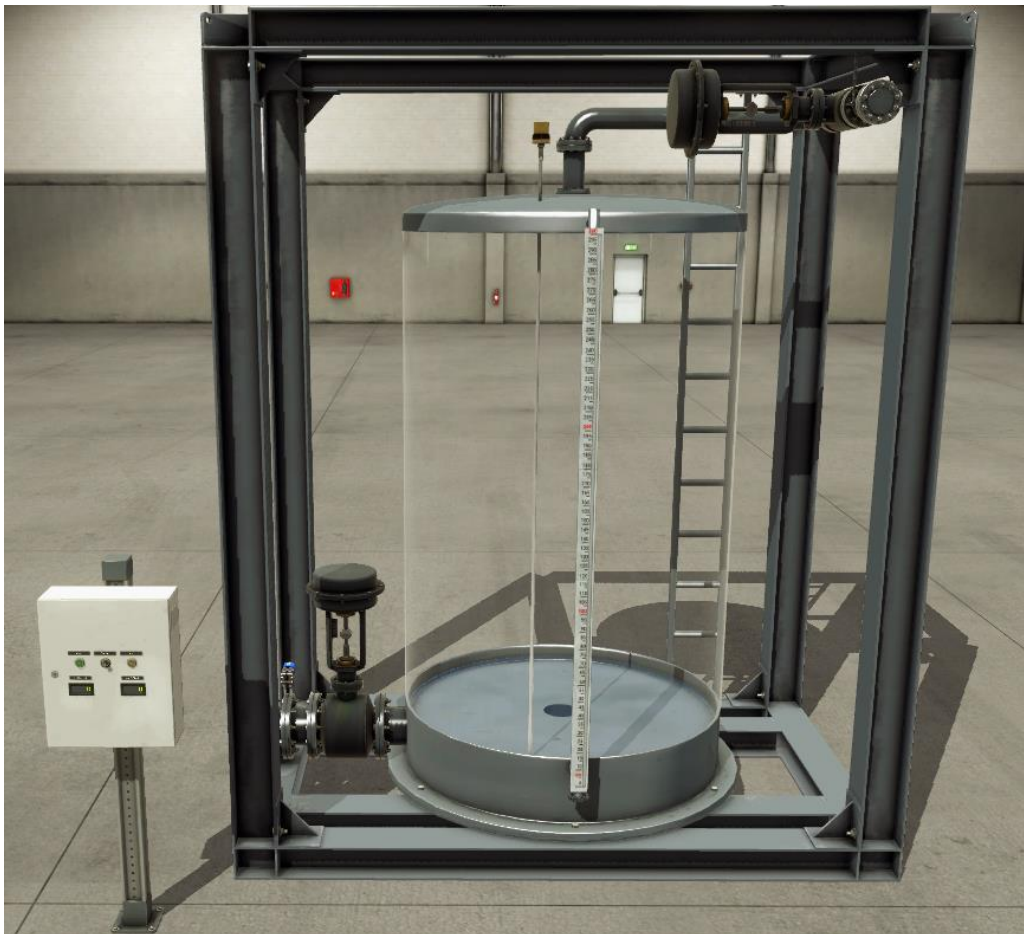


Figura 41. Vista general. Escena depósito de líquido

Para variar el nivel del tanque bastan con variar el valor del regulador hasta que en el display de “Litros solicitados” se muestre el valor deseado y pulsar el botón de vaciar o llenar correspondiente en función del nivel actual del tanque (mostrado en el display de “Litros en depósito”).

Cabe remarcar que tanto el potenciómetro, como el sensor de nivel del tanque y la apertura de las válvulas trabajan con valores de tensión entre 0 y 10V, por lo que es necesario escalar estas variables analógicas para emplearlas en el programa.

El programa del autómatas está dividido en diferentes bloques de programa OB, FC, FB y DB, conteniendo y estructurando la programación del PLC. La distribución de bloques del programa del autómatas se muestra a continuación:

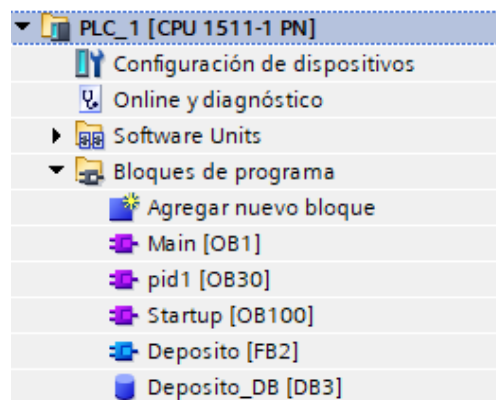


Figura 42. Distribución de bloques. Estación depósito de líquido

Como se comentó al final del [apartado 4.3](#), ciertos comandos adquieren nuevas restricciones con el cambio de versión del software, como es el caso del comando PID\_Compact, empleado en esta estación. La versión TIA Portal V17 no permite la simulación de este comando con PLCSIM [\[13\]](#), lo que obliga a simularlo con PLCSIM Advanced. Como este software de simulación del PLC está limitado a la gama S7-1500, es necesario cambiar el controlador en el programa de TIA portal. Al ser un PLC de otra gama, no permite cambiar el controlador en el mismo proyecto y manteniendo los bloques de programa, así que se ha tenido que crear un nuevo proyecto con un PLC S7-1500 y copiar y pegar entre ambos proyectos los bloques de programa y las variables.

Una vez realizado este proceso, la actualización de la práctica estaría completa, sin ser necesaria la FC para la conexión entre PLCSIM y Factory IO puesto que al emplear un S7-1500 se simulará con PLCSIM Advanced.

### 4.6.3 Corrección de errores

Revisando la programación del autómatas, se detectó que el bloque de gestión del llenado o vaciado del depósito, así como la configuración del PID\_Compact no cumplían con su finalidad, provocando que la estación no funcionara. Se cree que estos errores pueden venir de que el programa del que se parte como original no fuera el programa final del autómatas.

En el caso de la FB encargada de la gestión del tanque, el bloque de programa original no cumplía con su misión. Se tomó la decisión de programar uno nuevo con el objetivo de optimizar el tiempo, ya que era más rápido comenzar uno desde cero que corregir el original, pero manteniendo la descripción del funcionamiento de la estación descrito por el autor original. La FB creada es la siguiente:

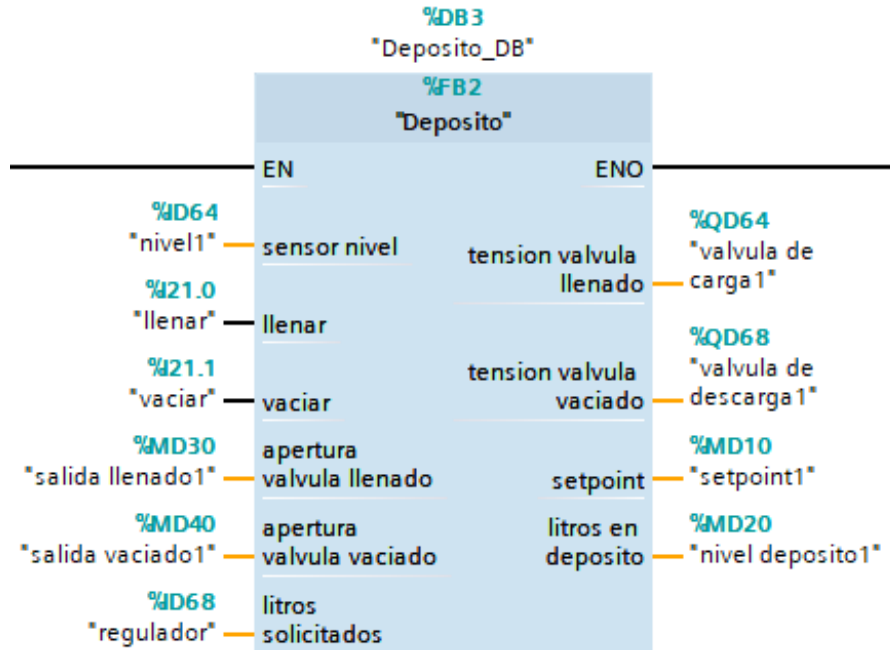


Figura 43. Bloque "Deposito" creado. Estación depósito de líquido

Por otra parte, el comando PID\_Compact permite realizar la configuración del PID (necesaria para su correcto funcionamiento) seleccionando la ventana remarcada en color rojo en la siguiente figura.

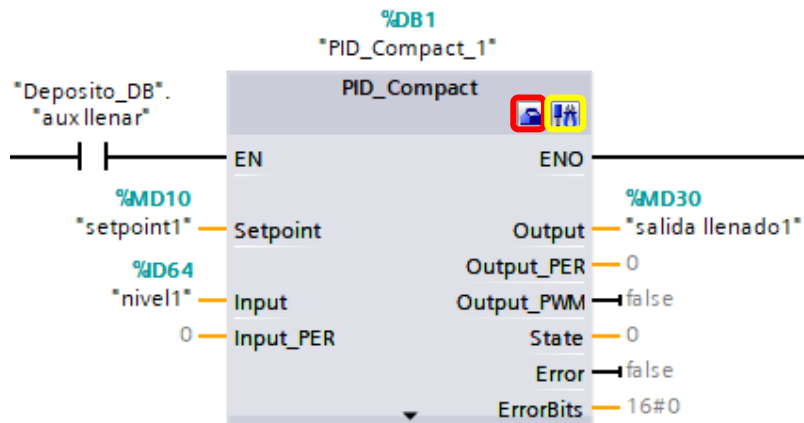


Figura 44. Comando PID\_Compact. Estación depósito de líquido

Además de los campos modificados por el autor original de esta estación, es necesario que el funcionamiento de ambos PID's se encuentre en automático para se activen y que el PID que controla el vaciado del tanque tenga el sentido de regulación invertido.

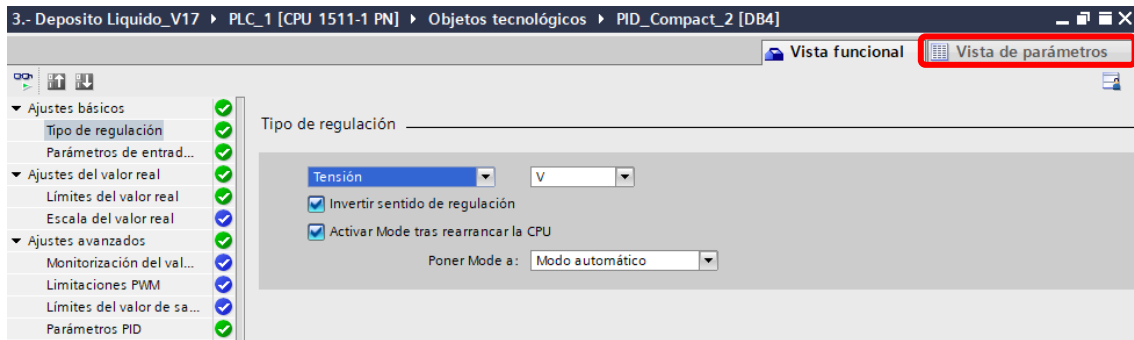


Figura 45. Configuración PID\_Compact de vaciado. Escena depósito de líquido

En adicción, se ha debido repetir el proceso de puesta en servicio que se describe a continuación para la determinación de los valores de los parámetros del PID ( $K_p$ ,  $T_i$  y  $T_d$ ), puesto que los determinados en el programa original provocaban un error estacionario excesivo.

Para realizar la sintonía de los parámetros del PID, una vez se simula la estación completa, es necesario abrir el cuadro de puesta en servicio (remarcado en amarillo en la figura 44). La ventana que se despliega es la que se muestra a continuación.

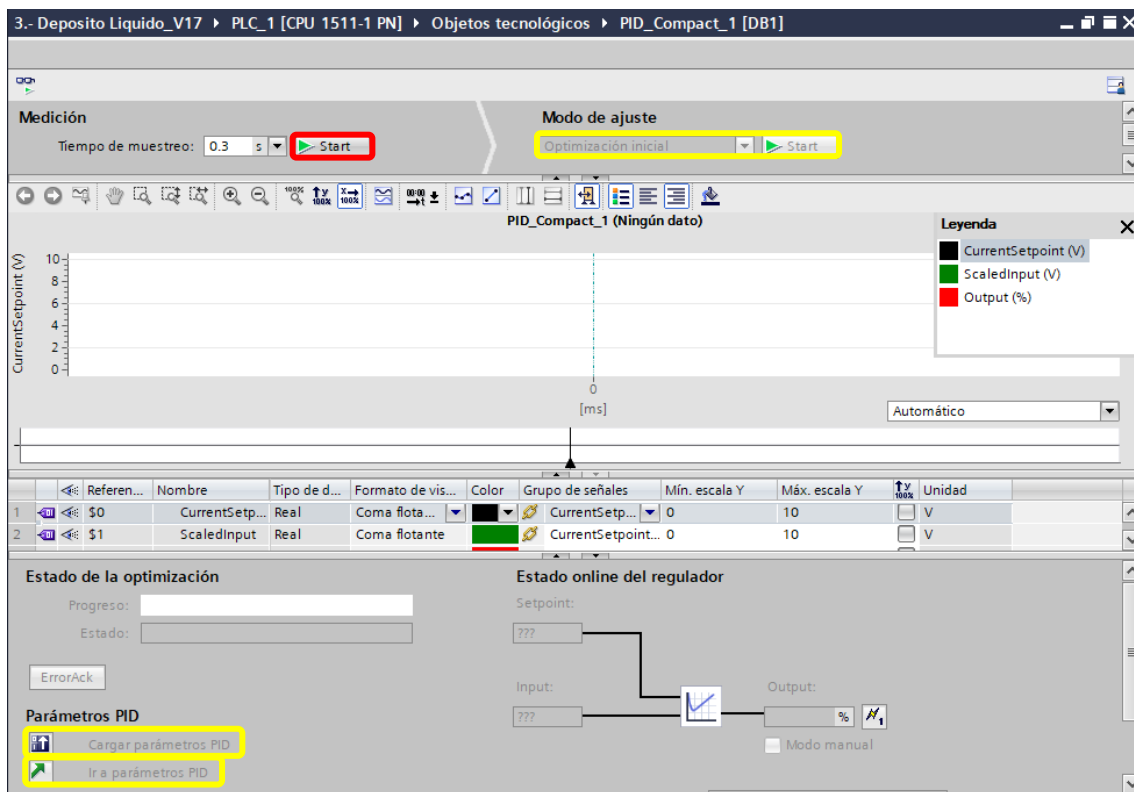


Figura 46. Ventana de puesta en servicio del PID. Escena depósito de líquido

En esta ventana se selecciona “Start” para comenzar la medición y se realizan varios saltos en el setpoint. En estos saltos se selecciona en el modo de ajuste, la opción de “Optimización inicial” y posteriormente la de “Optimización fina”, cargando los parámetros en el PID después de cada una de ellas. Para el último ajuste, se emplea el método de prueba y error descrito en el [apartado 2.4.5](#) variando de forma manual



los valores, en el apartado “Ir a parámetros PID”, hasta obtener una respuesta correcta del sistema.

Después de numerosos intentos para sintonizar ambos PID's, el ajuste obtenido es una respuesta sin sobrepico pero con error estacionario no nulo en los dos casos (aproximadamente un 8% sobre el nivel del tanque en caso del llenado y un 3% en caso del vaciado). Cabe destacar que el diseño de la estación por el autor original limitaba el uso de las válvulas de llenado y vaciado solo bajo la orden de llenado o vaciado respectivamente. Esto quiere decir que, si en la opción de llenado el nivel del tanque sobrepasaba el setpoint, no se podía abrir la válvula de salida para regularlo, y viceversa con la opción del vaciado. Un diseño del programa donde ambos PID's pudieran estar activos sin importar la operación que se estuviera demandando habría facilitado este control del nivel, pudiendo incluso, conseguir un error estacionario nulo.

## 4.7. ESTACIÓN 4: GESTIÓN DE STOCK DE UN ALMACÉN

### 4.7.1 Objetivos de aprendizaje

Es una estación con un nivel de dificultad alto. Tiene como objetivos principales el aprendizaje de la programación y simulación relacionada con el uso de un HMI y comprender y aplicar los sistemas de gestión FIFO y LIFO. Como objetivos secundarios se pueden remarcar el aprendizaje de la compartimentación del programa del autómatas, el uso de variables analógicas y el comando P\_TRIG como condición de activación.

- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** P\_TRIG.
  - **Variables:** Importar/exportar variables, conectarlas las variables del PLC con las del HMI y su representación gráfica en el HMI.
  - **Programa:** Aplicación de control FIFO y LIFO, aprender a programar un HMI, compartimentalización del programa en FC, FB, DB y OB.
  - **Conectividad:** Tipo de PLC, tipo de HMI, direccionamiento IP y conectividad con PLCSIM, WinCC y Factory IO.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Estación de la grúa apiladora y estantería.
  - **Conectividad:** Conectar con PLCSIM y WinCC.

### 4.7.2 Descripción de la estación

Se trata de una estación conformada por una grúa apiladora y una estantería que pretenden gestionar de forma automática el sistema de almacenamiento y salida de productos de una fábrica ofreciendo dos tipos de sistemas de gestión, LIFO o FIFO. El sistema de control está gestionado por la combinación de un PLC y un HMI.

Mediante una cinta de rodillo con sensores capacitivos, entran de forma aleatoria pallets con una pieza encima que puede ser de seis tipos diferentes. Al final de los rodillos, se encuentra la cinta de carga y descarga de la grúa apiladora y un sensor de visión que determina el tipo de pieza en función de su forma y color. Una vez determinado el tipo de pieza, la cinta de carga y descarga coloca el pallet en la grúa, que se encarga de posicionarlo en la fila correspondiente de la estantería según el tipo de pieza que porte.

En el caso de solicitar la extracción de piezas, es la misma grúa la encargada de retirar los pallets indicados de la estantería y posicionarlos en la cinta de carga y descarga. Desde esta, el pallet es transferido a otra cinta de rodillos con sensores capacitivos para su salida de la instalación. El método de selección de prioridad para la extracción de piezas está determinado por el tipo de gestión de stock que queramos implementar. Para esta gestión se dispone de dos opciones:

- **Sistema FIFO (First In-First out):** La primera pieza que ha entrado es la primera en salir.
- **Sistema LIFO (Last IN- First Out):** La última pieza que ha entrado es la primera en salir.

La distribución de la estación, explicada de forma escrita anteriormente, se muestra a continuación:



Figura 47. Vista general. Escena gestión de stock de un almacén

Para gobernar el comportamiento de la estación no se emplea un cuadro eléctrico dotado de botones y displays como en los casos anteriores, en vez de esto se emplea un HMI.

Las variables del panel de operador pueden ser internas o compartidas con el PLC, dependiendo de su función y de la programación realizada en el HMI y el PLC. Por lo tanto, se debe establecer una conexión entre ambos dispositivos de control. Esta conexión se realiza mediante conexión Ethernet.

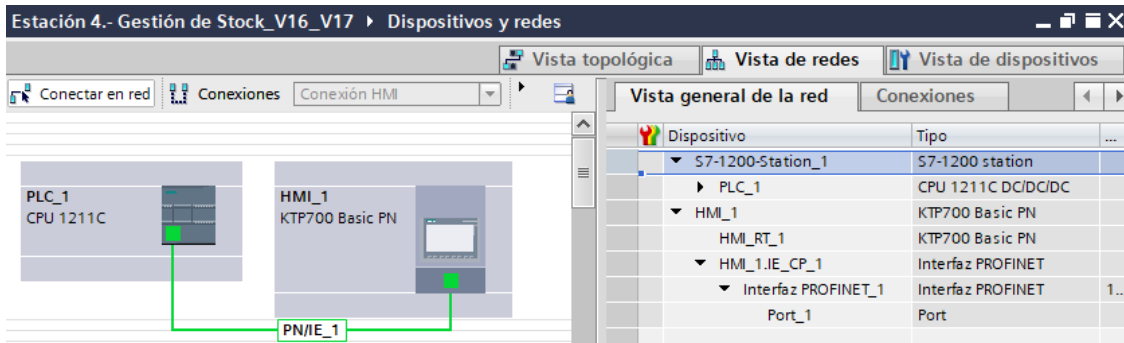


Figura 48. Vista dispositivos y redes. Estación gestión de stock de un almacén

El panel de operador posee diferentes pantallas desde donde se puede controlar la gestión de la marcha, la selección del tipo de gestión, el seguimiento del número de piezas de cada tipo a lo largo de la simulación y la solicitud de extracción de estas.

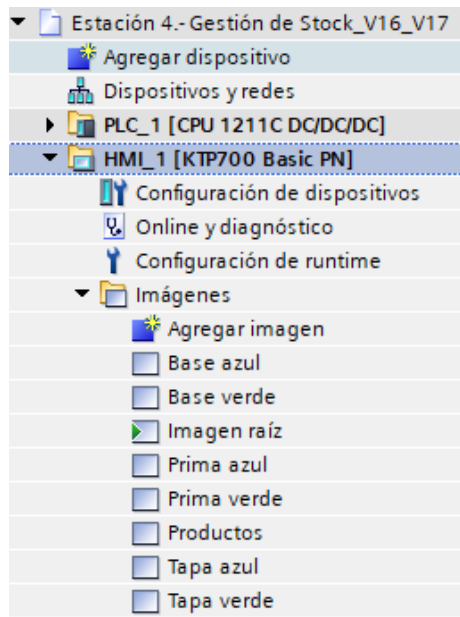


Figura 49. Distribución de imágenes. Estación gestión de stock de un almacén

La imagen raíz o imagen de partida, contiene un botón de marcha, uno de parada, un selector de tipo de gestión, un indicador de modo de trabajo y un pulsador para acceder a la pantalla de productos. De forma análoga a las estaciones anteriores, si se pulsa el botón de marcha el proceso automático comienza y si se pulsa el botón de paro la estación completa se detiene. Para reanudar la marcha, basta con pulsar el botón de marcha de nuevo. El modo en que se encuentre la estación (marcha o parada) se refleja mediante un objeto luminoso en la pantalla.

En la pantalla de productos se muestran los seis tipos de piezas, pudiendo seleccionar el tipo de pieza que se desee para visualizar una gráfica con la cantidad de piezas tratadas en la instalación a lo largo de la simulación y solicitar la extracción de cualquier número de ellas.

Tanto la pantalla de productos como las pantallas de cada tipo de pieza están dotadas de un botón que les permite volver a la pantalla anterior.

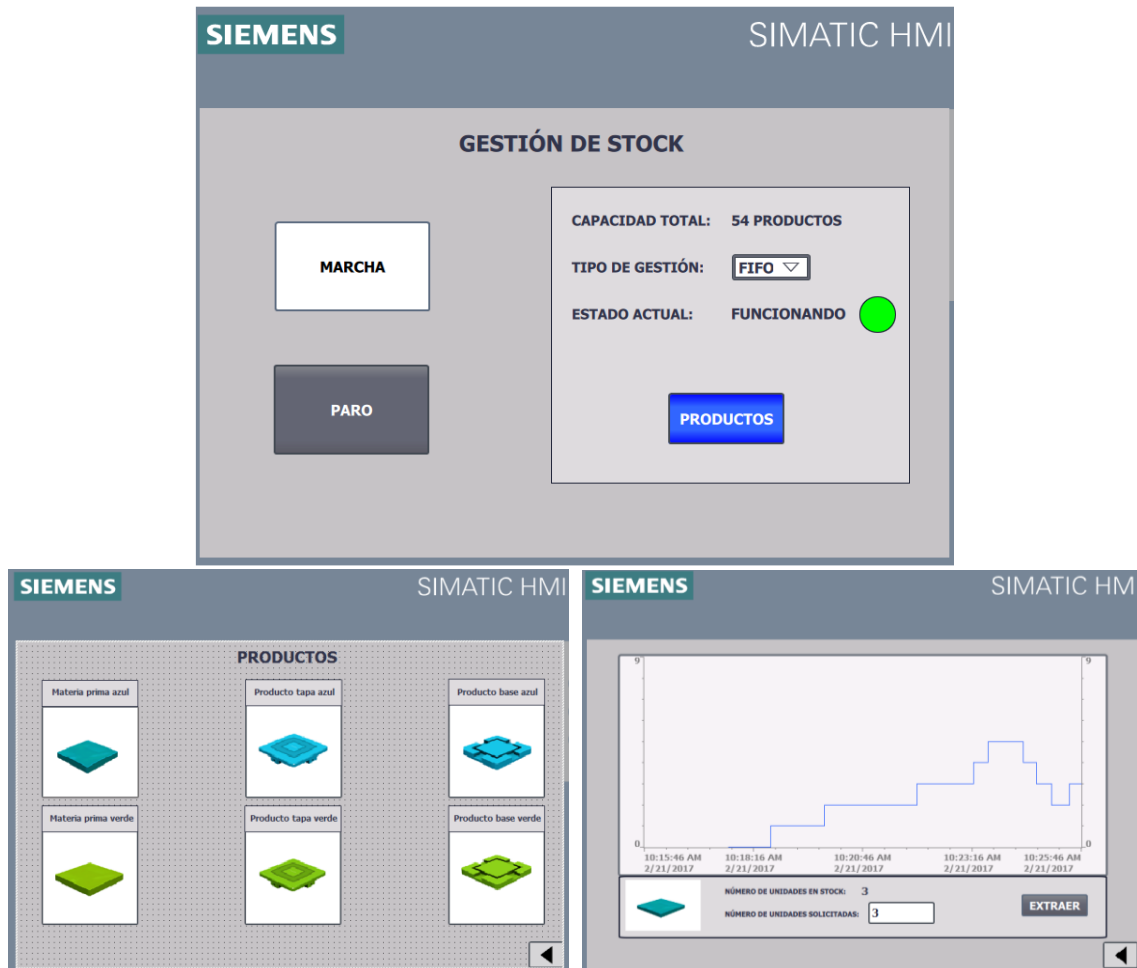


Figura 50. Pantallas del HMI. Estación gestión de stock de un almacén

Como se comentó al final del [apartado 4.3](#), el uso de un HMI puede requerir de un proceso especial de actualización, como es el caso de esta práctica. Cuando se elige un HMI para el proyecto, no solo elegimos el producto, sino que este viene con una versión asociada a él. Por lo tanto, como el proyecto de TIA Portal anterior estaba en la versión 13, el HMI también tenía esta versión. Para actualizar la versión del HMI basta con abrir el árbol de proyecto y hacer click derecho en el HMI, seleccionar la opción "Cambiar dispositivo 7 versión" y desplegará una venta como la que se muestra en la figura siguiente.

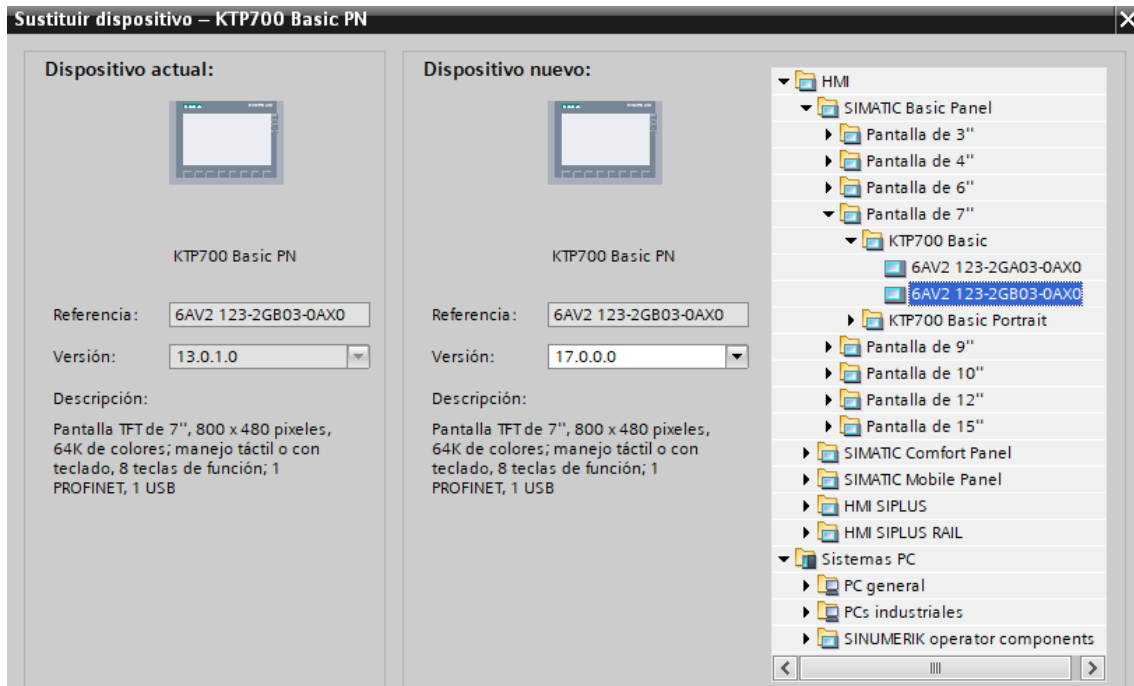


Figura 51. Pantallas del HMI. Estación gestión de stock de un almacén

En esta ventana se selecciona el mismo dispositivo que se tenía, se le asigna la versión V17 y se pulsa el botón de “Aceptar”. Con esto ya se tendría el HMI listo para simularse sin haber perdido las imágenes o las variables realizadas por el autor.

Por último, se muestra la distribución de bloques del programa del autómatas dividido en diferentes bloques de programa OB, FC, FB y DB.

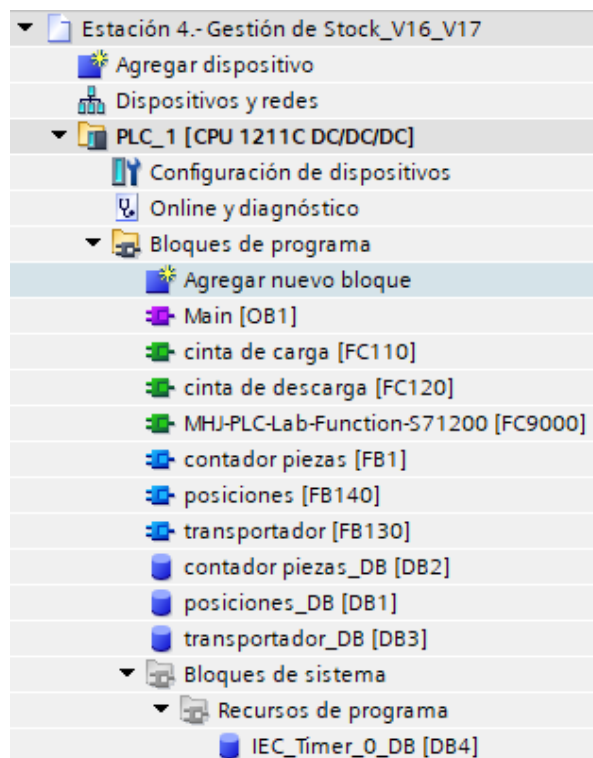


Figura 52. Distribución de bloques. Estación gestión de stock de un almacén



## 5. ESTACIONES DE SIMULACIÓN DISEÑADAS

Para la programación y simulación de las estaciones se emplean dos modelos diferentes de autómatas, ambos de la marca SIEMENS. El motivo de emplear dos gamas distintas de PLC es la adquisición de más competencias de simulación. El primero, es el S7-1200, que permite su simulación con la herramienta PLCSIM y conectarse con entornos virtuales diseñados en Factory IO. El segundo, es el S7-1500, que permite su simulación con PLCSIM y PLCSIM Advanced pudiéndose conectar con entornos virtuales diseñados tanto en Factory IO como en SIMIT.

En el caso de emplear un HMI en la simulación, se ha seleccionado el modelo de SIEMENS “SIMIT TP700 comfort”.

### 5.1. ESTACIÓN 5: CINTA TRANSPORTADORA

#### 5.1.1 Objetivo de la simulación

Es una estación muy simple con una programación básica, con esto se pretende inicializar y familiarizar al usuario con el programa de TIA Portal V17, así como con los principales comandos y variables para la programación de un PLC con este software. En lo referente a entornos de simulación, tanto en Factory IO como en SIMIT, se tiene como finalidad conocer los principales elementos y librerías dónde se encuentran, aprender a desplazarse en el entorno y cómo conectar el entorno virtual con la simulación del PLC.

- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** Set, Reset, SR, CTU, CTUD, MOVE, flancos, comparadores y operadores matemáticos.
  - **Variables:** Input, output, memoria e importar/exportar variables.
  - **Programa:** Cortar y pegar bloques de programa.
  - **Conectividad:** Tipo de PLC, dirección IP y conectividad con PLCSIM, PLCSIM Advanced, Factory IO y SIMIT.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Elementos básicos, librerías y colocación del layout.
  - **Conectividad:** Con PLCSIM y PLCSIM Advanced.
- **Conocimientos que adquirir en SIMIT:**
  - **Componentes:** Elementos básicos, librerías y colocación del layout.
  - **Conectividad:** Con PLCSIM Advanced.



### 5.1.2 Descripción de la escena

La estación tiene como finalidad la automatización de un sistema de transporte y conteo de cajas en una línea de cintas transportadoras.

Está compuesta por dos cintas transportadoras en serie de 2 y 6 metros (cinta 1 y cinta 2 respectivamente) con un sensor difuso al comienzo de la primera cinta y un sensor difuso en cada extremo de la segunda cinta.

Las cajas entran por la cinta 1 a la altura del primer sensor (*“sensor pieza”*), encargado de detectar la entrada de cajas y aumentar en uno el contador de cajas en la cinta 1. Al comienzo de la cinta 2, se encuentra el *“sensor inicio cinta 2”* encargado de detectar la salida de cajas de la cinta 1, disminuir una unidad el *“contador de cajas en la cinta 1”* y detectar que la caja ha entrado en la cinta 2 aumentando su contador una unidad. El sensor ubicado al final de la cinta 2, se ocupa de disminuir el *“contador de cajas en la cinta 2”* además de aumentar un *“contador de cajas totales transportadas”* por la estación completa.

En estado de marcha, si el *“contador de cajas en la cinta 1”* es mayor o igual a uno y el *“sensor final cinta 1”* no detecta caja, el motor de la cinta se activa. En el caso de que el *“contador de la cinta 2”* sea mayor o igual a uno, el motor de la cinta 2 se activa. De esta forma, se pretende reducir al máximo el gasto eléctrico de la estación, al no estar el motor de las cintas en funcionamiento de forma continua.

Con el fin de controlar la estación, se dispone de un cuadro eléctrico que contiene una seta de emergencia, un pulsador de marcha con iluminación, un pulsador de paro con iluminación, tres displays y tres pulsadores de reset.

Si se pulsa el botón de marcha, la estación entra en estado de marcha, esperando a que llegue la primera caja y funcionando de forma totalmente automática, indicándose con una luz verde en el propio pulsador de marcha. Si se pulsa el botón de paro, independientemente de la señal de los sensores, la estación entra en estado de reposo, indicándose con una luz roja en el propio botón de paro. Para reanudar el estado de marcha basta con volver a pulsar el botón de marcha.

En caso de emergencia, si se pulsa la seta de emergencia la estación se detiene. Al ser una seta de emergencia, será necesario girar la seta para quitar el enclavamiento y pulsar el botón de marcha. Si no se realiza esta operación en ese orden, no se reanudará la marcha para evitar que se produzca una situación potencialmente peligrosa para el operario o para la instalación.

Las cajas en la cinta 1, en la cinta 2 y el total de cajas transportadas por la estación se muestran en los tres displays del cuadro eléctrico. Cada botón de reset, reinicia el contador correspondiente.

Cabe destacar que, en el caso de reiniciar los contadores del número de cajas en las cintas, puede que la estación entre en un estado de inactividad permanente si no retiramos las cajas sobre estas cintas, puesto que la activación de los motores de las cintas transportadoras está condicionada al contador de cajas sobre ellas.

La distribución de todos los componentes descritos conforma la escena que se emplea en la simulación. Dicha colocación permite cumplir el objetivo buscado, obteniendo como resultado la estación que se muestra a continuación:



Figura 53. Vista general. Escena cinta transportadora



Figura 54. Vista cuadro eléctrico. Escena cinta transportadora



### GRAF CET cinta 1

Este diagrama de bloques refleja los diferentes estados de la cinta 1 en función de los sensores colocados en esta cinta transportadora y los botones del cuadro eléctrico. Al ser un GRAFCET no detallado, no se reflejan las variables que permiten las transiciones entre etapas, la situación de parada o el proceso asociado al contador de la cinta.

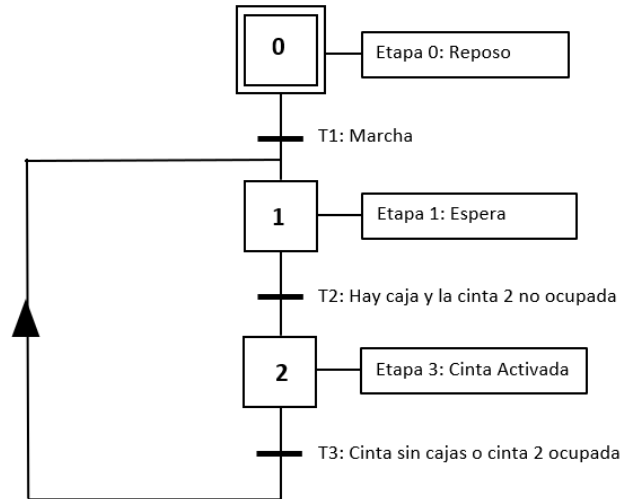


Figura 56. GRAFCET cinta 1. Estación cinta transportadora

### GRAF CET cinta 2

Este diagrama de bloques refleja los diferentes estados de la cinta 2 en función de los sensores colocados en esta cinta transportadora y los botones del cuadro eléctrico. Al ser un GRAFCET no detallado, no se reflejan las variables que permiten las transiciones entre etapas, la situación de parada o los procesos asociados al contador de la cinta o al general de cajas transportadas.

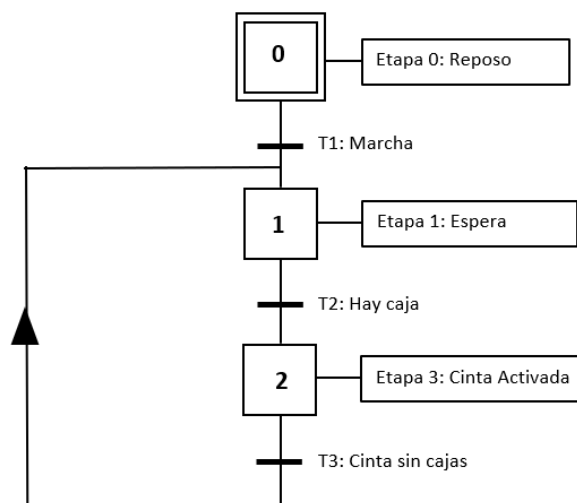


Figura 57. GRAFCET cinta 2. Estación cinta transportadora

Mediante la combinación de estos GRAFCET's, y desarrollando de forma completa cada proceso, es posible estructurar el programa del PLC.

### 5.1.4 Programación del autómatas

La programación del autómatas se realiza en lenguaje KOP ajustándose a los GRAFCET's descritos anteriormente. Al ser una estación con una programación relativamente sencilla no sería lógico estructurarlo en bloques de programa FC, FB o DB puesto que consumiría más recursos del PLC y la estructura del programa es suficientemente extensa y clara como para realizarse en el bloque de main.

Las variables empleadas para la programación del autómatas contienen las variables descritas en Factory IO, así como variables internas del PLC que son necesarias para la programación, siendo el conjunto de ellas las reflejadas a continuación:

Variables PLC					
		Nombre	Tabla de variables	Tipo de datos	Dirección ▲
1		Marcha	Tabla de variables e..	Bool	%I10.0
2		Paro	Tabla de variables e..	Bool	%I10.1
3		Emergencia	Tabla de variables e..	Bool	%I10.2
4		Sensor_Pieza	Tabla de variables e..	Bool	%I10.3
5		Sensor_Inicio_Cinta2	Tabla de variables e..	Bool	%I10.4
6		Sensor_Final_Cinta2	Tabla de variables e..	Bool	%I10.5
7		Reset_Cinta1	Tabla de variables e..	Bool	%I10.6
8		Reset_Cinta2	Tabla de variables e..	Bool	%I10.7
9		Reset_Total	Tabla de variables e..	Bool	%I11.0
10		Motor_Cinta1	Tabla de variables e..	Bool	%Q0.0
11		Motor_Cinta2	Tabla de variables e..	Bool	%Q0.1
12		Luz_Marcha	Tabla de variables e..	Bool	%Q0.2
13		Luz_Paro	Tabla de variables e..	Bool	%Q0.3
14		Display_Cinta1	Tabla de variables e..	Int	%QW30
15		Display_Cinta2	Tabla de variables e..	Int	%QW32
16		Display_Total	Tabla de variables e..	Int	%QW34
17		Eta1a1_Condiciones_OK	Tabla de variables e..	Bool	%M0.0
18		Flanco1	Tabla de variables e..	Bool	%M0.1
19		Flanco2	Tabla de variables e..	Bool	%M0.2
20		Eta1a2_Cinta1	Tabla de variables e..	Bool	%M0.3
21		Eta1a2_Cinta2	Tabla de variables e..	Bool	%M0.4

Tabla 4. Variables del PLC. Estación cinta transportadora

Como ya se ha mencionado anteriormente, todas las **variables de entrada y salida** son las empleadas en el programa de simulación del entorno virtual de Factory IO. Cabe remarcar la necesidad de asociar la misma dirección de variable en ambos programas, debido a que los dos simulan un único PLC con un conjunto de entradas y salidas determinadas.

Uno de los objetivos de esta estación es que el usuario conozca lo que son las variables de memoria y como usarlas. Para la programación se emplean cinco **variables de memoria** del PLC:

- **“Etapa1 Condiciones OK”**: Será necesaria para establecer el estado de marcha una vez que se pulse el botón de marcha, sin la necesidad de mantenerlo pulsado.
- **“Etapa2 Cinta1”** y **“Etapa2 Cinta2”**: Son variables intermedias para la activación de los motores de ambas cintas.
- **“Flanco1”** y **“Flanco2”**: Para registrar los flancos de subida y bajada de las variables asociadas a los sensores difusos. Es necesario almacenar en memoria el valor de estas variables para compararlas con los valores que tenían en el ciclo de scan anterior.

La estructura del programa es la siguiente:

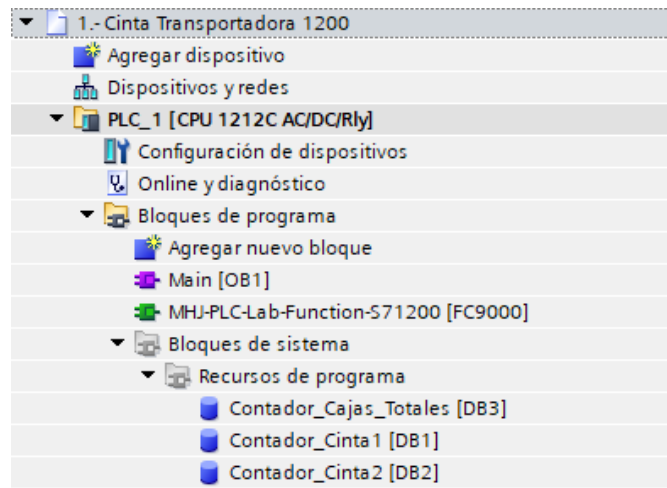


Figura 58. Distribución de bloques. Estación cinta transportadora

Se emplea el bloque FC9000 para la conexión entre Factory IO y PLCSIM. Este bloque se llama desde el main del programa, que también contiene la programación de ambas cintas con los contadores asociados, estructurada en segmentos de la siguiente manera:

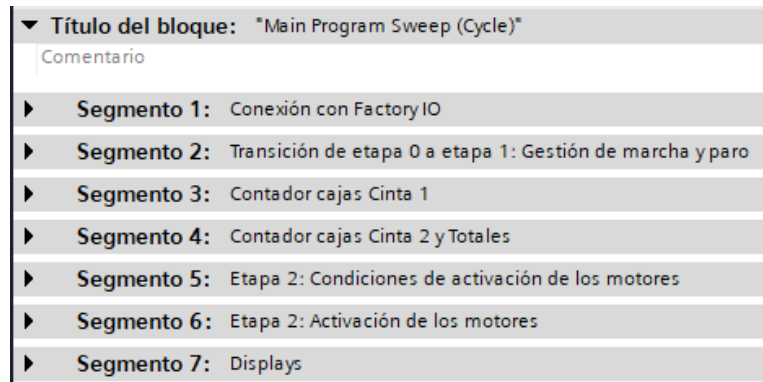


Figura 59. Distribución de segmentos Main. Estación cinta transportadora

El bloque main, consta de siete segmentos que permiten acceder fácilmente a la sección del programa que se desee de forma simple y visual. Con el objetivo de analizar la estructura del programa y relacionar los objetivos de la práctica con lo empleado, se muestran a continuación los segmentos necesarios.

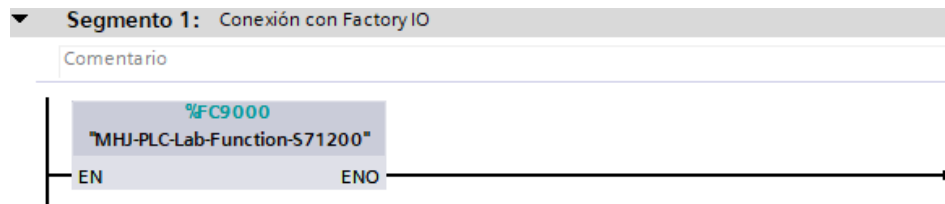


Figura 60. Segmento 1 Main. Estación cinta transportadora

La finalidad del primer segmento es llamar a la función que permite conectar las variables del autómatas virtual (simulado con PLCSIM) con Factory IO.

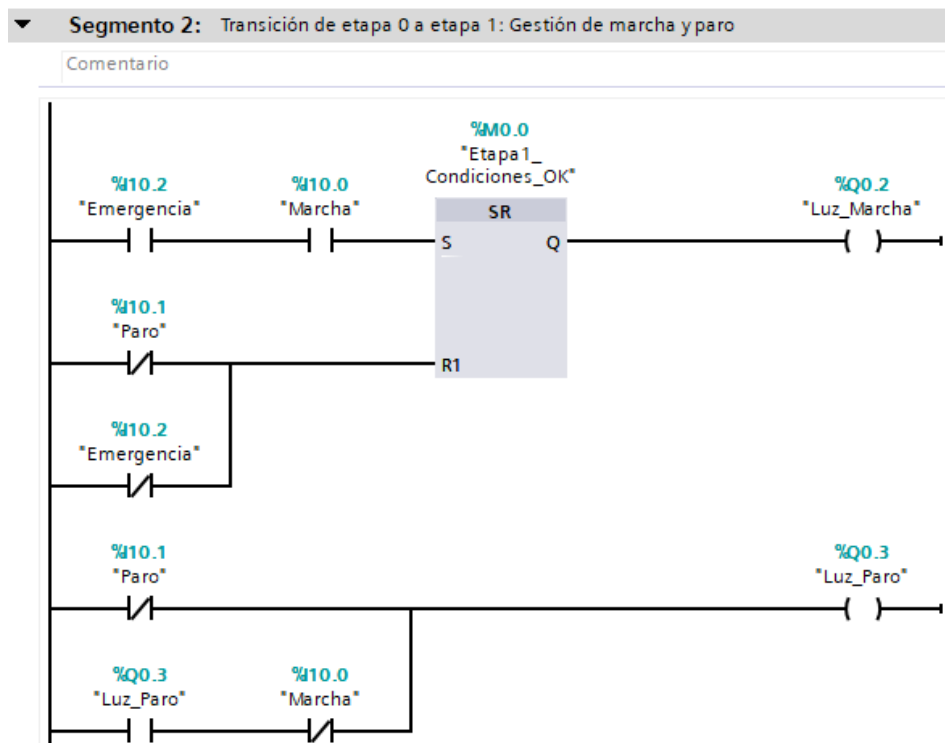


Figura 61. Segmento 2 Main. Estación cinta transportadora

El segundo segmento controla la marcha y paro de la estación mediante los botones del cuadro eléctrico, indicando con dos salidas luminosas el estado de la estación.

En esta figura se puede ver el uso del comando **SR**, para fijar el estado de una variable con un solo pulso (acciones de **set** y **reset** de una variable).

Mediante la gestión de marcha-paro de la estación, se consigue introducir la mayor parte de los elementos básicos de programación ladder.

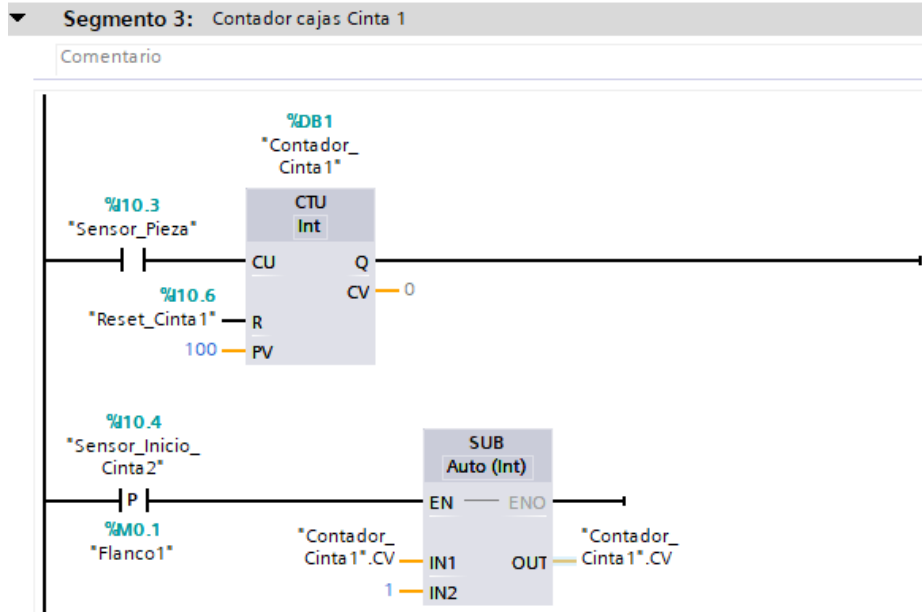


Figura 62. Segmento 3 Main. Estación cinta transportadora

El tercer segmento contiene la programación del contador de la cinta 1, mediante el uso de un contador ascendente (CTU), un flanco y operaciones matemáticas.

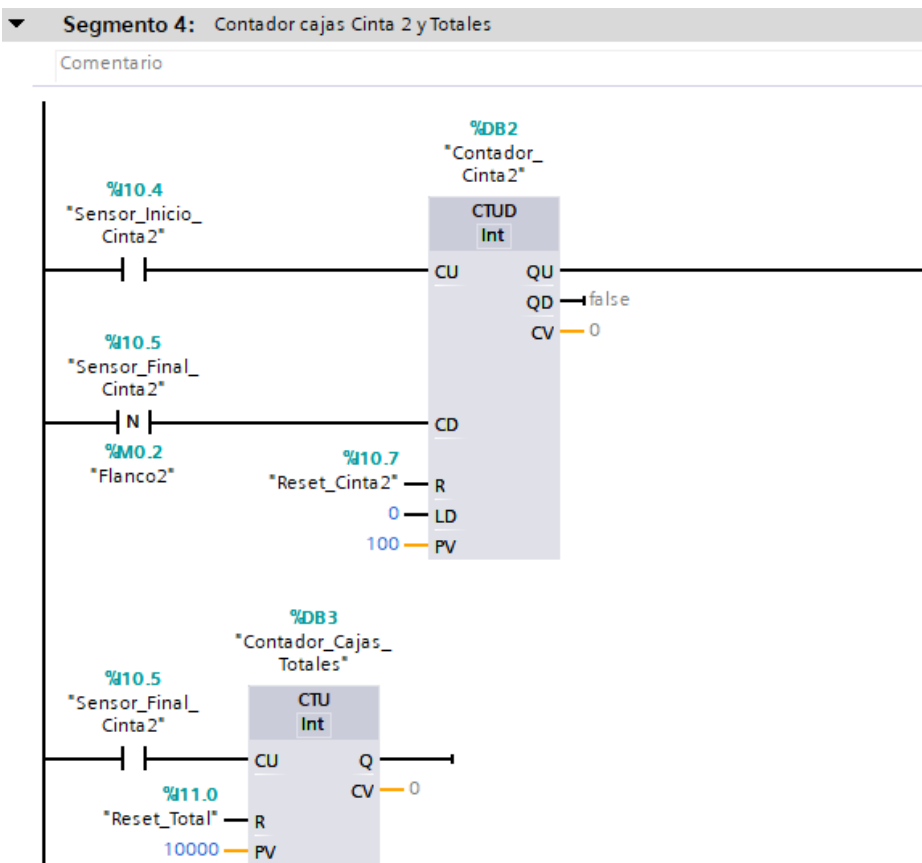


Figura 63. Segmento 4 Main. Estación cinta transportadora

De forma análoga, el segmento número cuatro, contiene la programación de los contadores de la cinta 2 y de las cajas totales transportadas por la estación. Aunque

se podría haber programado de igual forma que el segmento anterior, con la finalidad didáctica de introducir el concepto de contador ascendente-descendente, se ha empleado el **comando CTUD**.

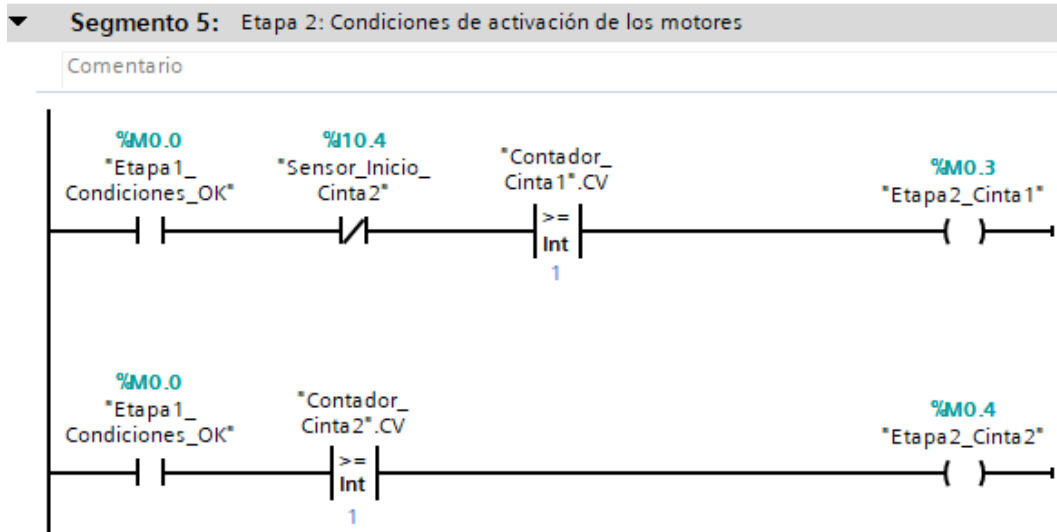


Figura 64. Segmento 5 Main. Estación cinta transportadora

El segmento cinco controla la activación de los motores de las cintas transportadoras mediante el uso de **comandos comparadores y variables de tipo marca**.

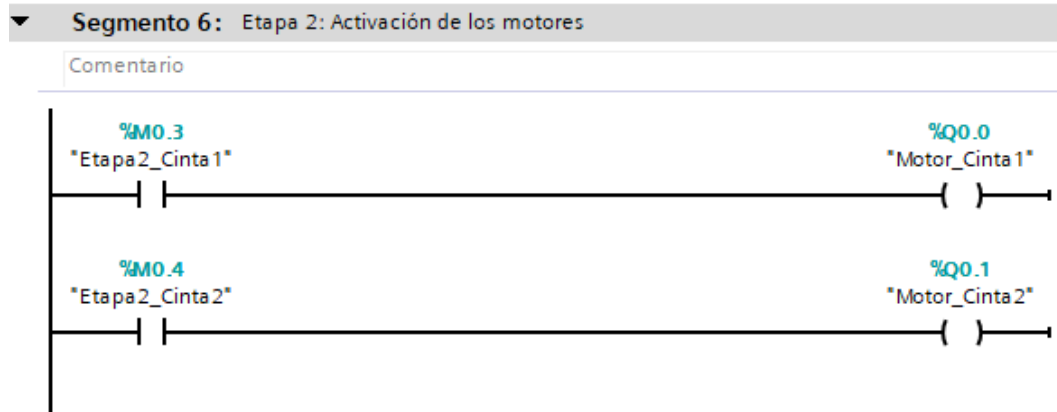


Figura 65. Segmento 6 Main. Estación cinta transportadora

El segmento seis activa los motores de las cintas transportadoras empleando las variables del segmento cinco. Es sencillo apreciar que no sería necesario, puesto que se podía haber activado las variables de salida directamente en el segmento 5. Sin embargo, para una mejor estructuración del programa y siguiendo los diagramas GRAFCET antes mencionados, se ha decidido emplear este segmento adicional.

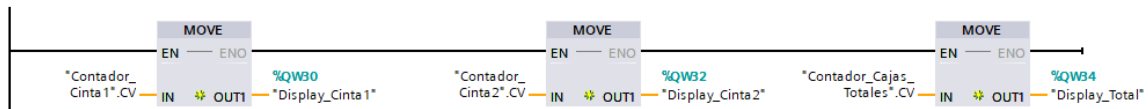


Figura 66. Segmento 7 Main. Estación cinta transportadora

Por último, el séptimo segmento se encarga de copiar los valores de los contadores en las variables de salida de los displays mediante el **comando MOVE** para que se muestren en el cuadro eléctrico.

La representación del PLC con la herramienta PLCSIM es del siguiente modo:

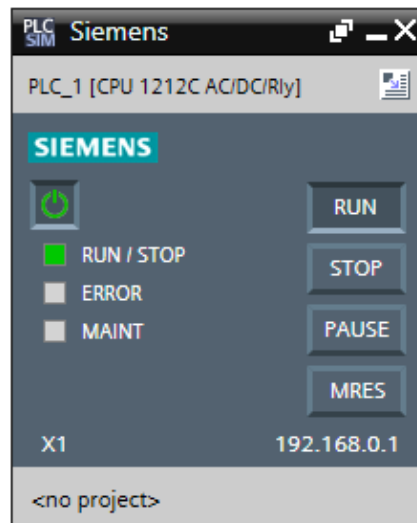


Figura 67. Simulador del PLC 1200. Estación cinta transportadora

Se puede concluir, que para la realización del programa ladder, es necesario conocer y aplicar correctamente los conocimientos marcados en los objetivos que adquirir al comienzo de la descripción de la estación. Por lo tanto, se podría justificar el cumplimiento de la finalidad didáctica de esta práctica.

### 5.1.5 Modificación con PLC S7-1500 y Factory IO

Con el fin de ampliar los conocimientos y herramientas de programación de autómatas del usuario, se pretende replicar el proceso con un PLC de la gama S7-1500. La gama S7-1500 de SIEMENS, ofrece la posibilidad de emplear el simulador PLCSIM Advanced que un autómata de la gama S7-1200 no permite.

Para ello, se crea un nuevo proyecto con el dispositivo de control indicado y se **importa** la tabla de **variables** (previamente **exportada** del anterior proyecto) y se **copia y pega el bloque main** del programa anterior. Por último, se elimina el primer segmento, puesto que para simular en PLCSIM Advanced no es necesario el bloque proporcionado por Factory IO para la correcta conexión de entradas y salidas.

Por consiguiente, la estructura del programa es la mostrada a continuación:

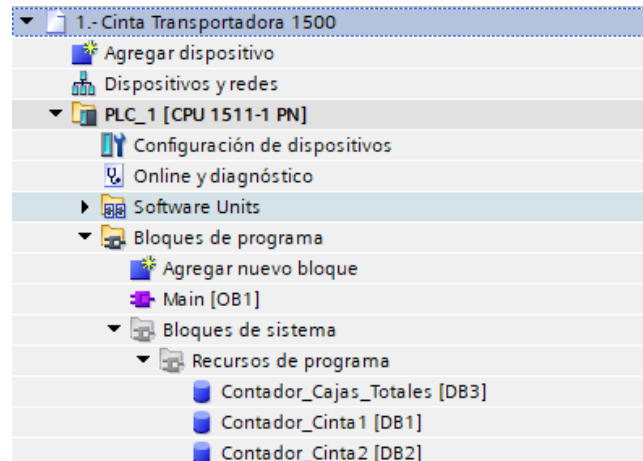


Figura 68. Distribución de Bloques. Estación cinta transportadora 1500

Al crear el PLC virtual, en el nuevo programa empleado para simular autómatas, se le debe asignar el nombre de “factoryio” puesto que el software de simulación de entornos virtuales solo busca autómatas bajo ese nombre al solicitarle la conexión. La representación del autómata con PLCSIM Advanced es de la siguiente manera:

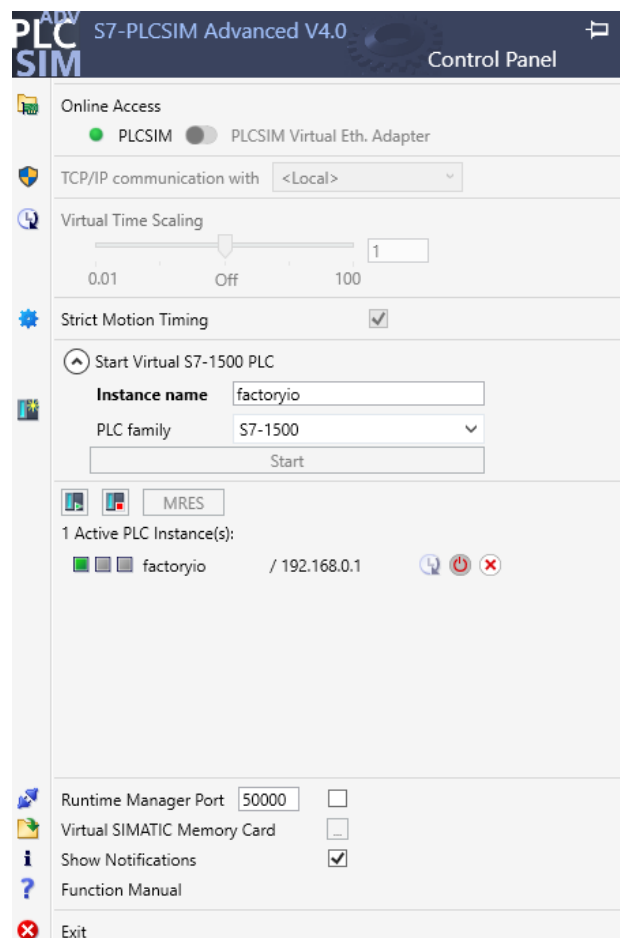


Figura 69. Simulador PLC 1500. Estación cinta transportadora 1500

En Factory IO solo se tiene que cambiar la configuración del driver antes seleccionado, debido a que tanto TIA Portal como Factory IO simulan un mismo autómatas. Por lo tanto, cambiar el controlador en TIA Portal implica tener que hacer lo consiguiente en Factory IO.



Figura 70. Configuración Factory IO. Escena cinta transportadora 1500

Con estos cambios la estación, el programa y la simulación del PLC estarían capacitados para un correcto funcionamiento.

En el caso de haber querido simular el proyecto con el PLC S7-1500 con PLCSIM en vez de con PLCSIM Advanced, se citan a continuación las diferencias que se tendrían que realizar en el proceso descrito en este apartado. Para el programa del autómatas, en vez de eliminar la FC que permite la conexión entre Factory IO y PLCSIM, se tendría que haber sustituido por la FC preparada para la gama 1500 en vez de la que habíamos usado anteriormente (preparada para la gama 1200), copiándola de la plantilla descargada de la página web de Factory IO y pegándola en el proyecto de la cinta transportadora. Por último, en la configuración del driver de Factory IO, se tendría que haber seleccionado la opción de “S7-1500” en vez de “S7-1500 (S7-PLCSIM Advanced)”.

### 5.1.6 Modificación con PLC S7-1500 y SIMIT

Como se había mencionado con anterioridad, uno de los objetivos de esta simulación es la adquisición de múltiples conocimientos empleando diversos programas de simulación. En este apartado se explica el proceso para simular la estación real con el **software de SIMIT**.

En primer lugar, se debe indicar como se realiza la conexión con el PLC simulado, que en este caso es con PLCSIM Advanced puesto que se usa el proyecto que emplea la gama S7-1500. En el árbol de proyecto de la interfaz de SIMIT, se añade un nuevo acoplamiento en el apartado de “*Coupling*” seleccionando como controlador el de PLCSIM Advanced y se selecciona el proyecto que se desea simular. De forma automática se exportan las variables de entrada y salida del proyecto como se muestra en la figura siguiente:

Default	Symbol name	Address	Data type	System	Device	Modul	Comment
<input type="checkbox"/>	Marcha	I10.0	BOOL	0	0	2	
<input type="checkbox"/>	Paro	I10.1	BOOL	0	0	2	
<input type="checkbox"/>	Emergencia	I10.2	BOOL	0	0	2	
<input type="checkbox"/>	Sensor_Pieza	I10.3	BOOL	0	0	2	
<input type="checkbox"/>	Sensor_Inicio_Cinta2	I10.4	BOOL	0	0	2	
<input type="checkbox"/>	Sensor_Final_Cinta2	I10.5	BOOL	0	0	2	
<input type="checkbox"/>	Reset_Cinta1	I10.6	BOOL	0	0	2	
<input type="checkbox"/>	Reset_Cinta2	I10.7	BOOL	0	0	2	
<input type="checkbox"/>	Reset_Total	I11.0	BOOL	0	0	2	

Symbol name	Address	Data type	System	Device	Modul	Comment
Motor_Cinta1	Q0.0	BOOL	0	0	3	
Motor_Cinta2	Q0.1	BOOL	0	0	3	
Luz_Marcha	Q0.2	BOOL	0	0	3	
Luz_Paro	Q0.3	BOOL	0	0	3	
	Q0.4	BOOL	0	0	3	
	Q0.5	BOOL	0	0	3	
	Q0.6	BOOL	0	0	3	
	Q0.7	BOOL	0	0	3	
	Q1.0	BOOL	0	0	3	

Figura 71. Variables SIMIT. Estación cinta transportadora 1500

En el apartado de “*Charts*” se dispone de dos gráficos, el primero contiene la programación de los sensores de las cintas y el segundo simula la estación industrial que se pretende representar y simular.

A diferencia de Factory IO, en SIMIT se debe de programar cuando se activan los sensores, no detectando de forma automática la interacción entre sus elementos de simulación. Es decir, en este software no hay elementos como motores, sensores y cajas que interactúen entre ellos modificando las señales de entrada, como era el caso del simulador anterior, sino que simplemente muestran elementos gráficos en la pantalla, pareciéndose más a una imagen de un HMI que a un entorno virtual de simulación de la planta.

Para simular la activación de los sensores ubicados al inicio de la cinta 1, al inicio de la cinta 2 y al final de la cinta 2, se ha empleado un conjunto de comandos “rampa” y puestas lógicas como se muestra a continuación:

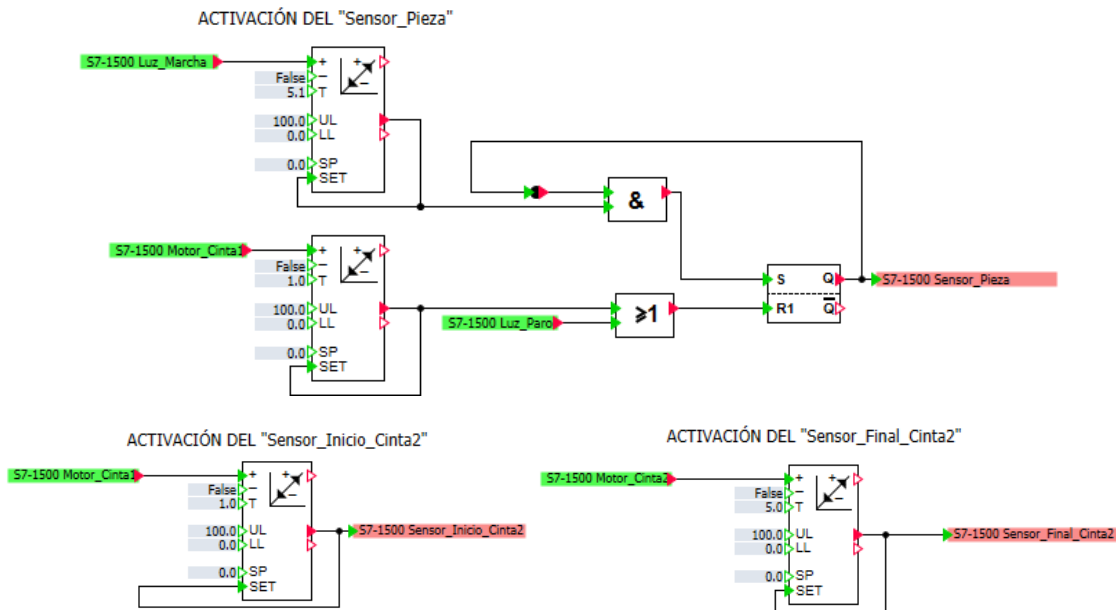


Figura 72. Chart de animaciones. Estación cinta transportadora 1500

La representación de la estación se ha realizado insertando una imagen de fondo de una cinta transportadora, creando dos grupos de rectángulos que simulan las cajas y dos paneles de visualización. Uno de los paneles simula el cuadro eléctrico del que está dotada la estación y el segundo panel se emplea únicamente para visualizar la activación de los sensores y motores de la estación, para verificar la correcta programación realizada en el gráfico de “Animaciones”.



Figura 73. Chart de cintas. Estación cinta transportadora 1500

La programación realizada en el chart de “Animaciones” no solo vale para activar o desactivar los sensores, sino que también se emplean algunas de las variables de los comandos rampa para simular el avance de las cajas en las cintas.

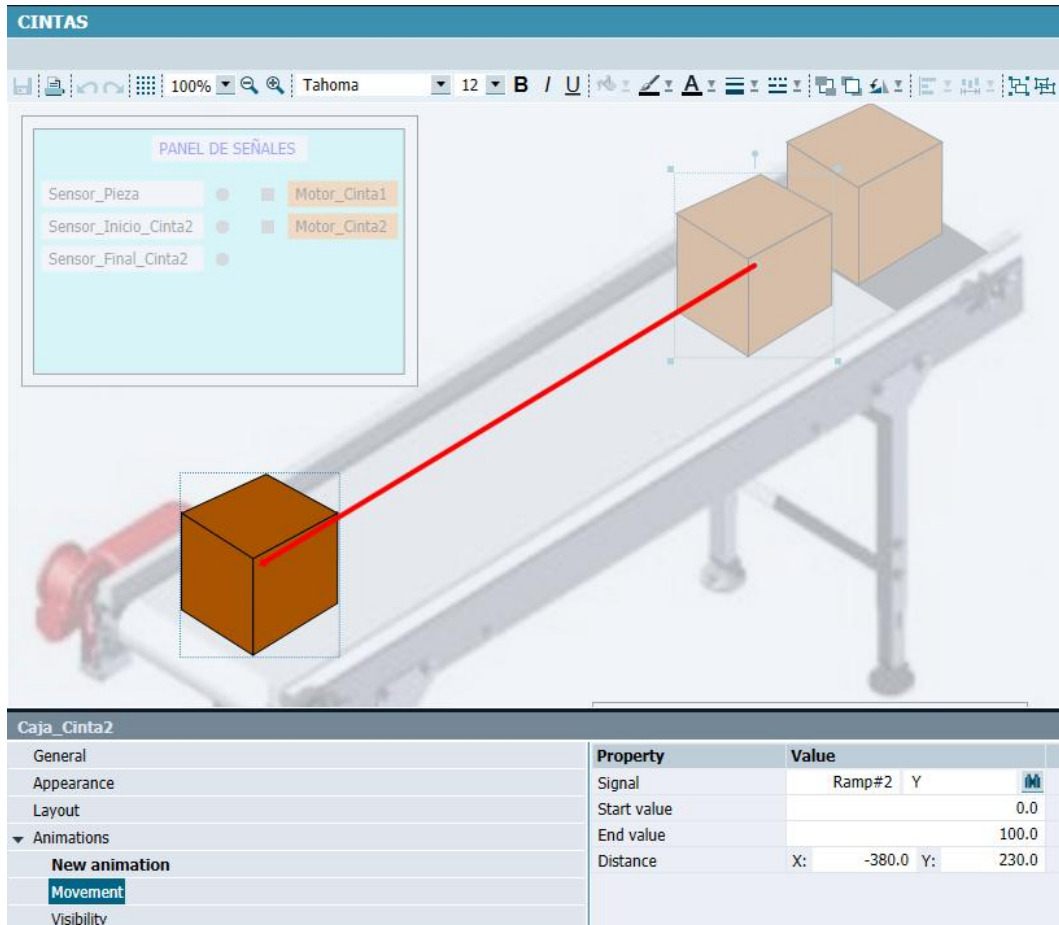


Figura 74. Animación caja chart de cintas. Estación cinta transportadora 1500

En la figura superior, se puede apreciar como el método de programación de las animaciones es muy similar al de un HMI. Proceso que no era necesario realizar con el software de Factory IO.

Gracias a la realización de esta práctica y **comparando** ambos simuladores, es claramente más intuitivo, visualmente más estético, rápido y fácil de emplear el software de **Factory IO frente al de SIMIT**. Es por estos motivos por los que en el resto de las estaciones solo se simularán en el entorno de Factory IO.

Una vez realizado todo este proceso de creación y programación del entorno virtual de simulación, solo faltaría iniciar la simulación. Para ello, bastaría con seleccionar la opción de “Start” en el árbol de proyecto de SIMIT y tener el PLC simulado con el programa cargado.

Con la realización de estas modificaciones empleando un tipo de PLC diferente y diferentes softwares de simulación, no solo se han extendido los conocimientos de herramientas de simulación, sino que también se ha tenido que dominar los conceptos de importar/exportar variables, así como copiar y pegar bloques de programas. Además, se ve reflejada la importancia de la selección de los dispositivos de control y con ello, de los softwares de simulación asociados.



## 5.2. ESTACIÓN 6: CALCULADORA

### 5.2.1 Objetivo de la simulación

Es una estación que persigue la iniciación a la programación y familiarización con las herramientas de automatización por parte del usuario, al igual que la anterior. Por este motivo su complejidad también es baja. El principal objetivo de esta estación es que el usuario aprenda a programar y simular un HMI. A nivel de programación del autómatas se busca afianzar la utilidad de las variables de memoria, los comandos de operaciones matemáticas e introducir el uso de los bloques de programa. En lo referente a entornos de simulación, esta estación solo requiere el uso de un HMI que se simula con el software WinCC y se programa con el propio TIA Portal V17.

- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** Operadores matemáticos y elementos básicos de un HMI.
  - **Variables:** Input, output, memoria, importar/exportar variables y su conexión con el HMI.
  - **Programa:** Uso de OB, FC y DB.
  - **Conectividad:** Tipo de PLC, dirección IP y conectividad con PLCSIM.
- **Conocimientos que adquirir en la programación de un HMI con TIA Portal:**
  - **Componentes:** Librerías, elementos básicos y su colocación en la pantalla.
  - **Conectividad:** Del HMI con el PLC y simulación en WinCC.

### 5.2.2 Descripción de la escena

Como se ha mencionado con anterioridad, se trata de una estación cuyo único elemento que la conforma es un panel de operador (o HMI). Este HMI simula el funcionamiento de una calculadora, ofreciendo la posibilidad de realizar ciertas operaciones matemáticas entre dos operandos o dar información tras su comparación.

Aunque esta escena no tendría casi ningún sentido en un proceso industrial, la finalidad de esta simulación es el aprendizaje de conceptos relacionados con la automatización por el usuario. Es importante remarcar el carácter docente al que está enfocado este proyecto.

Para la gestión de la calculadora existen tres interruptores (EN1, EN2 Y EN3) y un pulsador (EN4) que se muestran en las imágenes pertinentes de la pantalla.

La calculadora permite realizar la suma, resta, división o multiplicación de los operandos en caso de que el pulsador EN1 se encuentre activado. En caso contrario, se obtiene el máximo y el mínimo de entre los dos operandos y el logaritmo natural, el redondeo superior e inferior y un conjunto de operaciones trigonométricas únicamente del operando 1.

El HMI consta de tres imágenes que permiten visualizar los operandos con los que se están trabajando, la selección en cada instante de los pulsadores de mando del HMI y los resultados de las operaciones seleccionadas.

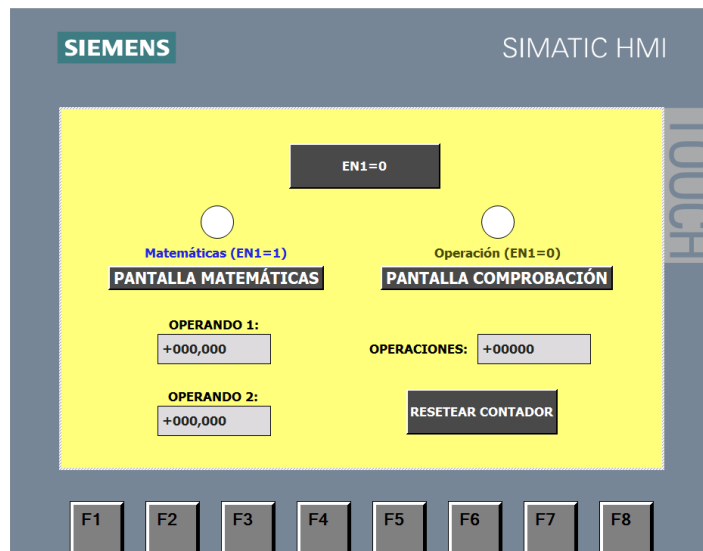


Figura 75. Imagen raíz. Estación calculadora

La pantalla de inicio, o imagen raíz, permite introducir los valores de los dos operandos, seleccionar mediante el interruptor EN1 el conjunto de operaciones que se quieren realizar y resetear el número de operaciones realizadas hasta el momento. Además, se puede acceder a la imagen “MATEMÁTICAS” mediante el botón “PANTALLA MATEMÁTICAS” y a la de “COMPROBACIÓN” mediante el botón “PANTALLA COMPROBACIÓN”.

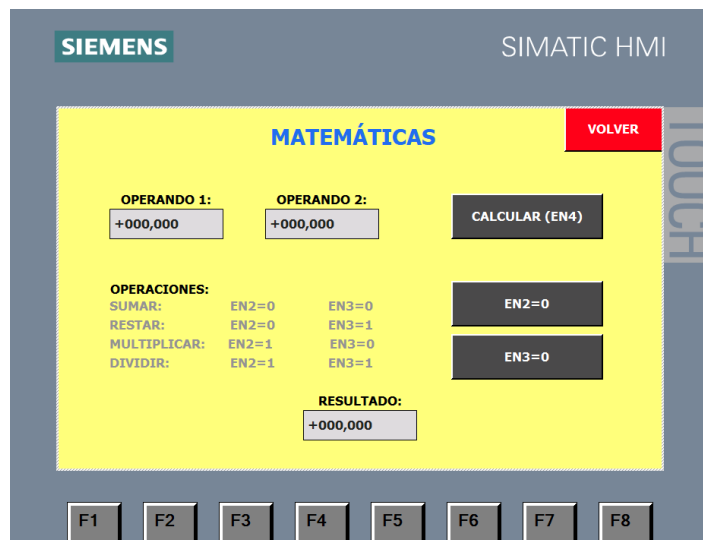


Figura 76. Imagen MATEMÁTICAS. Estación calculadora

En la imagen MATEMATICAS, siempre que EN1 este activado y mediante la combinación de los interruptores EN2 y EN3, se selecciona la operación que se desea realizar y al pulsar EN4 se muestra el resultado. En la misma pantalla se puede modificar los valores de los operandos para una mayor comodidad del usuario, así como volver a la imagen raíz con el botón “VOLVER”.



Figura 77. Imagen COMPROBACIÓN. Estación calculadora

La otra imagen a la que se podía acceder desde la imagen inicial es la de comprobación. En ella, además de poder modificar los operandos y volver a la imagen raíz como en la pantalla de matemáticas, muestra los diferentes datos que se observan en la imagen superior.

Antes de pasar al siguiente apartado, es importante conocer que la programación del PLC de esta estación es una de las prácticas de simulación que actualmente se les está pidiendo a los alumnos de los grados. El diseño de los controladores de la calculadora y la estructura del programa vienen descritos en el enunciado de la práctica y se ha decidido seguir este guion, por lo que la estructura de la programación no ha tenido que ser diseñada para este trabajo.

Sin embargo, se ha añadido la parte relacionada con el HMI para ampliar los conocimientos y para que la comprobación del correcto funcionamiento del programa del autómatas sea más visual; siendo diseño propio la estructura de las imágenes del panel de operador, así como la selección del HMI (TP700 Comfort).

Cabe destacar que el enunciado de esta práctica de simulación no estructura la programación de la calculadora en base a ningún GRAFCET. Aunque la programación de la calculadora se pudiera haber estructurado como un proceso secuencial en un GRAFCET, al no estar contemplado así en el enunciado original y teniendo en cuenta que no es un proceso estrictamente secuencial, se ha decidido no introducir la estructura GRAFCET para la programación.

### 5.2.3 Programación del autómeta

La programación del autómeta se realiza en lenguaje KOP, como en el resto de estaciones, sin embargo, no sigue ninguna estructura GRAFCET.

Las variables empleadas para la programación del autómeta están compuestas por el conjunto de interruptores y el pulsador como variables de entrada, y los diferentes datos matemáticos que se calculan y los operandos como variables de memoria, siendo el conjunto de ellas las reflejadas a continuación:

Variables PLC					
		Nombre	Tabla de variables e..	Tipo de datos	Dirección ▲
1		EN1	Tabla de variables e..	Bool	%I10.0
2		EN2	Tabla de variables e..	Bool	%I10.1
3		EN3	Tabla de variables e..	Bool	%I10.2
4		EN4	Tabla de variables e..	Bool	%I10.3
5		Resultado	Tabla de variables e..	Real	%MD8
6		Flanco1	Tabla de variables e..	Bool	%M12.0
7		Reset_Contador	Tabla de variables e..	Bool	%M12.1
8		Max	Tabla de variables e..	Real	%MD14
9		Min	Tabla de variables e..	Real	%MD18
10		Redondeo_Superior	Tabla de variables e..	Real	%MD22
11		Redondeo_Inferior	Tabla de variables e..	Real	%MD26
12		Raiz-Cuadrada	Tabla de variables e..	Real	%MD30
13		Log_Natural	Tabla de variables e..	Real	%MD34
14		Seno	Tabla de variables e..	Real	%MD38
15		Coseno	Tabla de variables e..	Real	%MD42
16		Tangente	Tabla de variables e..	Real	%MD46
17		Arco-Seno	Tabla de variables e..	Real	%MD50
18		Arco-Coseno	Tabla de variables e..	Real	%MD54
19		Arco_Tangente	Tabla de variables e..	Real	%MD58
20		Operando 1	Tabla de variables e..	Real	%MD64
21		Operando 2	Tabla de variables e..	Real	%MD68

Tabla 5. Variables del PLC. Estación calculadora

En esta simulación se emplean únicamente variables de entrada y de memoria, como se aprecia en la tabla anterior. Cabe destacar que, al estar compuesta la estación completa únicamente por un autómeta y un panel de operador, se podrían emplear todo variables de memoria, porque no hay ningún sensor o actuador que obligue a emplear las entradas o salidas del autómeta. Esto se debe a que la lectura y escritura de las variables se realiza mediante la conexión ETHERNET entre el PLC y el HMI dejando todos los pines de entradas y salidas de PLC libres.

Con el objetivo de que el usuario comprenda y aplique los bloques OB, FC y DB en la programación del autómeta, en esta estación se ha empleado un bloque de programa para cada operación matemática o conjunto de operaciones y para estructurar el programa.

Por lo tanto, la estructura del programa es la siguiente:

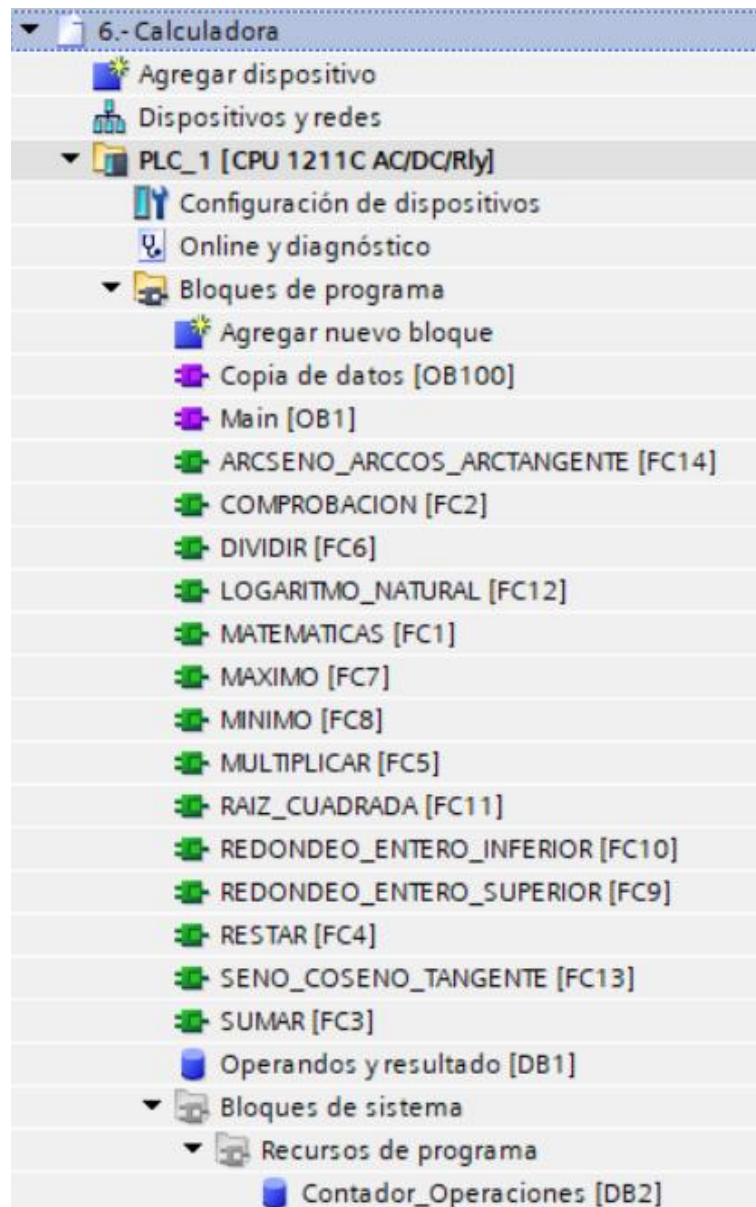


Figura 78. Distribución de bloques. Estación calculadora

A diferencia de las estaciones anteriores, no es necesario el bloque FC9000 para la conexión entre Factory IO y PLCSIM, puesto que no hay una escena formada por actuadores y sensores que simular en Factory IO.

Antes de comenzar a analizar el programa, es necesario clarificar dos aspectos para las personas que no estén familiarizadas con la programación de autómatas. El primero es el uso de un bloque de datos (DB1) donde se almacenan variables que se pueden usar en cualquier bloque de programa ya que la DB1 es global. El segundo aspecto que destacar es el uso de una OB distinta a la OB1. Las OB's tienen funciones diferentes según su numeración y sirven para organizar el programa.

A continuación, se analizan los bloques del programa mostrados en la figura anterior.

**Operandos y resultado [DB1]:** es el bloque de datos que almacena los valores, cargándose en el arranque del programa.

	Nombre	Tipo de datos	Offset	Valor de arranque
1	Static			
2	Operando 1	Real	0.0	0.0
3	Operando 2	Real	4.0	0.0
4	Resultado	Real	8.0	0.0
5	Maximo	Real	12.0	0.0
6	Mínimo	Real	16.0	0.0
7	Redondeo_Superior	Real	20.0	0.0
8	Redondeo_Inferior	Real	24.0	0.0
9	Raiz-Cuadrada	Real	28.0	0.0
10	Log_Natural	Real	32.0	0.0
11	Seno	Real	36.0	0.0
12	Coseno	Real	40.0	0.0
13	Tangente	Real	44.0	0.0
14	Arco-Seno	Real	48.0	0.0
15	Arco-Coseno	Real	52.0	0.0
16	Arco-Tangente	Real	56.0	0.0

Figura 79. Variables DB1 "Operandos y resultado". Estación calculadora

**Copia de datos [OB100]:** es un bloque de organización que se conoce como **OB de arranque** porque solo se ejecuta una única vez al pasar el PLC de STOP a RUN. Se emplea para mover los valores de arranque de la DB1 a los operandos 1 y 2 y para hacer el proceso inverso con el resto de las variables de memoria.

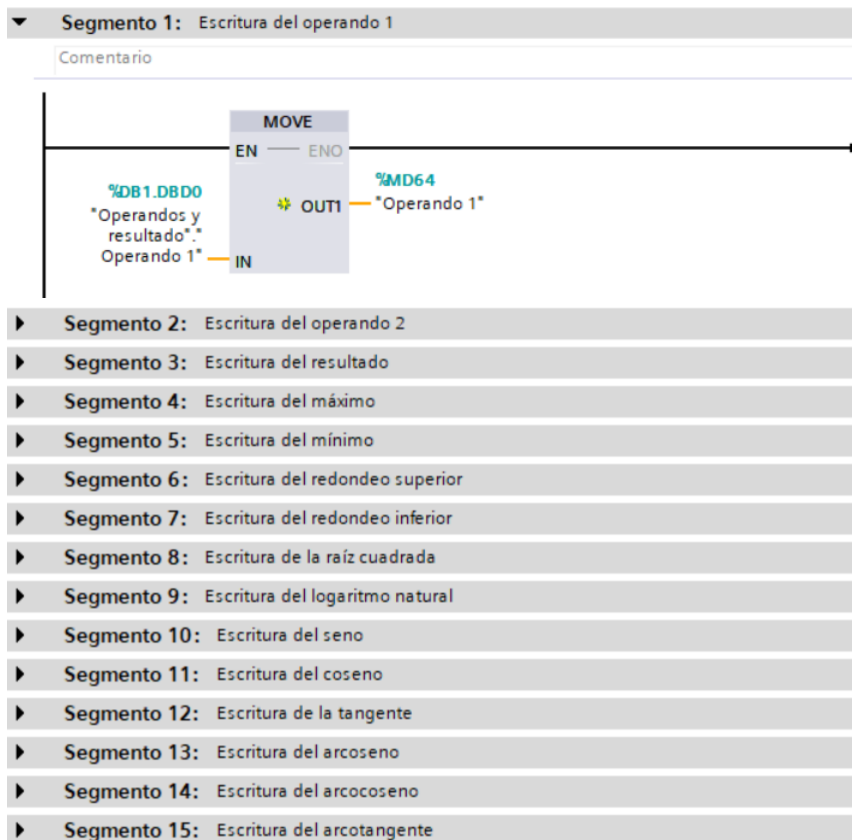


Figura 80. OB de arranque. Estación calculadora

**Main [OB1]:** contiene la selección del conjunto de operaciones que se desean calcular y el contador que registra el número de operaciones realizadas.

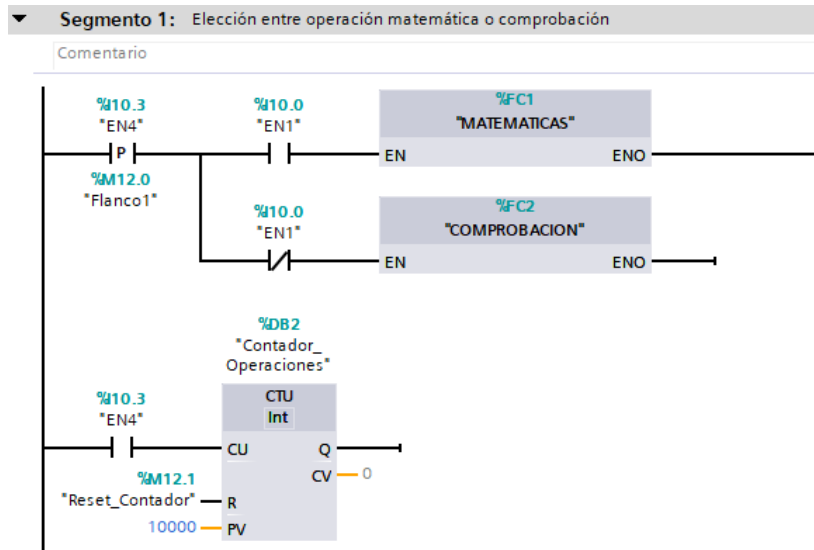


Figura 81. Bloque Main. Estación calculadora

**MATEMÁTICAS [FC1]:** se encarga de realizar el cálculo de 4 operaciones (suma, resta, multiplicación y división) en función de EN2 y EN3. Cada una de ellas se realiza en un **bloque FC** propio empleando los **comandos matemáticos** pertinentes.

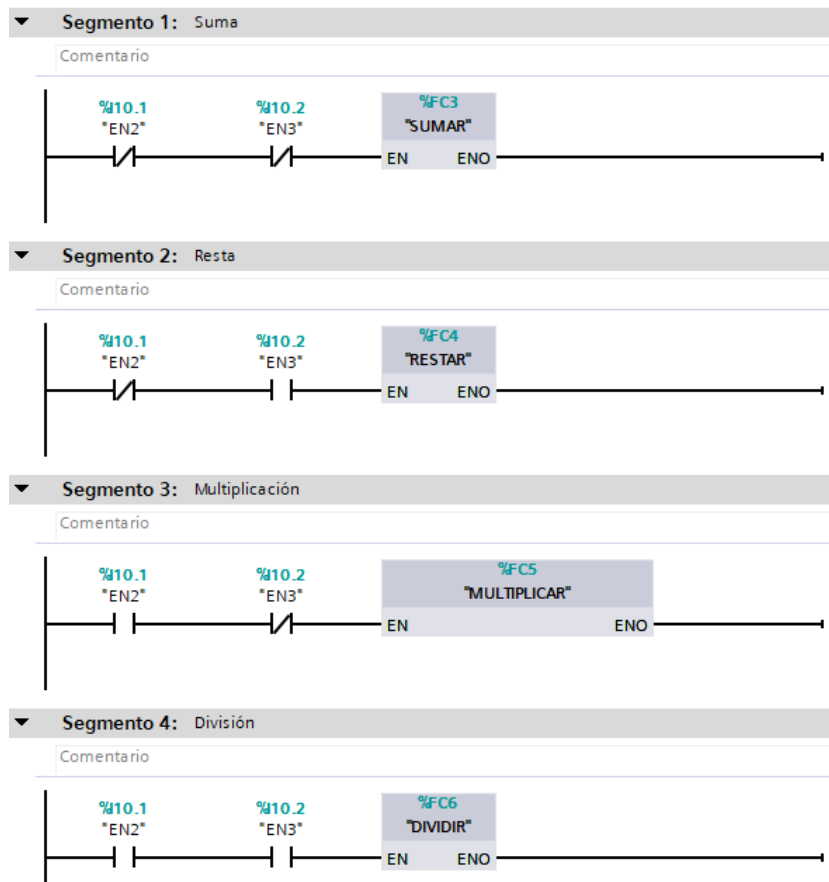


Figura 82. FC1 "MATEMÁTICAS". Estación calculadora

**COMPROBACIÓN [FC2]:** su función se reduce a un tema únicamente de estructuración del programa, encargándose de llamar (sin ninguna condición) a los **bloques FC's** encargados de realizar las diferentes operaciones matemáticas en su interior.

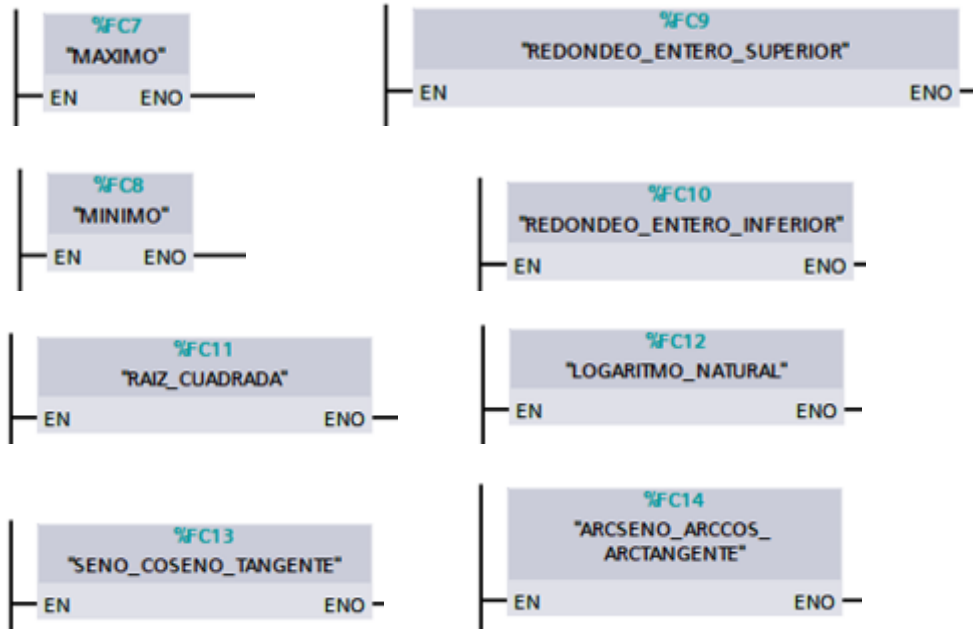


Figura 83. FC's contenidas en FC2 "COMPROBACIÓN". Estación calculadora

La representación del PLC con la herramienta PLCSIM es idéntica a la de la estación anterior y a todas las que empleen un PLC S7-1200, es por ese motivo que no se representará en lo sucesivo.

Se puede concluir, que para la realización del programa ladder, es necesario conocer y aplicar correctamente los conocimientos marcados en los objetivos a adquirir al comienzo de la descripción de la estación. Por lo tanto, se podría justificar el cumplimiento de la finalidad didáctica de esta práctica.

#### 5.2.4 Programación del HMI

Como se ha mencionado en el [apartado de descripción de la escena](#), esta estación está compuesta únicamente por un panel de operador. La programación de este elemento se realiza desde TIA Portal y comprende la creación de la conexión entre el PLC y el HMI, la vinculación de las variables pertinentes y su representación mediante diferentes elementos (pulsadores, botones o displays) en las imágenes que se desean mostrar por pantalla.

Para agregar el HMI al proyecto de TIA Portal, se selecciona el apartado de "Agregar dispositivo" en el árbol de proyecto y se añade un HMI TP700 Comfort. Realmente con un modelo más simple valdría para la función que se quiere desempeñar en esta simulación, pero como se ha mencionado anteriormente solo emplearemos un tipo de panel de operador.

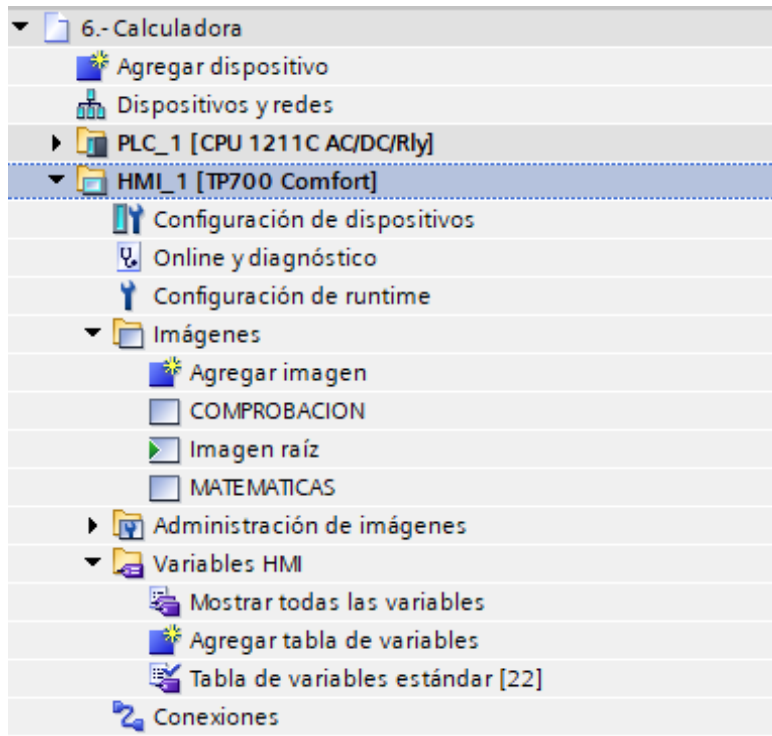


Figura 84. Árbol de proyecto del HMI. Estación calculadora

Una vez agregado el panel de operador, creamos la **conexión ETHERNET con el PLC**, asignando una dirección IP de la misma red y con un valor de 192.168.0.2. Mediante esta conexión se intercambia información entre ambos dispositivos de control.

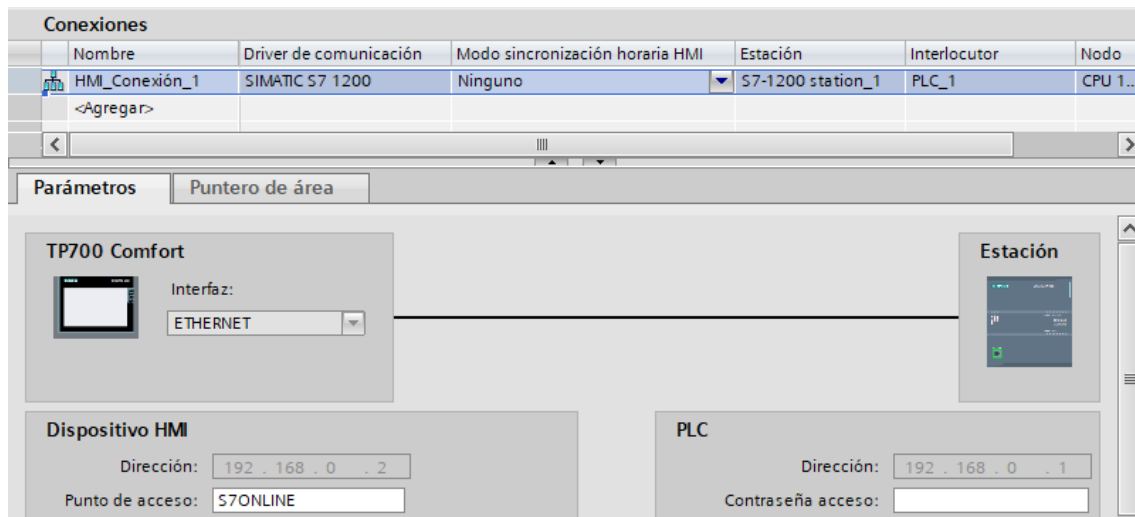


Figura 85. Conexión ETHERNET entre el HMI y el PLC. Estación calculadora

Como se aprecia en la *figura 84*, todas las variables empleadas en el HMI están contenidas en la tabla de variables estándar. Cabe destacar que para emplear las variables del PLC en el HMI se requiere asociarlas a una variable dentro del HMI.

Hay dos formas de **asociar las variables**, mediante acceso simbólico (se asocian por el nombre de la variable sin importar su dirección de memoria) y por acceso absoluto (se asocian a una dirección de memoria).

Las variables empleadas en el HMI son las siguientes:

Tabla de variables estándar							
	Nombre	Tipo de datos	Conexión	Nombre del PLC	Variable PLC	Direcci...	Modo de acceso
	HMI_Contador_Operaciones	Int	HMI_Conexión_1	PLC_1	Contador_Operaciones.CV		<Acceso simbólico>
	HMI_EN1	Bool	HMI_Conexión_1	PLC_1	EN1	%I10.0	<Acceso absoluto>
	HMI_EN2	Bool	HMI_Conexión_1	PLC_1	EN2	%I10.1	<Acceso absoluto>
	HMI_EN3	Bool	HMI_Conexión_1	PLC_1	EN3	%I10.2	<Acceso absoluto>
	HMI_EN4	Bool	HMI_Conexión_1	PLC_1	EN4	%I10.3	<Acceso absoluto>
	HMI_Reset_Contador	Bool	HMI_Conexión_1	PLC_1	Reset_Contador	%M12.1	<Acceso absoluto>
	HMI_Resultado	Real	HMI_Conexión_1	PLC_1	Resultado	%MD8	<Acceso absoluto>
	HMI_Maximo	Real	HMI_Conexión_1	PLC_1	Max	%MD14	<Acceso absoluto>
	HMI_Minimo	Real	HMI_Conexión_1	PLC_1	Mín	%MD18	<Acceso absoluto>
	HMI_Redondeo_Superior	Real	HMI_Conexión_1	PLC_1	Redondeo_Superior	%MD22	<Acceso absoluto>
	HMI_Redondeo_Inferior	Real	HMI_Conexión_1	PLC_1	Redondeo_Inferior	%MD26	<Acceso absoluto>
	HMI_Raiz_Cuadrada	Real	HMI_Conexión_1	PLC_1	"Raiz-Cuadrada"	%MD30	<Acceso absoluto>
	HMI_Log_Natural	Real	HMI_Conexión_1	PLC_1	Log_Natural	%MD34	<Acceso absoluto>
	HMI_Seno	Real	HMI_Conexión_1	PLC_1	Seno	%MD38	<Acceso absoluto>
	HMI_Coseno	Real	HMI_Conexión_1	PLC_1	Coseno	%MD42	<Acceso absoluto>
	HMI_Tangente	Real	HMI_Conexión_1	PLC_1	Tangente	%MD46	<Acceso absoluto>
	HMI_Arco-Seno	Real	HMI_Conexión_1	PLC_1	"Arco-Seno"	%MD50	<Acceso absoluto>
	HMI_Arco-Coseno	Real	HMI_Conexión_1	PLC_1	"Arco-Coseno"	%MD54	<Acceso absoluto>
	HMI_Arco-Tangente	Real	HMI_Conexión_1	PLC_1	Arco_Tangente	%MD58	<Acceso absoluto>
	HMI_Operando1	Real	HMI_Conexión_1	PLC_1	"Operando 1"	%MD64	<Acceso absoluto>
	HMI_Operando2	Real	HMI_Conexión_1	PLC_1	"Operando 2"	%MD68	<Acceso absoluto>

Tabla 6. Variables del HMI. Estación calculadora

Como se puede comprobar en la tabla anterior, no hay ninguna variable interna del HMI, sino que todas ellas son variables asociadas a las del PLC mediante la conexión que se había realizado previamente. En adicción a esta apreciación, también es destacable remarcar que la primera variable de la tabla tiene acceso simbólico, es por ese motivo por el cual no se muestra la dirección de la variable en el PLC a diferencia del resto.

En lo referente a las imágenes que contiene el HIM, como se han mostrado en el primer apartado relacionado con esta estación, se considera que no aportaría ninguna información adicional adjuntarlas también en este apartado. Sin embargo, sí que es importante recalcar que los elementos empleados en todas ellas se limitan a **cuadros de texto**, **botones**, **displays de salida** para visualizar los resultados y **displays de entrada y salida** para visualizar y modificar los operandos.

## 5.3. ESTACIÓN 7: CLASIFICADORA

### 5.3.1 Objetivo de la simulación

El nivel de complejidad de esta estación es algo más elevado que las dos anteriores, pudiéndose considerar necesarios unos conocimientos medios-bajos de automatización para la realización del programa del autómatas y la simulación de la estación completa. Con esta simulación, se persigue que el usuario comprenda y aplique las estructuras GRAFCET a la programación ladder en los diferentes bloques de programa. En lo referente al entorno de Factory IO, se tiene como finalidad conocer los principales elementos en una instalación más completa que las dos anteriores, así como el uso del sensor de visión.



- **Conocimientos que adquirir en TIA Portal:**
  - **Variables:** Uso de DB's globales o variables de memoria para los GAFSETS de las FC's.
  - **Programa:** Estructura GRAFCET en las FC's o FB's del programa.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Elementos básicos y sensor de visión.
  - **Conectividad:** Con PLCSIM.

### 5.3.2 Descripción de la escena

La finalidad de la estación es la automatización de un sistema de clasificación y conteo de piezas, empleando un sensor de visión, un conjunto de sensores difusos, cintas transportadoras y brazos pivotantes.

La estación industrial está compuesta por dos cintas transportadoras situadas de forma consecutiva, tres rampas perpendiculares a la segunda cinta transportadora y tres brazos pivotantes que empujan las piezas de la segunda cinta a la rampa correspondiente según el tipo de pieza.

Las piezas entran, en orden aleatorio, por la primera cinta ("*cinta generadora*") a la altura del primer sensor ("*sensor cinta generadora*") encargado de detectar la entrada de piezas y activar el motor de la cinta. Las piezas se detienen al llegar a un sensor de visión situado sobre la cinta generadora. Este sensor clasifica las piezas basándose en su forma y color en tres tipos mediante la asignación de un número (3= bases azules, 6= bases verdes y 2= tapas azules).

Una vez asignado el tipo de pieza, la cinta generadora se vuelve a activar hasta que la pieza pasa a la siguiente cinta ("*cinta distribuidora*"), suceso detectado por el flanco de bajada del "*sensor cinta distribuidora*". En el lateral derecho de esta cinta, y de forma consecutiva, se encuentran tres brazos pivotantes cuya función es clasificar las piezas en función de su tipo (ya asignado con el sensor de visión). Según se trate de bases azules, bases verdes o tapas azules, el brazo correspondiente pivota y activa su cinta. De esta forma, al llegar la pieza a su brazo correspondiente, este la empujará hacia la rampa situada a un lado de la cinta distribuidora, clasificando así, los tres tipos de piezas según por qué rampa salgan de la estación.

Con el fin de controlar la estación, se dispone de un cuadro eléctrico que contiene una seta de emergencia, un pulsador de marcha con iluminación, un pulsador de paro con iluminación, tres displays y un pulsador de reset.

Si se pulsa el botón de marcha, la estación entra en estado activo, esperando a que llegue la primera pieza y funcionando de forma totalmente automática, indicándose con una luz verde en el propio pulsador de marcha. Si se pulsa el botón de paro, independientemente de la señal de los sensores, la estación entra en estado de

reposo, indicándose con una luz roja en el propio botón de paro. Para reanudar el estado de marcha basta con volver a pulsar el botón de marcha.

En caso de emergencia, si se pulsa la seta de emergencia la estación se detiene y se activa la luz de emergencia. Al ser una seta de emergencia, será necesario girar la seta para quitar el enclavamiento y pulsar el botón de marcha. Si no se realiza esta operación en ese orden, no se reanudará la marcha para evitar que se produzca una situación potencialmente peligrosa para el operario o para la instalación.

Las piezas de cada tipo, clasificadas por la estación, se muestran en los tres displays del cuadro eléctrico. Encima de ellos se encuentra un botón de reset que reinicia todos los contadores.

La escena elaborada en Factory IO, explicada de forma escrita en los párrafos anteriores, se muestra a continuación de forma visual:

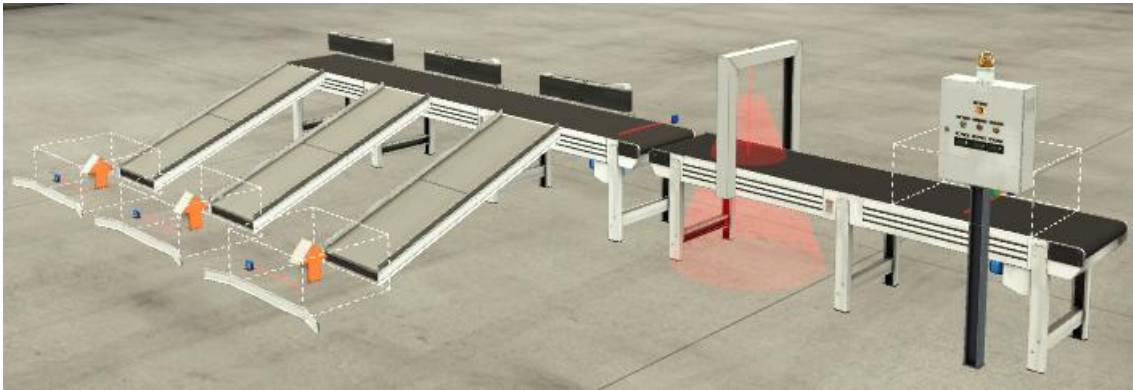


Figura 86. Vista general. Escena clasificadora



Figura 87. Vista cuadro eléctrico. Escena clasificadora

La distribución en el driver de Factory IO de las variables de entradas y salidas empleadas en el programa del PLC es la siguiente:

Sensor cinta generadora	%I15.0	%Q0.0	Generador de piezas
Sensor cinta distribuidora	%I15.1	%Q0.1	Motor cinta generadora
	%I15.2	%Q0.2	
	%I15.3	%Q0.3	
Sensor eliminación base azul	%I15.4	%Q0.4	Motor cinta distribuidora
Sensor eliminación base verde	%I15.5	%Q0.5	Señalización emergencia
Sensor eliminación tapa azul	%I15.6	%Q0.6	Señalización paro
	%I15.7	%Q0.7	Señalización marcha
Marcha	%I16.0	%Q1.0	Pistón base azul
Paro	%I16.1	%Q1.1	Pistón base verde
Reset contadores	%I16.2	%Q1.2	Pistón tapa azul
Parada de emergencia	%I16.3	%Q1.3	Cinta pistón base azul
	%I16.4	%Q1.4	Cinta pistón base verde
	%I16.5	%Q1.5	Cinta pistón tapa azul
	%I16.6	%Q1.6	
	%I16.7	%Q1.7	
Sensor de visión	%IW30 (INT)	(INT) %QW30	Número bases azules
	%IW32	(INT) %QW32	Número bases verdes
	%IW34	(INT) %QW34	Número tapas azules
	%IW36	%QW36	
	%IW38	%QW38	
	%IW40	%QW40	
	%IW42	%QW42	
	%IW44	%QW44	

Figura 88. Distribución de variables. Escena clasificadora

### 5.3.3 Descripción GRAFCET de la programación

Parte de la programación del PLC se guía por estructuras GRAFCET, que describen de forma parcial los estados de diferentes grupos de componentes de la estación. De tal forma, la agregación de GRAFCET's consolidan una programación estructurada por conjuntos de elementos. Todos estos GRAFCET son de nivel 1 empleando una descripción funcional del proceso secuencial, reflejando de forma genérica el proceso sin emplear variables del programa en el diagrama.

### GRAFSET cinta generadora

Este diagrama de bloques refleja los diferentes estados de la cinta generadora, describiendo su activación y desactivación, así como la solicitud de activación de la cinta distribuidora en función de las situaciones de las piezas que transporta.

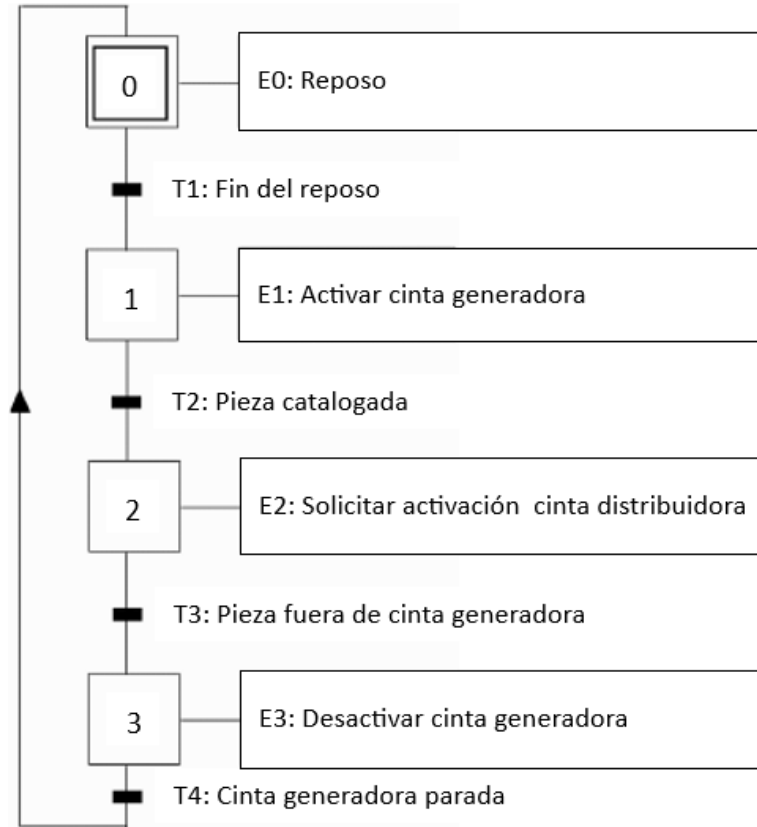


Figura 89. GRAFCET cinta generadora. Estación clasificadora

### GRAFSET cinta distribuidora

Este diagrama de bloques refleja los diferentes estados de la cinta distribuidora describiendo su activación y desactivación, en función de las piezas que transporta y de la cinta generadora.

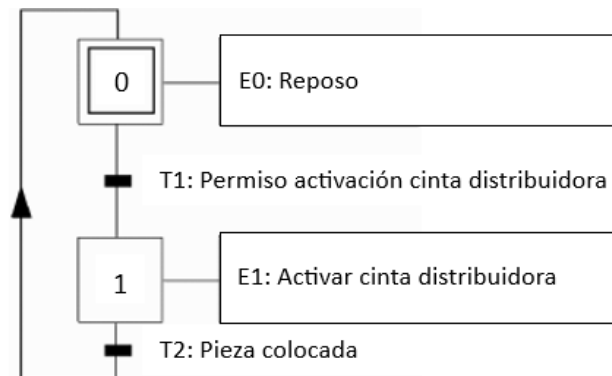


Figura 90. GRAFCET cinta distribuidora. Estación clasificadora

### GRAF CET barreras de la cinta distribuidora

Este diagrama de bloques refleja los diferentes estados de los brazos pivotantes situados en el lateral de la cinta distribuidora. Estos brazos actúan como barreras que empujan las piezas a las rampas correspondientes mediante unas cintas integradas en ellos. Sabiendo este funcionamiento del elemento, podemos comprender el diagrama que explica el funcionamiento de los brazos; activando su rotación y su cinta únicamente en los momentos precisos, permaneciendo retraídos y con la cinta apagada en el resto de los casos.

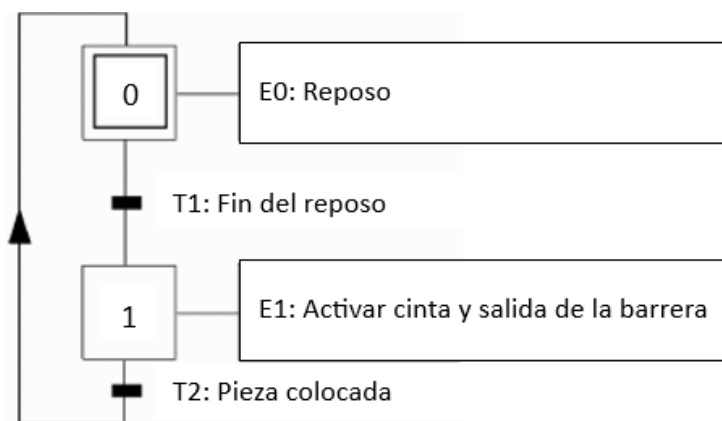


Figura 91. GRAFCET barreras de la cinta distribuidora. Estación clasificadora

Mediante la combinación de estos GRAFCET's, y desarrollando de forma completa cada proceso, seremos capaces de estructurar gran parte del programa del PLC. El resto de procesos se limitan a la gestión de la marcha, el aumento de los contadores y gestión del arranque del controlador. No siendo estos estrictamente secuenciales, no necesitan diagramas GRAFCET para estructurarlos.

#### 5.3.4 Programación del autómatas

La programación del autómatas se realiza en lenguaje KOP ajustándose a los GRAFCET's descritos anteriormente.

Como se ha mencionado con anterioridad, una de las principales dificultades de esta práctica es la estructuración de la programación en diversas FC's y FB's aplicando los diagramas GRAFCET. Para ello es necesario que las etapas se mantengan activas.

Este problema se soluciona fácilmente en las FB's asignando a cada etapa una variable estática para que su valor se mantenga entre ciclos de scan o activaciones de las FB's. Sin embargo, en las FC's es más complejo lograrlo al no tener un bloque de datos que se mantiene de forma permanente. Para estos casos lo que se emplean son marcas del PLC incluidas en la tabla de variables estándar.

Las variables empleadas en la programación del autómatá comprenden las entradas y salidas empleadas en el driver de Factory IO, así como variables internas del PLC estructuradas en DB's o definidas directamente como marcas y los DB's asociados a las FB's. Las variables de la tabla estándar del PLC se muestran a continuación:

Tabla de variables estándar				
		Nombre	Tipo de datos	Dirección
1		Sensor cinta generadora	Bool	%I15.0
2		Sensor cinta distribuidora	Bool	%I15.1
3		Sensor eliminacion base azul	Bool	%I15.4
4		Sensor eliminacion base verde	Bool	%I15.5
5		Sensor eliminacion tapa azul	Bool	%I15.6
6		Marcha	Bool	%I16.0
7		Paro	Bool	%I16.1
8		Reset contadores	Bool	%I16.2
9		Parada de emergencia	Bool	%I16.3
10		Sensor de vision	Int	%IW30
11		Generador de piezas	Bool	%Q0.0
12		Motor cinta generadora	Bool	%Q0.1
13		Motor cinta distribuidora	Bool	%Q0.4
14		Señalización emergencia	Bool	%Q0.5
15		Señalización paro	Bool	%Q0.6
16		Señalización marcha	Bool	%Q0.7
17		Piston base azul	Bool	%Q1.0
18		Piston base verde	Bool	%Q1.1
19		Piston tapa azul	Bool	%Q1.2
20		Cinta piston base azul	Bool	%Q1.3
21		Cinta piston base verde	Bool	%Q1.4
22		Cinta piston tapa azul	Bool	%Q1.5
23		Numero bases azules	Int	%QW30
24		Numero bases verdes	Int	%QW32
25		Numero tapas azules	Int	%QW34
26		Activado	Bool	%M10.0
27		etapa 1	Bool	%M110.0
28		etapa 2	Bool	%M110.1
29		etapa 3	Bool	%M110.2
30		tipo pieza	Int	%MW111
31		activar motor generadora	Bool	%M113.0
32		desactivar motor generadora	Bool	%M113.1
33		aux estado motor generadora	Bool	%M113.2
34		etapa 1	Bool	%M120.0
35		activar motor distribuidora	Bool	%M120.1
36		desactivar motor distribuidora	Bool	%M120.2
37		estado distribuidora	Bool	%M120.3
38		aux estado motor distribuidora	Bool	%M120.5
39		activar distribuidor	Bool	%M200.0

Tabla 7. Variables del PLC. Estación clasificadora

Como ya se ha mencionado anteriormente, las variables de entrada y salida son las empleadas en el programa de simulación del entorno virtual de Factory IO, compartiendo las mismas direcciones para el correcto funcionamiento de la simulación.

La estructura del programa es la siguiente:

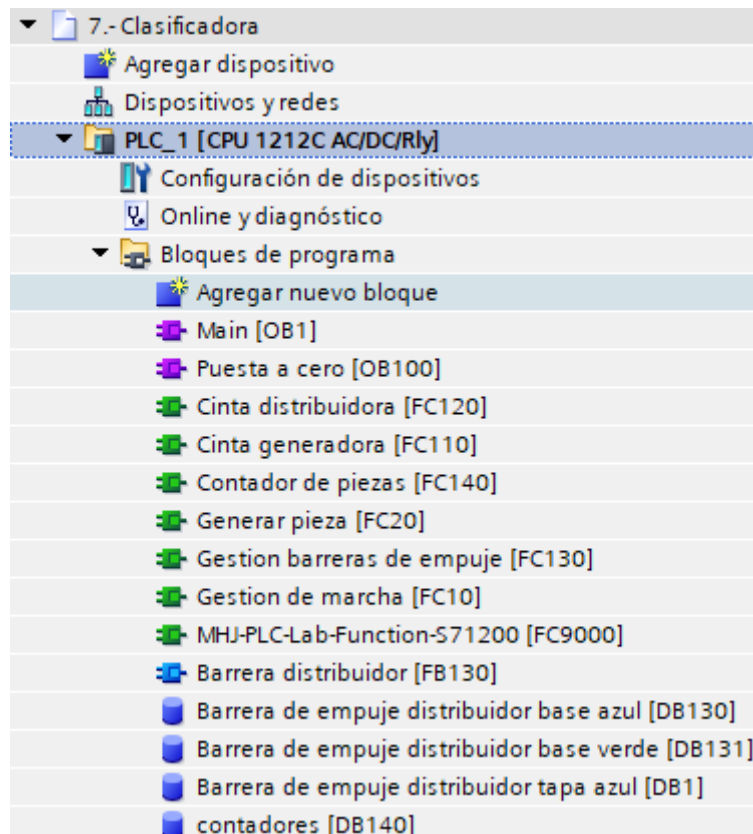


Figura 92. Distribución de bloques. Estación clasificadora

Como se observa en la figura superior, hay dos bloques de organización de programa:

**Puesta a cero [OB100]:** es la OB de arranque encargada de resetear todas las variables de salida y preparar la estación para que cada vez que el PLC pase a RUN la maquina se encuentre en una situación segura.

**Main [OB1]:** contiene el resto de la programación, encargándose de llamar al resto de bloques de programa y realizar las operaciones necesarias con las salidas de estos bloques.

Los bloques contenidos en el main y la descripción de la función que desempeñan se analizan a continuación:

La **FC9000**, como ya se ha mencionado y mostrado en numerosas ocasiones a lo largo del trabajo, sirve para conectar de forma correcta el PLC con Factory IO.

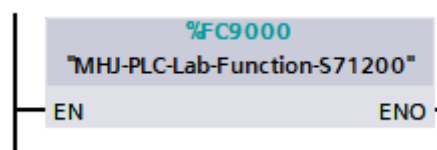


Figura 93. FC9000 conexión con Factory IO. Estación clasificadora

**Gestión de marcha [FC10]:** como su nombre indica, se encarga de gestionar la marcha y paro de la estación según lo explicado en la descripción de la estación, así como de encender las señales luminosas que indican el estado de la estación. Al trabajar con entradas y salidas del PLC o con marcas, no es necesario definir ninguna entrada o salida dentro de la FC.



Figura 94. FC10 Gestión de marcha. Estación clasificadora

**Generar pieza [FC20]:** en este bloque solamente se programa la llegada de una nueva pieza a la estación, controlando el elemento de “emisor” de la escena de Factory IO. Como en el caso anterior, solo trabaja con variables de entrada y salida del PLC y con marcas, por lo que tampoco es necesario definir entradas y salidas para la propia FC. Como se aprecia en la figura 95, su activación está condicionada al estado de activación de la estación completa, gestionada en la FC10.

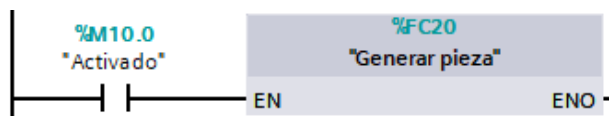


Figura 95. FC20 Generar pieza. Estación clasificadora

**Cinta generadora [FC110]:** contiene la programación estructurada según su GRAFCET, controlando la activación y desactivación del motor de su cinta y posibilitando la activación de la cinta distribuidora. Además, también contiene la asignación de cada tipo de pieza en función de la lectura del **sensor de visión**. Esta FC emplea las marcas M110.0, M110.1 y M110.2 para mantener las etapas de su GRAFCET activas cuando esto se requiera. Para que la FC cumpla con su función de ser reutilizable, sí que están definidas un conjunto de entradas y salidas.

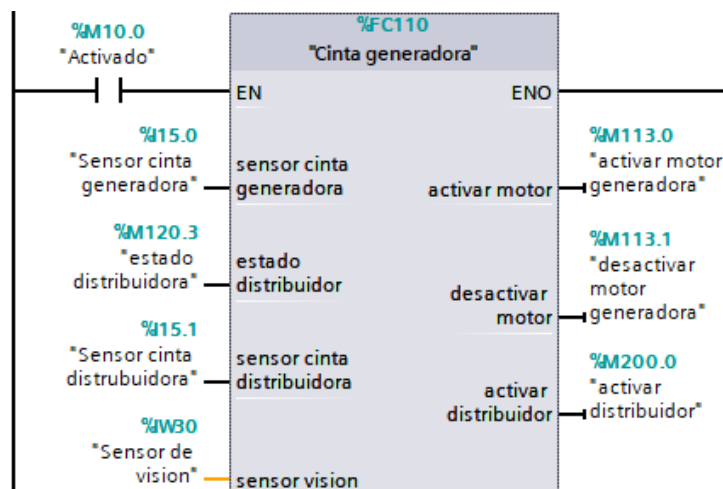


Figura 96. FC110 Cinta generadora. Estación clasificadora

**Cinta distribuidora [FC120]:** gestiona el control de la cinta distribuidora, ajustándose al GRAFCET de esta, empleando la **variable interna M120.0** para mantener su única etapa activa cuando esto se requiera. Como en la mayoría de los casos, esta FC también requiere de entradas y salidas para su correcto funcionamiento.

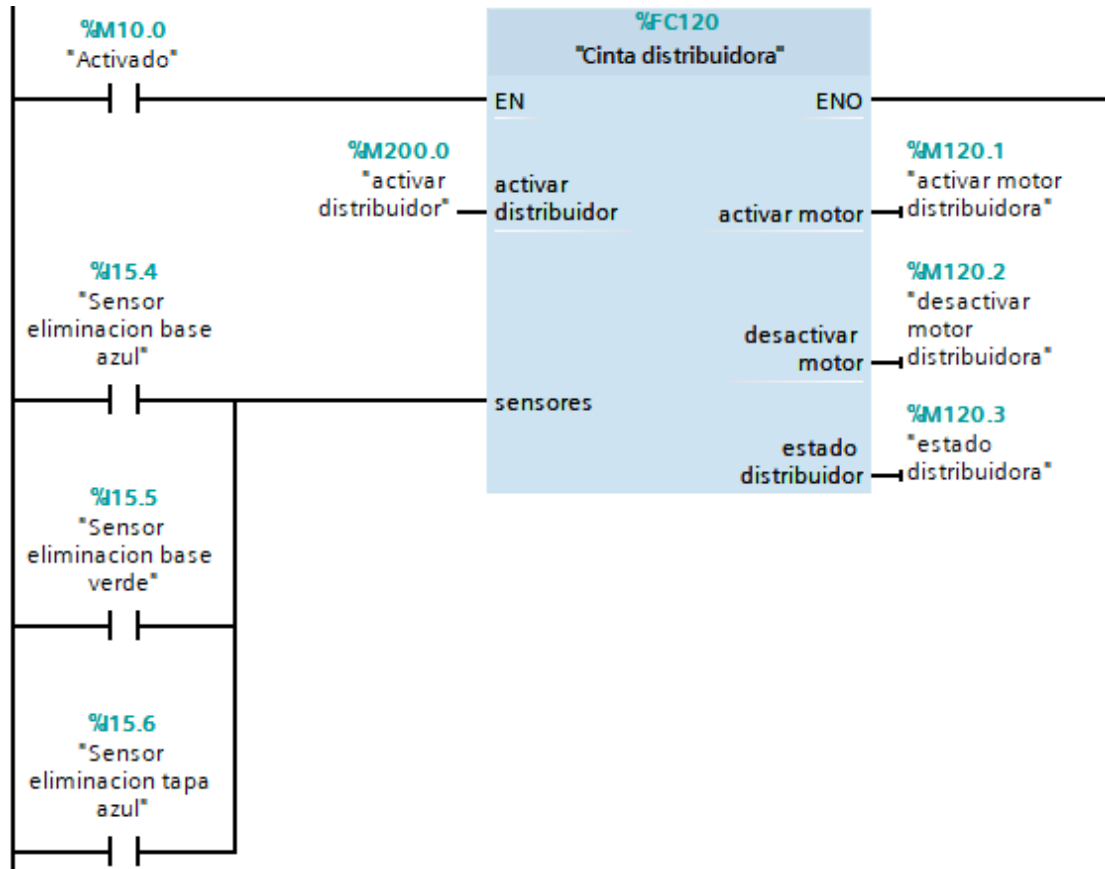


Figura 97. FC120 Cinta distribuidora. Estación clasificadora

**Gestión barreras de empuje [FC130]:** la única función de esta FC es la agrupación de las tres llamadas diferentes que se realiza a la FB encargada de la gestión de cada barrera, con el fin de que el programa sea más visual y estructurado. Como es lógico, puesto que cada FB tiene un bloque de datos asociado y sabiendo que solo se trabaja con entradas y salidas del PLC y variables de memoria, esta FC tampoco necesita ninguna entrada o salida propias.

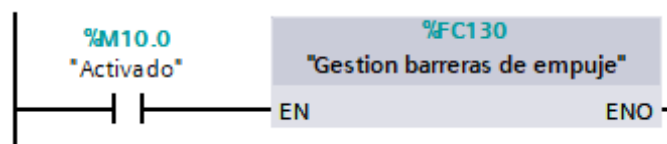


Figura 98. FC130 Gestión barreras de empuje. Estación clasificadora

**Barrera distribuidor [FB130]:** contiene el GRAFCET, programado en lenguaje KOP, que estructura el funcionamiento de los brazos giratorios. Este proceso secuencial, controla las condiciones de activación de la cinta del brazo y del pistón que lo gira. Los tres brazos que posee esta estación tienen el mismo funcionamiento, es por ello por lo que esta FB se puede emplear para los tres casos. Como es lógico, cada brazo está condicionado a unas entradas y activa unas salidas distintas al resto. Razón por la cual, cada uno requiere un **almacenamiento de sus etapas**. Por estos motivos este conjunto de instrucciones se realiza en una FB, puesto que con cada llamada se define un **DB de instancia** que almacena la información de cada brazo.

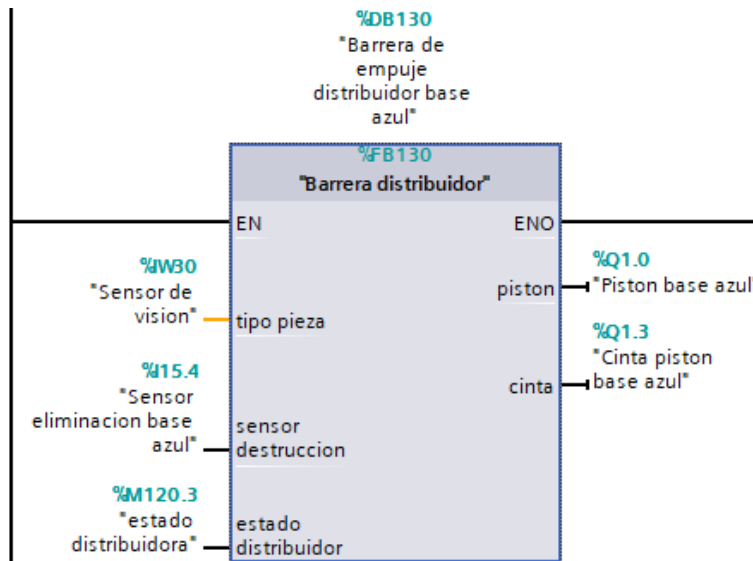


Figura 99. FB130 Barrera distribuidor bases azules. Estación clasificadora

**Contador de piezas [FC140]:** esta última FC contienen los tres contadores de piezas clasificadas por la estación.

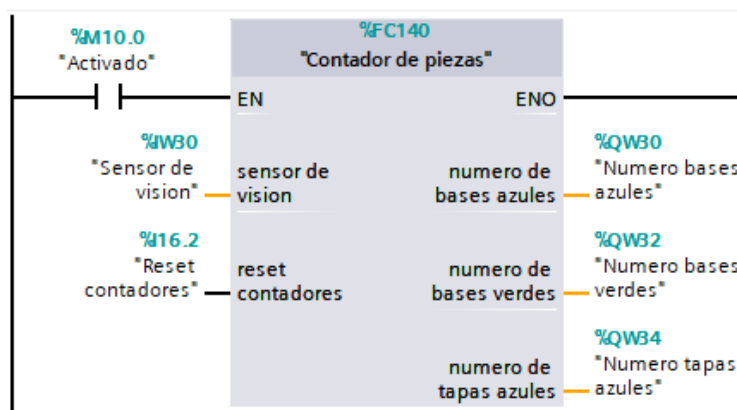


Figura 100. FC140 Contador de piezas. Estación clasificadora

Tras la explicación del programa, se determina la necesidad de conocer y aplicar todos los objetivos descritos al inicio. Siendo necesario emplear marcas y DB's para la estructura GRAFCET y conocer el funcionamiento y uso del sensor de visión tanto para su simulación en Factory IO como la lectura de su señal para el programa ladder.



## 5.4. ESTACIÓN 8: COLOCADOR POR PESO

### 5.4.1 Objetivo de la simulación

Es la estación de mayor complejidad junto con la estación de gestión de stock. A nivel conceptual de programación, no aporta ninguna información nueva, sin embargo, profundiza y clarifica el uso de todos los conocimientos empleados hasta el momento. La estación industrial que se simula en Factory IO es la más extensa y compleja de todo el proyecto, pero solo incluye un nuevo elemento, una báscula. A nivel de programación, el objetivo diseñado para el usuario no es programar de cero la estación completa, sino hacer uso de los bloques de programas de diversas estaciones anteriores, adaptarlos y emplearlos en la nueva estación. De esta forma, queda de manifiesto la característica reutilizable del código y los beneficios de estructurarlo en bloques de programa. Por contraposición, también se manifiesta la necesidad de adaptar el código o el tipo de bloque de programa según el funcionamiento de cada estación.

- **Conocimientos que adquirir en TIA Portal:**
  - **Programa:** Diferenciar FB y FC, usar correctamente los DB's y no usar marcas.
  - **Conectividad:** Tipo de PLC y dirección IP.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Elementos básicos, robots y báscula.

### 5.4.2 Descripción de la escena

La finalidad de la estación es la separación automatizada de tres tipos de cajas según su peso (pequeñas, medianas o grandes), almacenando, mediante varios robots, las cajas pequeñas en cajones apilables, las cajas medianas en pallets y las grandes de forma individual.

La distribución de la estación industrial es similar a una cruz compuesta por un conjunto de cintas transportadoras, una báscula transportadora y un distribuidor de rodillos en su centro. Al final de las cintas transportadoras de la derecha y de la izquierda se sitúa un transportador de rodillos y un robot en la intersección entre ambos elementos de transporte y al final de la cinta transportadora delantera una rampa por donde salen las cajas grandes.

Las cajas entran en un orden aleatorio por la cinta de entrada, detectando su presencia mediante el "sensor caja". Seguidamente, y si la báscula no está ocupada, pasan a la báscula transportadora, parándose a la altura del "sensor posición báscula" para su pesaje. Una vez pesadas y si el distribuidor no está ocupado, avanzan hasta el distribuidor de rodillos, encargado de redirigir las cajas a derecha, izquierda o de frente en función de si la caja es pequeña, mediana o grande respectivamente.



Las cajas pequeñas, ya redirigidas a la cinta transportadora derecha, avanzan por la cinta hasta llegar al “*sensor final cinta derecha*”, deteniéndose la cinta y activando el robot de dos ejes situado justo en frente del final de la cinta.

De forma perpendicular a esta cinta y debajo del robot de dos ejes, está emplazado un transportador de rodillos, por donde entran pallets con un cajón apilable encima. Los pallets avanzan hasta la mitad del transportador de rodillos, donde está emplazado un rodillo de parada, deteniendo el pallet debajo del robot y delante del final de la cinta transportadora derecha.

Si hay una caja al final de la cinta, el robot de dos ejes se activa, recogiendo la caja y situándola dentro del cajón apilable, cuando el pallet llegue a la posición de carga. Cuando el robot ha apilado 3 cajas pequeñas, el pallet con el cajón lleno avanza saliendo de la estación por el final del transportador de rodillos.

Por otro lado, la rama izquierda, correspondiente a las cajas medianas, es muy similar a la estación paletizadora. Las cajas medianas entran por la cinta transportadora izquierda, avanzando por la cinta hasta llegar al “*sensor final cinta derecha*”, deteniéndose la cinta y activando un posicionador de elementos, ubicado al final de ella, para situar la caja en la posición correcta donde lo agarrará el robot mediante una ventosa de succión.

De forma perpendicular a la cinta transportadora está emplazado un transportador de rodillos dotado con dos sensores difusos, uno donde entran los pallets y el otro en la posición donde se colocarán las cajas.

A la mitad de esta cinta se encuentra el robot cartesiano y un rodillo de parada que detecta que el pallet está situado en la posición correcta para la colocación de cajas. Los rodillos se detienen hasta que se sitúan seis cajas sobre el pallet con una colocación específica. Posteriormente, el rodillo de parada se retrae, permitiendo el avance del pallet cargado y los rodillos vuelven a activarse.

Por último, la rama delantera permite la salida de las cajas grandes mediante una cinta transportadora y una rampa.

Con el fin de controlar la estación, se dispone de un cuadro eléctrico que contiene una seta de emergencia, un pulsador de marcha con iluminación, un pulsador de paro con iluminación, cuatro displays y dos pulsadores de reset.

El funcionamiento de la gestión de la marcha y paro es la misma que en la estación cinta transportadora. Por este motivo no se vuelve a explicar.

Tres de los cuatro displays del cuadro de mando se emplean para mostrar los pallets llenos de cajas medianas, los pallets con los cajones apiladores llenos de cajas pequeñas y las cajas grandes procesadas por la estación. Estos tres displays se pueden reestablecer a cero empleando el botón de “*reset contadores*” del cuadro de mando. El otro botón de reset se emplea para resetear los contadores de cada una de las cintas que llevan la caja del distribuidor al final de su rama. El último display muestra el peso de la última caja que ha pasado por el proceso de pesaje.

Con el objetivo de visualizar mejor la estación industrial descrita, se muestra a continuación la escena que se emplea en la simulación.

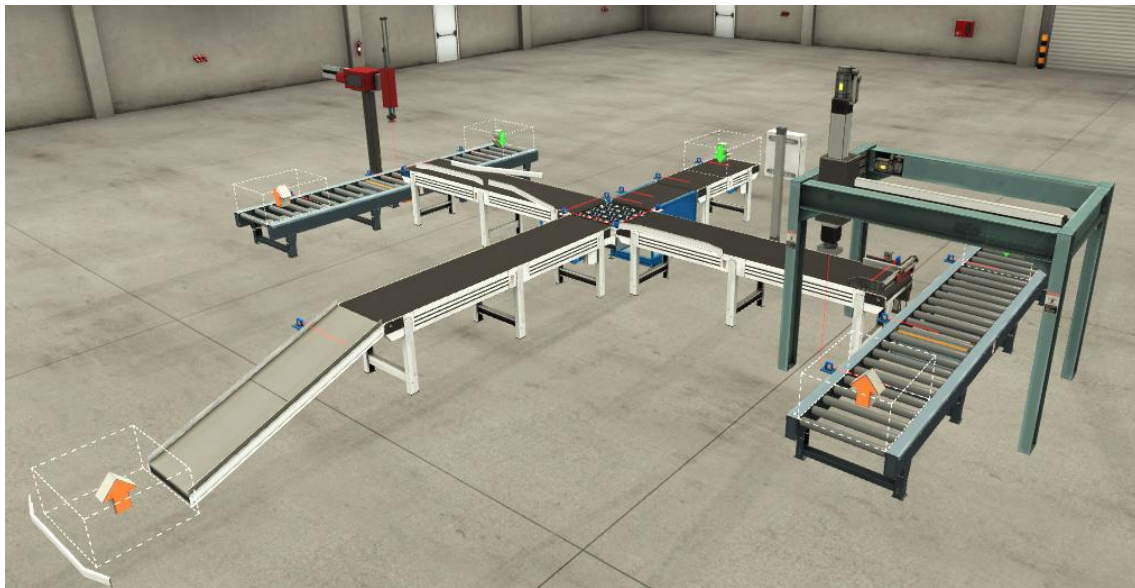


Figura 101. Vista general. Escena colocador por peso



Figura 102. Vista cuadro eléctrico. Escena colocador por peso

Para completar la simulación del entorno virtual, es necesario asociar las variables que se deseen emplear en la programación del PLC como entradas o salidas con una dirección del driver de Factory IO. El resultado de la distribución es el siguiente:

Marcha	%I10.0	%Q0.0	Luz_Marcha
Paro	%I10.1	%Q0.1	Luz_Paro
Emergencia	%I10.2	%Q0.2	
	%I10.3	%Q0.3	
	%I10.4	%Q0.4	
	%I10.5	%Q0.5	
	%I10.6	%Q0.6	
Reset Contadores Cintas	%I10.7	%Q0.7	
Sensor caja	%I11.0	%Q1.0	Motor cinta entrada
Reset Contadores	%I11.1	%Q1.1	Motor Bascula
Sensor inicio bascula	%I11.2	%Q1.2	Izquierda
Sensor posicion bascula	%I11.3	%Q1.3	Derecha
Sensor final bascula	%I11.4	%Q1.4	Delante
Sensor inicio cinta izquierda	%I11.5	%Q1.5	Motor cinta izquierda
Sensor final cinta izquierda	%I11.6	%Q1.6	Motor cinta derecha
Sensor inicio cinta derecha	%I11.7	%Q1.7	Motor cinta delantera
Sensor final cinta derecha	%I12.0	%Q2.0	Neumatica Piston
Sensor inicio cinta delantera	%I12.1	%Q2.1	Succion Cartesiano
Sensor final cinta delantera	%I12.2	%Q2.2	Giro pinza cartesiano
Sensor Piston	%I12.3	%Q2.3	Motor rodillos izquierda
Sensor inicio rodillos izquierda	%I12.4	%Q2.4	Rodillo parada Izquierda
Sensor final rodillos izquierda	%I12.5	%Q2.5	Succion_RobotXZ
Sensor pallet izquierdo en posicion	%I12.6	%Q2.6	Motor rodillos derecha
Sensor inicio rodillos derecha	%I12.7	%Q2.7	Rodillo parada Derecha
Sensor final rodillos derecha	%I13.0	%Q3.0	
Sensor pallet derecho en posicion	%I13.1	%Q3.1	
	%I13.2	%Q3.2	
	%I13.3	%Q3.3	
	%I13.4	%Q3.4	
	%I13.5	%Q3.5	
	%I13.6	%Q3.6	
	%I13.7	%Q3.7	
Peso	%ID30 (REAL)	(REAL) %QD20	Peso Actual
Cartesiano posicion X (V)	%ID34 (REAL)	(DINT) %QD24	Pallets Pequeños
Cartesiano posicion Y (V)	%ID38 (REAL)	(DINT) %QD28	Pallets Medianos
Cartesiano posicion Z (V)	%ID42 (REAL)	(DINT) %QD32	Cajas Grandes
RobotXZ_Posicion_X	%ID46 (REAL)	(REAL) %QD36	Cartesiano Desplazamiento X
RobotXZ_Posicion_Z	%ID50 (REAL)	(REAL) %QD40	Cartesiano Desplazamiento Y
		(REAL) %QD44	Cartesiano Desplazamiento Z
		(REAL) %QD48	RobotXZ_Desplazamiento_X
		(REAL) %QD52	RobotXZ_Desplazamiento_Z

Figura 103. Distribución de variables. Escena colocador por peso

### 5.4.3 Descripción GRAFCET de la programación

Parte de la programación del PLC se guía por estructuras GRAFCET, que describen de forma parcial los estados de diferentes grupos de componentes de la estación. De tal forma, la agregación de GRAFCET's consolidan una programación estructurada por conjuntos de elementos. Todos estos GRAFCET son de nivel 1 empleando una descripción funcional del proceso secuencial, reflejando de forma genérica el proceso sin emplear variables del programa en el diagrama.

#### GRAFCET cinta cajas medianas

El diagrama refleja los diferentes estados de la cinta transportadora de cajas medianas, el colocador de elementos situado al final de esta y la solicitud de activación del robot cartesiano. Estos estados dependen de los sensores situados en la cinta, en el posicionador y de las cajas que transporta.

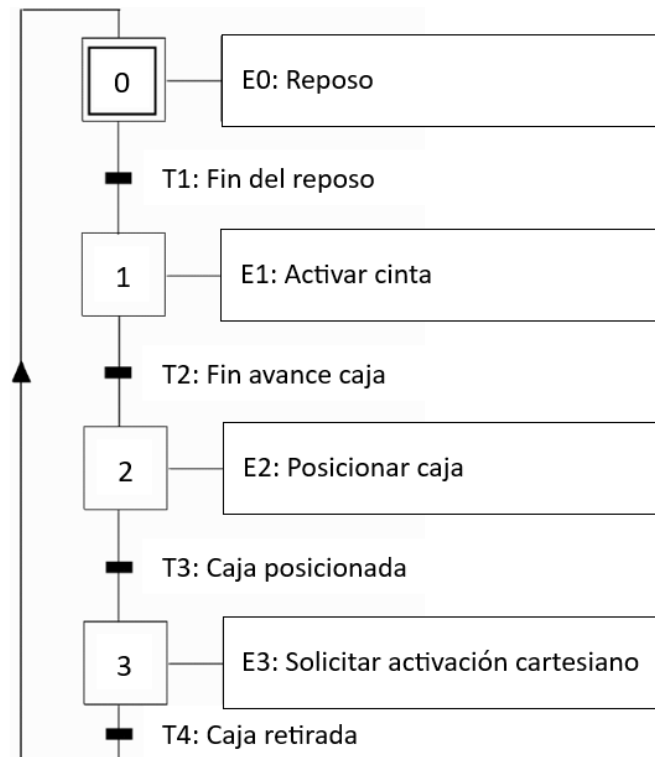


Figura 104. GRAFCET Cinta cajas medianas

### GRAFSET robot cartesiano

Refleja los diferentes estados del robot cartesiano en función de las cajas que llegan por la cinta transportadora de cajas medianas, las cajas ya colocadas, el pallet donde las coloca y la posición del extremo del robot.

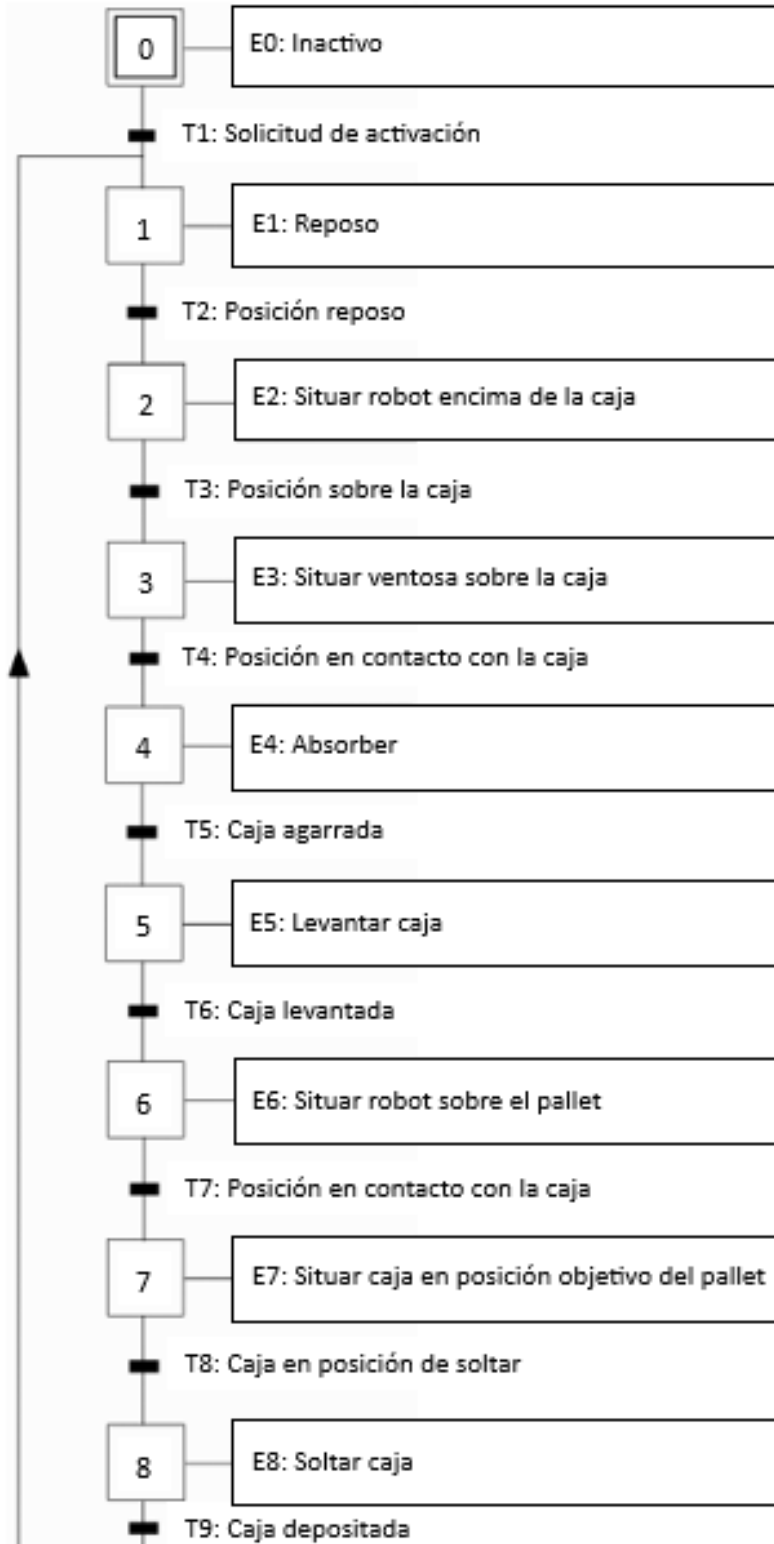


Figura 105. GRAFSET Robot cartesiano

### GRAF CET robot de dos ejes

Refleja los diferentes estados del robot de dos ejes en función de las cajas que llegan por la cinta transportadora de cajas pequeñas, las cajas ya colocadas, el cajón del pallet donde las coloca y la posición del extremo del robot.

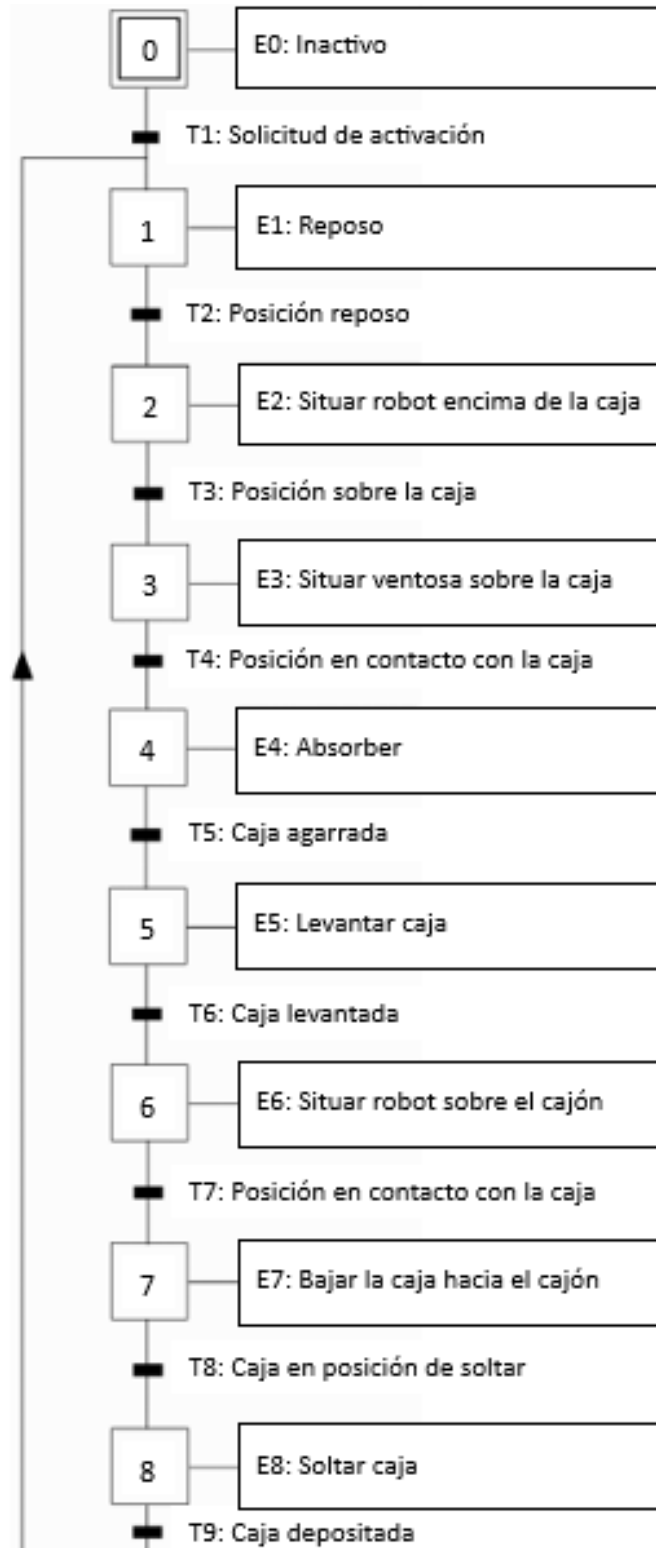


Figura 106. GRAFCET Robot de dos ejes

### GRAFCET transportador de rodillos

Describe la activación y desactivación del transportador de rodillos en función de los sensores colocados en ese transportador y las cajas colocadas en el pallet que transporta.

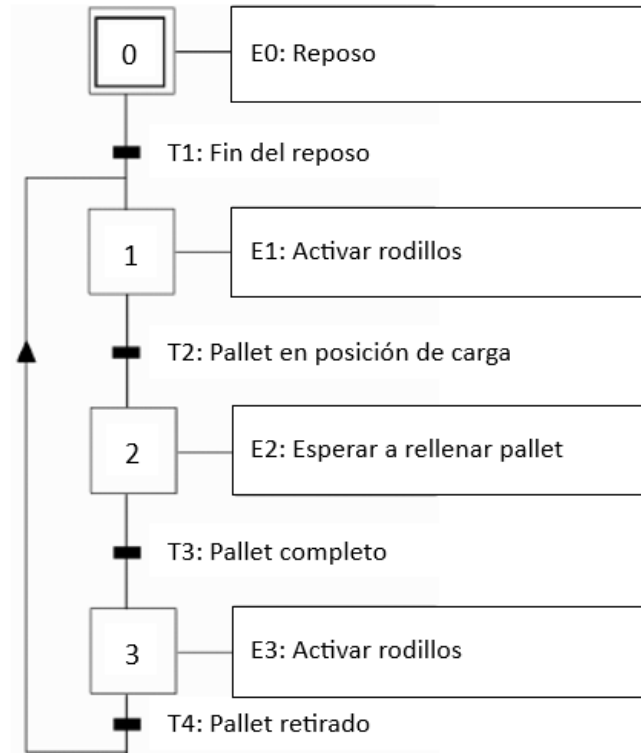


Figura 107. GRAFCET Transportador de rodillos

Mediante la combinación de estos GRAFCET's, y desarrollando de forma completa cada proceso, seremos capaces de estructurar gran parte del programa del PLC. El resto de los procesos no son estrictamente secuenciales, no requiriendo de forma obligatoria estructuras GRAFCET para estructurarlos.

#### 5.4.4 Programación del autómatas

La programación del autómatas se realiza en lenguaje KOP ajustándose a los GRAFCET's descritos anteriormente.

Como se comentó en el objetivo de esta simulación, es un programa complejo, con un gran número de entradas y salidas como se puede apreciar en la siguiente tabla. Además de estos dos tipos de variables, se han empleado DB's globales y de instancia para las FB's asociadas. Con el objetivo de no ampliar más la tabla de variables estándar, no se emplean marcas, ya que su función se puede cumplir con variables internas almacenadas en DB's globales.

Como en algunas de las estaciones anteriores, no se mostrarán mediante tablas las variables contenidas en DB's, solo las contenidas en el PLC, siendo el conjunto de ellas las reflejadas a continuación:

Variables PLC				
	Nombre	Tabla de variables	Tipo de datos	Dirección ▲
1	Marcha	Tabla de variables e.	Bool	%I10.0
2	Paro	Tabla de variables e.	Bool	%I10.1
3	Emergencia	Tabla de variables e.	Bool	%I10.2
4	Reset_Contadores_Cintas	Tabla de variables e.	Bool	%I10.7
5	Sensor_Caja	Tabla de variables e.	Bool	%I11.0
6	Reset_Contadores	Tabla de variables e.	Bool	%I11.1
7	Sensor_Inicio_Bascula	Tabla de variables e.	Bool	%I11.2
8	Sensor_Posicion_Bascula	Tabla de variables e.	Bool	%I11.3
9	Sensor_Final_Bascula	Tabla de variables e.	Bool	%I11.4
10	Sensor_Inicio_Cinta_Izquierda	Tabla de variables e.	Bool	%I11.5
11	Sensor_Final_Cinta_Izquierda	Tabla de variables e.	Bool	%I11.6
12	Sensor_Inicio_Cinta_Derecha	Tabla de variables e.	Bool	%I11.7
13	Sensor_Final_Cinta_Derecha	Tabla de variables e.	Bool	%I12.0
14	Sensor_Inicio_Cinta_Delanteras	Tabla de variables e.	Bool	%I12.1
15	Sensor_Final_Cinta_Delanteras	Tabla de variables e.	Bool	%I12.2
16	Sensor_Piston	Tabla de variables e.	Bool	%I12.3
17	Sensor_Inicio_Rodillos_Izquierda	Tabla de variables e.	Bool	%I12.4
18	Sensor_Final_Rodillos_Izquierda	Tabla de variables e.	Bool	%I12.5
19	Sensor_Pallet_Izquierdo_Esperando	Tabla de variables e.	Bool	%I12.6
20	Sensor_Inicio_Rodillos_Derecha	Tabla de variables e.	Bool	%I12.7
21	Sensor_Final_Rodillos_Derecha	Tabla de variables e.	Bool	%I13.0
22	Sensor_Pallet_Derecho_Esperando	Tabla de variables e.	Bool	%I13.1
23	Peso	Tabla de variables e.	Int	%W30
24	Cartesiano_Posicion_X	Tabla de variables e.	Real	%D34
25	Cartesiano_Posicion_Y	Tabla de variables e.	Real	%D38
26	Cartesiano_Posicion_Z	Tabla de variables e.	Real	%D42
27	RobotXZ_Posicion_X	Tabla de variables e.	Real	%D46
28	RobotXZ_Posicion_Z	Tabla de variables e.	Real	%D50
29	Luz_Marcha	Tabla de variables e.	Bool	%Q0.0
30	Luz_Paro	Tabla de variables e.	Bool	%Q0.1
31	Motor_Cinta_Entrada	Tabla de variables e.	Bool	%Q1.0
32	Motor_Bascula	Tabla de variables e.	Bool	%Q1.1
33	Motor_Distribuidora_Izquierda	Tabla de variables e.	Bool	%Q1.2
34	Motor_Distribuidora_Derecha	Tabla de variables e.	Bool	%Q1.3
35	Motor_Distribuidora_Delante	Tabla de variables e.	Bool	%Q1.4
36	Motor_Cinta_Izquierda	Tabla de variables e.	Bool	%Q1.5
37	Motor_Cinta_Derecha	Tabla de variables e.	Bool	%Q1.6
38	Motor_Cinta_Delanteras	Tabla de variables e.	Bool	%Q1.7
39	Neumatica_Piston	Tabla de variables e.	Bool	%Q2.0
40	Succion_Cartesiano	Tabla de variables e.	Bool	%Q2.1
41	Giro_Pinza_Cartesiano	Tabla de variables e.	Bool	%Q2.2
42	Motor_Rodillos_Izquierda	Tabla de variables e.	Bool	%Q2.3
43	Rodillo_Parada_Izquierda	Tabla de variables e.	Bool	%Q2.4
44	Succion_RobotXZ	Tabla de variables e.	Bool	%Q2.5
45	Motor_Rodillos_Derecha	Tabla de variables e.	Bool	%Q2.6
46	Rodillo_Parada_Derecha	Tabla de variables e.	Bool	%Q2.7
47	Display_Peso_Actual	Tabla de variables e.	Real	%QD20
48	Display_Pallets_Pequeños	Tabla de variables e.	DInt	%QD24
49	Display_Pallets_Medianos	Tabla de variables e.	DInt	%QD28
50	Display_Cajas_Grandes	Tabla de variables e.	DInt	%QD32
51	Cartesiano_Desplazamiento_X	Tabla de variables e.	Real	%QD36
52	Cartesiano_Desplazamiento_Y	Tabla de variables e.	Real	%QD40
53	Cartesiano_Desplazamiento_Z	Tabla de variables e.	Real	%QD44
54	RobotXZ_Desplazamiento_X	Tabla de variables e.	Real	%QD48
55	RobotXZ_Desplazamiento_Z	Tabla de variables e.	Real	%QD52

Tabla 8. Variables del PLC. Estación colocador por peso

Como en el resto de estaciones, las variables de entrada y salida son las empleadas en el programa de simulación del entorno virtual de Factory IO, compartiendo las mismas direcciones para el correcto funcionamiento de la simulación.

Los bloques de programa son muy numerosos en esta estación, es por este motivo, por el cual se ha decidido incluir al inicio del nombre del bloque un número en función de la programación que contenga el bloque para que sea más rápido y visual buscarlo en el árbol de proyecto. En el caso de los bloques que contengan una programación genérica para toda la estación (ya sean DB's, FC's o FB's), se añade el número 0. En el caso de que estén relacionados solo con cajas pequeñas se añade el número 1, en el caso de cajas medianas el 2 y en caso de las cajas grandes el 3. De esta manera la estructura del programa es la siguiente:

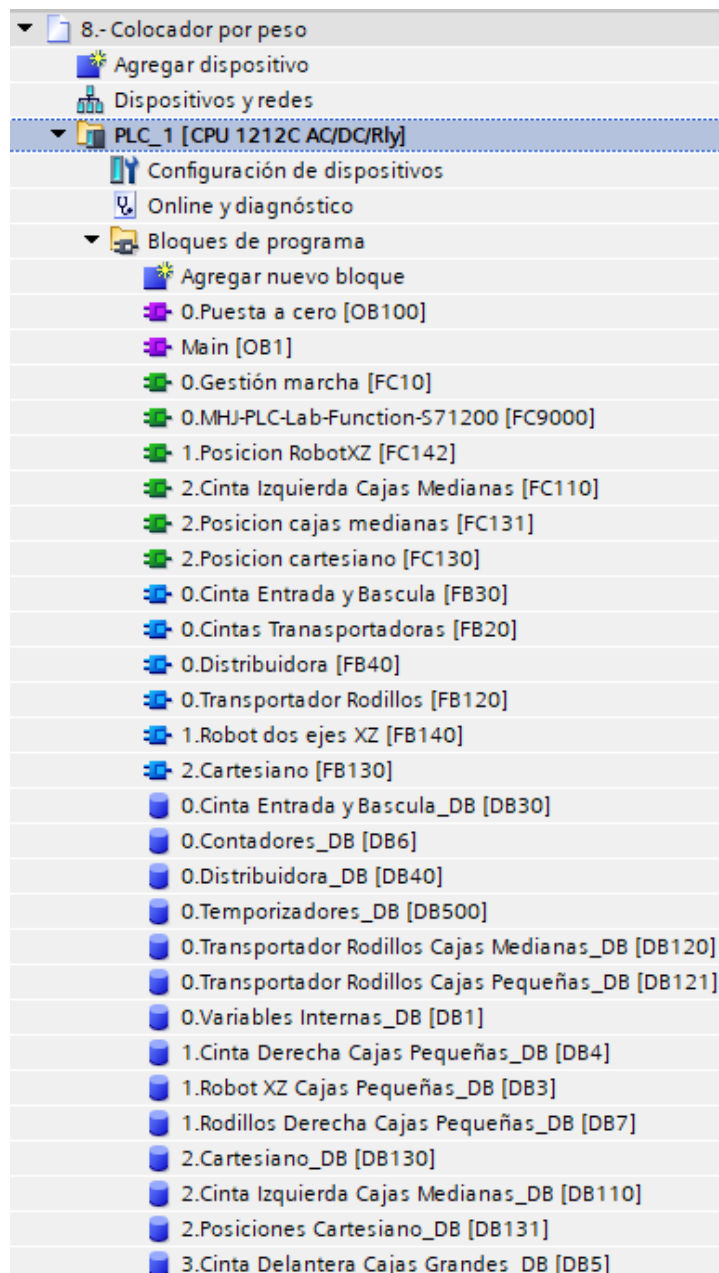


Figura 108. Distribución de bloques. Estación colocador por peso

Como se observa en la figura anterior, hay dos bloques de organización de programa:

**Puesta a cero [OB100]:** es la OB de arranque encargada de resetear todas las variables de salida y preparar la estación para que cada vez que el PLC pase a RUN, la maquina se encuentre en una situación segura.

**Main [OB1]:** contiene el resto de la programación, encargándose de llamar al resto de bloques de programa, realizar las operaciones necesarias con las salidas de estos bloques y aumentar los contadores de cajas grandes y pallets llenos de cajas pequeñas y medianas procesados por la estación.

Los bloques contenidos en el main y la descripción de la función que desempeñan se analizan a continuación:

Como en la estación anterior, se emplea la **FC9000** para conectar de forma correcta el PLC con Factory IO y la **Gestión de marcha [FC10]** para gestionar la marcha y paro de la estación según lo explicado en la descripción. Al ser los mismos bloques de programa, no se muestran unas figuras para representarlos ya que serían iguales a las *figuras 94 y 95 del apartado 5.3.4.*

**0.Cinta de entrada y báscula [FB30]:** contiene la programación encargada de gestionar la cinta por donde entran las cajas y la báscula transportadora; que engloba tanto la cinta que permite que las cajas avancen por la báscula hasta el distribuidor, como el proceso de pesaje y asignación de un tipo a cada caja en función de su peso, mostrando este último por pantalla. Este bloque se realiza íntegramente para esta estación, no pudiéndose aprovechar un bloque parecido de otras estaciones, puesto que contiene la programación de un **elemento nuevo, la báscula.**

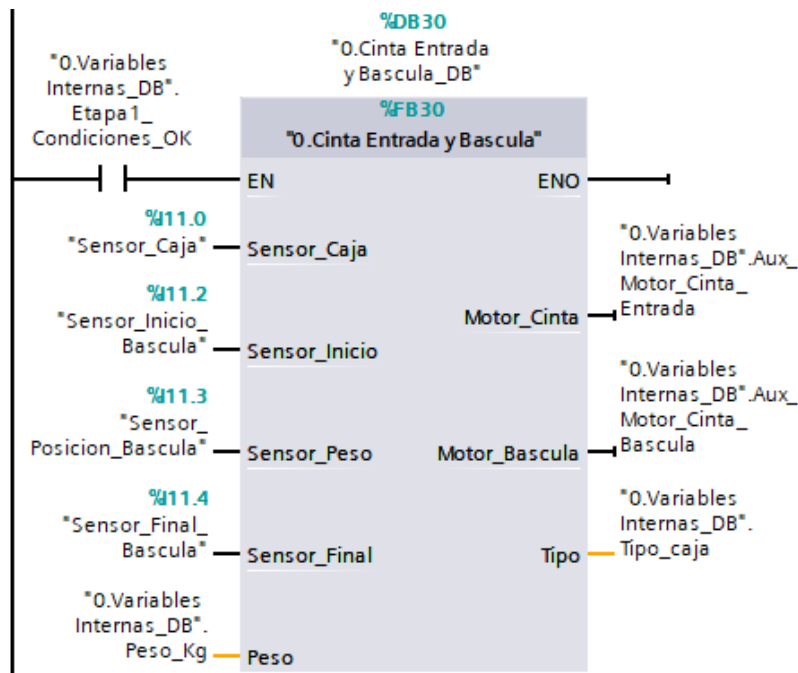


Figura 109. FB30 0.Cinta entrada y báscula. Estación colocador por peso

**0.Distribuidora [FB40]:** contiene la programación encargada del correcto funcionamiento del clasificador de ruegos emergentes, o distribuidor de ruedas emergentes. Al final de la báscula, se encuentra un sensor, que, al pasar las cajas por él, activa los rodillos empujándola a izquierda, derecha o de frente, según sea pequeña, mediana o grande respectivamente. Y en cada salida tiene ubicado un sensor (variables de entrada del bloque, junto con el tipo de caja) que se encarga de activar la cinta transportadora correspondiente a cada tipo de caja.

Aunque ya se había empleado este elemento en otra estación, el funcionamiento es notablemente diferente, por lo que se ha programado el bloque desde cero para cumplir con los requerimientos de esta estación.

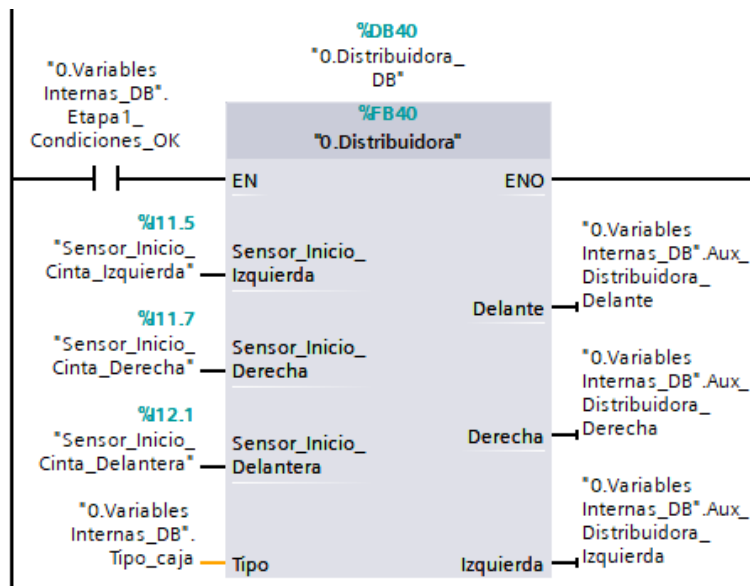


Figura 110. FB40 0.Distribuidora. Estación colocador por peso

**2.Cinta izquierda cajas medianas [FC110]:** contiene la programación estructurada según su GRAFCET, controlando la activación y desactivación de la cinta izquierda, el elemento posicionador y solicitando la activación del robot cartesiano. Para ello, cuenta con una serie de entradas, constituidas por los sensores ubicados en la cinta y en el elemento posicionador y una serie de salidas que se emplean para controlar dicha cinta, el elemento posicionador y solicitar la activación del cartesiano.

Como ya se ha comentado al inicio de la estación, se trata de un proceso industrial bastante extenso. Es por esta razón, por la cual se ha diseñado la estación industrial para hacer uso de los bloques de programas de diversas estaciones anteriores, adaptarlos y emplearlos, reflejando así la **característica reutilizable del código** y los **beneficios de estructurarlo en bloques** de programa. Como es lógico, no basta con emplear los bloques directamente, sino que se tomaran estos como partida para hacer las modificaciones pertinentes.

Toda la rama izquierda, correspondiente a las cajas medianas, es muy similar a la estación paletizadora, por lo que muchos de los bloques de programa de esta

estación parten de algunos de los bloques de la estación ya creada. Para la programación de este bloque, se ha partido del bloque “*transportador cinta [FC110]*” de la estación paletizadora, remplazando las variables de tipo marca por variables de un DB’s empleado solo por esta FC y las marcas de entrada y salida, por variables de un DB global.

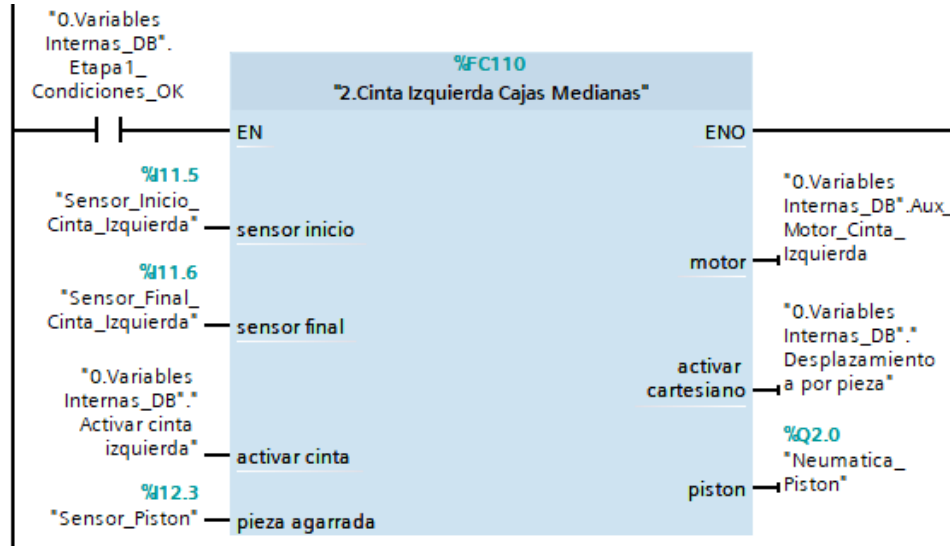


Figura 111. FC110 2.Cinta izquierda cajas medianas. Estación colocador por peso

**0.Cintas transportadoras [FB20]:** Gestiona la activación o desactivación del motor de la cinta y aumenta el contador de cajas que se encuentran sobre la cinta, en función de los sensores situados a su entra y salida y del botón de “*reset contadores cintas*”. Este bloque se emplea en las otras dos cintas (tanto la delantera como la derecha) puesto que ambas tienen el mismo funcionamiento, asignándoles un DB distinto en cada una de sus llamadas. Aunque este bloque se podía haber programado siguiendo una estructura GRAFCET, su programación era tan simple, que era más clara sin aplicar esta estructura.

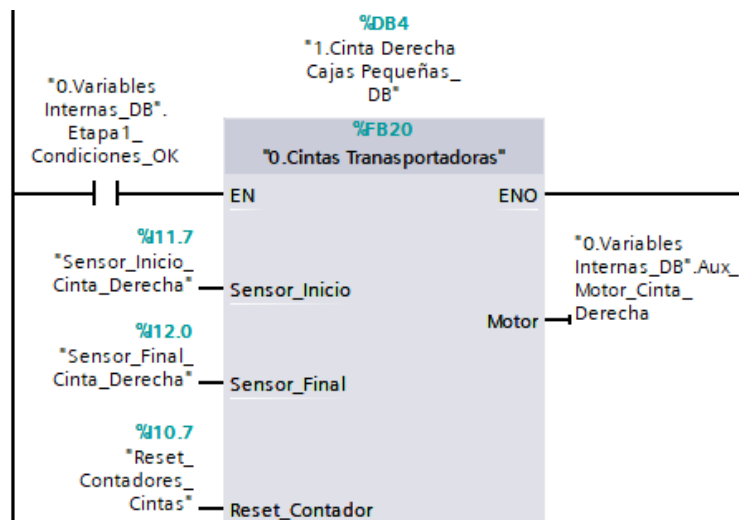


Figura 112. FB20 0.Cintas transportadoras. Estación colocador por peso

**0.Transportador rodillos [FB120]:** contiene el GRAFCET, programado en lenguaje KOP, que estructura el funcionamiento del transportador de rodillos. Este proceso secuencial, controla las condiciones de activación y desactivación de los rodillos y del rodillo de parada situado a mitad del transportador, en función de los sensores situados sobre el transportador y del número de cajas posicionadas sobre el pallet.

Para la programación de este bloque, se ha partido del bloque “transportador rodillos [FC120]” de la estación paletizadora. Es necesario resaltar que el bloque original es una FC y el bloque empleado es un FB. Como los dos transportadores de rodillos que posee la estación tienen el mismo funcionamiento, se emplea el mismo bloque FB para el control de ambos. Se podía haber mantenido la FC y crear una DB para cada uno de los transportadores de rodillos, empleando sus variables en nuevas entradas y salidas (caso parecido al de la FC110 de esta estación). Sin embargo, se ha decidido crear una FB nueva donde se han copiado y pegado los segmentos de la FC antigua, asignando a la FB un **DB de instancia distinto en cada llamada**. De esta forma queda de manifiesto una nueva manera de reutilizar código, **clarificando las diferencias entre una FB y una FC**.

Al igual que en el caso de la FC110 se reemplazan las variables de tipo marca por variables del DB de instancia correspondiente y las marcas de salida, por variables de un DB global.

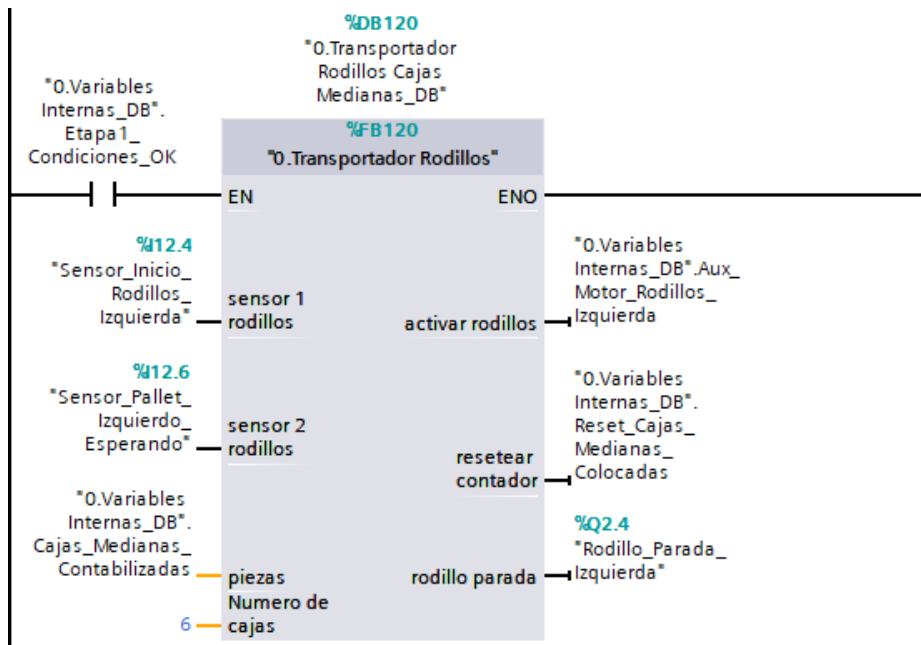


Figura 113. FB120 0.Transportador rodillos cajas medianas. Estación colocador por peso

**2.Cartesiano [FB130]:** Como su propio nombre indica, contiene la programación referente al control del robot cartesiano estructurada en forma de GRAFCET como ya se explicó anteriormente. Para ello tiene asociadas un conjunto de entradas y salidas constituidas por los sensores de posición del extremo del cartesiano, el número de cajas emplazadas en el pallet, las tensiones de salida para controlar el robot y para volver a solicitar la activación de la cinta transportadora izquierda.

Como en los otros dos casos anteriores donde se emplean bloques de programa para conjuntos de elementos de la rama de las cajas medianas, este bloque se ha recuperado de la estación paletizadora (“cartesiano [FB130]”) sustituyendo las marcas por variables de diversos DB’s.

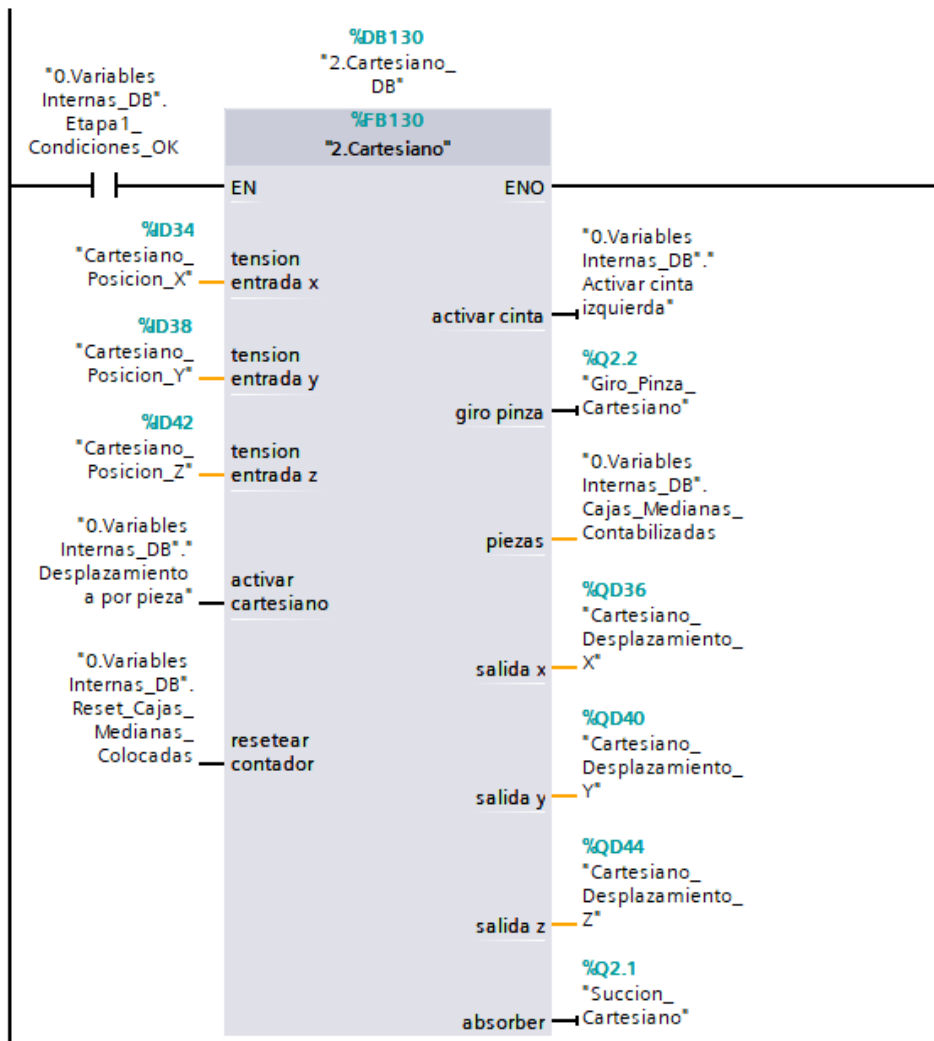


Figura 114. FB130 2.Cartesiano. Estación colocador por peso

Desde el interior de este bloque, se llama a otras dos FC's relacionadas también con el funcionamiento del robot cartesiano. La primera se encarga de trabajar con las lecturas del posicionamiento actual del robot y la segunda se ocupa de indicar la posición del pallet donde se debe colocar la caja que acaba de recoger el robot.

**2.Posición cartesiano [FC130]:** esta FC se llama desde la FB130 “2.Cartesiano” y se encarga de tratar los datos que se reciben de los sensores de posición del robot, para que se puedan comparar posteriormente con las posiciones donde se deben depositar las cajas. Este bloque se ha copiado y pegado de la estación paletizadora (“posición cartesiano [FC130]”), ya que al ser un proceso de escalado y redondeo, no es necesario realizar ninguna modificación sobre él.

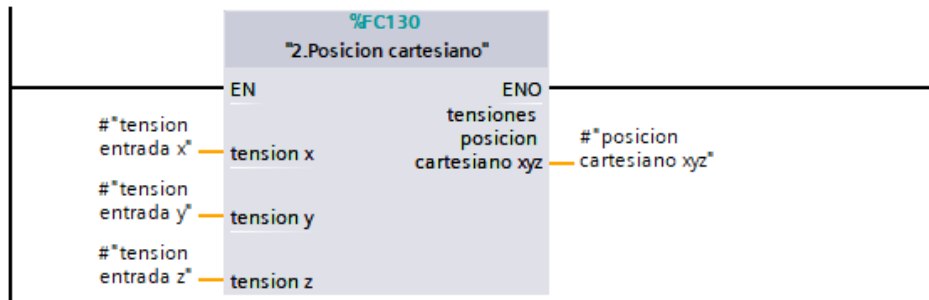


Figura 115. FC130 2.Posición cartesiano. Estación colocador por peso

**2.Posición cajas medianas [FC131]:** al igual que el bloque anterior, este también se llama desde la FB130 “2.Cartesiano”, e indica al robot cuál es la posición de la siguiente caja dentro del pallet, dependiendo del número de cajas ya colocadas.

Este bloque también ha sido copiado y pegado desde el programa de la estación paletizadora (“posición pieza [FC130]”), sin embargo, sí que ha sido necesario adaptarlo. Debido al cambio de escena dentro de Factory IO, el robot y el pallet no están situados exactamente igual que en la estación paletizadora, por lo que ha sido necesario volver a tomar todos los puntos del pallet y de la cinta transportadora de forma manual y sustituirlos en la variable array del bloque. A nivel de programación no se ha modificado nada más que estos valores.

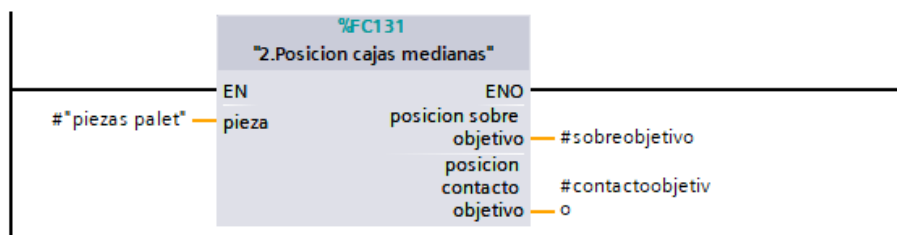


Figura 116. FC131 2.Posición cajas medianas. Estación colocador por peso

Con la combinación de estos tres últimos bloques de programa, se logra el control absoluto del robot, cumplimentando los requerimientos que a este se le piden.

**1.Robot dos ejes XZ [FB140]:** ajustándose a su GRAFCET, gestiona el control del robot de dos ejes, encargado de colocar las cajas pequeñas en el cajón apilable. Este conjunto de programación parte del bloque “robot automático [FB140]” de la estación distribuidora. Para que cumpliera con su nueva función, ha sido necesario cambiar alguna de sus entradas y eliminar varias de sus salidas. Además, las posiciones de recogida y colocación de las cajas también se han debido sustituir, tomando los valores de forma manual en la escena de Factory IO.

A diferencia del robot cartesiano, este robot recoge y suelta las cajas siempre en las mismas posiciones. Es por esta razón, por la cual no se emplea una variable array para almacenar las posiciones donde se colocan las cajas, sino que se integra esta operación con una instrucción “MOVE” dentro de la programación GRAFCET.

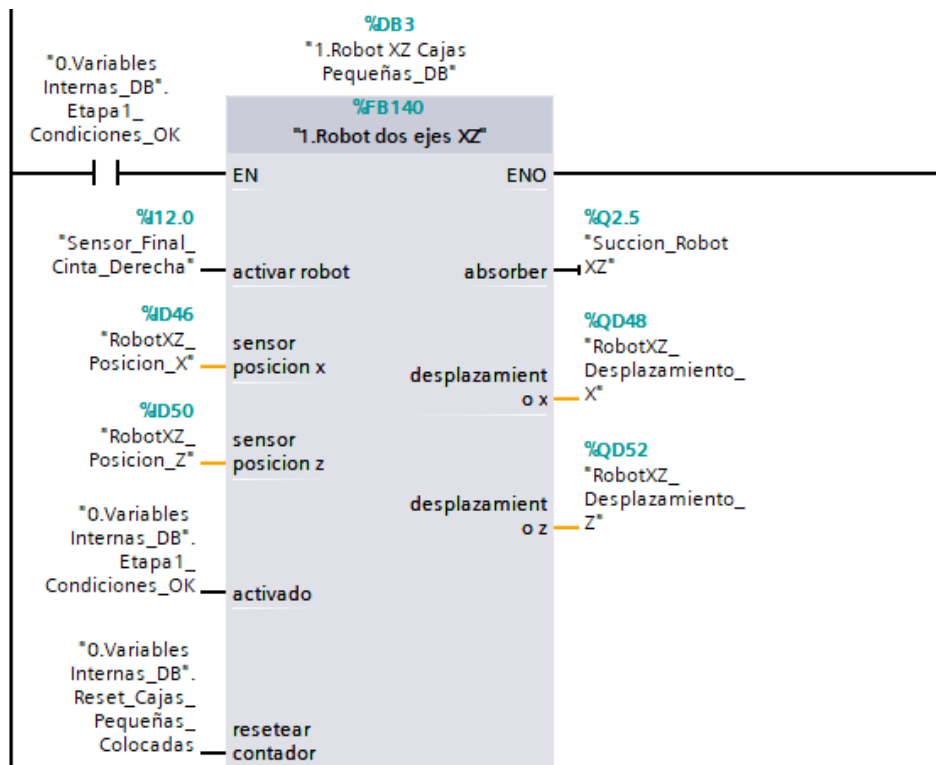


Figura 117. FB140 1.Robot dos ejes XZ. Estación colocador por peso

Originariamente, el bloque de la estación distribuidora llamaba a dos FC's en su interior. La primera se ocupaba de contabilizar las piezas ya colocadas dentro del cajón y la segunda se encargaba de trabajar con las lecturas del posicionamiento actual del robot. Teniendo en cuenta la nueva distribución de bloques que se realiza en la estación del colocador por peso y el GRAFCET del robot de dos ejes, se consideró más visual eliminar la primera FC e incluir su programación directamente en el bloque F140. Sin embargo, la segunda FC sí que se mantendrá, describiendo a continuación sus funcionalidades.

**1.Posición RobotXZ [FC142]:** este bloque se llama desde la FB140 “1.Robot dos ejes XZ”, encargándose de tratar los datos que se reciben de los sensores de posición del robot, para que se puedan comparar posteriormente con las posiciones donde se deben depositar las cajas. Este bloque se ha copiado y pegado de la estación distribuidora (“posición robot [FC142]”), ya que al ser un proceso de escalado y redondeo, no es necesario realizar ninguna modificación sobre él, como pasaba con el bloque de “2.Posición cartesiano”.

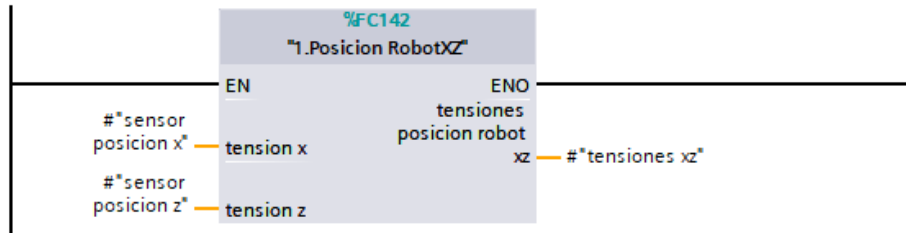


Figura 118. FC142 1.Posición robotXZ. Estación colocador por peso

Después de la programación completa de esta estación, queda de manifiesto la necesidad de comprender la diferencia entre FC y FB, el carácter reutilizado del código y la importancia de la programación en bloques. También se demuestra que cuantas menos marcas se usen, empleando variables de DB's globales en su lugar, más fácil y menos modificaciones se tendrán que realizar si se reutiliza código.

Por otro lado, en instalaciones de estas dimensiones, la importancia de una buena estructuración del código y asignación de nombres autoexplicativos y con unas ciertas reglas nemotécnicas, son buenas técnicas de programación debido a su alta complejidad y gran número de entradas y salidas.

Después de este análisis, se concluye que es necesario conocer y aplicar correctamente los conocimientos marcados en los objetivos que adquirir al comienzo de la descripción de la estación. Por lo tanto, se podría justificar el cumplimiento de la finalidad didáctica de esta práctica.

## 5.5. ESTACIÓN 9: PRODUCCIÓN EN LÍNEA CON SEGURIDAD

### 5.5.1 Objetivo de la simulación

Se trata de una estación cuya programación del PLC tiene un nivel de dificultad medio-alto, sin embargo, la dificultad de la programación del HMI es alta. El principal objetivo de esta estación es que el usuario aprenda herramientas relacionadas con la seguridad, tanto a nivel de programación del autómatas como del panel de operador. A diferencia de la estación de la calculadora, esta estación está dotada de un gemelo digital de la estación real en Factory IO, donde se usan recintos cerrados con robots y centros de mecanizado. La programación relacionada con estos recintos es donde se aplican los conocimientos de seguridad en la programación del PLC y la gestión de usuarios y permisos del HMI.



- **Conocimientos que adquirir en TIA Portal:**
  - **Comandos:** Temporizador TONR.
  - **Programa:** Protección Know-how de los bloques de programa.
  - **Conectividad:** Tipo de PLC, dirección IP, conectividad con PLCSIM Advanced y Factory IO.
- **Conocimientos que adquirir en la programación de un HMI con TIA Portal:**
  - **Componentes:** Visión de usuarios y permisos de seguridad de los componentes.
  - **Variables:** Su conexión con el PLC y colocación en diferentes tablas.
  - **Programa:** Administración de usuarios, pantallas emergentes y planificador de tareas.
  - **Conectividad:** Del HMI con el PLC y simulación en WinCC.
- **Conocimientos que adquirir en Factory IO:**
  - **Componentes:** Centros de mecanizado.
  - **Conectividad:** Con PLCSIM Advanced.

### 5.5.2 Descripción de la escena

La estación se encarga de producir tapas o bases a partir de materias primas mediante dos centros de mecanizado. Cada centro emplea un robot de seis ejes para introducir la materia prima y sacar las piezas del centro, colocándolas en unas cintas transportadoras. Las cintas de bases y de tapas se unen en una misma línea más adelante, por lo que se necesita realizar una gestión de prioridad. Todo este proceso está automatizado y controlado a través de un PLC y un HMI.

La estación está compuesta por un conjunto de cintas transportadoras y dos centros de tratamiento limitados con vallas, que contienen cada uno un centro de mecanizado y robot de seis ejes. Los centros de tratamiento cuentan con una puerta para poder acceder a los recintos cerrados. Esta puerta debe permanecer cerrada en todo momento para el correcto funcionamiento automático de la estación.

A la entrada de cada recinto vallado se encuentra una cinta transportadora por donde entran los bloques de materia prima. La cinta transportadora de entrada avanza hasta introducir la materia prima en el recinto cerrado.

Si el centro de tratamiento está activado y el sensor de visión colocado a la entrada detecta que ha entrado materia prima, el robot de seis ejes recoge la materia prima, y la introduce en el centro de mecanizado. Este se cierra y después de un periodo de tiempo el centro de mecanizado se vuelve a abrir. Según se haya configurado el elemento de Factory IO que simula al centro de mecanizado, se producirán tapas o bases y por lo tanto los tiempos de procesado son distintos.

Una vez finalizado el proceso de mecanizado del centro, el robot recoge la pieza correspondiente y la deposita en una rampa por la que sale del recinto vallado a un conjunto de dos cintas transportadoras en forma de “L”. Las cintas de salida de ambos recintos convergen en una misma línea de salida, por lo que es necesario realizar una gestión de prioridad entre las bases que salen de un centro y las tapas que salen del otro para que no colisionen en la cinta de salida. Esta prioridad dependerá de qué tipo de pieza salga antes del recinto, manteniendo prioridad frente al otro tipo durante dos segundos. Las cintas de salida en forma de “L” se activan durante un periodo de 11 segundos si tienen prioridad y se detecta que ha salido una pieza del recinto. Después de este periodo de tiempo, las piezas ya han salido de estas cintas, pasando a la línea de convergencia de ambas ramas. Esta cinta transportadora se encarga de transportar ambos tipos de piezas fuera de la estación.

Con el objetivo de visualizar mejor la estación industrial descrita, se muestra continuación la escena que se emplea en la simulación.



*Figura 119. Vista general. Escena producción en línea*

Para controlar la estación se dispone de un cuadro eléctrico principal, un HMI y dos cuadros eléctricos encargados únicamente de gestionar cada uno de los centros de tratamiento donde están ubicados.

Los cuadros eléctricos de los centros de tratamiento contienen una seta de emergencia, un pulsador de marcha, uno de paro, un display que muestra el número de piezas producidas por el centro y un pulsador de reset para reiniciar el contador de piezas que se muestra por el display. Las setas y estos pulsadores solo controlan la activación o desactivación del centro de tratamiento que controlan, no el de toda la estación industrial. La gestión de activación y desactivación es igual que la gestión de la marcha y paro de las estaciones anteriores. En caso de que el centro esté activado su control de marcha y de paro depende del estado de la estación completa. Por el contrario, si el centro está desactivado, es necesario activarlo desde el cuadro eléctrico o desde el HMI (con las variables conectadas con los pulsadores) para que se vuelva a activar, aunque la estación se encuentre en estado de marcha.



Figura 120. Vista cuadro eléctrico centro tapas. Escena producción en línea

El cuadro eléctrico principal está compuesto por una seta de emergencia, un pulsador de marcha con iluminación, un pulsador de paro con iluminación, un selector de funcionamiento de los centros de tratamiento, que puede variar entre automático y manual, y una luz que indica que alguno de los centros está funcionando bajo un forzado manual. En este caso, tanto la seta de emergencia como los pulsadores controlan el estado de marcha y paro de toda la estación industrial, incluidos los centros de tratamiento.

Si se pulsa el botón de marcha, la estación entra en estado activo, esperando a que llegue la materia prima y funcionando de forma totalmente automática, indicándose con una luz verde en el propio pulsador de marcha. Si se pulsa el botón de paro, independientemente de la señal de los sensores, la estación entra en estado de reposo, indicándose con una luz roja en el propio botón de paro. Para reanudar el estado de marcha basta con volver a pulsar el botón de marcha.

En caso de emergencia, si se pulsa la seta de emergencia la estación se detiene. Al ser una seta de emergencia, será necesario girar la seta para quitar el enclavamiento y pulsar el botón de marcha. Si no se realiza esta operación en ese orden, no se reanudará la marcha para evitar que se produzca una situación potencialmente peligrosa para el operario o para la instalación.



Figura 121. Vista cuadro eléctrico principal. Escena producción en línea

Como se ha mencionado con anterioridad, esta estación contiene un panel de operador (o HMI) para la supervisión y control de la estación completa. Antes de empezar a explicar el contenido y funcionamiento de sus imágenes, es importante mencionar un factor fundamental del diseño de esta estación. La estación industrial está diseñada para ser controlada por dos tipos de personal, operarios y jefes de planta, cada uno con unos permisos que les posibilitan realizar unas acciones concretas.

**Los operarios** podrán controlar toda la maquina en funcionamiento automático, siempre y cuando las puertas de los recintos que contienen los centros de tratamiento permanezcan cerradas.

**Los jefes de planta**, además de realizar las mismas funciones que los operarios, son capaces de realizar un forzado en funcionamiento manual de los centros de tratamiento a través del HMI, así como rearmar la seguridad si una puerta se abre. Esta opción se ha incluido, para simular labores de mantenimiento o en los casos de que se abra una puerta, que sea un jefe de planta el que revise que no se encuentra ningún operario dentro del área de trabajo de los robots y sea él el que rearme la seguridad y permita reanudar el funcionamiento automático de la estación de nuevo.

El HMI consta de tres imágenes que permiten visualizar y modificar el estado de la estación completa y de cada uno de los centros de tratamiento. Además de mostrar el estado de la estación, se puede modificar el valor de las variables asociadas a los tres cuadros eléctricos mediante el uso de botones en la pantalla.

En la pantalla de inicio, o imagen raíz, se visualiza el estado de ambos centros, de las cintas transportadoras, del funcionamiento manual o automático de la estación, la posibilidad de abrir o cerrar sesión del usuario que está controlando el HMI, el usuario registrado, la fecha y la hora. Además de mostrar toda esta información, permite variar el valor de marcha y paro de la estación completa y de cada centro si se tiene el permiso de “operación”.

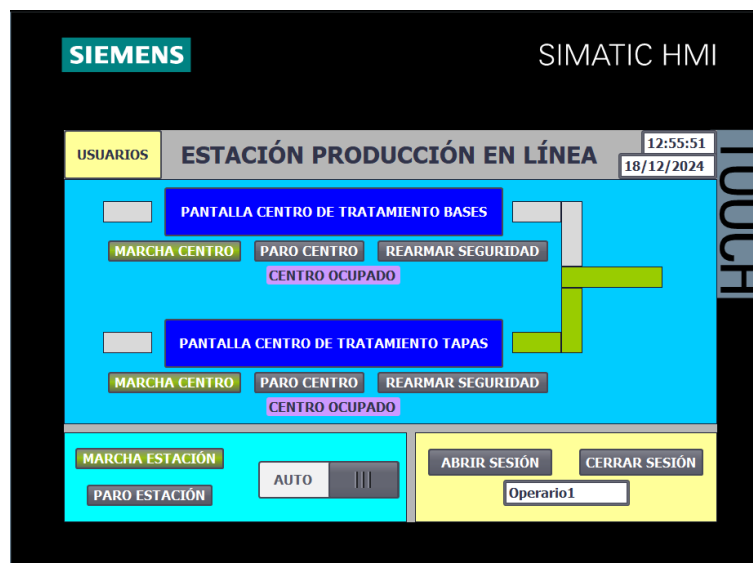


Figura 122. Imagen raíz. Estación producción en línea

Por último, esta pantalla da acceso a las pantallas de cada centro de tratamiento y a una pantalla emergente, si se tiene el permiso de “administración de usuarios”, donde se muestra el registro de usuarios conectados, pudiéndose modificar la contraseña de los dos tipos de usuarios del HMI.



Figura 123. Imagen emergente Administración de usuarios. Estación producción en línea

Las otras dos imágenes del HMI son idénticas, pero mostrando cada una la información y los comandos de control de cada uno de sus centros.

Estas imágenes están divididas en cuatro zonas. La primera, con fondo gris, contiene el título de la imagen, la fecha, la hora y un botón para volver a la imagen raíz. La segunda zona, con un fondo azul oscuro, contiene los controles de la estación y los del centro, además de mostrar diversos valores referentes al centro de tratamiento. La tercera, con fondo morado, muestra la gestión de seguridad. Y la última zona, con fondo amarillo, permite abrir y cerrar sesión de los usuarios del HMI, mostrando el usuario registrado en cada momento.

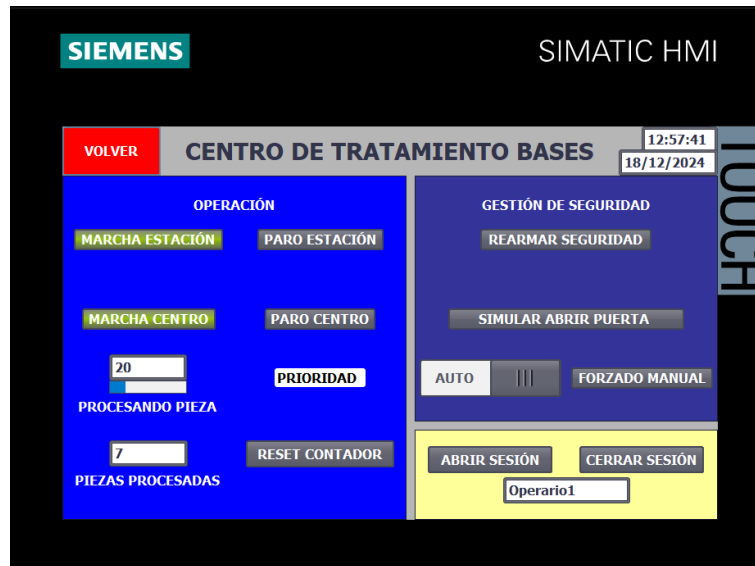


Figura 124. Imagen CENTRO BASES. Estación producción en línea

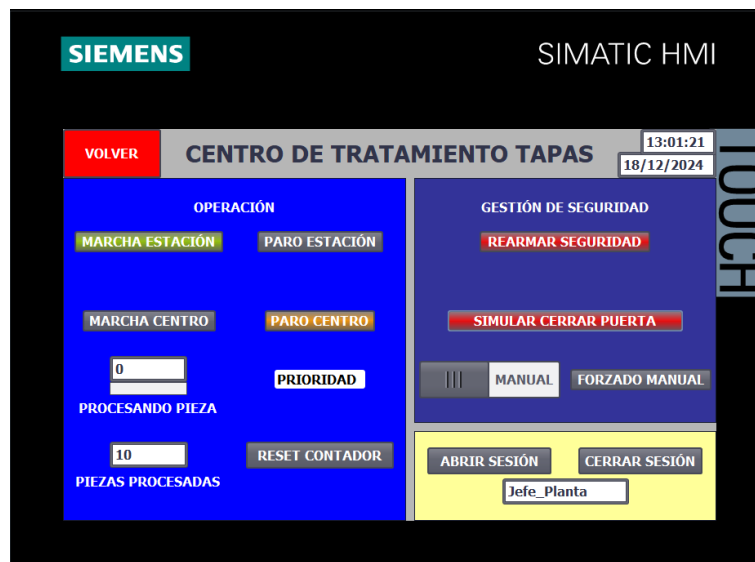


Figura 125. Imagen CENTRO TAPAS. Estación producción en línea

Los botones de todas las pantallas tienen unos permisos asociados que requieren que un usuario con ese permiso se registre para permitir interactuar con ellos. Los únicos botones que no requieren de ningún usuario registrado son los de navegación entre imágenes del HMI.

Habiendo analizado el comportamiento de la estación y de los elementos de control de esta, podemos profundizar en las diferencias entre el control automático y manual de los centros. La estación está diseñada para trabajar en funcionamiento automático por defecto, permitiendo al operario controlar toda la estación. En el caso de cambiar el selector del panel de control principal a manual, para que los centros funcionen es necesario que el jefe de planta mantenga pulsado el botón de “forzado manual” del centro, sin importar si la puerta está cerrada o no o la seguridad está rearmada o no.

Para completar la simulación del entorno virtual, es necesario asociar las variables que se deseen emplear en la programación del PLC como entradas o salidas con una dirección del driver de Factory IO. El resultado de la distribución es el siguiente:

Marcha	%I10.0	%Q0.0	Luz Marcha
Paro	%I10.1	%Q0.1	Luz paro
Emergencia	%I10.2	%Q0.2	Luz de Forzado Manual
Funcionamiento Manual	%I10.3	%Q0.3	
Funcionamiento Automático	%I10.4	%Q0.4	
	%I10.5	%Q0.5	
	%I10.6	%Q0.6	
	%I10.7	%Q0.7	
Centro de Tapas Abierto	%I11.0	%Q1.0	Marcha Centro Tapas
Centro de Bases Abierto	%I11.1	%Q1.1	Marcha Centro Bases
Centro de Tapas Ocupado	%I11.2	%Q1.2	Paro Centro Tapas
Centro de bases Ocupado	%I11.3	%Q1.3	Paro Centro Bases
Error Centro de tapas	%I11.4	%Q1.4	Resetear Contador Tapas
Error Centro de Bases	%I11.5	%Q1.5	Resetear Contador Bases
	%I11.6	%Q1.6	
	%I11.7	%Q1.7	
Sensor Final Cinta Entrada Tapas	%I12.0	%Q2.0	Motor Cinta Entrada Tapas
Sensor Final Cinta Entrada Bases	%I12.1	%Q2.1	Motor Cinta Entrada Bases
Sensor Salida Centro Tapas	%I12.2	%Q2.2	Motor Cinta1 Salida Tapas
Sensor Salida Centro Bases	%I12.3	%Q2.3	Motor Cinta1 Salida Bases
	%I12.4	%Q2.4	Motor Cinta2 Salida Tapas
	%I12.5	%Q2.5	Motor Cinta2 Salida Bases
	%I12.6	%Q2.6	Motor Cinta Salida
	%I12.7	%Q2.7	
	%ID30	(DINT) %QD30	Display Contador Tapas
	%ID34	(DINT) %QD34	Display Contador Bases
Centro de Tapas Funcionando	%ID38 (DINT)	%QD38	
Centro de Bases Funcionando	%ID42 (DINT)	%QD42	

Figura 126. Distribución de variables. Escena producción en línea

Cabe destacar que el diseño de esta práctica de simulación no estructura la programación de la estación en base a ningún GRAFCET. Aunque la programación se podría haber dispuesto como un proceso secuencial en un GRAFCET, al no ser un proceso estrictamente secuencial, se ha decidido no introducir esta estructura.

### 5.5.3 Programación del autómeta

La programación del autómeta se realiza en lenguaje KOP, como el resto de las estaciones, sin embargo, no sigue ninguna estructura GRAFCET.

Como en algunas de las estaciones anteriores, no se mostrarán mediante tablas las variables contenidas en DB's, solo las contenidas en el PLC, siendo el conjunto de ellas las reflejadas a continuación:

Variables PLC					
	Nombre	Tabla de variables	Tipo de datos	Dirección	
1	Marcha	Tabla de variables e..	Bool	%I10.0	
2	Paro	Tabla de variables e..	Bool	%I10.1	
3	Emergencia	Tabla de variables e..	Bool	%I10.2	
4	Funcionamiento_Manual	Tabla de variables e..	Bool	%I10.3	
5	Funcionamiento_Automático	Tabla de variables e..	Bool	%I10.4	
6	Centro_Tapas_Abierto	Tabla de variables e..	Bool	%I11.0	
7	Centro_Bases_Abierto	Tabla de variables e..	Bool	%I11.1	
8	Centro_Tapas_Ocupado	Tabla de variables e..	Bool	%I11.2	
9	Centro_Bases_Ocupado	Tabla de variables e..	Bool	%I11.3	
10	Error_Centro_Tapas	Tabla de variables e..	Bool	%I11.4	
11	Error_Centro_Bases	Tabla de variables e..	Bool	%I11.5	
12	Sensor_Final_Cinta_Entrada_Tapas	Tabla de variables e..	Bool	%I12.0	
13	Sensor_Final_Cinta_Entrada_Bases	Tabla de variables e..	Bool	%I12.1	
14	Sensor_Salida_Centro_Tapas	Tabla de variables e..	Bool	%I12.2	
15	Sensor_Salida_Centro_Bases	Tabla de variables e..	Bool	%I12.3	
16	Sensor_Cinta2_Salida_Tapas	Tabla de variables e..	Bool	%I12.4	
17	Sensor_Cinta2_Salida_Bases	Tabla de variables e..	Bool	%I12.5	
18	Centro_Tapas_Funcionando	Tabla de variables e..	DInt	%ID38	
19	Centro_Bases_Funcionando	Tabla de variables e..	DInt	%ID42	
20	Luz_Marcha	Tabla de variables e..	Bool	%Q0.0	
21	Luz_Paro	Tabla de variables e..	Bool	%Q0.1	
22	Luz_Forzado_Manual	Tabla de variables e..	Bool	%Q0.2	
23	Marcha_Centro_Tapas	Tabla de variables e..	Bool	%Q1.0	
24	Marcha_Centro_Bases	Tabla de variables e..	Bool	%Q1.1	
25	Paro_Centro_Tapas	Tabla de variables e..	Bool	%Q1.2	
26	Paro_Centro_Bases	Tabla de variables e..	Bool	%Q1.3	
27	Resetear_Contador_Tapas	Tabla de variables e..	Bool	%Q1.4	
28	Resetear_Contador_Bases	Tabla de variables e..	Bool	%Q1.5	
29	Actuador_Centro_Tapas	Tabla de variables e..	Bool	%Q1.6	
30	Actuador_centro_Bases	Tabla de variables e..	Bool	%Q1.7	
31	Motor_Cinta_Entrada_Tapas	Tabla de variables e..	Bool	%Q2.0	
32	Motor_Cinta_Entrada_Bases	Tabla de variables e..	Bool	%Q2.1	
33	Motor_Cinta1_Salida_Tapas	Tabla de variables e..	Bool	%Q2.2	
34	Motor_Cinta1_Salida_Bases	Tabla de variables e..	Bool	%Q2.3	
35	Motor_Cinta2_Salida_Tapas	Tabla de variables e..	Bool	%Q2.4	
36	Motor_Cinta2_Salida_Bases	Tabla de variables e..	Bool	%Q2.5	
37	Motor_Cinta_Salida	Tabla de variables e..	Bool	%Q2.6	
38	Display_Contador_Tapas	Tabla de variables e..	DInt	%QD30	
39	Display_Contador_Bases	Tabla de variables e..	DInt	%QD34	

Tabla 9. Variables del PLC. Estación producción en línea

Como en el resto de estaciones, las variables de entrada y salida son las empleadas en el programa de simulación del entorno virtual de Factory IO, compartiendo las mismas direcciones para el correcto funcionamiento de la simulación.

La estructura del programa es la siguiente:

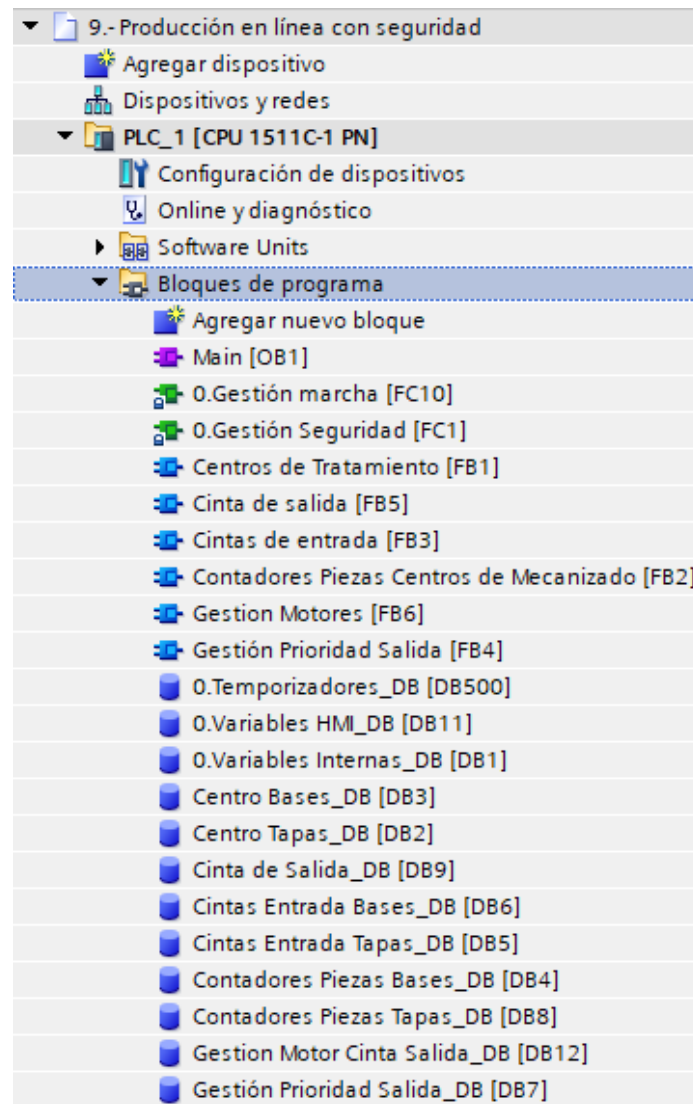


Figura 127. Distribución de bloques. Estación producción en línea

A diferencia de la mayoría de las estaciones anteriores, no es necesario el bloque FC9000 para la conexión entre Factory IO y PLCSIM, puesto que el PLC empleado es un S7-1500 y se simulará con PLCSIM Advanced.

Cabe destacar, de la figura anterior, los símbolos de los candados en los bloques de “gestión seguridad” y “gestión marcha”. Este símbolo indica que los bloques se encuentran bajo la **protección “Know-how”**, que impide abrir o modificar la programación contenida en el bloque sin la contraseña de ese bloque. Solamente se ha protegido estos dos bloques porque son los que contienen la programación relacionada con la seguridad y con marcha de la estación. Sin ellos, las condiciones de marcha y de seguridad no se podrían modificar. Si se desea abrir estos bloques, es necesario poner la contraseña de cada uno de ellos. Se ha dejado escrita la contraseña en el nombre del segmento del main donde se llama a cada bloque. Aunque esto no se debería hacer por motivos lógicos de seguridad, se ha decidido dejar constancia de la contraseña dentro del programa del PLC por si se olvida.

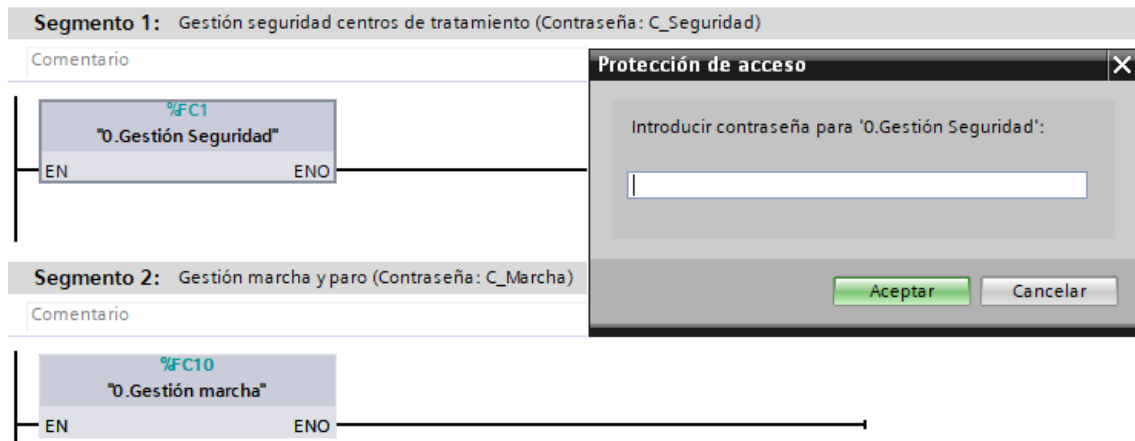


Figura 128. Protección Know-how. Estación producción en línea

Como se observa en la figura 127, solo se ha incluido un bloque de organización ya que se ha considerado que no era necesario incluir una OB de arranque.

**Main [OB1]:** contiene la programación completa de la estación, encargándose de llamar al resto de bloques de programa y realizar las operaciones necesarias con las salidas de estos bloques.

Los bloques contenidos en el main y su función se analizan a continuación:

**0.Gestión seguridad [FC1]:** como su nombre indica, se encarga de gestionar la seguridad de los centros de tratamiento, indicando que las condiciones de seguridad son correctas si la puerta del recinto está cerrada y la seguridad rearmada. En caso contrario, las condiciones de seguridad no se cumplen, lo que repercutirá en el estado de marcha de la estación según la programación de la FC10, "0.Gestión marcha". Al trabajar con entradas y salidas del PLC o con variables internas, no es necesario definir ninguna entrada o salida dentro de la FC.

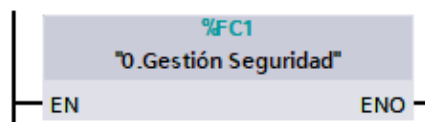


Figura 129. FC1 0.Gestión Seguridad. Estación producción en línea

**0.Gestión marcha [FC10]:** se encarga de gestionar la marcha y paro de la estación y de los centros de tratamiento, así como de encender las señales luminosas que indican sus estados. También incluye la programación del funcionamiento automático y manual, y su forzado. No es necesario definir ninguna entrada o salida dentro de la FC por el mismo motivo que en el bloque anterior.

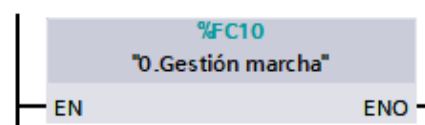


Figura 130. FC10 0.Gestión marcha. Estación producción en línea

**Gestión prioridad salida [FB4]:** contiene la programación encargada de asignar la prioridad a las bases o a las tapas. Como se ha mencionada anteriormente, las tapas y bases producidas por los centros de tratamiento se unen en una misma línea, por lo que hay que **asignar prioridades** para que no colisiones en la concurrencia de las cintas. En el caso de que ambas piezas salgan a la par, las bases tendrán prioridad frente a las tapas. Esta prioridad se asigna en función de la pieza que antes salga del centro, manteniendo una prioridad de un segundo frente a la pieza que espera.

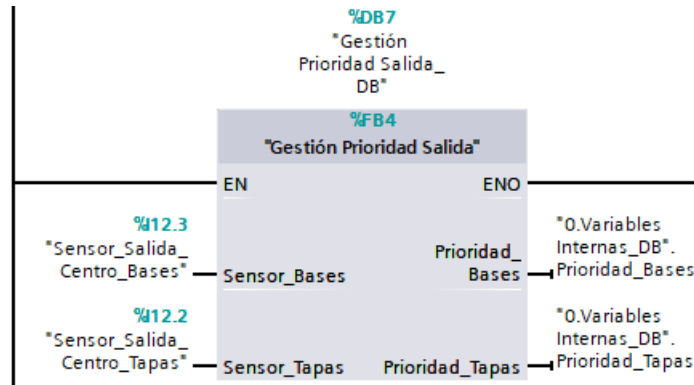


Figura 131. FB4 Gestión Prioridad Salida. Estación producción en línea

**Cintas de entrada [FB3]:** contiene la programación encargada de la activación o desactivación de las cintas por donde entra la materia prima a la estación. Este bloque se emplea para ambas cintas, ya que su funcionamiento es idéntico. Para el correcto funcionamiento del bloque hay que introducir en sus entradas los sensores y variables internas del programa pertinentes y en su salida el motor de la cinta.

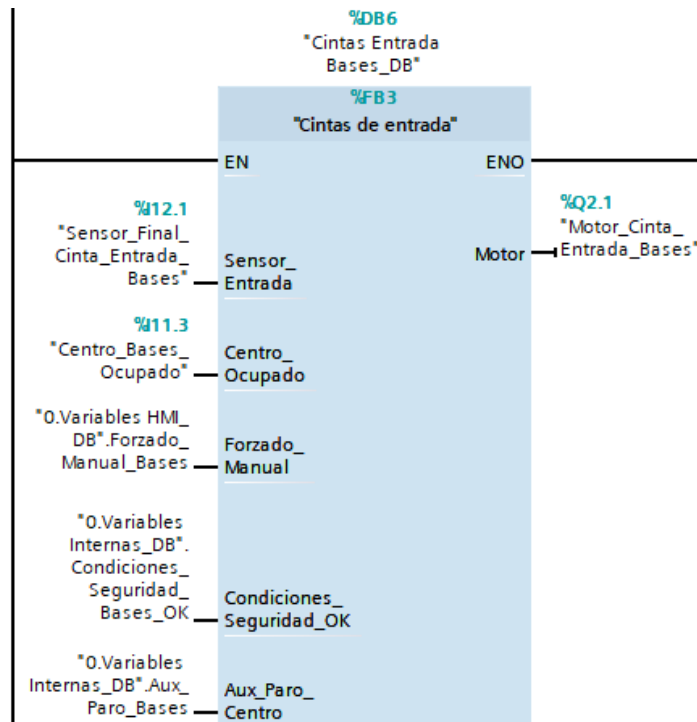


Figura 132. FB3 Cintas de entrada de las bases. Estación producción en línea

**Centros de tratamiento [FB1]:** gestiona la activación y desactivación de las cintas de salida de los centros de tratamiento en base a las prioridades asignadas. Para ello, es necesario indicar en sus entradas la variable de prioridad de cada pieza y el sensor ubicado al inicio de las cintas transportadoras. Como salida se emplea una variable auxiliar que activa las dos cintas de salida en forma de “L”. Ambos centros emplean el mismo bloque de programación.

Conviene recordar que estas cintas no disponen de un sensor a su salida, por lo que la desactivación de la cinta debe estar condicionada a otra variable. Esta variable es temporal y proviene de un **acumulador de tiempo TONR**. Es necesario emplear este temporizador y no cualquier otro ya que es indispensable mantener el tiempo de conteo si la estación entra en paro o emergencia, reanudando su marcha sin perder el tiempo que ya había estado en funcionamiento la cinta.

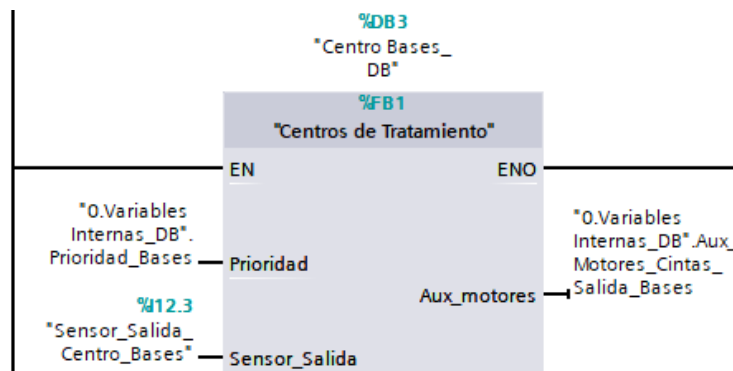


Figura 133. FB1 Centros de tratamiento de bases. Estación producción en línea

**Contadores piezas centros de mecanizado [FB2]:** contiene la programación de los contadores de piezas procesadas por cada centro. Este bloque se emplea en los dos centros ya que su funcionamiento es igual. Como es lógico, sus entradas están condicionadas a los sensores ubicados a la salida de cada centro y al pulsador de reset de su cuadro eléctrico. También convierte el valor de la variable del contador en un tipo de dato que se pueda mostrar por el display del cuadro eléctrico, siendo esta su única salida del bloque.

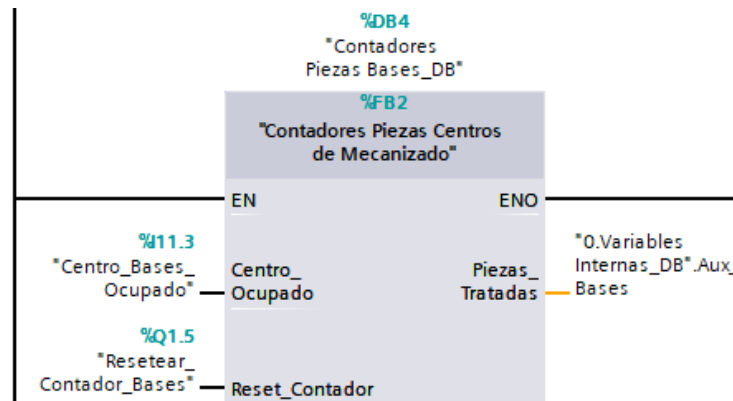


Figura 134. FB2 Contadores centros mecanizado. Estación producción en línea

**Cinta de salida [FB5]:** su programación se limita a detectar la solicitud de activación de la cinta de salida.

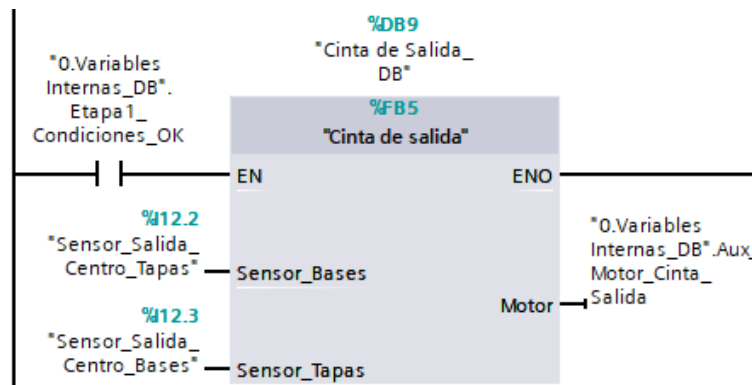


Figura 135. FB5 Cinta de salida. Estación producción en línea

**Gestión motores [FB6]:** contienen la programación de la activación y desactivación de la cinta de salida en función de la solicitud tratada en el bloque anterior y de la lectura de los sensores ubicados a la salida de los centros. Al igual que en el caso de las cintas de salida de los centros, su desactivación viene condicionada por una variable temporal de la **instrucción TONR**.

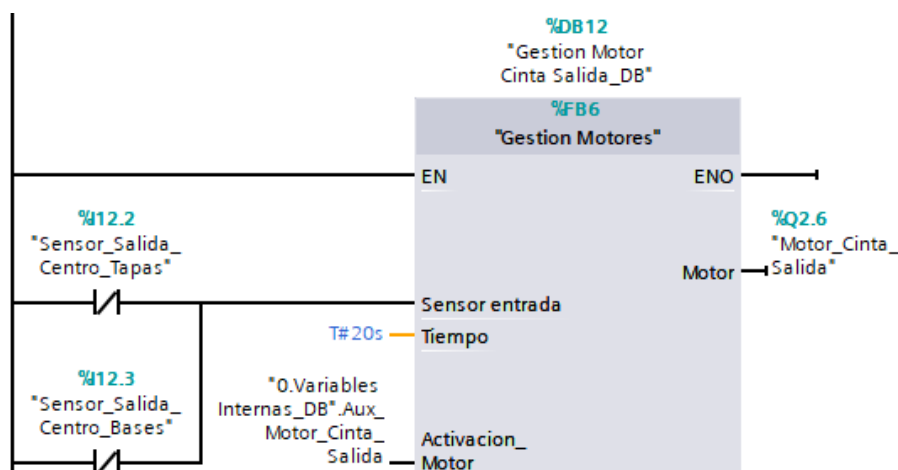


Figura 136. FB6 Gestión motores. Estación producción en línea

Se puede concluir, que para la realización del programa ladder, es necesario conocer y aplicar correctamente los conocimientos relacionados con TIA Portal marcados en los objetivos que adquirir al comienzo de la descripción de la estación.

### 5.5.4 Programación del HMI

La programación de este elemento se realiza desde TIA Portal y comprende la creación de la conexión entre el PLC y el HMI, la vinculación de las variables pertinentes y su representación mediante diferentes elementos en las imágenes que se desean mostrar por pantalla.

Se ha empleado un modelo HMI TP700 Comfort. Los modelos “*comfort*” permiten programar una administración de usuarios que los modelos “*basic*” no contemplan. Esta administración de usuarios se encarga de gestionar los permisos, limitando a cada usuario a interactuar únicamente con los elementos de las pantallas que emplean los permisos del grupo al que pertenece el usuario o los que no requieran permisos.

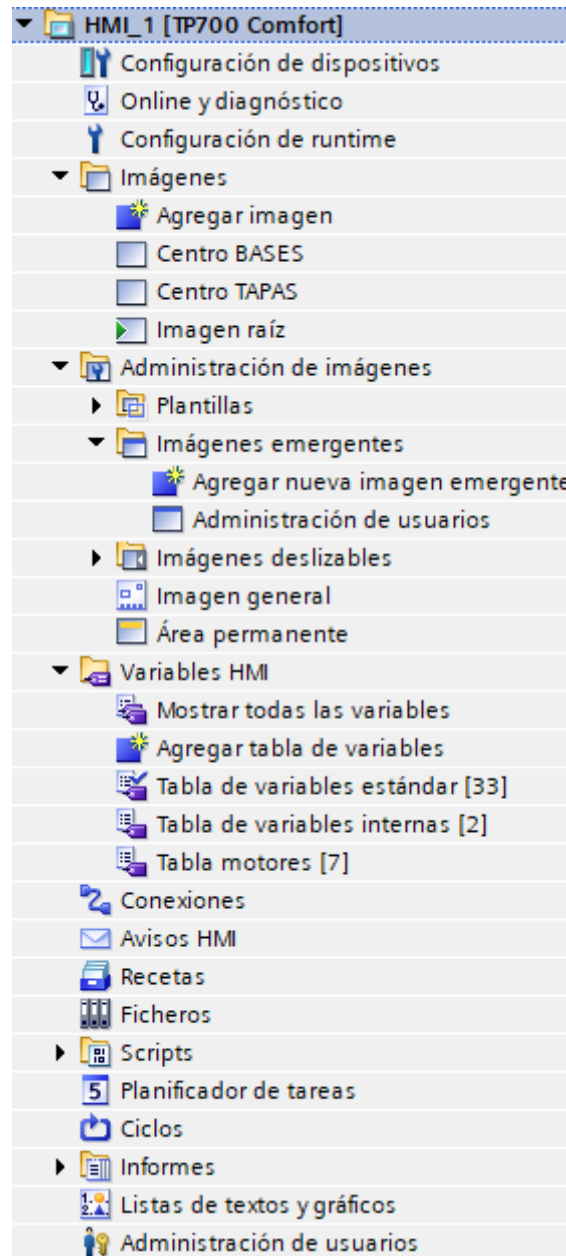


Figura 137. Árbol de proyecto del HMI. Estación producción en línea

Una vez agregado el panel de operador, se crea la **conexión ETHERNET con el PLC**, asignando una dirección IP de la misma red y con un valor de 192.168.0.2. Mediante esta conexión se intercambia información entre ambos dispositivos de control.

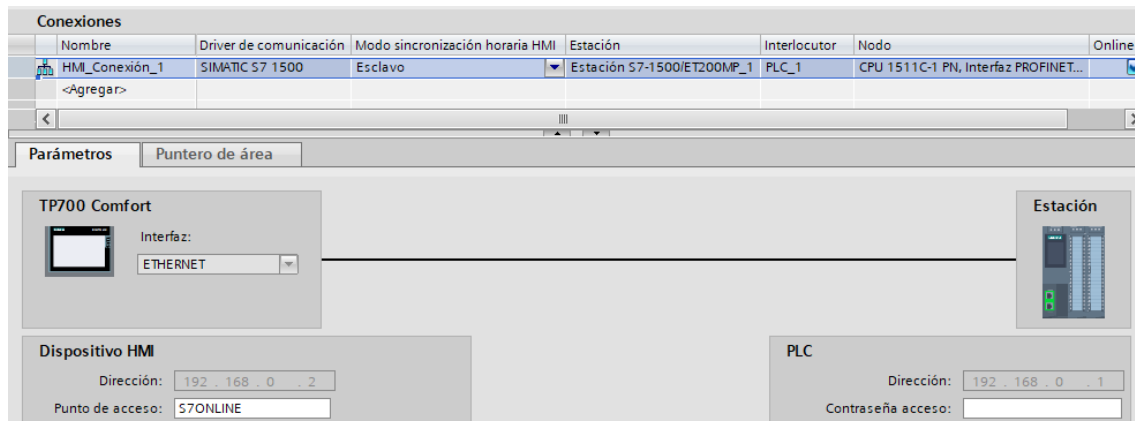


Figura 138. Conexión ETHERNET entre HMI y PLC. Estación producción en línea

Como se aprecia en la figura 137, las variables empleadas en el HMI están contenidas en tres tablas de variables. Cabe destacar que para emplear las variables del PLC en el HMI se requiere asociarlas a una variable dentro del HMI, este modo de acceso se hace mediante acceso simbólico en todos los casos. Las variables empleadas en el HMI son las siguientes:

Tabla de variables estándar						
Nombre	Tipo de datos	Conexión	Nombre del PLC	Variable PLC	Dirección	Modo de acceso
0.Variables Internas_DB_Condiciones_Seguridad_Bases_OK	Bool	HMI_Conexión_1	PLC_1	*0.Variables Internas_DB*.Condiciones_...		<Acceso simbólico>
0.Variables Internas_DB_Condiciones_Seguridad_Tapas_OK	Bool	HMI_Conexión_1	PLC_1	*0.Variables Internas_DB*.Condiciones_...		<Acceso simbólico>
HMI_Automatico	Bool	HMI_Conexión_1	PLC_1	Funcionamiento_Automatico		<Acceso simbólico>
HMI_Centro_Bases_Ocupado	Bool	HMI_Conexión_1	PLC_1	Centro_Bases_Ocupado		<Acceso simbólico>
HMI_Centro_Tapas_Ocupado	Bool	HMI_Conexión_1	PLC_1	Centro_Tapas_Ocupado		<Acceso simbólico>
HMI_Contador_Bases	DInt	HMI_Conexión_1	PLC_1	Display_Contador_Bases		<Acceso simbólico>
HMI_Contador_Tapas	DInt	HMI_Conexión_1	PLC_1	Display_Contador_Tapas		<Acceso simbólico>
HMI_Forzado_Manual_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Forzado_Manual_...		<Acceso simbólico>
HMI_Forzado_Manual_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Forzado_Manual_...		<Acceso simbólico>
HMI_Manual	Bool	HMI_Conexión_1	PLC_1	Funcionamiento_Manual		<Acceso simbólico>
HMI_Marcha	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Marcha_HMI		<Acceso simbólico>
HMI_Marcha_Centro_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Marcha_Centro_B...		<Acceso simbólico>
HMI_Marcha_Centro_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Marcha_Centro_T...		<Acceso simbólico>
HMI_Paro	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Paro_HMI		<Acceso simbólico>
HMI_Paro_Centro_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Paro_Centro_Bases		<Acceso simbólico>
HMI_Paro_Centro_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Paro_Centro_Tapas		<Acceso simbólico>
HMI_Prioridad_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables Internas_DB*.Prioridad_Base		<Acceso simbólico>
HMI_Prioridad_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables Internas_DB*.Prioridad_Tapa		<Acceso simbólico>
HMI_Puerta_Abierta_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Puerta_Abierta_Ba.		<Acceso simbólico>
HMI_Puerta_Abierta_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Puerta_Abierta-Ta.		<Acceso simbólico>
HMI_Rearmar_Bases	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Rearme_Bases		<Acceso simbólico>
HMI_Rearmar_Tapas	Bool	HMI_Conexión_1	PLC_1	*0.Variables HMI_DB*.Rearme_Tapas		<Acceso simbólico>
HMI_Reset_Bases	Bool	HMI_Conexión_1	PLC_1	Resetear_Contador_Bases		<Acceso simbólico>
HMI_Reset_Tapas	Bool	HMI_Conexión_1	PLC_1	Resetear_Contador_Tapas		<Acceso simbólico>
HMI_Tiempo_Base	DInt	HMI_Conexión_1	PLC_1	Centro_Bases_Funcionando		<Acceso simbólico>
HMI_Tiempo_Tapa	DInt	HMI_Conexión_1	PLC_1	Centro_Tapas_Funcionando		<Acceso simbólico>
Luz_Marcha	Bool	HMI_Conexión_1	PLC_1	Luz_Marcha		<Acceso simbólico>
Luz_Paro	Bool	HMI_Conexión_1	PLC_1	Luz_Paro		<Acceso simbólico>
Marcha_Centro_Bases	Bool	HMI_Conexión_1	PLC_1	Marcha_Centro_Bases		<Acceso simbólico>
Marcha_Centro_Tapas	Bool	HMI_Conexión_1	PLC_1	Marcha_Centro_Tapas		<Acceso simbólico>
Paro_Centro_Bases	Bool	HMI_Conexión_1	PLC_1	Paro_Centro_Bases		<Acceso simbólico>
Paro_Centro_Tapas	Bool	HMI_Conexión_1	PLC_1	Paro_Centro_Tapas		<Acceso simbólico>

Tabla 10. Variables estándar del HMI. Estación producción en línea

Tabla motores						
Nombre	Tipo de datos	Conexión	Nombre del PLC	Variable PLC	Dirección	Modo de acceso
HMI_Motor_Cinta_Entrada_Bases	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta_Entrada_Bases		<Acceso simbólico>
HMI_Motor_Cinta_Entrada_Tapas	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta_Entrada_Tapas		<Acceso simbólico>
HMI_Motor_Cinta_Salida	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta_Salida		<Acceso simbólico>
HMI_Motor_Cinta1_Salida_Bases	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta1_Salida_Bases		<Acceso simbólico>
HMI_Motor_Cinta1_Salida_Tapas	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta1_Salida_Tapas		<Acceso simbólico>
HMI_Motor_Cinta2_Salida_Bases	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta2_Salida_Bases		<Acceso simbólico>
HMI_Motor_Cinta2_Salida_Tapas	Bool	HMI_Conexión_1	PLC_1	Motor_Cinta2_Salida_Tapas		<Acceso simbólico>

Tabla 11. Variables de motores del HMI. Estación producción en línea

Nombre ▲	Tipo de datos	Conexión
ContraseñaBackup	WString	<Variable interna>
NombreUsuario	WString	<Variable interna>

Tabla 12. Variables internas del HMI. Estación producción en línea

En la *tabla 12* se muestran las **variables internas del HMI**. Estas variables no se emplean en el PLC, solo en el HMI. Por lo que no necesitan ninguna conexión o modo de acceso.

En lo referente a las imágenes que contiene el HIM, como se han mostrado en el primer apartado relacionado con esta estación, se considera que no aportaría ninguna información adicional adjuntarlas también en este apartado. Sin embargo, sí que es importante explicar el uso de tres elementos nuevos empleados en esta estación, como son los permisos asociados a la administración de usuarios, las pantallas emergentes y el planificador de tareas.

Dentro del árbol de proyectos del HMI, en el apartado de **“Administración de usuarios”** se ha creado 3 usuarios distintos asignándoles una contraseña a cada uno. El **“Operario1”** (con contraseña **“C\_Operario1”**), el **“Operario2”** (con contraseña **“C\_Operario2”**) y el **“Jefe\_Planta”** (con contraseña **“C\_J\_Planta”**). Los dos primeros pertenecen al **“grupo de operarios”** mientras que el último pertenece al **“grupo de administradores”**.

9.- Producción en línea con seguridad > HMI_1 [TP700 Comfort] > Administración de usuarios					
<b>Usuarios</b>					
	Nombre ▲	Contraseña	Cierre de sesión autom..	Tiempo de cierre de sesión	Número
	Jefe_Planta	*****	<input checked="" type="checkbox"/>	5	1
	Operario1	*****	<input checked="" type="checkbox"/>	60	2
	Operario2	*****	<input checked="" type="checkbox"/>	60	3
<b>Grupos</b>					
	Miembro de	Nombre	Número	Nombre de visualización	Caducidad de l..
	<input checked="" type="radio"/>	Grupo de administradores	1	ADMINISTRADORES	<input type="checkbox"/>
	<input type="radio"/>	Grupo de operarios	2	OPERARIOS	<input type="checkbox"/>

Figura 139. Administración de usuarios HMI. Estación producción en línea

Cada grupo posee unos **permisos** que les permitirán interactuar con unos elementos concretos de cada imagen, siendo los 4 tipos de permisos los siguientes:

- **Administración de usuarios:** puede supervisar el registro de usuarios que se han conectado al HMI y cambiar su contraseña desde el visor de usuarios en tiempo de ejecución.
- **Forzado:** Permite el forzado manual de los centros de tratamiento.
- **Operación:** Permite controlar el funcionamiento de la estación en condiciones normales.
- **Seguridad:** Para poder rearmar las seguridades de los centros de tratamiento después de que se abra la puerta del recinto.

Como se aprecia en la figura inferior, el grupo de operarios solo tiene el permiso de operación, puesto que con este permiso es posible controlar y supervisar el funcionamiento de la estación completa en condiciones normales. Por otro lado, los administradores tienen todos los permisos, permitiéndoles realizar acciones de forzado manual, rearmar las seguridades o modificar las contraseñas de cualquier usuario en tiempo de ejecución desde el HMI.

Grupos			
Nombre	Número	Nombre de visualización	Caducidad de l...
Grupo de administradores	1	ADMINISTRADORES	<input type="checkbox"/>
Grupo de operarios	2	OPERARIOS	<input type="checkbox"/>

Permisos			
Activo	Nombre	Nombre de visualización	Número
<input type="checkbox"/>	Administración de usuari...	Administración de usuarios	1
<input type="checkbox"/>	Forzado	Forzado	2
<input checked="" type="checkbox"/>	Operación	Operación	3
<input type="checkbox"/>	Seguridad	Seguridad	4

Figura 140. Permisos de usuarios HMI. Estación producción en línea

Una vez creados los usuarios, grupos y permisos, es necesario asociarlos a los elementos de las imágenes. En la figura inferior se puede ver que dentro de las propiedades de los elementos (en este caso el botón “Usuarios” de la imagen raíz) se encuentra un apartado de seguridad donde se le puede introducir un permiso para restringir la interacción con él. Si el usuario registrado en el momento no tiene el permiso requerido para interactuar con el elemento que acaba de seleccionar, salta una ventana de aviso y otra ventana para iniciar sesión con un usuario que sí que tenga el permiso.

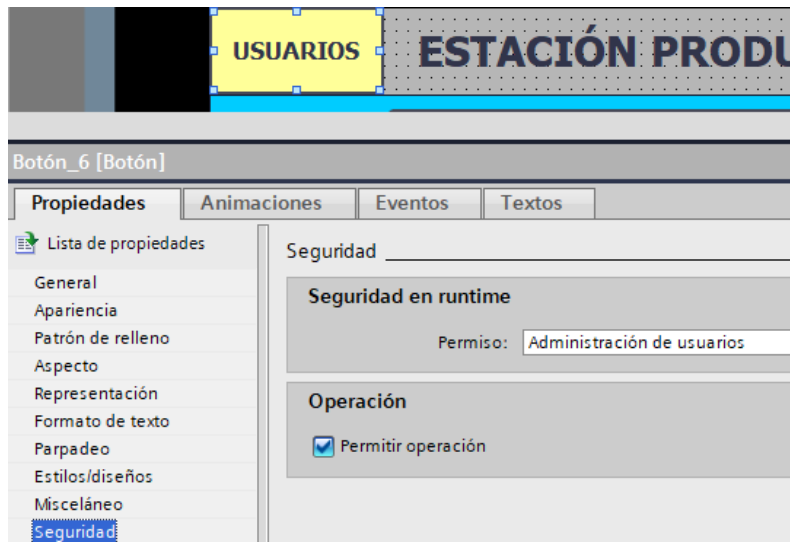


Figura 141. Propiedades del botón “Usuarios”. Estación producción en línea

Además de la administración de usuarios, se ha empleado una pantalla emergente si se pulsa en el botón de “USUARIOS” mostrado en la imagen anterior. Este tipo de pantallas se pueden crear en el árbol de proyectos del HMI, en el apartado de “Pantallas emergentes” dentro de “administración de imágenes”.

Por último, el “**planificador de tareas**” del HMI (también accesible desde el árbol de proyecto del panel de operador) permite crear tareas. Las tareas son eventos que se ejecutan cuando pasa un suceso llamado disparador.

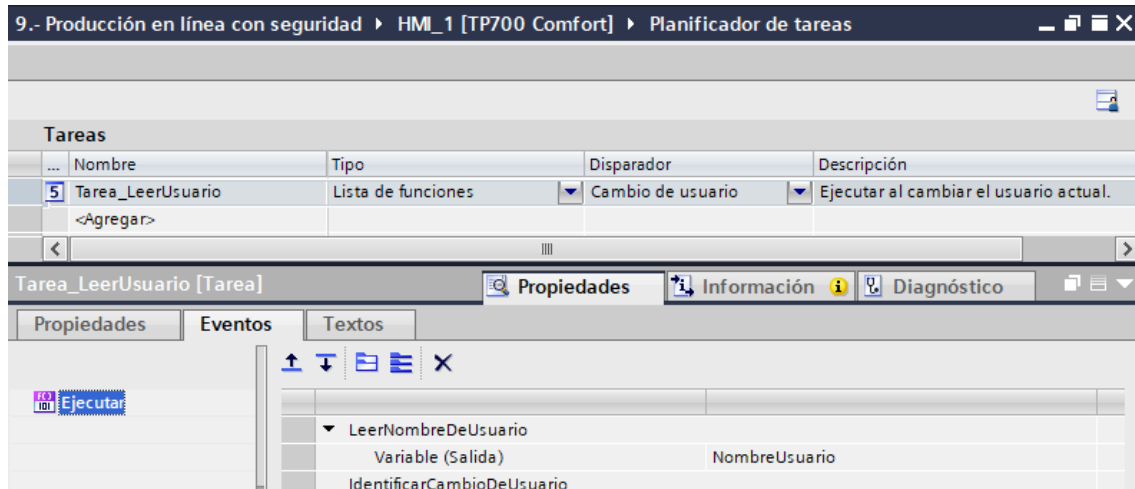


Figura 142. Planificador de tareas HMI. Estación producción en línea

En el caso de la única tarea creada en el programa del HMI, cada vez que se cambie de usuario se lee el nombre de usuario nuevo y se le asigna a la variable interna del HMI llamada “*NombreUsuario*”. Esta variable se emplea en las imágenes para mostrar por pantalla qué usuario está conectado.

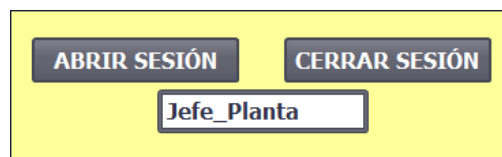


Figura 143. Detalle de usuario conectado del HMI. Estación producción en línea

Llegado este punto, la programación de todos los elementos de control de la estación, PLC y HMI, se ha realizado por completo. Queda de manifiesto que para la programación de ambos elementos se deben conocer y aplicar correctamente los conocimientos marcados en los objetivos que adquirir al comienzo de la descripción de la estación. Por lo tanto, se podría justificar el cumplimiento de la finalidad didáctica de esta práctica.



## 6. INTEGRACIÓN DE LAS PRÁCTICAS EN LA DOCENCIA

Uno de los objetivos de este trabajo es la integración de las estaciones realizadas en la docencia como prácticas de simulación en grados y másteres de la UVa. Para ello, se ha tomado las guías docentes de las asignaturas relacionadas con la automatización [14] y se ha estudiado cuál es su grado de adecuación en función de los objetivos y competencias de cada asignatura y los conceptos relacionados con el grado y máster en específico.

Los objetivos de aprendizaje cumplidos con las simulaciones son los siguientes:

ASIGNATURA	OBJETIVOS DE APRENDIZAJE CUMPLIDOS
Automatización Industrial (Grado en Ingeniería Electrónica Industrial y Automática)	Conocer y aplicar componentes y sistemas de control y automatización industriales.
	Adquirir conocimiento básico sobre uso y aplicación de sensores y transductores industriales.
	Analizar sistemas de automatización reconociendo sus módulos fundamentales y las técnicas utilizadas para su diseño.
	Tener actitud ingenieril para el tratamiento de los problemas de automatización.
	Utilizar métodos adecuados para la automatización.
	Conocer los dispositivos y tecnologías que se están utilizando en la automatización industrial.
	Diseñar sistemas de automatización.
	Conocer herramientas básicas de ingeniería para la realización de cálculos matemáticos, simulación y de programación utilizados en la automatización de procesos industriales.
	Afianzar los conceptos teóricos mediante las prácticas que se desarrollan en el laboratorio.
Adaptarse a la evolución que sufrirá este tipo de tecnología ya que se presenta gran perspectiva de futuro.	
Automatización Industrial (Grado en Ingeniería en Organización Industrial)	Entender el funcionamiento y aplicación de transductores y actuadores industriales.
	Seleccionar componentes para la Automatización Industrial.
	Analizar sistemas de automatización reconociendo sus módulos fundamentales y las técnicas utilizadas para su diseño.
	Conocer las tecnologías que se están utilizando en la Automatización Industrial.
	Manejar herramientas ingenieriles para la programación en la Automatización de procesos.
	Adaptarse a la evolución que sufrirá la tecnología asociada a la Automatización, ya que presenta una gran perspectiva de futuro.
	Programar y realizar la puesta en marcha de autómatas programables.
	Programar robots manipuladores. Identificar las restricciones temporales asociadas al control de sistemas industriales.
	Implementar controladores en sistemas informáticos.
	Diseñar controladores tolerantes a fallos y averías.
Seleccionar la arquitectura de comunicación más adecuada.	
Integrar dispositivos y sistemas distribuidos.	
Sistemas de Control Industrial (Grado en Ingeniería Eléctrica)	Analizar y diseñar sistemas de control basados en autómatas programables.
	Analizar e identificar los componentes funcionales de una arquitectura de red y seleccionar el protocolo adecuado en función de los requisitos de la aplicación.
	Determinar el tipo de red industrial más adecuado para una aplicación de automatización concreta, y seleccionar los protocolos y servicios correspondientes.
	Planificar y dimensionar entornos de control, supervisión y registro de datos de un proceso industrial gobernado por autómatas autónomos o en red.
Tecnología de Control (Máster en Ingeniería Industrial)	Comprender las técnicas de programación de sistemas de control en la industria.
	Conocer los conceptos básicos del control basado en sistemas inteligentes.
	Conocer los componentes de un robot, su sistema de control, sensores y accionamientos y los principios de su programación.
	Conocer los fundamentos de las líneas de producción y los retos actuales de seguridad asociados.

Tabla 13. Objetivos de aprendizaje cumplidos según las guías docentes

En el **grado de Ingeniería en Electrónica Industrial y Automática** se estudian conceptos no solo relacionados con la automatización, sino también con la robótica y el control de procesos.

Aunque en el **grado de Ingeniería en Organización Industrial** se impartan conocimientos de automatización de procesos, tiene un peso más importante la gestión de actividades y procesos industriales.

En el **grado de Ingeniería Eléctrica**, concretamente en la asignatura de Sistemas de Control Industrial, se trabajan conceptos de automatización haciendo énfasis en la comunicación en entornos industriales.

Por último, en el **Máster en Ingeniería Industrial**, se pretende dotar al alumno de conocimientos relacionados con el conjunto de carreras englobadas en las ingenierías industriales. Por lo tanto, todos los conceptos antes mencionados están incluidos en su temario. En la asignatura de Tecnología de control se hace hincapié en la automatización, la robótica y el control de procesos.

Realmente todas las simulaciones realizadas en este trabajo podrían emplearse en los grados y másteres antes mencionados, sin embargo, según la complejidad y los conocimientos tratados en ellas, se puede distinguir varios grados de adecuación. Para representar la idoneidad de las simulaciones a los estudios antes mencionados, se ha creado una tabla con diferentes grados de conveniencia donde XXX es muy conveniente, XX bastante conveniente y X algo conveniente:

ESTUDIOS \ ESTACIÓN	Ingeniería en Electrónica Industrial y Automática	Ingeniería en Organización Industrial	Ingeniería Eléctrica	Máster en Ingeniería Industrial
Paletizadora	XXX	XX	XX	XX
Distribuidora	XXX	XX	XX	XX
Depósito de líquido	XXX	XX	XX	XXX
Gestión de stock de un almacén	XX	XXX	XXX	XXX
Cinta transportadora	XX	XXX	XX	X
Calculadora	X	XXX	XXX	X
Clasificadora	X	XXX	X	X
Colocador por peso	XXX	XX	XX	XXX
Producción en línea	XXX	X	XXX	XXX

Tabla 14. Conveniencia de las simulaciones según los grados y másteres



## 7. ESTUDIO ECONÓMICO

Con el objetivo de realizar un seguimiento del coste asociado a la realización de este trabajo, se han analizado los diferentes factores y se han estimado los costes de su realización, dividiéndolos en costes directos e indirectos para facilitar su cálculo.

### Costes directos

Los costes directos son los que se pueden asociar de forma clara a un producto o servicio concreto, en el caso de este trabajo, a la creación de las simulaciones. Estos costes directos hacen referencia a los costes de personal.

Para la estimación de este coste, se ha simplificado el cálculo suponiendo que el trabajo ha sido realizado por un ingeniero recién titulado y con una estimación de lo que podría ser su sueldo. De esta forma, en función de las horas dedicadas a cada actividad, obtendríamos la siguiente tabla:

ACTIVIDAD		HORAS
Adquisición de conocimientos de programación de TIA Portal		60
Manejo de software Factory IO		5
Manejo de software SIMIT		10
Configuración de comunicaciones entre TIA Portal y Factory IO		1
Configuración de comunicaciones entre TIA Portal y SIMIT		4
Proceso de actualización de estaciones		42
Estación cinta transportadora	Diseño	2
	Programación	9
	Documentación	7
Estación calculadora	Diseño	1
	Programación	8
	Documentación	3
Estación clasificadora	Diseño	12
	Programación	30
	Documentación	4
Estación colocador por peso	Diseño	27
	Programación	43
	Documentación	7
Estación producción en línea	Diseño	9
	Programación	37
	Documentación	6
Redacción de memoria		50
<b>TOTAL</b>		<b>377</b>

Tabla 15. Coste de personal

Suponiendo el salario del ingeniero recién titulado de 18€/h, obtenemos el siguiente coste de personal:

$$18€/h \cdot 377 h = 6.786 €$$

### Costes indirectos

Son costes que no se pueden relacionar con un producto o servicio determinado, sino que son costes generales. Estos costes se dividen a su vez en costes de material y costes energéticos y de conexión.

- **Costes de material**

Como todas las estaciones son simulaciones y no se requiere de dispositivos físicos, los costes materiales están relacionados con los softwares y material de simulación y programación junto con sus licencias de uso.

MATERIALES Y SOFTWARES	COSTE (€)
Ordenador o PC	1.200'00
Software de programación SIEMENS (TIA Portal y SIMIT demo)	20'00
Software de simulación Factory IO	500'00
<b>TOTAL</b>	<b>1.720'00</b>

Tabla 16. Costes de material

Realmente tanto el ordenador como los softwares son materiales de los que ya dispone la Universidad de Valladolid, por lo tanto, los costes materiales para el organismo educativo para el que se ha pensado este proyecto son nulos. De todas formas, se ha decidido indicar los costes materiales para reflejar lo que supondría económicamente para una entidad que no disponga de estos materiales.

- **Costes energéticos y de conexión**

Teniendo en consideración que el trabajo ha tenido una duración de 377 horas, lo que equivale a casi 48 días laborables de 8 horas (2 meses y medio), los costes indirectos relacionados con el consumo eléctrico y el acceso a internet se muestra en la siguiente tabla:

CONCEPTO	COSTE (€)
Consumo eléctrico <a href="#">[15]</a>	62'28
Conexión a internet <a href="#">[16]</a>	135'00
<b>TOTAL</b>	<b>197'28</b>

Tabla 17. Costes energéticos y de conexión

Después de este estudio económico desglosado, se calcula el coste total del trabajo:

CATEGORÍA	COSTE (€)
Costes directos de personal	6.786'00
Costes indirectos de material	1.720'00
Costes indirectos energéticos y de conexión	197'28
<b>TOTAL</b>	<b>8.703'28</b>

Tabla 18. Costes totales



## 8. CONCLUSIONES Y LÍNEAS FUTURAS

### 8.1. CONCLUSIONES

Después de haber realizado todo el proceso de actualización, programación y simulación de las diversas estaciones mostradas en este trabajo, se pueden extraer distintas conclusiones.

La primera, y más importante, es el **cumplimiento del objetivo docente** perseguido en este proyecto. Para la realización de las estaciones, el alumno debe comprender y aplicar conocimientos relacionados con la programación y simulación de procesos industriales automatizados; no solo teóricos, sino también propios de software especializado en automatismos. De esta forma se relacionan los conceptos teóricos, descritos en la *tabla 13 “Objetivos de aprendizaje cumplidos según las guías docentes”*, con la programación práctica de PLC's, HMI's y sensores y actuadores de los gemelos digitales de las estaciones reales. En lo referente a softwares de programación, con las estaciones se adquieren conocimientos muy extensos de TIA Portal V17 (tanto de programación de PLC's como de HMI's), PLCSIM, PLCSIM Advanced, Factory IO y conocimientos básicos de SIMIT.

La segunda conclusión que se aprecia de este trabajo es la **diferencia entre los softwares que se han empleado para simular los gemelos digitales de las estaciones**. Si bien Factory IO no posee muchos elementos para simular, su manejo y programación son muy sencillos e intuitivos. Por el contrario, en SIMIT requiere más trabajo crear el gemelo digital, necesitando aprender las instrucciones que este software ofrece y disponiendo de una simulación en 2D en vez de en 3D. Aunque se han empleado ambos softwares con el objetivo de dotar a los alumnos de más conocimientos acerca de programas específicos para la automatización, si se tuviera que elegir solamente uno de ellos, está claro que, siempre y cuando los elementos de la estación a programar se encuentren en Factory IO, este programa es mucho más sencillo, rápido y visual que SIMIT.

La tercera conclusión que se puede extraer es la **importancia y funcionalidad de los softwares relacionados con la automatización** que se han empleado. No solo se siguen empleando estos programas en el mundo industrial (lo cual es un importante factor para su aplicación en la docencia), sino que además es un campo con mucha aplicación de cara al futuro. En adicción, el uso de software de simulación en vez de plantas reales abarata el coste de los proyectos, facilitando la puesta en marcha tras haber simulado previamente sin necesidad de ir al cliente a cargar el programa en la estación real.

El último aspecto que destacar es el **impacto positivo en el medioambiente** que puede generar el uso de gemelos digitales en la automatización de procesos industriales. Si bien, el avance y expansión de la industrialización produce un incremento del bienestar social y de la rentabilidad económica, conlleva un aumento de la contaminación debido a emisiones de carbono, desechos y otros impactos ambientales no deseados. Estos efectos negativos pueden reducirse con la aplicación de los gemelos digitales en la automatización industrial.



El uso de gemelos digitales contribuye a la optimización de los procesos, generando menor gasto energético, mayor aprovechamiento del material y, por lo tanto, menos generación de residuos. Esto se refleja con un menor consumo de recursos naturales. También minimiza el proceso de transporte entre empresas subcontratadas puesto que permite simular y programar sin necesidad de ir a la empresa cliente, lo que reduce las emisiones de carbono. Por último, promueve las prácticas circulares, fomentando la reutilización de las máquinas que ya tienen creadas sus gemelos digitales para ahorrar un coste mayor en nuevos proyectos, dando una nueva vida a las máquinas completas o elementos de ellas. En definitiva, los gemelos digitales permiten el cumplimiento de los objetivos económicos de la industria y de sostenibilidad, optimizando los procesos y reduciendo la huella ambiental que estos generan.

Después del análisis de estos cuatro puntos, se podría concluir que con la realización de estas simulaciones se consigue aplicar conocimientos de programación de PLC's, HMI's y PID's empleando software específico de automatización con un coste relativamente bajo y sin material gracias al uso de gemelos digitales. De esta forma, si se emplean las estaciones como prácticas docentes, se aportan tres nuevos conocimientos respecto al temario impartido con anterioridad. El primero es la programación y simulación de interfaces hombre-máquina (HMI), elemento muy importante en la industria y que completa los conocimientos de los dispositivos de control automático. El segundo hace referencia a la programación y control de dispositivos con un PID programado y ejecutado desde el mismo autómatas. Y finalmente, el uso de gemelos digitales, que facilita visualmente la comprobación del correcto funcionamiento del programa cargado en el PLC y el HMI.

En resumen, se podría concluir que con la realización de estas simulaciones se consiguen lograr los objetivos marcados en el [apartado 1.3](#) de esta memoria:

- **Adquirir y aplicar conocimientos** relacionados con la programación y simulación de la automatización de procesos industriales.
- **Comprender y programar** interfaces hombre-máquina.
- **Afianzar y emplear** competencias vinculadas con los controladores PID.
- **Comprender** las bases de los **protocolos de comunicación** entre los elementos de las estaciones industriales empleadas.
- **Actualizar cuatro estaciones de simulación**, indicando el proceso de actualización de los programas y dejándolas preparadas para su uso docente.
- **Programar** el proceso automático de varias estaciones industriales enfocadas al aprendizaje de la programación de la automatización de procesos.
- **Crear los entornos de simulación** de las estaciones industriales programadas.
- **Relacionar los conceptos teóricos y prácticos** empleados, con las **competencias y objetivos** de varias asignaturas de grado y másteres donde se podrían implementar.



## 8.2. LÍNEAS FUTURAS

En este apartado se indicarán posibles líneas futuras de trabajo, tomando como base este proyecto.

A nivel docente, los conocimientos básicos ya se han trabajado, empleado casi todos los elementos de Factory IO, una gran cantidad de instrucciones de TIA Portal V17 y dos simuladores distintos de autómatas programables. Sin embargo, si se quisiera profundizar más, hay cuatro aspectos que no se han trabajado en este proyecto debido a los objetivos y competencias de las asignaturas a las que está enfocado este trabajo.

En lo referente a TIA Portal V17, se podría crear una nueva estación que hiciera uso de un PLC de seguridad. Esto dotaría al alumno de conocimientos sobre el cableado y programación de las variables de entrada y salida relacionadas con la seguridad, así como la integración de los bloques de programa de seguridad con los del programa normal. Añadido, hay dos elementos relacionados con la programación de HMI's que no se han trabajado y son muy empleados en el mundo industrial, estos son el uso de recetas para la programación de ciertas operaciones y el uso de scripts.

En lo referente a programas de simulación, si bien se han empleado varios softwares de gemelos digitales de las estaciones, existen muchos otros. Un posible trabajo, que puede partir de la investigación y aplicación realizada en este proyecto, podría ser el estudio y comparación de más programas que realicen esta función.

Por último, cabe destacar que la selección del HMI empleado en las estaciones de la calculadora y la producción en línea no ha sido aleatoria, esto se debe a que la universidad dispone de un modelo TP700 Confort. Por lo tanto, una aplicación muy óptima de recursos sería emplear en la docencia estas dos estaciones, simulando su correcto funcionamiento primero con WinCC y posteriormente cargarlo y probarlo en el HMI real. Esta aplicación no supondría ningún coste añadido y dotaría al alumnado de una visión más completa de los elementos relacionados con la automatización industrial.



Universidad de Valladolid

Actualización, diseño y programación de varias estaciones de trabajo mediante el uso de PLC SIEMENS y softwares específicos



ESCUELA DE INGENIERÍAS  
INDUSTRIALES



## 9. BIBLIOGRAFÍA

- [1] J. L. Duran Moyano, H. Martínez García, J. Gámiz Caro, J. Domingo Peña, and A. Grau Saldes, *Automatismos eléctricos e industriales*. ALTAMAR S.A. y MARCOMBO, 2009.
- [2] U. Black, *Redes de ordenadores: protocolos, normas e interfaces*, 2nd ed. Rama, 1994.
- [3] Siemens AG, “Página web soporte técnico SIEMENS.” Accessed: Sep. 23, 2024. [Online]. Available: <https://support.industry.siemens.com/cs/start?lc=es-ES>
- [4] Siemens AG, “Página web oficial SIEMENS.” Accessed: Jan. 28, 2025. [Online]. Available: <https://www.siemens.com/global/en.html>
- [5] Siemens AG, “Controladores SIEMENS.” Accessed: Jan. 28, 2025. [Online]. Available: <https://www.siemens.com/global/en/products/automation/systems/industrial/controller-sw.html>
- [6] L. Moreno, S. Garrido, and C. Balaguer, *Ingeniería de control. Modelado y control de sistemas dinámicos*, 1st ed. Ariel Ciencia, 2003.
- [7] B. C. Kuo, *Sistemas de control automático*, 7th ed. Pearson Prentice Hall, 1996.
- [8] Siemens AG, “Enlace suministro TIA Portal V17.” Accessed: Sep. 26, 2024. [Online]. Available: <https://support.industry.siemens.com/cs/document/109784438/liberaci%C3%B3n-para-el-suministro-del-tia-portal-v17?dti=0&lc=es-ES>
- [9] Factory I/O, “Página web oficial de Factory IO.” Accessed: Jan. 28, 2025. [Online]. Available: <https://factoryio.com>
- [10] “Bloques de organización.” Accessed: Oct. 10, 2024. [Online]. Available: <https://instrumentationtools.com/main-cyclic-organization-block/>
- [11] “Información SIMIT.” Accessed: Jan. 28, 2025. [Online]. Available: <https://support.industry.siemens.com/cs/document/109746485/first-steps-with-simit-simulation-and-simatic-pcs-7?dti=0&dl=en&lc=fr-DZ>
- [12] “TFG actualizado.” Accessed: Jan. 28, 2025. [Online]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/23020/TFG-P-527.pdf?sequence=1&isAllowed=y>
- [13] Siemens AG., “MANUAL DE FUNCIONES SIMATIC S7-1200, S7-1500; Regulación PID,” 2023.
- [14] “Guías docentes.” Accessed: Jan. 28, 2025. [Online]. Available: <https://www.eii.uva.es/titulaciones/guiasDocentes/>
- [15] “Coste de luz medio en España.” Accessed: Jan. 28, 2025. [Online]. Available: <https://www.iberdrola.es/blog/luz/gasto-medio-de-luz-al-mes-dos-personas>



- [16] “Coste de red wifi medio en españa.” Accessed: Jan. 10, 2025. [Online]. Available: <https://www.adslzone.net/noticias/internet/cuanto-paga-comunidad-autonoma-internet/>