



## Control distribuido e integración SCADA

Clave(1095)

— Informe de proyecto N°1 —

# CONTROL Y SUPERVISIÓN DE SISTEMA SECUENCIAL ELECTRO-NEUMÁTICO

Apellidos y nombres:			
Grupo:		Profesor:	Fecha:
Semestre:	2021-2		EWS:

#### RÚBRICA DE EVALUACIÓN DE EXPOSICIÓN ORAL

CONCEPTOS, RUBROS O ASPECTOS A EVALUAR	BUENO (2 PUNTOS) Completo entendimiento del problema, realiza la actividad cumpliendo todos los requerimientos.	REGULAR (1 PUNTO) Bajo entendimiento del problema, realiza la actividad cumpliendo algunos de los requerimientos.	NO ACEPTABLE (0 PUNTOS) No demuestra entendimiento del problema o de la actividad.
1. Estructura y contenido	Presenta la información completa y sigue la secuencia lógica del tema	Presenta información parcial siguiendo la secuencia del tema	Presenta escasa información y carece de estructura
2. Cadencia	El volumen es adecuado y modula la voz de acuerdo con la situación	Expone el tema muy rápido sin hacer énfasis en ideas importantes	Habla demasiado bajo
3. Expresión oral	Emplea adecuadamente términos técnicos al referirse a los equipos, instrumentos y diagramas	Utiliza equivocadamente algunos términos	Hace uso de términos coloquiales frecuentemente
4. Dominio temático	Expone con claridad y argumenta cada etapa del diseño y desarrollo del proyecto correctamente	Responde parcialmente a las preguntas y no discute adecuadamente el tema	No explica las etapas del diseño y desarrollo
5. Expresión corporal	Establece contacto visual con toda la audiencia y hace uso adecuado del lenguaje corporal	Se desenvuelve con nerviosismo aunque emplea movimientos corporales de énfasis	Realiza movimientos que denotan inseguridad

ATRIBUTOS DEL EGRESADO A LOS QUE CONTRIBUYE INFORME DEL PROYECTO	NIVEL	Calificación
A3. Desarrollar y conducir experimentación adecuada; analizar e interpretar datos y utilizar el juicio ingenieril para establecer conclusiones. A4. Comunicarse eficazmente de manera oral y escrita sobre temas de ingeniería. A7. Trabajar en equipo con el objeto de diseñar, desarrollar, integrar, planear y poner en operación sistemas eléctricos y electrónicos, analizando riesgos.	Avanzado Avanzado Medio	

#### RÚBRICA DE EVALUACIÓN DEL INFORME DE PROYECTO

CONCEPTOS, RUBROS O ASPECTOS A EVALUAR	BUENO (2 PUNTOS) Completo entendimiento del problema, realiza la actividad cumpliendo todos los requerimientos.	REGULAR (1 PUNTO) Bajo entendimiento del problema, realiza la actividad cumpliendo algunos de los requerimientos.	NO ACEPTABLE (0 PUNTOS) No demuestra entendimiento del problema o de la actividad.
1. Planteamiento del proyecto	Plantea claramente el problema a resolver, los objetivos y establece el alcance del proyecto	Describe el problema y objetivos sin establecer el alcance	No formula el planteamiento del problema, objetivos ni alcance
2. Planeación de tareas	Realiza la planeación del proyecto siguiendo las recomendaciones del PMBOK y realiza el seguimiento del proyecto con un diagrama Gantt adecuado	Planea de parcialmente las tareas y actividades del proyecto y propone un diagrama Gantt con alguno errores	No planea las actividades ni realiza seguimiento del proyecto con el diagrama Gantt
3. Documentación técnica	Desarrolla correctamente diagramas de conexión al controlador empleando la simbología eléctrica IEC, hojas de especificaciones de instrumentos, tabla de ordenamiento de variables y galería fotográfica	Realiza los diagramas con algunos errores, no cumple con todos los requisitos o están incompletos	No entrega diagramas de conexión ni otra documentación técnica
4. Diseño y desarrollo	Diseña correctamente la rutina de control, código de rutinas de configuración, comunicación y el HMI con todas las especificaciones solicitadas	Se incluyen algunos códigos las rutinas y HMI con carencias u omisiones	No incluye rutina de control, comunicación, HMI o presenta inconsistencias
5. Evaluación, conclusiones y discusión	Presenta un funcionamiento correcto, reflexiona sobre las actividades, demuestra pensamiento crítico en el desarrollo del proyecto y aporta con conclusiones, discusión y recomendaciones	El funcionamiento no es del todo satisfactorio, las conclusiones generadas son únicamente un recuento de lo realizado y genera escasas observaciones	No demuestra un funcionamiento correcto ni genera conclusiones y observaciones

ATRIBUTOS DEL EGRESADO A LOS QUE CONTRIBUYE INFORME DEL PROYECTO	NIVEL	Calificación
A1. Identificar, formular y resolver problemas de ingeniería aplicando los principios de las ciencias básicas e ingeniería. A2. Aplicar, analizar y sintetizar procesos de diseño de ingeniería que resulten en proyectos que cumplen las necesidades especificadas. A3. Desarrollar y conducir experimentación adecuada; analizar e interpretar datos y utilizar el juicio ingenieril para establecer conclusiones. A7. Trabajar en equipo con el objeto de diseñar, desarrollar, integrar, planear y poner en operación sistemas eléctricos y electrónicos, analizando riesgos.	Intermedio Avanzado Avanzado Medio	

# Índice de Contenidos

<b>1. Planteamiento del problema</b>	<b>1</b>
1.1. Automatización del proceso . . . . .	1
1.2. Sistema de supervisión . . . . .	2
<b>2. Objetivos</b>	<b>2</b>
<b>3. Desarrollo del proyecto</b>	<b>2</b>
3.1. Planeación del proyecto según PMBOK . . . . .	2
3.1.1. Diagrama de Gantt . . . . .	2
3.2. Hojas de especificaciones . . . . .	4
3.3. Tabla de ordenamiento de variables . . . . .	5
3.4. Diagrama neumático . . . . .	5
3.5. Diagramas de conexión . . . . .	6
3.6. GRAFCET . . . . .	7
3.7. Rutina de control . . . . .	8
3.8. Código Delphi de rutina de comunicación y configuración . . . . .	16
3.9. Código Delphi de rutina de supervisión . . . . .	21
3.10. Código restante del proyecto . . . . .	23
3.11. HMI . . . . .	28
3.11.1. Botones <i>Start</i> , <i>Stop</i> y <i>Reset</i> . . . . .	29
3.11.2. Funcionamiento de pistones . . . . .	30
<b>4. Galería fotográfica</b>	<b>39</b>
<b>5. Aportaciones del grupo de trabajo</b>	<b>39</b>
<b>6. Conclusión</b>	<b>40</b>
<b>7. Discusión</b>	<b>40</b>
<b>8. Referencias</b>	<b>41</b>
<b>Anexo A. Hojas de datos</b>	<b>42</b>

# Índice de Figuras

1. Sistema neumático. . . . .	1
2. Secuencia para estaciones de ingeniería par. . . . .	1
3. Diagrama de Gantt. . . . .	3
4. Gant de seguimiento. . . . .	3
5. Tabla de ordenamiento de variables. . . . .	5
6. Diagrama neumático. . . . .	5
7. Diagrama de conexiones: entradas. . . . .	6
8. Diagrama de conexiones: salidas. . . . .	6

9.	GRAFCET. . . . .	7
10.	Rutina de control 1 de 8. . . . .	8
11.	Rutina de control 2 de 8. . . . .	9
12.	Rutina de control 3 de 8. . . . .	10
13.	Rutina de control 4 de 8. . . . .	11
14.	Rutina de control 5 de 8. . . . .	12
15.	Rutina de control 6 de 8. . . . .	13
16.	Rutina de control 7 de 8. . . . .	14
17.	Rutina de control 8 de 8. . . . .	15
18.	HMI. . . . .	28
19.	Secuencia de expulsión de pistones. . . . .	29
20.	Cilindro de doble efecto. . . . .	39
21.	Cilindro de retorno por muelle. . . . .	39
22.	Pistón. . . . .	39
23.	Led OFF. . . . .	39
24.	Led ON. . . . .	39
A.1.	Hoja de datos 1 de 6. . . . .	42
A.2.	Hoja de datos 2 de 6. . . . .	43
A.3.	Hoja de datos 3 de 6. . . . .	44
A.4.	Hoja de datos 4 de 6. . . . .	45
A.5.	Hoja de datos 5 de 6. . . . .	46
A.6.	Hoja de datos 6 de 6. . . . .	47

## Índice de Códigos

1.	Código Delphi de rutina de comunicación y configuración . . . . .	16
2.	Código Delphi de rutina de supervisión . . . . .	21
3.	Código Delphi restante del proyecto . . . . .	23
4.	Código botones <i>Start</i> , <i>Stop</i> y <i>Reset</i> . . . . .	29
5.	Código para la animación de los pistones . . . . .	31

# 1. Planteamiento del problema

Antes que nada se pretende desarrollar el proyecto siguiendo las recomendaciones del PMBOK, por lo que la planeación y el seguimiento del proyecto deberán realizarse utilizando la herramienta del Diagrama de Gantt.

## 1.1. Automatización del proceso

Un aspecto crucial es que el diseño de la rutina de control debe realizarse empleando la metodología GRAFCET, y para implementarlo en el programa, la rutina deberá traducirse al lenguaje LD IEC 61131-3.

El laboratorio de Automatización cuenta con un sistema de pistones neumáticos como el que se observa en la Figura 1, se desea automatizar dicho sistema a través de un controlador SLC 5/04 de la marca Allen Bradley.

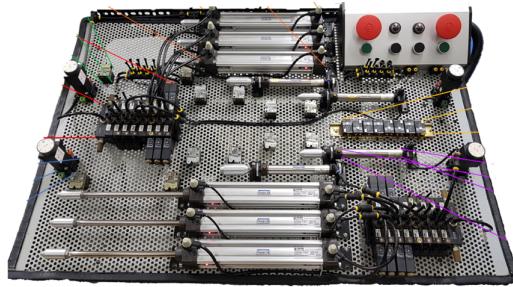


Figura 1: Sistema neumático.

Este sistema neumático está dividido en dos secciones que son idénticas, esto con el objetivo de que 2 alumnos puedan trabajar en la plataforma al mismo tiempo. Se dejaron 2 secuencias, la primera para estaciones de ingeniería impar y la segunda para estaciones de ingeniería par, en mi caso me tocó esta última, que se traduce en trabajar con la sección de abajo de la plataforma, la cual comprende los pistones P1B a PB5, tal secuencia se observa en la Figura 2.

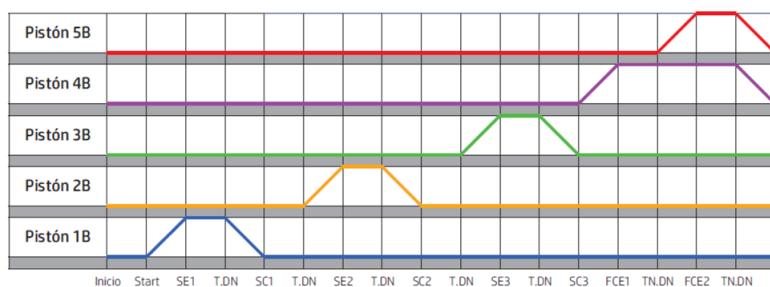


Figura 2: Secuencia para estaciones de ingeniería par.

En total son 5 pistones, 3 son de doble efecto y 2 de retorno por muelle, como se observa en la Figura 2, cada vástago se expulsa una vez de forma secuencial. Únicamente los 3 pistones de doble efecto serán activados por electroválvulas comandadas por el PLC, los otros 2 pistones son meramente

neumáticos y no tienen interacción alguna con el PLC.

Resumiendo, el PLC tendrá como entradas los sensores de contracción y expulsión de los primeros 3 pistones (de doble efecto) y tendrá también como entradas 2 botones físicos: *Start* y *Stop*, que al presionarse al mismo tiempo indicaran un *Reset* al sistema. Como salidas se tendrán las electrováulas que expulsaran y contraerán a los 3 pistones de doble efecto.

## 1.2. Sistema de supervisión

Una vez que la automatización del sistema sea validada, se debe crear un sistema de supervisión del proceso en el software Delphi, el cual debe tener capacidad de comunicación OPC para comunicarse con el proceso que esta controlando el PLC. En esta pantalla se debe observar la animación de la expulsión y contracción de los pistones que deben estar sincronizados con la plataforma real. Se debe observar además el prendido y apagado de los sensores de contracción y expulsión.

La pantalla deberá incluir 3 botones: *Start*, *Stop* y *Reset*. Por default, la secuencia debe repetirse tres veces y detenerse, sin embargo, debe ser posible la configuración del número de repeticiones de esta, para iniciar nuevamente la secuencia se debe presionar *Start*, otro detalle es que se debe desplegar el número de secuencias que se llevan. Por último, el tiempo de espera y retorno de cada vástago deberá ser configurable desde el sistema de supervisión.

Una vez concluido el proyecto, se debe realizar una presentación del resultado del producto, el cual debe funcionar y se debe presentar como si se le estuviera vendiendo a un cliente.

## 2. Objetivos

El presente proyecto tiene como objetivos:

- Automatización del sistema de pistones neumáticos mediante el controlador SLC 5/04.
- Implementar el control utilizando la metodología GRAFCET en LD IEC 61131-3.
- Desarrollo de sistema de supervisión desarrollado en Delphi RAD Studio.
- Administrar el proyecto siguiendo las recomendaciones del PMBOK.
- Realizar una presentación en donde se exponga el funcionamiento.

## 3. Desarrollo del proyecto

### 3.1. Planeación del proyecto según PMBOK

#### 3.1.1. Diagrama de Gantt

En la Figura 3 se pueden observar las actividades que se idearon para la realización del proyecto, se puede observar que cada una de estas se realizaron secuencialmente, es decir que primero debía acabarse una para continuar con la otra.

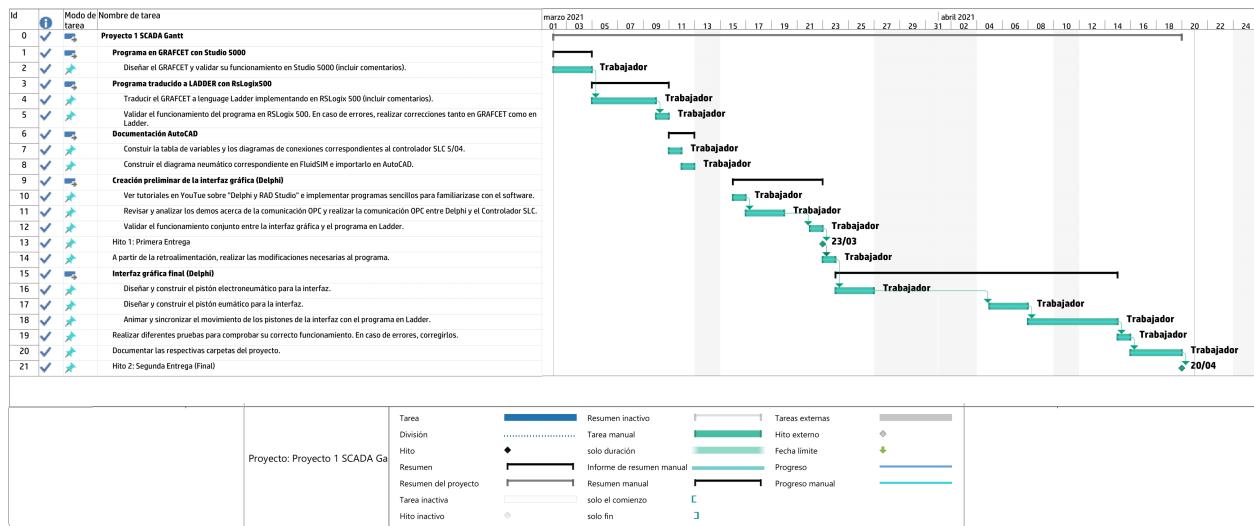


Figura 3: Diagrama de Gantt.



Figura 4: Gantt de seguimiento.

El proyecto se dividió en 5 fases:

- **Programa en GRAFCET con Studio 5000:** Se aprovecho que se contaba con este software, y se diseñó la rutina GRAFCET en este, esto fue de gran ayuda, pues mediante el simulador que posee el mismo software se valido que el programa funcionara correctamente.
- **Programa traducido a LADDER con RSLogix500:** Una vez validado el GRAFCET, se realizó la traducción en lenguaje LADDER y de igual forma se comprobó su correcto funcionamiento, que en un principio parecía funcionar bien, sin embargo, se encontró un ligero error posteriormente cuando se estaba trabajando con la interfaz gráfica.

- **Documentación AutoCAD:** Se realizaron los diagramas de conexiones de entradas y salidas al controlador. Lo mismo para la tabla de ordenamiento de variables para la cual fue fundamental tener prácticamente terminada la rutina de control final. Se construyó de igual forma el diagrama neumático correspondiente meramente a la parte neumática del proceso, para esto se hizo uso del software *FluidSIM*.
- **Creación preliminar de la interfaz gráfica:** Para el Hito 1, fue fundamental entregar la comunicación via OPC entre el controlador y la interfaz gráfica. Fue una de las cosas más complicadas de conseguir, pues no tenía experiencia con el entorno de desarrollo Delphi RAD Studio, por lo que me di a la tarea de primero conocer este y practicar el desarrollo de interfaces simples para ganar experiencia y confianza. Después de esto me informe acerca de la comunicación OPC y con ayuda de las clases de laboratorio conseguí realizar la comunicación solicitada.

En esta interfaz preliminar coloque unos leds (imágenes) para los sensores de contracción y expulsión y para las electroválvulas, fue ahí donde me di cuenta de que mis electroválvulas parpadeaban en lugar de permanecer fijas, lo cual evidencio un defecto en mi rutina de control, tardé un tiempo en encontrarlo, este consistía en que utilizaba para las electroválvulas varios *Latch* y *Unlatch* en varios lados del programa, una vez identificado esto, no fue problema darle solución a este.

- **Interfaz gráfica final (Delphi):** Al no contar con mucha experiencia con softwares de dibujo, antes de empezar a dibujar, primero me di a la tarea de identificar las cuestiones que debía considerar, más que nada lo importante era tener presente que la imagen debía contar con un fondo transparente para que no se viera el fondo blanco que usualmente traen las imágenes, lo cual le restaría lo atractivo a la interfaz gráfica. Me di cuenta que necesitaba 3 imágenes: El cuerpo del cilindro, el émbolo del cilindro, y 1 led para los sensores de contracción y expulsión. Finalmente me di cuenta de que podía recortar las fotos de los cilindros de la plataforma neumática real, de esta forma me ahorré el dibujo de estos.

El proceso del desarrollo del sistema de supervisión fue más tardado de lo que pensaba en un principio, pues había muchos detalles los cuales había que considerar para que la HMI funcionara sin ningún inconveniente.

### 3.2. Hojas de especificaciones

Al final del documento, específicamente en el apéndice A se añaden las hojas de especificaciones pertinentes respecto al equipo neumático del cual esta constituido la plataforma neumática.

### 3.3. Tabla de ordenamiento de variables

CREADO CON UNA VERSIÓN PARA ESTUDIANTES DE AUTODESK									
ENTRADAS				SALIDAS EN MEMORIA					
DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN	COMENTARIO	DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN	DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN
I/18	Start	Pulsador de arranque	Normalmente Abierto	B2/0	E0	Eje 0	E2/10	Start, Secundo	Pulsador de interfaz
I/19	Stop	Pulsador de paro	Normalmente Cerrado	B2/1	E1	Eje 1	E2/11	Stop, Remoto	Pulsador de interfaz
I/10	SC4	Sensor expandido 4	Para el Piso 38	B2/2	E2	Eje 2	E2/12	Reset, Remoto	Pulsador de interfaz
I/11	SI4	Sensor contráido 4	Para el Piso 38	B2/3	E3	Eje 3	E2/13	Reset, Físico	Start & Stop físicos activados si se hace clic
I/12	SC5	Sensor expandido 5	Para el Piso 28	B2/4	E4	Eje 4	E2/14	Auxiliar0	Proceso Activo
I/13	SI5	Sensor contráido 5	Para el Piso 28	B2/5	E5	Eje 5	E2/15	Auxiliar1	Para controlar los planteles en "VEZ"
I/14	SC6	Sensor expandido 6	Para el Piso 18	B2/6	E6	Eje 6	E2/16	Auxiliar2	Estimado el T4.0
I/15	SI6	Sensor contráido 6	Para el Piso 18	B2/7	E7	Eje 7	E2/17	Auxiliar3	Estimado el T4.1
				B2/8	E8	Eje 8	E2/18	Auxiliar4	Estimado el C5.0
				B2/9	E9	Eje 9	E2/19	SC4&_Act	3 de Activadas
				B2/10	E10	Eje 10	E2/20	Start	Start Global
				B2/11	E11	Eje 11	E2/21	Stop	Stop Global
				B2/12	E12	Eje 12	E2/22	Reset	Reset Global
OTROS									
DIRECCIÓN	SÍMBOLO	DESCRIPCIÓN	COMENTARIO						
T4.0	T1								Temperatura 1
T4.1	T2								Temperatura 2
C5.0	C1								Contador de secundas

UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO  
FACULTAD DE INGENIERÍA

PROYECTO: CONTROL Y SUPERVISIÓN DE SISTEMA ELECTRO - NEUMÁTICO  
TIPO TABLA DE ORDENAMIENTO DE VARIABLES  
MATERIAL CONTROL DISTRIBUIDO E INTERACCIÓN SACADA  
PANEL: ALUMNO: PROFESSION: FECHA: 23/03/2021  
MODULOS: 1 EJE: 3 ALUMNO: RAFAEL MUÑOZ MUÑOZ  
CONTRALICENCIA: JUAN RAYMUND  
PROFESOR: DR. MIGUEL ÁNGEL MUÑOZ MUÑOZ SEMESTRE: 2021-2

CREADO CON UNA VERSIÓN PARA ESTUDIANTES DE AUTODESK

DETALLE DE REVISIÓN: 0 23/03/2021 PRIMERA ENTREGA REV: FECHA:

Figura 5: Tabla de ordenamiento de variables.

### 3.4. Diagrama neumático

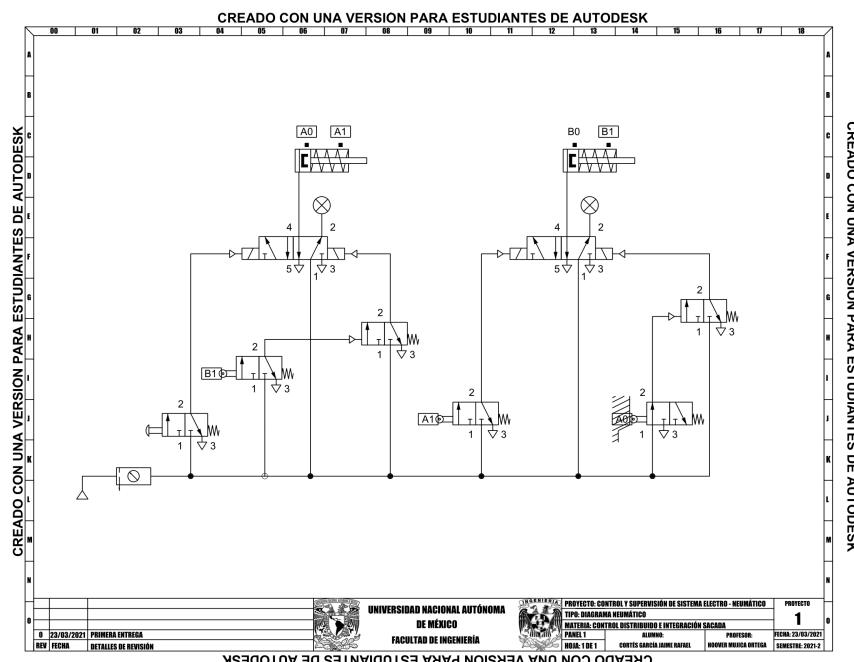


Figura 6: Diagrama neumático.

### 3.5. Diagramas de conexión

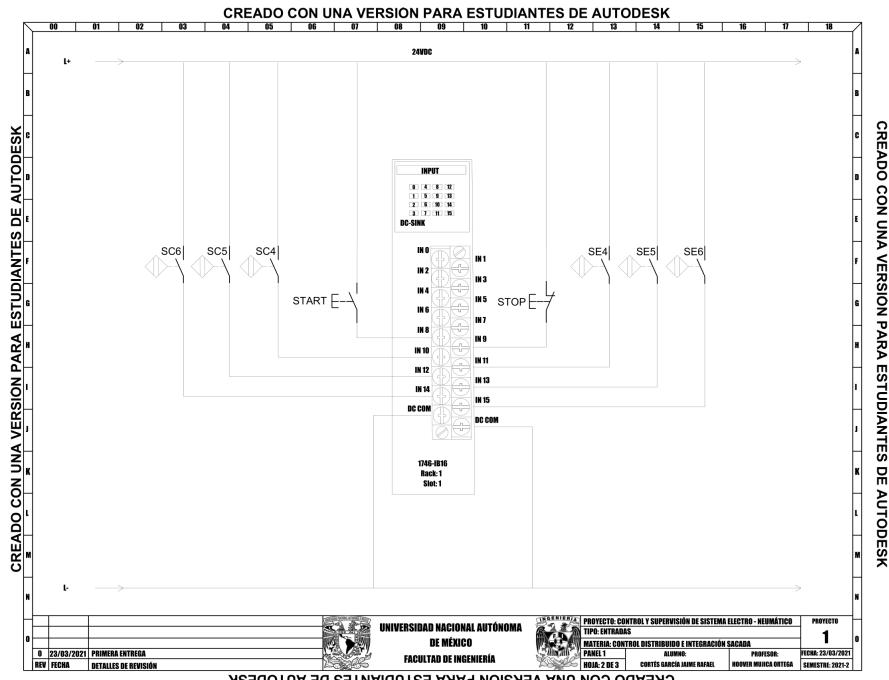


Figura 7: Diagrama de conexiones: entradas.

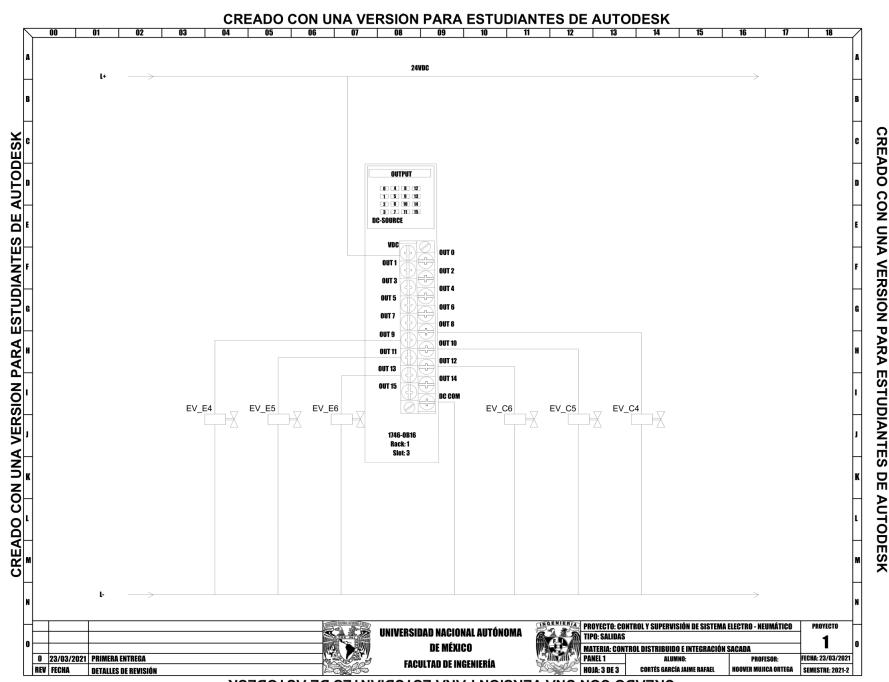


Figura 8: Diagrama de conexiones: salidas.

### 3.6. GRAFCET

En total el diseño del GRAFCET constituyó en 13 estados como se puede observar en la Figura 9, los nombres de las variables que aparecen en esta son idénticos a la rutina en LADDER, lo cual es un buen indicio de una correcta traducción de GRAFCET a LADDER.

Se utilizaron en total 2 temporizadores, el primero para el tiempo de espera para los 3 cilindros de doble efecto (accionadas por las electroválvulas) y el segundo para el tiempo de la parte neumática. Se utilizó un contador el cual tiene como propósito contar el número de secuencias que se llevan.

El *Reset* se incorpora 5 veces en el GRAFCET, es decir en 5 partes de este se pregunta por el estado del *Reset*, y si este está activado, se regresa al estado 0, en donde se resetean todas las variables a sus valores predeterminados.

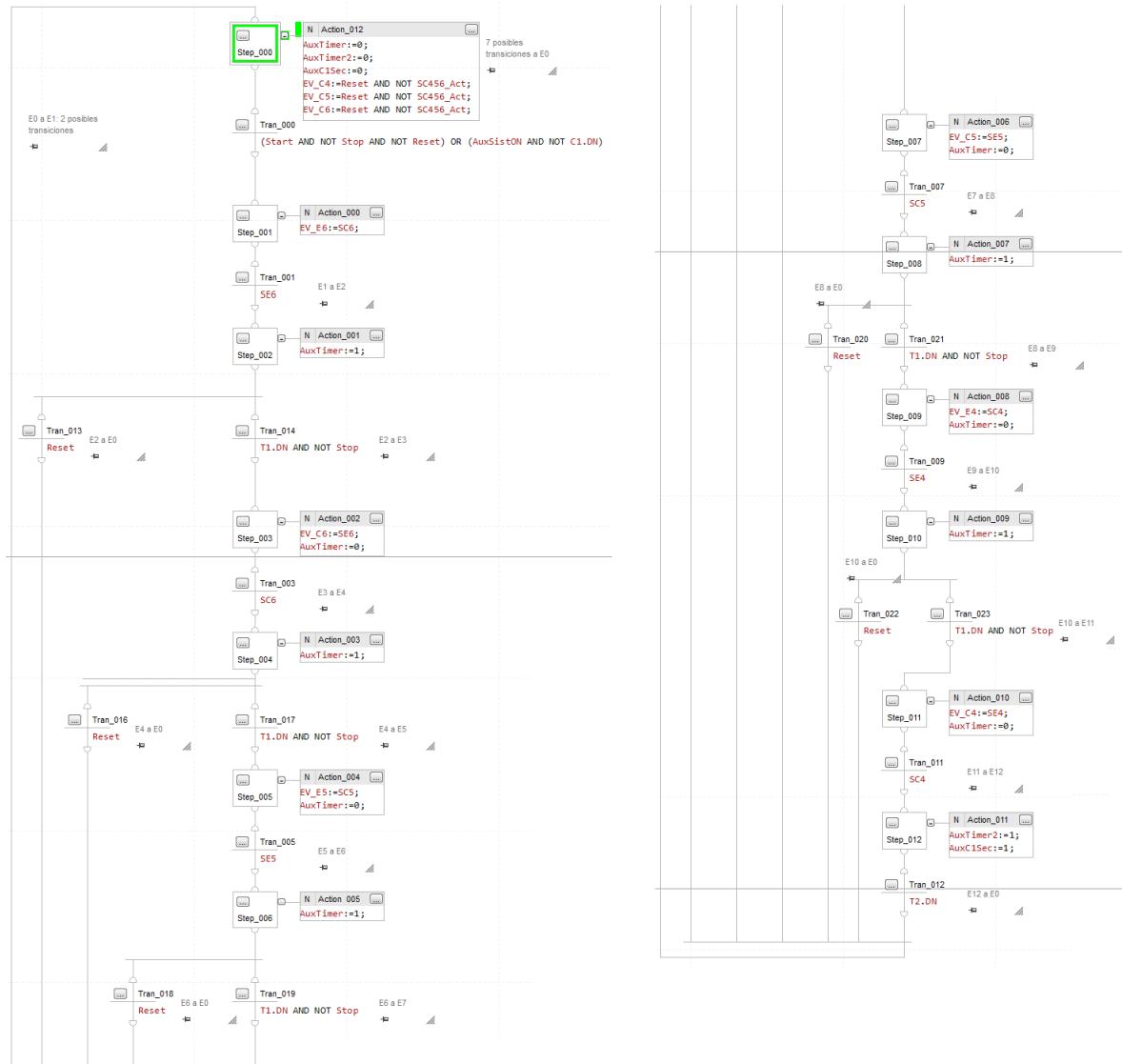


Figura 9: GRAFCET.

### 3.7. Rutina de control

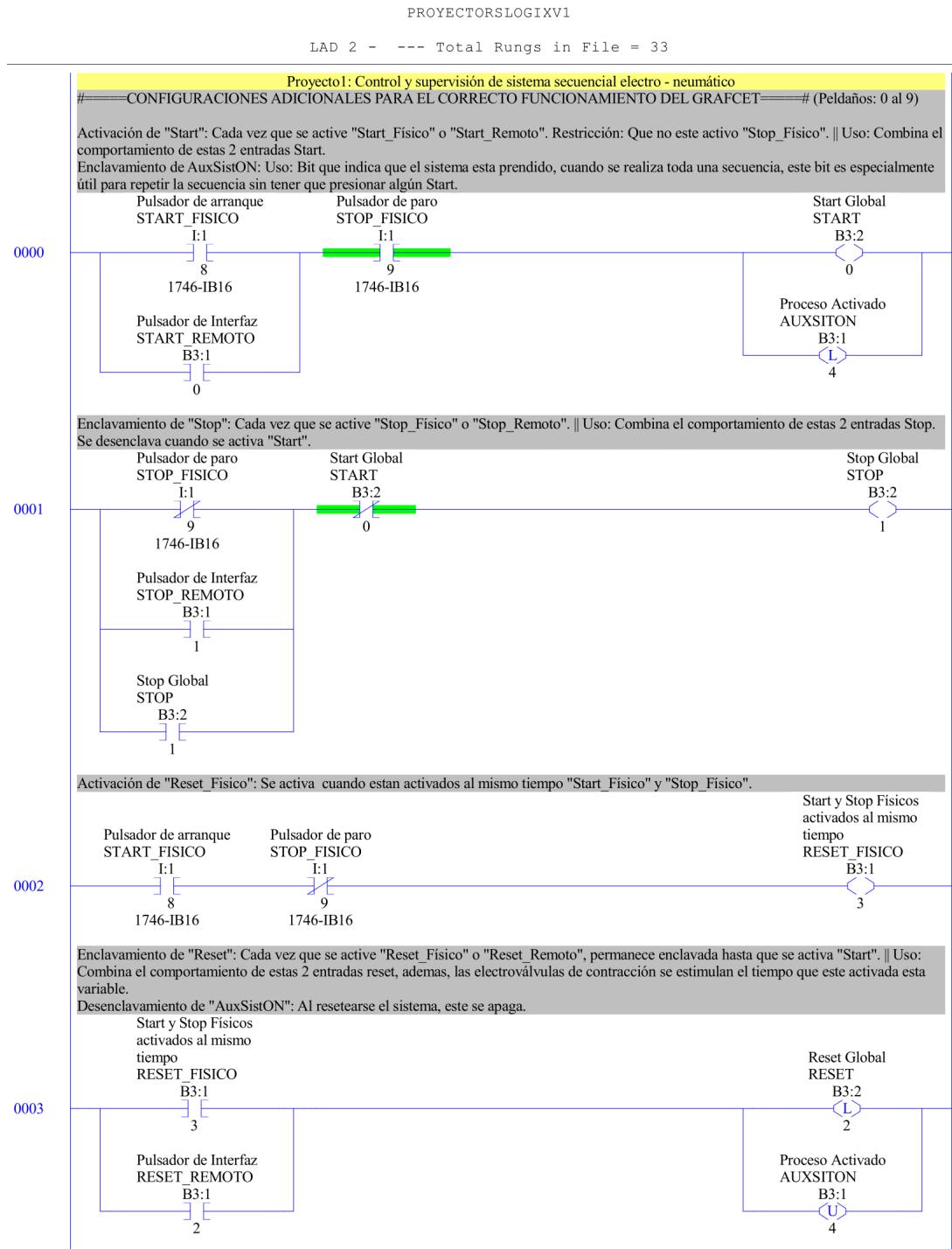


Figura 10: Rutina de control 1 de 8.

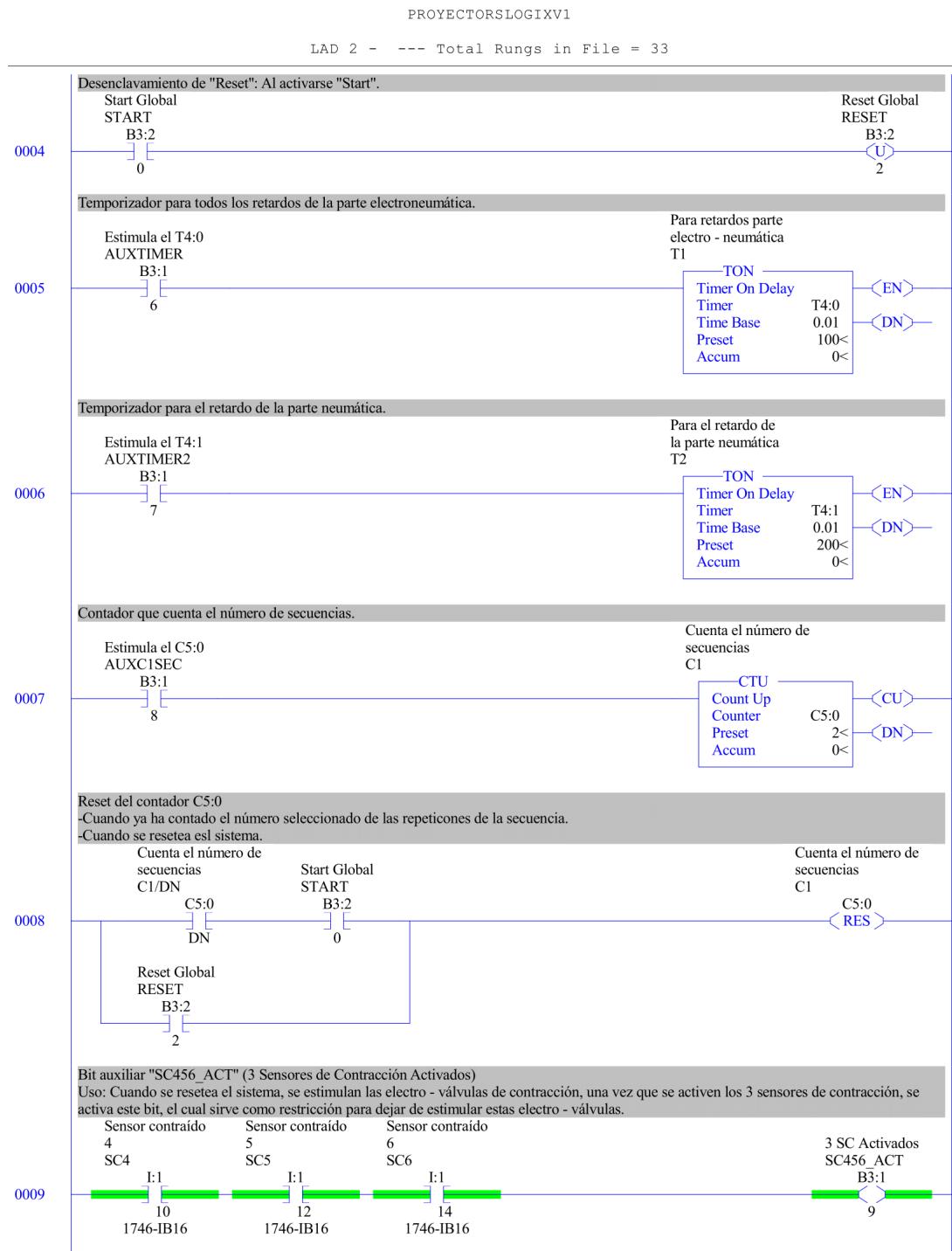


Figura 11: Rutina de control 2 de 8.

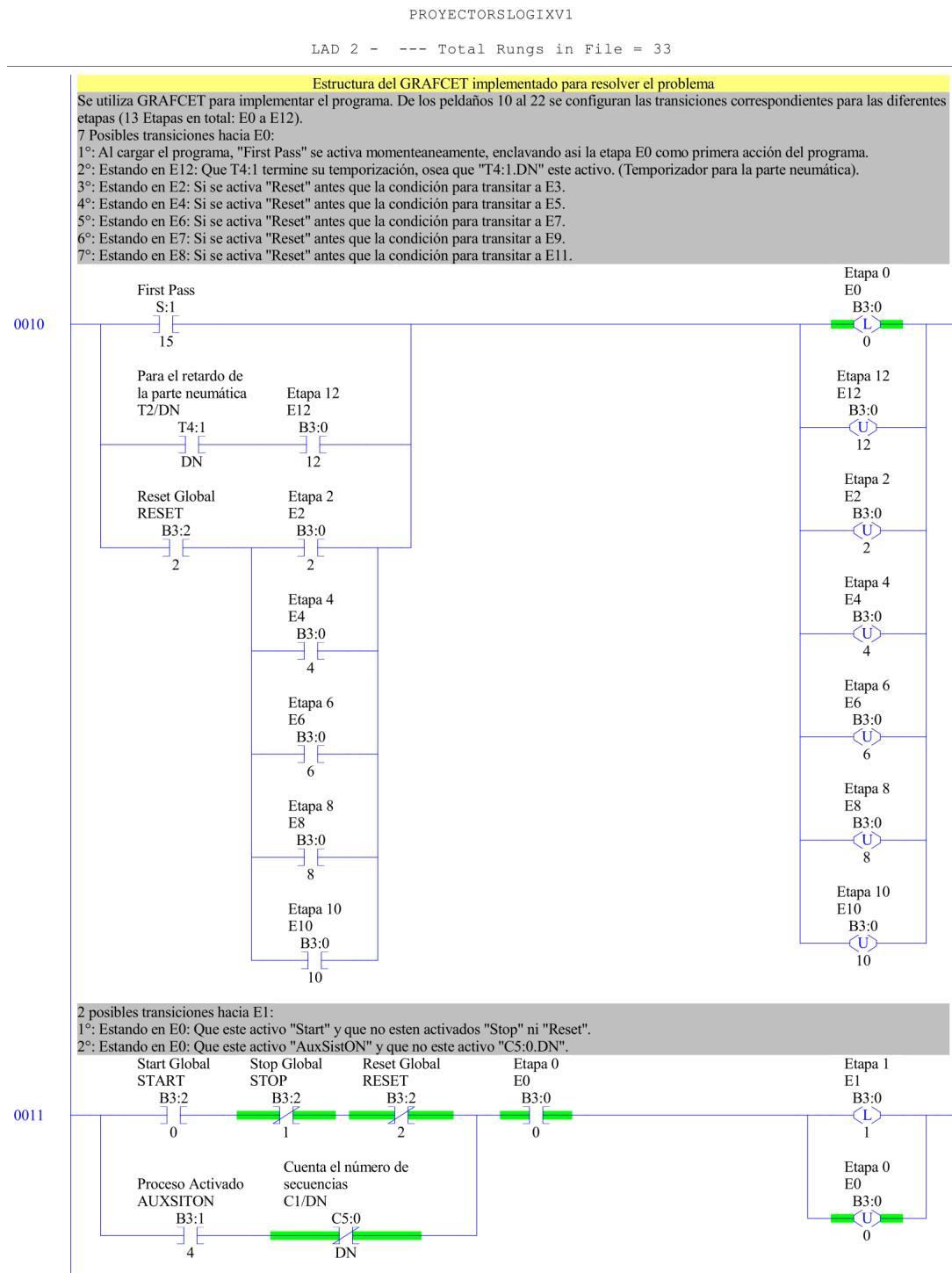


Figura 12: Rutina de control 3 de 8.

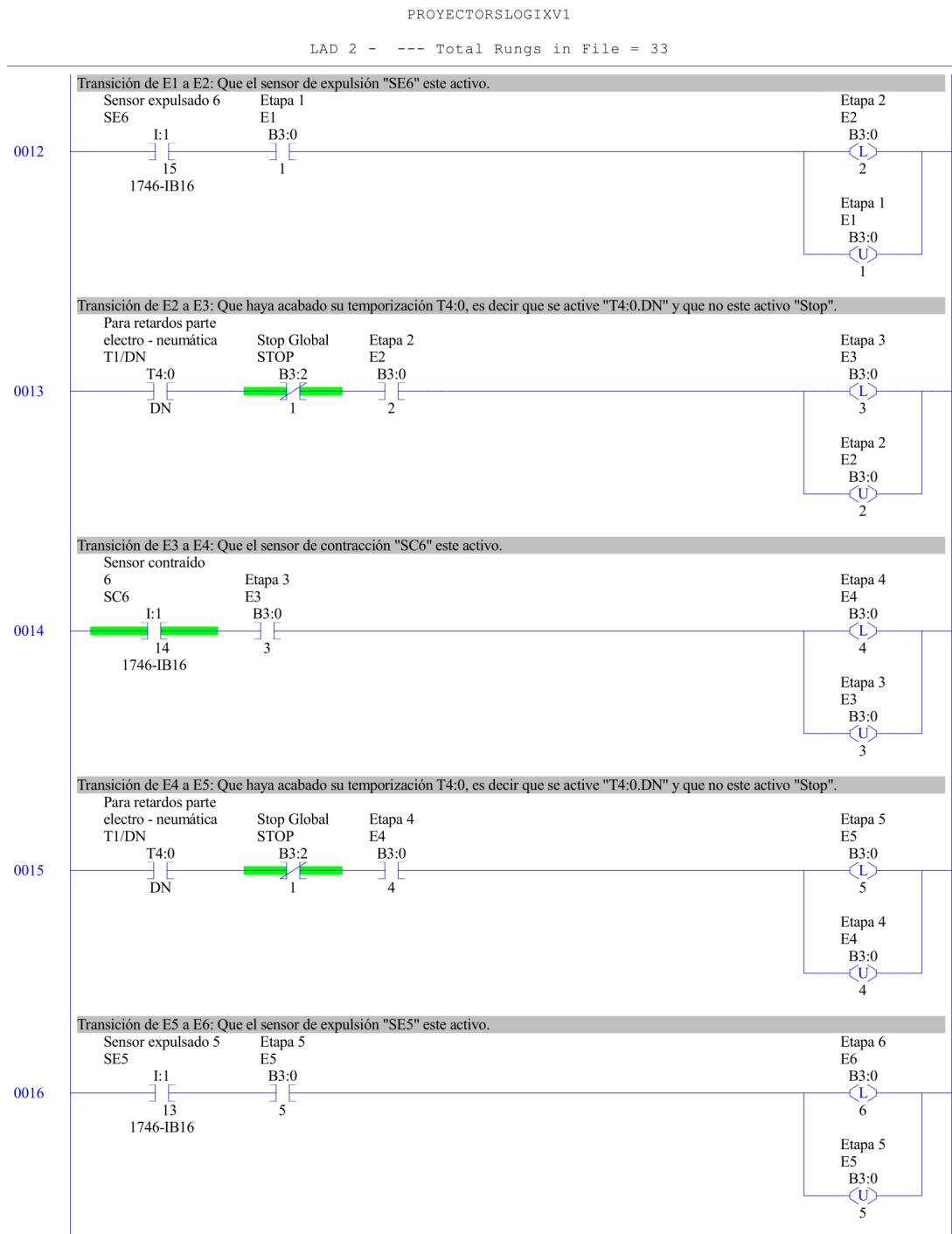


Figura 13: Rutina de control 4 de 8.

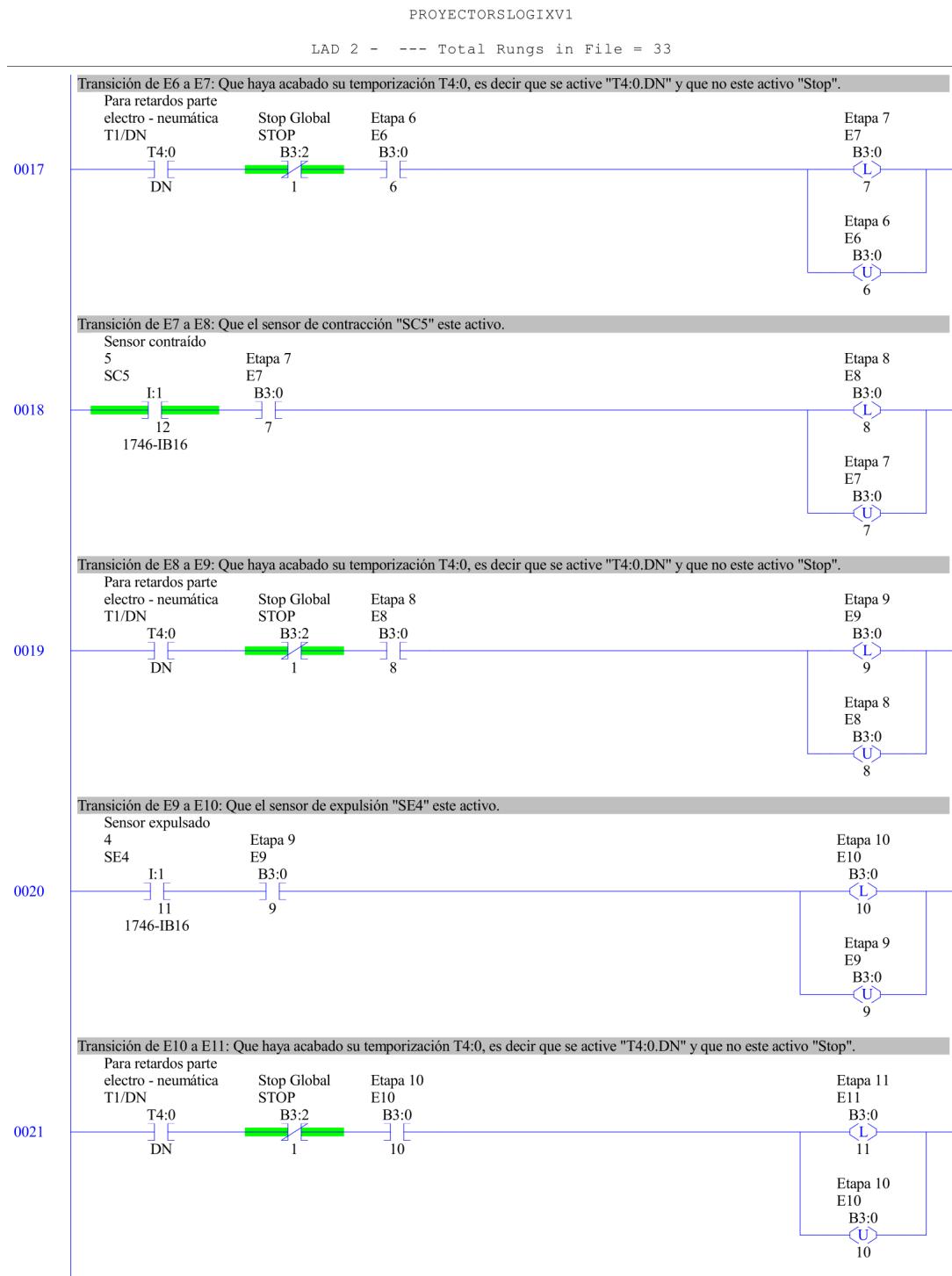


Figura 14: Rutina de control 5 de 8.

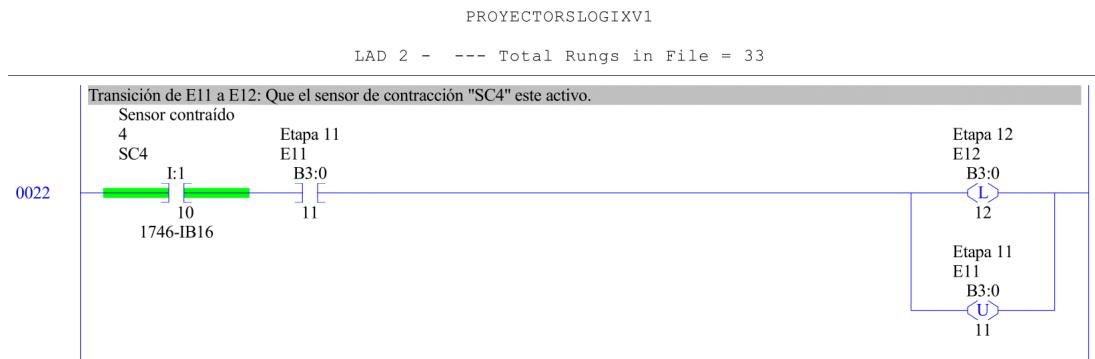


Figura 15: Rutina de control 6 de 8.

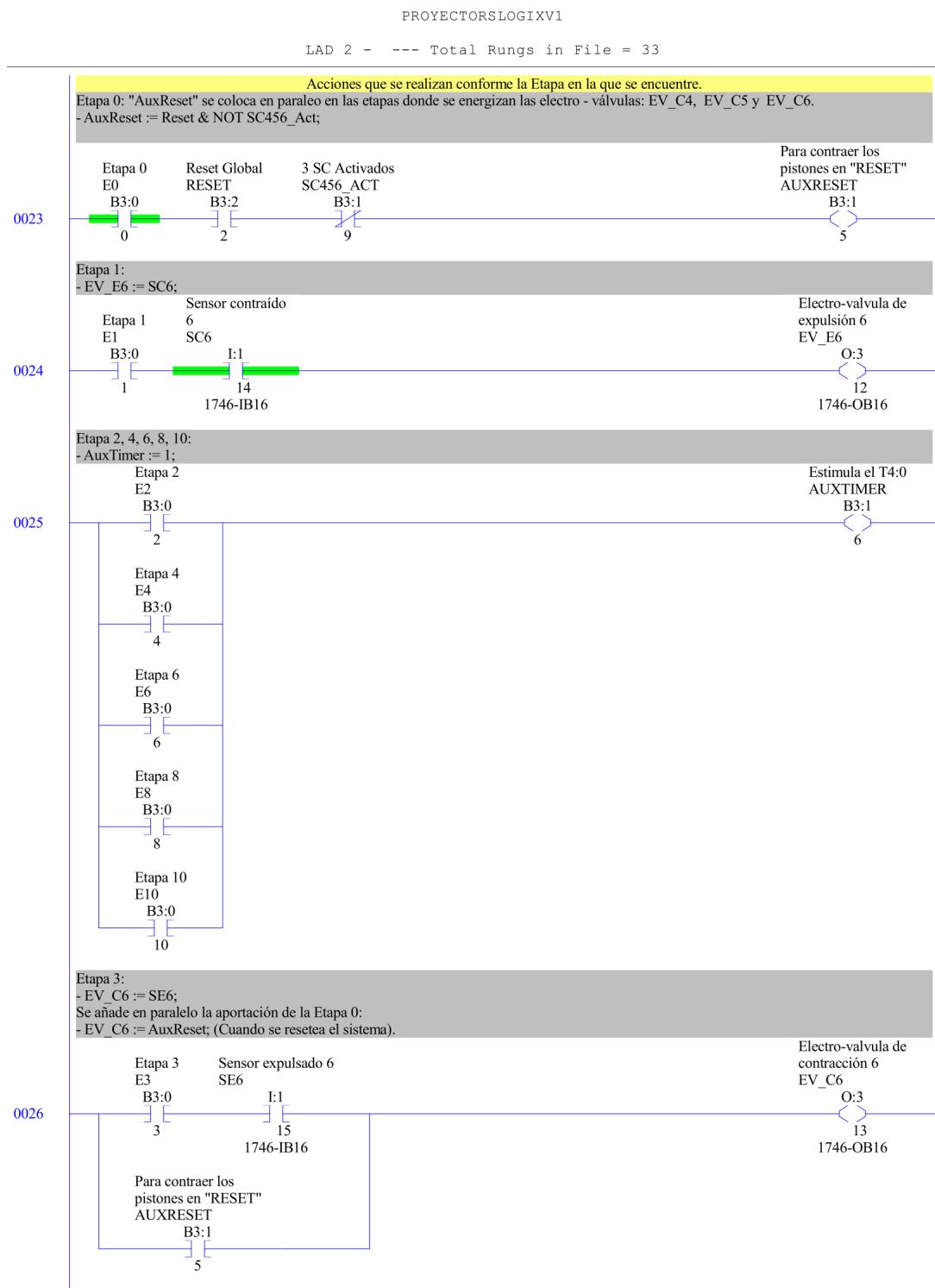


Figura 16: Rutina de control 7 de 8.

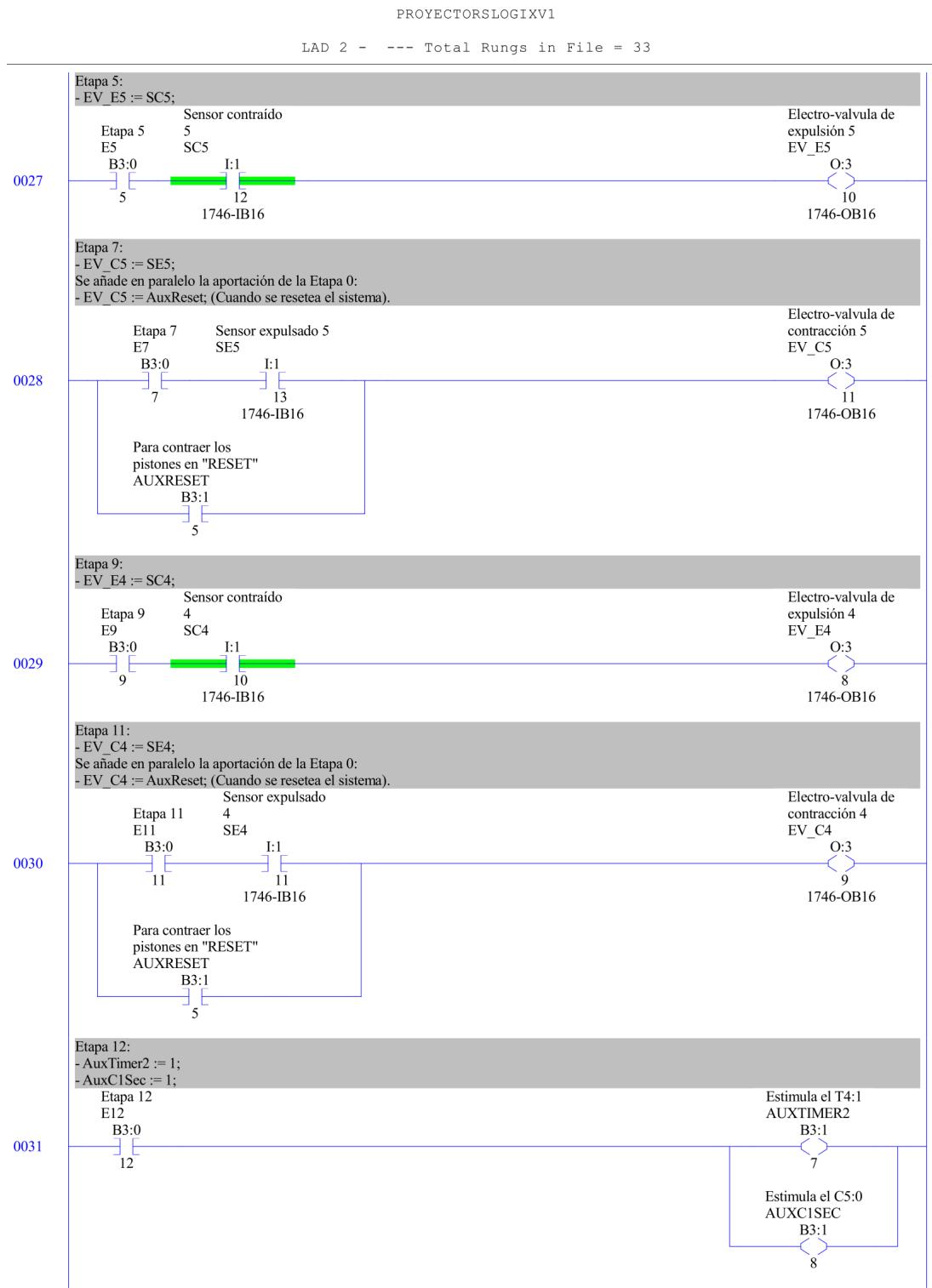


Figura 17: Rutina de control 8 de 8.

### 3.8. Código Delphi de rutina de comunicación y configuración

Código 1: Código Delphi de rutina de comunicación y configuración

```

1 //===== COMBOBOX "ServerCombo" ======
2 procedure TshSC5.ServerComboDropDown(Sender: TObject);
3 begin
4   if Trim(ServerCombo.Items.Text) = "" then //Enlista los servidores disponibles
5   begin
6     Screen.Cursor := crHourGlass;
7     GetOPCServers(ServerCombo.Items);
8     Screen.Cursor := crDefault;
9     ConecDesconec.Enabled := true; //Se habilita el botón "ConecDesconec"
10    end;
11  end;
12 //===== FIN COMOBOX "ServerCombo" ======
13
14 //===== CODIGO RELACIONADO CON EL "ComboTopico" ======
15 procedure Topicos(Browser: TdOPCBrowser; ItemList: TStrings; Level: integer = 0);
16
17 var
18   i      : integer;
19   Items  : TdOPCBrowseItems;
20   BrowseItem : TdOPCBrowseItem;
21   LevelStr : string;
22 begin
23   Browser.Browse;           //Se busca todo acerca del OPC Server
24   Items := TdOPCBrowseItems.Create; //Se crea una lista de los items
25   Items.Assign(Browser.Items); //Se asigna en lista
26
27   for i := 0 to Level do
28     ItemList.Add(Browser.CurrentPosition.Name); //Se muestra el nombre de la lista
29
30   for i := 0 to Items.Count-1 do //Se recorre para todos los items
31   begin
32     BrowseItem := Items[i];
33     if BrowseItem.IsFolder then //Se añaden los items a la lista
34       ItemList.Add(BrowseItem.ItemId)
35   end;
36 end;
37
38 procedure TshSC5.ComboTopicoDropDown(Sender: TObject);
39 var
40   Items : TdOPCBrowseItems;
41 begin
42   ComboTopico.Items.Clear;          //Limpia los items actuales
43   Topicos(dOPCServer1.Browser,ComboTopico.Items); //Llama a la función "Topicos"
44   ComboTopico.Items.Delete(0);
45 end;

```

```

46
47 procedure TshSC5.ComboTopicoChange(Sender: TObject);
48 begin
49   TAGS := dOPCServer1.OPCGroups.Add('TAGS');
50   //Definiendo los items que pertenecen al grupo TAGS
51   //Entradas
52   STARTV := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]B3:1/0'); //Start Virtual
53   STOPV := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]B3:1/1'); //Stop Virtual
54   RESETV := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]B3:1/2'); //Reset Virtual
55
56   //CLLINDRO NEUMATICO 6: Sensores y electrovalvulas de contraccion y expulsion
57   SC6 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/14');
58   SE6 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/15');
59   EV_C6 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/13');
60   EV_E6 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/12');

61   //CLLINDRO NEUMATICO 5: Sensores y electrovalvulas de contraccion y expulsion
62   SC5 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/12');
63   SE5 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/13');
64   EV_C5 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/11');
65   EV_E5 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/10');

66   //CLLINDRO NEUMATICO 4: Sensores y electrovalvulas de contraccion y expulsion
67   SC4 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/10');
68   SE4 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]I:1/11');
69   EV_C4 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/9');
70   EV_E4 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]O:3/8');

71   //TIMERS Y CONTADOR (Presets)
72   PrT1 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]T4:0.PRE');
73   PrT2 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]T4:1.PRE');
74   PrC1 := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]C5:0.PRE');

75   //RESET Fisico
76   ResFisic := TAGS.OPCItems.AddItem('P+ComboTopico.Text+]B3:1/3');

77   //Se habilitan y desabilitan los botones:
78   ComboTopico.Enabled := false;
79
80   bCargConf.Enabled := true;
81   bStart.Enabled := true;
82   bStop.Enabled := true;
83   bReset.Enabled := true;

84   Config := true;
85   auxNeum := false;
86   auxDecim := 0;
87   i := 0;
88
89   PrT1.WriteString(StrToInt(edTiempEspEV6.Text)*100); //Escribe el Preset en el T1
90   PrT2.WriteString(StrToInt(edTiemNeum.Text)*100); //Escribe el Preset en el T2
91   PrC1.WriteString(StrToInt(edNoSec.Text)); //Escribe el Preset en el C1

```

```
97   Timer1.Enabled := true; // Se activa el timer1
98 end;
99 //===== FIN CODIGO PARA EL "ComboTopico" ======
100
101 //===== BOTON "ConecDesconec" ======
102 procedure TshSC5.ConecDesconecClick(Sender: TObject);
103 begin
104   if dOPCServer1.Active=false then //Si la conexión esta inactiva:
105     begin
106       //Funciones
107       dOPCServer1.ServerName := ServerCombo.Text; //Nombre leido del combobox
108       dOPCServer1.Active := true; //Se activa la conexión
109       //Apariencia
110       ConecDesconec.Caption := 'Desconectar'; //Se cambia el texto del botón
111       ServerCombo.Enabled := false; //Desabilita comobox de servidor
112       ComboTopico.Enabled := true; //Habilita combobox de tópicos
113     end
114
115   else if dOPCServer1.Active=true then //Si la conexión esta activa:
116     begin
117       //Funciones
118       dOPCServer1.Active := false; //Desactiva conexión
119       dOPCServer1.OPCGroups.RemoveAll; //Borra todos los grupos
120       //Apariencia
121       ConecDesconec.Caption := 'Conectar'; //Cambiar texto del botón
122       ComboTopico.Enabled := false; //Desactiva combo box de tópicos
123       ServerCombo.Enabled := true; //Habilita combo box de servidores
124       bStart.Enabled := false; //Deshabilita botón Start
125       bStop.Enabled := false; //Deshabilita botón Stop
126       bReset.Enabled := false; //Deshabilita botón Reset
127       ComboTopico.Text := ''; //Limpia el texto de ComboTopico
128
129       //Se deshabilitan los edits, botones e imágenes de sensores
130       edTiempEspEV6.Enabled := false;
131       edTiempEspEV5.Enabled := false;
132       edTiempEspEV4.Enabled := false;
133       edTiempExpRetEV6.Enabled := false;
134       edTiempExpRetEV5.Enabled := false;
135       edTiempExpRetEV4.Enabled := false;
136       edNoSec.Enabled := false;
137       edTiemNeum.Enabled := false;
138       bCargConf.Enabled := false;
139       imSE4.Visible := false;
140       imSE5.Visible := false;
141       imSE6.Visible := false;
142       imSC4.Visible := false;
143       imSC5.Visible := false;
144       imSC6.Visible := false;
145
146       Timer1.Enabled := false; //Deshabilita el timer1
147     end;
```

```
148 end;
149 //===== FIN BOTON "ConecDesconec" ======
150
151 //===== ESCRITURA ======
152 // BOTÓN "Cargar Configuración" :
153 procedure TshSC5.bCargConfMouseDown(Sender: TObject; Button: TMouseButton;
154 Shift: TShiftState; X, Y: Integer);
155 begin
156 if Config then //SE HABILITA LA CONFIGURACIÓN
157 begin
158 bCargConf.Caption := 'Cargar configuración';
159 //Se habilitan los edits en los que el usuario ingresa parametros
160 edTiempEspEV6.Enabled := true;
161 edTiempEspEV5.Enabled := true;
162 edTiempEspEV4.Enabled := true;
163 edTiempExpRetEV6.Enabled := true;
164 edTiempExpRetEV5.Enabled := true;
165 edTiempExpRetEV4.Enabled := true;
166 edNoSec.Enabled := true;
167 edTiemNeum.Enabled := true;
168
169 Config := false; //Los nuevos parametros no se han cargado
170 end
171 else //SE DESHABILITA LA CONFIGURACIÓN
172 begin
173 //Restricción para tiempo de espera de EV6
174 if (edTiempEspEV6.Text = '0') or (edTiempEspEV6.Text = '') then
175 begin //Si se sale del rango del tiempo, se auto
176 edTiempEspEV6.Text := '1'; //configura en sus respectivos limites
177 end //Rango: [1-327]seg.
178 else if (StrToInt(edTiempEspEV6.Text) > 327) then
179 begin
180 edTiempEspEV6.Text := '327';
181 end;
182
183 //Restricción para tiempo de expulsión/retorno de EV6
184 if (edTiempExpRetEV6.Text = '0') or (edTiempExpRetEV6.Text = '') then
185 begin //Si se sale del rango del tiempo, se auto
186 edTiempExpRetEV6.Text := '1'; //configura en sus respectivos limites
187 end //Rango: [1-327]seg.
188 else if (StrToInt(edTiempExpRetEV6.Text) > 327) then
189 begin
190 edTiempExpRetEV6.Text := '327';
191 end;
192
193 //Restricción para tiempo de espera de EV5
194 if (edTiempEspEV5.Text = '0') or (edTiempEspEV5.Text = '') then
195 begin //Si se sale del rango del tiempo, se auto
196 edTiempEspEV5.Text := '1'; //configura en sus respectivos limites
197 end //Rango: [1-327]seg.
198 else if (StrToInt(edTiempEspEV5.Text) > 327) then
```

```
199 begin
200     edTiempEspEV5.Text := '327';
201 end;
202
203 //Restricción para tiempo de expulsión/retorno de EV5
204 if (edTiempExpRetEV5.Text = '0') or (edTiempExpRetEV5.Text = '') then
205 begin
206     //Si se sale del rango del tiempo, se auto
207     edTiempExpRetEV5.Text := '1'; //configura en sus respectivos limites
208 end
209 //Rango: [1-327]seg.
210 else if (StrToInt(edTiempExpRetEV5.Text) > 327) then
211 begin
212     edTiempExpRetEV5.Text := '327';
213 end;
214
215 //Restricción para tiempo de espera de EV
216 if (edTiempEspEV4.Text = '0') or (edTiempEspEV4.Text = '') then
217 begin
218     //Si se sale del rango del tiempo, se auto
219     edTiempEspEV4.Text := '1'; //configura en sus respectivos limites
220 end
221 //Rango: [1-327]seg.
222 else if (StrToInt(edTiempEspEV4.Text) > 327) then
223 begin
224     edTiempEspEV4.Text := '327';
225 end;
226
227 //Restricción para tiempo de expulsión/retorno de EV4
228 if (edTiempExpRetEV4.Text = '0') or (edTiempExpRetEV4.Text = '') then
229 begin
230     //Si se sale del rango del tiempo, se auto
231     edTiempExpRetEV4.Text := '1'; //configura en sus respectivos limites
232 end
233 //Rango: [1-327]seg.
234 else if (StrToInt(edTiempExpRetEV4.Text) > 327) then
235 begin
236     edTiempExpRetEV4.Text := '327';
237 end;
238
239 //Restricción para tiempo de parte neumatica
240 if (edTiemNeum.Text = '0') or (edTiemNeum.Text = '') then
241 begin
242     //Si se sale del rango del tiempo, se auto
243     edTiemNeum.Text := '1'; //configura en sus respectivos limites
244 end
245 //Rango: [1-327]seg.
246 else if (StrToInt(edTiemNeum.Text) > 327) then
247 begin
248     edTiemNeum.Text := '327';
249 end;
250
251 //Restricción para No. de secuencias
252 if (edNoSec.Text = '0') or (edNoSec.Text = '') then
253 begin
254     //Si se sale del rango de secuencias, se auto
255     edNoSec.Text := '1'; //configura en sus respectivos limites
256 end
257 //Rango: [1-32767]seg.
258 else if (StrToInt(edNoSec.Text) > 32767) then
259 begin
```

```

250     edNoSec.Text := '32767';
251   end;
252
253   bCargConf.Caption := 'Reconfigurar';
254   //Se deshabilitan los edits en los que el usuario ingresa parametros
255   edTiempEspEV6.Enabled := false;
256   edTiempEspEV5.Enabled := false;
257   edTiempEspEV4.Enabled := false;
258   edTiempExpRetEV6.Enabled := false;
259   edTiempExpRetEV5.Enabled := false;
260   edTiempExpRetEV4.Enabled := false;
261   edNoSec.Enabled := false;
262   edTiemNeum.Enabled := false;
263
264   Config := true; //Los nuevos parametros se han cargado
265   //Cargan los Preset correspondientes al controlador
266   PrT1.WriteSync(StrToInt(edTiempEspEV6.Text)*100); //Escribe el Preset en el T1
267   PrT2.WriteSync(StrToInt(edTiemNeum.Text)*100); //Escribe el Preset en el T2
268   PrC1.WriteSync(StrToInt(edNoSec.Text)); //Escribe el Preset en el C1
269 end;
270 end;
271 //===== FIN DE ESCRITURA ======

```

### 3.9. Código Delphi de rutina de supervisión

Código 2: Código Delphi de rutina de supervisión

```

1 //===== TIMER 1 ======
2 //MONITOREA TODOS LOS CAMBIOS DEL PROCESO Y ACTUALIZA LA INTERFAZ
3 procedure TshSC5.Timer1Timer(Sender: TObject);
4 begin
5   // RESET FISICO
6   if (ResFisic.Value = '1') then
7     begin
8       labNoSecuen.Caption := '0'; //Se resetea el conteo del No. de secuencias
9       auxNeum2 := false;
10      i := 0;
11    end;
12
13 // DEL CILINDRO No. 6
14 if (SC6.Value = '1') then //Comportamiento Sensor SC6
15 begin
16   imSC6.Visible := true;
17 end
18 else
19 begin
20   imSC6.Visible := false;
21 end;
22

```

```
23 if (SE6.Value = '1') then      //Comportamiento Sensor SE6
24 begin
25     imSE6.Visible := true;
26 end
27 else
28 begin
29     imSE6.Visible := false;
30 end;
31
32 if (EV_C6.Value = '1') and (SE6.Value = '1') then //Comportamiento EV_C6
33 begin
34     Timer1.Enabled := false;
35     Timer3.Enabled := true;
36 end;
37
38 if (EV_E6.Value = '1') then //Comportamiento EV_E6
39 begin
40     PrT1.WriteSync(StrToInt(edTiempEspEV6.Text)*100); //Escribe el Preset en el T1
41     Timer1.Enabled := false;
42     Timer2.Enabled := true;
43 end;
44
45 // DEL CILINDRO No. 5
46 if (SC5.Value = '1') then      //Comportamiento Sensor SC5
47 begin
48     imSC5.Visible := true;
49 end
50 else
51 begin
52     imSC5.Visible := false;
53 end;
54
55 if (SE5.Value = '1') then      //Comportamiento Sensor SE5
56 begin
57     imSE5.Visible := true;
58 end
59 else
60 begin
61     imSE5.Visible := false;
62 end;
63
64 if (EV_C5.Value = '1') and (SE5.Value = '1') then //Comportamiento EV_C5
65 begin
66     Timer1.Enabled := false;
67     Timer5.Enabled := true;
68 end;
69
70 if (EV_E5.Value = '1') then //Comportamiento EV_E5
71 begin
72     PrT1.WriteSync(StrToInt(edTiempEspEV5.Text)*100); //Escribe el Preset en el T1
73     Timer1.Enabled := false;
```

```

74     Timer4.Enabled := true;
75   end;
76
77 // DEL CILINDRO No. 4
78 if (SC4.Value = '1') then      //Comportamiento Sensor SC4
79 begin
80   imSC4.Visible := true;
81   if auxNeum then
82     begin
83       Timer1.Enabled := false;
84       Timer8.Enabled := true;
85     end;
86   end
87 else
88 begin
89   imSC4.Visible := false;
90 end;
91
92 if (SE4.Value = '1') then      //Comportamiento Sensor SE4
93 begin
94   imSE4.Visible := true;
95 end
96 else
97 begin
98   imSE4.Visible := false;
99 end;
100
101 if (EV_C4.Value = '1') and (SE4.Value = '1') then //Comportamiento EV_C4
102 begin
103   Timer1.Enabled := false;
104   Timer7.Enabled := true;
105 end;
106
107 if (EV_E4.Value = '1') then //Comportamiento EV_E4
108 begin
109   PrT1.WriteString(StrToInt(edTiempEspEV4.Text)*100); //Escribe el Preset en el T1
110   Timer1.Enabled := false;
111   Timer6.Enabled := true;
112 end;
113 end;
114 //===== FIN TIMER 1 ======

```

### 3.10. Código restante del proyecto

Código 3: Código Delphi restante del proyecto

```

1 unit Interfaz;
2
3 interface

```

```
4
5 uses
6 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics
    → ,
7 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
8 //Agregados por mi
9 Vcl.Imaging.pngimage, Vcl.ExtCtrls, dOPCGUI, dOPCIntf, dOPCCOM, dOPCDA,
10 dOPCComm, dOPCOld;
11
12 type
13 TshSC5 = class(TForm)
14 ServerCombo: TComboBox;
15 ComboTopico: TComboBox;
16 dOPCServer1: TDOPCServer;
17 ConecDescone: TButton;
18 bStart: TButton;
19 bStop: TButton;
20 bReset: TButton;
21 Timer1: TTimer;
22 Label1: TLabel;
23 Label2: TLabel;
24 Label3: TLabel;
25 Label4: TLabel;
26 Label5: TLabel;
27 Label6: TLabel;
28 Image3: TImage;
29 Image4: TImage;
30 Image5: TImage;
31 Image6: TImage;
32 Image7: TImage;
33 EmboloEV4: TImage;
34 EmboloEV5: TImage;
35 EmboloEV6: TImage;
36 EmboloPB4: TImage;
37 EmboloPB5: TImage;
38 imSC6: TImage;
39 imSE6: TImage;
40 imSE5: TImage;
41 imSC5: TImage;
42 imSE4: TImage;
43 imSC4: TImage;
44 edTiempEspEV6: TEdit;
45 edNoSec: TEdit;
46 Label14: TLabel;
47 bCargConf: TButton;
48 Timer2: TTimer;
49 Timer3: TTimer;
50 Label17: TLabel;
51 edTiempEspEV5: TEdit;
52 Label18: TLabel;
53 edTiempEspEV4: TEdit;
```

```
54  Label19: TLabel;
55  Label20: TLabel;
56  Label21: TLabel;
57  Label22: TLabel;
58  Label23: TLabel;
59  edTiempExpRetEV6: TEdit;
60  edTiempExpRetEV5: TEdit;
61  edTiempExpRetEV4: TEdit;
62  Label24: TLabel;
63  Label25: TLabel;
64  Label26: TLabel;
65  Label15: TLabel;
66  edTiemNeum: TEdit;
67  Label7: TLabel;
68  Label8: TLabel;
69  Label9: TLabel;
70  labNoSecuen: TLabel;
71  Timer4: TTimer;
72  Timer5: TTimer;
73  Timer6: TTimer;
74  Timer7: TTimer;
75  Timer8: TTimer;
76  Timer9: TTimer;
77  Timer10: TTimer;
78  Timer11: TTimer;
79  Timer12: TTimer;
80  Image8: TImage;
81  Image9: TImage;
82  Image10: TImage;
83  Image11: TImage;
84  Image12: TImage;
85  Image13: TImage;
86  Image14: TImage;
87  Image15: TImage;
88  Image16: TImage;
89  Image17: TImage;
90  Image18: TImage;
91  Image19: TImage;
92  Image20: TImage;
93  Label10: TLabel;
94  Label11: TLabel;
95  procedure ServerComboDropDown(Sender: TObject);
96  procedure ConecDesconecClick(Sender: TObject);
97  procedure ComboTopicoDropDown(Sender: TObject);
98  procedure ComboTopicoChange(Sender: TObject);
99  procedure bStartMouseDown(Sender: TObject; Button: TMouseButton;
100   Shift: TShiftState; X, Y: Integer);
101 procedure bStartMouseUp(Sender: TObject; Button: TMouseButton;
102   Shift: TShiftState; X, Y: Integer);
103 procedure bStopMouseDown(Sender: TObject; Button: TMouseButton;
104   Shift: TShiftState; X, Y: Integer);
```

```
105 procedure bStopMouseUp(Sender: TObject; Button: TMouseButton;
106   Shift: TShiftState; X, Y: Integer);
107 procedure bResetMouseDown(Sender: TObject; Button: TMouseButton;
108   Shift: TShiftState; X, Y: Integer);
109 procedure bResetMouseUp(Sender: TObject; Button: TMouseButton;
110   Shift: TShiftState; X, Y: Integer);
111 procedure Timer1Timer(Sender: TObject);
112 procedure FormCreate(Sender: TObject);
113 procedure bCargConfMouseDown(Sender: TObject; Button: TMouseButton;
114   Shift: TShiftState; X, Y: Integer);
115 procedure restringirEdTiemp(Sender: TObject; var Key: Char);
116 procedure Timer2Timer(Sender: TObject);
117 procedure Timer3Timer(Sender: TObject);
118 procedure Timer4Timer(Sender: TObject);
119 procedure Timer5Timer(Sender: TObject);
120 procedure Timer6Timer(Sender: TObject);
121 procedure Timer7Timer(Sender: TObject);
122 procedure Timer8Timer(Sender: TObject);
123 procedure Timer9Timer(Sender: TObject);
124 procedure Timer10Timer(Sender: TObject);
125 procedure Timer11Timer(Sender: TObject);
126 procedure Timer12Timer(Sender: TObject);
127 private
128   TAGS : TdOPCGroup;
129 public
130   { Public declarations }
131 end;
132
133 var
134   shSC5: TshSC5;
135   STARTV : TdOPCItem;
136   STOPV : TdOPCItem;
137   RESETV : TdOPCItem;
138   SC6    : TdOPCItem;
139   SE6    : TdOPCItem;
140   EV_C6  : TdOPCItem;
141   EV_E6  : TdOPCItem;
142   SC5    : TdOPCItem;
143   SE5    : TdOPCItem;
144   EV_C5  : TdOPCItem;
145   EV_E5  : TdOPCItem;
146   SC4    : TdOPCItem;
147   SE4    : TdOPCItem;
148   EV_C4  : TdOPCItem;
149   EV_E4  : TdOPCItem;
150   ResFisic: TdOPCItem;
151   PrT1   : TdOPCItem;
152   PrT2   : TdOPCItem;
153   PrC1   : TdOPCItem;
154   auxDecim: single; //Variable auxiliar que acumula la parte decimal
155   Config : Boolean; //Variable que indica si estan o no cargados los
```

```
156 //parámetros actuales de la interfaz
157 auxNeum : Boolean; //Variables que indica si es momento de accionar la
158 auxNeum2: Boolean; //simulación de la parte neumatica
159 i : integer;
160 implementation
161 {$R *.dfm}
163
164 //Configuaciones iniciales al crear el formulario
165 procedure TshSC5.FormCreate(Sender: TObject);
166 begin
167   Width := 1365;
168   Height := 730;
169   Position := poScreenCenter;
170   //BorderStyle := bsDialog; //Quita la opción de maximizar
171   DoubleBuffered := True;
172 end;
173
174 //Restringe los caracteres introducidos en los Edit's
175 procedure TshSC5.restringirEdTiemp(Sender: TObject; var Key: Char);
176 begin
177   if not(Key in ['0'..'9',#8]) then
178     Key := #0;
179 end;
```

### 3.11. HMI



Figura 18: HMI.

En la Figura 18 se observa la interfaz de supervisión final. Al principio se encuentran bloqueados todos los botones y *edit's* para evitar problemas, únicamente se habilitan en el momento adecuado.

En un principio, únicamente esta habilitado el *combobox* del servidor, al seleccionar este, se habilita el botón *Conectar*, al dar clic en este se habilita el *combobox* del tópico, es en este donde se selecciona el nombre del programa, al hacer esto se habilitan los botones *Start*, *Stop*, *Reset* y *Reconfigurar*, es en ese momento donde empieza la comunicación, se deben ver reflejados también el estado de los sensores de contracción y expulsión, siendo que un principio únicamente los primeros son los que se encuentran activados.

Por default, la interfaz tiene preconfigurados los tiempos de espera de los pistones, así como el tiempo de expulsión/retorno de estos y el tiempo estimado para la parte neumática, lo mismo para el número de repeticiones de la secuencia. Sin embargo, es posible reconfigurar todos estos parámetros para lo cual se debe presionar el botón *Reconfigurar*, lo cual habilita los *edit's* de todos estos parámetros, una vez ingresado los nuevos, se debe presionar de nuevo el botón *Reconfigurar*, el cual habrá cambiado su nombre a *Cargar parámetros*, esta acción bloqueara de nuevo los *edit's*.

Un aspecto que vale la pena mencionar es que estos *edit's* están restringidos para que únicamente se puedan ingresar números enteros mayores a 0. En la Figura 18 se puede observar que el tiempo máximo que se puede ingresar en estos son 327 segundos, lo cual se deriva del número máximo que se puede colocar en el preset del temporizador en *RSLogix500*, si el número ingresado en los *edit's* se sale del rango de estos, al momento de presionar el botón *Cargar configuración*, se reemplazara el valor por el límite máximo correspondiente, es decir si valor ingresado es cero o no se escribe ningún carácter, se sobrescribirá el valor de 1, por el contrario, si se escribe para un *edit* de tiempo un número mayor a 327, el número se sobrescribirá por este mismo valor.



Figura 19: Secuencia de expulsión de pistones.

La Figura 19 tiene como propósito el ilustrar como es que se va realizando la expulsión de pistones, se observa además como se van prendiendo los respectivos sensores de contracción y expulsión.

### 3.11.1. Botones *Start*, *Stop* y *Reset*

Al momento de presionar estos 3 botones, se escribe un 1 en un bit de memoria del PLC, que corresponden a las variables *Start\_Remoto*, *Stop\_Remoto* y *Reset\_Remoto*, es en la rutina LADDER donde se especifican las consecuencias que tiene su activación en el proceso. El *Reset* de la interfaz posee un par de líneas de código extra que sirven para el reseteo del despliegue del número de secuencias en la interfaz y para poner en falso la variable booleana *auxNeum2* que es un indicador que sirve para activar la animación de los pistones neumáticos.

Código 4: Código botones *Start*, *Stop* y *Reset*

```

1 // START VIRTUAL:
2 procedure TshSC5.bStartMouseDown(Sender: TObject; Button: TMouseButton;
3   Shift: TShiftState; X, Y: Integer);
4 begin
5   STARTV.WriteSync('1'); // Mientras se presione el botón, se escribe un '1'
6 end;
7
8 procedure TshSC5.bStartMouseUp(Sender: TObject; Button: TMouseButton;
9   Shift: TShiftState; X, Y: Integer);
10 begin
11   STARTV.WriteSync('0'); // Al dejar de presionar el botón, se escribe un '0'
12 end;
13
14 // STOP VIRTUAL:
15 procedure TshSC5.bStopMouseDown(Sender: TObject; Button: TMouseButton;
16   Shift: TShiftState; X, Y: Integer);
17 begin
18   STOPV.WriteSync('1'); // Mientras se presione el botón, se escribe un '1'
19 end;
20
21 procedure TshSC5.bStopMouseUp(Sender: TObject; Button: TMouseButton;
22   Shift: TShiftState; X, Y: Integer);
23 begin
24   STOPV.WriteSync('0'); // Al dejar de presionar el botón, se escribe un '0'
25 end;
26
27 // RESET VIRTUAL:
28 procedure TshSC5.bResetMouseDown(Sender: TObject; Button: TMouseButton;
29   Shift: TShiftState; X, Y: Integer);
30 begin
31   RESETV.WriteSync('1'); // Mientras se presione el botón, se escribe un '1'
32   labNoSecuen.Caption := '0'; //Se resetea el conteo del No. de secuencias
33   auxNeum2 := false;
34   i := 0;
35 end;
36
37 procedure TshSC5.bResetMouseUp(Sender: TObject; Button: TMouseButton;
38   Shift: TShiftState; X, Y: Integer);
39 begin
40   RESETV.WriteSync('0'); // Al dejar de presionar el botón, se escribe un '0'
41 end;

```

### 3.11.2. Funcionamiento de pistones

En total se utilizaron 12 *Timers*, el *Timer1* esta todo el tiempo sensando el estado de los sensores de contracción y expulsión, lo mismo para las electroválvulas de contracción y expulsión. Los *Timers 2* a *12* se encargan de la animación de la expulsión y contracción de todos los pistones. En particular el *Timer2* expulsa el piston PB1 y el *Timer3* lo contrae, el *Timer4* expulsa el piston PB2 y el *Timer5*

lo contrae, el *Timer6* expulsa el pistón PB3 y el *Timer7* lo contrae. Para la parte neumática el *Timer8* expulsa el pistón PB4, el *Timer9* espera un tiempo, el *Timer10* expulsa el pistón PB5, el *Timer11* espera otro tiempo y el *Timer12* contrae los pistones PB4 y PB5.

Cuando la interfaz se abre y se pone en funcionamiento, esta activo el *Timer1*, al presionar *Start* se activa la electroválvula de expulsión del pistón PB1, el *Timer1* reconoce esta acción, por lo que se desactiva a si mismo y activa al *Timer2* el cual expulsa al pistón, cuando este *Timer2* finaliza con la expulsión del pistón, se desactiva a si mismo y activa nuevamente el *Timer1*, al momento de que este detecta que se activa el sensor de contracción del PB1, se desactiva de nueva cuenta y activa el *Timer3* responsable de la contracción del pistón PB1. Esta misma lógica aplica para la expulsión y contracción de los 3 cilindros de doble efecto. Para los cilindros de retorno por muelle ocurre algo semejante pero no idéntico, el *Timer8* expulsa al pistón PB4, al concluirse esta expulsión, se desactiva este *Timer8* y es el *Timer9* el siguiente en activarse, para después pasar al *Timer10*, después al *Timer11* y finalmente al *Timer12*, que es el responsable de la contracción de los pistones PB4 y PB5, despues de esto se activa el *Timer1* empezando de nuevo el ciclo.

Código 5: Código para la animación de los pistones

```

1 //===== TIMER 2 ======
2 //SE ENCARGA DE LA EXPULSIÓN DE LA EV6
3 procedure TshSC5.Timer2Timer(Sender: TObject);
4 var
5   NoCiclos : integer;
6   PartEntera : integer;
7   PartDecimal : single;
8 begin
9   if (SC6.Value = '1') then //Se monitorea el sensor SC6 y se actualiza
10     imSC6.Visible := true; //su estado
11   if (SC6.Value = '0') then
12     imSC6.Visible := false;
13   //Si la imagen del embolo esta en rango [520-780], se recorre la la imagen poco
14   //a poco. Si SE6 se activa, la imagen se coloca de golpe en su posición final
15   if (EmboloEV6.Left <= 780) and (EmboloEV6.Left > 520) and (SE6.Value = '0') then
16   begin
17     //Este código es para calcular el valor que se le debe restar a la abcisa de
18     //la imagen para que lo haga en el tiempo seleccionado por el usuario
19     NoCiclos := StrToInt(edTiempExpRetEV6.Text)*1000 div 50;
20     PartEntera := 260 div NoCiclos;
21     PartDecimal := 260/NoCiclos - PartEntera;
22     auxDecim := auxDecim + PartDecimal;
23     if (auxDecim >= 1) then
24     begin
25       EmboloEV6.Left := EmboloEV6.Left - 1;
26       auxDecim := auxDecim - 1;
27     end;
28   end;
29   EmboloEV6.Left := EmboloEV6.Left - PartEntera ;
30 end
31 else
32 begin

```

```
33 //Se entra aqui una vez que el embolo hizo todo el recorrido
34 EmboloEV6.Left := 520;
35 auxDecim := 0;
36 Timer2.Enabled := false;
37 Timer1.Enabled := true;
38 end;
39 end;
40 ===== FIN TIMER 2 =====
41
42 =====TIMER 3 =====
43 //SE ENCARGA DE LA CONTRACCIÓN DE LA EV6
44 procedure TshSC5.Timer3Timer(Sender: TObject);
45 var
46   NoCiclos : integer;
47   PartEntera : integer;
48   PartDecimal : single;
49 begin
50   if (SE6.Value = '1') then //Se monitorea el sensor SC6 y se actualiza
51     imSE6.Visible := true; //su estado
52   if (SE6.Value = '0') then
53     imSE6.Visible := false;
54   //Si la imagen del embolo esta en rango [520-780), se recorre la la imagen poco
55   //a poco. Si SC6 se activa, la imagen se coloca de golpe en su posición final
56   if (EmboloEV6.Left >= 520) and (EmboloEV6.Left < 780) and (SC6.Value = '0') then
57   begin
58     NoCiclos := StrToInt(edTiempExpRetEV6.Text)*1000 div 50;
59     PartEntera := 260 div NoCiclos;
60     PartDecimal := 260/NoCiclos - PartEntera;
61     auxDecim := auxDecim + PartDecimal;
62     if (auxDecim >= 1) then
63       begin
64         EmboloEV6.Left := EmboloEV6.Left + 1;
65         auxDecim := auxDecim - 1;
66       end;
67     EmboloEV6.Left := EmboloEV6.Left + PartEntera ;
68   end
69 else
70 begin
71   //Se entra aqui una vez que el embolo hizo todo el recorrido
72   EmboloEV6.Left := 780;
73   auxDecim := 0;
74   Timer3.Enabled := false;
75   Timer1.Enabled := true;
76 end;
77 end;
78 ===== FIN TIMER 3 =====
79
80 ===== TIMER 4 =====
81 //SE ENCARGA DE LA EXPULSIÓN DE LA EV5
82 procedure TshSC5.Timer4Timer(Sender: TObject);
83 var
```

```
84  NoCiclos : integer;
85  PartEntera : integer;
86  PartDecimal : single;
87  begin
88  if (SC5.Value = '1') then //Se monitorea el sensor SC5 y se actualiza
89    imSC5.Visible := true; //su estado
90  if (SC5.Value = '0') then
91    imSC5.Visible := false;
92  //Si la imagen del embolo esta en rango (520-780], se recorre la la imagen poco
93  //a poco. Si SE5 se activa, la imagen se coloca de golpe en su posición final
94  if (EmboloEV5.Left <= 780) and (EmboloEV5.Left > 520) and (SE5.Value = '0') then
95  begin
96    //Este código es para calcular el valor que se le debe restar a la abcisa de
97    //la imagen para que lo haga en el tiempo seleccionado por el usuario
98    NoCiclos := StrToInt(edTiempExpRetEV5.Text)*1000 div 50;
99    PartEntera := 260 div NoCiclos;
100   PartDecimal := 260/NoCiclos - PartEntera;
101   auxDecim := auxDecim + PartDecimal;
102  if (auxDecim >= 1) then
103  begin
104    EmboloEV5.Left := EmboloEV5.Left - 1;
105    auxDecim := auxDecim - 1;
106  end;
107  EmboloEV5.Left := EmboloEV5.Left - PartEntera ;
108 end
109 else
110 begin
111  //Se entra aqui una vez que el embolo hizo todo el recorrido
112  EmboloEV5.Left := 520;
113  auxDecim := 0;
114  Timer4.Enabled := false;
115  Timer1.Enabled := true;
116 end;
117 end;
118 ===== FIN TIMER 4 =====
119
120 ===== TIMER 5 =====
121 //SE ENCARGA DE LA CONTRACCIÓN DE LA EV5
122 procedure TshSC5.Timer5Timer(Sender: TObject);
123 var
124  NoCiclos : integer;
125  PartEntera : integer;
126  PartDecimal : single;
127 begin
128  if (SE5.Value = '1') then //Se monitorea el sensor SE5 y se actualiza
129    imSE5.Visible := true; //su estado
130  if (SE5.Value = '0') then
131    imSE5.Visible := false;
132  //Si la imagen del embolo esta en rango [520-780), se recorre la la imagen poco
133  //a poco. Si SC5 se activa, la imagen se coloca de golpe en su posición final
134  if (EmboloEV5.Left >= 520) and (EmboloEV5.Left < 780) and (SC5.Value = '0') then
```

```

135 begin
136   NoCiclos := StrToInt(edTiempExpRetEV5.Text)*1000 div 50;
137   PartEntera := 260 div NoCiclos;
138   PartDecimal := 260/NoCiclos - PartEntera;
139   auxDecim := auxDecim + PartDecimal;
140   if (auxDecim >= 1) then
141     begin
142       EmboloEV5.Left := EmboloEV5.Left + 1;
143       auxDecim := auxDecim - 1;
144     end;
145   EmboloEV5.Left := EmboloEV5.Left + PartEntera ;
146 end
147 else
148 begin
149   //Se entra aqui una vez que el embolo hizo todo el recorrido
150   EmboloEV5.Left := 780;
151   auxDecim := 0;
152   auxNeum2 := true;
153   Timer5.Enabled := false;
154   Timer1.Enabled := true;
155 end;
156 end;
157 //===== FIN TIMER 5 ======
158
159 //===== TIMER 6 ======
160 //SE ENCARGA DE LA EXPULSIÓN DE LA EV4
161 procedure TshSC5.Timer6Timer(Sender: TObject);
162 var
163   NoCiclos : integer;
164   PartEntera : integer;
165   PartDecimal : single;
166 begin
167   if (SC4.Value = '1') then //Se monitorea el sensor SC4 y se actualiza
168     imSC4.Visible := true; //su estado
169   if (SC4.Value = '0') then
170     imSC4.Visible := false;
171   //Si la imagen del embolo esta en rango [520-780], se recorre la la imagen poco
172   //a poco. Si SE4 se activa, la imagen se coloca de golpe en su posición final
173   if (EmboloEV4.Left <= 780) and (EmboloEV4.Left > 520) and (SE4.Value = '0') then
174     begin
175       //Este código es para calcular el valor que se le debe restar a la abcisa de
176       //la imagen para que lo haga en el tiempo seleccionado por el usuario
177       NoCiclos := StrToInt(edTiempExpRetEV4.Text)*1000 div 50;
178       PartEntera := 260 div NoCiclos;
179       PartDecimal := 260/NoCiclos - PartEntera;
180       auxDecim := auxDecim + PartDecimal;
181       if (auxDecim >= 1) then
182         begin
183           EmboloEV4.Left := EmboloEV4.Left - 1;
184           auxDecim := auxDecim - 1;
185         end;

```

```
186 EmboloEV4.Left := EmboloEV4.Left - PartEntera ;
187 end
188 else
189 begin
190 //Se entra aqui una vez que el embolo hizo todo el recorrido
191 EmboloEV4.Left := 520;
192 auxDecim := 0;
193 Timer6.Enabled := false;
194 Timer1.Enabled := true;
195 end;
196 end;
197 //===== FIN TIMER 6 ======
198
199 //===== TIMER 7 ======
200 //SE ENCARGA DE LA CONTRACCIÓN DE LA EV4
201 procedure TshSC5.Timer7Timer(Sender: TObject);
202 var
203 NoCiclos : integer;
204 PartEntera : integer;
205 PartDecimal : single;
206 begin
207 if (SE4.Value = '1') then //Se monitorea el sensor SE4 y se actualiza
208 imSE4.Visible := true; //su estado
209 if (SE4.Value = '0') then
210 imSE4.Visible := false;
211 //Si la imagen del embolo esta en rango [520-780], se recorre la la imagen poco
212 //a poco. Si SC4 se activa, la imagen se coloca de golpe en su posición final
213 if (EmboloEV4.Left >= 520) and (EmboloEV4.Left < 780) and (SC4.Value = '0') then
214 begin
215 NoCiclos := StrToInt(edTiempExpRetEV4.Text)*1000 div 50;
216 PartEntera := 260 div NoCiclos;
217 PartDecimal := 260/NoCiclos - PartEntera;
218 auxDecim := auxDecim + PartDecimal;
219 if (auxDecim >= 1) then
220 begin
221 EmboloEV4.Left := EmboloEV4.Left + 1;
222 auxDecim := auxDecim - 1;
223 end;
224 EmboloEV4.Left := EmboloEV4.Left + PartEntera ;
225 end
226 else
227 begin
228 //Se entra aqui una vez que el embolo hizo todo el recorrido
229 EmboloEV4.Left := 780;
230 auxDecim := 0;
231 auxNeum := true;
232 Timer7.Enabled := false;
233 Timer1.Enabled := true;
234 end;
235 end;
236 //===== FIN TIMER 7 ======
```

```
237
238 //=====TIMER 8 ======
239 //SE ENCARGA DE LA EXPULSIÓN DEL PISTÓN PB4
240 procedure TshSC5.Timer8Timer(Sender: TObject);
241 var
242   NoCiclos : integer;
243   PartEntera : integer;
244   PartDecimal : single;
245   tiempo : integer;
246 begin
247   //Si la imagen del embolo PB4 esta en rango [550-810], se desplaza el embolo PB4
248   if (EmboloPB4.Left <= 810) and (EmboloPB4.Left > 550) then
249   begin
250     //Este código es para calcular el valor que se le debe restar a la abcisa de
251     //la imagen para que lo haga en el tiempo seleccionado por el usuario
252     tiempo := StrToInt(edTiemNeum.Text)*1000 div 5;
253     NoCiclos := tiempo div 50;
254     PartEntera := 260 div NoCiclos;
255     PartDecimal := 260/NoCiclos - PartEntera;
256     auxDecim := auxDecim + PartDecimal;
257     if (auxDecim >= 1) then
258     begin
259       EmboloPB4.Left := EmboloPB4.Left - 1;
260       auxDecim := auxDecim - 1;
261     end;
262     EmboloPB4.Left := EmboloPB4.Left - PartEntera ;
263   end
264   else
265   begin
266     //Se entra aqui una vez que el embolo hizo todo el recorrido
267     EmboloPB4.Left := 550;
268     auxDecim := 0;
269     Timer8.Enabled := false;
270     Timer9.Enabled := true;
271   end;
272 end;
273 //===== FIN TIMER 8 ======
274
275 //===== TIMER 9 ======
276 //TIEMPO DE ESPERA DEL PISTÓN PB4
277 procedure TshSC5.Timer9Timer(Sender: TObject);
278 var
279   NoCiclos : integer;
280   tiempo : integer;
281 begin
282   tiempo := StrToInt(edTiemNeum.Text)*1000 div 5;
283   NoCiclos := tiempo div 50;
284   i := i + 1;
285   if (i >= NoCiclos) then
286   begin
287     Timer9.Enabled := false;
```

```
288     Timer10.Enabled := true;
289   end;
290 end;
291 //===== FIN TIMER 9 ======
292
293 //=====TIMER 10 ======
294 //SE ENCARGA DE LA EXPULSIÓN DEL PISTÓN PB5
295 procedure TshSC5.Timer10Timer(Sender: TObject);
296 var
297   NoCiclos : integer;
298   PartEntera : integer;
299   PartDecimal : single;
300   tiempo : integer;
301 begin
302   //Si la imagen del embolo PB4 esta en rango (550-810], se desplaza el embolo PB4
303   if (EmboloPB5.Left <= 810) and (EmboloPB5.Left > 550) then
304     begin
305       //Este código es para calcular el valor que se le debe restar a la abcisa de
306       //la imagen para que lo haga en el tiempo seleccionado por el usuario
307       tiempo := StrToInt(edTiemNeum.Text)*1000 div 5;
308       NoCiclos := tiempo div 50;
309       PartEntera := 260 div NoCiclos;
310       PartDecimal := 260/NoCiclos - PartEntera;
311       auxDecim := auxDecim + PartDecimal;
312       if (auxDecim >= 1) then
313         begin
314           EmboloPB5.Left := EmboloPB5.Left - 1;
315           auxDecim := auxDecim - 1;
316         end;
317       EmboloPB5.Left := EmboloPB5.Left - PartEntera ;
318     end
319   else
320     begin
321       //Se entra aqui una vez que el embolo hizo todo el recorrido
322       EmboloPB5.Left := 550;
323       auxDecim := 0;
324       i := 0;
325       Timer10.Enabled := false;
326       Timer11.Enabled := true;
327     end;
328   end;
329 //===== FIN TIMER 10 ======
330
331 //===== TIMER 11 ======
332 //TIEMPO DE ESPERA DEL PISTÓN PB5
333 procedure TshSC5.Timer11Timer(Sender: TObject);
334 var
335   NoCiclos : integer;
336   tiempo : integer;
337 begin
338   tiempo := StrToInt(edTiemNeum.Text)*1000 div 5;
```

```
339  NoCiclos := tiempo div 50;
340  i := i + 1;
341  if (i >= NoCiclos) then
342    begin
343      Timer11.Enabled := false;
344      Timer12.Enabled := true;
345    end;
346  end;
347 //===== FIN TIMER 11 =====
348
349 //===== TIMER 12 =====
350 //SE ENCARGA DE LA CONTRACCIÓN DEL PISTÓN PB4 y PB5
351 procedure TshSC5.Timer12Timer(Sender: TObject);
352 var
353   NoCiclos : integer;
354   PartEntera : integer;
355   PartDecimal : single;
356   tiempo : integer;
357 begin
358   //Si la imagen del embolo PB5 esta en rango [550-810], se desplazan ambos embolos
359   if (EmbolоНum.Left >= 550) and (EmbolоНum.Left < 810) and (EV_E6.Value = '0') then
360     begin
361       //Este código es para calcular el valor que se le debe restar a la abcisa de
362       //la imagen para que lo haga en el tiempo seleccionado por el usuario
363       tiempo := StrToInt(edTiemNeum.Text)*1000 div 5;
364       NoCiclos := tiempo div 50;
365       PartEntera := 260 div NoCiclos;
366       PartDecimal := 260/NoCiclos - PartEntera;
367       auxDecim := auxDecim + PartDecimal;
368       if (auxDecim >= 1) then
369         begin
370           EmboloPB4.Left := EmboloPB4.Left + 1;
371           EmboloPB5.Left := EmboloPB5.Left + 1;
372           auxDecim := auxDecim - 1;
373         end;
374       EmboloPB4.Left := EmboloPB4.Left + PartEntera ;
375       EmboloPB5.Left := EmboloPB5.Left + PartEntera ;
376     end
377     else
378     begin
379       //Se entra aqui una vez que el embolo hizo todo el recorrido
380       EmboloPB4.Left := 810;
381       EmboloPB5.Left := 810;
382       auxDecim := 0;
383       i := 0;
384       //Si auxNeum = false, implica que activo Reset, asi que no debe aumentar
385       //la cuenta de No. Secuencias
386       if (auxNeum) and (auxNeum2) then
387         labNoSecuen.Caption := IntToStr(StrToInt(labNoSecuen.Caption)+1);
388       auxNeum := false;
389       auxNeum2 := false;
```

```
390     Timer12.Enabled := false;  
391     Timer1.Enabled := true;  
392   end;  
393 end;  
394 //===== FIN TIMER 12 ======
```

## 4. Galería fotográfica



Figura 20: Cilindro de doble efecto.



Figura 21: Cilindro de retorno por muelle.



Figura 22: Pistón.



Figura 23: Led OFF.



Figura 24: Led ON.

## 5. Aportaciones del grupo de trabajo

Se enlistan los nombres de los compañeros y la aportación que tome de ellos:

- De Carlos Rubio Becerra: La función *DoubleBuffered* utilizada para mejorar la animación de la expulsión y contracción de los pistones, pues sin esta, estos al moverse, como que parpadean.
- De Hoover Mujica Ortega: En Delphi, la opción del bloque OPC para cambiar la velocidad de comunicación OPC (*UpdateRate*), pues ignoraba la existencia de esta, por lo que estaba trabajando con el valor por default de 1000, lo que se reflejaba en un ligero desfase entre el cambio de estado de mis variables en el PLC y el despliegue de estos en mi interfaz gráfica.
- A todos mis compañeros del grupo de trabajo en lo que respecta a retroalimentación para mejorar mi exposición del proyecto, pues fui de los últimos en exponer y eso contribuyo a que pudiera mejorar el orden en el desarrollo de mi presentación y la calidad de las diapositivas de esta.

## 6. Conclusión

Al ser este un proyecto complejo que requirió de bastante tiempo para su realización, se observó que la herramienta del Diagrama de Gantt es muy útil para calendarizar las actividades y así tener un control del avance que se lleva del proyecto. También se observó que al principio uno concibe las partes del proyecto en su mente y como se deberían ir desarrollando estas, y en base a esto uno asigna las actividades que corresponderán con estas, sin embargo en el proceso uno se pueda dar cuenta de que hay ciertas cosas que no son como las concebimos al principio, algunas cosas que parecían fáciles y triviales de desarrollar, al final resultaron más complicadas de lo esperado, lo cual se vio reflejado en una mayor inversión de tiempo respecto al que se había pensado en un principio. En mi caso, para este proyecto en particular pensé que la culminación del sistema de supervisión con su respectiva animación no iba a ser complicado, sin embargo me encontré con compilaciones que no había considerado en un principio.

Se observó que proyectos de gran complejidad requieren de diversas habilidades para su desarrollo, me pareció bastante enriquecedor el proyecto en este sentido pues me hizo darme cuenta de las áreas en las que me encuentro más cómodo y en las que no, sin embargo, reconozco la importancia de tener fortalecidas todas estas áreas, y que debo trabajar para mejorar en las áreas donde no me considero tan hábil.

Algo crucial de lo que me he dado cuenta estos últimos semestres es la importancia de ser metódico, con esto me refiero más que nada con lo relacionado con la programación, para el desarrollo de la rutina de control la fui realizando poco a poco, validando que cada pedazo del GRAFCET funcionara correctamente, lo mismo hice para el desarrollo del código en Delphi. Esto me permitió que los errores que iban apareciendo en el proceso, pudiera identificarlos fácilmente e identificar de igual forma el por qué de ese error, de esta forma era capaz de darle solución, pues un ligero error puede condicionar toda la implementación posterior que se realice.

## 7. Discusión

La metodología GRAFCET es muy robusta, lo cual se comprobó al poner a prueba la rutina de control en LADDER de este proyecto, pues al jugar con las múltiples entradas con las que se contaba,

el proceso se mantuvo fiel a su comportamiento sin realizar cosas extrañas. Para procesos complejos es recomendable siempre la utilización de GRAFCET pues en su metodología esta inmersa una sólida robustez ante fallos.

La automatización del proceso está en manos meramente del PLC, este podría funcionar sin necesidad de la interfaz de supervisión, sin embargo contar con esta es de gran ayuda para el operador, y es aquí donde se observa la importancia de las interfaces de supervisión, pues en esta se despliega la información más pertinente del proceso de una forma mucho más amigable para el operador en comparación con observar la rutina de control. Se observa que existen muchas herramientas para tener un sistema de supervisión para nuestro proceso, en este proyecto se abordó la opción de crearla mediante el software Delphi RAD Studio, sin embargo, existen muchas opciones, por ejemplo para la marca Allen Bradley, *Studio5000* cuenta con su propio entorno para el desarrollo de estas interfaces, me refiero a *FactoryTalk View Studio*, SIEMENS de igual forma cuenta con su software propio para la creación de las rutinas de control y para su respectivo sistema de supervisión. En LabView por ejemplo creas a la par tu sistema de supervisión y el programa. Por tanto se puede concluir que existen muchas herramientas y conocer acerca de estas permitirá obtener un criterio apropiado al momento de seleccionar una de estas para un proceso y problema en particular.

## 8. Referencias

- [1] Project Management Institute, Inc., editor. (2017). Guía del PMBOK (Sexta edición, Vol. 6). Project Management Institute, Inc.

## Anexo A. Hojas de datos

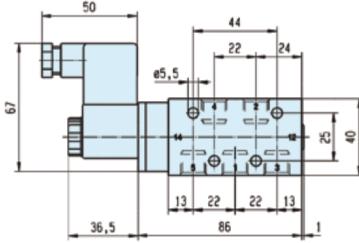
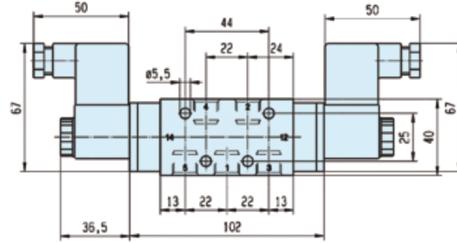
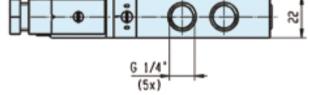
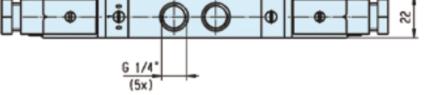
Hojas de especificaciones

PDE2600PNUK  
Parker Pneumatic

Fluid Controls

**Piped Valves - G1/4" Series**

**Solenoid Operated Versions**  
**P03 Versions with 22 mm Coil**

Port size	Orifice	$Q_n$	Admissible differential pressure (bar) maximum	Maximum admissible fluid temperature (°C) minimum = 0°C	Seat disc	Reference number	Consumption Power (Watt)	Weight (g)	Elect. Group	Dim. Ref.
G	mm	l/min	min	DC= AC=	Air & Neutral gases	Valve	Housing	Coil	DC	AC
<b>5/2 Solenoid operated Combined spring &amp; air return (monostable)</b>										
1/4	7	1250	2.5	10 10	50	NBR	341P03	-	496131	3 3 250 - 18
1/4	7	1250	2.5	10 10	50	NBR	341P03	-	496482	3 3 250 - 18
1/4	7	1250	2.5	10 10	50	NBR	341P03	-	496637	3 3 250 - 18
<b>5/2 Solenoid operated and return (bistable)</b>										
1/4	7	1250	2.5	10 10	50	NBR	347P03	-	496131	3 3 350 - 19
1/4	7	1250	2.5	10 10	50	NBR	347P03	-	496482	3 3 350 - 19
1/4	7	1250	2.5	10 10	50	NBR	347P03	-	496637	3 3 350 - 19
Dimensions Reference 18						Dimensions Reference 19				
										
										

Please consult the "How to Order" part at the end of each coil chapter.



Figura A.1: Hoja de datos 1 de 6.

PDE2600PNUK  
**Parker Pneumatic**

**ISO 15552 Cylinders - P1D**

**Design Variants**

**P1D-S Premier Cylinders**

The P1D-S series is the premier in ISO pneumatic cylinders. With various piston rod materials, seal options and supported by a full range of ISO mountings the P1D-S series is suitable for a wide range of applications.



**P1D-B Basic Cylinders**

The P1D-B series features a profile design and is the value line ISO cylinder for basic applications where no special options are required. This series is suitable for general industrial applications and is supported by a full range of ISO mountings.



**P1D-C Ultra Clean Cylinders**

This series offers an ultra clean external design of cylinders that are suitable for applications that require a clean profile. With particular design features for the food and packaging industries this product can also be used for applications on vehicles, in sawmills and bag-filling industries where a clean design is important.



**P1D-C Pro Clean Cylinders**

This series of clean design cylinders offers two T slots within one face of the tube allowing the possibility to add sensors. The position of the T slots can be specified on any single face using the order code key. These cylinders have a clean design but are intended for applications where sensors are required.



**P1D-T Tie Rod Cylinders**

The P1D-T range of tie rod cylinders is intended for use in a wide range of applications. Careful design and high quality manufacture throughout ensure long service life and optimum economy. Mounting dimensions fully in accordance with ISO 15552 (ISO 6431 and CETOP RP52P) greatly simplifies installation and world-wide interchangeability.



**P1D-X High and Low Temperature Cylinders**

For extreme conditions these cylinders for high and low temperatures have materials and sealing systems specially designed for their particular temperature ranges. End covers and pistons are made entirely from metal, to give optimum function at **high** or **low** temperature in combination with seals made from specially tested materials and special grease.



**P1D-L Rod Locking Cylinders**

P1D-L is a series of extremely compact rod lock cylinders for demanding applications. This version allows the piston rod to be locked in any position but can also be used as a brake (limits apply) thanks to the rigid design. With helical grooves on the precision clamping sleeves the locking function allows for applications where the piston rod is exposed to liquids and contamination.



Figura A.2: Hoja de datos 2 de 6.

PDE2600PNUK  
**Parker Pneumatic**

**Compact Valves - PVL Series**

Stacking high flow valves with air pilot or solenoid actuation. Lightweight plastic bodies feature push-in or threaded connections. Stacking valves feature modular inlet and exhaust facility.



- High flow, compact size
- Push-in or threaded connection
- DIN rail or block mounting
- Light weight construction



For ATEX specific products  
 contact Sales Office

**Operating information**

Working pressure;	2-10 bar
Pneumatically operated:	2-10 bar
Electrically operated, bistable:	2-10 bar
Electrically operated, monostable:	3-10 bar
Working temperature	-15 °C to +60 °C
<b>PVL-B</b>	<b>PVL-C</b>
Flow (Qmax):	900 l/min 1800 l/min
Flow Qn:	540 l/min 1100 l/min
Flow measured with valve stacked in island.	

**PVL-B & PVL-C Valve body - Stand alone and stackable design - Without pilot**

Pneumatic or 15 mm electric pilot must be ordered separately

Symbol	Description	Valve size	Connection	Weight (g)	Stand Alone Order code	Stackable Order code
	5/2 single acting / Spring return	PVL-B	Push-in Ø 6 mm	125	<b>PVL-B111606</b>	<b>PVL-B121606</b>
			G 1/8	125	<b>PVL-B111618</b>	<b>PVL-B121618</b>
	5/2 single acting / Air spring return	PVL-C	Push-in Ø 8 mm	240	<b>PVL-C111608</b>	<b>PVL-C121608</b>
			G 1/4	240	<b>PVL-C111619</b>	<b>PVL-C121619</b>
			G 3/8	240	<b>PVL-C111613</b>	<b>PVL-C121613</b>
			G 1/8	125	<b>PVL-B113606</b>	<b>PVL-B123606</b>
	5/2 double acting	PVL-B	Push-in Ø 6 mm	120	<b>PVL-B112606</b>	<b>PVL-B122606</b>
			G 1/8	120	<b>PVL-B112618</b>	<b>PVL-B122618</b>
	5/2 double acting	PVL-C	Push-in Ø 8 mm	230	<b>PVL-C112608</b>	<b>PVL-C122608</b>
			G 1/4	230	<b>PVL-C112619</b>	<b>PVL-C122619</b>
			G 3/8	230	<b>PVL-C112613</b>	<b>PVL-C122613</b>
			G 1/8	130	<b>PVL-B117606</b>	<b>PVL-B127606</b>
	5/3 APB (All Ports Blocked)	PVL-B	Push-in Ø 6 mm	130	<b>PVL-B117618</b>	<b>PVL-B127618</b>
			G 1/8	130	<b>PVL-B117608</b>	<b>PVL-C127608</b>
	5/3 Vented centre	PVL-C	Push-in Ø 8 mm	250	<b>PVL-C117619</b>	<b>PVL-C127619</b>
			G 1/4	250	<b>PVL-B118606</b>	<b>PVL-B128606</b>
			G 1/8	130	<b>PVL-B118618</b>	<b>PVL-B128618</b>
			G 1/4	250	<b>PVL-C118608</b>	<b>PVL-C128608</b>
	Double 3/2 Normally Closed (NC)	PVL-B	Push-in Ø 6 mm	130	<b>PVL-B115606</b>	<b>PVL-B125606</b>
			G 1/8	130	<b>PVL-B115618</b>	<b>PVL-B125618</b>

Parker Hannifin Corporation  
 Pneumatic Division - Europe



Figura A.3: Hoja de datos 3 de 6.

PDE2600PNUK  
**Parker Pneumatic**

**Interface - PS1 Series**

High speed poppet type solenoid valves with individual electrical connections. Light weight plastic bodies with DIN rail manifold. Ideal for cabinet installations.

- High speed poppet valve
- Push-in connection
- Built-in terminal block
- Pneumatic output indicator
- DIN rail mounting

Poppet valve, not suitable for use with Soft or Slow start Valves. When pressure is applied, the double solenoid interface takes a pre-determined position (non activated). In the absence of electrical signal, output 2 (yellow indicator) is activated, output 4 (red indicator) is non activated. Double solenoid version is delivered as standard with Non locking flush override.



**Operating information**

Working pressure:	3/2 and 4/2:	3-8 bar
Low pressure interface 3/2:		1-8 bar
Flow (Qmax):		200 l/min
Working temperature		-15 °C to +60 °C

For more information see [www.parker.com/euro\\_pneumatic](http://www.parker.com/euro_pneumatic)

**Head and tail sets - intermediate supply modules**

For 3/2 and 4/2 interface modules



Description	Characteristics	Pneumatic connection	Weight (g)	Order code
	Set for single air supply connection 1 electrical common terminal 1 main air supply port 1 exhaust port	Push-in Ø 6 mm G1/8	100 100	<b>PS1-E101</b> <b>PS1-E1018</b>
	Set for double air supply connection 1 electrical common terminal 2 main air supply ports 2 exhaust ports	Push-in Ø 6 mm G1/8	125 125	<b>PS1-E102</b> <b>PS1-E1028</b>
	Intermediate air supply module 1 air supply port 1 exhaust port (see description below)	G1/8	45	<b>PS1-E1038</b>

Incorporating an intermediate air supply module into a group of electro-pneumatic modules gives the following options :

**Air supply connection**

- Additional G1/8 port to supplement the air supply to the manifold
- Blank the common air supply, enabling different pressures to be supplied to groups of modules within the manifold.

**Exhaust connection**

- Additional exhaust flow through the G1/8 port
  - Blank the common exhaust, enabling separation of exhaust from module groups
- The intermediate air supply module is supplied with 4 interchangeable connectors to enable the above supply options to be achieved.

**Electro-pneumatic modules**

Modules without solenoid valve

Visual indication of pneumatic output status : Closed = Red      Open = Yellow

Symbol function	Description	Pneumatic connection	Weight (g)	Order code
	3/2 normally closed (NC)	Push-in Ø 4 mm M5 Push-in Ø 6 mm	52 55 55	<b>PS1-E111</b> <b>PS1-E115</b> <b>PS1-E116</b>
	3/2 normally open (NO)	Push-in Ø 4 mm M5 Push-in Ø 6 mm	52 55 55	<b>PS1-E121</b> <b>PS1-E125</b> <b>PS1-E126</b>
	4/2 single solenoid / spring return	Push-in Ø 4 mm M5 Push-in Ø 6 mm	120 120 125	<b>PS1-E181</b> <b>PS1-E185</b> <b>PS1-E186</b>
	4/2 double solenoid	Push-in Ø 4 mm M5 Push-in Ø 6 mm	120 120 125	<b>PS1-E191</b> <b>PS1-E195</b> <b>PS1-E196</b>



Figura A.4: Hoja de datos 4 de 6.

PDE2600PNUK  
**Parker Pneumatic**

**Logic Processing**

Miniature high-speed valves in stand alone, stackable or combined modules, incorporating standard logic functions. The range also includes timers and impulse modules.

- Complete range
- Stand alone, stackable or combinable modules
- Very fast response time
- Flexible and highly maintainable system
- DIN rail mounting
- Ø 4mm connection

 **For ATEX specific products contact Sales Office**



CE Ex

**Operating information**

Working pressure 3 to 8 bar  
Working temperature -15 °C to 60 °C  
Flow (Qmax) 180 l/min (PRD = 60 l/min)  
ATEX approval: CE Ex II 2 GD c 85°C

For more information see [www.parker.com/euro\\_pneumatic](http://www.parker.com/euro_pneumatic)

**Logic sequencer**

**Step modules**

	Order code
Without sub base	<b>PSM-A10</b>
Pneumatic output	
Visual indication of pneumatic output and manual override	<b>PSM-A12</b>
With sub base	<b>PSM-B12</b>
Without manual override	

**Step module subbase**

	Order code
Subbase	<b>PSB-A12</b>
Additional interlock	<b>PSV-A12</b>

**Set of head and tail modules and deviation modules**

	Order code
Head & tail set	<b>PSE-A12</b>
Deviation standard	<b>PSD-A12</b>
Deviation for remote	<b>PSD-B12</b>

**Logic elements**

**Line mounted elements**

	Order code
AND	<b>PLL-A11</b>
OR	<b>PLK-A11</b>
Clip on Din-rail	<b>PZM-L199</b>

**Combinable elements**

	Order code
AND	<b>PLL-B12</b>
OR	<b>PLK-B12</b>
NOT	<b>PLN-B12</b>
INPUT	<b>PLE-B12</b>

**Subbase mounted elements**

	Order code
AND	<b>PLL-C10</b>
NOT inhibit standard	<b>PLN-C10</b>
NOT inhibit threshold	<b>PLN-D10</b>
OR	<b>PLK-C10</b>
YES regenerated	<b>PLJ-C10</b>

3 port subbase to be ordered separately.

**Logic relays**

**Pressure switch**

To be used with 3 port subbase	
Order code	

With subbase

**PRE-A12**

Without subbase

**PRE-A10**

**Amplifier relays**

To be used with 4 port subbase	
Order code	

With subbase

**PRD-A12**

Without subbase

**PRD-A10**

**Memory relays**

To be used with 4 port subbase	
Order code	

With subbase

**PLM-A12**

Without subbase

**PLM-A10**

**Sensor relays**

	
Order code	
With subbase	<b>PRF-A12</b>

Without subbase

**PRF-A10**

To be mounted on 3 port subbase



**Function**

**Timing**

Output after timed period	0,1 to 3s 0,1 to 30s 10 to 180s
With subbase	0,1 to 30s
Output during timed period	0,1 to 3s 0,1 to 30s 10 to 180s
	<b>PRT-E10</b> <b>PRT-A10</b> <b>PRT-B10</b> <b>PRT-A12</b> <b>PRT-F10</b> <b>PRT-C10</b> <b>PRT-D10</b>

**Order code**

**Input module**

3 port "common input"  
3 port "cascade"

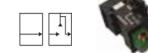
4 port subbase\*

**Order code**

**PZU-E12**  
**PZU-A12**  
**PZU-C12**  
**PZU-B12**

\* For combination with memory relay and amplified relay.

3 port and 4 port subbases



PDE2600PNUK  
**Parker Pneumatic**

Careful design and high quality manufacture throughout ensure long service life and optimum economy.  
 Mounting dimensions fully in accordance with ISO 6432 and CETOP RP52P greatly simplifies installation and world-wide interchangeability.

- Mini cylinder according to ISO 6432
- Available in 10 to 25 mm bores
- Corrosion resistant design and low weight construction
- Magnetic piston as standard
- End stroke buffers for long service life

**ISO 6432 Mini Cylinders - P1A-S**



**Operating information**

Working pressure: Max 10 bar  
 Temperature range: -20°C to +80°C Ø10-25mm

**Design variants**

**High temperature**

Ø12 and 16mm -10°C to +120°C Non-magnetic piston  
 Ø20 and 25mm -10°C to +150°C Non-magnetic piston

**External seals**

flourinated rubber -20°C to +80°C Magnetic piston

Prelubricated, further lubrication is not normally necessary.

If additional lubrication is introduced it must be continued.

**Double acting buffer cushioning**

**Ø10mm - (M5)**

Stroke mm	Order code
10	P1A-S010DS-0010
15	P1A-S010DS-0015
25	P1A-S010DS-0025
30	P1A-S010DS-0030
40	P1A-S010DS-0040
50	P1A-S010DS-0050
80	P1A-S010DS-0080
100	P1A-S010DS-0100
125	P1A-S010DS-0125

**Ø16mm - (M5)**

Stroke mm	Order code
10	P1A-S016DS-0010
15	P1A-S016DS-0015
25	P1A-S016DS-0025
30	P1A-S016DS-0030
40	P1A-S016DS-0040
50	P1A-S016DS-0050
80	P1A-S016DS-0080
100	P1A-S016DS-0100
125	P1A-S016DS-0125
160	P1A-S016DS-0160
200	P1A-S016DS-0200

**Ø25mm - (G1/8)**

Stroke mm	Order code
10	P1A-S025DS-0010
15	P1A-S025DS-0015
25	P1A-S025DS-0025
30	P1A-S025DS-0030
40	P1A-S025DS-0040
50	P1A-S025DS-0050
80	P1A-S025DS-0080
100	P1A-S025DS-0100
125	P1A-S025DS-0125
160	P1A-S025DS-0160
200	P1A-S025DS-0200
250	P1A-S025DS-0250
320	P1A-S025DS-0320

**Ø12mm - (M5)**

Stroke mm	Order code
10	P1A-S012DS-0010
15	P1A-S012DS-0015
25	P1A-S012DS-0025
30	P1A-S012DS-0030
40	P1A-S012DS-0040
50	P1A-S012DS-0050
80	P1A-S012DS-0080
100	P1A-S012DS-0100
125	P1A-S012DS-0125
160	P1A-S012DS-0160
200	P1A-S012DS-0200

**Ø20mm - (G1/8)**

Stroke mm	Order code
10	P1A-S020DS-0010
15	P1A-S020DS-0015
25	P1A-S020DS-0025
30	P1A-S020DS-0030
40	P1A-S020DS-0040
50	P1A-S020DS-0050
80	P1A-S020DS-0080
100	P1A-S020DS-0100
125	P1A-S020DS-0125
160	P1A-S020DS-0160
200	P1A-S020DS-0200
250	P1A-S020DS-0250
320	P1A-S020DS-0320

Cylinders are supplied complete with neck mounting and piston rod nuts.

Cylinders with Through piston rods are supplied with two piston rod nuts and one neck mounting nut.

**Sensors**



For sensors see page 77.

**Double acting adjustable cushioning**

**Ø16mm - (M5)**

Stroke mm	Order code
20	P1A-S016MS-0020
25	P1A-S016MS-0025
30	P1A-S016MS-0030
40	P1A-S016MS-0040
50	P1A-S016MS-0050
80	P1A-S016MS-0080
100	P1A-S016MS-0100
125	P1A-S016MS-0125
160	P1A-S016MS-0160
200	P1A-S016MS-0200

**Ø20mm - (G1/8)**

Stroke mm	Order code
20	P1A-S020MS-0020
25	P1A-S020MS-0025
30	P1A-S020MS-0030
50	P1A-S020MS-0050
80	P1A-S020MS-0080
100	P1A-S020MS-0100
125	P1A-S020MS-0125
160	P1A-S020MS-0160
200	P1A-S020MS-0200
250	P1A-S020MS-0250
320	P1A-S020MS-0320

**Ø25mm - (G1/8)**

Stroke mm	Order code
20	P1A-S025MS-0020
25	P1A-S025MS-0025
30	P1A-S025MS-0030
40	P1A-S025MS-0040
50	P1A-S025MS-0050
80	P1A-S025MS-0080
100	P1A-S025MS-0100
125	P1A-S025MS-0125
160	P1A-S025MS-0160
200	P1A-S025MS-0200
250	P1A-S025MS-0250
320	P1A-S025MS-0320



Figura A.6: Hoja de datos 6 de 6.