Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA

Mario Alberto Reyes Calderón
Asesorado por la Inga. Ingrid Salomé Rodríguez de Loukota

Guatemala, octubre de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MARIO ALBERTO REYES CALDERÓN
ASESORADO POR LA INGA.INGRID SALOMÉ RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, OCTUBRE DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Carlos Eduardo Guzmán Salazar
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 11 de febrero de 2019.

Mario Alberto Reyes Calderón

Guatemala 2 de septiembre de 2019

Ingeniero
Julio Cesar Solares Peñate
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Apreciable Ingeniero Solares.

Me permito dar aprobación al trabajo de graduación titulado ¨**Propuesta y desarrollo de prácticas sobre el uso y programación de controladores lógicos programables, para el mejoramiento en el área de automatización de la carrera de ingeniería electrónica**¨, del señor **Mario Alberto Reyes Calderón**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,

Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiado 5356

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 11 de septiembre de 2019

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA,** desarrollado por el estudiante **Mario Alberto Reyes Calderón**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Julio César Solares Peñate
Coordinador de Electrónica

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

REF. EIME 46. 2019.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación de el estudiante: MARIO ALBERTO REYES CALDERÓN titulado: PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA, procede a la autorización del mismo.

Ing. Armando Alonso Rivera Carrillo

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

GUATEMALA, 23 DE SEPTIEMBRE 2019.

Universidad de San Carlos
De Guatemala

Facultad de Ingeniería
Decanato

Ref. DTG.476-2019

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **PROPUESTA Y DESARROLLO DE PRÁCTICAS SOBRE EL USO Y PROGRAMACIÓN DE CONTROLADORES LÓGICOS PROGRAMABLES, PARA EL MEJORAMIENTO EN EL ÁREA DE AUTOMATIZACIÓN DE LA CARRERA DE INGENIERÍA ELECTRÓNICA,** presentado por el estudiante universitario: **Mario Alberto Reyes Calderón,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Aurelia Anabela Cordova Estrada
Decana

Guatemala, Octubre de 2019

AACE/asga
cc

ACTO QUE DEDICO A:

Dios	Padre de todo y dador de vida, sin ti nada sería posible.
Mis padres	Mario Alberto Reyes Sánchez y Marina Alejandra Calderón Roldán, fueron y serán siempre, uno de los pilares más importantes en mi vida.
Mis abuelos	Juan Francisco Calderón, Alejandra Roldán, Carlos Reyes y Antonieta Sánchez, este logro también es de ustedes.
Mis hermanos	En especial a Manuel Calderón, tu motivación fue una guía para mí.
Mi novia	Nelly Tórtola, porque este logro alcanzado, que logramos juntos, sea uno más, de todos los que aún nos faltan cumplir.

AGRADECIMIENTOS A:

Dios	Por permitirme alcanzar esta meta, por brindarme la salud y sabiduría para culminar mis estudios.
Mis padres	Mario Alberto Reyes Sánchez y Marina Alejandra Calderón Roldán, por todos sus consejos, desvelos, amor incondicional y apoyo en cada etapa de mi vida.
Mis abuelos	Por apoyarme y alentarme a seguir adelante.
Mi novia	Nelly Tórtola, por ser una inspiración en mi vida, por todo el apoyo, la fuerza y el amor brindado.
Universidad de San Carlos de Guatemala	Por permitirme el acceso a la educación superior.
Facultad de Ingeniería	Por dejarme crecer de manera personal y profesional.
Escuela de Ingeniería Mecánica Eléctrica	Por darme la oportunidad de trabajar y aprender más, tanto de catedráticos como compañeros, en especial, al Ing. Otto Andrino y a la Inga. Ingrid de Loukota.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
Hz	Hertz
MB	Megabyte
mA	Miliamperio
ms	Milisegundo
s	Segundo
V	Voltio

GLOSARIO

3D	Tercera dimensión.
A/D	Conversión analógica a digital, proceso para convertir valores continuos a valores discretos.
AC	Siglas en inglés para *alternating current*, corriente alterna, flujo de carga eléctrica variante en el tiempo.
Acimut	Ángulo o longitud de arco medido desde el horizonte hacia el sol, medido en grados.
Actuador	Dispositivo capaz de convertir su energía para la habilitación de un proceso, generando un cambio dentro de otro proceso automatizado.
Aguas abajo	Todo elemento o configuración de elementos que se encuentren debajo de un elemento de referencia dentro de un diagrama o programa.
Aguas arriba	Todo elemento o configuración de elementos que se encuentren por encima de un elemento de referencia dentro de un diagrama o programación.

Álgebra de Boole	Sistema matemático que utiliza variables y operadores lógicos para la obtención de sistemas lógicos.
ALU	Siglas en inglés para *arithmetic logic unit*, unidad lógica aritmética capaz de realizar cálculos aritméticos y lógicos.
AND	Compuerta lógica digital, función Y, su salida estará en estado alto únicamente cuando todas sus entradas posean un estado alto.
Asíncrono	Fenómeno en el que una señal de reloj no tendrá el mismo período de operación en estado alto y en estado bajo.
Autómata	Dispositivo electrónico programable, usado para controlar procesos secuenciales.
Bit	Dígito binario, representado por valores discretos, 0 o 1.
Byte	Conjunto de 8 bits, utilizados como unidad en computación y telecomunicaciones.
Conmutación	Hace posible la conexión entre dos circuitos o elementos a través del paso de corriente.

Contactor	Dispositivo electromecánico capaz de habilitar o no el paso de corriente hacia la carga.
CPU	Siglas en inglés para *central processing unit*, unidad central de procesamiento, encargado de interpretar las instrucciones de un programa.
D/A	Conversión digital a análoga, proceso para convertir valores discretos en continuos.
DC	Siglas en inglés para *direct current*, corriente continua, flujo de carga eléctrica constante en el tiempo.
E/S	Forma general para hacer referencia a un conjunto de entradas y salidas que posee el controlador.
EPROM	Siglas en inglés para *erasable programable read only memory*, memoria de solo lectura reprogramable.
Ethernet industrial	Red industrial basada en protocolos de comunicación en tiempo real por medio de Ethernet.
FIFO	Siglas en inglés para *first in first out*, sistema de almacenaje donde el primer dato que ingresa es el primero en salir.
Flanco de bajada	Momento en el que una señal entrante pasa de un estado lógico alto a un estado lógico bajo.

Flanco de subida	Momento en el que una señal entrante pasa de un estado lógico bajo a un estado lógico alto.
Hardware	Parte física de un sistema digital.
Histéresis	Capacidad de un material para mantener alguna de sus propiedades a pesar de no tener el estímulo que las ha generado.
HMI	Siglas en inglés para *human machine interface*, interfaz humano máquina, capaz de establecer la comunicación y el control que el operario tendrá sobre la máquina y los procesos relacionados.
LCD	Siglas en inglés para *liquid cristal display*, pantalla de cristal líquido, permite la visualización de textos en los controladores.
LED	Siglas en inglés para *light emitting diode*, diodo emisor de luz, indicador visual dentro de algunos controladores y procesos.
Leva	Elemento mecánico capaz de convertir el movimiento circular en un movimiento rectilíneo.
LIFO	Siglas en inglés para *last in first out*, sistema de almacenaje donde el último dato en ingresar es el primero en salir.

MMI	Siglas en inglés para *man machine interface*, interfaz de comunicación entre el hombre y la máquina por medio de elementos de entradas físicas.
MODBUS	Protocolo de comunicación industrial desarrollado en 1979, que permite la interacción entre dispositivos de automatización.
NOT	Compuerta lógica digital, función negada, de una entrada y una salida, en salida tendrá el estado negado de la entrada.
OFFSET	Valor de tensión que se suma a otra señal de tensión para obtener una ganancia o un desplazamiento en amplitud de la misma.
OR	Compuerta lógica digital, función O, su salida tendrá un estado lógico alto cuando por los menos una de sus entradas se encuentre en alto.
Orto	Momento en que el sol u otro astro sale por el horizonte.
PID	Siglas para control proporcional, integral y derivativo, sistema de control simultaneo por medio de realimentación.

PLC	Siglas en inglés para *programable logic controller*, controlador lógico programable o autómata programable.
Programación CNC	Siglas para control numérico por computadora, sistema capaz de controlar la posición de un elemento físico.
PWM	Siglas en inglés para *pulse width modulation*, modulación por ancho de pulso, técnica para alterar el ciclo de trabajo de una señal cuadrada.
RAM	Siglas en inglés para *ramdom access memory*, memoria de acceso aleatorio, encargada de almacenar los datos utilizados en el momento presente.
Realimentación	Parte de un sistema, encargada de llevar el dato de salida hacia la entrada nuevamente para que sea tratado.
Redundancia	Medida de seguridad en la que se tendrá por lo menos un equipo o componente de respaldo para afrontar situaciones de fallo.
Relé	Dispositivo electromagnético que al ser excitado cambiará de posición entre sus contactos, que permite el paso de corriente.

ROM	Siglas en inglés para *read only memory*, memoria de solo lectura, los datos guardados no se pueden modificar, solo leer, sin importar si hay o no alimentación.
Run	Palabra en inglés, verbo, ejecutar.
Rung	Palabra en inglés, peldaño o escalón, hace referencia a cada línea horizontal de programación en lenguaje de escalera.
SCADA	Siglas en inglés para *supervisory control and data acquisition*, supervisión, control y adquisición de datos, se le denomina así a cualquier aplicación que obtenga datos operativos de un sistema.
Sensor	Dispositivo capaz de variar una magnitud física como voltaje, corriente o resistencia ante un fenómeno físico existente.
Síncrono	Fenómeno en el que una señal de reloj tendrá el mismo período de operación en estado alto y en estado bajo.
Sistema combinacional	Aplicación en la que todas sus salidas dependerán únicamente de los valores que posea en sus entradas.

Sistema secuencial	Aplicación en la que los estados de las salidas dependen de los estados presentes en las entradas y también de los estados anteriores.
SMS	Siglas en inglés para *short message service*, servicio de mensajes cortos, disponible para teléfonos móviles y otros dispositivos.
Software	Parte intangible, no visible, de un sistema, hace referencia a todos los programas existentes, guardados en memoria.
Stop	Palabra en inglés, verbo, detener.
Transductor	Dispositivo capaz de transformar un fenómeno físico como presión, temperatura, entre otros, a una señal eléctrica.
Umbral	Valor mínimo requerido que debe tener una señal para para ser reconocida por un dispositivo, en algunos casos puede ser predeterminado por el usurario.
UTC	Siglas en inglés para *universal time coordinated*, tiempo universal coordinado, estándar con el que se regula el tiempo en el mundo.

Volátil	Elemento que cambia o varía con facilidad, en el caso de las memorias, la información se elimina en ausencia de alimentación.

RESUMEN

En el presente trabajo de graduación se desarrolla el material teórico y práctico para que personas, estudiantes y profesionales se enfoquen en el aprendizaje de la automatización, del controlador lógico programable y el uso del software de programación en dos lenguajes.

El primer capítulo describe la historia sobre la automatización para dar paso a la automatización en la actualidad; describe la arquitectura y los componentes básicos de un sistema automatizado.

En el segundo capítulo se describen los fundamentos sobre controladores lógicos programables, hace énfasis en la estructura externa e interna del controlador.

El capítulo tres describe las diferentes funciones y los lenguajes de programación para PLC, basados en el estándar IEC 61131-3.

En el capítulo cuatro se desarrolla una guía para la descarga, la instalación y el uso del software de programación para PLC.

El capítulo cinco enuncia ocho prácticas a realizarse: cuatro en lenguaje escalera y cuatro en lenguaje de bloques de funciones, cada una con su solución.

OBJETIVOS

General

Desarrollar prácticas de laboratorio sobre el uso y la programación de controladores lógicos programables.

Específicos

1. Introducir, en especial, a los estudiantes y profesionales al tema de la automatización.

2. Desarrollar un conjunto de fundamentos para facilitar la comprensión de los temas y encaminar la realización de las prácticas.

3. Facilitar una guía para el uso e instalación del software necesario para llevar a cabo las prácticas planteadas.

4. Crear y realizar las prácticas, para que el estudiante se familiarice y comprenda el uso y la programación de un controlador lógico programable.

INTRODUCCIÓN

El requerimiento de los sectores industriales, alimenticios, automovilísticos y en general, de producción, ha sido el de agilizar procesos, así como elevar la productividad, sin perder la calidad de sus productos, lo que ha provocado que se busque su automatización.

Esta situación ha dado como resultado la implementación de distintos elementos, como sustitutos de la mano de obra humana, para realizar, sin pausas, los procesos necesarios para culminar un producto. Estos elementos tienen la característica de que no pueden actuar por si solos; entonces, necesitan la ayuda de una variedad de sensores que determinarán las regiones, puntos y niveles sobre los que deben actuar dichos elementos. Aun así, con la existencia de los distintos elementos y sensores, no está completa la automatización de un proceso, ya que, hace falta la parte más importante, el cerebro de toda la operación; el elemento que será capaz de analizar cada una de las señales del proceso y a partir de esto podrá tomar las decisiones correspondientes a cada uno de los eventos posibles que puedan ocurrir dentro del proceso.

Entonces, el cerebro de toda la operación es el controlador lógico programable (PLC, por sus siglas en inglés) que contendrá guardada toda la programación necesaria para llevar a cabo todo el proceso requerido, por medio de la toma de decisiones.

En los últimos años las industrias han ido cambiando la manera como realizan sus procesos, con la implementación de la automatización, con el uso

de PLC, por lo tanto, el campo de trabajo ha ido creciendo, que hace más notoria la necesidad de profesionales que conozcan y dominen el tema, que cubren la demanda de trabajo.

Este trabajo está dirigido a profesionales y estudiantes que necesiten aprender o recordar los conceptos básicos sobre los elementos y programación de PLC, con el fin de adquirir los conocimientos necesarios en el tema de programación y los usos de los controladores lógicos programables.

1. AUTOMATIZACIÓN INDUSTRIAL

1.1. Antecedentes históricos

En la actualidad, cuando se habla de automatización, es imposible no pensar en grandes máquinas, robots y procesos totalmente automatizados, pero, para que todos estos avances ocurrieran, fue necesaria una variedad de antecedentes, entre estos se pueden mencionar:

- 1947: en los laboratorios Bell se desarrolla el primer transistor, gracias a los físicos John Bardeen, Walter Brattain y William Shockkley.

- 1959: se utilizó el primer controlador para una herramienta de maquinado, controlado por computadora y la lógica aún era a través de cableado.

- 1968: da inicio la historia del PLC con el control industrial modular de Dick Morley.

- 1978: antes de las computadoras los dispositivos de programación eran grandes y pesados, pero con la llegada de las computadoras aparece la programación CNC a nivel de máquinas.

- 1987: dentro de la industria fue notable que el uso de computadoras, no sería únicamente para recolectar datos, por lo que dio inicio la era de las computadoras en la automatización.

- 1997: se desarrollan más controles descentralizados e inteligentes y a su vez se hace posible que la variedad de componentes puedan comunicarse entre sí, dando paso al Ethernet Industrial. En este mismo año da inicio la creación y el uso de simuladores para el desarrollo de diseños de control de procesos.

- En la actualidad, ya es posible automatizar cualquier clase de tarea, teniendo control sobre ella; incluye nuevas herramientas como el uso de bases de datos en nubes; se implementan, incluso, el internet de las cosas.

Dados los antecedentes es posible darse cuenta de que la automatización ha existido desde hace mucho tiempo, en aplicaciones pequeñas, pero gracias a los avances tecnológicos que han ocurrido ha sido posible su implementación en tareas más complejas, todo debido al uso y evolución de las computadoras.

1.2. La automatización en la actualidad

La automatización se ha convertido en una parte esencial dentro de la industria, ya que, con su implementación es posible controlar gran variedad de procesos sin perder su calidad y se logra una reducción de tiempo en su realización.

El aumento en la automatización de procesos se debe también al hecho de que la ingeniería se mantiene en un constante crecimiento y actualización tecnológica y de infraestructura, lo que permite realizar controles y procesos automatizados, que provocan, a su vez, que la automatización de actualice a las necesidades de la industria.

La industria actual, cada vez más grande y competitiva, busca en la mayoría de los casos reemplazar la mano de obra humana para optimizar sus procesos; involucra una gran variedad de elementos, sistemas y controles, por lo que el campo de aplicación para la automatización se expande y por esa razón también se vuelve necesario que antes de cualquier implementación o actualización es importante conocer las necesidades de automatización que se van a satisfacer así como las complicaciones que puedan surgir en el proceso.

Así como se ha incrementado la necesidad de automatizar procesos, también se ha incrementado la seguridad con la que se realizan los trabajos ya que los equipos cuentan con más seguridad, incluyendo la seguridad del personal. Se tiene la posibilidad de generar copias de seguridad y en algunos casos es posible reemplazar equipos sin necesidad de parar el proceso que se está realizando.

1.3. Arquitectura y componentes

Un sistema o proceso automatizado se puede dividir en partes para facilitar la comprensión de su funcionamiento.

1.3.1. Interacción y monitorización

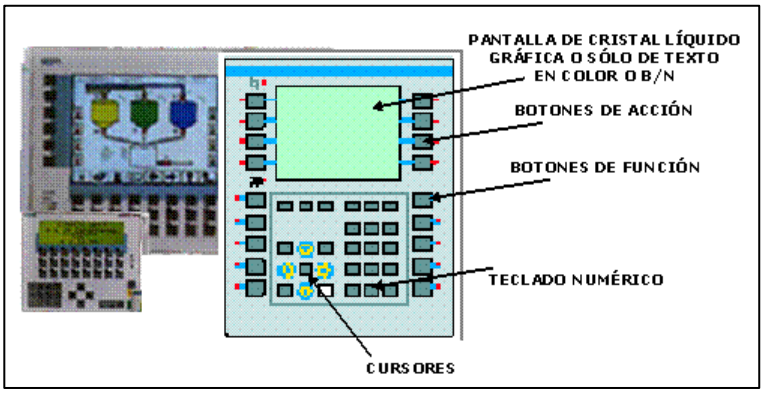
Esta parte es la que protagoniza el operador humano, realizando la supervisión en cada puesto con interfaz disponible, HMI, MMI, SCADA, para el control de uno o varios procesos dentro de un sistema automatizado.

La interacción puede darse con el uso de botoneras, paneles de membrana y pantallas táctiles, por su parte la monitorización puede ser a través de sistemas SCADA, a distancia y por teleoperación.

1.3.1.1. Botonera y panel de membrana

Es un panel que cuenta con un teclado de letras y números así como una pantalla que puede ser a color o monocromática, por ejemplo:

Figura 1. **Panel de membrana**



Fuente: Uniovi. *Paneles de operador de membrana*. http://isa.uniovi.es/domotica/Temas/T2/T2-PanelesOperadorMembrana.htm. Consulta: 3 de abril de 2019.

1.3.1.2. Pantalla táctil

Es una terminal para el operador, por lo que se convierte en una interfaz entre el humano y la máquina; la mayoría de dispositivos tienen la capacidad de comunicarse con distintos controladores al mismo tiempo por medio de diferentes protocolos de comunicación, por ejemplo:

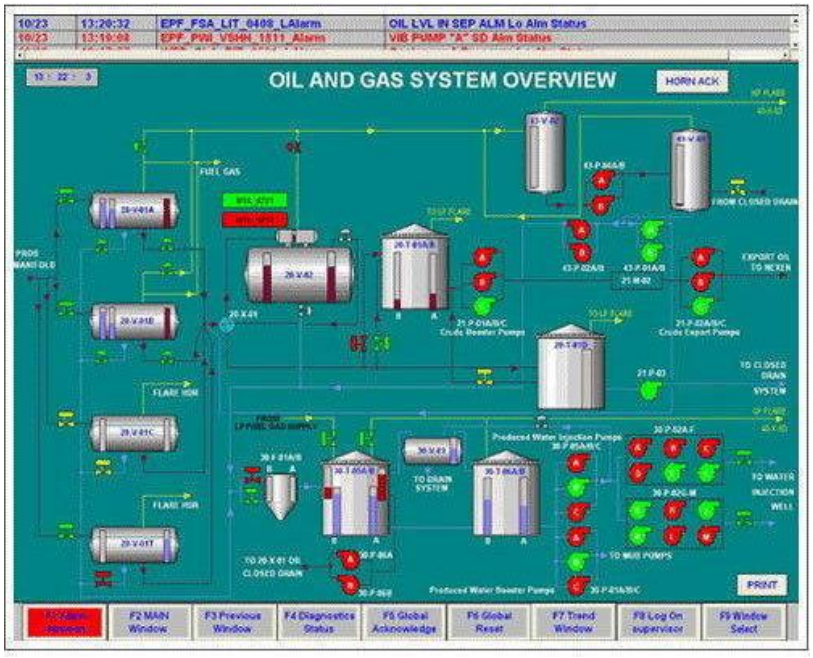
Figura 2. **PLC, Unitronics, con pantalla táctil integrada**

Fuente: Tecnelsat. *Autómatas Unitronics*. http://tecnelsat.com/?page_id=95. Consulta: 3 de abril de 2019.

1.3.1.3. Sistema SCADA

Es una aplicación que utiliza distintos recursos para supervisar, controlar y adquirir datos de un sistema o proceso y así poder optimizarlo, con la capacidad de recibir y transmitir datos, manipular base de datos, al igual que la representación gráfica de los datos, por ejemplo:

Figura 3. **Schneider SCADA para un sistema de aceite y gas**



Fuente: Indiamart. *Sistemas de control de proceso y equipos*. https://www.indiamart.com/proddetail/schneider-scada-systems-13202900712.html. Consulta: 3 de abril de 2019.

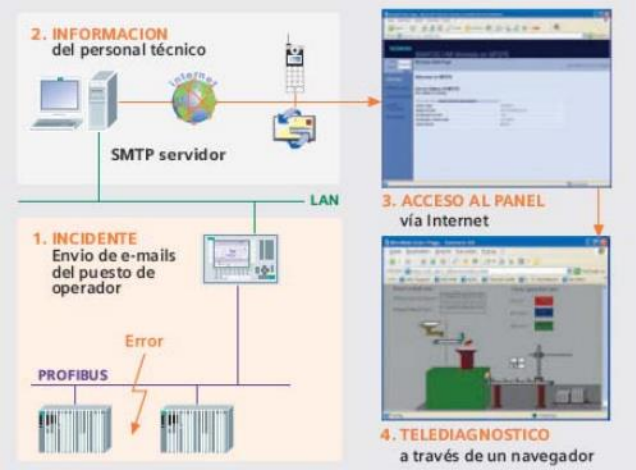
1.3.1.4. Supervisión a distancia

Existen tres métodos para la supervisión a distancia, por medio de internet, mensaje de texto y teleoperación.

1.3.1.4.1. Por medio de internet

Con el uso de redes y servidores es posible notificar al personal sobre problemas o incidentes dentro del proceso y a su vez es posible el acceso de manera remota vía internet por medio de un navegador y dar solución al problema, un ejemplo se observa en la figura 4.

Figura 4. **Control a distancia vía internet**

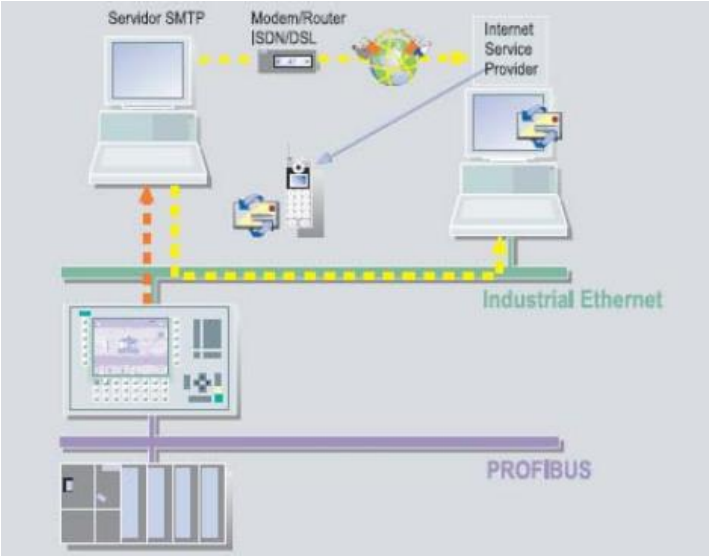


Fuente: TORRES, C. Jara. *Tema 10, Interacción y supervisión, F*. https://rua.ua.es/dspace/bitstream/10045/18439/1/Tema%2010_Interacci%C3%B3n%20y%20supervisi%C3%B3n.pdf. Consulta: 3 de abril de 2019.

1.3.1.4.2. Por medio de SMS

Incluye la utilización de internet para poder enviar los mensajes de a texto a las terminales telefónicas para poder avisar al personal sobre el estado del o los sistemas que se encuentra bajo monitoreo. Para ambos medios de monitorización la desventaja es que si en algún momento se llega a perder la señal de internet o el servicio telefónico el personal queda desinformado; por lo tanto, es importante tener siempre redundancia dentro de los sistemas. Un ejemplo de una supervisión por SMS se puede observar en la figura 5.

Figura 5. **Control a distancia vía SMS**



Fuente: TORRES, C. Jara. *Tema 10, Interacción y supervisión, F*. https://rua.ua.es/dspace/bitstream/10045/18439/1/Tema%2010_Interacci%C3%B3n%20y%20supervisi%C3%B3n.pdf. Consulta: 3 de abril de 2019.

1.3.1.5. Teleoperación

Este método brinda la oportunidad de realizar las tareas a distancia, en tiempo real, pero con la necesidad de tener operadores, esclavos, interfaces, canales de comunicación y sensores para realizar las tareas deseadas.

1.3.2. Sistemas de control

Un sistema es la combinación de componentes capaces de interactuar entre sí para cumplir un determinado objetivo; por otra parte, al hablar de un sistema de control, se hace referencia a un conjunto de componentes que

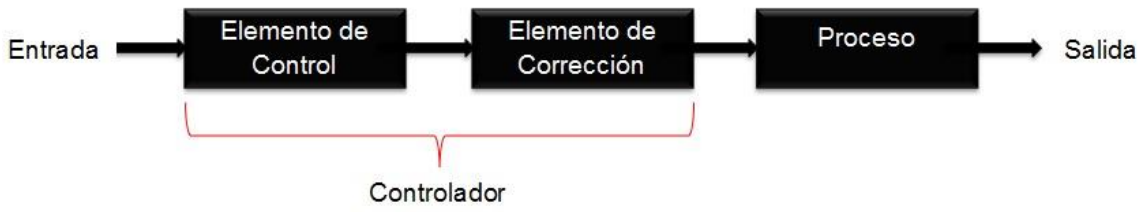
pueden regular su propia conducta o la de otro sistema y lograr un funcionamiento predeterminado, que reduce la posibilidad de fallos.

Para un sistema de control existe una variable de entrada, que con el hecho de cambiar su magnitud o condición puede alterar el estado del sistema, y existe una variable de salida, que pertenece al sistema pero su magnitud o condición es la que se mide por el sistema. Hay dos tipos de sistemas de control.

1.3.2.1. Sistema de control de lazo abierto

Esta clase de sistema de control carece de sensores que informen al sistema acerca del estado en que este se encuentra; a pesar de eso, el proceso se lleva a cabo sin comprobar su correcta realización.

Figura 6. **Diagrama de bloques para un sistema de lazo abierto**



Fuente: elaboración propia.

- Elemento de control: es la acción a tomar dada una entrada.

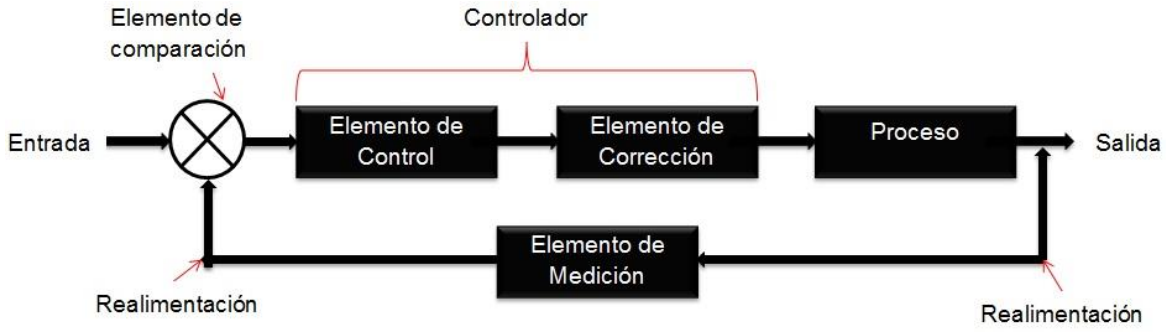
- Elemento de corrección: inicia la acción para llegar al valor requerido por el sistema.

- Proceso: llamado planta, es el sistema al que se le controla la variable.

1.3.2.2. Sistema de control de lazo cerrado

Este sistema posee sensores de vigilancia para el proceso automatizado, capaces de controlar la salida del sistema y a su vez llevar esta información hacia la entrada.

Figura 7. **Diagrama de bloques para un sistema de lazo cerrado**



Fuente: elaboración propia.

- Elemento de comparación: es el encargado de comparar el valor requerido con el valor medido que se obtuvo en salida, que produce la señal de error.

- Elemento de control: es el que toma una decisión dependiendo de la señal de error.

- Elemento de corrección: realiza los cambios en el proceso para eliminar el error.

- Proceso: sistema donde se controla la variable.

- Elemento de medición: produce una señal relacionada con la condición de la variable controlada y da la señal de realimentación.

1.3.2.3. Tecnologías en sistemas de control y tipos de señales

- Tecnología cableada

    - Por medio de tecnologías mecánicas, neumáticas e hidráulicas
    - Uso de sistemas electrónicos combinacionales y secuenciales
    - Dispositivos como relés y contactores

- Tecnología programada

    - Autómatas programables
    - Computadoras industriales
    - Sistemas empotrados

- Tipos de señales

    - Analógicas

        - Señales continuas
        - Señal análoga muestreada y digitalizada para su tratamiento

- Lógicas
    - Señales binarias, valores lógicos, como 0 o 1 bit
- Digitales
    - Agrupación de señales binarias
    - Bits
    - Bytes

1.3.3. Parte operativa

La parte operativa de un sistema es la que se relaciona directamente con las máquinas, es decir, todos los componentes capaces de mover la máquina para que ésta realice la tarea solicitada. Los componentes que la conforman son los transductores, sensores y los actuadores.

1.3.3.1. Transductores

- Transductor de dos estados: brinda en salida únicamente valores entre 0 y 1.
- Transductor binario: se obtiene en salida valores a partir de combinaciones binarias.
- Transductor análogo: en salida aparece una señal continua, la cual varía conforme el fenómeno físico que se mide.

1.3.3.2. Sensores

Los sensores son dispositivos que se encargan de medir una variable física y el dato medido lo envían al controlador y el mismo toma la decisión correspondiente. Existen dos grandes tipos de sensores

- Activos: este tipo de sensor será capaz de entregar en salida una variación de voltaje o de corriente dependiendo de la medición, siendo en pocas palabras un transductor.

- Pasivos: sensor que será capaz de variar únicamente su valor resistivo, capacitivo o inductivo, dependiendo de la medición que realice. Por lo tanto, necesita de un circuito externo o agregado capaz de generar una variación de voltaje o corriente con respecto a su propia variación.

Dentro de los sistemas automatizados algunos de los sensores utilizados son:

- Resistivos: aquellos dispositivos que dependiendo de la variable medida adoptarán cierto valor resistivo, como una fotorresistencia, con base en la luz incidente, un termistor, dependiendo de la temperatura.

- Capacitivos: funcionan con respecto a la variación de la distancia entre sus placas y por lo tanto del campo eléctrico, ya sea en caso industrial, con la percepción de objetos o en comunicaciones con la variación en las frecuencias de las señales.

- Inductivos: funcionan con respecto a la variación de su propio campo magnético para la detección de objetos metálicos.

- Piezoeléctricos: proporcionan un valor de voltaje en salida, proporcional a la fuerza mecánica que se aplica sobre sus placas.

- Ópticos: su funcionamiento depende de la emisión y recepción de un haz de luz; pueden dividirse en:

    - Óptico de barrera: emisor y receptor se encuentran colocados uno enfrente del otro.

    - Óptico por difusión: emisor y receptor se encuentran contiguos entre sí esperando que el haz de luz se refleje y retorne al receptor.

    - Óptico por difracción: emisor y receptor se encuentran contiguos entre sí, pero el haz de luz es reflejado en un prisma, regresando al receptor.

- Ultrasónicos: se encargan de medir una distancia utilizando ondas ultrasónicas, basándose en el tiempo que tarde en viajar la onda hacia el objeto y regresar al mismo sensor.

- De posición: son capaces de indicar la posición de determinado objeto, ofreciendo en salida un código, que en su mayoría de casos son binarios.

1.3.3.3. Preaccionadores

Estos dispositivos se encuentran previos a los accionadores o actuadores que estén dentro del sistema automatizado y son los encargados de realizar la

conmutación entre los circuitos lógicos y los de potencia para una acción eléctrica, neumática o hidráulica.

1.3.3.4. Actuadores

Incluidos dentro del diseño y operando dentro del proceso, deben de soportar una variedad de condiciones dentro de la planta y con base en las condiciones de operación se determina la mejor opción para realizar la operación.

- Eléctricos
- Neumáticos
- Hidráulicos
- Térmicos
- Piezoeléctricos
- Luminosos

1.4. Aplicación de un sistema automatizado

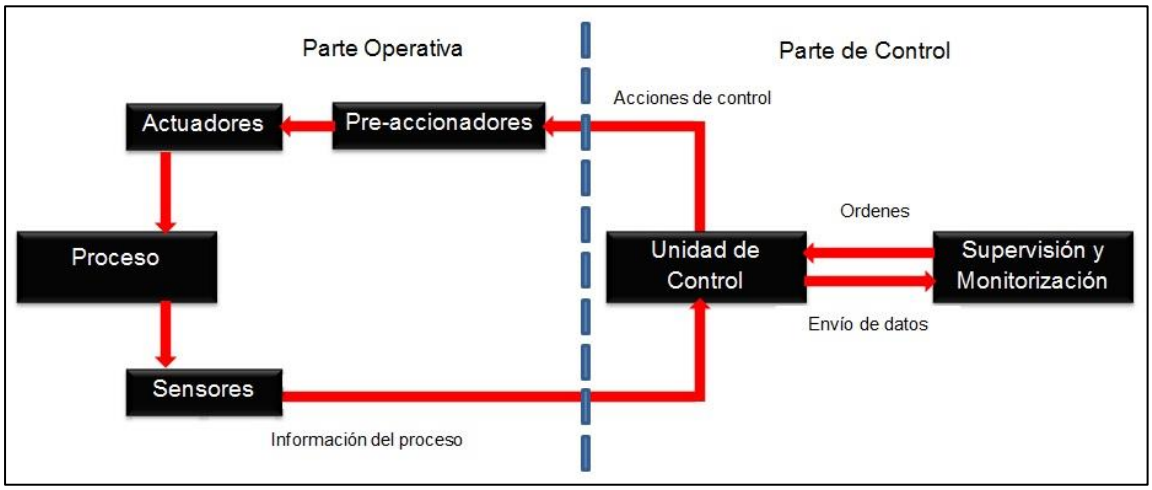
Un sistema automatizado contiene dos partes claramente definidas, la parte de mando o de control y la parte operativa, cada una con sus componentes internos.

- Parte de control
    - Supervisión o monitorización
    - Unidad de control

- Parte operativa

    - Sensores
    - Preaccionadores
    - Actuadores
    - Proceso

Figura 8. **Diagrama de bloques de un sistema automatizado**



Fuente: elaboración propia.

En la figura 8 se muestra una forma general de las partes que siempre debe llevar un sistema automatizado, pero cada proceso tiene sus propias complicaciones y necesidades; por lo tanto, también, se deben analizar otros parámetros, incluyendo el lugar de trabajo.

- Lugar de operación

    - Conocer la planta o proceso

- Experimentar las condiciones de operación
- Tener, con claridad, las necesidades del proceso
- Listado de variables a controlar

- Diseño

  - Grado de automatización

    - Verificación del proceso: es aplicada con el fin de observar el proceso pero sin intervenir dentro del mismo, únicamente informando al operador el estado del proceso y que este controle su correcto funcionamiento.

    - Semiautomatizado: dado el estado del proceso, la parte de control envía sugerencias de acciones a realizar hacia el operador, ayudando en la toma de decisiones, operando en un sistema de lazo abierto.

    - Automatizado: todo el proceso se encuentra automatizado y el operador ya solo es informado sobre las acciones realizadas, teniendo la posibilidad de actuar en caso de emergencia y anular la automatización y operar en modo manual.

  - Nivel de automatización

    - Nivel 1, por operación: la automatización se aplica solo a operaciones específicas que realizan partes mecánicas.

        - Nivel 2, por máquina: se automatizan las operaciones que solo pueden ser realizadas por una o varias máquinas.

        - Nivel 3, por proceso: la automatización se aplica a varias tareas, realizadas por distintos dispositivos, dentro de un sub-proceso de la línea.

        - Nivel 4, nivel integrado: todas los procesos o tareas dentro de toda la línea pasan a ser automatizados, controlados por una o varias unidades de control.

    - Seguridad
    - Redundancia
    - Tecnologías aplicadas

        - Cableada
        - Programada

    - Sensores y actuadores a utilizar
    - Tipos de programación, algoritmos y tipos de control
    - Tipo de equipo a utilizar, hardware y software

- Implementación

    - Simulación
    - Aplicación física
    - Detección de errores
    - Calibraciones

      Pruebas de seguridad

2. FUNDAMENTOS SOBRE LOS CONTROLADORES LÓGICOS PROGRAMABLES

Así como la automatización de procesos tiene una historia, la creación y el uso de los autómatas programables también la tiene.

Antes de los años 60, el control de procesos estaba basado en grandes circuitos, costosos y complicados, basados únicamente en relés; con el fin de eliminar este tipo de control, se buscó una solución, que fue el uso de los PLC, los cuales aparecen durante la misma década.

La necesidad de su creación se basó en que estos nuevos dispositivos de control permitieran una fácil programación, que fueran capaces de operar en ambientes difíciles y que tuvieran una rápida respuesta; sustituye los elementos mecánicos con elementos de estado sólido, sin poseer habilidades de comunicación.

Una década después, los PLC ya podían comunicarse entre sí, logra repartir o dividir las acciones de todo un proceso en varios controladores, siendo su primer método de comunicación el MODBUS; al mismo tiempo, tenían la capacidad de enviar y recibir voltajes; con el uso de señales análogas.

Durante los años 80 se intentó estandarizar los sistemas de comunicación, al igual que se buscó la reducción de tamaños en los dispositivos, lastimosamente en la actualidad, aun no existe un protocolo de comunicación estándar para los controladores.

En los años 90 se aplicaron mejoras a la mayoría de protocolos que venían de los años 80 y se buscó la unificación de los diferentes tipos de lenguajes dentro de un estándar de programación mundial.

En la actualidad, los PLC ofrecen una variedad en la forma de su programación, al igual que la posibilidad de mezclar los tipos de programación dentro de una sola aplicación; lastimosamente, aún hay variedad de protocolos de comunicación por lo que se dificulta el uso de dispositivos de distintas marcas; pero ya en la mayoría de marcas es posible acceder a la red y utilizar la misma, así como la posibilidad de utilizar y sustituir memorias con los programas ya cargados y operar desde la misma y ofrecen la reducción de tiempo perdido por paros en el proceso ya que ahora es posible cambiar o sacar de operación el controlador y sustituirlo por otro sin parar el proceso.

2.1. Controlador lógico programable

El autómata programable o controlador lógico programable es un dispositivo con la capacidad de programar su memoria con una variedad de instrucciones como operaciones lógicas, temporizadores, incluso aritméticas en donde la realización de cada una de las instrucciones se da de forma secuencial; responde de forma rápida y en tiempo real.

De forma muy general, lo que hace un PLC es:

- Tomar los datos que aparecen en sus entradas, ya sean digitales o análogos.
- Tomar decisiones a partir de sus instrucciones programadas.
- Guardar los datos en su memoria.
- Generar temporizadores.

- Cálculos aritméticos y lógicos.
- Gobernar dispositivos externos.
- Comunicarse con otros dispositivos.

2.2. Aplicación de los PLC

Dadas las características propias del PLC y su capacidad de operar en ambientes difíciles hace que su área de aplicación sea amplia; llega a cubrir necesidades como:

- Espacios reducidos
- Procesos industriales cambiantes
- Procesos secuenciales
- Maquinaria multipropósitos
- Procesos complejos

Algunas aplicaciones son:

- Procesos de maquinaria
    - Procesos de cemento
    - Procesos sobre muebles y madera
    - Procesos de plásticos
    - Procesos de ensamblaje
- Procesos de instalación
    - Relacionados a seguridad
    - Calefacción y aire acondicionado

- Embotellamiento
- Transporte y almacenaje
- Aplicaciones térmicas
- Procesos azucareros

- Industria automovilística y de neumáticos

  - Procesos de montaje y soldadura
  - Aplicación de pintura a piezas
  - Uso de herramientas, tornos, cargadores
  - Extrusión de goma
  - Realización de cubiertas
  - Refrigeración
  - Calderas
  - Vulcanización

- Plantas

  - Oleoductos
  - Refinamiento
  - Tratamientos de aguas
  - Pesaje
  - Dosificación
  - Mezclas

- Otras aplicaciones

  - Metalurgia
  - Alimentación

- Aserraderos y papeleras
- Generación de energía
- Control de tráfico
- Domótica

2.3. Ventajas y desventajas de un PLC

Se presentan algunas de las ventajas y desventajas de un PLC.

2.3.1. Ventajas

- Reducción de tiempo en la realización de proyectos.
- Baja necesidad de materiales o equipos extra en la aplicación del proyecto.
- Dada su facilidad para modificar las instrucciones se evita el cambio de cableado y la introducción de equipo nuevo.
- Espacio requerido para instalación reducido.
- El costo en la mano de obra se ve reducido ya que la mayor parte del proceso es controlado por el autómata.
- Reducción en gastos por mantenimiento.
- Capacidad de operar más de una herramienta con un solo controlador.
- Operación en ambientes complicados.

- Posibilidad de sustituir un controlador por otro sin apagar y sin parar el proceso.

- Reducción en pérdidas monetarias por paro en los procesos.

2.3.2. Desventajas

- Uso de una sola línea o marca de controladores y sus derivados, debido a que los protocolos de comunicación no son estándar entre todos los dispositivos y controladores existentes.

- El costo inicial de inversión es elevado por los cambios que se deben hacer dentro de la planta y sobre todo por lo adquisición del equipo necesario para pasar de procesos manuales a procesos automatizados.

2.4. Estructura de un PLC

En la actualidad existe una gran variedad de marcas dedicadas a la automatización, al igual una gran cantidad de modelos para los controladores, lo que permanece constante es su estructura; por lo tanto, se puede separar por hardware, software, estructura externa y estructura interna.

2.4.1. Hardware

En forma general, es la parte física, lo que se puede ver, en el caso del controlador, su hardware está conformado por la fuente de alimentación, batería, módulos de memoria, módulos de entradas y salidas análogas y digitales, la CPU, entre otras cosas.

2.4.2. Software

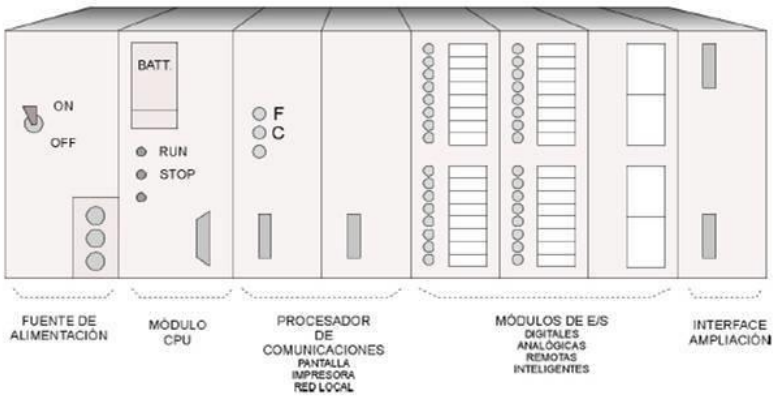
El software, por su parte, es el que no se puede ver; por ejemplo, los programas que se almacenan dentro de las memorias, rutinas guardadas en otras memorias del sistema.

2.4.3. Estructura externa

Es la que corresponde a la batería o fuente de alimentación, los módulos de entradas y salidas, luces indicadoras, puertos de comunicación, módulos de ampliación, incluso el chasis del propio controlador.

La alimentación de los controladores para la CPU pude ser de 24 V de DC o de 110 V a 220 V de AC, para los módulos restantes es la propia CPU la encargada de darles alimentación por medio del bus interno del controlador.

Figura 9. **Estructura externa básica del PLC**



Fuente: Wordpress. *Automatización industrial mediante PLCs*. https://davidrojasticsplc.wordpress.com/2009/01/14/arquitectura-y-apariencia-externa/. Consulta: 8 de abril de 2019.

2.4.4. Estructura interna

La estructura interna del controlador es la que involucra el procesador o CPU, las interfaces de entradas y salidas, módulos de memorias, buses de comunicación, entre otras cosas.

2.4.4.1. Procesador

La unidad de control es la encargada de ejecutar el programa que se ha cargado al PLC, realizando otras tareas como, manejar la comunicación el dispositivo de programación y la memoria, las interfaces de E/S, al igual que ejecutar programas de diagnóstico, pero todo es a partir de un sistema operativo cargado previamente por el fabricante y almacenado en una memoria no volátil. El procesador posee:

2.4.4.1.1. Acumulador

Encargado de guardar el último valor entregado por la ALU, por lo que se encuentra ligado a la misma y al bus de datos.

2.4.4.1.2. ALU

Unidad que se encarga de realizar las operaciones aritméticas básicas que se necesitan: suma, resta, multiplicación y división así como algunas operaciones lógicas como AND, OR y NOT. Las instrucciones u operaciones a realizar son enviadas por el procesador a la ALU y esta le devuelve los resultados al procesador para que los analice y envíe nuevas instrucciones a otros dispositivos.

Figura 10. **Interacción entre CPU y ALU**

Fuente: elaboración propia.

2.4.4.1.3. Contador de programa

Conocido también como puntero de instrucciones, es el encargado de guardar la dirección de la última instrucción ejecutada o leída y de esa forma conocer cuál es la instrucción que sigue dentro del programa.

2.4.4.1.4. Decodificador de instrucciones y secuenciador

Son parte de la unidad de control y trabajan en conjunto, cuando se va a realizar una acción ésta es leída por el decodificador, el cual a partir de la decodificación sabrá qué clase de instrucción es y qué tipo de operación se realizará, por su parte el secuenciador, con el uso del bus de control enviará las señales de control hacia cada componente que deberá actuar en su momento.

2.4.4.1.5. Banderas

Son posiciones reservadas de memoria, son valores binarios de uno hasta 64 bits, encargados de indicarle al dispositivo si ocurre algún suceso específico, por ejemplo:

- CF: por sus siglas en inglés, *carry flag*, es la bandera de acarreo, aparece cuando se realizan operaciones entre valores binarios, muestra un 1 lógico cuando existe acarreo y un 0 lógico en caso contrario.

- PF: por sus siglas en inglés, *parity flag*, es la bandera de paridad, si existe un número impar de bits, mostrará un 0 lógico y si llega a ser par, regresara a un 1 lógico.

- AF: por sus siglas en inglés, *auxiliar flag*, es una bandera de acarreo auxiliar, si existe un acarreo extra la bandera pasa a un 1 lógico de lo contrario se mantiene en 0 lógico.

- ZF: por sus siglas en inglés, *zero flag*, es la bandera de cero, la cual si detecta un cero, pasa su estado a 1 lógico y si detecta un valor distinto de cero, pasará a su estado de 0 lógico.

- SF: por sus siglas en inglés, *sign flag,* es la bandera de signo, luego de una operación, si el resultado es de signo positivo, pone en 0 lógico la bandera y si es un resultado de signo negativo, el estado pasa a 1 lógico.

- TF: por sus siglas en inglés, *trap flag*, es la bandera de trampa, cuando está en 1 lógico, el procesador pasa al modo de paso único, ejecutando una instrucción a la vez.

- IF: por sus siglas en inglés, *interruption flag*, bandera de interrupción que permite una interrupción solo si esta en 1 lógico de lo contrario no habrá interrupción.

- DF: por sus siglas en inglés, *direction flag*, es la bandera de dirección, indica la circulación de datos, si su estado es 0 lógico, la transferencia de datos será de izquierda a derecha, en el caso de que sea 1 lógico la transferencia de datos será de derecha a izquierda.

- OF: por sus siglas en inglés, *overflow flag*, es la bandera de desbordamiento, indica si existe un acarreo interno y externo en el bit de signo luego de alguna operación realizada.

Las banderas también se pueden dividir por categoría, estado, control y del sistema; al mismo tiempo, es necesario saber que dentro de los 64 bits disponibles hay espacios reservados para otros usos.

Tabla I. **Resumen de banderas**

# Bit	Abreviatura	Bandera	Categoría
0	CF	Bandera de acarreo	Estado
1		Reservado	
2	PF	Bandera de paridad	Estado
3		Reservado	
4	AF	Bandera de acarreo auxiliar	Estado
5		Reservado	
6	ZF	Bandera de cero	Estado
7	SF	Bandera de signo	Estado
8	TF	Bandera de trampa	Control
9	IF	Bandera de interrupción	Control
10	DF	Bandera de dirección	Control
11	OF	Bandera de desbordamiento	Estado

Fuente: elaboración propia.

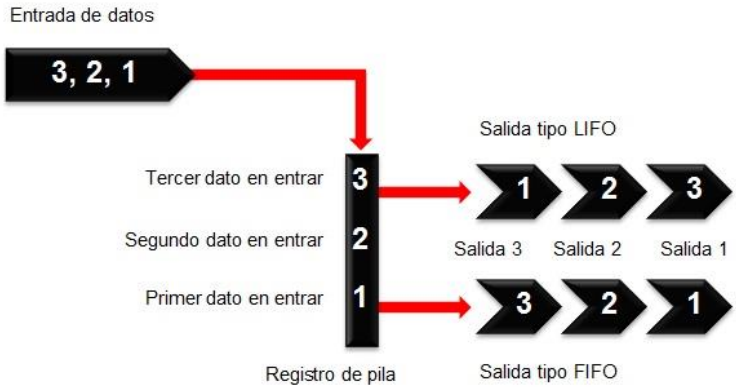
2.4.4.1.6. Monitor del sistema

Este componente de la unidad de control es el encargado de guardar la secuencia de inicio del controlador, rutinas de prueba, errores de ejecución; también, puede ser capaz de informar sobre el programa que se está realizando, tiempo, memoria, uso del procesador y espacio disponible.

2.4.4.1.7. Pila

Conocido como registro de pila, es una memoria de almacenamiento temporal, su único acceso es en la parte superior de la pila y su comportamiento es el de una estructura de datos tipo secuencial, LIFO y en algunos casos del tipo FIFO.

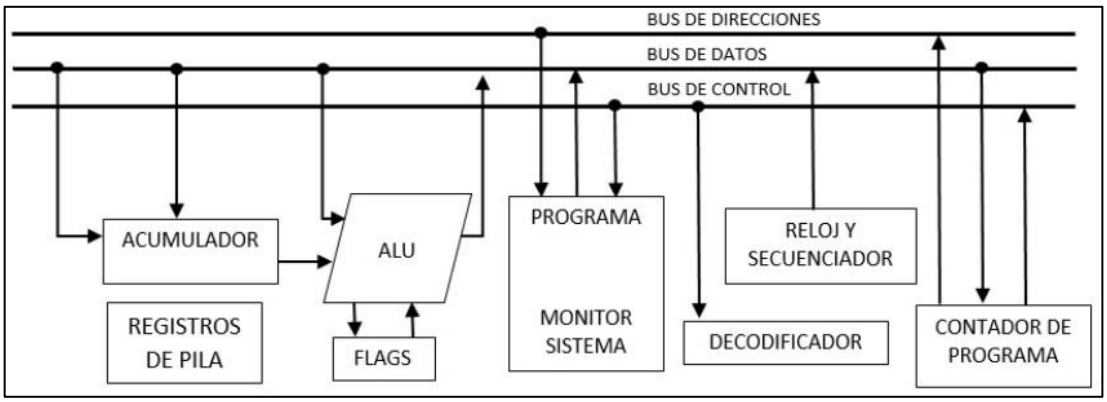
Figura 11. **Comportamiento de memoria tipo LIFO y FIFO**



Fuente: elaboración propia.

Los conceptos anteriores son los que componen únicamente al procesador.

Figura 12. **Componentes del CPU**



Fuente: Autracen. *Descubre la estructura interna de un PLC*. http://www.autracen.com/descubre-la-estructura-interna-plc/. Consulta: 8 de abril de 2019.

Continuando con la estructura interna de un controlador, se encuentran:

2.4.4.2. Memoria

Cualquier tipo de controlador necesita un espacio donde almacenar información y al mismo tiempo pueda retirar esa información; esos espacios o lugares los puede encontrar en las memorias, los controladores utilizan los siguientes tipos para distintas ocupaciones.

- RAM: tipo de memoria volátil que puede ser escrita y leída según las necesidades del programa, es de acceso aleatorio.

- ROM: tipo de memoria no volátil, únicamente para lectura, en ella se guardan instrucciones permanentes para el correcto funcionamiento del controlador y dispositivos adjuntos.

- PROM: memoria programable de solo lectura, igual que la ROM, con la característica que es programable una sola vez, con una capacidad menor que las memorias ROM.

- EEEPROM: memoria ROM con capacidad de ser programada, borrada y programada nuevamente de manera eléctrica.

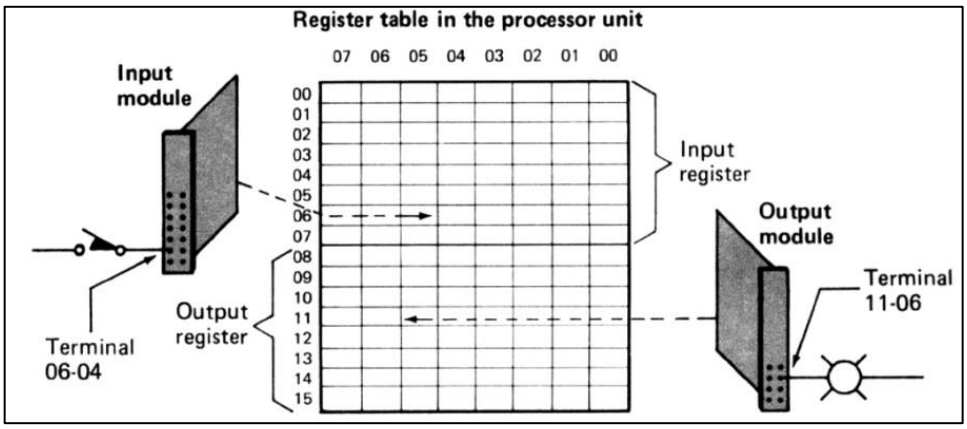
Estas memorias en general se utilizan para:

- Señales de E/S
- Variables internas
- Datos alfanuméricos y constantes
- El programa que se ejecutará
- La propia configuración del controlador

2.4.4.2.1. Memoria de datos

Conocida como tabla de registros, está relacionada con la unidad lógica aritmética, en ésta se escriben y almacenan datos para la ejecución de los programas, de igual forma para mantenerlos después de su ejecución. La memoria de datos también guarda información sobre el estado presente de los dispositivos de E/S; guarda cualquier cambio que ocurra en los dispositivos. Se relaciona con la información que brinda el microprocesador, algo importante es que los registros de la memoria están ligados con las terminales de dispositivos de E/S.

Figura 13. **Relación entre terminales de E/S con registros de memoria**

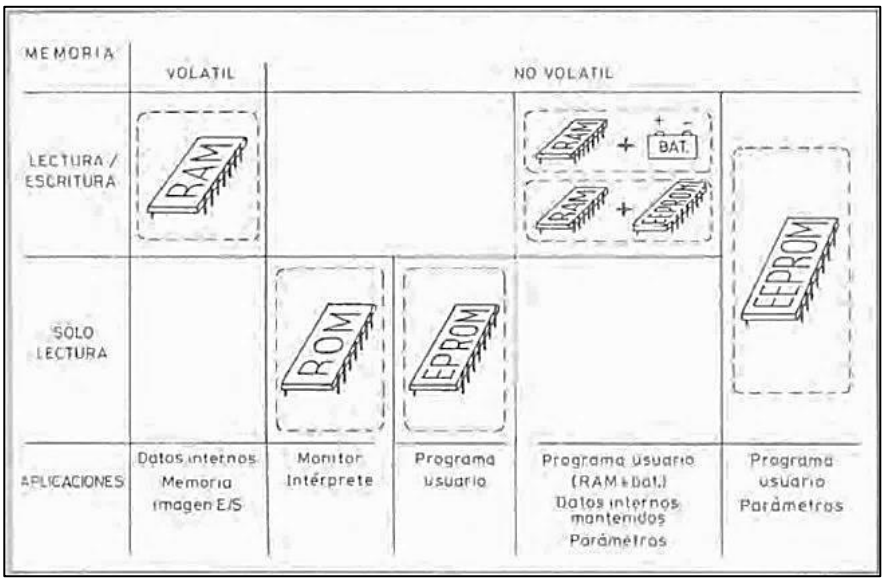


Fuente: *Automación Micromecánica*. http://www.microautomacion.com/capacitacion/Manual061ControladorLgicoProgramablePLC.pdf . Consulta: 8 de abril de 2019.

2.4.4.2.2. Memoria de usuario

Conocida como memoria de programa, es la memoria dedicada a guardar las instrucciones o programas que el usuario creó con base en las necesidades del proceso, dada la necesidad de que estos procesos sean rápidos es que se utilizan las memorias RAM y EEPROM. Tomando en cuenta que si es una RAM, será necesario el uso de baterías para evitar la pérdida del programa.

Figura 14. **Tipos de memoria**



Fuente: BALCELLS, Josep; ROMERAL, José Luis. *Autómatas programables*. p. 71.

2.4.4.3. Interfaz de entrada y salida

Las interfaces son las encargadas de comunicar el controlador con los dispositivos periféricos, de forma que se esté enviando y recibiendo información para que sea analizada en el CPU y que a partir de ese punto se llevan a cabo las acciones indicadas en el programa.

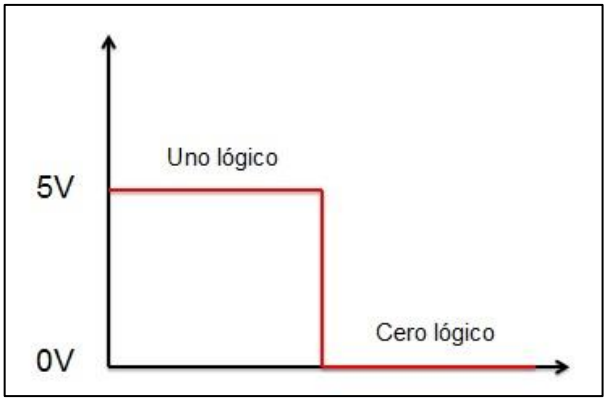
2.4.4.3.1. Dispositivos de entrada

Se encargan de enviar información con respecto a una situación específica de su ambiente, es decir, sobre un fenómeno o condición en particular, como temperatura, flujo, volumen, peso, entre otros. Todos estos tipos de sensores serán los dispositivos de entrada que irán conectados al controlador.

El controlador es capaz de aceptar dos tipos de entradas:

- Digitales: son los valores de voltaje, en este caso del tipo binario, basados en la existencia o no de tensión en la entrada, valores lógicos de uno (valor máximo de voltaje) o cero (ausencia de voltaje), respectivamente. Los controladores tienen la posibilidad de aceptar valores entre 5 V hasta 48 V de DC y en otros casos, valores de 110 V hasta 220 V de AC, dependiendo de la interfaz que maneje el controlador.

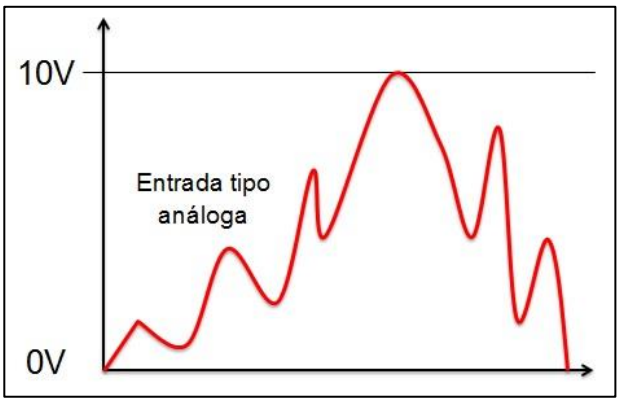
Figura 15. **Ejemplo de valores de entrada digitales**



Fuente: elaboración propia.

- Analógicas: este tipo de entradas permite valores intermedios dentro de un rango establecido, ya sea de voltaje o corriente, los rangos deben ser valores estándar, 0 V a 10 V de DC o 4 mA a 20 mA en el caso de corriente. Los módulos de entradas analógicas poseen un convertidor análogo a digital, A/D, para convertir el valor de entrada en un nuevo valor binario para que le PLC lo interprete y puede realizar las instrucciones correspondientes.

Figura 16. **Ejemplo de entrada de tipo análogo**



Fuente: elaboración propia.

Este tipo de entradas, sin importar si son de tipo continuo o de tipo alterno pasan por ciertas etapas, el tiempo que tarda en pasar por todas las etapas se conoce como tiempo de respuesta a la entrada.

- Rectificación: utilizada para convertir una señal de AC a DC; también, para evitar un cambio de polaridad en la entrada.

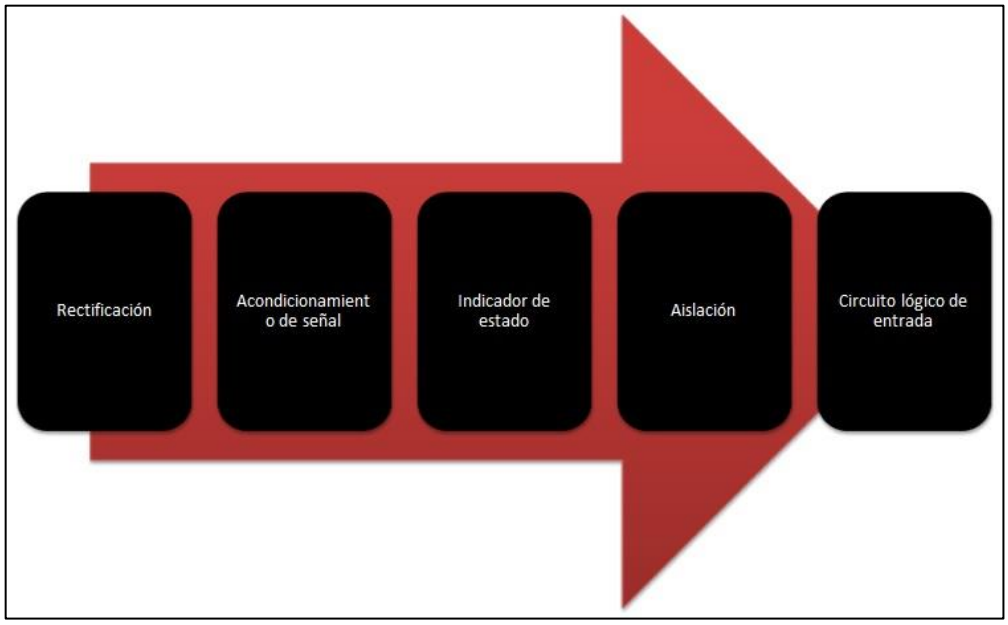
- Acondicionamiento de señal: proceso donde se elimina el ruido eléctrico de las señales, se detectan los niveles de señal y los lleva a valores que puede aceptar la unidad de control.

- Indicador de estado: este indicador es para hacer ver, por medio de un LED en cada entrada, si hay voltaje o no en las entradas, en caso de que exista un valor de tensión se encenderá y de lo contrario se mantendrá apagado.

- Aislación: las entradas de los PLC se aíslan por seguridad, ante algún caso de sobretensión que pueda llegar a la entrada, en caso de que llegara a pasar solo se vería afectada esa entrada y la sobretensión no llegaría al interior del controlador.

- Circuito lógico de entrada: es el encargado de informar a la CPU el estado en que se encuentra la entrada.

Figura 17. **Proceso de la interfaz de entrada**



Fuente: elaboración propia.

Luego de que el valor de tensión llega a la entrada y pasa por cada una de las etapas mencionadas con anterioridad, transcurre un tiempo, este tiempo se conoce como tiempo de respuesta a la entrada; por lo tanto, el valor de tensión debe permanecer por lo menos en la entrada el tiempo de respuesta que posee el controlador que se esté utilizando.

2.4.4.3.2. Dispositivos de salida

Los dispositivos de salida son todos aquellos que responden ante un estímulo enviado por el PLC, con el fin de realizar una acción que mantenga o modifique la realización de uno o varios procesos, algunos dispositivos de salida pueden ser los relés, contactores, electroválvulas, entre otros. Existen cuatro tipos de salidas:

- Salida a relé: es la más utilizada, permite la conexión de dispositivos para DC y AC, con una corriente máxima de 2 A. Este tipo de salida posee un tiempo de respuesta aproximado de 10 ms para conexión y 10 ms para desconexión.

- Salida a transistor: este tipo de salida se puede utilizar únicamente con corriente directa, con cargas de baja potencia, a una corriente máxima de aproximadamente 0,5 A, su durabilidad es más alta que los relés, al igual que su tiempo de respuesta, llegando a 1 ms en conexión y 1 ms en desconexión.

- Salida a triac: la salida por medio de triac está dedicada a cargas de corriente alterna, manteniendo las características de una salida a transistor.

- Salida analógica: internamente el controlador maneja únicamente valores de señales digitales por lo que posee un convertidor digital a análogo y obtener un valor analógico en salida, estos valores dependerán de la resolución, número de bits usados, y del tiempo, tasa de muestreo.

Los valores que llegan a salir del controlador también deben pasar por ciertas etapas al igual que los valores de entrada y el tiempo que le toma a la señal pasar por cada etapa y llegar al punto de salida se conoce como tiempo de respuesta en salida.

- Circuitos lógicos de salida: encargados de recibir la información que envía la CPU.

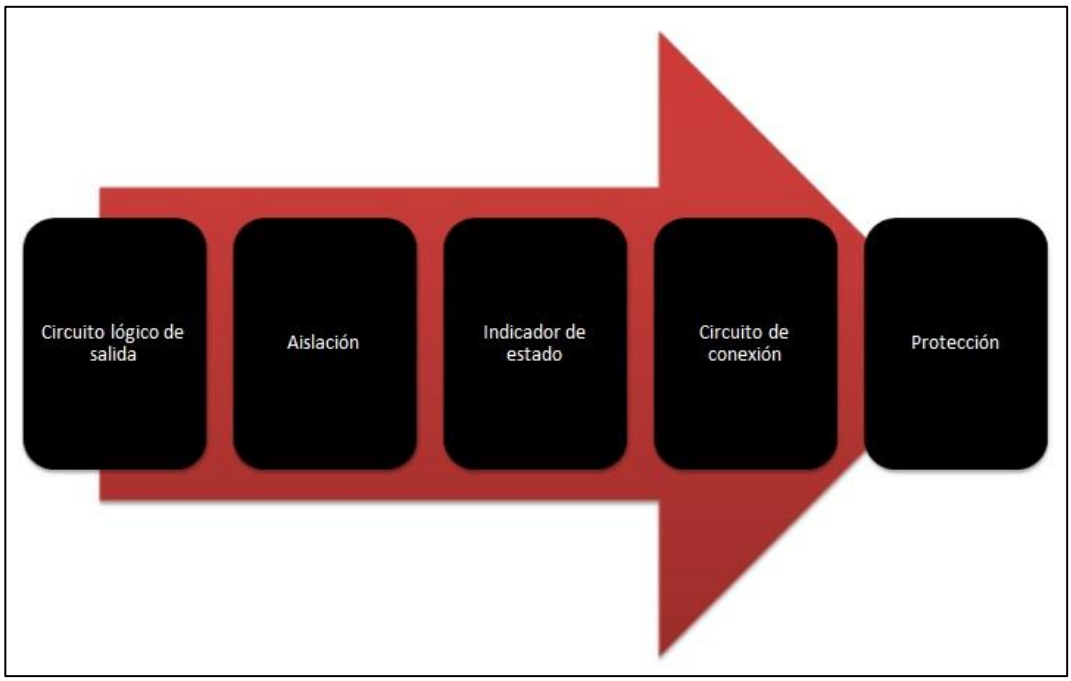
- Aislación: encargado de aislar cada salida por separado para evitar problemas de sobre corriente.

- Indicador de estado: el objetivo de este indicador es hacer ver por medio de un led en cada salida, si hay voltaje o no en éstas; en caso de que exista un valor de tensión se encenderá y, de lo contrario, se mantendrá apagado.

- Circuitos de conexión: se compone del elemento de salida con respecto a la carga que se está utilizando, los tipos de conexión a la salida se mencionaron anteriormente.

- Protección: se encuentra en el interior del PLC, va desde fusibles en los contactos de salida, hasta protecciones electrónicas por sobrecarga.

Figura 18. **Etapas de la interfaz de salida**



Fuente: elaboración propia.

2.4.4.4. Buses de comunicación y campo

Los buses son los encargados de transportar la información, permiten la comunicación entre las diferentes unidades de memoria, la CPU y con las interfaces de entrada y salida; en otros casos se encargan de la comunicación entre varios dispositivos.

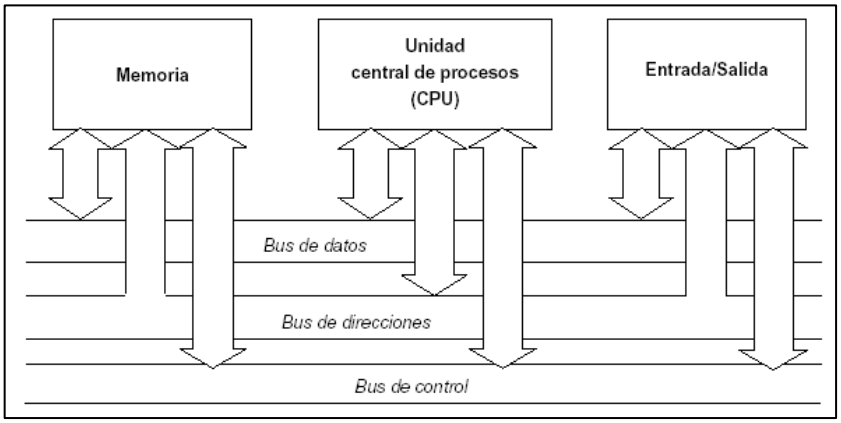
- Bus de comunicación de control: encargado de manejar el tráfico de información para evitar colisiones de información dentro del sistema ya que cada dispositivo dentro del controlador capaz de comunicarse está conectado a los mismos canales de los buses.

- Bus de comunicación de datos: se dedica a llevar los datos del sistema de un dispositivo a otro, su rapidez dependerá del tamaño del bus, el cual se divide en canales y cada uno de éstos representa un bit. Puede tener hasta 32 canales y así ahorrar ciclos de reloj al momento de enviar las instrucciones.

- Bus de comunicación de dirección: dentro del controlador hay algunos dispositivos que puede ser direccionados como la memoria y algunos dispositivos por lo que este bus, independiente del bus de datos, se encarga de llevar la dirección de memoria o periférico del dato que está en tránsito.

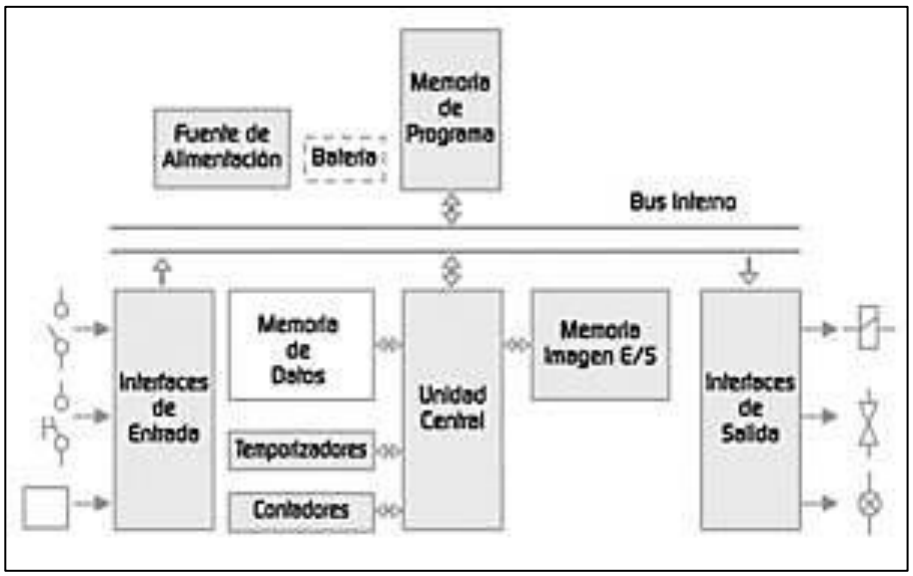
- Bus de campo: se dedica a la comunicación y al intercambio de información entre distintos dispositivos por medio de los protocolos de comunicación que estos puedan aceptar.

Figura 19. **Buses de comunicación**



Fuente: GONZÁLEZ, Jorge. *Buses de comunicación*.
http://80.26.97.21/ele/pro/CURSO%202002-2003/Jorge%20Gonzalez/1.htm.
Consulta: 23 de abril de 2019.

Figura 20. **Estructura interna del PLC**



Fuente: Autracen. *Descubre la estructura interna de un PLC*. http://www.autracen.com/descubre-la-estructura-interna-plc/. Consulta: 23 de abril de 2019.

2.5. Clasificación de los PLC

Existen una variedad de marcas dedicadas al manejo de controladores, pero a pesar de eso todos los controladores cumplen con algunas características, por eso es posible clasificarlos.

2.5.1. Capacidad de entradas y salidas

Una forma de clasificar los PLC es por medio de la cantidad de entradas y salidas que pueda tener el controlador.

- PLC micro: un controlador es clasificado micro cuando posee 64 E/S o menos.

- PLC pequeño: se clasifica como controlador pequeño cuando tiene menos de 256 E/S.

- PLC medianos: es considerado mediano cuando el controlador tiene menos de 1024 E/S.

- PLC grande: el controlador se considera grande a partir de 1024 E/S en su estructura.

2.5.2. Estructura

- PLC nano: controlador que integra su fuente de alimentación, CPU y la capacidad de manejar una cantidad muy pequeña de E/S, menor a 64; incluye entradas y salidas digitales y algunos módulos especiales, sin capacidad de expansión.

- PLC compacto: es así cuando todos sus componentes se encuentran dentro del mismo chasis, tiene más capacidad para manejar E/S y posibles expansiones.

- PLC modular: los controladores modulares son más robustos y brindan la capacidad de poder ser montados sobre una base con todo el equipo, pequeños módulos, que sea necesario para la aplicación, son capaces de expandirse a los requerimientos del proceso, posee instrucciones más complejas, lenguaje de programación más completo y permitiendo diferentes tipos de comunicaciones.

2.6. Funcionamiento de los PLC

En general, sin importar las diferentes marcas de controladores que hay en el mercado, un PLC funciona de manera cíclica y realiza sus operaciones de manera secuencial, una después de la otra, y repitiendo las mismas siempre y cuando el controlador este alimentado. Algunas actividades son realizadas por defecto por el controlador con el fin de hacer pruebas para diagnóstico propio y garantizar su correcto funcionamiento.

2.6.1. Tiempo de barrido

Tiempo que le toma al controlador realizar sus actividades o controles necesarios antes de iniciar con cualquier programa que tenga cargado.

- Autodiagnóstico: en el momento en el que el controlador es conectado y encendido, se trata de una revisión de todos los circuitos que posee y si llegara a tener algún problema, este será reportado e indicado por medio de una señal luminosa; indica el tipo de problema o error encontrado.

- Lectura del registro de entradas: el controlador ve cada entrada y confirma su estado, en alto o en bajo, y con esto genera una imagen de memoria con el estado de cada entrada.

- Lectura y ejecución del programa: la CPU lleva a cabo la ejecución del programa creado por el usuario; usa la imagen de memoria de las entradas y salidas, todo se ejecuta instrucción por instrucción, secuencialmente y en orden, la toma de decisiones se basa en los datos que fueron guardados en memoria.

- Estado de las salidas: luego de ejecutar el programa realizado por el usuario, el controlador hace una actualización del estado de cada salida, creando una imagen de salidas nuevamente.

2.6.2. Ciclo de funcionamiento

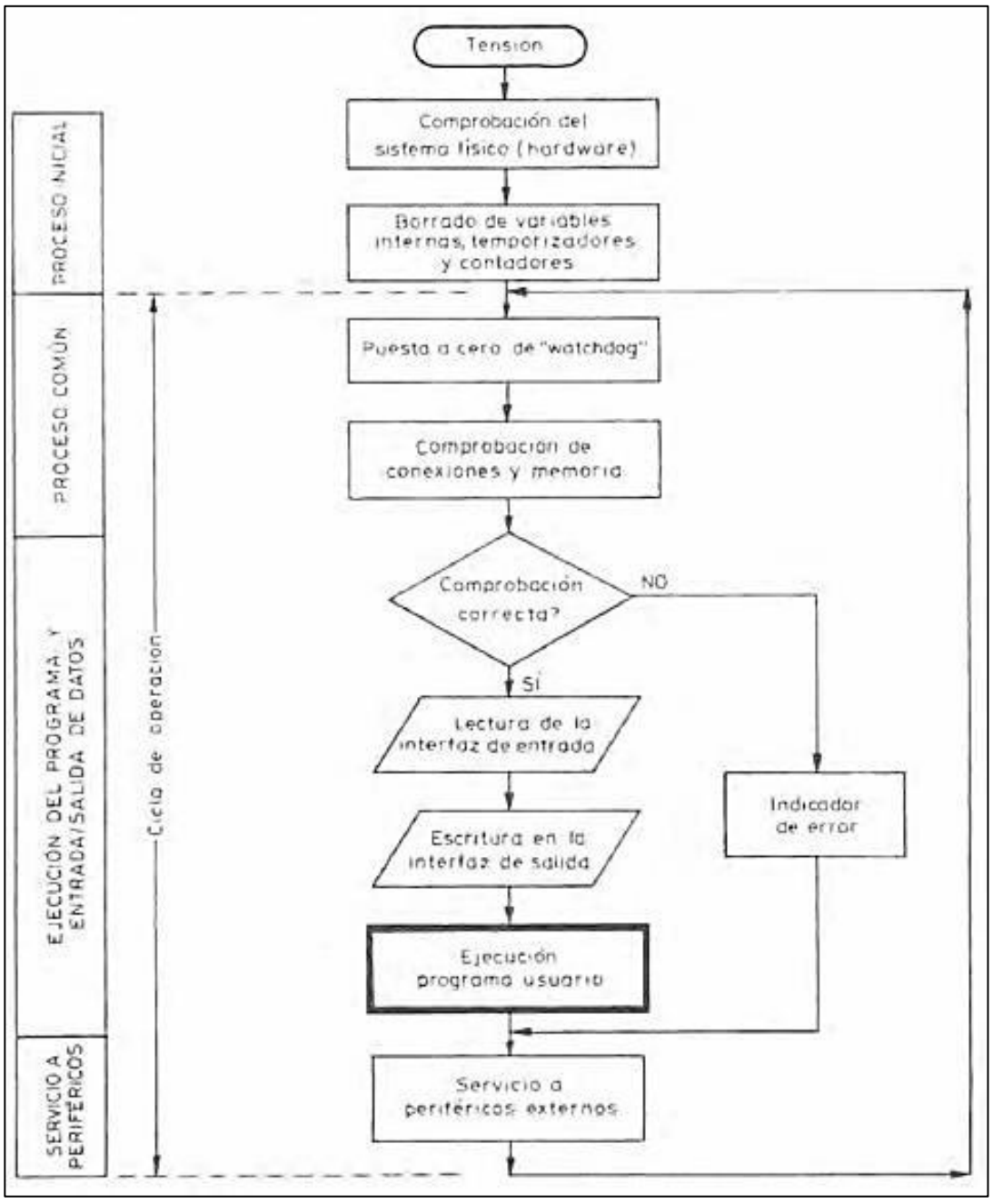
- Funcionamiento en RUN: el controlador realiza el tiempo de barrido sin problemas y es capaz de confirmar las entradas, ejecutar el programa cargado por el usuario y llevar a cabo los procesos establecidos actualizando sus salidas; opera con normalidad es todos los aspectos.

- Funcionamiento en STOP: se detiene la ejecución del programa en el controlador por orden del usuario, todas las salidas pasan a un estado bajo, pero los dispositivos y herramientas internas como relés, registros, contadores, temporizadores, entre otros, mantienen sus estados guardados en memoria interna.

- Funcionamiento en ERROR: en este caso se detiene la ejecución del programa por orden del autómata debido a un error en el funcionamiento y el controlador queda inutilizable hasta que el error sea corregido y todas las salidas pasan a estado bajo.

- *Watchdog*: el reloj de guarda es un temporizador interno del controlador que es inaccesible para el usuario y es el que indica el tiempo máximo de ejecución de un ciclo de operación, entre 0,1 y 0,5 segundos; si ese tiempo es sobrepasado el controlador pasara a STOP y se indicará un error, estos errores pueden ir desde problemas en la sintaxis del programa hasta problemas con la CPU.

Figura 21. **Ciclo de funcionamiento**



Fuente: BALCELLS, Josep; ROMERAL, José Luis. *Autómatas programables*. p. 78.

3. FUNCIONES Y LENGUAJES DE PROGRAMACIÓN DE UN PLC

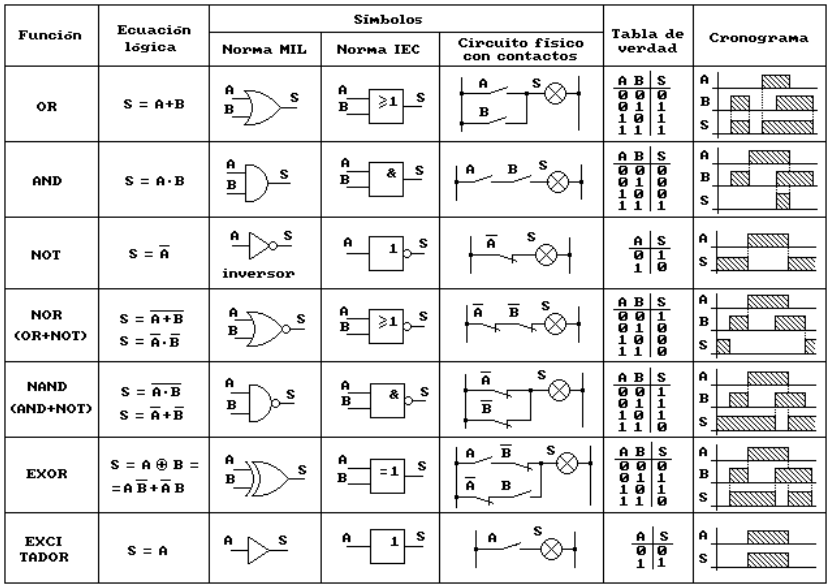
El mercado dedicado a la automatización es amplio, especialmente los distribuidores de PLC, por lo que hay gran variedad de equipos, todos con características diferentes dependiendo del fabricante; pero, en general, todos los controladores cuentan con algunas funciones por defecto.

3.1. Funciones generales

Realizar operaciones combinacionales a partir de los bloques de funciones básicas.

- Función AND: se utiliza para aplicar la condicionante 'y', se basa en la multiplicación.
- Función OR: se aplica la condición 'o', realiza la operación suma.
- Función NOT: encargada de invertir el valor de la variable en su entrada.
- Función NAND: es la versión negada de la función AND.
- Función NOR: es la versión negada de la función OR.
- Función EXOR: función exclusiva, da una respuesta solo cuando las entradas son diferentes.

Figura 22. **Tipos de funciones**



Fuente: Blogspot. *Introducción a la mecatrónica. Controladores lógicos programables*. http://1509720.blogspot.com/2012/11/26-controladores-logicos-programables.html. Consulta: 16 de mayo de 2019.

3.2. Funciones especiales

- Temporizador con retardo a la conexión: encargado de activar una o varias salidas luego de llegar al tiempo que se le programó y lograr la conexión con otros dispositivos.

- Temporizador con retardo a la desconexión: encargado de desactivar una o varias salidas al cumplir con el tiempo estipulado.

- Generador de pulsos: su función es generar pulsos de reloj, con la característica que sus tiempos de estado alto y bajo son iguales.

- Reloj: encargado de activar y desactivar una salida en un día y hora especificada.

3.3. Bloques secuenciales básicos

- Biestables: estos bloques son capaces de mantener un estado siempre que el controlador esté encendido, este es un elemento de memoria ya que su estado no depende de su combinación actual si no de los valores que posee en su entrada. Trasladado a un circuito, es el *latch* de un *flip-flop* tipo *set-reset*.

- Temporizadores: es el dispositivo capaz de retrasar una orden de salida, existen cuatro tipos de funciones.

    - Impulso: la salida permanece activa mientras dura la señal de mando hasta un tiempo máximo, tiempo de impulso.

    - Retardo a la conexión.

    - Retardo a la desconexión.

    - Monoestable: permanece activo durante un tiempo predeterminado a partir de la señal de mando, pero independiente de la misma.

- Contadores: capaz de contar los cambios de nivel en una señal de entrada, activando una salida después de contar hasta un valor indicado con anterioridad.

- Registros de desplazamiento: son funciones compuestas por una serie de biestables, permitiendo el paso de bits de uno a otro ya sea hacia la derecha o izquierda.

3.4. Estructuras de programación

A continuación, se muestra las diferentes estructuras en la programación.

3.4.1. Programación lineal

Esta programación, conocida también como programación monotarea, es la que toma todo el conjunto que involucra el sistema, con todas las variables internas, de entradas y salidas por lo que la programación se hace con instrucciones dentro de una secuencia lineal, una tras otra. En este tipo de programación no es posible acceder desde el programa a subprogramas o módulos con intercambios o no de variables.

Esta programación es de las más usadas para llevar a cabo proyectos que involucran el uso de nano y micro PLC, todo se ejecuta en secuencia pero también por prioridad, se puede trabajar todo por separado y también por bloques de programas, con el fin de ahorrar ciclos de reloj y que pueda ser más rápida la ejecución.

3.4.2. Programación estructurada

La programación estructurada logra simplificar la realización de las tareas que conlleva la ejecución de un proyecto, esto debido a que se tiene el programa separado en partes o subprogramas que pueden ser llamados por un

programa principal. Este tipo de programación es más utilizado para el desarrollo de tareas complejas, ofreciendo las siguientes ventajas:

- La tarea se divide en partes que pueden ser analizadas de forma independiente.

- Permite que la programación sea elaborada por varios programadores.

- En algunos casos, dependiendo del software, se puede realizar el programa en diferentes lenguajes de programación.

- Las subrutinas pueden aprovecharse para otros proyectos.

Este tipo de programación se basa en tres estructuras de control:

- Estructura secuencial: bajo este control, el programa se ejecuta línea por línea.

- Estructura selectiva: se basa en condicionantes y es conocida como la estructura 'si – verdadero – falso', partiendo del uso de funciones como IF, CASE, entre otras.

- Estructura iterativa: basada en el 'Hacer mientras que', es con el fin de ejecutar una acción de forma repetitiva mientras se cumple una condición establecida.

3.4.3. Programación multitarea

Si la tarea es muy compleja y necesita que variables independientes se utilicen entre sí, se aplica la multitarea, donde cada tarea tiene su programa principal ejecutado de forma periódica o no periódica con subprogramas elaborados de forma lineal o estructurada. Cada tarea posee sus propias entradas y salidas al igual que su propio tiempo de operación y funciona de manera concurrente sobre el controlador, se aplican métodos de planificación prioritarios y de gestión de tiempo.

- Planificación prioritaria: con este método se asigna un nivel de prioridad a cada tarea y el controlador se encarga de activar la tarea de mayor prioridad dentro de las solicitadas.

- Planificación de gestión de tiempo: llamada también tiempo compartido, se basa en asignarle ciertas unidades de tiempo a cada tarea y al momento de arrancar el controlador, este irá ejecutando las tareas según una plantilla de tiempos predeterminada y si hubiese alguna colisión, la resolverá con el nivel de prioridad.

3.5. Lenguajes de programación

Actualmente, existen varios tipos de programación para controladores, el uso de los lenguajes es decisión del usuario o programador idealmente, con base en la tarea o proyecto que se quiera desarrollar. Existe la posibilidad de mezclar el uso de los lenguajes de programación dentro de un controlador, pero esto se puede hacer solo si el equipo lo permite.

3.5.1. Estándar IEC 61131-3

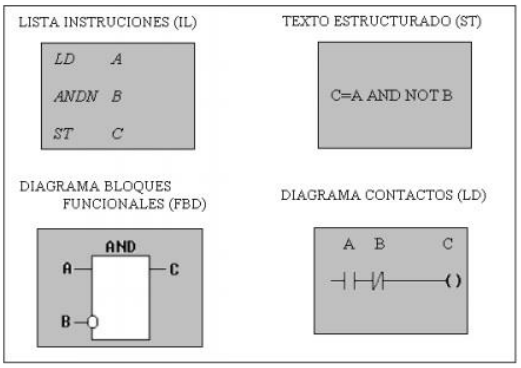
La norma 61131-3 de la comisión electrotécnica internacional define cuatro tipos de lenguajes de programación, por lo que su sintaxis se encuentra definida y no puede haber cambios o características particulares dentro de los mismos, aunque sean aplicados a equipos de diferentes marcas.

Los cuatro tipos de programación son:

- Diagrama de escalera o contactos: *ladder diagram*, LD
- Diagrama de bloques funcionales: *function block diagram*, FBD
- Lista de instrucciones: *instruction list*, IL
- Texto estructurado: *structured text*, ST

Todos los lenguajes se relacionan entre sí y es posible su uso de forma conjunta para resolver tareas complejas.

Figura 23. **Ejemplo de una instrucción en los 4 lenguajes**



Fuente: Uniovi. *Estandarización en la programación del control industrial*. http://isa.uniovi.es/docencia/IngdeAutom/transparencias/iec1131-3%20espa%F1ol.pdf. Consulta: 20 de mayo de 2019.

3.5.2. Diagrama de contactos

Este tipo de lenguaje se basa en la representación gráfica de la lógica de relés, se conoce también como lenguaje de escalera y es un tipo de lenguaje visual. El lenguaje es utilizado en la mayoría de los casos para señales de tipo booleano y no para señales análogas.

Es un lenguaje totalmente gráfico y posee dos barras verticales en los extremos derecho e izquierdo que representan la alimentación del circuito, todo se reduce al uso de contactos, bobinas y algunos bloques de funciones, todo se va realizando de arriba hacia abajo, ejecuta los circuitos de izquierda a derecha, ver figura 25.

Al ser un lenguaje normalizado, la simbología que se usa también lo está, los símbolos más básicos son los siguientes y se muestran en la figura 24.

- Contacto NA: normalmente abierto, forma parte de la entrada del circuito y se activa cuando se detecta un uno lógico en el elemento que representa.

- Contacto NC: normalmente cerrado, pertenece a las entradas de los circuitos pero se activa al detectar un cero lógico.

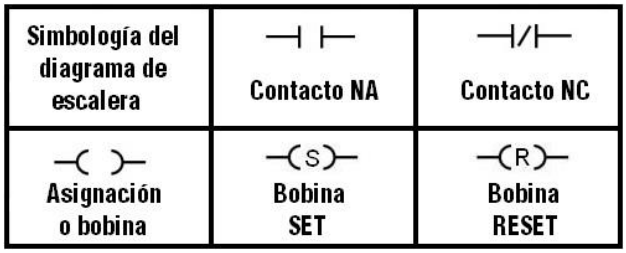
- Bobina: representa la salida de los circuitos, se activa al recibir un uno lógico de la combinación de contactos que estén en la entrada del circuito.

- Bobina SET: representa un elemento de salida, igual que la bobina, tiene la característica de que al ser activada con un uno lógico, no se puede

desactivar solo con un cero lógico, es necesario que la desactivación sea por medio de la bobina RESET.

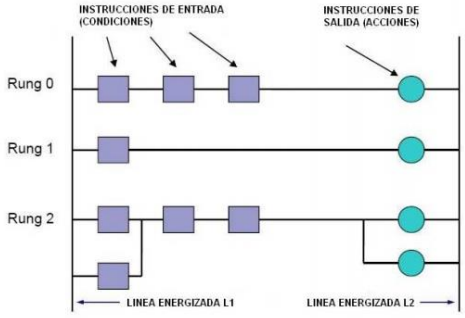
- Bobina RESET: complemente de la bobina SET, se encarga de desactivar su bobina SET previamente activada.

Figura 24. **Simbología básica**



Fuente: Ikastaroak. *Representación de esquemas de automatización*. https://ikastaroak.ulhi.net/edu/es/DFM/RGFM/RGFM03/es_DFM_RGFM03_Contenidos/website_141_lenguaje_de_programacin.html. Consulta: 21 de mayo de 2019.

Figura 25. **Partes de la programación en escalera**



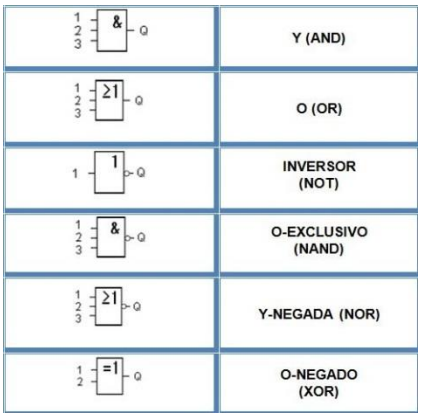
Fuente: Rocatek. *Programación ladder PLC básica*. http://www.rocatek.com/downloads/Programacion%20Ladder.pdf. p.1. Consulta: 21 de mayo de 2019.

3.5.3. Diagrama de bloques funcionales

Es un lenguaje gráfico en el que los bloques funcionales son como circuitos integrados, es un lenguaje muy utilizado en Europa; este lenguaje se asemeja al diseño de un circuito, todo es por medio de bloques interconectados para poder llevar a cabo una tarea.

Es un lenguaje de alto nivel ya que brinda la oportunidad de reducir funciones en bloques y que el usuario solo se preocupe por la funcionalidad de la rutina; se basa en funciones de lógica booleana y en funciones especiales, como se muestran en las figuras 26 y 27, respectivamente.

Figura 26. **Funciones básicas**



Fuente: Index. *Tecnología técnica*. http://www.tecnologia-tecnica.com.ar/sistemacontrollogo/index%20controllogo_archivos/Page598.htm. Consulta: 21 de mayo de 2019.

Tabla II. **Verdad para la función AND**

Entrada 1	Entrada 2	Salida Q
0	0	0
0	1	0
1	0	0
1	1	1

Fuente: elaboración propia.

Tabla III. **Verdad para la función OR**

Entrada 1	Entrada 2	Salida Q
0	0	0
0	1	1
1	0	1
1	1	1

Fuente: elaboración propia.

Tabla IV. **Verdad para la función NOT**

Entrada	Salida
0	1
1	0

Fuente: elaboración propia.

Tabla V. **Verdad para la función NAND**

Entrada 1	Entrada 2	Salida Q
0	0	1
0	1	1
1	0	1
1	1	0

Fuente: elaboración propia.

Tabla VI. **Verdad para la función NOR**

Entrada 1	Entrada 2	Salida Q
0	0	1
0	1	0
1	0	0
1	1	0

Fuente: elaboración propia.

Tabla VII. **Verdad para la función XOR**

Entrada 1	Entrada 2	Salida Q
0	0	0
0	1	1
1	0	1
1	1	0

Fuente: elaboración propia.

Figura 27. **Funciones especiales**

	Retardo de activación
	Retardo de desactivación
	Relé de impulsos
	Reloj de temporización
	Relé de parada automática
	Emisor de cadencias
	Retardo de activación memorizable
	Contador de horas de servicio
	Relé disipador
	Contador adelante/atrás
	Discriminador
	Generador de impulsos asíncrono
	Temporizador anual

Fuente: Index. *Tecnología técnica*. http://www.tecnologia-tecnica.com.ar/sistemacontrollogo/index%20controllogo_archivos/Page598.htm. Consulta: 21 de mayo de 2019.

3.5.4. Lista de instrucciones

Es un lenguaje de programación de tipo literal, muy utilizado en Europa, es de bajo nivel, un modelo de lenguaje ensamblador, su mejor aplicación es en tareas poco complejas o de corta extensión.

El control del programa se logra por medio de los saltos y de llamar las subrutinas existentes dentro del mismo programa; su ejecución es línea por línea; tiene la característica de que siempre inicia con el operador LD y se finaliza con el operador ST porque corresponden a las entradas y salidas respectivamente. El lenguaje de manera general contiene:

- Etiquetas: es el nombre que el usuario o programador le pondrá a las rutinas dentro del programa general.

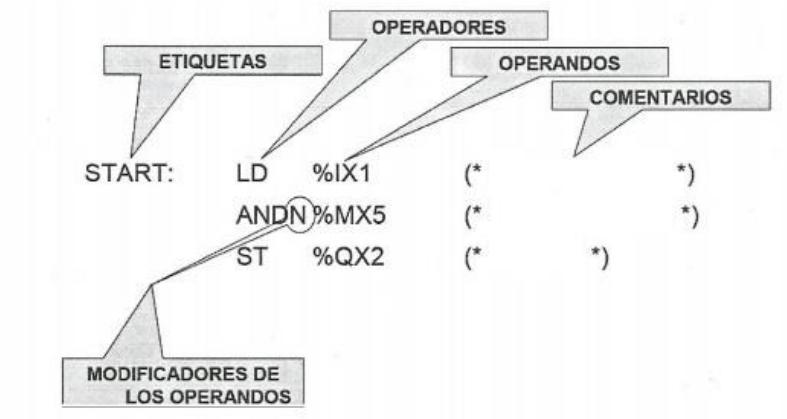
- Operadores: es la acción que se realizará.

- Operandos: es el elemento al que se le aplicará la acción del operador.

- Modificadores de los operandos: capaces de cambiar el funcionamiento de los operadores, por ejemplo, convertir una función AND en una función NAND

- Comentarios: propios de cada usuario, con el fin de recordar el funcionamiento del programa y/o referencias a otras partes del programa.

Figura 28. **Estructura del lenguaje IL**



Fuente: Programación de PLC´s. *Lenguaje en lista de instrucciones*. http://www.myelectronic.mipropia.com/Aplicaciones/curso%20de%20plc/Lenguaje%20en%20lista%20de%20instrucciones.pdf?i=1. p.3. Consulta: 21 de mayo de 2019.

Tabla VIII. **Operadores básicos**

Operador	Uso
LD	Carga el valor actual del operando
ST	Almacena el resultado en el operando
S	Cambia el valor del operando a 1
R	Cambia el valor del operando a 0
JMP	Realiza el salto a la etiqueta indicada
CAL	Llama un bloque de funciones específico.
RET	Retorna el valor de la función o bloque de función
ADD	Operación de suma
SUB	Operación de resta
MUL	Operación de multiplicación
DIV	Operación de división
&, AND	Función Y
OR	Función O
XOR	Función O exclusiva
GT	Uso del mayor que (>)

Continuación de la tabla VIII.

GE	Uso del mayor igual (>=)
EQ	Uso del igual a (=)
NE	Uso del mayor que y menor que (< >)
LE	Uso del menor igual (<=)
LT	Uso del menor que (<)

Fuente: elaboración propia.

Figura 29. **Ejemplo de función AND en lenguaje IL**

Fuente: Unicen. *Descripción de lenguajes*. http://www.exa.unicen.edu.ar/catedras/tldcaut/Software%20Lenguajes%20LD%20y%20IL.pdf. p.20. Consulta: 21 de mayo de 2019.

3.5.5. Texto estructurado

El lenguaje de texto estructurado fue de los últimos en aparecer para el desarrollo de programas dentro de los controladores, es un lenguaje de alto nivel, basado en lenguajes de programación conocidos como Pascal, Basic y C.

Es similar al lenguaje de lista de instrucciones ya que ambos son de tipo literal, pero la ventaja del texto estructurado es que permite resolver tareas de más complejidad al igual que brinda una construcción más clara; permite el uso de ciclos como IF, WHILE, FOR, CASE, entre otros; incluso es posible la realización de operaciones matemáticas de forma más simple.

La estructura y sintaxis del lenguaje está basado en el lenguaje C, algunos de los operadores y un ejemplo se muestran en la tabla IX y la figura 30, respectivamente.

Tabla IX. **Operadores básicos del lenguaje ST**

Operador	Función
:=	Asignar valor a una variable (A := 4)
+, -, *, /	Operaciones matemáticas
MOD	Obtener el módulo de una división
<, >, <=, >=, <>, =	Funciones de comparación e igualdad
AND, OR, NOT, XOR	Funciones booleanas
IF, FOR, WHILE, CASE	Ciclos

Fuente: elaboración propia.

Figura 30. **Ejemplo de lenguaje ST**

```
Q1 := (IN1 AND IN2) OR (NOT IN3);
IF Q1 == 1 THEN Q2 == 1;
     ELSE Q2 == 0;
END_IF;
```

Fuente: elaboración propia.

4. SOFTWARE DE PROGRAMACIÓN

En la actualidad y dentro del mercado laboral existe una gran variedad de software para la programación de controladores lógicos, a pesar de que los lenguajes están estandarizados, cada proveedor le agrega características extras y/o herramientas especiales dedicadas a la marca del controlador.

Todos los software de programación existentes son excelentes y brindan una gran capacidad de desarrollo, los de mayor capacidad permiten el desarrollo de escenarios y su respectiva aplicación en un entorno 3D y, al mismo tiempo, poder observar su comportamiento como si fuera un escenario totalmente real.

Dado que los paquetes de software con estas características requieren del pago de una licencia para su utilización, además que sus versiones de demostración tienen límites de uso o vencimiento después de algunos días, se complica la adquisición para el usuario y a su vez imposibilita su uso para fines didácticos.

Considerando lo anterior y con el fin de que las personas interesadas en aprender sobre los controladores y su programación, se utilizará un software de descarga gratuita con la posibilidad de programar por lo menos en dos de los lenguajes estandarizados para los controladores.

4.1. Zelio Soft

Es un software distribuido por el grupo Shneider Electric, empresa con más de 180 años dentro del mercado, ofrece distintas versiones del software, siendo la más reciente y la que se usará, Zelio Soft 2 en su versión 5.2, publicada en la página principal de la empresa el 20 de diciembre del 2018.

El software permite el lenguaje de programación en escalera y lenguaje de bloques de funciones, permite 240 líneas de programación en lenguaje escalera, y ofrece compatibilidad con el sistema operativo de Windows 10 para 32 y 64 bits.

4.1.1. Descarga e instalación del software

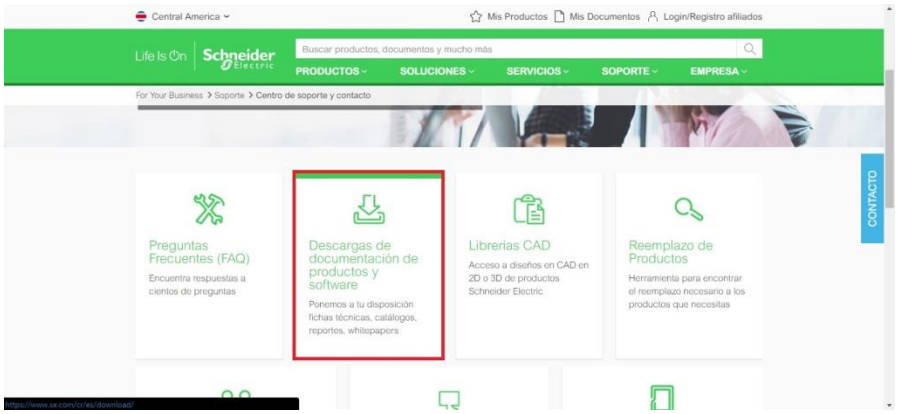
El grupo Shneider Electric posee un sitio oficial de internet, https://www.se.com/gt/es/, para Guatemala, al ingresar al sitio se debe seleccionar la opción de soporte, ver figura 31; se inicia así el proceso de descarga.

Figura 31. **Opción de soporte**

Fuente: *Shneider Electric*. https://www.se.com/gt/es/. Consulta: 28 de mayo de 2019.

Luego de seleccionar la opción de soporte, se debe buscar la opción descargas de documentación de productos y software, figura 32.

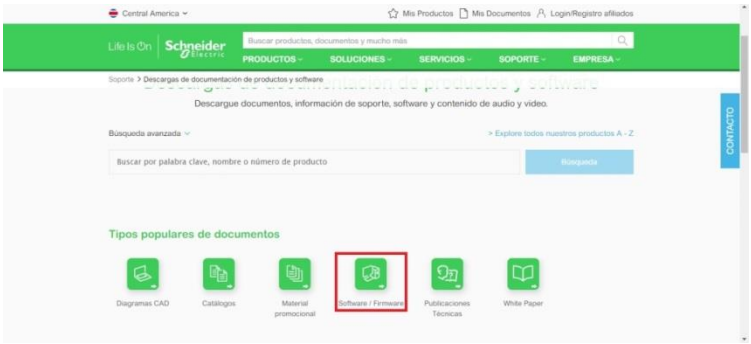
Figura 32. **Opción de descargas de documentación de productos y software**



Fuente: *Shneider Electric*. https://www.se.com/cr/es/work/support/support-center.jsp. Consulta: 28 de mayo de 2019.

El paso a seguir es seleccionar la opción de software/firmware, figura 33.

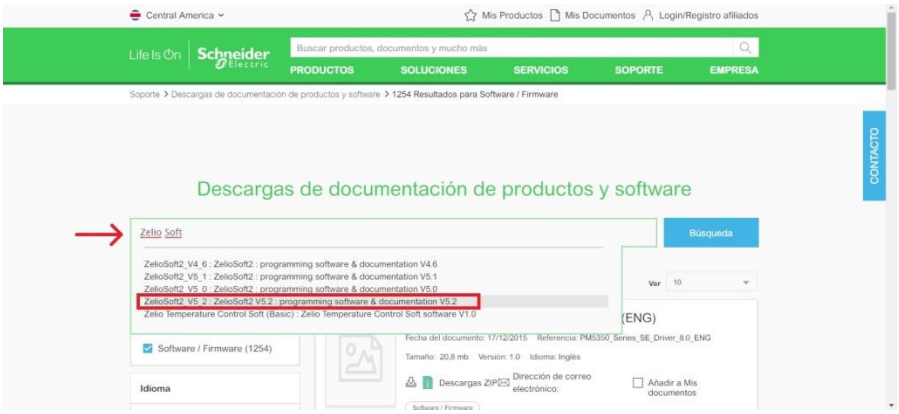
Figura 33. **Opción de software/firmware**



Fuente: *Shneider Electric*. https://www.se.com/cr/es/download/. Consulta: 28 de mayo de 2019.

Al seleccionar esta opción, aparecerá una barra de buscador y en esta se deberá escribir Zelio Soft; a continuación, se desplegará una serie de opciones y se seleccionará la opción de ZelioSoft2 V5.2, ver figura 34.

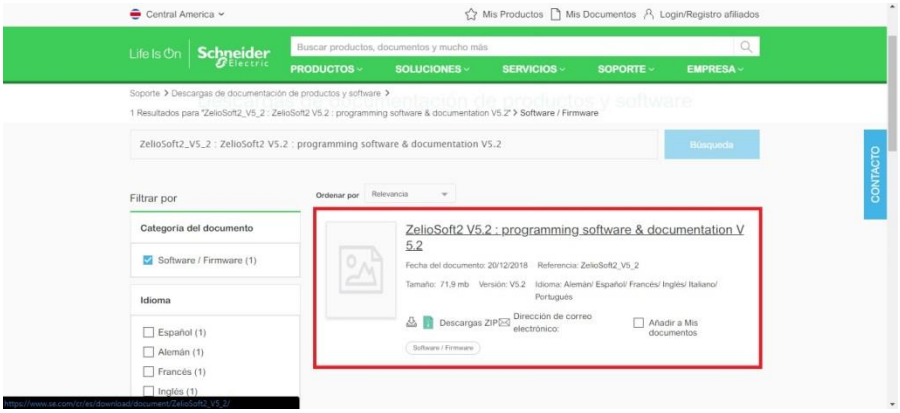
Figura 34. **Selección del software**



Fuente: *Shneider Electric*. https://www.se.com/cr/es/download/doc-group-type/8428037-Software%20/%20Firmware/?sortByField=Popularity. Consulta: 28 de mayo de 2019.

Al seleccionar esta opción, la nueva opción que aparecerá en el sitio será la del software en su versión 5.2, como se observa en la figura 35.

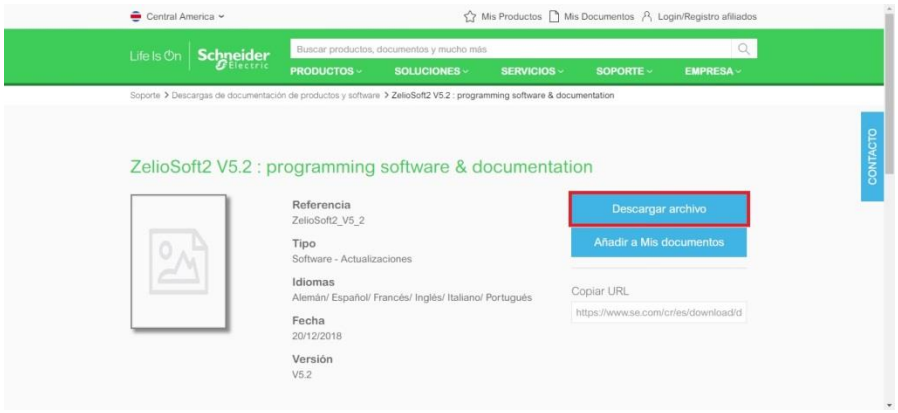
Figura 35. **ZelioSoft2 V5.2**



Fuente: *Shneider Electric*. https://www.se.com/cr/es/download/doc-group-type/8428037-Software%20/%20Firmware/?keyword=ZelioSoft2_V5_2+%3A+ZelioSoft2+V5.2+%3A+programming+software+%26+documentation+V5.2. Consulta: 28 de mayo de 2019.

Por último, se debe seleccionar la opción de descargar archivo, el archivo tiene un tamaño de 71,9 MB.

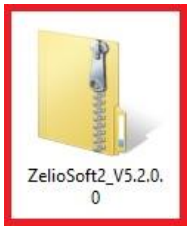
Figura 36. **Descarga del software**



Fuente: *Shneider Electric*. https://www.se.com/cr/es/download/document/ZelioSoft2_V5_2/. Consulta: 28 de mayo de 2019.

Descargado el archivo, en formato ZIP, aparecerá con el nombre de ZelioSoft2_V5.2.0.0, partiendo de este punto, se inicia el proceso de instalación.

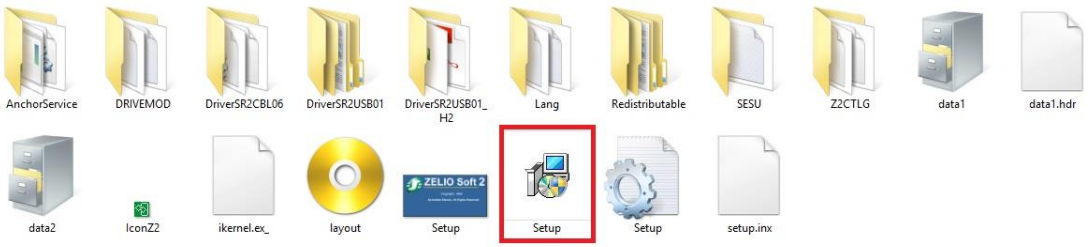
Figura 37. **Archivo con el software de programación**



Fuente: elaboración propia.

Teniendo el archivo almacenado en la carpeta de su elección, deberá extraer los archivos en la misma carpeta o en otra de su elección; se añadirán nuevos archivos dentro de la carpeta donde se realizó la extracción; dentro de los elementos de la carpeta deberá ejecutar el archivo con el nombre Setup para instalar el software de programación en la computadora, ver figura 38.

Figura 38. **Extracción de los archivos**



Fuente: elaboración propia.

Ejecutando el instalador, se genera la primera ventana, figura 39, donde se instalarán los componentes necesarios para el software; al terminar este proceso aparecerá una segunda ventana donde se seleccionará la opción finalizar, figura 40.

Figura 39. **Primera ventana de instalación**

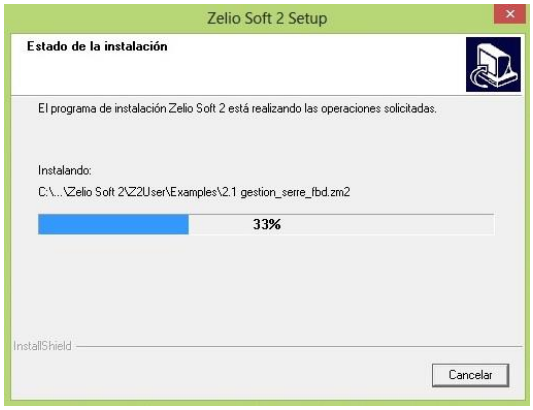
Fuente: elaboración propia.

Figura 40. **Segunda ventana de instalación**

Fuente: elaboración propia.

Una tercera ventana, figura 41, se mostrará con una barra de progreso, relacionada al estado de la instalación de todo el software de programación; cuando esta llegue al cien por ciento habrá terminado toda la instalación y la ventana se cerrará automáticamente.

Figura 41. **Tercera ventana, progreso de instalación**



Fuente: elaboración propia.

4.1.2. Uso del software Zelio Soft 2

Instalado el software, se creará un acceso directo en el escritorio de la computadora con su respectivo ícono; se deberá ejecutar el acceso directo para iniciar el software; lo primero que aparecerá es una ventana, figura 42, con diferentes opciones, que dependiendo de la necesidad, se tomará una.

- Crear un nuevo programa: seleccionando esta opción se iniciará un nuevo programa desde cero, en cualquiera de los lenguajes que ofrece el software.

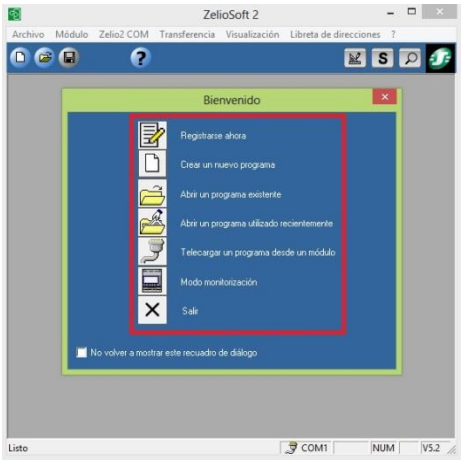
- Abrir un programa existente: si hubiese uno o más programas guardados, con esta opción será posible su uso, buscando el archivo que se necesita ejecutar dentro de las carpetas que pueda tener el equipo.

- Abrir un programa utilizado recientemente: el usuario, sin necesidad de buscar programas en carpetas o memorias, podrá abrir desde el software los programas vistos y usados por última vez.

- Telecargar un programa desde un módulo: si se tiene el acceso a módulos reales, tener de forma física el equipo, con esta opción se podrá transferir la información por medio de cable al controlador y así poder cargarle el programa y ejecutarlo de una manera real.

- Modo monitorización: esta opción permite ejecutar el programa directamente en el controlador y a su vez en el simulador, logrando observar de mejor manera el comportamiento del equipo.

Figura 42. **Opciones de trabajo en Zelio Soft 2**



Fuente: elaboración propia.

Se Iniciará con la opción para crear un nuevo programa; a continuación, saldrá la ventana de selección de módulo, figura 43, en esta se deberá elegir el que tipo de módulo se utilizará, contando con las siguientes opciones.

- Opción uno, esquina superior izquierda, se puede elegir:
    - 6 entradas digitales y 4 salidas.
    - 4 entradas digitales más 4 entradas mixtas (digital/análogo) y 4 salidas.
- Opción dos, esquina inferior izquierda, se puede elegir:
    - 10 entradas digitales más 2 entradas mixtas y 8 salidas
    - 6 entradas digitales más 6 entradas mixtas y 8 salidas
    - 12 entradas digitales y 8 salidas
- Opción 3, centrada hacia arriba, ofrece:
    - 4 entradas digitales más 4 entradas mixtas y 4 salidas
    - 6 entradas digitales y 4 salidas
    - 8 entradas digitales y 4 salidas
- Opción 4, centrada hacia abajo, ofrece:
    - 2 entradas digitales más 4 entradas mixtas y 4 salidas
    - 6 entradas digitales y 4 salidas

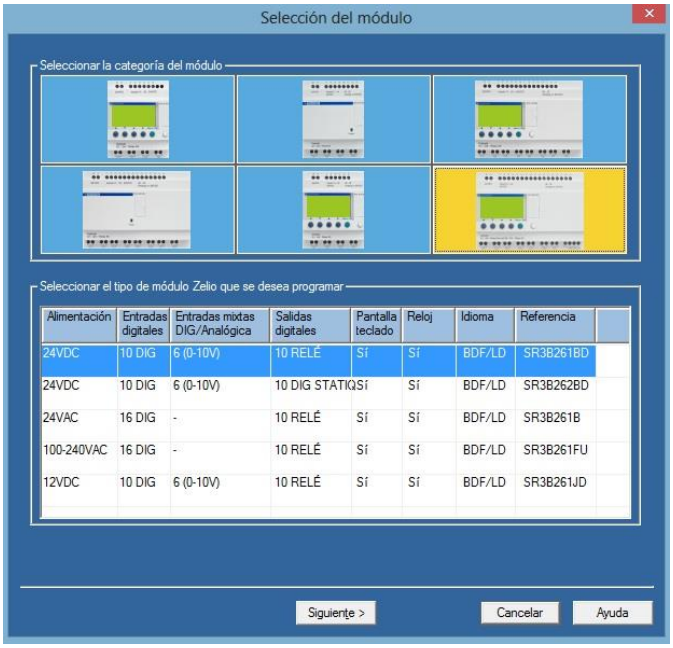
- Opción 5, esquina superior derecha, tiene capacidad para

    - 6 entradas digitales más 6 entradas mixtas y 8 salidas
    - 10 entradas digitales más 2 entradas mixtas y 8 salidas
    - 12 entradas digitales y 8 salidas

- Opción 6, esquina inferior derecha, puede operar con:

    - 10 entradas digitales más 6 entradas mixtas y 10 salidas
    - 16 entradas digitales y 10 salidas

Figura 43. **Selección de módulo en Zelio Soft 2**

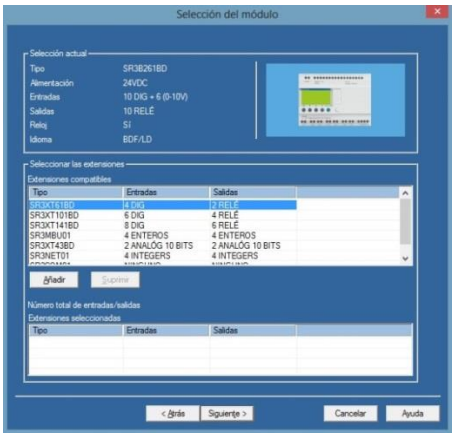


Fuente: elaboración propia.

Todas las opciones son módulos o controladores de gama baja; luego de seleccionar el controlador que se utilizará, se deberán de elegir también las condiciones bajo las que operará el controlador, como se muestra en la figura 43; la serie de letras y números que se muestran en el cuadro de referencia es para casos en donde se tiene acceso al módulo de forma física, ya que ese módulo deberá tener la misma serie de letras y números.

En algunos casos, la tarea que se debe resolver es extensa y el módulo por si solo no es capaz de soportar su ejecución, por falta de entradas o salidas o algún elemento específico que no posea; en esos casos se necesita agregar una extensión al módulo principal, ver figura 44; de lo contrario, no será necesario agregar extensiones.

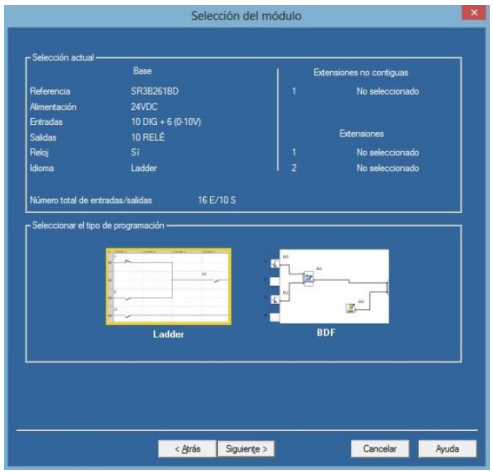
Figura 44. **Añadir extensiones al módulo**



Fuente: elaboración propia.

Por último, antes de empezar la programación, se debe determinar bajo que lenguaje se programará el módulo, puede ser con lenguaje de escalera o lenguaje de bloques de funciones, como se muestra en la figura 45.

Figura 45. **Selección del lenguaje de programación**



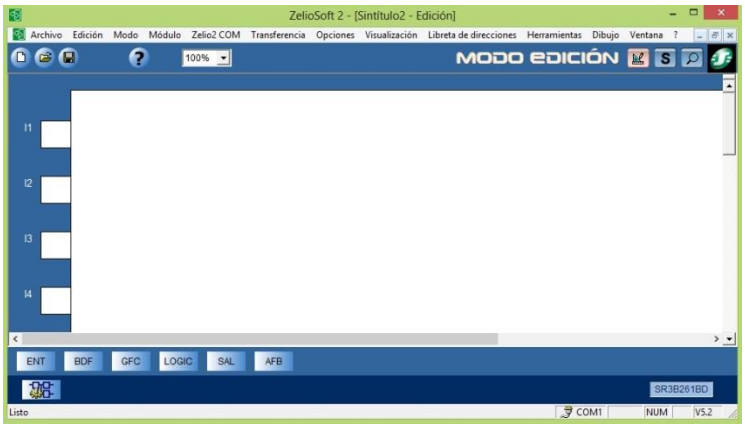
Fuente: elaboración propia.

Al determinar el lenguaje de programación que se utilizará, se abrirá finalmente el entorno de programación, como se muestra en la figura 46 y 47 para lenguaje de escalera y de bloques de funciones respectivamente.

Figura 46. **Entorno de programación en lenguaje escalera**

Fuente: elaboración propia.

Figura 47. **Entorno de programación en lenguaje de bloques de funciones**



Fuente: elaboración propia.

Ambos lenguajes se programan en el modo de edición, donde cada contacto o bloque de función debe ser seleccionado y arrastrado al lugar donde se desea colocar; el software no permite la colocación de elementos en lugares incorrectos, por ejemplo, colocar un elemento de salida en la posición de elementos de entrada.

Toda prueba que se quiera realizar deberá ser en el modo de simulación para que el software ejecute el programa realizado; si existiera algún problema, este será notificado por medio de un aviso y se detendrá la simulación; en caso de querer realizar alguna modificación al programa, esta deberá ser realiza en el modo de edición; de lo contrario, no se podrá realizar ningún cambio.

4.1.3. Programación con lenguaje escalera

El entorno de programación para este lenguaje es el que se observó en la figura 46; sin embargo, hay detalles que se deben tomar en cuenta antes de iniciar a programar en este lenguaje.

- Por defecto el programa muestra los elementos por medio de símbolos eléctricos, si se desea se pueden cambiar a símbolos *ladder*, en la pestaña de visualización.

- Es posible cambiar los colores que representan el estado activo o no activo de cada elemento, para esto se debe buscar la pestaña de opciones y seleccionar el apartado de modificar colores.

- El área de programación se encuentra dividida en 240 peldaños de programación y cada uno tiene capacidad para 5 contactos y su respectiva bobina de salida.

- Las uniones o conexiones entre cada uno de los contactos se realiza por medio de cada línea punteada, únicamente haciendo clic con el ratón sobre la línea para establecer la conexión.

- Es importante saber que posee herramientas para:

    - Revisar la coherencia del programa y detectar posibles errores, ver figura 48.

    - Trabajar en modo edición para realizar la programación y la opción de hacer pruebas en modo simulación; para llevar a cabo la

simulación se debe selección la opción *run* para dar inicio a la misma, ver figura 49 y 50.

- El modo simulación posee cuatro elementos, entradas, pulsadores, salidas y tiempo de simulación, se muestran en la figura 51.

Figura 48. **Herramienta para revisar la coherencia del programa**

Fuente: elaboración propia.

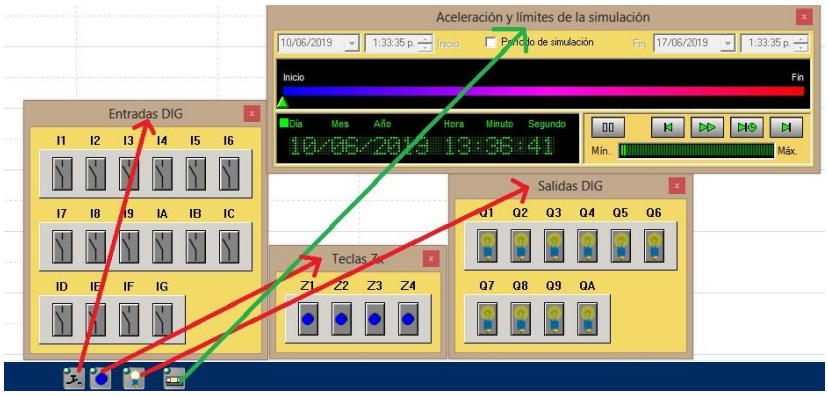
Figura 49. **Modo de edición**

Fuente: elaboración propia.

Figura 50. **Modo de simulación**

Fuente: elaboración propia.

Figura 51. **Elementos del modo simulación**



Fuente: elaboración propia.

El entorno de programación en lenguaje escalera posee 13 elementos, estos se pueden observar en la figura 52 y serán explicados de izquierda a derecha en el mismo orden como se ven en la figura mencionada.

Figura 52. **Elementos de programación en lenguaje escalera**

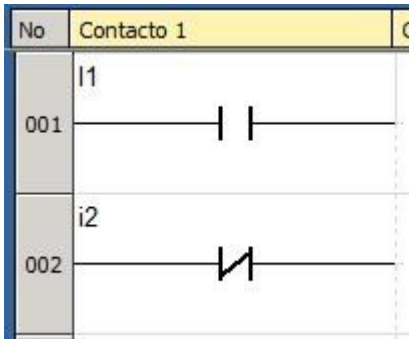
Fuente: elaboración propia.

4.1.3.1. Entradas digitales

Las entradas se utilizarán dentro de la programación únicamente como contactos, estos contactos van a representar los posibles estados de las entradas reales; cada contacto cuenta con identificador, siendo I o i, para un contacto normalmente abierto o normalmente cerrado respectivamente; al igual que cuenta con un número de entrada, así como se muestra en la figura 53.

Los contactos por defecto son normalmente abiertos; si se desea cambiar el estado del contacto, se puede hacer seleccionando el contacto presionando la barra espaciadora, o dando clic derecho y cambiando el estado del contacto.

Figura 53. **Contactos de entradas digitales**



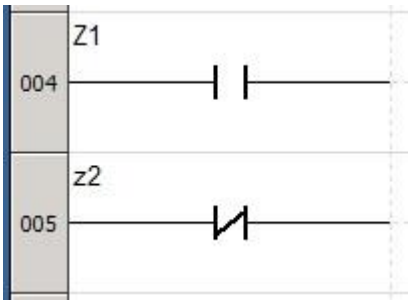
Fuente: elaboración propia.

4.1.3.2. Teclas Zx

Estos elementos se conocen como teclas de navegación, estos elementos pertenecen a las teclas o botones físicos que pueda tener el módulo que se esté utilizando; entonces, son usados exclusivamente como pulsadores.

Al igual que las entradas digitales, son contactos dentro del programa, y poseen sus propios identificadores, con su respectivo número de contacto, por defecto, el contacto estará normalmente abierto (Z) y si se desea cambiar a normalmente cerrado (z), se realiza el mismo procedimiento que con las entradas digitales, en la figura 54 se muestra el elemento.

Figura 54. **Contactos de teclas Zx**



Fuente: elaboración propia.

4.1.3.3. Relés auxiliares

Los relés auxiliares son contactos que se comportan como salidas digitales, la característica es que no tienen un contacto de salida eléctrica, estos son utilizados como variables internas.

Estos elementos se pueden usar entonces como contactos o como bobinas, brindan la capacidad de almacenar estados con contactos asociados.

Estos elementos tienen cuatro posibles usos como bobina y dos como contactos, cada uno con su respectivo símbolo, así como se muestran en las figuras 55 y 56, respectivamente, y explicando su funcionamiento a continuación en el mismo orden.

- Contacto normalmente abierto: se denota como (M), su funcionamiento es como el de un contacto estándar, pero su activación dependerá de la bobina que tenga asignada.

- Contacto normalmente cerrado: se denota como (m), igual que el contacto normalmente abierto, su activación dependerá de la bobina que tenga asignada.

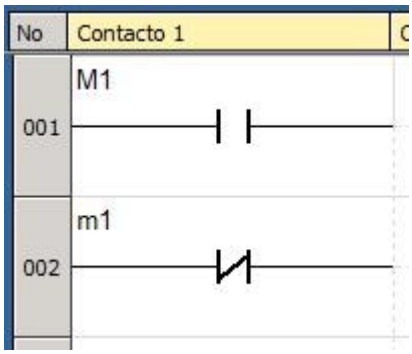
- Bobina de ajuste: este tipo de bobina se activará cuando reciba una señal positiva y permanecerá en este estado, aun cuando haya perdido la señal de excitación.

- Bobina de restablecimiento: pasará a su estado activo, justo en el momento en que reciba una señal positiva, al momento de cambiar de estado lo que hará es desactivar la bobina de ajuste que tenga asignada.

- Bobina directa: este elemento de salida se activa únicamente cuando recibe una señal positiva.

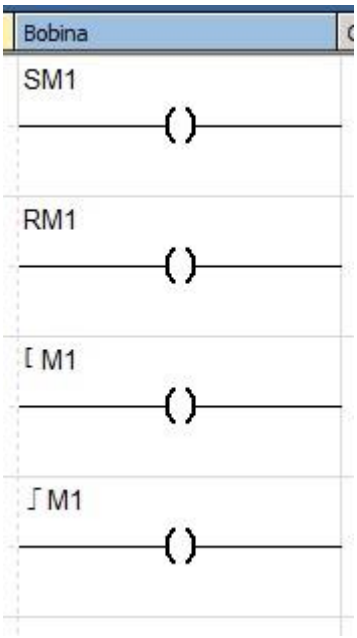
- Bobina de impulsión: también elemento de salida que se activará solamente con una señal de pulso o flanco ascendente.

Figura 55. **Contactos de relé auxiliar**



Fuente: elaboración propia.

Figura 56. **Bobinas de relé auxiliares**



Fuente: elaboración propia.

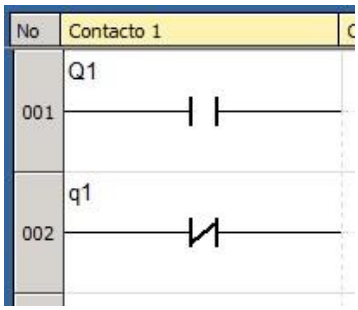
4.1.3.4. Salidas digitales

Las salidas digitales son elementos de tipo bobina, pero al igual que los relés auxiliares, también se pueden usar como contactos; la característica de las salidas digitales es que estas si están ligadas a las salidas del módulo lógico y se denotan con la letra Q y su respectivo número, como se observar en las figuras 57 y 58, ofrecen los mismos tipos de elementos que los relés auxiliares y se comportan de la misma manera.

- Contacto normalmente abierto
- Contacto normalmente cerrado
- Bobina de ajuste

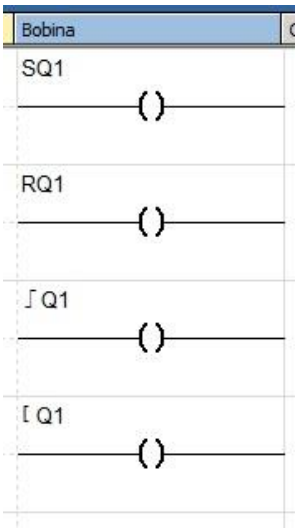
- Bobina de restablecimiento
- Bobina directa
- Bobina de impulsión

Figura 57. **Contactos de salidas digitales**



Fuente: elaboración propia.

Figura 58. **Bobinas de salidas digitales**



Fuente: elaboración propia.

4.1.3.5. Temporizadores

Estos elementos son los encargados de prolongar, retardar y permitir acciones durante cierto tiempo predeterminado, los valores de tiempo son manejados por el programador o por las necesidades del usuario.

Estos elementos también se pueden utilizar como contactos y como bobinas, se denotan con la letra (T) y su respectivo número; posee dos tipos de contactos y dos tipos de bobinas; estos se explicarán a continuación y se pueden observar en las figuras 59 y 60 en el mismo orden.

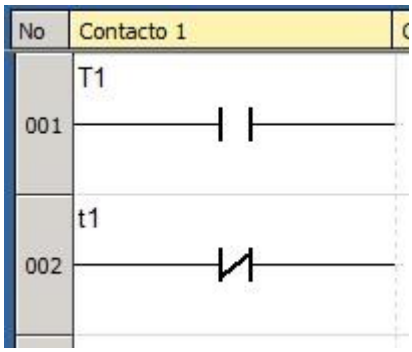
- Contacto normalmente abierto: se encuentra ligado a la salida del temporizador, si esta activa el contacto se cerrará permitiendo la conducción.

- Contacto normalmente cerrado: su funcionamiento depende del temporizador, cuando la salida está activa el contacto se abrirá, interrumpiendo la conducción.

- Bobina TT: conocida como entrada de función, se divide en diferentes tipos con funciones particulares.

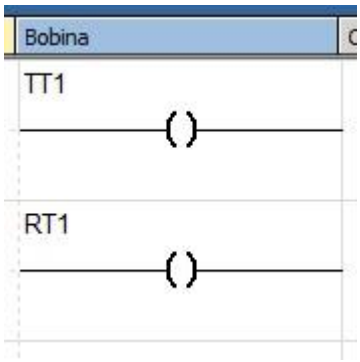
- Bobina RT: llamada también como entrada de puesta a cero, al momento de excitar la bobina se llevará a cero el valor del temporizador y estará disponible para realizar un nuevo ciclo.

Figura 59. **Contactos de temporizadores**



Fuente: elaboración propia.

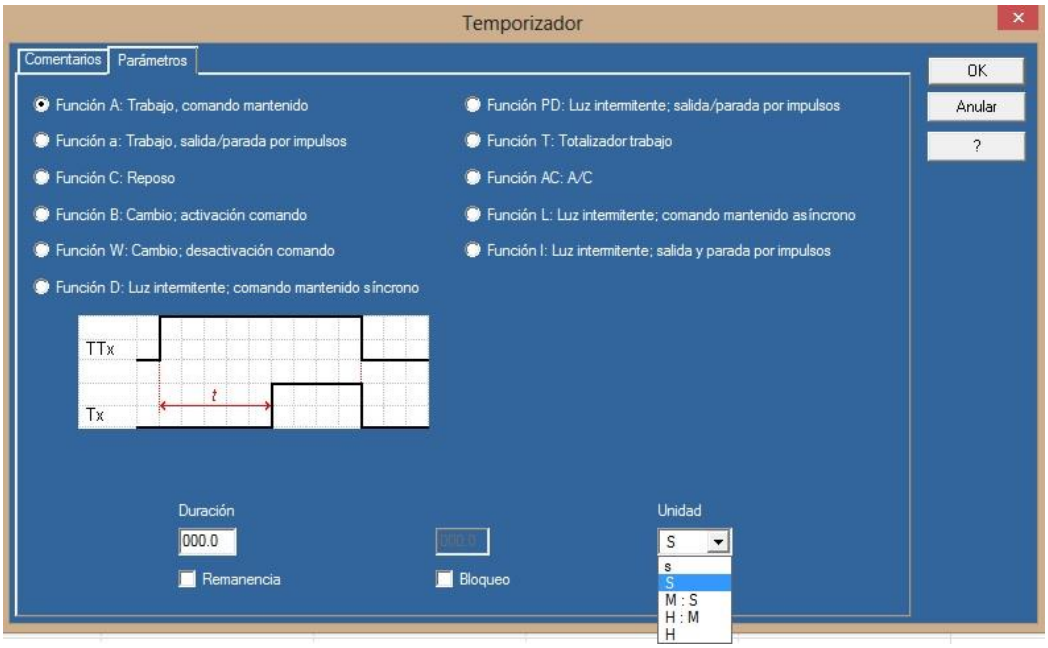
Figura 60. **Bobinas de temporizadores**



Fuente: elaboración propia.

Debido a que estos elementos poseen distintas formas de operación, estas se deben de conocer, para acceder a todas las opciones disponibles se debe de posicionar el cursor sobre el contacto o bobina y hacer clic derecho sobre el mismo y seleccionar la opción de ventana de parámetros, esta se muestra en la figura 61.

Figura 61. **Ventana de parámetros**



Fuente: elaboración propia.

Observando la ventana de parámetros, queda claro que hay muchas opciones para elegir:

- Tipo de temporizador
    - A: trabajo; comando mantenido.
    - a: trabajo; salida/parada por impulsos.
    - C: retardo de desconexión.

- B: cambio; activación comando, el temporizador se calibra a partir de un flanco se subida.
- W: cambio; desactivación comando, el temporizador se calibra por el flanco de bajada.
- D: luz intermitente, comando mantenido síncrono.
- PD: luz intermitente, salida/parada por impulsos.
- T: totalizador de trabajo.
- AC: A/C, es una combinación del tipo A y C.
- L: luz intermitente con función mantenida asíncrona.
- I: luz intermitente, salida y parada por impulsos.

- Valor de preselección: esta función depende del tipo de temporizador que se vaya a usar:
  - Para los tipo A, a, C, B, W, D, PD y T
    - Retardo a la conexión o retardo a la desconexión.
  - Para los tipo AC, L e I
    - Retardo a la conexión para AC y estado activo para L e I.

        - Retardo a la desconexión para AC y estado inactivo para L e I.

- Unidad de tiempo: se determina la unidad de tiempo que manejará el temporizador:

    - s: se maneja 1/100 de segundo

    - S: la unidad es 1/10 de segundo

    - M : S: se utiliza como unidad, minutos : segundos

    - H : M: la unidad usada es, hora : minuto

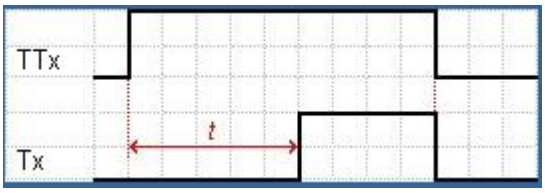
    - H: se utiliza la unidad en horas, únicamente para el tipo T

- Bloqueo de parámetros: su función es evitar que se modifiquen los parámetros de los temporizadores en uso.

- Remanencia: es utilizada para guardar el estado de los valores del temporizador, evitando perder el estado por pérdida de la alimentación.

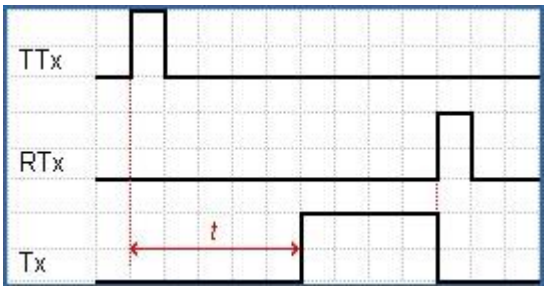
En las figuras, de la 62 a la 72, se muestra el funcionamiento de cada uno de los temporizadores mencionados anteriormente; todos los temporizadores se basan en la estimulación de la bobina TT.

Figura 62. **Temporizador A**



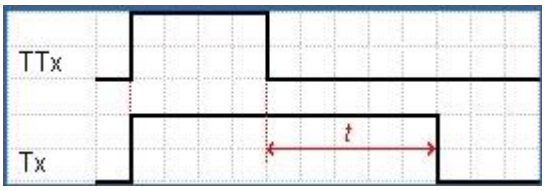
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 63. **Temporizador a**



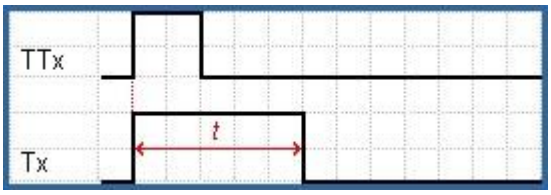
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 64. **Temporizador C**



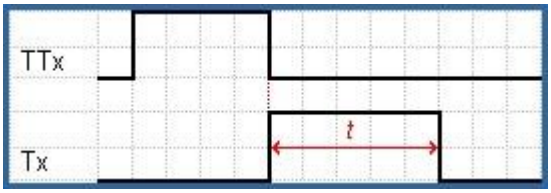
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 65. **Temporizador B**



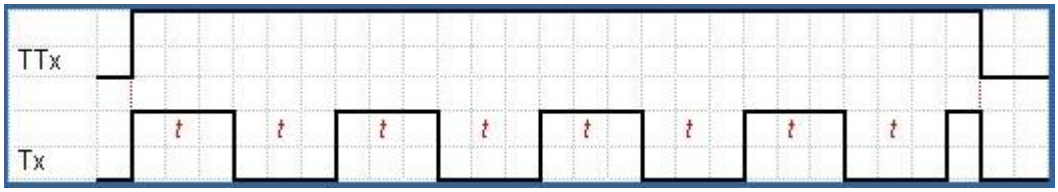
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 66. **Temporizador W**



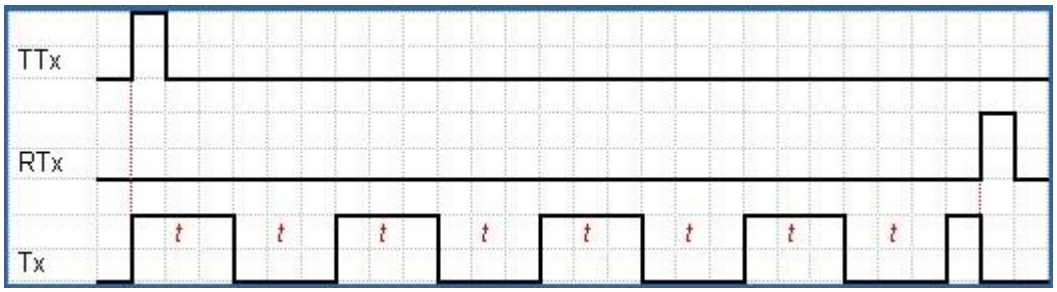
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 67. **Temporizador D**



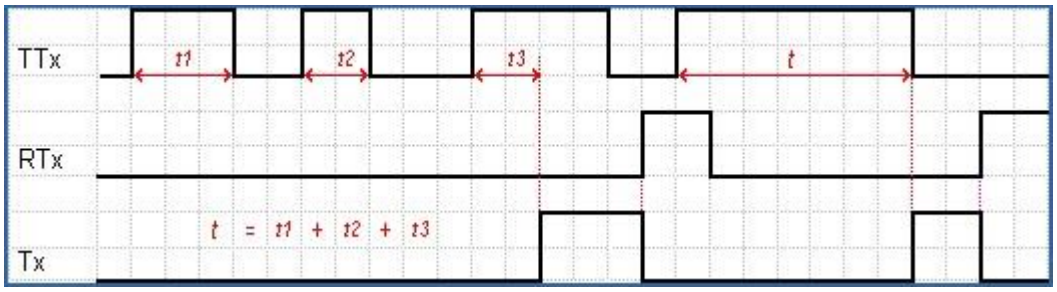
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 68. **Temporizador PD**



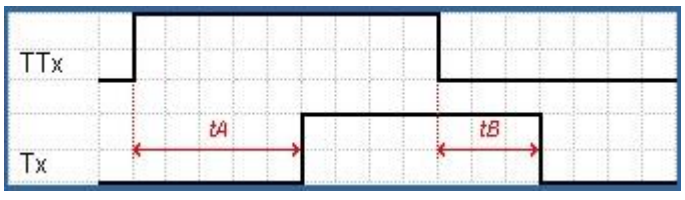
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 69. **Temporizador T**



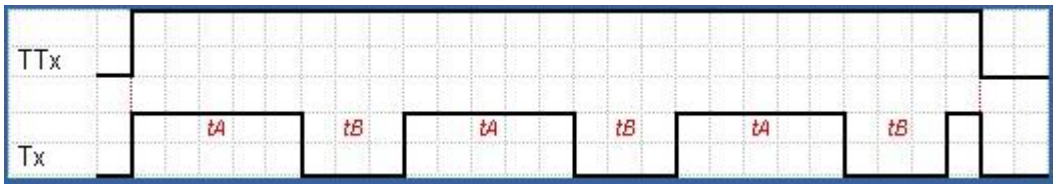
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 70. **Temporizador AC**



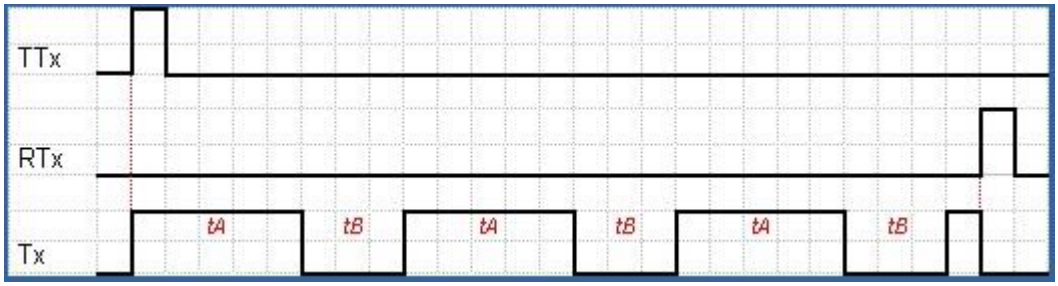
Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 71. **Temporizador L**



Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 72. **Temporizador I**



Fuente: elaboración propia, empleando Zelio Soft 2.

4.1.3.6. Contadores

Los contadores se encargan de llevar un conteo de pulsos, ya sea de forma ascendente o descendente, dependiendo de las necesidades del proyecto; los elementos se denotan con la letra C, enumerados del 1 al 9 y también con las letras desde la A hasta la V. Estos elementos se pueden usar como contactos, figura 73, dependientes de los parámetros, y como bobinas, figura 74, el valor de conteo se encuentra de 0 a 32767.

- Contacto normalmente abierto: depende del tipo de contador que tenga ligado.

    - Activo si el contador es ascendente y ha llegado al valor predeterminado.

    - Activo si el contador es descendente y ha llegado a cero.

- Contacto normalmente cerrado: funcionará de acuerdo al contador que tenga asociado.

    - Activo si el contador es ascendente y mientas el conteo no haya alcanzado el valor predeterminado.

    - Activo si el contador es descendente y mientras el valor del contador sea cero.

- Bobina CC: recibe el nombre de entrada de impulso de conteo, cada vez que la bobina se estimula, esta realizará un conteo en el sentido que se necesite.

- Bobina RC: se conoce como entrada puesta al estado inicial del contador, si se estimula esta bobina, se encargará de regresar el contador a su valor predeterminado, ya sea para un conteo ascendente o descendente.

- Bobina DC: entrada de sentido de conteo, dependiendo de si la entrada es estimulada o no, hará un conteo descendente o ascendente,

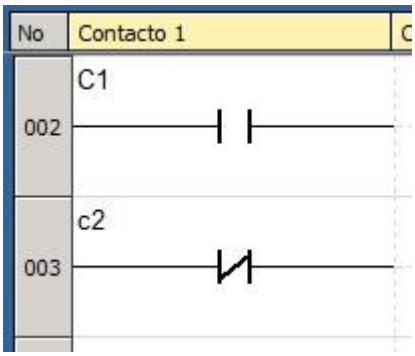
respectivamente; si la bobina no se encuentra cableada, el conteo será ascendente.

Estos elementos también poseen dos parámetros más:

- Remanencia: es usada para que en caso de falla en la alimentación, los valores que puedan tener los contadores en ese momento se guarden y cuando se recupere la alimentación los contadores no se reinicien a sus valores predeterminados.

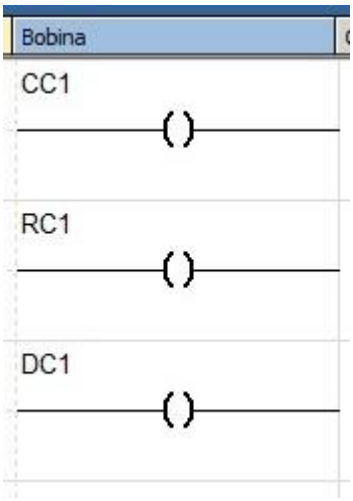
- Bloqueo: con esta función se evita que hayan modificaciones sobre los parámetros del contador.

Figura 73. **Contactos de contadores**



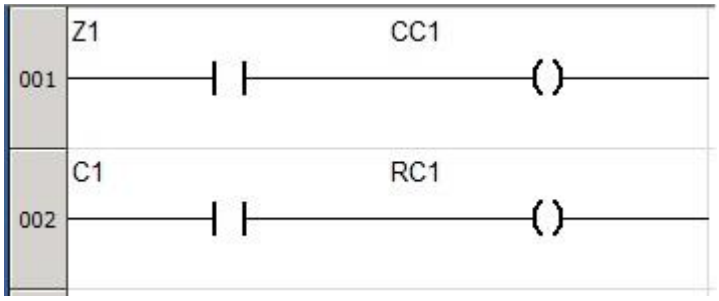
Fuente: elaboración propia.

Figura 74. **Bobinas de contadores**



Fuente: elaboración propia.

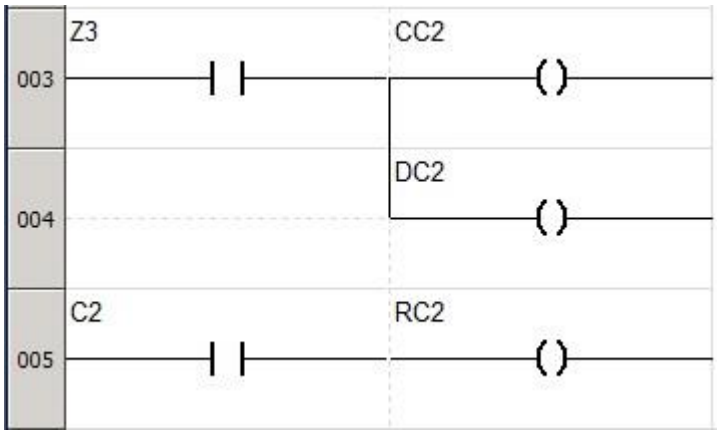
Figura 75. **Ejemplo de conteo ascendente**



Fuente: elaboración propia.

En la figura 75 se muestra un contador ascendente, por cada pulso recibido de Z1 se aumentará el contador en 1 y cuando este llegue a su valor predeterminado se activará el contacto C1, el cual a su vez reiniciará el contador nuevamente.

Figura 76. **Ejemplo de conteo descendente**



Fuente: elaboración propia.

En la figura 76 se observa un contador descendente, donde Z3 se encarga de enviar los pulsos a la bobina CC2 y a la DC2, ya que así se determina que el conteo sea regresivo, al momento de llegar a cero se activa el contacto C2, ligado a CC2, y se reinicia el conteo nuevamente.

4.1.3.7. Contador rápido

Estos elementos permiten ejecutar conteos ascendentes y descendentes, pero con la capacidad de contar pulsos con una frecuencia máxima de 1 000 Hz; al igual que otros elementos, existe la opción de utilizar contactos y bobinas, se denotan por la letra (K), figuras 77 y 78; algo que se debe tomar en cuenta para este elemento de programación es que tiene asociados los contactos I1 e I2 y que no se puede simular su funcionamiento.

- Contacto normalmente abierto: se encuentra ligado al contador y puede estar activo si:

- El contador ha llegado al valor predeterminado, tipo ascendente
- El contador ha llegado a cero, tipo descendente

- Contacto normalmente cerrado: asociado al contador y se activará si:

  - El contador no ha llegado al valor predeterminado, forma ascendente

  - El contador no ha llegado a cero, forma descendente

- Bobina TK: es la entrada para validación de función, cuando se encuentra activa, cada flanco de subida que ingresa de I1 e I2 hacen que el contador aumente o disminuya, respectivamente.

- Bobina RK: se usa para regresar al estado inicial el contador, puede ser cero si el conteo es ascendente o regresar a un valor predeterminado en caso de un conteo descendente.

Este elemento también cuenta con otros parámetros que se deben tomar en cuenta.

- Ciclo de operación: determina el comportamiento del contador rápido cuando alcanza el valor predeterminado o cero, dependiendo del tipo de conteo.

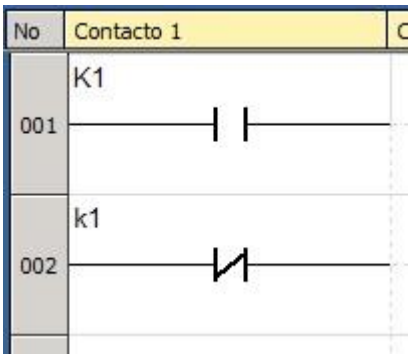
  - Único: el contador opera de forma continua y la salida se activará cuando el valor del contador supere el valor predeterminado o cuando el valor del contador sea inferior al cero.

- Repetitivo: cuando el contador llega al valor predeterminado o valor de cero, el mismo se reinicia a su valor original, en el instante de reinicio se activa la salida y permanece así durante un tiempo, el cual se puede establecer por medio de la duración de impulso.

- Remanencia: se aplica con el fin de almacenar el estado del contador y que el mismo no se pierda ante cualquier pérdida de energía en la alimentación.

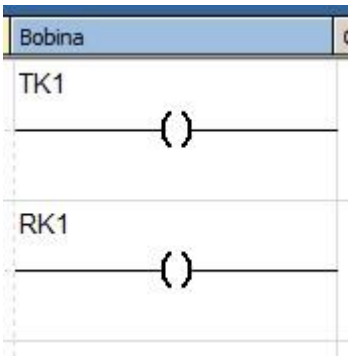
- Bloqueo: evita la modificación de parámetros establecidos dentro de la programación.

Figura 77. **Contactos de contador rápido**



Fuente: elaboración propia.

Figura 78. **Bobinas de contador rápido**



Fuente: elaboración propia.

4.1.3.8. Comparadores de contadores

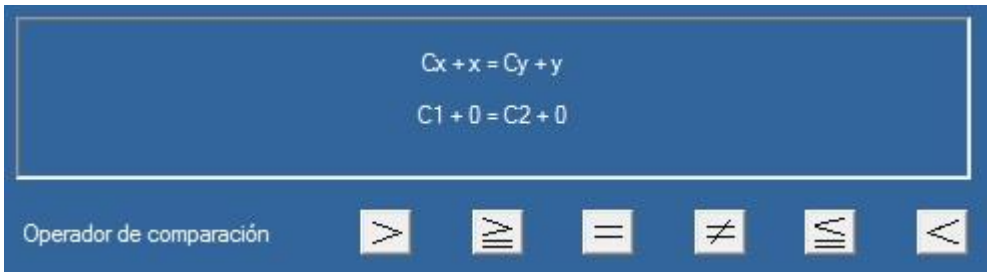
Estos elementos de programación permiten la comparación de dos contadores o bien la comparación de un contador con un valor fijo, estos elementos se pueden usar únicamente como contactos, denotados por la letra (V).

- Contacto normalmente abierto: en este caso el contacto se activa cuando la condición que se estableció se llega a cumplir, de lo contrario permanece inactivo.

- Contacto normalmente cerrado: el contacto cambia de estado cuando la condición establecida no se cumple.

Cuando se ingresa a la configuración de los parámetros de comparación, se observa la ventana de la figura 79.

Figura 79. **Configuración de comparadores de contador**



Fuente: elaboración propia.

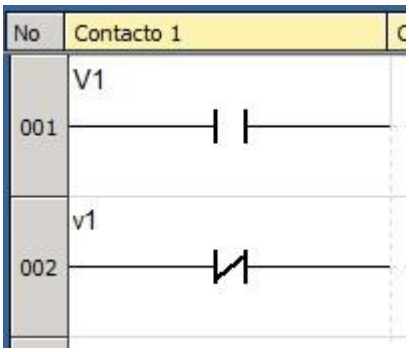
En la figura 79 se observa la fórmula general de comparación que se utiliza, Cx y Cy representan los comparadores que se quieren comparar, x y y representan un valor agregado al que pueda tener el contador o si no se selecciona un contador, pasaría a ser un valor constante de comparación.

Los operadores de comparación se indican a continuación, mencionados con respecto a la figura 79, de izquierda a derecha:

- Mayor que
- Mayor o igual que
- Igual a
- Diferente de
- Menor o igual que
- Menor que

Al igual que en otros elementos, se puede aplicar un bloqueo para que no se puedan modificar los parámetros establecidos dentro de la programación.

Figura 80. **Contactos del comparador de contadores**



Fuente: elaboración propia.

4.1.3.9. Comparadores analógicos

Estos elementos se pueden utilizar sólo como contactos, figura 81, teniendo la capacidad de:

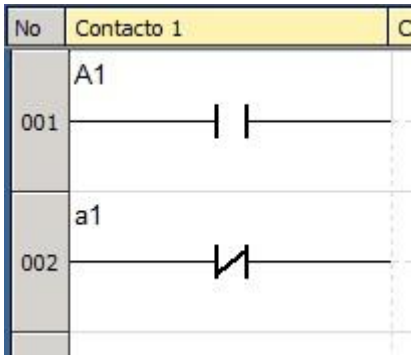
- Comparar el valor análogo de una entrada con un valor de referencia
- Comparar dos valores de entradas analógicas
- Compara dos valores análogos con un parámetro de histéresis

Si se necesita el uso de estos comparadores, es necesario que las entradas a comparar sean capaces de recibir un valor análogo, por eso, se deben usar las entradas IB hasta IG, con valores entre 0V y 9.9V.

- Contacto normalmente abierto: se encuentra asociado únicamente a los datos de comparación y cambiará a su estado activo en el momento en que se compruebe la condición establecida.

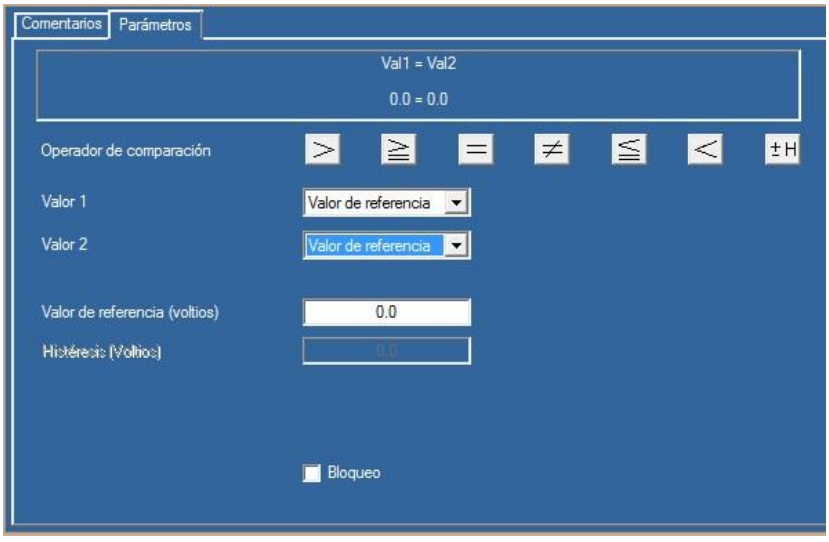
- Contacto normalmente cerrado: también ligado a los datos de comparación, permaneciendo en estado activo mientras la condición establecida no se verifica.

Figura 81. **Contactos de comparador analógico**



Fuente: elaboración propia.

Figura 82. **Configuración de comparador análogo**



Fuente: elaboración propia.

En la figura 82 se observa la fórmula del operador de comparación; donde val1 representa una de las entradas y val2 representa la otra entrada; si la comparación es con un valor de referencia, el mismo se debe establecer en el espacio correspondiente.

Los operadores pueden ser:

- Mayor que
- Mayor o igual que
- Igual
- Diferente de
- Menor o igual que
- Menor que
- Agregar parámetro de histéresis

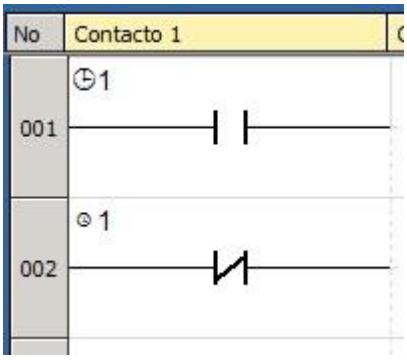
4.1.3.10. Relojes

Los relojes son elementos que permiten la ejecución de acciones dentro de la programación en horarios o franjas específicas, es posible el uso de 8 relojes, únicamente como contactos, denotados con el símbolo de un reloj y su respectivo número, cada elemento cuenta con 4 canales programables y opera de manera semanal.

- Contacto normalmente abierto: el elemento está ligado al reloj y estará en estado activo mientras el reloj se encuentre en un período válido de operación.

- Contacto normalmente cerrado: elemento también ligado al reloj y este permanecerá en estado activo mientras el reloj no esté en período de validez.

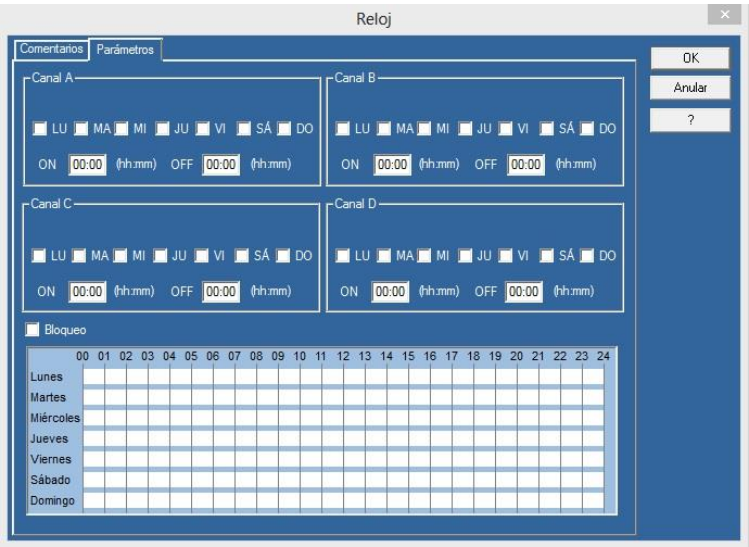
Figura 83. **Contactos de reloj**



Fuente: elaboración propia.

Para la configuración se debe seleccionar un reloj y abrir la ventana de parámetros para establecer los horarios de operación, ver figura 84, incluye la opción de bloqueo para que ningún parámetro pueda ser modificado.

Figura 84. **Configuración de reloj**



Fuente: elaboración propia, empleando Zelio Soft 2.

4.1.3.11. Bloques de texto

Con esta función es posible mostrar en la pantalla LCD algunos textos, fechas, horas y valores numéricos; muy importante es saber que se debe utilizar un solo elemento de texto para establecer todo el texto que se mostrará en la pantalla, el contenido puede ser:

- Solo texto, con capacidad máxima de 72 caracteres.

- Valores numéricos que correspondan a la salida de una función usada dentro de la aplicación.

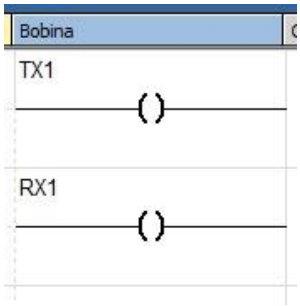
- Valores de fecha u hora, tomados de la función de relojes.

Los elementos de la función texto se pueden utilizar únicamente como bobinas, cuenta con 16 bloques de texto; en la pantalla se mostrará el bloque que se encuentre activado; si hay más de uno activado, se mostrará el bloque con el número más alto.

- Bobina TX: se encarga de sacar el contenido a la pantalla LCD, siempre y cuando se encuentre en estado activo.

- Bobina RX: encargada de desactivar la visualización del contenido en la pantalla al momento de ser activada.

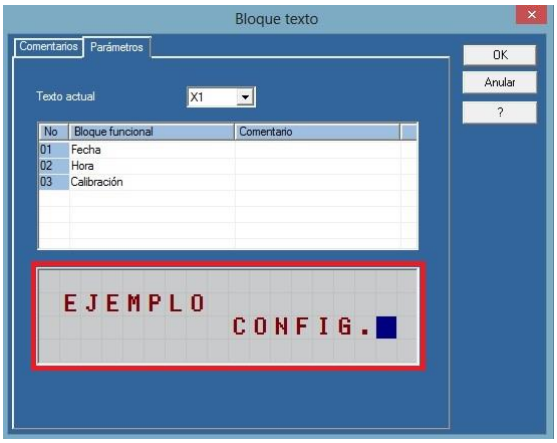
Figura 85. **Bobinas de función texto**



Fuente: elaboración propia.

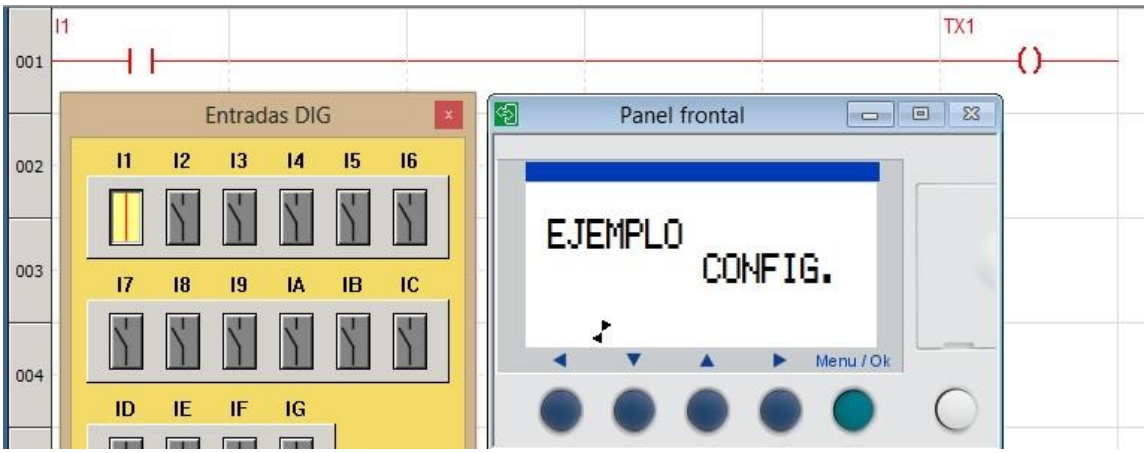
La configuración del texto se hace en la ventana de parámetros, el contenido que se desea mostrar se ingresa en el área encerrada por el cuadro rojo, ver figura 86, usando el teclado, pasando al modo de simulación; antes de iniciar la simulación se debe buscar en la pestaña de ventana la opción de panel frontal; luego iniciar la simulación y ver en el panal frontal el texto que se ingresó, ver figura 87.

Figura 86. **Configuración de texto**



Fuente: elaboración propia.

Figura 87. **Simulación de la función texto**



Fuente: elaboración propia.

4.1.3.12. Retroiluminación LCD

La función de retroiluminación para la pantalla LCD se utiliza para controlar la iluminación de la pantalla, se usa únicamente como bobina, denotada por TL; en el momento que se activa por el pulso de una tecla Zx, permite que la pantalla se ilumine.

4.1.3.13. Cambio de horario de verano/invierno

Esta función tiene la característica que durante el horario de invierno se mantiene inactivo, cuando hay cambio al horario de verano, la función también cambia y pasa a su estado activo. Su uso es solo como contacto, denotado con la letra (W). Para activar la función se debe ir a la pestaña de edición, seleccionar configuración del programa, ir a la opción de formato de fecha y activar la casilla de cambio de horario; por último, se definen las fechas, usando la zona geográfica o de manera manual.

4.1.4. Programación con lenguaje de diagrama de bloques de funciones

El entorno de programación para este lenguaje es el que se mostró en la figura 47; algunos aspectos a tomar en cuenta son:

- El programa permite modificar los colores para cada uno de los estados de los elementos, en la simulación, el fondo y para entradas y salidas, buscando la pestaña de opciones y seleccionando modificar los colores.

- Por defecto, el software muestra un fondo totalmente blanco, pero es posible agregarle fondo punteado para que usarse como guía al

momento de realizar las conexiones entre cada bloque; solo se debe buscar la pestaña de visualización; luego rejilla y seleccionar visualizar rejilla.

- Se pueden insertar figuras como, líneas, rectángulos, elipses, cuadros de texto, incluso imágenes; se ingresa a la pestaña de dibujo y seleccionando la figura que se desea utilizar.

- No cuenta con un botón especial para la verificación del programa por lo que se debe hacer de forma manual; se busca la pestaña de edición y seleccionando la opción de verificar el programa.

- Se accede al modo de edición y simulación igual que con el lenguaje escalera, como se mostraron en la figura 49 y 50, respectivamente.

- Para la programación, el entorno muestra seis opciones, ver figura 88, las cuales se listan a continuación para su posterior explicación.

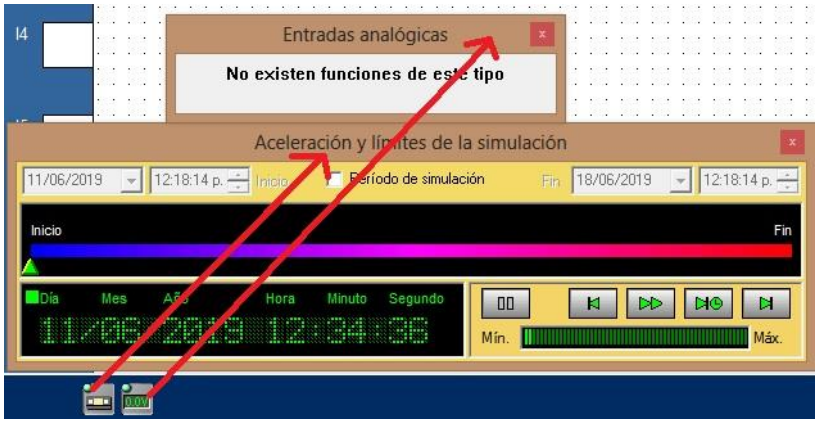
    - ENT: funciones de entradas
    - BDF: funciones estándar para lenguaje de bloques de funciones
    - GFC: funciones para diagramas funcionales en secuencia
    - LOGIC: funciones de operaciones lógicas
    - SAL: funciones de salidas
    - AFB: funciones de bloques de aplicación

- El entorno de simulación solo posee dos elementos, entradas analógicas y el control de simulación, figura 89.

Figura 88. **Elementos de programación en lenguaje de bloques de funciones**

Fuente: elaboración propia.

Figura 89. **Elementos de simulación para el lenguaje de bloques de funciones**



Fuente: elaboración propia.

4.1.4.1. Funciones de entradas

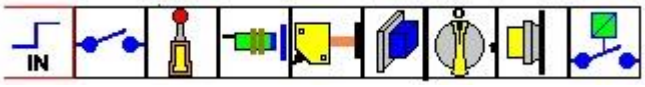
Las funciones de entrada se dividen en 15 elementos y algunos se dividen en subelementos para otras funciones, estos son:

- Entrada DIG: es una función que permite el ingreso de valores digitales, es la función por defecto y a parte se pueden escoger otras funciones,

todas tienen su propio símbolo, se muestran en la figura 90 en el mismo orden, estas son:

- Contacto
- Interruptor de fin de carrera
- Detector de proximidad
- Detector de presencia
- Botón pulsador luminoso
- Conmutador
- Botón pulsador
- Relé normalmente abierto

Figura 90. **Funciones de entrada DIG**



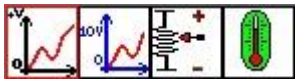
Fuente: elaboración propia.

- Entrada analógica: esta función se encuentra disponible únicamente en los controladores que se alimentan con corriente continua, la tensión de entrada se convierte a un valor entero entre 0 y 255 por medio de un D/A; las funciones solo se pueden usar en las entradas IB e IG; tiene la posibilidad de usarse como entrada entre 0 V y 10 V o como entrada tipo potenciómetro, presenta las siguientes opciones, ver figura 91:

  - Entrada predeterminada
  - Entrada
  - Potenciómetro

- Temperatura

Figura 91. **Funciones de entrada analógica**



Fuente: elaboración propia.

- Entrada digital filtrada: para esta función se antepone un filtro, con el fin de atenuar y de ser posible eliminar cualquier tipo de perturbación. La señal entrante se filtra con el uso de un algoritmo de detección de nivel, cero o uno, medida durante un tiempo establecido; entonces, si la entrada se mantiene estable durante el tiempo de detección, la salida de la función tomará el valor de la entrada; de lo contrario, permanecerá inestable, el símbolo de la función se muestra en la figura 92.

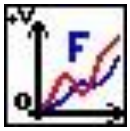
Figura 92. **Función de entrada digital filtrada**

Fuente: elaboración propia.

- Entrada analógica filtrada: a esta función también se le agrega un filtro pasa bajos antes de la entrada, se encuentran disponibles en controladores alimentados con corriente continua; la entrada se convierte a la salida en valores comprendidos de 0 a 255 por medio de un A/D; la función se puede usar en las entradas IB a IG; puede ser como entrada

de 0 V a 10 V o como potenciómetro, el único parámetro que se debe ingresar es la frecuencia de corte del filtro comprendida entre 0,06 Hz y 88,25 Hz, su símbolo se muestra en la figura 93.

Figura 93. **Función de entrada analógica filtrada**



Fuente: elaboración propia.

- Entradas especiales del lenguaje de bloques: por el tipo de programación, existen entradas especiales:

  - Tipo constante digital: tiene dos tipos, constante en 1 y constante en 0, se usan para poner las entradas en 1 o 0.

  - Constante numérica: es un valor numérico, entero, que va de -32768 a +32767, se puede usar en funciones como ganancia, comparación y disparo.

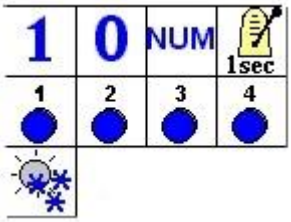
  - Tipo intermitente: entrada de tipo intermitente que se activa cada segundo.

  - Tipo botón: estas corresponden a las teclas Zx del lenguaje escalera, existen cuatro para su uso.

  - Horario de verano/invierno: opera con respecto al horario, en horario de verano se encuentra activa, de lo contrario permanece

inactiva, su configuración es igual que en el caso de lenguaje escalera.

Figura 94. **Funciones especiales de entrada**



Fuente: elaboración propia.

- Entrada de un entero: función capaz de que a la entrada exista un número entero de 16 bits, -32768 a +32767, proveniente únicamente de las entradas de extensiones que estén conectadas al módulo central; pueden ir de la entrada J1XT1 a J4XT1, el símbolo se observa en la figura 95.

Figura 95. **Función de entrada de un entero**

Fuente: elaboración propia.

- Entrada analógica de un entero de 10 bits: es un tipo de entrada disponible solo para controladores compatibles con extensiones de

entradas y salidas analógicas de código SR3XT43BD; la salida puede ir de 0 a 1023; su símbolo se muestra en la figura 96.

Figura 96. **Función de entrada analógica de un entero de 10 bits**

Fuente: elaboración propia.

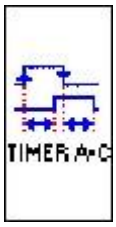
4.1.4.2. Funciones estándar para lenguaje de bloques de funciones

Estas funciones se dividen en 33 elementos, que solo se pueden aplicar en lenguaje de bloques.

- Temporizador A/C: función que permite retardar, prolongar y realizar controles sobre otras funciones durante un tiempo predeterminado, permite aplicar remanencia y bloqueo de parámetros, puede operar bajo:

    - Función A: retardo a la conexión
    - Función C: retardo a la desconexión
    - Función A/C: combinación entre la función A y la C

Figura 97. **Función de temporizador A/C**



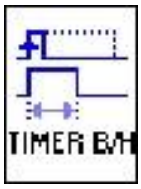
Fuente: elaboración propia.

- Temporizador B/H: se conoce como tiempo de espera, es una función que crea un impulso en la salida a partir del flanco de subida de la entrada; se puede aplicar remanencia y bloque de parámetros; puede operar bajo:

    - Función B: sin importar la duración del pulso de activación, la salida quedará en alto el tiempo que se haya predeterminado.

    - Función H: la salida permanecerá en alto por un tiempo predeterminado si y solo si el pulso de activación tiene una duración igual o mayor; de lo contrario, la salida durará activa hasta el flanco de bajada del pulso de entrada.

Figura 98. **Función de temporizador B/H**



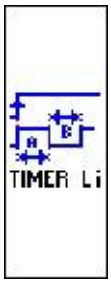
Fuente: elaboración propia.

- Temporizador LI: conocido como temporizador doble, su función es generar una serie de impulsos a partir del flanco de subida de la entrada y mientras este activa; permite establecer la duración del impulso y el tiempo entre cada uno; es posible aplicar remanencia y el bloqueo de parámetros, puede operar bajo:

    - Función Li: la salida se inicia en alto cuando aparece el flanco de subida en la entrada.

    - Función L: la salida inicia en bajo a partir del flanco de subida de la entrada.

Figura 99. **Función de temporizador LI**

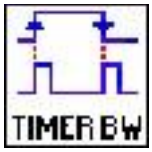


Fuente: elaboración propia.

- Impulsos en flanco: se conoce también como temporizador BW, su función es sacar un impulso en la salida a partir de los flancos que llegue a la entrada, tanto la entrada como la salida son de valores digitales, el impulso puede darse cuando:

    - La entrada pasa de inactiva a activa, flanco de subida.
    - La entrada pasa de activa a inactiva, flanco de bajada.
    - La entrada pasa por ambos flancos, de activa a inactiva y de inactiva a activa.

Figura 100. **Función de impulsos en flancos**



Fuente: elaboración propia.

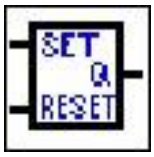
- Telerruptor: conocido como bistable, su función es que en cada flanco de subida que aparezca en la entrada, el estado de la salida va a cambiar, si se llega a activar la puesta a cero del bloque, la salida permanecerá inactiva siempre.

Figura 101. **Función de telerruptor**

Fuente: elaboración propia.

- Báscula RS: función de SET/RESET, posee dos entradas y una salida; cuando se activa la entrada SET, la salida permanece activa aún cuando ya no exista el pulso en la entrada; por su parte, si se activa la entrada RESET, la salida se desactivará y permanecerá en ese estado hasta que se vuelva a activar la entrada SET.

Figura 102. **Función de báscula RS**



Fuente: elaboración propia.

- Función booleana: esta función cuenta con cuatro entradas, por lo que hay 16 combinaciones posibles a la entrada; cada combinación se encuentra en la tabla de verdad y el valor de salida para cada combinación se puede ajustar; la única condición es que sus entradas se encuentren conectadas; se puede trabajar con:

    - Salida alta si el resultado es verdadero: en ese caso se mantiene el valor de salida que se ingresó a la tabla.

    - Salida baja si el resultado es verdadero: salida tomará el valor contrario al que se configuró en la tabla de verdad.

Figura 103. **Función booleana**

Fuente: elaboración propia.

- Programador de levas: esta función se encarga de controlar 8 ruedas de levas, cada rueda representa una salida y su estado se relaciona con la posición actual de las ruedas del árbol, su máximo es de 50 posiciones; la dirección de la leva, adelante o inversa, se controlan con la activación de su entrada respectiva; se puede aplicar un bloque de parámetros y también un reinicio que obligará a la rueda a regresar a su posición inicial.

Figura 104. **Función de programación de levas**

Fuente: elaboración propia.

- Contador progresivo/regresivo con preselección: esta función permite realizar conteos de forma ascendente o descendente, de 0 al valor preseleccionado o viceversa; el valor tope es 32 767, se puede aplicar remanencia y bloque de parámetros; todos los valores son digitales.

Figura 105. **Función de contador progresivo/regresivo con preselección**

Fuente: elaboración propia.

- Contador progresivo/regresivo: función que permite realizar un conteo ascendente o descendente, pero con la característica de que el valor preestablecido, ya sea para un conteo ascendente o descendente será tomado de un bloque externo; también, se puede aplicar remanencia y bloqueo de parámetros a la función.

Figura 106. **Función de contador progresivo/regresivo**

Fuente: elaboración propia.

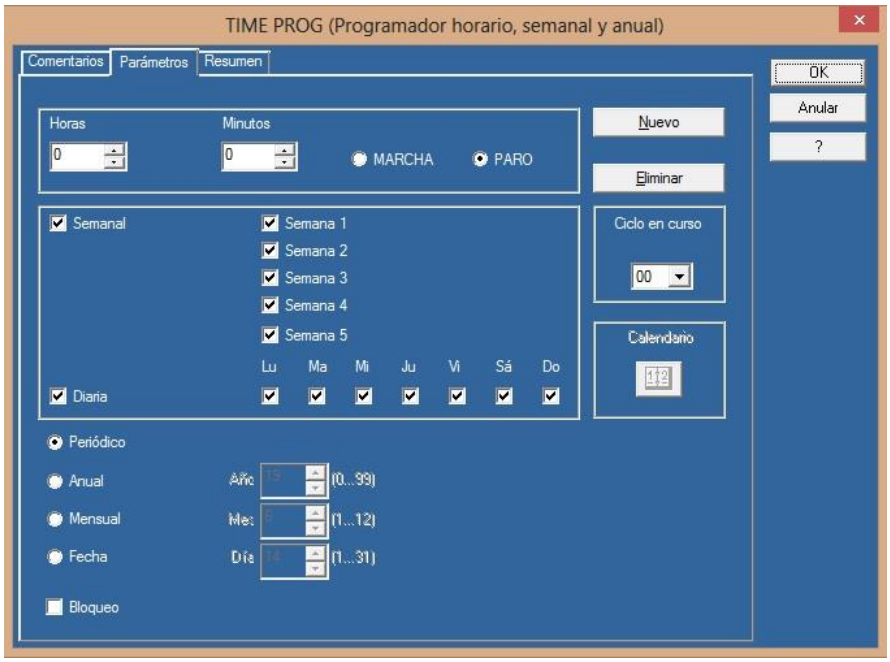
- Contador horario de preselección: con esta función es posible medir la duración de activación de la entrada; cuando se alcanza un tiempo predeterminado, la salida pasará a un estado activo; se puede ajustar en horas y minutos; activando la puesta a cero, se desactiva la salida; se puede aplicar remanencia a la función, todos los valores son de tipo digital.

Figura 107. **Función de contador horario de preselección**

Fuente: elaboración propia.

- Programador horario, semanal y anual: función encargada de activar acciones en los horarios o franjas de operación establecidas previamente; es posible definir si la salida estará activa o no durante el ciclo; el ciclo puede ser anual, mensual o semanal; se puede aplicar bloque de parámetros a la función. En la figura 108 se observa el cuadro de configuración; en la pestaña de resumen se puede ver cada uno de los horarios establecidos.

Figura 108. **Configuración de la función de programación de horario, semanal y anual**



Fuente: elaboración propia, empleando Zelio Soft 2.

Figura 109. **Función de programador horario, semanal y anual**

Fuente: elaboración propia.

- Ganancia: función que permite convertir valores análogos por cambio de escala y agregarle un valor de *offset*; esta función posee dos entradas y

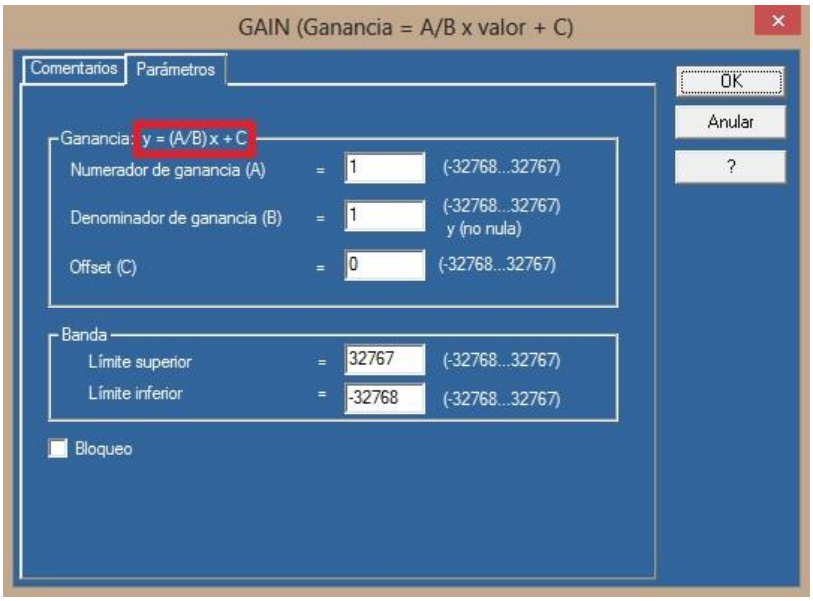
una salida; también, es posible aplicar el bloque de parámetros para evitar alteraciones en los valores ingresados; la configuración y la fórmula se muestran en la figura 110, encerrada en el cuadro rojo.

- Validación de función: es una entrada de tipo digital, determina el la activación o no de la función con la presencia o ausencia de un valor en su entrada respectivamente.

- Entrada de cálculo: es el valor analógico de tipo entero que se le aplicará la fórmula para obtener otro dato en salida.

- Salida de cálculo: será el nuevo valor, obtenido de la aplicación de la fórmula al dato que se encuentre en la entrada de cálculo.

Figura 110. **Configuración de la función ganancia**



Fuente: elaboración propia.

Para la fórmula mostrada en la figura anterior, y representa el valor de salida, (A/B) es la ganancia que se aplicará a dato de entrada x; C es el valor de *offset* que se dará a la salida.

Figura 111. **Función de ganancia**

Fuente: elaboración propia.

- Disparador: esta función permite analizar un valor análogo, utilizando dos valores de referencia para establecerlos como umbrales; entonces, la salida cambiará de estado si el valor de entrada es inferior al valor mínimo o si la entrada es superior al valor máximo; de lo contrario, la salida no cambiará su estado; los valores de comparación provienen de otros bloques, no son predeterminados.

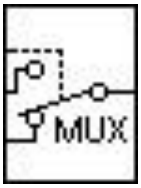
Figura 112. **Función de disparo**

Fuente: elaboración propia.

- Multiplexado: es una función que consta de tres entradas y una salida; las entradas son conocidas como canal A, canal B; ambas aceptan

únicamente datos de tipo entero; la tercera entrada se conoce como selección, con esta se determina que canal estará presente en la salida; por tanto, si selección está activa, la salida será el canal B; de lo contrario, será el canal A.

Figura 113. **Función de multiplexado**



Fuente: elaboración propia.

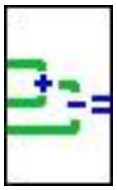
- Comparación: conocida como comparación en zona, es capaz de comparar un valor entre dos límites, los límites máximo y mínimo corresponden a la zona; el valor a comparar y los límites deben ser de tipo entero; puede operar bajo dos términos: marcha en zona y paro en la zona, donde la salida estará activa si el valor está en la zona o inactiva cuando este en la zona, respectivamente.

Figura 114. **Función de comparación en zona**

Fuente: elaboración propia.

- Función aritmética ADD/SUB: función dedicada a realizar operaciones sencillas de suma y resta con enteros; posee 3 entradas y una salida; la función por defecto suma la entrada 1 con la entrada 2 y resta la entrada 3; si se desee realizar una suma sencilla no se usa la entrada 3; si se desea hacer una resta sencilla no se usa la entrada 1 o la entrada 2.

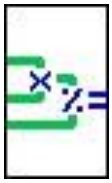
Figura 115. **Función de suma y resta**



Fuente: elaboración propia.

- Función aritmética MUL/DIV: función dedicada a realizar operaciones sencillas de multiplicación y división con enteros; posee 3 entradas y una salida; la función, por defecto multiplica la entrada 1 con la entrada 2 y divide con la entrada 3; si se desee realizar una multiplicación sencilla no se usa la entrada 3; si se desea hacer una división sencilla no se usa la entrada 1 o la entrada 2.

Figura 116. **Función de multiplicación y división**



Fuente: elaboración propia.

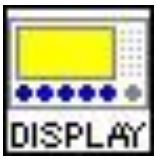
- Texto: función utilizada con el fin de visualizar textos, una fecha, una hora y valores numéricos en la pantalla LCD; se debe utilizar un solo bloque de función para establecer todo el contenido que se mostrará en la pantalla, con un máximo de 72 caracteres.

Figura 117. **Función de texto**

Fuente: elaboración propia.

- Visualización en la pantalla LCD: es una variación de la función texto; es posible visualizar texto, fecha, hora y valores numéricos en la pantalla LCD, con un máximo de 72 caracteres y aceptando valores con una coma decimal.

Figura 118. **Función visualización en pantalla LCD**



Fuente: elaboración propia.

- Comparación de dos valores: función que permite comparar dos valores analógicos; los valores deben ser de tipo entero y la habilitación debe ser

de tipo digital; las comparaciones que se pueden utilizar son las de mayor que, mayor o igual que, igual a, diferente de, menor o igual que, menor que.

Figura 119. **Función de comparación de dos valores**

Fuente: elaboración propia.

- Estado: función que permite acceder a los estados del módulo y poder modificar el programa con base a los estados; es un bloque sin entradas; posee únicamente siete salidas.

    - Estado de alarma: se activa en el momento de encontrar un error o una alarma en el módulo, el código de error estará presente en una de las siete salidas, número de alarma.

    - Monitorización RUN: el estado se activa cuando el programa se ejecuta correctamente en el módulo.

    - Parámetros RUN: saca un impulso cuando el programa se ejecuta de manera correcta y cuando se trata de modificar parámetros de otros bloques.

- Inicio en frío: se encuentra en un nivel alto durante el primer ciclo de ejecución de un programa, justo cuando pasa de paro a marcha.

- Inicio en caliente: se encuentra en un nivel alto durante el primer ciclo de ejecución de un programa, justo cuando se pasa del modo marcha a paro dentro de la simulación o cuando se restablece la energía luego de un corte y el programa está en marcha.

- Ciclo de parpadeos: función que durante cada ejecución del programa saca una señal periódica de flancos ascendentes y descendentes.

Figura 120. **Función de estado**

Fuente: elaboración propia.

- Archivo: función capaz de realizar copias de seguridad para dos valores de manera simultánea, todo relacionado con los datos de fecha y hora; posee cuatro entradas y ocho salidas.

  - Entrada de memorización: cada vez que esta entrada pasa de estado inactivo a activo por medio de un flanco positivo, se memoriza el dato que este en las entradas de valor.

- Entrada de puesta a cero: siempre que se activa esta entrada, la salida de archivo válido pasara a estado inactivo de forma forzada.

- Entrada de valor 1: el dato que se encuentre, será el primero en almacenarse, incluyendo su fechado.

- Entrada de valor 2: es el segundo dato en almacenarse.

- Salida de archivo válido: se encarga de indicar que si llevo a cabo la memorización de los datos.

- Salida de minutos: se encarga de mostrar el minuto de la información de fechado de los datos, de 0 a 59.

- Salida de hora: muestra la hora de fechado de los datos, de 0 a 23.

- Salida de día: muestra el día de fechado de los datos, de 1 a 31.

- Salida de mes: indica el mes de fechado de los datos, de 1 a 12.

- Salida de año: indica el año de fechado de los datos, entre 0 y 99.

- Salida de archivo 1 y 2: indica los valores presentes en ambas entradas.

Figura 121. **Función de archivo**

Fuente: elaboración propia.

- Contador rápido: función con capacidad de contar impulsos con una frecuencia máxima de 1KHz. Es posible aplicarle remanencia y bloqueo de parámetros; el contador se encuentra ligado a las entradas I1 e I2; estas entradas no se deben de cablear, un flanco ascendente en I1, aumenta el contador y un flanco ascendente en I2, disminuye el contador.

Figura 122. **Función de contador rápido**

Fuente: elaboración propia.

- Conversión de palabras a bits: función con capacidad de aceptar, en su única entrada, datos de tipo entero, de 16 bits, para que se pueda realizar la conversión a un valor binario, repartido en las 16 salidas

disponibles, siendo el bit 1, menos significativo y el bit 16, el más significativo.

Figura 123. **Función de conversión de palabras a bits**

Fuente: elaboración propia.

- Conversión de bits a palabras: la función dispone de 16 entradas, donde el bit 1 es el menos significativo y el bit16 es el más significativo; los 16 bits componen una palabra y la función se encarga de convertir ese valor binario a uno de tipo entero y mostrarlo en la única salida disponible en el bloque.

Figura 124. **Función de conversión de bits a palabras**

Fuente: elaboración propia.

- Entrada de enlace serie: conocida como SLIn, permite transmitir por medio de un enlace de tipo serial, los datos a segmentos de memoria con

dirección fija dentro del controlador; la función cuenta únicamente con ocho salidas, nombradas de entrada 1 a entrada 8; con esto se pueden emplear los datos almacenados en los segmentos de memoria a través de la aplicación programada; se tienen un rango de selección para los segmentos que se desean utilizar, 1-8, 9-16 y 17-24.

Figura 125. **Función de entrada de enlace serie**

Fuente: elaboración propia.

- Salida de enlace serie: conocida como SLOut, capaz de enviar datos almacenados en segmentos de memoria con dirección fija hacia otros equipos, por medio de un enlace serie, contiene ocho entradas de tipo entero, con estas es posible escribir los datos que se almacenarán en los segmentos de memoria; también, se pueden seleccionar tres rangos de segmentos, 25-32, 33-40 y 41-48.

Figura 126. **Función de salida de enlace serie**

Fuente: elaboración propia.

- Posición del sol: función capaz de calcular la posición del sol, la función consta de cuatro entradas y dos salidas, indicando la posición del sol.

    - Entrada de activación: se debe mantener en estado activo para que la función opere de manera correcta.

    - Entrada de longitud: debe ser de tipo entero, comprendido de -18 000 a +18 000, capaz de representar la longitud de la ubicación del equipo, de 180 grados oeste hasta 180 grados este.

    - Entrada de latitud: debe ser un dato de tipo entero, entre -9 000 hasta +9 000, para representar la latitud de ubicación del equipo, de 90 grados sur hasta 90 grados norte.

    - Entrada de zona horaria: dato de tipo entero que representa la diferencia horaria, en minutos, entre el UTC y el país donde se encuentre el controlador.

    - Salida de ángulo de elevación: dato de tipo entero, representa la altura del sol, de 90 grados sur a 90 grados norte; el ángulo positivo o negativo indica si el sol está por encima o por debajo del horizonte respectivamente.

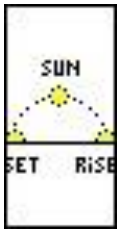
    - Salida de ángulo de acimut: dato de tipo entero que representa la rotación necesaria para estar uno mismo frente al sol desde la dirección norte; el valor está entre -18 000 y +18 000, equivalente a 180 grados oeste y 180 grados este.

Figura 127. **Función de posición del sol**

Fuente: elaboración propia.

- Orto/Ocaso: función dedicada a calcular las horas de orto y ocaso con base en la longitud y latitud donde se encuentra el controlador; su salida se encontrará en alto cuando el sol haya salido y en bajo cuando el sol se haya puesto.

Figura 128. **Función de orto/ocaso**



Fuente: elaboración propia.

4.1.4.3. Funciones para diagramas funcionales en secuencia

Este tipo de funciones son similares al *grafcet*, este permite representar de manera gráfica y estructurada el funcionamiento de un sistema secuencial. Un gráfico con el uso de estas funciones se debe leer de arriba hacia abajo; está

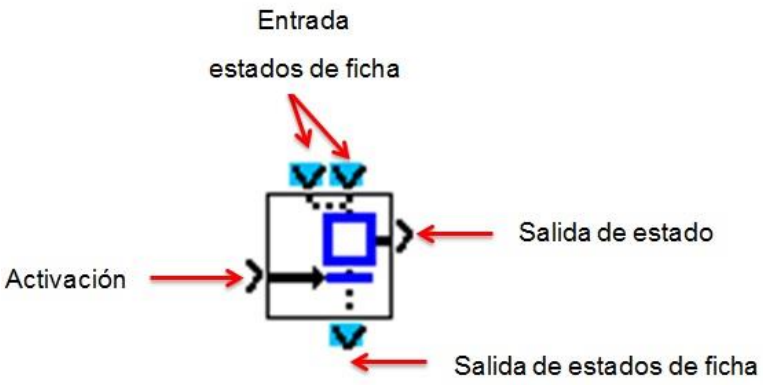
compuesto por etapas y transiciones; las etapas operan una tras otra, gobernadas por las transiciones y cada etapa posee una salida de etapa que se encarga de enviar las órdenes a otras funciones.

Las etapas y transiciones se usan para representar fases y al mismo tiempo poder controlarlas; cada fase se representa por una etapa, cuando la etapa esté en funcionamiento se dice que se encuentra en estado activo y la etapa contiene una ficha de estado.

Los bloques de etapas deben activarse para que los estados que se encuentren presentes en las entradas de ficha puedan pasar a la salida de estado de ficha, que irá conectada a otra etapa o transición; además, el bloque de etapa cuenta con una salida que se puede enviar a otros bloques BDF, en la figura 129 se muestran las partes de la etapa.

Se encuentran tres tipos de etapas disponibles: etapa inicial, ver figura 130, siempre debe de haber por lo menos una de estas para inicializar todo el gráfico; otra es la etapa inicial reiniciable, ver figura 131, con la capacidad de que si se activa la entrada de puesta a cero, se reiniciará todo el gráfico; otra es la etapa común, ver figura 132, la cual se encontrará aguas abajo de la etapa de inicialización.

Figura 129. **Partes del bloque etapa**

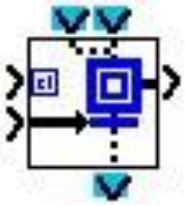


Fuente: elaboración propia.

Figura 130. **Etapa inicial**

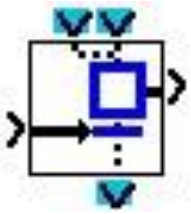
Fuente: elaboración propia.

Figura 131. **Etapa inicial reiniciable**



Fuente: elaboración propia.

Figura 132. **Etapa**

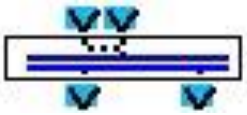


Fuente: elaboración propia.

Dentro de las funciones GFC también se encuentran las divergencias y convergencias, ambas con las opciones de funcionamiento Y u O.

La divergencia Y, ver figura 133, no necesita de activación, únicamente necesita que la salida de estado de ficha de la etapa aguas arriba esté presente en algunas de sus entradas y así la ficha presente se dividirá en dos fichas para que se activen dos fases por medio de otras etapas y que operen de forma paralela.

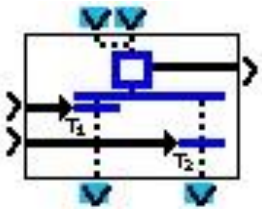
Figura 133. **Divergencia Y**



Fuente: elaboración propia

El bloque de divergencia O, ver figura 134, consta de dos entradas de activación y dos salidas de estado de ficha, lo que permite el uso de una salida de ficha u otra dependiendo de la entrada activada y en algunos casos se pueden utilizar ambas salidas.

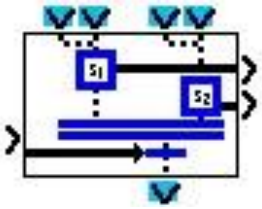
Figura 134. **Divergencia O**



Fuente: elaboración propia.

Convergencia Y, figura 135, esta función realiza el proceso inverso de la divergencia Y, la función posee dos entradas de ficha, entonces, el bloque pasa de tener dos fases; simultáneas, a una sola fase, por tanto, para que se inicie la única fase, las fases anteriores deben terminar también de manera simultánea, si al momento de activar la transición solo hay una ficha, no ocurrirá nada.

Figura 135. **Convergencia Y**



Fuente: elaboración propia.

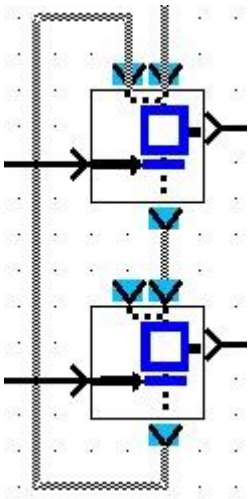
La convergencia O, ver figura 136, opera de manera inversa a la divergencia O, la función puede recibir fichas de dos funciones aguas arriba en sus entradas, al momento de activar la transición; a la salida circularán una de las fichas o ambas, si se encuentran conectadas, pasando así a la función que se encuentre aguas abajo.

Figura 136. **Convergencia O**

Fuente: elaboración propia.

Con las funciones GFC es posible realizar bucles, estos se hacen con la finalidad de que las fases o series de fases se mantengan siempre activadas, el bucle se realiza a partir de la salida de ficha de estado de la última etapa y llevándola hacia la entrada de ficha de estado de alguna etapa ubicada aguas arriba para que toda la función se quede operando de manera infinita.

Figura 137. **Ejemplo de bucle**

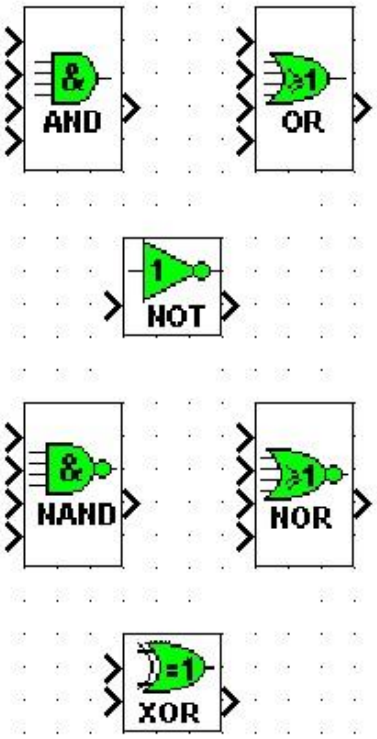


Fuente: elaboración propia.

4.1.4.4. Funciones lógicas

El lenguaje de bloques de funciones también permite el uso de funciones lógicas, es posible el uso de siete funciones, AND, OR, NAND, NOR, NOT y XOR, las primeras cuatro funciones poseen cuatro entradas y las últimas solo una y dos entradas respectivamente, todas funciones son de una sola salida de tipo digital.

Figura 138. **Bloques de funciones lógicas**



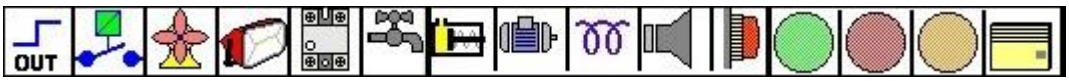
Fuente: elaboración propia.

4.1.4.5. Salidas digitales

- Salidas: solo pueden ser de dos tipos: de estado sólido para algunos módulos lógicos de corriente continua y salidas de relé para módulos de corriente alterna; las salidas que se tienen disponibles son las que se muestran en la figura 139 y que se enumeran a continuación en el mismo orden.

    - Salida digital
    - Relé normalmente abierto
    - Ventilador
    - Lámpara
    - Relé estático
    - Válvula
    - Cilindro
    - Motor
    - Resistencia
    - Señal acústica
    - Indicador luminoso
    - Indicador luminoso verde
    - Indicador luminoso rojo
    - Indicador luminoso naranja
    - Calefacción

Figura 139. **Tipos de salidas digitales**



Fuente: elaboración propia.

- Retroiluminación de la pantalla LCD: permite controlar por medio del programa la iluminación de la pantalla que posee el módulo; si la entrada permanece activa, la pantalla también lo estará; teniendo en cuenta que esta función no se debe conectar a las salidas del controlador, el símbolo, figura 140, estará en verde y amarillo cuando la pantalla está inactiva y activa, respectivamente.

Figura 140. **Función de retroiluminación de la pantalla LCD**

Fuente: elaboración propia.

- Salida de entero: es una función que tiene la capacidad para enviar un entero con formato de 16 bits, de -32 768 a +32 767, a determinadas extensiones de los controladores, específicamente, salidas de O1XT1 a O4XT1.

Figura 141. **Función de salida de entero**

Fuente: elaboración propia.

- Salida de entero de 10 bits: es una función que utiliza la extensión SR3XT43BD, específicamente los espacios de salida QBXT2 y QCXT2, por medio de un A/D de 10 bits se convierte el entero en una tensión de salida que varía entre 0 V y 10 V, todos los bits en alto representan la tensión máxima.

Figura 142. **Función de salida de 10 bits**

Fuente: elaboración propia.

4.1.4.6. Función de bloque de aplicación

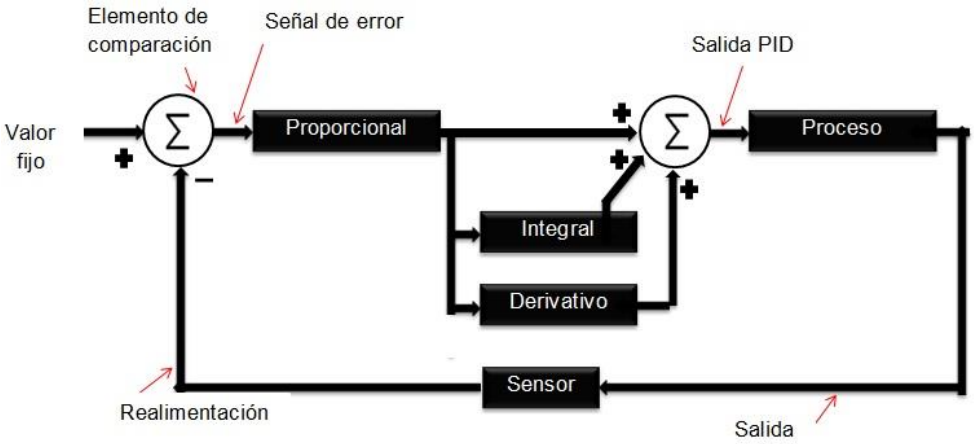
Los bloques de aplicación se encuentran dentro del lenguaje de bloques de funciones con el fin de ahorrar espacio de memoria en el controlador, estos bloques se guardan solo si se encuentran incluidos en la aplicación.

Sin importar el bloque, cada uno ocupará cierta cantidad de ranuras de memoria, se tiene un total de 76 ranuras disponibles; el software incluye un solo bloque de aplicación, el de PID, que ocupa 5 ranuras de memoria.

El PID se usa para aplicaciones de regulación, aplica tres controles: proporcional, integral y derivativo, se aplica con el fin de mantener un valor medido en el nivel de un valor estipulado; si llegase a existir una diferencia entre ambos valores, el bloque, por medio de un algoritmo matemático, realiza

los cálculos necesarios para aplicar una señal correctiva dentro del proceso para llegar una diferencia nula entre los valores.

Figura 143. **Sistema de PID**



Fuente: elaboración propia.

- Control proporcional: a la salida del sistema se tendrá un valor proporcional a la señal de error que exista en ese momento, la señal de error se multiplica por una contante proporcional y dejará de aplicar correcciones hasta que la señal de error sea cero.

- Control integral: realiza la integral de la señal de error a lo largo de un período de tiempo, hasta que la diferencia se aproxime a cero.

- Control derivativo: analiza la tasa de cambio que tiene la señal de error con respecto al tiempo, corrigiendo errores futuros que pueda tener el sistema.

El bloque de aplicación cuenta con cuatro entradas y seis salidas.

- Entrada de habilitación: habilita la entrada de la función PID.
- Medida: se encarga de medir la entrada, verificando que sea de 16 bits.
- Preselección de consigna: valor de 16 bits que utiliza la función, si y solo si la activación de consigna se encuentra activa.
- Activación de consigna: valida el uso de la preselección de consigna; de lo contrario, usará solo el valor de consigna.
- Salida analógica: se tiene un valor de 0 a 1 023.
- Salida PWM: señal de tipo booleano para PID.
- Kp: constante de proporción, en el rango de 0,1 a 100.
- Ti: tiempo sobre el que se realizará la integral, de 1 s a 900 s.
- Td: tiempo de aplicación derivativa, de 0 s a 60 s.

Figura 144. **Configuración del PID**

Fuente: elaboración propia.

En la ventana de configuración se selecciona si la salida es análoga o de tipo PWM, en el caso de la acción, directa significa que si el valor medido es menor al predeterminado, el PID aumenta, en caso contrario disminuye, en inversa hará las acciones de manera contraria, también es posible asignar valores a los parámetros del PID, incluyendo el tiempo de muestreo de todo el sistema, en un rango de 0,5 a 10 segundos.

El bloque de aplicación trae de forma predeterminada un sistema de PID para aplicaciones de temperatura, presión, nivel y flujo.

Figura 145. **Bloque de aplicación PID**

Fuente: elaboración propia.

5. PRÁCTICAS DE PROGRAMACIÓN PARA PLC

5.1. Práctica #1: secuencia de luces

- Objetivo general

    - Realizar una secuencia repetitiva para un cambio de luces en lenguaje LD.

- Objetivo específico

    - Familiarizarse con el software de programación y sus elementos
    - Realizar circuitos de memoria con los elementos auxiliares
    - Aplicar temporizadores dentro del programa

La práctica consiste en que a partir de un pulsador de encendido, se activará la salida Q1 durante 10 segundos; después de ese tiempo, se activará la salida Q2 por otros 10 segundo, pasando a la salida Q3 que permanecerá activa durante 5 segundos, regresando nuevamente a la salida Q1, quedando la secuencia de manera cíclica; hasta que sea presionado otro pulsador para desactivar toda la secuencia. Asegúrese de revisar la coherencia del programa antes de iniciar la simulación.

5.2. Práctica #2: activación por conteo

- Objetivo general
    - Activar una salida en función de un contador en lenguaje LD
- Objetivo específico
    - Combinar temporizadores para la realización de la práctica
    - Utilizar contadores

El programa debe contar con dos pulsadores, encendido y apagado general respectivamente; un tercer pulsador será el encargado de mandar los pulsos correspondientes al contador; cuando el conteo llegue a 5, la salida Q1 deberá esperar 5 segundos para activarse, permaneciendo activa durante otros 5 segundos, paralelo a Q1 deberá mostrar un texto en la pantalla LCD. El programa debe tener la capacidad de volver a contar sin necesidad de parar y reiniciar la simulación. Asegúrese de revisar la coherencia del programa antes de iniciar la simulación.

5.3. Práctica #3: semáforo doble

- Objetivo general
    - Programar el funcionamiento de dos semáforos en lenguaje LD
- Objetivo específico
    - Combinar diferentes tipos de temporizadores.

- Relacionar contactos de elementos de salida con elementos que forman parte de las entradas del programa.

La activación y desactivación de los semáforos dependerá de dos pulsadores respectivamente; activados los semáforos, iniciará su ciclo de operación hasta que se presione el pulsador de apagado. Un semáforo iniciará en rojo y el otro en verde; los tiempos de operación son:

- Rojo por 10 segundos
- Verde por 10 segundos
- Amarillo por 5 segundos, debe ser intermitente

Asegúrese de revisar la coherencia del programa antes de iniciar la simulación.

5.4. Práctica #4: elevador de tres pisos

- Objetivo general

  - Controlar la operación de un elevador de tres pisos en lenguaje LD.

- Objetivo específico

  - Aplicar los conocimientos adquiridos en las prácticas anteriores
  - Lograr el correcto funcionamiento del elevador
  - Contemplar los posibles fallos y situaciones de operación reales

El elevador no contará con pulsadores de encendido o apagado; al momento de iniciar la simulación, el usuario decidirá en qué piso inicia el elevador.

Todo el sistema contará únicamente con 6 pulsadores: 3 representan el final de carrera que activará el elevador al momento de llegar a un piso, los otros 3 serán para indicar hacia donde se dirige el elevador o para llamar al elevador; es decir, los pulsadores internos y externos se encuentran en paralelo.

La salida Q 1 representa que elevador va hacia arriba y la salida Q 2 que el elevador va hacia abajo.

Debe cumplir con todas las posibles combinaciones y recuérdese utilizar las entradas como pulsadores; asegúrese de revisar la coherencia del programa antes de iniciar la simulación.

5.5. Práctica #5: elevador de tres pisos

- Objetivo general
    - Elaborar el diagrama de operación para un elevador de tres pisos en lenguaje BDF.
- Objetivo específicos
    - Familiarizarse con el entorno de programación BDF.
    - Utilizar de la mejor manera los elementos de entrada para que la simulación se entienda.

Se recomienda realizar un análisis sobre los posibles valores que pueden tomar las entradas así como los valores que se deberían obtener en salida.

Deberá seguir los lineamientos de la práctica #4, tomando en cuenta los siguientes cambios; los finales de carrera no serán representados por pulsadores, serán los finales de carrera que proporciona el software y en el caso de las salidas Q1 y Q2 que representan subir y bajar respectivamente, se cambiarán por un motor cada una.

5.6. Práctica #6: juego de totito

- Objetivo general
    - Recrear la lógica de juego para el totito en lenguaje BDF
- Objetivo específico
    - Aplicar la lógica de compuertas para el correcto funcionamiento del juego.

El juego cuenta con 9 divisiones, en cada una de ellas es posible marcar una equis o un círculo; pero no los dos al mismo tiempo al igual que no es posible cambiar de decisión.

Cada división contará con dos pulsadores para seleccionar equis o círculo; dependiendo de la selección, se activará una luz verde o una roja, respectivamente.

Se podrá ganar en cualquiera de las 8 formas posibles para los dos jugadores; al momento de que un jugador gane se deberá activar su una luz de su color y adicional otra luz indicadora y una señal auditiva.

5.7. Práctica #7: control de parqueo

- Objetivo general

    - Controlar la disponibilidad de parqueos dentro de un establecimiento a partir de la entrada y salida de vehículos en lenguaje BDF.

- Objetivo específico

    - Aplicar el uso de contadores y memorias.
    - Utilizar los elementos para desplegar mensajes en la pantalla LCD.

En el establecimiento habrá dos garitas: entrada y salida; ambas tendrán un sensor de llegada para el vehículo y un pulsador para levantar la talanquera, el total de espacios disponibles es de 10.

Las garitas tendrán una luz roja cuando no haya vehículo y la talanquera permanecerá abajo por medio de un motor; de igual forma, tendrán una luz verde que se activará cuando llegue un vehículo y presione el pulsador, levantando la talanquera con otro motor para permitir el ingreso o salida.

Cada vez que entre o salga un vehículo se aumentará o decrecerá un contador, con el cual, dependiendo de su estado, se mostrará en la pantalla

LCD la cantidad de espacios disponibles o parqueo lleno en el caso de que hayan ingresado 10 vehículos.

5.8. Práctica #8: proceso de mezcla y calentado

- Objetivo general
    - Simular un proceso de dispensado, mezcla y calentado de materiales por medio de tiempos sobre una banda transportadora en lenguaje BDF.

- Objetivo específico
    - Utilizar sensores de proximidad, temporizadores y comparadores

El proceso contará con un inicio de seguridad, presionar dos pulsadores, y un paro de emergencia.

Iniciado el proceso debe haber un sensor de proximidad, esperando a que se coloque el recipiente de forma manual; ya colocado, el motor se activará por 5 segundos, llevando el recipiente a la siguiente etapa, activando otro sensor, el cual activará una válvula por 5 segundos para dispensar el primer material; luego, se activará nuevamente el motor por 5 segundos para llegar a la siguiente etapa; llegando el recipiente se activa otro sensor, que su vez activará otra válvula, por 5 segundos, para dispensar el segundo material y al mismo tiempo se activará un motor y un pistón, ambos por 8 segundos, que simularán el proceso de mezcla; terminando la mezcla el motor se activará por 5 segundos, llegando a otro sensor que iniciará el proceso de calentado por medio de una resistencia con un sensor de temperatura asociado que tendrá

una variación de 0 V a 10 V; en el momento que supere los 8 V, se activará el motor durante 5 segundos, llevando el recipiente a un último sensor que dará por terminado el proceso.

- La solución de cada práctica se encuentra en el siguiente enlace: https://www.dropbox.com/sh/1klvnwew4ku07eq/AAArRNSpZGqPSKmi4efBwg5-a?dl=0.

CONCLUSIONES

1. Se introdujo el tema de automatización para que estudiantes y profesionales tengan el acceso a la información de forma concreta y en un solo documento.

2. Se desarrolló un conjunto de fundamentos para facilitar la comprensión de los temas y para llevar a cabo cada una de las prácticas.

3. Se elaboró una guía para el uso e instalación del software donde se desarrollan las prácticas.

4. Se crearon y desarrollaron prácticas, cada una con su respectiva solución, para que el estudiante aplique los conocimientos adquiridos y desarrolle su lógica de programación.

RECOMENDACIONES

1. Verificar que el software instalado sea ZelioSoft2, versión 5,2, ya que el uso de otra versión, anterior o posterior, no tendrá las mismas herramientas y no será compatible con los archivos.

2. Previo a iniciar con la programación de cada práctica, lea detenidamente el enunciado y realice el diagrama de funcionamiento, tablas de verdad o lógica de operación para facilitar su realización.

3. Revisar continuamente la coherencia de cada programa para evitar problemas futuros o en último caso revisarla antes de iniciar la simulación.

4. Dentro de la simulación es importante abrir las pestañas que muestran los estados de cada elemento en uso dentro de la programación para la detección de fallos y posibles problemas.

5. Para ver el estado de la LCD y sus respectivos textos, se debe minimizar el tamaño de la ventana de simulación en programación escalera y bloques de funciones.

6. Si se quisiera probar alguna de las prácticas de forma real, físicamente, deberá contar con alguno de los modelos que soporta el software; verificar que sea el seleccionado para su programación, también, deberá instalar el cable de comunicación antes de cargar el programa.

BIBLIOGRAFÍA

1. AGUILERA MARTÍNEZ, Patricia. *Programación de PLC´S.* Trabajo de graduación de Maestría en Ciencias de la Ingeniería de Manufactura con Especialidad en Automatización. Universidad Autónoma de Nuevo León, Facultad de Ingeniería, 2002. 84 p.

2. Autracen. *Descubre la estructura interna de un PLC*. [en línea]. <http:// www.autracen.com/descubre-la-estructura-interna-plc/>. [Consulta: 8 de abril de 2019].

3. BALCELLS, Josep, ROMERAL, José Luis. *Autómatas programables.* Barcelona, España: Marcombo, 2007. 439 p.

4. IEC 61131-3. *Estandarización en la programación del control industrial.* [en línea]. <http://isa.uniovi.es/docencia/IngdeAutom/transparencias/iec1131-3%20espa%F1ol.pdf>. [Consulta: 20 de mayo de 2019].

5. Introducción a la mecátronica. *Controladores lógicos programables.* [en línea]. <http://1509720.blogspot.com/2012/11/26-controladores-logicos-programables.html>. [Consulta: 16 de mayo de 2019].

6. Microautomacion. *Automación micromecánica*. [en línea]. <http://www.microautomacion.com/capacitacion/Manual061ControladorLgicoProgramablePLC.pdf>. [Consulta: 8 de abril de 2019].

7. Myelectronic. *Programación de PLC´s: lenguaje en lista de instrucciones.* [en línea]. <http://www.myelectronic.mipropia.com/Aplicaciones/curso%20de%20plc/Lenguaje%20en%20lista%20de%20instrucciones.pdf?i=1>. [Consulta: 21 de mayo de 2019].

8. Rocatek. *Programación ladder PLC básica.* [en línea]. <http://www.rocatek.com/downloads/Programacion%20Ladder.pdf>. [Consulta: 21 de mayo de 2019].

9. ROJAS, David. *Automatización industrial mediante PLCs.* [en línea]. <https://davidrojasticsplc.wordpress.com/2009/01/14/arquitectura-y-apariencia-externa/>. [Consulta: 8 de abril de 2019].

10. Shneider Electric. *ZelioSoft.* [en línea]. <https://www.se.com/gt/es/>. [Consulta: 28 de mayo de 2019].

11. TORRES, Fernando, JARA, Carlos. *Tema 10: interacción y supervisión.* [en línea]. <https://rua.ua.es/dspace/bitstream/10045/18439/1/Tema%2010_Interacci%C3%B3n%20y%20supervisi%C3%B3n.pdf>. [Consulta: 3 de abril de 2019].

12. TORRES, Fernando, JARA, Carlos. *Tema 7: autómatas programables II.* [en línea]. <https://rua.ua.es/dspace/bitstream/10045/18437/1/Tema%207_Automatas%20II.pdf>. [Consulta: 3 de abril de 2019].

13. TORRES, Fernando. *Tema 1: introducción a la automatización y control.* [en línea]. <https://rua.ua.es/dspace/bitstream/10045/18432/1/Tema%201_Introduccion.pdf>. [Consulta: 3 de abril de 2019].

14. UNICEN. *Descripción de lenguajes*. [en línea]. <http://www.exa.unicen.edu.ar/catedras/tldcaut/Software%20Lenguajes%20LD%20y%20IL.pdf>. [Consulta: 21 de mayo de 2019].