



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

**DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA,  
PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300,  
UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL  
PARA EL LABORATORIO DE ELECTRÓNICA 6**

**Freddy Josué López Pérez**

Asesorado por el Ing. Cristian Antonio Orellana López

Guatemala, marzo de 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA,  
PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300,  
UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL  
PARA EL LABORATORIO DE ELECTRÓNICA 6**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**FREDDY JOSUÉ LÓPEZ PÉREZ**

ASESORADO POR EL ING. CRISTIAN ANTONIO ORELLANA LÓPEZ

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN ELECTRÓNICA**

GUATEMALA, MARZO DE 2023

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. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Christian Antonio Orellana López
EXAMINADOR	Ing. Carlos Alberto Navarro Fuentes
EXAMINADORA	Ing. Walter Jacobo Galicia García
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:

**DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA,  
PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300,  
UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL  
PARA EL LABORATORIO DE ELECTRÓNICA 6**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 4 de junio de 2020.

**Freddy Josué López Pérez**

Guatemala 6 de Agosto 2022

Ingeniero  
Julio César 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 " DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA, PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300, UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL PARA EL LABORATORIO DE ELECTRÓNICA 6 ", del señor Freddy Josué López Pérez, por considerar que cumple con los requisitos establecidos.

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

Sin otro particular, me es grato saludarle.

Atentamente,



Ing. Christian Antonio Orellana López

Colegiado 11,939

Asesor.





Guatemala, 6 de agosto de 2022

**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: **DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA, PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300, UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL PARA EL LABORATORIO DE ELECTRÓNICA 6**, desarrollado por el estudiante **Freddy Josué López Pérez**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

**ID Y ENSEÑAD A TODOS**

**Ing. Julio César Solares Peñate**  
**Coordinador de Electrónica**

REF. EIME 79.2022.

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 del estudiante Freddy Josué López Pérez: **“DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA, PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300, UTILIZANDO SIMULADOR PLCSIM EN ENTORNO DE PROGRAMACIÓN TIA PORTAL PARA EL LABORATORIO DE ELECTRÓNICA 6”**, procede a la autorización del mismo.



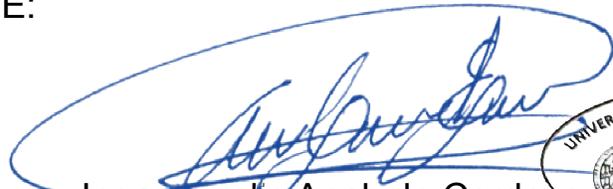
Ing. Armando Alonso Rivera Carrillo

Guatemala, 30 de noviembre de 2022.

LNG.DECANATO.OI.278.2023

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 **DISEÑO DE GUÍA INTRODUCTORIA DE PROGRAMACIÓN EN LENGUAJE DE ESCALERA, PARA EL CONTROLADOR LÓGICO PROGRAMABLE SIEMENS SIMATIC S7-300, UTILIZANDO SIMULADOR PCSIM EN ENTORNO DE PROGRAMACIÓN VIA PORTAL PARA EL LABORATORIO DE ELECTRÓNICA 6**, presentado por: **Freddy Josué López Pérez**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Aurelia Anabela Cordova Lara

Decana

Guatemala, marzo de 2023

AACE/gaoc

## **ACTO QUE DEDICO A:**

- Dios** Por permitirme despertar con salud y vida durante mi tiempo estudiantil, por ayudarme a llegar a este punto y serlo todo en mi vida.
- Mi familia** Para mis padres, Lic. Freddy López y Mgtr Velvet Pérez. Por todo el apoyo, amor, paciencia y cariño que me han dado, sin ellos no estaría acá. Mis hermanos Juan Francisco y Natalia, quienes han sido mis compañeros de aventuras, consejeros y apoyo a lo largo de mi vida. Ellos son los pilares de mi vida, los amo y les agradezco todo.
- Mi abuelita** Doña Aury, por ser un apoyo en todo el proceso, motivación y amor siempre. Parte fundamental de este logro.
- Mi novia** Pamela Calvillo, por el apoyo incondicional en todo momento y creer en mí siempre.
- Mis amigos** A todos en general, por su apoyo y cariño durante todo el proceso, estamos cumpliendo la meta.

## **AGRADECIMIENTOS A:**

### **Mi familia**

Para mis tías, quienes siempre me han dado su cariño y apoyo incondicional. Y mis tíos, quienes me dieron sus consejos para seguir adelante y crecer.

### **Mis primos**

Para todos en general, por las alegrías, aventuras y sobre todo el buen ejemplo y consejos que todos me dieron desinteresadamente.

### **Facultad de Ingeniería**

Por albergarme en sus aulas y darme los conocimientos esenciales en la carrera, y regalarme la oportunidad de conocer a personas increíbles.

### **Ings. Christian y Marlon Orellana**

Por todo el apoyo, paciencia y su valiosa asesoría para la elaboración del presente trabajo de graduación.

### **Centro de Investigaciones de Ingeniería (CII)**

Por permitirme realizar los ensayos en sus instalaciones y por la asesoría prestada por parte de sus colaboradores.



1.3.	Elementos de programación.....	8
1.4.	National Electrical Manufacturers Association (NEMA) .....	9
1.5.	IEC .....	9
1.6.	La norma IEC 1131-3 .....	10
1.7.	Hardware de PLC .....	11
1.8.	Siemens S7-300 .....	12
1.9.	Aspecto físico PLC Siemens S7-300 .....	14
1.10.	Software del PLC S7-300 .....	15
1.11.	Especificaciones mínimas de PC .....	16
2.	PROGRAMACIÓN DE PLC S7-300 EN LENGUAJE DE ESCALERA....	19
2.1.	Definición.....	19
2.2.	Programación simbólica .....	19
2.2.1.	Sintaxis de programación .....	22
2.2.2.	Declaración de variables .....	23
2.3.	Tipos de operaciones .....	24
2.3.1.	Lógica binaria.....	24
2.3.2.	Lógica binaria en lenguaje de escalera .....	25
2.4.	Tipos de datos para PLC.....	26
2.4.1.	Números binarios .....	27
2.4.2.	Números enteros.....	27
2.4.3.	Números en coma flotante .....	28
2.5.	Variables de entrada digital y analógica .....	29
2.5.1.	Digitales.....	29
2.5.2.	Analógicas.....	30
2.6.	Escalado utilizando SCALE_X y NORM_X.....	30
2.6.1.	Variables de salida digitales .....	31
2.6.2.	Marcas.....	32
2.6.3.	Temporizadores .....	33

2.6.4.	Contadores .....	34
3.	INICIAR PROYECTO TIA PORTAL .....	37
4.	UTILIZANDO PLCSIM.....	41
4.1.	Definición .....	41
4.2.	Implementación del proyecto .....	41
4.3.	Uso y visualización .....	43
4.4.	Ejemplo .....	45
4.5.	Ejercicios prácticos en simulador.....	45
	CONCLUSIONES .....	47
	RECOMENDACIONES.....	49
	BIBLIOGRAFÍA.....	51
	APÉNDICES .....	53



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

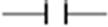
1.	Ejemplo de diagrama de funciones .....	7
2.	Datos técnicos S7-300 .....	14
3.	Aspecto físico Siemens S7-300.....	15
4.	Verificación de cuenta Siemens .....	17
5.	Procedimiento correcto para descarga.....	18
6.	Sintaxis de programación .....	23
7.	Uso de Tia Portal.....	31
8.	Iniciar proyecto paso 1 .....	37
9.	Iniciar proyecto paso 2 .....	38
10.	Iniciar proyecto paso 3 .....	39
11.	Iniciar proyecto paso 4 .....	40
12.	Implementación del proyecto.....	42
13.	Cargar proyecto.....	43
14.	Activar visualización .....	44
15.	Simulación.....	44
16.	Ejemplo de simulación de puesta en marcha .....	45

## TABLAS

I.	Ejemplo de programación .....	4
II.	Contactos típicos de Ladder.....	8
III.	Bobinas típicas de Ladder.....	9
IV.	Simbología .....	19

V.	Simbología Timers .....	20
VI.	Simbología de comparaciones.....	21
VII.	Simbología de operaciones .....	21
VIII.	Operaciones binarias.....	24
IX.	Temporizadores .....	33
X.	Contadores .....	34

## LISTA DE SÍMBOLOS

Símbolo	Significado
	Bobina de salida
°C	Grados Celsius
±	Incerteza
<b>Mb</b>	Megabyte
<b>mm</b>	Milímetro
	Normalmente abierto
	Normalmente cerrado
%	Porcentaje



## GLOSARIO

<b>BIT</b>	Binary digit o dígito binario, es la unidad más pequeña de información y permite representar dos valores, 0 y 1.
<b>BOOL</b>	Tipo de resultado de los operadores de comparación e igualdad.
<b>CPU</b>	Unidad central de procesamiento.
<b>NEMA</b>	National Electrical Manufacturers Association.
<b>PLC</b>	Controlador lógico programable.
<b>PLCSIM</b>	Genera controladores virtuales para simular programación de PLC y comprobar su funcionamiento.
<b>RAM</b>	Memoria de acceso aleatorio.
<b>TIA Portal</b>	Totally Integrated Automation Portal.



## RESUMEN

El trabajo de tesis de diseño de guía introductoria de programación en lenguaje de escalera, para el controlador lógico programable Siemens Simatic S7-300, utilizando simulador PLCSIM en entorno de programación TIA Portal, para el laboratorio de electrónica 6 en la Universidad de San Carlos de Guatemala, se realiza con el fin de crear una guía introductoria para mejorar el conocimiento de los controladores lógicos programables y ayudar a los estudiantes a enfrentar de mejor manera los proyectos y problemas que tendrán al salir a ponerlo en práctica.

En el primer capítulo se conocerán los conceptos básicos referentes a todo lo que se necesita conocer sobre los controladores lógicos programables, conceptos necesarios para poner en práctica en los siguientes capítulos, en los cuales se mostrará específicamente el funcionamiento de los controladores y cómo estos se programan para lograr el programa deseado.

Se cuenta también con ejercicios prácticos para que el estudiante, luego de tener el conocimiento sobre cómo programar los dispositivos, cómo funcionan y para qué lo hacen, pueda poner en práctica y conocer de forma más eficaz cómo estos sistemas pueden solucionar múltiples problemas en la industria, buscando que el estudiante pueda programar estos sistemas incluso si no son de la marca que se está utilizando en este trabajo, ya que los conceptos y bases son aplicables a los distintos controladores que encontrarán en la industria.



## **OBJETIVOS**

### **General**

Diseñar una guía introductoria de programación en lenguaje de escalera para el controlador lógico programable Siemens Simatic S7-300, utilizando simulador PLCSIM en entorno de programación TIA Portal para el curso electrónica 6.

### **Específicos**

1. Proporcionar material didáctico para el desarrollo de prácticas y sistemas basados en controladores lógicos programables.
2. Brindar herramientas a los estudiantes para enfrentarse a los problemas del día a día en la industria.
3. Brindar recursos y apoyo para la programación en lenguaje de contactos para el controlador lógico programable Siemens S7-300.
4. Enseñar cómo utilizar el simulador PLCSIM del software de programación TIA Portal.



## INTRODUCCIÓN

A finales de la década de 1960 la industria buscaba nuevas formas de tecnología electrónica, para optimizar el uso de los sistemas de control basados en circuitos eléctricos con interruptores, botones de emergencia, relés y otros componentes utilizados para el control de un sistema ordinario. Debido a que estos cuentan con partes mecánicas móviles su tiempo de duración era menor, presentando irregularidades luego de cierto tiempo de uso, y necesitaba un mantenimiento muy estricto, por lo que la empresa Bedford Associates (Bedford, MA) propuso un sistema al que llamó MODICON, este fue el primer PLC vendido comercialmente. La National Electrical Manufacturers Association (NEMA) tiene la siguiente definición (parafraseada) para un controlador lógico programable (PLC):

Es un dispositivo electrónico digital, que utiliza una memoria para almacenar instrucciones e implementar funciones específicas de lógica, secuencial, temporizado, conteo y aritmética, para controlar en tiempo real máquinas y procesos. Los PLC's podrían trabajar alejados de la maquinaria que controlaban, pero, debido a que no contaban con un estándar, la comunicación entre los dispositivos se tornó difícil, por lo que en los años 90 se realizó el estándar IEC 1131-3 que combina todos los lenguajes de programación de un PLC en un solo estándar internacional.

La programación de este dispositivo se puede realizar en lenguaje Ladder (escalera), también conocido como diagrama de contactos. Es un lenguaje de programación gráfico que está basado en los esquemas eléctricos de un control clásico.



# 1. MARCO TEÓRICO

## 1.1. Controles lógicos programables

Los controladores lógicos programables son una pieza fundamental en la automatización, ya que con estos se controlan los procesos industriales de forma automática y personalizada, permitiendo eficientar de gran manera la forma en que se desarrollan los procesos, disminuyendo los costos de producción y aumentando la velocidad de generación de productos

### 1.1.1. Definición

De acuerdo con la definición de la National Electrical Manufacturers Association (NEMA) un PLC es:

Un aparato electrónico operado digitalmente, que usa una memoria programable para el almacenamiento interno de instrucciones para implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas para controlar, a través de módulos de entrada/salida digitales (*ON/OFF*) o analógicos (1-5 VDC, 4-20 mA, etc.), varios tipos de máquinas o procesos.<sup>1</sup>

Un controlador lógico programable, más conocido por sus siglas en inglés PLC (Programmable Logic Controller), se trata de una computadora utilizada en la automatización industrial para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas. Sin embargo, la definición más precisa de estos dispositivos es la dada por la NEMA ya citada en el párrafo anterior.

---

<sup>1</sup> National Electrical Manufacturers Association. *Definición de PLC.* s/p.

## **1.1.2. Tipos y características**

Los tipos y características se pueden clasificar en 6 tipos:

### **1.1.2.1. PLC´s compactos**

Se caracterizan por tener un CPU, entradas y salidas, fuentes de alimentación en un solo paquete, cuentan con un número limitado y fijo de entradas y salidas y uno o dos canales de comunicación para la conexión del PLC, también cuentan con entradas y salidas analógicas, la cantidad de estas a menudo es menor a la cantidad de las salidas digitales.

### **1.1.2.2. PLC´s modulares**

En esta categoría de PLC´s se encuentran los más potentes, estos cuentan con más funciones que los PLC´s compactos, soportan un número elevado de entradas y salidas, tanto digitales como analógicas, soportan programas más grandes y pueden operar tareas distintas simultáneamente. En esta categoría de PLC´s se encuentra el Siemens S7-300.

### **1.1.2.3. PLC´s montados en *rack***

Esta categoría tiene prácticamente las mismas funciones y capacidades que un PLC modular, se diferencia en el *rack* donde se colocan los módulos, por medio de las ranuras que tienen sirven como bus de conexión entre los distintos módulos.

#### **1.1.2.4. PLC con panel Operador y Controlador Lógico Programable (OPLC)**

Cuenta con una interfaz que facilita y optimiza su funcionamiento, permite la observación y supervisión de los procesos que se realizan en tiempo real en las diferentes máquinas de operación, presenta pantallas *touch* para la visualización e interacción con las mismas o una pantalla LCD y teclas físicas.

#### **1.1.2.5. Ordenador industrial (PC industrial)**

Es una tarjeta que cuenta con todas las funciones de cualquier CPU de un PLC normal, permite cambiar directamente la información entre aplicaciones del software, y cuenta por lo menos con un canal de comunicación para conectar entradas y salidas con otros PLC's.

#### **1.1.2.6. PLC tipo software**

Se trata de los PLC's más modernos, ya que trata en si de un PLC virtual, que está diseñado para que pueda ser adaptado a cualquier ordenador personal para controlar, el gran problema con esta categoría es la falta de memoria y la pérdida de datos al interrumpirse el suministro eléctrico.

### **1.2. Lenguajes de programación**

Para la programación de los controladores lógicos programables se cuenta con distintos lenguajes de programación con la misma funcionalidad, diferenciándose principalmente en la forma de programación visual y la facilidad de aprendizaje de cada lenguaje.

### 1.2.1. Lenguajes textuales

Los lenguajes textuales se representan con letras, se realiza la programación por medio de cadenas textuales que crean las condiciones y parámetros de programación para que la funcionalidad sea correcta.

#### 1.2.1.1. Lista de instrucciones

Utiliza la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones o nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control. El lenguaje Lista de Instrucciones es una forma de lenguaje Booleano.<sup>2</sup>

Tabla I. Ejemplo de programación

A	I	2.3
A	I	4.1
O	I	3.2
=	Q	1.6

Fuente: elaboración propia.

#### 1.2.1.2. Texto estructurado

Texto estructurado (ST) es un lenguaje de alto nivel que permite la programación estructurada, lo cual significa que muchas tareas complejas pueden ser divididas en unidades más pequeñas. ST se parece mucho a los lenguajes de computadoras BASIC o PASCAL, que usan subrutinas para llevar

---

<sup>2</sup> YUGSI, R. *Lenguajes de programación de PLC'S*.  
<http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>. Consulta: 18 de agosto de 2019.

a cabo diferentes partes de las funciones de control y paso de parámetros y valores entre las diferentes secciones del programa.

Al igual que LD, FBD e IL, el lenguaje de texto estructurado utiliza la definición de variables para identificar entradas y salidas de dispositivos de campo y cualquier otra variable creada internamente.

Incluye estructuras de cálculo repetitivo y condicional, tales como: FOR... TO; REPEAT... UNTIL X; WHILE X....; IF... THEN... ELSE. Además, soporta operaciones Booleanas (AND, OR, etc.) y una variedad de datos específicos, tales como fecha y hora.

La programación en Texto Estructurado es apropiada para aplicaciones que involucran manipulación de datos, ordenamiento computacional y aplicaciones matemáticas que utilizan valores de punto flotante. ST es el mejor lenguaje para la implementación de aplicaciones de inteligencia artificial, lógica difusa, toma de decisiones, etc.<sup>3</sup>

Ejemplo:

```
IF Manual AND Alarm THEN
Level = Manual Level;
Mixer = Start AND NOT Reset
ELSE IF      Other Mode THEN
      Level = Max_level;
ELSE      Level = (Level_Indic X100)/Scale;
END IF;
```

---

<sup>3</sup> YUGSI, R. *Lenguajes de programación de PLC'S.*  
<http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>. Consulta: 18 de agosto de 2019.

## 1.2.2. Lenguajes gráficos

Los lenguajes gráficos utilizan principalmente imágenes, signos o esquemas para la estructuración del programa, son lenguajes de programación más visuales, por lo que la programación suele ser más sencilla o lógica

### 1.2.2.1. Diagrama de bloques

Es un lenguaje gráfico que permite al usuario programar elementos (bloque de funciones del PLC), de tal forma que ellos aparecen interconectados al igual que un circuito eléctrico. Generalmente utilizan símbolos lógicos para representar al bloque de función. Las salidas lógicas no requieren incorporar una bobina de salida, porque la salida es representada por una variable asignada a la salida del bloque.

El diagrama de funciones lógicas resulta especialmente cómodo de utilizar, para técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente.

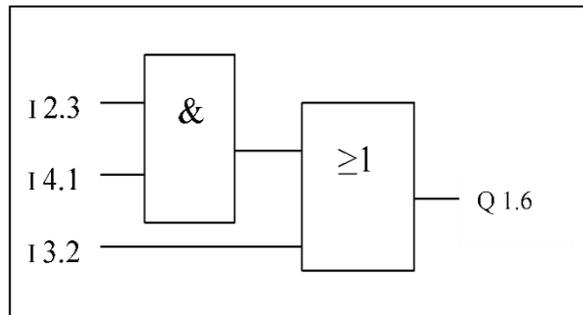
“Adicionalmente a las funciones lógicas estándares y específicas del vendedor, el lenguaje FBD de la Norma IEC 1131-3 permite al usuario construir sus propios bloques de funciones, de acuerdo a los requerimientos del programa de control”<sup>4</sup>.

Ejemplo de programación mediante diagrama de funciones:

---

<sup>4</sup> YUGSI, R. *Lenguajes de programación de PLC'S.*  
<http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>. Consulta: 18 de agosto de 2019.

Figura 1. **Ejemplo de diagrama de funciones**



Fuente: YUGSI, R. *Lenguajes de programación de PLC'S*.  
<http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>. Consulta: 18 de agosto de 2019.

### **1.2.2.2. Lenguaje de contactos**

También conocido como lenguaje de escalera, es el más utilizado a nivel americano, ya que la gran mayoría de los PLC's de fabricación americana o japonesa permiten su programación en el lenguaje de contactos. Es muy fácil adaptarse a la programación en este tipo de lenguaje, son necesarios conocimientos eléctricos mínimos. Sus símbolos básicos están normalizados por la norma NEMA y brindan la gran ventaja que todos los fabricantes de PLC's emplean estas normas en sus productos, por lo que, al tener conocimientos de la programación en lenguajes de contactos, se adquiere el conocimiento para programar cualquiera de los distintos PLC's que están regulados por las normas NEMA.

Después de conocer los símbolos de programación, es necesario conocer cómo realizar la estructura de un programa y el orden de ejecución de lo programado. Se colocan los contactos a la derecha y las bobinas de salida se

colocan a la derecha, de la misma forma es posible conectar contactos en paralelo y bobinas en paralelo, dependiendo de las necesidades del programa.

### 1.3. Elementos de programación

En la programación de los controladores se utilizan distintos elementos que cuentan con funcionalidades distintas, cada uno de estos elementos representa una función específica para el programa y demuestra el estado de las variables.

Tabla II. **Contactos típicos de Ladder**

Tipo de contacto	Visualización	Permite cuando...
Normalmente abierto	—   —	Referencia On
Normalmente Cerrado	— / —	Referencia <i>OFF</i>

Fuente: elaboración propia.

Tabla III. **Bobinas típicas de Ladder**

Tipo de Bobina	Visualización	¿Corriente?	Resultado
Normalmente abierta	—( )—	Sí	Referencia <i>ON</i>
		No	Referencia <i>OFF</i>
<i>Set</i>	—(S)—	Sí	Referencia <i>ON</i> hasta que se apague con “R”
		No	Sigue Igual
<i>Reset</i>	—(R)—	Sí	Referencia <i>OFF</i> hasta que se active con “S”
		No	Sigue Igual

Fuente: elaboración propia.

#### 1.4. **National Electrical Manufacturers Association (NEMA)**

La Asociación Nacional de Fabricantes Eléctricos es una asociación industrial de Estados Unidos, con sede en Virginia, responsable de numerosos estándares industriales comunes usados en el campo de la electricidad. Entre otros, la NEMA ha establecido una amplia gama de estándares para encapsulados de equipamientos eléctricos.

#### 1.5. **IEC**

La Comisión Electrotécnica Internacional (IEC) es la principal organización del mundo que prepara y publica estándares internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas. Fundada en 1906, la IEC es la organización líder en el mundo para la elaboración y publicación de las normas internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas. Estas se conocen colectivamente como electrotécnica. Cuenta con expertos de

la industria, el comercio, el gobierno, la prueba y los laboratorios de investigación, las universidades y los grupos de consumidores que participan en el trabajo de normalización IEC.

IEC proporciona una plataforma para las empresas, las industrias y los gobiernos para hacer frente a la discusión y el desarrollo de las normas internacionales que se requieren.

Millones de dispositivos que contienen electrónica, y usan o producen electricidad, se basan en normas internacionales de la IEC y en los sistemas de evaluación de la conformidad para trabajar de manera segura.

#### **1.6. La norma IEC 1131-3**

La IEC desarrolló el estándar IEC 1131, en un esfuerzo para estandarizar los controladores programables. Uno de los objetivos del comité fue crear un conjunto común de instrucciones que podría ser usado en todos los PLC's. Aunque el estándar 1131 alcanzó el estado de estándar internacional en agosto de 1992, el esfuerzo para crear un PLC estándar global ha sido una tarea muy difícil, debido a la diversidad de fabricantes de PLC's y a los problemas de incompatibilidad de programas entre marcas de estos.

El estándar IEC 1131 para controladores programables consiste de cinco partes, una de las cuales hace referencia a los lenguajes de programación y es referida como la IEC 1131-3.

El estándar IEC 1131-3 define dos lenguajes gráficos y dos lenguajes basados en texto, para la programación de PLC's. Los lenguajes gráficos utilizan

símbolos para programar las instrucciones de control, mientras que los lenguajes basados en texto usan cadenas de caracteres para programar las instrucciones.

Adicionalmente, el estándar IEC 1131-3 incluye una forma de programación orientada a objetos llamada Secuencial Function Chart (SFC). SFC es a menudo categorizado como un lenguaje IEC 1131-3, pero éste es realmente una estructura organizacional que coordina los cuatro lenguajes estándares de programación (LD, FBD, IL y ST). La estructura del SFC tuvo sus raíces en el primer estándar francés de Grafcet (IEC 848).<sup>5</sup>

## 1.7. Hardware de PLC

El hardware de un PLC es unidad de procesamiento (CPU), memoria, entradas y salidas, fuente de alimentación y dispositivo de programación.

- CPU: basado en microprocesador, permite operaciones aritméticas, operaciones lógicas, movimientos de memoria, interfaces de computadora, redes de áreas locales y funciones.
- Memoria ROM-Sistema: permite el guardado permanente del sistema operatorio de los datos compuestos usados por el CPU.
- Memoria RAM: utilizada para la data, es el lugar en el que la información es almacenada en el estado de dispositivos de entradas y salidas, se almacenan también los valores de los *timers* y contadores.
- Entradas y salidas: entradas-monitoreo de dispositivos de campo como los interruptores y sensores.
- Salidas-control de otros dispositivos como motores, bombas, luces, entre otros.

---

<sup>5</sup> YUGSI, R. *Lenguajes de programación de PLC'S*. <http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>. Consulta: 18 de agosto de 2019.

- Fuente de alimentación: la mayoría de los PLC's trabajan con 12 VDC o 220VAC, algunos utilizan la fuente de alimentación como un módulo separado.
- Dispositivo de programación: es utilizado para ingresar el programa requerido en la memoria del procesador, el programa se desarrolla en el dispositivo de programación y es transferido a la unidad de memoria del PLC.

### **1.8. Siemens S7-300**

El Simatic S7-300 está concebido para soluciones de sistema innovadoras con especial énfasis en tecnología de fabricación y, como sistema de automatización universal, constituye una solución óptima para aplicaciones en estructuras centralizadas y descentralizadas.

- Potentes módulos centrales con interfaz industrial Ethernet / PROFINET, funciones tecnológicas integradas o versión de seguridad en un sistema coherente evitan inversiones adicionales.
- El S7-300 se puede configurar de forma modular, no hay ninguna regla de asignación de *slots* para los módulos periféricos. Hay disponible una amplia gama de módulos, tanto para estructuras centralizadas como para estructuras descentralizadas con ET-200M.
- El uso de la Micro Memory Card como memoria de datos y programa hace innecesaria una pila tampón y ahorra costos de mantenimiento. Además, en esta tarjeta de memoria se puede guardar un proyecto asociado con símbolos y comentarios para simplificar el trabajo del servicio técnico.

- Asimismo, la Micro Memory Card permite la actualización sencilla del programa o del *firmware* sin programadora. Además, se puede utilizar durante el funcionamiento para guardar y consultar datos, por ejemplo, para archivar medidas o para procesar recetas.
- Además de la automatización estándar, en un S7-300 también se pueden integrar funciones de seguridad y control de movimiento.
- Muchos de los componentes S7-300 también están disponibles en una versión SIPLUS para condiciones ambientales extremas como, por ejemplo, rango de temperatura ampliado (de -40/25 a -60/70°C) y utilización en atmósfera agresiva/condensación.

Figura 2. **Datos técnicos S7-300**

<b>Datos técnicos generales S7-300, S7-300F</b>	
Grado de protección	IP20 según IEC 60 529
Temperatura ambiente	
• con montaje horizontal	0 ... 60 °C
• con montaje vertical	0 ... 40 °C
Humedad relativa	5 ... 95 %, sin condensación (grado de sollicitación RH 2 según IEC 61131-2)
Presión atmosférica	795 ... 1080 hPa
Aislamiento	
• circuitos 24 V DC	Tensión de ensayo 500 V DC
• circuitos 230 V AC	Tensión de ensayo 1460 V AC
Compatibilidad electromagnética CEM	Requerimientos de la Directiva CEM  Inmunidad a las descargas electrostáticas conforme a IEC 61000-6-2, ensayo según: IEC 61000-4-2, 61000-4-3, IEC 61000-4-4, IEC 61000-4-5, IEC 61000-4-6  Emisión de perturbaciones según EN 50081-2, comprobación según EN 55011, clase A, grupo 1
Cargas mecánicas	
• Vibraciones, ensayo según / ensayadas con	IEC 60068, parte 2-6/10 ... 58 Hz; amplitud constante 0,075 mm; 58 ... 150 Hz; aceleración constante 1 g; duración de la vibración: 10 barridos de frecuencia por eje en cada sentido de los tres ejes perpendiculares entre sí
• Choques ensayados según/con	IEC 60068, Parte 2-27/semisinusoidal: intensidad del choque 15 g (valor cresta), duración 11 ms

Fuente: Siemens. *Productos para Totally Integrated Automation y Micro Automation Catálogo ST 70 • 2009*. [http://www.grupdap.es/descargas-industrias/01-catalogos/01-simatic-s57/SIMATIC\\_ST70\\_2009\\_SP.pdf](http://www.grupdap.es/descargas-industrias/01-catalogos/01-simatic-s57/SIMATIC_ST70_2009_SP.pdf)

### 1.9. Aspecto físico del PLC Siemens S7-300

Es necesario conocer el aspecto físico del controlador lógico programable para saber la posición de sus puntos de salidas y entradas, además de conocer el espacio necesario para la instalación del mismo.

Figura 3. **Aspecto físico del Siemens S7-300**



Fuente: elaboración propia.

### **1.10. Software del PLC s7-300**

Existen actualmente una gran cantidad de softwares dedicados a la programación de controladores lógicos programables, los fabricantes más importantes a nivel mundial cuentan con sus propios softwares de programación, para los cuales se necesita una licencia de uso que se adquiere al comprar sus productos o puede ser comprada en otro momento.

Se utilizará el software para controladores lógicos programables de Siemens TIA Portal versión 13.

### 1.11. Especificaciones mínimas de PC

- Procesador: Core™ i5-3320M 3.3 GHz o similar
- Memoria principal: 8 GB de memoria (recomendado) o más
- Disco duro: 300 GB SSD
- Gráficos: Mín. 1920 x 1080
- Pantalla: 15,6" *display* de pantalla ancha (1920 x 1080)
- Instalación en Windows:
  - Windows 7 (64 bits)
  - Windows 7 Home Premium SP1 \*
  - Windows 7 Professional SP1
  - Windows 7 Enterprise SP1
  - Windows 7 Ultimate SP1
  - Windows 10 (64 bits)
  - Windows 10 Home Version 1709, 1803 \*
  - Windows 10 Professional Version 1709, 1803
  - Windows 10 Enterprise Version 1709, 1803
  - Windows 10 Enterprise 2016 LTSC
  - Windows 10 IoT Enterprise 2015 LTSC
  - Windows 10 IoT Enterprise 2016 LTSC

Sin embargo, con especificaciones menores se podrá utilizar, pero la respuesta será un poco más lenta. Para fines de aprendizaje no se tendrá problema con computadoras un poco menos potentes.

- Procedimiento de obtención de permiso para instalación de TIA Portal v13

En caso de no tener la licencia oficial de TIA portal, se puede crear una cuenta en la página oficial de Siemens para descargar SIMATIC STEP 7 (TIA Portal) V13 SP1 TRIAL versión 21 días, suficiente para realizar los ejercicios y aprender. Se puede descargar del siguiente enlace: [https://support.industry.siemens.com/cs/document/106448872/simatic-step-7-\(tia-portal\)-v13-sp1-trial-download?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/106448872/simatic-step-7-(tia-portal)-v13-sp1-trial-download?dti=0&lc=en-WW).

- Se realiza la verificación de la cuenta en la que es necesario colocar los datos de la persona, para obtener la autorización de descargar el software. Se recibe la respuesta en 3 días hábiles aproximadamente.

Figura 4. Verificación de cuenta Siemens



The image shows a web form titled "Datos de acceso" for account verification. It contains several input fields and a "Guardar" button. The fields are: Salutación (radio buttons for Sr. and Sra.), Apellido (Lopez), Nombre (Freddy), Login (freddyjosue27), E-Mail (freddylopez@gmail.com), Su nombre en los foros (freddy2712), Teléfono (30345102), Empresa (Universidad de San Carlos de Guatemala), Dirección (4 calle 0-44 zona 8 de mixco, Condomi), CP (01057), Ciudad (Guatemala), País / Región (Guatemala), and Idioma (Spanish). A checkbox for "Descarga de software con limitación de exportación" is checked. A note at the bottom states "\* campo obligatorio".

Datos de acceso	
Salutación *	<input checked="" type="radio"/> Sr. <input type="radio"/> Sra.
Apellido *	<input type="text" value="Lopez"/>
Nombre *	<input type="text" value="Freddy"/>
Login *	<input type="text" value="freddyjosue27"/>
E-Mail *	<input type="text" value="freddylopez@gmail.com"/>
Su nombre en los foros *	<input type="text" value="freddy2712"/>
Teléfono *	<input type="text" value="30345102"/>
Empresa *	<input type="text" value="Universidad de San Carlos de Guatemala"/>
Dirección *	<input type="text" value="4 calle 0-44 zona 8 de mixco, Condomi"/>
CP *	<input type="text" value="01057"/>
Ciudad *	<input type="text" value="Guatemala"/>
País / Región *	<input type="text" value="Guatemala"/>
Idioma *	<input type="text" value="Spanish"/>
<input checked="" type="checkbox"/> Descarga de software con limitación de exportación	
* campo obligatorio	
<input type="button" value="Guardar"/>	

Fuente: elaboración propia.

- Se obtiene respuesta satisfactoria y se procede a descargar el software. Son 10 archivos de 650Mb, 1 archivo de 330 Mb y otro de 2.7 Mb.

Figura 5. **Procedimiento correcto para descarga**

Export identification code ECCN	N
Are download recipient and final recipient identical?	<input checked="" type="radio"/> yes <input type="radio"/> No
<b>Download recipient</b>	
Name	Lopez
First name	Freddy
Company	Universidad de San Carlos de Guatemala
E-mail	freddylopez@gmail.com
City	Guatemala
Country	Guatemala
<b>Final recipient</b>	
Name	Lopez
First name	Freddy
Company	Universidad de San Carlos de Guatemala
Department	Guatemala
Street	4 calle 0-44 zona 8 de mixco, Condominio Santa Barbara.
ZIP code	01057
City	Guatemala
Country	Guatemala
Check date	10/30/2019 4:44:50 PM
Check result	
Export authorization exists	
<input type="button" value="Print"/> <input type="button" value="Continue to download"/>	

Fuente: elaboración propia.

- Descargar PLCsim de la misma página web, es un archivo de 2.5Mb, otro de 665 Mb y el último de 616Mb.

## 2. PROGRAMACIÓN DE PLC S7-300 EN LENGUAJE DE ESCALERA

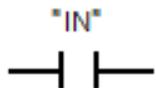
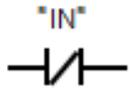
### 2.1. Definición

El lenguaje de escalera, conocido también como lenguaje de contactos, es el tipo de programación más popular y sencillo para la programación de controladores lógicos programables en el mercado actual, está basado en los esquemas eléctricos básicos conocidos, por lo que su adaptabilidad y comprensión del lenguaje son muy sencillas si se tienen los conocimientos básicos de electricidad. La fuente de energía es representada por 2 líneas verticales, la línea vertical izquierda representa el voltaje positivo y la línea vertical derecha representa tierra.

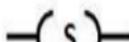
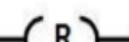
### 2.2. Programación simbólica

La programación en lenguaje de escalera es realizada totalmente por símbolos, los cuales representan distintas acciones que pueden realizar:

Tabla IV. Simbología

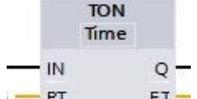
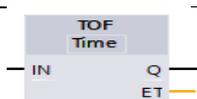
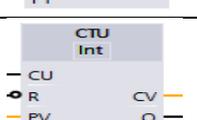
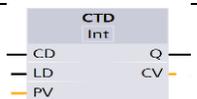
<b>Contacto abierto:</b> se activa y permite el paso cuando el bit de entrada es 1, para desactivarlo es necesario enviar un bit con valor 0.	
<b>Contacto cerrado:</b> se mantiene activado y permite el paso cuando el bit de entrada es 0, para interrumpir el paso se debe enviar bit con valor 1.	

Continuación de la tabla IV.

<p><b>Salida (bobina):</b> elemento de salida final, simboliza la salida del PLC, se activa cuando se cumplen las condiciones colocadas a su izquierda.</p>	
<p><b>Salida set:</b> al activarse asigna valor a bit de 1 hasta que este es reseteado.</p>	
<p><b>Salida reset:</b> al activarse se encarga de resetear la salida set y permanece con valor 0 hasta que es activada de nuevo.</p>	

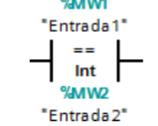
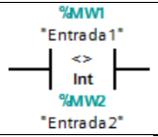
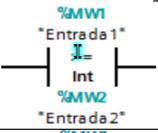
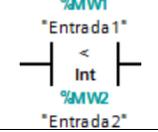
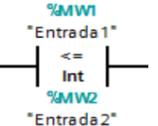
Fuente: elaboración propia.

Tabla V. **Simbología Timers**

<p><b>TON:</b> retardo a conectar, al activarse espera el tiempo ya establecido para que conecte.</p>	
<p><b>TOFF:</b> retardo a desconectado, al activarse espera el tiempo ya establecido para desconectar.</p>	
<p><b>CTU:</b> cuenta adelante a partir del valor actual al activarse, se establece un límite para que, al llegar a este límite, se active el bit del contador.</p>	
<p><b>CTD:</b> cuenta hacia atrás a partir del valor determinado, al llegar a cero el contador se activa el bit de este.</p>	

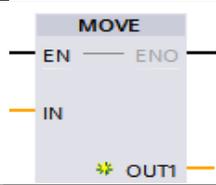
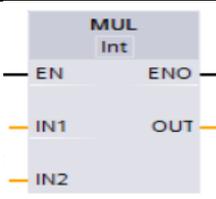
Fuente: elaboración propia.

Tabla VI. **Simbología de comparaciones**

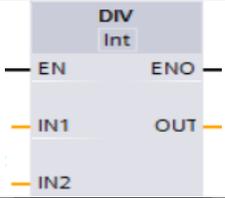
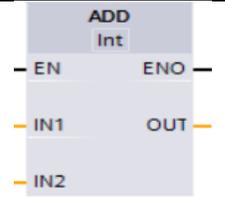
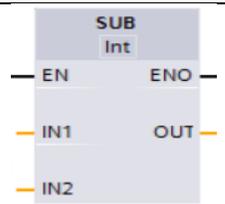
<p><b>Igual a:</b> se activa bit de salida si la entrada 1 es igual a la entrada.</p>	
<p><b>Diferente a:</b> se activa bit de salida si la entrada 1 es distinta a la entrada.</p>	
<p><b>Mayor que:</b> se activa bit de salida si la entrada 1 es mayor que la entrada 2.</p>	
<p><b>Mayor igual que:</b> se activa bit de salida si la entrada 1 es mayor o igual que la entrada 2.</p>	
<p><b>Menor que:</b> se activa bit de salida si la entrada 1 es menor que la entrada 2.</p>	
<p><b>Menor igual que:</b> se activa bit de salida si la entrada 1 es menor o igual que la entrada 2.</p>	

Fuente: elaboración propia.

Tabla VII. **Simbología de operaciones**

<p><b>MOVE:</b> envía una copia de un elemento guardado en una dirección a una dirección distinta.</p>	
<p><b>MUL:</b> multiplica el valor IN1 con el valor IN2 y se obtiene el resultado en OUT.</p>	

Continuación de la tabla VII.

<p><b>DIV:</b> divide el valor IN1 por el valor IN2 y se obtiene el resultado en OUT.</p>	
<p><b>ADD:</b> suma el valor IN1 con el valor IN2 y permite consultar el resultado en OUT.</p>	
<p><b>SUB:</b> resta el valor IN2 del valor IN1 y se obtiene el valor de la respuesta en OUT.</p>	

Fuente: elaboración propia.

### 2.2.1. Sintaxis de programación

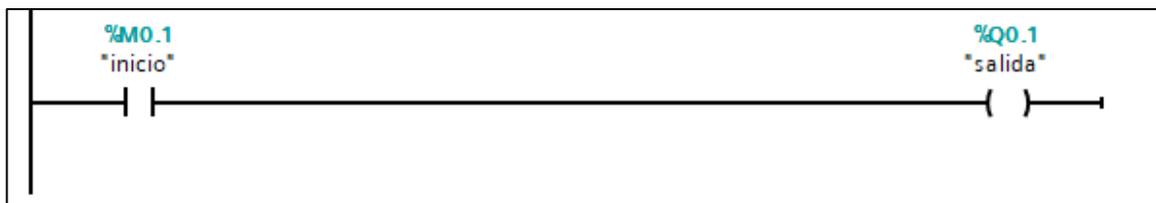
El lenguaje de escalera es el mismo para todos los distintos controladores lógicos programables, solo se deben tomar en cuenta la cantidad de entradas y salidas que tienen los diferentes PLC, ya que no todos cuentan con el mismo número de estas, se puede realizar la programación en una aplicación industrial o una aplicación de prueba con un simulador como es en este caso, en el cual el entorno de programación es exactamente el mismo al ser utilizado en la industria, con la salvedad de que no es necesario contar con un PLC físico y se puede utilizar un simulador para conocer si la programación es correcta.

La sintaxis de la programación en escalera no es más que programar por medio de cumplimiento de condiciones. Al cumplirse ciertas condiciones ya

establecidas estas producen un cambio de variable en la salida. Los contactos normalmente abiertos y normalmente cerrados realizan una comparación de las condiciones y se obtiene el resultado de la evaluación, la cual es enviada a las bobinas de salidas que examinan el resultado de la evaluación hecha por los contactos, y si es verdadera realizan una acción.

En el lado izquierdo de la pantalla, al programar, se encuentran las instrucciones de entrada, y en la parte derecha las salidas y funciones, básicamente la programación en escalera es procesar las señales de entrada y el resultado se observa en las variables de salida.

Figura 6. **Sintaxis de programación**



Fuente: elaboración propia.

### 2.2.2. Declaración de variables

- Paso 1: en pestaña de PLC, dar clic en pestaña variables de PLC y luego clic en tabla de variables estándar.
- Paso 2: asignar el nombre deseado a la variable.
- Paso 3: asignar el tipo de dato que será esa variable, *bool*, *int*, *Word*, entre otros.

- Paso 4: asignar dirección de la variable, esto representa la dirección física que tendrá la variable o el valor de marca asignada, se debe evitar el traslape entre estas.

## 2.3. Tipos de operaciones

Es necesario realizar operaciones matemáticas dentro de la programación de los controladores para el manejo de variables internas y externas, estas operaciones ayudan a utilizar las variables para convertir los datos que se tienen en datos útiles para el programa que genere acciones dependientes del valor que se obtenga.

### 2.3.1. Lógica binaria

La lógica binaria es una de las operaciones más básicas con las que trabaja el controlador, con esta lógica se pueden realizar múltiples operaciones de variables para el uso del programa. La lógica binaria basa su funcionamiento en unos y ceros para las operaciones matemáticas, que es el lenguaje universalmente conocido por las computadoras.

Tabla VIII. Operaciones binarias

<p><b>Compuerta and:</b> esta función obtiene únicamente una salida alta cuando todos los valores de la función sean altos, con un valor de la función que sea bajo automáticamente la salida es baja, la función and de forma sencilla se comporta como un multiplicador porque la salida es el producto de sus entradas. Para comprender de mejor manera esta función se utiliza la siguiente tabla de verdad:</p>	<b>INPUT</b>		<b>OUTPUT</b>
	A	B	A AND B
	0	0	0
	0	1	0
	1	0	0
1	1	1	

Continuación de la tabla VIII.

<p><b>Compuerta or:</b> esta función obtiene salida alta si por lo menos una de esas entradas es alta, la única forma en que su salida puede ser baja es si todas las entradas son bajas, si todas las entradas son altas la salida también será alta. Para comprenderlo de mejor manera se utiliza la siguiente tabla de verdad:</p>	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>A OR B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	A OR B	0	0	0	0	1	1	1	0	1	1	1	1
INPUT		OUTPUT																	
A	B	A OR B																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
<p><b>Compuerta not:</b> también conocida como inversor, esta función implementa una negación lógica, si la entrada es alta su salida será baja y si su entrada es baja su salida será alta. Para comprenderlo de mejor manera se utiliza la siguiente tabla de verdad</p>	<table border="1"> <thead> <tr> <th>INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>NOT A</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT	OUTPUT	A	NOT A	0	1	1	0										
INPUT	OUTPUT																		
A	NOT A																		
0	1																		
1	0																		
<p><b>Compuerta xor:</b> esta presenta una salida alta si únicamente una de todas las entradas es alta, si ambas entradas son altas o ambas entradas son bajas la salida será baja, en palabras más sencillas la compuerta xor es “uno o el otro, pero no ambos”. Para comprenderlo de mejor manera se utiliza la siguiente tabla de verdad:</p>	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>A XOR B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	A XOR B	0	0	0	0	1	1	1	0	1	1	1	0
INPUT		OUTPUT																	
A	B	A XOR B																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
<p><b>Compuerta xnor:</b> su función es inversa a la compuerta XOR, la salida es alta únicamente cuando las dos entradas son iguales, para obtener la salida en alto todas las entradas deben ser altas o deben ser bajas, si no todas las entradas son del mismo valor la salida será baja. Para comprenderlo de mejor manera se utiliza la siguiente tabla de verdad:</p>	<table border="1"> <thead> <tr> <th colspan="2">INPUT</th> <th>OUTPUT</th> </tr> <tr> <th>A</th> <th>B</th> <th>A XNOR B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	INPUT		OUTPUT	A	B	A XNOR B	0	0	1	0	1	0	1	0	0	1	1	1
INPUT		OUTPUT																	
A	B	A XNOR B																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

Fuente: elaboración propia.

### 2.3.2. Lógica binaria en lenguaje de escalera

- AND: dos o más contactos en serie, para tener una salida alta, todos los contactos normalmente abiertos deben estar en *on*.
- NAND: dos o más contactos en serie, para tener una salida alta, todos los contactos normalmente cerrados deben estar en *off*.

- OR: dos o más contactos en paralelo, para tener una salida alta, cualquiera de todos los contactos normalmente abiertos debe estar *on* y permitir el paso de la corriente, con solo un contacto que esté en la salida será alta.
- NOR: dos o más contactos en paralelo, para tener una salida alta, cualquiera de todos los contactos normalmente cerrados debe estar *off* y permitir el paso de la corriente, con solo un contacto que esté *off* la salida será alta.
- NOT: la salida será inversa a la entrada, si la entrada es alta, la salida será baja, si la entrada es baja la salida será alta.
- XOR: contacto normalmente cerrado en serie con contacto normalmente abierto, en paralelo con otro arreglo igual, la salida será alta si cualquiera de las entradas está *on*, pero no si las dos entradas están *on*.
- XNOR: dos contactos normalmente cerrados en serie, en paralelo con dos contactos normalmente abiertos en serie, la salida será alta si, ya sea si las entradas están en *on* o si las dos entradas están en *off*, una entrada esta *off* y la otra entrada esta *on*, así la salida es baja.

#### **2.4. Tipos de datos para PLC**

Se utilizan distintos tipos de datos para el manejo y operaciones de variables, se utilizan principalmente 3 tipos de datos con los que se controlan todos los datos obtenidos.

### 2.4.1. Números binarios

- BOOL (BIT) BOOLEANO: su dirección TIA Portal es (M0.1...).
  - Representa un valor de bit y contiene los valores 1 (alto) y 2 (bajo).
- BYTE: su dirección en TIA Portal es (MB10....).
  - Es una secuencia de 8 bits.
  - Enteros con signo: -128 a 127, enteros sin signo: 0 a 255.
- WORD: dirección en TIA Portal (MW10...).
  - Es una secuencia de 16 bits.
  - Entero con signo: -32,768 a 32,767.
  - Enteros sin signo: 0 a 65,535.
- DWORD: dirección en TIA Portal (MD10...).
  - Es una secuencia de 32 bits.
  - Enteros con signo: -2,147,483,648 a 2,147,483,648.
  - Enteros sin signo: 0 a 4,294,967,295.

### 2.4.2. Números enteros

- SINT (*Short Int*): tiene una longitud de 8 bits. Consta de dos componentes: un signo y un valor numérico. Los bits del 0 al 6 representan el valor numérico y el bit 7 el signo (0 positivo y 1 negativo).
  - Enteros con signo (sistema decimal) de -128 a 127.
- USINT (*Unsigned Short Int*): tiene una longitud de 8 bits y contiene valores sin signo.
  - Entero sin signo: 0 a 255.
- INT: longitud de 16 bits. Los bits del 0 al 14 representan el valor numérico y el bit 15 el signo (0 positivo y 1 negativo).

- Con signo: -32,768 a 32,767.
- UINT: longitud de 16 bits.
  - Sin signo: 0 a 65,535.
- DINT: longitud 32 bits.
  - Con signo: -2,147,483,648 a 2,147,483,648.
- DUINT: longitud 32 bits.
  - Sin signo: 0 a 4,294,967,295.

### **2.4.3. Números en coma flotante**

- Real: longitud de 32 bits.
  - El signo es determinado por el bit 31 (0 positivo y 1 negativo).
  - Exponentes de 8 bits en base 2. El exponente se incrementa en una constante (base, +127) y presenta valores entre 0 y 255.
- LReal: longitud de 64 bits.
  - El signo es determinado por el bit 63 (0 positivo y 1 negativo).
  - Exponentes de 11 bits en base 2. El exponente se incrementa en una constante (base, +1023) y presenta valores entre 0 y 2047.

## **2.5. Variables de entrada digital y analógica**

El PLC permite leer y transmitir variables de forma analógica y variables, permitiendo también que las variables analógicas se puedan convertir a variables digitales para su uso en la programación.

### **2.5.1. Digitales**

El primer conjunto de símbolos es de variables de entrada, denominadas como contacto normalmente abierto (N.A) o contacto normalmente cerrado (N.C.). Se encargan de informar el estado lógico en que se encuentran los botones utilizados en el sistema, las variables físicas captadas por los sensores, posición de relés, cualquier dispositivo que esté conectado y su salida sea encendido o apagado.

Si la salida del dispositivo está encendida el contacto normalmente abierto se activará y permitirá y el contacto normalmente cerrado se abrirá y no permitirá el paso, por el contrario, si la entrada del dispositivo está apagada el contacto normalmente abierto estará desactivado y no permitirá el paso y el contacto normalmente cerrado está activado permitiendo el paso. En resumen, cambian de estado cuando los contactos de entrada son manipulados. Con estos contactos se genera la lógica de programación del controlador lógico programable, esta consiste en ir permitiendo y quitando el paso dependiendo de las necesidades de la programación al activar y desactivar distintos contactos para encender o apagar las salidas.

### 2.5.2. Analógicas

El controlador lógico programable puede leer valores entre 0 y 27648, estos valores equivalen a rango de señal de entrada. Se debe realizar un escalado de la señal analógica de entrada para que esta sea entendida por el controlador, por lo que se realiza un escalamiento de los valores de entrada con el rango de lectura del controlador.

Suponiendo que se tiene una entrada entre 4mA y 20mA, se debe realizar el escalamiento de la entrada para que esta sea entendida por el controlador 0, que representa 0 mA, por lo que 5530 representa 4mA y 27648 20mA. Se utiliza la siguiente ecuación:

$$\frac{\text{valor escalado} - \text{valor mínimo de escala}}{\text{valor máximo de escala} - \text{valor mínimo de escala}} = \frac{\text{valor analogico} - \text{valor rango 4mA}}{\text{valor rango 20 mA} - \text{valor rango 4 mA}}$$

- Valor analógico: lectura de sensor
- Valor máximo de escala: valor máximo del sensor
- Valor mínimo de escala: valor mínimo del sensor
- Valor rango 4 mA: 5530
- Valor rango 20 mA: 27648
- Valor escalado: resultado

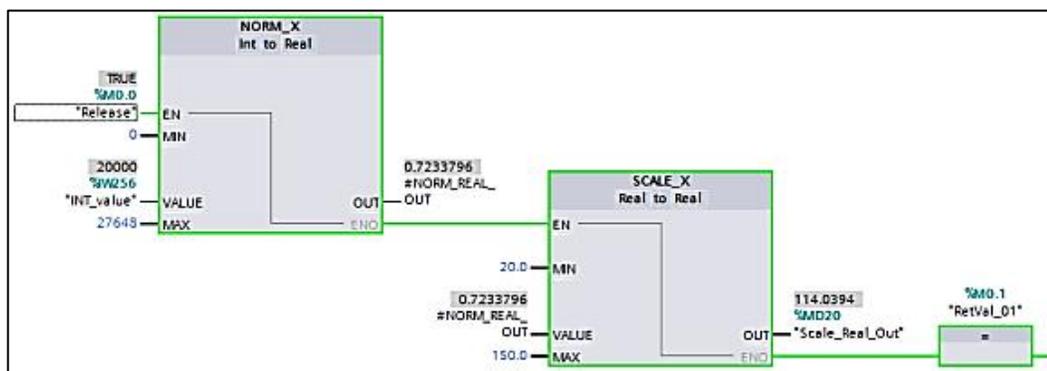
### 2.6. Escalado utilizando SCALE\_X y NORM\_X

- NORM\_X: instrucción utilizada para normalizar el valor de la variable de entrada y la representa en una escala lineal, los valores máximo y mínimo representarán los límites de un rango de valores reflejado en la escala, si el valor a normalizar es igual al valor mínimo de entrada la salida será baja

y, si el valor a normalizar es igual al valor máximo de entrada, la salida será alta.

- **SCALE\_X:** se encarga de escalar el valor de la entrada analógica, mapeando en un rango de valores el número en coma flotante de la entrada se escala al rango de valores definidos por los parámetros mínimo y máximo, el resultado al escalar será un número entero en la salida.

Figura 7. **Uso de TIA Portal**



Fuente: GÚTIEZ, Iñigo. *Cómo realizar el escalado de una señal analógica en TIA Portal.*

[https://programacion Siemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado\\_de\\_una\\_senal\\_analogica\\_de\\_forma\\_calculada](https://programacion Siemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado_de_una_senal_analogica_de_forma_calculada). Consulta: 18 de agosto de 2019.

### 2.6.1. Variables de salida digitales

Estas variables de salida pueden ser utilizadas para distintas funciones, desde encender luces piloto para conocer el estado del proceso hasta activar elementos de potencia como motores, tanto de corriente alterna como de corriente directa. La simbología de las variables de salida tiene el mismo significado que una bobina de un relevador en un diagrama eléctrico. Existen dos

tipos de salidas: de salida externa, esta realmente va a energizar una bobina física, y el otro tipo de salida es de memoria, conocida como marca, la cual será utilizada para la lógica interna del programa y no representa ninguna salida física a bobina. Estas variables serán activadas dependiendo de los requisitos que se encuentran antes de la salida, lo que indicará si la salida permanece en estado encendido o apagado.

Es posible encontrar también las variables de salida *SET* y *RESET*, este tipo de variables presentan una salida en la que el estado de la salida permanece sin necesidad de estar teniendo todo el tiempo los requisitos anteriores para mantenerlo activado, es necesario encender su opuesto para que este cambie de estado. La principal ventaja de este tipo de bobinas es que únicamente con un pulso el estado de la bobina cambia, no necesita estar recibiendo una señal de encendido para que esta permanezca en su estado, y cambiará de estado hasta que su opuesta sea presionada.

### **2.6.2. Marcas**

Las marcas (M) son bits internos a la CPU. El tamaño del área de marcas dependerá de la CPU, ya que dependiendo del modelo tiene más o menos memoria interna, no es necesario utilizar una salida o entradas reales para el uso de las marcas, por lo que son muy utilizadas en las simulaciones.

Las marcas pueden verse en forma de:

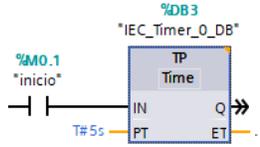
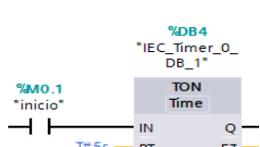
- Bit, por ejemplo, M0.0
- Byte (8bits), por ejemplo, MB0
- *Word* o palabra (16bits), por ejemplo, MW0
- Doble palabra (32bits), por ejemplo, MD0

Todos los bits de marca están en estado nulo cuando se arranca la CPU (se pasa de estado STOP a RUN). Esto es así salvo que se haya configurado la CPU para que parte de las marcas sean remanentes, esto es, que mantengan su valor aun parando el programa o incluso apagando la CPU (mientras que dure la batería de esta).<sup>6</sup>

### 2.6.3. Temporizadores

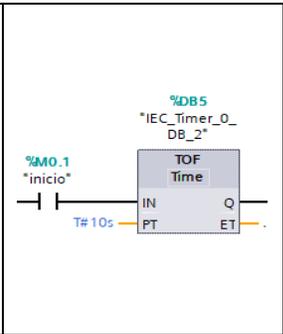
Se utilizan los temporizadores para el retraso de uso de alguna variable en la programación, permitiendo que una acción se lleve a cabo hasta que el tiempo programado termine, muy útil para procesos que requieren acciones con tiempos específicos en la industria. Se cuentan con diferentes tipos de temporizadores según la necesidad de uso.

Tabla IX. Temporizadores

<p><b>TP:</b> genera impulso de una duración determinada. Se activa la salida Q por un tiempo programado, al tener una entrada alta empieza a contar el tiempo programado en el parámetro pt, cada vez que se pulse la entrada IN no influye en el estado de la salida Q, en la salida ET se puede consultar el valor del tiempo actual, no es necesario dejar pulsado.</p>	
<p><b>TON:</b> retardo al conectar: la salida será alta hasta que el tiempo preestablecido se cumpla, al cumplirse el tiempo la salida será alta. Retarda la activación de la salida Q por el tiempo programado en PT. La instrucción inicia cuando IN cambia de 0 a 1, la salida Q permanece activa todo el tiempo que la entrada IN está en alto, para reiniciar se debe llevar a 0 y luego otra vez a 1 la entrada IN. El valor de tiempo actual se puede consultar en la salida ET.</p>	

<sup>6</sup> GÚTIEZ, Iñigo. *Cómo realizar el escalado de una señal analógica en TIA Portal*. [https://programacion Siemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado\\_de\\_una\\_senal\\_analogica\\_de\\_forma\\_calculada](https://programacion Siemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado_de_una_senal_analogica_de_forma_calculada). Consulta: 18 de agosto de 2019.

Continuación de la tabla IX.

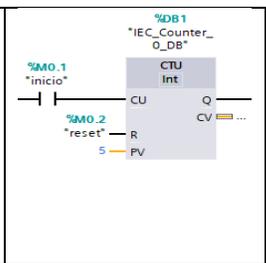
<p><b>TOF:</b> retardo al desconectar: la salida será alta hasta que el tiempo preestablecido se cumpla, al cumplirse el tiempo la salida será baja. La salida Q se activa cuando la entrada IN cambia a estado lógico 1 y cuando pasa a 0 lógico la entrada el tiempo PT empieza a contar, al transcurrir el tiempo PT programado se desactiva la salida Q, el valor del tiempo actual se puede consultar en la salida ET, si está en 0 la entrada IN, y se activa de nuevo antes de terminar el tiempo, este se reiniciará.</p>	
---	---

Fuente: elaboración propia.

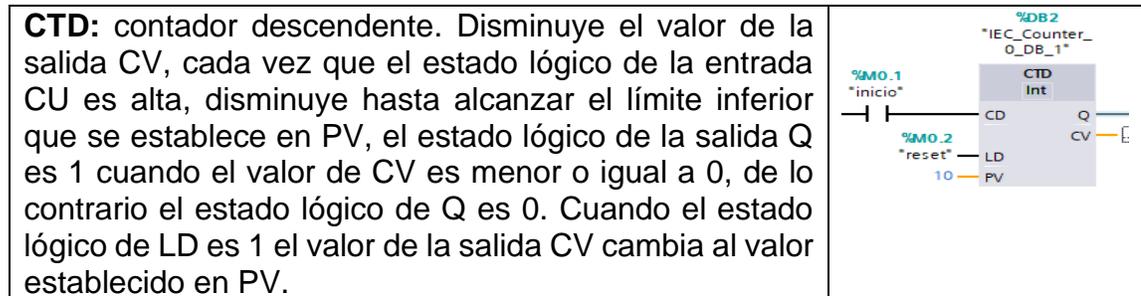
### 2.6.4. Contadores

Se utilizan contadores para tener el control de las veces que una variable se repite y al llegar a la cantidad deseada continuar con el siguiente paso. Esto es útil para conteos dentro del proceso industrial, para conocer cantidades de ciertos elementos. Se cuenta con diferentes tipos de contadores según la necesidad de uso:

Tabla X. Contadores

<p><b>CTU:</b> contador ascendente. Incrementa el valor en la salida CV, cada vez que el estado lógico de la entrada CU es alta, hasta alcanzar el valor límite que se establece en la entrada PV, el valor de la salida Q es 1, siempre que el valor de CV sea mayor o igual al valor en PV, en los demás casos la salida Q es 0, mientras la entrada r tenga estado lógico 1 se resetea el valor de CV.</p>	
---	---

Continuación de la tabla X.



Fuente: elaboración propia.



### 3. INICIAR PROYECTO TIA PORTAL

- Paso 1: clic en crear proyecto:

Figura 8. Iniciar proyecto paso 1



Fuente: elaboración propia.

- Paso 2: colocar nombre, destino y comentario y clic en crear:

Figura 9. Iniciar proyecto paso 2

Crear proyecto

Nombre proyecto: Primer Proyecto tesis

Ruta: C:\Users\fredd\Documents\Automation

Autor: freddy

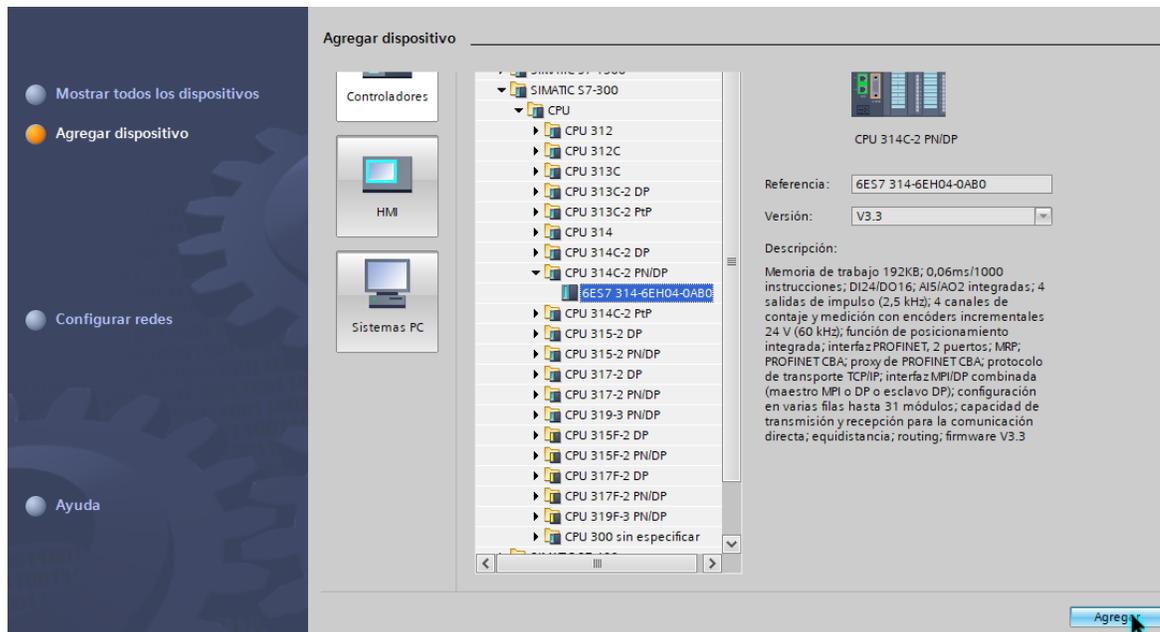
Comentario: creando primer proyecto

Crear

Fuente: elaboración propia.

- Paso 3: clic en configurar dispositivo, agregar dispositivo, abrir pestaña SIMATICS S7-300, abrir CPU, se puede escoger cualquiera para la simulación, pero se deben consultar las distintas entradas y salidas de cada uno, en este caso se usará el CPU 314c-2 PN/DP, seleccionar y agregar.

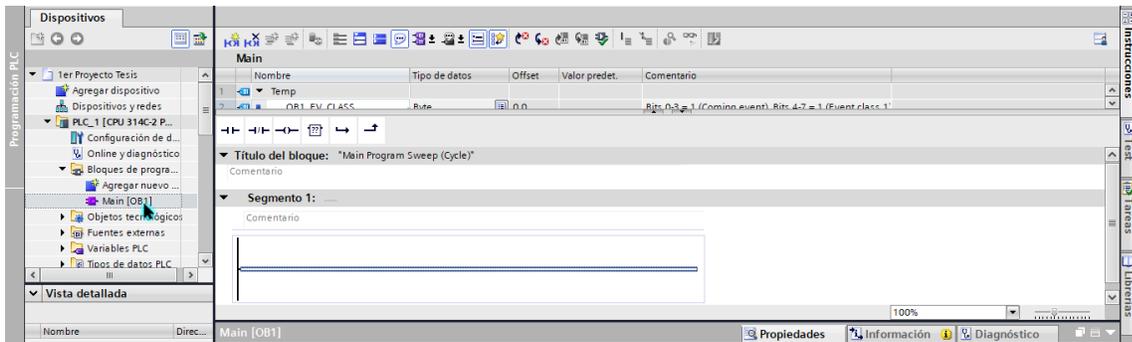
Figura 10. Iniciar proyecto paso 3



Fuente: elaboración propia.

- Paso 4: de lado izquierdo abrir pestaña del PLC, abrir pestaña bloques de programa, y clic en *main*.

Figura 11. Iniciar proyecto paso 4



Fuente: elaboración propia.

## 4. UTILIZANDO PLCSIM

### 4.1. Definición

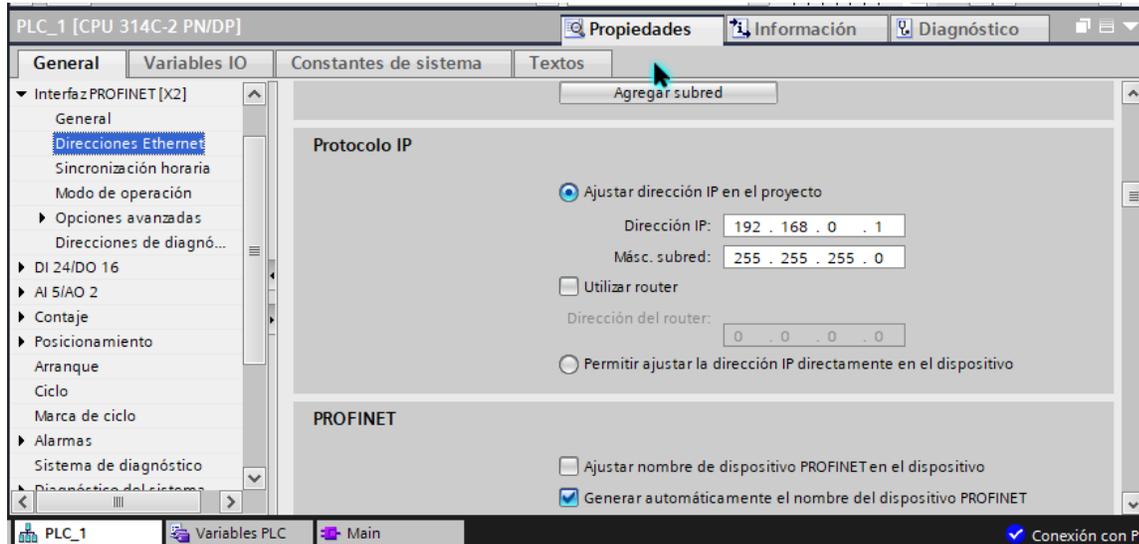
PLCSIM permite probar los programas del PLC en un PLC simulado sin necesidad de hardware real, lo cual brinda una gran ventaja para el aprendizaje del controlador. Con PLCSIM es posible avanzar a través de la lógica del programa, observar el efecto de las entradas y salidas simuladas, así como ajustar la lógica de su programa según sea necesario para obtener el resultado esperado. Se simula de la misma forma en la que se llevaría a cabo en un PLC real.

### 4.2. Implementación del proyecto

Ya con el programa terminado, se procede a descargar en el PLC virtual el programa y de esta forma observar la simulación de lo que realiza el PLC.

- Paso 1: colocar IP a PLC escogido. Clic en configurar dispositivo, se abrirá la pestaña del PLC, dar clic derecho sobre PLC (s7-300) y luego clic en propiedades. En la pestaña general de propiedades se debe dar clic en Interfaz PROFINET, luego en direcciones *ethernet*, y ahí se coloca la dirección IP y máscara de subred deseada.

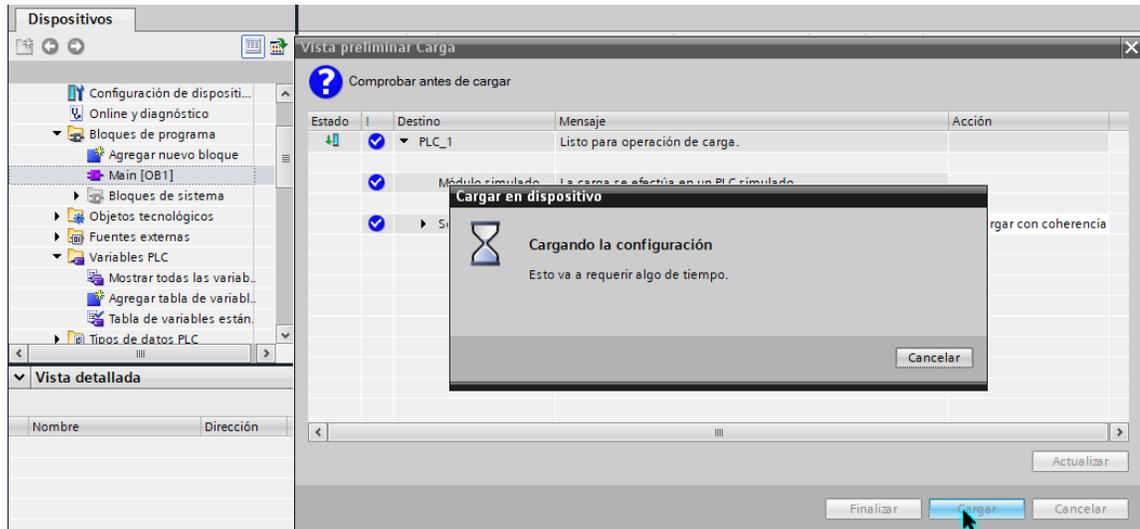
Figura 12. Implementación del proyecto



Fuente: elaboración propia.

- Paso 2: clic en iniciar simulación, se abrirá una ventana que indica que se desactivan todas las interfaces *online* restantes, dar clic en aceptar y se abrirá la pestaña de carga avanzada, en tipo de interfaz debemos seleccionar PN/IE y en interfaz PG/PC se selecciona PLCSIM y se da clic en iniciar búsqueda. Se selecciona el dispositivo con la dirección IP antes establecida y se da clic en cargar, se abrirá otra pestaña en la que se debe seleccionar sobrescribir o cargar de todos modos y, al obtener todos con un *check* azul, se procede a cargar.

Figura 13. Cargar proyecto



Fuente: elaboración propia.

Aceptar y esperar a que cargue el simulador PLCSIM.

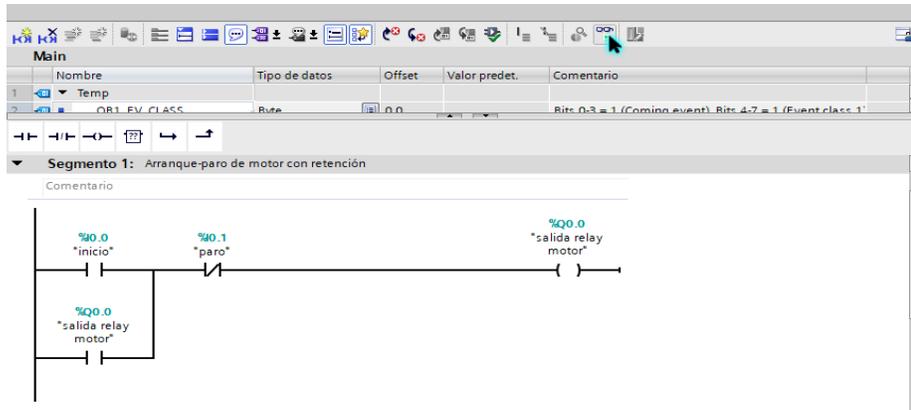
Nota: al volver a cargar el programa es necesario que el simulador esté en stop.

### 4.3. Uso y visualización

Una vez cargado el programa al PLC virtual, se abrirá una ventana en la cual se pueden observar las entradas y salidas virtuales de este, estas se pueden manipular al gusto, simulando las entradas en alto y las entradas en bajo de los sensores, se puede simular de la misma forma los conteos.

- Paso1: activar o desactivar observación:

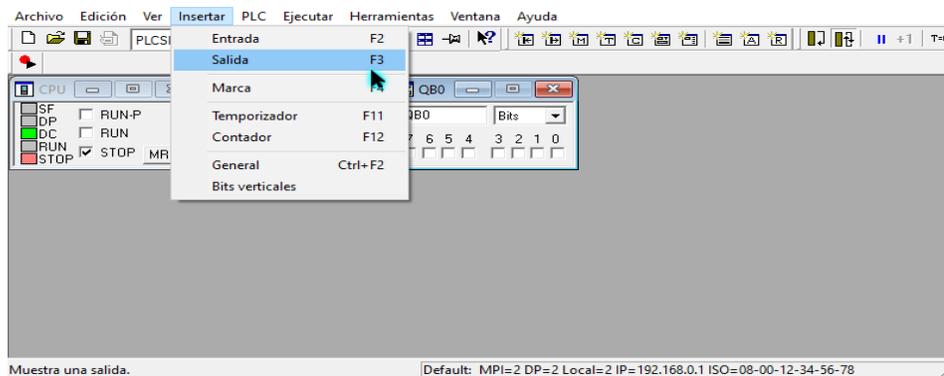
Figura 14. **Activar visualización**



Fuente: elaboración propia.

- Paso 2: en la ventana de PLCSIM, insertar lo que se desee simular: entradas, salidas, contadores, entre otros.

Figura 15. **Simulación**

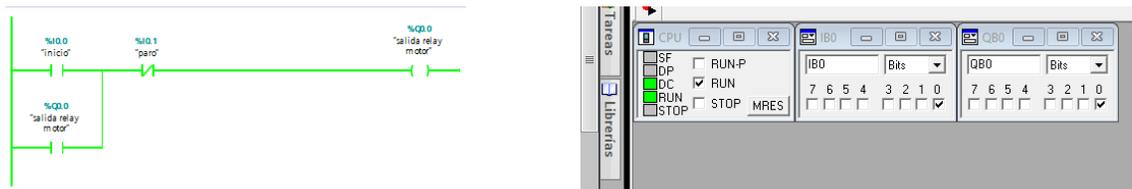


Fuente: elaboración propia.

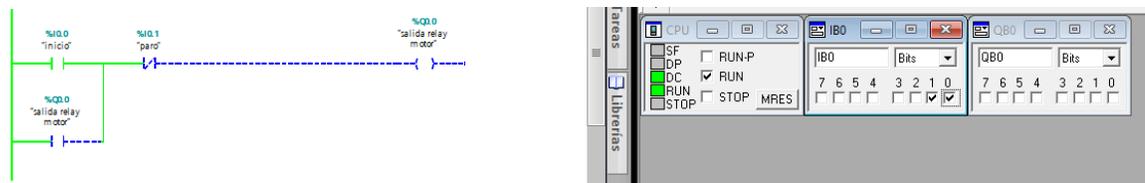
#### 4.4. Ejemplo

Simulación puesta en marcha de motor con pulsador.

Figura 16. Ejemplo de simulación de puesta en marcha



Simulación paro de motor con pulsador.



Fuente: elaboración propia.

#### 4.5. Ejercicios prácticos en simulador

Se deben realizar los siguientes ejercicios en entorno de programación TIA Portal con simulador PLCSIM. Los ejercicios pueden funcionar con distintas lógicas de programación siempre y cuando se cumplan los requerimientos establecidos en los enunciados.

- Implementar la marcha-parada de un motor mediante un pulsador de arranque con retención y otro de parada. Implementar con bobinas comunes y con bobinas de *set* y *reset*.

- Realizar la simulación de un semáforo utilizando *timers*, este debe ser automático. Color verde (M0.1): 7 segundos, color amarillo (M0.2): 2 segundos y color rojo (M0.3): 5 segundos.
- Se debe realizar el control de marcha-para de una cinta transportadora de cajas, para lo cual es necesario contar la cantidad de cajas que han ingresado, se deben llenar por 4 segundos y seguir con el recorrido.
- Se debe realizar el control de llenado de un silo, la bomba de llenado debe iniciar con un pulsador no retentivo y llevar el líquido al silo hasta llenarlo, al estar lleno se debe calentar por 10 segundos y luego ser expulsado por una electroválvula ubicada en el fondo del silo y automáticamente repetir el proceso al quedar vacío el silo, se cuenta con sensor de lleno y vacío.
- Se debe realizar el control de un motor trifásico, este debe funcionar con pulsadores retentivos y debe poder controlarse el giro en derecho y reversa, se realiza por medio de contactores. Se debe evitar que los dos contactores se activen al mismo tiempo para evitar que se queme el motor y se debe evitar que el cambio sea brusco, por lo que, al cambiar el giro, el motor debe parar y 5 segundos después girar al lado contrario, debe contar con paro de emergencia.

## CONCLUSIONES

1. El presente trabajo de investigación se llevó a cabo con el fin de ser un material didáctico práctico y teórico para conocer el uso y funciones de los controladores lógicos programables.
2. Se realizó una herramienta que ayudará a los estudiantes a enfrentarse a las distintas pruebas que encontrarán en el ámbito laboral de mejor manera.
3. Se brindan recursos para configurar el controlador lógico programable S7'300 de Siemens, el cual es un controlador general, por lo cual lo aprendido es aplicable a otros controladores, incluso de marcas distintas.
4. El uso del software PLCSIM permite practicar la programación sin la necesidad de contar con un controlador físico.



## RECOMENDACIONES

1. Dar un seguimiento a los estudiantes en el proceso de aprendizaje de la programación de los controladores, resolviendo dudas y solucionando problemas de lógica al momento de programar.
2. Utilizar los ejercicios prácticos como una evaluación para los estudiantes y conocer el avance del conocimiento adquirido.
3. Entender la lógica del programa o ejercicio antes de empezar con la programación, ya que esto ampliará la perspectiva de cuanto se desea realizar, logrando que la programación sea más sencilla y efectiva.
4. Recordar que en campo se deben tomar en cuenta distintas variables de magnitudes físicas que se pueden presentar que alteran las magnitudes medidas y la forma de recibirlas el controlador, por lo cual puede ser necesario utilizar ecuación matemática para normalizar las entradas analógicas al PLC.
5. Continuar posteriormente con la investigación y práctica de los controladores lógicos programables para aumentar el conocimiento, ya que este tema es sumamente amplio, con muchísimas aplicaciones y soluciones.
6. Tomar en cuenta que es necesario mantenerse en actualización constante de los sistemas y controladores que estarán revolucionando día a día el ámbito laboral de la automatización industrial.

7. Continuar con la elaboración de guías académicas para permitir la mejora constante de los egresados de la Escuela de Mecánica Eléctrica.

## BIBLIOGRAFÍA

1. Amelero. *Forzado permanente entradas y salidas S7-300 con simulador PLCSIM*. [en línea]. <<https://www.amelero.com/recursos/instalaciones-electrot%C3%A9cnicas/forzado-entradas-y-salidas-step-7-tia-portal/>>. [Consulta: 18 de agosto de 2019].
2. Anónimo. *PLC Hardware*. [en línea]. <<http://www.plcmanual.com/plc-hardware>>. [Consulta: 18 de agosto de 2019].
3. GÚTIEZ, Iñigo. *Cómo realizar el escalado de una señal analógica en TIA Portal*. [en línea]. <[https://programacionsiemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado\\_de\\_una\\_senal\\_analogica\\_de\\_forma\\_calculada](https://programacionsiemens.com/escalado-de-una-senal-analogica-en-tia-portal/#Escalado_de_una_senal_analogica_de_forma_calculada)>. [Consulta: 18 de agosto de 2019].
4. Infopl. *Norma IEC 1131-3. Lenguajes de programación*. [en línea]. <<https://www.infopl.net/documentacion/67-estandares-programacion/8-lenguajes-de-programacion-norma-iec-1131-3>>. [Consulta: 18 de agosto de 2019].
5. Siemens. *S7-300 Sistema de automatización S7-300. Datos de los módulos*. Nürnberg, Alemania: Siemens AG. 2017. 608 p.

6. YUGSI, R. *Lenguajes de programación de PLC'S*. [en línea]. <<http://bibdigital.epn.edu.ec/bitstream/15000/9276/6/LENGUAJES%20DE%20PROGRAMACION%20DE%20PLC.doc>>. [Consulta: 18 de agosto de 2019].

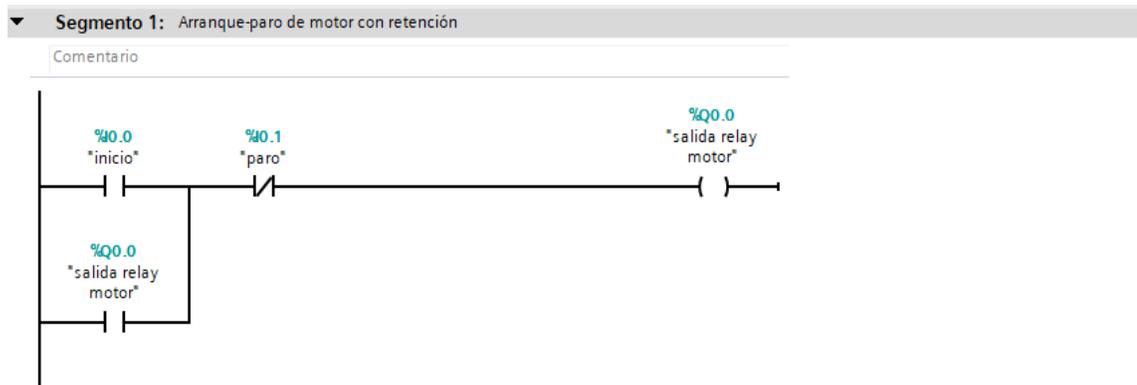
# APÉNDICES

## Apéndice 1. Ejercicio A: arranque-paro de motor

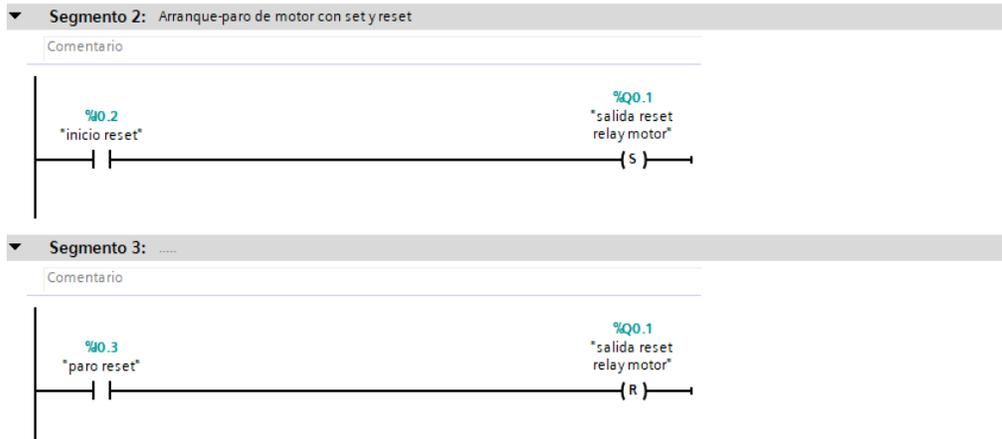
### Tabla de variables

Tabla de variables estándar							
	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	inicio	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	paro	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	salida relay motor	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	inicio reset	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	paro reset	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	salida reset relay motor	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	<input type="text" value="&lt;Agrega&gt;"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

- Programación de escalera



Continuación del apéndice 1.

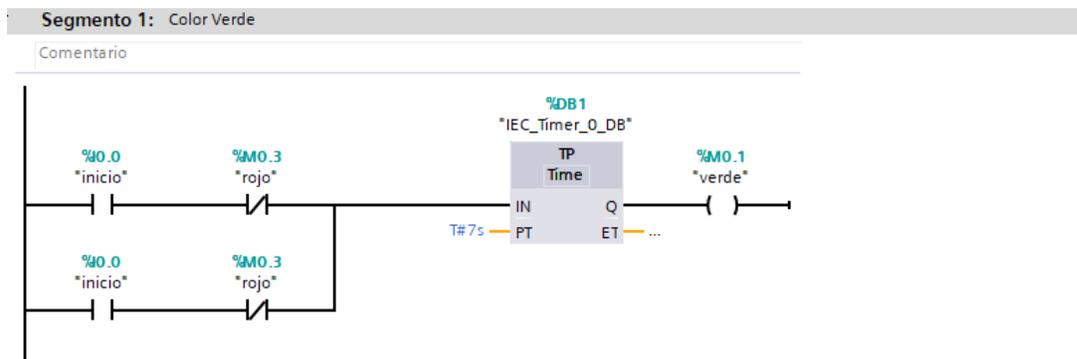


Fuente: elaboración propia.

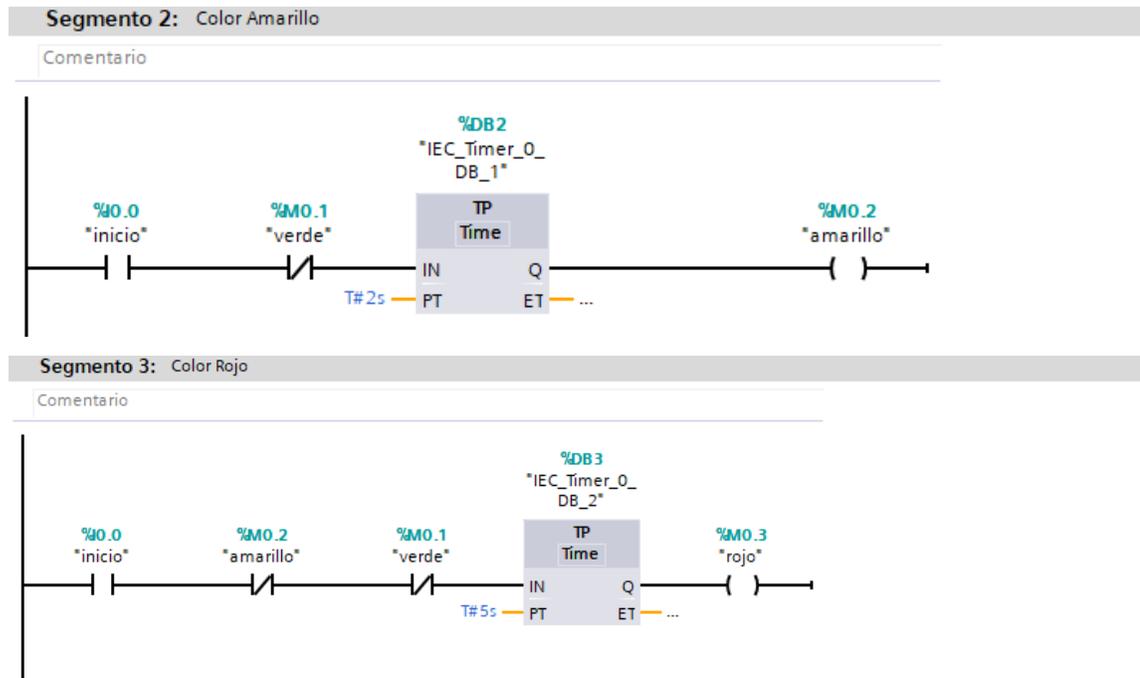
## Apéndice 2. Ejercicio B: semáforo

- Tabla de variables

	Nombre	Tabla de variables e..	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	inicio	Tabla de variables e..	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	time verde	Tabla de variables e..	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	verde	Tabla de variables e..	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	amarillo	Tabla de variables e..	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	rojo	Tabla de variables e..	Bool	%M0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	<Agrega>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Continuación del apéndice 2.



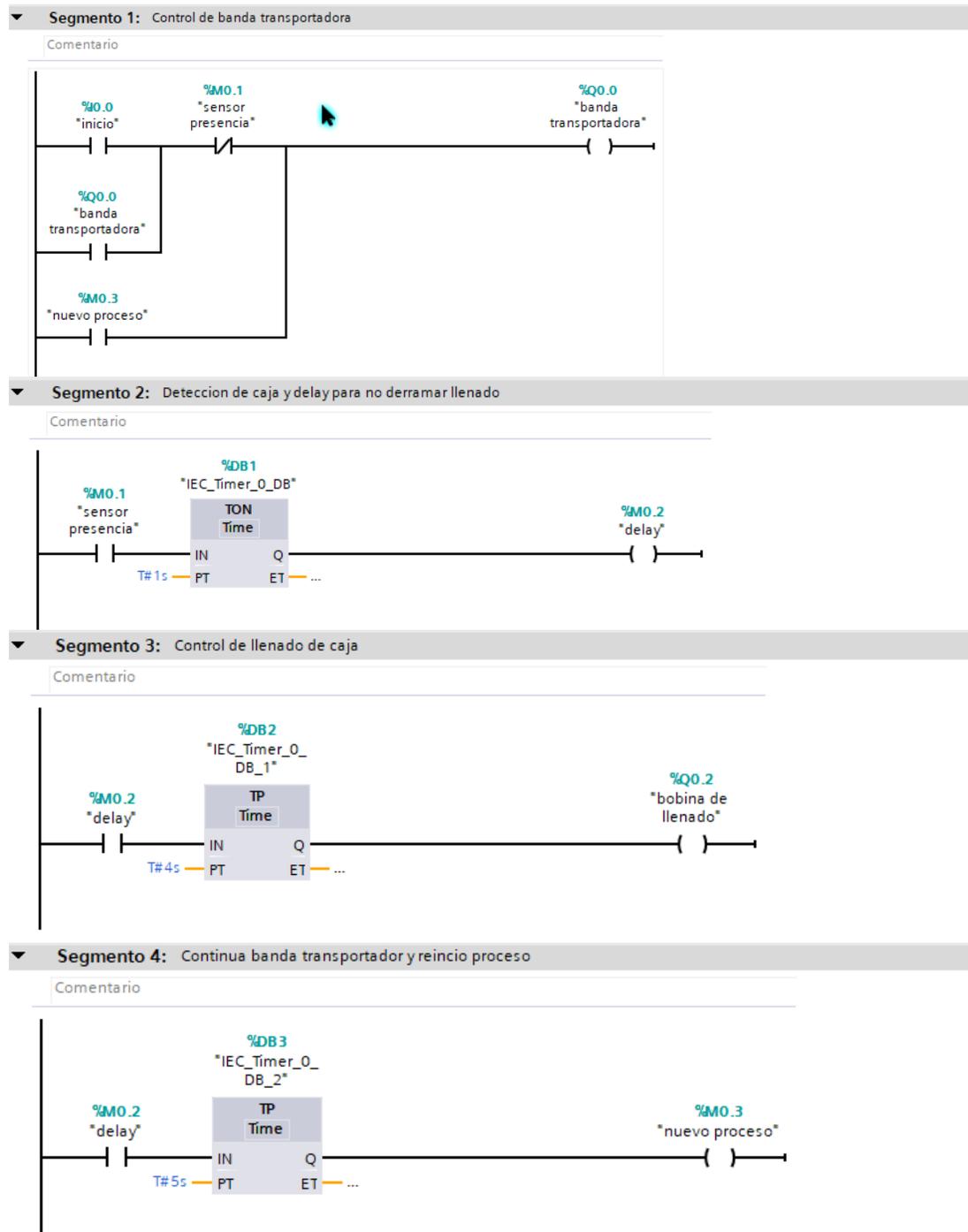
Fuente: elaboración propia.

### Apéndice 3. Ejercicio C: contador de cajas

- Tabla de variables

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	inicio	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	banda transportadora	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	sensor presencia	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	bobina de llenado	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	delay	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	nuevo proceso	Bool	%M0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	<Agrega>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

### Continuación del apéndice 3.



Fuente: elaboración propia.

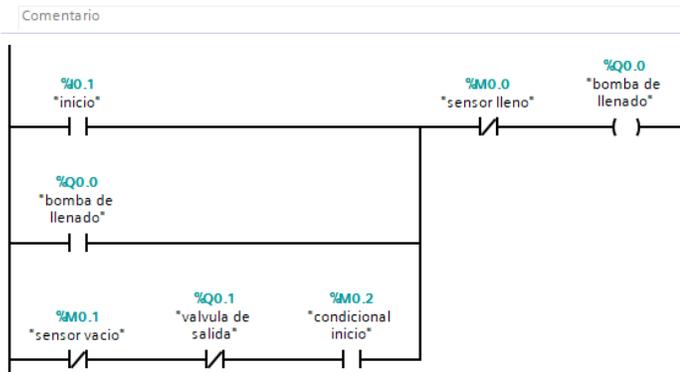
## Apéndice 4.

## Ejercicio D: llenado de silo

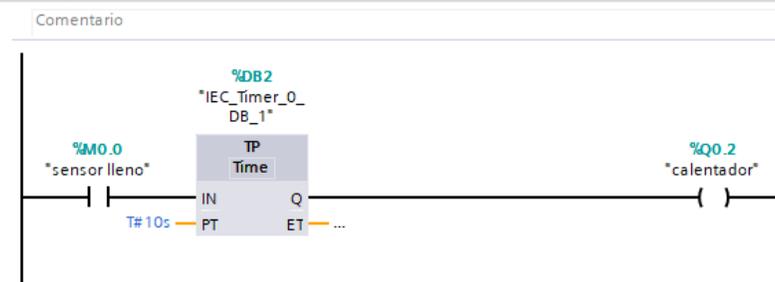
- Tabla de variables

	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	bomba de llenado	Tabla de variables e..	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	sensor lleno	Tabla de variables e..	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	sensor vacio	Tabla de variables e..	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	valvula de salida	Tabla de variables e..	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	calentador	Tabla de variables e..	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	inicio	Tabla de variables e..	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	condicional inicio	Tabla de variables e..	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	<Agregar>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

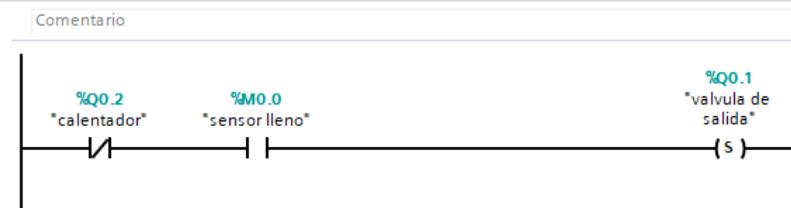
### Segmento 1: Control de inicio de llenado



### Segmento 2: Tiempo de calentado



### Segmento 3: Vaciado de silo al terminar de calentar



Continuación del apéndice 4.

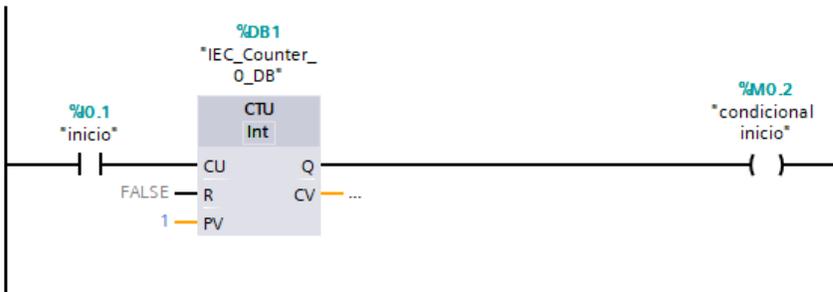
**Segmento 4: Cerrado de electrovalvula de vaciado para llenado nuevo**

Comentario



**Segmento 5: Condicional para inicio automatico**

Comentario



Fuente: elaboración propia.

**Apéndice 5. Ejercicio E: control del motor trifásico**

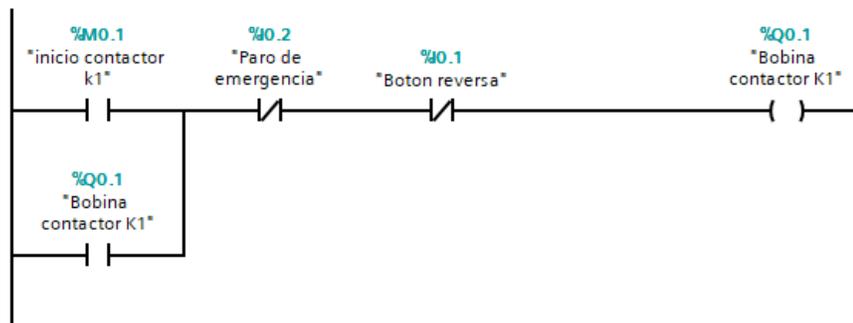
- Tabla de variables

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	<A> Boton giro	Bool	%IO.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	<A> Boton reversa	Bool	%IO.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	<A> Bobina contactor K1	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	<A> Bobina contactor K2	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	<A> Paro de emergencia	Bool	%IO.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	<A> inicio contactor k1	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	<A> inicio contactor k2	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	<A> temp1	Bool	%M0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	<A> temp2	Bool	%M0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	<A> <Agregar>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

## Continuación del apéndice 5.

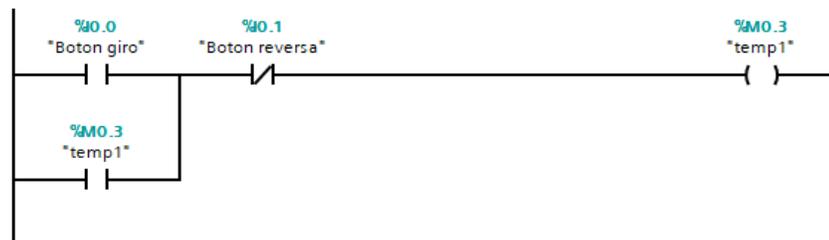
### ▼ Segmento 1: Control de bobina de contactor giro

Comentario

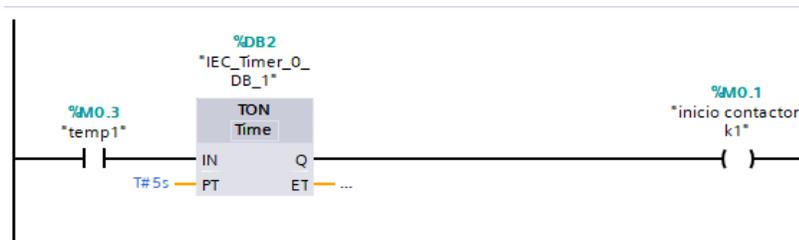


### ▼ Segmento 2: Boton para inicio de tiempo

Comentario

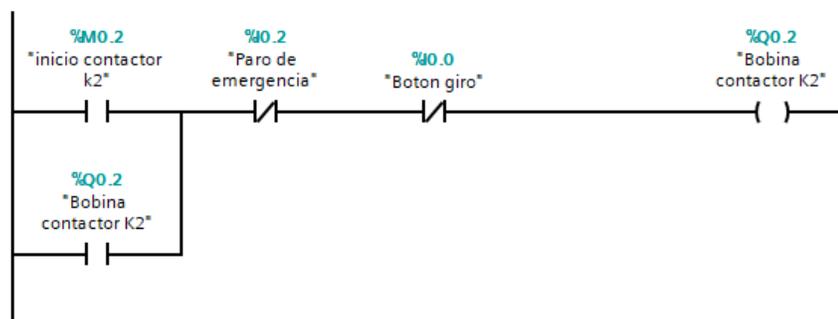


### ▼ Segmento 3: Tiempo de desconexion bobina contactor 1



### ▼ Segmento 4: Control de contactor giro reversa

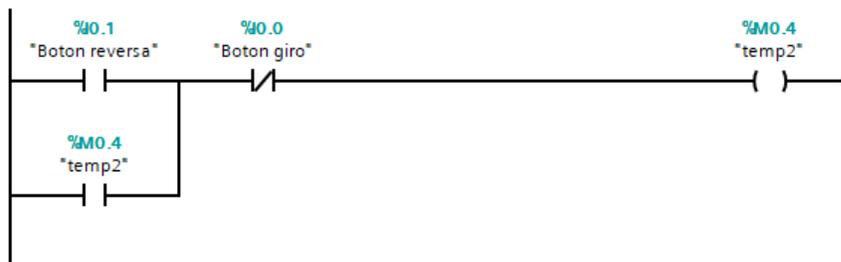
Comentario



## Continuación del apéndice 5.

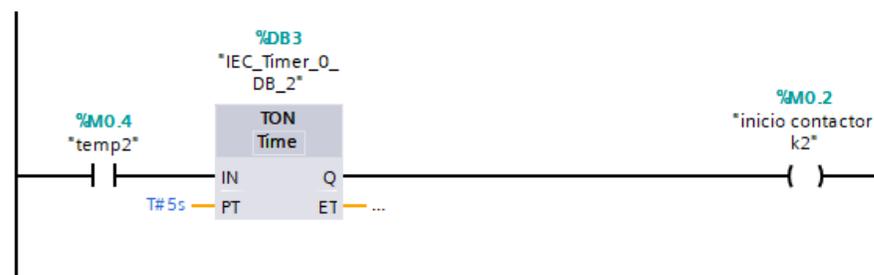
### ▼ Segmento 5: Boton para inicio de tiempo

Comentario



### ▼ Segmento 6: Tiempo de desconexion bobina contactor 2

Comentario



Fuente: elaboración propia.