



# **Fundamentos de Automatización y Redes Industriales**



# **Fundamentos de Automatización y Redes Industriales**

Mireya Zapata, Liliana Topón-Visarrea, Edgar Tipán

**Fecha de publicación:** 20 de junio de 2021

**Autoridades**

Ing. Saúl Lara – Canciller Dr. Franklin Tapia – Rector

Ing. Jorge Cruz, Ph.D – Vicerrector

Ing. Diego Lara – Director Institucional Académico

Ing. Janio Jadán, Ph.D – Director Institucional de Investigación

©**Autores:** Mireya Zapata<sup>1</sup>, Liliana Topón-Visarrea<sup>1</sup>, Edgar Tipán<sup>2</sup>.

<sup>1</sup> Centro de Investigación en Mecatrónica y Sistemas Interactivos MIST, Facultad de Ingeniería y Tecnologías de la Información y la Comunicación, Universidad Tecnológica Indoamérica, Av. Machala y Sabanilla, Quito-Ecuador.

Correo: mireyazapata@uti.edu.ec, blancatopon@uti.edu.ec

<sup>2</sup> Departamento de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, Universidad de las Fuerzas Armadas (ESPE), Sangolquí-Ecuador. Correo: eftipan@espe.edu.ec

**ISBN** 978-9942-821-21-8

**N° Derecho de autor:** 060335

Revisado y aprobado para su publicación por el Comité Editorial de la Universidad Tecnológica Indoamérica (Quito, Ecuador) y por los revisores MSc. Bernardo Vallejo Mancero (Universitat Politècnica de Catalunya, España) y MSc. Milton Alvear Alvear (Instituto Superior Tecnológico Central Técnico, Ecuador).

**Editor:** Ing. Hugo Arias Flores, MBA.

Editorial de la Universidad Tecnológica Indoamérica. Quito, Ecuador.



Queda rigurosamente prohibida la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la fotocopia y el tratamiento informático, sin autorización escrita del titular del Copyright, bajo las sanciones previstas por las leyes.

Para citar este libro:

Zapata, M., Topón, L. Tipán, E. (2021). Fundamentos de Automatización y Redes Industriales. Quito, Ecuador: Editorial Universidad Tecnológica Indoamérica.

## Índice General

Prólogo .....	11
---------------	----

### Capítulo 1

#### Introducción a la Automatización y Autómatas Programables ...13

1.1. ¿Cómo se define a la Automatización? .....	16
1.1.1. Ventajas.....	16
1.1.2. Desventajas.....	17
1.1.3. Opciones de Automatización.....	17
1.2. Metodología de creación de proyectos de automatización industrial.....	20
1.2.1. Gestión del proyecto vs. trabajo del proyecto.....	20
1.2.2. Metodología de gestión del proyecto.....	21
1.2.3. Metodología de trabajo del proyecto: Modelo V.....	21
1.2.4. Iteración del modelo V .....	23
1.3. Autómatas programables .....	25
1.3.1. Arquitectura de PLC .....	27
1.3.2. Tipos de PLC.....	33
1.3.3. Lenguajes de programación.....	35
1.3.4. Funcionamiento .....	37
1.4. Preguntas de repaso .....	39

### Capítulo 2

#### Creación de Proyectos con TIA Portal.....41

2.1. Introducción .....	43
2.1.1. Requisitos mínimos .....	44
2.1.2. Entorno de trabajo .....	44
2.2. Tipo de datos.....	46

2.2.1. Tipo de datos binarios y secuencia de bits.....	47
2.2.2. Tipos de datos matemáticos .....	49
2.2.2. Tipo de datos de tiempo .....	50
2.3. Creación de proyectos - ejemplo de aplicación .....	51
2.3.1. Crear un proyecto .....	51
2.3.2. Crear un programa .....	53
2.3.3. Simulación.....	55
2.4. Interfaz HMI.....	60
2.4.1. Ventajas de usar un HMI.....	61
2.4.2. Agregar Interfaz HMI .....	62
2.4.3. Carga del programa al PLC y HMI.....	67
2.5. Preguntas de repaso .....	68

### **Capítulo 3**

<b>Lenguaje de programación Ladder .....</b>	<b>71</b>
3.1. Introducción .....	73
3.2. Estructura del diagrama Ladder .....	74
3.3. Simbología.....	75
3.4. Asignación de etiquetas.....	76
3.5. Propositiones lógicas.....	76
3.5.1. Proposición lógica piloto.....	77
3.5.2. Proposición lógica memorizada .....	78
3.5.3. Proposición lógica con elemento de arranque y paro.....	79
3.5.4. Proposición lógica con sistema de bloqueo.....	80
3.5.5. Proposición lógica con condicionamiento al encendido.....	81
3.5.6. Proposición lógica con condicionamiento al apagado.....	82
3.5.7. Proposición lógica de dos contactares con alternancia.....	83
3.5.8. Ejemplos utilizando proposiciones lógicas.....	84
3.6. Temporizadores, contadores y comparadores .....	94



3.6.1. Temporizadores.....	94
3.6.2. Tipo de temporizadores.....	94
3.6.3. Contadores.....	101
3.6.4. Comparadores .....	104
3.6.5. Ejemplos utilizando comparadores y contadores .....	104
3.7. Funciones de transferencia .....	106
3.7.1. Función MOVE .....	107
3.8. Funciones matemáticas.....	107
3.9. Señales Analógicas.....	109
3.9.1. Entradas y salidas analógicas.....	110
3.9.2. Como escalar señales de entradas analógicas ..	111
3.9.3. Cómo configurar salidas analógicas.....	120
3.10. Ejercicios propuestos .....	123

## Capítulo 4

<b>Lenguaje de programación GRAFCET.....</b>	<b>127</b>
4.1. Introducción .....	129
4.2. Niveles de descripción .....	130
4.3. Elementos de GRAFCET .....	130
4.3.1. Etapas .....	130
4.3.2. Acciones asociadas .....	132
4.3.3. Transición y receptividad.....	134
4.3.4. Líneas de enlace .....	134
4.3.5. Bifurcaciones OR .....	134
4.3.6. Bifurcaciones AND.....	135
4.4. Macroetapas.....	136
4.5. Reglas de evolución .....	137
4.6. Conversión de GRAFCET a Ladder.....	143
4.6.1 Bifurcaciones condicionales OR.....	144
4.6.2. Bifurcaciones AND.....	145
4.6.3. Saltos condicionales .....	146
4.6.4. Acciones asociadas .....	147

4.7. Ejercicios propuestos .....	150
----------------------------------	-----

## **Capítulo 5**

<b>Redes Industriales .....</b>	<b>155</b>
---------------------------------	------------

5.1. Pirámide de Automatización .....	157
5.1.1. Nivel de campo .....	159
5.1.2. Nivel de control .....	160
5.1.3. Nivel de supervisión .....	161
5.1.4. Nivel de gestión de la producción .....	162
5.1.5. Nivel de planificación de recursos empresariales .....	163
5.2. Sistemas de comunicación .....	164
5.2.1. Elementos de una comunicación .....	165
5.2.2. Clasificación de los sistemas de comunicación....	166
5.2.3. Modelo de interconexión de sistemas abiertos (Open System Interconnection, OSI) .....	175
5.2.4. Protocolos de comunicación .....	177
5.2.5. Protocolo RS-232 .....	178
5.2.6. Protocolo RS-422 .....	181
5.2.7. Protocolo RS-485 .....	182
5.2.8. Ethernet .....	184
5.2.9. Clasificación de las redes industriales .....	187
5.2.10. Modbus .....	190
5.2.11. Profibus .....	191
5.2.12. Profinet .....	194
5.3. Ejemplos de configuración .....	196
5.3.1. Configuración Modbus .....	196
5.3.2. Configuración Profibus DP .....	204
5.3.3. Configuración Profinet .....	212
5.4. Ejercicios propuestos .....	220

<b>Bibliografía .....</b>	<b>227</b>
---------------------------	------------

## Prólogo

El desarrollo de nuevas tecnologías nos enfrenta a desafíos actuales y futuros que deben ser abordados desde la ciencia y el conocimiento. En este sentido, esta obra dirigida a estudiantes y profesionales pretende generar un campo de inducción a la automatización de procesos industriales, a través del estudio de la arquitectura y programación de uno de los dispositivos más importantes y, por consiguiente, más utilizado en la industria: el controlador lógico programable, conocido por sus siglas como PLC.

Este documento cubre aspectos que van desde conceptos generales de la automatización y su metodología de diseño hasta enfocarse en la explicación y desarrollo de ejemplos prácticos orientados al uso y programación de los PLC's. El desarrollo de los capítulos si bien se enfocan en la familia SIEMENS a través de la herramienta de desarrollo Software Totally Integrated Automation Portal (TIA Portal), las definiciones presentadas se pueden aplicar como base para entender las nociones de una arquitectura genérica, conexión con periféricos y bases de programación con cualquier otro tipo de fabricante. No obstante, se destacan las ventajas de TIA Portal para implementar tareas de automatización ya que es una alternativa eficiente con una gestión conjunta de datos en un entorno de ingeniería unificado para tareas de control, visualización y accionamiento. Con este software, los programadores pueden seleccionar diferentes lenguajes de programación, no obstante en esta obra se analizarán a detalle Ladder y GRAFCET.

Finalmente, se presenta el tema de Redes Industriales, señalando que la implementación de una red a nivel industrial brinda a las empresas una serie de ventajas competitivas en el mejoramiento de

los procesos de producción, aumento de la eficiencia y, sobre todo, un adecuado posicionamiento dentro de la nueva era de la Industria 4.0. Para el entendimiento de las redes industriales se describen los fundamentos teóricos de los niveles de la pirámide de automatización y los sistemas de comunicación, con una amplia explicación de los protocolos más utilizados a nivel industrial, así como su configuración y puesta en marcha en ejercicios prácticos.

**Capítulo 1**  
**Introducción**  
**a la Automatización y**  
**Autómatas Programables**



Todas las industrias constan de una serie de máquinas; algunas sencillas, otras más complejas. Luego de un proceso y una transformación secuencial, consiguen cambiar las materias primas en un producto final. Como una forma de mejorar estos productos, se han realizado esfuerzos para automatizar las configuraciones industriales. En un principio, las actividades eran completamente manuales, pero con el pasar del tiempo se ha ido reemplazando ojos, manos, cerebro de los trabajadores con alternativas eléctricas y electrónicas, tales como sensores, actuadores y controladores lógicos programables. Las operaciones automáticas se basan en estos tres elementos: a través de los sensores, se obtienen señales y medidas necesarias recopiladas del proceso controlado; posteriormente, esta información será analizada por el controlador lógico programable; finalmente, se ejecutará una acción a través de los actuadores. Así, la evolución de múltiples tecnologías ha permitido mejorar los procesos, siendo capaz de descentralizar la industria y sincronizar múltiples procesos industriales.

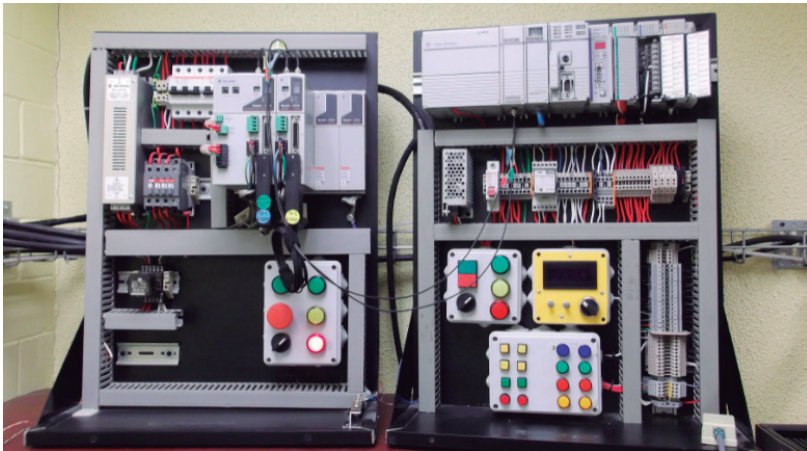


Figura 1.1: Tablero de control

## 1.1. ¿Cómo se define a la Automatización?

El término automatización ha sido utilizado desde la antigüedad. Viene del griego *autos* que significa “por sí mismo”, y *maiomai*, lanzar. Traduciendo estos términos, automatización significa “actuar por sí mismo” [1].

Para que un sistema pueda funcionar por sí solo, este deberá utilizar diferentes tecnologías para monitorear y controlar máquinas y dispositivos, sin intervención humana.

Por lo tanto, la automatización permite reducir y optimizar los procesos de producción actuando directamente sobre la máquina, utilizando elementos de mando y control.

### 1.1.1. Ventajas

Cada vez más empresas apuestan por mejorar los procesos utilizando máquinas automáticas para simplificar las tareas repetitivas. Entonces, la automatización se ha convertido en un beneficio para la empresa que decida invertir en ella; algunos de estos beneficios se muestran a continuación:

- Disminución de probabilidad de accidentes o enfermedades profesionales. Las tareas repetitivas o de alto riesgo realizadas de forma manual por un operador puede ser reemplazadas por máquinas, con el objetivo de proteger al operador.
- Reducción del tiempo de realización de tareas. Las tareas manuales conllevan tiempos elevados. Un sistema automático permite disminuir tiempos de producción mediante la repetición de una tarea con exactitud y eficiencia.
- Mayor productividad: La disminución de la mano de obra representa en la empresa una optimización del trabajo, ya que reduce costos de operación, aumenta ingresos en líneas de producción continuas y, por ende, incrementa la productividad.



### 1.1.2. Desventajas

La automatización representa un avance real en la empresa o industria que decida invertir en ella. Sin embargo, también posee algunas desventajas:

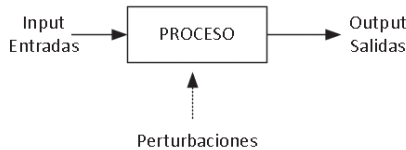
- La mano de obra debe ser calificada debido al mantenimiento y actualizaciones que requiera el sistema, lo que implica costos elevados.
- La dependencia tecnológica también será un factor a considerar, debido a que la empresa se vinculará con el proveedor en cuestiones de desarrollo y mantenimiento.
- La inversión inicial es elevada; además, se debe considerar el valor de depreciación y amortización de la máquina.
- El reemplazo de personal por máquinas puede generar desempleo.

### 1.1.3. Opciones de Automatización

Recordemos que la automatización utiliza tecnologías para monitorear y controlar máquinas y dispositivos. Este monitoreo y control puede ser cableado o eléctrico programado. Ahora bien, para definir y presentar los componentes de la lógica cableada y programada, es necesario conceptualizar algunos términos [2].

- **Mando:** Acción ejecutada sobre un sistema de control para modificar un estado.
- **Control:** Método para manejar el estado de un aparato, máquina o proceso.
- **Contacto:** Elemento que permite formar o interrumpir un circuito.
- **Actuador:** Elemento que actúa directamente sobre el proceso.
- **Sensor:** Elemento que permite medir diversas señales, por ejemplo, presión, temperatura, nivel, presencia.
- **Transmisor:** Capta la variable del proceso por medio del sensor y transmite una señal normalizada a distancia.

En la Fig. 1.2, podemos observar una variable que ingresa a un sistema, proceso o planta con el objetivo de ser transformada o controlada. Existen diferentes alternativas de control, entre las más populares están el control en lazo abierto y el de lazo cerrado.

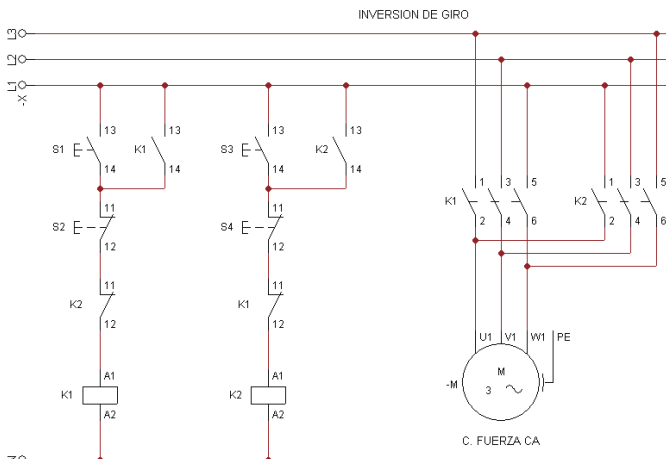


**Figura 1.2:** Control de procesos

Las opciones de automatización de acuerdo con una clasificación tecnológica son:

### *Lógica cableada*

También conocida como lógica de contactos; utiliza circuitos eléctricos, contactos eléctricos, relés eléctricos y demás elementos que serán conectados siguiendo una secuencia lógica de control para su funcionamiento. Se contempla un circuito de mando y un circuito de potencia. Un ejemplo de este tipo de configuración es la que se encuentra en la Fig. 1.3, la cual muestra el diseño del sistema de control para la inversión de giro de un motor eléctrico.

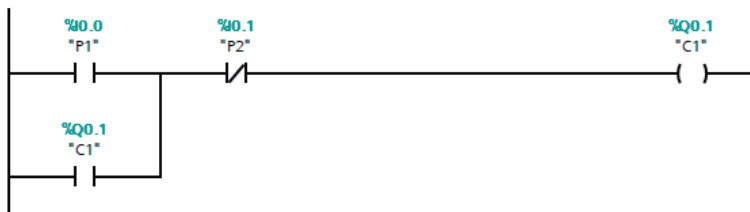


**Figura 1.3:** Circuito de control para la inversión de giro de un motor con lógica cableada

Cualquier cambio que se necesite en el circuito implicará modificar todo el cableado y elementos que forman parte. Este tipo de circuitos se suele utilizar en instalaciones sencillas.

### *Lógica programada*

Este tipo de diseño reemplaza los elementos de mando, tales como relés eléctricos, contactores auxiliares, temporizadores, contadores, etc; por PLC's también conocidos como autómatas o relés programables. Para que pueda funcionar el sistema con lógica programada, se debe crear un software en los lenguajes de programación que manejan los fabricantes de PLC's, basados en las normas de la Comisión Electro-técnica Internacional (IEC). En este caso, cualquier cambio en la programación no necesariamente incluye una modificación en el cableado de las entradas y salidas al PLC [3]. Podemos visualizar un ejemplo de esta lógica en la Fig. 1.4; es un programa en lenguaje Ladder, para que arranque o pare un motor.

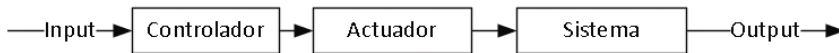


**Figura 1.4:** Programa para arrancar o parar un motor con lógica programada.

Tenemos dos formas de hacer control de procesos: cuando la señal de salida es independiente de la señal de entrada y cuando la señal de salida si influye en la señal de entrada.

### *Control en lazo abierto*

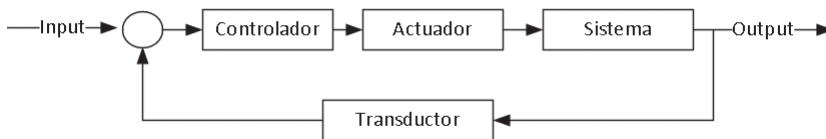
En este sistema, la señal de entrada que ingresa al proceso sufre una transformación para obtener la salida (ver Fig. 1.5).



**Figura 1.5:** Control de procesos en lazo abierto

### *Control en lazo cerrado*

Este es un tipo de control por realimentación. El controlador recibe la señal de error que se produce por la diferencia entre la variable medida y la referencia (*set point*), efectuando una acción de compensación dependiendo de la magnitud del error. Se realiza con el objetivo de llevar la salida del sistema al valor deseado (ver Fig. 1.6)



**Figura 1.6:** Control de procesos en lazo cerrado

## **1.2. Metodología de creación de proyectos de automatización industrial**

### **1.2.1. Gestión del proyecto vs. trabajo del proyecto**

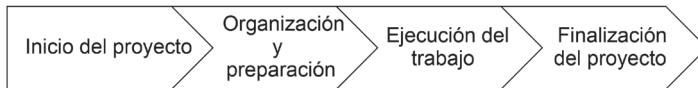
En el diseño de un proyecto de automatización industrial se encuentran involucradas variables de diversas fuentes, por lo que se necesita establecer la diferencia entre la metodología de gestión de proyectos y la de desarrollo de tecnología o planta que conlleva el diseño e implementación del proceso de automatización (ver Fig. 1.7).



**Figura 1.7:** Metodología de gestión vs. metodología de trabajo del proyecto

## 1.2.2. Metodología de gestión del proyecto

Se refiere al ciclo que debe tener la gestión del proyecto con respecto al ciclo de vida (ver Fig. 1.8) que debe cumplir, con las fases de inicio, organización y preparación, ejecución del trabajo y finalización del mismo[4].

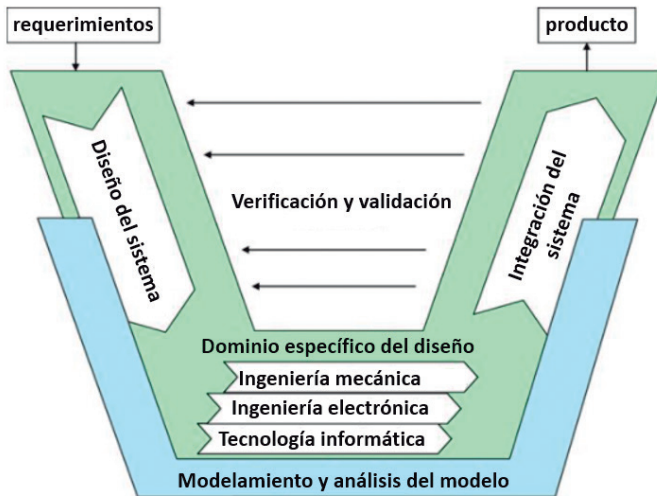


**Figura 1.8:** Fases de la gestión de proyecto

## 1.2.3. Metodología de trabajo del proyecto: Modelo V

En la denominada Cuarta Revolución Industrial o Industria 4.0, la necesidad de adoptar tecnologías emergentes en procesos productivos con un alto grado de automatización es cada vez más evidente. La automatización en plantas industriales se aceleró a partir de la invención del PLC en 1968 [5] en paralelo con otros controladores, cambiando drásticamente los ciclos productivos. En este contexto, la automatización está orientada a la maximización de la productividad y la eficiencia, disminuyendo riesgos y minimizando la intervención humana en procesos productivos. No obstante, su implementación requiere de la aplicación de varias herramientas de hardware y software que conlleva el desarrollo de sistemas mecatrónicos. Así, la mecatrónica integra tecnologías en campos multidisciplinarios de ingeniería —la mecánica y la electrónica junto con ciencias de la computación—, en la búsqueda

de implementación de productos industriales, procesos y operaciones en plantas; para ello, una de las metodologías más utilizada es la denominada Modelo-V.



**Figura 1.9:** Modelo V (VDI Guideline 2206: Design methodology for mechatronic systems) [6]

La segunda metodología se introduce como referencia de desarrollo tecnológico al modelo V (VDI Guideline 2206: Design methodology for mechatronic systems) [7] para el desarrollo de productos mecatrónicos o tecnológicos (ver Fig. 1.9), en los que interviene la ejecución de tareas como:

- Identificación de requerimientos.
- Especificaciones del proyecto de desarrollo tecnológico: se estudian las necesidades, expectativas del cliente de acuerdo con el análisis del caso de negocio y la cultura de la organización, se identifican las variables a controlar en concordancia con los requerimientos de los sistemas mecánicos, eléctricos, electrónicos y la integración informática que se encuentra resumida en el diseño de un sistema de control.

- Diseño y arquitectura del sistema a controlar: con los insumos de las especificaciones, se define el sistema y, de acuerdo con la especialidad del producto, se empieza a desarrollar el proceso mecánicamente; es decir, se establecen los movimientos, la cinemática y la dinámica que debe cumplir el proceso. Luego, se establece el diseño y la arquitectura del sistema a controlar, donde se identifican las entradas y salidas del controlador. De esta manera, se procede a la selección de sensores y actuadores para los accionamientos que requiere integrar el algoritmo de control.
- Implementación de los elementos del sistema o los sistemas, integración de los mismos: se simula el sistema mecánico y el eléctrico-electrónico de acuerdo con las prestaciones proyectadas para optimizar el sistema de la planta prototipo. Una vez optimizada la modelación CAD 3D, se despliegan los planos 2D para su fabricación y ensamble con la tecnología complementaria de potencia y control para la seguridad y autonomía eléctrica, así como de respaldo de la información. Por último, se desarrolla un plan de verificación y validación; para ello, se establece una pieza con los defectos posibles que detonen los sistemas de alerta del sistema. Esta pieza es conocida como la muestra dummy.
- Transición del producto al mercado: una vez validado el producto, se tiene que pasar a su explotación comercial.

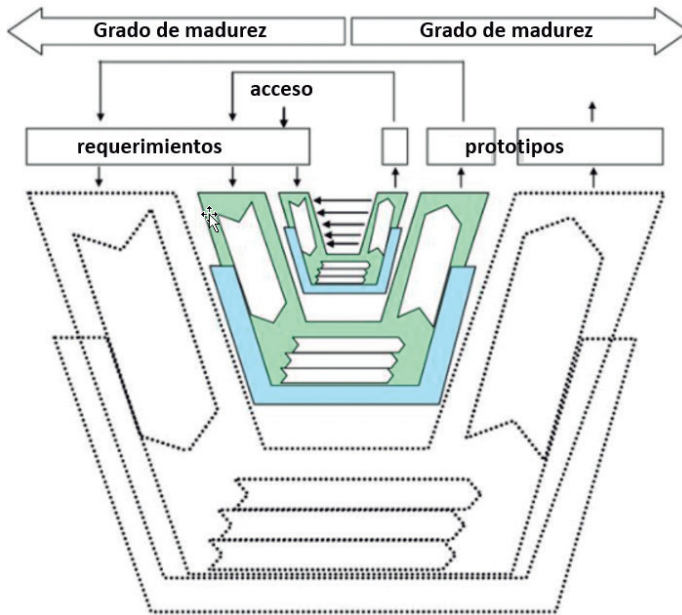
#### **1.2.4. Iteración del modelo V**

El modelo V tiene dos esquemas procedimentales de solución de problemas:

- El ciclo general de resolución de problemas del microciclo.
- El modelo V en macrociclo.

Asimismo, a través de la iteración, se puede mejorar el grado de madurez tecnológica de los prototipos o productos resultantes. Se debe tener en cuenta que esta metodología no es un proceso secuencial, ya que depende de la retroalimentación de cada etapa por la verificación

de los requisitos con los resultados y validación de los requerimientos con las prestaciones de producto [8] (ver Fig. 1.10).



**Figura 1.10:** Macro ciclos del Modelo V [6]

En los procesos industriales, las operaciones tienen una secuencia lógica que establece la relación entre entradas y salidas. Cuando estas no tienen concordancia con los resultados esperados, hay 3 componentes en base a los que se determina un problema:

- Un estado inicial no satisfactorio.
- Un objetivo deseable.
- Obstáculos que impiden la transformación de un estado inicial no satisfactorio a un objetivo deseable.

De esta manera, surge la necesidad de mejorar u optimizar el proceso, a través de la solución del problema. Una vez que este ha sido



identificado, se debe emplear creatividad en la toma de decisiones mientras se sistematiza la solución. Para resolver el problema, se deben tener en cuenta las siguientes actividades:

- Reconocimiento de dependencias: el problema se debe descomponer en subproblemas hasta identificar actividades esenciales que le permitan desarrollar el producto.
- Estimar la importancia y urgencia: con la identificación de la dependencia, se deben priorizar las actividades, sobre todo aquellas que se relacionan con las paradas de producción.
- Persistencia y flexibilidad: *persistencia* significa un enfoque continuo en alcanzar las metas. *Flexibilidad* implica la habilidad de adaptación al cambio.

### 1.3. Automatas programables

Un autómata programable es un dispositivo electrónico que se utiliza dentro de entornos industriales para el control, monitoreo y registro de procesos industriales, con una alta capacidad de procesamiento, limitada memoria de datos y programa, recursos de comunicación y programación acorde con el desarrollo tecnológico, que puede ser programado en lenguajes no informáticos.

Tiene su origen ante el requerimiento de la empresa General Motors a sus proveedores de equipamiento eléctrico en 1968. Este especificó su necesidad de reducción de costos de mantenimiento e implementación de la automatización de sus procesos basados en lógica cableada, utilizando relés y contactores, al igual que disminuir el espacio que estos sistemas ocupaban dentro de sus instalaciones, estableciendo la necesidad de un sistema flexible, apto para el entorno industrial, de fácil programación, mantenimiento reducido y reutilizable de forma que se reduzca el tiempo de inactividad de la máquina ante

cambios en la configuración del proceso y con capacidad de expansión a futuro. Así inició el desarrollo de los Controladores Lógicos Programables (PLC's) que tienen un espacio ganado dentro de la automatización de procesos y la industria.

En la actualidad, constituyen un recurso prácticamente indispensable en un proceso industrial que se pueda considerar automatizado, ya que su gran potencial de reprogramación y adaptabilidad a nuevos requerimientos en el desarrollo de productos cada vez más personalizados los ha constituido un recurso altamente requerido en el escenario de la Industria 4.0.

Sus recursos y capacidades han ido evolucionando en paralelo con el desarrollo tanto de la electrónica como de las telecomunicaciones, de tal forma que, actualmente, ya no es solo un dispositivo de trabajo dentro de entorno industrial cerrado y limitado, sino que se los puede conectar a dispositivos comerciales y de oficina. Esto permite la comunicación entre la parte productiva y la administrativa de una empresa, siempre y cuando esté acompañado de recursos de conectividad, buses y redes de datos tanto a nivel industrial como administrativo.

IEC 61131 es el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo lenguajes de programación. Esta norma se divide en cinco partes:

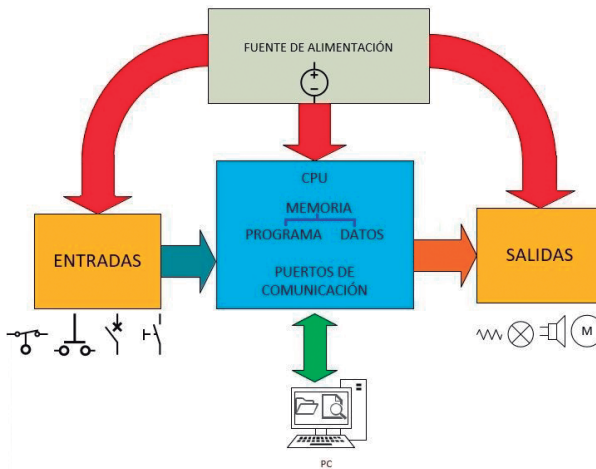
- Parte 1: Vista general
- Parte 2: Hardware
- Parte 3: Lenguaje de programación
- Parte 4: Guías de usuario
- Parte 5: Comunicación

IEC 61131-3 es la base real para estandarizar los lenguajes de programación en la automatización industrial, haciendo el trabajo independiente de cualquier marca o modelo, lo que ha ayudado a la estandarización y facilitado el entendimiento de programas desarrollados en diferentes marcas y modelos de PLC's, ya que la simbología y método de programación está regulado por la norma.

### 1.3.1 Arquitectura de PLC

Su arquitectura interna es muy similar a la de un computador, es decir, tiene básicamente un cerebro central basado en un microprocesador o microcontrolador, memoria de datos, memoria de programa, entradas, salidas y puertos de comunicación, pudiendo estar todo esto dentro de un solo bloque que constituye la CPU (Unidad Central de Procesamiento), sin embargo en algunos casos las salidas, entradas y otras funcionalidades pueden adicionarse externamente, a través de los denominados módulos de expansión, que pueden ser de diferentes tipos como digitales, analógicos y de comunicación. (Ver Fig. 1.11).

- Unidad Central de Procesamiento: es el elemento basado en microprocesador o microcontrolador que gestiona y maneja el funcionamiento del PLC; es el encargado de ejecutar el programa que se almacena en la memoria y conectarse con las entradas, salidas y dispositivos conectados al PLC a través de los puertos de comunicación.
- Memoria de datos: es el espacio destinado al almacenamiento de la información resultante de la lectura de las entradas, escritura en las salidas y los resultados parciales de las operaciones lógicas o aritméticas que realiza el programa del PLC. Básicamente, puede almacenar tipos de datos, como bit, byte (8 bits), Word (16 bits), double Word (32 bits), real (flotante), timers, contadores. Anteriormente, se dividía en carpetas o tipos que definían el tipo de dato que podía ser almacenado en un bloque determinado. Actualmente la memoria de datos es de propósito general y puede utilizarse libremente para almacenar los tipos de datos aceptados por el PLC.
- Memoria de programa: es el encargado de almacenar el código del programa que rige la lógica de funcionamiento del PLC; está compuesta de un bloque principal denominado Main y se le pueden adicionar tipos de bloques para ayudar a dividir el funcionamiento en subrutinas, funciones o bloques de datos.



**Figura 1.11:** Arquitectura PLC

- **Entradas:** comunican el estado de elementos de medición, pulsadores, interruptores, sensores o cualquier elemento que pueda proporcionar información del mundo exterior al PLC, requieren una conexión física con el elemento a medir y puede ser digital (prendido o apagado) o analógica, determinada por una resolución y destinada a la conexión de sensores analógicos como termocuplas, celdas de carga, galgas extensiométricas, potenciómetros, entre otros. Internamente, cuentan con protecciones optoeléctricas que las aíslan para cuidar al resto del equipo.
- **Salidas:** se comunican y se conectan con los actuadores, destinados a controlar o modificar el estado de una variable a controlar, como motores, niquelinas, lámparas, luces piloto, electroválvulas, entre otros. Al igual que las entradas, pueden ser digitales (prendido o apagado) o analógicas. Internamente, cuentan con protecciones optoeléctricas que las aíslan para resguardar al resto del equipo.
- **Puertos de comunicación:** por lo general en la actualidad todos los PLC ofrecen diferentes puertos de comunicación que facilitan la comunicación con los entornos de programación, que por lo general se instalan en PC's, siendo en la actualidad el puerto

Ethernet es comúnmente más disponible en los PLC, el cual ha desplazado a los puertos seriales RS-232 o RS-485 que eran los comúnmente disponibles en 20 modelos anteriores.

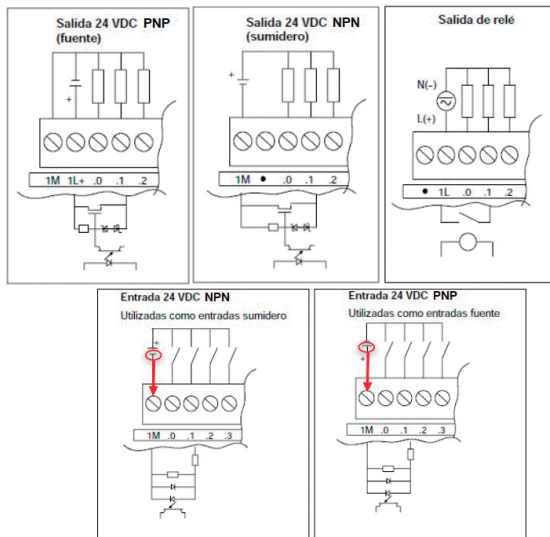
### *Tipos de I/O*

El PLC puede recibir o entregar dos tipos de señales desde o al exterior. Estas son señales digitales binarias, es decir 1 o 0 y señales analógicas dentro de un rango de voltaje (0 a 10 V) o corriente (4 a 20 mA). Sin embargo, la forma como se recibe o entrega esta señal del o al exterior puede ser diferente y depende del tipo de elemento o actuador que se quiere manipular y el módulo con el que se va a trabajar. Se debe considerar que el módulo requiere de una conexión a la fuente de alimentación y que internamente proporciona un sistema de aislamiento galvánico, acondicionamiento, filtrado y protección que evita problemas de ruido, conexión o señales inadecuadas tanto en las entradas o salidas del PLC afecten al CPU.

En cuanto a las entradas digitales, pueden ser:

- Tipo voltaje continuo PNP o fuente, por el terminal común del módulo recibe al terminal negativo de la fuente de alimentación (0 VCC) y el terminal positivo (24 VCC) se recibe a través del elemento externo conectado (interruptor, sensor, pulsador o cualquier dispositivo que genere la diferencia de potencial necesaria) al terminal de la entrada del módulo. Si bien es cierto que 24 VCC es el voltaje más común de las entradas, también se encuentran voltajes de 12 o 50 VCC. Ver Fig. 1.12.
- Tipo voltaje continuo NPN o sumidero, por el terminal común del módulo recibe al terminal positivo de la fuente de alimentación (24 VCC), y el terminal negativo (0 VCC) se recibe a través del elemento externo conectado (interruptor, sensor, pulsador o cualquier dispositivo que genere la diferencia de potencial necesaria) al terminal de la entrada del módulo. Ver Fig. 1.12.

- Tipo corriente (4 o 20 mA), por el terminal común del módulo recibe al terminal negativo de la fuente de alimentación (0 VCC) y la terminal de la entrada del módulo recibe la corriente de 4 mA para el 0 lógico o 20 mA para el 1 lógico. Este tipo de conexión es recomendado para distancias relativamente largas entre el módulo y el dispositivo de entrada; adicionalmente, se puede determinar problemas de conexión con el desmotivo, cuando la señal sea de 0mA.
- Tipo voltaje alterno, por el terminal común del módulo recibe al terminal neutro de la fuente de alimentación y la fase se recibe a través del elemento externo conectado (interruptor, sensor, pulsador o cualquier dispositivo que genere la diferencia de potencial necesaria) al terminal de la entrada del módulo. Se pueden encontrar en voltajes de 24, 48, 110 o 220 VAC.
- Adicionalmente, existen módulos de entradas especiales, como contadores rápidos que se utilizan para sensores que proporcionan señales de alta frecuencia como encoders.



**Figura 1.12:** Tipos de entradas y salidas de PLC's [9]

En cuanto a las entradas analógicas, pueden ser: tipo voltaje en rangos de 0 a 10 V o -10 a 10 V o corriente de 4 a 20 mA. Transforman internamente las señales de sus entradas a valores digitales, a través de conversores análogos digitales, por lo que una de las principales características de este tipo de entradas es su resolución, que puede ser de 10, 12 o 16 bits (mientras mayor es el número de bits de la resolución, mayor exactitud de la medición, ya que el mínimo cambio que puede ser detectado por el módulo es menor). Por lo general, entregan su valor en un Word en un valor entero o flotante. También existen módulos de entradas analógicas especiales que se utilizan para conectar sensores específicos, ya que ofrecen el acondicionamiento necesario para trabajar con ese tipo de sensor, eliminando el trabajo de acondicionamiento, filtrado y transmisión de señales, como termocuplas o celdas de carga.

Con las entradas digitales, suele ser necesario conectar el cable que lleva la señal negativa (o 0 V) al módulo donde se encuentran las entradas, ya que desde estas se introduce el polo positivo e, internamente, es necesaria la diferencia de potencial para activarlas. Todo esto siempre y cuando se esté trabajando con lógica positiva.

En cuanto a las salidas digitales, pueden ser:

- Tipo voltaje continuo PNP colector abierto, por el terminal común del módulo recibe al terminal negativo de la fuente de alimentación (0 VCC) y al terminal positivo (24 VCC) transmite el voltaje al elemento externo conectado (luz piloto, motor, electroválvula, relé o cualquier otro tipo de salida) al terminal de salida del módulo. Ver Fig. 1.12.
- Tipo voltaje continuo NPN colector abierto, por el terminal común del módulo recibe al terminal positivo de la fuente de alimentación (24 VCC) y al terminal negativo (0 VCC), transmite el voltaje al elemento externo conectado (luz piloto, motor, electroválvula, relé o cualquier otro tipo de salida) al terminal de salida del módulo. Ver Fig. 1.12.

- Tipo corriente (4 o 20 mA), por el terminal común del módulo recibe al terminal negativo de la fuente de alimentación (0 VCC) y en el terminal de la salida del módulo entrega la corriente respectiva de 4 mA para el 0 lógico o 20 mA para el 1 lógico. Este tipo de conexión es recomendado para distancias relativamente largas entre el módulo y el dispositivo de salida; adicionalmente, se pueden determinar problemas de conexión con el desmotivo, cuando la señal sea de 0 mA en el elemento.
- Tipo voltaje alterno TRIAC, por el terminal común del módulo recibe al terminal neutro de la fuente de alimentación y la fase se proporciona a través del terminal de salida al elemento externo conectado (motor, relé, lámpara, etc.) al terminal de salida del módulo.
- Tipo relé, es el más utilizado. Puede trabajar tanto con señales AC o DC; por el terminal común del módulo recibe un terminal de la fuente y por el de salida, cierra el circuito. Por lo general, tiene un solo contacto normalmente abierto, aunque existen PLC's que proveen un contacto normalmente abierto y otro cerrado. Sus desventajas son el tiempo de respuesta: por ser electro-mecánico, le toma más tiempo la conmutación respecto a los NPN, PNP o TRIAC y el desgaste de los contactos puede reducir su vida útil. Ver Fig. 1.12.
- Adicionalmente, existen módulos de salidas especiales como modulación por ancho de pulsos (PWM) que se utilizan para elementos como motores.

En cuanto a las salidas analógicas, pueden ser: tipo voltaje en rangos de 0 a 10 V o -10 a 10 V o corriente de 4 a 20 mA. Transforman internamente los valores digitales de sus registros internos en valores de voltaje analógico en sus salidas, a través de conversores digitales analógicos. Por ello, una de las principales características de este tipo de salidas es su resolución, encontrándose de 10, 12 o 16 bits.

Indistintamente del tipo de entrada o salida con el que se trabaje, se debe tomar en cuenta que las características de los elementos



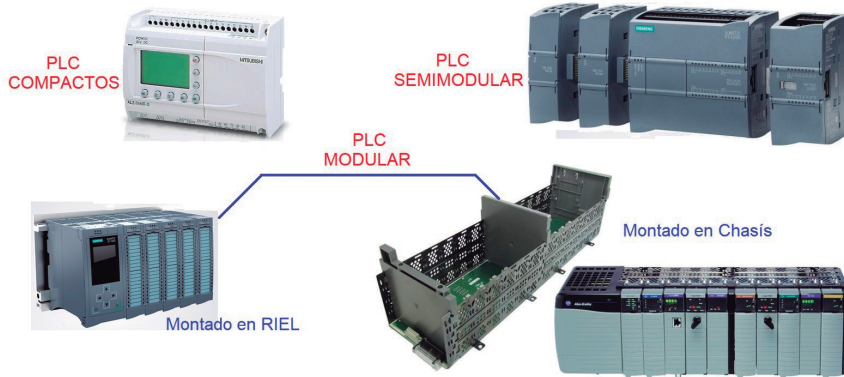
a conectar no superen las del módulo (corriente, voltaje, potencia o impedancia), para evitar problemas o daños y garantizar la vida útil de los módulos de entrada o salida y el correcto funcionamiento de los elementos conectados. Por ello, se recomienda verificar la hoja técnica de los módulos antes de conectar o utilizarlos inadecuadamente.

### 1.3.2. Tipos de PLC

Existen criterios a partir de los cuales se pueden clasificar los PLC's. Se mencionarán las más conocidas.

Por su estructura y apariencia física, el PLC puede ser:

- **Compacto:** un solo bloque contiene todos los elementos necesarios para su funcionamiento (Fuente de alimentación, CPU, memorias y E/S); son más económicos pero limitados en sus características, especialmente en el número de entradas y salidas.
- **Modular:** formado a partir de la combinación de módulos por separado, incluyendo la CPU, la fuente de alimentación y módulos de expansión de comunicación o E/S. Puede estar insertado dentro de una sola estructura que lo protege y almacena que se denomina chasis rack o agrupado únicamente a través de una riel común y conectores que unen a todos los módulos ajustados a través de tornillos. Es más flexible y adaptable a los requerimientos del proceso, ya que puede ir expandiendo sus módulos según las variantes de un proceso, sin embargo es más costoso y tiene un límite definido de módulos que puede admitir.
- **Semimodular:** tiene la apariencia de un PLC compacto, pero admite módulos de expansión en menor cantidad que un PLC modular, lo que lo hace versátil aunque limitado a aplicaciones pequeñas o medianas. En la actualidad, muchas marcas y familias de PLC's tienen esta característica de poseer entradas y salidas incluidas y admitir módulos de expansión; por ejemplo, la familia de PLC S7-1200 de Siemens (ver Fig. 1.13).



**Figura 1.13:** Clasificación de los PLC's por su estructura

Por sus características internas —es decir, por sus capacidades y prestaciones, asociado con que el PLC es un dispositivo de propósito general— se debe considerar que no todas las aplicaciones son iguales. Por lo tanto, se debe seleccionar un PLC que se ajuste a los requerimientos del proceso, ya que el incremento de capacidades del PLC está asociado con su costo. Así, el PLC puede ser:

- Gama baja: está formado por PLC's con capacidades muy limitadas de memoria de datos, programa, entradas y salidas que, dependiendo de la nomenclatura y clasificación del fabricante, en algunos casos no se considera como PLC, sino más bien un relé inteligente. Por lo general, son compactos o semicompactos; en Siemens está constituido en la familia conocida como Logo.
- Gama media: son PLC's con capacidades de memoria de datos y memoria, entradas, salidas y módulos de expansión que los hacen aptos para aplicaciones medianas y grandes. Tienen una gran funcionalidad, pero sus capacidades internas son limitadas. Pueden ser semicompactos o modulares en racks; en Siemens está constituida por la familia S7-1200 o S7-300.
- Gama alta: estos PLC's son de altas prestaciones y grandes capacidades de almacenamiento de datos y programas. Por general, se

los utiliza para manejar procesos muy grandes y comunicarse con otros PLC's dentro de una red industrial; mayormente son modulares a través de rieles y conectores y de alto costo. En Siemens está constituido por la familia S7-1500 o S7-400. Ver Fig. 1.14.



**Figura 1.14:** Clasificación de los PLC's por sus capacidades y prestaciones

### 1.3.3. Lenguajes de programación

Si bien los lenguajes de programación han sido regulados por la norma IEC 61131-3, los diferentes fabricantes de PLC ofrecen cada vez más sofisticados entornos de programación para la creación de proyectos de automatización. Por ejemplo, Siemens tiene TIA Portal; Rockwell Automation tiene Studio 5000; entre otros. Con TIA Portal, los programadores pueden seleccionar entre diferentes lenguajes de programación como: diagrama de escalera (Ladder LAD), el diagrama de bloques de funciones (FBD), lenguaje de programación estructurado (SLC), lenguaje de programación gráfico (GRAFSET), entre otros, dependiendo del modelo y familia del PLC. Las características de todo entorno actual de programación de PLC incluyen capacidades de creación de proyectos, adición de módulos de expansión, diagnóstico del sistema, diagnóstico de procesos, simulación de programas, configuración de recursos de comunicación, documentación de los proyectos de automatización, conexión con otros dispositivos como PLC's y HMI's.

S7-PLCSIM es un paquete opcional para TIA Portal que permite la simulación de una plataforma de control SIMATIC S7 y la simulación y comprobación de un programa de usuario en una PC. Esto permite probar y perfeccionarla antes de la instalación física del hardware. Al realizar pruebas al inicio del desarrollo de un proyecto, se puede mejorar la calidad general. De este modo, la instalación y la puesta en servicio son más rápidas y con menos costos, ya que las fallas del programa se pueden detectar y corregir desde las fases iniciales, durante el desarrollo e incluso en el funcionamiento [10].

- Diagrama de escalera (Ladder LD): tiene sus orígenes en Estados Unidos. Está basado en la presentación gráfica de la lógica de relés, está asociado con la simbología y lógica de diseño de sistemas basados en relés y contactores.
- El Diagrama de Bloques Funcionales (FBD): es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa. Se asemeja a la implementación de circuito digital basado en el uso de compuertas digitales y flip-flops.
- El lenguaje Texto estructurado (SLC ST): es un lenguaje de alto nivel con orígenes en Pascal o C. Puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (REPEAT-UNTIL; WHILE-DO), ejecución condicional (IF-THEN-ELSE; CASE), funciones (SQRT, SIN, etc.); adicionalmente, se puede considerar como la opción de programación textual basada en lenguaje de alto nivel.
- Lenguaje de programación gráfica (GRAPH SFC), basada en GRAFCET, facilita la implementación de programación secuencial de procesos. Dependiendo de su complejidad, existe un aumento en la implementación de este tipo de programación, porque permite mantener un orden y secuencia ordenada de la

lógica del proceso; sin embargo, no está disponible en todos los modelos de PLC's. Está principalmente orientado a PLC's gama alta, en los cuales se programan procesos más complejos.

- Existen varios lenguajes adicionales, pero, con el tiempo y el desarrollo tecnológico, han perdido vigencia y se encuentran disponibles en menor cantidad de PLC's. Por lo tanto, son menos utilizados como lista de comandos (IL). Ver Fig. 1.15.

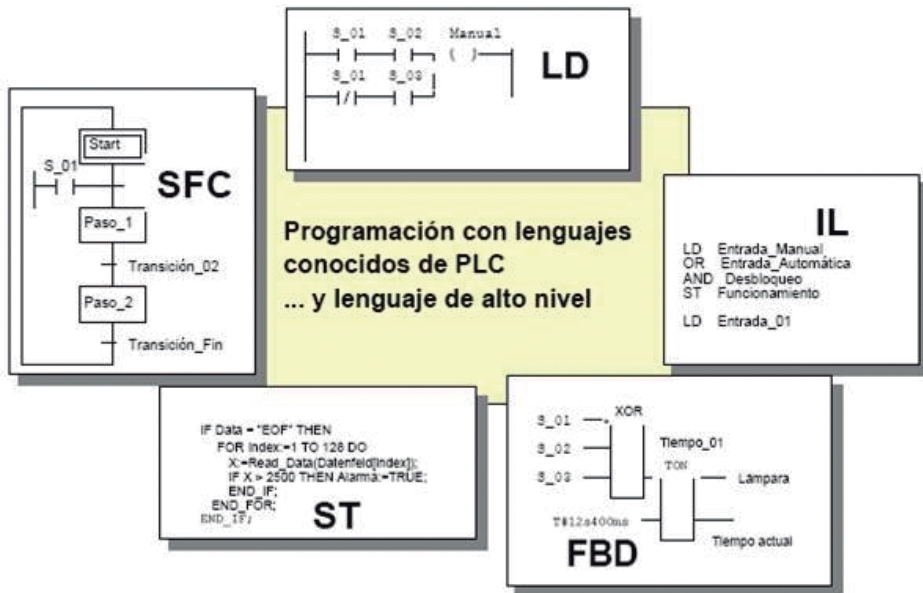


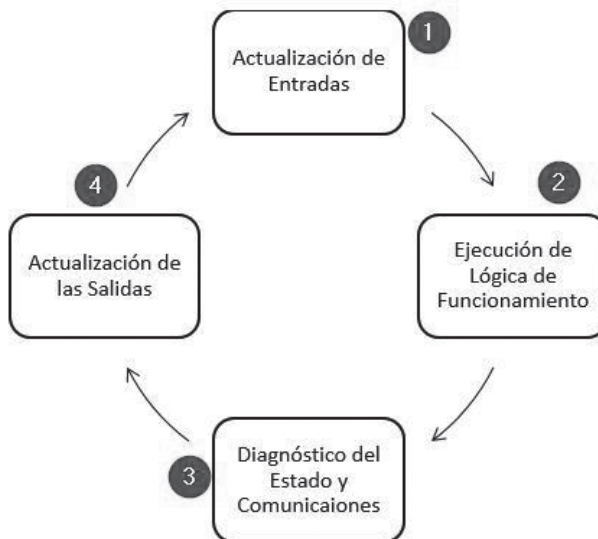
Figura 1.15: Lenguajes de Programación de PLC's [11]

### 1.3.4. Funcionamiento

El funcionamiento del PLC es secuencial. Por ello, la ejecución de la lógica de funcionamiento tiene 4 pasos que se ejecutan de manera cíclica y continua.

- Actualización de entradas: consiste en la lectura de los datos encontrados en las entradas del PLC y en su transferencia a las memorias internas que el PLC tiene designadas para este propósito.

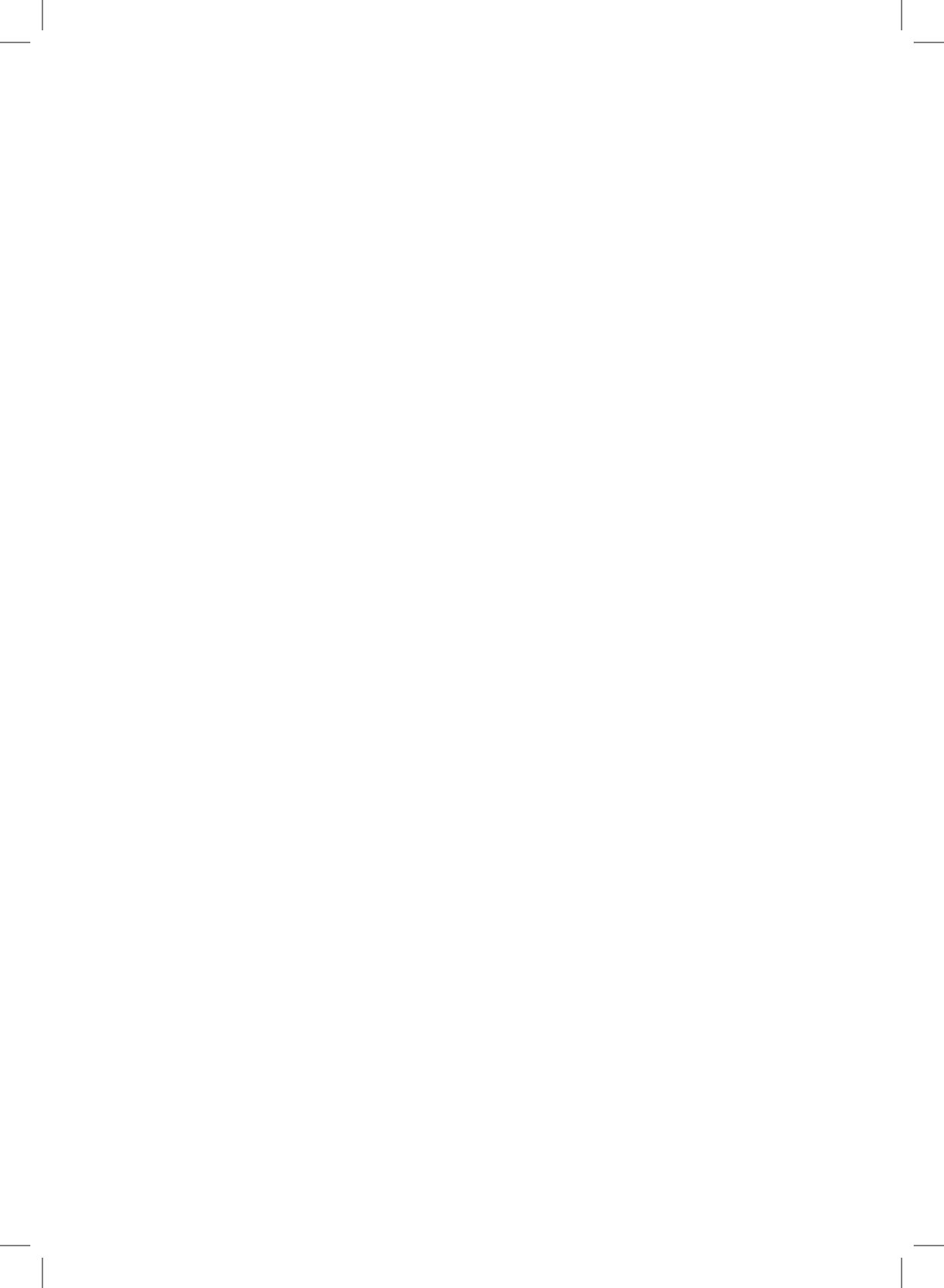
- Ejecución de la lógica de funcionamiento: comando por comando programado, se ejecuta la lógica de funcionamiento establecida y se determinan los nuevos valores a ser cargados en las salidas según los cambios de las entradas leídas anteriormente. En el caso de Ladder y Bloques de funciones, la ejecución se da de izquierda a derecha y de arriba hacia abajo.
- Diagnóstico del estado del PLC y comunicaciones: se realiza el análisis del estado del PLC y se atienden los pedidos de información o instrucciones enviadas al PLC desde los puertos de comunicación. Este proceso es fundamental para el sistema ya que es el que permite comunicarse con el entorno de programación y la recepción de datos a través de redes o buses de campo o control.
- Actualización de las salidas: se escriben a través de los registros de memoria establecidos en los estados de las salidas del PLC convertidas en los niveles y características acorde al tipo de salida respectiva. Ver Fig. 1.16.



**Figura 1.16:** Funcionamiento del PLC

## 1.4. Preguntas de repaso

1. Analice. ¿Cómo influencia la automatización de procesos industriales en el desempleo?
2. ¿Qué efectos tiene la automatización sobre los trabajadores y el mercado laboral?
3. Realice una investigación exploratoria respecto al grado de automatización del sector industrial en el país.
4. ¿Qué sugerencias puede dar para armonizar las actividades entre máquinas y trabajadores?
5. ¿Cuáles son las diferencias entre lógica cableada y lógica programada?
6. ¿Qué elementos deben tomarse en cuenta para automatizar un proceso?
7. ¿Qué criterios se deben tomar en cuenta para seleccionar el tipo de PLC a utilizar en la implementación de un proceso automático?
8. ¿Qué elementos se pueden conectar a la entrada de un PLC?
9. ¿Qué elementos se pueden conectar a la salida de un PLC?
10. ¿Qué consideraciones se debe tener en cuenta el momento de seleccionar el lenguaje de programación del PLC?



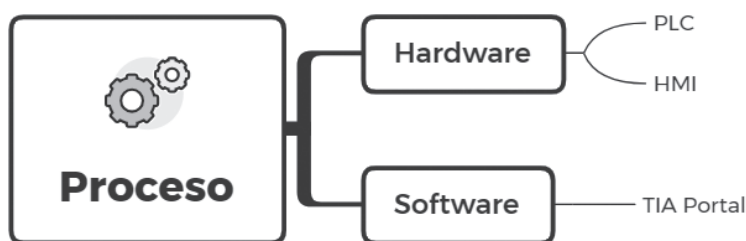


**Capítulo 2**  
**Creación de Proyectos**  
**con TIA Portal**



En este capítulo se introducirá el uso del software Totally Integrated Automation Portal (TIA Portal). Este trabaja con herramientas que interactúan entre sí con la ventaja de tener todo integrado en una misma plataforma sin necesidad de utilizar diferentes programas. TIA Portal proporciona una alternativa de automatización eficiente con una gestión conjunta de datos, interfaz gráfica para configuración, facilidad de configuración, para crear soluciones de automatización etc. [12].

## 2.1. Introducción



**Figura 2.1:** Soluciones de automatización

La automatización de un proceso involucra la selección del hardware de control (PLC) y visualización (HMI), así como su programación, como se muestra en la Fig. 2.1. Los PLC, de acuerdo con su marca y gama, proporcionan a los usuarios el software necesario para la programación. Es así que, a lo largo de este capítulo, nos centraremos en el manejo del PLC de marca Siemens S7-1200 para lo cual detallaremos el proceso de programación en el software TIA Portal [12]. Este permite configurar tanto el controlador como la interfaz de visualización en un único proyecto. Los componentes STEP 7 para programación del

PLC y WinCC para la configuración de la visualización en el HMI son editores que acceden a una base de datos común de información. Con lo cual, a través de una interfaz de usuario común se accede a todas las funciones de programación y visualización.

### 2.1.1. Requisitos mínimos

Para una correcta instalación del Software de TIA Portal V15.1, el computador debe cumplir con requisitos listados en la Tabla 2.1. El uso de máquina virtual no es aconsejable por la cantidad de recursos de memoria que requiere la aplicación.

**Tabla 2.1:**  
Requisitos mínimos de instalación de TIA Portal V15.1. Tomado de [12]

Hardware y software	Requerimiento
Procesador	Intel Core i5-3320 M 3,3 GHz o superior
RAM	Mínimo 8 GB
Disco duro	SSD con al menos 50 GB de espacio libre de almacenamiento
Sistema operativo (64 bits)	<ul style="list-style-type: none"> <li>• Microsoft Windows 7 Home Premium SP1 o superior</li> <li>• Microsoft Windows 7 o superior (Professional SP1, Enterprise SP1, Ultimate SP1)</li> <li>• Microsoft Windows 8.1 Microsoft Windows 8.1 (Professional, Enterprise)</li> <li>• Microsoft Server 2008 R2 StdE SP1 (solo STEP 7 Professional)</li> <li>• Microsoft Server 2012 R2 StdE</li> </ul>
Tarjeta gráfica	32 MB RAM
Red	Ethernet de 10 Mbits/s

### 2.1.2. Entorno de trabajo

Cuando se crea un nuevo proyecto, el entorno de trabajo de TIA Portal proporciona dos tipos de vistas:

- Vista de portal: aquí se puede realizar la configuración de las herramientas orientadas a tareas del proyecto. Las funciones desplegadas dependerán de los productos instalados.
- Vista de proyecto: entorno en el cual se realiza la configuración de los objetos y componentes de un proyecto. La Fig. 2.2 detalla los elementos que conforman la vista.

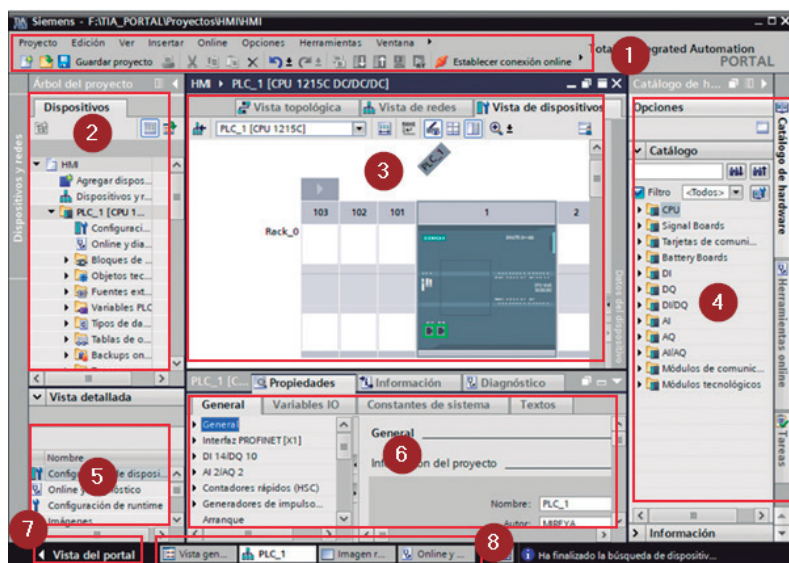


Figura 2.2: Vista de proyecto

- ① Menú y barra de herramientas.
- ② Árbol de proyecto: a través de este, se puede acceder a todos los componentes y datos del proyecto.
- ③ Área de trabajo.
- ④ Task Card: cambia en función del objeto seleccionado.
- ⑤ Vista detallada: despliega componentes del objeto seleccionado en el árbol del proyecto.
- ⑥ Ventana de inspección.
- ⑦ Cambio a la vista de portal.
- ⑧ Barra del editor.

## 2.2. Tipo de datos

Las instrucciones que ofrece TIA Portal requieren parámetros que se definen a partir del uso de variables. El tipo de dato asignado a las variables determina sus propiedades; por ejemplo, el formato, área de memoria asignada en la CPU, entre otros. Un parámetro puede aceptar más de un tipo de dato. Las variables a ser asignadas, en general, se crean en la pestaña *Variables PLC* del árbol del proyecto o directamente en la interfaz de los bloques del programa.

Existen variables predefinidas que corresponden a las entradas (I), salidas (Q) y las marcas (M). Sin embargo, el usuario puede definir sus propios tipos de datos —que se crean a partir de bloques de datos globales (DB)—. Como caso particular, los tipos de datos String, Struct, Array y Data-Time-Long (DTL) se pueden crear únicamente en un DB. TIA Portal puede definir los siguientes tipos de variables:

- Variables PLC: pueden ser utilizadas en
  - Bloques de organización (OB)
  - Bloques de función (FB)
  - Función (FC)

No pueden existir dos variables con el mismo nombre; su definición puede darse en el mismo editor de programas o en la tabla de variables PLC.

- Variables DB de bloque de datos globales: todos los bloques pueden hacer uso de este tipo de variables. Su definición se realiza en la tabla de declaración del bloque de datos global.
- Variables DB de bloques de datos de instancia: son de carácter local y se utilizan dentro del bloque donde se encuentran definidas.

Para definir una variable PLC se requieren tres campos: nombre, tipo de datos y dirección. El tipo de dato (por ejemplo, Bool, Int, Real, etc.) define el rango de valores admisibles y su formato de representación.

La dirección de una variable especifica el área de memoria utilizada para almacenar su valor. Se puede definir dirección de variables para entradas físicas (I), salidas físicas (Q) y marcas (M).

En la Fig. 2.3 se muestra cómo se relaciona la tabla de variables con el programa de usuario, definiendo variables correspondientes a entradas físicas (I0.0) y salidas físicas (Q0.0), así como la localidad de memoria asignada a la marca tipo bit (M0.1).

### 2.2.1. Tipo de datos binarios y secuencia de bits

Las marcas son variables internas globales empleadas para almacenar datos; en otras palabras, no tienen relación con las entradas o salidas físicas del PLC. Estas ocupan los espacios libres de la memoria temporal de la CPU. Por lo cual, para cada modelo de PLC´s se debe consultar la disponibilidad de direccionamiento máximo.

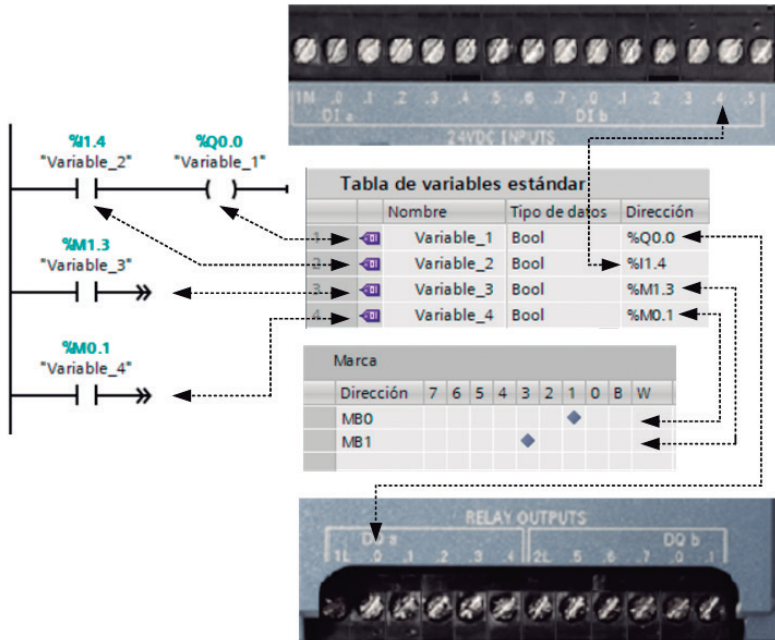
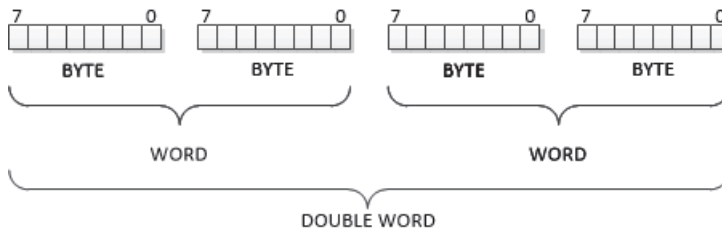


Figura 2.3: Tabla de variables. Modificado de [13]



**Figura 2.4:** Equivalencias de BYTE, WORD y DOUBLE WORD

Para la serie 1200 CPU 1215C se dispone de 8191 direcciones de memoria. Se debe tomar en cuenta que cuando el CPU del PLC inicia su arranque y ejecución, el estado de las marcas es nulo. Así también, un aspecto importante es la longitud de las variables declaradas para evitar que en las direcciones de memoria no exista solapamiento, es decir no sobrescribir en direcciones ya ocupadas. La Tabla 2.2 muestra los tipos de marcas permitidos, con ejemplos de asignación sin solapamiento. La equivalencia entre tipos de datos y número de bits se muestra en la Fig. 2.4.

**Tabla 2.2:** Tipos de datos binarios y secuencias de bits

Tipo	Tamaño (bits)	Ejemplo
BOOL	1	M5.3
BYTE	8	MB10 MB11 MB12 MB13
WORD	16	MW14 MW16 MW18
DOUBLE WORD	32	MD20 MD24 MD28



En el caso particular de las marcas tipo bit, su formato está dado por la dirección de memoria (0 a 8191) seguido por la posición del bit (de 0 a 7) separados por punto, como se muestra en la Fig. 2.5.

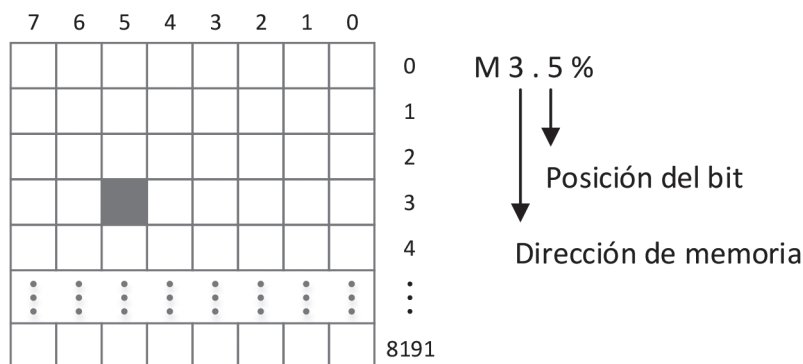


Figura 2.5: Memoria interna PLC S7-1200

## 2.2.2. Tipos de datos matemáticos

Para realizar operaciones matemáticas mediante las instrucciones disponibles en TIA Portal, se debe tener en cuenta el tipo de datos así como su rango para configurar la instrucción. La Tabla 2.3 muestra los tipos de datos más utilizados para realizar operaciones matemáticas en la serie 1200. En el caso de realizar una asignación incorrecta, la etiqueta declarada se mostrará en rojo, lo que indica que debe ser modificada.

Tabla 2.3: Tipos de datos matemáticos

Tipos de datos		Longitud (bits)	Tipos de datos		Longitud (bits)
Enteros	SINT	8	Coma flotante	REAL	32
	INT	16		LREAL	64
	DINT	32			
	USINT	8			
	UINT	16			
	UDINT	32			

## 2.2.2. Tipo de datos de tiempo

Son utilizados para definir configuraciones de temporizadores y de fecha y hora utilizando el real time clock (RTC). Por defecto, un operador tipo TIME está dado en milisegundos. Para declarar una variable, se pueden usar los sufijos días (d), horas (h), minutos (m), segundos (s) y milisegundos (ms). En la Tabla 2.4 se listan los tipos de datos válidos. Los límites o rangos permitidos para el tipo de datos DTL no podrán exceder a los mostrados en la Tabla 2.5.

**Tabla 2.4:** Tipos de datos de tiempo

Tipo de datos	Longitud (bits)	Ejemplo
TIME	32	T#14d_20h_30m_20s
DATE	16	D#2020-08-14
TOD (TIME OF DAY)	32	TOD#14:33:20.500
DTL	12 bytes	DTL#2020-08-15-14:30:20.200

**Tabla 2.5:** Tipos de datos DTL

Dato	Rango
Año	1970 a 2554
Mes	1 a 12
Día	1-31
Día de la semana	De 1 (domingo) a 7 (sábado)
Hora	0 a 23
Minuto	0 a 59
Segundo	0 a 59
Nanosegundo	0 a 999999999

## 2.3. Creación de proyectos - ejemplo de aplicación

En esta sección, se describe los pasos para crear, configurar y simular un proyecto con un ejercicio de aplicación simple, como un circuito de marcha-paro. A partir del mismo ejemplo en la sección 2.4, se realizará una interfaz HMI de visualización y se describirán los pasos para descargar los programas del PLC y HMI en el hardware correspondiente.

### 2.3.1. Crear un proyecto

**Tabla 2.6:** Características técnicas del PLC S7-1200 CPU 1215C 6ES7215-1AG40-0XB0. Tomado de [14]

Características S7 1200 CPU 1215C DC\DC\DC	
Tensión de alimentación	24 $V_{DC}$
Número de entradas digitales	14 (24 $V_{DC}$ )
Número de salidas digitales	10 (24 $V_{DC}$ )
Frecuencia de conmutación de las salidas	100 kHz
Número de entradas analógicas	2 (0 a +10 $V_{DC}$ )
Número de salidas analógicas	2 (0 a 20 mA)
Tamaño de área de marcas (número máximo)	8 Kbyte
Resolución del convertor (bits incluido signo)	10 bits
Tipo de interfaz	PROFINET

Los pasos para crear un proyecto se listan a continuación:

1. Al abrir el software TIA Portal, se despliega la Vista de Portal. Se debe seleccionar la opción Crear proyecto (ver Fig. 2.6). Se recomienda crear un directorio cerca de la raíz, a fin de evitar rutas de acceso demasiado largas.

2. Escoger la opción *Configurar un dispositivo*, como se muestra en la Fig. 2.7.
3. Seleccionar el controlador. Debe corresponder con el hardware disponible. En este caso, como se muestra en la Fig. 2.8, se escoge la serie SIMATIC 1200 CPU 1215C DC\DC\DC 6ES7 25-1AG40 – 0XB0. Las características técnicas de esta serie en particular se resumen en la Tabla 2.6. Se da clic al botón *Agregar* y se despliega la Vista de proyecto que muestra el controlador creado y el resto de las opciones de configuración y programación del PLC.

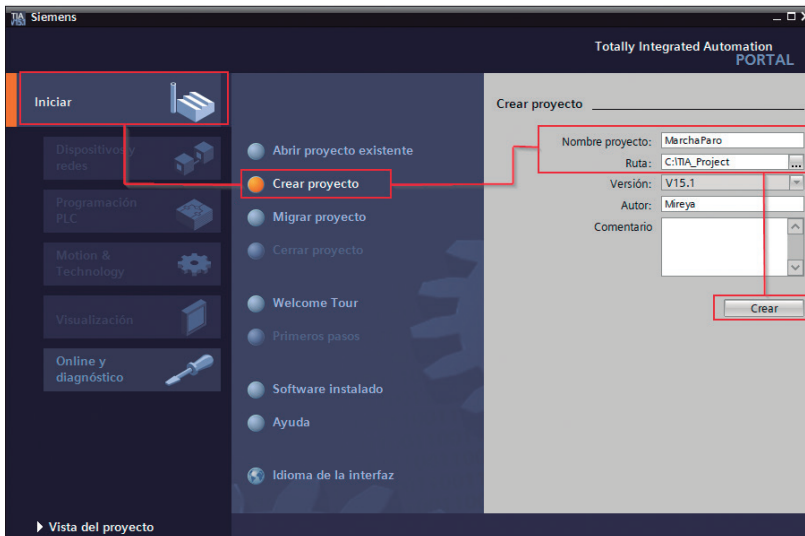


Figura 2.6: Creación de proyectos - Vista de portal

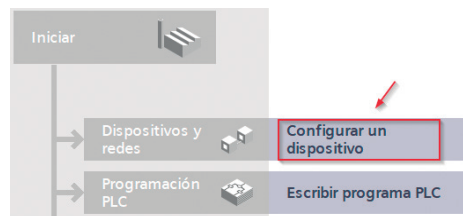


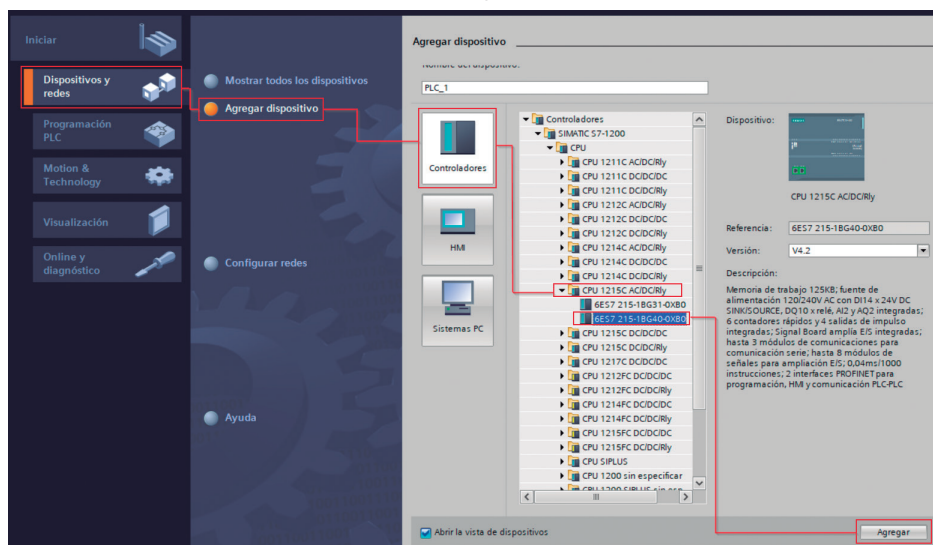
Figura 2.7: Insertar controlador

## 2.3.2. Crear un programa

En TIA Portal, los programas se escriben en bloques. Cuando se crea un proyecto, se genera automáticamente un bloque de organización denominado “Main [OB1]”, donde se aloja el programa de usuario que determina el comportamiento del controlador. Debe existir al menos un bloque de organización. No obstante, se pueden crear varios tipos bloques para conseguir una programación estructurada.

### *Tipos de bloques de programa de usuario*

Existen tres tipos de bloques de programa:



**Figura 2.8:** Selección del controlador

- Bloques de Organización (OBs): el sistema operativo del controlador llama al OB y ejecuta las instrucciones del programa de usuario en un ciclo infinito de forma cíclica. Un programa puede implementarse en un único OB; sin embargo, esto requerirá un mayor esfuerzo de programación y uso de memoria. TIA Portal ofrece la posibilidad de organizar y estructurar tareas en funciones (subrutinas) que se llaman desde el OB.

- Bloques de función (FB): ofrece una parametrización flexible. Cada FB tiene su propio bloque de memoria denominado “Instancia de bloque de datos” (DB). Mantienen sus valores de forma permanente cuando se les asigna un bloque de datos (DB).
- Funciones (FC): son bloques sin almacenamiento de datos. Se usan para programar subfunciones recurrentes. Los datos locales de una función se pierden tras su ejecución.

El uso de bloques brinda mayor facilidad para visualizar y entender un programa que ejecuta tareas complejas de automatización, ya que se implementa una programación modular con lo que, además, se consigue reducir los tiempos de depuración. Es así que los FB y FC son creados una sola vez en la memoria de la CPU, pero pueden ser llamados o invocados cualquier cantidad de veces en el programa.

En el ejemplo presentado, la programación se realiza en el bloque de organización MAIN [OB1]. Al realizar la selección, se abre el entorno de desarrollo para crear el programa.

### Editor de programas

El editor de programas está dividido en segmentos utilizados para estructurar el programa, se pueden crear hasta 999 segmentos.

En el primer segmento, se utilizan los símbolos de instrucción para representar los operandos de contactos y bobinas de la paleta de favoritos; se construye el programa que operará como un circuito de marcha-paro, tal como se muestra en la Fig. 2.9.

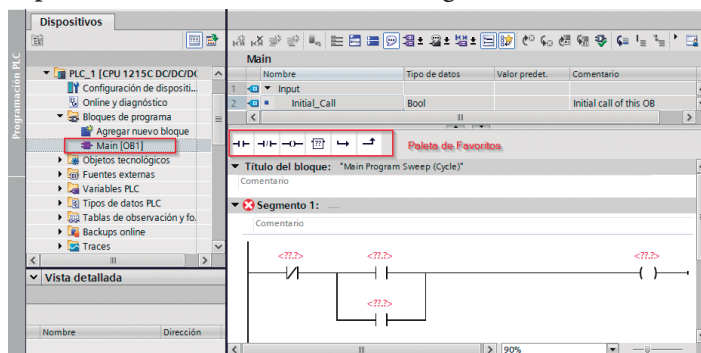


Figura 2.9: Circuito marcha-paro

**Definición de variables:** El siguiente paso para completar el programa de marcha-paro es la asignación de variables a los operandos. Una variable se utiliza para almacenar datos que pueden tomar distintos valores de acuerdo con las condiciones del programa en cada ciclo de ejecución.

En el árbol del proyecto, en la pestaña *Variables PLC*, se encuentra la opción *Tabla de variables estándar*. Aquí se crean las variables del ejemplo de marcha-paro, como se muestra en la Fig. 2.10. Las variables asignadas son de tipo booleano y corresponden a entradas y salidas físicas.

Tabla de variables estándar			
	Nombre	Tipo de datos	Dirección
1	START	Bool	%IO.0
2	STOP	Bool	%IO.1
3	MOTOR	Bool	%Q0.1

Figura 2.10: Tabla de variables marcha-paro

Como siguiente paso, se debe regresar al bloque de organización Main [OB1] y realizar la asignación de variables a cada una de las instrucciones KOP, como se muestra en la Fig. 2.11.

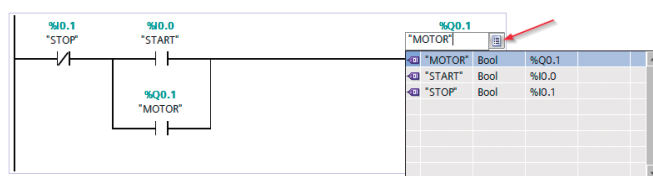


Figura 2.11: Asignación de variables

### 2.3.3. Simulación

La herramienta de simulación utilizada en TIA Portal es S7-PLCSIM v15.1, la cual permite realizar la comprobación y depuración de un programa sin necesidad del hardware. Esta versión simula los PLC de la serie S7-1200. TIA Portal es la interfaz que permite que S7-PLCSIM y STEP 7 trabajen conjuntamente para simular en función de la configuración del hardware y el programa de usuario descargado al PLC simulado en S7-PLCSIM. Los valores de la tabla simulación son equivalentes a las

entradas del PLC físico. Además, se pueden crear secuencias de ingreso de datos que permiten acelerar las tareas de depuración.

Para arrancar una simulación, se deben seguir los siguientes pasos:

1. En la ventana de árbol del proyecto en la pestaña **PLC\_1[CPU 1215CDCDCDC]**, con clic derecho, abrir el menú emergente y seleccionar la opción *Iniciar simulación* (ver Fig. 2.12). También se puede acceder a esta opción a través del menú, con la opción *Online→Simulación*.

Posteriormente, se abre una ventana de notificación para indicar que se desactivarán las interfaces online restantes, a lo cual se debe dar clic en el botón *Aceptar*.

2. En la ventana de *Carga avanzada* se realiza la conexión lógica entre el PC y PLC virtual (ver Fig.2.13). Para la detección y conexión se debe seguir los siguientes pasos:

- ① Seleccionar la conexión entre las interfaces PN/IE y PLCSIM.
- ② Para identificar el dispositivo, se debe dar clic en *Iniciar búsqueda*. Una vez hallado, se muestra la dirección IP asignada y se visualiza la conexión virtual entre PC y PLC.
- ③ Dar clic en botón *Cargar*.

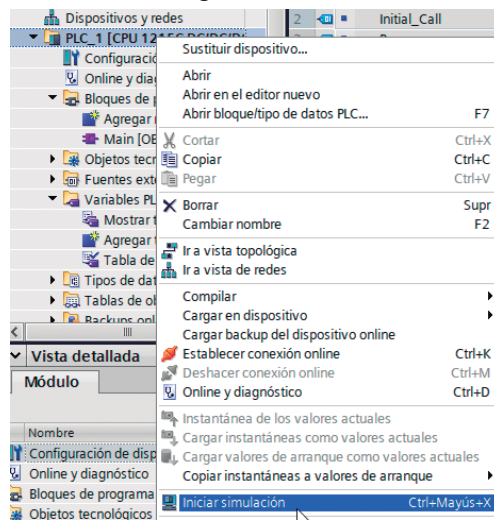



Figura 2.12: Asignación de variables



3. Una vez realizada la conexión, se compila y carga el programa de usuario en el PLC simulado. Si esta está libre de errores, se mostrará la ventana *Vista preliminar*. Esta carga muestra el diagnóstico entre la configuración del proyecto y la del PLC (ver Fig. 2.14). Si el diagnóstico es correcto, dar clic al botón *Cargar*.
4. En la ventana *Resultados de la operación de carga*, escoger la opción *Arrancar módulo* (ver Fig. 2.15) y presionar el botón *Finalizar*.
5. La ventana compacta del simulador en la Fig. 2.16 muestra que el PLC simulado está en modo RUN cuando su led es verde. Para abrir la ventana expandida, se debe presionar el botón de su extremo superior derecho.
6. Una vez abierta la ventana extendida, se debe crear un nuevo proyecto de simulación. Este proceso toma tiempo en completarse. Para proceder, se debe seleccionar la opción *Tablas SIM → TABLA SIM\_1* y exportar las variables a simular del proyecto presionando *Cargar variables del proyecto*,  como se muestra en la Fig. 2.17.
7. Finalmente, el proyecto entra en modo de simulación y es posible activar y desactivar las señales de entrada para depurar y comprobar la lógica del programa. Para observar cómo se comporta el programa KOP en TIA

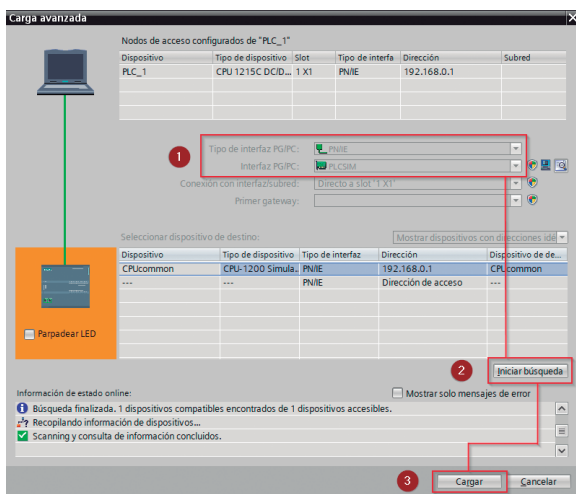


Figura 2.13: Carga avanzada

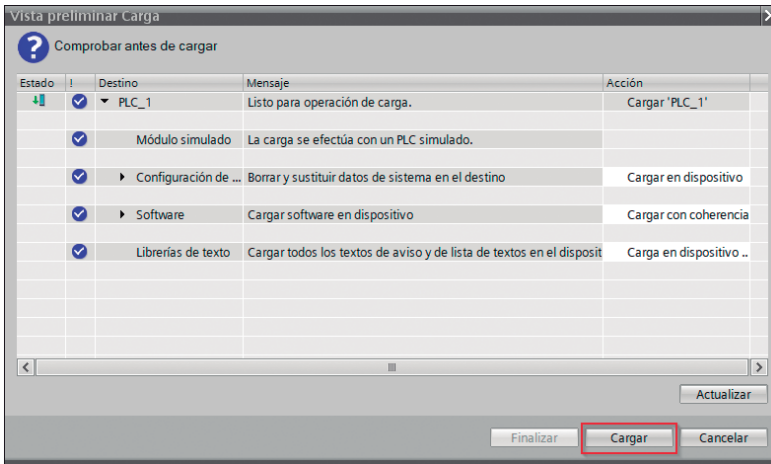


Figura 2.14: Vista preliminar *Carga*

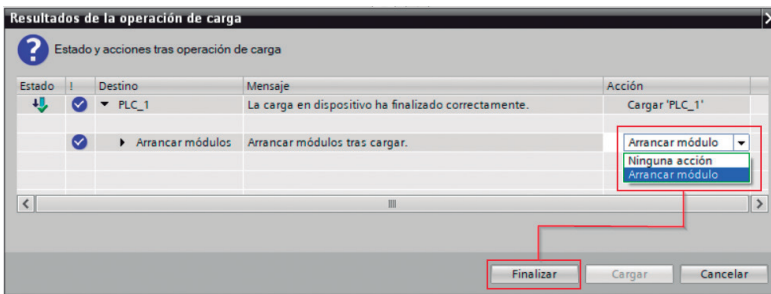


Figura 2.15: Arrancar módulo

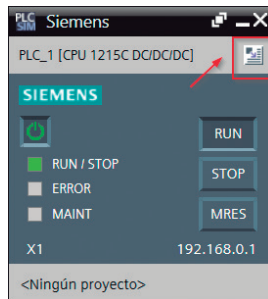


Figura 2.16: Ventana compacta de PLCSIM

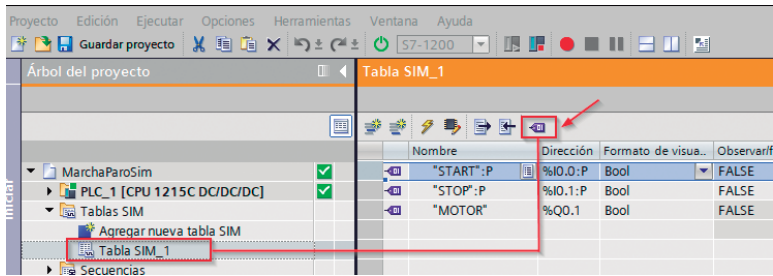


Figura 2.17: Asignación de variables

Portal se debe dar clic en el botón *Activar/Desactivar observación*. Cuando finaliza la acción, los indicadores en verde que aparecen en el árbol del proyecto indican que los elementos del programa online y offline son iguales.

En este estado, se puede observar los cambios de cada variable de forma interactiva de acuerdo con el accionamiento dado en la Tabla SIM\_1 del simulador como se muestra en la Fig. 2.18. PLCSIM también permite simular Secuencias, donde se pueden asignar tiempos de activación y desactivación a cada variable de estado. Esta opción es muy útil cuando se trabaja con programas más complejos.

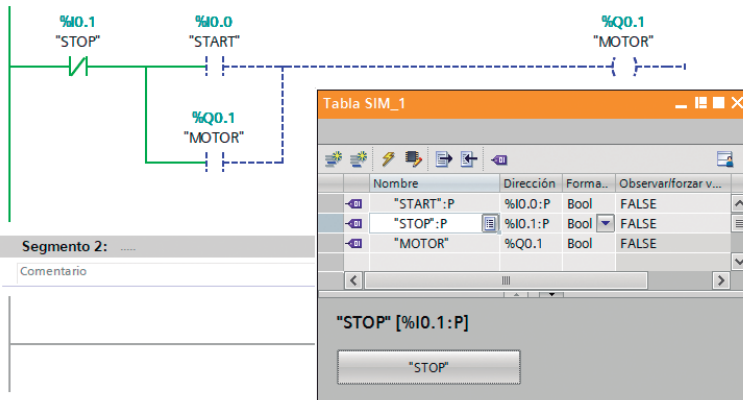


Figura 2.18: Simulación PLCSIM

## 2.4. Interfaz HMI

La visualización de operaciones en tiempo real en un proceso industrial que permite la interacción remota con los equipos para control y supervisión, representa una gran ventaja para los operarios, por lo cual, el uso de interfaces humano-computador es cada vez más requerido en la industria.

Un HMI (por sus siglas en inglés Human Machine Interface) es un dispositivo para desarrollar aplicaciones gráficas que sirven de interfaz entre un sistema de automatización industrial y los operarios. Por medio de una pantalla táctil, se puede observar el estado de los dispositivos industriales de forma gráfica, simulando sus características y comportamiento. Se pueden simular motores, válvulas, pulsadores, pilotos luminosos, actuadores analógicos, entre otros. Además, puede ser configurada para mostrar información de nivel, temperatura, presión, secuencias, contadores y gráficos interactivos, entre otros (ver Fig. 2.19). Esta herramienta permite a los supervisores de las líneas de producción recolectar datos, llevar históricos, realizar diagnósticos y monitorear el estado de ciertas variables, así como controlar su evolución en un ciclo de control [15] [16].

Los HMIs, por lo general, se encuentran montados en el panel de control o en el cuarto de mando. Manejan su propio lenguaje de programación, que cambia dependiendo del fabricante. El software permite al usuario diseñar lo que se visualizará en la pantalla, por ejemplo: los botones que pueden ser pulsados, estados de alerta, cuadros de textos con valores de variables analógicas, gráficos interactivos que cambien en función del estado de una variable, etc. El programador debe relacionar cada botón e indicador con la respectiva dirección de entrada y salida del PLC. Es así que el HMI debe ser compatible con el autómata programable con el cual interactúa para establecer la comunicación y realizar la lectura/escritura de las variables manipuladas. Ambos dispositivos deben manejar el mismo protocolo de comunicación. Entre los más comunes están Modbus, Ethernet IP y Profibus [18].



Figura 2.19: HMI Básicas. Tomado de [17]

### 2.4.1. Ventajas de usar un HMI

Un HMI puede ser utilizado en una planta industrial que requiera operaciones de monitorización y control de forma dinámica como procesos de almacenamiento, montaje, embotellado, pasteurización, dosificación, ensamblaje [19], entre otros. A continuación, se listan las siguientes ventajas:

- Reducción de errores.
- Facilidad de mantenimiento.
- Obtención y actualización de información en tiempo real.
- Facilidad de operaciones de supervisión.
- Incremento de la productividad en aplicaciones industriales, ya que permitirá agilizar determinados procesos. Se producirá más en menor tiempo.
- La conexión con otros dispositivos también es posible, lo que amplía su campo de aplicaciones en el campo de IoT que va desde la monitorización de redes industriales hasta el control acceso remoto.
- El almacenamiento de datos se realiza mediante la programación del HMI para que envíe automáticamente los parámetros

definidos a una unidad de almacenamiento en la red. Estos datos podrán ser usados para tomar decisiones, ajustar configuraciones, etc.

## 2.4.2. Agregar Interfaz HMI

Para simular el HMI del circuito marcha-paro que se implementó en la sección anterior, las variables de START y STOP se cambian por las marcas M0.0 y M0.1 respectivamente en la *Tabla de variables del PLC*.

A continuación, mediante la opción *Agregar dispositivo*, aparecerá la pantalla de la Fig. 2.20 donde se selecciona el modelo del HMI, en este caso, KTP700 Basic 6AV 123-2G03-0AX0 que corresponde a una pantalla táctil de 7 pulgadas con conexión PROFINET y 8 teclas de función.

A partir de esto, se abre el *Asistente de Panel de Operador* en el cual se establece la conexión con el PLC, presionando el botón *Examinar* y seleccionando el PLC al cual el HMI va a estar conectado (ver Fig.2.21). El resto de pantallas permiten configurar opciones de visualización. Para este ejemplo, se ha desactivado el encabezado, avisos y botones.

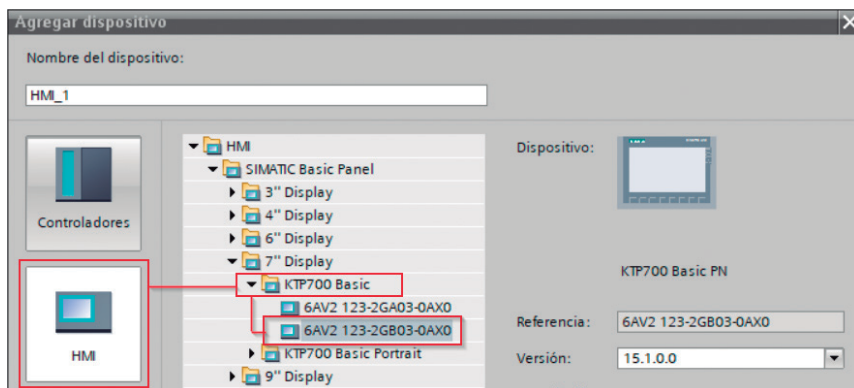


Figura 2.20: Agregar HMI

En este ejemplo de HMI, se insertarán 2 botones START y STOP que se conectan a las variables del PLC. Se insertará un objeto gráfico

de un motor que cambiará su color a verde cuando se presione el botón START, indicando que está encendido. Al presionar el botón STOP, el gráfico cambiará a gris. Además, se incluirá un texto interactivo que indica el estado del motor encendido/apagado. En la pantalla gráfica se arrastran los siguientes objetos de la ventana:

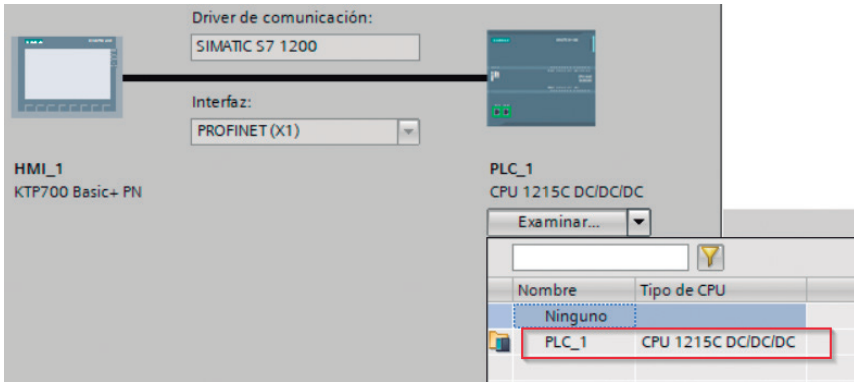


Figura 2.21: Conexión entre PLC y HMI

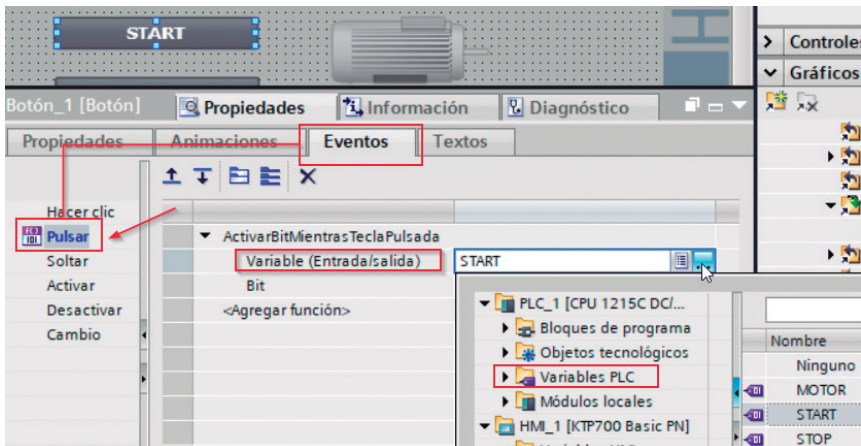


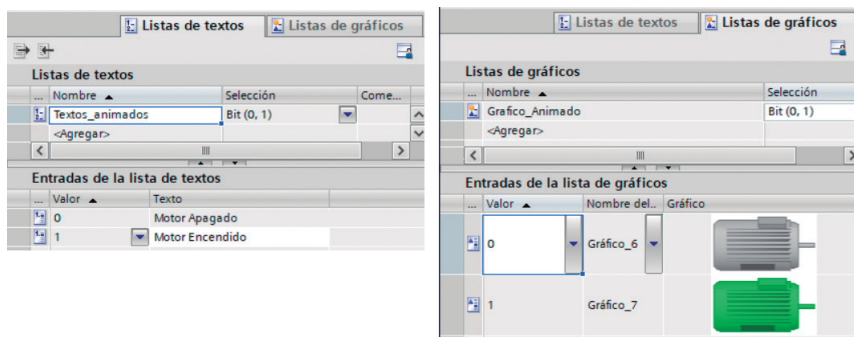
Figura 2.22: Asignación de eventos a botones

*Herramientas→Elementos.*

- 2 botones (para START y STOP)
- Campo de E/S gráfico (para gráfico de motor interactivo)
- Campo de E/S simbólico (para mensaje interactivo)

Para configurar los botones, se debe acceder a *Propiedades* a través de la pestaña ubicada en la parte inferior de TIA Portal o mediante clic derecho → *Propiedades*. En la pestaña *Eventos*, seleccionamos *Pulsar* y en *Agregar la función*, seleccionar *Procesamiento de bits* → *Activar Bit Mientras Tecla Pulsada*.

Para la asignación de variable, se abre la ventana emergente y de la Tabla de variables del PLC, se asigna la que corresponda —como se muestra en la Fig. 2.22—. Cuando se utilizan las entradas físicas, se debe crear una tabla de variables en la HMI y definir las correspondencias con las variables del PLC.



**Figura 2.23:** Asignación de eventos a botones

Para configurar los campos simbólicos y gráficos, se debe seleccionar en el árbol del proyecto, en las opciones del HMI, la *Lista de textos y gráficos*. Para cada pestaña, se debe agregar un objeto. Su configuración se muestra en la Fig. 2.23. Los gráficos se insertan desde la ventana *Herramientas* → *Gráficos* de TIA Portal.




La configuración del Campo E/S simbólico se realiza accediendo a *Propiedades*, en la pestaña *General*, se deben realizar las siguientes asignaciones:

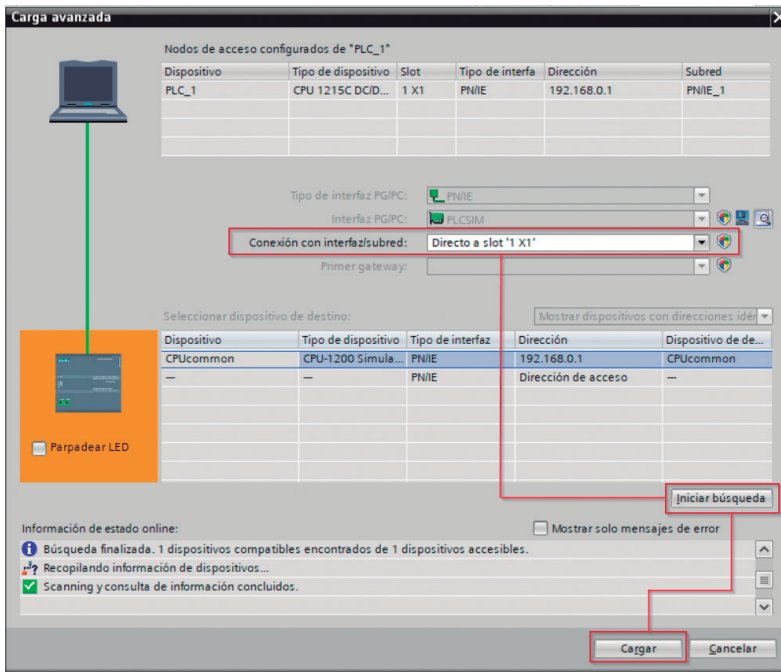
- Proceso: Asignar la variable relacionada de la tabla de variables del PLC, en este caso, MOTOR (% Q0.1).
- Modo: Dado que no se va a ingresar ningún dato, se selecciona la opción *Salida*.
- *Contenido*: Abrir la ventana de selección y marcar la etiqueta *Gráfico\_ Animado* que se creó en el paso anterior. Esta acción permite asociar los estados de la variable MOTOR (%Q0.1) con las figuras declaradas en la lista de gráficos.

Se sigue el mismo procedimiento para la configuración del Campo E/S simbólico; se asigna la misma variable MOTOR, y se selecciona la lista *Textos\_animados*.

### *Simulación*

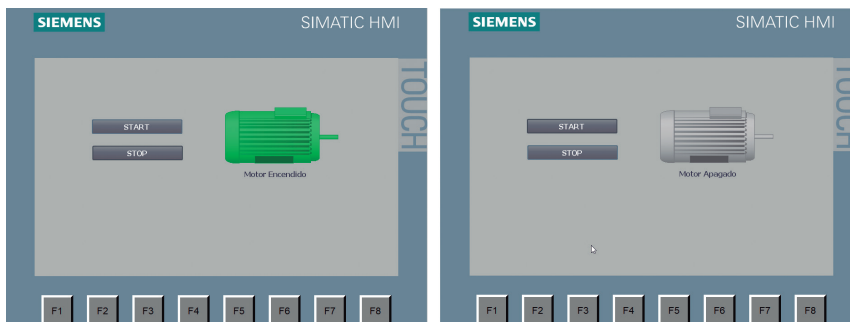
Antes de ejecutar la simulación, es necesario compilar el HMI para detectar errores. Se puede realizar directamente pulsando el botón  en la barra de botones o, en el árbol del proyecto sobre el HMI, se escoge la opción *Compilar* → *Software (compilar todo)*.

Para simular la aplicación HMI, es necesario cargar el programa del PLC al simulador PLC SIM. En la pantalla mostrada en la Fig. 2.24, seleccione *Conexión con interfaz/subred* → *Directo a slot "1 X1"* para simular el HMI.



**Figura 2.24:** Conexión de dispositivos virtuales para simulación

Tras cargar el PLC, sobre el HMI en el árbol del proyecto, dar clic derecho y seleccionar *Iniciar simulación*. En la Fig. 2.25 se muestra la pantalla de visualización del HMI diseñado.



**Figura 2.25:** Visualización del HMI


### 2.4.3. Carga del programa al PLC y HMI

Una vez que se ha comprobado mediante simulación que la lógica del programa y la visualización del HMI funciona de acuerdo a las condiciones establecidas, se procede con la descarga del programa en el hardware (PLC y HMI). Para que pueda existir comunicación entre PC y PLC, se debe realizar la conexión física y la configuración lógica.

La física se logra interconectando ambos equipos mediante cable Ethernet; se puede utilizar un switch para establecer una red de PLC's y HMI. La configuración lógica se emplea para identificar los dispositivos en la red. En los equipos PC, PLC y HMI, es necesario establecer una dirección IP distinta para cada uno pero en la misma subred para que se puedan comunicar. Se deben evitar direcciones de red, broadcast y gateway como IP de dispositivos.

Los pasos para cargar el PLC y el HMI con el programa son:

1. Conectar mediante cable Ethernet el PC, PLC y HMI.
2. Asignar o verificar la dirección IP de la PC para el puerto de la tarjeta Ethernet conectado con el PLC. Si se trabaja con Windows, se puede acceder a la opción *Abrir Configuración de red e internet* → *Ethernet* y asignar una dirección IPv4 manual, por ejemplo:
  - Dirección IP: 192.168.0.10
  - Máscara de subred: 255.255.255.0
  - Gateway: 192.168.0.1
3. Verificar que la dirección IP del PLC se encuentre en la misma subred. Para acceder a dicha configuración en el árbol de proyecto sobre el PLC, dar clic derecho y seleccionar *Propiedades*. En las opciones de la pestaña *General*, acceder a *Interfaz PROFINET [X1]* y desde aquí es posible modificar la configuración del Protocolo IP. Se propone como ejemplo:

- Dirección IP: 192.168.0.15
  - Máscara de subred: 255.255.255.0
  - Gateway: 192.168.0.1
4. Verificar la dirección IP del HMI tal que se encuentre en la misma subred. Para acceder a dicha configuración en el árbol de proyecto sobre el HMI, dar clic derecho y seleccionar *Propiedades*. En las opciones de la pestaña *General*, acceder a *Interfaz PROFINET [X1]*; desde aquí es posible modificar la configuración del Protocolo IP. Se propone como ejemplo:
- Dirección IP: 192.168.0.20
  - Máscara de subred:255.255.255.0
  - Gateway: 192.168.0.1
5. Para verificar que todos los dispositivos se pueden comunicar, seleccionar *Accesos online* en el árbol de proyecto para desplegar la lista de dispositivos de conexión. Identificar la tarjeta de red del PC a la cual se conecta el PLC y seleccionar *Actualizar dispositivos accesibles*. Si todos los dispositivos conectados son accesibles, aparecerán en una lista con su respectiva IP.
6. Descargar el programa pulsando el botón  .

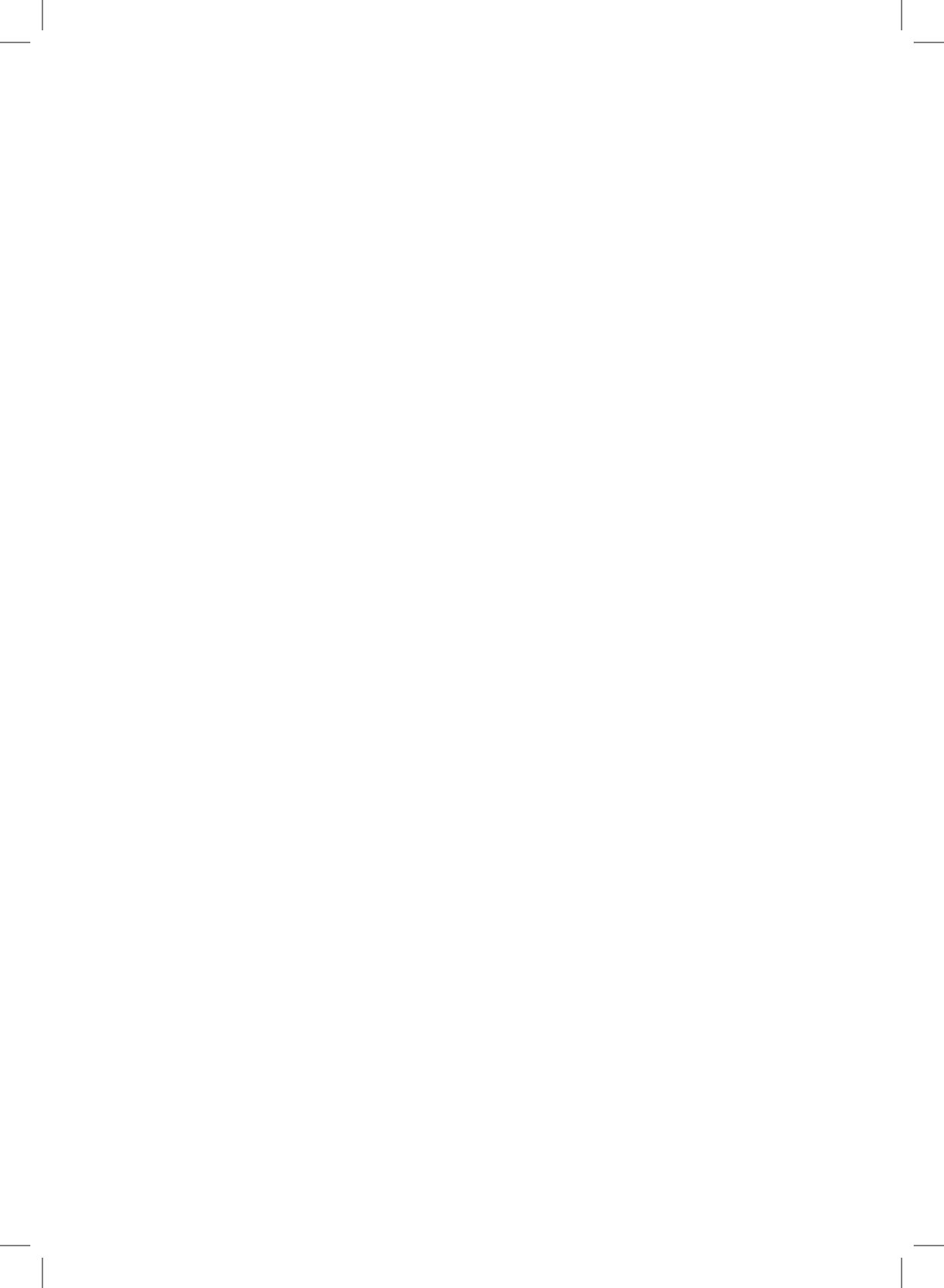
## 2.5. Preguntas de repaso

1. ¿Se pueden programar todos los PLC's de cualquier serie de Siemens en TIA Portal?
2. ¿Cuál es la función de STEP 7 y WinCC en TIA Portal?
3. ¿Qué campos se deben definir para crear una variable en TIA Portal?

4. ¿Cuáles son los tipos de variables que se pueden definir en TIA Portal?
5. ¿Qué tipo de funciones de bloque están disponibles para estructurar un programa en TIA Portal?
6. ¿Es necesario volver a cargar el programa al PLC cuando se apaga el PLC y se vuelve a encender?
7. ¿Qué ventajas tiene la vista de portal en TIA Portal?
8. ¿Bajo qué condiciones se considera recomendable incorporar un HMI a un proceso industrial?
9. ¿Puede un HMI realizar operaciones de lectura y escritura con un PLC? Explique con un ejemplo.
10. Un PC tiene la siguiente configuración de red:
  - Dirección IP: 172.16.0.5
  - Máscara de subred: 255.255.0.0
  - ¿Qué dirección IP y máscara de red asignaría a un PLC y un HMI para que puedan comunicarse todos los elementos en una misma red? Escriba el ítem.

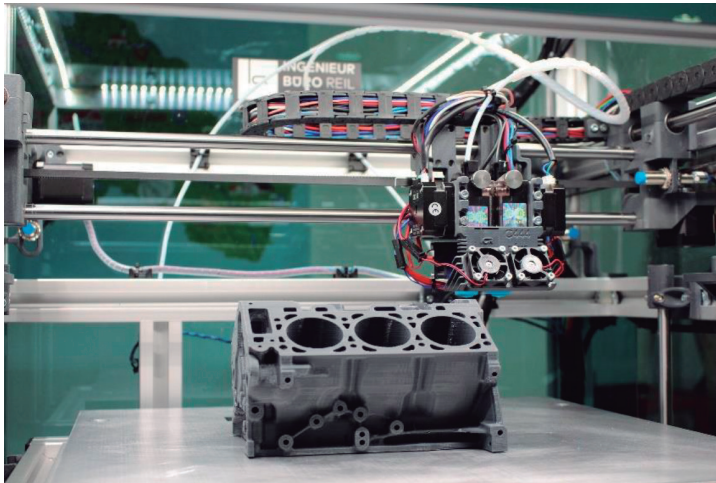


**Capítulo 3**  
**Lenguaje de programación**  
**Ladder**





Para comunicarse entre máquinas y usuarios, se han diseñado lenguajes de programación utilizando símbolos, caracteres, secuencias, temporizadores e instrucciones. Esta forma de comunicación ha permitido crear sistemas automáticos para controlar la operación de máquinas y procesos. Este lenguaje de programación ha sido estandarizado bajo la norma IEC 61131, que trata de unificar el sistema de programación de todos los autómatas en un único estándar internacional. En este capítulo se explicará la programación en lenguaje de escalera, uno de los más utilizados a nivel industrial.



**Figura 3.1:** Industria-Tecnología

### 3.1. Introducción

Dentro de los lenguajes de programación se encuentra el KOP (Kontaktplan), también conocido como diagrama de contactos, de escalera o Ladder. Su funcionamiento es por medio de la conexión en serie o en paralelo de contactos eléctricos. El lenguaje de escalera, es el lenguaje

gráfico más popular en la programación de PLC's, ya que es símil a un circuito de control eléctrico [20].

Para programar a un PLC se tienen dos caminos:

- Hacerlo directamente sobre el PLC, siempre y cuando este disponga de botones y una pantalla; tal es el caso del micro PLC Logo. En la pantalla, se pueden visualizar números, letras y líneas entre, otros. Utilizando estos botones se accede a menús, comandos y demás elementos. Programar de esta forma es incómodo, pero no imposible.
- Programar mediante una computadora a través de los drivers del PLC y luego conectarlo y cargar el programa al controlador. La mayoría de las industrias utilizan esta tecnología, basada en los controladores lógicos programables, ya que tiene algunas ventajas, tales como los bajos costos de implementación a diferencia de otras tecnologías.

## 3.2. Estructura del diagrama Ladder

La estructura del diagrama Ladder que se muestra en la Fig. 3.2 es bastante simple. Se pueden apreciar dos líneas verticales que representan las líneas de alimentación. La vertical izquierda es la fase y la línea vertical derecha es el neutro; las horizontales son consideradas escalones o, en inglés, row.

El diagrama Ladder que se muestra en la figura tiene tres escalones; los símbolos que suelen estar al lado izquierdo del diagrama Ladder son las entradas —las cuales pueden ser interruptores, sensores, pulsadores o memoria—. Es decir, cada símbolo puede representar a estos tipos de elementos de control dentro de una instalación eléctrica. Los símbolos que están al lado derecho representan las salidas que van conectadas a los actuadores. En la Fig. 3.2, se muestran tres salidas que van conectadas a elementos tales como motores, cilindros, válvulas o memorias.

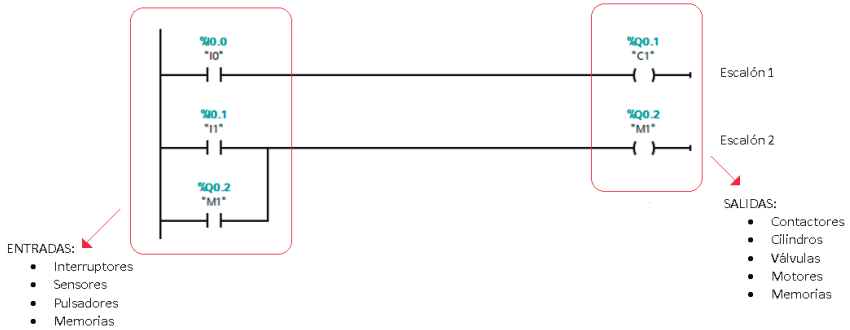


Figura 3.2: Estructura diagrama Ladder

### 3.3. Simbología

De acuerdo con la normativa IEC61131, se ha normalizado la simbología utilizada en el lenguaje de escalera. En la Fig. 3.3 se presenta un resumen.

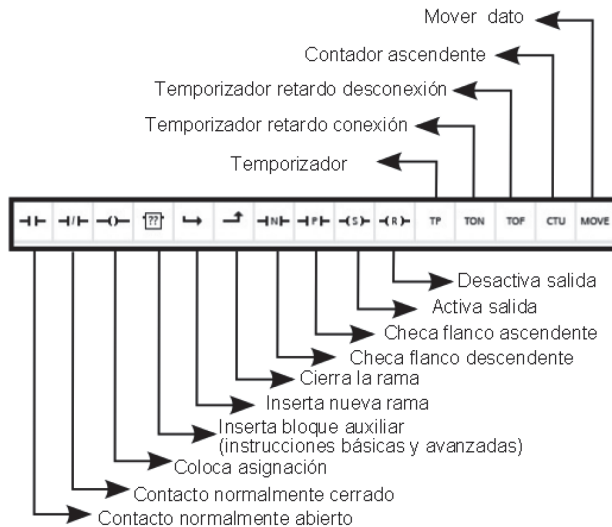


Figura 3.3: Simbología Ladder

### 3.4. Asignación de etiquetas

En el lenguaje de programación Ladder se utilizan una serie de etiquetas que permite diferenciar los elementos en la programación. En la Tabla 3.1 se presentan los tipos de etiquetas que maneja TIA Portal.

**Tabla 3.1:** Tipo de etiquetas

Tipo de elemento	Tipo de etiqueta
Entrada	I
Salida	Q
Marca auxiliar	M
Comando SET	S
Comando RESET	R

Las etiquetas se manejan en inglés, la I de *input* como entrada, la Q como *salida*, las *memorias* etiquetadas con una M, los comandos *set* con una S y *reset* con una R. Se debe tomar en cuenta que las etiquetas son variables; se pueden colocar las etiquetas que deseemos, siempre y cuando no esté estandarizado en los drivers del PLC o del simulador que se va a utilizar.

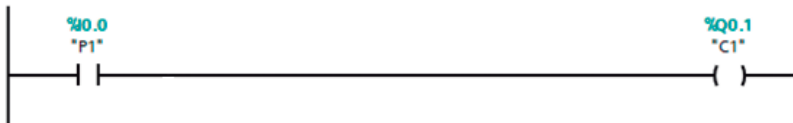
### 3.5. Proposiciones lógicas

Las proposiciones lógicas son funciones lógicas que se aplican también en la programación dentro de lo que es el lenguaje Ladder para PLC. Existen siete básicas para empezar; se pueden hacer combinaciones para el diseño de sistemas más complejos [21].

### 3.5.1. Proposición lógica piloto

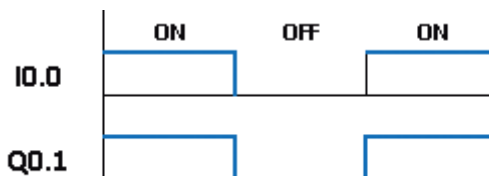
Para probar el controlador lógico programable, se utiliza un elemento de entrada que permita controlar la operación de una salida del PLC.

En este programa, el elemento de entrada conectado en I0.0 puede ser un interruptor, pulsador, sensor de nivel o sensor de presión, etc. Este elemento permitirá activar o desactivar la salida Q0.1. El estado de operación de la salida Q0.1 dependerá únicamente del estado en el que se encuentre I0.0. (Ver Fig. 3.4).



**Figura 3.4:** Proposición lógica piloto en Ladder

Para programar este sistema, se debe configurar la entrada I0.0 y la bobina Q0.1 del PLC y conectarlos en serie. Se selecciona el modo simulación para verificar el funcionamiento. Se realiza de la siguiente forma: mientras se tenga accionada la entrada I0.0, la salida Q0.1 estará activada; cuando la entrada I0.0 sea desactivada, la salida también se desactivará. Se puede demostrar el funcionamiento por medio del diagrama de tiempos en la Fig. 3.5



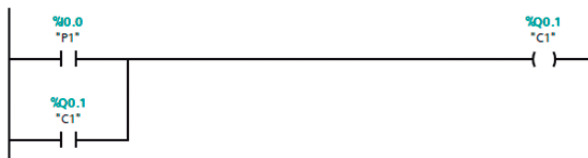
**Figura 3.5:** Diagrama de tiempos de la proposición lógica piloto

### 3.5.2. Proposición lógica memorizada

Otra forma de proposición lógica es utilizar dos entradas conectadas en paralelo. En esta configuración, la entrada I0.0 representa un elemento piloto que puede ser sensor, pulsador, interruptor, etc.; y la entrada Q0.1 realiza una acción de memoria del elemento de salida.

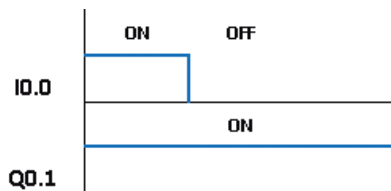
El elemento de entrada conectado en I0.0 permitirá activar la salida Q0.1, quedando activado, a pesar de desactivar la entrada. Para que esto suceda, se necesita conectar en paralelo un contacto normalmente abierto de la salida activada (Q0.1) al elemento de entrada (I0.0) que provocó la energización.

Para programar este sistema, se necesita un contacto de entrada I0.0, una bobina de salida Q0.1 y un contacto normalmente abierto de la salida Q0.1. La entrada I0.0 se conecta en serie a la bobina Q0.1 y el contacto normalmente abierto de Q0.1 se conecta en paralelo a la entrada I0.0, como se ve en la Fig. 3.6.



**Figura 3.6:** Proposición lógica memorizada en Ladder

Cuando se acciona la entrada I0.0, inmediatamente, se acciona la salida Q0.0 y, al mismo tiempo, el contacto abierto pasa a cerrado, el cual genera una autoalimentación indefinida. Se puede visualizar este comportamiento en el diagrama de tiempos de la Fig. 3.7.



**Figura 3.7:** Diagrama de tiempos de un arranque memorizado

### 3.5.3. Proposición lógica con elemento de arranque y paro

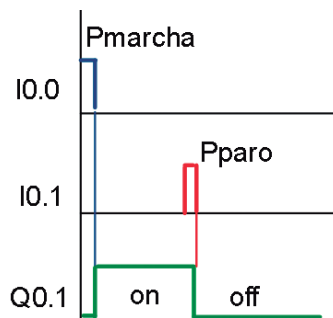
La proposición lógica con elementos de arranque y paro tiene dos entradas conectadas en paralelo y una en serie. La entrada I0.0 representa un elemento piloto, la segunda Q0.1 corresponde a la memoria y la tercera entrada I0.1 conectada en serie corresponde a un elemento que desactivará el circuito indicado. Para ello, se utiliza un pulsante normalmente cerrado, como se muestra en la Fig. 3.8.



**Figura 3.8:** Proposición lógica arranque y paro

Este sistema funciona así: cuando el operador pulsa la entrada I0.0, la salida Q0.1 queda energizada debido a la memoria conectada en paralelo a la entrada. El elemento I0.1 conectado en serie permite desenergizar el circuito; es por ello que esta proposición se denomina también marcha-paro.

Se puede visualizar el comportamiento de la proposición marcha-paro en el diagrama de tiempos de la Fig. 3.9.



**Figura 3.9:** Diagrama de tiempos proposición arranque y paro

### 3.5.4. Proposición lógica con sistema de bloqueo

Existen aplicaciones a nivel industrial que requieren que uno o varios aparatos no funcionen si otros lo están haciendo. Ese es el objetivo general del sistema bloqueo. Para que las dos salidas Q0.3 y Q0.4 nunca funcionen al mismo tiempo, se debe conectar un contacto normalmente cerrado de la salida Q0.3 en serie a la bobina de la salida Q0.4; y un contacto normalmente cerrado de la salida Q0.4 en serie a la bobina de la salida Q0.3. La programación en lenguaje Ladder se puede ver en la Fig. 3.10.

El sistema mostrado en la Fig. 3.10 tiene un conjunto de proposiciones para su funcionamiento, tales como piloto, memorizado y marcha-paro. Para la proposición marcha-paro de la salida Q0.3 se tiene a la entrada I0.0 para el arranque y la entrada I0.1 para el paro. En el caso de la salida Q0.4, se tiene a la entrada I0.2 para el arranque y la entrada I0.3 para el paro. Para el sistema bloqueo, se puede ver un contacto normalmente cerrado de Q0.4 en serie a la salida Q0.3 y, en el otro escalón, se tiene un contacto normalmente cerrado de Q0.3 en serie a la salida Q0.4. Estos contactos impedirán que las dos salidas se enciendan al mismo tiempo, generando el sistema de bloqueo. (Ver Fig. 3.10).

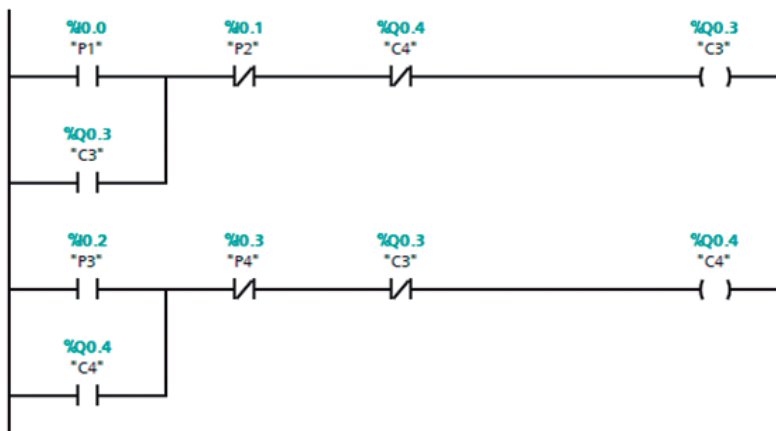


Figura 3.10: Proposición lógica con sistema bloqueo

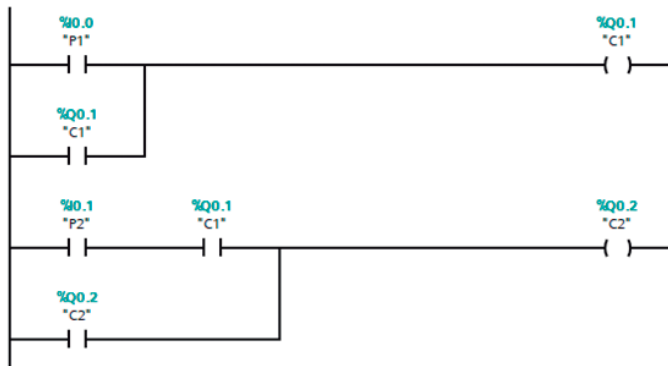


*Observación: Las aplicaciones de este sistema son diversas, una de las más populares es la inversión de giro de motores, donde se desea que los contactos que activan los sentidos de giro del motor no funcionen al mismo tiempo.*

### 3.5.5. Proposición lógica con condicionamiento al encendido

Es muy común encontrar empresas en las que se requiere que una máquina esté encendida para dar paso a que otra se active. Es decir, que se enciendan en secuencia: primero la una y después la otra. En el caso de la programación en Ladder, será necesario accionar primero una salida Q0.1 y solo si esta ya se encuentra prendida, se pueda accionar otra Q0.2. Se puede tener dos condicionantes para esta proposición:

- Verificar que Q0.1 este encendido, antes de que se encienda Q0.2  
Para conseguir esta condición, se debe conectar en serie al elemento que enciende Q0.2 un contacto normalmente abierto de Q0.1 (ver Fig. 3.11).
- Verificar que Q0.1 esté apagado, antes de que se encienda Q0.2.  
Para conseguir esta condición, se debe conectar en serie al elemento que enciende Q0.2 un contacto normalmente cerrado de Q0.1 (ver Fig. 3.12).



**Figura 3.11:** Proposición lógica condicionamiento al encendido 1

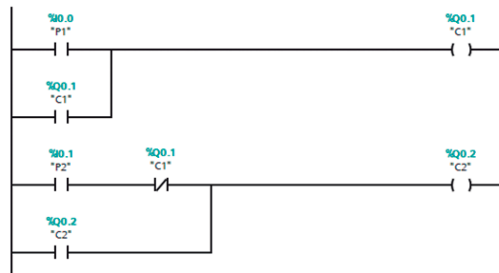


Figura 3.12: Proposición lógica condicionamiento al encendido 2

### 3.5.6. Proposición lógica con condicionamiento al apagado

Muy similar a la proposición anterior, se pueden encontrar ciertos procesos en los que, para desconectar un equipo, previamente se debe desactivar otro.

En el caso de la programación en Ladder, será necesario desactivar primero una salida Q0.2 y solo si esta ya se encuentra desactivada, se puede desactivar Q0.1 Se pueden tener dos condicionantes:

- Verificar que Q0.2 esté apagado antes de que se apague Q0.1.

Para conseguir esta condición, se debe conectar en paralelo al elemento que apaga la salida Q0.1, un contacto normalmente abierto de Q0.2 (ver Fig. 3.13).

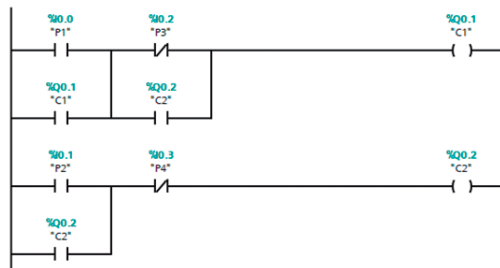


Figura 3.13: Proposición lógica condicionamiento al apagado 1

- Verificar que Q0.2 esté encendido antes de que se apague Q0.1. Para conseguir esta condición, se debe conectar en paralelo el elemento que apaga la salida Q0.1 un contacto normalmente cerrado de Q0.2 (ver Fig. 3.14).

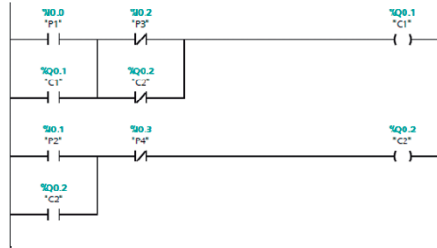


Figura 3.14: Proposición lógica condicionamiento al apagado 2

### 3.5.7. Proposición lógica de dos contactores con alternancia

Para activar de forma alternada dos salidas Q0.1 y Q0.2 a través de un solo elemento de entrada I0.0, se necesitará un elemento auxiliar o una marca M0.0.

En la Fig. 3.15 se presenta la programación en Ladder del mando alternado de dos salidas.

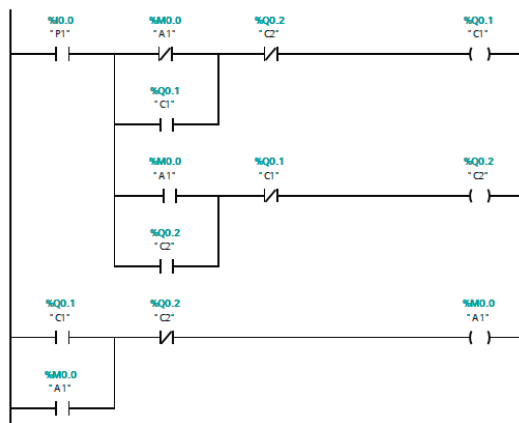


Figura 3.15: Proposición lógica alternada

Para conseguir la alternancia entre dos salidas, se han utilizado varias proposiciones: memoria, bloqueo, condicionamiento al encendido (Fig. 3.15). Para activar la salida Q0.1, se debe tener presionada la entrada I0.0. Así, la activación de la salida Q0.1 dará paso a que se quede activada la salida M0.0. Cuando el operador deje de activar la entrada I0.0, se desactivará Q0.0 y permanecerá encendida la salida auxiliar M0.0. La siguiente vez que presione la entrada I0.0, se activará la salida Q0.2 e, inmediatamente, se desactivará la salida auxiliar M0.0. Los contactos normalmente cerrados Q0.1 y Q0.2 evitan que se activen las dos salidas al mismo tiempo.

### 3.5.8. Ejemplos utilizando proposiciones lógicas

#### *Puerta control manual*

Se requiere controlar de forma manual la apertura y el cierre de una puerta. El sistema consta de los siguientes elementos: un motor de corriente directa, tres pulsadores —dos normalmente abiertos y uno normalmente cerrado—, dos finales de carrera, dos lámparas —una roja y una verde—. Funcionará de la siguiente forma: la puerta empezará cerrada y será detectada por el final de carrera FCcerrado; si se desea abrirla, se utilizará el pulsador normalmente abierto P1; la puerta se abrirá completamente y será detectada por el final de carrera FCabierto. Para cerrarla, se debe presionar el pulsador P2, el cual activará el motor hasta que la puerta se cierre por completo. El pulsador normalmente cerrado P0 se empleará como paro de emergencia. Las lámparas funcionarán a modo de semáforo: la luz verde se encenderá solo cuando la puerta esté completamente abierta; caso contrario, estará encendida la roja.

*Observación: Cuando la puerta se esté abriendo o cerrando no podrá parar hasta que haya terminado por completo.*

En la Fig. 3.16, se presenta la estructura del sistema de la puerta con control manual del problema.

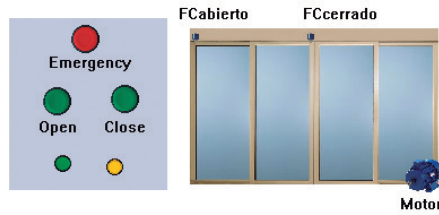


Figura 3.16: Puerta con control manual

### Solución ejemplo 1

En la Fig. 3.17, se presenta la programación en Ladder que soluciona el problema.

Se han utilizado algunas preposiciones básicas, tales como bloqueo, memorizado y piloto.

La salida Q0.1 representa la apertura de la puerta, la salida Q0.2 representa el cierre. La salida Q0.3 es la lámpara verde —que se activa solo cuando el final de carrera Fcerrado está activado—. La salida Q0.4 representa la lámpara roja —la cual estará encendida cuando la puerta esté cerrada o cuando esté en proceso de apertura o cierre.

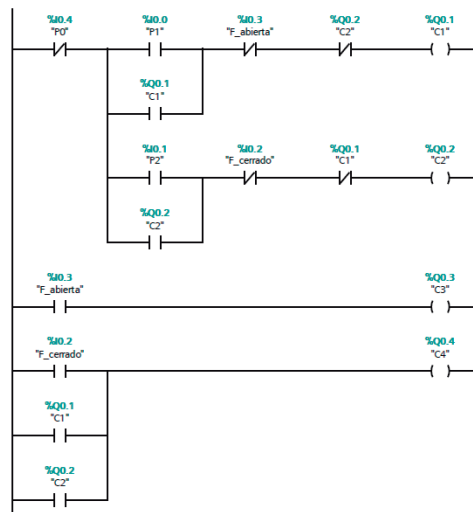
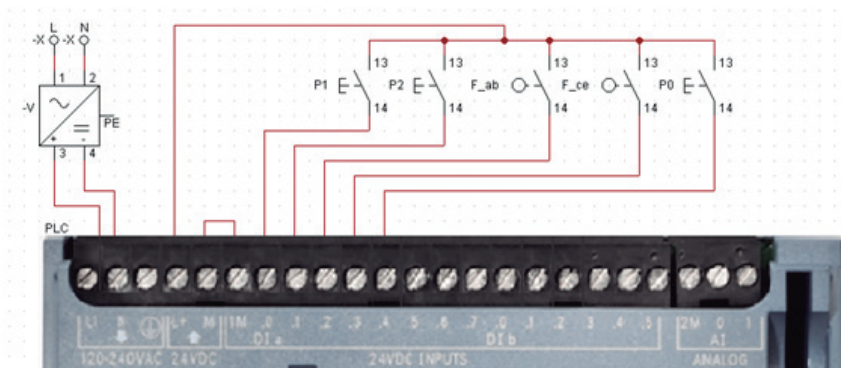


Figura 3.17: Control manual de puerta

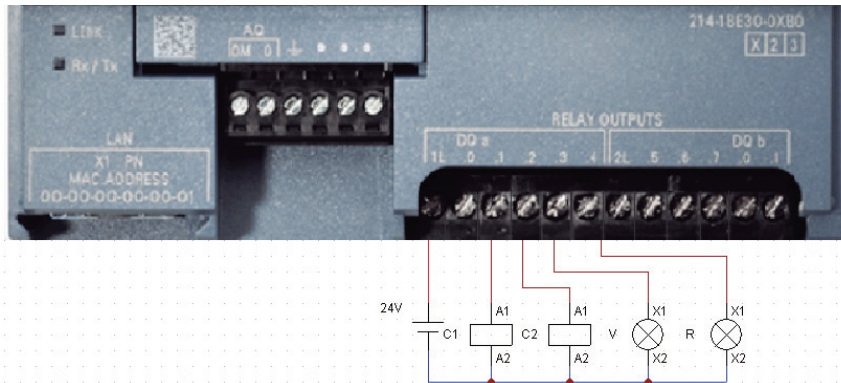
En la Fig. 3.18 se puede apreciar la forma de conexión de las entradas al PLC S7-1200 de Siemens.

Inicialmente, el final de carrera conectado en la entrada I0.3 estará activado, ya que la puerta debe iniciar cerrada. Para comenzar el ciclo de operación, se debe presionar el pulsador P1 conectado a la entrada I0.0, que permitirá abrir la puerta. El contactor conectado a la salida Q0.1 se activará y el motor iniciará el giro para abrir la puerta. El motor se apagará cuando el final de carrera conectado a la entrada I0.2 se active, indicando que la puerta está completamente abierta. Para cerrarla, se presionará el pulsador P2 conectado a la entrada I0.1, la cual activará el segundo contactor, conectado a la salida Q0.2. El motor invertirá su giro y permitirá cerrar la puerta; y se apagará una vez que el final de carrera Fcerrado se haya activado indicando que la puerta está completamente cerrada.



**Figura 3.18:** Diagrama de conexión de entradas del ejemplo 1

En la Fig. 3.19 se puede apreciar la forma de conexión de las salidas al PLC S7-1200 de Siemens.



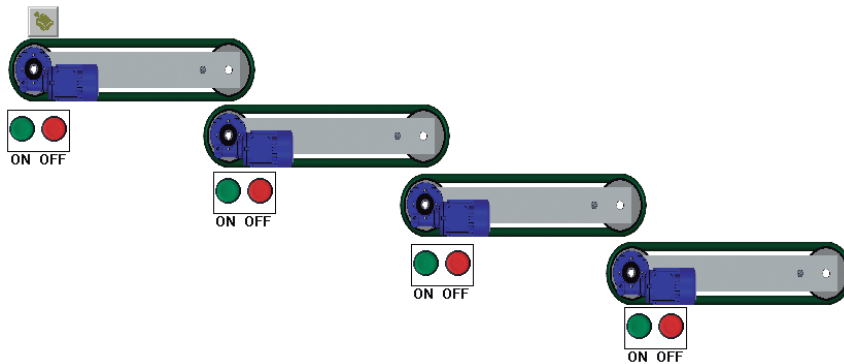
**Figura 3.19:** Diagrama de conexión de salidas del ejemplo 1

*Observación:* Una de las aplicaciones más conocidas para la proposición lógica bloqueo es la inversión de giro de motores. Aquí se presenta un ejemplo para abrir y cerrar una puerta. Más aplicaciones pueden ser el subir y bajar la canasta de un ascensor, que una cinta transportadora gire de izquierda a derecha y subir y bajar una barra de paso peatonal, entre otras.

### *Sistema de cintas transportadoras*

Se requiere controlar un sistema de cintas transportadoras con cuatro motores que operan con las siguientes condiciones: cada motor tendrá su propio accionamiento para el encendido y apagado. Los motores de 1 a 4 deben arrancar exclusivamente en orden consecutivo. Esto se efectuará apretando, en la misma sucesión, los pulsadores de arranque y, cada vez que se pulse uno de ellos, arrancará el motor correspondiente. Al parar el motor 1, se detendrán todos los motores. Al parar cualquier motor pulsando su respectivo botón de paro o cuando se dispare su respectivo relé térmico, se detendrá tanto este motor como los que tengan el número de orden más alto. Por ejemplo, si se detiene el motor 2, se deberán detener también, el motor 3 y 4.

En la Fig. 3.20 se presenta la estructura del sistema de bandas transportadoras del problema.



**Figura 3.20:** Bandas transportadoras en cascada

### *Solución ejemplo 2*

En la Fig. 3.21, se presenta la programación en Ladder que soluciona el problema. Para la solución del ejercicio de bandas transportadoras, se han utilizado algunas preposiciones básicas, tales como condicionamiento al encendido, memorizado y piloto.

Las salidas Q0.1, Q0.2, Q0.3 y Q0.4 representan las salidas que gobernarán a los contactores, que posteriormente activarán los motores de las cintas transportadoras. En serie a las salidas se han conectado los pulsadores normalmente cerrados que desactivarán las salidas; también están dispuestos en serie los contactos auxiliares de los relés térmicos conectados a cada motor, los cuales están provistos para su protección ante sobrecargas.

En la Fig. 3.22 se puede apreciar la forma de conexión de las entradas al PLC S7-1200 de Siemens.

De forma inicial, el sistema está apagado. El problema indica que el encendido es secuencial, es decir C1-C2-C3-C4; entonces, si alguien presiona el pulsador P2, P3 o P4 no debe activarse ninguna salida, ya



que el ciclo inicia con P1 y C1. Para encender todas las bandas, deben presionarse los pulsadores P1-P2-P3-P4 para conseguir la secuencia de encendido C1-C2-C3-C4. El condicionamiento al encendido está ligado, en esta ocasión, al apagado en cascada. Así, el elemento que provoca el condicionamiento al encendido se conecta fuera de la memoria; con esta conexión, se consigue el apagado en cascada.

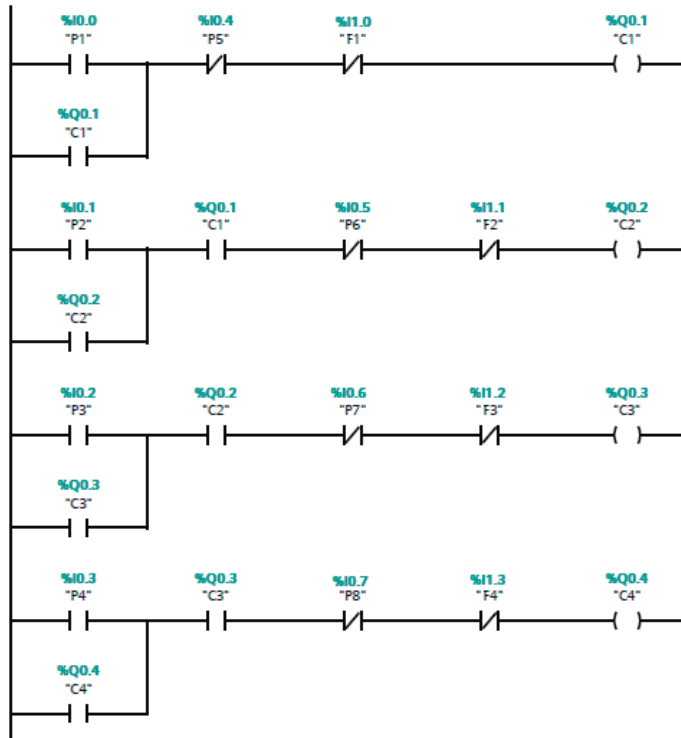


Figura 3.21: Sistema de cintas transportadoras

En la Fig. 3.23 se puede apreciar la forma de conexión de las salidas al PLC S7-1200 de Siemens.

*Observación: Este sistema de regulación se emplea con frecuencia en los equipos de cinta transportadora para evitar una posible acumulación de material, si se detiene una de las secciones intermedias.*

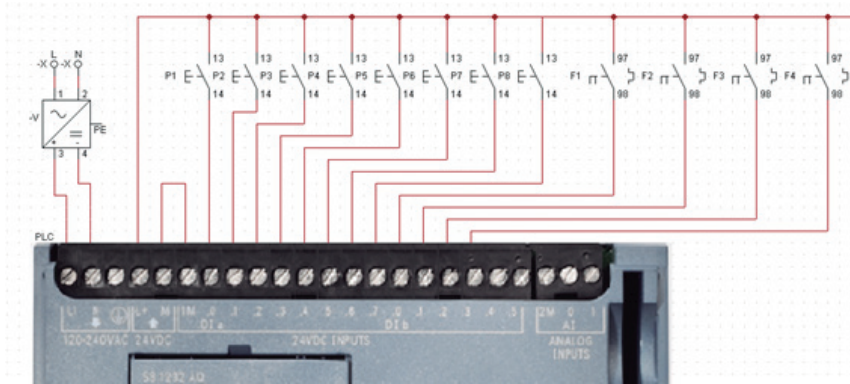


Figura 3.22: Diagrama de conexión de entradas del ejemplo 2

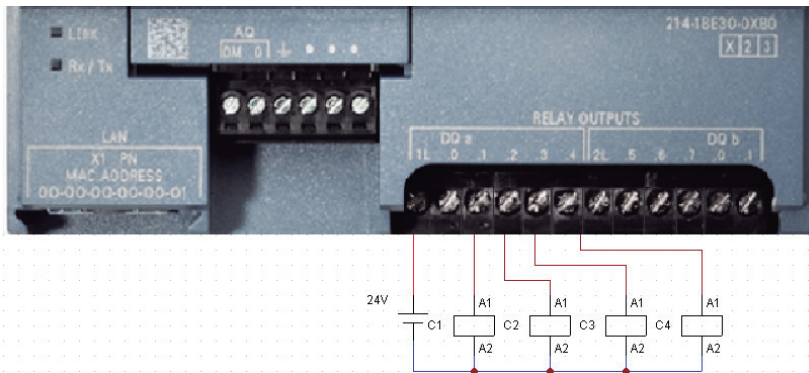


Figura 3.23: Diagrama de conexión de salidas del ejemplo 2

### Molino de granos

Se dispone de un tanque con molino —como el mostrado en la Fig. 3.24—, el cual tiene dos motores: el primero, M1, activa una banda transportadora para llevar granos de café hacia el molino. El segundo, M2, activa un tornillo sin fin que los molerá. El tanque dispone de tres sensores del tipo capacitivo que permitirán conocer si el tanque está con nivel bajo H1, medio H2 o alto H3. La activación de los dos motores será de la siguiente forma.

- El sistema funcionará de modo automático; el motor M1 se activará cuando el tanque esté vacío y se apagará cuando el nivel del tanque sobrepase la altura a H3.
- El motor M2 se activará cuando el nivel esté entre H1 y H3 y se apagará cuando el tanque esté vacío.

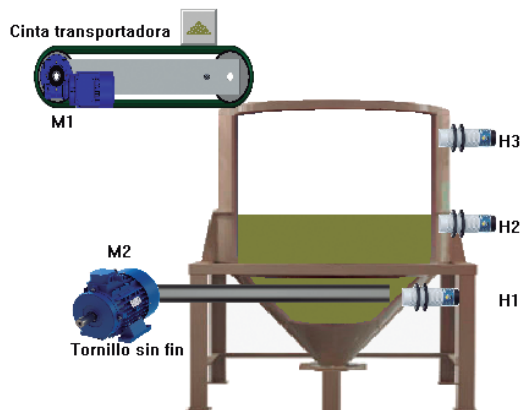
**Tabla 3.2:** Condiciones de encendido del Motor 1

H1	H2	H3	M1
0	0	0	<b>1</b>
1	0	0	<b>1</b>
1	1	0	<b>1</b>
1	1	1	<b>0</b>

Si existe sobrecarga en cualquiera de los dos motores, el sistema deberá detenerse. Una lámpara roja se encenderá para indicar que existe un problema.

### *Solución ejemplo 3*

En la Fig. 3.24, se presenta la estructura del sistema de molino de granos del problema.



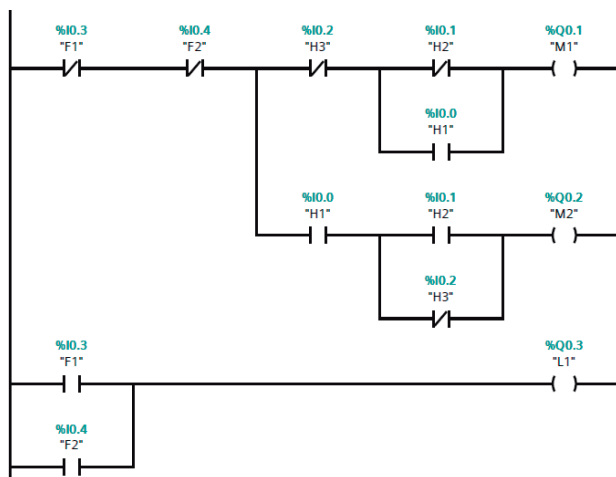
**Figura 3.24:** Molino de granos

Para la solución del caso del molino se ha utilizado lógica binaria. Para resolver este ejercicio, se deben identificar las condiciones del problema: verificar que los sensores activan y desactivan cada motor M1 y M2. La bobina Q0.1 representa el motor M1 —el cual se activará en las condiciones mostradas en la Tabla 3.2—, la bobina Q0.2 representa al motor M2, que se activará en las condiciones mostradas en la Tabla 3.3.

El estado 1 lógico en los sensores de las Tablas 3.2 y 3.3 representa el momento en que los sensores se activan, debido a que han detectado que el tanque ha alcanzado determinado nivel y, en consecuencia, su combinación genera un estado alto en el motor correspondiente. En la Fig. 3.25 se observa la programación en Ladder que satisface las condiciones del problema.

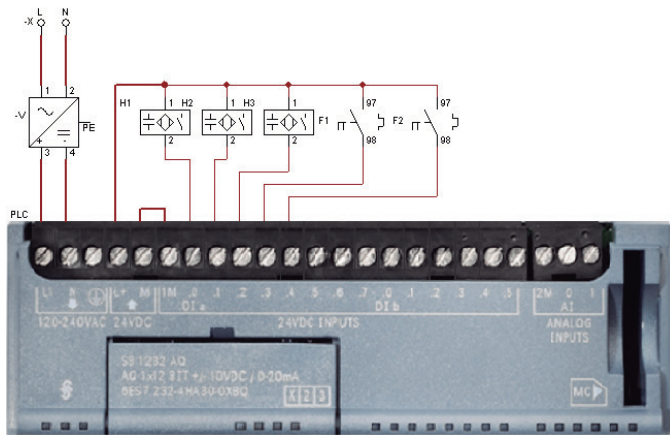
**Tabla 3.3:** Condiciones de encendido del Motor 2

H1	H2	H3	M1
0	0	0	<b>0</b>
1	0	0	<b>1</b>
1	1	0	<b>1</b>
1	1	1	<b>1</b>



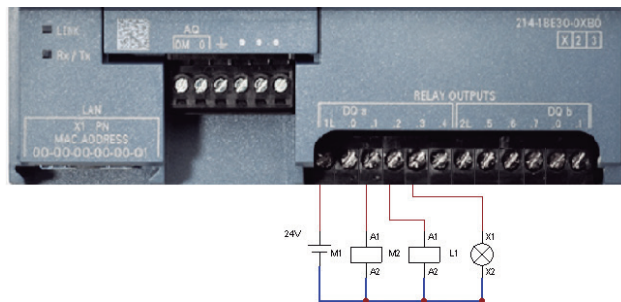
**Figura 3.25:** Solución en Ladder del ejemplo 3

En la Fig. 3.26 se puede observar el diagrama de conexiones de las entradas al PLC S7-1200 de Siemens. El sistema mostrado, además, consta de elementos de protección que corresponden a relés térmicos que se conectarán a cada motor, en caso de sobrecarga los contactos F1 y F2 conectados a I0.3 e I0.4, respectivamente; se activarán desconectando el sistema de sensores. Además, una luz roja L1 conectada en la salida Q0.3 se activará a modo de alerta.



**Figura 3.26:** Conexión de entradas al PLC del ejemplo 3

En la Fig. 3.27, se puede observar el diagrama de conexiones de las salidas al PLC.



**Figura 3.27:** Conexión de salidas al PLC del ejemplo 3

## 3.6. Temporizadores, contadores y comparadores

### 3.6.1. Temporizadores

Uno de los elementos programables más utilizados es el conocido temporizador. Este permite medir el tiempo de encendido o de apagado de determinado circuito. Un factor importante en los temporizadores es la precisión del tiempo a medir. Es un elemento de tipo auxiliar, en el cual los contactos esperan un tiempo programado antes de cambiar de estado.

### 3.6.2. Tipo de temporizadores

Existen tipos de temporizadores en TIA Portal, los cuales se utilizan en función de las necesidades del usuario y el tipo de activación requerida. En la Fig. 3.28, se pueden ver los tipos.

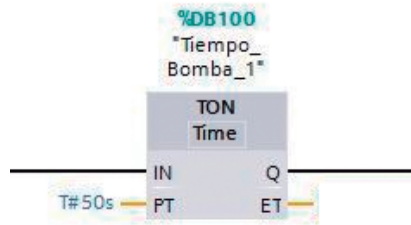
Temporizadores	
TP	Impulso
TON	Retardo al conectar
TOF	Retardo al desconectar
TONR	Acumulador de tiempo
(I) -(TP)-	Arrancar temporizador como impulso
(I) -(TON)-	Arrancar temporizador como retardo a la conexión
(I) -(TOF)-	Arrancar temporizador como retardo a la desconexión
(I) -(TONR)-	Acumulador de tiempo
(I) -(RT)-	Inicializa temporizador
(I) -(PT)-	Cargar tiempo

Figura 3.28: Tipos de temporizadores en TIA Portal

#### *Temporizador con retardo a la conexión: TON*

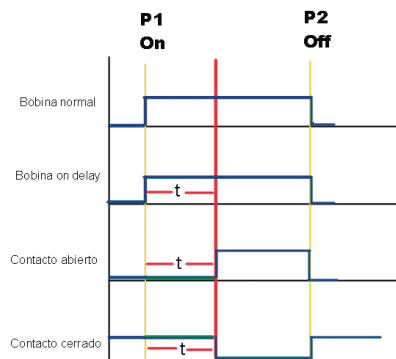
El temporizador con retardo a la conexión trabaja de la siguiente manera: una vez que la bobina ha sido activada, el tiempo empieza a correr; los contactos abiertos y cerrados están esperando que el tiempo programado transcurra para cambiar de estado. La simbología del temporizador se puede ver en la Fig. 3.29.

El funcionamiento de este temporizador se puede ver en el diagrama de tiempo de la Fig. 3.30. En el reglón 1 se ha diagramado el funcionamiento de una bobina de contactor o salida normal. Esta se activa con el pulsador P1 y se desactiva con el pulsador P2.



**Figura 3.29:** Bloque de programación de temporizador con retardo a la conexión

En el reglón 2 se observa el comportamiento de una bobina de un temporizador on-delay. Al accionar el pulsador P1, empieza a correr un tiempo “t”. En el reglón 3 y 4, se mira el funcionamiento de los contactos normalmente abierto y normalmente cerrado de un temporizador con retardo a la conexión; como se observa en el diagrama de tiempo, el cambio de estado de estos contactos se da cuando ha finalizado el tiempo “t”. La bobina y los contactos del temporizador pasan a su condición inicial solo cuando se ha presionado el pulsador P2 de apagado.



**Figura 3.30:** Diagrama de tiempos con retardo a la conexión

Para insertar un temporizador on-delay en el programa, se debe arrastrar y soltar el ícono correspondiente a TON en el segmento de TIA Portal a programar. Cuando se realiza esta acción, aparece una pantalla para nombrar al temporizador. Este nombre estará asociado a un bloque de datos; se puede mantener el nombre que aparece por defecto o asignar otro (ver Fig. 3.31).

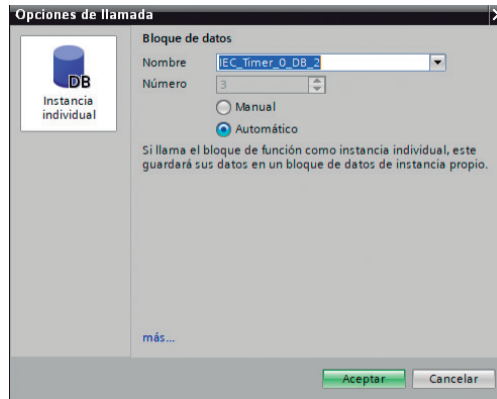


Figura 3.31: Bloque de datos temporizador

Una vez colocado el nombre, se da clic en *Aceptar*, en la pantalla de bloque de datos temporizador. Cuando se haya insertado el bloque, se solicita ingresar el tiempo asignado a este temporizador. En la Fig. 3.32, se puede visualizar el bloque del temporizador configurado con 10 milisegundos. El tiempo se puede programar en milisegundos, segundos, minutos u horas.

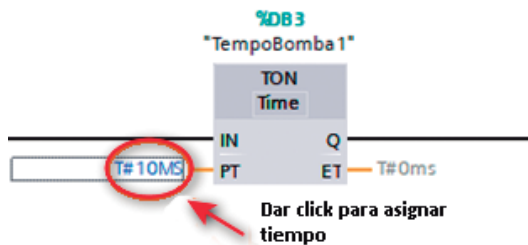


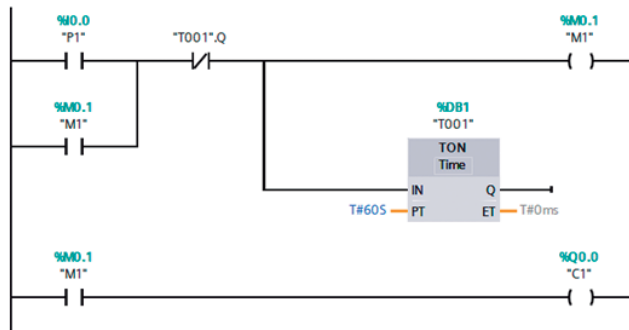
Figura 3.32: Temporizador en línea de programación Ladder



### Ejemplo temporizador on-delay

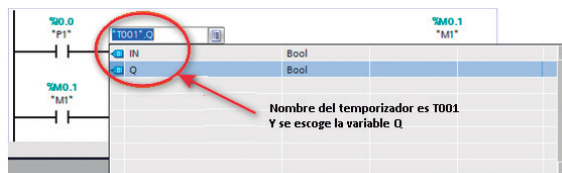
Se solicita programar en Ladder el encendido manual y apagado automático de una lámpara. Una vez encendida la lámpara, deberá permanecer en ese estado por 60 segundos luego de lo cual se apagará automáticamente.

*Solución:* La solución a este problema se presenta en la Fig. 3.33.



**Figura 3.33:** Apagado automático de una lámpara

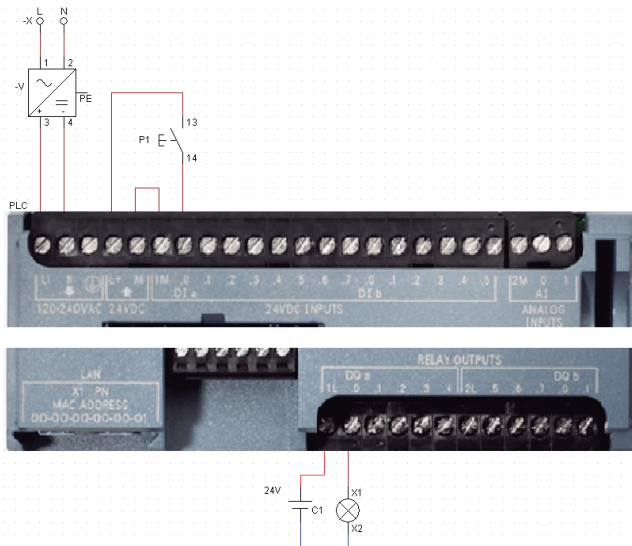
En el primer renglón de la Fig. 3.33 se puede observar un contacto normalmente abierto I0.0, el cual representa el pulsador que da inicio al programa, en serie, un contacto cerrado T001.Q, el cual representa un contacto normalmente cerrado del temporizador on-delay. Cuando se escoge el nombre de la variable, se debe seleccionar primero el temporizador y después la variable Q (ver Fig. 3.34) y en serie a una salida M0.1, la cual funcionará como auxiliar para mantener memorizado al temporizador y para activar la salida Q0.0.

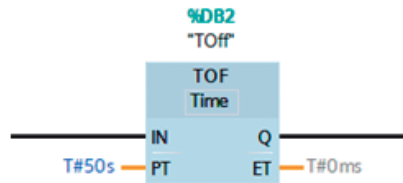


**Figura 3.34:** Configuración de contactos de temporizador

El temporizador, al estar conectado en paralelo a M0.1, se encenderá al mismo tiempo. Una vez encendido, se puede observar que empieza a contar el tiempo programado. Terminado este tiempo, el contacto cerrado se abrirá, desconectando la salida M0.1 y el temporizador. Esta acción apagará a todo el sistema, incluida la lámpara conectada a la salida Q0.0.

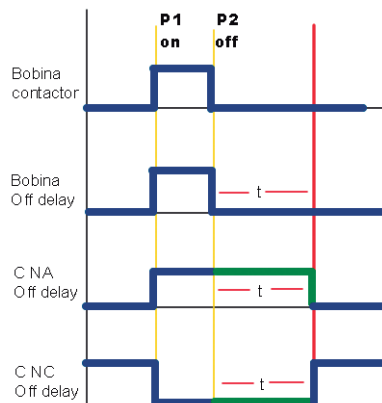
En la Fig. 3.35 se observa el diagrama de conexiones de entradas y salidas al PLC.





**Figura 3.36:** Bloque de programación de temporizador con retardo a la desconexión

El funcionamiento de este temporizador se puede ver en el diagrama de tiempo de la Fig. 3.37. En el reglón 1, se ha diagramado el funcionamiento de una bobina de contactor o salida normal, la cual se activa con el pulsador P1 y se desactiva con el pulsador P2. En el reglón 2 se observa el comportamiento de una bobina de un temporizador off-delay. Al accionar el pulsador P2, empieza a correr un tiempo “t”. En el reglón 3 y 4 se observa el funcionamiento de los contactos normalmente abierto y normalmente cerrado de un temporizador con retardo a la conexión; en el diagrama de tiempo, el cambio de estado de estos contactos se da cuando ha finalizado el tiempo “t”. La bobina y los contactos del temporizador pasan a su condición inicial solo cuando se ha presionado el pulsador P2 de apagado.



**Figura 3.37:** Diagrama de tiempos con retardo a la desconexión

Para insertar un temporizador off-delay en el programa, se debe arrastrar y soltar el ícono correspondiente a TOF en el segmento de TIA Portal a programar. Entonces, aparece una pantalla para nombrar al temporizador. Este nombre estará asociado a un bloque de datos (ver Fig. 3.31). Se puede mantener el que aparece por defecto o asignar otro.

Una vez nombrado, se da clic en *Aceptar* en la pantalla de bloque de datos temporizador. Cuando se haya insertado el bloque, se solicita ingresar el tiempo asignado a este temporizador. En la Fig. 3.36, se puede visualizar el bloque del temporizador configurado con 50 segundos. El tiempo se puede programar en milisegundos, segundos, minutos u horas.

### *Ejemplo temporizador off-delay*

Se solicita programar en Ladder el encendido manual y apagado automático de una lámpara. Una vez encendida la lámpara, deberá permanecer en ese estado por 5 minutos. Luego, se apagará automáticamente. Este ejemplo es muy parecido al ejemplo de temporizadores on-delay, pero, en este caso, se solicita diseñar con un temporizador off-delay.

*Solución:* La solución a este problema se presenta en la Fig. 3.38.

En el primer renglón de la Fig. 3.38 se puede observar un contacto normalmente abierto I0.0, el cual representa el pulsador piloto que da inicio al programa y en serie el temporizador off-delay. El diagrama representa un sistema piloto; cuando suelte el pulsador, el temporizador off-delay empezará a contar el tiempo programado. El contacto normalmente abierto de T001 permanecerá cerrado 5 minutos y solo cuando termine de contar, la salida Q0.0 se apagará. Entonces, se puede observar que empieza a contar el tiempo programado y una vez terminado, el contacto pasará a su estado normal, desconectando la salida Q0.0. El diagrama de conexiones de entradas y salidas al PLC es similar al de la Fig. 3.35.

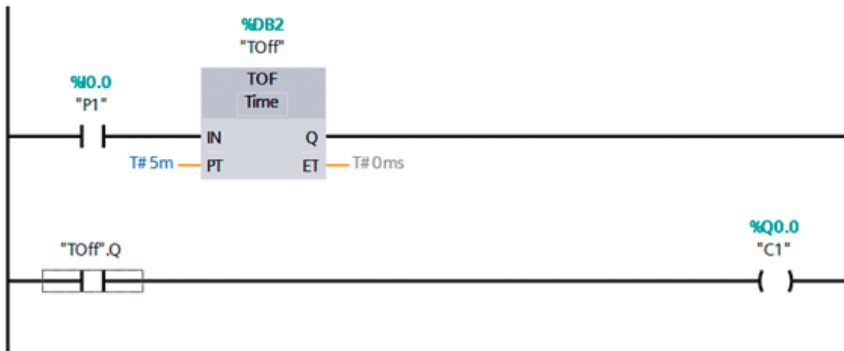


Figura 3.38: Apagado automático de una lámpara

### 3.6.3. Contadores

Los contadores son variables internas dentro del PLC; su funcionamiento se basa en el cambio lógico de 0 a 1 de una marca, entrada o salida. En TIA Portal existen varios modelos de contadores. En la Fig. 3.39 se pueden encontrar los tipos.

+3 Contadores	
CTU	Contador ascendente
CTD	Contador descendente
CTUD	Contador ascendente - descendente

Figura 3.39: Tipos de contadores en TIA Portal

#### *Contador ascendente CTU*

El contador ascendente permite incrementar el valor de conteo de la variable CU partiendo desde 0. El bloque de programación se puede ver en la Fig. 3.40.

Al insertar el bloque del contador, Tía Portal preguntará si se desea cambiar el nombre del bloque o se deja el que aparece por defecto. Ver Fig. 3.41. Una vez insertado el bloque, se puede escoger con qué tipo de datos se va a trabajar. Ver Fig. 3.42.

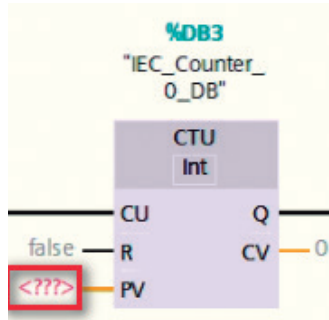


Figura 3.40: Contador tipo CTU



Figura 3.41: Bloque de llamada a contador ascendente

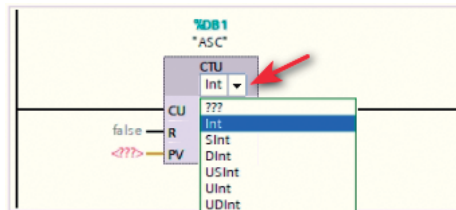


Figura 3.42: Selección de tipo de dato en contador ascendente

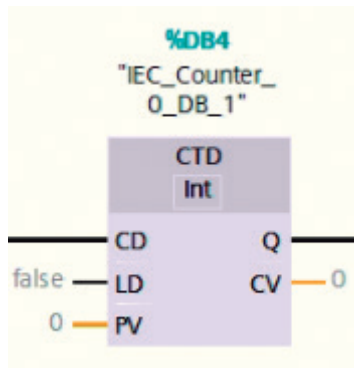
En el siguiente ejemplo se analizarán los datos que se necesitan para realizar un conteo de forma ascendente.

**Tabla 3.4:** Tipo de datos para uso en comparadores

<b>Int</b>	Entero
<b>DInt</b>	Doble entero
<b>Real</b>	Real
<b>Byte</b>	Byte
<b>Word</b>	Word
<b>DWord</b>	Double Word
<b>Time</b>	Tiempo

### *Contador descendente CTD*

El contador descendente permite disminuir el valor de conteo de la variable CD a partir del valor asignado en la variable PV. El bloque de programación se puede ver en la Fig. 3.43.



**Figura 3.43:** Contador tipo CTD

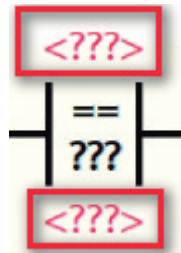
### 3.6.4. Comparadores

TIA Portal dispone de varias funciones de comparación, como igual, mayor, mayor igual, menor, menor igual y diferente de otro valor. La tarea que cumplen los comparadores es verificar si se cumple la comparación.

Los tipos de datos que se puede comparar se muestran en la Tabla 3.4.

#### *Nomenclatura comparadores*

Todos los comparadores necesitan dos valores para ser comparados. En la Fig. 3.44 se observa el bloque de comparación igual. Si el primer valor es igual al segundo, el bloque entrega un 1 lógico a la salida. Todos los bloques tienen la misma estructura.



**Figura 3.44:** Ejemplo de comprador igual

### 3.6.5. Ejemplos utilizando comparadores y contadores

Contador de personas que ingresan a un local.

Se requiere contar la cantidad de personas que ingresan a un local. En este ejemplo se contará estableciendo un máximo número de personas, de manera que, cuando llegue al máximo, la puerta ya no se pueda abrir.

En la Fig. 3.45 se puede ver el planteamiento del problema.



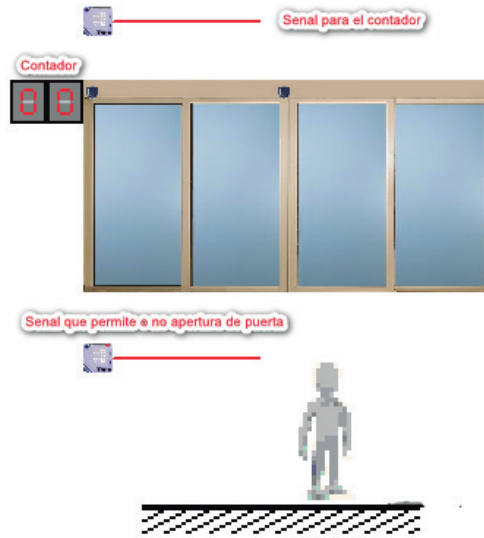


Figura 3.45: Bosquejo del problema

En la Fig. 3.46 se presenta el segmento 1 de la solución al ejercicio. La entrada I0.0 representa un sensor que permitirá enviar la señal al contador a través de pin CU cada vez que ingrese una persona. La entrada I0.1 será un pulsador de *reset* para volver a las condiciones iniciales al contador.

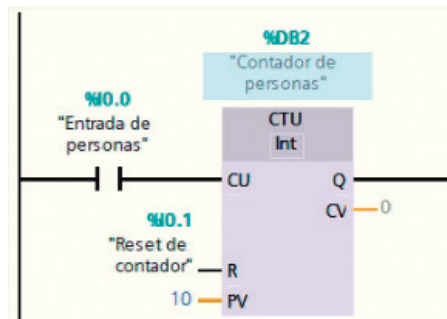


Figura 3.46: Programación de contador ascendente

En la Fig. 3.47 se presenta el segmento 2 de la solución al ejercicio. La salida Q0.0 será conectada a un contactor que permitirá que la puerta se abra cada vez que ingrese una persona, siempre y cuando no haya superado las 5 personas. Esto se realizará con la ayuda de un bloque comparador menor o igual. La entrada de este bloque será la persona que está parada frente a la puerta y esta se abrirá solo si hay espacio en el local; caso contrario, permanecerá cerrada.



Figura 3.47: Programación del ingreso de personas al local

## 3.7 Funciones de transferencia

En Tía Portal se encuentran también las funciones de tipo transferencia. En la Fig. 3.48 se puede ver un resumen de estas.

Transferencia	
MOVE	Copiar valor
Deserialize	Deserializar
Serialize	Serializar
MOVE_BLK	Copiar área
MOVE_BLK_VARIANT	Copiar área
UMOVE_BLK	Copiar área sin interrupciones
FILL_BLK	Rellenar área
UFILL_BLK	Rellenar área sin interrupciones
SCATTER	Dispersión de una secuencia de bits en bits individuales
SCATTER_BLK	Dispersión de elementos de un ARRAY of secuencia de bits en bits indiv
GATHER	Recopilación de bits individuales en una secuencia de bits
GATHER_BLK	Recopilación de bits individuales en varios elementos de un ARRAY of s.
SWAP	Cambiar disposición

Figura 3.48: Tipos de bloques de funciones de transferencia

### 3.7.1. Función MOVE

La instrucción MOVE descrita en el manual de usuario de Tía PORTAL como copiar valor, transfiere el valor de la entrada IN a la salida OUT. Pero, se debe verificar que los datos de la entrada y salida sean los mismos. Adicionalmente, esta instrucción permite transferir datos individuales a un tipo de datos CHAR o WCHAR. En resumen, la función MOVE permite transferir datos entre datos iguales; por ejemplo, BYTE con BYTE, WORD con WORD, etc., y hacer copia exacta del origen en el destino. También se pueden transferir datos diferentes; por ejemplo, DINT a INT. Para hacer este tipo de transferencia, se debe tener precaución y verificar que tienen el mismo tamaño o al menos que el valor de entrada sea menor que el valor de salida. El bloque de función MOVE se puede ver en la Fig. 3.49.

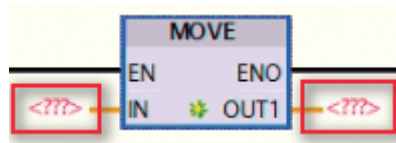
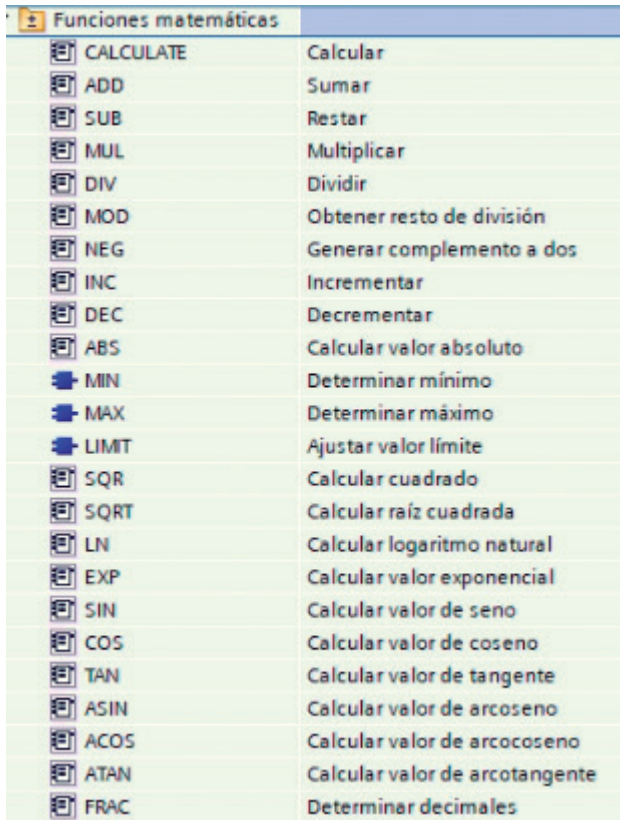


Figura 3.49: Bloque de función MOVE

## 3.8. Funciones matemáticas

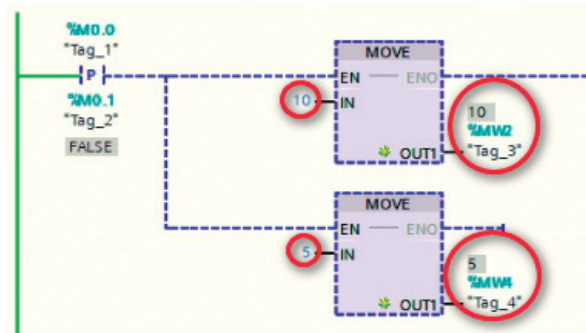
En TIA Portal se encuentra otro bloque de funciones; esta vez del tipo matemático. En la Fig. 3.50 se puede ver un resumen de todas las funciones matemáticas que se dispone: suma, resta, multiplicación y división.



Funciones matemáticas	
CALCULATE	Calcular
ADD	Sumar
SUB	Restar
MUL	Multiplicar
DIV	Dividir
MOD	Obtener resto de división
NEG	Generar complemento a dos
INC	Incrementar
DEC	Decrementar
ABS	Calcular valor absoluto
MIN	Determinar mínimo
MAX	Determinar máximo
LIMIT	Ajustar valor límite
SQR	Calcular cuadrado
SQRT	Calcular raíz cuadrada
LN	Calcular logaritmo natural
EXP	Calcular valor exponencial
SIN	Calcular valor de seno
COS	Calcular valor de coseno
TAN	Calcular valor de tangente
ASIN	Calcular valor de arcoseno
ACOS	Calcular valor de arcocoseno
ATAN	Calcular valor de arcotangente
FRAC	Determinar decimales

**Figura 3.50:** Tipos de bloques de funciones matemáticas

Para realizar las aplicaciones con los bloques de función matemática, primero se cambian los datos de las variables enteras a palabras (Word) MW2 y MW4 (2 bytes) utilizando la función MOVE. Esta acción se puede ver en la Fig. 3.51. Una vez que se hayan cambiado las funciones, se pueden insertar las operaciones que se necesiten sea suma, resta, multiplicación, etc.



**Figura 3.51:** Conversión de variables enteras a tipo Word usando MOVE

Existe un bloque de función completo llamado CALCULATE; permite realizar cualquier operación matemática. Esta instrucción se revisará más adelante.

### 3.9. Señales Analógicas

Una señal analógica es un valor que cambia de manera continua entre un máximo y un mínimo. Por ejemplo, una entre 0 y 5 voltios puede adoptar cualquier valor comprendido entre 0 y 5, por ejemplo, 3.7 V, 4.3 V; lo cual es muy diferente a una señal digital que solo puede tomar valor de 0 y 1. El PLC por sí solo no trabaja con señales analógicas, de manera que lo que hace es convertirla en digital.

Para realizar esta conversión, se plantea el siguiente ejemplo. Suponiendo una señal de 0 a 10 voltios, se requiere convertirla en una señal digital, se divide el valor de 10 V en cuatro partes, como se muestra en la Fig. 3.52.

Los bits correspondientes serían:

- Para valores de 0 V a 2.5 V, sería 00
- Para valores de 2.5 V a 5 V, sería 10
- Para valores de 5 V a 7.5 V, sería 01
- Para valores de 7.5 V a 10 V, sería 11

Pero estos valores no son tan precisos; es decir, el PLC no encontraría diferencias entre 2.5 V y 2.7 V. Por lo que se debe aumentar la resolución de la señal. El PLC dispone de dos entradas de 16 bits, el canal 0 (IW64) y el canal 1 (IW66). Esto significa que puede tomar valores entre -32768 y 32767. Pero, para realizar la conversión, se debe considerar que 0 equivale a 0 V y 27648 equivale a 10 V; con estos valores se escala cualquier valor.

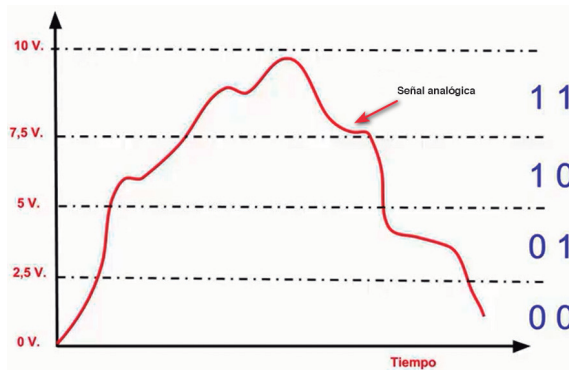


Figura 3.52: Señal analógica de 0 a 10 V

### 3.9.1. Entradas y salidas analógicas

#### *Entradas analógicas*

En los PLC's se dispone de módulos e interfaces que admiten señales de entrada valores de voltaje (0-10 V) o corriente (0-20 mA o 4-20 mA) dentro de un rango. Este número será guardado en el PLC. Los módulos se encargan de traducir unas señales de voltaje o corriente proveniente de un sensor de temperatura, velocidad, humedad, posición, para que el PLC lo pueda interpretar.

#### *Salidas analógicas*

Los módulos de salidas analógicas permiten que el valor de una variable interna del PLC se convierta en una señal de voltaje o corriente.

Internamente, el PLC realiza una conversión digital-análoga, ya que —como dijimos antes— el PLC solo trabaja con señales digitales. A través de estas salidas se pueden controlar elementos como variadores de velocidad, válvulas proporcionales, reguladores de temperatura, etc.

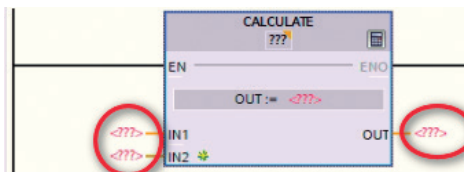
### 3.9.2. Como escalar señales de entradas analógicas

Se sabe que el PLC lee valores de 0 a 27648. Si se requiere leer, por ejemplo, la señal de un sensor de presión que entrega valores entre 3 a 15 psi, para adquirir esta señal a través del modelo de entradas analógicas del PLC, se puede utilizar una ecuación matemática conocida como ecuación de la recta que pasa por 2 puntos.

#### *Calculate*

TIA Portal dispone del bloque de función CALCULATE. (Ver Fig. 3.53), el cual dispone de los siguientes ítems:

1. IN1 - Resolución mínima: Es el valor mínimo de la escala de la tarjeta.
2. IN2 - Resolución máxima: Es el valor máximo de la escala de la tarjeta.
3. IN3 - Sensor: Señal del sensor.
4. IN4 - Sensor mínimo: Es el valor mínimo que entrega el sensor.
5. IN5 - Sensor máximo: Es el valor máximo que entrega el sensor.
6. OUT - Escalado: Entrega el valor escalado de la variable de entrada.



**Figura 3.53:** Bloque de función Calculate

La instrucción CALCULATE es usada como una calculadora. En ella se realizan una serie de operaciones matemáticas. También se

prueban funciones trigonométricas, exponentes, negativos, sumas, restas, entre otras. En la Fig. 3.54 se puede ingresar cualquier ecuación matemática para resolver.

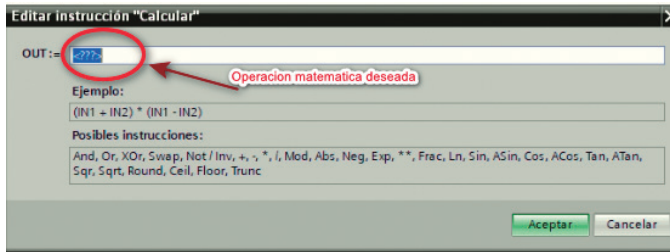


Figura 3.54: Editar la instrucción CALCULATE

### Scale X y Norm X

Utilizando estos dos bloques de programación, se puede realizar el escalado de las variables de entrada y salida.

- Convertir los valores de 5530 a 27648 a valores entre 0 y 1
- Con los valores entre 0 y 1, obtener el valor proporcional, por ejemplo, entre 0 y 6 psi.

### Norm X

La instrucción NORM X presenta el valor de entrada en una escala lineal. Los parámetros MIN y MAX definen el rango de valores que se verán en la escala. En la Fig. 3.55 se puede ver el bloque de función.

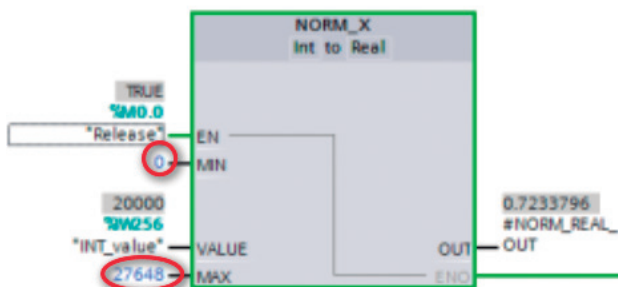
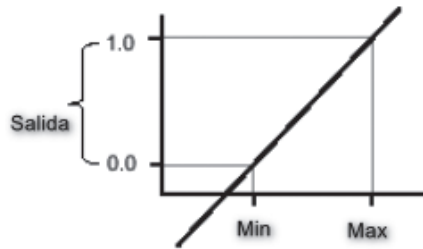


Figura 3.55: Instrucción NORM X



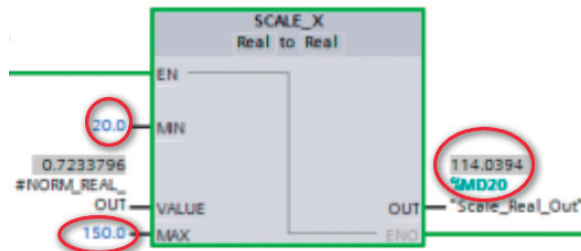
En la Fig. 3.56 se puede visualizar el comportamiento de la salida de esta instrucción. Si el valor a normalizar es MIN, se obtiene 0 a la salida; si es MAX, se obtiene 1 a la salida.



**Figura 3.56:** Escala lineal de valores normalizados con NORM X

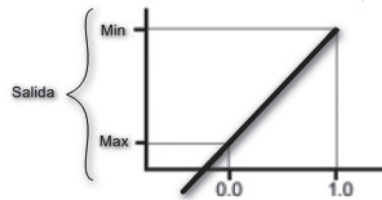
### Scale X

La instrucción SCALE X escala el valor de entrada entre un determinado rango de valores. Se debe declarar entre qué valores MIN y MAX se quiere escalar y el resultado es un valor entero entregado por la salida OUT. El bloque de función se puede ver en la Fig. 3.57.



**Figura 3.57:** Instrucción SCALE X

En la Fig. 3.58 se puede visualizar el comportamiento de la salida de esta instrucción. El valor entre 0 y 1 que entrega el bloque NORM-X permitirá escalar entre MIN y MAX a la salida OUT de la instrucción de SCALE X.



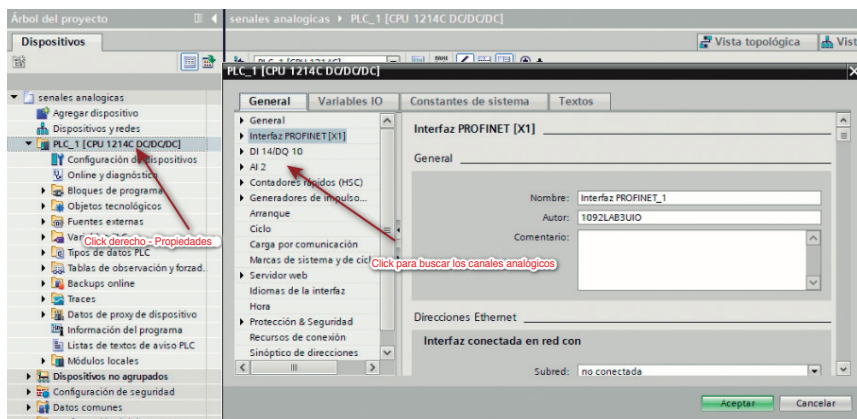
**Figura 3.58:** Escala lineal de valores normalizados con SCALE X

### *Ejemplo de entrada analógica*

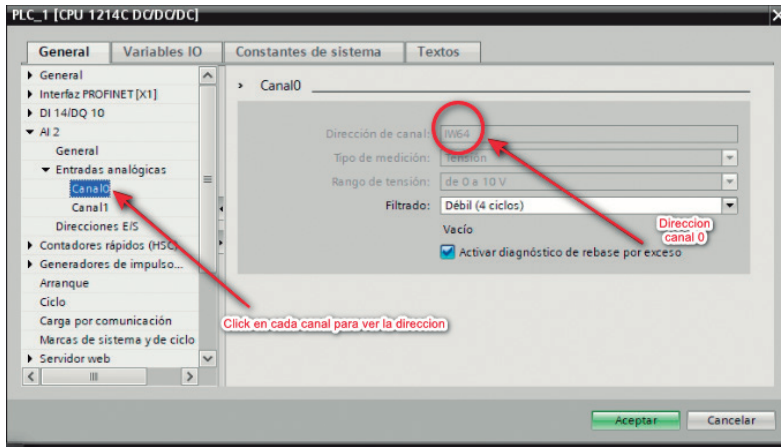
A través de un PLC se desea realizar un control de presión con un sensor que varía de 0 a 6 psi y entrega una salida de 0 a 10 V, donde 0 V corresponde a 0 psi y 6 psi a 10 V; también se explica que el sensor se conecta a dos hilos.

Antes de comenzar, se debe verificar que el PLC disponga de entradas analógicas. Adicionalmente, se debe identificar el direccionamiento de la entrada. En este caso, IWxx significa entrada física tipo Word.

Se debe ingresar a TIA Portal a las propiedades del PLC (ver Fig. 3.59). En la ventana que aparece, se debe seleccionar la opción AI2 y buscar entradas analógicas, para verificar la dirección del canal (ver Fig. 3.60). Dando clic en cada canal ya sea 0 o 1 se conoce la dirección del canal de estas entradas.



**Figura 3.59:** Ventana para buscar la dirección del canal analógico



**Figura 3.60:** Mirar canales y direcciones de entradas analógicas

La dirección de los canales analógicos en esta configuración son IW64 y IW66. Ahora, se deben colocar las instrucciones de acondicionamiento NORM y SCALE.

Recordar que las dos trabajan de la mano.

Para escalar, en este caso, se va a comenzar configurando NORM X y se selecciona en la entrada variables tipo INT o WORD.

1. MIN Poner la resolución mínima del canal.
2. MAX Poner la resolución máxima del canal (27648 en SIMATIC).
3. VALUE Canal de lectura.
4. OUT ss valor REAL por unidad.

En la Fig. 3.61 se puede ver la configuración del bloque SCALE X, que permitirá obtener señales adquiridas del sensor en valores escalados. Es decir, los valores que ingresan de 0 a 6 psi se convertirán a valores escalados entre 0 y 10 V.

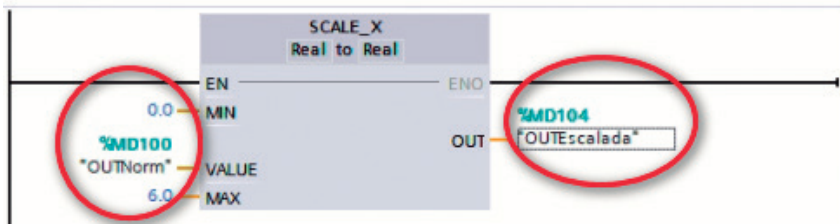


Figura 3.61: Bloque SCALE X configurado

### Cómo crear una subrutina

Otra forma de escalar valores analógicos es utilizar subrutinas. Aunque estas tienen cientos de aplicaciones, a continuación, se presenta un ejemplo para adquirir una señal analógica.

El bloque principal de organización (OB) permite realizar funciones principales, pero se tienen otros tipos, como bloques de función (FB), función (FC) y bloque de datos (DB).

Para agregar una nueva subrutina, se deben seguir los siguientes pasos: Ingresar y dar doble clic en *Agregar nuevo bloque* (ver Fig. 3.62).

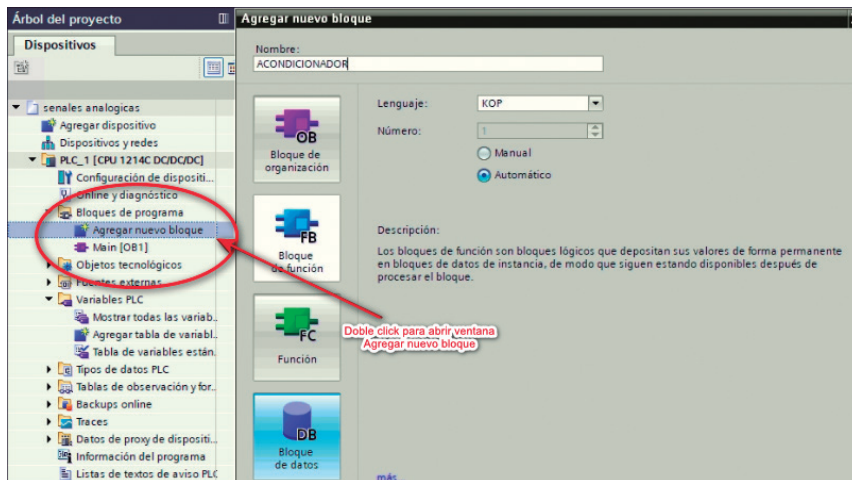


Figura 3.62: Agregar nuevo bloque

Se puede observar que en el árbol del proyecto se creó un nuevo bloque celeste, debajo del bloque principal lila. Se despliega un área de programación similar al bloque principal, pero corresponde a la programación de la subrutina ACONDICIONADOR (ver Fig. 3.63).

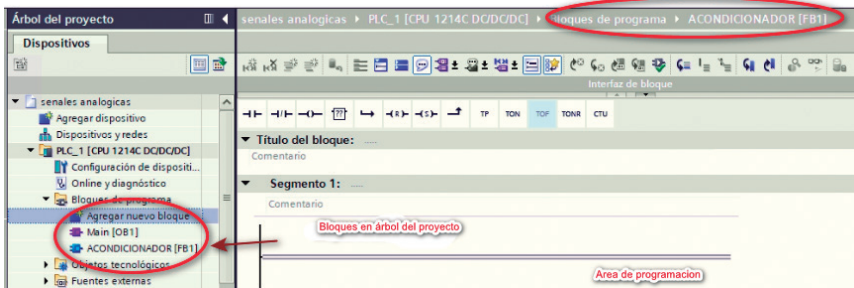


Figura 3.63: Creación de subrutina

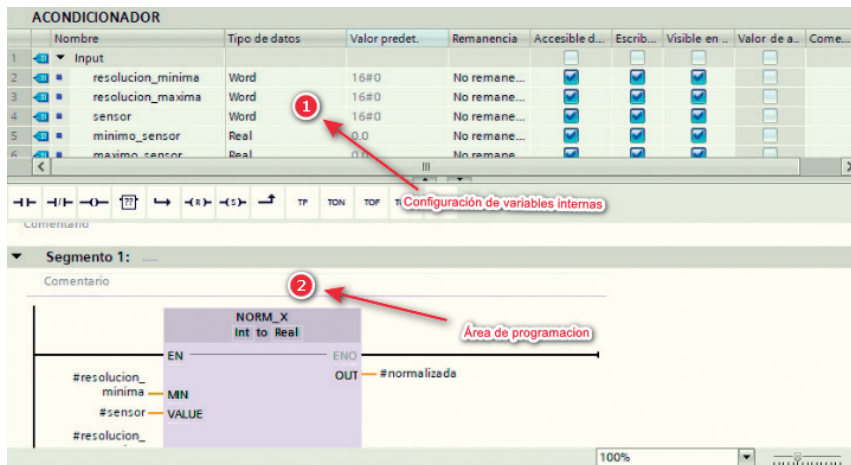
Se necesita configurar las variables internas de entradas y salida para esta subrutina. Así, para hacer el escalado de una entrada, se realizará la declaración de variables mostrada en la Fig. 3.64.

ACONDICIONADOR		
	Nombre	Tipo de datos
1	Input	
2	resolucion_minima	Word
3	resolucion_maxima	Word
4	sensor	Word
5	minimo_sensor	Real
6	maximo_sensor	Real
7	<Agregar>	
2	Output	
9	escalado	Real
10	<Agregar>	
11	InOut	
12	<Agregar>	
3	Static	
14	normalizada	Real
15	<Agregar>	

Figura 3.64: Declaración de variables para la subrutina

En la Fig. 3.64 se pueden observar tres bloques de variables internas:

- ① Serán las variables de entrada, necesarias para realizar el escalado.
- ② Será la variable de salida de la subrutina, en este caso, la variable escalada.
- ③ Es una variable que actúa como entrada y salida; en este caso, la salida normalizada del bloque NORM X, que también se comporta como entrada para el bloque SCALE X.



**Figura 3.65:** Área de programación de la subrutina

En la Fig. 3.65, se ven las áreas donde se registran las variables internas y el área de programación.

Para la programación de la subrutina se seguirá el mismo procedimiento visto en el uso de los bloques SCALE y NORM.

En la Fig. 3.66 se puede observar la configuración y declaración de variables para obtener un bloque denominado “acondicionador”.

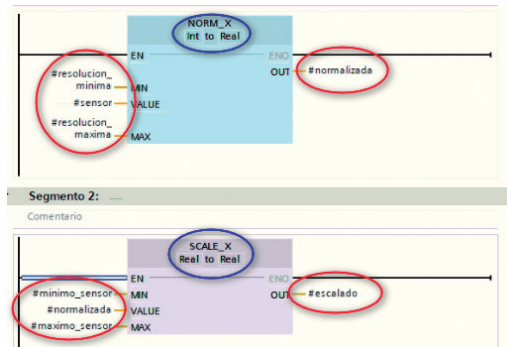


Figura 3.66: Programación de la subrutina “acondicionador”

Una vez programada la subrutina, se puede regresar al bloque de programación principal MAIN, para insertar la subrutina creada.

En la Fig. 3.67 se revisa lo siguiente:

- ① Se ha creado un bloque debajo de main denominado “acondicionador”.
- ② Se debe arrastrar ese bloque hasta insertarlo en el de programación.
- ③ En la parte inferior, se puede navegar entre la ventana de programación principal main y la de la subrutina.

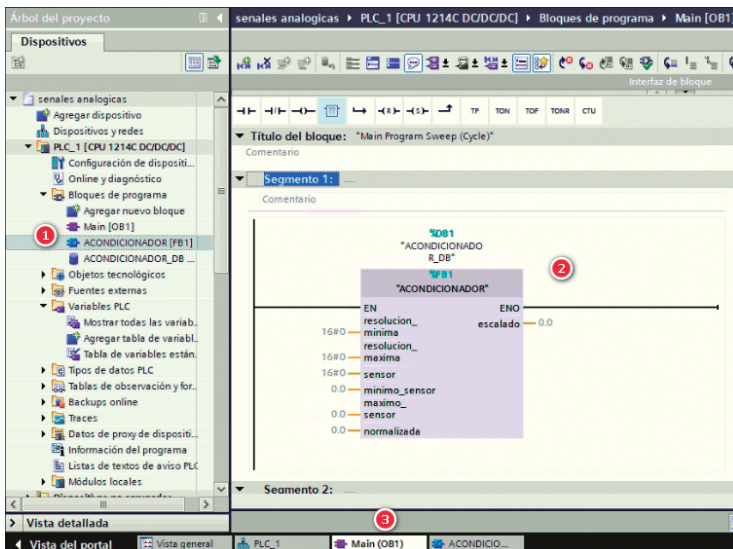


Figura 3.67: Bloque de subrutina en main

En la Fig. 3.68 se puede ver el bloque de subrutina programado, el cual solo requiere valores numéricos; el mismo bloque se puede usar para adquirir la señal de dos o más sensores.

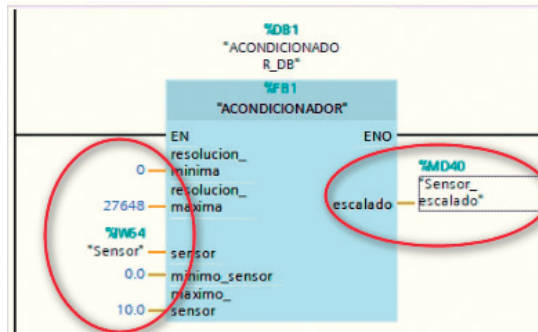


Figura 3.68: Subrutina “acondicionador”

### 3.9.3. Cómo configurar salidas analógicas

Se debe verificar que el dispositivo PLC disponga de salidas analógicas: se ingresa al árbol de proyecto, dando clic derecho en el dispositivo PLC y eligiendo *Propiedades* (ver Fig. 3.69).

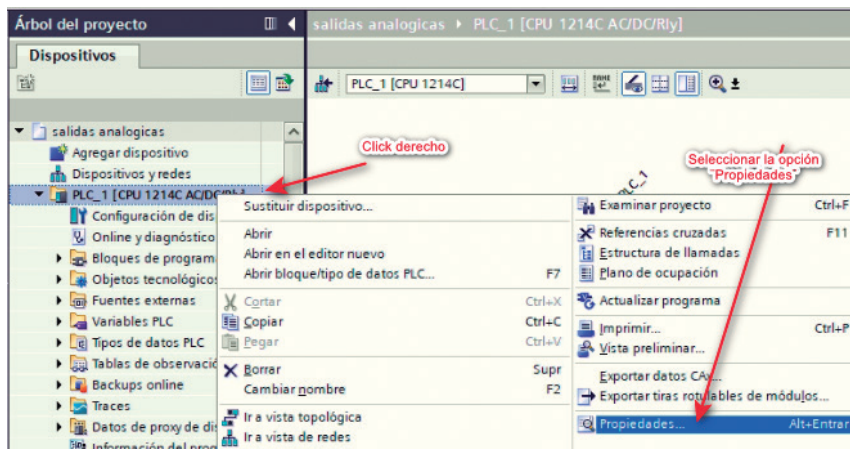


Figura 3.69: Ver direcciones de salidas analógicas del PLC



En el caso de PLC CPU 1214C DC/DC/DC, no dispone directamente de un módulo de salidas analógicas, así que se lo insertará. Para agregar, se expande la ventana de Catálogo de Hardware, y se escoge módulo AI/AQ. Luego, se da doble clic para agregar al CPU principal el módulo de salida analógica (ver Fig. 3.70).

En este caso, este módulo dispone de dos canales con dirección QW96 y QW98. Ahora, se debe investigar qué configuración tiene el actuador que se utilizará debido a que la salida analógica puede estar en tensión o en corriente, lo cual se debe seleccionar en el menú de opciones del dispositivo (ver Fig. 3.71).

Se desplegará inmediatamente una ventana donde se observa la configuración de las entradas y salidas analógicas. Se puede ver la dirección y cambiar el tipo de señal, sea de tensión o corriente dependiendo de la necesidad (ver Fig. 3.72).

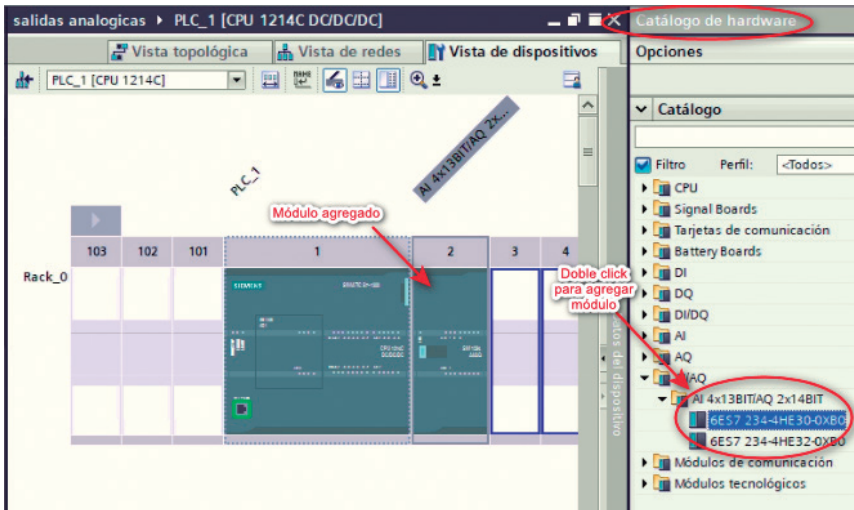


Figura 3.70: Ver direcciones de salidas analógicas del PLC

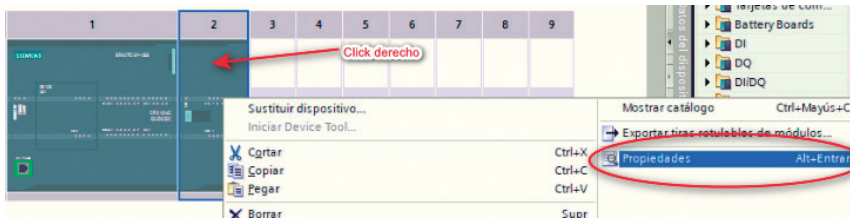


Figura 3.71: Propiedades del bloque de salidas analógicas

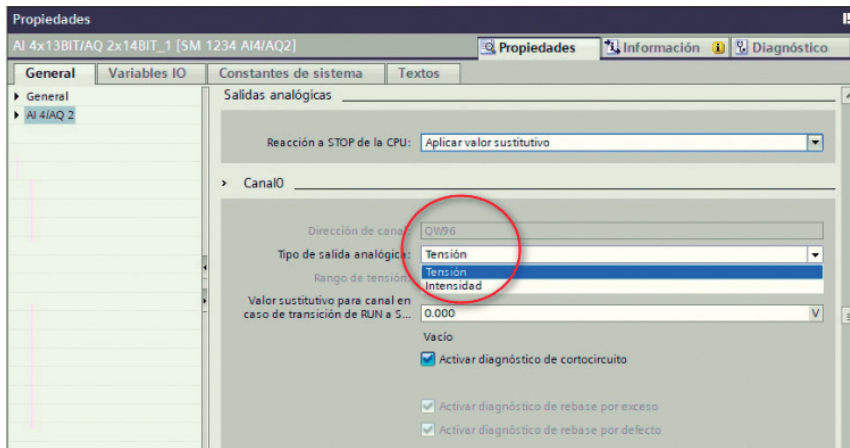


Figura 3.72: Ventana de propiedades del bloque de salidas analógicas

### *Ejemplo de salidas analógicas*

Se debe revisar la documentación de la salida a utilizar; por ejemplo, si es una válvula, se debe verificar el tipo de señal que maneja. Existen las que trabajan con señales de 0 a 20 mA.

Para controlar esta válvula, se va a crear un bloque de función denominado “salida analógica”. En este se programará el control de la válvula que funciona con valores de 0 a 20 mA, pero el PLC no maneja esos valores, el controlador maneja valores entre 0 y 27648. Así que, para este caso, 0 significa 0 mA y 27648 significa 20 mA. Se utilizarán los bloques NORM X y SCALE X (ver Fig. 3.73).

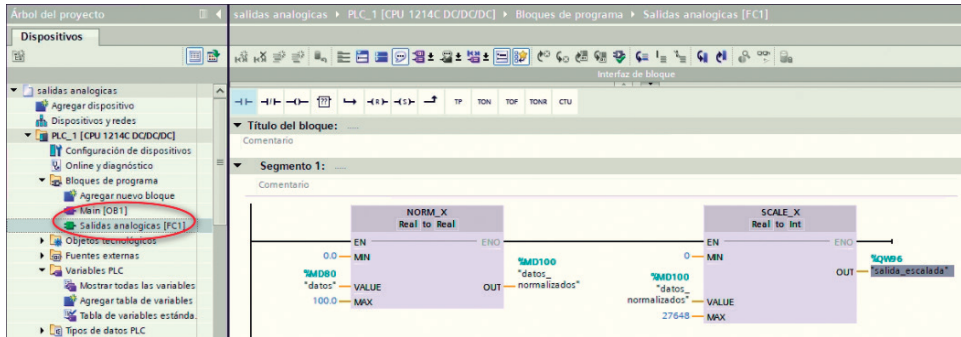


Figura 3.73: Configuración de salida analógica

## 3.10. Ejercicios propuestos

1. Realice la programación en Ladder, el diagrama de conexiones de entradas y salidas al PLC para un sistema de control manual de 3 contactores. Los contactores irán conectados a las salidas Q0.0, Q0.1 y Q0.2 del PLC. Se requiere que se enciendan manualmente por medio de pulsadores, siguiendo un orden estricto. Se deberá encender primero el contactor conectado a la salida Q0.1, después Q0.2 y, finalmente, Q0.0. El apagado también será de forma manual por medio de pulsadores y de la siguiente forma: primero, la salida Q0.2, después la Q0.1 y, finalmente, la Q0.0.
2. Revise el ejercicio del molino de café presentado en este capítulo, modifique el programa para que el sistema tenga 2 funciones, modo manual y modo automático, a través de un selector de dos posiciones. En el modo automático funcionará de la misma forma, pero en el manual deberá hacerlo a través de pulsadores de marcha y paro. Estos pulsadores accionarán cada motor, independientemente de la acción de los sensores.
3. Realice la programación en Ladder, diagrama de conexiones de entradas y salidas al PLC para una mezcladora que tiene

funcionamiento manual; se dispondrá para este proceso de 4 pulsadores P1, P2, P3, P4. El pulsador P1 permite que el motor de la mezcladora gire en sentido horario; el P2 permite que la mezcladora se detenga; el pulsador P3 permite que el motor de la mezcladora gire en sentido antihorario y P4 que la mezcladora se detenga. En caso de sobrecarga, el motor deberá detenerse.

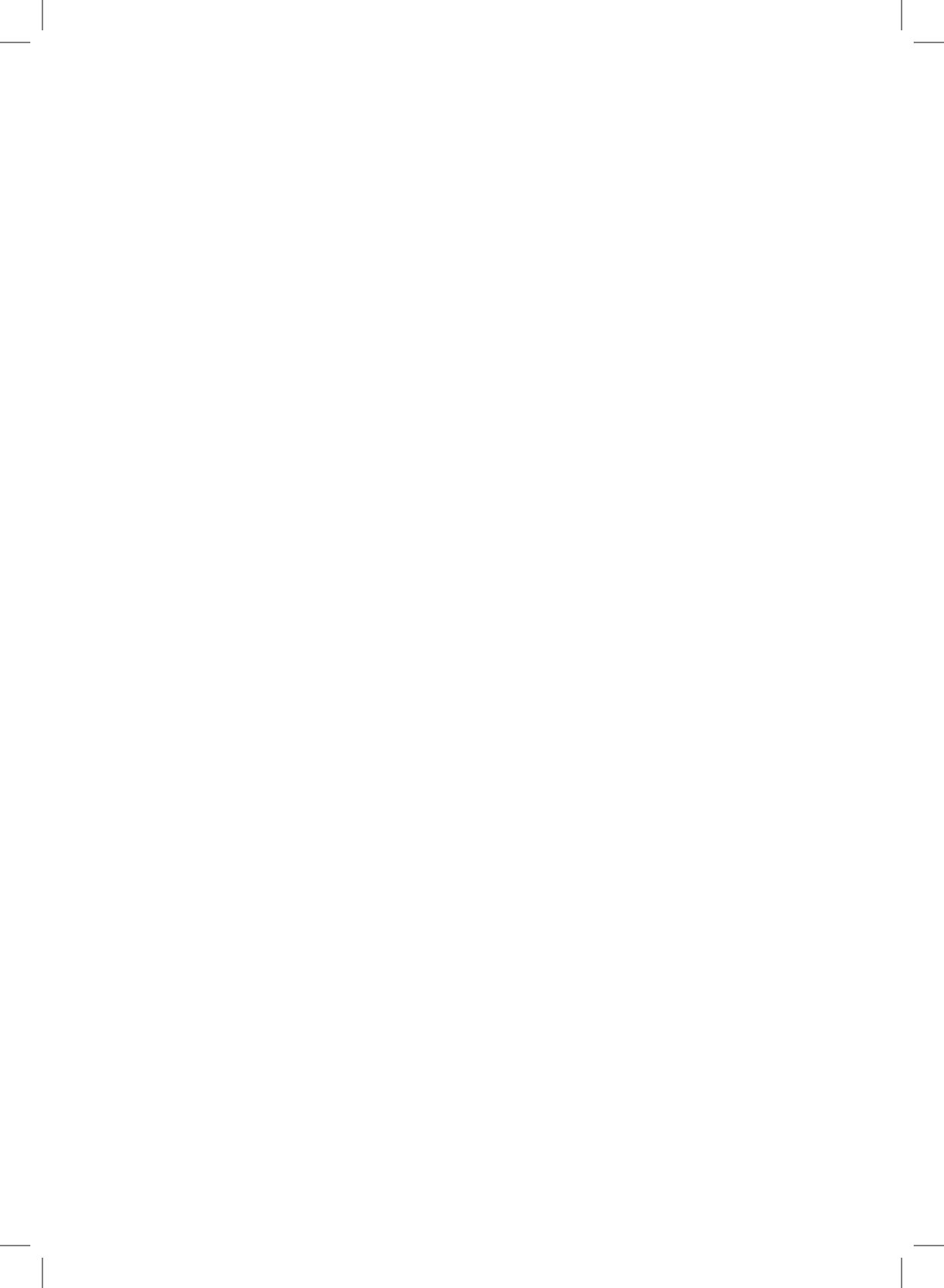
4. Realice la programación en Ladder, diagrama de conexiones de entradas y salidas al PLC, para el siguiente problema: se dispone de dos bombas de agua trifásicas que alimentan a un edificio. El tanque dispone de un sensor de nivel el cual se activa cuando el tanque está vacío y se desactiva cuando está lleno; las bombas funcionan de forma alternada: cuando se vacíe por primera vez, se activará la bomba 1 cuando se active el sensor por segunda vez, se activará la bomba 2; así, alternada y sucesivamente.
5. Se requiere automatizar un paso con barrera. Este consta de los siguientes elementos: un motor de corriente directa, dos finales de carrera, un pulsador normalmente abierto, un pulsador normalmente cerrado y dos luces una verde y una roja. El sistema funcionará de la siguiente forma:
  - La barrera debe iniciar cerrada y será detectada por el final de carrera que se llamará Fcerrado.
  - Si alguien desea pasar, se deberá presionar el pulsador normalmente abierto P1 y la barrea subirá y permanecerá abierta durante 10 segundos. Este tiempo correrá cuando sea activado el final de carrera Fabierta. Luego de lo cual regresará a su posición inicial.
  - Las luces se utilizarán a modo de semáforo. Cuando la puerta esté completamente abierta, se encenderá la luz verde; caso contrario, en cualquier otra condición, se encenderá la roja.
  - El pulsador normalmente cerrado será utilizado como paro de emergencia.

6. Se requiere automatizar una mezcladora de líquidos que cumpla las siguientes condiciones:

- El proceso inicia con un pulsador P1, el cual permite que ingrese el líquido 1 a través de una válvula de líquidos Va. La válvula se cerrará cuando el tanque alcance el 50 % de su capacidad; este porcentaje será detectado por medio de un sensor de nivel SM.
- Con el sensor de nivel SM activado, se procede a encender una mezcladora en sentido horario por 50 segundos, luego de lo cual se apagará.
- A continuación de la primera mezcla, se activará la válvula de líquidos 2 denominada Vb. Esta permitirá llenar el tanque hasta el 90 % de su capacidad, lo cual será detectado por un sensor de nivel SH; a continuación, la válvula se cerrará.
- Con el tanque lleno al 90 % de su capacidad, iniciará el proceso de mezcla. El motor de la mezcladora inicia en sentido horario por 40 segundos, se detiene y espera por 10 segundos, invierte su giro y mezcla en sentido antihorario por 40 segundos más.
- Finalizada la mezcla, se activará una válvula Vc de desfogue, la cual se cerrará una vez que el tanque esté vacío.
- En caso de sobrecarga del motor de mezcla, la válvula Vc deberá activarse permitiendo el desfogue total de la mezcla y regresando el proceso a sus condiciones iniciales.



**Capítulo 4**  
**Lenguaje de programación**  
**GRAFCET**





## 4.1. Introducción

Un sistema de control tiene diferentes alternativas para su implementación, por ejemplo: álgebra booleana, diagramas de escalera, diagramas de transiciones, flujogramas, etc. Sin embargo, estos métodos requieren de ciertos conocimientos técnicos que pueden dificultar su aprendizaje. A raíz de esta necesidad, en 1979 un grupo de investigación francés reunió académicos e industriales, denominado AFCET (acrónimo de French Association for Economical and Applied Cybernetics) e inventó GRAFCET bajo la premisa de ser un lenguaje de programación robusto y aplicable a problemas de la vida real [22].

GRAFCET está definido bajo el estándar IEC 848 [23] como “Sequencial Function Charts” (SCF). Esta es una guía de su lógica de programación, soporta el manejo de funciones booleanas; sin embargo, funciones más complejas como PID no se encuentran definidas. Entre sus ventajas tenemos:

- Facilidad para traducirlo a lenguajes de bajo nivel, como LADDER, LIST o Assembler.
- La programación en GRAFCET es gráfica, compacta y de sintaxis simple, tal que 1 página de LADDER puede ser equivalente a más de 5 de LADDER.
- Su metodología de diseño es top-down, lo cual facilita la identificación y depuración de errores.
- Al existir diferentes niveles de especificación, GRAFCET permite incluir información técnica de sensores y actuadores en el mismo programa.

- Entre el personal técnico encargado del proceso de automatización, resulta más fácil de explicar e interpretar. Además, puede aportar a la mejora del automatismo. No es necesario un nivel avanzado de conocimientos de programación, al ser un lenguaje de descripción gráfica.

## 4.2. Niveles de descripción

Un diagrama puede definirse en 3 niveles de descripción, lo que según [24] facilita el flujo de información desde un nivel operativo hasta un nivel de especificación tecnológica.

1. **Nivel 1. Descripción funcional:** se desarrolla el diagrama GRAFCET sin detallar o de forma independiente de la tecnología de los sensores y actuadores. En este nivel se describen las acciones que el automatismo debe efectuar.
2. **Nivel 2. Descripción tecnológica:** en el diagrama GRAFCET se detallan las señales de los sensores y actuadores que afectan al automatismo.
3. **Nivel 3. Descripción operativa:** se describen las señales del PLC de entrada y salida conectadas a los sensores y actuadores externos, así como marcas internas que serán utilizada para el control.

## 4.3. Elementos de GRAFCET

### 4.3.1. Etapas

Las etapas simbolizan estados discretos del automatismo. Un programa está compuesto por una serie de etapas que representan la secuencia

de evolución del proceso [25]. Existen dos tipos: las etapas normales y la o las etapas iniciales (ver Fig. 4.1). Entre sus características tenemos:

- Durante la ejecución, las etapas pueden estar activadas o desactivadas.
- Todo programa debe tener al menos una etapa inicial.
- Las etapas iniciales están activas cuando se inicia la ejecución del programa.
- Una vez iniciada la ejecución del programa, las etapas iniciales tienen el mismo comportamiento que las etapas normales.
- Las etapas normales se simbolizan con un cuadrado, mientras que las etapas iniciales, con doble cuadrado.

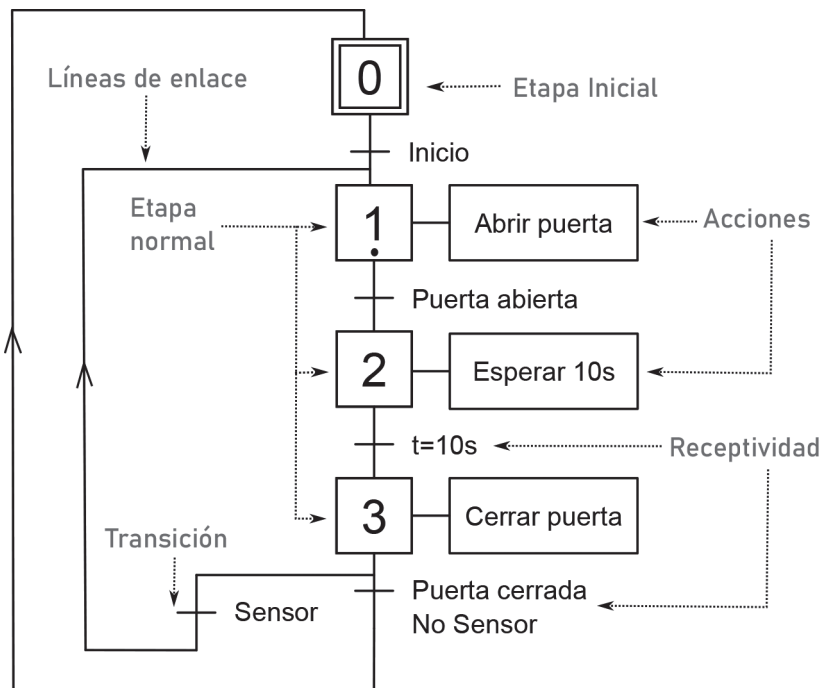
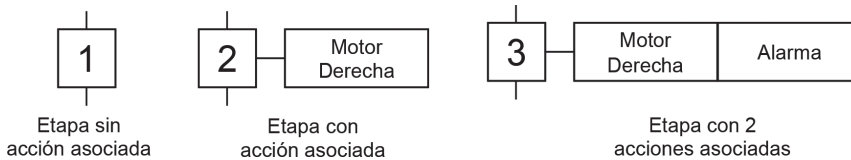


Figura 4.1: Soluciones de automatización

- Todas las etapas deben estar numeradas y su índice no se puede repetir en otra. La numeración no implica un orden determinado.
- Las etapas pueden tener una o varias acciones asociadas. Sin embargo, también se puede dar el caso de etapas sin acciones asociadas (ver Fig. 4.2).



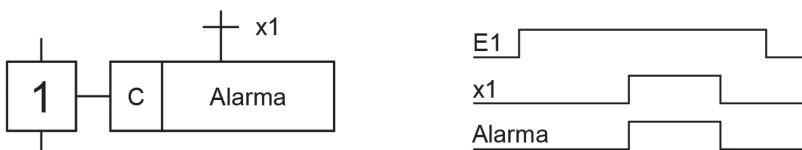
**Figura 4.2:** Etapas con y sin acciones asociadas

- Etapas sin acciones asociadas sirven para introducir estados de espera o de ayuda para cumplir con las reglas de programación de GRAFCET.

### 4.3.2. Acciones asociadas

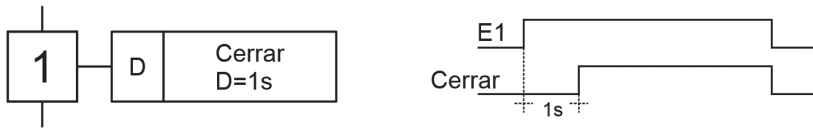
Se representan con un rectángulo adjunto a una etapa. Una acción asociada simboliza la activación de una o varias salidas, misma que permanecerá activa mientras la etapa esté activa.

- Los elementos definidos en las acciones asociadas se activan mientras la etapa relacionada esté activa. La norma IEC 848 [23] define los siguientes tipos de acciones:
  - (C) Acción condicionada, es otra forma de representar una acción condicionada.



**Figura 4.3:** Acción condicionada

- (D) Acción retardada, la acción se activa luego de un retardo.



**Figura 4.4:** Acción retardada

- (L) Acción limitada, la acción se activa con la etapa, pero su activación es limitada a un periodo temporal definido, siempre y cuando siga activa la etapa.



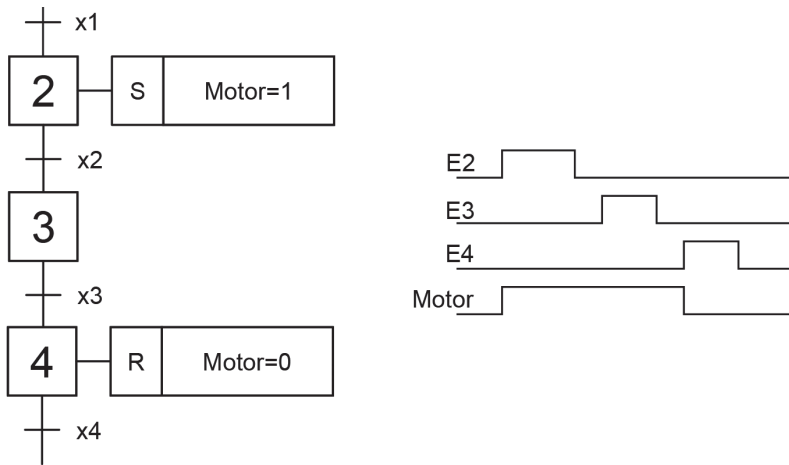
**Figura 4.5:** Acción limitada

- (P) Acción impulsional, representa una acción que se ejecuta en forma de impulso.



**Figura 4.6:** Acción impulsional

- (S) Acción memorizada, cuando la etapa se activa también lo hace la acción. Para su desactivación requiere de otra acción memorizada de Reset en una etapa posterior; permanecerá en este valor aun si la etapa se desactiva.



**Figura 4.7:** Acción memorizada

- Las acciones pueden condicionar su activación a una variable, lo que se denomina acciones asociadas condicionadas; se representan mediante una línea perpendicular al rectángulo de la etapa.

### 4.3.3. Transición y receptividad

Las transiciones son líneas horizontales representadas entre 2 etapas. Cada transición se asocia con una receptividad, que son condiciones que se deben validar para que se produzca el salto a la siguiente etapa. Pueden ser señales funciones lógicas o señales de contadores temporizadores.

### 4.3.4. Líneas de enlace

Son líneas que unen etapas con transiciones y transiciones con etapas. Por defecto, su sentido siempre será de arriba hacia abajo, a menos que se especifique lo contrario mediante flechas. Se pueden representar saltos en cualquier sentido dependiendo de la activación de la transición.

### 4.3.5. Bifurcaciones OR

En este caso, la selección de las ramas que continuará su secuencia de evolución será a través de la receptividad. Estas deben ser excluyentes

entre sí, es decir, una sola rama continuará con la ejecución del programa (ver Fig. 4.8). No es necesario que las ramas contengan el mismo número de etapas.

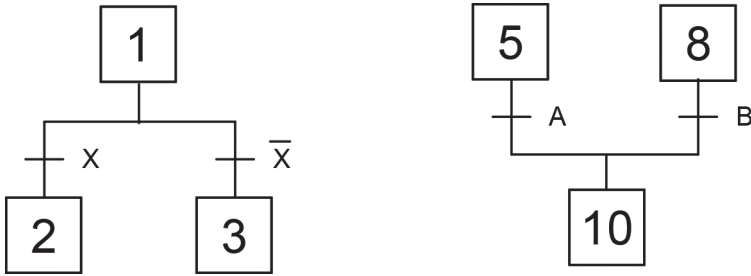


Figura 4.8: Ejemplo de bifurcación OR. Selección de secuencia

### 4.3.6. Bifurcaciones AND

Se utilizan para representar el caso en el que una transición permite la activación de más de una rama que continuará con la secuencia de ejecución de forma paralela. También sirven para sincronizar la finalización de la evolución de las ramas. Como se muestra en la Fig. 4.9, al final de cada rama, las etapas de espera deben estar todas activas para continuar la evolución del programa.

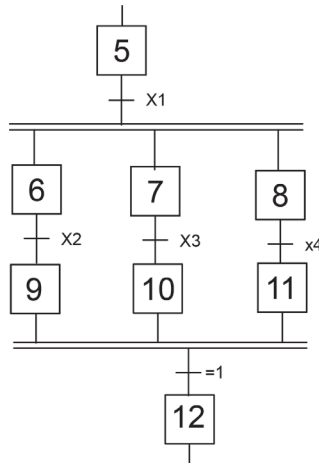
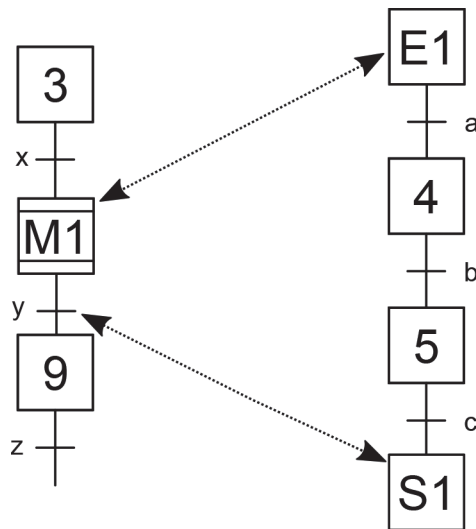


Figura 4.9: Ejemplo de bifurcación AND. Secuencias paralelas

## 4.4 Macroetapas

Las macroetapas representan subrutinas; se usan cuando un bloque de etapas se repite varias veces en un programa. Como se muestra en la Fig. 4.10, su simbología es un cuadrado con doble línea horizontal y sus reglas son las siguientes:

- Siempre tendrá una sola etapa de entrada y una sola de salida.
- Cuando se activa la macroetapa, se activará la etapa de entrada (E).
- La salida de la macroetapa (S) debe cumplir con la validación de la transición posterior a la macroetapa.

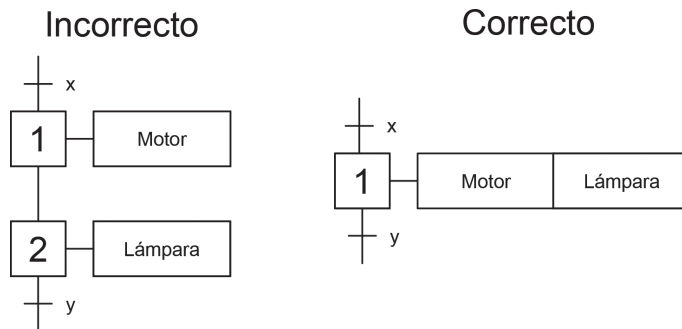


**Figura 4.10:** Macroetapas en diagramas GRAFCET



## 4.5. Reglas de evolución

- Al inicio de la ejecución del programa, todas las etapas iniciales deben estar activas.
- Como se muestra en la Fig. 4.11, no pueden existir 2 etapas seguidas sin transiciones de por medio, así como tampoco transiciones seguidas (ver Fig. 4.12).
- Una etapa puede tener asociada más de una acción.
- El proceso se descompone en una sucesión de etapas que se activan una a continuación de otra dependiendo de la validación de las transiciones.
- La validación de una transición implica la activación de la etapa siguiente y la desactivación de todas las anteriores simultáneamente.
- Si existe un lazo cerrado en una etapa, al evaluarse la transición, la etapa debe activarse y desactivarse al mismo tiempo. Se debe cumplir con el criterio de prioridad que determina la etapa que debe permanecer activa (ver Fig. 4.13).



**Figura 4.11:** Ejemplo de acciones seguidas

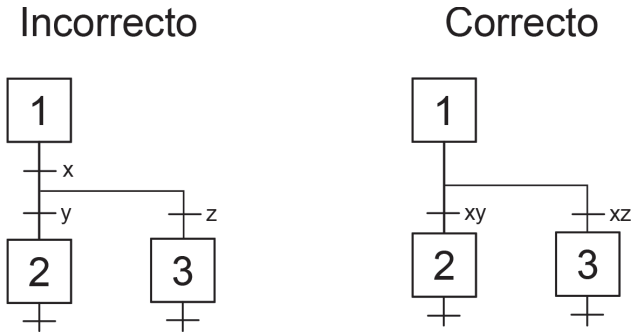


Figura 4.12: Ejemplo de transiciones seguidas

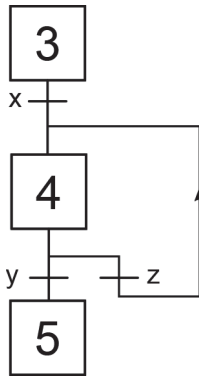


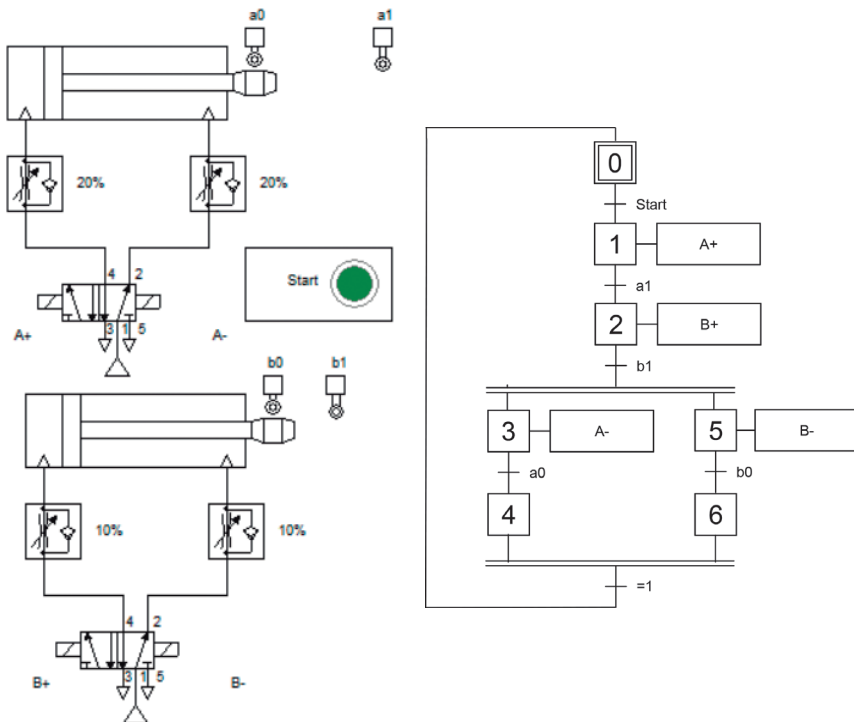
Figura 4.13: Ejemplo de prioridad de activación

*Ejemplo 1*

Los cilindros de doble efecto mostrados en la Fig. 4.14 inician su secuencia de funcionamiento presionando el botón *Start* que hará que el pistón A salga hasta llegar al fin de carrera a1. En este punto, el pistón B sale hasta llegar al fin de carrera b1. Cuando este último termina su recorrido, ambos pistones regresan a sus posiciones iniciales accionando los fines de carrera a0 y b0, para luego reiniciar nuevamente el ciclo. La Tabla 4.1 muestra variables del sistema.

**Tabla 4.1:** Tabla de señales de control

Señal	Descripción
Start	Botón de arranque
A+	Solenoides de electroválvula
A-	Solenoides de electroválvula
B+	Solenoides de electroválvula
B-	Solenoides de electroválvula
a0	Fin de carrera del pistón A posición inicial
a1	Fin de carrera del pistón A posición final
b0	Fin de carrera del pistón B posición inicial
b1	Fin de carrera del pistón B posición final



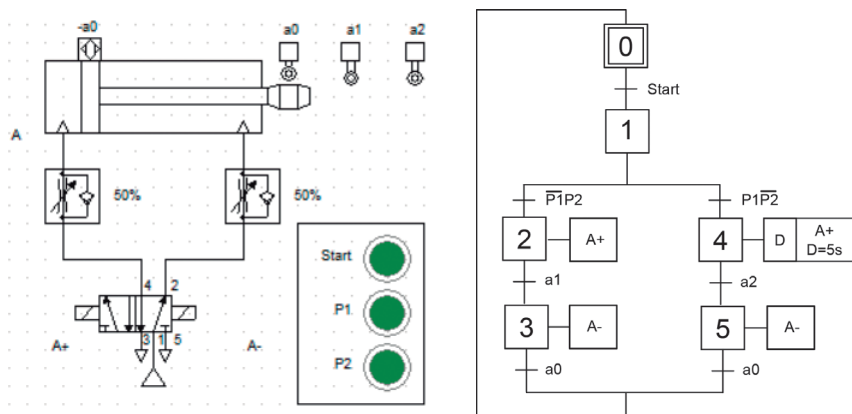
**Figura 4.14:** Solución - Ejemplo 1

## Ejemplo 2

Dado el pistón de la Fig. 4.15, la secuencia de funcionamiento inicia presionando el botón *Start*. Al pulsar el botón P, el pistón se mueve hasta llegar al fin de carrera a1, para luego contraerse a su posición inicial a0. Si se pulsa el botón Q, el pistón espera 5s antes de iniciar su recorrido hasta la posición a2 para luego regresar a su posición inicial y reiniciar su ciclo. La Tabla 4.2 muestra variables del sistema.

**Tabla 4.2:** Tabla de señales de control

Señal	Descripción
Start	Botón de arranque
P1	Botón de inicio de secuencia 1
P2	Botón de inicio de secuencia 2
A+	Solenoides de electroválvula
A-	Solenoides de electroválvula
a0	Fin de carrera del pistón A posición inicial
a1	Fin de carrera del pistón A posición media
a2	Fin de carrera del pistón B posición final



**Figura 4.15**  
Solución - Ejemplo 2

### Ejemplo 3

Un taladro de columna, como el que se muestra en el esquema de la Fig. 4.16, posee 2 motores: uno para hacer girar la broca y otro para subir y bajar el taladro. El sistema posee 2 interruptores de fines de carrera superior e inferior que detectan las posiciones iniciales y finales del taladro. El taladro únicamente se accionará si está posicionado en la parte superior (posición inicial) y se presiona el botón *Start*. Una vez activado el motor de la broca, girará durante 4 segundos para alcanzar la velocidad requerida para luego bajar hasta alcanzar el fin de carrera FC2 y, posteriormente, levantar el taladro hasta la posición inicial. Terminado el ciclo, se reinicia la secuencia.

Si el taladro está activado pero no alcanza los interruptores de fines de carrera en un tiempo de 15s, ambos motores se pararán y se activará una alarma por 5 segundos. Para reiniciar la secuencia de funcionamiento, el operador debe volver a presionar el botón *Start*. A continuación, en la Tabla 4.3 se muestran las variables del sistema y en la Fig. 4.17 el diagrama en GRAFCET de la solución del ejercicio planteado.

**Tabla 4.3:** Tabla de señales de control

Señal	Descripción
Start	Botón de inicio
FC1	Fin de carrera superior
FC2	Fin de carrera inferior
M1	Motor de la broca
M2+	Motor de desplazamiento vertical (baja el vástago)
M2-	Motor de desplazamiento vertical (regresa el vástago)
Alarm	Alarma

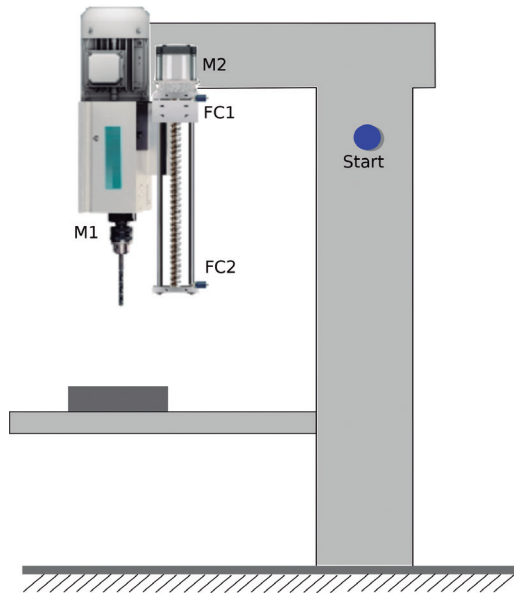


Figura 4.16: Esquema del proceso

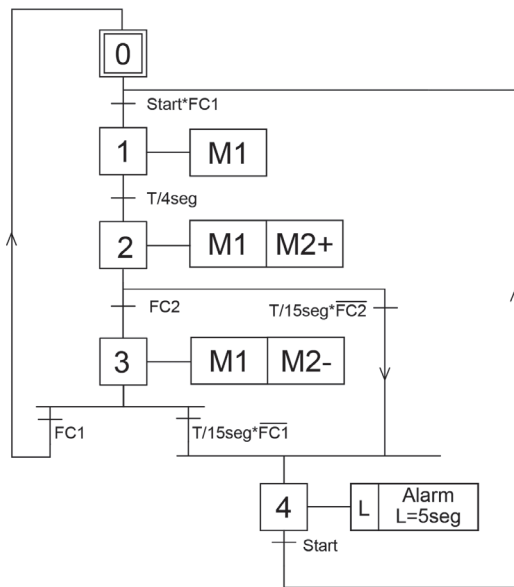


Figura 4.17 Diagrama GRAFCET

## 4.6. Conversión de GRAFCET a Ladder

Cuando se ha logrado la descripción del proceso a automatizar en GRAFCET, se puede realizar la descripción del automatismo respecto a la activación y evolución de las etapas definidas utilizando lenguaje Ladder (KOP) o mediante programación basada en bloques lógicas (FUP) [26].

Para realizar dicha conversión, se asigna una marca a cada etapa del GRAFCET, siendo las señales de transición las que permiten la evolución del sistema, activando secuencialmente una etapa a la vez, para lo cual se utilizan bobinas Set y Reset (cada par de bobinas representa un biestable RS).

Siguiendo las reglas de GRAFCET, la etapa inicial E0 debe estar siempre activa al inicio del proceso. En la Fig. 4.18a, se observa que esta condición se cumple al validar que el resto de etapas estén desactivadas. Así también, para activar una etapa, por ejemplo E2, se implementa una función AND entre la etapa anterior (E1) y la función de transición asociada (f2), a su vez haciendo un Reset de la etapa actual al desactivarla (E1), tal como se muestra en la Fig. 4.18b. Dado que se utilizan bobinas Set para las etapas, estas permanecerán activadas hasta que se active la bobina Reset correspondiente.

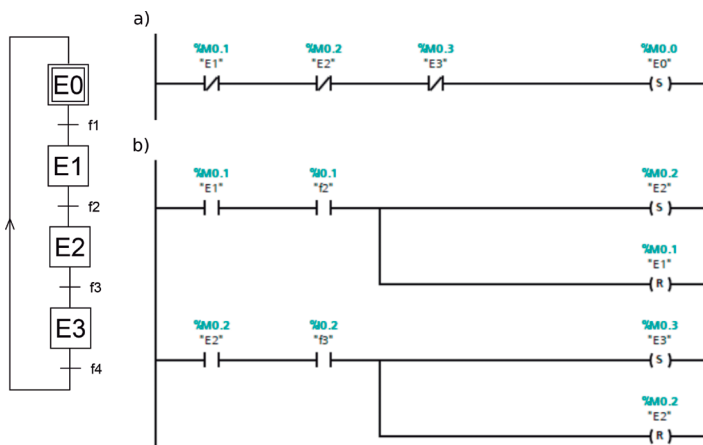


Figura 4.18: Conversión de etapas

### 4.6.1 Bifurcaciones condicionales OR

El caso de bifurcaciones mutuamente excluyentes como la mostrada en la Fig. 4.19, donde la evolución depende del estado de una función ( $f4$ ), se traduce a Ladder implementando una función AND entre la etapa actual ( $E3$ ) y la función de transición para cada caso. Dado que las bifurcaciones son mutuamente excluyentes, el proceso evolucionará por un solo lazo a la vez, es decir, se activará la etapa  $E4$ , si la función  $f4$  está activada; caso contrario, el proceso evolucionará por la etapa  $E5$ . La primera condición que se cumpla desactivará la etapa  $E3$ .

De forma similar, para salir de este tipo de bifurcaciones (ver Fig.4.20), se condiciona la activación de cualquiera de las etapas anteriores ( $E6$  o  $E9$ ) junto a su señal de transición asociada. La primera condición que se cumpla desactivará todas las etapas anteriores ( $E6$  y  $E9$ ).

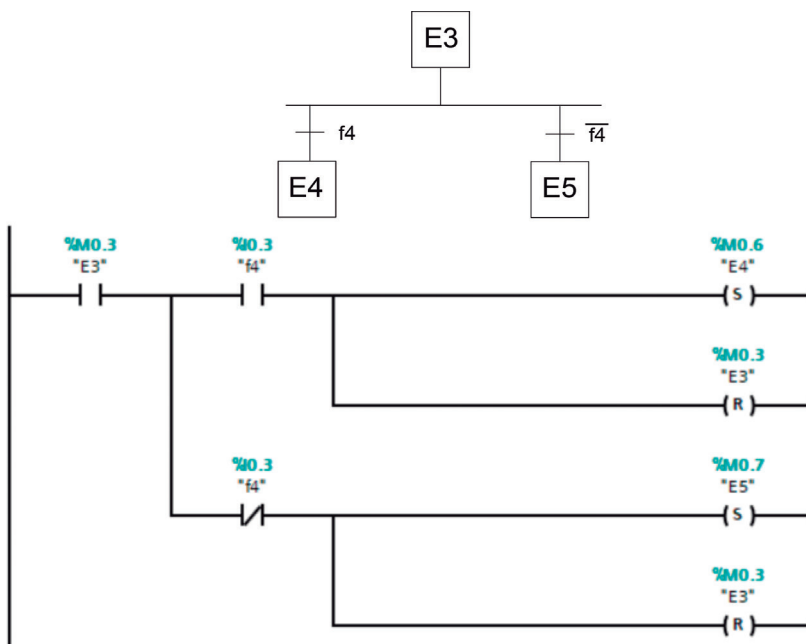


Figura 4.19: Bifurcación OR



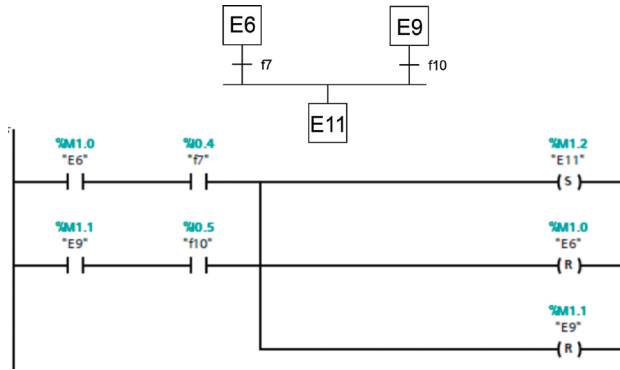


Figura 4.20: Cierre del lazo de bifurcación OR

### 4.6.2. Bifurcaciones AND

En el caso de procesos que requieren la ejecución de secuencias paralelas o simultáneas, como la mostrada en la Fig. 4.21, la conversión a Ladder se realiza evaluando la activación de la etapa actual (E5) y la función de transición asociada (f6). Cuando la operación AND es verdadera, la evolución se dará por las bifurcaciones de las etapas E6 y E7 de forma paralela.

Para salir de una bifurcación AND (ver Fig. 4.22), se debe esperar hasta que las etapas E8 y E6 estén activadas, así como la función de transición f8.

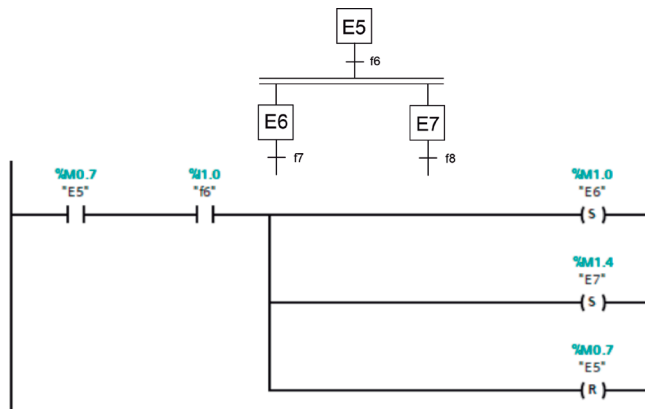


Figura 4.21: Conversión de bifurcación AND

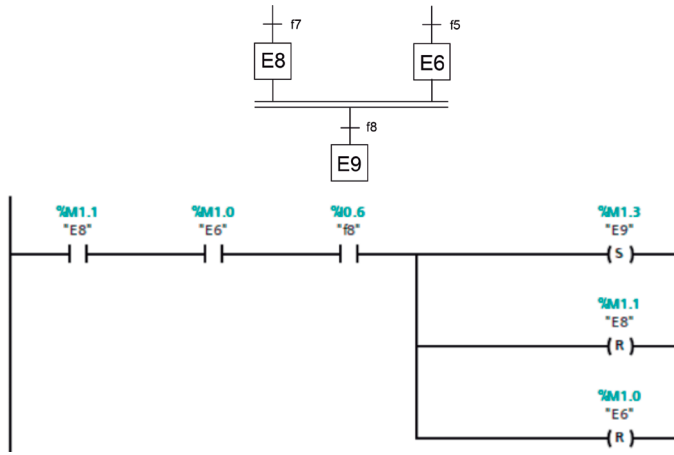


Figura 4.22: Conversión de cierre de lazo de bifurcación AND

### 4.6.3. Saltos condicionales

Los saltos condicionales ascendentes entre etapas, como el mostrado en la Fig. 4.23, se implementan en Ladder. Estos condicionan la etapa actual y la activación\desactivación de las funciones de transición asociadas. En el ejemplo, se producirá el salto a la etapa E1 cuando E5 y p estén activadas y f6, desactivada.

De forma similar, para saltos condicionales descendentes como el mostrado en la Fig. 4.24, se activará la etapa E6 cuando se evalúe como verdadera la condición: etapa E1 y función b activadas y función f2 desactivada.

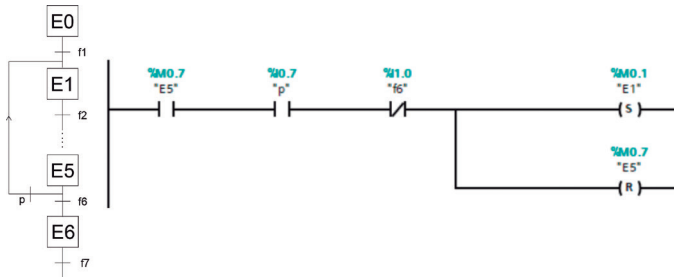


Figura 4.23: Saltos ascendentes

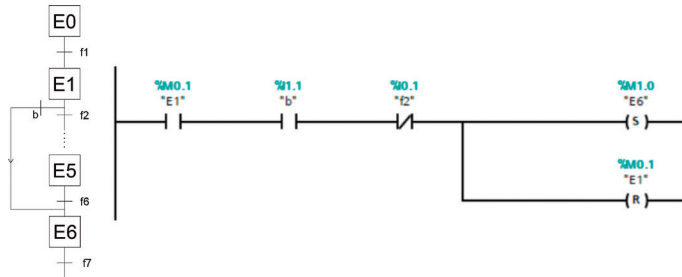


Figura 4.24: Saltos descendentes

### 4.6.4. Acciones asociadas

Durante la evolución del proceso, cada etapa asociada a una acción deberá activar el o los actuadores definidos en dicha etapa. Estos permanecerán activos mientras la etapa esté activa y en espera de la señal de transición para poder continuar con la secuencia de evolución. A continuación, se detallan los casos más comunes de implementación:

- La acción asociada está activa mientras la etapa está activa, como se muestra en la Fig. 4.25. La marca de la etapa activará el actuador.

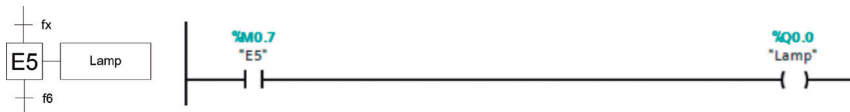


Figura 4.25: Acciones asociadas

- Si la acción asociada está condicionada —como el ejemplo de la Fig. 4.26—, la bobina Motor se activará en función de activación de la etapa E7 y la señal condicionada a (operación AND entre dichos contactores).

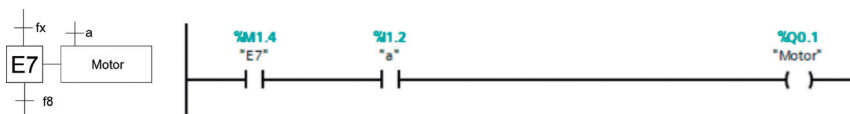


Figura 4.26: Acciones asociadas condicionadas

- En el caso que un actuador deba activarse en determinada etapa y desactivarse en otra —como el caso en la Fig. 4.27—, se puede usar una bobina Set Reset para implementar esta opción.

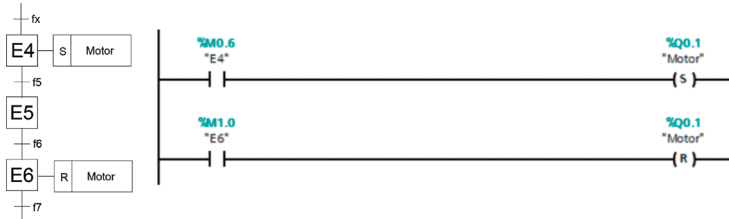


Figura 4.27: Acciones biestables

### Ejercicio Ladder

En la Fig. 4.28, se muestra el equivalente Ladder para la codificación de las etapas y acciones del GRAFCET del ejemplo 2 que realiza la secuencia de control de un cilindro.

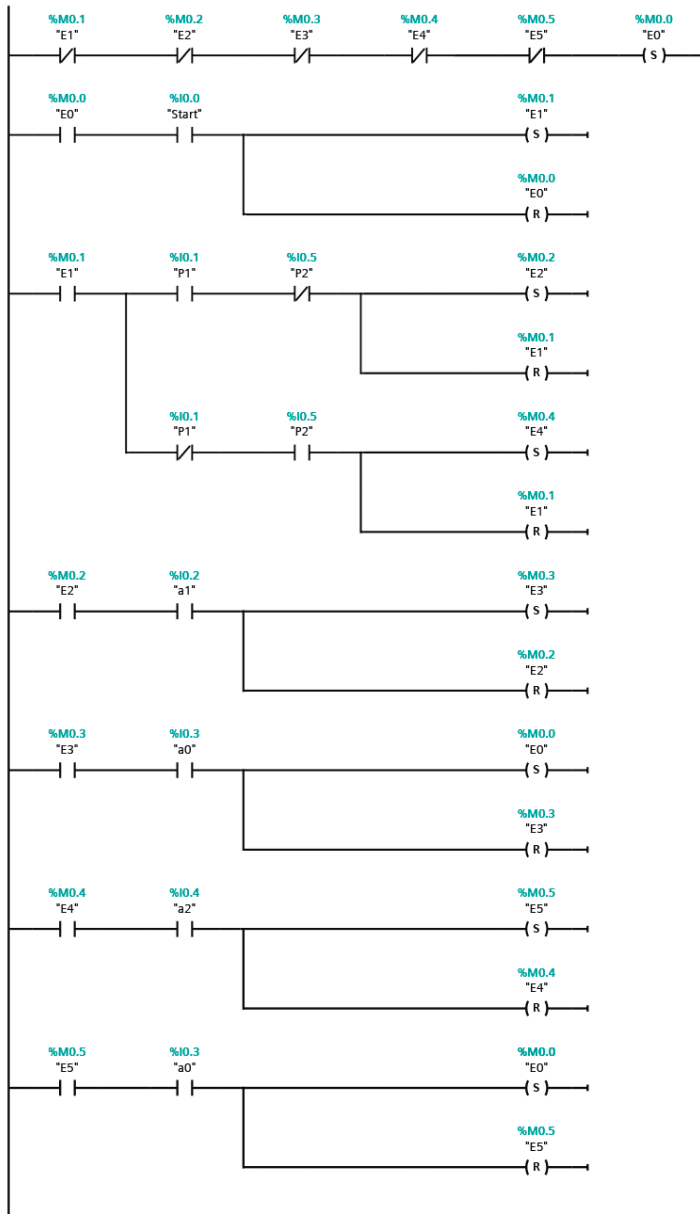


Figura 4.28: Equivalente Ladder de ejemplo 2

## 4.7. Ejercicios propuestos

1. Realice la conversión a Ladder del GRAFCET mostrado en la Fig. 4.29.

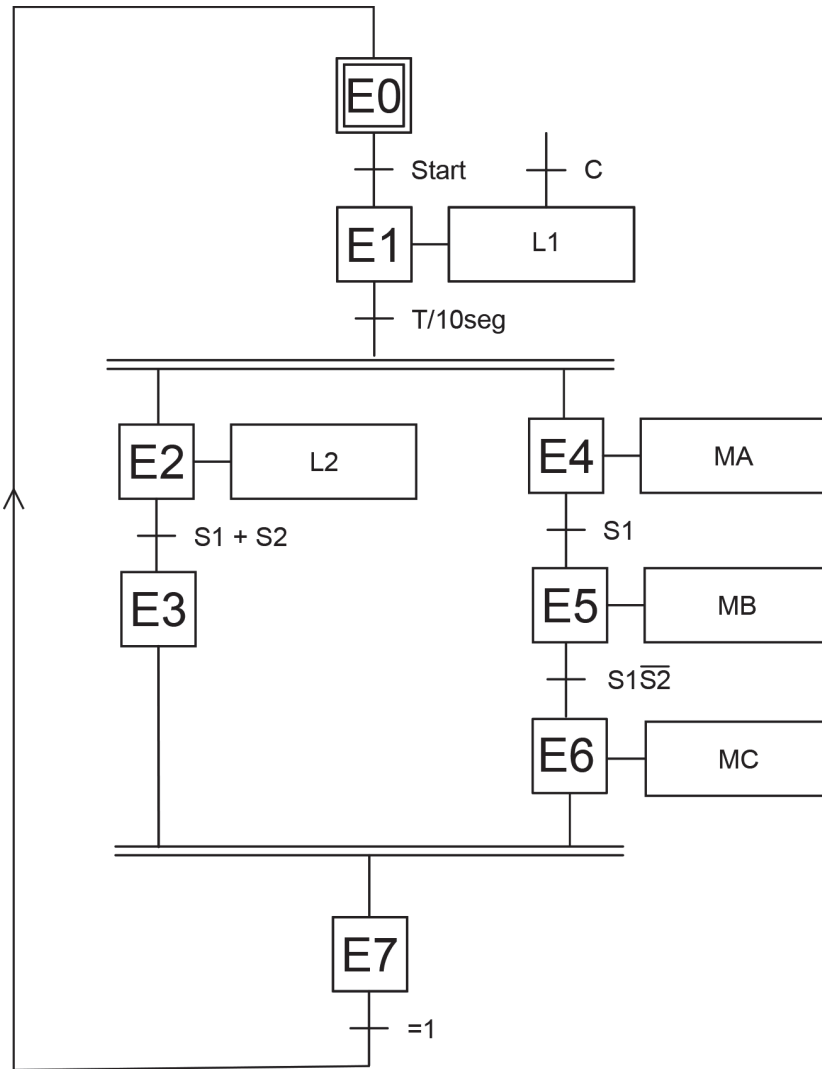


Figura 4.29: GRAFCET

2. Modifique la secuencia del ejercicio 3 para añadir un botón *Stop*. Este, al ser presionado en cualquier momento, detendrá la broca del taladro y lo moverá a su posición inicial.
3. Realice la tabla de señales de entrada y salida del proceso y el diagrama GRAFCET para controlar un sistema de transporte y empaque de piezas mediante 2 bandas transportadoras, tal como se muestra en el esquema de la Fig. 4.30. En su estado inicial, los pistones están contraídos. Al presionar el botón *Start*, se activan las dos bandas y se envía una señal al emisor para que inyecte las cajas al sistema. A través de un sensor, se detecta cuando la pieza llega al pistón de elevación, parando la banda 1 y al emisor, expandiendo el émbolo del pistón para levantar la caja. Cuando llega a su posición final, se extiende el pistón de desplazamiento horizontal para colocar la caja en la banda 2. Ambos pistones regresan a su posición inicial al mismo tiempo y se activa nuevamente la banda 1 y el emisor. Cuando se completan 8 cajas impulsadas a la banda 2, se activa la luz indicadora “Empacar piezas” y se para el sistema. Para reiniciar el proceso, se debe volver a presionar el botón *Start*. Cuando se presiona el botón *Stop*, se debe parar completamente todo el proceso.

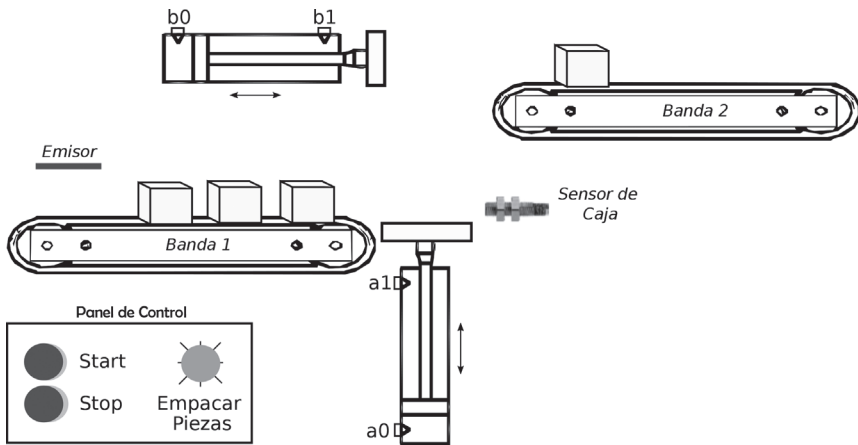


Figura 4.30: Esquema del proceso de transporte y empaquetado

4. Diseñe el control de un circuito neumático mediante 2 pistones de doble efecto que sigan la secuencia A+B+B-A- al presionar el botón *Start* de forma cíclica 4 veces. La secuencia se detendrá al presionar el botón *Stop*. Se debe realizar:

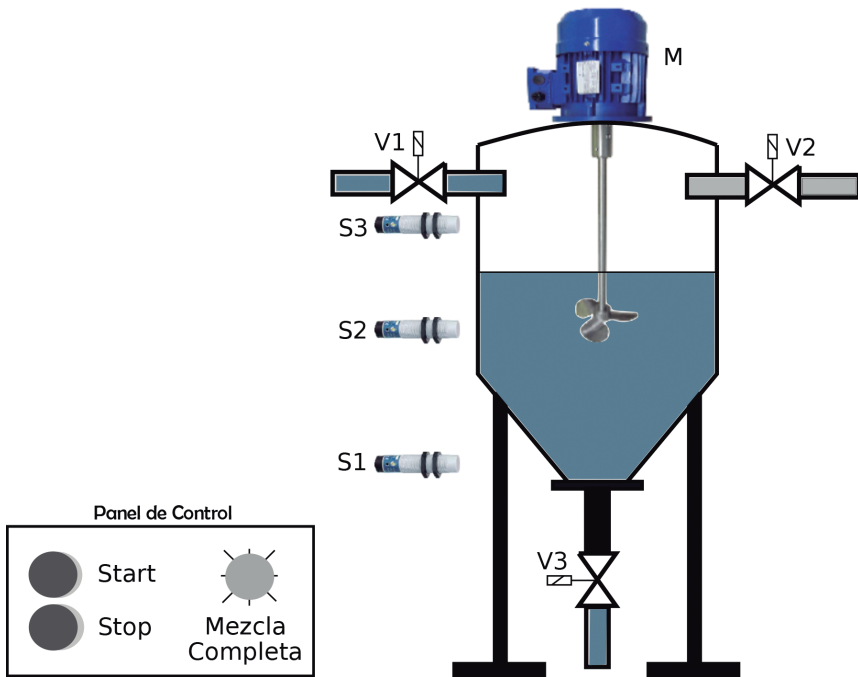
- Tabla de señales de entrada y salida del proceso.
- Esquema del circuito neumático.
- Diagrama de conexión al PLC y el programa de control en GRAFCET.

5. En el proceso de mezclado industrial ilustrado en la Fig. 4.31, al presionar el botón *START* se inicia la operación de llenado activando la válvula V1 y permitiendo el vertido del líquido al reservorio mientras el nivel esté por debajo de S2. Para completar el llenado, se abre la válvula V2 para permitir el ingreso del líquido de mezcla hasta alcanzar S3. Cuando la tolva esté llena, se activará el motor M por 15 segundos antes de abrir la válvula de vaciado V3. Se necesita llenar un tanque de 90 litros y cada reservorio tiene una capacidad de 30 litros, por lo que se requiere repetir el proceso de llenado y mezcla 3 veces. Cuando este ciclo se completa, se enciende la luz indicadora de “Mezcla completa” por 5 segundos y el proceso se reinicia al presionar el botón *Start*. Si se activa el botón *Stop*, se deben cerrar todas las válvulas automáticamente.

Realice:

- Tabla de señales de entrada y salida del proceso.
- Diagrama de conexión al PLC y el programa de control en GRAFCET.





**Figura 4.31:** Proceso de mezclado y envasado

- A partir del GRAFCET, realice la conversión a Ladder.



**Capítulo 5**  
**Redes Industriales**



En la actualidad, la implementación y utilización de una red a nivel industrial permite establecer una serie de ventajas competitivas a una empresa o industria, desde la reducción del cableado, mejora de tiempos de respuesta, flexibilidad en la producción, manejo, almacenamiento y análisis de datos, detección de errores, menores costos de mantenimiento, información en tiempo real y vinculación con herramientas y recursos de gestión, entre otros. Ante un mundo cada vez más competitivo, esto permite contar con información del estado y funcionamiento de un proceso industrial y establecer políticas y criterios de trabajo que incrementen la eficiencia y producción de una industria, lo que orienta a la empresa dentro de los desafíos de la Industria 4.0.

En este capítulo se presenta una breve descripción de las principales opciones de comunicación entre dispositivos de automatización, como las redes industriales; para esto, debe comprenderse inicialmente que, dentro de un proceso industrial, existen niveles de equipos y automatización, lo cual se ha integrado y descrito en lo que se denomina pirámide de automatización. Posteriormente, se describen las opciones de comunicación y protocolos más utilizados como medios de comunicación y bases para los protocolos de comunicación industrial. Finalmente, se estudian y aplican tres protocolos de comunicación industrial aplicables a los niveles inferiores de la pirámide de automatización, como Modbus, Profibus DP y Profinet; de estos se presenta una descripción teórica seguida de un ejemplo práctico de uso.

## 5.1. Pirámide de Automatización

La pirámide de automatización es una representación gráfica de los niveles de integración y automatización que puede tener o alcanzar

una industria. Integra tanto el proceso productivo, en el cual se utilizan e implementan tecnologías de operación y equipos industriales, como la parte administrativa o de gestión, en la cual se encuentran las tecnologías de la información en combinación con equipos de tipo informático comercial, infraestructuras de la comunicación, recursos de software para la administración y gestión. El propósito de la pirámide es ilustrar y representar una industria en general, para evaluar, medir el nivel de automatización de sus procesos industriales o administrativos e integrar ambos, que se constituye actualmente en el nuevo desafío para toda industria en esta era de la comunicación e información.

Como se muestra en la Fig. 5.1, la pirámide está constituida por 5 niveles que representan los niveles de automatización de un proceso industrial, abarcando toda la cadena productiva y la parte administrativa. Se representa por una pirámide debido a que, para llegar a una automatización adecuada de los niveles superiores, requiere previamente la automatización de los niveles inferiores. Por ello, la automatización de un proceso industrial debe realizarse de abajo hacia arriba, si se desea obtener un sistema acorde a las necesidades y requerimientos actuales.

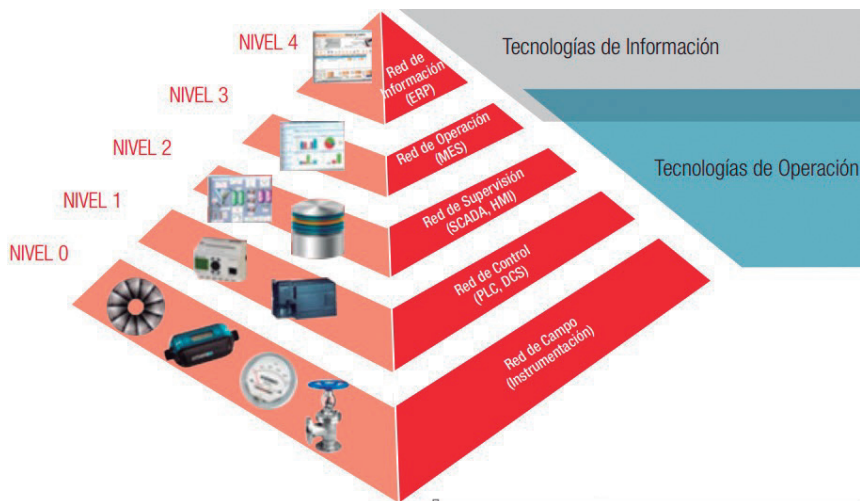


Figura 5.1: Pirámide de automatización [27]

### 5.1.1. Nivel de campo

También identificado como nivel de sensores y actuadores, es aquel en el cual los elementos que la componen están en contacto directo con la materia prima y dentro del proceso productivo, es decir, está formado por todos los sensores que miden y evalúan el estado de una variable del proceso y por los actuadores, encargados de establecer y modificar el estado de una variable del proceso. Por lo tanto, son los que están dentro del proceso de transformación de la materia prima en un sub-producto, producto o servicio.

Se caracteriza por presentar una gran cantidad de elementos, formada por toda la instrumentación necesaria para controlar y monitorear un proceso, es decir, todos los sensores y actuadores que supervisan un proceso. Por lo tanto, dentro de este nivel existe el mayor número de componentes de una red industrial, pero, a su vez, estos elementos son los que tienen muy poca información que proporcionar, debido a que, al tratarse de sensores digitales on-off, la señal de interés será únicamente un 1 o 0 lógico que representa su activación o desactivación y, para sensores analógicos, su valor dependerá de la resolución del sensor sin superar los 2 bytes. Por ello, la cantidad de datos que tienen estos elementos para transmitir es muy baja (están en el orden de los bytes), por lo que sus velocidades de transmisión no necesitarán tampoco ser muy altas (están en el orden de los kbps). Adicionalmente, las distancias a cubrir por estas redes pueden ser bastante extensas, por lo que deberán ser capaces de abarcar centenas de metros, llegando incluso a los miles de metros. También debe considerarse que, al tratarse de procesos productivos, los tiempos de respuesta toman una importante relevancia, por lo que la latencia y manejo del tiempo real (real time) son de gran importancia; por ello, deben manejar sistemas de transmisión y recepción determinísticos.

Finalmente, este nivel, al estar en contacto con el proceso, cruza por todo el proceso productivo, por lo que en su camino puede encontrarse con fuentes de ruido e incluso puede entrar contacto con residuos químicos del proceso, por lo que los medios de transmisión

y cables utilizados en este nivel deben ser lo suficientemente robustos como para soportar estos ambientes hostiles, tanto desde el punto de vista de transmisión del dato, ruido e interferencias en la comunicación, como temperatura, humedad, contaminación, contacto con residuos químicos tóxicos y corrosión, entre otros que pueden afectar su vida útil.

### **5.1.2. Nivel de control**

Es el nivel en el cual se encuentran los elementos que contienen la lógica de control de los procesos; en otras palabras, los autómatas programables, controladores lógicos programables (PLC's), sistemas de control distribuido (DCS), variadores de velocidad, controladores de variables específicas, entre otros. Los PLC's, en la actualidad, predominan dentro del nivel de control, en el proceso a partir de la lógica o algoritmo con el que han sido programados; también pueden ser conectados a interfaces humano máquina (HMI) o en red con otros PLC's cuando los procesos así lo requieren, ya sea por la complejidad del algoritmo de control, número de variables o elementos de entrada y salida a manejar o por la distancia a cubrir en el control de un proceso industrial.

La cantidad de elementos de control es mucho menor a la de sensores actuadores; sin embargo, cada vez se implementan procesos más complejos o con un mayor nivel de automatización, que requieren un mayor número de este tipo de elementos y algoritmos cada vez más complejos.

Los requerimientos de una red industrial en el nivel de control son diferentes a los del nivel de campo debido a que siempre habrá que integrar un menor número de equipos, pero los requerimientos de información serán mayores: por lo general, deberán comunicar a otros equipos de control el estado de los sensores y actuadores conectados y, de igual manera, requerirán información del resto de equipos, por lo que el flujo y tamaño de datos será mayor (en el orden de bytes y kbytes).

En cuanto a los tiempos de respuesta y las distancias a cubrir, los requerimientos son muy similares a los del nivel de campo. Se deben considerar los mismos aspectos de latencia, tiempo real y robustez en



el medio de transmisión de los datos; a este nivel también se necesitan sistemas de comunicación determinísticos.

### 5.1.3. Nivel de supervisión

Como se muestra en la Fig. 5.2, está constituido por el Sistema de Supervisión Control y Adquisición de Datos (Scada). Este está conformado por toda la infraestructura de hardware y software requerida para la implementación de un sistema de supervisión, monitoreo y control remoto del proceso. Su principal elemento es el cuarto de control central, desde donde se puede visualizar el funcionamiento parcial o total de un proceso industrial, y se constituye en la herramienta principal para identificar y modificar el estado de una planta industrial.



**Figura 5.2:** Nivel de supervisión [28]

Dentro de los elementos de hardware se pueden encontrar PLC's, HMI's, servidores informáticos, monitores de gran tamaño, routers, elementos y accesorios de red, sistemas de comunicación, entre otros. Dentro de los elementos de software están sistemas OPC Server,

SCADAs, software de análisis de datos, bases de datos, gráficos históricos y en tiempo real, entre otros. Por lo tanto, en este nivel se evidencia la transición y unión de la producción con la administración.

Este nivel se constituye en un interfaz entre la parte industrial — encargada de la producción del bien, servicio o producto— y la gestión —responsable del manejo de la parte administrativa de una empresa—; integra ambas partes, permitiendo el intercambio de información entre ambas, transformando, por ejemplo, los pedidos de clientes en órdenes de trabajo, de despacho de materia prima y producto terminado.

### 5.1.4. Nivel de gestión de la producción

Está formado por toda la parte administrativa y de gestión necesaria para la planificación, gestión y organización de un proceso industrial, es decir, es el encargado de la logística, adquisición de materia prima, almacenaje, entrega de productos terminados, cadena logística, talento humano, control de calidad, compras y contabilidad, entre otras.

Es la parte de la gestión de una industria que se realiza en planta: verifica, monitorea el estado, establece y planifica las tareas a ser desarrolladas por la mano de obra que opera en una instalación industrial (ver Fig. 5.3).

Los requerimientos de este nivel van a nivel de equipos informáticos, servidores, recursos de software y, en base a la información de los niveles inferiores, establece políticas y tareas para los operarios.

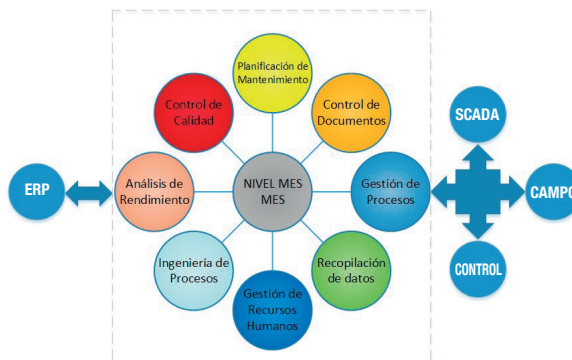


Figura 5.3: Nivel de gestión de la producción

### 5.1.5. Nivel de planificación de recursos empresariales

El nivel de planificación de recursos empresariales (Enterprise Resource Planning, ERP) dentro del entorno productivo desempeña el papel de sistema para planificar los recursos. En otras palabras, gestiona la información y satisface las necesidades de gestión empresarial de una forma estructurada e integral, estableciendo una relación muy fuerte entre el proceso productivo y la parte administrativa. Además, permite a las empresas evaluar, controlar y gestionar fácilmente un negocio en todos sus ámbitos. Se caracterizan por ser modulares y estandarizados e incluyen interfaces de conexión con otros sistemas de información [29]. Esto permite la integración del entorno productivo al administrativo; su implementación alcanza la mejora de procesos productivos, reducción de inventario, optimización de espacios de almacenamiento, establecimiento de cadenas logísticas y gestión de ventas, planificación y ejecución de pedidos.

Este nivel está a la cabeza de la organización y, por lo tanto, para la correcta toma de decisiones, requiere información adecuada y precisa del estado de todos los niveles inferiores, ya que es la encargada de vincular el estado interior del proceso productivo con el mundo exterior, tal como muestra la Fig. 5.4.

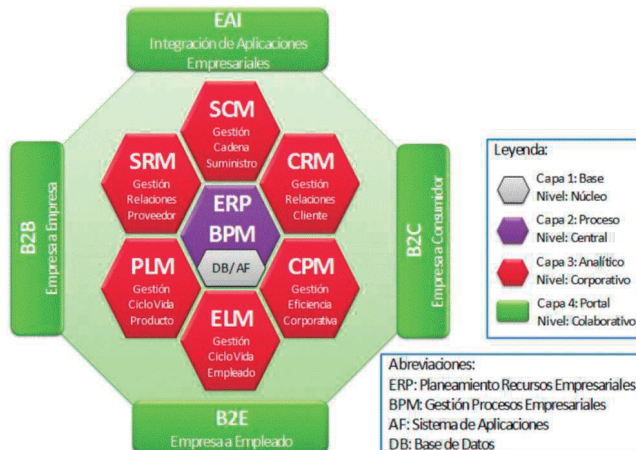


Figura 5.4: Nivel de gestión de recursos empresariales [30]

## 5.2. Sistemas de comunicación

Los requerimientos actuales de los procesos industriales crean la necesidad de infraestructuras de la comunicación que permitan la transmisión y recepción de datos del estado de las variables, sensores y actuadores de un proceso industrial, acorde a las necesidades del proceso y que se adapten a las características y limitantes del entorno tanto industrial como comercial.

Con la información, las empresas pretenden establecer un posicionamiento en el mercado a partir de la homogenización de la calidad de sus productos, optimización de los procesos productivos, reducción de niveles de bodega, entregas anticipadas a clientes, entre otras. Requieren de estas infraestructuras que facilitan la comunicación, evitando intermediarios y permitiendo conectar directamente a los partícipes del proceso productivo y administrativo de una industria, generando un nivel mayor de automatización, como se muestra en la Fig. 5.5

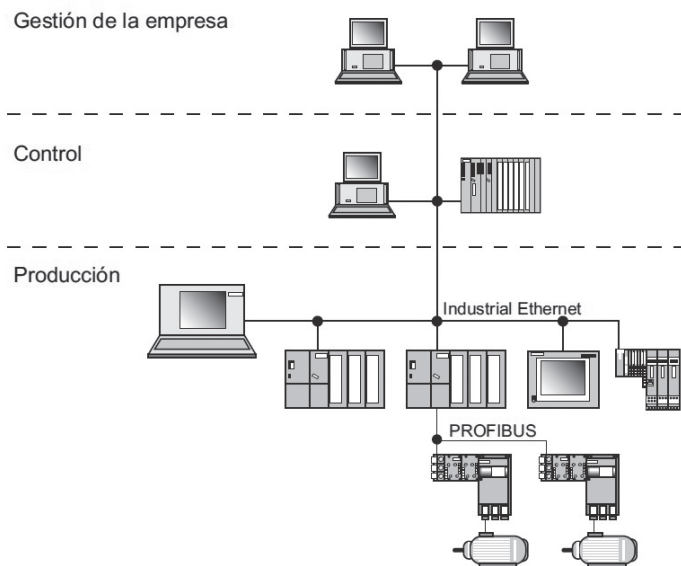


Figura 5.5: Sistema de comunicación [31]

### 5.2.1. Elementos de una comunicación

La comunicación tiene elementos particulares. En cualquier comunicación debe haber una fuente (en comunicaciones de datos, a menudo esto se denomina emisor o transmisor) y uno o más destinos (generalmente, llamado receptor o destino). El propósito de la comunicación es transmitir datos desde la fuente hasta los destinos. Los datos son transmitidos a través de un canal o medio de un tipo u otro (que varía según la tecnología utilizada). Cuando una persona habla, por ejemplo, usa ondas de audio para proyectar sonido a través del aire como canal; en comunicaciones de datos, los canales, por lo general, son conductores eléctricos (conocidos como conexiones alámbricas), haz de luz a través de guías o tubos (cableado de fibra óptica) y transmisión electromagnética (inalámbrica o de radio) [32].

Adicionalmente, como se observa en la Fig. 5.6, pueden intervenir factores externos que afectan la comunicación, como el ruido, que dentro de una red industrial es el más importante. Finalmente, es necesario, en algunos casos, que el receptor indique al emisor que el mensaje fue recibido a través de una realimentación.

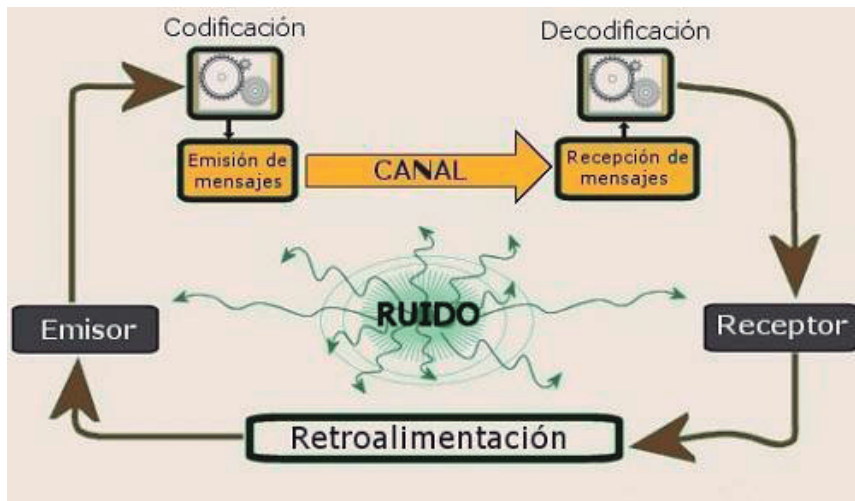
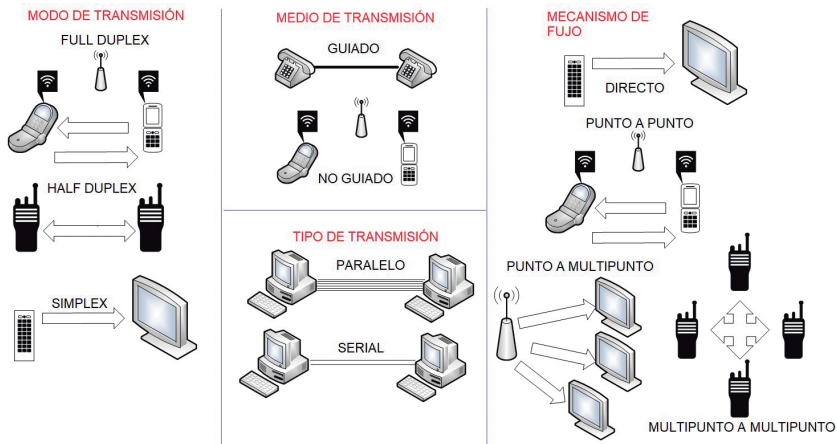


Figura 5.6: Elementos de la comunicación [33]

## 5.2.2. Clasificación de los sistemas de comunicación



**Figura 5.7:** Clasificación de los sistemas de comunicación

Los sistemas de comunicación se clasifican según varios parámetros que permiten definir las características de la comunicación y la forma como se van a transmitir los datos hacia el o los receptores. Se visualiza algunos en la Fig. 5.7.

### *Por el tipo de transmisión*

**Paralela:** Los datos se transmiten a través de varias líneas de comunicación, transmitiendo simultáneamente varios datos. Por ejemplo, comunicación paralela (LPT).

**Serial:** Los datos se transmiten a través de un único medio, como un tren de pulsos; es actualmente la forma de transmisión más utilizada. Ejemplo: RS-232, USB, SATA, etc.

### *Por el modo de transmisión*

**Full dúplex:** Los equipos pueden transmitir y recibir datos simultáneamente. Ejemplo: telefonía.

**Half dúplex:** Los equipos pueden transmitir y recibir datos, pero no simultáneamente. Ejemplo: walkie talkie, radio handy.

**Simplex:** Los equipos pueden solo transmitir o recibir datos, es decir, está definido si el equipo es transmisor o receptor. Ejemplo: juguetes a control remoto.

*Por el mecanismo de flujo*

**Enlace directo:** Los equipos están conectados y comunicados directamente, sin ningún equipo o dispositivo intermediario. Ejemplo: entrevista personal.

**Enlace punto a punto:** Los equipos están conectados y comunicados a través de un equipo o dispositivo intermediario. Ejemplo: entrevista vía telefónica.

**Enlace punto a multipunto:** Un solo equipo transmite los datos y varios equipos los reciben. Ejemplo: transmisión de canales de televisión.

**Enlace multipunto a multipunto:** Todos los equipos pueden transmitir y recibir datos en una comunicación con más de dos equipos. Ejemplo: grupo de chat.

*Por el medio de transmisión*

**Guiado:** La transmisión de los datos se hace a través de un medio físico, el cual transmite los datos. Por lo general, existen medios eléctricos representados por los tipos de conductores eléctricos que pueden ser utilizados en las comunicaciones, como par simple, par trenzado, cable coaxial, buses de datos, entre otros. También hay medios ópticos representados por la fibra óptica, cada uno con aplicaciones específicas derivadas de las características eléctricas, mecánicas y de transmisión que poseen. Ejemplos: teléfono (par simple), redes de datos (par trenzado), TV cable (coaxial), Internet (fibra óptica).

**No guiado:** La transmisión se realiza sin utilizar ningún medio físico, sino que los datos se difunden a través del medio ambiente. Por lo general, se hace a través de ondas y señales electromagnéticas.

Existe una amplia gama y distribución de frecuencias de transmisión, establecidas y reguladas a nivel nacional e internacional por los entes reguladores correspondientes y dependientes del área de cobertura; también hay sistemas de transmisión no guiado de tipo ópticas, utilizadas principalmente a nivel de equipos domésticos en sistemas de control remoto infrarrojos y láser. Ejemplo: transmisiones de radio, TV, telefonía móvil, control remoto de TV, comunicaciones satelitales.

### *Por el tipo de sincronismo*

**Sincrónico:** Cuando dentro de las señales a ser enviadas y recibidas se incluye una de reloj que sincroniza la comunicación entre los equipos. Ejemplo: Protocolo I2C.

**Asincrónico:** Cuando dentro de las señales a ser enviadas y recibidas no se incluye una señal de reloj, en cuyo caso la comunicación se sincroniza a partir de bits banderas o de inicio y parada agregados en la transmisión de los datos. Ejemplo: Comunicación RS-232.

**Isócrona:** No existe una señal de sincronismo; los equipos se sincronizan a partir de otros mecanismos. Falla si hay un retraso o se desincronizan los equipos; se utiliza en transmisiones en las cuales los datos deben ser procesados y presentados en tiempo real. Por ejemplo, las transmisiones de televisión implican enviar 30 imágenes por segundo, la transmisión isócrona garantiza tales transmisiones de modo que las imágenes lleguen a la misma velocidad para que puedan ser percibidas correctamente. Una parte importante del ancho de banda se asigna a uno o dos dispositivos para la transferencia de datos en modo isócrono. Este método se utiliza para la transferencia de datos a alta velocidad [34].

### *Por el tipo de acceso al medio*

**Determinístico:** Cuando se garantiza el acceso al medio de transmisión de manera establecida y definida, en un orden y tiempo establecido para todos los equipos. Ejemplo: horario de clases o cronograma de exposición.



**Probabilístico:** Cuando se garantiza el acceso al medio de transmisión, pero no de manera definida ni en un orden ni tiempo establecido para todos los equipos. Es decir, todos tienen acceso al medio, pero no en un orden ni tiempo definido, por lo que, el acceso al medio para la transmisión se considera aleatoria y depende de la cantidad de equipos y datos que necesiten transmitirse, lo que puede generar el denominado tráfico de red. Ejemplo: Internet, el tiempo de acceso a un servicio o página depende del tráfico existente en la red y del número de usuarios que intentan acceder al servicio de manera simultánea.

*Por el método de acceso al medio*

**Sondeo o polling:** Sistema determinístico que se presenta cuando un dispositivo maestro realiza preguntas individuales secuenciales a los esclavos sobre su estado. Estos responden de forma individual al maestro, si se dispone de la información requerida. El inconveniente que presenta es que puede generar tráfico innecesario debido a que pueden existir esclavos que no tengan información para entregar al maestro. Debe ser utilizado en sistemas en los cuales los esclavos tengan poca información que entregar, como en las redes a nivel de campo.

**Selección:** Sistema determinístico que se presenta cuando un dispositivo maestro selecciona el esclavo del cual va a requerir la información; este responde si tiene o no la información requerida través de un mensaje. El inconveniente es que se puede monopolizar el medio, comunicándose exclusivamente con un determinado esclavo, sin atender la información de los demás. Puede ser utilizado cuando la cantidad de información a entregar no es alta y la prioridad es determinada por el maestro.

**Paso de testigo o token:** Sistema determinístico que se basa en el manejo o paso de un testigo, que es el que da acceso al medio y que pasa de equipo en equipo secuencialmente, dando acceso a todos, los cuales, al momento de recibir el testigo, pueden transmitir su información para pasar el testigo al siguiente equipo. Es completamente independiente de la configuración o ubicación física de los equipos,

más bien depende de la dirección lógica que se asigna a cada uno. Existen dos tipos de sistemas de paso de testigo, en bus (token bus) o en anillo (token ring).

En token bus existe un dispositivo encargado de transmitir el testigo en base a un orden o criterio establecido y se identifican dos tipos de tramas: trama testigo o trama de datos. Además, se define un tiempo límite para que el dispositivo pase el testigo, que puede ser diferente para cada dispositivo; está definido en la norma IEEE 802.4.

En token ring todos los equipos se consideran iguales y cualquiera puede controlar el paso del testigo. De igual manera, existen dos tipos: de tramas, de control y datos. No hay un tiempo límite para el uso del testigo; por el contrario, se espera a que el equipo transmita toda la información antes de pasar el testigo, está definido en la norma IEEE 804.5.

**Acceso múltiple por detección de portadora con detección de colisión (Carrier Sense Multiple Access with Collision Detection CSMA/CD):** Se basa en el acceso al medio probabilístico a través de la detección del medio que está disponible: se escucha al medio a través de la detección de niveles de voltaje y, en caso de no detectar señales, transmite los datos; sin embargo, pueden existir momentos en que dos o más equipos intenten transmitir al mismo tiempo, generando lo que se denomina colisión, en cuyo caso se detiene la transmisión por un tiempo aleatorio antes de volver a transmitir. Está definido en la norma IEEE 802.3.

**Por eventos:** Sistema probabilístico que se basa en dejar el medio libre y solo ocuparlo cuando un dispositivo tenga datos a transmitir. Es ideal en procesos en los cuales se establecen prioridades para los datos y se requiere de un canal disponible para transmitir los datos cuando se requiere, siempre y cuando la cantidad de datos es baja, y se va a dejar libre nuevamente el medio.

### *Por la topología física*

**Bus:** Cuando existe un medio o bus común al cual están conectados todos los equipos. Es muy común en sistemas de comunicación industrial, a nivel de campo y control, pues permite cubrir distancias

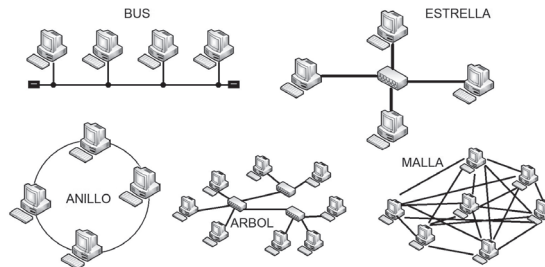
más largas. Requiere terminadores, que se colocan en ambos extremos del bus y permiten mantener la impedancia del cable, asegurando la transmisión de datos. Tiene la desventaja de que cuando el medio falla en cualquier punto, toda la comunicación falla.

**Árbol:** La red de comunicaciones tiene una distribución que se asemeja a un árbol, es decir tiene varios concentradores de los cuales se derivan a manera de ramas los equipos conectados. Es muy similar a la topología estrella, con la diferencia que no existe un equipo central, sino más bien varios concentradores.

**Anillo:** Cuando existe un único medio o bus común pero forma un camino cerrado, es decir, los extremos del bus están unidos formando un anillo. La principal diferencia respecto a la topología bus es que elimina la necesidad de los terminadores; tiene el mismo inconveniente de la topología bus, sin embargo, puede establecerse un sistema de doble anillo que reduzca el riesgo de fallo en la comunicación, la que se denomina anillo redundante.

**Estrella:** Cuando existe un dispositivo central al cual están conectados todos los equipos y es el responsable de la comunicación y conexión entre todos los que conforman la red. Tiene el inconveniente de que, en caso de una falla en el equipo central, toda la comunicación falla.

**Malla:** Cuando para la comunicación entre dos o más equipos se establecen diferentes caminos por donde pueden transmitirse los datos; funciona a partir de la conexión entre todos los equipos que forman parte de la red de comunicación y es ideal para asegurar la transmisión de los datos, pero su implementación es muy costosa. Los tipos de topologías se muestran en la Fig. 5.8.



**Figura 5.8:** Topologías físicas de comunicación

### *Por la relación funcional*

**Entre iguales o peer to peer:** En una comunicación de este tipo, todos los equipos son considerados iguales y tienen el mismo acceso al medio para transmitir sus datos; se asemeja a una reunión de amigos.

**Ciente - servidor:** Está definido un cliente, el cual solicita un servicio y el servidor es quien debe ofrecer o proveer el servicio al cliente que lo solicitó. Se asemeja a la atención de un mesero en un restaurante, el cual ofrece un menú al cliente y este último selecciona lo que desea que le sirvan.

**Maestro - esclavo:** En esta configuración se define un maestro, el cual maneja y dispone del acceso al medio; el resto de equipos esperan las disposiciones y asignaciones del maestro, el cual, en el caso de sistemas sincrónicos, es el encargado de generar y transmitir la señal de sincronismo; los esclavos reciben la señal. Se utiliza cuando existe un dispositivo que posee características superiores al resto de dispositivos y, por lo tanto, es el encargado de controlarlos, monitorearlos y comunicarse con los mismos con el fin de generar un trabajo ordenado. Se aplica dentro de las redes industriales en la conexión de las redes de sensores y actuadores a un PLC, el cual hace las veces del maestro.

### *Por el modelo de comunicación*

**Fuente destino:** Se presenta cuando cada equipo transmite sus datos hacia un destinatario definido. En caso de que se requiera enviar los mismos datos hacia diferentes equipos, debe transmitir el mensaje por varias ocasiones hasta completar el envío para todos. Por ello, el tiempo que se tarda el envío de los datos hacia todos los equipos destino depende del número de equipos destino. Ejemplo: Comunicación a través de correo convencional.

**Productor - consumidor:** En este modelo los mensajes se envían a todos los equipos y es trabajo de los equipos que reciben los datos el determinar si el dato recibido es de interés, por lo que el tiempo que le toma transmitir el dato al equipo productor es independiente del número de equipos consumidores. Ejemplo: Transmisión de canales de televisión o radio.

*Por el área de cobertura*

**Red de área personal (Personal Area Network PAN):** Son redes de equipos usadas para la comunicación entre dispositivos dentro de una área de trabajo no superior a 5 metros que constituyen el área de trabajo de una persona. Es decir, está compuesta, por ejemplo, por una computadora y todos los elementos periféricos adicionales (teclado, mouse, impresora y teléfono móvil, entre otros) y que están cerca de una persona.

**Red de área local (Local Area Network LAN):** Son redes de equipos que tienen una extensión de entre 10 metros a 1 kilómetro. Por lo general, son redes pequeñas que integran oficinas, pequeñas industrias, colegios y edificios, entre otros.

**Red de área metropolitana (Metropolitan Area Network MAN):** Está constituida por redes de equipos que pueden cubrir el área metropolitana de una ciudad, llegando a alcanzar distancias superiores a los 10 kilómetros. Se utilizan en empresas y organizaciones que tiene distribuidas oficinas a lo largo de la ciudad; también se aplican en el monitoreo de servicios dentro del sector público, como control de servicio de emergencia, control del sistema de semaforización inteligente y monitoreo de la red de transporte, entre otros.

**Red de área extendida (Wide Area Network WAN):** Permite cubrir una gran extensión geográfica, integrando una amplia cantidad de equipos y recursos de comunicación que abarcan un área geográfica extensa y delimitada, como un país o región.

*Por el tipo de señal a transmitir*

**Señales eléctricas:** Se dan cuando la transmisión de datos se realiza a partir de la variación de una señal o de sus características eléctricas, como voltaje, corriente o frecuencia.

Para el propósito de este capítulo, las de interés son las transmisiones basadas en variación de voltaje, definiéndose como las basadas en diferencia de potencial; es decir, en el cambio de diferencia de potencial de positivo a negativo o viceversa para identificar la transmisión

de un 1 o 0. Por lo general definen 2 terminales para la transmisión de los datos D+ y D-, los cuales varían su potencial en función del dato a transmitir; se utiliza, por ejemplo, en la comunicación USB. Otro tipo de transmisión es la basada en nivel de voltaje, la cual establece un nivel de voltaje respecto a una tierra común que define si el dato es un 1 o 0; por lo general, existe un terminal TX de transmisión y un GND común, por ejemplo, RS232. También están las transmisiones por corriente, siendo un estándar a nivel industrial las de 4 a 20 mA; en base a esta variación de corriente se define la transmisión de un 1 o 0; se utilizan cuando se requiere alcanzar distancias largas. Finalmente, también se utiliza a nivel industrial la transmisión por variación de frecuencia de portadora, la cual define una frecuencia que identifica un 1 y otra diferente para un 0.

Señales ópticas: Basan la transmisión de datos en la emisión o no de una señal luminosa, identificando a través de ella la transmisión de un 0 o 1 como dato. Se utiliza tanto en medios guiados como fibras ópticas y en no guiados como transmisiones infrarrojas. La Fig. 5.9 muestra esta forma de clasificación de las comunicaciones.

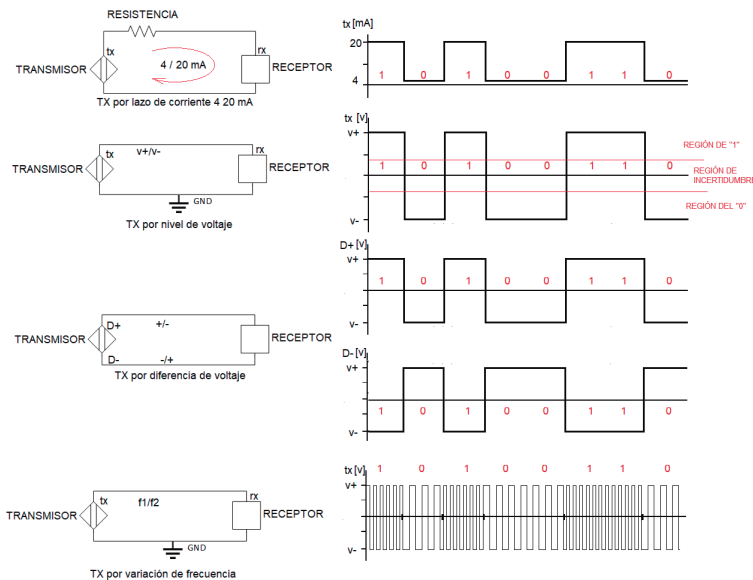


Figura 5.9: Clasificación por el tipo de señal a transmitir

### 5.2.3. Modelo de interconexión de sistemas abiertos (Open System Interconnection, OSI)

El modelo OSI, desarrollado por la Organización Internacional de Normalización (ISO) y lanzado a finales de la década de 1970, ha ganado rápidamente el apoyo de la industria informática, constituyéndose en 1983 como un estándar para el diseño conceptual de las redes de datos que permite entender cómo se generan y transmiten los datos a través de una red. El modelo OSI divide la transmisión de datos a través de una red en siete capas o niveles, como se muestra en la Fig. 5.10. Cada capa tiene una responsabilidad y debe cumplir con una función dentro de esta estructura de comunicación; sin bien es cierto esta estructura es únicamente conceptual, con el paso del tiempo y desarrollo de las tecnologías de comunicación ha ido relacionándose con protocolos reales que se utilizan actualmente en la transmisión de datos a través de redes. A continuación se describe brevemente la función de las capas del modelo OSI.

Cada capa adiciona a los datos su componente de información a ser transmitida de arriba hacia abajo a través de las capas del modelo OSI, sumando todo y formando lo denominado como trama de datos. Esta es enviada al receptor, que realiza el proceso inverso; es decir, desde abajo hacia arriba de lectura de la información hasta llegar a los datos propiamente dichos, como se muestra en la Fig. 5.10.

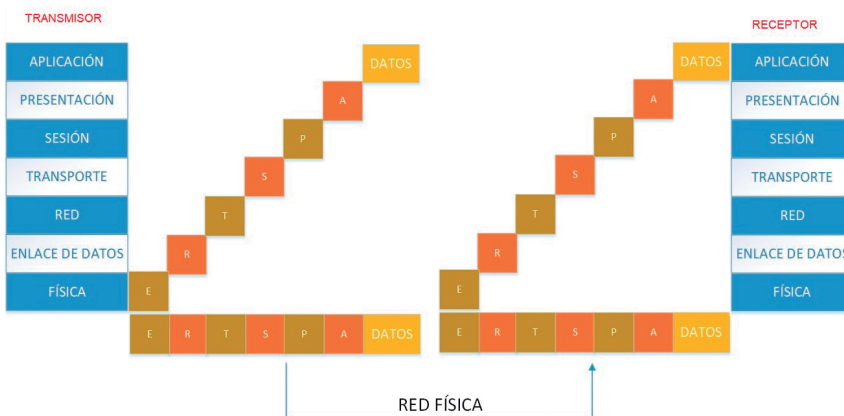


Figura 5.10: Capas del modelo OSI

**Capa de aplicación:** Es la encargada de la provisión de servicios de red a los programas de aplicación del usuario. Ahora bien, los programas de aplicación reales del usuario no están en esta capa, sino los protocolos o aplicaciones internas que se utilizan para proveer determinados servicios a través de las aplicaciones que visualiza y maneja un usuario; por ejemplo, para el servicio de correo electrónico, esta capa consta de los protocolos Post Office Protocol (POP) y Simple Mail Transport Protocol (SMTP), que son los gestores de este servicio para la recepción y el envío de correos electrónicos, respectivamente.

**Capa de presentación:** Asigna las representaciones de datos en un formato de datos externos que permitirá la interpretación y presentación correcta de la información al recibirla; en base al formato de presentación adecuado como textos, sonidos, imágenes, valores numéricos e instrucciones, entre otros—, la capa de presentación interpreta las estructuras de las informaciones intercambiadas por los procesos de la aplicación y las transforma convenientemente. La asignación también puede incluir cifrado o compresión de datos.

**Capa de sesión:** Encargada del control de las comunicaciones entre usuarios. Esto incluye la agrupación de mensajes y la coordinación de la transferencia de datos entre capas agrupadas. También efectúa puntos de control y verificación de contraseñas para el inicio, la recuperación y el cierre de sesiones.

**Capa de transporte:** Se encarga de la gestión de las comunicaciones entre los dos sistemas finales, detección o corrección de errores, mantener el flujo de datos en la red; está a cargo de proporcionar un transporte de datos confiable y seguro entre los usuarios fuente y destino.

**Capa de red:** es la responsable del control de la red, del enrutamiento de datos, direccionamiento de red, fragmentación de paquetes en tamaños más pequeños, congestión y control de flujo de datos en la red. Para el enrutamiento, es la encargada de establecer criterios como velocidad, seguridad, distancia y retardos, entre otros. Los equipos encargados del enrutamiento son los routers.



**Capa de enlace de datos:** Envía la trama de datos de un sistema a otro, garantizando que los errores en el flujo de bits recibido no pasen al resto de capas, utilizando técnicas de detección y corrección de errores.

**Capa física:** Define las conexiones eléctricas, como nivel de voltaje, funciones de pines o velocidad de transmisión y mecánicas, como forma del conector, tamaño de cables o número de pines del conector a nivel físico del canal de comunicación. Las responsabilidades funcionales incluyen modulación, multiplexación y generación de señal. La capa física define, pero no incluye el medio; esto se encuentra debajo de la capa física.

#### 5.2.4. Protocolos de comunicación

Los protocolos son las estructuras utilizadas dentro de un sistema de comunicaciones para que, por ejemplo, una computadora pueda comunicarse con una impresora. Tradicionalmente, los desarrolladores de plataformas de software y hardware han generado protocolos que solo sus productos pueden usar. Para desarrollar sistemas de instrumentación y control más integrados se requiere estandarizar estos protocolos. Los estándares pueden evolucionar a partir del amplio uso del protocolo de un fabricante (un estándar de facto) o pueden ser desarrollados específicamente por organismos que representan una industria, permiten a los fabricantes desarrollar productos que se comunicarán con equipos existentes, lo que para el cliente simplifica la integración de productos de diferentes fabricantes.

Para el caso de las comunicaciones, un estándar de comunicación define los detalles eléctricos y mecánicos que permiten conectar y comunicar equipos de diferentes fabricantes, a partir de definiciones y características técnicas definidas en el protocolo.

Existen diferentes organizaciones que se encargan de establecer normas y protocolos, dentro del área de las comunicaciones como el Institute of Electrical and Electronic Engineers (IEEE), International Electrotechnical Commission (IEC), Electronic Industries Association (EIA), Telecommunication Industries Association (TIA), entre otras a nivel regional y mundial.

RS es una organización voluntaria de estándares en Estados Unidos, especializada en las características eléctricas y funcionales de los equipos de comunicación. Representa principalmente a los fabricantes de equipos electrónicos. Desde la fusión de RS y TIA en 1988, TIA representa el sector de telecomunicaciones de RS y sus iniciales aparecen en ciertos documentos estándar de RS.

RS ha establecido varios protocolos de comunicación serial, como RS232, RS422, RS485, entre otros, que han sido ampliamente utilizados a nivel industrial como comercial.

### **5.2.5. Protocolo RS-232**

La norma RS-232C fue definida por la Asociación de Industrias en Electrónica (Electronic Industries Association EIA), en 1969, en la definición del conector y del cable. Fue creada para comunicar un computador con un módem, sin embargo, en la actualidad, se utiliza a nivel industrial para comunicar punto a punto diferentes componentes y equipos de automatización industrial. El estándar RS-232C cubre cuatro áreas, como características mecánicas, paso de señales eléctricas, función de cada señal y subconjuntos de señales para ciertas aplicaciones [35]. Si bien en su momento se constituyó en un estándar muy usado a nivel comercial, ha sido desplazado por otros protocolos de comunicación, como el bus serial universal (Universal Serial Bus USB), el cual es actualmente el estándar comercial que se encuentra en los equipos de computación comercial; sin embargo, aún sigue siendo utilizada a nivel industrial: para la conexión a equipos comerciales se requiere de conversores USB a RS-232, los cuales se encuentran disponibles en el mercado.

#### *Características mecánicas de la interface*

Dentro de las características mecánicas definidas para este estándar están:

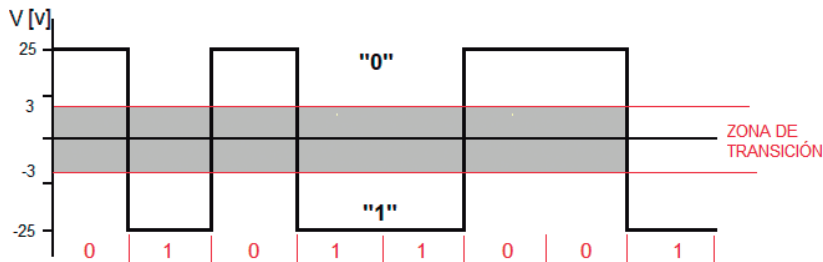
- Asignación de señales del conector de 25 pines o reducido de 9 pines, siendo este el más usado y conocido actualmente.

- Conector macho en los equipos de computadores y conector hembra para el cable y módem.
- Longitud máxima del cable de 15 metros.
- Capacitancia máxima del cable de 2500 pf.

### *Paso de señales eléctricas por la interface*

Dentro de las principales características de las señales eléctricas definidas para el estándar RS-232 constan:

- Velocidades iguales o menores a 20kbps.
- Define niveles de voltaje de -12v a +12v para los estados de la señal, estableciendo un rango de voltaje de transición de entre -3 v y +3 v, como se muestra en la Fig. 5.11.
- El nivel ideal de voltaje para reducción de interferencias es un nivel de voltaje de -25 v para un 1 lógico y +25 v para un 0 lógico.



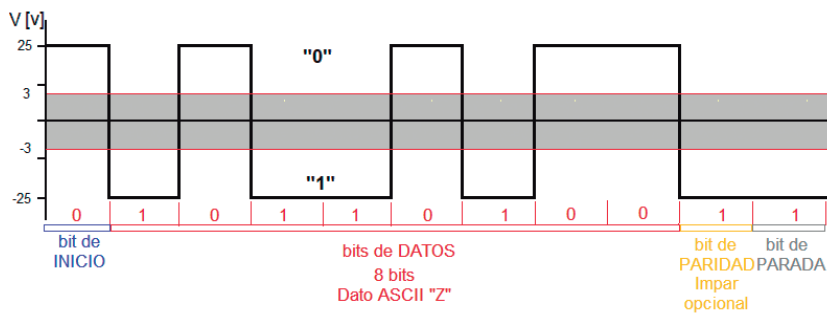
**Figura 5.11:** Niveles de voltaje del protocolo RS-232

La transmisión de los datos es asincrónica; es decir, no tiene una señal de sincronismo que establezca la sincronización de la transmisión, por lo que este protocolo establece la posibilidad de verificación del envío y recepción de los datos a partir de señales adicionales que se indican en la Tabla 5.1, mientras que en la parte del dato como tal, la trama está compuesta por un bit de inicio, que indica el inicio del dato como tal a partir de la transmisión de un 0 lógico, 8 bits de datos, un bit de paridad par o impar que se utiliza para la verificación de errores y

es un elemento opcional y un bit de parada que es representado por un 1 lógico, que indica el fin de la trama, como se muestra en la Fig. 5.12.

**Tabla 5.1:** Pines RS-232, conector de 9 pines

PIN	Señal	Función	Origen	Tipo de señal	Relación PIN
1	CD	Detector de portadora	Equipo	Líneas de módem	CD
2	RD	Recepción de datos	Equipo	Datos	TD
3	TD	Transmisión de datos	PC	Datos	RD
4	DTR	Terminal de datos listo	PC	Control de flujo	DSR
5	SG	Señal de tierra	GND	Tierra común	SG
6	DSR	Modem o línea de datos lista	Equipo	Control de flujo	DTR
7	RTS	Petición de envío	PC	Control de flujo	CTS
8	CTS	Preparado para transmitir	Equipo	Control de flujo	RTS
9	RI	Indicador de señal de llamado acústico	Equipo	Líneas de módem	RI



**Figura 5.12:** Trama de datos del protocolo RS-232

### *Función de señales y subconjunto de señales*

En la Tabla 5.1 se muestra la asignación de pines del conector de 9 pines, relacionando la señal y la función que cumple para el estándar RS-232.

Cabe mencionar que la conexión punto a punto entre dos dispositivos a través de RS-232 implica el cruce de varios pines que representan que la señal de entrada de uno de los equipos son los pines de salida del otro y viceversa; esta relación se muestra en la Tabla 5.1 basado en conectores de 9 pines.

### 5.2.6. Protocolo RS-422

Basado en un sistema de transmisión a partir de diferencia de voltaje, permite la transmisión de datos en ambientes ruidosos, a través de una transmisión de dos hilos, sin referencia a tierra y manejando el concepto de diferencia de voltaje entre los dos hilos; para determinar la transmisión de un dato 1 o 0 se fija una diferencia de voltaje de +6 voltios para un 1 y -6 voltios para un 0, con rango de incertidumbre entre +2 y -2 voltios, como se muestra en Fig. 5.13, permitiendo la transmisión de datos a mayor velocidad [35].

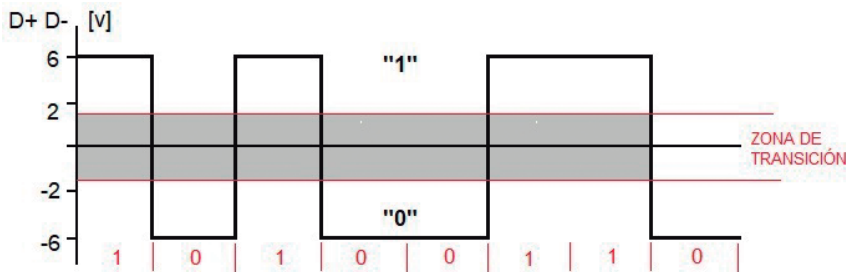


Figura 5.13: Diferencias de voltaje del protocolo RS-422

La norma RS-422 se caracteriza por:

- Utilizar dos hilos o cables, referenciado a masa, por lo que utiliza dos pares trenzados para comunicarse en comunicación full dúplex asincrónica.
- El dato se obtiene de la diferencia de voltaje de la señal eléctrica entre los dos hilos que componen el canal de comunicación; es decir, maneja el concepto de diferencia de voltaje para la transmisión de los datos.

- Tipo de transmisión full dúplex: cada equipo puede enviar y recibir simultáneamente al utilizar canales diferentes, en una comunicación maestro esclavo; es decir, el maestro tiene un canal con el cual se comunica con todos los esclavos y los esclavos tienen un canal de transmisión compartido con el cual se comunican con el maestro, sin que los esclavos puedan comunicarse entre sí.
- Esta norma RS-422 permite velocidades de hasta 10 Mbps a distancias de 12 metros; y hasta un máximo de 1200 metros a 100 kbps; es decir, tiene una relación velocidad-distancia inversamente proporcional.
- Permite la configuración de una comunicación con un máximo de 10 receptores o esclavos.
- Utiliza un conector mini-DIN-8.
- Requiere el uso de terminadores de 120 ohmios en los extremos de cada canal para mantener la impedancia, en una topología bus, como muestra la Fig. 5.14.

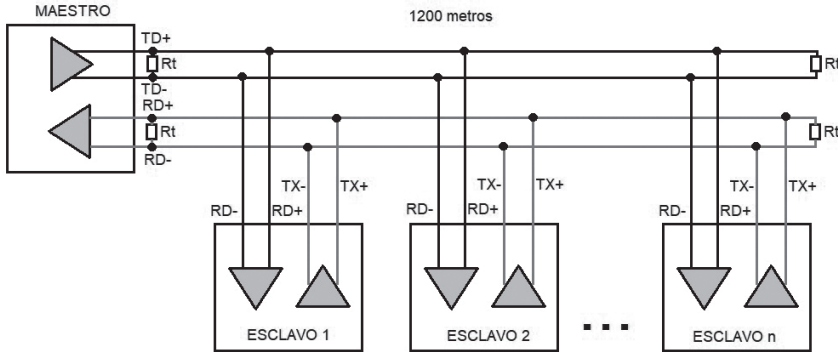


Figura 5.14: Esquema de conexión del protocolo RS-422

### 5.2.7. Protocolo RS-485

En la búsqueda de solucionar los problemas de las anteriores normas respecto al tipo de comunicación, el número de dispositivos a comunicar, entre otros, la EIA definió un nuevo estándar identificado como

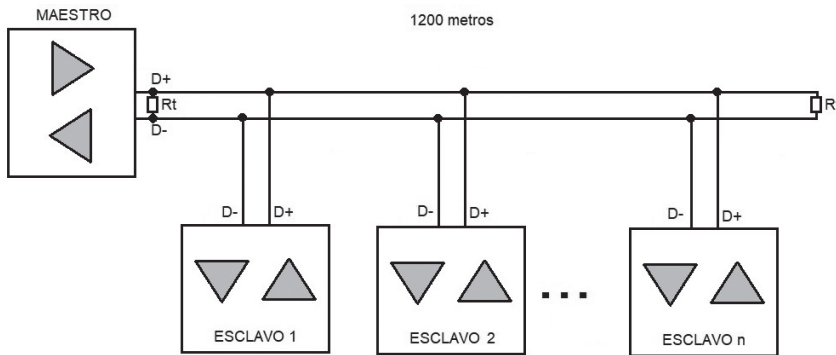
RS-485. Introducida en 1983, es la versión mejorada de RS-422. Se considera como interface multipunto y permite la comunicación de hasta 32 equipos emisores-receptores en un bus de datos común, por lo que dispone del tercer estado (habilitación) para que no existan colisiones en el canal de comunicación, satisfaciendo al mismo tiempo los requerimientos de RS-422 [35].

La norma RS-485 se basa en un par trenzado con terminales D+ y D- bidireccional transmitiendo en un sistema diferencial de voltaje, similar al del protocolo RS-422, que permite eliminar los posibles ruidos que se puedan incorporar en el canal de comunicación, agregando un tercer estado de espera a los equipos conectados.

La norma RS-485 se caracteriza por:

- Utilizar dos hilos o cables, referenciado a masa, por lo que utiliza un par trenzado para comunicación half dúplex asincrónica.
- El dato se obtiene de la diferencia de voltaje de la señal eléctrica entre los dos hilos que componen el canal de comunicación; es decir, maneja el concepto de diferencia de voltaje para la transmisión de los datos.
- Tipo de transmisión half dúplex, cada equipo puede enviar y recibir de forma no simultánea al utilizar el mismo canal para transmitir y recibir datos, en una comunicación maestro-esclavo; es decir, el maestro ordena y gestiona el manejo del bus a través de sus mensajes en el canal bidireccional y pasa a esperar la respuesta de los esclavos; existe la posibilidad de comunicación entre esclavos, ya que cuando cualquier equipo transmite, todos los restantes reciben los datos.
- Esta norma permite velocidades de hasta 10 Mbps a distancias de 12 metros y un máximo de 1200 metros a 100 kbps (tiene una relación velocidad-distancia inversamente proporcional).
- Permite la configuración de una comunicación con un máximo de 32 equipos.

- Utiliza un conector DB9, similar al del RS232, pero en configuración de conector del equipo y del cable inversa, para evitar errores de conexión, es decir, el equipo posee un conector hembra y el cable, un conector macho.
- Requiere el uso de terminadores de 120 ohmios en los extremos del canal para mantener la impedancia, en una topología bus, como muestra la Fig. 5.15.



**Figura 5.15:** Esquema de conexión del protocolo RS-485

Es ampliamente utilizado como protocolo físico en varios protocolos a nivel de campo y control utilizados hasta la actualidad a nivel industrial, como Profibus, Modbus, entre otros.

### 5.2.8. Ethernet

Diseñado y lanzado por Xerox en la década de 1970, inicialmente, se utilizaba como medio el cable coaxial grueso de 50 ohmios de impedancia; actualmente, utiliza un par trenzado y fibra óptica. Es la forma de comunicación más difundida y usada a nivel mundial; es un estándar de comunicación en redes de datos de tipo LAN. Permiten la comunicación a través de protocolos comunes; además, define todos los dispositivos requeridos para la comunicación de varios equipos dentro de una sola red de datos y comunicación con equipos remotos, a través de las redes de WAN o internet.



Ethernet está definida por el estándar IEEE 802.3 solo desde el aspecto físico y especificaciones de la capa de enlace de datos. Se han implementado muchas tecnologías de red sobre Ethernet, lo que les permite coexistir en la misma red pero no interactuar entre sí. Por ejemplo, Internet usa TCP/IP; las redes Novell, un protocolo de nivel superior llamado IPX; y las redes de Microsoft, NetBEUI, todo para los mismos propósitos. Ethernet se ha utilizado también para redes de nivel de control patentado en automatización industrial durante muchos años. Proveedores de sistemas de control Siemens, Foxboro, Siebe, Invensys, Rockwell, Koyo, Automation Direct, Schneider y Square-D tienen sus propios protocolos únicos de nivel superior que utilizan Ethernet para las capas de enlace físico y de datos [36].

Ethernet trabaja dentro del modelo OSI, específicamente en las dos capas inferiores, es decir, en la capa de física y en la mitad inferior de la capa de enlace de datos, que es denominada subcapa de control de acceso al medio (Media Access Control, MAC). Ethernet dentro la capa física define señales, los de bits y tramas que se transmiten en el medio, componentes físicos, como conectores que transmiten las señales a los medios y distintas topologías. En la capa de enlace, se encarga de la compatibilidad tecnológica con otros equipos y la comunicación con el equipo. La subcapa MAC es responsable de los elementos físicos que se utilizan para transmitir la información y prepara los datos para enviarlos a través de los medios. La subcapa control de enlace lógico (Logical Link Control, LLC) sigue siendo relativamente independiente del equipo físico que se utilizará para el proceso de comunicación.

El protocolo Ethernet cumple con objetivos originales planteados desde su diseño y desarrollo, como simplicidad, bajo costo, compatibilidad, direccionamiento flexible, equidad, progreso, bajo retardo, estabilidad, mantenimiento y arquitectura en capas.

Para Ethernet, el estándar IEEE 802.2 describe las funciones de la subcapa LLC, define de los tipos de Ethernet (ver Tabla 5.2); y el estándar 802.3 describe las funciones de la subcapa MAC, métodos de acceso al medio y de la capa física. El control de enlace lógico se encarga

de la comunicación entre las capas superiores y el software de red y las capas inferiores del hardware.

### *Tipos de Ethernet*

Existen versiones de Ethernet que especifican y se diferencian por 3 aspectos principales: velocidad de transferencia de datos, longitud máxima de segmentos, tipo de medio y métodos de señalamiento y acceso al medio. La Tabla 5.2 muestra los diferentes tipos de Ethernet.

En la actualidad, si bien es cierto que se encuentran a nivel industrial versiones tanto de Ethernet como de Giga Ethernet, el más utilizado es el Fast Ethernet.

**Tabla 5.2:** Tipos de Ethernet

<b>Tipo</b>	<b>Implementación</b>	<b>Velocidad</b>	<b>Tipo de cable</b>	<b>Distancia</b>
Ethernet	10 Base-T	10 Mbps	UTP CAT 3	100 m
	10 Base-5	10 Mbps	RG8 O RG11	500 m
	10 Base-2	10 Mbps	RG58	185 m
Fast Ethernet	100 Base-TX	100 Mbps	UTP CAT5	100 m
	100 Base-FX	100 Mbps	Fibra óptica	2 000 m
	100 Base-T	100 Mbps	UTP CAT5	100 m
Gigabit Ethernet	1 000 Base-T	1 000 Mbps	UTP CAT5e	100 m
	1 000 Base-TX	100 Mbps	UTP CAT6	100 m
	1 000 Base-SX	1 000 Mbps	Fibra óptica	500 m
	1 000 Base-LX	1 000 Mbps	Fibra óptica	5 000 m
10 Gigabit Ethernet	10 GBase-SR	10 Gbps	Fibra óptica	300 m
	10 GBase-LR	10.3 Gbps	Fibra óptica	10 000 m
	10 GBase-LX4	10.3 Gbps	Fibra óptica	300/10 000 m
	10 GBase-ER	10.312 Gbps	Fibra óptica	40 000 m
	10 GBase-LMR	10.312 Gbps	Fibra óptica	220 m
	10 GBase-CX4	10.312 Gbps	Cable de cobre	15 m
	10 GBase-T	10.00 Gbps	Fibra óptica	100 m

## 5.2.9. Clasificación de las redes industriales

Por su relación con la pirámide de automatización, dentro del campo industrial, las redes se clasifican en base al nivel en el cual se encuentran respecto a la pirámide. En otras palabras, existen redes o buses de campo, control y a nivel supervisión cabe recordar, como ya se mencionó anteriormente, que en los niveles de gestión se utilizan redes y equipos comerciales, por lo que no entran dentro de este análisis desde el punto de vista de su conectividad utilizando redes y equipos industriales.

### *Redes o buses de campo*

Se encuentra en el nivel de campo de la pirámide de automatización y debe satisfacer los siguientes requerimientos, algunos de ellos ya mencionados en la descripción del nivel de campo:

- Los sensores y actuadores que la componen están en contacto directo con el proceso y la materia prima.
- Se conecta una gran cantidad de elementos, formada por toda la instrumentación necesaria para controlar y regular un proceso.
- La cantidad de información es baja, debido a que, en el caso de tratarse sensores digitales on-off, la señal de interés será únicamente 1 o 0 que representa su activación o desactivación. Para sensores analógicos, su valor depende de la resolución del sensor que está contenida en un máximo de 2 bytes.
- Las velocidades de transmisión son bajas; por lo general, existe una relación distancia velocidad inversa.
- Cubren distancias en el orden de las centenas de metros, llegando incluso a los miles.
- Presentan tiempos de respuesta, latencia bajos y manejo del tiempo real (real time), por lo que deben manejar sistemas de transmisión y recepción determinísticos.
- En su camino puede encontrarse con fuentes de ruido e incluso puede entrar en contacto con residuos químicos del proceso, por

lo que los medios de transmisión y cables utilizados en este nivel deben ser lo suficientemente robustos.

- Las redes de campo se basan en protocolos que, por lo general, no trabajan en todas las capas del modelo OSI; por el contrario, para simplificar su funcionamiento, se basan, en la mayoría de los casos, en 3 capas: capa de aplicación, de enlace de red y física, lo que reduce las tramas de datos y facilita la rápida comunicación entre los equipos.
- Por lo general, utiliza topologías de bus o anillo, aunque en versiones y protocolos más modernos ya se ha abierto a otras topologías.
- Se clasifican en tres tipos de redes: de sensores, de actuadores y de dispositivos, dependiendo de una relación entre funcionalidad, velocidad y distancias.

**Redes de sensores:** En el nivel más bajo de funcionalidad dentro de la red, está la red de sensores. Están diseñados para ser económicos, ya que un proceso industrial requiere de muchos sensores. Estos proporcionan datos básicos al sistema de control, como la posición de un objeto o el valor de una variable física como la temperatura. Sus mecanismos varían según la precisión y fiabilidad necesarias para el proceso y los requerimientos de calidad del producto a ser elaborado.

Estas redes se caracterizan por su baja funcionalidad, es decir, concentra sus esfuerzos en la comunicación con la menor latencia y variación del tiempo de actualización del dato, pues, en este nivel, como ya se mencionó en la pirámide de automatización, prima la velocidad de respuesta de la red y se sabe que la cantidad de datos es baja pese a que pueden existir muchos dispositivos a conectar.

A este nivel hay una gran cantidad de protocolos disponibles, como ASi, CAN, Profibus-PA, Fieldbus, DEVICENet, entre otros; depende de las necesidades, requerimientos, accesorios, costos, disponibilidad, número de equipos, distancias, entre otros factores, los que determinen la opción a implementarse.

### *Redes o buses de control*

Son redes o buses que tienen como propósito específico la comunicación en topología bus, estrella, árbol o anillo de autómatas programables o PLC's, por lo que tienen una amplia variedad de opciones y protocolos: Controlnet, Profibus DP, Data Highway +, Industrial Ethernet, entre otros. Sin embargo, debe considerarse que, con el desarrollo tecnológico y la inserción acelerada de las tecnologías de la información en el ámbito industrial, relacionada con la implementación de la Industria 4.0, los buses basados en sistemas de comunicación serial y protocolos como RS-485 están siendo sustituidos por las redes basadas en industrial ethernet. Entonces, la línea que separa estos buses o redes del nivel de supervisión es cada vez más delgada. Por esto, en determinados casos pueden verse implementados a través de los mismos protocolos que forman parte del nivel de supervisión como Industrial Ethernet o Profinet.

### *Redes a nivel de supervisión*

Están más cercanas a los niveles administrativos; se puede considerar como un interface entre el área productiva y la administrativa de una industria, por lo que en la actualidad ha tomado un gran impulso y relevancia para la implementación de plataformas y tecnologías relacionadas con la Industria 4.0 y sus recursos —big data y análisis de datos—. En este nivel se tienen principalmente equipos informáticos, como PC, servidores, routers, impresoras y monitores, entre otros, en lo que tiene que ver con hardware. Por ello, la conexión de estos equipos es principalmente por medio de redes de datos, por lo que Industrial Ethernet o Profinet son las principales opciones de conexión con la parte de producción, en combinación con Ethernet para la conexión con la parte administrativa, conviviendo en el mismo entorno de red. Respecto al software, se pueden encontrar recursos de conectividad, control y monitoreo de procesos como OPC server, bases de datos, herramientas de análisis de datos, softwares SCADA como Intouch, Factory Talk, WinCC, entre otros.

### 5.2.10. Modbus

El Modicon Bus (Modbus) es un protocolo de comunicación industrial a nivel de campo y control. Tiene una amplia difusión y uso a nivel industrial por ser uno de los primeros buses de campo lanzados a nivel industrial basado en las normas físicas RS-232 y RS-485, ser un protocolo abierto, que se ha actualizado con el paso del tiempo a una versión compatible con Ethernet, como es Modbus TCP.

Modbus es un protocolo de comunicaciones serial ubicado en el nivel 2 del modelo OSI, desarrollado y publicado por Modicon para su gama de controladores lógicos programables y lanzado en 1979. No es propiamente un estándar de red, incluyendo todos los aspectos, desde el nivel físico hasta el de aplicación, sino a un protocolo de enlace. Puede, por tanto, implementarse con varios tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio que permite parametrizar sus productos. De hecho, en su versión serial, trabaja únicamente en 3 capas del modelo OSI, como la mayoría de los protocolos a nivel de campo y control; es decir, en las capas de aplicación, enlace y física.

La organización de datos de todos los PLC's es como un conjunto de registros direccionables organizados en conjuntos como E/S, relé de control, entradas analógicas, salidas analógicas y variables. Se entiende que las E/S del PLC están organizadas de modo que cada dispositivo de entrada digital discreta aparezca como un solo bit en los registros de E/S de acuerdo con su ubicación en el hardware. Esto, generalmente, significa que la salida discreta digital a menudo se mezcla en algunos de los registros, lo que requiere una máscara de registro para definir las salidas. El conjunto de comandos idéntico para Modbus se ha implementado utilizando varias capas físicas (ver Tabla 5.3) [36].

**Tabla 5.3:** Tipos de Modbus [36]

Capa física	Estándar	Velocidad	Comentario
Modbus	EIA-232	19.2 kbps	Modbus serial original
	EIA-485	Superior a 1 Mbps	Modbus serial de alta velocidad
Modbus Plus	ISO 3309 sobre EIA-485	Superior a 1 Mbps	Bus propietario multipunto con paso de testigo
Modbus TCP	IEEE 802.3	10/100/1 000 Mbps	Modbus sobre Ethernet

### 5.2.11. Profibus

El bus de proceso de campo (Process Field Bus Profibus) es un protocolo de comunicación a nivel de campo y control. Posee tres versiones que permiten la comunicación tanto a nivel de campo —entre PLC's, variadores de velocidad, HMI's y periféricas descentralizadas, entre otros— y a nivel de campo —con sensores y actuadores, siendo un protocolo normalizado por la norma IEC 61158 y EN 50170, abierto, robusto y respaldado por las principales empresas de la industria de automatización que proveen equipos y accesorios compatibles con este estándar de bus abierto.

Constituye un recurso tecnológico comprobado y de amplia utilización dentro de las instalaciones industriales, por lo que su instalación y uso es muy frecuente en segmentos de la industria productiva nacional. Pese a ser considerado un clásico dentro de los buses de campo y control, es aún vigente por la alta penetración de sus equipos dentro de la automatización de procesos, teniendo una amplia difusión, soporte técnico y aplicaciones exitosas.

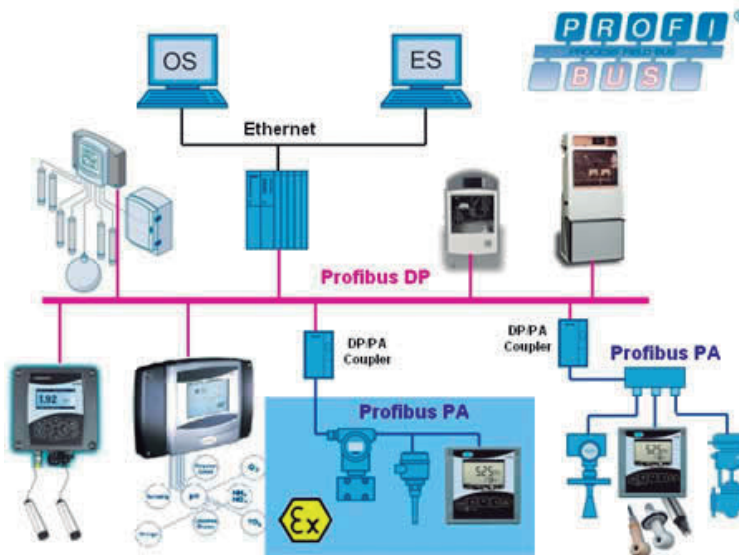


Figura 5.16: Profibus

Inicialmente, se desarrolló Profibus FMS (Fieldbus Message Specification). Se trataba de un protocolo complejo que ya no se utiliza en la actualidad, desplazado por las redes industriales. Después, trabajaron para crear el protocolo Profibus DP (periféricos descentralizados) orientado a controladores y periféricas descentralizadas en nivel de control y, en 1998, se lanzó Profibus PA (automatización de procesos), destinado a conectar sensores y actuadores en nivel de campo (ver Fig. 5.16). Profibus, por lo tanto, se divide en tres tipos de arquitectura principales:

### *Profibus FMS*

Ofrece una amplia selección de funciones en comparación con otras variantes. Es la solución en el estándar de comunicación universal que puede utilizarse para resolver tareas de comunicación complejas entre los PLC's y los DCS. Soporta la comunicación entre sistemas de



automatización, además del intercambio de datos entre equipos inteligentes, generalmente utilizados en el nivel de control. Recientemente, como su función principal es la comunicación entre equipos, está siendo sustituida por el protocolo Ethernet Industrial.

### *Profibus DP*

Es la solución de alta velocidad, basada en protocolos de bus RS-485, desarrollada específicamente para la comunicación entre los equipos automatización como PLC's y los equipos descentralizados, por lo que es aplicable a nivel de control. Enfatiza el acceso a dispositivos distribuidos. Utiliza la interfaz estándar de la capa física de comunicación RS-485 o fibra óptica. Requiere menos de dos minutos para transmitir 1 Kbyte de E/S y se usa principalmente en sistemas de automatización industrial.

Profibus DP está disponible en tres versiones: DP-V0 (1993), DP-V1 (1997) y DP-V2 (2002). El origen de cada versión se relaciona con el avance tecnológico y la creciente demanda de las aplicaciones. Por su versatilidad y características fue ampliamente difundido en la década de 1980 y 1990, por lo que pese a la evolución y desarrollo de las redes industriales, aún sigue vigente como bus de campo.

### *Profibus PA*

Cumple con los requisitos de la automatización industrial, donde los sistemas de automatización y los de control de procesos se conectan con los equipos de campo, como transmisores de presión y temperatura, convertidores, posicionadores, etc. También puede sustituir el estándar de 4 a 20 mA.

Existen ventajas potenciales para el uso de esta tecnología, que conlleva ventajas funcionales, como la transmisión de información confiable, manejo de estados variables, sistemas de seguridad de fallas y la característica de equipo de autodiagnóstico, medición con alta resolución, integración con control de alta velocidad, etc.

Profibus PA permite la medición y el control a través de una línea y dos cables individuales. También alimenta los equipos de campo en zonas intrínsecamente seguras. Permite el mantenimiento y la conexión/desconexión de los equipos incluso durante el funcionamiento sin interferir en otras estaciones en zonas [37].

Pese a manejar la conexión a través de tecnología de bus, su gran difusión y uso en sistemas de automatización lo mantienen vigente pese al paso del tiempo; en la actualidad existen nuevas aplicaciones que lo utilizan.

### **5.2.12. Profinet**

En el contexto de la evolución de las comunicaciones y la integración de los sistemas planteada por TIA Portal, PROFINET es considerado la evolución lógica del bus de campo Profibus DP y de Industrial Ethernet [35].

La Red de Proceso de Campo (Process Field Net PROFINET) es un estándar abierto que cumple la norma IEE 61158 para la automatización industrial y que se basa en Industrial Ethernet. Utiliza los estándares de TI hasta el nivel de campo y permite emplear una misma ingeniería para toda la instalación.

A través de este protocolo se pueden integrar elementos de campo y control como periféricas descentralizadas, PLC's y HMI's, entre otros en combinación con elementos y dispositivos comerciales, como PC e impresoras, sin necesidad de recursos o requerimientos adicionales en los equipos a más de una tarjeta de red convencional. Asimismo, establece la utilización de cables, conectores y elementos de red estándar, lo que la hace de fácil instalación, administración e implementación; obviamente, dentro del ámbito productivo debe contar con elementos lo suficientemente robustos para trabajar en escenarios agresivos, por lo que dispone de varias tecnologías y recursos complementarios para su utilización, como IWLAN (Industrial Wireless Lan), RCoax (cable radiante) y cables, switch, routers y conectores industriales con mayor nivel de robustez y aptos para trabajar en ambientes hostiles.

Dado que Profinet se basa en TCP/IP estándar, estas comunicaciones no pueden configurarse para actividades estrictas en tiempo real en las que se requiere una respuesta determinista en menos de 50 ms. Entonces, para respuestas en tiempo real entre 5 ms y 50 ms, existe un software de extensión en tiempo real (SRT) para Profinet que normalmente ofrecen los proveedores de sistemas de control. SRT es una modificación de TCP/IP, lo que la hace más adecuada para un servicio de LAN local al eliminar las adaptaciones originalmente utilizadas por Internet. Para adaptar Profinet aún más para velocidades muy altas con tiempos de respuesta de aproximadamente 1 ms en aplicaciones como control de movimiento, existe un protocolo asistido por hardware llamado isócrono en tiempo real (IRT) que se agrega a Profinet. En IRT, Ethernet, el hardware en cada dispositivo de red contiene un conmutador de 4 puertos para permitir la redundancia, cuando sea necesaria y para eliminar la sobrecarga de red TCP/IP; solo el segmento IRT utiliza estas extensiones de la tecnología de conmutación Ethernet, pero sigue siendo compatible con TCP/IP estándar [36].

Profinet es un protocolo que se acopla a las exigencias actuales de redes de comunicación industrial y TI, ofreciendo tres opciones de comunicación:

- TCP/IP: Adecuada para funciones no deterministas, como parametrización, transmisiones de video y audio y transferencia de datos a sistemas verticales de TI; se utiliza principalmente a nivel de supervisión, dependiendo del número de dispositivos conectados y la cantidad de datos a nivel de control.
- Profinet en tiempo real (RT): Las capas TCP/IP se omiten para lograr un protocolo con rendimiento determinista para las aplicaciones de automatización, en el rango de 1 a 10 ms. Profinet RT es una solución basada en software adecuada para aplicaciones de I/O típicas, incluyendo control de movimiento y requisitos de alto rendimiento, por lo que se utiliza principalmente a nivel de campo o control.

- Profinet en tiempo real isócrono (IRT): La priorización de señales y la conmutación programada ofrecen sincronización de alta precisión para aplicaciones críticas de alta exigencia en el tiempo de respuesta, tales como control de movimiento. Es posible lograr velocidades de ciclo en rangos inferiores a los milisegundos con el jitter en el rango de los microsegundos. Este servicio requiere compatibilidad de hardware y la utilización de equipos específicos compatibles con esta tecnología [38].

No es una red propietaria, sino, más bien es ofertada por varios fabricantes, destacándose Siemens, Phoenix Contact, ABB, Bosch, Festo, entre otros. Así, proporciona un amplio soporte para el protocolo y acceso a muchas soluciones de automatización en todos los sectores.

## 5.3. Ejemplos de configuración

A continuación se describen los procedimientos básicos y consideraciones que se deben seguir para configurar una red o bus de datos para los protocolos más comúnmente utilizados en la actualidad, como Profinet DP, Profinet y Modbus, basado en dispositivos Siemens y utilizando el software TIA Portal V15.

### 5.3.1. Configuración Modbus

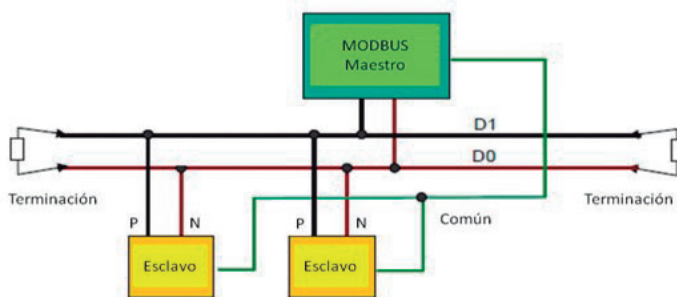
#### *Antecedentes*

Modbus es uno de los primeros buses de campo en alcanzar un gran índice de penetración en el mercado por características como su robustez, fácil configuración, entre otros. Fue desarrollado por la empresa MODICON en 1979, para conectar los PLC's de su marca a través de este bus, basado en protocolos RS-232 y RS-485, conectado por medio de 2 o 4 hilos, en versiones half duplex o full duplex con arquitecturas

maestro-esclavo o cliente-servidor; puede conectar un maestro hasta con 247 esclavos, como muestra la Fig. 5.17. Sigue vigente como un protocolo ampliamente utilizado a nivel industrial, siendo liberado en 2004 y, actualmente, Modbus Organization promueve su uso. Además, ofrece la versión Modbus TCP/IP que es compatible con Ethernet, lo que ha facilitado mantenerlo vigente en el mercado.

Los datos en Modbus pueden transmitirse de dos formas:

- RTU (Remote Terminal Unit): Formato de transmisión binario que envía 8 bits de datos en cada byte transmitido. Es el formato habitual para la transmisión de datos entre equipos.
- ASCII: Formato basado en caracteres ASCII, en que cada grupo de 4 bits del formato RTU se codifica con un carácter ASCII, es decir, el dato ocupa una mayor cantidad de bytes para transmitir la misma información.



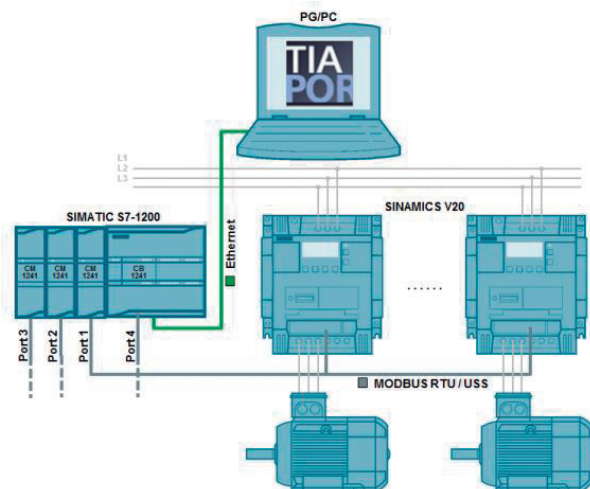
**Figura 5.17:** Conexión Modbus RS-485 Half Duplex [39]

Modbus RTU utiliza transmisión de datos en serie con CM 1241 RS232, CM 1241 RS485 o CB 1241 RS485. En la red Modbus (RTU), una estación maestra controla la transferencia de datos y las estaciones esclavas reciben o proporcionan los datos. En ambas estaciones, la función de comunicación MB\_COMM\_LOAD configura el conector (puerto) en el módulo o en la placa. En la estación maestra, la función MB\_MASTER controla la transferencia de datos; en la esclava, la función MB\_SLAVE, la de datos [40].

Modbus TCP utiliza la interfaz Profinet del PLC. En la red Modbus TCP, una estación cliente controla la transferencia de datos y las estaciones servidor reciben o proporcionan los datos. En la estación del cliente, la función MB\_CLIENT controla la transferencia de datos; en la del servidor, se utiliza la función MB\_SERVER controla la transferencia de datos [40].

Sin embargo, para conectar a través de Modbus con dispositivos específicos y disponibles dentro del catálogo de equipos de TIA Portal, Siemens provee varias librerías que facilitan la configuración y puesta en operación de una red que combine PLC's y estos dispositivos utilizando este protocolo. Tal es el caso de la familia de variadores de frecuencia SINAMICS V20, la cual dispone de una librería de configuración para versiones de TIA Portal y que pueden ser descargados de la página oficial de Siemens:

[https://support.industry.siemens.com/cs/document/63696870/sinamics-v\%3A-speed-control-of-a-v20-with-s7-1200-\(tia-portal\)-via-uss\%C2%AE-protocol-modbus-rtu-with-hmi?dti=0&dl=en&lc=es-ES](https://support.industry.siemens.com/cs/document/63696870/sinamics-v\%3A-speed-control-of-a-v20-with-s7-1200-(tia-portal)-via-uss\%C2%AE-protocol-modbus-rtu-with-hmi?dti=0&dl=en&lc=es-ES) (ver Fig. 5.18).



**Figura 5.18:** Conexión Modbus de Variadores Sinamics V20 con PLC S7-1200 [41]

## Configuración

En este ejemplo se plantea el desarrollo de la comunicación entre el PLC S7-1214C y un variador de frecuencia SINAMICS V20, como se muestra en la Fig. 5.19. Un requisito previo para la configuración de la conexión Modbus propuesta es la instalación de la librería 63696870\_V20\_at\_S7\_1200-MB\_lib\_V15, descargada de la página oficial de Siemens citada en la sección anterior. Una vez descargada, se debe descomprimir el archivo. Para la instalación de la librería se deben seguir los siguientes pasos:

- ① Se abre o crea un proyecto basado en un PLC 1200, en este caso en un PLC 1214C.
- ② Se selecciona la pestaña *Librerías*.
- ③ Se selecciona la pestaña *Librerías globales*.
- ④ Se da clic en *Abrir una librería global*.
- ⑤ Se abre el archivo descargado *\_V20\_at\_S7-1200\_MB\_lib\_V15* (ver Fig. 5.20).

Para la adición de la librería instalada en un proyecto se deben seguir los siguientes pasos:

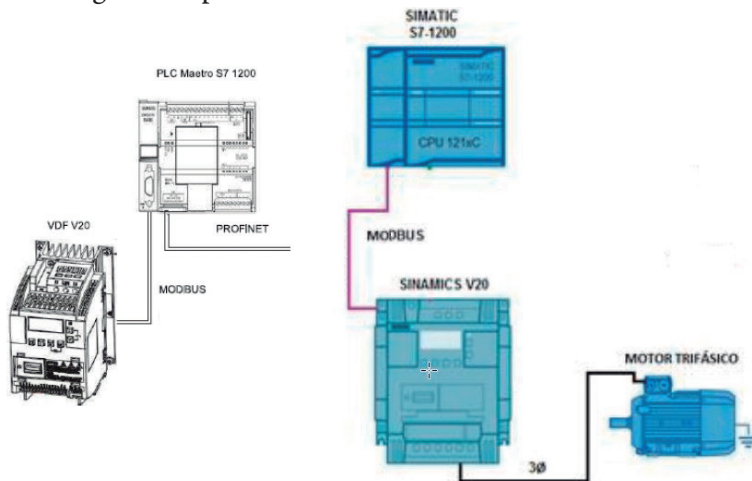


Figura 5.19: Topología del ejemplo propuesto

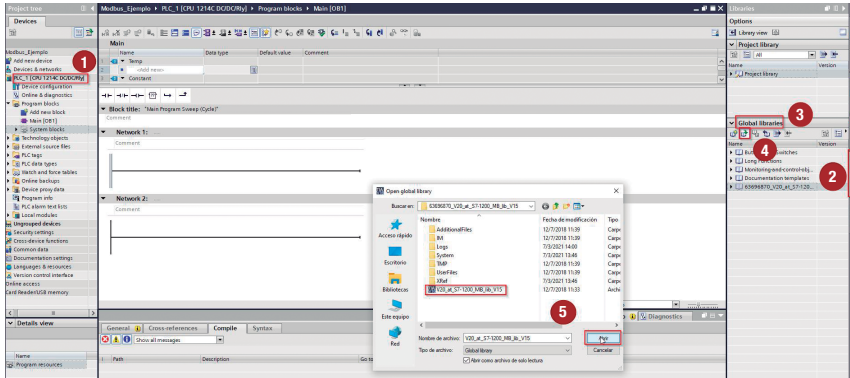


Figura 5.20: Instalación de la Librería Simanics V20 Modbus

- ① En la librería adicionada, seleccionar *Master copies*.
- ② Arrastrar el bloque V20\_Modbus al Ladder y soltar.
- ③ Aceptar la creación del bloque de datos solicitado.
- ④ Adicionar el bloque de programación (ver Fig. 5.21).

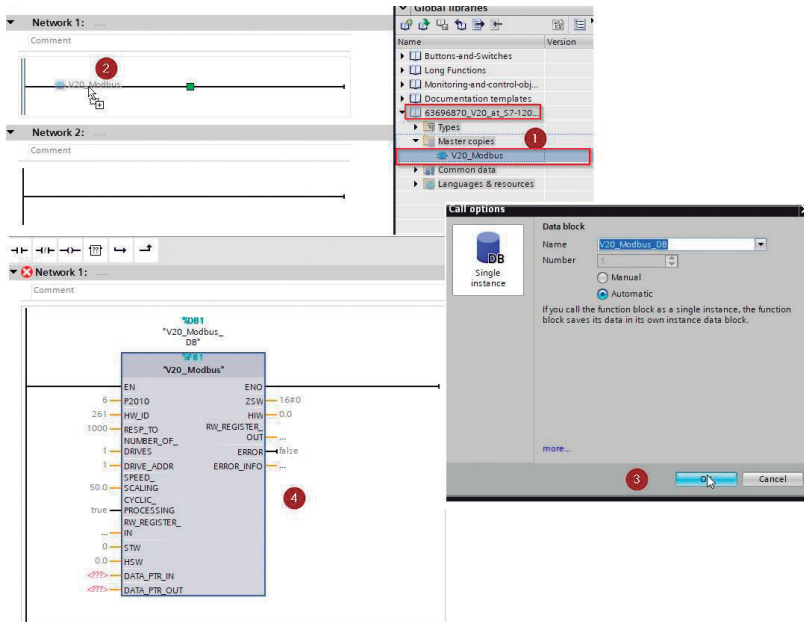


Figura 5.21: Adición de la Instrucción Simanics V20 Modbus



La adición del Módulo de comunicación CM 1241(RS-485) se describe a continuación:

- ① En el catálogo de hardware, seleccionar *Módulos de comunicación*.
- ② Seleccionar la carpeta CM 1241(RS-485).
- ③ Seleccionar y arrastrar el módulo.
- ④ Soltar a la izquierda del PLC.

En *Propiedades*, verificar o modificar los parámetros de trabajo del bus en la pestaña IO LINK (ver Fig. 5.22).

Para el funcionamiento de la instrucción V20\_Modbus se necesitan activar los relojes de memoria y del sistema, que facilitan en el caso del reloj del sistema, el acceso a las banderas del sistema como la de primer scan y, en el caso del reloj de memoria, generación de señales de frecuencias y periodos determinados. El procedimiento se describe a continuación.

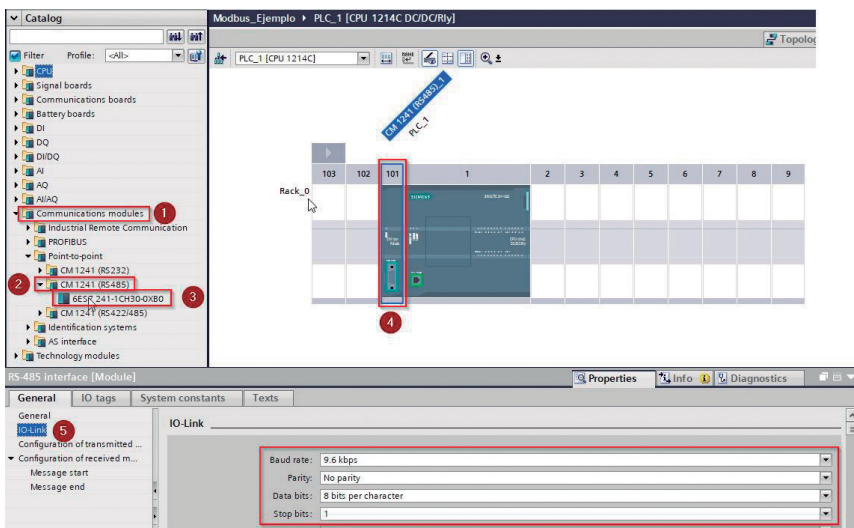


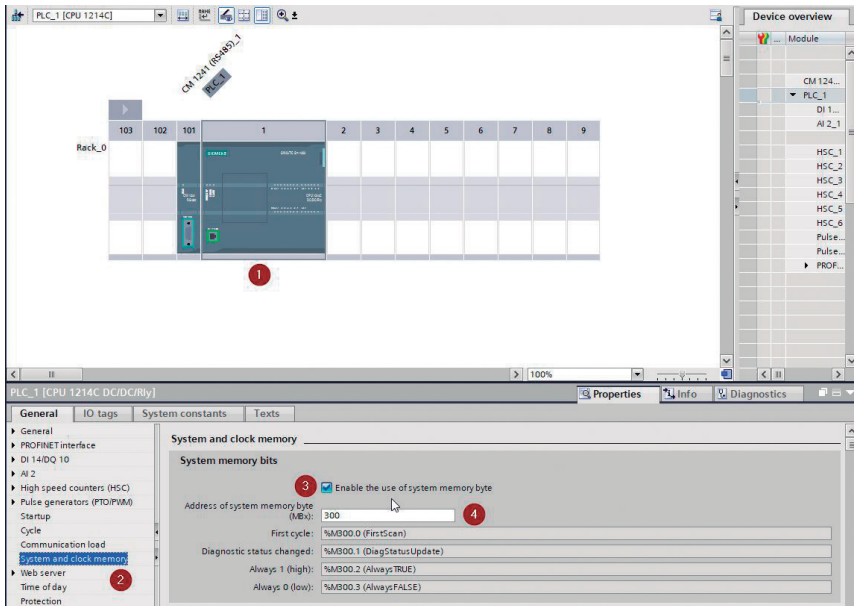
Figura 5.22: Adición del módulo CM 1241(RS-485)

- ① Seleccionar la CPU del PLC adicionado.
- ② Seleccionar relojes de memoria y del sistema.
- ③ Seleccionar la activación del uso del byte de memoria del sistema.
- ④ Asignar un byte de memoria, en el cual se encontrarán los bits del sistema requeridos; por ejemplo, en este caso se designa al byte MB300. En este caso no se necesita la activación del byte del reloj del sistema (ver Fig. 5.23).

Para la configuración de los parámetros de funcionamiento del sistema, la instrucción V20\_Modbus cuenta con parámetros por defecto —que pueden ser modificados con conocimiento del manejo— y los de configuración del variador Sinamics V20 y de la instrucción. Sin embargo, al no ser este el propósito de este ejemplo, se puede trabajar, comunicarse y controlar el funcionamiento del variador a través de dos punteros de registros tipo double Word DB, DATA\_PTR\_IN, a través del cual se puede controlar el variador y registro DB y DATA\_PTR\_OUT que, en cambio, permite el monitoreo del estado del variador.

Los bits de cada registro son los que permiten identificar y controlar el estado del variador, como sentido de giro, frecuencia del variador y activación, entre otros.

Posteriormente, debe configurarse el variador físicamente, lo que permite la comunicación y control del variador a través del PLC por Modbus; el proceso de configuración básica se describe a continuación. Para mayor profundidad en el proceso y parámetros que pueden ser configurados, debe acudir al manual del variador proporcionado por el fabricante, por lo que se recomienda revisarlo y el ejemplo incluido con la librería.



**Figura 5.23:** Configuración de los relojes de memoria y del sistema

- Para resetear el variador, se debe seleccionar la palabra P0970 e ingresar 21. Esto es recomendable en caso de no conocer los parámetros de configuración actual, por lo que, al resetear el variador, se vuelve a los parámetros de fábrica que se pueden conocer al verificar el manual del variador.
- Se deben configurar los parámetros del motor a controlar como voltaje nominal P0304, corriente nominal P0305, potencia en HP P0307, frecuencia del motor P0310 y velocidad en RPM; los datos, en este caso, deben ser tomados de la placa del motor a controlar.
- Configurar los parámetros de la comunicación en P0700 selección de fuente de comando 5 para USS/Modbus RS485, P1000 para USS/Modbus 5, P2010 para 9600 bps 6, P2021 dirección Modbus 1, P2023 para protocolo Modbus 2.

- Configurar los parámetros comunes de funcionamiento, como rampa de arranque en segundos P1120 y rampa de parada en segundos P1121 y P3900 para fin de configuración 1.

Finalmente, en base al mapa de registros, se cargan los valores según el funcionamiento que se le quiere dar al motor en el Ladder del PLC, para proceder a cargar el programa en el PLC y verificar su funcionamiento. Esta comunicación no puede ser simulada en las herramientas de simulación disponibles para TIA Portal.

### 5.3.2. Configuración Profibus DP

#### *Antecedentes*

Profibus DP ofrece una interfaz de acuerdo con el estándar internacional IEC 61158/61784 para la transmisión de datos de proceso entre un módulo de interfaz y dispositivos de campo. Este “módulo de interfaz” se conoce como maestro DP y los dispositivos de campo, como esclavos DP.

El maestro DP es la estación activa en Profibus. Intercambia datos cíclicamente con sus esclavos DP. Un maestro DP puede ser un CPU con interfaz Profibus integrada con las letras DP en la denominación corta; por ejemplo, CPU 315-2 PN/DP o un módulo de comunicación en la estación PLC, por ejemplo, CM1243-5, módulo de comunicación compatible con PLC's S7-1200.

Los esclavos DP son las estaciones pasivas en Profibus DP. Pueden ser estaciones con entradas y salidas de proceso, enrutadores, módulos de enlace u otros PLC que realizan tareas específicas en un proceso. Ejemplos de esclavos DP de periferia descentralizada ET200 son ET200eco, ET200M, ET200S y ET200pro; esclavos DP con datos útiles pueden ser esclavos DP compactos que se direccionan como un solo módulo, esclavos DP modulares que pueden contener varios módulos o submódulos que se abordan individualmente o esclavos DP inteligentes con un área de transferencia configurada como interfaz de datos de usuario al maestro DP.

El maestro DP y los esclavos DP controlados por él, forman un sistema maestro Profibus DP. El tiempo de actualización dentro del cual un esclavo DP recibe datos de su maestro DP y, a su vez, envía datos al maestro DP, depende del número de esclavos DP en el sistema maestro. Profibus DP normalmente opera como un sistema mono-maestro, es decir, un solo maestro DP en un segmento de bus controla varios esclavos DP; el maestro DP es el único maestro en el bus.

También puede instalar varios sistemas maestros DP en una subred Profibus (“sistema multimaestro”). Sin embargo, esto aumenta el tiempo de respuesta en casos individuales, ya que, una vez que un maestro DP ha suministrado sus esclavos DP, los privilegios de acceso se asignan al siguiente maestro DP, que, a su vez, suministra a sus esclavos DP, etc.

Cada estación de la subred PROFIBUS tiene una dirección única dentro de la subred, la dirección de la estación (número de estación), que la distingue de todas las demás estaciones de la subred. La estación (maestra DP o esclava DP) está direccionada en Profibus mediante la dirección de esta estación. TIA Portal asigna las direcciones de las estaciones automáticamente y puede cambiarlas dentro del rango especificado. La dirección de estación más alta se establece en las propiedades del subred o sistema maestro DP en *Configuración de red* [40].

### *Configuración*

Un requisito previo para la configuración de un proyecto basado en Profibus DP es un proyecto con un PLC. Para seleccionar el PLC involucrado, inicie la configuración del hardware en la vista *Red* y haga doble clic sobre el PLC que desee configurar como maestro. Ver Fig. 5.24.

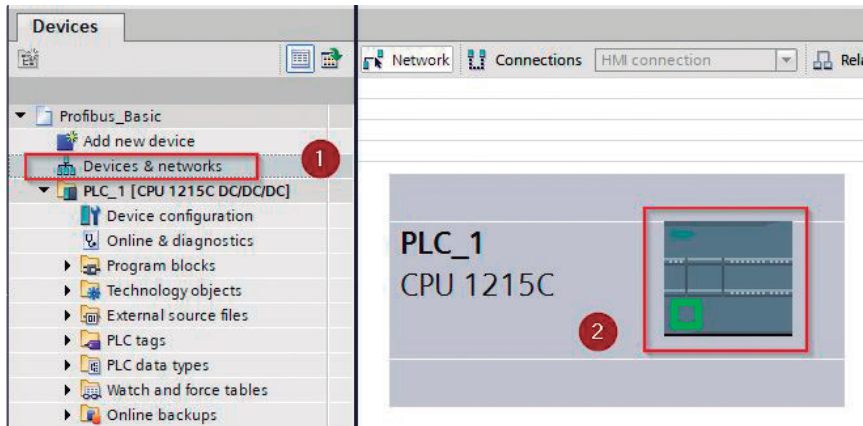


Figura 5.24: Selección del PLC a Configurar como maestro Profibus DP

- ① Seleccionar vista de red.
- ② Hacer doble clic sobre el PLC a configurar

El punto de partida de la configuración del maestro DP para un CPU 1200 es adicionar el módulo de comunicación CM 1243-5, para lo cual debe ser arrastrado y soltado a la izquierda de la CPU (ver Fig. 5.25).

- ① En el catálogo de *Hardware*, seleccionar *Módulos de comunicación*.
- ② En el menú que se despliega, seleccionar *Profibus*.

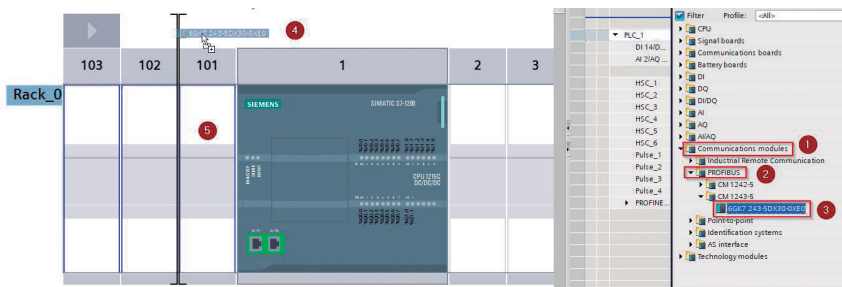


Figura 5.25: Adición del módulo CM 1243-5

- ③ Seleccionar el módulo CM 1243-5.
- ④ Arrastrar el módulo seleccionado con el ratón a la izquierda del PLC.
- ⑤ Soltar el módulo y se adicionará al PLC seleccionado.

Se asigna un sistema maestro Profibus DP a la interfaz Profibus DP; la subred Profibus requerida se crea automáticamente en el proceso; también se pueden configurar los parámetros del bus si es necesario (dirección Profibus más alta, tasa de transferencia, perfil) (ver Fig. 5.26).

- ① Dar clic en el puerto Profibus del módulo CM 1243 adicionado.
- ② En el menú *Propiedades*, seleccionar *Interface DP*.
- ③ Seleccionar el modo de operación.
- ④ Verificar que esté activado *Maestro DP*.
- ⑤ Seleccionar *Profibus address* para configurar los parámetros del bus (opcional, se puede trabajar con los valores por defecto).

Para este caso, se debe adicionar al proyecto un nuevo PLC que hará la función de esclavo Profibus DP. Para lo cual se adiciona al PLC un módulo CM 1242-5 de forma similar a la adición del módulo 1243-5 del maestro Profibus DP (ver Fig. 5.27).

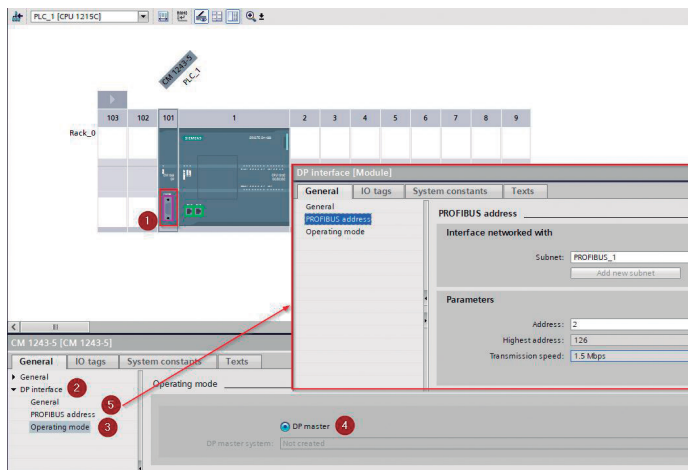


Figura 5.26: Configuración del maestro Profibus DP

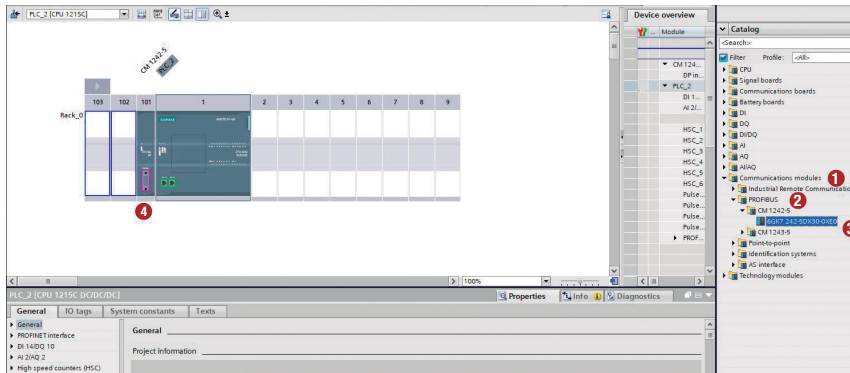


Figura 5.27: Adición del esclavo Profibus DP

- ① En el catálogo de hardware, seleccionar *Módulos de comunicación*.
- ② En el menú que se despliega, seleccionar *Profibus*.
- ③ Seleccionar el módulo CM 1242-5.
- ④ Arrastrar y soltar el módulo seleccionado con el ratón a la izquierda del PLC.

Vincular el esclavo DP al maestro DP, a través de las propiedades del módulo CM 1242-5 (ver Fig. 5.28) o asignando el maestro al esclavo DP y uniendo el bus del esclavo con el ratón al bus del maestro DP en la vista de red (ver Fig. 5.29). Este procedimiento debe repetirse para cada esclavo DP adicional.

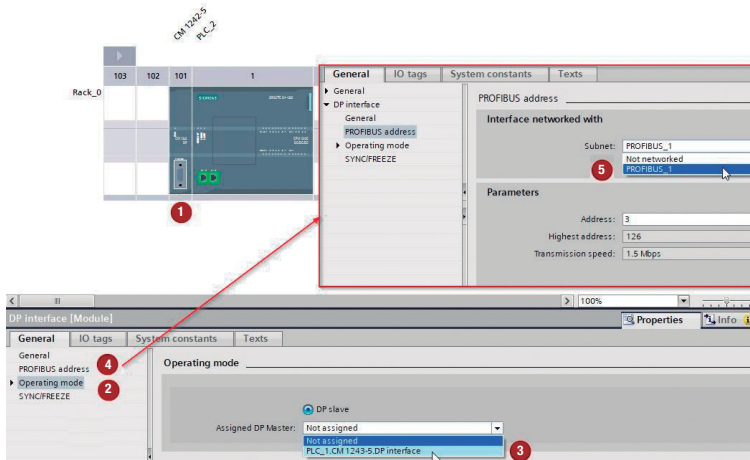
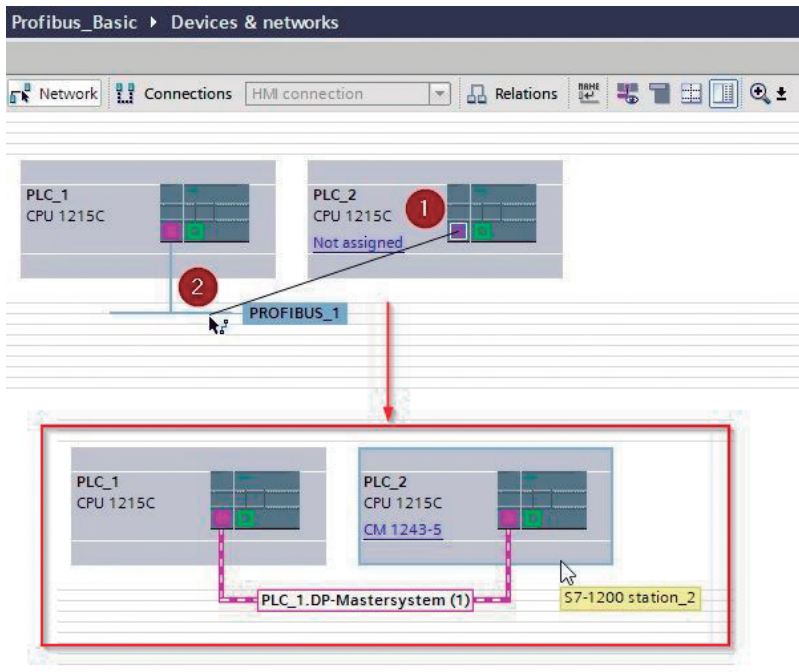


Figura 5.28: Configuración y asignación del esclavo Profibus DP



- ① Dar clic en el puerto Profibus del módulo CM 1242 adicionado.
- ② En el menú *Propiedades*, seleccionar *Interface DP*.
- ③ Seleccionar modo de operación.
- ④ Verificar que esté activado *Esclavo DP* y asignar al PLC\_1 como maestro.
- ⑤ Seleccione *Profibus address* para configurar los parámetros del bus, similares a los de la configuración del maestro DP. En subnet, seleccionar el bus Profibus\_1, si no se modificó en la configuración del maestro.



**Figura 5.29:** Conexión maestro-esclavo Profibus DP

- ① Seleccionar *Vista de red*.
- ② Seleccionar y arrastra con el ratón el puerto de red del módulo del esclavo hasta conectarlo con el bus Profibus del maestro y soltarlo.

- ③ La conexión debe quedar con un bus más grueso que el inicial, como se muestra en la Fig. 5.29.

Configurar el área de transferencia de la interfaz de datos de usuario. En este caso, se adicionaron dos bloques de datos: uno de lectura y otro de escritura, ya que el propósito del ejemplo es demostrar la bidireccionalidad de la comunicación. Ver Fig. 5.30.

- ① En las propiedades del módulo CM 1242-5, seleccionar *Modos de operación*.  
 ② Seleccionar *Comunicaciones esclavo*.

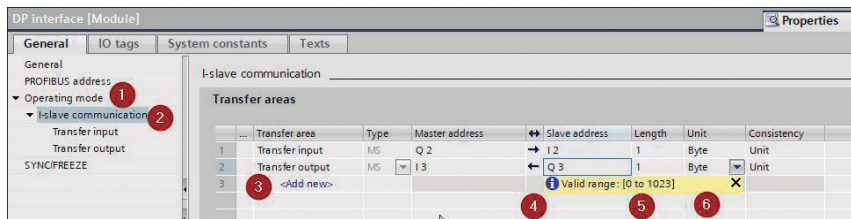


Figura 5.30: Configuración del área de transferencia de datos

- ③ Seleccionar *Add new* y nombrar el bloque de datos.  
 ④ Dar clic en la flecha de sentido de la comunicación como de lectura o escritura respecto al esclavo.  
 ⑤ Seleccionar las direcciones de entrada o salida a asignar tanto en el maestro como en el esclavo.  
 ⑥ Establecer la longitud del bloque de datos; en este caso, está configurado 1.  
 ⑦ Establecer el tipo de dato; puede ser byte (8 bits) o Word (16 bits).

Una vez configurado, falta únicamente la programación del maestro y el esclavo que, en este caso, es el movimiento de los bytes de datos correspondientes a las salidas o entradas válidas, previo a la compilación y carga de los programas en ambos PLC's; el programa del maestro

y el esclavo se presenta en Fig. 5.31 y Fig. 5.32. Para comprobar la comunicación, queda la conexión de los PLC's, la carga de los respectivos programas en los PLC's y la comprobación del funcionamiento.

Finalmente, cabe mencionar que este protocolo no puede ser simulado, ya que los simuladores y herramientas o recursos de virtualización permiten principalmente la simulación de protocolos basados en Ethernet.

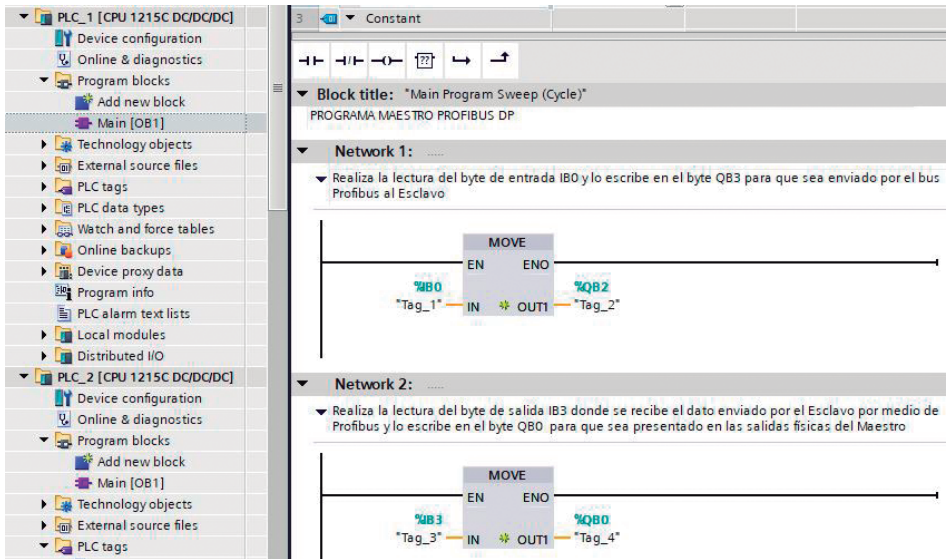


Figura 5.31: Programa del maestro DP

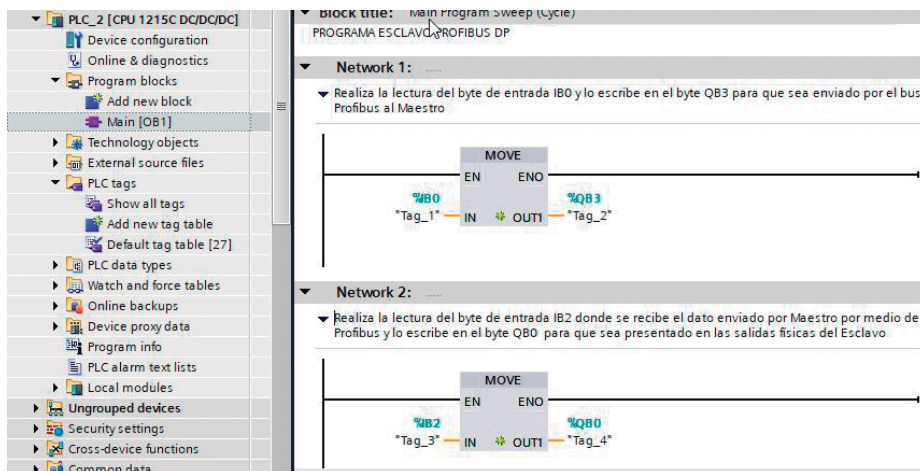


Figura 5.32: Programa del esclavo DP

### 5.3.3. Configuración Profinet

#### *Antecedentes*

Profinet ofrece una comunicación compatible con el protocolo Ethernet, lo que quiere decir que funciona sobre cualquier red de datos, siempre y cuando sean configuradas adecuadamente las direcciones y se conecten los equipos utilizando los cables y conectores en base al nivel de robustez requerido para la comunicación; en este caso, conectores RJ45 comerciales y, para ambientes más robustos, M12.

Profinet permite la conexión con varios tipos de elementos industriales, como PLC's, periféricas descentralizadas, variadores, elementos de red industrial —como switch, routers y WiFi industrial, entre otros— y es compatible con los equipos de tipo comercial como PC, impresoras y servidores, entre otros, lo que lo hace muy versátil y, por lo tanto, es muy utilizado en la actualidad a nivel de control y supervisión.

Es compatible con el protocolo TCP/IP, lo que hace que la configuración de la dirección del equipo consista principalmente en la asignación de la dirección de red, ya que la velocidad de transmisión

está establecida en 10 o 100 Mbps, tal como se establece para redes de datos. Además, en la actualidad, todos los PLC's modernos de la marca Siemens vienen por defecto con un puerto Profinet incorporado y, de ser necesario, también existe un módulo de comunicación.

Antes de conectarse a Ethernet, debe parametrizarse la interfaz Profinet de la CPU a través de la dirección IP y máscara de subred. La comunicación de usuario abierta es un procedimiento para transmitir datos entre dos estaciones conectadas a la subred Ethernet. El intercambio de datos se puede implementar utilizando los protocolos TCP o UDP.

La transmisión de datos con TCP e ISO-on-TCP es segura (con reconocimiento) a través de conexiones configuradas. La transmisión de datos con UDP no es segura (sin reconocimiento) y no tiene conexión. El requisito previo para la comunicación abierta con el usuario es la conexión en red de las estaciones participantes a través de una red Ethernet. La conexión se produce a través de un área de datos. Las funciones de comunicación en tiempo de ejecución utilizan los datos de dirección en esta área de datos para establecer la conexión para los protocolos TCP e ISO-on-TCP o para configurar el punto de acceso de comunicación para UDP.

Para una comunicación de usuario abierta con TCP o ISO-on-TCP, utilice las funciones de comunicación TSEND\_C y TRCV\_C en el programa de usuario. Estas establecen una conexión, controlan el tráfico de datos y, según la programación, finalizan la conexión. Para el protocolo UDP, la función de comunicación TCON configura el punto de acceso para las estaciones asociadas participantes. Las funciones de comunicación TUSEND y TURCV controlan el tráfico de datos, donde es posible el direccionamiento dinámico de la estación remota en tiempo de ejecución; son alternativas de comunicación.

Para la comunicación entre dispositivos industriales como PLC's, se requiere la utilización de los recursos de la denominada comunicación S7; los datos se pueden intercambiar con otras estaciones S7 mediante la comunicación S7 a través de Profinet o Profibus. Las conexiones de comunicación se configuran en la tabla de conexiones.

En la estación local, las funciones de comunicación controlan el tráfico de datos: GET solicita datos de la estación remota, PUT los escribe en la estación remota.

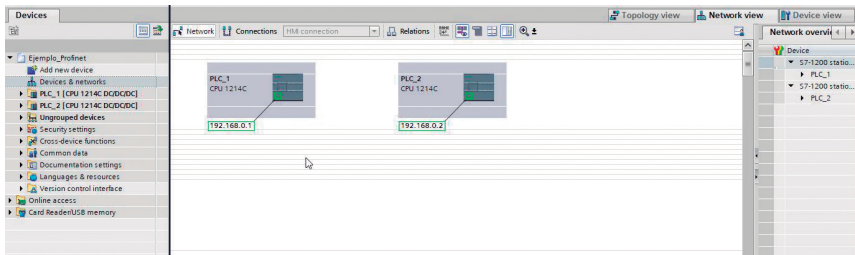
No se requieren funciones de comunicación en el programa de usuario en la estación remota (estación asociada). El sistema operativo de la CPU realiza el trabajo necesario y se considera un intercambio de datos unidireccional. En el programa de usuario de la estación local, se llama a la función de comunicación GET, si los datos deben leerse desde la estación remota, o a la función de comunicación PUT, si los datos deben escribirse en la estación remota. En la estación remota, el sistema operativo controla el tráfico de datos [40].

Los parámetros ADDR\_1 a ADDR\_4, definen las áreas de datos en la estación remota con direccionamiento absoluto. Con la función de comunicación GET, se definen las direcciones de recepción para los datos leídos en los parámetros RD\_1 a RD\_4; en cambio, con la función de comunicación PUT, las direcciones de envío de los datos a escribir en los parámetros SD\_1 a SD\_4. No se requiere ningún programa de usuario para enviar o recibir en la estación remota y esta puede transmitir los datos en los modos de funcionamiento STOP y RUN.

El requisito para la comunicación S7 es la conexión en red de los PLC's que participan en el intercambio de datos a través de Industrial Ethernet. Para la comunicación S7, se debe configurar una conexión S7 en la tabla de conexiones. Las funciones de comunicación para la comunicación S7 se llaman en el programa principal; por ejemplo, en un bloque de función que se llama en el bloque de organización OB\_1 u otro con la clase de evento Ciclo de programa.

### *Configuración*

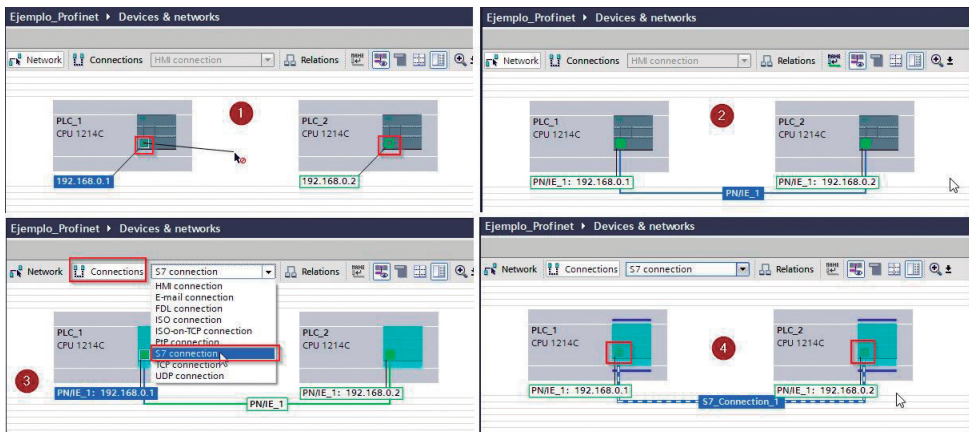
Para la configuración de una comunicación S7, se requiere de un proyecto con la adición de dos o más dispositivos a comunicar. Las direcciones IP de cada PLC deben ser diferentes, pero deben estar dentro de la misma red. Por ejemplo, PLC\_1 dirección 192.168.0.1 y PLC\_2 dirección 192.168.0.2 como se muestra en la Fig. 5.33.



**Figura 5.33:** Configuración de un proyecto con dos PLC's y direcciones IP diferentes

Una vez adicionadas y configuradas las direcciones IP de los PLC's, se procede a la configuración de la comunicación Profinet por mensajería S7.

- ① En dispositivos y redes, unir los puertos de red de los PLC's seleccionando el puerto Profinet de cualquiera de los dos PLC's.
- ② Arrastrarlo hasta unir con el puerto Profinet del otro PLC y soltar.
- ③ Seleccionar, en la barra superior, la opción *Comunicación* y en el menú desplegable, *S7 conexión*.
- ④ Volver a unir los puertos Profinet de los PLC's, tal como en el paso 1 (ver Fig. 5.34).



**Figura 5.34:** Configuración de la red Profinet

Una vez realizado el procedimiento, en la sección *Propiedades* se pueden visualizar las características de la conexión S7, tal como muestra la Fig. 5.35.

Para este ejemplo, se va a configurar una comunicación desde el PLC\_1 local, el cual va a leer el puerto de entrada y va a escribir el valor leído en el puerto de salida, pero para esto va a utilizar los registros MB0 y MB1, el programa del PLC\_2 remoto (ver Fig. 5.36).

Previamente, para la programación del PLC\_1 Local se va a configurar el reloj de memoria en el registro MB0, para utilizar una de sus señales como indicación de activación de los comandos GET y PUT. Esto definirá el tiempo de actualización de los datos.

1. Seleccionar el PLC\_1 en el árbol del proyecto.
2. Escoger *Configuración del dispositivo*.
3. En *Propiedades*, escoger *General*.

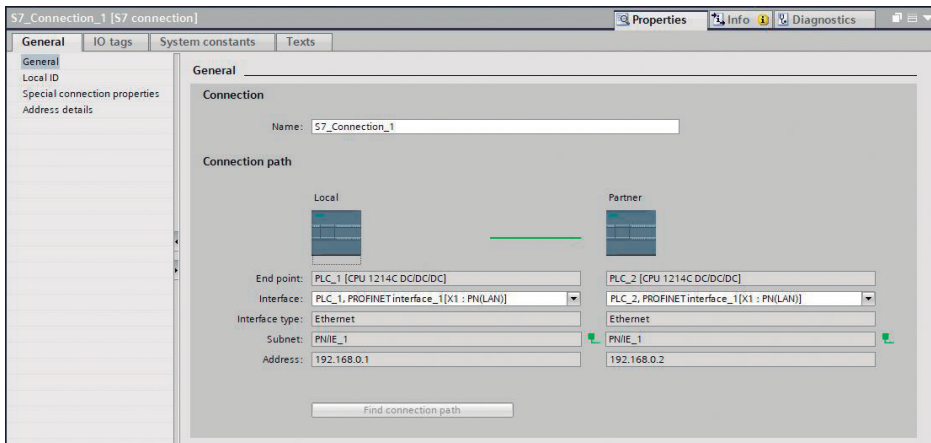


Figura 5.35: Características de la comunicación Profinet S7



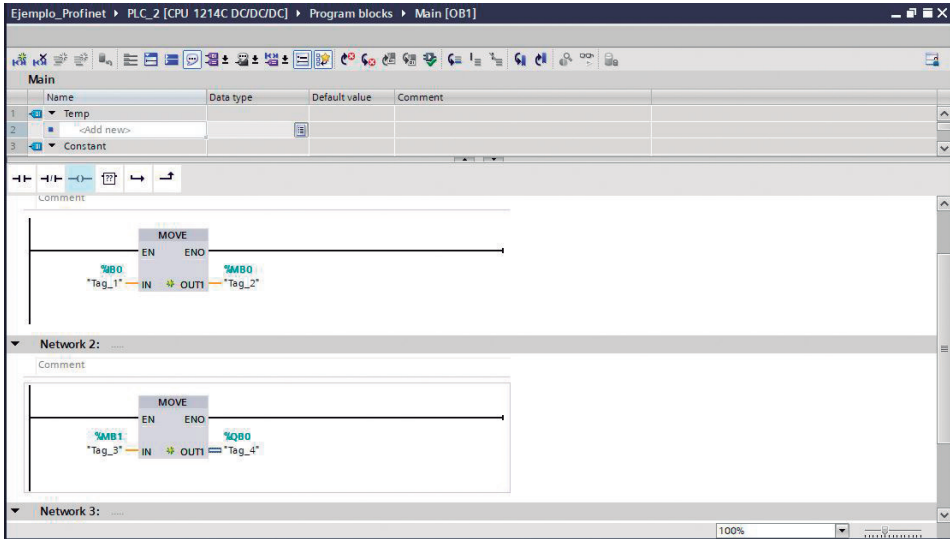


Figura 5.36: Programa del PLC\_2 remoto

4. Dar clic en la opción *Relojes de memoria y sistema*.
5. Habilitar el uso del byte del reloj de memoria.
6. Asignar el registro de memoria en el que se van a activar, en este caso, MB0 (ver Fig. 5.37).

A continuación, programar el PLC\_1 Local, donde se van a colocar las instrucciones GET para la lectura de las entradas.

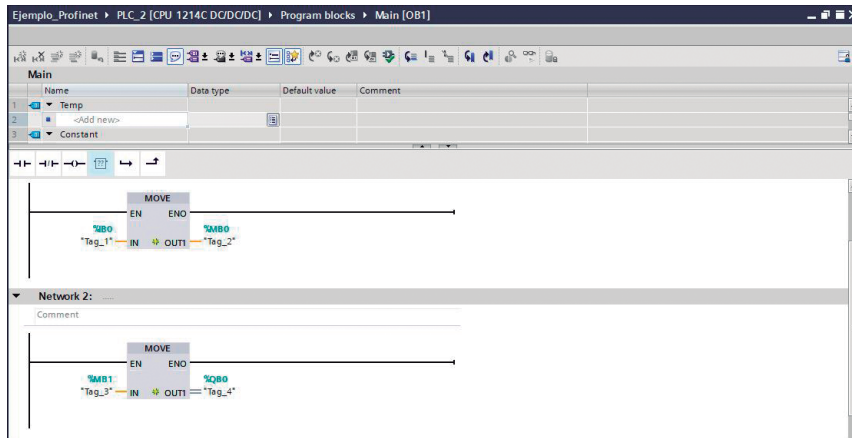
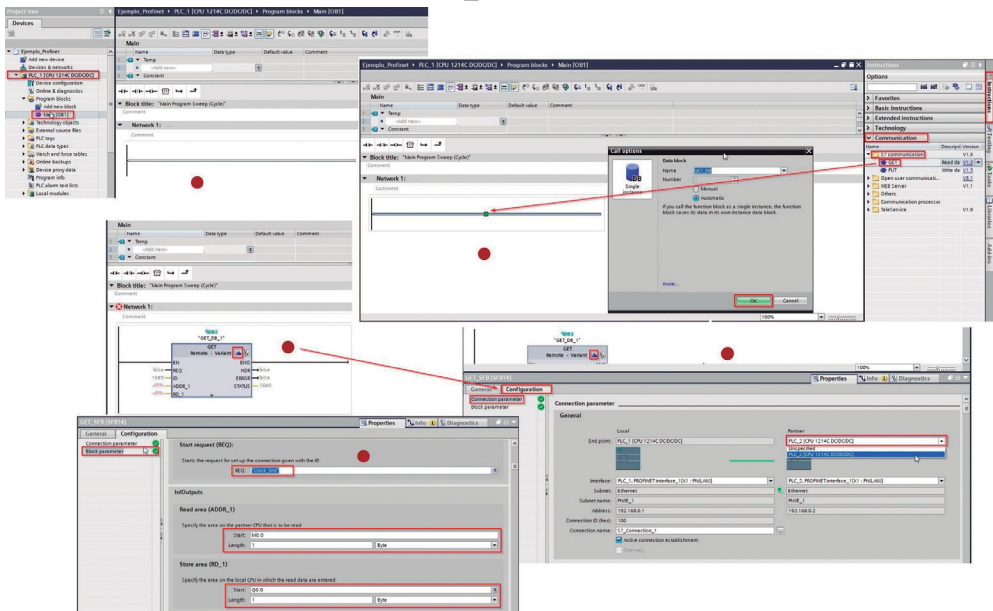


Figura 5.37: Configuración del reloj de memoria del PLC\_1

- ① Se escoge el PLC\_1 del árbol del proyecto y se selecciona el Main [OB1].
- ② En la sección *Instrucciones*, se selecciona la etiqueta *Comunicaciones*, la carpeta *Comunicaciones S7*; se arrastra al Ladder el comando GET y se acepta la adición del comando.
- ③ Una vez insertada la instrucción en el Ladder, se debe hacer clic en el ícono de *Configuración de la instrucción*, lo que abre en *Propiedades* su menú.
- ④ En la sección *Configuración*, escoger *Parámetros de conexión* y seleccionar el PLC del cual se van a obtener los datos; en este caso, PLC\_2.
- ⑤ En la etiqueta *Parámetros del bloque*, asignar las direcciones de activación, origen y destino de los datos. En este caso, la señal de activación del reloj de memoria, la señal de 5 Hz de frecuencia asignada en la dirección M0.1. Esto quiere decir que el dato se actualizará 5 veces por cada segundo, lo que puede ser modificado dependiendo del requerimiento, el origen de los datos del PLC\_2, la marca M0.0 y el destino la salida Q0.0 del PLC\_1; en ambos casos se selecciona una longitud de 1 byte, lo que puede ser modificado dependiendo del tamaño de datos a transmitir requerido. Ver Fig. 5.38.

Posteriormente, se van a colocar las instrucciones PUT para la escritura en las salidas del PLC\_2.



**Figura 5.38:** Programa del PLC\_1, configuración del comando GET

- ① Se escoge el PLC\_1 del árbol del proyecto y se selecciona el Main [OB1]; en la parte superior del Ladder, se selecciona el comando con signos de interrogación.
- ② Se arrastra hacia el Ladder y se suelta en la línea a programar.
- ③ Se da doble clic sobre el bloque adicionado, se digita PUT y se selecciona el comando PUT que aparece.
- ④ Se acepta la adición del comando.
- ⑤ Se debe hacer clic en el ícono de *Configuración de la instrucción*, lo que abre en *Propiedades* su menú.
- ⑥ En la etiqueta *Parámetros del bloque*, asignar las direcciones de activación, origen y destino de los datos; en este caso, la señal de activación del reloj de memoria, la señal de 5 Hz de frecuencia asignada en la dirección M0.1. Esto quiere decir que el dato se actualizará 5 veces por cada segundo, lo que puede ser modificado dependiendo del requerimiento, el origen de los datos del

PLC\_2 M1.0 y el destino la salida Q0.0 del PLC\_1. En ambos casos, se selecciona una longitud de 1 byte, lo que puede ser modificado dependiendo del tamaño de datos a transmitir requerido (ver Fig. 5.39).

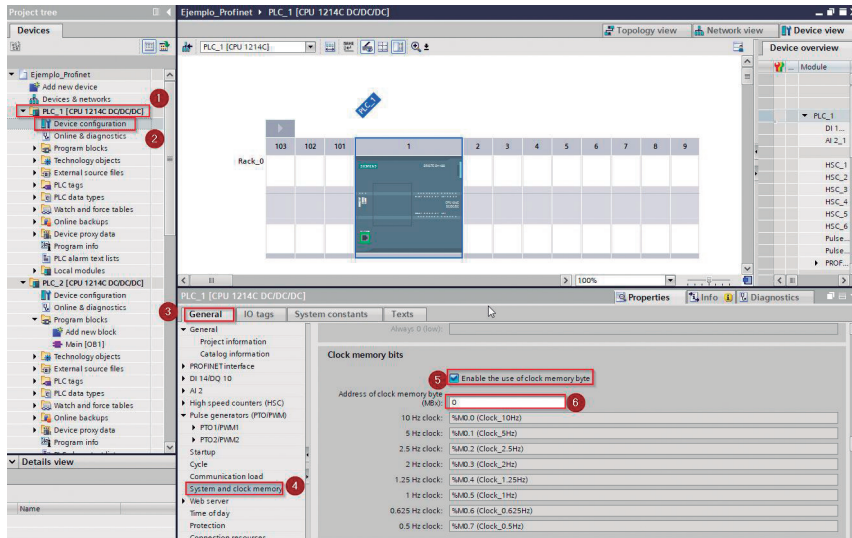


Figura 5.39: Programa del PLC\_1, configuración del comando PUT

Una vez creado el programa, se compila y carga en ambos PLC los programas y se procede a comprobar el funcionamiento del sistema; note que —tal como se mencionó en la introducción— la comunicación se da sin la necesidad de ningún cambio en la programación del PLC\_2. Esta sí puede ser configurada y simulada a través de la herramienta de simulación PLC Advanced de Siemens y compatible con TIA Portal.

## 5.4. Ejercicios propuestos

1. Se requiere establecer una red de comunicación que permita conectar un panel de control a un PLC y un variador de

frecuencia Sinamics V20, de tal manera que se pueda controlar desde el HMI los parámetros de funcionamiento básico del motor, —como sentido de giro, velocidad, encendido— y cuente con indicadores de estado. Desarrolle la topología y arquitectura de red, considerando que la comunicación entre el PLC y el HMI se va a realizar a través de Industrial Ethernet y entre el PLC y el variador por medio de Modbus RS-485.

2. Un sistema de control de encendido de 16 lámparas divididas en dos grupos de 8 cuenta con un panel de operador, que tiene:

- Una ventana inicial donde se debe ingresar un usuario y su clave.
- Una ventana de operador que tiene un botón de *Prendido* que enciende las lámparas configuradas y un pulsador de *Apagado* que las desactivará.
- Una ventana de configuración que, por seguridad, apaga las lámparas cuando es abierta y permite su configuración. Tiene un slider que configura el tiempo de encendido de 1 a 20 segundos, un interruptor de parpadeo que activa la opción de parpadeo de las lámparas a frecuencia de 0.5 Hz durante el tiempo configurado, un pulsador que selecciona el grupo de lámparas a configurar (G1 o G2) y un botón de *Restablecer* que vuelve al grupo de lámparas seleccionado a los parámetros de fábrica (5 segundos de encendido sin parpadeo).

La comunicación entre el panel de operador y el PLC1 es por Ethernet y el control es realizado por dos PLC's (PLC1 y PLC2) unidos a través de un bus Profibus DP (PLC1 master y PLC2 esclavo). Cada uno tiene sus propias 8 lámparas que funcionan según la configuración de su respectivo grupo (G1 o G2); adicionalmente, cada PLC tiene un botón de *Start* que prende todas las lámparas acorde a la configuración establecida de cada grupo.

3. Un sistema de mezcla con 3 bombas que controlan el ingreso de líquidos A, B y C cuenta con un panel HMI que tiene una ventana de configuración con un slider por el cual se configura el intervalo de tiempo de encendido de cada bomba en segundos desde 0 a 15. Un interruptor general permite configurar si el ingreso es simultáneo o secuencial (primero A, luego B y finalmente C), un selector que permite seleccionar la bomba del líquido a configurar, un pulsador de *Carga* que configura el tiempo de cada bomba, un botón de *Restablecer* que vuelve al sistema a su configuración inicial (A, B y C 10 segundos), una ventana de operador que tiene un botón de *Prendido* que enciende las bombas según el tiempo y el tipo de ingreso configurado, un pulsador de *Apagado* que apaga las bombas de llenado y enciende la bomba de vaciado hasta detectar si el tanque está vacío a través de un sensor. Una vez encendido el sistema, ingresan los líquidos. Luego de eso, activa un sistema de mezcla por un tiempo igual a la suma de los tiempos cargados dividido para 3, para luego activar una bomba que vacía el tanque. Una vez detectado el tanque vacío, se vuelve a iniciar el proceso.

Por cuestión de distancias entre los equipos, el sistema requiere de dos PLC para el control de las bombas: uno para A y C, y otro para B, el sistema de mezcla y la bomba de vaciado. Ambos PLCs deben estar comunicados entre sí a través de mensajería Profinet S7 de tal forma que permitan el correcto funcionamiento del sistema.

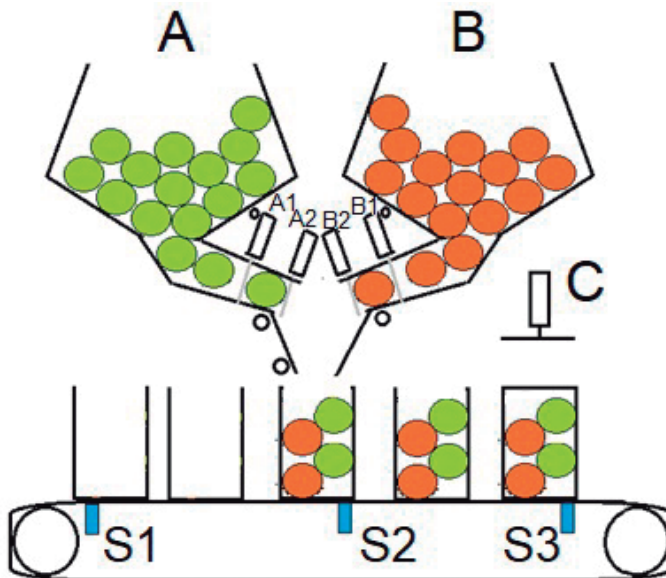
4. Diseñar un sistema de control de encendido de un motor DC desde un panel de operador HMI, que cuenta con una ventana de configuración que tiene un selector de velocidad baja, media o alta, un interruptor para el sentido de giro y dos pulsadores —uno de *Carga* que carga los parámetros configurados y otro de *Restablecer* que vuelve a los parámetros anteriores—, una ventana de operador que tiene un botón de *Prendido* que enciende el motor, un pulsador de *Apagado*, indicadores de la configuración

y estado a partir de animación del motor y un contador de tiempo en segundos de encendido total del motor desde su última configuración; este se encera cada que se cambia la configuración y activa una alarma cada que el motor ha completado 30 segundos de funcionamiento acumulado a través de una ventana emergente y una ventana de real trend que despliega el tiempo de encendido del motor. El control del motor lo realiza el PLC S7-1200, el cual puede también ser controlado por otro PLC S7-1200 que tiene tres pulsadores: uno de *Inicio* que arranca el motor, otro de *Sentido* de giro que cambia el sentido y otro de *Parada* que detiene el motor. Ambos PLC's están comunicados a través de Profibus DP.

5. Resolver el ejercicio propuesto 5 del capítulo 4 considerando que el motor M es controlado por un variador Sinamics V20 y que está conectado al PLC a través de un bus Modbus RS-485 (ver Fig. 4.30).
6. Se trata de automatizar un sistema para llenar botes con cuatro pelotas de dos colores (pelotas de color A y pelotas de color B) de forma alternada (A B A B). Para el ingreso de botes, el sistema cuenta con una banda transportadora. La ubicación del bote dentro de la banda se detecta a través de tres sensores capacitivos: el primero, al momento de ingresar al sistema que activa la banda; el segundo activa el proceso de llenado, detiene la banda hasta completarlo o hasta que se retire el bote en la banda; el tercero también detiene la banda y activa el cilindro de sellado. Al activarse el fin de carrera, espera 4 segundos, regresa y activa nuevamente la banda.

El proceso de llenado se inicia haciendo tres movimientos de avance y retroceso de los cilindros dispensadores correspondientes. Por ejemplo, para las pelotas de color A, se activa el cilindro A1, que abre la cámara hasta detectar el ingreso de una pelota para cerrarla. Posteriormente, se activa el cilindro A2 provocando la caída de la pelota al bote y luego lo cierra. Se cuentan con sensores de apertura de A1, de pelota en la cámara y de pelota

en el bote (común para los dos colores de pelota). Un contador registrará las pelotas que han ingresado al bote (ver Fig. 5.40). El sistema genera una alarma que no detiene el proceso: cuando se abre la compuerta y no se detecta el ingreso de una pelota en la cámara correspondiente por al menos 5 segundos, aparece una ventana emergente y se presenta la opción de abrir nuevamente la compuerta de la cámara correspondiente, la cual se cerrará automáticamente al detectar el ingreso de la pelota.



**Figura 5.40:** Sistema de llenado de botes

El sistema genera una alarma que detiene el proceso: si no se completan las cuatro pelotas en el bote en un tiempo máximo de 20 segundos, el proceso genera una alarma y se detiene de forma automática y se muestra una alarma piloto intermitente. El operario solucionará el inconveniente y retirará el bote, momento en el que el piloto intermitente dejará de lucir, espera 3 segundos y volverá a iniciarse el proceso desde el principio.



Para el control del sistema tiene 2 PLC's, el primero es el encargado de controlar la banda y el dispensador 1, y el segundo el dispensador 2 y el sistema de llenado, ambos comunicados a través de un sistema de mensajería Profinet S7.

Los sensores capacitivos de la banda y todos los fines de carrera se representan por entradas a través de pulsadores o switches y todos los actuadores del sistema (salidas) deben ser salidas físicas del PLC respectivo.

El sistema cuenta con un HMI para su supervisión que tiene las siguientes ventanas: de usuario, de alarmas, de supervisión y de mando.

Existen 3 niveles de usuario: alto —que tiene acceso a la todas las ventanas y funciones—, medio —que no tiene acceso a la ventana de monitoreo y al botón de *Reset* de la ventana control— y básico —con acceso a la ventana de mando.

La ventana de supervisión permite ver los datos del proceso, como número total de botes, pelotas de cada modelo utilizadas y número de botes en la banda.

La ventana de mando permite iniciar y detener el proceso y encerar los contadores de pelotas utilizadas, número de botes llenos y número de botes en la banda.

La ventana de alarma indica los problemas en el sistema.

7. Resolver el ejercicio propuesto 3 del capítulo 4 considerando que el panel de control, banda 1 y el cilindro A son manejados por un PLC (maestro) y la banda 2 y el cilindro B es controlado por otro PLC (esclavo). Considerar que ambos PLC's están comunicados por Profibus DP (ver Fig. 4.31).



# Bibliografía



- [ 1 ] E. Córdova, “Manufactura y Automatización”. Disponible en [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0120-56092006000300014](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-56092006000300014), 2006. Fecha de acceso: 2021-02-10.
- [ 2 ] E. García, *Automatización de Procesos Industriales*. España: Universidad Politécnica de Valencia.
- [ 3 ] J. Álvarez, *TIA Portal. Aplicaciones de PLC*. Colombia: Instituto Tecnológico de Medellín.
- [ 4 ] Project Management Institute, “Guía de los Fundamentos para la dirección de proyectos. (Guaiac PMBOOK, Sexta Edición)” p. 18, tech. rep., Project Management Institute Inc, 2017.
- [ 5 ] W. S. Levine, “Control system fundamentals, 1ra ed.”, Boca Raton, FL, USA: CRC Press, 2000.
- [ 6 ] J. Gausemeier y S. Moehringer, “Vdi 2206- a new guideline for the design of mechatronic systems,” *IFAC Proceedings Volumes*, vol. 35, N.º 2, pp. 785-790, 2002. 2nd IFAC Conference on Mechatronic Systems, Berkeley, CA, USA, 9-11 December.
- [ 7 ] I. Graessler y J. Hentze, “Transformations in a product development to enable globally distributed self-organizing production systems”, en CIRP Design Conference 2019.
- [ 8 ] M. Stania y R. Stetter en *Mechatronics engineering on the example of a Multipurpose Mobile Robot*, 2019.
- [ 9 ] L. Echeverría, “Trabajando con PLCs”. Disponible en: <https://www.lawebdelprogramador.com/pdf/15584-Trabajando-con-PLCs.html>, 2015. Fecha de acceso: 2021-03-19.
- [ 10 ] W. Bolton, *Programmable Logic Controllers*. Elsevier Newnes, cuarta ed., 2006.
- [ 11 ] D. Rojas, “Programación de autómatas industriales”. Disponible en: <https://davidrojasplc.wordpress.com/category/8-programacion-de-plc/>. 2009. Fecha de acceso: 2021-03-15.
- [ 12 ] SIEMENS, “SIMATIC S7 Controlador programable S7-1200 V4.2.”. Disponible en [https://media.automation24.com/manual/es/91696622\\_s71200\\_system\\_manual\\_es-ES\\_es-ES.pdf](https://media.automation24.com/manual/es/91696622_s71200_system_manual_es-ES_es-ES.pdf). Fecha de acceso: 2020-09-01.

- [ 13 ] SIEMENS, “SIMATIC S7 TIA Portal STEP 7 Basic V10.5.” Disponible en [https://cache.industry.siemens.com/dl/files/542/40263542/att\\_829830/v1/GS\\_STEP7Bas105esES.pdf](https://cache.industry.siemens.com/dl/files/542/40263542/att_829830/v1/GS_STEP7Bas105esES.pdf). Fecha de acceso: 2020-11-15.
- [ 14 ] S. S7-100, “Data sheet SIMATIC S7-1200, CPU1215C 6Es7215-1AG40-0XB0”. Disponible en: <https://media.automation24.com/datasheet/en/6ES72151AG400XB0.pdf>. Fecha de acceso: 2020-11-20.
- [ 15 ] M. R. Anwar, O. Anwar, S. F. Shamim y A. A. Zahid, “Human machine interface using opc (ole for process control),” en *Student Conference On Engineering, Sciences and Technology*, pp. 35–40, 2004.
- [ 16 ] R. Jiménez y F. Rodolfo, *Las interfaces humano-máquina (HMI) y su importancia en el control de procesos industriales*.
- [ 17 ] DATA 2000 SISTEMAS Soluciones Integrales, “HMI Básicas”. Disponible en: <https://www.data2000sistemas.com/hmi-básicas/>. Fecha de acceso: 2021-05-14.
- [ 18 ] REALPARS, “What is an HMI? Learn some of the basics of an Human Machine Interface”. Disponible en: <https://realpars.com/what-is-hmi/>. Fecha de acceso: 2020-09-02.
- [ 19 ] ACE, “HMI Productos”. Disponible en: <https://www.acesa.es/cs/p1/hmi>. Fecha de acceso: 2020-08-26.
- [ 20 ] IEC, “Norma UNE EN 61131-1”. Disponible en: <http://isa.uniovi.es/docencia/IngdeAutom/transparencias/Pres%20IEC%2061131.pdf>, 2013. Fecha de acceso: 2021-01-18.
- [ 21 ] J. Barbado, *Automatismos industriales*. México: Alfaometa, Marcombo.
- [ 22 ] P. Baracos, *Tutorial and Reference Guide to the Grafset Language*. St. Laurent, Que: Famic Automation, Inc.
- [ 23 ] IEC, “Norma IEC 848”. Disponible en: [http://ftp.demec.ufpr.br/disciplinas/TM265/GRAFSET\\_utfpr\\_iec\\_848.pdf](http://ftp.demec.ufpr.br/disciplinas/TM265/GRAFSET_utfpr_iec_848.pdf), 1988. Fecha de acceso: 2020-09-18.
- [ 24 ] J. L. R. Josep Balcells, *Autómatas programables*. México: Alfaometa, Marcombo.
- [ 25 ] R. Sanchis, J. Romero, y C. Arino, *Automatización Industrial*. Universitat Jaume I, 2010.
- [ 26 ] J. Romero, J. A. Lorite, y S. Montoro, *Automatización, Problemas resueltos con autómatas programables*. Editorial Paraningo, 1994.
- [ 27 ] C. Borghello, “Ciberseguridad en la pirámide de automatización industrial”. Disponible en: <https://blog.segu-info.com.ar/2017/06/ciberseguridad-en-la-piramide-de.html>, 2017. Fecha de acceso: 2020-03-02.

- [ 28 ] OCPEcuador, “Nivel de supervisión”. Disponible en: <https://ocpecuador.com/es/sala-de-prensa/documentación/fotos/cuarto-de-control-quito-0>, 2013. Fecha de acceso: 2020-03-05.
- [ 29 ] L. Muniz, ERP: *Guía práctica para la selección e implantación*. Gestión 2000, 2004.
- [ 30 ] C. Moller, “Erp ii: a conceptual framework for next generation enterprise systems?”, *Journal of Enterprise Information Management*, 2005.
- [ 31 ] Siemens, *PROFINET descripción del sistema, manual de sistema*. Siemens AG, 2006.
- [ 32 ] L. Thompson y T. Shaw, *Industrial Data Communications*. International Society of Automation ISA, 2015.
- [ 33 ] P. González, “Cuáles son los elementos de la comunicación”, 2018.
- [ 34 ] S. Kumar Sen, *Fieldbus and Networking in Process Automation*. CRC Press, 2014.
- [ 35 ] R. Guerrero, Vicente; Yuste y L. Martínez, *Comunicaciones Industriales*. Marcobo, 2010.
- [ 36 ] D. Caro, *Automation Network Selection: A Reference Manual*. International Society of Automation ISA, 2009.
- [ 37 ] N. SMAR, “¿Qué es Profibus?”. Disponible en: <https://www.smar.com/espanol/profibus>, 2020. Fecha de acceso: 2021-02-20.
- [ 38 ] P. N. América, “Profinet: rendimiento en tiempo real” Disponible en: <https://us.profinet.com/tecnologia/profinet-es/>, 2020. Fecha de acceso: 2021-02-20.
- [ 39 ] E. Flores, “Implementación de protocolo Modbus RTU en la comunicación de un PAC y un variador de velocidad para el control automático de motores eléctrico”. Disponible en: <https://www.uaeh.edu.mx/scige/boletin/tizayuca/n5/p7.html>, 2017. Fecha de acceso: 2021-03-07.
- [ 40 ] H. Berger, *Automating with SIMATIC S7-1200*. Siemens Aktiengesellschaft, Berlín y Múnich, segunda edición, 2013.
- [ 41 ] Siemens, “Sinamics v: Speed control of a v20 with s7-1200 (TIA Portal) via uss@ protocol/modbus RTU with HMI”. Disponible en: [https://support.industry.siemens.com/cs/document/63696870/sinamics-v%3A-speed-control-of-a-v20-with-s7-1200-\(tia-portal\)-via-uss%C2%AE-protocol-modbus-rtu-with-hmi?dti=0&dl=en&lc=es-DO](https://support.industry.siemens.com/cs/document/63696870/sinamics-v%3A-speed-control-of-a-v20-with-s7-1200-(tia-portal)-via-uss%C2%AE-protocol-modbus-rtu-with-hmi?dti=0&dl=en&lc=es-DO). 2017.

