

Plan de contingencia Pedagógica Para 6to Año  
Incluye Bibliografía y Ejercicios de Repaso



Instrucciones: Leer el material y luego tratar de resolver los ejercicios. Ante cualquier duda consultar al siguiente mail:  
[hugowojczys@yahoo.com.ar](mailto:hugowojczys@yahoo.com.ar)

# Programación del PLC

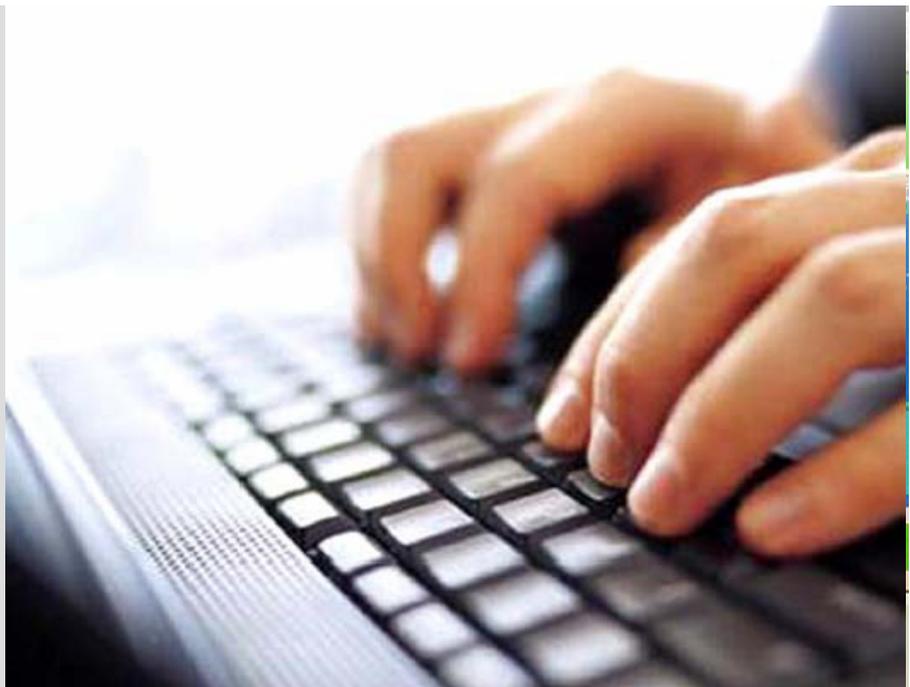
## TEMAS DEL CAPÍTULO 6

---

<b>6.1 Lenguajes de Programación</b>	<b>66</b>
<b>6.2 Designación de Entradas y Salidas</b>	
<b>6.3 Caso de Estudio: Control de los movimientos de subida y bajada de un ascensor</b>	

El aumento de la complejidad en la programación de los autómatas programables requiere más que nunca de la estandarización de la misma.

En este capítulo se desarrollarán los lenguajes que han sido definidos.



## 6.1 Lenguajes de programación

Bajo la dirección del IEC el estándar IEC 1131-3 (IEC 65) ha sido definido para la programación del PLC. Alcanzó el estado de Estándar Internacional en Agosto de 1992. Los lenguajes gráficos y textuales definidos en el estándar son la base para la programación de PLCs.

Con la idea de hacer el estándar adecuado para un gran abanico de aplicaciones, cinco lenguajes han sido definidos en total:

- 1 Gráfico secuencial de funciones (grafcet)
- 2 Lista de instrucciones (LDI)
- 3 Texto estructurado
- 4 Diagrama de flujo
- 5 Diagrama de contactos (Ladder)

### 1 Gráfico secuencial de funciones (grafcet)

El gráfico secuencial de funciones (STL, SFC o Grafcet) es un lenguaje gráfico que **proporciona una representación en forma de diagrama de las secuencias del programa**. Soporta selecciones alternativas de secuencia y secuencias paralelas. Los elementos básicos son pasos y transiciones. Los pasos consisten en partes de programa que son inhibidas hasta que una condición especificada por las transiciones es conocida. Como consecuencia de que las aplicaciones industriales funcionan en forma de pasos, el SFC es la forma lógica de especificar y programar al más alto nivel un PLC.

### 2 Lista de instrucciones (LDI)

La lista de instrucciones (IL) es un lenguaje de bajo nivel, similar al lenguaje ensamblador. Con IL sólo una operación es permitida por línea (ej. Almacenar -store- cargar un valor en un registro). Este lenguaje es adecuado para pequeñas aplicaciones y para optimizar partes de una aplicación.

### 3 Texto estructurado

El texto estructurado (structured text o ST) es un **lenguaje de alto nivel, estructurado por bloques, que posee una sintaxis parecida al PASCAL**.

El ST puede ser empleado para realizar rápidamente sentencias complejas que manejen variables con un amplio rango tipos de datos, incluyendo valores analógicos y digitales. También especifica tipos de datos para el manejo de horas, fechas y temporizaciones, algo importante en procesos industriales. El lenguaje posee soporte para bucles repetitivos como: REPEAT UNTIL, ejecuciones condicionales empleando sentencias IF-THEN-ELSE-FOR-NEXT y funciones como SQRT (raíz cuadrada) y SIN (seno).

#### 4 Diagrama de flujo

El diagrama de funciones (function block diagram o FBD) es un lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre sí de forma análoga al esquema de un circuito.

FBD es adecuado para aplicaciones que involucren el flujo de información o datos entre componentes de control.

#### 5 Diagrama de contactos

El diagrama de contactos (ladder diagram LD) es un lenguaje que utiliza un conjunto estandarizado de símbolos de programación. En el estándar IEC los símbolos han sido racionalizados (se ha reducido su número).

### Organización de tareas

El estándar también define una nueva arquitectura para la organización e interacción de tareas con PLCs. **Una tarea controla la ejecución de un programa ejecutándolo periódicamente o en respuesta a un evento específico.** Para optimizar los recursos del controlador, una aplicación puede ser fragmentada en un número de pequeños programas concretos.

Cada programa tiene el control de una tarea que se ejecuta a la velocidad que requiera la E/S asociada.

### Bloques de Funciones

Los bloques de funciones (FBs) son bloques estándar que ejecutan algoritmos como reguladores PID.

El estándar IEC asegura que los FBs son definidos empleando una metodología estándar. Hay controles empleando parámetros externos, mientras que los algoritmos internos permanecen ocultos empleando Programación Orientada a Objetos.

#### ACTIVIDAD 10.

Marque la opción correcta.



¿Por qué se definieron cinco lenguajes?

Para tener un estándar para distintas aplicaciones.

Para utilizar el que mejor comprenda el programador.

¿De qué depende la velocidad de ejecución de tareas?

De los requerimientos de las entradas/salidas.

De la eficiencia en la programación.

## 6.2 Designación de Entradas y Salidas

### Entradas y Salidas

#### EJEMPLO



Las siguientes son designaciones usuales de entradas y salidas:

X: para entrada Mitsubishi

I: para entrada Allen Bradley o Klockner

Y: para salida Mitsubishi

Q: para salida Klockner

O: para salida Allen Bradley

La línea Siemens identifica a las entradas como E E32.0, E32.1, etc. y las salidas como A siendo posible A 32.0, A 32.1, etc.

- Salvo excepciones y ampliaciones, los autómatas presentan 8 entradas normales (X, I, E) de un 1bit. Algunos con entradas especiales de 1bit, tienen la peculiaridad de funcionar como entradas digitales o como entrada de alarma y entrada rápida. La denominación depende de la marca del PLC, por lo tanto, se debe verificar en el manual del fabricante.
- Hay 6 salidas, típicamente empleando las letras (Y, Q, O, A), de 1bit cada una.

### Marcas de memoria

También son denominadas como variables de memoria. Son de propósito general, es decir, podemos emplearlas en lo que deseemos. Se distinguen dos tipos de marcas de memoria:

- **Remanentes:** Estas marcas permanecerán en memoria aunque apaguemos el autómata. Hay diferentes denominaciones de memoria, según el fabricante, por lo tanto se debe verificar en el respectivo manual. En algunos casos son configurables por el usuario.
- **No Remanentes:** Estas marcas de memoria se borrarán en cuanto apaguemos el autómata.

Usualmente a las memorias se las designa con la letra M. Se organizan en forma básica como bit M0, M1, ... , M15 etc. en Melsec, Siemens y Klockner Moeller. Otra designación es B, es decir, B0, B1, ... en Allen Bradley y Telemacanique, etc.

## Registros y Acumuladores

Todas las operaciones que hagamos con las entradas y las salidas se deben efectuar en algún sitio. Para ello, se definen:

- **Registro de estado:** Su tamaño es de 1bit. Aquí es donde efectuaremos las instrucciones combinacionales, la carga de entradas y la asignación de salidas a nivel de bit.
- **Acumuladores (D, R, AKKU1 y AKKU2):** Sus tamaños son de 16 y 32 bits cada uno. Cada vez que carguemos un dato en los acumuladores se seguirá la siguiente secuencia:

Contenido de D2 → se pierde el contenido  
Contenido de D1 → D2  
DATO → D1

A su vez, cuando realicemos una operación entre registros (como suma o resta) el resultado se introducirá en el D1, perdiéndose el valor antes allí contenido.

## Temporizadores y Contadores

Varían en función de marcas y modelos, pero los más usados suelen incorporar **32 temporizadores:** T0, ... , T31 **32 contadores:** C0, ... , C31 para Melsec y Allen Bradley. En Siemens se designan como T y Z.

Existen contadores que no se borran al desconectar el autómatas (son remanentes), dichos contadores también deben verificarse en los respectivos manuales. Para consultar el estado de cada uno de ellos podremos usarlos como si fueran entradas (mediante operaciones combinacionales) o introduciendo su valor en los registros.

## Constantes

A la hora de cargar datos en acumuladores, temporizadores, registros, etc. hay varias posibilidades en la forma de introducir el dato:

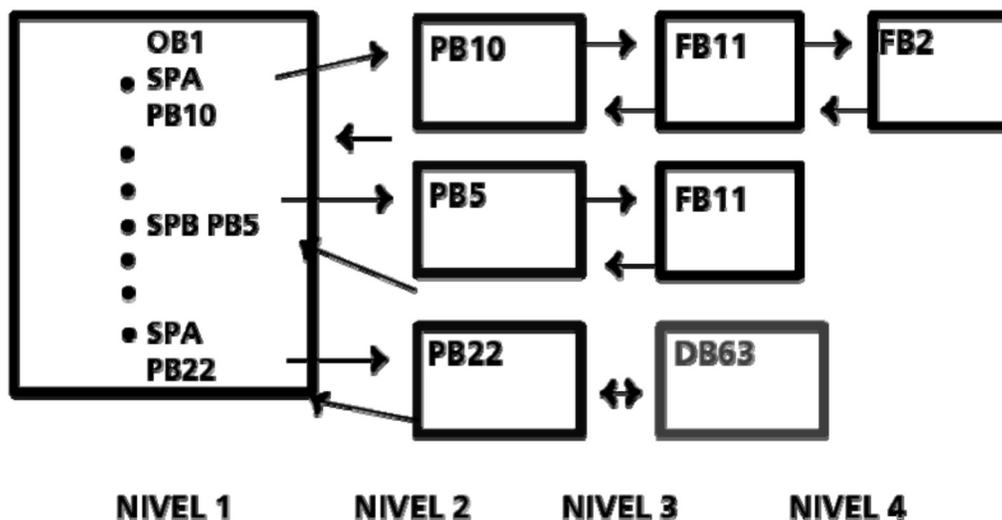
- KB: 8 bits (0 a 255 en decimal)
- KW 16 bit
- Como número decimal
- Como número hexadecimal

La modalidad de trabajar con constantes también depende del fabricante. Existen diversas prácticas para ello. En general, todos los casos admiten trabajar con valores de 8 bit (0--255), 16 bit (0-64535) y 32 bit.

## Estructura del programa

Vamos a tener dos opciones para escribir el programa:

- **Lineal:** Se emplea un único módulo de programa (principal). Este módulo se procesa cíclicamente, es decir, tras la última instrucción se volverá a ejecutar la primera. Si la tarea a controlar es simple, ésta es la mejor forma.
- **Estructurada:** Para el caso de tareas complejas es más conveniente dividir el programa en módulos (sub programas). De esta forma logramos un programa más claro, con la posibilidad de poder llamar a un módulo desde distintas partes del programa (lo que evita repetir el código).



En la programación estructurada se comienza y termina en el módulo principal (llamados OB1 en Siemens) desde el cual saltaremos y retornaremos a los módulos que nos interesen. Se podrá saltar desde un módulo a otro (anidado), siempre que no superemos determinada cantidad de niveles de salto que permita como máximo un autómata.

Otras limitaciones son:

- El salto de un módulo a otro debe ser siempre hacia adelante (ej. se podrá saltar de PB1 a PB2 Siemens, pero no a la inversa).
- No se pueden dar dos saltos a un mismo módulo desde el módulo actual (ej. no se podrá saltar dos veces a PB3 desde PB2, pero sí se puede saltar a PB3 desde distintos módulos).
- Tanto en la programación lineal como en la estructurada, los módulos terminan con la instrucción BE en Siemens o con END.
- La memoria de un autómata generalmente está limitada a 2K bytes como mínimo. Cada instrucción ocupa generalmente 2 bytes, por los que se dispone 1000 líneas de programa aproximadamente.

## Tipos de módulos



Generalmente existen cuatro tipos de módulos en cualquier autómatas programable Siemens:

- **Módulos Fuente de Poder:** este módulo incluye fuente de alimentación para todo el programador y sus módulos.
- **Módulos de Programa (CPU):** son los que incluyen el programa de usuario dividido, normalmente, según aspectos funcionales o tecnológicos.
- **Módulos de entradas:** son módulos de características especiales según tipo de entrada digital, analógica, lector de código barras, etc.
- **Módulos de salidas:** al igual que el anterior tienen características especiales según tipo de salida.

Las características de estos módulos son diferentes por su función en disposición de ubicación física y de hardware propio.

La línea Siemens denomina como módulos:

**1 Módulos de organización (OB):** son los que gestionan el programa de usuario. Numerados OB1, OB3, OB13 Y OB22.  
 OB1: Es el módulo del programa principal;  
 OB3: Es el que contiene el programa controlado por alarma;  
 OB13: Es el módulo para programas controlados por tiempo;  
 OB22: Es empleado por el sistema operativo.

**2 Módulos de programa (PB):** son los que incluyen el programa de usuario dividido, normalmente según aspectos funcionales o tecnológicos PB0 ... PB63.

**3 Módulos funcionales (FB):** son módulos de programa especiales. Aquí se introducen las partes de programa que aparecen con frecuencia o poseen gran complejidad. Poseen un juego de instrucciones ampliado. FB0 ... FB63.

**4 Módulos de datos (DB):** en ellos se almacenan datos para la ejecución del programa, como valores reales, textos, etc. Adoptan los valores: DB0 ... DB63. Los módulos DB1 y DB2 se emplean para definir las condiciones internas del autómeta, por lo que no están disponibles.

**5** 256 palabras de datos: para emplear un módulo de datos es necesario activarlo previamente (como se verá más adelante).

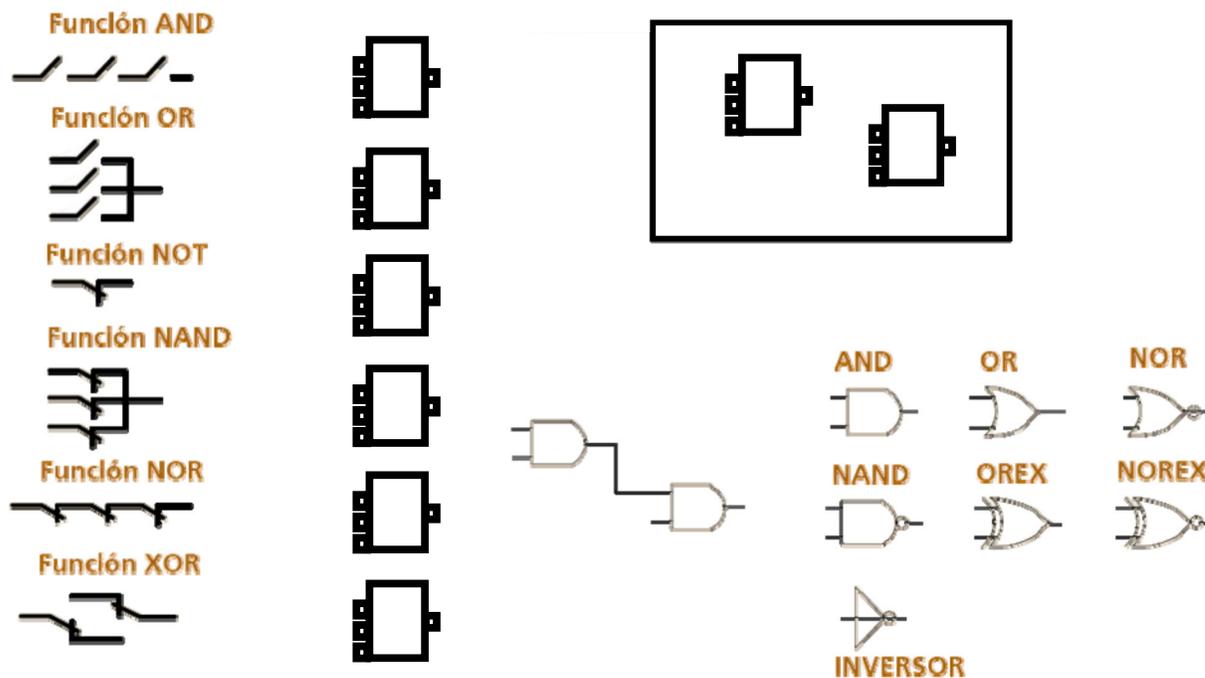
La mayor ventaja que aporta el trabajo con módulos, es la facilidad para variar el proceso que controlan, ya que para ello basta cambiar el programa en el autómeta en la mayoría de los casos. Otra ventaja es que el autómeta también nos permite saber el estado del proceso, incluyendo la adquisición de datos para un posterior estudio.

## Funciones Generales

Las operaciones combinatorias más comunes se realizan con los bloques de funciones básicas, conexión serie, paralelo, negación, etc. Todas las funciones AND, OR, XOR, NAND Y NOR tienen tres entradas y una salida. Si deseamos realizar operaciones con más de tres entradas, se conectan varios bloques en cascada:

### EJEMPLO

#### Ejemplo según normas DIN Y NEMA

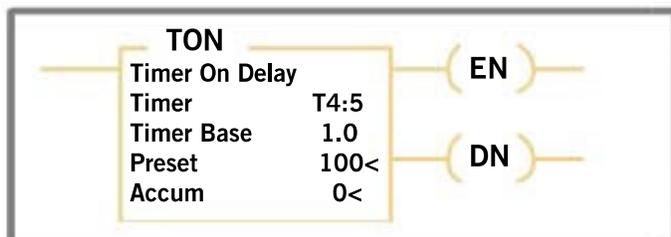


La función inversora NOT, tiene una entrada y una salida. La función OR exclusiva (XOR) posee dos entradas y una salida.

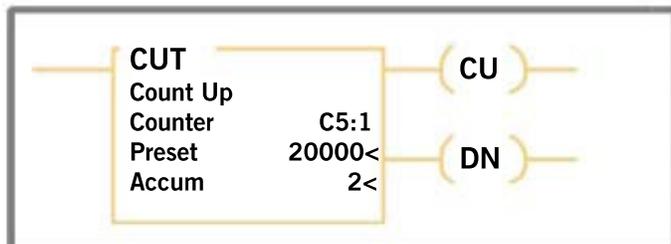
## Funciones Especiales Din Siemens

TRA R A		Temporizador con retardo a la conexión con memoria.	<b>Temporizador con retardo a la conexión.</b> Activa la salida Q una vez que ha transcurrido el tiempo programado.
R CNT DIR PAR		Contador progresivo/regresivo	<b>Temporizador con retardo a la desconexión.</b> Desactiva la salida una vez transcurrido el tiempo programado. El temporizador se pone en marcha en flanco descendente.
R EN RAL PAR	 01	Contador de horas de servicio.	<b>Relé de impulsos.</b> Tienen el mismo funcionamiento que un telerruptor. La salida cambia de estado, de 0 a 1, cada vez que cambia la señal en la entrada Trg.
TRA T	 01	Relé de supresión.	<b>Reloj.</b> Permite controlar los instantes de activación y desactivación de la salida en un día de la semana y a una hora determinada.
FRE PAR	 01	Conmutador de valor de umbral para frecuencia.	<b>Relé de automantenimiento</b> Función biestable R-S. Permite realizar la función paro-marcha típica de los automatismos a contactores. La situación no permitida R=1 S=1 se soluciona dando preferencia a R.
TRA T		Temporizador con retardo a la conexión.	<b>Generador de pulsos.</b> Genera pulsos de reloj a intervalos iguales. Funcionamiento similar a un intermitente.
TRA R T		Temporizador con retardo a la desconexión.	<b>Temporizador</b> De funcionamiento similar al temporizador a la conexión, pero con la característica que no es necesario mantener la señal en Trg.
TRA R (PAR)		Relé de Impulsos (Telerruptor)	<b>Contador progresivo/regresivo.</b> Permite contar y descontar los pulsos aplicados a su entrada CNT.
NO1 NO2 NO3		Reloj Horario	<b>Contador de horas de servicio.</b> Permite medir el tiempo que está activada la entrada En. Esta función solamente se puede utilizar como bloque inicial.
S R (PAR)		Relé de automantenimiento. Biestable R-S.	<b>Relé de supresión</b> Activa la salida hasta que haya transcurrido el tiempo de T. Si éste no ha terminado y Trg se pone a 0 la salida también lo hace. Esta función solamente se puede utilizar como bloque inicial.
EN T		Generador de pulsos de reloj.	<b>Conmutador de valor de umbral para frecuencias.</b> Permite contar los impulsos aplicados a su entrada y dependiendo de éstos conmutar la salida. En el Logo! L con letras 24v, la entrada I12 está preparada para procesos de cómputos rápidos: función solamente se puede como bloque inicial.

### Temporizador con retardo a la conexión Allen Bradley



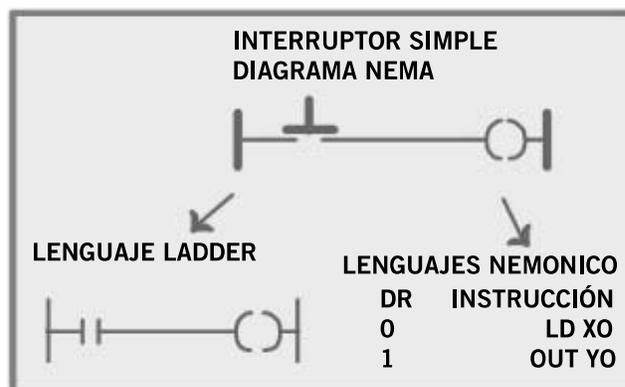
Activa la salida DN (Done) una vez que ha transcurrido el tiempo programado de 100 seg.



Se trata de un contador ascendente. Activa la salida DN una vez que se alcanza la cuenta 20000.

Los lenguajes de programación básicos más comunes son el Ladder y el Nemónico. En estos lenguajes, las instrucciones son equivalentes a los símbolos para contactos usados en los relés (lógica cableada) para el primero y similar a las definiciones del álgebra de Boole (lógica digital).

En un diagrama, se hace a representación típica de una línea, que implemente una función de control, para una salida en lenguajes Ladder y Nemónico como se parecía en la figura.

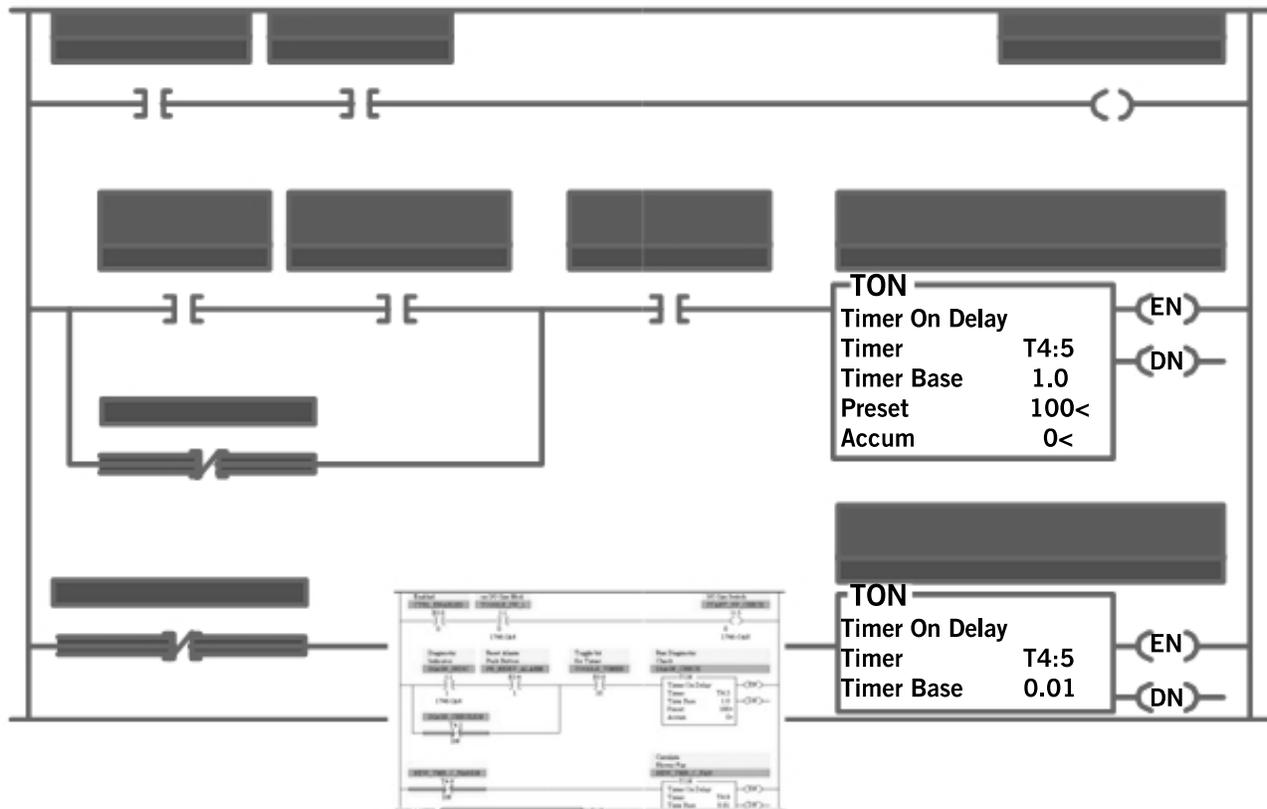


Los símbolos representados son muy parecidos al NEMA eléctrico, aún más son tan simples que emplean el símbolo de auxiliar abierto o cerrado para todo lo que se designe como “entrada” y para todo lo que es “salida” se emplea un círculo incompleto como se indica en la figura anterior.

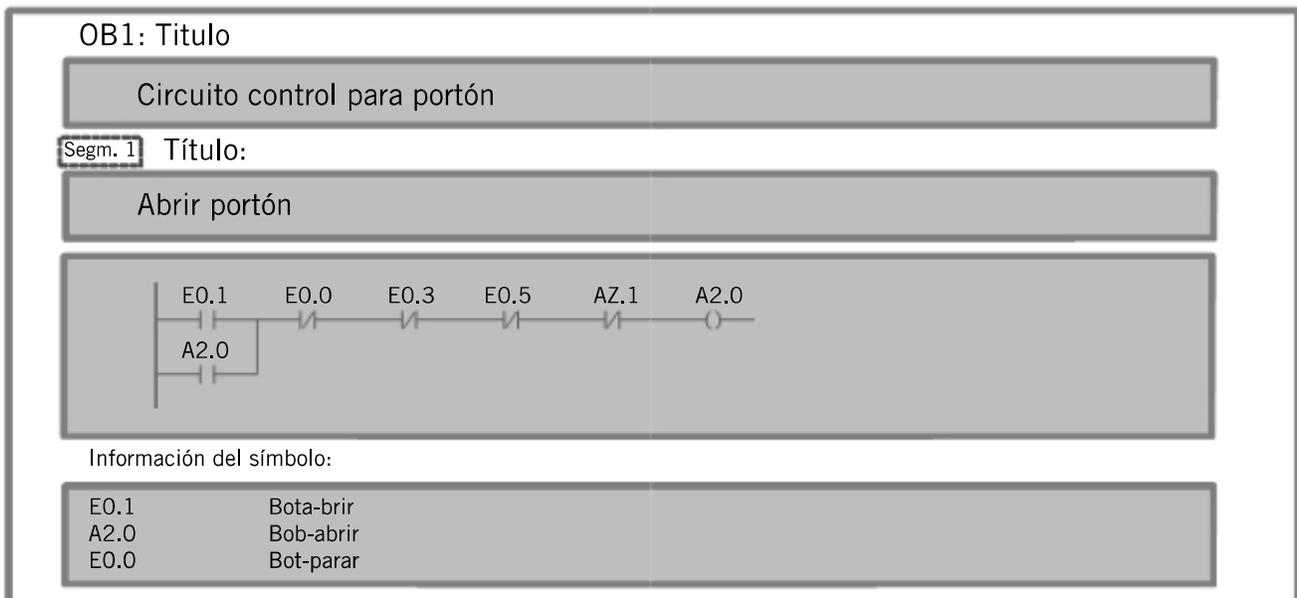
A pesar de la semejanza, hay diferencias:

1. Las salidas pueden ser del tipo interno y externo. Es decir, salida del tipo real (contactor, válvulas, etc.) o del tipo virtual (sólo en memoria).
2. El programa examina los contactos N.A. y los N.C. buscando encontrar un 1 o un 0 lógicos.
3. La salida en una línea de control tradicional se activa si cualquier paso o camino tiene todos sus contactos cerrados.
4. La salida de un control programado se activa si cumple la lógica booleana de la ecuación que ella representa.
5. Una salida interna se usa únicamente dentro del programa pues no tiene una existencia real.

### Programa Ladder Allen Bradley



### Programa Ladder Siemens



## Lenguaje Nemónico o Lista de Instrucciones Siemens

OB1: Título:  
Circuito control para portón

Segm. 1: Título:  
Abrir portón

V(					
O	E	0.1	"bot-abrir"	—	
O	A	2.0	"bob-abrir"	—	
)					
UN	E	0.0	"bot-parar"	—	
UN	E	0.3	"lim-abrir"	—	
UN	E	0.5	"termico"	—	
UN	A	2.1	"bob-cerrar"	—	
=	A	2.0	"bob-abrir"	—	
—					

## Lenguaje de Programación Grafcet

Es el llamado Gráfico de Orden Etapa Transición. Ha sido especialmente diseñado para resolver problemas de automatismo secuenciales. Las acciones están asociadas a las etapas y las condiciones a cumplir en las transiciones.

Muchos de los autómatas que existen en el mercado permiten la programación en GRAFCET, tanto en modo gráfico o como por lista de instrucciones.

### IMPORTANTE

Este lenguaje resulta enormemente sencillo de interpretar por operarios sin conocimientos de automatismos eléctricos.

## Plano de Funciones: FBD

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

