

# **CONTROLADORES LOGICOS PROGRAMABLES (PLC)**

## Historia

---

### ORIGEN

GM: Problemas con lógicas de relés:

- Baja confiabilidad
- Paradas imprevistas
- Díficil mantenimiento y de alto costo
- Inflexibilidad

Especificaciones de la Solución:

- Programable
- Sencillo
- Que resuelva lógicas

### RESULTADO

GM Hydramatic (1968)

- Modicon 084 (Modular Digital Controller)
- 255 E/S
- Memoria: 4K
- Peso: 46 kg

Allen Bradley: PLC (Programmable Logic Controller)

1972: Lenguaje de lógica de Escalera

- 1975: Microprocesador: Más chico y mayor capacidad.

## CONTROLADORES LOGICOS PROGRAMABLES

Memorias de mayor capacidad, almacenamiento de programas, etc.

Principio de 80's: E/S remotas, Módulos inteligentes

80's: Integración con Soft de PC, DCS

90's: Mayor capacidad, procesamiento analógico y apertura

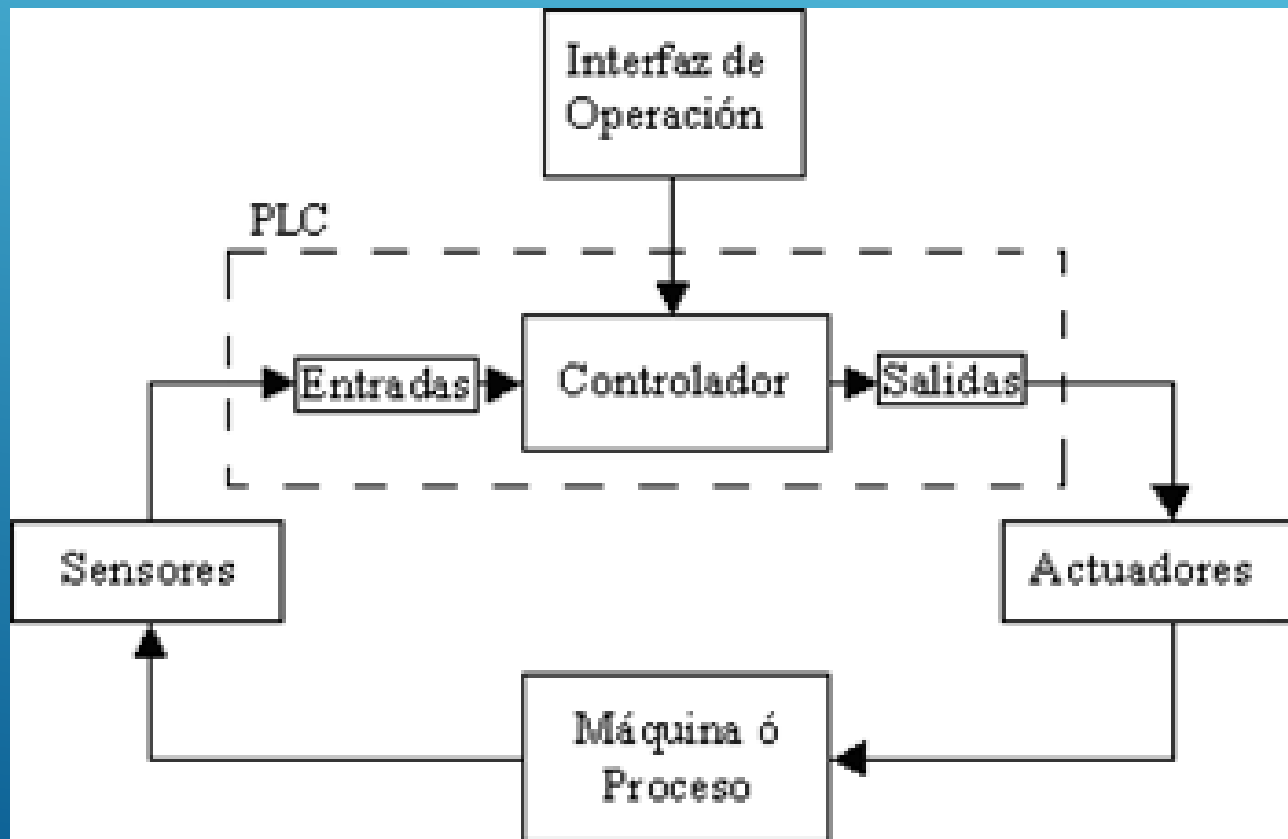
2000's: Integración con redes administrativas y buses de campo

# CONTROLADORES LOGICOS PROGRAMABLES

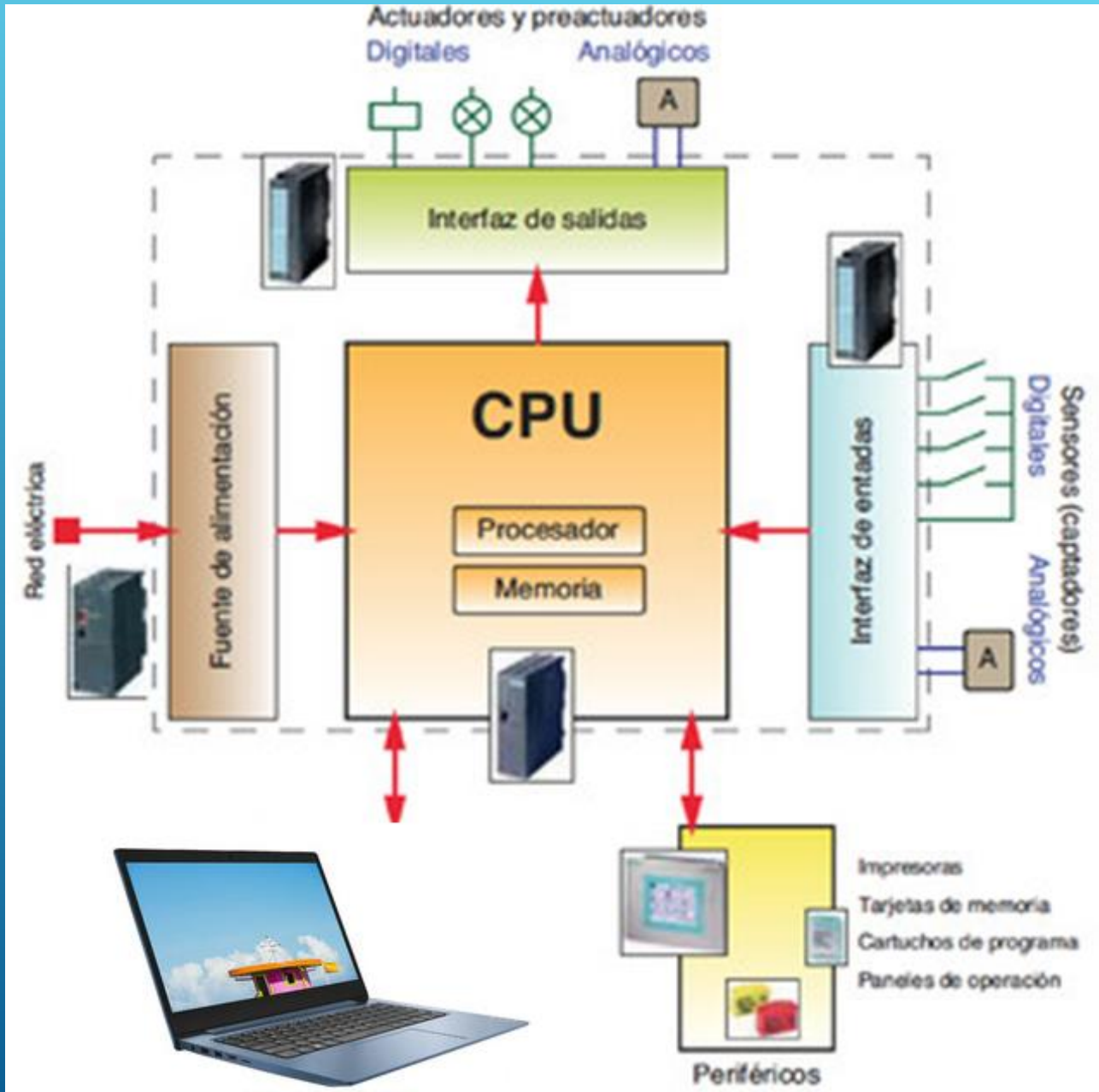
Es un dispositivo electrónico programable por el usuario y que está destinado a gobernar, dentro de un entorno industrial, máquinas o procesos lógicos combinatoriales y/o secuenciales.

Un controlador lógico programable, es más conocido por sus siglas en inglés:

## PLC (Programmable Logic Controller)

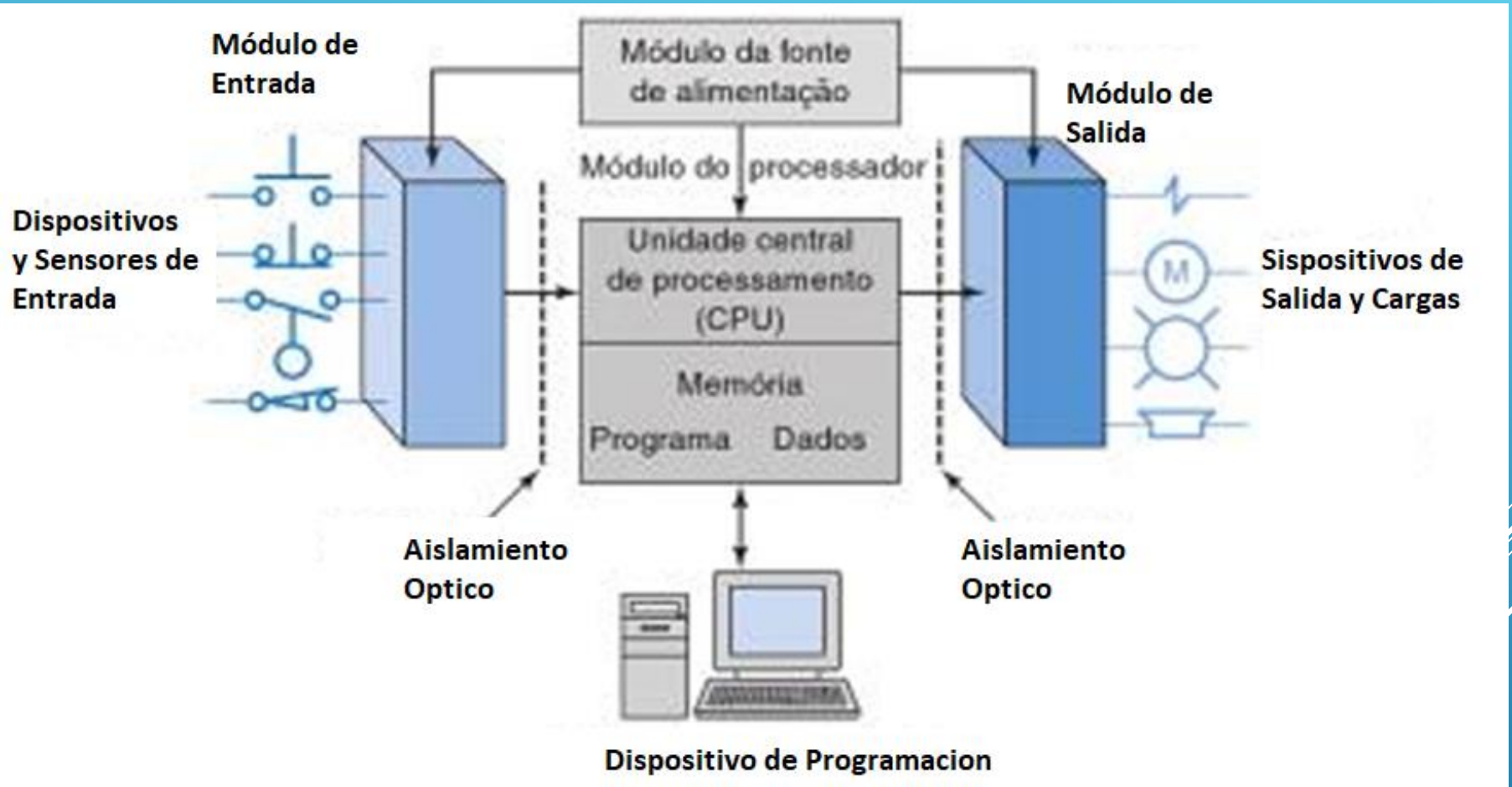


# ESTRUCTURA INTERIOR DE UN PLC



# CONTROLADORES LOGICOS PROGRAMABLES

## ESTRUCTURA INTERIOR DE UN PLC



# CONTROLADORES LOGICOS PROGRAMABLES

## ESTRUCTURA INTERIOR DE UN PLC

**CPU:** contiene la parte de procesamiento del controlador y está basada en un microprocesador que permite utilizar aritmética y operaciones lógicas para realizar diferentes funciones.

**MEMORIA:** lugar donde se guardan los datos y las instrucciones del programa.

Hay 2 tipos :

- **MEMORIA PERMANENTE:** Se guarda allí el programa que ejecuta el PLC. Utiliza memorias ROM, EPROM, EEPROM , etc.
- **MEMORIA OPERACIONAL:** Se guardan allí las operaciones y datos "on line" del PLC. Utiliza memorias RAM.

**INTERFAZ DE ENTRADAS/SALIDAS (I/O):** son aquellos módulos de señal que coordinan la entrada y salida de las señales, con aquellas internas del PLC.

Pueden ser: I/O Digitales ,Analógicas y otros tipos específicos (Módulos especiales).

# ***TIPOS DE ENTRADAS y SALIDAS***

## **ENTRADAS DIGITALES**

Pulsadores  
Llaves  
Presostatos  
Sensores de Proximidad

## **SALIDAS DIGITALES**

Lámparas  
Relés  
Contactores  
Válvulas Solenoides

## **ENTRADAS ANALOGICAS**

Transmisores de diferentes variables (Temperatura – Presion – Nivel – Caudal – PH – Conductividad – etc)

## **SALIDAS ANALOGICAS**

Válvulas de Control  
Variadores de Velocidad

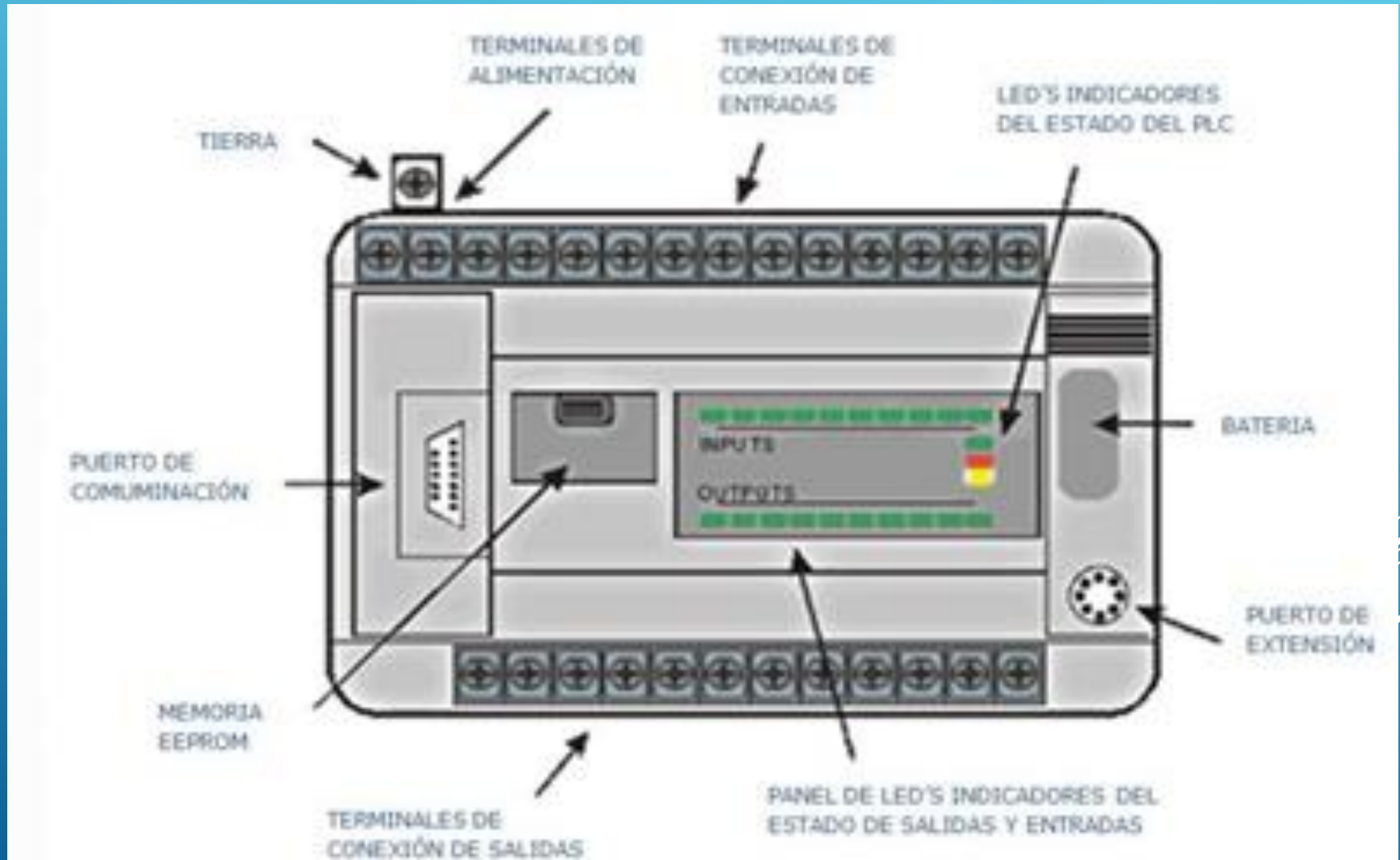
## **MODULOS ESPECIALES**

Termocuplas  
Termorresistencias  
PID



# CONTROLADORES LOGICOS PROGRAMABLES

## EXTRUCTURA EXTERIOR DE UN PLC



# CONTROLADORES LOGICOS PROGRAMABLES

## ***ESTRUCTURA EXTERIOR DE UN PLC***

**TERMINAL DE ALIMENTACION:** conexión a la fuente de energía (220Vca / 24Vcc / otras tensiones).

**FUENTE DE ALIMENTACION:** Dependiendo del PLC puede ser exterior (modulo independiente) ó interna.

**TERMINALES DE CONEXIÓN I/O:** según el tipo de PLC, compacto o modular tienen una distribución diferente.

**PUERTOS DE PROGRAMACION, CONEXIÓN A RED, AMPLIACION**

**PANEL CON LEDS INDICADORES DE ESTADO:** muestra el estado de las I/O.

## **TIPOS DE PLC**

**PLC COMPACTO** Está constituido por un solo módulo que contiene todos los componentes necesarios para el funcionamiento del PLC (I/Os , Fuente, CPU, Módulo de Comunicación, etc).

**PLC COMPACTO con programación local** : También llamado **RELE PROGRAMABLE**. Es modular (al igual que el anterior), pero posee teclas de programación y un pequeño visor

**PLC MODULAR:** Está constituido por DIFERENTES MÓDULOS, necesarios para el funcionamiento del PLC (I/Os , Fuente, CPU, Módulo de Comunicación, etc). Son ampliables, y pueden alcanzar una gran cantidad de I/Os  
Se utilizan Racks de Conexión para la inserción de los módulos

# CONTROLADORES LOGICOS PROGRAMABLES

## TIPOS DE PLC

### PLC compacto



**TIPOS DE PLC**

**RELES Programables**



## CONTROLADORES LOGICOS PROGRAMABLES

### *PLC Modular*



CONTROLADORES LOGICOS PROGRAMABLES

## ***PLC Modular RACKS DE CONEXION***



**FUENTE DE ALIMENTACION**



**MODULO CPU**

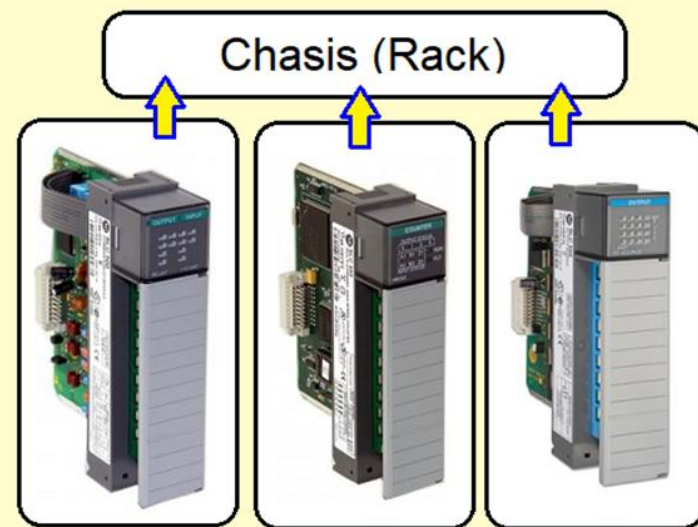
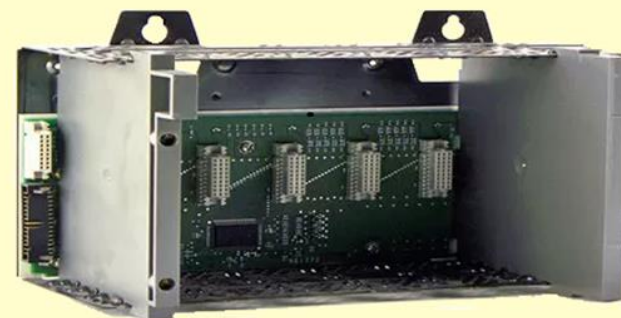
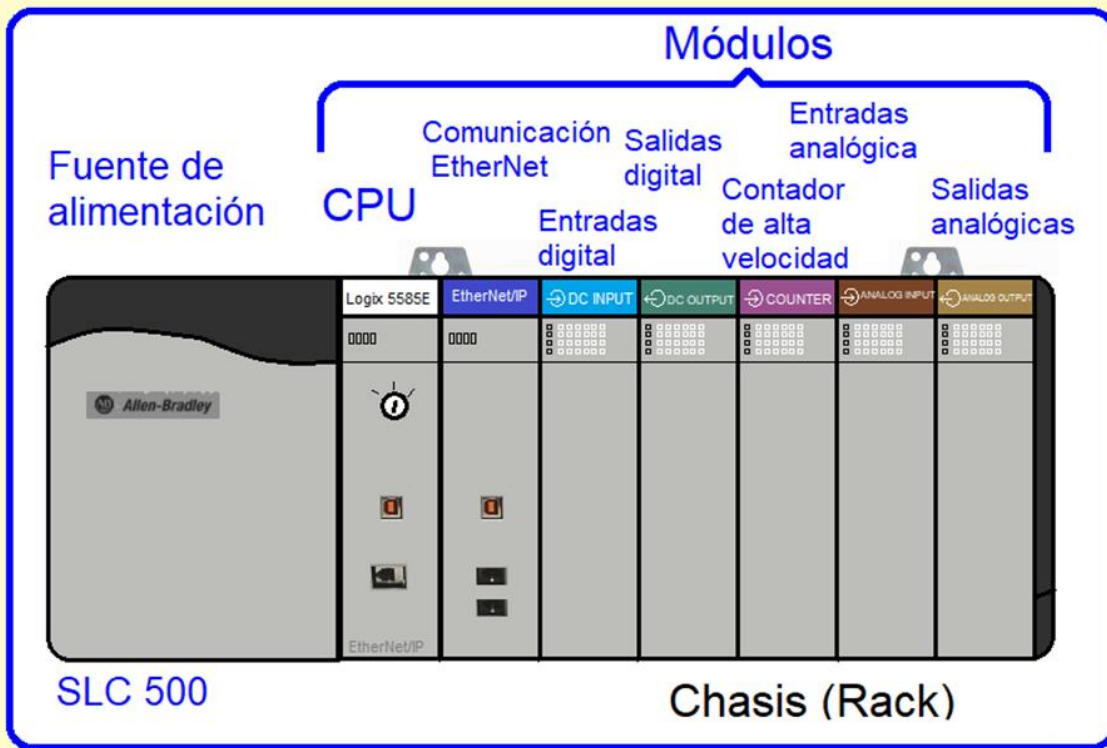




***PLC Modular***  
***MODULOS DE ENTRADAS Y SALIDAS***



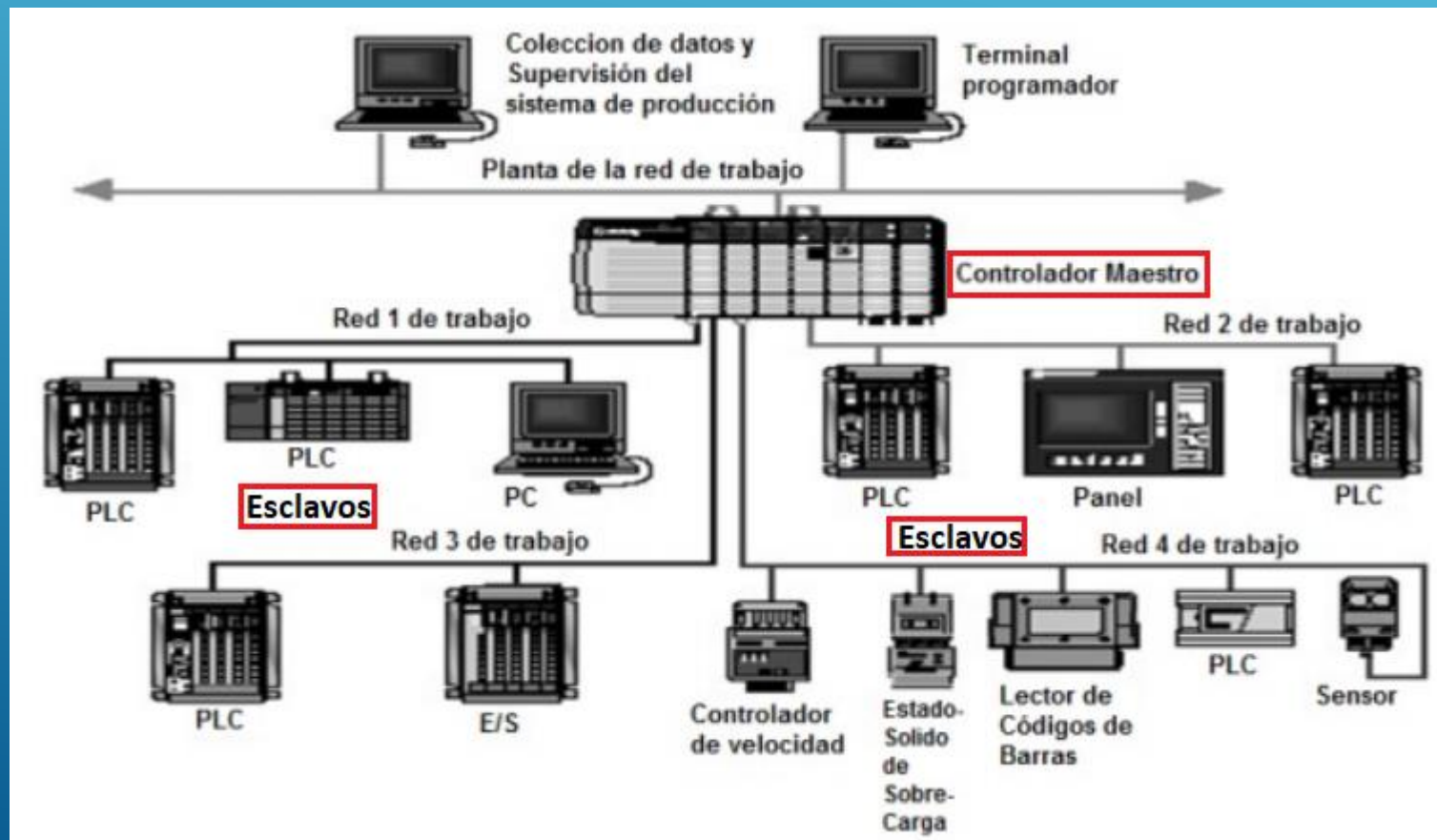
# PLC Modular

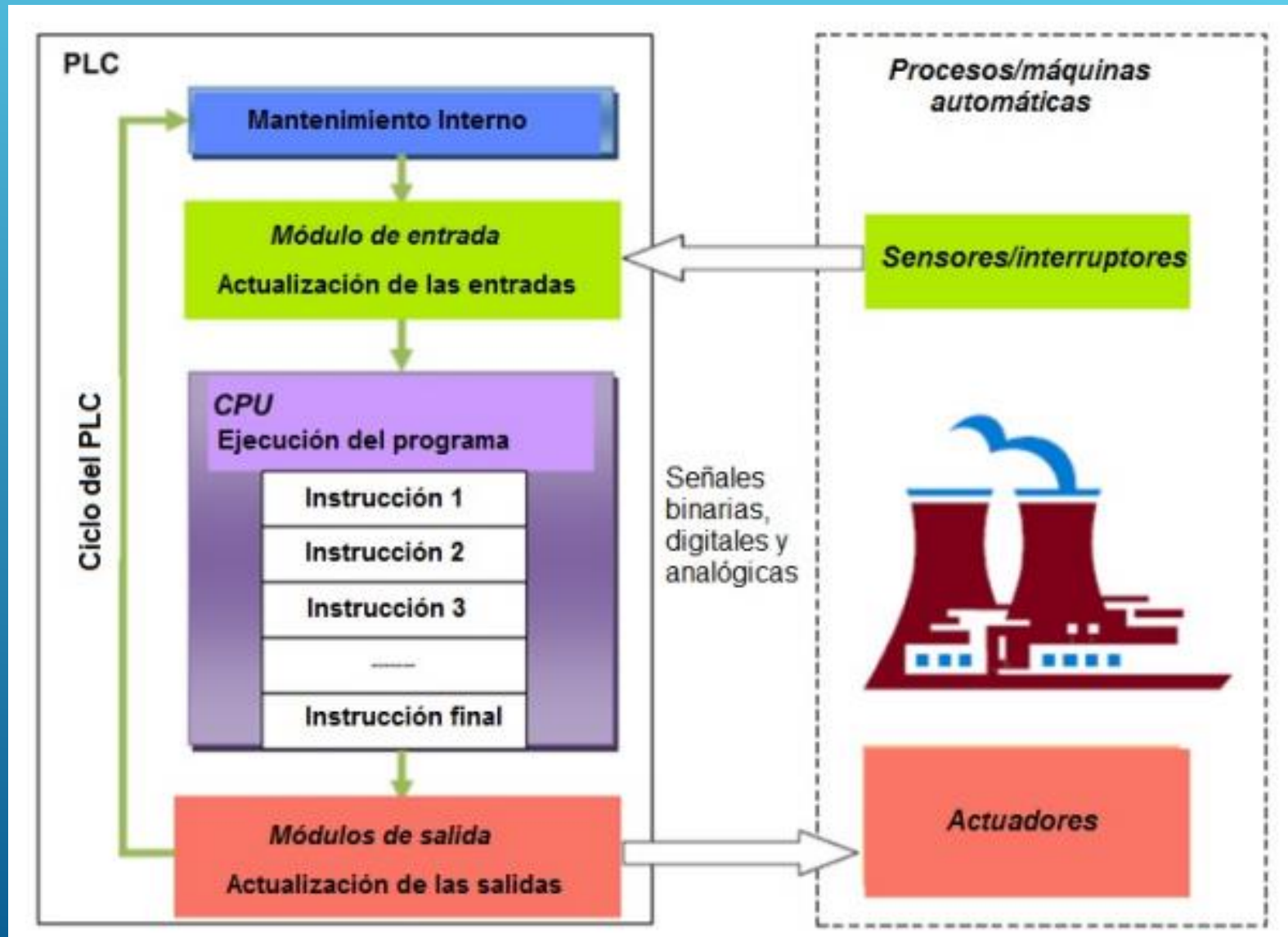


## ARQUITECTURA MAESTRO ESCLAVO EN PLCs

Este esquema de comunicación entre PLCs está ampliamente difundido en la industria para redes de integración de equipos de control.

Utiliza la denominada configuración "MAESTRO-ESCLAVO", y se utiliza en comunicaciones entre PLC y otros sistemas como SCADA's y en DCS's.



**PRINCIPIO DE FUNCIONAMIENTO DE UN PLC**

## ***PRINCIPIO DE FUNCIONAMIENTO DE UN PLC***

Un PLC funciona cíclicamente, como se describe a continuación:

- 1)** Cada ciclo comienza con un trabajo interno de mantenimiento del PLC como el control de memoria, diagnóstico etc. Esta parte del ciclo se ejecuta muy rápidamente de modo que el usuario no lo perciba.
- 2)** El siguiente paso es la actualización de las entradas. Las condiciones de la entrada de los Módulos de Señal se leen y convierten en señales binarias o digitales. Estas señales se envían a la CPU y se guardan en los datos de la memoria.
- 3)** Después, la CPU ejecuta el programa del usuario, el cual ha sido cargado secuencialmente en la memoria (cada instrucción individualmente). Durante la ejecución del programa se generan nuevas señales de salida.
- 4)** El último paso es la actualización de las salidas. Tras la ejecución de la última parte del programa, las señales de salida (binaria, digital o analógica) se envían a los módulos I/O desde los datos de la memoria. Estas señales son entonces convertidas en las señales apropiadas para las señales de los actuadores.

Al final de cada ciclo el PLC comienza un ciclo nuevo.

- 5)** WatchDog (Perro guardian): es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado o superado un tiempo preestablecido.

# ***DIMENSIONAMIENTO***



## ***DIMENSIONAMIENTO DE UN PLC***

Previamente a la selección de un PLC, se deben realizar un dimensionamiento de cantidad de I/O que el mismo debe poseer.

Se deben computar cantidad de:

- Entradas y Salidas Digitales
- Entradas y Salidas Analógicas
- Posibilidad de aumentar las I/Os
- Capacidad de procesamiento de datos del PLC
  - Automatización de Maquinarias
  - Automatización de Procesos
- Capacidad de Comunicación
  - Con otros autómatas (modo serial o IP)
  - Con un SCADA ó DCS

# ***PROGRAMACION***





# ***PROGRAMACION DE UN PLC***

## ***Norma IEC 1131***

### **CONTENIDO**

- **IEC 1131-1- Informaciones generales (1992)**
- **IEC 1131-2- Especificaciones y ensayos de equipos (1992)**
- **IEC 1131-3- Lenguajes de programación (1993)**
- **IEC 1131-4- Recomendaciones al usuario**
- **IEC 1131-5- Especificaciones de servicios de mensajería**

***Norma IEC 1131-3***

**La norma define :**

**Los lenguajes de programación ,**

**y para todos los lenguajes :**

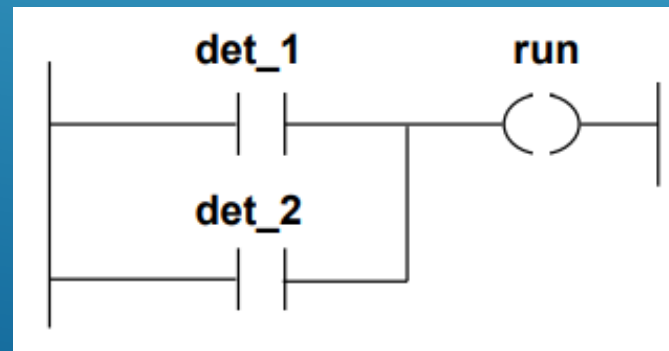
- **la sintaxis y representación gráfica de los objetos**
- **la estructura de programas**
- **la declaración de variables .**

**CONTROLADORES LOGICOS PROGRAMABLES*****Norma IEC 1131-3*****Lenguajes normalizados :**

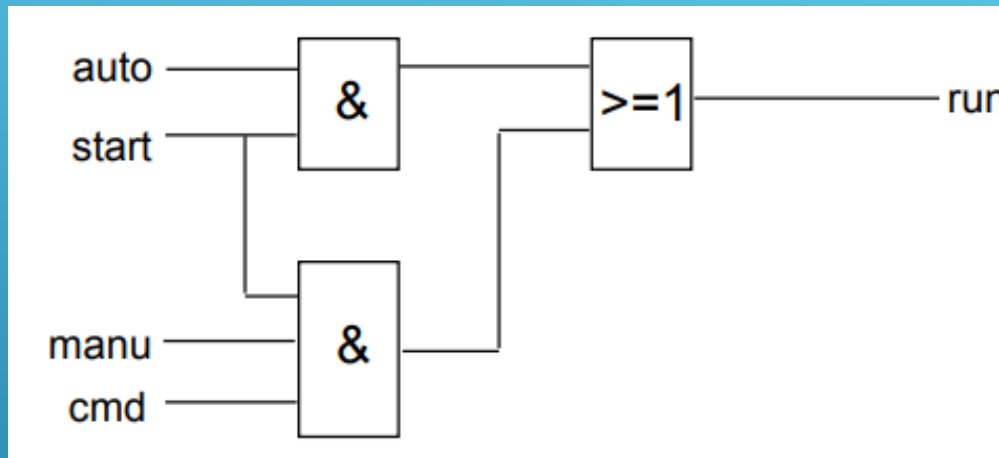
- **LADDER DIAGRAM ( LD ) o lenguaje (diagrama) de contactos**
- **FUNCTION BLOCK DIAGRAM ( FBD ) o esquema de bloques funcionales**
- **INSTRUCTION LIST ( IL ) o lista de instrucciones**
- **STRUCTURED TEXT ( ST ) o lenguaje textual estructurado**
- **SEQUENTIAL FUNCTION CHART ( SFC ) o diagrama funcional de secuencias (basado en el GRAFCET).**

***Norma IEC 1131-3*****DIAGRAMA LADDER (LD)**

- Elementos gráficos organizados en redes conectadas por barras de alimentación
- Forma gráfica de los elementos impuesta
- Evaluación de la red por elementos interconectados
- Elementos utilizados : contactos , bobinas , funciones , bloques funcionales
- Elementos de control de programa (salto , return ,...)



## CONTROLADORES LOGICOS PROGRAMABLES

**Norma IEC 1131-3****DIAGRAMA DE BLOQUE DE FUNCIONES (FBD)****LISTA DE INSTRUCCIONES (IL)**

<i>Etiqueta</i>	<i>Operador</i>	<i>Operando</i>	<i>Comentario</i>
MARCHA :	<b>LD</b>	<b>%IX1</b>	(* pulsador *)
	<b>ANDN</b>	<b>%MX5</b>	
	<b>ST</b>	<b>%QX2</b>	(* marcha *)

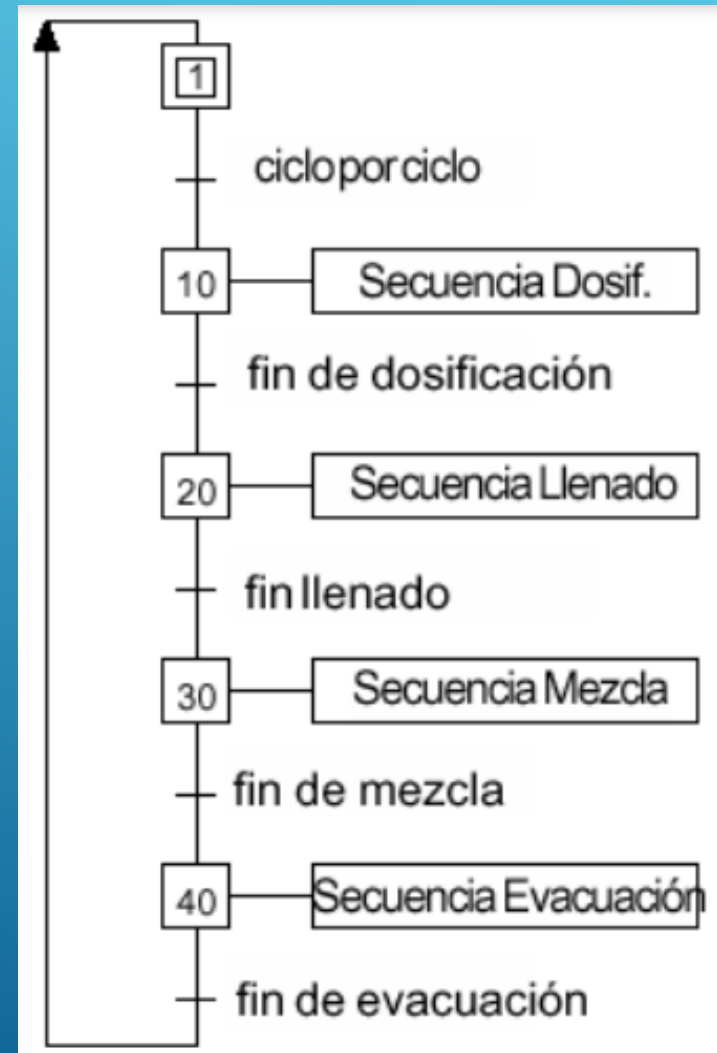
## CONTROLADORES LOGICOS PROGRAMABLES

**Norma IEC 1131-3**TEXTO ESTRUCTURADO (ST)

```

J:=1 ;
WHILE J<=100 AND X1< >X2 DO ;
J:=J+2 ;
END_WHILE ;

```

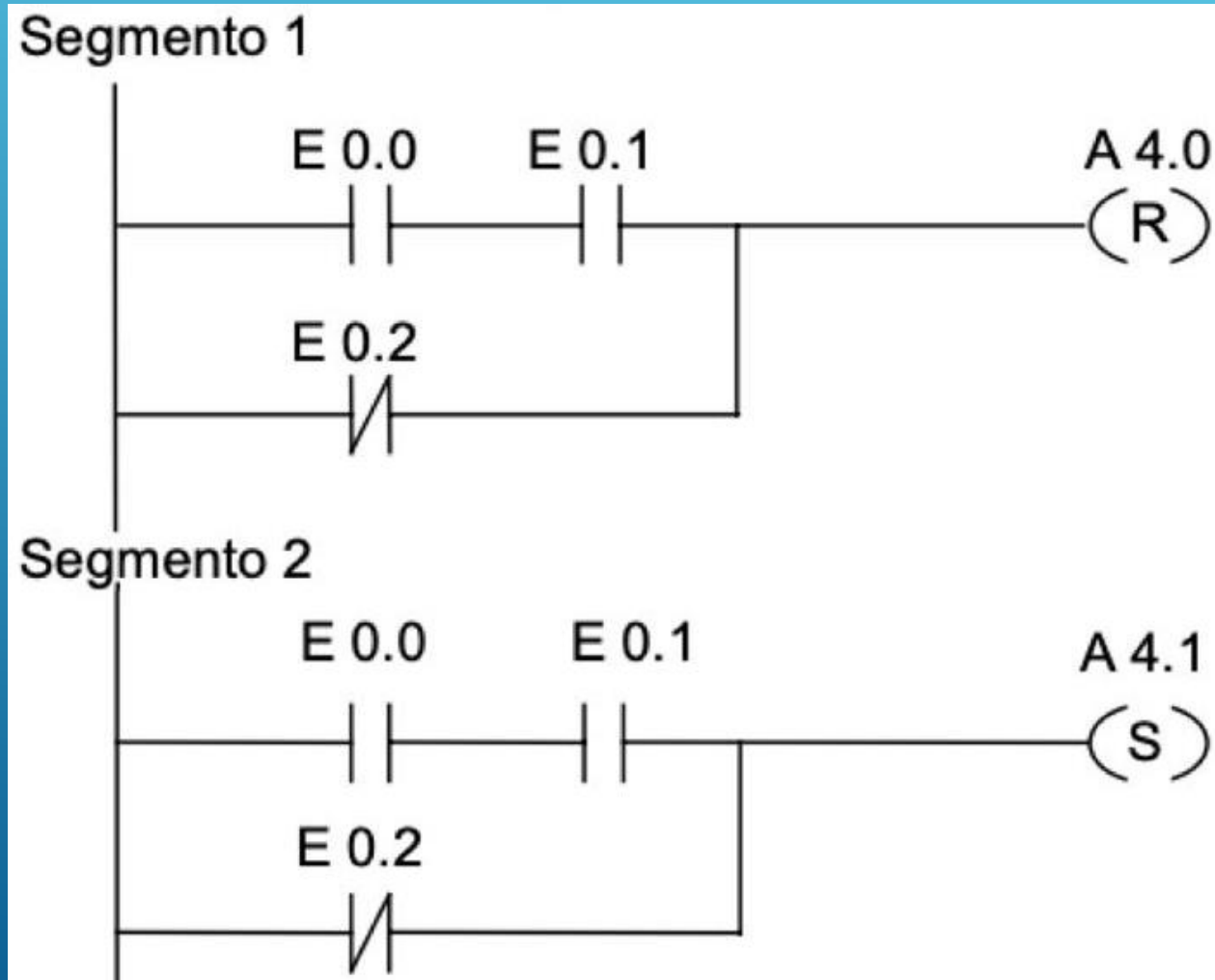
SEQUENTIAL FUNCTION CHART(SFC)

*PROGRAMACION*

# *Diagramas LADDER*

**DIAGRAMA LADDER**

Es el lenguaje de programación mas comúnmente utilizado

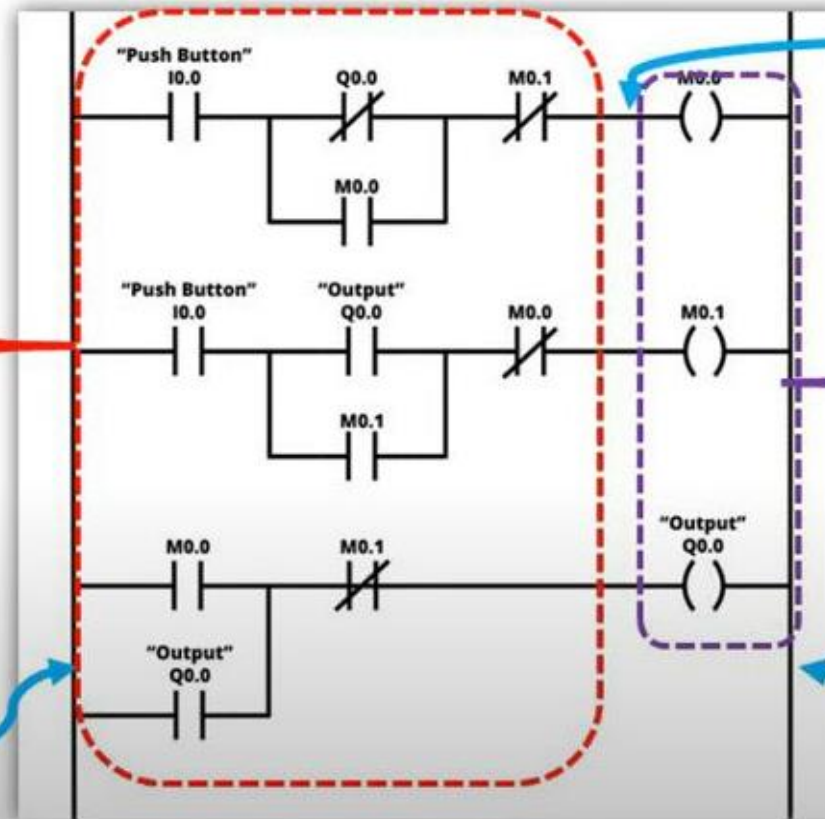




## CONTROLADORES LOGICOS PROGRAMABLES

**DIAGRAMA LADDER****CONDICIONES**  
Elementos de  
**ENTRADA**

- ✓ pulsadores
- ✓ Interruptores
- ✓ Sensores
- ✓ Marcas / memoria



rung

**ACCIONES**  
Elementos de  
**SALIDA**

- ✓ Actuadores (motores, cilindros, válvulas, etc).
- ✓ Indicadores.
- ✓ Marcas / memoria

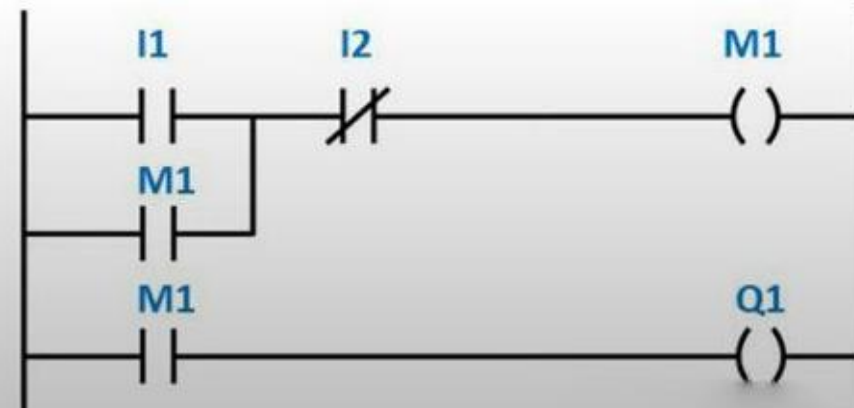
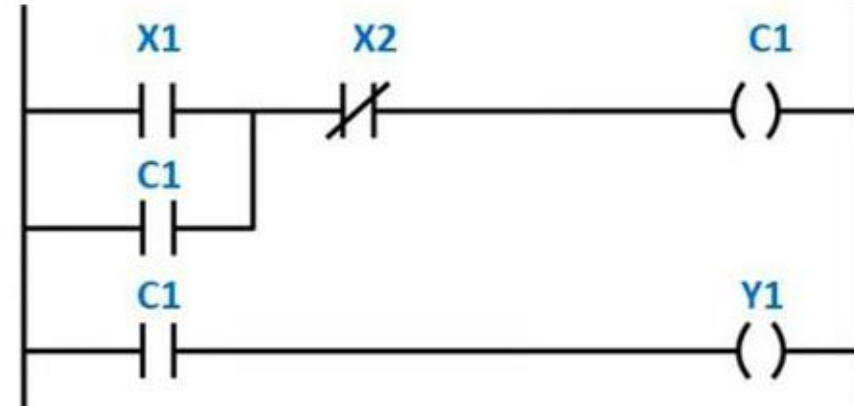
Línea energizada L1

Línea energizada L2

## CONTROLADORES LOGICOS PROGRAMABLES

**DIAGRAMA LADDER****ETIQUETAS**

Símbolo 1	Símbolo 2	Concepto
X	I	Entrada
Y	Q	Salida
C	M	Bandera o Memoria
T	T	Temporizador
C	C	Contador
S	S	Instrucción Set
R	R	Instrucción Reset



## *Norma IEC 1131-3*

### *DIAGRAMA LADDER (ESCALERA)*

#### ENTRADAS

# % I x.y

%: Símbolo de Programación

I : Entrada (Input)

x (Nro de PLC) . y (Nro de Entrada)

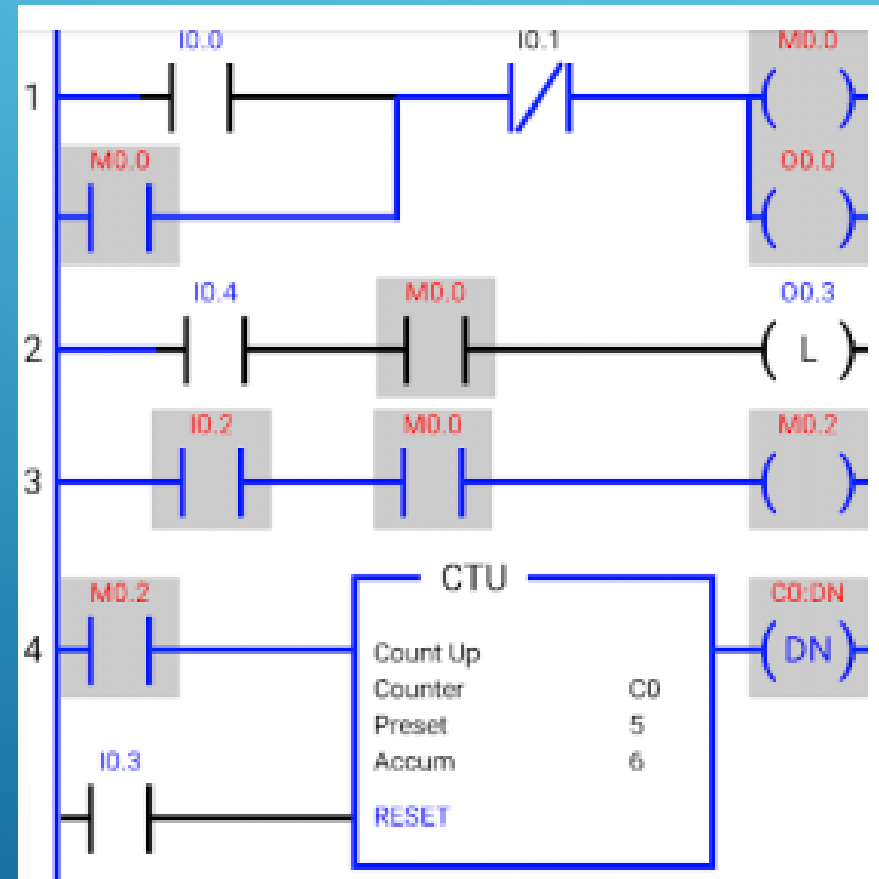
#### SALIDAS

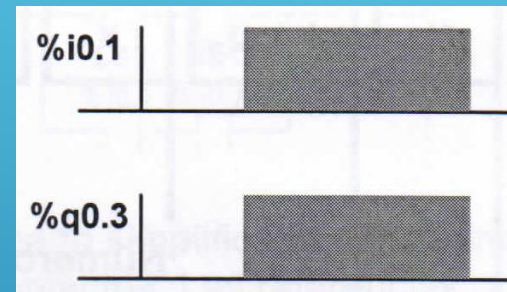
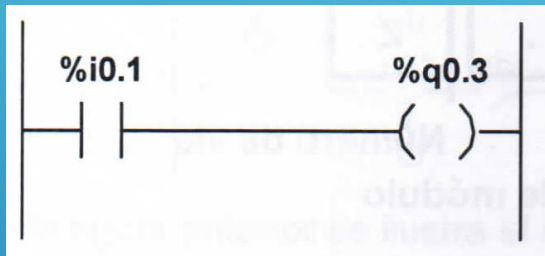
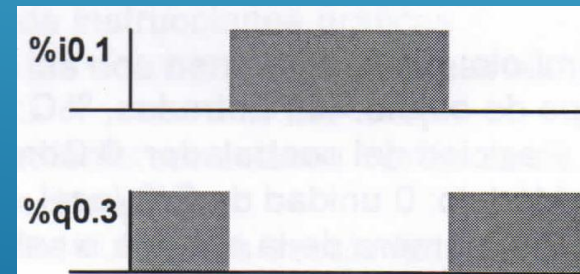
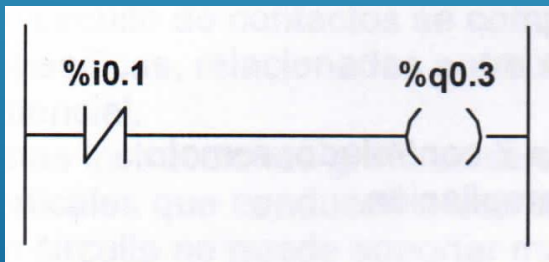
# % Q x.y

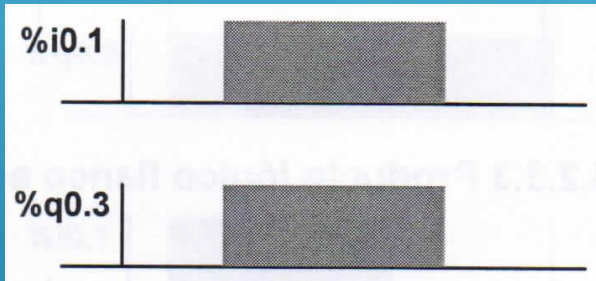
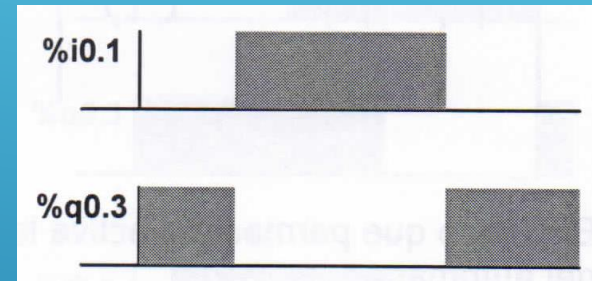
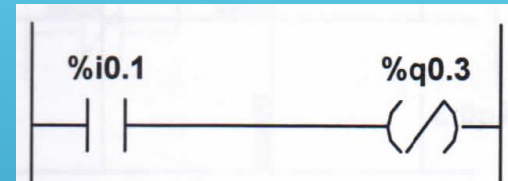
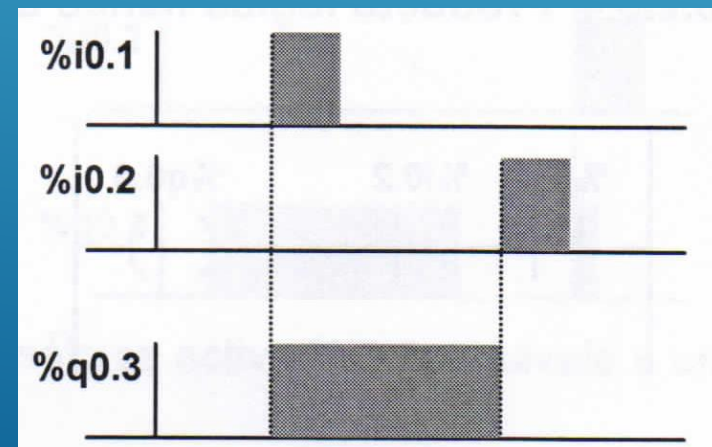
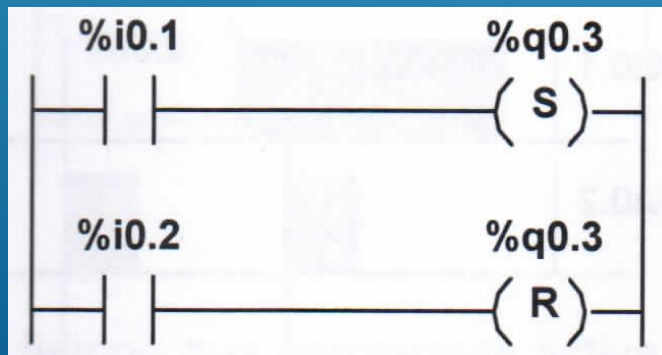
%: Símbolo de Programación

Q : Salida (Quit)

x (Nro de PLC) . y (Nro de Salida)

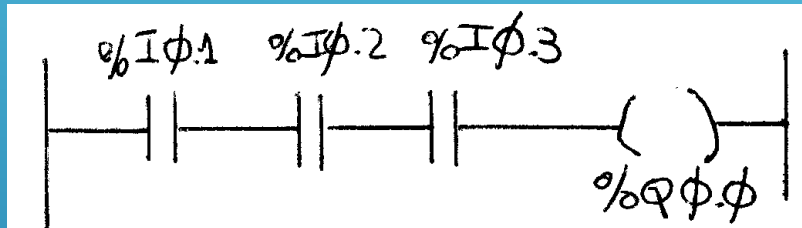


**DIAGRAMA LADDER (ESCALERA)****INSTRUCCIONES DE CARGA****CONTACTO NORMAL ABIERTO****CONTACTO NORMAL CERRADO**

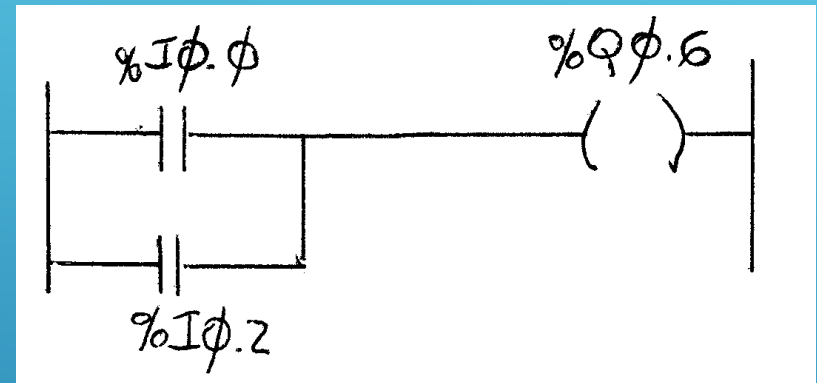
**DIAGRAMA LADDER (ESCALERA)****INSTRUCCIONES DE ASIGNACION****BOBINA DIRECTA (salida)****BOBINA INVERSA (salida)****BOBINAS SET Y RESET**

## ESQUEMAS DE CONEXIONADO

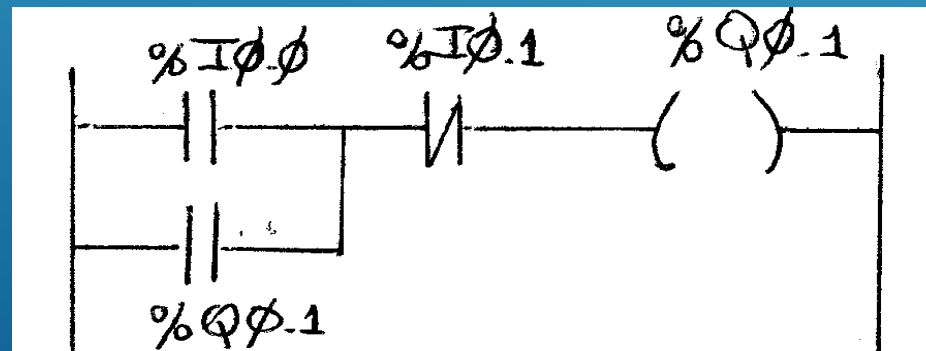
### CONTACTOS SERIE (PRODUCTO LOGICO)

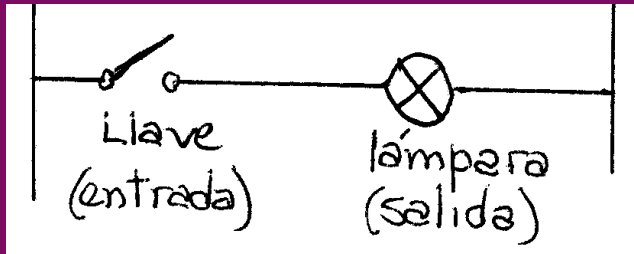
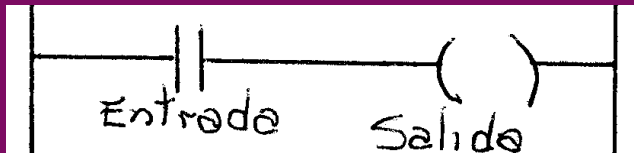
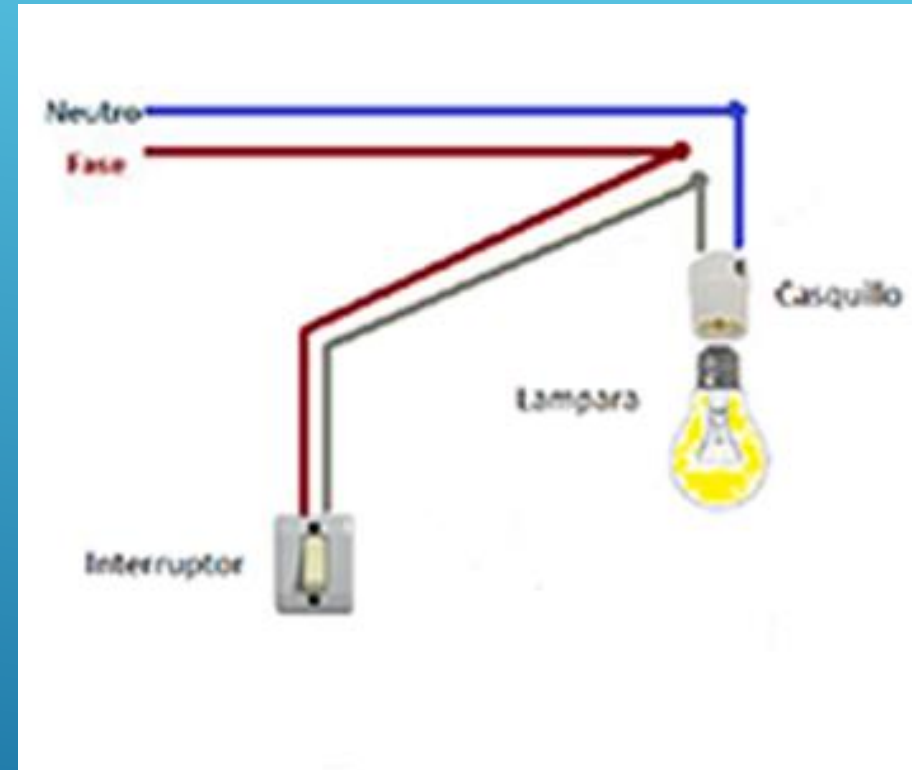
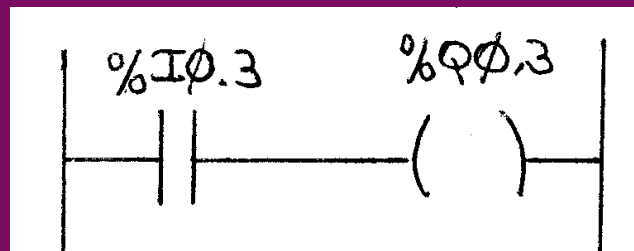


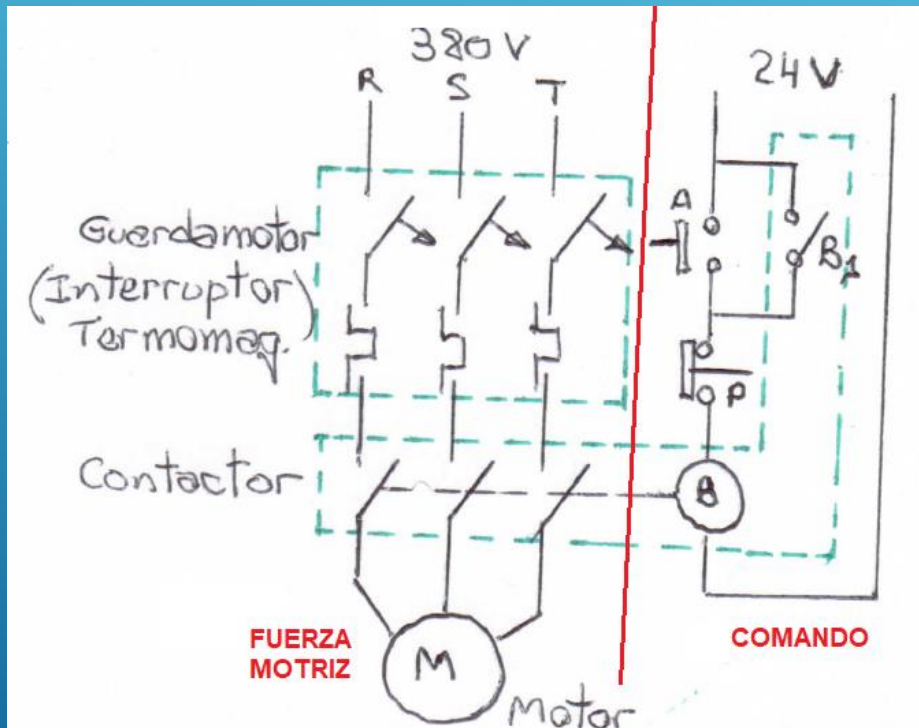
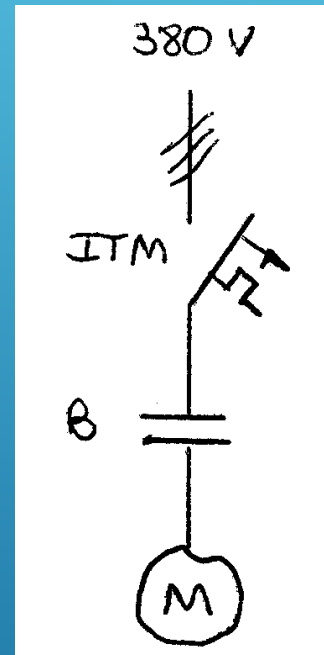
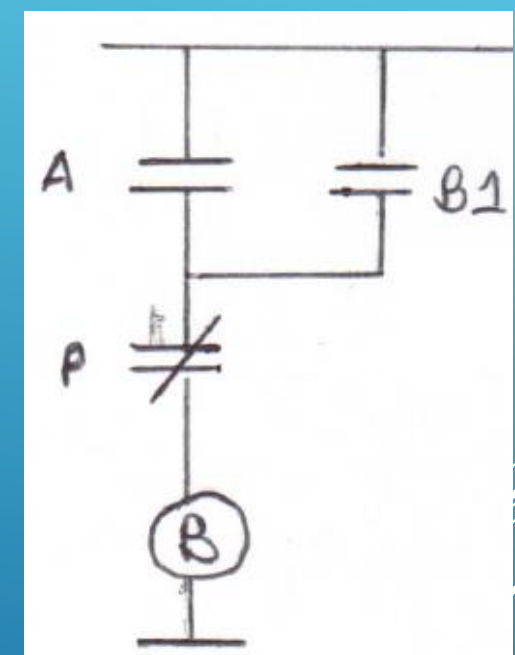
### CONTACTOS PARALELO (SUMA LOGICA)



### CONTACTOS COMBINADOS (SERIE/PARALELO)



**DIAGRAMA LADDER (ESCALERA)****EJEMPLO N°1: ENCENDIDO DE UNA LAMPARA****CONTACTOS SERIE****ESQUEMA ELECTRICO (Simplificado)****ESQUEMA LADDER**

**DIAGRAMA LADDER (ESCALERA)****EJEMPLO N°2: ARRANQUE/PARADA DE MOTOR****COMANDO Y FUERZA MOTRIZ****DIAGRAMA UNIFILAR FUERZA MOTRIZ****DIAGRAMA UNIFILAR COMANDO**

A: Pulsador de Marcha

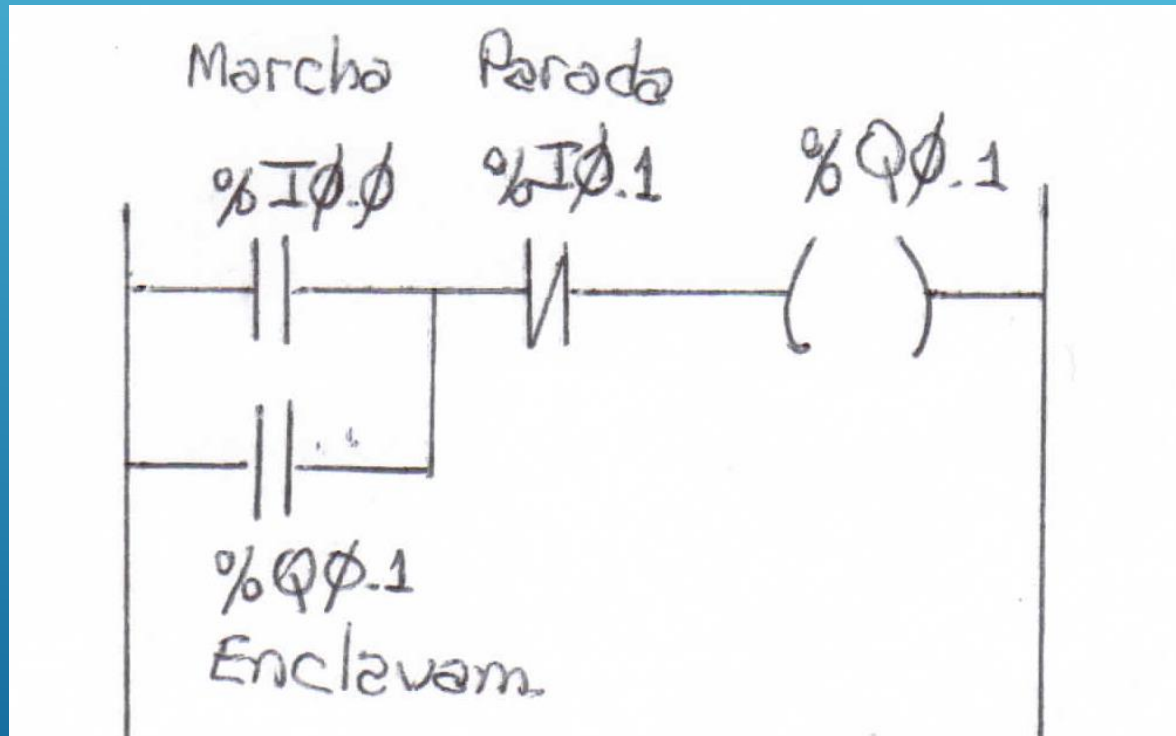
P: Pulsador de Parada

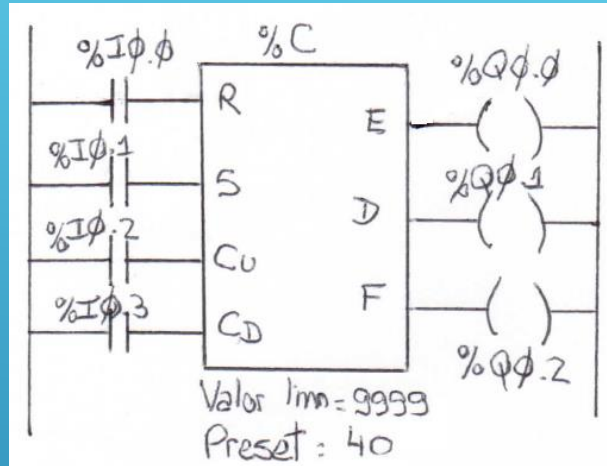
B: Contactor

B1: Contacto auxiliar

ITM: Guardamotor



**DIAGRAMA LADDER (ESCALERA)****EJEMPLO N°2: ARRANQUE/PARADA DE MOTOR****DIAGRAMA LADDER**

**DIAGRAMA LADDER****BLOQUE CONTADOR**

**CU** : Cuenta en forma ascendente

**CD** : Cuenta en forma descendente

**S** : Se ajusta el contador al valor de preselección (Preset)

**R** : Se pone a Cero el Contador

**D** : se pone en "1" cuando se llega al valor de preselección

39 → D= "0"

40 → D= "1"

41 → D= "0"

**E** : Se pone en "1" al llegar a Cero

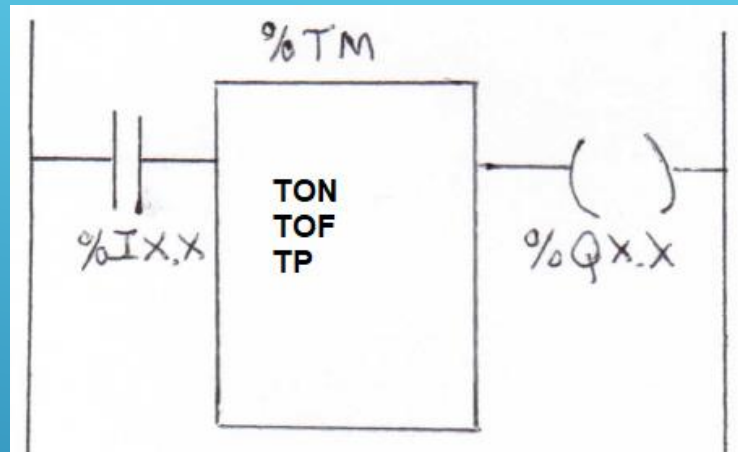
02 → E= "0"

01 → E= "0"

00 → E= "1"

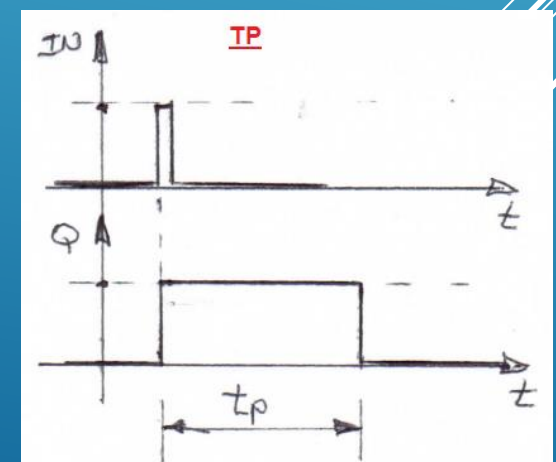
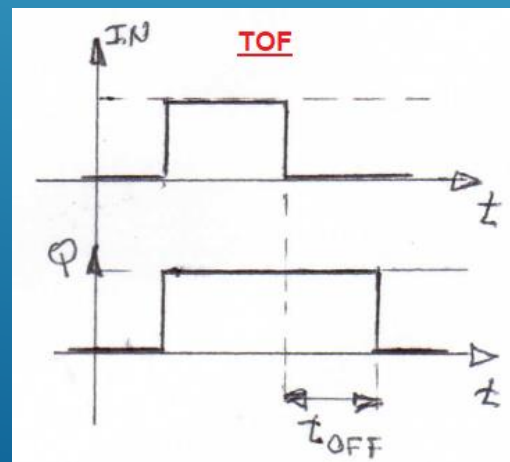
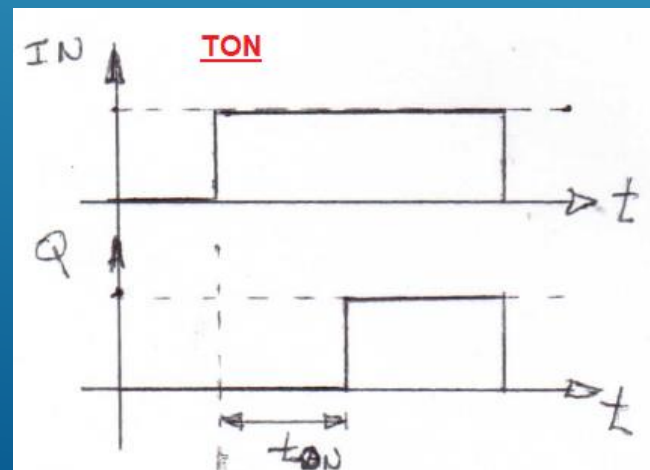
9999 → E= "0"

**F (Full)**: Se pone en "1" al superar el valor limite ("9999")

**DIAGRAMA LADDER****BLOQUE TEMPORIZADOR****TIPOS DE TEMPORIZADORES**

- TON : Temporizado con retardo a la conexión
- TOF : Temporizado con retardo a la desconexión
- TP : Temporizado monoestable

t : tiempo de ajuste



## BITS Y PALABRAS de un PLC

Los PLCs poseen Bits y Palabras, que pueden ser

Bits / Palabras de SISTEMA : ya definidos en el PLC

Bits / Palabras de USUARIO : definidos por el usuario o el Proceso que este controlando el PLC

BIT: Estado que puede tomar una variable digital . La misma puede ser "0" ó "1"

BYTE: Formato digital, constituido por un Conjunto de 8 Bits.

Este formato tiene la siguiente forma

$x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1 = 1 \text{ Byte}$  (donde  $X_i$  puede ser "0" ó "1")

Ejemplo:

$1 0 0 1 1 1 0 1 = 1 \text{ Byte}$

PALABRA (Word): Formato digital, constituido por 2 Bytes (B2 B1)

Ejemplo:

$1 1 1 0 1 0 0 1 \quad 0 0 1 0 0 0 1 1 : W$

# OBJETOS BITS Y PALABRAS

## OBJETOS BIT

### BITs DE ENTRADA/SALIDA

Estos bits son las "imágenes lógicas" de los estados eléctricos de las E/S.

Se almacenan en la memoria de datos y se actualizan durante cada exploración de la lógica del programa.

Sintaxis: **%I0.1** - **%Q0.2**

### BITs INTERNOS

Los bits internos son áreas de memoria interna utilizadas para almacenar valores intermedios durante la ejecución de un programa.

Sintaxis: **%Mi**

### BITs DE SISTEMA

Los bits de sistema supervisan el funcionamiento correcto del autómatas y la correcta ejecución del programa de aplicación.

Sintaxis: **%Si**

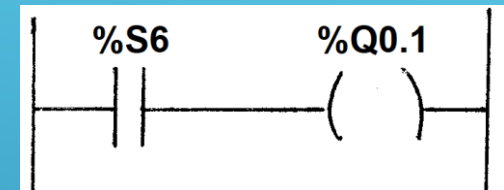
# OBJETOS BITS Y PALABRAS

## OBJETOS BIT

### BITs DE SISTEMA

#### Ejemplos:

**%S6:** Base de Tiempo de 1 segundo  
Es un Reloj interno del autómeta



- %S11:** Desbordamiento del Watchdog del PLC .  
Cuando el Tiempo de Ejecución del programa supera al Tiempo de Exploración Máximo (Watchdog), %S11= 1
- %S12:** Este bit refleja el estado de ejecución del autómeta. El sistema establece el bit en 1 cuando el autómeta está en ejecución.

# OBJETOS BITS Y PALABRAS

## OBJETOS PALABRA

Objetos de palabra direccionados en forma de palabras de 16 bits, almacenados en la memoria de datos.

Pueden Ser:

- Palabras internas (**%MWi**) (palabras de memoria)
- Palabras constantes (**%KWi**)
- Palabras de intercambio de E/S (**%Iwi.j**, **%Qwi.j**) entre autómatas
- Palabras de sistema (**%SWi**) Bloques de función (datos de tiempo de ejecución o configuración)

# OBJETOS BITS Y PALABRAS

## OBJETOS PALABRA

### PALABRAS INTERNAS

Palabras empleadas para almacenar valores durante la operación en la memoria de datos. Son dinámicas. Se van refrescando con la ejecución del programa y la actualización de las I/Os.

Sintaxis: **% MWi (i= 1,2,3,....)**

### PALABRAS CONSTANTES

Almacenan constantes o mensajes alfanuméricos. Su contenido sólo se puede escribir o modificar mediante el Soft de Programación utilizado, durante la configuración.

Sintaxis: **% KWi (i= 1,2,3,....)**

### PALABRAS DE INTERCAMBIO DE RED

Asignadas a los autómatas conectados como conexiones remotas. Estas palabras se utilizan para la comunicación entre los PLCs.

Sintaxis: **% INW i.j - %QNW i.j**



# OBJETOS BITS Y PALABRAS

## OBJETOS PALABRA

### PALABRAS SISTEMA

Proporcionan acceso a los datos que proceden directamente del autómata mediante la lectura de las palabras %SWi.

Sintaxis: **% SWi (i= 1,2,3,...)**

Ejemplo:

Palabras de sistema	Función	Descripción
%SW11	Valor del watchdog del software	Contiene el valor máximo del watchdog. El valor (de 10 a 500 ms) se define mediante la configuración.
%SW54 %SW55 %SW56 %SW57	Fecha y hora de la última parada	Palabras de sistema que contienen la fecha y la hora del último corte de alimentación o de la última detención del autómata
		%SW54 SS Segundos
		%SW55 HHMM Hora y minutos
		%SW56 MMDD Mes y día
		%SW57 SSAA Siglo y año