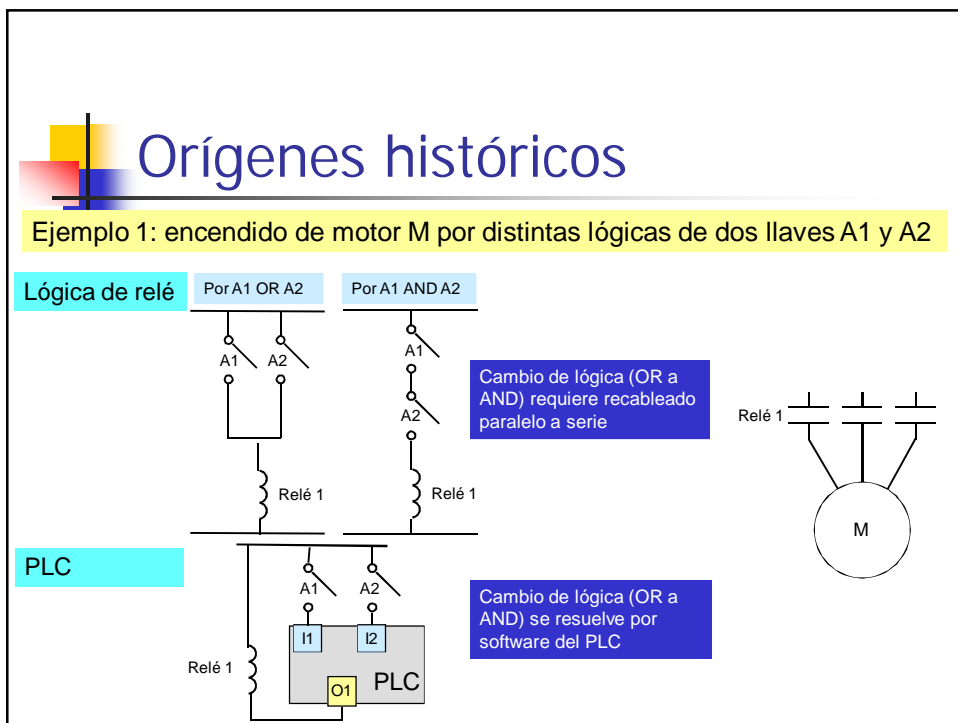


# PLC

## LENGUAJE LD

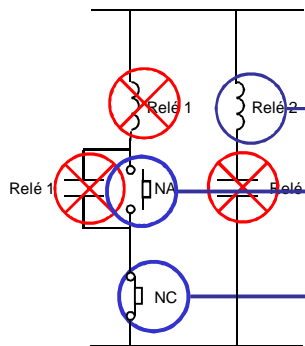
(Ladder)



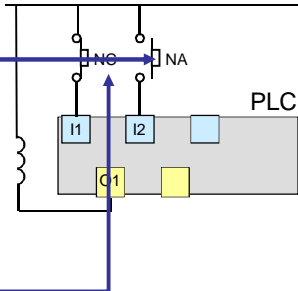
## Orígenes históricos

### Ejemplo 2: encendido-apagado de motor por pulsadores

#### Circuito por lógica de relé



#### Circuito por PLC



El relé intermedio se reemplaza por software del PLC

## Orígenes históricos

### Objetivos del PLC:

Aumentar la confiabilidad

Aumentar la flexibilidad

Mantener la facilidad de soporte

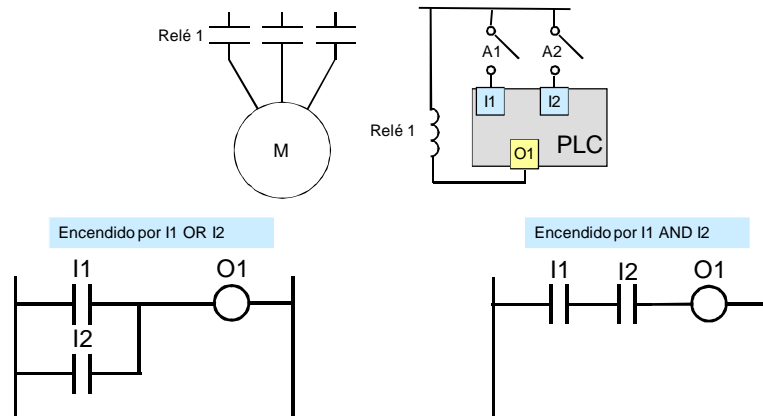
Lenguaje de programación fácilmente entendido por electricistas de planta

LADDER (LD)



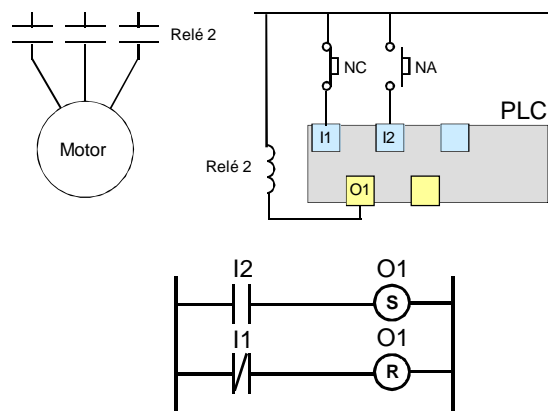
## Orígenes históricos

### Programas LD del ejemplo 1



## Orígenes históricos

### Programa LD del ejemplo 2





## Estructura programa LD

Lenguaje gráfico

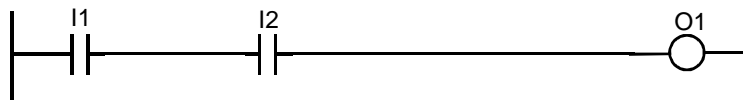
Programa consiste en una secuencia de escalones (rungs)

Estructura de cada escalón:

- 1 Comienza en una barra de alimentación izquierda (positivo de la fuente)
- 2 Condiciones y acciones, conectadas por líneas de conexión
- 3 Termina en una barra de alimentación derecha (negativo de la fuente)

Los escalones se ejecutan de arriba hacia abajo

Cada escalón se ejecuta de izquierda a derecha



## Diferencias entre fabricantes

Diferencias entre fabricantes: implementación y nomenclatura de las instrucciones

Este curso usa la nomenclatura de los PLCs del laboratorio:

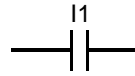
En algunos aspectos estándar

Fácilmente comprensible

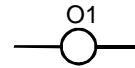


## Símbolos básicos

Contacto (entrada)



Bobina (salida)



Cada símbolo tiene asociado un bit de memoria

Un bit de memoria se refiere por:

Su dirección

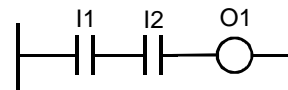
Su etiqueta (label)



## Operaciones Básicas

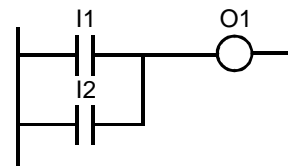
AND (conexión serie):

$(O1 = 1)$  si  $(I1 = 1)$  y  $(I2 = 1)$



OR (conexión paralelo):

$(O1 = 1)$  si  $(I1 = 1)$  o  $(I2 = 1)$





## Instrucciones con BITs

Contacto directo



Contacto normalmente abierto

Verdadero si bit vale 1

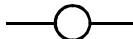
Contacto invertido



Contacto normalmente cerrado

Verdadero si bit vale 0

Bobina directa

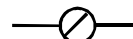


Análoga a la bobina de un relé

Si el escalón es 1, escribe 1 en bit asociado

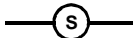
Si el escalón es 0, escribe 0 en bit asociado

Bobina invertida



Función inversa de bobina directa

SET (o LATCH)



Instrucción de salida retentiva

Si el escalón es 1, escribe 1 en bit asociado

Si el escalón es 0, no hace nada

RESET (UNLATCH)



Se usa en conjunto con el SET

Si el escalón es 1, escribe 0 en bit asociado

Si el escalón es 0, no hace nada

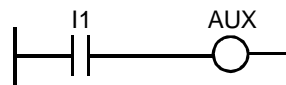
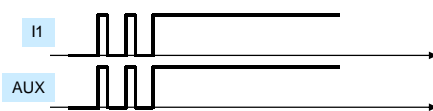


## Instrucciones con BITs

### Ejemplo: BOBINA vs. SET/RESET

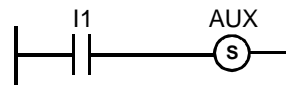
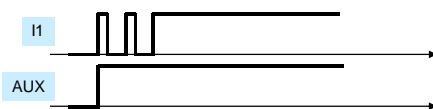
Contador de pulsos con rebotes

Solución con bobina



Siguen instrucciones para contar pulsos

Solución con SET



Siguen instrucciones para contar pulsos y realizar RESET luego de filtrado



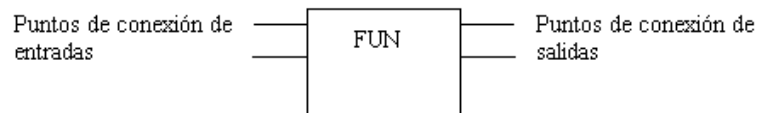
## Bloques funcionales

- n Las instrucciones de aquí en adelante se representan gráficamente como bloques funcionales
- n Bloque funcional: objeto gráfico que se representa por un rectángulo, con puntos de conexión de entradas, conexión de salidas y un identificador



## Bloques funcionales

- n El identificador describe función del bloque
- n Ejemplo: bloque funcional que implementa la función FUN, con dos entradas y dos salidas



- n Las entradas y salidas son datos. El tipo de dato de cada una depende del bloque.

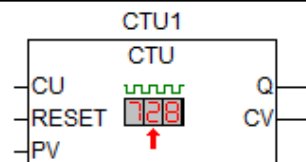


## Contadores

- n Existen tres tipos:
  - n UP - Counters: CTU
  - n DOWN – Counters: CTD
  - n UP-DOWN Counters: CTUD
  
- n Rango de cuenta: depende de fabricante. En PLCs de laboratorio:
  - n CTU cuenta desde 0 a 32767
  - n CTUD cuenta desde 0 a 32767



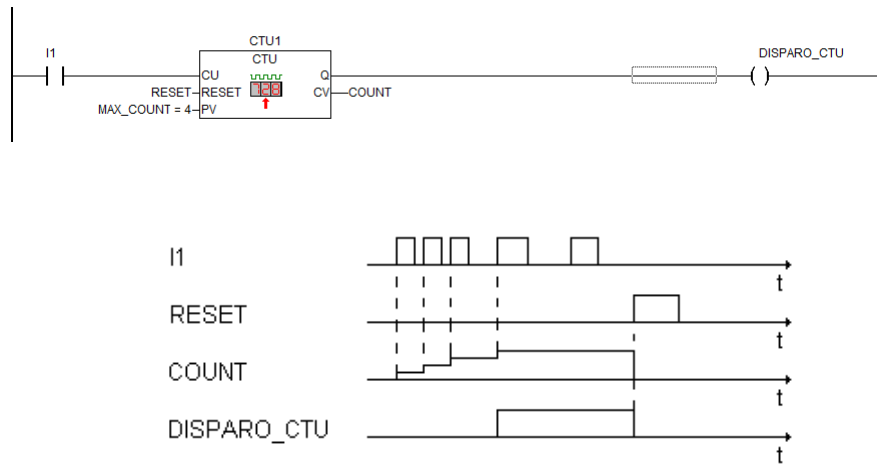
## Contadores



- n Entrada CU (bit): "Pulso", se conecta al tren de pulsos que se cuentan
- n Entrada RESET (bit): escribe 0 en acumulador
- n Entrada PV (Word): límite máximo de cuenta (Preset Value)
- n Salida Q (bit): "Done", indica si acumulador  $\geq$  PV
- n CV (Word) = cuenta, acumulador (Current Value)



## Contadores: Ejemplo



## Contadores: Ejemplo

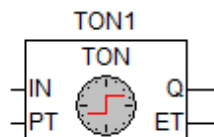
- n Observación: el contador se incrementa a intervalos de tiempo variables (dependen del pulso I1)

## Timers

- n Timer: instrucción destinada a esperar cierto tiempo antes de una acción
- n Tres tipos de timers:
  - n TON: Timer On Delay (retardo en el encendido): luego que la entrada pasa a 1 durante X seg, la salida pasa a 1.
  - n TOF: Timer Off Delay (retardo en el apagado): luego que la entrada pasa 0 durante X seg, la salida pasa a 0.
  - n Timers retentivos: no existen en los PLCs de lab. Cuenta el tiempo que la entrada es 1, congelando la cuenta con cambios de 1 a 0.

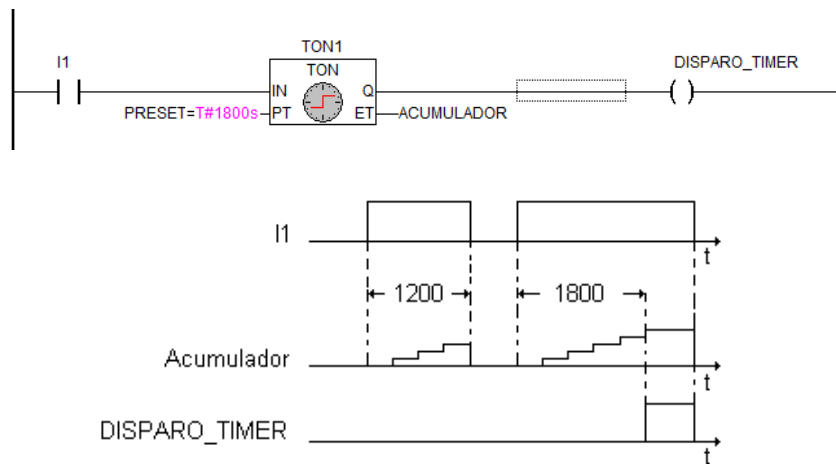
## Timers: TON

- n Timer On Delay TON:

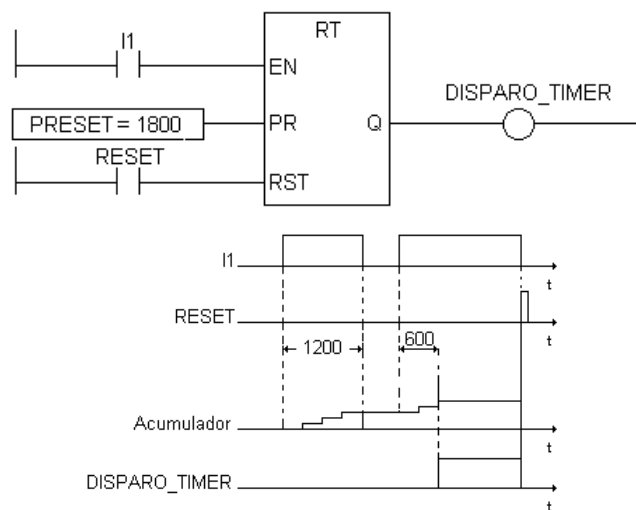


- n Entrada IN (bit): "Habilitación", se conecta al pulso que el timer retarda
- n Entrada PT (time): "Preset", determina valor del acumulador para el que se ejecuta acción del timer
- n Salida Q (bit): "Done", indica la expiración del tiempo de retardo
- n Salida ET (time): "Elapsed Time", la cuenta del tiempo transcurrido

## Timers: Ejemplo TON



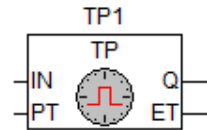
## Timers: Ejemplo Retentivo





## Timers: TP

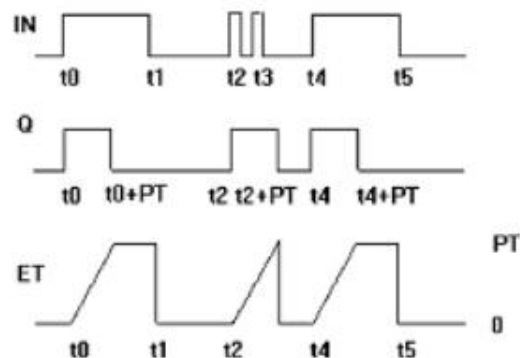
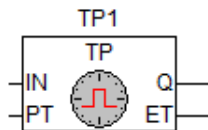
n Timer de Pulso TP:



- n Entrada IN (bit): entrada que genera el pulso cuando pasa a 1
- n Entrada PT (time): "Preset", duración del pulso de salida
- n Salida Q (bit): se mantiene en 1 durante el tiempo PT luego de 1 en la entrada
- n Salida ET (time): "Elapsed Time", la cuenta del tiempo transcurrido



## Timers: Ejemplo TP





## Timers

---

- n La cuenta del timer es independiente de la ejecución del programa (a cargo del sistema operativo)
- n La cuenta del contador SI depende de la ejecución del programa



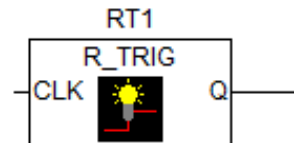
## Timers

---

- n Precisión
  - n Demora en la entrada: 1 tiempo de ciclo
  - n Demora en ejecución del timer: 1 tiempo de ejecución (máx 1 ciclo)
  - n Demora en la salida: 1 tiempo de ciclo
  - n Total: 3 tiempos de ciclo
  - n Ejemplo: tiempo de ciclo de 5 mseg lleva a error de 15 mseg
- n Retardos en filtrado y en la electrónica

## Triggers: R\_TRIG

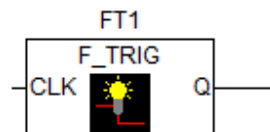
- n Trigger Ascendente (Rising):



- n Detección de flanco ascendente
- n Entrada CLK (bit): activa la salida cuando pasa a 1 (flanco)
- n Salida Q (bit): se mantiene en 1 durante 1 ciclo de ejecución ante flanco en CLK, luego pasa a 0

## Triggers: F\_TRIG

- n Trigger Ascendente (Falling):



- n Detección de flanco descendente
- n Entrada CLK (bit): activa la salida cuando pasa a 0 (flanco)
- n Salida Q (bit): se mantiene en 1 durante 1 ciclo de ejecución ante flanco en CLK, luego pasa a 0



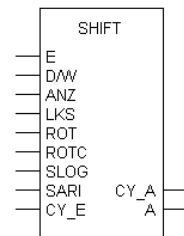
## Shift Registers

- n Instrucción que almacena eventos en área de memoria por desplazamiento o rotación de los bits individuales
- n En el PLC del laboratorio, se denomina "SH\_L" o "SH\_R"



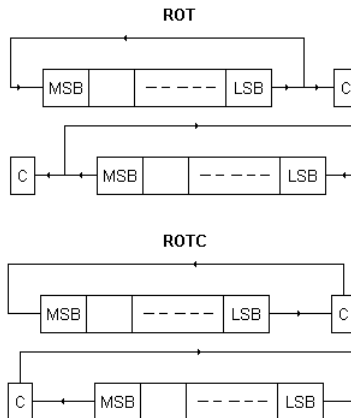
## Shift Registers

- n Entradas de instrucción SHIFT:
  - n E (W o DW): operando (almacena los eventos)
  - n D/W (bit): tipo de dato E
  - n ANZ (W): número de posiciones del movimiento
  - n LKS (bit): dirección
  - n ROT (bit): operación es rotación
  - n ROTC (bit): operación es rotación con CY\_E
  - n SLOG (bit): operación es desplazamiento
  - n SARI (bit): operación es desplazamiento aritmético
  - n CY\_E (bit): valor inicial de carry flag
- n Salidas de instrucción SHIFT:
  - n Salida CY\_A (bit): valor final de carry flag
  - n Salida A (W o DW): resultado de la operación



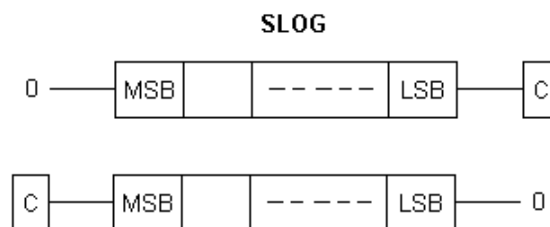
## Shift Registers

n Operaciones ROT/ROTC de SHIFT:



## Shift Registers

n Operación SLOG de SHIFT:



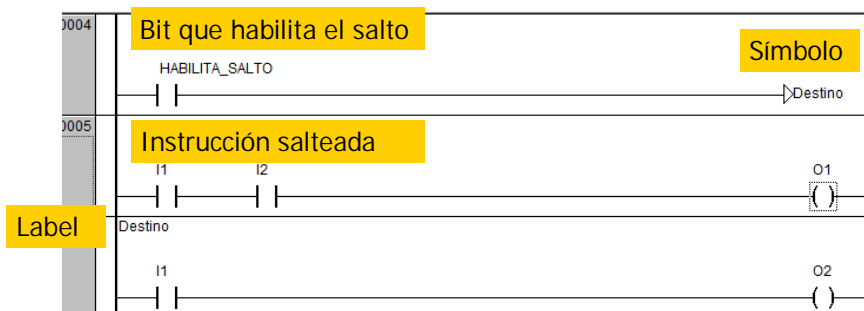


## Control de Flujo

- n JSR (Jump to subroutine)
- n En PLC de laboratorio, salto condicional a valor TRUE de un bit, definido por:
  - n Jump symbol: símbolo asociado a instrucción JUMP
  - n Bit que define el salto
  - n Jump label: posición donde salta (siempre hacia adelante)

## Control de flujo

- n Esquema instrucción JUMP PLC lab:





## Otras Instrucciones

- n Existen muchas instrucciones en forma de bloques funcionales (varias se estudian en otras partes del curso)
- n Desde el punto de vista del programa LADDER se clasifican en:
  - n Instrucciones de entrada: evalúan si el escalón es verdadero o falso
  - n Instrucciones de salida: se ejecutan según resultado del escalón



## Otras Instrucciones

- n Ejemplos de instrucciones de entrada:
  - n Instrucciones de comparación: Igual (EQ), Mayor (GT), etc. Salida verdadera o falso en función de comparación de entradas (tipo Bool, Word, etc.)



## Otras Instrucciones

- n Ejemplos de instrucciones de salida:
  - n Operaciones aritméticas o lógicas: ADD, AND, MUL, etc. Entradas y salida tipo Word, Real, etc.
  - n Operaciones de movimiento de memoria: permiten copiar áreas de memoria
  - n Funciones de control PID
  - n Funciones de comunicaciones: permiten intercambio de mensajes entre PLCs



## Ejemplo

Se desea escribir un programa que controle el encendido - apagado de una bomba.

La bomba será encendida si:

- 1) Se pulsa el botón de arranque.
- 2) La protección térmica está deshabilitada.
- 3) Está abierto el botón de alarma.
- 4) Está abierto el botón de parada.

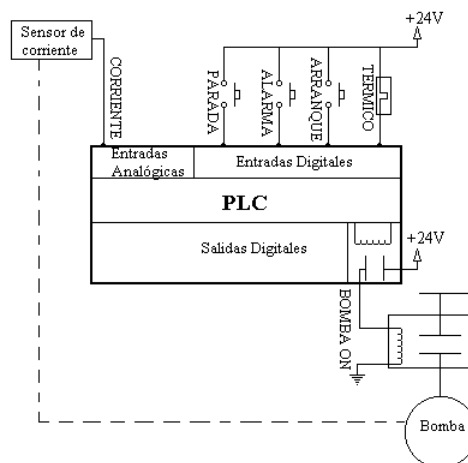
Desde un tiempo T después del encendido, no puede haber ni sobre corriente ni baja corriente. Expresado de otra forma, desde un tiempo T después del arranque, la corriente I debe cumplir  $I_{\text{MIN}} < I < I_{\text{MAX}}$ , siendo  $I_{\text{MIN}}$  e  $I_{\text{MAX}}$  límites prefijados.

## Ejemplo

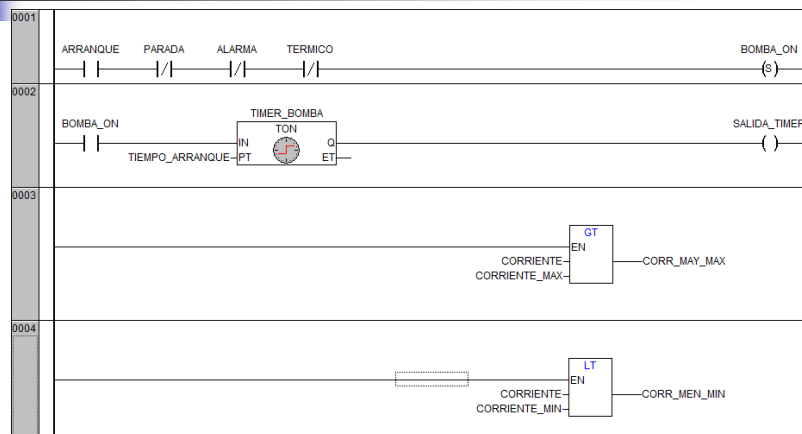
El motor de la bomba se apagará si:

- 1) Se pulsa el botón de parada.
- 2) Se cierra la protección térmica.
- 3) Se pulsa el botón de alarma.
- 4) Los límites de corriente no son los correctos.

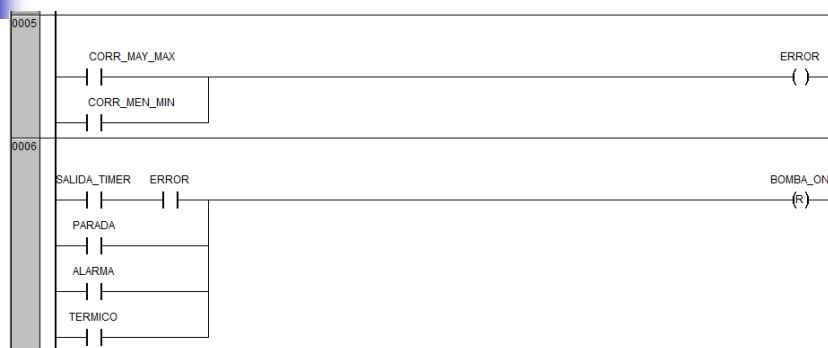
## Conexiones al PLC



## Ejemplo – Parte 1



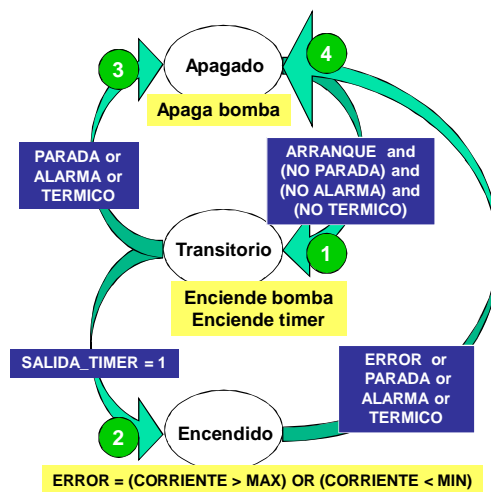
## Ejemplo – Parte 2



## Diagrama de Estados

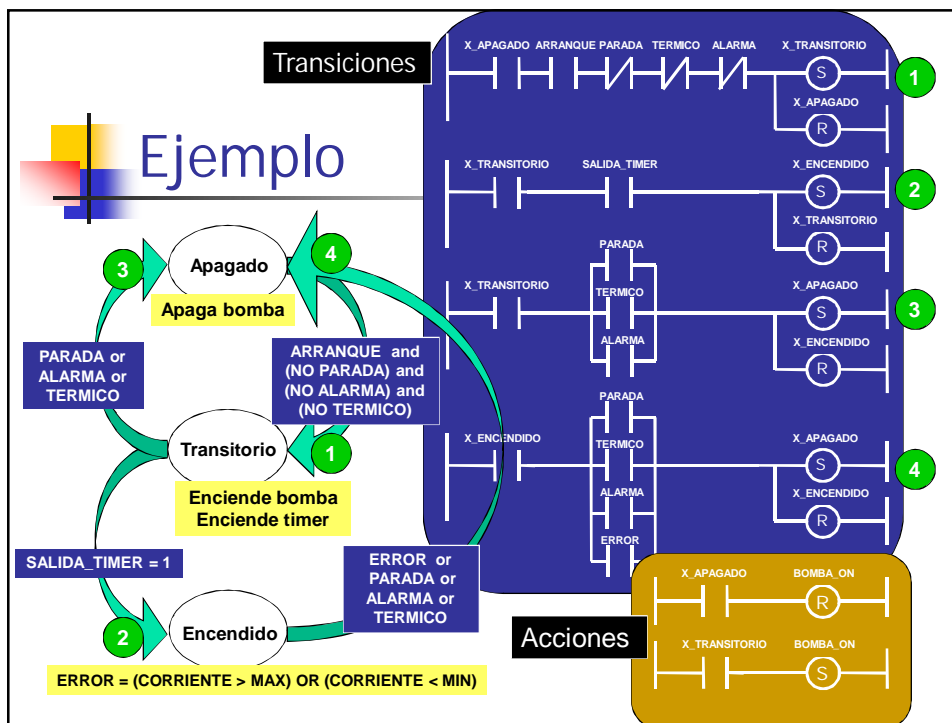
- n STD – State Transitions Diagram
- n No es un lenguaje
- n Es una metodología para representar cierta lógica de funcionamiento en base a estados y transiciones

## Ejemplo



## Diagrama de Estados en LD

- n Método para convertir un diagrama de estados al lenguaje LD





## Ventajas / Desventajas

- ↑ n Método sistemático
- ↑ n Minimiza/elimina errores posteriores
- ↑ n LD disponible en todos los PLCs
- ↓ n Poco intuitivo, difícil de comprender sin el diagrama de estados asociado
- ↓ n Cambios requieren estudio del diagrama previamente