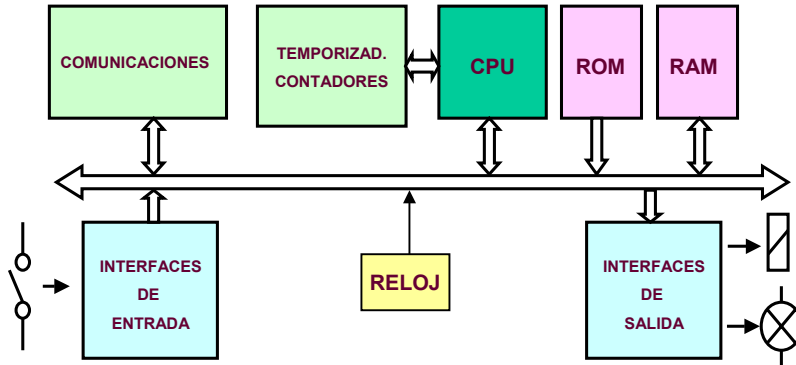


Autómata Programable (PLC)

Hardware del autómata

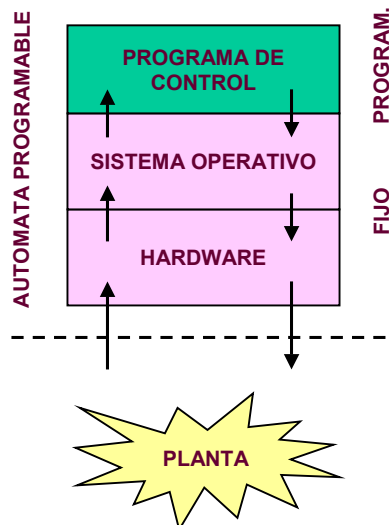
- **Sistema digital basado en un microprocesador**
- **CPU**
- **Reloj**
 - Marca el funcionamiento de todo el sistema (20 Mhz).
- **Memoria no volátil (ROM)**
 - Parámetros fijos, sistema operativo, drivers para el hardware.
- **Memoria volátil (RAM)**
 - Datos y programa de usuario.
- **Interfaces de entrada/salida**
 - Adaptadas para enviar y recibir señales desde la planta.
- **Fuente de alimentación**
- **Comunicaciones**
 - Serie
- **Temporizadores, contadores.**

Diagrama de bloques del autómata programable

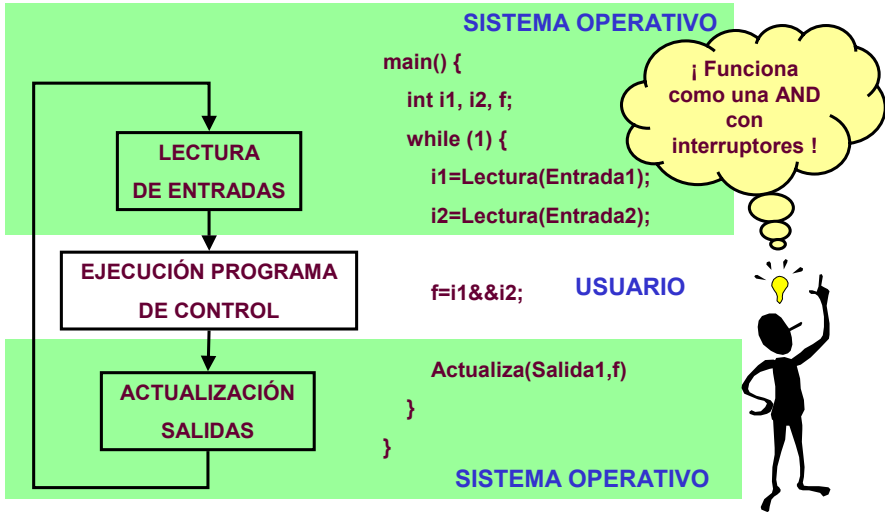


Software del autómata

- **Sistema Operativo**
 - Residente en ROM
 - Gobierna el hardware del autómata
 - Permite la programación del autómata.
 - Facilita la comunicación del programa de control con el hardware
- **Programa específico de control.**
 - Secuencia de operaciones a realizar sobre la planta.

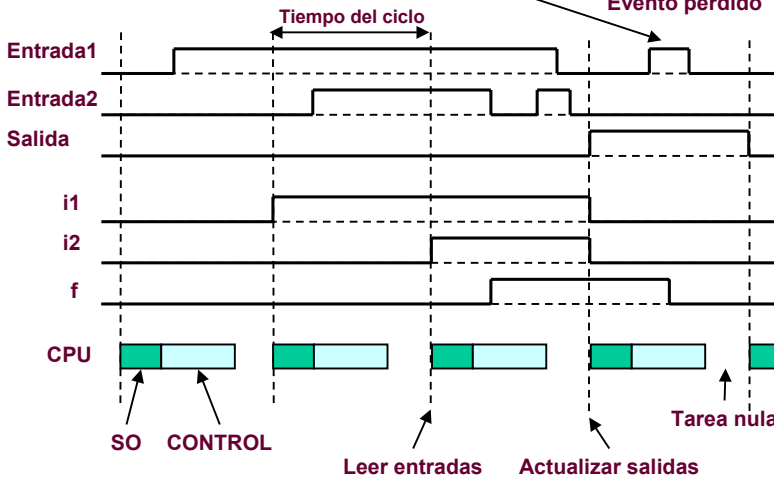


Ciclo simple de funcionamiento de un autómata

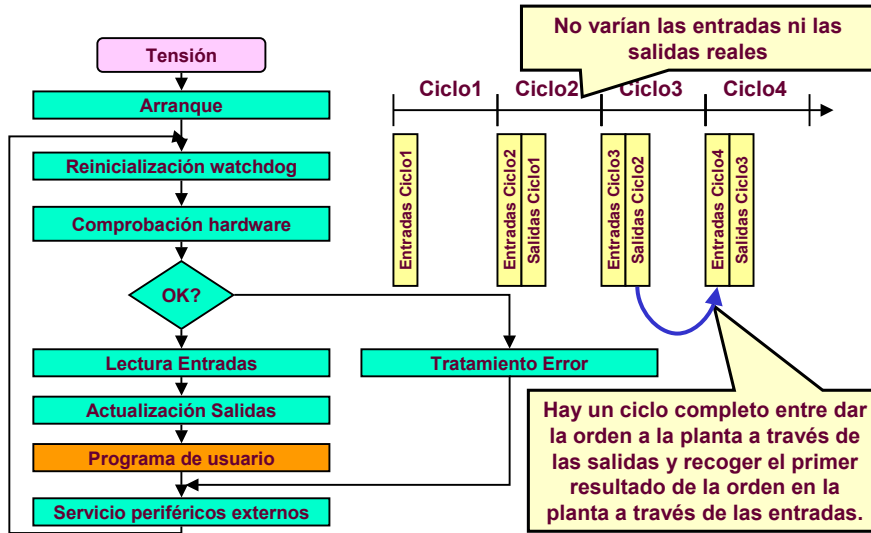


INTERACCIÓN ENTRE AUTÓMATA Y MUNDO REAL

¡¡ Sistema en el límite de funcionamiento!!

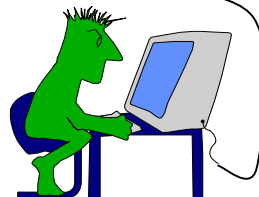
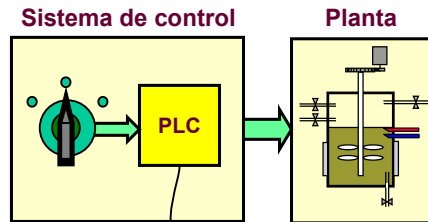


Ciclo de funcionamiento del autómat



¿Cómo programar el PLC para controlar la planta?

- **Lenguaje de programación**
 - Instrucciones para realizar operaciones lógicas
 - Variables para almacenar resultados intermedios
 - Variables que actualicen con el valor de las entradas
 - Variables conectadas a las salidas.
 - Orientado al bit (tamaño de la variable lógica).
- **Fácil de programar**
 - No se necesita un gran cualificación para programar automatismos sencillos
- **Flexible**
 - Programar controles complejos

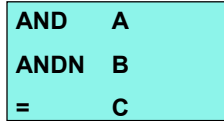


Comunicación serie (RS-232, RS-485)

Sistemas básicos de programación de los PLC's

NORMA IEC 1131-3

Lista de instrucciones



Programación estructurada

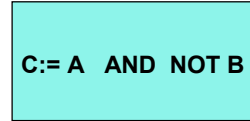


Diagrama de funciones

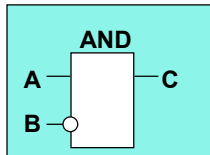
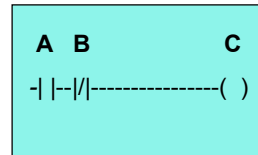
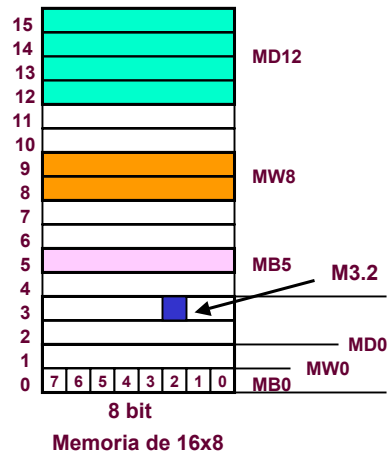


Diagrama de contactos



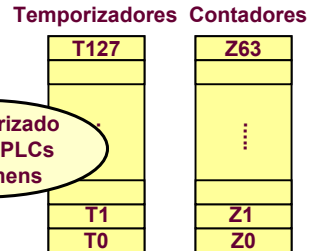
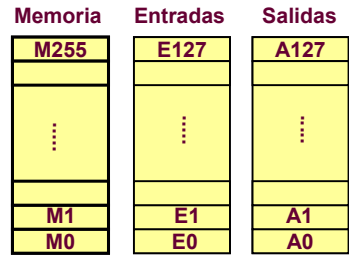
Modelo de memoria del PLC para su programación

- La memoria está organizada como una matriz:
 - Cada fila tiene un número asociado: la **dirección**, que se corresponde con el ordinal de la fila.
 - Cada columna está dividida en bits. El tamaño típico son 8 bits (1 byte).
- Métodos para acceder a la información en memoria:
 - Bit (marca): M0.3
 - Trabajar con variables lógicas
 - Byte: MB0
 - Word: MW0, MW2.
 - Long word: MD0, MD4
- Son las variables del programa



Tipos básicos de datos (Variables) en un PLC

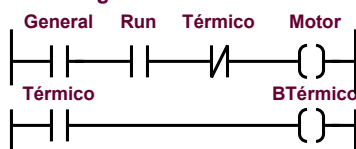
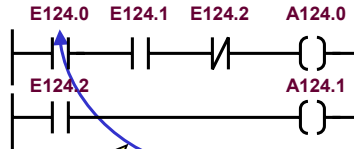
- **Memoria (marcas)**
 - M0.1, MB3, MW4, MD0
 - Variables de propósito general
- **Entradas digitales**
 - E124.0, EB124, EW124
 - Variables que se actualizan al inicio del ciclo con las entradas reales si existen (**imagen de proceso de las entradas**).
- **Salidas digitales**
 - A124.0, AB124, AW124
 - Variables que se escriben en las salidas (**imagen de proceso de las salidas**).
- **Temporizadores**
 - T1 (todos 16 bits)
 - Variables actualizadas a ritmo de reloj.
- **Contadores**
 - Z5 (todos 16 bits)



Particularizado para los PLCs de Siemens

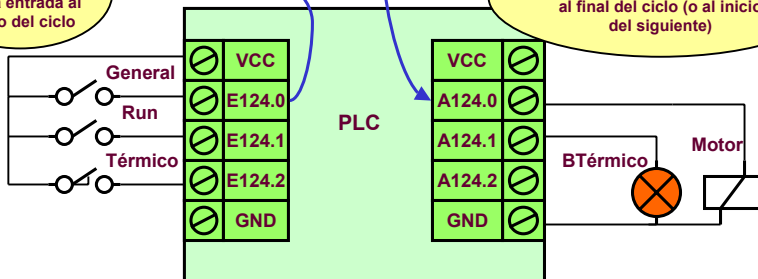
Programación en Diagrama de Contactos

Programa con símbolos



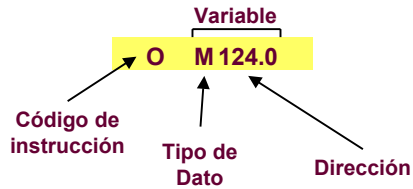
La posición de memoria se actualiza con la entrada al inicio del ciclo

La salida se actualiza con la posición de memoria al final del ciclo (o al inicio del siguiente)



Programación con Lista de instrucciones

- **Formato de la instrucción:**



- **Tipos de instrucciones:**

- Booleanas:
 - Sólo trabajan con bits
 - Instrucciones lógicas: **O,ON,U(A),UN(AN),X,XN**
 - Instrucciones de asignación **=,S,R**
 - Se pueden utilizar paréntesis
 - Trabajan con stack (como las HP)
- Numéricas:
 - Trabajan con byte, word y long word.
 - Es un nivel intermedio entre ensamblador y alto nivel.

Norma IEC1131-3

* Entre paréntesis se indica la codificación en norma IEC.

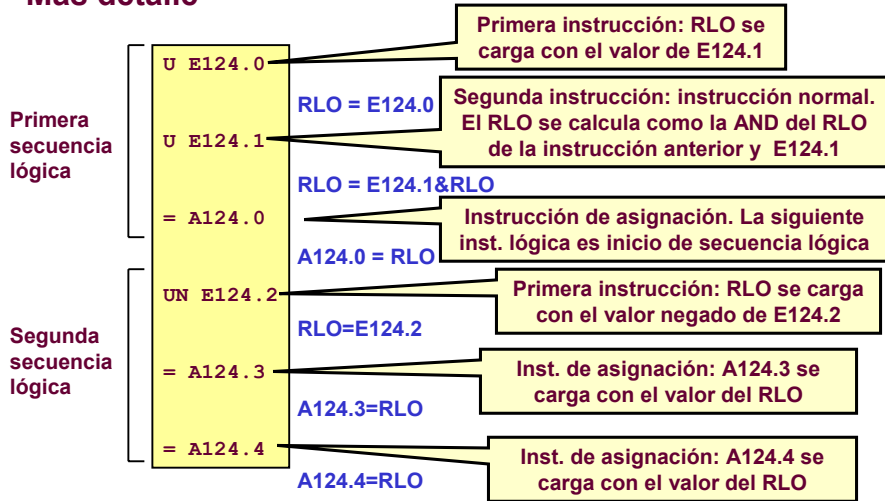
*Se utiliza codificación Siemens para ser compatibles con los PLC's que hay en el laboratorio.

Mecánica de las instrucciones lógicas: RLO

- Cada instrucción lógica opera el RLO obtenido en la instrucción anterior con el operando de la instrucción
- RLO almacena el resultado de la última instrucción lógica ejecutada
- Inicio de secuencia lógica:
 - La primera instrucción lógica de una secuencia lógica no puede operar con el RLO anterior.
 - El RLO resultado de la instrucción lógica es simplemente cargar el valor lógico del operando en el RLO. Si la instrucción es de negación, se carga el operando negado en RLO.
- ¿Cuándo se inicia una nueva secuencia lógica?
 - Primera instrucción de un programa o segmento.
 - Después de una instrucción de asignación (=,S,R)

U E124.0	RLO = E124.0
U E124.1	RLO = E124.1&RLO
= A124.0	A124.0 = RLO
UN E124.2	RLO = E124.2'
= A124.3	A124.3 = RLO
= A124.4	A124.4 = RLO

Más detalle

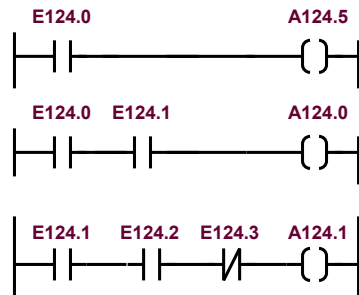


Función AND

• **Función AND**

- U (A)
- UN (AN): AND del RLO anterior con el dato negado
- Si está al inicio de una cadena lógica el RLO se carga con el valor del dato (U) o su negado (UN).

U	E124.0
=	A124.5
U	E124.0
U	E124.1
=	A124.0
U	E124.1
U	E124.2
UN	E124.3
=	A124.1

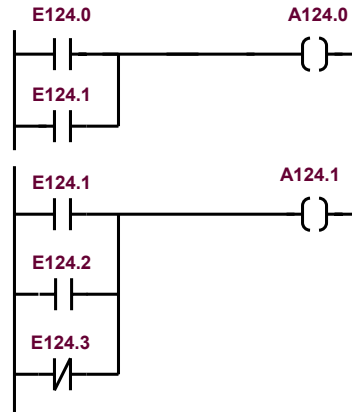


Función OR

- **Función OR**

- O
- ON: OR del RLO anterior con el dato negado
- Si está al inicio de una cadena lógica el RLO se carga con el valor del dato (O) o su negado (ON).

O	E124.0
O	E124.1
=	A124.0
O	E124.1
O	E124.2
ON	E124.3
=	A124.1

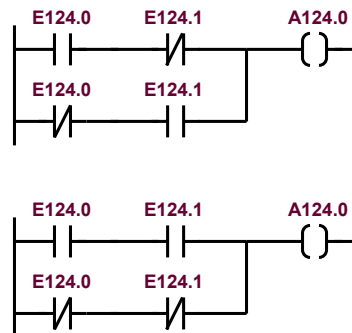


Función XOR

- **Función OR**

- X
- XN: XOR del RLO anterior con el dato negado

X	E124.0
X	E124.1
=	A124.0
X	E124.1
XN	E124.2
=	A124.1

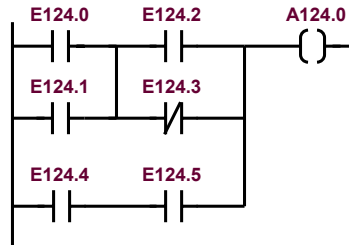


Paréntesis

- Funciona como una calculadora HP
 - U(
 - O(
 -)
- La instrucción con paréntesis manda el RLO actual al stack.
- La instrucción de cierra paréntesis saca el RLO último introducido en el stack y lo opera (según la instrucción que envió éste al stack) con el RLO actual.

```

U (
O   E124.0
O   E124.1
)
U (
O   E124.2
ON  E124.3
)
O (
U   E124.4
U   E124.5
)
=   A124.0
    
```

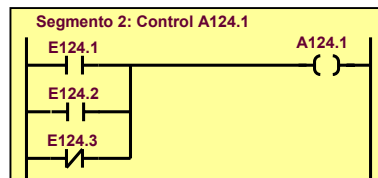
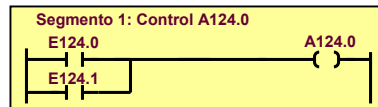


Organización básica de un programa

- Segmentos
 - Los programas se dividen en bloques de instrucciones o segmentos relativamente independientes
- Segmentos en lista de instrucciones
 - No hay limitación en el tamaño. No se recomienda más de 20 instrucciones
 - Pueden haber varias secuencias lógicas
- Segmentos en diagrama de contactos
 - Desde el punto de vista de lista de instrucciones sólo puede haber una secuencia lógica.
- Organización más compleja
 - Funciones
 - Programación estructurada

```

Segmento 1: Programa
O   E124.0
O   E124.1
=   A124.0
O   E124.1
O   E124.2
ON  E124.3
=   A124.1
    
```



Orden de ejecución para lista de instrucciones

- **Se ejecutan todas las instrucciones una tras otra:**
 - Según el orden de los segmentos
 - Según el orden de aparición en cada segmento.
- **Recuerde que las variables de entrada no se actualizan mientras se ejecuta el programa. Lo mismo ocurre con las salidas reales.**
 - Durante la ejecución del programa, tanto las variables de entrada como de salida son variables que podemos cambiar su valor en cualquier momento.
 - En las variables de salida, sólo el último valor asignado es el que se carga en la salida real.

Segmento 1: Programa		
O	E124.0	1
O	E124.1	2
=	A124.0	3
O	E124.1	4
O	E124.2	5
ON	E124.3	6
=	A124.1	7

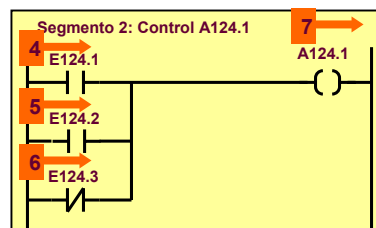
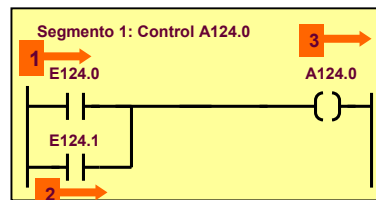
Orden de ejecución

U	E124.0
=	A124.0
U	E124.1
=	E124.0

Anula el resultado de la asignación anterior

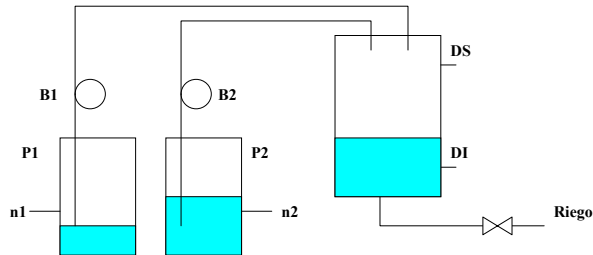
Orden de ejecución para diagrama de contactos

- **Se ejecuta en el mismo orden que su lista de instrucciones equivalentes**
- **Los contactos paralelos no significan instrucciones en paralelo**
 - De izquierda a derecha
 - Si al ir de izquierda a derecha hay ramas paralelas antes de pasar a la derecha, se evalúan las ramas paralelas de arriba abajo (Puede ser de abajo a arriba).
 - Si el diagrama de contactos representa directamente un circuito eléctrico (no es el caso del PLC), entonces todo trabaja en paralelo.



Ejemplo simple de automatización con PLC (I)

- Se quiere controlar un sistema de riego, que consta de dos pozos (P1 y P2), dos bombas (B1 y B2) y un depósito. Mediante las bombas B1 y B2 se controla el nivel del depósito. El depósito tiene dos boyas (DI y DS) para indicar el nivel. Cada pozo lleva instalado un sensor de nivel (n1 y n2) para saber si hay agua suficiente. Si no hay agua suficiente en el pozo, la bomba correspondiente no debe funcionar.



Ejemplo simple de automatización con PLC (II)

- **El sistema debe funcionar de la siguiente manera:**
 - Si el nivel del depósito supera la boya DS, las bombas están paradas.
 - Si el nivel del depósito está entre la boya DI y la DS, funciona la bomba B1, si hay agua suficiente en el pozo 1. Si no hay agua en el pozo 1 pero la hay en el 2, funciona la bomba B2.
 - Si el nivel del depósito está por debajo de la boya DI, se activa la bomba B2, además de la B1 si es posible.
- **Pasos para el diseño de la automatización**
 - Comprobar la especificación
 - Entender cómo debe funcionar la planta
 - Lista de entradas/salidas
 - Saber cuál es la lógica: sensores de nivel (1- supera nivel)
 - Diseño de las ecuaciones lógicas que deben regir la planta según la especificación
 - Implantación mediante el PLC

Ejemplo simple de automatización con PLC (III)

- Lista de entradas/salidas

ENTRADAS	LOGICA	ETIQUETA	DIRECCIÓN
Nivel Pozo 1	P	EN1	E124.0
Nivel Pozo 2	P	EN2	E124.1
Nivel Superior Depósito	P	EDS	E124.2
Nivel Inferior Depósito	P	EDI	E124.3

SALIDAS	LOGICA	ETIQUETA	DIRECCIÓN
Bomba Pozo 1	P	SB1	A124.0
Bomba Pozo 2	P	SB2	A124.1

- Ecuaciones lógicas
 - $SB1 = EN1 \cdot EDS'$ (primero las condiciones de parada)
 - $SB2 = EN2 \cdot (EDI' + EDS' \cdot EN1')$

Ejemplo simple de automatización con PLC (IV)

Lista de instrucciones

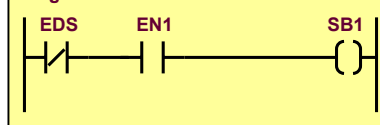
Segmento 1: Programa

```

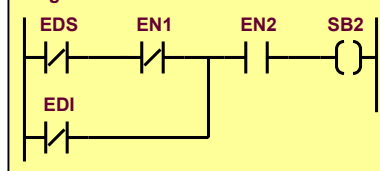
UN  EDS
U   EN1
=   SB1
ON  EDI
O(
UN  EDS
UN  EN1
)
U   EN2
=   SB2
  
```

Diagrama de contactos

Segmento 1: Control Bomba 1



Segmento 1: Control Bomba 2



Pasos para realizar la programación

- **1. Especificación del control:** determinar qué debe hacer el sistema de control y en qué orden.
- **2. Identificar las señales de entrada/salida de la planta a controlar.**
- **3. Representar mediante un modelo el sistema de control, indicando todas las funciones que intervienen, las relaciones entre ellas, y la secuencia que deben seguir**
 - Modelo algebraico: Boole.
 - Modelo gráfico: Grafcet, máquina de estados (¿CÓMO?).
- **4. Asignar direcciones de entrada/ salida o internas a cada uno de los componentes que aparecen en el modelo.**
- **5. Codificar la representación anterior según la herramienta de programación del PLC.**
- **6. Cargar programa en el PLC y verificar funcionamiento.**
- **7. Puede haber una fase de simulación sin la planta real.**