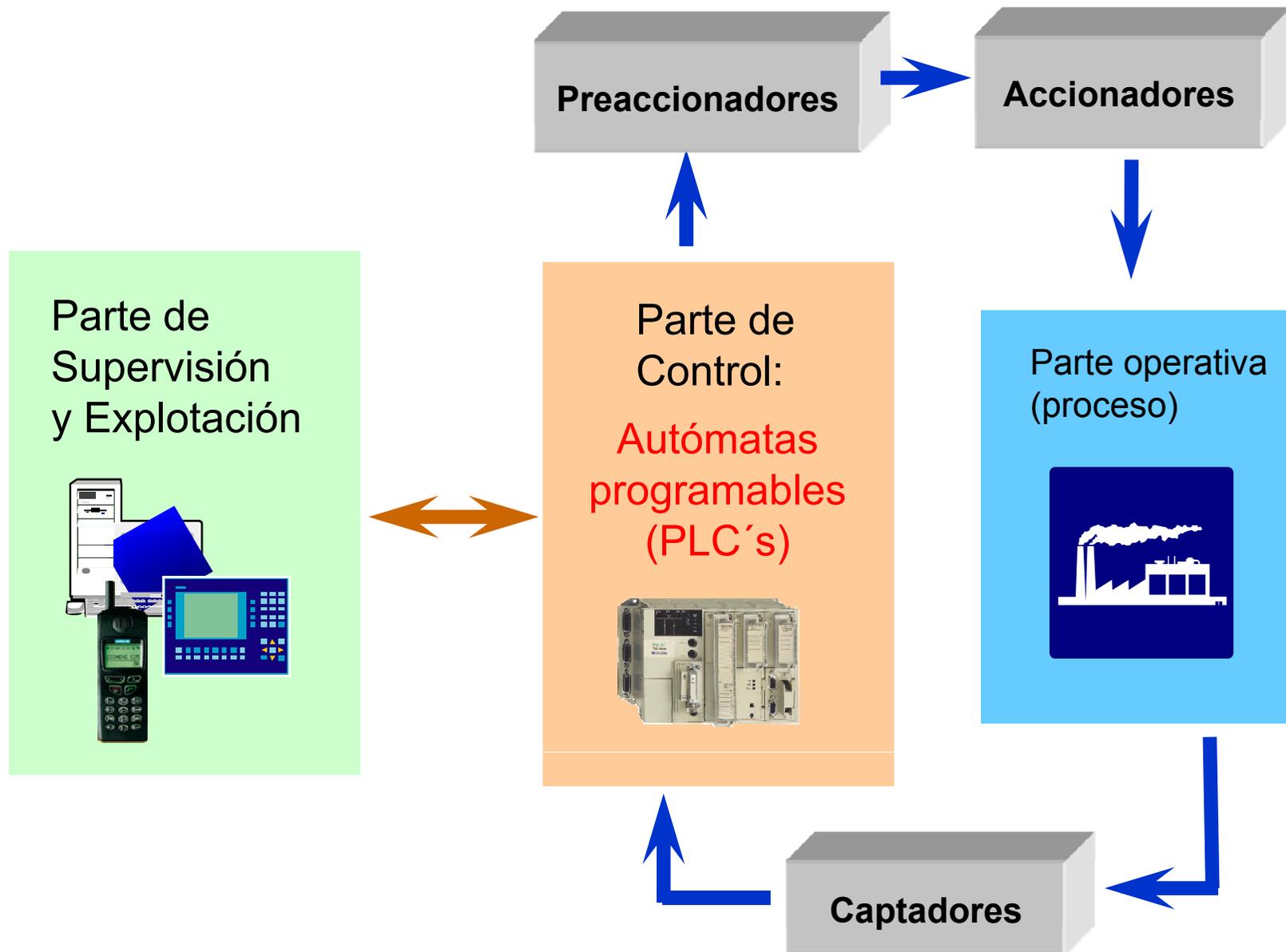


Universidad de Oviedo
Area de Ingeniería de Sistemas y Automática

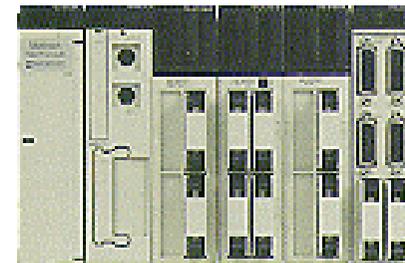
introducción a los autómatas

estructura de un sistema automatizado



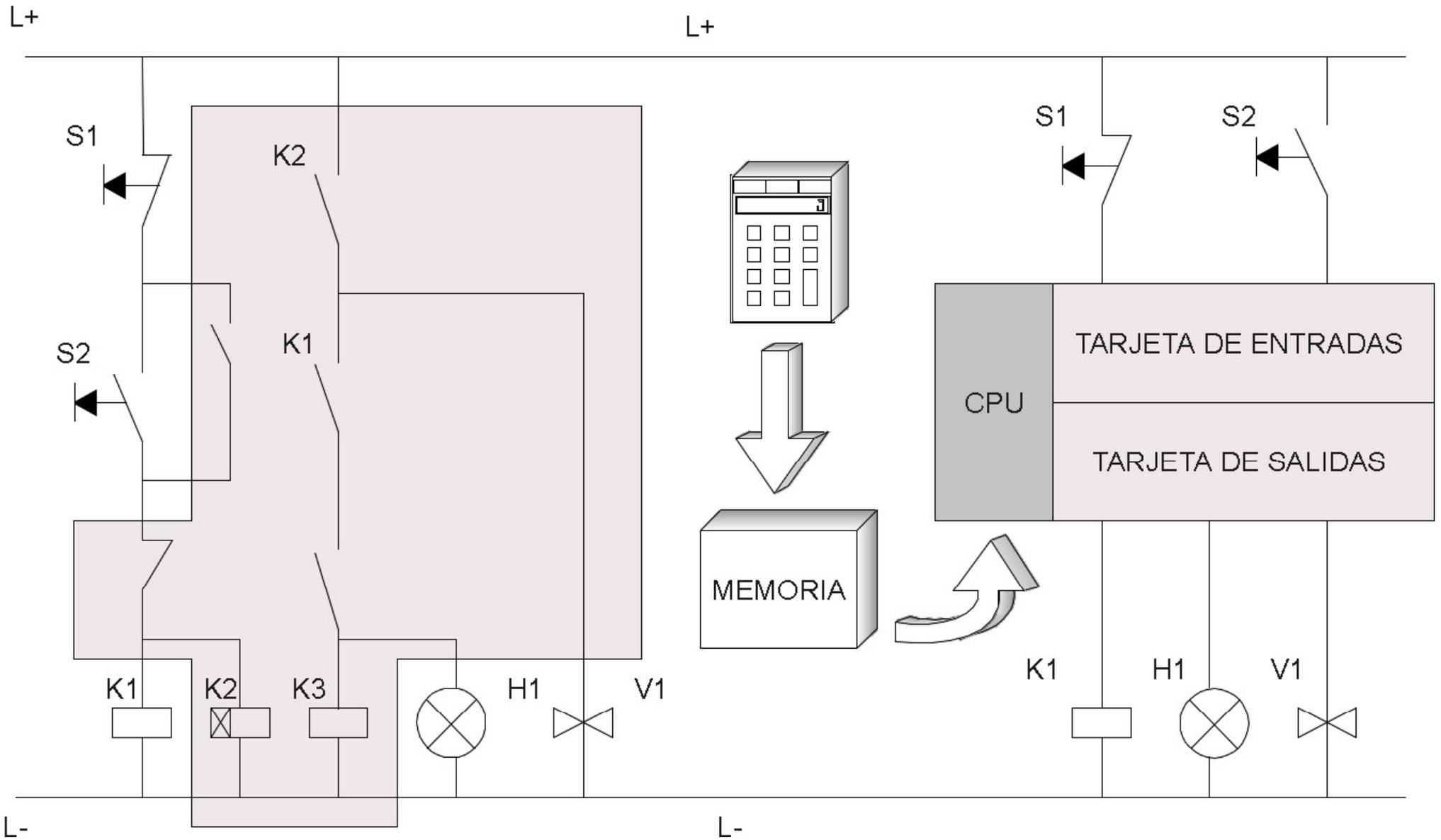
definición IEC 61131

un autómata programable (AP) es una máquina electrónica programable diseñada para ser utilizada en un entorno industrial (hostil), que utiliza una memoria programable para el almacenamiento interno de instrucciones orientadas al usuario, para implantar soluciones específicas tales como funciones lógicas, secuencias, temporizaciones, recuentos y funciones aritméticas, con el fin de controlar mediante entradas y salidas, digitales y analógicas diversos tipos de máquinas o procesos



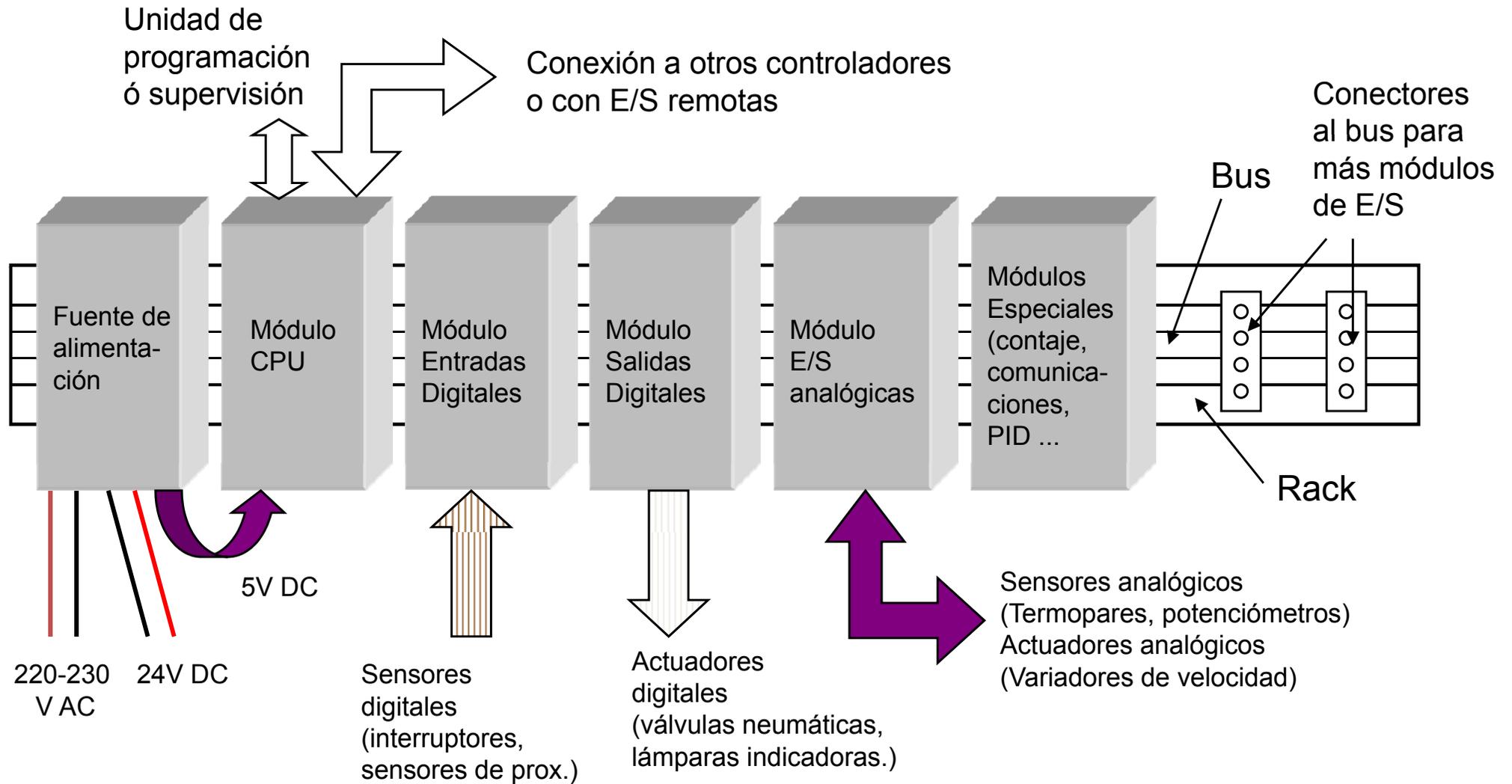
introducción a los autómatas

lógica cableada vs. lógica programada



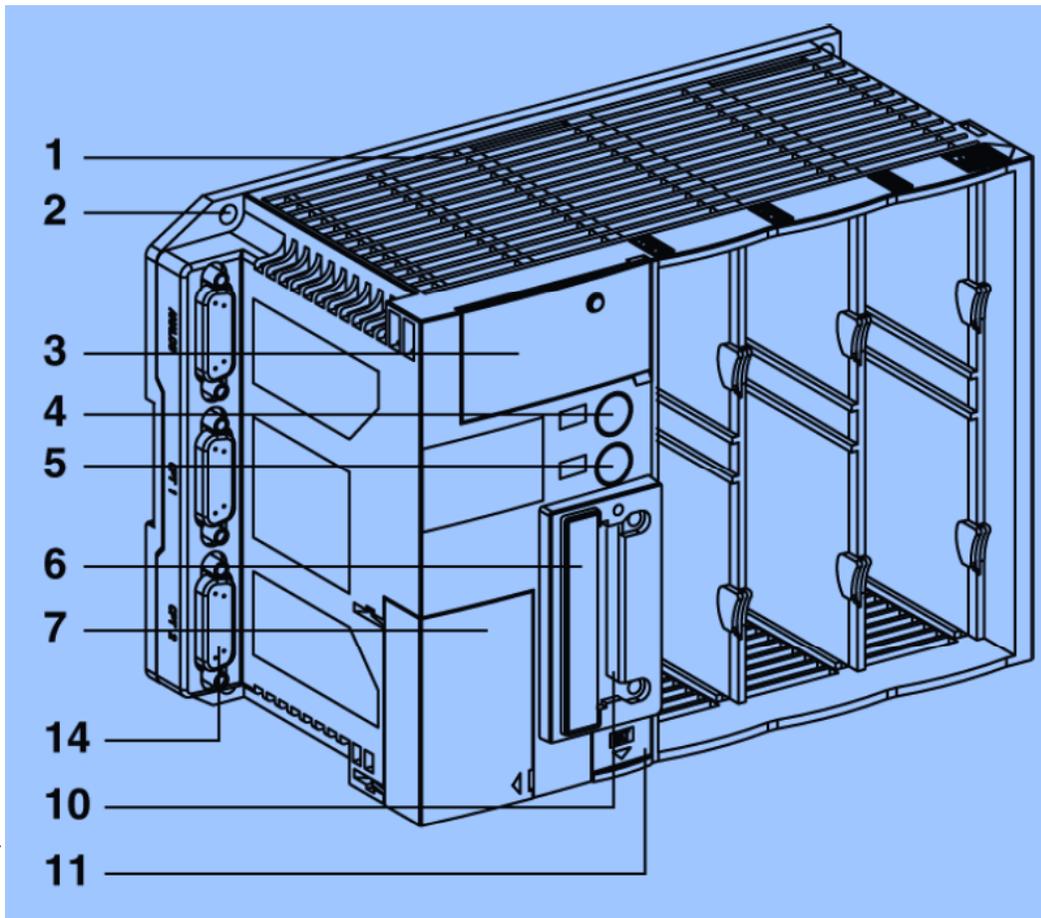
introducción a los autómatas

arquitectura típica de un autómata programable



introducción a los autómatas

arquitectura típica de un autómata programable



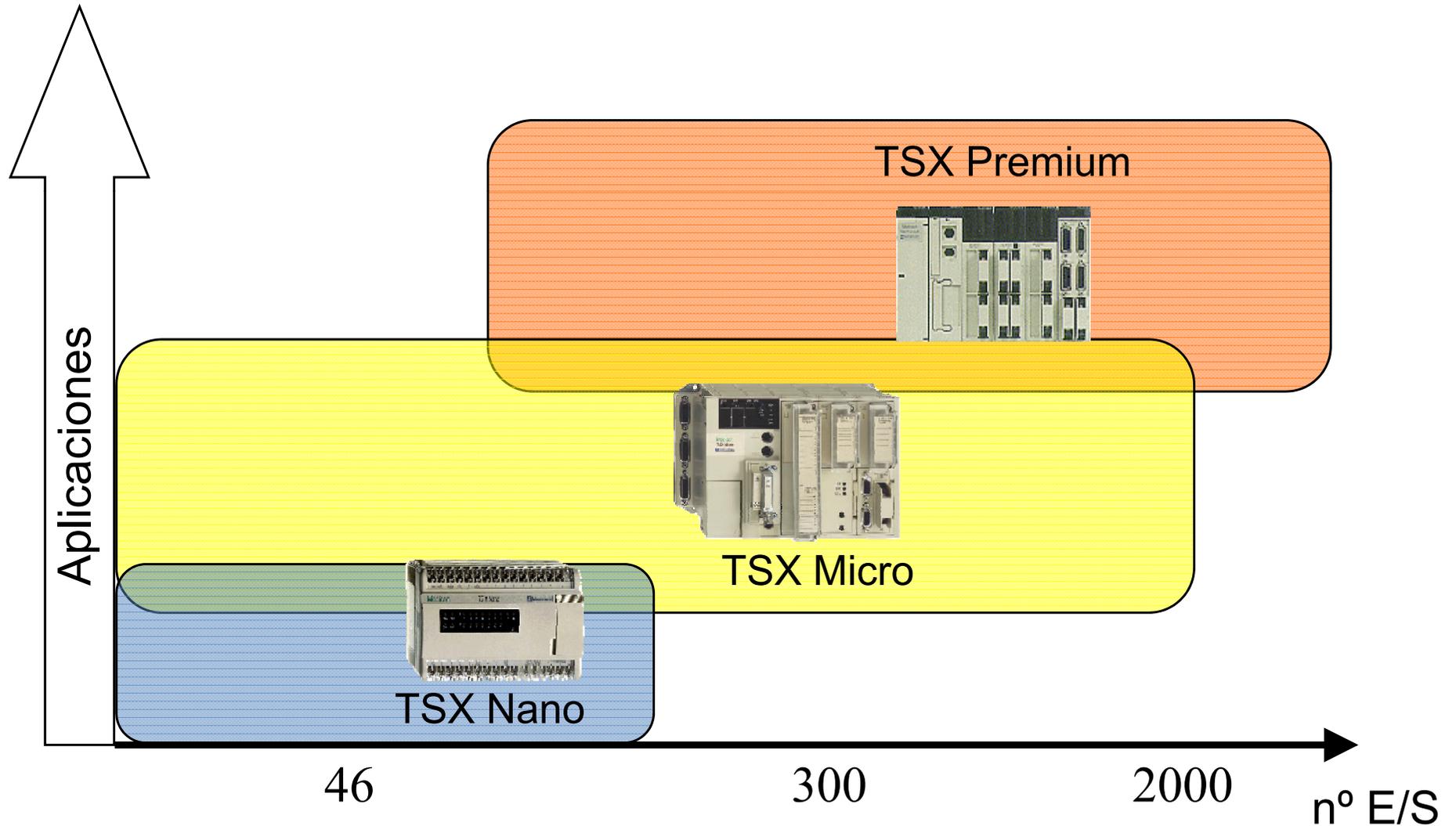
- 1 Rack de 3 emplazamientos, que integra la alimentación, el procesador y la memoria básica
- 2 Orificio de fijación del autómata
- 3 Bloque de visualización centralizado
- 4 Conector terminal TER
- 5 Terminal de diálogo con el operador AUX.
- 6 Emplazamiento para una tarjeta de extensión de memoria. Si no hubiera tarjeta, este emplazamiento se equipará con una tapa que es obligatoria, ya que su extracción provoca:• la parada del autómata,• la inactivación de la toma terminal
- 7 Trampilla de acceso a los terminales de alimentación
- 8 Etiqueta de información para el cambio de la pila
- 9 Terminales de alimentación
- 10 Emplazamiento para un acoplador de comunicación
- 11 Trampilla de acceso a la pila opcional y al conmutador de protección contra escritura del sistema operativo
- 12 Conector de conexión del mini-rack de extensión, protegido de serie por una tapa amovible
- 13 Dispositivo para el montaje en un perfil DIN
- 14 Conectores para las funciones analógicas y de conteo integradas



introducción a los autómatas

gama de autómatas TSX (Schneider)

Complejidad



elementos de programación

4 Lenguajes de programación + Grafcet (SFC)

- Lenguajes gráficos

 - Diagrama de escalera (“Ladder Diagram”, LD)

 - Diagrama de Bloques Funcionales (“Function Block Diagram, FBD)

- Lenguajes literales

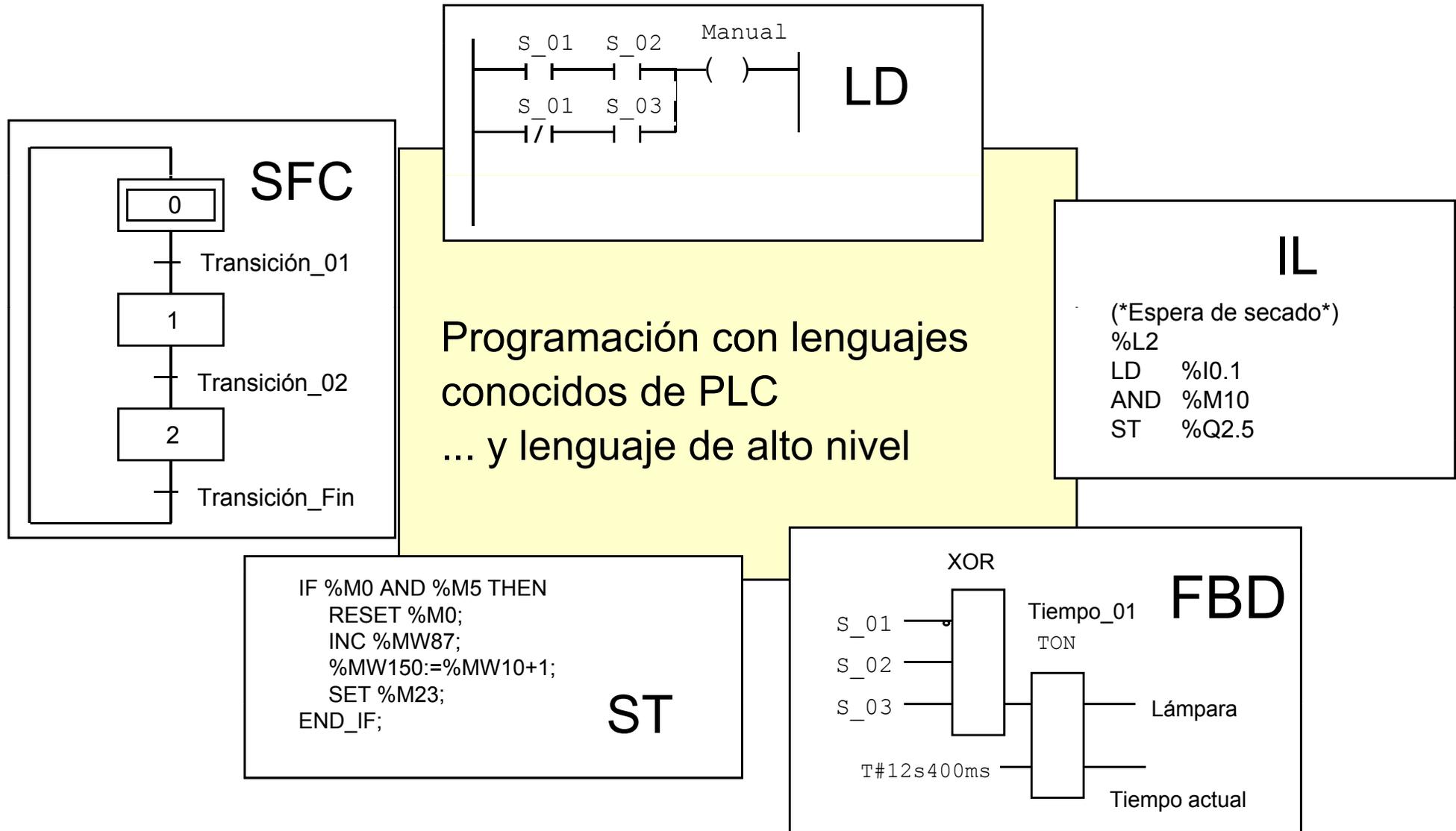
 - Lista de instrucciones (“Instruction List”, IL)

 - Texto estructurado (“Structured Text”, ST)

La selección del lenguaje de programación depende de la experiencia del programador, de la aplicación concreta, del nivel de definición de la aplicación, de la estructura del sistema de control y del grado de comunicación con otros departamentos de la empresa...

introducción a los autómatas

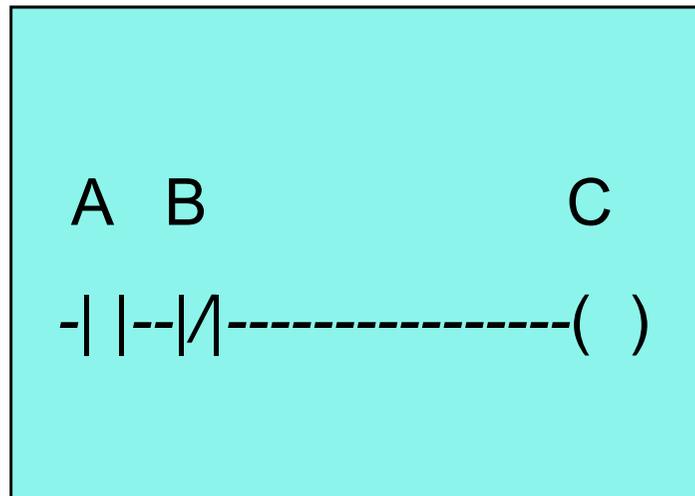
lenguajes de programación



introducción a los autómatas

lenguajes de programación · **diagrama de escalera (LD)**

- conjunto estandarizado de símbolos de programación de “relés en escalera”
- nace en los EEUU como reflejo de los esquemas eléctricos estándar empleados en lógica cableada



introducción a los autómatas

lenguajes de programación · lista de instrucciones (IL)

- modelo de ejecución basado en un acumulador (pila) simple
- basado en el lenguaje Alemán “Anweisungsliste”, AWL
- sólo se permite una operación por línea
- cada operación opera sobre uno o dos operandos

```
LD    A
ANDN  B
ST    C
```



introducción a los autómatas

lenguajes de programación · texto estructurado (ST)

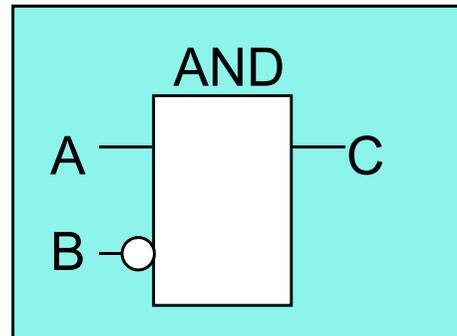
- lenguaje de alto nivel, estructurado en bloques
- sintaxis parecida a PASCAL
- posibilidad de utilizar expresiones complejas e instrucciones anidadas
- soporte para
 - bucles (REPEAT-UNTIL; WHILE-DO)
 - ejecución condicional (IF-THEN-ELSE; CASE)
 - funciones (SQRT(), SIN())

```
C:= A AND NOT B;
```

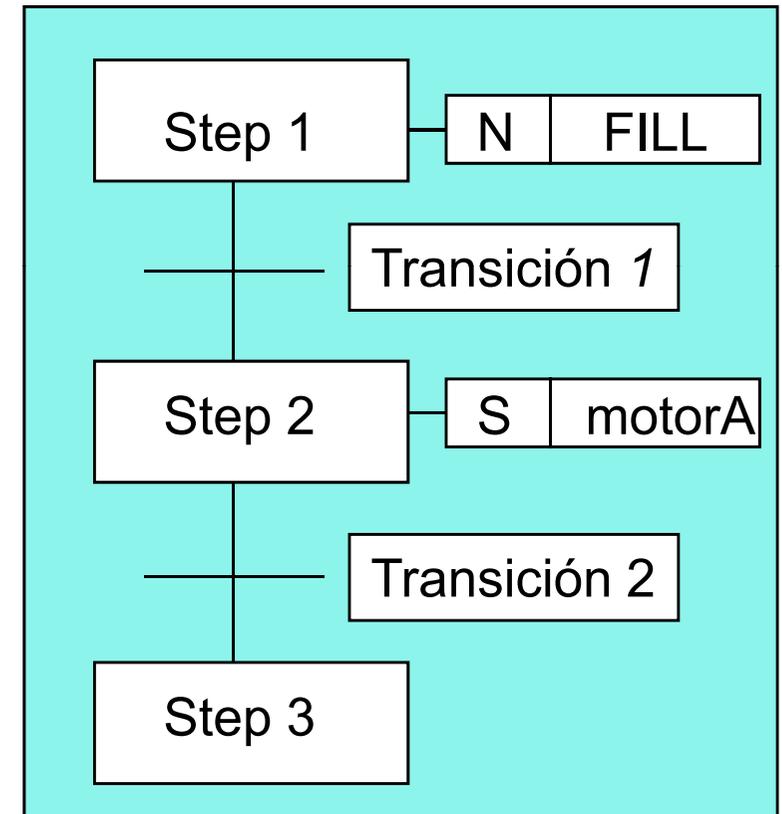
introducción a los autómatas

lenguajes de programación · **diagrama de bloques funcionales (FBD)**

- lenguaje gráfico ampliamente usado en Europa
- permite elementos de programa que aparecen como bloques conectados en una forma análoga a un diagrama de circuito electrónico de puertas lógicas
- se usa en muchas aplicaciones que implican el flujo de información o datos entre componentes de control

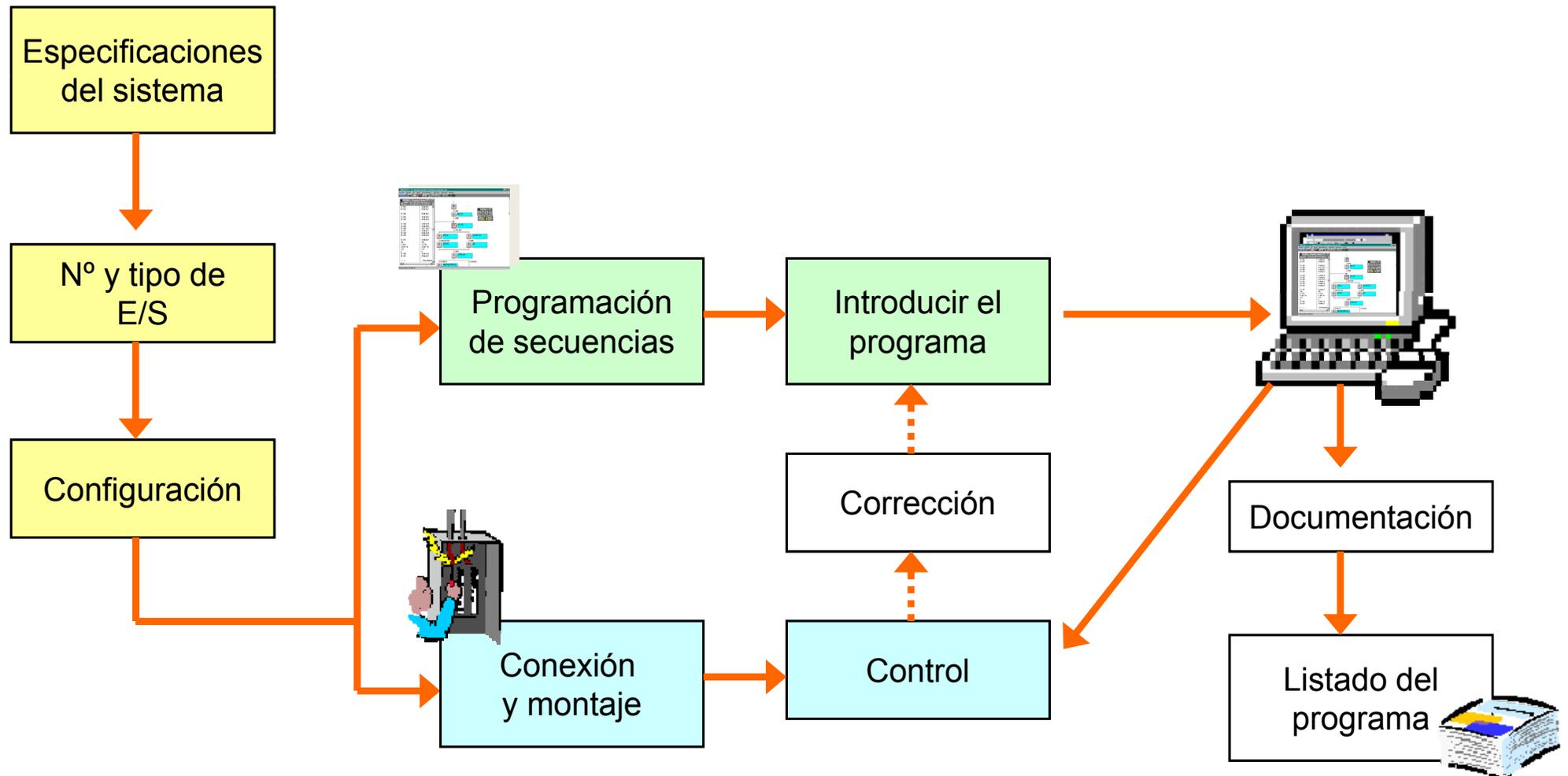


- potente técnica gráfica para describir el comportamiento secuencial de un programa de control
- se usa para particionar un problema de control
- facilita el rápido diagnóstico de problemas en el algoritmo de control
- los elementos son ETAPAS con BLOQUES DE ACCIÓN y TRANSICIONES
- permite secuencias alternativas y paralelas



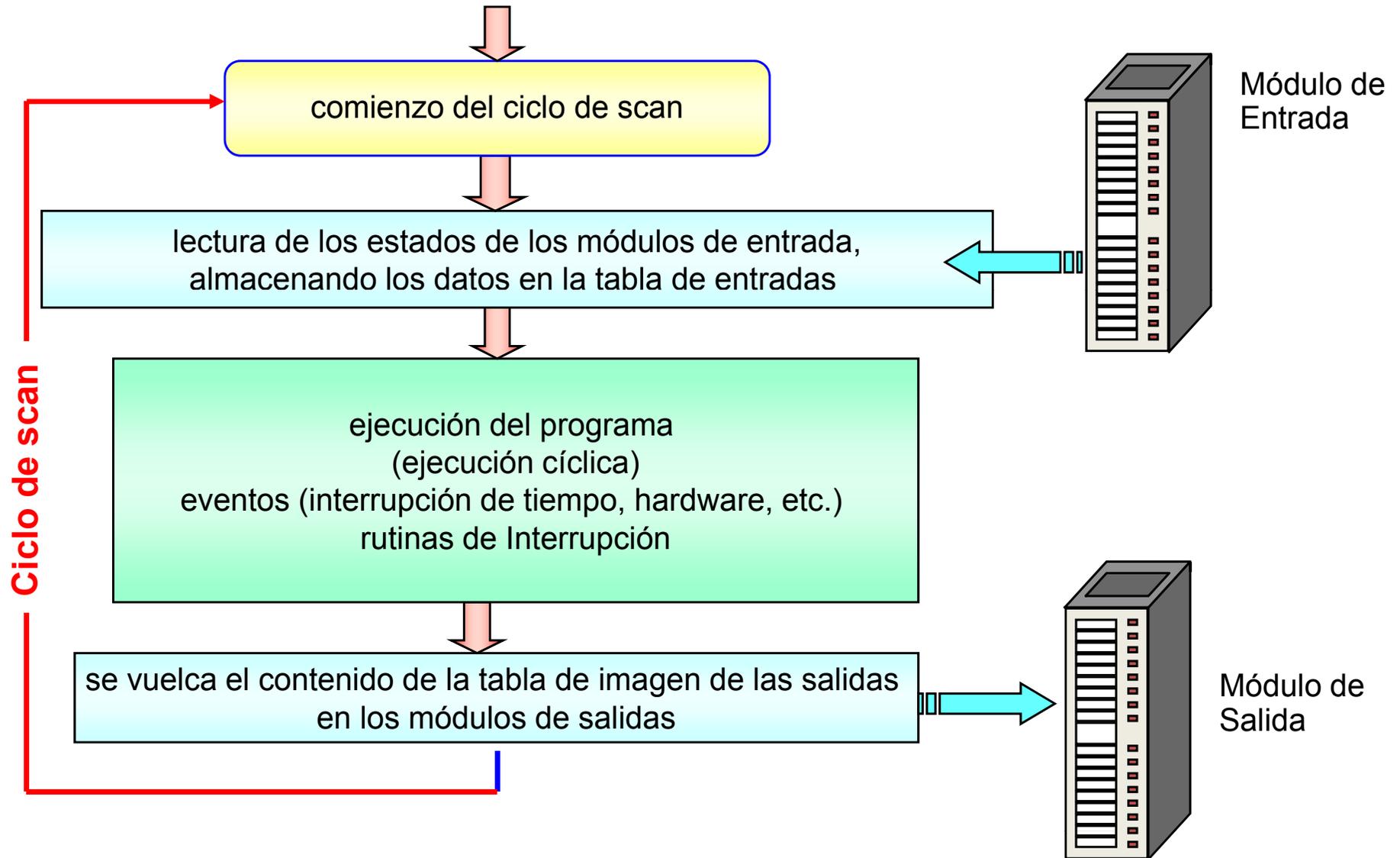
introducción a los autómatas

desarrollo de un proyecto con PLCs



introducción a los autómatas

concepto de ciclo de scan



estados tecnológicos

los componentes se fabrican en dos estados tecnológicos, así llamados, por su forma de trabajar

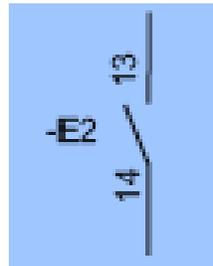
diremos que un componente está en estado de reposo, cuando coincide con el estado tecnológico de fabricación

diremos que un componente está en estado de activo, cuando no coincide con el estado tecnológico de fabricación

contacto NA

estado de reposo = "0"

estado activo = "1"



contacto NC

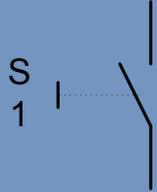
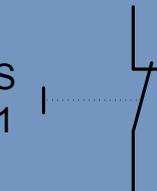
estado de reposo = "1"

estado activo = "0"



introducción a los autómatas

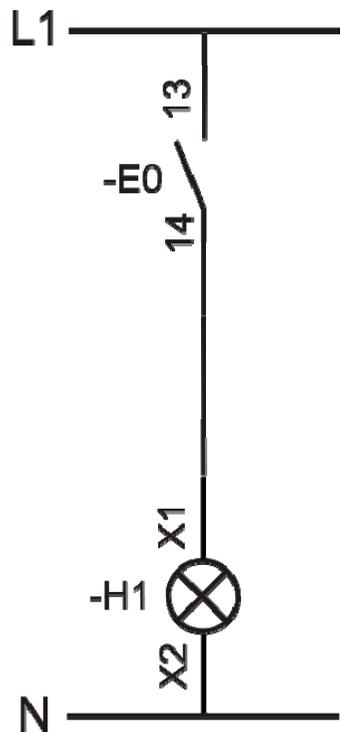
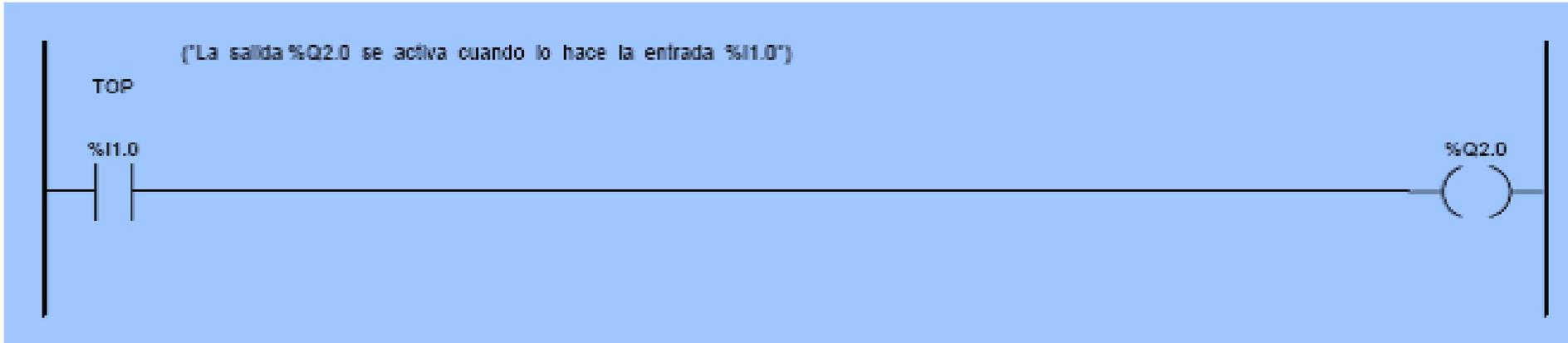
concepto de acción y estado de señal

El emisor de señal es un	El emisor está	Tensión en la entrada	Estado de la señal en la entrada
Contacto NA 	accionado 	existente	1
	no accionado 	no existente	0
Contacto NC 	accionado 	no existente	0
	no accionado 	existente	1

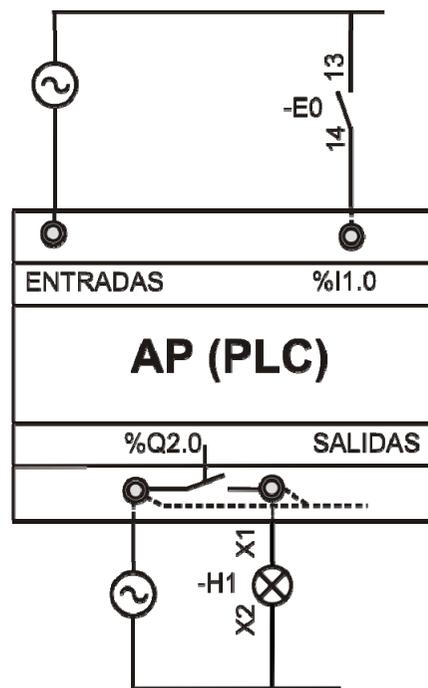
automatismos combinacionales

concepto de estado de señal (1) · se activa H1 cuando lo hace E0

lenguaje de contactos



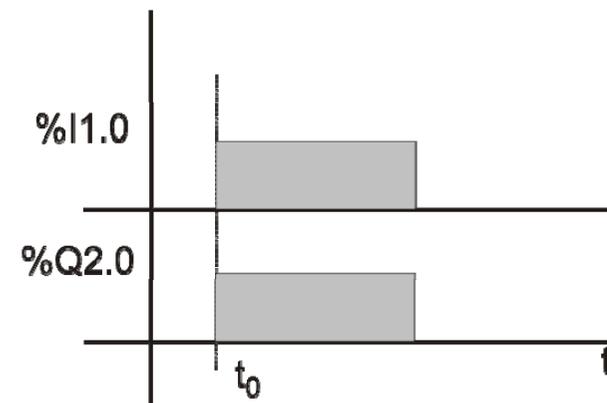
Lógica Programada



lenguaje estructurado

```
%Q2.0:= %I1.0;
```

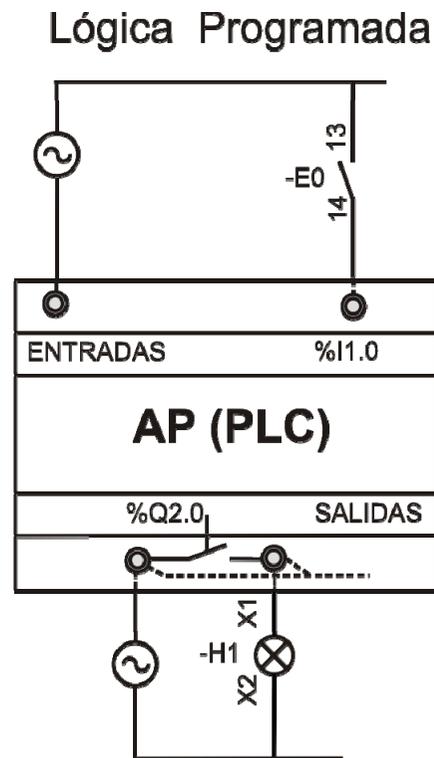
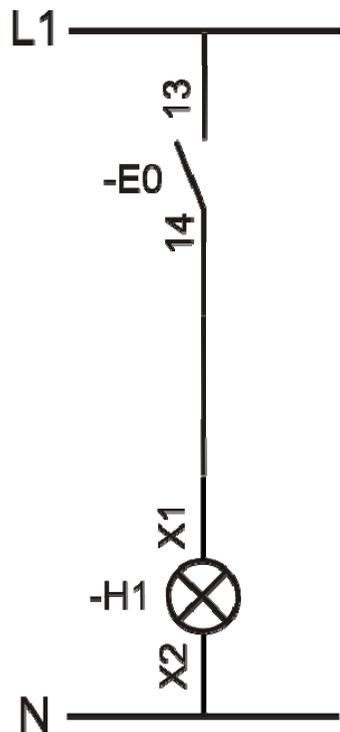
estado de señal



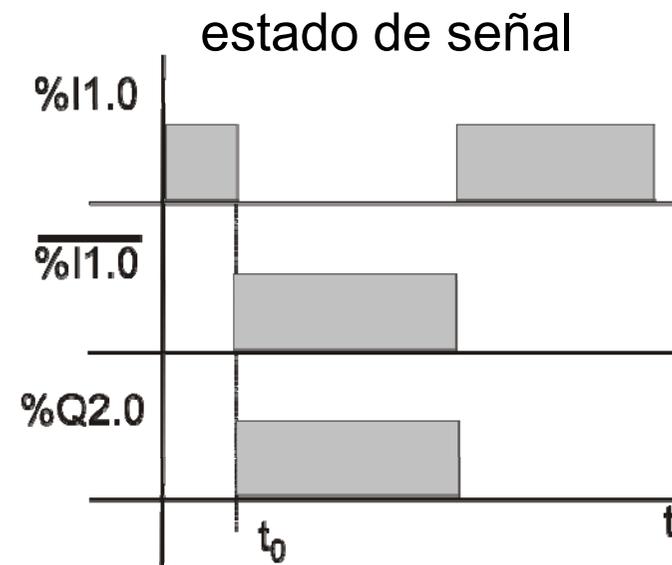
automatismos combinacionales

concepto de estado de señal (2) · se activa H1 cuando no lo hace E0

lenguaje de contactos



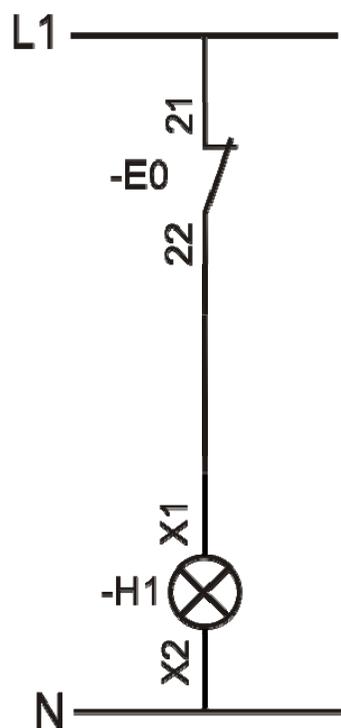
lenguaje estructurado
`%Q2.0:= NOT %I1.0;`



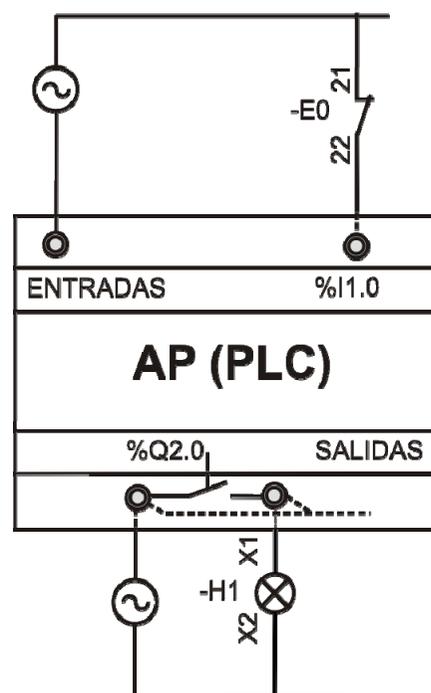
automatismos combinacionales

concepto de estado de señal (3) · se activa H1 cuando lo hace E0

lenguaje de contactos



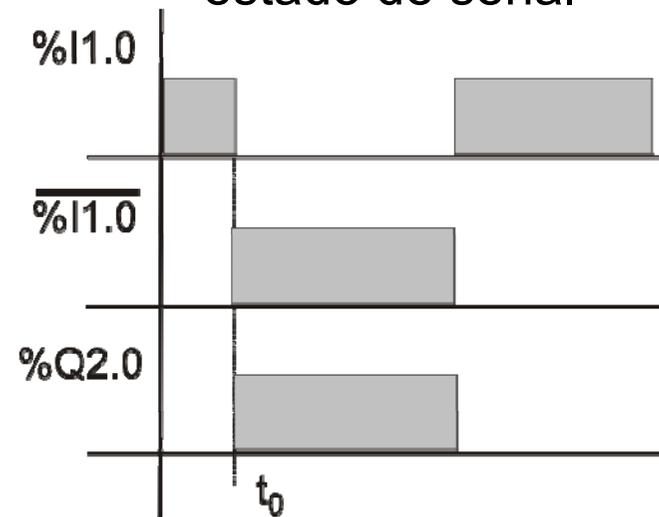
Lógica Programada



lenguaje estructurado

```
%Q2.0:= NOT %I1.0;
```

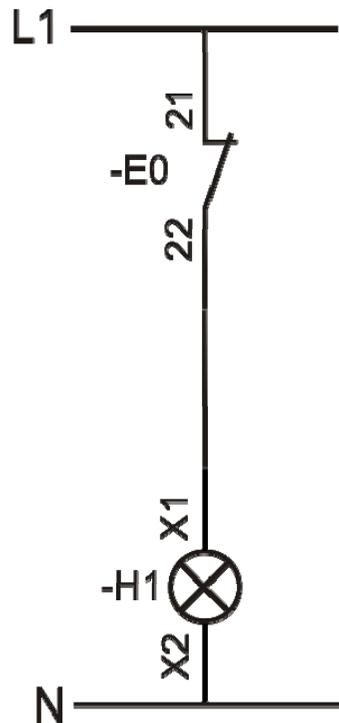
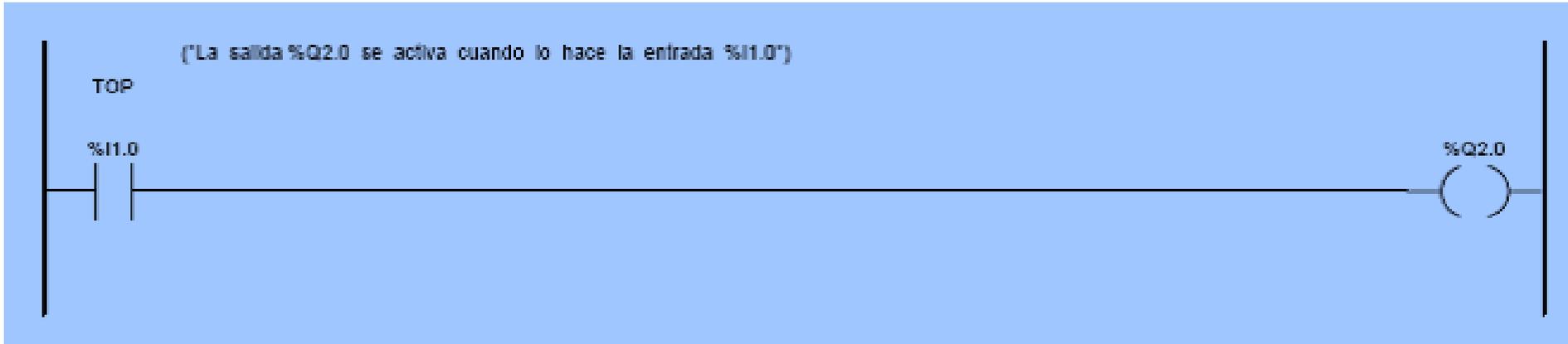
estado de señal



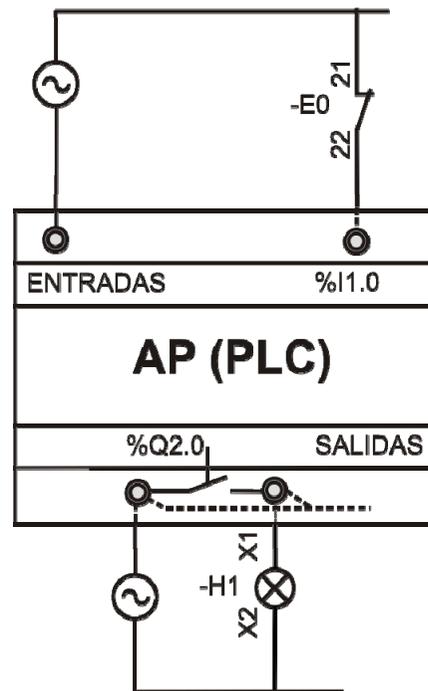
automatismos combinacionales

concepto de estado de señal (4) · se activa H1 cuando no lo hace E0

lenguaje de contactos



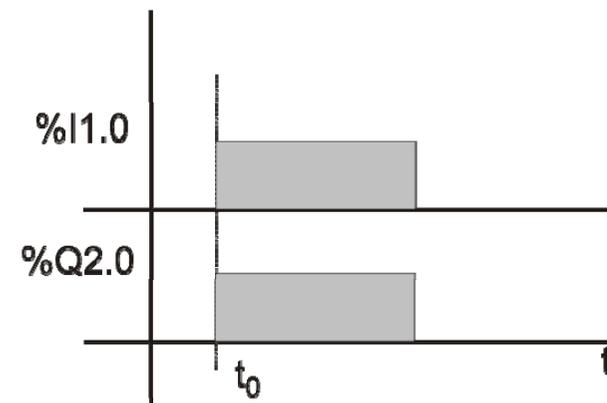
Lógica Programada



lenguaje estructurado

`%Q2.0:= %I1.0;`

estado de señal



introducción a los autómatas

ejercicio · las tres bombillas deben encenderse con S1 activo y S2 no activo

