



# GUÍA BÁSICA DE AUTÓMATAS

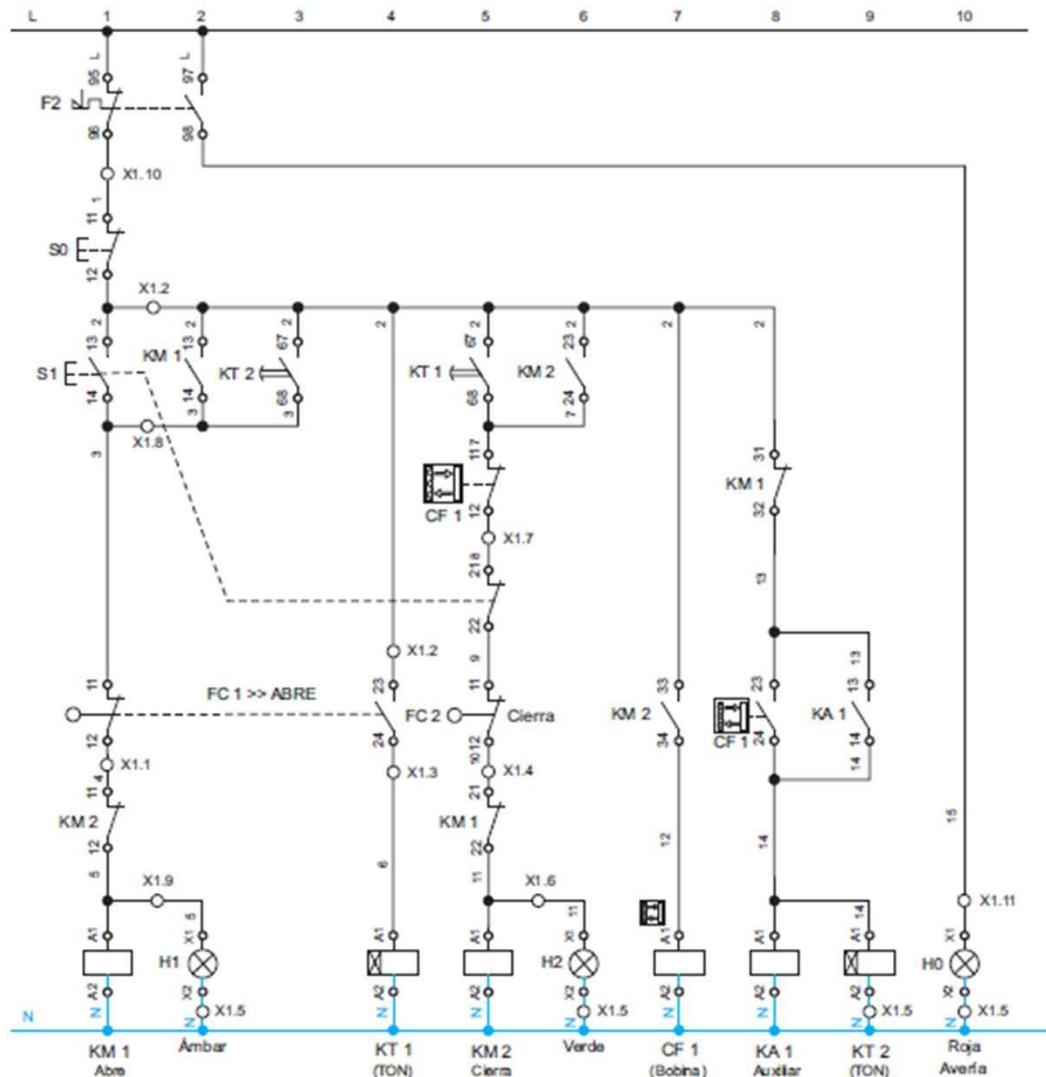
## 1 Lógica cableada

Cuando el funcionamiento de una operación automática se realiza con la alimentación principalmente de relés y contactores, producidas o provocadas por los accionamientos de elementos mecánicos tipo interruptor, pulsador, final de carrera, entre otros, se dice que la lógica del circuito es cableada; de hecho, una modificación en el funcionamiento supone la reestructuración de parte del cableado existente. Si la instalación es permanente con un proceso de funcionamiento sin proyectos de modificación, no es necesario que la gestione un PLC. Note el esquema de mando de la figura 10.1. referida al control motorizado de una puerta de garaje.

El pulsador S1, pone en marcha el sistema, que consiste en la apertura de la puerta (KM1), tiempo de reposo para entrada o salida de vehículos (KT1), y cierre de puerta (KM2). Como medida de seguridad, una célula fotoeléctrica que controla el ancho de la puerta mientras ésta se está cerrando, y en caso de actuación, provoca la parada de cierre de puerta, una pausa de dos segundos (KT2), comenzando el proceso de apertura.

Ahora proponemos algunas modificaciones:

- Si es de noche, además de abrir la puerta, se activarán las luminarias del garaje.
- Desde el interior del garaje existe un interruptor que anula el mando a distancia.
- Si se interrumpe tres veces seguidas el proceso de cierre de la puerta, el sistema se bloquea.



¿Qué problema plantean las modificaciones propuestas?

!! El completo recableado de la instalación !!

## 2 Nacimiento del autómata programable

Desde el comienzo de la industrialización, el hombre ha buscado y mejorado constantemente los procedimientos y medios para que las labores de producción se hicieran cada vez más rápidas, menos repetitivas para el operario, mejorando el puesto de trabajo y consiguiendo un rendimiento cada vez más eficaz, ayudado por la tecnología eléctrica basada en control y lógica cableada.

En 1968, una división de una fábrica de automóviles, propone a través de un concurso, la creación de un instrumento tipo controlador electrónico, que sustituyera a los sistemas de control cableados (interruptores, relés, contactores..), por un control programado, con el objetivo de ahorrar costes en los procesos de fabricación. Conocido como el padre del PLC, el desarrollador del proyecto fue Dick Morley.

El MODICON 084 (MODular DIGITAL CONtroller), fue el primer autómata programable comercializado. En 1972, los PLCs, ya comenzaron a programarse con un lenguaje parecido a lo que hoy conocemos como Lenguaje de Contactos Ladder Diagram, con símbolos que provenían del diseño de antiguos cuadros eléctricos.

### 3 Lógica programada

Si el número de captadores a utilizar en un proyecto es amplio, aunque se estime que la instalación no va a sufrir cambios, el coste del microcontrolador será insignificante si sólo una vez decidíramos realizar una modificación de control (recableado, pruebas, puesta en marcha, verificación, tiempo perdido, parada de producción, etc.).

Sirva el ejemplo; si proponemos gestionar las lámparas de los semáforos de un cruce de dos calles, la instalación la pueden realizar automatismos convencionales (relés y temporizadores), en una cantidad determinada; pero si el número de calles se amplía, ya no tiene sentido utilizar automatismos cableados, cuyo volumen sería exagerado; se haría con control programable. La siguiente figura, muestra el esquema de bloques de un autómata programable. Note, la cantidad de interacciones posibles.

#### PLC

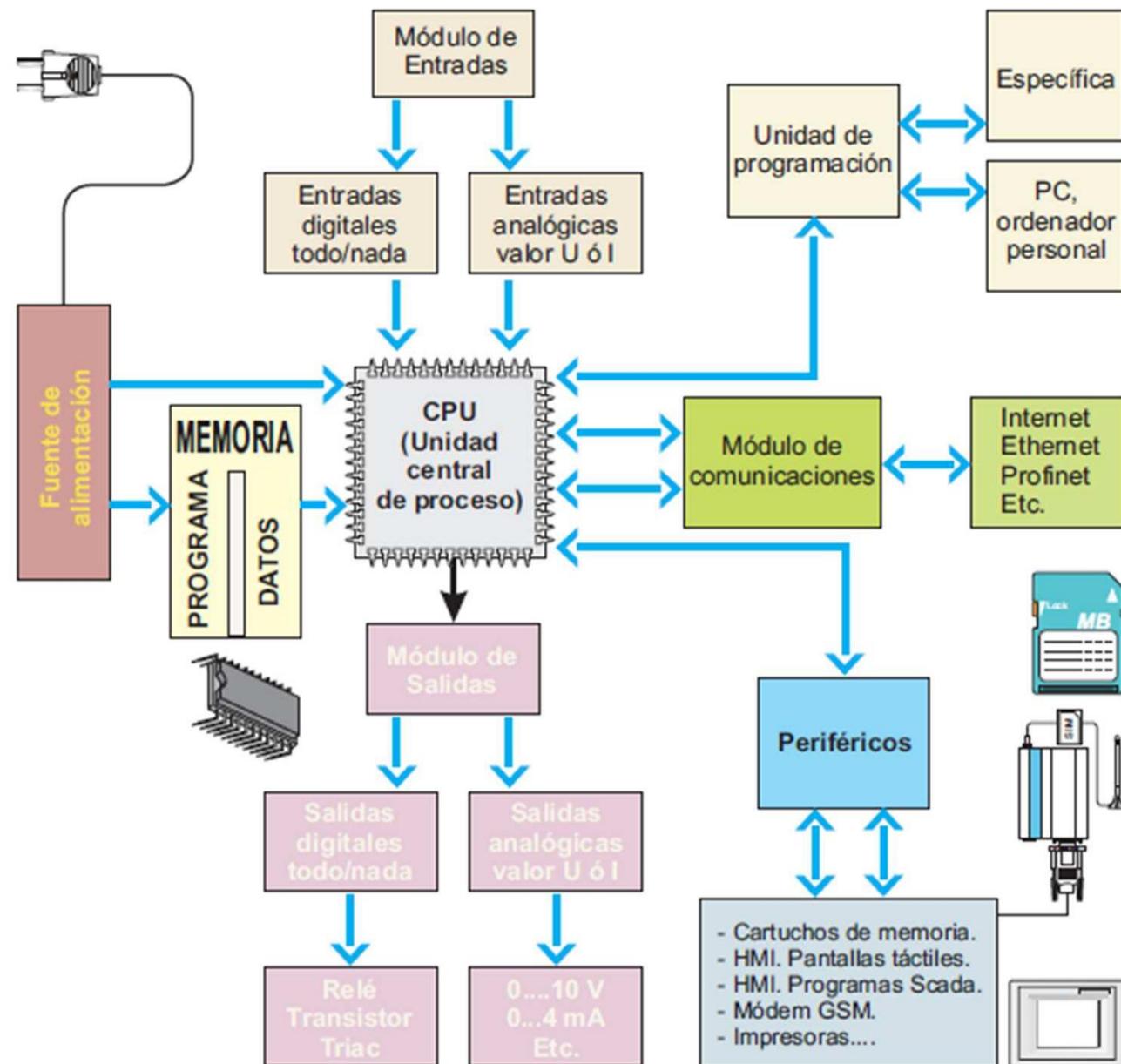
El término PLC proviene de "Programmable Logic Controller", que traducido viene a ser Controlador Lógico Programable, y se usará indistintamente con la designación de Autómata Programable.

#### Domótica

La automatización de viviendas y edificios, se denomina "Domótica".

Algunos de los sistemas domóticos están basados en autómata programable.

Esto supone la adaptación de PLCs industriales a entornos domésticos, donde cambia la tipología de sensores, actuadores y sobre todo, sistemas de comunicación.



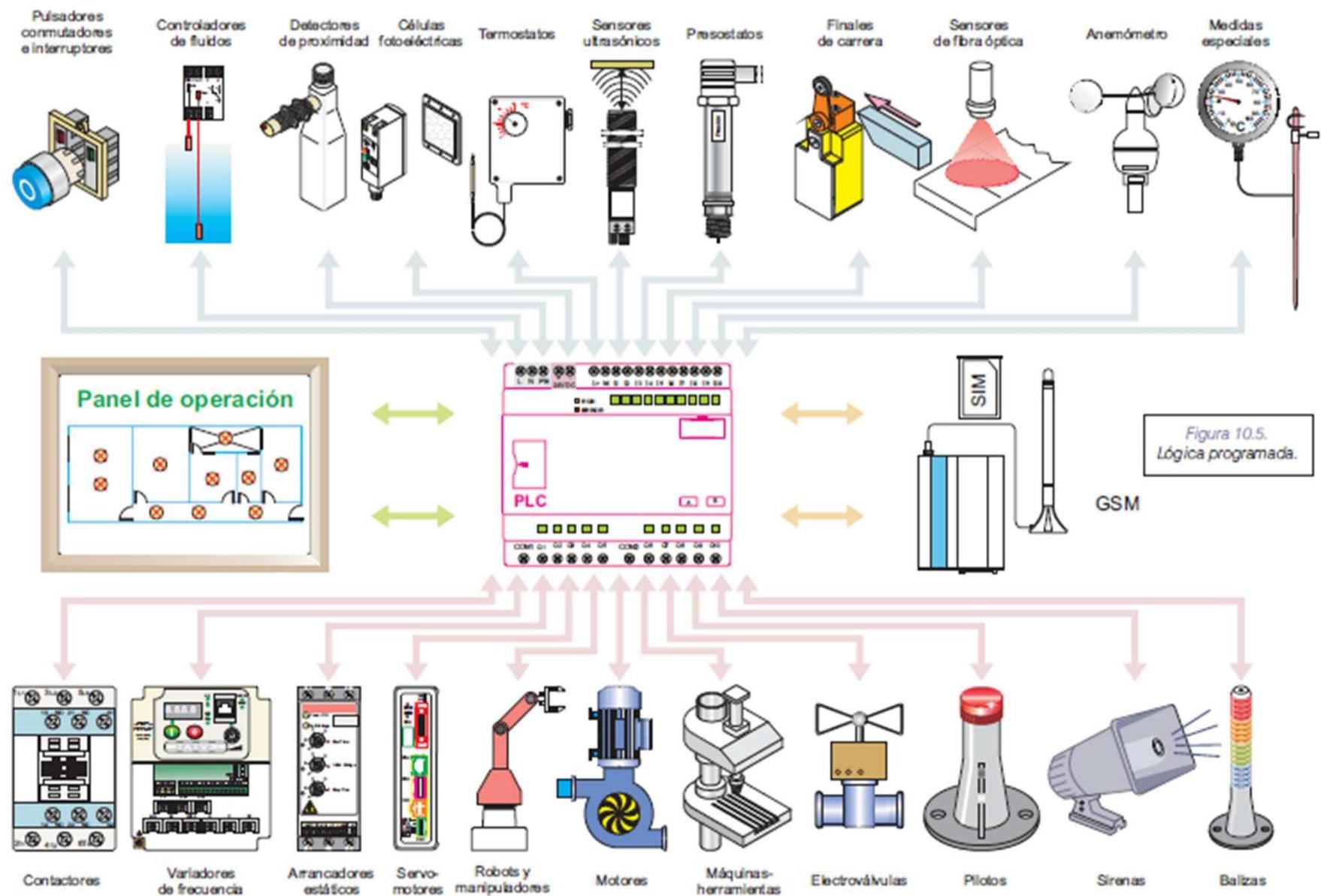


Figura 10.5.  
Lógica programada.

## Operaciones que debe realizar un autómata programable

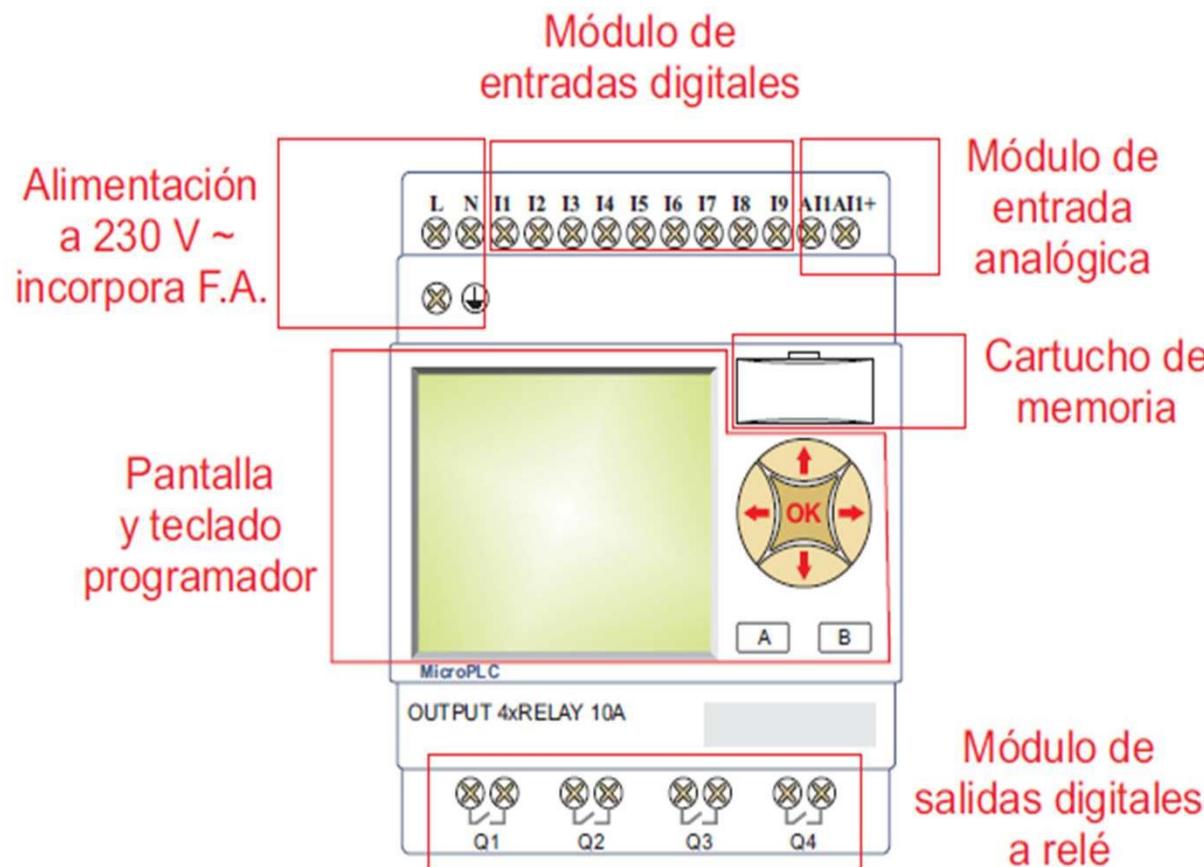
- Operaciones lógicas (AND, OR, NOT, NOR, NAND, XOR).
- Operaciones con entradas digitales.
- Operaciones con entradas analógicas.
- Operaciones con salidas digitales.
- Operaciones con salidas analógicas.
- Operaciones con marcas internas.
- Funciones aritméticas (sumas, restas, multiplicaciones, divisiones, raíces cuadradas, trigonometría, logaritmos...).
- Transferencia de datos de diferente tipología.
- Conversión de datos de unos sistemas de numeración a otros (entero, binario, BCD, hexadecimal, entre otros).
- Funciones de tiempo. Temporizadores y relojes.
- Funciones de cómputo. Contadores y contadores rápidos.
- Comparación de datos (mayor que, menor que e igual a.....).
- Creación de subrutinas, que es una forma de dividir el programa en fragmentos más pequeños, y todos juntos forman la programación final.

## Operaciones que debe realizar un autómata programable

- Funciones de interrupción del programa.
- Funciones de autochequeo.
- Funciones de red, para la interconexión con entes de origen superior e inferior, pudiendo establecerse comunicaciones tipo maestro-esclavo, esclavo-maestro, o simplemente compartir una red al mismo nivel de control con otros dispositivos.
- Control de visualización y control a través de pantallas y sistemas SCADA.
- Control remoto de los sensores y captadores.
- Control remoto de los sistemas de visualización y control incluso para diagnosis y reparación.
- Permite la grabación de los programas en varios formatos.
- Permite la redundancia con otros autómatas; esto supone que en caso de avería de uno de ellos, el otro -de idénticas características y misma programación- se hace cargo de la automatización.
- Se pueden programar en diferentes lenguajes.

## 5 MicroPLCs

### Micro autómatas programables



Prácticamente, todas las marcas de autómatas programables, disponen de un modelo de bajo nivel, limitado en entradas, salidas, y bloques operacionales, pero muy útiles en cuadros eléctricos para control discreto de operaciones.

Los nombres para su designación son diversos:

- Relé lógico.
- Caja de relés.
- NanoPLC.
- Mini autómata.
- Relé programable.
- Relé industrial.
- Etc.

## 6 Módulos de entrada

Los módulos de entrada podrán recibir principalmente señales digitales todo/nada, o analógicas en formato tensión (ejemplo 0...10 V DC) o intensidad (ejemplo 0...20 mA).

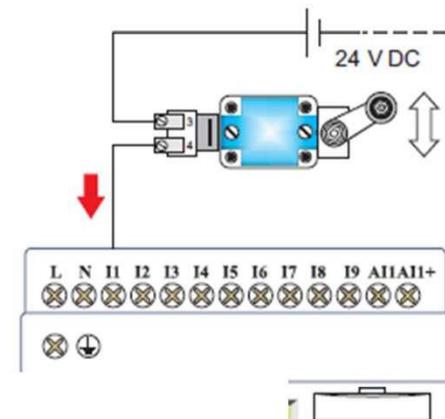
### 6.1. Señales digitales (todo-nada)

Los terminales de los módulos de entrada o simplemente los terminales de entrada digitales todo/nada, recibirán un valor de tensión de captadores tales como:

- Pulsadores.
- Interruptores.
- Finales de carrera.
- Termostatos.
- Presostatos y vacuostatos.
- Detectores capacitivos, inductivos o fotoeléctricos.

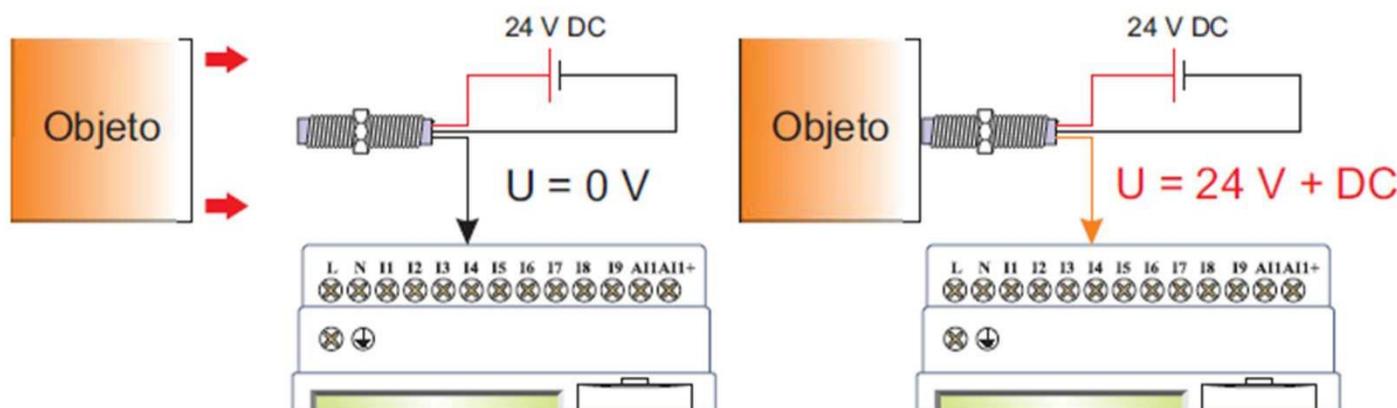
## Ejemplo

Suponemos que el módulo de entradas digitales de un microPLC admite una tensión de 24 V DC. Cada vez que el captador -final de carrera- permite el paso de dicha tensión al micro-autómata, estará enviando una señal que el programa de usuario tendrá que interpretar y actuar en consecuencia.



## Ejemplo

El siguiente detector capacitivo implementará 24 V (+), sólo cuando se acerque un objeto. Para poder funcionar, el detector tiene que estar alimentado por 24 V DC.



## 6.2. Señales analógicas

Los terminales de los módulos de entrada de señales analógicas, recibirán un valor de tensión o intensidad equivalente a la magnitud real medida. El técnico debe calibrar la señal procedente del sensor de forma correcta para evitar que la lectura sea errónea.

### Valores estándar de tensión

-10 V a + 10 V.  
0 a +10 V c.c.  
+2 a +10 V c.c.

### Valores estándar de intensidad

0 a 20 mA.  
4 a 20 mA.  
+1 a -5 mA  
0 a +5 mA.

### Transductor

Es un dispositivo, que transforma una magnitud física (presión, temperatura, velocidad, longitud, humedad...) en una señal -generalmente de carácter eléctrica-, que posteriormente será tratada.

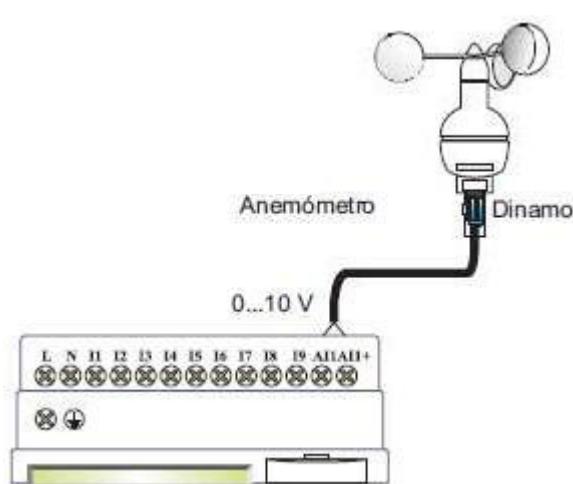
12,24 m



### Ejemplo

Un anemómetro mide la velocidad del viento, y en su composición, se encuentra una pequeña dinamo solidaria al eje principal del mismo. Según la velocidad de giro, la dinamo generará una determinada tensión; sirvan los valores:

- Anemómetro parado, genera 0V DC.
- Anemómetro girando a 50 km/h, genera 4 voltios DC.
- Anemómetro girando a 100 km/h, genera 8 voltios DC.

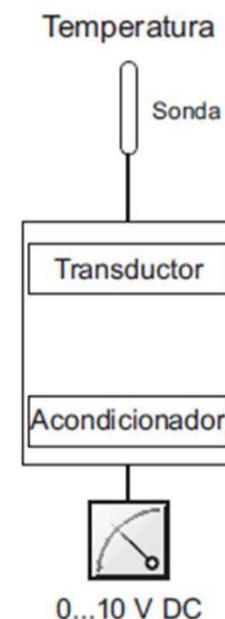


## 7 Módulos de salida

Los módulos de salida permiten alimentar los dispositivos que hacen el trabajo de las instalaciones. Como sucede con los módulos de entradas, las salidas aportarán señales todo/nada, o señales analógicas, como valores de tensión o intensidad variables. Las salidas digitales todo/nada alimentarán principalmente:

- Sistemas de alumbrado.
- Timbres o avisadores acústicos.
- Electroválvulas.
- Contactores.
- Relés.
- Aparatos de caldeo.
- Arrancadores.
- Variadores de frecuencia.

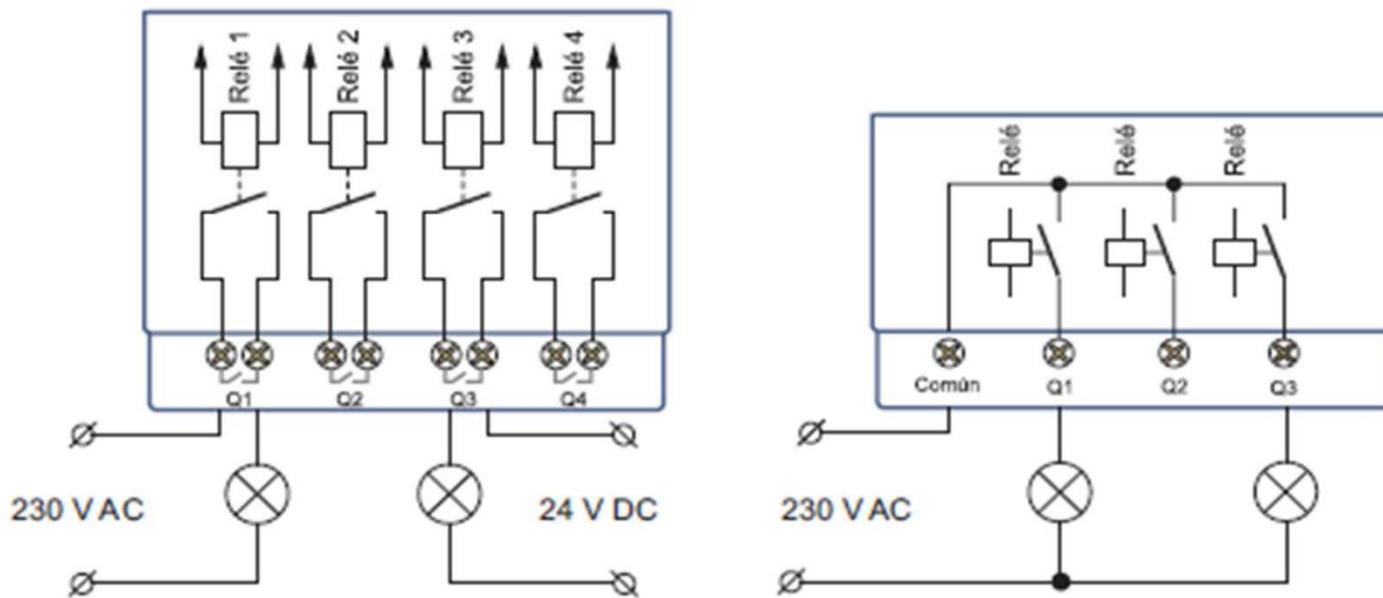
Un termómetro analógico será capaz de emitir a su salida un valor de 0 a 10 V corriente continua, equivalentes a la temperatura de 0 a 100 °C.



## 7.1. Salidas a relé

Probablemente, los autómatas con salidas a relé son los más empleados. Un relé es versátil; por sus contactos puede circular corriente continua, o alterna, y puede manejar valores superiores a 10 amperios. En su contra, la lentitud en las conmutaciones, y al emplear componentes mecánicos, éstos sufren desgaste.

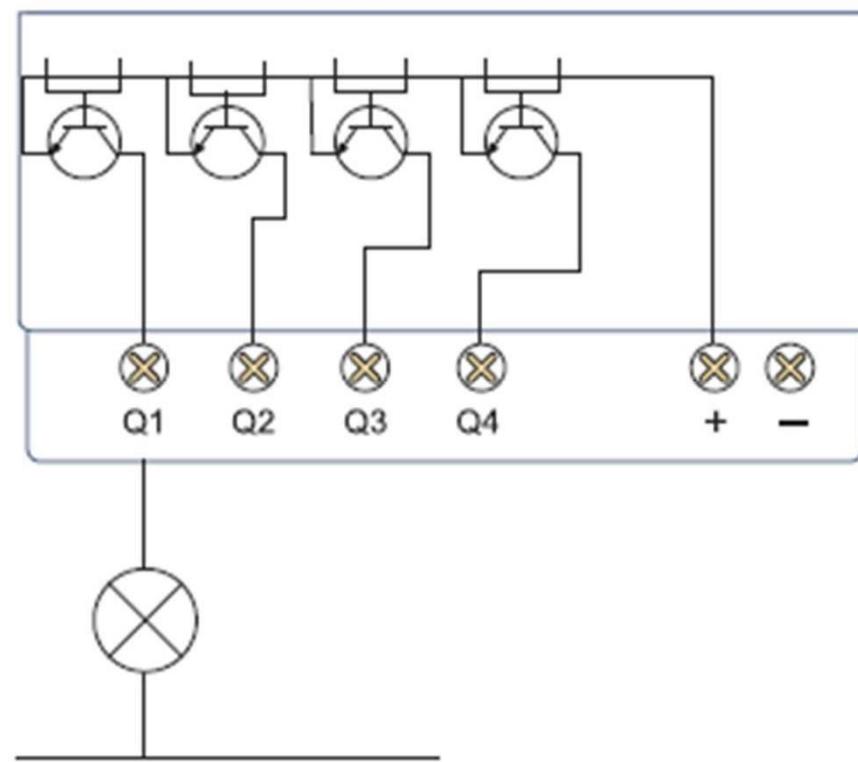
En el siguiente gráfico aparecen dos modelos de salidas a relé. En el primer caso, cada salida es operada por un relé diferente, lo que permite usar diferentes tensiones en los receptores. El segundo caso, muestra una misma línea de alimentación para tres relés, las cuales tendrán obligatoriamente el mismo potencial.



## 7.2. Salidas a transistores

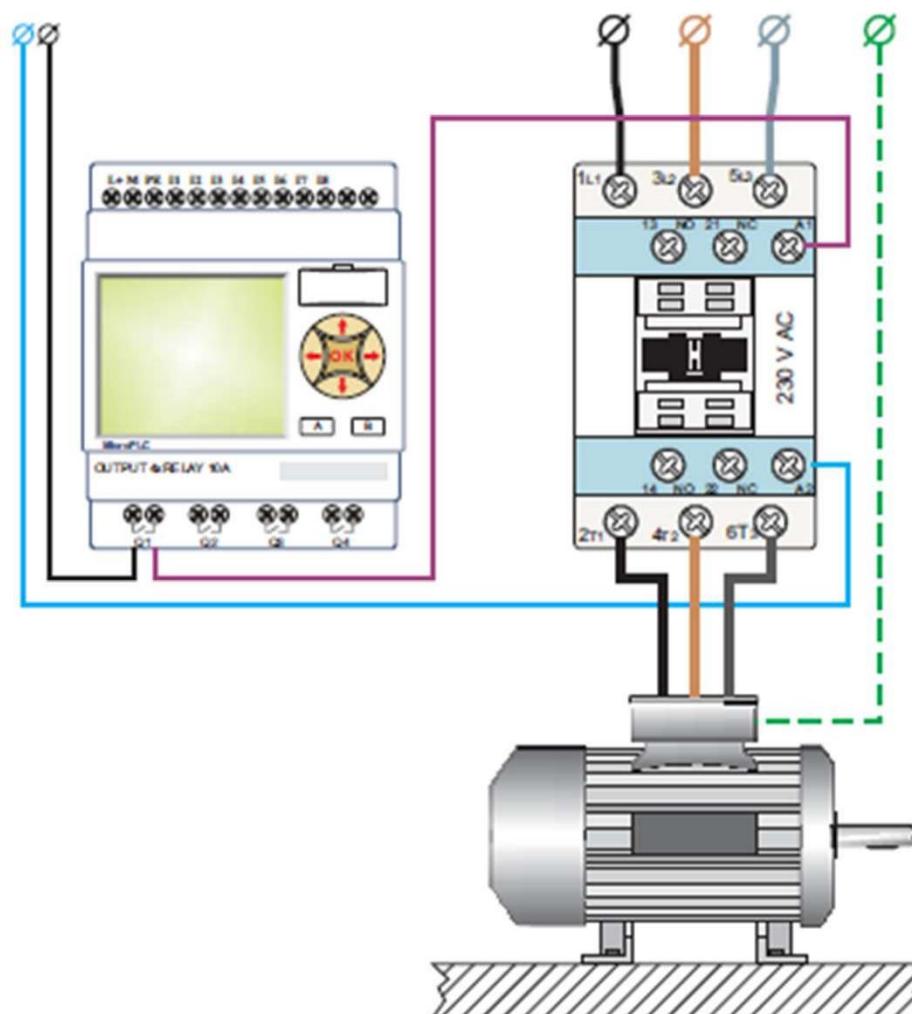
Los transistores son dispositivos de estado sólido. Son elementos electrónicos que no tienen partes móviles, por tanto, no tienen desgaste.

Los transistores conmutan corriente continua y son muy rápidos (incluso miles de veces por segundo). En su contra, la corriente de paso.



El contactor como recurso en las salidas del PLC

Si la carga a gobernar por el autómata programable es elevada, tanto para salida a relé, transistor o triac, se puede recurrir al empleo de un contactor y de esta forma, el PLC sólo tendrá que alimentar la bobina del contactor. La operación no debe presentar problemas técnicos ni eléctricos. Figura 10.19.



## 8 Unidad central de proceso CPU

Es el cerebro en el más amplio sentido de la palabra del autómata programable. La CPU procesa todas las señales que provienen de los captadores de entrada, los evalúa consultando al programa de aplicación y ordena a las salidas que efectúen sus oportunas aplicaciones.

Consideremos que un autómata tiene un sistema operativo, que es un programa interno que permite que el PLC pueda realizar las operaciones. Este sistema operativo no es modificable por el usuario y viene instalado de fábrica. Asimismo, el programa de usuario que es el programa en el cual el operador diseña y realiza la programación, es modificable y borrable. El programa de usuario se realiza mediante una unidad de programación -por ejemplo un PC- y se transfiere al autómata mediante un cable de conexión; esta acción no es definitiva, es decir, se puede modificar el programa, hasta que el autómata realice las especificaciones deseadas. El encargado de coordinar el programa de usuario con los datos de los sensores de entrada y ordenar que las salidas se activen, es el sistema operativo.

## 8 Unidad central de proceso CPU

El programa de usuario se realiza con un lenguaje específico más o menos fácil (dependiendo de la profundidad de la programación) que puede ser textual o gráfico. Sin duda, los usuarios acostumbrados a utilizar esquemas eléctricos, prefieren los lenguajes gráficos. Esto no quiere decir que el PLC reconozca esos gráficos que suponen la programación que el autómata tiene que cumplir. Cuando la unidad de programación transfiere el programa -realizado con cualquier lenguaje- los datos que viajan al PLC y por tanto, los que reconoce, están en un lenguaje denominado lenguaje máquina .

Dentro de la CPU encontramos principalmente el microprocesador ( $\mu$ P) y las memorias. El microprocesador que habitualmente no es una unidad física sino varios circuitos integrados, realiza funciones de comparación, cálculo aritmético, entre otros, y en definitiva, la transferencia de información dentro del autómata.

Dentro de éste, aparecen circuitos que desarrollan diferentes funciones: la unidad de control, la unidad aritmética lógica y un circuito de registros.

## 8.1. Memorias

Las memorias son unos elementos capaces de almacenar información en forma de ceros y unos (bits). Existen varios tipos de memoria:

Memoria RAM; (random access memory) memoria de lectura y escritura. Es el lugar donde reside el programa de usuario. Es requerida por el microprocesador para consultar el programa. Por tanto, es modificable en todos sus aspectos: programación, desprogramación, borrado.

Memoria EPROM; memoria de sólo lectura, es borrable y programable. Esta memoria es parecida a la ROM (Read Only Memory) pero modificable, esto es para poder cambiar el sistema operativo del autómata por otra versión. En caso de falta de alimentación, no se borra su contenido.

## 8.2. Datos del PLC

Si un pulsador está activado o desactivado, esa información la procesa el autómata como un dato, y ese dato será usado para realizar unas operaciones u otras, según la programación dispuesta para esa entrada en la que está conectado el pulsador.

Un dato ocupa una posición de memoria; ésta puede usar 8 posiciones que son independientes, pero que comparten la misma dirección. Hablamos sólo de ceros y unos (0y 1).



## 8.2. Datos del PLC

Cada cero o uno (0 ó 1) se llama bit. Ocho bits (b) forman un byte (B). El uso del byte es lo que habitualmente determina la capacidad de memoria de un dispositivo, así si decimos que un aparato tiene una memoria de 4 KB, estaremos diciendo que puede almacenar  $4 \times 1024 = 4096$  Bytes; ya que 1 KB equivale a 1024 Bytes.

En autómatas programables se habla de palabras (Word) cuando la información o datos utiliza dos bytes de memoria y de palabras dobles (Double Word) cuando la información utiliza cuatro bytes.

Además de lo anterior, existen otros formatos de mayor calibre, como el REAL, de 32 bits, que permite operaciones más precisas.

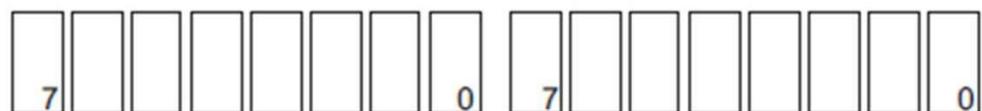
Un bit es la unidad más pequeña (0 ó 1)



Un conjunto de 8 bits es un byte (B)



Un conjunto de 2 bytes es una palabra (Word)



Un conjunto de 4 bytes es una doble palabra (DW)

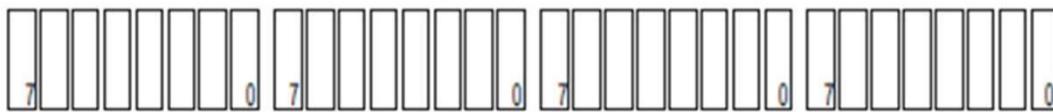
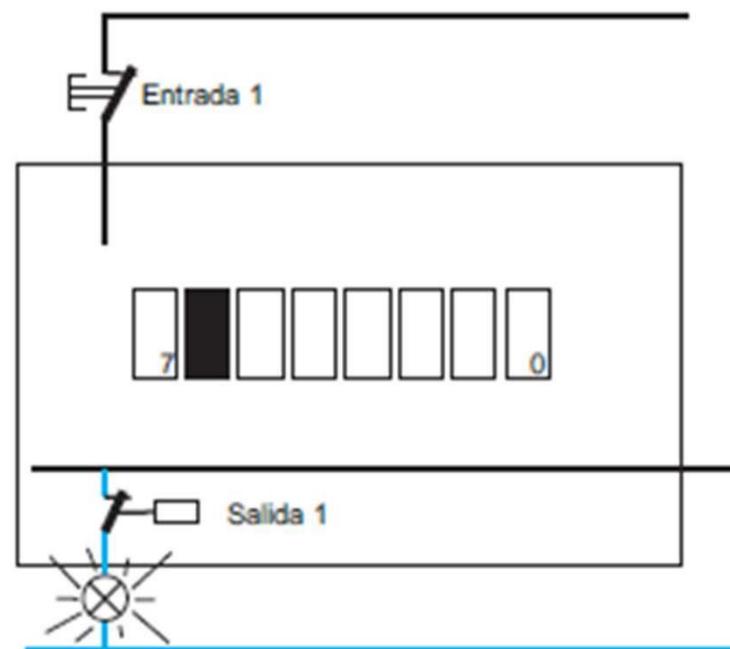
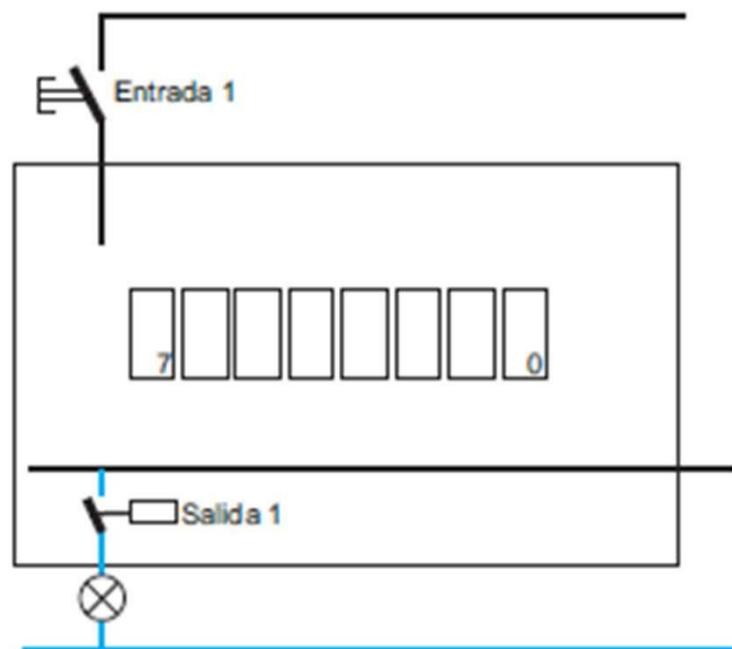


Figura 10.24. Representación de bit, byte, palabra y doble palabra.

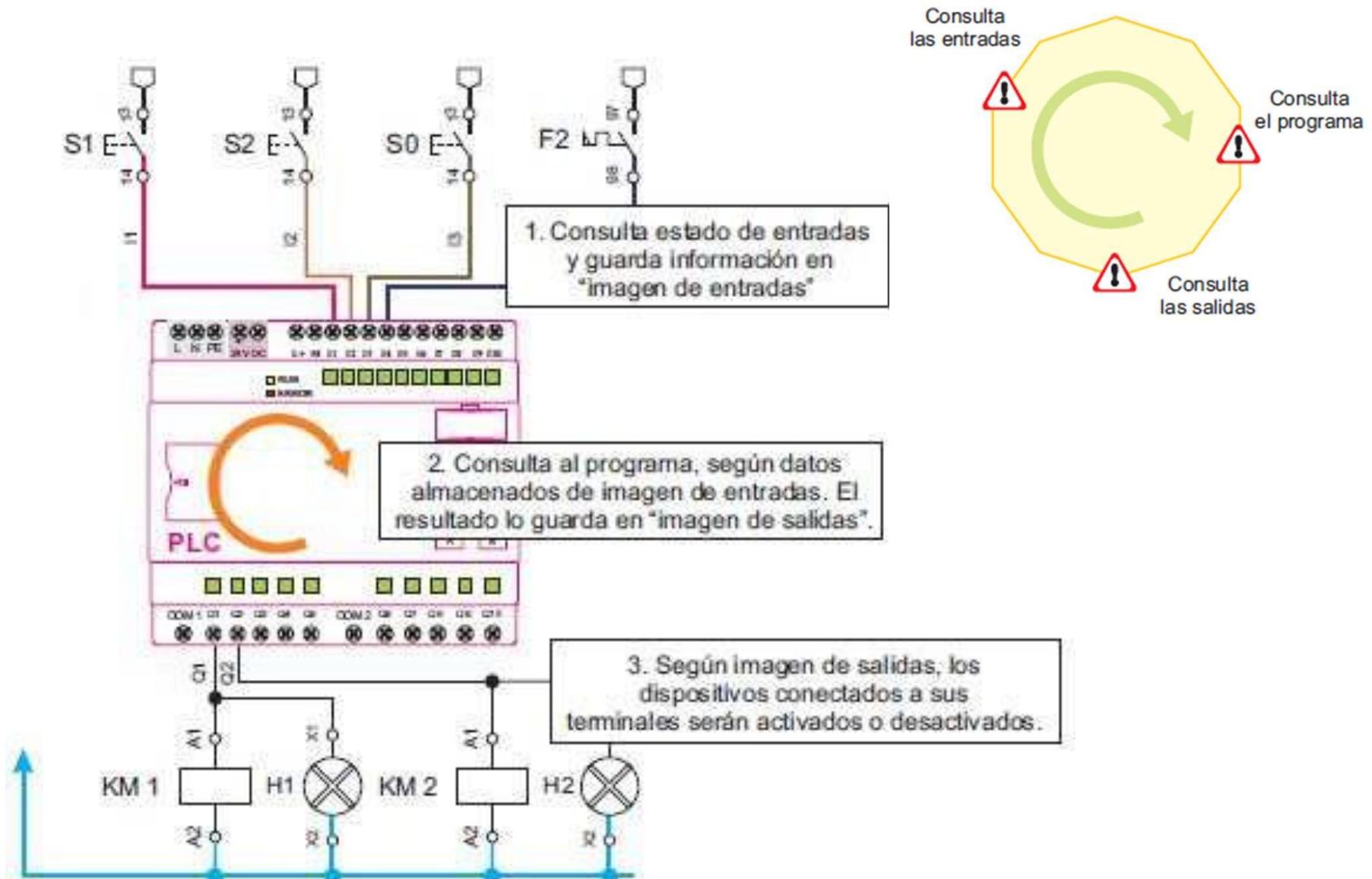
Aun así, el usuario en las aplicaciones puede operar con la unidad más pequeña, bit. Pongamos un ejemplo de ocupación de memoria; si un sensor un pulsador es presionado, y da señal eléctrica a una entrada del autómata, esta orden estará ocupando un bit, que indicará ocupado o no ocupado , si le aplicamos corriente o no. Con lo cual, la programación podría ser la siguiente: si el bit 6 tiene valor de 1, entonces que se active la salida 1 .



### 8.3. Ciclo de programa

Un ciclo de programa es un proceso en el cual el autómata realiza un análisis del estado de las entradas, consulta al programa, y actualiza el estado de las salidas en función de la información que ha obtenido del análisis de entradas y consulta al programa. Esta operación se realiza periódicamente, y se le llama tiempo de ciclo al tiempo que transcurre durante un ciclo de programa, que ronda los milisegundos, aunque dependerá de la máquina.

- 1- Se analiza el estado de las entradas; las que están activadas (On) y las que están desactivadas (Off), a continuación esa información se guarda de forma temporal en un archivo de memoria denominado **imagen de entradas** .
- 2- Se analiza y ejecuta el programa de usuario; esto es que el autómata consulta las instrucciones en el programa lógico, y usando el estado de las entradas (el estado que se almacena temporalmente en el archivo **imagen de entradas** ) decide qué salida debe ser activada o no. El resultado que tendrán las salidas ahora, se guarda en otro archivo de memoria denominado **imagen de salidas** .
- 3- Análisis de salidas. Según los datos que se encuentren en la **imagen de salidas**, el autómata conectará o desconectará los circuitos de salida, realizando de esta forma la gestión de los dispositivos de conexión a los terminales de salida.



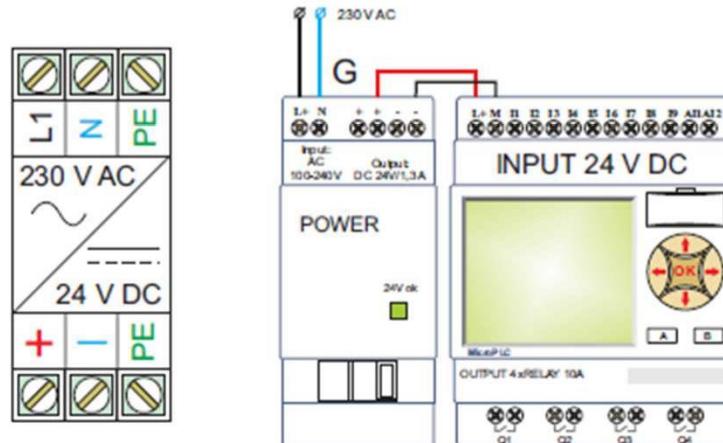
*Ciclo operativo de programa.*

## 9 Fuente de alimentación

Es el mecanismo que proporciona la alimentación eléctrica a los componentes internos del autómata programable. La tensión eléctrica que le llega a éstos, será menor que la de red. También, la fuente de alimentación protegerá al autómata contra alteraciones eléctricas en la red.

El dispositivo puede ser interno o externo, y además de alimentar al autómata programable, puede hacerlo hacia dispositivos periféricos como pantallas HMI, o incluso sensores; por este motivo es importante calibrar bien sus dimensiones.

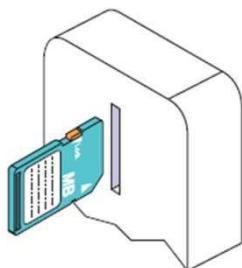
De la fuente de alimentación (si es externa) debemos conocer principalmente la tensión de alimentación, la intensidad de trabajo, y en su caso, la temperatura ambiente que soportará el dispositivo.



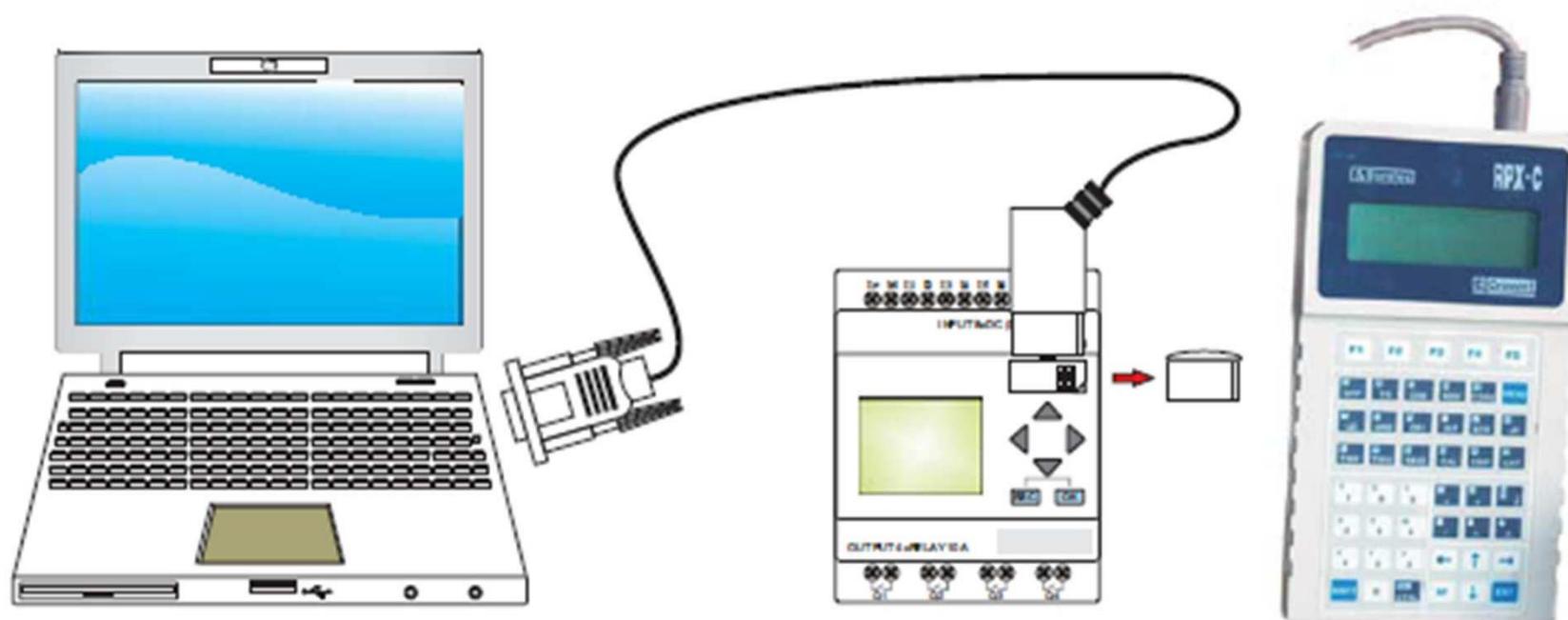
## 10 Periféricos

Los periféricos son los dispositivos que se conectan a un autómata programable, y que completan, facilitan y amplían, las labores de automatización. Destacamos entre otros:

- Impresoras. Además de imprimir los programas de usuario, permitirán crear entre otros, históricos de avisos, averías, paradas no programadas, etc.
- Cartuchos de memoria EEPROM. Para guardar el programa de usuario en el propio PLC.
- Visualizadores y pantallas táctiles. Son dispositivos hombre-máquina que sustituyen pulsadores y avisadores convencionales. Con estos dispositivos se podrá controlar uno o varios procesos en tiempo real, incluso se podrá simular la operación en la pantalla, por ejemplo, llenado de botellas.
- Pasarelas de comunicación. Módem GSM, conexión a ethernet, internet, etc. Permiten exportar, importar y compartir los datos que procesa el PLC.

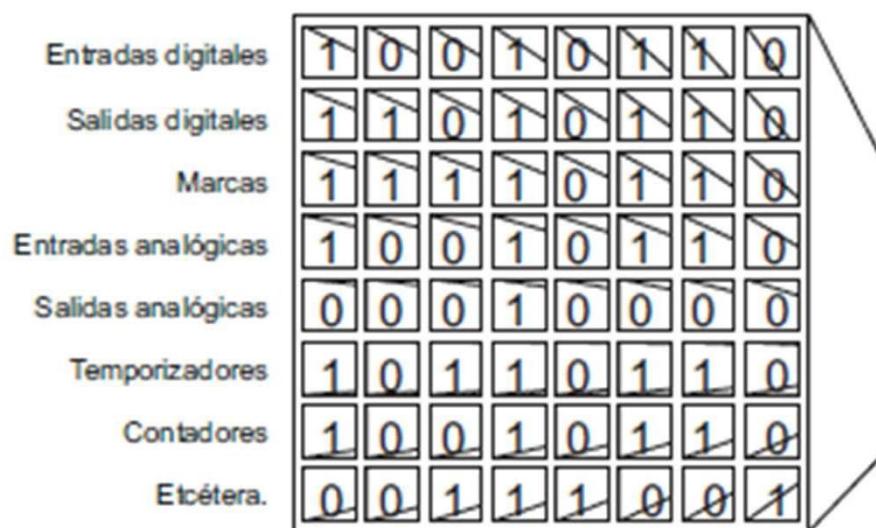


## 11 Dispositivos de programación de autómatas programables



## 12 Variables

En un autómata programable, y en programación en general, una variable es un lugar donde se guardan cierto tipo de datos. Estos datos podrán ser diversos; textuales, imágenes, sonido, etc. En autómatas programables, los datos se guardan en formato tipo: bit, byte, palabra y doble palabra preferentemente, aunque este concepto depende de la tecnología de la firma comercial. Es fundamental comprender que una variable de un tipo, se debe guardar sólo en el lugar destinado a ello, por ejemplo, un dato de Palabra (16 bits), no se puede ubicar en una variable de bit. La llamada a una variable es inequívoca, de tal forma que no existirán dos variables con el mismo nombre.



## 12.1. Variables de entrada digitales

Las variables digitales que relaciona el autómata con los dispositivos de entrada se identifican como I de input, por ejemplo: I1, entrada 1; I4 entrada 4; I0.0.1 entrada del módulo 0, del bastidor 0, entrada 1, etc. Estas variables operan con datos tipo bit (0 ó 1), todo o nada, activado o no activado. También se llaman datos Booleanos.

## 12.2. Variables de entrada analógicas

Las variables analógicas necesitan más capacidad de almacenaje, ya que los valores equivalentes a la magnitud medida pueden ser infinitos. Por ejemplo, la variable AI1 (Analogic Input 1), utilizará almacén de datos de 10 bits. Otro ejemplo, un anemómetro medirá la velocidad del viento a través de la variable AI2 de 16 bits.

El autómata programable dispondrá de las herramientas necesarias para poder convertir, transferir, y en definitiva operar con los diferentes tipos de datos.

## 12.3. Variables de salida digitales

Las variables de salida son de estado 1 / 0, variables booleanas. Las variables Q (Output), se identifican con los dispositivos a los que el autómata alimentará, por tanto, si leemos Q2, entendemos que es la segunda variable de salida y lo que hubiera conectado a ese conector, será excitado.

## 12.4. Variables de salida analógicas

Como ya ocurre con las entradas analógicas, el valor de salida se mueve en un margen muy amplio de datos, desde 10 a 32 bits, preferentemente. Por ejemplo, una salida analógica regulará la intensidad del alumbrado, a través de la salida AQ4 de 12 bits.

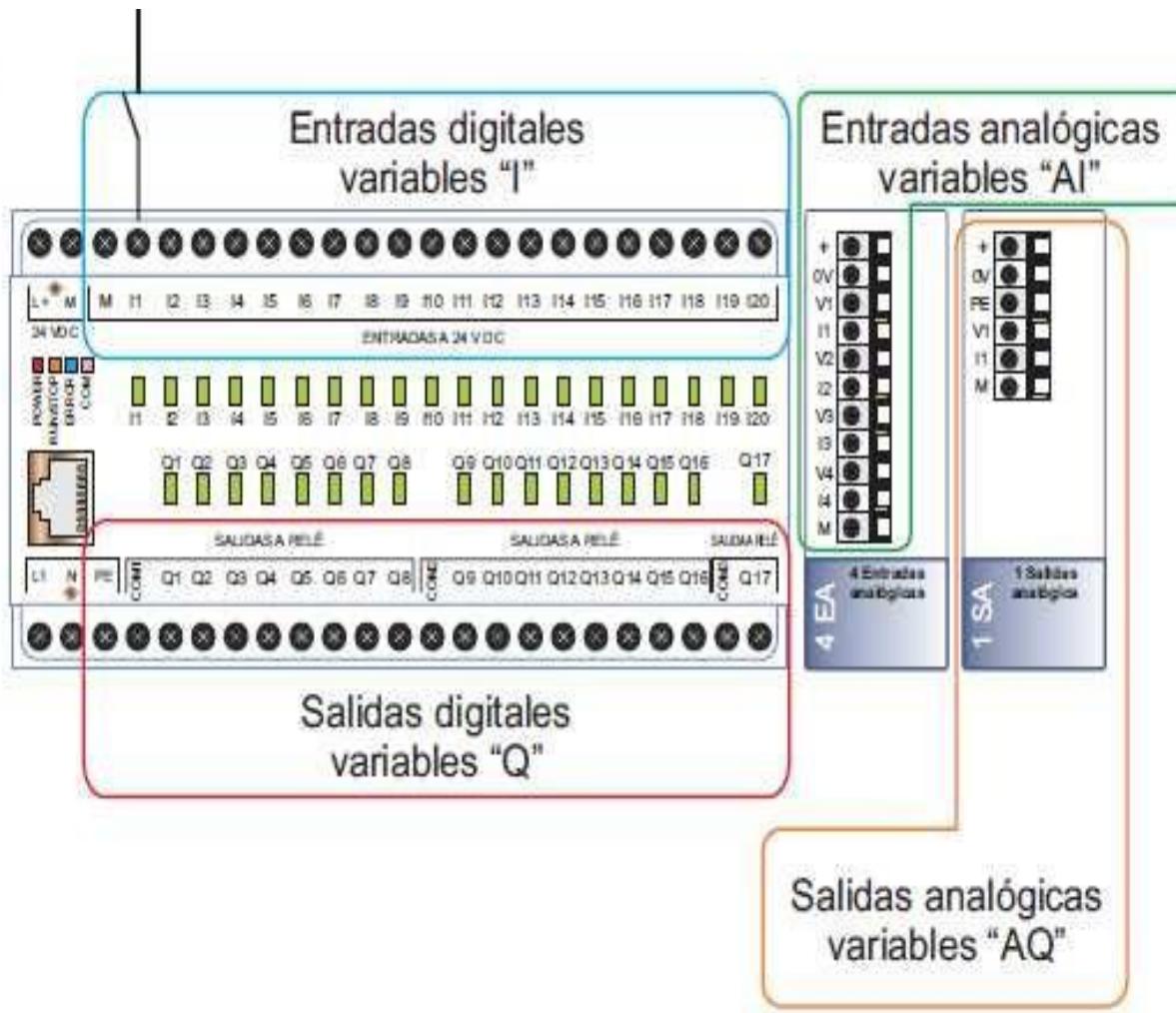
### Áreas de datos

La tabla que se muestra a continuación es meramente ilustrativa, ya que no existe un modelo común de área de datos para los diferentes autómatas, inclusive dentro de una misma firma comercial.

Área de datos	Identificador
Entradas Digitales	I
Entradas analógicas	AI
Salidas digitales	Q
Salidas analógicas	AQ
Marcas	M
Contadores	C
Contadores rápidos	HC
Temporizadores	T
Relojes	T

## 12.5. Variables de marcas (memorias internas)

Prácticamente todos los autómatas programables, disponen de unas variables internas, que sirven de apoyo en los programas. Las variables M de memoria (o marca), hacen referencia a salidas internas dentro del PLC; son salidas que no tienen una actuación visible directamente, y actúan como relés dentro del propio dispositivo. Además, las variables M pueden ocupar direcciones de memoria mayores, como MB -marca de byte de 8 bits-, MW -marca de palabra de 16bits-, entre otros.



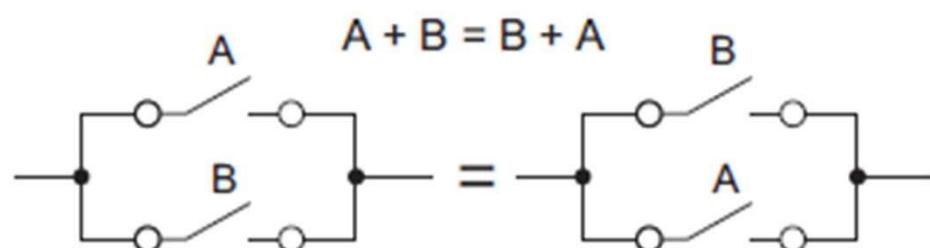
## 12.6. Variables específicas

Las que operan con datos internos del PLC, como temporizadores, contadores, contadores rápidos, etc.

## 13 Álgebra de Boole

Álgebra, en el que sus componentes sólo pueden tomar dos estados. Se considera un álgebra especial para el sistema binario. Proporciona el modo de expresar el funcionamiento de un circuito lógico compuesto por la combinación de puertas lógicas, en la que se puede saber el resultado de la salida, según la combinación de los valores de entrada.

Propiedad CONMUTATIVA:

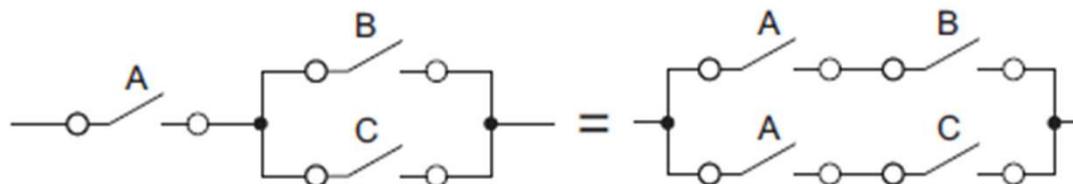


$$A \cdot B = B \cdot A$$

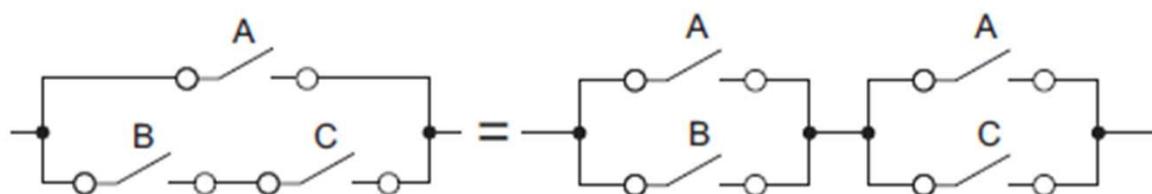


Propiedad DISTRIBUTIVA:

$$A \cdot (B+C) = A \cdot B + A \cdot C$$



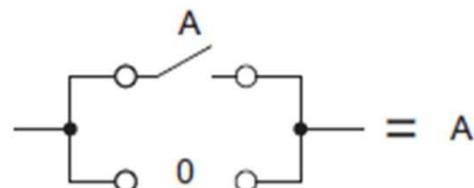
$$A + B \cdot C = (A+B) \cdot (A+C)$$



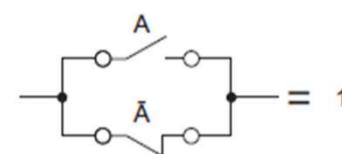
Existencia de elementos neutros (0 y 1), para las operaciones (+ y ·), respectivamente.

$$A + 0 = A$$

Para cada elemento A, existe su complemento  $\bar{A}$



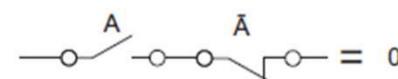
$$A + \bar{A} = 1$$



$$A \cdot 1 = A$$



$$A \cdot \bar{A} = 0$$



### 13.1. Tabla de la verdad

Es una tabla que muestra una serie valores claramente ordenados y ejecuta uno o varios resultados de la relación entre ellos.

Ejemplo 1. Contacto abierto.

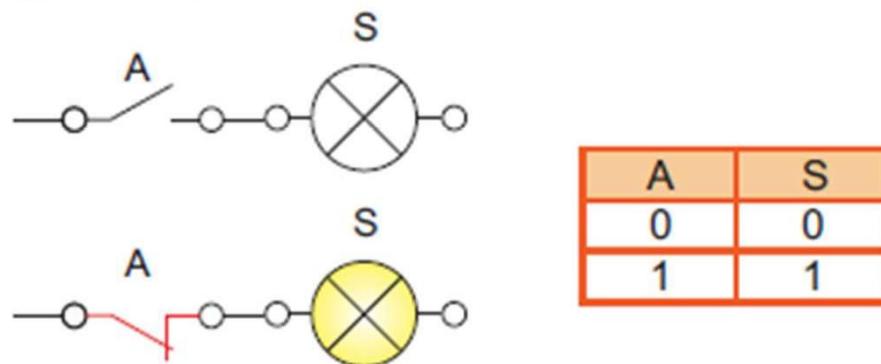


Figura 10.39. Tabla de la verdad de un solo contacto NA.

Ejemplo 2. Dos contactos en paralelo. Operación A + B.

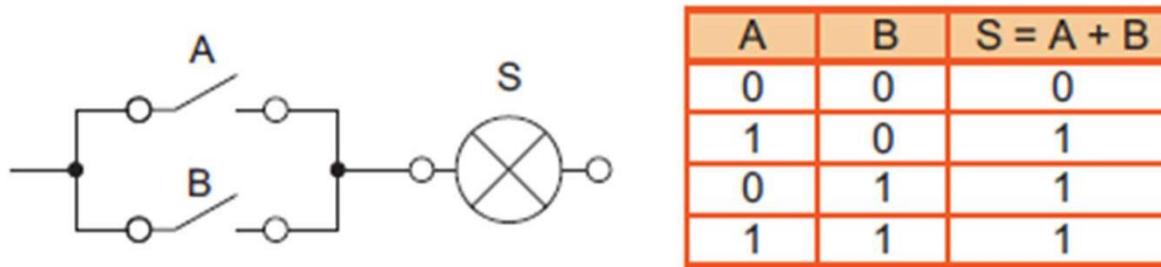
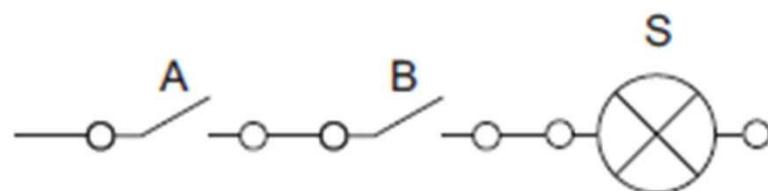


Figura 10.40. Tabla de la verdad de dos contactos NA en paralelo.

Ejemplo 3. Dos contactos en serie. Operación A · B.



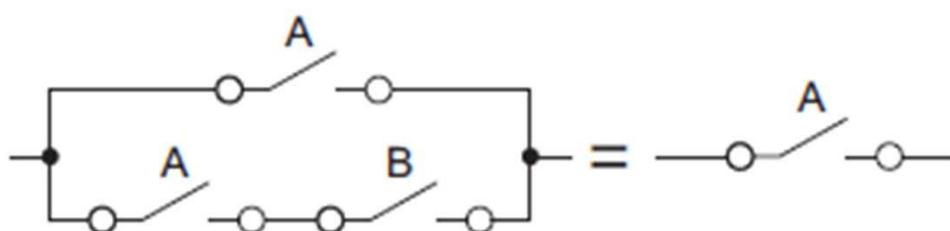
A	B	$S = A \cdot B$
0	0	0
1	0	0
0	1	0
1	1	1

Figura 10.41. Tabla de la verdad de dos contactos NA en serie.

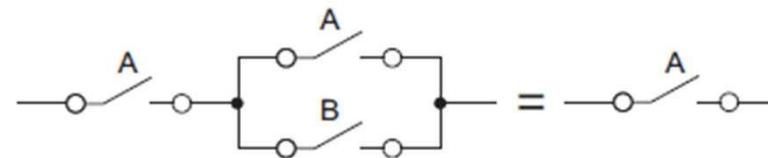
## 13.2. Teoremas del Álgebra de Boole

Principio de dualidad. Cualquier identidad algebraica, deducible a los postulados del Álgebra de Boole, permanece válida, si cambiamos + por · y 1 por 0.

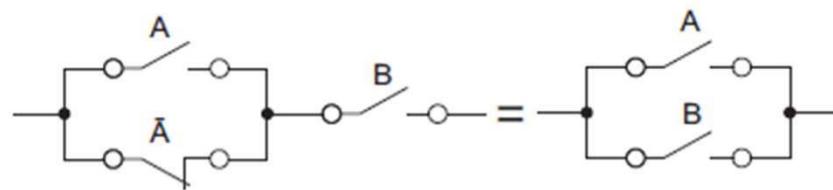
**Teorema 1.**  $A + A - B = A$



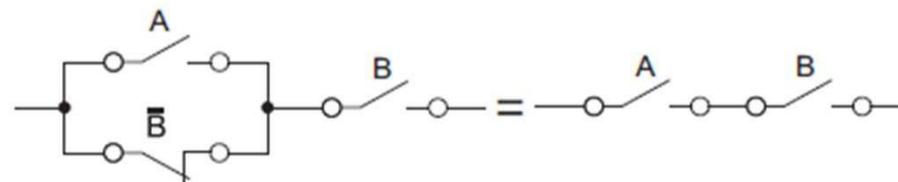
Teorema 2.  $A \cdot (A + B) = A$



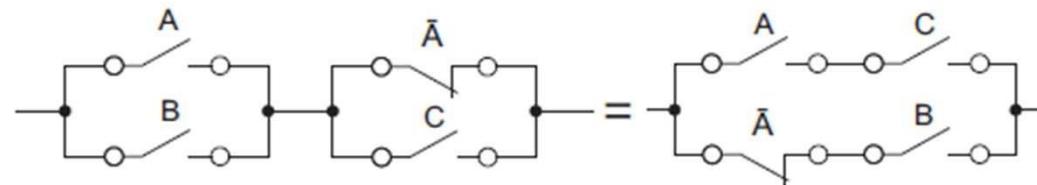
Teorema 3.  $A + \bar{A} \cdot B = A + B$



Teorema 4.  $(A + \bar{B}) \cdot B = A \cdot B$



Teorema 5.  $(A + B) \cdot (\bar{A} + C) = A \cdot C + \bar{A} \cdot B$



## Teoremas de Morgan

Teorema 6. Primer teorema de Morgan.  $\overline{A+B} = \overline{A} \cdot \overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$A+B$	$\overline{A+B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	0	1	1
1	0	0	1	1	0	0
0	1	1	0	1	0	0
1	1	0	0	1	0	0

Tabla 10.3. Tabla de la verdad del teorema 6, primer teorema de Morgan.

Teorema 7. Segundo teorema de Morgan.  $\overline{A \cdot B} = \overline{A} + \overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	1	1	0	1	1
1	0	0	1	0	1	1
0	1	1	0	0	1	1
1	1	0	0	1	0	0

## Teoremas de Morgan

Dentro del álgebra de Boole, los teoremas de Morgan, son ampliamente utilizados en operaciones binarias.

Primer teorema:

El complemento (o la inversa), de un producto lógico de variables, es igual a la suma lógica de las inversas de las variables.

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

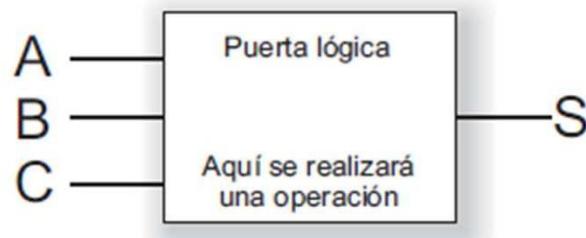
Segundo teorema:

El complemento (o la inversa), de una suma lógica de variables, es igual al producto lógico de las inversas de las variables.

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

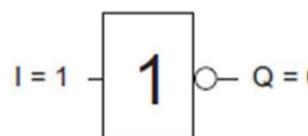
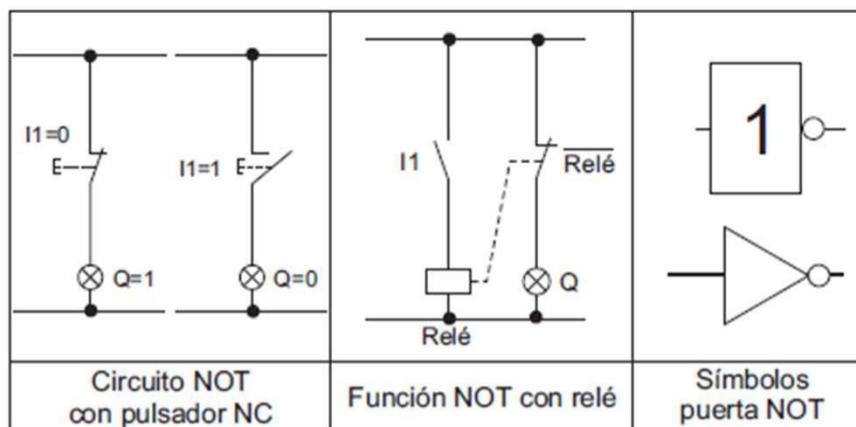
### 13.3. Puertas lógicas

Para representar las funciones lógicas, se utilizan unos símbolos llamados puertas lógicas. Dentro de este bloque funcional se realizará una operación, que produce un resultado sobre una salida.



#### 13.3.1 Función NO o inversión NOT

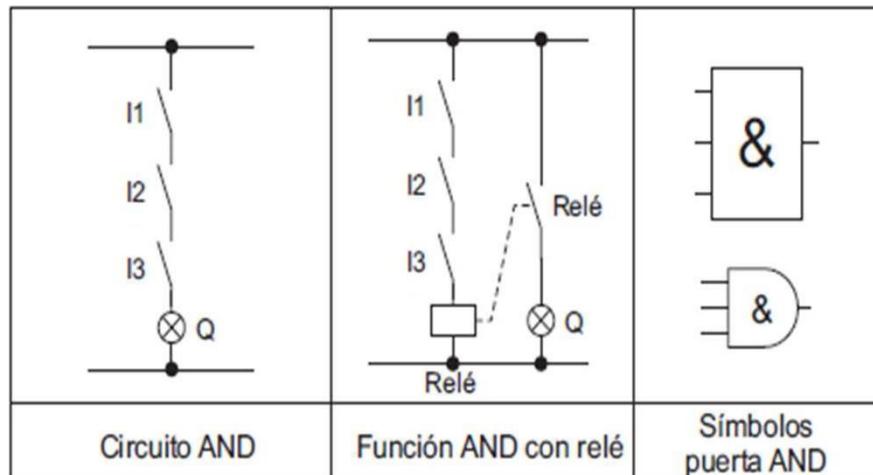
La salida Q, tomará el valor contrario a la entrada.



I	Q = $\overline{I}$
0	1
1	0

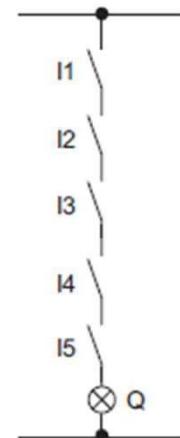
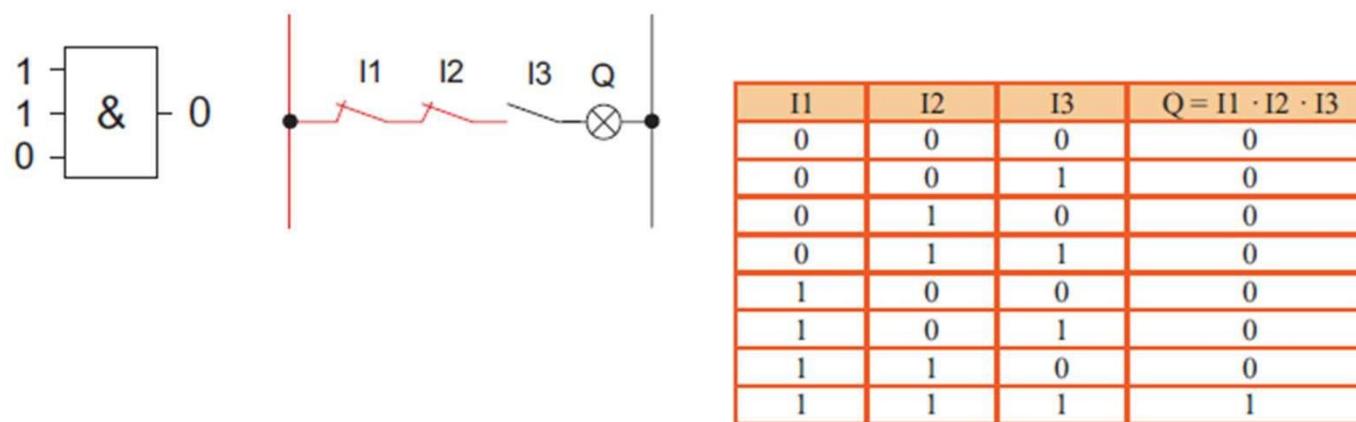
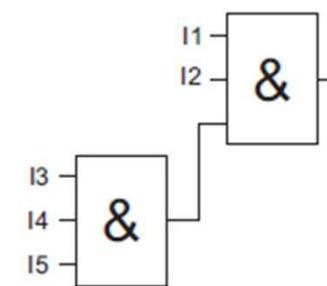
### 13.3.2. Función Y, AND

La salida tendrá valor 1 , cuando todas las variables de entrada tengan valor 1 .



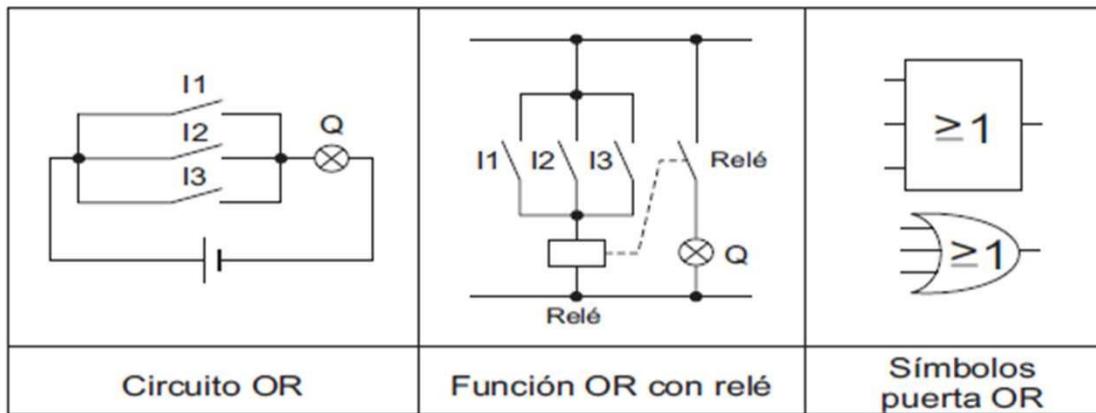
### Puerta AND

Si una función AND requiere de más variables de las que admite la "puerta", se pueden asociar, siendo el mismo efecto que si fuera una única puerta.

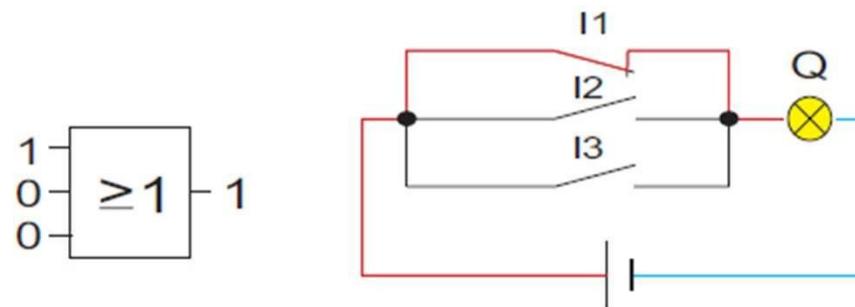


### 13.3.3. Función O, OR

La salida tendrá valor 1 , si cualquiera de las variables de entrada tiene valor 1 .

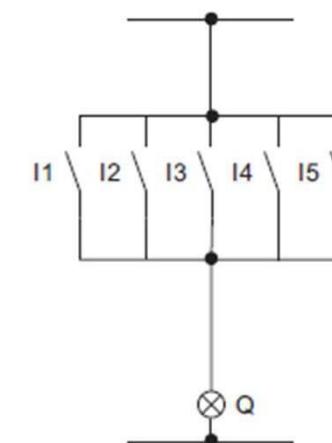
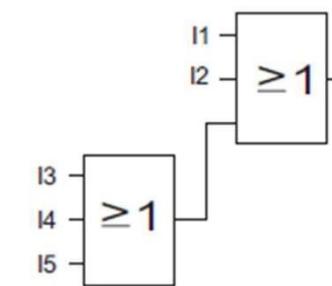


I1	I2	I3	$Q = I_1 + I_2 + I_3$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



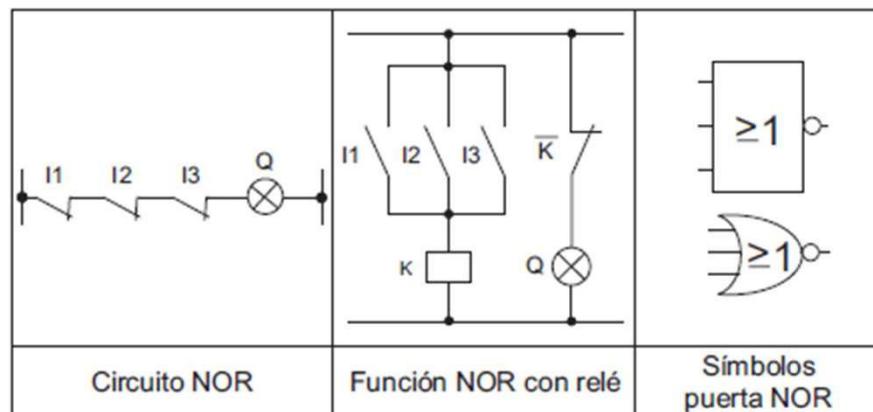
### Puerta OR

Si una función OR requiere de más variables de las que admite la "puerta", se pueden asociar, siendo el mismo efecto que si fuera una única puerta.

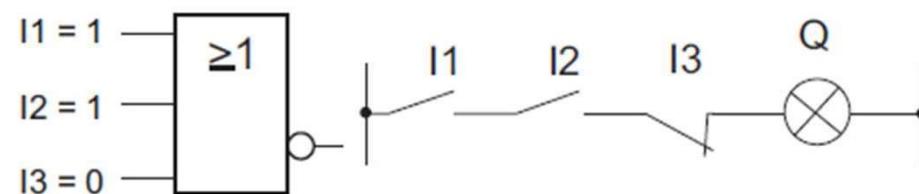


### 13.3.4. Funciones lógicas inversas. NO-O; NOR

La función inversa NOR realiza la misma operación que la función OR, sólo que invierte el resultado.

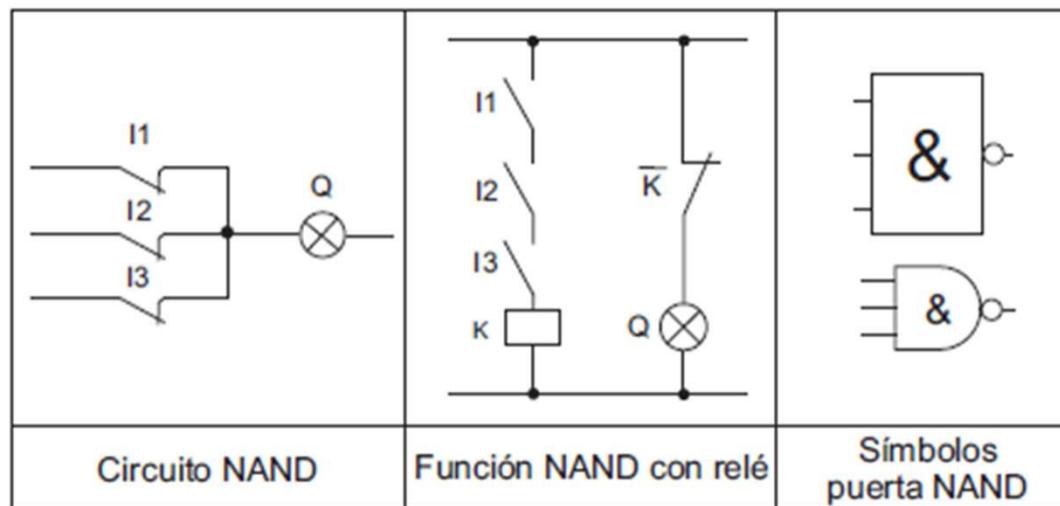


I1	I2	I3	$Q = \overline{I1 + I2 + I3}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

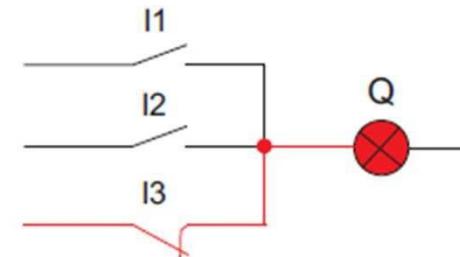
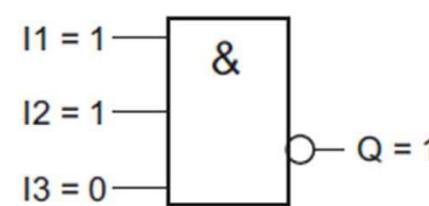


### 13.3.5 Funciones lógicas inversas. NO-Y, NAND

La función inversa NAND realiza la misma operación que la función AND, sólo que invierte el resultado.

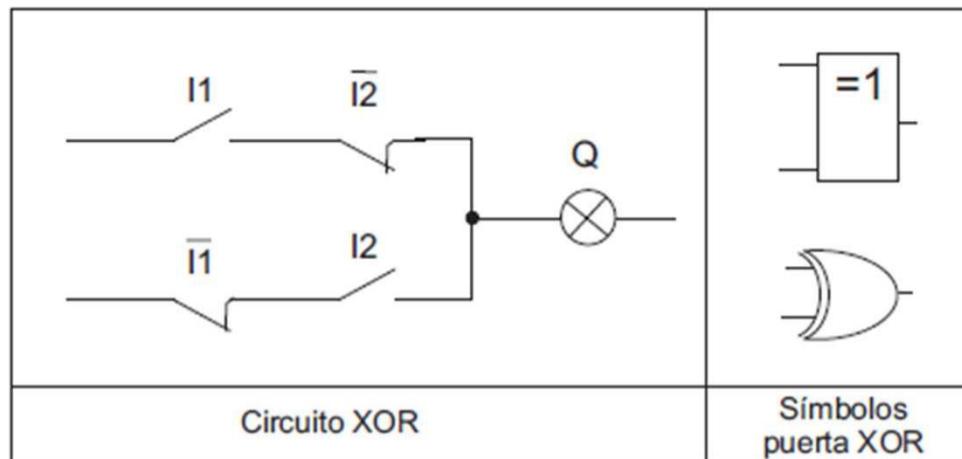


I1	I2	I3	$Q = \overline{I1 \cdot I2 \cdot I3}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

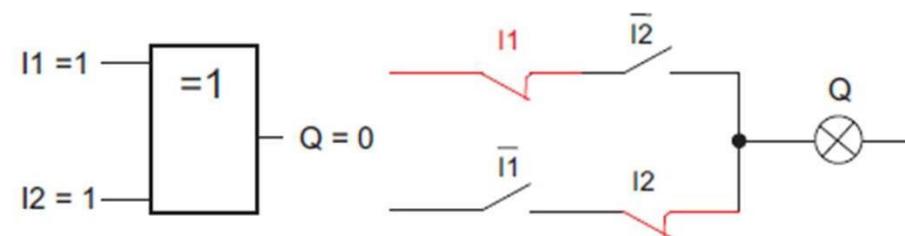


### 13.3.6. Funciones lógicas especiales. OR-Exclusiva; XOR

En primer lugar, esta función sólo tiene dos variables de entrada, y la salida adoptará el valor de 1 cuando el estado de las dos variables sea diferente.



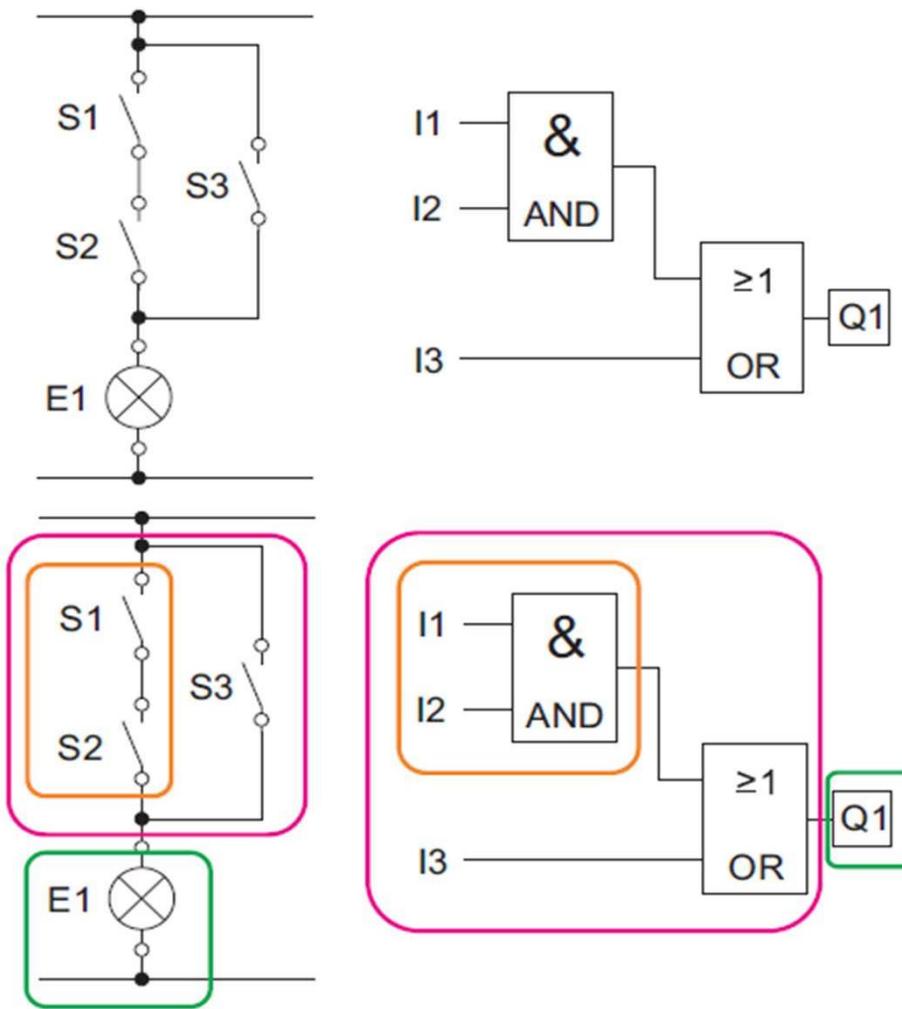
I <sub>1</sub>	I <sub>2</sub>	$Q = I_1 \oplus I_2$
0	0	0
0	1	1
1	0	1
1	1	0



### 13.3.7. Aplicaciones de las puertas lógicas

Como se verá más adelante, las puertas lógicas además de servir de base en el desarrollo de circuitos electrónicos, permite crear un lenguaje de programación para autómatas programables.

Note el siguiente circuito eléctrico, consistente en dos interruptores en serie (S1 y S2), y a su vez, en paralelo con un tercer interruptor (S3). A la derecha se muestra la equivalencia mediante puertas lógicas.



### Tabla de la verdad

Tabla de la verdad del circuito propuesto y equivalencia de sus componentes

I1	I2	I3	Q1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Circuito eléctrico	Funciones lógicas
INTERRUPTOR 1	S1 I1
INTERRUPTOR 2	S3 I2
INTERRUPTOR 3	S3 I3
LÁMPARA	E1 Q1

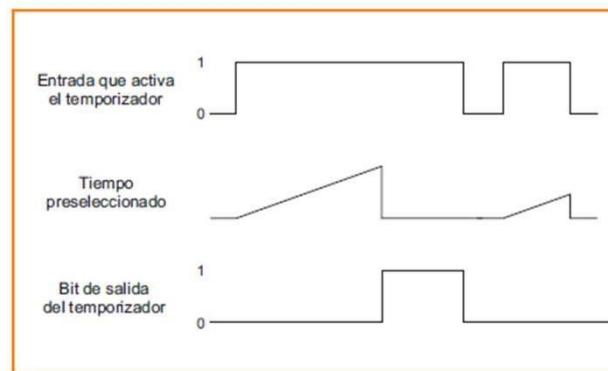
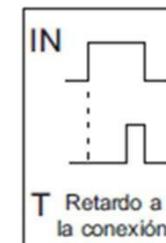
## 14 Lenguajes de programación

Para configurar las acciones que ha de desarrollar el autómata programable, el diseñador dispone de unas herramientas que le van a permitir además de programar: diagnosticar, modificar, comprobar, simular las aplicaciones, incluso antes de volcarlas al PLC. El lenguaje de programación maneja el juego de instrucciones que realizarán las funciones lógicas y de cálculo de la unidad central de proceso.

#### 14.1.1. Temporizadores

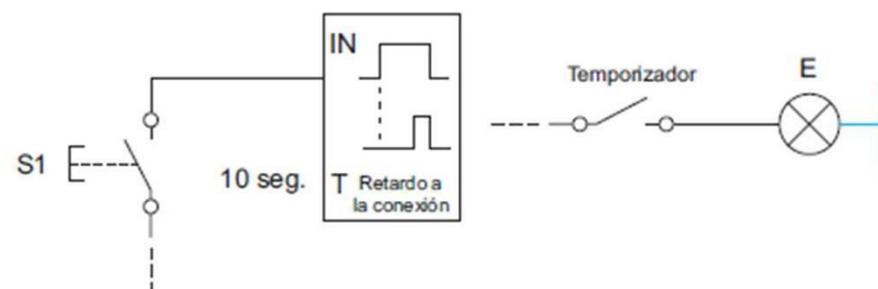
##### Temporizador con retardo a la activación TON

Este temporizador activará su salida, si es excitada su entrada (IN) y pasa el tiempo programado, aunque si la entrada es igual a cero, la salida también lo será. Normalmente, de este temporizador usaremos dos variables; el bit digital de salida (T), que podrá ser 0/1, y el dato del tiempo actual, medido entre otros, en formato de 16 bits (Word).



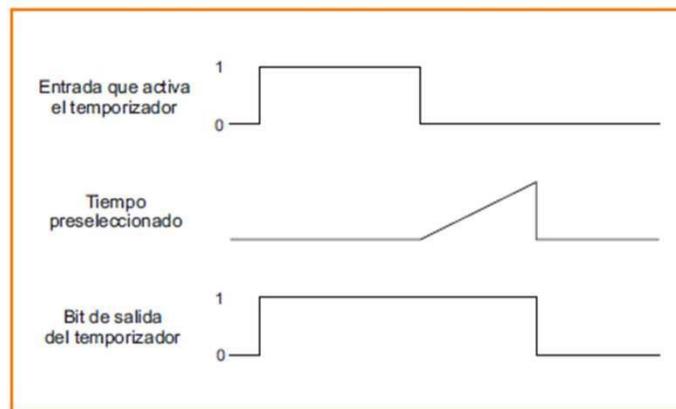
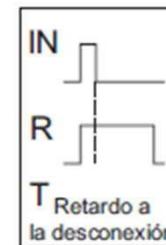
##### Ejemplo

El pulsador S1 activa el temporizador con retardo a la activación. ¿Qué pasará a los 10 segundos de activar la entrada del citado temporizador?



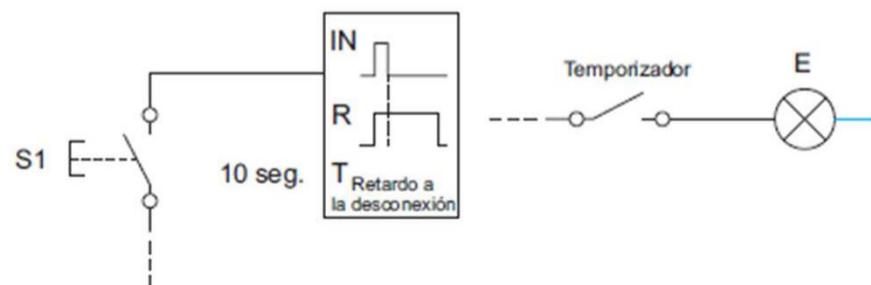
## Temporizador con retardo a la desactivación TOF

Este temporizador activará su salida en el mismo instante que se excita su entrada (IN). Una vez la entrada no está activa, comienza a contar el tiempo que resta para la desactivación de la salida. Además, el temporizador puede contar con una entrada RESET (R), es decir, puesta a cero de la salida y el tiempo. Normalmente, de este temporizador usaremos dos variables; el bit digital de salida (T), que podrá ser 0/1, y el dato del tiempo actual, medido entre otros, en formato de 16 bits.



### Ejemplo

El pulsador S1 activa el temporizador con retardo a la desactivación, y en ese mismo instante, se activa la lámpara E. ¿Qué pasará a los 10 segundos de activar la entrada del citado temporizador?



#### 14.1.2. Función SET-RESET (RS)

El bloque funcional set-reset, implica a su salida un 1, si en la entrada SET ha existido un impulso (1), es decir, el circuito se realimenta. Observe el cronograma.

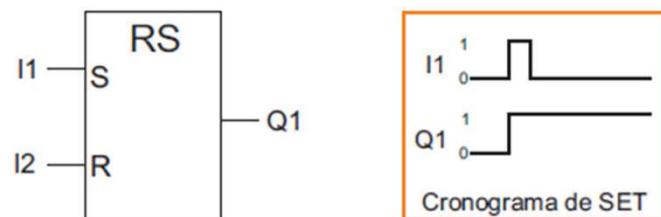
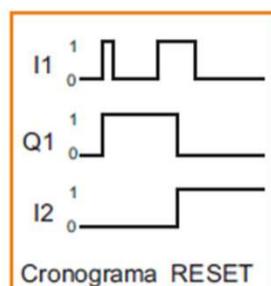


Figura 10.65. Bloque RS, y cronograma función SET.

Sin embargo, si el impulso (1) es aplicado en la entrada RESET, obliga a que su salida sea 0. Note el cronograma.



Ejemplo

El pulsador S1 activa la entrada SET del bloque. Pasado un tiempo, son presionados a la vez los pulsadores S1 y S2. ¿Qué le ha pasado a la lámpara E?

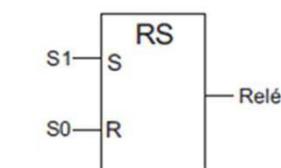
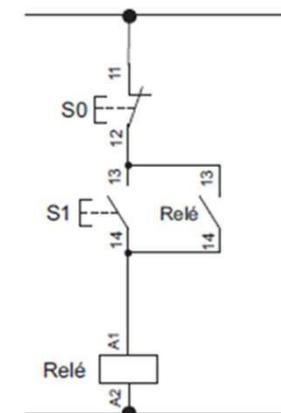
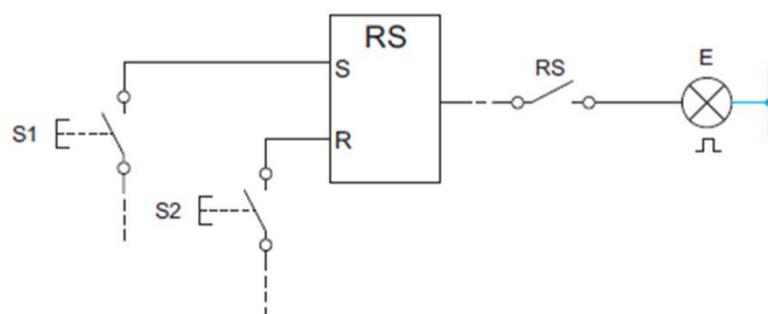
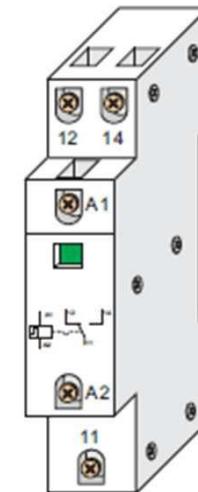
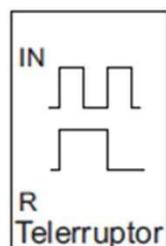


Figura 10.68. Comparación circuito con realimentación a la función SET-RESET.

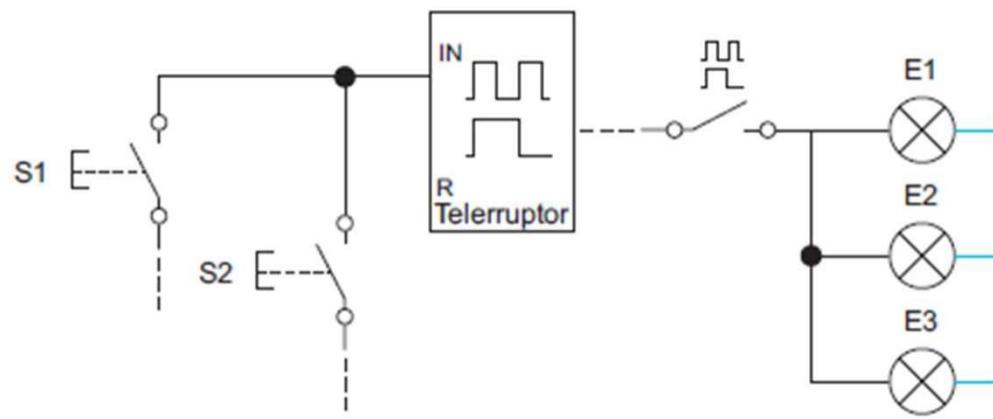
### 14.1.3. Telerruptor

El bloque funcional telerruptor, tiene la misma utilidad que el telerruptor físico; con un impulso (IN) la salida será 1 y con un nuevo impulso, será 0. Además, cuenta con una entrada RESET, que obliga la salida a 0. Para los sistemas de programación que no dispongan de bloque telerruptor, la construcción de éste con funciones convencionales es simple.



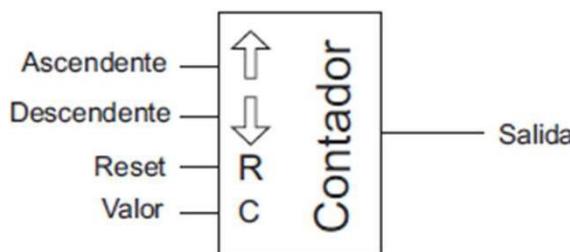
## Ejemplo

Desde varios pulsadores en paralelo ( $S_1, S_2\dots$ ), dispuestos en una nave industrial, se podrá conectar-desconectar el alumbrado, que está constituido por luminarias en paralelo ( $E_1, E_2\dots$ ).

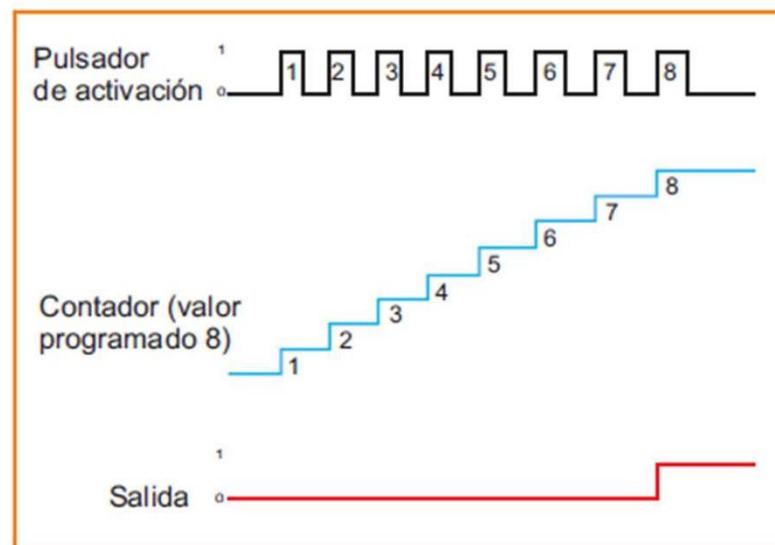


#### 14.1.6. Contadores

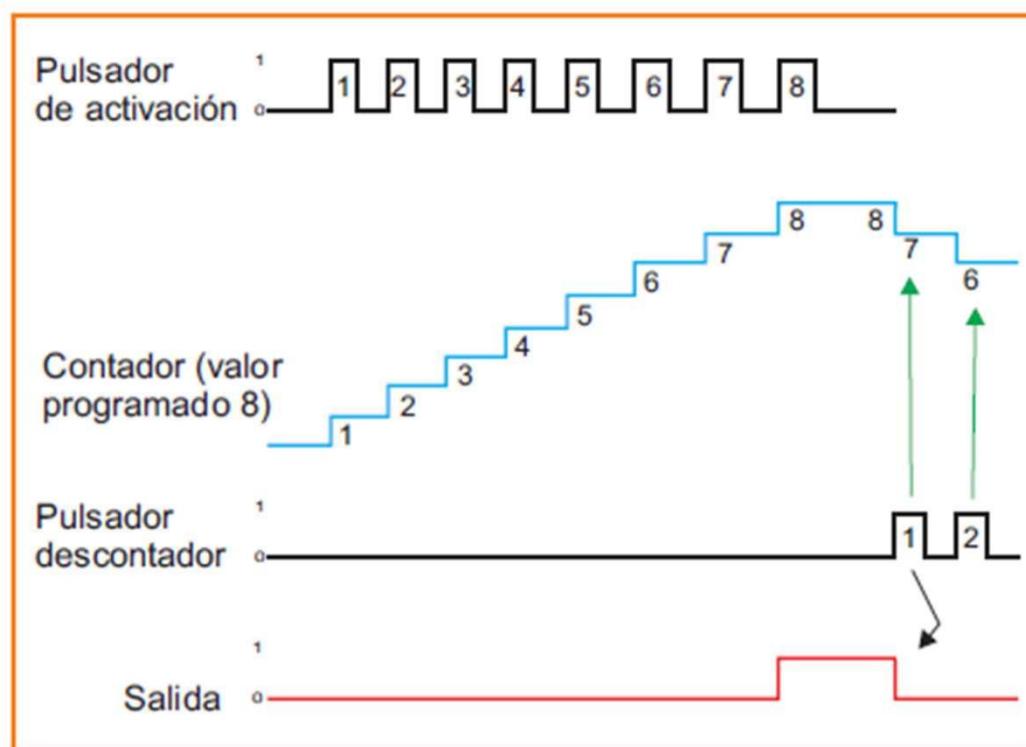
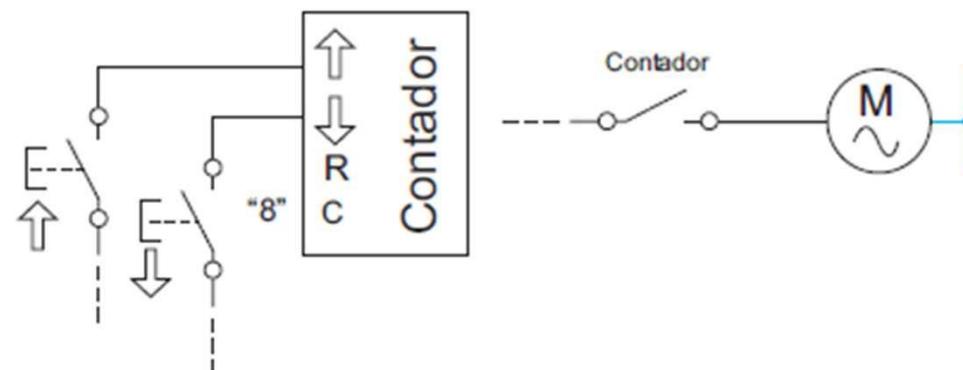
Un contador realizará una acción, cuando una entrada sea activada un número programado de veces. Técnicamente, los contadores activarán o desactivarán una salida, o un bit (bit a 1 ó bit a 0) cuando se alcanza un número predeterminado de conexiones en su entrada. Este número de activaciones, es la programación. Los contadores podrán computar en positivo, en negativo o en ambos sentidos. Además, también pueden contar con una entrada de RESET, que pone el cómputo y la salida a cero.



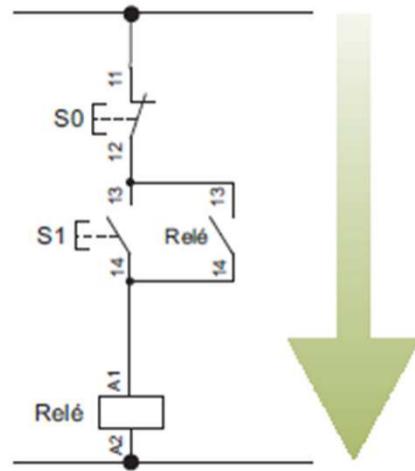
Por ejemplo, cuando la entrada del contador se conecte 8 veces, se activará la salida Q.



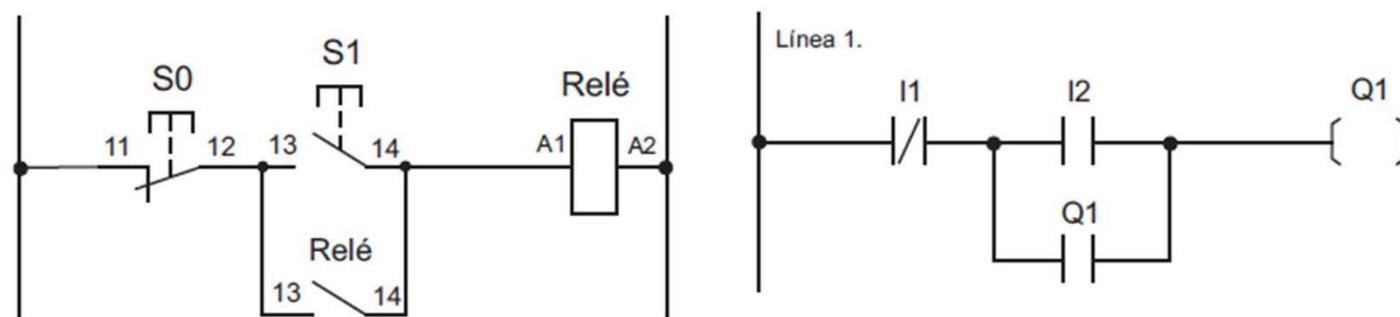
En el siguiente gráfico se muestra como el cómputo llega a 8, la salida se activa, pero el descontador baja el cómputo a un valor inferior a 8, con lo cual, la salida se desconecta.



La semejanza de este sistema de programación a los circuitos eléctricos basados en relés y contactores, aporta en el diseño cierta facilidad, ya que partimos de la base, de que a la bobina, tiene que llegarle corriente eléctrica. Nos referimos a circuitos en escalera.



La conversión por tanto entre un esquema de relés y lenguaje LD, es sencilla, aunque hay que buscar la equivalencia entre los componentes.



### Símbolos LD

En el lenguaje de programación LD, los símbolos (que son elementos de programación aunque sean parecidos a esquemas eléctricos) más usados son:

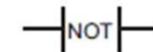
Contacto abierto



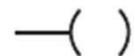
Contacto cerrado



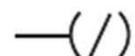
Negación



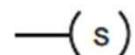
Bobina directa



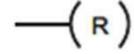
Bobina inversa

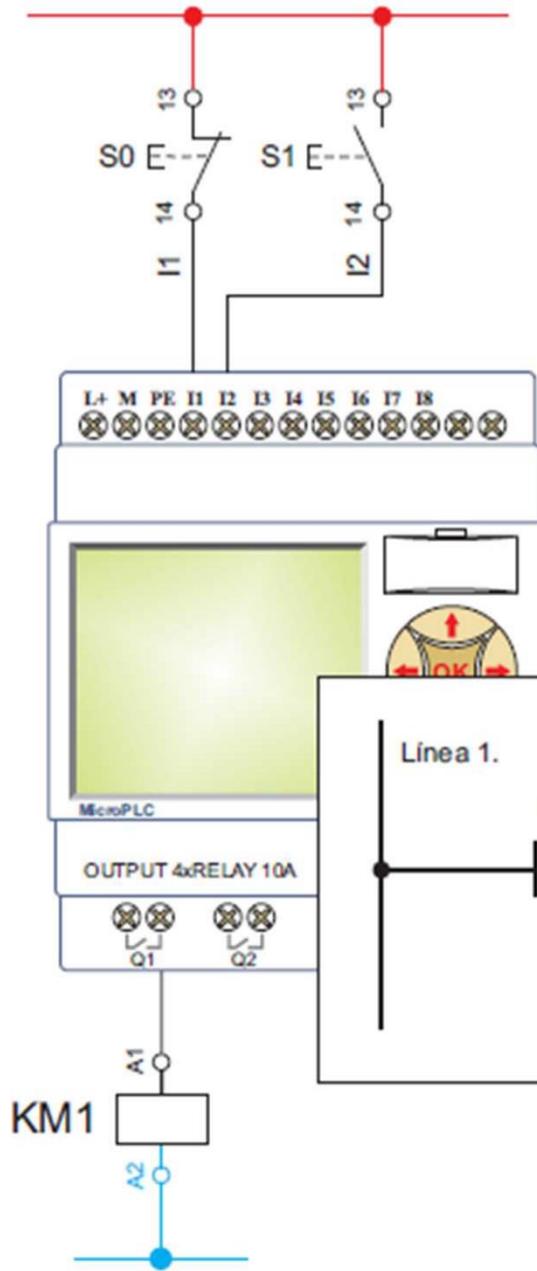


Activación de bobina en SET



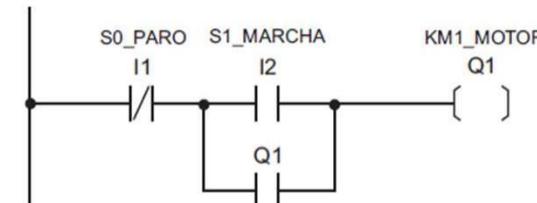
Desactivación de bobina en RESET





Según este esquema eléctrico de cableado del PLC, deben quedar claros unos conceptos:

- El pulsador S0 **NO ES** el contacto I1, ya que S0 es físico e I1 es la representación virtual de S0 en el programa. De otro modo; un contacto físico exterior de un pulsador, final de carrera, detector, etc. No tiene porqué representarse con la misma condición de abierto o cerrado en el programa. Por tanto, I1, es el terminal o conector que une el pulsador de paro S0 al autómata, y por donde el pulsador le aporta información de su estado -abierto o cerrado- al PLC.
- Del mismo modo, I2 **NO ES** S1, sino una representación del estado que tenga actualmente el pulsador S1.

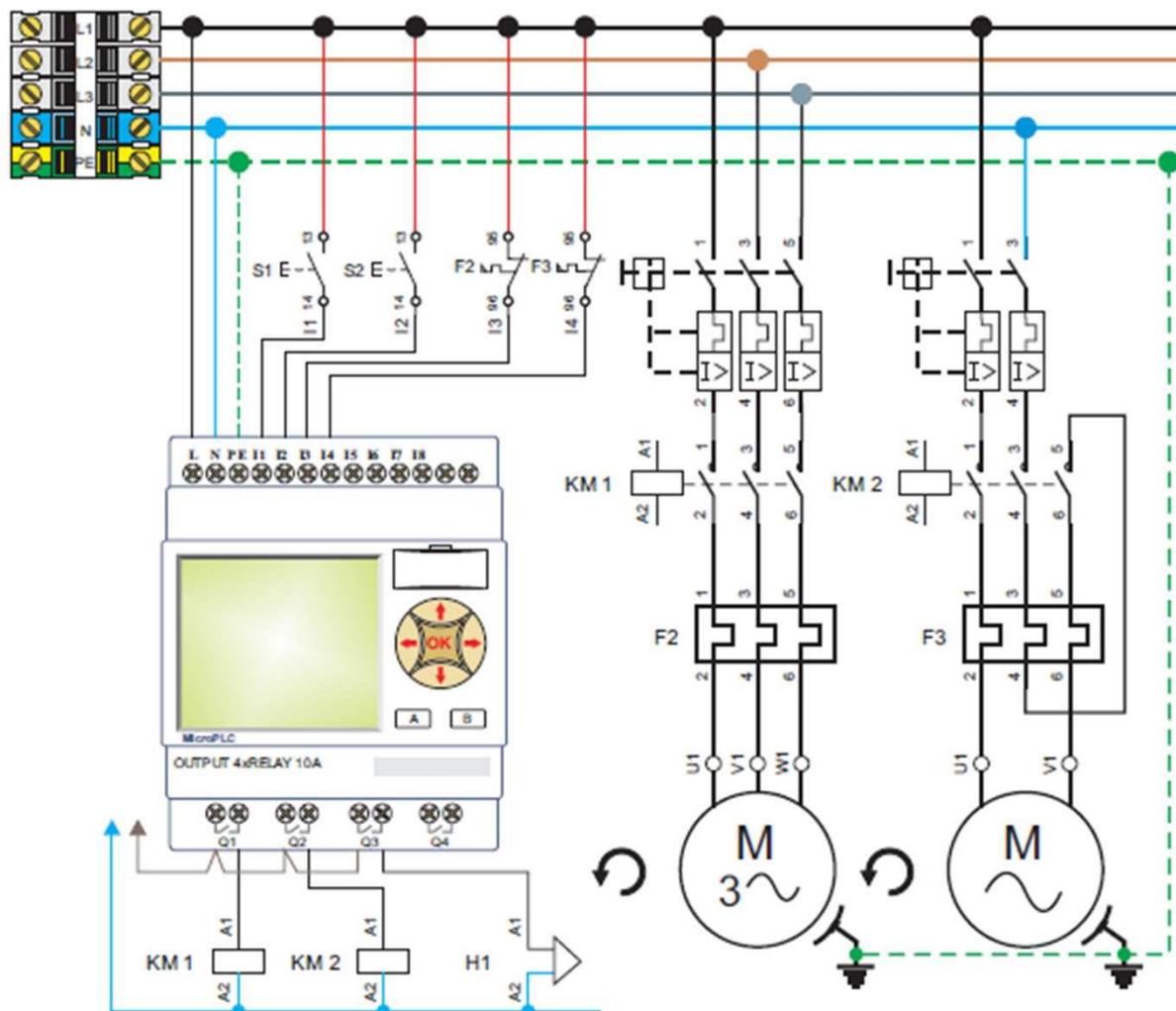


SÍMBOLO	VARIABLE	COMENTARIO
S0_PARO	I1	Pulsador de parada
S1_MARCHA	I2	Pulsador de marcha
KM1_MOTOR	Q1	Motor trifásico en caldera

## Ejercicio

Un pulsador S1, pone en marcha un sistema de riego, que consiste en la activación de un motor-bomba trifásico, por un tiempo total de 5 horas, pasadas las cuales, se conectarán un segundo motor-bomba monofásico, al tiempo que se detiene el primer motor. El segundo motor funcionará un tiempo máximo de 4 horas. Existirá un pulsador de parada S2, además de protección por relé térmico de cada motor.

SÍMBOLO	VARIABLE	COMENTARIO
S1_MARCHA	I1	Pulsador de marcha
S2_PARO	I2	Pulsador de parada
F2_RT_MOT_TRIFÁS.	I3	Relé térmico motor trifásico
F3_RT_MOT_MONO.	I4	Relé térmico motor monofásico
MOTOR1	Q1	Motor trifásico
MOTOR2	Q2	Motor monofásico
AVERÍA	Q3	Aviso por avería



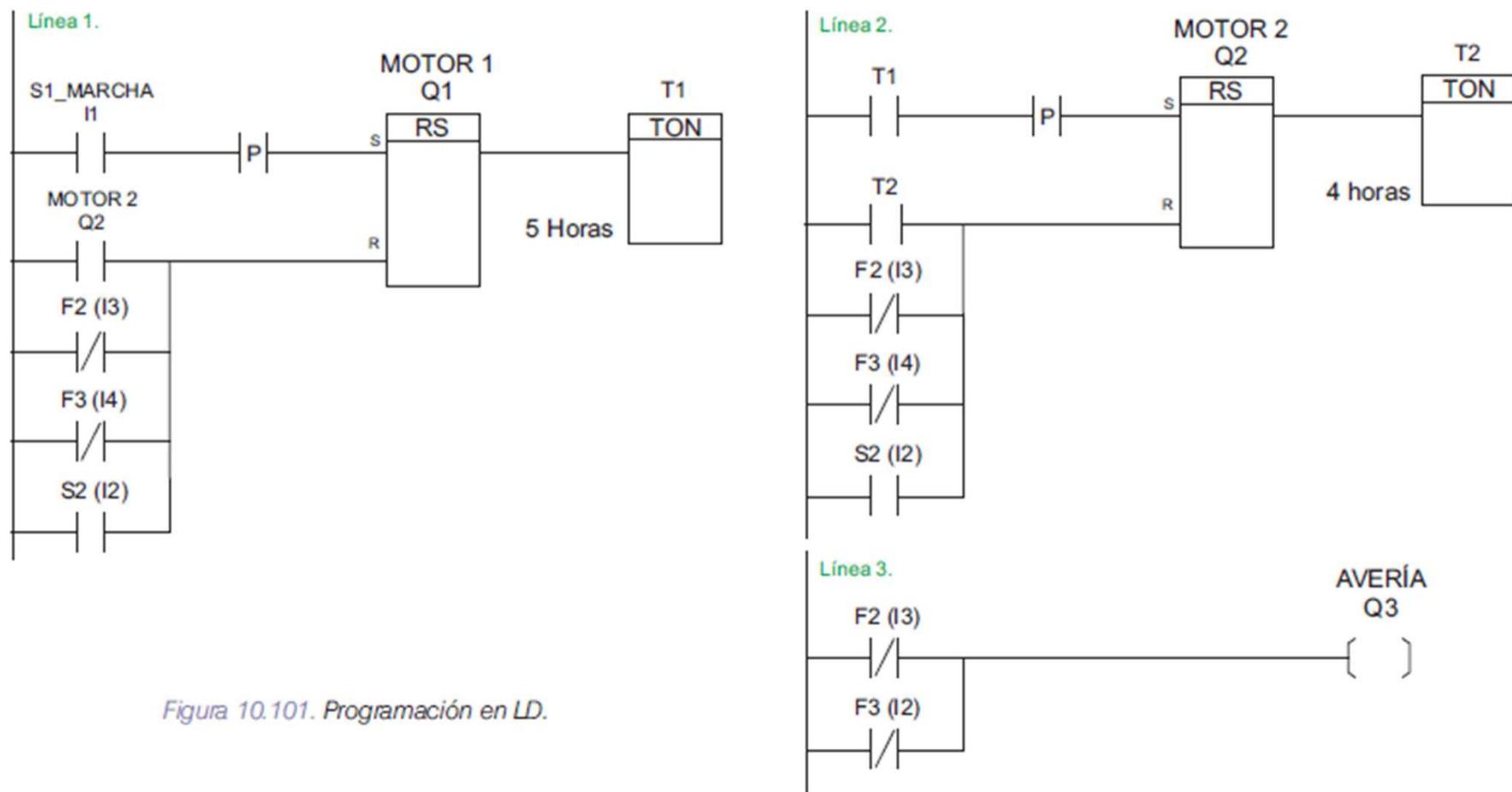
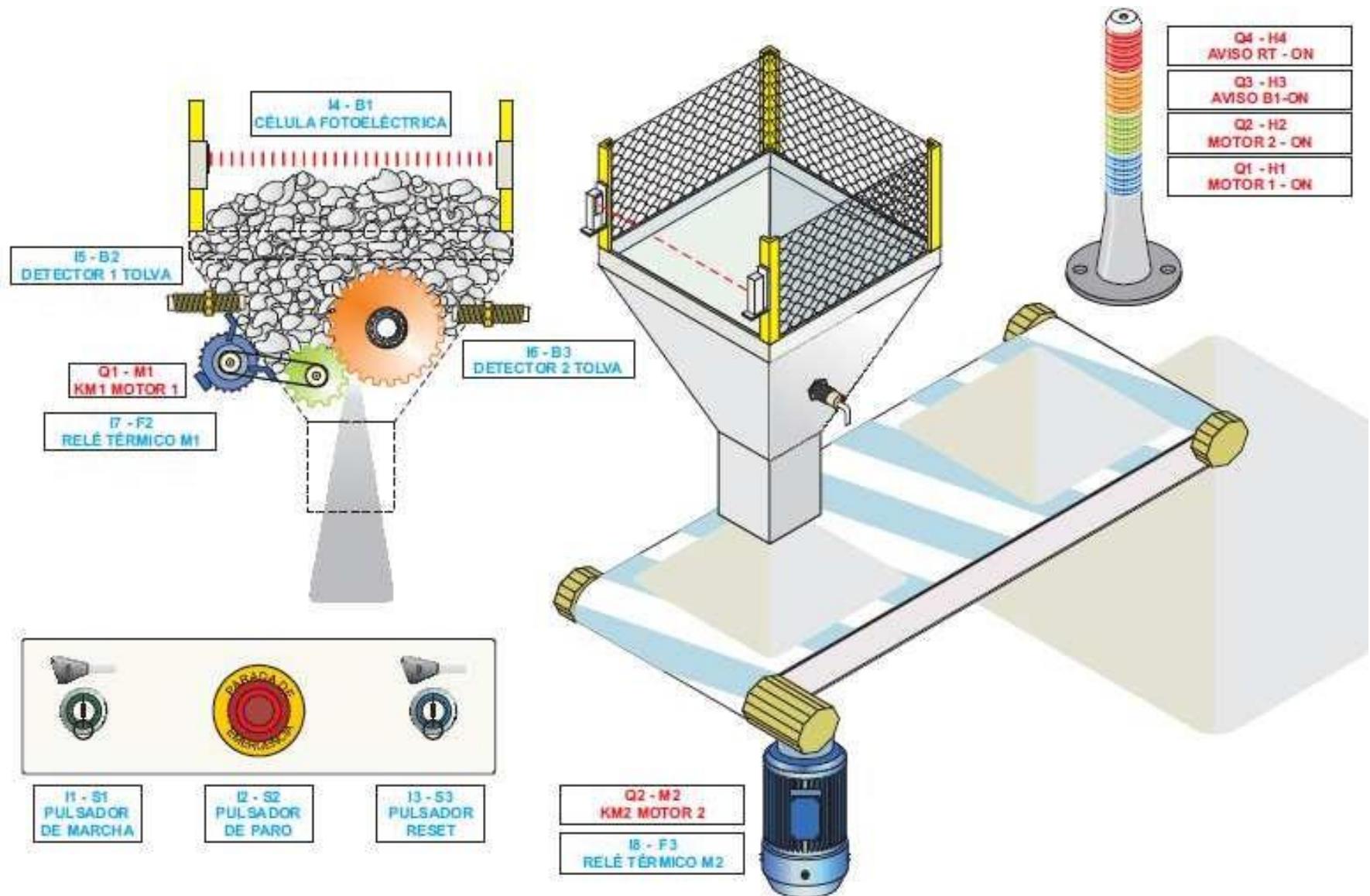


Figura 10.101. Programación en LD.



SÍMBOLO	VARIABLE	COMENTARIO
S1	I1	Pulsador de marcha
S2	I2	Pulsador de paro. Seta
S3	I3	Pulsador de reset
B1	I4	Célula fotoeléctrica
B2	I5	Primer detector tolva
B3	I6	Segundo detector tolva
F2	I7	Relé térmico motor tolva
F3	I8	Relé térmico motor cinta
MOTOR1	Q1	Motor tolva
MOTOR2	Q2	Motor cinta transportadora
H3	Q3	Aviso, célula fotoeléctrica activa
H4	Q4	Aviso, algún relé térmico activo

