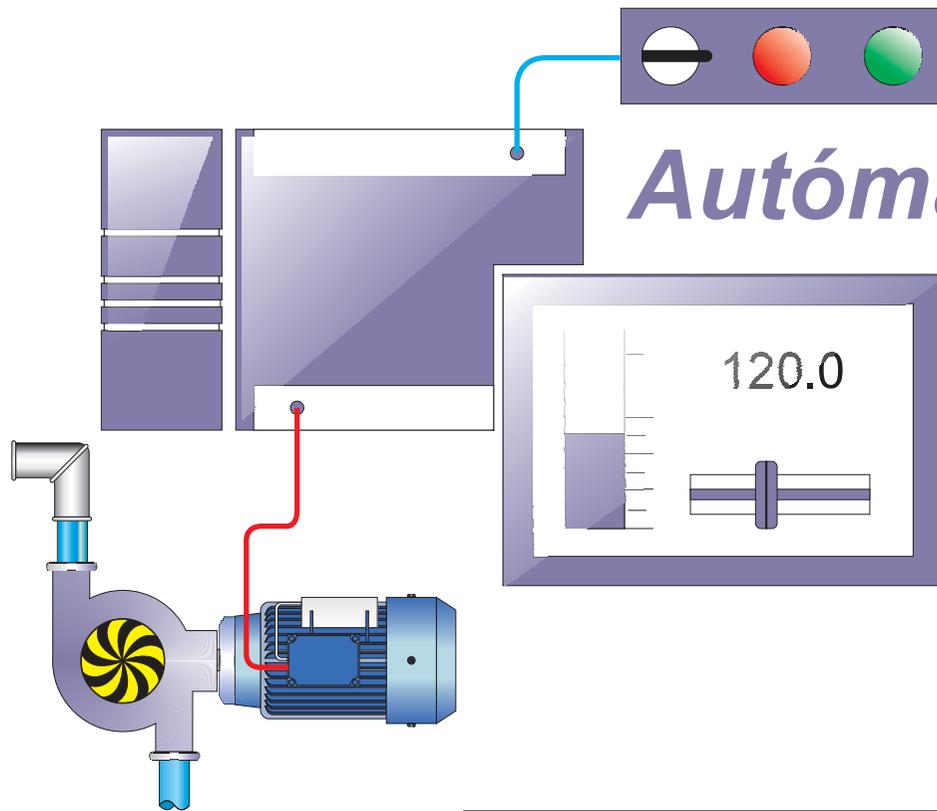


# Introducción a los autómatas programables



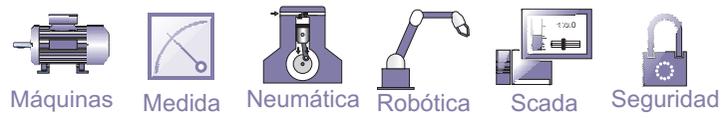
## *Autómatas Programables*



¡¡AVISO!! El conexionado de componentes mostrado en este documento es meramente ilustrativo. Para el cableado real de dispositivos, consulte el manual del fabricante.



# Recursos [aulaelectrica.es](http://aulaelectrica.es)

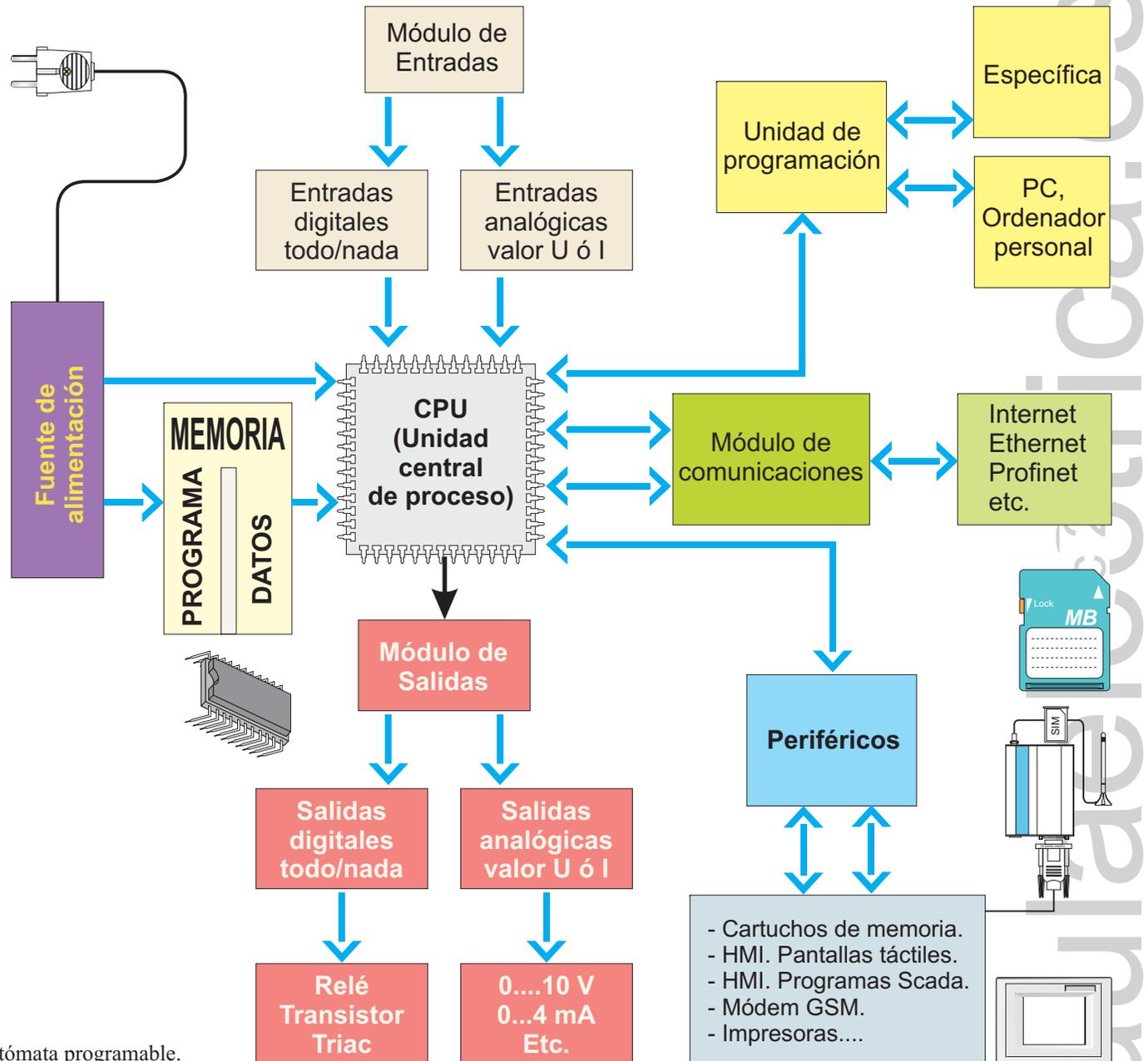


### 1 El autómata programable

Desde el comienzo de la industrialización, el hombre a 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 sustituya los sistemas de control cableados (interruptores, relés, contactores..), por un control programado, con el objetivo de ahorrar costes en los procesos de fabricación.

El crecimiento fue rápido, y los autómatas programables industriales aumentaban en funcionalidad y rapidez de operación. La figura muestra el esquema de bloques de un PLC.

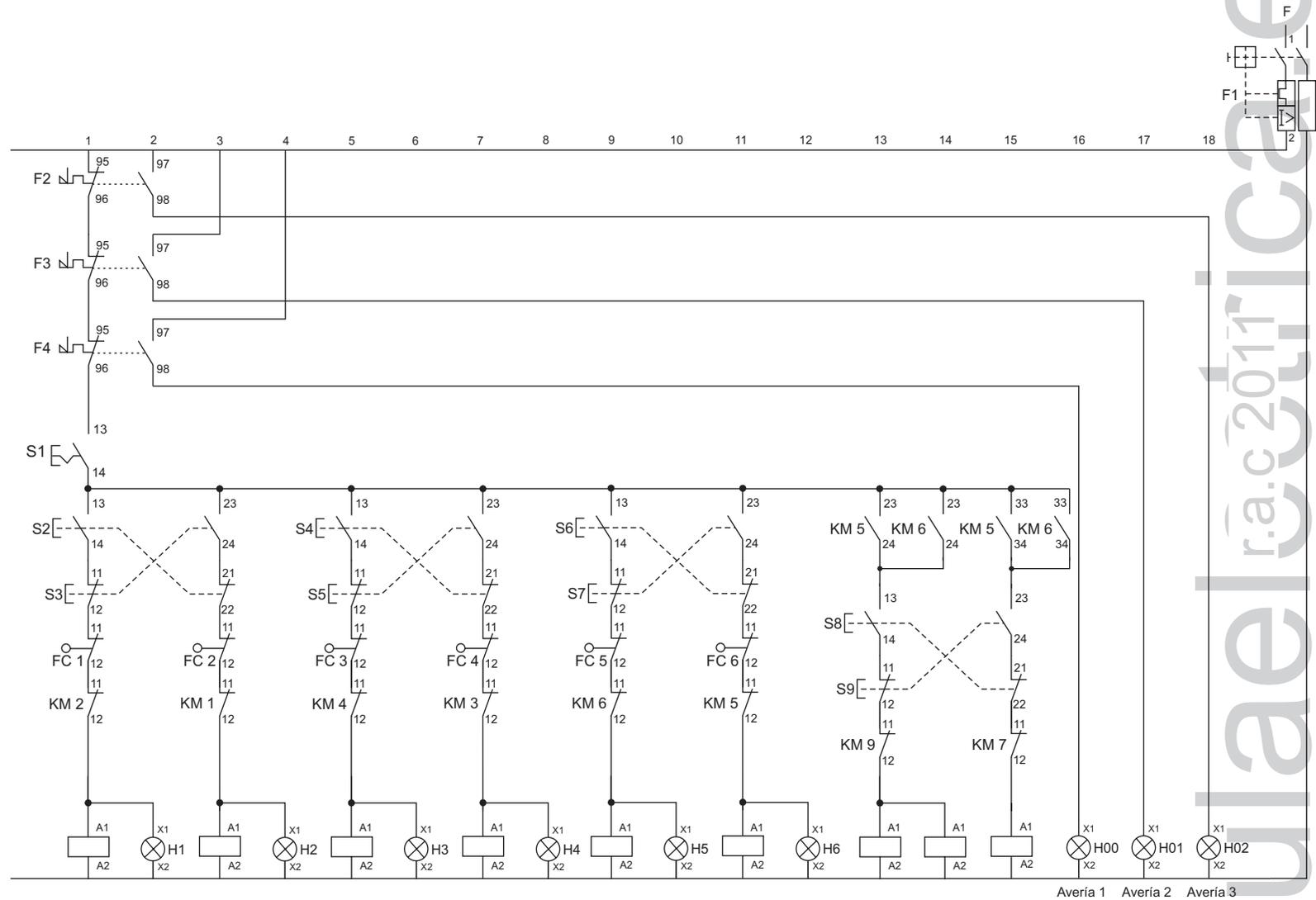


Esquema de bloques de un autómata programable.

## 2 Lógica cableada

Cuando el funcionamiento de una operación automática se realiza con la alimentación principal de relés y contactores, producidas o provocadas por los accionamientos de elementos mecánicos tipo interruptor, pulsador, final de carrera, etc., 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 a gobernar cuenta con varios sensores de entrada, y la misma es vulnerable de modificar por razones funcionales o de producción, sale más rentable utilizar un microcontrolador que realizar la instalación con lógica cableada. Por contra, si la instalación es permanente con un proceso de funcionamiento sin proyectos de modificación, no es necesario que la gestione un PLC.

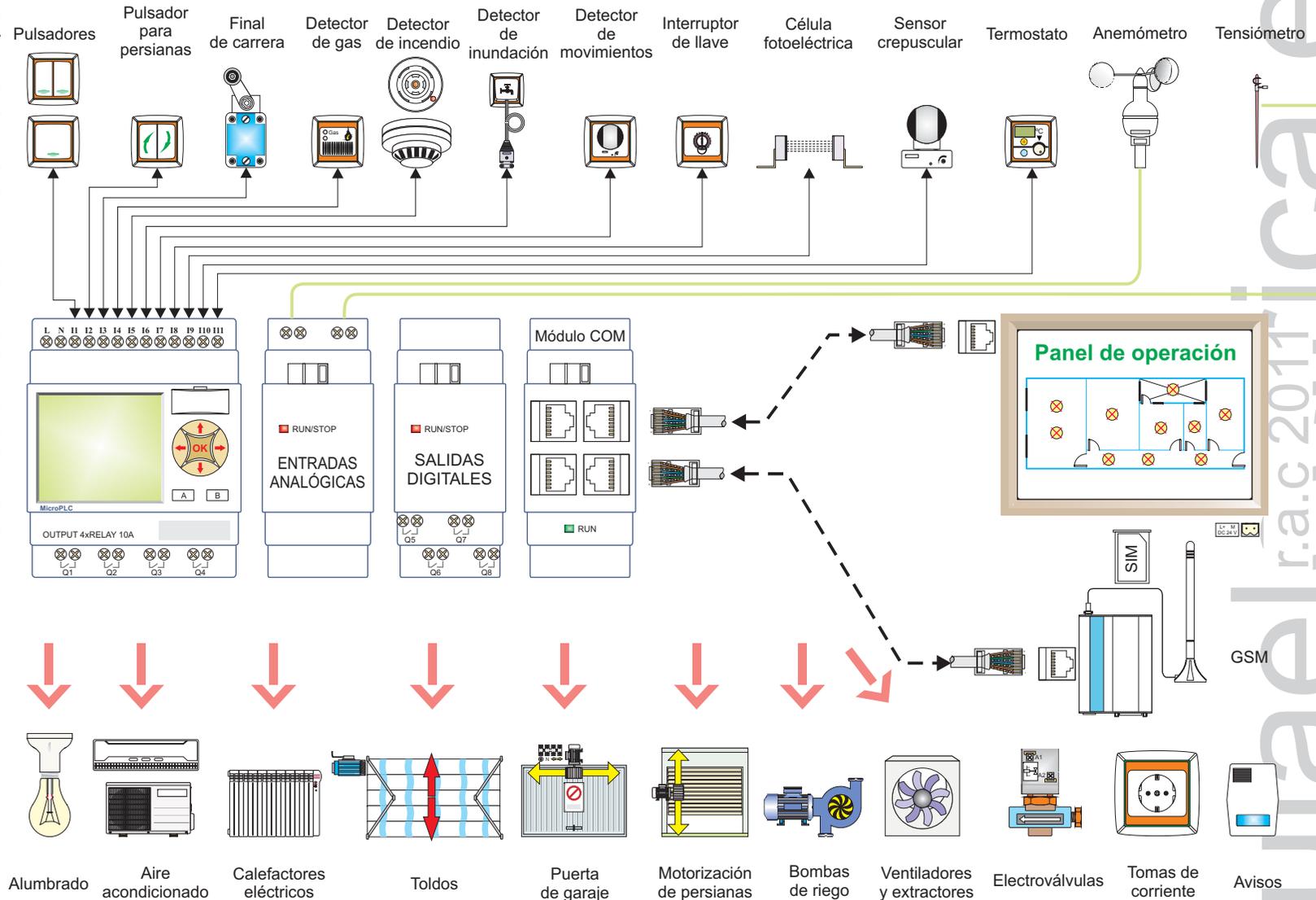


Lógica cableada.

aulaenr.a.c 2011caes

### 3 Lógica programada

Si el número de captadores es amplio, aunque creamos que la instalación no va a sufrir modificaciones, el coste del microcontrolador será insignificante si sólo una vez decidiéramos realizar una modificación de control (recableado, pruebas, puesta en marcha, verificación, tiempo perdido, parada de producción, etc.). Por ejemplo, si decidimos controlar las lámparas de los semáforos de un cruce de dos calles, la instalación la pueden realizar automatismos convencionales, aunque sean varios; 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 las posibilidades de programación de un autómata.



Lógica programada.

auraelra.c 2011 ca.es

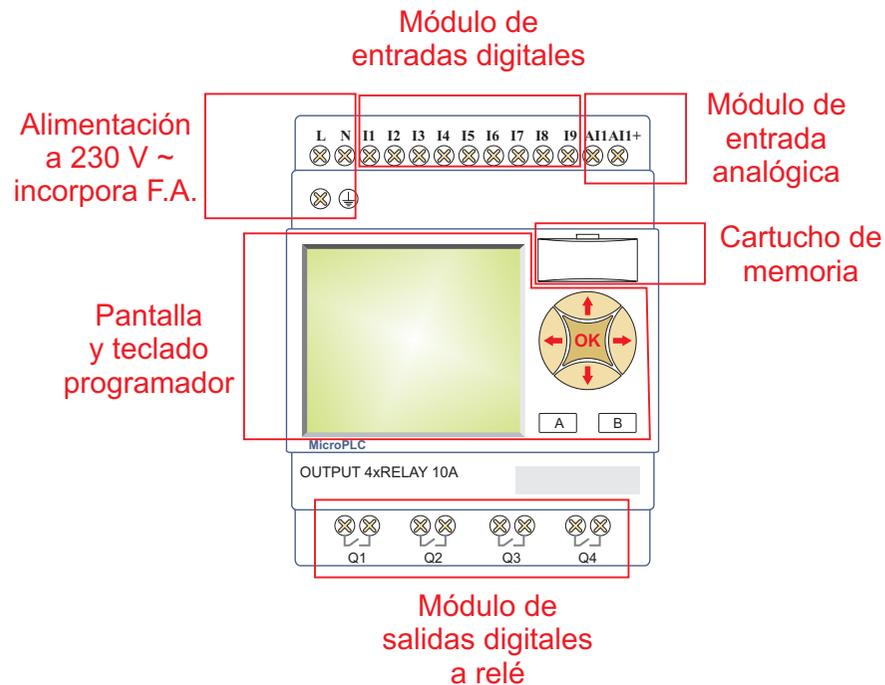
#### 4 MicroPLCs

El autómata programable es designado también como Controlador Lógico Programable (PLC).

La referencia de microPLC no está claramente determinada.

Algunas empresas afirman que un PLC será "micro", si el número de entradas y salidas que gobierna no es superior a 32.

Observe la composición de un microPLC, "estándar", que podemos buscar en el mercado. Incorpora en un sólo módulo, la mayoría de los componentes básicos para su funcionamiento, es decir, fuente de alimentación, entradas digitales, una o varias entradas analógicas, cartucho de memoria donde guardar los programas de usuario, salidas digitales, y lo más práctico, una pantalla y teclado programador que evita el uso de una unidad de programación. Adicionalmente, el modelo permitirá la ampliación de módulos para entradas/salidas, módulo GSM, módulo de comunicaciones, etc.



#### 5 Módulos de entrada

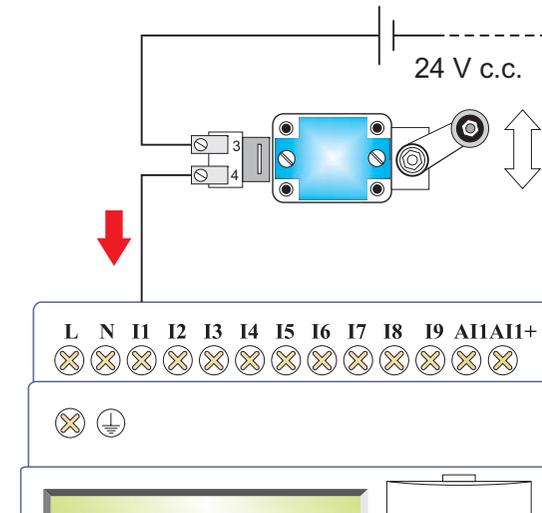
Los módulos de entrada podrán recibir principalmente señales digitales todo/nada, o analógicas en formato tensión (ejemplo 0...10 V) o intensidad (ejemplo 0...10 mA).

##### 5.1. 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.
- Detectores capacitivos, inductivos o fotoeléctricos.
- Etcétera.

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.



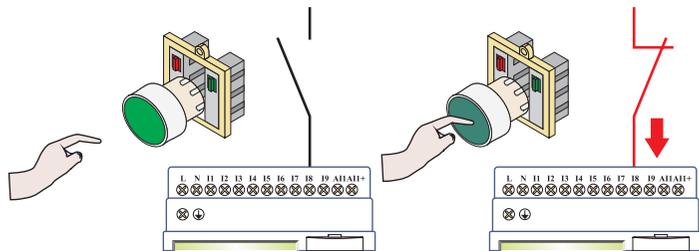
El final de carrera implementa un valor de tensión en la entrada, cuando es activado.

aulaenrca.com

Los sensores que aportan señales digitales todo/nada, pueden a su vez ser pasivos y activos.

→ Captadores pasivos. Funcionan preferentemente con un movimiento mecánico y no necesitan de una fuente de energía adicional para estar operativos. Entre ellos, interruptores, pulsadores y finales de carrera. En esencia, el movimiento ejercido sobre el dispositivo conmutará uno o varios contactos, que son los que permiten el fluido de corriente eléctrica.

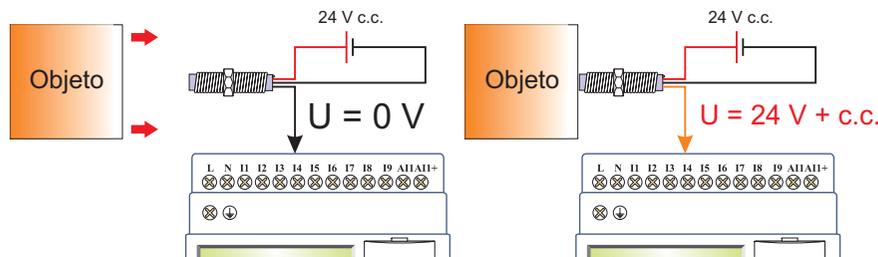
Note en la imagen, que cuando es presionado el pulsador, se cierra un contacto del mismo, que permite el paso de corriente al PLC, y por tanto el envío de una señal que será analizada por el programa.



El pulsador es un captador pasivo.

→ Captadores activos. Requieren de una fuente adicional de energía para operar. Algunos son: detectores capacitivos, inductivos, células fotoeléctricas...

Por 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 c.c.



El detector capacitivo es activo. Necesita alimentación.

### 5.2. 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.

Los valores estándar de tensión son:

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

Los valores estándar de intensidad son:

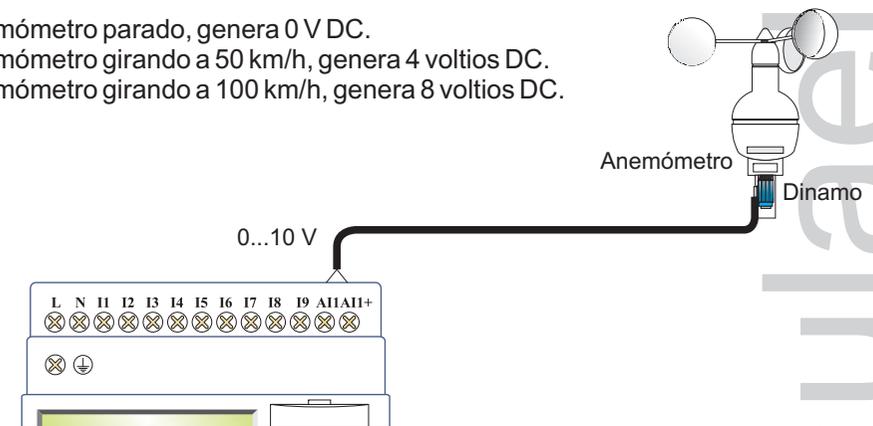
- 0 a 20 mA.**
- 4 a 20 mA.
- +1 a -5 mA
- 0 a +5 mA.

El técnico debe calibrar la señal procedente del sensor de forma correcta para evitar que la lectura sea errónea.

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



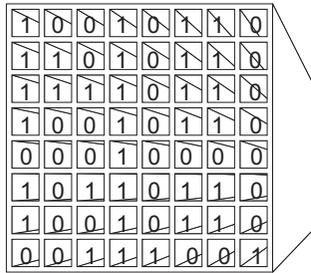
El anemómetro se conecta a la entrada analógica del autómata.

## 6 Variables

En un autómata programable, y en programación en general, una variable es un “lugar” donde se guardan cierto tipo de datos.

Los datos podrán ser diversos; textuales, imágenes, sonido, etc. En autómatas programables, los datos se guardan en formato de bit, byte, palabra y doble palabra preferentemente.

La llamada a una variable es inequívoca, de tal forma que no existirán dos variables con el mismo nombre.



Representación del almacén de datos.

### 6.1. Variables de entrada

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, etc. Estas variables operan con datos tipo bit (0 ó 1), es decir, todo o nada, activado o no activado. También se llaman datos Booleanos.

Las variables analógicas necesitan más capacidad de almacenaje, ya que los valores equivalentes a la magnitud medida pueden ser infinitos. Por ejemplo, la variable AI 1 (entrada analógica 1), utilizará almacén de datos en formato Real, esto es, 32 bits.

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

## 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.

Las salidas analógicas aportarán señal de control o visualización, principalmente para:

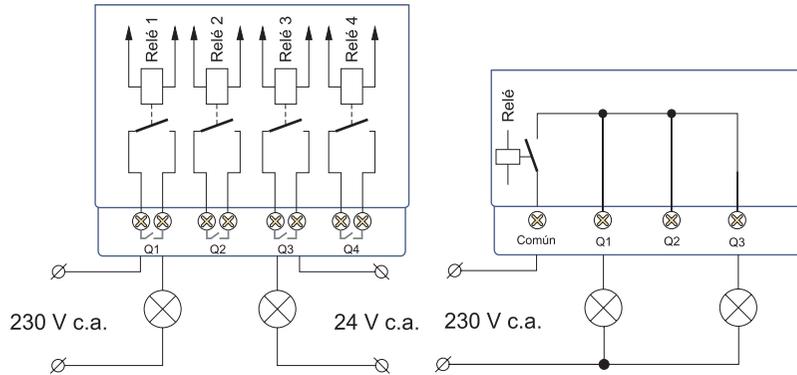
- Displays numéricos.
- Regulación de iluminación.
- Apertura o cierre en % de ciertos conductos.
- Variadores de frecuencia.
- Etc.

### 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 un relé común a tres salidas, las cuales tendrán obligatoriamente la misma alimentación.

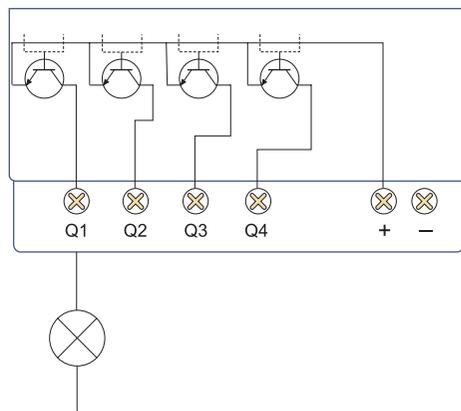


Dos modelos internos de salidas a relé.

### 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.

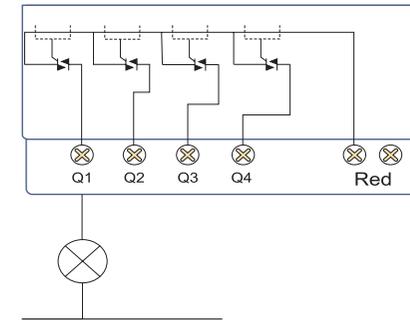
Los transistores conmutan corriente continua y son muy rápidos (algunos transistores en variadores de frecuencia conmutan más de 25.000 veces por segundo). En su contra, la corriente de paso.



Autómata con salidas a corriente continua por transistor.

### 7.3. Salidas a triac

El triac es también un dispositivo de estado sólido sin partes móviles. Al contrario que el transistor, funciona con corriente alterna y se asemeja en la rapidez de sus conmutaciones. Las altas temperaturas son perjudiciales, tanto para triac como para transistor.

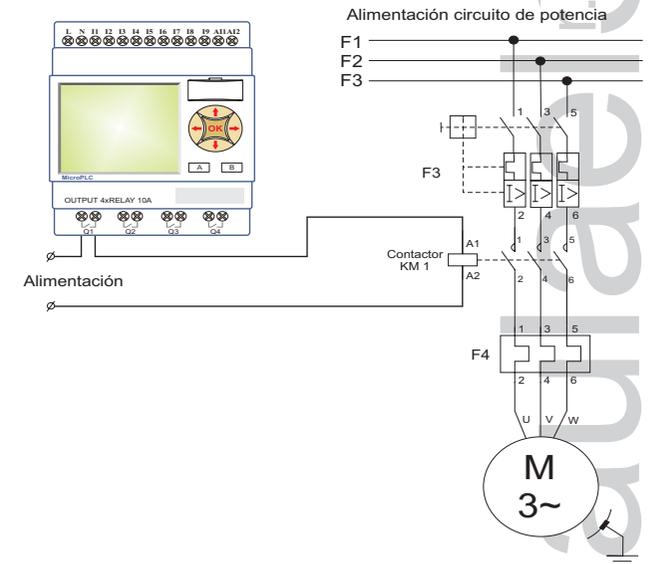


Autómata con salidas a corriente alterna por triac.

### 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.

El autómata activa un contactor, y éste a su vez excita el motor.



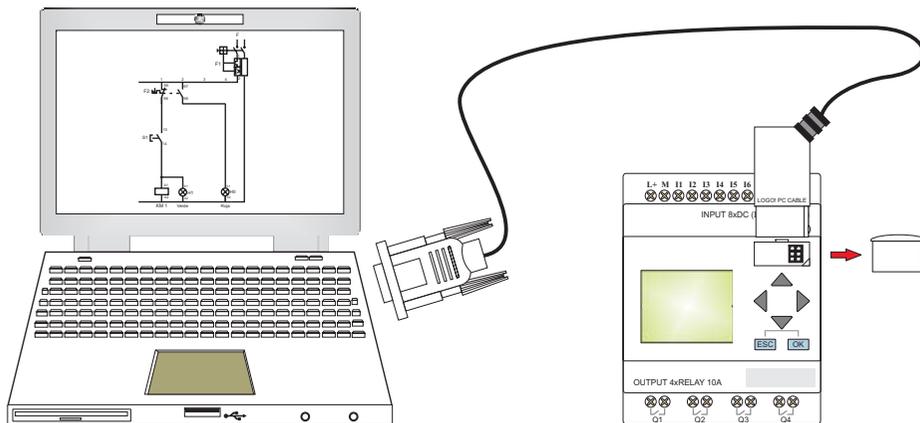
## 8 Variables de salida

Principalmente, las variables de salida usadas en micro-autómatas son de estado 1 / 0, variables booleanas. Las variables "Q" 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. 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.

## 9 Puesta en marcha de las instalaciones con autómatas programables

- Realizamos acopio de los materiales necesarios.
- Elaboramos esquemas de conexionado, según material.
- Elaboramos la programación.
- Transferimos la programación.
- Arrancamos el autómata.
- Depuramos errores.

Para elaborar la programación se requiere de una unidad de programación o un ordenador personal.



El programa se elabora en un ordenador y se transfiere a través de un cable de comunicación al PLC.

## 10 Lenguajes de programación

Cuando se tiene que programar un autómata programable para que éste realice una función automática determinada, el usuario dispone de unas herramientas que van a permitir diseñar, comprobar, modificar, la aplicación que deberá realizar el PLC. El lenguaje de programación maneja el juego de instrucciones que realizará las funciones lógicas y de cálculo de la unidad central de proceso.

Son cada vez más las marcas de autómatas programables que se acogen a una norma (IEC 1131-3) que regula los procedimientos de programación en PLCs. Esto supone que se crea un estándar común, consiguiendo que la labor de programación no sea desconocida entre modelos de diferentes firmas.

La utilización de entornos gráficos en unidades de programación, como por ejemplo un PC (dibujos orientativos, viñetas, ayudas, etc.) ha hecho de la programación un acto "menos técnico" y más fácil que los programadores antecesores.

De los cinco lenguajes que define el estándar, tres se desarrollan en forma gráfica y dos en forma textual; haremos hincapié en dos de ellos, ambos gráficos, uno por su parecido a los esquemas eléctricos y otro porque simplifica las programaciones.

Es importante destacar que los programas (software) actuales, permiten realizar programaciones en diferentes lenguajes, lo que facilita la labor del programador que puede elegir y cambiar de lenguaje a conveniencia.

Los cinco lenguajes son:

### 10.1 Lista de instrucciones (IL, Instrucción List)

Este lenguaje es adecuado para personas que no tienen gran conocimiento en esquemas eléctricos, ya que la programación se realiza de forma textual.

Realmente es un lenguaje que se aproxima a la forma de operar de la CPU, y utiliza caracteres alfanuméricos que definen las líneas de operaciones lógicas. Asimismo permite insertar comentarios informativos; esto supone que cada línea de programación puede ser "explicada" en el propio programa, y que posteriormente podrá ser impreso.

Un ejemplo de programación en lista de instrucciones (IL) sería el siguiente:

## Línea 3.

## Primera línea de programación

Modo automático: Si el detector (I5), advierte una pieza, se activa la marca auxiliar (M3), para orden de bajada, a través de la salida (Q1). Esta marca podrá ser anulada por:

- La activación del relé térmico del conjunto arriba-abajo.
- Final de carrera inferior.
- Un contacto de seguridad de motor sube (Q2).
- La memoria (M1), automático debe estar activa.

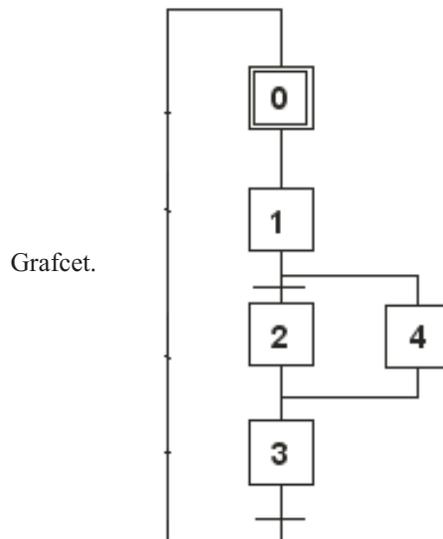
```
LD Detec_piezas
LD RT_conjunto
O FC_inferior
O Motor_sube
ON M_automático
NOT
LPS
A M_de_Q01
= M_de_Q01
```

Comentarios a la primera línea de programación

Programación

## 10.2. Gráfico secuencial de funciones (Grafcet)

El Grafcet es un lenguaje gráfico que representa las secuencias del programa en forma de diagrama. Para conocer este lenguaje, es necesario tener conocimientos en circuitos secuenciales.



## 10.3. Texto estructurado (ST, Structured Text)

Es un lenguaje de alto nivel tipo Pascal o Basic, utilizado para programaciones complejas.

## 10.4. Diagrama de contactos (LD, Ladder Diagram)

El lenguaje gráfico LD, es sin duda de los más utilizados para programar autómatas programables, ya que la simbología utilizada, es parecida a los esquemas eléctricos empleados en instalaciones eléctricas con relés o contactores. La utilización de entornos gráficos, hace que la programación se pueda “enriquecer” con textos complementarios explicativos, como sucede en el lenguaje por lista de instrucciones IL.

Note un ejemplo de programación en LD, con comentarios:

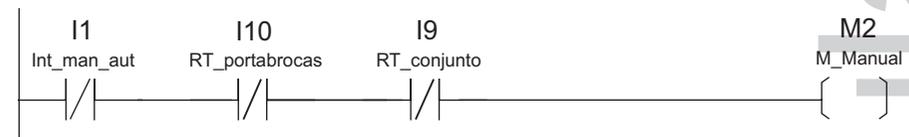
## Línea 1.

El interruptor-conmutador (I1), activará de forma directa la memoria (M1), que es indicativo de modo automático. La excitación de cualquier relé térmico del circuito, impedirá el funcionamiento.



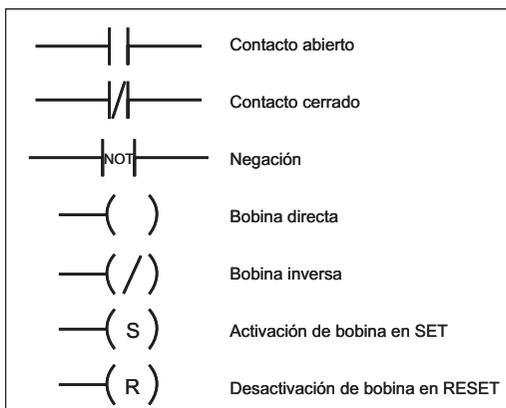
## Línea 2.

Si el interruptor-conmutador (I1), NO está activo, se activa la marca (M2), que es indicativo de modo manual. La excitación de cualquier relé térmico del circuito, impedirá el funcionamiento.

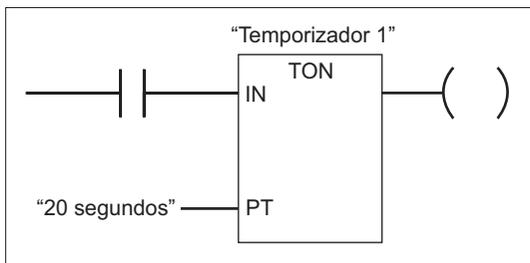


Observamos en el gráfico anterior dos formas de programar; en la primera, las variables toman su "dirección de variable" (I0.0; Q0.0; I0.1), pero también la descripción de las variables es sustituida por un "nombre" que las identifica. Esto sirve para facilitar más el proceso de programación.

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



Existen otros símbolos dentro de este lenguaje que representan operaciones como, temporización, contaje (cómputo), transferencia, suma, resta, multiplicación, reloj en tiempo real, etc; veamos un ejemplo:

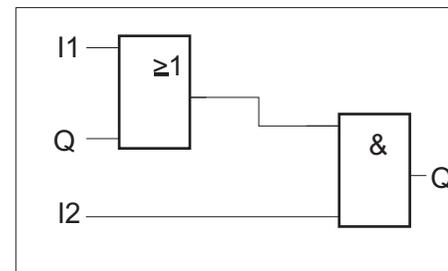


En el gráfico apreciamos como una entrada puede "activar" un temporizador. En este caso, si el bit de la entrada antecesora al temporizador es 1 y permanece en esa posición 20 segundos, la salida se activará, es decir, tendrá bit 1.

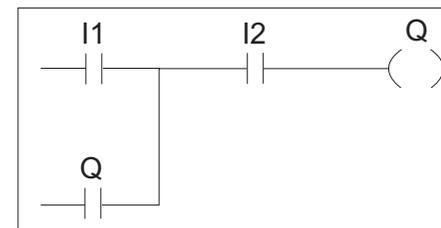
### 10.5. Lenguaje de funciones lógicas (FBD, Function Block Diagram)

El lenguaje gráfico FBD es parecido a los esquemas utilizados en electrónica digital, utilizando funciones lógicas. También se le pueden insertar textos explicativos.

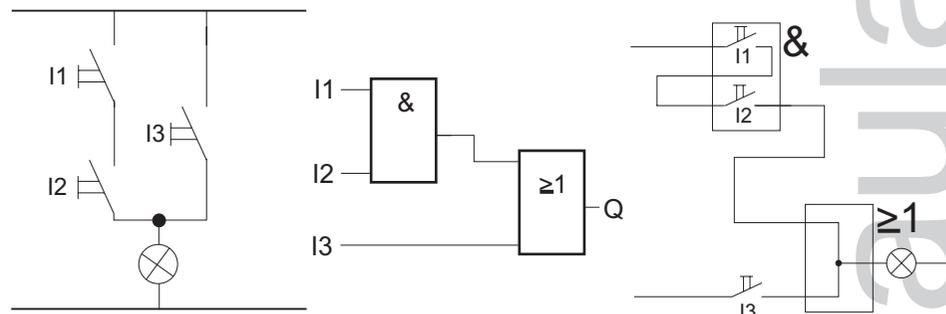
Veamos un ejemplo de una programación sencilla:



De la cual conocemos las entradas (I1 e I2) y la salida (Q), sin embargo lo que quieren decir esas "cajas" con esos símbolos en su interior ( ≥1 ; & ) lo desconocemos. Podemos decir que su programación equivalente en lenguaje LD es la que se muestra a continuación. Quiere decir que el autómata hará lo mismo tanto si se programa de una manera (FBD) como de otra (LD), son dos lenguajes diferentes, aunque los más usados:



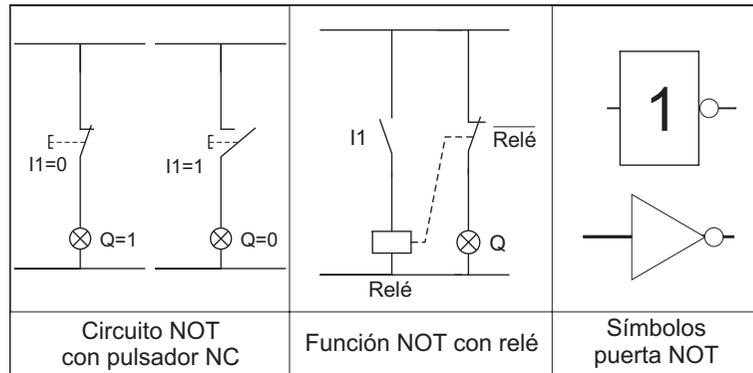
Ejemplo de equivalencia entre lenguaje FBD y contactos eléctricos



Para comprender el lenguaje de programación FBD hay que conocer las funciones lógicas básicas. Se describirán a continuación y tienen como base el álgebra de Boole.

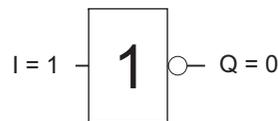
→ Función NO o inversión NOT

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



Función NOT.

Ejemplo; ¿qué valor tomará la salida Q, si la entrada tiene valor =1?



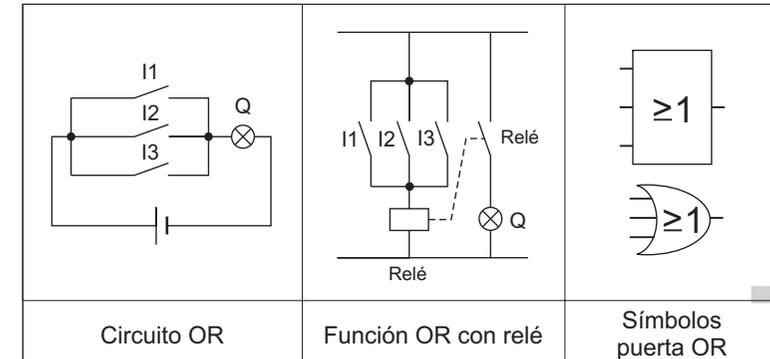
Respuesta: la salida estará desactivada = 0.

La tabla de la verdad de la función NOT es:

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

→ Función O, OR

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



Función OR.

Ejemplo, ¿qué valor tomará la salida Q, si sólo la entrada I1, tiene valor 1?

Respuesta: 1, activada.

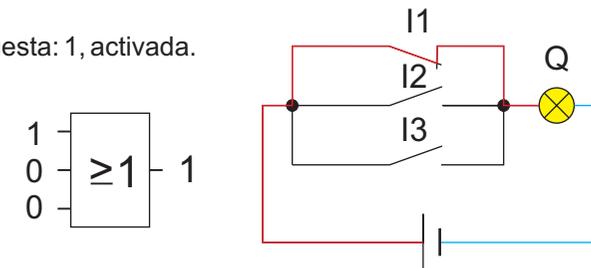
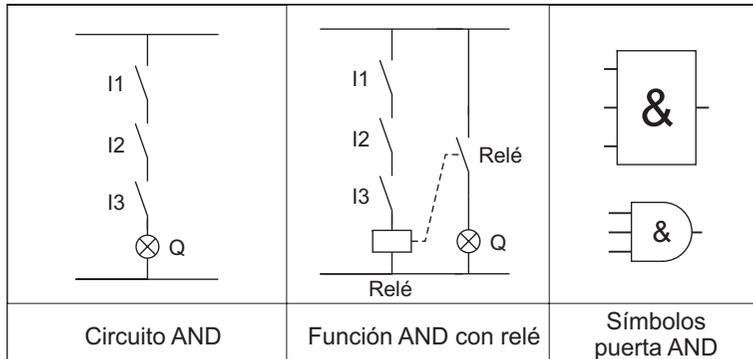


Tabla de la verdad de la función OR, con tres variables:

I1	I2	I3	Q = I1 + I2 + I3
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

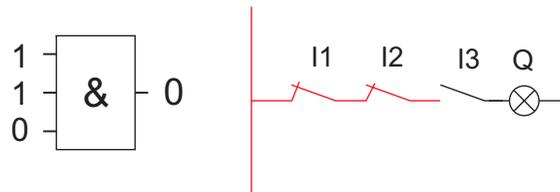
→ Función Y, AND

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



Función AND.

Ejemplo, ¿qué valor tendrá la salida Q, si sabemos que I1=1, I2=1 e I3=0?



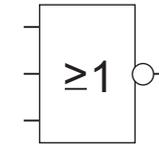
Respuesta: la salida estará desactivada.

Tabla de la verdad de la función AND, con tres variables:

I1	I2	I3	Q = I1 · I2 · I3
0	0	0	0
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	1

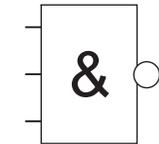
Las funciones NOR y NAND son funciones inversas a OR y AND

Símbolo y tabla de la verdad de la función NOR:



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

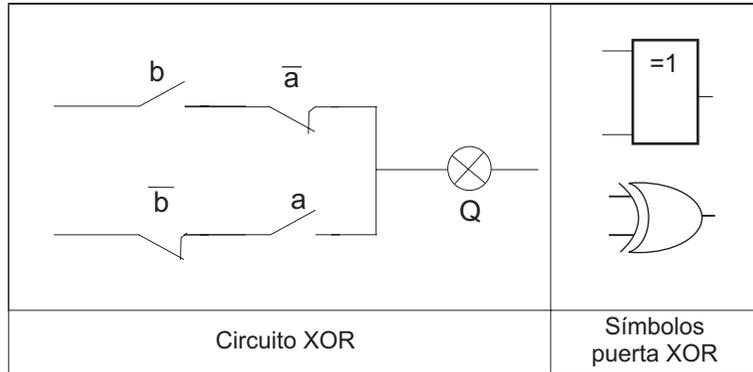
Símbolo y tabla de la verdad de la función NAND:



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

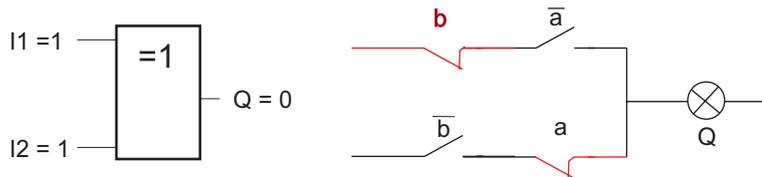
→ Función lógica XOR

La función tiene dos entradas, y la salida tomará el valor de "1" cuando una de sus entradas no tenga el valor de la otra.



Función XOR.

Ejemplo, ¿qué valor tomará la salida Q, si las dos variables de entrada tiene valor = 1?



Respuesta: desactivada = "0".

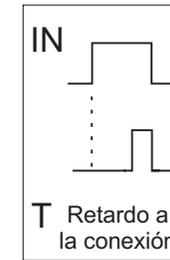
Tabla de la verdad de la función XOR:

I1	I2	Q = I1 ⊕ I2
0	0	0
0	1	1
1	0	1
1	1	0

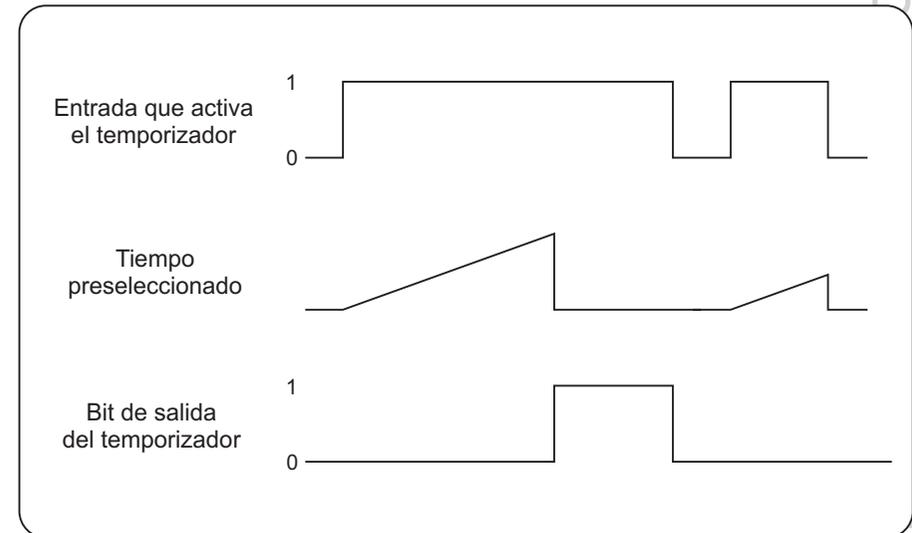
11 Funciones típicas de programación

11.1. Temporizador con retardo a la activación

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 en formato de 16 bits (una palabra -Word-).

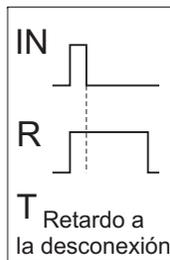


Cronograma

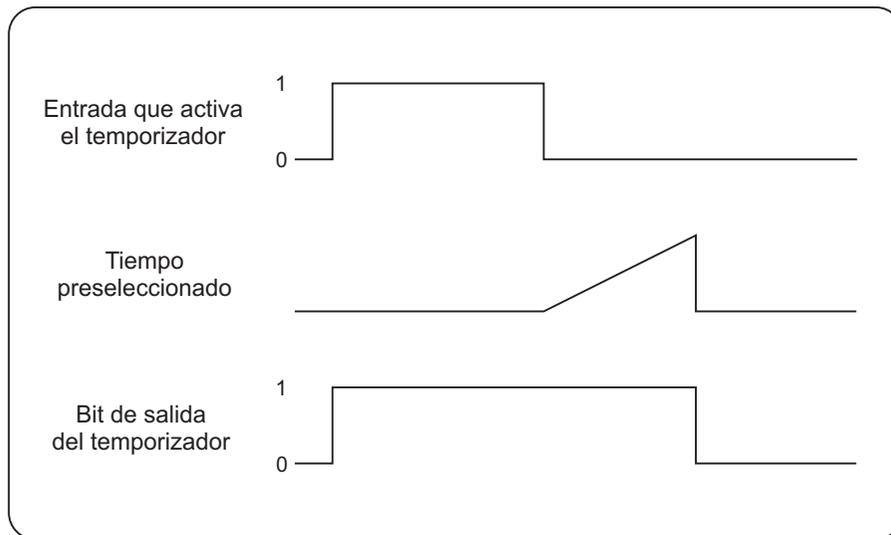


### 11.2. Temporizador con retardo a la desactivación

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 en formato de 16 bits (una palabra -Word-).



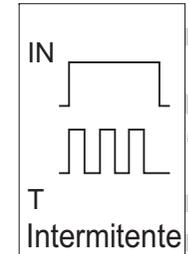
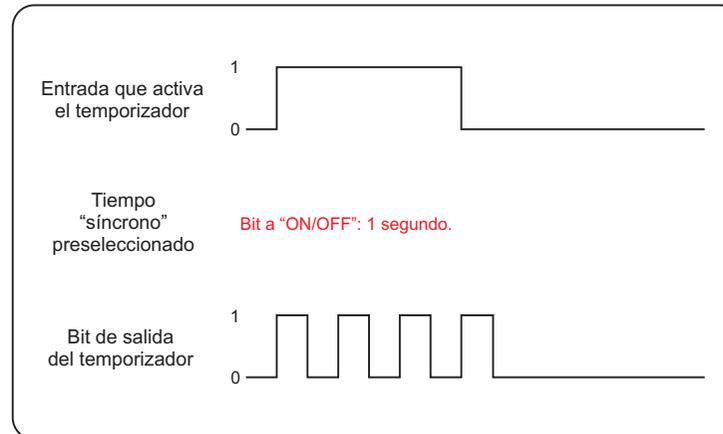
Cronograma



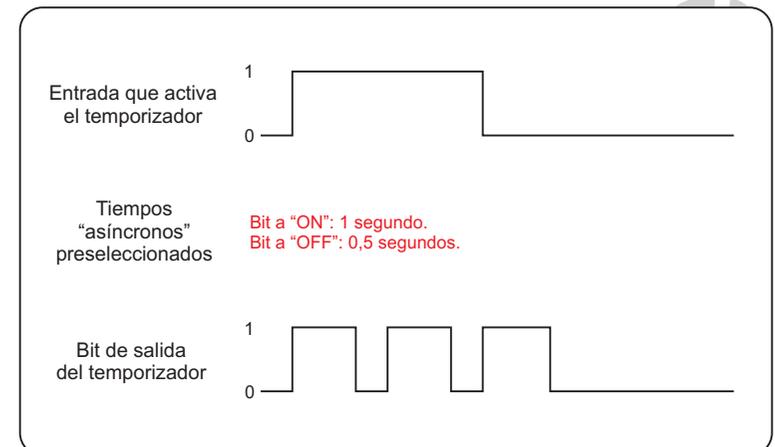
### 11.3. Temporizador con salida intermitente parametrizable

Permite un estado on/off a su salida, si la entrada (IN) está activa. Encontramos temporizadores de salida síncrona, es decir, el estado "1" y el "0" de la salida ocupan el mismo tiempo y temporizadores con salida asíncrona, donde el tiempo de activación de la salida, no tiene por que coincidir con el de descanso.

Cronograma

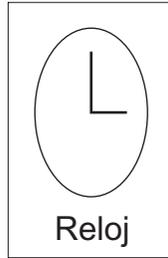


Cronograma



### 11.4. Reloj horario

Activa-desactiva su salida un espacio de tiempo parametrizable, por ejemplo, de las 08:34 hasta las 23:45 horas.

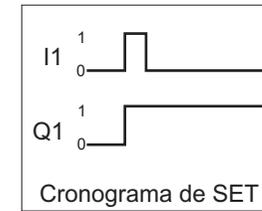
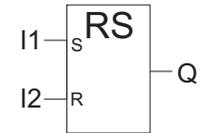


Ejemplo:

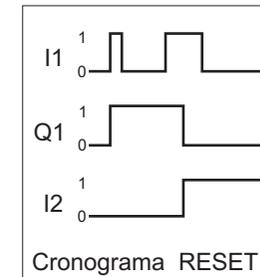
Lunes	ON: 12:45 OFF: 22:34
Martes	
Miércoles	
Jueves	
Viernes	ON: 12:00 OFF: 18:34
Sábado	
Domingo	

### 11.5. Bloque SET-RESET (RS)

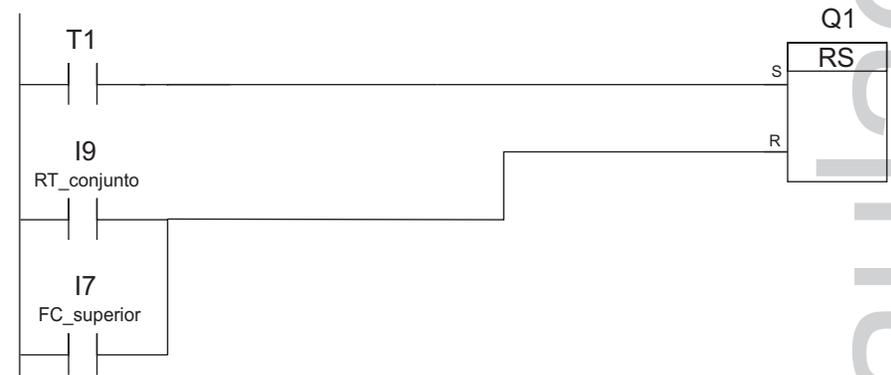
El bloque funcional set-reset, implica a su salida un "1", si en la entrada SET ha existido un impulso ("1"), es decir, el circuito se realimenta. Observe el cronograma.



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

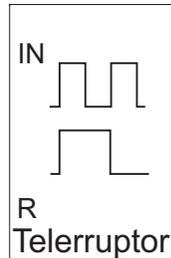


Ejemplo:



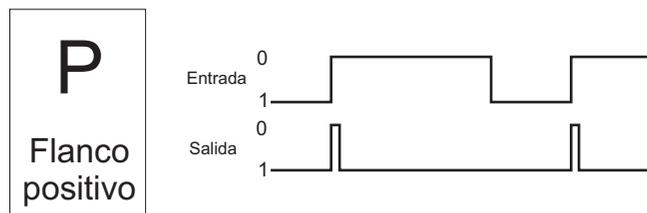
### 11.6. Telerruptor

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".



### 11.7. Flanco positivo

La función flanco positivo detecta un cambio de "0" a "1" en su entrada, implementando a su salida un "1" durante un ciclo de programa, es decir, un instante.

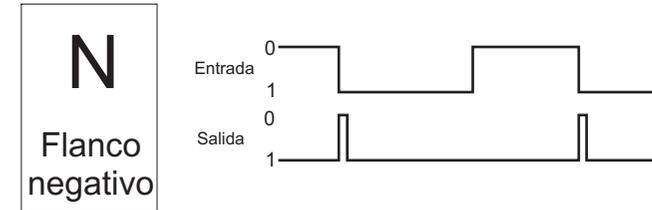


Ejemplo. La salida Q1, se activará en SET, con un sólo impulso de I3.



### 11.8. Flanco negativo

La función flanco negativo hará lo propio, si el cambio que detecta en su entrada es de "1" a "0".



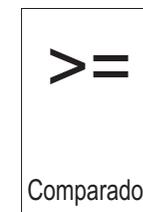
Ejemplo. La salida Q1, se desactivará en RESET cuando I4 pase de "1" a "0".



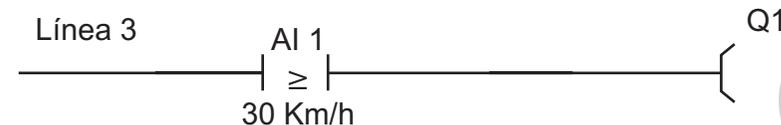
### 11.9. Comparador

Esta función compara dos valores de entrada y permite el paso de corriente a su salida si se cumple la relación entre ambos valores, que podrá ser:

- Igual que.
- Mayor que.
- Menor que.
- Mayor o igual que.
- Menor o igual que.

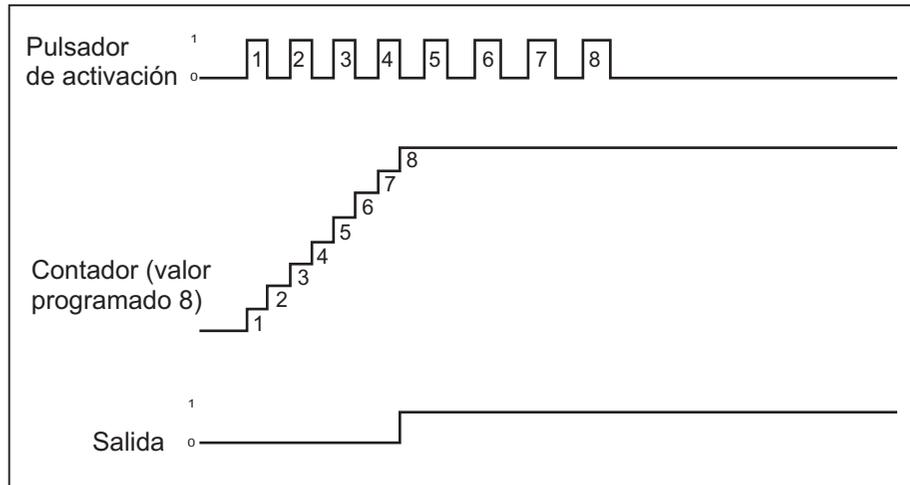


Por ejemplo: si el valor de la entrada analógica AI 1 es mayor o igual a 30 km/h, se activará la salida Q1.



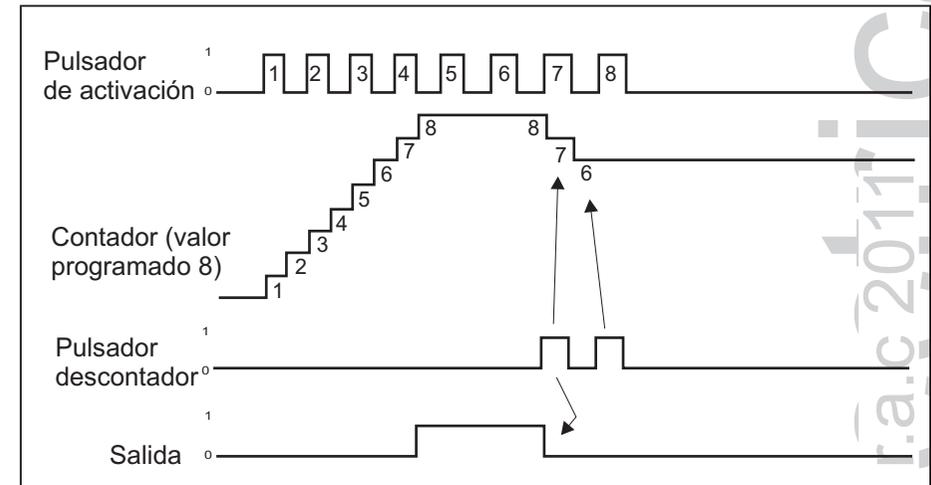
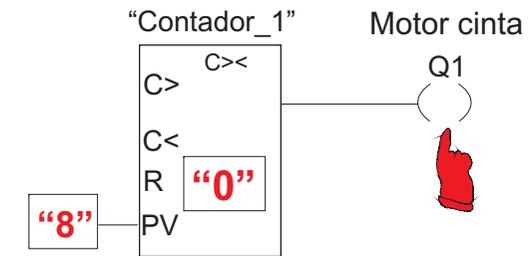
### 11.10. Contadores

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. Por ejemplo, cuando la entrada del contador se active 8 veces, se activará la salida Q. Veamos el gráfico:



- Primero, el contador está programado a 8; esto no quiere decir que la entrada no pueda ser activada más, si no que al llegar a 8, la salida se activará.

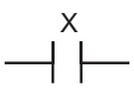
- Segundo, el contador puede disponer de “descontador”, lo cual hace que, si el número total del cómputo ascendente no es 8 la salida se desactivará. En el siguiente gráfico se muestra como el cómputo llega a 8, la salida se activa, pero el “descontador” baja el cómputo a un valor menor de 8, con lo cual la salida se desconecta.

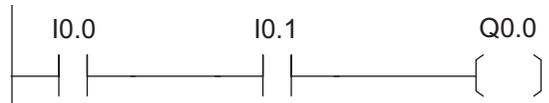
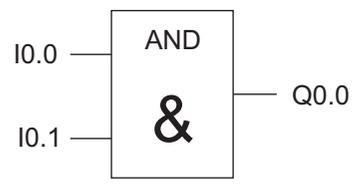
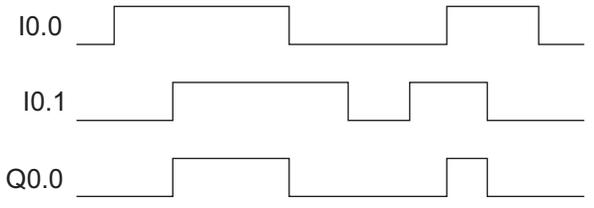


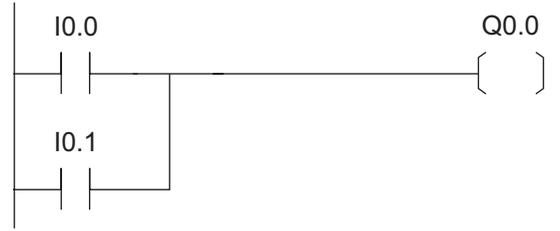
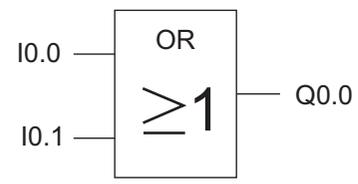
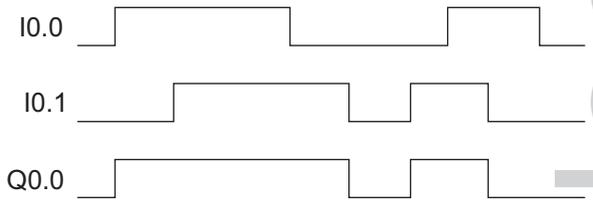
Además, los contadores disponen de una entrada de Reset (R), que hace que el número de entradas acumuladas pase a ser “0”, así como la salida.

Normalmente, de un contador usaremos dos variables; el bit digital de salida (C), que podrá ser 0/1, y el dato del cómputo, medido en formato de 16 bits (una palabra - Word-).

12 Operaciones lógicas con contactos

Lenguaje LD	Función
	El contacto abierto se cierra, es decir, permite el paso de corriente, cuando X = 1. La variable X puede ser principalmente I, Q, M, T, C, V, etc.
	El contacto cerrado inicialmente permite el paso de corriente. Se abrirá, es decir, no permitirá el paso de corriente cuando X = 1. La variable X puede ser principalmente I, Q, M, T, C, V, etc.

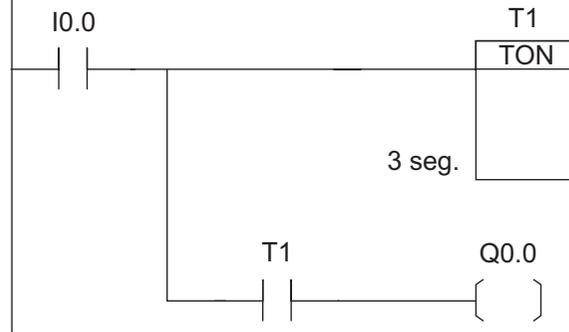
Los contactos abiertos o cerrados, se podrán asociar en serie	
En LD	
En FBD	
Cronograma	

Los contactos abiertos o cerrados, se podrán asociar en paralelo	
En LD	
En FBD	
Cronograma	

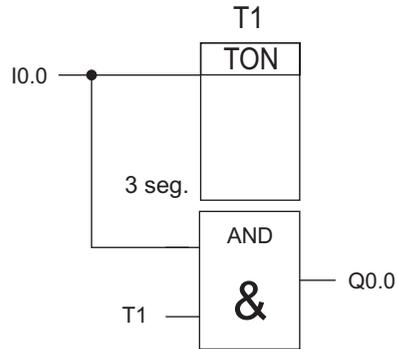
**Ejemplo**

Una vez se active la entrada I0.0, la salida Q0.0, hará lo mismo pasados tres segundos. Si la entrada I0.0 = 0, la salida también adoptará ese estado.

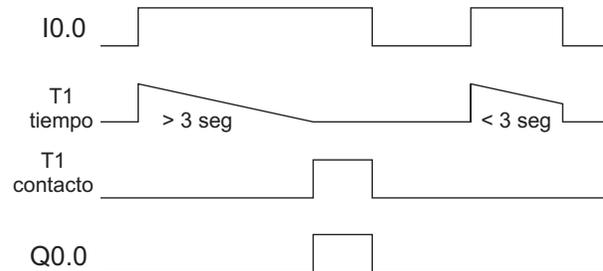
En LD



En FBD



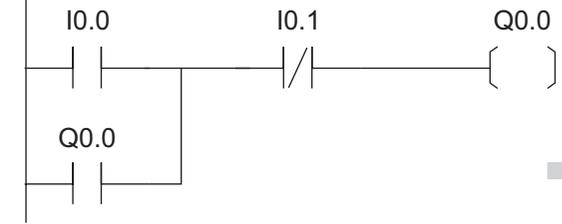
Cronograma



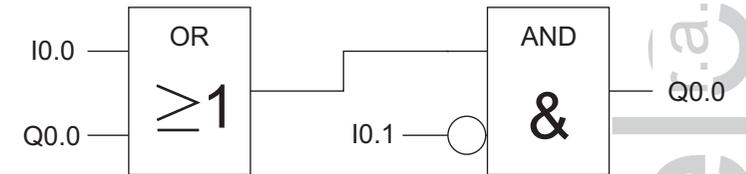
**Ejemplo**

La entrada I0.0, activará la salida Q0.0, que a su vez se realimentará a través de un contacto. Para desactivar el circuito se usará un contacto cerrado I0.1.

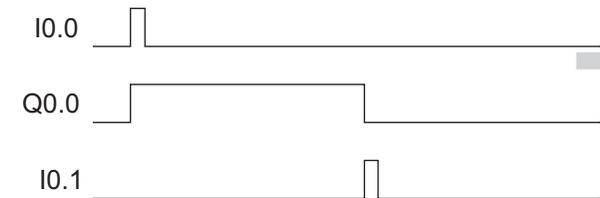
En LD



En FBD



Cronograma



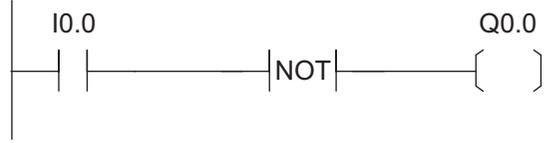
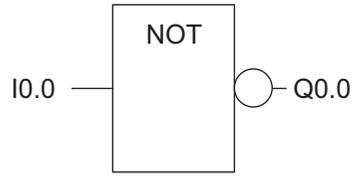
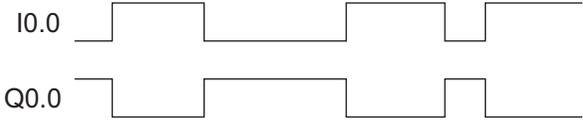
aula electrica.es

Lenguaje LD	Función
	El contacto NOT, realiza dos operaciones; si llega corriente a su entrada, interrumpe la misma a su salida; por contra, si no le llega corriente a su entrada, permite el fluido de la misma a su salida.

Lenguaje LD	Función
	El contacto Flanco Positivo, permitirá el paso de corriente durante un ciclo de programa, si su entrada cambia de 0 a 1, es decir, de no activa a activa. Es importante reconocer el hecho de que sólo permitirá el paso de corriente durante un instante.

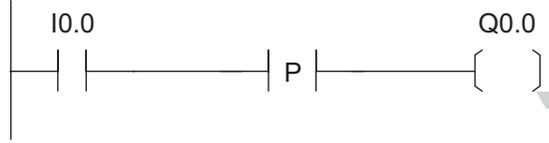
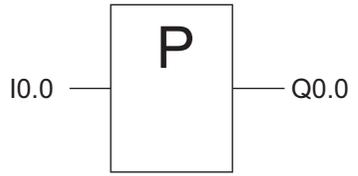
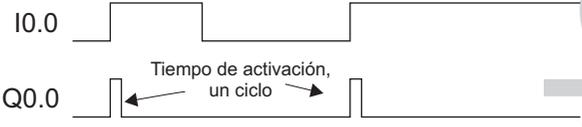
Ejemplo

La salida Q0.0 tomará el valor contrario a la entrada I0.0

En LD	
En FBD	
Cronograma	

Ejemplo

La salida Q0.0 se activará sólo durante un ciclo de programa.

En LD	
En FBD	
Cronograma	

Lenguaje LD	Función
	El contacto Flanco Negativo, permitirá el paso de corriente durante un ciclo de programa, si su entrada cambia de 1 a 0, es decir, de activa a no activa. Es importante reconocer el hecho de que sólo permitirá el paso de corriente durante un instante.

**Ejemplo**

La salida Q0.0 se activará sólo durante un ciclo de programa.

En LD	
En FBD	
Cronograma	

**Ejemplo**

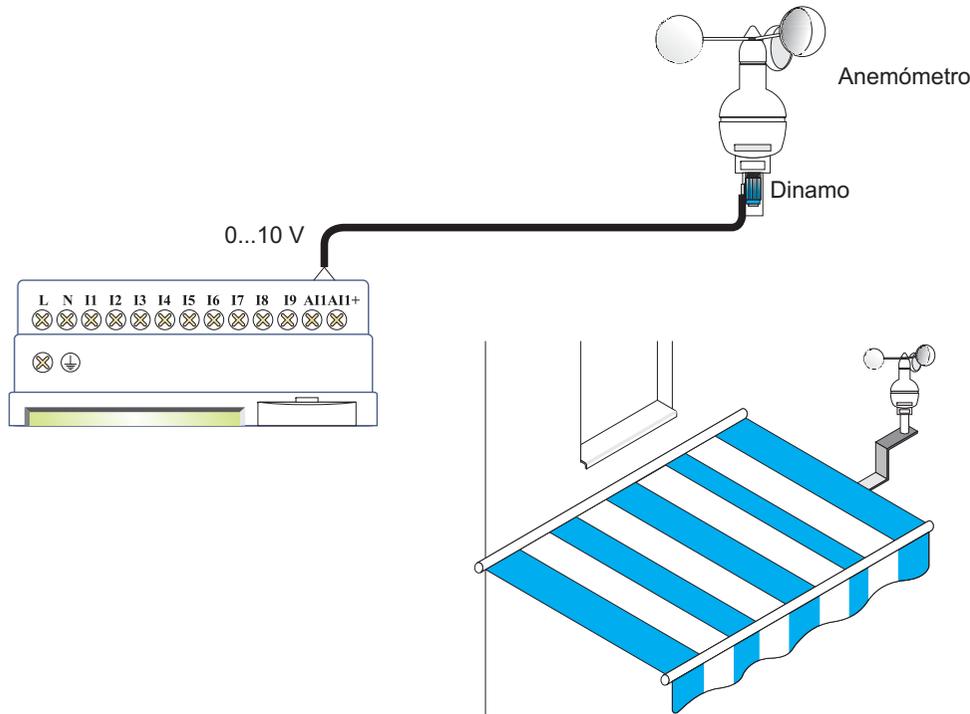
La entrada I0.0, permitirá el paso de corriente sólo un ciclo de programa, aunque se presione más tiempo, aunque este hecho no impedirá activar la salida Q0.0, que a su vez se realimentará a través de un contacto. Para desactivar el circuito se usará un contacto cerrado I0.1.

En LD	
En FBD	
Cronograma	

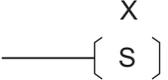
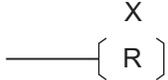
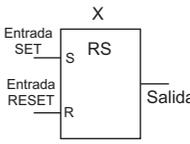
Lenguaje LD	Función
$\begin{array}{c} a \\   \\ \text{---} = X \text{---} \\   \\ b \end{array}$	El contacto COMPARAR = = permite el paso de corriente si el valor de "a" <b>es igual</b> al valor de "b". El carácter X representa el tipo de dato que se compara, que podrá ser B=byte; I=entero (16 bits); D= entero (32 bits) o R= real (32 bits).
$\begin{array}{c} a \\   \\ \text{---} > = X \text{---} \\   \\ b \end{array}$	El contacto COMPARAR > = permite el paso de corriente si el valor de "a" es <b>mayor o igual</b> al valor de "b". El carácter X representa el tipo de dato que se compara, que podrá ser B=byte; I=entero (16 bits); D= entero (32 bits) o R= real (32 bits).
$\begin{array}{c} a \\   \\ \text{---} < = X \text{---} \\   \\ b \end{array}$	El contacto COMPARAR < = permite el paso de corriente si el valor de "a" es <b>menor o igual</b> al valor de "b". El carácter X representa el tipo de dato que se compara, que podrá ser B=byte; I=entero (16 bits); D= entero (32 bits) o R= real (32 bits).

Ejemplo: control del viento en instalaciones de toldos y celosías.

Un anemómetro emite un valor analógico (de tensión o intensidad variables), que el autómata interpreta en un formato determinado, por ejemplo, valor entero (I), a través de la variable VW50. Cuando el valor de dicha variable sea mayor o igual a 100, se activará la salida Q0.0, que se corresponde con una alarma de aviso por viento fuerte, incluso la recogida automática del toldo.

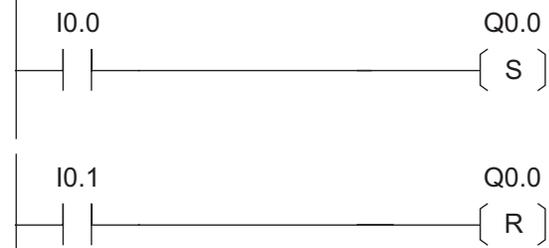
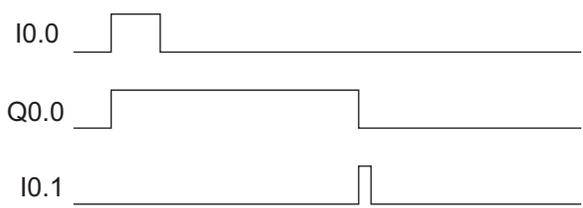


En LD	
En FBD	
Cronograma	

Lenguaje LD	Función
	La operación "Poner a 1 en SET", activa la salida "X" con un sólo impulso de corriente a su entrada, es decir, internamente se realimenta. La variable X puede ser principalmente Q, M, V, etc.
	La operación "Poner a 0 en RESET", desactiva la salida "X" con un sólo impulso de corriente a su entrada. La variable X puede ser principalmente Q, M, V, etc.
	El bloque funcional SET-RESET, permite en una sola operación activar la salida "X", si entrada SET se excita un impulso. Del mismo modo, la salida "X" se desactiva si la entrada RESET se excita. Normalmente, la entrada RESET tiene preferencia sobre la SET. La variable X puede ser principalmente Q, M, V, etc.

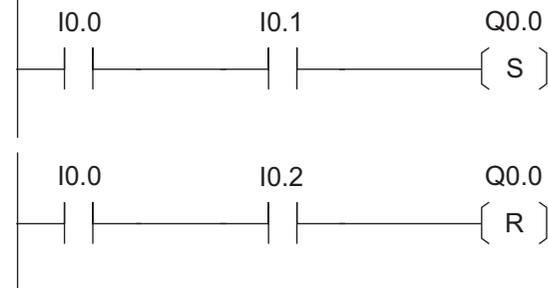
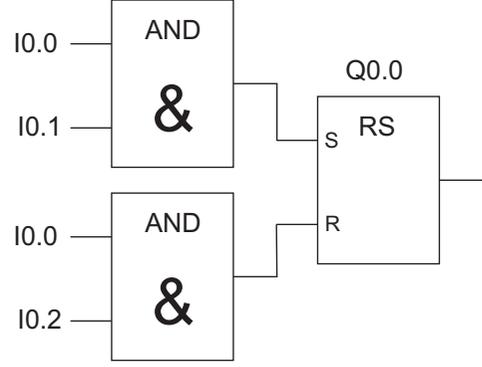
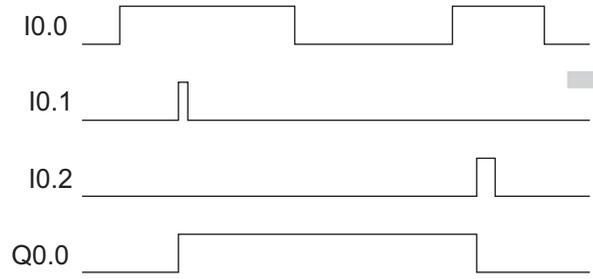
**Ejemplo**

La entrada I0.0, activará la salida Q0.0 en SET. La entrada I0.1, desactivará la salida Q0.0 en RESET.

En LD	
Cronograma	

**Ejemplo**

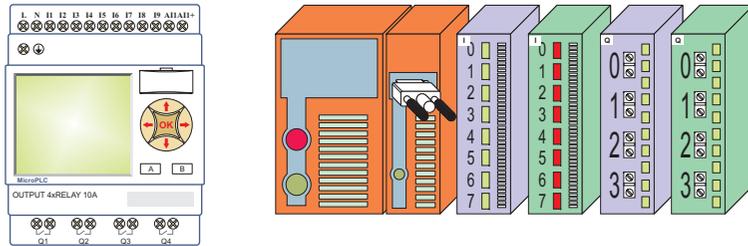
La salida Q0.0 se activará en SET, si se excitan a la vez las entradas I0.0 e I0.1. Para desactivar a Q0.0 en RESET, se deben excitar a la vez las entradas I0.0 e I0.1.

En LD	
En FBD	
Cronograma	

### 13 Complementos de los autómatas programables

En primer lugar, el autómata programable puede ser modular o compacto. Si es modular, tiene la ventaja de que su configuración puede ir creciendo según las necesidades de la instalación.

Si es compacto, tiene limitados sus recursos, principalmente en entradas y salidas digitales, aunque en la práctica, encontraremos autómatas compactos que pueden ser ampliados con módulos complementarios.

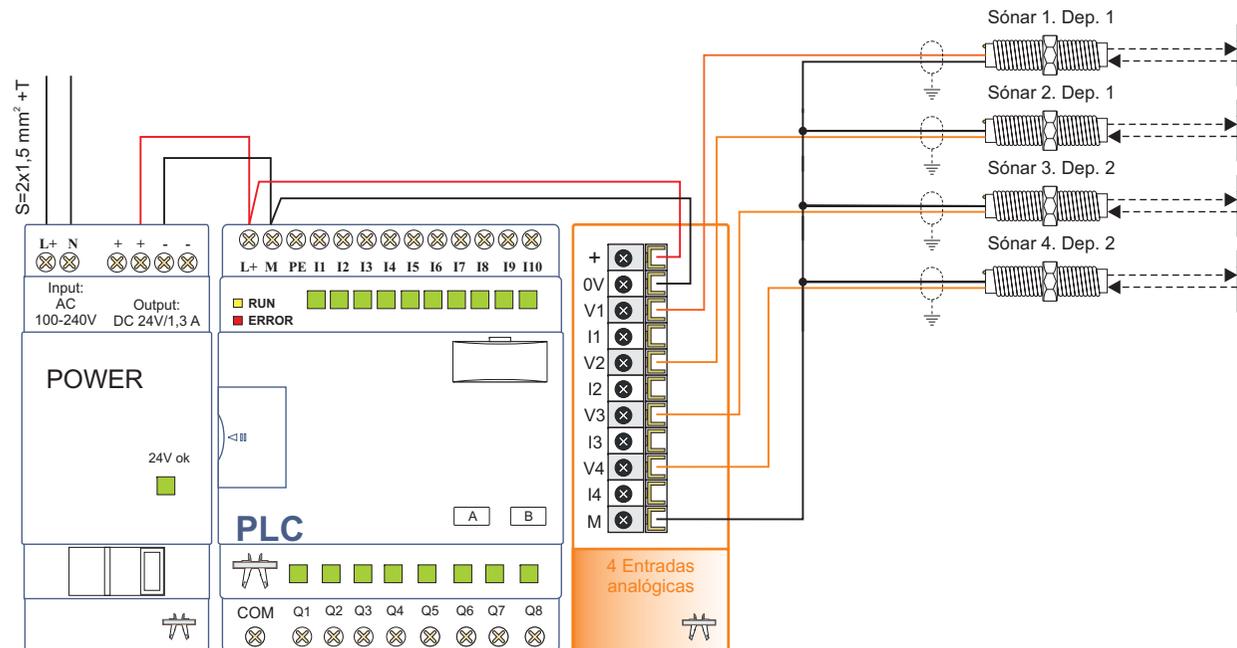
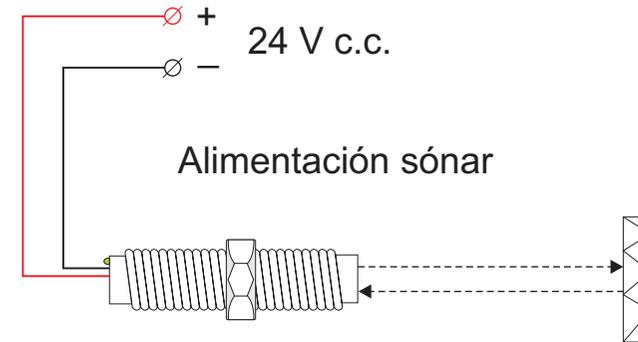


Autómatas: compacto y modular.

### 13.1 Módulo de ampliación de entradas analógicas

Es específico para recibir señales de tensión o intensidad variables, equivalentes a una magnitud (presión, velocidad, distancia, peso, etc).

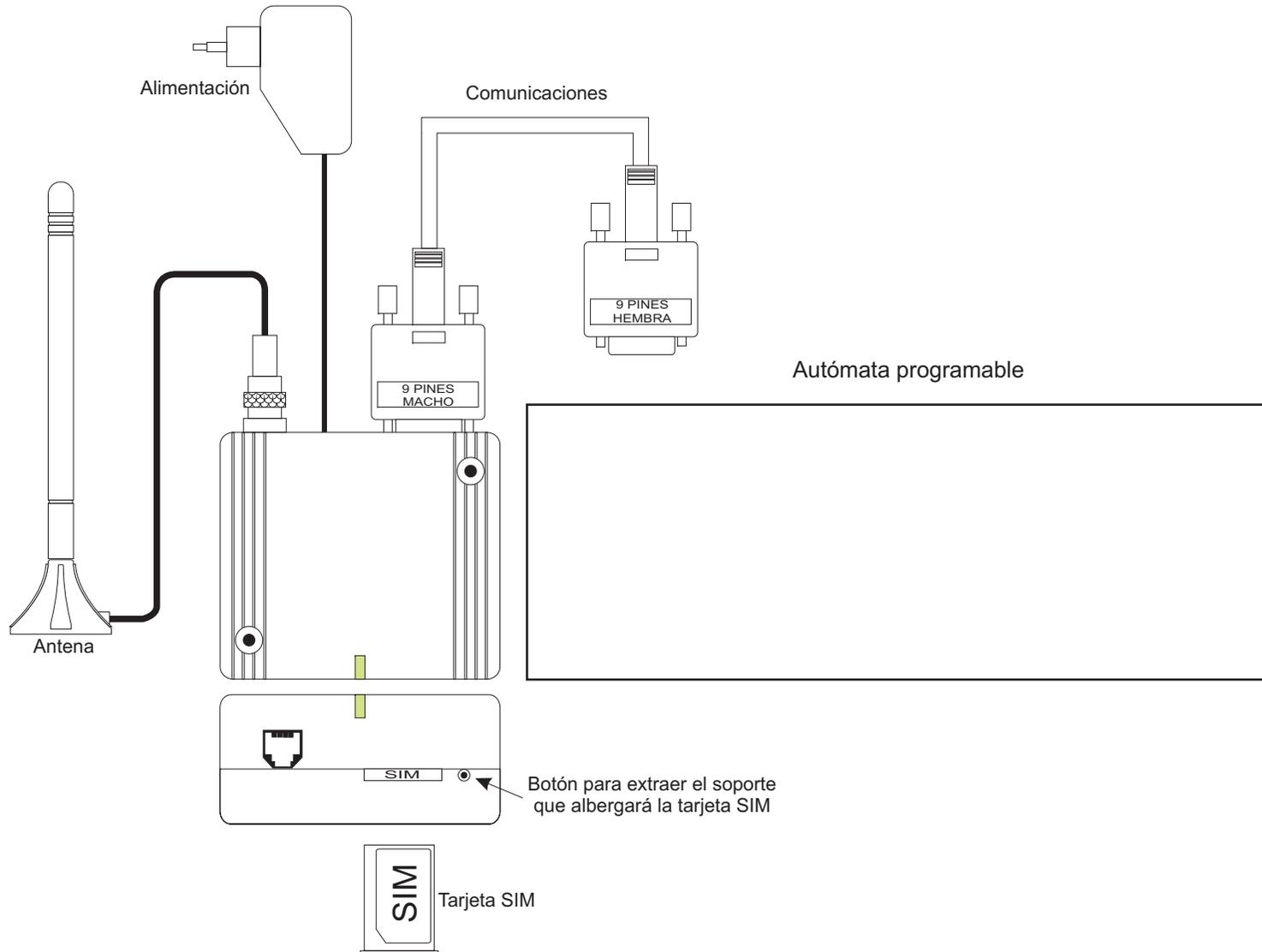
Note en el ejemplo, como 4 dispositivos sónar, están conectados a un módulo de entradas analógicas. Los sónar, además de enviar señales hacia el módulo, también han de estar alimentados. Un sónar permite medir distancia.





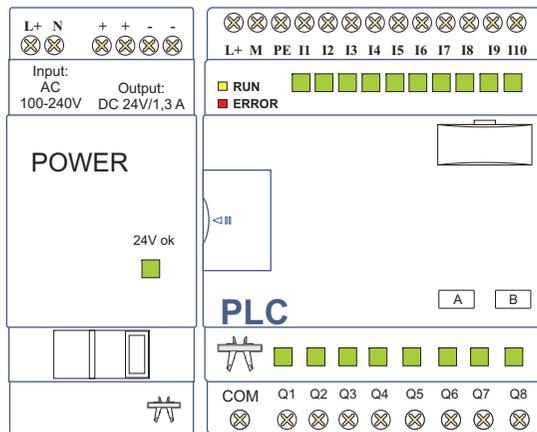
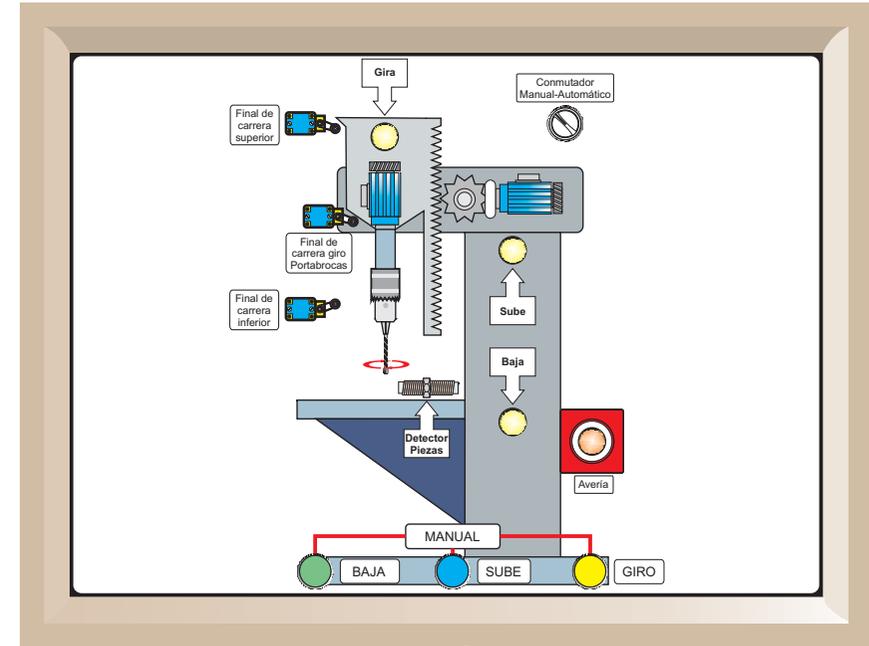
### 13.3. Módulo módem

Permite la comunicación con el PLC de forma inalámbrica.



### 13.4. Visualizadores y pantallas táctiles

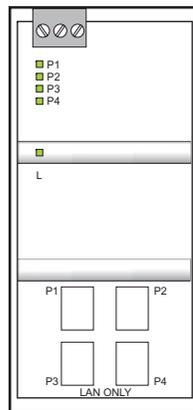
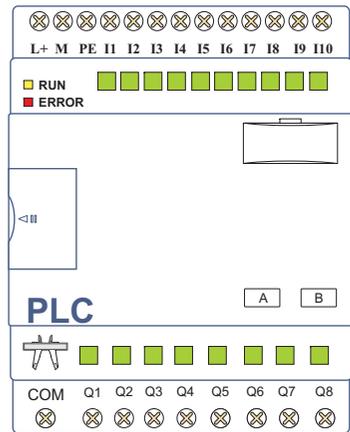
Son periféricos. Permiten el control de una aplicación de forma directa. Su entorno gráfico sugiere un control cómodo y de fácil manejo.



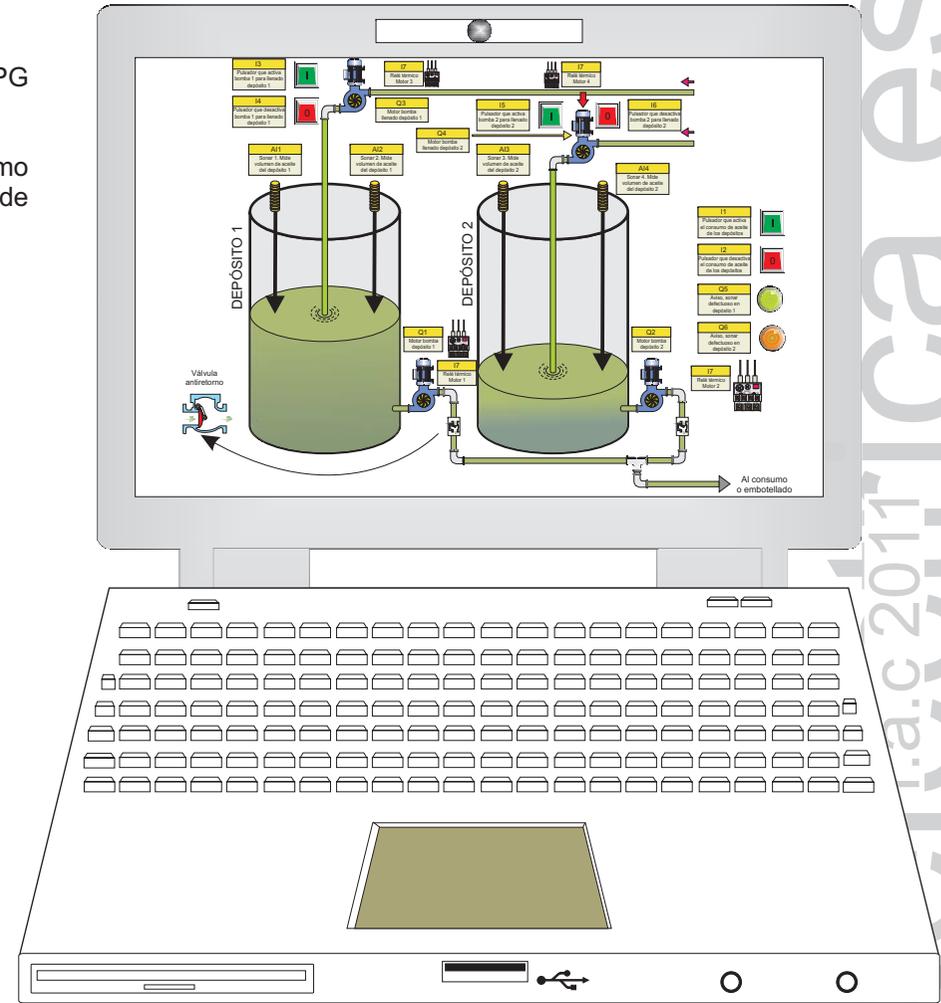
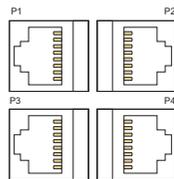
### 13.5. Sistemas Scada

Adquirir datos en tiempo real de la instalación, supervisar y controlar la misma a través de PG (unidades de programación) u ordenadores personales dedicados.

Los sistemas scada, permitirán un control integral de las aplicaciones automáticas, al mismo tiempo que muestran en diferentes pantallas el estado actual de los procesos, niveles de llenado/vaciado, tiempos de funcionamiento, etc.



Módulo de comunicaciones



aulaica.es