

# 5

## Sistemas domóticos basados en autómatas programables. Siemens Logo!

### Contenidos

- 5.1. Los autómatas programables
- 5.2. La estructura del autómata programable
- 5.3. Interfaz hombre-máquina
- 5.4. La programación de autómatas
- 5.5. El autómata programable Logo! de Siemens
- 5.6. La programación del Logo!
- 5.7. La programación del Logo! mediante diagramas de función
- 5.8. Bloques de función de conectores y constantes
- 5.9. Bloques de funciones básicas
- 5.10. Bloques de funciones especiales
- 5.11. Funciones definidas por el usuario (UDF)
- 5.12. El trabajo con señales analógicas
- 5.13. La comunicación entre varios Logo!
- 5.14. El servidor web del Logo!
- 5.15. La programación del Logo! mediante el esquema de contactos

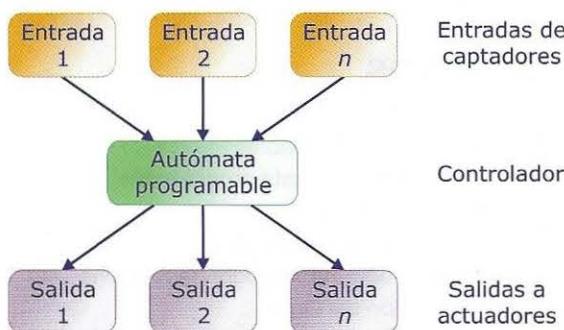
### Objetivos

- Saber qué es un autómata programable y su estructura.
- Distinguir entre los diferentes modelos y módulos.
- Realizar la instalación y conexión del autómata Logo!
- Desarrollar la programación del Logo! tanto en bloques de función como en lenguaje de contactos.
- Establecer la comunicación entre varios autómatas conectados en red.
- Utilizar y personalizar el servidor web del autómata.

**Los sistemas domóticos basados en autómatas programables aportan una gran versatilidad en el área de la automatización de las viviendas y edificios, pero, como inconveniente, su programación requiere un profesional técnico cualificado. Dentro de la amplia gama comercial de autómatas programables, los destinados al área de domótica no necesitan de una alta capacidad de procesamiento y cálculo, sino que sus necesidades están más relacionadas con el número de entradas y salidas disponibles y, teniendo en cuenta que poseen módulos de ampliación, esto los hace idóneos tanto por prestaciones como por precio. Dentro de esta gama, un autómata programable que goza de una buena aceptación es el modelo Logo! de Siemens.**

## 5.1. Los autómatas programables

Un sistema domótico se puede desarrollar empleando los autómatas programables. Un sistema con autómata programable consiste en un elemento central que hace las funciones de controlador del sistema, supervisando los sensores y detectores conectados a sus entradas y, según un programa informático desarrollado por el técnico, activar los diferentes actuadores conectados en sus salidas.



**Figura 5.1.** Esquema de un sistema domótico basado en autómatas programables.

A la hora de realizar el diseño de una instalación domótica basada en autómatas, el primer paso es la selección de dicho autómata. Su elección dependerá de la complejidad que quiera tratarse. Existe una serie de aplicaciones donde no es necesario contar con una elevada capacidad de procesamiento ni de un gran número de entradas o salidas. Por ello, los fabricantes de este tipo de material cuentan con varias gamas de autómatas programables en las que, principalmente, varía la capacidad de procesamiento, ya que el número de entradas y salidas es modular. Los autómatas programables más sencillos también suelen recibir el nombre de *relés programables*.

## 5.2. La estructura del autómata programable

Atendiendo a su estructura externa, los autómatas se clasifican en:

- **Compactos.** Todos los elementos necesarios están agrupados y se integran en un dispositivo. Son ideales para pequeñas aplicaciones con pocas entradas y salidas. No obstante, en la mayoría de ellos, se pueden ampliar y añadir otras funcionalidades.
- **Modulares.** Se componen de varios módulos, donde cada módulo cumple con una función específica, por ejemplo: entradas analógicas, entradas digitales, salidas, comunicaciones, etc. Permiten fácilmente la ampliación del sistema hasta cubrir las necesidades.



**Figura 5.2.** Autómatas programables de Logo! (Siemens) y Zelio (Schneider Electric).



**Figura 5.3.** S7-1200. Fuente: Siemens.

A nivel interno, un autómata se compone de las siguientes partes:

- **CPU.** Es la parte más importante del autómata. Se encarga de leer las entradas y activar las salidas en función de un programa.
- **Memorias.** Las hay de dos tipos: las de tipo ROM (ROM, EPROM y EEPROM), que son memorias de solo lectura y no pierden su información ante un corte de energía, y las memorias RAM, que son unas memorias de lectura/escritura y, ante un corte de energía, pierden su información.
- **Interfaz de entrada.** Adapta las señales de entrada para que las entienda la CPU.
- **Interfaz de salida.** Es la encargada de preparar las órdenes de la CPU en valores de salida.
- **Interfaz de periféricos.** Son el resto de los elementos que pueden conectarse al autómata: una consola de programación, un panel de operación, otros PLC, impresora, etcétera.

### 5.2.1. La CPU

Es la parte más importante del autómata programable y cumple varias funciones:

- **Procesar las instrucciones del programa.** Se encarga de ejecutar las instrucciones del programa de manera secuencial y de forma cíclica. La velocidad de procesamiento es muy alta.
- **Leer el estado de las entradas.**



- Activar las salidas.
- Comprobar todo el sistema. Se cuenta con una serie de alarmas que el autómata vigila.

Los diversos fabricantes agrupan sus modelos en función de la CPU.

### 5.2.2. La fuente de alimentación

Los autómatas programables funcionan internamente a tensiones reducidas en corriente continua, por ello necesitan de una fuente de alimentación. Algunos autómatas integran internamente dicha fuente, pero otros emplean un módulo externo.

Si la fuente de alimentación es interna, el autómata se conecta directamente a la red eléctrica entre fase y neutro. Por el contrario, si el autómata necesita de una fuente externa, su alimentación es a 24 Vcc.

La fuente de alimentación, además de suministrar energía al propio autómata, se emplea para alimentar a los sensores o dispositivos que lo necesiten. Algunos autómatas con fuente interna cuentan con unos terminales para esta función.



Figura 5.4. Fuente de alimentación para Logo! (Siemens) y para Zelio (Schneider Electric).

### Recuerda

La alimentación eléctrica de los autómatas programables puede ser varios tipos: los que se conectan directamente a la red eléctrica y los de tensión reducida (normalmente, 24 V en corriente continua), en este último caso, se necesita emplear una fuente de alimentación externa. Una equivocación en su conexionado puede dar lugar a que se estropee el autómata programable.

### 5.2.3. Los módulos de entradas y salidas

Los módulos de entradas y salidas para los autómatas programables se clasifican en función del tipo de dato que emplean, así se tienen:

- **Módulos digitales o binarios.** Utilizan datos a nivel de bit, es decir, 1 y 0.
- **Módulos analógicos.** Poseen cualquier valor dentro de un rango. Utilizan datos a nivel de byte (8 bits) o word (16 bits). Los módulos de entradas analógicas se emplean para leer magnitudes que no se pueden expresar en valores binarios como, por ejemplo, temperatura, presión, distancia, etcétera.

Estos módulos son de entradas o de salida, aunque también los hay mixtos, donde combinan tanto las entradas como las salidas, sin embargo, no mezclan el tipo de datos (analógicos o binarios).

En las entradas del autómata, se conectan los **captadores**, que pueden ser de dos tipos: **pasivos**, que no necesitan conexión eléctrica (por ejemplo, un pulsador o un final de carrera), o **activos**, que necesitan conexión eléctrica (por ejemplo, un sensor inductivo o una barrera fotoeléctrica).

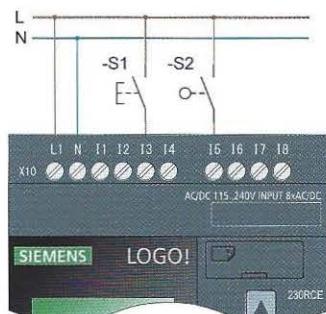


Figura 5.5. Representación de la conexión de varias entradas pasivas al autómata programable.

Para los captadores que necesitan conexión eléctrica, esta se puede obtener de una fuente de alimentación externa o del propio autómata, el cual suele contar con unos terminales destinados a este fin.

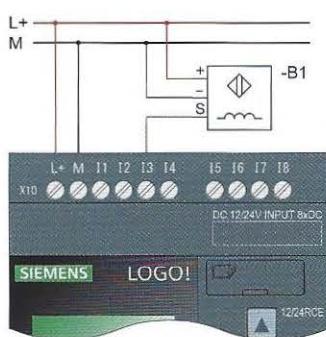


Figura 5.6. Representación de la conexión de varias entradas activas al autómata programable.

En las salidas del autómata, se conectan los **actuadores**. Existen varios tipos de salidas digitales:

- **Salidas a relé.** Tienen dos terminales y actúan como un contacto. En sus bornes, no existe tensión eléctrica, por ello se llaman *libres de potencial*. Pueden trabajar con cualquier tipo de tensión, alterna o continua, y de cualquier valor.

Las conexiones de salida pueden tener dos bornes o se pueden agrupar por bloques. Cada bloque contará con un contacto común. Tener varios bloques facilita la conexión de actuadores de diferente tensión eléctrica, a la vez que reduce las dimensiones.

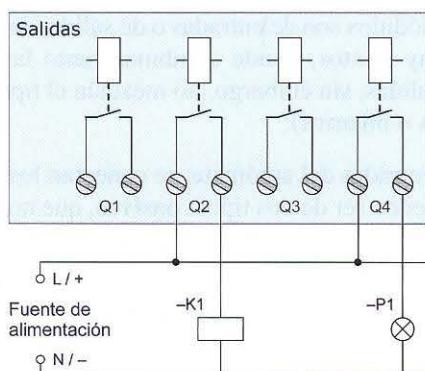


Figura 5.7. Representación de las salidas a relé con dos bornes.

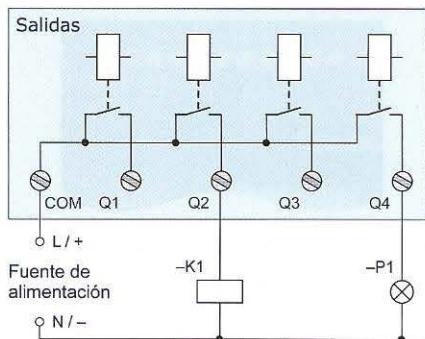


Figura 5.8. Representación de las salidas a relé con borne común.

- **Salidas a transistor.** La salida es con tensión, por ello los actuadores deben de ser del mismo valor de tensión eléctrica. Son adecuados para actuadores de corriente continua.

Las salidas a relé pueden trabajar para cualquier tipo de voltaje y permiten trabajar con mayores corrientes que con las salidas a transistor. Sin embargo, la velocidad de commutación del transistor es superior a las de relé. Por otro lado, la vida útil de los relés es inferior a la de los transistores.

A la hora de trabajar con los módulos de salidas digitales o binarias, se debe tener en cuenta cuál es la corriente máxima que puede circular por cada contacto del relé de salida del autómata programable.

### 5.3. Interfaz hombre-máquina

En la mayoría de las instalaciones domóticas, es interesante contar con algún tipo de comunicación gráfica con el usuario que se puede emplear para supervisar el estado, además de poder dar directamente órdenes sin tener que accionar cada mecanismo.

Los autómatas pueden disponer externamente de algún panel de operación, también llamado *human machine interface* (HMI).

Estos paneles o consolas se componen de dos partes: una parte es la encargada de visualizar información (tales como la situación de una máquina o proceso, alarmas, etc.) y otra parte es la encargada de recoger información que proporciona el operario (dar alguna orden concreta) mediante un conjunto de teclas.

Existen dos tipos de pantallas: las **pantallas alfanuméricas**, que proporcionan información en formato texto, y las **pantallas gráficas**, que proporcionan información en formato gráfico.

Estas pantallas se combinan con una serie de teclas, aunque, actualmente, los paneles más modernos integran ambas funciones mediante las pantallas táctiles. Estos paneles requieren una programación que se realiza mediante un programa informático.



Figura 5.9. Panel de operación HMI. Fuente: Schneider Electric.

Otra forma de realizar esta comunicación gráfica con el usuario de la instalación domótica es mediante una página web alojada en el interior del autómata. Esta opción solo está disponible en algunos autómatas programables, que son aquellos que incorporan un servidor web (*web server*), estando conectados en red. Para acceder a esta página web, se ha de utilizar el protocolo de red HTTP e ir a la dirección del autómata programable, la cual mostrará una serie de información para la supervisión, a la vez que es posible controlar el autómata. Por defecto, la página web alojada en el autómata proporciona una información que suele ser la misma que muestra físicamente, aunque es posible sustituir esta página web por otra propia.



## 5.4. La programación de autómatas

Mediante una serie de órdenes, se indica al autómata cómo procesar los datos de los captadores conectados a las entradas para activar o desactivar las salidas del autómata. Estas órdenes se expresan mediante los lenguajes de programación.

Hasta hace muy poco, los sistemas de control eran específicos de cada fabricante, con una programación y *hardware* dependiente. Esta falta de normalización significaba una escasa flexibilidad y costes elevados para el usuario. Con el objetivo de eliminar estos inconvenientes, se creó el estándar IEC 61131 y, de esta manera, unificar criterios.

### 5.4.1. Los lenguajes de programación

Existen cinco lenguajes de programación normalizados. Tres de ellos son de tipo gráfico y los otros dos son de tipo texto.

- Tipo gráfico:

- **Diagrama de bloques funcionales** (FBD, del inglés *function block diagram*). Se basa en el empleo de bloques que realizan una función concreta.
- **Diagrama de contactos** (LD, del inglés *ladder diagram*). Se basa en el empleo de la lógica de relés.

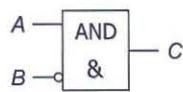


Figura 5.10. Diagrama de bloques funcionales.

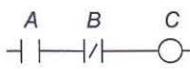


Figura 5.11. Diagrama de contactos.

- **Bloques de función secuencial** (SFC, del inglés *sequential function chart*). Se basa en la descripción de las diferentes fases por las cuales evolu-

Tabla 5.1. Tipos de datos

| Clasificación  | Tipo  | Tamaño | Valores admisibles                      |
|----------------|-------|--------|---|
| Booleano       | BOOL  | 1 bit  | 0/1 FALSE/TRUE                          |
|                | BYTE  | 1 byte | 16#00 a 16#FF                           |
|                | WORD  | 2 byte | 16#0000 a 16#FFFF                       |
| Cadena de bits | DWORD | 4 byte | 16#00000000 a 16#FFFFFFFF               |
|                | LWORD | 8 byte | 16#0000000000000000 a 16#FFFFFFFFFFFFFF |
| Entero decimal | SINT  | 1 byte | De -128 a +127                          |
|                | INT   | 2 byte | De -32768 a +32767                      |

ciona un sistema. Un ejemplo es la programación GRAFCET.

- Tipo texto:

- **Lista de instrucciones** (IL, del inglés *instruction list*). Se basa en los principios del lenguaje ensamblador.
- **Texto estructurado** (ST, del inglés *structured text*). Se basa en los lenguajes de programación de alto nivel.

LD A  
ANDN B  
ST C  
C = A AND NOT B

Figura 5.12. Lista de instrucciones.

Figura 5.13. Texto estructurado.

La elección de un lenguaje u otro va a depender, principalmente, de las preferencias y conocimientos del programador, de la dificultad del problema y del autómata que ha de emplearse, ya que no todos los entornos de programación de autómatas incorporan todos estos lenguajes, aunque la tendencia es a su incorporación. El Logo! permite la programación en lenguaje de bloques de funciones y en lenguaje de contactos.

### 5.4.2. Tipos de datos

Todos los datos y variables que maneja el autómata se guardan en memoria. No todas las variables ocupan el mismo espacio, por ejemplo, si se desea guardar un valor binario (0 o 1), solo se necesita un bit, pero, si se desea guardar un número entero o decimal, ocupará más espacio.

Los tipos de datos comunes son: BOOL, INT, REAL, BYTE y WORD. También es posible definir tipos de datos propios. El empleo de uno u otro depende de las características del valor que vaya a manejarse.

| Clasificación                    | Tipo          | Tamaño | Valores admisibles                             |
|----------------------------------|---------------|--------|--|
| Entero decimal<br>(continuación) | DINT          | 4 byte | De -2147483648 a +2147483647                   |
|                                  | LINT          | 8 byte | De -9223372036854775808 a +9223372036854775807 |
|                                  | USINT         | 1 byte | De 0 a 255                                     |
|                                  | UINT          | 2 byte | De 0 a 65535                                   |
|                                  | UDINT         | 4 byte | De 0 a 4294967295                              |
|                                  | ULINT         | 8 byte | De 0 a 18446744073709551615                    |
| Coma flotante                    | REAL          | 4 byte | 7 dígitos significativos                       |
|                                  | LREAL         | 8 byte | 15 dígitos significativos                      |
| Fecha y hora                     | DATE          | 8 byte | Las fechas dependen del fabricante             |
|                                  | TIME_OF_DAY   | 8 byte |  |
|                                  | DATE_AND_TIME | 8 byte |  |
| Cadena de caracteres             | STRING        |        |  |

La forma de manejar los datos depende del autómata. Normalmente, consiste en un identificador junto a un número, por ejemplo: I1 (entrada número 1) o Q3 (salida número 3). Otros autómatas antes del identificador le añaden un símbolo para especificar que el siguiente carácter es una variable, por ejemplo, %I 1.3 (Figura 5.14). En otros, para acceder a un bit, se debe indicar a qué byte pertenece ese bit. Por ello, es importante familiarizarse con los tipos de datos y su acceso según el autómata que se esté utilizando.

Tabla 5.2. Identificadores de datos básicos

| Identificador | Tipo         |
|---------------|--------------|
| I             | Entrada      |
| Q             | Salida       |
| M             | Marca        |
| C             | Contador     |
| T             | Temporizador |

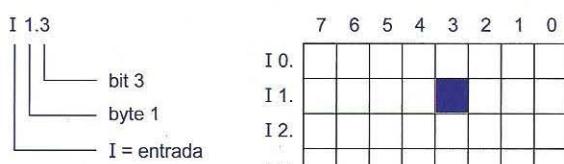


Figura 5.14. Entrada con identificación de byte y bit.

### 5.4.3. La tabla de asignación de entradas y salidas

La tabla de asignación de variables es una tabla donde se refleja qué elementos se han conectado a las entradas y salidas del autómata. También se incluyen otras variables como pueden ser las marcas, los temporizadores y otros. Al realizarla con orden y agrupándolas, permite de una manera rápida determinar e identificar cualquier elemento que intervenga en la programación del autómata.

Tabla 5.3. Ejemplo de asignación de variables

| Elemento eléctrico | Conexión autómata | Descripción            |
|--------------------|-------------------|------------------------|
| S1                 | I1                | Pulsador de marcha     |
| S2                 | I2                | Pulsador de paro       |
| Q1                 | Q1                | Contactor del motor    |
| P1                 | Q2                | Piloto de señalización |
|                    | T1                | Temporizador de marcha |
|                    | C1                | Contador de ciclos     |

La tabla de asignación de entradas y salidas se debe confeccionar antes de iniciar la tarea de programación. Permite la reducción de errores al tener claro cual es el cometido de cada variable que interviene en el programa, evitando la confusión entre ellas.



## 5.5. El autómata programable Logo! de Siemens

El Logo! es el autómata programable básico de Siemens. Inició su andadura comercial en el año 1996 y, desde entonces, está en continua evolución. Sus primeras versiones eran muy simples, pero, debido a su gran aceptación, ha incorporado nuevas funciones y capacidades, siendo muy empleado hoy en día para pequeñas instalaciones.

### 5.5.1. Tipos de Logo!

Los módulos base de Logo! están disponibles para dos clases de tensión:

- Clase 1 ≤ 24 V.** Comprende los modelos de 12 Vcc, 24 Vcc y 24 Vca.
- Clase 2 > 24 V.** Comprende los modelos de 115 Vca/cc a 240 Vca/cc.

También existen dos versiones:

- Logo! basic.** Versión con pantalla LCD.
- Logo! pure.** Versión sin pantalla LCD.



Figura 5.15. Logo! basic y pure. Fuente: Siemens AG.

Este autómata dispone de 8 entradas y 4 salidas digitales, aunque se puede ampliar hasta un total de 24 entradas y 20 salidas digitales y 8 entradas y 8 salidas analógicas.

### 5.5.2. Partes del Logo!

El autómata programable Logo! de Siemens consta de las siguientes partes:



Figura 5.16. Partes del autómata Logo!. Fuente: Siemens AG.

- Alimentación eléctrica.** Consta de dos bornes para la alimentación eléctrica. Si es un modelo de 230 V en corriente alterna, se debe proteger mediante un interruptor magnetotérmico. Si es un modelo de 24 V o de 12 V, es necesario contar con una fuente de alimentación de dicho valor.
- Entradas.** Cuenta con 8 entradas digitales, numeradas desde la I1 hasta la I8.
- Salidas.** Cuenta con 4 salidas digitales, numeradas desde la Q1 hasta la Q4.
- Interfaz RJ-45 para conexión a Ethernet.** Permite conectar el autómata a una red Ethernet. Desde este conector, se puede realizar la transferencia del programa del ordenador al autómata.
- Botonera.** Dispone de un panel de control con 4 teclas de cursor más 2 teclas de función (*ok* y *esc*). Las teclas de cursor se pueden utilizar a nivel interno mediante programa.
- Display o pantalla LCD.** Permite visualizar mensajes, así como la gestión de menús de programación. Este *display* permite retroiluminación en tres colores configurables.
- Ranura para tarjeta microSD.** Permite la inserción de una memoria de tipo microSD. Esta tarjeta permite almacenar tanto el programa como datos.
- Interfaz de ampliación.** Dispone en un lateral de un conector a través del cual es posible su ampliación mediante módulos.

### SABÍAS QUE...

El Logo! se puede reconocer por su identificador, que proporciona la siguiente información:

- **12/24:** versión de 12/24 Vcc.
- **230:** versión de 115 Vca/cc a 240 Vca/cc.
- **R:** salidas de relé (sin R, salidas de transistor).
- **C:** reloj en tiempo real integrado.
- **E:** interfaz Ethernet,
- **o:** versión sin *display* (Logo! pure).
- **DM:** módulo digital.
- **AM:** módulo analógico.
- **TDE:** visualizador de textos con interfaces Ethernet.

### 5.5.3. Conexión a la línea de alimentación eléctrica

El Logo! se puede conectar a la línea de alimentación eléctrica de dos naturalezas: líneas de corriente alterna y líneas de corriente continua.

El fabricante recomienda protegerlo de la siguiente manera:

- Protección del circuito con tensión alterna.** Para suprimir picos de tensión en las líneas de alimentación, se puede utilizar un varistor de óxido metálico (MOV). La tensión de servicio de este variador debe ser, como mínimo, un 20 % superior a la tensión nominal.
- Protección del circuito con tensión continua.** Para suprimir picos de tensión en las líneas de alimentación, se puede instalar un fusible.

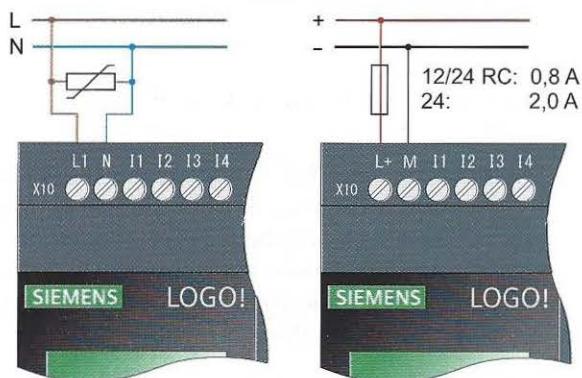


Figura 5.17. Conexión del Logo! a la línea de alimentación eléctrica según su naturaleza.

### 5.5.4. Conexión de las entradas

Las entradas de señal se conectarán a la parte de las entradas del autómata programable, de tal manera que reciban fase o positivo (en función del modelo). Esta tensión la detectará el sistema de entrada y la interpretará como un 1 o un 0 lógico.

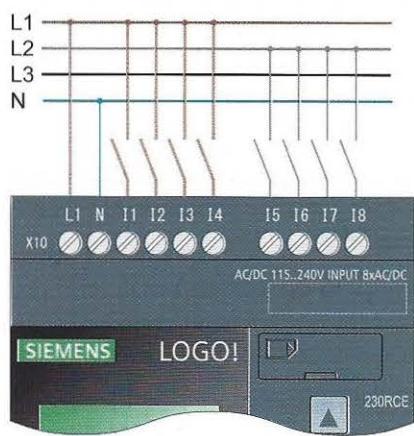


Figura 5.18. Conexión de las entradas a las líneas de fase.

En el modelo de corriente alterna, las entradas están dispuestas en dos grupos de cuatro entradas cada uno, de I1 a I4 y de I5 a I8. Solo puede haber fases diferentes entre los bloques, no dentro de cada uno de ellos.

### 5.5.5. Conexión de las salidas

Las salidas del autómata Logo! pueden ser a relé (aparece en el identificador una R), que son del tipo libre de potencial y, por ello, se puede conectar cualquier carga siempre y cuando se respete la corriente máxima de salida. Estas salidas a relé permiten una corriente máxima de 10 A. El otro tipo de Logo! tiene salida a transistor (no aparece el identificador R) y la corriente máxima se reduce a 0,3 A. En ambos casos, las salidas deben contar con algún sistema de protección como, por ejemplo, un interruptor magneto térmico o con fusible.

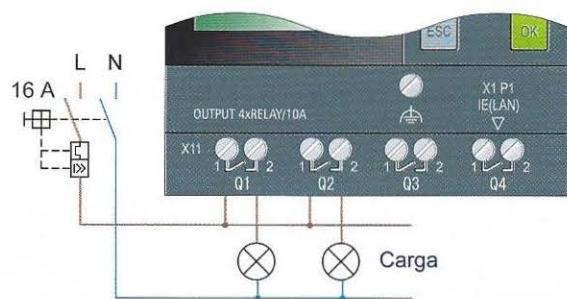


Figura 5.19. Conexión de las salidas con protección mediante magneto térmico.

### 5.5.6. Los módulos de expansión

El Logo! de Siemens dispone de una serie de módulos de expansión que mejoran sus prestaciones:

- Módulo de ampliación de entradas, tanto digitales como analógicas.
- Módulo de ampliación de salidas, tanto digitales como analógicas.
- Módulo para sensores de temperatura de tipo Pt-100 y Pt-1000.
- Display de texto externo denominado TDE. Con visualización y cuatro teclas de función. Posee conexión mediante interfaz Ethernet.
- Módulo de comunicación KNX. El estándar EIB-KNX se emplea en el área de domótica.
- Módulo de comunicación ASi. Se facilita la conexión eléctrica entre los actuadores y sensores mediante un bus de comunicación, con la consiguiente reducción del cableado.



**Figura 5.20.** Módulo de entradas analógicas.  
Fuente: Siemens AG.



**Figura 5.21.** Módulo de bus AS-i. Fuente: Siemens AG.



**Figura 5.22.** Display TDE. Fuente: Siemens AG.

## 5.6. La programación del Logo!

El autómata Logo! se puede programar mediante los lenguajes de programación de tipo gráfico, es decir, mediante el lenguaje de bloques de funciones (DBF o FUP) o mediante el lenguaje de bloques de contactos (LD o KOP).

Para realizar esta programación, se emplea el *software* Logo!Soft Comfort proporcionado por Siemens.

### Recuerda

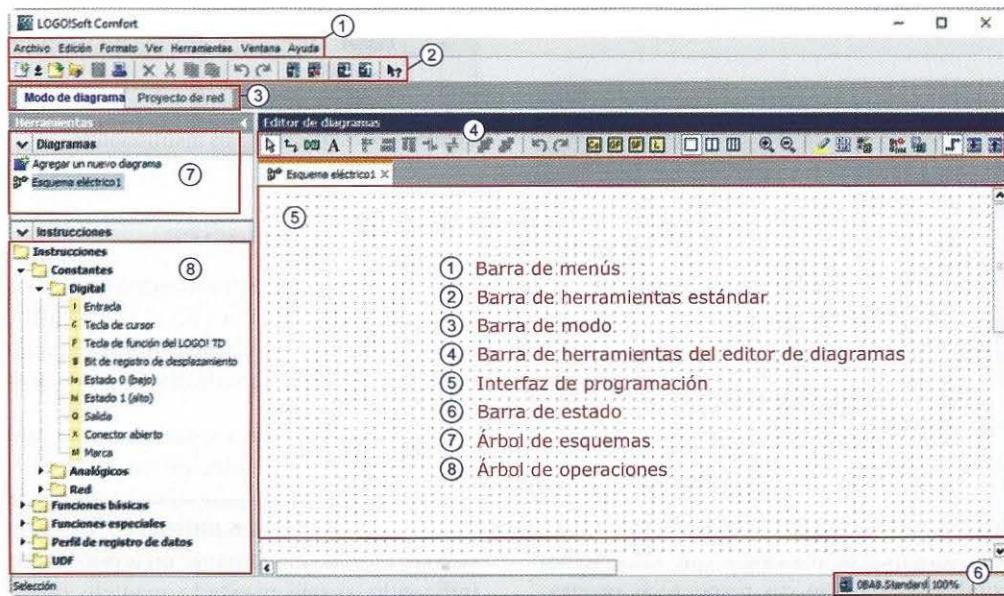
El Logo!Soft Comfort está en constante evolución y mejora y, por ello, puede ser que algunas funciones o características varíen según la versión instalada. En este libro, se emplea la versión 8.2 y el Logo! 0BA8.

### 5.6.1. Descripción general de la interfaz de usuario

Una vez instalado el programa Logo!Soft Comfort, al iniciarse en modo programa, aparece la interfaz de programación vacía (Figura 5.23), donde se pueden distinguir varias áreas.

**La barra de menús** aparece situada en la parte superior y contiene los comandos de edición y gestión, así como los de ajustes predeterminados y de transferencia del programa.

**La barra de herramientas estándar** está situada debajo de la barra de menús de la interfaz de programación y contiene los botones de uso general que permiten crear y editar el programa, así como transferirlo hacia o desde el Logo!



**Figura 5.23.** Interfaz de usuario Logo!Soft Comfort v.8.2.

**La barra de herramientas del editor de diagramas** está situada encima de la interfaz de programación y sus botones se emplean en las tareas de programación (conectar, comentar y de ayuda visual), simulación y prueba *online*.

**La barra de estado** está situada en la parte inferior derecha y proporciona información sobre el dispositivo Logo!, el estado del programa y el nivel de *zoom*.

**El árbol de esquema** contiene los esquemas que se han creado o abierto. Realizando un doble clic sobre ellos, se puede crear un esquema nuevo o mostrar el esquema seleccionado.

**El árbol de operaciones** contiene en orden jerárquico todos los elementos que permiten crear un programa. Los elementos de las instrucciones están agrupados por tipos permitiendo expandirlos y agruparlos para facilitar su uso.

**La interfaz de programación** contiene el área de trabajo para crear y editar los esquemas.

## 5.6.2. Proyecto de red

En un proyecto de red, pueden participar diversos dispositivos, como pueden ser varios autómatas, pantallas de visualización de texto TDE, paneles HMI, etcétera. Desde el proyecto de red, se pueden configurar los aspectos relacionados con la red (por ejemplo, direcciones IP), así como acceder a los esquemas de los diferentes Logo! (Figura 5.24).

En este modo, en el árbol de herramientas situado debajo de la barra de modo, cambia la vista de diagramas por proyecto de red, dando la opción de incorporar un nuevo dispositivo a la red o acceder a los ya insertados. También en este modo aparece la vista de redes, con los elementos participantes en ella, y cada dispositivo muestra su nombre y su dirección IP, siendo posible editar estos datos.

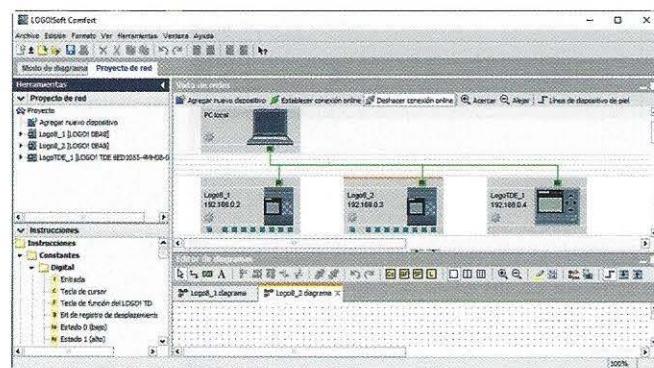


Figura 5.24. Modo de proyecto de red en Logo!Soft Comfort v.8.2.

En el editor de programas, situado debajo de la vista de redes, aparecen los esquemas de los Logo! que intervienen en la red en los cuales se realizan sus respectivas programaciones.

## 5.6.3. La creación de un programa

La confección de un programa se realiza en la interfaz de programación. Desde el árbol de operaciones, se selecciona la instrucción deseada, arrastrándola y situándola sobre la interfaz de programación. También se puede seleccionar la instrucción y hacer clic con el ratón donde se desee colocar (Figura 5.25).

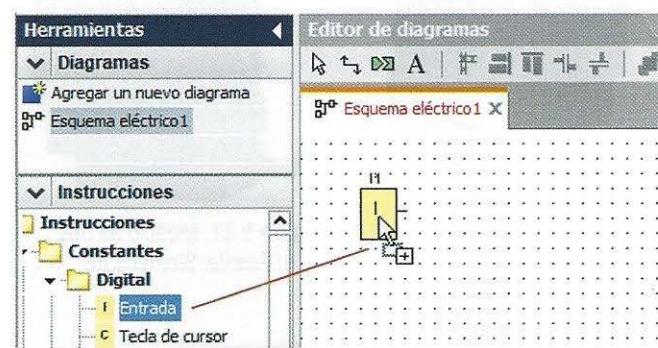


Figura 5.25. Colocación de las instrucciones para realizar el esquema.

Una vez colocados los bloques de funciones, solo queda conectarlos entre sí, para ello se selecciona el botón de conectar situado en la barra de herramientas y, haciendo clic con el ratón sobre la conexión y arrastrando hasta la otra conexión, se unen.

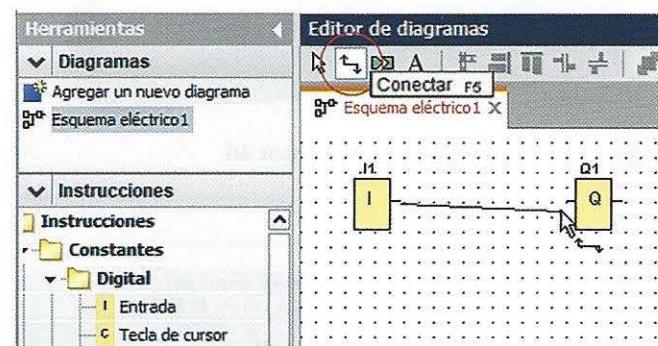


Figura 5.26. Conectando las instrucciones.

## 5.6.4. La simulación

Una vez terminada la programación, es conveniente verificar su funcionamiento. Por ello el Logo!Soft Comfort dispone de la utilidad de simulación que se encuentra en la barra de herramientas del editor de diagramas o pulsando la tecla F3.

Al entrar en modo simulación, se muestra la barra de herramientas de simulación en la parte inferior. En ella, aparecen los botones para interactuar con las entradas, mostrando solo las entradas utilizadas. A su lado, se muestran las salidas utilizadas mediante un ícono en forma de bombilla, indicando, si está en estado apagado, que la salida es de valor digital 0 y, si está encendida, de valor digital 1. Además,



las líneas de unión de los bloques cambian del color negro del modo de edición al color azul para indicar el estado de 0 lógico y rojo para indicar el estado de 1 lógico.

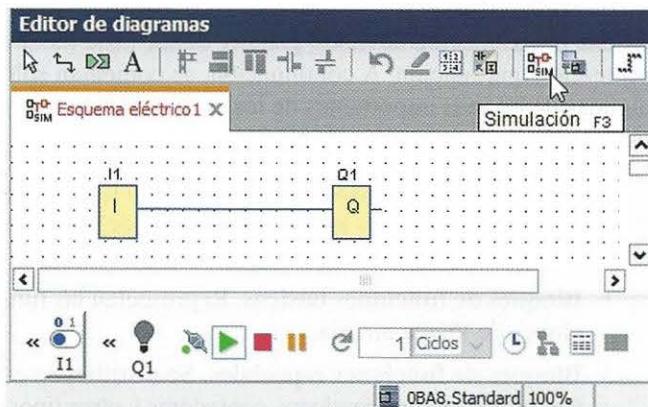


Figura 5.27. Simulación.

También se dispone de los botones de control de la simulación (iniciar, cancelar y pausar), además de botones para controlar el tiempo.

### 5.6.5. La transferencia del programa al autómata

Una vez terminado y verificado el programa, este se transfiere al autómata. Existen dos formas de transferirlo:

1. Conexión punto a punto entre el ordenador y el autómata.
2. Conexión a través de una red.

#### Conexión punto a punto

Este método de transferencia consiste en conectar directamente un cable Ethernet con conector RJ-45 desde el ordenador al Logo!.

Para que ambos equipos se puedan comunicar, se deben configurar. Si se trabaja con el sistema operativo de Windows, hay que dirigirse al panel de control, redes e internet y conexión de red (Figura 5.28).

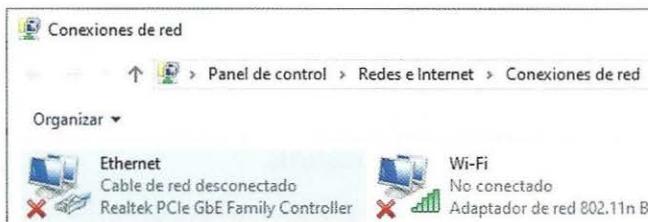


Figura 5.28. Conexiones de red.

Se realiza un doble clic sobre la red Ethernet para entrar en sus propiedades y, posteriormente, sobre el protocolo de internet versión 4. Aquí se le pone una dirección IP para

identificar al ordenador, por ejemplo 192.168.0.100, aunque puede ser cualquier otra dentro de esta red.

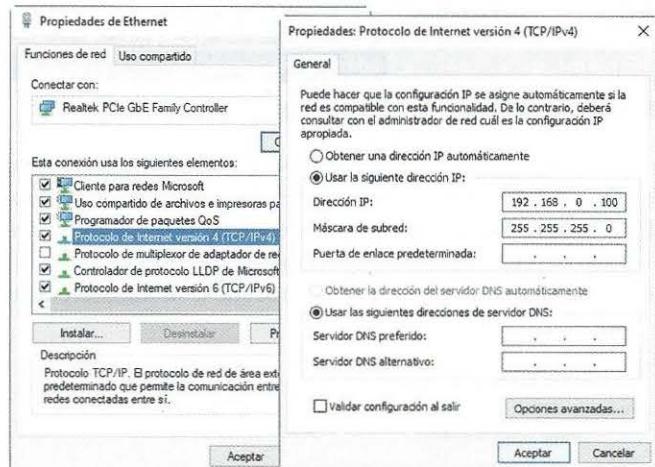


Figura 5.29. Configuración IP del ordenador en Windows.

El siguiente paso es configurar el Logo!, que se realiza desde el menú de *network*. Se coloca una dirección IP para el autómata, por ejemplo, 192.168.0.201 (Figura 5.30). Es importante que todos los equipos estén en la misma red, en este caso 192.168.0.X. El cuarto número hace referencia al equipo.

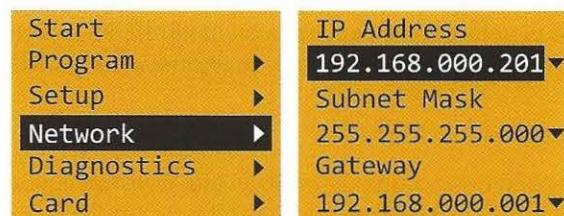


Figura 5.30. Configuración IP del Logo!

De tal manera que la conexión, a modo de ejemplo, queda como en la Figura 5.31.

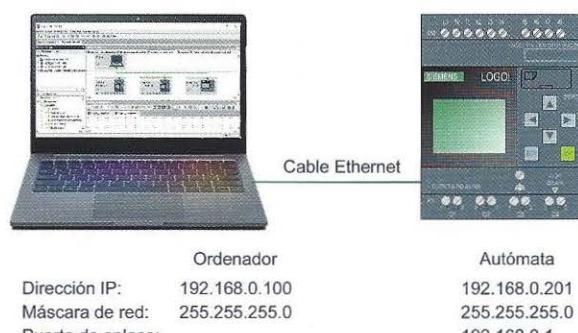


Figura 5.31. Conexión y configuración entre el Logo! y el ordenador. Direcciones IP a modo de ejemplo.

Una vez que ambos equipos están conectados y configurados, se procede a su transferencia. Esta tarea se realiza desde el menú Herramientas, Transferir, PC -> Logo!



Figura 5.32. Transferencia de PC a Logo!

A continuación, se abre la interfaz para la transferencia, donde se debe indicar cuál es la IP del autómata de destino. Este dato se puede añadir a la libreta de direcciones. Una vez indicado el destino, se puede probar la comunicación. Si todo es correcto, se acepta y se procede a su transferencia.

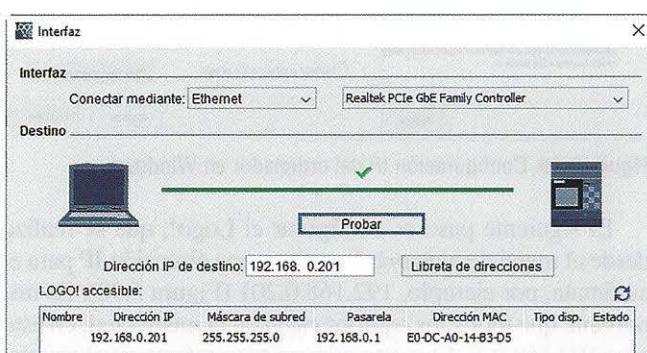


Figura 5.33. Interfaz de transferencia.

Si el autómata se encuentra en modo RUN, se detiene, se transfiere y, posteriormente, se retorna al modo RUN.

### Conexión inalámbrica

En ocasiones es cómodo realizar una transferencia de la programación mediante vía inalámbrica. En este caso, se necesita un punto de acceso inalámbrico, que puede ser el que contiene el *router*. Además, el *router* permite un acceso desde internet y, a través de sus puertos, permite la conexión de varios equipos y autómatas. El *router* se encarga de la gestión de la red y es el que fija las direcciones IP, por tanto, la dirección de la red y de la puerta de enlace. El autómata se debe configurar con estos datos. Por otro lado, el ordenador se conecta vía wifi a esta red.

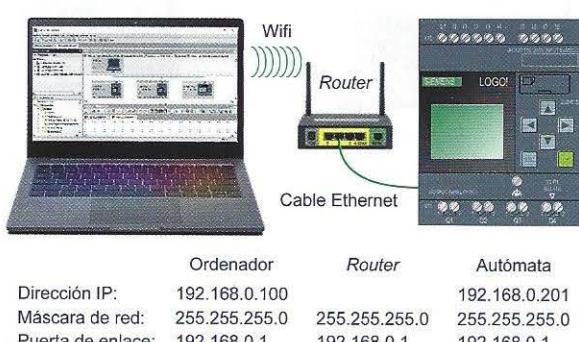


Figura 5.34. Conexión y configuración entre el Logo! y el ordenador. Direcciones IP a modo de ejemplo.

## 5.7. La programación del Logo! mediante diagramas de función

Es una programación de modo gráfico, donde cada bloque representa una función. A continuación, se comentan solo algunas de las más importantes de los cinco grupos de instrucciones que incorpora la versión 8 de Logo!Soft Comfort.

- Bloque de conectores y constantes.** Son las funciones relacionadas con la entrada y salida de información al autómata.
- Bloques de funciones básicas.** Representan las funciones de lógica binaria.
- Bloques de funciones especiales.** Se distribuyen en tres grupos: temporizadores, contadores y otros tipos.
- Bloque de registro de datos.**
- UDF.** Son funciones definidas por el usuario y consisten en un conjunto de bloques de función que se puede reutilizar en varias partes del programa. Son subprogramas o rutinas que realizan una tarea.

## 5.8. Bloques de función de conectores y constantes

Los bloques de función para conectores y constantes se dividen según el tipo de dato por manejar, teniendo para datos digitales, analógicos y de red. Representan, básicamente, a los diferentes tipos de entradas y salidas.

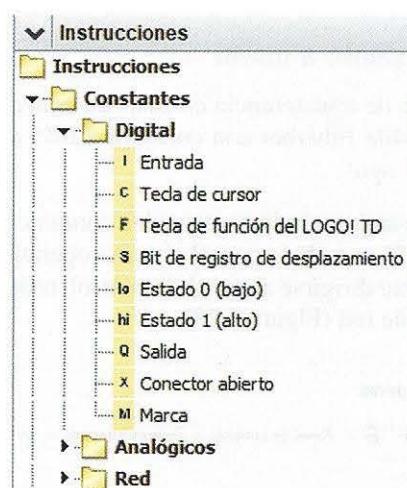


Figura 5.35 Instrucciones de constantes.

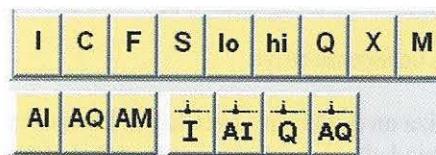


Figura 5.36. Tipos de constantes y conectores.



Tabla 5.4. Tipos de constantes y conectores

|           |   |           |   |
|-----------|---|-----------|---|
| <b>I</b>  | <b>Entradas.</b> Su número depende de la configuración del autómata.                | <b>M</b>  | <b>Marcas.</b> Representan a los relés auxiliares internos. |
| <b>C</b>  | <b>Teclas de cursor.</b> Se programan como entradas.                                | <b>AI</b> | <b>Entrada analógica.</b>                                   |
| <b>F</b>  | Son las cuatro <b>teclas de función</b> que posee el panel de operaciones externo.  | <b>AQ</b> | <b>Salida analógica.</b>                                    |
| <b>S</b>  | Son los bits de <b>registro de desplazamiento</b> . Su cantidad depende del modelo. | <b>AM</b> | <b>Marcas analógicas.</b>                                   |
| <b>hi</b> | Representa un nivel lógico de <b>1</b> .  | <b>I</b>  | <b>Entrada de red.</b>                                      |
| <b>lo</b> | Representa un nivel lógico de <b>0</b> .  | <b>AI</b> | <b>Entrada analógica de red.</b>                            |
| <b>Q</b>  | <b>Salidas.</b> Su número depende de la configuración del autómata.                 | <b>Q</b>  | <b>Salida de red.</b>                                       |
| <b>X</b>  | Son <b>conectores abiertos</b> o no utilizados.                                     | <b>AQ</b> | <b>Salida analógica de red.</b>                             |

Una **entrada** representa la conexión física de un captañor (pulsador, detector o sensor) conectado a una de las entradas de las que dispone el autómata. Para el Logo!, hay tres tipos de entradas: digitales (I), con un máximo de 24; analógicas (AI), con un máximo de 8, y entradas de red (I), con un máximo de 64.

Si esta entrada de información proviene desde una de las cuatro teclas de cursor que posee el Logo!, se emplea la función de **teclas de cursor**. Similar a esta, se encuentran las **teclas de función** que posee la pantalla TDE.

Cuando se necesita tener un valor constante de señal, se emplea la función de **nivel alto** (Hi) o de **nivel bajo** (Lo).

Las **marcas** se emplean cuando se necesita memorizar un estado y es el equivalente a los relés en los circuitos eléctricos de lógica cableada, aunque existen marcas que realizan funciones especiales como, por ejemplo, para indicar si se está ejecutando el primer ciclo del programa o para poner la retroiluminación en algún color.

Las **salidas** permiten conectar un actuador o receptor. Para el Logo!, hay tres tipos de salidas: digitales (Q), con un máximo de 20; analógicas (AQ), con un máximo de 8, y, salidas de red, con un máximo de 64.

## ACTIVIDAD RESUELTA 5.1

Se desea controlar mediante el autómata Logo! 230RCE un contactor conectado a la salida 1, el cual se activará mientras esté accionado un pulsador conectado a la entrada 4.

- Elabora la tabla de asignaciones de entradas y salidas.
- Realiza el esquema eléctrico del autómata con las conexiones a las entradas y a las salidas. Incluye la protección para el autómata y las entradas y para las salidas.
- Desarrolla el programa para el autómata en lenguaje de bloques de funciones.
- Simula el programa.

### Solución:

El primer paso es realizar la tabla de asignación de entradas y salidas.

Tabla 5.5. Tabla de asignaciones de entradas y salidas de la actividad

| Elemento eléctrico | Conexión autómata | Descripción         |
|--------------------|-------------------|---------------------|
| S1                 | I4                | Pulsador de marcha  |
| K1                 | Q1                | Contactor del motor |

Teniendo en cuenta que el autómata es de alimentación a 230 V y salida a relé, el esquema eléctrico es el siguiente:

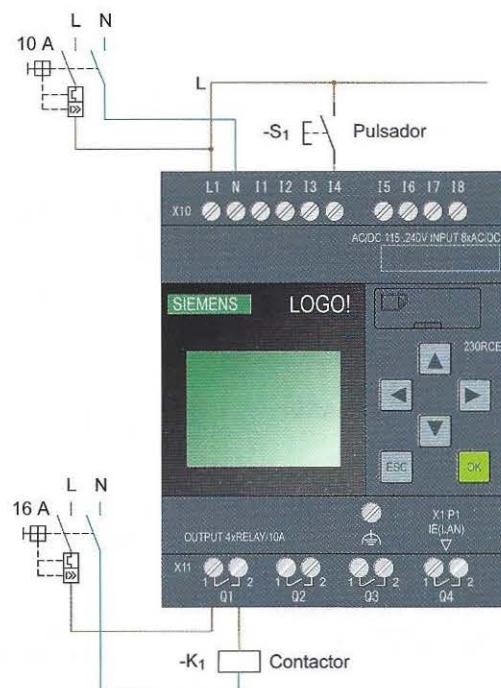


Figura 5.37. Montaje eléctrico de la actividad.

Se realiza el programa y se simula para verificar su correcto funcionamiento. Al realizar la simulación, se observa que el programa responde correctamente.

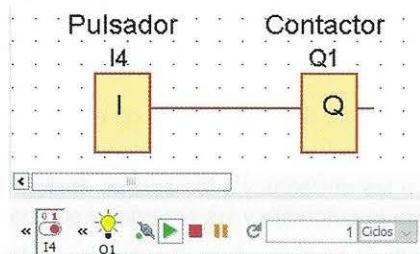


Figura 5.38. Simulación del programa.

## 5.9. Bloques de funciones básicas

Los bloques de funciones básicas son un conjunto de funciones destinadas a realizar operaciones de lógica binaria.

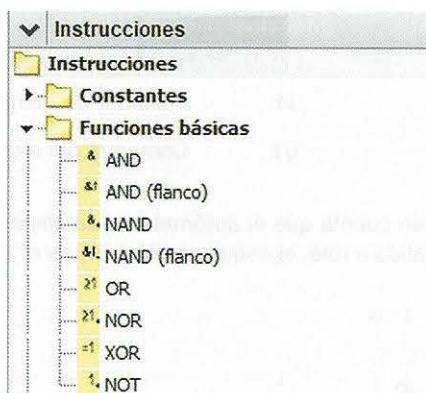


Figura 5.39. Instrucciones de funciones básicas.

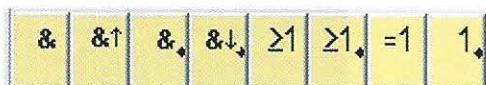


Figura 5.40. Tipos de funciones básicas.

Tabla 5.6. Tipos de funciones básicas

|  |  |  |  |
|--|--|--|--|
|  | Función lógica <b>AND</b> .<br>Por niveles.  |  | Función lógica <b>OR</b> .                     |
|  | Función lógica <b>AND</b> .<br>Por flancos.  |  | Función lógica <b>NOR</b> .                    |
|  | Función lógica <b>NAND</b> .<br>Por niveles. |  | Función lógica <b>XOR</b> .<br>(OR exclusiva). |
|  | Función lógica <b>NAND</b> .<br>Por flancos. |  | Función lógica <b>NOT</b> . Inversor.          |

### 5.9.1. La función OR

La función OR realiza la suma lógica de dos o más señales y el resultado lo coloca en la salida del bloque.

Tabla 5.7. Tabla de verdad de la función OR

| Entradas |    | Salida | Símbolo (Logo!) |
|----------|----|--------|-----------------|
| I1       | I2 | Q      |                 |
| 0        | 0  | 0      | B001            |
| 0        | 1  | 1      |                 |
| 1        | 0  | 1      |                 |
| 1        | 1  | 1      |                 |

El bloque de función que representa a la función lógica OR en el Logo! consta de cuatro entradas y una salida. De las entradas, solo se utilizarán las necesarias.

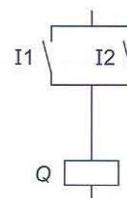


Figura 5.41. Función OR (circuito eléctrico).

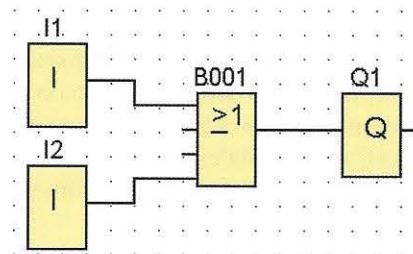


Figura 5.42. Función OR (bloques de función).

La función lógica es:  $Q = I1 + I2$ .

La función OR representa la conexión de dos o más contactos en paralelo. La salida vale 1 cuando alguna de sus entradas tenga un estado lógico de 1.

### Recuerda

La función OR, al representar un circuito paralelo, con estar cerrado (1) alguno de sus contactos, ya es suficiente para que se cierre el circuito, obteniendo un estado lógico de 1 en su salida.



## ACTIVIDAD RESUELTA 5.2

Se desea controlar la alarma de intrusión de un sistema domótico, la cual se activará si se detecta la apertura de cualquiera de las dos ventanas de una sala. Los detectores son de tipo magnético y la alarma es de tipo visual (un piloto de señalización).

- Elabora la tabla de asignaciones de entradas y salidas.
- Diseña el esquema eléctrico de las conexiones de las entradas al autómata. Ten en cuenta que son detectores a dos hilos y que el autómata es de alimentación a 230 V en corriente alterna.
- Realiza el programa para el autómata en lenguaje de bloques de funciones.

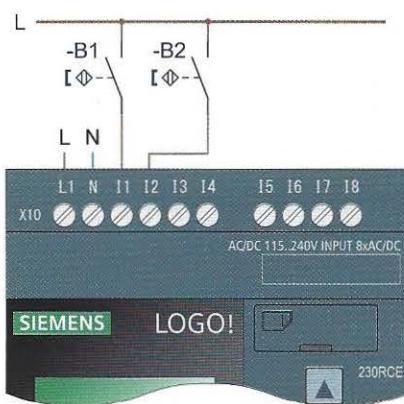
### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

**Tabla 5.8.** Tabla de asignaciones de entradas y salidas de la actividad

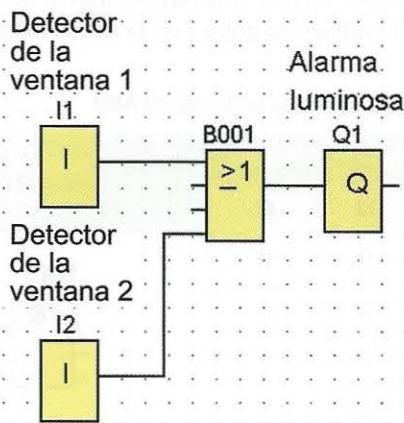
| Elemento eléctrico | Conexión autómata | Descripción                        |
|--------------------|-------------------|------------------------------------|
| B1                 | I1                | Detector magnético de la ventana 1 |
| B2                 | I2                | Detector magnético de la ventana 2 |
| P1                 | Q1                | Piloto luminoso de la alarma       |

Teniendo en cuenta la tabla de asignación de las entradas y salidas, la conexión de las entradas al autómata es la siguiente:



**Figura 5.43.** Conexión de las entradas al autómata de la actividad.

La programación es la siguiente:



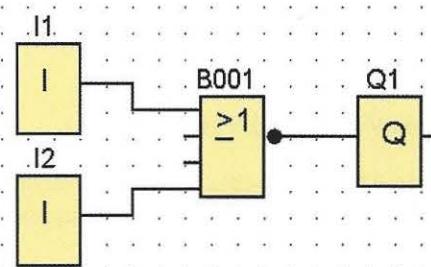
**Figura 5.44.** Programa de la actividad.

### 5.9.2. La función NOR

La función NOR es la función OR a la cual se invierte su resultado.

**Tabla 5.9.** Tabla de verdad de la función NOR

| Entradas |    | Salida | Símbolo (Logo!)   |
|----------|----|--------|-------------------|
| I1       | I2 | Q      |                   |
| 0        | 0  | 1      |                   |
| 0        | 1  | 0      |                   |
| 1        | 0  | 0      |                   |
| 1        | 1  | 0      | B001<br>--->1---● |



**Figura 5.45.** Función NOR (bloques de función).

La función lógica es:  $Q = \overline{I1 + I2}$ .

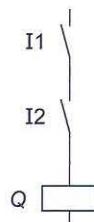
La función NOR realiza la suma lógica del valor de sus entradas y, luego, invierte el resultado, es decir, que, si la suma es un 1, en la salida coloca un 0, y viceversa. Se puede observar que la salida vale 1 cuando todas sus entradas tengan un estado lógico de 0.

### 5.9.3. La función AND

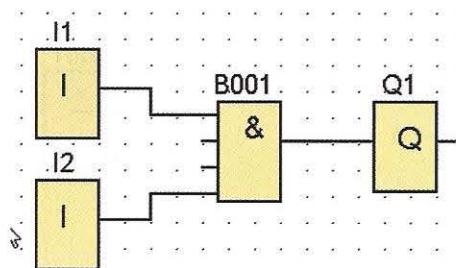
La función AND realiza el producto lógico de dos o más señales y el resultado lo coloca en la salida del bloque.

**Tabla 5.10.** Tabla de verdad de la función AND

| Entradas |    | Salida | Símbolo<br>(Logo!)  |
|----------|----|--------|---|
| I1       | I2 | Q      | B001<br> |
| 0        | 0  | 0      |   |
| 0        | 1  | 0      |   |
| 1        | 0  | 0      |   |
| 1        | 1  | 1      |   |



**Figura 5.46.** Función AND (circuito eléctrico).



**Figura 5.47.** Función AND (bloques de función).

La función lógica es:  $Q = I1 \cdot I2$ .

La función AND representa la conexión eléctrica de dos o más contactos en serie. La salida vale 1 cuando todas sus entradas tengan un estado lógico de 1.

### ACTIVIDAD RESUELTA 5.3

Se desea controlar una lámpara de iluminación exterior que se encienda cuando, siendo de noche, se detecte la presencia de alguna persona. Los detectores son con contactos libres de potencial y de conexión a la red monofásica.

- Elabora la tabla de asignaciones de entradas y salidas.
- Diseña el esquema eléctrico de las conexiones de las entradas al autómata.
- Realiza el programa para el autómata en lenguaje de bloques de funciones.

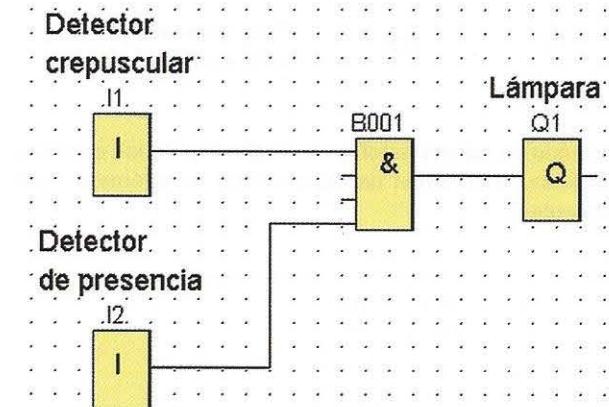
#### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

**Tabla 5.11.** Tabla de asignaciones de entradas y salidas de la actividad

| Elemento eléctrico | Conexión autómata | Descripción            |
|--------------------|-------------------|------------------------|
| B1                 | I1                | Detector crepuscular   |
| B2                 | I2                | Detector de presencia  |
| E1                 | Q1                | Lámpara de iluminación |

La programación es la siguiente:



**Figura 5.48.** Programa de la actividad.

### Recuerda



La función AND, al representar un circuito serie, todos sus contactos deben estar cerrados (1) para que circule la corriente y se tenga un 1 a su salida. Si se abre cualquier contacto, se interrumpe el paso de la corriente.

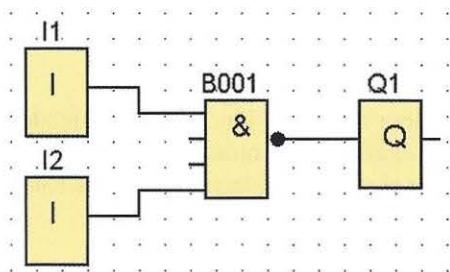


#### 5.9.4. La función NAND

La función NAND es la función AND a la cual se invierte su resultado.

**Tabla 5.12.** Tabla de verdad de la función NAND

| Entradas |    | Salida | Símbolo<br>(Logo!)  |
|----------|----|--------|---|
| I1       | I2 | Q      |   |
| 0        | 0  | 1      |   |
| 0        | 1  | 1      |   |
| 1        | 0  | 1      |   |
| 1        | 1  | 0      |  |



**Figura 5.49.** Función NAND (bloques de función).

La función lógica es:  $Q = \overline{I_1 \cdot I_2}$ .

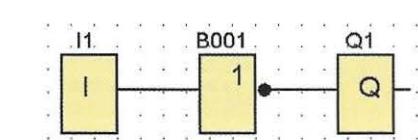
La función NAND realiza el producto lógico del valor de sus entradas y, luego, invierte el resultado, es decir, que, si el producto es un 1, en la salida, coloca un 0, y viceversa. Se puede observar que la salida vale 0 cuando todas sus entradas tengan un estado lógico de 1.

### 5.9.5. La función NOT

La función NOT se encarga de realizar la inversión del valor de su entrada.

**Tabla 5.13.** Tabla de verdad de la función NOT

| Entradas | Salida | Símbolo<br>(Logol)  |
|----------|--------|---|
| I        | Q      |  |
| 0        | 1      |   |
| 1        | 0      |   |



**Figura 5.50.** Función NOT (bloques de función).

La función lógica es:  $Q = \bar{I}$ .

La salida vale 1 cuando su entrada tenga un estado lógico de 0, y viceversa.

Se puede aplicar la función NOT tanto a resultados de bloques como a las entradas. En este caso, un contacto abierto negado es lo mismo que un contacto cerrado.

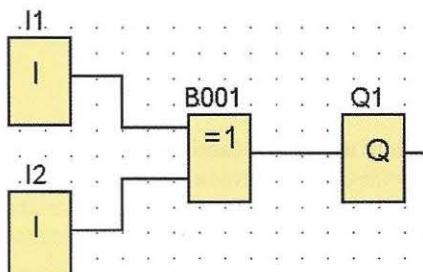
### 5.9.6. La función XOR

La función XOR, también denominada *OR exclusiva*, obtiene en su salida 1 si, en sus entradas, hay un solo 1, es decir, si sus entradas son desiguales.

La función lógica es:  $Q = I1 \oplus I2$ .

**Tabla 5.14.** Tabla de verdad de la función XOR

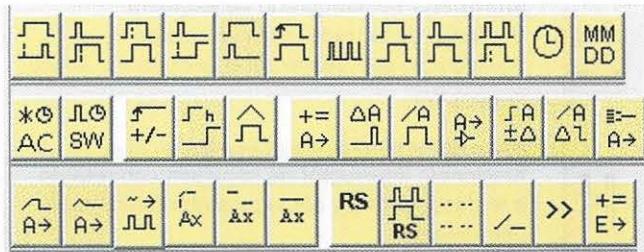
| Entradas |    | Salida | Símbolo<br>(Logo!)  |
|----------|----|--------|---|
| I1       | I2 | Q      |  |
| 0        | 0  | 0      |   |
| 0        | 1  | 1      |   |
| 1        | 0  | 1      |   |
| 1        | 1  | 0      |   |



**Figura 5.51.** Función XOR (bloques de función).

## 5.10. Bloques de funciones especiales

El autómata Logo! dispone de una serie de funciones, denominadas *especiales*, relacionadas con la gestión del tiempo (temporizadores), contadores, manejo de valores y funciones analógicas, entre otras.



**Figura 5.52.** Tipos de funciones especiales.

### 5.10.1. Temporizadores

Los temporizadores son dispositivos en los cuales interviene la variable tiempo, distinguiéndose varios en función de su comportamiento.

Tabla 5.15. Tipos de funciones especiales de temporizadores

|  |   |
|--|---|
|  | <b>Retardo a la conexión.</b> La salida se activa transcurrido un tiempo.<br>Entrada <b>Trg</b> ( <i>trigger</i> ). Dispara el temporizador.  |
|  | <b>Retardo a la desconexión.</b> La salida se desactiva tras haber transcurrido el tiempo configurado.<br>Entrada <b>Trg</b> . Dispara el temporizador.<br>Entrada <b>R</b> . Una señal en la entrada reinicia el temporizador.         |
|  | <b>Retardo a la conexión/desconexión.</b> La salida se activa transcurrido un tiempo y se desactiva.  |
|  | <b>Retardo a la conexión con memoria.</b> Se activa con una señal de impulso.<br>Entrada <b>Trg</b> y entrada <b>R</b> .  |
|  | <b>Relé de barrido (salida de impulso).</b> Una señal de entrada genera una señal de duración configurable.<br>Entrada <b>Trg</b> .   |
|  | <b>Relé de barrido (activado por flanco).</b> Un impulso de entrada genera un número predeterminado de impulsos de salida con una relación impulso/pausa definida.<br>Entrada <b>Trg</b> y <b>R</b> y se configura el número de ciclos. |
|  | <b>Generador de impulsos asíncrono.</b> Genera una onda de impulsos configurable.<br>Entrada <b>En</b> . Activa y desactiva el generador de impulsos.<br>Entrada <b>INV</b> . Invierte la onda de salida.                               |
|  | <b>Generador aleatorio.</b> La salida del generador se activa y desactiva dentro de un tiempo configurable.   |
|  | <b>Interruptor de alumbrado de escalera.</b> Un flanco de entrada inicia un tiempo configurable que transcurrido desactiva la salida.<br>Entrada <b>Trg</b> .   |
|  | <b>Interruptor bifuncional.</b> Cuenta con dos funciones: interruptor de impulsos con retardo a la desconexión y como pulsador (alumbrado permanente).<br>Entrada <b>Trg</b> y <b>R</b> .   |

|  |  |
|--|--|
|  | <b>Temporizador semanal.</b> La salida se controla en función de un día de la semana.<br>Dispone de tres programaciones.     |
|  | <b>Temporizador anual.</b> La salida se controla en función de la fecha.   |
|  | <b>Reloj astronómico.</b> La salida se controla en función de la salida y puesta del Sol según unas coordenadas geográficas. |
|  | <b>Cronómetro.</b>   |

#### Temporizador a la conexión

El temporizador a la conexión, también llamado *retardo a la conexión*, al activarse (entrada Trg), comienza la cuenta (*t*) y, al llegar al valor establecido, activa su salida (Q).

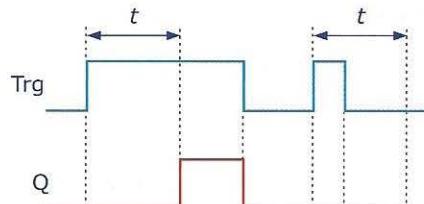


Figura 5.53. Cronograma de un temporizador con retardo a la conexión.

Este bloque de función dispone de una entrada (Trg) y una salida (Q) y, si se realizar doble clic sobre él, se entra a realizar su configuración.

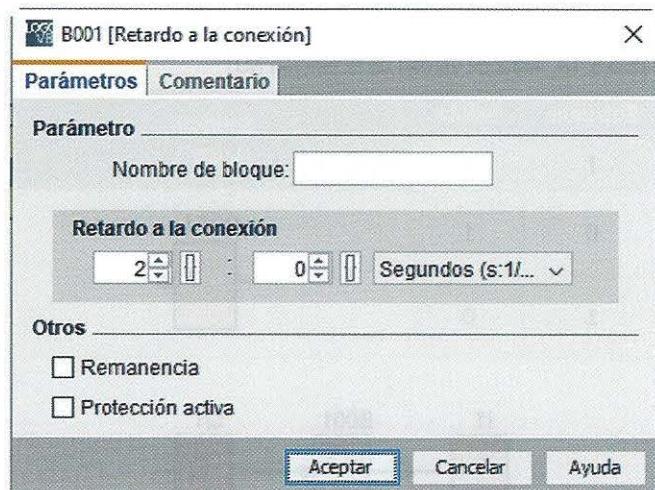


Figura 5.54. Configuración de las propiedades del temporizador con retardo a la conexión.



La configuración (Figura 5.54) consiste en indicar el tiempo de actuación, que se puede indicar en segundos, minutos u horas. También es posible ponerle un nombre al temporizador. Otro parámetro es la remanencia, que consiste en conservar la información después de un corte de alimentación, es decir, después de un corte de alimentación y su restablecimiento, el temporizador continúa por donde se haya quedado.

## ACTIVIDAD RESUELTA 5.4

Realiza un programa para que, al activar una entrada, una de sus salidas se active transcurridos 2 segundos.

### Solución:

Se selecciona la instrucción del bloque de funciones especiales del retardo a la conexión. La configuración de este bloque se ha llevado a cabo para un valor de 2 segundos (Figura 5.55). Como entrada, se ha tomado la I1 y, como salida, la Q1.

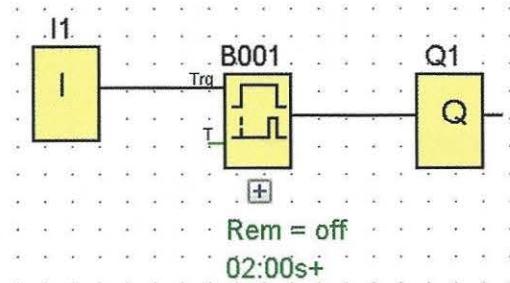


Figura 5.55. Programa de ejemplo de aplicación de un temporizador con retardo a la conexión.

### Temporizador a la desconexión

El temporizador a la desconexión, también llamado *retardo a la desconexión*, al recibir una señal a la entrada (Trg), activa la salida (Q) y comienza la cuenta del tiempo ( $t$ ) establecido. Transcurrido este tiempo, su salida se desactiva.

Si durante la cuenta del tiempo ( $t$ ), recibe otro nivel alto a su entrada, este tiempo se reinicia y, al desactivarse la entrada, comienza de nuevo la cuenta del tiempo.

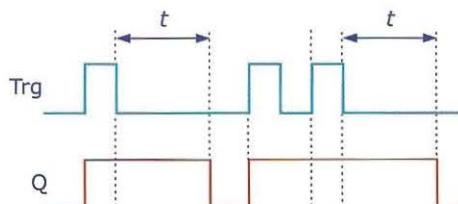


Figura 5.56. Cronograma de un temporizador con retardo a la desconexión.

La configuración del tiempo en el temporizador con retardo a la desconexión es similar al modo de configurar el temporizador con retardo a la conexión.

## ACTIVIDAD RESUELTA 5.5

Desarrolla un programa para que, al activar una entrada, una de sus salidas se active y, transcurridos 2 segundos, se desactive.

### Solución:

Se selecciona la instrucción del bloque de funciones especiales del retardo a la desconexión.

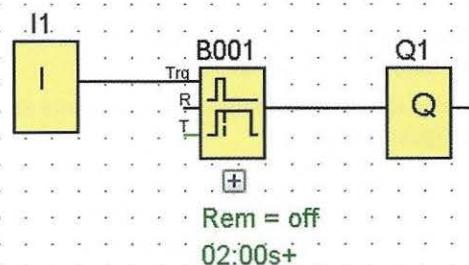


Figura 5.57. Programa de ejemplo de aplicación de un temporizador con retardo a la desconexión.

### Temporizador a la conexión/desconexión

Este bloque de función se comporta de manera similar a la suma del temporizador con retardo a la conexión y el temporizador con retardo a la desconexión, incorporando ambas funciones en un solo bloque.

### Temporizador semanal

El bloque del temporizador semanal permite general señales de activación en función del día de la semana y durante un periodo de tiempo.

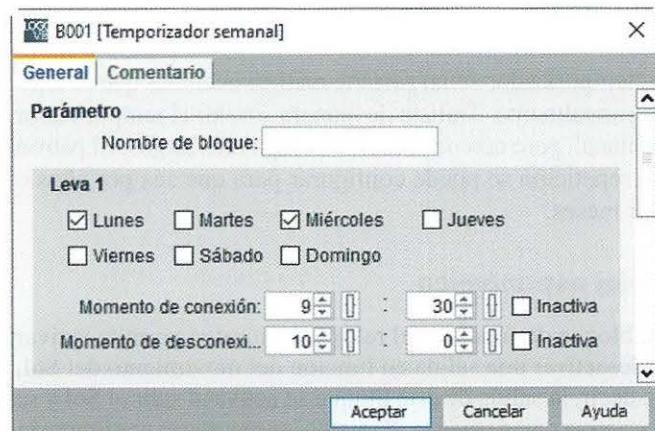


Figura 5.58. Configuración de las propiedades del temporizador semanal.

La configuración consta de tres levas y, en cada leva, se indica el momento de la conexión y de la desconexión, así como qué días de la semana tendrá lugar este evento. En el ejemplo mostrado en la Figura 5.58, se ha configurado para que la leva 1 se active todos los lunes y miércoles desde las 9:30 horas hasta las 10:00 horas.

## ACTIVIDAD RESUELTA 5.6

Lleva a cabo la automatización del riego de un jardín mediante un autómata para que se realice todos los lunes y miércoles de 9:30 horas a 10:00 horas.

### Solución:

Se selecciona la instrucción del bloque de funciones especiales del temporizador semanal.

La programación de este temporizador semanal se muestra en la Figura 5.59.

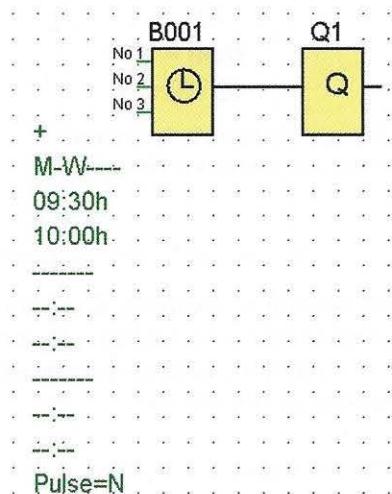


Figura 5.59. Programa de ejemplo de aplicación de un temporizador semanal.

### Temporizador anual

El temporizador anual permite realizar acciones que se repiten anualmente. Trabaja de manera similar al temporizador semanal, pero con períodos de tiempo más largos. El patrón de repetición se puede configurar para que sea por años o por meses.

### Reloj astronómico

El bloque de función del reloj astronómico permite activar y desactivar una salida en función del movimiento del Sol, es decir, la salida de este bloque se activa al salir el Sol y se desactiva con su puesta. Para ello, es necesario configurarlo con la ubicación geográfica (longitud y latitud). Este bloque se adapta al cambio horario de verano e invierno.

### 5.10.2. Contadores

Los contadores son elementos destinados a llevar una cuenta de las veces que recibe un pulso por su entrada. Los contadores pueden realizar una cuenta en modo ascendente o descendente. También se dispone de una variante que realiza una cuenta del tiempo que ese bloque está activado, siendo útil principalmente para tareas de mantenimiento.

Tabla 5.16. Tipos de funciones especiales de contadores

|  |   |
|--|---|
|  | <b>Contador adelante/atrás.</b> Realiza una cuenta en modo ascendente o descendente.<br>Entrada <b>Cnt.</b> Cuenta los pulsos (flancos de subida).<br>Entrada <b>Dir.</b> Indica la dirección de la cuenta (0 ascendente y 1 descendente).<br>Entrada <b>R.</b> Reinicia el contador. |
|  | <b>Contador de las horas de funcionamiento.</b>   |
|  | <b>Selector de umbral.</b>  |

### Contador adelante/atrás

El bloque de función del contador dispone de una entrada de pulsos (Cnt) que cuenta los pulsos que recibe. El modo de realizar la cuenta, es decir, si es ascendente o descendente, se indica por la entrada de dirección (Dir), en la cual, si tiene un nivel lógico de 0, la cuenta es ascendente y, si es un 1, es descendente. Si esta entrada no se emplea, la considera como un 0, es decir, ascendente. Si se necesita poner al valor inicial la cuenta, se activa la entrada de reset (R).

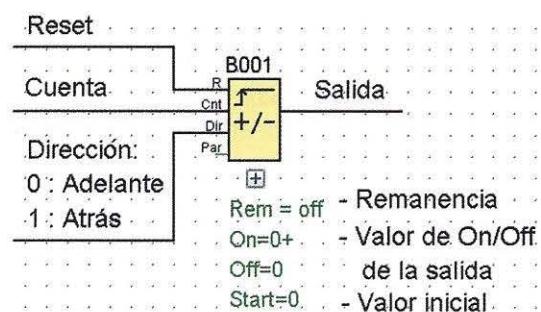


Figura 5.60. Parámetros del boque de función contador adelante/atrás.

La configuración de este contador consiste en indicar cuál es el valor inicial (Start), normalmente será 0, aunque es posible indicarle otro valor. La salida se activará cuando se alcance el valor denominado *on* y se desactivará al alcanzar el valor *off*, ambos configurables. Si se desea que el contador tenga remanencia (Rem), es decir, que conserve el



valor de la cuenta incluso ante un corte de la alimentación eléctrica, se activa.

En la Figura 5.61, se muestra una configuración en la cual se comenzará la cuenta desde el valor inicial 2 y esta función activará la salida cuando se alcance la cuenta de valor 5 y se desactivará cuando alcance el valor 8.

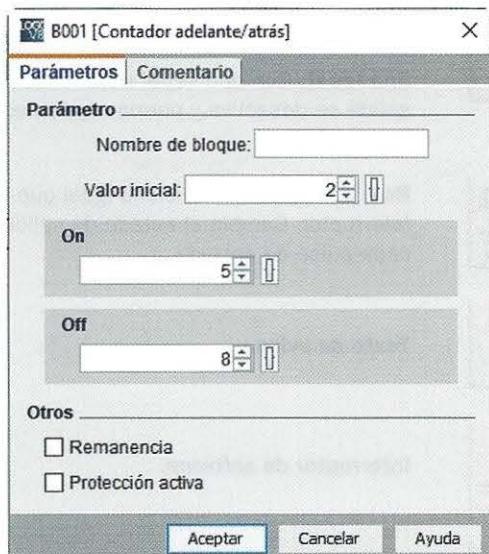


Figura 5.61. Configuración de las propiedades del contador adelante/atrás.

## ACTIVIDAD RESUELTA 5.7

A una sala se entra por una puerta y se sale por otra puerta distinta. En ambas puertas, hay un sensor de barrera fotoeléctrica. Se desea automatizar el motor de ventilación de tal manera que, cuando en la sala haya 5 o más personas, se active este motor de ventilación. Se dispone de un pulsador de puesta a cero del contador. Realiza la tabla de asignación de entradas y salidas y el programa para el autómata.

### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

Tabla 5.17. Tabla de asignaciones de entradas y salidas de la actividad

| Elemento eléctrico | Conexión autómata | Descripción                                  |
|--------------------|-------------------|--|
| -B1                | I1                | Detector fotoeléctrico situado en la entrada |
| -B2                | I2                | Detector fotoeléctrico situado en la salida  |
| -S1                | I3                | Pulsador de puesta a cero                    |
| -K1                | Q1                | Contactor para el motor de ventilación       |

El programa es el siguiente:

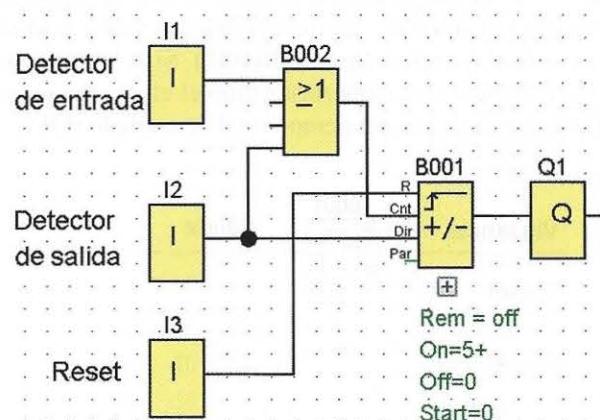


Figura 5.62. Programa de ejemplo de aplicación de un contador adelante/atrás.

Por la entrada Cnt del contador, se reciben los pulsos provenientes de los detectores situados a la entrada y a la salida. Cuando recibe un pulso del detector de salida, se activa a la vez la dirección, indicando que la cuenta es descendente y suma en modo atrás (o lo que es lo mismo, resta). Mediante el pulsador de reset, se pone la cuenta al valor inicial.

El contador se ha configurado con un valor inicial 0 (Start). Se activa al alcanzar el valor 5 (On=5+) y, al no poner ningún valor en el parámetro de desactivación (Off), se indica que no tiene un límite superior para desactivar la salida. En este caso, la remanencia es indiferente, por eso se configura a off.

### Contador de horas de funcionamiento

El contador de horas de funcionamiento es una función útil, sobre todo, para las tareas de mantenimiento.

La configuración del contador de horas de funcionamiento consiste en indicar cuál es el intervalo de funcionamiento (MI). Mientras esté activada la entrada de vigilancia (En), el contador lleva la cuenta del tiempo. Al alcanzarse este tiempo, se activa la salida (Q), la cual indica que se ha alcanzado dicho tiempo y que, por ejemplo, se debe realizar algún tipo de mantenimiento sobre el objeto de vigilancia.

Una vez que se realiza el mantenimiento, el contador se puede poner a cero activando la entrada de reset (R), pero el contador sigue llevando una cuenta acumulada (OT). Si se desea poner a cero tanto la cuenta de tiempo como la acumulada, se activa mediante un pulso la entrada de *reset all* (Ral).

En la Figura 5.63, se muestra un ejemplo de una simulación de un contador de horas de funcionamiento. Se ha configurado con un intervalo de mantenimiento (MI) de 3 minutos. En la Figura 5.63, también se observa que quedan

2 minutos (MN) para activar la salida y que lleva un tiempo total de funcionamiento (OT) de 1 minuto. Si se activa la entrada reset (R) el tiempo restante (MN), se iniciaría, igualándose al intervalo de mantenimiento (MI), pero se conservaría el tiempo transcurrido (OT). Si se pulsa *reset all* (Ral), además de comportarse como el efecto del reset, también pondría a cero el tiempo total acumulado (OT).

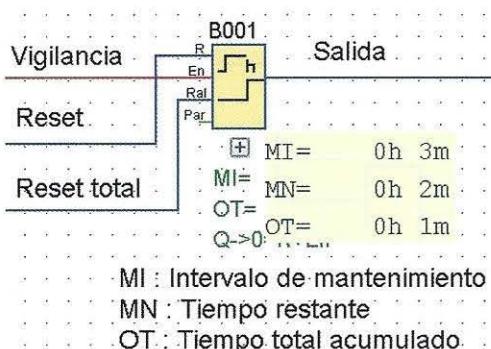


Figura 5.63. Simulación de un contador de horas de funcionamiento.

### 5.10.3. Funciones especiales de tipo analógico

Cuando se trabajan con señales que no son de tipo bit, como son valores con números reales o decimales, las necesidades son diferentes a las vistas hasta ahora. Para ello se han diseñado una serie de funciones para estas características.

Tabla 5.18. Funciones especiales de tipo analógico

|  |  |  |   |
|--|--|--|---|
|  | Instrucción aritmética.                    |  | Rampa analógica.                        |
|  | Comparador analógico.                      |  | Regulador PI.                           |
|  | Conmutador analógico de valor umbral.      |  | Modulación por ancho de impulsos (PWM). |
|  | Amplificador analógico.                    |  | Filtro analógico.                       |
|  | Vigilancia de valor analógico.             |  | Máximo/mínimo.                          |
|  | Conmutador analógico de valor diferencial. |  | Valor medio.                            |
|  | Multiplexor analógico.                     |  |   |

### 5.10.4. Otros bloques de funciones

A parte de las funciones vistas, existen otras que no entran en las categorías anteriores.

Tabla 5.19. Otros bloques de función

|  |   |
|--|---|
|  | <b>Relé autoenclavador</b><br>Entrada <b>S</b> . Con una señal de entrada, la salida se activa y permanece en ese estado.<br>Entrada <b>R</b> . Con una señal de entrada, la salida se desactiva y permanece en ese estado. |
|  | <b>Relé de impulsos.</b> Funciona igual que un telerruptor. Cambia el estado de salida por cada pulso de entrada.   |
|  | <b>Texto de aviso.</b>  |
|  | <b>Interruptor de software.</b>   |
|  | <b>Registro de desplazamiento.</b>  |
|  | <b>Detección de error de la instrucción aritmética.</b>   |

#### Relé autoenclavador

La función de relé autoenclavador se emplea como memoria en la cual se almacena un estado. Se suele emplear, por ejemplo, para utilizar un par de pulsadores, mediante el cual uno de ellos activa este bloque y el otro lo desactiva, es decir, memoriza este estado. Este relé autoenclavador también recibe el nombre de *bloque RS* o *báscula RS*.

Consta de una entrada (set) que activa este bloque y otra entrada (reset) que lo desactiva. Si se activan las dos entradas, tiene preferencia la orden de desactivación (reset).

## ACTIVIDAD RESUELTA 5.8

Realiza el programa para un autómata mediante el cual, con un pulsador, se encienda una lámpara conectada a una de las salidas y, mediante otro pulsador, se apague. Utiliza un relé autoenclavador o bloque RS. Elabora la tabla de asignación de entradas y salidas y el programa para el autómata.



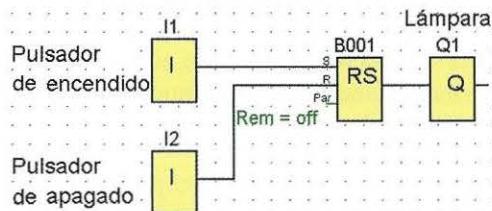
### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

**Tabla 5.20.** Tabla de asignaciones de entradas y salidas de la actividad

| Elemento eléctrico | Conexión autómata | Descripción            |
|--------------------|-------------------|------------------------|
| -S1                | I1                | Pulsador de encendido  |
| -S2                | I2                | Pulsador de apagado    |
| -E1                | Q1                | Lámpara de iluminación |

El programa es el siguiente.



**Figura 5.64.** Programa de ejemplo de aplicación de un relé autoenclavador o báscula RS.

### Relé de impulsos

El bloque de relé de impulso es una variante del relé autoenclavador o báscula RS, el cual cambia de estado cada vez que recibe un pulso por la entrada Trg. Tiene un comportamiento similar al telerruptor. En caso necesario, dispone de una entrada de set (S) para forzar la activación y una entrada de reset (R) para forzar su desactivación. También se puede configurar cuál de las dos entradas tiene prioridad en caso de que se activen las dos a la vez.

## ACTIVIDAD RESUELTA 5.9

Desarrolla el programa para un autómata mediante el cual, con un solo pulsador, se pueda controlar el encendido y apagado de una lámpara de iluminación, cambiando de estado cada vez que se accione. Realiza la tabla de asignación de entradas y salidas y el programa para el autómata.

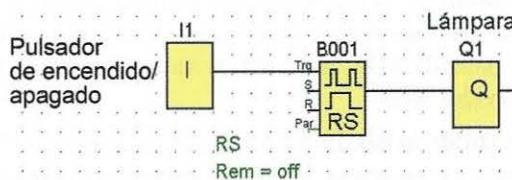
### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

**Tabla 5.21.** Tabla de asignaciones de entradas y salidas de la actividad

| Elemento eléctrico | Conexión autómata | Descripción                   |
|--------------------|-------------------|-------------------------------|
| -S1                | I1                | Pulsador de encendido/apagado |
| -E1                | Q1                | Lámpara de iluminación        |

El programa es el siguiente.



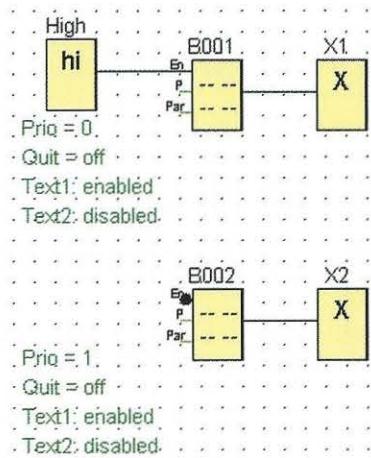
**Figura 5.65.** Programa de ejemplo de aplicación de un relé de impulsos.

### Textos de aviso

Los textos de aviso, que se encuentran en el árbol de operaciones de la parte izquierda en la sección Otros, permiten mostrar mensajes de texto, así como informar del estado de las variables. Una vez insertado el bloque de función del texto de aviso y realizando un doble clic, se entra en la parte de configuración.

Una vez puesto, este bloque debe tener puesta la entrada EN (enable) a uno o nivel alto. Se puede realizar mediante un bloque High o negar esta entrada EN.

La salida de este bloque de texto de aviso no puede dejarse al aire o sin conectar, ya que, de lo contrario, mostrará un error al utilizarlo con el Logo! Se soluciona empleando el bloque de función de conector abierto.



**Figura 5.66.** Conexión del bloque de función de texto de aviso con las dos formas de conectar la entrada de habilitación.

Si se entra en la pantalla de configuración, aparecen varias secciones:

- **Configuración de textos de aviso.** La prioridad es importante cuando se prevea que pueden aparecer varios mensajes a la vez, en este caso, se debe indicar cuál de los mensajes tiene mayor prioridad para mostrar ese.

- Configuración del ticker.** Se emplea cuando se necesita mostrar un texto largo y que este sea deslizante.
- Destino de aviso.** Se le indica dónde se van a mostrar estos mensajes, que pueden ser en el Logo!, en la pantalla TDE, así como si va a tener salida en el servidor web.
- Texto de aviso.** En esta sección, se escribe el texto que va a mostrarse. En su parte superior, aparece una serie de iconos para indicar qué variables mostrar y de qué forma.

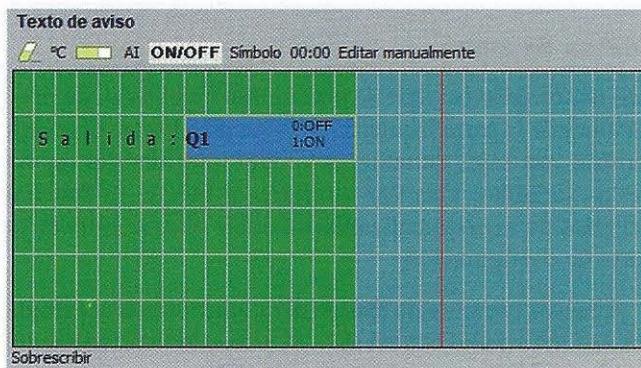


Figura 5.67. Configuración de la sección de textos de aviso.

## ACTIVIDAD RESUELTA 5.10

Realiza el programa para un autómata que, mediante un interruptor conectado a la entrada I1, activa la salida Q1. Muestra mediante un texto de aviso el estado de esta salida.

### Solución:

El programa es el siguiente:

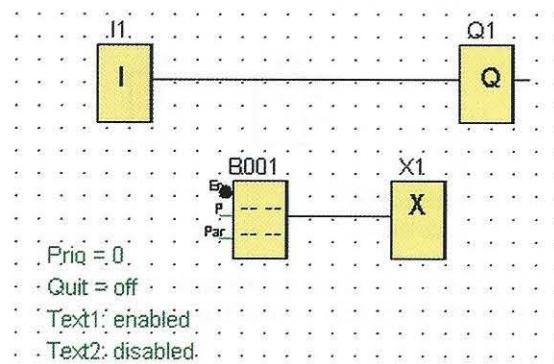


Figura 5.68. Programa de ejemplo de aplicación de los textos de aviso.

Al elegir la variable que va a mostrarse, que es de tipo On/Off (Figura 5.69), se puede personalizar el texto cuando está activado (true) o desactivado (false).

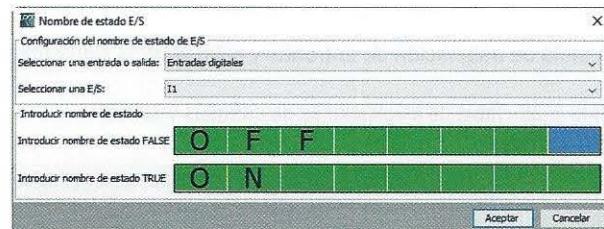


Figura 5.69. Personalización de los nombres o texto que va a mostrarse según su estado.

## 5.11. Funciones definidas por el usuario (UDF)

Cuando una sección de un programa se repite en otros programas, es interesante guardar esa función como UDF para que pueda ser reutilizada.

Para crear una UDF, se realiza el programa; posteriormente, se marcan aquellas partes que compondrán la UDF, y, mediante el botón derecho del ratón, se guarda como UDF.

## ACTIVIDAD RESUELTA 5.11

Diseña un programa que distinga entre una pulsación corta sobre un pulsador y una pulsación larga (más de 1 segundo). Mediante la pulsación corta, se controla la salida Q1 y, mediante la pulsación larga, se controla la salida Q2.

### Solución:

El programa es el siguiente:

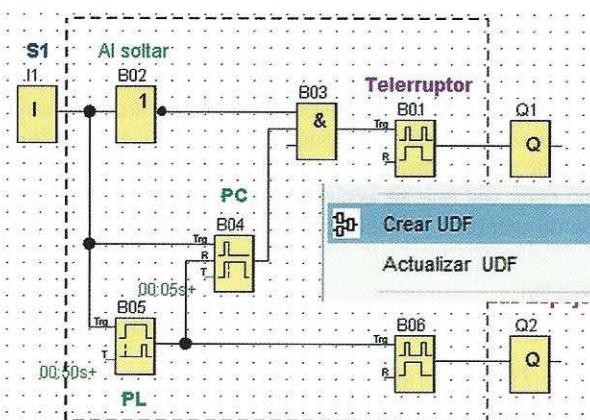


Figura 5.70. Programa con los elementos remarcados para UDF.

A continuación, se pasa a la vista de UDF en la cual se debe guardar. Para poder utilizarlo en otro esquema, en el árbol donde se encuentran las UDF, se acciona el botón



secundario y se pulsa en configurar UDF y agregar (Figura 5.70). Se puede observar que el mismo programa queda visualmente más simplificado (Figura 5.71).

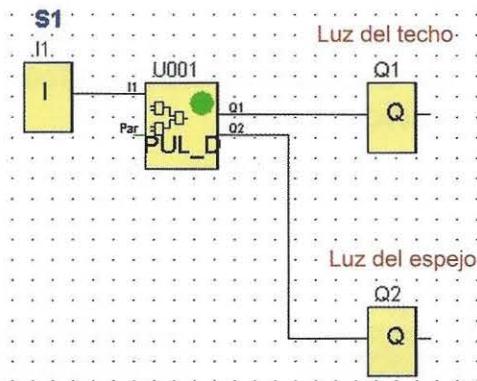


Figura 5.71. Programa con el UDF.

## 5.12. El trabajo con señales analógicas

El Logo! en su versión 24 Vcc tiene las entradas I7 e I8 como analógicas por compatibilidad hacia modelos anteriores y también se pueden utilizar las entradas I1 e I2, pero para ello se deben configurar previamente. Estas dos entradas analógicas por defecto (I7 e I8) se pueden utilizar como digitales.

### Ajuste AI integr.

Si se admiten 4 AI en LOGO!, ¿desea habilitar 2 AI o 4 AI?

Para garantizar la compatibilidad con dispositivos anteriores= habilite 2 AI.

Habilitar 0 AI

No hay AI disponibles para su programa de circuito.

Habilitar 2 AI

Solo pueden utilizarse AI1 y AI2, correspondientes a los bornes de entrada I7 e I8= en el programa de circuito.

Habilitar 4 AI

AI1 y AI2, correspondientes a los bornes de entrada I7 e I8, están disponibles para su uso en su programa de circuito.

Además, AI3 y AI4, correspondientes a los bornes de entrada I1 e I2, están disponibles para su uso.

Figura 5.72. Selección de entradas analógicas.

En el programa, al utilizar una entrada analógica, se identifica como AI1, aunque físicamente se conecta a la entrada I7, tal y como se detalla en su configuración (Figura 5.72).

Junto con el bloque de entrada analógica, se suele utilizar un conmutador analógico de valor umbral, que es un bloque situado en la sección de analógicos. A continuación, el conmutador se debe configurar y para ello se marcan los

umbrales en los cuales se desea que se active. En el ejemplo de la Figura 5.73, se han configurado sus umbrales para que active con un valor analógico de tensión de entradas de 0 V hasta 5 V. El Logo! toma lecturas en tensión de 0 V-10 V y los muestra como valores de 0-1000. Al llenar los valores de rango de medida, al autómata los parametriza.

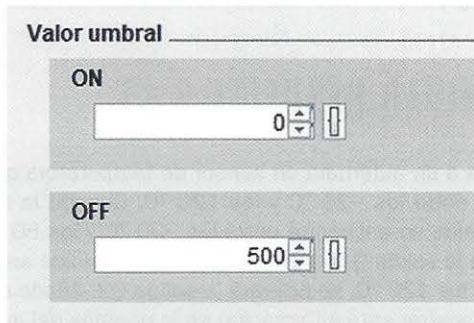


Figura 5.73. Configuración de los umbrales en un conmutador analógico de valor umbral.

## ACTIVIDAD RESUELTA 5.12

Realiza un programa que lea los valores de una entrada analógica (AI1) y active la salida Q1 cuando el nivel de tensión de la entrada analógica sea entre 0 V y 5 V y la salida Q2 sea entre 5 V y 10 V.

### Solución:

El programa es el siguiente:

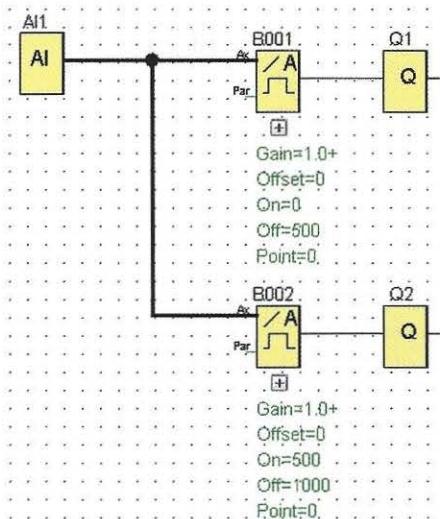


Figura 5.74. Programa de la actividad.

La configuración de los umbrales es:

- Conmutador 1/salida Q1: 0-500.
- Conmutador 2/salida Q2: 500-1000.

### 5.12.1. Escalado de señales analógicas

En función de la señal eléctrica de la entrada analógica (en corriente 4 mA 20 mA o tensión 0 V-10 V), se convierten esos valores entre 0 y 1000. Se debe establecer una relación entre la escala de la magnitud física y los valores con los cuales trabaja el autómata. Esta conversión se denomina *decalaje* y el autómata la realiza automáticamente.

## ACTIVIDAD RESUELTA 5.13

Conecta a un autómata un sensor de temperatura que lea valores entre los  $-20^{\circ}\text{C}$  y los  $120^{\circ}\text{C}$ . Cuando la temperatura leída se encuentre entre los  $-20^{\circ}\text{C}$  y los  $50^{\circ}\text{C}$ , se activará la salida Q1 y, cuando esté comprendida entre los  $50^{\circ}\text{C}$  y los  $120^{\circ}\text{C}$ , se activará la salida Q2. Añade que se pueda mostrar esta información en la pantalla del autómata mediante una barra deslizante.

#### Solución:

El programa es el siguiente:

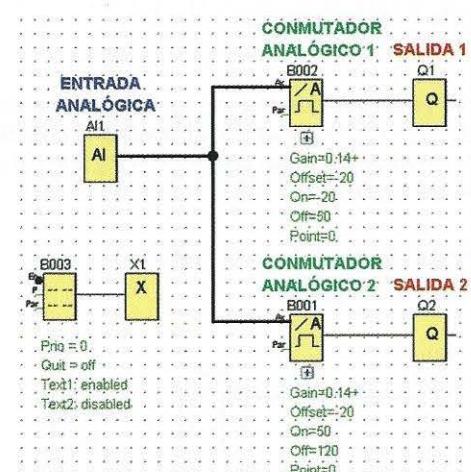


Figura 5.75. Programa de la actividad.

La parametrización de los comparadores es la siguiente:

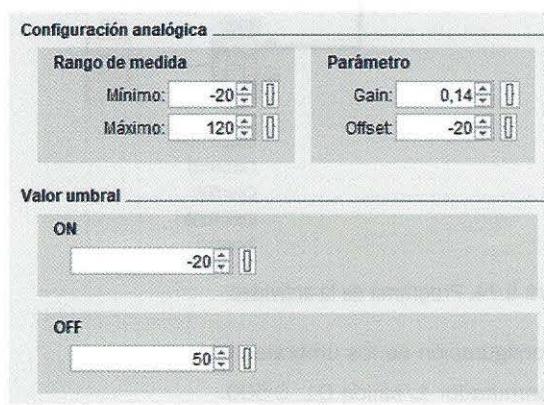


Figura 5.76. Parametrización del comparador 1.

La configuración de los umbrales es:

- Comparador 1/salida Q1: de  $-20$  a  $50$ .
- Comparador 2/salida Q2: de  $50$  a  $120$ .

## 5.13. La comunicación entre varios Logo!

El autómata Logo!, en sus últimas versiones, está preparado para trabajar en red de una manera fácil y de rápida programación.

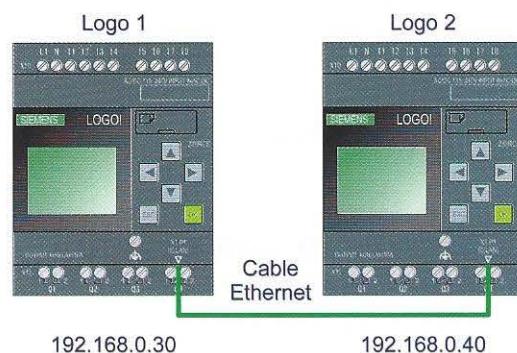


Figura 5.77. Conexión de dos Logo! en red.

Se debe comenzar con un nuevo programa en vista de proyecto de red, donde se añaden los dispositivos que formarán la red (Logo!, ordenador, pantalla TDE, etc.). Al añadir los dispositivos, se configuran con las direcciones de red que van a emplear. Al realizar doble clic sobre el dispositivo, se entra en la parte de programación de este. Una vez finalizada la programación, se procede a su transferencia a cada uno de los dispositivos.

## ACTIVIDAD RESUELTA 5.14

Conecta dos autómatas Logo! en red (Logo1: 192.168.0.30 y Logo2: 192.168.0.40). Cada uno debe disponer de una entrada y una salida digital. Además, la entrada de un Logo! (I1) debe controlar la salida (Q1) del otro.

#### Solución:

Se insertan los dos Logo! en red y se añaden las entradas y salidas correspondientes a cada uno de ellos (Figura 5.78).

Si se quiere conectar una entrada digital del Logo\_1 con la salida digital del Logo\_2, simplemente se arrastra (*drag*) el conector de la entrada del Logo\_1 y se suelta (*drop*) en la salida del Logo\_2. Automáticamente, el programa generará, por un lado, en el Logo\_1, una salida de red y, en el Logo\_2, una entrada de red, estando ambas conectadas entre sí, y, por otro lado, se creará un canal de conexión entre ambos Logo! (en azul).



La conexión queda parametrizada de la siguiente manera:

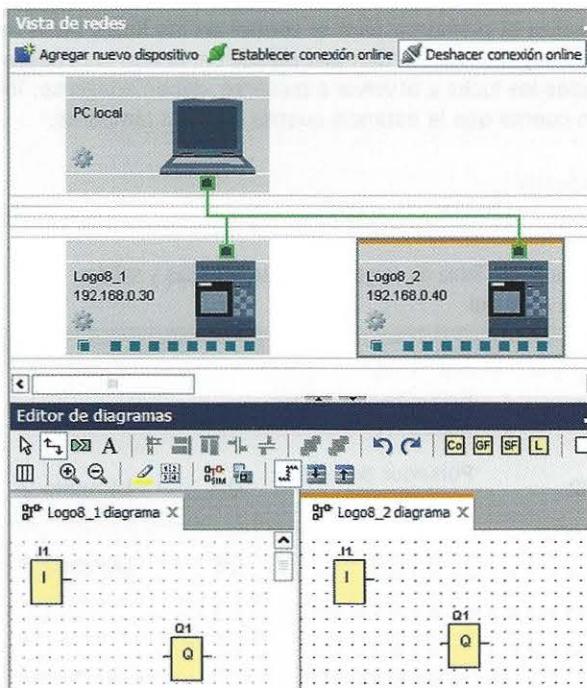


Figura 5.78. Programa de la actividad.

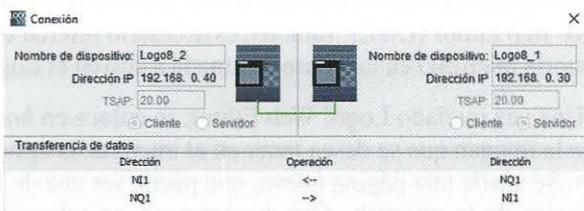


Figura 5.80. Conexión entre Logo!

También se podría haber hecho con la parametrización por el área de variables de entradas/salidas con dispositivos remotos (V0.0).

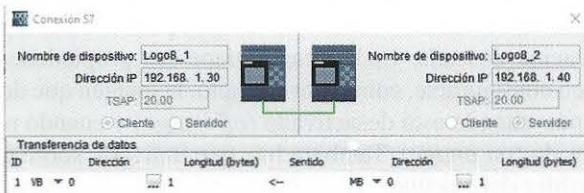


Figura 5.81. Conexión entre Logo! mediante área de variables.

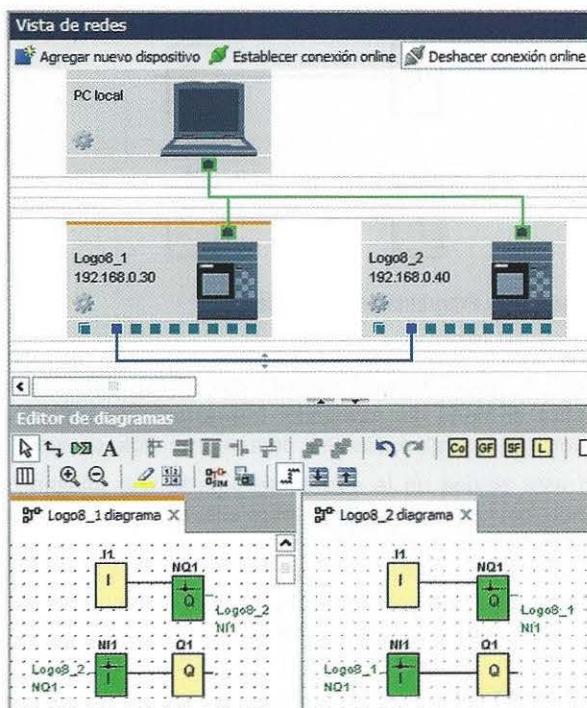


Figura 5.79. Programa con las conexiones de red.

## 5.14. El servidor web del Logo!

El Logo!, a partir de la versión 8, dispone de un servidor web. De esta manera es posible monitorizar e interactuar desde un navegador web, como puede ser el de un ordenador o teléfono inteligente.

SIEMENS



Figura 5.82. Página web que visualiza el autómata Logo! al acceder desde un ordenador.

Recuerda



La contraseña por defecto para entrar en el servidor web del Logo! es LOGO en mayúsculas.

Esta página web, que, por defecto, se proporciona cuando se accede a él, se puede cambiar por otra personalizada.

Para llevar a cabo esta modificación, es necesario programar desde el software Logo!Soft Comfort versión 8.2, utilizando la herramienta proporcionada por Siemens denominada *Logo! Web Editor* (LWE). También es necesario insertar una memoria microSD en la ranura destinada a ello en el Logo!

Una vez iniciado Logo! Web Editor, se coloca en *home page* la imagen que se desea tener en el inicio de la aplicación. Se añade una página nueva, que puede ser una de las estancias de la vivienda. Con *background*, se selecciona la imagen de fondo que, previamente, se ha ubicado en el directorio de instalación *my graph*.

Hay elementos, como puede ser un pulsador o una lámpara, que cambian según su estado (accionado, encendido y apagado). Para ello, se selecciona desde el menú *component/digital value*, arrastrándolo a la imagen para situarlo en la ubicación deseada. Cada imagen tiene sus propiedades y deben configurarse, como, por ejemplo, la imagen que debe adoptar cuando está desactivado (*off image*) y cuando está activado (*on image*). También hay que indicarle sobre qué variables debe actuar.

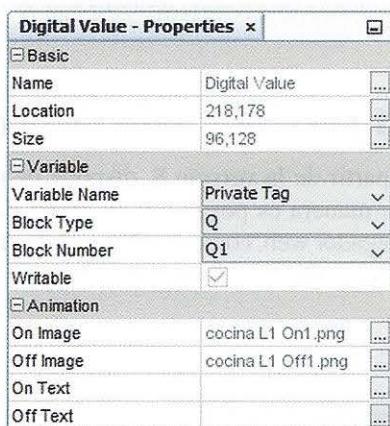


Figura 5.83. Propiedades de un campo *digital value*.

Otro aspecto interesante cuando el proyecto tiene varias páginas es la barra de navegación (*navigator*) que se emplea para poder desplazarse entre las diferentes páginas que componen el proyecto.

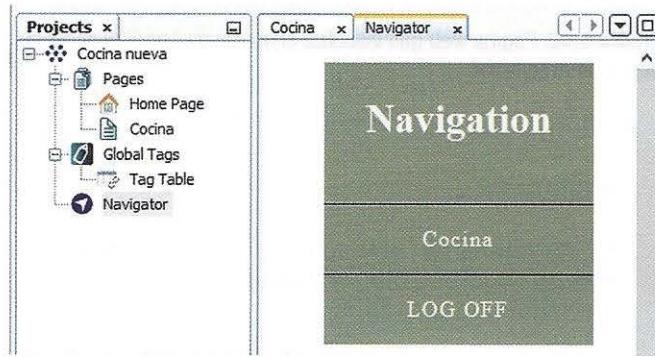


Figura 5.84. Propiedades del navegador.

## ACTIVIDAD RESUELTA 5.15

Realiza el programa para el control de las luces de la estancia de la cocina. Con una pulsación, deben encenderse todas las luces y, al volver a pulsarse, deben apagarse. Ten en cuenta que la estancia cuenta con tres lámparas.

### Solución:

La tabla de asignación de entradas y salidas es la siguiente:

Tabla 5.22. Tabla de asignaciones de entradas y salidas de la actividad

| Entrada | Descripción              | Salidas | Descripción |
|---------|--------------------------|---------|-------------|
| I1      | Pulsador de la lámpara 1 | Q1      | Lámpara 1   |
| I2      | Pulsador de la lámpara 2 | Q2      | Lámpara 2   |
| I3      | Pulsador de la lámpara 3 | Q3      | Lámpara 3   |

El programa es el siguiente:

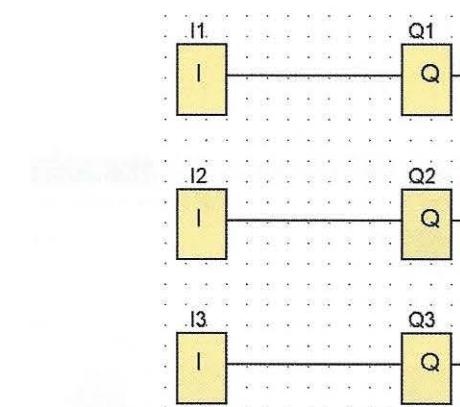
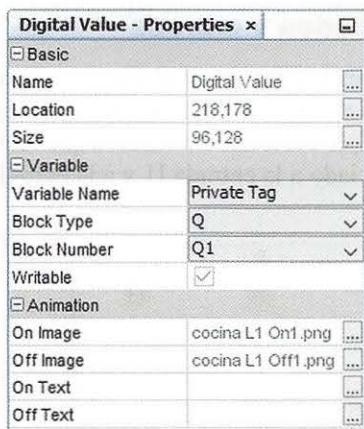


Figura 5.85. Programa de la actividad.

Aparte de la página principal o de inicio (*home page*), se crea una nueva llamada *cocina* (Figura 5.84). También se realiza el menú del navegador (*navigator*) y se enlaza el menú con la página que va a presentarse (*link=cocina*).

En esta página de la cocina, se inserta una imagen de fondo (*background image*) y, sobre ella, se colocan los elementos que se desee como, por ejemplo, la imagen que va a cambiar. Se deben tener preparadas las dos imágenes, una para cuando esté activada y la otra cuando esté desactivada, que se indican mediante la propiedad *on image* y *off image*. También se debe configurar sobre qué variable va a actuar (Q1).

Lo que se le ha indicado mediante las propiedades es que, en función del estado de Q1 (*variable*), muestre una imagen (*animation*).



**Figura 5.86.** Propiedades de la imagen correspondiente a la primera lámpara.

Para el resto de las imágenes, se actúa de similar manera. Por último, se transfiere al autómata.

## 5.15. La programación del Logo! mediante el esquema de contactos

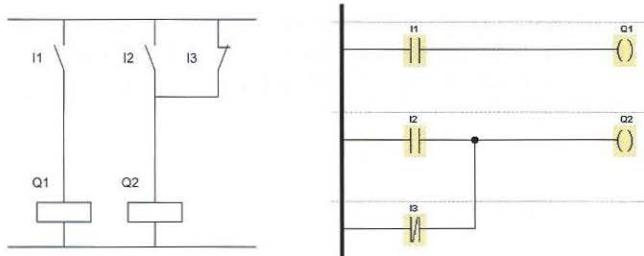
El autómata Logo! de Siemens también permite la programación mediante el esquema o diagrama de contactos o lenguaje *ladder*. Esta forma de realizar la programación es más parecida a los esquemas de electricidad. Consiste en una sucesión de órdenes, donde cada orden se compone de un conjunto de contactos que activa una o varias salidas denominadas *bobinas*.

La simbología básica empleada en el lenguaje o esquema de contactos es la que se muestra en la Tabla 5.23.

**Tabla 5.23.** Símbolos básicos del lenguaje de contactos y su equivalente eléctrico

| Símbolo eléctrico | Símbolo ladder | Descripción      |
|-------------------|----------------|------------------|
| \                 |                | Contacto abierto |
| /                 | /              | Contacto cerrado |
| [ ]               | ( )            | Bobina           |
|                   | (/)            | Bobina invertida |

Los contactos pueden ser abiertos (NO) o cerrados (NC), de similar manera a los contactos empleados en los circuitos eléctricos. En la Figura 5.87, se muestra un ejemplo de comparativa entre los esquemas eléctricos y los esquemas de programación en lenguaje de contactos.



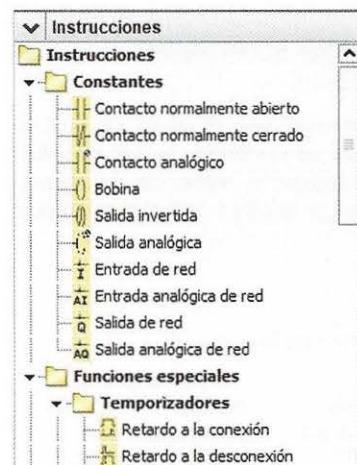
**Figura 5.87.** Comparativa entre un circuito eléctrico y su programa para el Logo!

Los esquemas eléctricos se leen de izquierda a derecha y de arriba abajo, mientras que los esquemas de programación se leen de arriba abajo y de izquierda a derecha.

En este lenguaje de programación, los contactos se colocan a la izquierda y, por último, se colocan las bobinas.

### 5.15.1. La interfaz de programación

La interfaz de programación en esquemas de contactos del Logo!Soft Comfort es similar a la programación en bloques de funciones. En este caso, cambia el árbol de operaciones, que es donde están colocadas las instrucciones para adaptarse a este lenguaje.



**Figura 5.88.** Vista parcial de las instrucciones del árbol de operaciones.

### 5.15.2. Las funciones lógicas

Las funciones lógicas OR y AND en lenguaje de contactos se reducen a una conexión en paralelo y en serie, respectivamente.

## ACTIVIDAD RESUELTA 5.16

Resuelve la Actividad 5.2 empleando el lenguaje de contactos.

Se desea controlar la alarma de intrusión de un sistema domótico, la cual se activará si se detecta la apertura de cualquiera de las dos ventanas de una sala (entradas I1 e I2). Los detectores son de tipo magnético y la alarma (salida Q1) es de tipo visual.

### Solución:

La programación es la siguiente:

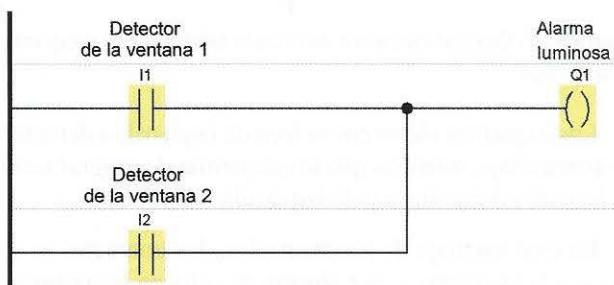


Figura 5.89. Programa de la actividad.

Se observa que la activación de cualquiera de las dos entradas provocará la activación de la salida.

### 5.15.3. Empleo de entradas, salidas y marcas

Un contacto (es indiferente si es abierto o cerrado) se puede utilizar tantas veces como sea necesario, así, el esquema de la Figura 5.91 es correcto. Se observa que se ha repetido el contacto asociado a la entrada I1 y a la entrada I3.

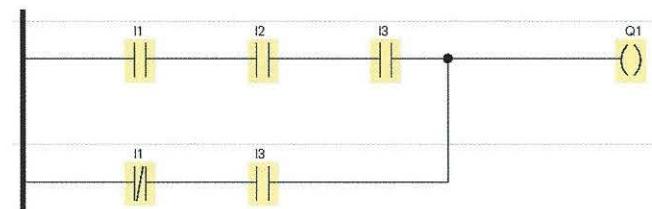


Figura 5.91. Programa con varios contactos asociados a una misma entrada.

Sin embargo, las salidas no se pueden repetir y, por ello, el autómata no dejará realizar dicha operación (Figura 5.92).

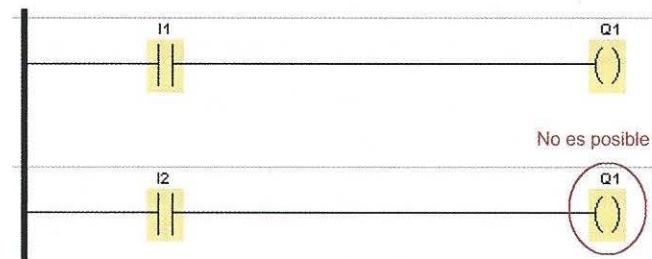


Figura 5.92. Programa con una misma salida duplicada.

En este caso, se soluciona empleando las marcas. Cada entrada activa una marca y estas marcas activan la salida. Las marcas se activan como bobinas y se emplean como contactos, tal y como se muestra en la Figura 5.93.

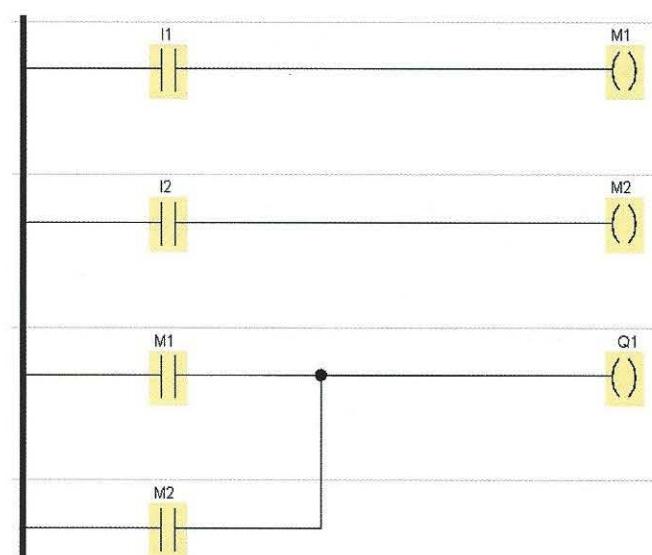


Figura 5.93. Uso de marcas para activar una salida.



## ACTIVIDAD RESUELTA 5.18

Elabora el esquema de programación para el arranque de un motor empleando un pulsador marcha y otro de paro.

### Solución:

La programación es la siguiente:

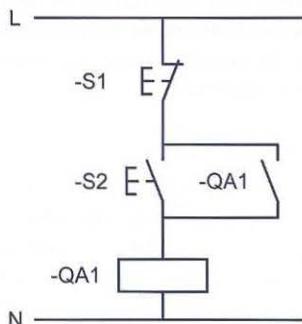


Figura 5.94. Esquema eléctrico del arranque de un motor con pulsador de marcha y paro.

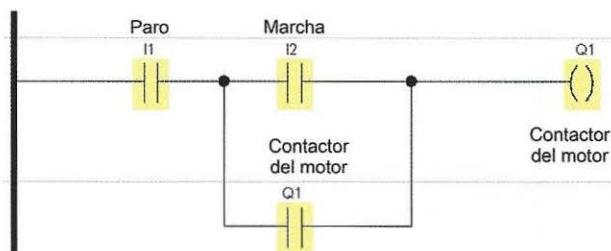


Figura 5.95. Programa de la actividad.

Se observa que este esquema es una traducción del esquema eléctrico. El pulsador de paro, eléctricamente, es de tipo normalmente cerrado (NC), por tanto, en reposo, está cerrado, es decir, deja pasar la corriente. El pulsador de marcha es de tipo normalmente abierto (NO) y, en paralelo, tiene un contacto de la bobina que representa al contactor y que ejerce de realimentación.

### 5.15.4. La función RS

La función del relé autoenclavador o báscula RS se emplea mucho por las ventajas que incorpora, principalmente porque se reúnen las condiciones de activación en la entrada set (S) y las de desactivación en la entrada reset (R), facilitando la compresión del programa.

## ACTIVIDAD RESUELTA 5.19

Diseña el esquema de programación para el arranque de un motor empleando un pulsador marcha y otro de paro. Emplea un relé autoenclavador o báscula RS.

### Solución:

La programación es la siguiente:

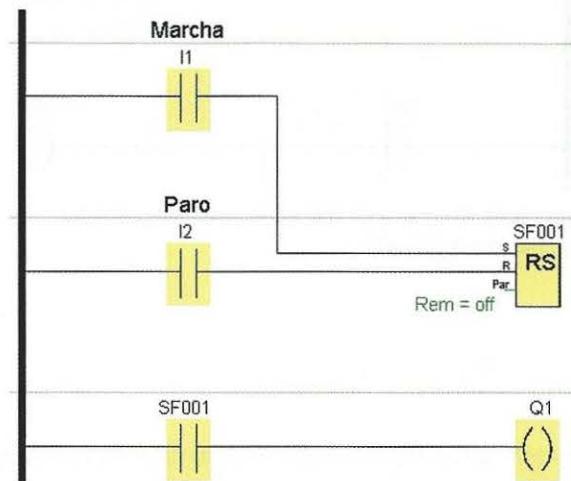


Figura 5.96. Control de una salida con relé RS y pulsadores abiertos.

En este caso, tanto el pulsador de marcha como el de paro son eléctricamente abiertos. Si se quisiera poner el pulsador de paro cerrado, se debería negar el valor de dicha entrada.

### 5.15.5. Temporizadores

El temporizador, como el resto de las funciones, consta de la parte de la bobina y la parte del contacto. Una vez activada la bobina del temporizador, este comienza su trabajo y actúa sobre su contacto. Su configuración es idéntica a la estudiada con los bloques de función.

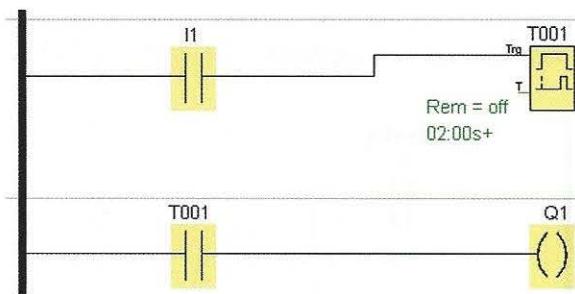
## ACTIVIDAD RESUELTA 5.20

Repite la Actividad 5.4 utilizando el lenguaje de contactos para resolverla.

Realiza un programa para que, al activar una entrada, una de sus salidas se active transcurridos 2 segundos.

### Solución:

Se emplea un temporizador con retardo a la conexión. El pulsador activará la bobina del temporizador configurado para 2 segundos, de tal manera que un contacto de este temporizador activará la salida Q1 transcurrido dicho tiempo. El pulsador I1 debe estar activado durante esta maniobra.



**Figura 5.97.** Programa de aplicación de un temporizador con retardo a la conexión.

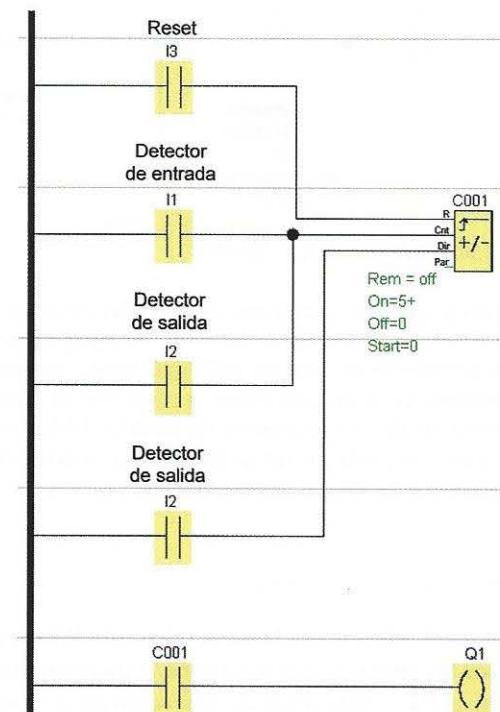
## ACTIVIDAD RESUELTA 5.22

Repite la Actividad 5.7 empleando el lenguaje de contactos para resolverla.

A una sala se entra por una puerta y se sale por otra puerta distinta. En ambas puertas, hay un sensor de barrera fotoeléctrica. Se desea automatizar el motor de ventilación de tal manera que, cuando en la sala haya 5 o más personas, se active este motor de ventilación. Se dispone de un pulsador de puesta a cero del contador. Realiza el programa para el autómata.

### Solución:

El contador tiene tres entradas. Por la entrada Cnt, recibe los pulsos que ha de contar provenientes de los detectores de entrada y salida. Por la entrada de dirección, se indica si la cuenta la realiza adelante o atrás. Si tiene un nivel lógico bajo (0), la dirección es ascendente y, si tiene un nivel lógico alto (1), la dirección es descendente, por ello recibe la dirección desde el sensor de salida. Una tercera entrada (I3) se encarga de poner el contador a su estado inicial (reset).



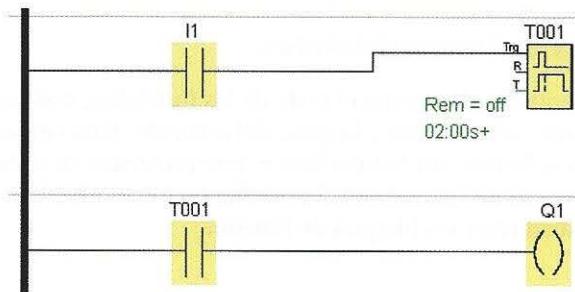
## ACTIVIDAD RESUELTA 5.21

Resuelve la Actividad 5.5 utilizando el lenguaje de contactos.

Desarrolla un programa para que, al activar una entrada, una de sus salidas se active y, transcurridos 2 segundos, se desactive.

### Solución:

Se emplea un temporizador con retardo a la desconexión. El pulsador envía un pulso al temporizador, que se activa, cierra su contacto y activa la salida. Transcurrido el tiempo indicado, se desactiva el contacto del temporizador desconectando la salida Q1.



**Figura 5.98.** Programa de aplicación de un temporizador con retardo a la desconexión.

### 5.15.6. Contador

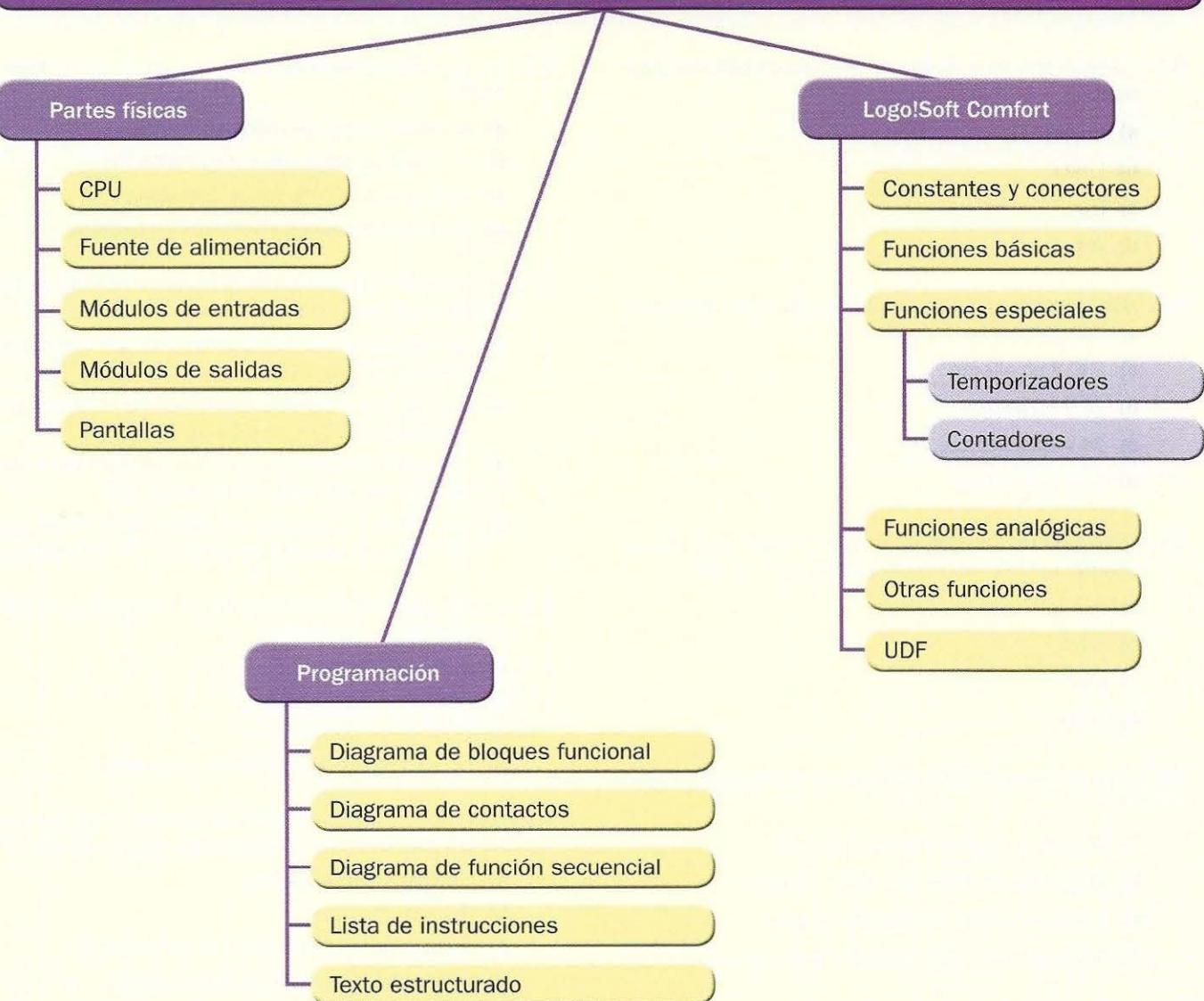
La configuración del contador adelante/atrás es idéntica a la estudiada con la programación en bloques de función y, para su uso, se emplea la parte de bobina para su activación y el contacto asociado para realizar las acciones.

**Figura 5.99.** Programa de aplicación de un contador adelante/atrás.

# Mapa conceptual



## Autómatas programables. Siemens Logo!



## Actividades de comprobación

- 5.1.** ¿De qué tipo es la topología del conexionado del autómata?
- Estrella.
  - Línea.
  - Bus.
  - Árbol.
- 5.2.** ¿Qué tensión se obtiene en bornes de un autómata con salidas libres de potencial?
- 230 V en alterna.
  - 24 V en alterna.
  - 24 V en continua.
  - Es configurable.
- 5.3.** ¿Qué tipo de variable no se puede usar si se desea guardar el valor de una temperatura de 24 °C?
- BOOL.
  - SINT.
  - UINT.
  - DINT.
- 5.4.** ¿A qué circuito eléctrico es equivalente la puerta lógica OR?
- Al circuito serie de varios contactos.
  - Al circuito paralelo de varios contactos.
  - Al circuito mixto de varios contactos.
  - Al circuito de varios inversores en cadena.
- 5.5.** Si, en una red compuesta por dos autómatas, uno de ellos tiene la dirección IP 192.168.45.88, ¿cuál de las siguientes direcciones es válida para el otro autómata?
- 192.168.45.88.
  - 192.192.45.45.
  - 192.168.45.2.
  - 1.1.1.88.
- 5.6.** ¿A qué circuito eléctrico es equivalente la puerta lógica AND?
- Al circuito serie de varios contactos.
  - Al circuito paralelo de varios contactos.
  - Al circuito mixto de varios contactos.
  - Al circuito de varios inversores en cadena.
- 5.7.** ¿Cuál de las siguientes afirmaciones sobre las salidas del autómata es correcta?
- Las salidas a relé solo pueden trabajar con receptores a 230 V en alterna.
  - Las salidas a transistor solo pueden trabajar con receptores a 230 V en alterna.
  - Las salidas a transistor tienen una mayor velocidad de comutación respecto a las de relé.
  - Las salidas a relé no necesitan tener protección eléctrica, ya que están protegidas por el autómata.
- 5.8.** ¿A qué es equivalente una señal de valor lógico 1 cuando se invierte dos veces?
- A un nivel lógico de 0.
  - A un nivel lógico de 1.
  - Provoca un cortocircuito.
  - Un valor no se puede invertir dos veces.
- 5.9.** ¿Qué tipo de elemento se debe emplear para que una lámpara se encienda 2 minutos desde que se activa?
- Contador adelante/atrás.
  - Relé de impulsos.
  - Temporizador de retardo a la desconexión.
  - Temporizador de retardo a la conexión.
- 5.10.** ¿Cuál de las siguientes afirmaciones sobre el relé autoenclavador es correcta?
- Si se pone a 1 la entrada S, la salida se pone a 0.
  - Si se pone a 0 la entrada S, la salida se pone a 0.
  - Si se pone a 1 la entrada R, la salida se pone a 1.
  - Si se pone a 1 la entrada S y a 1 la entrada R, la salida se pone a 0.



## Actividades de aplicación

**5.11.** Realiza el control de la iluminación de una estancia, sabiendo que, con un pulsador, se enciende y, con otro pulsador, se apaga.

Indica:

- a) La tabla de asignación de entradas y salidas.
- b) El esquema eléctrico de conexiones del autómata.
- c) La programación en bloques de funciones sin emplear el relé autoenclavador.
- d) La programación en bloques de funciones empleando el relé autoenclavador.
- e) La programación en lenguaje de contactos sin emplear el relé autoenclavador.
- f) La programación en lenguaje de contactos empleando el relé autoenclavador.

**5.12.** Desarrolla el control de la iluminación de un pasillo que tenga tres pulsadores.

Indica:

- a) La tabla de asignaciones de entradas y salidas.
- b) El esquema eléctrico de conexiones del autómata.
- c) La programación en bloques de funciones.
- d) La programación en lenguaje de contactos.

**5.13.** Lleva a cabo el control de la iluminación de un dormitorio que tiene dos lámparas, una de techo y otra en la mesita de noche. La lámpara del techo se controla desde dos posiciones (en la entrada y en la mesita), mientras que la lámpara de la mesita de noche, se maneja desde una posición. Emplea pulsadores para ello.

Indica:

- a) La tabla de asignaciones de entradas y salidas.
- b) El esquema eléctrico de conexiones del autómata.
- c) La programación en bloques de funciones.
- d) La programación en lenguaje de contactos.

**5.14.** Establece un control de la iluminación compuesto por dos lámparas, donde, al accionar el pulsador de encendido, se encienda la lámpara 1 y, a los 5 segundos, se encienda la lámpara 2. Ambas deben apagarse al accionar el pulsador de apagado.

Indica:

- a) La tabla de asignaciones de entradas y salidas.
- b) La programación en bloques de funciones.
- c) La programación en lenguaje de contactos.

**5.15.** Un cuarto de baño consta de una lámpara y de un extractor de aire. Al accionar un pulsador, se enciende la lámpara y el extractor y, al volver a accionarlo, se apaga la luz, pero el extractor sigue funcionando 5 segundos más.

Indica para este caso:

- a) La tabla de asignaciones de entradas y salidas.
- b) La programación en bloques de funciones.
- c) La programación en lenguaje de contactos.

**5.16.** La iluminación de un jardín consta de dos zonas de iluminación. Al accionar el pulsador de encendido, se ilumina la zona 1 y, 3 segundos después, se ilumina la zona 2. Al accionar el pulsador de apagado, se desactiva la iluminación de la zona 1 y, 5 segundos después, la zona 2.

Realiza para este caso:

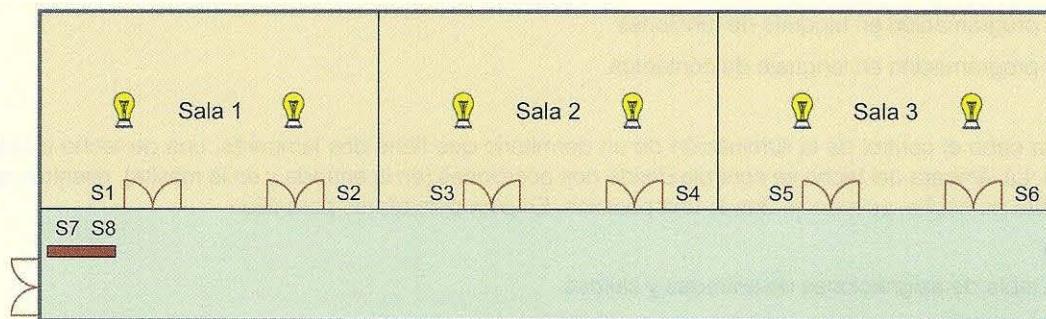
- La tabla de asignaciones de entradas y salidas.
- La programación en bloques de funciones.
- La programación en lenguaje de contactos.

**5.17.** Se desea realizar la iluminación de una zona de un jardín compuesta por cuatro lámparas para que sea secuencial, de tal manera que, al accionar el pulsador de encendido, se ilumine la lámpara 1, un segundo después, la lámpara 2, y así sucesivamente. Con otro pulsador, se apagan todas de golpe.

Determina para este caso:

- La tabla de asignaciones de entradas y salidas.
- La programación en bloques de funciones.
- La programación en lenguaje de contactos.

**5.18.** Realiza el control de la iluminación de un restaurante que tiene tres salas de comedor. Cada sala dispone de dos pulsadores para encender y apagar la luz de esa sala. Además, desde la recepción, se pueden encender y apagar todas a la vez de manera centralizada.



Indica:

- La tabla de asignaciones de entradas y salidas.
- La programación en bloques de funciones.
- La programación en lenguaje de contactos.

**5.19.** Realiza un programa para un aparcamiento que cuenta con 20 plazas, donde un sensor de espira detecte tanto la entrada (I1) como la salida (I2) de un vehículo y levante la barrera correspondiente. Cuando hayan pasado por el detector de infrarrojos, deberá bajar la barrera correspondiente.

Visualiza los datos de plazas libres y de aparcamiento libre/completo a través de una pantalla Logo! TDE.

**5.20.** Automatiza la puerta de un garaje convencional. Para ello hay que disponer de un interruptor de apertura por mando a distancia IR (I1), un motor que accione la puerta de subida y bajada (Q1 subida y Q2 bajada) y dos finales de carrera para la detección de que la puerta está completamente abierta (I3 e I4). También debe poseer un piloto exterior que avise durante todo el tiempo de la maniobra de apertura y cierre (Q3).

En todo momento, a través de la pantalla del Logo!, o Logo! TDE, debe poder verse el estado de la puerta.

Al darle al mando de apertura, la puerta ha de abrirse hasta llegar al FC superior, donde esperará un tiempo predefinido y comenzará a descender hasta llegar al FC inferior.

Si, bajando la puerta, se cruza algún objeto y es detectado por la fotocélula o se le da orden de subida, la puerta debe cesar la bajada, esperar un tiempo también predefinido y comenzar la subida de nuevo, repitiendo el ciclo de subida.