

# 7 Programación de PLC's: Lenguaje en Esquemas de Contacto

## 1. Lenguaje en esquemas de contacto

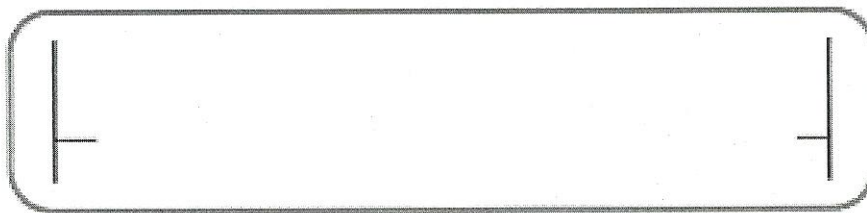
La lógica de escalera o Ladder Diagram (LD) es el lenguaje de programación más usado para la programación de PLCs. Fue el primero con el que se comenzó a programar, de ahí que presente grandes semejanzas con los diagramas eléctricos de escalera utilizados por los técnicos anteriormente a la aparición del autómata.

Este lenguaje está especialmente indicado para facilitar el cambio de un sistema de control realizado con relés por un PLC. Otra ventaja de este lenguaje es que reduce el tiempo de formación de los programadores.

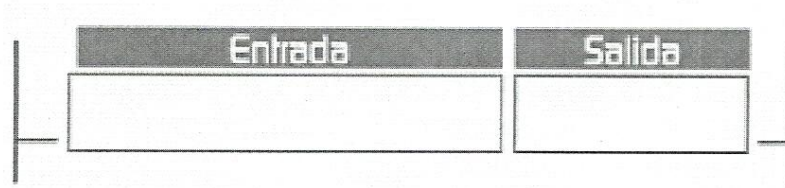
El lenguaje en plano de contactos es aquel en el cual las operaciones lógicas que rigen un automatismo se resuelven mediante contactos normal abierto y normal cerrado, además de otros bloques especiales.

El diagrama de contactos (ladder diagram LD) es un lenguaje que utiliza un juego estandarizado de símbolos de programación. La gran ventaja de este lenguaje estriba en que los símbolos básicos, están normalizados según normas NEMA y son empleados por todos los fabricantes. En el estándar IEC el número de símbolos ha sido reducido.

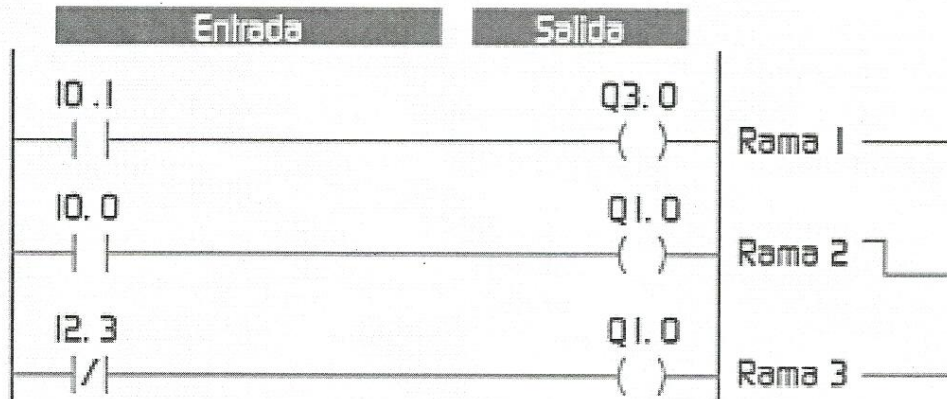
El esquema se realiza entre dos líneas o barras de alimentación dispuestas verticalmente a ambos lados del diagrama, entre ellas se dibujan los elementos del lenguaje.



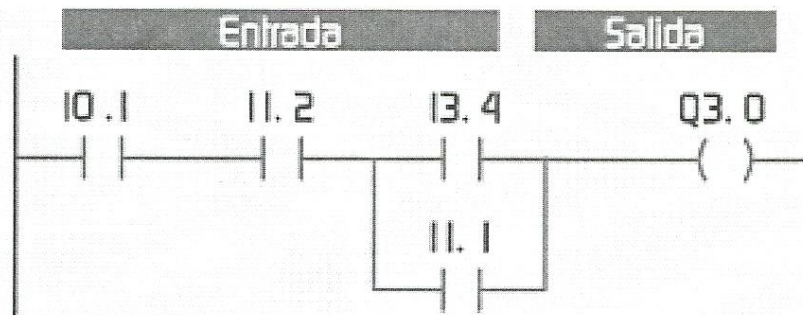
A la derecha del esquema se ubican los elementos de salida y a la izquierda los de entrada



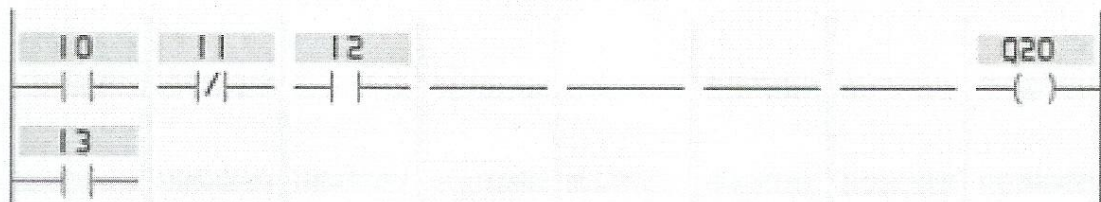
El diagrama puede tener varias ramas o escalones también llamadas líneas de programa (RUNG) o **SEGMENTOS**.



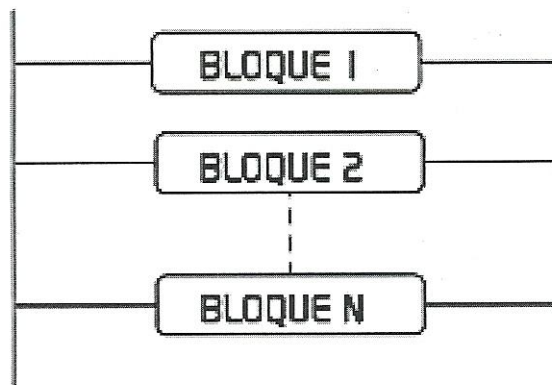
Cada rama permite ubicar varios elementos de entrada pero sólo uno de salida.



La programación en cada bloque de contactos se realiza en el orden de izquierda a derecha.



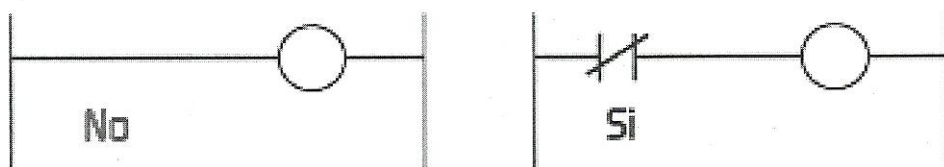
El sentido de programación de los bloques de contactos de un programa o líneas de programa se ejecuta en el sentido de arriba abajo.



### 1.1. Reglas del lenguaje

El número de contactos que se pueden colocar en un bloque, desde el comienzo de la línea principal hasta la salida, es ilimitado. No obstante como limitación práctica se tendrá en cuenta la anchura del papel cuando queramos sacar el programa por impresora o anchura del periférico de programación que se esté utilizando.

No se puede conectar una salida directamente a la línea principal, en estos casos se intercala un contacto cerrado de una marca, bit o relé interno cualquiera.



A cada entrada o salida disponible en el autómata, este le asigna un número. Este número de entrada o salida, dependiendo del software de programación de cada fabricante, se le puede nombrar con una etiqueta la cual se suele definir por la función de esa entrada o salida en el sistema a automatizar, este hecho ayuda a la programación y comprensión del programa. Con relación a estos números, con que el autómata define a sus entradas y salidas es imprescindible tener presente lo siguiente:

- **Contactos de entrada:** El número de entrada, normalmente abierta o normalmente cerrada, se puede utilizar ilimitadamente a lo largo de las líneas que componen el programa.
- **Acciones o contactos de salida:** El número de salida es fijo, por lo que no se puede repetir un mismo número de salida.

## 1.2. Elementos del lenguaje

Se clasifican en elementos de entrada y salida. Su estado es evaluado por el PLC para determinar un valor lógico, que recibe distintas denominaciones dependiendo del contexto de trabajo. A continuación, se presenta una tabla donde se relacionan las distintas denominaciones que se pueden encontrar (activo e inactivo).

Contexto	Activo	Inactivo
Informática	True / Verdadero	False / Falso
Algebra de Boole	V / 1	F / 0
Electrónica digital	High / H	Low / L

### Elementos de entrada

Los contactos, únicos elementos que se colocan a las entradas pueden ser de los tipos que se describen a continuación. Encima del contacto se escribe la variable a la cual hace referencia. El valor lógico del contacto depende directamente del valor lógico de su variable.

- **Normalmente abierto.** Activa la línea hacia la derecha de la instrucción cuando el contacto se activa. Si la variable es 1 el contacto también será 1 y, si la variable es 0 el contacto será 0





- **Normalmente cerrado.** Desactiva la línea hacia la derecha de la instrucción cuando el contacto se activa. Este contacto toman el valor inverso de su variable, si la variable es 1 el contacto será evaluado como 0 y viceversa.

--|/|--

- **Transición positiva o flanco positivo.** Activa la línea hacia la derecha de la instrucción cuando el contacto está desactivo en el scan anterior y activo en el scan actual.

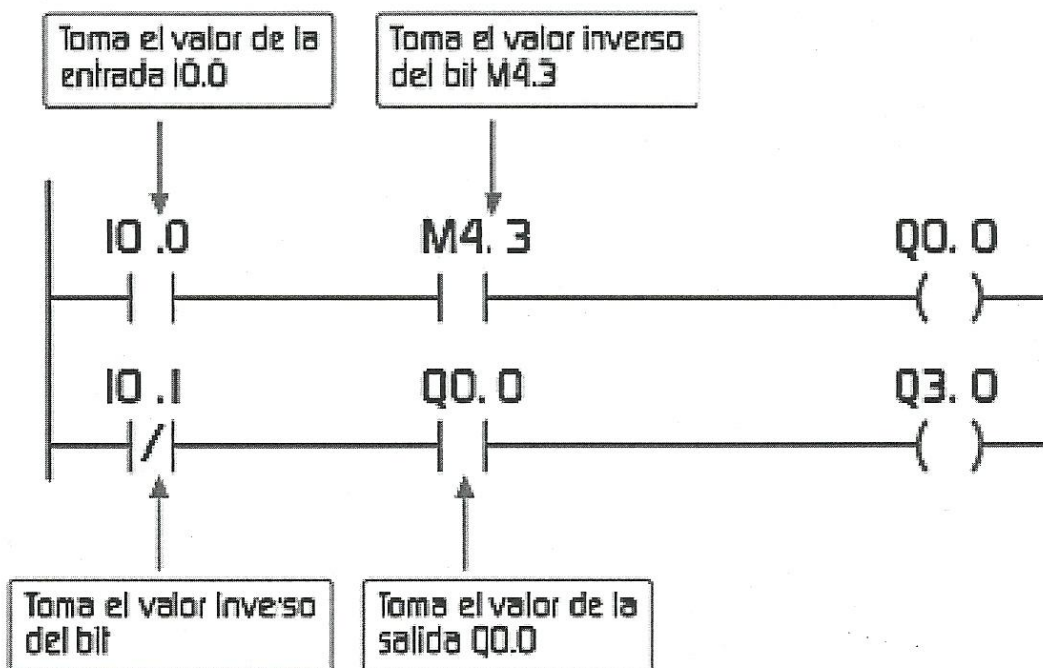
--|P|--

- **Transición negativa o flanco negativo.** Activa la línea hacia la derecha de la instrucción cuando el contacto está activo en el scan anterior y desactivo en el scan actual.

--|N|--

Las variables a las cuales pueden referirse los contactos son:

- Entradas digitales que toman el valor de la entrada digital (I1.3).
- Salidas digitales que toman el valor de la salida digital (Q1.5).
- Bits en memoria, son bits localizados en la memoria con posibilidad de ser definido por el usuario, también se conoce como relés internos, bits de estado) y se utilizan muy a menudo para el control de temporizadores y contadores.



### Elementos de salida

También llamados acción. A los elementos de salida, al igual que para la entrada, se les escribe la variable a la cual están referidos. El valor lógico del elemento de salida es determinado por el PLC a partir de la combinación serie paralelo de los elementos de entrada. El elemento de salida principal se denomina Asignación o Bobina. Estas acciones pueden ser de los tipos que se describe a continuación.

- **Acción simple:** su valor lógico es igual al resultado de la combinación de los contactos en la rama. Si el resultado de la evaluación de los contactos es V entonces la bobina será V; si el resultado es F, la bobina toma el valor F.



- **Acción negada:** su valor lógico es igual al inverso del resultado de la combinación de los contactos en la rama. Si el resultado de la evaluación de los contactos es V entonces la bobina será F; si el resultado es V, la bobina toma el valor V.



- **Puesta a uno (SET):** cuando llega el valor V a esta bobina, su variable asociada se pone y mantiene indefinidamente en estado V sin importar que a la bobina llegue posteriormente un valor F. Una vez retenida la variable en el valor V, para pasarla a F será necesario el uso de una bobina de puesta a 0.

--(S)--

- **Puesta a cero (RESET):** Cuando llega un valor V a esta bobina, su variable asociada se pone y mantiene indefinidamente en estado F sin importar que a la bobina llegue posteriormente un valor F. la única manera de cambiar el estado de la variable es usando una bobina de puesta a 1.

--(R)--

- **Acción activa por flanco de subida:** activa un bit cuando la instrucción de entrada transiciona de false a true.

--(P)--

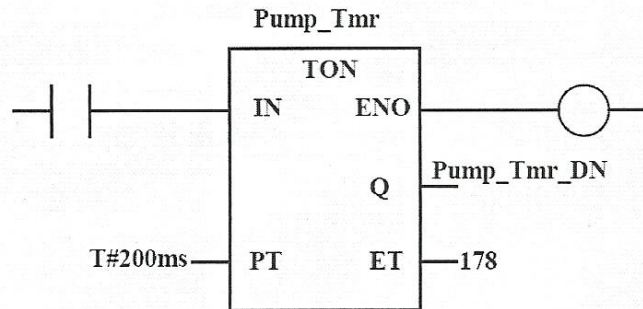
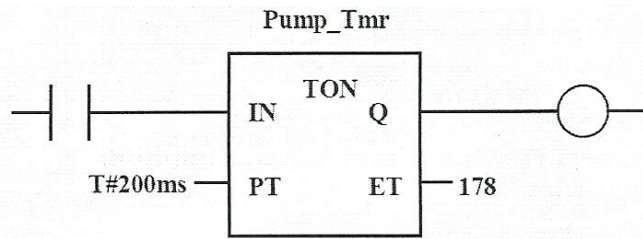
- **Acción activa por flanco de bajada:** activa un bit cuando la instrucción de entrada transiciona de true a false.

--(N)--

Otros tipos de elementos de salida son: Temporizadores, Contadores, Saltos, Llamadas y Retornos, estos elementos también se pueden implementar en el resto de lenguajes de programación. Es importante recordar que los tipos los siguientes elementos, así como su funcionamiento dependen no solo del fabricante sino de la CPU o gama de PLC dentro del mismo fabricante.

- **Temporizadores:** Dependiendo del fabricante del PLC pueden existir distintos tipos de contadores, a continuación se explican los más usuales a través de su diagrama de tiempos. Estos temporizadores pueden estar definidos mediante símbolos de acciones o más corrientemente, por ser de más fácil comprensión, mediante bloques funcionales, tal y como se especifica en la siguiente imagen.





Se puede observar una serie de parámetros de programación y de activación así como de salida:

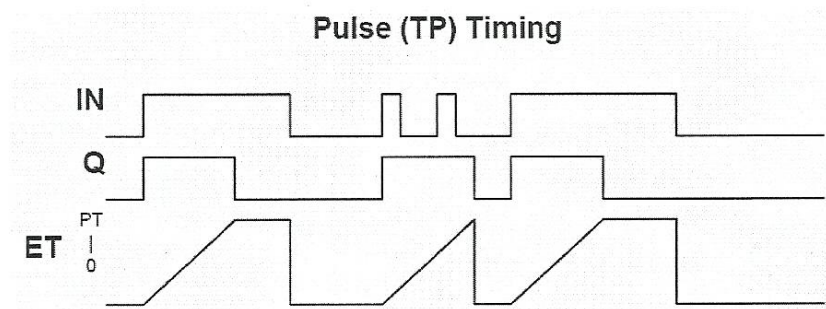
**IN:** instrucción de entrada de la línea de contactos que activan esta salida o acción.

**Q:** Resultado de la temporización, varía con el tipo de temporización (0 o 1). La salida del temporizador es cualquier contacto al cual se le haya asignado como variable de referencia el nombre del temporizador.

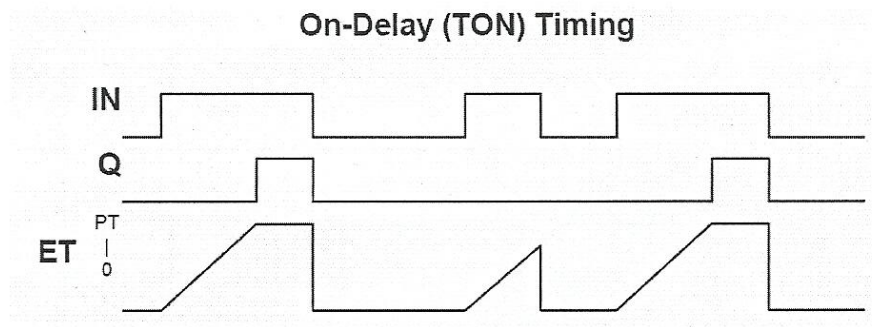
**PT:** (Preset Time) parámetro que indica cuanto tiempo se va a temporizar o asignación de la base de tiempo del temporizador.

**ET:** (Elipse Time), indica el transcurso de la temporización hasta que llega a la base de tiempo definida en PT.

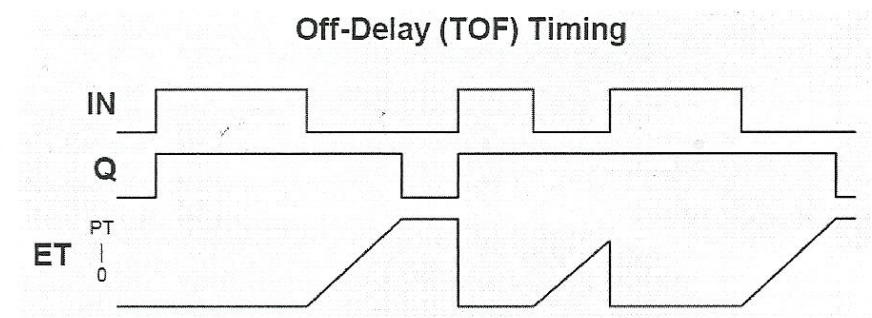
**-Temporizador de impulso:**



**-Temporizador con retardo a la conexión:**



**-Temporizador con retardo a la desconexión:**



- **Contadores:** existen contadores ascendentes, descendentes y ascendentes-descendentes. En el último caso, con un mismo bloque se puede contar o descontar. Las opciones de programación de los contadores son, dependiendo del autómatas tiene un número determinado de contactos disponibles:

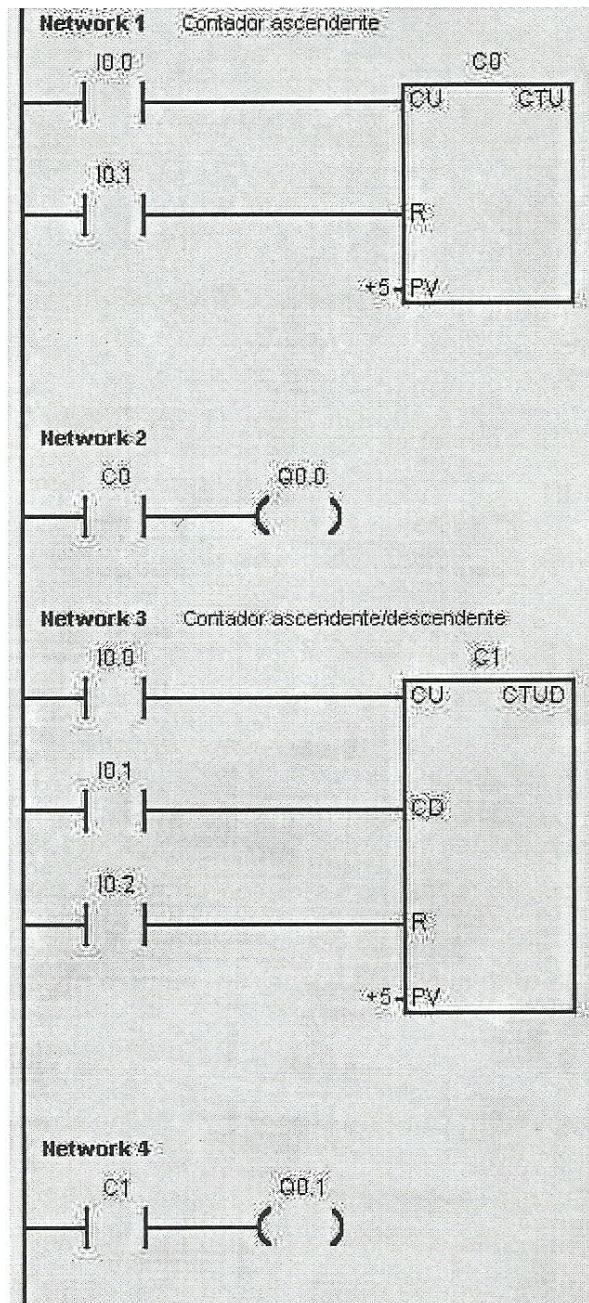
- **(PV)Asignación:** con este elemento se define el valor inicial de la cuenta

- **(CU) Cuenta ascendente:** un flanco de subida en la entrada del elemento hace que el valor de la cuenta se incremente en 1. El flanco de subida se define como el cambio de una señal de F a V.

- **(CD) Cuenta descendente:** con un flanco de subida se hace que el valor de la cuenta descienda en 1.

- **(R) Reposición:** esta entrada obliga a que el contador se reinicie con su valor inicial.

La salida de un contador es un contacto cuya variable de referencia sea el nombre del contador, la variable es F mientras el valor de la cuenta sea 0 y es V si la cuenta es diferente de 0.

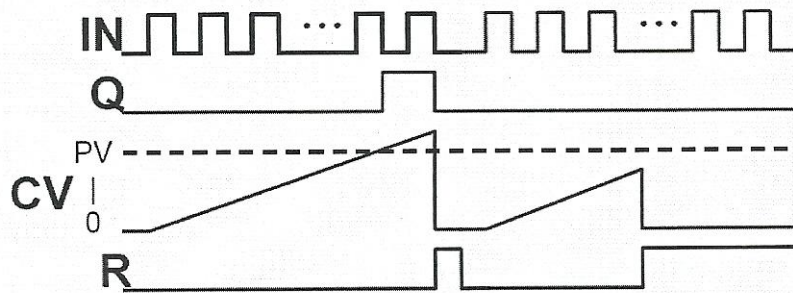


A continuación se indican los diagramas para los distintos contadores:

- **Contador Ascendente :**

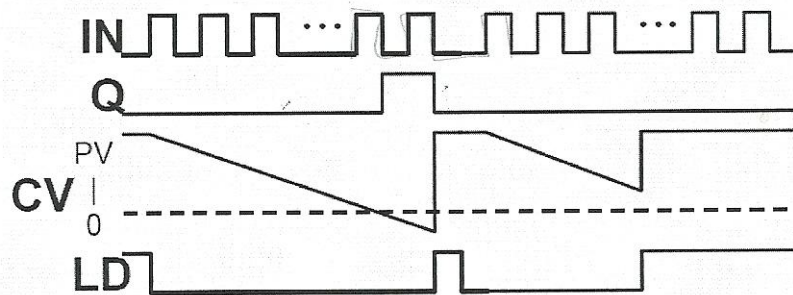


### Count Up (CTU) Counter

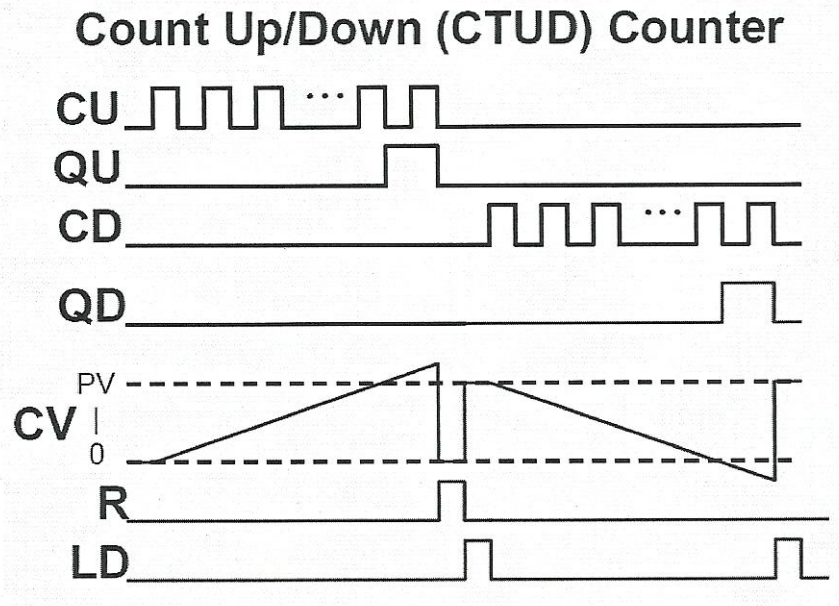


- Contador Descendente :

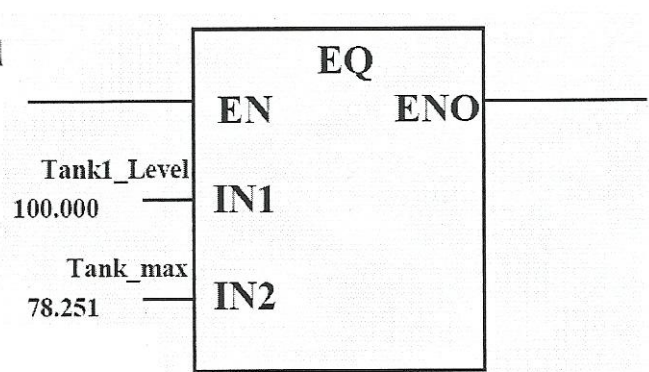
### Count Down (CTD) Counter



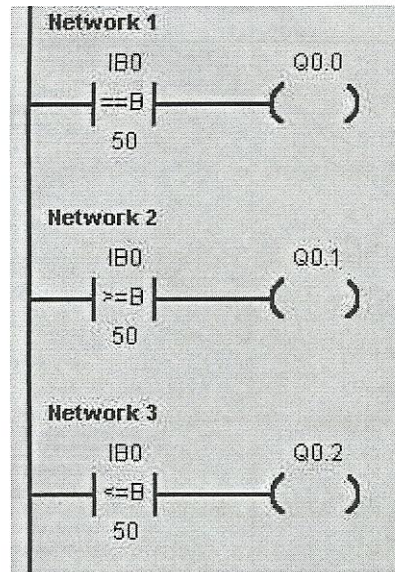
- Contador Ascendente-Descendente :



- **Comparadores:** si la línea de entrada está activa (**EN**), la instrucción ejecuta la operación y activa la línea de salida (**ENO**) cuando el parámetro **IN1** es igual al parámetro **IN2**. Se pueden comparar distintos tipos de números (bits, enteros, dobles enteros...)

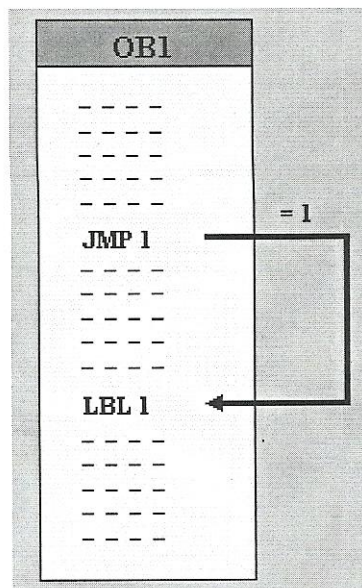


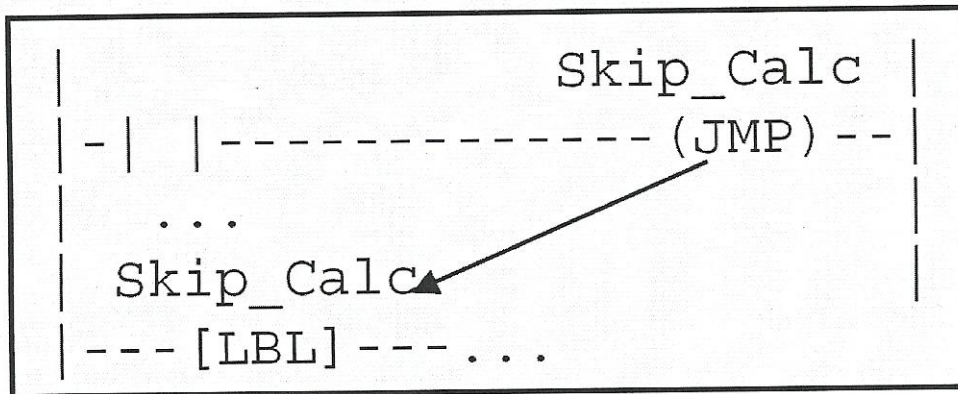




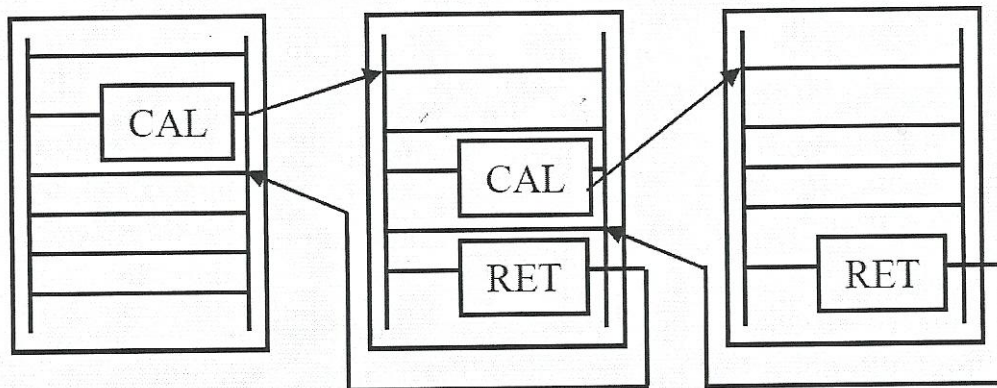
- **Ruptura de la secuencia de ejecución:** suelen ser de dos tipos.

**-Instrucciones de salto a etiquetas o metas:** salta a un bloque de código del programa. **JMP** ejecuta el salto cuando se activa la instrucción o línea de entrada, el salto se produce a la línea de programa con la marca LBL o Label, en el cual se indica el nombre de la etiqueta de la operación de salto

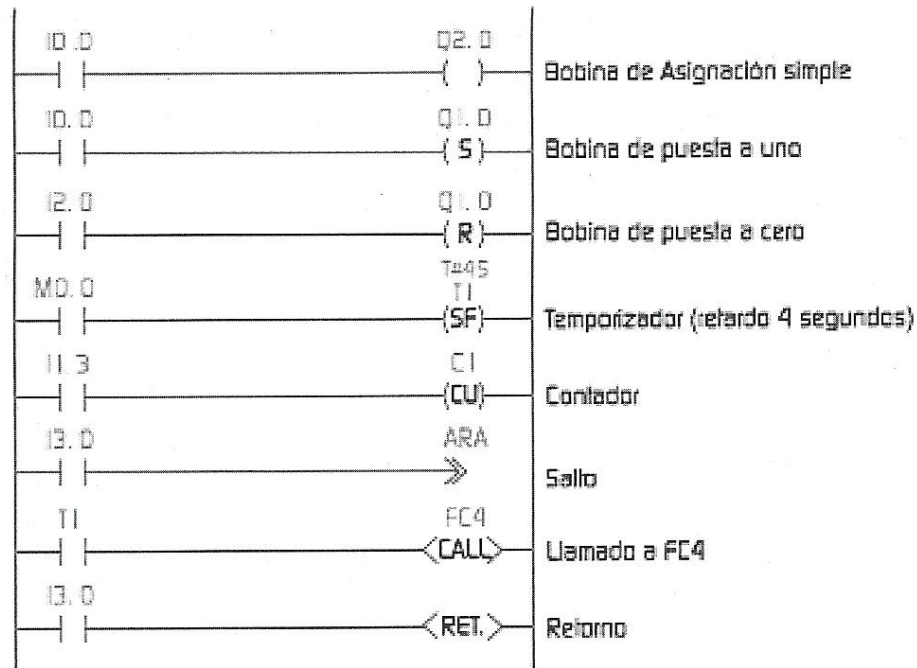




**-Instrucciones de salto a subrutinas:** salta a un bloque de código encapsulado como una subrutina. Esta compuesto por la operación **CALL** que pasa el control a otra función y la función **RET** que retorna al punto siguiente desde donde fue llamada la subrutina



En la siguiente imagen se muestran algunos de los elementos vistos anteriormente en lenguaje de esquema de contactos para un determinado autómeta.



### 1.3. Ejemplo resuelto mediante esquema de contactos

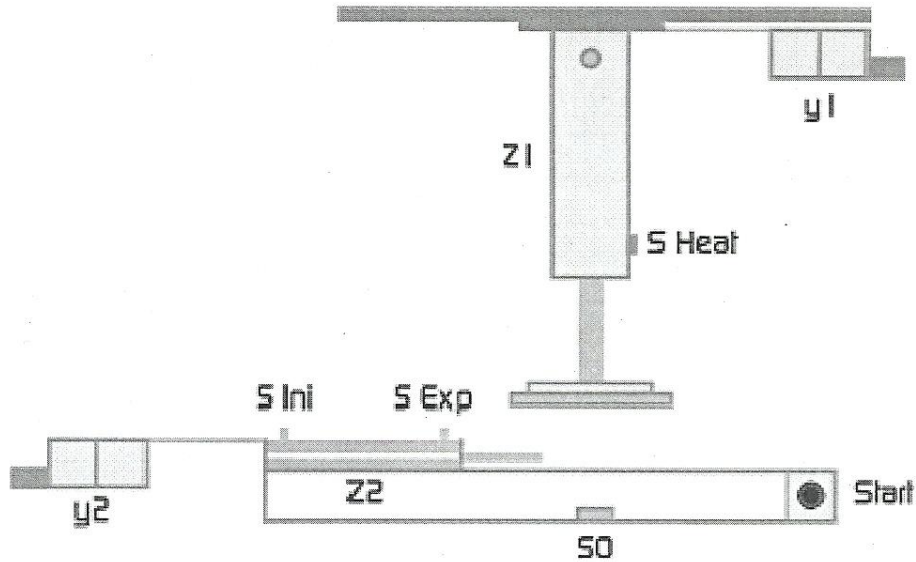
- **Estampadora semiautomática:**

En una línea de producción en serie se estampan 1500 piezas, el proceso inicia al presionar el pulsador y termina cuando se enciende una luz y suena una sirena para dar aviso que se han fabricado las 1500 piezas. Cuando se posiciona una pieza en el área de estampado se acciona la prensa la cual se mantiene presionando la pieza durante 10 segundos, al cabo de los cuales se retira la prensa. Se esperan tres segundos adicionales para que la pieza repose y entonces se retira. Desde un proveedor se desplazará, por gravedad, otra pieza al área de estampado.

#### Proceso del Estampado

1. Posicionamiento de la pieza (en bruto) en el área de estampado
2. Accionamiento de la prensa
3. Presión de 10 seg.

4. Apertura de la prensa
5. Reposo de la pieza 3 seg.
6. Retiro de la pieza fabricada



El sistema consta de los siguientes componentes:

1. Z1, cilindro de simple efecto, encargado de realizar el estampado
2. Y1, electroválvula para comando del cilindro Z1.
3. S\_HEAT, sensor de fin de carrera, el cual detecta cuando Z1 está completamente extendido
4. Z2, cilindro de simple efecto, encargado de expulsar la pieza
5. Y2, electro válvula para comando de Z2
6. S\_INI, sensor de fin de carrera, que indica cuando Z2 está retraído
7. S\_EXP, sensor que indica cuando Z2 está extendido
8. S0, sensor de presencia que indica cuando la pieza se halla en el área de estampado
9. LUZ, luz indicadora de fin de producción
10. SIREN, sirena indicadora de fin de producción



11. START, pulsador inicio de producción

Una vez que se saben las entradas y salidas (sensores y actuadores) que intervienen en el proceso a automatizar se procede a la asignación de estas variables a las entradas y salidas del PLC:

Símbolo	Variable	Descripción
START	10.0	Pulsador, accionado por el operario para iniciar la producción
S_HEAT	10.1	Fin de carrera, indica que Z1 está en posición de estampado
S_INI	10.2	Fin de carrera, indica que Z2 está en su posición inicial
S_EXP	10.3	Fin de carrera, indica que Z2 acaba de expulsar la pieza
S0	10.4	Detector de proximidad, indica que hay una pieza en el área de estampado
Y1	Q0.0	Cilindro de simple efecto, conforma la prensa de estampado
Y2	Q0.1	Cilindro de simple efecto, para retirar la pieza procesada
LUZ	Q0.2	Bombilla, indica al operario que la producción se completó
SIREN	Q0.3	Sirena, indica al operario que la producción se completó
T_ESTAMP	T1	Temporizador de retardo de conexión utilizado para contar los 10 segundos de prensado de la pieza
T_COLD	T2	Temporizador de retardo de conexión memorizado, para contar el tiempo de reposo de la pieza
CUENTA	C1	Contador, para contar los 1500 estampados
MO	M0.0	Marcador interno, 1 = en producción, 0 = fin de producción



MO, bit en memoria, señala que el proceso de producción esta en marcha, se activa al presionar el botón de START y se desactiva cuando el contador desciende hasta 0 (cero).

El contador CUENTA es inicializado por START. MO debe encargarse de que la CUENTA sólo se inicie una vez durante la producción ya que el botón START podría presionarse varias veces, accidentalmente, durante el proceso.

(SET M0) = (NOT MO) AND (START).  
 (C1, #1500) = (NOT MO) AND (START).

Z1 avanza cuando se está en producción, caso en el cual una hay pieza localizada en el área de estampado, y por ende el cilindro expulsor Z2 esta retraído.  
 (SET Y1) = M0 AND INI AND S0.

Z1, se retraerá cuando T\_STAMP completa su tiempo de retardo.  
 (RESET Y1) = T\_STAMP.

El disparo a T\_STAMP se efectúa cuando Z2 alcanza la posición HEAT.  
 (T\_STAMP, T#10Seg)= HEAT.

Se aprovecha HEAT para disparar T\_COLD, por 3 segundos más.  
 (T\_COLD, T#13Seg)= HEAT.

Al cabo del retardo de T\_COLD se inicia el proceso de expulsión.  
 (SET Y2)= T\_COLD.

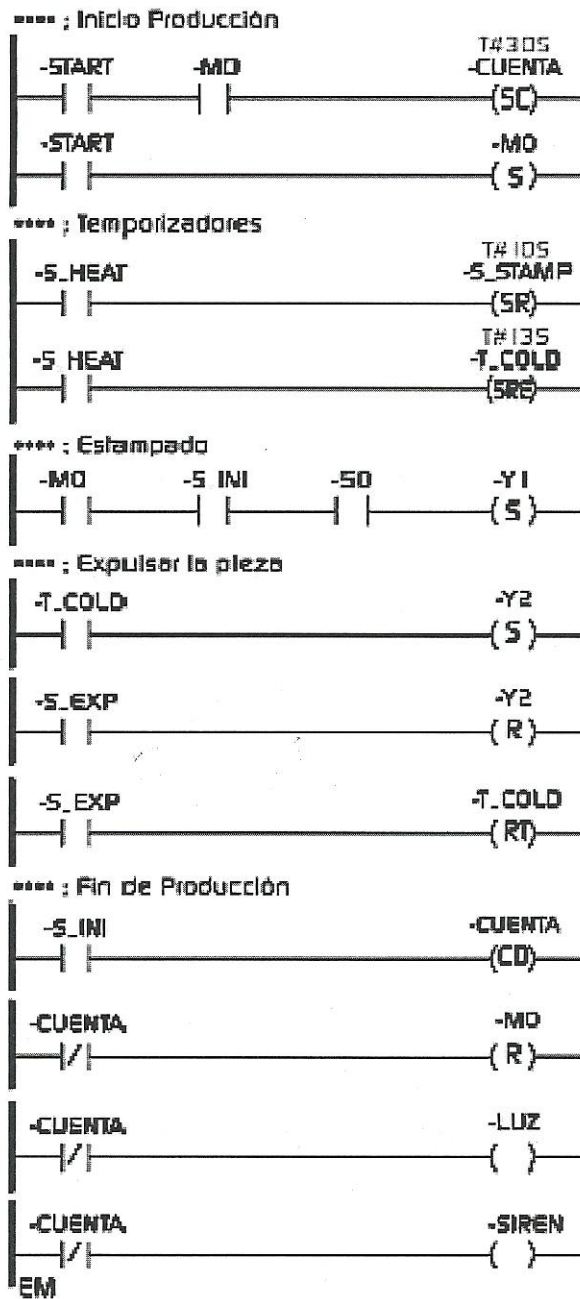
Z2 se retrae con la señal S\_EXP, siendo necesario efectuar la reposición del temporizador T\_COLD  
 (RESET Y2)= S\_EXP.  
 (RESET T\_COLD)= S\_EXP.

Al alcanzar Z2 a INI se aprovecha para el contador CUENTA descienda (conteo descendente CUENTA)= INI.

En este punto el sistema está listo para un nuevo ciclo de estampado. Al cumplirse los 1500 ciclos el valor de la cuenta habrá llegado a 0 (cero), con lo cual se debe desactivar MO y encender la luz y la sirena.

(RESET MO)= (NOT CUENTA).  
 LUZ= (NOT CUENTA).  
 SIREN= (NOT CUENTA).

A continuación se inserta como quedaría el Programa:



## RECUERDA

- La lógica de escalera o Ladder Diagram (LD) es el lenguaje de programación más usado para la programación de PLCs. Fue el primero con el que se comenzó a programar, de ahí que presente grandes semejanzas con los diagramas eléctricos de escalera utilizados por los técnicos anteriormente a la aparición del autómeta.
- El diagrama de contactos (Ladder Diagram LD) es un lenguaje que utiliza un juego estandarizado de símbolos de programación. La gran ventaja de este lenguaje estriba en que los símbolos básicos, están normalizados según normas NEMA y son empleados por todos los fabricantes. En el estándar IEC el número de símbolos ha sido reducido.
- El número de contactos que se pueden colocar en un bloque, desde el comienzo de la línea principal hasta la salida, es ilimitado. No obstante como limitación práctica se tendrá en cuenta la anchura del papel cuando queramos sacar el programa por impresora o anchura del periférico de programación que se esté utilizando.
- Con relación a estos números, con que el autómeta define a sus entradas y salidas es imprescindible tener presente lo siguiente:
  - **Contactos de entrada:** el número de entrada, normalmente abierta o normalmente cerrada, se puede utilizar ilimitadamente a lo largo de las líneas que componen el programa.
  - **Acciones o contactos de salida:** el número de salida es fijo, por lo que no se puede repetir un mismo número de salida.
- Los elementos del lenguaje se clasifican en elementos de entrada y salida. Su estado es evaluado por el PLC para determinar un valor lógico, que recibe distintas denominaciones dependiendo del contexto de trabajo.