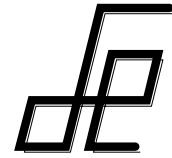




Universidad
de Alcalá

www.infoPLC.net



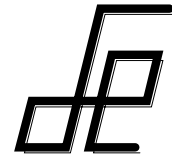
DEPARTAMENTO DE
ELECTRÓNICA

Máster universitario en automatización de procesos industriales

Departamento de Electrónica
Universidad de Alcalá



Universidad
de Alcalá



DEPARTAMENTO DE
ELECTRÓNICA

Programación en Ladder. 1ª parte

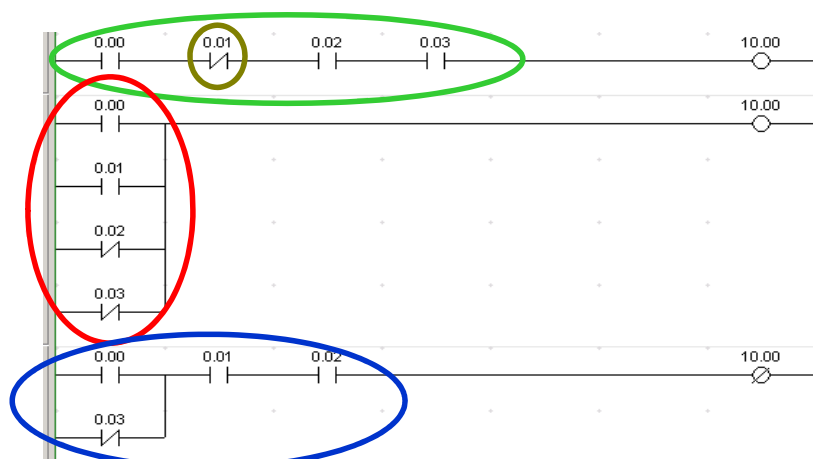


- ⌘ Existen diferentes lenguajes para programar los PLCs de Omron: SCT, Grafcet o Ladder.
- ⌘ En este curso se empleará únicamente el lenguaje de contactos Ladder.
- ⌘ Todas las funciones disponibles en Ladder son extensibles a otros lenguajes de programación de PLCs.
- ⌘ A continuación se detallan las funciones más habituales.



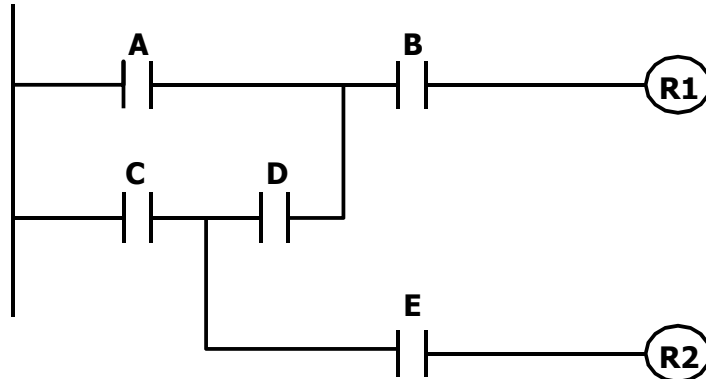
- ❖ Un programa en Ladder / diagrama de relés (L.D.) es una serie de redes o ramas de circuito.
- ❖ Una rama (network) está compuesta de una serie de contactos, conectados en serie o en paralelo, que dan origen a una salida (activación de una bobina o de una función especial).

- ❖ Permite situar:
 - ❖ Relés en serie
 - ❖ En paralelo
 - ❖ Relés negados
 - ❖ Combinaciones de los anteriores



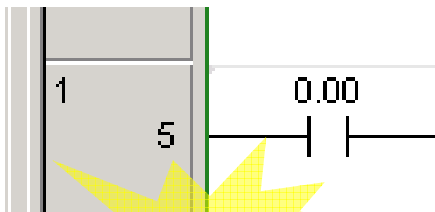


- ❖ Las ramas de circuitos tienen origen en una barra vertical puesta a la izquierda del diagrama.
- ❖ El flujo de la señal va de izquierda a derecha y de arriba abajo.



⌘ Instrucciones básicas:

⊞ LD, OUT, END



Inicia la línea lógica.
Todas las ramas de circuito se inician con una instrucción LOAD.

Transfiere el resultado de las condiciones lógicas que preceden a la instrucción a un bit determinado



Indica el fin de programa

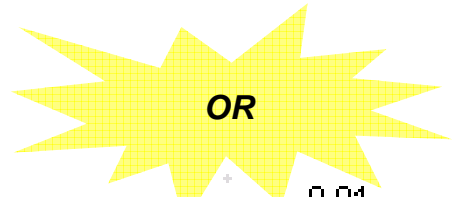


⌘ Instrucciones AND, OR, NOT

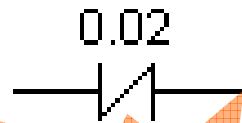
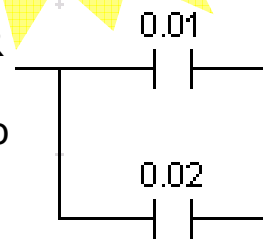


AND

Realiza una AND lógica con un bit determinado



Realiza una OR lógica con un bit determinado

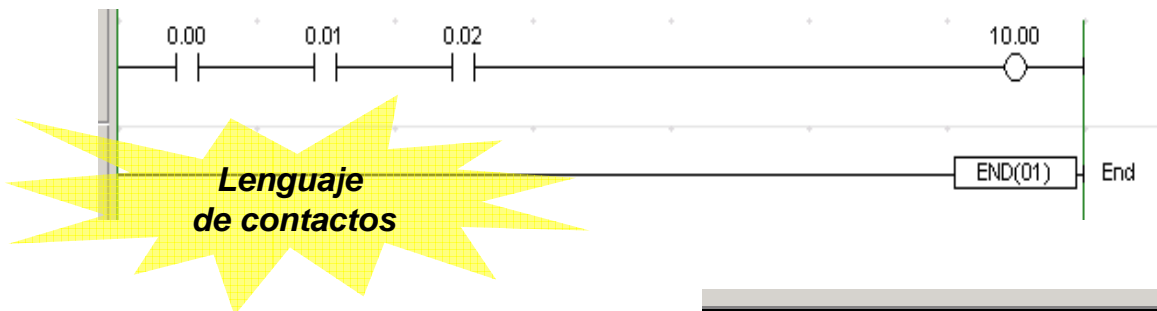


NOT

Niega el estado del bit al cual está asociado



- ❖ A una rama de circuito en ladder, corresponde una secuencia de instrucciones en forma mnemónica

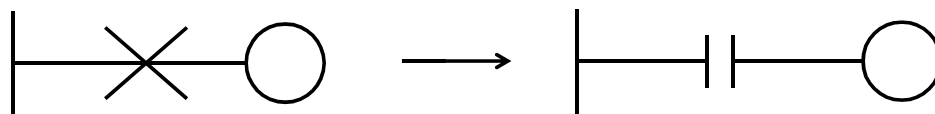


Lista de instrucciones

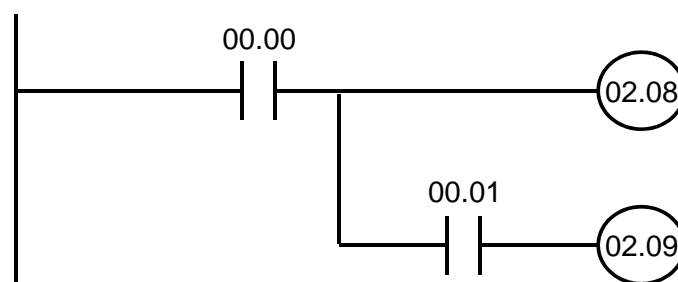
Step	Instruction	Operand
0	LD	0.00
1	AND	0.01
2	AND	0.02
3	OUT	10.00
4	END(01)	



- ❖ Una bobina **no** puede estar conectada directamente a la barra de inicio.
- ❖ A la derecha de una bobina **no** es posible programar ningún contacto.
- ❖ El número de contactos posibles en serie o en paralelo es prácticamente **ilimitado**.

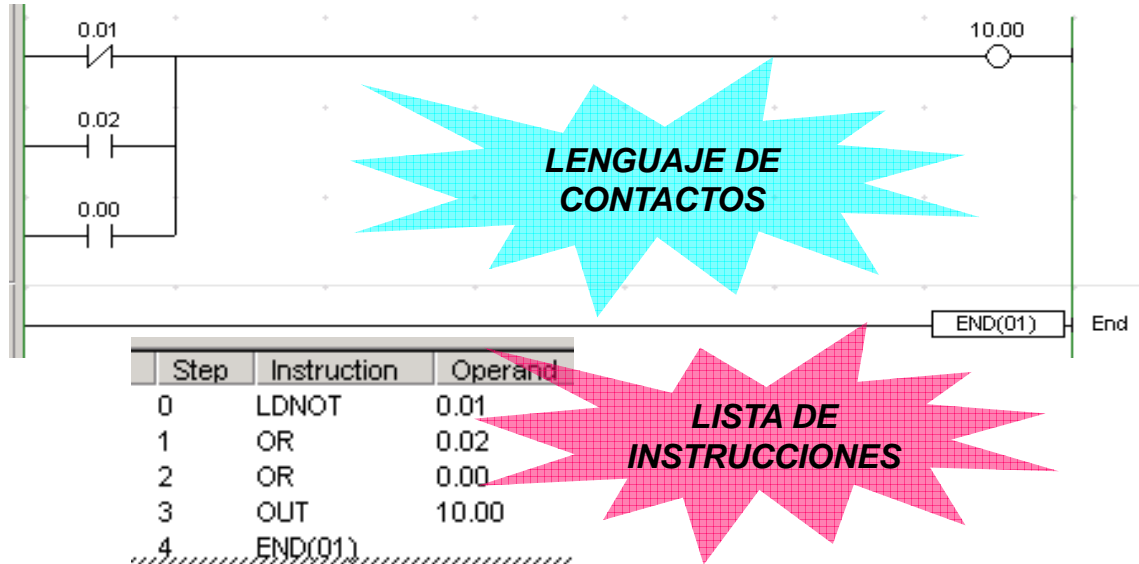


- ❖ Es aconsejable no programar una salida más de una vez.
- ❖ Es posible utilizar el contacto de una salida como entrada auxiliar.
- ❖ Es posible colocar en paralelo 2 o más bobinas.

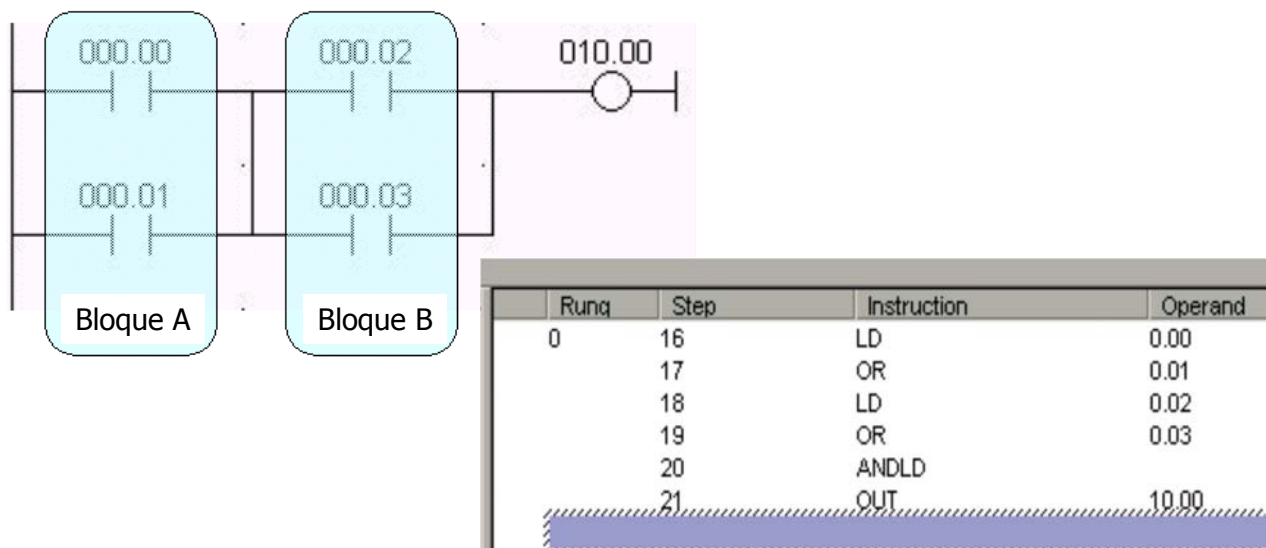




Ejemplo: Se desea implementar un circuito lógico que active la salida 10.00 cuando la entrada 0.01 esté a OFF o cuando las entradas 0.02 o 0.00 estén a ON.

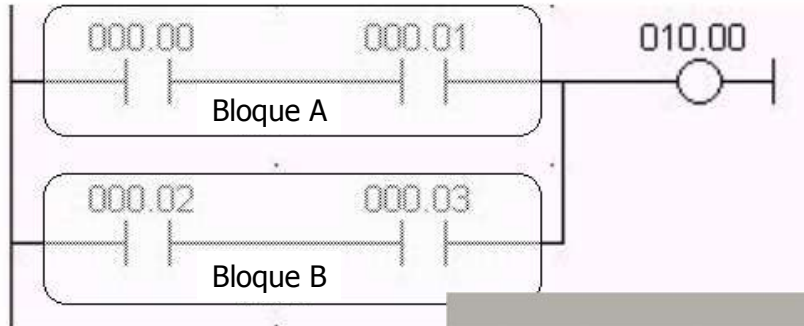


⌘ La instrucción **AND LOAD (ANDLD)** permite colocar en serie dos bloques lógicos, o sea, permite realizar la AND lógica entre dos bloques lógicos.





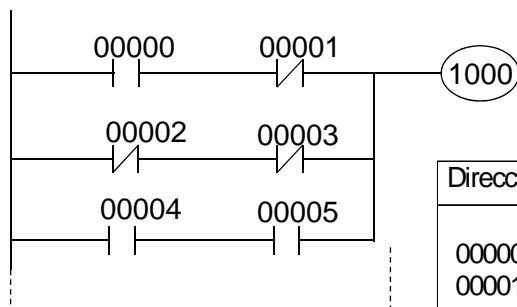
⌘ La instrucción **OR LOAD (ORLD)** permite colocar en paralelo dos bloques, o sea, permite realizar una OR lógica entre dos bloques.



Rung	Step	Instruction	Operand
0	23	LD	0.00
	24	AND	0.01
	25	LD	0.02
	26	AND	0.03
	27	ORLD	
	28	OUT	10.00



❖ Ejemplo de varios bloques en paralelo



1º Método

Dirección	Instrucción	Datos
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	OR LD	-
00005	LD	00004
00006	AND	00005
00007	OR LD	-
...
00012	OUT	1000

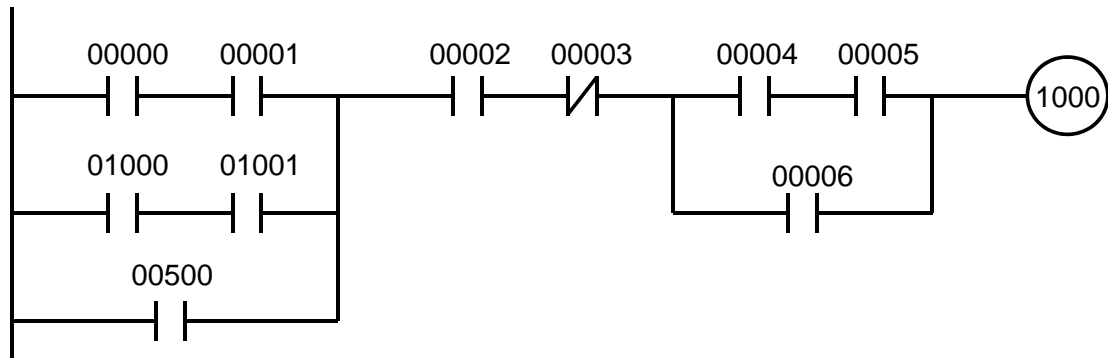
2º Método

Dirección	Instrucción	Datos
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	LD	00004
00005	AND	00005
...
00013	OR LD	-
00014	OR LD	-
...
00019	OUT	1000

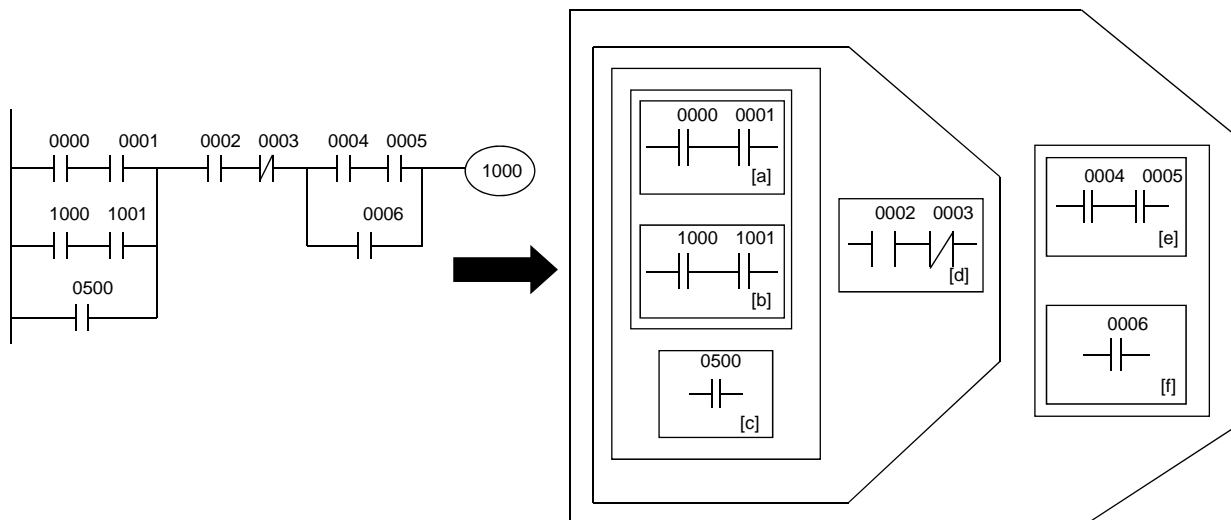
❖ Con el 2º método de codificación, como máximo se pueden utilizar 8 niveles de LD.



- ❖ Es fundamental conocer la correspondencia entre el diagrama de relés y la lista de instrucciones

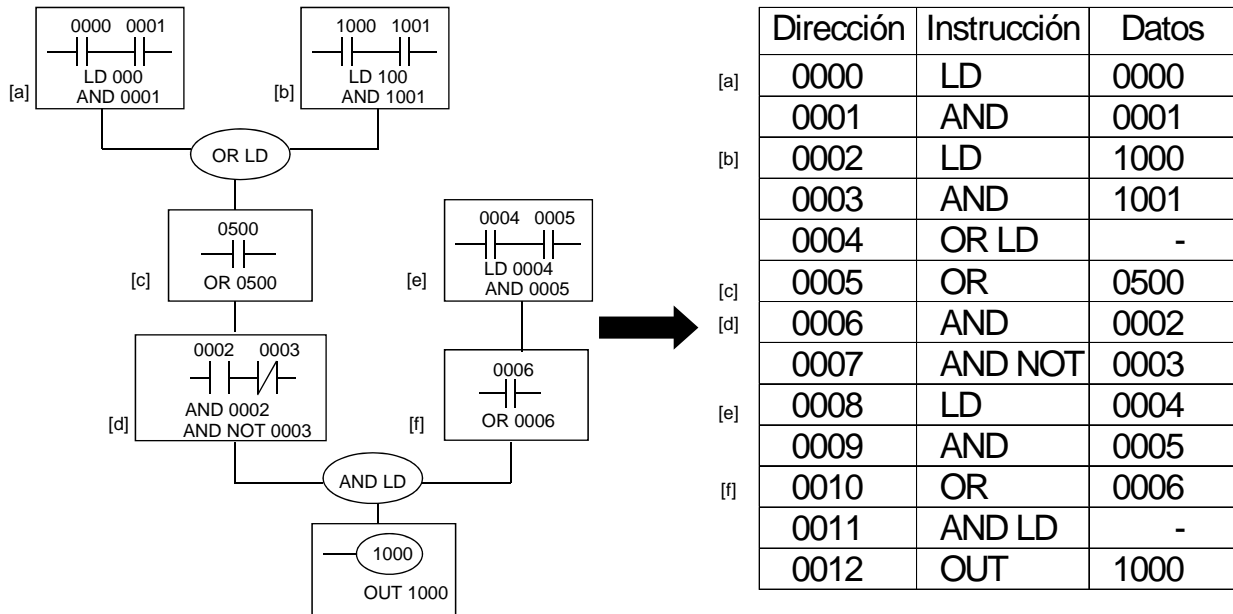


- ❖ Para codificar en lista de instrucciones una red en diagrama de relés, es necesario primero identificar bloques elementales, formando simples circuitos en SERIE o en PARALELO.



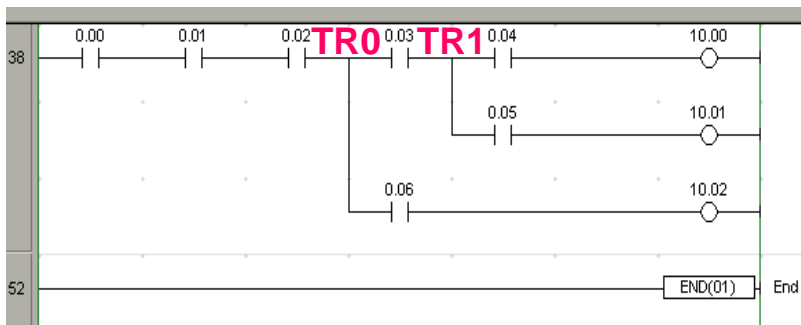


- Una vez realizado se programan los bloques elementales y se unen procediendo de arriba abajo y de izquierda a derecha



Área TR

- Son bits internos que van desde TR0 a TR7.
- Se utilizan en puntos de bifurcación de salidas múltiples, y almacenan (OUT) el resultado en ese punto para poder reutilizarlo después (LOAD).
- Sólo admiten instrucciones LD y OUT.



Step	Instruction	Operand
0	LD	0.00
1	AND	0.01
2	AND	0.02
3	OUT	TR0
4	AND	0.03
5	OUT	TR1
6	AND	0.04
7	OUT	10.00
8	LD	TR1
9	AND	0.05
10	OUT	10.01
11	LD	TR0
12	AND	0.06
13	OUT	10.02
14	END(01)	



❖ Ejercicio 1

- ❖ Realizar el programa en diagrama ladder, correspondiente al siguiente programa en lista de instrucciones:

LD	2960.00
LD	2960.01
AND	2960.02
LD	2960.03
ANDNOT	2960.04
ORLD	
ANDLD	
AND	2960.05
OUT	2961.00
END(001)	



❖ Ejercicio 2

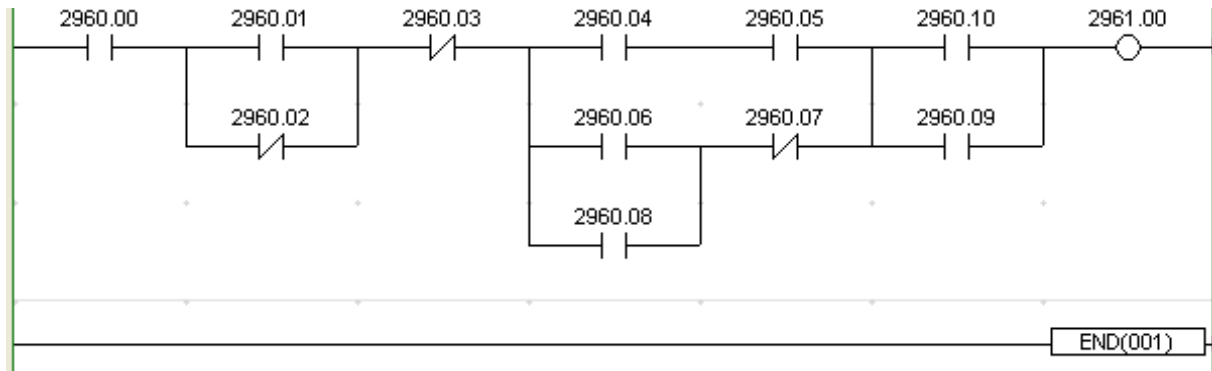
- ❖ Realizar el programa en diagrama ladder, correspondiente al siguiente programa en lista de instrucciones:

LD	2960.00
LD	2960.01
AND	2960.02
LDNOT	2960.07
AND	2960.08
ORLD	
ANDLD	
LD	2960.03
OR	2960.06
ANDLD	
AND	2960.04
OUT	2961.00
AND	2960.05
OUT	2961.01



❖ Ejercicio 3

- ❖ Realizar el programa en lista de instrucciones, correspondiente al siguiente programa en diagrama ladder:



❖ Ejercicio 4

- ❖ Escribir en lista de instrucciones el siguiente programa utilizando bits TR.

