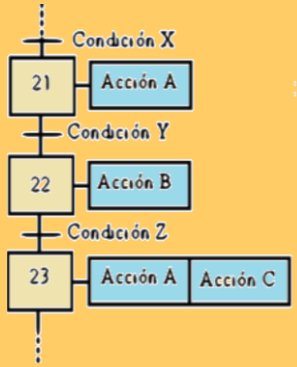
PASOS PARA GENERAR UN GRAFCET

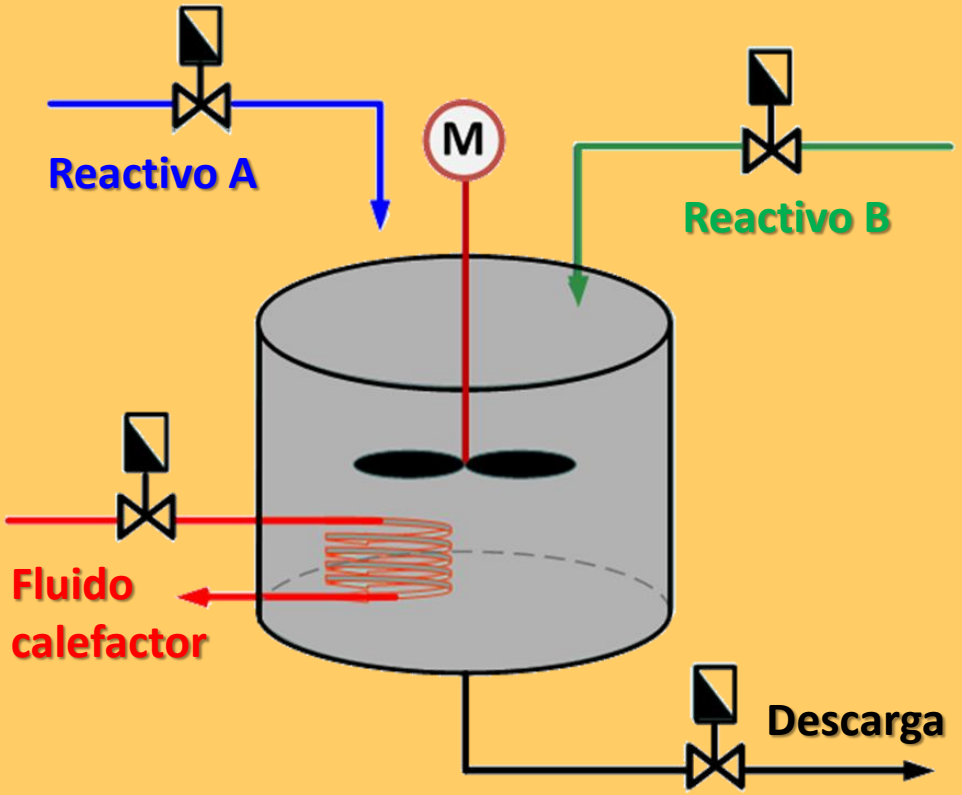
1. **Identificar todas las etapas (estados) del sistema.**
2. **Generar el Grafo de transición de estados (si lo cree conveniente)**
3. **Definir todas las salidas (Outputs) que actuarán sobre el sistema.**
4. **Definir las transiciones entre cada uno de los estados**
5. **Definir las acciones de cada etapa**



GRAFCET – ESTUDIO DE UN CASO

Se quiere automatizar la operación de un reactor batch para que siga una secuencia en forma automática. Iniciar la actividad mediante un pulsador de Inicio. Primero se debe alimentar el reactivo A hasta la parte media. Luego alimentar el reactivo B hasta completar la capacidad del tanque.

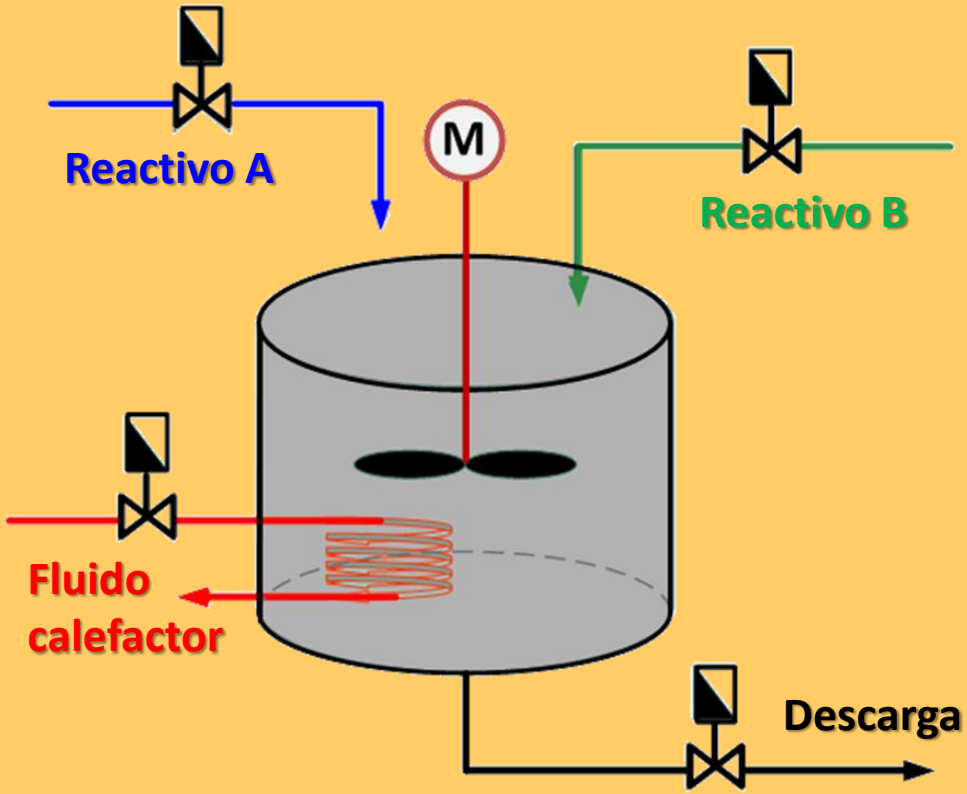
Una vez lleno el reactor activar la agitación y 1 minuto después abrir la válvula de fluido calefactor hasta que la temperatura de la masa reactiva alcance 60 °C. La reacción debe mantenerse por el término de 1 hora. Concluido este proceso, se detiene la agitación y se descarga el tanque



GRAFCET – ESTUDIO DE UN CASO

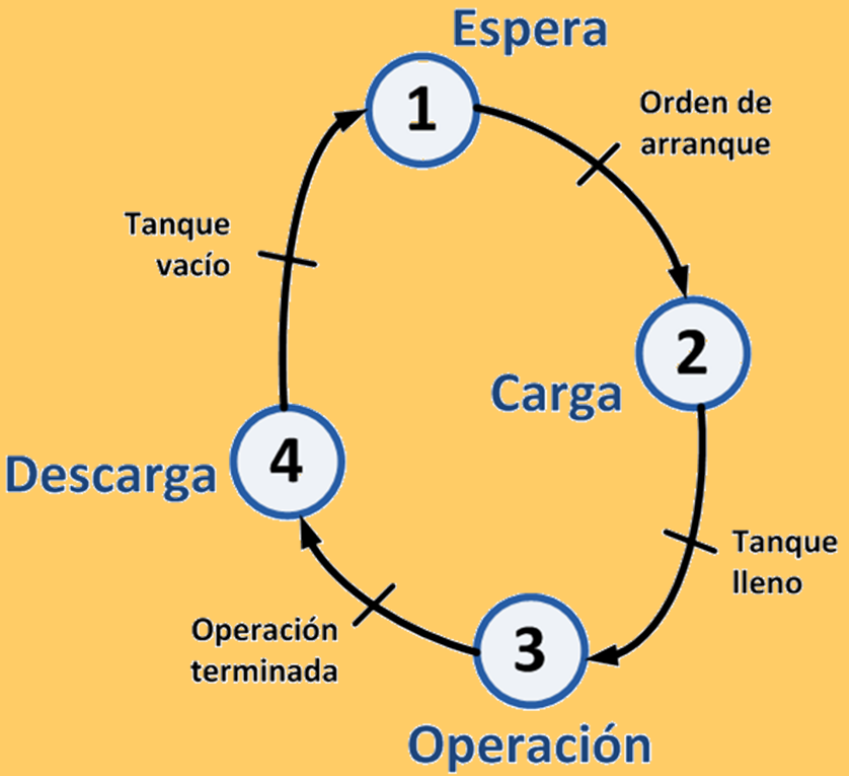
Identificación de las etapas

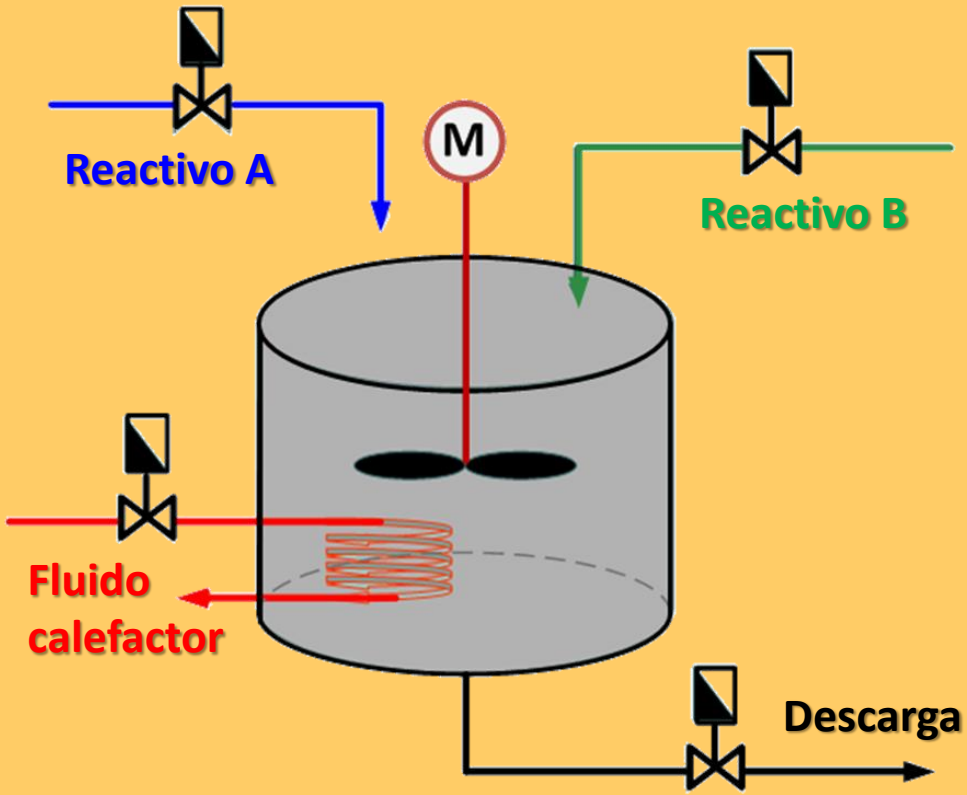
N°	ETAPA
1	**ESPERA** Inicio desde reposo
2	**CARGA** Alimentación de A y de B
3	**OPERACIÓN** Agitación y calefacción por un lapso determinado
4	**DESCARGA**



GRAFCET – ESTUDIO DE UN CASO

Grafo de transición de estados

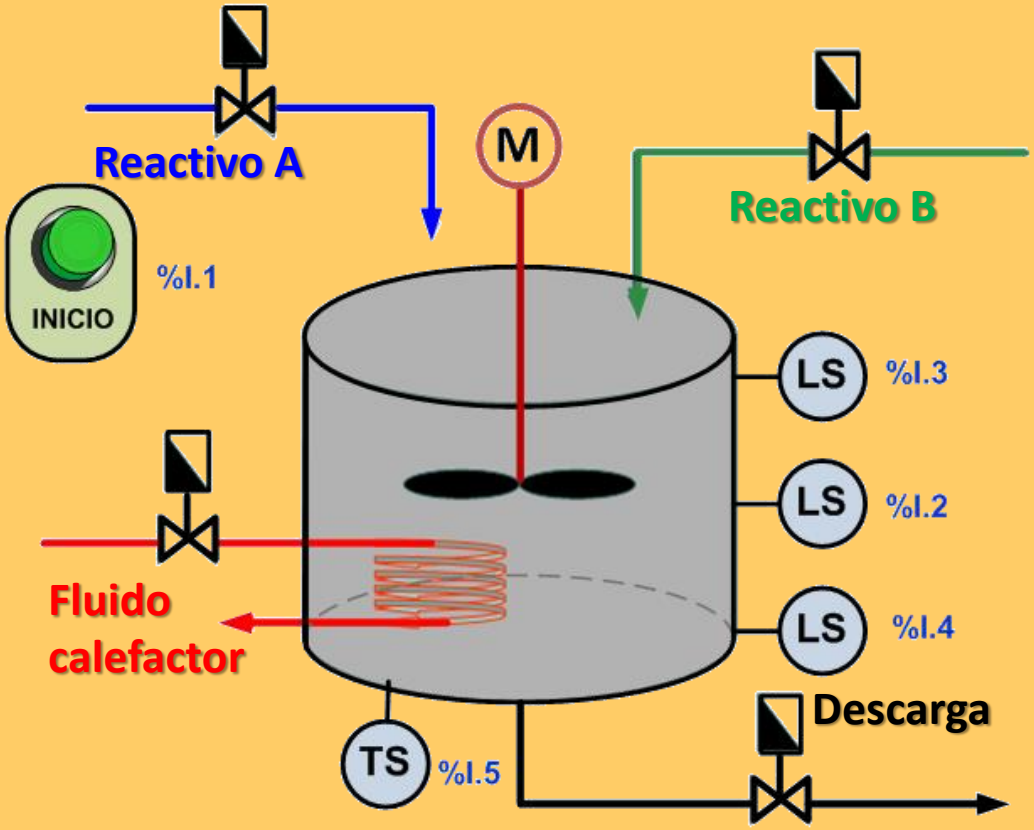




GRAFCET – ESTUDIO DE UN CASO

Inputs del sistema

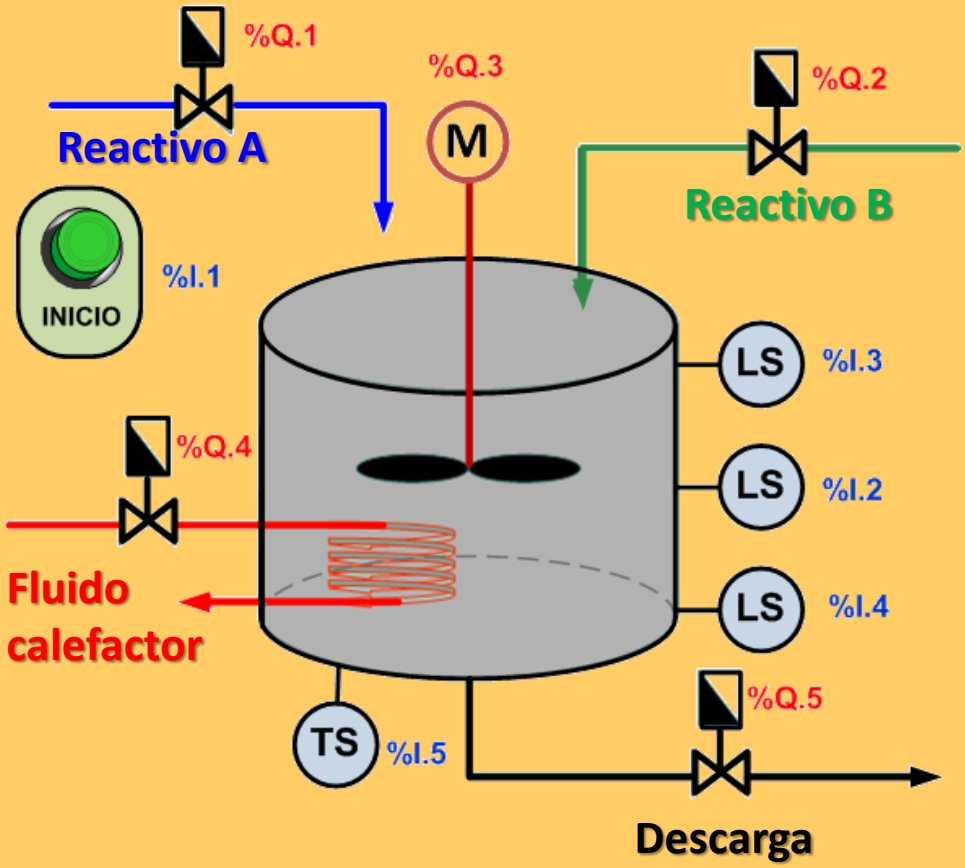
Id	INPUT
%I.1	Pulsador para Inicio de la operación
%I.2	Detector de nivel medio del tanque.
%I.3	Detector de nivel máximo (tanque lleno)
%I.4	Detector de nivel mínimo (tanque vacío).
%I.5	Detector de temperatura máxima (60 °C).



GRAFCET – ESTUDIO DE UN CASO

Outputs del sistema

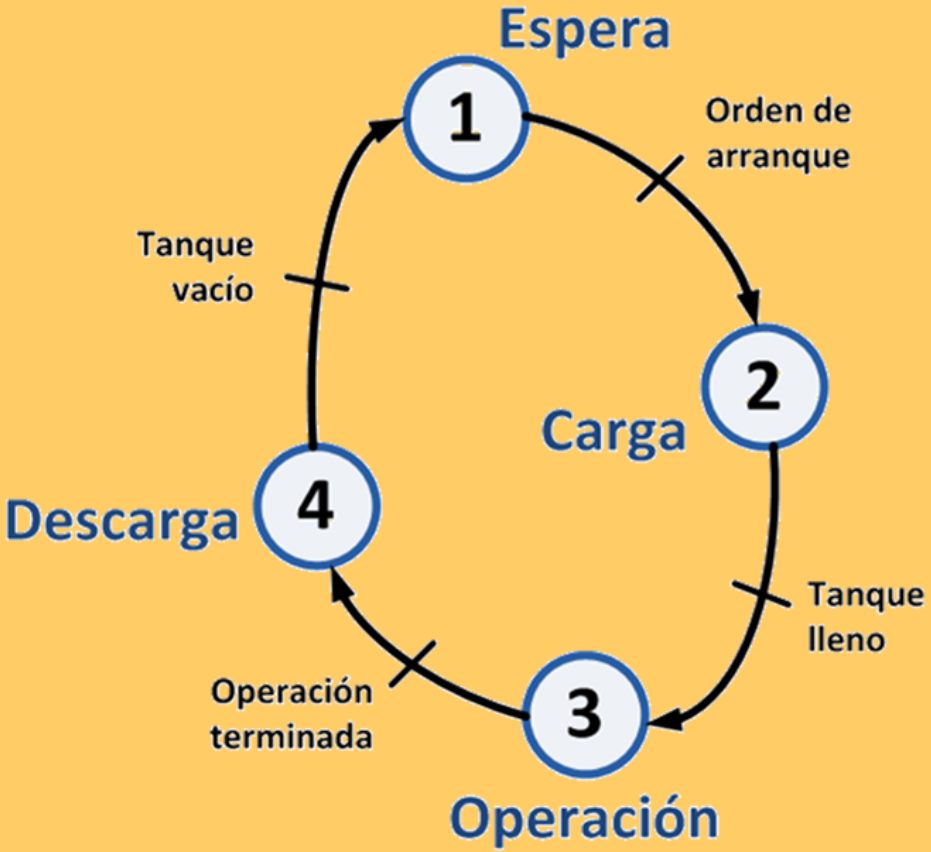
Id	OUTPUT
%Q.1	Actuador de la válvula de alimentación de A
%Q.2	Actuador de la válvula de alimentación de B
%Q.3	Motor del agitador
%Q.4	Actuador de la válvula de fluido calefactor
%Q.5	Actuador de la válvula de descarga



GRAFCET – ESTUDIO DE UN CASO

Transiciones

Id	TRANSICIÓN
1 - 2	**ORDEN DE ARRANQUE** Pulsar arranque. %I.1 ↑
2 - 3	**TANQUE LLENO** Detector %I.3 en ON
3 - 4	**OPERACIÓN TERMINADA** Tiempo de operación igual a 3600 segundos
4 - 1	**TANQUE VACÍO** Detector %I.4 en OFF



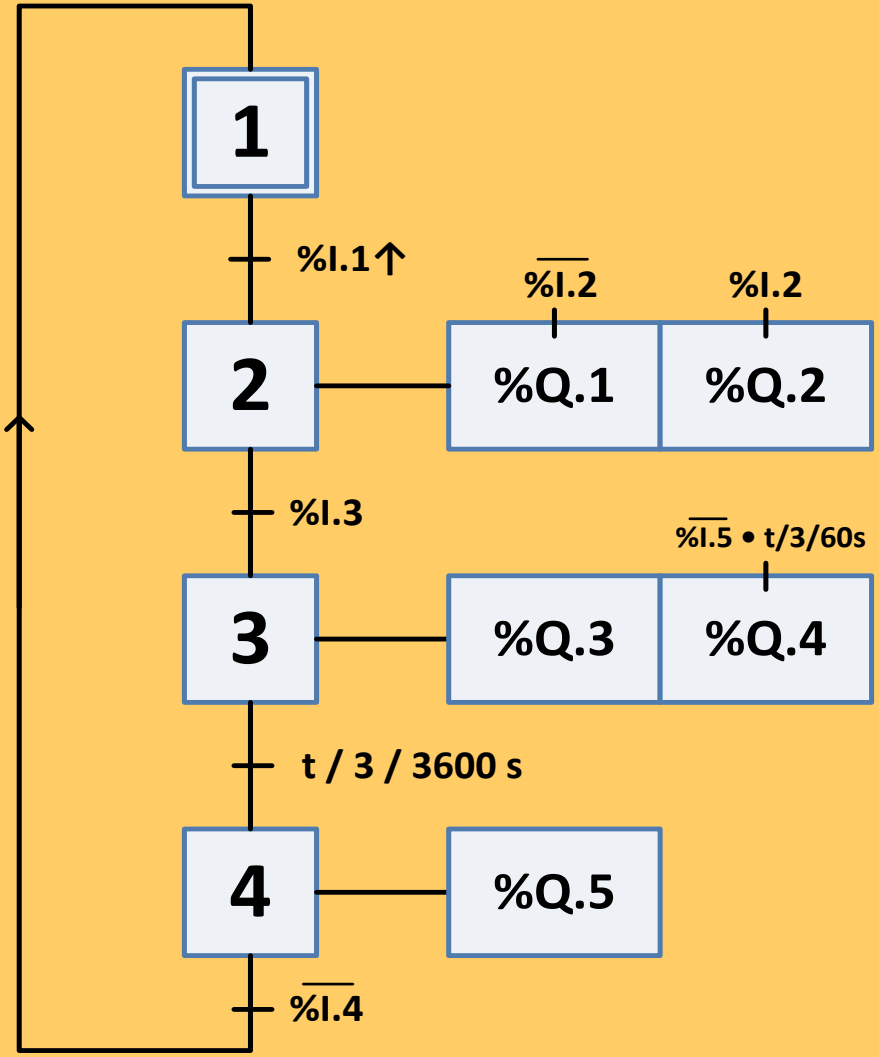
GRAFCET – ESTUDIO DE UN CASO

Acciones

Id	ETAPA	ACCIONES
1	ESPERA	Ninguna
2	CARGA	1. Alimentación de A (%Q.1 en ON) hasta alcanzar el nivel medio (Detector %I.2 se ponga en ON). 2. Alimentar B (%Q.2 en ON) cuando haya terminado la alimentación de A y hasta que se llene el tanque (Detector %I.3 se ponga en ON).
3	OPERACIÓN	1. Agitar poniendo en ON el motor %Q.3 2. Alimentación de fluido calefactor (%Q.4 activado) 60 s después de iniciar la agitación y hasta alcanzar los 60 °C (Señal del switch de temperatura %I.5 en ON).
4	DESCARGA	Activar el actuador de la válvula de descarga (%Q.5)

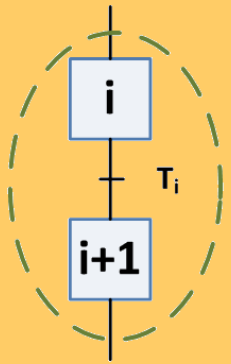
GRAFCET

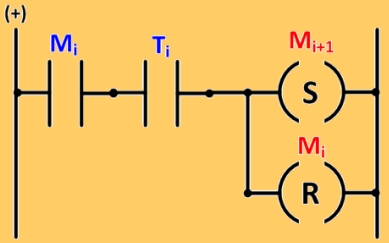
ESTUDIO DE UN CASO



EQUIVALENCIA GRAFCET - LADDER

Se vio que una secuencia puede programarse en lenguaje de contactos a partir de las ecuaciones lógicas que relacionan entradas y salidas. Otra forma de programar consiste en **"traducir"** el GRAFCET de una secuencia a Ladder. Se aprovecha la **comprensión visual** del ciclo de trabajo que ofrece el **GRAFCET** con la potencia de programación del Ladder.





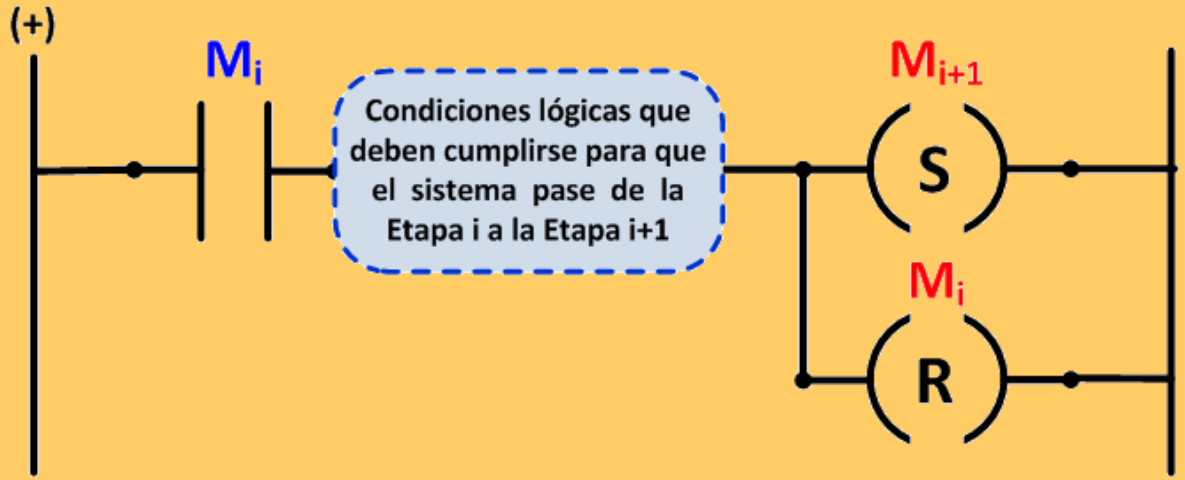
La traducción de GRAFCET a Ladder se consideran dos aspectos:

- **Control de la Secuencia** de las etapas
- **Acciones** que deben realizarse **en cada etapa** cuando se hallen activas

EQUIVALENCIA GRAFCET – LADDER

Control de la secuencia

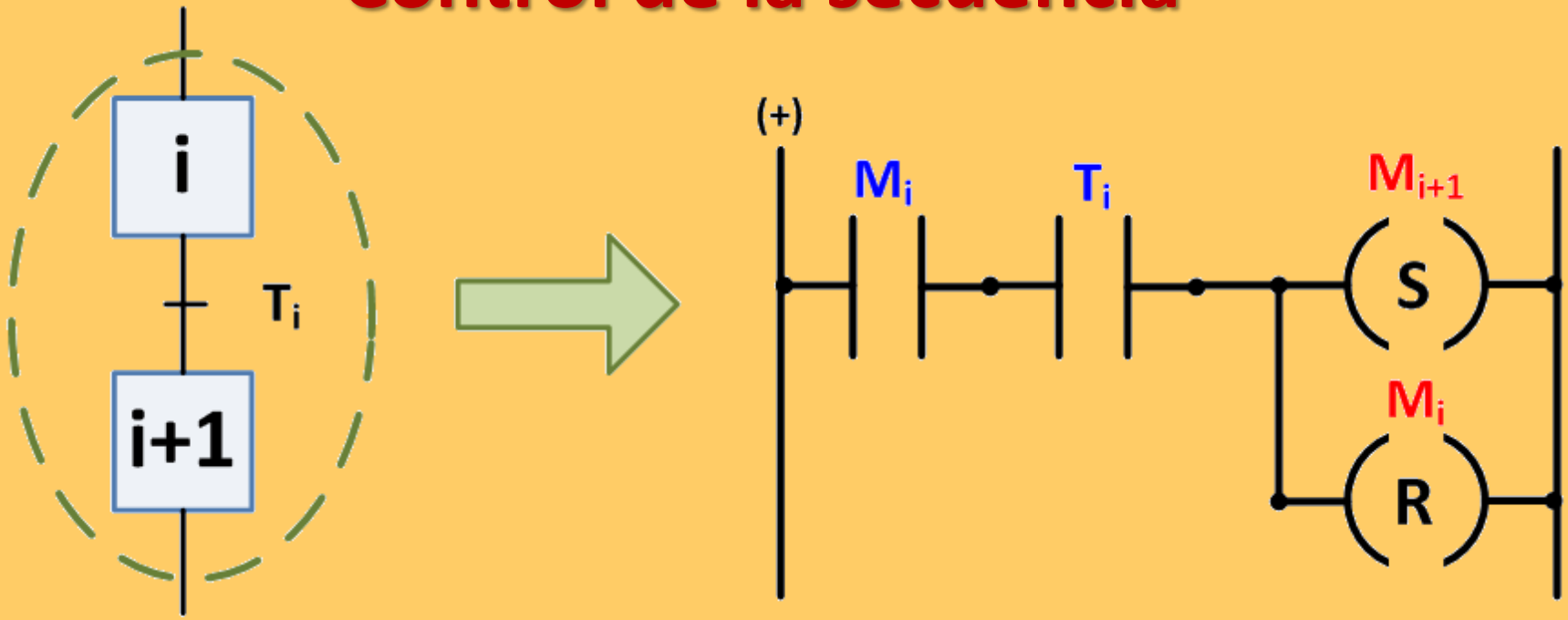
Hay que asignar en cada etapa una marca interna (bit interno) que estará en "1" el tiempo que se encuentre activa la etapa asociada. Para esto se usa bobinas SET y RESET. Esta parte de Ladder controla la evolución de la secuencia del proceso, etapa por etapa, transición por transición.



Estando activa la etapa i, y si se cumple la transición, se desactiva la etapa i (RESET) y de activa la nueva (SET).

EQUIVALENCIA GRAFCET – LADDER

Control de la secuencia



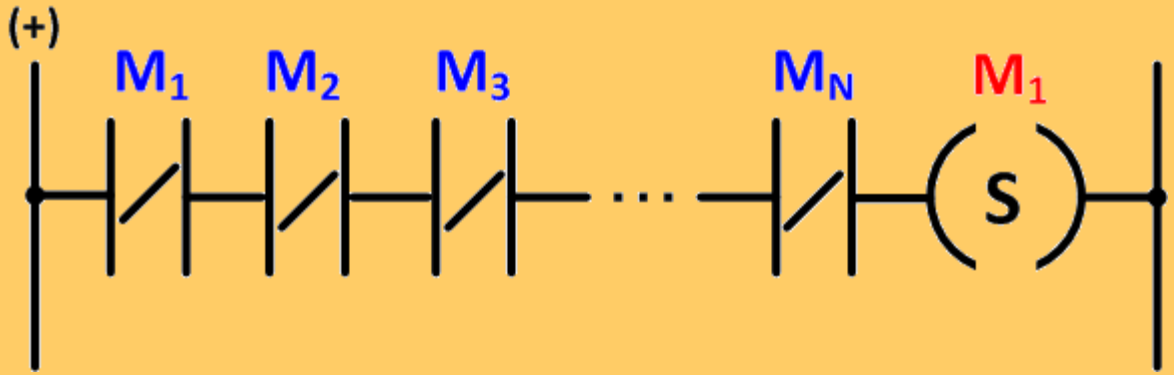
Estando activa la etapa i ($M_i = 1$) y si se cumple la transición ($T_i = 1$) , se desactiva la etapa i (RESET $\rightarrow M_i = 0$) y de activa la nueva (SET $\rightarrow M_{i+1} = 1$)

EQUIVALENCIA GRAFCET – LADDER

Control de la secuencia

¿Qué sucede cuando se inicia el control?

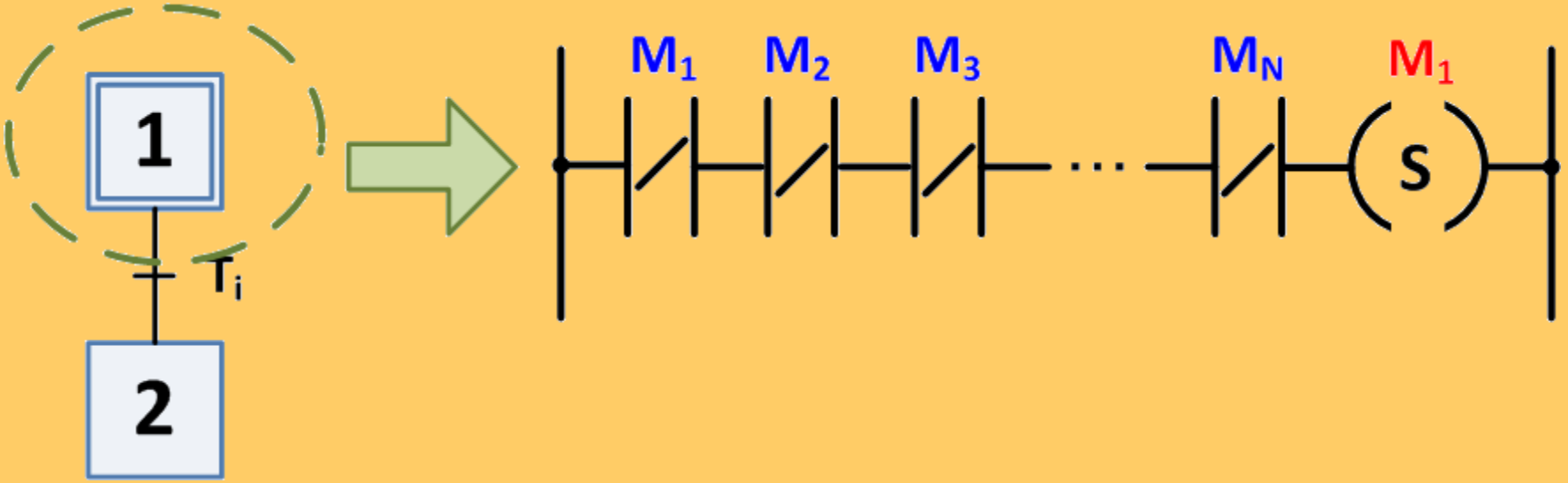
Ninguna etapa está activa, todos los bits internos asociados a etapas valen cero y por lo tanto ninguna acción se ejecuta. Por esa razón debe existir una primera línea de programación que indique cual es la primera etapa a ejecutar.



En el inicio, ninguna etapa está activada y por esta única vez, se activa la primera de las etapas (SET → M_1 = 1)

EQUIVALENCIA GRAFCET – LADDER

Control de la secuencia

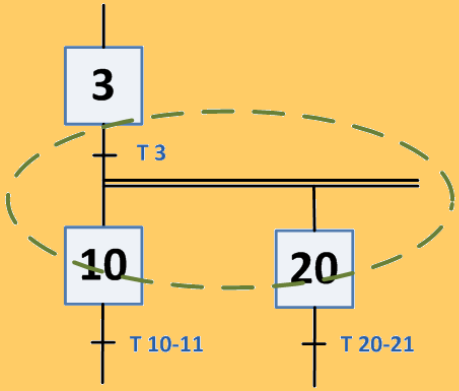


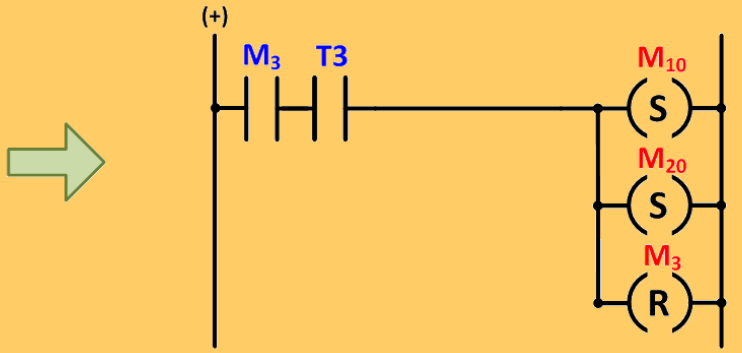
En el inicio, ninguna etapa está activada y por esta única vez, se activa la primera de las etapas (SET → M_1 = 1)

EQUIVALENCIA GRAFCET – LADDER

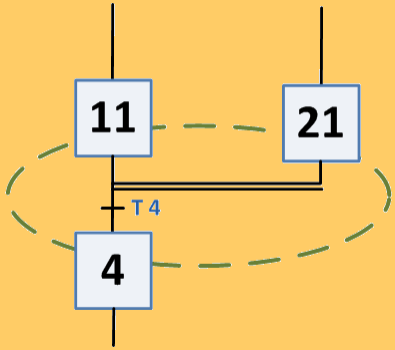
Control de la secuencia

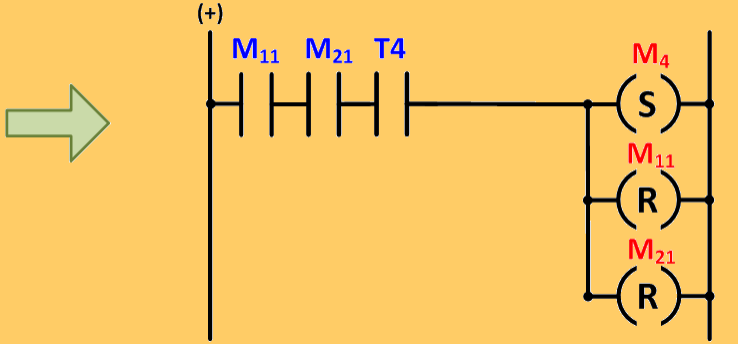
Secuencias simultáneas Divergencia AND





Secuencias simultáneas Convergencia AND

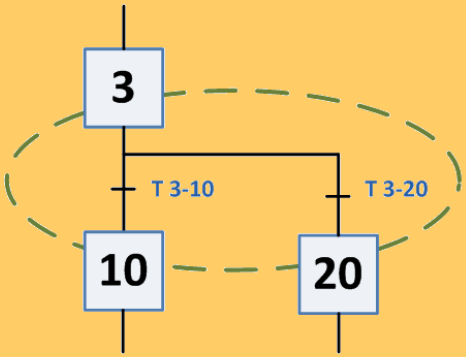


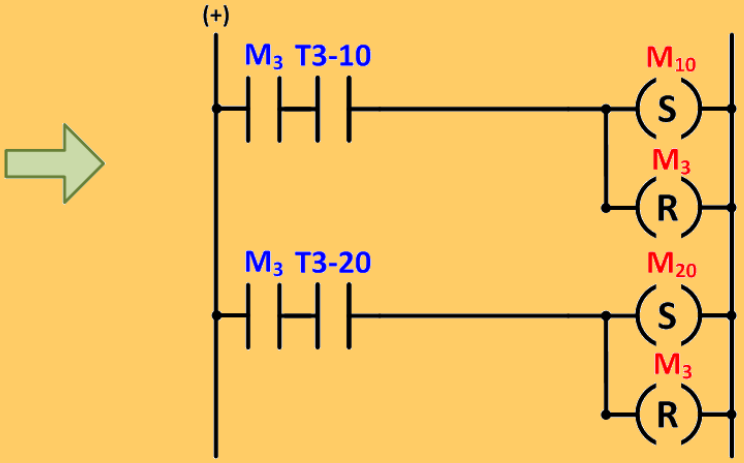


EQUIVALENCIA GRAFCET – LADDER

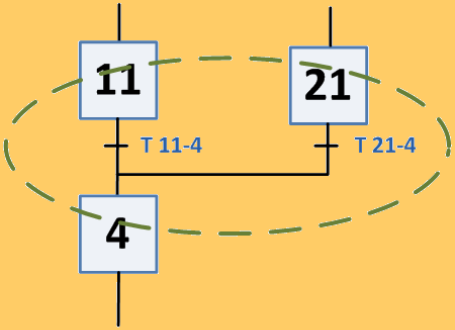
Control de la secuencia

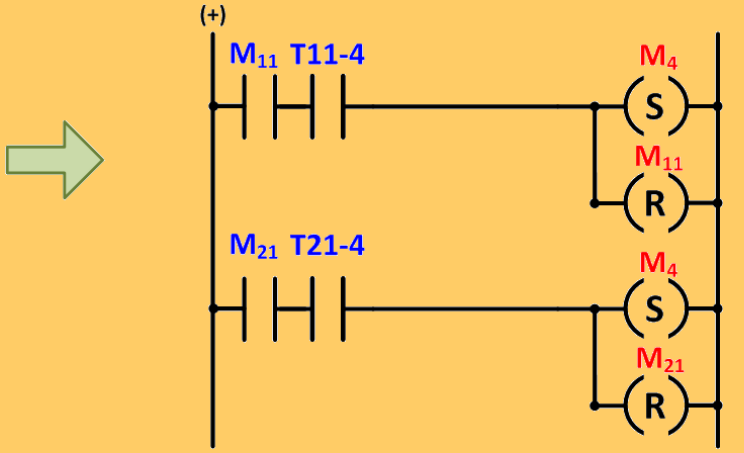
Secuencias alternativas Divergencia OR





Secuencias alternativas Convergencia OR

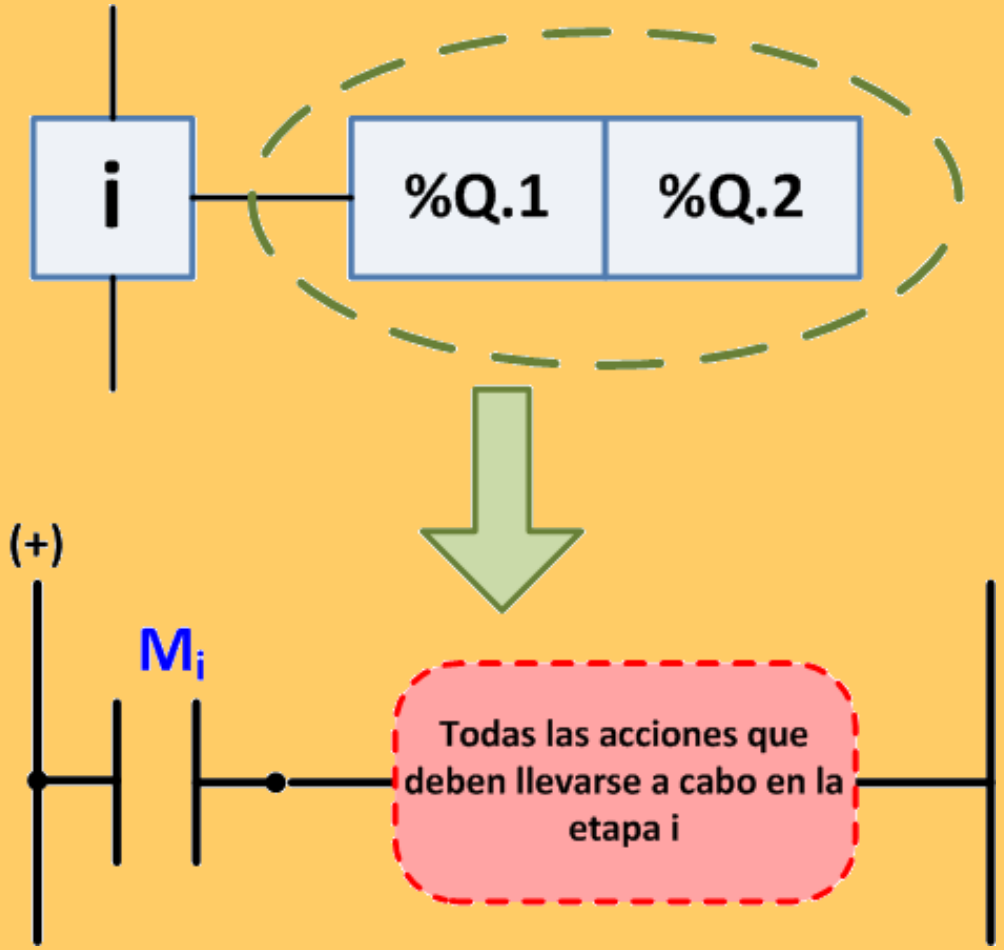




EQUIVALENCIA GRAFCET – LADDER

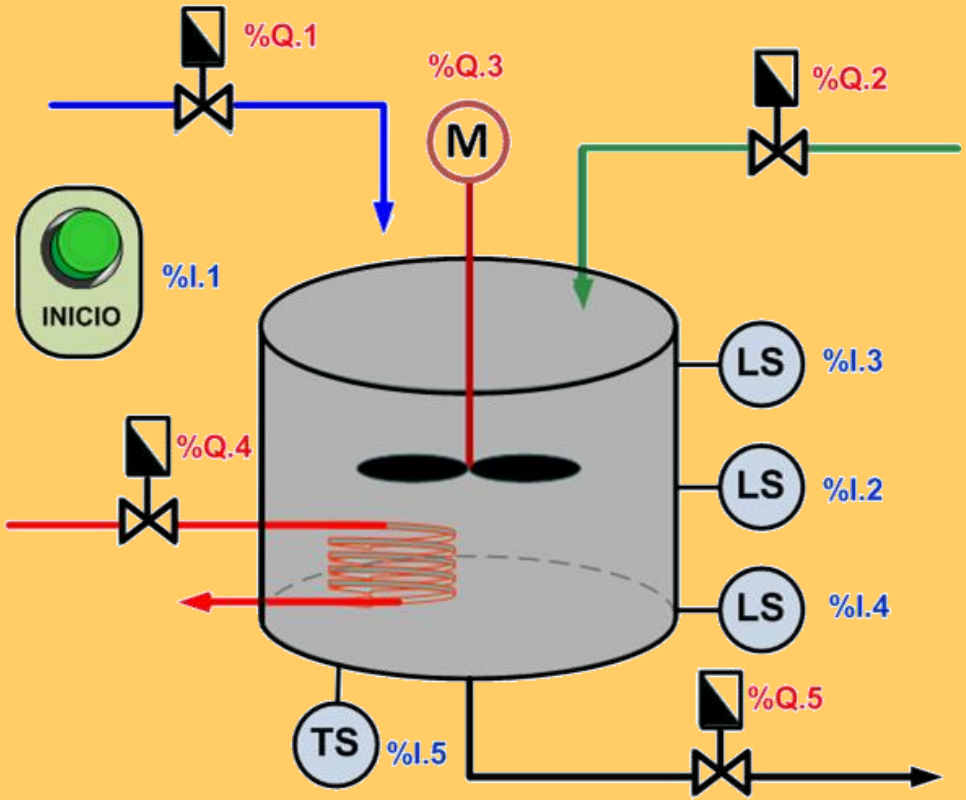
Acciones en cada etapa

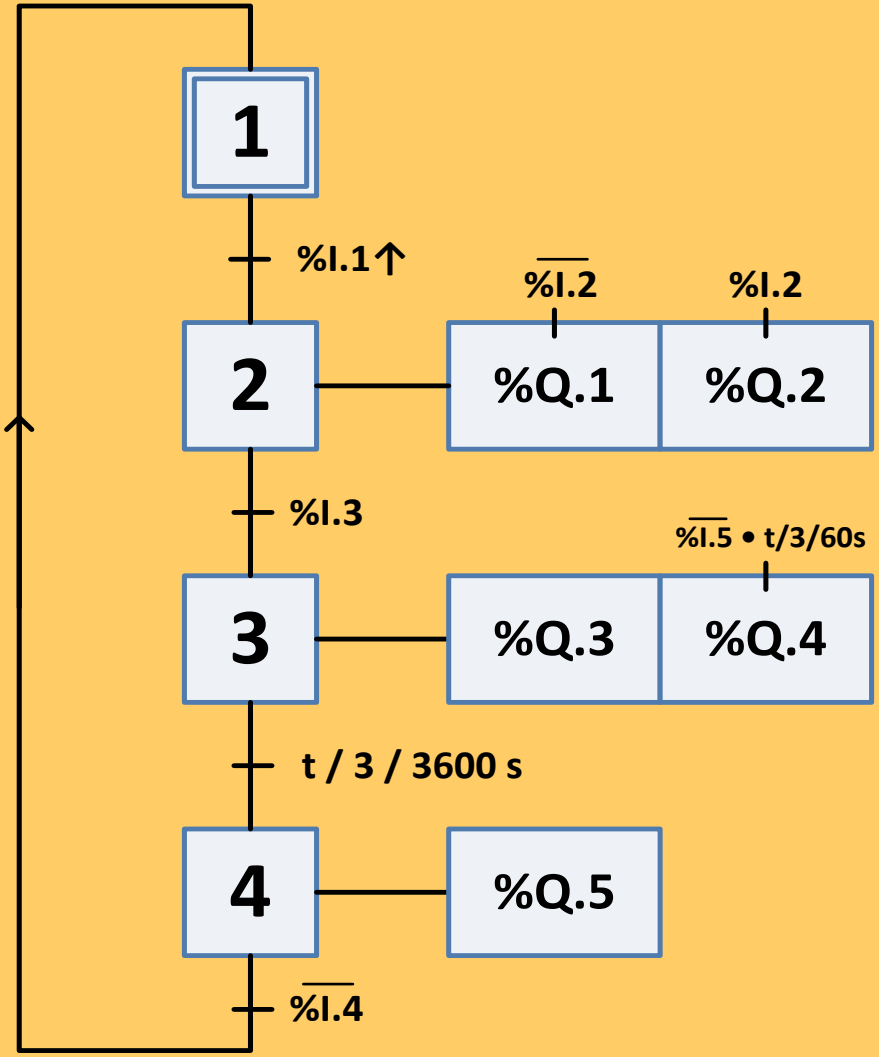
El programa en lenguaje Ladder deberá contemplar las acciones asociadas a cada etapa, considerando las marcas internas activas que corresponden a cada una.



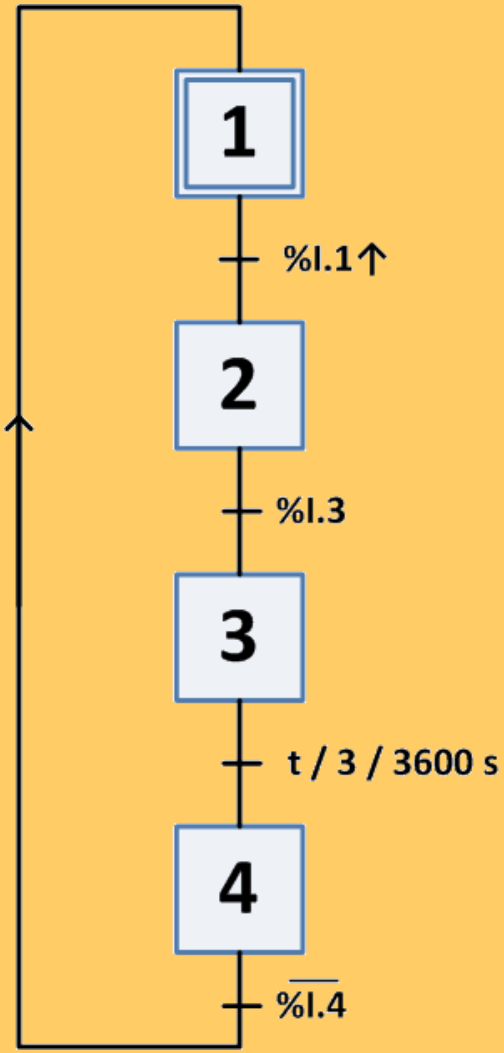
EQUIVALENCIA GRAFCET – LADDER

Estudio de un caso

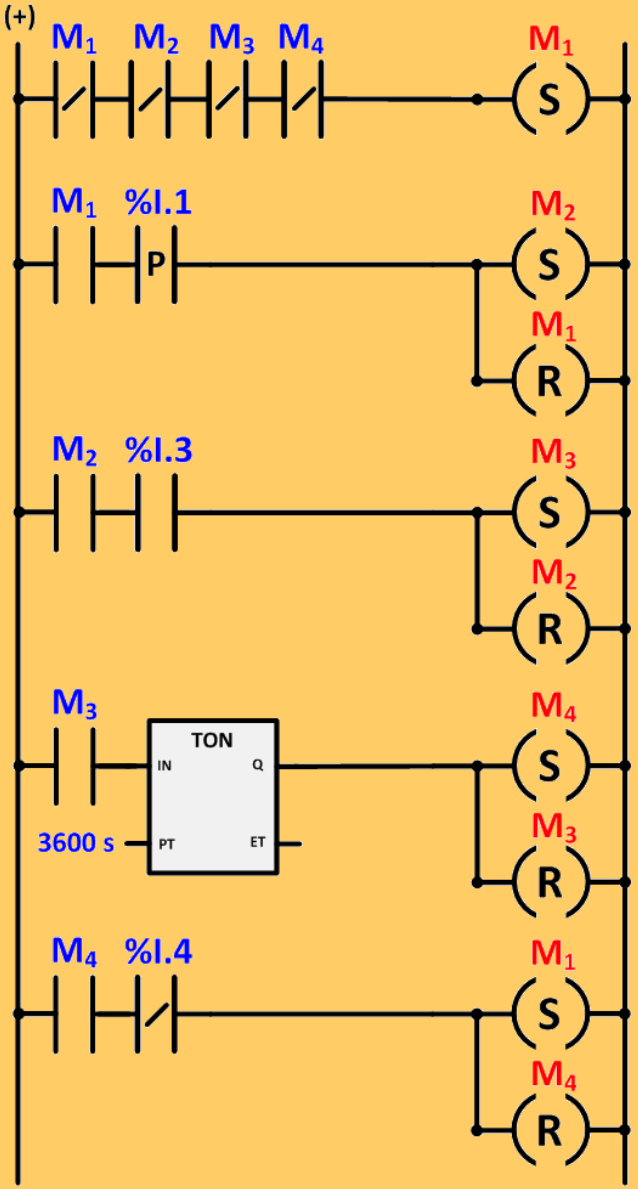




GRAFCET – LADDER

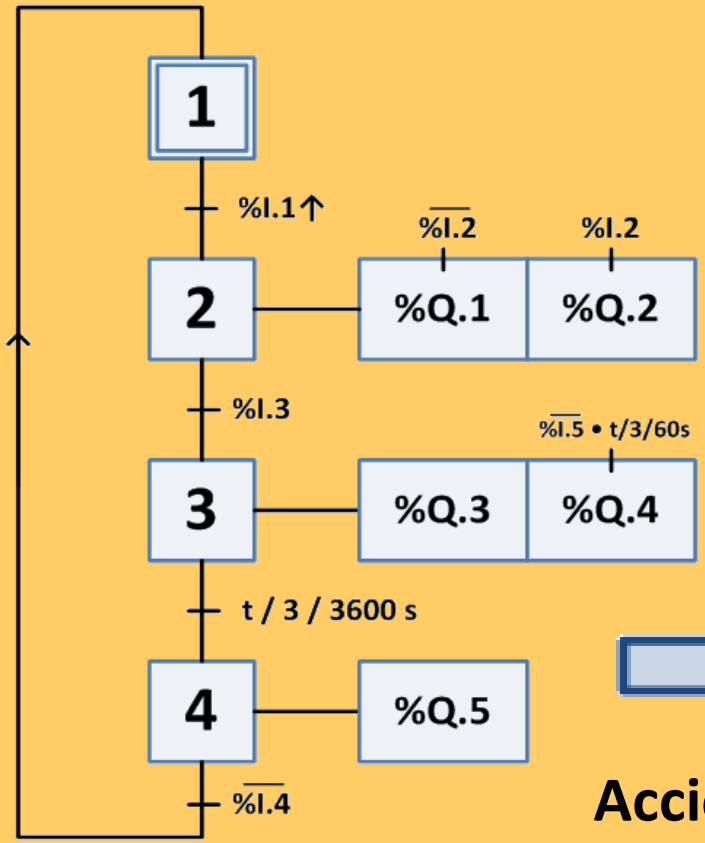


Control de la Secuencia de etapas

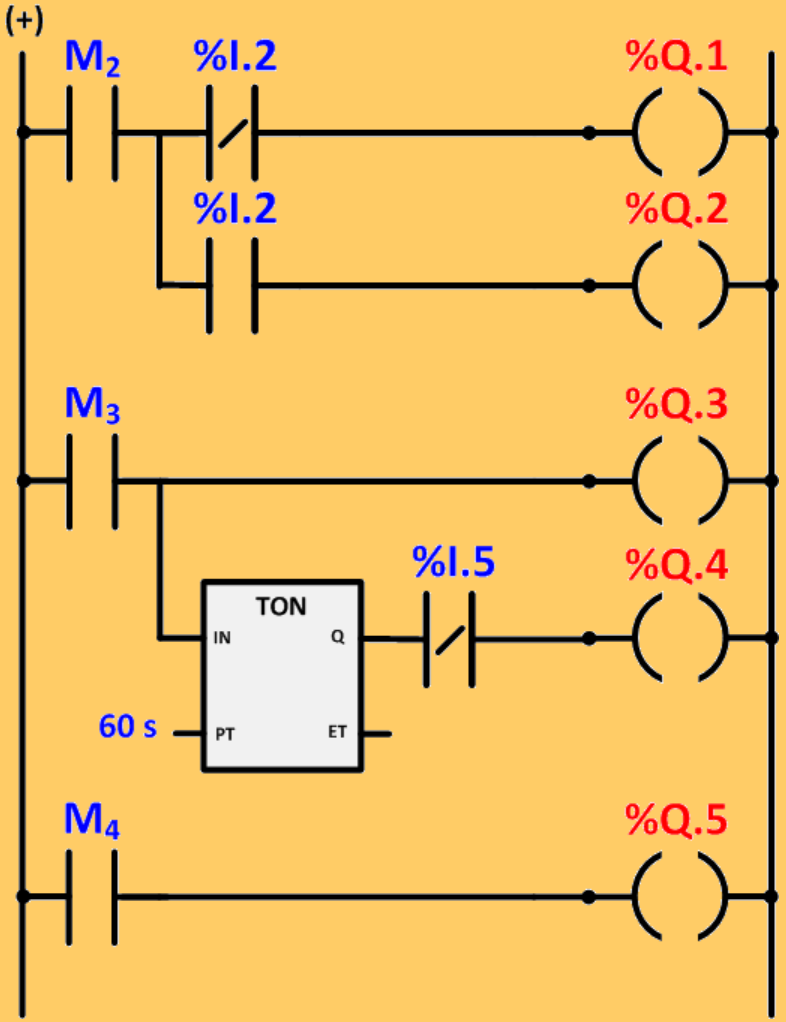


GRAFCET A LADDER

Estudio de un caso



Acciones de cada etapa



PLC

MÓDULO DE ENTRADAS
Actualización de todas las entradas

CPU

Temporiza-dores

Contadores

Marcas internas

Ejecución del programa

Ciclo de trabajo del PLC

MÓDULO DE SALIDAS
Actualización de todas las salidas

MANTENIMIENTO INTERNO

Señales de Sensores

%I.1

%I.2

%I.3

%I.4

%I.5

Señales a Actuadores

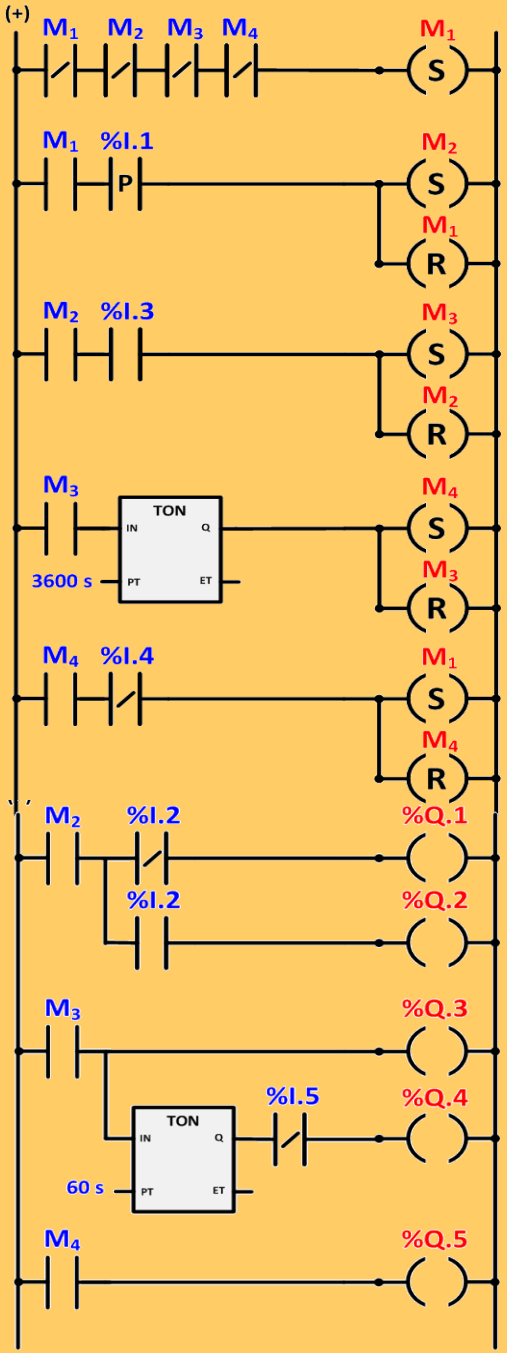
%Q.1

%Q.2

%Q.3

%Q.4

%Q.5



PROGRAMACIÓN EN LADDER
Caso de tener dos estados

Se dijo que en el desarrollo de un programa en lenguaje Ladder se desarrollan en dos partes:

- **Control de la Secuencia**
- **Desarrollo de las Acciones de cada etapa**

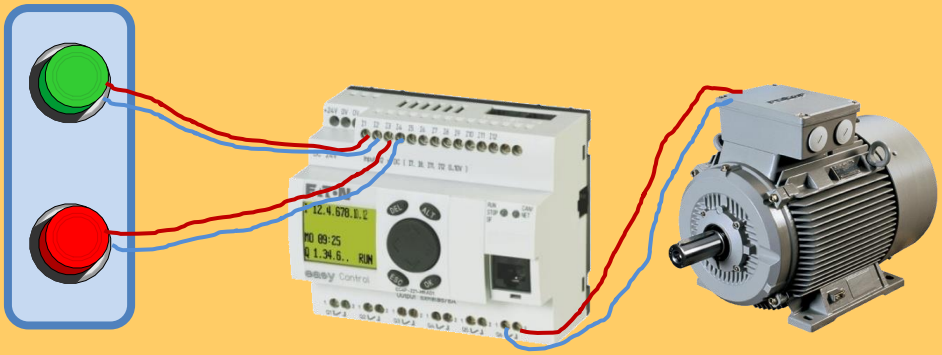
En este caso especial, como son solo dos los estados, **se puede usar sola marca** (bit interno) para definir los estados:

M = 0 ⇒ Sistema en un determinado estado

M = 1 ⇒ Sistema en el otro estado

PROGRAMACIÓN EN LADDER (dos estados)

EJEMPLO. Sistema de ARRANQUE–PARADA de motores por medio de pulsadores.

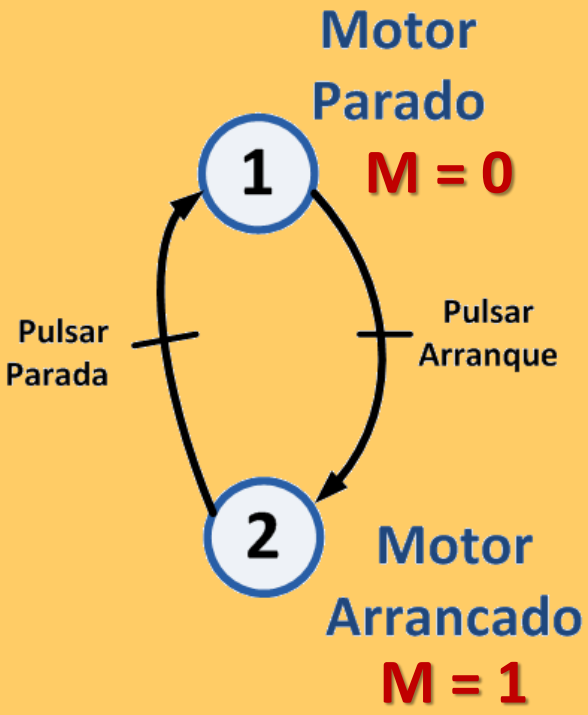


ENTRADAS

- Señal del pulsador de arranque (A)
- Señal del pulsador de parada (P)

SALIDA

- Relé de arranque o parada del motor (Q)

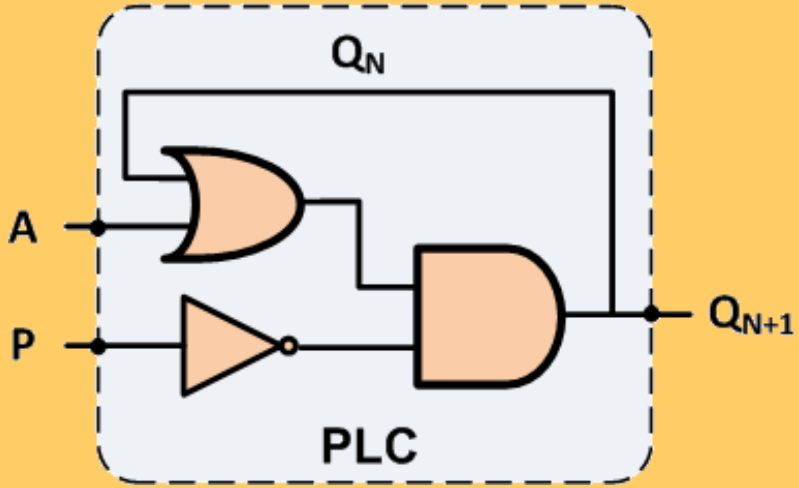


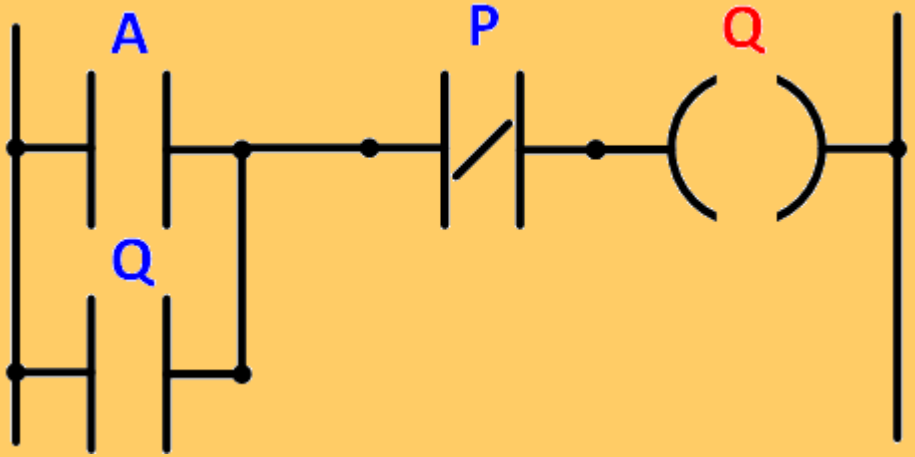
PROGRAMACIÓN EN LADDER (dos estados)

El análisis usando la Tabla de la verdad condujo a la solución:

$$Q_{N+1} = (A + Q_N) \bullet \overline{P}$$

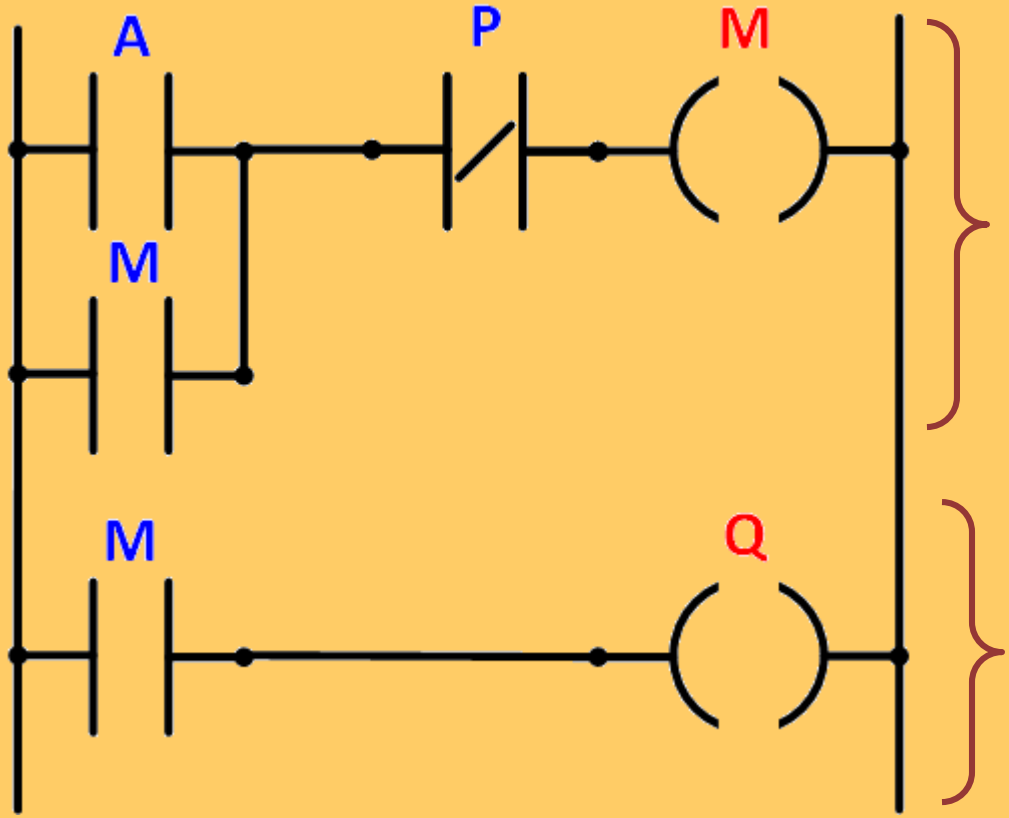
Que al traducir en lenguaje Ladder resultó:





PROGRAMACIÓN EN LADDER (dos estados)

Una alternativa, que es consistente con el desarrollo del Ladder para múltiples estados, resulta:



En esta línea de programa se determina la etapa (estado) del sistema (valor de M)

En esta línea se ejecutan las acciones vinculadas al estado activo

PROGRAMACIÓN EN LADDER (dos estados)

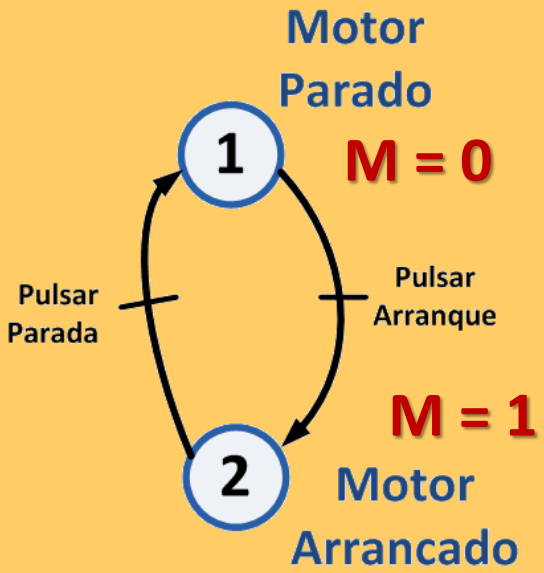
EJEMPLO. Sistema de ARRANQUE–PARADA de motores por medio de dos pulsadores. Además, si el motor está apagado, se debe encender una luz roja y si está funcionando, debe prenderse una luz verde.

ENTRADAS

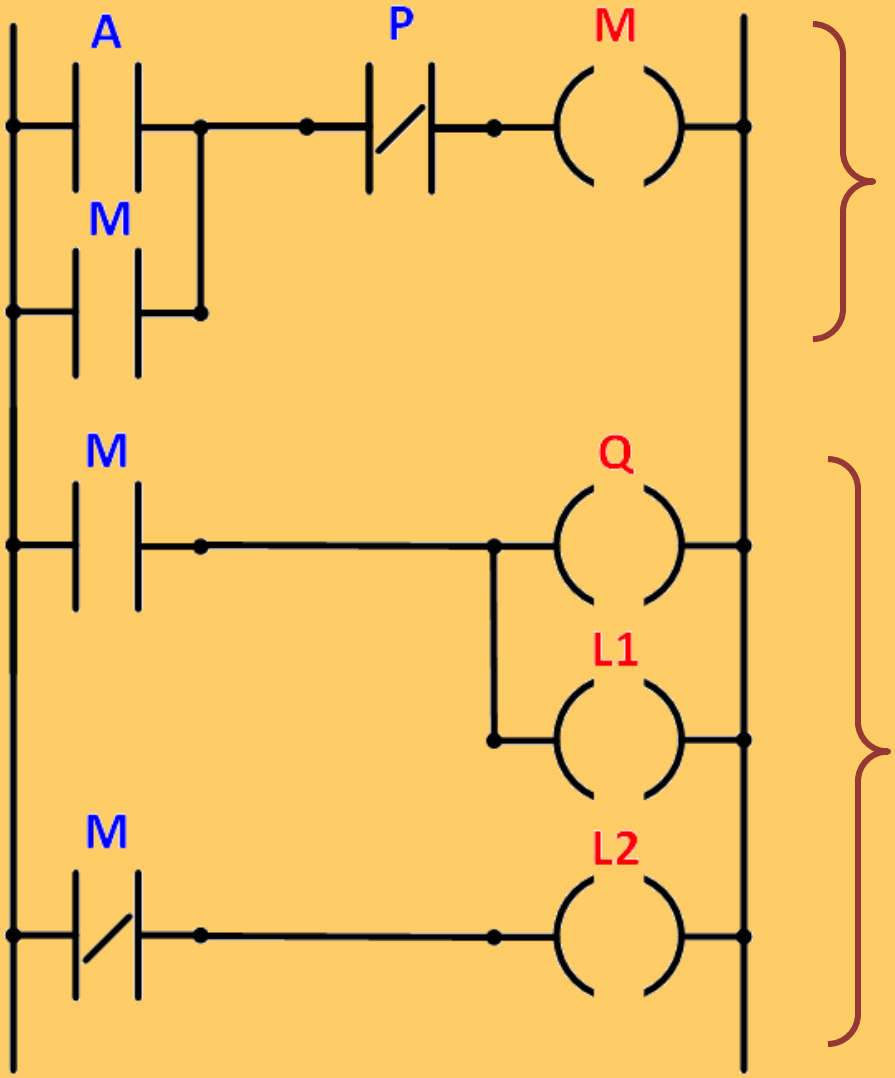
- Señal del pulsador de arranque (A)
- Señal del pulsador de parada (P)

SALIDAS

- Relé de arranque o parada del motor (Q)
- Luz verde de motor arrancado (L1)
- Luz roja de paro (L2)



PROGRAMACIÓN EN LADDER (dos estados)



Sección de programa en el que se determina el estado del sistema (valor de M)

Sección del programa en el que se ejecutan las acciones que corresponden a cada estado.

En este caso en cada uno de los estado se ejecutan acciones.