



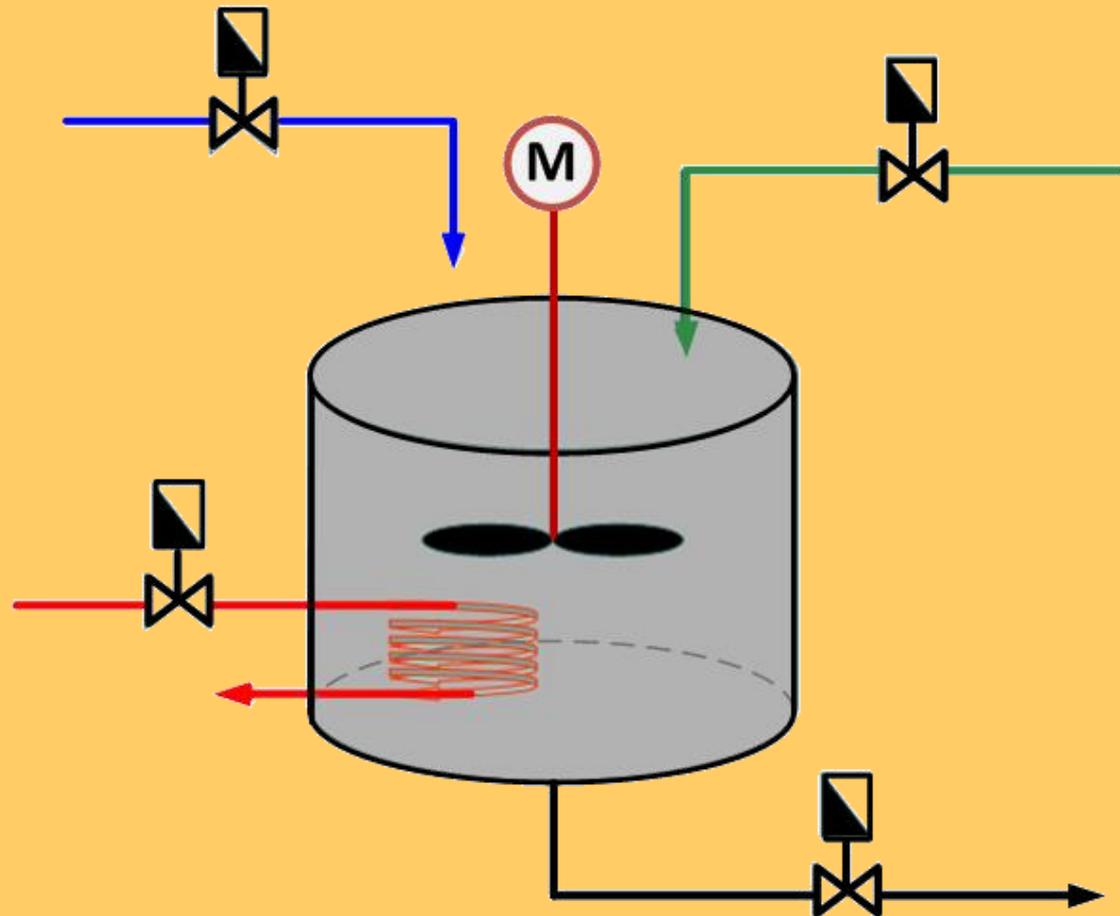
# PROGRAMACIÓN DE PROCESOS SECUENCIALES

Hablamos de **Control Secuencial** cuando hay una sucesión de etapas de operación con acciones específicas y condiciones de transición entre ellas.

## EJEMPLO

Operación de un reactor batch con las etapas:

1. Espera
2. Carga
3. Operación
4. Descarga

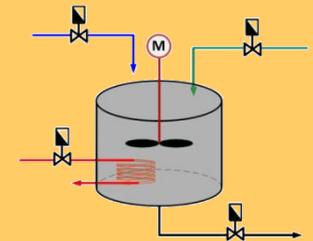




# PROGRAMACIÓN DE PROCESOS SECUENCIALES

Para modelar sistemas secuenciales hay que establecer tres parámetros:

- ▶ **ESTADO O ETAPA:** etapa en la que se encuentra el sistema en un dado período de tiempo en las que se realizan una o más acciones.
- ▶ **TRANSICIONES:** son las condiciones que deben darse para pasar de un estado a otro. Se formulan como funciones lógicas de las entradas. Una transición se activa cuando la función lógica asociada tiene un valor cierto.
- ▶ **ACCIONES:** son todas las actividades que el sistema de control debe encargarse que se ejecuten durante una etapa.



En la programación de Control Secuencial se deben determinar los parámetros en forma clara y precisa .



# GRAFO DE TRANSICIÓN DE ESTADOS (máquina de estados)

Es una **representación gráfica** que permite poner en evidencia la forma en la que se opera el sistema secuencial con determinadas entradas y salidas. Se construye a partir de la descripción de la operación.

Consta de un conjunto de estados que sirven de intermediarios en esta relación de entradas y salidas. El historial de señales de entrada determina para cada instante, un estado para la máquina, de forma tal que la salida depende únicamente del estado y las entradas actuales. Se usan lo siguientes símbolos:

**ESTADO**  
Identificado con  
un número



**TRANSICIÓN**  
Asociado a condiciones  
lógicas





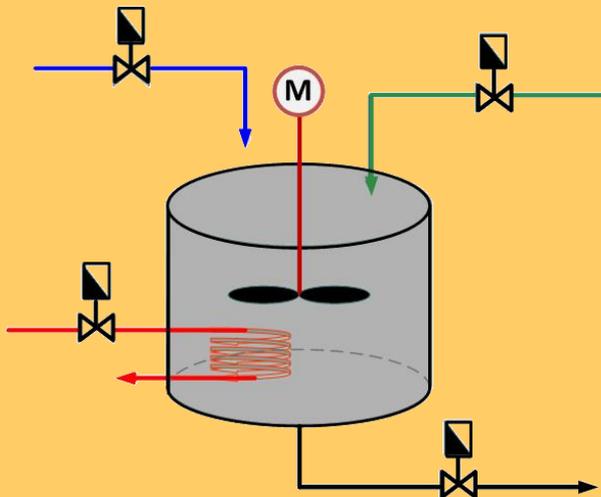
# GRAFO DE TRANSICIÓN DE ESTADOS

## EJEMPLO

Operación de un reactor batch con las etapas:

1. Espera
2. Carga
3. Operación
4. Descarga

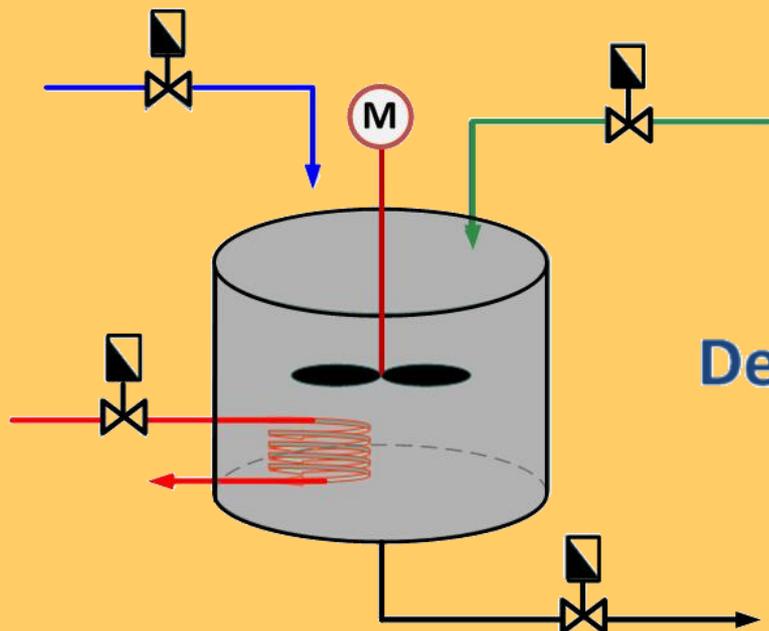
Nº	ESTADO	ACCIONES
1	Espera	Ninguna
2	Carga	Apertura de válvulas de alimentación
3	Operación	1. Arranque del motor de agitación 2. Apertura de válvula del fluido calefactor
4	Descarga	Apertura de la válvula de descarga



TRANSICIÓN	CONDICIÓN
1 - 2	Orden de arranque
2 - 3	Tanque lleno
3 - 4	Operación concluida
4 - 1	Tanque vacío

# GRAFO DE TRANSICIÓN DE ESTADOS

Se puede visualizar cómo va evolucionar cíclicamente el sistema y es la base para la programación en lenguajes gráficos.





# SISTEMAS DE DOS DE ESTADOS - Ejemplo

Un sistema muy difundido es el **ARRANQUE – PARADA** de motores por medio de pulsadores.

**Automatismo Secuencial** (la salida depende de las entradas y del estado en el que se encuentra el sistema).

## ENTRADAS

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

## SALIDA

- ▶ Relé de arranque o parada del motor (**Q**)

Sistema de **2 entradas y 1 salida**



# SISTEMAS DE DOS DE ESTADOS - Ejemplo

## ESTADOS (ETAPAS)

Es evidente que existen sólo **dos estados**:

- ▶ Motor parado (**1**)
- ▶ Motor arrancado (**2**)



## ACCIONES

- ▶ Con motor parado (**1**) la salida tiene que ser motor apagado
- ▶ Con motor arrancado (**2**) la señal de salida es motor encendido

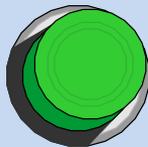
## TRANSICIONES

- ▶ Parado a Arrancado (1 a 2)  
Pulsar **arranque** 
- ▶ Arrancado a Parado (2 a 1)  
Pulsar **paro** 

# SISTEMAS DE DOS DE ESTADOS - Ejemplo

## PULSADOR DE ARRANQUE

Tipo Normal Abierto (N/A), cuando se pulsa, cierra un circuito.



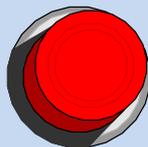
Arranque



Motor  
Parado

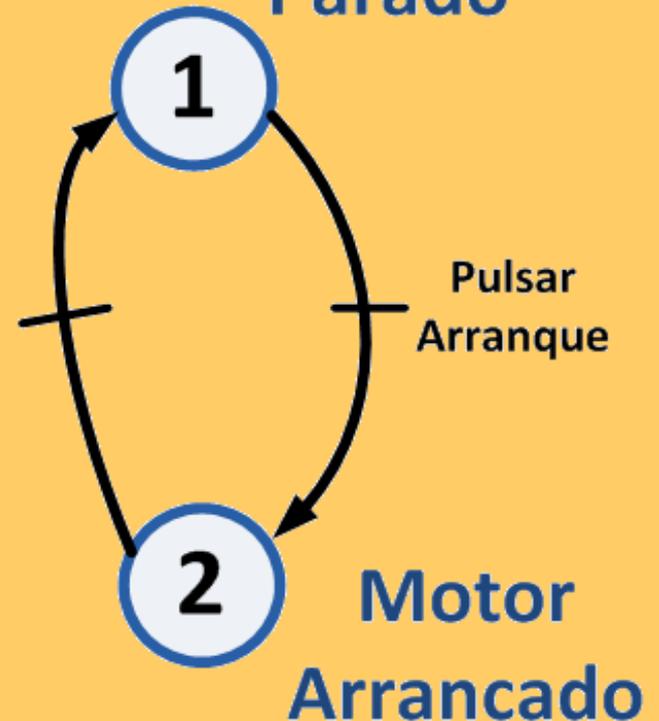
## PULSADOR DE PARADA

Tipo Normal Cerrado (N/C), cuando se pulsa, abre un circuito.



Paro

Pulsar  
Parada



Motor  
Arrancado



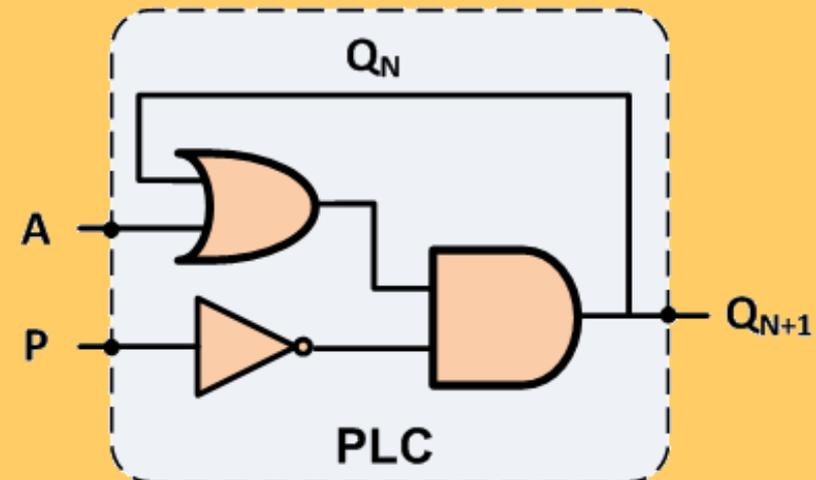
# SISTEMAS DE DOS DE ESTADOS - Ejemplo

La tabla de la verdad en este caso se construye con las dos variables de entrada (**A** y **P**) y con el **estado actual** del sistema ( $Q_N$ ) para poder determinar la **salida siguiente** ( $Q_{N+1}$ ).

$Q_N$ (actual)	A	P	$Q_{N+1}$ (siguiente)
0	0	0	0
0	0	1	0
0	1	0	1
0	<b>1</b>	<b>1</b>	<b>No factible</b>
1	0	0	1
1	0	1	0
1	1	0	1
1	<b>1</b>	<b>1</b>	<b>No factible</b>

De la Tabla y simplificando se puede probar que:

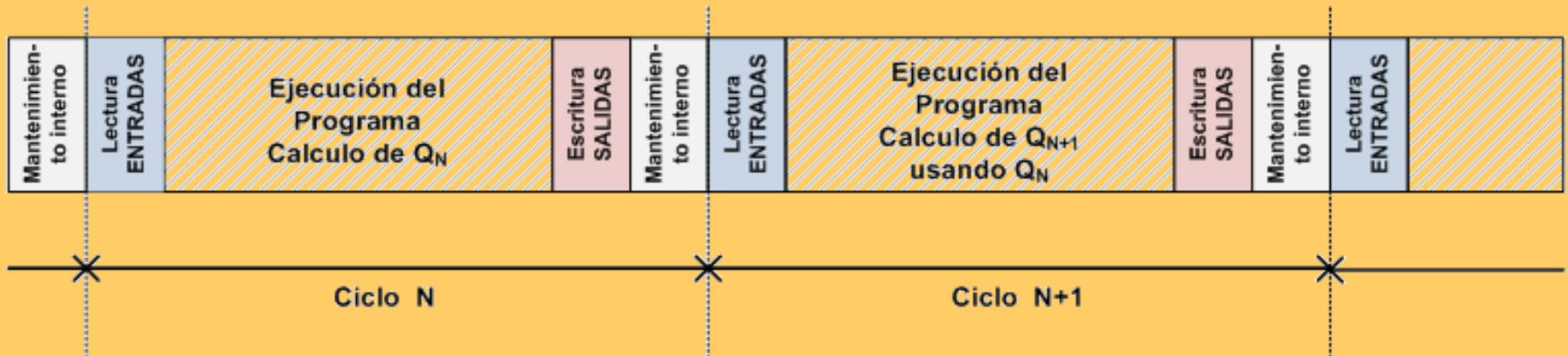
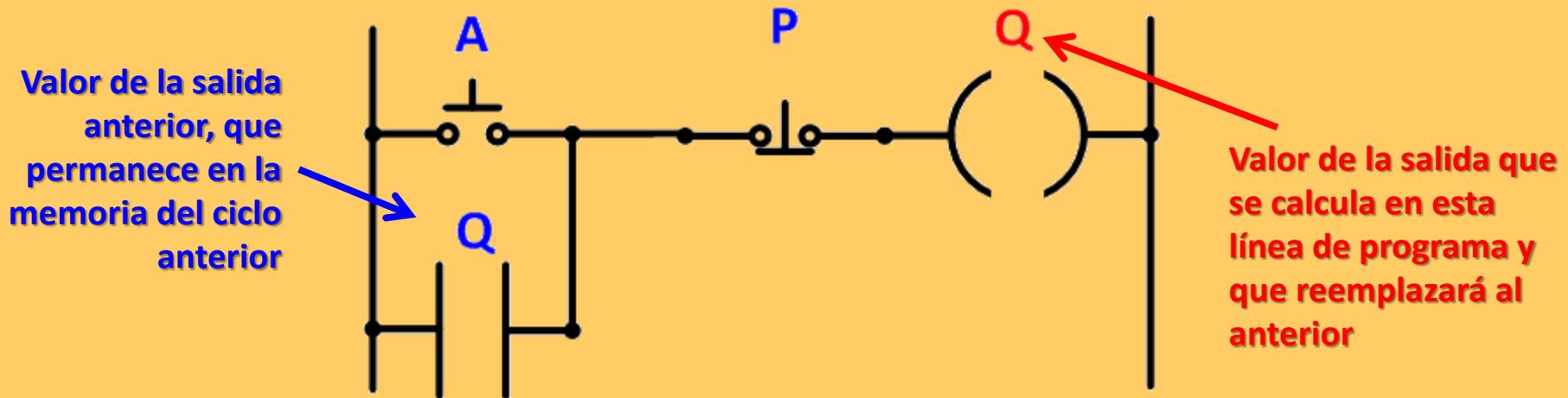
$$Q_{N+1} = (A + Q_N) \cdot \bar{P}$$





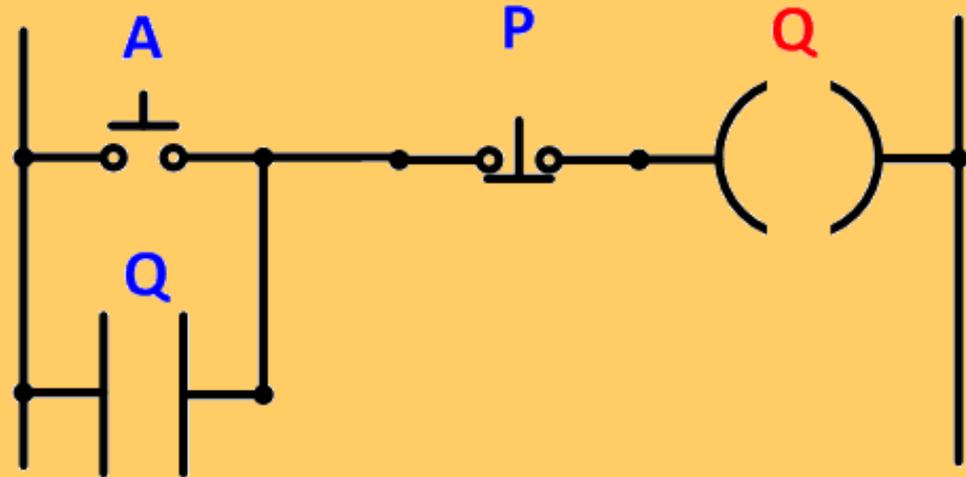
# SISTEMAS DE DOS DE ESTADOS - Ejemplo

En lenguaje de contactos se puede representar usando pulsadores:

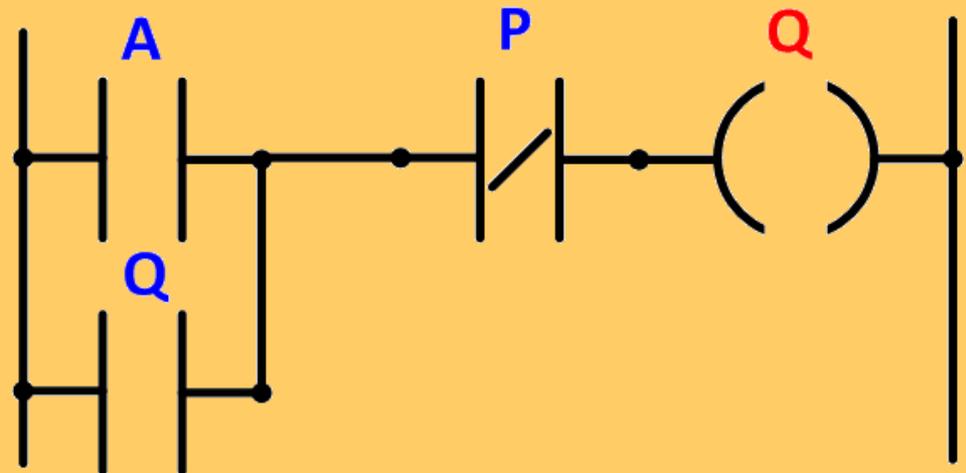


# SISTEMAS DE DOS DE ESTADOS - Ejemplo

En lenguaje de contactos con pulsadores (didáctico)



Representación con contactos



# SISTEMAS DE DOS DE ESTADOS - Ejemplo

Un diagrama del tiempo muestra la evolución del sistema ante distintas entradas





# DIAGRAMA DE FUNCIONES SECUENCIALES (SFC) o GRAFCET

**GRAFCET** es el acrónimo **GRA**fico **F**uncional de **C**ontrol de **E**tapas y **T**ransiciones y es un **lenguaje gráfico de programación de autómatas**.

En forma gráfica describe el comportamiento de un proceso secuencial compuesto por etapas y transiciones.

La norma **IEC 61131-3** define los elementos gráficos y las reglas de evolución que reflejan la dinámica del comportamiento del sistema.



# GRAF CET

La norma IEC 61131-3 establece:

## ETAPAS

- ▶ El proceso se descompone en Etapas (estados).
- ▶ Se representan por cuadrados y se les asigna un número único.
- ▶ La etapa inicial se indica con un cuadrado doble.

## CONDICIONES DE TRANSICIÓN

- ▶ Es la condición necesaria para la evolución de una etapa a otra.
- ▶ Entre dos etapas siempre tiene que haber una condición.
- ▶ Cada transición debe estar asociada una condición lógica.

## PROCEDIMIENTO DE EJECUCIÓN

- ▶ Partiendo de la etapa inicial, las etapas se activarán cuando, estando activa la anterior, se cumple la condición de transición.
- ▶ Al activarse una etapa se desactiva la anterior.



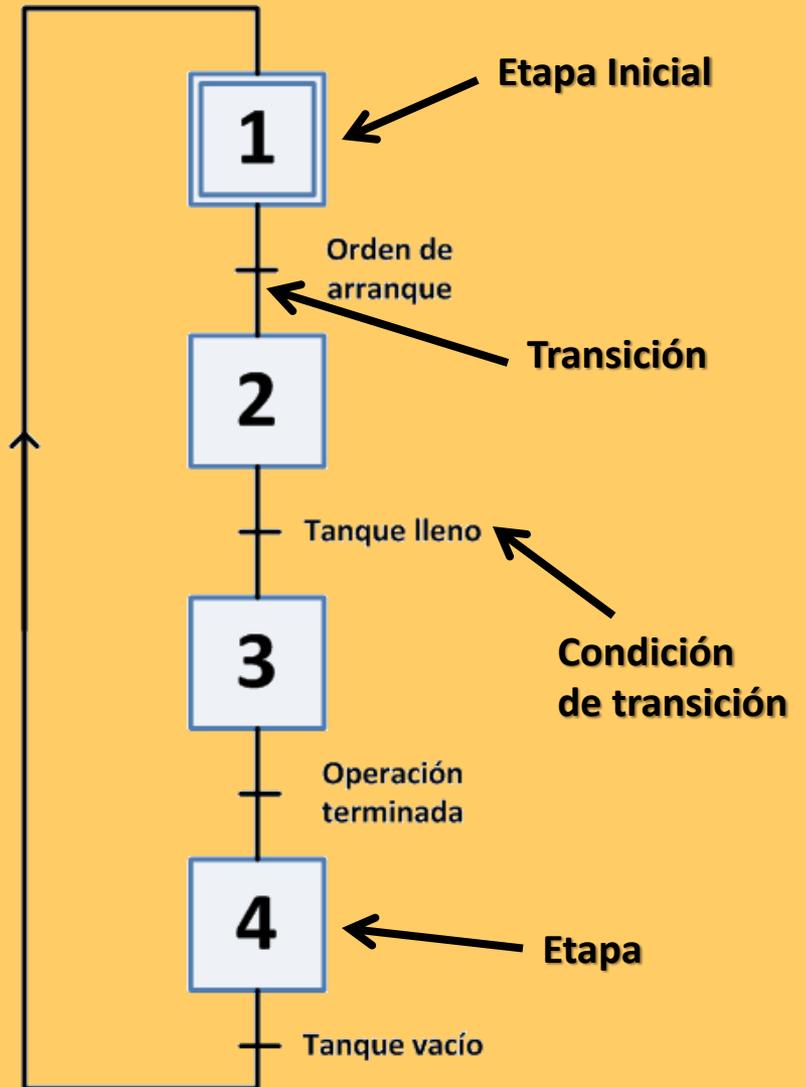
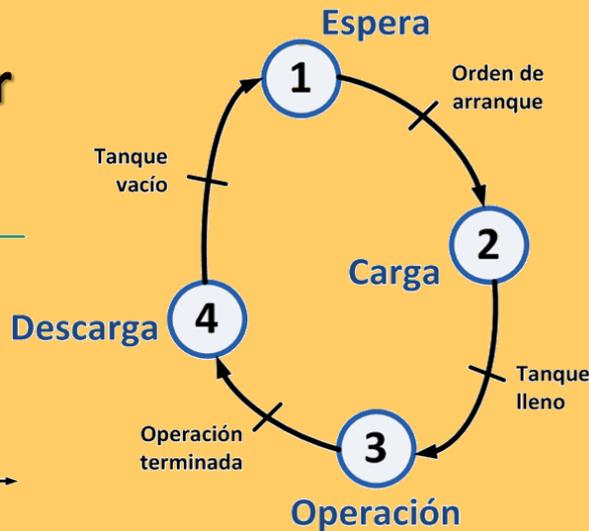
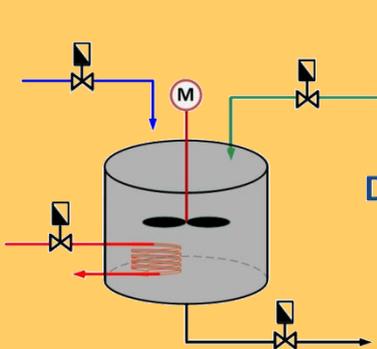
# GRAFCET

- ▶ Al arrancar el autómeta se establecen las condiciones iniciales y a partir de ese instante se consultan de forma sucesiva los diferentes estados y se calculan las diferentes expresiones lógicas que gobiernan las transiciones asociadas con cada uno de ellos.
- ▶ Cada etapa indica el estado en que se encuentra el proceso.
- ▶ Las etapas **implican acciones** asociadas.
- ▶ Sólo algunas de las etapas pueden encontrarse activas en cada ciclo del autómeta.
- ▶ Las transiciones gobiernan los cambios de estado.
- ▶ El GRAFCET se puede generar en forma directa a partir del grafo de transición estados.



# GRAFSET

## Ejemplo del reactor



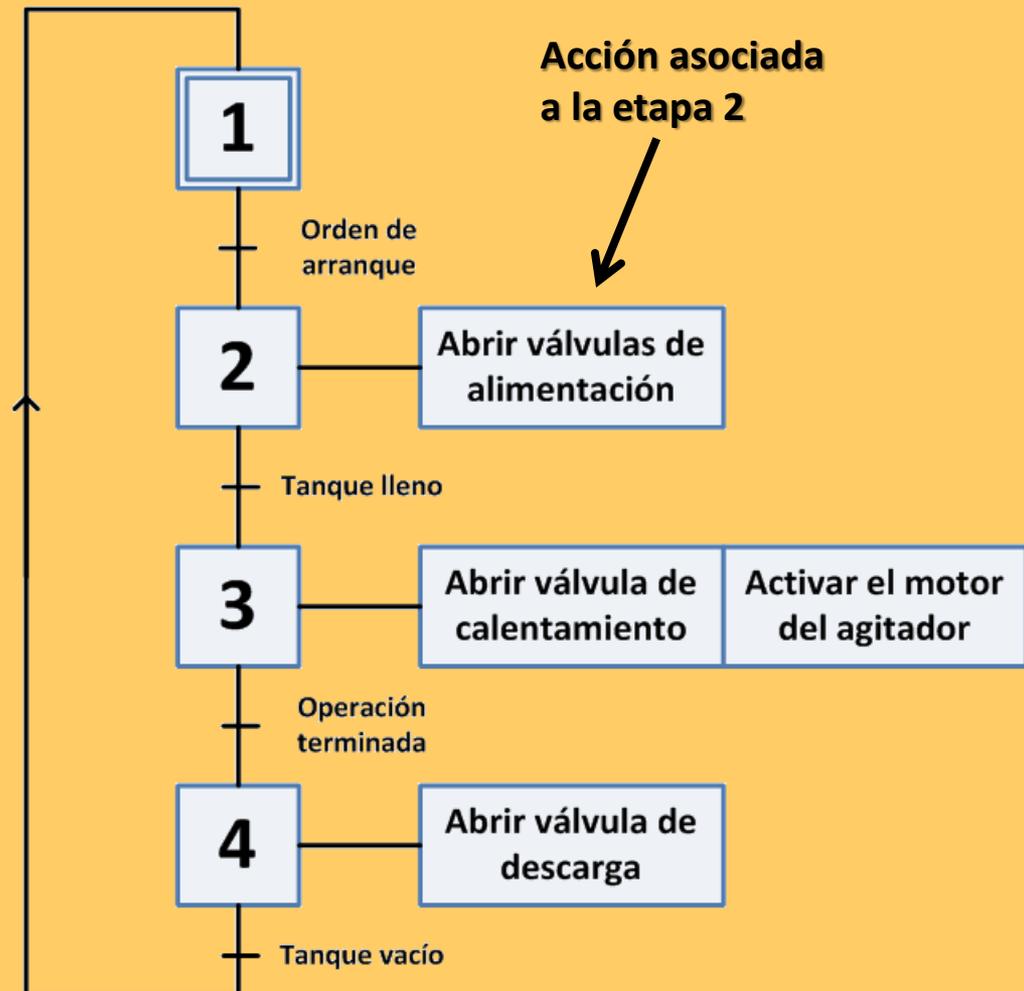
Las líneas de evolución unen entre sí las etapas que representan actividades consecutivas. Se entenderán siempre orientadas de arriba hacia abajo, a menos que se represente una flecha en sentido contrario.



# GRAFFCET

A cada etapa se la identifica con un número que **no necesariamente debe guardar un orden coherente a la secuencia del proceso.**

Las **acciones** que llevan asociadas las etapas se representan con un rectángulo lateral donde se indica el tipo de acción a realizar (sólo las activas). Una etapa puede llevar asociadas varias acciones.

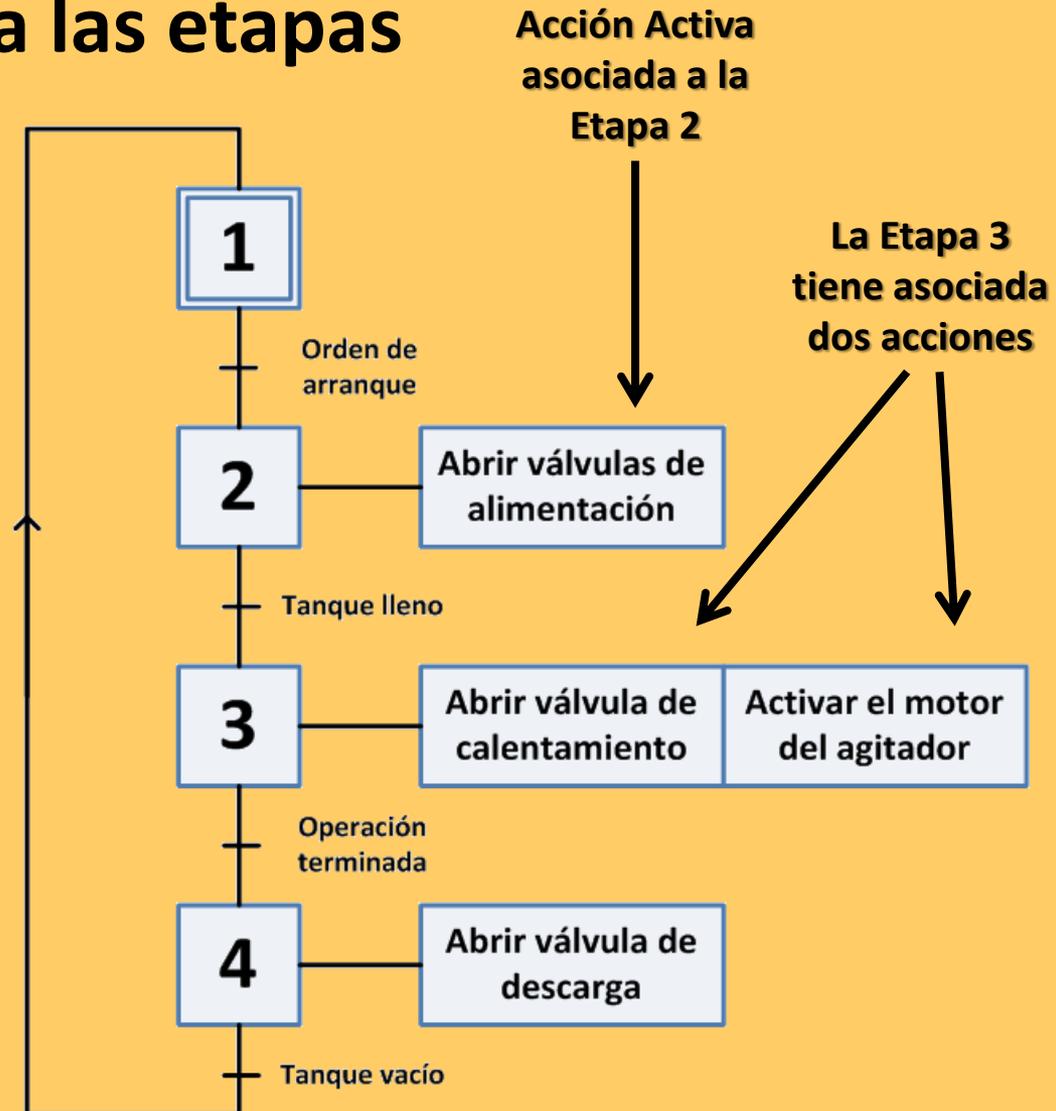




# GRAFFCET

## Acciones asociadas a las etapas

- ▶ Dentro de una etapa pueden asociarse **una o más acciones** elementales o complejas.
- ▶ Las acciones se realizarán **sólo** cuando la etapa correspondiente esté activa.

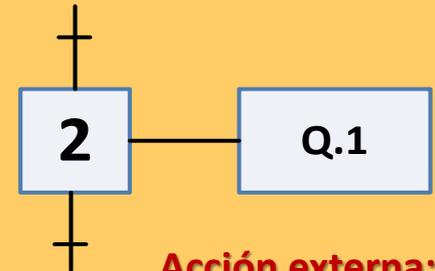




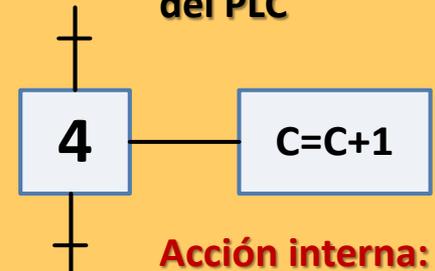
# ACCIONES EN GRAFCET

Las acciones asociadas a una etapa pueden ser:

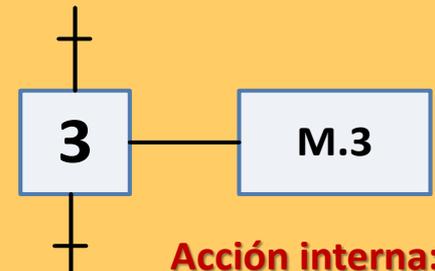
- ▶ **Externas:** activación o desactivación de alguna salida.
- ▶ **Internas:** arranque de temporizadores, arranque de contadores, activación de bits (marcas), etc.



**Acción externa:**  
Activa la salida Q.1 del PLC

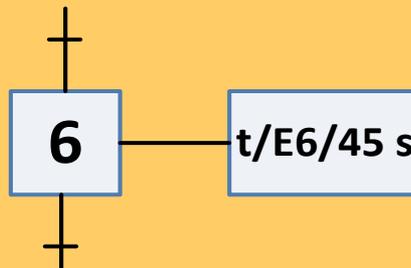


**Acción interna:**  
Incrementa el contador C en una unidad



**Acción interna:**  
Graba un 1 en el bit de memoria M.3

**Acción interna:**  
Activa un temporizador TON con un retardo de 45 s

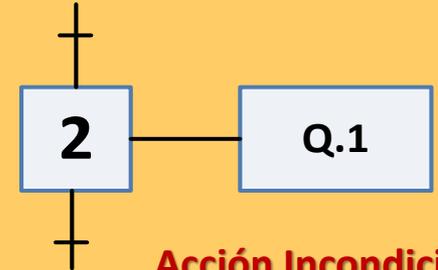




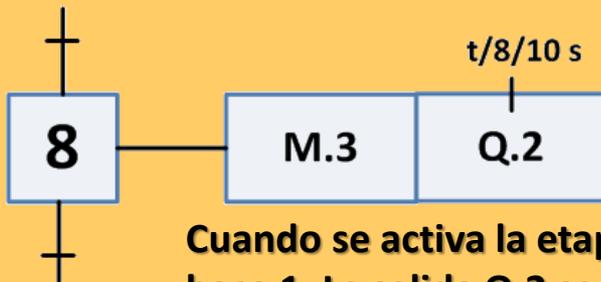
# GRAFFCET

## Acciones asociadas a las etapas

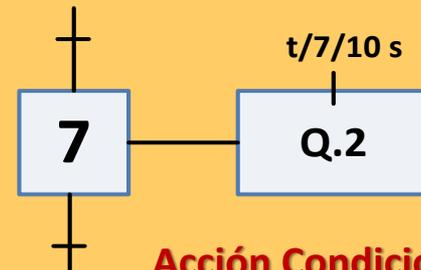
- ▶ Las acciones **incondicionadas** son las que deben ejecutarse siempre cuando la etapa está activa.
- ▶ La acción es **condicionada** cuando hay una condición lógica que cuando es cierta se ejecuta, de otro modo no.



**Acción Incondicionada**  
 Q.1 se activa siempre (se pone en 1) mientras la etapa 2 esté activa



Cuando se activa la etapa 8, el bit interno M.3 se hace 1. La salida Q.2 se activa después de 10 s que se activó la etapa 8 (TON)

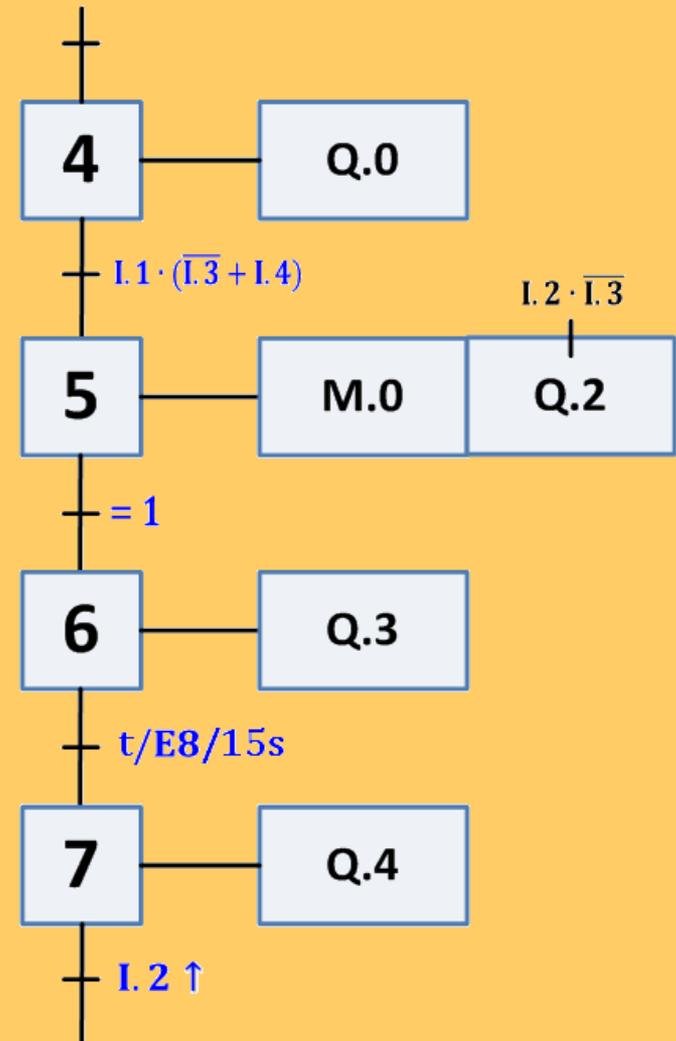


**Acción Condicionada**  
 Q.2 se activa 10 s después de que se activó la etapa 7 (TON)



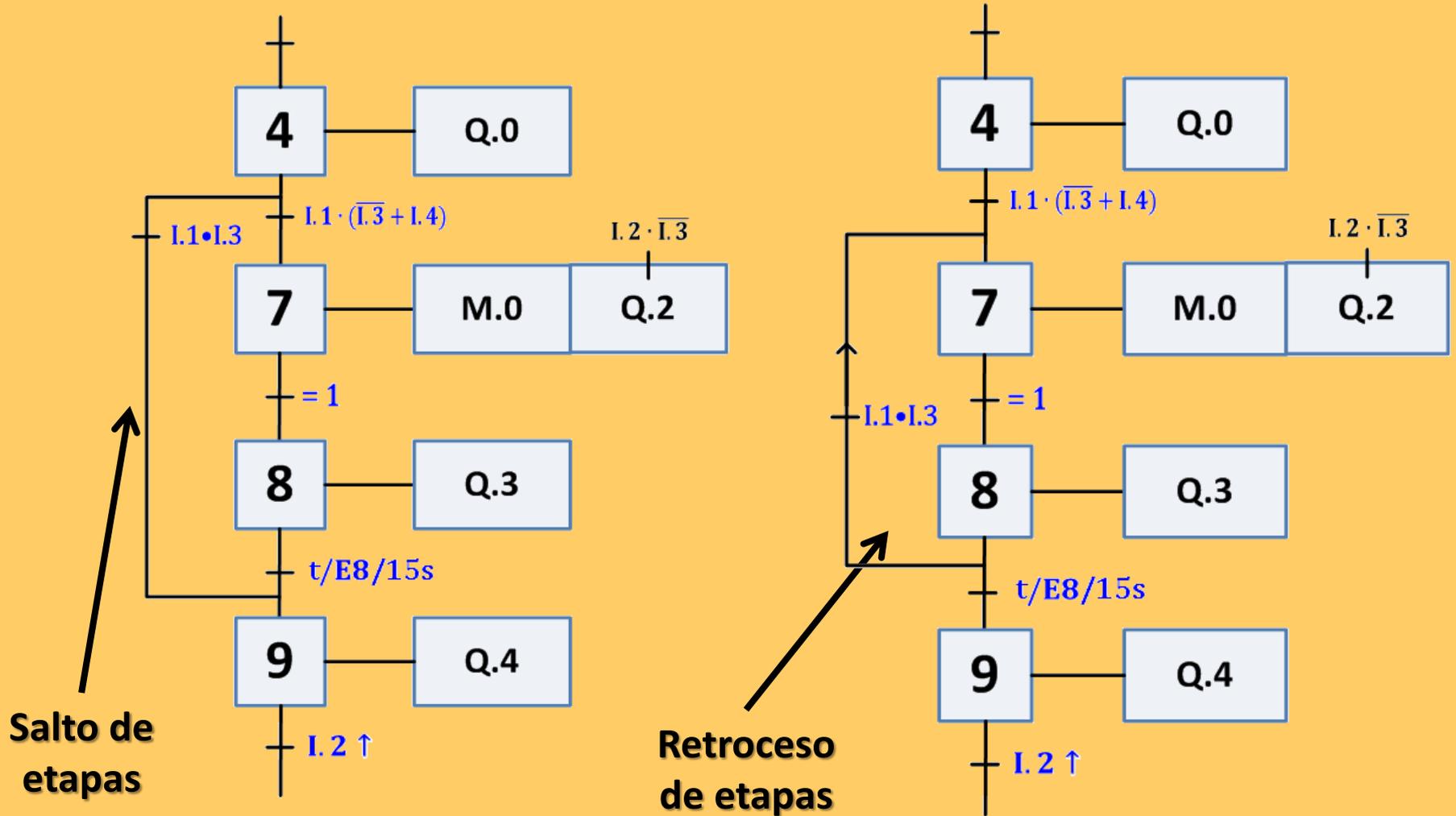
# TRANSICIONES

- ▶ Las condiciones de las transiciones **se escriben a la derecha** de la línea que las une.
- ▶ Las transiciones son una **función lógica** de variables de entrada e internas del sistema.
- ▶ Hay dos formas de indicar las condiciones:
  - Simbólica
  - Literal
- ▶ Para indicar una condición siempre verdadera se usa “= 1”





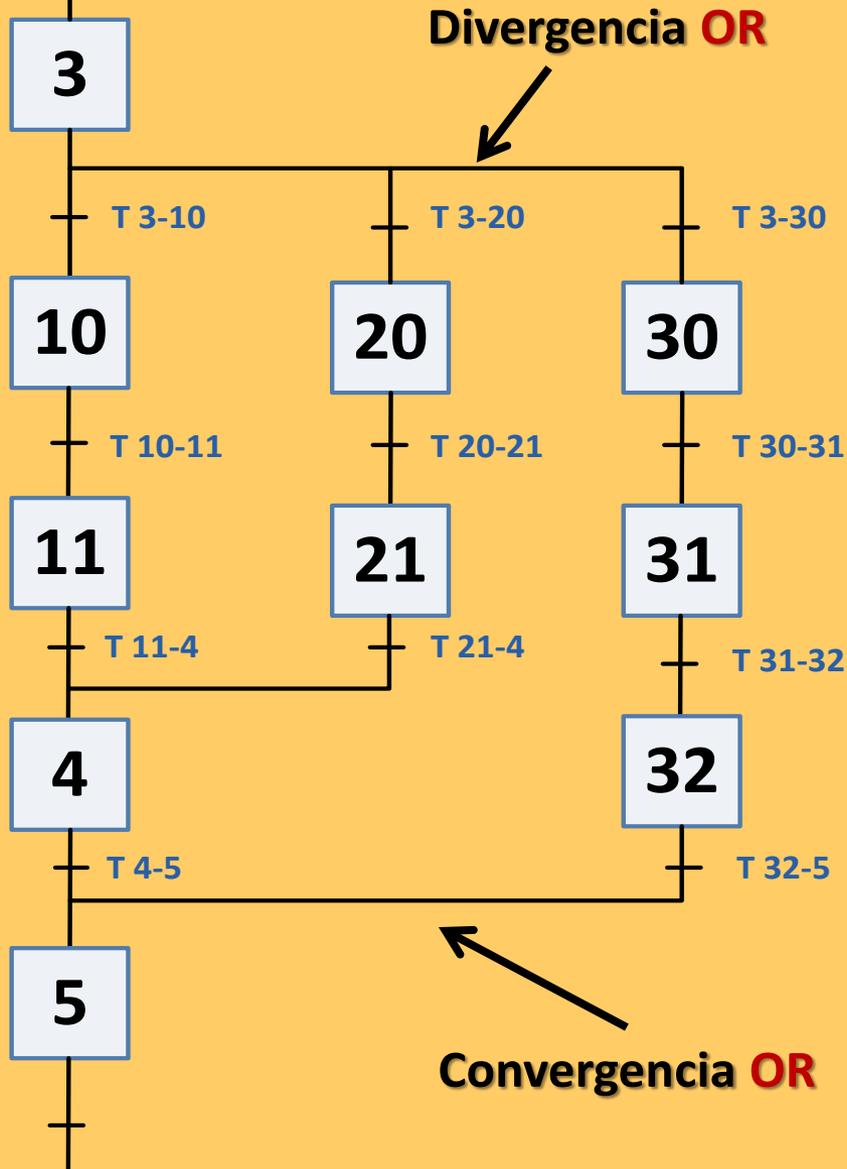
# TRANSICIONES





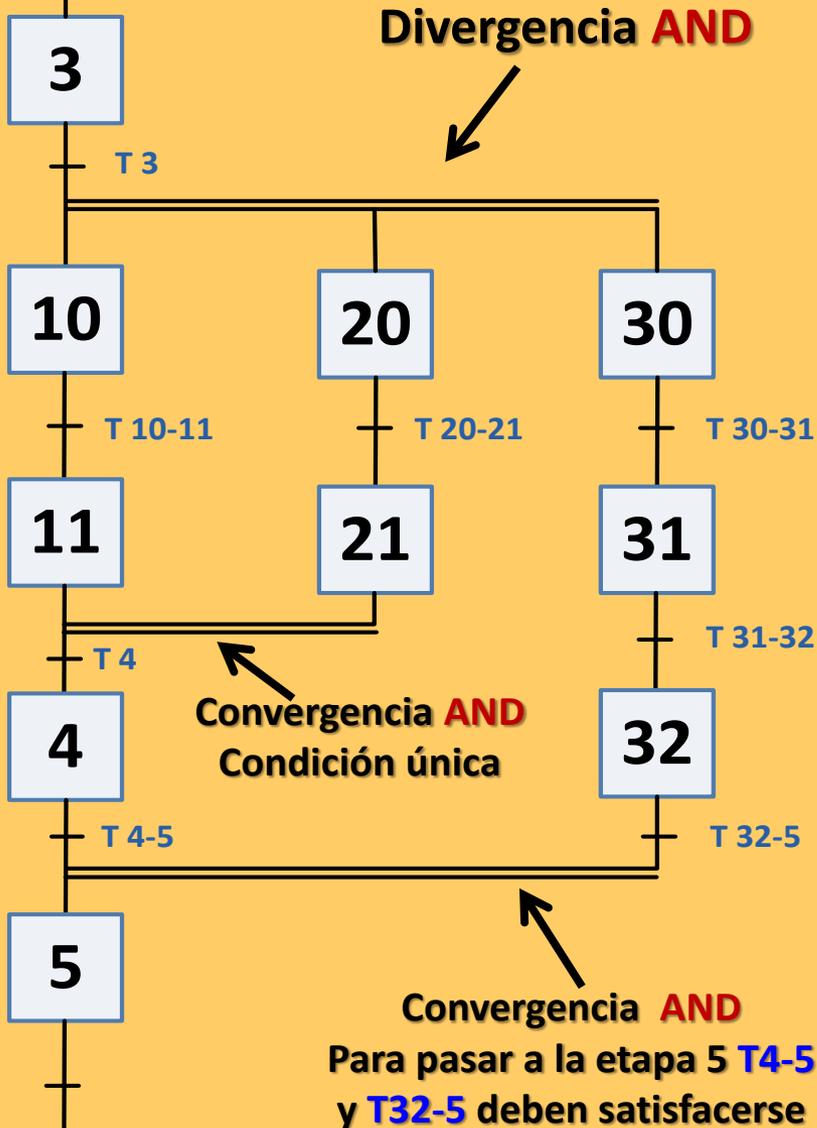
# ESTRUCTURAS DEL GRAFCET

- ▶ **Secuencia lineal:** a continuación de cada etapa haya una transición y después de cada transición una etapa (Ejemplo del reactor).
- ▶ **Secuencias alternativas (OR):** después de una etapa se presentan dos o más caminos posibles (dos o más transiciones) de los cuales sólo uno es el que puede tomar la lógica del proceso; esto se determina haciendo que cada transición posea una condición distinta y excluyentes entre sí.
- ▶ **Secuencias simultáneas (AND):** a partir de cierta etapa del proceso, al producirse una transición se inician dos o más secuencias simultáneamente y luego finalizadas cada una y todas las secuencias, se continúa con la secuencia lineal.



# SECUENCIAS ALTERNATIVAS

- ▶ Sub-procesos alternativos
- ▶ En una etapa se pueden distintas secuencias alternativas.
- ▶ Solo una secuencia se activa, por lo que las condiciones en la bifurcación deben ser mutuamente excluyentes.



# SECUENCIAS SIMULTÁNEAS

- ▶ Sub-procesos simultáneos
- ▶ En una etapa se inician varios caminos paralelos .
- ▶ El proceso evolucionará por varios caminos ejecutando tareas simultáneas
- ▶ Cuando los caminos convergen en un estado, todas las ramas deben haberse ejecutado completamente.