

# Automática

2º Curso del Grado en Ingeniería Mecánica



# Copyright

## Autor:

Juan Antonio García Fortes, 2013  
Dpto. Ingeniería de Sistemas y Automática  
Universidad de Málaga  
jagarciaf@uma.es

## Modificado y adaptado de:

victortorreslópez, 2011  
Dpto. Ingeniería de Sistemas y Automática  
Universidad de Málaga  
vetorres@uma.es

## Licencia:

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

- 9.1. Introducción
- 9.2. Principios generales
- 9.3. Símbolos básicos
- 9.4. Estructuras básicas
- 9.5. Diseño de automatismos con GRAFCET
- 9.6. Transcripción de GRAFCET: lenguaje SFC
- 9.7. Ejemplo: doble carro va y viene

# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

### 9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

# Introducción I

---

## Definiciones

- **GRAFCET** (**GRA**phe **Fonctionnel** de **Com**mande **Etape** **T**ransition): lenguaje de especificación para la descripción funcional del comportamiento de la parte secuencial de un sistema.
- **IEC 60848:2002**: lenguaje de especificación GRAFCET para diagramas secuenciales de función.

# Introducción II

---

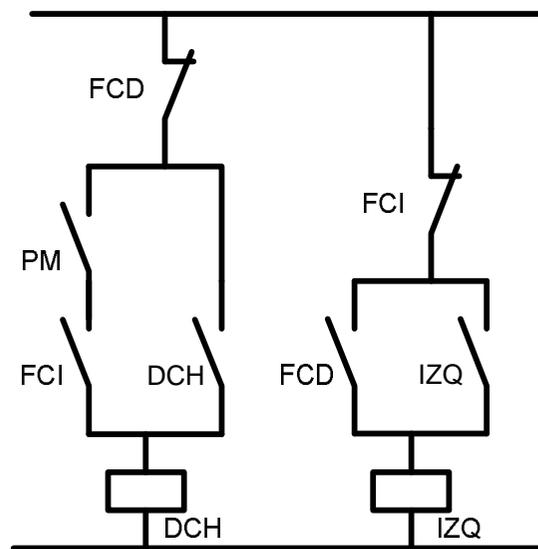
## Historia GRAFCET

- 1962: Kommunikation mit Automaten, C.A. Petri
- 1975: grupo AFCET, Michael Blanchard
- 1977: nacimiento del GRAFCET
- 1982: NF C03-190 (AFNOR)
- 1988: IEC 848 (CEN): fundamentos
- 2002: IEC 60848 (CEN): estructuración

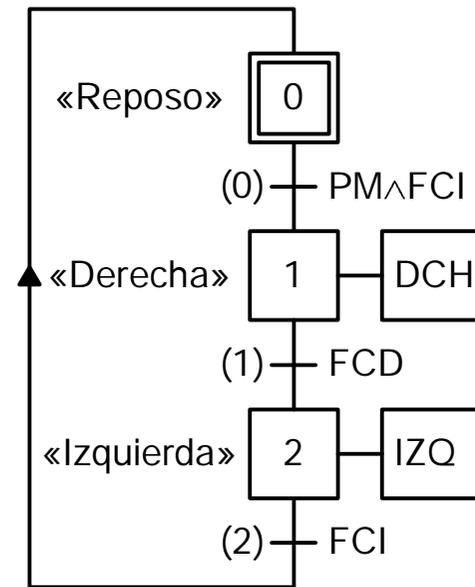
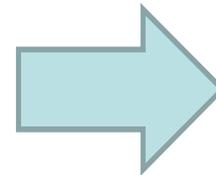
# Introducción III

## DRC vs GRAFCET

La diferencia principal entre DRC y GRAFCET es que el diseño en GRAFCET está orientado al estado y en DRC está orientado a las salidas.



DRC



GRAFCET

## Características GRAFCET

- Normalizado
- Orientado al estado
- Estado distribuido
- Representa adecuadamente la secuencia de funcionamiento del sistema
- Permite representar el paralelismo
- Dispone de mecanismos para la organización y jerarquización de los diseños

# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

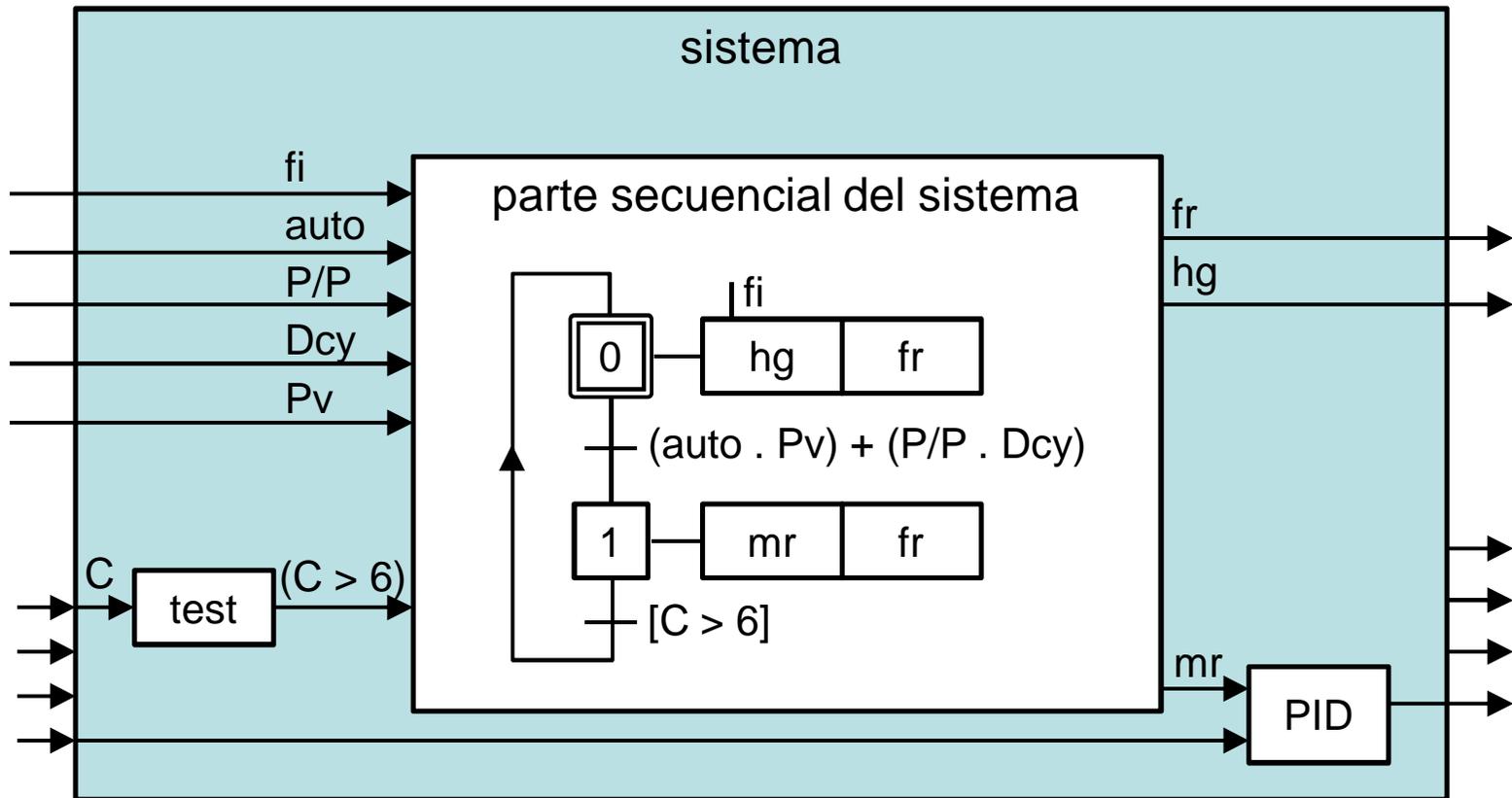
9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

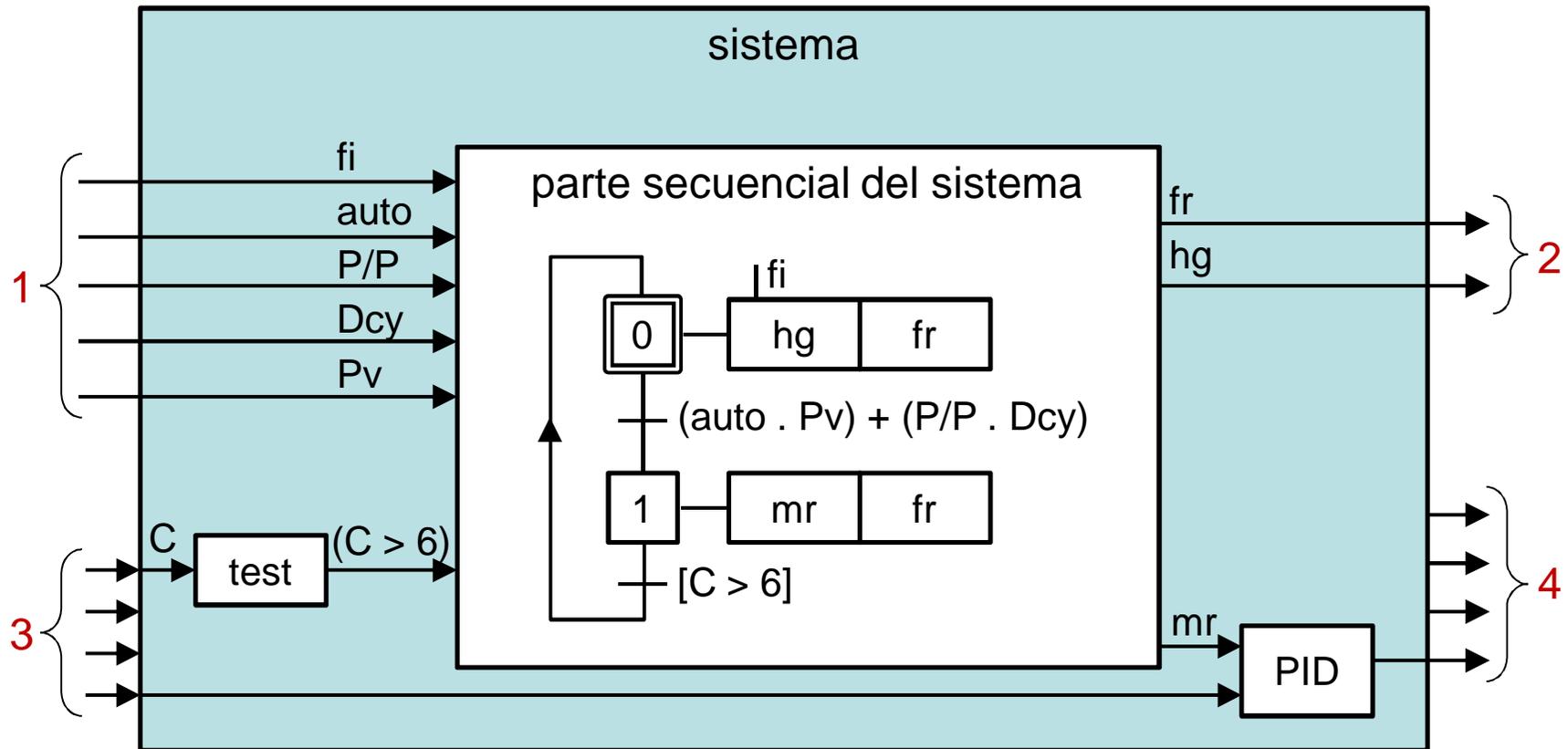
# Principios generales I

## Contexto



# Principios generales I

## Contexto



1: entradas booleanas  
 3: entradas no booleanas

2: salidas booleanas  
 4: salidas no booleanas

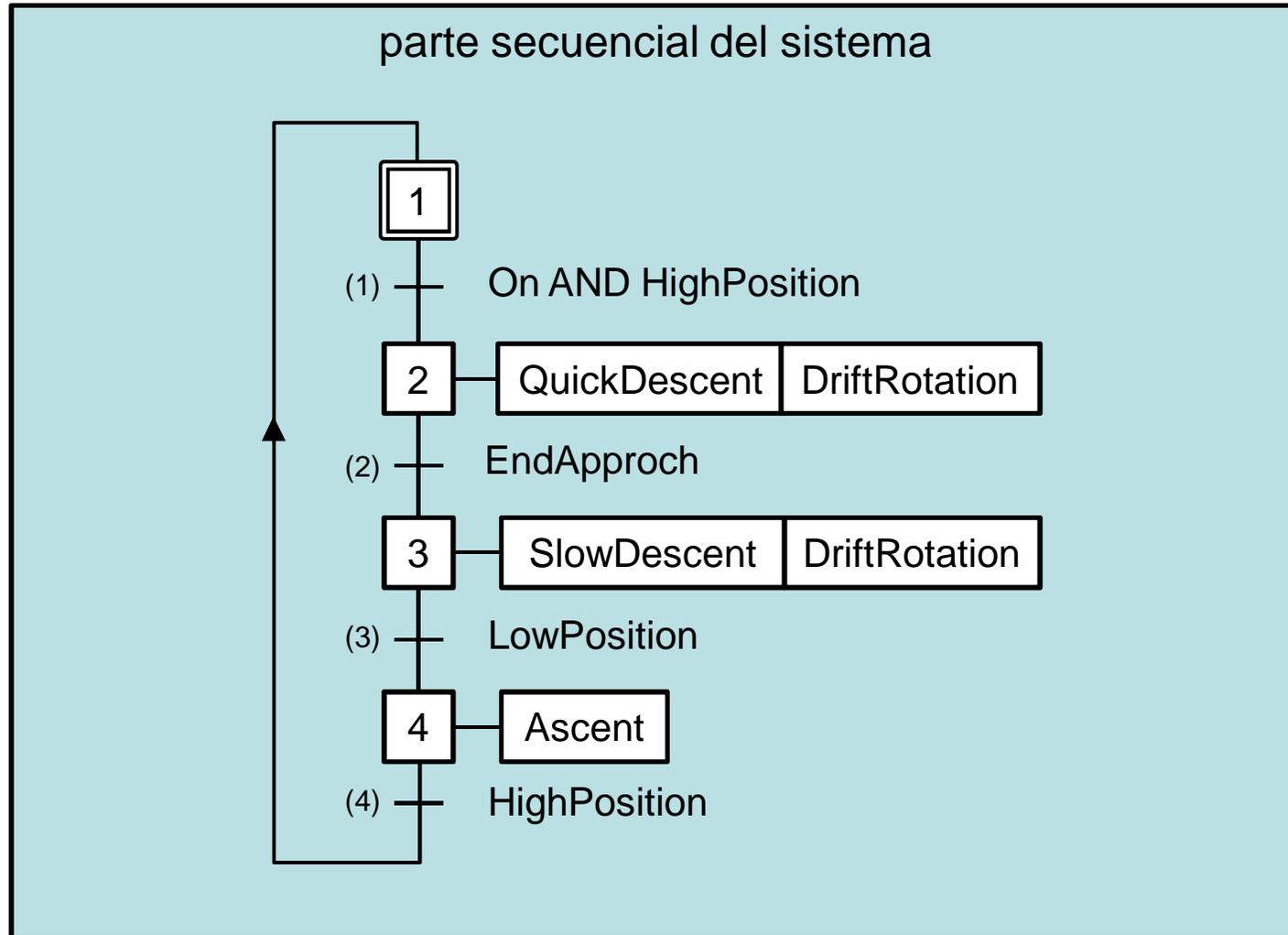
# Principios generales II

---

## Representación I

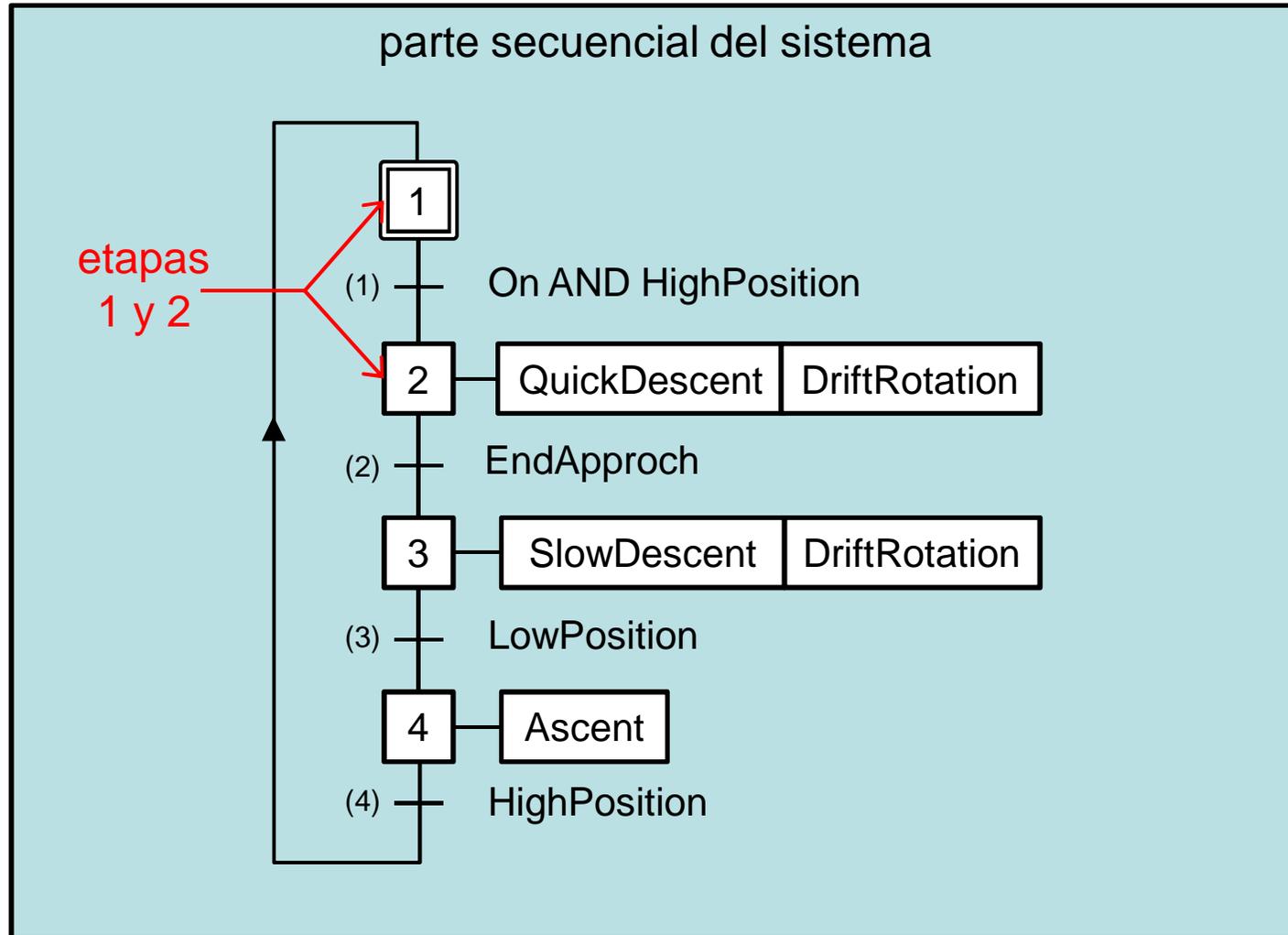
- Estructura:
  - Etapas
  - Transiciones
  - Enlaces dirigidos
- Interpretación:
  - Condiciones de transición
  - Acciones

## Representación II



# Principios generales III

## Representación II



# Principios generales II

---

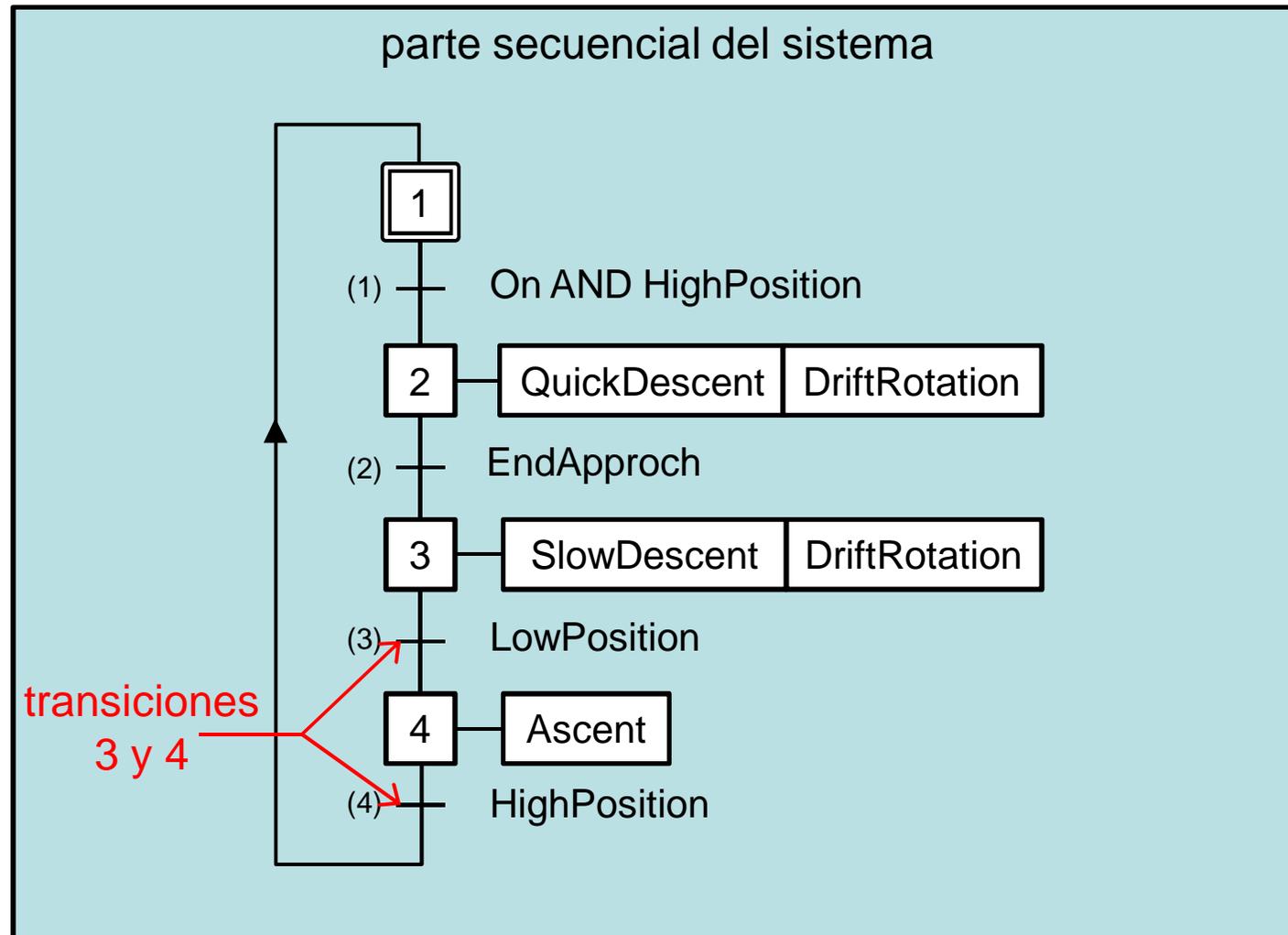
## Representación I

- Estructura:
  - Etapas

Elementos del lenguaje GRAFCET usados para la definición del estado de la parte secuencial del sistema.

- Pueden estar activas o inactivas.
- El conjunto de etapas activas, representan la situación del sistema.

## Representación II



# Principios generales II

---

## Representación I

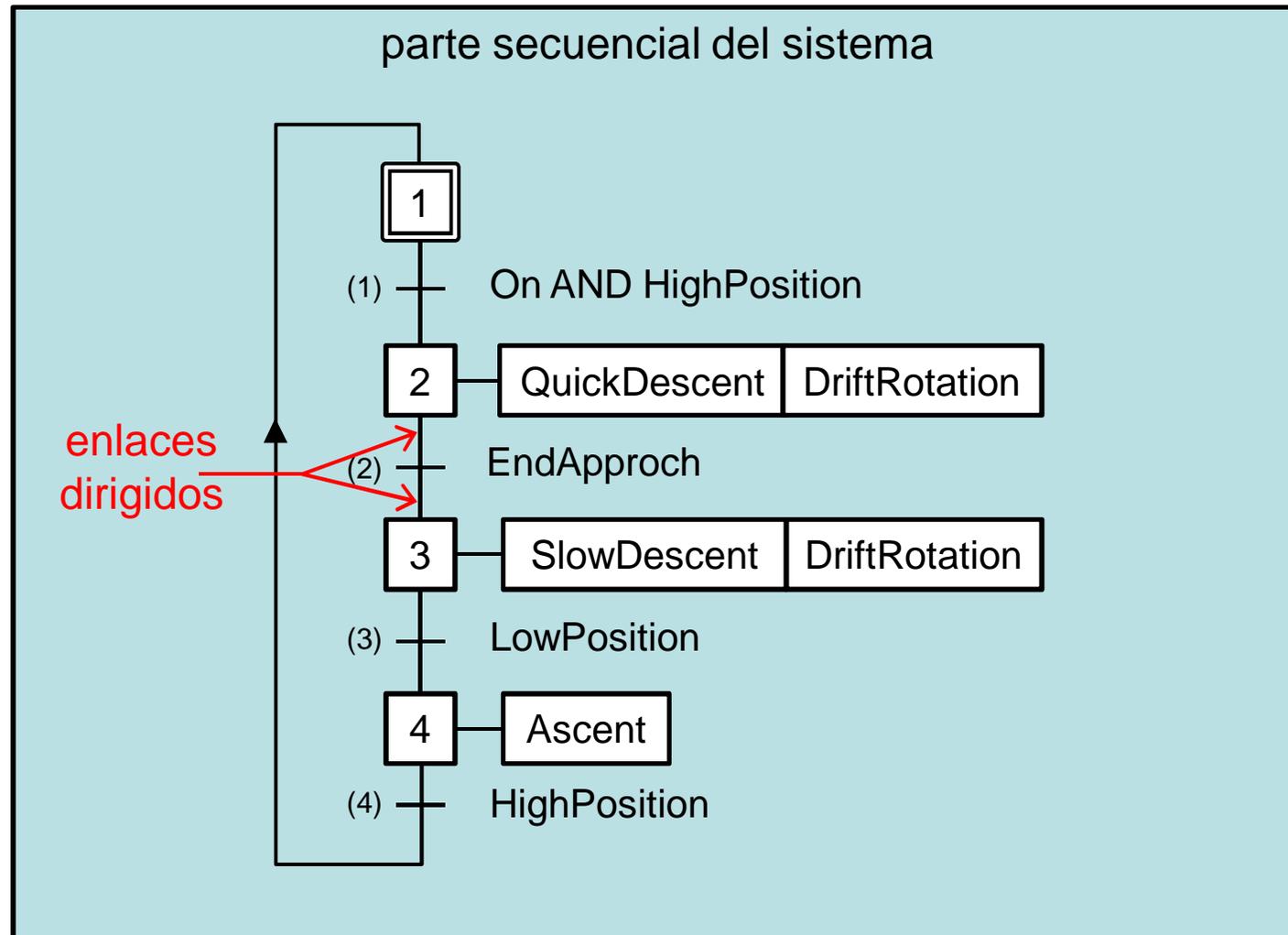
- Estructura:
  - Etapas
  - Transiciones

Elementos del lenguaje GRAFCET que indican una posible evolución de la actividad entre dos o más etapas.

- La posible evolución se lleva a cabo disparando la transición.

# Principios generales III

## Representación II



# Principios generales II

---

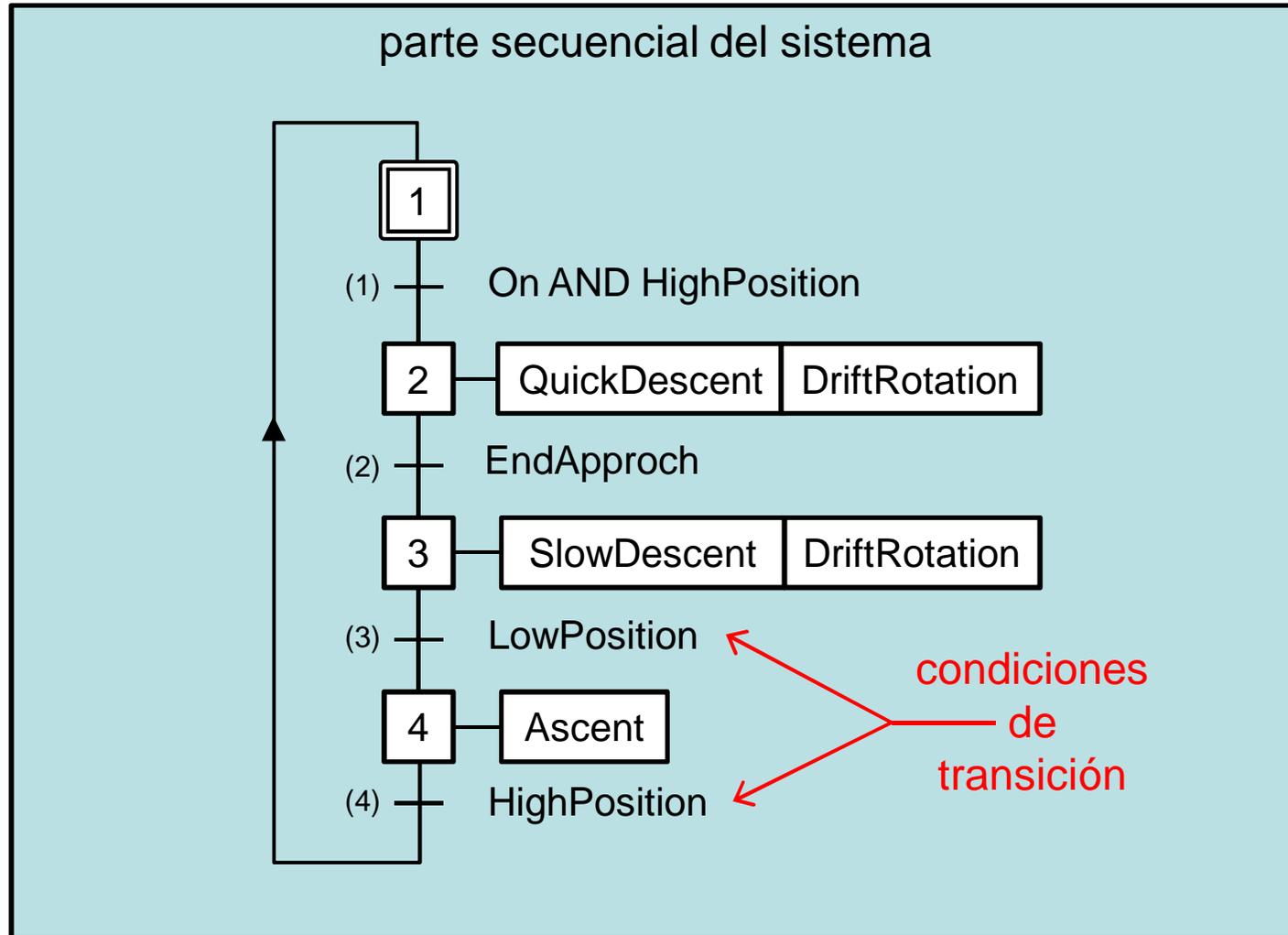
## Representación I

- Estructura:
  - Etapas
  - Transiciones
  - Enlaces dirigidos

- Elementos del lenguaje GRAFCET que indican el camino de evolución entre etapas, conectando etapas a transiciones y transiciones a etapas.

# Principios generales III

## Representación II



# Principios generales II

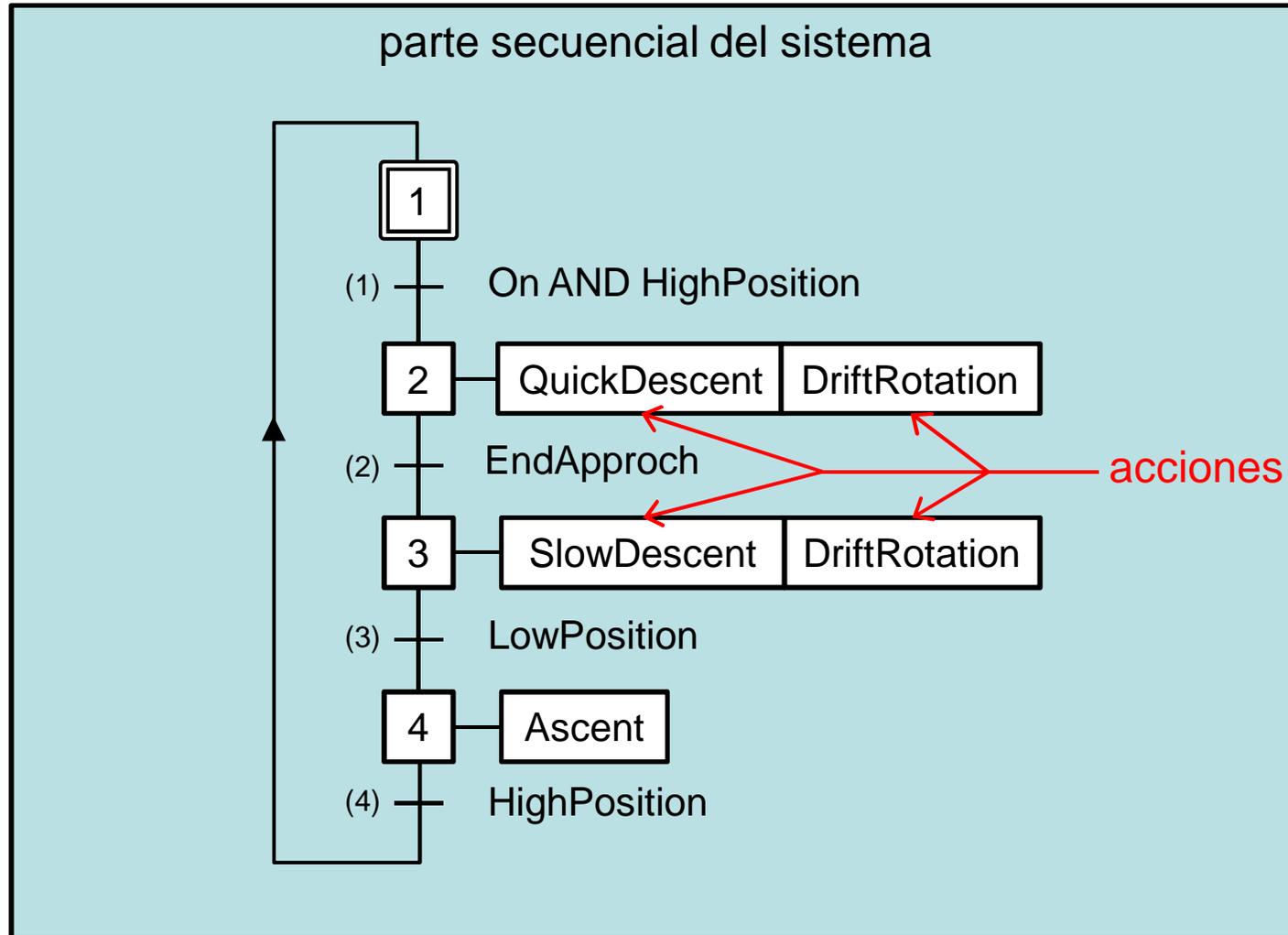
## Representación I

- Estructura:
  - Etapas

Elementos del lenguaje GRAFCET asociados a las transiciones que indican el resultado de una expresión booleana.

- Puede ser falsa o verdadera.
- Condiciones de transición
- Acciones

## Representación II



# Principios generales II

---

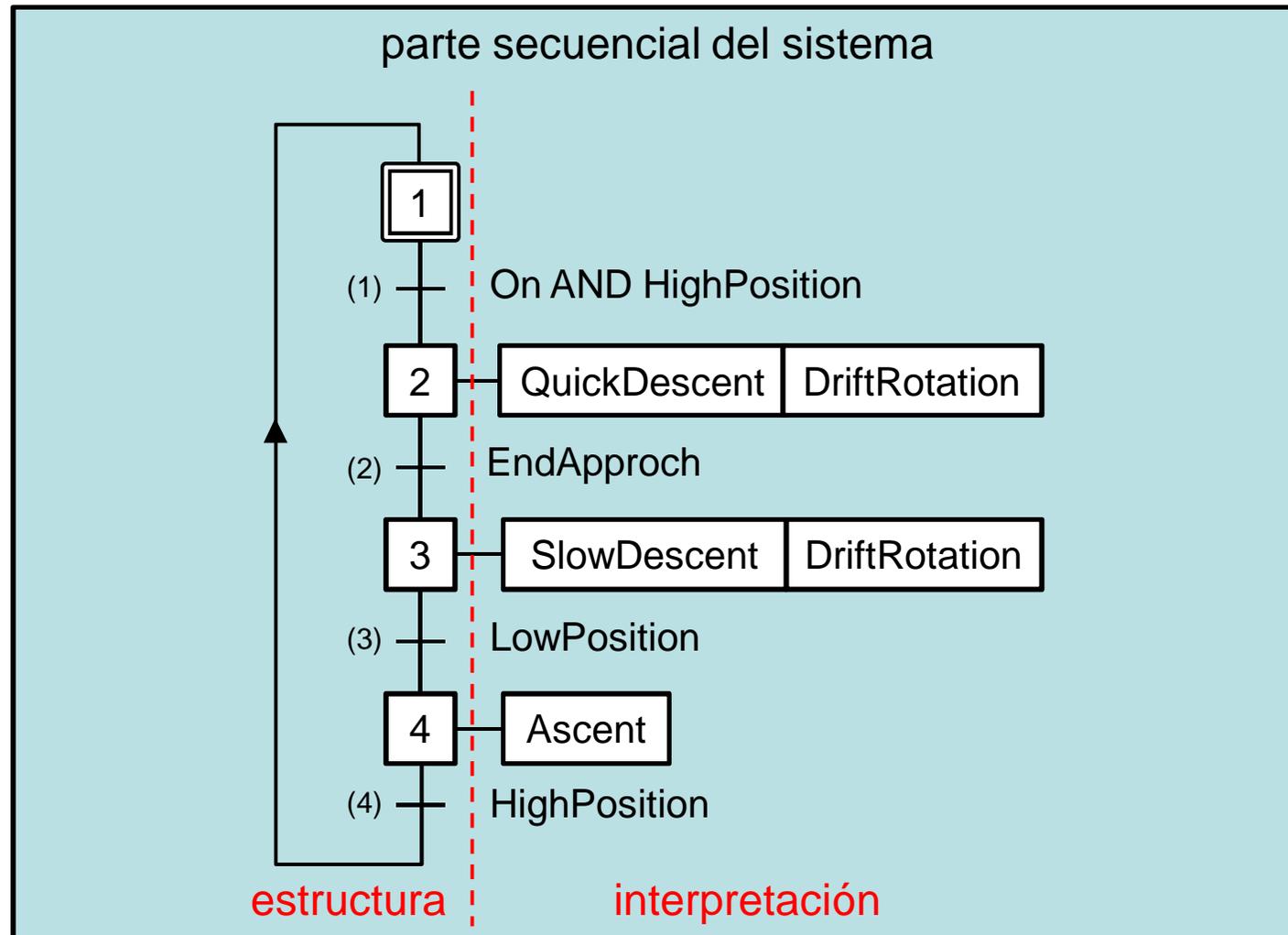
## Representación I

- Estructura:
  - Etapas
  - Transiciones
  - Enlaces dirigidos

- Elementos del lenguaje GRAFCET asociados con las etapas, indican una actividad que se realiza sobre una variable de salida.

- Acciones

## Representación II



# Principios generales II

---

## Representación I

- Estructura:

Parte del GRAFCET que describe las posibles evoluciones entre situaciones. La situación es el nombre del estado del sistema descrito por GRAFCET y caracterizado por las etapas activas en un instante dado.

- Condiciones de transición
- Acciones

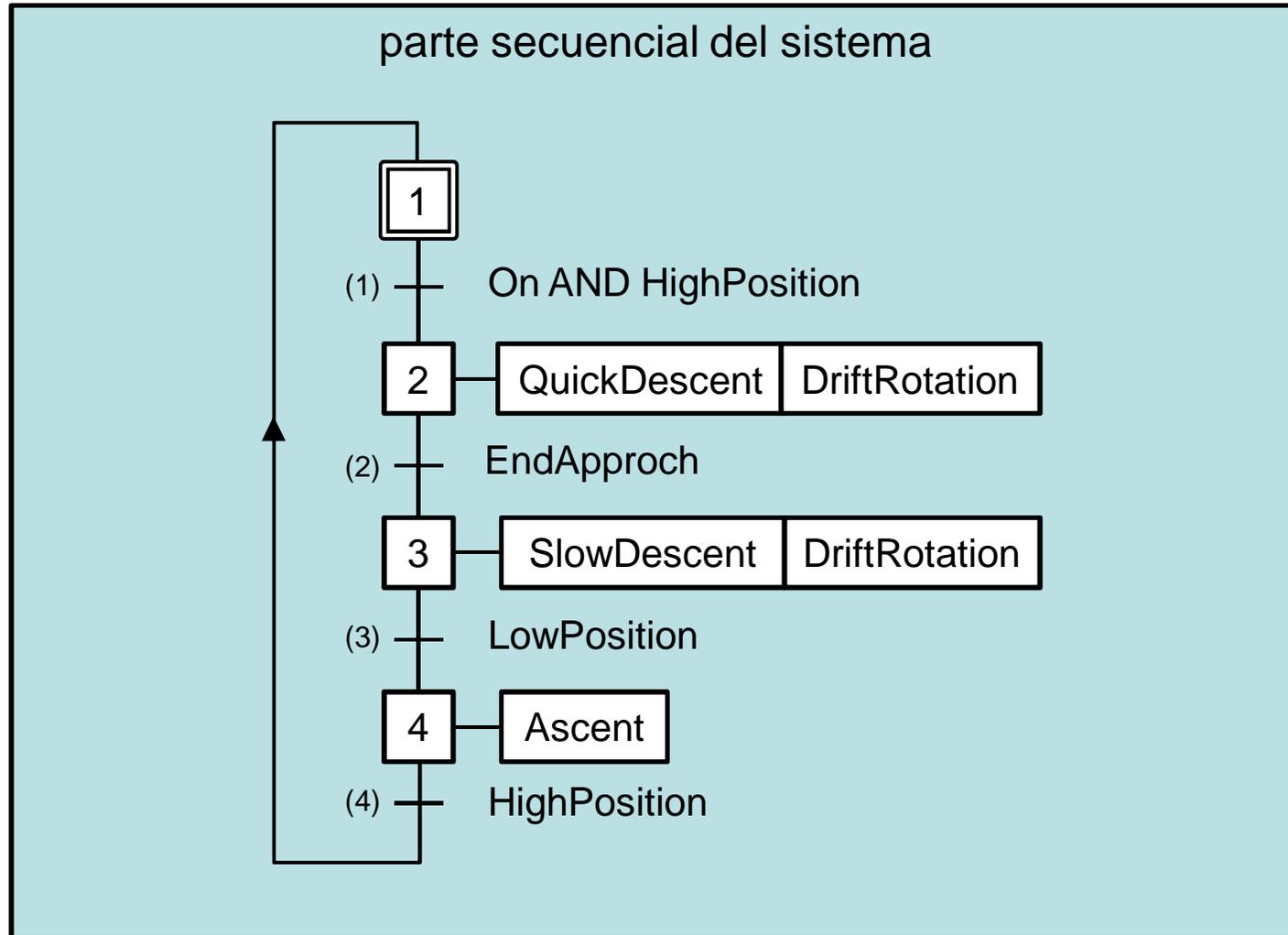
# Principios generales II

---

## Representación I

- Parte del GRAFCET que conecta las variables de entrada y la estructura a través de las condiciones de transición, y las variables de salida y la estructura a través de las acciones
- Interpretación:
  - Condiciones de transición
  - Acciones

## Representación II



# Principios generales IV

---

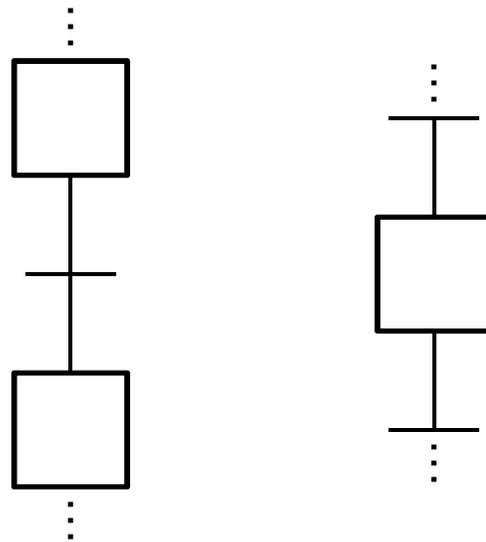
## Regla sintáctica

- La alternancia etapa transición y transición etapa se debe respetar siempre, sea cual sea la secuencia.

# Principios generales IV

## Regla sintáctica

- La alternancia etapa transición y transición etapa se debe respetar siempre, sea cual sea la secuencia.

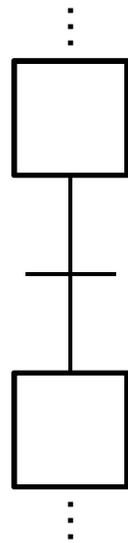


correcto

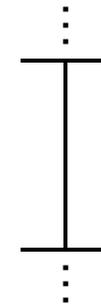
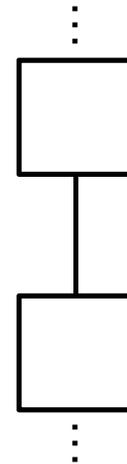
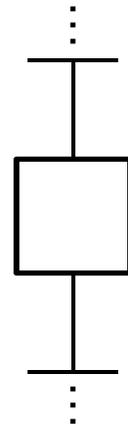
# Principios generales IV

## Regla sintáctica

- La alternancia etapa transición y transición etapa se debe respetar siempre, sea cual sea la secuencia.



correcto



incorrecto

# Principios generales V

---

## Reglas de evolución I

- **Regla 1:** la situación inicial, elegida por el diseñador, es la situación en el instante inicial.
- **Regla 2:** se dice que una transición está habilitada cuando todas las etapas anteriores conectadas a esa transición están activas. El disparo de la transición ocurre cuando la transición está habilitada y cuando la condición asociada a la transición es cierta.

# Principios generales VI

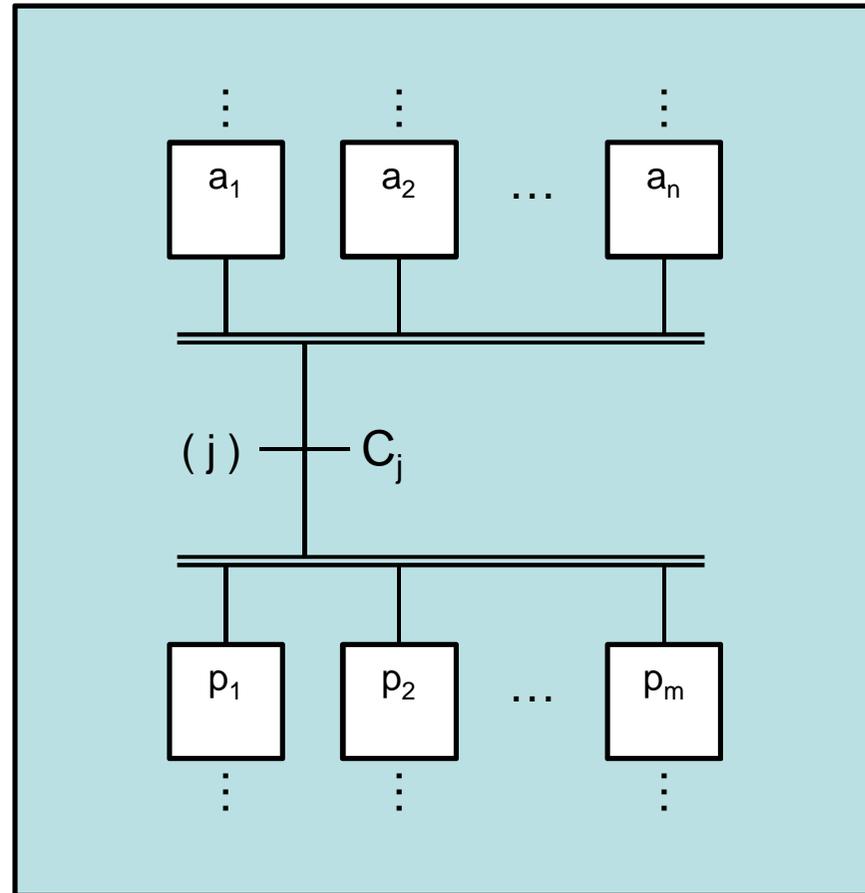
---

## Reglas de evolución II

- **Regla 3:** el disparo de una transición provoca simultáneamente la activación de todas las etapas posteriores y la desactivación de todas las etapas anteriores conectadas a esa transición.
- **Regla 4:** si varias transiciones pueden ser disparadas simultáneamente, entonces son disparadas simultáneamente.
- **Regla 5:** si durante la evolución, una etapa activa es simultáneamente activada y desactivada, entonces permanece activa.

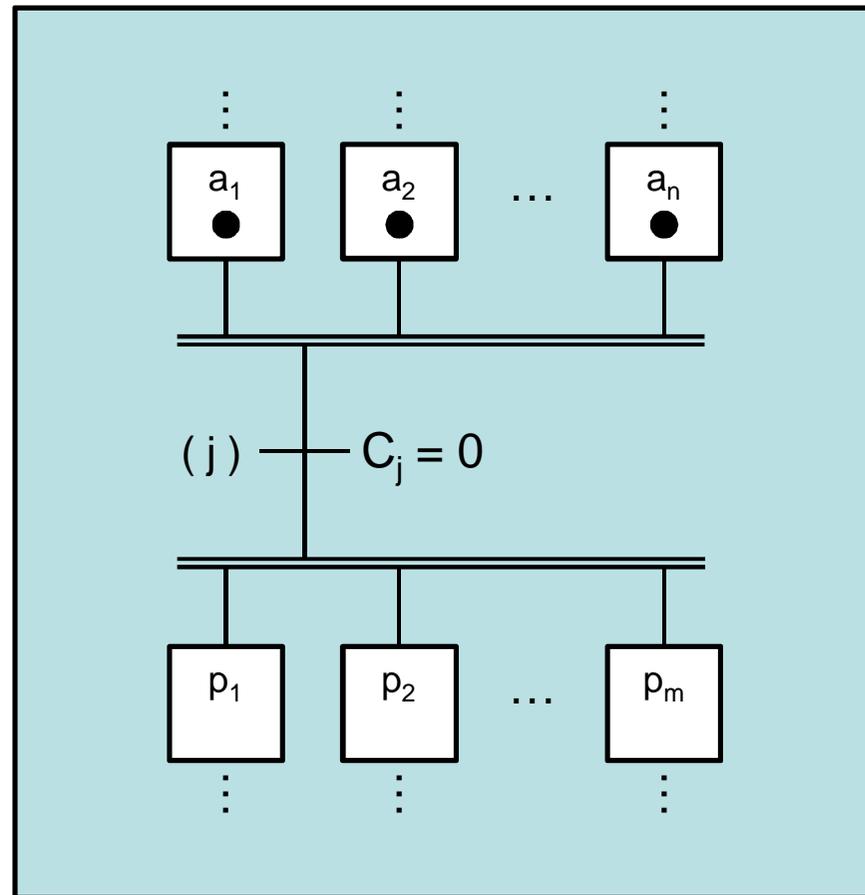
# Principios generales VII

## Disparo de una transición



# Principios generales VII

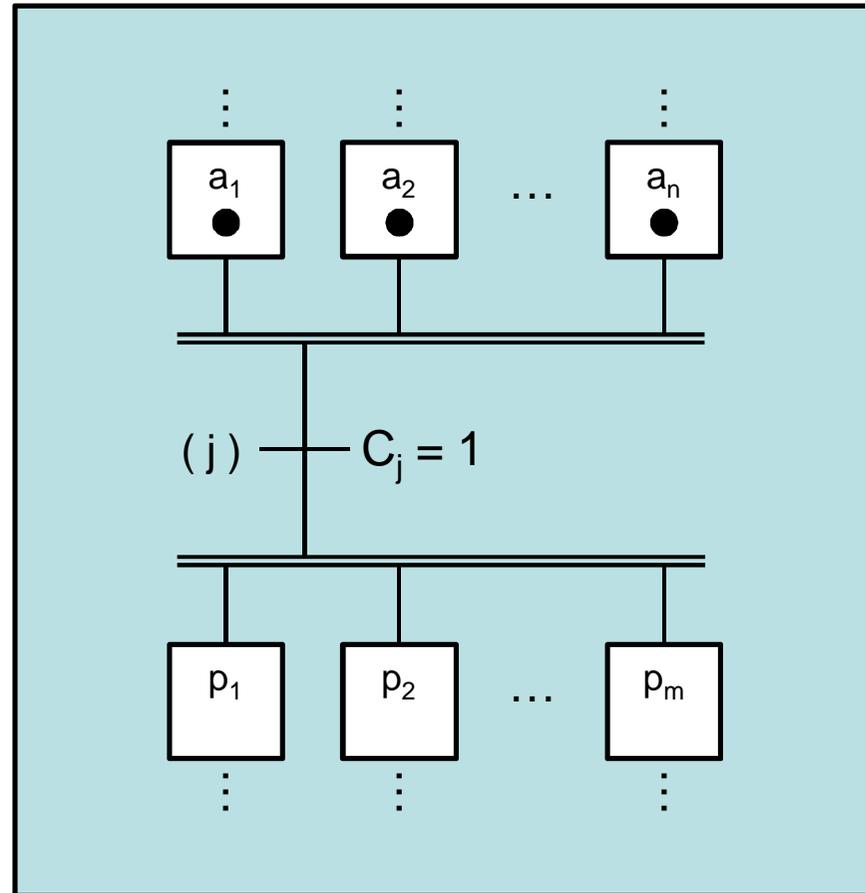
## Disparo de una transición



transición habilitada y condición falsa

# Principios generales VII

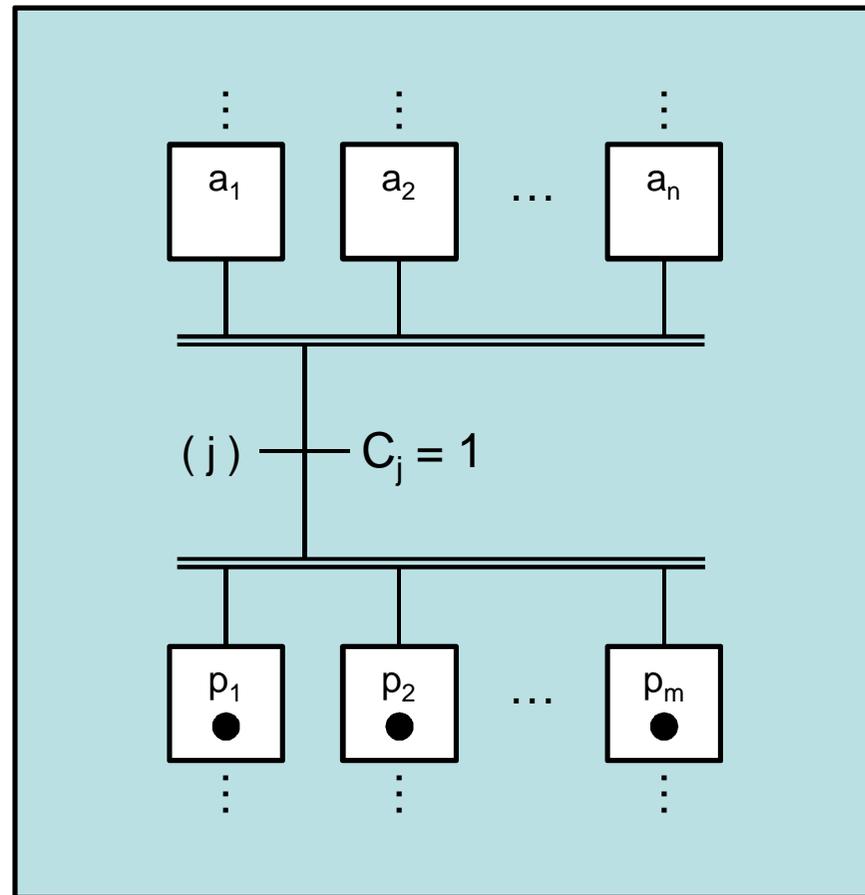
## Disparo de una transición



transición habilitada y condición cierta

# Principios generales VII

## Disparo de una transición



disparo de la transición

# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

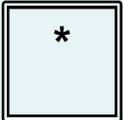
9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

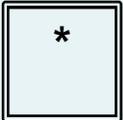
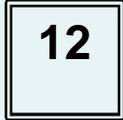
# Símbolos básicos I

## Etapas

n.	símbolo	nombre y ejemplo
[1]		<u>etapa</u> 
[2]	X*	<u>variable de etapa</u> X8
[3]		<u>etapa inicial</u> 

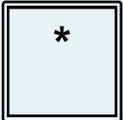
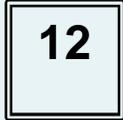
# Símbolos básicos I

## Etapas

n.	símbolo	nombre y ejemplo
[1]		<u>etapa</u>  marca (opcional)
[2]	X*	<u>variable de etapa</u> X8
[3]		<u>etapa inicial</u> 

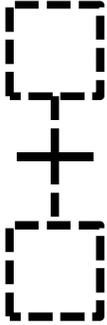
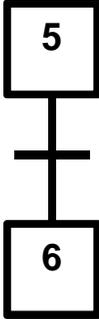
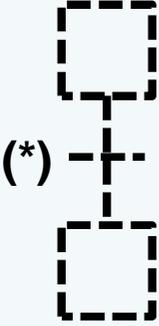
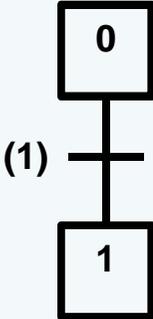
# Símbolos básicos I

## Etapas

n.	símbolo	nombre y ejemplo
[1]		<u>etapa</u> 
[2]	X*	<u>variable de etapa</u> X8
[3]		<u>etapa inicial</u> 

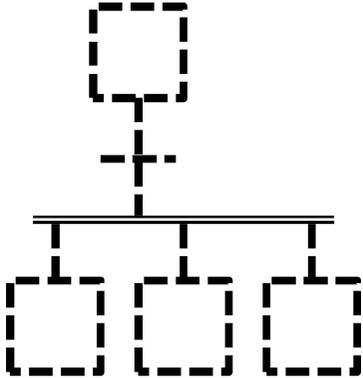
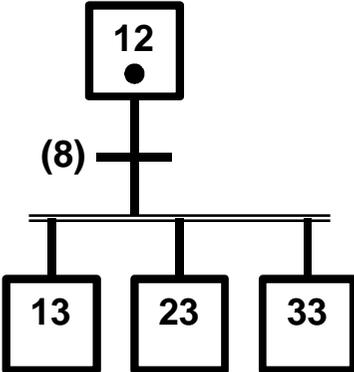
# Símbolos básicos II

## Transiciones I

n.	símbolo	nombre y ejemplo
[7]		<u>transición</u> 
[8]		<u>denominación de transición</u> 

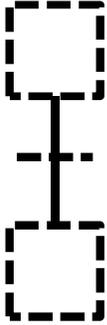
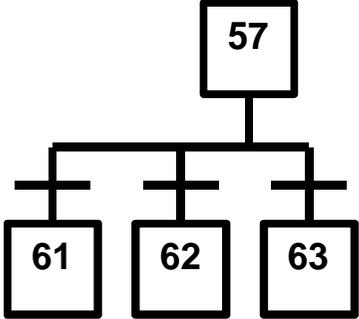
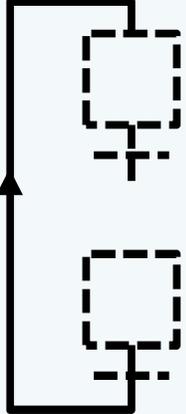
# Símbolos básicos III

## Transiciones II

n.	símbolo	nombre y ejemplo
[9]		<p><u>sincronización</u></p> 

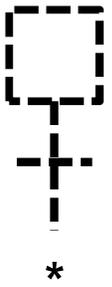
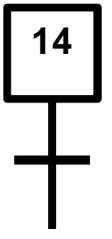
# Símbolos básicos IV

## Enlaces directos I

n.	símbolo	nombre y ejemplo
[10]		<u>de arriba a abajo</u> 
[11]		<u>de abajo a arriba</u>

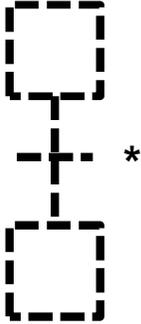
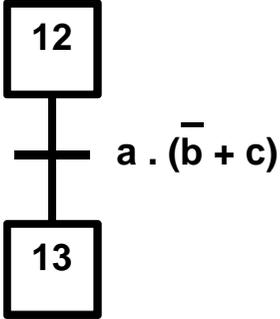
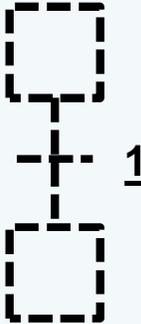
# Símbolos básicos V

## Enlaces directos II

n.	símbolo	nombre y ejemplo
[12]		<p><u>etiqueta</u></p>  <p>etapa 83 página 13</p>

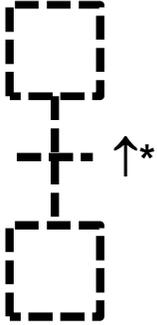
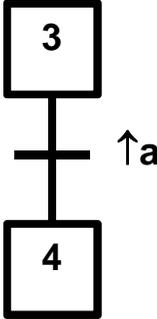
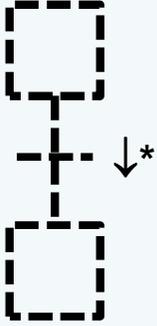
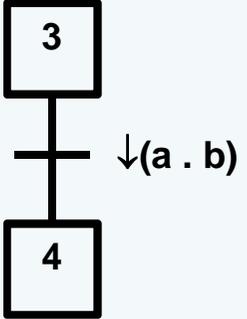
# Símbolos básicos VI

## Condiciones de transición I

n.	símbolo	nombre y ejemplo
[13]		<p><u>condición de transición</u></p> 
[14]		<p><u>siempre cierto</u></p>

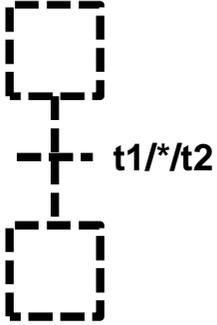
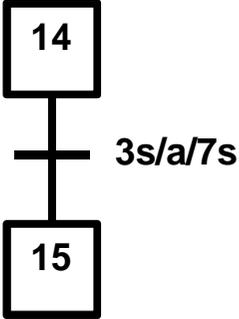
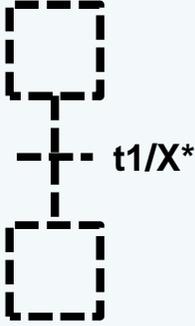
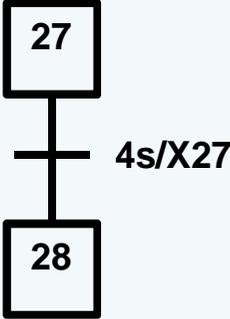
# Símbolos básicos VII

## Condiciones de transición II

n.	símbolo	nombre y ejemplo
[15]		<p><u>flanco de subida</u></p> 
[16]		<p><u>flanco de bajada</u></p> 

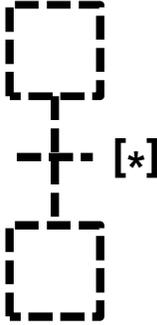
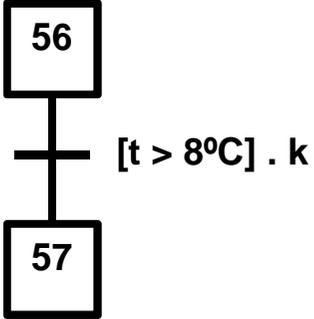
# Símbolos básicos VIII

## Condiciones de transición III

n.	símbolo	nombre y ejemplo
[17]		<p><u>dependiente del tiempo</u></p> 
[18]		<p><u>simplificación</u></p> 

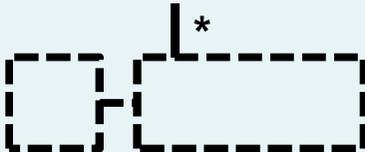
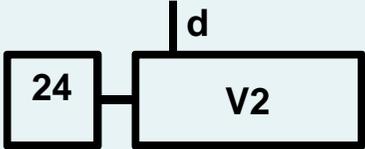
# Símbolos básicos IX

## Condiciones de transición IV

n.	símbolo	nombre y ejemplo
[19]		<p><u>predicado</u></p> 

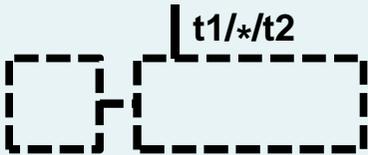
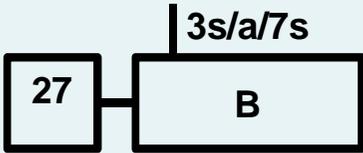
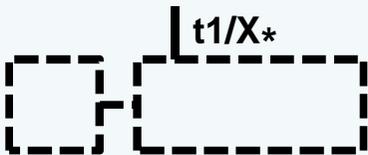
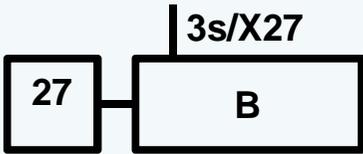
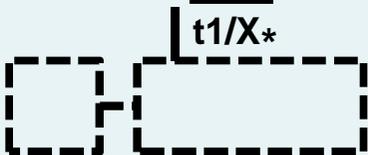
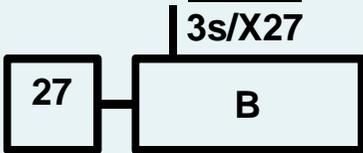
# Símbolos básicos X

## Acciones continuas I

n.	símbolo	nombre y ejemplo
[20]		<u>acción continua</u>
[21]		<u>nombre de salida</u> 
[22]		<u>acción condicionada</u> 

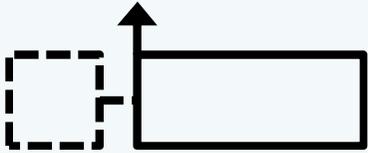
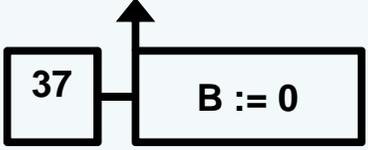
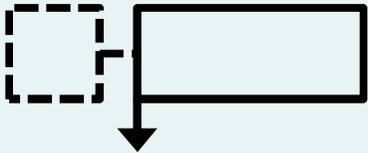
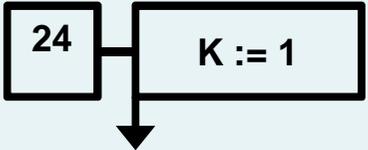
# Símbolos básicos XI

## Acciones continuas II

n.	símbolo	nombre y ejemplo
[23]		<p><u>acción temporizada</u></p> 
[24]		<p><u>acción retardada</u></p> 
[25]		<p><u>acción limitada</u></p> 

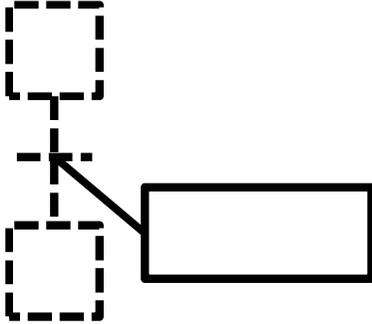
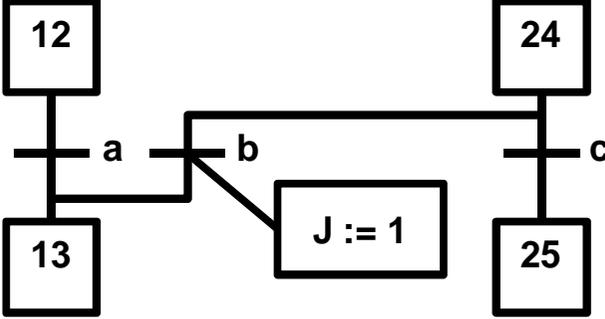
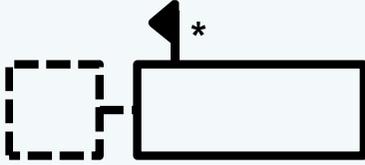
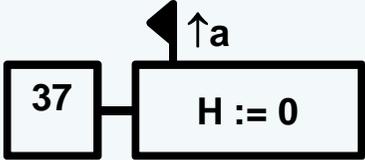
# Símbolos básicos XII

## Acciones memorizadas I

n.	símbolo	nombre y ejemplo
[26]		<u>almacenamiento</u> 
[27]		<u>acción a la activación</u> 
[28]		<u>acción a la desactivación</u> 

# Símbolos básicos XIII

## Acciones memorizadas II

n.	símbolo	nombre y ejemplo
[29]		<p><u>acción al disparo</u></p> 
[30]		<p><u>acción al evento</u></p> 

# Símbolos básicos XIV

## Comentarios

n.	símbolo	nombre y ejemplo
[31]	“*”	<u>comentario</u> <span style="border: 1px solid black; padding: 2px;">45</span> “etapa de espera”

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

**9.4. Estructuras básicas**

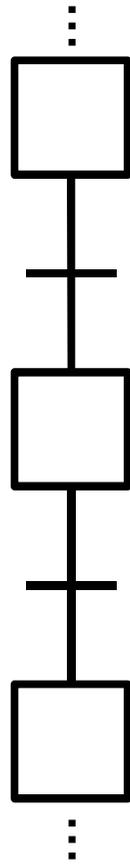
9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

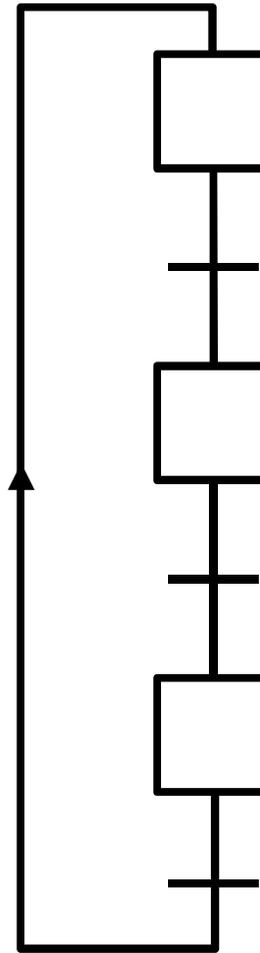
9.7. Ejemplo: doble carro va y viene

# Estructuras básicas I

## Secuencia y ciclo



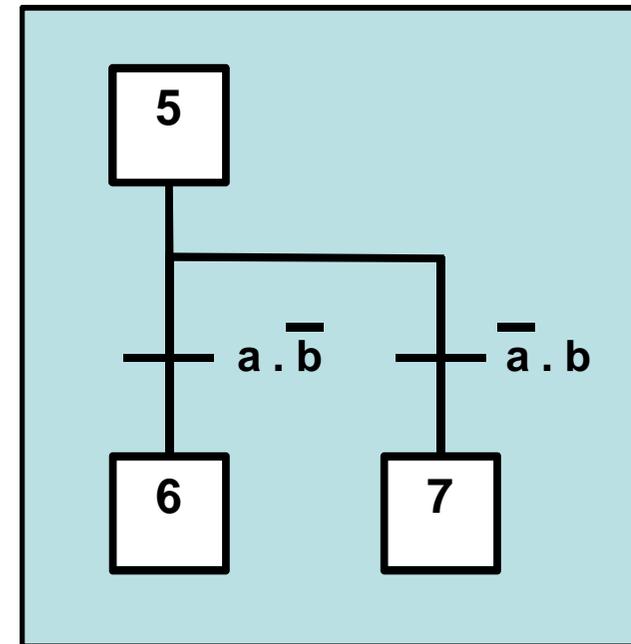
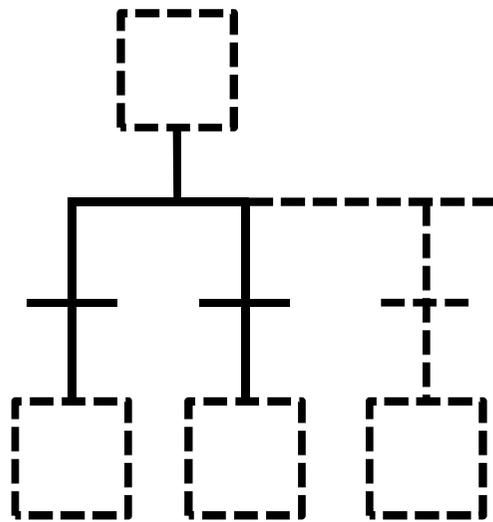
secuencia



ciclo

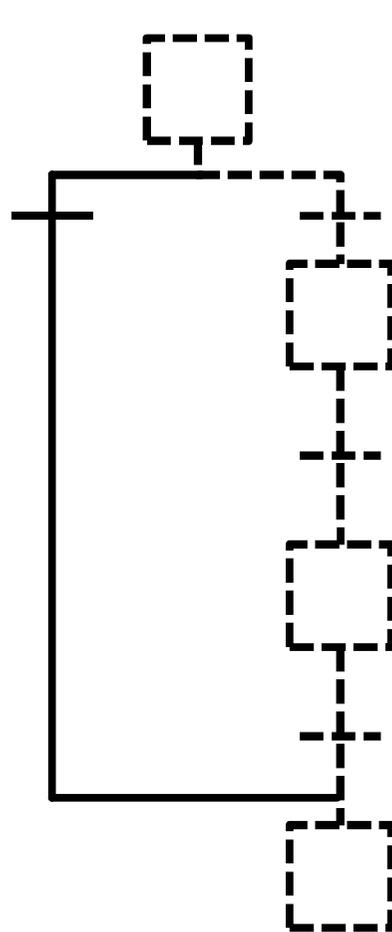
# Estructuras básicas II

## Selección de secuencias

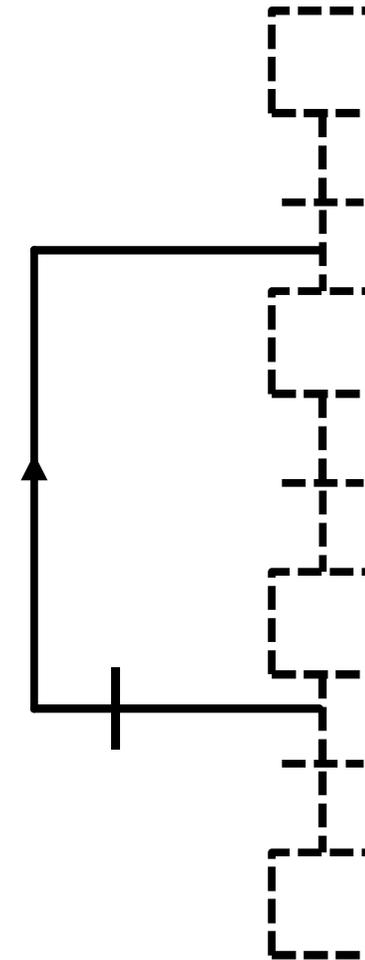


# Estructuras básicas III

## Salto de etapas



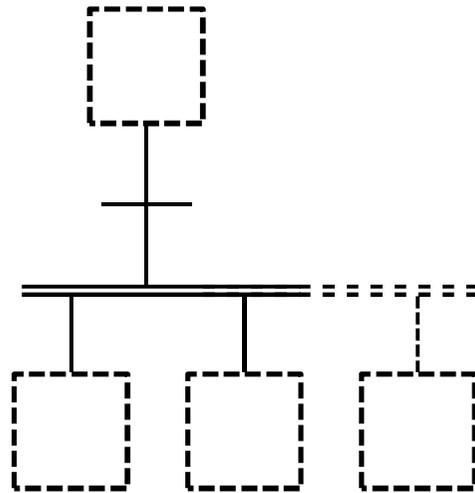
adelante



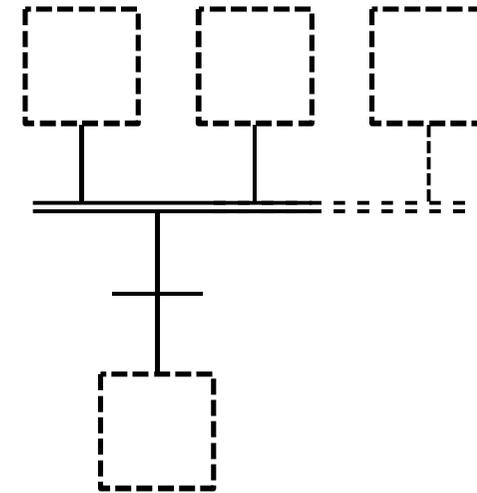
atrás

# Estructuras básicas IV

## Secuencias paralelas



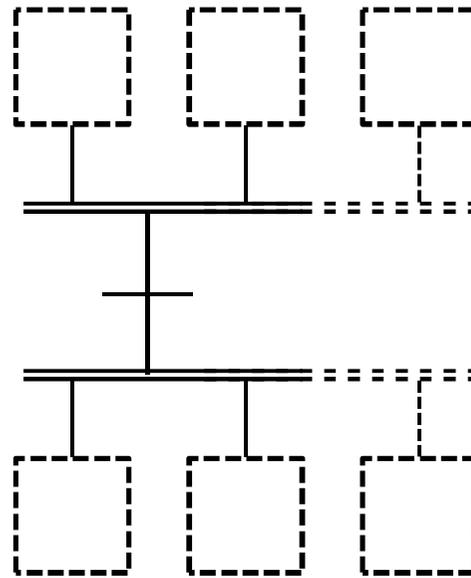
activación



sincronización

# Estructuras básicas V

## Sincronización y activación de secuencias paralelas



## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

**9.5. Diseño de automatismos con GRAFCET**

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

# Diseño con GRAFCET I

---

## Metodología

1. Identificar y nominar las entradas y salidas
2. Identificar la situación inicial
3. Determinar el evento esperado
4. Identificar la nueva situación
5. Asociar lista de acciones
6. Repetir 3 a 5 hasta cerrar todos los ciclos
7. Resolver conflictos

# Diseño con GRAFCET II

---

## Directrices

- Construcción progresiva
- Seguir la secuencia natural
- Orden
  - Secuencia principal
  - Secuencias secundarias o extraordinarias
- Etapas con sentido físico
- Evitar etapas transitorias

# Diseño con GRAFCET III

---

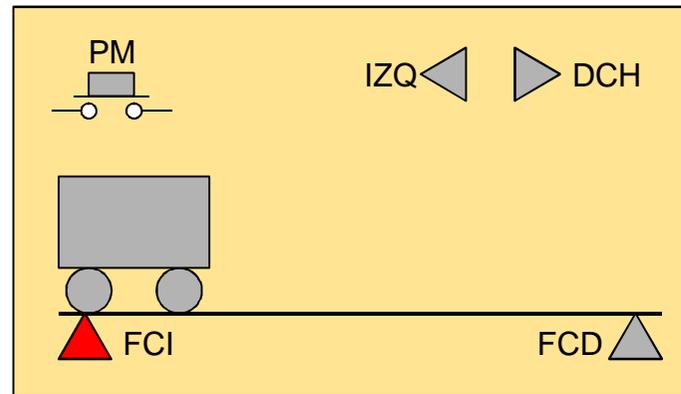
## Ejemplo: carro va y viene

Un carro dispone de señales de mando para activar su movimiento a izquierda (IZQ) y a derecha (DCH), sensores de final de carrera izquierda (FCI) y derecha (FCD) y un pulsador de marcha (PM).

Al accionarse el pulsador de marcha, si el carro se encuentra en el extremo izquierdo de la vía, debe realizar un recorrido de ida al extremo derecho y vuelta. Si al finalizar el recorrido de vuelta el pulsador de marcha está accionado, el ciclo se debe repetir.

# Diseño con GRAFCET IV

## Ejemplo: carro va y viene



$E = \{ PM, FCI, FCD \}$

$S = \{ DCH, IZQ \}$

entradas y salidas

# Diseño con GRAFCET I

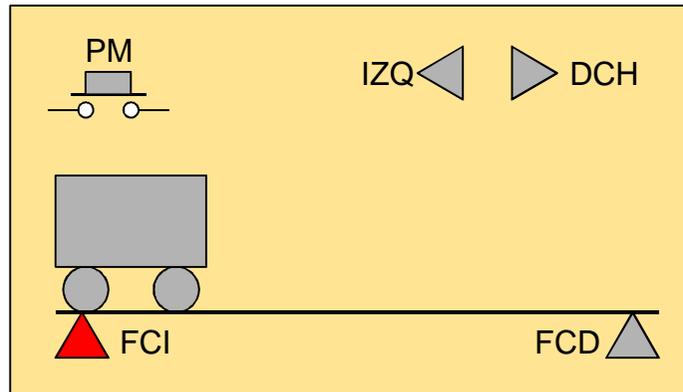
---

## Metodología

1. Identificar y nominar las entradas y salidas
2. Identificar la situación inicial
3. Determinar el evento esperado
4. Identificar la nueva situación
5. Asociar lista de acciones
6. Repetir 3 a 5 hasta cerrar todos los ciclos
7. Resolver conflictos

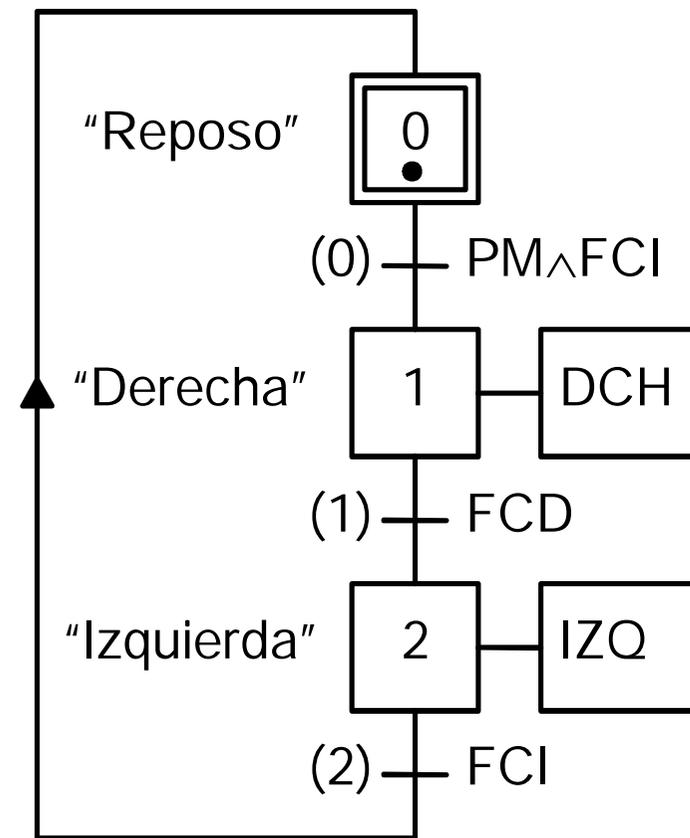
# Diseño con GRAFCET IV

## Ejemplo: carro va y viene



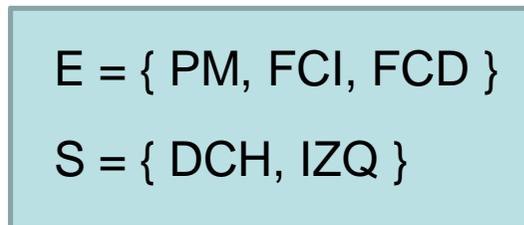
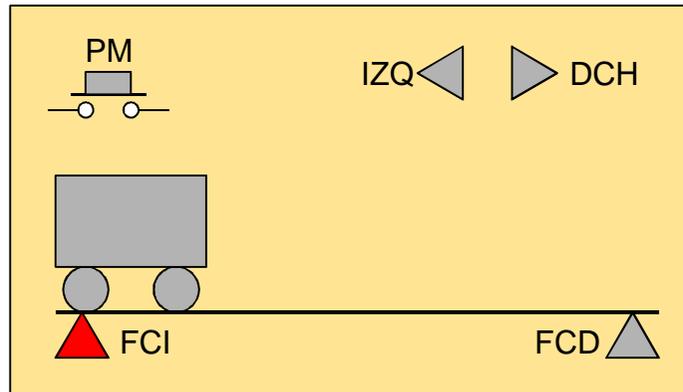
$E = \{ PM, FCI, FCD \}$   
 $S = \{ DCH, IZQ \}$

entradas y salidas

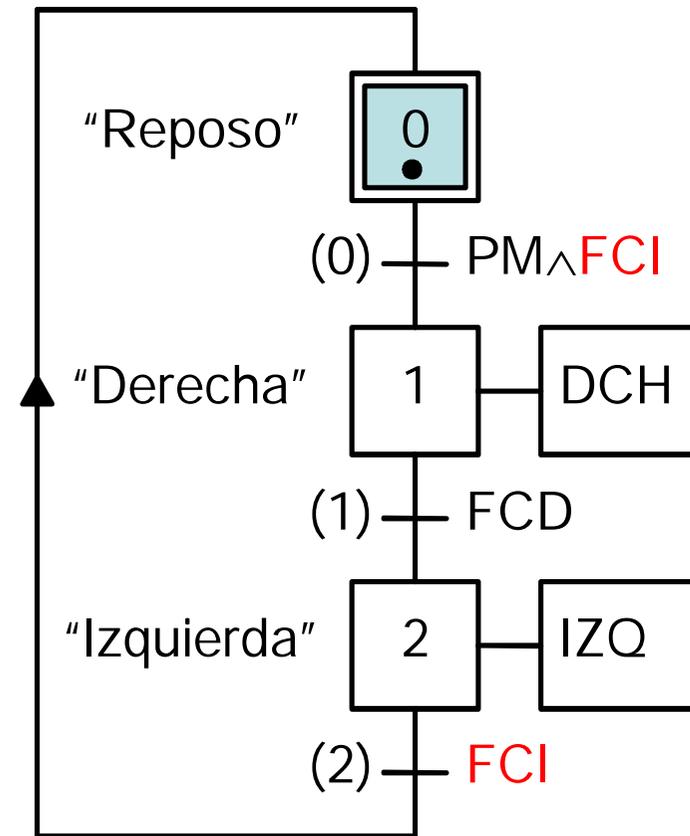


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

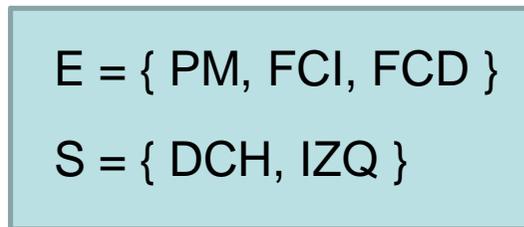
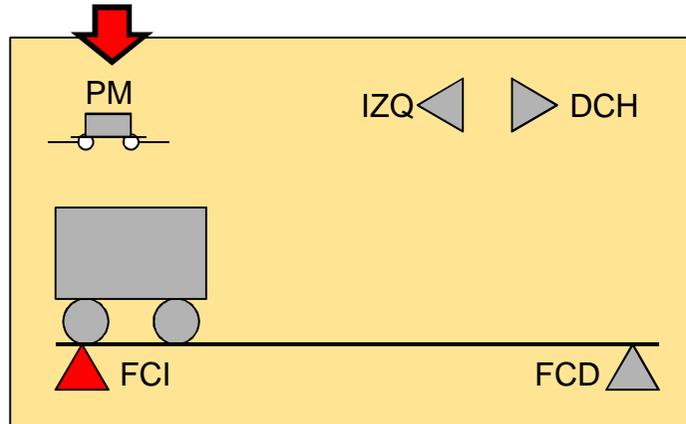


entradas y salidas

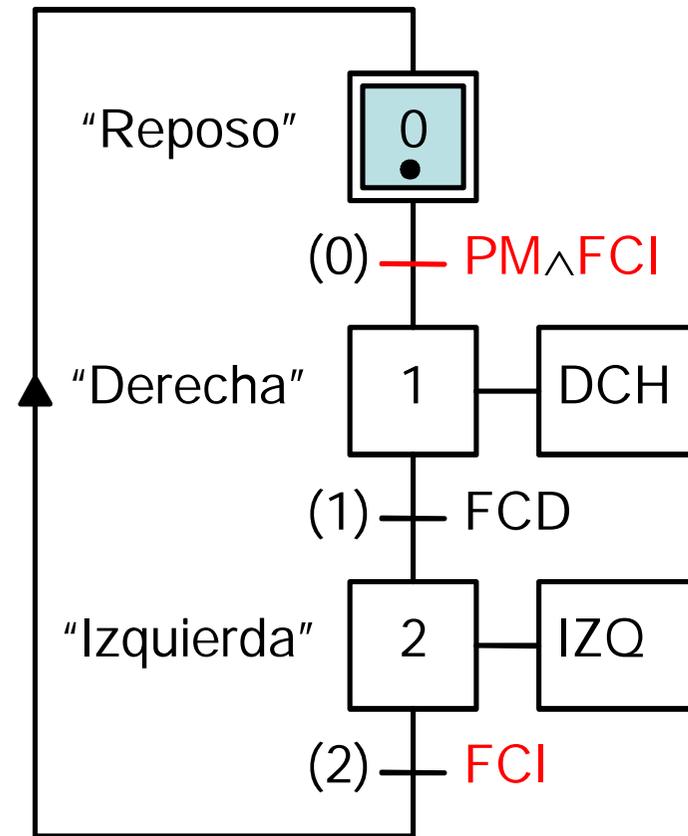


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

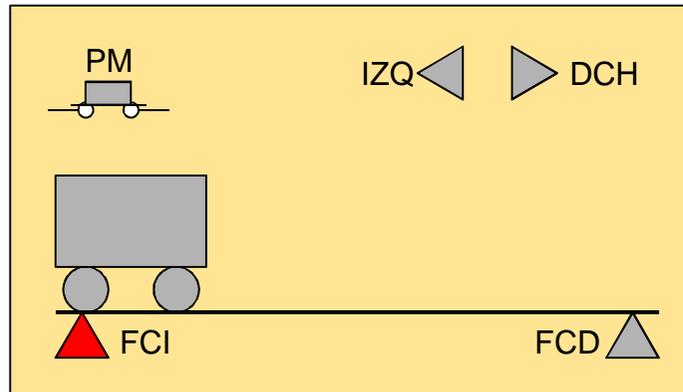


entradas y salidas



# Diseño con GRAFCET IV

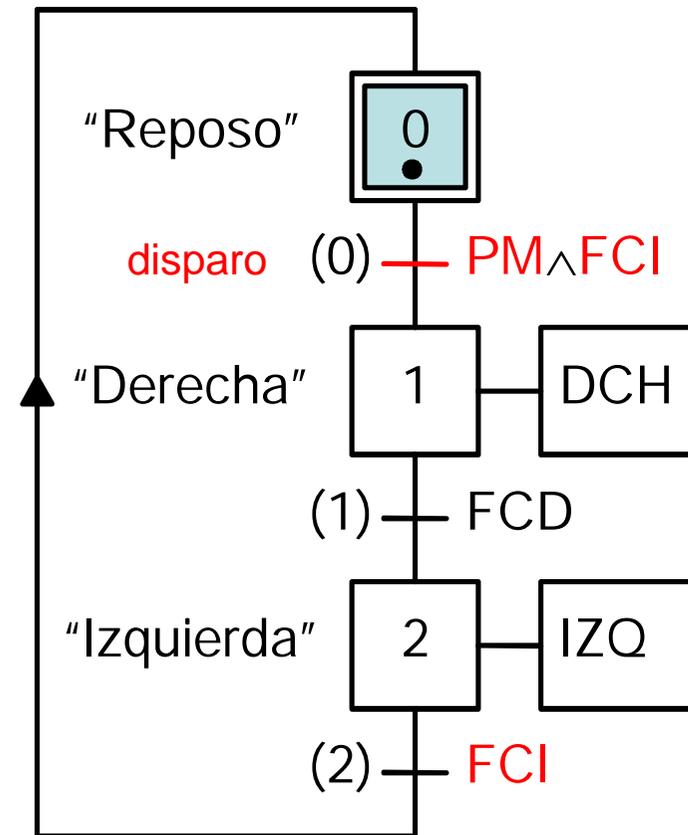
## Funcionamiento: carro va y viene



$E = \{ PM, FCI, FCD \}$

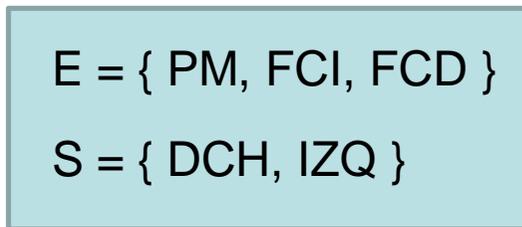
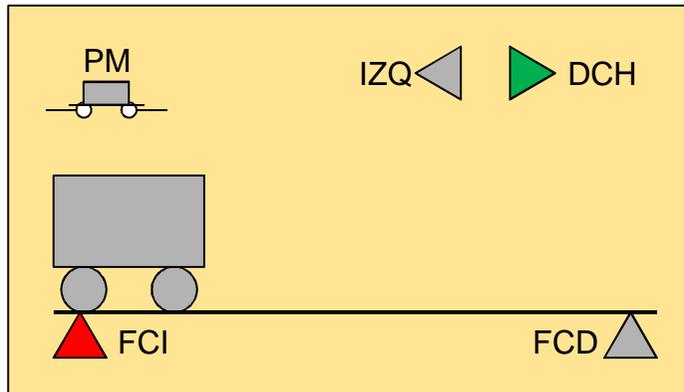
$S = \{ DCH, IZQ \}$

entradas y salidas

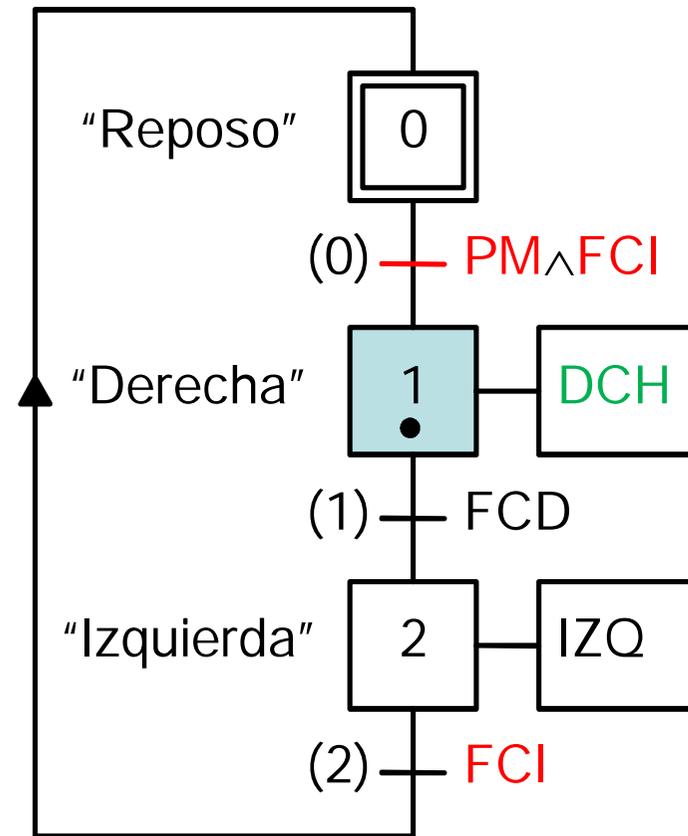


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

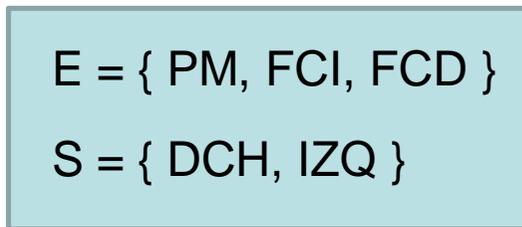
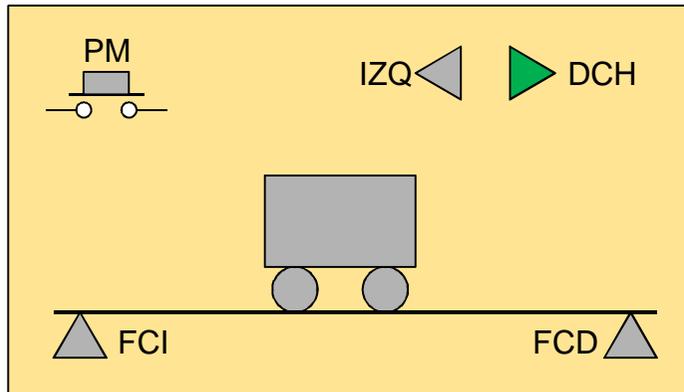


entradas y salidas

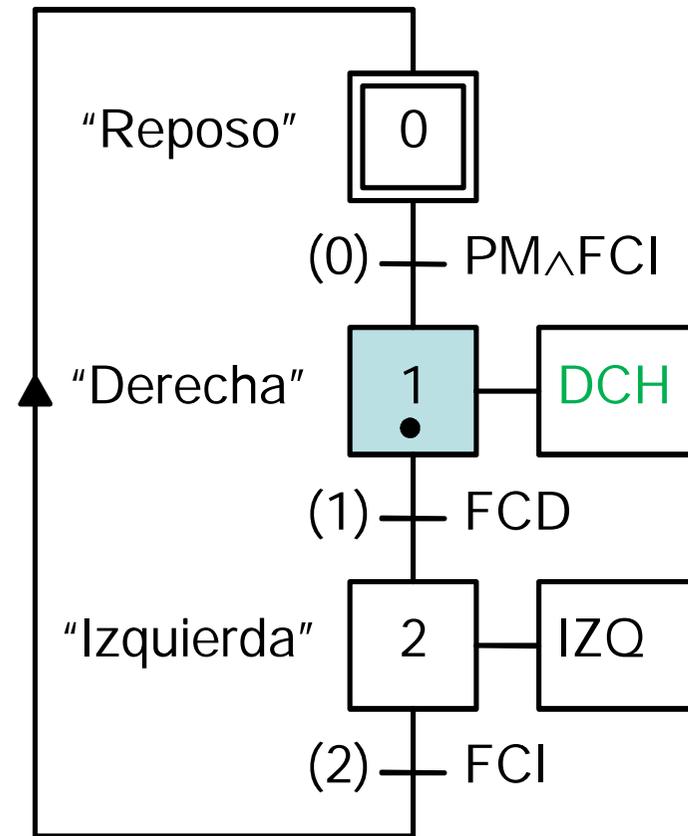


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

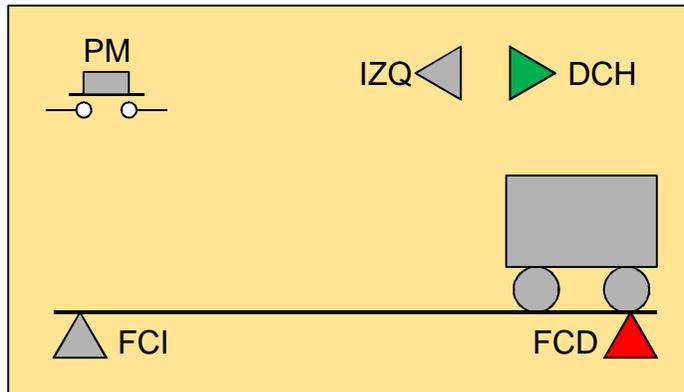


entradas y salidas



# Diseño con GRAFCET IV

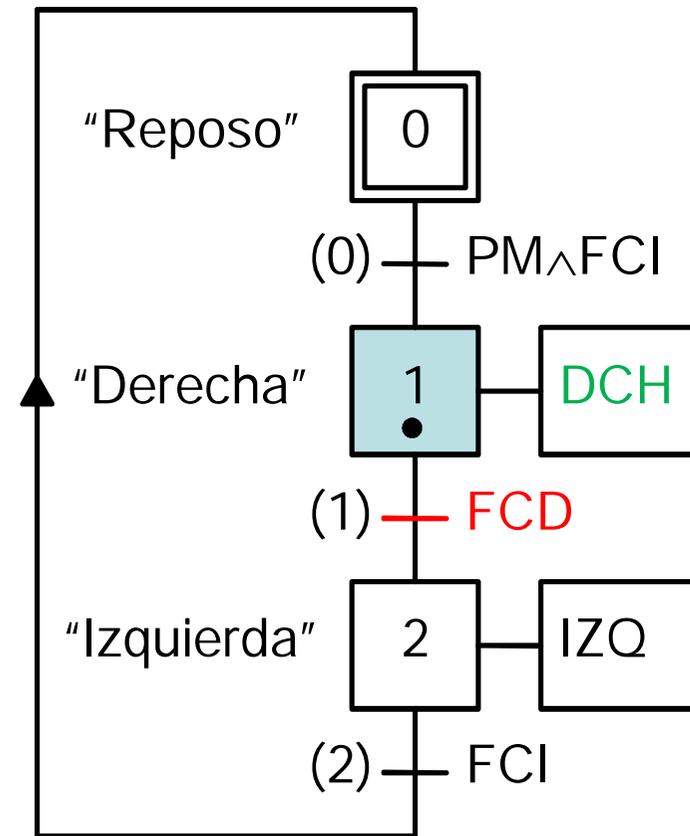
## Funcionamiento: carro va y viene



$E = \{ PM, FCI, FCD \}$

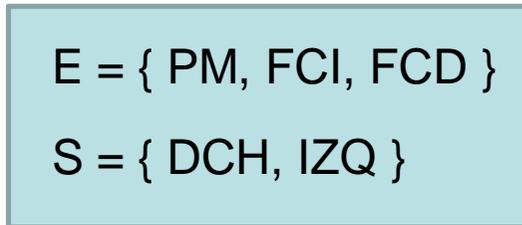
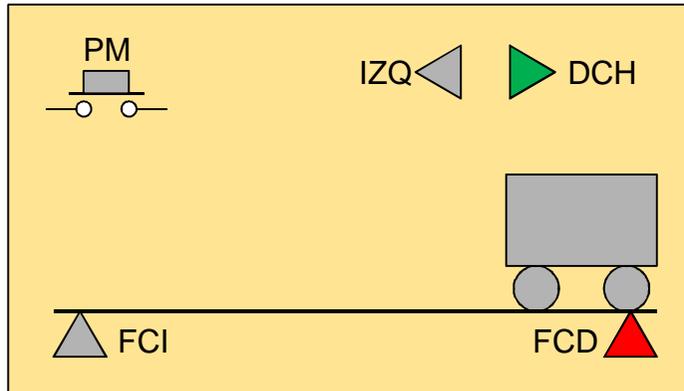
$S = \{ DCH, IZQ \}$

entradas y salidas

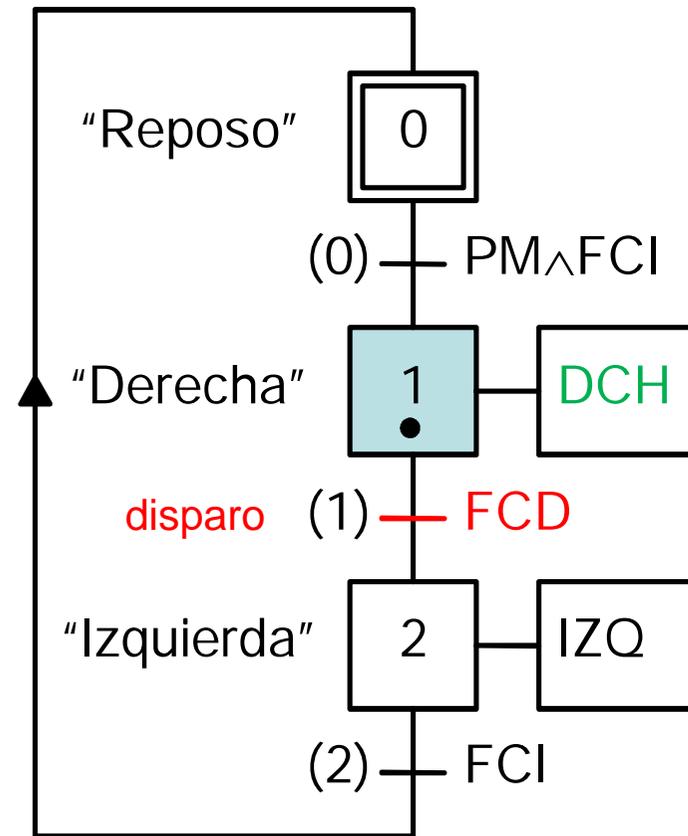


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

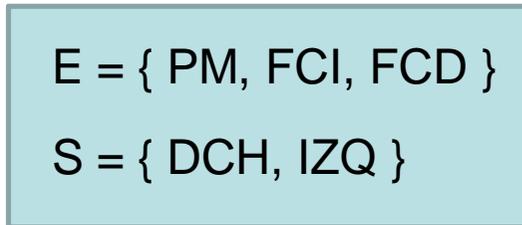
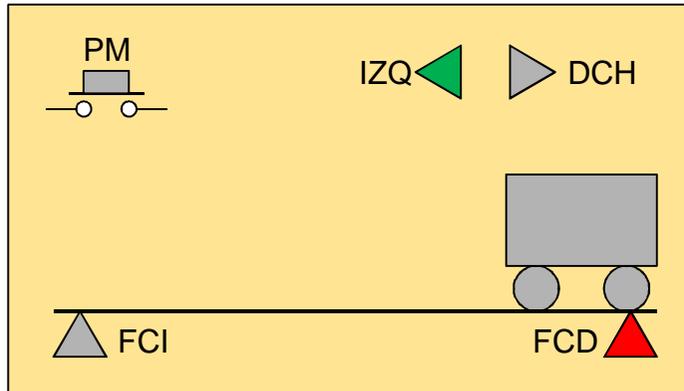


entradas y salidas

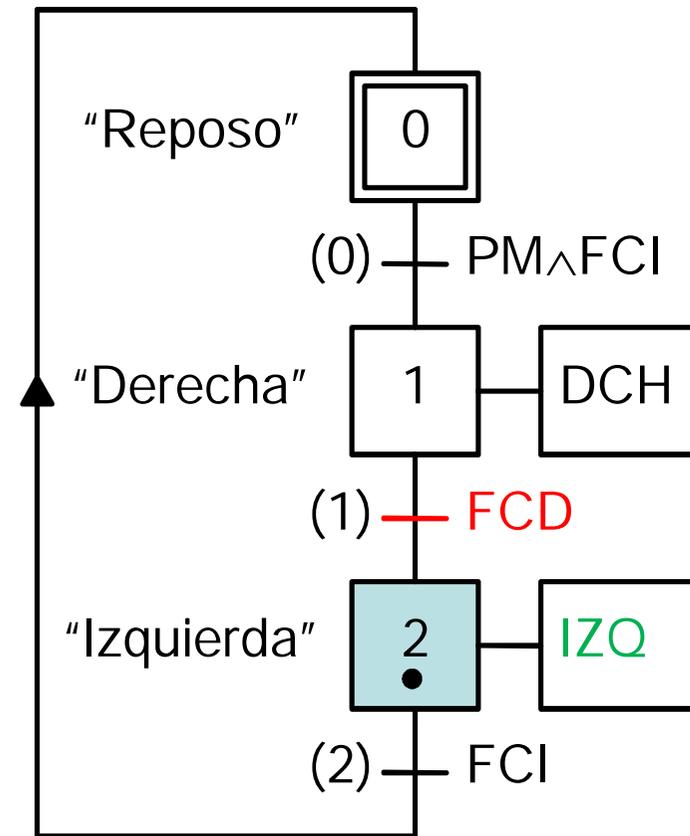


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

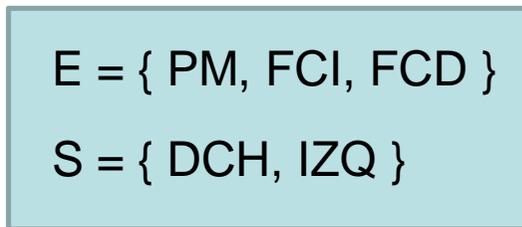
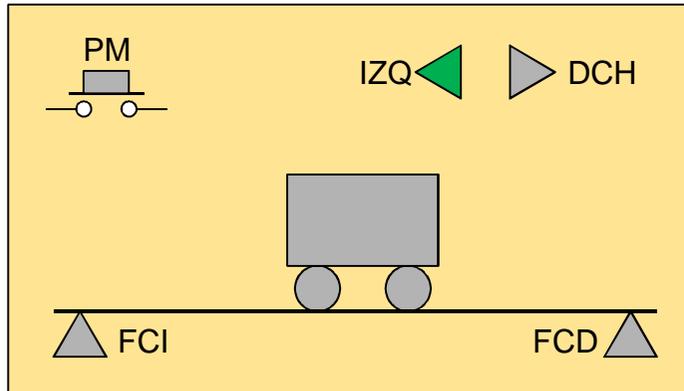


entradas y salidas

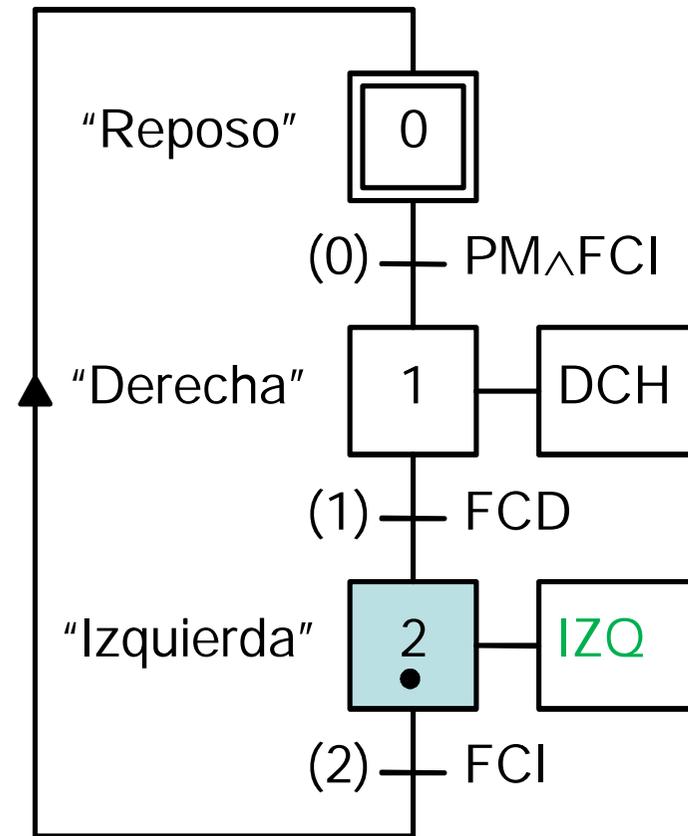


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

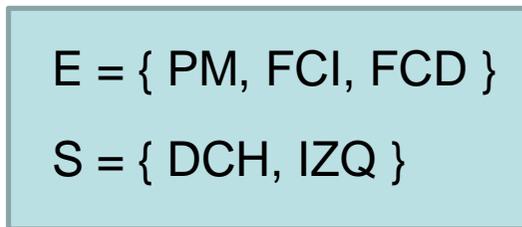
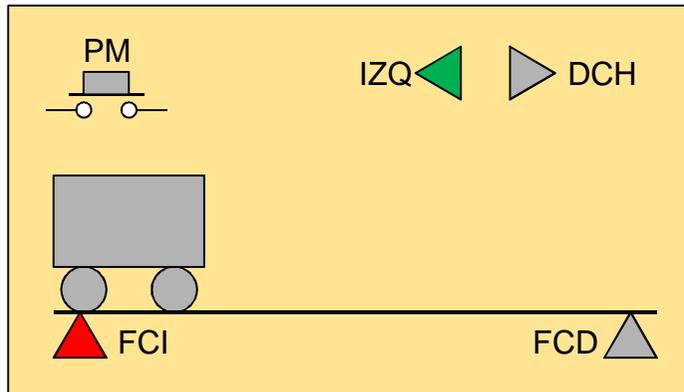


entradas y salidas

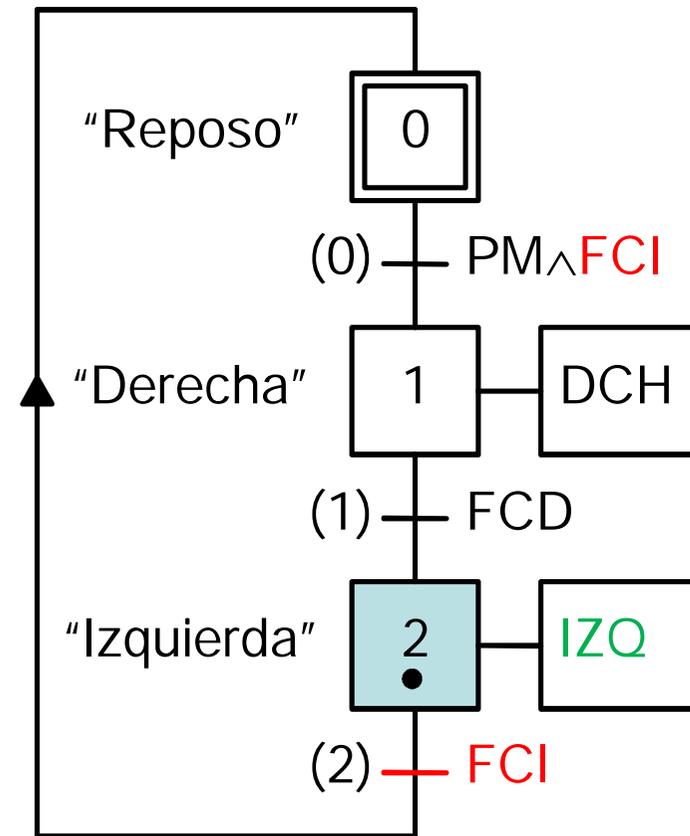


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

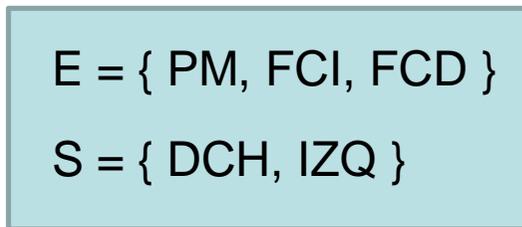
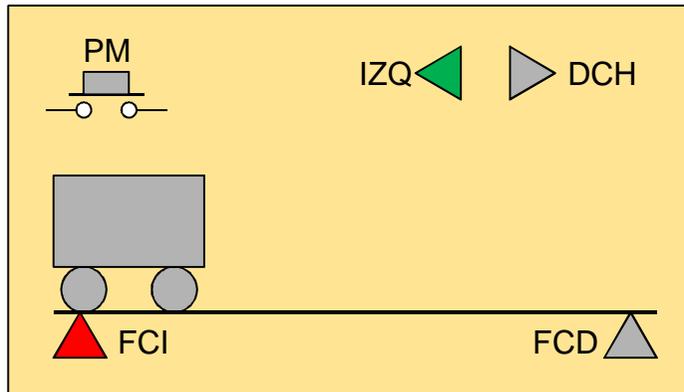


entradas y salidas

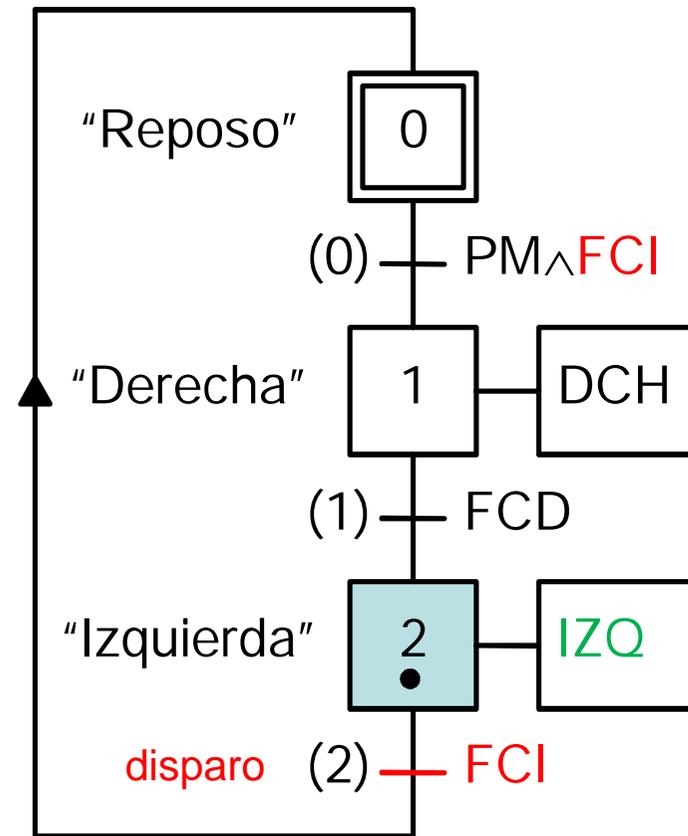


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

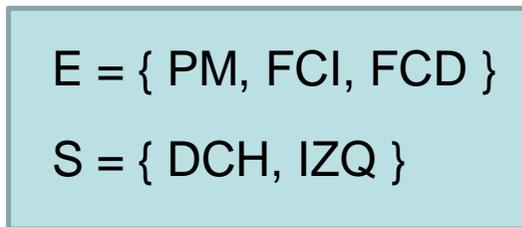
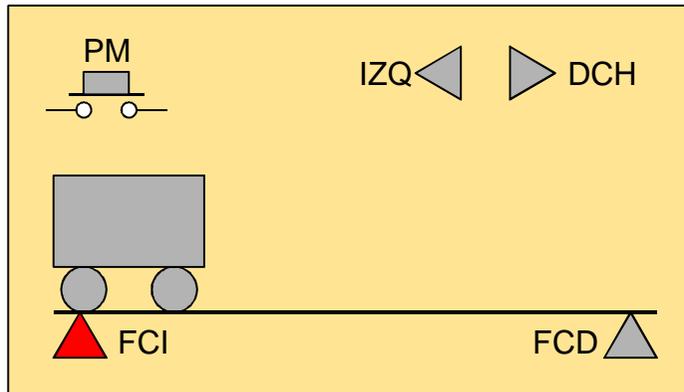


entradas y salidas

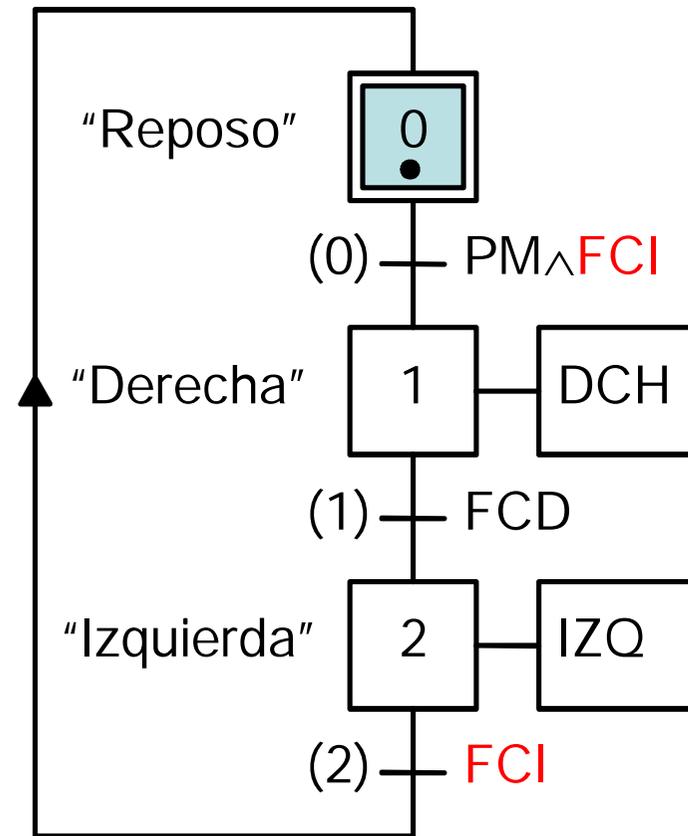


# Diseño con GRAFCET IV

## Funcionamiento: carro va y viene

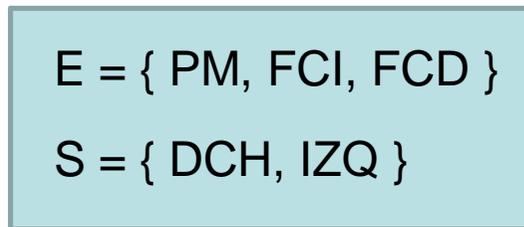
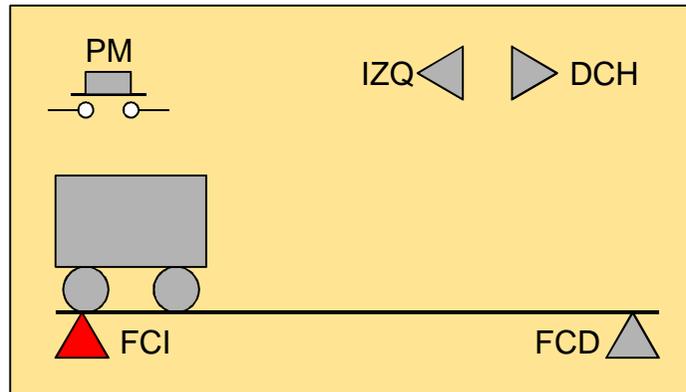


entradas y salidas

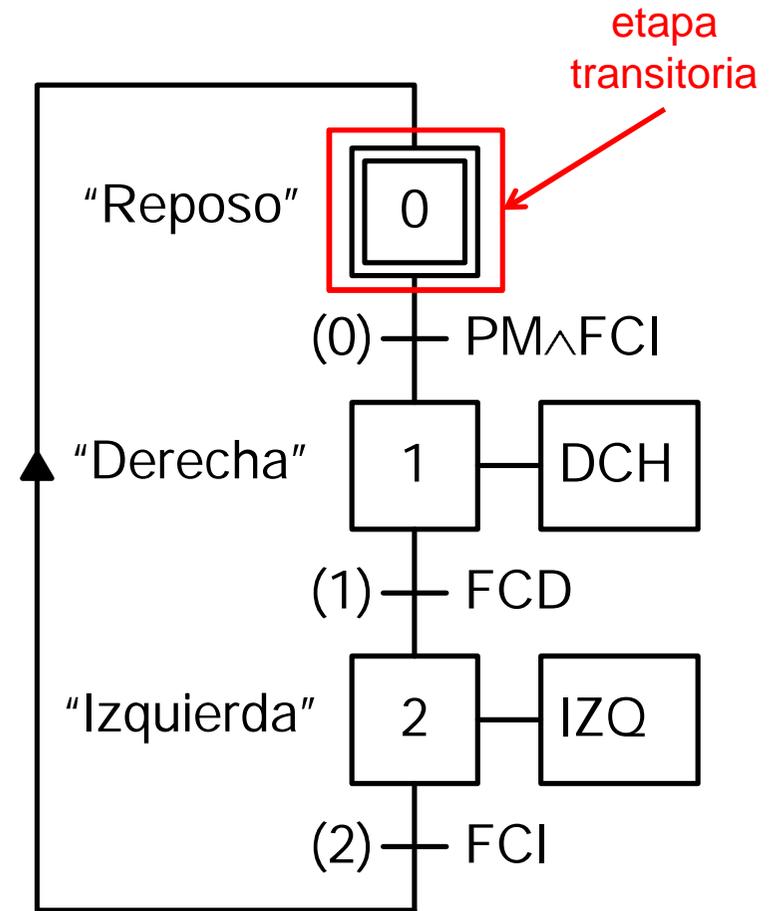


# Diseño con GRAFCET IV

## Ejemplo: carro va y viene



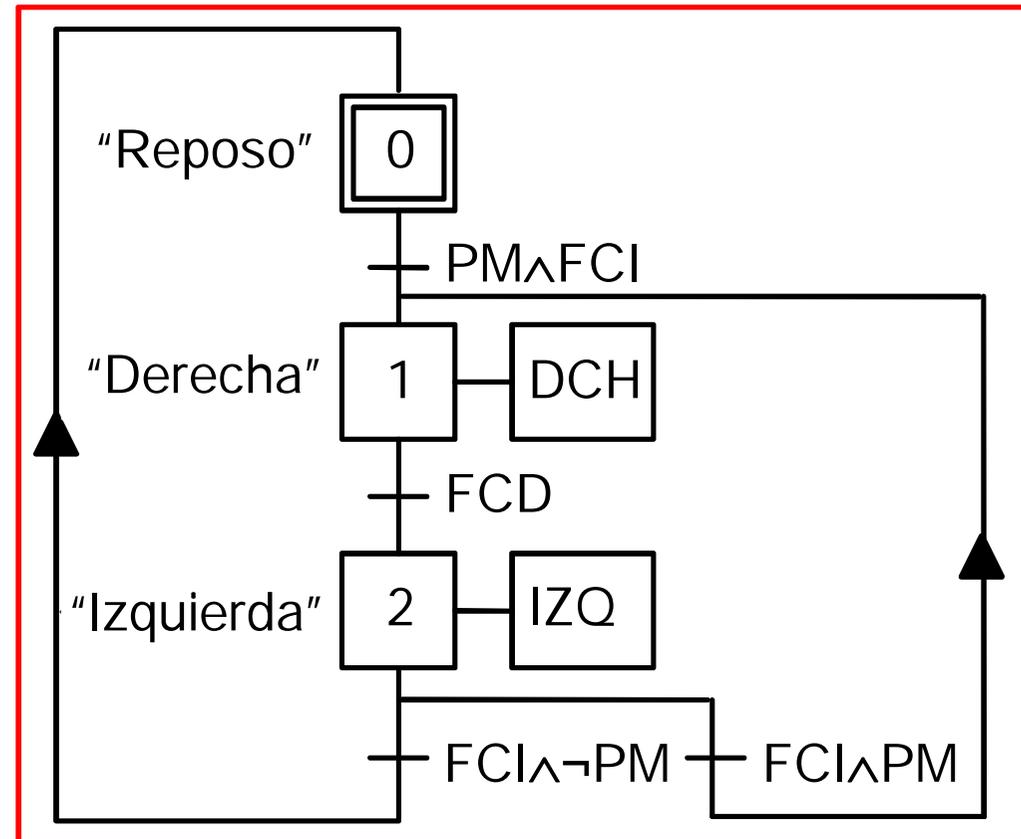
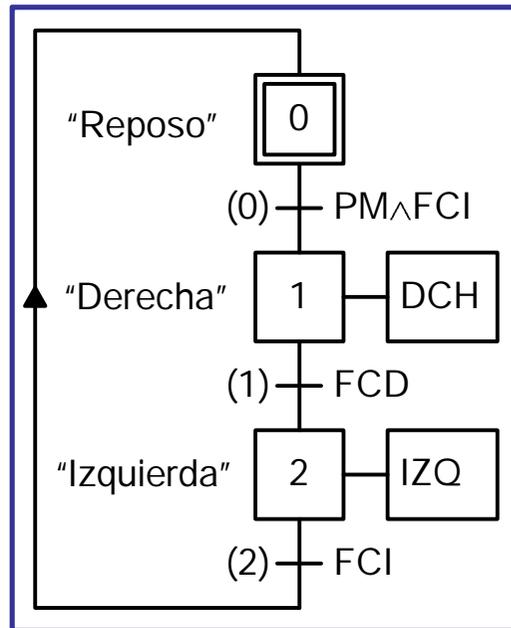
entradas y salidas



# Diseño con GRAFCET V

## Ejemplo: carro va y viene

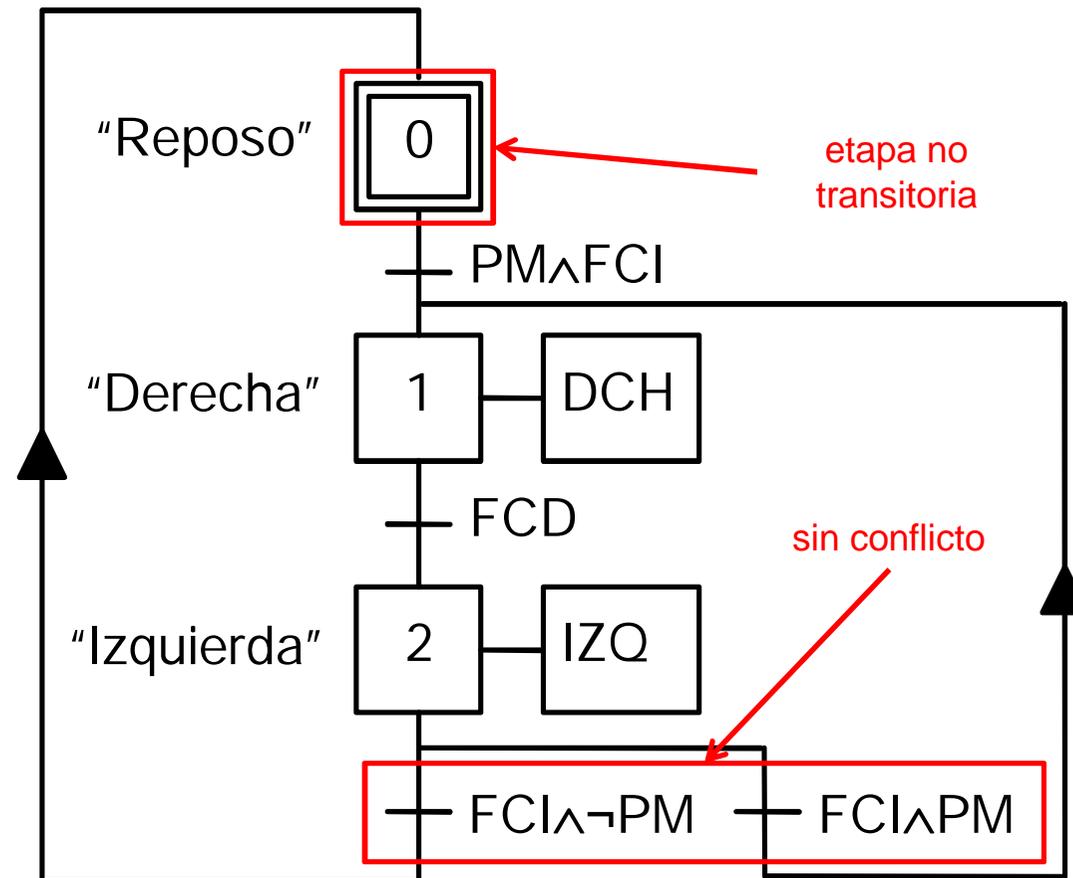
### Eliminación etapas transitorias



# Diseño con GRAFCET VI

## Ejemplo: carro va y viene

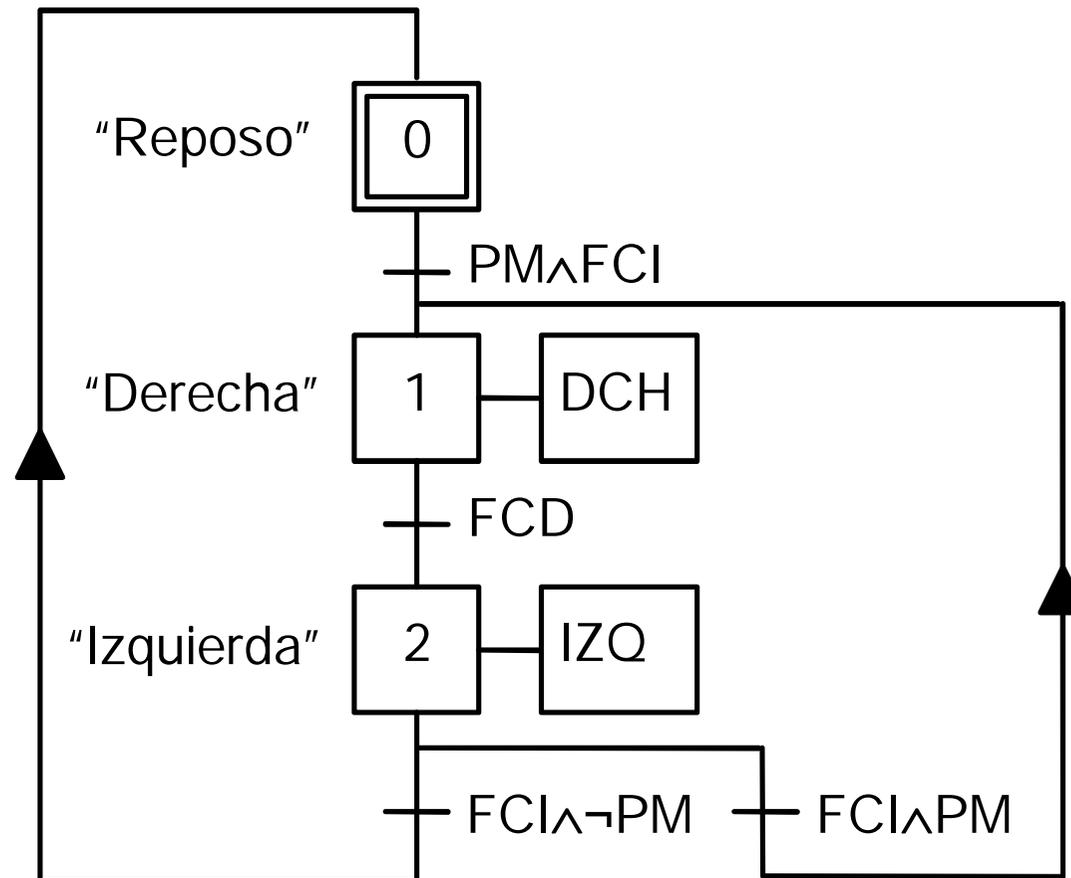
### Eliminación etapa transitoria



# Diseño con GRAFCET VI

## Ejemplo: carro va y viene

### Eliminación etapa transitoria



# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

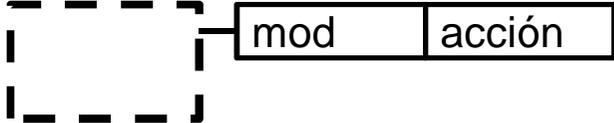
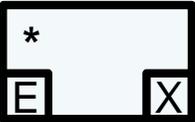
# Transcripción de GRAFCET I

## Elementos lenguaje SFC I

nombre	símbolo	comentario
etapa		<p>cada etapa tiene dos variables implícitas:</p> <p>&lt;nombre etapa&gt;.x</p> <p>&lt;nombre etapa&gt;.t</p>
etapa inicial		<p>cada POU programado en SFC únicamente puede contener una etapa inicial</p>
transición	 condición	<p>la condición debe ser una expresión booleana</p>

# Transcripción de GRAFCET II

## Elementos lenguaje SFC II

nombre	símbolo	comentario
acción		<p>cada etapa puede tener asociada una o varias acciones. Cada acción consta de un modificador y un nombre de acción</p>
acción de entrada y salida		<p>cada etapa puede tener asociada una acción de entrada y una de salida que se ejecutan al activar y desactivar la etapa respectivamente</p>

# Transcripción de GRAFCET III

## Elementos lenguaje SFC III

modificador	comentario
N: acción continua	acción activa mientras etapa activa
S: set acción almacenada	activa acción almacenada
R: reset acción almacenada	desactiva acción almacenada
L: acción limitada	acción activa durante un tiempo dado o hasta que se desactiva la etapa
D: acción retardada	activa acción después de un tiempo si la etapa sigue activa
P: acción pulso	acción ejecutada una sola vez

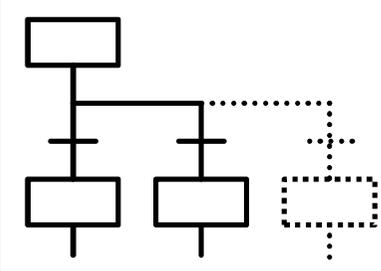
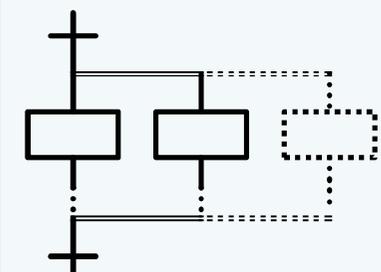
# Transcripción de GRAFCET IV

## Elementos lenguaje SFC IV

modificador	comentario
SD: acción almacenada y retardada	activa acción almacenada después de un tiempo si no se desactiva previamente la acción
DS: acción retardada y almacenada	activa acción almacenada después de un tiempo si la etapa sigue activa
SL: acción almacenada y limitada	activa acción almacenada durante un tiempo dado o hasta que se desactive la acción

# Transcripción de GRAFCET V

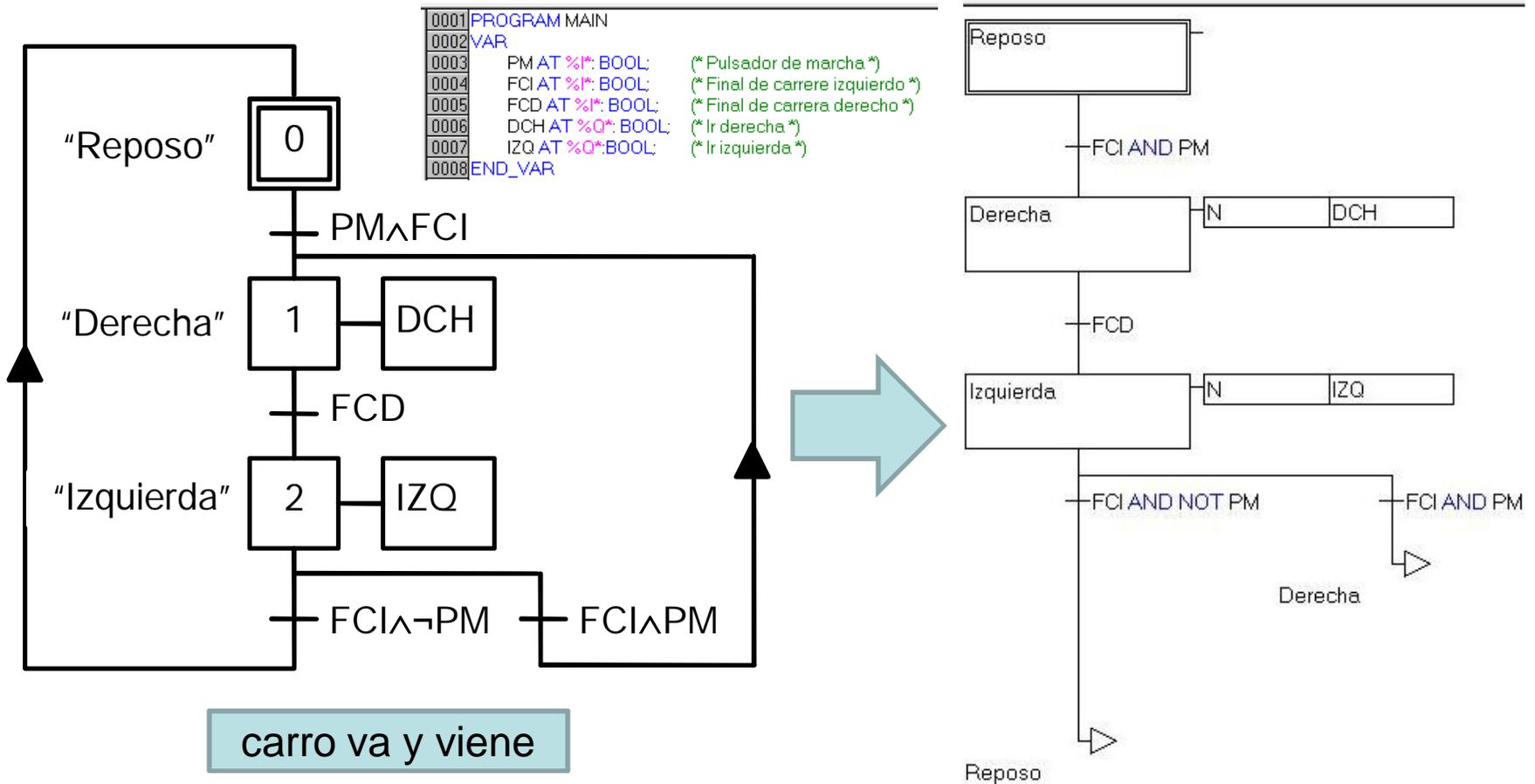
## Elementos lenguaje SFC V

nombre	símbolo	comentario
secuencia alternativa		<p>cada secuencia alternativa se alcanza a través de transiciones diferentes</p>
secuencia paralela		<p>todas las secuencias paralelas se alcanzan a través de una única transición común</p>
salto		<p>permite realizar saltos en la secuencia</p>

# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

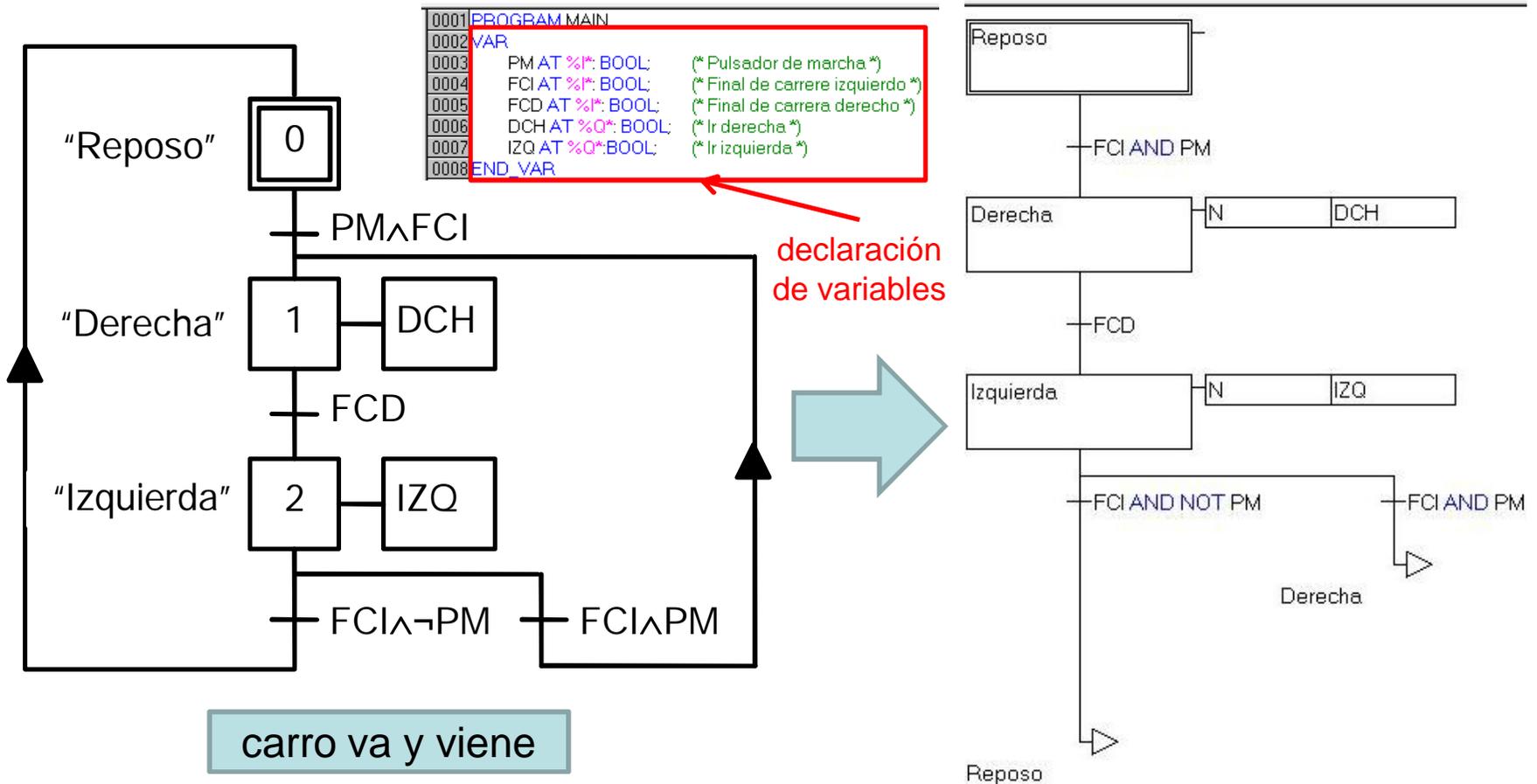
La transcripción es prácticamente directa.



# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

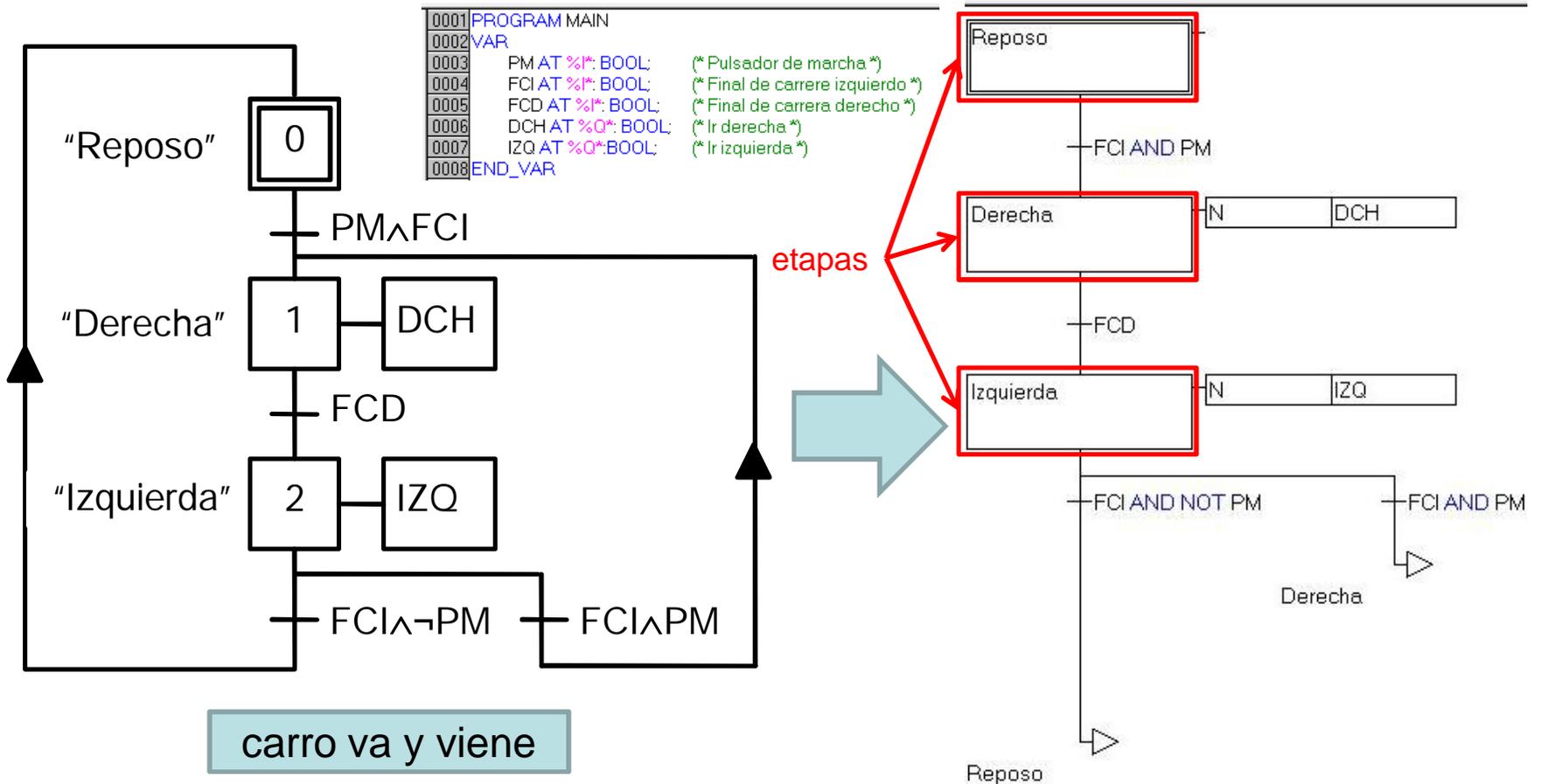
La transcripción es prácticamente directa.



# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

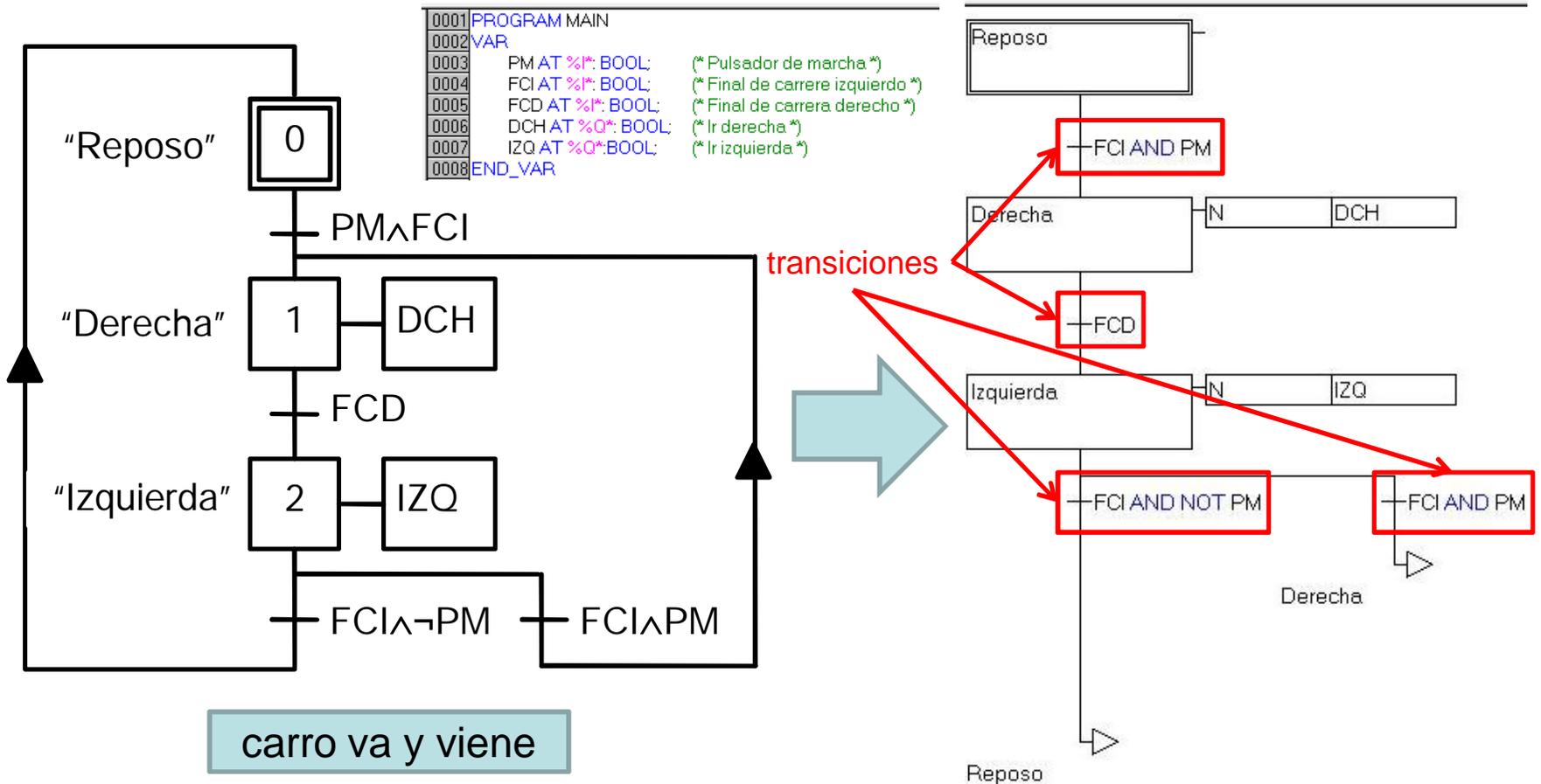
La transcripción es prácticamente directa.



# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

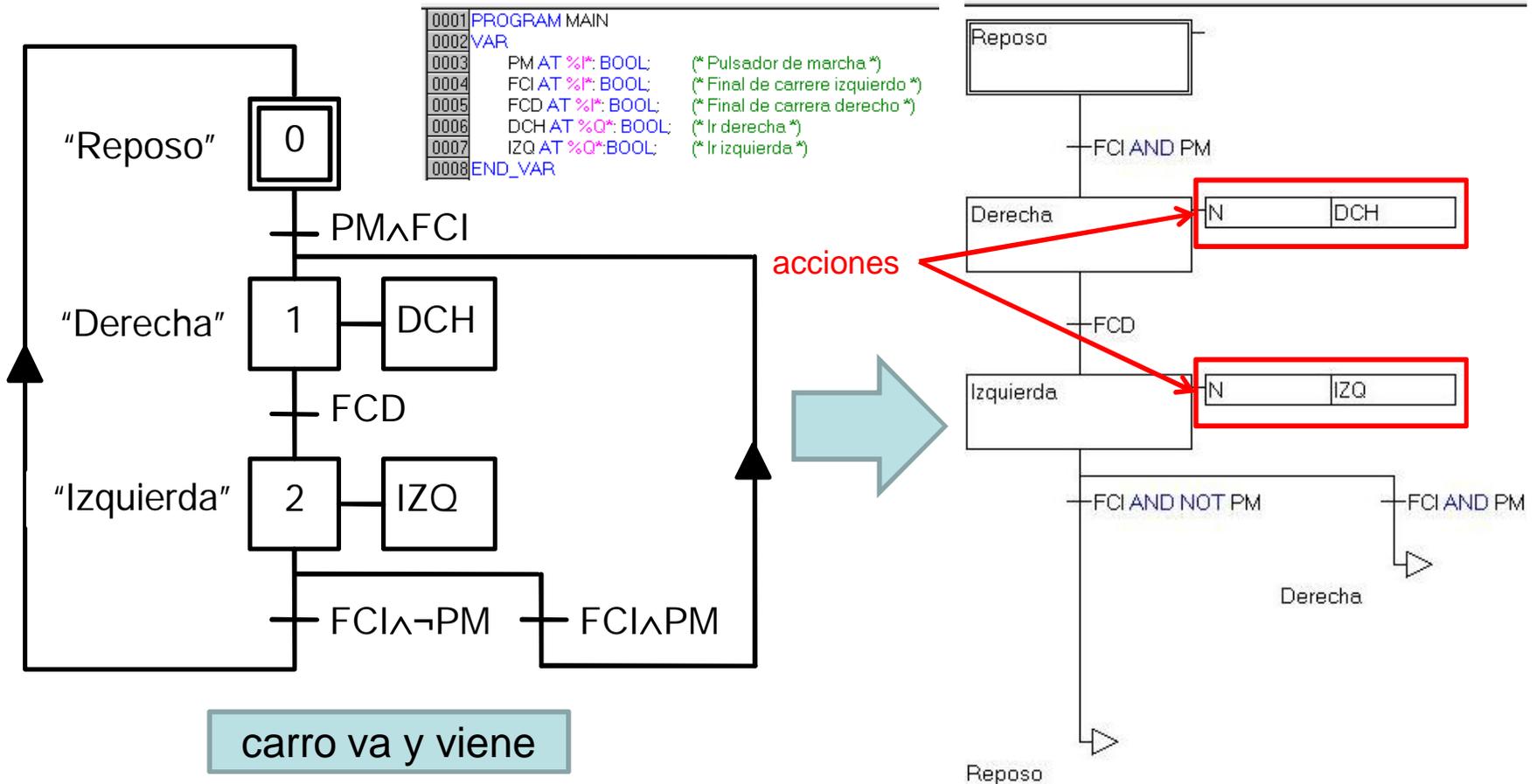
La transcripción es prácticamente directa.



# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

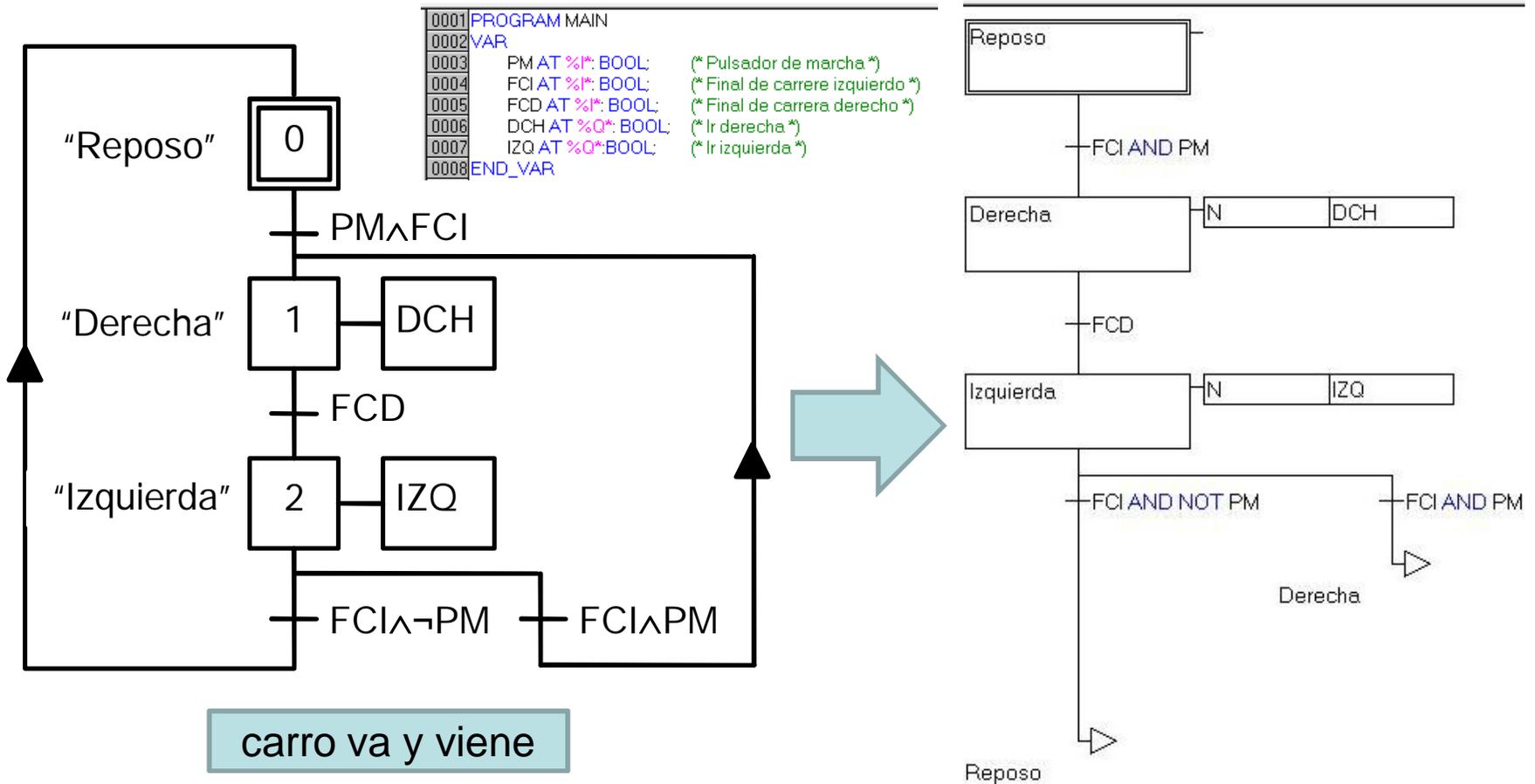
La transcripción es prácticamente directa.



# Transcripción de GRAFCET VI

## Transcripción de GRAFCET a SFC

La transcripción es prácticamente directa.



## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

9.1. Introducción

9.2. Principios generales

9.3. Símbolos básicos

9.4. Estructuras básicas

9.5. Diseño de automatismos con GRAFCET

9.6. Transcripción de GRAFCET: lenguaje SFC

9.7. Ejemplo: doble carro va y viene

# Ejemplo: doble carro va y viene I

---

## Doble carro va y viene simple

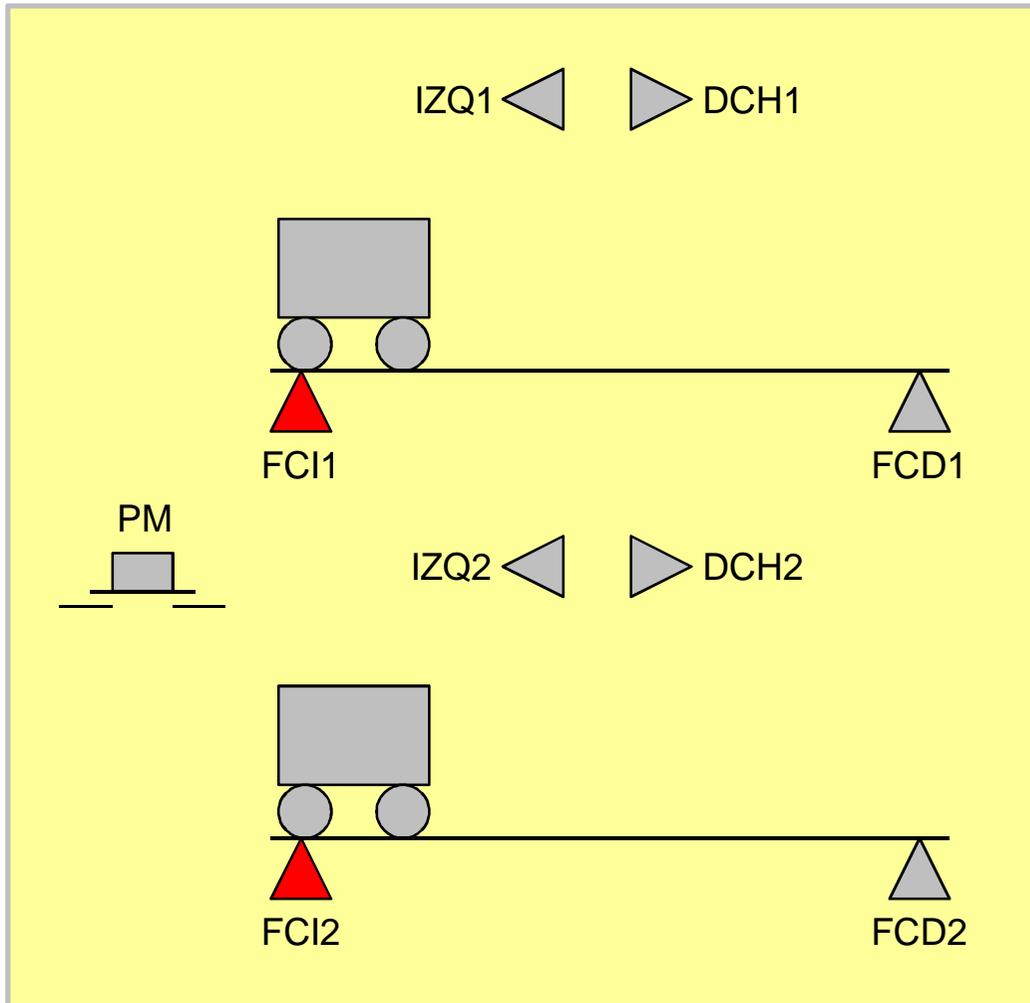
Se le añade un nuevo carro al ejemplo del carro va y viene.

El comportamiento de los carros es el mismo que se describió anteriormente, sólo que antes de iniciar el movimiento hacia la derecha ambos carros se deben encontrar en la izquierda, y antes de iniciar el movimiento hacia la izquierda ambos carros se deben encontrar en la derecha.

Ambos carros deben encontrarse inicialmente en la izquierda para comenzar el movimiento.

# Ejemplo: doble carro va y viene II

## Doble carro va y viene simple



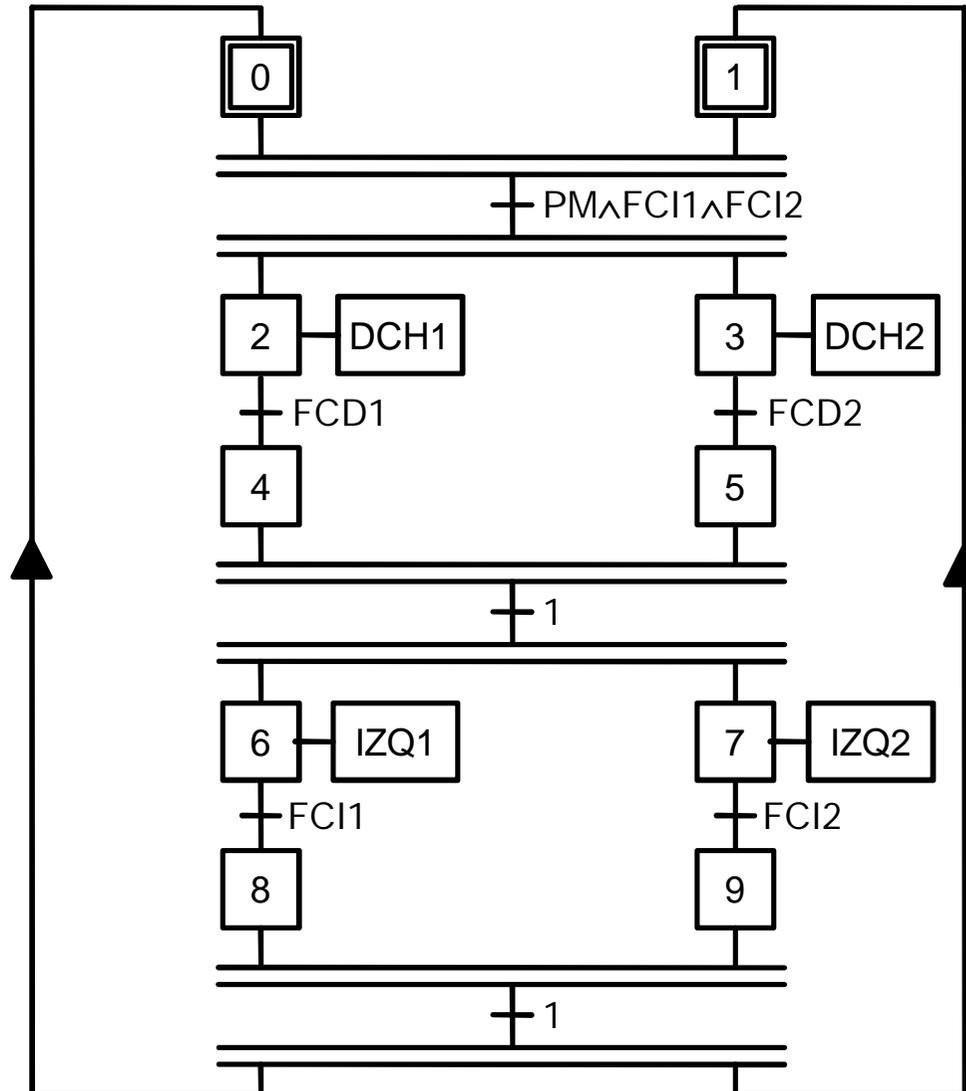
$E = \{ PM, FCI1, FCD1, FCI2, FCD2 \}$

$S = \{ DCH1, IZQ1, DCH2, IZQ2 \}$

entradas y salidas

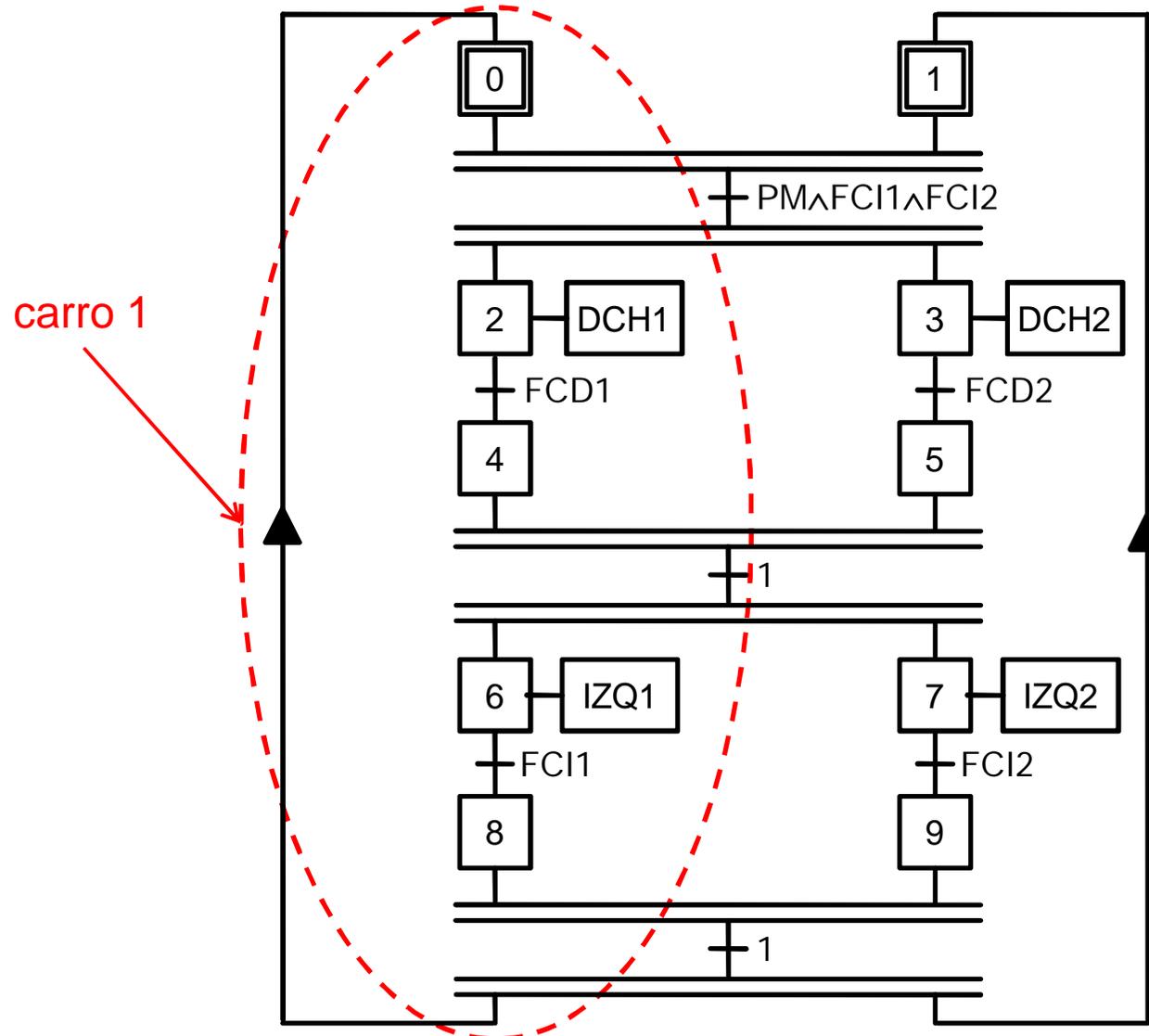
# Ejemplo: doble carro va y viene III

## Doble carro va y viene simple: GRAFCET



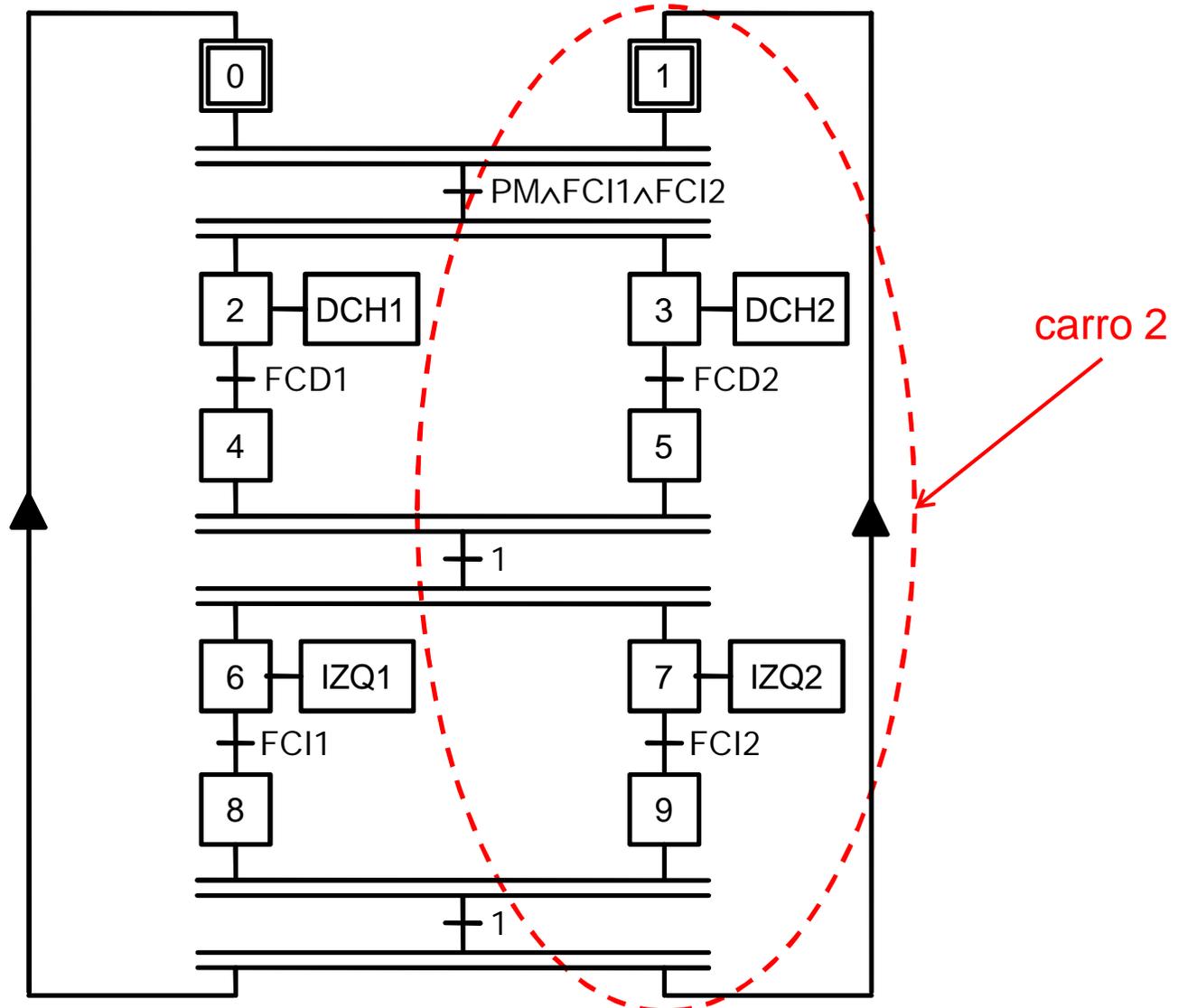
# Ejemplo: doble carro va y viene III

## Doble carro va y viene simple: GRAFCET



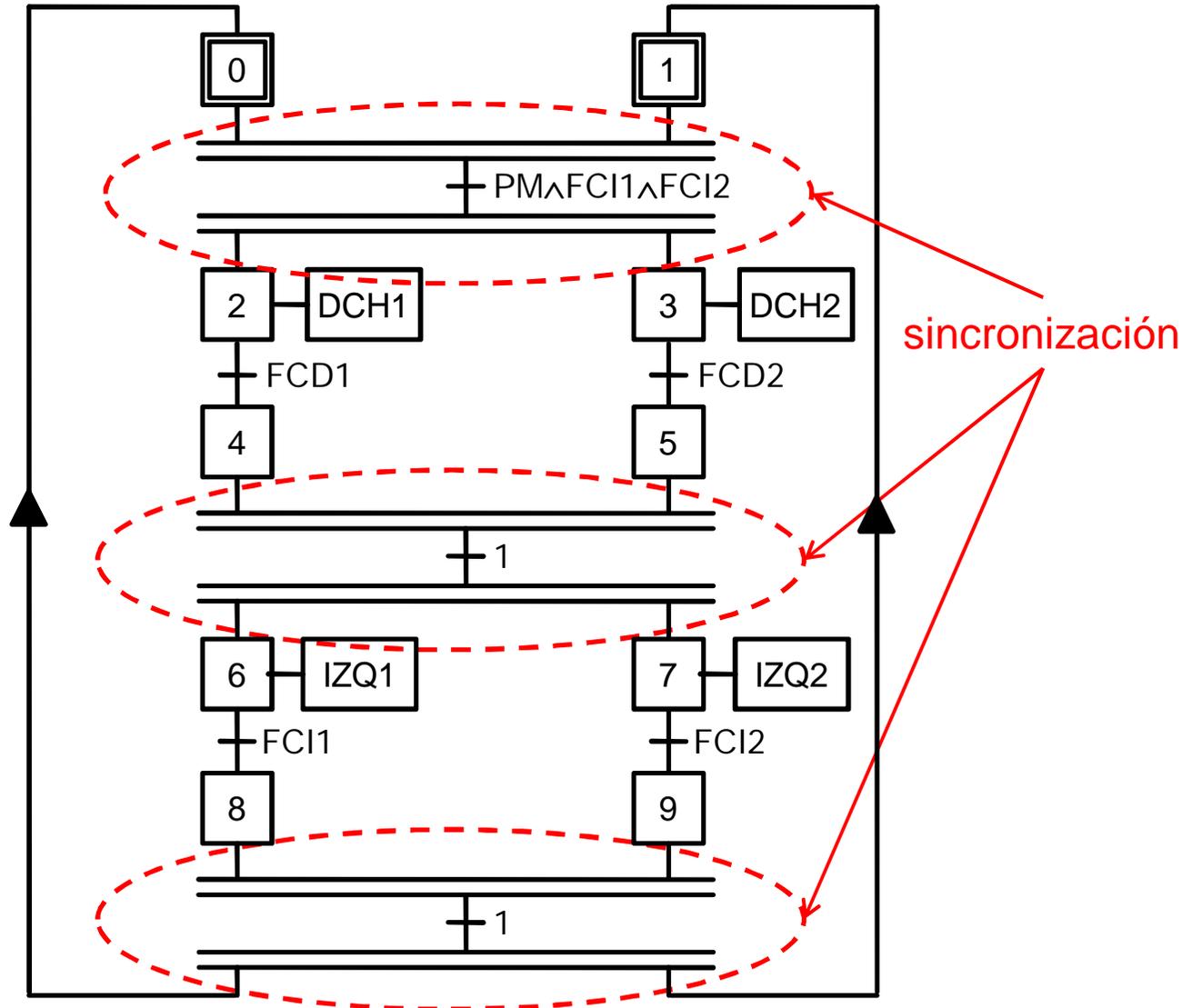
# Ejemplo: doble carro va y viene III

## Doble carro va y viene simple: GRAFCET



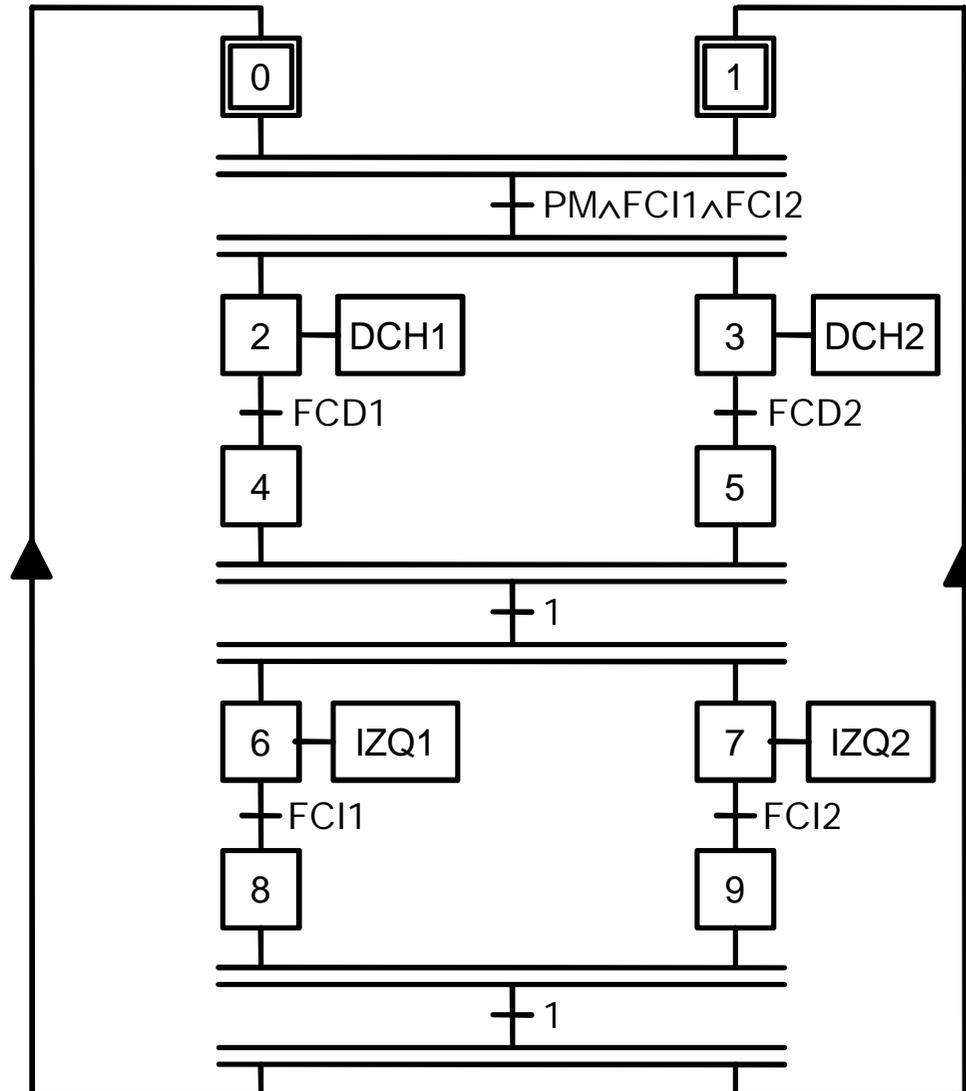
# Ejemplo: doble carro va y viene III

## Doble carro va y viene simple: GRAFCET



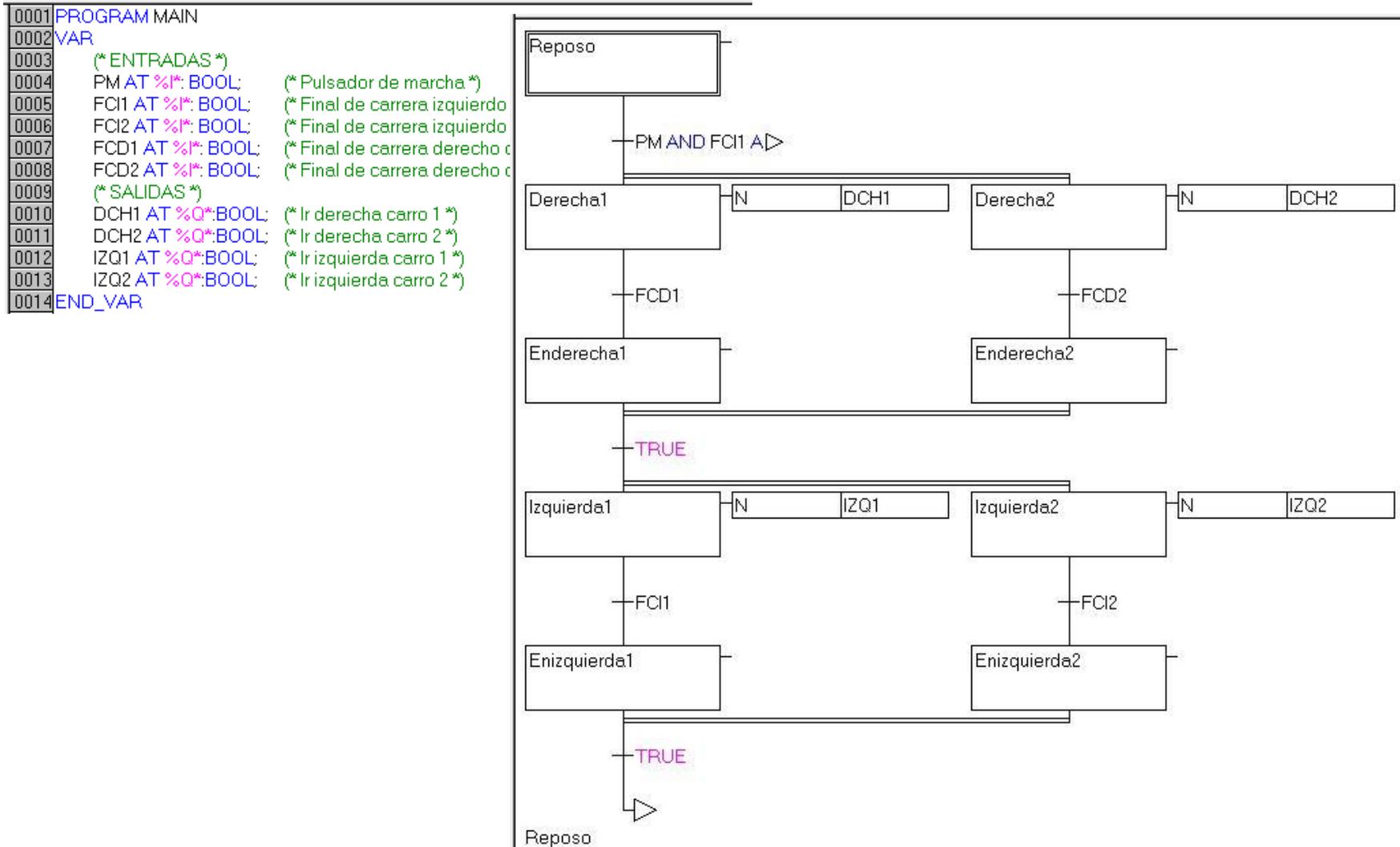
# Ejemplo: doble carro va y viene III

## Doble carro va y viene simple: GRAFCET



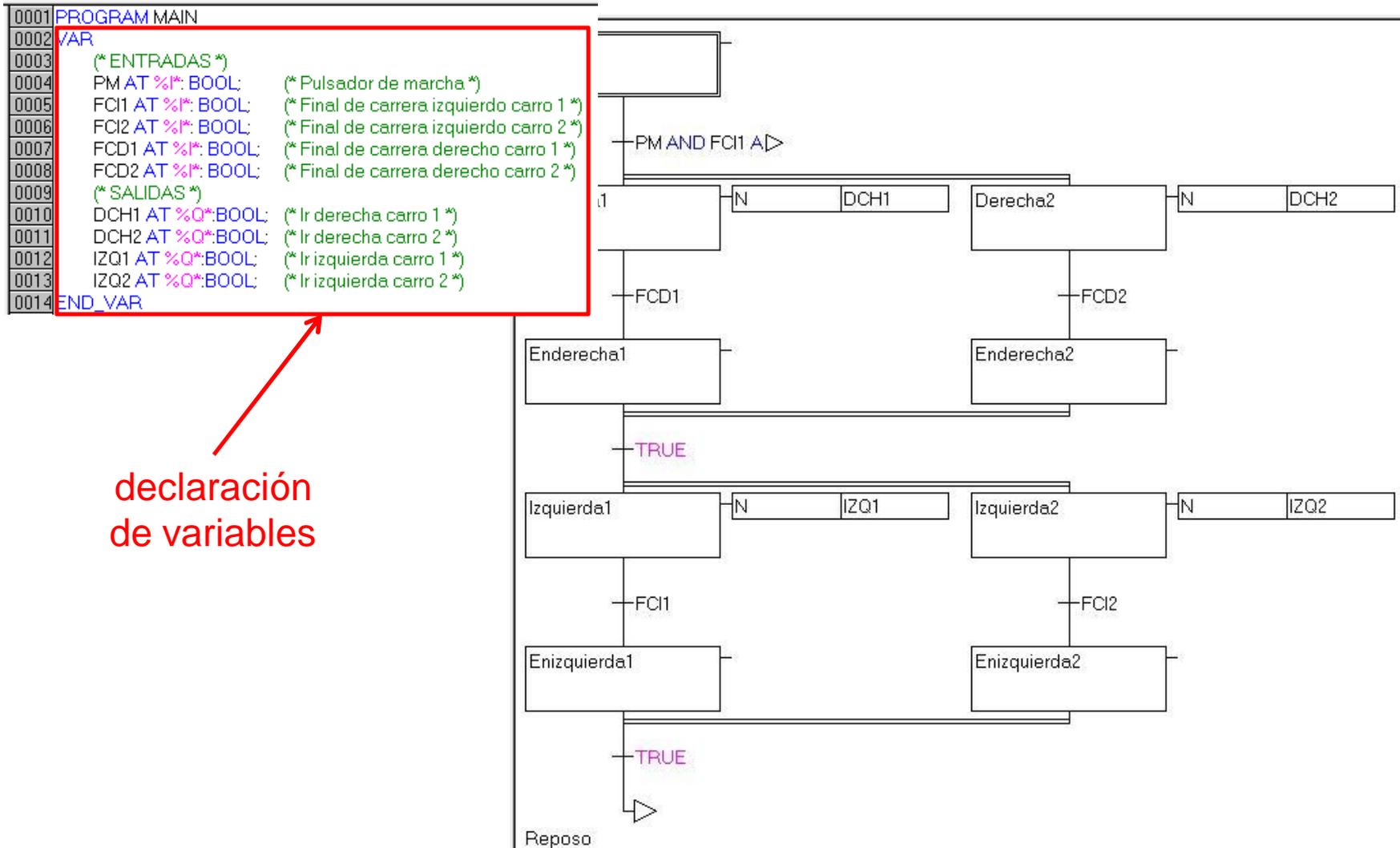
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



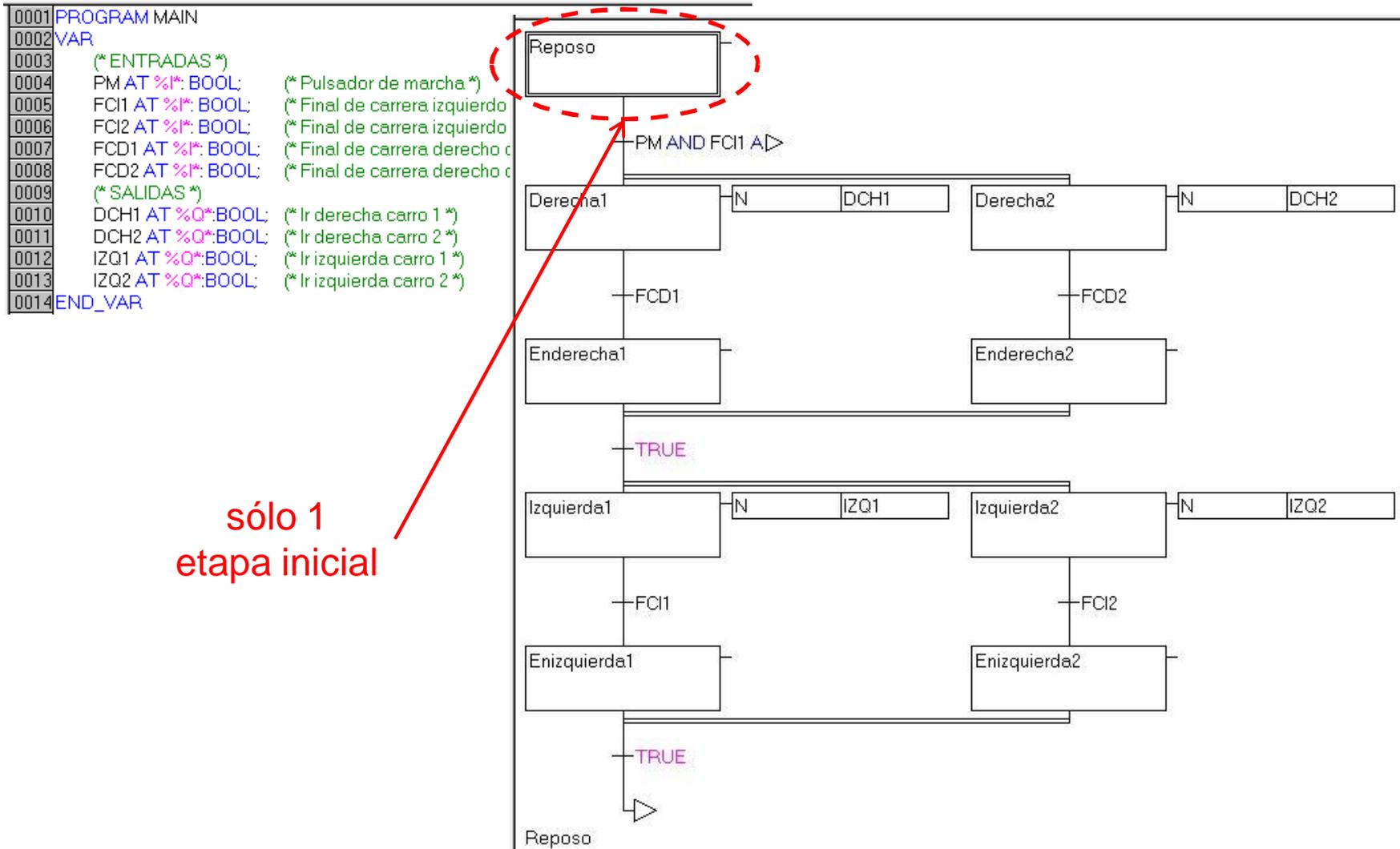
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



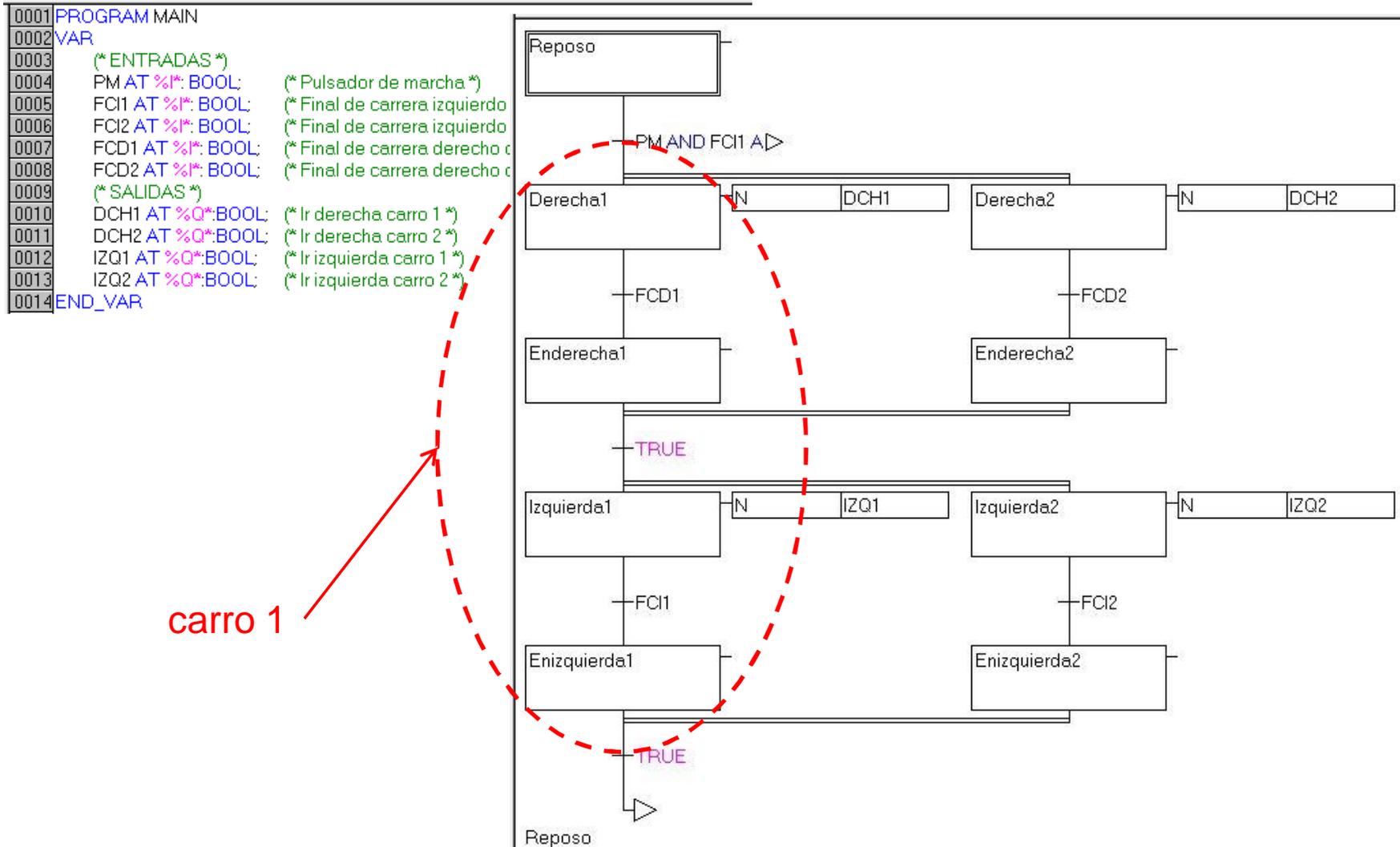
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



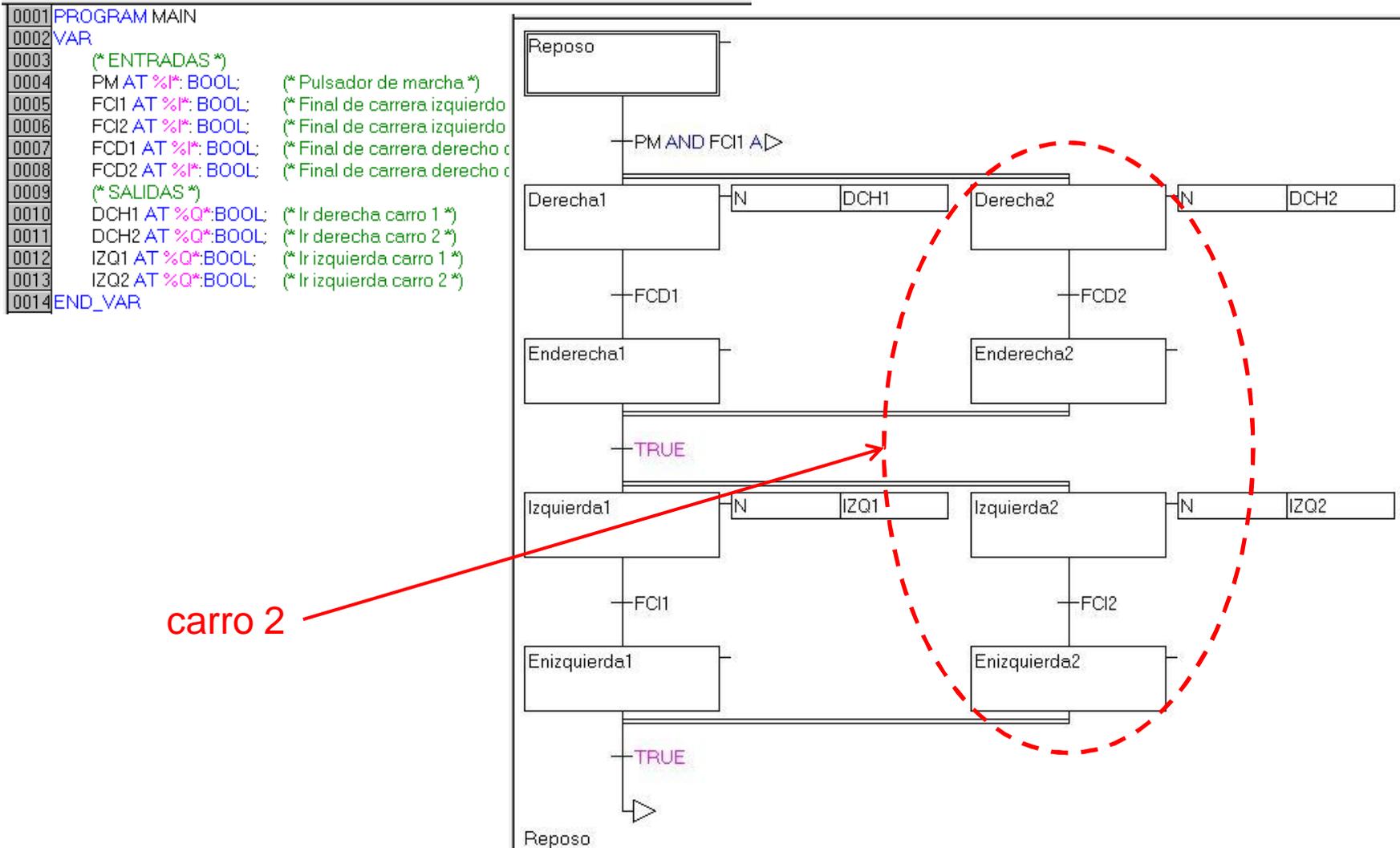
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



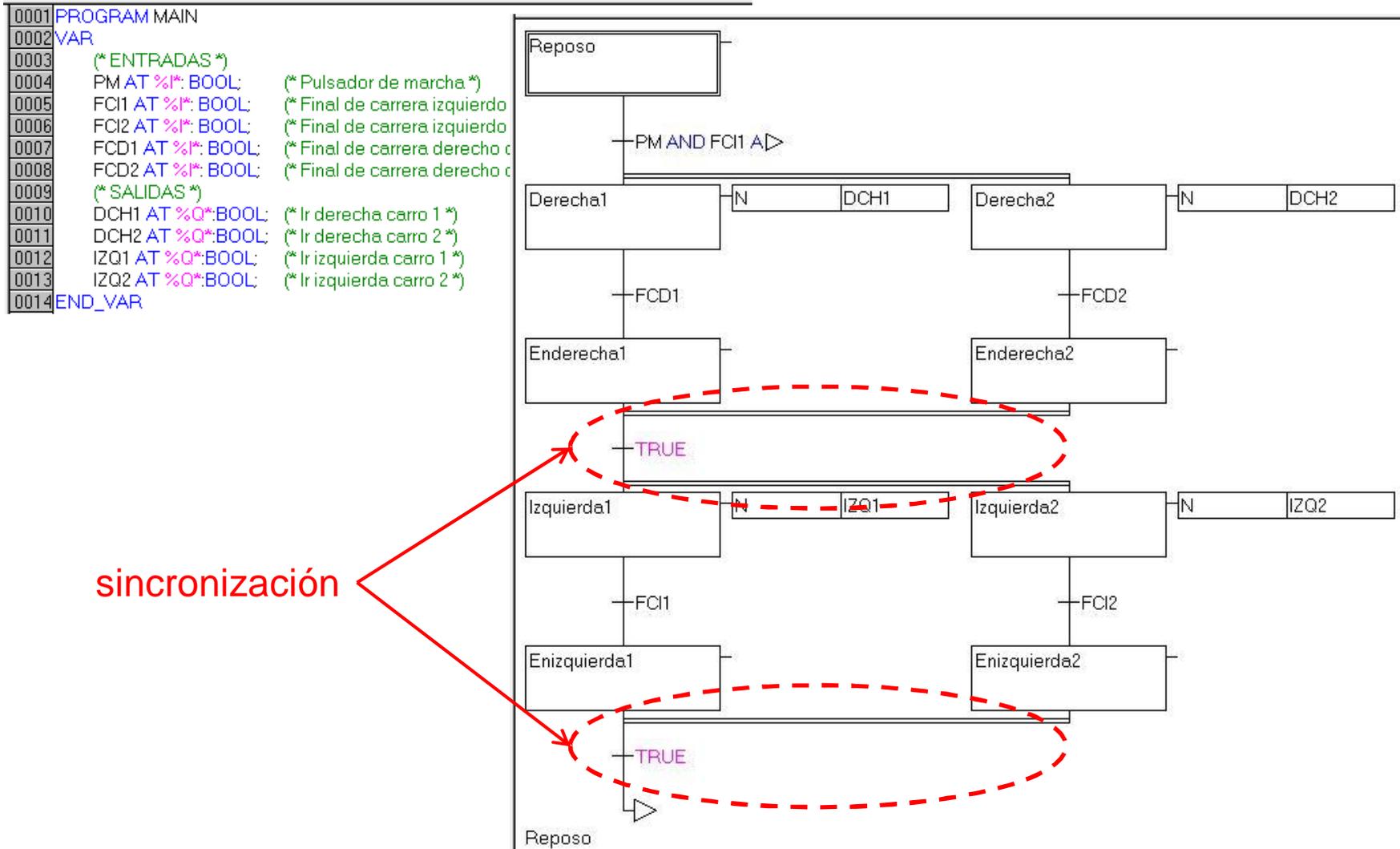
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



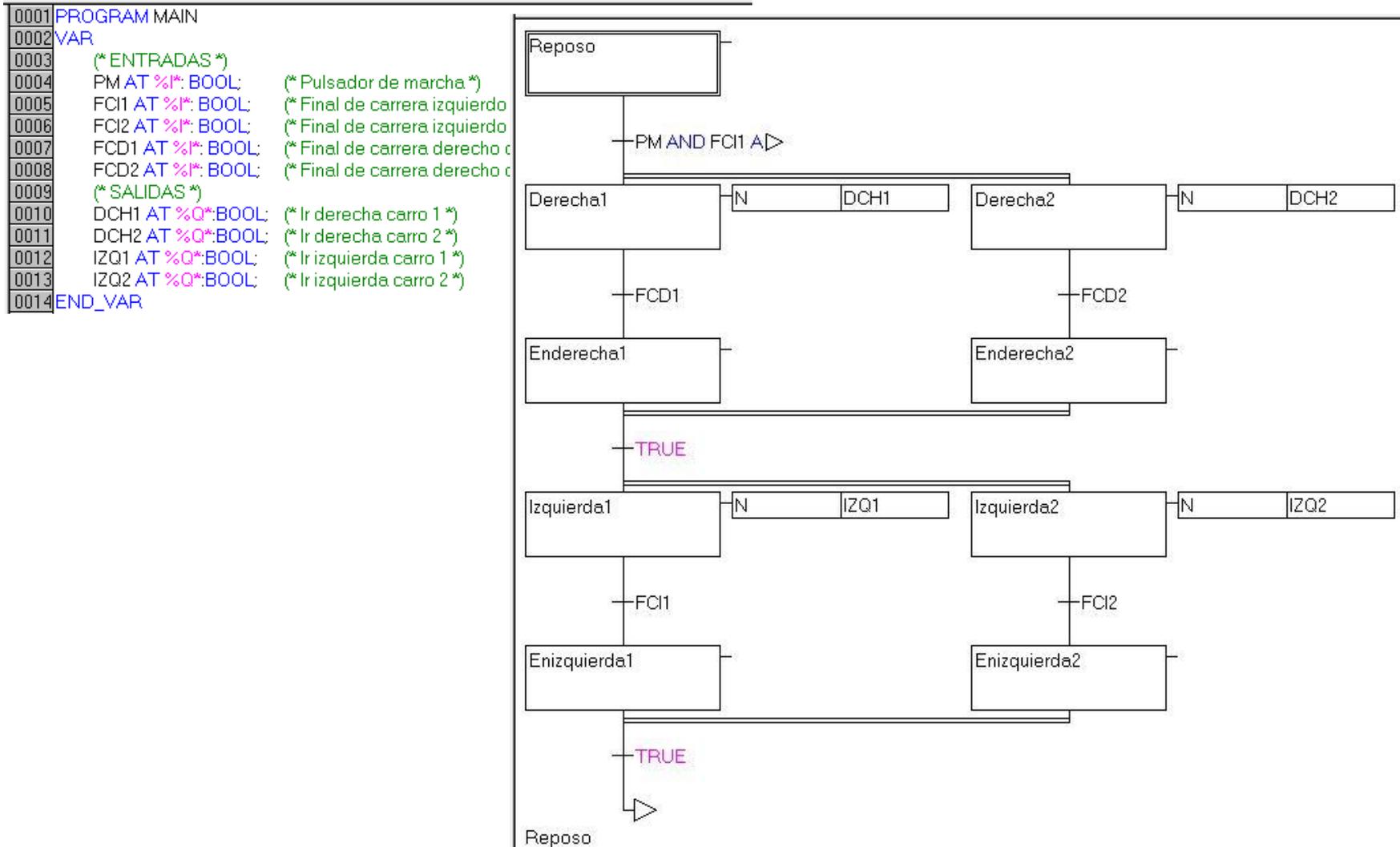
# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



# Ejemplo: doble carro va y viene IV

## Doble carro va y viene simple: SFC



# Contenido

---

## **Tema 9.- Diseño y programación de automatismos con GRAFCET (IEC 60848)**

- 9.1. Introducción
- 9.2. Principios generales
- 9.3. Símbolos básicos
- 9.4. Estructuras básicas
- 9.5. Diseño de automatismos con GRAFCET
- 9.6. Transcripción de GRAFCET: lenguaje SFC
- 9.7. Ejemplo: doble carro va y viene