COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Diseño e implementación de sistemas secuenciales usando GRAFCET

Prof. Dr. José Antonio Rodríguez Mondéjar
mondejar@comillas.edu

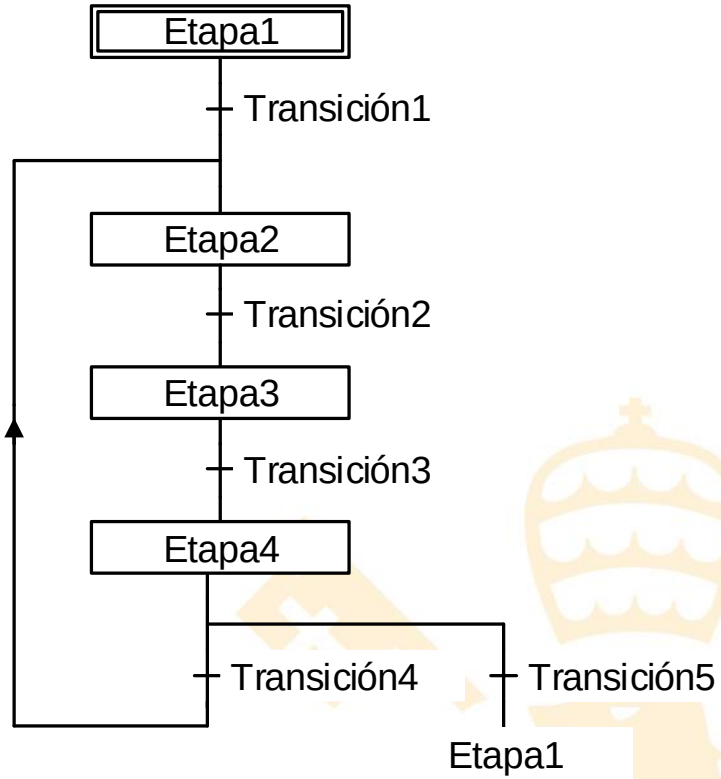
Escuela Técnica Superior de Ingeniería ICAI
Departamento de Electrónica, Automática y Comunicaciones

Enero 2023

comillas.edu

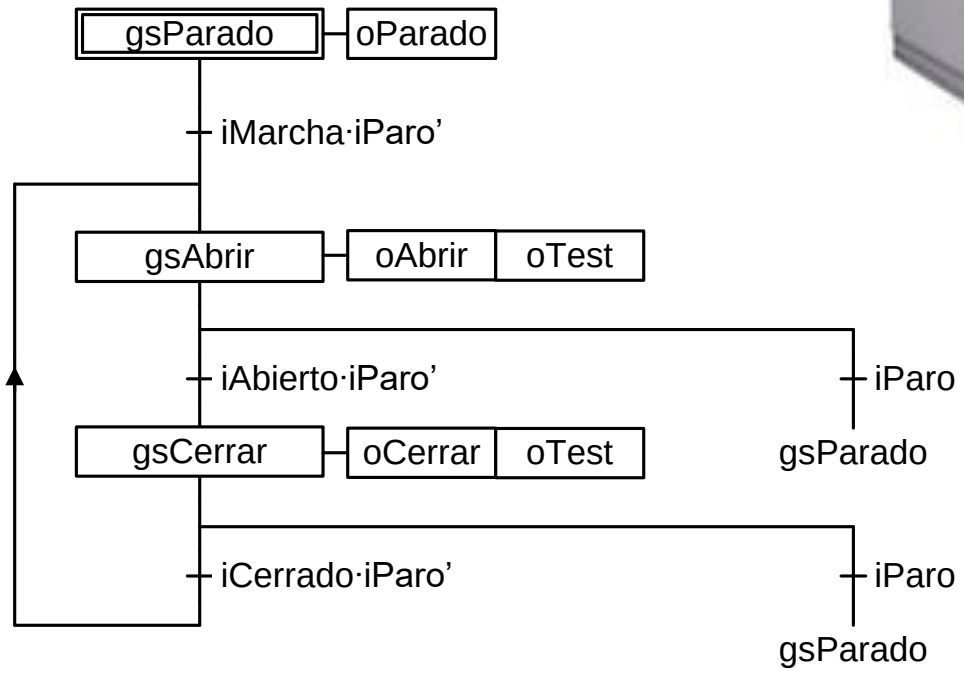
GRAFCET

- **Es una metodología para especificar el comportamiento de un sistema secuencial**
  - Definida en el estándar IEC 60848
- **Utiliza un lenguaje gráfico que organiza el control secuencial como un conjunto de etapas y transiciones interconectadas mediante uniones, denominado grafcet**
  - GRAFCET vs grafcet

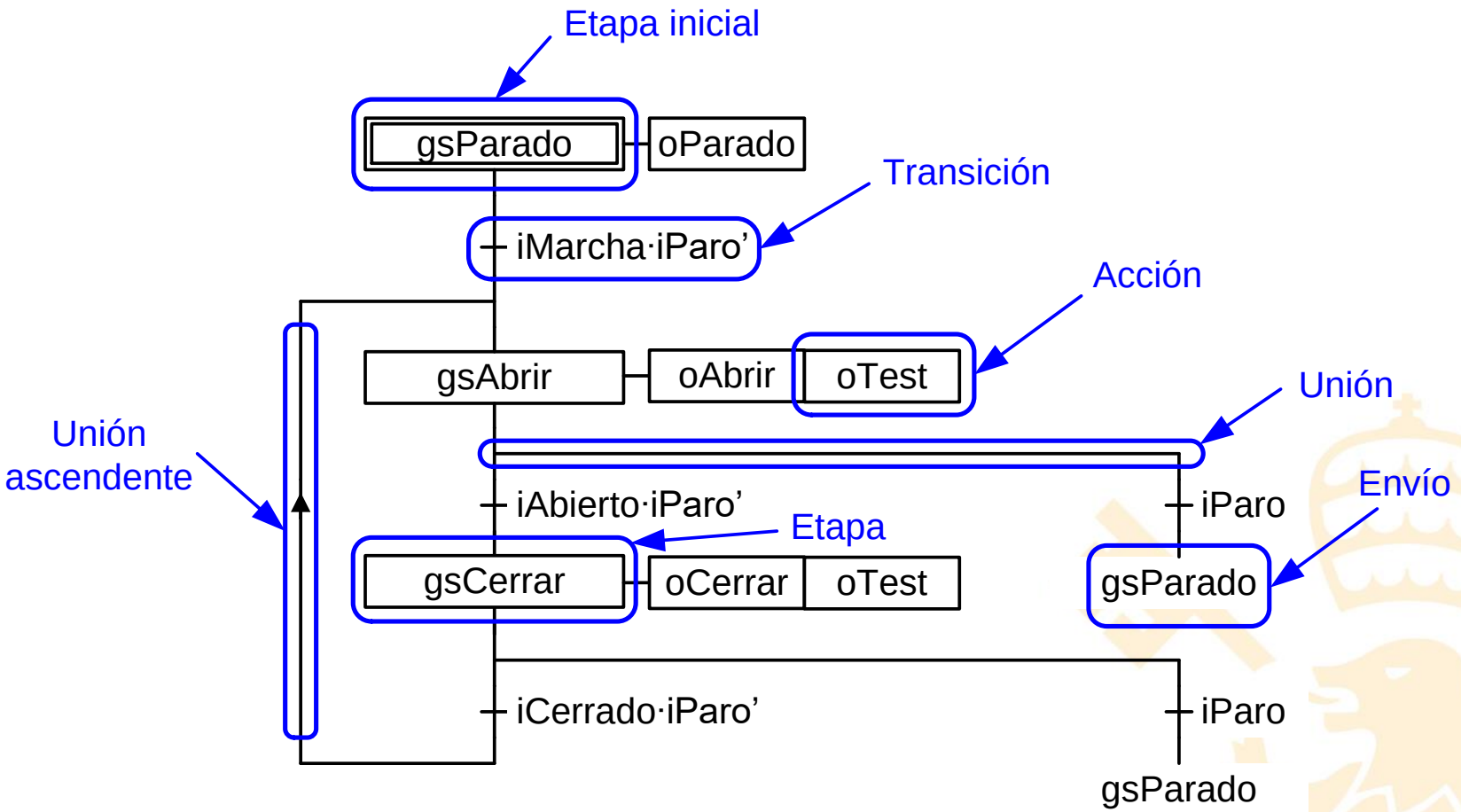


Ejemplo de grafcet

- **Sistema para ensayo de guías de cajón**
  - Abrir y cerrar cajón
  - Caso real

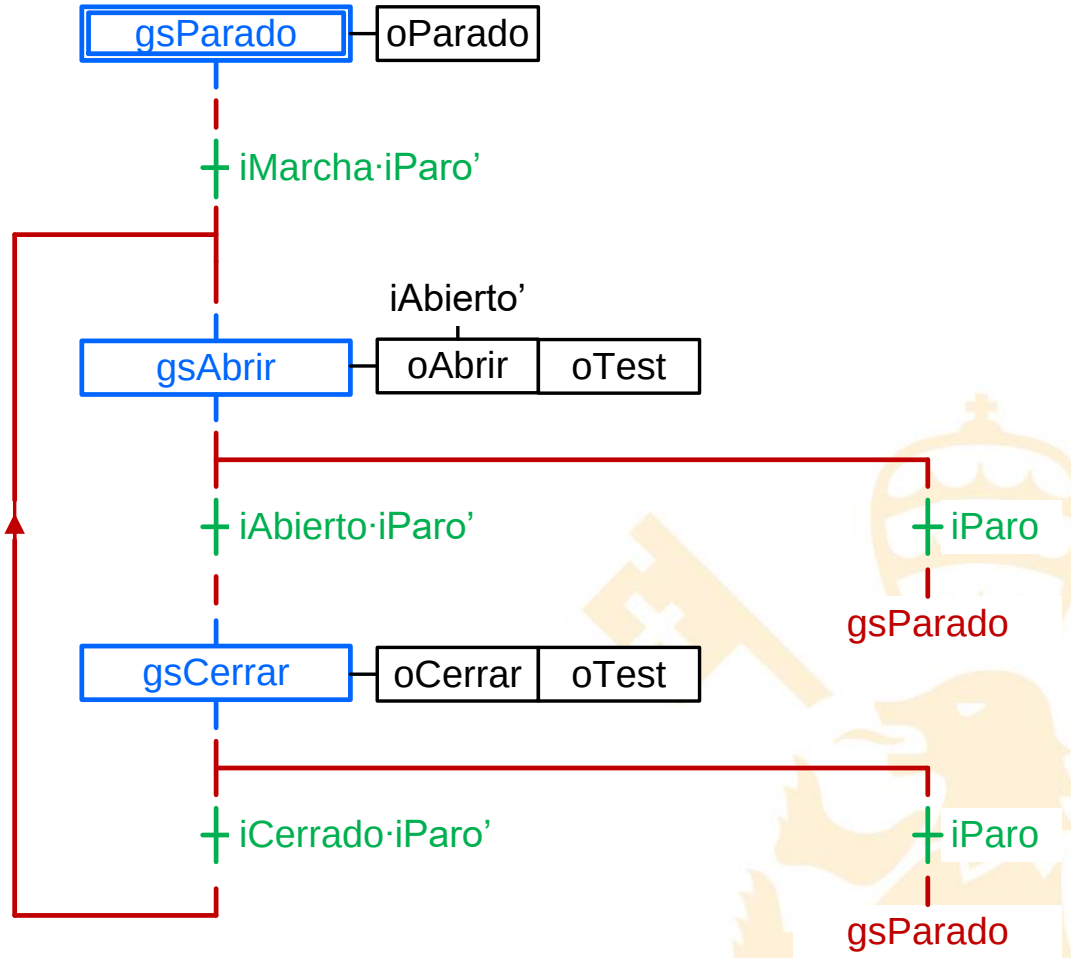


Elementos de un grafcet



Más detalle de los elementos que forman el grafcet

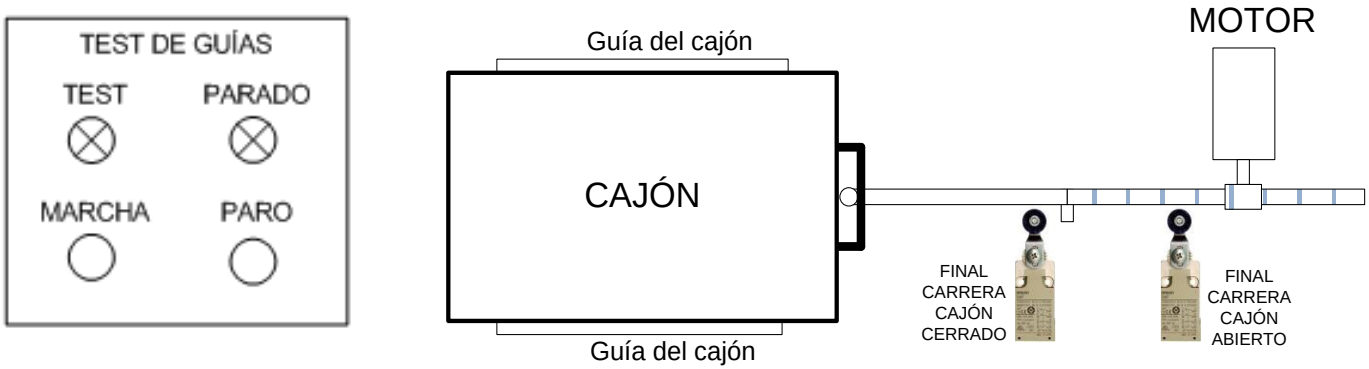
- **Azul: etapas**
- **Verde: transiciones**
- **Rojo: uniones**
- **Negro: acciones**



¿Cómo se resuelve un problema secuencial mediante GRAFCET?

- **Paso 1: Entender los requisitos**
  - Entender el funcionamiento que debe seguir el proceso una vez automatizado
  - Comprender la interacción con el operador
- **Paso 2: Diseñar el grafcet descriptivo**
  - Grafcet descriptivo: borrador del grafcet final
  - Objetivo: idea general sobre las etapas y las transiciones sin entrar en detalles tecnológicos
- **Paso 3: Diseñar el hardware**
  - Conexión física entre PLC, proceso y operador
- **Paso 4: Diseñar el grafcet tecnológico**
  - Definir tabla de variables de entrada y salida
  - Convertir el grafcet descriptivo en tecnológico:
    - Nombre de etapas adecuadas, condiciones lógicas en las transiciones y acciones según diseño hardware
- **Paso 5: Programación del grafcet tecnológico**
  - Lenguaje SFC (Graph en Siemens)

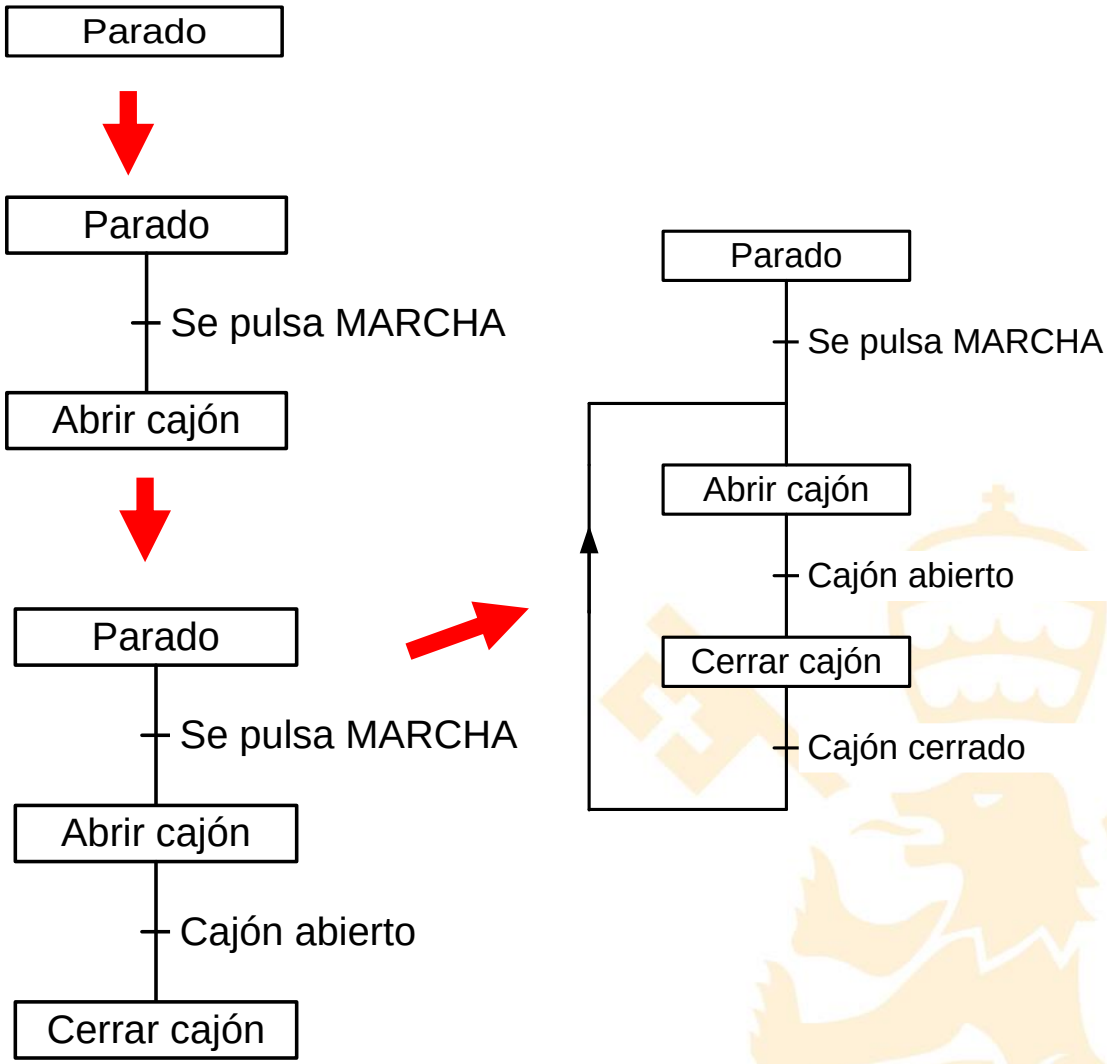
Ejemplo de automatizar usando GRAFCET: test de guías de cajón



- **Paso 1: Requisitos:**
  - Cuando se alimenta el sistema, evoluciona a la situación de parado.
  - Al pulsar MARCHA, el sistema de test se pone en marcha abriendo y cerrando el cajón continuamente.
  - Si se pulsa PARO, el test se para.

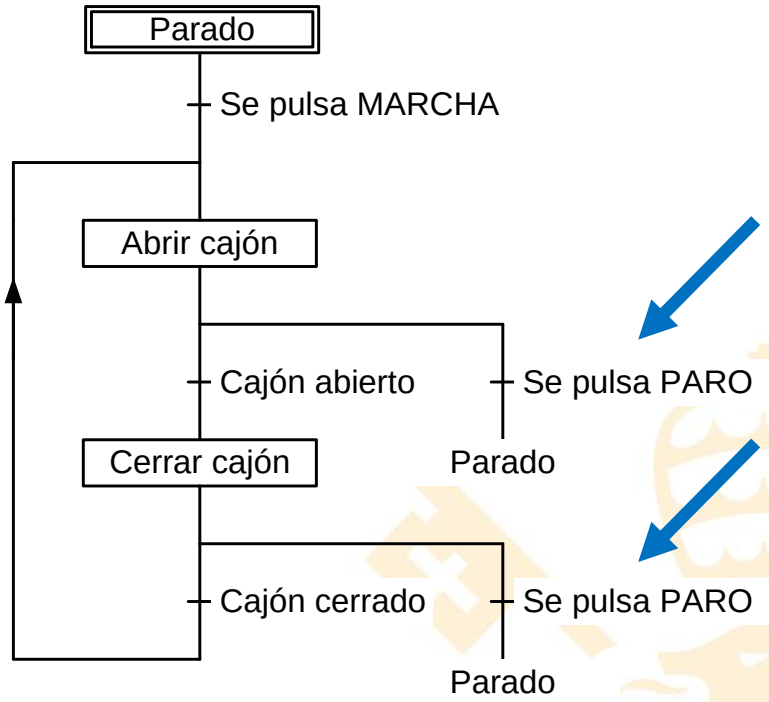
Paso 2: grafcet descriptivo ideal

- **¿Cómo comenzar?**
  - Generar todas las etapas y a continuación las transiciones
  - Comenzar desde una etapa conocida (Ejemplo: PARADO) e ir añadiendo etapas
- **Nombre de las etapas**
  - Frase corta que describe la situación
- **Condición en la transición**
  - Proposición lógica evaluable fácilmente como 0 o 1



Paso 2: grafcet descriptivo completo

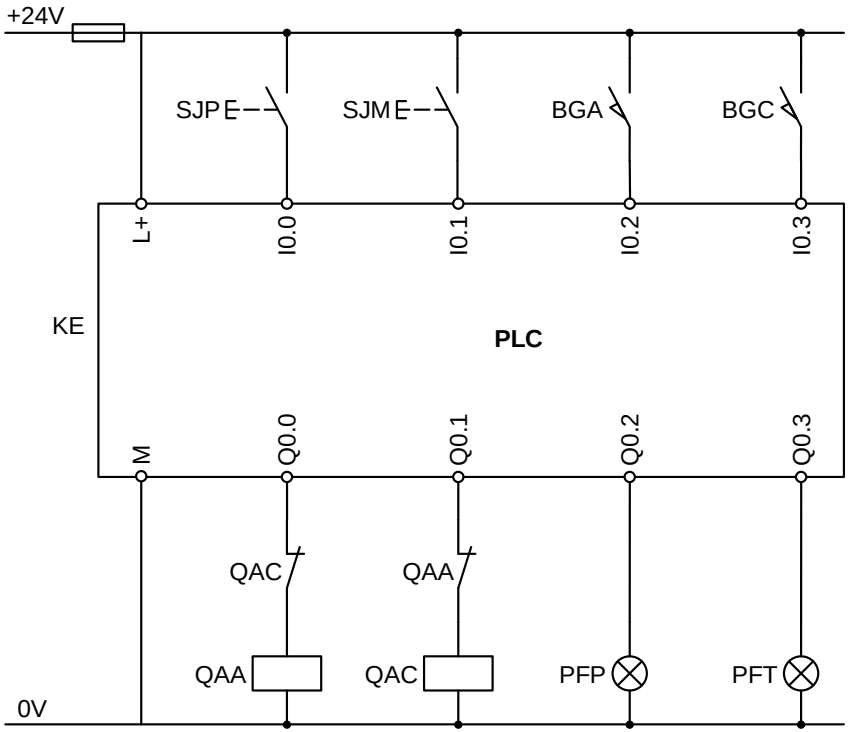
- **Incluir condiciones de parada y de avería**
  - Sólo incluir cuando el descriptivo recoge perfectamente el funcionamiento ideal
- **Resultado: grafcet descriptivo que debe seguir el proceso automatizado**



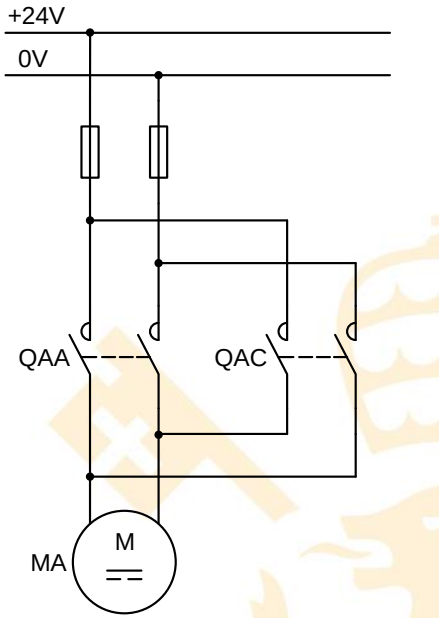
Tercer paso: hardware de la implementación

- **Definir sensores, accionamientos, equipos de control**
- **Esquemas de conexión**

ESQUEMA DEL CIRCUITO DE CONTROL



ESQUEMA DEL CIRCUITO DE POTENCIA



comillas.edu

Cuarto paso: definir variables de entrada y salida

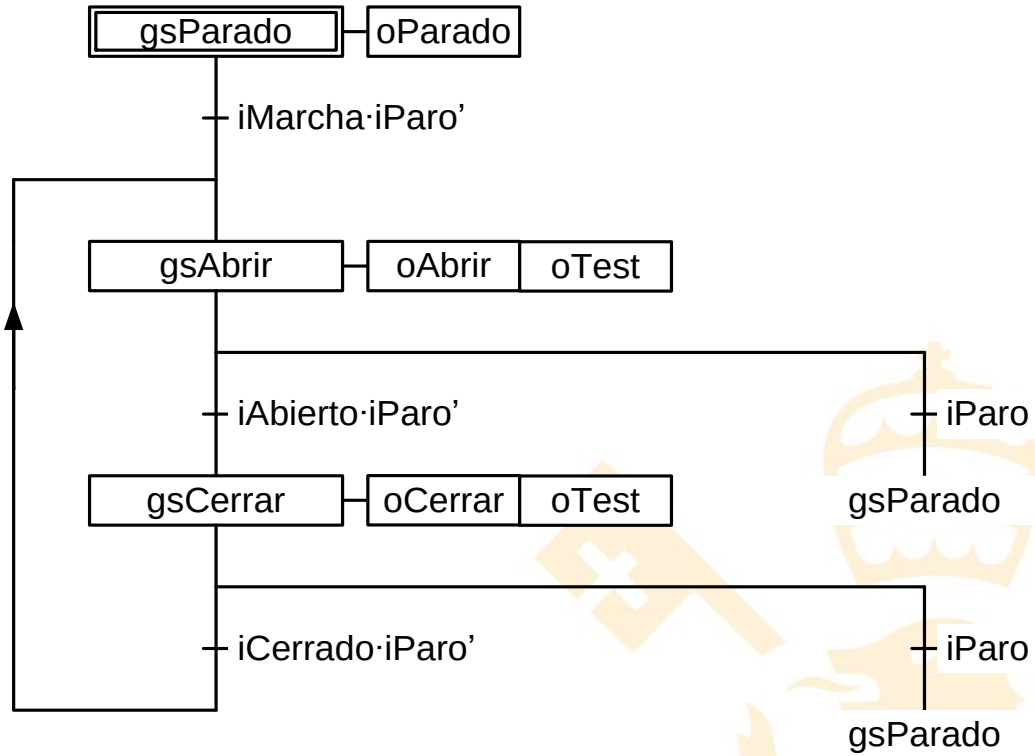
- **Paradas: de momento lógica positiva**

Fuente	Nombre variable de entrada PLC	Significado valores variable de entrada	Entrada física PLC
Pulsador SJP	iParo	1: Orden de paro 0: No hay orden de paro	%I0.0
Pulsador SJM	iMarcha	1: Orden de marcha 0: No hay orden de marcha	%I0.1
Final de carrera BGA	iAbierto	1: Final de carrera de cajón abierto activado 0: Final de carrera de cajón abierto no activado	%I0.2
Final de carrera BGC	iCerrado	1: Final de carrera de cajón cerrado activado 0: Final de carrera de cajón cerrado no activado	%I0.3

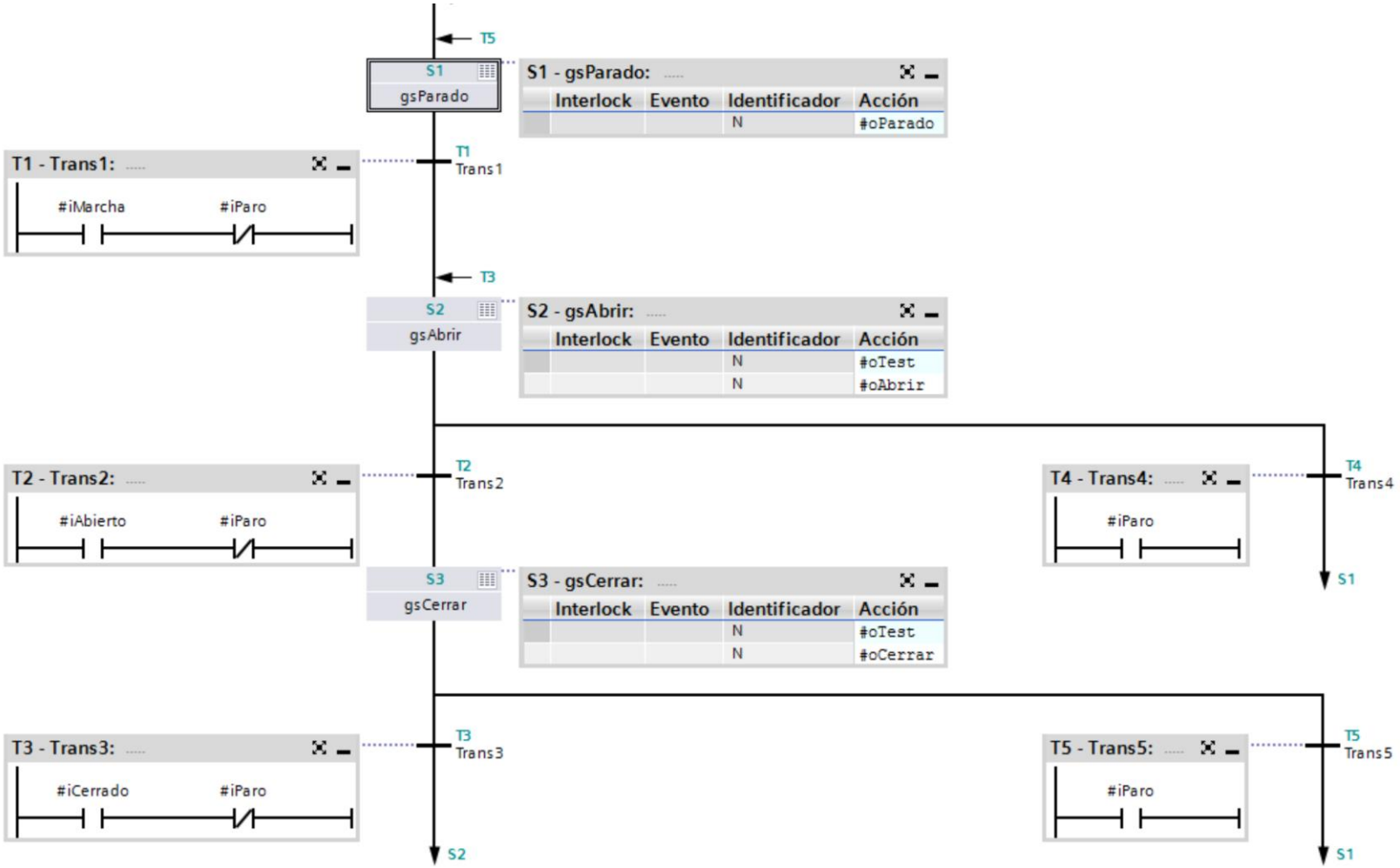
Receptor	Nombre variable salida PLC	Significado valores variable de salida	Salida física PLC
Contactor QAA	oAbrir	1: Mantener cerrado contactor de abrir cajón 0: Mantener abierto contactor de abrir cajón	%Q0.0
Contacto QAC	oCerrar	1: Mantener cerrado contactor de cerrar cajón 0: Mantener abierto contactor de cerrar cajón	%Q0.1
Piloto PFP	oParado	1: Sistema parado 0: No está en situación de “Sistema parado”	%Q0.2
Piloto PFT	oTest	1: Test en marcha 0: No está en situación de “Test en marcha”	%Q0.3

Cuarto paso: grafcet tecnológico

- **Convertir grafcet descriptivo en tecnológico**
  - Identificadores de etapas adecuados
    - Equivalentes a nombres de variables
  - Transiciones con condiciones lógicas correctas según hardware
  - Acciones según identificadores de variables de salida



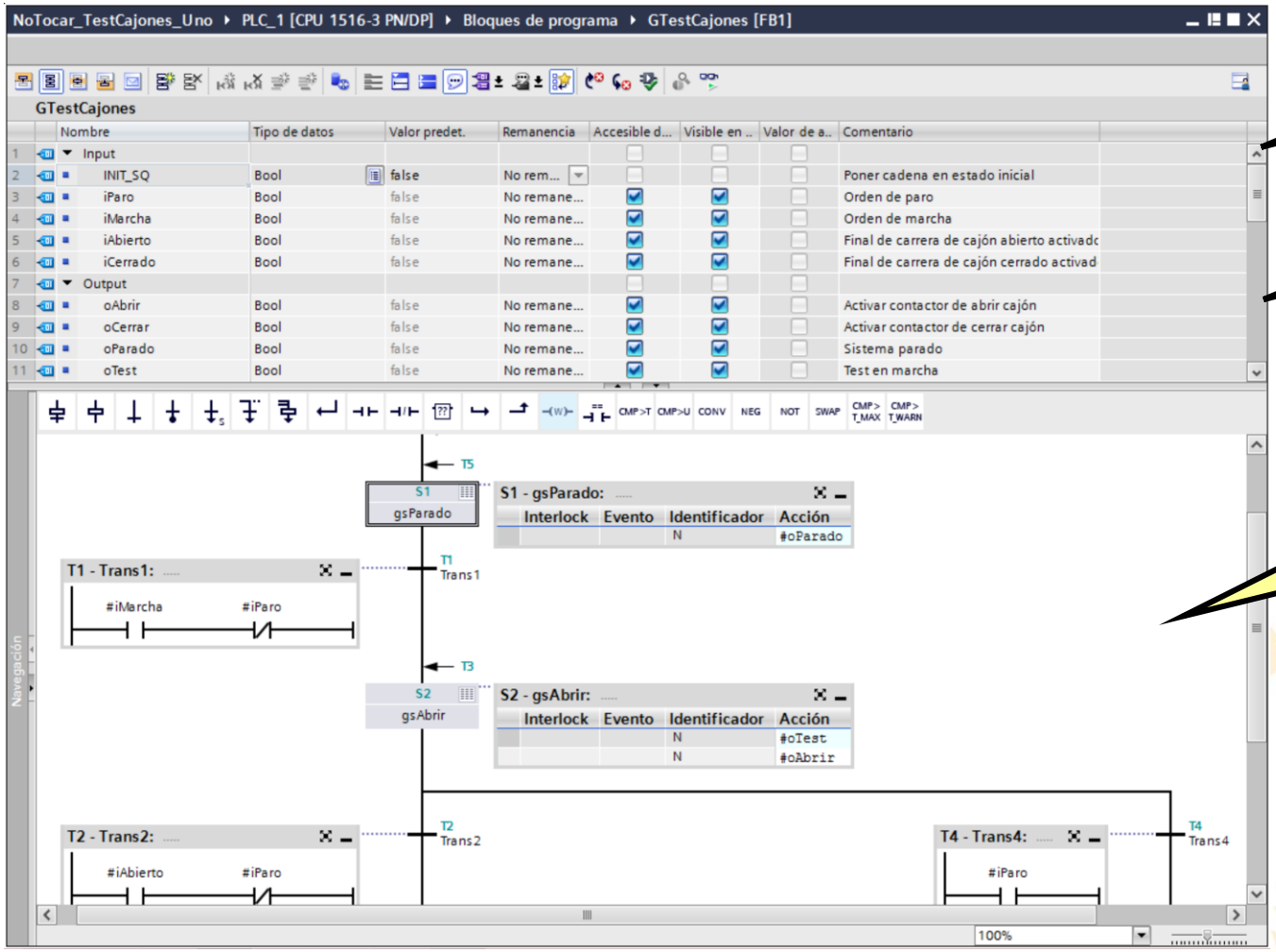
Quinto paso: programación en SFC (Graph-Siemens)



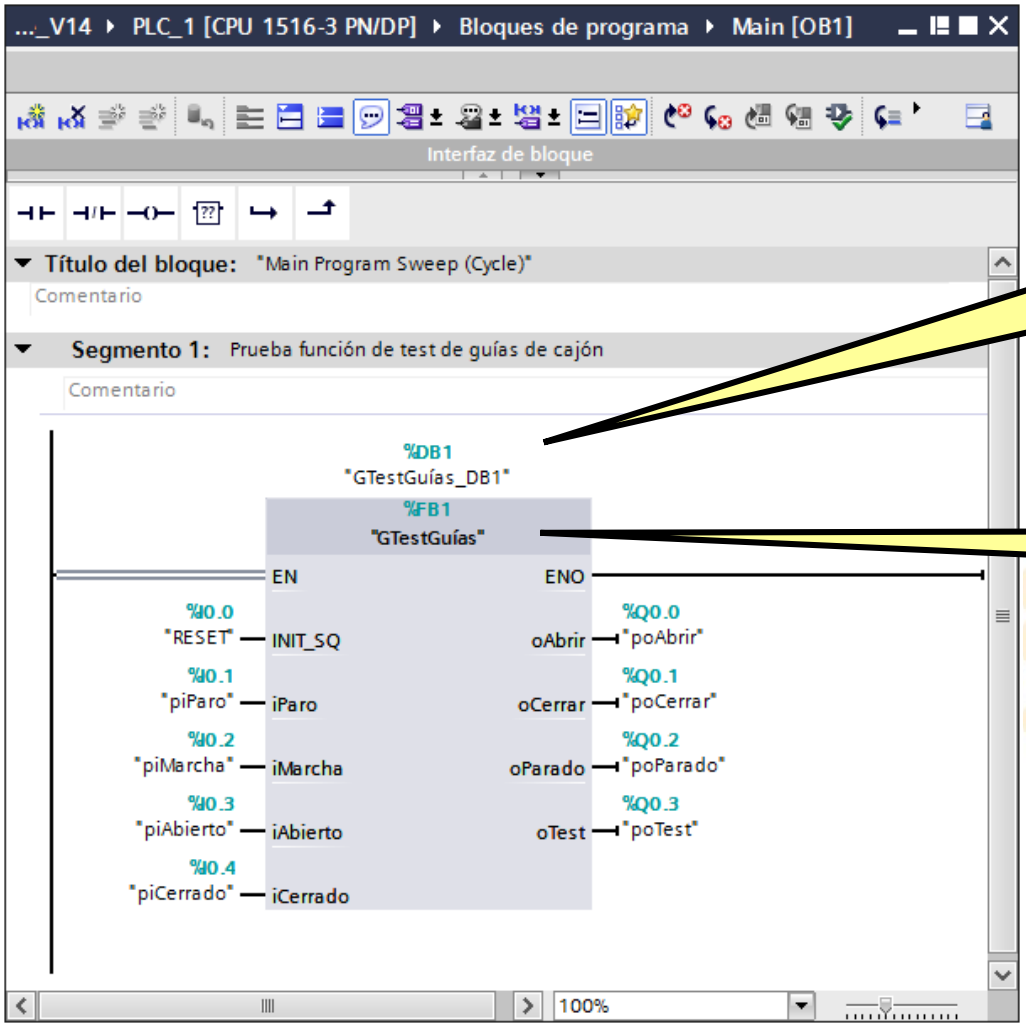
Programación en SFC

- **SFC: lenguaje definido por IEC 61131-3 que implementa GRAFCET**
- **GRAPH: subconjunto de SFC implementado por Siemens**
- **Siemens no admite el grafcet directamente en OB1**
- **Pasos para programar un Grafcet en GRAPH:**
  - Primero: Crear bloque FB para alojar el grafcet
    - FB: Estructura de programación donde se declaran variables de entrada, de salida e internas, así como las operaciones a realizar sobre ellas.
    - No es directamente ejecutable
  - Segundo: programar bloque FB según grafcet
    - Seleccionar lenguaje GRAPH
  - Tercero: Instanciar el bloque FB en OB1
    - Instanciar: llamar al bloque FB con un DB (bloque de datos) concreto. Cada DB representa una instancia del FB.
      - El DB mantiene los datos entre una llamada y la siguiente.
    - Habrá tantas instancias (DBs) como ejecuciones independientes del mismo código se necesiten.
- **Detalle de la programación en el laboratorio**

Bloque FB: grafcet del test de guías de cajón



Llamada a la instancia del FB en OB1 (programa principal)



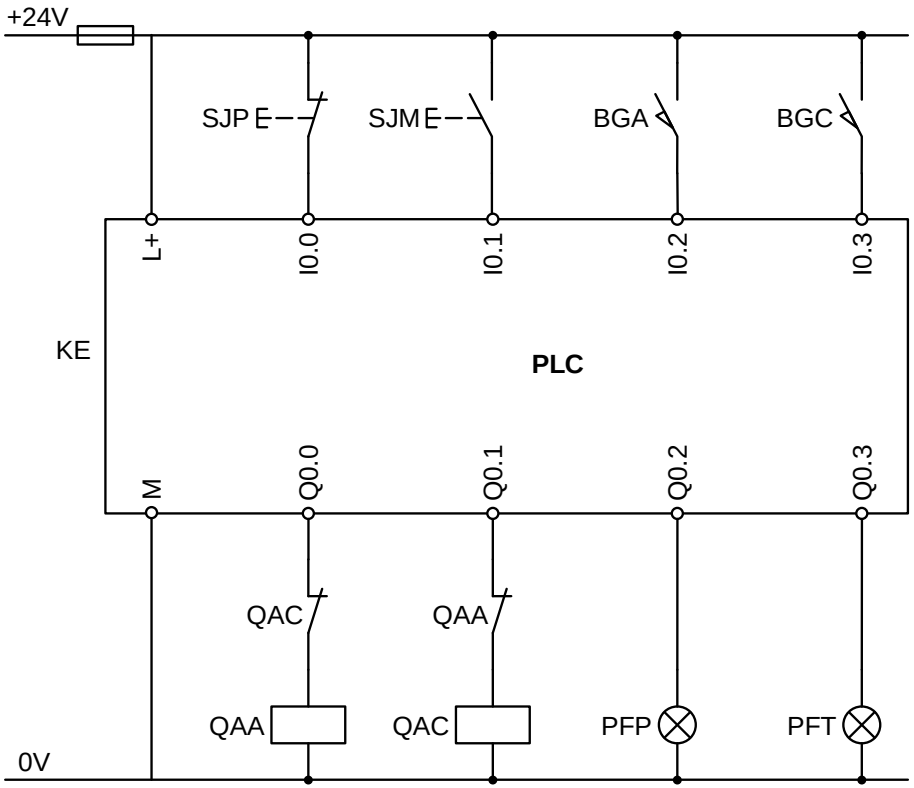
Referencia al bloque de datos que va a utilizar el FB

Referencia al FB

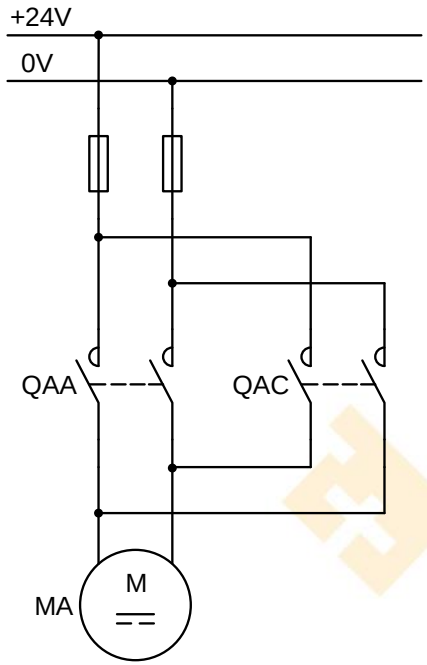
Solución de la orden de parada: parte hardware

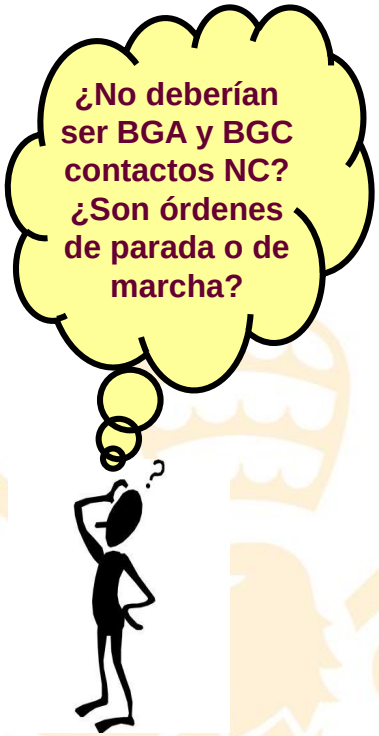
- **Orden de parada por lógica negativa**

ESQUEMA DEL CIRCUITO DE CONTROL



ESQUEMA DEL CIRCUITO DE POTENCIA

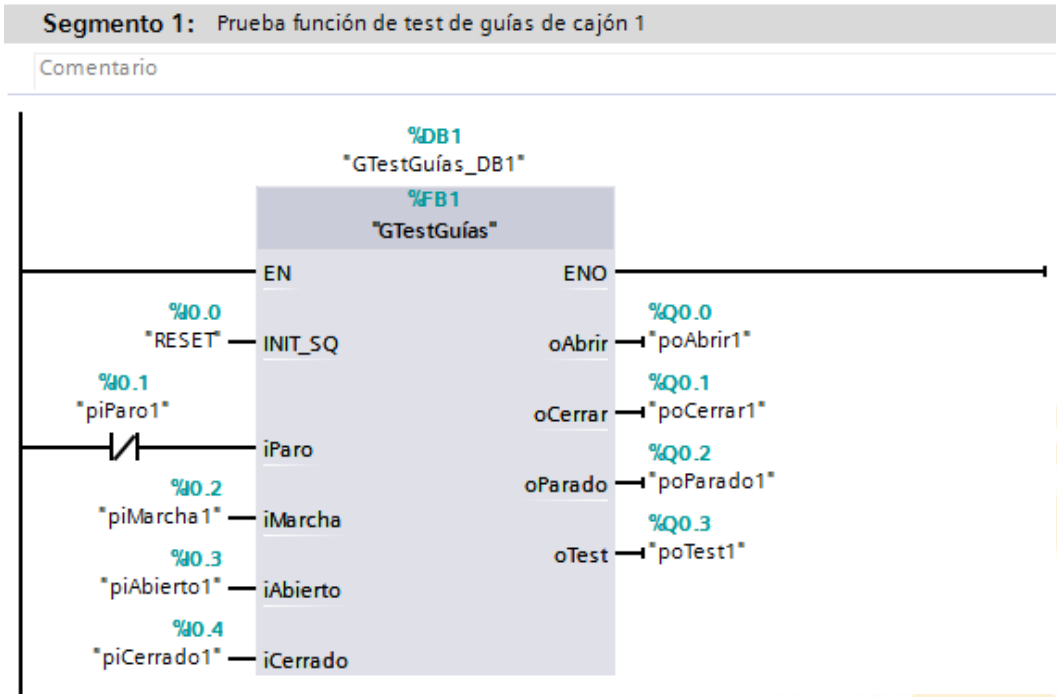




comillas.edu

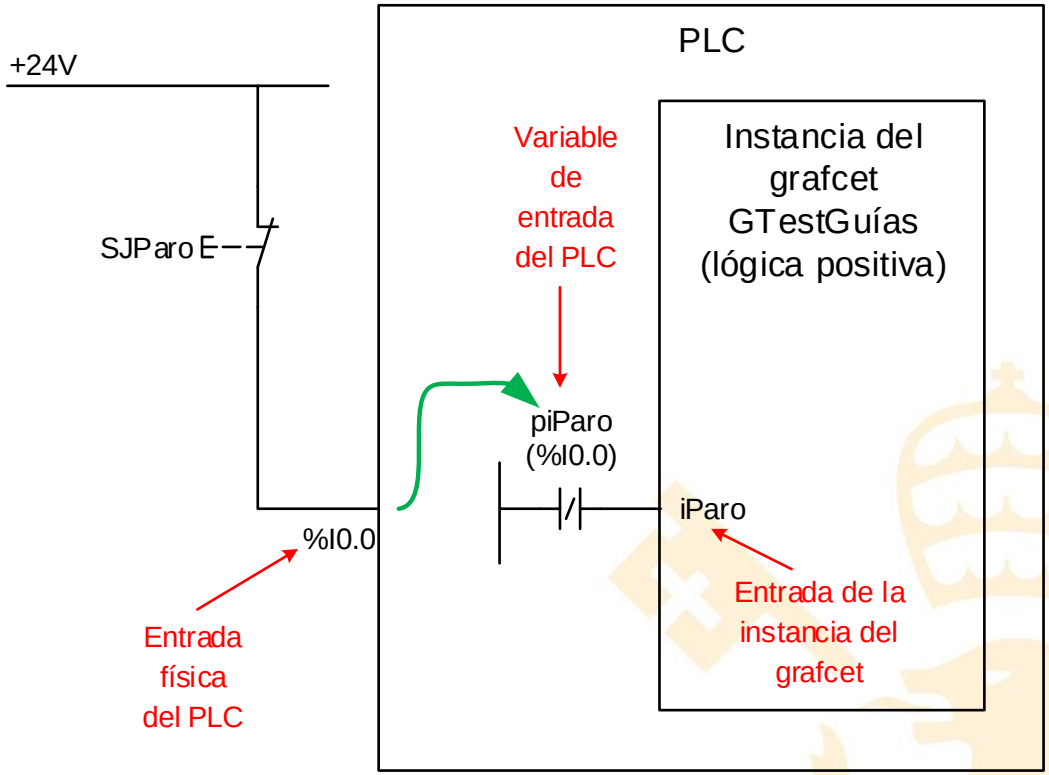
Solución de la orden de parada: parte software

- **Inversión de las entradas al bloque**
  - Dentro del bloque: lógica positiva
    - Fácil de entender
  - Fuera del bloque: lógica invertida
    - Un único punto donde se invierten las entradas
      - Fácil para solucionar errores



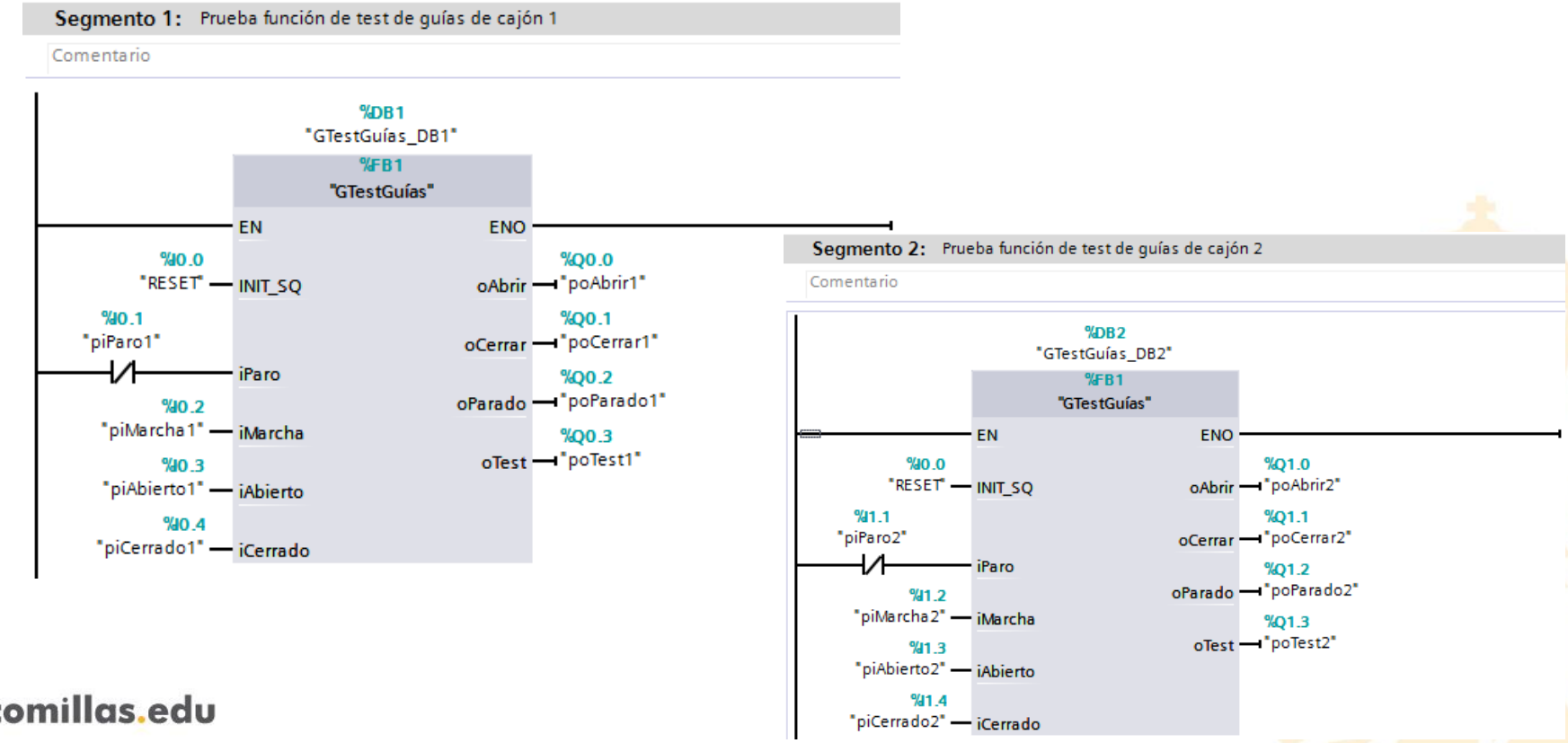
Interacción entre hardware y software en la parada

- **PLC lee entrada física conectada a pulsador SJParo con contactor NC**
  - Lógica negativa
    - 0 significa hacer algo
- **PLC guarda valor en variable de entrada piParo (alias de %I0.0)**
- **PLC ejecuta grafcet suministrando a iParo el valor negado de piParo**
  - Cambio a lógica positiva
    - 1 significa hacer algo
- **Orden de paro (pulsar SFJParo) = 1 lógico en la entrada iParo**



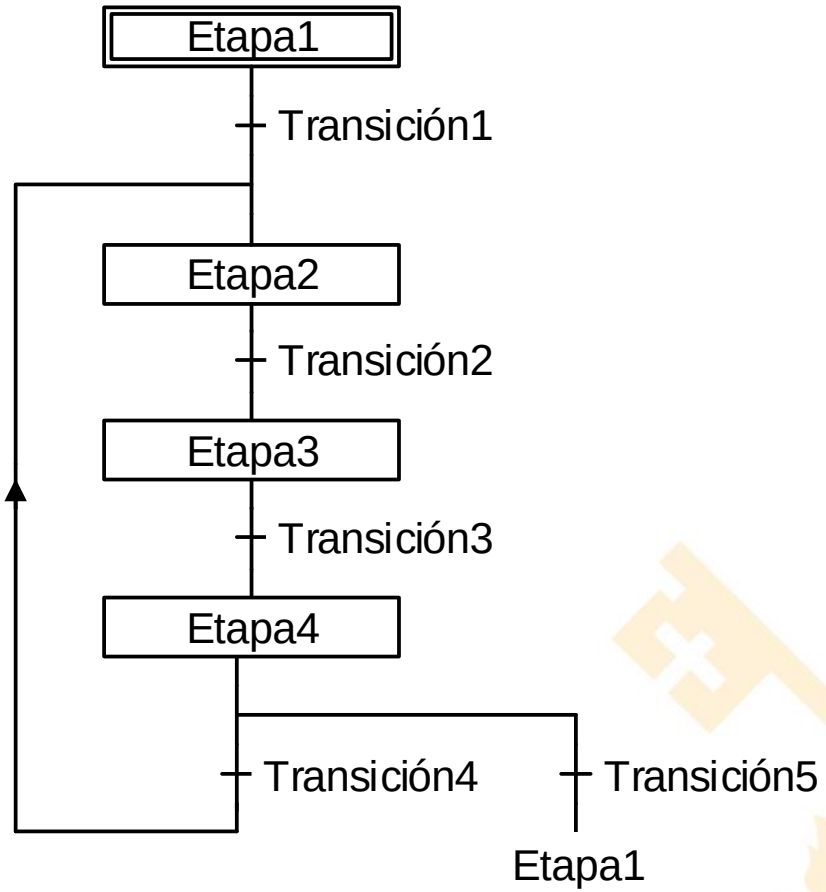
¿Cómo manejar dos sistemas de test de guías de cajón?

- **Mismo FB**
- **Dos DBs diferentes: 2 instancias del FB**
  - Cada uno almacena el funcionamiento de un sistema de test



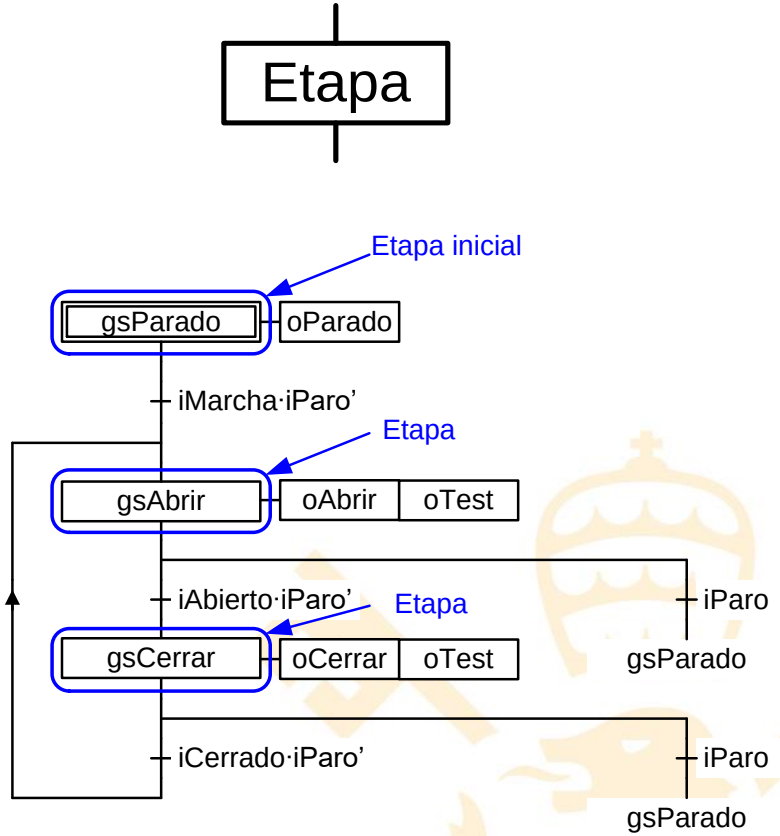
Segunda parte: Descripción detallada del lenguaje GRAFCET

- **Etapas**
- **Transiciones**
- **Uniones**
- **Condiciones**

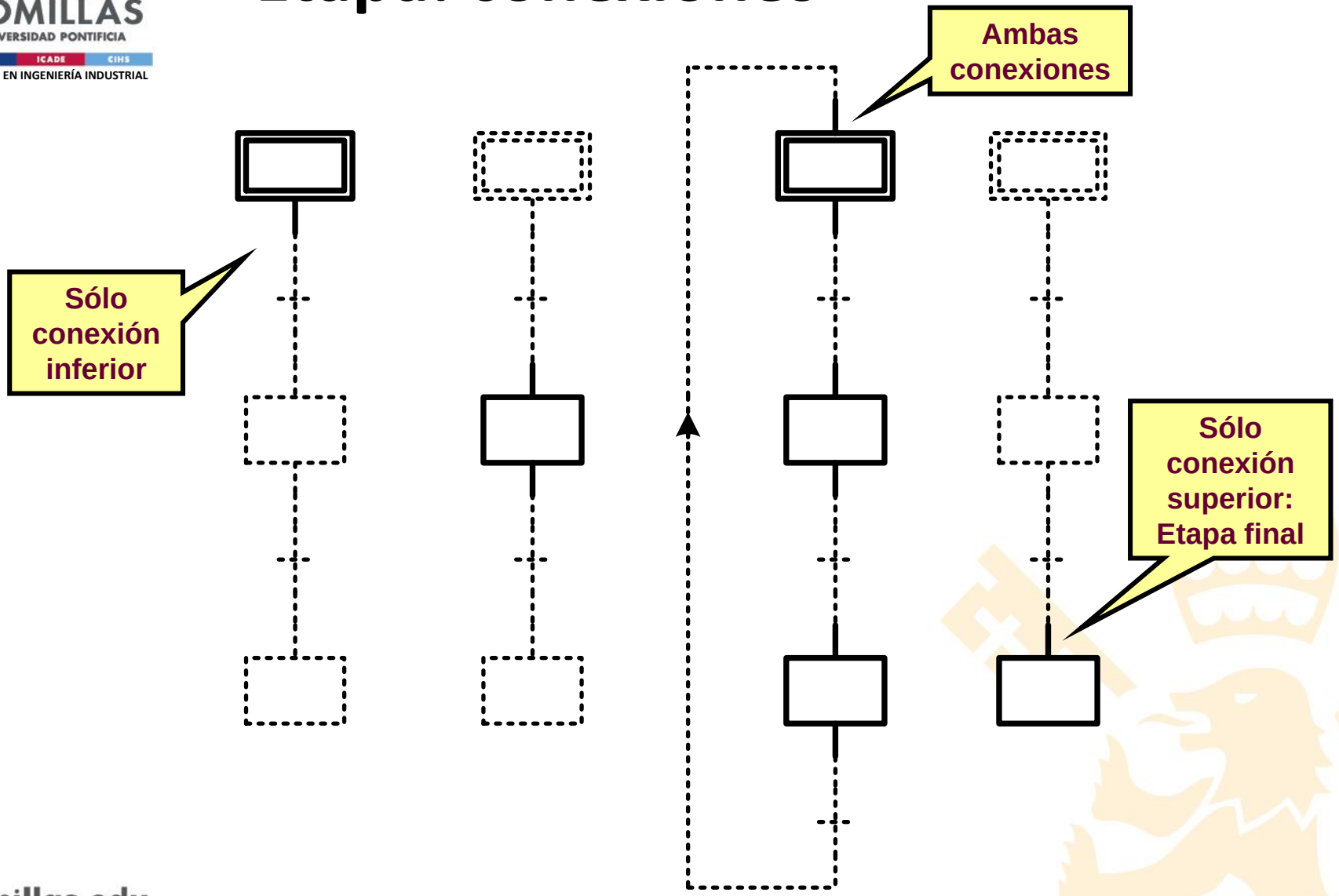


Etapa

- **Rectángulo con identificador en su interior**
  - Conexión superior si se necesita
  - Conexión inferior si se necesita
  - Doble rectángulo en caso de etapa inicial
- **Representa una situación en la secuencia que debe seguir un sistema: barrera bajada, barrera subiendo, barrera en avería,…**
- **Estados de la etapa**
  - Activa
  - Inactiva
- **El conjunto de etapas activas representa el estado actual del grafcet**



Etapa: conexiones

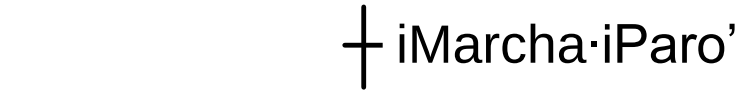


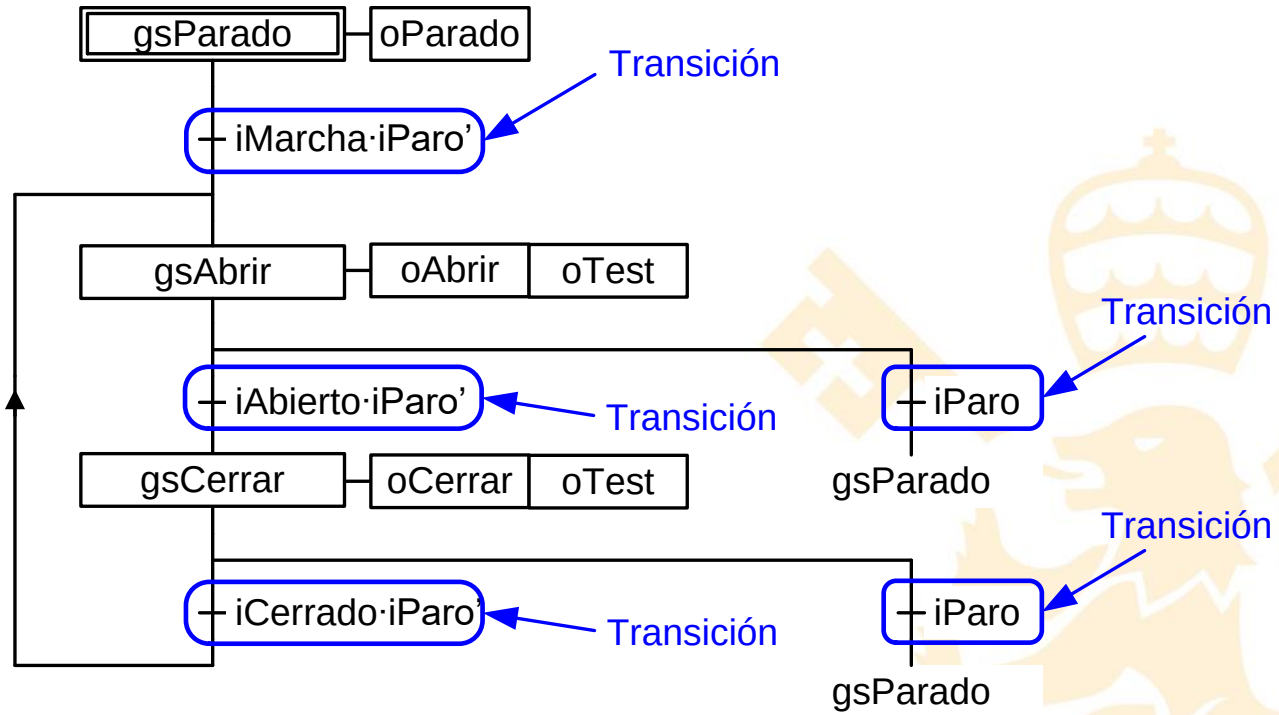
Etapa: identificador

- **Grafcet descriptivo**
  - Frase que describe la situación
  - Ejemplo:
    - Esperar orden, Parado, Moviendo hacia la derecha, Subiendo,…
- **Grafcet tecnológico**
  - Utilizable como identificador de variable
  - Prefijo “gs”
    - Grafcet step
  - Concatenación de palabras
    - Primera letra en mayúscula (lowerCamelCase)
    - Longitud no debe ser larga
    - Abreviaturas de dominio común
  - Ejemplo:
    - gsParado, gsAbriendo, gsAbierta, gsCerrando, gsAbierto, gsAvería, gsAbriendoPuerta1, gsAbriendoPuerta2

Transición

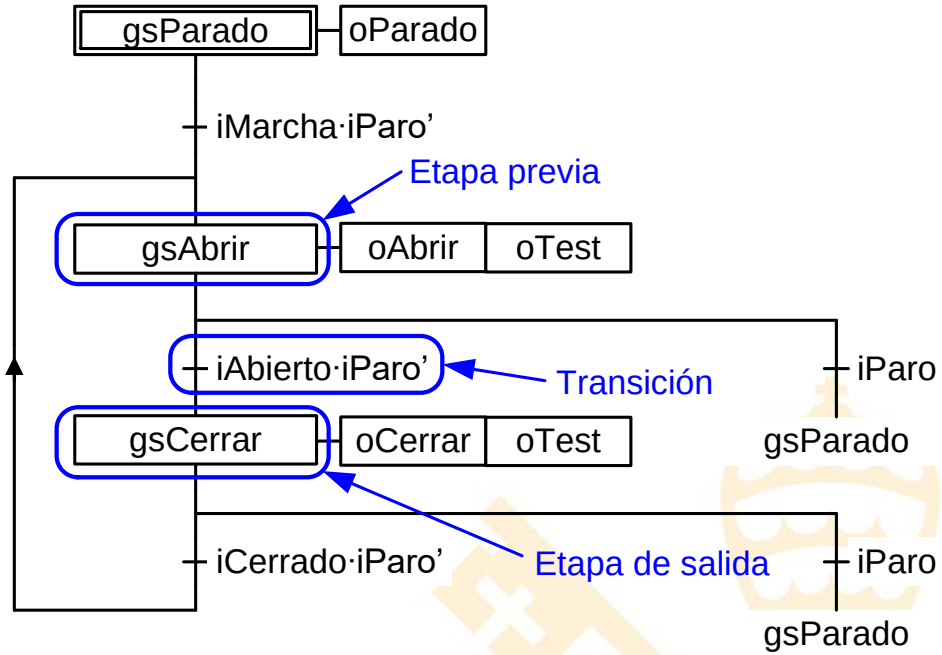
- **Línea corta horizontal que corta a vertical + texto con la condición lógica**
  - Condición para evolucionar de una etapa a otra (u otras)
    - Proposición lógica
    - O expresión lógica





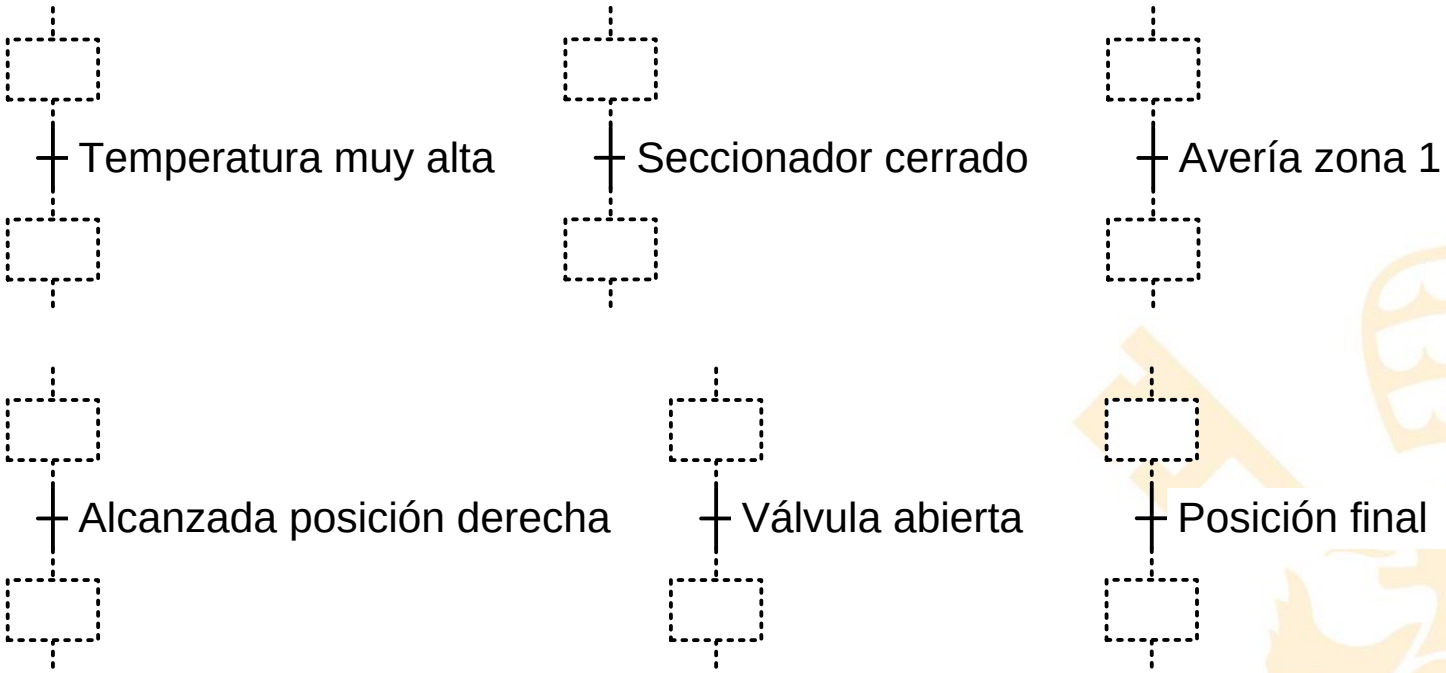
Transición: etapa previa y etapa de salida

- **Ejecución de una transición**
  - Si todas las etapas previas a la transición están activas según unión
  - Y se cumple la condición lógica de la transición
  - Se desactivan las etapas previas y se activan las etapas de salida según unión.
- **Estados de una transición: no activa, activa (pero no se ejecuta), se ejecuta**



Condición transición: Proposición lógica

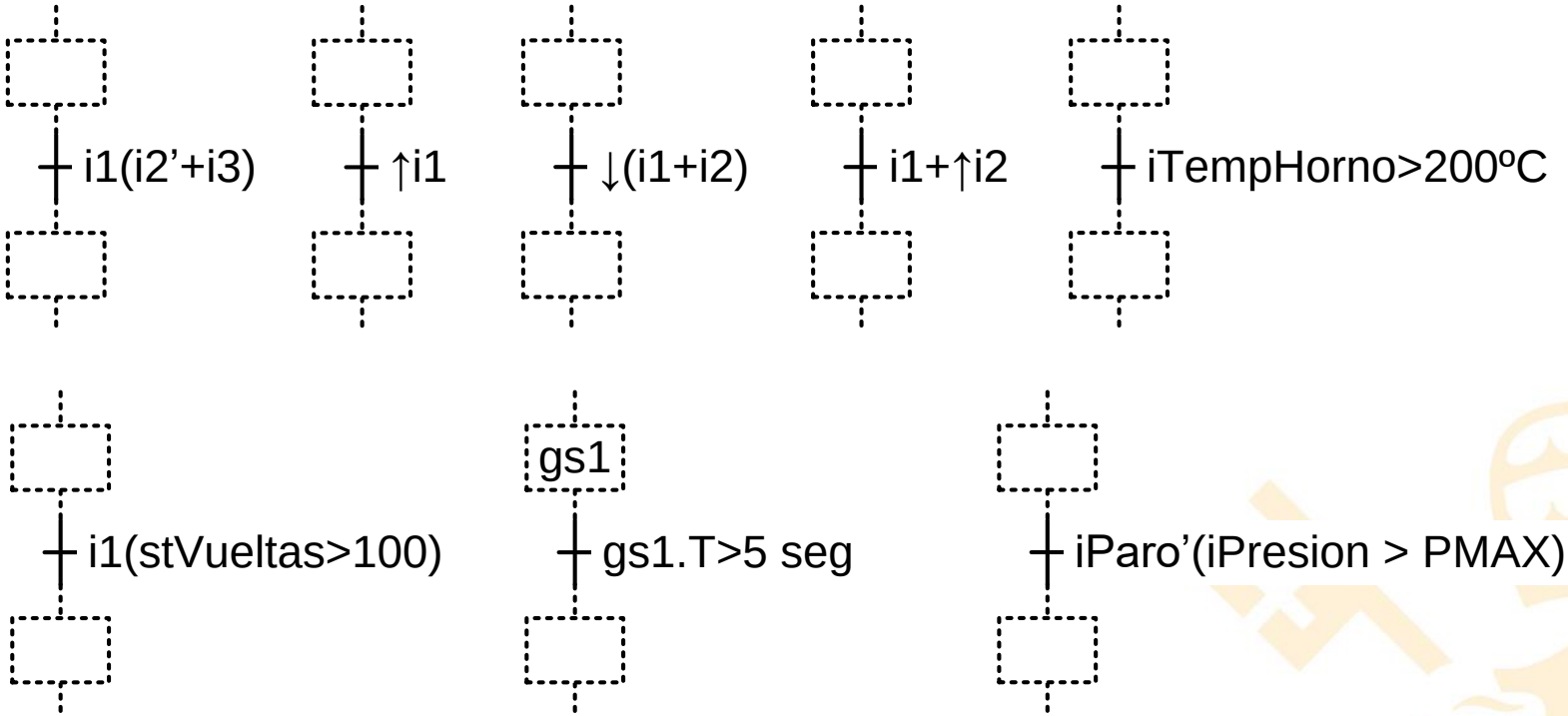
- **Frase que una vez evaluada es o verdadera o falsa.**
- **No utilizable en el grafcet tecnológico definitivo**



Condición transición: Expresión lógica

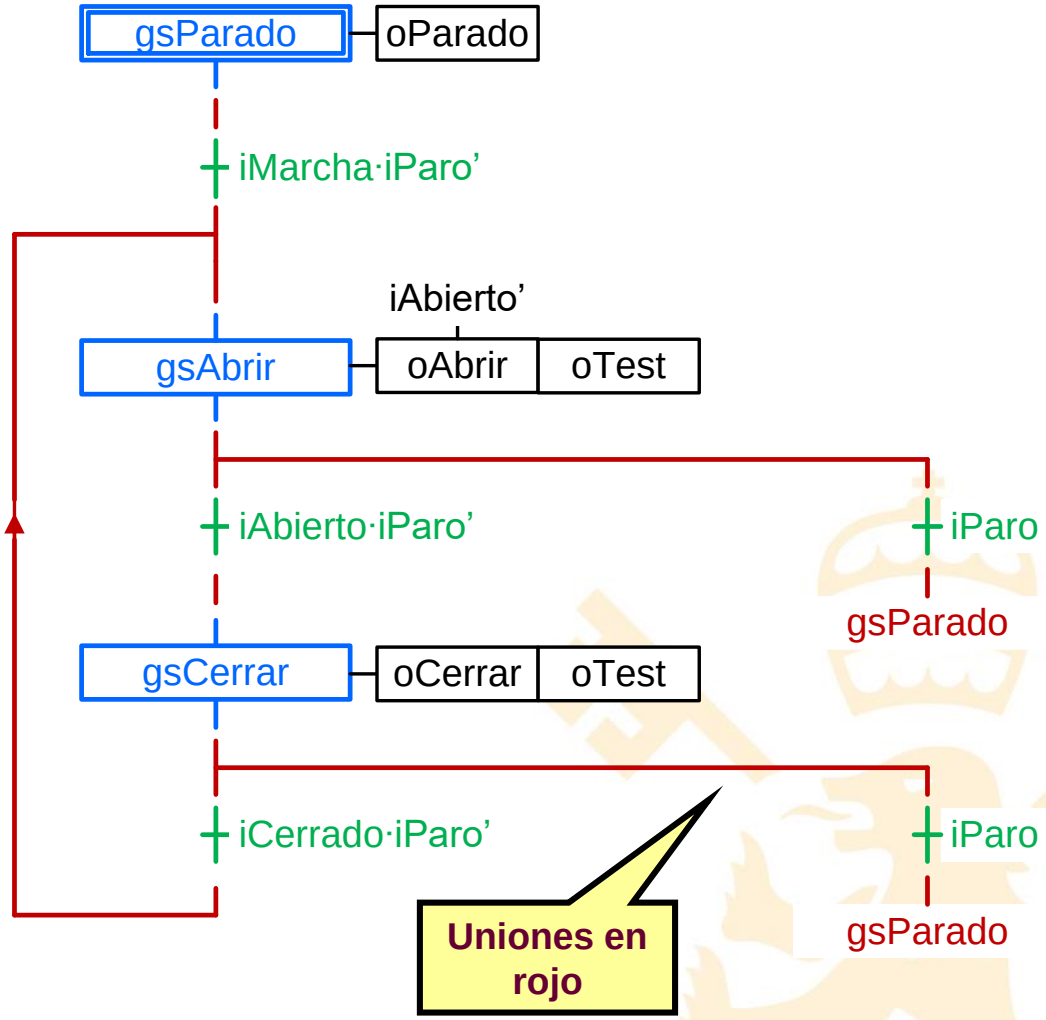
- **Expresión construida con:**
  - O una variable lógica
  - O varias variables lógicas combinadas mediante operadores lógicos
  - O dos variables del mismo tipo combinadas mediante operadores relacionales
  - O combinación de los anteriores combinadas mediante operadores lógicos
  - Las variables o parte de ellas pueden ser sustituidas por constantes
- **Operadores lógicos típicos: +, ·, ’, agrupación mediante ()**
    - Operadores lógicos de flanco: ↑ (positivo), ↓ (negativo)
- **Operadores relacionales: >, >=, <, <=, = (o ==), ≠ (o !=, o <>)**
- **Al sustituir las variables de la expresión por sus valores actuales la expresión toma uno de los posibles valores: verdadero, falso**

Ejemplos de expresiones lógicas



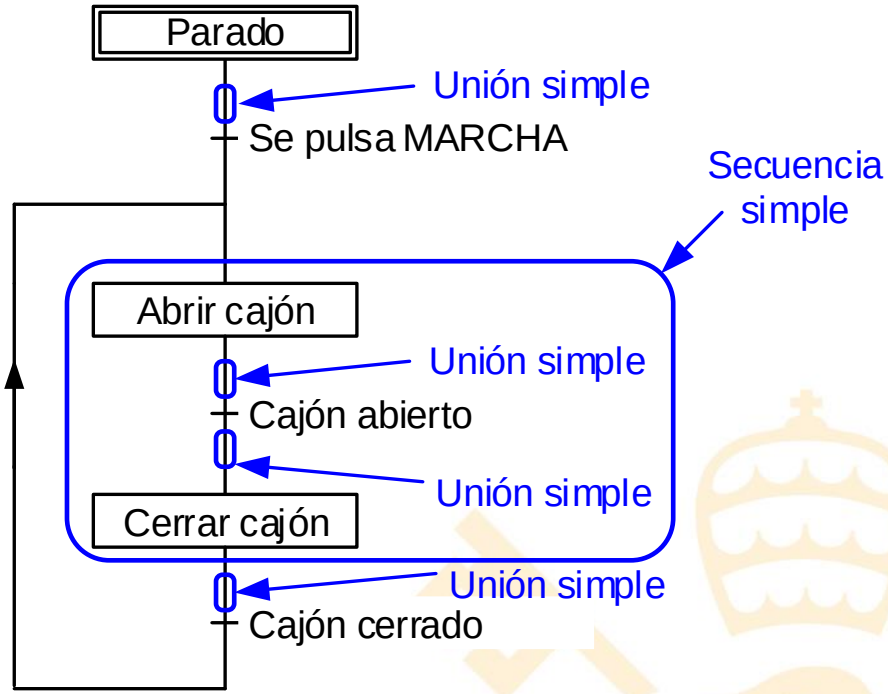
Unión

- **Símbolo que une:**
  - Etapas con transiciones
  - Transiciones con etapas
- **Tres tipos de uniones**
  - Unión simple
    - Secuencias simples
  - Unión de selección
    - Secuencias alternativas
  - Unión de sincronización
    - Secuencias simultáneas
- **Dos símbolos auxiliares**
  - Unión ascendente
  - Envío



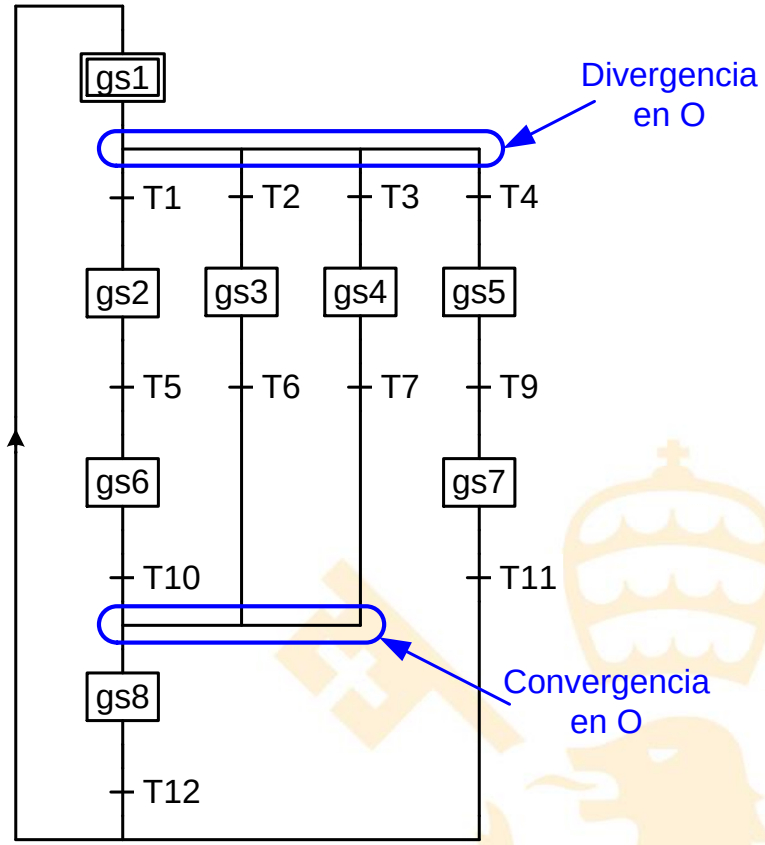
Unión simple

- **Une una etapa con una sola transición o una transición con una sola etapa**
- **Secuencia simple: conjunto de etapas y transiciones unidas sólo entre sí mediante uniones simples**
  - No hay caminos alternativos ni simultáneos
- **En una secuencia simple sólo puede estar activa una etapa**

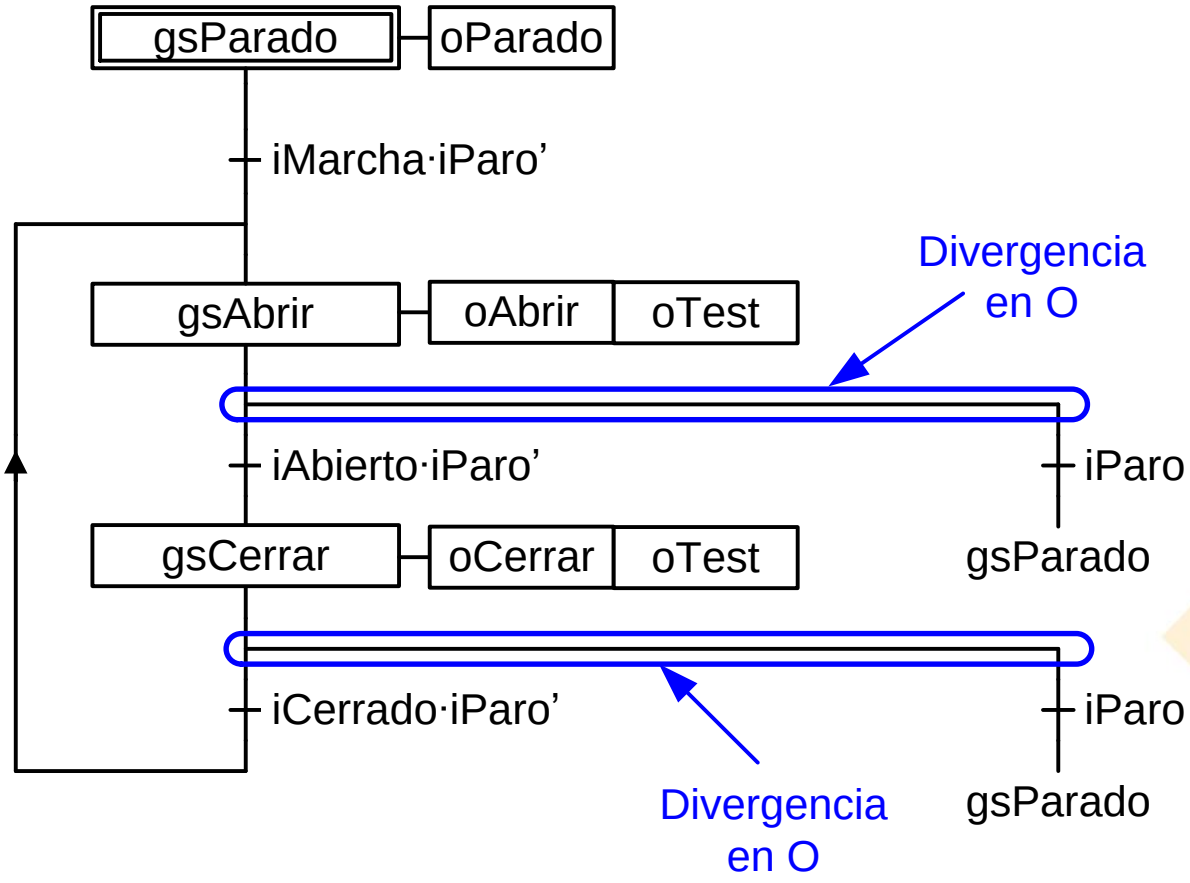


Unión de selección

- **Definir secuencias alternativas**
- **Dos tipos**
  - Divergencia en O
    - Seleccionar ir desde una etapa a una entre varias secuencias disponibles
    - Dibujo: conecta una etapa con varias transiciones
  - Convergencia en O
    - Llegar a una etapa desde diferentes secuencias
    - Dibujo: conecta varias transiciones con una etapa

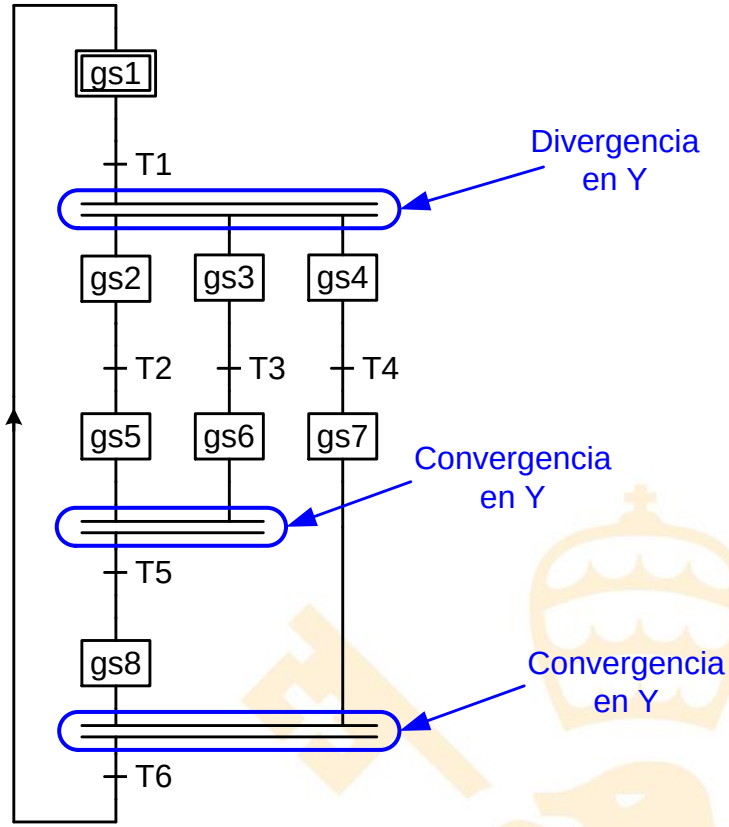


Ejemplo de unión de selección



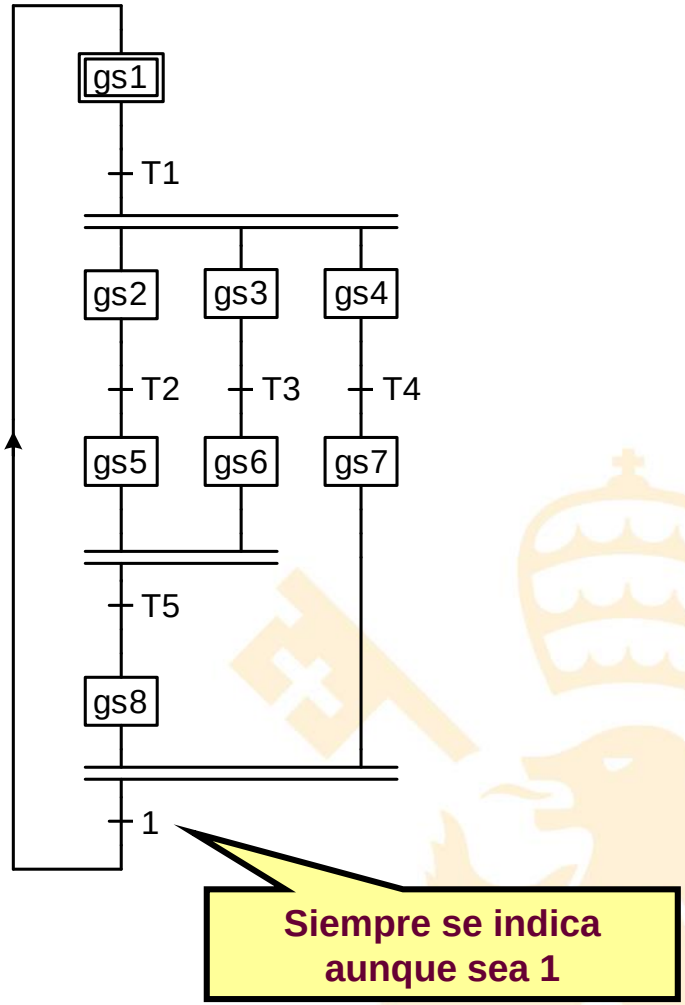
Unión de sincronización

- **Definir secuencias que se ejecutan de forma simultánea**
- **Dos tipos**
  - Divergencia en Y
    - Arrancar varias secuencias que se van a ejecutar en paralelo.
    - Dibujo: conecta una o varias transiciones con varias etapas
  - Convergencia en Y
    - Finalizar de forma sincronizada varias secuencias de etapas que se ejecutaban en paralelo.
    - Dibujo: conecta varias etapas con una o varias transiciones.

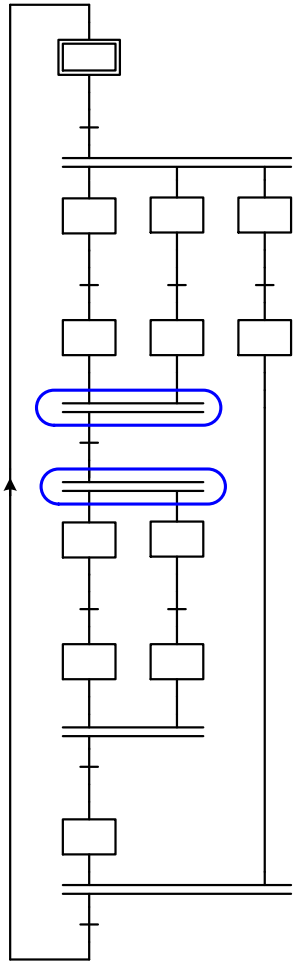


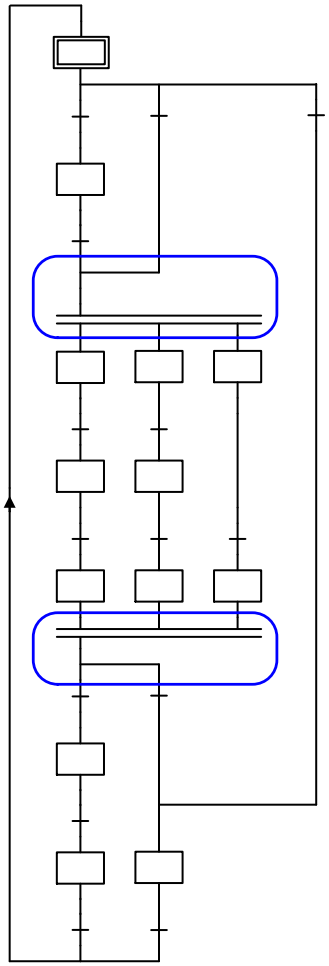
Caso típico en la convergencia en Y: transición sin condición

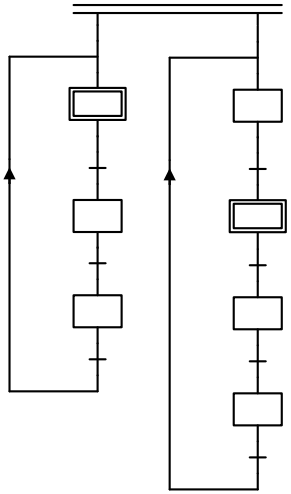
- **Para evolucionar sólo se necesita que las etapas previas a la convergencia en Y estén activas**
  - Basta con llegar a las etapas finales de las secuencias en paralelo
- **La condición lógica en la transición es 1: siempre se cumple**

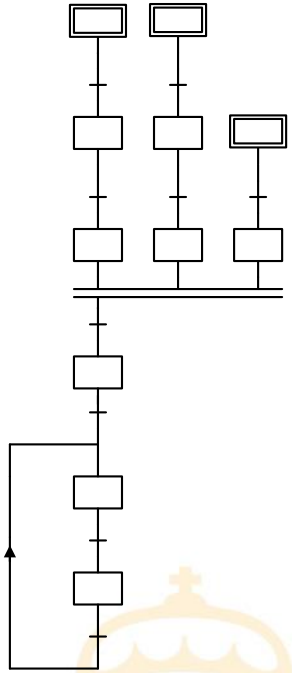


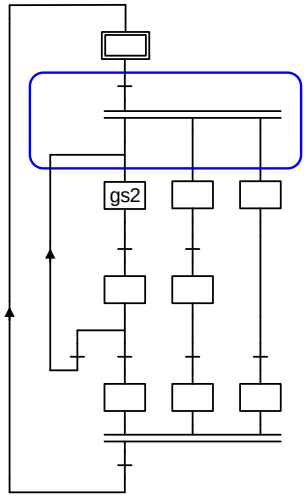
Unión de sincronización más complejas

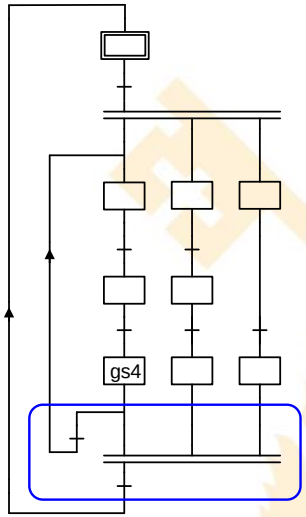


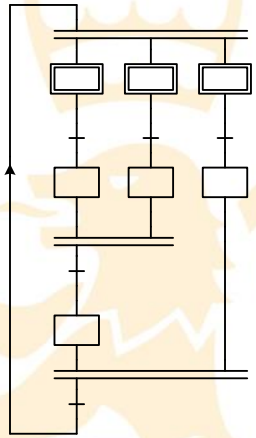






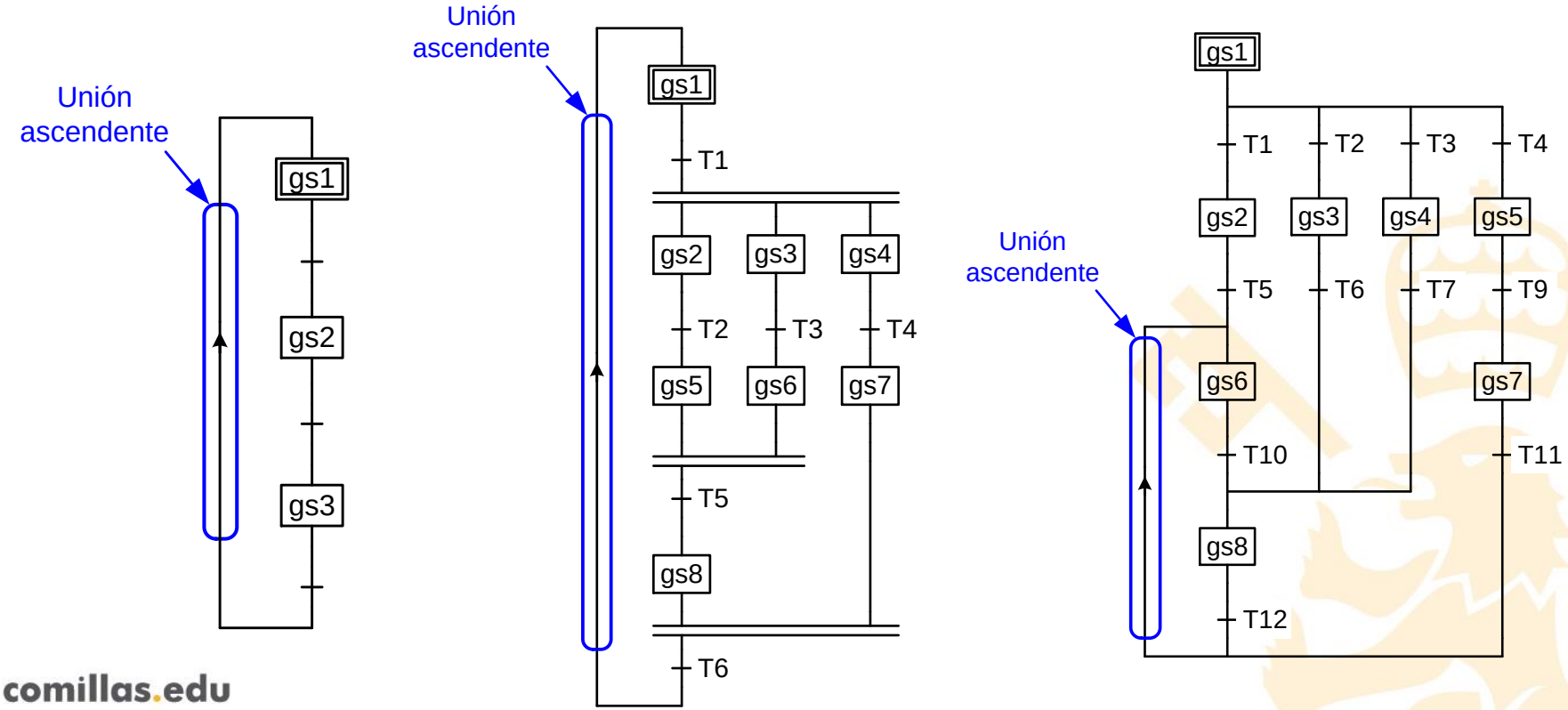




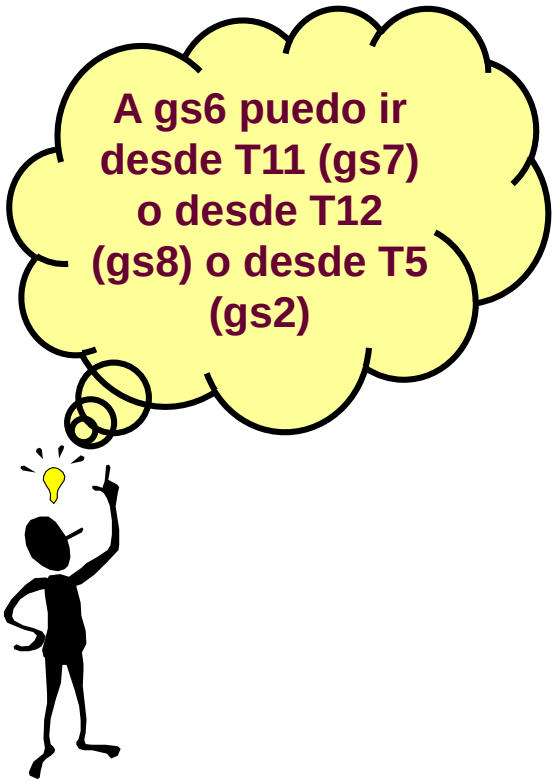


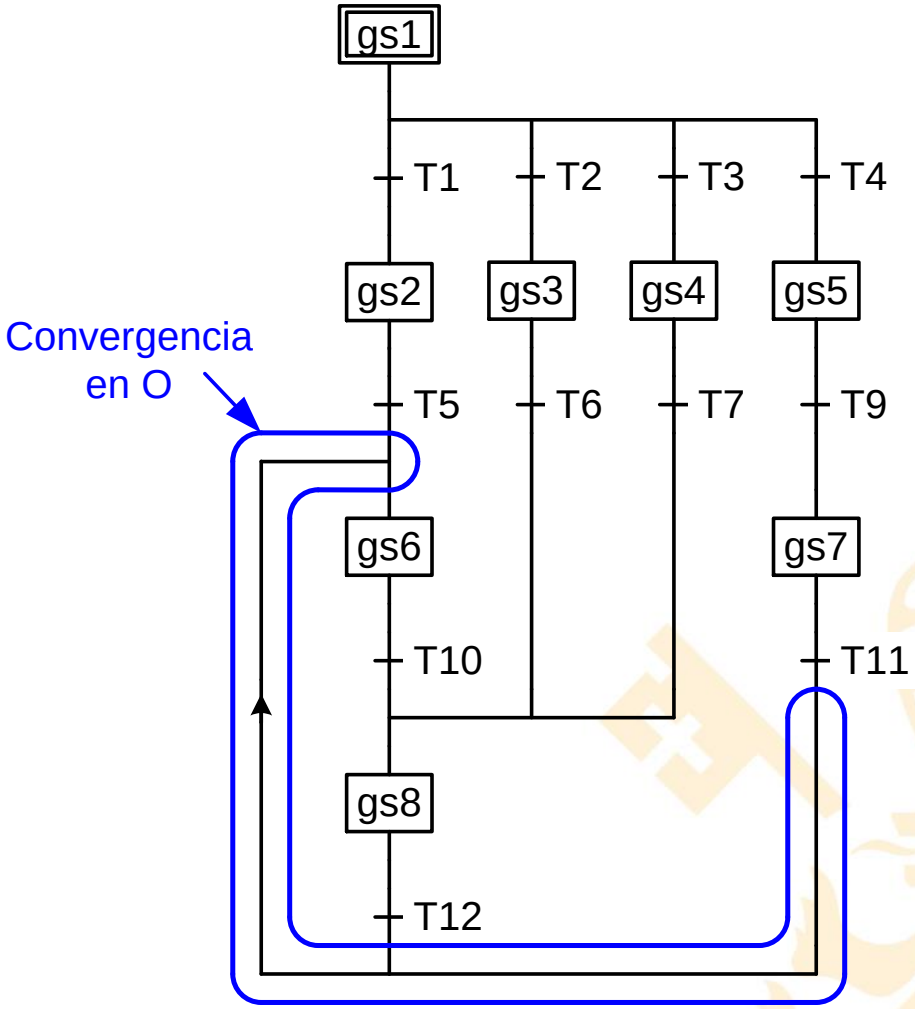
Unión ascendente

- **Ir desde una transición inferior a una etapa superior**
  - Cerrar una secuencia simple
  - Conectar una convergencia en O que está partida



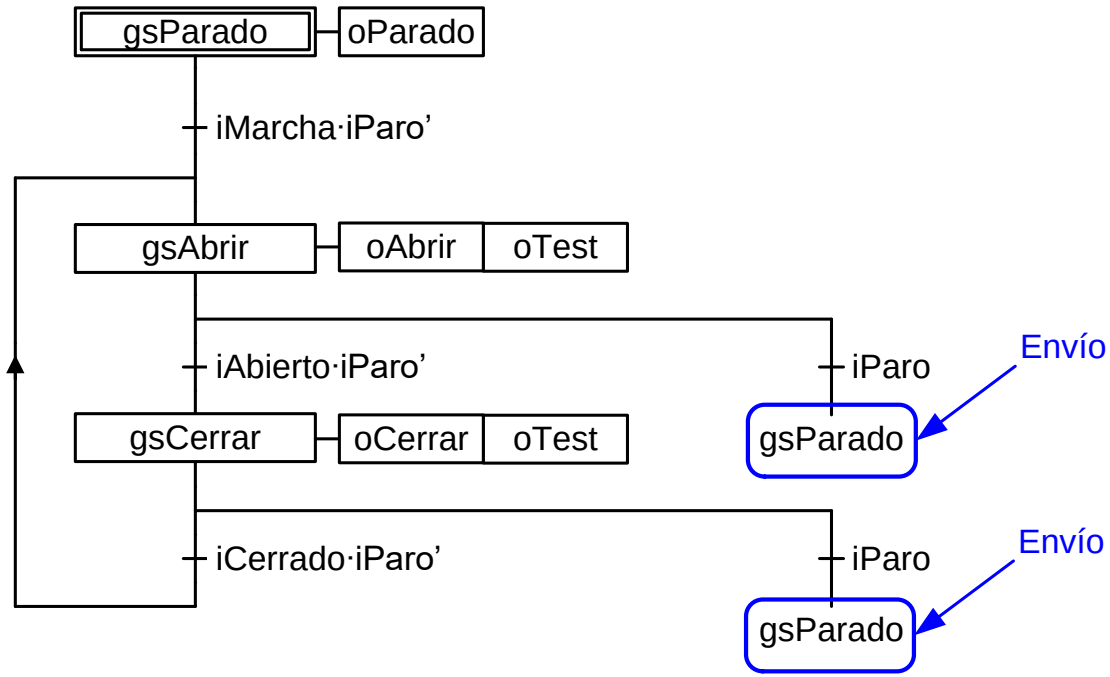
Detalle de la convergencia en O partida

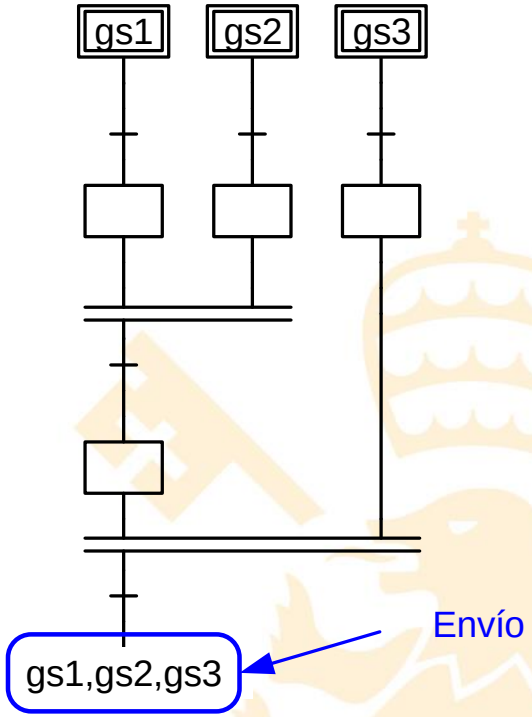




Envío

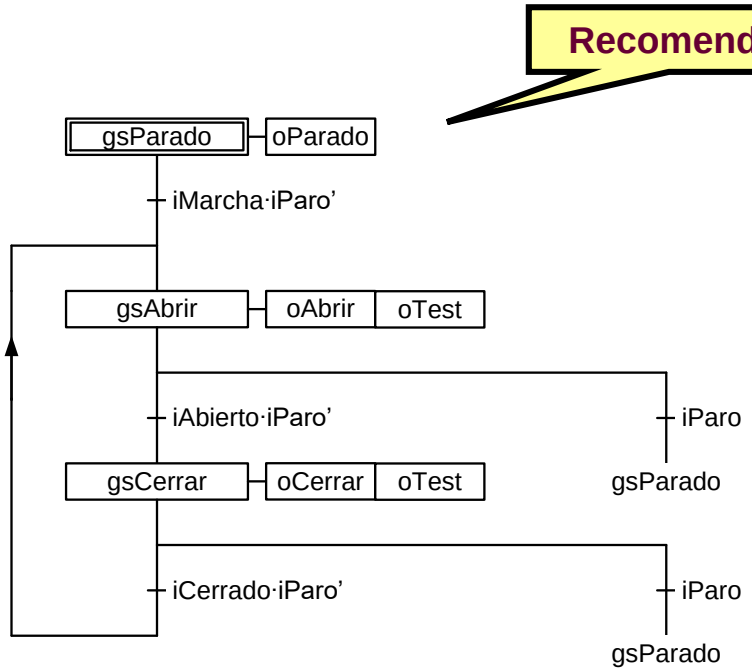
- **Conecta una transición con una etapa (o varias) a través de su identificador sin necesidad de línea**

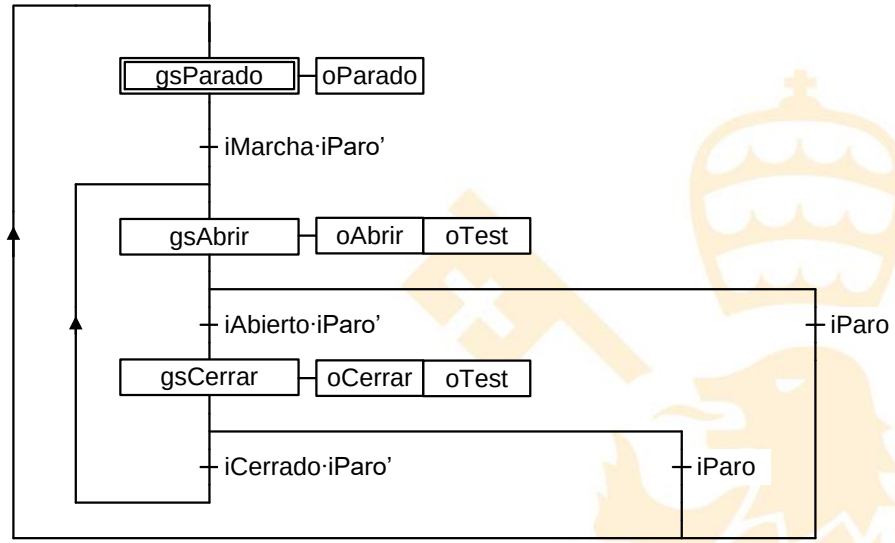




Envío y unión ascendente

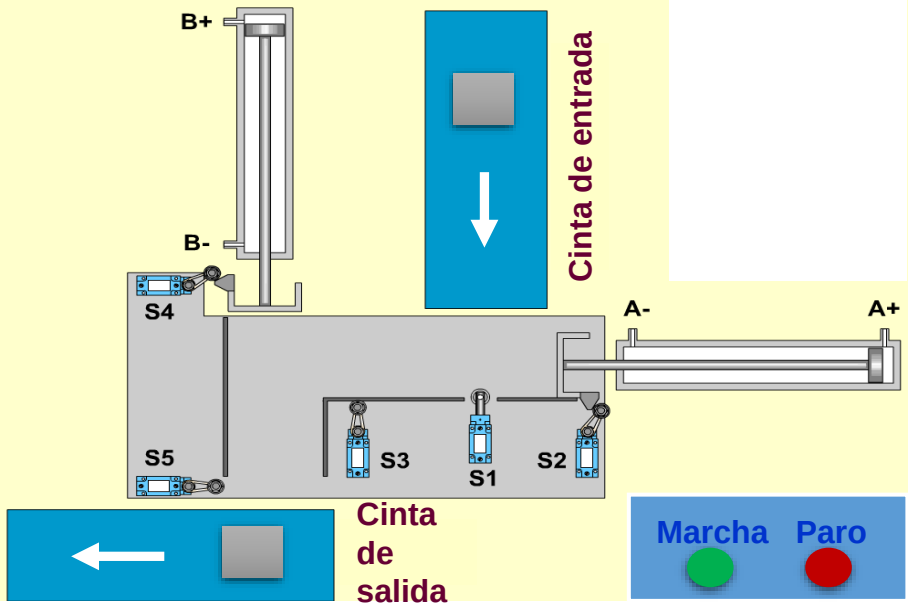
- **Unión ascendente para señalar el comportamiento principal**
- **Envío para comportamientos eventuales (paradas, averías, …) o para eliminar cruces de líneas**

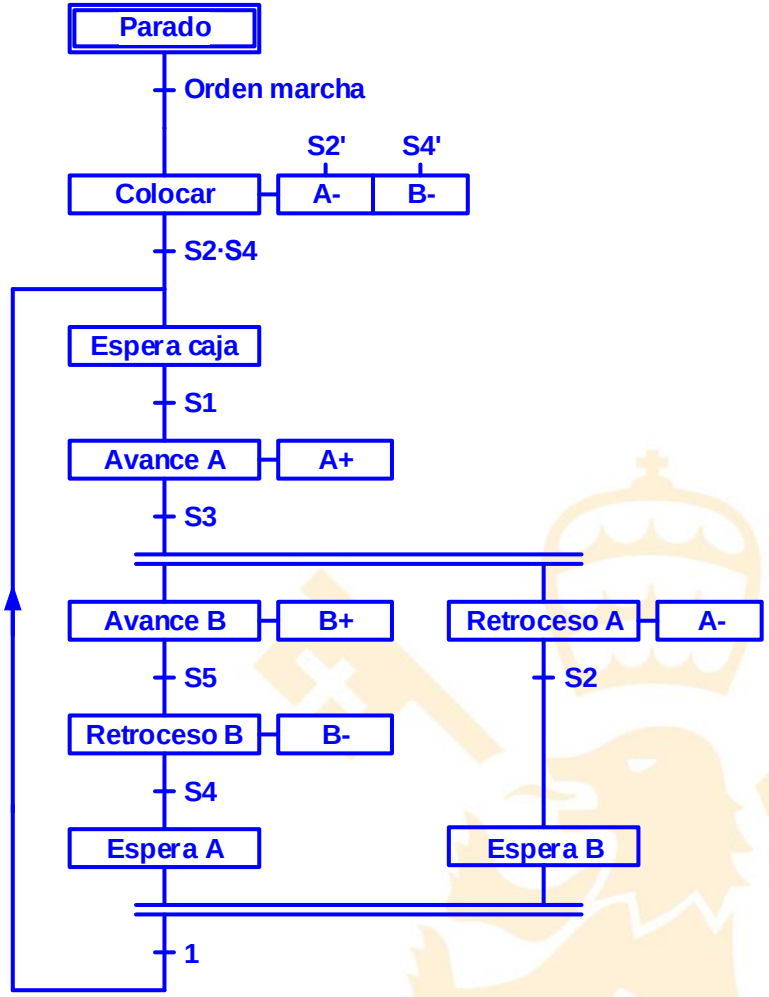




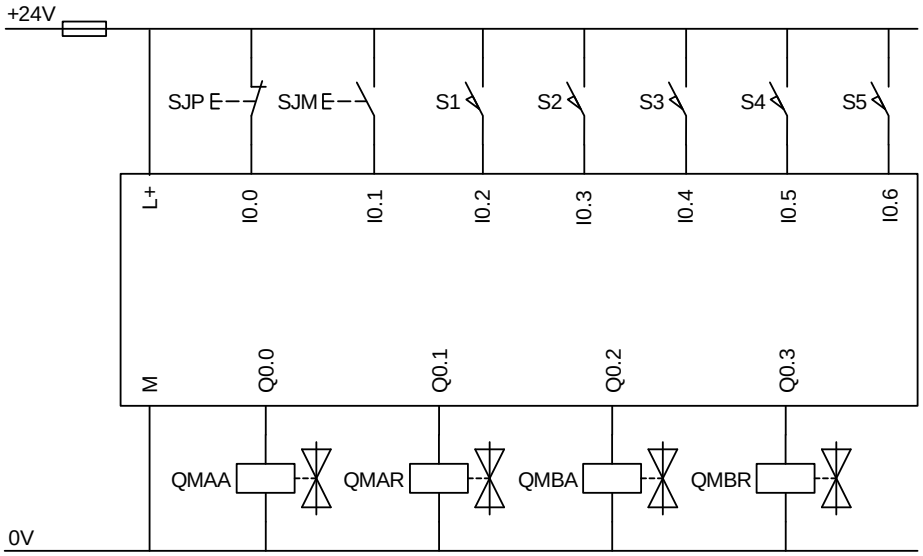
Ejemplo de unión de sincronización: descriptivo

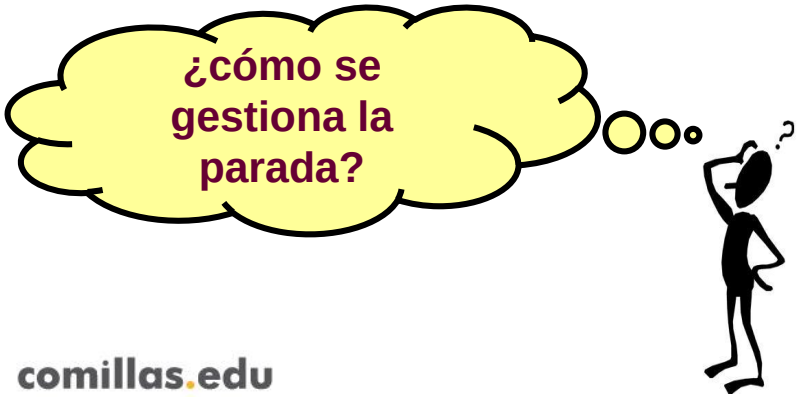
- Simulación del ejemplo: EPSI
- Neumática: cilindros (pistones) y electroválvulas
- Grafcet descriptivo

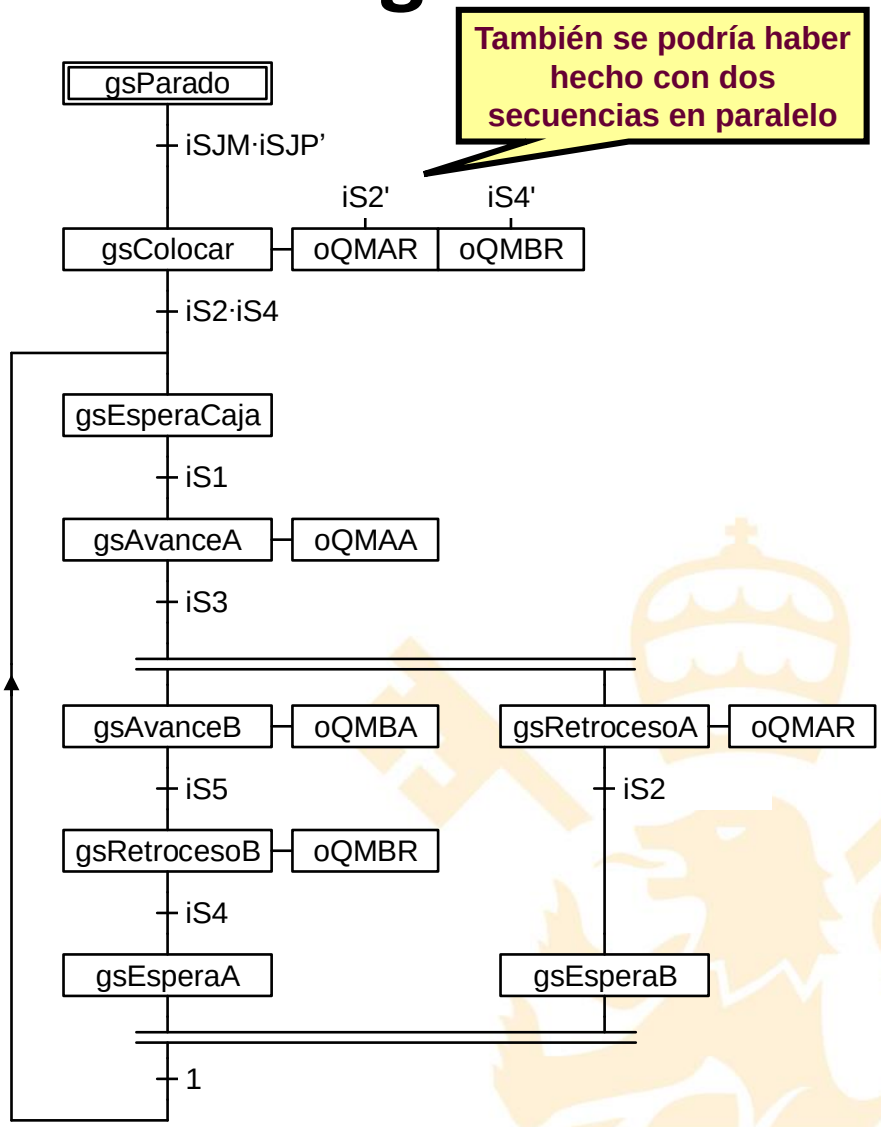




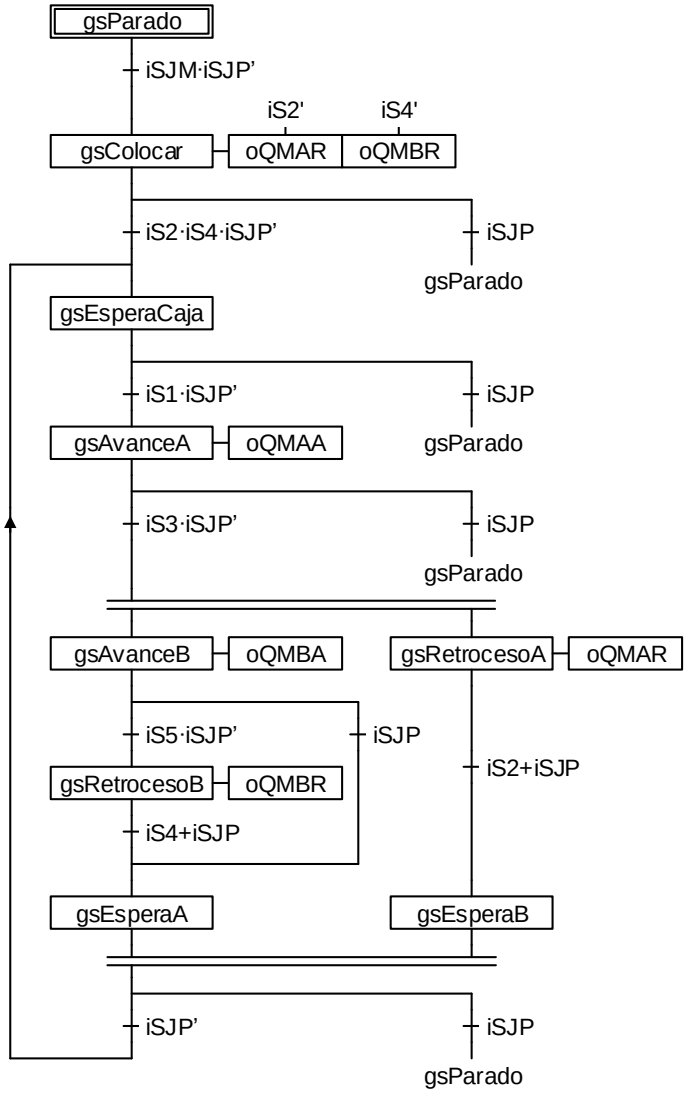
Ejemplo de unión de sincronización: tecnológico



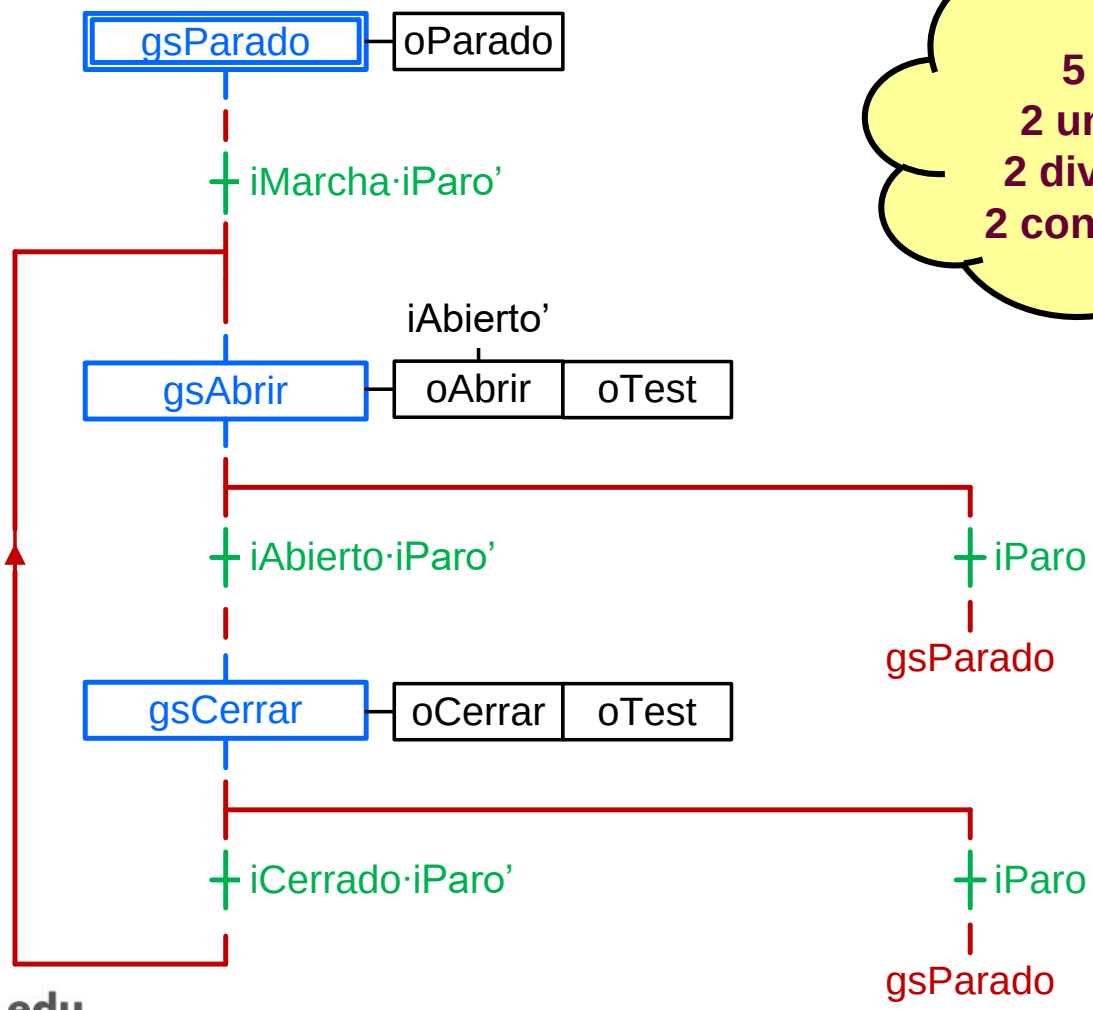


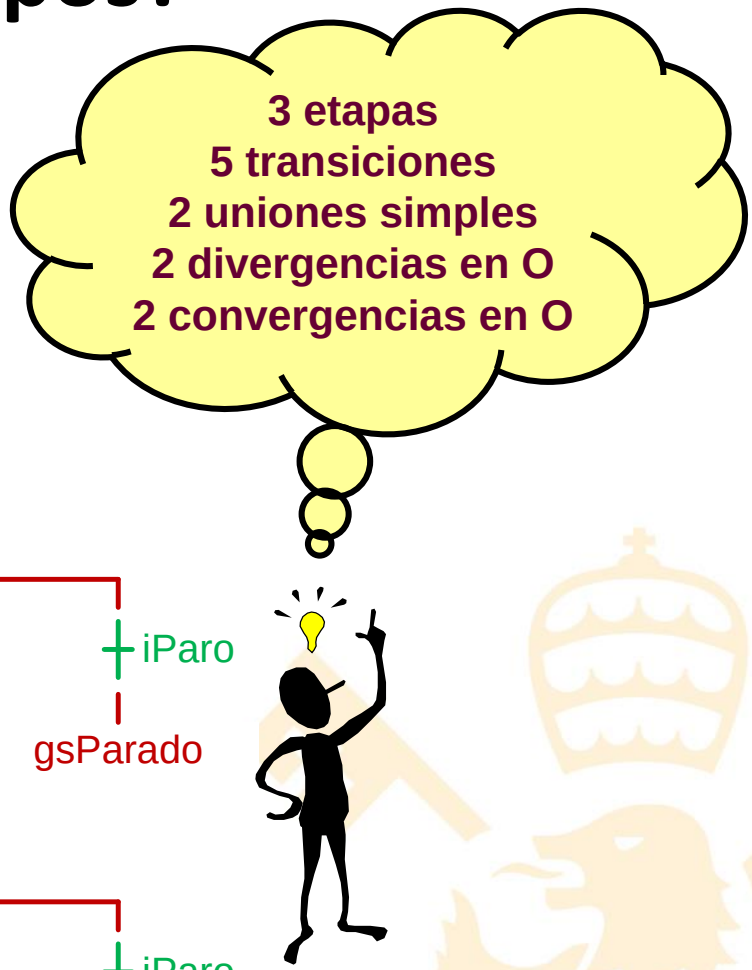


Ejemplo de unión de sincronización: parada



¿Cuántas uniones, etapas y transiciones?¿Tipos?

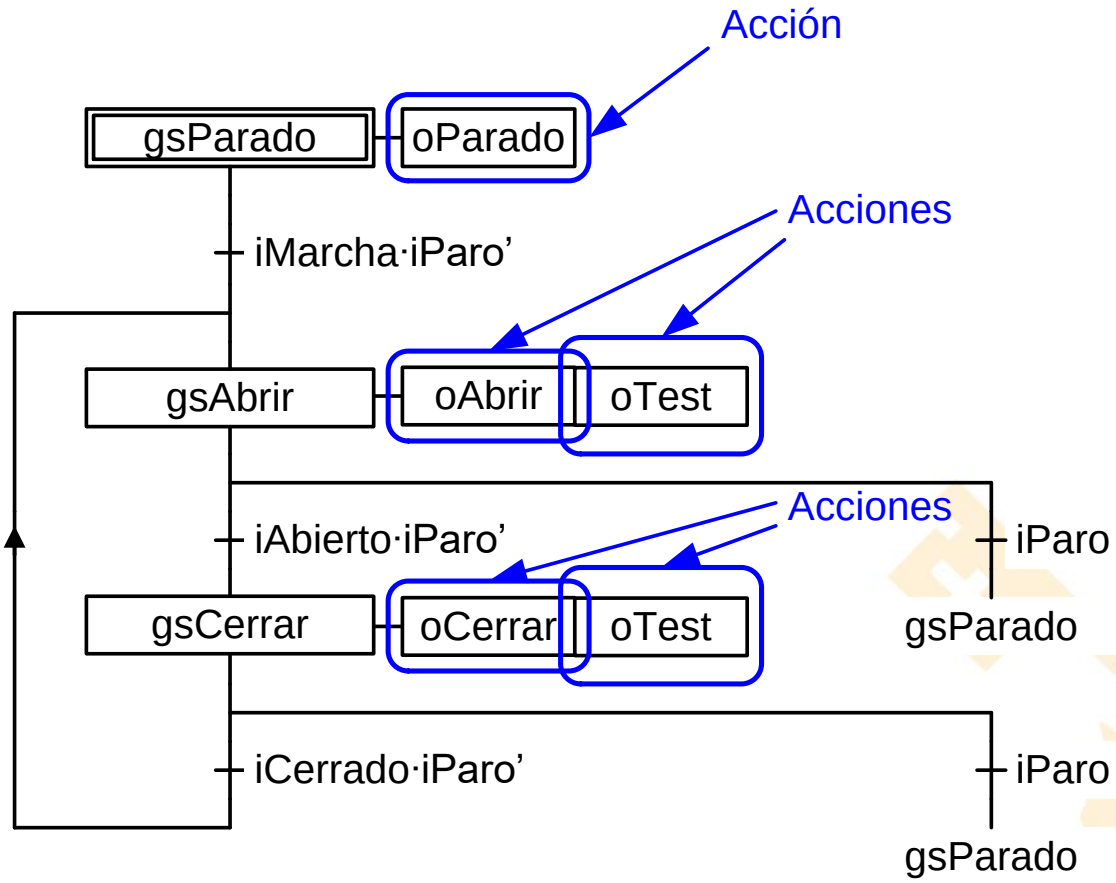




comillas.edu

Acción

- **Rectángulo con la acción a realizar**

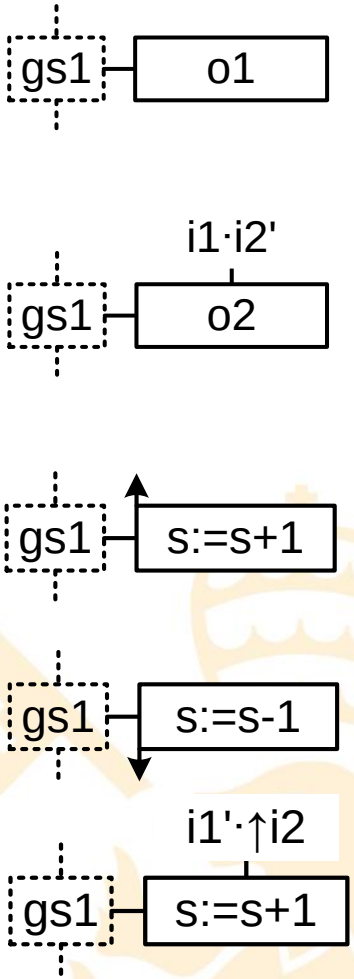


Descripción de la acción

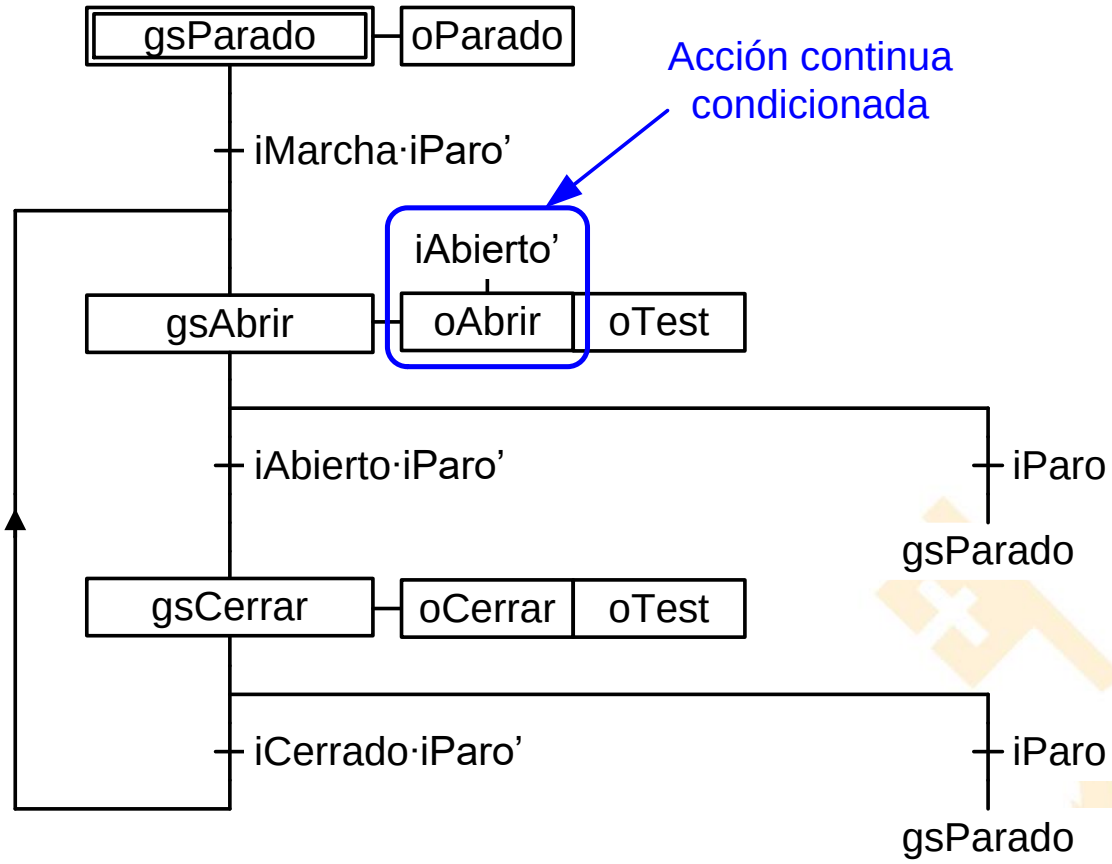
- **Texto libre**
  - Forma imperativa
    - Ejemplos: Abrir válvula, Piloto alarma encendido, Arrancar temporizador
  - Sólo grafcets descriptivos
- **Sentencia ejecutable desde un lenguaje de programación**
  - Ejemplo: s:=s+1, ArrancarTemporizador (T1,30s)
- **Identificador de variable**
  - Sólo acciones de tipo continuo
  - Ejemplo: oAbrir, oCerrar, oCerrarPuerta1, oCerrarPuerta2

Tipos de acción

- **Acción continua no condicionada**
  - La variable toma valor 1 cuando está asociada a una etapa que está activa
  - Si no está asociada a ninguna etapa activa, toma valor 0.
  - No se suelen utilizar sentencias del tipo s:=s+1
- **Acción continua condicionada**
  - Igual que continua pero toma el valor de la condición
    - La condición no puede tener eventos (flancos)
  - No se suelen utilizar sentencias del tipo s:=s+1
- **Acción asociada a la activación de la etapa**
  - Se ejecuta la sentencia al entrar en la etapa
  - No se utiliza variable
- **Acción asociada a la desactivación de la etapa**
  - Se ejecuta la sentencia al salir de la etapa
  - No se utiliza variable
- **Acción asociada a evento en la etapa**
  - Se ejecuta la sentencia cada vez que se da el evento
    - Eventos en la condición: flancos
  - No se utiliza variable

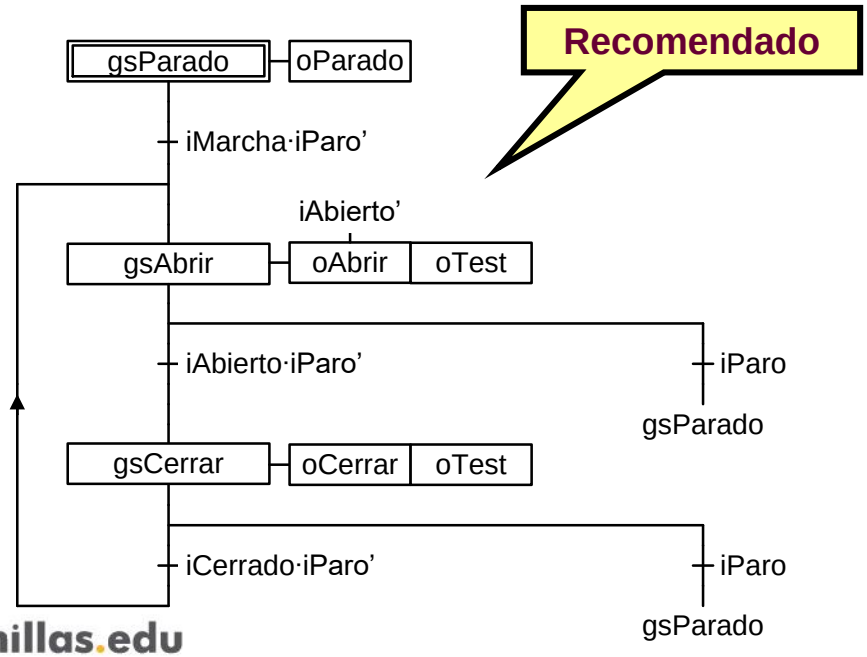


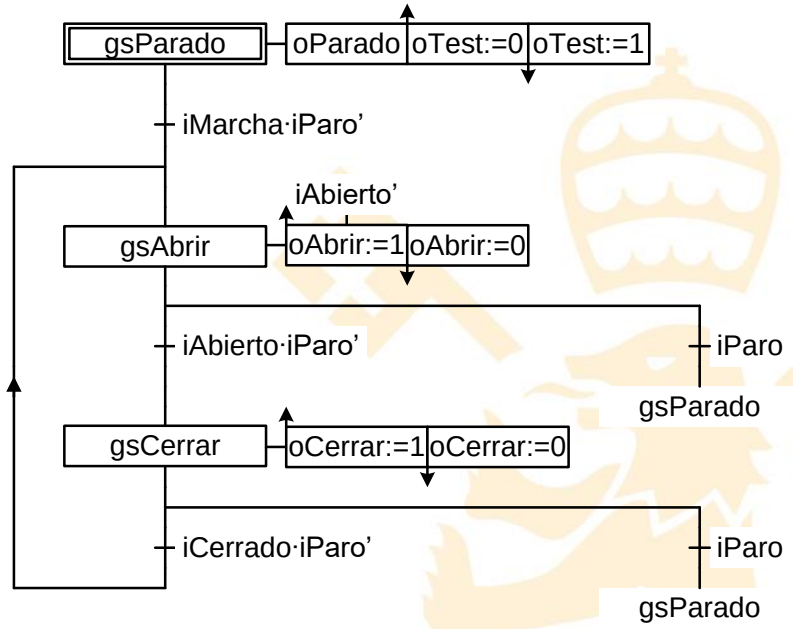
Ejemplo de acción continua condicionada



Acciones de tipo continuo y asociadas a entrada y salida

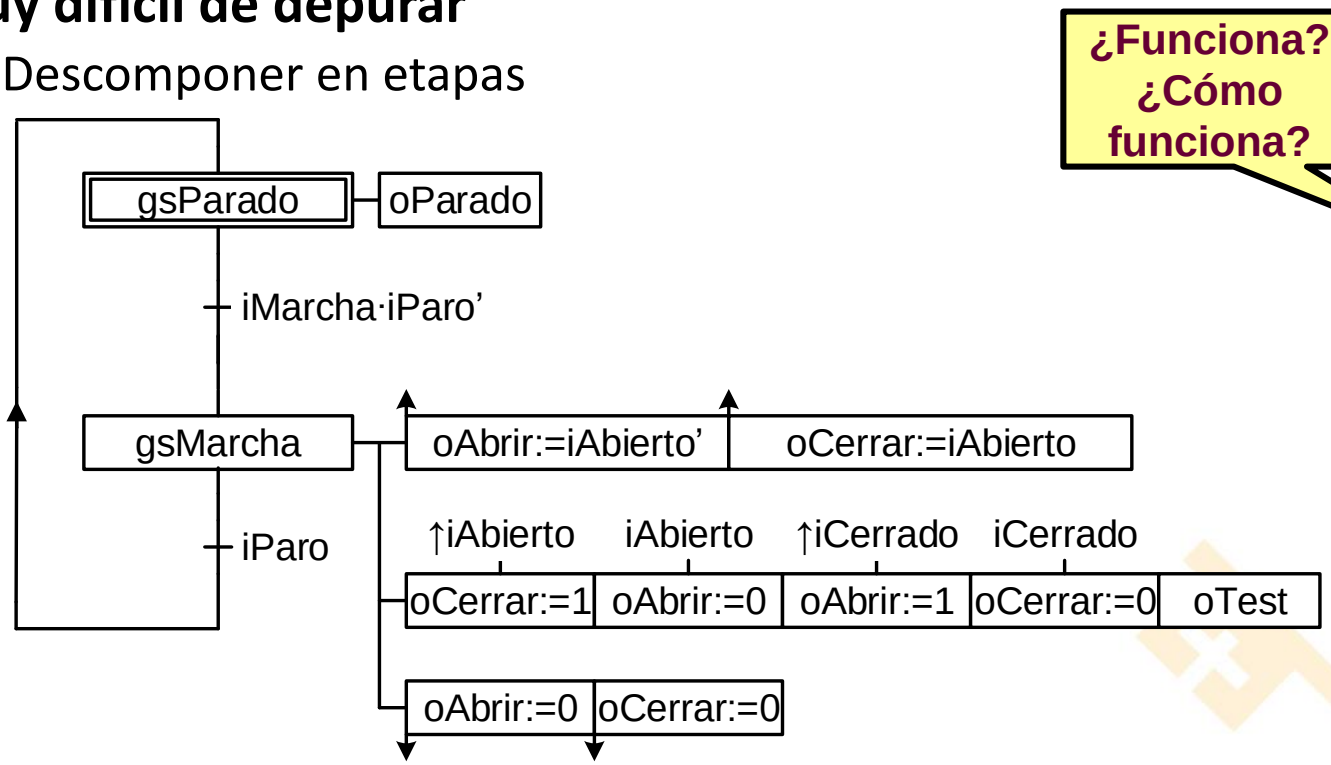
- **Son equivalentes**
- **Mejor asignar las acciones dentro de la etapa frente a distribuir las acciones entre etapas**
  - Si se aumenta el número de etapas, hay más garantía de buen funcionamiento
  - Etapas construidas son etapas definitivas





Error: secuencias de acciones dentro de una etapa

- **Muy difícil de depurar**
  - Descomponer en etapas



- **No está permitido su uso en la asignatura**
  - Se puntúa como 0 la pregunta.

Reglas de sintaxis y ejecución (I)

Reglas cuyo no cumplimiento se considera muy grave: 0 en la solución

1. En todo camino deben alternarse etapas y transiciones
   - Dos etapas nunca pueden estar conectadas por una unión
   - Dos transiciones nunca puede estar conectadas por una unión
2. Varias etapas sólo pueden compartir una misma transición a través de una divergencia en Y o una convergencia en Y
3. En una secuencia simple no puede haber más de una etapa activa.
4. Todas la etapas deben ser alcanzables
5. Todo grafcet tiene que tener al menos una etapa inicial
6. No pueden aparecer acciones de tipo continuo asociadas a inicio o fin de etapa o a evento
7. Sólo utilizar los símbolos gráficos del lenguaje GRAFCET
8. La etapa previa y la etapa de salida no puede ser la misma
   - Al salir de una etapa se va a una etapa diferente

Reglas de sintaxis y ejecución (II)

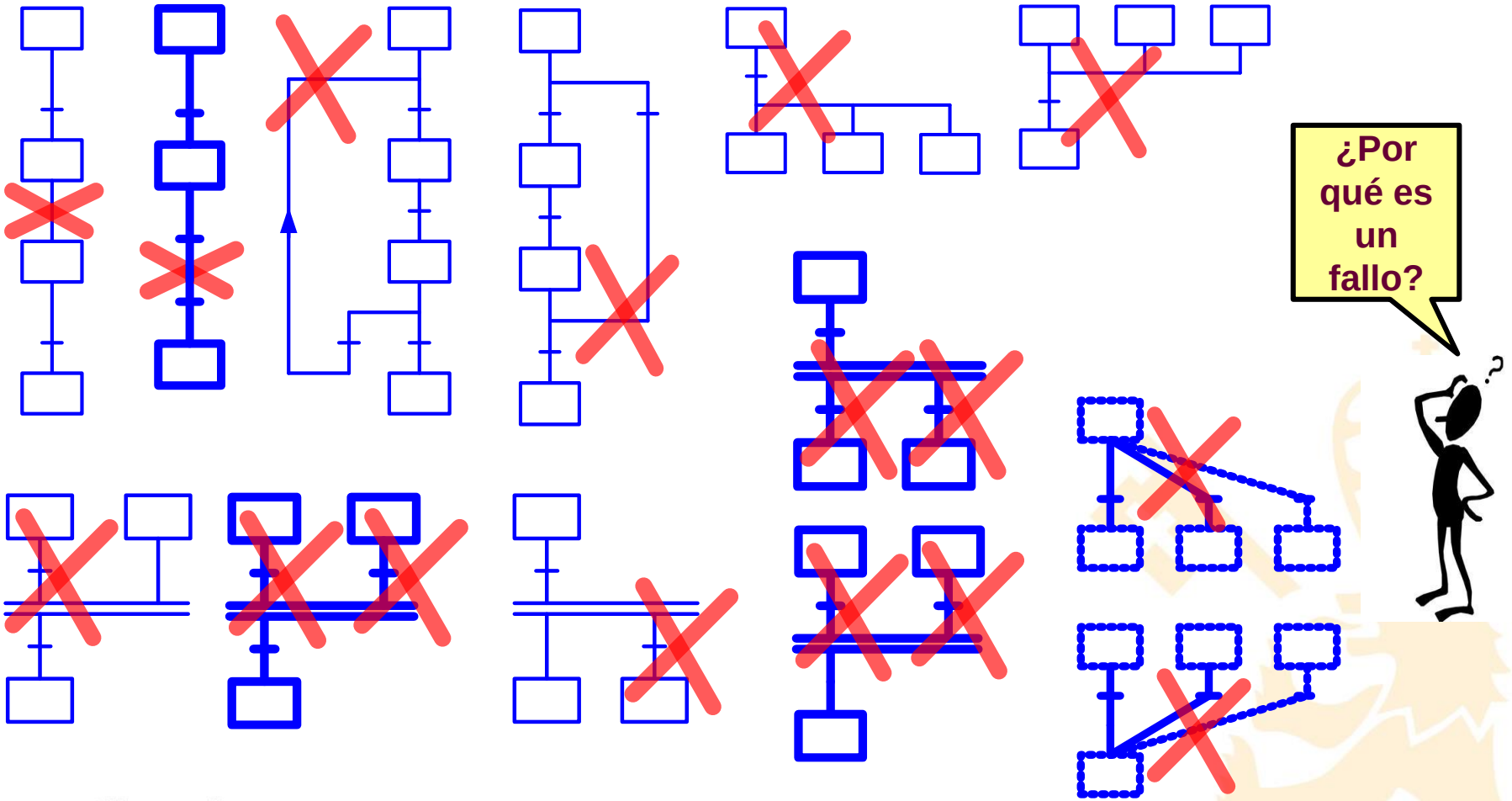
Reglas cuyo no cumplimiento se considera grave: puntuación negativa

1. **Las transiciones de una divergencia en O deben ser mutuamente excluyentes**
2. **Acciones que nunca son ejecutables aunque se cumplan sus condiciones lógicas asociadas.**

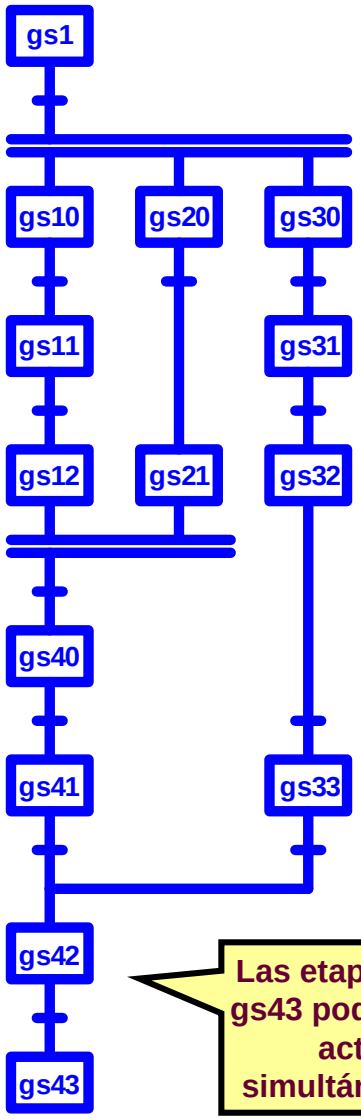
Recomendaciones

- **Todas las divergencias en O deberían terminar a través convergencias en O**
- **Todas las divergencias en Y deberían terminar a través de convergencias en Y**
- **Grafcets estéticos**

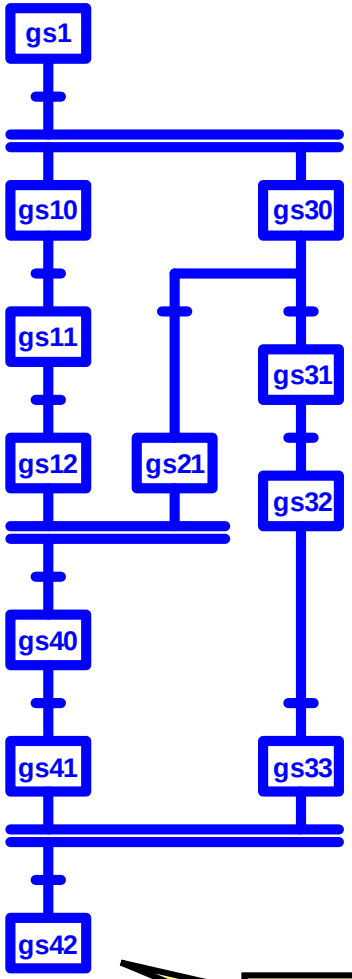
Errores en los grafcets



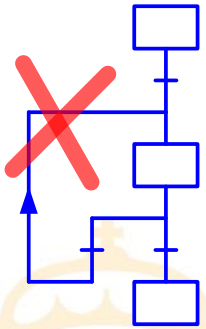
Más fallos



Las etapas gs42 y gs43 podrían estar activas simultáneamente

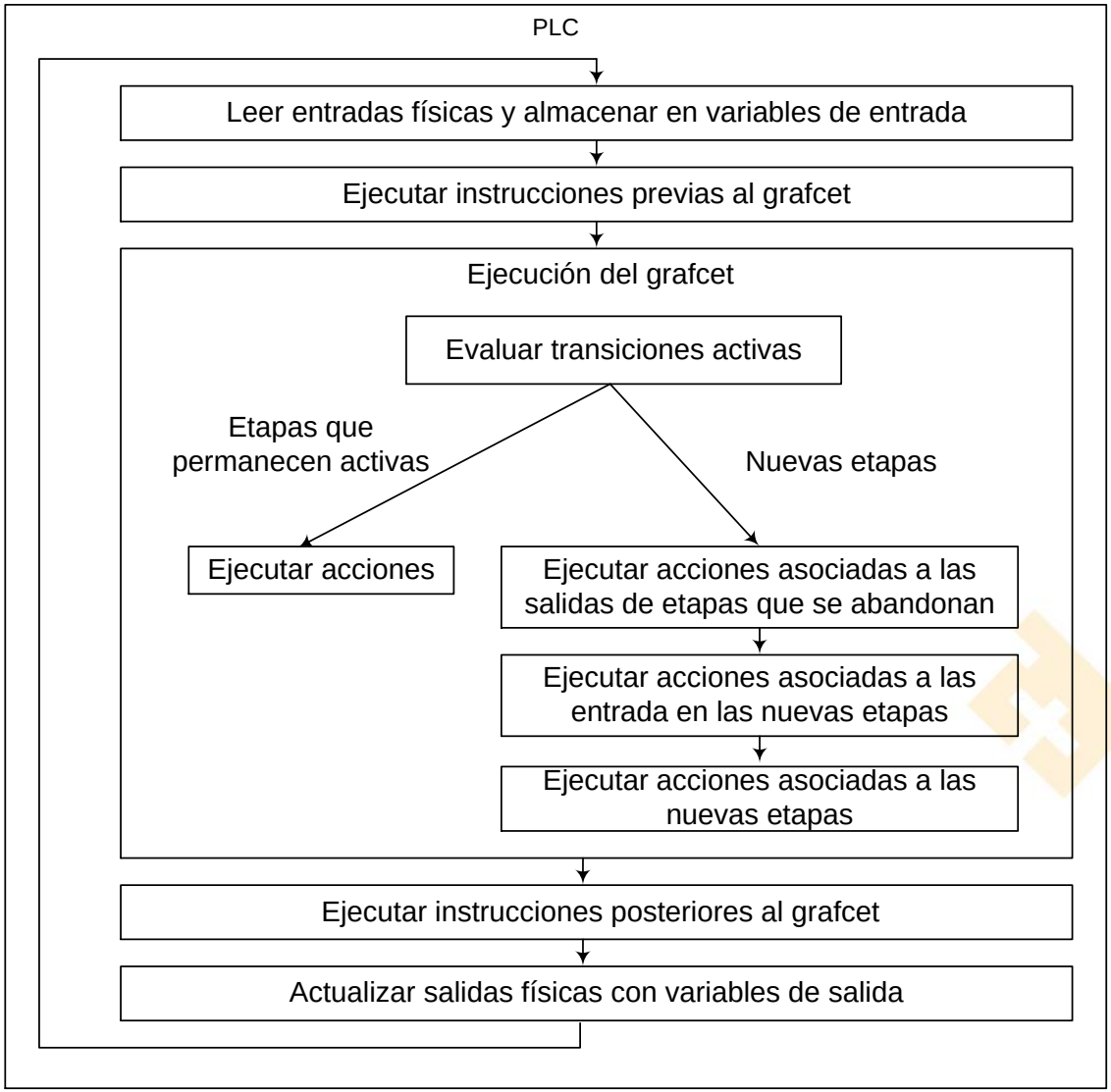


La etapa gs42 nunca se alcanza

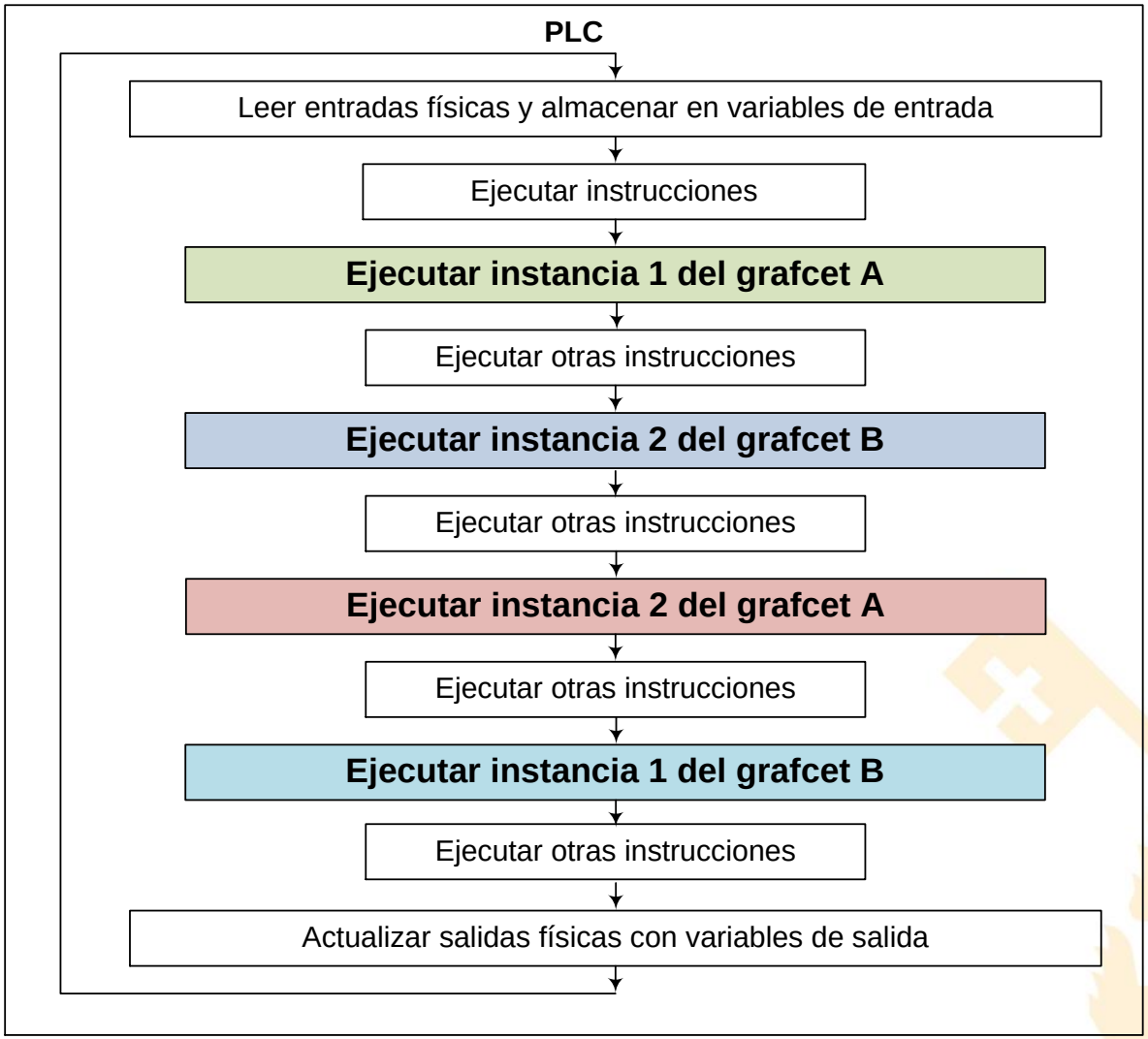


El salto hacia atrás es innecesario

Modelo de ejecución de un grafcet (en un PLC)



Ejecución de varios grafcets



Simplificación de operaciones para crear un grafcet

- **Crear etapa nueva en secuencia simple**
  - Secuencia simple
- **Crear secuencias alternativas desde una etapa**
  - Divergencia en O
- **Crear secuencias en paralelo desde una etapa**
  - Divergencia en Y
- **Volver a una secuencia simple desde secuencias alternativas**
  - Convergencia en O
- **Finalizar secuencias en paralelo**
  - Convergencia en Y

- **¡Cuidado!**
  - Cuando se finalizan las secuencias alternativas o en paralelo se pueden cometer errores
  - Revisar que la secuencia gráfica etapa-unión-transición-unión-etapa se cumple

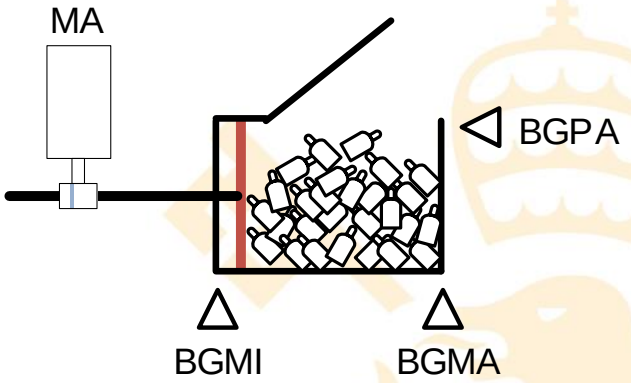
Problema 1: Compactadora de botellas de plástico

Diseñar el control mediante un PLC de una compactadora que tiene la siguiente interfaz hombre-máquina:

- Pulsador SJA para dar orden de avance del pistón de la compactadora.
- Pulsador SJR para dar orden de retroceso del pistón de la compactadora. Tanto avance o retroceso sólo se puede ordenar si la puerta de la compactadora está cerrada.
- Pulsador SJP para parar el avance o el retroceso. Si se abre la puerta tiene el mismo efecto.
- Piloto PFCP para indicar que el sistema de control está activo pero la compactadora está parada.
- Piloto PFMA para indicar pistón en posición de máxima presión de compactado.
- Piloto PFMI para indicar pistón en posición de mínima presión de compactado.
- Piloto PFPA para indicar puerta abierta.

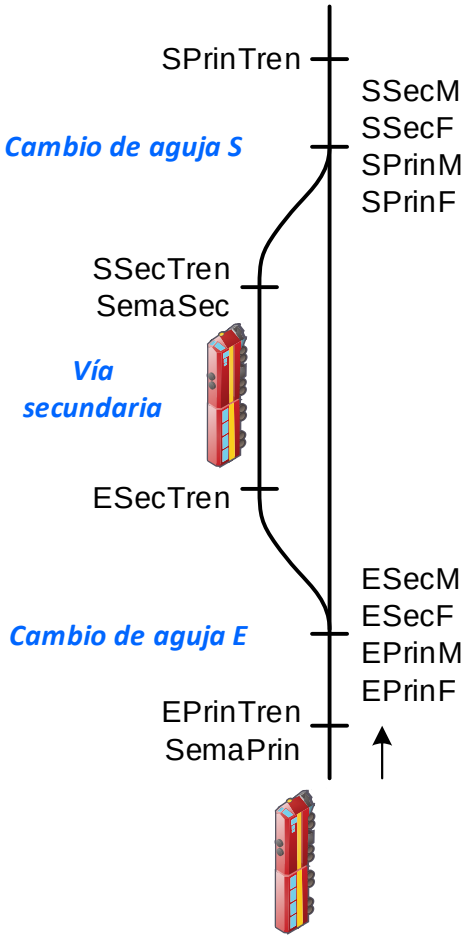
La compactadora tiene tres finales de carrera:

- BGPA: indica puerta abierta.
- BGMA: indica pistón en posición de máxima presión.
- BGMI: indica pistón en posición de mínima presión.



Diseñar circuito de control, circuito de potencia y grafcet a implementar en el PLC

Problema 2: Vía secundaria para adelantamiento

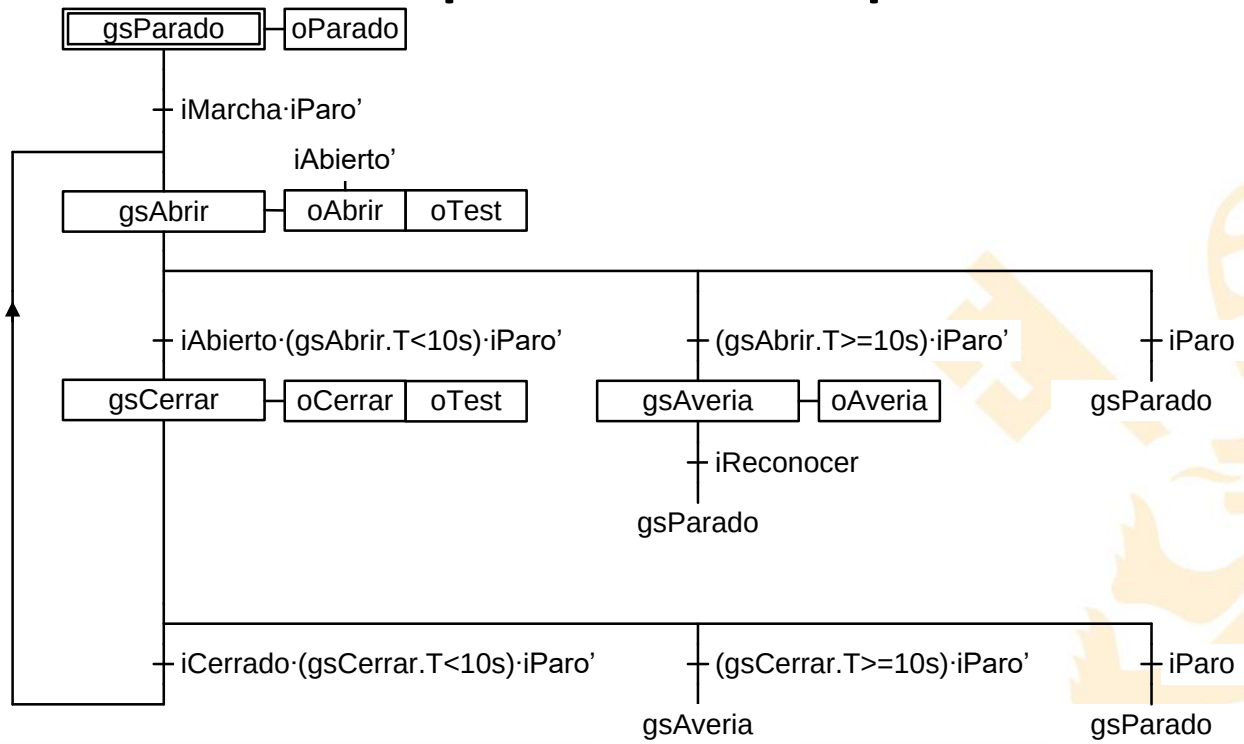


- **Automatizar el desvío a vía secundaria para adelantamiento entre trenes (ej: rápidos a lentos)**
- **Dos cambios de aguja: E y S (x=E,S)**
  - xSecM: orden motor para mover cambio a vía secundaria
  - xSecF: cambio en vía secundaria (final de carrera)
  - xPrinM: orden motor para mover cambio a vía principal
  - xPrinF: cambio en vía principal (final de carrera)
- **Cuatro sensores de barrera fotoeléctrica que se activan al pasar el tren: EPrinTren, ESecTren, SSecTren, SPrinTren**
- **Dos semáforos: SemaPrin y SemaSec (0-rojo, 1-verde)**
- **Dos pulsadores en el pupitre de control**
  - PrinSec: introducir siguiente tren en vía secundaria si ya no hay tren en vía secundaria
  - SecPrin: sacar tren a vía principal si está libre tramo entre E y S
- **Sólo se obedece a la maniobra pedida por el operador si no hay posibilidad de choque**
- **Distancias compatibles con longitud de trenes y distancia de frenado**

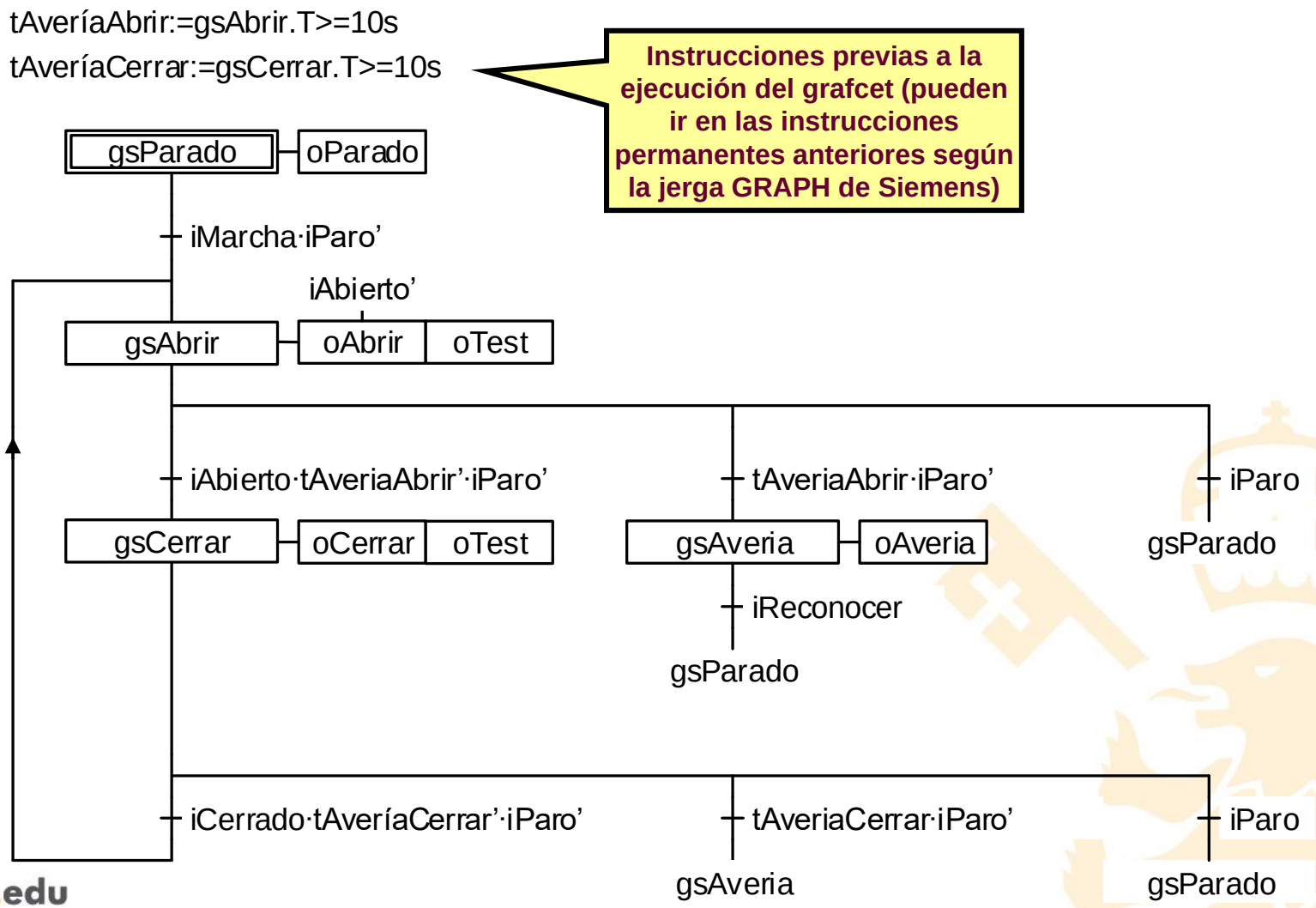
Diseñar el grafcet (o grafcets)

Temporizar: tiempo máximo en una etapa

- **Cada etapa tiene asociado un temporizador**
  - [Identificador etapa].T indica el tiempo en la etapa
  - Transición: [Identificador etapa].T > 1d23h30m20s700ms
- **Ejemplo: limitación de tiempo máximo de apertura o cierre**



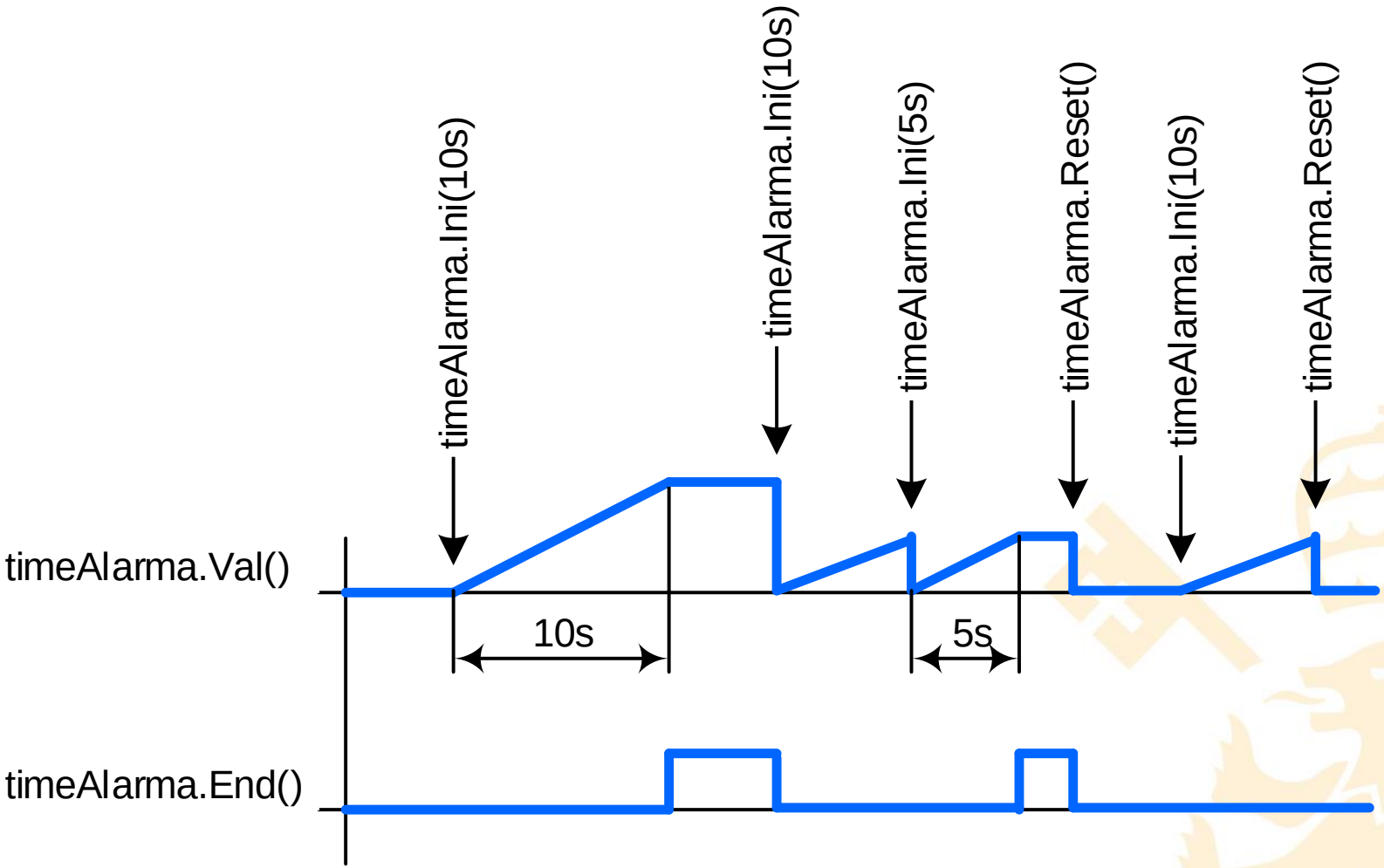
Test de guías de cajón con control de avería por tiempo máximo (versión con cálculos previos)



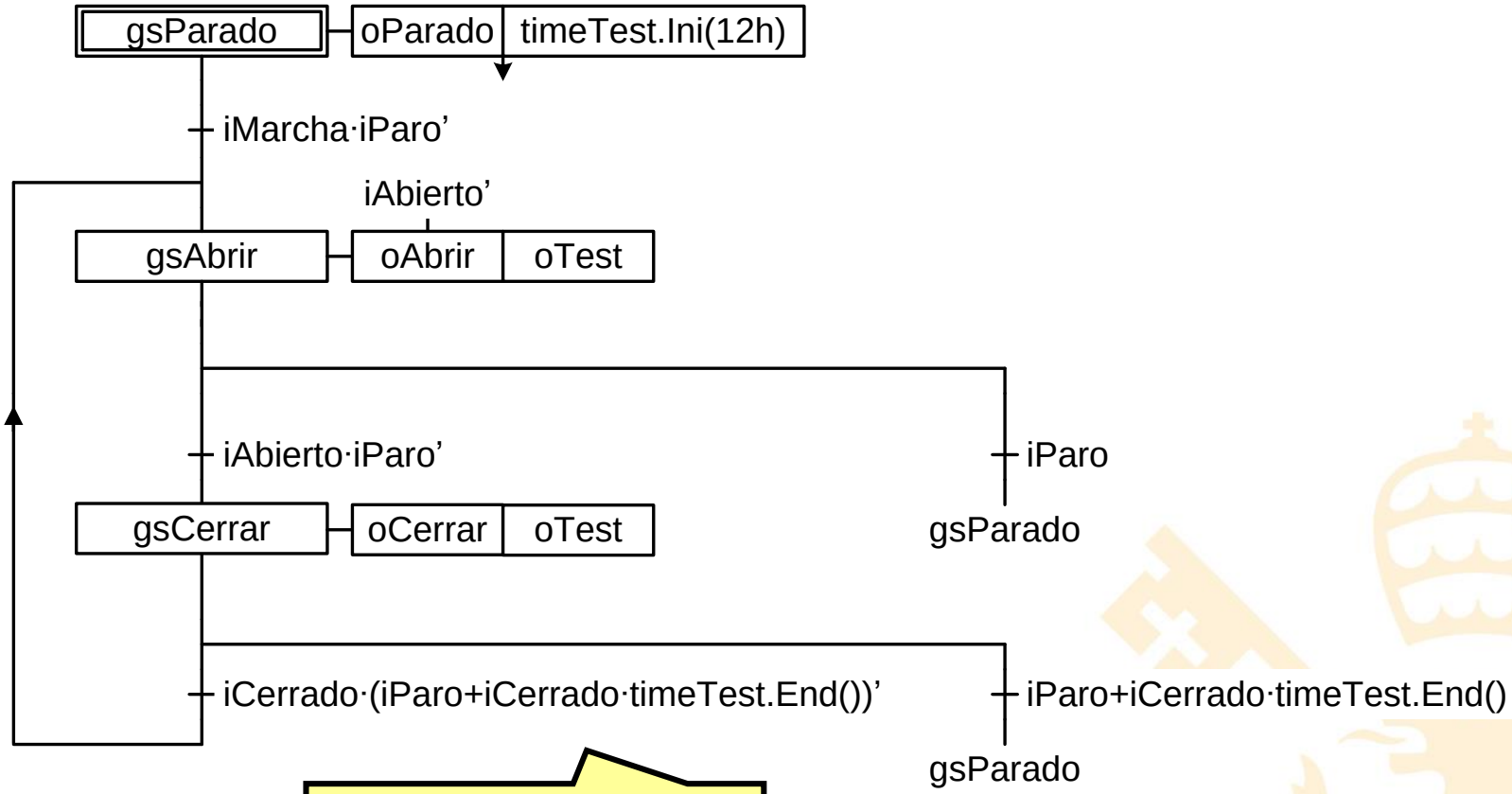
Temporizar: tiempo máximo en una secuencia

- **Existen variables de tipo temporización (timeXXXX) con cuatro funciones asociadas:**
  - *timeXXXX.Ini(tiempo)*
    - Arranca la temporización con el valor indicado en *tiempo*
      - sintaxis: 1d23h30m20s700ms
    - Cada vez que se llama a Ini se reinicializa la temporización.
  - *timeXXXX.End()*
    - Devuelve 1 si ha finalizado la temporización.
    - Devuelve 0 si no ha finalizado o está parada.
  - *timeXXXX.Reset()*
    - Para la temporización que haya en curso, si no estaba ya parada.
  - *timeXXXX.Val()*
- **Uso**
  - Se arranca el temporizador al salir o al entrar en una etapa.
    - Ejemplo: *timeMarcha.Ini(12h30m)*
  - Se consulta si ha finalizado en las transiciones que lo necesiten
- **Ejemplo: Limitar el test de guías de cajón a 12 horas**

Ejemplo de cronograma de temporizador



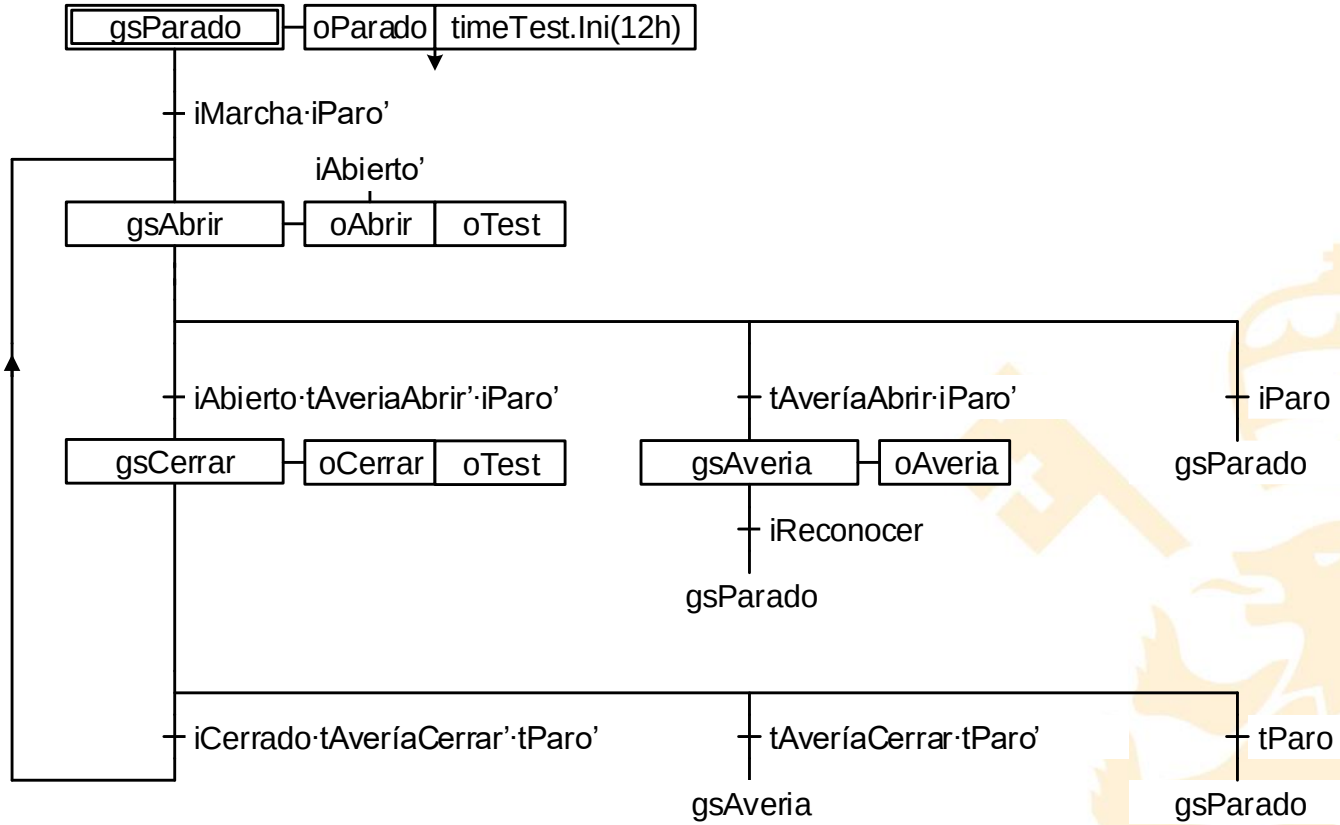
Test de guías de cajón que finaliza por tiempo con el cajón cerrado (versión sin avería)



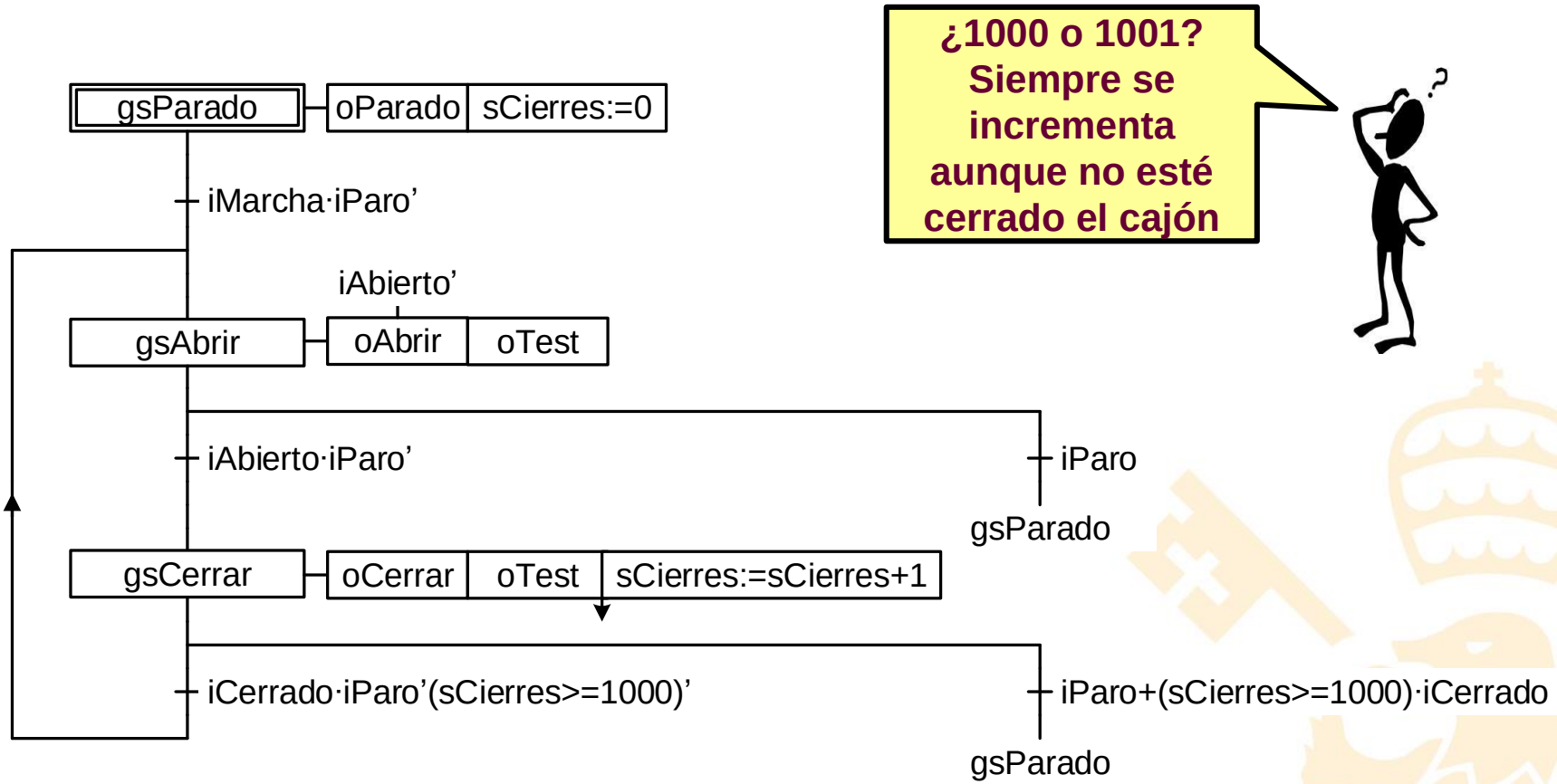
Se podría simplificar. Se ha mantenido para mostrar claramente que las condiciones son mutuamente excluyentes

Test de guías de cajón que finaliza por tiempo (versión con avería)

tAveríaAbrir:=gsAbrir.T>=10s
tAveríaCerrar:=gsCerrar.T>=10s
tParo:=iParo+iCerrado·timeTest.End()

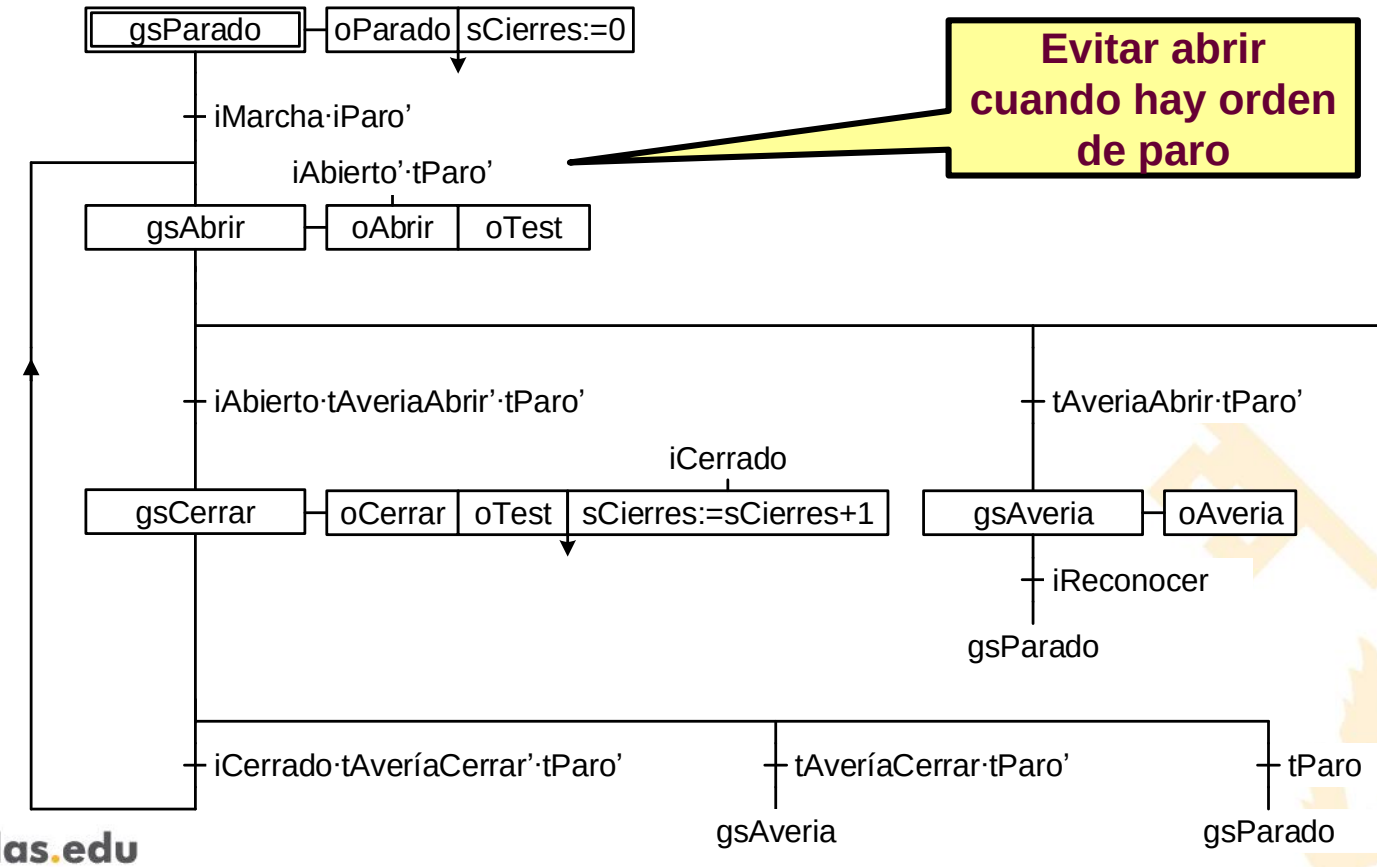


Acabar test por número de cajones cerrados



Acabar de forma exacta y con el cajón cerrado

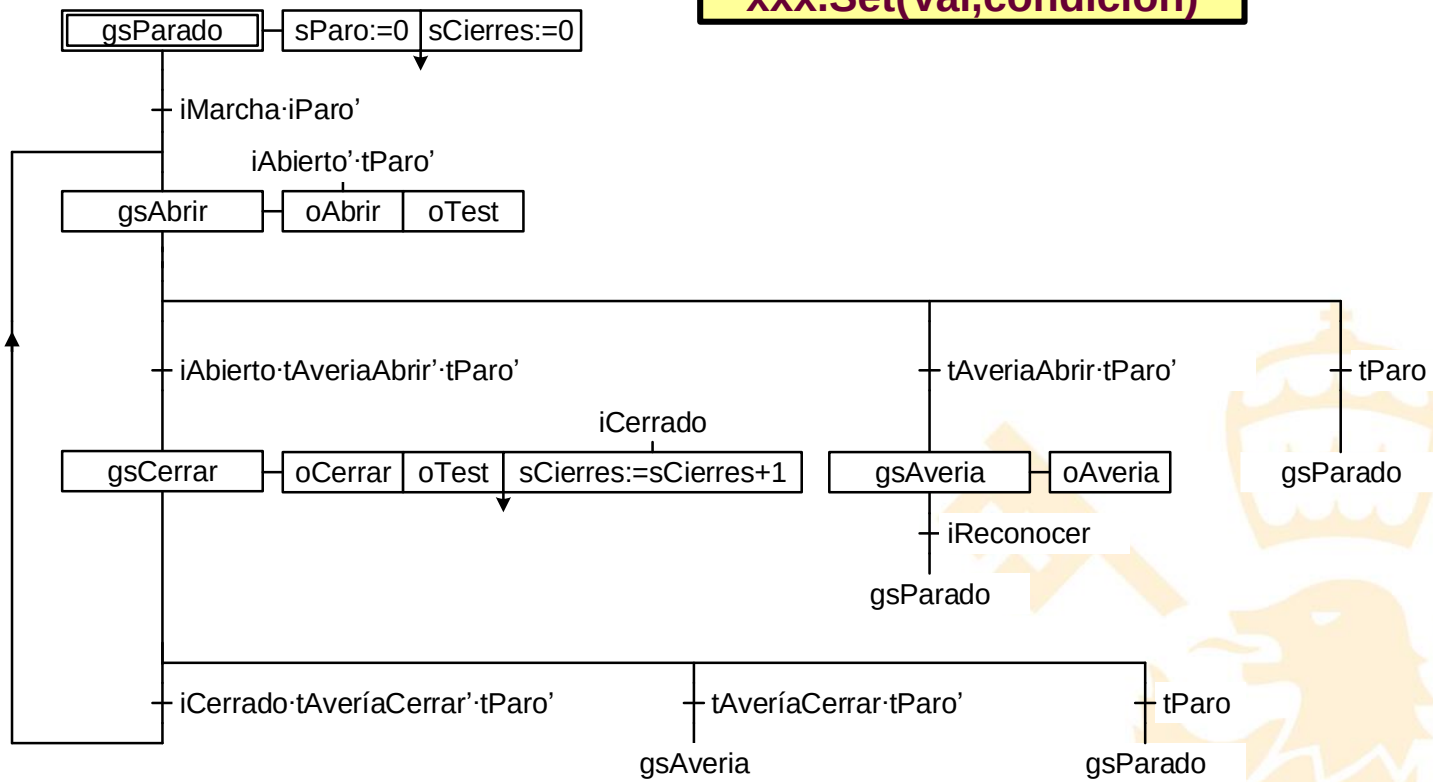
tAveríaAbrir:=gsAbrir.T>=10s
tAveríaCerrar:=gsCerrar.T>=10s
tParo:=iParo+(sCierres>=1000)



Ejemplo de parada a final de ciclo: cajón siempre cerrado

sParo.Set(1,iParo)
tAveríaAbrir:=gsAbrir.T>=10s
tAveríaCerrar:=gsCerrar.T>=10s
tParo:=iCerrado·sParo+(sCierres>=1000)

Asignar valor de si se cumple condición: xxx.Set(Val,condición)

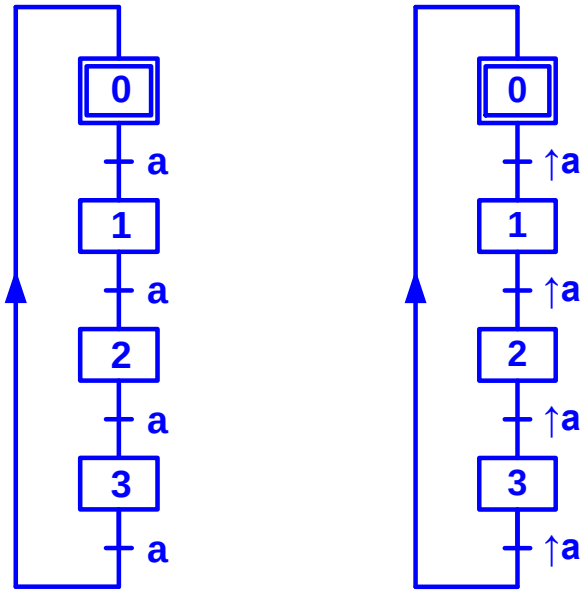


oParado:=gsParado.X+sParo·Clock_1Hz

Ecuaciones lógicas posteriores

Detección de flanco

- **Una orden de parada nunca lleva flanco**
- **Una orden de marcha puede exigir flanco**
  - Garantizar que el operador vuelve a dar la orden de marcha
- **Una secuencia de dos o más etapas con la misma condición en la transición, muy probablemente necesita flanco**



Identificador de una variable en un grafcet tecnológico

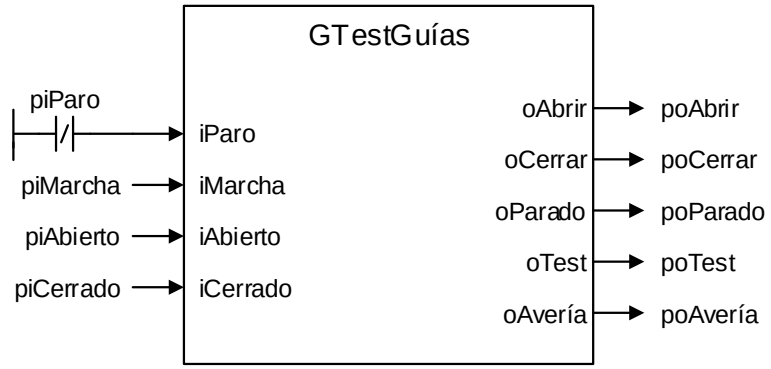
- **Prefijo + resto nombre siguiendo notación CamelCase**
- **Prefijos**
  - "i": entradas al grafcet (iParo, iMarcha)
  - "o": salidas del grafcet (oAbrir, oCerrar)
  - "t": variable para cálculo temporal (tParo)
    - Se calcula en la zona de cálculos previos
    - Uso: calcular expresiones lógicas que se repiten mucho, simplificar las condiciones lógicas de las transiciones y acciones
  - "time": variable temporizador (timeReceta)
    - Funciones asociadas: Ini(tiempo), End(), Reset(), Val()
  - "s" variable que mantiene su valor entre una ejecución y las siguientes del grafcet si no hay cambios (sParo)
    - Tiene asociada una función: Set(valor, condición lógica)
      - Variable igual a valor si condición lógica se cumple
      - sParo.Set(1, iParo)
    - Uso: memorizar una orden de paro
  - "gs": variable de tipo estructura para gestionar una etapa
    - gsParado.X indica etapa activa
    - gsParado.T indica tiempo que está la etapa activa

Declaración del tipo de las variables y operaciones

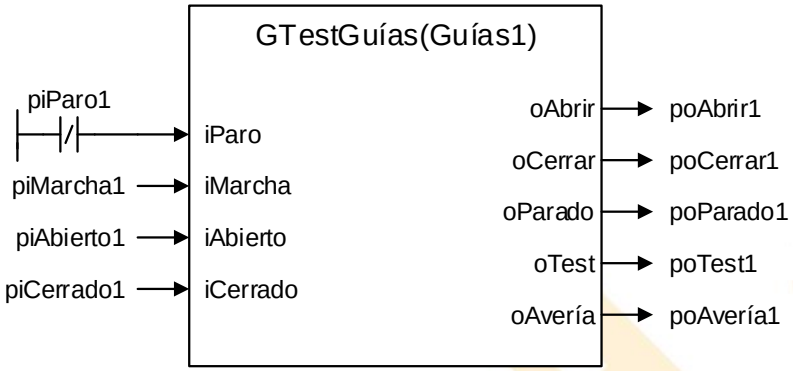
- **Dos tipos**
  - Variables lógicas
  - Variables numéricas
- **Declaración implícita**
  - No se declara el tipo
  - El uso de la variable (contexto) indica el tipo
    - s1:=s1+1 => s1 es de tipo numérico
    - s2:=a·b' => s2 es de tipo lógico
- **Al operar mediante operadores aritméticos una expresión numérica con una expresión lógica, la lógica promociona a numérica**
  - s1:=s1+↑s2 => s1 se incrementa en uno cada vez que hay flanco en s2
- **Al operar mediante operadores lógicos expresiones numéricas, el resultado es lógico conforme al operador**
  - s1:=s1+↑(tTemperatura<-20°C) => s1 se incrementa en uno cada vez que la temperatura baja de -20°C

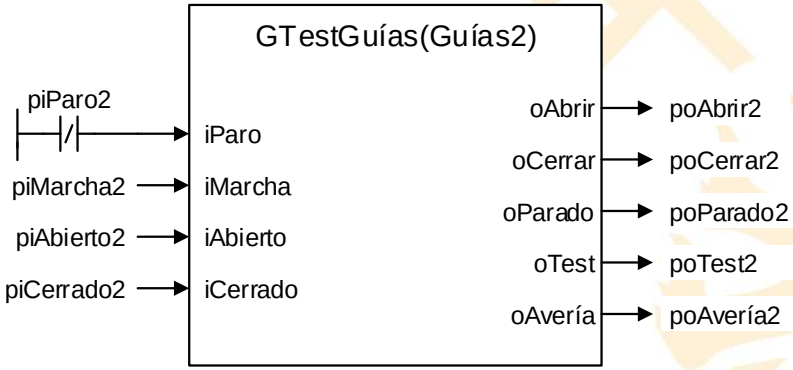
Grafcet como caja negra

- **Si hay una sola instancia del grafcet: nombre original**



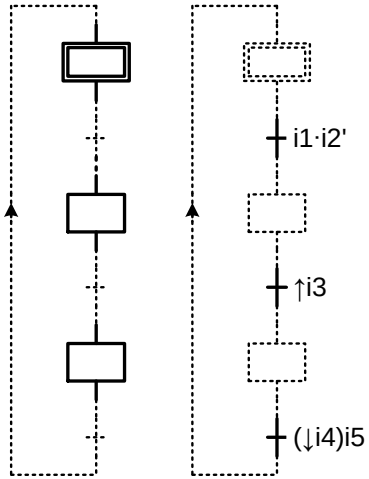
- **Si hay varias instancias del mismo grafcet: paréntesis con el nombre de cada instancia**

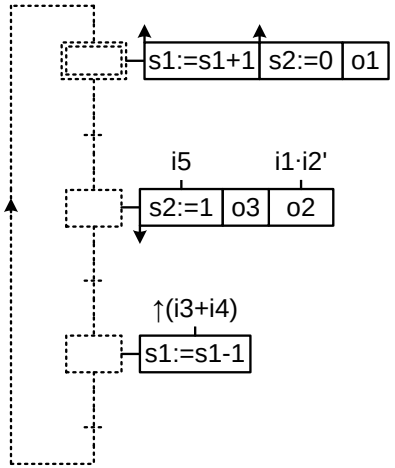


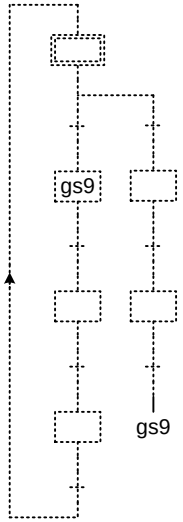


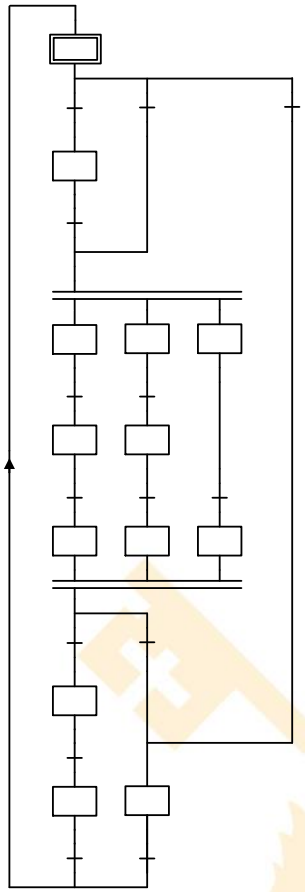
COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS
MASTER EN INGENIERÍA INDUSTRIAL

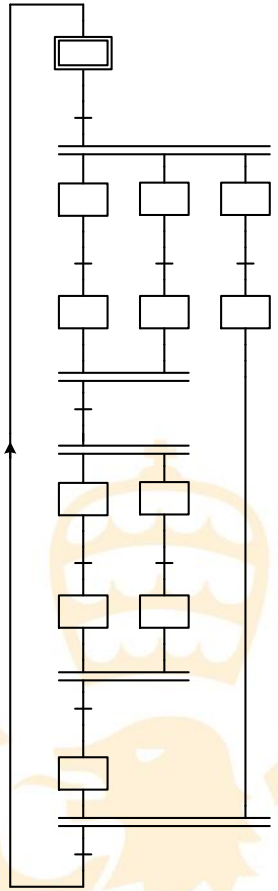
Resumen símbolos grafcet a través de ejemplos

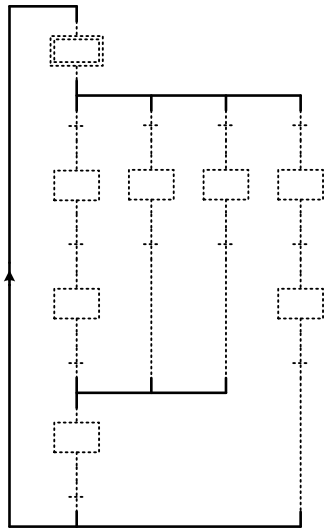


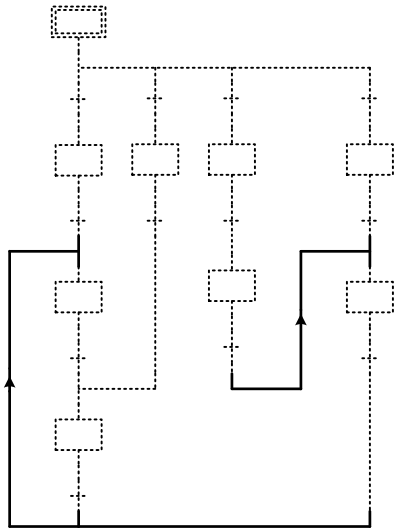


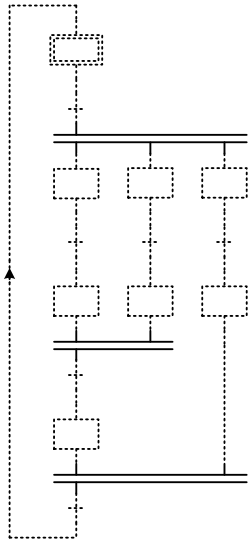








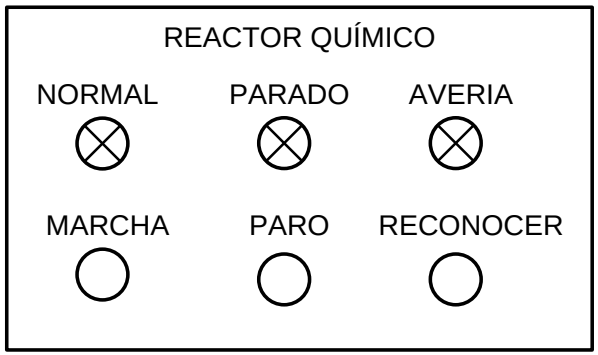


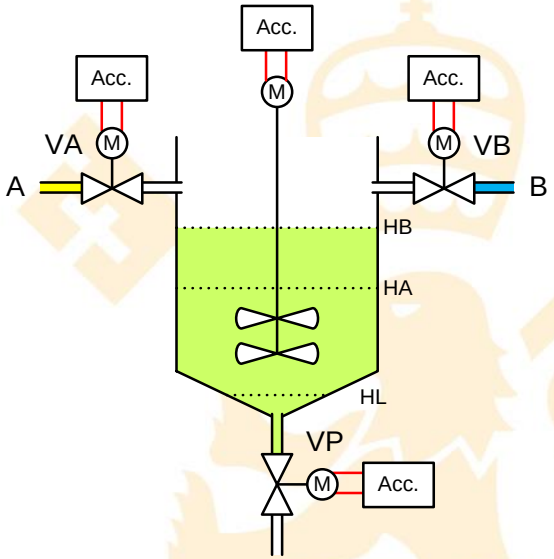


comillas.edu

Problema: Automatizar reactor químico

- **Receta a ejecutar cuando se pulsa Marcha y no hay avería:**
  - Llenar con producto A hasta HA.
  - Llenar con producto B hasta HB.
  - Agitar durante 30 minutos, alternando ciclos de 30 segundos a derecha y 30 segundo a izquierda
  - Vaciar hasta HL
  - Volver a repetir hasta completar 50 ciclos.
- **Si se pulsa Paro el sistema espera a completar el ciclo actual antes de parar**
- **Por seguridad, para poner en marcha la receta se debe asegurar que el operador ha vuelto a pulsar el pulsador de Marcha**
- **El sistema va a estado de avería si:**
  - Más de 30 segundos en alcanzar el nivel objetivo.
  - Más de 10 segundos sin abrirse o cerrarse la válvula
- **Señales de las válvulas (Vx)**
  - VxMC: cerrar la válvula
  - VxMA: abrir la válvula
  - VxC: válvula cerrada
  - VxA: válvula abierta
- **Señales del motor**
  - MD: girar a derechas
  - MI: girar a izquierda
- **Grafcet del sistema automatizado**





Problema: Automatizar puerta de nave de seguridad

La entrada, que es única, tiene dos puertas: Puerta1 y Puerta2, con una cámara intermedia. La Puerta1 permite el acceso desde la calle a la cámara intermedia; y la puerta 2 permite el acceso desde la cámara intermedia al interior de la nave. Cada puerta está dotada de 2 sensores para indicar si está subida o bajada (Subida1, Bajada1, Subida2, Bajada2) y de un motor que puede subir o bajar la puerta (M1S, M1B, M2S, M2B). Además cada puerta tiene asociado un sensor de posición (SensorP1, SensorP2) que se activa cuando un coche o camión está cruzando la puerta. Junto a la entrada hay situado un lector de tarjetas. Éste lector tiene dos salidas (TC y TI): una genera un 1 lógico durante 5 segundos cuando la tarjeta es correcta (TC) y la otra genera un 1 lógico cuando la salida es incorrecta (TI), también durante 5 segundos.

La secuencia de funcionamiento de la entrada es la siguiente:

- El conductor del coche llega a la entrada, introduce su tarjeta en el lector de tarjetas y la retira. Independientemente de si la tarjeta es correcta o incorrecta, comienza a abrirse la puerta 1.
- Una vez que el vehículo ha cruzado la puerta 1 (está en la cámara intermedia), comienza a bajar dicha puerta.
- Si la tarjeta es incorrecta, una vez bajada la puerta1, suena la alarma (ALARMA): los cacos están en la trampa. Para desactivar la alarma y abrir las puertas hay que introducir una tarjeta correcta en el lector de tarjetas. Si nuevamente se vuelve a introducir la tarjeta correcta se bajan las puertas.
- Si la tarjeta es correcta, comienza a abrirse la puerta 2. Cuando cruza el vehículo esta puerta, se cierra.
- Si el vehículo tarda más de 20 segundos en atravesar alguna de las puertas, también suena la alarma. Para desactivarla se procede de igual forma que en el caso de tarjeta incorrecta: se introduce la tarjeta correcta para desactivar la alarma y abrir las puertas, y una segunda introducción de la tarjeta correcta baja las puertas.
- Cuando se inicializa el sistema se cierran las puertas si no estaban cerradas.
- Diseñar el grafcet de la automatización.

Resumen

- **Definición de la metodología grafcet**
- **Pasos para construir un grafcet**
- **Definición detallada de los elementos de un grafcet**
  - Etapa
  - Transición
  - Unión: simple, selección, sincronización
  - Acción
    - Tipo continuo (no condicionada, condicionada)
    - Tipo evento (activación de etapa, desactivación de etapa, evento en la etapa)
- **Modelo de ejecución de un grafcet en un PLC**
- **Manejo de temporizadores**
  - Tiempo en una etapa
  - Tiempo en una secuencia de etapas
- **Manejo de contadores**
- **Manejo de flancos**

Preguntas

- **¿Para qué sirve un grafcet?**
- **¿Cuáles son los elementos fundamentales de un grafcet?**
- **¿Cuál es la diferencia entre un grafcet descriptivo y un grafcet tecnológico?**
- **¿Cuál es la diferencia entre el lenguaje grafcet y el lenguaje SFC?**
- **¿Cómo se diferencia una etapa normal de una etapa inicial?**
- **¿Cuál es la diferencia entre una unión de selección y una unión de sincronización?**
- **¿Cuántos tipos de acción hay?**
- **¿Es correcto incrementar un contador en una acción no condicionada? ¿Por qué?**
- **¿Es correcto que sólo aparezca el identificador de una variable en una acción asociada a la activación de una etapa? ¿Por qué?**
- **¿Cómo se define que una etapa de un grafcet se ejecute durante 30 segundos?**
- **¿Cómo se define que una secuencia se ejecute durante al menos una hora?**
- **¿Por qué no se debe encadenar acciones (cuando termina una comienza la siguiente) dentro de una etapa?**
- **¿Por qué determinados dibujos no cumplen con la sintaxis del lenguaje grafcet?**
- **¿Cómo es el modelo de ejecución del grafcet en un PLC?**
- **¿Por qué una orden de parada nunca lleva flanco?**
- **¿Por qué una secuencia de etapas que utilizan la misma condición lógica en la transición puede ser incorrecta, si no es de tipo flanco?**
- **¿Cuáles son los prefijos y su significado, que se recomiendan para definir el identificador de una variable?**