

Programación GRAFCET en CodeSys

SFC (Sequential Function Chart)

(SoMachine)

Juan M. Mendoza
juanmenbra@gmail.com



1. Introducción.

El término GRAFCET es el acrónimo tanto de Graph Fonctionnel de Commande Etape-Transition (en español, grafo funcional de control etapa-transición) y de graphe du groupe AFCET (gráfico del grupo AFCET).

Surge en Francia en 1977 como iniciativa de algunos fabricantes de autómatas (Telemecanique, Aper y otros) junto con los organismos oficiales AFCET (Asociación Francesa para la Cibernética, Economía y Técnica) y ADEPA (Agencia Nacional para el Desarrollo de la Producción Automatizada).

Fue homologado inicialmente en Francia (norma UTE NF C 03-190) en 1982 y con posterioridad por la Comisión Internacional de Electrotecnia (IEC 60848) en 1988.

La construcción de un sistema automático requiere, entre otras cosas, establecer las relaciones causa/efecto entre los eventos de entrada y las acciones deseadas (salidas). En este contexto, se denomina parte secuencial del sistema la que se circunscribe a las relaciones entre variables entrada y salida de tipo booleano.

El GRAFCET es una especificación de modelado y su realización es un diagrama, que denominaremos diagrama grafcet (o grafcet para simplificar) para separar el dibujo de las reglas de sintaxis. Posterior a la norma IEC 608048, la norma IEC 61131-3 (1ª edición en 1993) define 5 lenguajes de programación enfocados a los autómatas programables industriales.

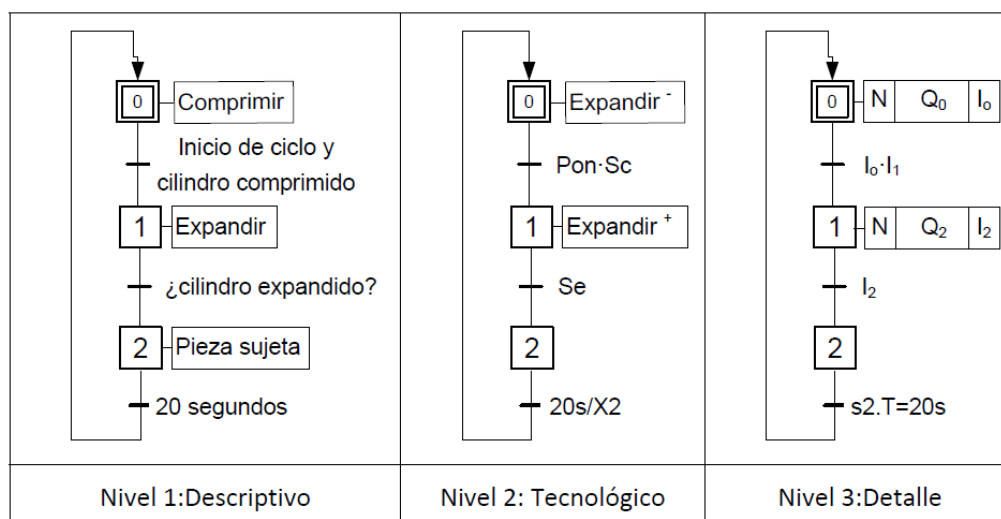
Uno de ellos está directamente inspirado en el lenguaje GRAFCET y lo denomina SFC (diagrama funcional secuencial, del inglés Sequential Function Chart). A diferencia del GRAFCET, la realización de una especificación SFC es un programa (gráfico) implementable en un autómata programable.

2. Niveles de diagrama GRAFCET.

Normalmente en la realización de un automatismo existen diferentes fases que van desde el análisis económico de viabilidad pasando por su diseño, su implementación (por ejemplo como programa ejecutable en un PLC), para finalizar con una fase de verificación offline mediante simulación y su puesta en marcha en el proceso real previa fase de pruebas.

Es claro que en las fases previas a la implementación los niveles de detalle del automatismo que se requieren conocer en la toma de decisiones son diferentes.

En función del nivel de abstracción que se represente se pueden distinguir los siguientes tipos de diagramas grafcet, ordenados de mayor a menor nivel de detalle:



- **Grafcet de nivel 1:** Descripción global poco detallada del automatismo que permite entender en poco tiempo su funcionamiento general. Es, por ejemplo, el nivel de detalle que las entidades financieras quieren conocer para decidir la inversión. Los grafkets de nivel 1 emplean descripciones en lenguaje natural para describir las acciones y transiciones y no contiene referencias a las tecnologías que se van a utilizar.
- **Grafcet de nivel 2:** Descripción de la tecnología. El grado de detalles en las descripciones debe ser lo suficientemente operativo para que todas las tecnologías empleadas en el automatismo (relés normales de enclavamiento, válvulas neumáticas normales o biestables, pulsadores, contactores etc.) queden representadas. Para este nivel y el anterior se suelen emplear la especificación GRAFCET de la norma IEC 60848.
- **SFC de nivel 3:** Descripción de la realización del automatismo: nivel de implementación. Para ello resulta adecuado la norma IEC 61131-3 que especifica un lenguaje de programación gráfico completo (diagrama SFC).

En la práctica no existe una regla fija para la elección de una u otra representación. Es más, es razonable mezclar elementos sintácticos de los diferentes niveles (por ejemplo acciones continuas condicionadas usando el calificador C que no aparece explícitamente en la norma IEC 61131-3) cuando ayudan a la legibilidad de la representación global.

3. Etapas, transiciones y arcos.

Un grafcet está formado por una sucesión de **etapas** y **transiciones** conectadas entre sí por **arcos** orientados.

Cada etapa puede tener asociada una o varias **acciones** a realizar sobre el proceso. Las etapas se representan como un cuadrado y un número (o a veces símbolo con un subíndice numérico) que indica su posición que las identifica de manera unívoca en el diagrama.

Se denomina **etapa inicial** o **etapa de reposo** al estado de control correspondiente al arranque, el estado inicial del grafcet.

Un mismo grafcet puede tener más de una etapa inicial, siendo el reposo de la ley de control entonces la activación simultánea de todas las etapas iniciales. Una etapa inicial se representa como una etapa normal con un recuadro exterior y se numera con 0 siempre que sea posible.

Una **transición** representa la condición por la que el sistema evoluciona de las etapas que la preceden a las etapas que la suceden.

Se dibuja como una barra horizontal que corta transversalmente al enlace entre las etapas denominado arco. Toda transición lleva una condición asociada denominada receptividad, resultado de la evaluación de una única fórmula lógica booleana (resultado 0 o 1).

Cuando se verifica la receptividad se dice que la transición es receptiva (o franqueable).

Los **arcos** vinculan etapas con transiciones, pero nunca etapas con etapas o transiciones con transiciones.

Son enlaces orientados que definen una relación de orden entre etapas y transiciones. En la figura 1-4A la etapa 0 precede a la transición "receptividad", al igual que la etapa 1 precede a "receptividad2" en la figura 1-4B.

Como regla general, el grafcet se lee de arriba abajo y los arcos tienen sentido descendente. Cuando el arco tiene sentido ascendente debe indicarse obligatoriamente con una flecha (como en el arco que conecta "receptividad2" con la etapa de reposo en la figura 1-4B).

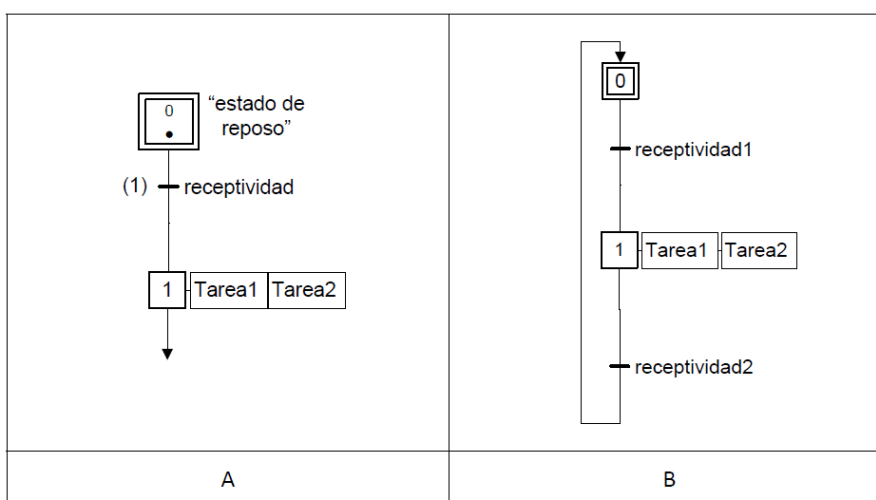


Figura 1-4. Símbolos del Grafcet.

Una transición se dice que está validada cuando están activas las etapas que la anteceden. Si una transición está validada y es receptiva entonces se franquea activándose todas las etapas posteriores y desactivándose todas las etapas anteriores.

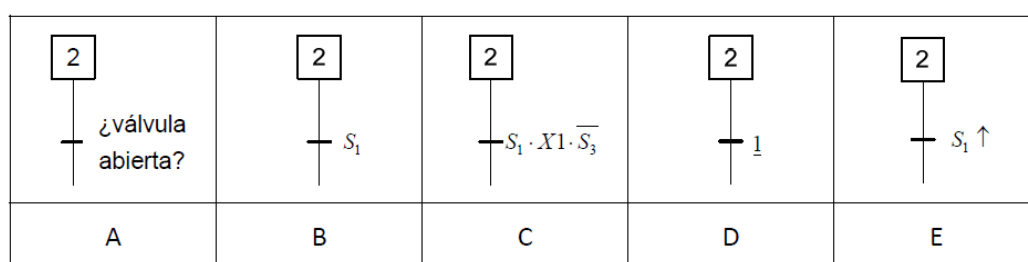


Figura 1-5. Ejemplos de sintaxis válidas para receptividades

La figura 1-5 muestra algunas de las diferentes notaciones válidas para receptividades en la especificación GRAFCET.

- 1-5A es una descripción en lenguaje natural de la función booleana.
- 1-5B indica que se tiene que verificar la variable S_1 .
- 1-5C es una función lógica con tres variables (donde la segunda es una variable de etapa).
- 1-5D la transición siempre es receptiva.
- 1-5E representa el evento flanco de subida de la señal S_1 .

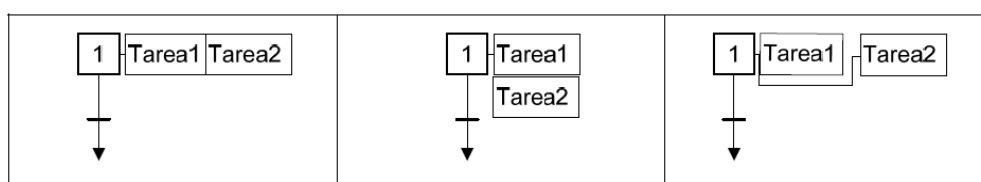
4. Acciones.

Una etapa puede llevar asociada desde cero a un número indeterminado de acciones. Una etapa sin acciones se denomina etapa de espera (en inglés wait step). Se emplea para representar una ausencia de evolución mientras que la transición que sucede a la etapa no sea receptiva.

Múltiples acciones asociadas se ejecutan de manera concurrente cuando la etapa está activa. Desde el punto de vista del comportamiento del sistema, la acción asociada a una etapa se denomina acción continua y su comportamiento dura lo que dura la actividad de la etapa.

Gráficamente, las acciones continuas se representan dentro de una caja rectangular asociada a la etapa. En el interior del rectángulo se indica bien en lenguaje natural, bien de manera simbólica, la semántica de la acción.

La figura a continuación muestra notaciones válidas para representar acciones continuas concurrentes:



También es posible (y muy frecuente) condicionar la realización de la acción continua a la verificación de una fórmula booleana. Este tipo de acciones se denominan condicionadas. La figura 1-6A muestra la sintaxis de una acción condicionada y en B una sintaxis descriptiva.

La acción se ejecuta cuando la etapa está activa y se verifica la condición asociada.

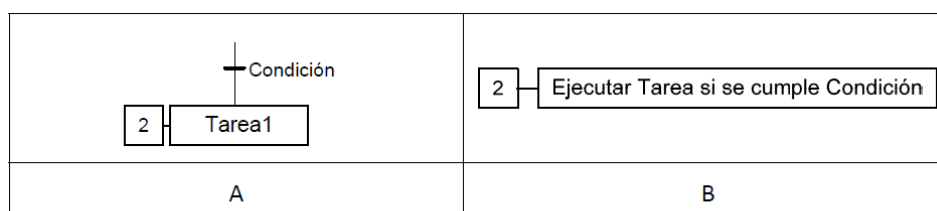


Figura 1-6. Representaciones de acciones condicionadas.

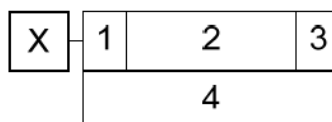
4.1 Símbolos normalizados de acciones en diagramas SFC.

Además de las acciones continuas de la sección anterior, la especificación GRAFCET ya recogía otro vínculo acción-etapa denominado acción memorizada.

Frente a las acciones continuas, las acciones memorizadas se vinculan a la propia evolución del grafcet (activación/desactivación de etapas, franqueamiento de transiciones etc.) y duran hasta que exista una orden de desactivación.

En lo relativo a relaciones acción-etapa, la norma IEC 61131-3 (diagramas SFC) formaliza ésta y otras ideas de una manera rigurosa y los elementos clave se describen en esta sección. Representaciones equivalentes en GRAFCET se expondrán cuando complementen, de alguna manera, a la especificación SFC.

Gráficamente la norma IEC 61131-3 representa las acciones asociadas a etapas como bloques con cuatro campos, de los cuales solo el segundo es obligatorio:



- **El campo 1** se denomina campo calificador (del inglés qualifier) y describe el tipo de vínculo entre la etapa y la acción asociada.

	Símbolo	Descripción
1	ninguno	acción continua mientras dura la etapa
2	N	acción continua mientras dura la etapa
3	R	desenclavamiento de la acción
4	S	enclavamiento de la acción
5	L	acción limitada tras la activación de la etapa
6	D	acción retardada tras la activación de la etapa
7	P	flanco de activación de la etapa
8	SD	acción memorizada y retardada
9	DS	acción retardada y memorizada
10	SL	acción memorizada y limitada en el tiempo
11	P1	flanco de activación de la etapa
12	P0	flanco de desactivación de la etapa

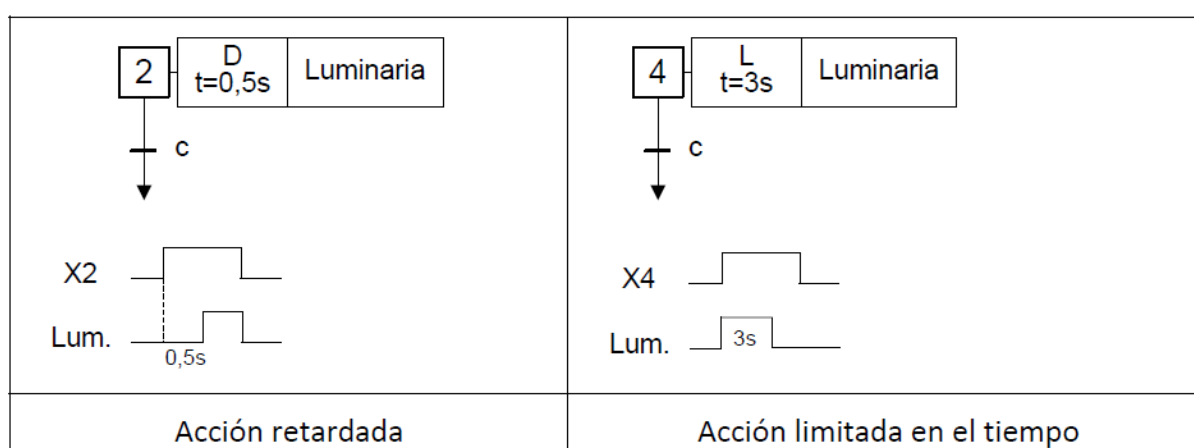
- **El campo 2** es el campo nombre donde se describe el comportamiento de la acción.
- **El campo 3** es booleano y se denomina campo indicador (indicator field en inglés). Permite, opcionalmente, especificar variables booleanas adicionales que pueden activarse para indicar eventos como que la acción ha terminado, condiciones de error, rebasamiento de un tiempo límite etc.
- **El campo 4** se usa para describir acciones complejas.

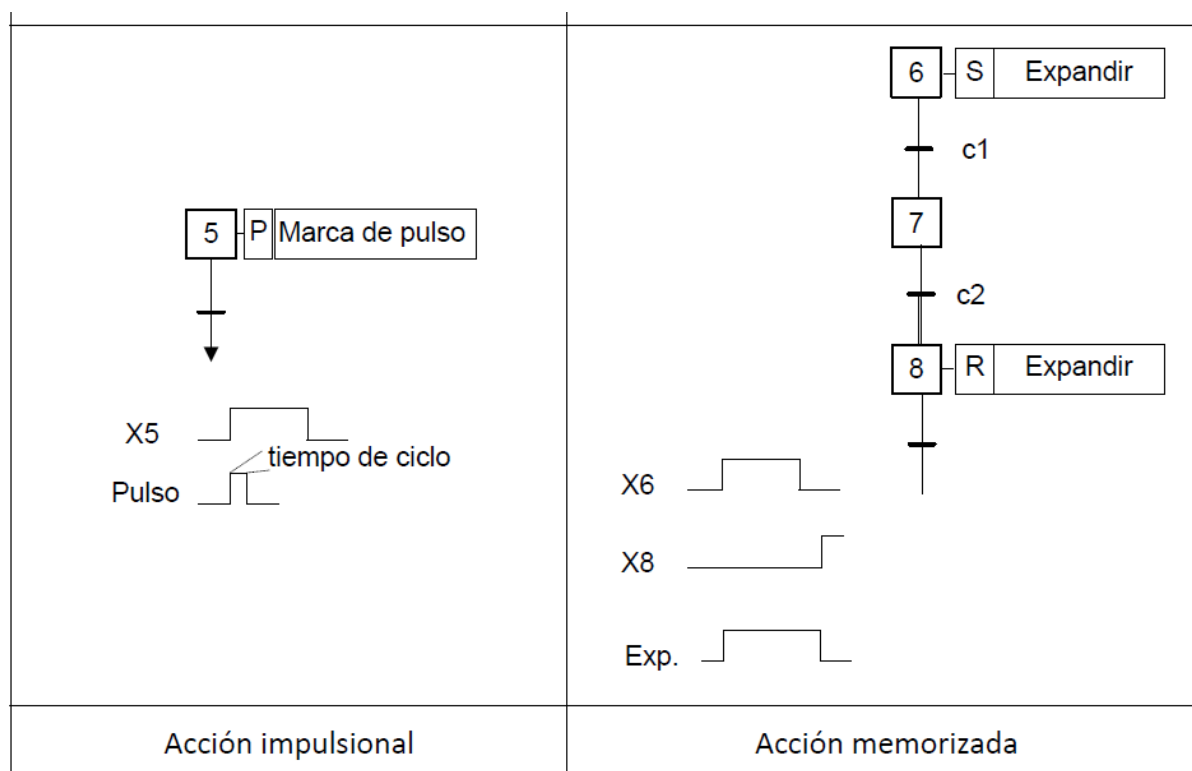
Existen 4 tipos principales de relaciones etapa-acción, además de la acción continua ya descrita, aceptadas por la norma:

- Acciones retardadas (**L**): La acción comienza un cierto tiempo después de la activación de la etapa.
- Acciones limitadas en tiempo (**D**): La acción termina tras un cierto tiempo de la activación de la etapa, aunque ésta siga activa.
- Acciones impulsionales (**P**): La acción dura el ciclo de operación de la activación de la etapa.
- Acciones memorizadas (**S**): La acción se enclava tras la activación de la etapa y perdura tras su desactivación. Será necesario una etapa posterior para desenclavarla.

4.2 Control de acciones.

En esta sección se describe el comportamiento previsto por la norma IEC 61131-3 para los calificadores más extendidos cuando la acción se puede modelar como una variable booleana (pertenece a la parte secuencial del automatismo).



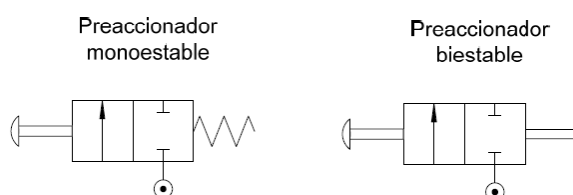


- La acción retardada (D)** empieza un cierto tiempo después de la activación de la etapa y dura hasta que la etapa deja de estar activa. En el ejemplo, la luz se encenderá medio segundo después de que se active al etapa 2 y se apagará tras verificarse la receptividad c. Si c se verifica con anterioridad al medio segundo la luz nunca se llegará a encender.
- Una acción limitada (calificador L)** se inicia con la activación de la etapa y finaliza bien tras su desactivación, bien tras un tiempo límite prefijado. En el ejemplo, la luz se enciende tras la activación de la etapa 4 y dura un tiempo de 3 segundos a pesar de que la etapa 4 sigue activa (no se ha cumplido c).
- La acción impulsional (calificador P Pulse)** es una acción de control endógena (no tiene visibilidad en el sistema controlado) que dura un tiempo muy corto, pero suficiente para conseguir el efecto deseado. En la práctica suele equivaler a la marca de pulso del flanco de la etapa a la que va asociada (en el ejemplo, la etapa 5). El calificador P1 especifica el flanco de subida y el calificador P0 el flanco de bajada.
- Las acciones memorizadas (letra S de Set)** se enclavan con la activación de la etapa y se desenclavan en una etapa posterior (acción con el calificador R de Reset). En el ejemplo de la figura una válvula de dos posiciones controla el movimiento de un cilindro neumático de simple efecto. En la etapa 6 la válvula se sitúa en posición de expansión y el cilindro se mantiene expandido hasta que en la etapa 8 la válvula recupera su posición original y el cilindro se comprime.

4.3 Efecto del tipo de preaccionador.

Existen condicionantes de carácter tecnológico que influyen en la representación del diagrama del automatismo. En particular, tiene especial importancia el hecho que los preactuadores sean de tipo común o de tipo biestable.

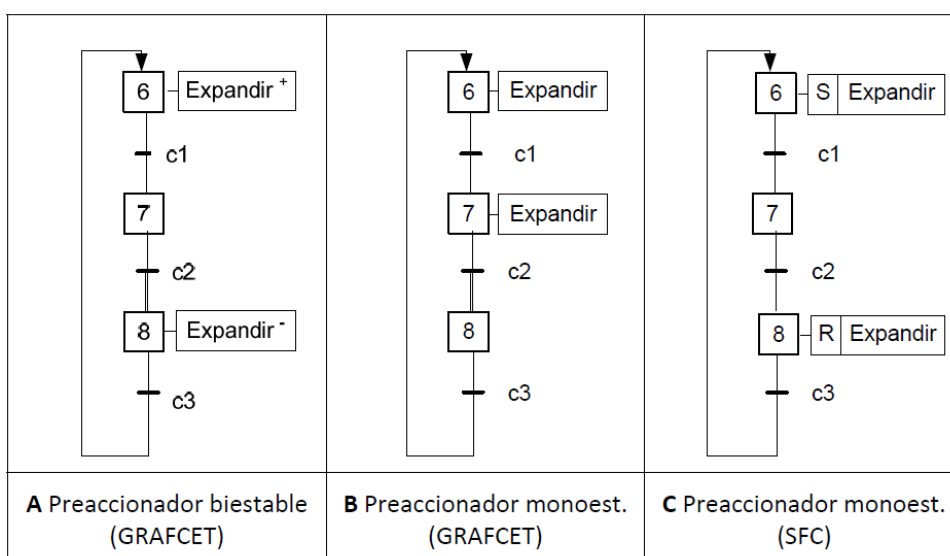
Un ejemplo de ambos tipos aparece en la figura. A la izquierda una válvula hidráulica de 2 vías y dos posiciones (2/2) con actuador manual monoestable y a la derecha la versión biestable.



En el primer caso, es necesario mantener la tensión en el actuador para conseguir la ejecución de la acción de control (en el ejemplo mantener pulsado el botón para que la válvula permita el paso del fluido) y se representan en el grafcet mediante una misma acción continua en etapas sucesivas.

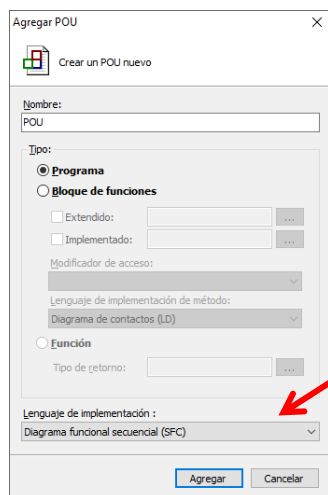
Los preaccionadores biestables, por contra, se enclavan en la posición nueva y se representan añadiendo a la descripción de la acción continua un + o un —, para indicar enclavamiento y desenclavamiento respectivamente.

Este tipo de consideraciones tecnológicas deben representarse en el grafcet de nivel 2. En la figura A, aparece la descripción de un ciclo expansión-compresión con preaccionador biestable, y en B el mismo sistema con válvulas monoestables. En C aparece una descripción equivalente de B, pulsando la sintaxis más operativa del lenguaje de programación SFC. En este caso, la acción expandir se ejecuta y memoriza al activarse la etapa 6 y se para con la activación de la 8, no siendo necesaria asociarla con la etapa intermedia 7.



5. Programación en GRAFCET

5.1 Programa Inicial

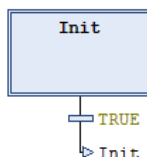


SFC - Diagrama funcional secuencial.

Seleccionar el lenguaje SFC cuando añada un nuevo objeto POU al proyecto.

El editor SFC está disponible en la parte inferior de la ventana que se abre cuando se edita un objeto POU.

El grafcet que aparece en el editor, por defecto, tiene la etapa principal (*Init*) que se reconoce por ser la primera y porque es un cuadro con doble línea, la transición y el retorno a la etapa inicial.

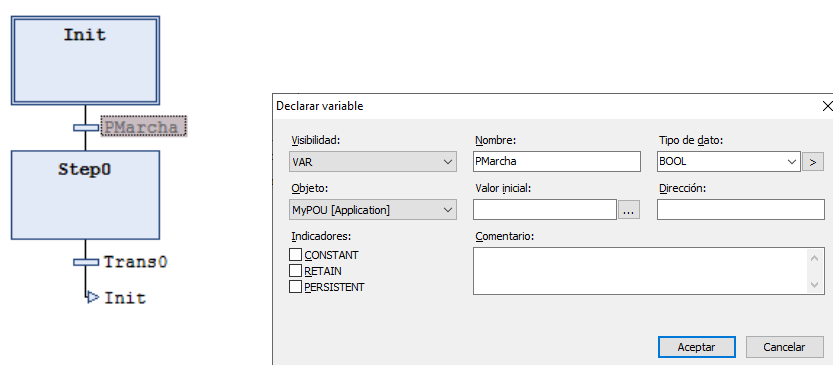


Para anexar más etapas se debe ubicar el cursor sobre la transición inicial e ir a la barra de herramientas y dar click sobre “transición de paso después”

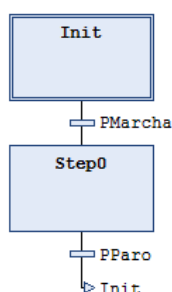


La nueva etapa se agregará después de la transición, si se quiere anexar la nueva etapa antes de la transición inicial se dará click sobre el icono al lado del utilizado.

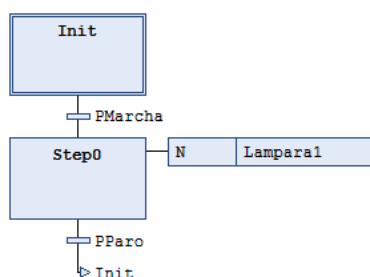
Para crear variables posicionar el cursor sobre la transición inicial, en este caso se escribe el nombre del sensor o el mando, Pulsador marcha: **PMarcha**, al terminar de asignar nombre aparecerá la ventana para la declaración de variables, en él se selecciona el tipo de variable en “Visibilidad”, seleccionar **VAR**.



Al igual que la transición inicial, se configura la transición 0 (Trans0) pero ahora la variable asociada será el pulsador de paro: **PParo**.



Para añadir acciones a las etapas, se activa la etapa (en este caso la etapa **Step0**) y se selecciona la herramienta *Insertar asociación de acción* el menú SFC.



Para crear variables de la acción posicionar el cursor sobre la acción, en este caso se escribe el nombre del actuador o componente, Lámpara: **Lámpara1**, al terminar de asignar nombre aparecerá la ventana para la declaración de variables, en él se selecciona el tipo de variable en "Visibilidad", seleccionar VAR.

En esta solución cuando se pulse el pulsador de marcha **PMarcha** se activará la etapa **Step0** y la lámpara **Lámpara1** se encenderá. Cuando se pulse el pulsador de paro **PParo** se activará la etapa **Init** y la lámpara se apagará.

5.2 Programación con acciones.

Una **acción** puede contener una serie de instrucciones escritas en uno de los lenguajes de programación válidos. Se asigna a un paso o etapa y, en modalidad online, se procesará de acuerdo con la secuencia de procesamiento definida.

Cada acción que se utilice con etapas SFC debe estar disponible como una POU válida en el SFC POU o el proyecto.

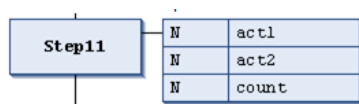


Acción de paso conforme a IEC (acción IEC)

Es una acción conforme a la norma IEC61131-3 que se procesará de acuerdo con su descriptor cuando el paso se active. En caso de asignar varias acciones a un paso o etapa, la lista de acciones se ejecutará de arriba abajo.

- A diferencia de las acciones de paso normales, para las acciones de paso IEC se pueden utilizar descriptores diferentes.
- Una diferencia adicional respecto a las acciones de paso normales es que cada acción de paso IEC está provista de un indicador de control. Esto permite que, aunque otro paso también llame a la acción, la acción se ejecute siempre sólo una vez al mismo tiempo. Esto no es así en el caso de las acciones de paso normales.
- Una acción de paso IEC se representa mediante un módulo doble conectado a la parte derecha de un paso mediante una línea de conexión. En la parte izquierda muestra el descriptor de la acción y en la parte derecha el nombre de la acción. Se pueden editar las dos en línea.
- Las acciones de paso IEC se asocian a un paso por medio del comando Insertar asociación de acción. Puede asociar una o varias acciones con un paso. La posición de la nueva acción depende de la posición actual del cursor y del comando.

Lista de acciones de paso conformes a IEC asociadas con un paso:



Cada módulo de acción de la primera columna muestra el descriptor y, en la segunda, se muestra el nombre de la acción.

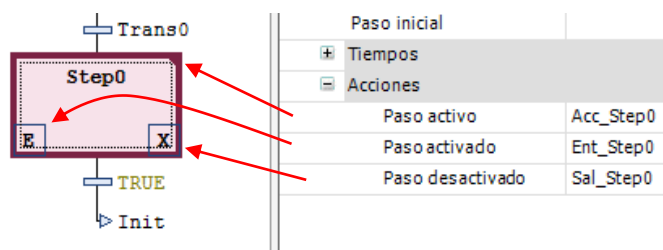
Acciones de paso extendidas IEC

Se trata de acciones que extiende la norma IEC. Deben estar disponibles como objetos debajo del objeto SFC. Se definen en las propiedades del paso o etapa.

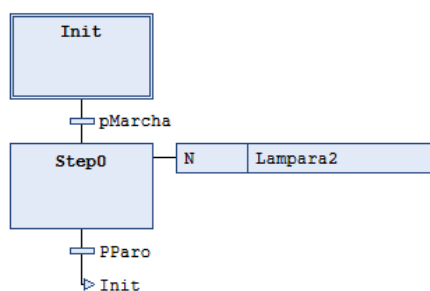
La tabla muestra las acciones de paso que extienden IEC:

Tipo de acción	Procesado	Asociación	Representación
acción de entrada del paso (paso activado)	Este tipo de acción de paso se procesará en cuanto el paso se active y antes de la acción activa del paso.	La acción se asocia a un paso por medio de una entrada en el campo Paso activado de las propiedades de paso.	Se representa mediante una E en la esquina inferior izquierda del módulo de paso respectivo.
acción activa de paso (acción de paso)	Este tipo de acción de paso se procesará cuando el paso se haya activado y después de que una posible acción de entrada de paso de este paso se haya procesado. No obstante, a diferencia de una acción de paso IEC, no se vuelve a ejecutar cuando se desactiva y no puede obtener descriptores asignados.	La acción se asocia a un paso por medio de una entrada en el campo Paso activo de las propiedades de paso.	Se representa mediante un pequeño triángulo en la esquina superior derecha del módulo de paso respectivo.
acción de salida del paso (paso desactivado)	Una acción de salida se ejecutará una vez cuando el paso se desactive. No obstante, esta ejecución no se realizará en el mismo ciclo sino al principio del ciclo posterior.	La acción se asocia a un paso por medio de una entrada en el campo Paso desactivado de las propiedades de paso.	Se representa mediante una X en la esquina inferior derecha del módulo de paso respectivo.

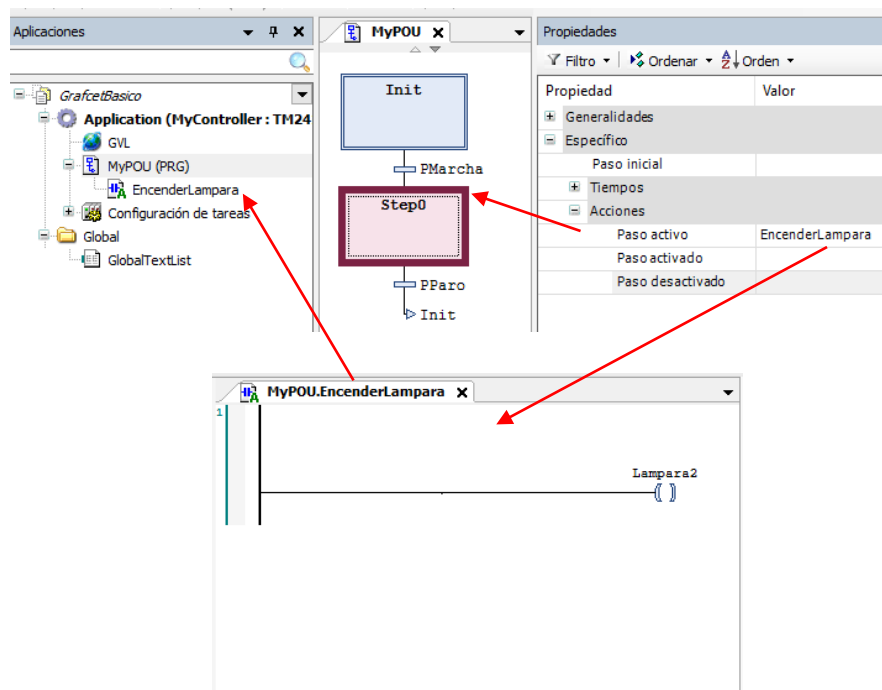
Acciones de paso extendidas de IEC:



El primer ejemplo es la programación con acciones conforme a IEC:



El segundo ejemplo es la programación con acciones extendidas IEC:



5.3 Bifurcaciones.

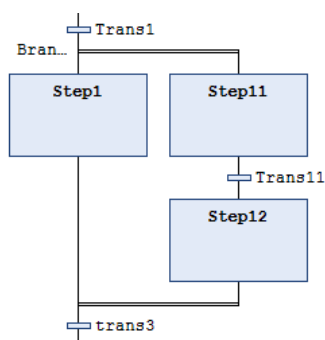
Un diagrama funcional secuencial (SFC) puede ramificarse; es decir, las líneas de procesamiento se pueden bifurcar en 2 o más líneas adicionales (bifurcaciones).

Las **bifurcaciones simultáneas** se procesarán en paralelo (simultáneamente). La bifurcación va precedida de una **línea horizontal doble**.

En el caso de las **bifurcaciones alternativas**, sólo se procesará una de acuerdo con la condición de transición anterior. La bifurcación va precedida de una **línea simple**.

Cada bifurcación de un diagrama también finaliza con la línea que la ha generado o con un salto.

Bifurcación simultánea



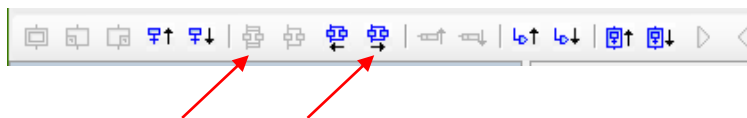
Una bifurcación simultánea debe empezar y finalizar con un paso. Las bifurcaciones simultáneas pueden contener bifurcaciones alternativas u otras bifurcaciones simultáneas.

Procesamiento en modalidad online: si la transición anterior (Trans1 en el ejemplo que se muestra en la parte izquierda) es verdadera (TRUE), los primeros pasos de todas las bifurcaciones simultáneas se activarán (Step1 y Step11). Las bifurcaciones específicas se procesarán simultáneamente antes de que la transición (trans3) se reconozca.

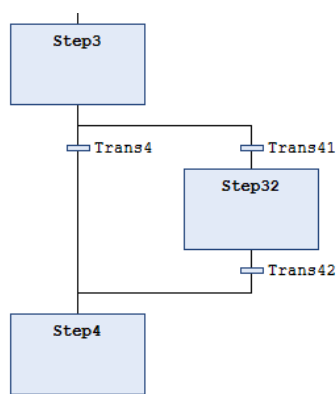
Para insertar una bifurcación simultánea, seleccione un paso y ejecute el comando **Insertar rama a la derecha**.

Puede transformar bifurcaciones simultáneas y alternativas entre sí ejecutando los comandos **Paralelo** o **Alternativa**.

Se añadirá automáticamente una etiqueta de bifurcación en la línea horizontal anterior a la bifurcación que lleva por nombre *Branch<n>* donde n es un número consecutivo que empieza por 0.



Bifurcación alternativa



Una bifurcación alternativa debe empezar y finalizar con una transición.

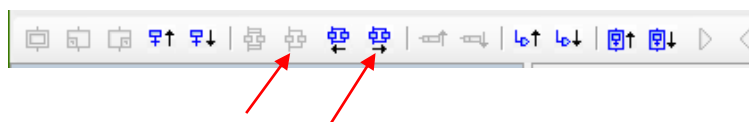
Las bifurcaciones alternativas pueden contener bifurcaciones simultáneas y otras bifurcaciones alternativas.

Si el paso que precede a la línea de inicio alternativa está activo, entonces la primera transición de cada bifurcación alternativa se evaluará de izquierda a derecha.

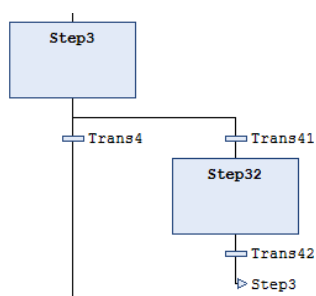
La primera transición de la izquierda cuya condición de transición tenga el valor TRUE se abrirá y se activarán los pasos siguientes.

Para insertar bifurcaciones alternativas, seleccione una transición y ejecute el **comando Insertar rama a la derecha**.

Puede transformar bifurcaciones simultáneas y alternativas entre sí ejecutando los comandos **Paralelo** o **Alternativa**.



Salto



Un salto se representa mediante una línea de conexión vertical más una flecha horizontal y el nombre del destino del salto.

Define el siguiente paso que se debe procesar en cuanto la transición anterior tenga el valor TRUE.

Se pueden utilizar saltos para evitar que las líneas de procesamiento se crucen o conduzcan hacia arriba.

Además del salto predeterminado al final del diagrama, un salto sólo se puede utilizar al final de una bifurcación.

Para insertar un salto, seleccione la última transición de la bifurcación y ejecute el comando **Insertar salto**. El destino del salto se especifica mediante la cadena de texto asociada, que se puede editar. Puede ser un nombre de paso o la etiqueta de una bifurcación simultánea.



5.4 Variables e indicadores SFC

Cada paso SFC y acción IEC proporciona variables generadas de forma implícita para la observación del estado de los pasos y las acciones IEC durante el tiempo de ejecución.

También puede definir variables para observar y controlar la ejecución de un SFC (tiempos de espera, restablecimiento, modo de información). Estas variables también las puede generar implícitamente el objeto SFC.

Básicamente, para cada paso y cada acción IEC, se genera una variable implícita.

Estado de paso o etapa

Para cada paso o etapa IEC se crea una variable de estructura implícita de tipo *SFCStepType*. Los componentes de la estructura (indicadores) describen el estado de un paso o etapa, o el tiempo transcurrido de un paso activo.

Están disponibles los siguientes indicadores booleanos para los estados de paso o etapa:

Indicador booleano	Descripción
<nombre de paso>.x	muestra el estado de activación actual
<nombre de paso>._x	muestra el estado de activación para el siguiente ciclo

Si <nombre de paso>.x = TRUE, el paso se ejecutará en el ciclo actual.

Si <nombre de paso>._x = TRUE y <nombre de paso>.x = FALSE, el paso se ejecutará en el siguiente ciclo. Esto significa que <nombre de paso>._x se copia a <nombre de paso>.x al principio de un ciclo.

Estado de acciones

Para cada acción IEC se crea una variable de estructura implícita de tipo *SFCStepType*. Los componentes de la estructura (indicadores) describen el estado de una acción.

Están disponibles los siguientes indicadores booleanos para los estados de acciones:

Indicador booleano	Descripción
_<nombre de acción>.x	es TRUE si se ejecuta la acción
_.<nombre de acción>._x	es TRUE si la acción está activa

Estado de tiempo de activación paso o etapa

El indicador **t** ofrece el intervalo de tiempo actual que ha transcurrido desde que se activó el paso.

Esto sólo se aplica a los pasos, no importa si existe o no un tiempo mínimo configurado en los atributos de paso.

Indicador tiempo	Descripción
<nombre de paso>.t	Indica el tiempo transcurrido desde la activación del paso.

Variables de sistema en ejecución SFC

Se puede utilizar algunas variables del sistema implícitas, también denominadas indicadores SFC para controlar el funcionamiento de un SFC. Por ejemplo, para indicar los desbordamientos de tiempo o activar el modo de información para cambiar las transiciones.

Para poder acceder a estos indicadores, se deben declarar y activar. Para ello, se utiliza la opción *Configuración SFC*. Es una opción de la herramienta Propiedades.

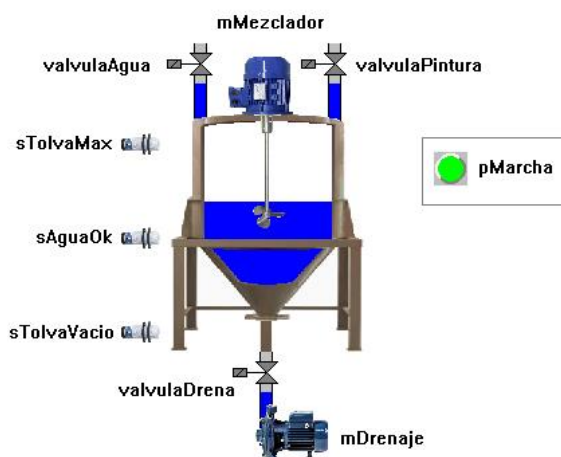
Se describen algunas de las variables implícitas (indicadores).

Variable	Tipo	Descripción
SFCInit	BOOL	Si esta variable se convierte en TRUE, el SCF se establece de nuevo en el paso inicial. Todos los pasos y acciones y otros indicadores SFC se restablecerán. El paso inicial se mantendrá activo, pero no se ejecutará mientras la variable sea TRUE.
SFCReset	BOOL	Esta variable se comporta de forma similar a SFCInit. A diferencia de esta última, sin embargo, se realiza un posterior procesamiento después de la inicialización del paso inicial. Por lo tanto, en este caso, se puede realizar un restablecimiento a FALSE del indicador SFCReset en el paso inicial.
SFCErrror	BOOL	Tan pronto como se produzca un <i>timeout</i> en uno de los pasos en el SFC, esta variable se convertirá en TRUE.
SFCPause	BOOL	Siempre que esta variable sea TRUE, la ejecución del diagrama SFC se detendrá.
SFCTrans	BOOL	Esta variable se convierte en TRUE tan pronto como se acciona una transición.
SFCCurrentStep	STRING	Esta variable almacena el nombre del paso activo, independientemente de la supervisión del tiempo. En caso de secuencias simultáneas, se registrará el nombre del paso exterior derecho.

6. Ejemplo de programación

6.1 Control de mezclador pintura.

Realizar el programa en GRAFCET de un sistema de mezcla de agua y pintura en una tolva.



El sistema de mezcla se ejecutará de acuerdo bajo el siguiente procedimiento:

- El programa de mezcla espera hasta que se presiona el botón de inicio **pMarcha**.
- Se agrega agua en la tolva accionando la electroválvula **valvulaAgua** hasta que el sensor **sAguaOk** detecta el nivel adecuado.
- Posteriormente se activa el motor mezclador **mMezclador** y la pintura se empieza a agregar a la tolva accionando la electroválvula **valvulaPintura**. Se añadirá pintura hasta que el nivel de la mezcla detecte el sensor superior **sTolvaMax**.
- El tiempo de mezclado debe ser de 30 segundos desde que se detecta toda la mezcla en la tolva.
- El drenaje de la mezcla se realiza activando la válvula **valvulaDrena** y la bomba de drenaje **mDrenaje**.
- Este proceso de drenaje se completa cuando el sensor inferior de la tolva **sTolvaVacio** deja de dar señal.
- Se volverá a la situación inicial.

6.2 Solución con acciones conforme a IEC.

Las variables utilizadas en el programa se detallan a continuación:

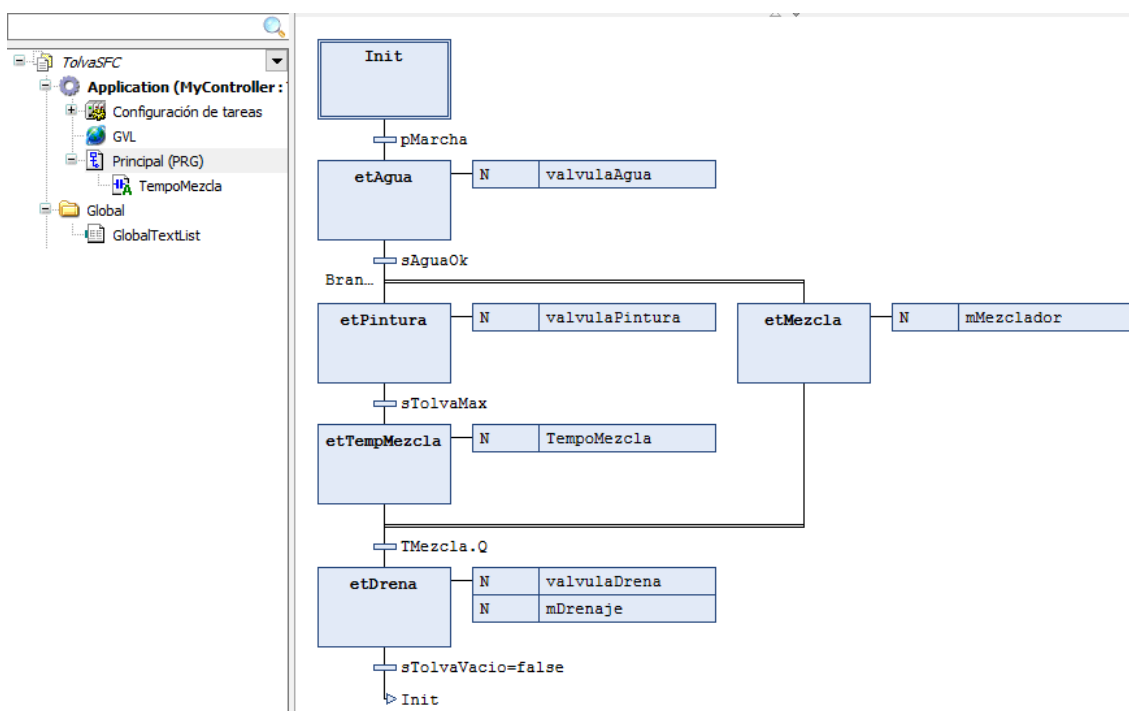
```

1  PROGRAM Principal
2  VAR
3      pMarcha: BOOL;
4      valvulaAgua: BOOL;
5      sAguaOk: BOOL;
6      valvulaPintura: BOOL;
7      mMezclador: BOOL;
8      sTolvaMax: BOOL;
9      TMezcla: TON;
10     valvulaDrena: BOOL;
11     mDrenaje: BOOL;
12     sTolvaVacio: BOOL;
13 END_VAR

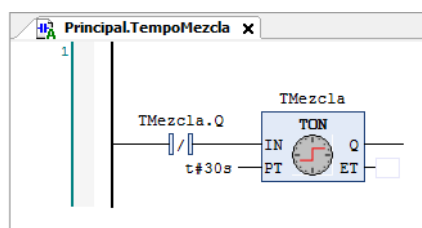
```

El programa SCF es el siguiente:

Los pasos activan las variables directamente en la etapa que les corresponde. El paso *etTempMezcla* lo que activa es una acción (*TempoMezcla*) en la que hay programada la temporalización del motor mezclador.



A continuación la programación de la acción *TempoMezcla*.



6.3 Solución con acciones extendidas IEC.

Las variables utilizadas en el programa se detallan a continuación:

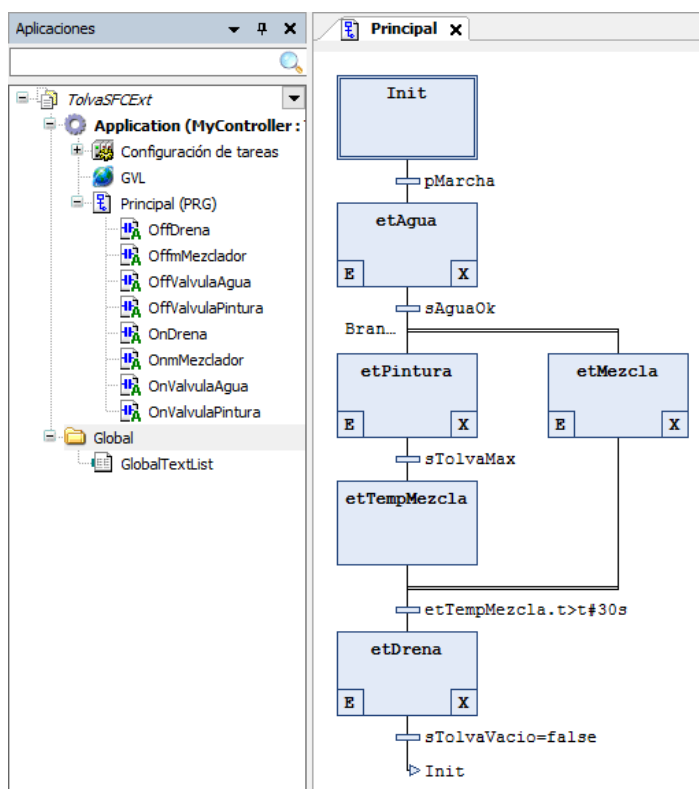
```

1  PROGRAM Principal
2  VAR
3      pMarcha: BOOL;
4      valvulaAgua: BOOL;
5      sAguaOk: BOOL;
6      valvulaPintura: BOOL;
7      mMezclador: BOOL;
8      sTolvaMax: BOOL;
9      TMezcla: TON;
10     valvulaDrena: BOOL;
11     mDrenaje: BOOL;
12     sTolvaVacio: BOOL;
13 END_VAR

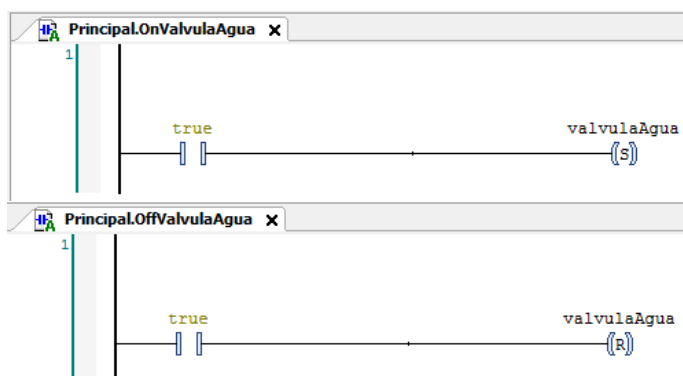
```

El programa SCF es el siguiente:

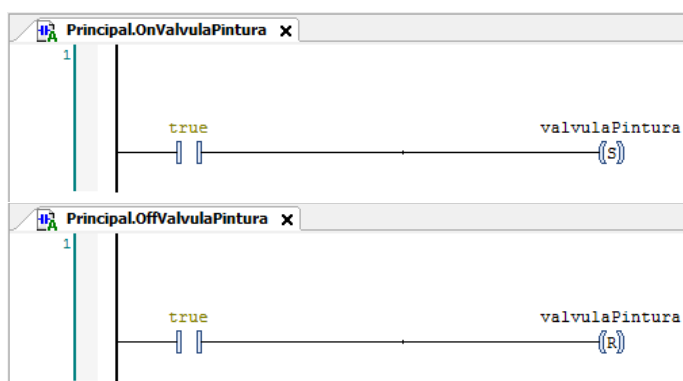
Los pasos tienen asociados acciones de entrada (**E**) y acciones de salida (**X**). La temporalización del paso **etTempMezcla** se realiza mediante una variable de estado (**etTempMezcla.t**)



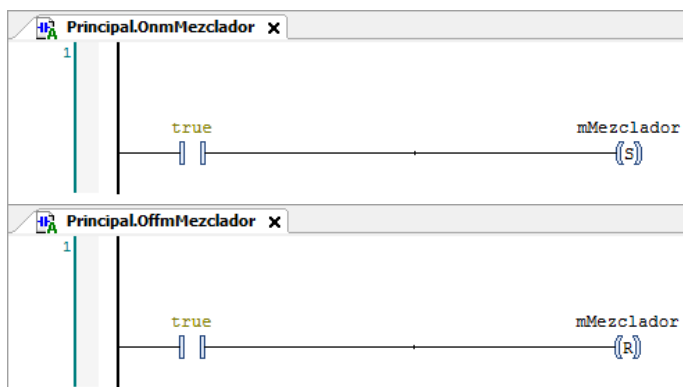
Las acciones asociadas a la etapa **etAgua**:



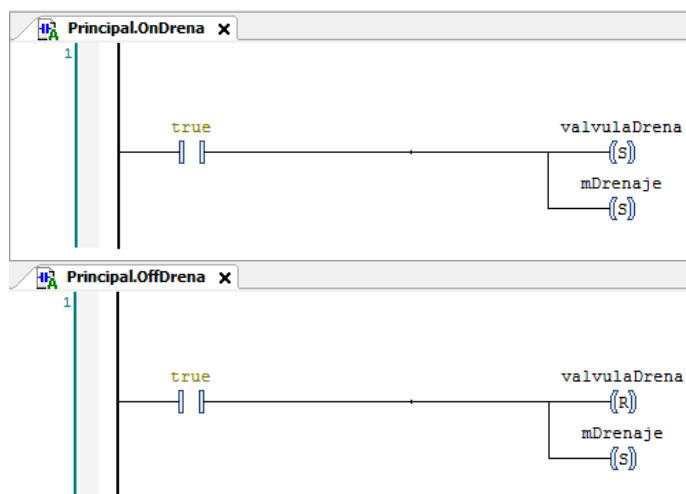
Las acciones asociadas a la etapa **etPintura**:



Las acciones asociadas a la etapa **etMezcla**:



Las acciones asociadas a la etapa **etDrena**:



7. Bibliografía y referencias

- INTRODUCCIÓN AL MODELADO GRAFCET - Universidad Politécnica de Madrid -UPM
- INICIACION CODESYS – GRAFCET – INFOPLC-Ana Silvia Cruz Fuentes
- ARCHIVO DE AYUDA – Programa SoMachine.