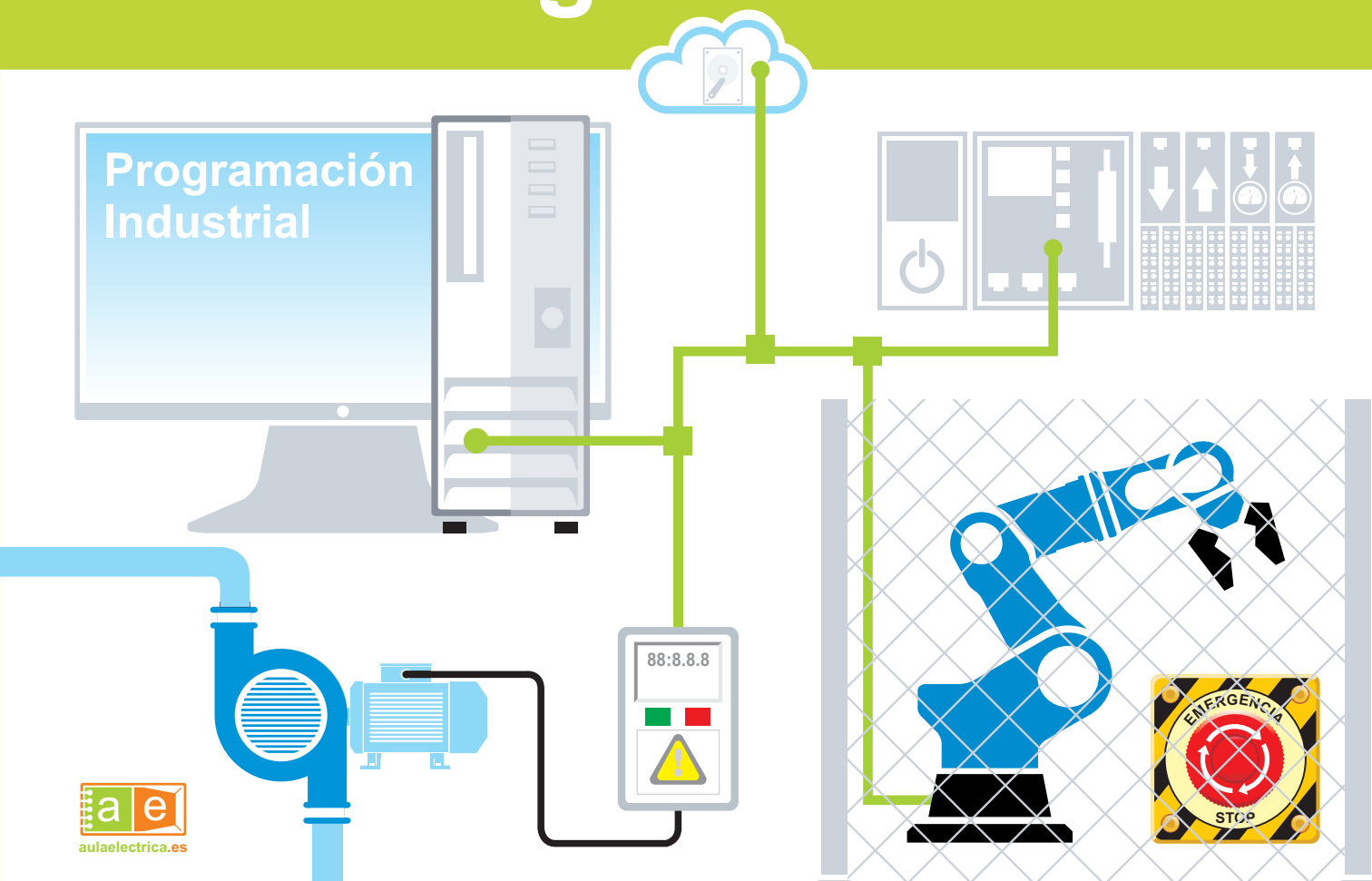


Sistemas Secuenciales Programables



Ciclo Formativo de Grado Superior

Automatización y Robótica Industrial

Formación Profesional

ISBN 978-84-940516-7-8

Rafael Arjona



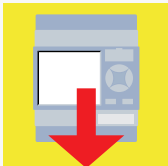
MULTIMEDIA



Audio explicativo



Vídeo explicativo



Descarga del archivo para controlador programable



Descarga de documento en pdf

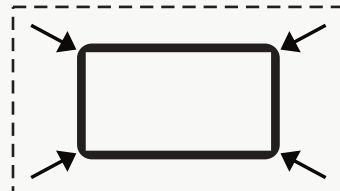
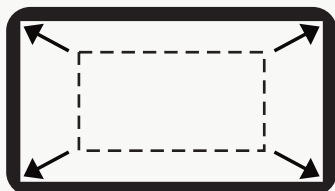


Descarga de imagen



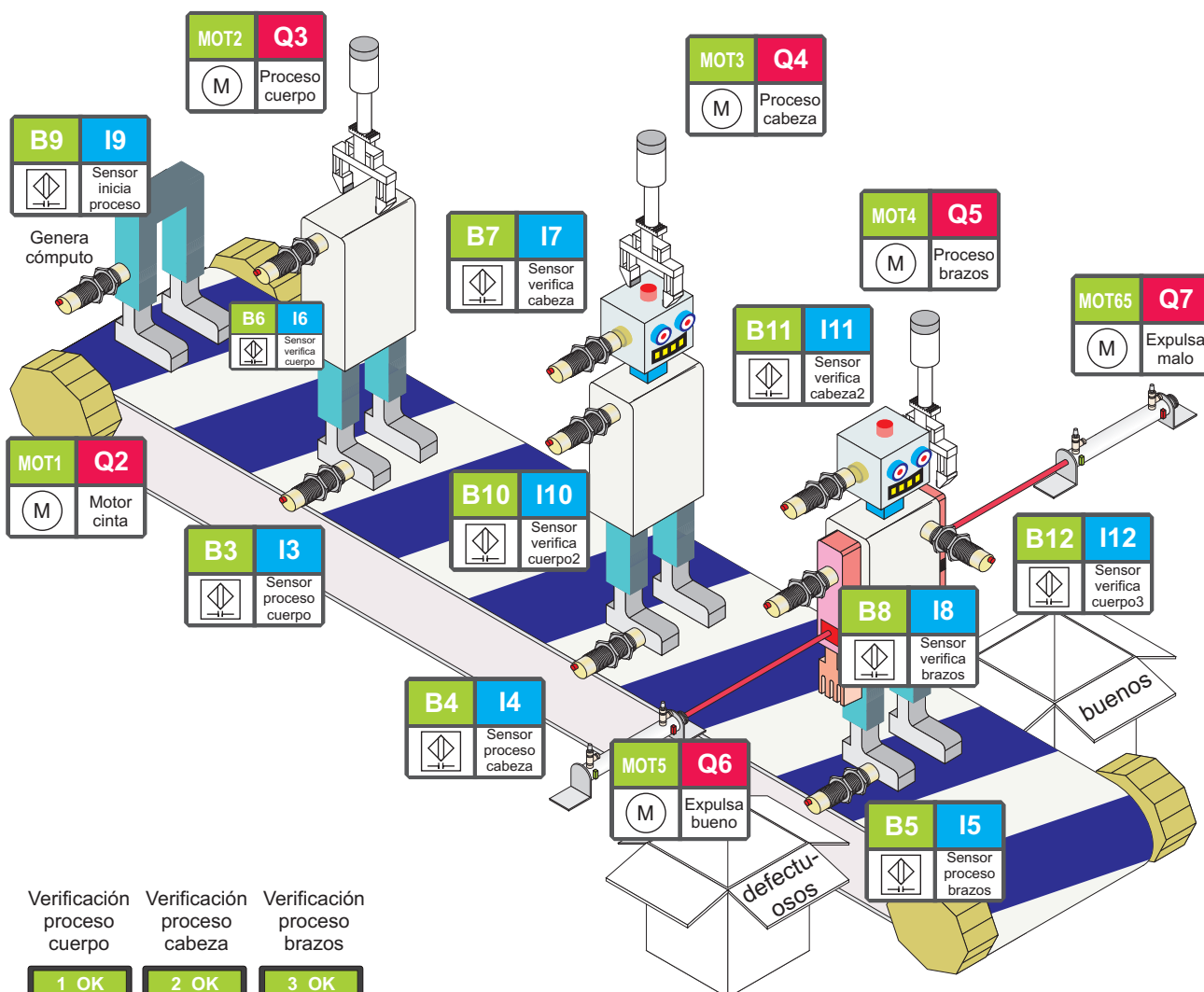
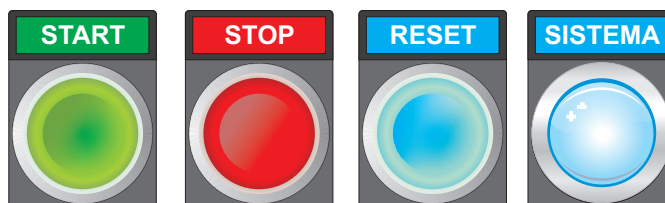
Descarga en general de archivos

Modo pantalla completa



5

GRAFSET



1 Gráfico secuencial de funciones (GRAF CET)



El GRAFCET es un lenguaje gráfico que representa las secuencias del programa en forma de diagrama. Para conocer este lenguaje, es necesario comprender los circuitos secuenciales.

2 Etapas



Una etapa implica una acción, dentro del proceso secuencial. Abrir una válvula, activar un motor, cerrar una compuerta, etc. En GRAFCET, se sucederán diferentes etapas, y la suma de todas ellas será el proceso productivo. Cuando una etapa está activa, la anterior lo ha sido previamente.

La etapa se representa con un cuadrado (o rectángulo), y dentro del mismo, se indica el número de orden. La primera etapa o etapa inicial, se representa con un cuadrado doble, y tiene que ser activada en primer lugar, antes del resto. Tenga en cuenta, que dentro de una misma rama, no se podrán activar dos etapas a la vez.

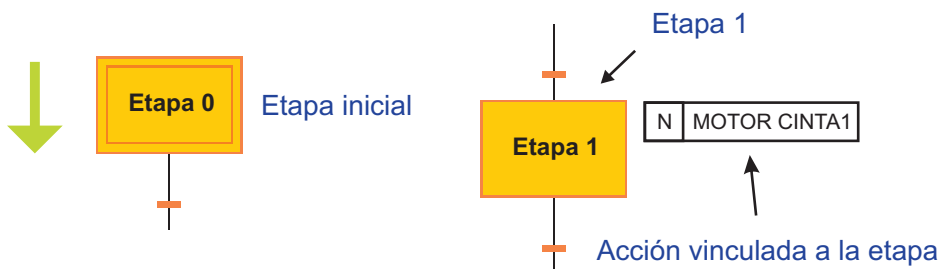


Figura 5.1. Etapa inicial y etapa 1, asociada a una acción.

Una etapa puede asociarse a una o más acciones, que estarán operativas cuando la etapa está activa.

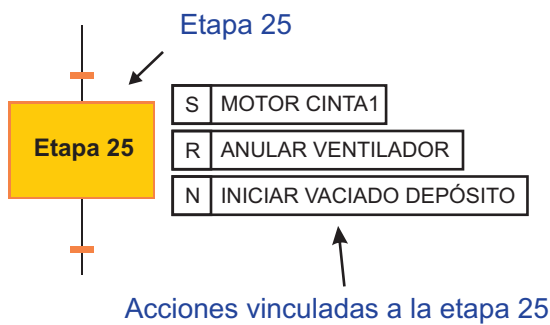
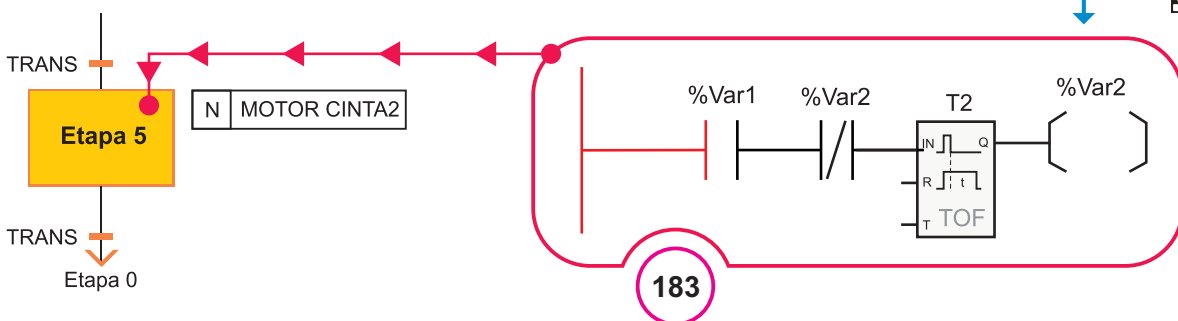


Figura 5.2. Una etapa puede contener varias acciones.

En la definición de un programa en GRAFCET, a cada etapa se le asocian una o varias acciones trabajo, pero en la práctica, algunos programas son capaces de incorporar "dentro" de la etapa, una pequeña subrutina, en un lenguaje determinado, que realizará una acción mientras esté activa la etapa, al terminar la etapa, o previamente a la activación de la etapa.



GRAF CET y SFC

GRAF CET

GRAFico Funcional de Control Etapa Transición.

SFC

Esquema secuencial de funciones. Sequential Function Chart

Secuencial

En un sistema secuencial, el valor del estado actual de una secuencia depende, además de valores de entrada de dicha secuencia, del estado de secuencias o etapas anteriores.

Ventajas del GRAFCET

- ▶ Permite un diagnóstico rápido del proceso.
- ▶ Simplifica los esquemas lineales.
- ▶ Permite distribuir un problema a otras secuencias.
- ▶ Se puede realizar la conversión a otros lenguajes de programación.

Figura 5.3. Algunos compiladores permiten la incorporación de subrutinas dentro de la etapa.



3 Transiciones

En el proceso secuencial, para que se ejecute una acción, ha tenido que efectuarse alguna condición que permita —o no— que se desarrolle la etapa. Estas condiciones se denominan transiciones, y son las que permiten que las etapas se desarrollen una tras otra. Cuando se produce una transición, se activa la etapa siguiente y se desactiva la anterior. Tenga en cuenta, que un programa en GRAFCET debe terminar por una transición.

En el siguiente gráfico, la condición es un final de carrera, que permite la activación de la etapa 15, que a su vez tiene asociadas dos acciones. La transición pertenece a la etapa anterior, en este caso, la 14.

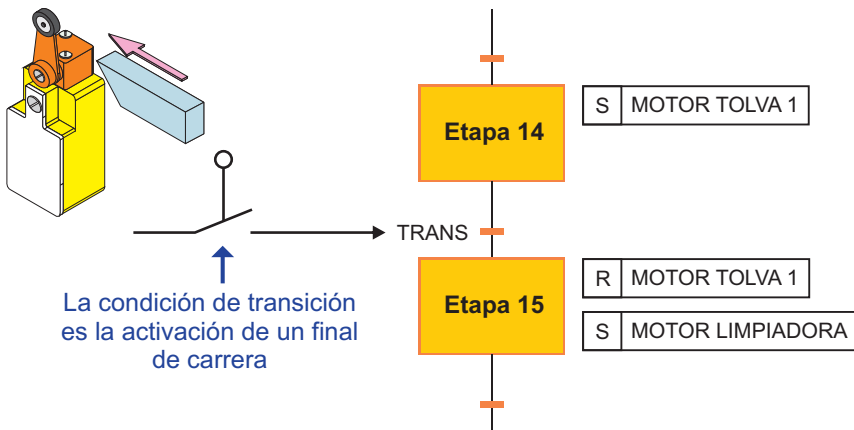
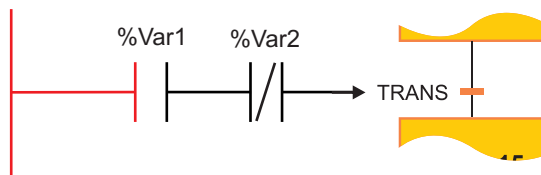


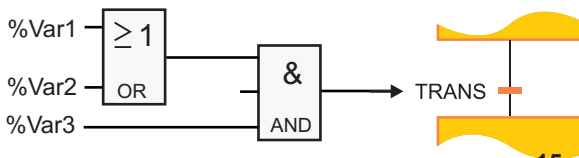
Figura 5.4. Transición.

Las condiciones de transición, pueden ser diversas como se muestra en la parte derecha de la página, y dependiendo del controlador, va a permitir que se puedan hacer en varios lenguajes de programación. Se destacan los lenguajes LD, FBD y ST.

Transición en Ladder (LD)



Transición en Puertas lógicas (FBD)



Transición en Texto estructurado (ST)

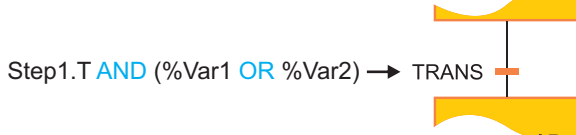
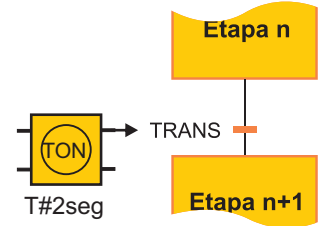


Figura 5.5. Condiciones de transición en diferentes lenguajes.

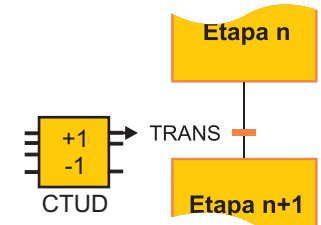
Condiciones de transición

Las condiciones de transición pueden ser varias y efectuar diferentes operaciones:

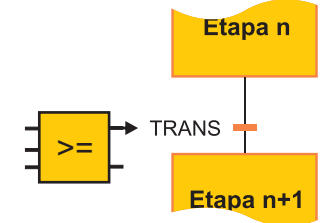
Temporizadores



Contadores



Comparadores



Asociaciones en serie



Asociaciones en paralelo



Flancos P y N



Negaciones



Figura 5.6. Condiciones de transición.

Entre etapa y etapa, debe existir una única transición, no permitiéndose otras opciones.

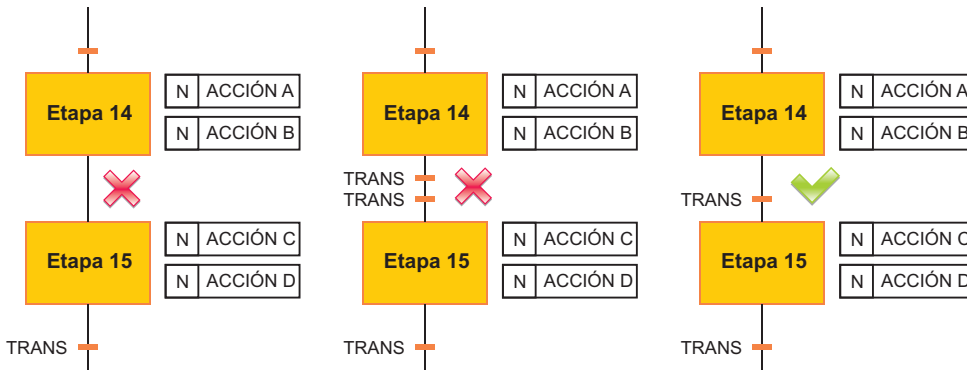


Figura 5.7. Entre etapa y etapa sólo se permite una transición.

3.1. La transición es un programa externo

Algunos controladores permiten realizar esta operación, que consiste en crear una subrutina en un lenguaje determinado, y que culmina a modo de transición.

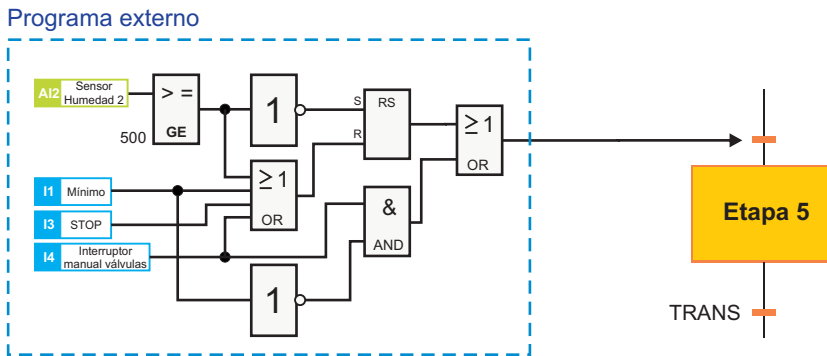


Figura 5.8. La transición es un programa externo.

3.2. La última transición

Cuando termina el proceso productivo, suelen desactivarse todas las acciones del mismo, para que el GRAFCET culmine hacia las condiciones iniciales. Para ello es preciso que todas las variables adopten los valores requeridos para la llamada a un nuevo ciclo de trabajo. Si el procedimiento termina en una acción, es necesario poner una transición, y si ésta no es necesaria, se implementa con un tiempo reducido o una variable de sistema TRUE.

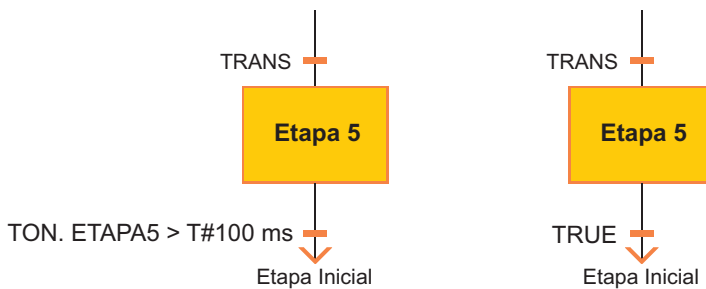
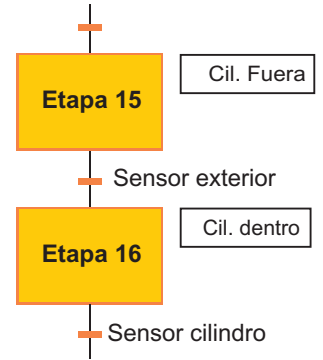


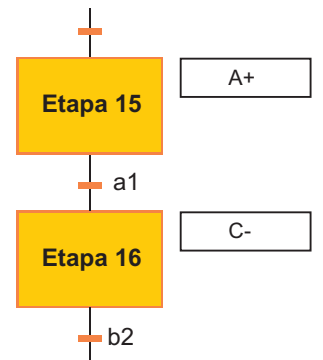
Figura 5.9. La última transición cuando no es vinculante con el programa.

Niveles del GRAFCET

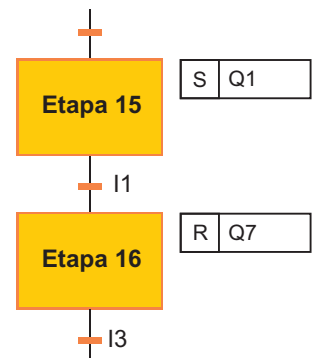
Nivel 1. Descripción funcional. Explica el funcionamiento pero no ofrece detalles.



Nivel 2. Descripción tecnológica. Define la tecnología usada.



Nivel 3. Descripción operativa. Explica el automatismo.



4 Operaciones en GRAFCET

4.1. Características de las acciones



Dentro de un cuadro de acción, se puede indicar un comportamiento o calificador para un componente en concreto. Los principales se muestran en una tabla.

N	Non-stored	La acción permanece activa como el paso (mientras dura la etapa).
R	Overriding Reset	La acción se desactiva.
S	Set (Stored)	La acción se ejecuta en el momento que un paso (una etapa) se torna activo y continua siendo ejecutado, aún cuando el paso ya ha sido desactivado, hasta que recibe un reset.
L	Time Limited	La acción se ejecuta mientras el paso (la etapa) está activo y se ejecuta como mínimo durante el lapso de tiempo indicado, sin embargo como máximo hasta que el paso sea desactivado.
D	time Delayed	La ejecución de la acción se inicia recién, cuando tras la activación de paso haya concluido el tiempo de retardo indicado y el paso aún continúa activo. La acción se ejecuta, hasta que el paso se desactiva.
P	Pulse	La acción se ejecuta exactamente una vez (un pulso) cuando se activa el paso.
SD	Stored and time Delayed	La ejecución de la acción se inicia, cuando tras activarse el paso ha concluido el tiempo de retardo indicado. Esta se ejecuta hasta tanto reciba un reset.
DS	Delayed and Stored	La ejecución de la acción se inicia recién, cuando tras la activación de paso haya concluido el tiempo de retardo indicado y el paso aún continúa activo. Esta acción se ejecuta entonces, hasta que reciba un reset.
SL	Stored and time limited	La acción se ejecuta, en el momento que se activa el paso. Se ejecuta, hasta tanto esté concluido el tiempo indicado o reciba un reset.



Cuando hay una única rama de GRAFCET, el sentido lógico sugiere que cuando termina la última etapa, comienza de nuevo por la primera con un salto ascendente.

Según el programa, esta operación se guiará con flechas o con indicadores.

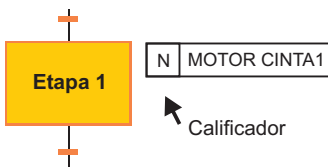


Tabla 5.1. Calificadores más usuales.

Las indicaciones de tiempo en los calificadores L, D, SD, DS y SL, se especifican en formato TIME. Por otro lado, cabe indicar que los calificadores más empleados suelen ser N, S y R.

4.2. Salto condicional de etapa



Se puede establecer en sentido ascendente, o descendente, y supone que se omite la ejecución de ciertas etapas.

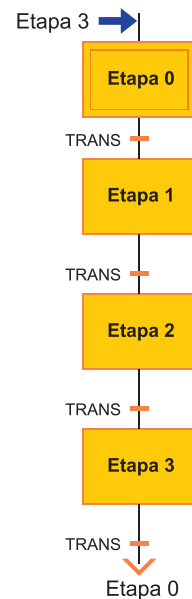
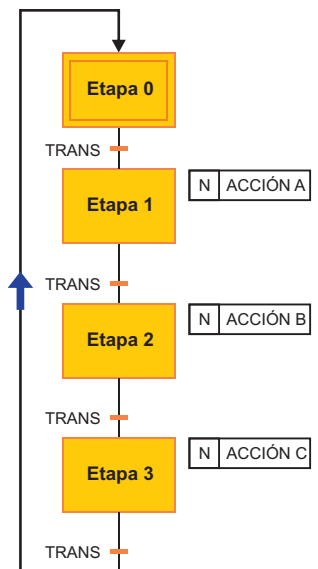
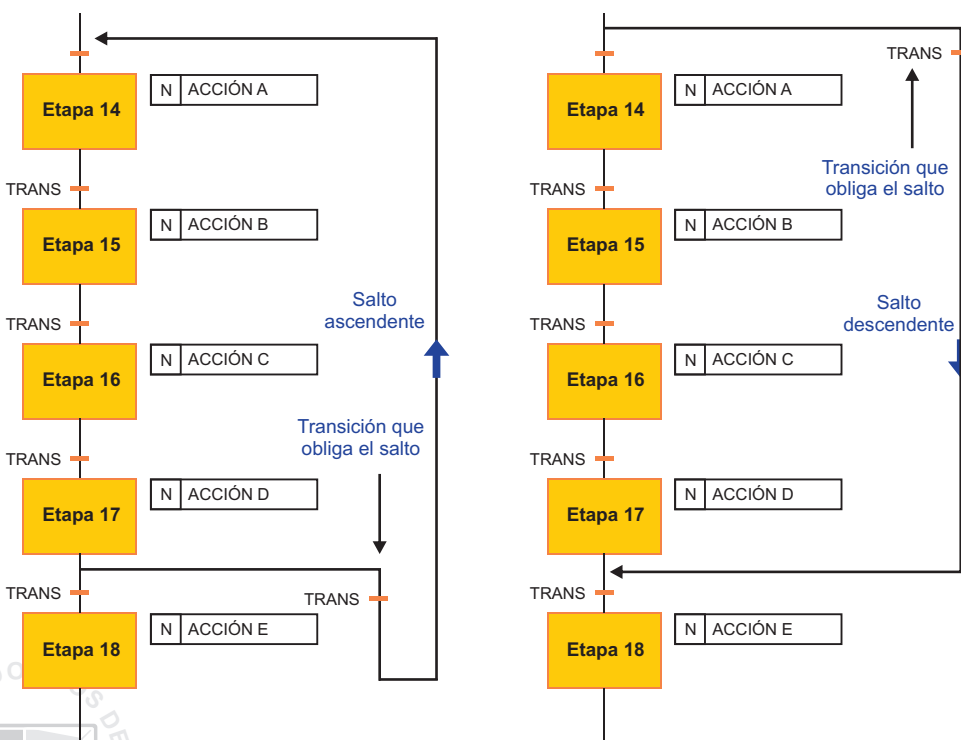


Figura 5.11. Retorno de etapa y dos formas de indicarlo.

Figura 5.10. Operaciones de salto condicional.

4.3. Direccinamiento condicional. Ramas alternativas



La ejecucin de un programa, puede solicitar que las condiciones de trabajo, varien a lo largo del proceso productivo. El programa en GRAFCET, ser capaz de elegir la rama de trabajo necesaria para los cambios producidos. Para hacer esto, se pueden usar ramas alternativas o paralelas.

Una rama alternativa, permite cambiar un tramo del programa por otro determinado, ordenado por una transicin especfica.



En una rama alternativa, siempre debe preceder una transicin por rama, del mismo modo, debe existir otra al finalizar cada rama alternativa.

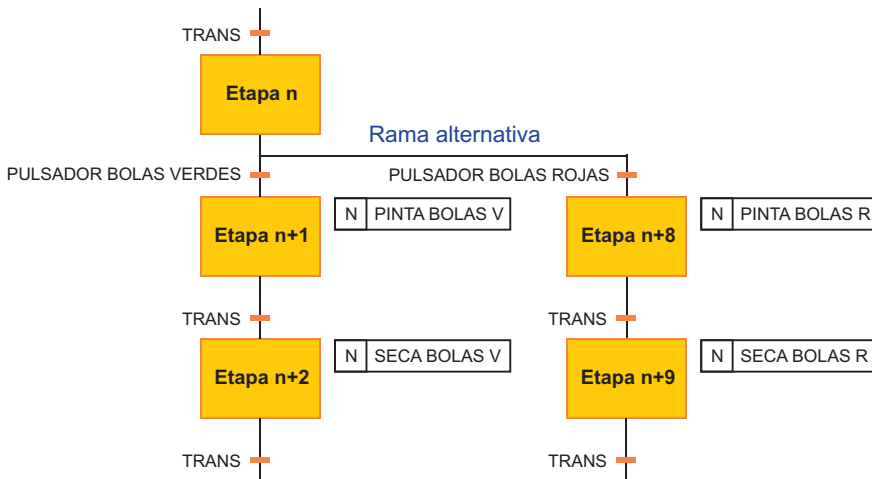


Figura 5.12. Rama alternativa.

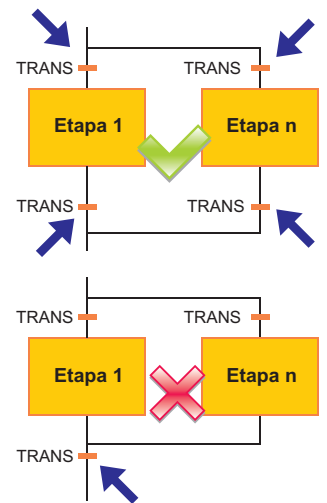


Figura 5.14. Transiciones en ramas alternativas.

Ejemplo

Proceso de embotellado, taponado y etiquetado. Unos sensores determinarn el tamao de la botella, y en consecuencia, el volumen de lquido, tamao de tapn y dimensin de la etiqueta.

Segn este detector, se efectuar un camino u otro de secuencias. Note, que las diferentes fases pueden culminar en una nica lnea de programacin.

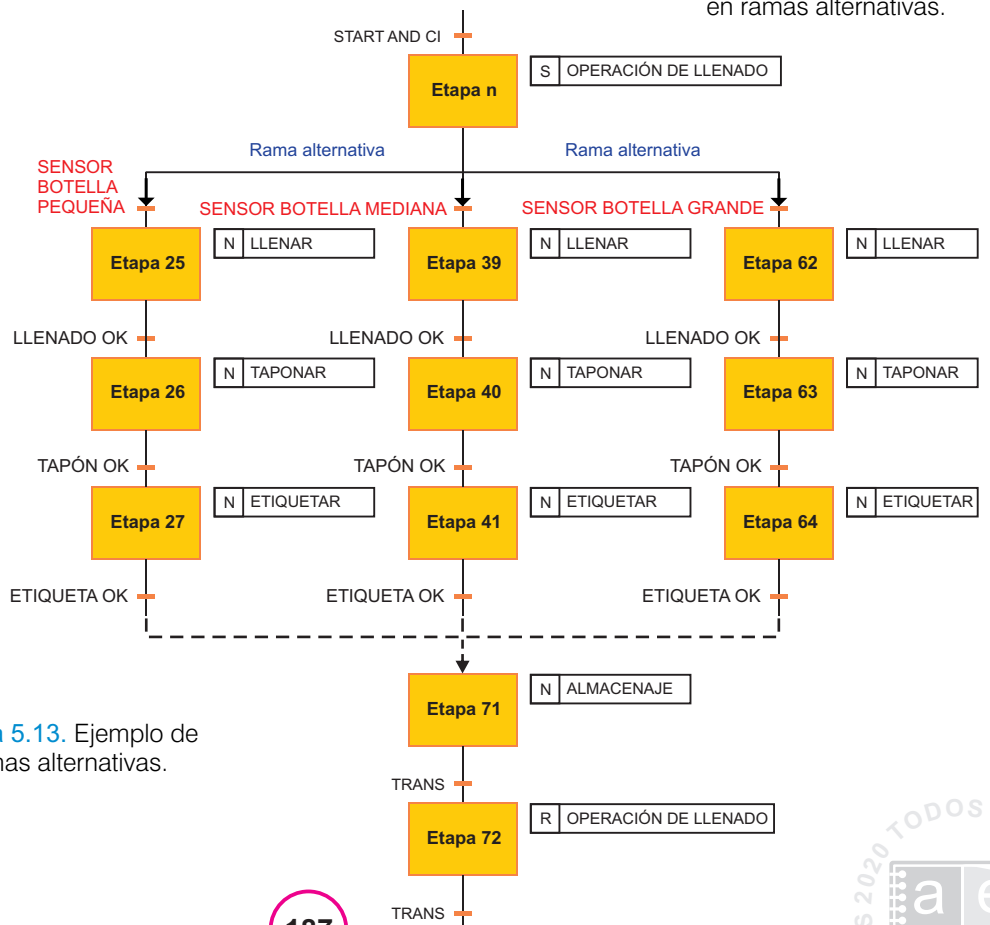


Figura 5.13. Ejemplo de ramas alternativas.

4.4. Secuencias simultáneas. Ramas paralelas

Dos trazos en paralelo, nos indican que se van a realizar varios estados operativos al mismo tiempo. El retorno de las diferentes secuencias también pueden converger en una línea.

Cuando se crea una rama paralela, sólo se usará una única transición, ya que al ejecutarse a la vez, no se admite más que una condición.

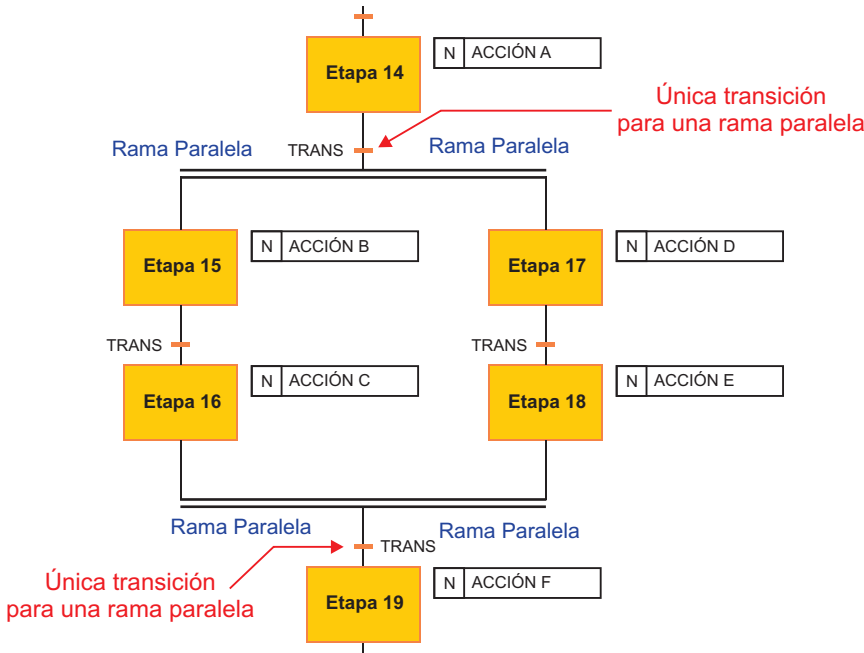


Figura 5.15. Ramas paralelas.

Ejemplo

Semáforo simple con intermitentes. Se compone de tres lámparas, Roja, Verde, Ámbar, y dos lámparas adicionales para los muñecos: Muñeco verde y Muñeco rojo. Las variables de los muñecos son las ramas paralelas.

Adicionalmente, se ha insertado una pequeña programación en ladder para generar el intermitente del cambio del muñeco verde al rojo.



Figura 5.16. Ejemplo para explicar las ramas paralelas.

Transiciones en ramas paralelas

En una rama paralela, se establece una única transición por bifurcación, no permitiéndose otras opciones. Del mismo modo, el retorno de ramas paralelas convergen en una única transición.

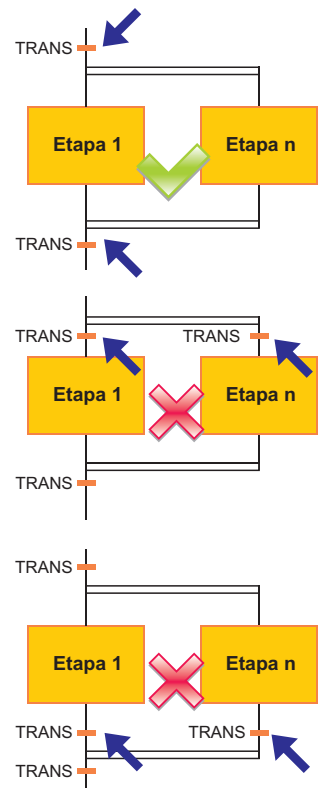


Figura 5.17. Transiciones en ramas paralelas.

ENTRADA	
START	Comienzo de ciclo
SALIDAS	
ROJA	Lámpara roja del semáforo
VERDE	Lámpara verde del semáforo
AMBAR	Lámpara ámbar del semáforo
MVERDE	Muñeco verde
MROJO	Muñeco rojo
AUXILIARES	
MVERDE1	Auxiliar 1 muñeco verde
MVERDE2	Auxiliar 2 muñeco verde

Tabla 5.2. Propuesta de variables a usar.

Para conseguir que una misma variable esté fija e intermitente, se han usado dos variables auxiliares, más la variable final, del siguiente modo:

◆ La variable MVERDE1, representa la activación de la variable final MVERDE (muñeco verde), de forma continua.

◆ La variable MVERDE2, representa la activación de la variable final MVERDE de forma intermitente.

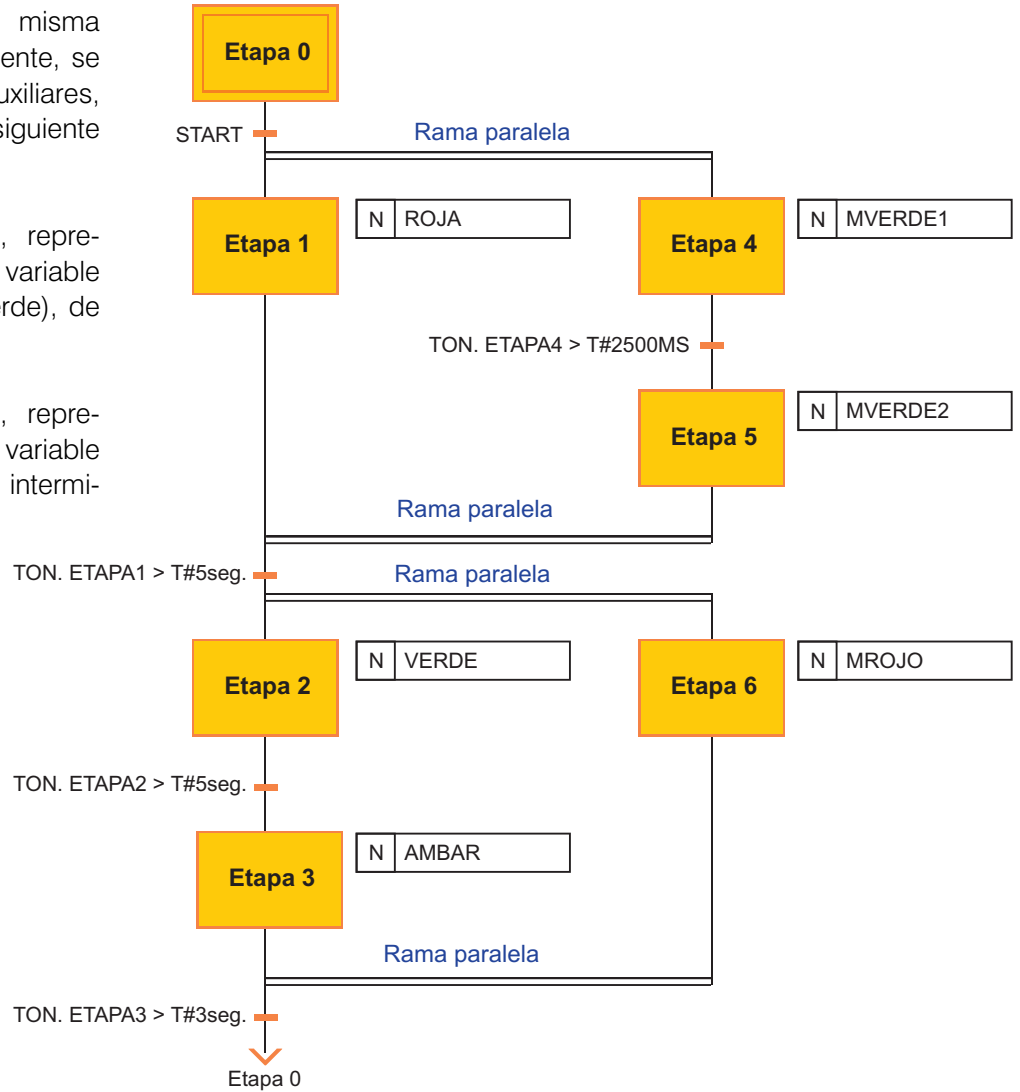
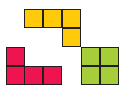


Figura 5.18. Programación en GRAFCET del ejemplo propuesto para explicar las ramas paralelas.



Ejercicio



1. Realiza el programa en GRAFCET que desarrolla el ejemplo de uso de ramas paralelas.

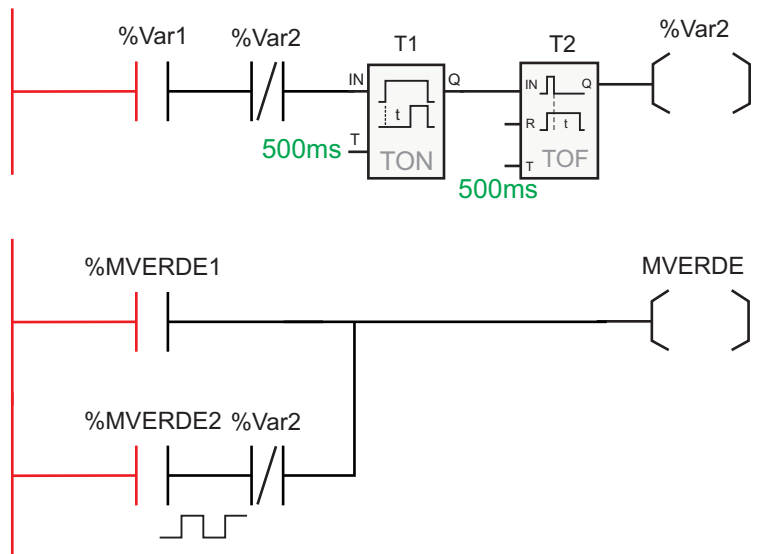


Figura 5.19. Subrutina para provocar un efecto intermitente en la lámpara Muñeco verde, antes de pasar a Muñeco rojo.

5 Condiciones iniciales



Son los requisitos previos, que debe presentar la instalación automática, antes de la puesta en marcha del proceso de manufactura. Representan por tanto, los estados de diferentes sensores, actuadores, valores específicos, posiciones, entre otros, antes de que comience la secuencia de funcionamiento.

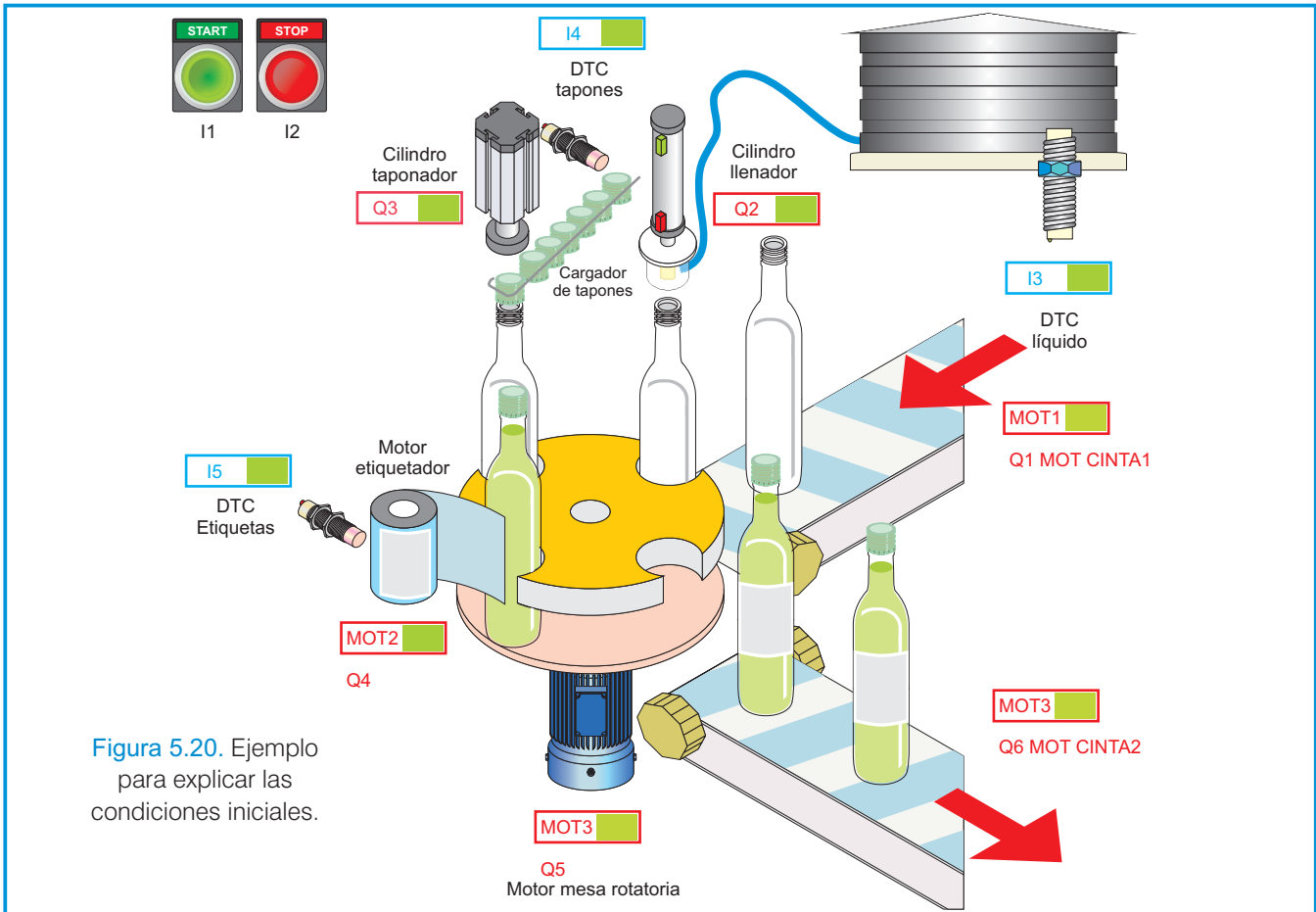


Figura 5.20. Ejemplo para explicar las condiciones iniciales.

Ejemplo. Planta embotelladora rotativa

Una planta embotelladora, llena, tapona y etiqueta botellas de un producto determinado.

Las condiciones iniciales deben ser: que exista fluido suficiente para el llenado de un ciclo, que existan tapones, y también etiquetas. Si se cumple todo esto, se podrá efectuar un ciclo de trabajo, que podrá operar con 1 ó 20 botellas a la vez, en función del tipo de máquina.

Los sensores son los que aportan información al programa

Cada ciclo de programa se hace con una botella.

El funcionamiento es el siguiente:

1. Al presionar I1 Start, —si se cumplen las condiciones iniciales—, se pone en marcha Q1 Motor cinta1. Desplaza la botella a la zona de llenado.

2. El siguiente evento es detener la cinta 1 y activar Q2 Cilindro llenador, que desciende y llena la botella con líquido procedente de un depósito al efecto. Una vez concluido el proceso de llenado, el cilindro llenador se recoge. La condición en esta fase la indica el sensor I3 que advierte presencia de líquido en el depósito.

3. El motor de la mesa rotativa Q5, gira 45°; se activa el cilindro taponador, que coloca un tapón a la botella. La condición en esta fase es el sensor de tapones I4.

4. El motor de la mesa rotativa Q5, gira 45°; se activa el motor de la etiquetadora Q4, que le adhiere el adhesivo a la botella. La condición en esta fase, es el sensor de rollo de etiquetas I5.

5. El motor de la mesa rotativa Q5, gira 45°, y el motor de la cinta 2 Q6, se pone en marcha desplazando la botella completa a la zona de almacén....

Por lo tanto, las condiciones iniciales van a asegurar, que al menos una botella se va a completar correctamente, ya que los sensores indican que existe producto para completar un ciclo. La expresión podría ser la siguiente:

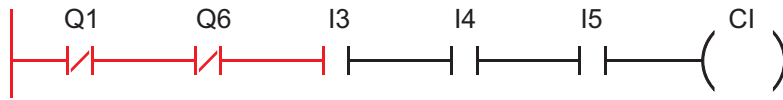


Figura 5.21. Programa para generar una variable de condiciones iniciales.

Cuyo significado es: para que se active la variable de condiciones iniciales CI, los motores de las cintas han de estar desactivados (Q1 y Q6 = OFF), el sensor de líquido del depósito debe registrar producto (I3=TRUE), el sensor de tapones, debe comprobar que hay tapones en su cargador (I4=TRUE), y el detector de rollo de etiqueta, indica que éstas existen (I5=TRUE).

Note el programa GRAFCET; la condición inicial CI está asociada en serie con el pulsador START, para comenzar el ciclo de trabajo. Pero si no se cumple CI=TRUE, se inicia la rama alternativa, que activa un aviso que indica que al presionar START, el ciclo no arranca, porque no se cumplen todas las condiciones iniciales.

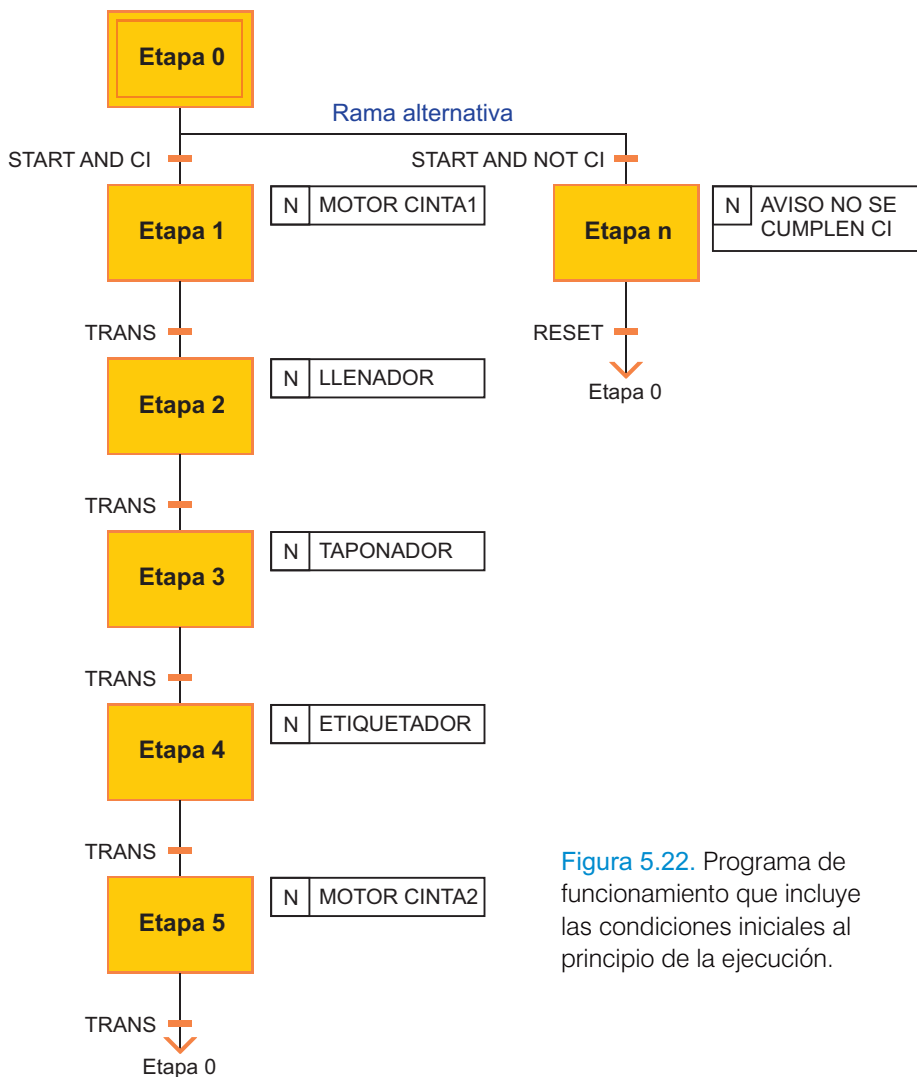


Figura 5.22. Programa de funcionamiento que incluye las condiciones iniciales al principio de la ejecución.

Programa CI

El diseñador-programador decide donde ubicar el programa que ejecuta las condiciones iniciales.

Si el programa no es muy extenso, se ubicará en el principal, y el GRAFCET en una subrutina.

Si el programa es muy grande, se ubicará donde el programador entienda que se relaciona con el programa GRAFCET.

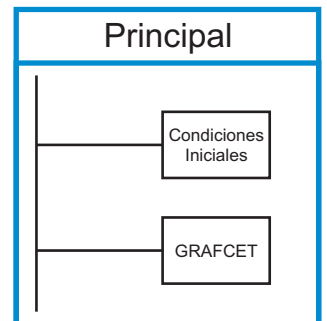
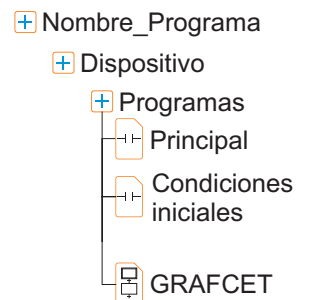


Figura 5.23. Las condiciones iniciales se pueden agrupar en una subrutina, o en el programa principal.

Ejercicio



2. Realiza el programa en GRAFCET que desarrolla el ejemplo mostrado. Incluye las condiciones iniciales en una subrutina específica.

Ejemplo. LLenado por dosificadora

El ejemplo responde a la siguiente secuencia:

1. Al presionar START se activa el motor de la cinta transportadora.
2. Llega un vaso, y al situarse a la altura de B1, se detiene la cinta.
3. Se activa el cilindro A+, con el propósito de cargar líquido del depósito como una jeringa.
4. Se activa el cilindro B+ que gira el conjunto 45°.
5. Se desconecta el cilindro A+, vertiendo el contenido de la jeringa al vaso.
6. Se desconecta el cilindro B+ que gira el conjunto a la posición inicial.
7. Se activa el motor de la cinta. El vaso se desplaza y el sistema está a la espera de la llegada de un nuevo vaso.

Las condiciones iniciales CI de este proceso, podrían ser:

- a) Que el recipiente principal disponga de líquido. B0=1.
- b) Que no esté activo el cilindro B+. B+=0.
- c) Que el cilindro A+ esté extendido. A+=1.
- d) Que el motor de la cinta no esté activo. MOT=0.

Las condiciones iniciales que definen lo anterior:

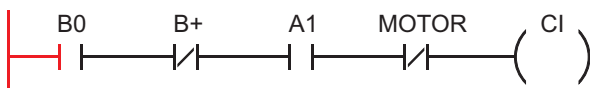


Figura 5.24. Condiciones iniciales.



3. Realiza el programa en GRAFCET que desarrolla el ejemplo mostrado "Llenado por dosificadora". Incluye las condiciones iniciales en una subrutina específica.

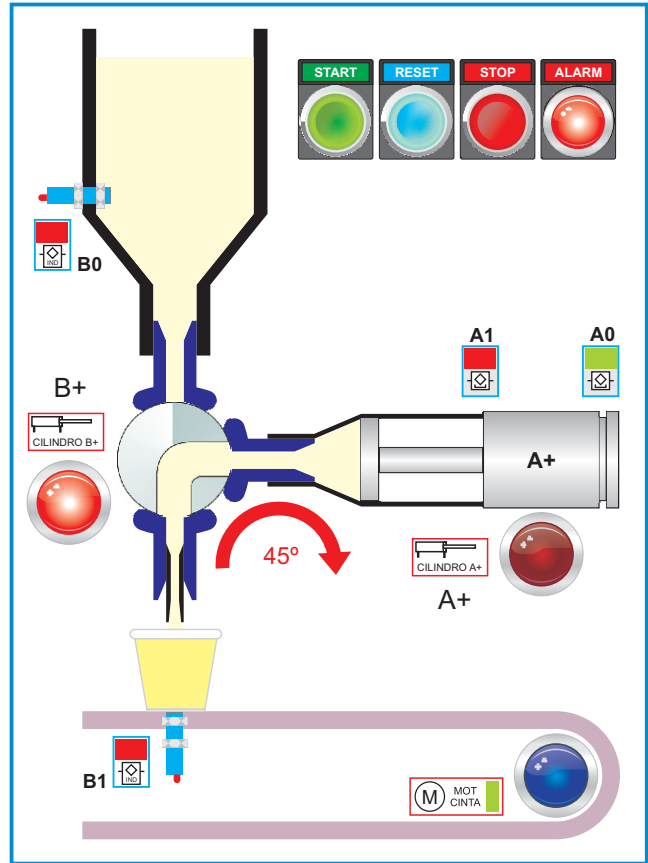


Figura 5.26.

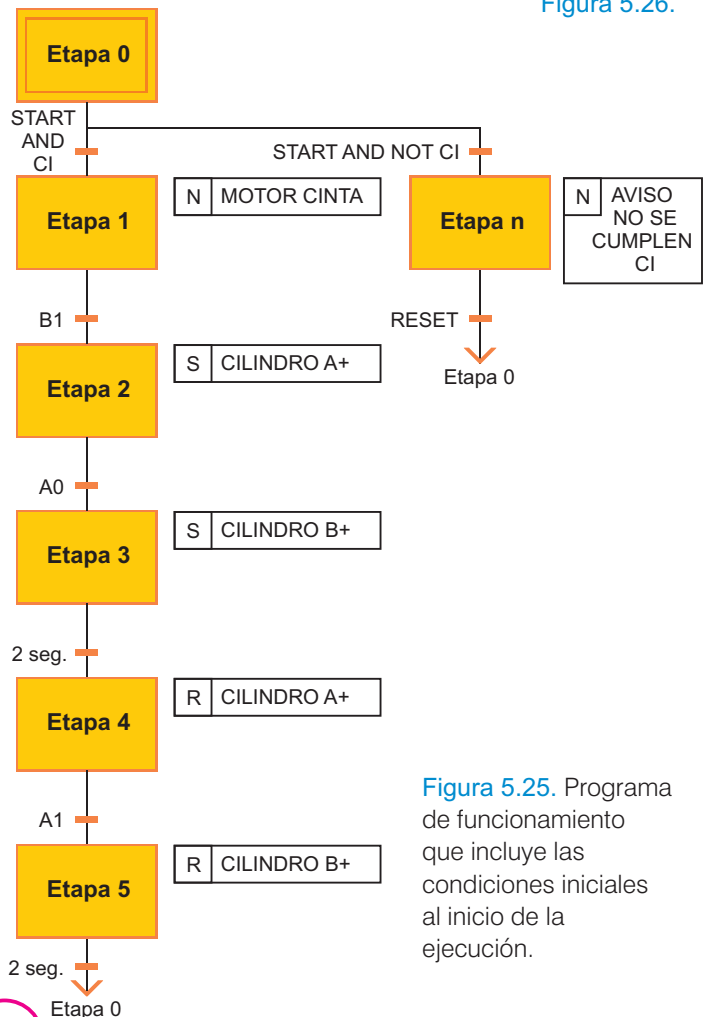
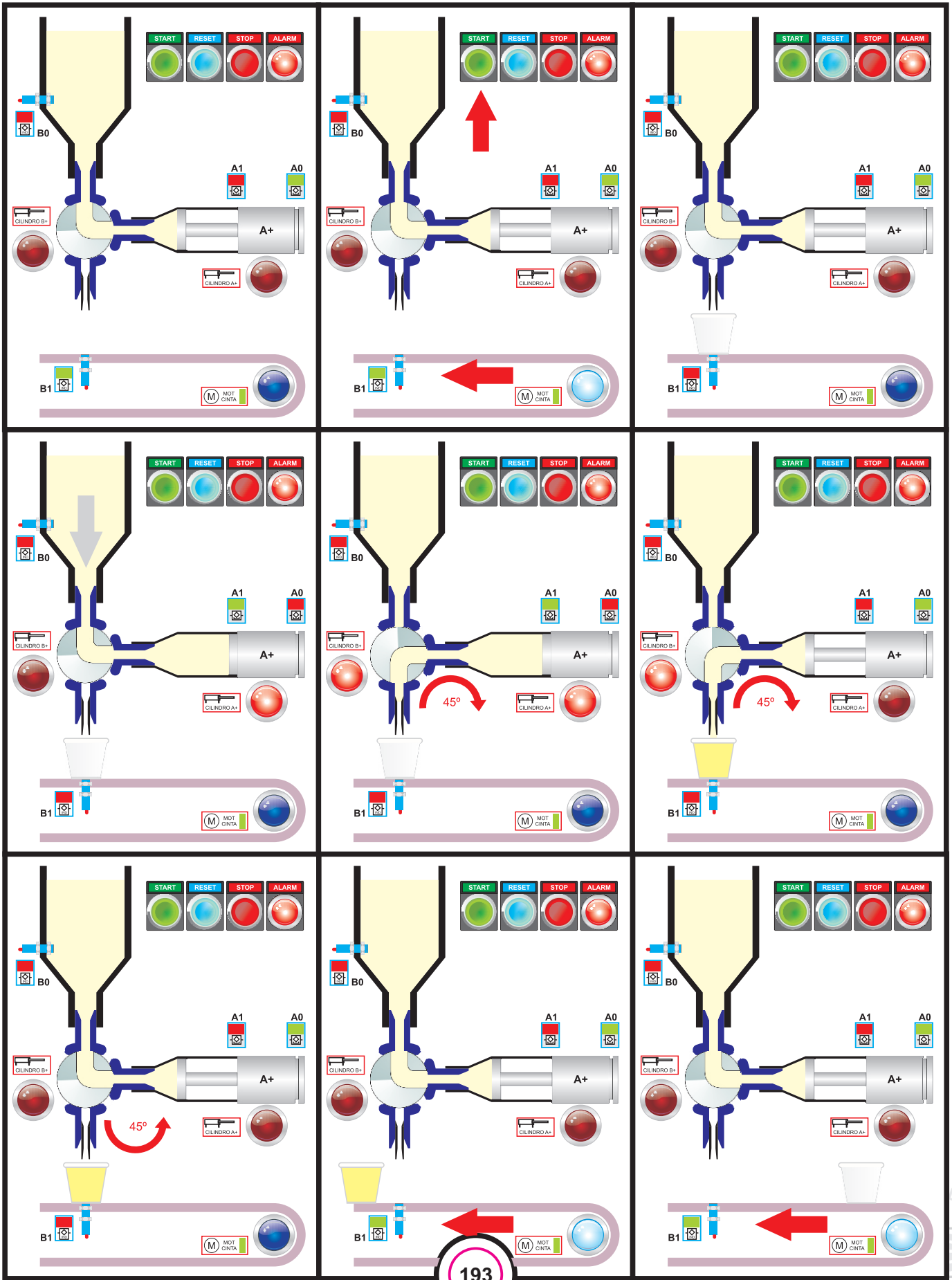


Figura 5.25. Programa de funcionamiento que incluye las condiciones iniciales al inicio de la ejecución.

Figura 5.27. Llenado por dosificadora. Secuencia de funcionamiento.



6 GRAFCET con funciones SET-RESET

No todos los controladores programables disponen del lenguaje GRAFCET. La definición de este lenguaje, se puede hacer perfectamente utilizando funciones SET-RESET con otro lenguaje.

El comportamiento del GRAFCET respecto a funciones SET-RESET, es el siguiente:

- ◆ La etapa en curso habilita el circuito.
- ◆ La condición de transición alimenta las funciones SET-RESET.
- ◆ La función SET, habilita la etapa siguiente y la función RESET anula la etapa actual.
- ◆ El ciclo se repite con la siguiente etapa-transición.

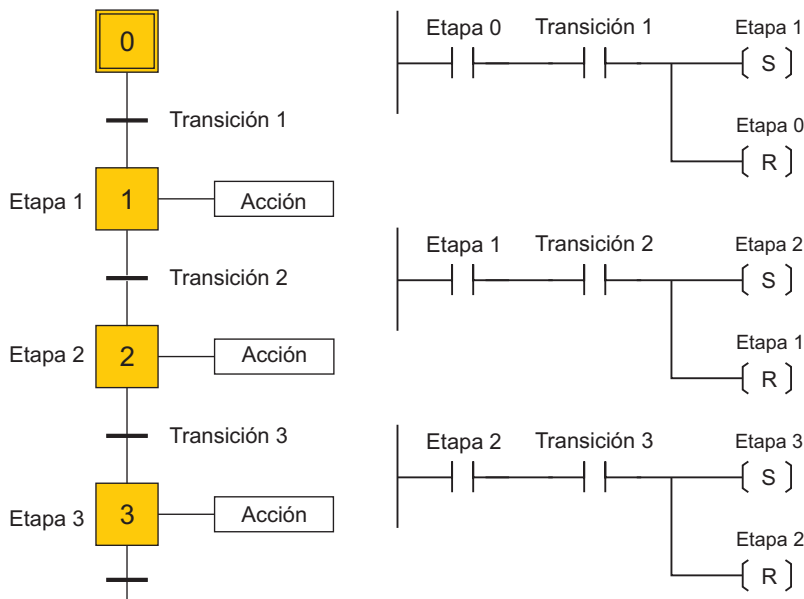


Figura 5.28. GRAFCET equivalente con funciones SET-RESET.

Ejemplo. Desplazamiento de objetos por circuito neumático

El ejercicio consiste en un sistema electroneumático capaz de coger un tornillo en un punto determinado, y desplazarlo a otro, a través de dos cilindros neumáticos y una pinza, también neumática. El sistema electroneumático está compuesto por:

- ◆ Un cilindro sin vástago A, movido por una electroválvula biestable, donde A+, mueve el conjunto sentido directo, y A- hace lo propio con el sentido inverso. Este cilindro dispone de tres sensores magnéticos, A0, A1 y A2, para determinar la posición del conjunto en sus extremos y en el centro.
- ◆ Un cilindro B, que está sujeto al conjunto A, para desplazarse sentidos directo e inverso, para elevar ó descender una pinza que lleva consigo. Está manejado por una electroválvula monoestable B+. Este cilindro dispone de dos sensores magnéticos B0 y B1, para determinar la posición del vástago.
- ◆ Un cilindro tipo pinza neumática C, con el propósito de coger el tornillo. Está manejado por una válvula monoestable C+. Este cilindro dispone de dos sensores magnéticos C0 y C1, para determinar si la pinza está cerrada o abierta.
- ◆ Un detector capacitivo, que advierte la presencia del tornillo.

Condiciones de transición en GRAFCET con funciones RS

En una programación realizada en GRAFCET, la condición de transición puede ser perfectamente única, y el programa responde correctamente, ya que siempre prevalece la etapa en curso.

En caso de GRAFCET RS, las condiciones de transición deben ser un mínimo de dos, un contacto de la variable que indica la etapa anterior, y la condición propia del desarrollo del programa.

A continuación, la variable de la etapa en curso habilita las acciones pertinentes.

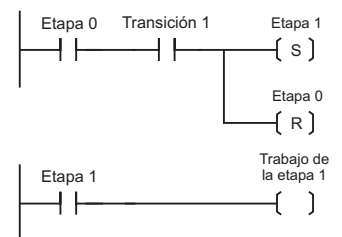


Figura 5.29. Forma correcta de realizar las transiciones en GRAFCET RS.

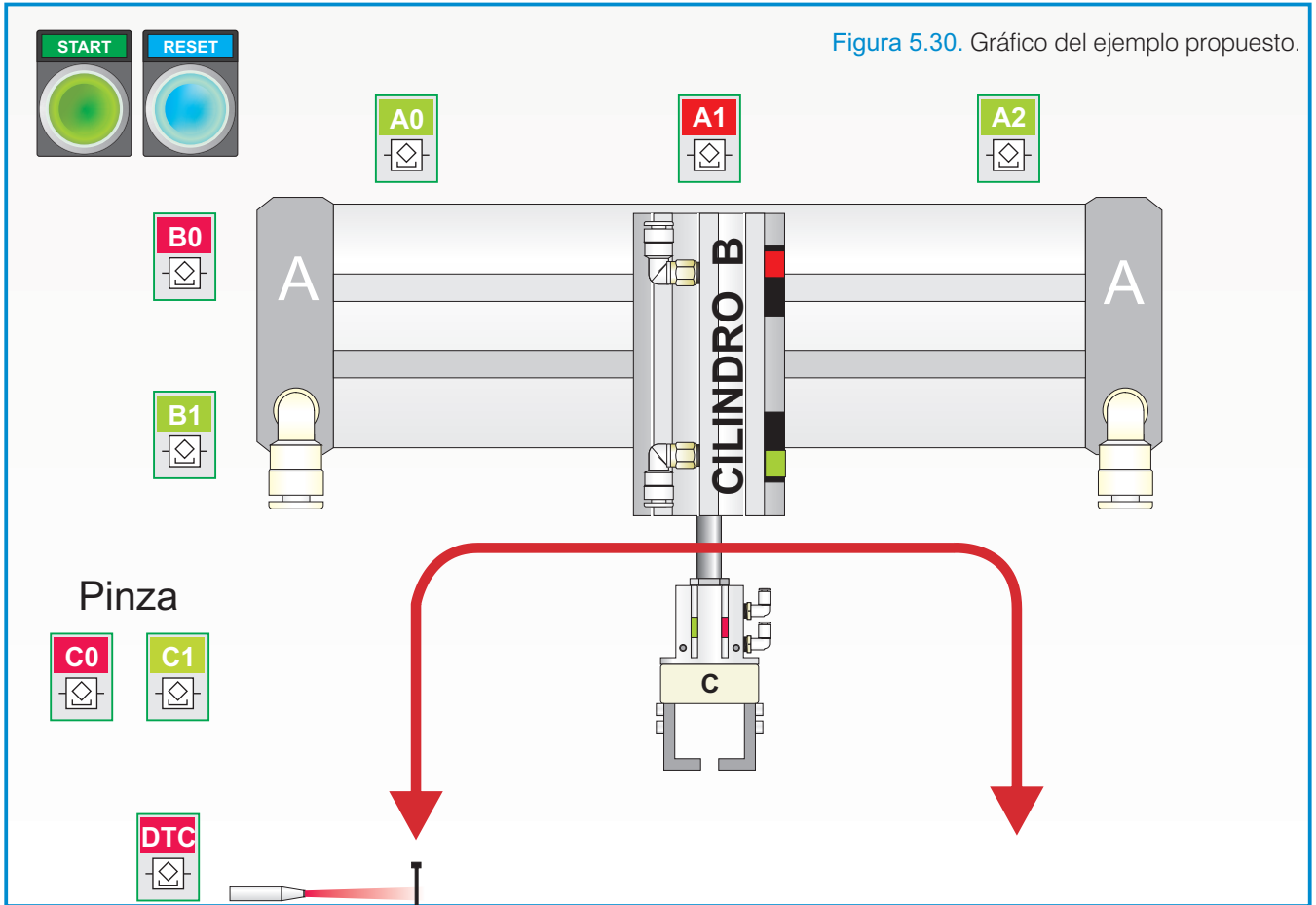


Figura 5.30. Gráfico del ejemplo propuesto.

1. Las condiciones iniciales son:

- ▶ Que el cilindro B esté en la posición A1.
- ▶ Que el detector DTC, registre que hay una pieza. DTC=1.
- ▶ Que la pinza C, esté abierta. C0=1.
- ▶ Que el vástago del cilindro B, esté recogido. B=0.

2. Al presionar START, se activa la electroválvula A-, y mueve el conjunto B y C, sentido inverso.

3. Cuando llega a la altura de A0, se conecta la electroválvula B+, que hace que el vástago B descienda.

4. Cuando El vástago B llega a su extremo inferior (B1=1), se cierra la pinza (C1=1) fijando el tornillo.

5. En ese instante, se anula B+, y el vástago se recoge, elevando el tornillo consigo.

6. Cuando B se ha recogido completamente (B0=1), se anula A-, y se activa A+, que obliga al conjunto a desplazarse sentido directo.

7. Cuando el conjunto llega hasta A2, se activa la electroválvula B+=1, que hace que descienda el conjunto.

8. Cuando llega hasta el extremo inferior (B1=1), se abre la pinza (C0=1). Se libera el tornillo.

9. En ese instante, se anula B+ y se eleva, hasta recogerse completamente (B0=1).

10. Se anula A+, y se activa A-, desplazando el conjunto sentido inverso hasta llegar a A2, donde se anula A-, y el sistema vuelve a estar en condiciones iniciales.

VARIABLES	
ENTRADAS	
START	Puesta en marcha
RESET	Puesta cero
E1 a E9	Etapas
A0	Sensor posición directa cilindro A
A1	Sensor posición intermedia cilindro A
A2	Sensor posición inversa cilindro A
B0	Sensor cilindro B recogido
B1	Sensor Cilindro B extendido
C0	Sensor pinza C abierta
C1	Sensor pinza C cerrada
DTC	Detector de tornillos
SALIDAS	
A+	Electroválvula biestable sentido directo
A-	Electroválvula biestable sentido inverso
B+	Electroválvula monoestable B
C+	Electroválvula para pinza C

Tabla 5.3. Variables del ejercicio propuesto.

Figura 5.31. Secuencia de funcionamiento. 1 de 2.

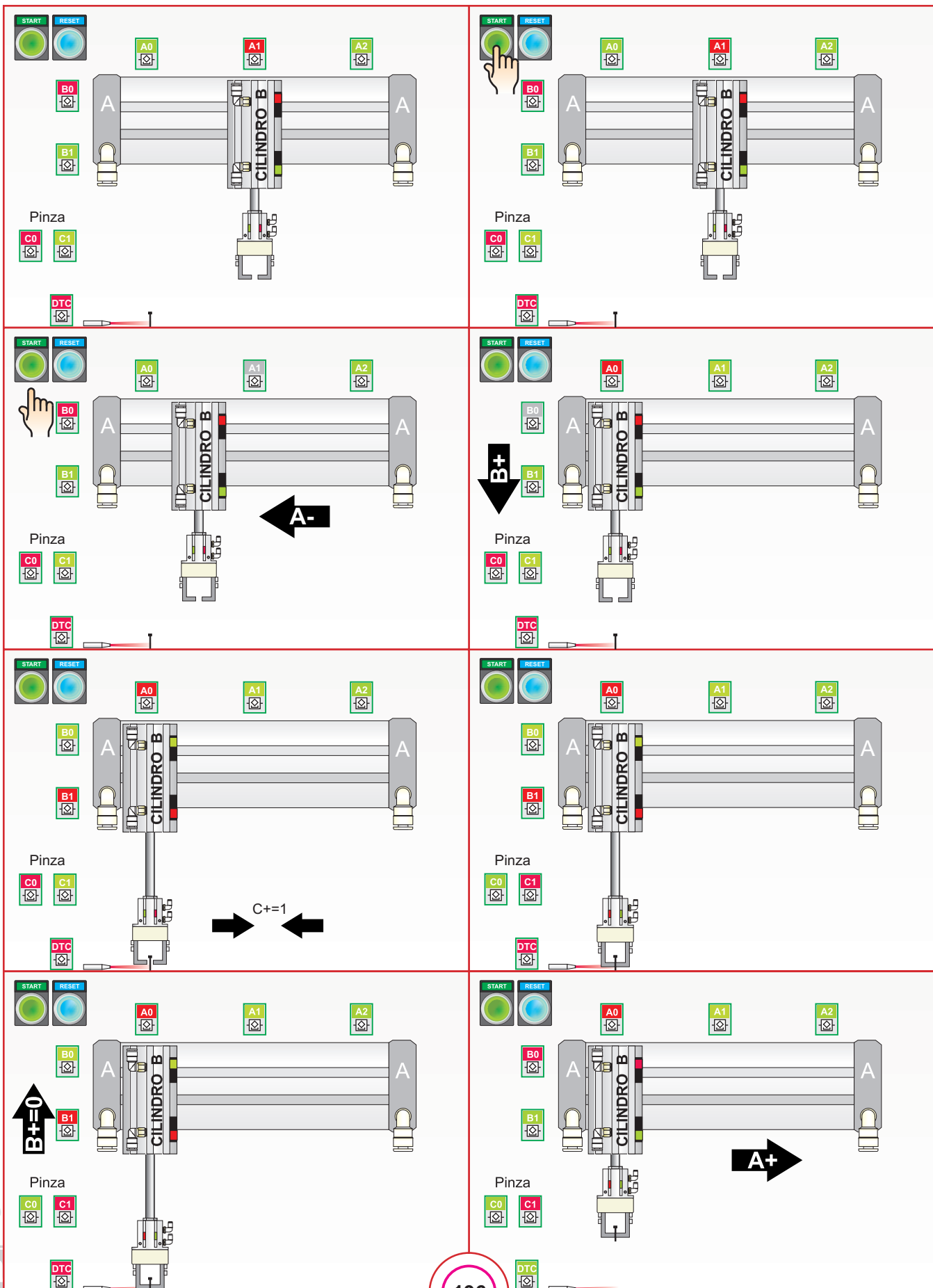
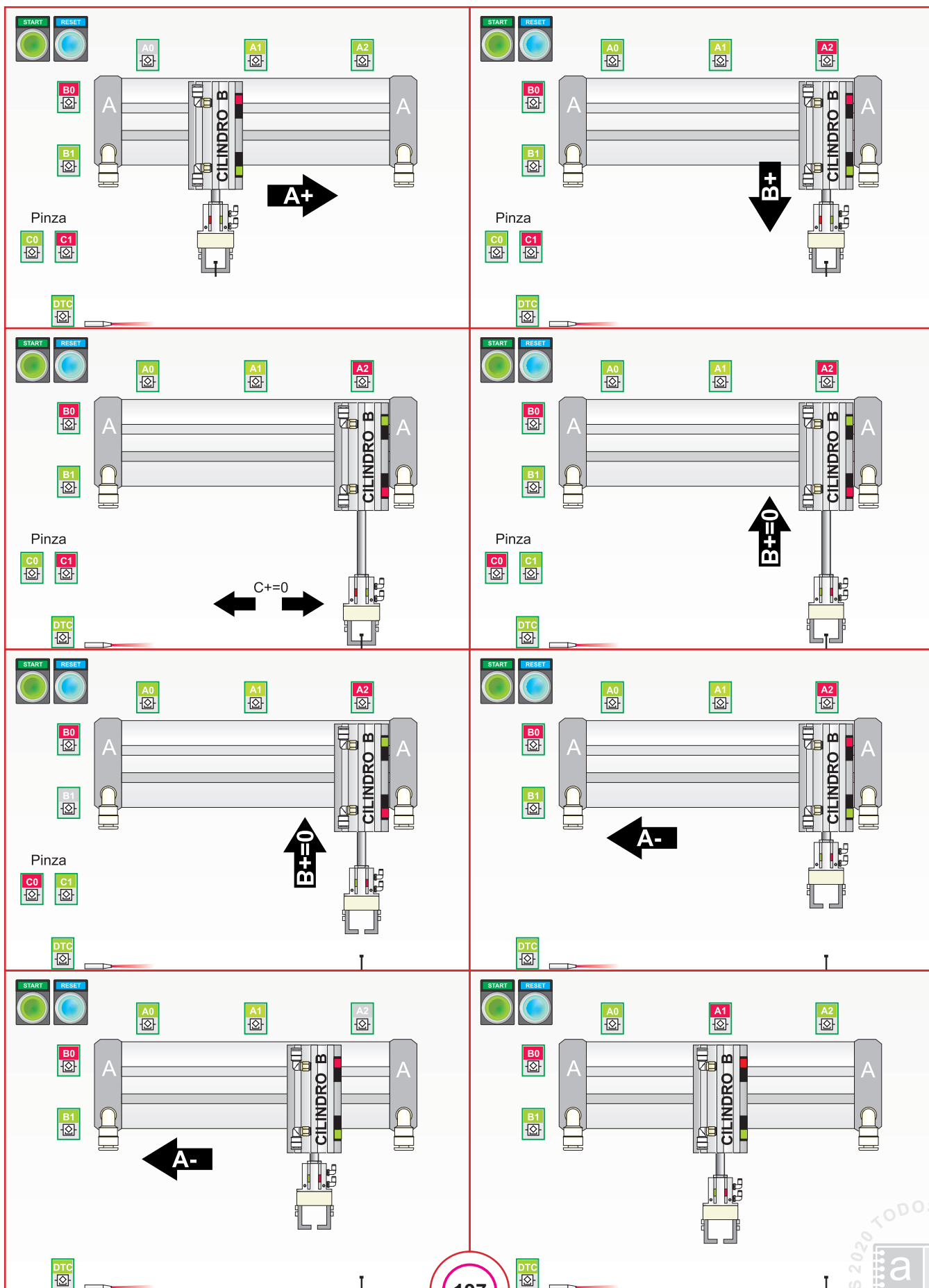
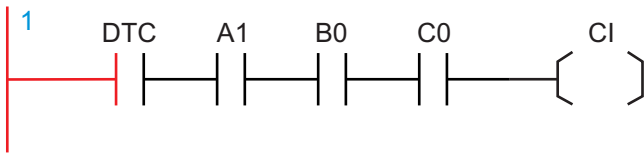


Figura 5.32. Secuencia de funcionamiento. 2 de 2.

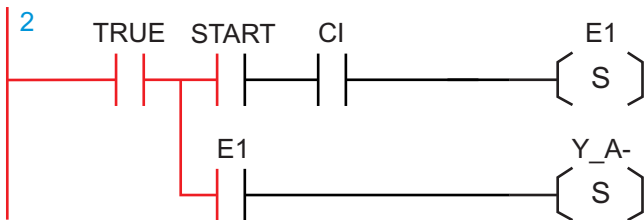


Programa

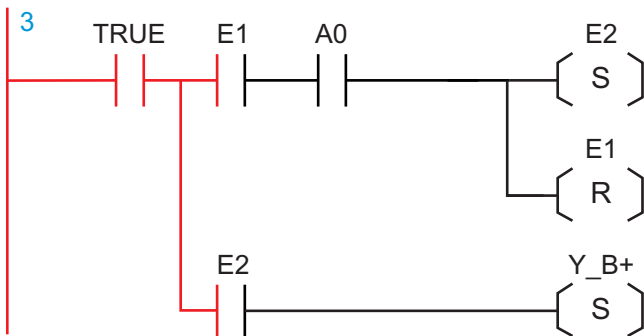
1. La primera línea de programación es la definición de las condiciones iniciales CI.



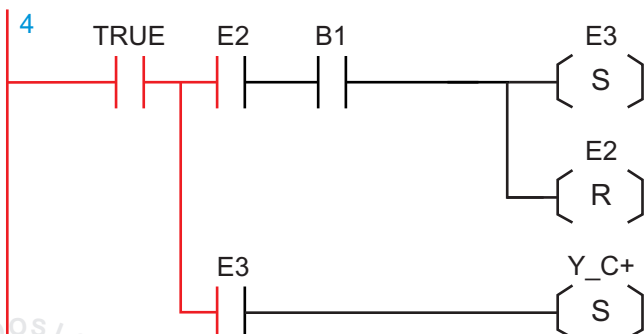
2. Si se cumplen las CI, al presionar START, se activa la Etapa 1, y ésta conecta la válvula A- sentido inverso.



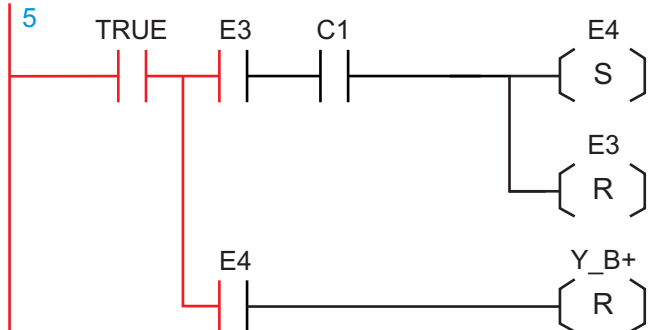
3. Cuando el conjunto llega hasta A0, se activa la etapa 2, se anula E1, y se conecta el cilindro de bajada B+.



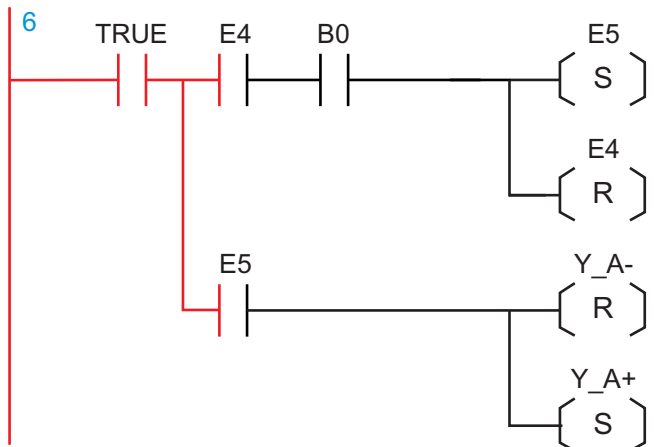
4. El cilindro B, desciende hasta su extremo B1, que habilita la etapa 3, y anula E2. Se cierra la pinza C+=1.



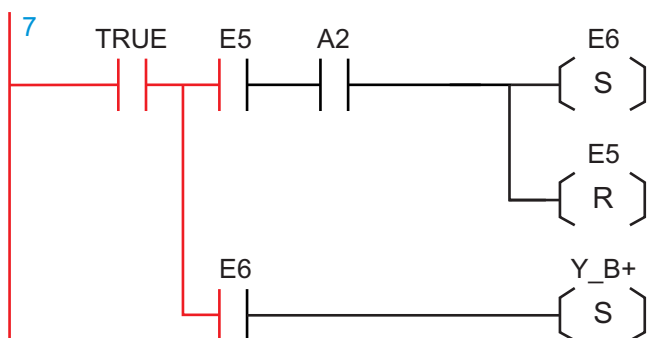
5. El cierre de la pinza provoca la activación de la etapa 4, anulación de E3, y anulación del cilindro B+, que se eleva.



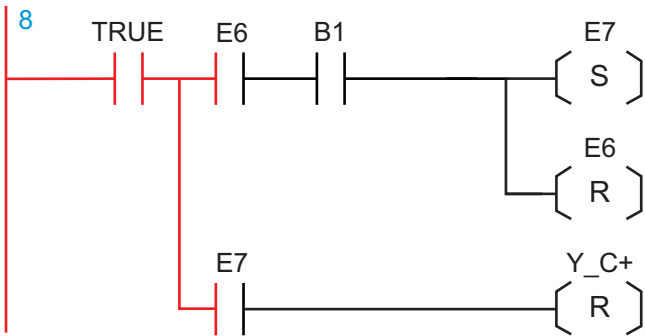
6. Al elevarse el cilindro B, provoca dos acciones, anula a A-, conecta a A+, para desplazar el conjunto sentido directo. Todo ello con la activación de la etapa 5 y anulación de la etapa 4.



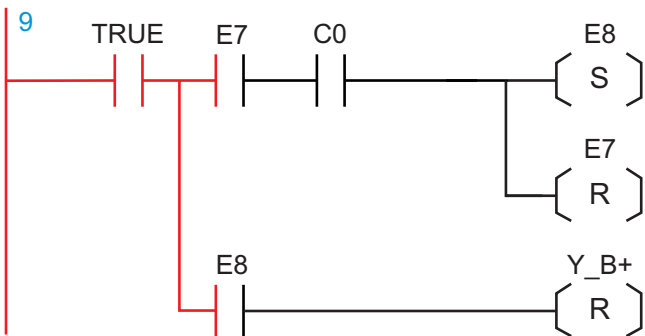
7. Al llegar al extremo opuesto A2, se inicia la etapa 6, se anula E5, y se excita B+ para bajar el conjunto.



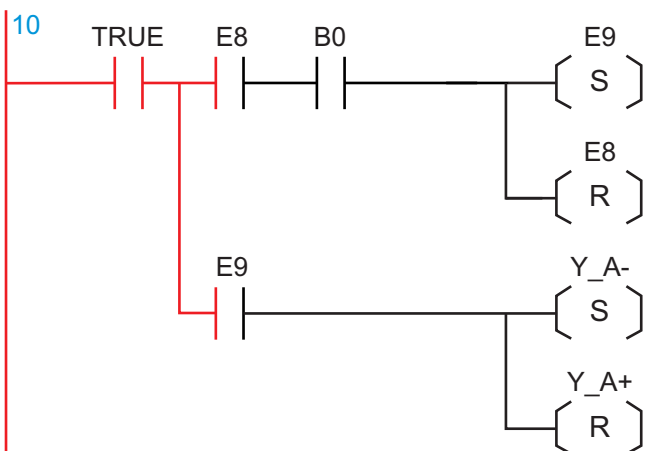
8. Al llegar al extremo inferior B1, se inicia la etapa 7, se anula E6, y C+.



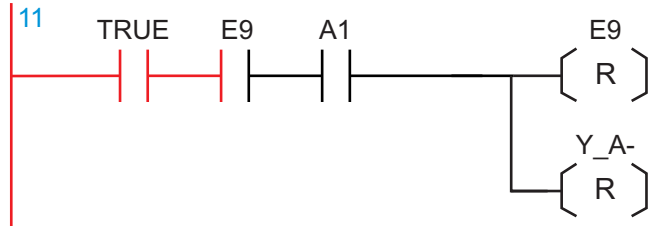
9. La pinza libera el tornillo, se abre completamente C (C0=1), y se activa la etapa 8, que anula a B+, para que suba el conjunto.



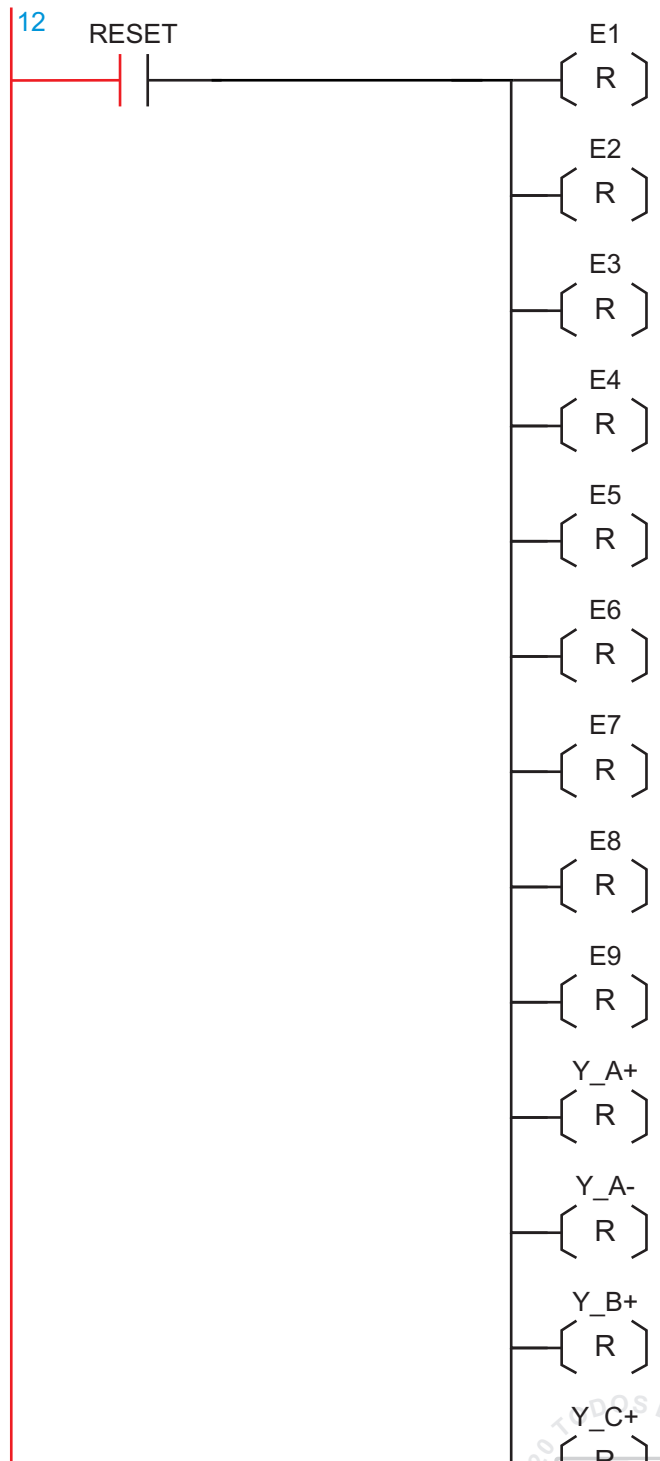
10. El conjunto se eleva hasta B0=1. Se activa la etapa 9, que realiza dos acciones; por un lado anula Y_A+, y por otro conecta A-, para desplazar el conjunto sentido inverso.



11. Cuando el conjunto se desplaza hasta A1, se anula el programa, reseteando la etapa en curso, y la electroválvula A-. El sistema está en condiciones iniciales.



12. La variable RESET, anula todas las etapas y las electroválvulas posibles.



El mismo ejemplo, programado en GRAFCET

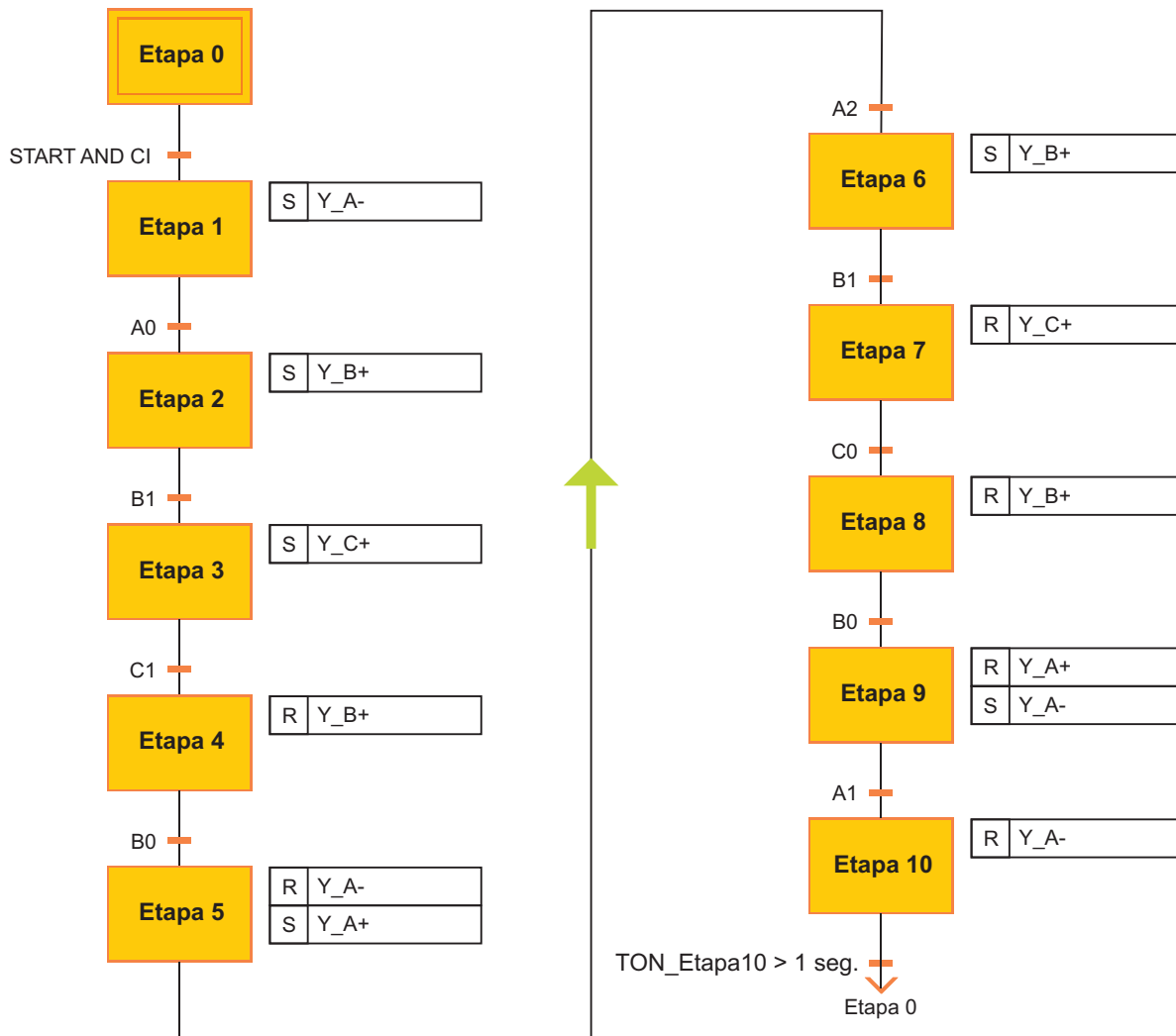
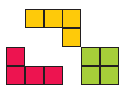


Figura 5.33. Programa del ejercicio “Desplazamiento de objetos por circuito neumático” en GRAFCET.



Ejercicios

4. Con el programa que dispongas y que permita programar tanto en lenguaje Ladder como en GRAFCET:

Realiza ambas propuestas del ejercicio propuesto para el ejemplo “Desplazamiento de objetos por circuito neumático”, de una programación en GRAFCET y GRAFCET con funciones RS.

Para el caso de GRAFCET, emplea una subrutina para generar las condiciones iniciales.



GRAF CET



GRAF CET
RS



Secuencia
neumática



GRAF CET



GRAF CET
RS



SECUENCIA



Fondo Gráfico

5. Realiza en ambos lenguajes de programación (GRAF CET y GRAFCET RS) un programa que sea compatible con la puesta en marcha y parada de un motor con protecciones (guardamotor).



GRAF CET



GRAF CET
RS

7 Stop en una secuencia GRAF CET 

Cuando se inicia un programa en GRAF CET, suele estar precedida —además de la orden de marcha—, de una variable que acoge las condiciones iniciales de una secuencia al completo, como se ha visto anteriormente.

Posteriormente en el siguiente ciclo, si un sensor indica que no existe producto suficiente, el proceso no continuará, pero este hecho estará dentro de la “normalidad” del proceso.

El problema surge cuando se desea terminar un proceso secuencial en GRAF CET, en cualquier punto del programa, actuando sobre una variable Stop, o a causa de un proceso de emergencia, independientemente del resultado técnico (botellas a medio llenar, temperaturas críticas, adhesivos, líquidos, etc.), el programa será capaz de permitir esta acción. A continuación se exponen varios casos para poder hacer este supuesto.

7.1. El programa no permite parada general del GRAF CET

En este primer caso, el programa no permite detener-reanudar el proceso secuencial con herramientas específicas. Para lograrlo, en cada etapa, se colocará una rama alternativa con una variable Stop, que deriva a una parte concreta de la secuencia, por ejemplo a la etapa inicial.

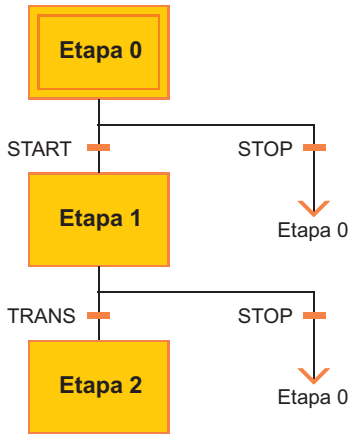
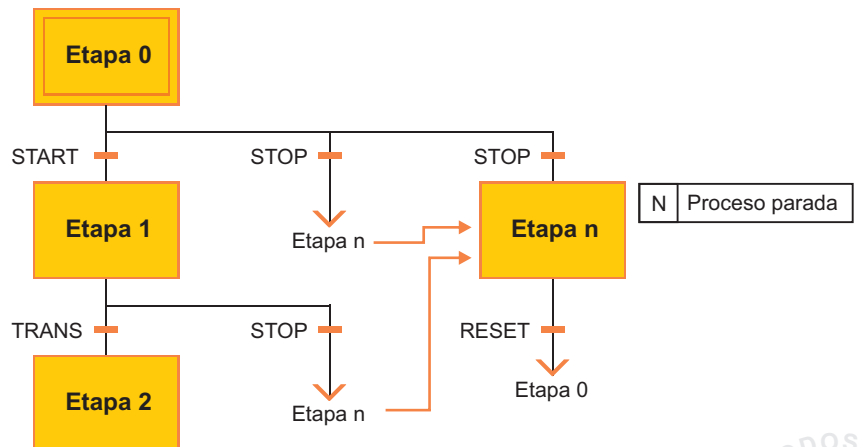



Figura 5.34. En cada etapa, se colocará una rama alternativa para detener la secuencia.



Esto permite realizar programas más ajustados; por ejemplo, cuando es actuado Stop, la rama alternativa lleva la secuencia a una etapa en concreto (n). Esta etapa significa avisos acústico-luminosos, la activación de procesos secundarios de paradas controladas, etc. Para retomar el ciclo, se actuará en otra variable tipo Reset.

Figura 5.35. La actuación sobre Stop, da como resultado una etapa de acciones independientes.



 STOP y RESET

Son dos órdenes distintas dentro de un proceso secuencial programado en GRAF CET.

Stop, detiene el GRAF CET en un momento determinado del proceso secuencial, mientras que Reset se va a utilizar para la reposición de valores a un estado concreto, por ejemplo, el establecimiento de posiciones para generar condiciones iniciales, reseteo de contadores y tiempos, etc.

En lo que respecta a los sensores, Stop va a utilizar una cámara cerrada, mientras que Reset, lo hará con contacto abierto.

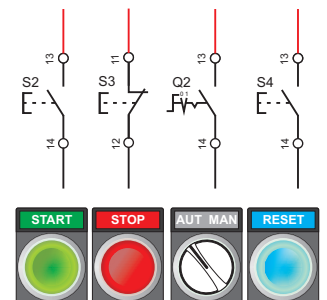


Figura 5.36. Botones típicos de un sistema automático.

7.2. El programa permite parada general del GRAFCET

El programa en GRAFCET, se suele realizar en una subrutina. En este caso, la actuación sobre el programa en sí, se hace sobre la llamada al programa secundario.

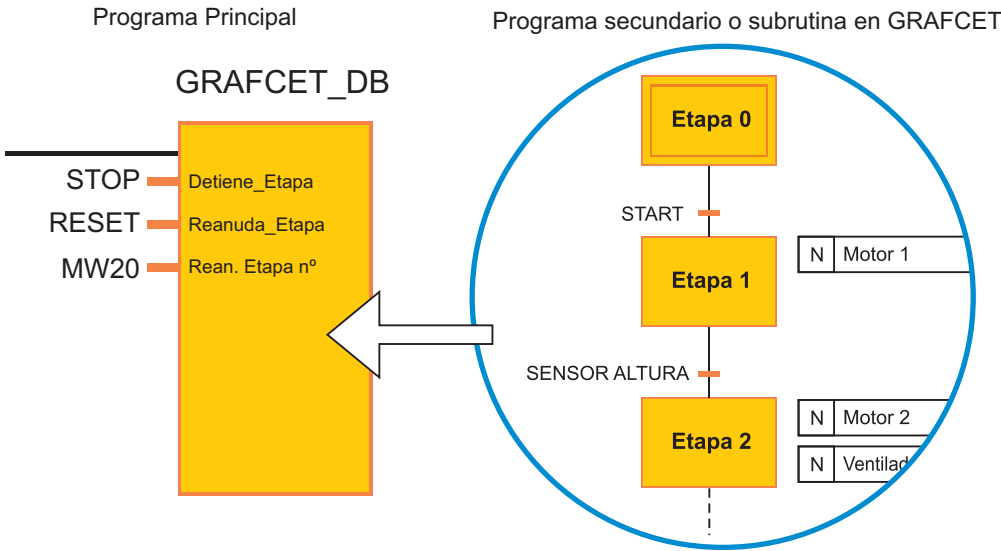


Figura 5.37. El GRAFCET se controla desde un bloque de llamada.

Este procedimiento va a permitir actuar sobre el programa en GRAFCET de manera indirecta, con acciones tales como Detener etapa, Reanudar etapa, Reanudar por etapa n°, etc.

7.3. El programa permite actuar dentro de cada etapa de manera independiente

El procedimiento es programar dentro de cada etapa una subrutina que permita, por ejemplo, la parada total o de ciertas partes del programa. Tiene el inconveniente de que se generan tantos programas de parada como etapas existan.

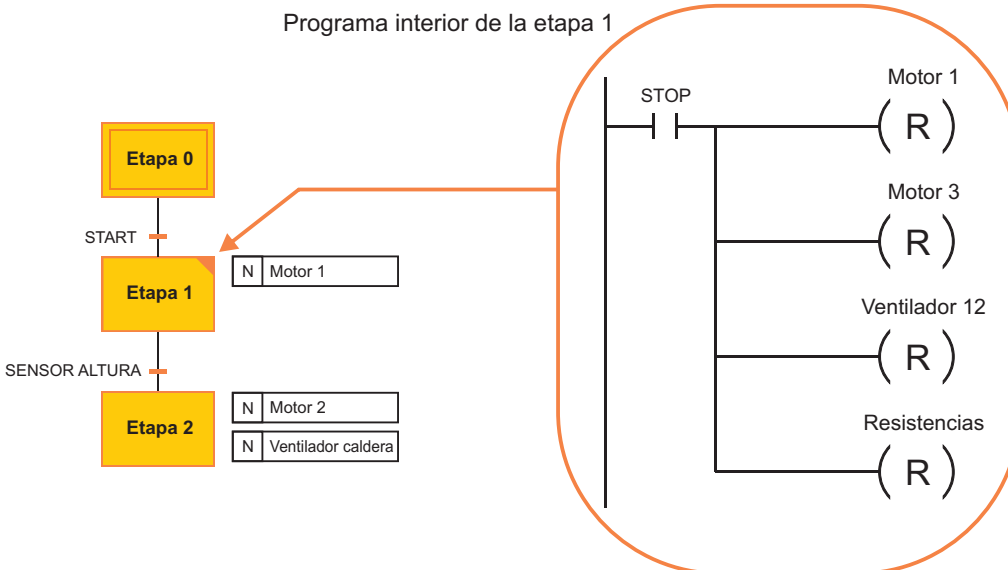
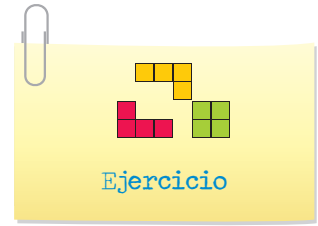


Figura 5.38. El GRAFCET se controla desde un bloque de llamada.



6. Con el programa que dispongas y que permita programar en lenguaje GRAFCET.

Diseña un ejercicio, donde aparezca un pulsador de START, otro de STOP, y un tercero de RESET, además de las variables que consideres oportunas, para el correcto funcionamiento.



Ejercicio 7. Trazabilidad

Una línea de montaje de juguetes tipo robot será capaz de reconocer si los diferentes procesos de montaje se realizan correctamente o con errores. El seguimiento en el proceso se conoce como trazabilidad.

1. Al presionar I1 START, se pone en marcha la variable luminoso Q1 SISTEMA. Si no funciona esta salida, no funcionará nada. La variable Q1 SISTEMA, se detiene por I2 STOP.
2. Al presionar I1 START, se pone en marcha el motor de la cinta transportadora a través de Q2.
3. La cinta alberga los pies del robot, que se debe completar con el cuerpo, la cabeza y los brazos.

Cuando los pies del robot llegan a la altura del sensor I3 SENSOR PROCESO CUERPO, el motor de la cinta se detiene Q2=0, se conecta la salida Q3 PROCESO CUERPO.

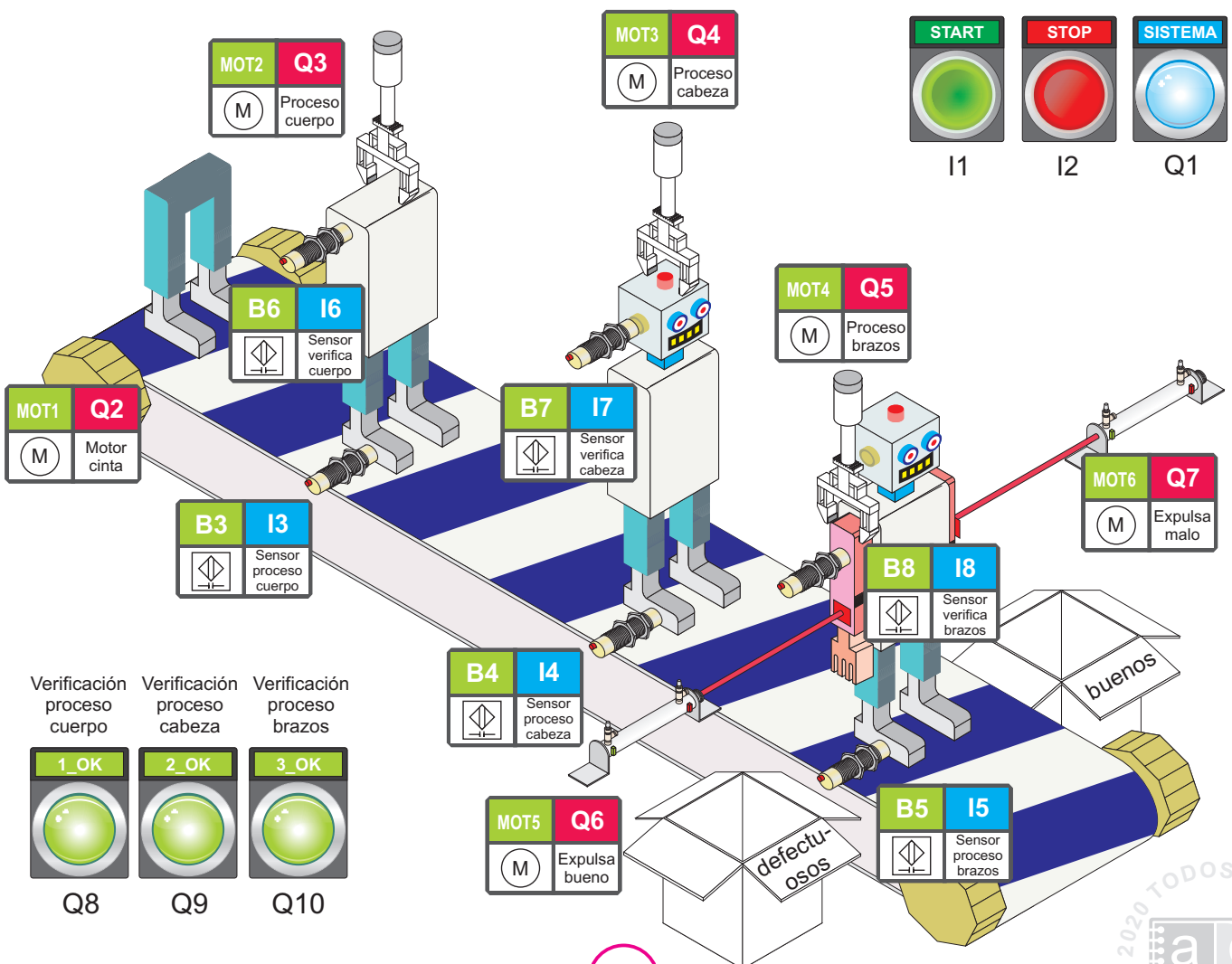
La salida Q3 representa todo el procedimiento para colocar el cuerpo del robot sobre las piernas.

Un temporizador controla el tiempo del proceso para colocar el cuerpo. Cuando este tiempo concluye, se pone en marcha de nuevo la cinta transportadora Q2 para buscar el proceso de colocar la cabeza del muñeco.

4. Por lo tanto, cuando este proceso termina, pueden ocurrir dos situaciones; que el cuerpo esté bien colocado o que no lo esté, a causa de algún defecto de la pieza o simplemente porque no se colocó bien.

La entrada I6 SENSOR VERIFICA CUERPO, se activará si la pieza está bien colocada cuando Q3=0, es decir, cuando termina el proceso Q3, si previamente se ha activado I6, se encenderá la salida Q8 que representa un luminoso indicando que el proceso de colocación del cuerpo fue satisfactoria.

↓ Figura 5.39. Gráfico del ejercicio propuesto.



5. Un temporizador activa la cinta, que se pone de nuevo en marcha hacia el proceso de colocación de la cabeza.

6. Cuando el robot llega a la altura del sensor I4 SENSOR PROCESO CABEZA, el motor de la cinta se detiene $Q2=0$; se conecta la salida Q4 PROCESO CABEZA. La salida Q4 representa todo el procedimiento para colocar la cabeza del robot sobre el cuerpo del mismo.

Un temporizador controla el tiempo del proceso para colocar la cabeza. Cuando este tiempo concluye, se pone en marcha de nuevo la cinta transportadora Q2 para buscar el proceso de colocar los brazos del muñeco. Pero antes, se comprueba la verificación de la pieza.

7. Cuando este proceso termina, pueden ocurrir dos situaciones; que el cuerpo esté bien colocado o que no lo esté, a causa de algún defecto de la pieza o simplemente porque no se ubicó bien.

La entrada I7 SENSOR VERIFICA CABEZA, se activará si la pieza está bien colocada cuando $Q4=0$, es decir, cuando termina el proceso Q4, si previamente se ha activado I7, se encenderá la salida Q9 que representa un luminoso indicando que el proceso de colocación de la cabeza fue satisfactoria.

8. Un temporizador pone de nuevo en marcha la cinta hacia el proceso de colocación de los brazos.

9. Cuando el robot llega a la altura del sensor I5 SENSOR PROCESO BRAZOS, el motor de la cinta se detiene $Q2=0$, se conecta la salida Q5 PROCESO BRAZOS. La salida Q5 representa todo el procedimiento para colocar la cabeza del robot sobre el cuerpo del mismo.

Un temporizador controla el tiempo del proceso para colocar los brazos. El proceso de colocación de componentes ha terminado. Ahora resta comprobar que los brazos se han colocado correctamente.

10. Cuando este proceso termina, pueden ocurrir dos situaciones; que los brazos estén bien colocados o que no lo estén, a causa de algún defecto de la pieza o simplemente porque no se ajustaron bien.

La entrada I8 SENSOR VERIFICA BRAZOS, se activará si la pieza está bien colocada cuando $Q5=0$; cuando termina el proceso Q5, si previamente se ha activado I8, se encenderá la salida Q10 que representa un luminoso indicando que el proceso de colocación de los brazos fue satisfactorio.

11. Para finalizar el proceso, pueden ocurrir dos situaciones; que los tres sub-procesos se han efectuado satisfactoriamente, en ese caso, cuando termina Q5 se abre un tiempo de 2 segundos, en el cual, se activa la salida Q6 que es un cilindro que expulsa el muñeco a la caja de "buenos", durante un tiempo de 4 segundos.

12. Por el contrario, se inicia una programación para expulsar el muñeco a la caja de "malos" durante 1 segundo. El cilindro expulsor es activado por Q7.

13. En cualquier caso, cuando finaliza el proceso con "buenas" o "malas" se resetean las salidas Q8, Q9 y Q10, y el sistema se pone de nuevo en marcha para un nuevo muñeco.

Realiza un programa en GRAFCET, que sea capaz de dar respuesta a los requerimientos expuestos.

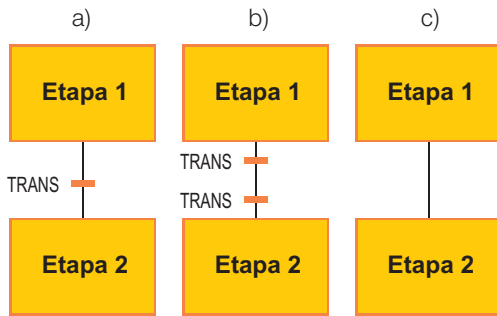
Variables	Denominación
I1	START
I2	STOP
I3	SENSOR PROCESO CUERPO
I4	SENSOR PROCESO CABEZA
I5	SENSOR PROCESO BRAZOS
I6	SENSOR VERIFICA CUERPO
I7	SENSOR VERIFICA CABEZA
I8	SENSOR VERIFICA BRAZOS
Q1	SISTEMA OK
Q2	MOTOR CINTA
Q3	PROCESO CUERPO
Q4	PROCESO CABEZA
Q5	PROCESO BRAZOS
Q6	CILINDRO EXPULSA BUENO
Q7	CILINDRO EXPULSA MALO
Q8	VERIFICACIÓN CUERPO OK
Q9	VERIFICACIÓN CABEZA OK
Q10	VERIFICACIÓN BRAZOS OK

Tabla 5.4. Variables propuestas para el ejercicio.

1. GRAFCET, significa:

- a) Gráfico de programación.
- b) Gráfico de bloques de funciones.
- c) Gráfico de contactos de programación.
- d) Gráfico funcional de control etapa-transición.

2. Dentro del GRAFCET, elige la composición correcta.



3. La condición que habilita la siguiente etapa se llama:

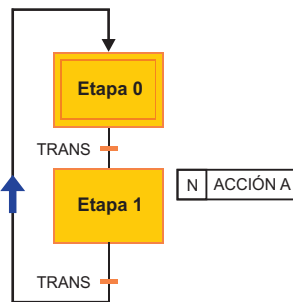
- a) Salto.
- b) Transición.
- c) Acción.
- d) Subrutina.

4. El indicador de una acción de GRAFCET: "La acción se ejecuta mientras la etapa está activa", corresponde a la letra:

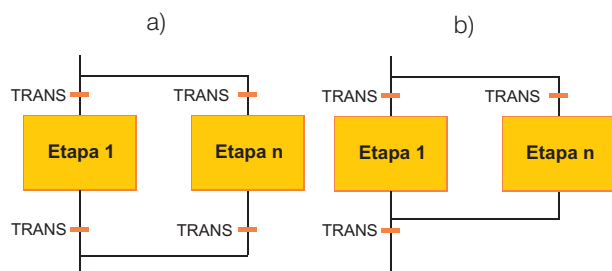
- a) S.
- b) R.
- c) N.
- d) DS.

5. La operación del gráfico es:

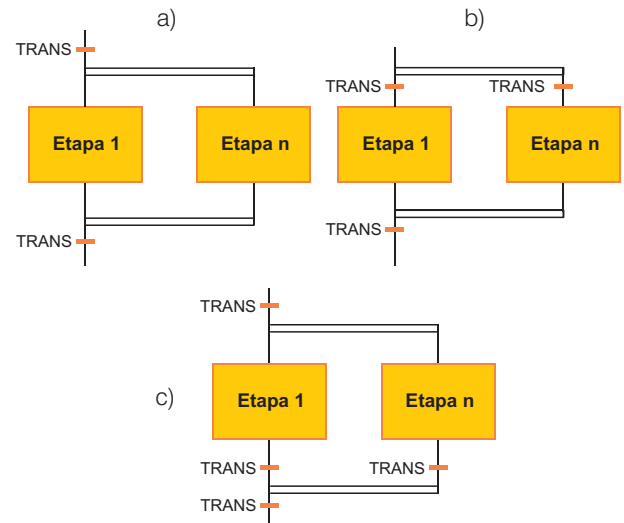
- a) Rama alternativa.
- b) Rama paralela.
- c) Salto condicional de etapa.
- d) Ninguna de las anteriores.



6. Elige la operación permitida.



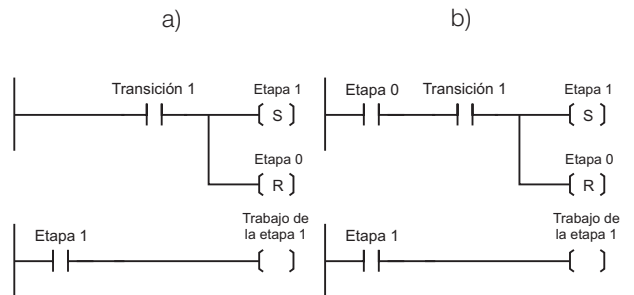
7. Elige la operación permitida.



8. Los requisitos previos, que debe presentar la instalación automática, antes de la puesta en marcha del proceso de manufactura, son:

- a) Condiciones de funcionamiento.
- b) Condiciones del programador.
- c) Condiciones iniciales.
- d) Condiciones del dueño de la fábrica.

9. Programación en GRAFCET con funciones SET-RESET. Elige la composición correcta.



10. En GRAFCET, la función STOP:

- a) Detiene el GRAFCET en un momento determinado.
- b) Repone los valores a un estado en concreto.
- c) Sirve para reiniciar el funcionamiento.
- d) Detiene la corriente de la máquina.

11. En GRAFCET, la función RESET:

- a) Detiene el GRAFCET en un momento determinado.
- b) Repone los valores a un estado en concreto.
- c) Sirve para reiniciar el funcionamiento.
- d) Detiene la corriente de la máquina.

12. Verdadero o falso. El seguimiento de un producto durante su proceso de fabricación es trazabilidad.

- a) Verdadero.
- b) Falso.

