

Objetivo de aprendizaje

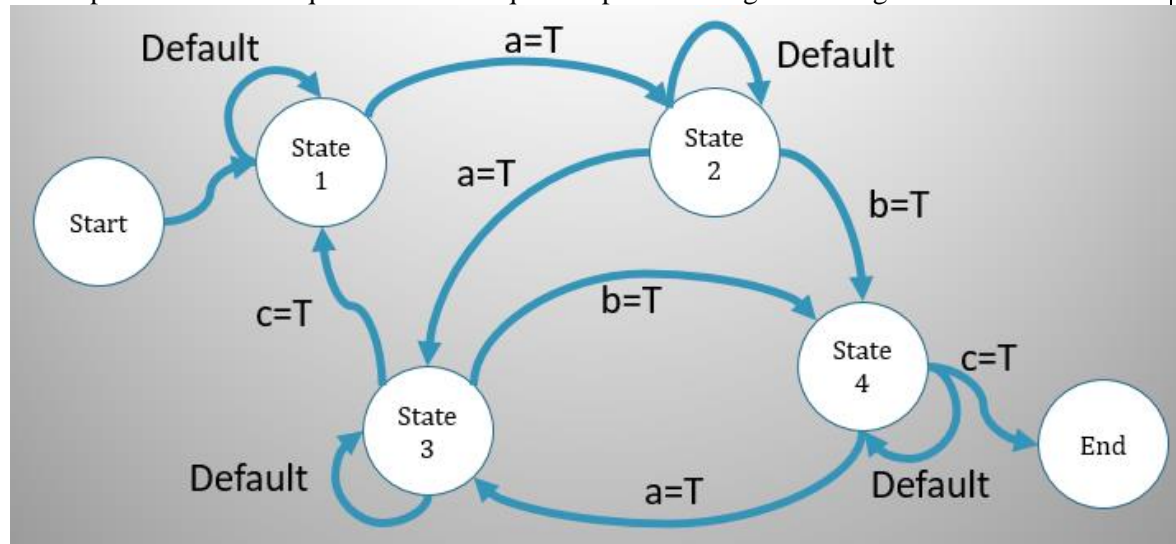
- Implementar una máquina de estados en Labview.

Resultados de aprendizaje

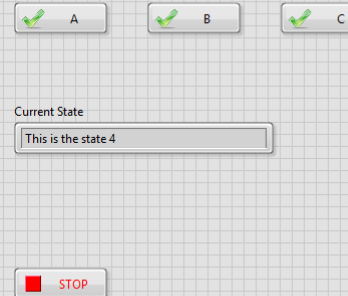
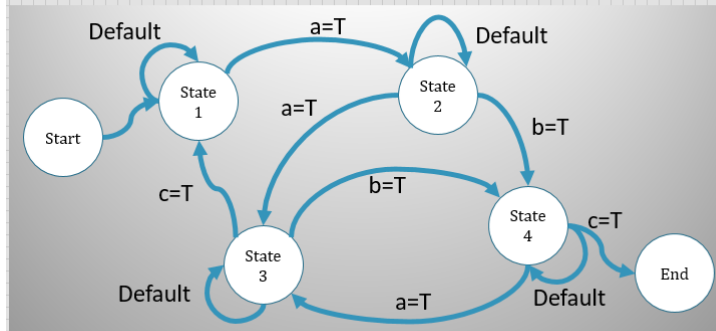
- Desarrollar una aplicación que contenga una máquina de estados.
- Comprender las partes y componentes de una máquina de estados.

Ejercicio de programación

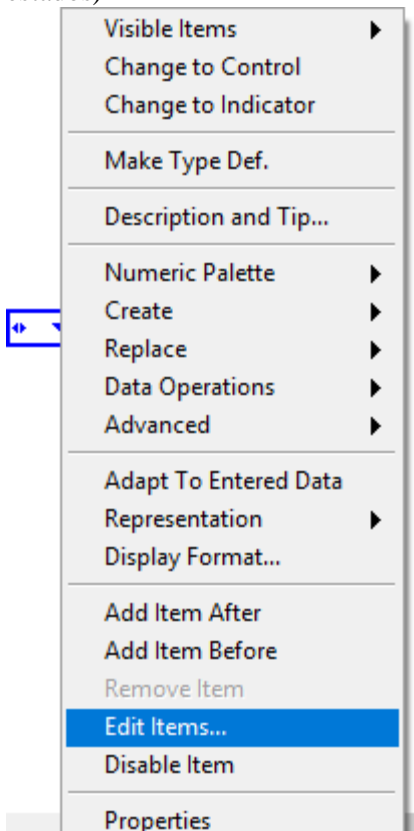
1. Implementar una máquina de estados que cumpla con el siguiente diagrama de estados:



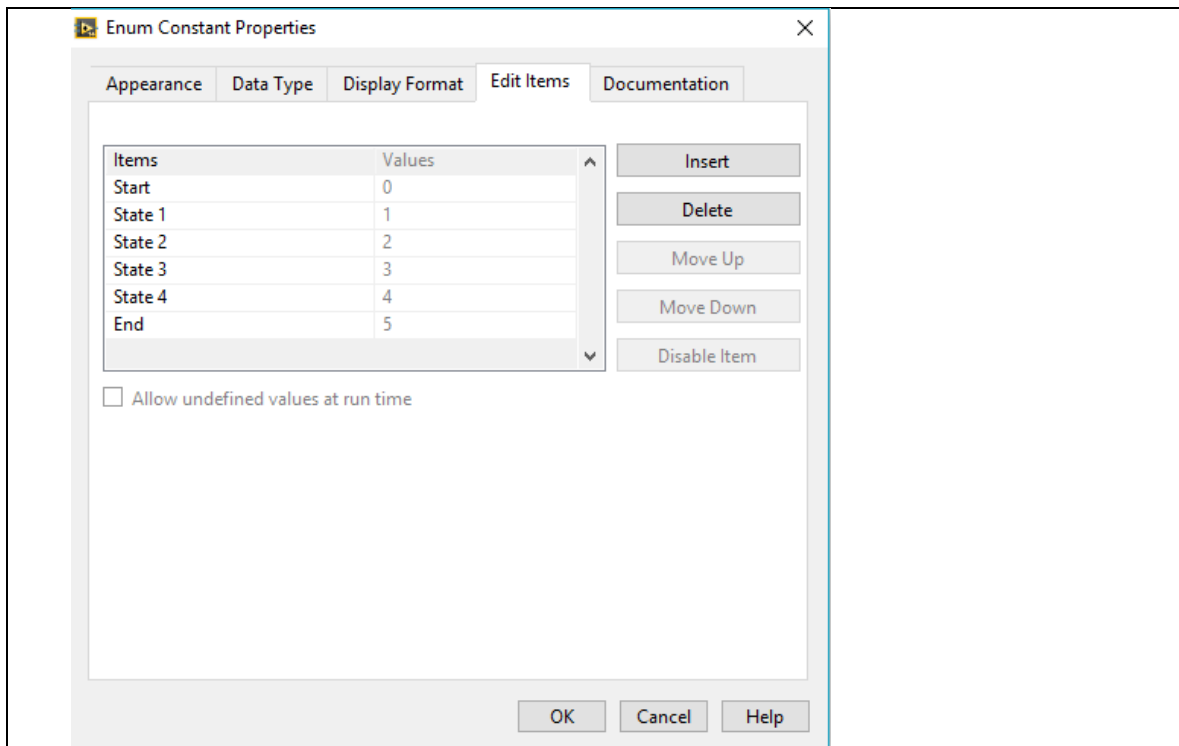
1. Abre el vi del ejercicio llamado stateMachines.vi dentro de la carpeta de recursos de la clase.
2. Agrega tres OK Button Silver y llámalos A, B y C respectivamente, configura su comportamiento en mechanical action, dejándolos con la opción Latch When Pressed. Estos botones serán usados para crear las condiciones de transición del diagrama de estados.
3. Agrega un String Indicator Silver y llámalo Current State, en él se observará el mensaje string que informe el estado en el que se encuentra la máquina.
4. Al finalizar estos pasos deberías tener un front panel luciendo similar a la siguiente imagen.



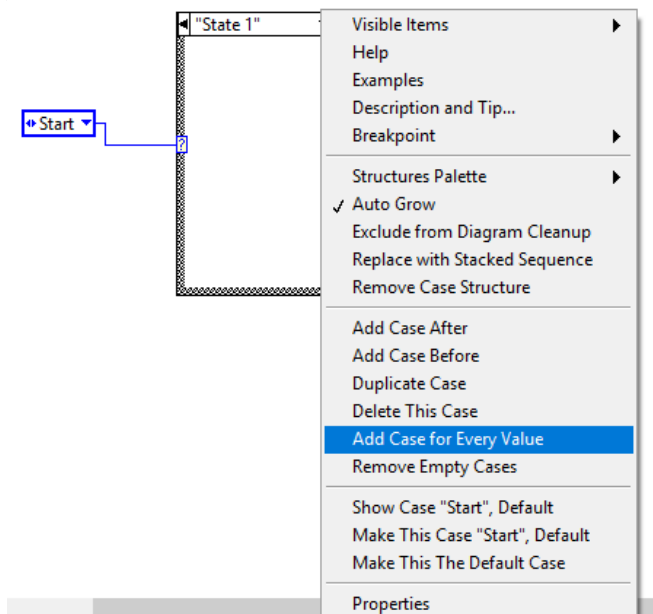
5. Ve al block diagram, busca y agrega con el quick drop un **enum constant**>>click derecho propiedades>>edit items. (El enum constant es el encargado de guardar los nombres de los estados)



6. En edit items, agrega con insert los estados de la siguiente forma, puedes cambiar el orden con los botones Move up y Move down:



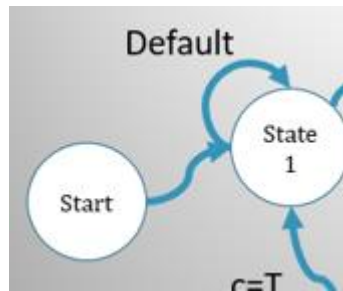
7. Conecta el enum constant al case selector y luego selecciona en el borde del case structure click derecho>>Add case for every value mira la imagen de ayuda siguiente (Esto crea un caso para cada elemento del constant enum, que tiene los casos)... haz creado los estados.



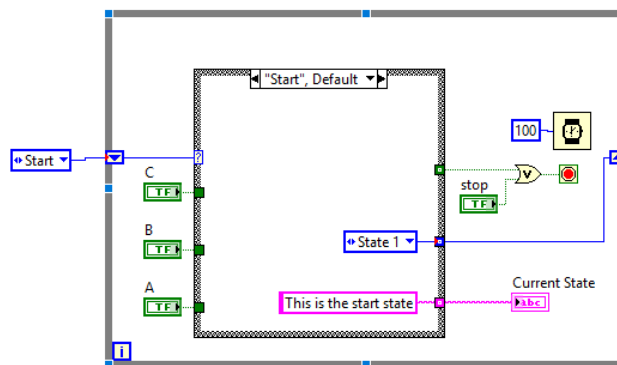
8. Recuerda de acuerdo a los tipos de transiciones las opciones que tienes de la clase de teoría.

Posiciónate en el estado llamado Start, ahí deberás crear el código correspondiente a dicho estado.

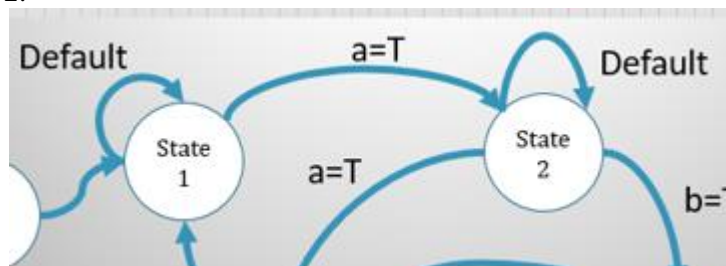
Observa con detenimiento que la transición del estado “Start” al “State 1” es directa y no requiere ninguna validación ni comprobación.



Por esta razón el código es muy simple, simplemente agrega una copia del system enum que almacena los estados y selecciona el State 1, al igual agrega un string constant y conéctalos tal como se muestra en la siguiente imagen.



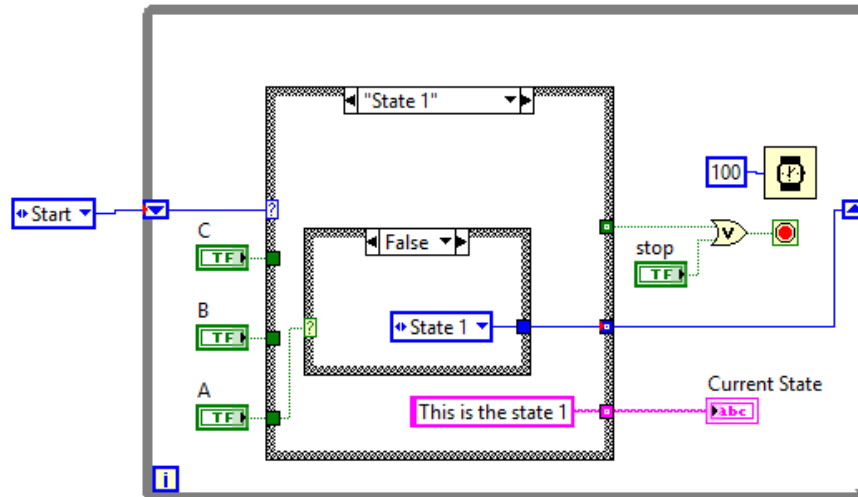
- Para el Estado 1, observa que existe un default que es permanecer en el estado 1 y con $a=True$, pasar al estado 2.



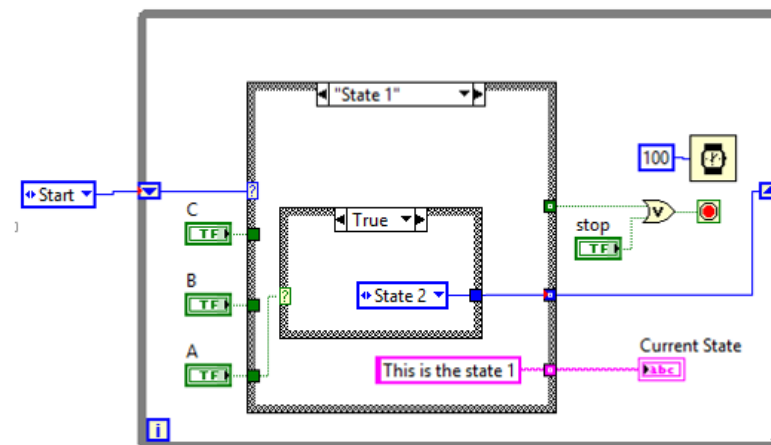
Por esta razón podrías crear en el Estado 1 un case sequence con las dos opciones teniendo en el case selector el botón A, en el caso verdadero deberías agregar un enum contact seleccionado a State 2 y en el False dejar un Enum constant con el State 1, crea y conecta el string constant para este estado.

Observa las siguientes dos imágenes, en ellas se ven como quedaría para el caso A=True y A=False

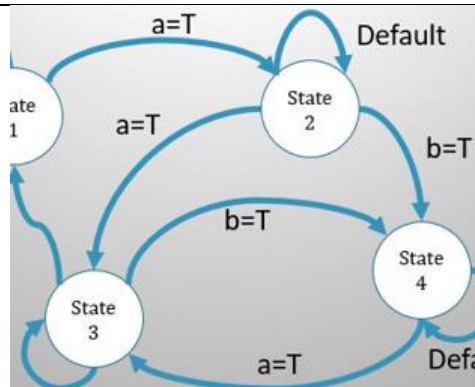
Caso en el que Boton A=False



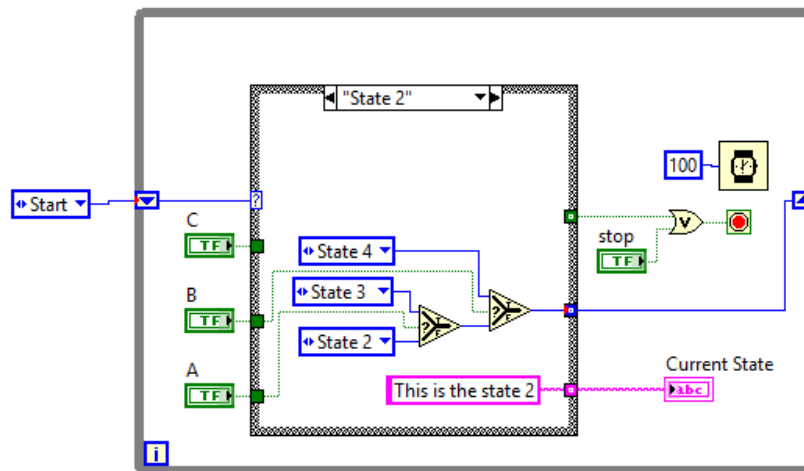
Caso en que A=TRUE



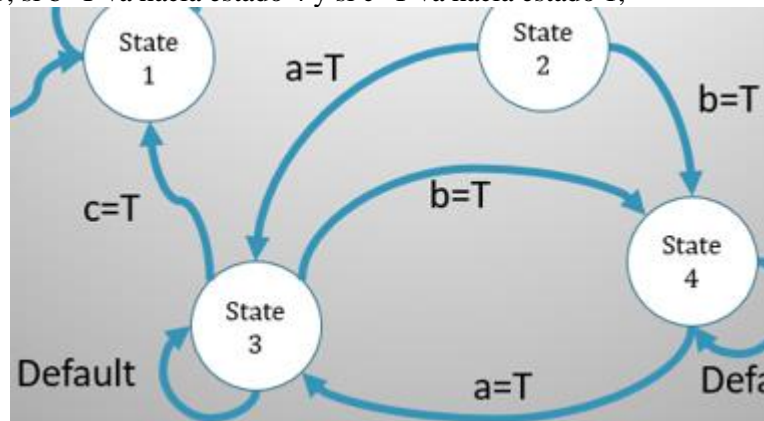
10. Ahora es turno de hacer el código de transición para el State 2, de acuerdo al diagrama de estados, observa que si $A=T$ pasa a Estado 3 y si $b=T$ pasa a estado 4, si no pasa ninguno de esas dos condiciones debe permanecer en el estado 2.



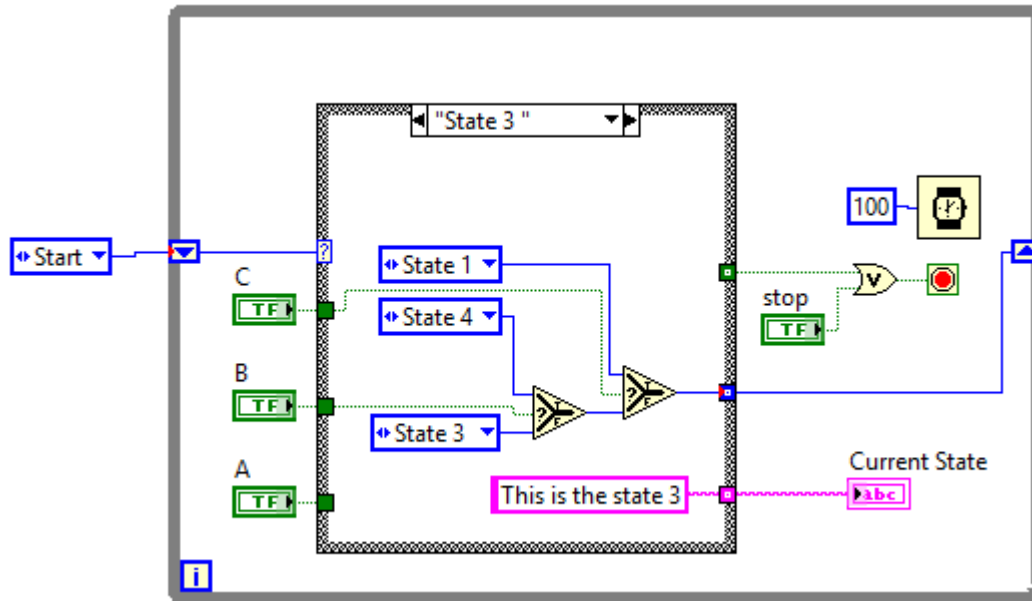
Utiliza funciones select de acuerdo a la siguiente imagen, verifica que está de acuerdo a lo requerido, si $a=F$ o $b=F$, el estado que va hacia el registro es State 2 (permaneciendo en el estado por default), si $a=T$ la salida sería State 3 y si $b=T$ la salida sería State 4; crea y conecta el string constant de este estado.



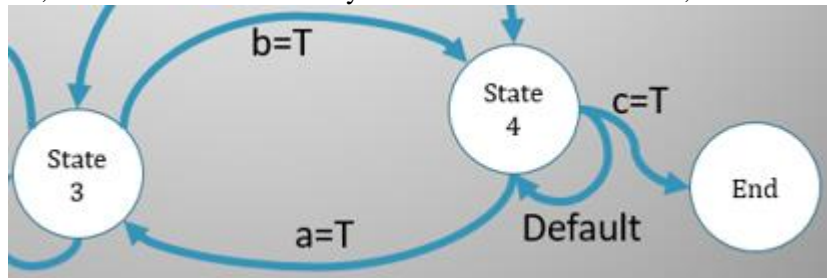
11. En el estado 3, si $b=T$ va hacia estado 4 y si $c=T$ va hacia estado 1,



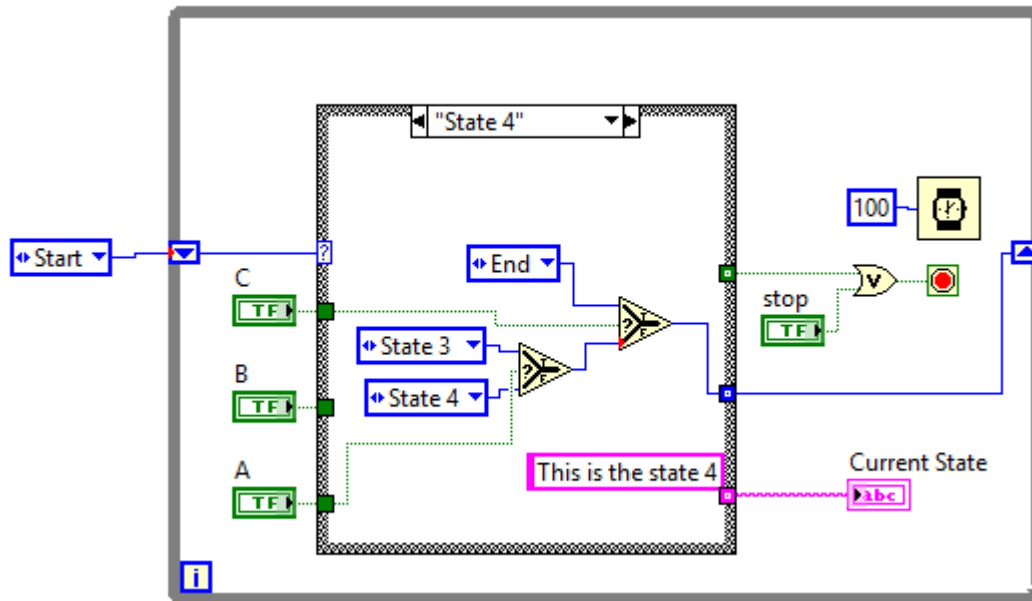
Crea el string constant de este estado y cablea los elementos para lograr un programa como el mostrado en la siguiente imagen.



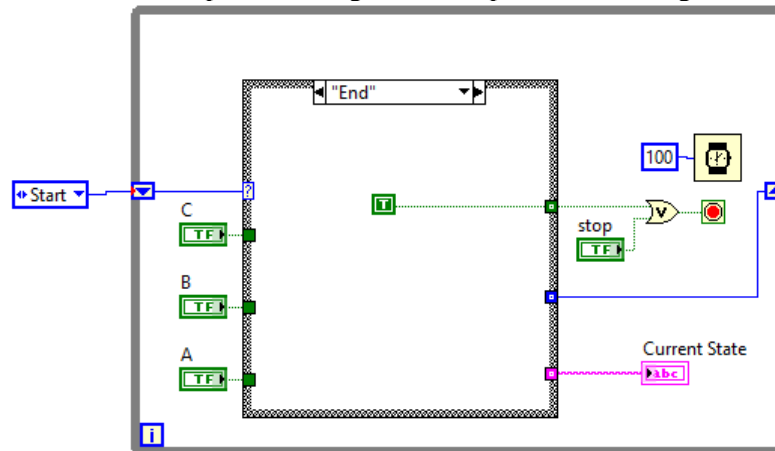
12. En el estado 4, si a=T va hacia estado 3 y si c=T va hacia estado End,



Agrega el string constant de este estado y crea un programa como el mostrado en la siguiente imagen.



13. Por último el estado End, este Estado también detiene la ejecución de la maquina así como lo haría el botón de stop de la máquina, es muy simple, en este estado no hay más transiciones y solo requieres detener el vi, replica el código como se presenta en la siguiente imagen.



14. Ahora valida que tu código cumpla con las secuencias que te da el diagrama de estados propuesto... el código de estado aquí fue muy sencillo, simplemente usar un string constant con el nombre del estado en el que se encuentra la maquina pero en otros ejercicios podrías incluir dentro de cada estado, subVi, programas o lo que desees para tu ejercicio... así es como se implementa una máquina de estados en Labview.

Fin de la lección.