Ciclos Labview - Estructuras de control

Desarrollo de Aplicaciones

20/05/2016

Luis Iván Morett Arévalo

13300226

Contenido

[Ciclos While 3](#_Toc451502006)

[Ciclos Infinitos 4](#_Toc451502007)

[Túneles de Estructura 5](#_Toc451502008)

[Ciclos For 6](#_Toc451502009)

[Añadir Temporización a los Ciclos 7](#_Toc451502010)

[Función de Espera 7](#_Toc451502011)

[Estructuras de Casos 8](#_Toc451502012)

[Seleccionar un Caso 9](#_Toc451502013)

[Túneles de Entrada y Salida 10](#_Toc451502014)

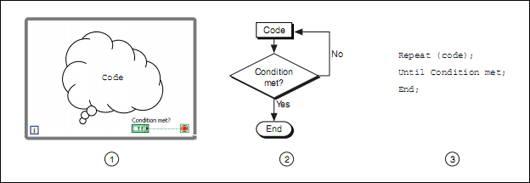
[Otras Estructuras 10](#_Toc451502015)

Estructuras de control

Las estructuras de ejecución contienen secciones de código gráfico y controlan cómo y dónde el código dentro se ejecuta. Las estructuras de ejecución más comunes son Ciclos While, Ciclos For y Estructuras de Casos los cuales puede usar para ejecutar la misma sección del código varias veces o para ejecutar una sección diferente del código basada en alguna condición.

# Ciclos While

Similar a un Ciclo Do o a un Ciclo Repeat-Until en lenguajes de programación basados en texto, un Ciclo While, que se muestra en la siguiente figura, ejecuta el código que contiene hasta que ocurre una condición.



El Ciclo While está ubicado en la paleta Structures. Seleccione el Ciclo While en la paleta y después use el cursor para arrastrar una selección rectangular alrededor de la sección del diagrama de bloques que desea repetir. Cuando suelte el botón del mouse, un borde del Ciclo While encierra la sección que seleccionó.

Añada objetos del diagrama de bloques al Ciclo While al arrastrarlos dentro del Ciclo While. El Ciclo While ejecuta el código que contiene hasta la terminal condicional, una terminal de entrada, recibe un valor Booleano específico.

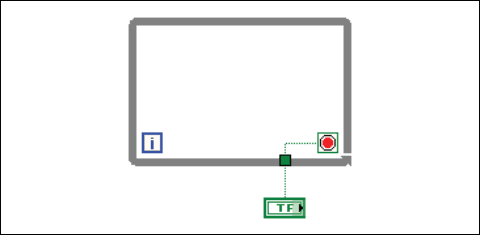
También puede realizar manejo básico de error usando la terminal condicional de un Ciclo While. Cuando cablea un cluster de error a la terminal condicional, solamente el valor True o False del parámetro de estatus del cluster de error pasa a la terminal. También, los elementos del menú Stop if True y Continue if True cambian a Stop if Error y Continue while Error.    
La terminal de iteración es una terminal de salida que contiene el número de iteraciones terminadas. La cantidad de iteraciones para el Ciclo While siempre comienza en cero.

**Nota:** El Ciclo While siempre se ejecuta por lo menos una vez.

## Ciclos Infinitos

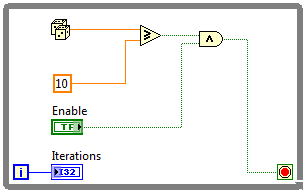
Los ciclos infinitos son un error común de programación que involucra un ciclo que nunca se detiene.

Si la terminal condicional es Stop if True, usted coloca la terminal de un control Booleano afuera de un Ciclo While. Si el control es FALSE cuando el ciclo comienza, provoca un ciclo infinito.



Cambiar el valor del control no detiene al ciclo infinito ya que el valor es de lectura una vez y eso sucede antes que el ciclo inicie. Para usar un control para detener un Ciclo While, debe colocar la terminal del control dentro del ciclo. Para detener un ciclo infinito, debe abortar e VI al dar clic en el botón Abort Execution de la barra de herramientas.

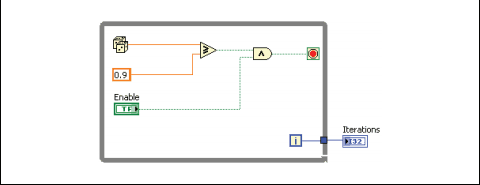
En la siguiente f**igura** el Ciclo While se ejecuta hasta que la salida de la función Random Number es mayor o igual que 10 y el control Enable es True. La función Añadir regresa True solamente su ambas entradas son True. De lo contrario, regresa False. Hay un ciclo infinito ya que la función Random nunca generara un valor igual o mayor que 10.



## Túneles de Estructura

Los túneles alimentan datos desde y hacia estructuras. El túnel aparece como un bloque sólido en el borde el Ciclo While. El bloque es el color del tipo de datos cableado al túnel. Los datos salen fuera de un ciclo después de que el ciclo termina. Cuando un túnel pasa datos a un ciclo, el ciclo se ejecuta solamente después que los datos llegan al túnel.

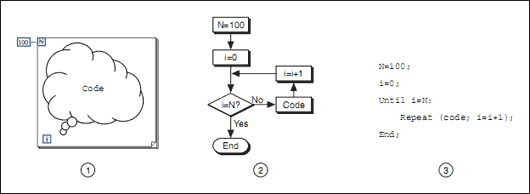
En la siguiente f**igura**  la terminal de iteración está conectada a un túnel. El valor en el túnel no pasa al indicador de iteraciones hasta que el Ciclo While termina de ejecutarse.



Solamente el último valor de la terminal de iteración se muestra en el indicador de iteraciones.

# Ciclos For

Un Ciclo For ejecuta un subdiagrama un número de veces establecido. **La Figura 5**muestra un Ciclo For en LabVIEW, un diagrama de flujo equivalente a la funcionalidad de Ciclo For y un código de ejemplo pseudo de la funcionalidad del Ciclo For.



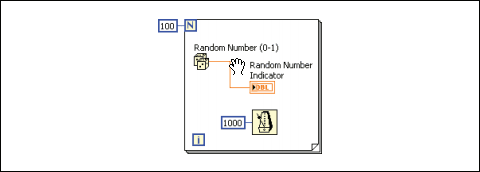
El Ciclo For está en la paleta Structures. También puede colocar un Ciclo While en el diagrama de bloques, dar clic con botón derecho en el Ciclo While y seleccionar Replace with For Loop en el menú de acceso directo para cambiar un Ciclo While Loop a Ciclo For. La terminal de conteo es una terminal de entrada cuyos valores indican cuantas veces se repite el subdiagrama.

La terminal de iteración es una terminal de salida que contiene el número de iteraciones terminadas.

La cantidad de iteraciones para el Ciclo For siempre comienza en cero.

El Ciclo For difiere del Ciclo While en que el Ciclo For ejecuta un número de veces establecido. Un Ciclo While detiene la ejecución solamente si existe el valor en la terminal condicional.

El Ciclo For en la siguiente f**igura** genera un número aleatorio cada segundo por 100 segundos y muestra los números aleatorios en un indicador numérico.



## Añadir Temporización a los Ciclos

Cuando un ciclo termina de ejecutar una iteración, inmediatamente comienza la próxima iteración, a menos que alcance una condición de paro. A menudo se necesita controlar la frecuencia y la temporización de la iteración Por ejemplo, si está adquiriendo datos y desea adquirir los datos una vez cada 10 segundos, necesita una manera de temporizar las iteraciones del ciclo para que ocurran una vez cada 10 segundos. Aún si no necesita que la ejecución ocurra a una cierta frecuencia, necesita proporcionar al procesador el tiempo para completar otras tareas, como responder a la interfaz de usuario.

### Función de Espera

Coloque una función de Espera dentro del ciclo para permitir que un VI se duerma por un cierto tiempo. Esto permite que su procesador maneje otras tareas durante el tiempo de espera. Las funciones de espera utilizan el reloj de milisegundos del sistema operativo.

http://www.ni.com/images/gettingstarted/neutral/lv_exe_12.gif  
La función de Espera (ms) espera hasta que el contador de milisegundos cuenta una cantidad igual a la entrada que usted especificó. Esta función garantiza que la razón de ejecución del ciclo sea por lo menos la cantidad de la entrada que usted especificó.

# Estructuras de Casos

http://www.ni.com/images/gettingstarted/neutral/lv_exe_8.gif  
Una estructura de Casos tiene dos o más subdiagramas o casos.

Solamente un subdiagrama es visible a la vez y la estructura ejecuta solamente un caso a la vez. Un valor de entrada determina cual subdiagrama se ejecuta. La estructura de Caso es similar a las instrucciones del interruptor o las instrucciones si...después... en lenguajes de programación basados en texto.

http://www.ni.com/images/gettingstarted/neutral/lv_exe_7.gif  
La etiqueta del selector de caso en la parte superior de la estructura de Caso contiene el nombre del valor del selector que corresponde al caso en el centro y a las flechas de incremento y reducción a cada lado.

Haga clic en las flechas de incremento y reducción para desplazarse en los casos disponibles. También puede hacer clic en la flecha hacia abajo a lado del nombre del caso y seleccionar un caso en el menú desplegable.

Cablee un valor de entrada o selector a la terminal del selector para determinar qué caso se ejecuta.

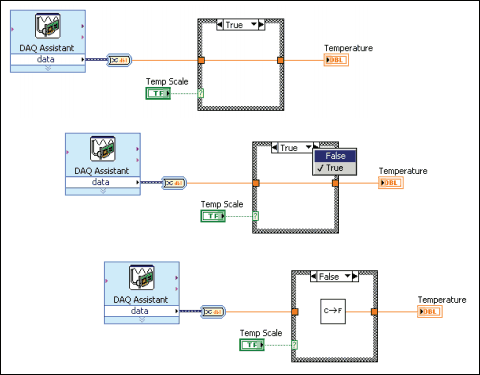
Debe cablear un entero, valor Booleano, secuencia o valor de tipo enumerado a la terminal del selector. Puede colocar la terminal del selector en cualquier parte del borde izquierdo de la estructura de Casos. Si el tipo de datos de la terminal del selector es Booleano, la estructura tiene un caso True o un caso False. Si la terminal del selector es un entero, secuencia o valor de tipo enumerado, la estructura tiene cualquier número de casos.

Si no especifica un caso de forma predeterminada para la estructura de Casos para manejar los valores fuera del rango, debe enlistar de forma explícita cada valor de entrada. Por ejemplo, si el selector es un entero y usted especifica los casos para 1, 2 y 3, debe especificar un caso de forma predeterminada a ejecutar si el valor de entrada es 4 o cualquier otro valor entero no especificado.

Haga clic con botón derecho en el borde de la estructura para añadir, duplicar, eliminar o reorganizar casos y para seleccionar un caso predeterminado.

## Seleccionar un Caso

La siguiente figuramuestra un VI que utiliza una estructura de Caso para ejecutar código diferente dependiendo si el usuario selecciona °C o °F para las unidades de temperatura. El diagrama de bloques de arriba muestra el caso True en primer plano. En el diagrama de bloques del centro, el caso Falso es seleccionado. Para seleccionar un caso, proporcione el valor en el identificador del selector de casos o use la herramienta de Etiquetado para editar los valores. Después que seleccionó otro caso, ese caso se ve en el diagrama de bloques, como se muestra en la parte inferior del diagrama de bloques.



Si usted proporciona un valor de selector que no es del mismo tipo que el objeto cableado a la terminal del selector, el valor aparece en rojo. Esto indica que el VI no se ejecutará hasta que elimine o edite el valor. También, por el posible redondeo de error inherente a la aritmética de punto flotante, no puede usar los números de punto flotante como valores de selector de casos. Si cablea un valor de punto flotante al caso, LabVIEW redondea el valor al entero más cercano. Si escribe un valor de punto flotante en la etiqueta del selector de casos, el valor aparece en rojo para indicar que debe eliminar o editar el valor antes que la estructura pueda ejecutarse.

# Túneles de Entrada y Salida

Puede crear múltiples túneles de entrada y salida para una estructura de Casos. Las entradas están disponibles para todos los casos, pero los casos no necesitan usar cada unidad. Sin embargo, debe definir un túnel de salida para cada caso.

Considere el siguiente ejemplo: una estructura de Casos en el diagrama de bloques tiene un túnel de salida, pero por lo menos en uno de los casos, no hay valor de salida cableado al túnel. Si ejecuta este caso, LabVIEW no sabe qué valor regresar a la salida. LabVIEW indica este error al dejar el centro del túnel en blanco. El caso no cableado no debe ser el caso que está visible actualmente en el diagrama de bloques.

Para corregir este error, muestre los casos que contienen el túnel de salida sin cablear y cablee una salida al túnel. También puede dar clic con botón derecho en el túnel de salida y seleccionar Use Default If Unwired en el menú para usar el valor predeterminado para el tipo de datos de túnel para todos los túneles no cableados. Cuando la salida está cableada en todos los casos, el túnel de salida está en color sólido.

Evite usar la opción Use Default If Unwired. Al usar esta opción no documenta el diagrama de bloques bien y puede confundir a otros programadores que utilizan su código. La opción Use Default If Unwired también hace difícil la depuración del código. Si utiliza esta opción, tome en cuenta que el valor predeterminado usado es el valor predeterminado para el tipo de datos que es cableado al túnel. Por ejemplo, si el tipo de datos es Booleano, el valor predeterminado es FALSE.

| **Tipo de Datos** | **Valor Predeterminado** |
| --- | --- |
| Numérico | 0 |
| Booleano | FALSE |
| Cadena de caracteres | vacío (“”) |

# Otras Estructuras

LabVIEW tiene otros tipos más avanzados de estructuras de ejecución como estructuras de Eventos (usadas para manejar tareas dirigidas por interrupciones como interacción UI) y estructuras de Secuencia (usadas para forzar el orden de ejecución).