
Programación de eventos

Los patrones de diseño basados en eventos permiten crear aplicaciones más eficaces y flexibles. Los patrones de diseño basados en eventos usan la estructura Event para responder directamente a los eventos de usuario o de otro tipo. Esta lección describe programación orientada a eventos mediante la estructura Event y los patrones de diseño que usan esta estructura.

Temas

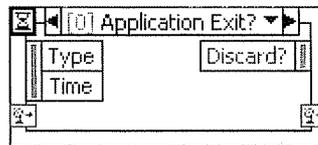
- A. Programación orientada a eventos
- B. Patrones de diseño basados en eventos

A. Programación orientada a eventos

En la lección 1, *Técnicas de diseño típicas*, aprendió cómo la programación orientada a eventos amplía el entorno de flujo de datos de LabVIEW para permitir la interacción del usuario con el panel frontal. También aprendió que los eventos generados programáticamente permiten implementar fácilmente arquitecturas avanzadas, como las máquinas de estados con colas.

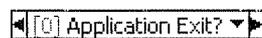
Componentes de la estructura Event

Use la estructura Event, mostrada a continuación, para controlar eventos en un VI.



La estructura Event funciona como una estructura Case con una función Wait on Notification integrada. La estructura Event puede tener varios casos, cada uno de los cuales es una rutina de control de eventos distinta. Puede configurar cada caso para controlar uno o más eventos, pero sólo uno de estos eventos puede ocurrir a la vez. Cuando se ejecuta la estructura Event, espera hasta que ocurra uno de los eventos configurados y después ejecuta el caso correspondiente a ese evento. La estructura Event completa su ejecución tras controlar exactamente un evento. No hace implícitamente un bucle para controlar varios eventos. Como una función Wait on Notification, la estructura Event puede superar su límite de tiempo mientras espera una notificación de evento. Cuando esto ocurre, se ejecuta un caso Timeout específico.

La etiqueta del selector de evento en la parte superior de la estructura Event, mostrada a continuación, indica qué eventos provocan la ejecución del caso mostrado actualmente.



Vea otros casos de eventos haciendo clic en la flecha abajo junto al nombre del caso y seleccionando otro caso del menú contextual.



El terminal Timeout de la esquina superior izquierda de la estructura Event, mostrada a la izquierda, especifica el número de milisegundos que hay que esperar antes de que un evento supere el tiempo límite.

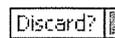
El valor predeterminado es -1 , que especifica esperar indefinidamente hasta que ocurra un evento. Si cablea un valor al terminal Timeout, debe proporcionar un caso Timeout.

El Event Data Node, mostrado a continuación, actúa de igual modo que la función Unbundle By Name.



Este nodo se acopla al borde interior izquierdo de cada caso de evento. El nodo identifica los datos que proporciona LabVIEW cuando ocurre un evento. Puede cambiar el tamaño de este nodo verticalmente para añadir más elementos de datos y puede configurar éstos en el nodo para acceder a cualquier elemento de datos de evento. El nodo proporciona varios datos en cada caso de la estructura Event en función de qué eventos configure para que controle ese caso. Si configura que un solo caso controle varios eventos, el Event Data Node proporciona sólo los datos de evento que son comunes para todos los eventos configurados para ese caso.

El Event Filter Node, mostrado a continuación, es similar al Event Data Node.



Este nodo se acopla al borde interior derecho de los casos de eventos de filtrado. El nodo identifica el subconjunto de datos disponibles en el Event Data Node que puede modificar el caso de evento. El nodo muestra varios datos en función de qué eventos configure para que controle ese caso. De forma predeterminada, estos elementos se encuentran en los elementos de datos correspondientes del Event Data Node. Si no cablea un valor a un dato de un Event Filter Node, ese dato permanecerá inalterado.

Consulte la sección *Eventos de notificación y de filtrado* de esta lección para obtener información adicional acerca de los eventos de filtrado.



Puede acceder a los terminales de eventos dinámicos, mostrados a la izquierda, haciendo clic con el botón derecho en la estructura Event y seleccionando **Show Dynamic Event Terminals** en el menú contextual.

Estos terminales sólo se usan para el registro de eventos dinámicos.

Consulte los temas *Dynamic Event Registration* y *Modifying Registration Dynamically* de la *Ayuda de LabVIEW* para obtener información adicional acerca del uso de estos terminales.



Nota Al igual que una estructura Case, la estructura Event admite túneles. Sin embargo, de forma predeterminada no tiene que cablear túneles de salida de la estructura Event en cada caso. Todos los túneles sin cablear usan el valor predeterminado para el tipo de dato de túnel. Haga clic con el botón derecho en un túnel y deseleccione **Use Default If Unwired** en el menú contextual para volver al comportamiento de la estructura Case predeterminada en el que debe cablear los túneles en todos los casos. También puede configurar los túneles para cablear los túneles de entrada y salida automáticamente en los casos sin cablear.

Consulte la *Ayuda de LabVIEW* para obtener información de los valores predeterminados para cada tipo de datos.

Eventos de notificación y de filtrado

Existen dos tipos de eventos de interfaz de usuario: notificación y filtrado.

Los eventos de notificación indican que ya ha ocurrido una acción del usuario, como cambiar el valor de un control. Use eventos de notificación para responder a un evento cuando haya ocurrido y LabVIEW lo haya procesado. Puede configurar cualquier número de estructuras Event para responder al mismo evento de notificación en un objeto concreto. Cuando ocurre el evento, LabVIEW envía una copia de ésta a cada estructura Event configurada para controlar el evento en paralelo.

Los eventos de filtrado le informan de que el usuario ha realizado una acción antes de que LabVIEW la procese, lo que permite personalizar el modo en que el programa responde a interacciones con la interfaz de usuario. Use eventos de filtrado para participar en el control del evento, posiblemente invalidando el comportamiento predeterminado del evento. En un caso de estructura Event de un evento de filtrado, puede validar o cambiar los datos del evento antes de que LabVIEW termine de procesarlo, o puede desechar el evento totalmente para evitar que el cambio afecte al VI. Por ejemplo, puede configurar una estructura Event para desechar el evento Panel Close?, que evitaría que el usuario cierre interactivamente el panel frontal del VI.

Los eventos de filtrado tienen nombres que terminan con un signo de interrogación, como Panel Close?, para ayudar a distinguirlos de eventos de notificación. La mayoría de los eventos de filtrado tienen un evento de notificación asociado con el mismo nombre, pero sin el interrogante, que genera LabVIEW tras el evento de filtrado si ningún caso de evento desechó el evento.

Por ejemplo, puede usar los eventos de filtrado Mouse Down? y Shortcut Menu Activation? para mostrar un menú contextual cuando haga clic con el botón izquierdo en un control. Para realizar esta acción, modifique los datos que devuelve el campo de datos de evento **Button** del evento de filtrado Mouse Down?. El valor del botón izquierdo del ratón es 1 y el valor del botón derecho del ratón es 2. Para ver el menú contextual cuando hace clic con el botón izquierdo en un control, cambie el campo de datos de evento **Button** a 2, para que LabVIEW considere el clic con el botón izquierdo igual que con el derecho. Consulte el VI Left-click Shortcut Menu del directorio labview\examples\general para ver un ejemplo de uso de eventos de filtrado.

Al igual que con los eventos de notificación, puede configurar cualquier número de estructuras Event para responder al mismo evento de filtrado en un objeto concreto. Sin embargo, LabVIEW envía eventos de filtrado secuencialmente a cada estructura Event configurada para el evento. El orden en que LabVIEW envía el evento a cada estructura Event depende de en qué orden se registraran los eventos. Cada estructura Event debe completar su caso de evento para el evento antes de que LabVIEW pueda notificar la siguiente estructura Event. Si una estructura Event cambia cualquiera de los datos de evento, LabVIEW pasa los datos cambiados a las siguientes estructuras Event de la cadena. Si una estructura Event de la cadena desecha el evento, LabVIEW no pasa éste a las estructuras Event que queden en la cadena. LabVIEW completa el procesamiento de la acción del usuario que activó el evento sólo cuando todas las estructuras Event configuradas controlen el evento sin desecharlo.



Nota National Instruments recomienda usar eventos de filtrado sólo cuando desee intervenir en el control de la acción del usuario, desechando el evento o modificando los datos del evento. Si sólo desea saber que el usuario realizó una acción concreta, use eventos de notificación.

Los casos de la estructura Event que controlan eventos de filtrado tienen un Event Filter Node. Puede cambiar los datos de evento cableando nuevos valores a estos terminales. Si no cablea un valor al dato del Event Filter Node, el valor predeterminado será el valor que devuelva el elemento correspondiente del Event Data Node. Puede desear completamente cualquier evento cableando un valor TRUE al terminal **Discard?**.



Nota Un solo caso de la estructura Event no puede controlar eventos de notificación y de filtrado. Un caso puede controlar varios eventos de notificación pero puede controlar varios eventos de filtrado sólo si los datos del evento son idénticos para todos los eventos.

Consulte la sección *Uso de eventos en LabVIEW* de esta lección para obtener información adicional acerca del registro de eventos.



Consejo En el cuadro de diálogo **Edit Events**, los eventos de notificación se indican con una flecha verde, mientras que los eventos de filtrado tienen una roja.

Uso de eventos en LabVIEW

LabVIEW puede generar numerosos eventos distintos. Para evitar generar eventos indeseados, use el registro de eventos para especificar qué eventos desea que le notifique LabVIEW. LabVIEW admite dos modelos de registro de eventos: estático y dinámico.

El registro estático permite especificar qué eventos del panel frontal de un VI desea controlar en cada caso de la estructura Event en el diagrama de bloques de ese VI. LabVIEW registra estos eventos automáticamente cuando se ejecuta el VI, por lo que la estructura Event empieza a esperar eventos en cuanto se ejecuta el VI. Cada evento se asocia con un control del panel frontal del VI, la ventana del panel frontal del VI en conjunto o la aplicación de LabVIEW. No puede configurar estáticamente una estructura Event para controlar eventos para el panel frontal de otro VI. La configuración es estática porque no puede cambiar en el tiempo de ejecución qué eventos controla la estructura Event.

El registro de eventos dinámico evita las limitaciones del registro estático integrando el registro de eventos con el VI Server, que permite usar referencias de aplicación, VI y control para especificar en el tiempo de ejecución los objetos para los que desea generar eventos. El registro dinámico ofrece más flexibilidad en el control de qué eventos genera LabVIEW y cuándo los genera. Sin embargo, el registro dinámico es más complejo que el estático porque requiere usar referencias de VI Server con funciones del diagrama de bloques para registrar y desregistrar eventos en lugar de controlar el registro automáticamente usando la información que configuró en la estructura Event.



Nota En general, LabVIEW genera eventos de interfaz de usuario sólo como resultado de la interacción directa del usuario con el panel frontal activo. LabVIEW no genera eventos, como Value Change, cuando usa variables compartidas, variables globales, variables locales, DataSocket, etc. Sin embargo, puede usar la propiedad Value (Signaling) para generar un evento Value Change programáticamente. En muchos casos puede usar eventos generados programáticamente en lugar de colas y notificadores.

Los datos de evento que proporciona un evento de LabVIEW siempre incluyen una marca de hora, una enumeración que indica qué evento ocurrió, así como una referencia de VI Server al objeto que activó el evento. La marca de hora es un contador de milisegundos que puede usar para calcular el tiempo transcurrido entre dos eventos o para determinar el orden en que sucedieron. La referencia al objeto que generó el evento se escribe estrictamente en la clase de VI Server de ese objeto. Los eventos se agrupan en clases según el tipo de objeto que genera el evento, como una aplicación, VI o control. Si un solo caso controla varios eventos para objetos de distintos tipos de VI Server, el tipo de referencia es la clase primaria común de todos los objetos. Por ejemplo, si configura un solo caso en la estructura Event

para controlar eventos en un control numérico y un control de rampa de color, el tipo de referencia de control del origen del evento es numérico porque los controles numérico y rampa de color están en la clase Numeric. Si se registra para el mismo evento en la clase VI y Control, LabVIEW genera el evento VI primero.



Nota Los clusters son los únicos objetos contenedores para los que puede generar eventos. LabVIEW genera eventos Control para clusters, antes de generar eventos para los objetos que contienen, salvo en el caso del evento Value Change. El evento Value Change genera el evento en un elemento del cluster, después en el cluster mismo. Si el caso de la estructura Event de un evento VI o de un evento Control en un objeto de un contenedor desecha el evento, LabVIEW no generará más eventos.

Cada estructura Event y función Register For Events del diagrama de bloques posee una cola que usa LabVIEW para almacenar eventos. Cuando ocurre un evento, LabVIEW coloca una copia suya en cada cola registrada para él. Una estructura Event controla todos los eventos de su cola y los eventos de las colas de cualquier función Register For Events que cableó a los terminales de eventos dinámicos de la estructura Event. LabVIEW usa estas colas para que los eventos se entreguen correctamente a cada estructura Event registrada en el orden en que ocurren los eventos.

De forma predeterminada, cuando un evento entra en una cola, LabVIEW bloquea el panel frontal que contiene el objeto que generó ese evento. LabVIEW mantiene bloqueado el panel frontal hasta que todas las estructuras Event terminen de procesar el evento. Mientras está bloqueado el panel frontal, LabVIEW no procesa la actividad del panel frontal sino que coloca las interacciones en un búfer y las controla cuando se desbloquea el panel frontal.

Por ejemplo, un usuario podría prever que un caso de evento lance una aplicación que requiera introducir texto. Como el usuario ya sabe que hay que escribir texto, podría empezar a escribir antes de que la aplicación aparezca en el panel frontal. Si está habilitada la opción **Lock front panel until the event case for this event completes**, una vez que se lance la aplicación y aparezca en el panel frontal, procesará las pulsaciones de teclas en el orden en que ocurrieron. Si está inhabilitada la opción **Lock front panel until the event case for this event completes**, las pulsaciones de teclas podrían procesarse en otra parte del panel frontal, ya que LabVIEW no pone en cola su ejecución para que dependa de que termine el caso de evento.

El bloqueo del panel frontal no afecta a ciertas acciones, como mover la ventana, interactuar con las barras de desplazamiento y hacer clic en el botón **Abort**.

LabVIEW puede generar eventos aunque ninguna estructura Event esté esperando para controlarlos. Como la estructura Event controla sólo un evento cada vez que se ejecuta, coloque la estructura Event en un bucle While para que ésta pueda controlar todos los eventos que ocurran.



Precaución Si no se ejecuta una estructura Event para controlar un evento y se habilita el bloqueo del panel frontal, la interfaz de usuario del VI no responderá. Si esto ocurre, haga clic en el botón **Abort** para detener el VI. Puede desactivar el bloqueo del panel frontal haciendo clic con el botón derecho en la estructura Event y desmarcando la casilla de verificación **Lock front panel until the event case for this event completes** en el cuadro de diálogo **Edit Events**. No puede desactivar el bloqueo del panel frontal para eventos de filtrado.

Registro de eventos estáticos

El registro de eventos estáticos sólo se permite para eventos de la interfaz de usuario. Use el cuadro de diálogo **Edit Events** para configurar una estructura Event para que controle un evento registrado estáticamente. Seleccione el origen de evento, que puede ser la aplicación, el VI o un solo control. Seleccione un evento que pueda generar el origen del evento, como Panel Resize, Value Change, etc. Modifique el caso para controlar los datos de evento conforme a los requisitos de la aplicación.

LabVIEW registra estáticamente eventos de forma automática y transparente cuando ejecute un VI que contenga una estructura Event. LabVIEW genera eventos para un VI sólo mientras se está ejecutando el VI o cuando otro VI en ejecución llama al VI como un subVI.

Cuando ejecuta un VI, LabVIEW configura ese VI de alto nivel y la jerarquía de subVIs a los que llama el VI en su diagrama de bloques para un estado de ejecución llamado reservado. No puede modificar un VI ni hacer clic en el botón **Run** mientras el VI está en estado reservado, porque se puede llamar al VI como un subVI en cualquier momento mientras su VI primario se ejecuta. Cuando LabVIEW configura un VI al estado reservado, automáticamente registra los eventos que configuró estáticamente en todas las estructuras Event del diagrama de bloques de ese VI. Cuando el VI de alto nivel termina la ejecución, LabVIEW lo configura (y su jerarquía de subVI) al estado de ejecución inactivo y automáticamente desregistra los eventos.

Consulte `labview\examples\general\uievents.llb` para ver ejemplos de uso del registro de eventos estáticos.

Configuración de eventos

Antes de configurar eventos para que los controle la estructura Event, consulte la sección *Alarmas y recomendaciones al usar eventos en LabVIEW* de la *Ayuda de LabVIEW*.

Siga estos pasos para configurar un caso de estructura Event para controlar un evento.

1. (Opcional) Si desea configurar la estructura Event para que controle el evento de un usuario, un control booleano de un control de botones de radio o un evento de interfaz de usuario que se genera basándose en una referencia a una aplicación, VI o control, primero debe registrar dinámicamente ese evento. Consulte el tema *Dynamically Registering Events* de la *Ayuda de LabVIEW* para obtener información adicional acerca de usar eventos dinámicos.
2. Haga clic con el botón derecho en el borde de la estructura Event y seleccione **Edit Events Handled by This Case** en el menú contextual para ver el cuadro de diálogo **Edit Events** y modificar el caso actual. También puede seleccionar **Add Event Case** en el menú contextual para crear un nuevo caso.
3. Especifique un origen de evento en el panel **Event Sources**.
4. Seleccione el evento que desee configurar para el origen de evento, como **Key Down**, **Timeout** o **Value Change** en la lista **Events**. Cuando seleccione un origen de evento dinámico en la lista **Event Sources**, la lista **Events** mostrará ese evento. Es el mismo evento que seleccionó cuando registró el evento. Si ha registrado eventos dinámicamente y ha cableado **event reg refnum out** en el terminal de eventos dinámicos, los orígenes aparecerán en la sección **Dynamic**.
5. Si desea añadir eventos adicionales para que los gestione el caso actual, haga clic en el botón + y repita los pasos 3 y 4 para especificar cada evento adicional. La sección **Event Specifiers** de la parte superior del cuadro de diálogo muestra todos los eventos que controlará el caso. Cuando hace clic en un elemento de esta lista, la sección **Event Sources** se actualiza para resaltar el origen de evento que seleccionó. Puede repetir los pasos 3 y 4 para redefinir cada evento o hacer clic en el botón **X** para eliminar el evento seleccionado.
6. Haga clic en el botón **OK** para guardar la configuración y cerrar el cuadro de diálogo. Los casos de eventos que configuró aparecerán como opciones de selección en la etiqueta del selector de eventos en la parte superior de la estructura Event y el nodo Event Data mostrará los datos comunes a todos los eventos controlados en ese caso.

7. (Opcional) Puede usar un evento Timeout para configurar una estructura Event para que espere un tiempo determinado a que ocurra un evento. Cablee un valor al terminal Timeout en la parte superior izquierda de la estructura Event para especificar el número de milisegundos que debe esperar la estructura Event a que ocurra un evento antes de generar un evento Timeout. El valor predeterminado del terminal Timeout es -1, que especifica esperar indefinidamente hasta que ocurra un evento.
8. Repita los pasos 1 al 6 para cada caso de evento que desee configurar.

Consulte los siguientes VIs para ver ejemplos de uso de eventos:

labview\examples\general\dynaminevents.llb

labview\examples\general\uievents.llb

Ejemplo de eventos

La figura 3-1 muestra una estructura Event configurada con el evento Menu Selection (User). Este VI usa la estructura Event para capturar selecciones de menú realizadas con el menú definido por el usuario y llamado `sample.rtm`. El `ItemTag` devuelve el elemento de menú que se seleccionó y `MenuRef` devuelve el refnum a la barra de menús. Esta información se pasa a la función `Get Menu Item Info`. Consulte `examples\general\uievents.llb` para ver más ejemplos de uso de eventos.

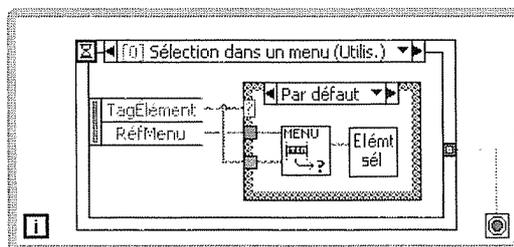


Figura 3-1. Evento Menu Selection (User)



Nota Si usa la función `Get Menu Selection` con una estructura Event configurada para controlar el mismo elemento de menú, la estructura Event tiene prioridad, y LabVIEW ignora la función `Get Menu Selection`. En cualquier VI, use la estructura Event o la función `Get Menu Selection` para controlar eventos de menú, no ambas.

Ejercicio 3-1 Concepto: experimentar con estructuras Event

Objetivo

Experimente con el funcionamiento de la estructura Event en un VI.

Descripción

Use y modifique un VI que contenga una estructura Event.

1. Abra el NI Example Finder.
 - Seleccione **Help»Find Examples**.
2. Abra el ejemplo de VI New Event Handler.
 - Vaya a **Building User Interfaces»Acquiring User Input»General** y haga doble clic en `New Event Handler.vi`.
3. Habilite el resaltado de ejecución en el diagrama de bloques.
4. Ejecute el VI.
5. Observe el funcionamiento del VI cuando haga clic en los botones del panel frontal.
6. Detenga el VI.
7. Modifique el VI para que responda a un evento Value Change con un nuevo control en el panel frontal.
 - Cambie al panel frontal del VI.
 - Cree una copia de un botón booleano del panel frontal.
 - Cambie el texto del botón y la etiqueta del botón a `New Button`.
 - Haga clic con el botón derecho del nuevo botón y compruebe que **Mechanical Action** esté configurado en **Latch When Released**.
 - Cambie al diagrama de bloques del VI.
 - Haga clic con el botón derecho en el borde de la estructura Event y seleccione **Add Event Case** en el menú contextual para abrir el cuadro de diálogo **Edit Events**.

- ❑ Seleccione **New Button** en la sección **Event Sources** y seleccione **Value Change** en la sección **Events** como se muestra en la figura 3-2.

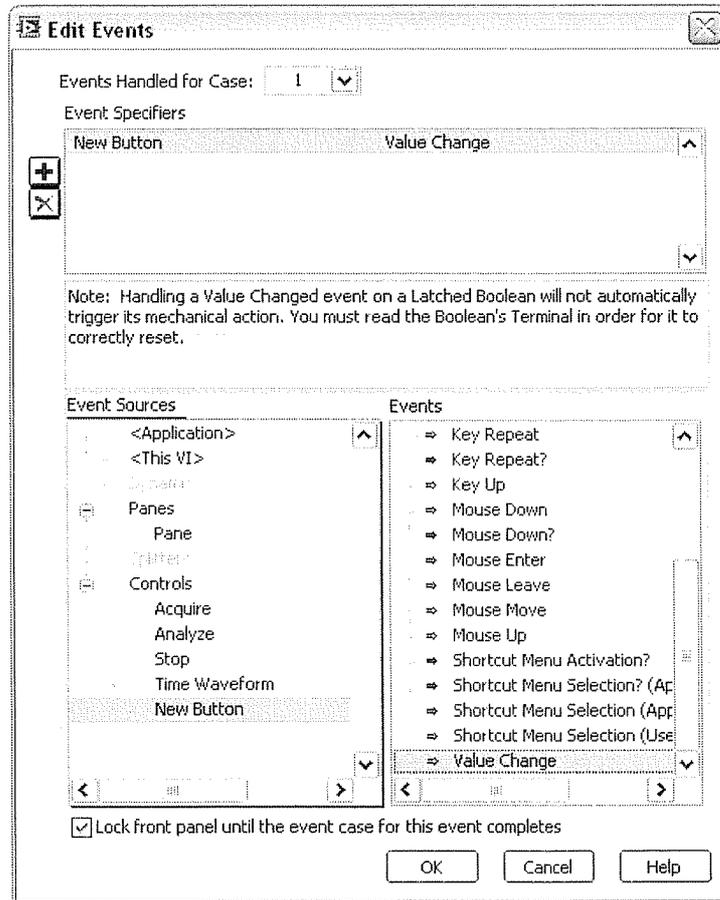


Figura 3-2. Evento New Button Value Change

8. Haga clic en **OK** para crear el nuevo caso de estructura Event.
 - ❑ Coloque una función One Button Dialog en el caso de evento New Button y cablee una constante de cadena de caracteres a la entrada **message**. Configure la constante de cadena de caracteres en New Event Case.
 - ❑ Cablee el dato Time Waveform al caso.
 - ❑ Cablee una constante False al túnel booleano.
 - ❑ Ejecute el VI y haga clic en **New Button**. Se abrirá un cuadro de diálogo y mostrará el mensaje **New Event Case**.

9. Modifique la estructura Event para añadir un Filter Event para desechar el evento Panel Close.
 - Haga clic con el botón derecho en la estructura Event y seleccione **Add Event Case** en el menú contextual para abrir el cuadro de diálogo **Edit Events**.
 - Seleccione **<This VI>** en la sección **Event Sources**.
 - Seleccione **Panel Close?** en la lista **Events** y haga clic en el botón **OK**.
 - Cablee una constante True al Event Filter Node Discard? en el caso Panel Close?.
 - Cablee el dato Time Waveform al caso.
 - Cablee una constante False al túnel booleano.
10. Ejecute el VI.
11. Intente cerrar el VI cerrando el panel frontal.
12. Haga clic en el botón **STOP** para detener el VI.
13. Abra el NI Example Finder.
14. Abra el ejemplo de VI Old Event Handler.
 - Vaya a **Building User Interfaces»Acquiring User Input»General** y haga doble clic en `Old Event Handler.vi`.
15. Habilite el resaltado de ejecución en el diagrama de bloques.
16. Ejecute el VI.
17. Observe el funcionamiento del VI Old Event Handler y compare y contraste el funcionamiento con el VI New Event Handler.
18. Cierre todos los VIs sin guardar los cambios.

Fin del ejercicio 3-1

B. Patrones de diseño basados en eventos

Los patrones de diseño basados en eventos ganan en eficiencia porque sólo responden cuando ocurre un evento. Cuando LabVIEW ejecuta la estructura Event, el VI que contiene la estructura Event se suspende hasta que ocurren o se generan eventos registrados. Cuando se genera un evento registrado, la estructura Event se activa automáticamente y ejecuta el subdiagrama apropiado para controlar el evento.

Patrón de diseño del controlador de eventos de la interfaz de usuario

El patrón de diseño del controlador de eventos de la interfaz de usuario incluye una arquitectura potente y eficaz para controlar la interacción del usuario con LabVIEW. Use el controlador de eventos de la interfaz de usuario para detectar cuándo un usuario cambia el valor de un control, mueve el ratón o hace clic en él o pulsa una tecla.

La plantilla estándar del controlador de eventos de la interfaz de usuario consta de una estructura Event contenida en un bucle While, como en la figura 3-3. Configure la estructura Event para tener un caso para cada categoría de evento que desee detectar. Cada caso de evento contiene el código de control que se ejecuta inmediatamente después de que ocurra un evento.

Como el bucle del controlador de eventos se activa precisamente cuando ocurre un evento y se suspende entre eventos, no tiene que sondear ni leer valores de control constantemente para detectar cuándo un usuario hace clic en un botón. El controlador de eventos de la interfaz de usuario permite minimizar el uso del procesador sin sacrificar la interactividad.

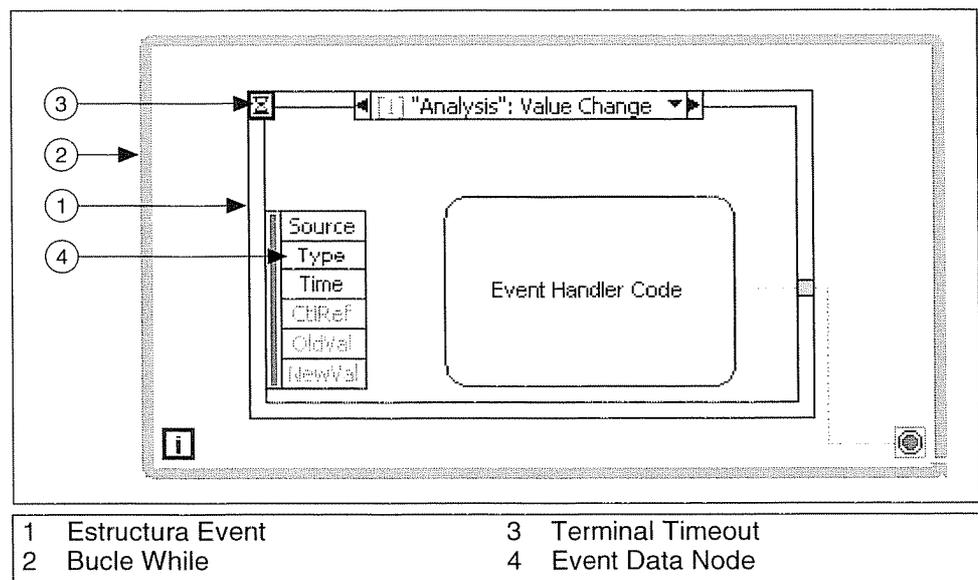


Figura 3-3. Patrón de diseño del controlador de eventos de la interfaz de usuario

Un problema común al usar el controlador de eventos de la interfaz de usuario es que computa el final del bucle While antes de que se ejecute la estructura Event. Esto puede hacer que el bucle While se itere una vez más de lo esperado. Para evitar esta situación, calcule el final del bucle While en todo el código de control de eventos.

El código del controlador de eventos debe ejecutarse rápidamente, en general en menos de 200 ms. Si es más lento, puede parecer que la interfaz de usuario se bloquea. Asimismo, si el código del controlador de eventos tarda mucho en ejecutarse, la estructura Event podría bloquearse. De forma predeterminada, el panel frontal se bloquea mientras se controla un evento. Puede desactivar el bloqueo del panel frontal para cada caso de evento para que la interfaz de usuario sea más sensible. Sin embargo, los nuevos eventos que se generan mientras se está controlando un evento no se controlarán inmediatamente. Por lo tanto, la interfaz de usuario aún parecerá que no responde.

Todo código que esté en un caso de evento no puede compartirse con otra estructura Event. Debe utilizar un buen diseño de código al usar la estructura Event. Modularice el código que se compartirá entre varios casos de estructura Event.

La estructura Event incluye un evento Timeout, que permite controlar cuándo se ejecuta el evento Timeout. Por ejemplo, si configura un Timeout de 200 ms, el caso del evento Timeout se ejecuta cada 200 ms en ausencia de otros eventos. Puede usar el evento Timeout para realizar la temporización crítica de su código.

Patrón de diseño de (eventos) productor/consumidor

Uno de los patrones de diseño más versátiles y flexibles combina los patrones de diseño del productor/consumidor y el controlador de eventos de interfaz de usuario. Un VI creado usando el patrón (eventos) productor/consumidor responde a la interfaz de usuario asincrónicamente, para que la interfaz de usuario pueda responder continuamente al usuario. El bucle consumidor de este patrón responde cuando ocurren eventos, al igual que el bucle consumidor del patrón de diseño de (datos) productor/consumidor.

El patrón de diseño de (eventos) productor/consumidor usa la misma implementación que el patrón de diseño de (datos) productor/consumidor, salvo que el bucle productor usa una estructura Event para responder a los eventos de la interfaz de usuario, como se ve en la figura 3-4. La estructura Event permite la respuesta continua a la interacción del usuario.

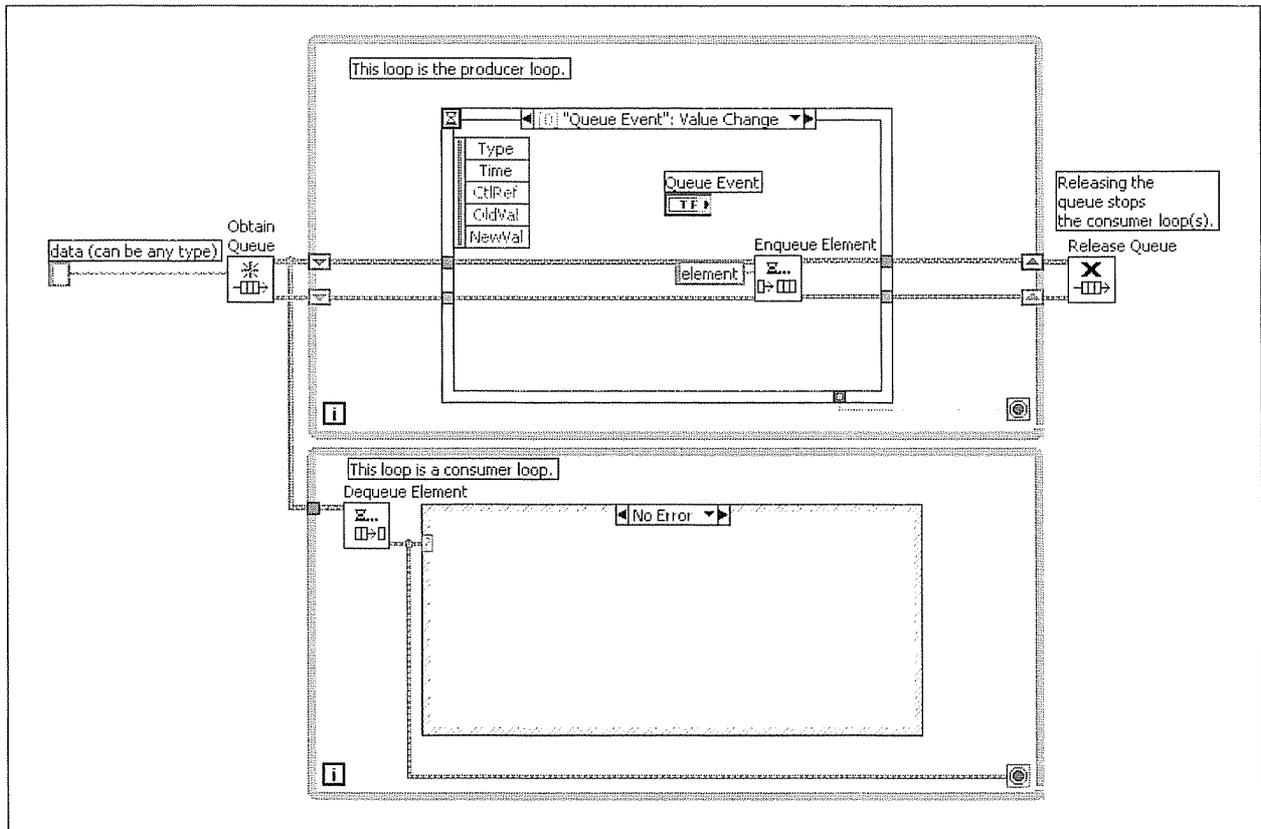


Figura 3-4. Patrón de diseño de (eventos) productor/consumidor

La figura 3-4 muestra cómo puede usar los VIs Synchronization y las funciones para añadir funcionalidad al patrón de diseño.

Las colas pueden transferir cualquier tipo de datos. El tipo de datos transferido de la figura 3-4 es una cadena de caracteres. Una cadena de caracteres no es el tipo de datos más eficaz para pasar datos en patrones de diseño. Un tipo de datos más eficaz para pasar datos en patrones de diseño es un cluster que consta de un control de tipo enumerado y uno de tipo "variant".

Ejercicio 3-2 **Concepto: experimentar con patrones de diseño basados en eventos**

Objetivo

Observe la funcionalidad y el diseño de patrones de diseño de LabVIEW basados en eventos integrados.

Descripción

Explore las plantillas del VI de patrones de diseño basados en eventos que incluye LabVIEW y observe cómo funcionan. Use las plantillas de VI de patrones de diseño basados en eventos de LabVIEW como base para la arquitectura subyacente de los VIs que cree.

1. Seleccione **File»New** para abrir el cuadro de diálogo **New**.
2. En la sección **Create New**, expanda el árbol **VI»from Template»Frameworks»Design Patterns**.
3. Seleccione el patrón de diseño **User Interface Event Handler** desde el árbol y haga clic en el botón **OK** para abrir la plantilla.
4. Abra el diagrama de bloques, active el resaltado de ejecución y ejecute el VI.

Observe cómo controla el patrón de diseño los siguientes elementos:

- Transiciones de estado
 - Flujo de datos
 - VIs usados para crear patrones de diseño
5. Cierre el patrón de diseño. No guarde los cambios.
 6. Repita los pasos 1 al 5 para observar el funcionamiento del patrón de diseño **Producer/Consumer Design Pattern (Events)**.

Fin del ejercicio 3-2

Autorrevisión: cuestionario

1. Las estructuras Event se suelen utilizar en bucles While.
 - a. Verdadero
 - b. Falso
2. ¿Cuáles de los siguientes son eventos de interfaz de usuario?
 - a. Hacer clic con el ratón
 - b. Pulsación de tecla
 - c. Event Filter Node
 - d. Cambio de valor de un control
3. Los eventos de interfaz de usuario permiten sincronizar acciones del usuario en el panel frontal con la ejecución del diagrama de bloques.
 - a. Verdadero
 - b. Falso

Autorrevisión: respuestas al cuestionario

1. Las estructuras Event se suelen utilizar en bucles While.
 - a. **Verdadero**
 - b. Falso
2. ¿Cuáles de los siguientes son eventos de interfaz de usuario?
 - a. **Hacer clic con el ratón**
 - b. **Pulsación de tecla**
 - c. Event Filter Node
 - d. **Cambio de valor de un control**
3. Los eventos de interfaz de usuario permiten sincronizar acciones del usuario en el panel frontal con la ejecución del diagrama de bloques.
 - a. **Verdadero**
 - b. Falso