
Relacionar datos

En ocasiones es conveniente agrupar datos relacionados entre sí. Use arrays y clusters para agrupar datos relacionados en LabVIEW. Los arrays combinan datos del mismo tipo en una estructura de datos; los clusters combinan datos de diferente tipo en una estructura de datos. Use definiciones tipo para definir clusters y arrays personalizados. Esta lección explica arrays, clusters y definiciones tipo, así como aplicaciones donde su uso puede resultar beneficioso.

Temas

- A. Arrays
- B. Clusters
- C. Definiciones tipo

A. Arrays

Un array consta de elementos y dimensiones. Los elementos son los datos que conforman el array. Una dimensión es la longitud, altura o profundidad de un array. Un array puede tener una o más dimensiones y un máximo de $(2^{31}) - 1$ elementos por dimensión, si lo permite la memoria.

Puede crear arrays de tipos de datos numéricos, booleanos, rutas, cadenas de caracteres, forma de onda y cluster. Use arrays cuando trabaje con un conjunto de datos similares y cuando realice cálculos repetitivos. Los arrays son ideales para almacenar datos que recopile de formas de onda o datos generados en bucles, donde cada iteración de un bucle produce un elemento del array.



Nota Los índices de array en LabVIEW están basados en cero. El índice del primer elemento del array, independientemente de su dimensión, es cero.

Restricciones

No puede crear arrays de arrays. Sin embargo, puede usar un array multidimensional o crear un array de clusters donde cada cluster contenga uno o más arrays. Asimismo, no puede crear un array de controles subpanel, controles de fichas, controles .NET, controles ActiveX, gráficos tipo “chart” o gráficos tipo “XY graph” de varias curvas.

Consulte la sección de clusters de esta lección para obtener información adicional acerca de los clusters.

Un ejemplo de un array simple es un array de texto que enumera los nueve planetas del sistema solar. LabVIEW representa esto como un array 1D de cadenas de caracteres con nueve elementos.

Los elementos de array están ordenados. Un array utiliza un índice para que pueda acceder fácilmente a un elemento concreto. El índice está basado en cero, lo que significa que está en el intervalo de 0 a $n - 1$, siendo n el número de elementos del array. Por ejemplo, $n = 9$ para los nueve planetas, de modo que el intervalo del índice es de 0 a 8. Tierra es el tercer planeta, por lo que su índice es 2.

La figura 5-1 muestra un ejemplo de un array de valores numéricos. El primer elemento que aparece en el array (3 . 00) está en el índice 1 y el segundo elemento (1 . 00) está en el índice 2. El elemento del índice 0 no aparece en esta imagen, porque el elemento 1 está seleccionado en la visualización del índice. El elemento seleccionado en el índice siempre hace referencia al elemento que aparece en la esquina superior izquierda de la visualización del array.

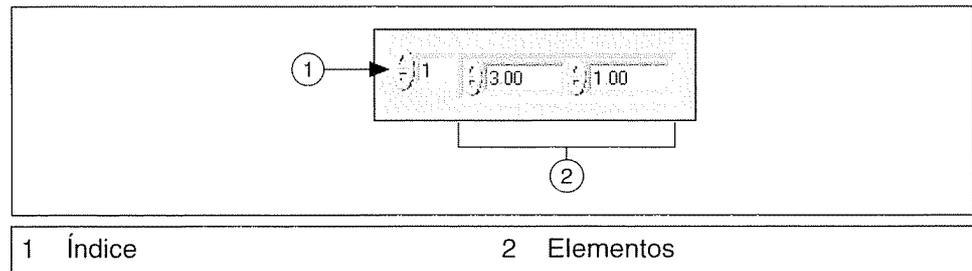


Figura 5-1. Control array de valores numéricos

Creación controles e indicadores de tipo array

Cree un control o indicador de array en el panel frontal añadiendo una estructura array al panel frontal, como se muestra en el siguiente panel frontal, y arrastrando un objeto o elemento de datos, que puede ser un control o indicador numérico, booleano, de cadena de caracteres, de ruta, de refnum o de cluster, en la estructura array.

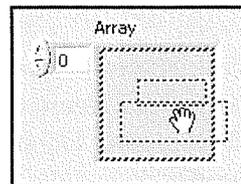


Figura 5-2. Colocar un control numérico en una estructura array

Si intenta arrastrar un control o indicador no válido a la estructura array, no podrá colocar el control o el indicador en la estructura array.

Debe insertar un objeto en la estructura array antes de usar el array en el diagrama de bloques. De lo contrario, el terminal del array aparecerá negro, con un paréntesis vacío y no tendrá tipos de datos asociados a él.

Arrays bidimensionales

Los ejemplos anteriores usan arrays 1D. Un array 2D almacena elementos en una cuadrícula. Requiere un índice de columna y un índice de fila para buscar un elemento, ambos basados en cero. La siguiente ilustración muestra un array de 8 columnas por 8 filas, que contiene $8 \times 8 = 64$ elementos.

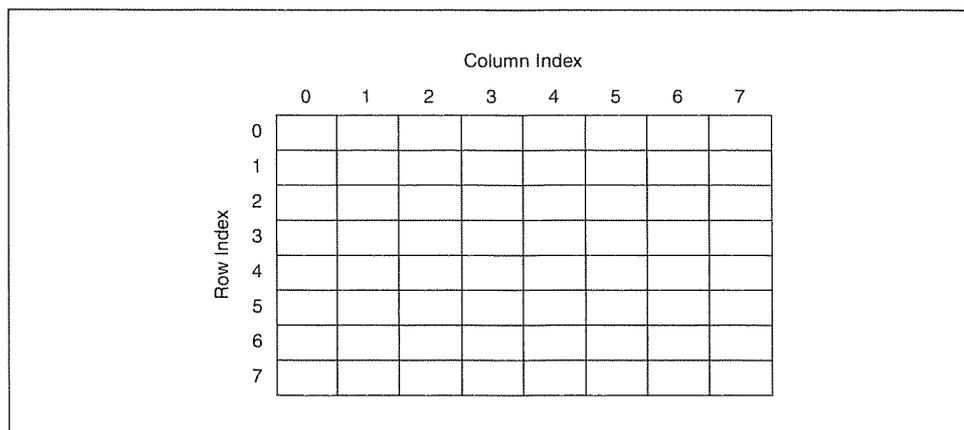


Diagrama de un array 2D de 8 columnas por 8 filas. El eje horizontal superior está etiquetado como "Column Index" con valores de 0 a 7. El eje vertical izquierdo está etiquetado como "Row Index" con valores de 0 a 7. El array se muestra como una cuadrícula de 8x8 celdas vacías.

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

Para añadir un array multidimensional al panel frontal, haga clic con el botón derecho en la pantalla del índice y seleccione **Add Dimension** en el menú contextual. También puede cambiar el tamaño del índice hasta que tenga tantas dimensiones como desee.

Inicialización de arrays

Puede inicializar un array o dejarlo sin inicializar. Para inicializar un array, debe definir el número de elementos de cada dimensión y el contenido de cada elemento. Un array sin inicializar contiene un número fijo de dimensiones pero ningún elemento. La figura 5-3 muestra un control de array 2D sin inicializar. Observe que todos los elementos están atenuados. Esto indica que el array no está inicializado.

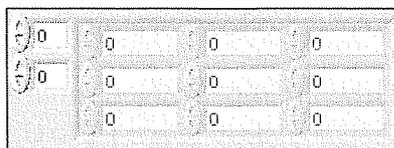


Figura 5-3. Array 2D sin inicializar

En la figura 5-4, seis elementos están inicializados. En un array 2D, tras inicializar un elemento de una fila, los elementos restantes de ésta se inicializan y se rellenan con el valor predeterminado para el tipo de dato. Por ejemplo, en la figura 5-4, si escribe 4 en el elemento de la primera columna, tercera fila, los elementos de la segunda y tercera columna de la tercera fila se rellenan automáticamente con un 0.

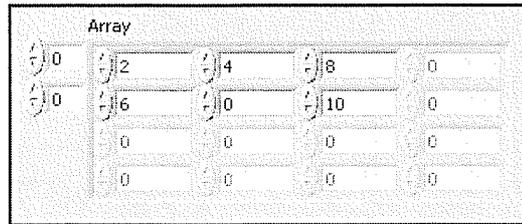


Figura 5-4. Array 2D inicializado con seis elementos

Creación de constantes de arrays

Para crear una constante de array en el diagrama de bloques, seleccione una constante de array en la paleta de **Funciones**, coloque la estructura array en el diagrama de bloques y coloque una constante de cadena de caracteres, una constante numérica, una constante booleana o una constante de cluster en la estructura array. Puede utilizar una constante de array para almacenar datos de constantes o para comparar con otro array.

Autoindexado de entradas de array



Si cablea un array a o desde un bucle For o While, puede vincular cada iteración del bucle en un elemento de ese array habilitando el autoindexado. La imagen del túnel cambia de un cuadrado relleno a la imagen de la izquierda para indicar el autoindexado. Haga clic con el botón derecho en el túnel y seleccione **Enable Indexing** o **Disable Indexing** desde el menú contextual para cambiar el estado del túnel.

Entradas de arrays

Si habilita el autoindexado en un array cableado a un terminal de entrada de un bucle For, LabVIEW configurará el terminal de conteo en el tamaño del array, por lo que no tiene que cablear el terminal de conteo. Dado que puede utilizar bucles For para procesar arrays elemento por elemento LabVIEW habilita el autoindexado de forma predeterminada para todos los arrays que cablee a un bucle For. Puede desactivar el autoindexado si no necesita procesar arrays, elemento por elemento.

En la figura 5-5, el bucle For se ejecuta un número de veces equivalente al número de elementos del array. Normalmente, si el terminal de conteo del bucle For no está cableado, la flecha de ejecución aparece rota. Sin embargo, en este caso la flecha de ejecución no aparece rota.

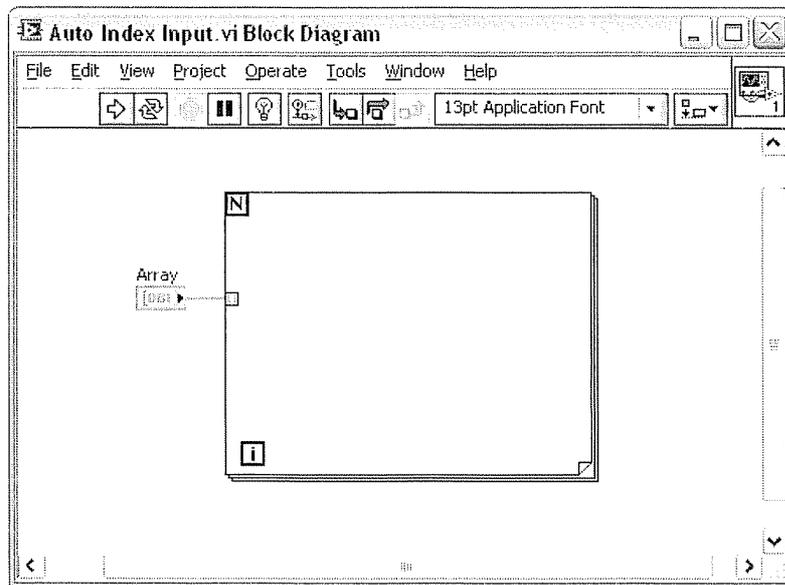


Figura 5-5. Array utilizado para configurar el contaje de bucles For

Si habilita el autoindexado para más de un túnel o si cablea el terminal de contaje, el número real de iteraciones será el equivalente a la menor de las elecciones. Por ejemplo, si dos arrays autoindexados entran en el bucle, con 10 y 20 elementos respectivamente, y cablea un valor de 15 al terminal de contaje, el bucle se seguirá ejecutando sólo 10 veces, indizando todos los elementos del primer array pero sólo los 10 primeros elementos del segundo array.

Salidas de arrays

Cuando autoindexa un túnel de salida de un array, el array de salida recibe un nuevo elemento de cada iteración del bucle. Por lo tanto, el tamaño de los arrays de salida autoindexados tienen siempre un tamaño igual al número de iteraciones del bucle For.

El cable del túnel de salida al indicador de array aparece más grueso cuando se produce el cambio a un array en el borde del bucle, y el túnel de salida contiene corchetes que representan a un array, como en la figura 5-6.

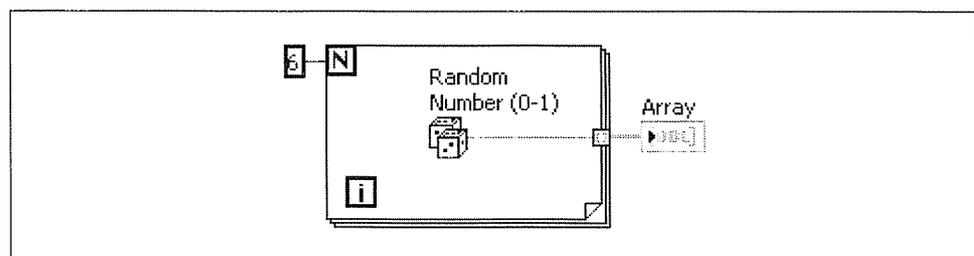


Figura 5-6. Salida autoindexada

Haga clic con el botón derecho en el borde del bucle y seleccione **Enable Indexing** o **Disable Indexing** en el menú contextual para habilitar o deshabilitar el autoindexado. El autoindexado de bucles While está deshabilitado de forma predeterminada.

Por ejemplo, deshabilite el autoindexado si necesita que sólo el último valor que salga por el túnel.

Creación de arrays bidimensionales

Puede usar dos bucles For, anidados uno dentro del otro, para crear un array 2D. El bucle For externo crea los elementos de la fila, mientras que el interno crea los elementos de la columna, como en la figura 5-7.

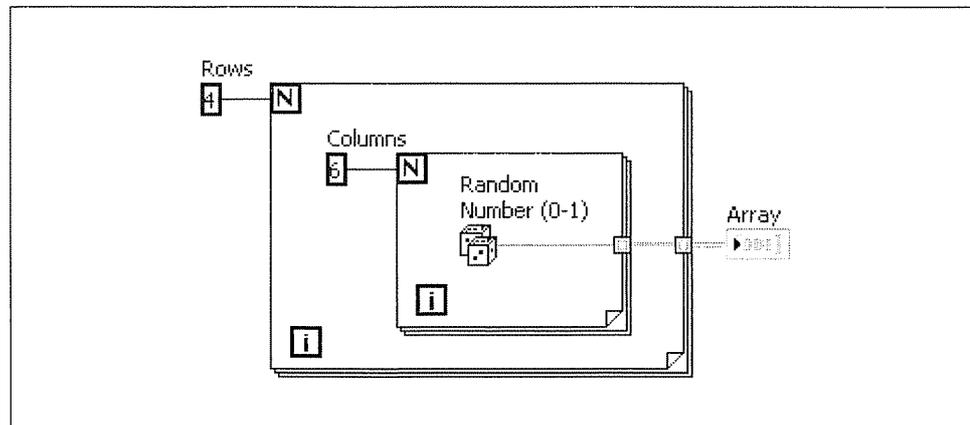


Figura 5-7. Creación de un array 2D

Ejercicio 5-1 Concepto: manipulación de arrays

Objetivo

Manipular arrays con varias funciones de LabVIEW.

Descripción

Recibe un VI y debe mejorarlo para varios fines. Para cada parte de este ejercicio, empiece con el Array Investigation.vi situado en el directorio <Exercises>\LabVIEW Basics I\Manipulating Arrays. El panel frontal de este VI aparece en la figura 5-8.

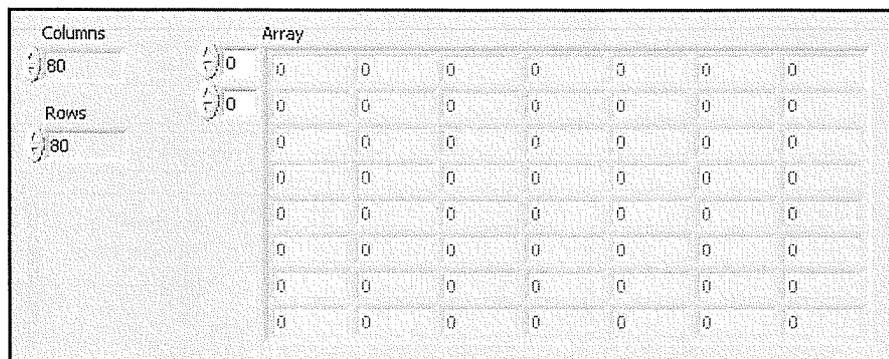


Figura 5-8. Panel frontal del VI Array Investigation

La figura 5-9 muestra el diagrama de bloques de este VI.

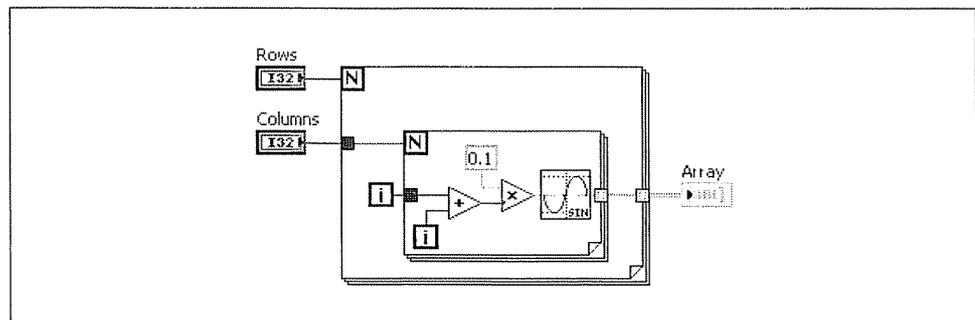


Figura 5-9. Diagrama de bloques del VI Array Investigation

Este ejercicio se divide en dos partes. Recibirá primero el escenario de cada parte. Consulte el final de este ejercicio para ver instrucciones detalladas de implementación de cada parte.

Parte 1: Iterar, modificar y graficar el array

Modifique el VI Array Investigation para que, después de crear el array, éste se indexe dentro de bucles For donde multiplique cada elemento del array por 100 y convierta cada elemento en el número entero más cercano. Grafique el array 2D resultante en un gráfico tipo “intensity graph”.

Parte 2: Iterar, modificar y graficar el array (simplificado)

Modifique el VI Array Investigation o la solución de la parte 1 para conseguir los mismos objetivos sin usar los bucles For anidados.

Part 3: Crear arrays de subconjunto

Modifique el VI Array Investigation para que el VI cree un nuevo array que albergue el contenido de la tercera fila, y otro array nuevo que albergue el contenido de la segunda columna.

Parte 1: Implementación

Modifique el VI Array Investigation para que, después de crear el array, éste se indexe dentro de bucles For donde multiplique cada elemento del array por 100 y convierta cada elemento en el número entero más cercano. Grafique el array 2D resultante en un gráfico tipo “intensity graph”.

1. Abra el Array Investigation.vi situado en el directorio <Exercises>\LabVIEW Basics I\Manipulating Arrays.
2. Guarde el VI como Array Investigation Part 1.vi.
3. Añada un gráfico tipo “intensity graph” al panel frontal del VI y autoescale los ejes X e Y, como en la figura 5-10. Para autoescalar los ejes, haga clic con el botón en el gráfico tipo “intensity graph” y seleccione **X Scale»AutoScale X** y **Y Scale»AutoScale Y**.



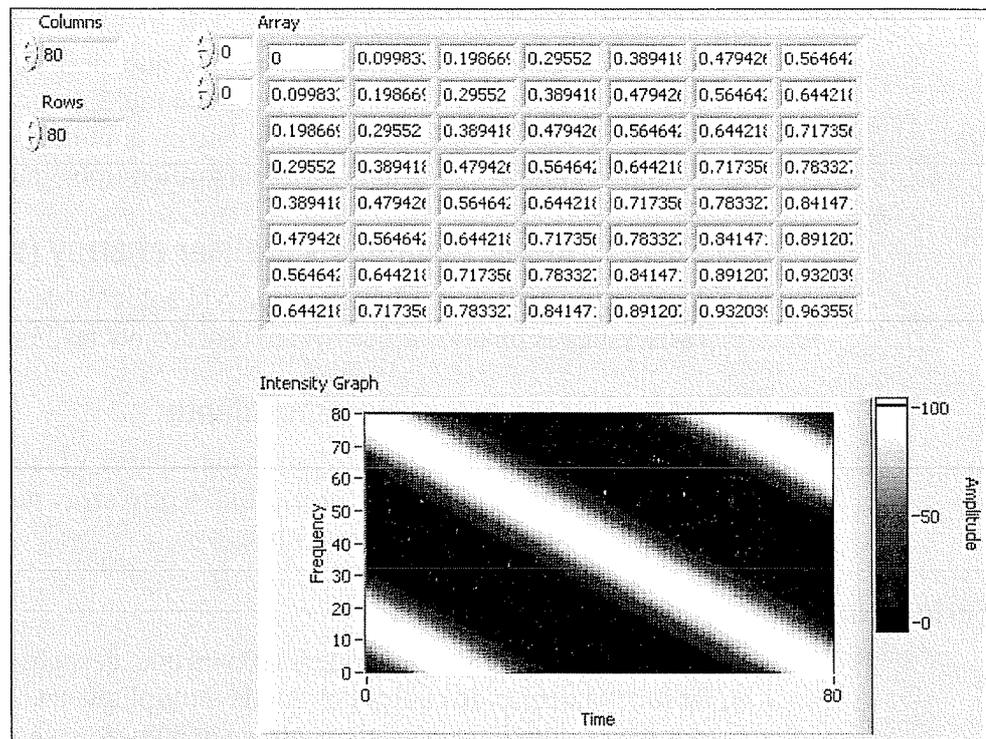


Figura 5-10. Panel frontal del VI Array Investigation Part 1

4. Abra el diagrama de bloques del VI.

Siga las instrucciones inferiores para crear un diagrama de bloques similar al de la figura 5-11.

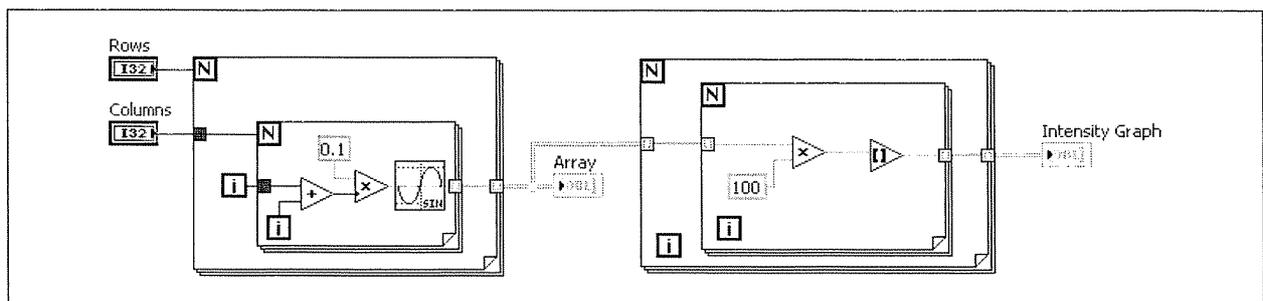


Figura 5-11. Diagrama de bloques del VI Array Investigation Part 1

5. Itere el array.



- Añada un bucle For a la derecha del código existente.
- Añada un segundo bucle For en el primer bucle For.
- Cablee el terminal del indicador de array con el borde del bucle For interior. Esto crea un túnel de entrada autoindexado en ambos bucles For.

6. Multiplique cada elemento del array por 100.



- Añada una función Multiply al bucle For interior.
- Cablee el túnel de entrada indexado a la entrada **x** de la función Multiply.
- Haga clic con el botón derecho en la entrada **y** y seleccione **Create» Constant** en el menú contextual.
- Escriba 100 en la constante.

7. Redondee cada elemento al número entero más cercano.



- Añada una función Round To Nearest a la derecha de la función Multiply.
- Cablee la salida de la función Multiply a la entrada de la función Round To Nearest.

8. Cree un array 2D en la salida de los bucles For para volver a crear el array modificado.

- Cablee la salida de la función Round To Nearest con el bucle For externo. Esto crea un túnel de salida autoindexado en ambos bucles For.

9. Cablee el array de salida con el indicador Intensity Graph.

10. Cambie al panel frontal.

11. Guarde el VI.

12. Escriba valores para **Rows** y **Columns**.

13. Ejecute el VI.

Parte 2: Implementación

Modifique la parte 1 para conseguir los mismos objetivos sin usar los bucles For anidados.

1. Abra el `Array Investigation Part 1.vi` si aún no está abierto.
2. Guarde el VI como `Array Investigation Part 2.vi`.
3. Abra el diagrama de bloques.

4. Haga clic con el botón derecho en el borde del bucle For interior, que contiene la función Multiply y la función Round to Nearest, y seleccione **Remove For Loop**.
5. Haga clic con el botón derecho en el borde del bucle For restante y seleccione **Remove For Loop** del menú contextual. Su diagrama de bloques debe parecerse al de la figura 5-12.

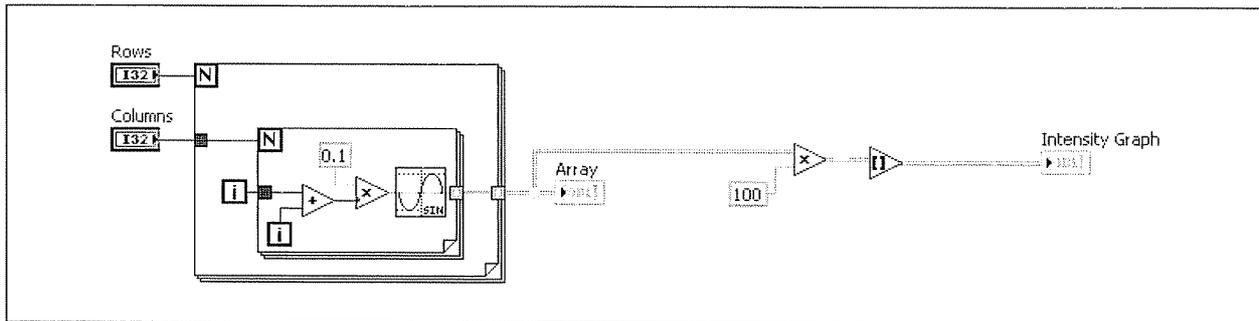


Figura 5-12. Diagrama de bloques del VI Array Investigation Part 2

6. Guarde el VI.
7. Cambie al panel frontal.
8. Escriba valores para **Rows** y **Columns**.
9. Ejecute el VI.

Observe que el VI se comporta del mismo modo que la solución de la parte 1. Esto se debe a que las funciones matemáticas son polimórficas. Por ejemplo, como la entrada x de la función Multiply es un array bidimensional y la entrada y es un escalar, la función Multiply multiplica cada elemento del array por el escalar, y produce un array de la misma dimensión que la entrada x .

Parte 3: Implementación

Modifique el VI Array Investigation para que el VI cree un nuevo array que albergue el contenido de la tercera fila, y otro array nuevo que albergue el contenido de la segunda columna.

1. Abra el Array Investigation.vi situado en el directorio <Exercises>\LabVIEW Basics I\Manipulating Arrays.
2. Guarde el VI como Array Investigation Part 3.vi.
3. Abra el diagrama de bloques del VI.

En los siguientes pasos creará un diagrama de bloques similar al de la figura 5-13.

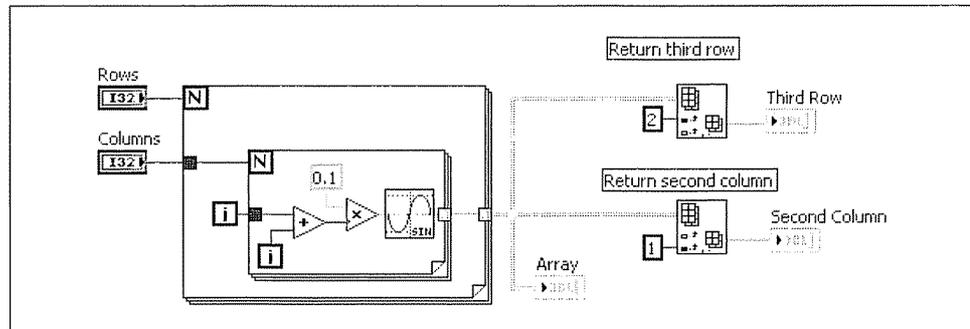


Figura 5-13. Diagrama de bloques del VI Array Investigation Part 3

4. Recupere la tercera fila de datos de **Array** utilizando la función Index Array.



- Añada la función Index Array al diagrama de bloques.
- Cablee **Array** a la entrada **array** de la función Index Array.



Consejo La función Index Array acepta un array n -dimensional. Tras cablear el array de entrada a la función Index Array, los nombres del terminal de entrada y salida cambian para coincidir con la dimensión del array cableado. Por lo tanto, cablee el array de entrada a la función Index Array antes de cablear cualquier otro terminal.

- Haga clic con el botón derecho en la entrada **index(row)** de la función Index Array.
- Seleccione **Create»Constant** en el menú contextual.
- Escriba 2 en la constante para recuperar la tercera fila. Recuerde que el índice empieza por cero.
- Haga clic con el botón derecho en la salida **subarray** de la función Index Array.
- Seleccione **Create»Indicator** en el menú contextual.
- Llame al indicador **Third Row**.

5. Recupere la segunda columna de datos de Array utilizando la función Index Array.



- Añada otra función Index Array al diagrama de bloques.
- Cablee **Array** a la entrada **array** de la función Index Array.
- Haga clic con el botón derecho en la entrada **disabled index(col)** de la función Index Array.
- Seleccione **Create»Constant**.
- Escriba 1 en la constante para recuperar la segunda columna dado que el índice empieza por cero.
- Haga clic con el botón derecho en la salida **subarray** de la función Index Array.
- Seleccione **Create»Indicator**.
- Llame al indicador `Second Column`.

6. Guarde el VI.

7. Cambie al panel frontal.

8. Escriba valores para **Rows** y **Columns**.

9. Ejecute el VI.

Fin del ejercicio 5-1

B. Clusters

Los clusters agrupan elementos de datos de tipos mixtos. Un ejemplo de un cluster es el cluster de error de LabVIEW, que combina un valor booleano, uno numérico y una cadena de caracteres. Un cluster es similar a un registro o una estructura en lenguajes de programación basados en texto.

Unir varios elementos de datos en clusters elimina el abarrotamiento de cables en el diagrama de bloques y reduce el número de terminales del panel de conectores que necesitan los subVIs. El panel de conectores tiene como mucho 28 terminales. Si su panel frontal contiene más de 28 controles e indicadores que desee pasar a otro VI, agrupe algunos en un cluster y asigne éste a un terminal del panel de conectores.

La mayoría de los clusters del diagrama de bloques tiene un terminal y un patrón de cables rosa. Los clusters de error tienen un terminal y un patrón de cables amarillo oscuro. Los clusters de valores numéricos, a veces denominados puntos, tienen un terminal y un patrón de cables marrón. Puede cablear clusters numéricos marrones a funciones Numeric, como Add o Square Root, para realizar la misma operación simultáneamente en todos los elementos del cluster.

Orden de elementos del cluster

Aunque los elementos del cluster y del array están ordenados, con la función Unbundle debe desunir todos los elementos del cluster enseguida. Puede utilizar la función Unbundle By Name para desunir elementos del cluster por nombre. Si utiliza la función Unbundle By Name, cada elemento del cluster debe tener una etiqueta. Los clusters también difieren de los arrays en que tienen un tamaño fijo. Al igual que un array, un cluster puede ser control o indicador. Un cluster no puede contener una mezcla de controles e indicadores.

Creación de controles e indicadores de clusters

Cree un control o indicador de clusters en el panel frontal añadiendo una estructura cluster al panel frontal, como se muestra en el siguiente panel frontal, y arrastrando un objeto o elemento de datos, que puede ser un control o indicador numérico, booleano, de cadena de caracteres, de ruta, refnum o cluster, en la estructura cluster.

Cambie el tamaño de la estructura cluster arrastrando el cursor mientras coloca ésta.

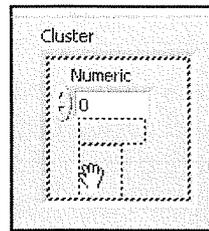


Figura 5-14. Creación de un control de cluster

La figura 5-15 es un ejemplo de un cluster que contiene tres controles: una cadena de caracteres, un interruptor booleano y un número. Un cluster puede ser un control o un indicador. No puede contener una mezcla de controles e indicadores.

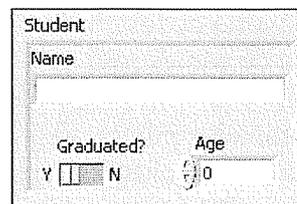


Figura 5-15. Ejemplo de control de cluster

Crear constantes de clusters

Para crear una constante de cluster en el diagrama de bloques, seleccione una constante de cluster en la paleta **Funciones**, coloque la estructura cluster en el diagrama de bloques y coloque una constante de cadena de caracteres, una constante booleana o una constante de cluster en la estructura cluster. Puede utilizar una constante de cluster para almacenar datos de constantes o para comparar con otro cluster.

Si tiene un control o indicador de cluster en la ventana del panel frontal y desea crear una constante de cluster que contenga los mismos elementos en el diagrama de bloques, puede arrastrar ese cluster desde la ventana del panel frontal al diagrama de bloques o hacer clic con el botón derecho en la ventana del panel frontal y seleccionar **Create»Constant** en el menú contextual.

Orden de clusters

Los elementos de clusters tienen un orden lógico no relacionado con su posición en la estructura. El primer objeto que coloca en el cluster es el elemento 0, el segundo es el elemento 1, etc. Si elimina un elemento, el orden se ajusta automáticamente. El orden del cluster determina el orden en que aparecen los elementos como terminales en las funciones **Bundle** y **Unbundle** en el diagrama de bloques. Puede ver y modificar el orden del cluster haciendo clic derecho en su borde y seleccionando **Reorder Controls In Cluster** en el menú contextual.

La barra de herramientas y el cluster cambian, como en la figura 5-16.

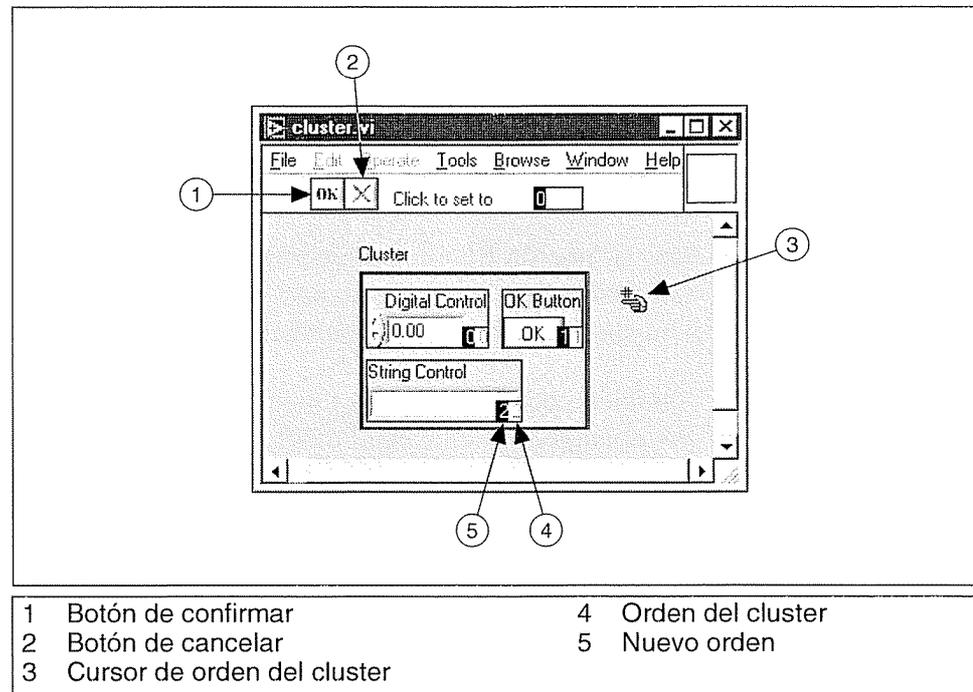


Figura 5-16. Reordenar un cluster

El cuadro blanco de cada elemento muestra su lugar actual en el orden del cluster. El cuadro negro muestra el nuevo lugar en el orden para un elemento. Para establecer la posición de un elemento en el cluster, escriba el número de orden nuevo en el cuadro de texto **Click to set to** y haga clic en el elemento. La posición del elemento en el cluster cambia, y se ajusta el orden del resto de elementos del cluster. Guarde los cambios haciendo clic en el botón **Confirm** de la barra de herramientas. Vuelva al orden original haciendo clic en el botón **Cancel**.

Para cablear clusters entre sí, ambos clusters deben tener el mismo número de elementos. Los elementos correspondientes, determinados por el orden del cluster, deben tener tipos de datos compatibles. Por ejemplo, si un valor numérico en coma flotante y de doble precisión de un cluster se corresponde según el orden del cluster con una cadena de caracteres en el segundo cluster, el cable del diagrama de bloques aparecerá roto y el VI no se ejecutará. Si los valores numéricos son representaciones distintas, LabVIEW los convertirá en la misma representación.

Uso de funciones de cluster

Use las funciones de cluster para crear y manipular clusters. Por ejemplo, puede realizar tareas similares a las siguientes:

- Extraer elementos de datos individuales de un cluster.
- Añadir elementos de datos individuales a un cluster.
- Dividir un cluster en sus elementos de datos individuales.

Use la función Bundle para ensamblar un cluster, use la función Bundle y la función Bundle by Name para modificar un cluster y use la función Unbundle y la función Unbundle By Name para desensamblar clusters.

También puede colocar las funciones Bundle, Bundle by Name, Unbundle y Unbundle by Name en el diagrama de bloques haciendo clic con el botón derecho en un terminal del cluster en el diagrama de bloques y seleccionando **Cluster, Class & Variant Palette** en el menú contextual. Las funciones Bundle y Unbundle contienen automáticamente el número correcto de terminales. Las funciones Bundle by Name y Unbundle by Name aparecen con el primer elemento del cluster. Use la herramienta de posicionamiento para cambiar el tamaño de las funciones Bundle by Name y Unbundle by Name para mostrar los otros elementos del cluster.

Ensamblaje de clusters

Use la función Bundle para ensamblar un cluster desde elementos individuales o para cambiar los valores de elementos individuales de un cluster existente sin tener que especificar nuevos valores para todos los elementos. Use la herramienta de posicionamiento para cambiar el tamaño de la función o haga clic con el botón derecho en una entrada de elemento y seleccione **Add Input** en el menú contextual.

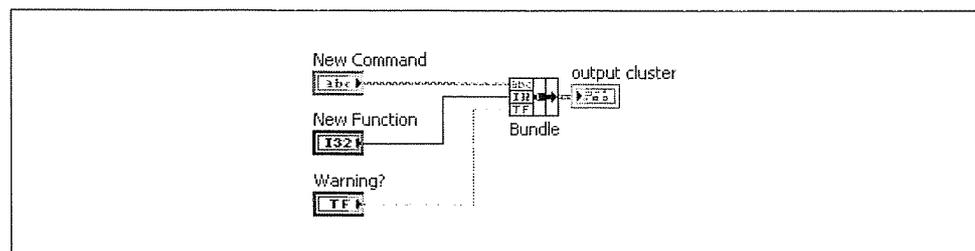


Figura 5-17. Ensamblaje de un cluster en el diagrama de bloques

Modificación de un cluster

Si cablea la entrada del cluster, puede cablear sólo los elementos que desee cambiar. Por ejemplo, el cluster Input Cluster de la figura 5-18 contiene tres controles.

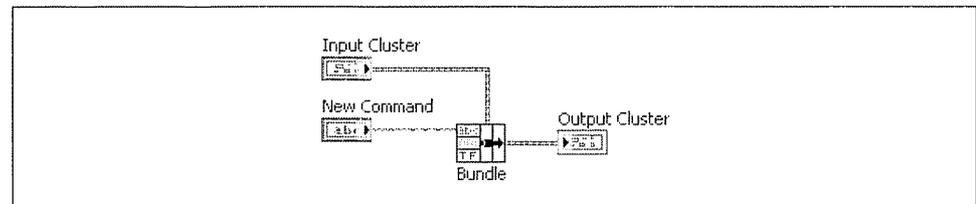


Figura 5-18. Función Bundle utilizada para modificar un cluster

Si conoce el orden del cluster, puede usar la función Bundle para cambiar el valor **Command** cableando los elementos de la figura 5-18.

También puede utilizar la función Bundle by Name para sustituir o acceder a elementos etiquetados de un cluster existente. La función Bundle by Name actúa como la función Bundle, pero en lugar de hacer referencia a elementos del cluster por su orden en el cluster, hace referencia a ellos por sus etiquetas propietarias. Puede acceder sólo a elementos con etiquetas propietarias. El número de entradas no tiene por qué coincidir con el número de elementos de **output cluster**.

Use la herramienta de posicionamiento para hacer clic en un terminal de entrada y seleccionar un elemento del menú desplegable. También puede hacer clic con el botón derecho en la entrada y seleccionar el elemento del menú contextual **Select Item**.

En la figura 5-19, puede utilizar la función Bundle by Name para cambiar **New Command** y **New Function**.

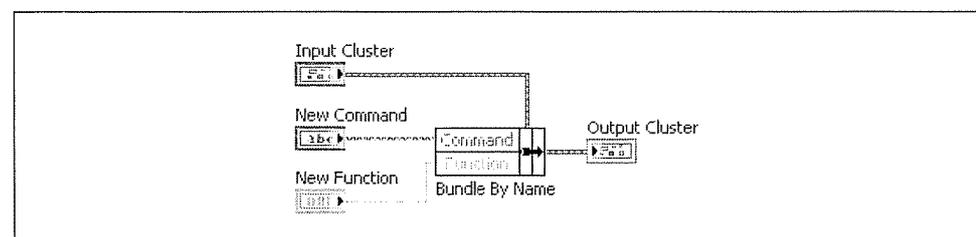


Figura 5-19. Función Bundle By Name utilizada para modificar un cluster

Use la función Bundle by Name para estructuras de datos que podrían cambiar durante el desarrollo. Si añade un nuevo elemento al cluster o modifica su orden, no necesita volver a cablear la función Bundle by Name porque los nombres siguen siendo válidos.

Desensamblaje de clusters

Use la función Unbundle para dividir un cluster en sus elementos individuales.

Use la función Unbundle by Name para devolver los elementos del cluster cuyos nombres especifique. El número de terminales de salida no depende del número de elementos del cluster de entrada.

Use la herramienta de posicionamiento para hacer clic en un terminal de salida y seleccionar un elemento del menú desplegable. También puede hacer clic con el botón derecho en el terminal de salida y seleccionar el elemento del menú contextual **Select Item**.

Por ejemplo, si usa la función Unbundle con el cluster de la figura 5-20, tiene cuatro terminales de salida que corresponden con los cuatro controles del cluster. Debe conocer el orden del cluster para poder asociar el terminal booleano correcto del cluster desunido con el correspondiente interruptor en el cluster. En este ejemplo, los elementos están ordenados de arriba a abajo empezando por el elemento 0. Si usa la función Unbundle by Name, puede tener un número arbitrario de terminales de salida y acceder a elementos individuales por nombre en cualquier orden.

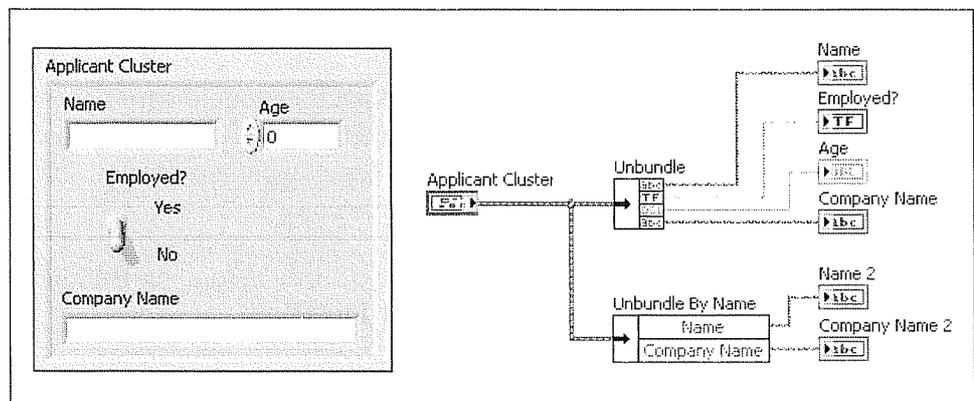


Figura 5-20. Unbundle y Unbundle By Name

Clusters de error

LabVIEW contiene un cluster personalizado llamado cluster de error.

LabVIEW usa los clusters de error para pasar información de error.

Un cluster de error contiene los siguientes elementos:

- **status:** es un valor booleano que produce TRUE si ocurrió un error.
- **code:** es un número entero con signo de 32 bits que identifica el error numéricamente.
- **source:** es una cadena de caracteres que identifica dónde ocurrió el error.

Consulte la lección 3, *Resolución de problemas y depuración de VIs*, para obtener información adicional acerca del uso de los clusters de error.

Consulte el tema *Handling Errors* de la *Ayuda de LabVIEW* para obtener información adicional acerca del uso de clusters de error.

Ejercicio 5-2 Concepto: clusters

Objetivo

Cree clusters en la ventana del panel frontal, reordene clusters y use las funciones de clusters para ensamblar y desensamblar clusters.

Descripción

En este ejercicio, siga las instrucciones para experimentar con clusters, orden del cluster y funciones del cluster. El VI que cree no tiene aplicaciones prácticas, pero resulta útil para entender los conceptos del cluster.

1. Abra un VI nuevo.
2. Guarde el VI como `Cluster Experiment.vi` en el directorio `<Exercises>\LabVIEW Basics I\Clusters`.

Siga las instrucciones inferiores para crear un panel frontal similar al de la figura 5-21.

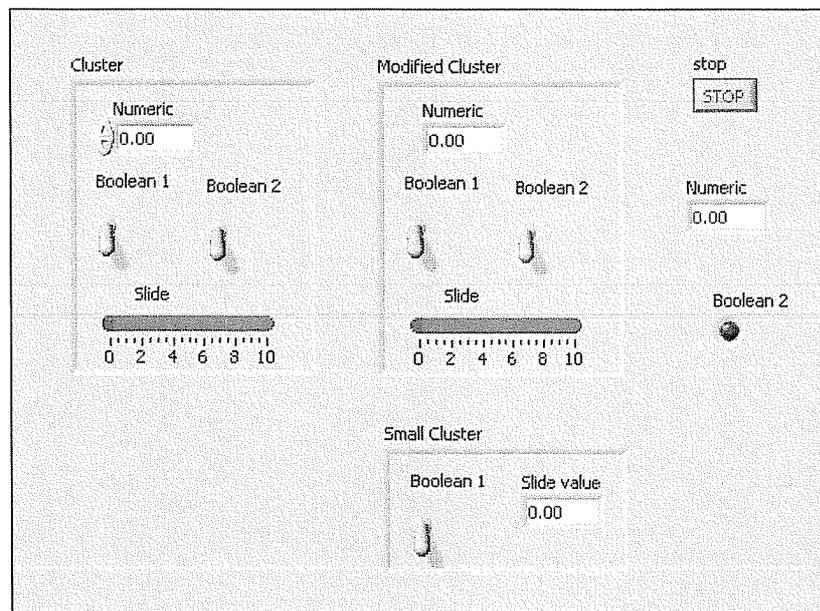


Figura 5-21. Panel frontal del VI Cluster Experiment

3. Añada un botón de parada a la ventana del panel frontal.
4. Añada un indicador numérico a la ventana del panel frontal.
5. Añada un LED redondo al panel frontal.
6. Llame al LED `Boolean 2`.

7. Cree un cluster llamado **Cluster**, que contenga un valor numérico, dos interruptores y un control deslizante.
 - Añada una estructura cluster al panel frontal.
 - Añada un control numérico al cluster.
 - Añada dos interruptores verticales al cluster.
 - Llame a los interruptores **Boolean** Boolean 1.
 - Añada una barra horizontal de deslizamiento de tipo relleno al cluster.
8. Cree **Modified Cluster**, que alberga los mismos contenidos que **Cluster**, pero indicadores en lugar de controles.
 - Cree una copia de **Cluster**.
 - Llame a la copia Modified Cluster.
 - Haga clic con el botón derecho en la estructura **Modified Cluster** y seleccione **Change to Indicator** en el menú contextual.
9. Cree **Small Cluster**, que contiene un indicador booleano y otro numérico.
 - Cree una copia de **Modified Cluster**.
 - Llame a la copia Small Cluster.
 - Elimine el segundo interruptor basculante.
 - Elimine el indicador del selector de relleno horizontal.
 - Haga clic con el botón derecho en **Small Cluster** y seleccione **Autosizing»Size to Fit**.
 - Vuelva a etiquetar el indicador numérico como Slide value.
 - Cambie el tamaño del cluster si es necesario.

10. Compruebe el orden de los elementos de **Cluster**, **Modified Cluster** y **Small Cluster**.

- Haga clic con el botón derecho en el límite de **Cluster** y seleccione **Reorder Controls in Cluster** en el menú contextual.
- Confirme el orden de clusters que aparece en la figura 5-22.
- Haga clic en el botón **Confirm** en la barra de herramientas para configurar el orden del cluster y salir del modo de edición del orden del cluster.
- Haga clic con el botón derecho en el límite de **Modified Cluster** y seleccione **Reorder Controls in Cluster** en el menú contextual.
- Confirme el orden del cluster que aparece en la figura 5-22. **Modified Cluster** debe tener el mismo orden de cluster que **Cluster**.
- Haga clic en el botón **Confirm** en la barra de herramientas para configurar el orden del cluster y salir del modo de edición del orden del cluster.
- Haga clic con el botón derecho en el límite de **Small Cluster** y seleccione **Reorder Controls in Cluster** en el menú contextual. Haga clic en el botón **OK** en la barra de herramientas para configurar el orden de clusters y salir del modo de edición del orden del cluster.
- Confirme el orden de los clusters que aparece en la figura 5-22.

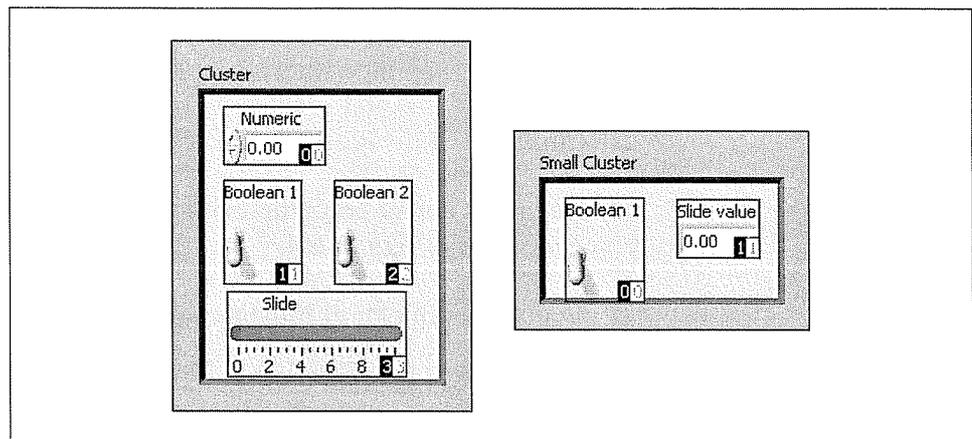


Figura 5-22. Orden de clusters

En los siguientes pasos, cree el diagrama de bloques de la figura 5-23.

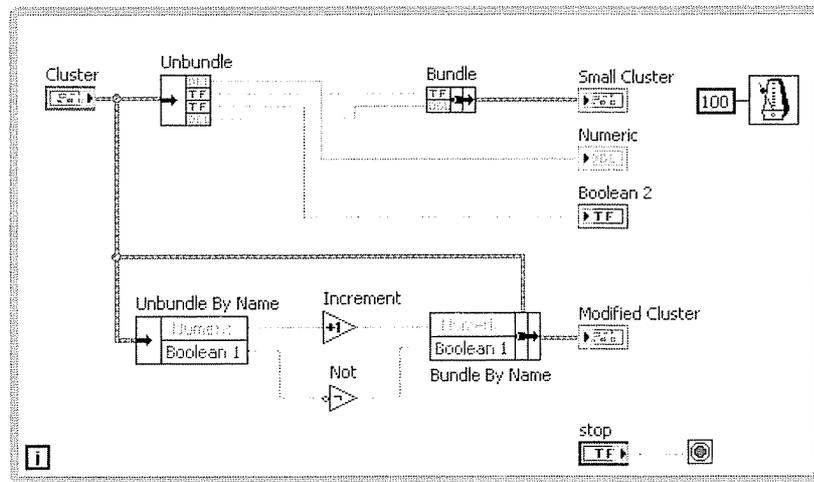


Figura 5-23. Diagrama de bloques del VI Cluster Experiment



11. Añada el bucle While desde la paleta **Structures** al diagrama de bloques.

12. Desensamble **Cluster**.



- Añada la función Unbundle al diagrama de bloques.
- Cablee **Cluster** a la entrada de la función Unbundle para cambiar automáticamente el tamaño de la función.

13. Ensamble **Small Cluster**.



- Añada la función Bundle al diagrama de bloques.
- Cablee la función Bundle como en la figura 5-23.

14. Ensamble **Modified Cluster**.



- Añada la función Unbundle by Name al diagrama de bloques.
- Cablee el **Cluster** a la función Unbundle by Name.
- Cambie el tamaño de la función Unbundle by Name para tener dos terminales de salida.
- Seleccione **Numeric** en el primer nodo y **Boolean 1** en el segundo nodo. Si el nombre de la etiqueta no es correcto, use la herramienta de operaciones para seleccionar el elemento correcto.



- Añada la función Increment al diagrama de bloques.
- Cablee la salida **Numeric** de la función Unbundle By Name en la entrada de la función Increment. Esta función suma 1 al valor de **Numeric**.



- Añada la función Not al diagrama de bloques.
- Cablee la salida **Boolean 1** de la función Unbundle By Name en la entrada **x** de la función Not. Esta función devuelve la lógica opuesta del valor de **Boolean**.



- Añada la función Bundle by Name al diagrama de bloques.
 - Cablee **Cluster** a la entrada input cluster.
 - Cambie el tamaño de esta función para tener dos terminales de entrada.
 - Seleccione **Numeric** en el primer nodo y **Boolean 1** en el segundo nodo. Si el nombre de la etiqueta no es correcto, use la herramienta de operaciones para seleccionar el elemento correcto.
 - Cablee la salida de la función Increment a Numeric.
 - Cablee la salida de la función Not a Boolean 1.
 - Cablee la salida de la función Bundle By Name al indicador Modified Cluster.
15. Añada una función de espera para facilitar al procesador el tiempo para completar otras tareas.



- Añada la función Wait Until Next ms Multiple al diagrama de bloques.
 - Haga clic con el botón derecho en el terminal **millisecond multiple** de la función Wait Until Next ms Multiple.
 - Seleccione **Create»Constant** en el menú contextual.
 - Escriba 100 en la constante.
16. Complete el diagrama de bloques y cablee los objetos como en la figura 5-23.
17. Guarde el VI.

18. Abra el panel frontal.
19. Ejecute el VI.
20. Escriba distintos valores en **Cluster** y observe cómo los valores introducidos en **Cluster** afectan a los indicadores **Modified Cluster** y **Small Cluster**. ¿Es el comportamiento que esperaba?
21. Haga clic en el botón **Stop** cuando esté listo.
22. Cambie el orden de clusters de **Modified Cluster**. Ejecute el VI. ¿Cómo afectó el cambio de orden al comportamiento?
23. Cierre el VI. No guarde los cambios.

Fin del ejercicio 5-2

C. Definiciones tipo

Puede usar definiciones tipo para definir clusters y arrays personalizados.

Controles personalizados

Use controles e indicadores personalizados para ampliar el conjunto de objetos disponibles del panel frontal. Puede crear componentes de interfaz de usuario personalizados para aplicaciones cuyo aspecto de controles e indicadores difiera del de los integrados en LabVIEW. Puede guardar un control o indicador personalizado que creó en un directorio o LLB y usar el control o indicador personalizado en otros paneles frontales. También puede crear un icono para el control o indicador personalizado y añadirlo a la paleta **Controls**.

Consulte el tema *Creating Custom Controls, Indicators, and Type Definitions* de la *Ayuda de LabVIEW* para obtener información adicional acerca de crear y usar controles personalizados y definiciones tipo.

Use la ventana del Editor de Controles para personalizar controles e indicadores. Por ejemplo, puede cambiar el tamaño, color y posición relativa de los elementos de un control o indicador e importar imágenes en éstos.

Puede ver la ventana del Editor de Controles de estos modos:

- Haga clic con el botón derecho en un control o indicador del panel frontal y seleccione **Advanced»Customize** en el menú contextual.
- Use la herramienta de posicionamiento para seleccionar un control o indicador del panel frontal y seleccionar **Edit»Customize Control**.
- Use el cuadro de diálogo **New**.

El Editor de Controles aparece con el objeto del panel frontal seleccionado en su ventana. El Editor de Controles tiene dos modos: edición y personalización.

La barra de herramientas de la ventana del Editor de Controles indica si está



en el modo de edición o de personalización. La ventana del Editor de



Controles se abre en el modo de edición. Haga clic en el botón **Edit Mode** para cambiar al modo de personalización. Haga clic en el botón **Customize Mode** para volver al modo de edición. Puede cambiar entre modos seleccionando **Operate»Change to Customize Mode** o **Operate»Change to Edit Mode**.

Use el modo de edición para cambiar el tamaño o color de un control o indicador y para seleccionar opciones de su menú contextual, como haría en el modo de edición de un panel frontal.

Use el modo de personalización para realizar grandes cambios en controles e indicadores cambiando cada parte de un control o indicador.

Modo de edición

En el modo de edición puede hacer clic con el botón derecho en el control y manipular sus ajustes como haría en el entorno de programación de LabVIEW.

1	Modo de edición	5	Distribuir objetos
2	Estado de definiciones tipo	6	Cambiar el tamaño de objetos
3	Texto	7	Reordenar objetos
4	Alinear objetos		

Modo de personalización

En el modo de personalización puede mover de sitio cada componente del control. Para ver una lista de lo que puede manipular en el modo de personalización, seleccione **Window»Show Parts Window**.

1	Modo de edición	5	Distribuir objetos
2	Estado de definiciones tipo	6	Cambiar el tamaño de objetos
3	Texto	7	Reordenar objetos
4	Alinear objetos		

Una forma de personalizar un control es cambiar su estado de definición tipo. Puede guardar un control como un control, una definición tipo o una definición tipo estricta, en función de la selección que se vea en el anillo **Type Def. Status**. La opción de control es la misma que un control que seleccionaría de la paleta **Controls**. Puede modificarla del modo que necesite, y cada copia que realice y cambie conservará sus propiedades individuales.

Guardar controles personalizados

Tras crear un control personalizado, puede guardarlo para usarlo después. De forma predeterminada los controles guardados en el disco tienen una extensión `.ctl`.

También puede usar el Editor de Controles para guardar controles con sus propios ajustes predeterminados. Por ejemplo, puede usar el Editor de Controles para modificar los ajustes predeterminados de un gráfico tipo “waveform graph”, guardarlo y después utilizarlo en otros VIs.

Definiciones tipo

Use definiciones tipo y definiciones tipo estrictas para vincular todas las copias de un control o indicador personalizado a un archivo de control o indicador personalizado guardado. Puede realizar cambios en todas las copias del control o indicador personalizado modificando sólo el archivo de control o indicador personalizado, lo cual resulta útil si usa el mismo control o indicador personalizado en varios VIs.

Cuando coloca un control o indicador personalizado en un VI, no existe conexión entre el control o indicador personalizado que guardó y la copia del control o indicador personalizado en el VI. Cada copia de un control o indicador personalizado es independiente del resto. Por lo tanto, los cambios que realice en un control o indicador personalizado no afectan a los VIs que ya utilizan ese control o indicador personalizado. Si desea vincular copias de un control o indicador personalizado con el archivo de control o indicador personalizado, guarde el control o indicador personalizado como una definición tipo o una definición tipo estricta. Todas las copias de una definición tipo o de una definición tipo estricta se vinculan con el archivo original desde donde las creó.

Cuando guarda un control o indicador personalizado como una definición tipo o una definición tipo estricta, cualquier cambio de tipo de dato que realice en la definición tipo o tipo estricta afecta a todas las copias de la definición tipo o tipo estricta en todos los VIs que la utilicen. Asimismo, los cambios de aspecto que realice en una definición tipo estricta afectan a todas las copias de la definición tipo estricta del panel frontal.

Las definiciones tipo identifican el tipo de dato correcto para cada copia de un control o indicador personalizado. Cuando cambia el tipo de dato de una definición tipo, se actualizarán automáticamente todas las copias de la definición tipo. En otras palabras, el tipo de dato de las copias de la definición tipo cambia en cada VI en el que se haya usado. No obstante, como las definiciones tipo identifican sólo el tipo de dato, sólo se actualizarán los valores que formen parte del tipo de dato. Por ejemplo, en controles numéricos, el rango de datos no forma parte del tipo de dato. Por lo tanto, las definiciones tipo de controles numéricos no definen el rango de

datos para las copias de las definiciones tipo. Asimismo, como los elementos de los controles de anillo no definen el tipo de dato, los cambios en los nombres de los elementos de un control de anillo en una definición tipo no cambian los nombres de los elementos en las copias de la definición tipo. Sin embargo, si cambia los nombres de los elementos de la definición tipo para un control de tipo enumerado, las copias se actualizan porque los nombres del elemento forman parte del tipo de dato. Una copia de una definición tipo puede tener su propio título, etiqueta, descripción, ayuda, valor predeterminado, tamaño, color o estilo de control o indicador único, como en el caso de un mando en lugar de un control deslizante.

Si cambia el tipo de dato de una definición tipo, LabVIEW convierte el antiguo valor predeterminado de las copias de la definición tipo al nuevo tipo de dato, si es posible. LabVIEW no puede conservar el valor predeterminado de la copia si el tipo de dato cambia a un tipo incompatible, como sustituir un control o indicador numérico por un control o indicador de cadena de caracteres. Cuando cambia el tipo de dato de una definición tipo a un tipo de dato incompatible con la definición tipo anterior, LabVIEW configura el valor predeterminado de copias como el valor predeterminado para el nuevo tipo de dato. Por ejemplo, si cambia una definición tipo de un tipo numérico a un tipo de cadena de caracteres, LabVIEW sustituye cualquier valor predeterminado asociado con el antiguo tipo de dato numérico por cadenas de caracteres vacías.

Definiciones tipo estrictas

Una definición tipo estricta fuerza que la copia sea idéntica a la definición de tipo estricta, excepto en el título, etiqueta, descripción, ayuda y valor predeterminado. Al igual que las definiciones tipo, el tipo de dato de una definición tipo estricta permanece invariable en cualquier lugar donde use la definición tipo estricta. Las definiciones tipo estrictas también definen otros valores, como pueden ser la comprobación de rango en controles numéricos y los nombres de los elementos en controles de anillo.

No puede impedir que una copia de una definición tipo estricta se actualice automáticamente, a menos que elimine el vínculo entre la copia y la definición tipo estricta.

Las definiciones tipo y las definiciones tipo estrictas crean un control predeterminado utilizando un cluster de varios controles. Si necesita añadir un nuevo control y pasar un nuevo valor a cada subVI, puede añadir el nuevo elemento al control cluster del control personalizado. Esto evita tener que añadir el nuevo control al panel frontal de cada subVI y realizar un nuevo cableado y nuevos terminales.

Ejercicio 5-3 Definición tipo

Objetivo

Explore las diferencias entre una definición tipo y una definición tipo estricta.

Descripción

1. Abra un VI nuevo.
2. Cree un control personalizado con un estado de definición tipo estricta.
 - Añada un control numérico a la ventana del panel frontal.
 - Haga clic con el botón derecho en el control y seleccione **Advanced»Customize** en el menú contextual. Se abrirá el Editor de Controles.
 - Seleccione **Strict Type Def.** en el menú desplegable **Control Type**.
 - Haga clic con el botón derecho en el control numérico y seleccione **Representation»Unsigned Long** en el menú contextual.
 - Seleccione **File»Save**.
 - Llame al control U32 `Numeric.ctl` en el directorio `<Exercises>\LabVIEW Basics I\Type Definition`.
 - Cierre la ventana del Editor de Controles.
 - Haga clic en **Yes** cuando se le pregunte si desea sustituir el control original.
3. Explore el valor numérico personalizado definido estrictamente.
 - Haga clic con el botón derecho en el control numérico y seleccione **Properties** en el menú contextual. Observe que las únicas opciones disponibles son **Appearance**, **Documentation** y **Key Navigation**. La definición tipo estricta define el resto de propiedades.
 - Haga clic en **Cancel** para salir del cuadro de diálogo **Properties**.
 - Haga clic con el botón derecho sobre el control numérico de nuevo. Observe que la representación no está disponible en el menú contextual. Observe también que puede abrir la definición tipo o desconectarse de la definición tipo.

4. Cambie el tipo de control a la definición tipo.
 - Haga clic con el botón derecho en el control numérico y seleccione **Open Type Def.** en el menú contextual.
 - Seleccione **Type Def.** en el menú desplegable **Control Type.**
 - Guarde el control.
 - Cierre la ventana del Editor de Controles.
5. Explore el valor numérico personalizado con definición de tipo.
 - Haga clic con el botón derecho en el control numérico y seleccione **Properties** en el menú contextual. Observe que existen más elementos disponibles, como Data Entry y Display Format.
 - Haga clic en **Cancel** para salir del cuadro de diálogo **Properties.**
 - Haga clic con el botón derecho sobre el control numérico de nuevo. Observe que **Representation** está atenuado en el menú contextual porque la definición tipo define el tipo de dato. Observe también que puede elegir actualizar automáticamente con la definición tipo.
6. Añada otra copia del control personalizado a la ventana del panel frontal y desconéctela de la definición tipo.
 - Seleccione **Select a Control** en la **Paleta de Controles.**
 - Seleccione U32 Numeric.ctl en el directorio <Exercises>\LabVIEW Basics I\Type Definition.
 - Haga clic en **OK.**
 - Haga clic con el botón derecho en el nuevo control y seleccione **Disconnect from Type Def** en el menú contextual.
 - Haga clic en **OK.**
 - Haga clic con el botón derecho en el control de nuevo y observe que ahora puede cambiar la **Representation** porque el valor numérico ya no está vinculado a la definición tipo.
7. Cierre el VI tras terminar. No necesita guardar el VI.

Fin del ejercicio 5-3

Autorrevisión: cuestionario

1. Puede crear un array de arrays.
 - a. Verdadero
 - b. Falso
2. Tiene dos arrays cableados a la entrada de un bucle For. El autoindexado está habilitado en ambos túneles. Un array tiene 10 elementos y el segundo tiene cinco. Un valor de 7 se cablea al terminal Count, como se ve en la figura 5-24. ¿Cuál es el valor del indicador Iterations tras ejecutar este VI?

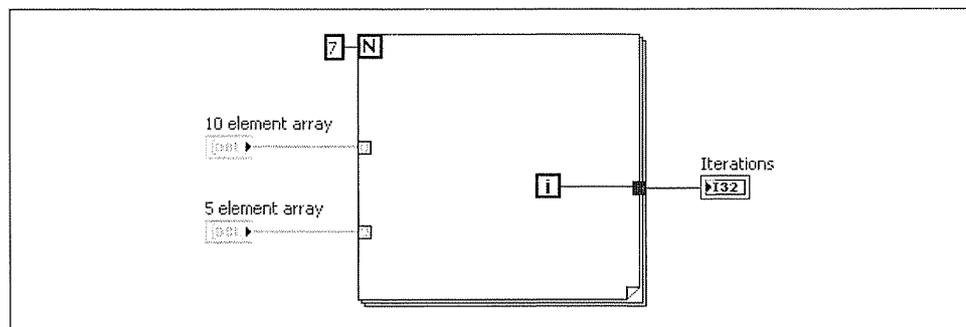


Figura 5-24. ¿Cuál es el valor del indicador Iteration?

3. Usted personaliza un control, selecciona **Control** en el menú desplegable **Control Type** y guarda el control como un archivo `.ctl`. A continuación, usa una copia del control personalizado en la ventana de su panel frontal. Si abre el archivo `.ctl` y modifica el control, ¿cambiará el control en la ventana del panel frontal?
 - a. Sí
 - b. No
4. Está introduciendo datos que representan un círculo. Los datos del círculo incluyen una posición x , una posición y y un radio. Los tres datos son de doble precisión. En el futuro quizá necesite expandir todas las copias de los datos del círculo para incluir el color del círculo, representado como un número entero. ¿Cómo representaría el círculo en la ventana de su panel frontal?
 - a. Tres controles distintos para las dos posiciones y el radio.
 - b. Un cluster que contiene todos los datos.
 - c. Un control personalizado que contiene un cluster.
 - d. Una definición tipo que contiene un cluster.
 - e. Un array con tres elementos.

Autorrevisión: respuestas al cuestionario

1. Puede crear un array de arrays.
 - a. Verdadero
 - b. Falso**
2. Tiene dos arrays cableados a la entrada de un bucle For. El autoindexado está habilitado en ambos túneles. Un array tiene 10 elementos y el segundo tiene cinco. Un valor de 7 se cablea al terminal Count, como se ve en la figura siguiente. ¿Cuál es el valor del indicador Iterations tras ejecutar este VI?

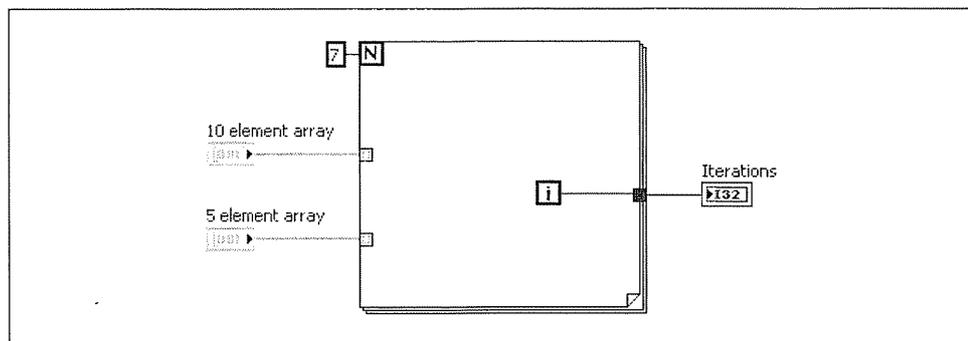


Figura 5-25. ¿Cuál es el valor del indicador Iteration?

Valor de Iterations = 4

LabVIEW no supera el tamaño del array. Esto ayuda a proteger de errores de programación. Las funciones matemáticas de LabVIEW actúan del mismo modo: si cablea un array de 10 elementos a la entrada x de la función Add, y un array de 5 elementos a la entrada y de la función Add, la salida será un array de 5 elementos.

Aunque el bucle For se ejecuta 5 veces, las iteraciones están basadas en cero, por lo que el valor del indicador Iterations es 4.

3. Usted personaliza un control, selecciona **Control** en el menú desplegable **Control Type** y guarda el control como un archivo `.ctl`. A continuación, usa una copia del control personalizado en la ventana de su panel frontal. Si abre el archivo `.ctl` y modifica el control, ¿cambiará el control en la ventana del panel frontal?
 - a. Sí
 - b. No**

4. Está introduciendo datos que representan un círculo. Los datos del círculo incluyen una posición x , una posición y y un radio. Los tres datos son de doble precisión. En el futuro quizá necesite expandir todas las copias de los datos del círculo para incluir el color del círculo, representado como un número entero. ¿Cómo representaría el círculo en la ventana de su panel frontal?
 - a. Tres controles distintos para las dos posiciones y el radio.
 - b. Un cluster que contiene todos los datos.
 - c. Un control personalizado que contiene un cluster.
 - d. Una definición tipo que contiene un cluster.**
 - e. Un array con tres elementos.