
Almacenamiento de datos de medida

Hasta el momento ha aprendido a adquirir datos y visualizarlos, pero el almacenamiento de los datos suele ser muy importante en cualquier proyecto. En esta lección aprenderá a almacenar datos.

Temas

- A. Comprender la E/S de ficheros
- B. E/S de ficheros de alto nivel
- C. Comprender la E/S de ficheros de bajo nivel

A. Comprender la E/S de ficheros

La E/S de ficheros escribe datos o lee datos de un fichero.

Una operación típica de E/S de ficheros incluye el siguiente proceso.

1. Cree o abra un fichero. Tras abrir el fichero, un identificador único denominado refnum representará el fichero.

Consulte el tema *Building the Front Panel* de la *Ayuda de LabVIEW* para obtener información adicional acerca de refnums.

2. La función o el VI File I/O lee en el fichero o escribe en él.
3. Cierre el fichero.

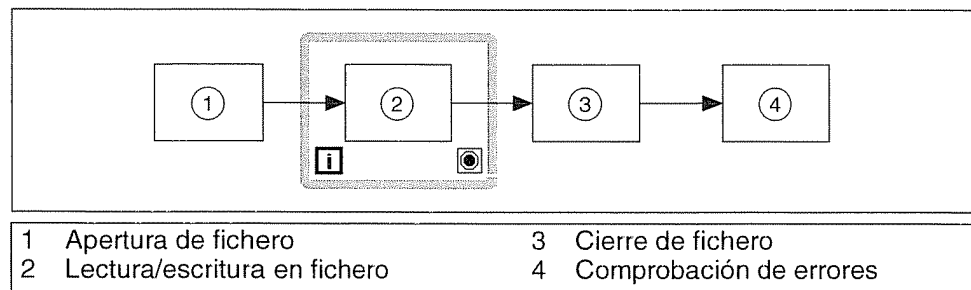


Figura 6-1. Pasos de una operación típica de E/S de ficheros

Formatos de ficheros

LabVIEW puede usar o crear los siguientes formatos de fichero: Binario, ASCII, LVM y TDMS.

- **Binario:** los ficheros binarios son el formato de fichero subyacente del resto de formatos de fichero.
- **ASCII:** un fichero ASCII es un tipo específico de fichero binario y es un estándar que utilizan la mayoría de los programas. Consta de una serie de códigos ASCII. Los ficheros ASCII también se denominan ficheros de texto.
- **LVM:** el fichero de datos de medición de LabVIEW (.lvm) es un fichero de texto delimitado con tabulaciones que puede abrir con una aplicación de hoja de cálculo o un editor de texto. El fichero .lvm incluye información sobre los datos, como la fecha y la hora en que se generaron. Este formato de ficheros es un tipo específico de un fichero ASCII creado para LabVIEW.
- **TDMS:** este formato de ficheros es un tipo específico de un fichero binario creado para productos de National Instruments. Consta de dos ficheros distintos: uno binario que contiene datos y almacena propiedades sobre los datos y un fichero de índice binario que ofrece información consolidada sobre todos los atributos y punteros del fichero binario.

En este curso aprenderá a crear ficheros de texto (ASCII). Use ficheros de texto cuando desee acceder al fichero desde otra aplicación, si el espacio en disco y la velocidad de E/S de ficheros no es crucial, si no necesita realizar lecturas o escrituras de acceso aleatorio y si la precisión numérica no es importante.

Utilizó un fichero LVM en la lección 2, *Explorando LabVIEW*. Para conocer más acerca de ficheros binarios y TDMS, consulte la *Ayuda de LabVIEW* o el curso *LabVIEW Básico II: Desarrollo*.

Directorio LabVIEW Data

Puede utilizar el directorio predeterminado LabVIEW Data para almacenar los ficheros de datos que genere LabVIEW, como ficheros .lvm o .txt. LabVIEW instala el directorio LabVIEW Data en el directorio de ficheros predeterminado de su sistema operativo para ayudarle a organizar y encontrar los ficheros de datos que genere LabVIEW. De forma predeterminada, el VI Express Write LabVIEW Measurement File almacena los ficheros .lvm que genera en este directorio, y el VI Express Read LabVIEW Measurement File lee desde este directorio. La constante Default Data Directory y la propiedad Default Data Directory también devuelven el directorio LabVIEW Data de forma predeterminada.



Seleccione **Tools»Options** y **Paths** en la lista **Category** para especificar un directorio de datos predeterminado distinto. El directorio de datos predeterminado difiere del directorio predeterminado, el cual usted especifica para nuevos VIs, controles personalizados, plantillas de VI u otros documentos LabVIEW que cree.

B. E/S de ficheros de alto nivel

Algunos VIs de E/S de fichero realizan los tres pasos de un proceso de E/S de ficheros: abrir, leer/escribir y cerrar. Si un VI realiza los tres pasos, se denomina VI de alto nivel. Sin embargo, estos VIs quizá no sean tan eficientes como las funciones y los VIs de bajo nivel diseñados para partes concretas del proceso. Si está escribiendo en un fichero en un bucle, use los VIs de E/S de ficheros de bajo nivel. Si está escribiendo en un fichero en una única operación, puede utilizar los VIs de E/S de ficheros de alto nivel.

LabVIEW incluye los siguientes VIs de E/S de ficheros de alto nivel:

- **Write to Spreadsheet File:** convierte un array 2D o 1D de números de doble precisión en una cadena de caracteres de texto y escribe ésta en un nuevo fichero ASCII o anexa los datos de la cadena de caracteres en un fichero existente. También puede transponer los datos. El VI abre o crea el fichero antes de escribir en él y lo cierra después. Puede utilizar este VI para crear un fichero de texto legible para la mayoría de aplicaciones de hojas de cálculo.
- **Read From Spreadsheet File:** lee un número concreto de líneas o filas de un fichero de texto numérico empezando por un desplazamiento de caracteres concreto y convierte los datos en un array 2D de números de doble precisión. El VI abre el fichero antes de leerlo y lo cierra después. Puede utilizar este VI para leer un fichero de hoja de cálculo guardado en formato de texto.
- **Write to Measurement File:** un VI Express que escribe datos en un formato de fichero de medidas basado en texto (.lvmm) o un fichero de medidas binario (.tdms). Puede especificar el método de guardado, el formato de fichero (.lvmm o .tdms), el tipo de encabezado y el delimitador.
- **Read from Measurement File:** un VI Express que lee datos de un formato de fichero de medidas basado en texto (.lvmm) o un fichero de medidas binario (.tdms). Puede especificar el nombre del fichero, el formato del fichero y el tamaño de segmentos.



Consejo Evite colocar los VIs de alto nivel en bucles, ya que los VIs realizan operaciones de apertura y cierre cada vez que los ejecuta.

Ejercicio 6-1 VI Spreadsheet Example

Objetivo

Guardar un array 2D en un fichero de texto para que una aplicación de hoja de cálculo pueda acceder al fichero y explorar cómo mostrar datos numéricos en una tabla.

Descripción

Complete los pasos siguientes para examinar un VI que guarda arrays numéricos en un fichero con un formato al que puede acceder con una hoja de cálculo.

1. Abra el VI Spreadsheet Example situado en el directorio <Exercises>\LabVIEW Basics I\Spreadsheet Example. Esta ventana del panel frontal ya está creada.

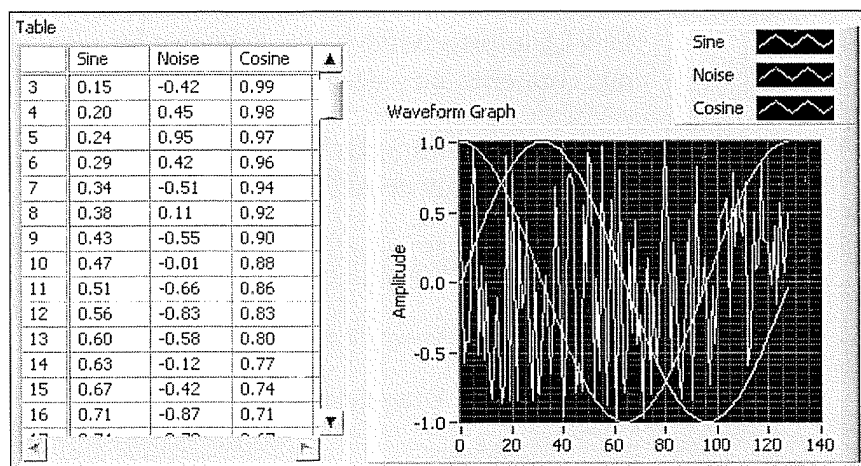


Figura 6-2. Panel frontal del VI Spreadsheet Example

2. Ejecute el VI.

El VI genera un array 2D de 128 filas \times 3 columnas. La primera columna contiene datos para una forma de onda sinusoidal, la segunda para una forma de onda de ruido y la tercera para una forma de onda cosenoidal. El VI traza cada columna en un gráfico tipo “graph” y muestra los datos en una tabla.

3. Cuando aparezca el cuadro de diálogo **Choose file to write**, guarde el fichero como `wave.txt` en el directorio <Exercises>\LabVIEW Basics I\Spreadsheet Example y haga clic en el botón **OK**. Posteriormente examinaremos este fichero.

4. Abra y examine el diagrama de bloques de este VI.

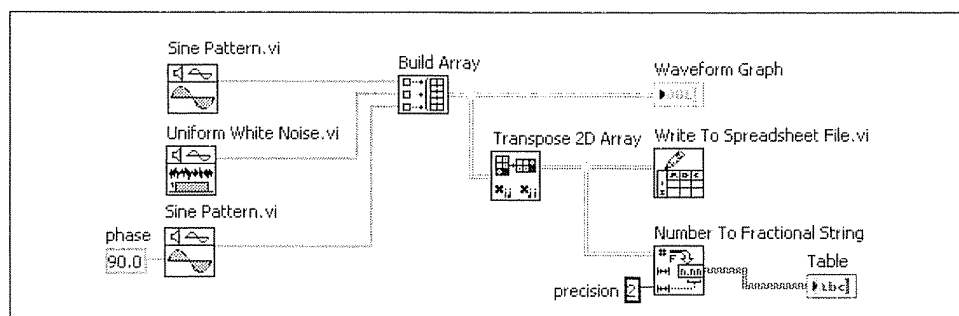


Figura 6-3. Diagrama de bloques del VI Spreadsheet Example



El VI Sine Pattern devuelve un array numérico de 128 elementos que contienen un patrón sinusoidal. La constante 90.0, en la segunda copia del VI Sine Pattern, especifica la fase del modelo sinusoidal o del modelo cosenoidal.



El VI Uniform White Noise devuelve un array numérico de 128 elementos que contienen un patrón de ruido.

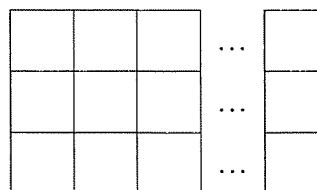


La función Build Array crea el siguiente array 2D a partir del array sinusoidal, array de ruido y array cosenoidal.

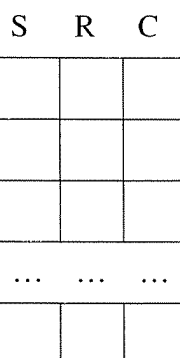
Array sinusoidal

Array de ruido

Array cosenoidal

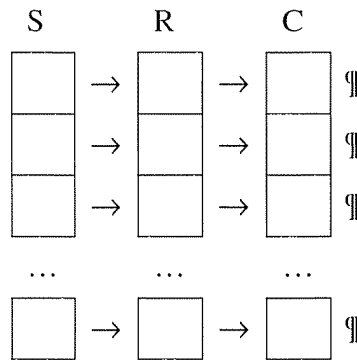


La función Transpose 2D Array reordena los elementos del array 2D para que el elemento $[i, j]$ se convierta en el elemento $[j, i]$, de este modo.





El VI Write To Spreadsheet File convierte el array 2D en una cadena de caracteres de hoja de cálculo y escribe la cadena de caracteres en un fichero. La cadena de caracteres tiene el siguiente formato, donde una flecha (→) indica una tabulación y un símbolo de párrafo (§) indica el final de un carácter de línea.



La función Number To Fractional String convierte un array de valores numéricos en un array de cadenas de caracteres que se muestra en la tabla.

5. Cierre el VI. No guarde los cambios.



Nota Este ejemplo almacena sólo tres arrays en el fichero. Para incluir más arrays, aumente el número de entradas en la función Build Array.

6. Abra el fichero `wave.txt` con un procesador de texto, una aplicación de hoja de cálculo o un editor de texto y vea su contenido.
 - Abra un procesador de texto, una aplicación de hoja de cálculo o un editor de texto, como el Bloc de notas o WordPad.
 - Abra `wave.txt`. Los datos de la forma de onda sinusoidal aparecen en la primera columna, los datos de la forma de onda aleatoria aparecen en la segunda columna y los datos de la forma de onda cosenoidal aparecen en la tercera columna.
7. Salga del procesador de texto o de la aplicación de hoja de cálculo y vuelva a LabVIEW.

Fin del ejercicio 6-1

C. Comprender la E/S de ficheros de bajo nivel

Los VIs y funciones de E/S de fichero de bajo nivel realizan sólo una parte del proceso de E/S de fichero cada uno. Por ejemplo, existe una función para abrir un fichero ASCII, otra para leerlo y otra para cerrarlo. Use funciones de bajo nivel cuando la E/S de fichero esté realizándose en un bucle.

Flujo continuo de datos a disco con funciones de bajo nivel

Puede utilizar funciones de E/S de ficheros para operaciones de flujo continuo de datos a disco, que ahorran recursos de memoria al reducir el número de veces que la función interactúa con el sistema operativo para abrir y cerrar el fichero. El flujo continuo de datos a disco es una técnica para mantener los ficheros abiertos mientras realiza varias operaciones de escritura, por ejemplo, dentro de un bucle.

Consulte el VI Write Datalog File Example en `labview\examples\file\datalog.llb` para ver un ejemplo de uso de VIs de E/S de fichero para operaciones de flujo continuo de datos a disco.

Si cablea un control de ruta o una constante a la función **Write to Text File**, la función **Write to Binary File** o el VI **Write To Spreadsheet File**, añade el retardo de abrir y cerrar el fichero cada vez que se ejecuta la función o el VI. Los VIs pueden ser más eficientes si evita abrir y cerrar los mismos ficheros con frecuencia.

Para evitar abrir y cerrar el mismo fichero, debe pasar un refnum al fichero en el bucle. Cuando abre un fichero, dispositivo o conexión de red, LabVIEW crea un refnum asociado a ellos. Todas las operaciones que realice en ficheros, dispositivos o conexiones de red abiertos usan los refnums para identificar cada objeto.

Los ejemplos de la figura 6-4 y 6-5 muestran las ventajas de usar el flujo continuo de datos a disco. En la figura 6-4, el VI debe abrir y cerrar el fichero durante cada iteración del bucle. La figura 6-5 usa el flujo continuo de datos a disco para reducir el número de veces que debe interactuar el VI con el sistema operativo para abrir y cerrar el fichero. Al abrir el fichero una vez antes de que empiece el bucle y cerrarlo cuando termina el bucle, ahorra dos operaciones de fichero en cada iteración del bucle.

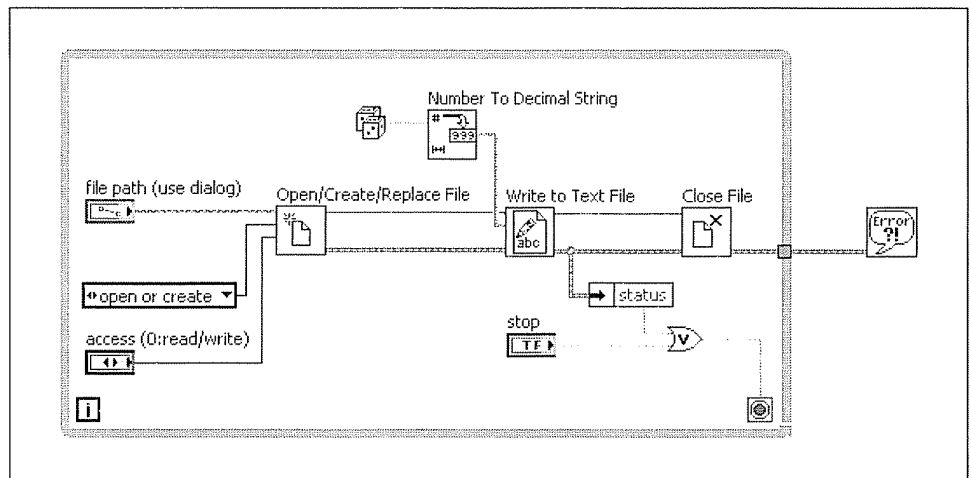


Figura 6-4. Ejemplo sin flujo continuo de datos a disco

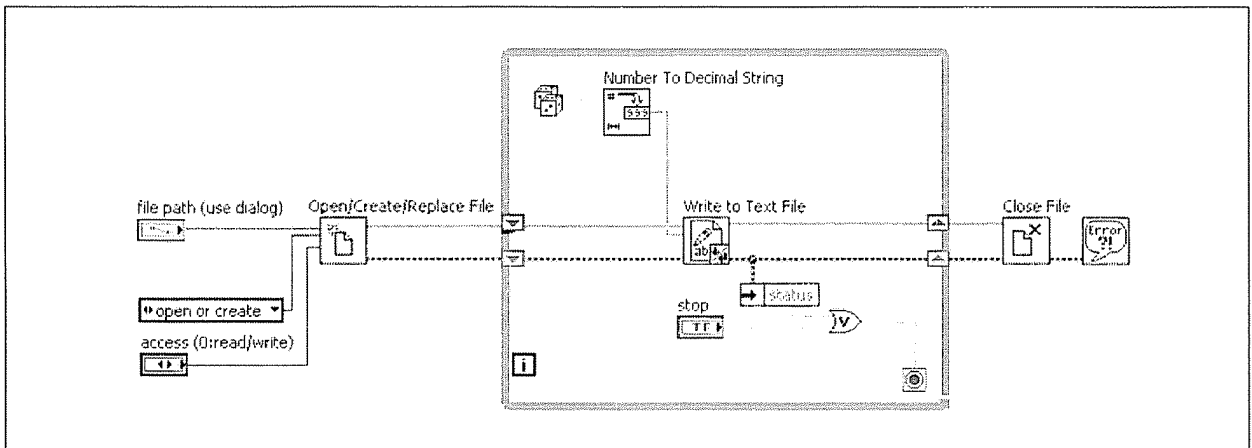


Figura 6-5. Ejemplo con flujo continuo de datos a disco



Ejercicio 6-2 VI Temperature Log

Objetivo

Modificar un VI para crear un fichero ASCII usando el flujo continuo de datos a disco.

Descripción

Ha recibido un VI que representa la temperatura actual y la media de las tres últimas temperaturas. Modifique el VI para registrar la temperatura actual en un fichero ASCII.

Implementación

1. Si dispone de hardware, siga las instrucciones de la columna **Con hardware instalado**. De lo contrario, siga las instrucciones de la columna **Sin hardware instalado**.

Con hardware instalado	Sin hardware instalado
Abra Temperature Multiplot.vi en el directorio <Exercises>\LabVIEW Basics I\Temperature Multiplot.	Abra Temperature Multiplot.vi en el directorio <Exercises>\LabVIEW Basics I\No Hardware Required\Temperature Multiplot.
Seleccione File»Save As y llame al VI Temperature Log.vi en el directorio <Exercises>\LabVIEW Basics I\Temperature Log.	Seleccione File»Save As y llame al VI Temperature Log.vi en el directorio <Exercises>\LabVIEW Basics I\No Hardware Required\Temperature Log.

En las instrucciones inferiores, modificará el diagrama de bloques para que se asemeje al de la figura 6-6.

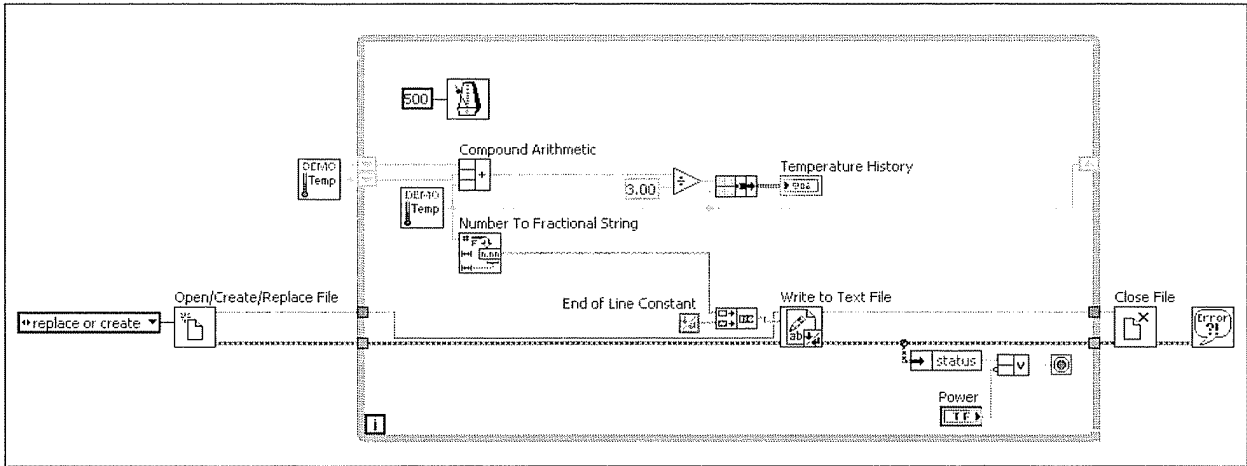


Figura 6-6. Diagrama de bloques del VI Temperature Log

2. Cambie el tamaño del bucle While a fin de añadir espacio para las funciones de E/S de ficheros.
3. Cree un fichero o sustituya uno existente para el registro de datos.



- Añada la función Open/Create/Replace File a la izquierda del bucle While.
- Haga clic con el botón derecho en la entrada **operation** de la función Open/Create/Replace File y seleccione **Create»Constant**.
- Seleccione **replace or create** en la constante enumerada que aparece.

4. Escriba los datos de temperatura en el fichero, añadiendo una constante End of Line a cada dato.



- Añada una función Number to Fractional String dentro del bucle While.



- Añada una constante End of Line dentro del bucle While.



- Añada una función Concatenate Strings dentro del bucle While.



- Añada una función Write to Text File dentro del bucle While.

- Cablee las entradas de la función Write to Text File como en la figura 6-6.

5. Detenga el bucle si ocurre un error o si el usuario desactiva el interruptor Power.

- Elimine el cable que conecta el control Power Boolean al terminal condicional.
- Haga clic con el botón derecho en Loop Condition y seleccione **Stop if True**.



- Añada una función Compound Arithmetic junto al terminal condicional.
 - Con la herramienta de selección, haga clic en el terminal de salida y seleccione **Change Mode»Or**.
 - Haga clic con el botón derecho en el terminal de salida inferior izquierdo de la función Compound Arithmetic y seleccione **Invert**.
 - Cablee el control Power al terminal de entrada inferior izquierdo de la función Compound Arithmetic.



- Añada una función Unbundle By Name al bucle While.
- Cablee el terminal condicional como en la figura 6-6.

6. Cierre el fichero y realice la gestión de errores que puedan haber ocurrido.



- Añada una función Close File a la derecha del bucle While.
- Añada un VI Simple Error Handler a la derecha de la función Close File.



- Termine de cablear el diagrama de bloques como en la figura 6-6.

7. Guarde el VI.

8. Pruebe el VI.

- Ejecute el VI.
- Dé al fichero de texto un nombre y una ubicación.
- Apague el interruptor Power una vez que el VI se haya ejecutado durante varias muestras.
- Abra el fichero de texto creado y explórelo.

9. Cierre el VI y el fichero de texto tras terminar.

Fin del ejercicio 6-2

Ejercicio 6-3 Autoestudio: VI Read VCard

Objetivo

Leer un fichero ASCII en LabVIEW.

Escenario

Los contactos de las tarjetas de presentación de su empresa están almacenados en la Libreta de direcciones de Windows. Debe extraer datos específicos de una tarjeta de presentación individual y mostrarlos en un indicador de texto de LabVIEW.

Diseño

Entradas y salidas

En este VI, la salida aparece en un cuadro de diálogo y las entradas son de un fichero. Por lo tanto, no hay entradas ni salidas necesarias en la ventana del panel frontal de este VI.

Diagrama de flujo

Para comprender cómo diseñar este programa, primero debe ver el fichero de texto que creó la Libreta de direcciones.

1. Abra WordPad en **Inicio»Todos los programas»Accesorios»WordPad**.
2. Seleccione **Archivo»Abrir**.
3. Navegue al directorio <Exercises>\LabVIEW Basics I\
Read VCard.
4. Cambie el tipo de fichero a **All Documents**.
5. Seleccione uno de los ficheros de tarjeta de presentación de este directorio.

Éste es un ejemplo del fichero de texto creado.

Inicio del fichero de texto VCard

```
BEGIN:VCARD☐  
VERSION:2.1☐  
N:McGillicuttey;Heather;Louise;Ms.☐  
FN:Heather Louise McGillicuttey☐  
NICKNAME:Lou☐  
ORG:National Instruments;Internal Affairs☐
```

```
TITLE:President¶
NOTE:I am an imaginary person.¶
TEL;WORK;VOICE:512-555-1212¶
TEL;HOME;VOICE:512-555-1212¶
TEL;CELL;VOICE:512-555-1212¶
TEL;PAGER;VOICE:512-555-1212¶
TEL;WORK;FAX:512-555-1212¶
TEL;HOME;FAX:512-555-1212¶
ADR;WORK:;Corner;11500 N. Mopac
Expressway;Austin;Texas;78759;USA¶
LABEL;WORK;ENCODING=QUOTED-PRINTABLE:Corner=0D=0A11500
N. Mopac Expressway=0D=0AAustin, Texas 78759=0D=0AUSA¶
ADR;HOME:;;111 Easy Street;Austin;Texas;78759;USA¶
LABEL;HOME;ENCODING=QUOTED-PRINTABLE:111 Easy
Street=0D=0AAustin, Texas 78759=0D=0AUSA¶
EMAIL;PREF;INTERNET:heather@ni.com¶
REV:20050818T150422Z¶
END:VCARD
```

Fin del fichero de texto VCard

Observe que el fichero contiene los siguientes caracteres:

- etiquetas de inicio y fin: puede utilizar estas etiquetas para determinar cuándo detener la lectura del fichero.
- fin de línea de caracteres después de cada etiqueta
- dos puntos entre la etiqueta y los datos correspondientes
- un punto y coma que separa las partes de cada elemento de datos.

Toda esta información es de utilidad para escribir un VI que se utilizará para dar formato a los datos.

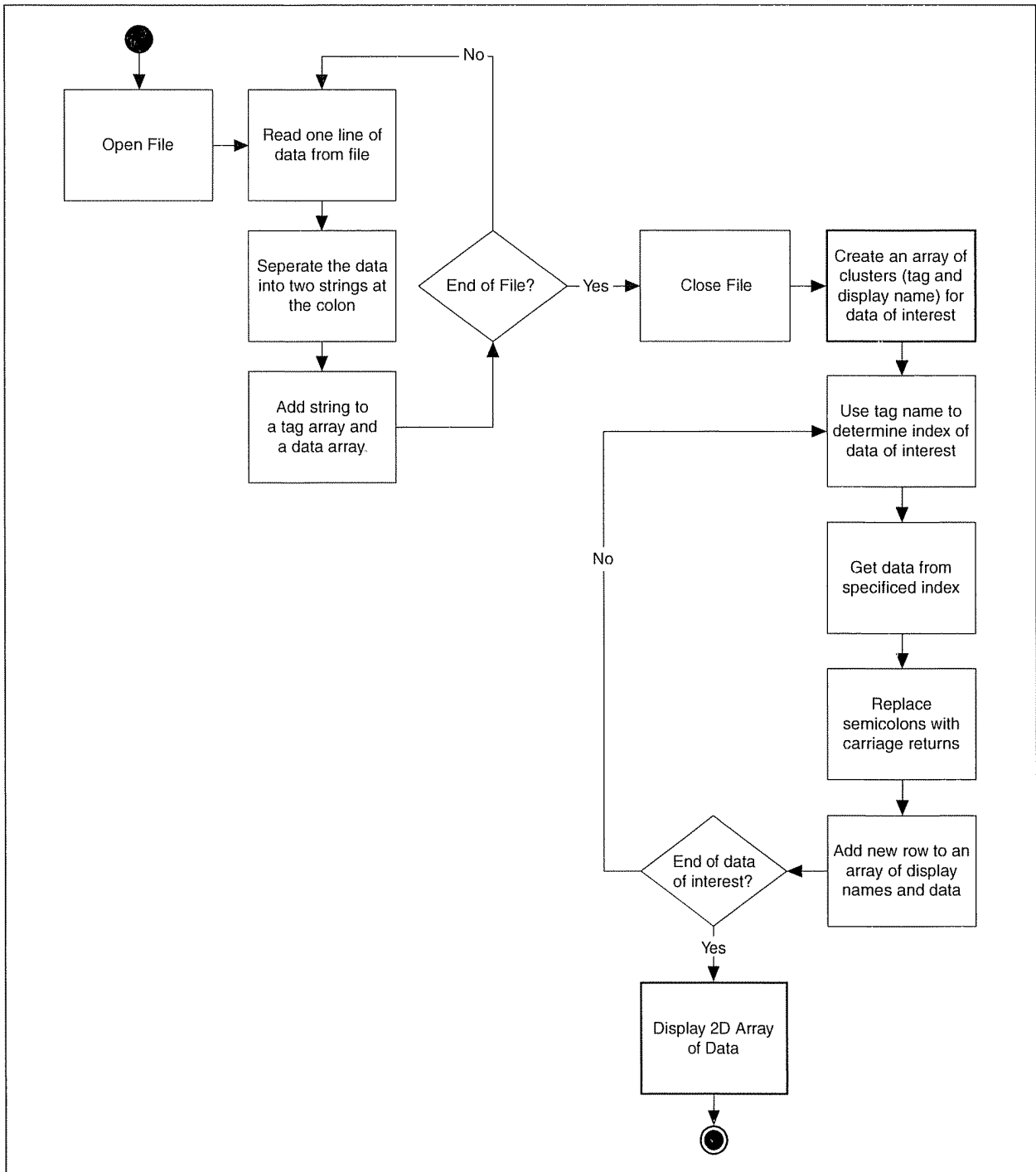


Figura 6-7. Diagrama de flujo del VI Read VCard

Este algoritmo consta de dos bucles. Un bucle lee los datos del fichero de tarjeta de presentación línea por línea. El segundo bucle elige qué datos mostrar, sustituye cada nombre de etiqueta por otro más significativo y sustituye los puntos y comas por caracteres de fin de línea. Los elementos del diagrama de flujo de la figura 6-7 con un borde más grueso representan VIs que ya se han creado para este ejercicio.

Implementación

1. Cree una entrada de tarjeta de presentación en la Libreta de direcciones de Windows.
 - Abra la Libreta de direcciones en **Inicio»Todos los programas»Accesorios»Libreta de direcciones**.
 - Seleccione **Archivo»Nuevo contacto**.
 - Rellene los campos que desee con su información de contacto o con otra imaginaria.
 - Haga clic en **Aceptar** cuando haya terminado.
 - Seleccione **Archivo»Exportar»Tarjeta de presentación (vCard)**.
 - Navegue al directorio <Exercises>\LabVIEW Basics I\
Read VCard.
 - Haga clic en **Guardar**.
 - Seleccione **Archivo»Salir** para cerrar la Libreta de direcciones de Windows.
2. Abra un VI nuevo.
3. Guarde el VI como Read VCard.vi en el directorio <Exercises>\LabVIEW Basics I\
Read VCard.
4. Abra el diagrama de bloques.

Siga las instrucciones inferiores para crear un diagrama de bloques similar al de la figura 6-8. En este diagrama de bloques, se lee la vCard que acaba de crear como un array 2D de cadenas de caracteres. La primera dimensión del array contiene las etiquetas y la segunda contiene los datos.

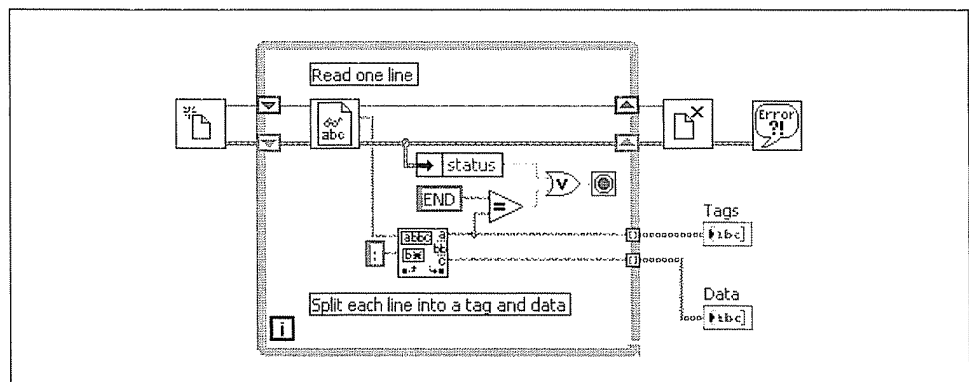


Figura 6-8. Diagrama de bloques del VI Read VCard

5. Abra el fichero de texto.

- Añada una función Open/Create/Replace File al diagrama de bloques. No necesita cablear ninguna de las entradas de esta función para usar los ajustes predeterminados.

6. Lea los datos del fichero de texto.

- Añada un bucle While desde la paleta de **Structures** al diagrama de bloques a la derecha de la función Open/Create/Replace File.



- Añada una función Read From Text File dentro del bucle While.

- Haga clic con el botón derecho en la función Read From Text File y seleccione **Read Lines** para leer el fichero línea por línea.

- Cablee la salida **refnum out** de la función Open/Create/Replace File a la entrada **file (use dialog)** de la función Read From Text File.

- Cablee la salida **error out** de la función Open/Create/Replace File a la entrada **error in** de la función Read From Text File.



- Añada una función Match Pattern tras la función Read from Text File.

- Cablee la salida **text** de la función Read Text a la entrada **string** de la función Match Pattern.

- Haga clic con el botón derecho en el terminal **regular expression** de la función Match Pattern y seleccione **Create»Constant** en el menú contextual.

- Escriba dos puntos (:) en la constante de cadena de caracteres.

- Cablee el terminal **before substring** de la función Match Pattern para crear un túnel de salida desde el bucle While.

- Haga clic con el botón derecho en el túnel de salida y seleccione **Enable Indexing** en el menú contextual.

- Cablee el terminal **after substring** de la función Match Pattern para crear un túnel de salida desde el bucle While.

- Haga clic con el botón derecho en el túnel de salida y seleccione **Enable Indexing** en el menú contextual.

7. Detenga el bucle While si ocurre un error o si se ha llegado al final del fichero.

- Cablee la salida **error out** de la función Read From Text File para crear un túnel de salida en el bucle While.
- Haga clic con el botón derecho en el túnel y seleccione **Replace with Shift Register** en el menú contextual. El cursor cambiará a un registro de desplazamiento, indicando que debe elegir el lado de entrada del registro de desplazamiento.
- Haga clic en el túnel de entrada de error en el lado izquierdo del bucle While para cambiar el túnel de entrada a un registro de desplazamiento.



- Añada una función Unbundle By Name dentro del bucle While.
- Cablee la salida **error out** desde la función Read From Text File a la función Unbundle By Name.



- Añada una función Or al bucle While.
- Cablee el elemento **status** del cluster de error a la entrada **x** de la función Or.



- Añada una función Equal? al bucle While.
- Cablee la salida **before substring** de la función Match Pattern a la entrada **y** de la función Equal?.
- Haga clic con el botón derecho en la entrada **x** de la función Equal?.
- Seleccione **Create»Constant**.
- Escriba **END** en la constante String. Es importante usar sólo letras mayúsculas.
- Cablee la salida de la función Equal? a la entrada **y** de la función Or.
- Cablee la salida de la función Or al terminal condicional del bucle While.



8. Cierre el fichero de texto.

- Cablee la salida **refnum out** de la función Read From Text File para crear un túnel de salida en el bucle While.
- Haga clic con el botón derecho en el túnel y seleccione **Replace with Shift Register**. El cursor cambiará a un registro de desplazamiento, indicando que debe elegir el lado de entrada del registro de desplazamiento.
- Haga clic en el túnel de entrada del refnum en el lado izquierdo del bucle While para sustituir el túnel por un registro de desplazamiento.
- Añada una función Close File a la derecha del bucle While.
- Cablee el túnel de salida del refnum a la entrada **refnum input** de la función Close File.
- Cablee el túnel de salida de error a la entrada **error in** de la función Close File.



9. Muestre los arrays generados en la salida del bucle While.

- Haga clic con el botón derecho en el túnel de salida indexado de la subcadena de caracteres y seleccione **Create»Indicator** en el menú contextual.
- Llame a los indicadores de array Tags.
- Haga clic con el botón derecho en el túnel de salida de array inferior y seleccione **Create»Indicator** en el menú contextual.
- Llame a los indicadores de array Data.

10. Compruebe si hay errores.

- Añada un VI Simple Error Handler a la derecha de la función Close File.
- Cablee la salida **error out** de la función Close File a la entrada **error in** del VI Simple Error Handler.

11. Guarde el VI.

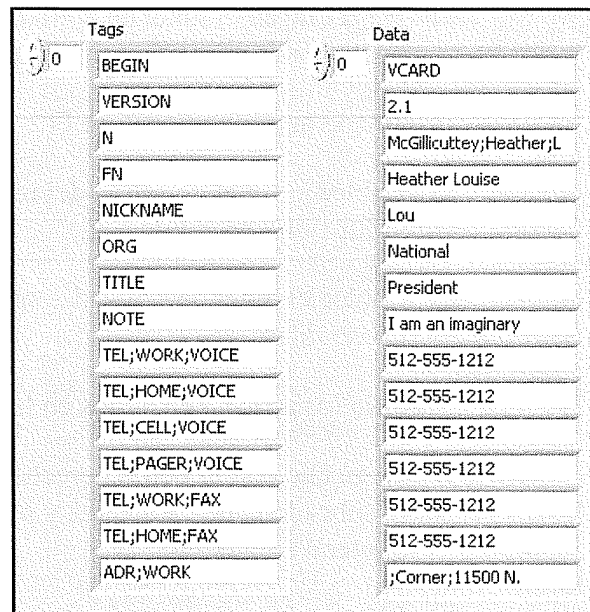
12. Abra la ventana del panel frontal.

13. Expanda los indicadores para mostrar varios elementos de los arrays.

14. Ejecute el VI.

15. Seleccione VCard File cuando se abra el cuadro de diálogo.

La figura 6-9 muestra un ejemplo del panel frontal tras ejecutar este VI. Observe que es muy parecido a abrir el fichero de texto. Los nombres utilizados para cada categoría no resultan muy claros. En el resto de este ejercicio, modifique el VI para que dé formato a los datos, facilitando su lectura al usuario.



Tags	Data
BEGIN	VCARD
VERSION	2.1
N	McGillicutty;Heather;L
FN	Heather Louise
NICKNAME	Lou
ORG	National
TITLE	President
NOTE	I am an imaginary
TEL;WORK;VOICE	512-555-1212
TEL;HOME;VOICE	512-555-1212
TEL;CELL;VOICE	512-555-1212
TEL;PAGER;VOICE	512-555-1212
TEL;WORK;FAX	512-555-1212
TEL;HOME;FAX	512-555-1212
ADR;WORK	;Corner;11500 N.

Figura 6-9. Panel frontal del VI Read VCard sin formato de datos

En los siguientes pasos añadirá la funcionalidad necesaria al diagrama de bloques para dar formato a los datos de los arrays. Para simplificar este proceso, ya se han creado dos VIs. Uno de estos VIs crea un array donde cada elemento de éste es un cluster que contiene una etiqueta y un nombre de sustitución para la misma. El segundo VI abre un cuadro de diálogo que muestra los datos finales en una tabla.

16. Cambie al diagrama de bloques.

17. Elimine los indicadores del array de etiqueta y de datos.

18. Elimine el cable que conecta la función Close File con el Simple Error Handler.

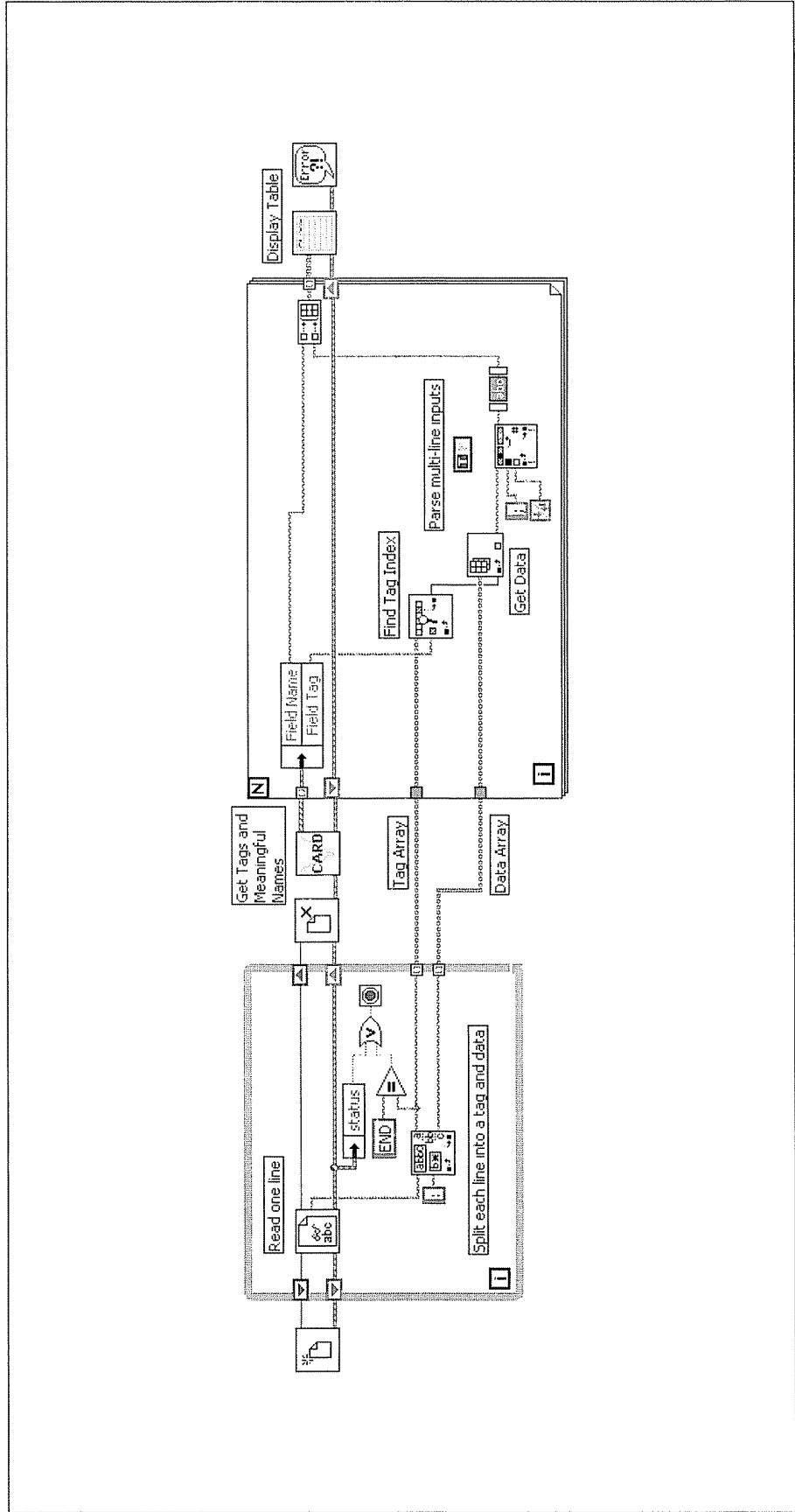


Figura 6-10. Diagrama de bloques del VI Read VCard

19. Aparte el Simple Error Handler. Usará este VI después en el ejercicio.

20. Acceda al array de nombres de etiquetas y de sustitución.

- Coloque el VI Vcard Tags a la derecha de la función Close File. Este VI se encuentra en el directorio <Exercises>\LabVIEW Basics I\Read VCard.



Consejo Use la categoría **Select a VI** de la paleta **Functions** para buscar un VI que no forme parte de la paleta de **Funciones**. Tras haber colocado el VI en el diagrama de bloques, puede hacer doble clic en el VI para abrirlo y examinar su diagrama de bloques.

- Cablee la salida **error out** de la función Close File a la entrada **error in** del VI Vcard Tags.

21. Cree un array 2D que contenga los nombres de sustitución y los datos correspondientes.

- Añada un bucle For a la derecha del VI Vcard Tags.
- Cablee el terminal **Array** del VI Vcard Tags para crear un túnel de entrada en el bucle For. Observe que se ha habilitado automáticamente el indexado.



- Añada una función Unbundle By Name dentro del bucle For.
- Cablee el túnel de entrada indexado a la función Unbundle by Name.
- Expanda la función Unbundle by Name para mostrar dos elementos.



- Añada una función Search 1D Array a la derecha de la función Unbundle by Name.
- Cablee el elemento **Field Tag** de la función Unbundle By Name en la entrada **element** de la función Search 1D Array.
- Cablee el array de etiquetas a la entrada **1D array** de la función Search 1D Array. El array de etiqueta es el túnel de salida indexado superior del bucle While.
- Desactive el indexado en el túnel de entrada del array de etiquetas del bucle For.
- Añada una función Index Array a la derecha de la función Search 1D Array.

- Cablee el array de datos al terminal **array** de la función Index Array. El array de datos es el túnel de salida indexado inferior del bucle While.
- Desactive el indexado en el túnel de entrada del array de datos del bucle For.
- Cablee la salida **index of element** de la función Search 1D Array a la entrada **index** de la función Index Array.
- Añada una función Search and Replace String a la derecha de la función Index Array.
- Cablee la salida **element** del Index Array a la entrada **input string** de la función Search and Replace String.
- Haga clic con el botón derecho en el terminal **search string** de la función Search and Replace String y seleccione **Create»Constant**.
- Escriba un punto y coma (;) en la constante de cadena de caracteres.
- Coloque una constante End of Line bajo la constante de cadena de caracteres.
- Cablee la constante End of Line a la entrada **replace string** de la función Search and Replace String.
- Haga clic con el botón derecho en la entrada **replace all?(F)** de la función Search and Replace String y seleccione **Create»Constant** en el menú contextual.
- Use la herramienta de operaciones para cambiar la constante a True.
- Añada un VI Trim Whitespace a la derecha de la función Search and Replace String.
- Cablee la salida **result string** de la función Search and Replace String a la entrada del VI Trim Whitespace.
- Añada una función Build Array a la derecha del VI Trim Whitespace.
- Cambie el tamaño de la función Build Array para tener dos nodos.
- Cablee el elemento **Field Name** de la función Unbundle By Name al nodo superior de la función Build Array.



- Cablee la salida **trimmed string** del VI Trim Whitespace al nodo inferior de la función Build Array. Ahora tiene un array 2D con descripciones significativas en la primera dimensión y los datos correspondientes en la segunda dimensión.

22. Muestre el array generado.

- Cablee la salida de la función Build Array para crear un túnel de salida en el bucle For.
- Confirme que el túnel de salida está autoindexado.
- Coloque el VI Table Dialog, situado en el directorio <Exercises>\LabVIEW Basics I\Read VCard, a la derecha del bucle For.
- Cablee el túnel de salida indexado a la entrada **Contact Information** del VI Table Dialog.
- Cablee el cluster de error del VI Vcard Tags al VI Table Dialog.
- Sustituya los túneles del cluster de error por registros de desplazamiento.
- Mueva el VI Simple Error Handler a la derecha del VI Table Dialog.
- Cablee el terminal **error out** del VI Table Dialog al terminal **error in** del VI Simple Error Handler.

23. Cambie a la ventana del panel frontal.

24. Guarde el VI.

Prueba

1. Ejecute el VI.
2. Cuando se le indique, navegue al fichero de tarjeta de presentación que creó anteriormente en este mismo ejercicio.
3. Cierre la ventana Table Dialog para detener el VI.
4. Cierre el VI tras terminar.

Fin del ejercicio 6-3

Autorrevisión: cuestionario

1. Su programa de prueba que se ejecuta continuamente registra en un fichero los resultados de todas las pruebas que se realizan en una hora según se van calculando. Si le preocupa la velocidad de ejecución de su programa, ¿debería usar VIs de E/S de ficheros de bajo o alto nivel?
 - a. VIs de E/S de ficheros de bajo nivel
 - b. VIs de E/S de ficheros de alto nivel
2. Si desea ver datos en un editor de texto como el Bloc de notas, ¿qué formato debería usar para guardar los datos?
 - a. ASCII
 - b. TDMS

Autorrevisión: respuestas al cuestionario

1. Su programa de prueba que se ejecuta continuamente registra en un fichero los resultados de todas las pruebas que se realizan en una hora según se van calculando. Si le preocupa la velocidad de ejecución de su programa, ¿debería usar VIs de E/S de fichero de bajo o alto nivel?
 - a. **VIs de E/S de fichero de bajo nivel**
 - b. VIs de E/S de fichero de alto nivel
2. Si desea ver datos en un editor de texto como el Bloc de notas, ¿qué formato debería usar para guardar los datos?
 - a. **ASCII**
 - b. TDMS