

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

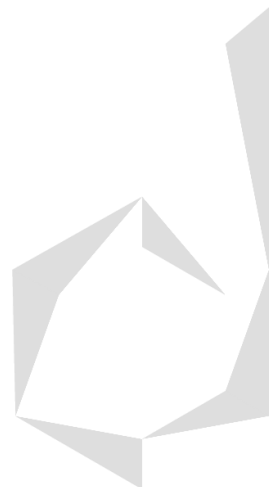
INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Leer un Archivo y Convertirlo
a un String de LabVIEW

Contenido

Introducción Teórica de LabVIEW:.....	2
Introducción al Entorno de LabVIEW:.....	2
Front Panel: Ventana Gris con la Interfaz del Programa	4
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)	4
Front Panel o Block Diagram - Show Context Help: Descripción de Bloques	5
Front Panel y Block Diagram: Navegar de una Ventana a Otra	6
Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel	7
Block Diagram - Highlight Execution: Correr Más Lento el Programa	8
Coertion dot: Conversión Automática de Datos por Parte de LabVIEW	8
Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI	8
Programa: Leer Archivos y Convertirlos a Strings	9
Desarrollo del Programa: Leer un Archivo y Convertirlo a String	9
Block Diagram - Open/Create/Replace File: Abrir, Crear o Reemplazar Archivo	9
Block Diagram - Read From Text File: Leer texto de un Archivo	10
Block Diagram - Close File: Cerrar Archivo Abierto	12
Ejecución del Programa: Leer Archivos y Convertir su Contenido a String	13
Archivo .txt	13
Archivo Página Web HTML	15
Archivo Word Microsoft Office	15



Introducción Teórica de LabVIEW:

LabView sirve para poder usar la computadora como instrumento de medición, monitoreo, control y análisis de procesos y operaciones, esto se hace a través de una frecuencia de muestreo que se relaciona con mediciones de los dispositivos digitales y tiene que ver con la señal de reloj de la tarjeta de desarrollo, indicando cada cuánto tiempo se hará un muestreo de cualquier señal del mundo real.

La diferencia entre los instrumentos virtuales de medición y los reales es más que nada el precio, ya que un osciloscopio cuesta alrededor de \$10,000 y se puede hacer la misma función con LabView y un Arduino, que cuesta alrededor de \$170, además de que es modular, esto implica que se pueden agregar o quitar funcionalidades. La mejor tarjeta de desarrollo para hacer esto es la de NI Instruments, que es la creadora de LabVIEW.

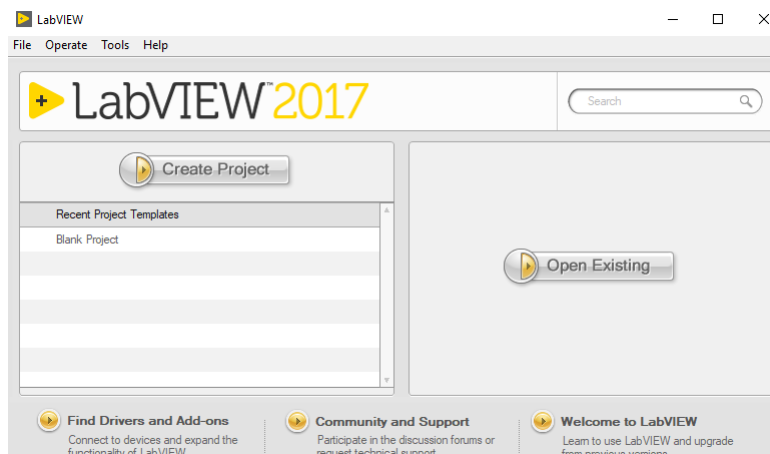
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

La instrumentación virtual es empleada para la gestión de sistemas industriales y muy utilizado en compañías como: Ford, SpaceX, Accenture, Bosch, etc.

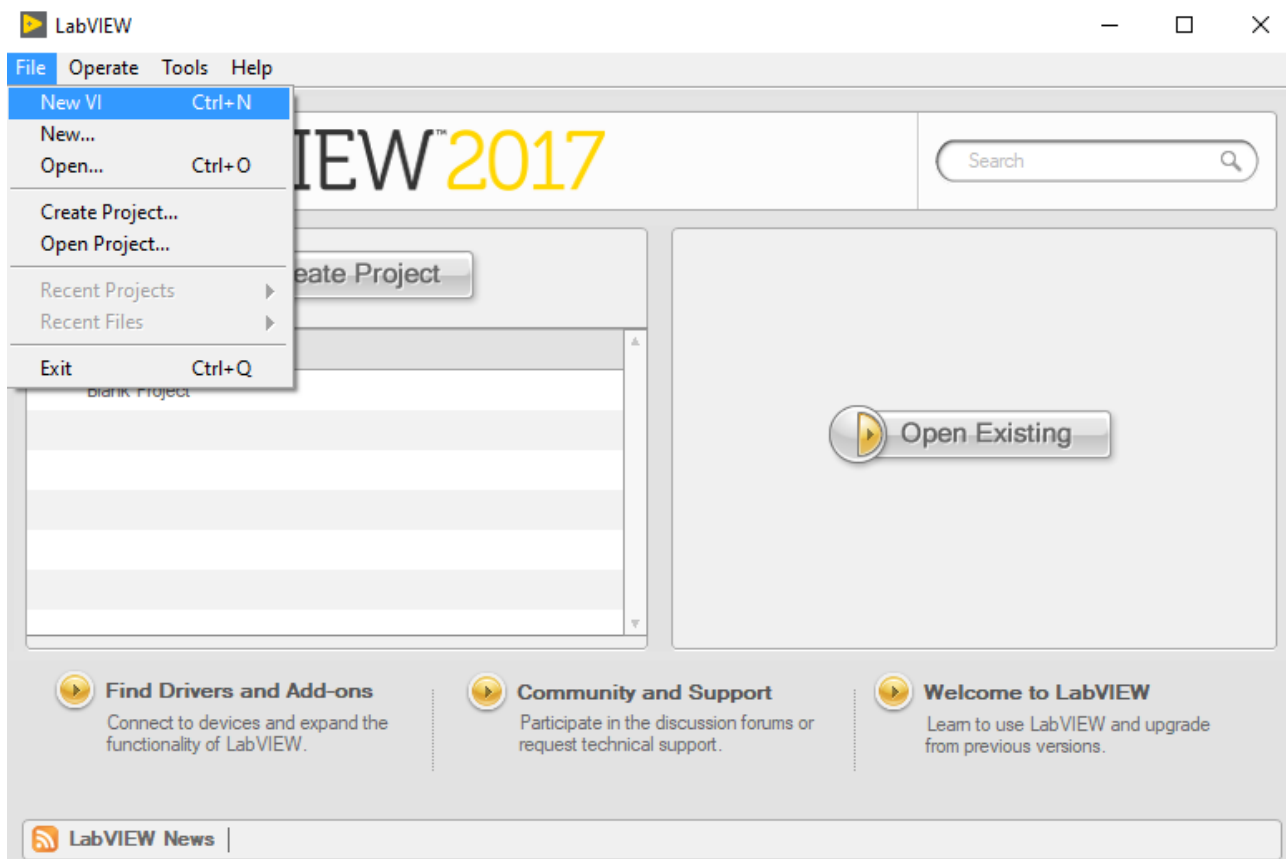


Introducción al Entorno de LabVIEW:

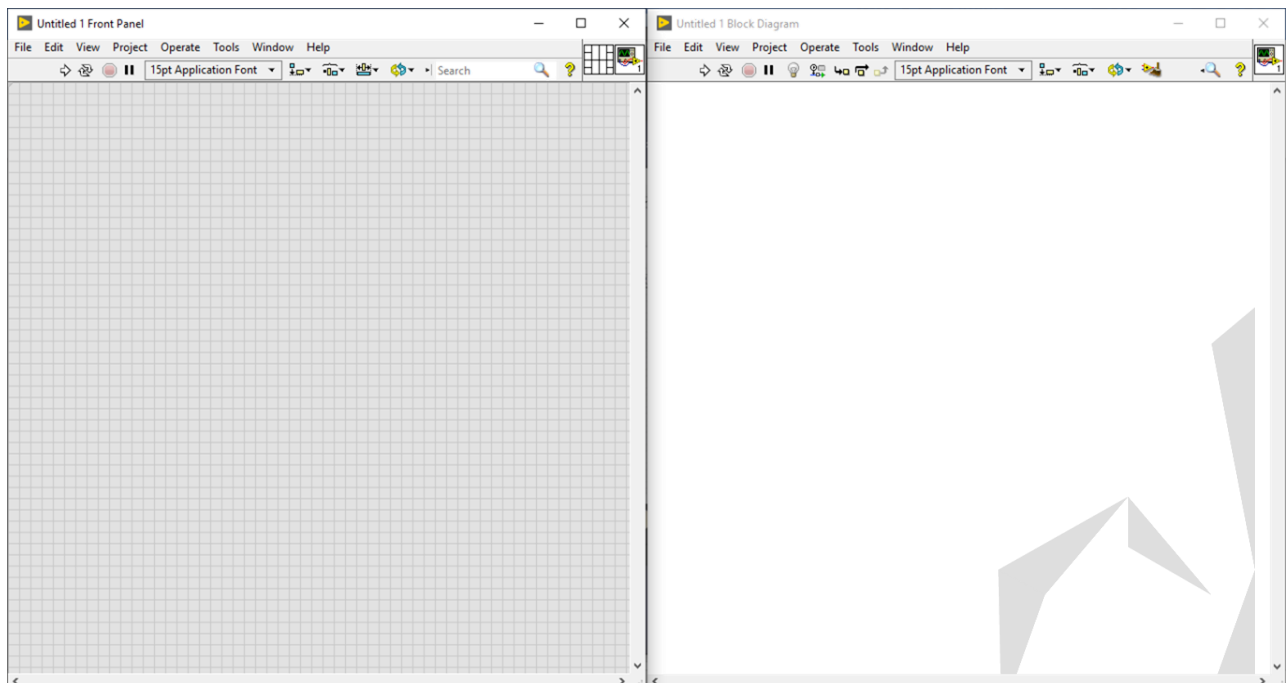
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

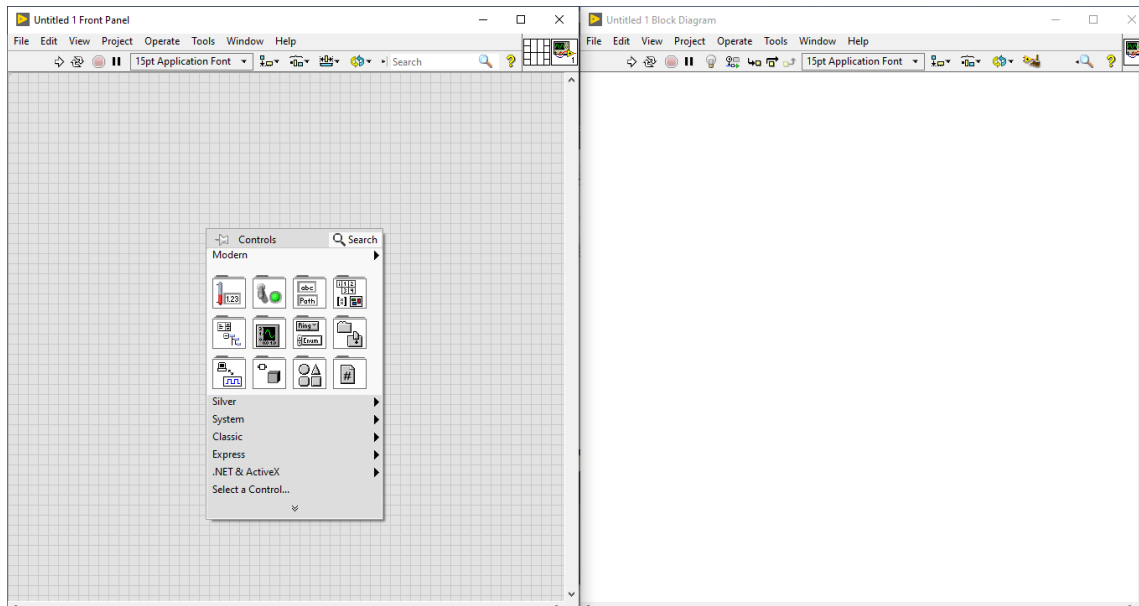


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



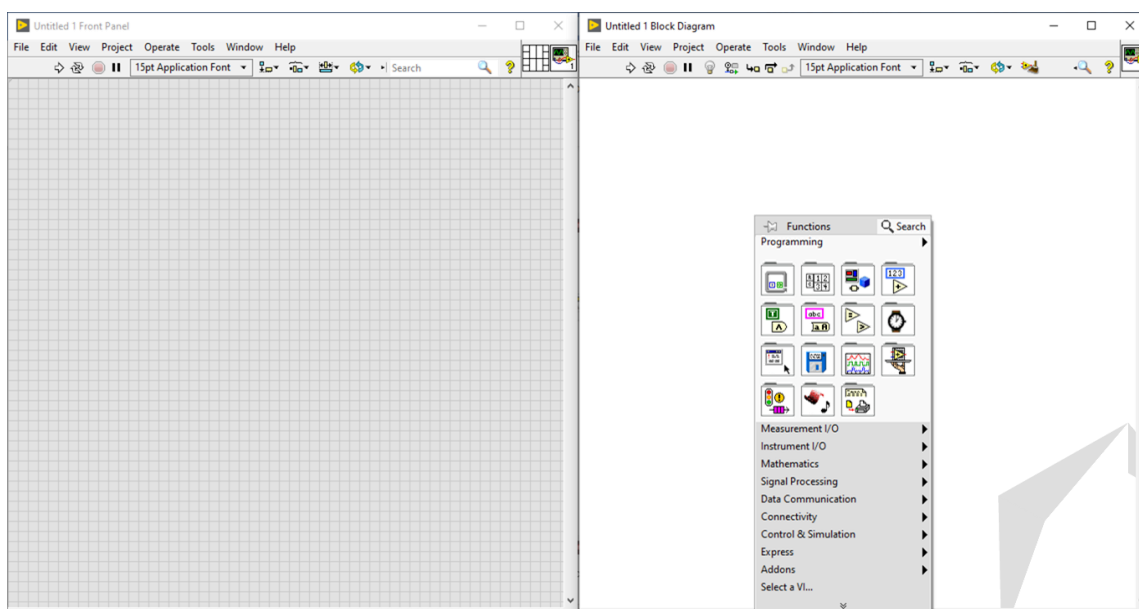
Front Panel: Ventana Gris con la Interfaz del Programa

En la ventana gris llamada **Front Panel**, es donde se observa la interfaz del Programa y se cuenta con el **control palette** que sirve para poder añadir elementos gráficos a la interfaz y aparece dando clic derecho en la pantalla gris. Si no aparece la otra ventana (blanca) por default, se debe seleccionar la opción **Window → Show Block Diagram** y con ello aparecerá.



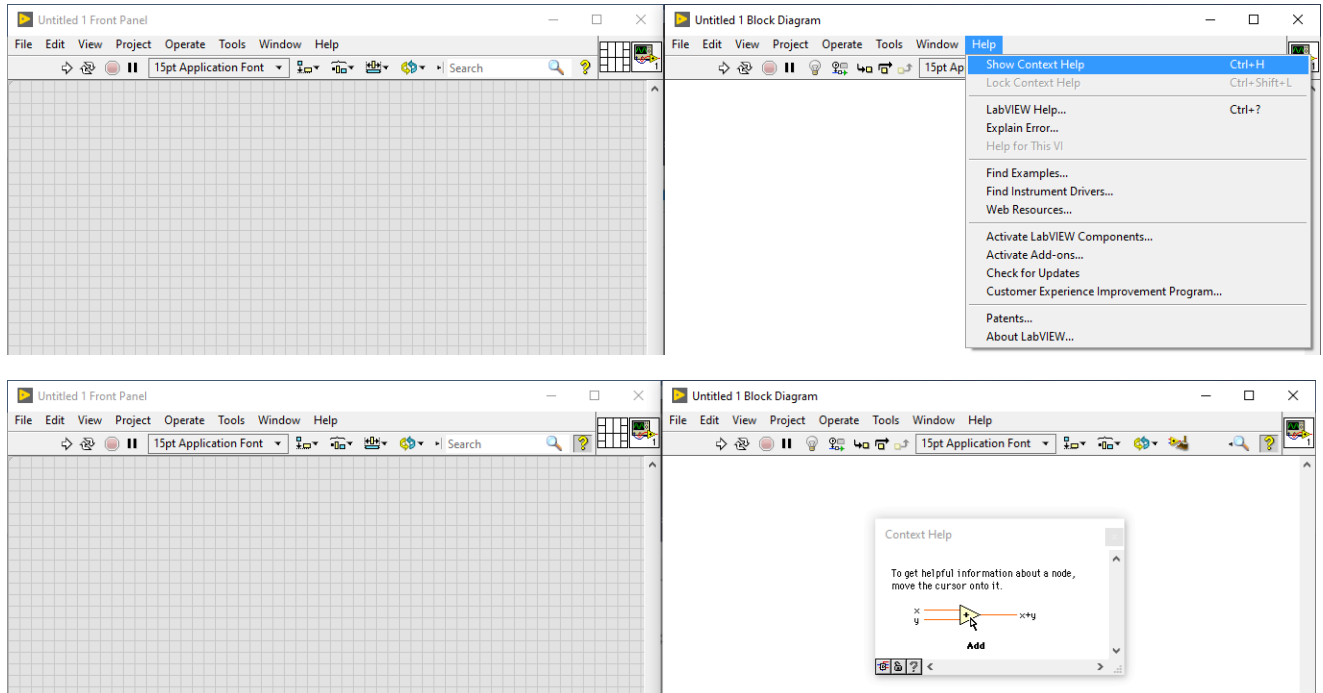
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

En la ventana blanca llamada **Block Diagram** aparece la **paleta de funciones** que sirve para introducir los elementos de programación en forma de bloques que se conectarán entre ellos y describirán la función del programa, aparece dando clic derecho en la pantalla gris. Si no aparece la ventana gris se debe seleccionar la opción **Windows → Show Front Panel** y con ello aparecerá.



Front Panel o Block Diagram - Show Context Help: Descripción de Bloques

Seleccionando la opción de Help → Show Context Help, aparecerá una ventana emergente que explicará las propiedades de los bloques que se puede seleccionar, mostrando una descripción de su función, imágenes explicativas y significado de sus pines de entrada y salida.



Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:

VIs Express, VIs y Funciones

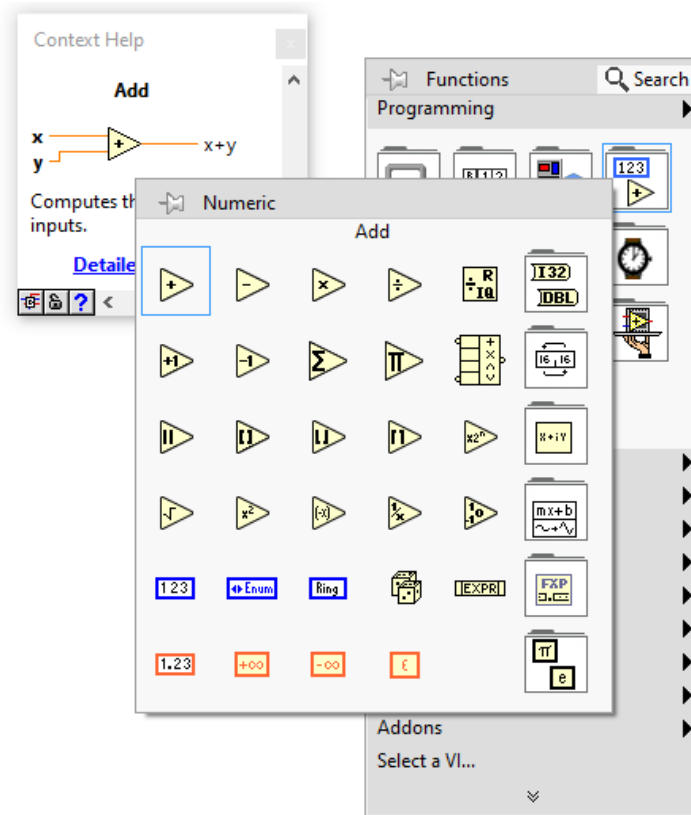


- **VIs Expreso:** VIs interactivos con pagina de dialogo configurable
- **VIs estándar:** VIs modulares y personalizables mediante cableado
- **Funciones:** Elementos fundamentales de operación de LabVIEW; no contiene panel frontal o diagrama de bloque



Función

En un bloque de código, las **terminales que aparezcan en negritas** son las que a fuerza deben estar **conectadas a algo**, las que no estén en negritas no deben estar conectadas a nada forzosamente.

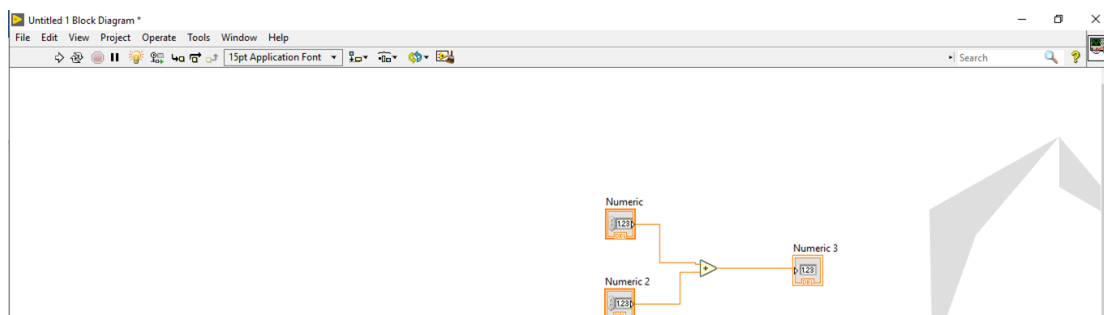


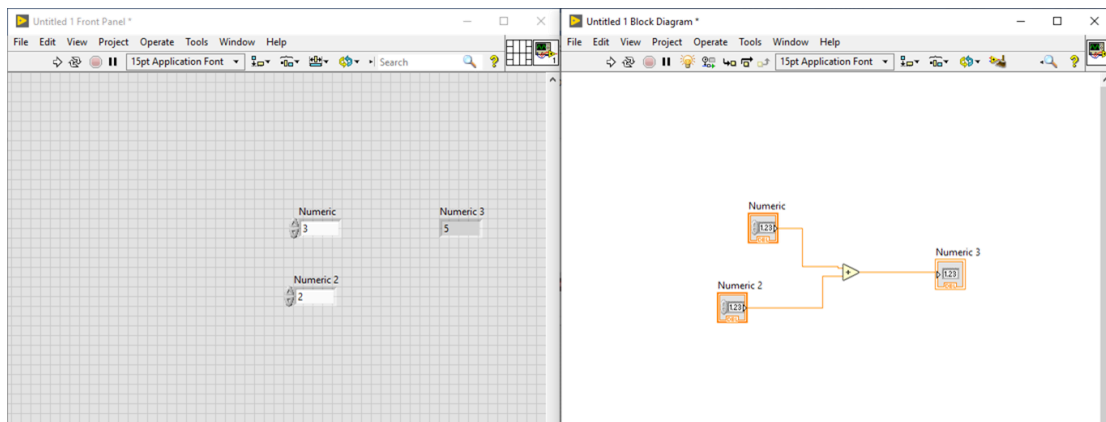
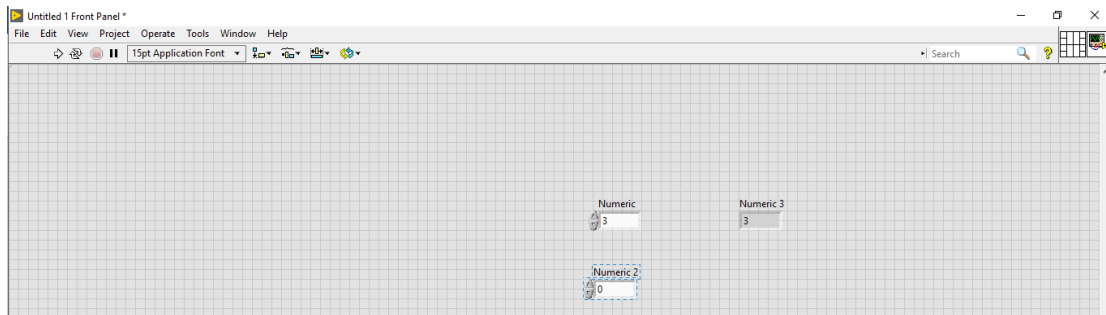
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



Front Panel y Block Diagram: Navegar de una Ventana a Otra

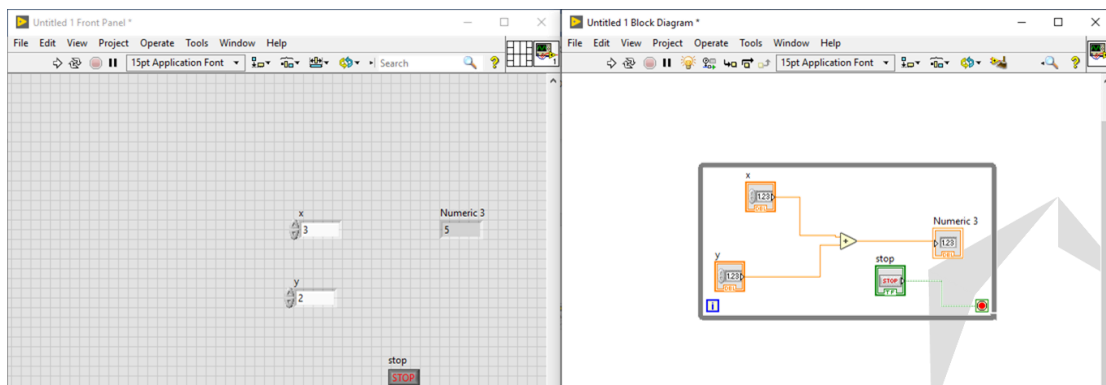
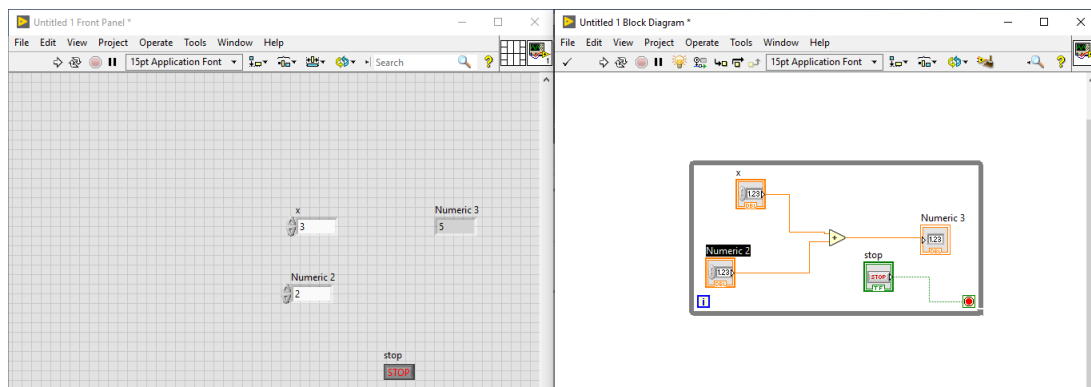
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

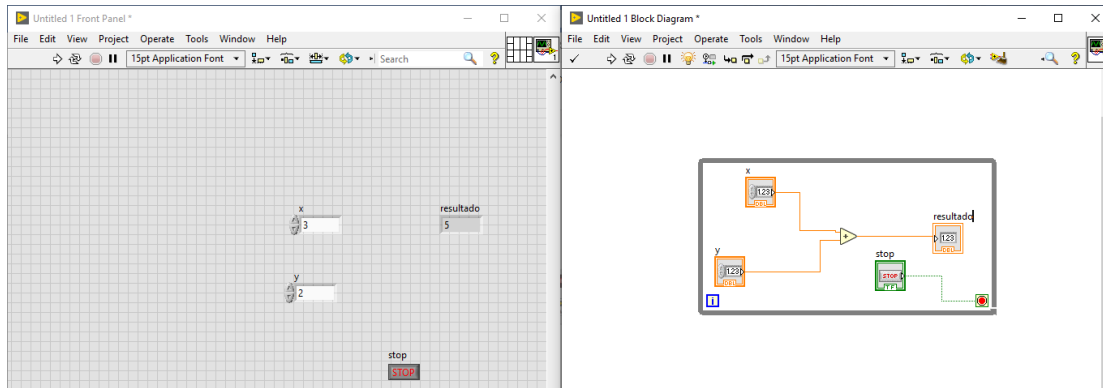




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

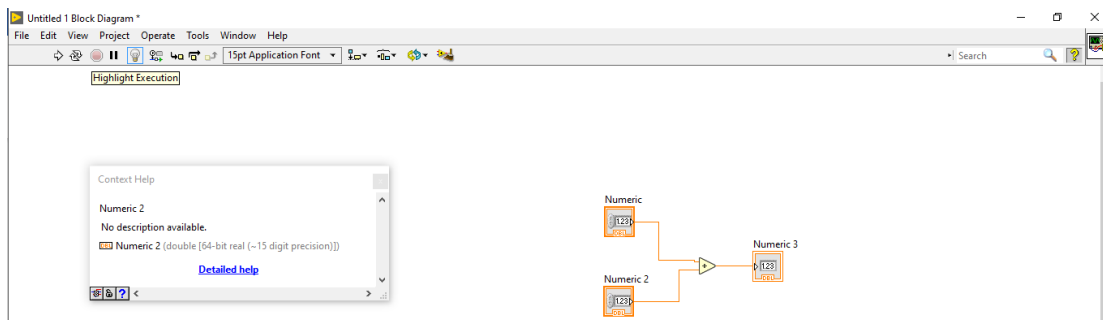
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





Block Diagram - Highlight Execution: Correr Más Lento el Programa

Podemos presionar el foquito del menú superior para ver el funcionamiento de programa de manera más lenta.

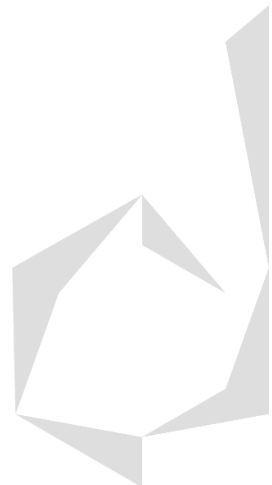


Coertion dot: Conversión Automática de Datos por Parte de LabVIEW

Aparece un punto rojo en la terminal del bloque llamado coercion dot, este lo que me dice es que los tipos de datos en la conexión son distintos, por lo que LabVIEW está forzando una conversión de un tipo de dato a otro, el problema es que en este tipo de conversión yo no sé si se están perdiendo datos, por eso debemos evitar el uso de coercion dots porque usa direcciones de memoria o recursos de la computadora sin que yo tenga control de ellos.

Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI

Con el botón de Clean Up Diagram que se encuentra en la parte superior derecha del Block Diagram se organizan mejor y de forma automática mis elementos.



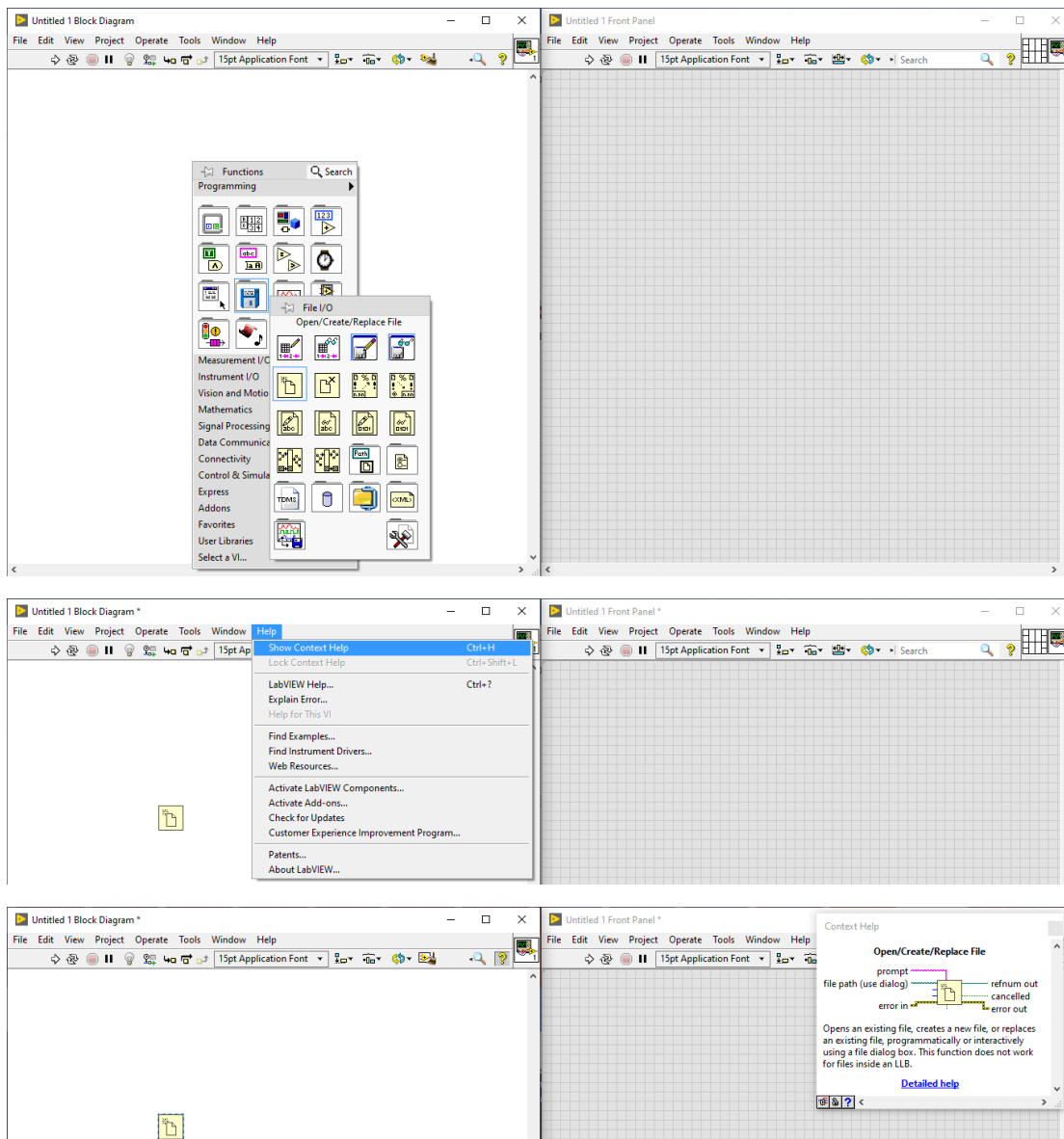
Programa: Leer Archivos y Convertirlos a Strings

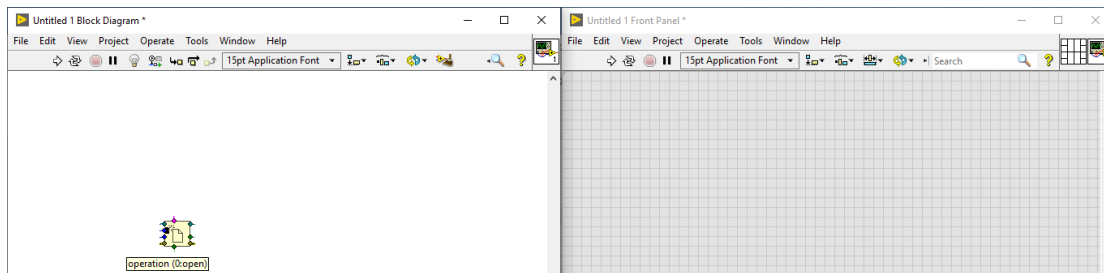
Se leerá un archivo cualquiera, ya sea de Word, PowerPoint, Excel, txt, etc. Para convertir su contenido en String.

Desarrollo del Programa: Leer un Archivo y Convertirlo a String

Block Diagram - Open/Create/Replace File: Abrir, Crear o Reemplazar Archivo

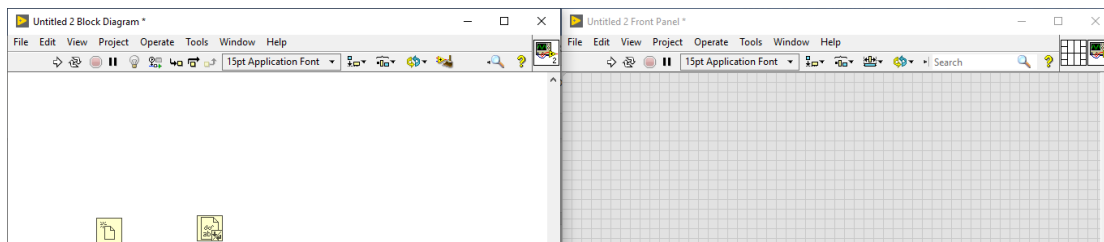
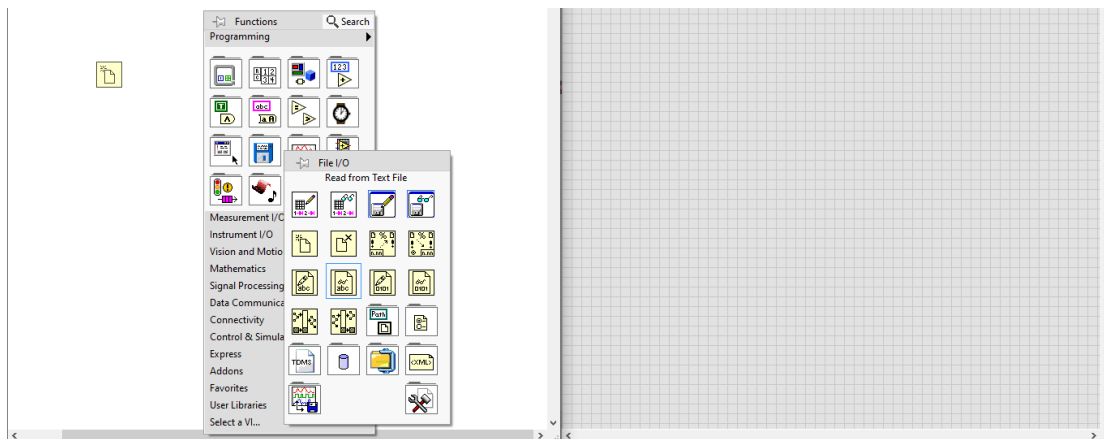
Ahora si vamos a trabajar con archivos, metiendo en ellos un String, ya que a los archivos solo pueden enviarse Strings no números ni otro tipo de datos.



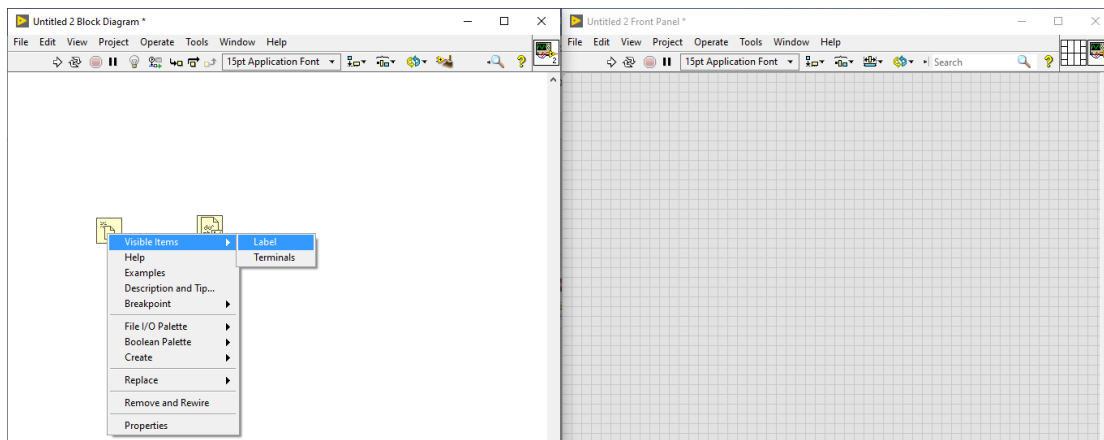


Block Diagram - Read From Text File: Leer texto de un Archivo

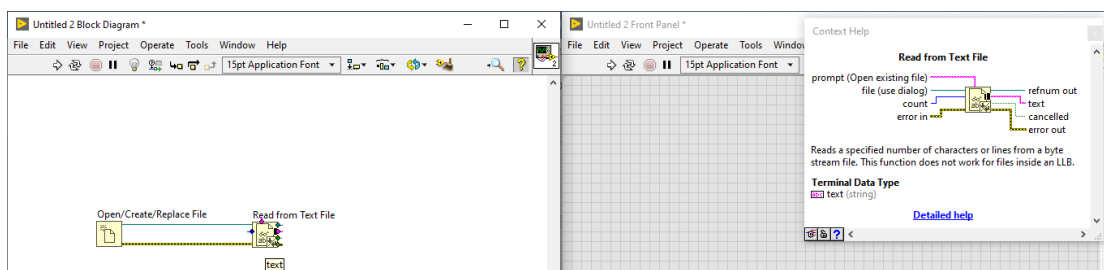
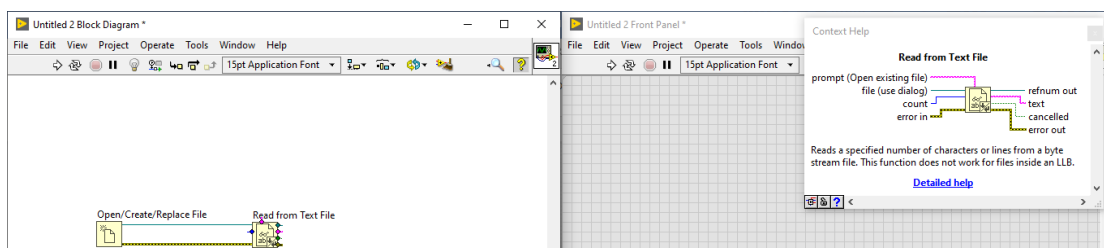
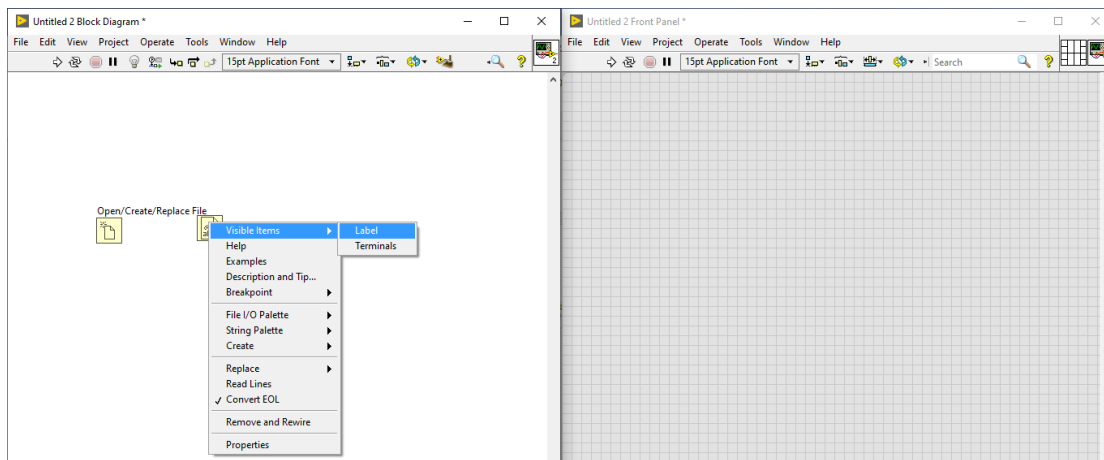
El bloque **Read From Text File** se usa en conjunto con el de **Open/Create/Replace File** para que se pueda leer un archivo.



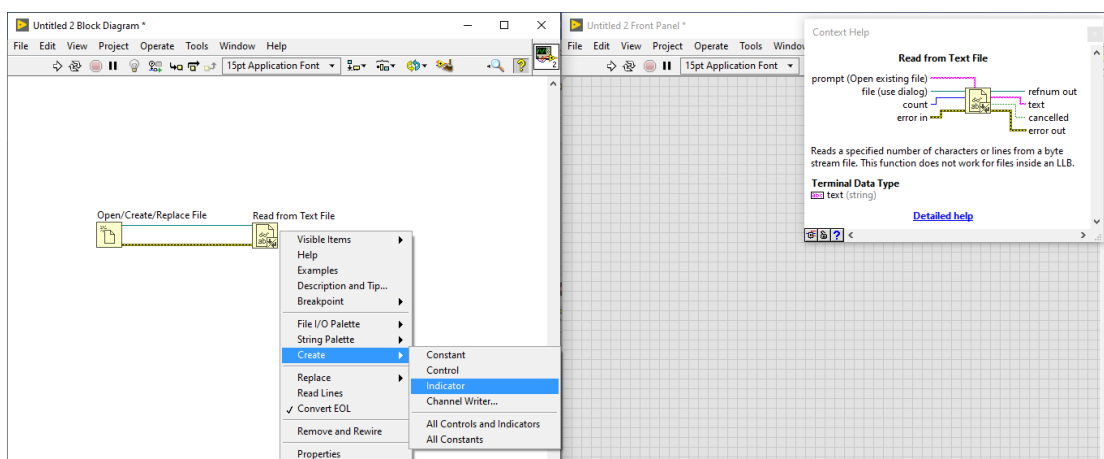
Mostrar nombre del bloque: Clic derecho → Visible Items → Label.



En específico se conecta la salida de refnum out del bloque **Open/Create/Replace File** con la entrada de file (use dialog) del bloque **Read From Text File** y la terminal del error también se conecta.

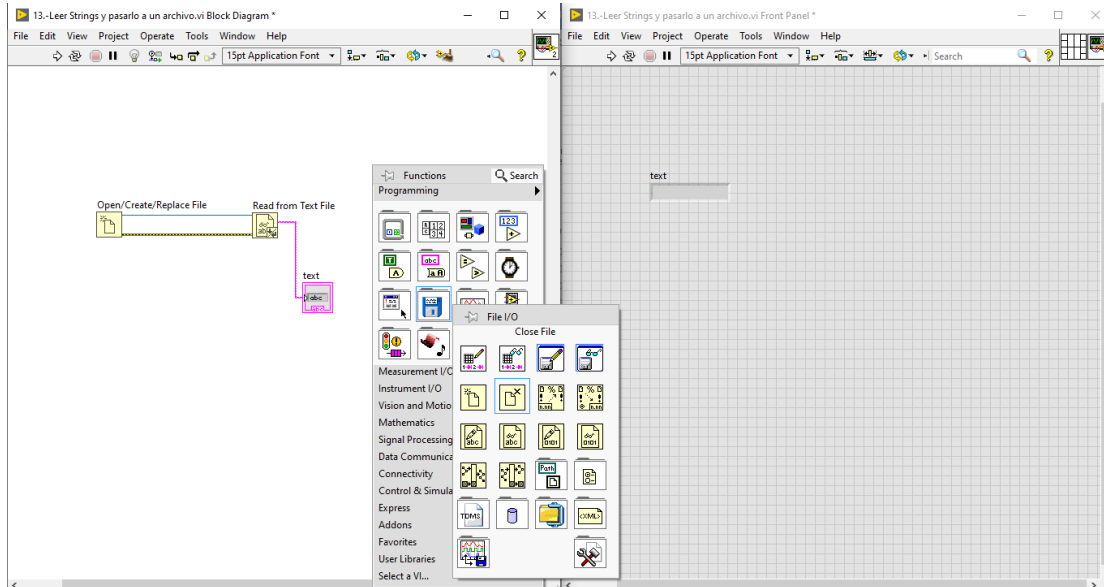


Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator. Con esto se crea un elemento de la interfaz donde sea visible el valor del error.

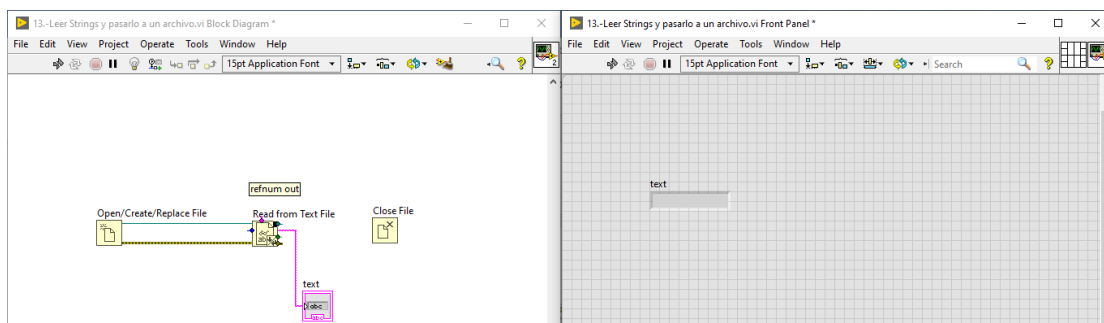
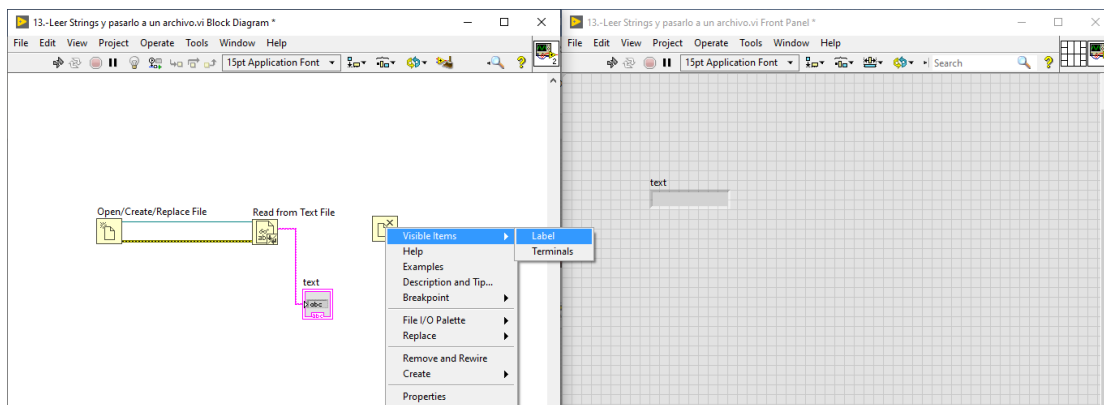


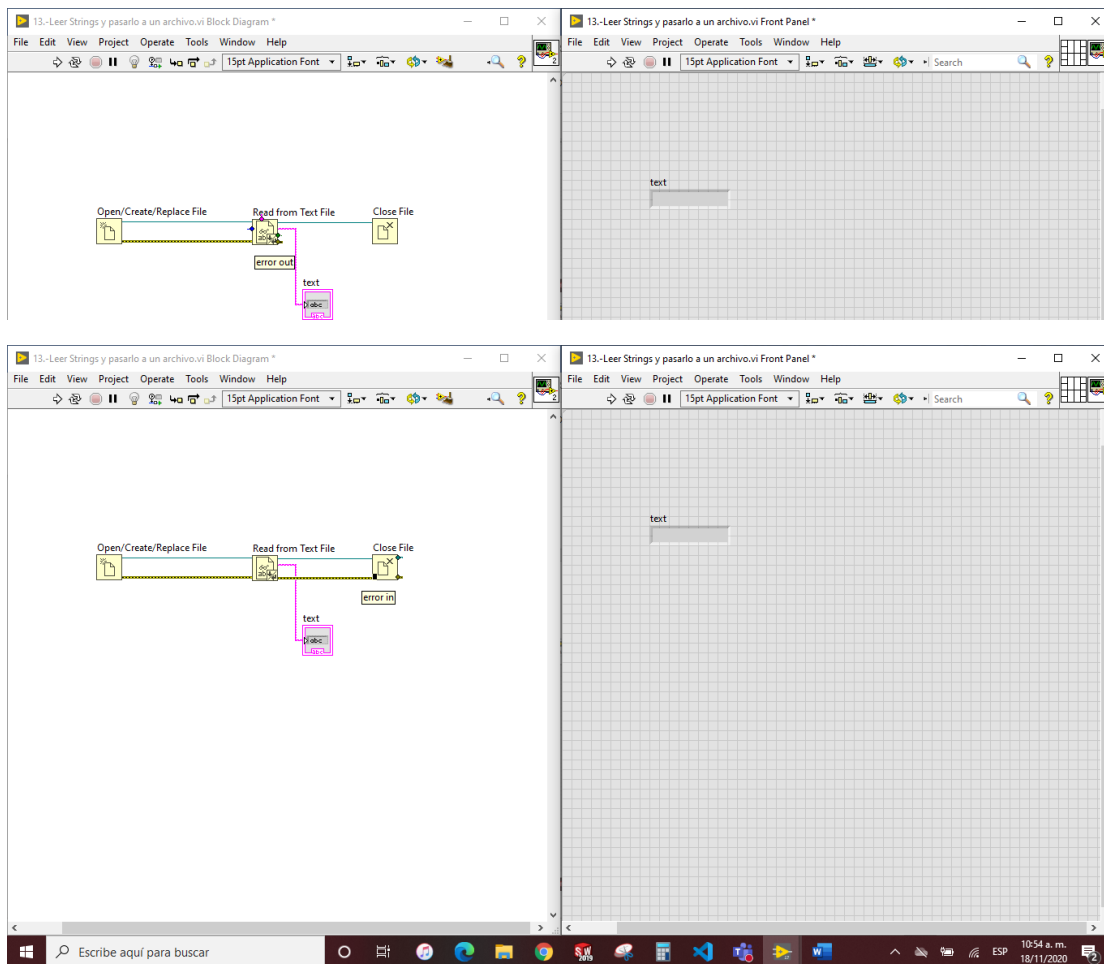
Block Diagram - Close File: Cerrar Archivo Abierto

El bloque **Close File** se debe usar en conjunto con el de **Read From Text File** y **Open/Create/Replace File** para que se pueda cerrar un archivo. Si este no se usa, el archivo se mantendrá en uso después de cerrar el programa y de hecho marcará un error fatal cuando lo quiera abrir por fuera de LabVIEW.



Mostrar nombre del bloque: Clic derecho → Visible Items → Label.

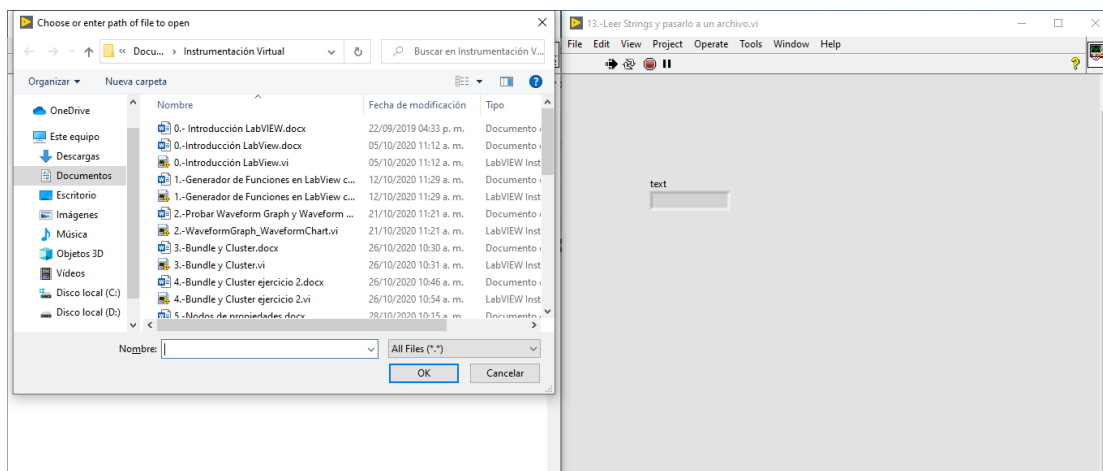


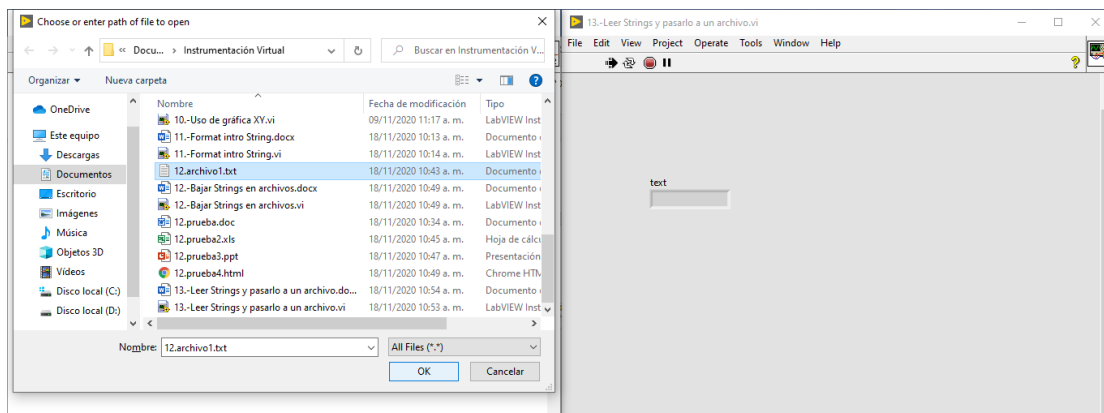


Ejecución del Programa: Leer Archivos y Convertir su Contenido a String

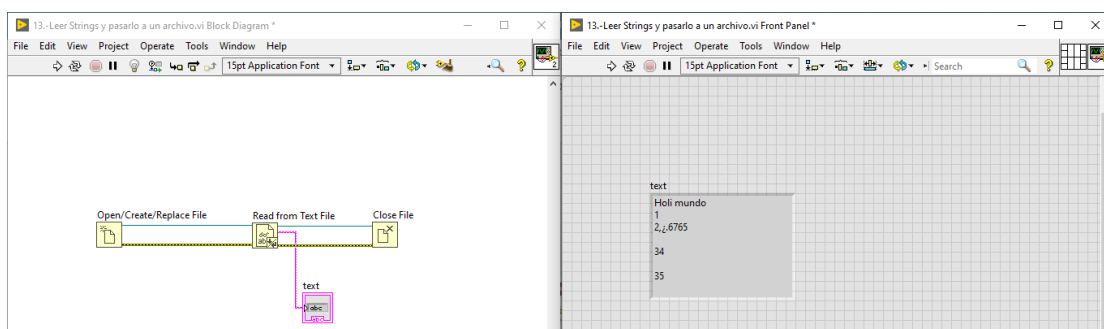
Archivo .txt

Al correr el programa me dirá que seleccione un archivo, puedo abrir uno con extensión .txt usando el bloc de notas.

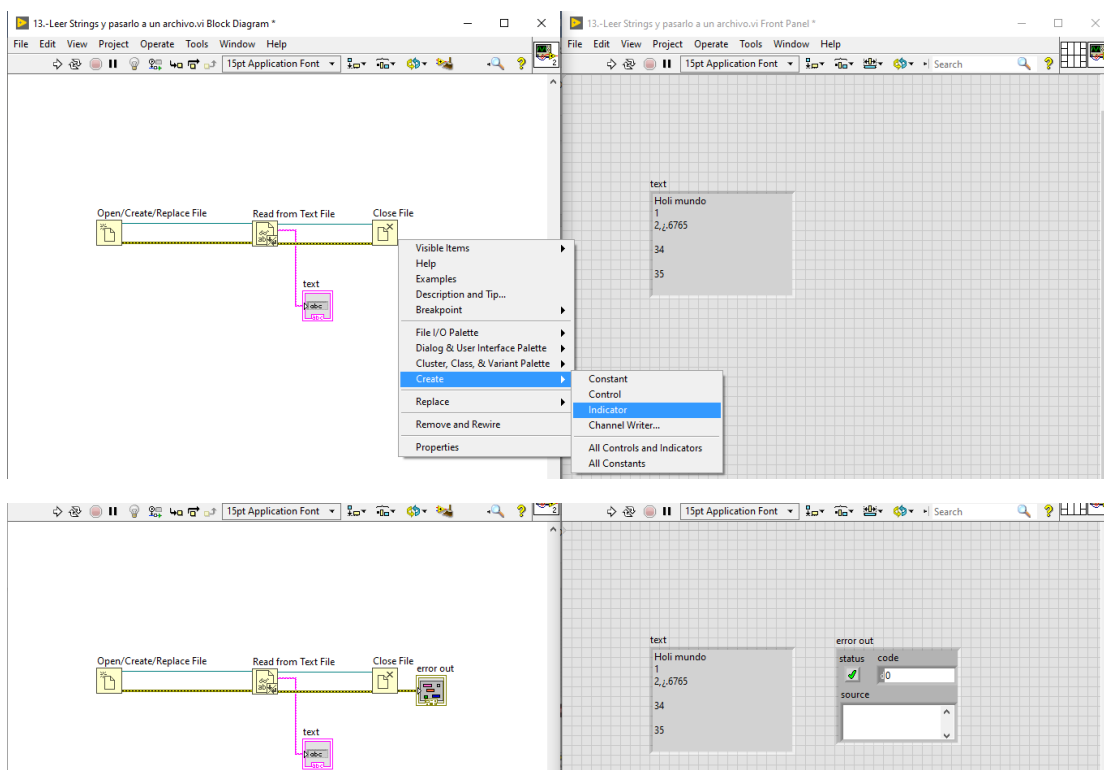




Al hacerlo se verá en el programa de LabVIEW el contenido del archivo de texto.

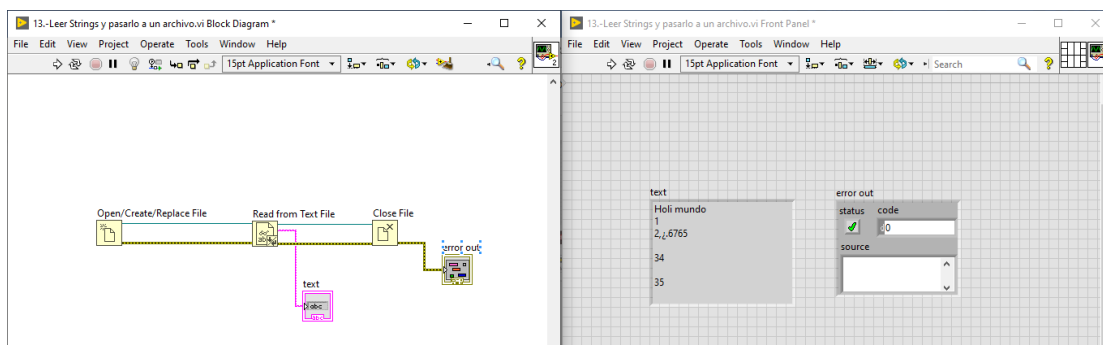
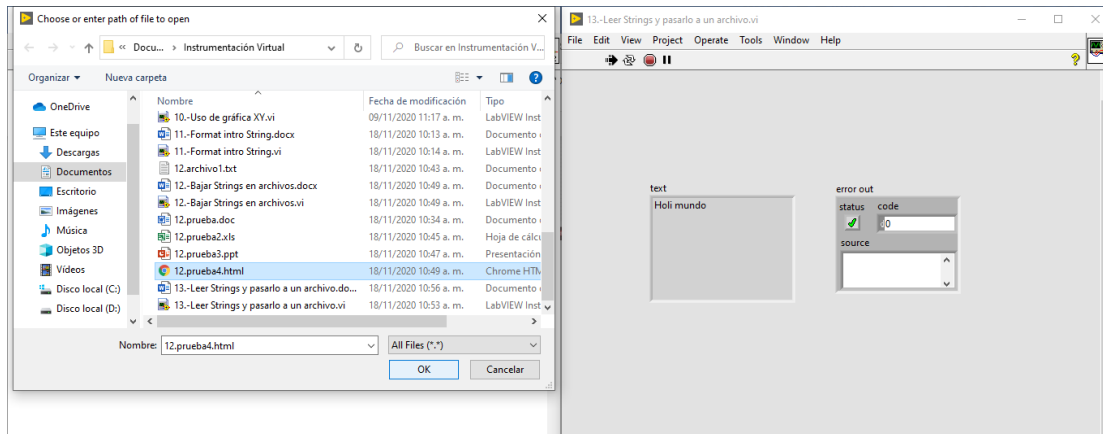


Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator. Con esto se crea un elemento de la interfaz donde sea visible el valor del error.



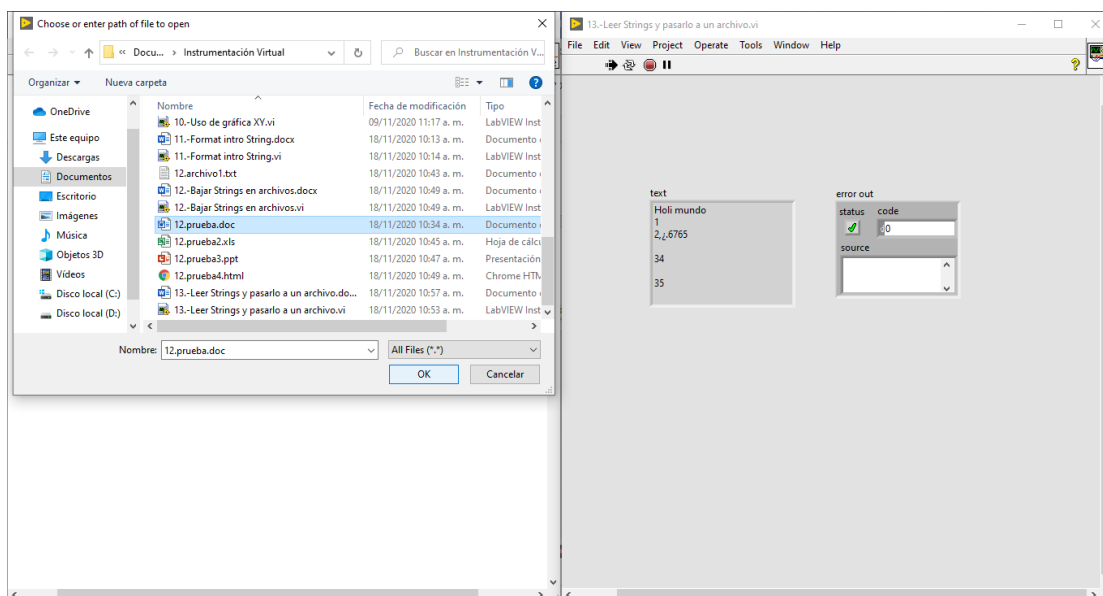
Archivo Página Web HTML

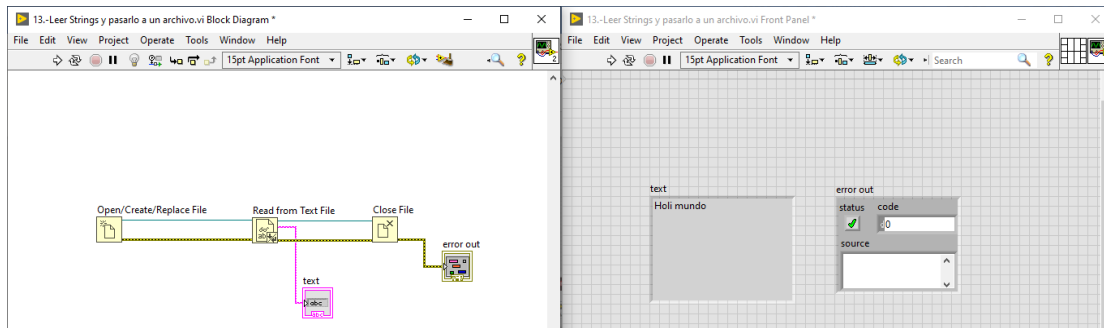
Incluso lo podríamos leer del código .html de una página web.



Archivo Word Microsoft Office

Puedo leerlo de otro tipo de archivos como el .doc, no se usa la extensión .docx porque hay veces que el archivo Word no lo abre bien LabVIEW.





Si no ponemos el bloque de close file, al parecer no pasa nada cuando se ejecute el programa, pero el problema es que como que no cerramos el archivo, el programa lo considera como abierto, aunque nosotros no lo veamos, por lo que el programa percibirá que metemos datos al archivo, pero no lo guardamos ni cerramos, por lo que en la memoria de LabView estará abierto.

