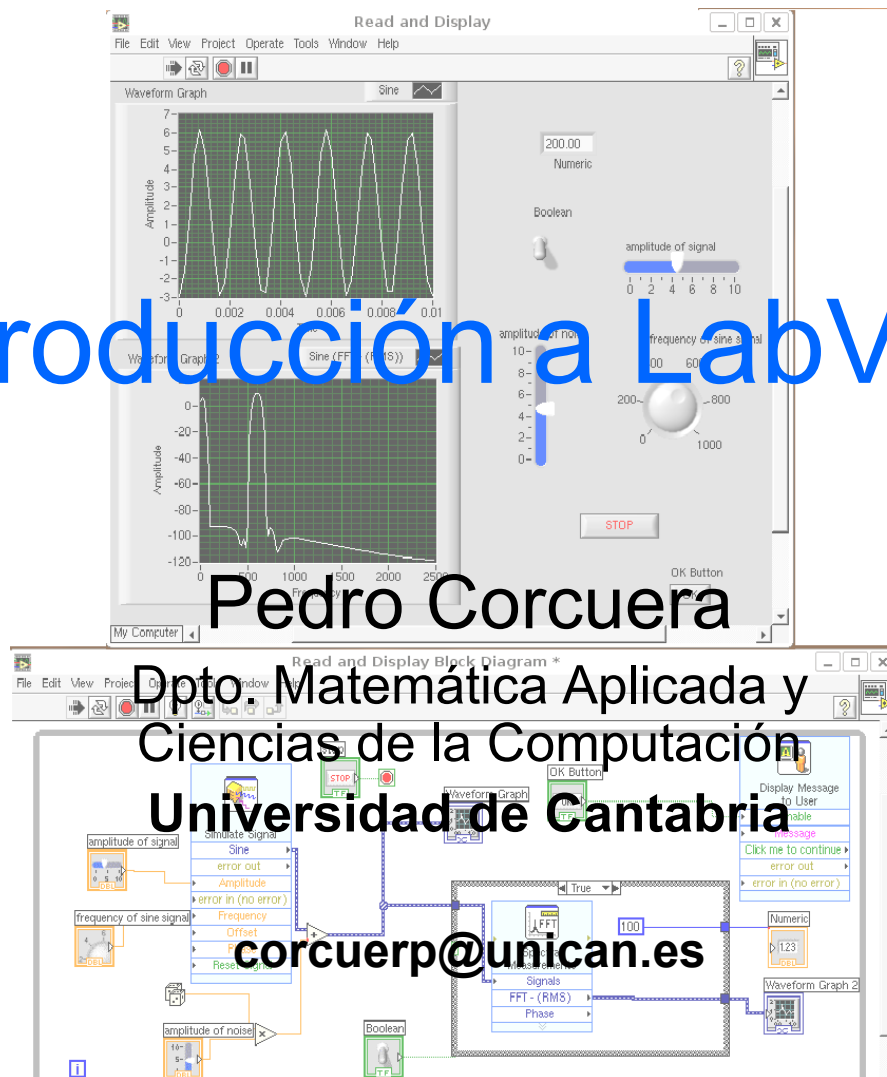


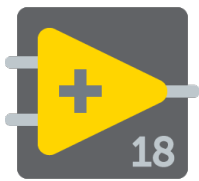
Introducción a LabView

Pedro Corcuera

Dpto. Matemática Aplicada y
Ciencias de la Computación
Universidad de Cantabria

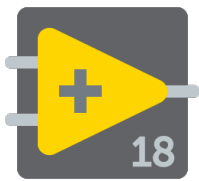
corcuerp@unican.es





Objetivos

- Introducir LabVIEW y las funciones comunes de LabVIEW
- Utilizar LabVIEW para crear aplicaciones.
- Entender paneles frontales, diagramas de bloque, iconos y paneles de conexión.
- Comprender los componentes de un instrumento virtual
- Construir aplicaciones de adquisición de datos



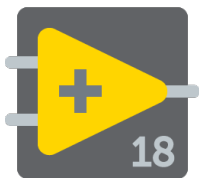
Indice

- Términos de LabVIEW
- Componentes de aplicación de LabVIEW
- Herramientas de programación de LabVIEW
- Crear una aplicación en LabVIEW



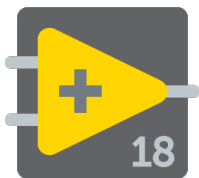
Introducción a LabView

- LabVIEW (acrónimo de **L**aboratory **V**irtual **I**strumentation **E**ngineering **W**orkbench) es una plataforma y entorno de desarrollo para un lenguaje de programación visual de [National Instruments](https://www.ni.com).
- El lenguaje gráfico se llama "G".
- LabVIEW se usa comunmente para adquisición de datos, control de instrumentación y automatización industrial para diferentes plataformas como Microsoft Windows, UNIX, Linux, y Mac OS X



Dataflow Programming

- El lenguaje de programación usado en LabVIEW (G) es un lenguaje de programación de flujo de datos.
- La ejecución es determinada por la estructura de un diagrama gráfico de bloques (código Fuente LV) consiste en una serie de bloques conectados.
- Las conexiones propagan las variables y en cualquier nodo puede ejecutarse tan pronto todas las entradas de datos estén disponibles. G es inherentemente capaz de ejecución paralela que aprovecha el multiproceso y multihilo.

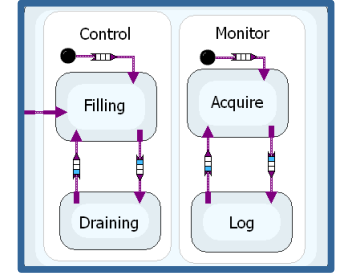


Programas LabView

- Los Programas en LabVIEW son llamados instrumentos virtuales (VIs)
 - Controles = entradas, Indicadores = salidas
 - Cada VI contiene tres partes principales:
 - Panel frontal– Cómo el usuario interacciona con el VI.
 - Diagrama de bloque – El código que controla el programa.
 - Icono/Conector – Medios para conectar un VI con otros VIs.
 - Cada control o indicador del panel frontal tiene una terminal correspondiente en el diagrama de bloques
-

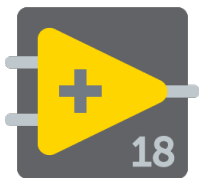


Statechart



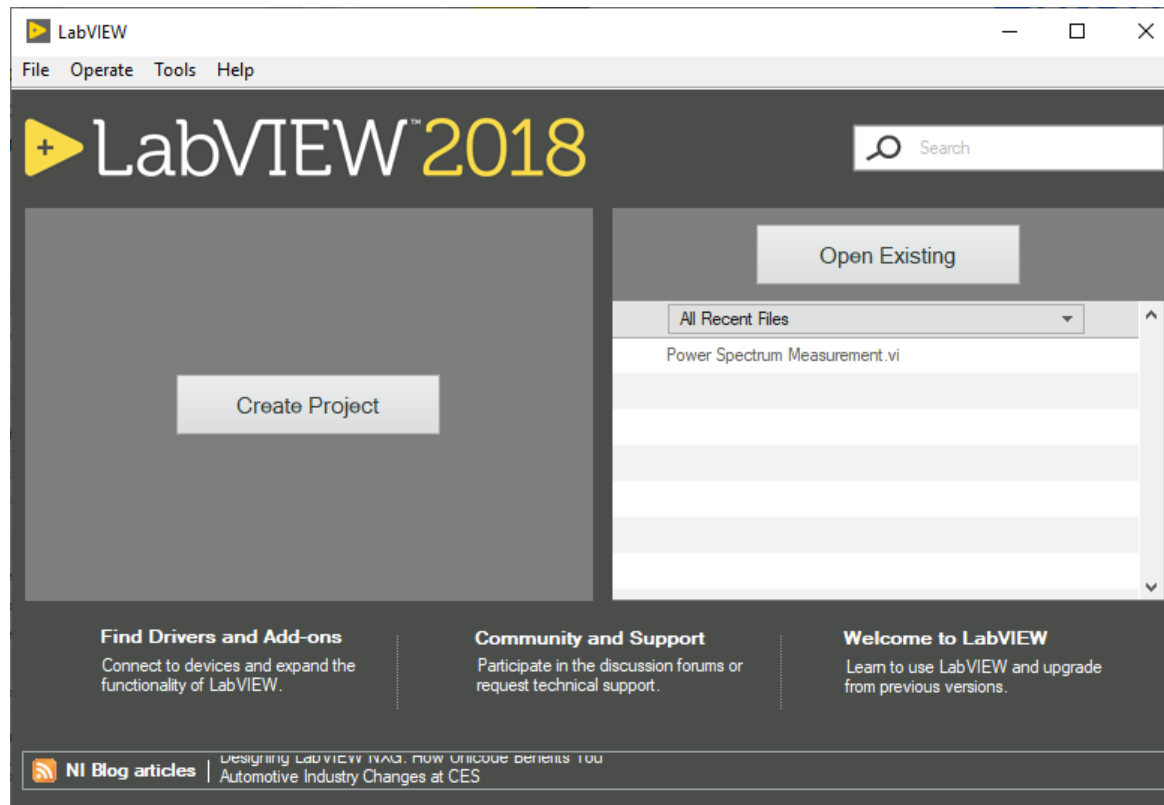
Graphical System Design Platform

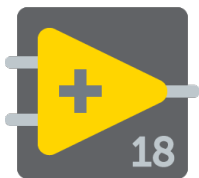
Embedded Platform



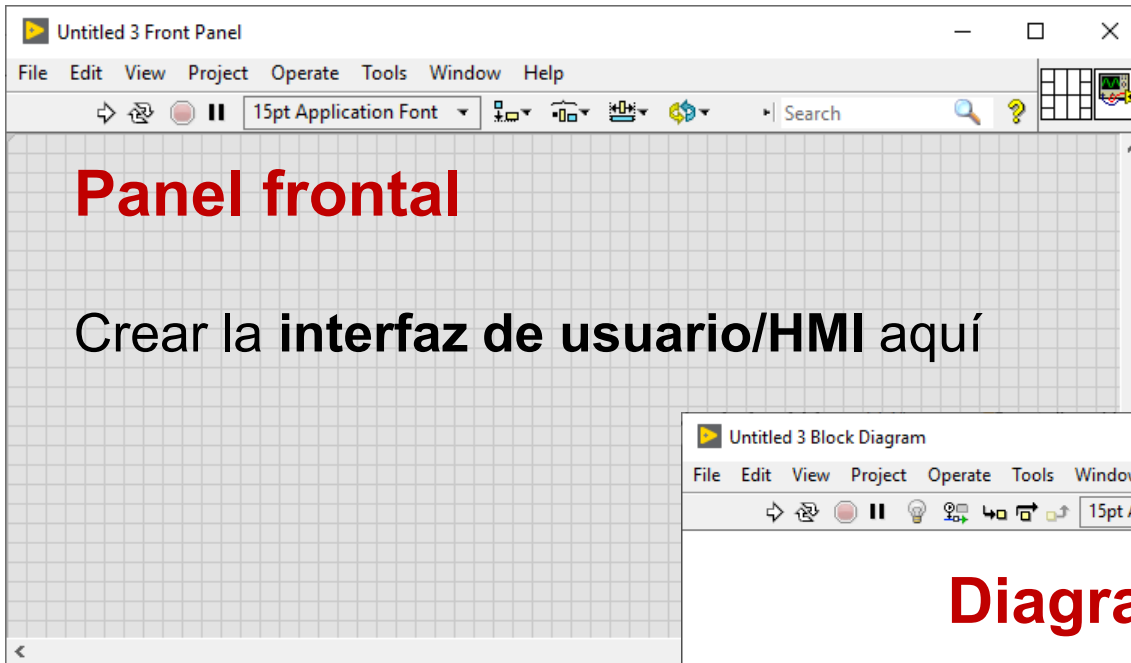
Entorno LabView

- Para crear un nuevo, seleccionar File»New VI o para crear un nuevo proyecto seleccionar “Create project”.

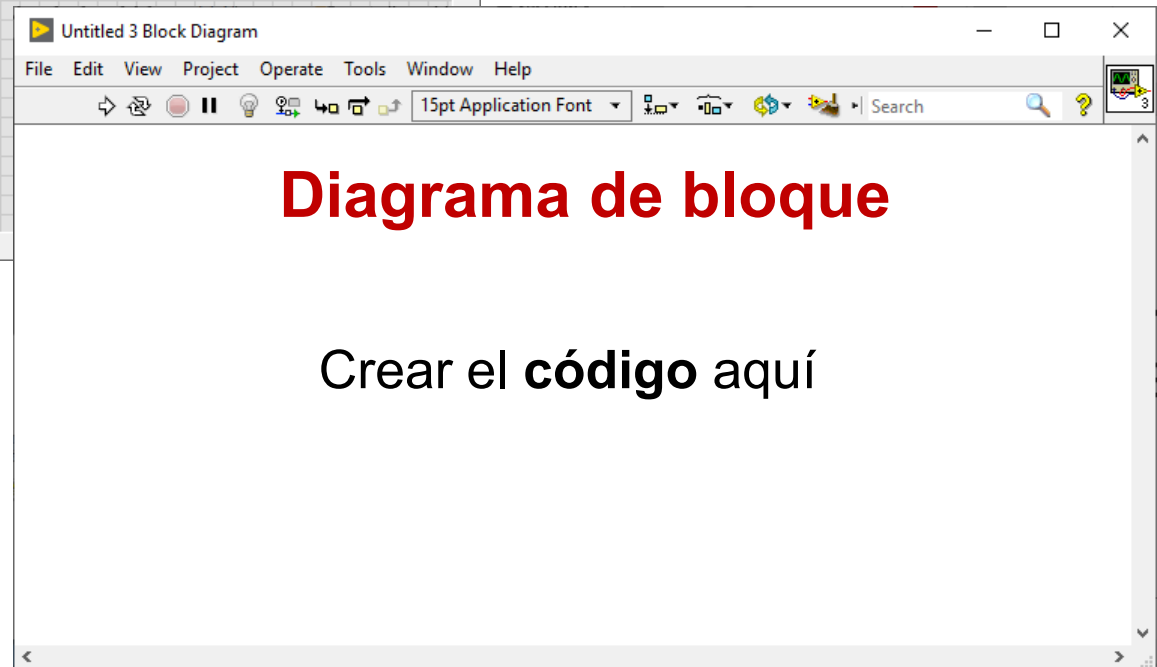




Programas LabView



Nota: ambos paneles se guardan en un fichero .vi.
VI = Virtual Instrument



Para intercambiar: **Ctrl + E**

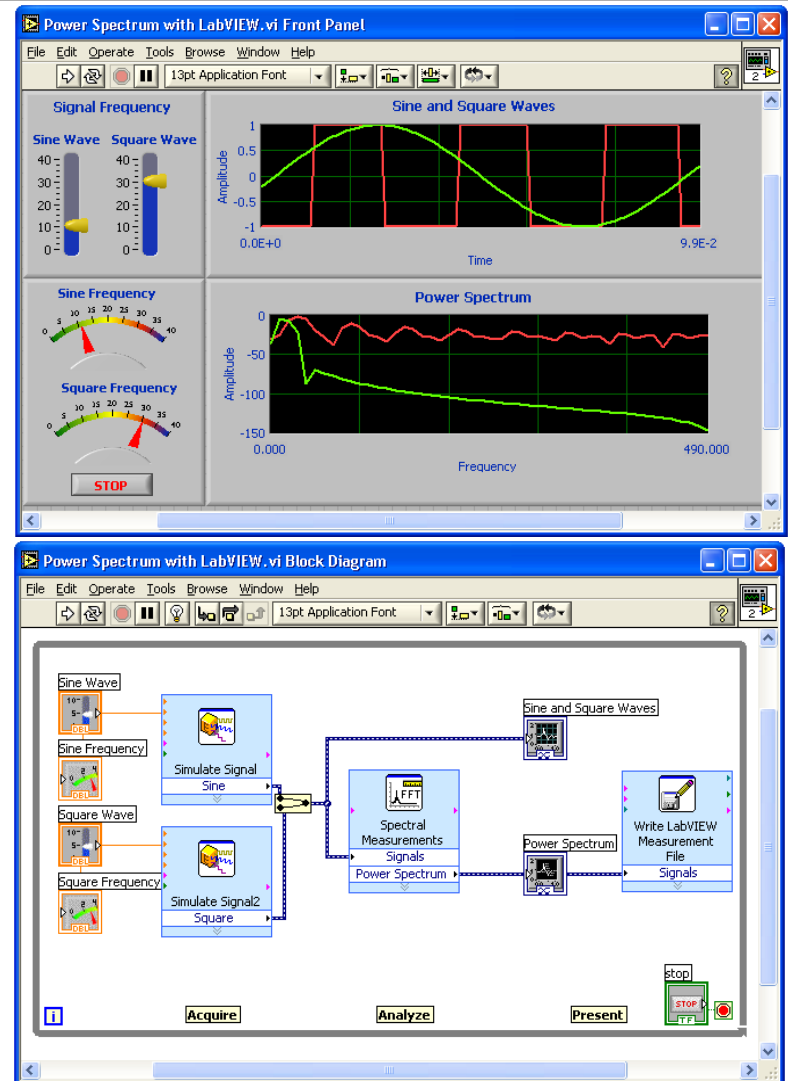
Programas LabView

Panel frontal

- Controles = entradas
- Indicadores = salidas

Diagrama de bloque

- Programa de acompañamiento para el panel frontal
- Componentes “cableados” entre sí



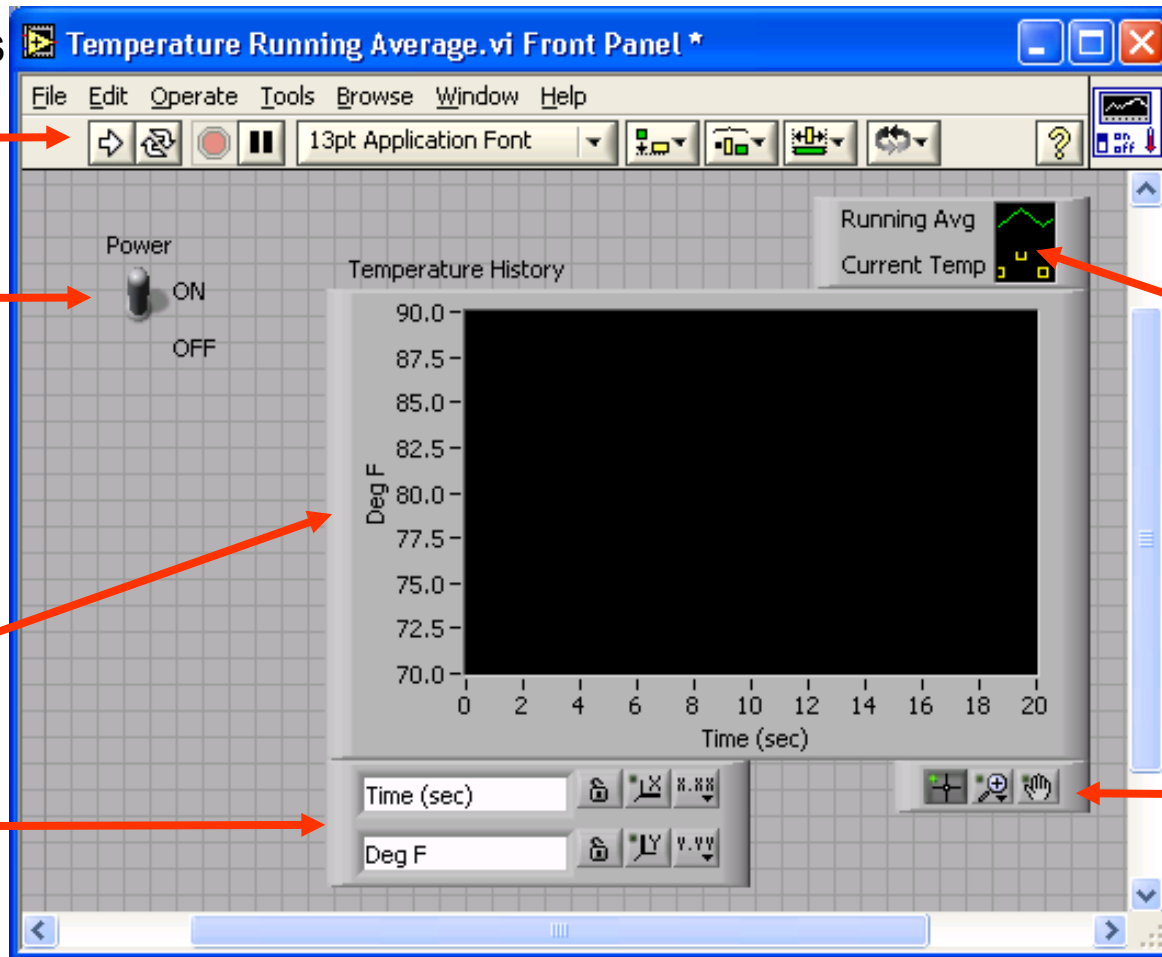
Panel frontal de VI

Barra de herramientas
del panel frontal

Control
Booleano

Gráfico de
Forma de onda

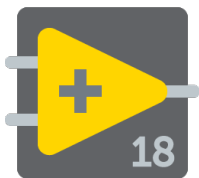
Leyenda de
La grafica



Icono

Leyenda del
gráfico

Leyenda de
la escala



VI Diagrama de bloque

Barra de Herramientas del diagrama de Bloque

SubVI

Funcion de dividir

Terminal grafica

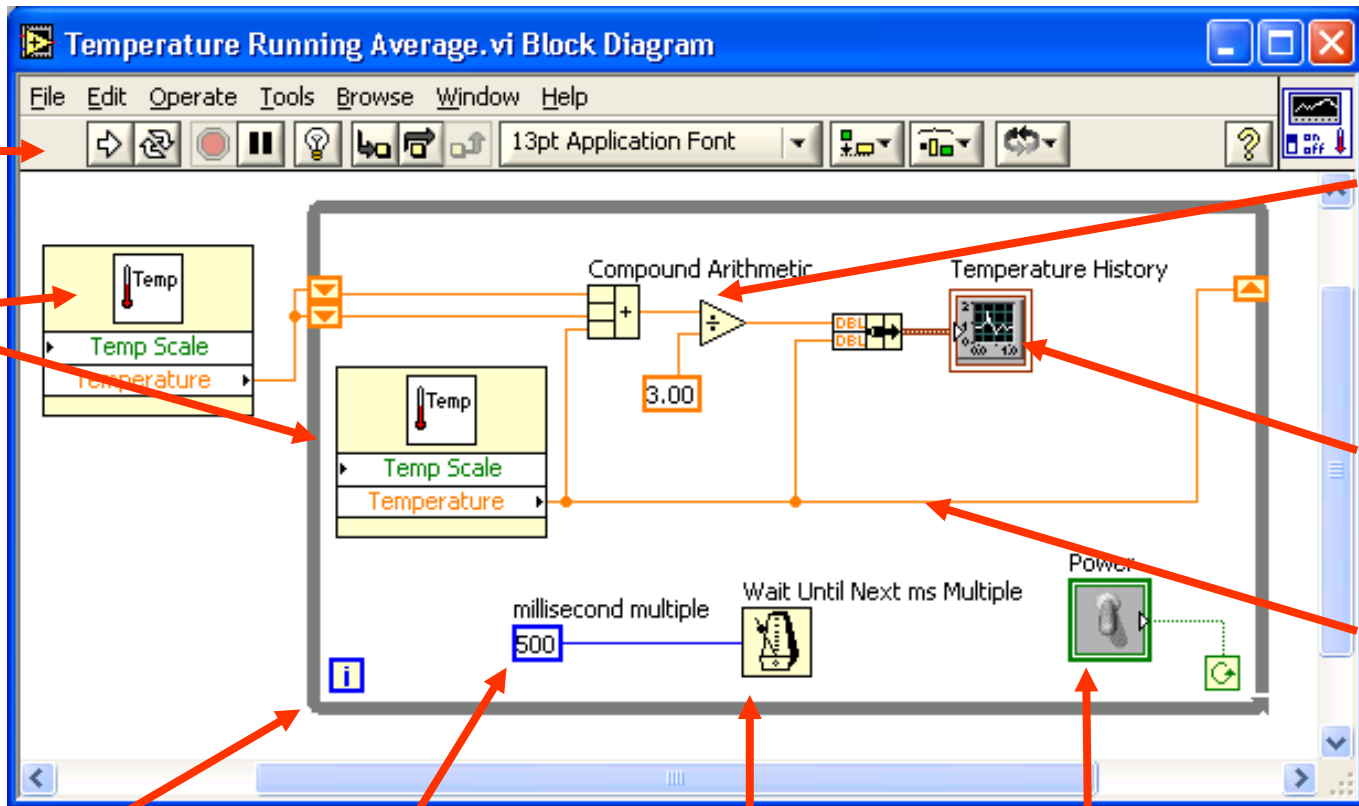
Cableado de datos

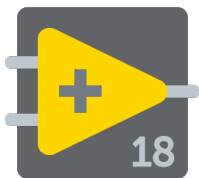
Estructura While loop

Constante numerico

Funcion de tiempo

Terminal de control booleano

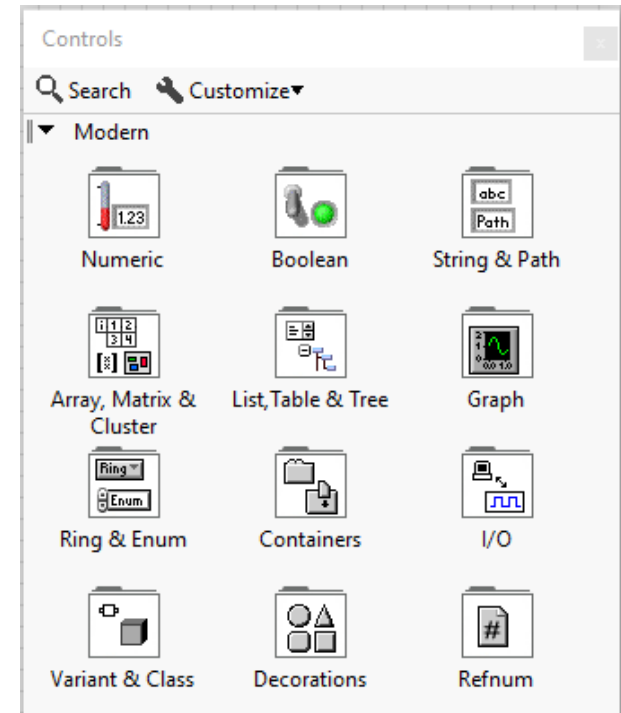


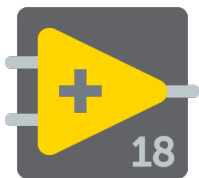


Paleta de controles

Paleta de control (Ventana del panel frontal)

Seleccionar View»Controls palette o hacer clic derecho en el espacio de trabajo en el panel frontal

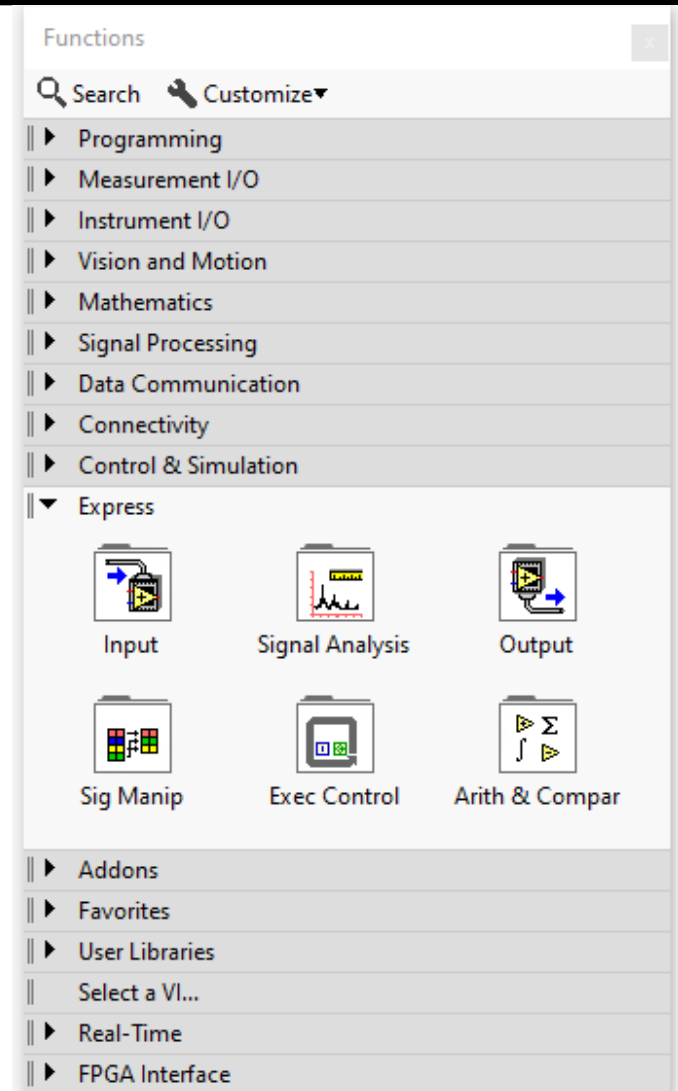




Paleta de funciones

Paleta de funciones (Ventana de diagrama de bloque)

Seleccionar View» Functions
Palette o hacer clic derecho en
el espacio de trabajo del
diagrama de bloque. También
se obtiene haciendo clic
derecho en una área abierta
del diagrama de bloques



Paleta de herramientas



- Seleccionar View» Tools Palette
- **Paleta flotante**
- Utilizado para operar y modificar objetos en el panel frontal y en el diagrama de bloques.



Herramienta de selección automática



Herramienta de operación



Herramienta de desplazamiento



**Herramienta de posicionamiento
y redimensión**



Herramienta de punto de parada



Herramienta de prueba



Herramienta de etiquetado



Herramienta para copia de color



Herramienta de cableado



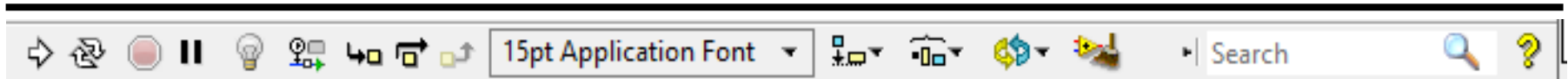
Herramienta para colorear



Herramienta de menú



Barra de herramientas de Estado



Botón de ejecución (Run)



**Botón de ejecución continua
(Continuous Run)**



**Cancelación de ejecución
(Abort Execution)**



Botón de pausa/continuación



**Botón de ejecución resaltada
(Highlight Execution)**



Retiene valores en conexiones



**Botón de entrada a nodo
(Step Into)**



**Botón de ejecución de nodo
(Step Over)**



**Finaliza ejecución de nodo
(Step Out)**

15pt Application Font

**Configuración de textos
(Text Settings)**



**Alineamiento de objetos
(Align Objects)**



**Distribución de objetos
(Distribute Objects)**



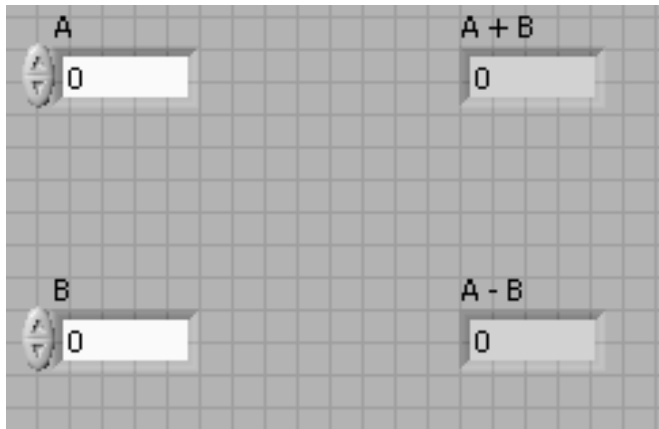
Reordenamiento



Reordena Diagrama

Creando un VI

Ventana de Panel Frontal

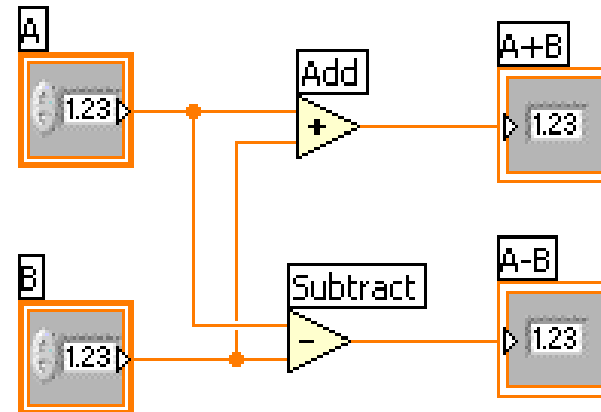


Los **controles** tienen una flecha en el lado derecho y tienen un borde grueso. Los **indicadores** tienen una flecha en el lado izquierdo y un borde fino.

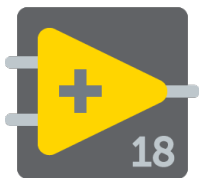
Terminales de Control

En general, las terminales de un color deben unirse (cablearse) con las terminales del mismo color, aunque hay compatibilidades.

Ventana de Diagrama de Bloques



Terminales de Indicador

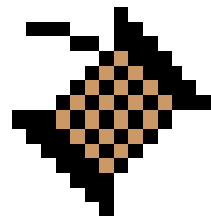
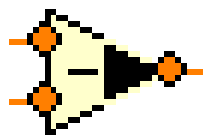


Creando un VI

- Seleccionar File » New VI para obtener un panel frontal y diagrama de bloques vacío
- Cuando se crea un objeto en el panel frontal, se crea una terminal en el diagrama de bloques.
- Cada terminal contiene información útil referente al objeto al cual corresponde en el panel frontal. Por ejemplo, los números de punto flotante y de doble-precisión, aparecen con terminales en color naranja y las letras DBL.

Creando un VI - Diagrama de bloques

Además de los terminales del panel frontal, el diagrama de bloques contiene funciones. Cada función puede tener múltiples terminales de entradas y salidas. La conexión de estas terminales es una parte muy importante de la programación en LabVIEW.



X-Y



Visible Items

Help

Description and Tip...

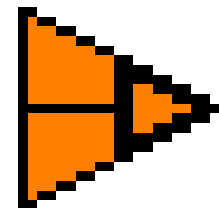
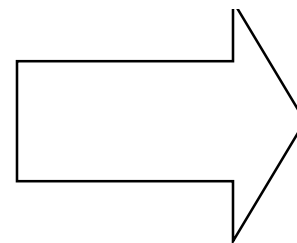
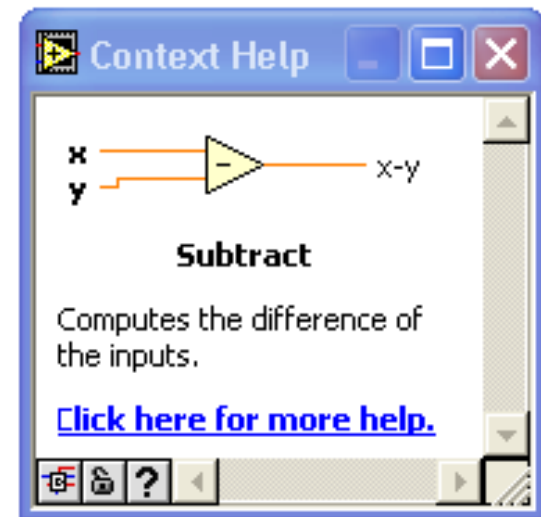
Set Breakpoint

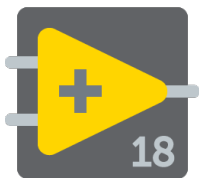
Create

Replace

Label

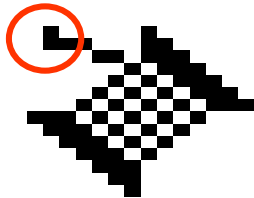
Terminals



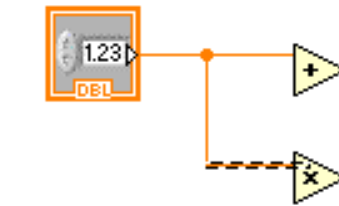


Consejos para conectar - Diagrama de bloques

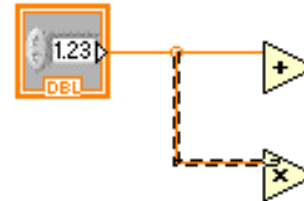
“Punto Caliente” de Cableado



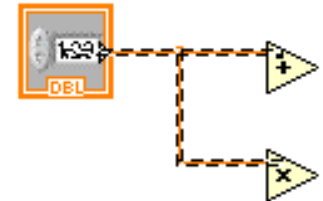
Hacer Clic para Seleccionar los Cables



single-click

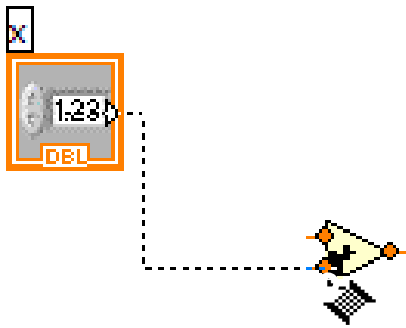


double-click

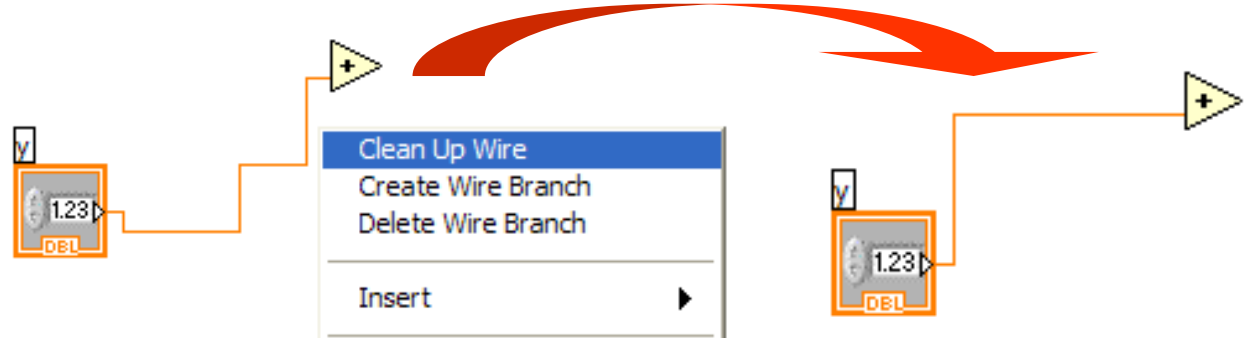


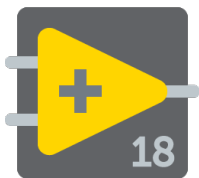
triple-click

Utilizar ruta automática del Cable



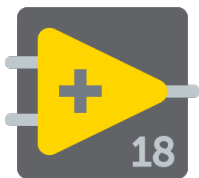
Reordenando el Cableado





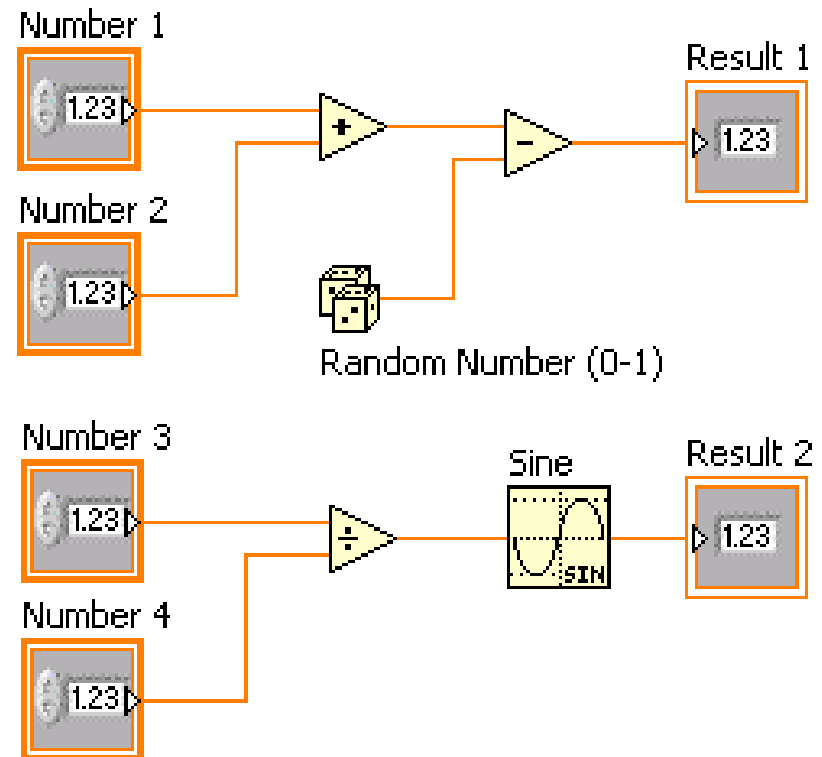
Creación de un VI

Demo - Taller

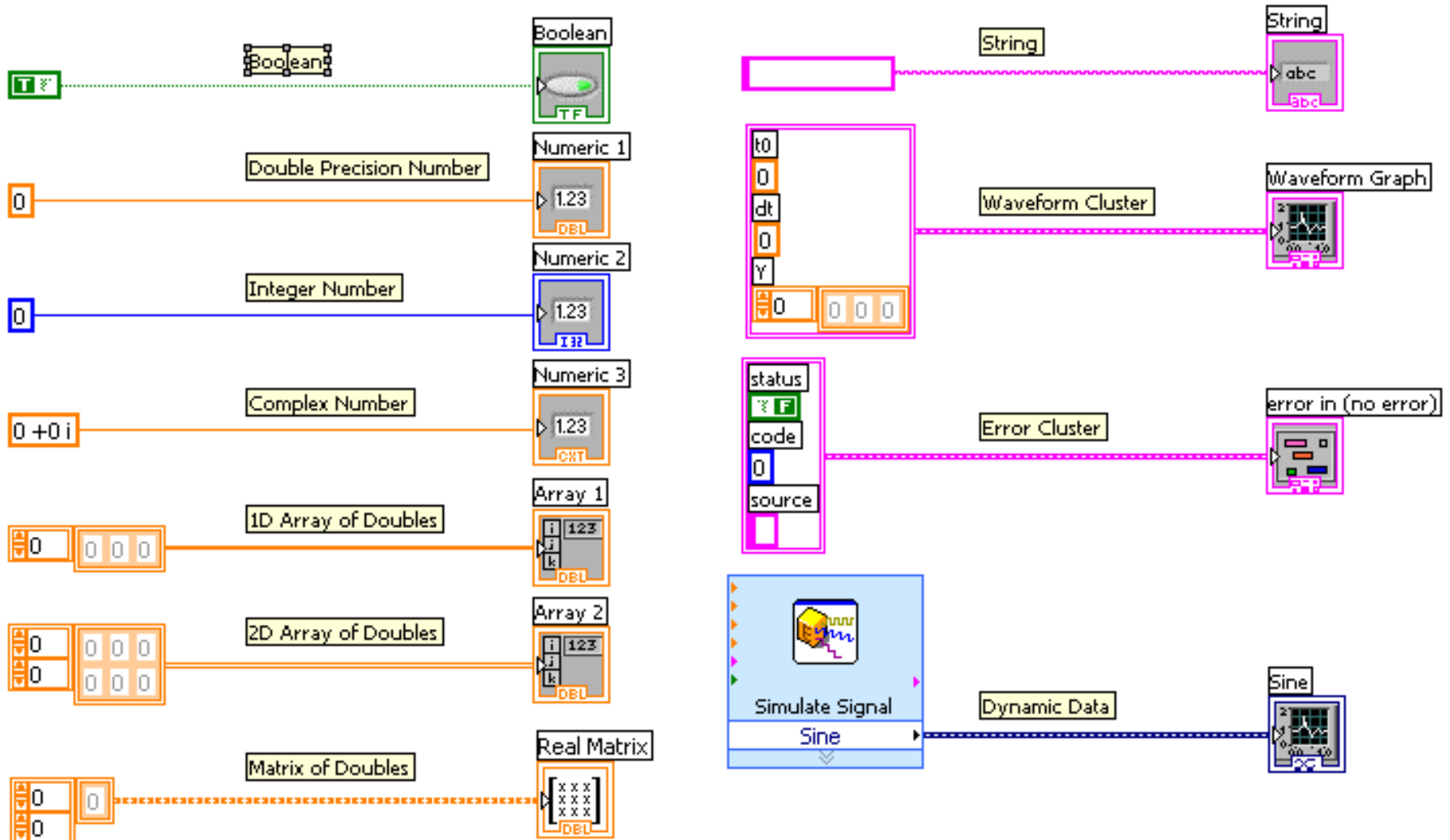


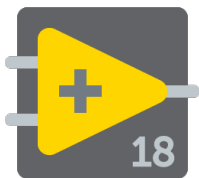
Programando el flujo de datos

- El diagrama de bloque se ejecuta dependiendo del flujo de los datos; el diagrama de bloques no se ejecuta de izquierda a derecha
- El nodo se ejecuta cuando los datos están disponibles para todos los terminales de entrada.
- Los nodos suministran datos a todos los terminales de salida cuando termina.



Tipos de datos en LabVIEW

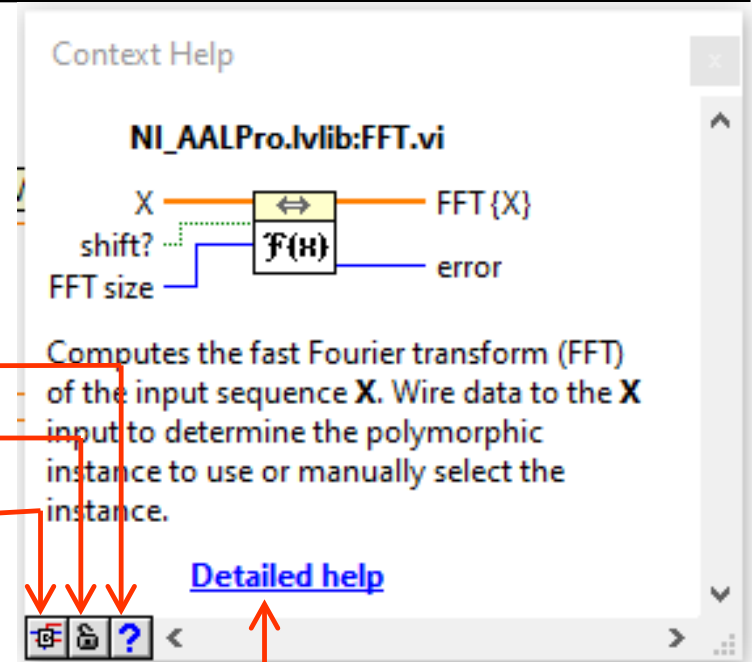




Opciones de Ayuda

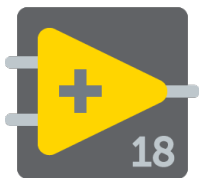
Ayuda Contextual

- Ayuda en línea
- Congelar Ayuda
- Ayuda del Diagrama Simple/Complejo
- Ctrl + H

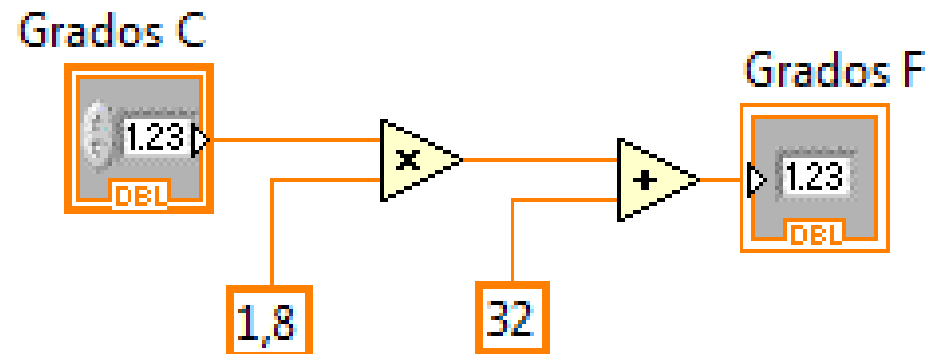
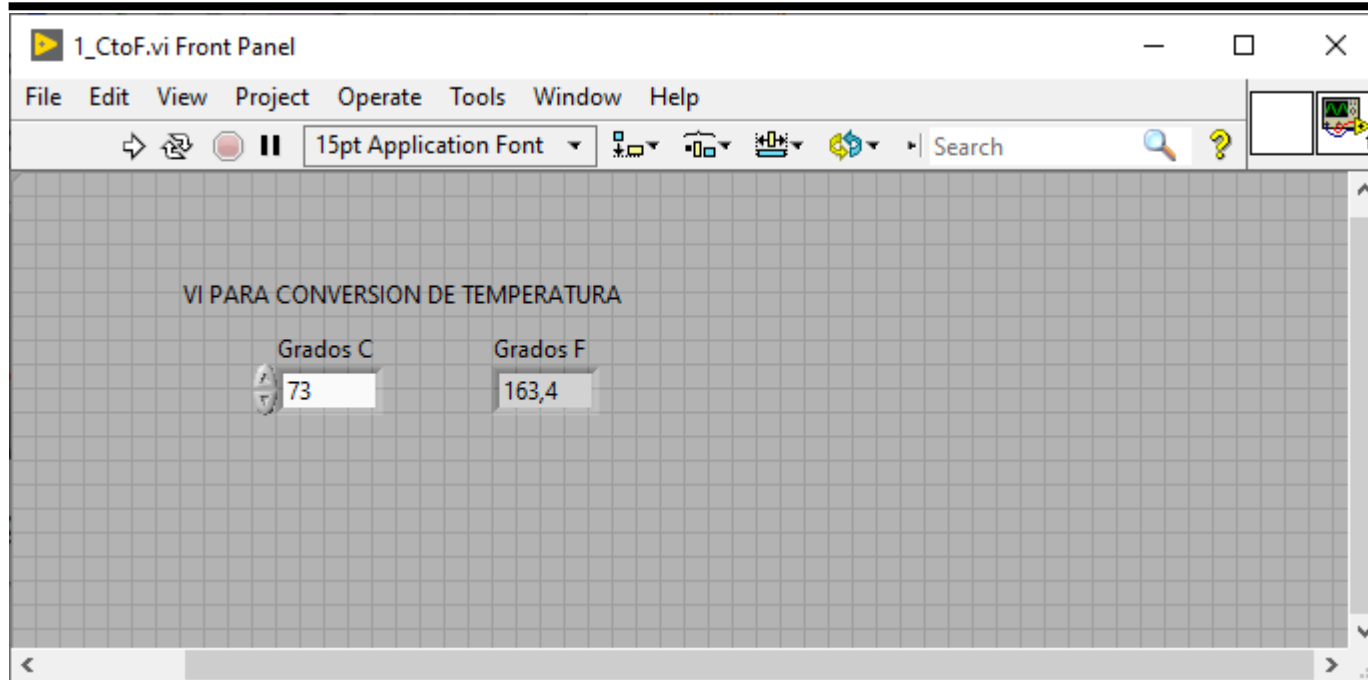


Referencias en Línea

- Todos los menús en línea
- Clic en las funciones del diagrama para tener acceso directo a la información en línea.



Ejercicio 1: Conversión de °C a °F





Técnicas para eliminar errores

- **Encontrando los Errores**



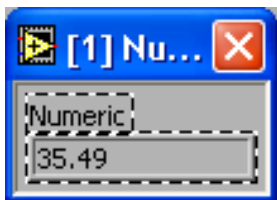
Hacer clic en el botón de “ejecución” que esta roto;
Aparece una ventana mostrando los errores

- **Resaltar la Ejecución**



Hacer clic en el botón de ejecución resaltada; el flujo de datos es animado utilizando burbujas. Los valores se despliegan en los cables.

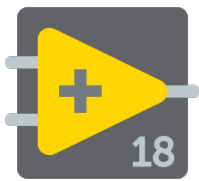
- **Herramienta de Prueba**



Hacer clic con el botón derecho sobre el cable para mostrar la ventana de prueba y así ver los datos mientras fluyen por el segmento de cable.

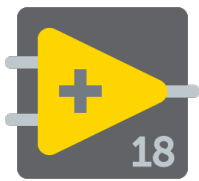


También se puede seleccionar la herramienta de prueba desde la paleta de herramientas y hacer un clic en el cable.



SubVIs

- Un SubVI es un VI que puede ser utilizado dentro de otro VI
- Similar a una subrutina
- Ventajas
 - Modular
 - Fácil para eliminar errores
 - No tiene que crear códigos
 - Requiere menos memoria

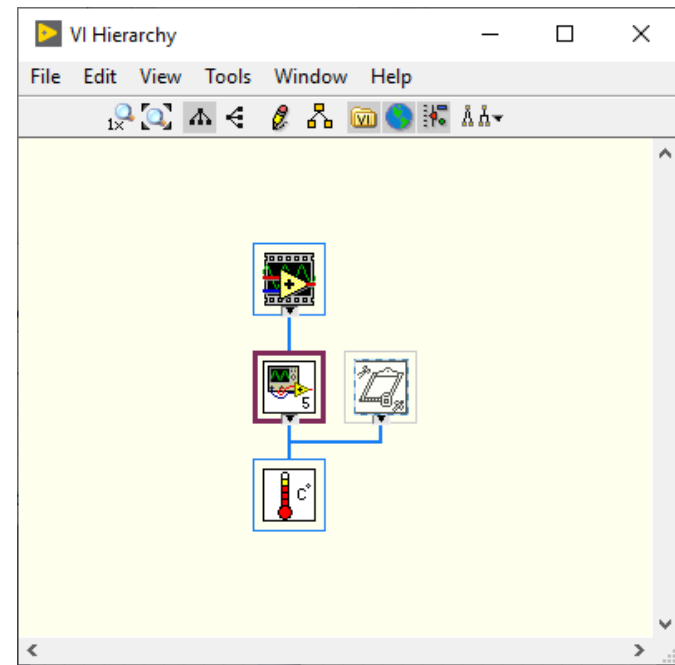
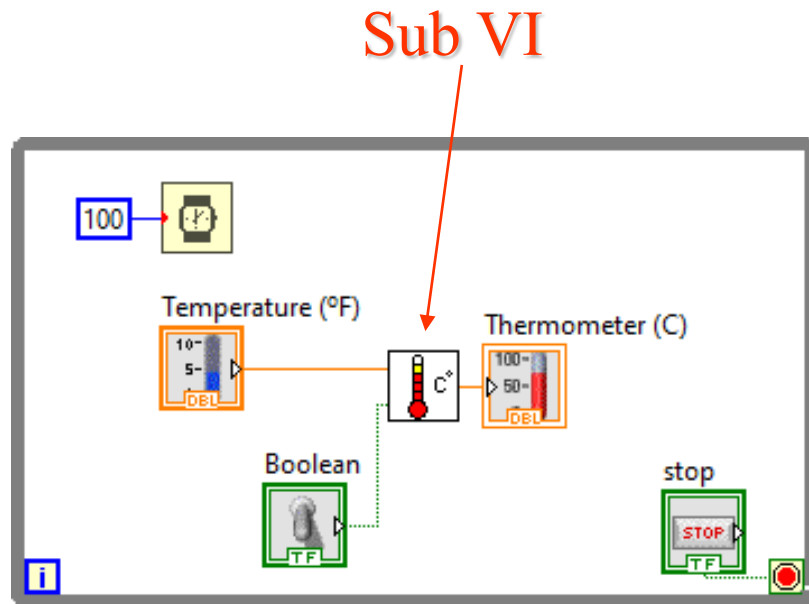


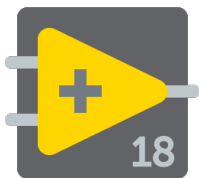
Pasos para crear un SubVI

- Crear el icono
- Crear el conector
- Asignar terminales
- Guardar el VI
- Insertar el VI dentro del VI principal

SubVIs

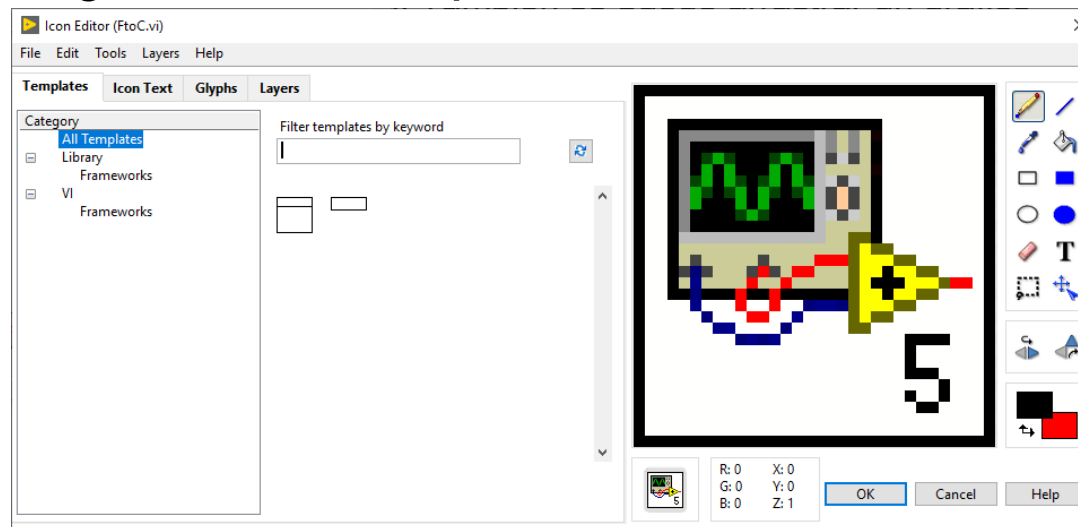
- Para ver el panel frontal de un subVI hacer doble clic sobre el subVI. Se puede ver la jerarquía de subVIs dentro de un VI principal haciendo un clic en View » VI Hierarchy

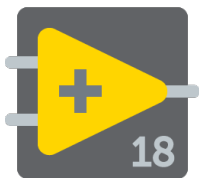




Pasos para crear un SubVI: crear el icono

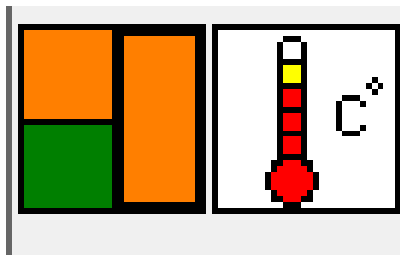
- Hacer clic derecho sobre el icono en el diagrama de bloque o panel frontal y seleccionar Edit Icon, o haciendo doble clic en el icono en la esquina superior derecha del panel frontal.
- Se usa las herramientas para crear el diseño del icono en el área de edición. También se puede arrastrar un gráfico desde un archivo a la esquina superior derecha del panel frontal o diagrama de bloque.

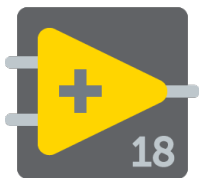




Pasos para crear un SubVI: crear el conector

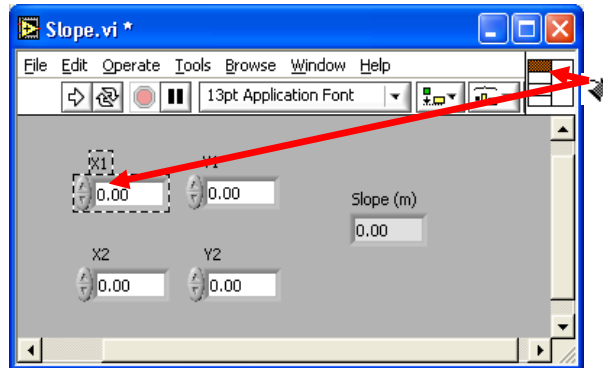
- El conector define las entradas y las salidas que se puede cablear al VI para utilizarlo como subVI, similar a los parámetros de una función de un programa..
- Para definir un conector: sobre el icono de conectores en la esquina superior derecha de la ventana del panel frontal, cada rectángulo una terminal y se utilizan para asignar entradas y salidas. El número de terminales mostrados depende del número de controles e indicadores en el panel frontal y se pueden configurar.

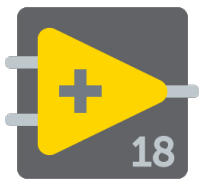




Pasos para crear un SubVI: asignar terminales

- Después de seleccionar un patrón para el conector, se definen las conexiones asignando un control del panel frontal o indicador a cada uno de los terminales del conector.
- Para ello se colocan las entradas en la izquierda y las salidas a la derecha para evitar patrones complicados y poco claros.
- Para asignar un terminal se selecciona un terminal del conector y se hace clic en el control del panel frontal o indicador. También puede seleccionar el control o el indicador primero y después seleccionar la terminal.



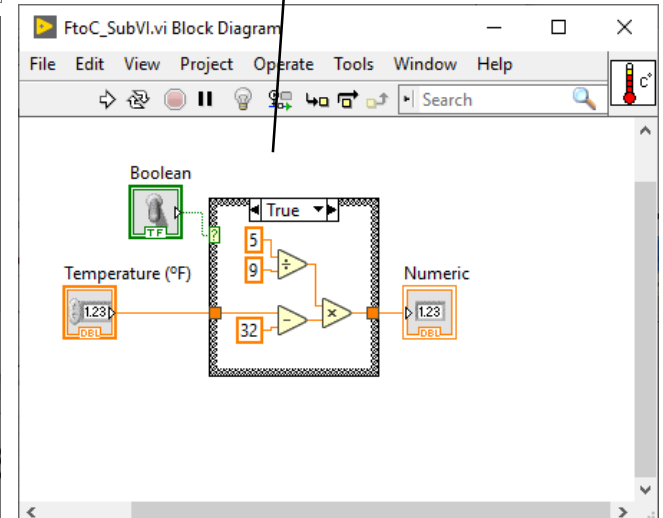
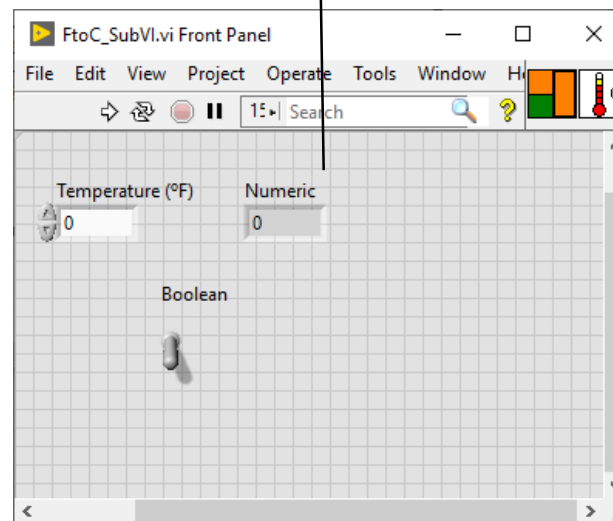
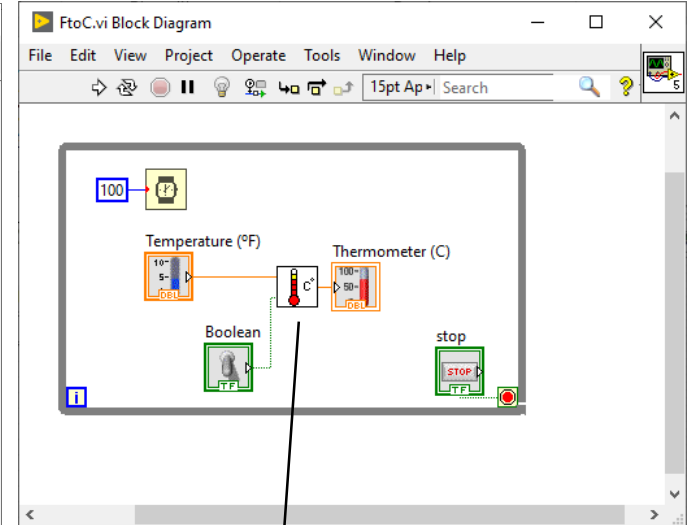
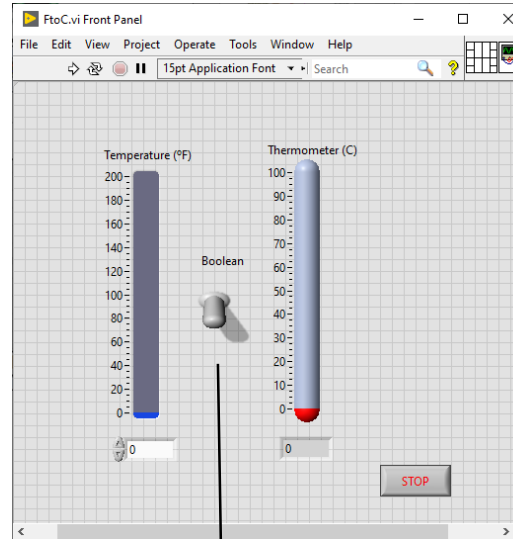


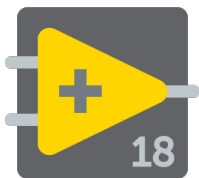
Pasos para crear un SubVI: guardar el VI

- Escoger un sitio o un carpeta fácil de recordar
- Organizar por funcionalidad
 - guardar los VIs similares en un mismo directorio (Ej. Utilidades_matematicas)
- Organizar por aplicación
 - Guardar todos los VIs usados para una aplicación específica dentro de un directorio o un archivo de librería (Ej. Lab_1_RespuetadeFrecuencia)
 - Archivo de librería (.llbs) combina todos los VIs en un solo archivo, ideal para transferir aplicaciones enteras

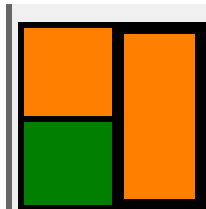
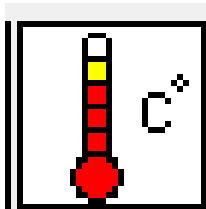
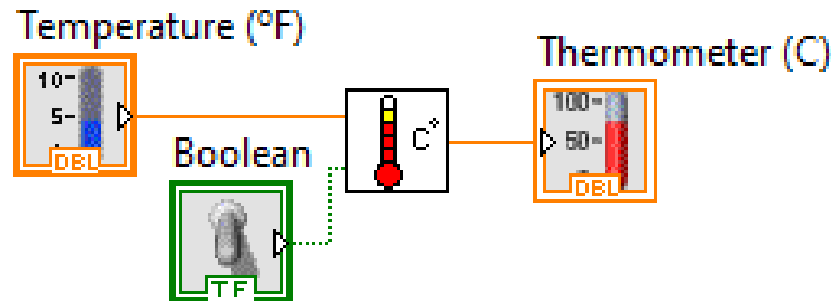
Sub VIs

- Que es un subVI?
- Elaboración de un icono y un conector para un subVI
- Utilizando un VI como un subVI

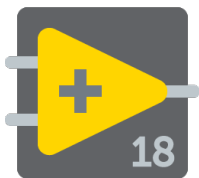




Icono y conector



- Un icono representa un VI en otro diagrama de bloque
- El conector muestra terminales disponibles para transferir datos

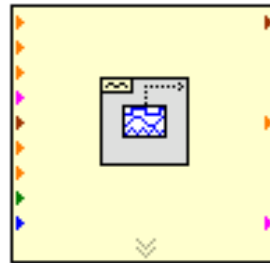


Nodos del diagrama de bloques

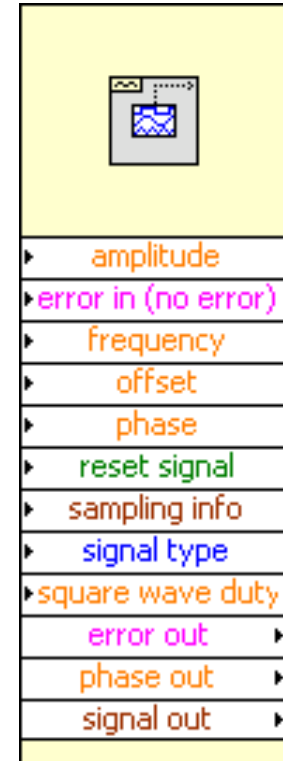
Icono



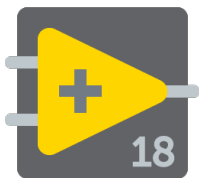
Nodo Expandible



Nodo Expandido

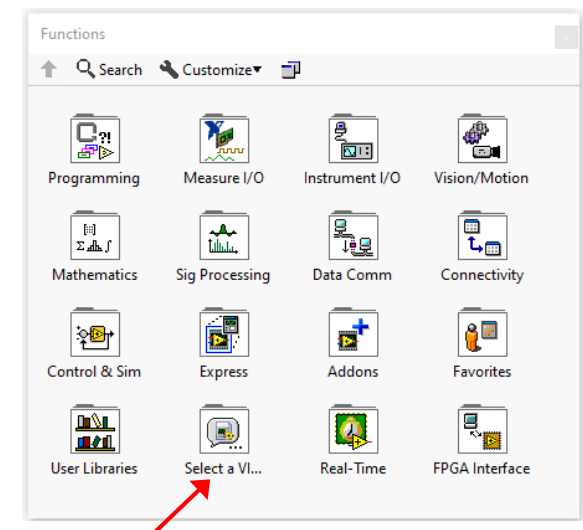
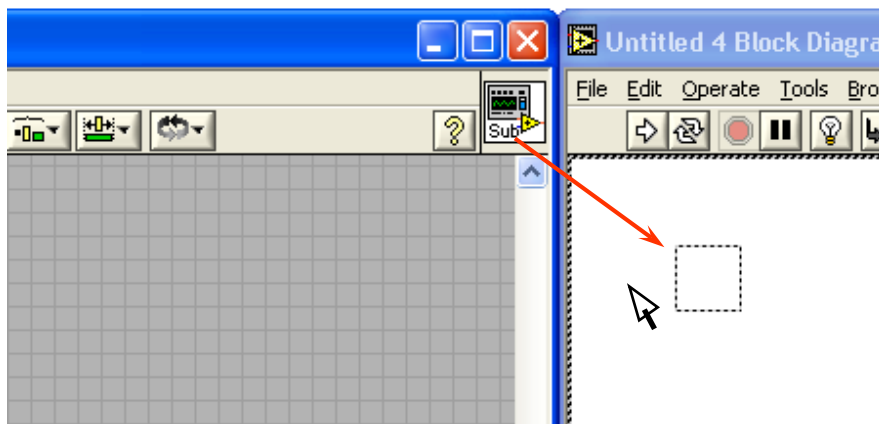


- VI de Generador de Funciones
- El mismo VI, visto en tres maneras diferentes.
- El campo amarillo designa un VI Estándar.
- El campo azul designa un VI Expreso



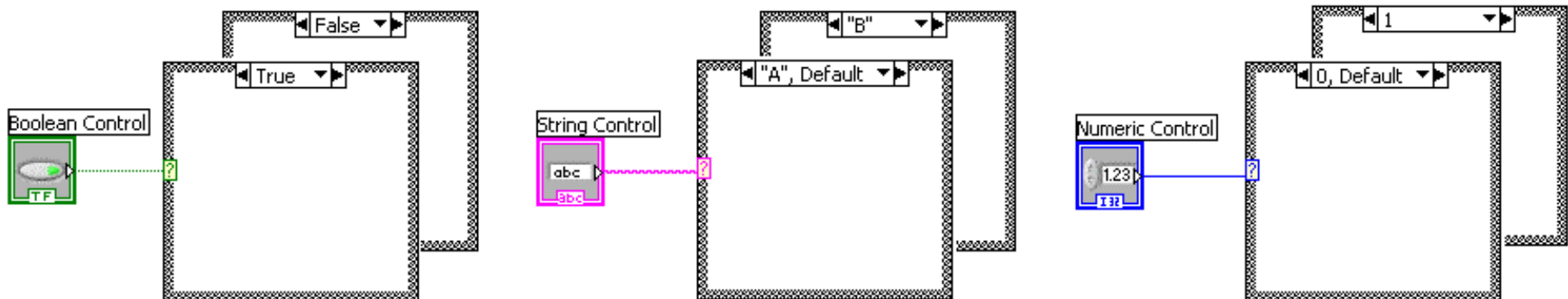
Insertar el SubVI dentro de un VI Principal

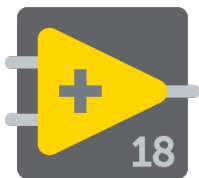
- Seleccionar Functions>> Select VI. Navegar y hacer doble-clic en el VI que se desea utilizar como subVI y colocarlo en el diagrama de bloques.
- También se puede hacer clic en el icono en la esquina superior derecha del panel frontal o diagrama de bloque del VI que se desea utilizar como subVI y arrastrar el icono al diagrama de bloque del otro VI.



Estructuras Case

- Las estructuras Case permiten elegir un curso de acción dependiendo del valor de entrada, semejante a los enunciados if-then-else en otros lenguajes
- Se encuentra en la subpaleta Execution Control de la paleta Functions





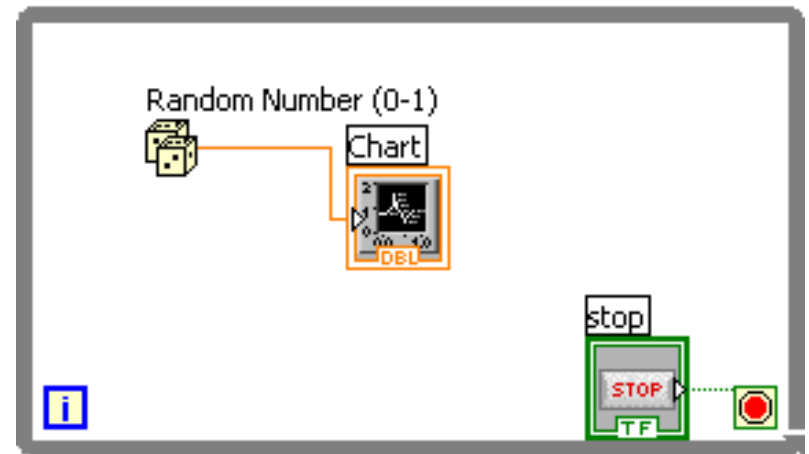
Controles para ciclos

- Localizados en la paleta Functions»Structures

While Loop

- **Ciclo while**

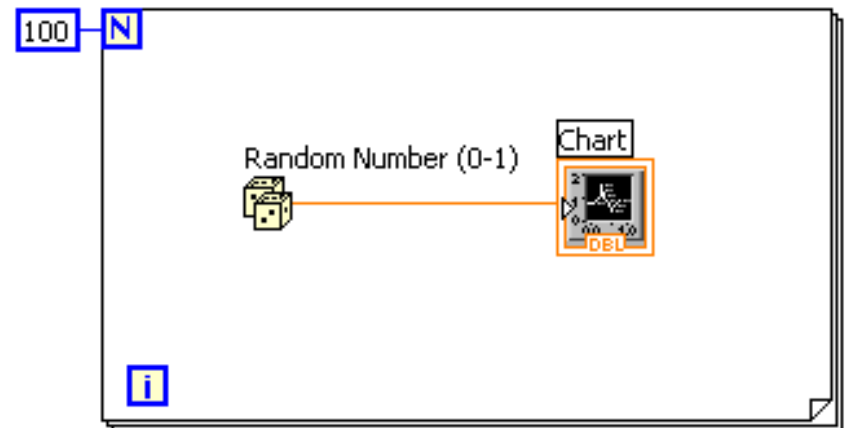
- Tiene una Terminal de Iteración
- Se ejecuta al menos una vez
- Se ejecuta de acuerdo a la Terminal Condicionante



For Loop

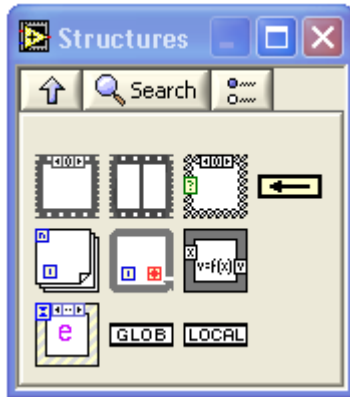
- **Ciclo for**

- Tiene una Terminal de Iteración
- Se ejecuta de acuerdo a la entrada **N** de la Terminal de Contador

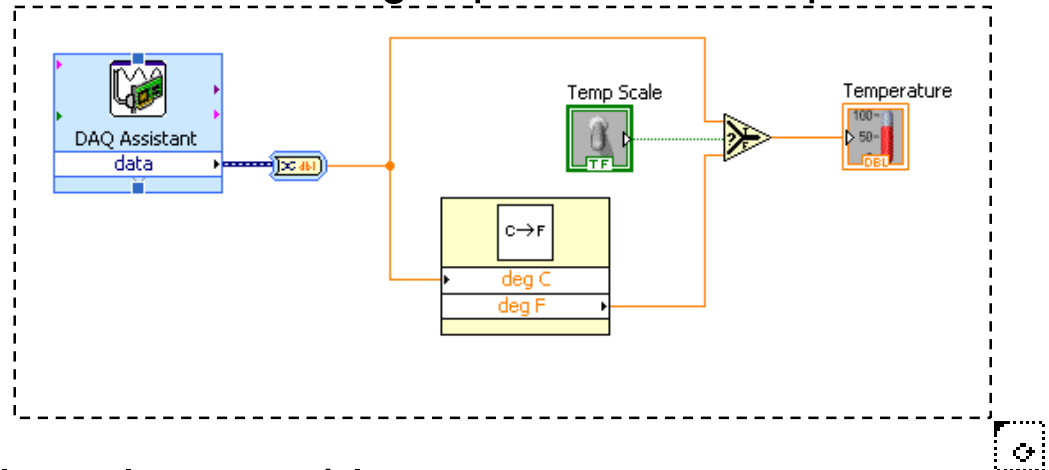


Controles para ciclos

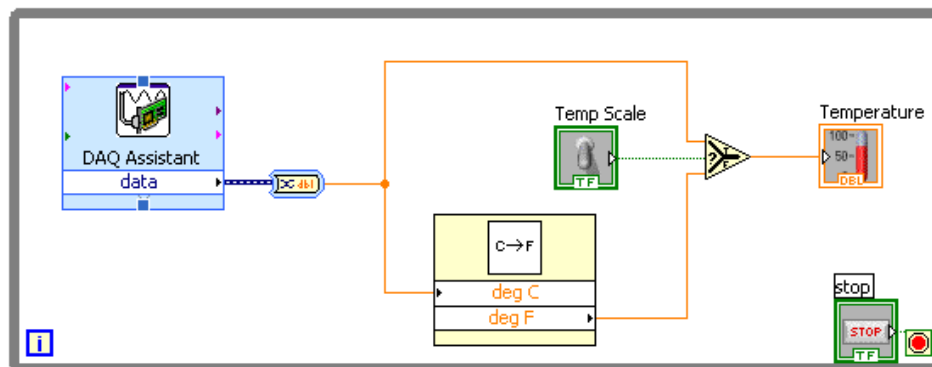
1. Seleccionar el ciclo



2. Encerrar código que va a ser repetido

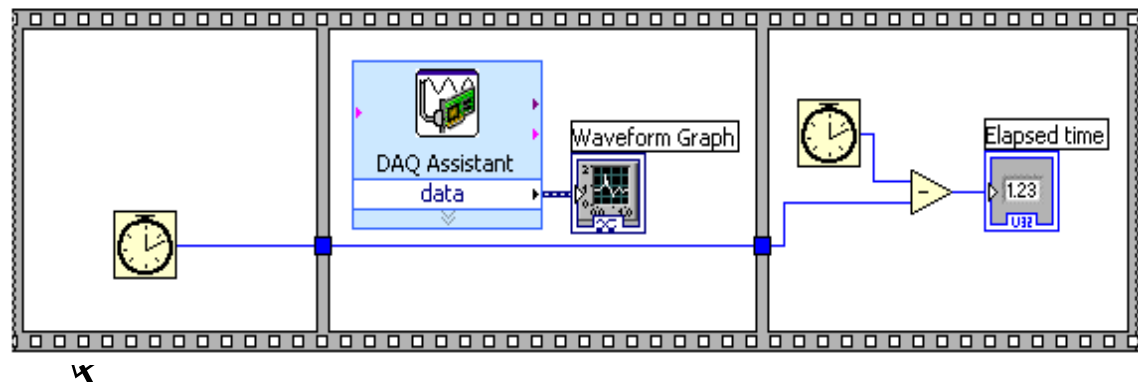
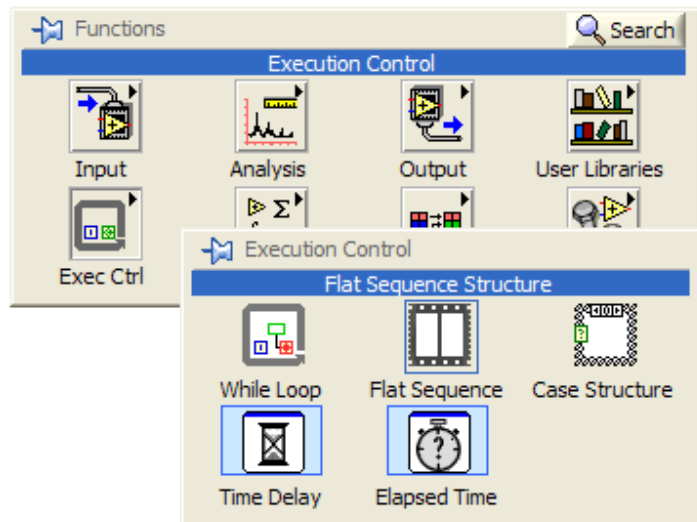


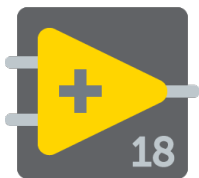
3. Arrastrar nodos adicionales y luego cablear



Estructuras Sequence

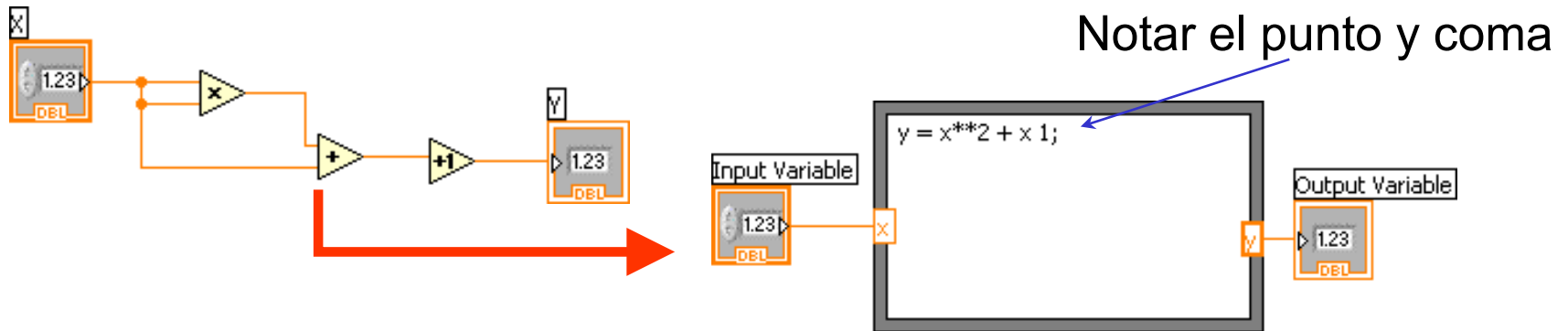
- En LabView un nodo se ejecuta cuando los datos están disponibles en todas sus terminales de entrada
- Cuando se necesita controlar el orden de la ejecución del código en el diagrama de bloques, se puede utilizar una Sequence Structure (Flat)
- Se encuentra en la paleta de Execution Control





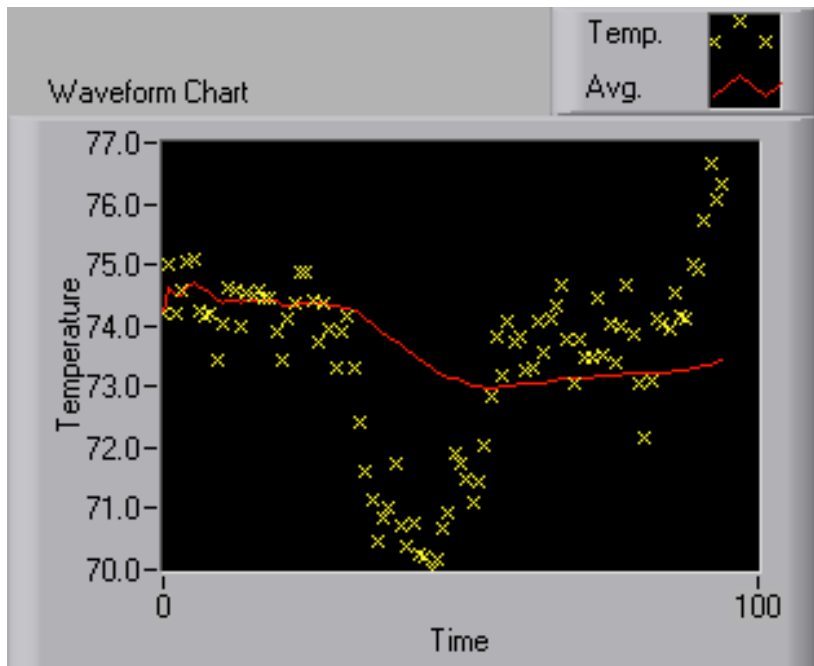
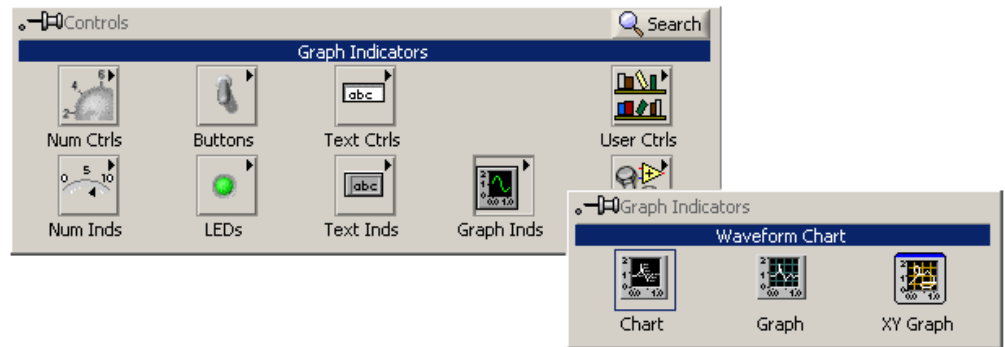
Formula Node

- Algunas veces es preferible programar expresiones matemáticas con funciones basadas en texto, en lugar de hacerlo con iconos
- Los Formula Node permite escribir ecuaciones complicadas usando instrucciones basadas en texto.
- Se encuentra en la subpaleta de Structures
- Para agregar variables, se presiona el botón derecho del mouse y se escoge la opción Add Input o Add Output
- Las declaraciones deben terminar con un punto y coma (;)



Gráficos

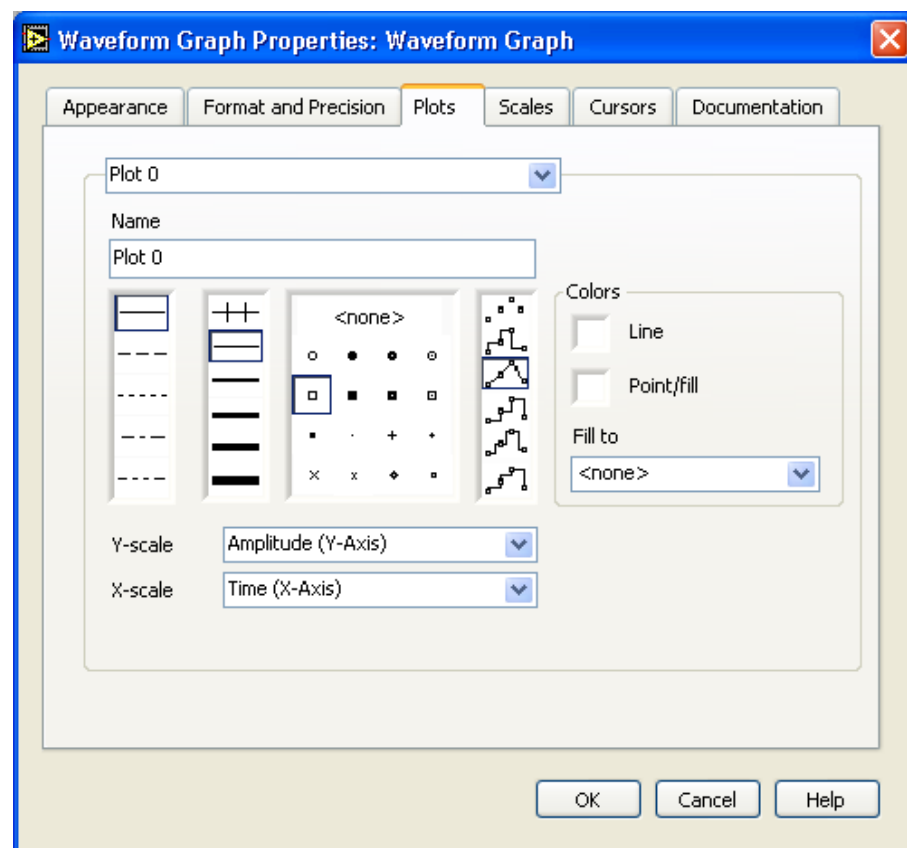
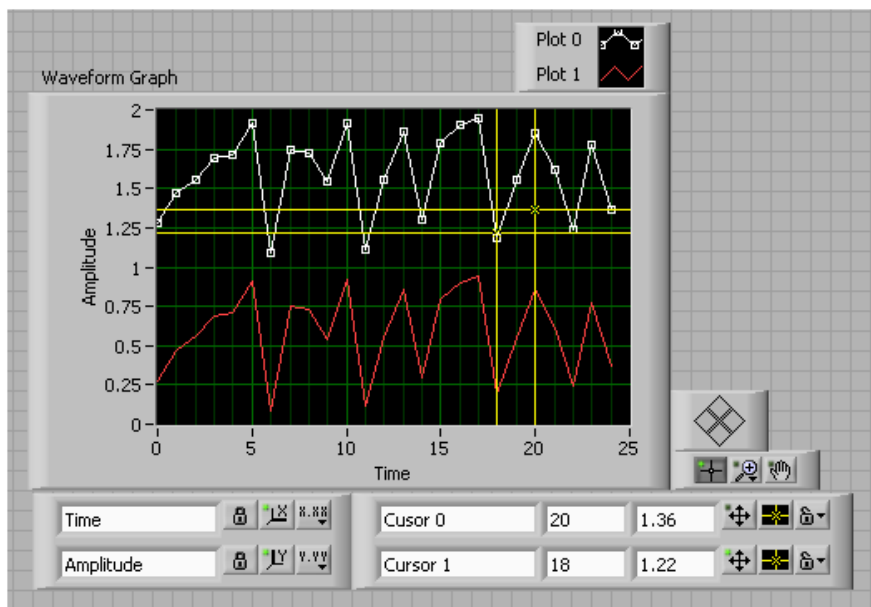
Gráfico Waveform:
indicador numérico
especial que puede
mostrar una historia de
valores



Se encuentra en la paleta
Controls»Graph Indicators.
Los gráficos Waveform
muestran gráficos simples o
múltiples. Se puede cambiar
los valores min y max del eje x
e y y otras propiedades

Gráficos - Propiedades

Para personalizar interactivamente el componente gráfico hacer clic derecho en el gráfico y seleccionar Properties





Cableando datos a las Gráficos

Se puede cablear una salida escalar directamente a un gráfico Waveform para mostrar un diagrama. Para mostrar múltiples diagramas en un gráfico, se usa la función Merge Signals encontrada en la paleta Functions >> Signal Manipulation.

Gráfico de diagrama simple

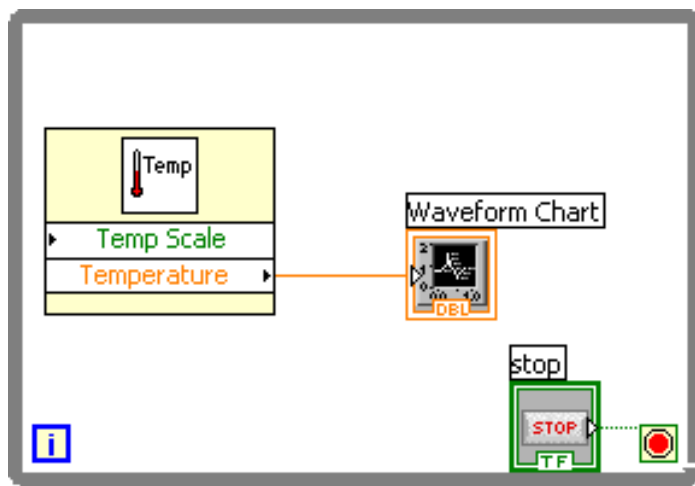
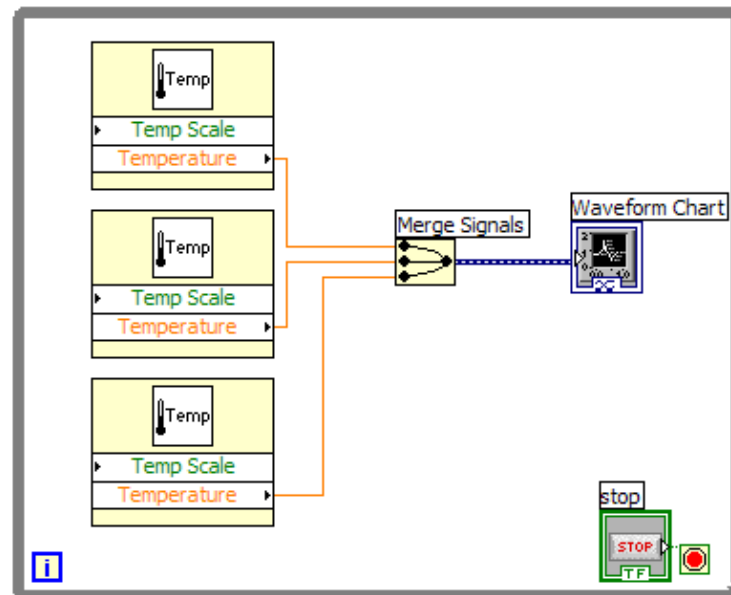
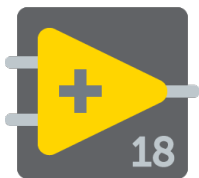


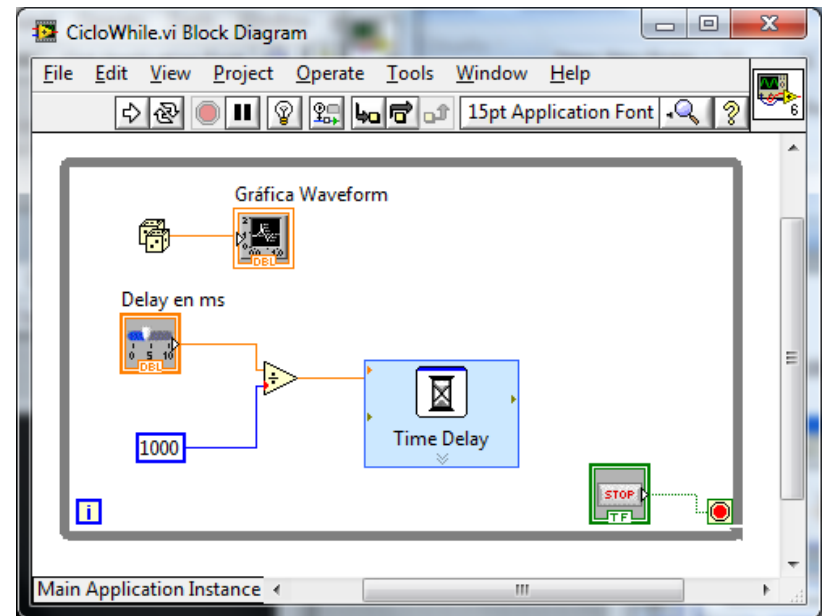
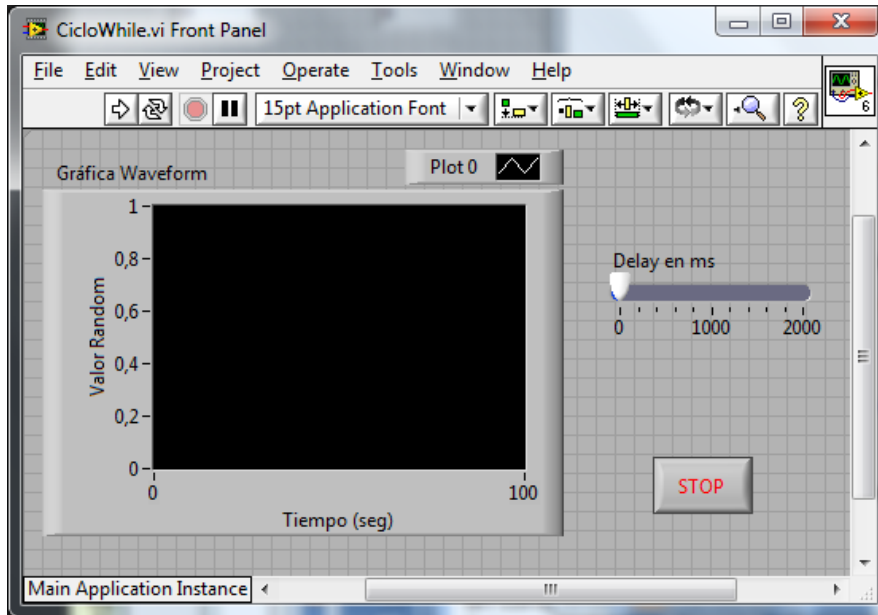
Gráfico de diagrama múltiple





Ejercicio 2: Uso de ciclos y gráficos

Generar un número aleatorio a una velocidad específica y mostrar las lecturas en una gráfica de forma de onda hasta que el usuario lo pare. Agregar un Waveform Chart y un slider control al panel de control con rango 0 a 2000 conectado a la función Time Delay dentro de su ciclo While. Guardar el VI como CicloWhile.vi





Otros tipos de Gráficos

Los gráficos situados en la paleta Controls» All Controls» Graph incluyen el gráfico Waveform y el gráfico XY. El gráfico Waveform traza funciones de una sola variable con puntos distribuidos uniformemente a lo largo del eje x. Los gráficos XY muestran cualquier set de puntos, muestreados uniformemente o no.

Gráfico Waveform – Gráfico de un array de números vs. sus índices

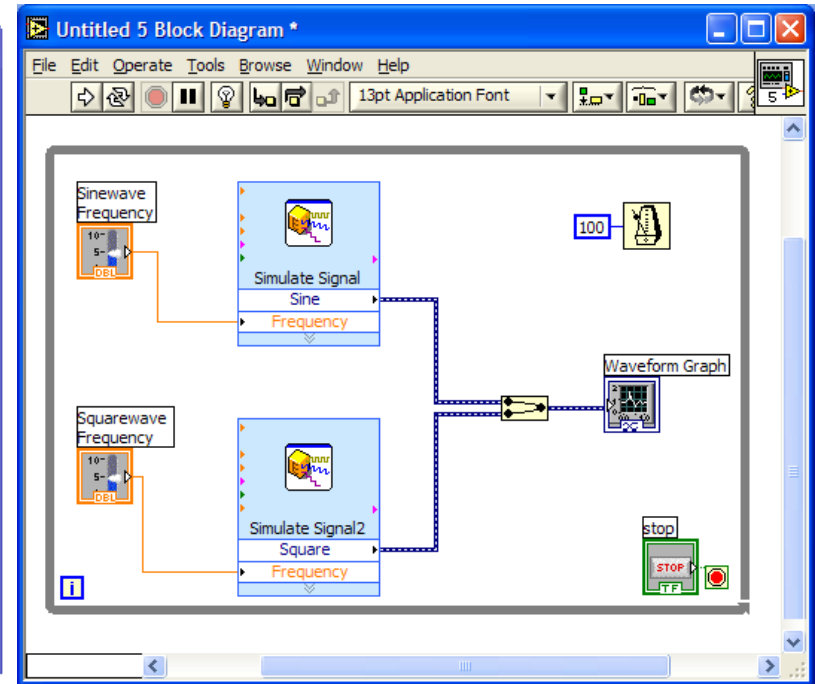
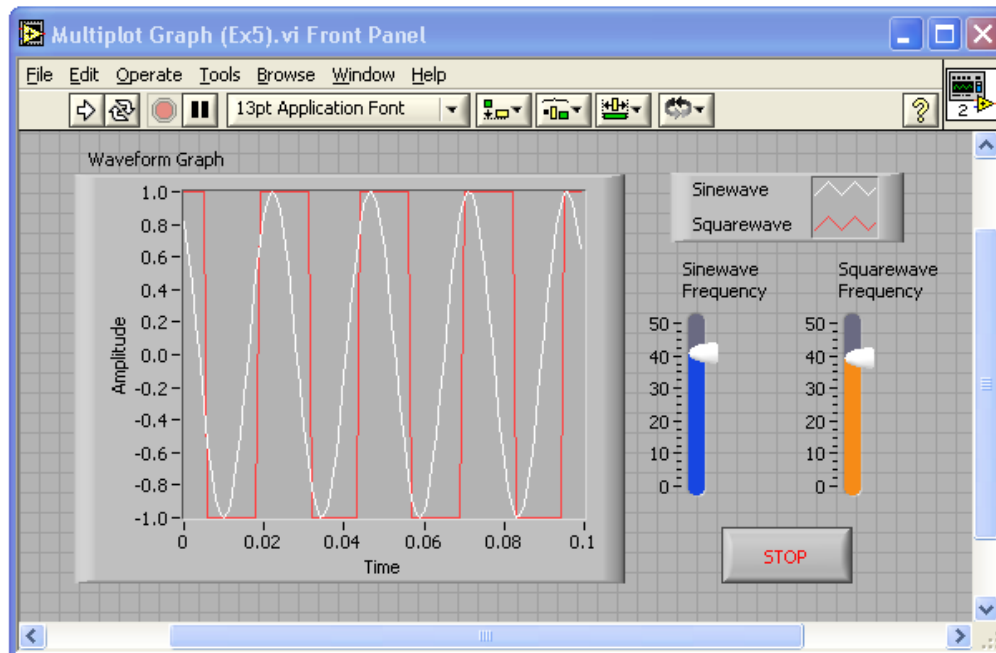
Gráfico Express XY – Gráfico de un array vs. otro

Gráfico Digital Waveform – Gráfico de datos binarios



Ejercicio: Gráficos múltiples

Crear un VI con un Ciclo While con 100 ms de duración para generar formas de onda seno y cuadradas. Utilizar el VI express Simulate Signal de la paleta Functions» Input para generar las señales. La frecuencia para cada función es elegida por el usuario. Se usa gráfico Waveform múltiple.



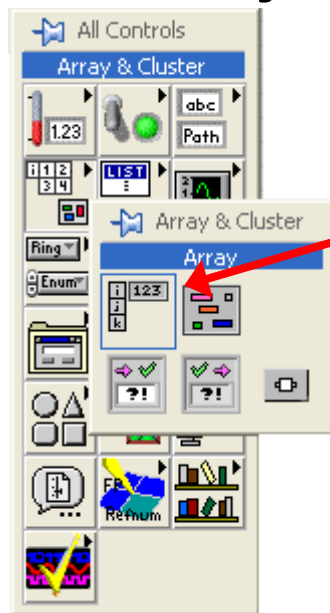


Arrays

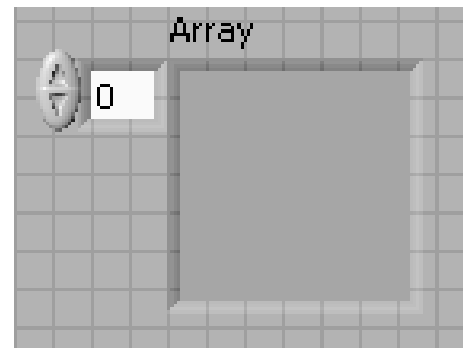
- Los arrays agrupan datos del mismo tipo. Un array consiste de elementos y dimensiones. Los elementos son los datos que componen el arreglo. Un array puede tener una o más dimensiones ($2^{31} - 1$)
- Se puede construir arrays de tipos de datos numéricos, Booleanos, rutas, cadenas, formas de onda y cluster data.
- Los arrays son ideales para almacenar información que se obtiene de formas de ondas o información generada en los ciclos, donde cada iteración de un ciclo produce un elemento del array.

Arrays

- Para acceder a los elementos del array se usa un índice, que empieza en cero (rango de 0 a $n - 1$, donde n es el número de elementos en el array).
- Para agregar un array: sub-paleta de **Controls** » **All Controls** » **Array and Cluster**, seleccionar el **Array Shell**

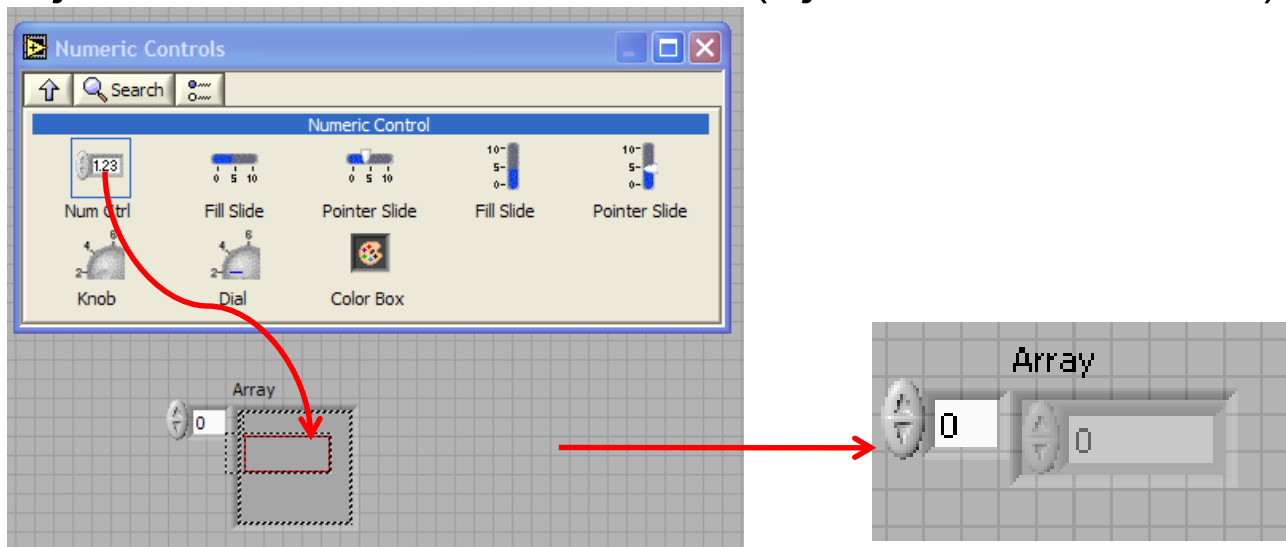


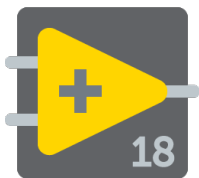
Colocararlo en el panel



Arrays

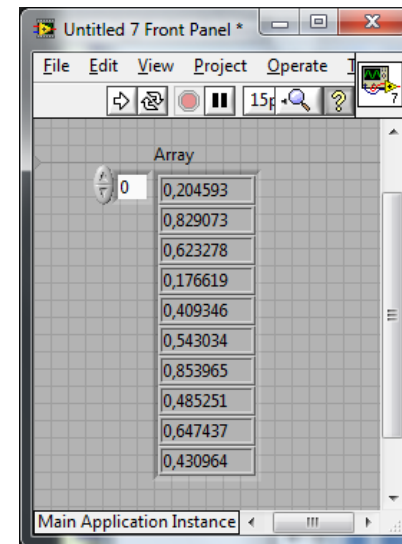
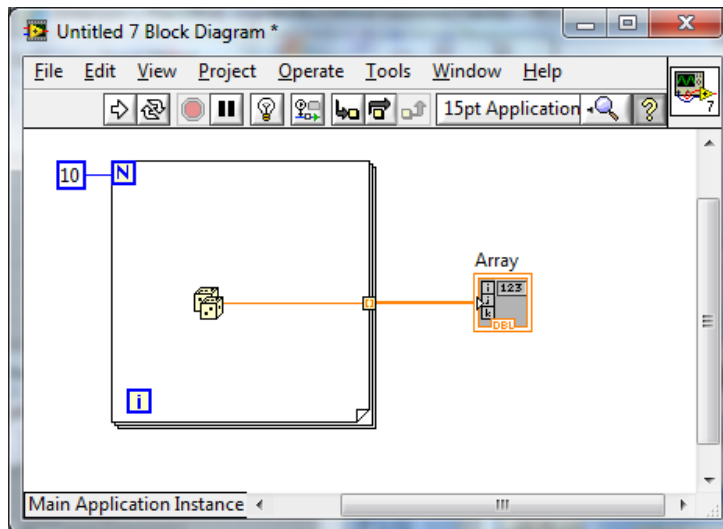
- Para añadir elementos se arrastra un control o un indicador dentro del array y se redimensiona. No es posible añadir dentro del array controles inválidos.
 - Para aumentar las dimensiones de un array, hacer clic derecho en el índice y seleccionar Add Dimension.
- Situar el objeto de datos dentro del shell (Ej. Control Numérico)





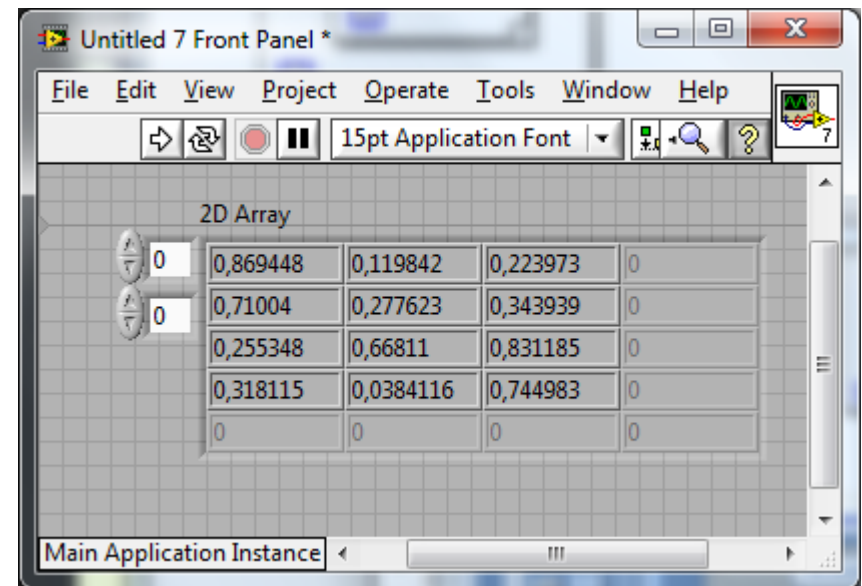
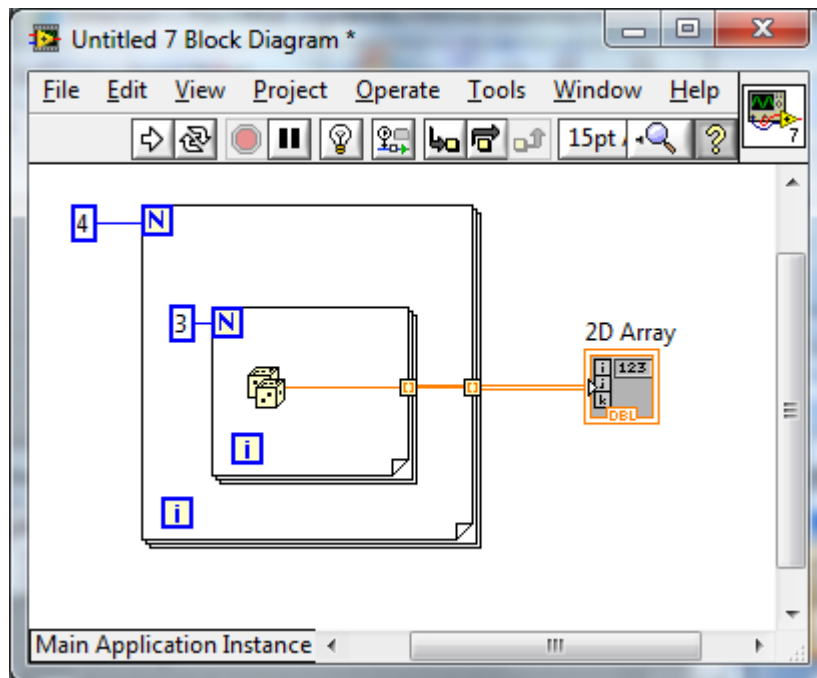
Creando un array con un ciclo

- Si se conecta un array a un túnel de entrada de un ciclo For o un ciclo While se puede leer y procesar cada elemento en ese array al activar el auto-indexing.
- Cuando se utiliza el auto-índice en un array de túnel de salida el array de salida recibe un nuevo elemento por cada iteración del ciclo.



Creando un array bidimensional

- Se pueden utilizar dos Ciclos For, uno dentro del otro, para crear arrays bidimensionales. El Ciclo For externo crea las filas de elementos y el Ciclo For interno crea las columnas de elementos.

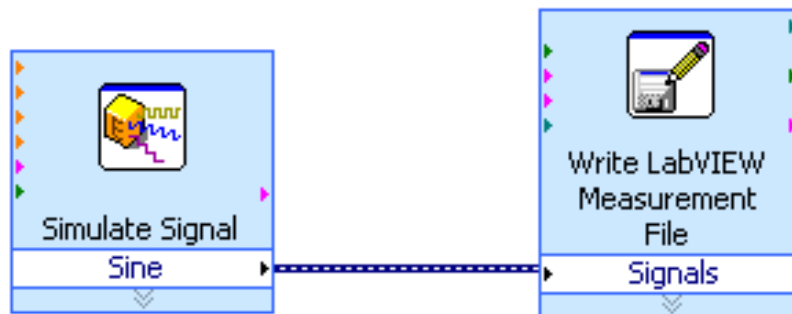




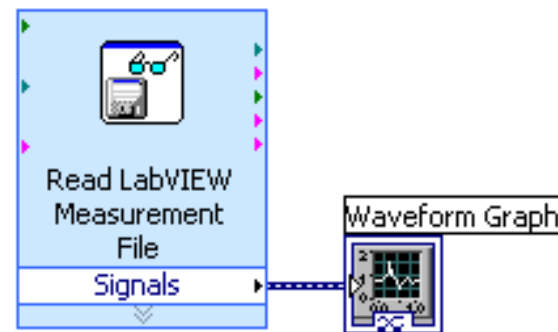
Archivos

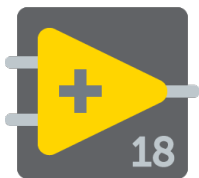
- Las operaciones de Entrada/Salida de archivo (I/O) transfieren datos desde y hacia archivos. Se usan los VI de File I/O y las funciones localizadas en la paleta Functions » File I/O para manejar todos aspectos de I/O de archivo
- Los Archivos pueden ser binarios, texto, hojas de cálculo o Archivos LabVIEW Measurements (*.lvm)

Escritura en Archivo LVM



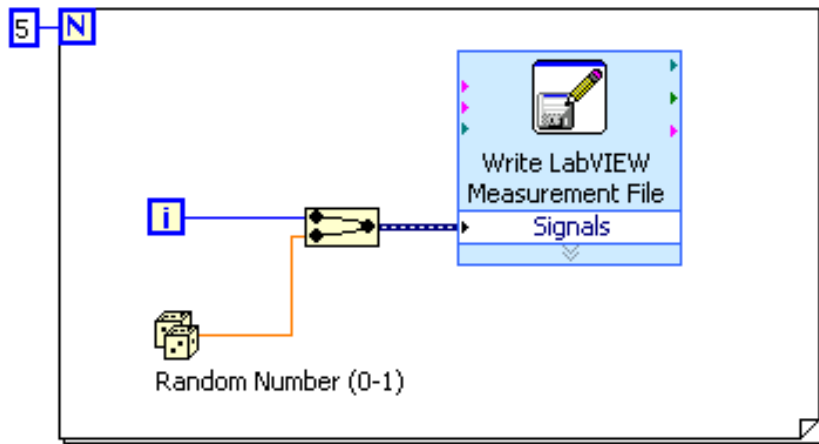
Lectura de un Archivo LVM



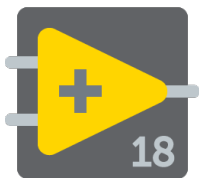


Escritura de Archivos LabVIEW Measurement

- Incluye las funciones de apertura, escritura, cierre y manejo de errores en la paleta VIs expreso: Write LVM y Read LVM
- Maneja el formateo de cadenas de caracteres (strings) ya sea con delimitación por medio de tabuladores o comas
- La función de concatenar señales es usada para combinar datos en datos de tipo dinámico

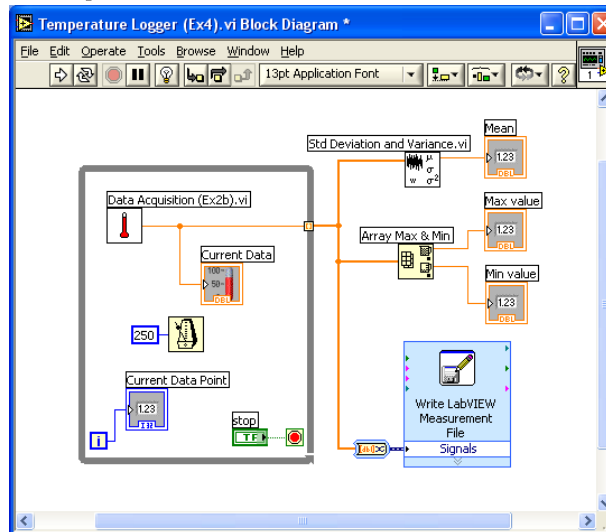


	A	B	C	D
1		0	0.385055	
2		1	0.23516	
3		2	0.985184	
4		3	0.177893	
5		4	0.935915	
6				
7				



Ejercicio 3: Análisis e Introducción de datos

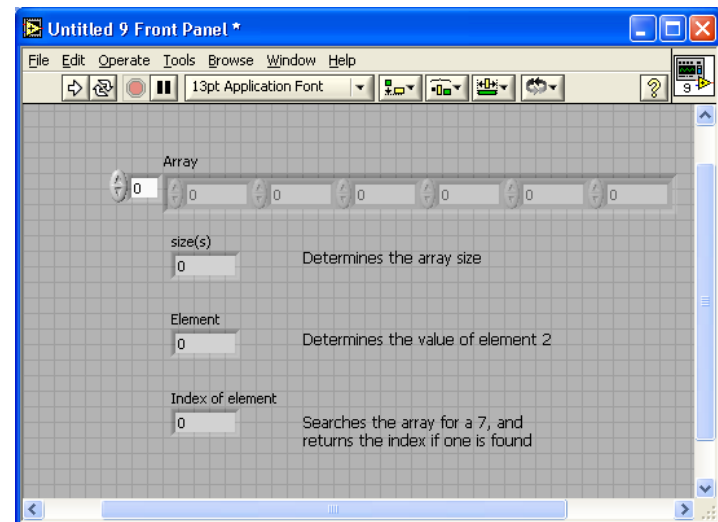
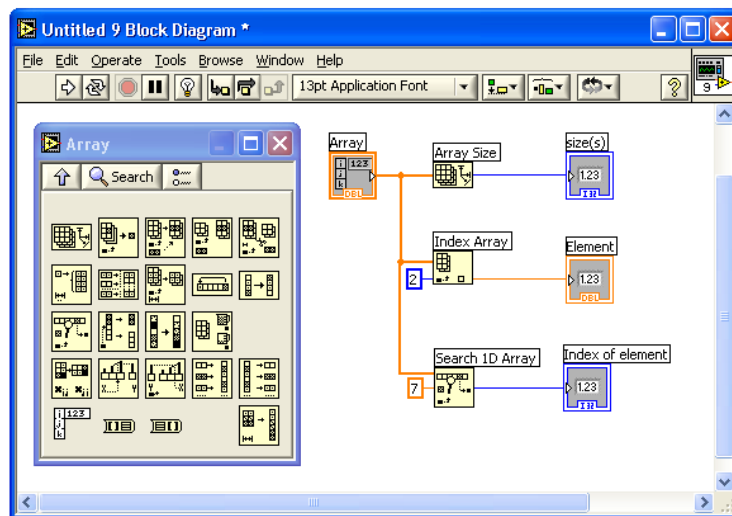
Crear un VI que adquiera y exhiba datos de temperatura a una velocidad fija hasta que el usuario lo detenga. Utilizar el Digital Thermometer.vi de la paleta de funciones. Después de ejecutarse, el VI realiza análisis en los datos colectados: máximo, mínimo, y valor medio y se muestran en indicadores numéricos. Guardar los datos utilizando el Write Measurements File Express VI.





Funciones para arrays

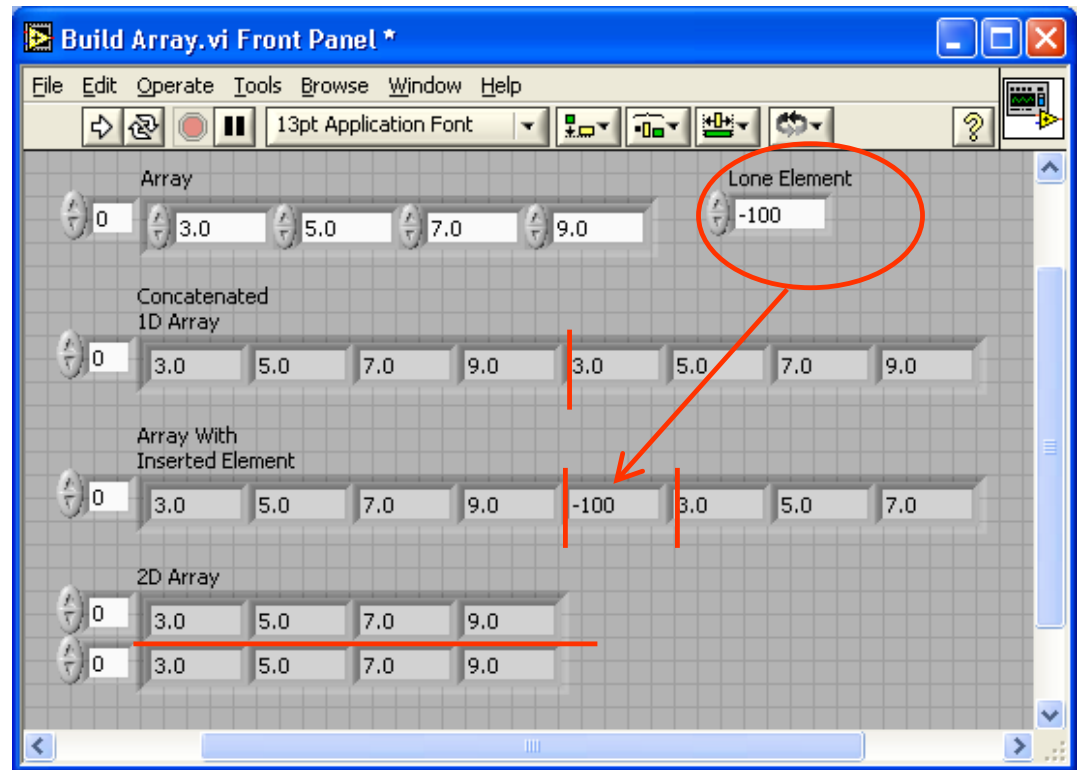
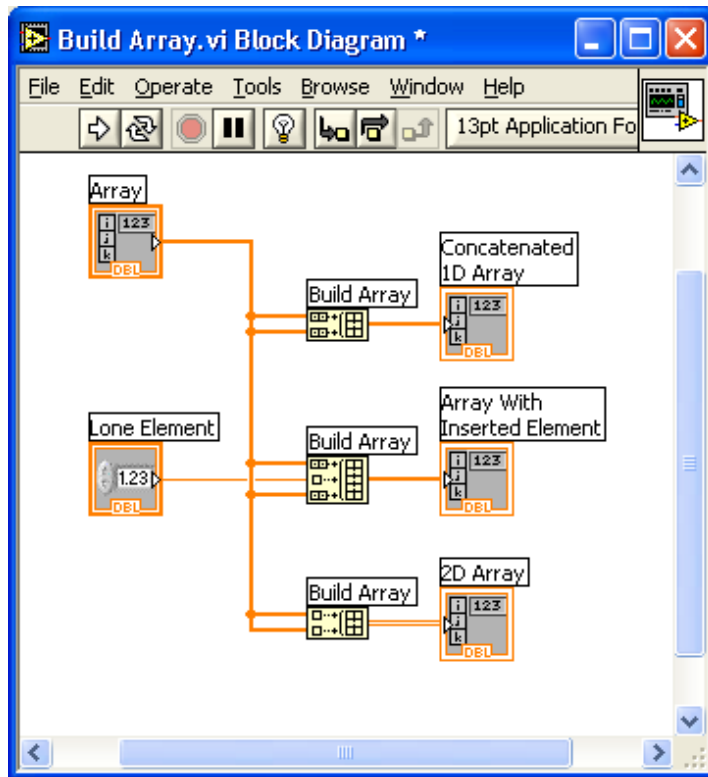
- Las funciones para crear y para manipular arrays están en la paleta Functions» All Functions» Array. Algunas son:
 - Array Size
 - Initialize Array
 - Build Array
 - Array Subset
 - Index Array

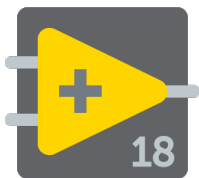




Funciones para arrays – Build Array

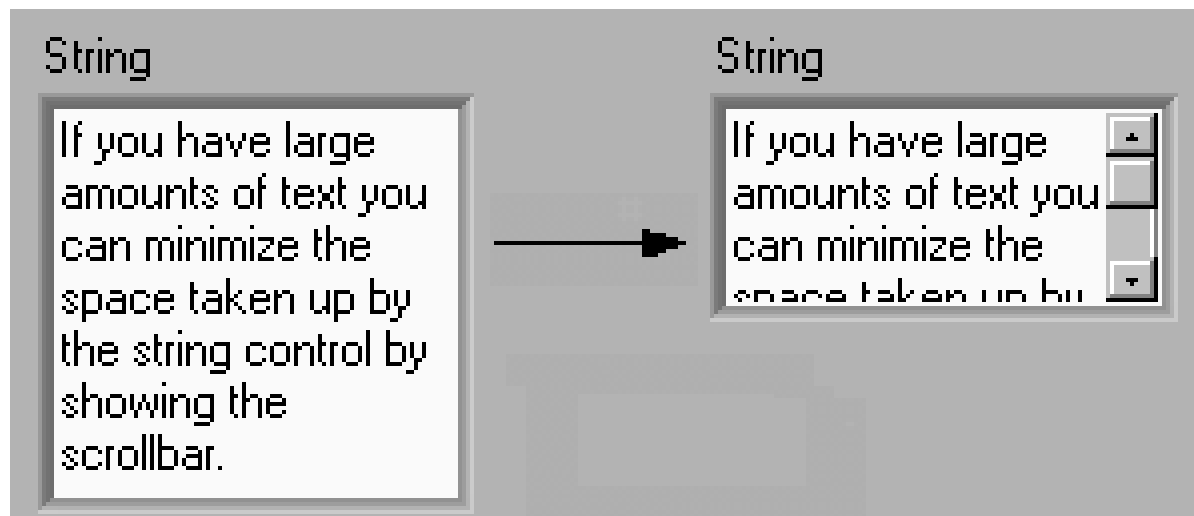
- Build Array permite concatenar múltiples arrays o añade elementos a un array de n dimensiones.





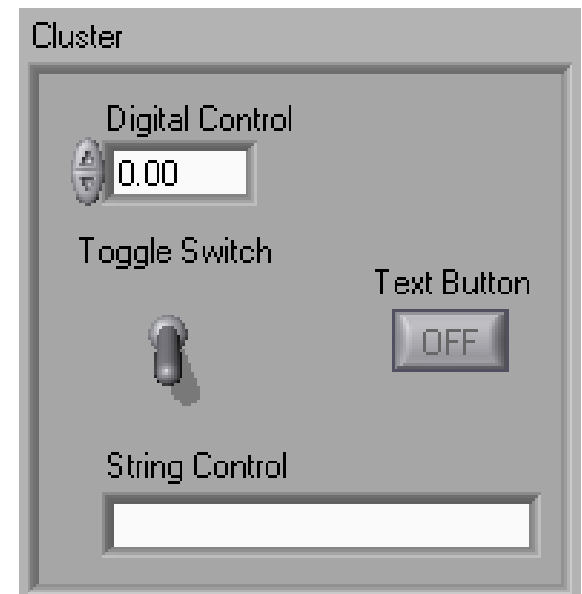
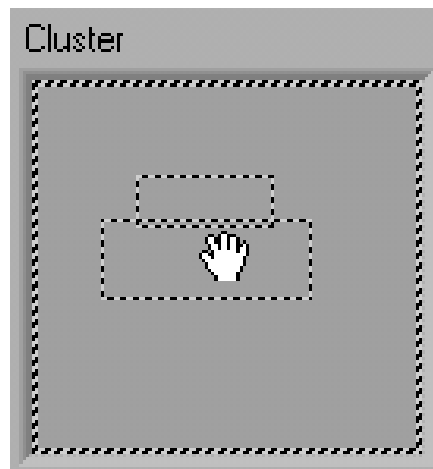
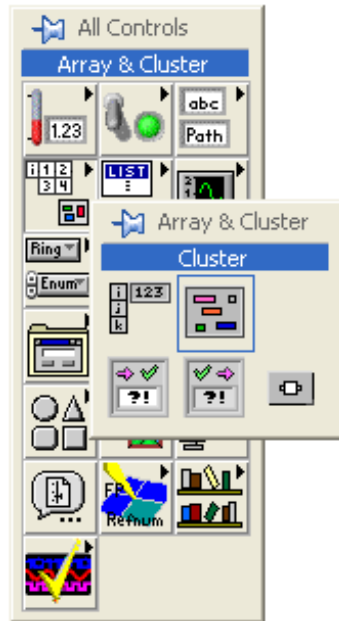
Strings – cadenas de caracteres

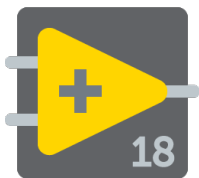
- Una cadena es una secuencia de caracteres visibles o no visibles (ASCII)
- Tienen varios usos: mostrar mensajes, control de instrumentos, archivos de entrada y Salida (I/O)
- El controlador/indicador de cadenas se encuentra en **Controls »Text Control o Text Indicator**



Clusters

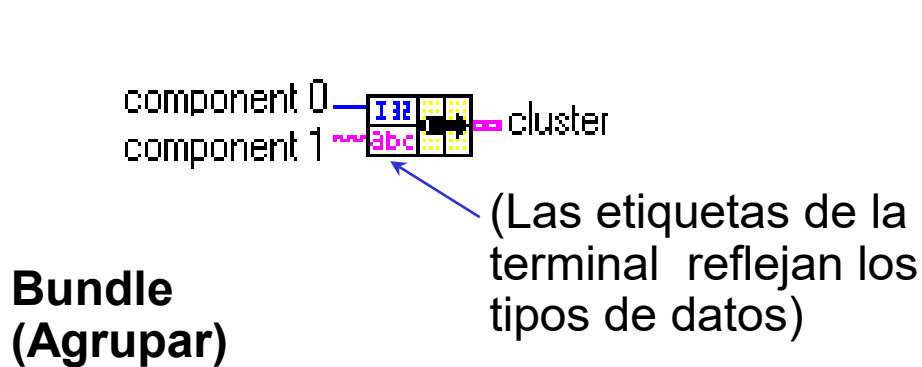
- Agrupa componentes de diferentes tipos, semejante a una estructura (struct) en C. Los elementos deben de ser todos Controles o todos Indicadores.
- Para crear un cluster seleccionar **Controls » All Controls » Array & Cluster** y colocar objetos dentro



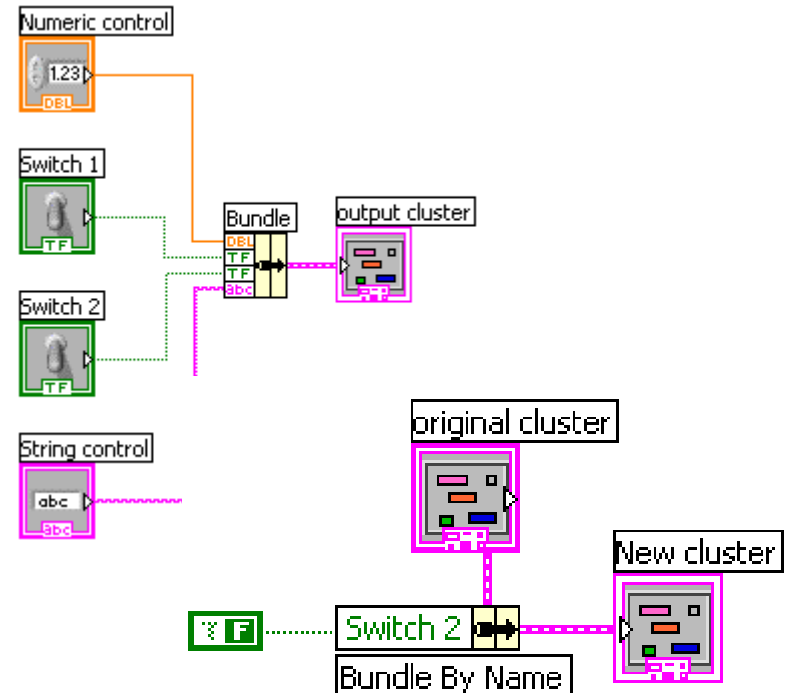


Funciones del Cluster

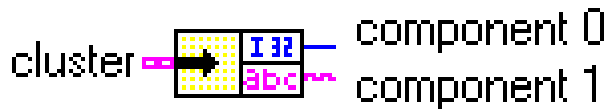
- Están ubicadas en la subpaleta de Cluster de la paleta Functions>>All functions.
- También puede ser accesada haciendo clic con el botón derecho del mouse en la terminal del Grupo.



**Bundle by name
(Agrupar por nombre)**



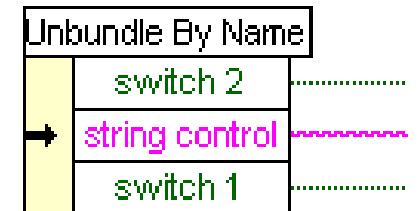
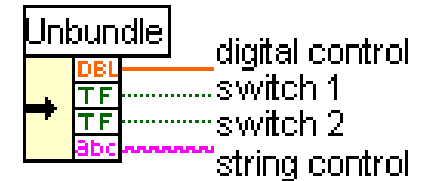
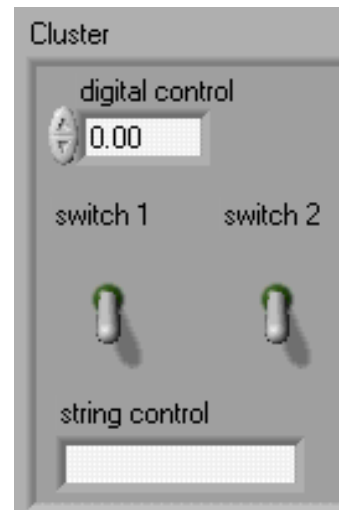
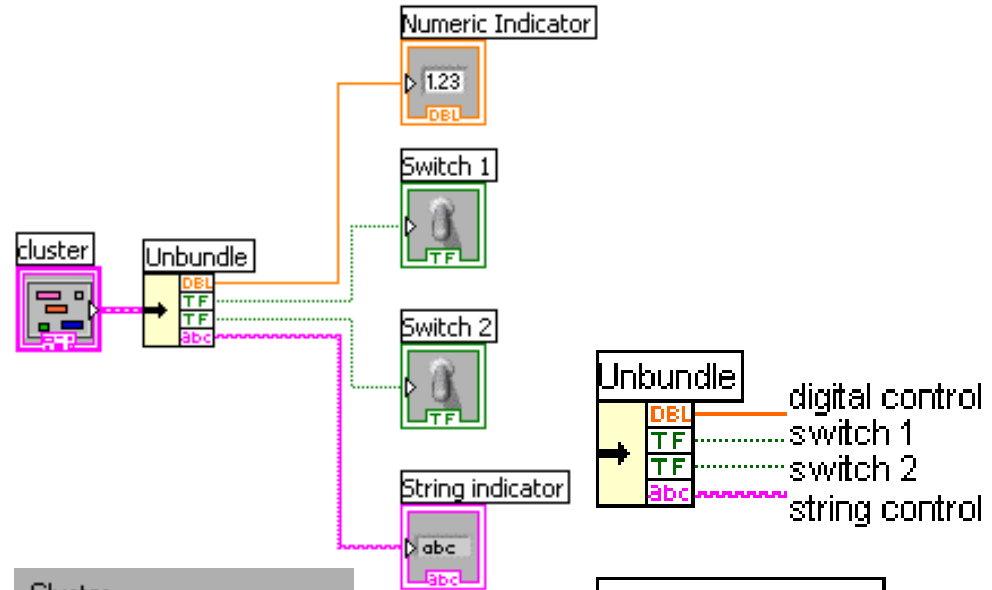
Funciones del Cluster



**Unbundled
(separar)**



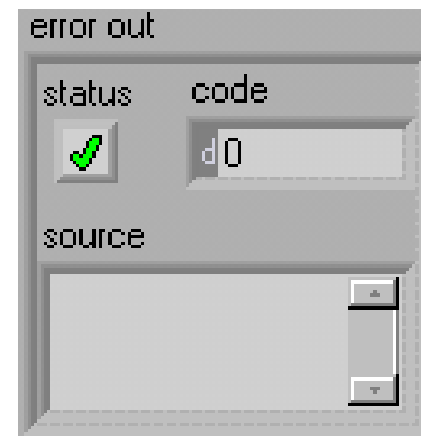
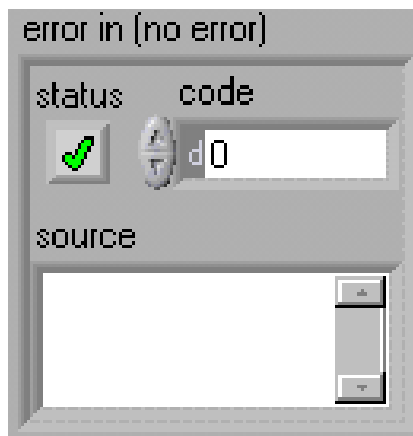
**Unbundled by name
(separar por nombre)**



**Ejemplo de un
Cluster separado
(Unbundled Cluster)**

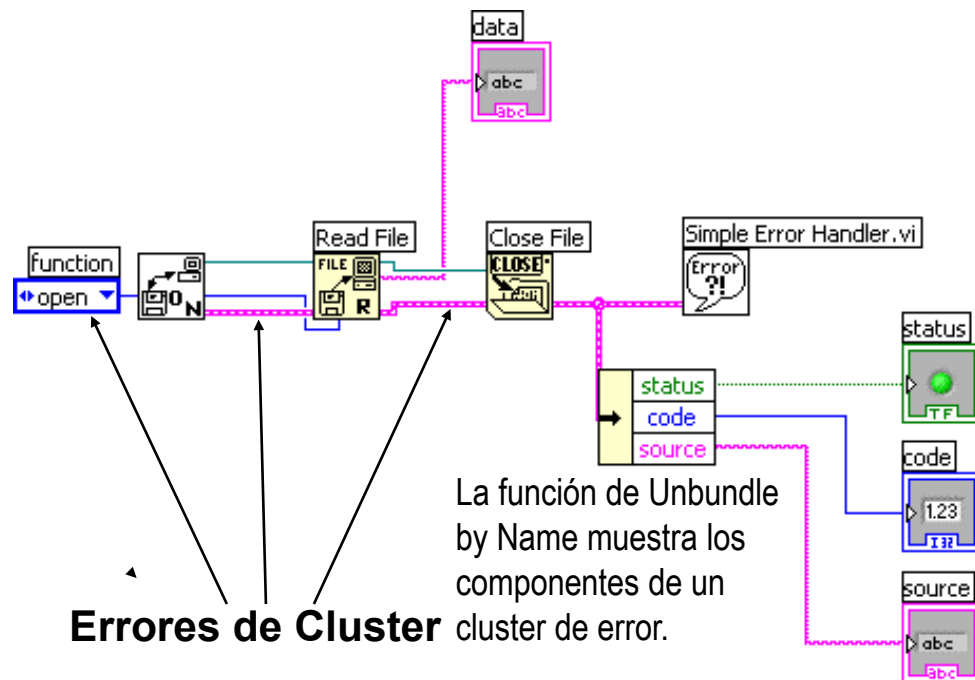
Errores de Cluster

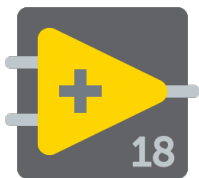
- Son un medio poderoso para manejar errores de los VI de adquisición de datos, las funciones de manejo de entradas y salidas de archivo, los VI de networking, y muchos otros VI
- El cluster para errores contiene los siguientes elementos:
 - **status**, booleano que toma un valor verdadero si ocurre algún error.
 - **code**, valor numérico con un código de error.
 - **source**, cadena de caracteres que identifica el VI del error.



Técnicas de manejo de errores

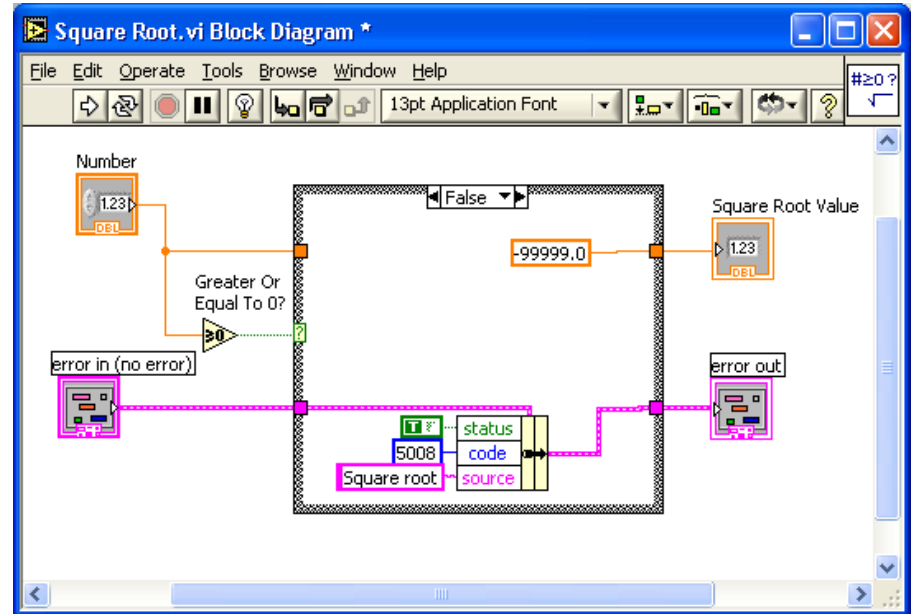
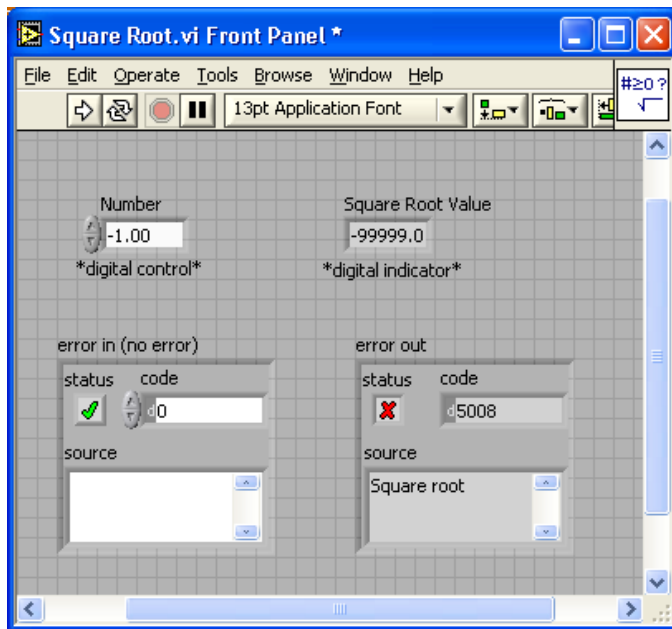
- La información de un Error es transmitida de un subVI al siguiente
- Si un Error ocurre en un subVI, todos los subVI subsecuentes no serán ejecutados de la manera usual
- Los errores de Cluster contienen todas las condiciones del Error
- Manejo automático de Errores

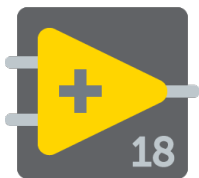




Ejercicio: Errores de Cluster y su Manejo

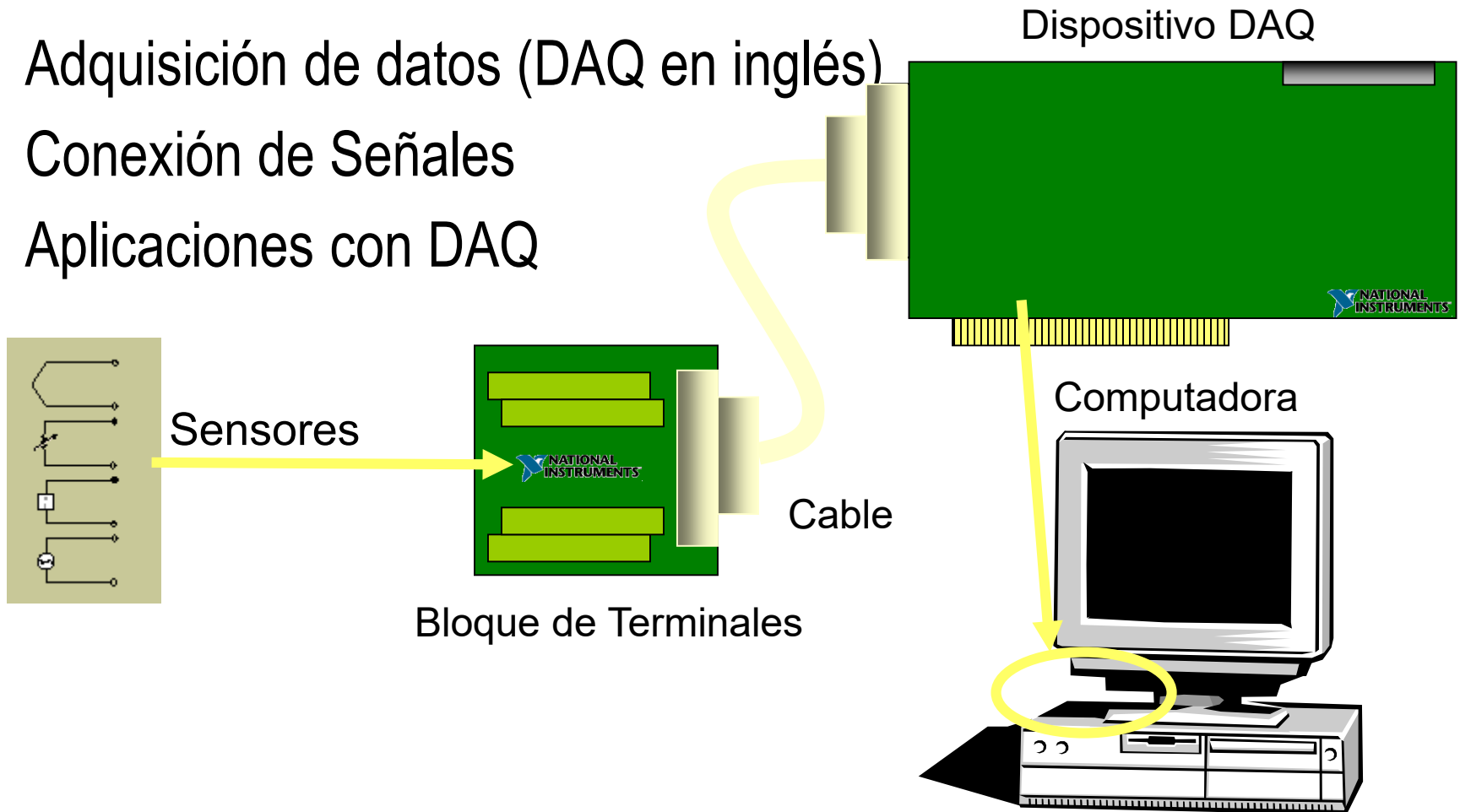
Crear un VI para calcular la raíz cuadrada de un número. Si el numero es mayor o igual a cero, el VI debe devolver el valor de la raíz cuadrada y no generar ningún error. Si el número es menor que cero debe devolver un resultado con valor de -9999.90 e insertar un error dentro del cluster para errores

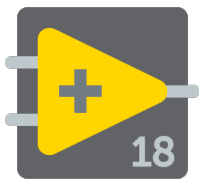




Adquisición de Datos en LabVIEW

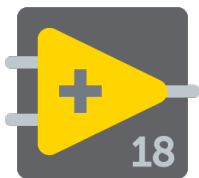
- Adquisición de datos (DAQ en inglés)
- Conexión de Señales
- Aplicaciones con DAQ





Adquisición de Datos en LabVIEW

- Contiene una plataforma para la NI-DAQ tradicional y una para NI-DAQmx.
 - Los VIs tradicionales se dividen por el tipo de medición, los VIs DAQmx se dividen por el tipo de tarea.
- Pasos para usar la Adquisición de Datos de VIs
 - El software NI-DAQ debe estar instalado
 - Tener instalado una tarjeta E-series DAQ y configurada con el Explorador de Automatización & Mediación (Measurement & Automation Explorer (MAX)).

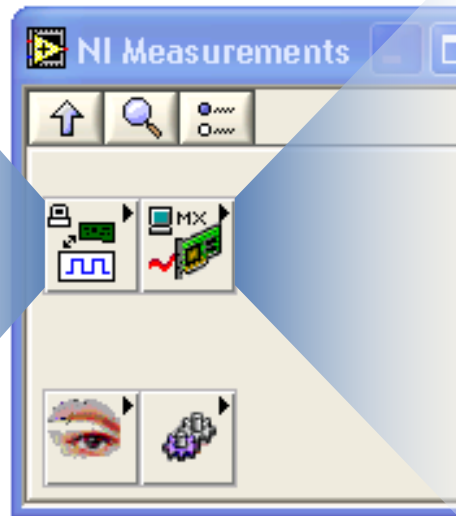


Adquisición de Datos en LabVIEW

NI-DAQ Tradicional

VIs específicos para realizar:

- Entrada Análoga
- Salida Análoga
- I/O (entrada/salida) Digital
- Operaciones de conteo



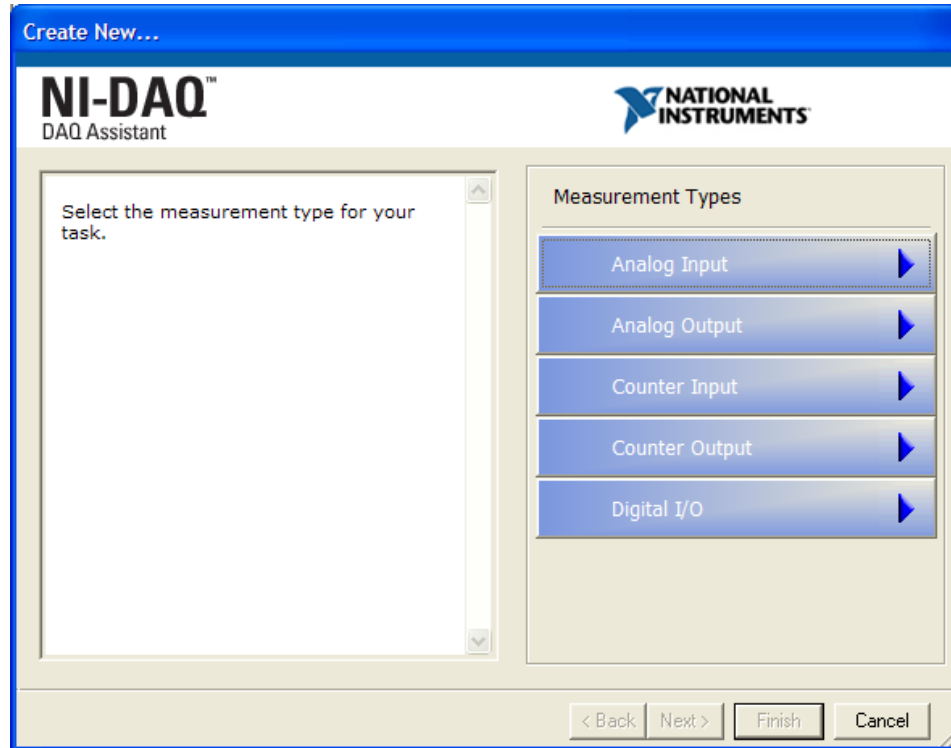
NI-DAQmx

Controlador (Driver) de siguiente generación:

- VIs para ejecutar una tarea
- Un serie de VIs para todos los tipos de mediciones

Adquisición de Datos en LabVIEW

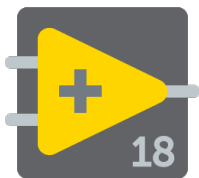
- LabView proporciona herramientas de ayuda para la lectura o generación de señales desde dispositivos





Términos en Adquisición de Datos

- **Resolución:** para adquirir datos se usan convertidores Análogo-a-Digital (ADC) que toma una señal análoga y la convierte a un número binario. La resolución se refiere al número de niveles binarios que el ADC puede utilizar para representar una señal
- **Rango:** A diferencia de la resolución del ADC, el rango del ADC es seleccionable. Se elige el rango cuando se configura su dispositivo en NI-DAQ
- **Gain (ganancia)** – Amplifica o atenúa la señal para un mejor ajuste del rango



Conexiones Hardware en Adquisición de Datos

BNC-2120



SC-2075



SCB-68

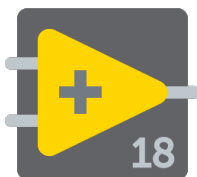


NI-ELVIS



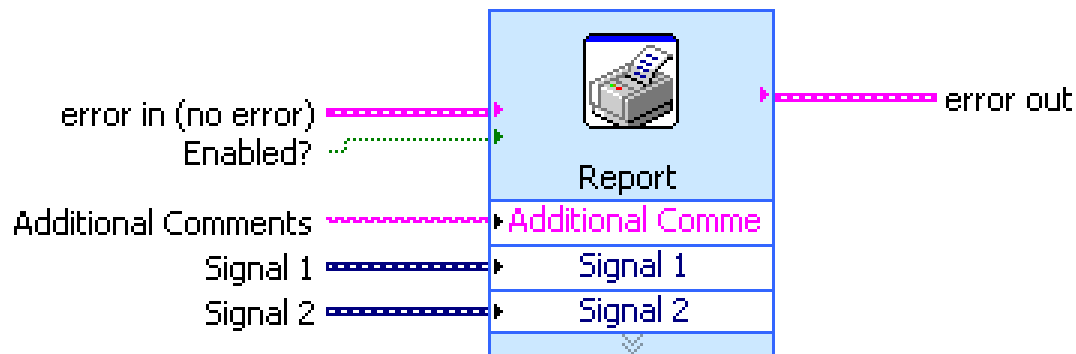
NI-WLS-9211

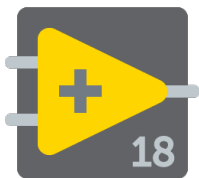




Impresión

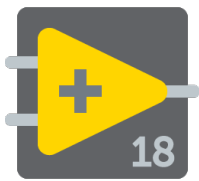
- La opción **File » Print...**, permite imprimir una copia del VI, o se puede generar una copia para guardarla en un archivo o publicarla.
- El VI Print Panel (**Functions » All Functions » Application Control**) permite imprimir programáticamente los resultados
- Hay herramientas que generan reportes como el VI **Express » Output » Report**.





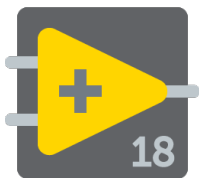
Documentación de VIs

- VI Properties » Documentation
 - Provee una Descripción e Información de Ayuda para el VI
- VI Properties » Revision History
 - Comprueba cambios entre diferentes versiones de un VI
- Individual Controls » Description and Tip...
 - Clic con el botón derecho del mouse para proveer una descripción y ayuda (Description and Tip Strip)
- Conviene utilizar la herramienta de etiquetado (Labeling Tool) para documentarlos paneles frontales y diagramas de bloques



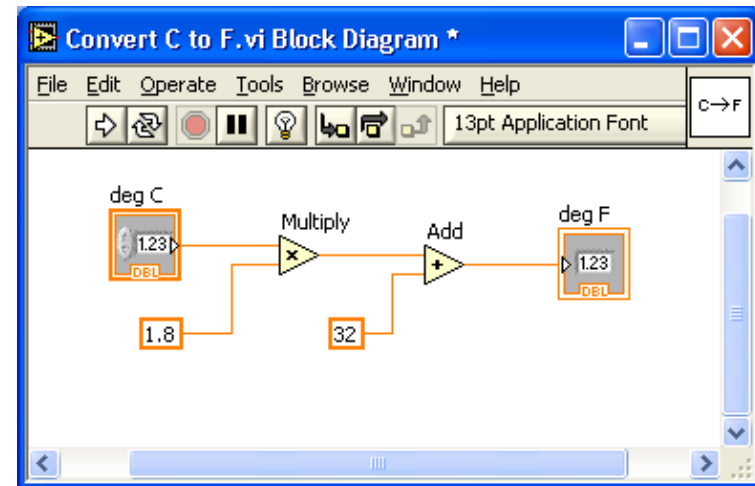
Arquitectura básica de Programación

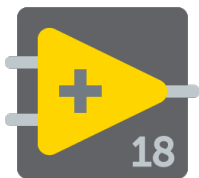
- Se puede estructurar los VIs dependiendo de la funcionalidad que desea que tengan.
- Algunas de las arquitecturas de VI más comunes son:
 - Arquitectura de un VI simple
 - Arquitectura de un VI general
 - Arquitectura de máquina de estado



Arquitectura de un VI simple

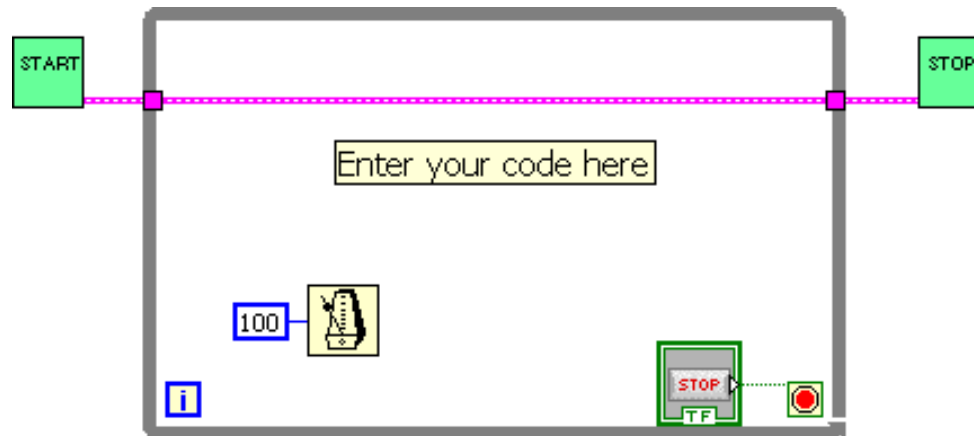
- En pruebas de laboratorio rápidas no se necesita una arquitectura complicada: consistente de un solo VI que mide, hace cálculos, y despliega los resultados o los guarda en el disco. La medición se inicia cuando se pulsa run arrow.
- Esta arquitectura también se usa para componentes “funcionales” dentro de aplicaciones más grandes.
- Ejemplo: Convertir C a F.vi





Arquitectura de un VI general

- Generalmente al diseñar una aplicación se siguen tres pasos:
 - **Startup**: para inicializar el hardware, leer configuración o preguntar por la localización de datos de archivos
 - **Main application**: generalmente consiste en un ciclo que se repite hasta que el usuario decide salir del programa, o termina por otras razones como la terminación de una entrada/salida (I/O)
 - **Shutdown**: para cerrar archivos, escribir datos de configuración o reestablecer los datos de entrada/salida (I/O)





Arquitectura de máquina de estado

- Se pueden hacer diagramas más compactos con el uso de una estructura Case simple para manejar todos los eventos.
- Ventajas
 - Fácil de modificar y depurar
- Desventajas
 - Se pierden eventos si dos ocurren al mismo tiempo

Pasos:

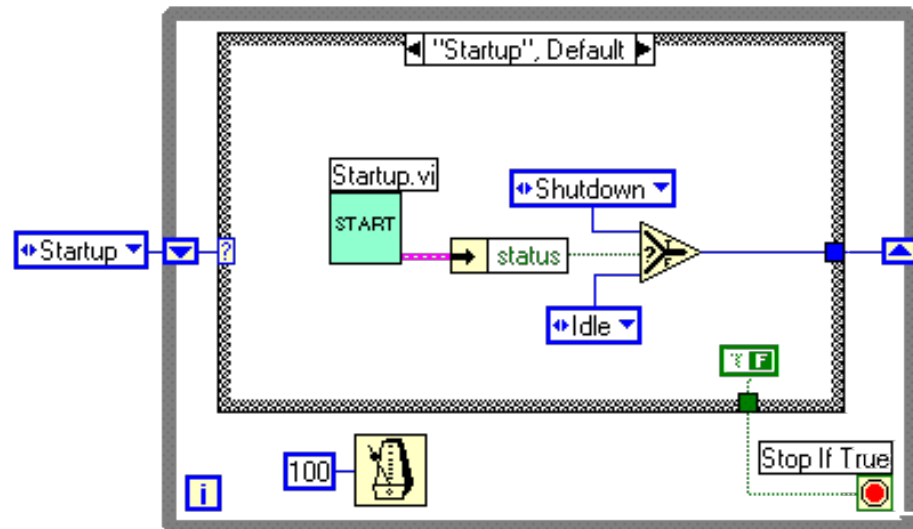
0: Inicio (Startup)

1: Ocio (Idle)

2: Evento 1

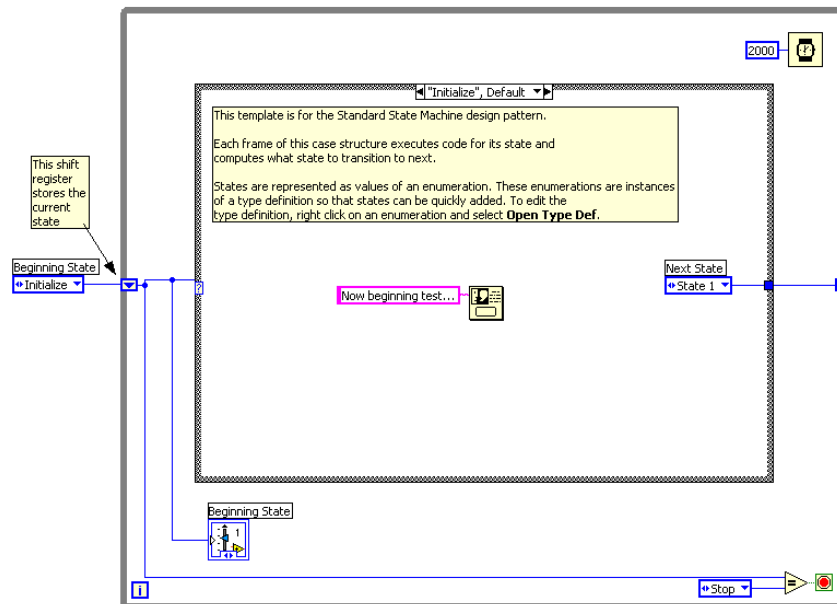
3: Evento 2

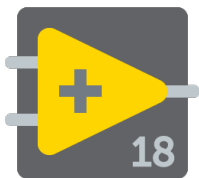
4: Cierre (Shutdown)



Ejercicio: máquina de estado

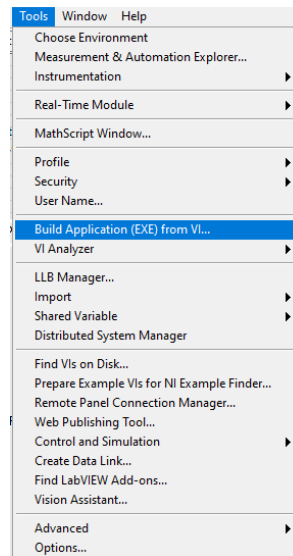
- Crear un VI usando la arquitectura de máquina de estado. Cuando se pulsa un botón, se va al estado 1 que genera una diálogo que permite ir al estado 2 o volver a empezar. En el estado 2 se puede cerrar el programa o iniciar nuevamente.
- Se usa la plantilla de VI State Machine.vit





Generación de programas ejecutables

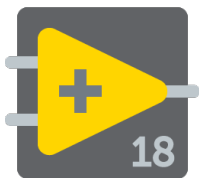
- Para generar un ejecutable (.exe) a partir de las VIs de LabView es necesario disponer del Application Builder add-on.
- A partir de un vi o un proyecto se utiliza Tools » Build Application (EXE) from VI y se configure los elementos necesarios.





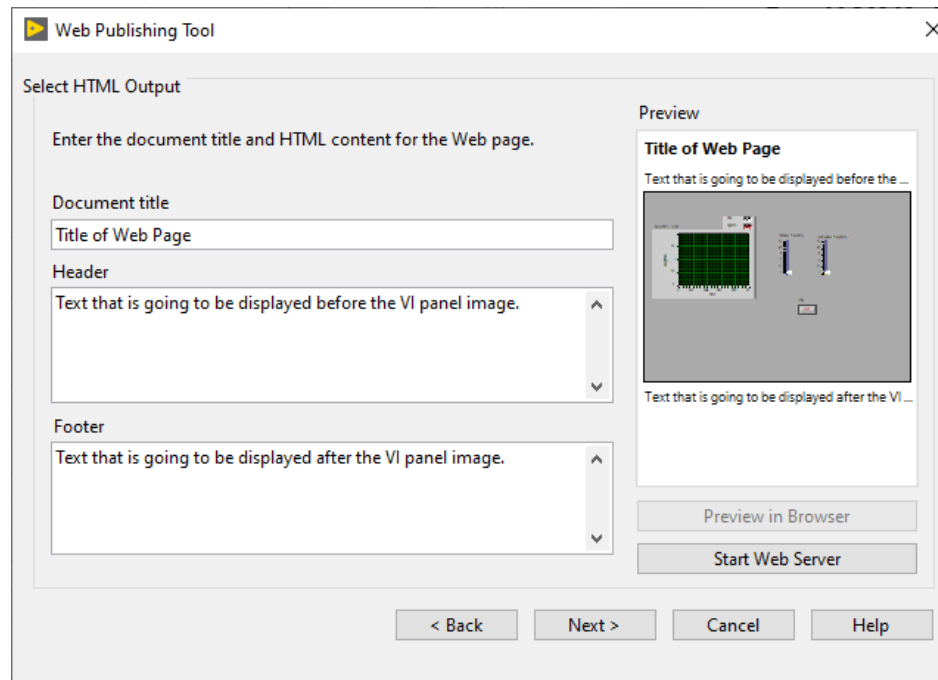
Panel frontal remoto

- LabView permite observar y controlar paneles frontales desde un Navegador Web
- No requiere programación
- Usuarios en lugares remotos pueden ver en tiempo real las actualizaciones del panel frontal
- Múltiples clientes pueden observar el panel frontal simultáneamente
- Sólo un cliente puede controlar el panel frontal a la vez



Herramienta para publicar un panel remoto

- Abrir la herramienta Web Publishing Tool en el menú Tools. Guardar el archivo html en el directorio www del servidor web de LabVIEW.
- Se puede personalizar la página web con otro editor de html.





Temas adicionales

- LabVIEW tiene muchas otras herramientas disponibles para su desarrollo.
 - **Property Nodes:** se crean o leen las cualidades de un objeto en el panel (color de foreground/background, formato y precisión de datos, visibilidad, texto descriptivo, tamaño y posicionamiento, etc.)
 - **Local Variables:** permiten que los datos se pasen sin necesidad de cables.
 - **Global Variables:** tipo especial de VI usado para guardar datos en objetos del panel frontal para intercambiar datos entre VIs.
 - **DataSocket:** plataforma independiente de intercambio de datos entre computadoras y aplicaciones, basada en TCP-IP.
 - **Archivos Binarios I/O:** permite transferir datos a un disco en aplicaciones de alta velocidad.



Más información y/o ejemplos

- Ejemplos de Programas (Help » Find Examples...) Nota: es necesario activar el servicio NI Service Locator
- LabVIEW Community Edition ([link](#))
- Programas de ejemplo ([link](#))
- Instrument Driver Network ([link](#))
- Labview Quick Reference Card ([link](#))