

LABVIEW

INTRODUCCIÓN

¿Qué es LabVIEW?

- **LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench)** es un entorno de desarrollo gráfico con funciones integradas para realizar *adquisición de datos, control de instrumentos, análisis de medidas y presentaciones de datos*.
- Es un software creado por la empresa National Instruments en 1976 y sacado al mercado en 1986.
- Al desarrollar un programa en LabVIEW, se crea un *Instrumento Virtual* o *VI* que contiene la interfaz gráfica del programa y el diagrama de bloques (código).

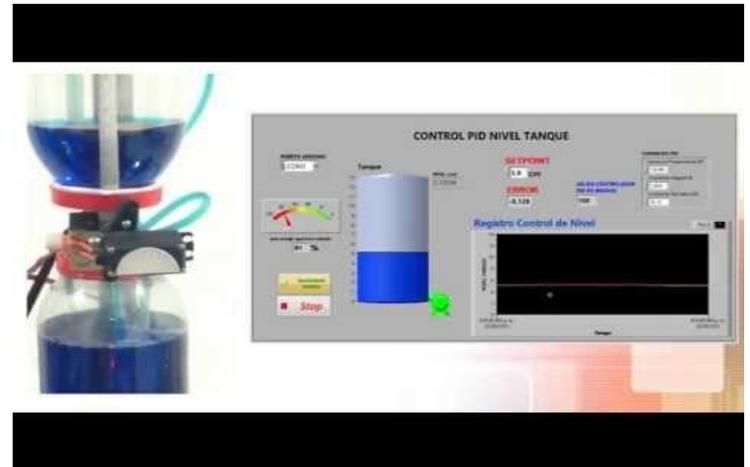
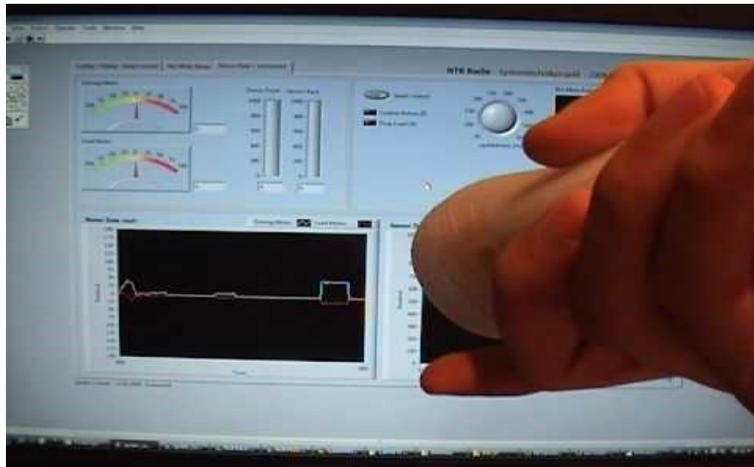
¿Qué es LabVIEW?

- Lenguaje desarrollado para medición, control y automatización
 - A diferencia de los lenguajes de propósito general, LabVIEW tiene funciones específicas para acelerar el desarrollo de aplicaciones de medición, control y automatización.
- Fácil integración con instrumentos y dispositivos de medida
 - LabVIEW se puede conectar de manera transparente con todo tipo de hardware incluyendo instrumentos, controladores lógicos programables (PLCs).
- LabVIEW para investigación y análisis
 - Puede utilizarse LabVIEW para analizar y registrar resultados reales para aplicaciones en amplios sectores orientados a la ingeniería mecánica, eléctrica, electrónica, biomédica, robótica etc.
- LabVIEW para control de procesos y automatización en fábricas.
 - Puede utilizarse LabVIEW para numerosas aplicaciones de control de procesos y automatización, realizar medidas y control de alta velocidad y con muchos canales.

¿Qué es LabVIEW?

- LabVIEW es principalmente utilizado por los ingenieros para el manejo de datos a través de la comunicación entre un ordenador y un instrumento o circuito externo, por lo que LabVIEW puede comunicarse con interfaces como:
 - Puerto serie
 - Puerto paralelo
 - GPIB
 - PXI
 - VXI
 - PCP/IP
 - IrDA
 - Bluetooth
 - USB
 - OPC

¿Qué es LabVIEW?



Interfaz



Interfaz

The image displays the LabVIEW software interface for a 'Power Spectrum' application. It is divided into two main windows: the 'Front Panel' and the 'Block Diagram'.

- Panel frontal:** This window contains the user interface. It features two vertical sliders for 'Sine Wave' and 'Square Wave' amplitude, both ranging from 0 to 40. Below these are two frequency meters for 'Sine Frequency' and 'Square Frequency', both ranging from 0 to 40. A 'STOP' button is located at the bottom. The main display area is split into two plots: 'Sine and Square Waves' showing amplitude over time, and 'Power Spectrum' showing amplitude over frequency.
- Paleta de herramientas:** A 'Tools' palette is shown on the right side of the interface, containing various icons for editing and development.
- Diagrama de bloques:** The 'Block Diagram' window shows the underlying logic. It includes 'Simulate Signal Sine' and 'Simulate Signal2 Square' blocks, which feed into a 'Spectral Measurements Signals' block. This block outputs to a 'Power Spectrum' block, which then feeds into a 'Write LabVIEW Measurement File' block. The diagram also includes a 'stop' button and a 'STOP TFL' button.
- Contexto de la ayuda:** A 'Context Help' window is shown in the bottom left, providing information about a node. It displays a diagram of an adder block with inputs 'x' and 'y' and output 'x+y', and the label 'Add'.

Annotations with arrows point to the 'Panel frontal', 'Paleta de herramientas', 'Contexto de la ayuda', and 'Diagrama de bloques'.

Paleta de herramientas



- 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 paro



Herramienta de prueba



Herramienta de etiquetado



Herramienta para copia de color



Herramienta de cableado



Herramienta para colorear



Herramienta de menú (atajo)

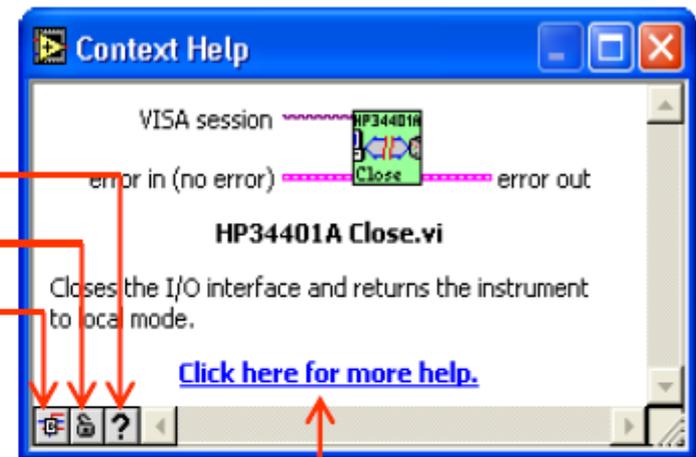
Contexto de la ayuda

Contexto de la Ayuda

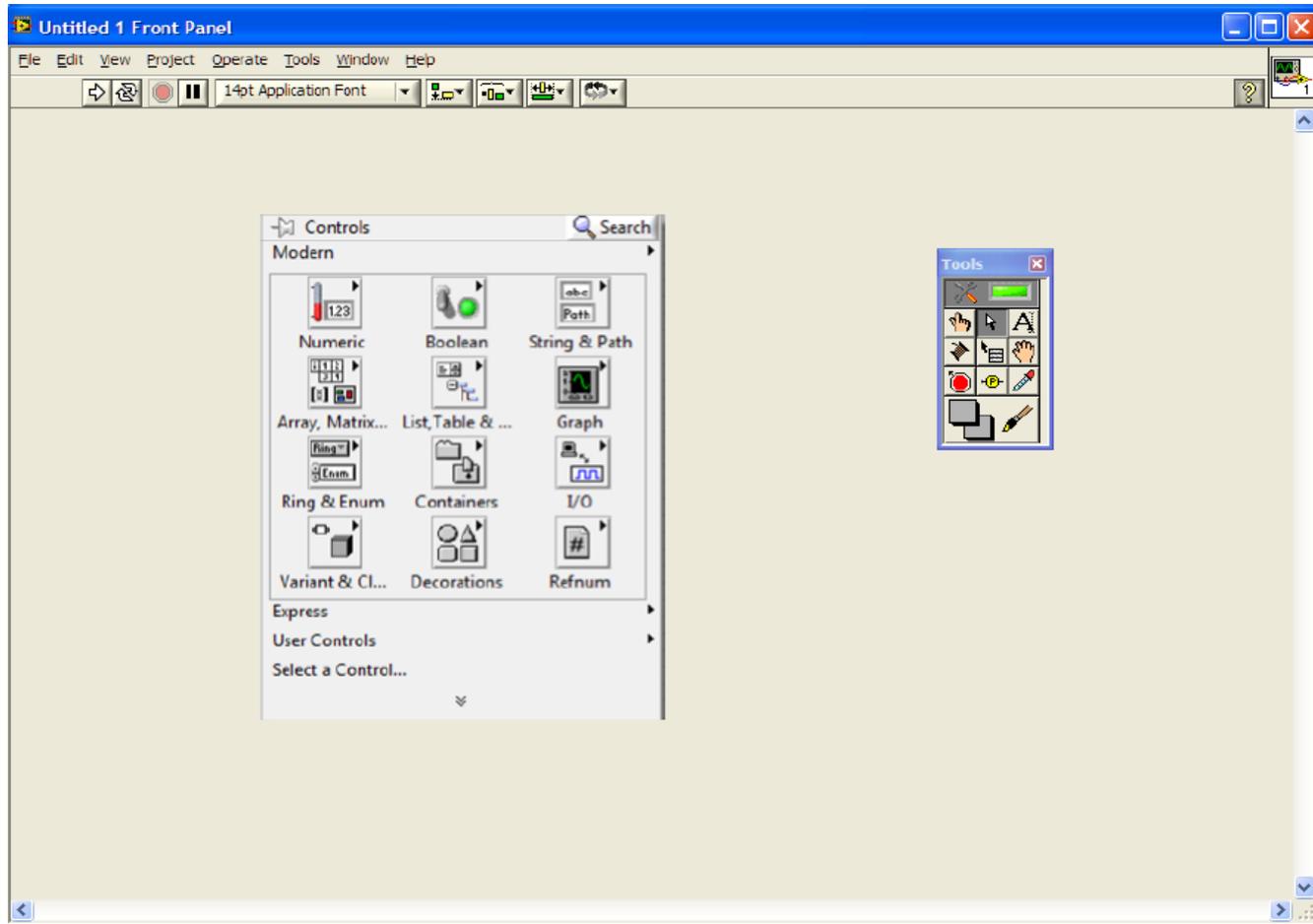
- Ayuda en línea
- Congelar Ayuda
- Path ubicación
- Ctrl + H

Referencias en Línea

•Clic en las funciones del diagrama para tener acceso directo a la información en línea.



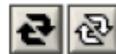
Panel frontal



Panel frontal



Botón de ejecución (Run)



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



**Cancelación de ejecución
(Abort Execution)**



Botón de pausa/continuación

13pt Application Font

**Configuración de textos
(Text Settings)**



**Alineamiento de objetos
(Align Objects)**



**Distribución de objetos
(Distribute Objects)**

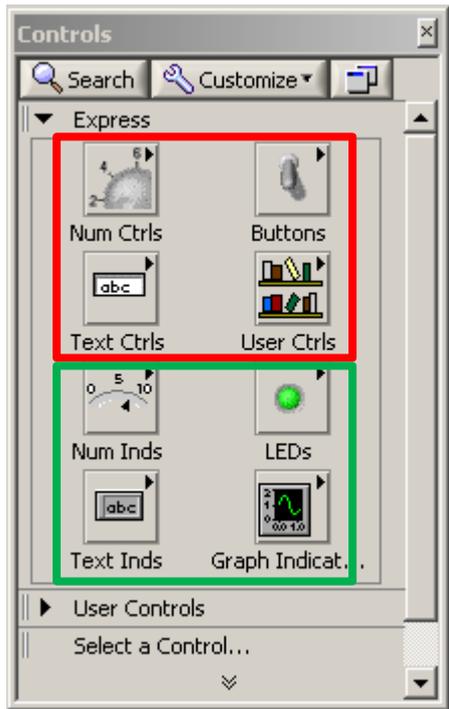


Reordenamiento



**Redimensionamiento de objetos
de panel frontal (Resize Objects)**

Panel frontal



Paleta de controles

Controles
(entradas)

Indicadores
(salidas)

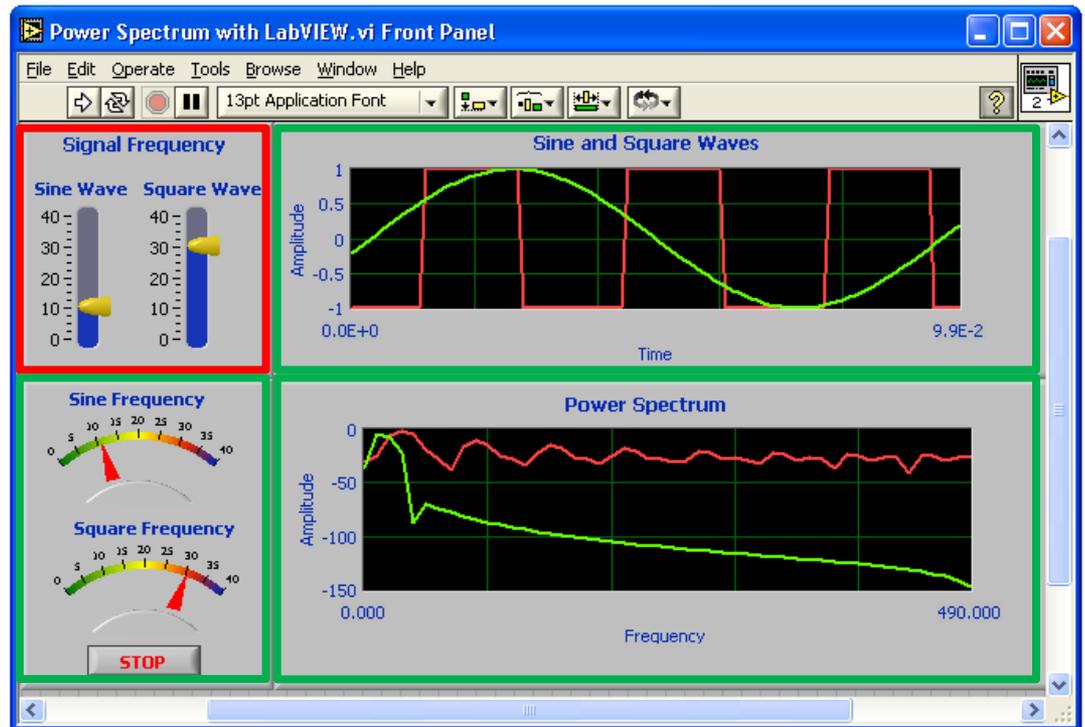


Diagrama de bloques

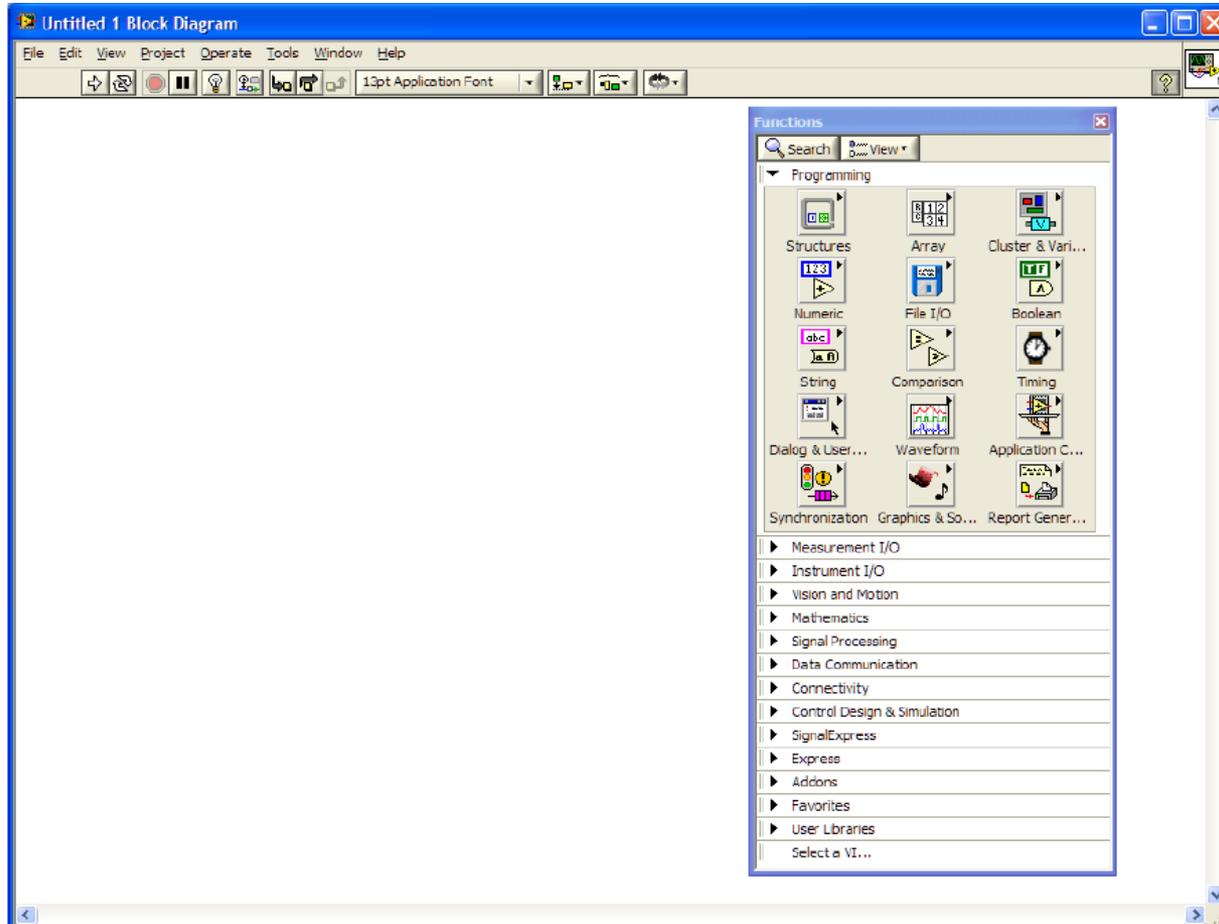


Diagrama de bloques

-  **Botón de ejecución resaltada (Highlight Execution)**
-  **Botón de entrada al ciclo (Step Into)**
-  **Botón sobre (Step Over)**
-  **Botón de salida del ciclo (Step Out)**

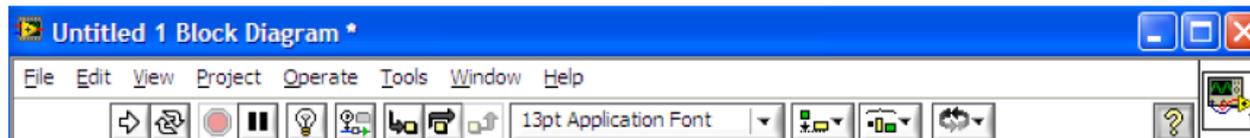
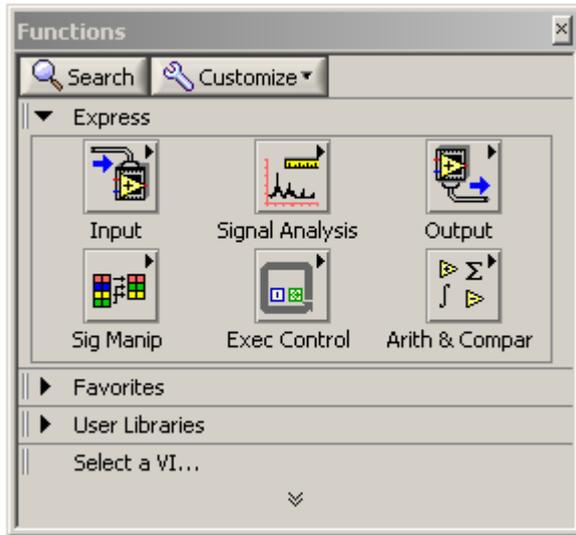
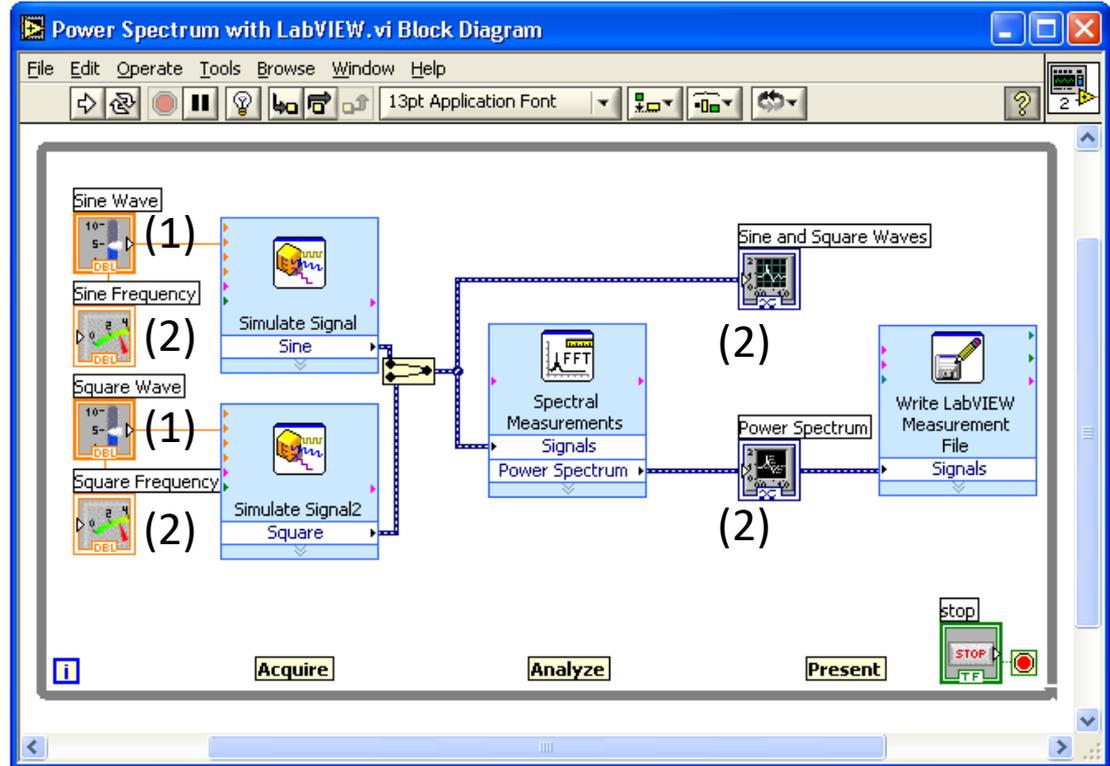


Diagrama de bloques



Paleta de funciones

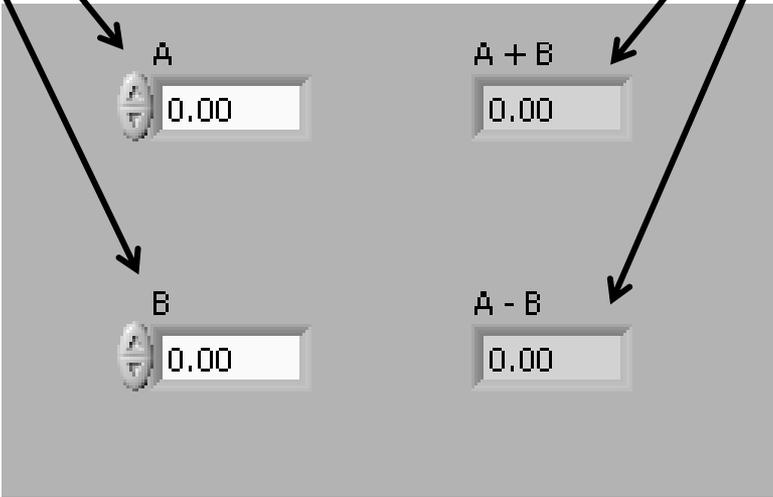


- Terminales de controles (1) e indicadores (2) del panel frontal.
- Funciones, subVIs, constantes y estructuras de programación representadas por **bloques** unidos mediante **cables** siguiendo el flujo de un programa.

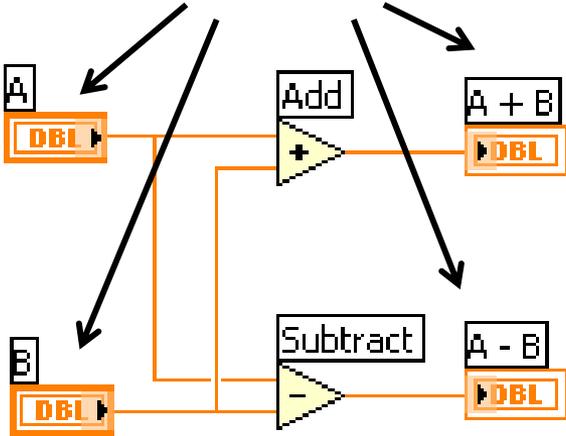
Creación de VIs

Controles

Indicadores

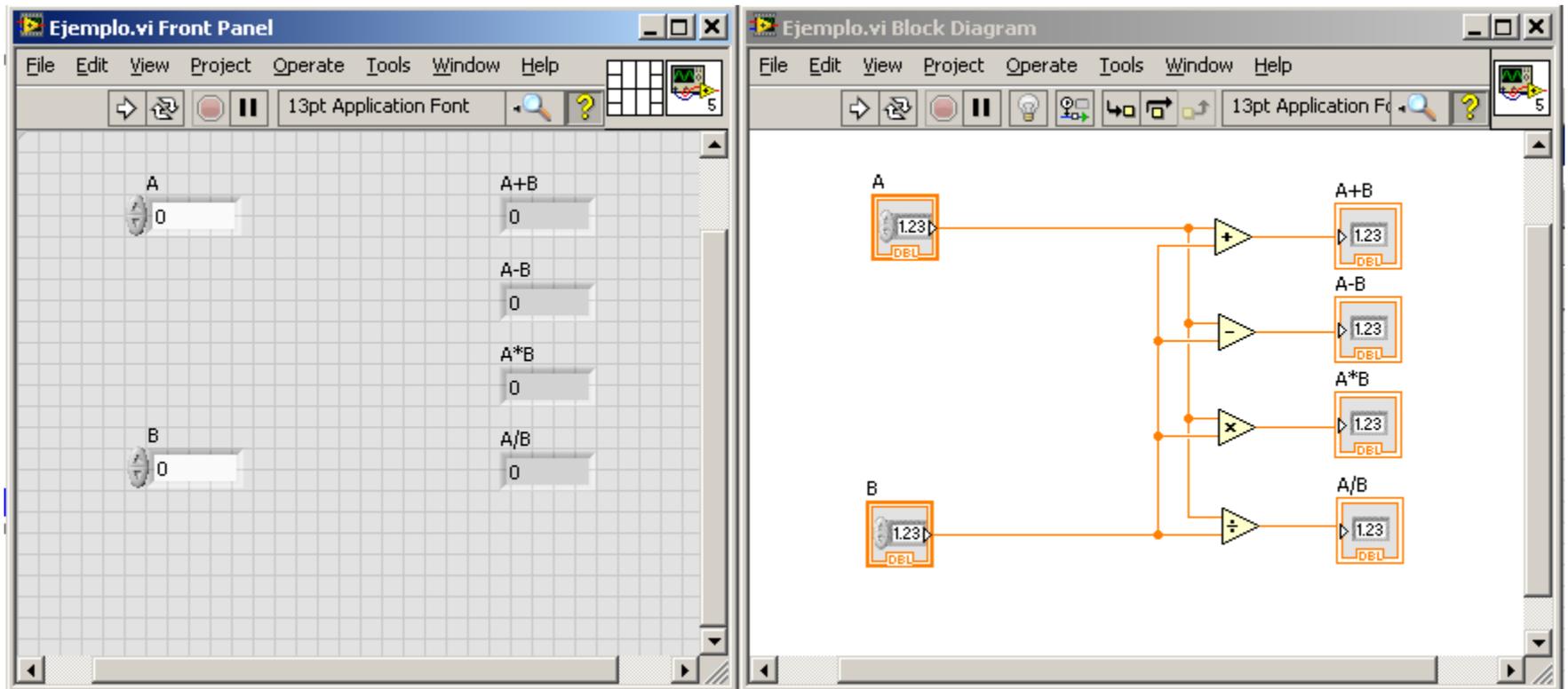


Terminales



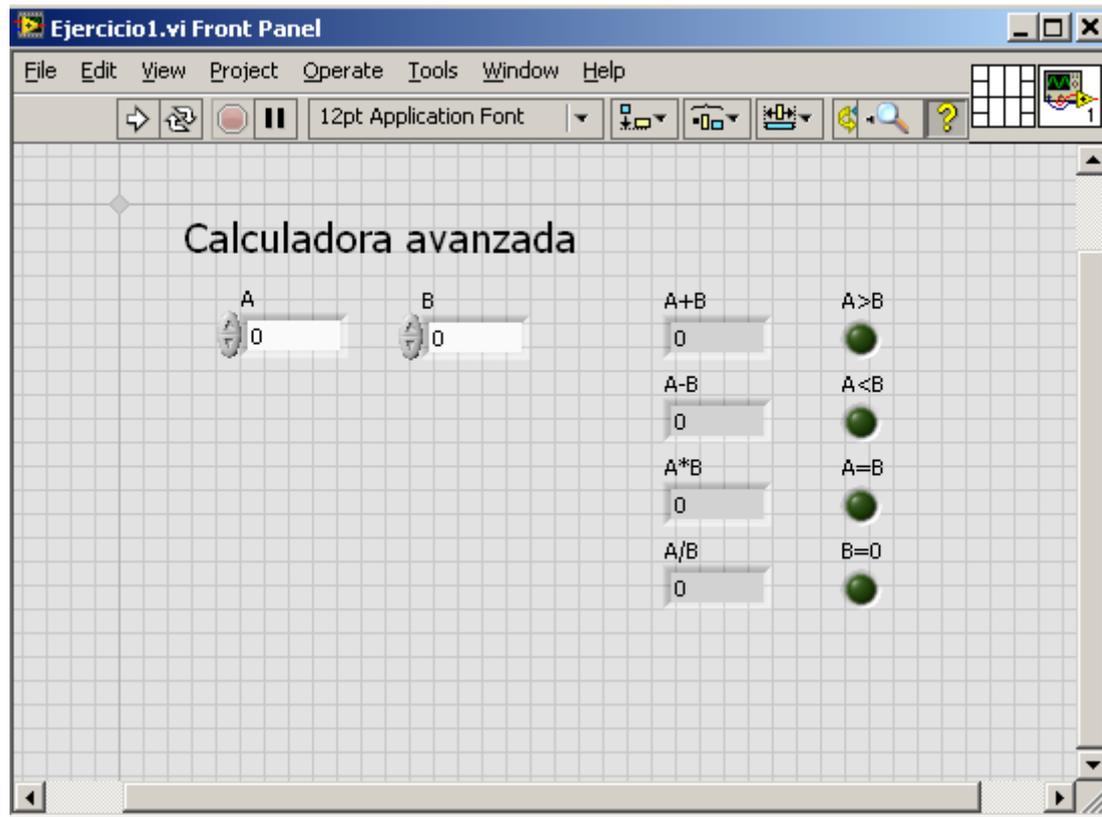
Creación de VIs

Ejemplo: Calculadora



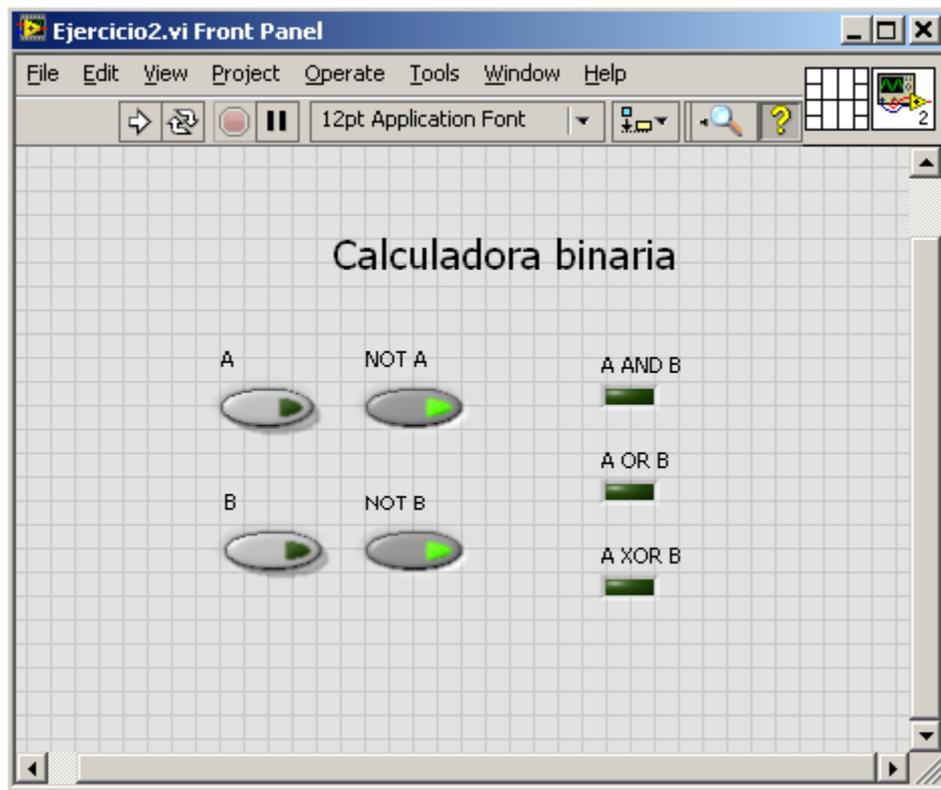
Creación de VIs

Ejercicio 1: Calculadora avanzada



Creación de VIs

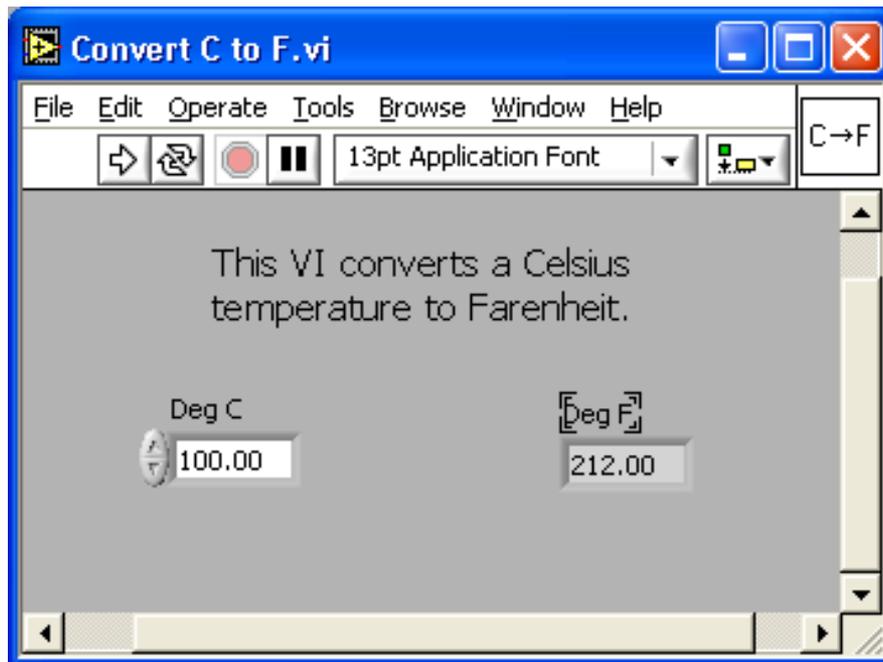
Ejercicio 2: Calculadora binaria



Nota: los botones “NOT A” y “NOT B” actúan a modo de indicador.

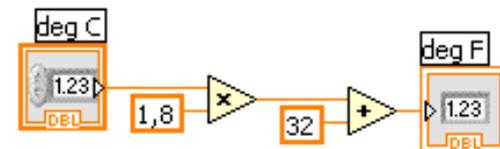
Creación de VIs

Ejemplo: Conversor °C en °F



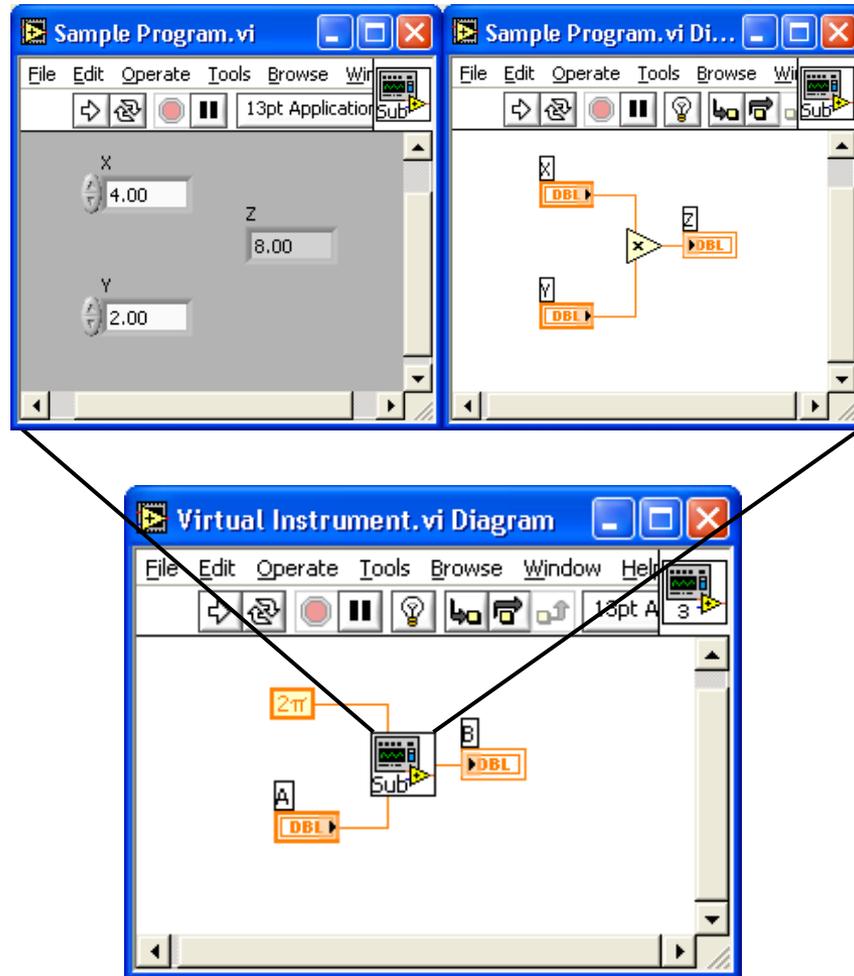
Ayuda:

$$^{\circ}F = (^{\circ}C \times 1.8) + 32$$



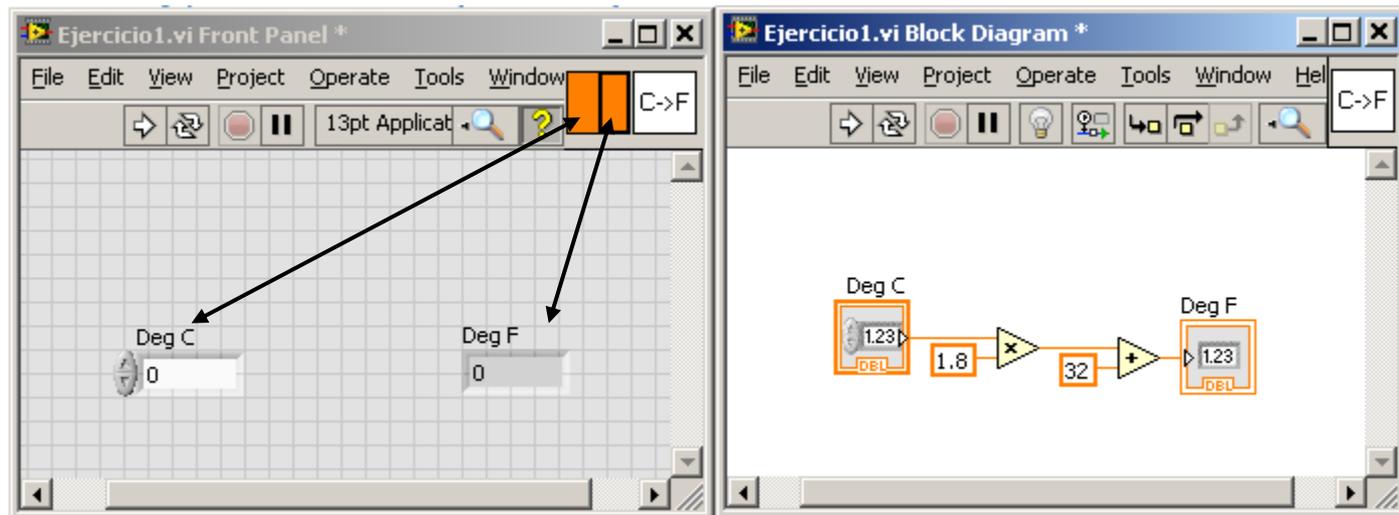
Creación de VIs

VI
↓
subVI



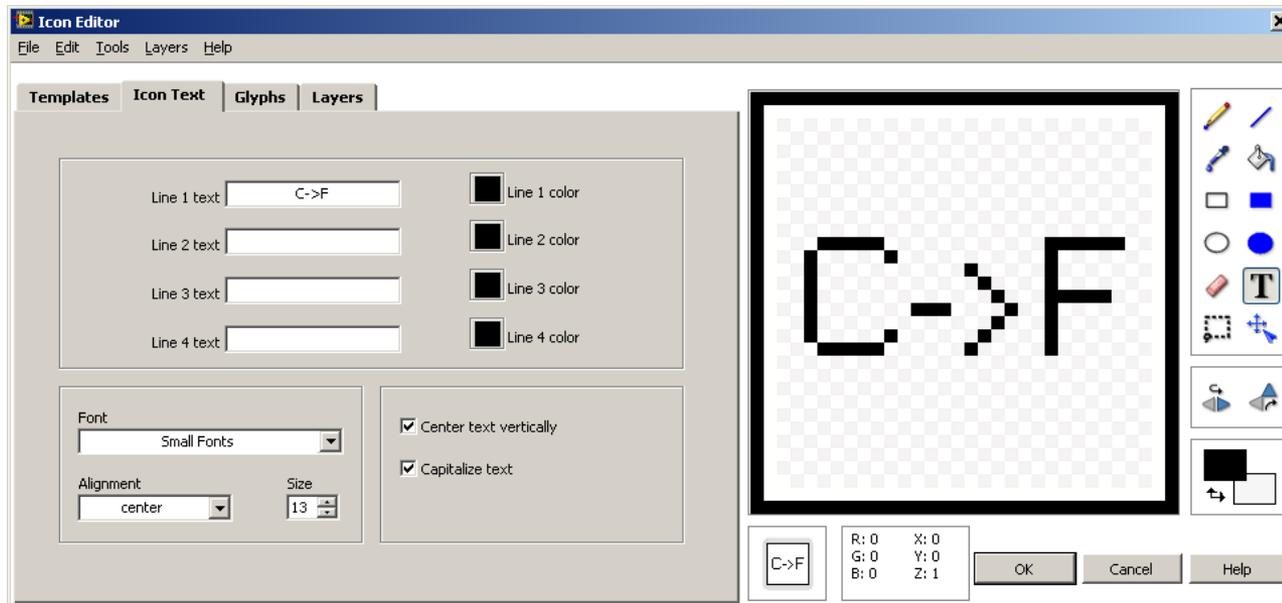
Creación de subVIs

1. Crear el instrumento virtual
2. Editar el icono
3. Asignar entradas y salidas a conectores



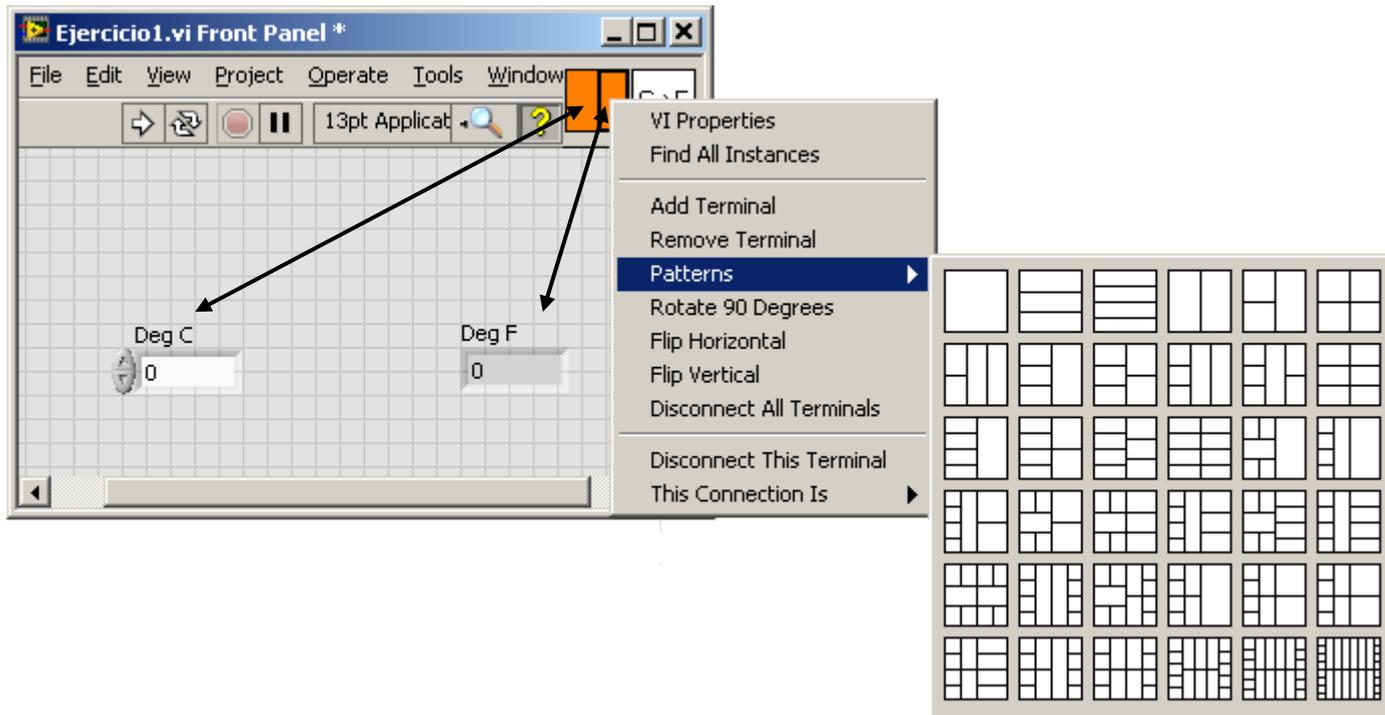
Creación de subVIs

1. Crear el instrumento virtual
2. Editar el icono
3. Asignar entradas y salidas a conectores



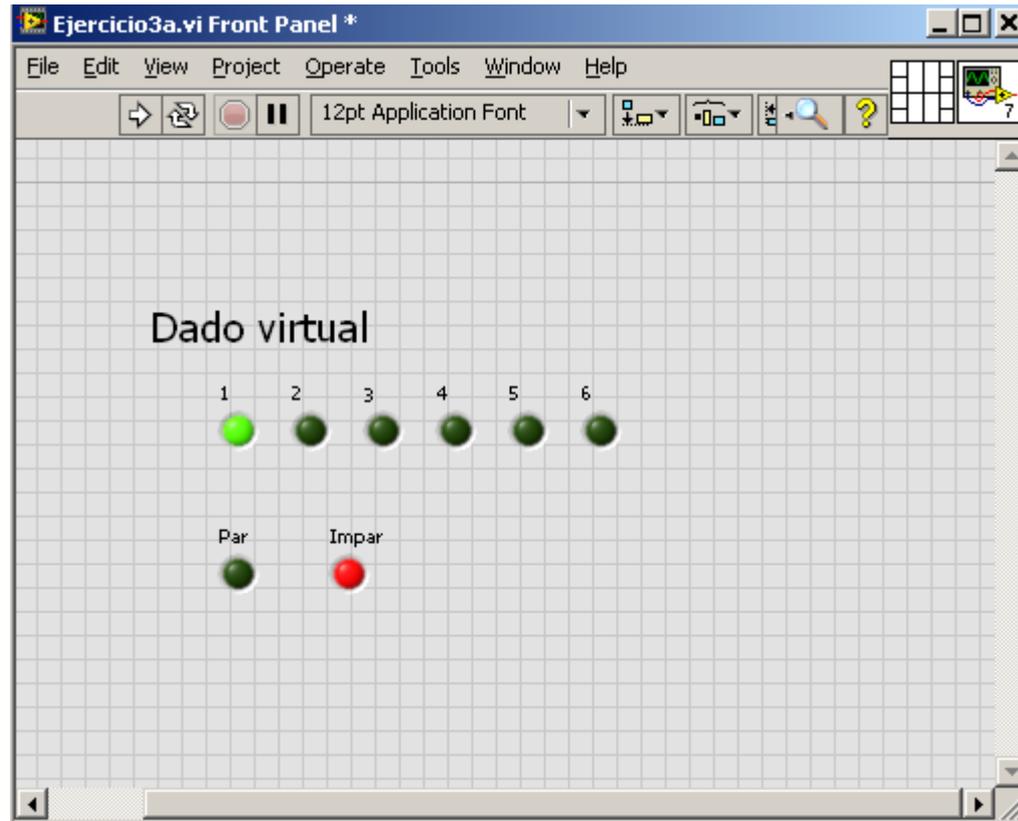
Creación de subVIs

1. Crear el instrumento virtual
2. Editar el icono
3. Asignar entradas y salidas a conectores



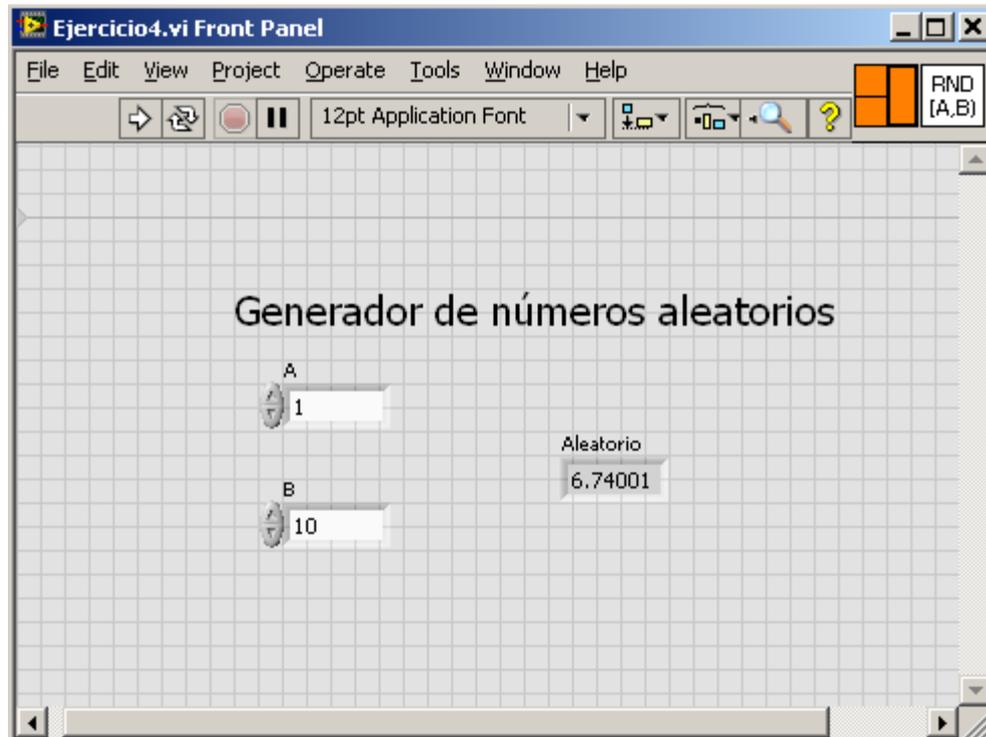
Creación de subVIs

Ejercicio 3: Dado virtual



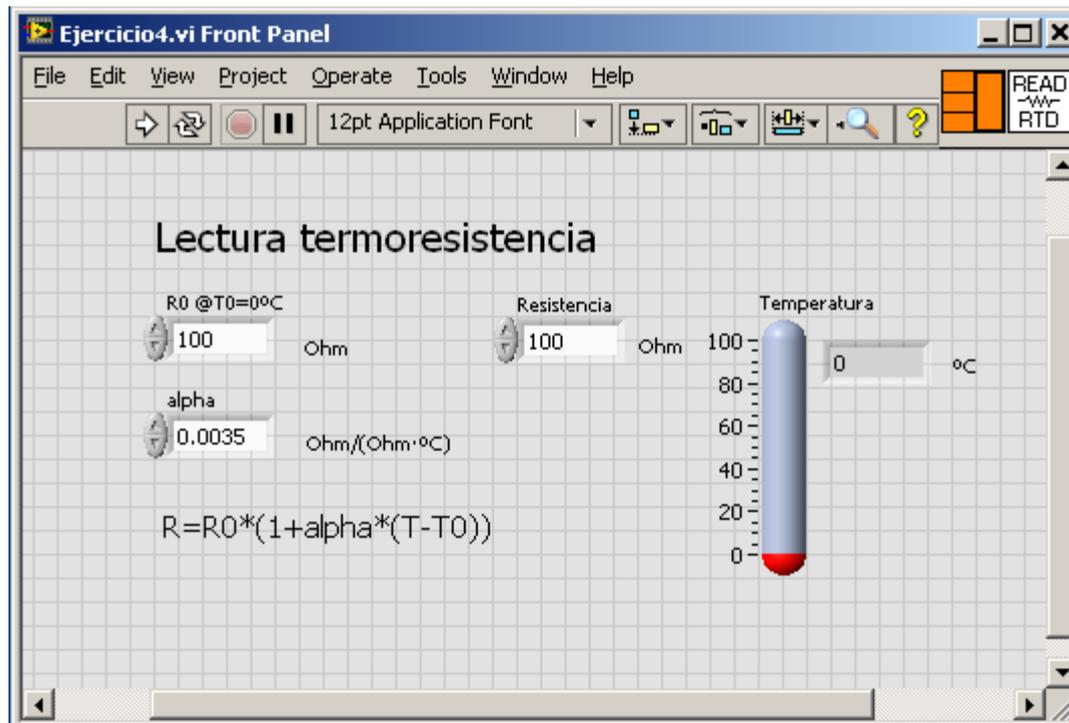
Creación de subVIs

Ejercicio 4: Generador aleatorio (subVI)



Creación de subVIs

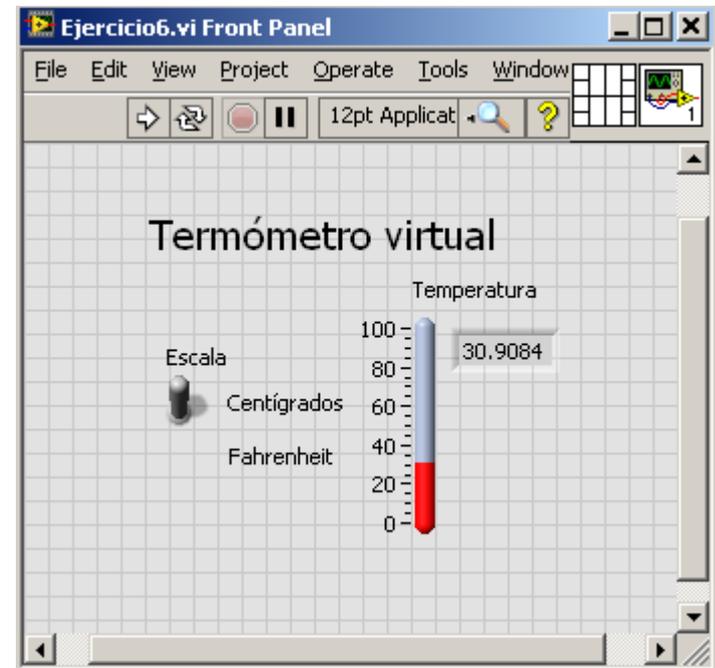
Ejercicio 5: Lectura termoresistencia (subVI)



Creación de subVIs

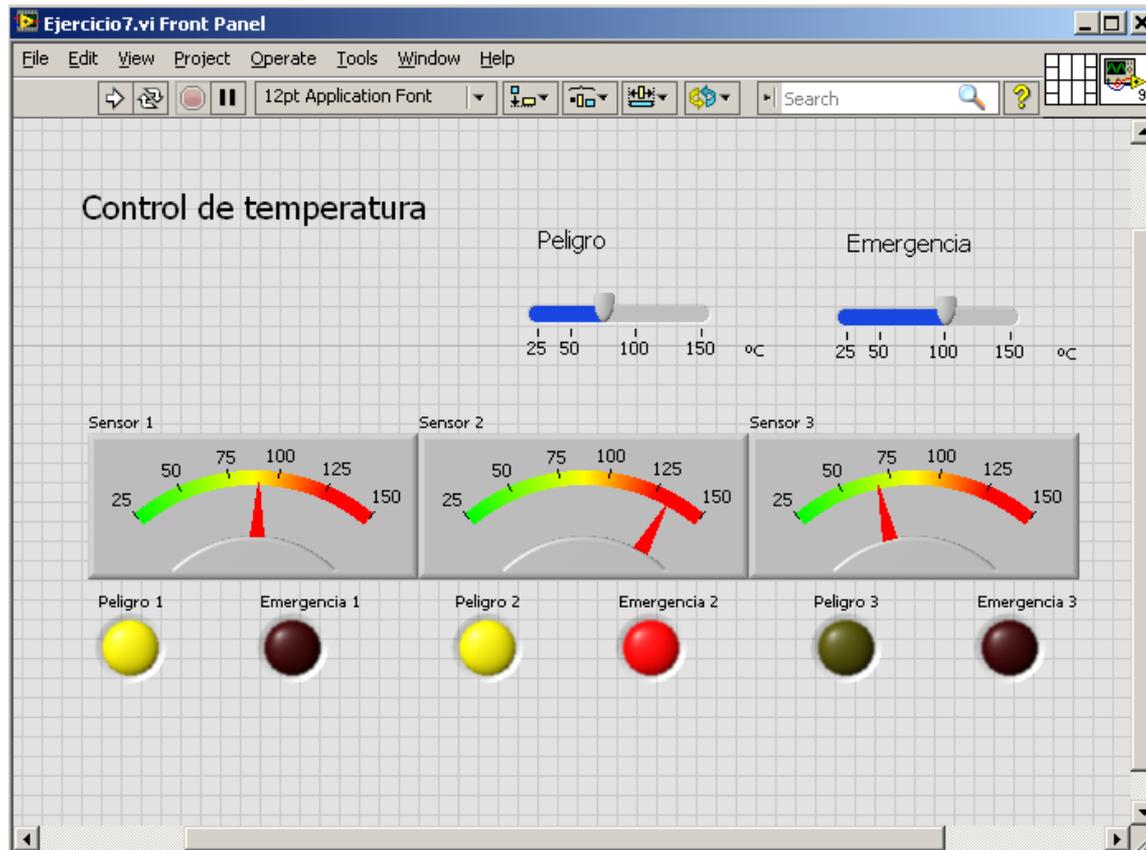
Ejercicio 6: Termómetro virtual

Crear un termómetro virtual que muestre la temperatura en grados centígrados o Fahrenheit usando un selector. El valor de temperatura vendrá dado por la lectura de una termoresistencia ($R_0=1000 \Omega$, $\alpha=0.00385 \Omega/\Omega/^\circ\text{C}$) cuyos valores se deben generar aleatoriamente en el rango $1100\pm 20\Omega$.



Creación de subVIs

Ejercicio 7: Control temperatura



Creación de subVIs

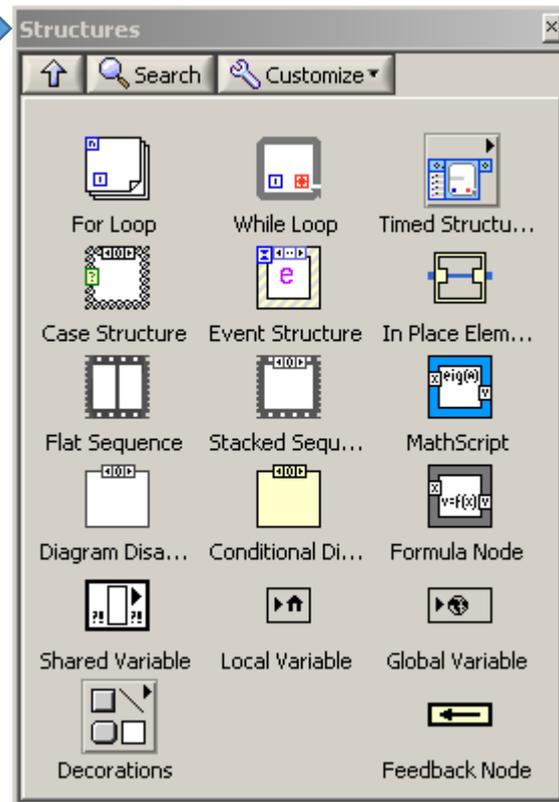
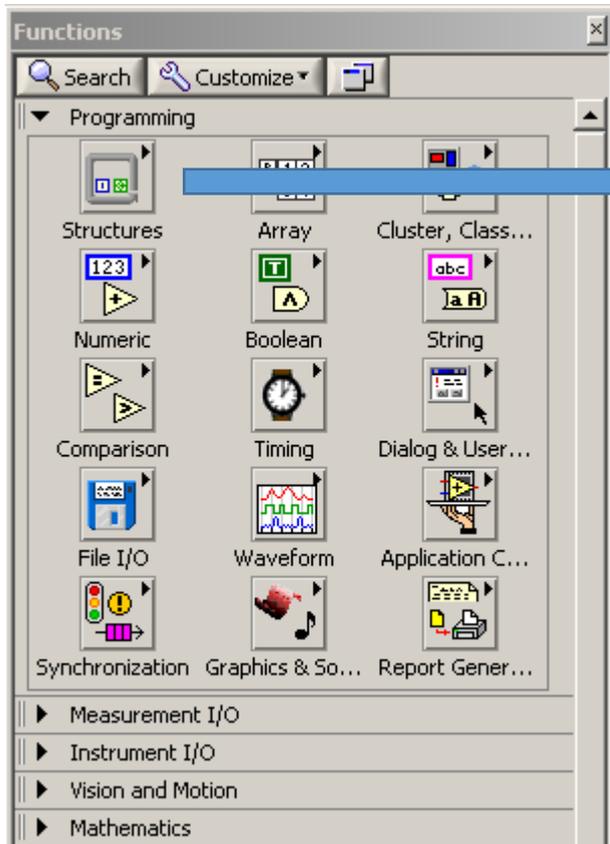
Ejercicio 7: Control temperatura

Los valores de temperatura se leen desde 3 termoresistencias PT1000 ($R_0=1000 \Omega$, $\alpha=0.00385 \Omega/\Omega/^\circ\text{C}$), con valores de resistencia generados aleatoriamente en el rango $1300\pm 200\Omega$.

Cuando el valor de temperatura de cualquier sensor sobrepase el umbral correspondiente, se deberá encender el led apropiado.

LABVIEW

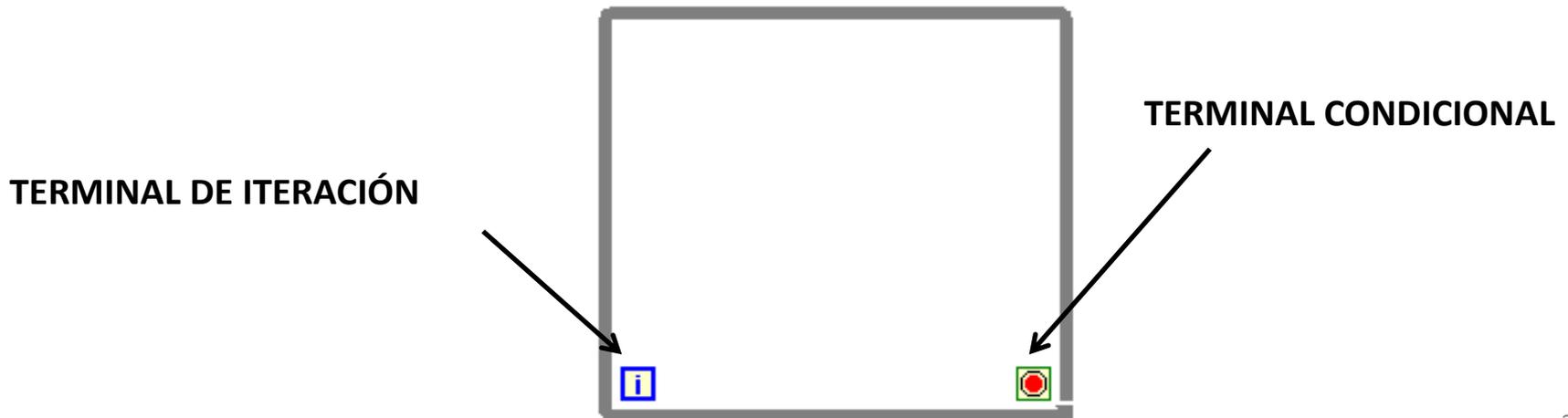
PROGRAMACION ESTRUCTURADA



While Loop

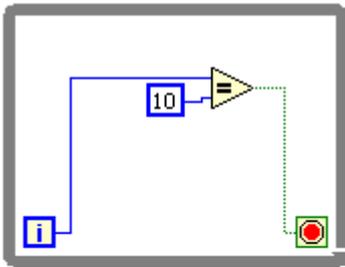
Ejecuta el contenido del sub-diagrama mientras se cumpla la condición

- La condición se evalúa al final de cada ciclo
- Siempre se ejecuta al menos una vez
- El terminal de iteración contienen el número de veces que se ha ejecutado el bucle

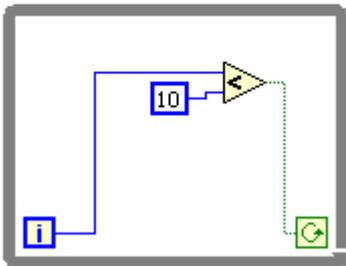


While Loop

Criterio de parada



Ejecuta el bucle **HASTA** que se cumpla la condición



Ejecuta el bucle **MIENTRAS** que se cumpla la condición

Haciendo clic en el terminal condicional se puede pasar de un tipo de criterio a otro.

While Loop

Temporalización de la ejecución

Wait (ms)



Agrega el tiempo de espera al tiempo de ejecución para alcanzar el retardo programado

Wait Until Next ms Multiple

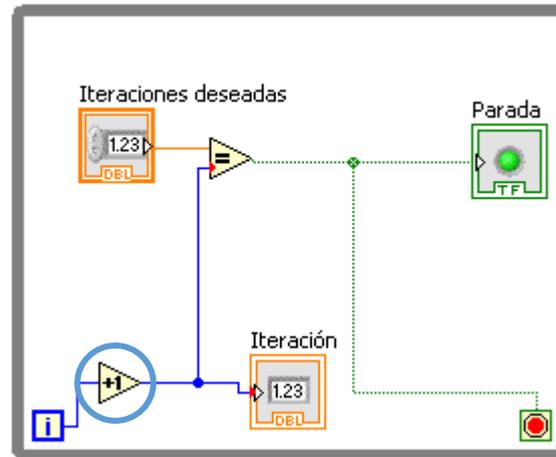


Monitoriza un contador de ms y espera hasta que el contador alcanza un múltiplo de la cantidad que hemos especificado. Esta función espera hasta que el reloj interno del ordenador se encuentra en el múltiplo especificado

While Loop

Ejemplo: Contador básico

Construir un VI que cuente la cantidad de iteraciones para alcanzar la igualdad con un número introducido por un control.

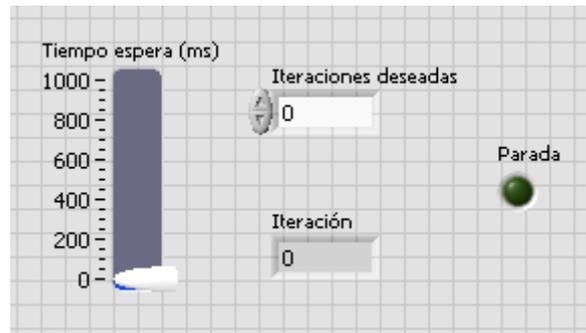


Se pone "incremento 1" para contar el nº real de iteraciones, dado que se inician en cero

While Loop

Ejercicio 1: Contador con espera

Añadir un control al contador anterior que permita elegir cuántos milisegundos ha de esperar el bucle entre ciclo y ciclo.



Ayuda: Programming > Timming > Wait (ms)

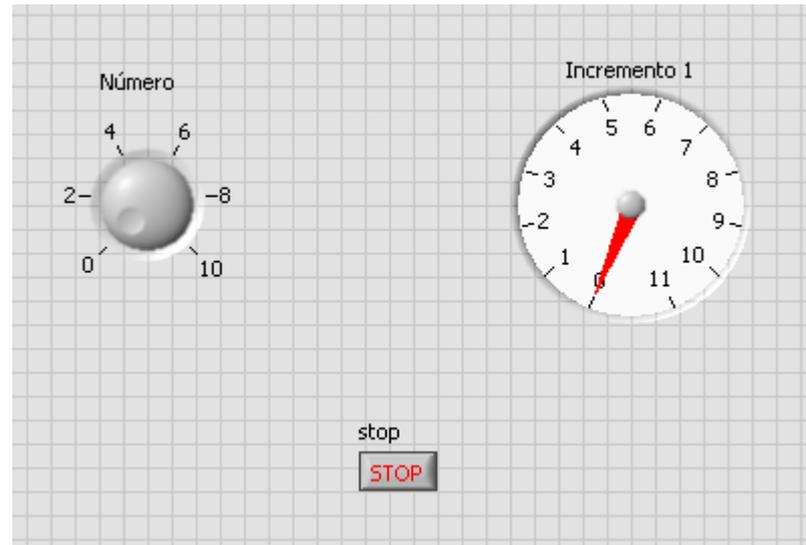


While Loop

Ejercicio 2: Incremento 1 continuo

Deseamos introducir un valor numérico y obtener a la salida ese mismo número incrementado en uno. Esta operación deseamos se realice de modo continuo hasta que pulsemos el botón STOP.

Nota: Prohibido el uso de la ejecución continua



For Loop

Ejecuta el contenido del sub-diagrama un número determinado de veces

Terminal contador

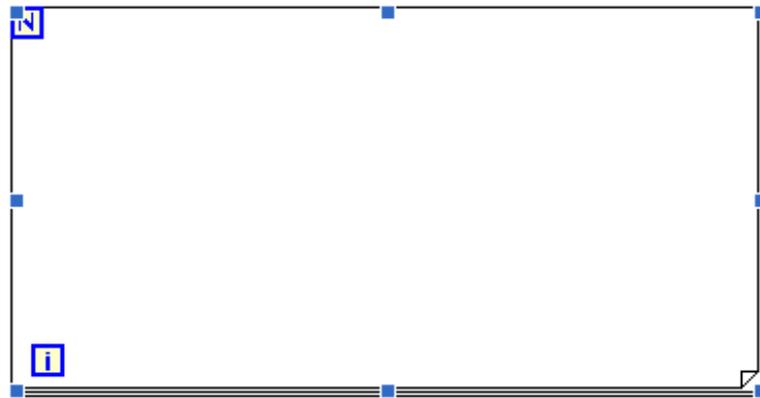


Contiene el número de veces que se ejecuta el subdiagrama. El valor del contador se fijará externamente

Terminal de iteración



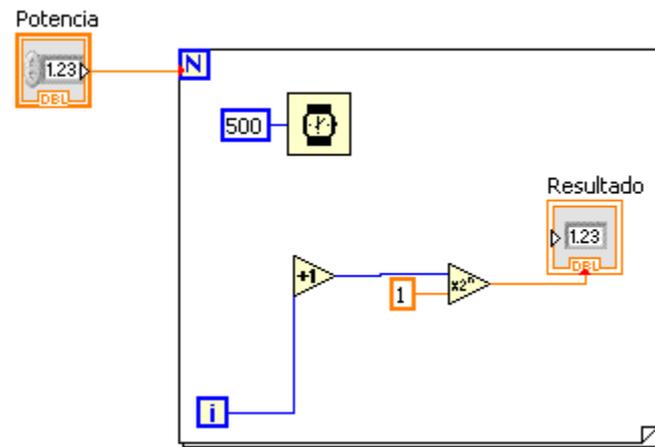
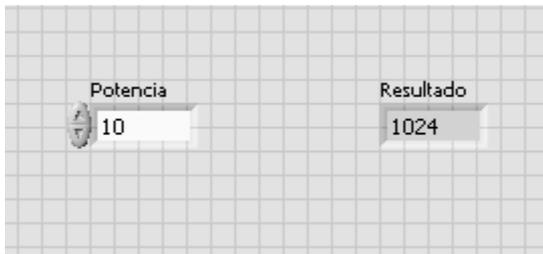
Indica el número de veces que se ha ejecutado la estructura, cero durante la primera iteración hasta N-1



For Loop

Ejemplo: Potencia de 2

Realizar un VI que calcule la potencia n-ésima (dada por un control numérico) de 2 usando un bucle *for*. Introducir una espera de 500 ms en cada ciclo.



For Loop

Ejercicio 3: Llenado de un tanque

Simular el llenado de un tanque de 100 unidades con un bucle *for*. La velocidad de llenado (tiempo de espera) se podrá elegir con un control, entre 1 y 10 unidades cada 100 ms.

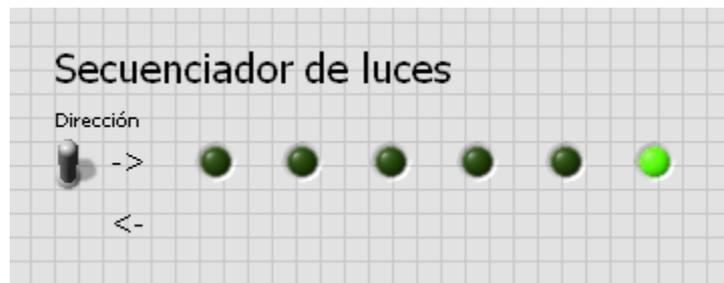
Permitir reajustar la velocidad durante el llenado.



For Loop

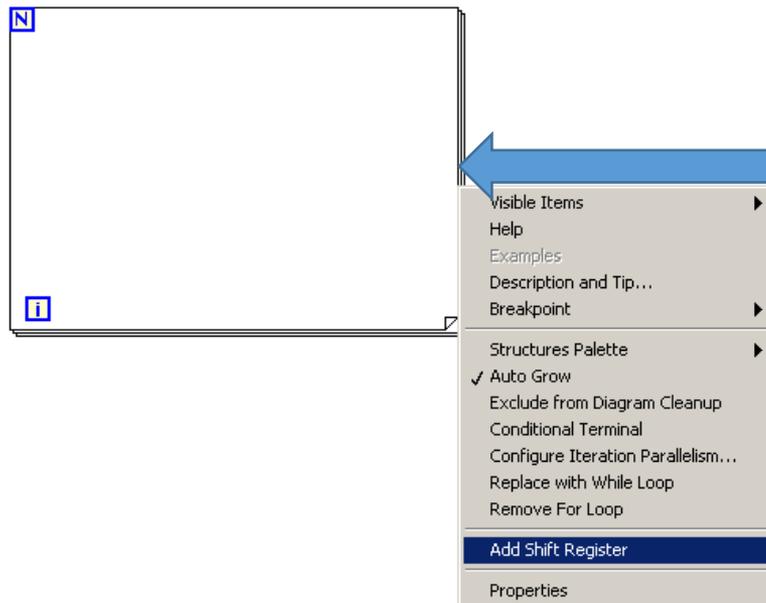
Ejercicio 4: Secuenciador de luces

Simular el encendido secuencial de 6 luces led con un control que permita elegir la dirección. Introducir un retardo de 200 ms.



Shift Register

Los registros de desplazamiento son variables locales, disponibles tanto para el *While Loop* como para el *For Loop*, que permiten transferir los valores del final de la iteración al principio de la siguiente.



Clic derecho en el borde de un *While Loop* o *For Loop*

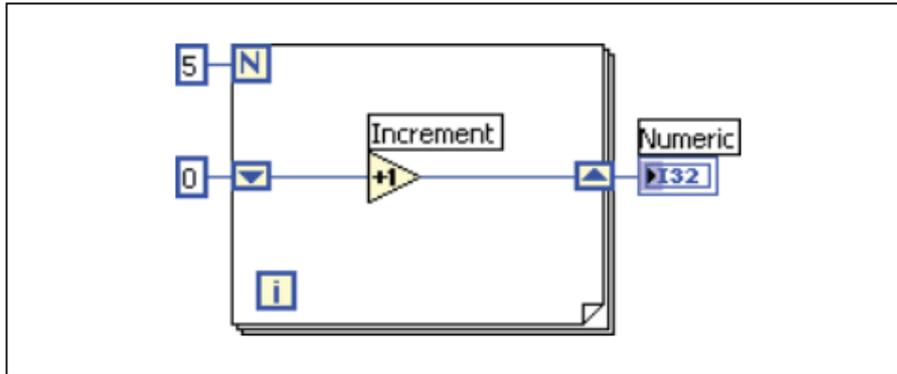
Seleccionar “Add Shift Register”

Shift Register

- Los datos a cablear en los terminales de cada registro de desplazamiento deben ser del mismo tipo.
- Un registro de desplazamiento puede tener más de un terminal en el lado izquierdo. Cuántos más elementos tenga en el lado izquierdo, más valores de iteraciones anteriores podemos almacenar.

Shift Register

Ejemplo: Registro de desplazamiento iniciado

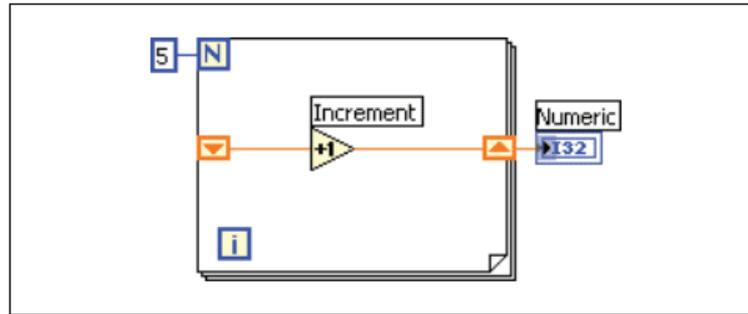


```
acumulador=0;
for i=1:5
    acumulador=acumulador+1;
end
disp(acumulador)
```

- El *For Loop* se ejecuta cinco veces, incrementando el valor del registro de desplazamiento en 1 por ciclo. Después de cinco iteraciones del *For Loop*, el registro de desplazamiento pasa el valor final, 5, al indicador y el VI se detiene.
- Cada vez que ejecuta el VI, el registro de desplazamiento comienza con un valor de 0. Si no se inicia el registro de desplazamiento, el ciclo utiliza el valor escrito en el registro de desplazamiento cuando el último ciclo es ejecutado o el valor predeterminado para el tipo de datos si el ciclo nunca ha sido ejecutado.
- Use un registro de desplazamiento no iniciado para conservar la información del estado entre ejecuciones subsecuentes de un VI

Shift Register

Ejemplo: Registro de desplazamiento no iniciado

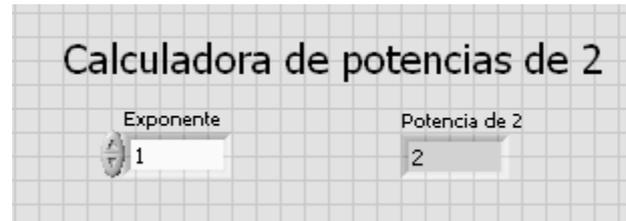


- El *For Loop* se ejecuta cinco veces, incrementando el valor del registro de desplazamiento en 1 por ciclo.
- La primera vez que se ejecuta el VI, el registro de desplazamiento comienza con un valor de 0 (predeterminado para un entero de 32 bits)
- Después de cinco iteraciones, el registro de desplazamiento pasa el valor final, 5, al indicador y el VI se detiene.
- La próxima vez que se ejecute el VI, el registro de desplazamiento comienza con un valor de 5, el último valor de la ejecución anterior.
- Después de cinco iteraciones, el registro de desplazamiento pasa el valor final, 10, al indicador y el VI se detiene.
- Si se ejecuta el VI otra vez, el registro de desplazamiento comienza con un valor de 10 y así sucesivamente.
- Los registros de desplazamiento no iniciados retienen el valor de la iteración anterior hasta que se cierra el VI.

Shift Register

Ejercicio 5: Potencia de 2

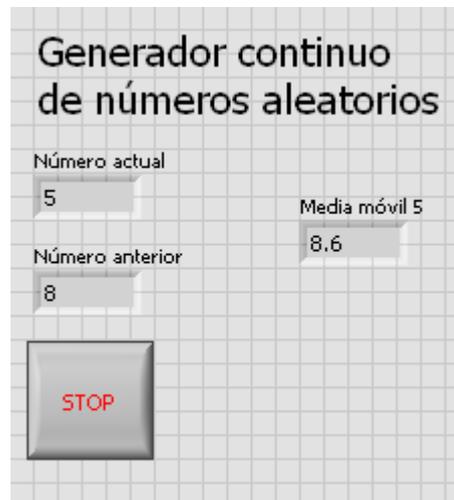
Crear un instrumento virtual que calcule la potencia de 2 de un número entero.



Shift Register

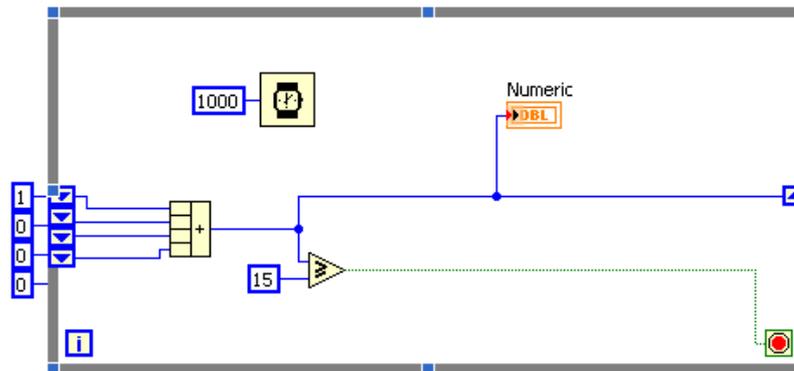
Ejercicio 6: Número aleatorio continuo

Crear un instrumento virtual que muestre cada segundo un número entero aleatorio entre 0 y 10. Añadir un registro de desplazamiento para visualizar el número aleatorio anterior y la media de las 5 últimas iteraciones.



Shift Register

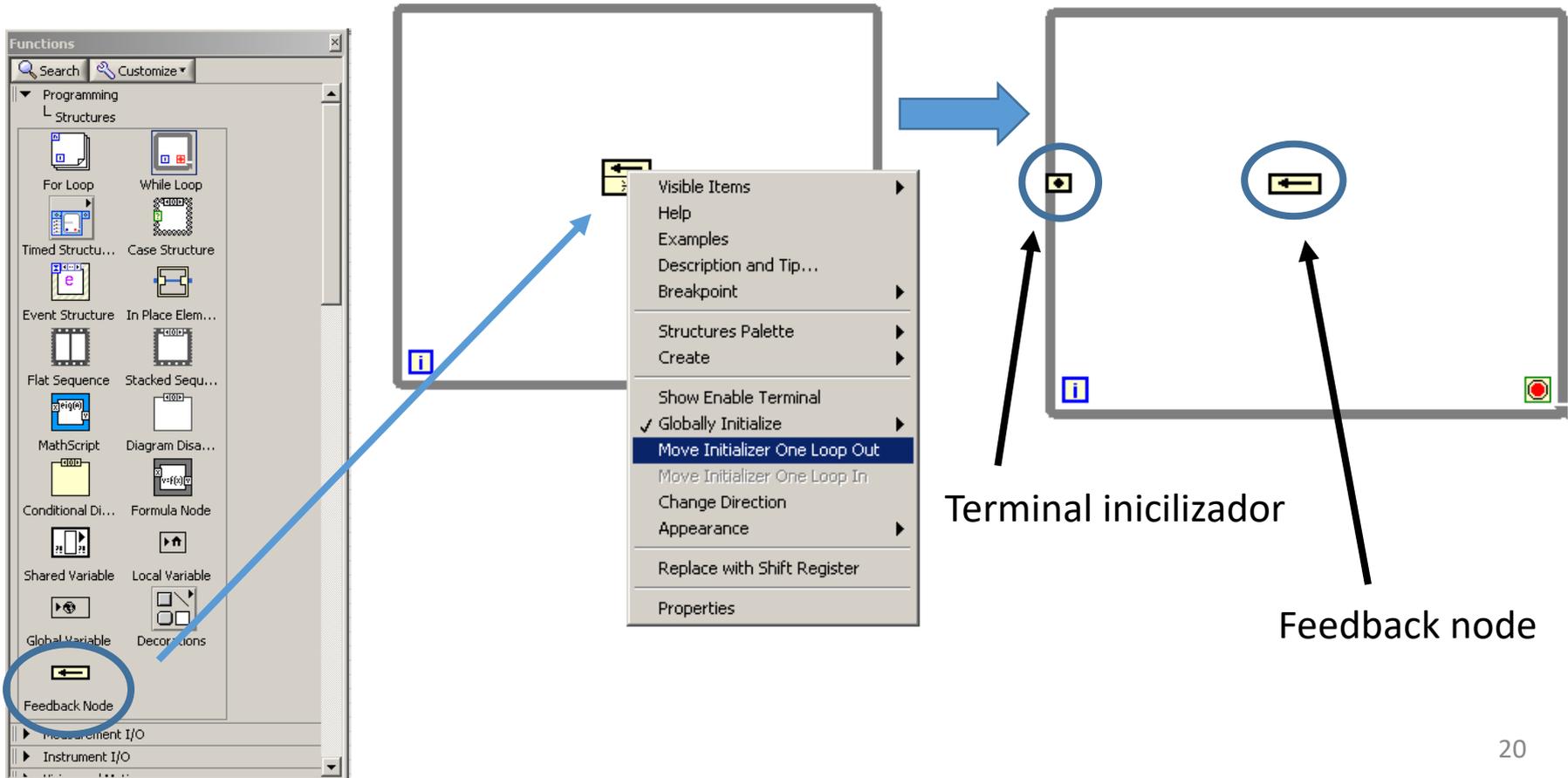
Ejercicio 7: Completar la tabla del siguiente VI



Iteración	Registro izquierdo 1	Registro izquierdo 2	Registro izquierdo 3	Registro izquierdo 4	Registro derecho
1					
2					
3					
4					
5					

Feedback Node

Permite usar información de iteraciones anteriores



Feedback Node

- **Terminal inicializador**

- Permite dar un valor inicial al nodo y equivale a conectar un valor al terminal izquierdo en un registro de desplazamiento.
- Este terminal se coloca siempre en el borde izquierdo de la estructura a la misma altura que el Feedback node.

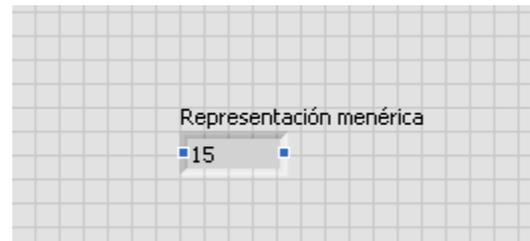
- **Feedback node**

- Es el otro terminal y tiene forma de flecha.
- En el extremo derecho se le conectará la salida, cuyo valor será leído por el extremo izquierdo en la siguiente iteración.
- El primer valor leído por el extremo izquierdo será el conectado al terminal inicializador.

Feedback Node

Ejercicio 8: Contador feedback

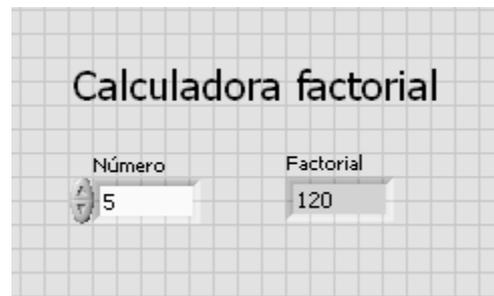
Diseñar un VI que represente en un indicador numérico los números entre 10 y 15 usando un *feedback node*.



Feedback Node

Ejercicio 9: Factorial

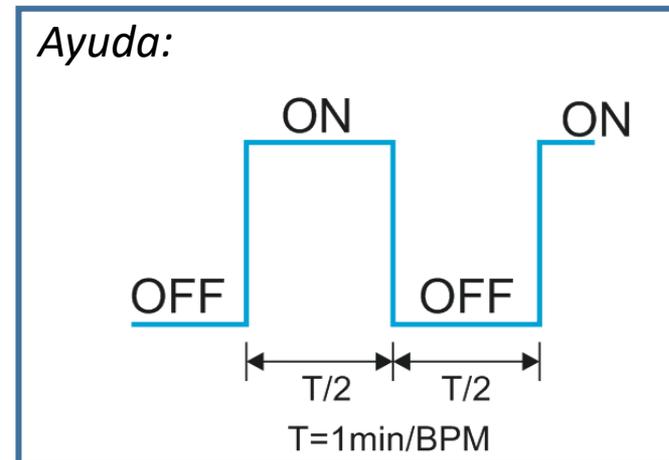
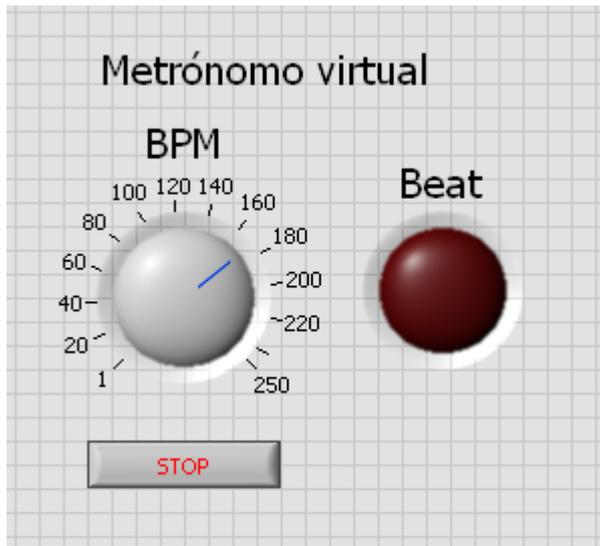
Diseñar un VI que calcule el factorial de un número entero usando un *feedback node*.



Feedback Node

Ejercicio 10: Metrónomo

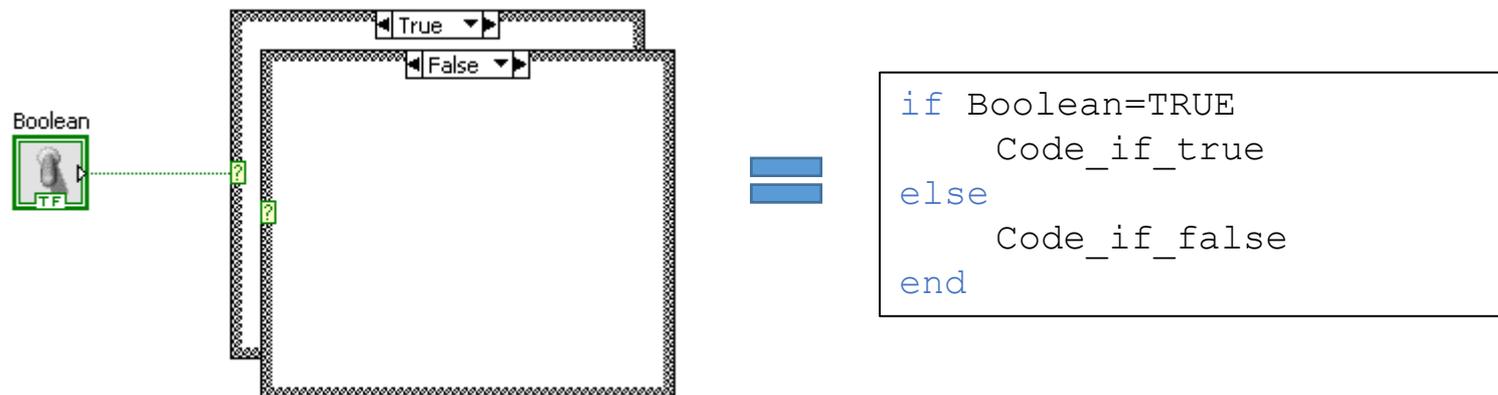
Realizar el siguiente metrónomo virtual con un indicador luminoso que se encenderá tantas veces por minuto (BPM) como se indique en el control dial.



Case

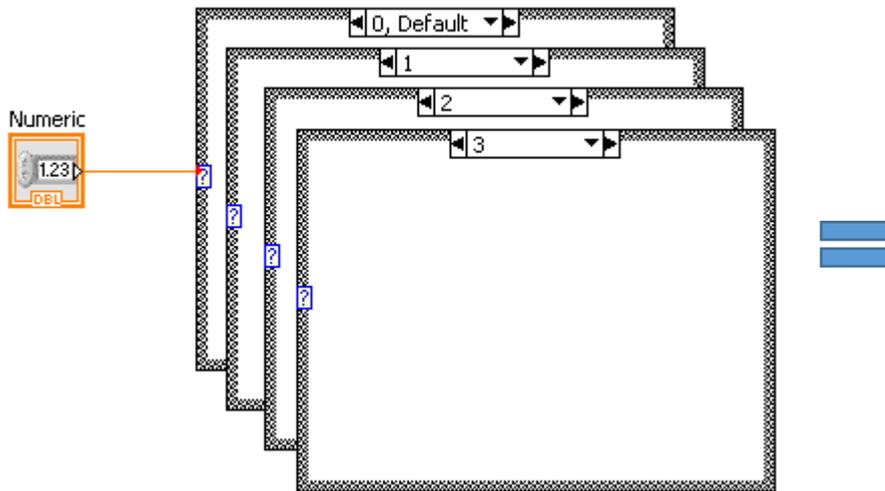
La estructura *Case* consiste en 2 o más subdiagramas, de los cuales sólo uno de ellos se ejecutará dependiendo del valor del terminal de selección.

Por defecto, la estructura *Case* tiene dos posibles opciones: *True* y *False*.



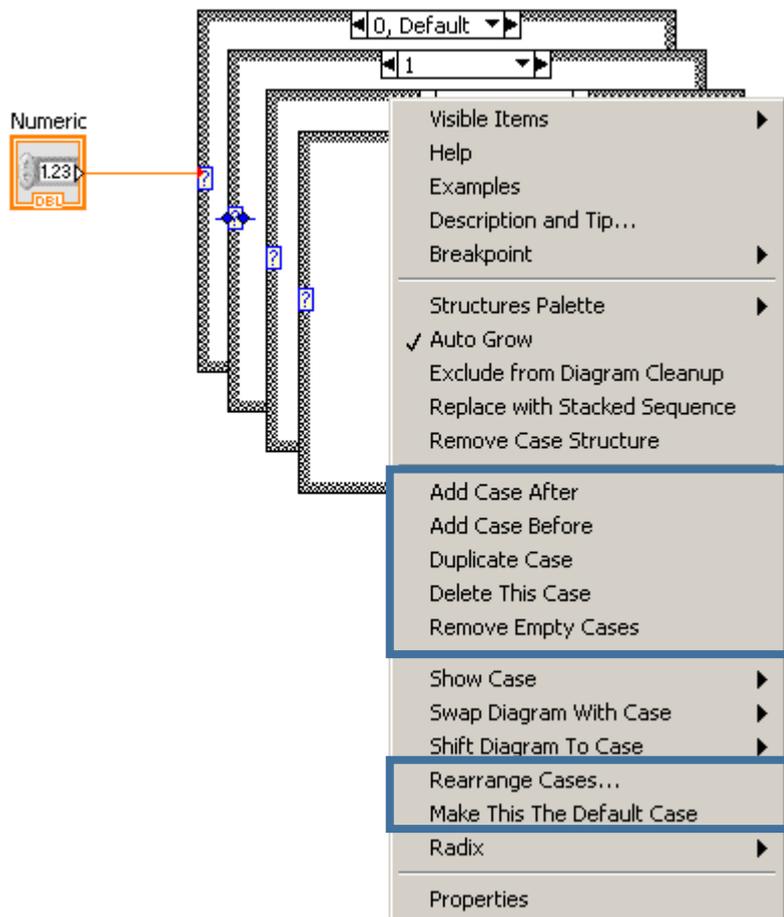
Case

En caso de colocar un control numérico en el terminal de selección, la estructura *Case* permitirá más de 2 opciones.



```
switch Numeric
  case 0
    Code_if_0
  case 1
    Code_if_1
  case 2
    Code_if_2
  case 3
    Code_if_3
  otherwise
    Default_code
end
```

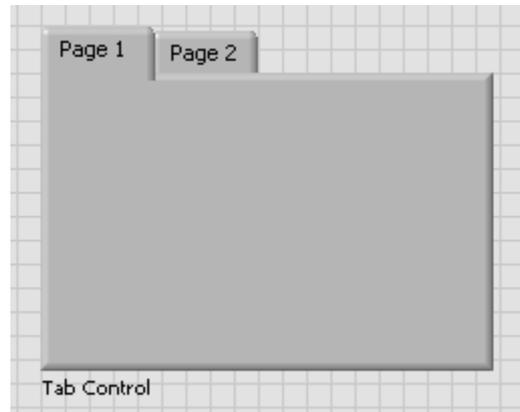
Case



En el menú emergente de la estructura *Case* es posible añadir, eliminar y reordenar los casos.

Case

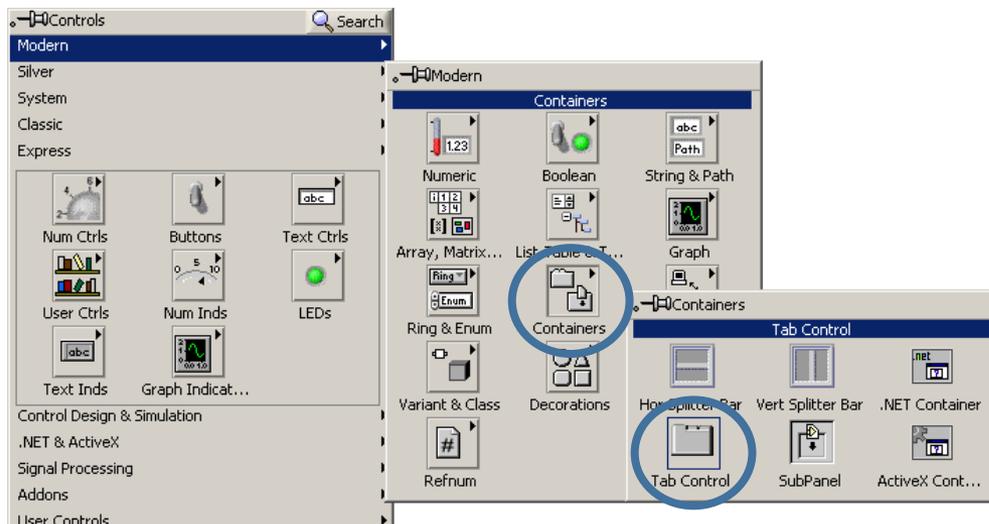
La estructura *Case* también se puede controlar con un *Tab Control*.



Case

Los pasos para usar un *Tab Control* con la estructura *Case* son los siguientes:

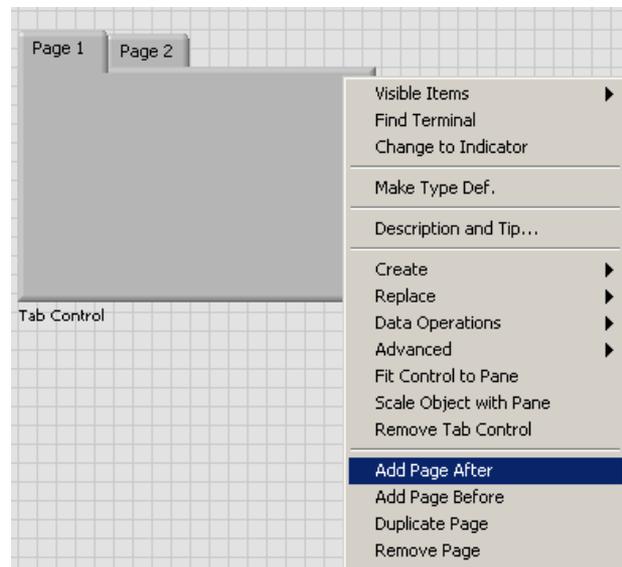
1. Crear el *Tab Control*.
2. Añadir las páginas necesarias en el control.
3. Editar las etiquetas de las páginas.
4. Conectar el *Tab Control* a la estructura *Case*.
5. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Tab Control* con la estructura *Case* son los siguientes:

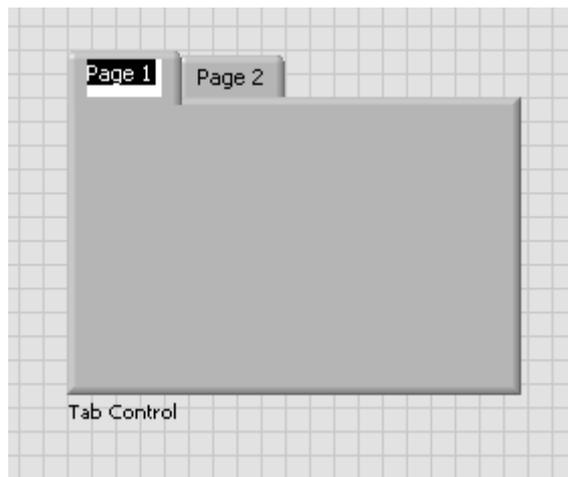
1. Crear el *Tab Control*.
2. Añadir las páginas necesarias en el control.
3. Editar las etiquetas de las páginas.
4. Conectar el *Tab Control* a la estructura *Case*.
5. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Tab Control* con la estructura *Case* son los siguientes:

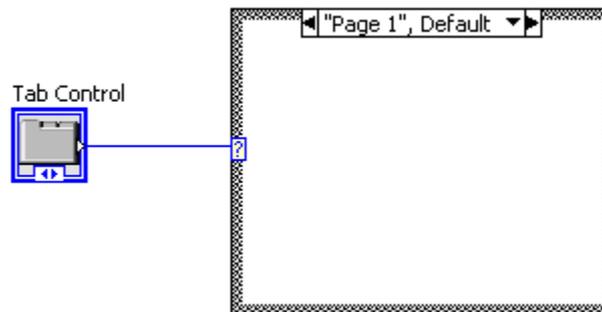
1. Crear el *Tab Control*.
2. Añadir las páginas necesarias en el control.
3. Editar las etiquetas de las páginas.
4. Conectar el *Tab Control* a la estructura *Case*.
5. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Tab Control* con la estructura *Case* son los siguientes:

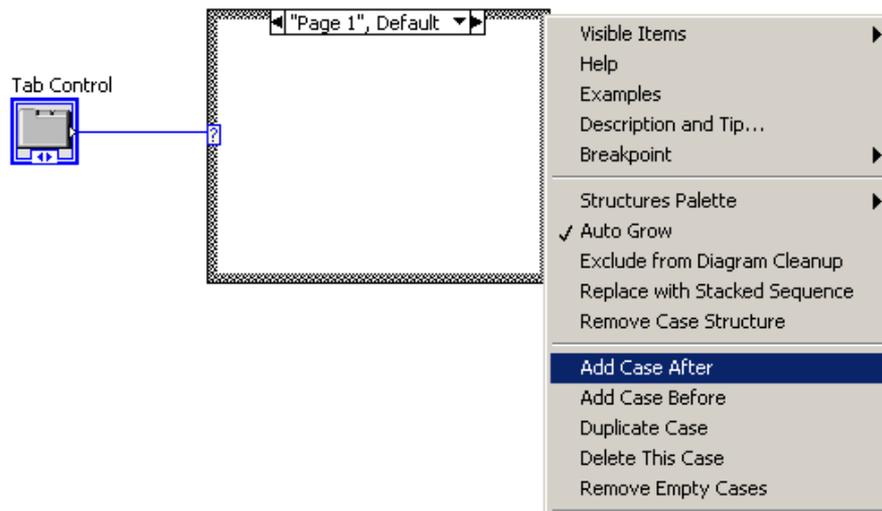
1. Crear el *Tab Control*.
2. Añadir las páginas necesarias en el control.
3. Editar las etiquetas de las páginas.
4. Conectar el *Tab Control* a la estructura *Case*.
5. Añadir los casos necesarios en la estructura *Case*.



Case

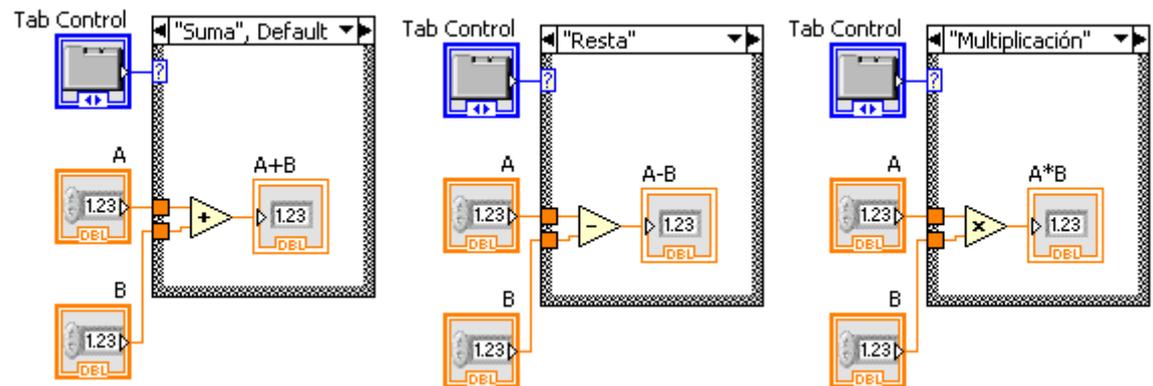
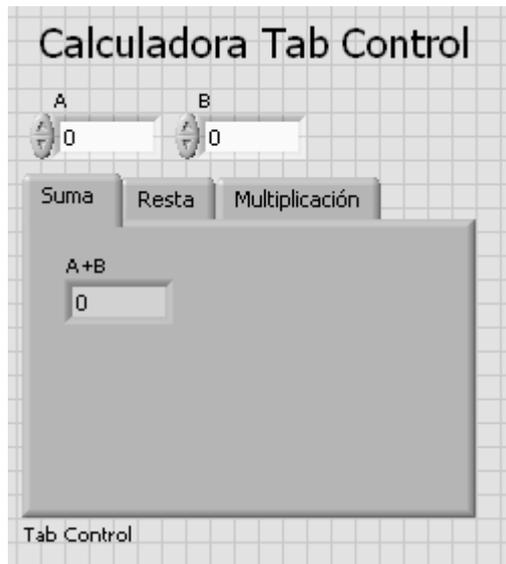
Los pasos para usar un *Tab Control* con la estructura *Case* son los siguientes:

1. Crear el *Tab Control*.
2. Añadir las páginas necesarias en el control.
3. Editar las etiquetas de las páginas.
4. Conectar el *Tab Control* a la estructura *Case*.
5. Añadir los casos necesarios en la estructura *Case*.



Case

Ejemplo: Calculadora Tab Control



Case

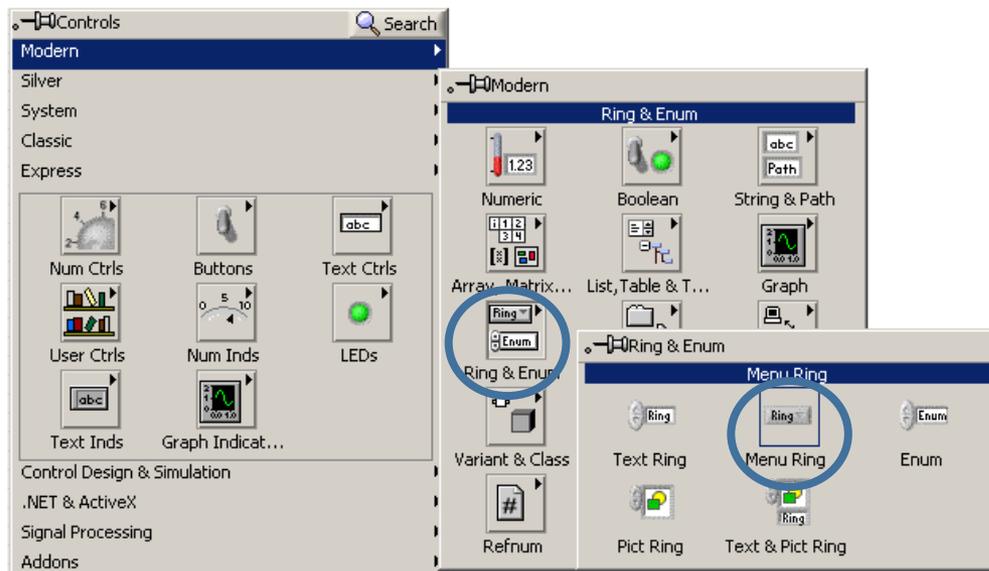
La estructura *Case* también se puede controlar con un *Ring Control*.



Case

Los pasos para usar un *Ring Control* con la estructura *Case* son los siguientes:

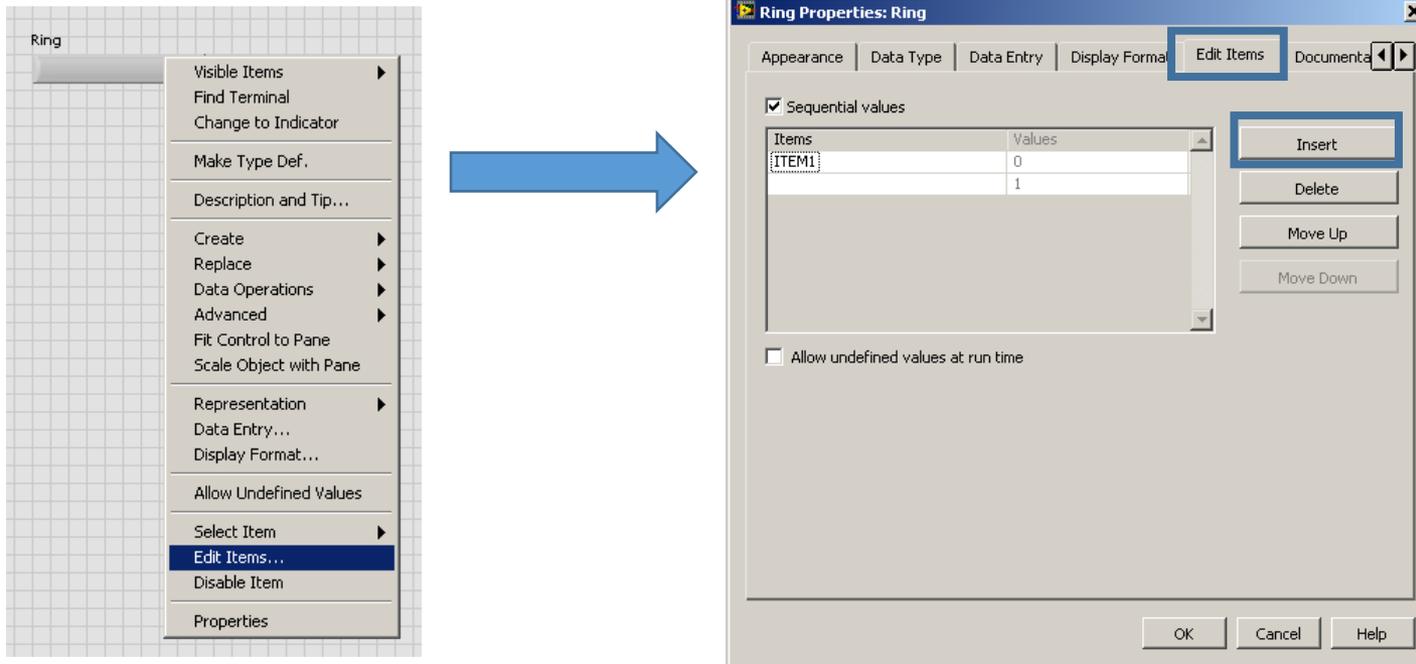
1. Crear el *Ring Control*.
2. Añadir los elementos necesarios en el control.
3. Conectar el *Ring Control* a la estructura *Case*.
4. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Ring Control* con la estructura *Case* son los siguientes:

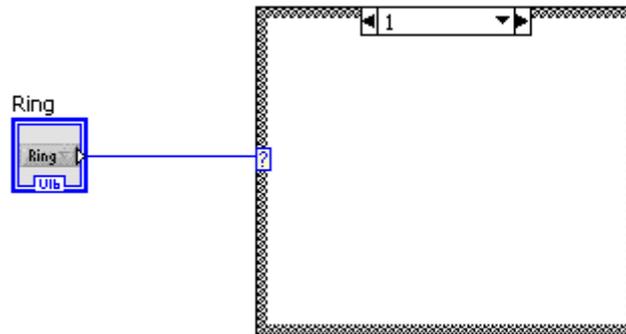
1. Crear el *Ring Control*.
2. Añadir los elementos necesarios en el control.



Case

Los pasos para usar un *Ring Control* con la estructura *Case* son los siguientes:

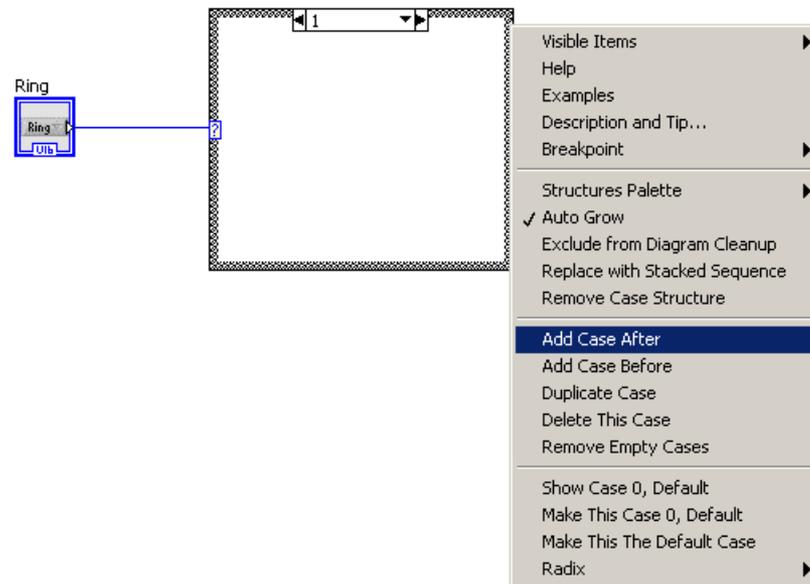
1. Crear el *Ring Control*.
2. Añadir los elementos necesarios en el control.
3. Conectar el *Ring Control* a la estructura *Case*.
4. Añadir los casos necesarios en la estructura *Case*.



Case

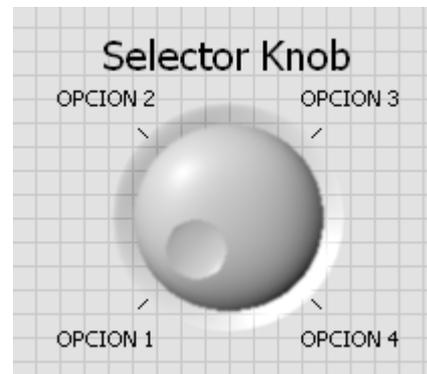
Los pasos para usar un *Ring Control* con la estructura *Case* son los siguientes:

1. Crear el *Ring Control*.
2. Añadir los elementos necesarios en el control.
3. Conectar el *Ring Control* a la estructura *Case*.
4. Añadir los casos necesarios en la estructura *Case*.



Case

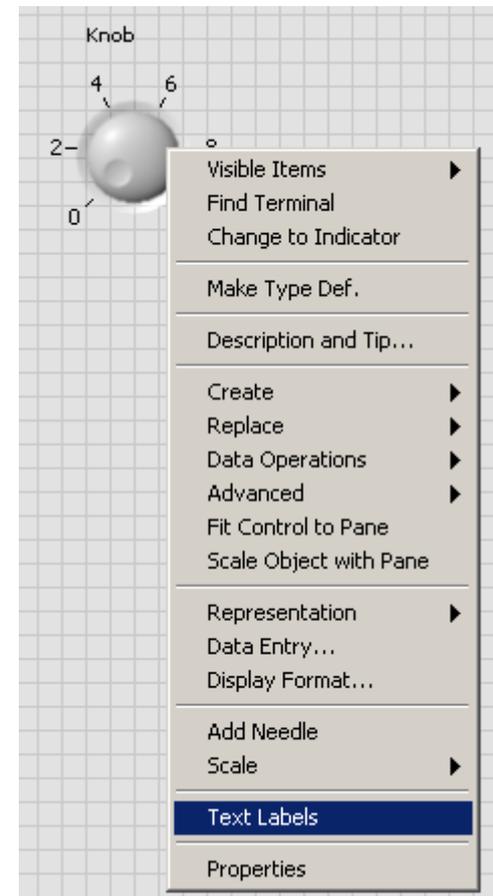
También es posible emplear un control *Dial* o *Knob* como selector para la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

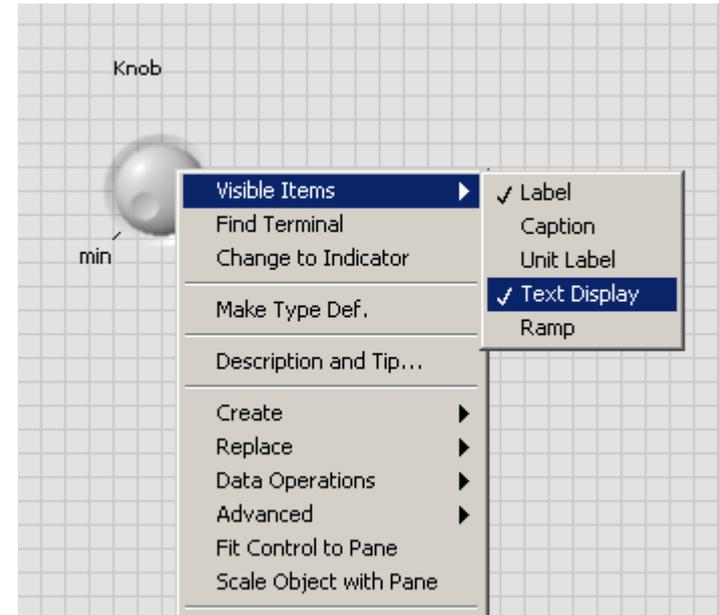
1. Crear el control.
2. **Activar *Text Labels*.**
3. Ocultar *Text Display*.
4. Editar la lista *Text Labels*.
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

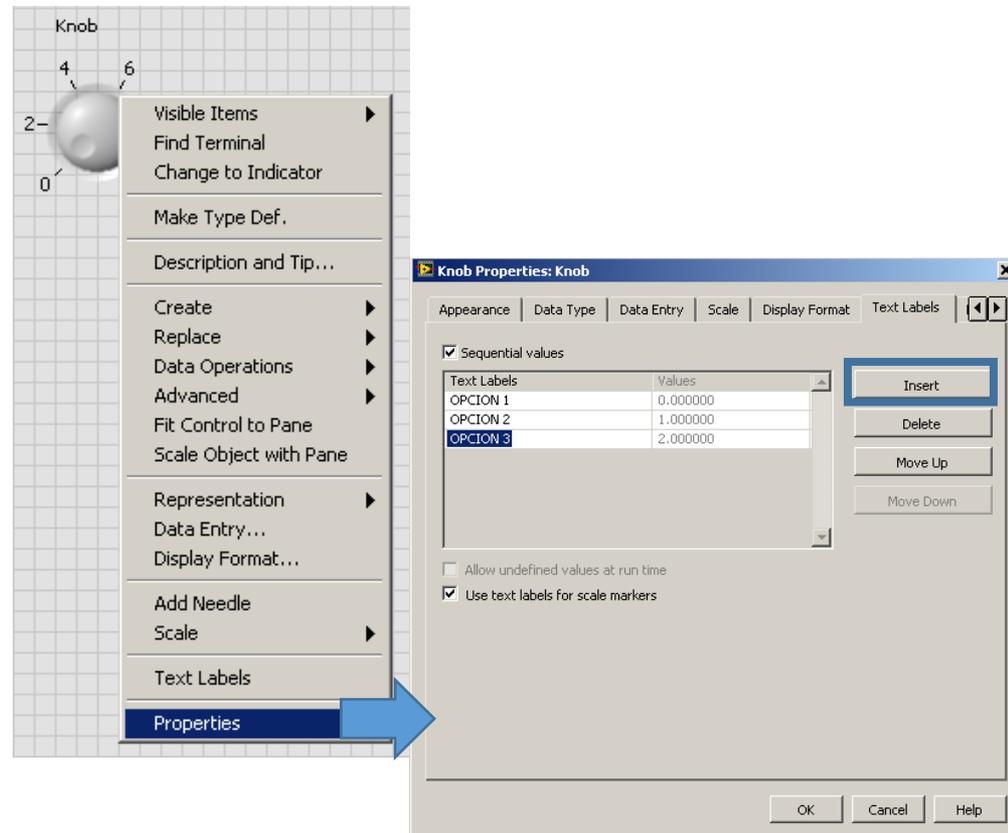
1. Crear el control.
2. Activar *Text Labels*.
3. Ocultar *Text Display*.
4. Editar la lista *Text Labels*.
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

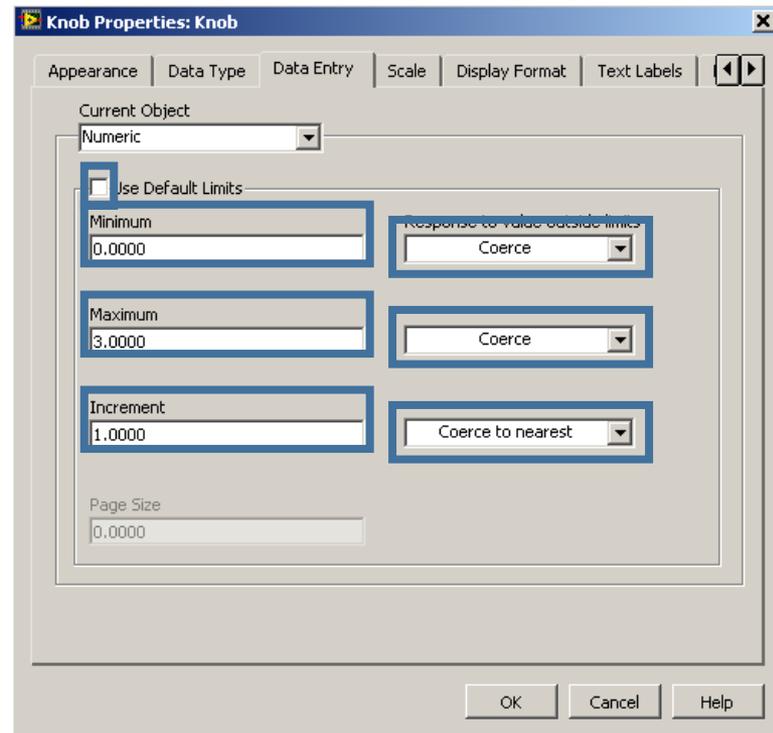
1. Crear el control.
2. Activar *Text Labels*.
3. Ocultar *Text Display*.
4. **Editar la lista *Text Labels*.**
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

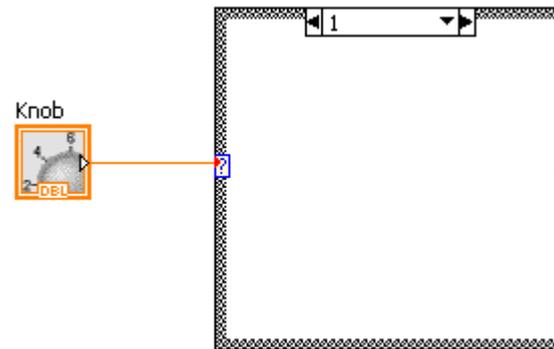
1. Crear el control.
2. Activar *Text Labels*.
3. Ocultar *Text Display*.
4. Editar la lista *Text Labels*.
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

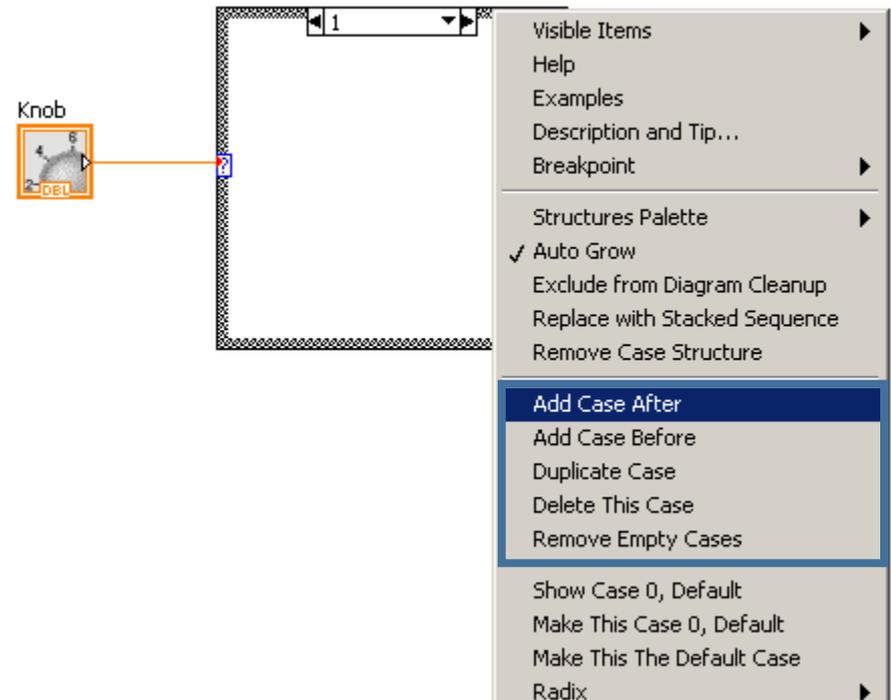
1. Crear el control.
2. Activar *Text Labels*.
3. Ocultar *Text Display*.
4. Editar la lista *Text Labels*.
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Los pasos para usar un *Knob/Dial Control* con la estructura *Case* son los siguientes:

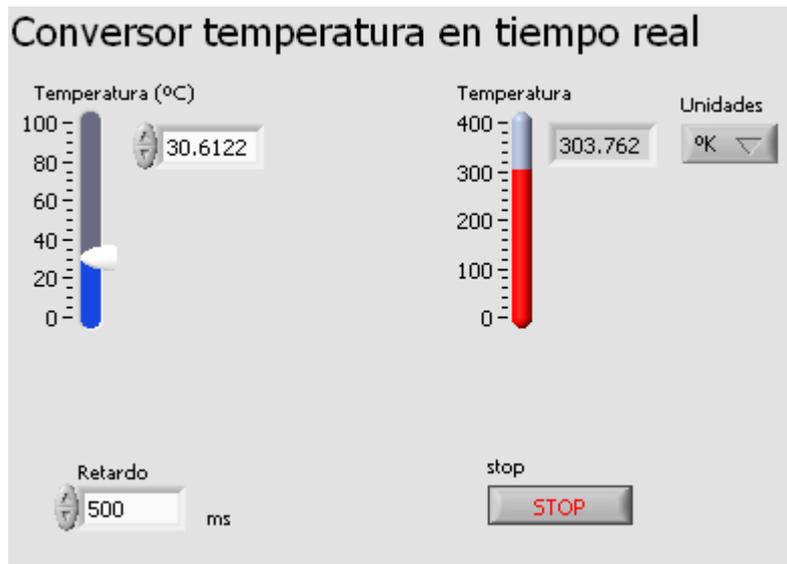
1. Crear el control.
2. Activar *Text Labels*.
3. Ocultar *Text Display*.
4. Editar la lista *Text Labels*.
5. Restringir los datos de entrada.
6. Conectar a la estructura *Case*.
7. Añadir los casos necesarios en la estructura *Case*.



Case

Ejercicio 11: Conversor temperatura en tiempo real

Diseñar el siguiente VI que convierte continuamente grados Celsius en Celsius, Fahrenheit y Kelvin, usando un *Ring Control*.



Ayuda:

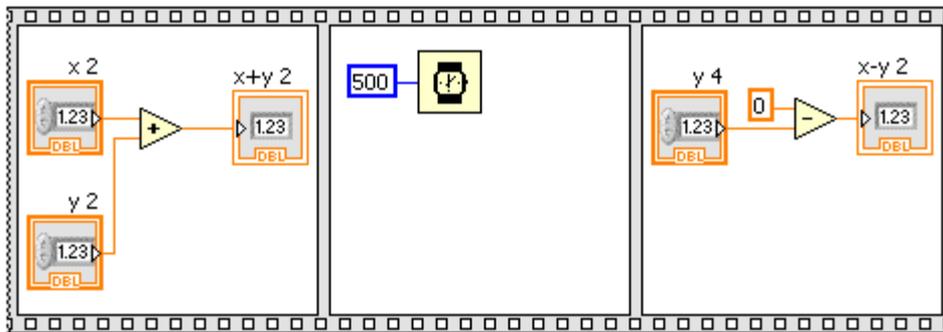
$$T(^{\circ}F) = 32 + 1.8 \cdot T(^{\circ}C)$$

$$T(^{\circ}K) = T(^{\circ}C) + 273.15$$

Sequence

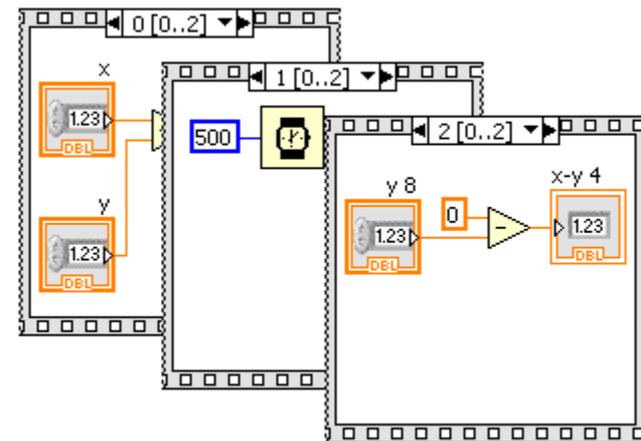
La estructura *Sequence* consiste en uno o más subdiagramas que se ejecutan consecutivamente.

Hay dos tipos:



Flat Sequence

Los subdiagramas se ejecutan secuencialmente de izquierda a derecha.

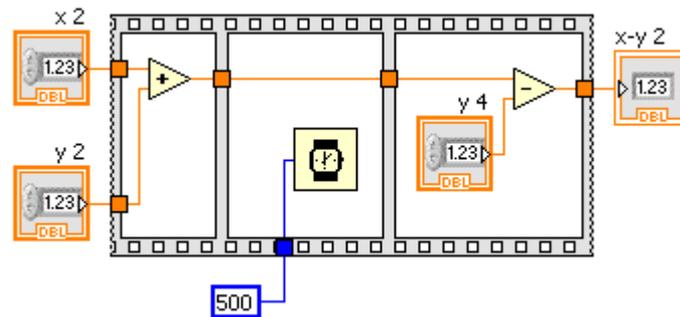


Stacked Sequence

Ahorra espacio en el diagrama de bloques ya que en una sola ventana se pueden diseñar todas las secuencias.

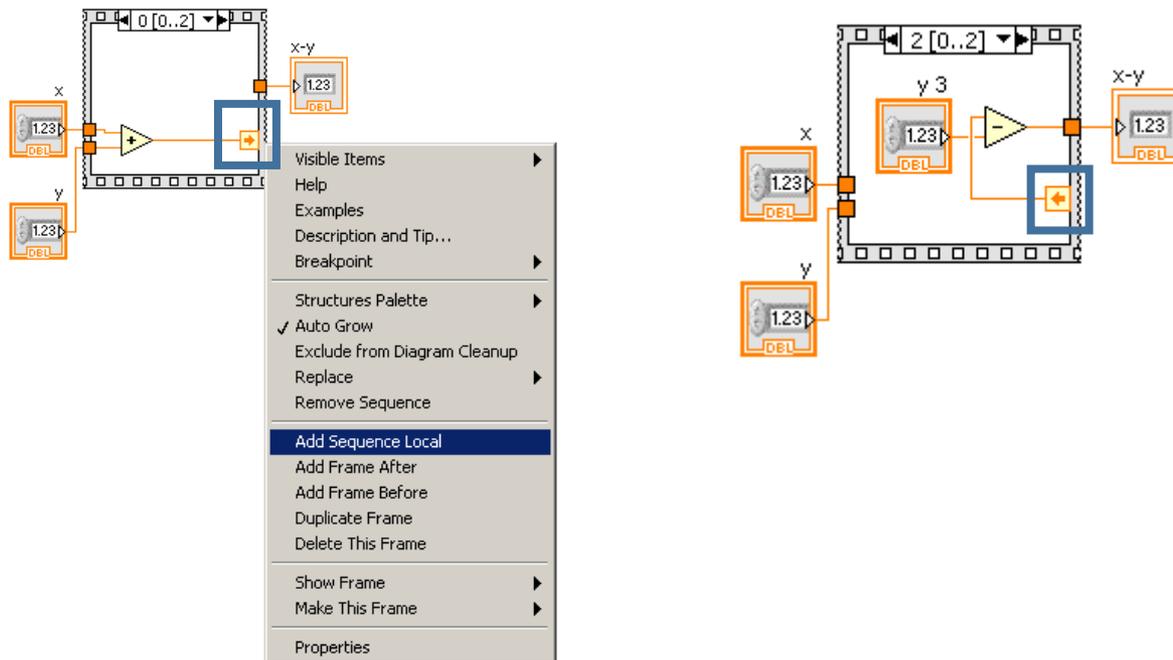
Sequence

Es posible usar túneles para introducir o extraer información de las estructuras *Sequence*. En las *Flat Sequence* también se puede intercambiar información entre *frames*.



Sequence

En las *Stacked Sequence* también se puede intercambiar información con el exterior. Sin embargo, para intercambiar información entre *frames* hay que usar un *Sequence local*.

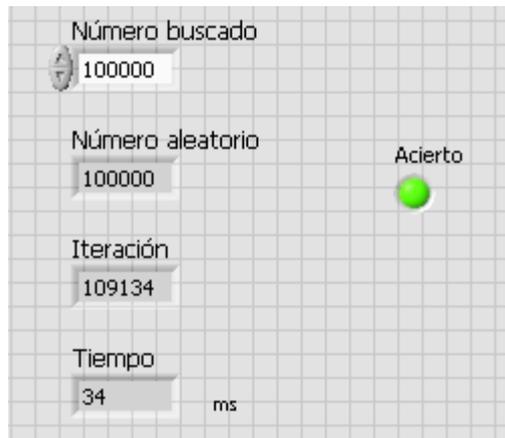


Sequence

Ejercicio 12: Contador intentos

Dado un valor numérico entero entre 0 y 1000000, y empleando la variable aleatoria, indicar cuantas veces intenta el programa la concordancia con el n^o propuesto hasta alcanzar el valor especificado.

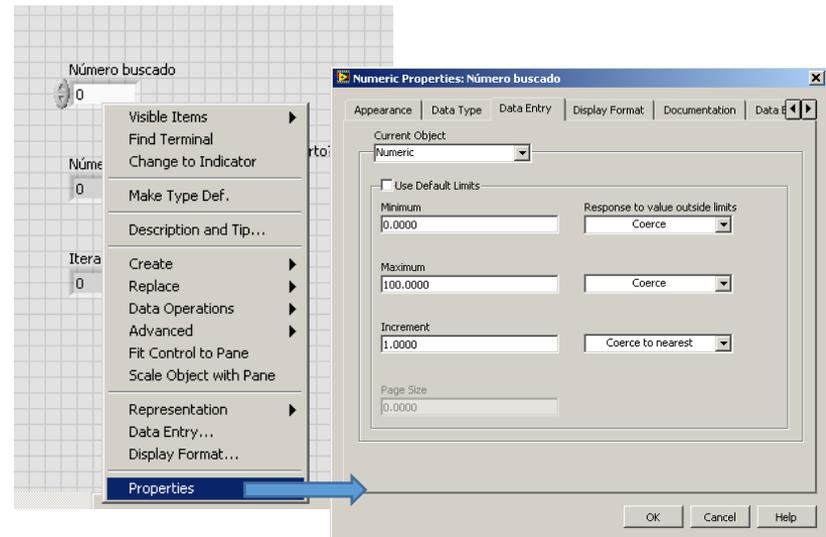
Añadir un indicador que cuente el tiempo transcurrido hasta encontrarlo.



Ayuda: Programming > Timing > Tick Count (ms)



Ayuda: Restringir un control en rango numérico.



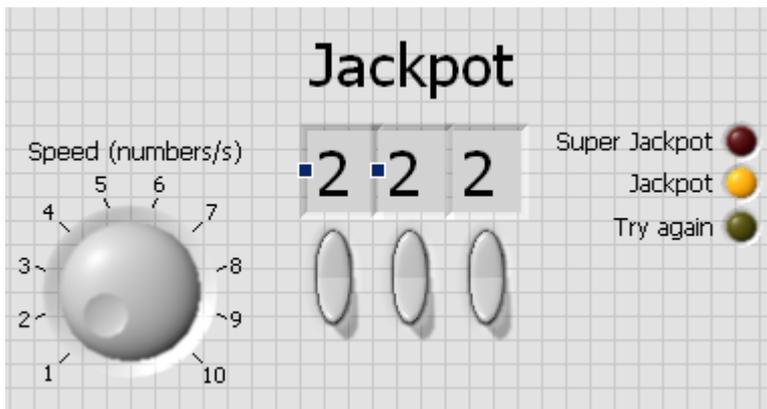
Sequence

Ejercicio 13: Jackpot

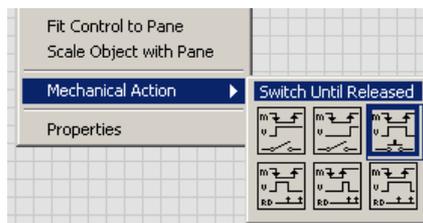
Diseñar el siguiente VI con 3 contadores cíclicos (0 al 9) que se activarán automáticamente y de forma secuencial de izquierda a derecha. Cada contador se detendrá al pulsar el botón correspondiente.

Habrán 3 indicadores led que se encenderán en función del resultado:

- 777 -> "Super Jackpot"
- Tres números iguales, distintos de 7 -> "Jackpot"
- Cualquier otro caso -> "Try again"



Ayuda 1: Usar botón como pulsador



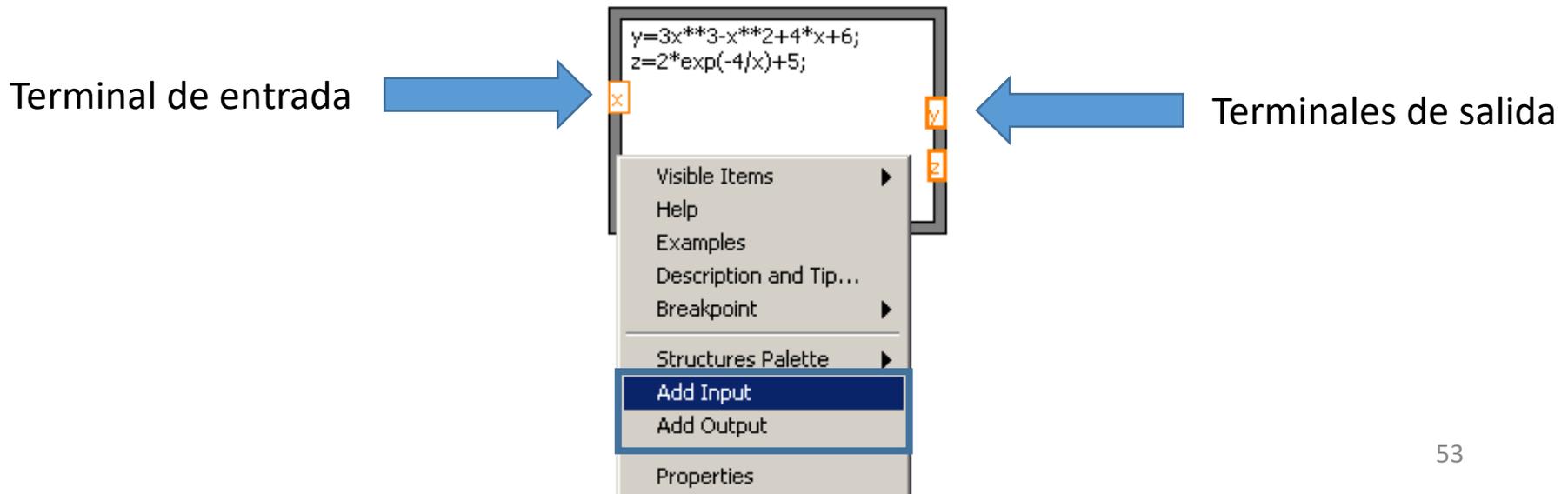
Ayuda 2: Resto de división
Programming > Numeric >
Quotient & Remainder



Formula Node

Es un tipo de estructura que contiene una o más fórmulas separadas por un punto y coma.

Una vez escrita la fórmula en el interior del rectángulo, sólo hay que añadir los terminales que harán la función de variables de entrada o salida.



Formula Node

Operadores

**	exponentiation
+, -, !, ~, ++, and --	unary plus, unary negation, logical not, bit complement, pre- and post-increment, pre- and post-decrement
*, /, %	multiplication, división, modulus (remainder)
+ and -	addition and subtraction
>> and <<	arithmetic shift right and shift left
>, <, >=, and <=	greater, less, greater or equal, and less or equal
!= and ==	inequality and equality
&, &&	bit and, logical and
,	bit or, logical or
^	bit exclusive or
? :	conditional evaluation
= op=	assignment, shortcut operate and assign op can be +, -, *, /, >>, <<, &, ^, , %, or **. = op= is not available in Expression Nodes

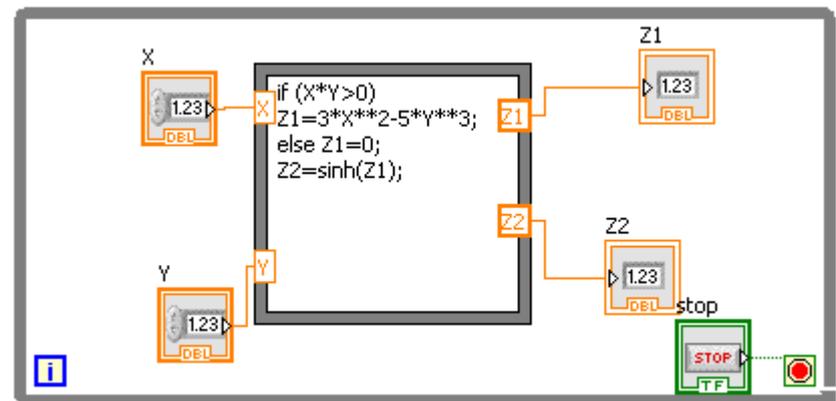
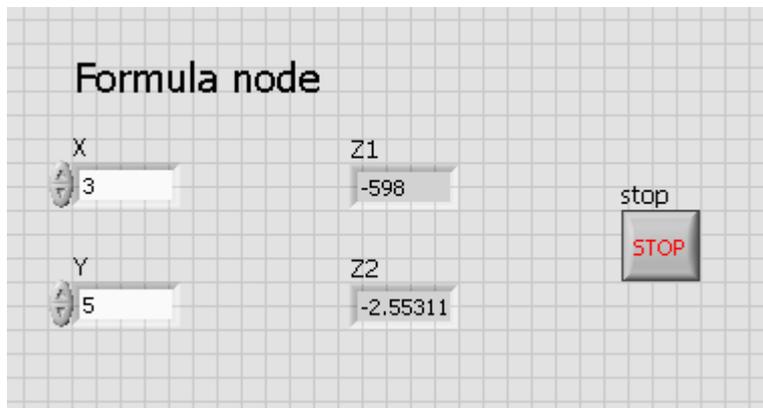
Formula Node

Funciones

abs	acos	acosh	asin	asinh
atan	atan2	atanh	ceil	cos
cosh	cot	csc	exp	expm1
floor	getexp	getman	int	intrz
ln	lnp1	log	log2	max
min	mod	pow	rand	rem
sec	sign	sin	sinc	sinh
sizeofDim	sqrt	tan	tanh	

Formula Node

Ejemplo: Implementar el siguiente instrumento virtual.

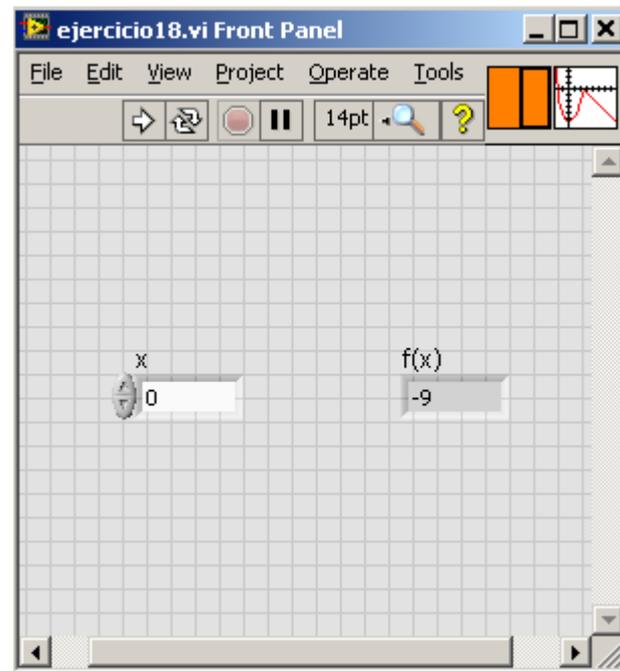


Formula Node

Ejercicio 14: Función definida a trozos

Diseñar un sub-instrumento virtual que evalúe la siguiente función definida a trozos:

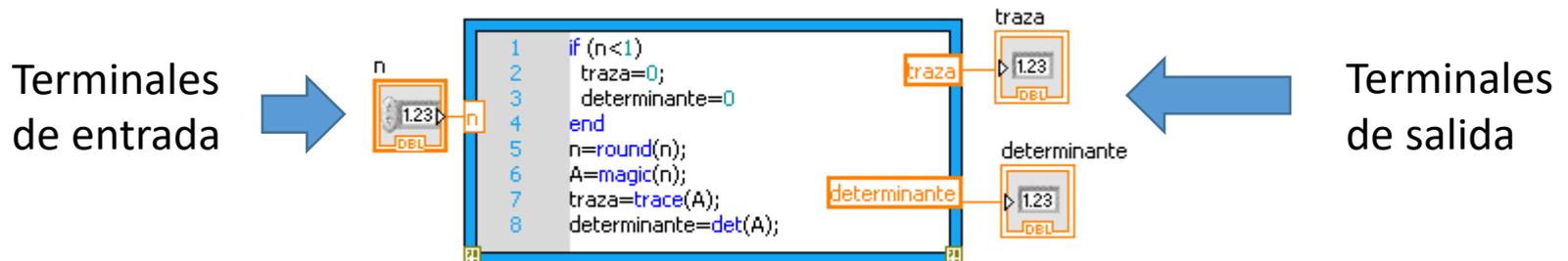
$$f(x) = \begin{cases} x^2 - 9, & x < 3 \\ -x + 3, & x \geq 3 \end{cases}$$



MathScript Node

Al igual que el *Formula Node*, el *MathScript Node* añade matemática textual a LabVIEW, con un compilador para archivos .m, desarrollados en MATLAB® o GNU Octave.

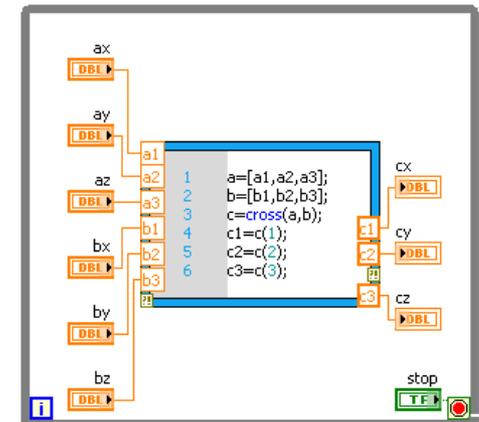
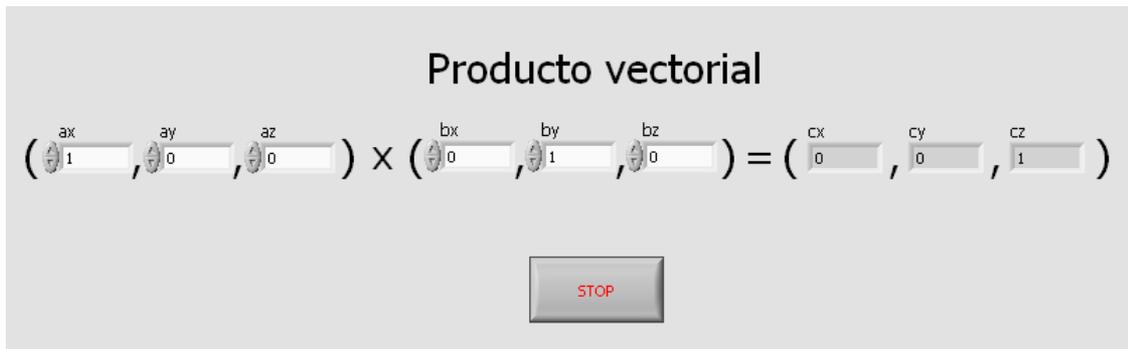
Se pueden importar código directamente o escribir uno propio, usando la sintaxis propia de esos lenguajes de programación.



MathScript Node

Ejemplo: Producto vectorial

Diseñar un VI que calcule las raíces de cualquier ecuación polinómica de segundo grado, dados los valores de los coeficientes.



MathScript Node

Ejercicio 15: Raíces ecuación de segundo grado

Diseñar un VI que calcule las raíces de cualquier ecuación polinómica de segundo grado, dados los valores de los coeficientes.

Raíces ecuación segundo grado

$y = a*x^2 + b*x + c$

a	<input type="text" value="2"/>	Real	Imaginario	<input type="button" value="STOP"/>	
b	<input type="text" value="0"/>	x1	<input type="text" value="1"/>		<input type="text" value="0"/>
c	<input type="text" value="-2"/>	x2	<input type="text" value="-1"/>		<input type="text" value="0"/>

Property node

- Permiten leer o escribir las propiedades de los elementos del panel frontal desde el diagrama de bloques.
- Es posible leer/escribir más de una propiedad del mismo elemento usando un solo *Property Node*.

Numeric

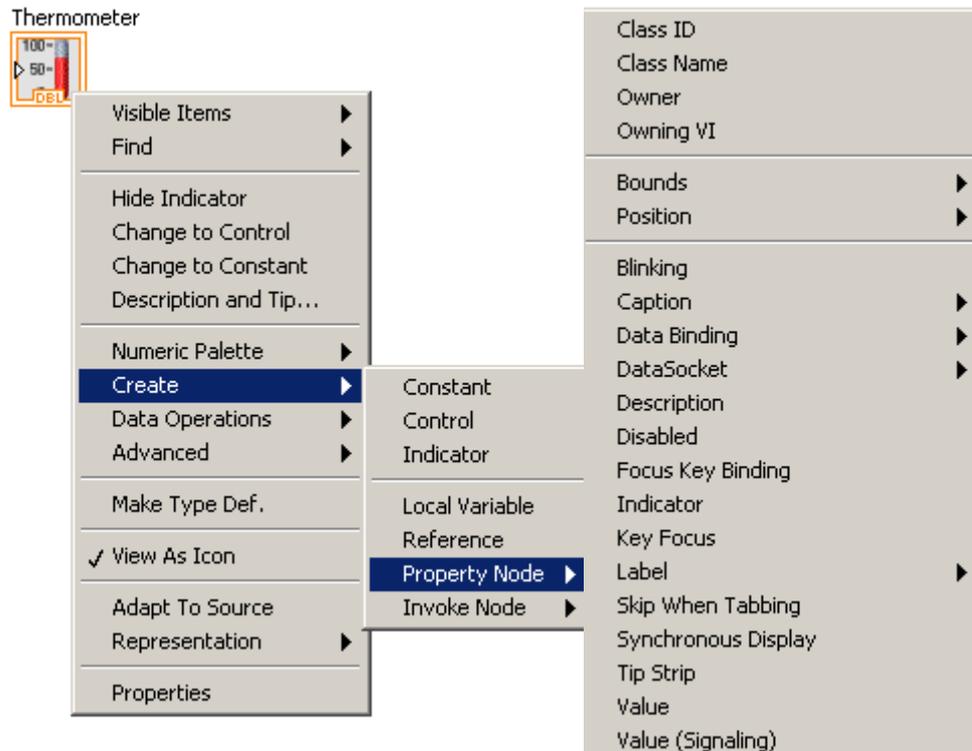
Value	▶
▶ Val(Sgnl)	
Visible	▶
Container Bounds	▶
Height	▶
Width	▶
Is XControl?	▶
Data Entry Limits	▶
Increment	▶

Boolean

SyncDisp	▶
TipStrip	▶
Value	▶
▶ Val(Sgnl)	
Visible	▶
Container Bounds	▶
Height	▶
Width	▶
Is XControl?	▶

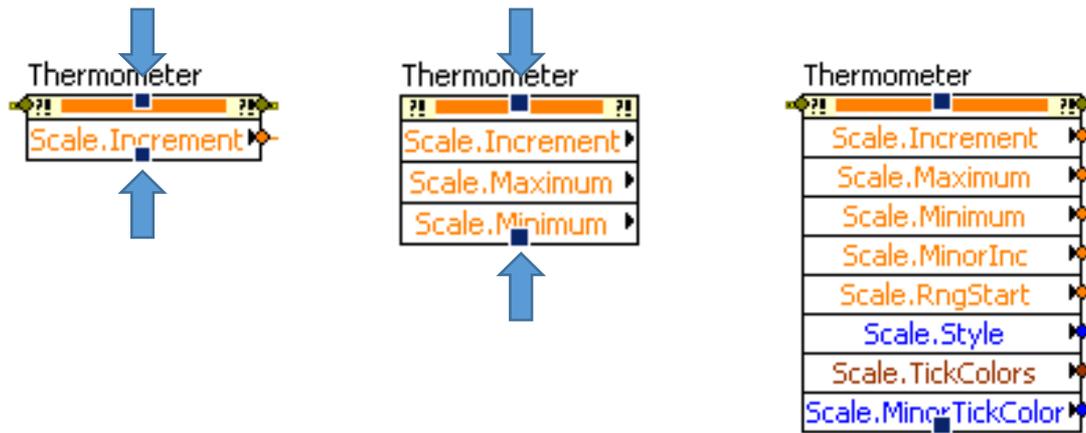
Property node

- La forma más sencilla de crear un *Property Node* es a través del menú emergente de cualquier elemento en el diagrama de bloques:



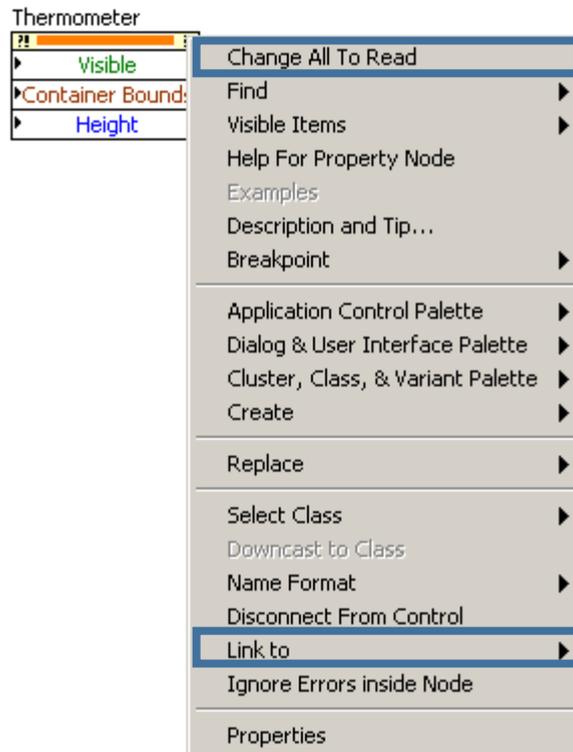
Property node

- Una vez creado, se pueden añadir más propiedades usando los tiradores:



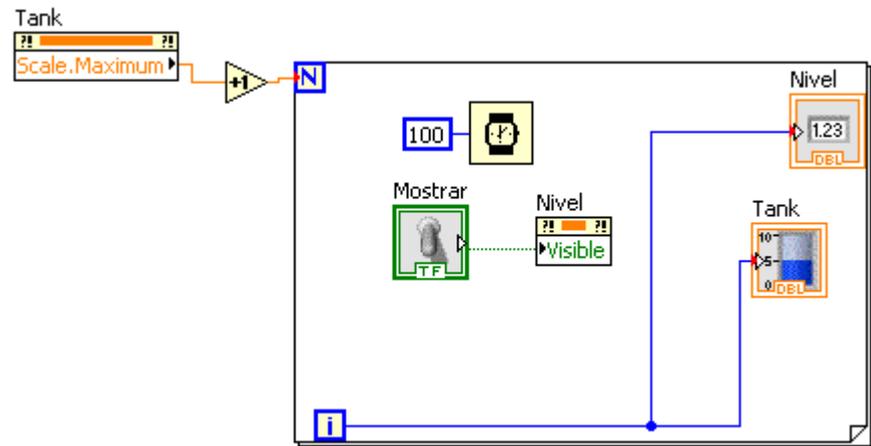
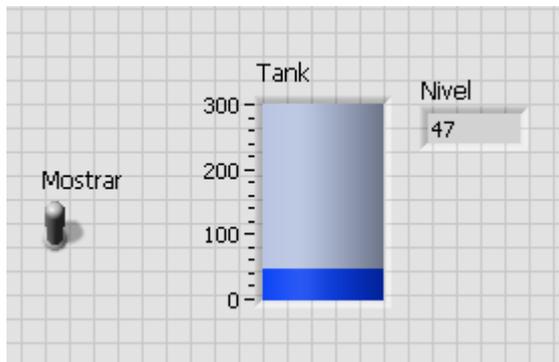
Property node

- En el menú contextual se pueden cambiar el modo lectura/escritura, acceder a otras propiedades o unir el *Property Node* a otro elemento:



Property node

- **Ejemplo:** Simular el completo llenado de un tanque a un ritmo de 1 unidad cada 100 ms, independientemente del valor capacidad del mismo. Un botón controlará la visibilidad de un display digital asociado al nivel del tanque.



Property node

Ejercicio 16: Calculadora avanzada

Diseñar la siguiente calculadora. Las operaciones se efectuarán al accionar el botón correspondiente, mostrando el resultado en el indicador.

- Añadir un control adicional para mostrar/ocultar las operaciones exponenciación (^) y resto de división (%).
- La operación resto de división entera se realizará sobre el valor entero más cercano de ambos operandos.
- Siempre que el operando B sea cero, desactivar los botones división y resto de división.
- Cada vez que un resultado sea 10, cambiar la etiqueta del indicador por “¡Enhorabuena!”

