

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Formula Node: Código C  
Dentro de LabVIEW

# Contenido

Introducción Teórica de LabVIEW:.....	2
Introducción al Entorno de LabVIEW:.....	2
<b>Front Panel:</b> Ventana Gris con la Interfaz del Programa .....	4
<b>Block Diagram:</b> Ventana Blanca con la Lógica del Programa (Bloques) .....	4
<b>Front Panel o Block Diagram</b> - Show Context Help: Descripción de Bloques .....	5
<b>Front Panel y Block Diagram:</b> Navegar de una Ventana a Otra .....	6
<b>Block Diagram</b> - Cambiar Nombre a los Bloques: Nombre de los elementos en el <b>Front Panel</b> .....	7
<b>Block Diagram</b> - Highlight Execution: Correr Más Lento el Programa .....	8
<b>Coertion dot:</b> Conversión Automática de Datos por Parte de LabVIEW .....	8
<b>Block Diagram</b> - Clean Up Diagram: Organizar Automáticamente los Bloques del VI .....	8
Programa: Formula Node .....	9
Desarrollo del Programa: Señal Tangente con Formula Nodes .....	9
<b>Block Diagram</b> - Formula Node: Uso de Código C en LabVIEW .....	9
<b>Block Diagram</b> - Bucle For: Iteraciones Finitas para la Graficación de una Señal .....	13
<b>Block Diagram</b> - Divide: División de dos Números Cualquiera .....	13
<b>Front Panel</b> - Waveform Graph: Ventana que Muestra una Señal ( <b>Array</b> ) .....	15
Ejecución del Programa: Waveform Graph (Arrays) de un Formula Node .....	16



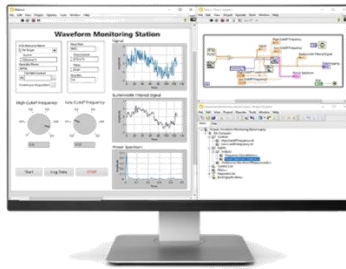
# Introducción Teórica de LabVIEW:

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

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

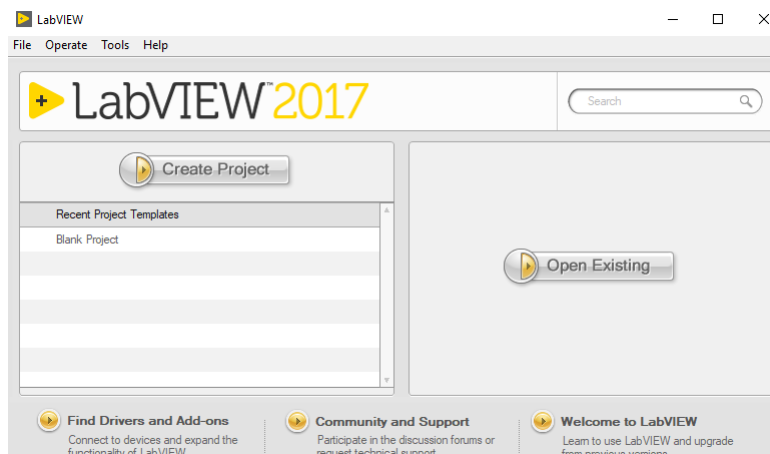
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

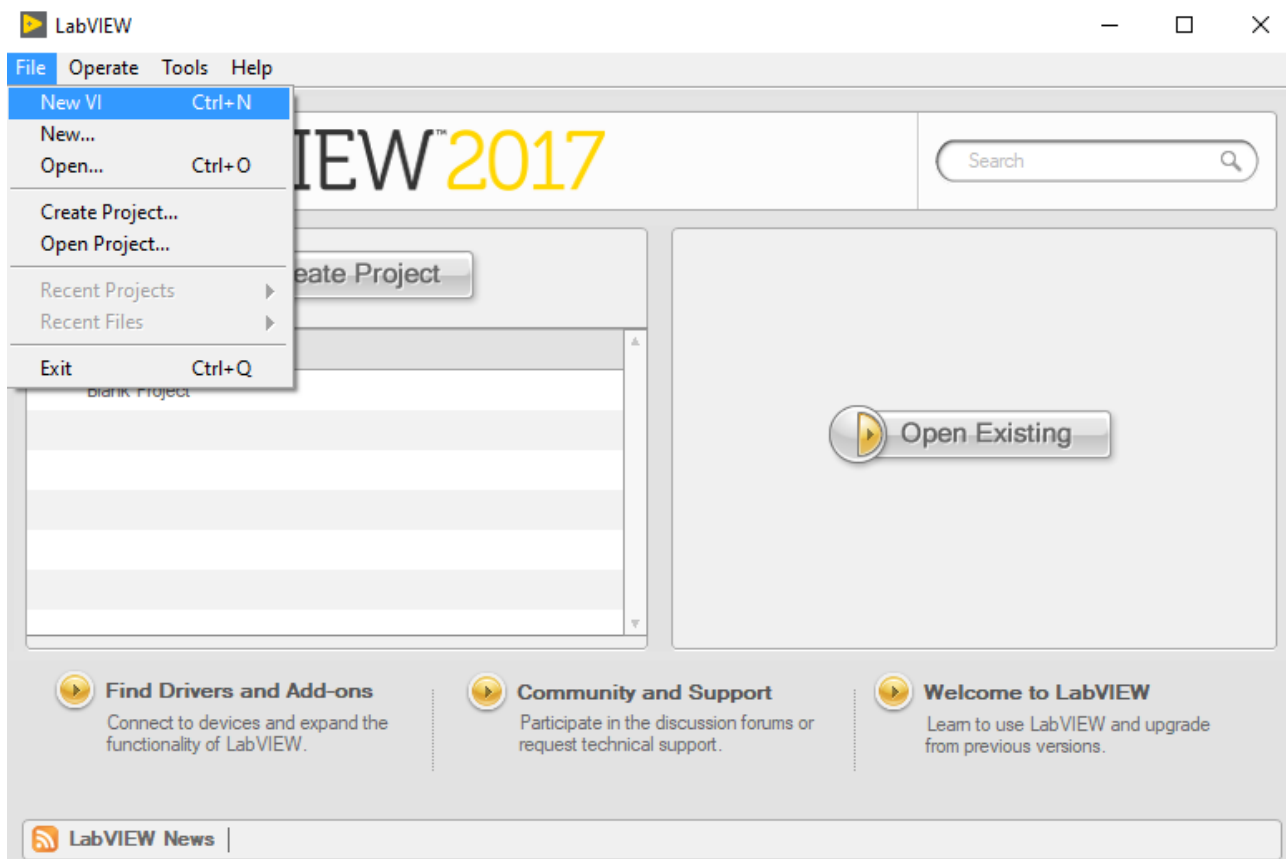


# Introducción al Entorno de LabVIEW:

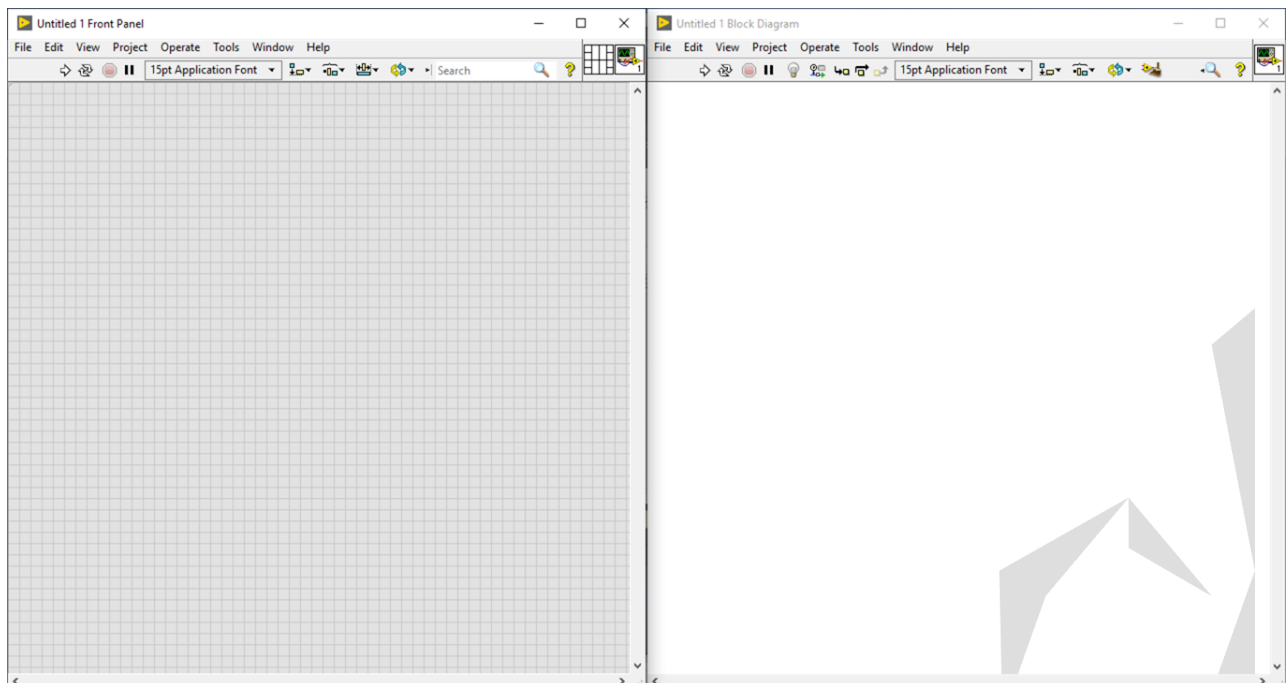
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

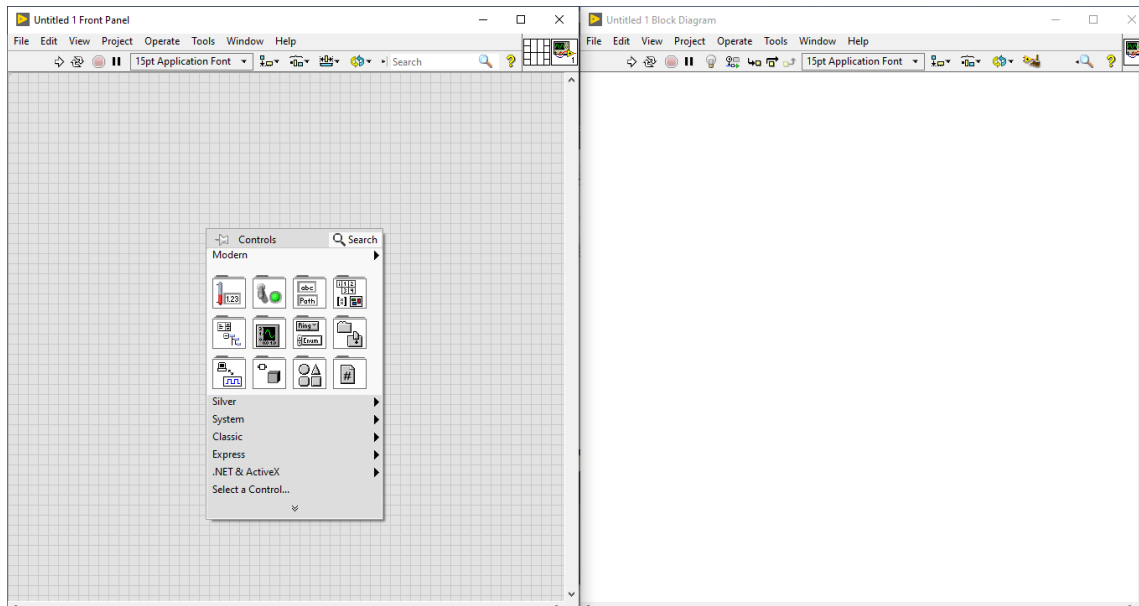


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



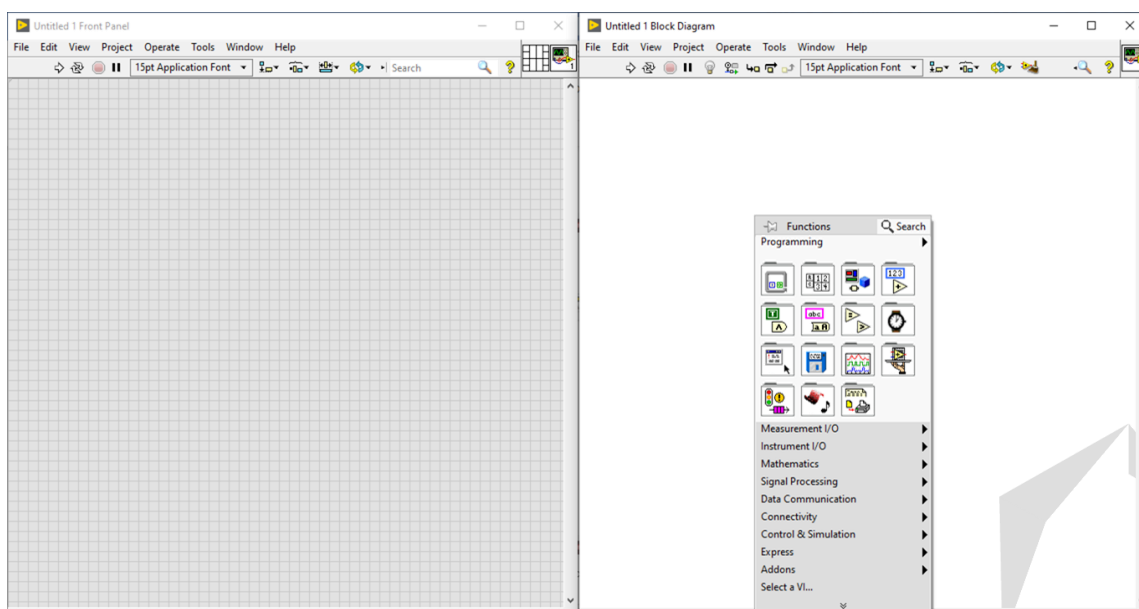
## Front Panel: Ventana Gris con la Interfaz del Programa

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



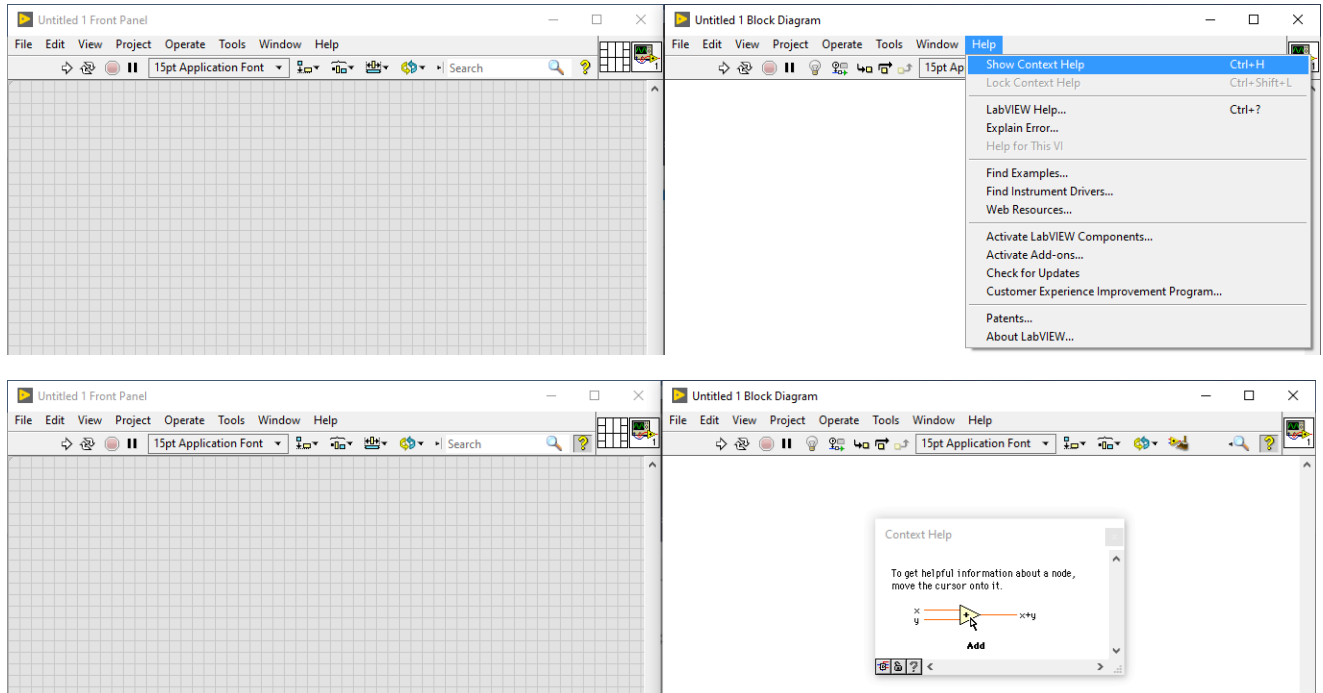
## Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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



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

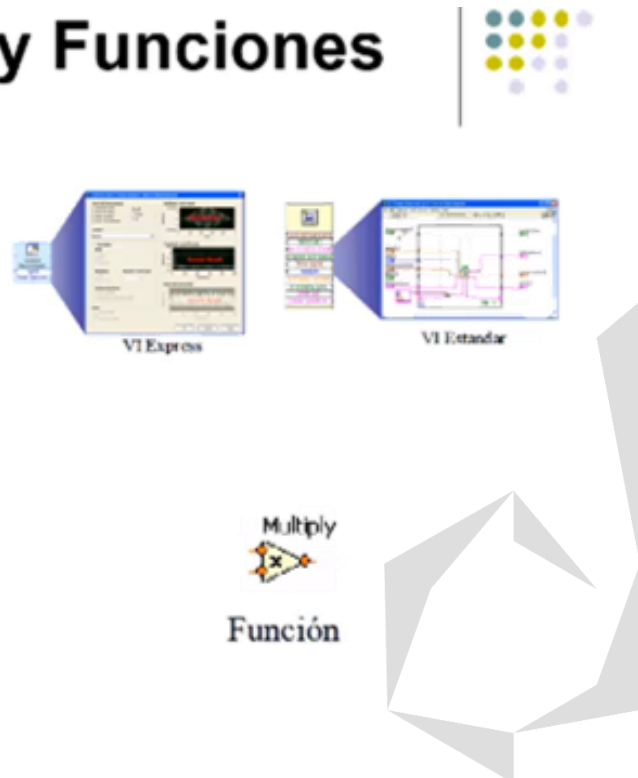
Seleccionando la opción de Help → Show Context Help, aparecerá una ventana emergente que explicará las propiedades de los bloques que se puede seleccionar, mostrando una descripción de su función, imágenes explicativas y significado de sus pines de entrada y salida.



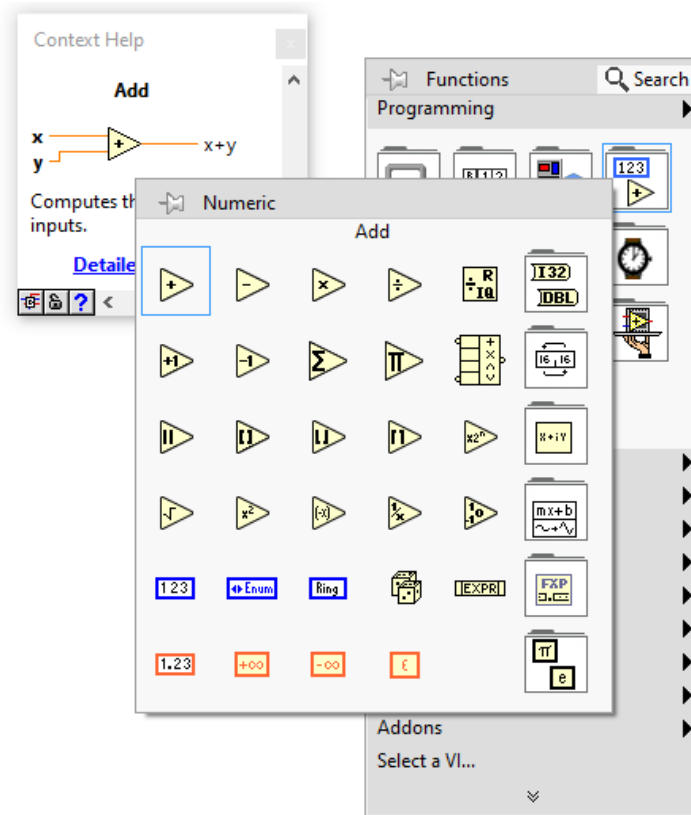
Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:

## VIs Express, VIs y Funciones

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



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

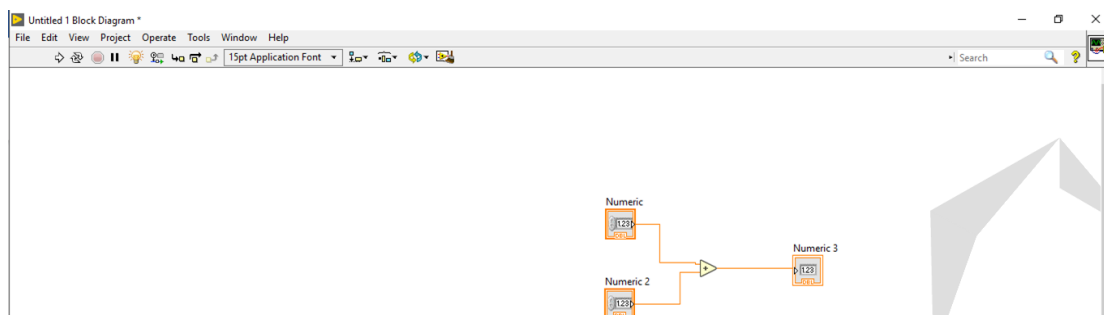


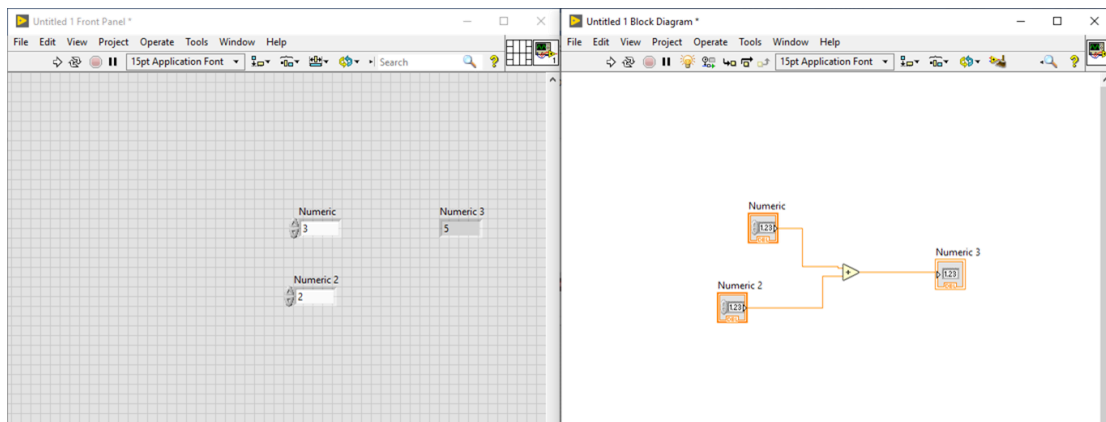
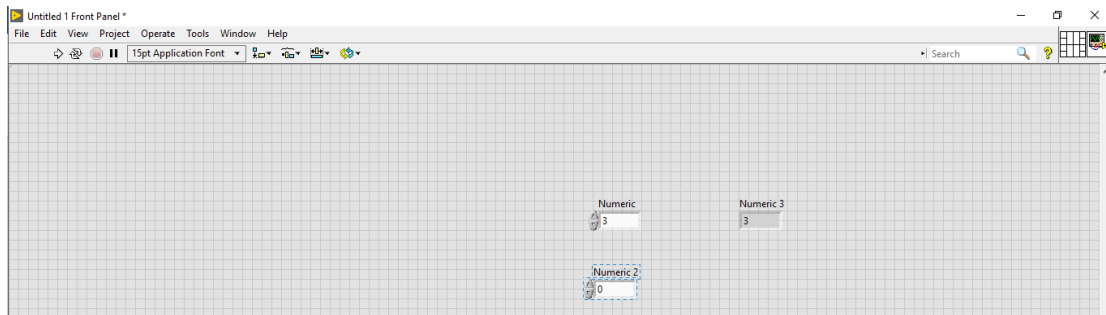
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

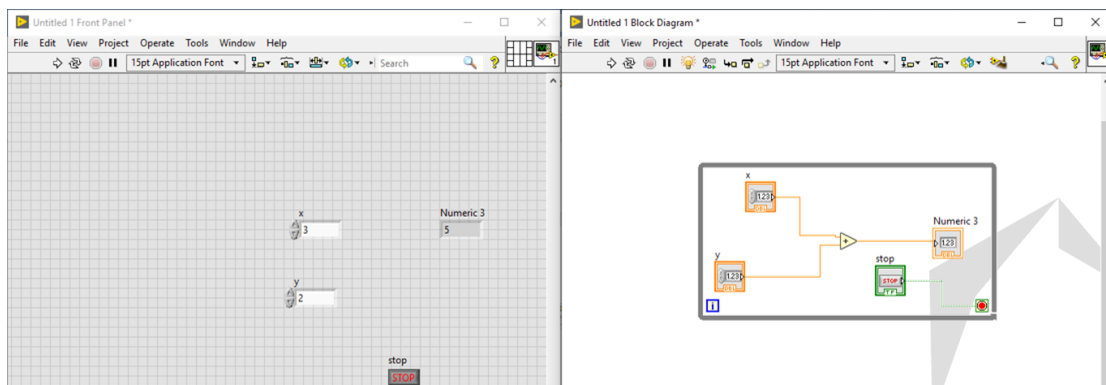
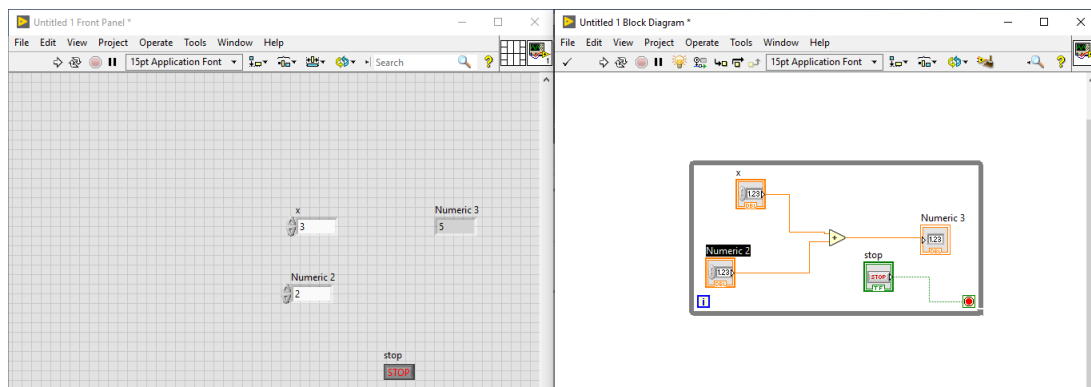
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.



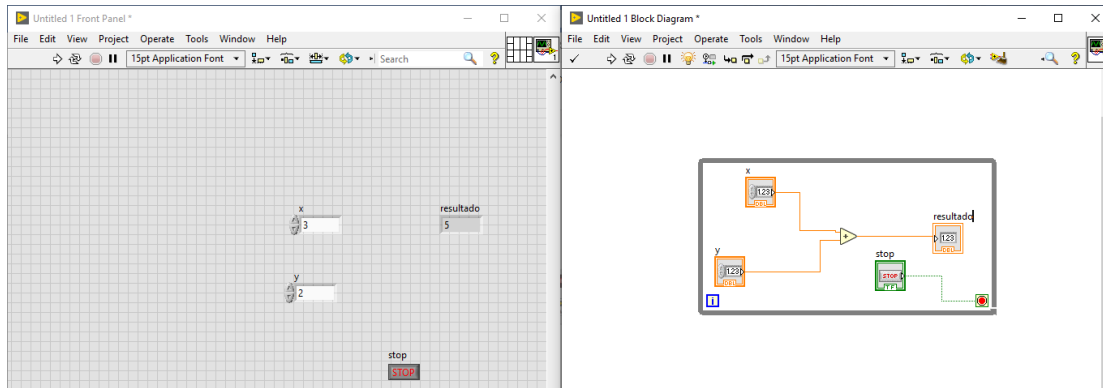


## Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

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

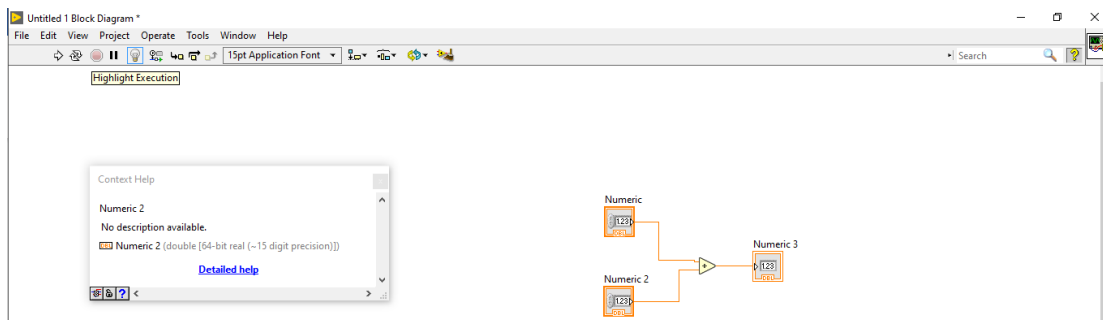






## Block Diagram - Highlight Execution: Correr Más Lento el Programa

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

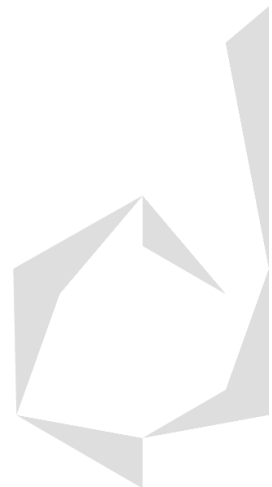


## Coertion dot: Conversión Automática de Datos por Parte de LabVIEW

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

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

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



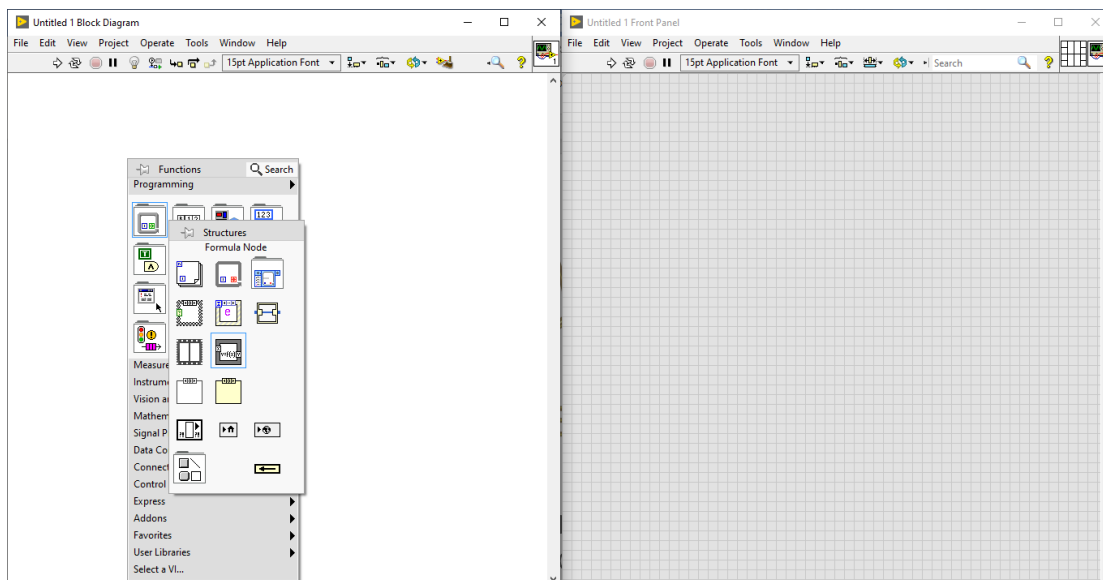
# Programa: Formula Node

Dentro del entorno de NI hay una forma de usar un tipo de código C que es específico de LabVIEW, esto se hace a través de un bloque llamado Formula Node, con él se graficará una señal a través de código.

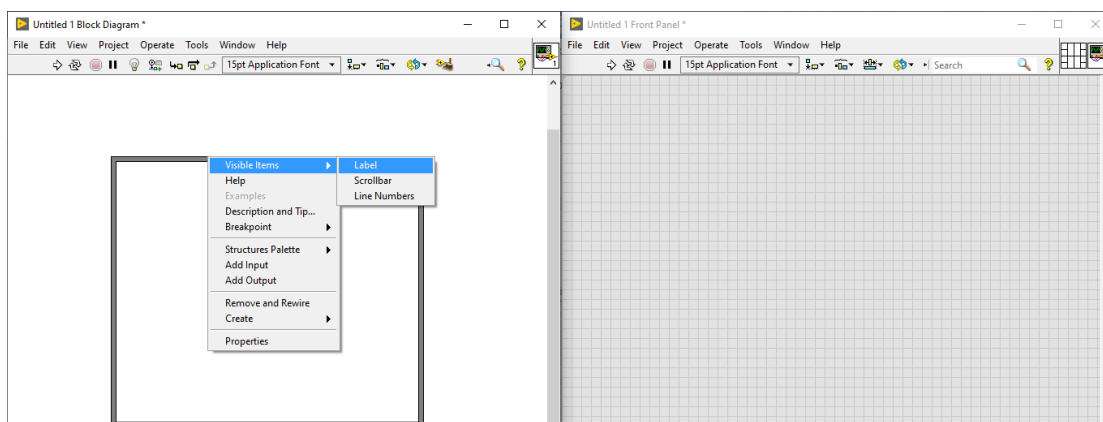
## Desarrollo del Programa: Señal Tangente con Formula Nodes

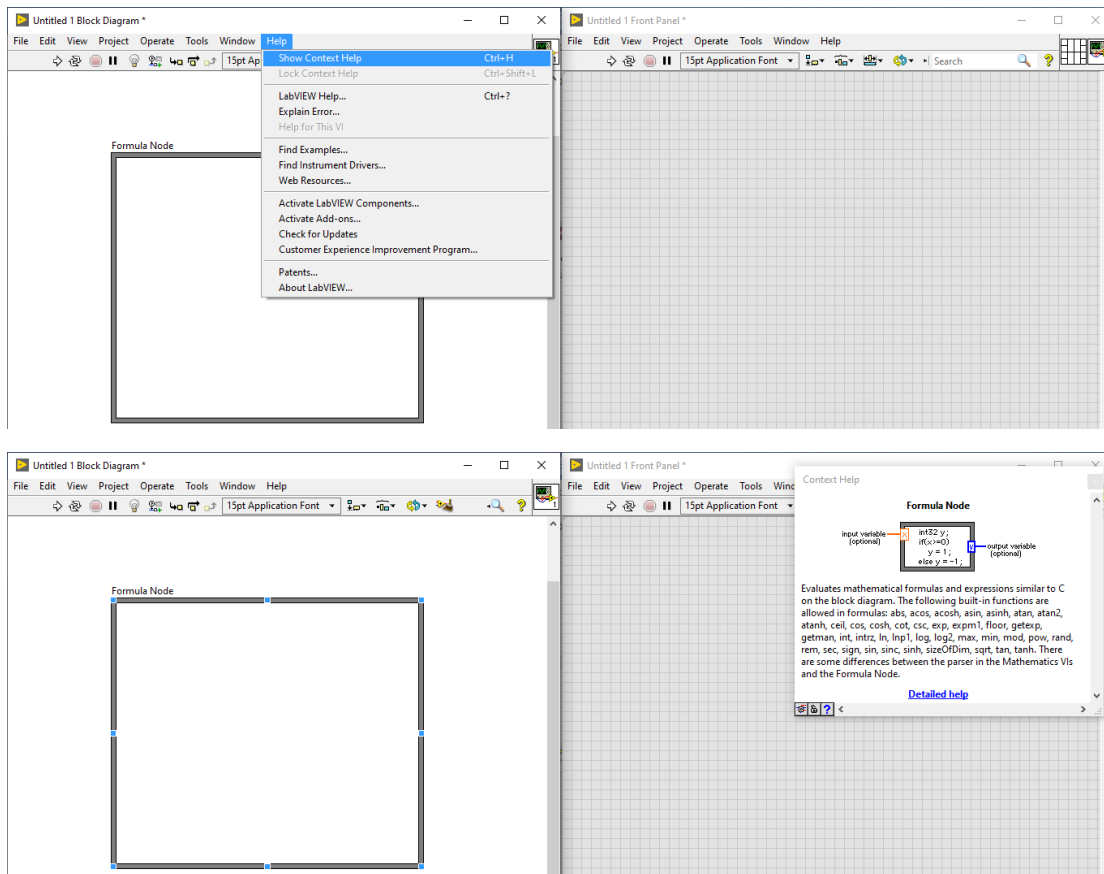
### Block Diagram - Formula Node: Uso de Código C en LabVIEW

Dentro de este bloque llamado Formula Node podemos poner el código que nosotros queramos de forma escrita por medio del lenguaje C ANSI, que es el lenguaje C original de forma estándar que contenía solo 32 comandos, es el más sencillo ya que no reconoce librerías de C#, C++, ni ninguna de ellas.



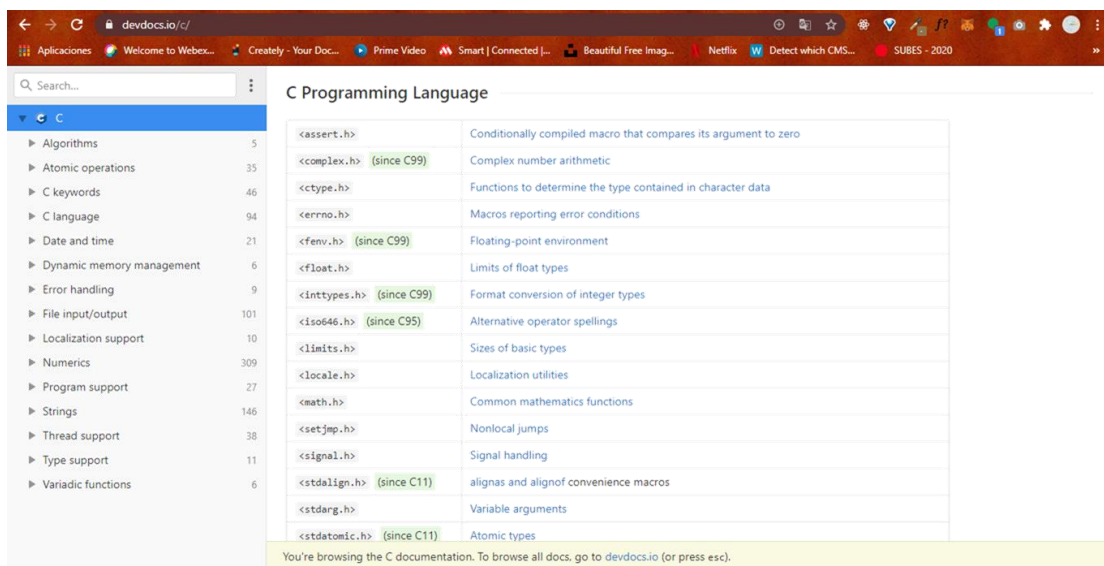
**Mostrar nombre del bloque: Clic derecho → Visible Items → Label.**



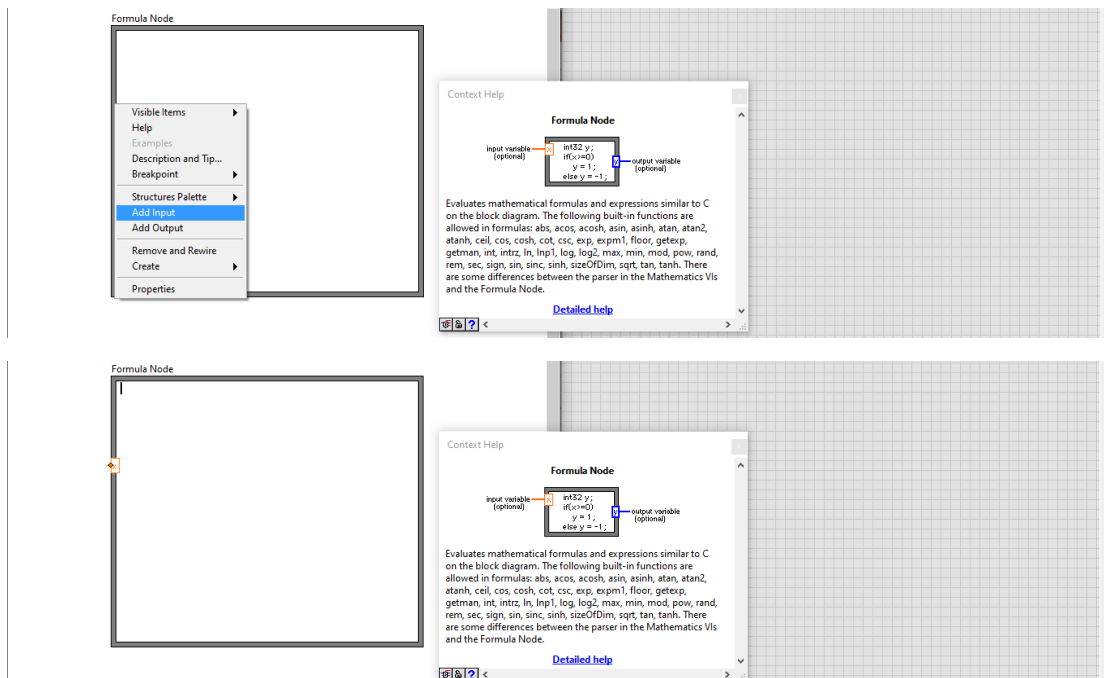


La documentación de C ANSI se encuentra en el link mostrado a continuación:

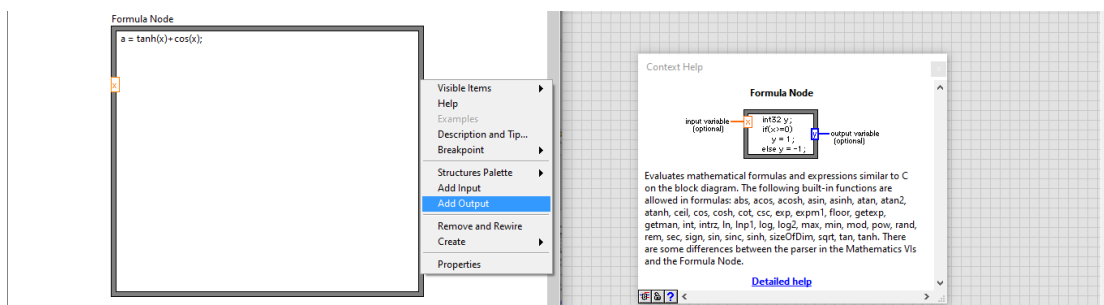
<https://devdocs.io/c/>



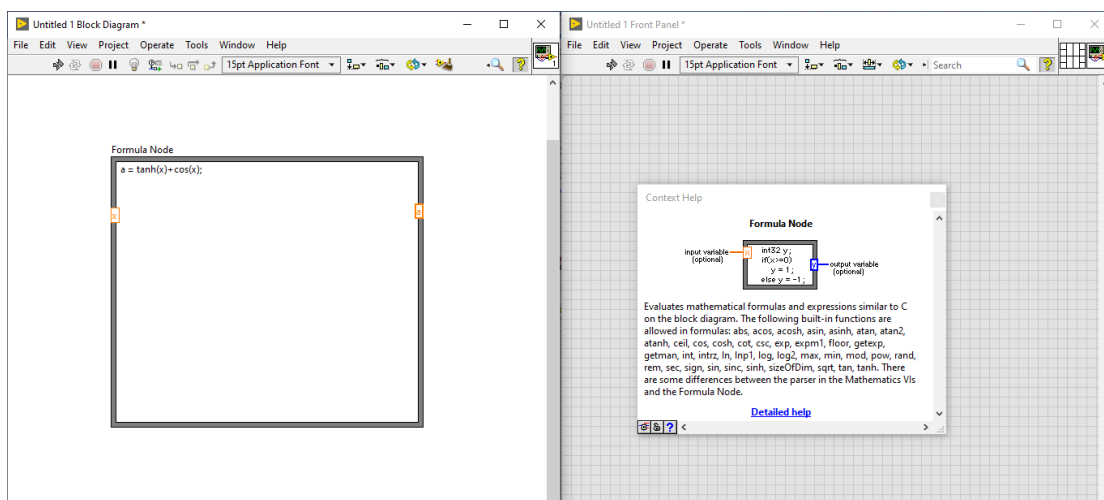
No necesito crear con código las variables de entrada y salida, para poner esas debo dar clic derecho en el bloque del Formula Node y seleccionar la opción de Add input o Add output y se creará un túnel de entrada o salida donde deberé conectar mis diferentes bloques.



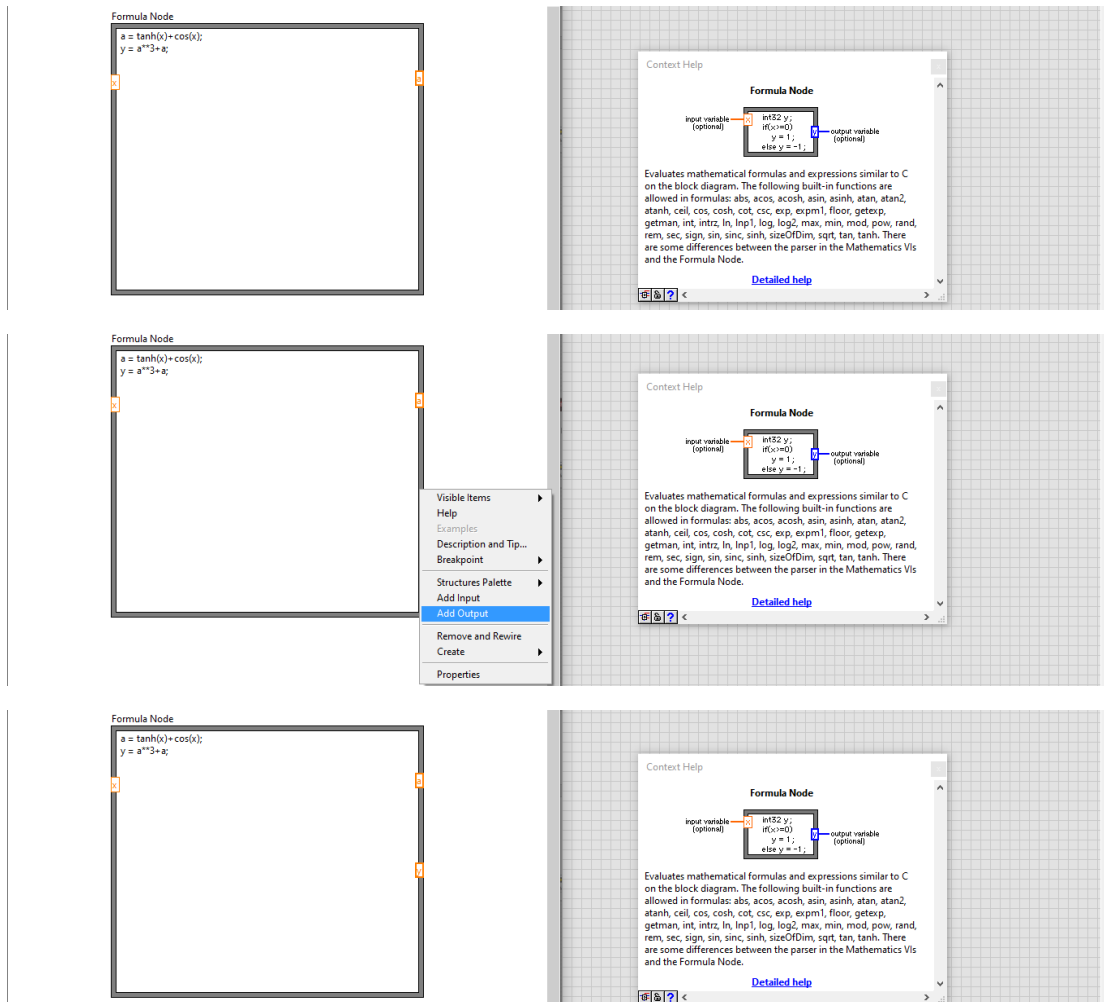
Siempre las declaraciones de código las debo cerrar con punto y coma (;).



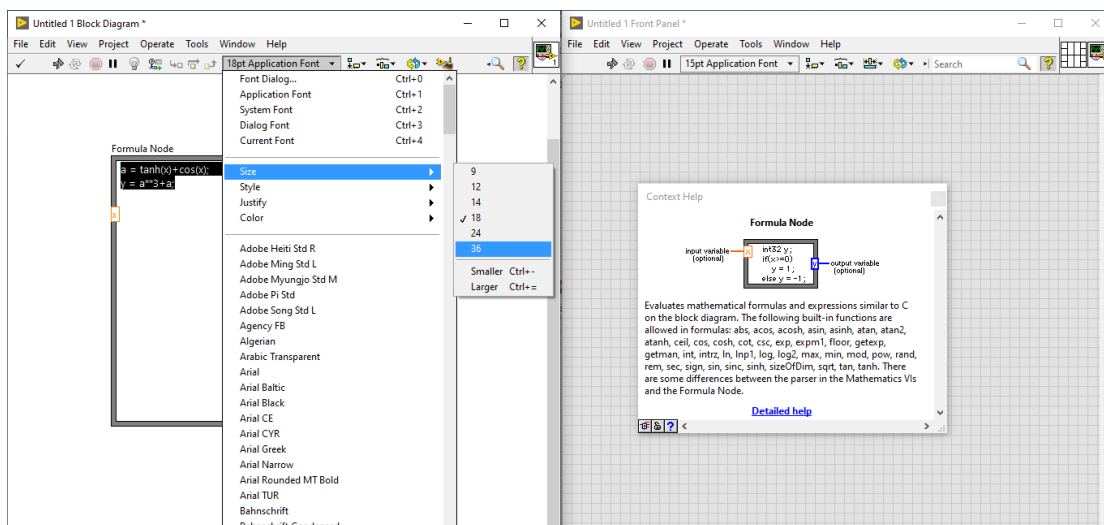
Así definimos cuales son las entradas y cuales las salidas.



**Operaciones Matemáticas en el Formula Node:** Poner \*\* significa exponente, por lo tanto, si declaro la ecuación  $y = a^{*3}$ , estoy diciendo  $a^3$ , ya que el símbolo  $^$  no lo reconoce el bloque.

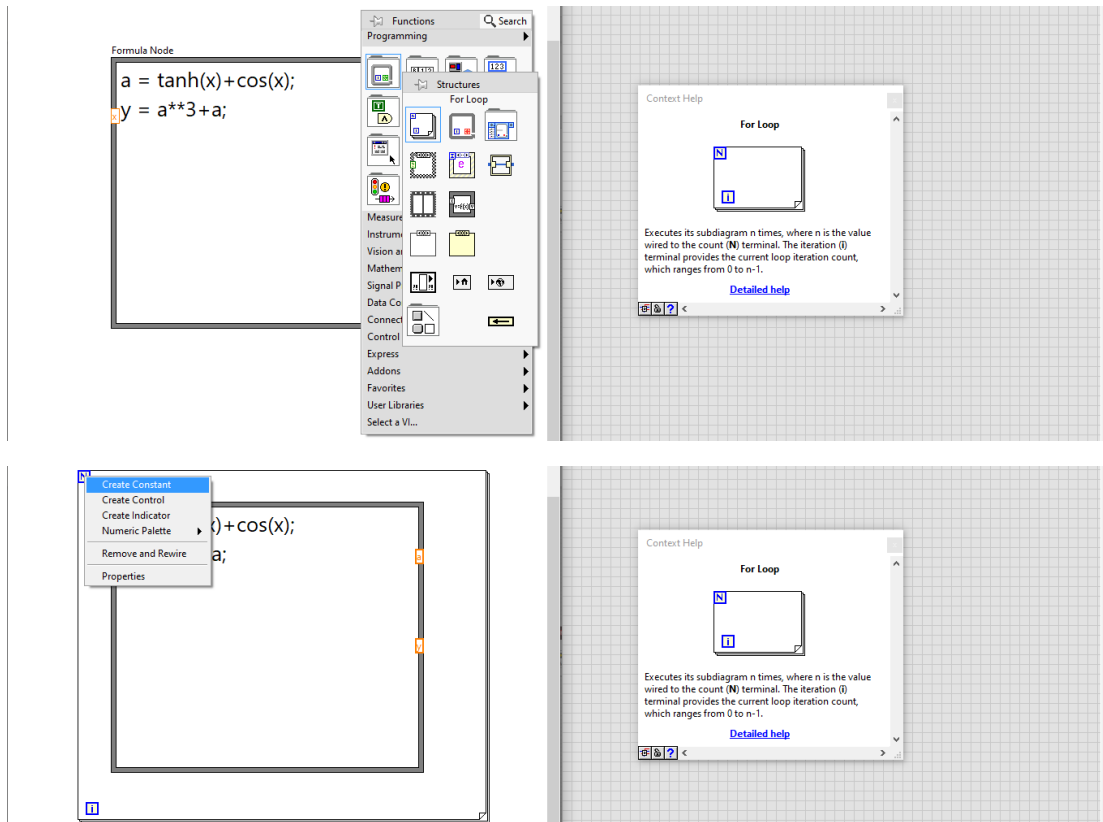


**Cambiar Tamaño de Letra:** Menú Superior → Application Font → Size → Tamaño de Letra.



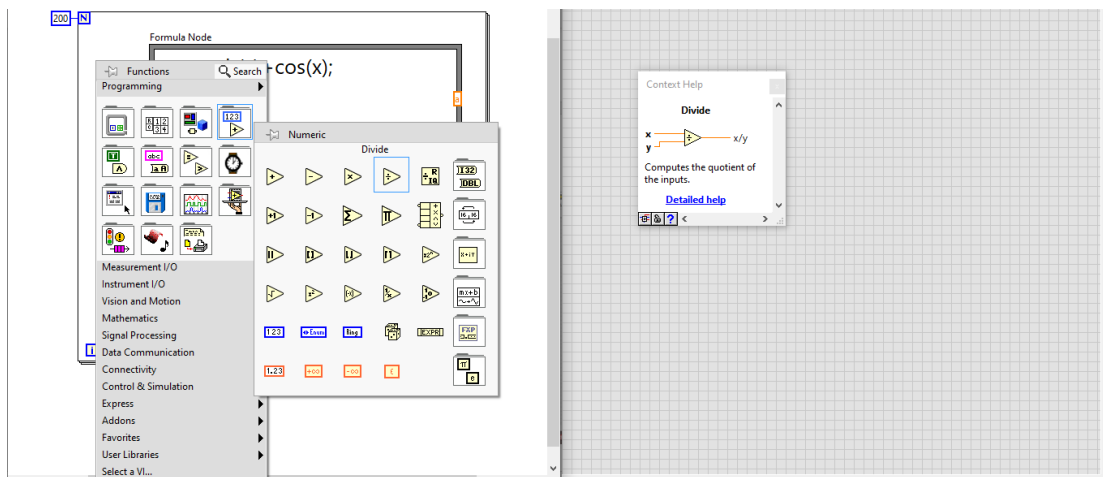
## Block Diagram - Bucle For: Iteraciones Finitas para la Graficación de una Señal

El ciclo for hace que el programa se ejecute un número finito de veces, indicado por las variables N e i, en el Bucle la variable N nos dice cuántas iteraciones se van a hacer y la i es una variable que indica el paso del conteo, en otras palabras, nos dice de cuanto en cuanto vamos contando hasta llegar a N.

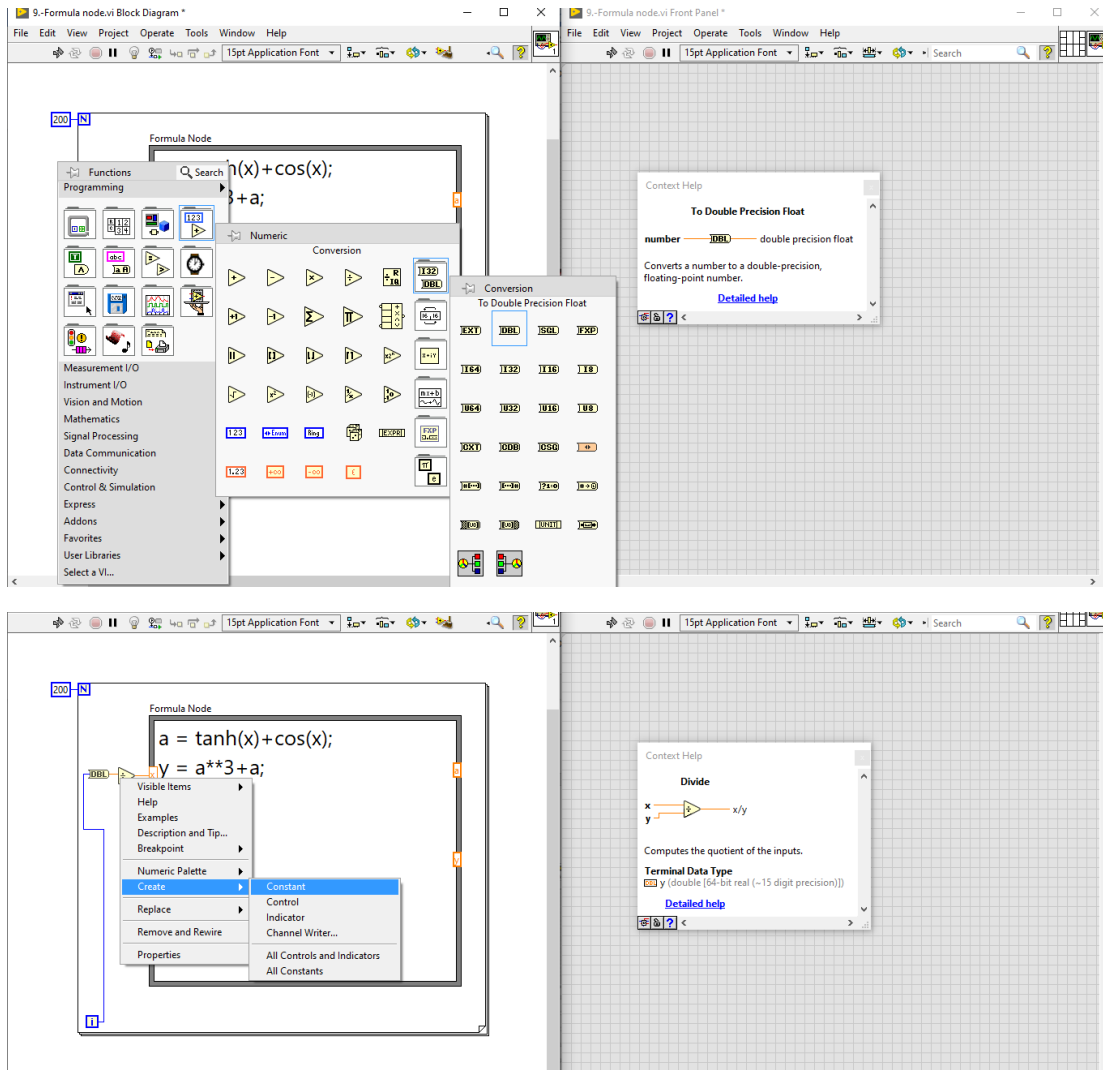


## Block Diagram - Divide: División de dos Números Cualquiera

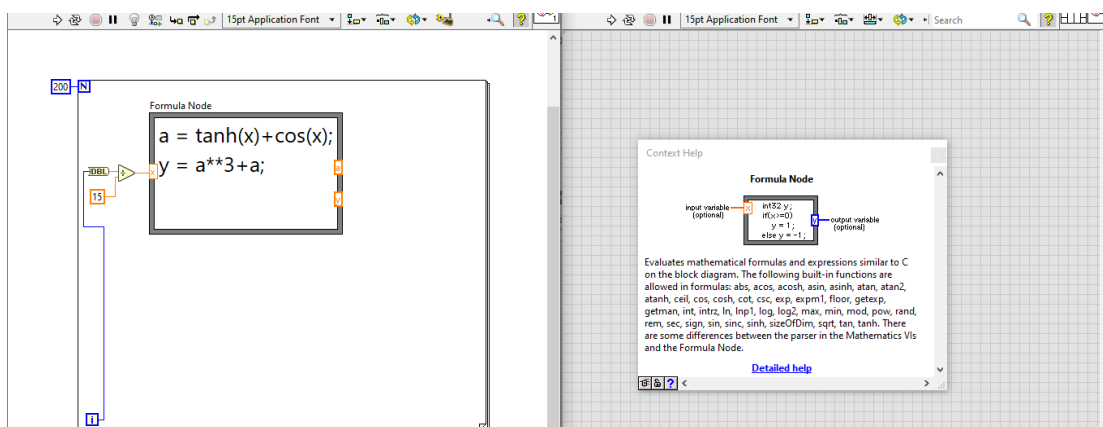
En este caso se utilizará la variable i del bucle for para indicar el paso dx de la gráfica tangente creada con el Formula Node, para ello cada uno de sus valores se dividirá entre 15 para que sea un valor decimal muy pequeñito y la señal tangente tenga una buena definición de datos.

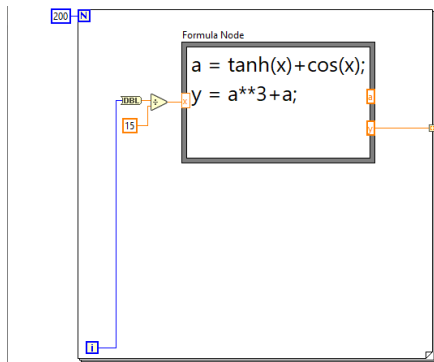
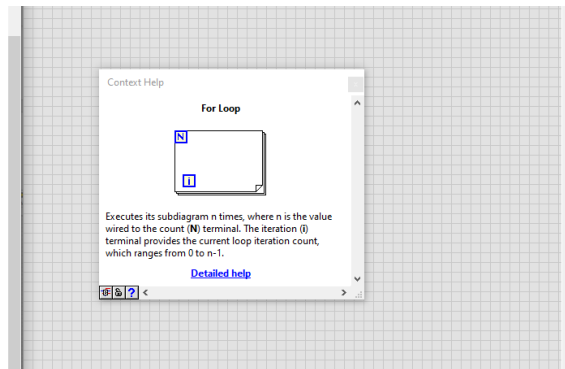
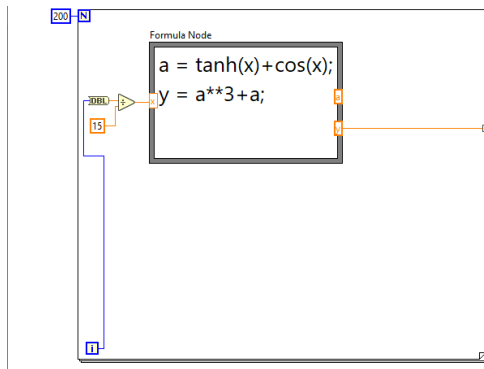


**Cambiar el tipo de dato de mi elemento:** Clic derecho en el bloque → Representation → Tipo de Dato.  
Como cree primero la constante, debo cambiar el tipo de dato para evitar que se cree un coercion dot.



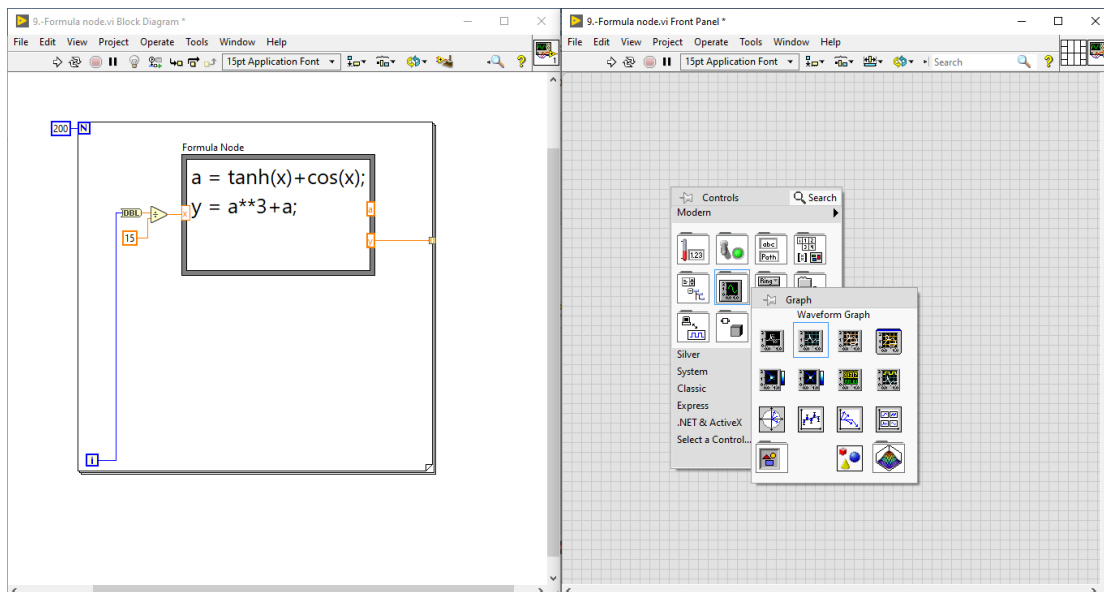
Este 15 es porque quiero generar un valor decimal pequeño cualquiera para que fuera la variable x.





## Front Panel - Waveform Graph: Ventana que Muestra una Señal (Array)

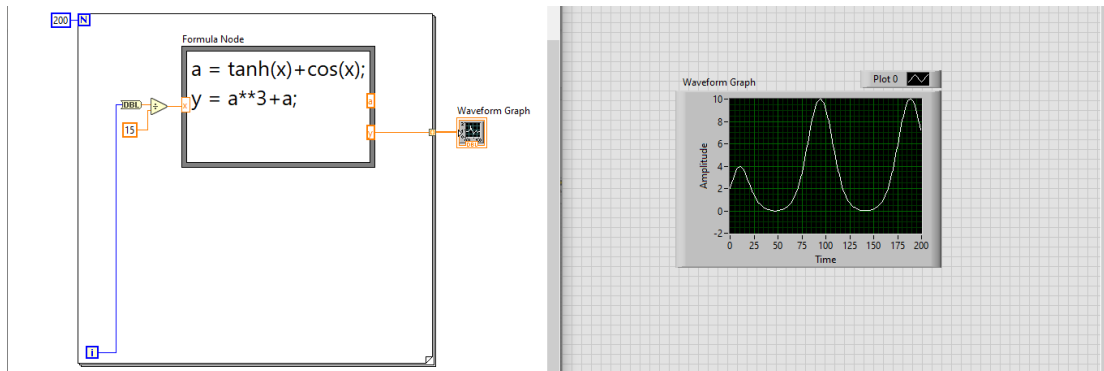
El Waveform Graph muestra gráficas de tipo Array, las cuales son cualquier tipo de grupos de números.



Del túnel de salida saldrá un vector que se irá al bloque del Waveform Graph.

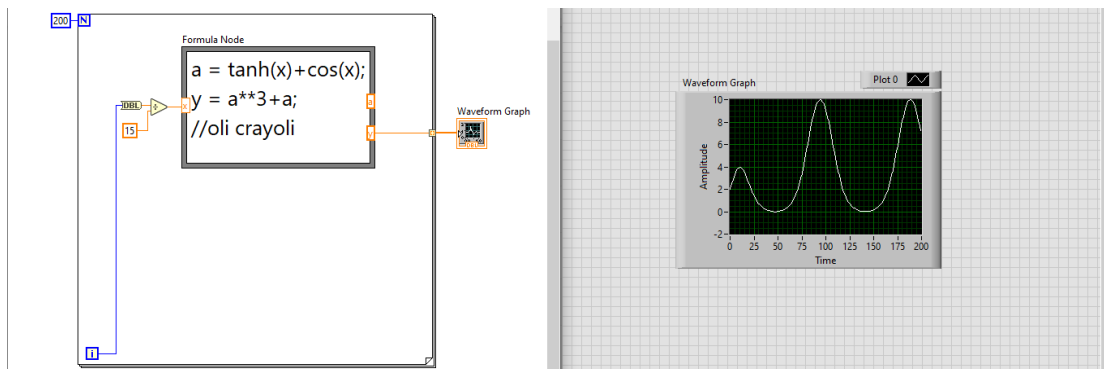
Pongo un Waveform Graph porque no es necesario que se esté graficando el mismo vector dinámicamente, sino que cada que le dé STOP al programa, se graficará como fue cambiando el valor del vector.





La gráfica no se mostrará hasta que deje de correr el programa.

**Añadir comentarios al Formula Node:** Se puede añadir comentarios anteponiendo dos diagonales como se ve en el VI.



## Ejecución del Programa: Waveform Graph (Arrays) de un Formula Node

La gráfica no se actualizará hasta que detenga el programa y debo borrar su historial manualmente, además su ejecución se da en un instante, no se ejecuta de forma continua porque no es necesario y además no existe un bucle while que lo mantenga corriendo, por lo cual cuando se termina la ejecución del bucle for, que cuenta de 0 a 200, el programa se deja de ejecutar y en ese momento muestra la gráfica obtenida en el Waveform Graph (Arrays).

