

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

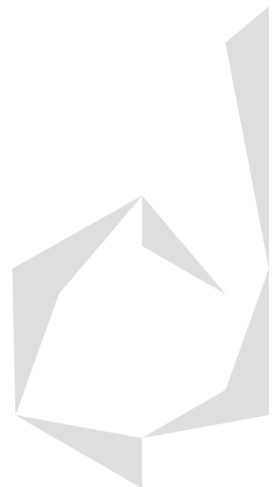
INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Gráfica XY

Contenido

Introducción Teórica de LabVIEW:.....	2
Introducción al Entorno de LabVIEW:.....	2
Front Panel: Ventana Gris con la Interfaz del Programa	4
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)	4
Front Panel o Block Diagram - Show Context Help: Descripción de Bloques	5
Front Panel y Block Diagram: Navegar de una Ventana a Otra	6
Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel	7
Block Diagram - Highlight Execution: Correr Más Lento el Programa.....	8
Coertion dot: Conversión Automática de Datos por Parte de LabVIEW	8
Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI	8
Programa: Gráfica XY	9
Desarrollo del Programa: Identidad Trigonométrica con Gráfica XY	9
Block Diagram - Bucle For: Iteraciones Finitas para la Graficación de una Señal	9
Block Diagram - Sine & Cosine: Función Matemática Seno y Coseno.....	9
Block Diagram - Multiply: Multiplicación de dos Números Cualquiera	10
Block Diagram - Multiply: Multiplicación de dos Números Cualquiera	11
Block Diagram - Bucle For - Túnel (index): Sacar Información del Bucle en un Vector.....	11
Block Diagram - Bundle: Juntar Varios Tipos de Datos Para Mandarlos a un Cluster.....	13
Front Panel - XY Graph y Express XY Graph: Gráfica XY de dos Funciones Matemáticas	13
Block Diagram - Bucle While: Ejecución Continua del Programa.....	16
Ejecución del Programa: Graficación XY de Seno/Coseno Controlado con Perilla	17



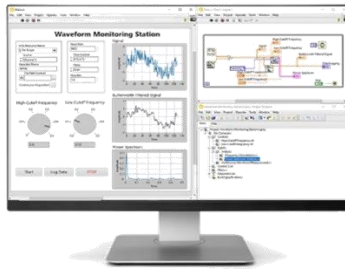
Introducción Teórica de LabVIEW:

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

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

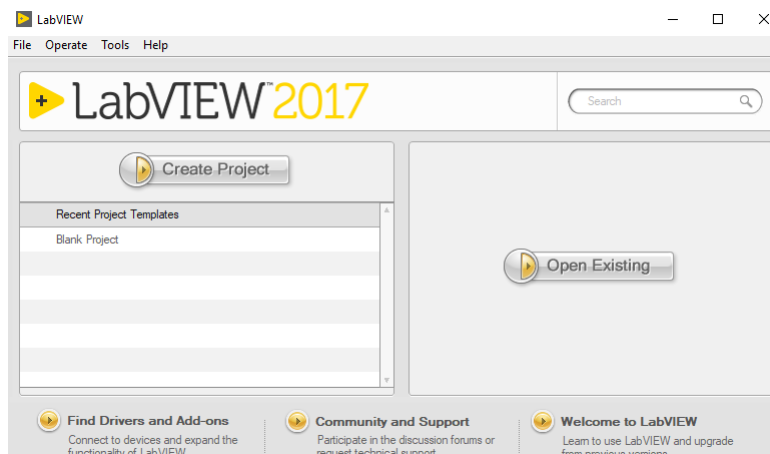
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

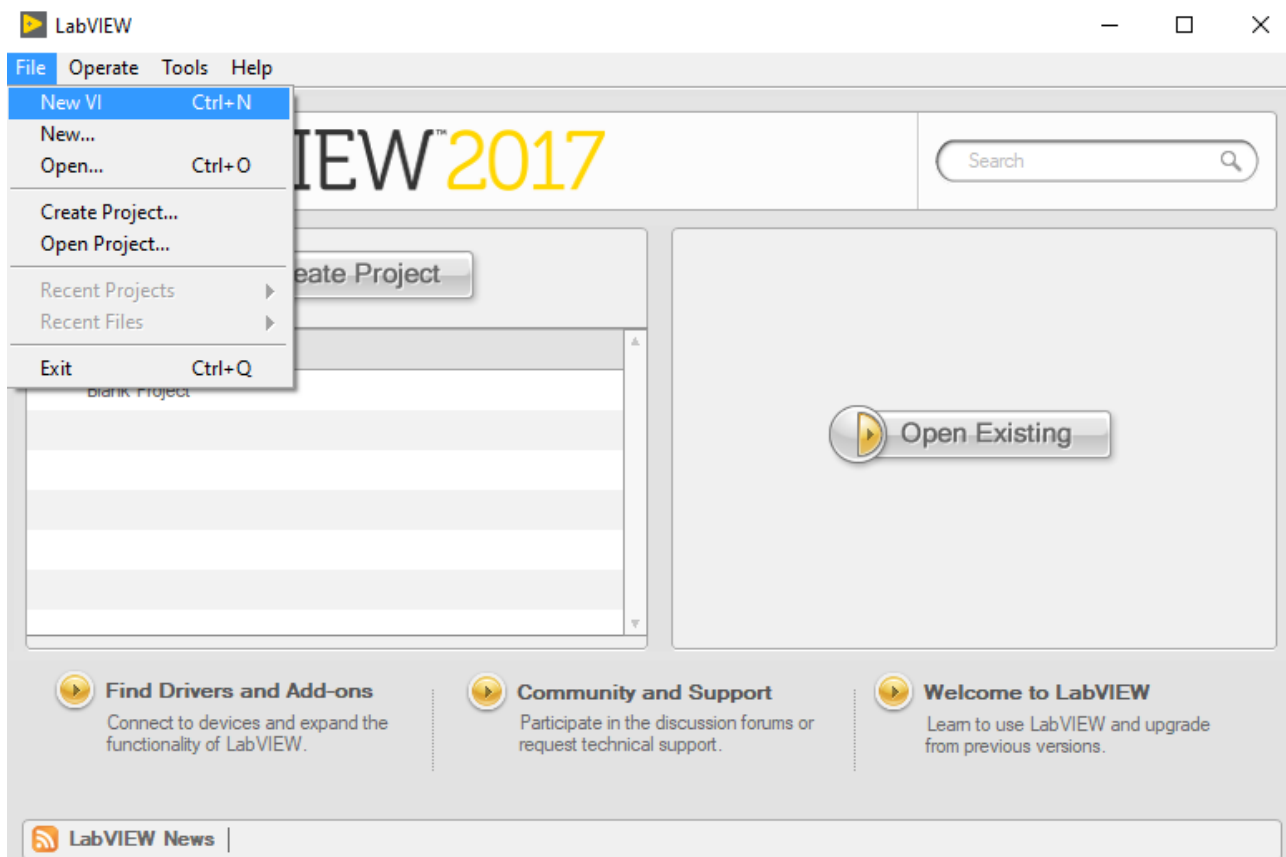


Introducción al Entorno de LabVIEW:

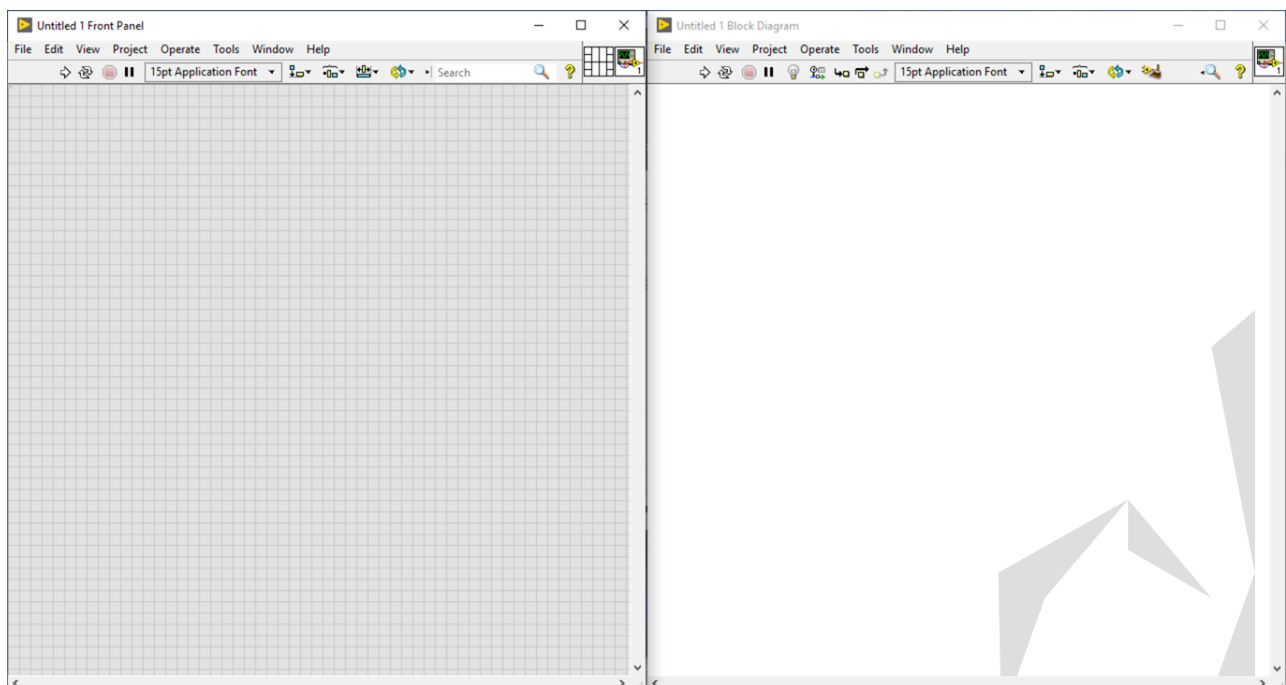
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

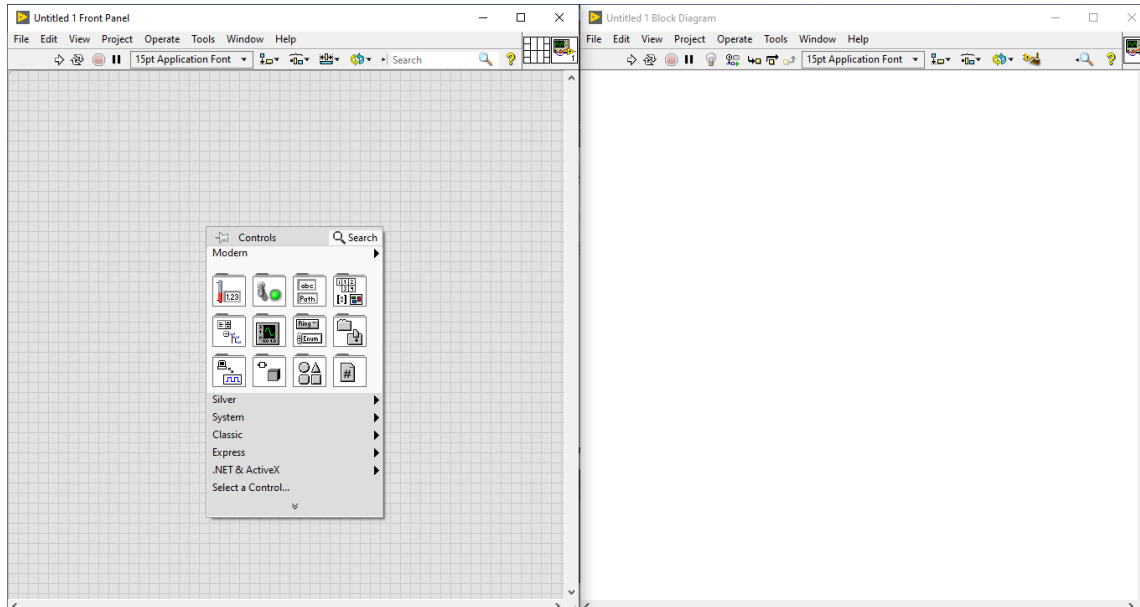


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



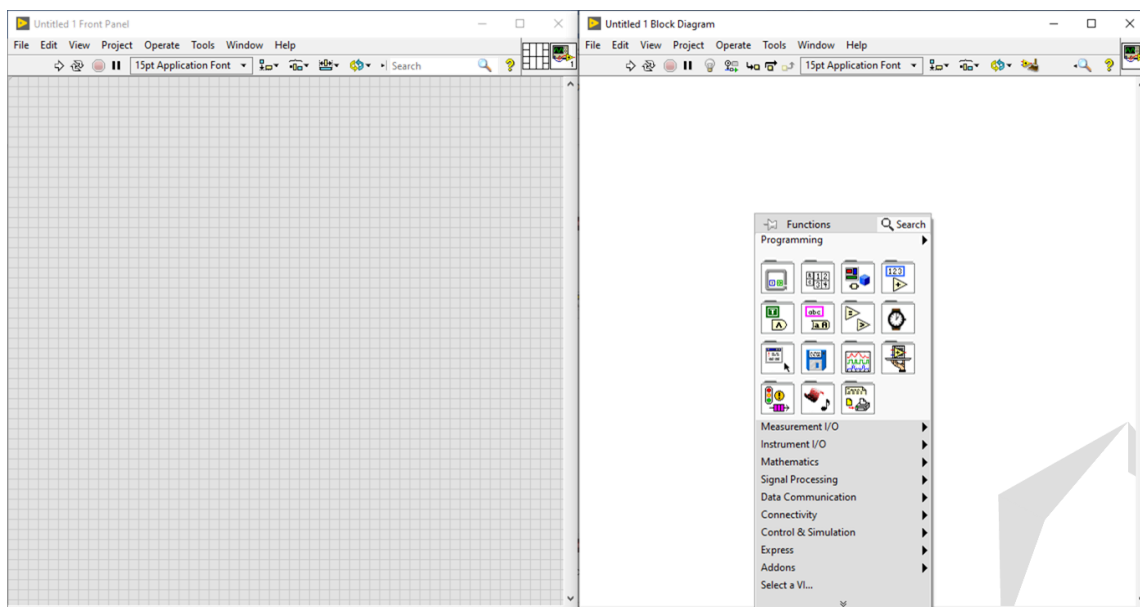
Front Panel: Ventana Gris con la Interfaz del Programa

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



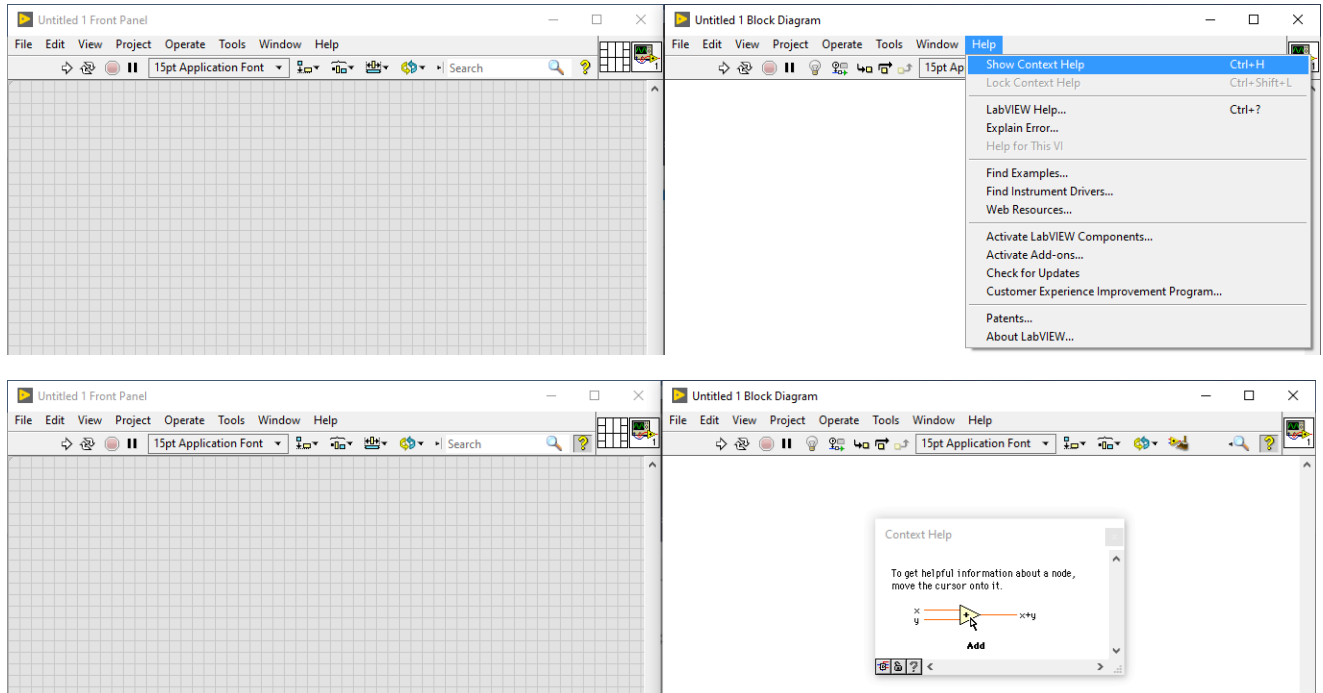
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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



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

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



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

VIs Express, VIs y Funciones

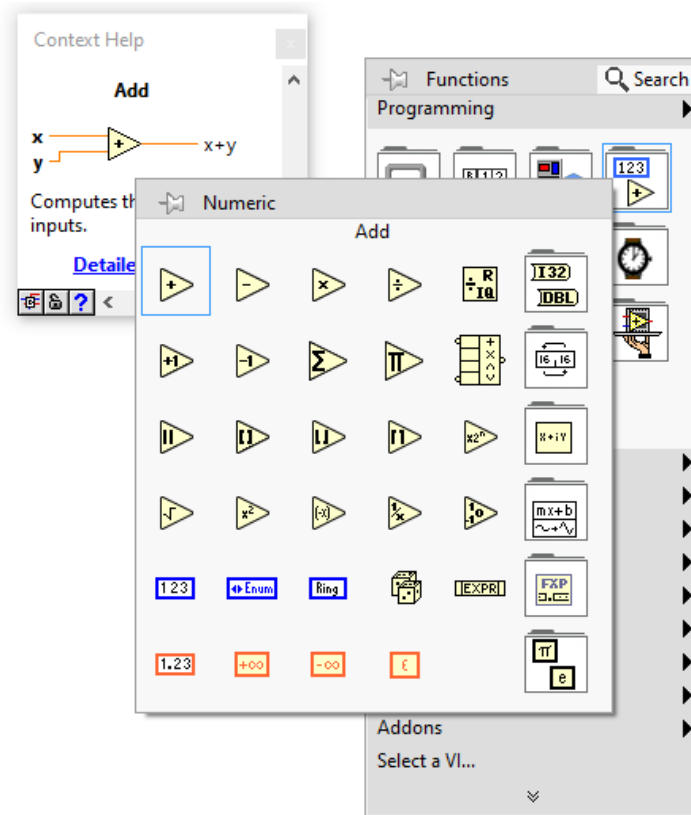


- **VIs Expreso:** VIs interactivos con pagina de dialogo configurable
- **VIs estándar:** VIs modulares y personalizables mediante cableado
- **Funciones:** Elementos fundamentales de operación de LabVIEW; no contiene panel frontal o diagrama de bloque



Función

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

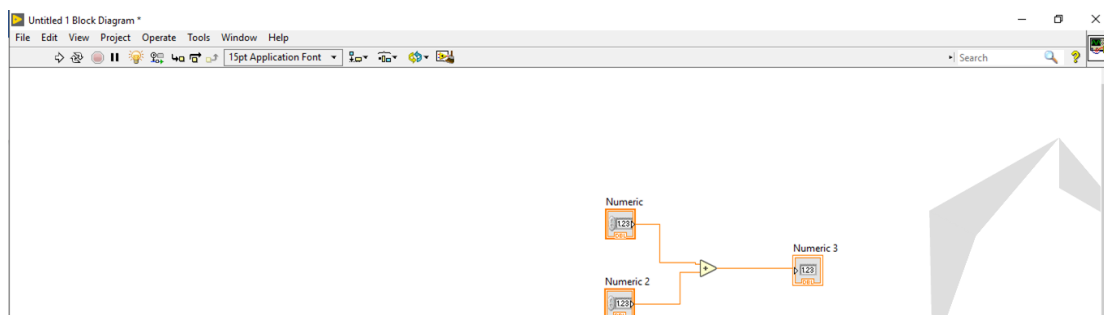


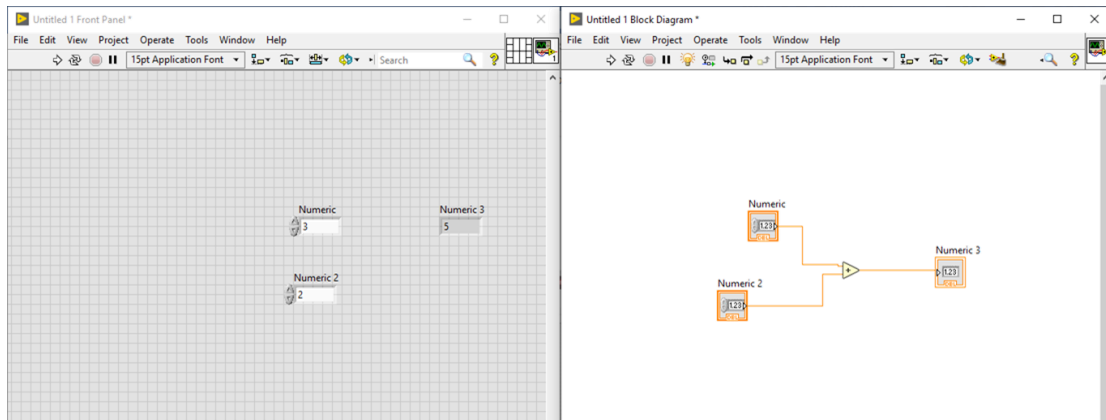
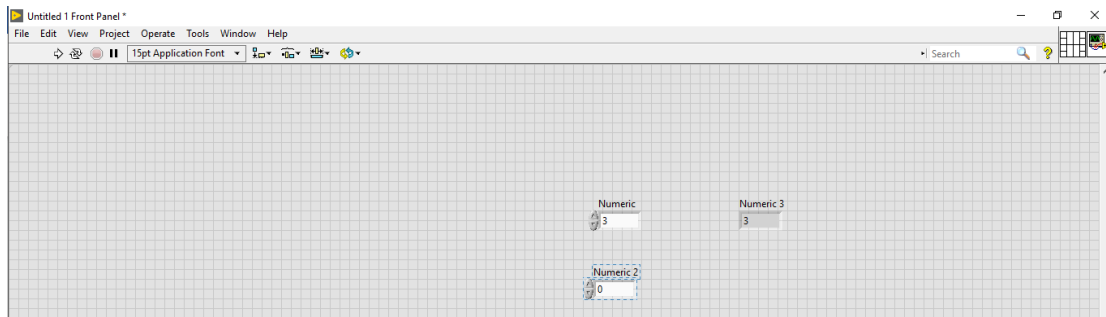
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

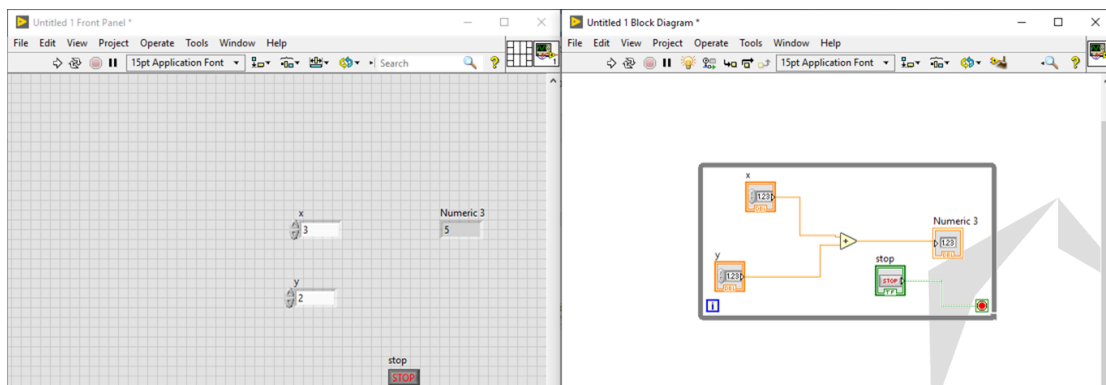
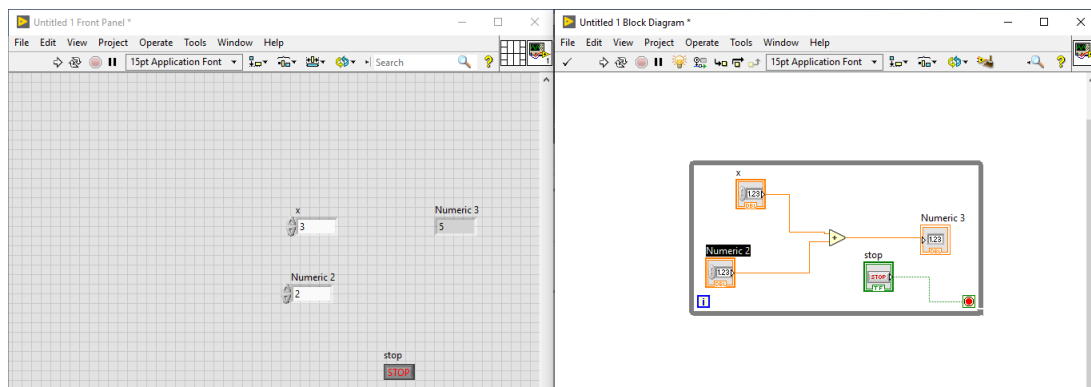
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

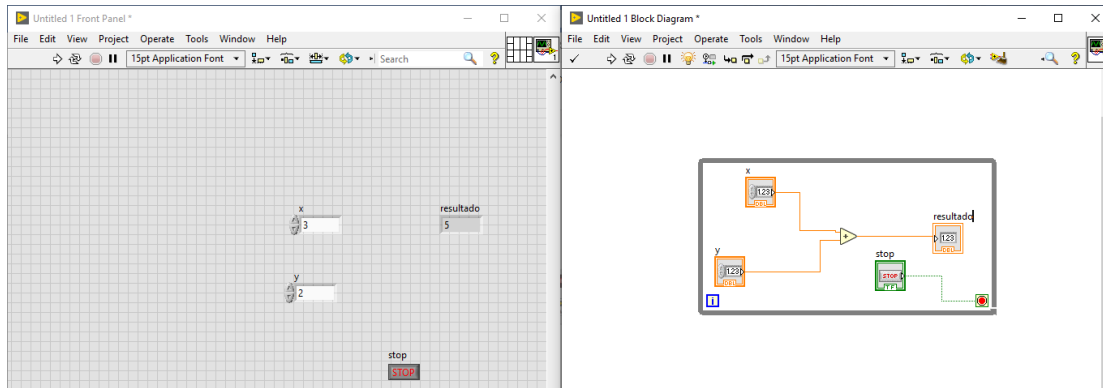




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

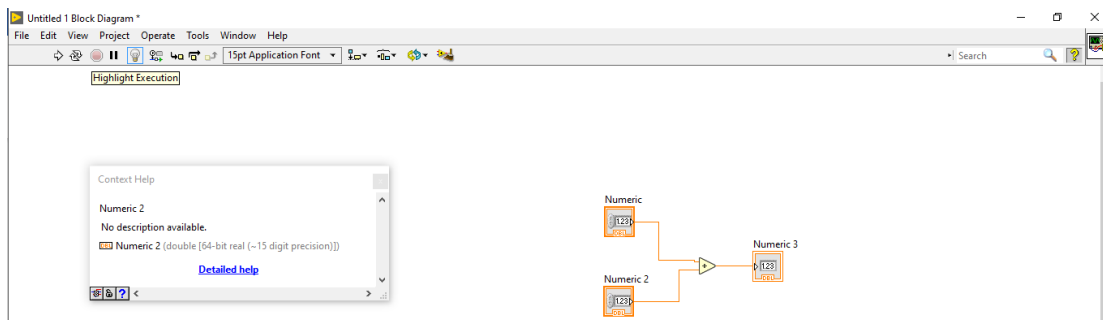
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





Block Diagram - Highlight Execution: Correr Más Lento el Programa

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

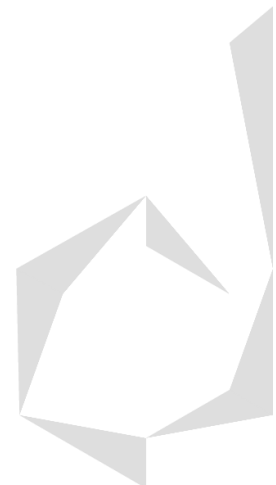


Coertion dot: Conversión Automática de Datos por Parte de LabVIEW

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

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

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



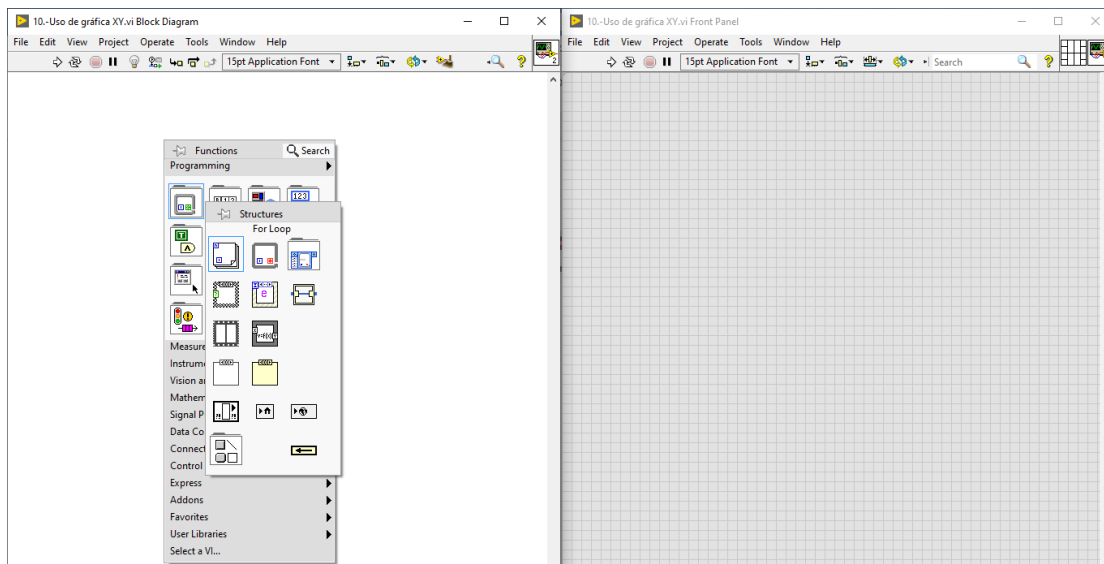
Programa: Gráfica XY

Vamos a graficar la identidad trigonométrica del seno/coseno con el bloque de graficación XY.

Desarrollo del Programa: Identidad Trigonométrica con Gráfica XY

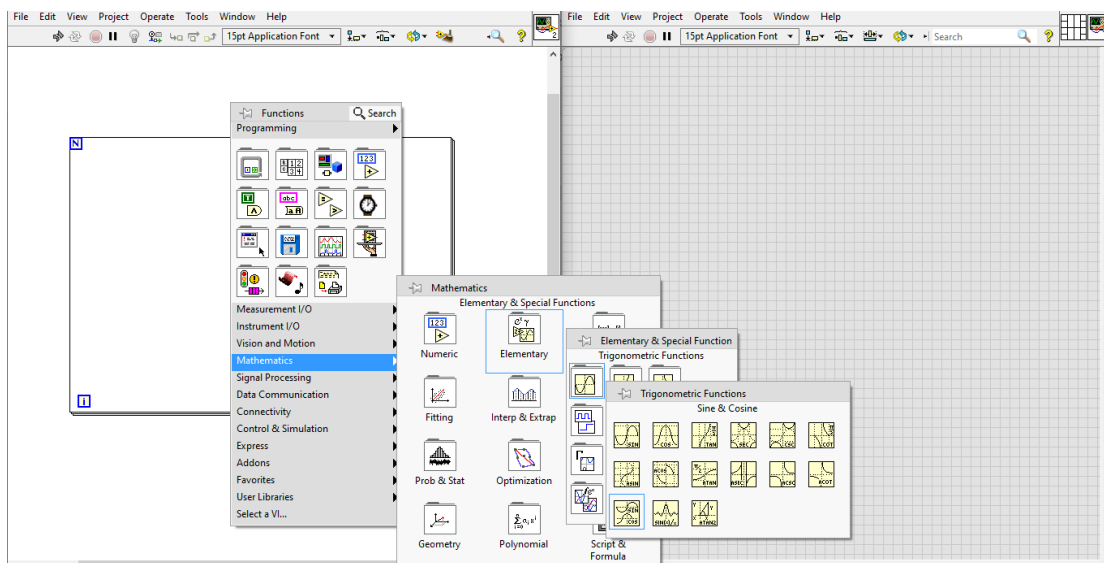
Block Diagram - Bucle For: Iteraciones Finitas para la Graficación de una Señal

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.

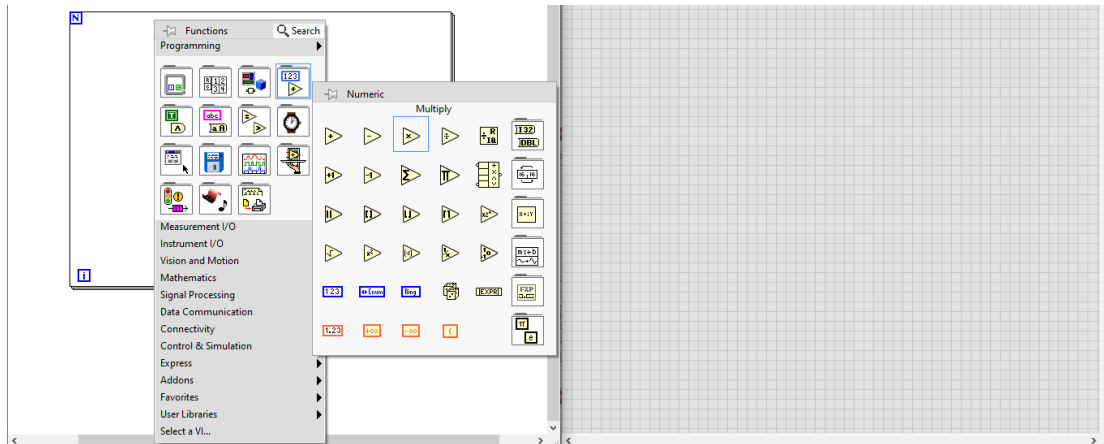


Block Diagram - Sine & Cosine: Función Matemática Seno y Coseno

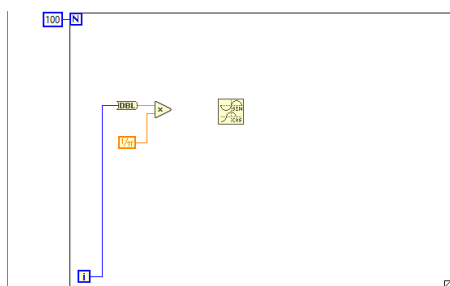
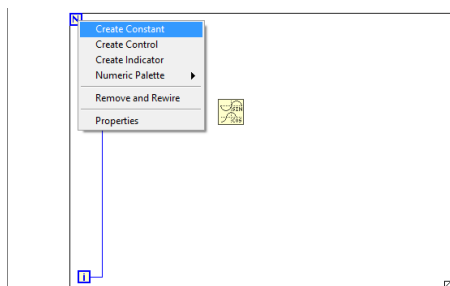
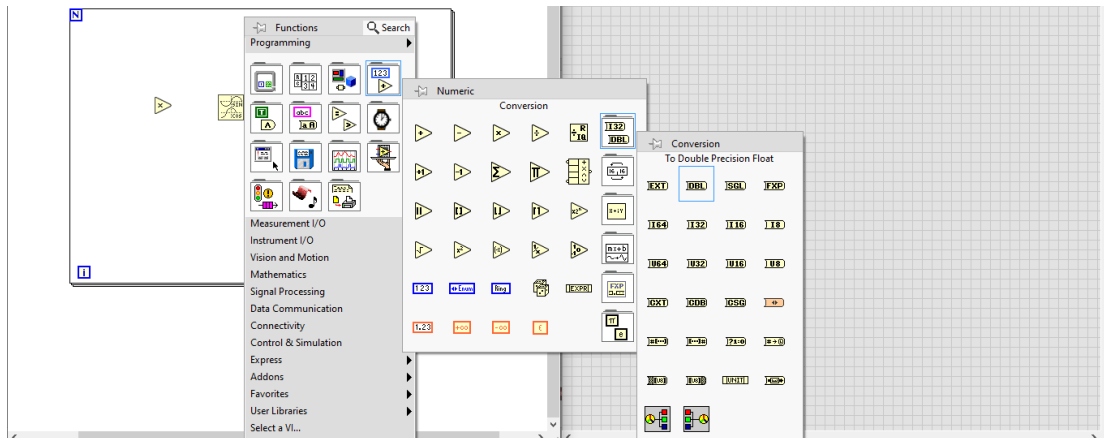
De la terminal superior se obtiene la salida de una función seno y de la inferior una función coseno.

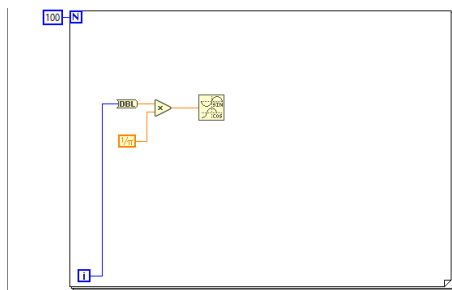


Block Diagram - Multiply: Multiplicación de dos Números Cualquiera



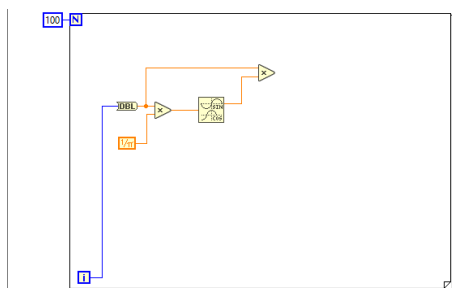
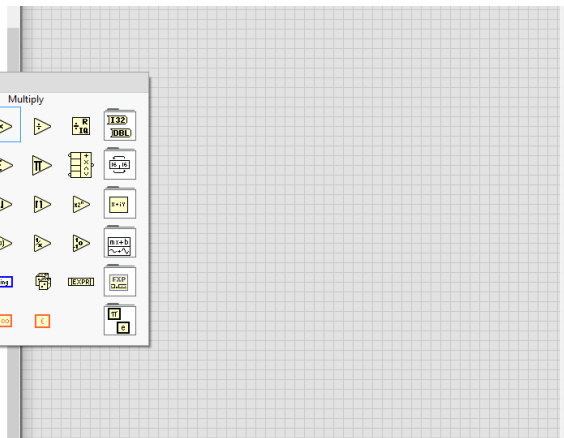
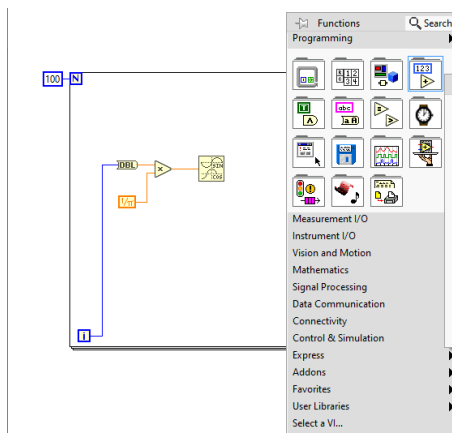
Cambiar el tipo de dato de mi elemento: Clic derecho en el bloque → Representation → Tipo de Dato.
Como cree primero la constante, debo cambiar el tipo de dato para evitar que se cree un coercion dot.





Block Diagram - Multiply: Multiplicación de dos Números Cualquiera

Se realizan las multiplicaciones para alimentar la función trigonométrica de seno/coseno.

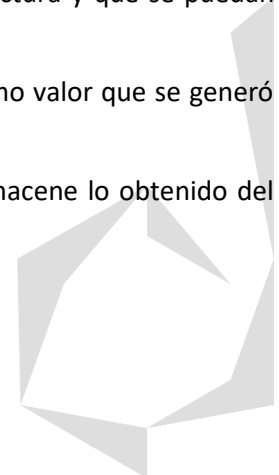


Block Diagram - Bucle For - Túnel (index): Sacar Información del Bucle en un Vector

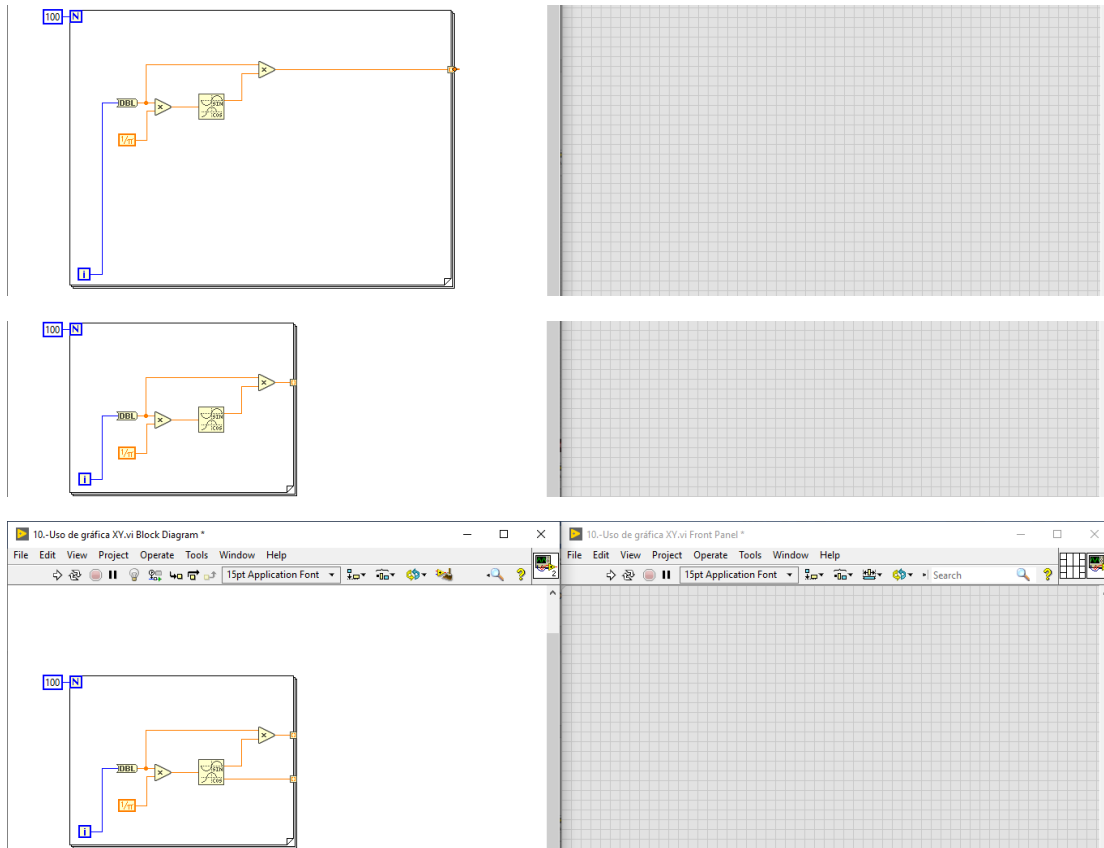
Un túnel es una parte de la pared del bucle que sirve sacar valores de su estructura y que se puedan utilizar afuera (en las demás partes del programa).

Por medio de un túnel del bucle for puedo obtener un escalar que tenga el último valor que se generó del bloque de función seno/coseno para sacarlo y mostrarlo.

Con la opción de indexing aplicada al túnel, estoy generando un vector que almacene lo obtenido del bucle for.

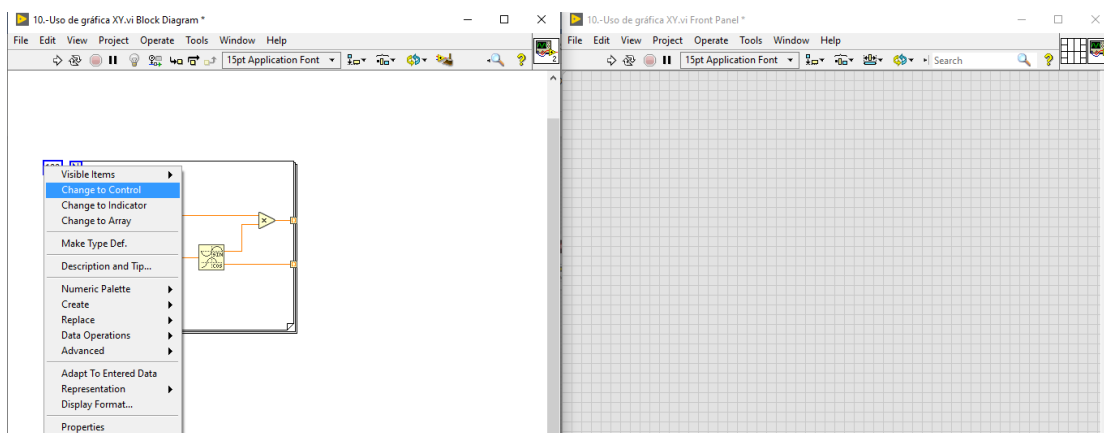


El túnel del ciclo for lo que hará es aparecer indexado, esto implica que lo que vamos a estar generando aquí es un vector, por lo tanto, los datos se van a estar almacenando en un vector que podemos obtener de ese punto.



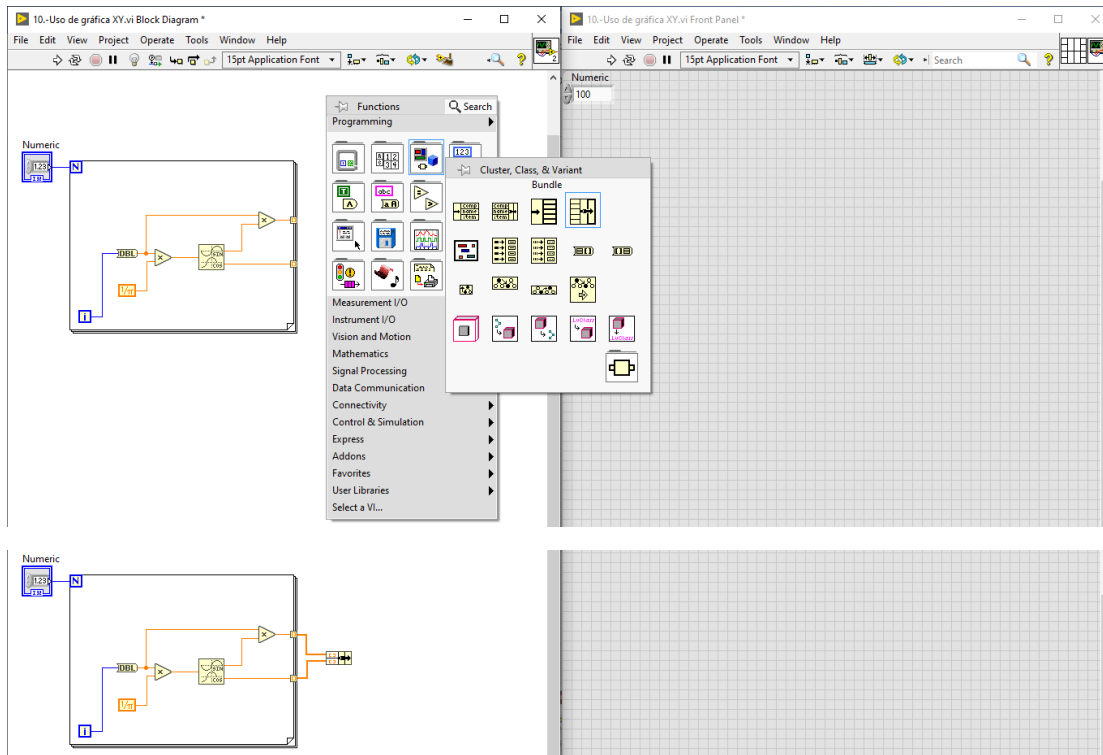
Las salidas indexadas, que son los túneles que salen del ciclo for, tienen vectores con el resultado de toda la función matemática, ya sea si se toma la terminal del seno o del coseno.

Cambiar una Constante por un Control: Clic derecho en la constante de interés → Change to Control. Vamos a cambiar la constante que define las veces que se va a correr el bucle for por un control.



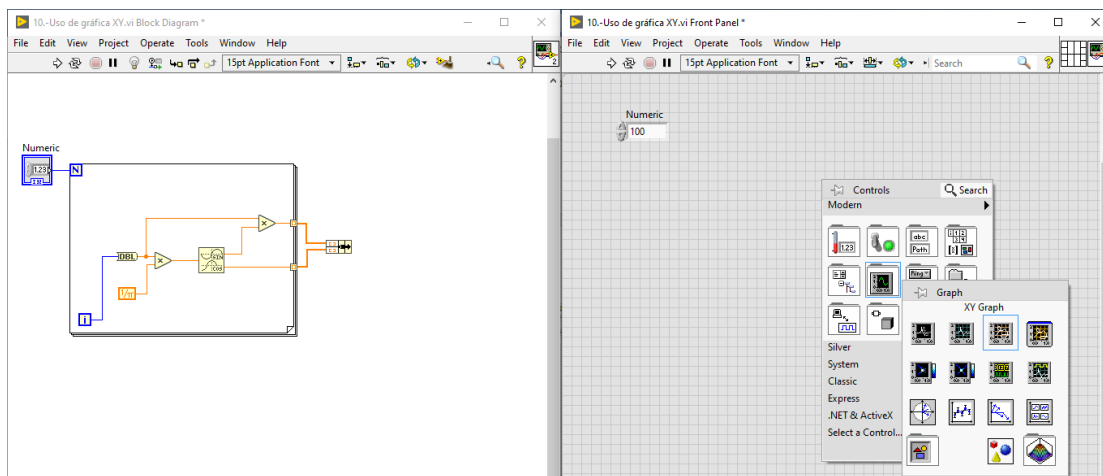
Block Diagram - Bundle: Juntar Varios Tipos de Datos Para Mandarlos a un Cluster

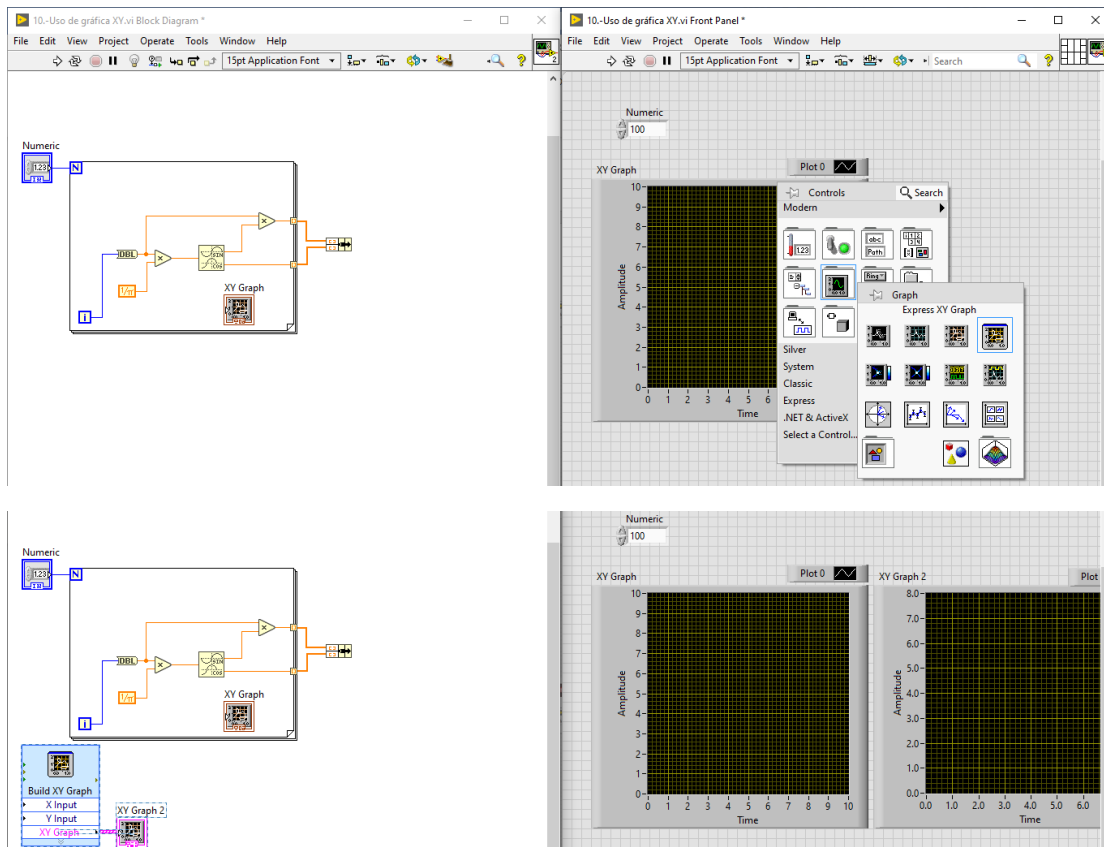
El tipo de dato Cluster es perteneciente únicamente al entorno de desarrollo de LabVIEW y representa un tipo de dato definido por el usuario que recibe y encapsula varios, pero para poder realizar esto se debe incluir un bloque intermedio llamado Bundle, que se encarga de juntar todos los tipos de datos distintos o iguales antes de enviarlos al bloque de Cluster.



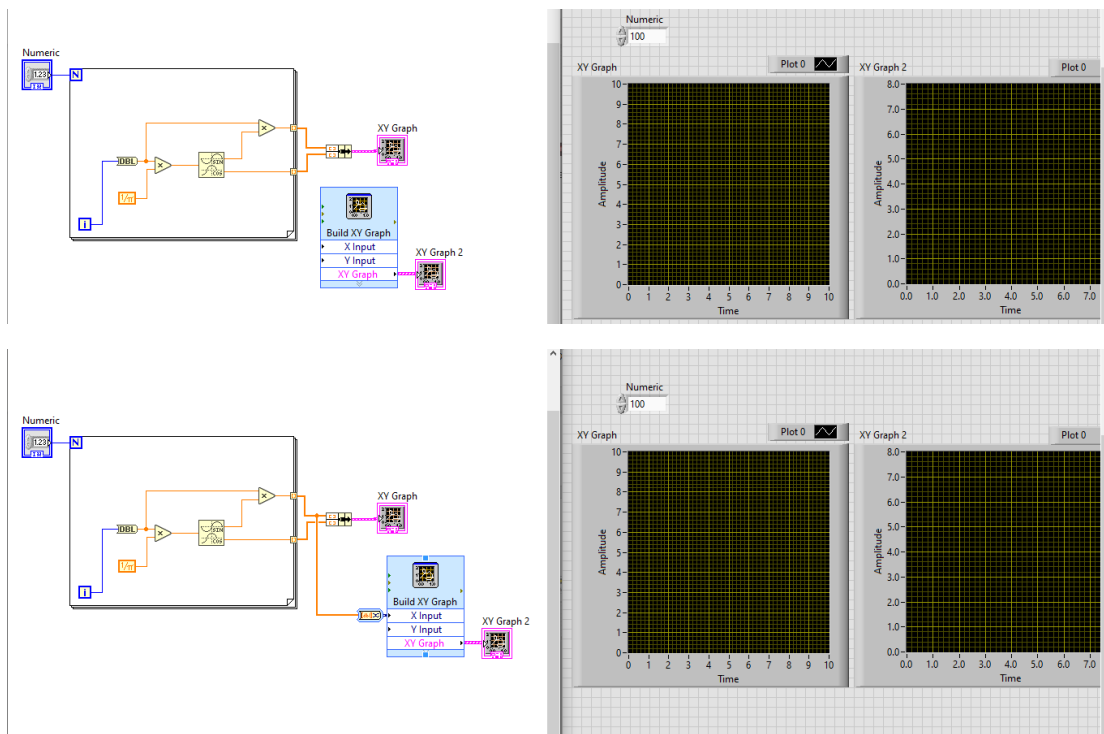
Front Panel - XY Graph y Express XY Graph: Gráfica XY de dos Funciones Matemáticas

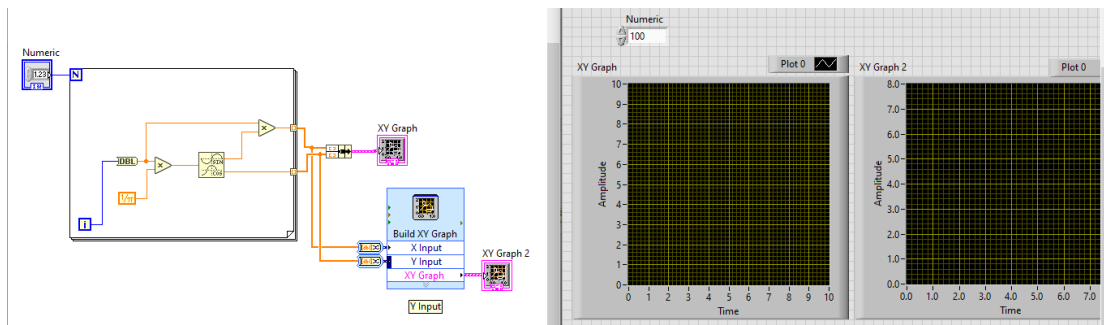
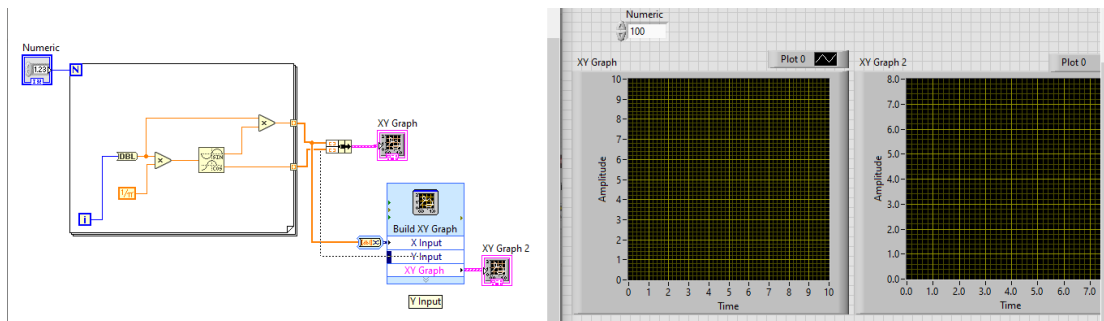
El XY Graph muestra gráficas de tipo Array, las cuales están hechas de dos funciones matemáticas con el mismo número de datos.



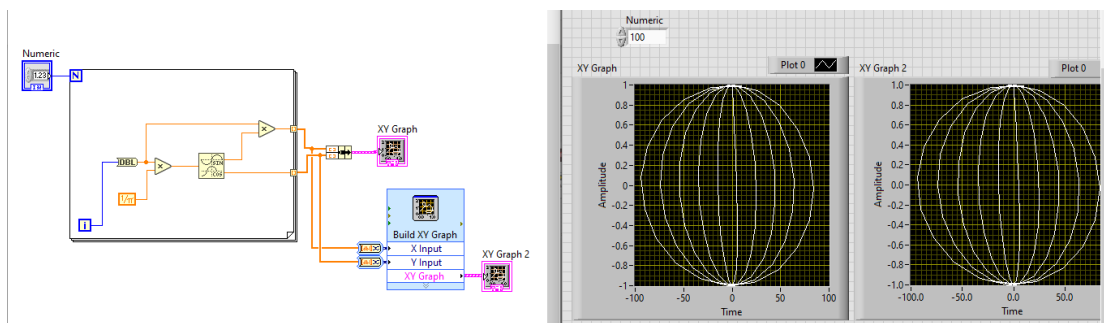


Con la XY Graph debo usar un bundle pero con el Express XY graph no es necesario ya que aparece por sí solo un convertidor que convierte un dato cualquiera en un dato dinámico.

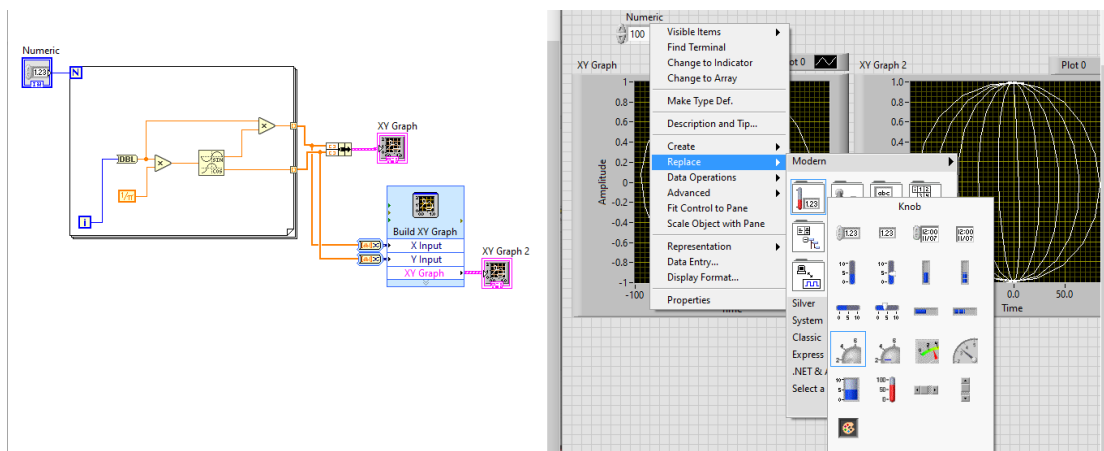


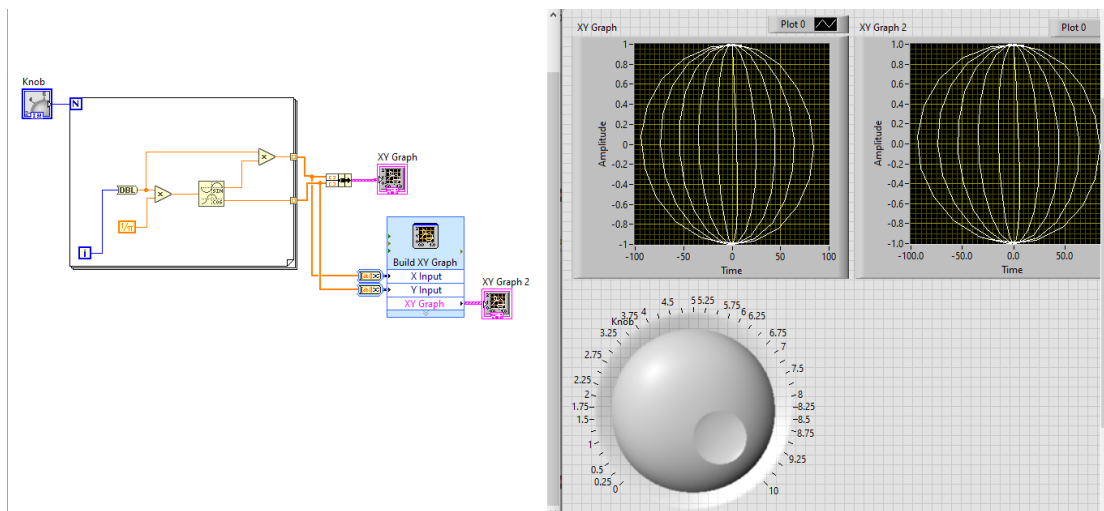


Esa es la única diferencia de los 2 graficadores, el XY Graph necesita un bundle para poder meter los vectores a la gráfica y el Express XY Graph no.



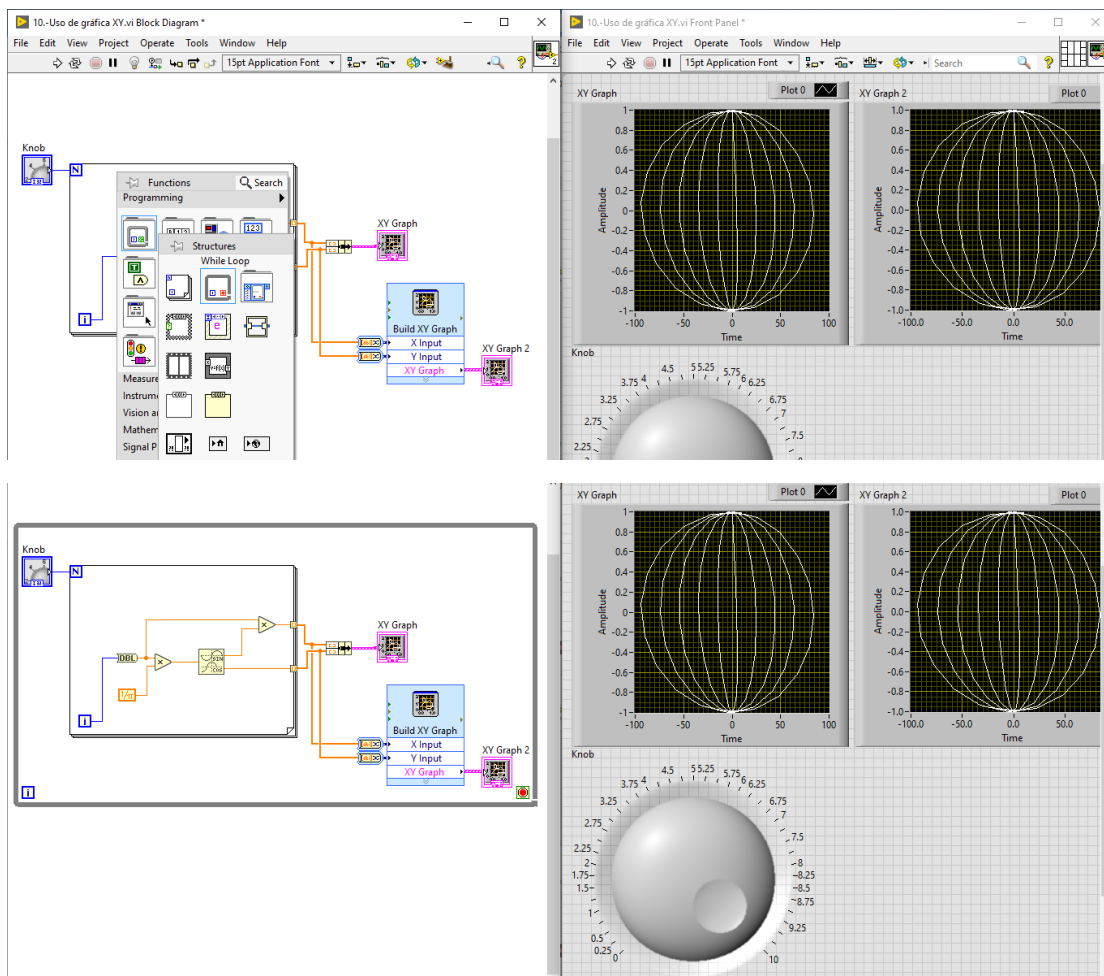
Reemplazar un Control por una Perilla: Clic derecho en el control → Replace → Knob. Podemos cambiar el control por una perilla en el Front Panel.

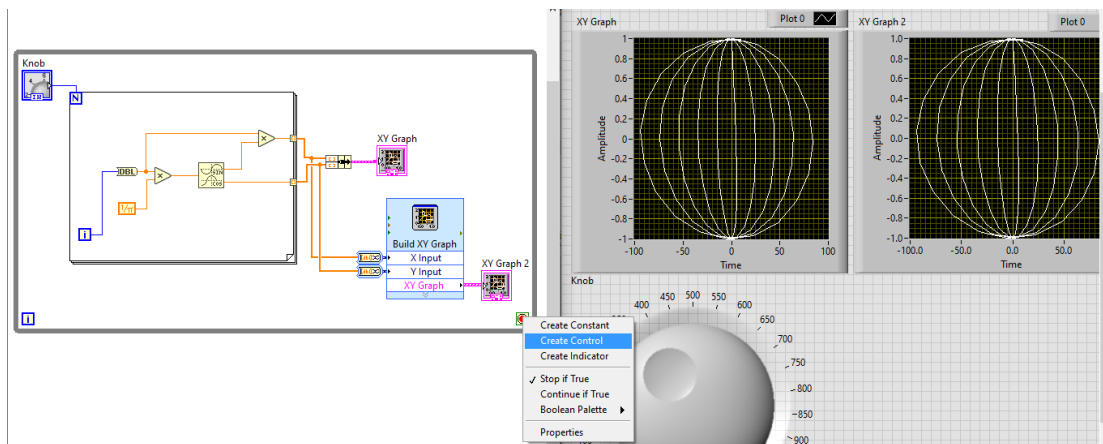




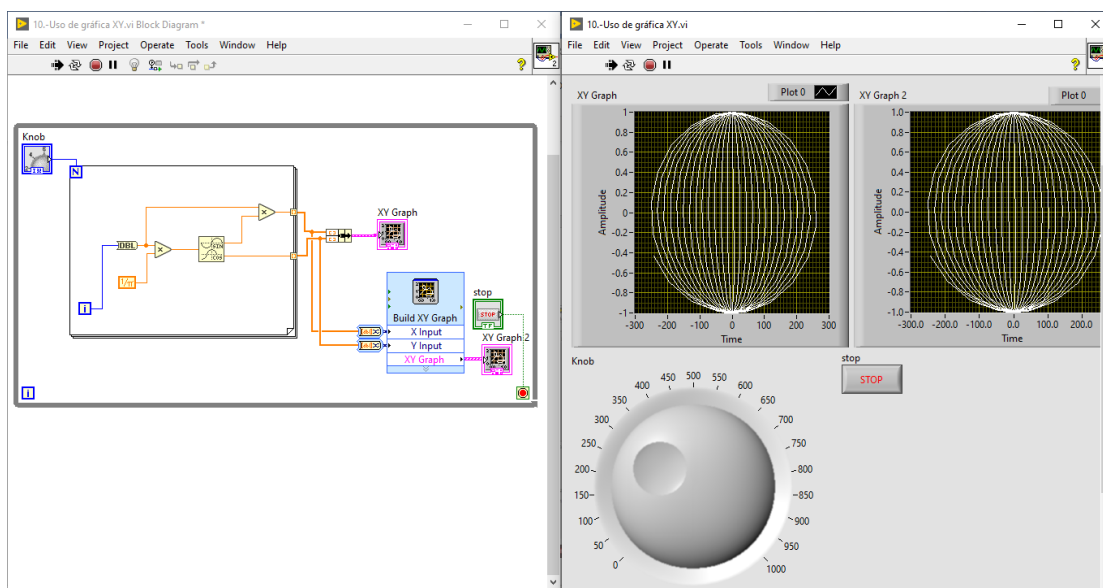
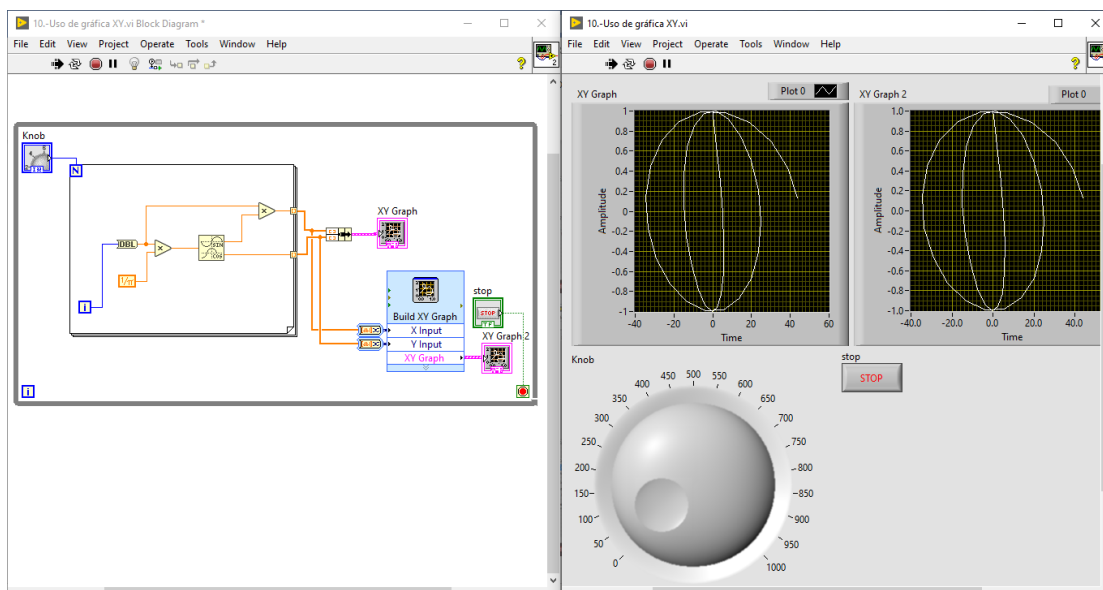
Block Diagram - Bucle While: Ejecución Continua del Programa

El ciclo while hace que el programa se ejecute hasta que dé clic en el botón de STOP, por eso todo el diagrama de bloques que tengo actualmente lo voy a encerrar en un ciclo while para que esté ejecutando de manera continua.





Ejecución del Programa: Graficación XY de Seno/Coseno Controlado con Perilla



La perilla lo que está controlando son las veces que se corren las funciones seno y coseno, por eso es que al subir el número de ejecuciones del bucle for, el círculo se hace más definido.

