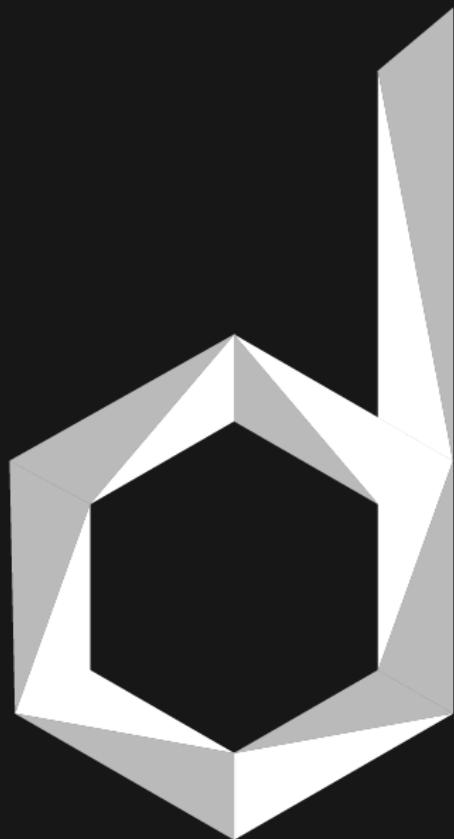


INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Generador de Funciones con
Señal de Prueba con Ruido

Contenido

Introducción Teórica de LabVIEW:.....	3
Introducción al Entorno de LabVIEW:.....	3
Front Panel: Ventana Gris con la Interfaz del Programa	5
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)	5
Front Panel o Block Diagram - Show Context Help: Descripción de Bloques	6
Front Panel y Block Diagram: Navegar de una Ventana a Otra	7
Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel	8
Block Diagram - Highlight Execution: Correr Más Lento el Programa.....	9
Coertion dot: Conversión Automática de Datos por Parte de LabVIEW	9
Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI	9
Programa: Generador de Funciones.....	10
Desarrollo del Programa: Creación de un Simulador de Funciones	10
Block Diagram - Simulate Sig: Creación de un Simulador de Funciones	10
Front Panel: Creación de un Simulador de Funciones.....	14
Front Panel - Waveform Chart: Ventana que Muestra una Señal (Dynamic Data)	14
Block Diagram - Bucle While: Generación de Muestras de la Señal	16
Block Diagram - Bucle While: Botón de Stop Para Detener de forma Manual el Bucle.....	19
Ejecución del Programa: Gráfico de Señal Senoidal con Ruido	20
Block Diagram - FFT Power Spectrum and PSD: Medir la Frecuencia de una Señal	21
Block Diagram - From DDT: Convierte una Señal a un Array, Evitando un Coertion dot	23
Ejecución del Programa: Conversión de Datos Dynamic a Array Exitosa	26
Front Panel - Waveform Graph: Ventana que Muestra una Señal (Array)	26
Cluster: Varios Tipos de Datos Agrupados en Uno	28
Block Diagram - FFT Power...: Dominio de Frecuencia de una Señal (Análisis de Fourier)	29
Ejecución del Programa: Gráfica de la Señal y su Dominio de Frecuencia	30
Block Diagram - Simulate Sig: Agregar un Control para la Frecuencia.....	32
Front Panel - Replace - Knob: Poner una Interfaz de Control en Forma de Perilla	33
Ejecución del Programa: Error al Subir Mucho la Frecuencia de la Señal	35
Ejecución del Programa: Sin Error con Frecuencia ≤ 500 en el Control.....	37
Front Panel - Waveform Chart: Agregar Cursos	38
Block Diagram - Filter: Agregar un Filtro para Quitar el Ruido de la Señal	40

Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI	42
Ejecución del Programa: Gráficas de Frecuencias Con y Sin Ruido Filtrado	44
Ejecución del Programa: Gráficas Con y Sin Filtro con Mayor Muestreo	47
Block Diagram - Simulate Sig: Manejo de Errores Cuando se lee la Señal.....	50
Cluster: Varios Tipos de Datos Agrupados en Uno	53
Block Diagram - Compuerta Lógica OR: Detener el Programa por un Error o con Stop	56
Ejecución del Programa: Gráficas Con Curosres, Con y Sin Filtro y con Manejo de Errores	57



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.

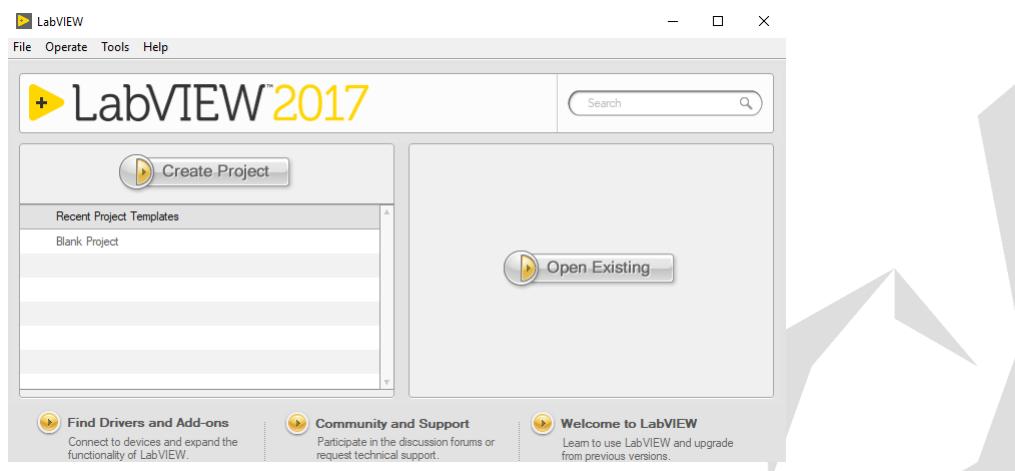
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

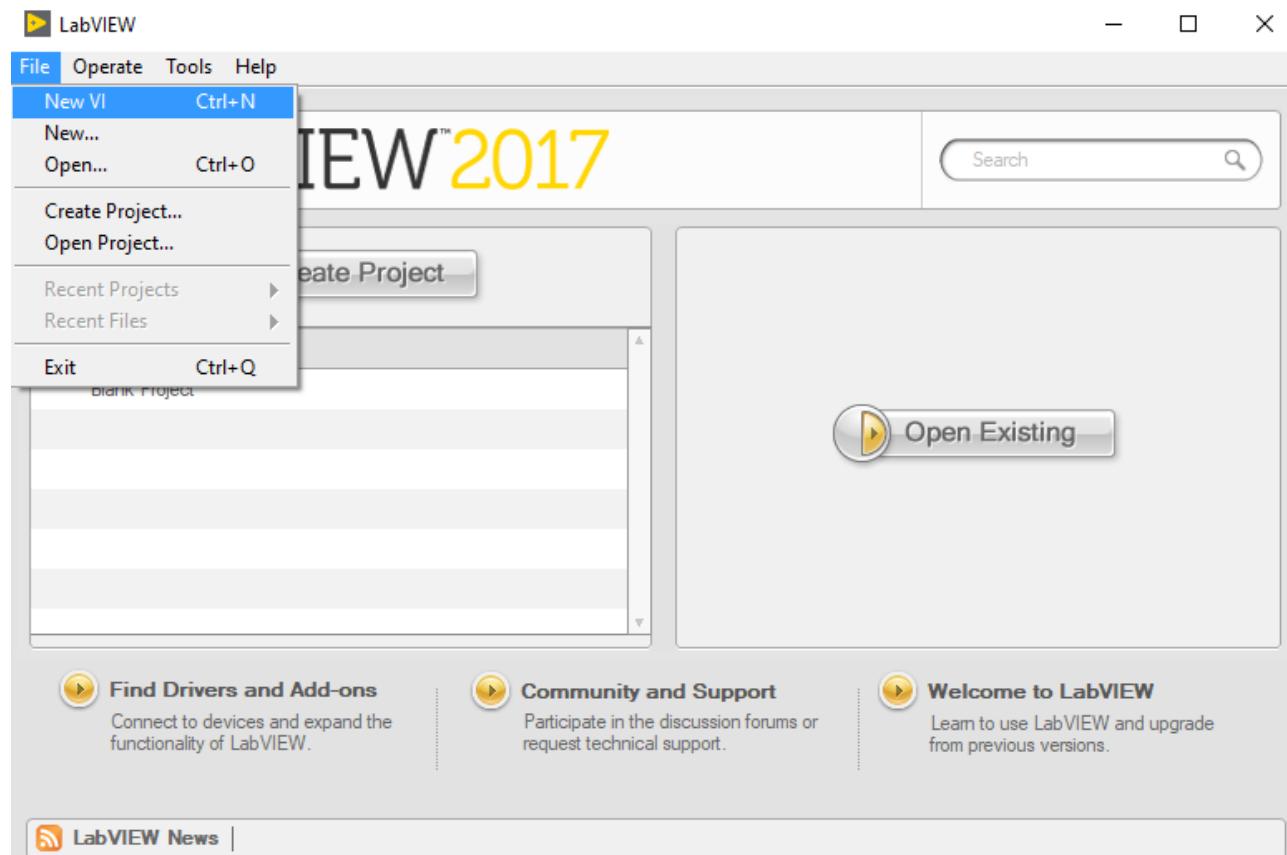


Introducción al Entorno de LabVIEW:

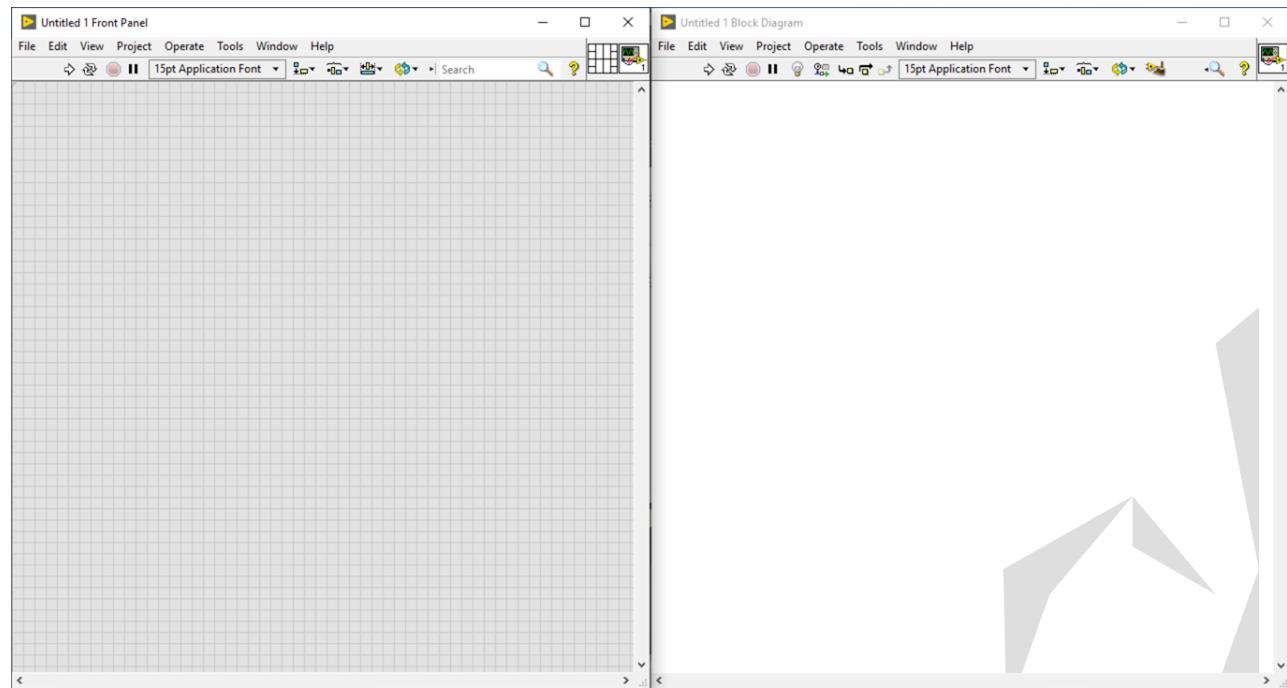
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

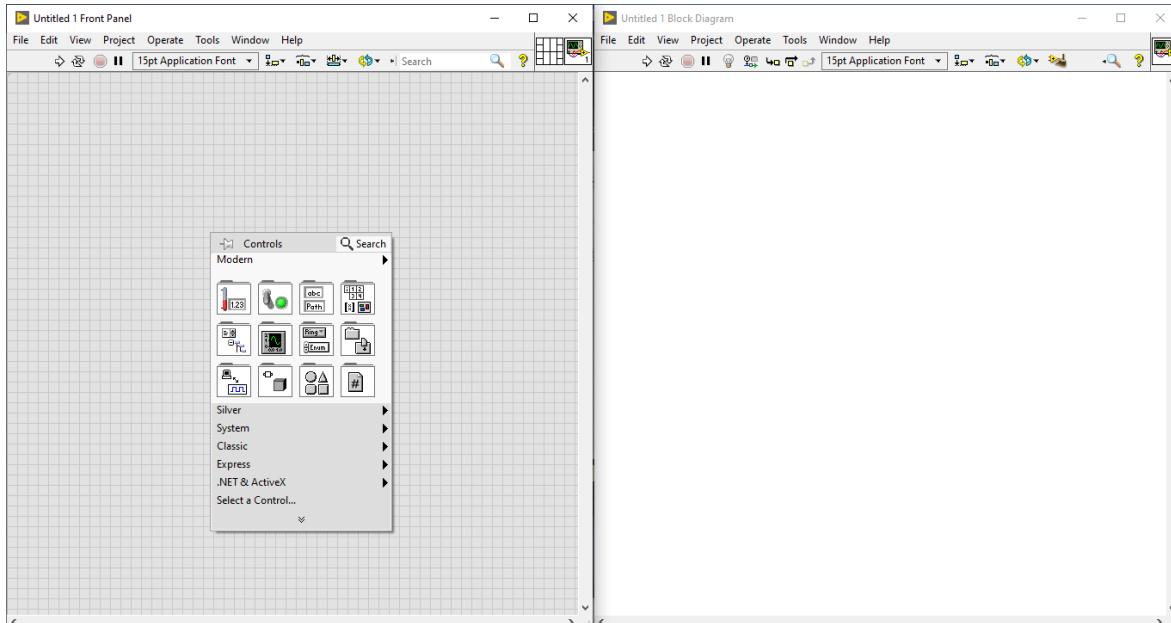


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



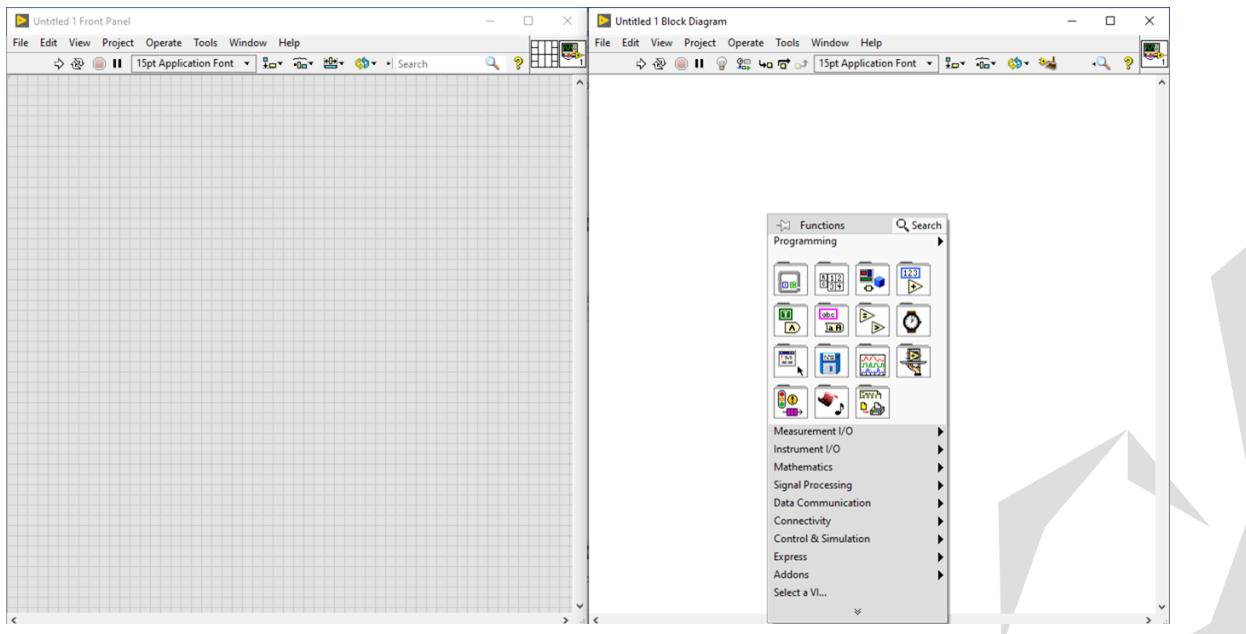
Front Panel: Ventana Gris con la Interfaz del Programa

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.



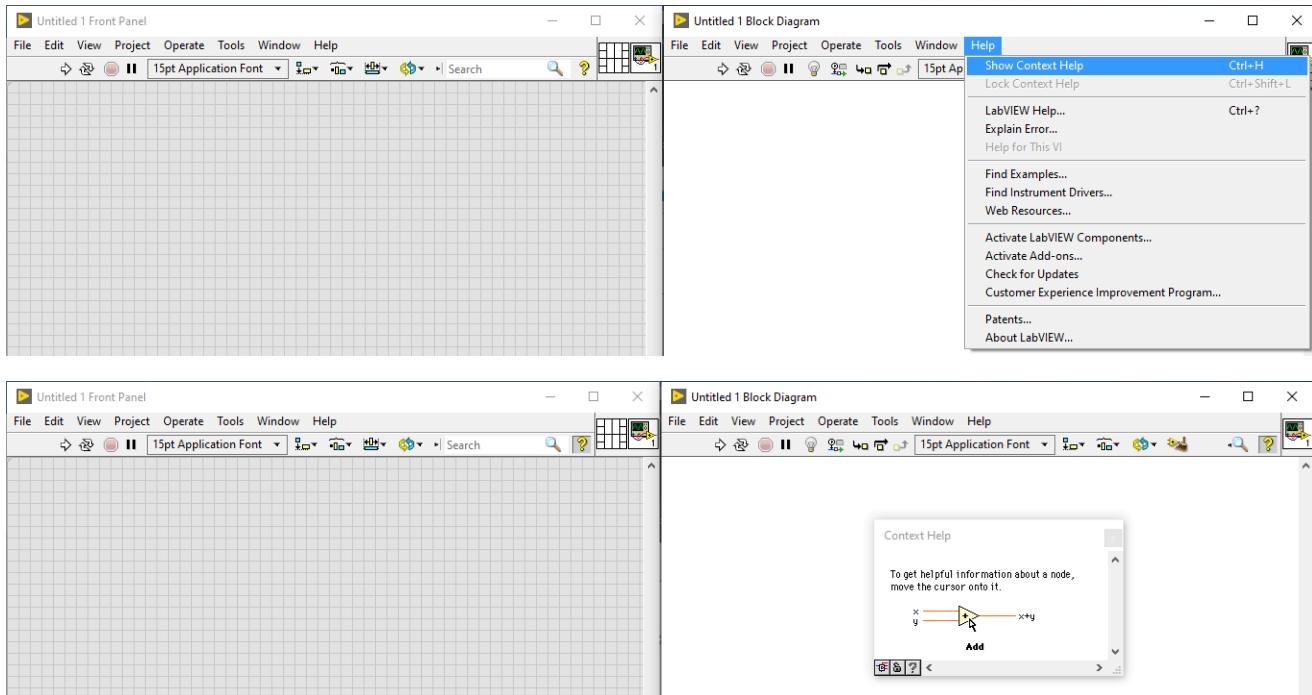
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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.



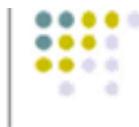
Front Panel o Block Diagram - Show Context Help: Descripción de Bloques

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:

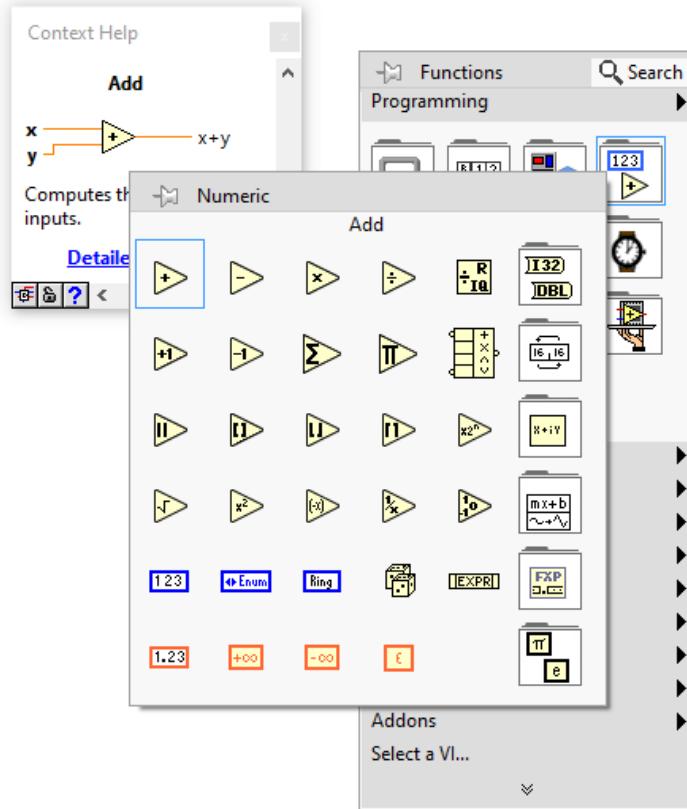
VIs Express, VIs y Funciones



- **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.

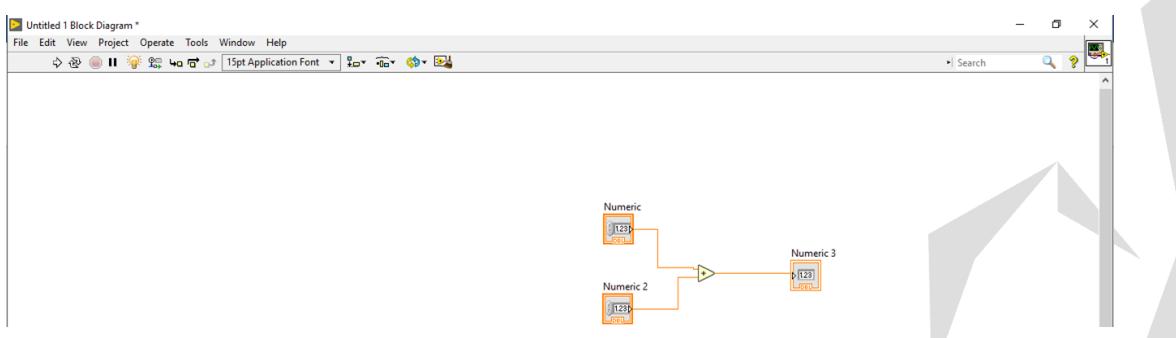


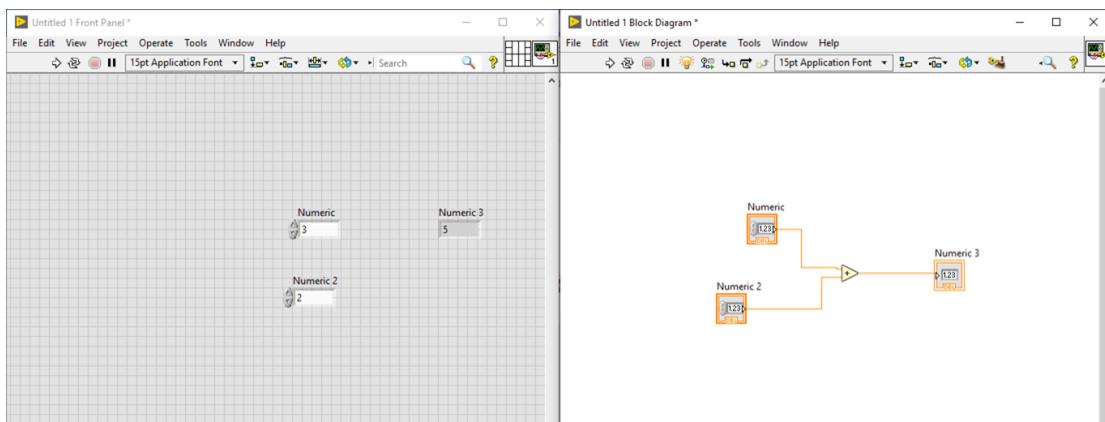
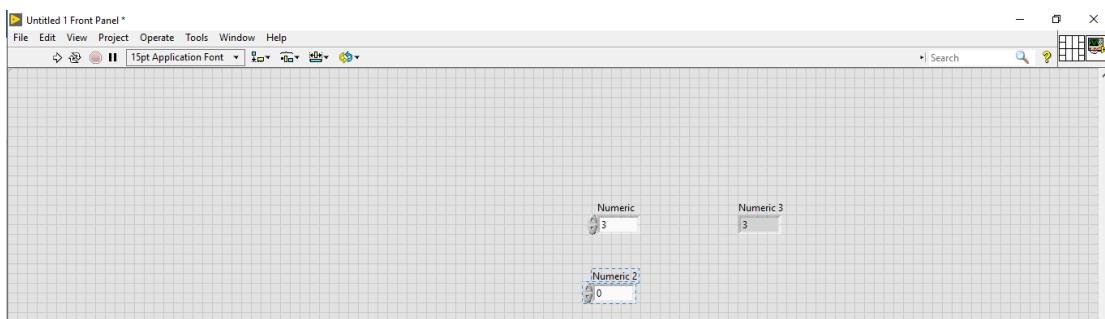
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

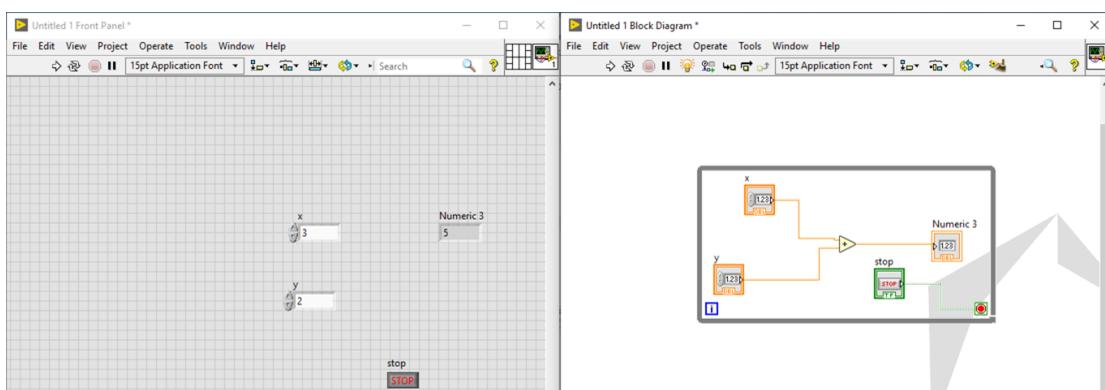
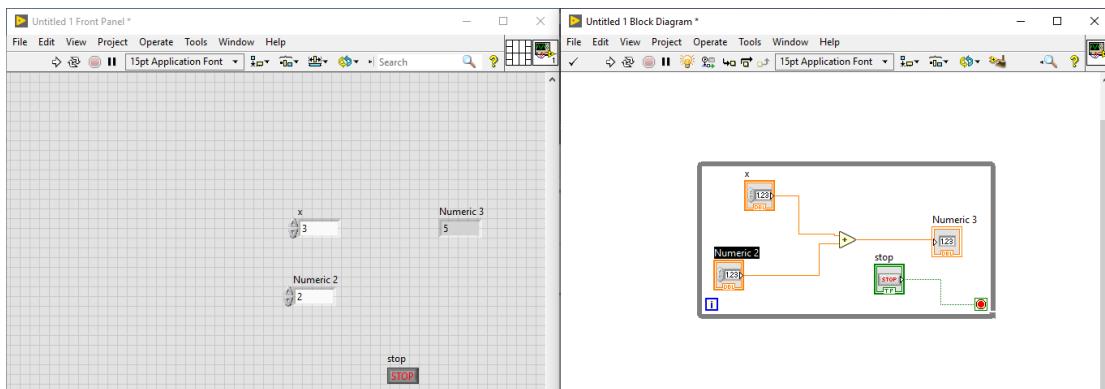
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

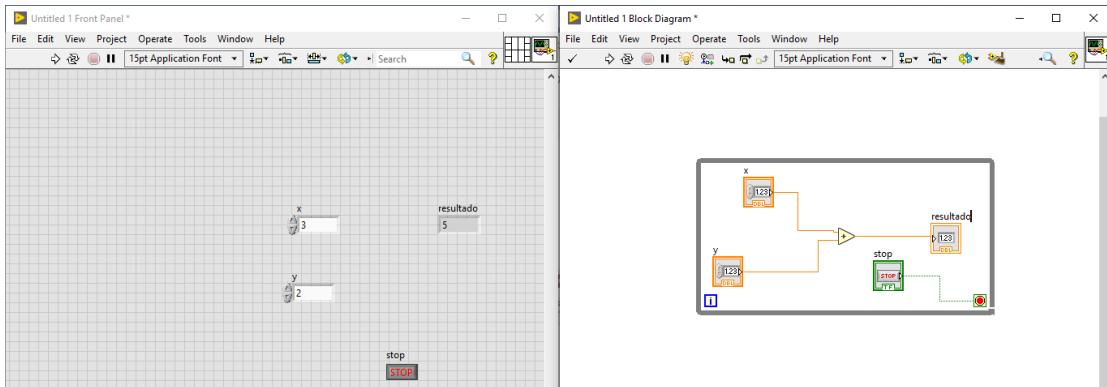




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

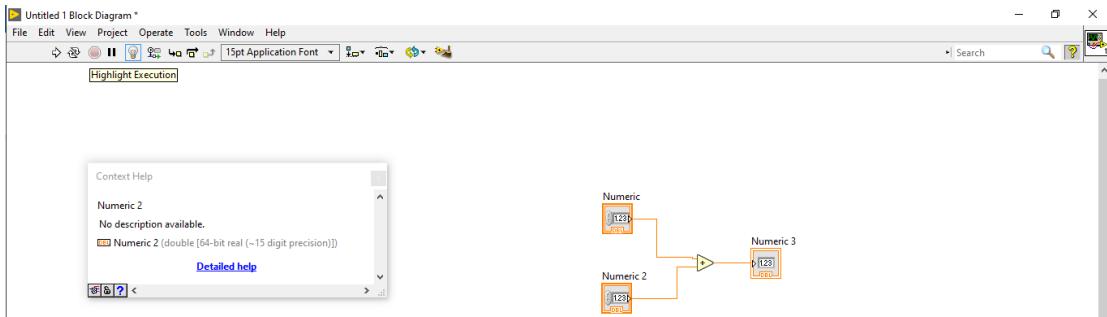
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





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.



Coersion 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.

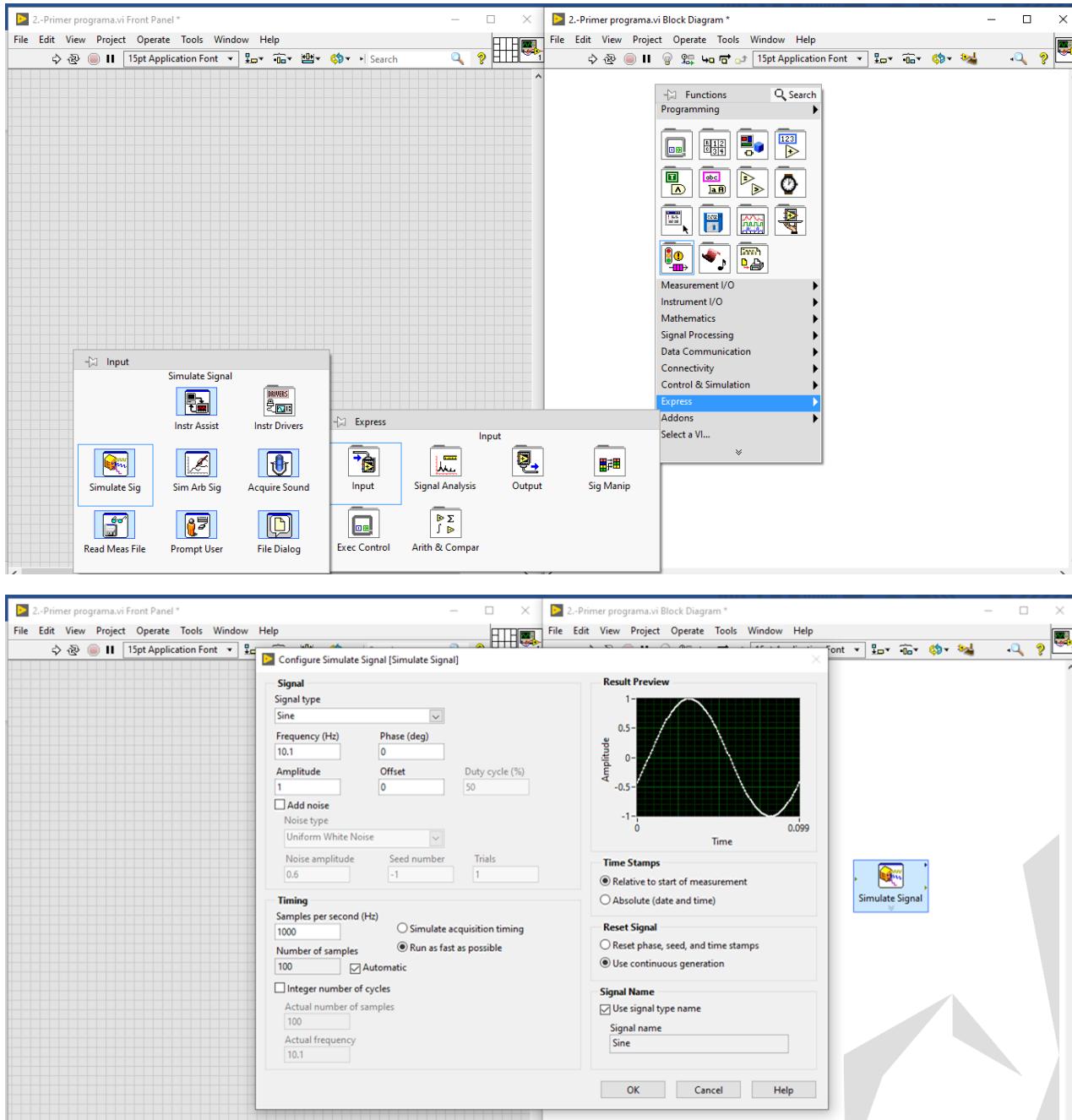
Programa: Generador de Funciones

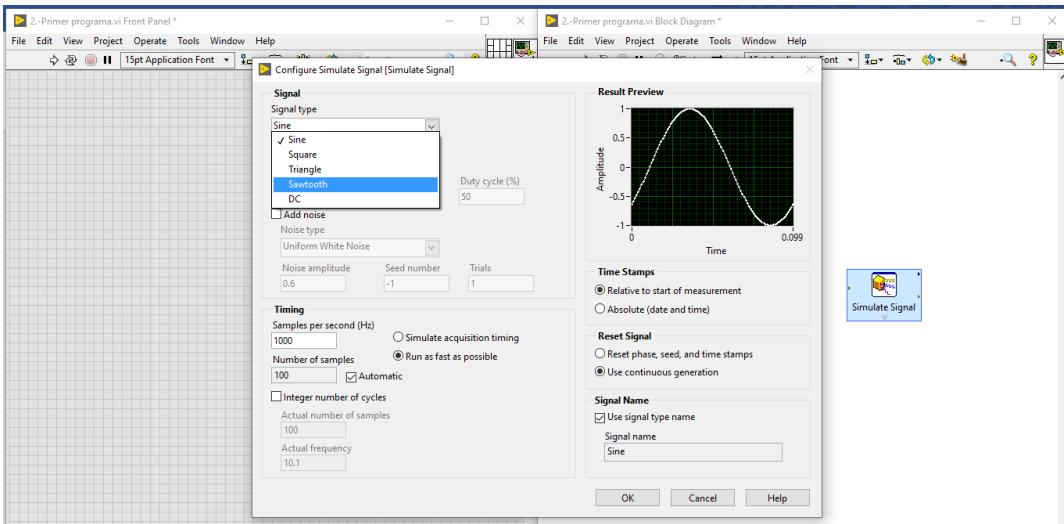
Generador de funciones en LabView con ruido y un análisis de Fourier para encontrar el armónico de la frecuencia y eliminar las restantes con un filtro.

Desarrollo del Programa: Creación de un Simulador de Funciones

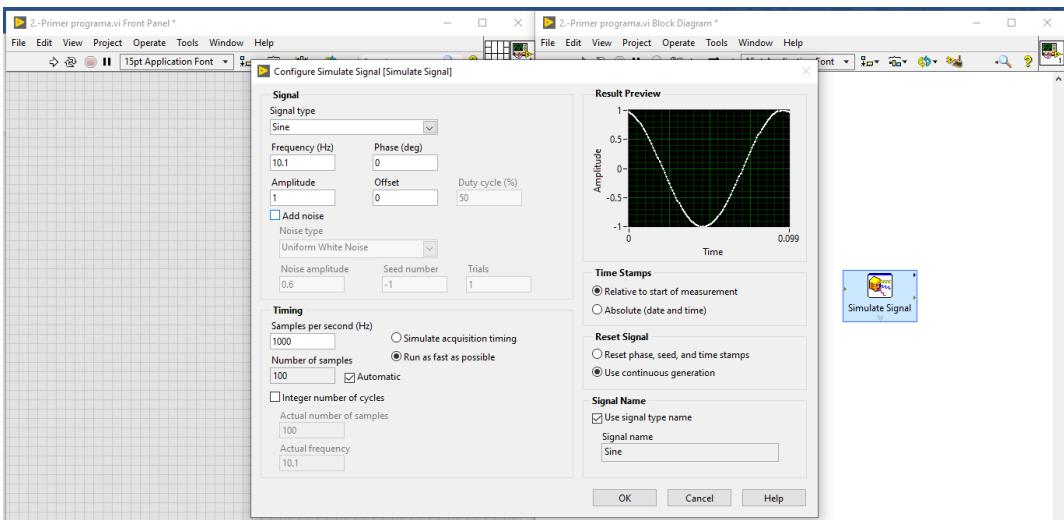
Block Diagram - Simulate Sig: Creación de un Simulador de Funciones

Vamos a crear un simulador de funciones.

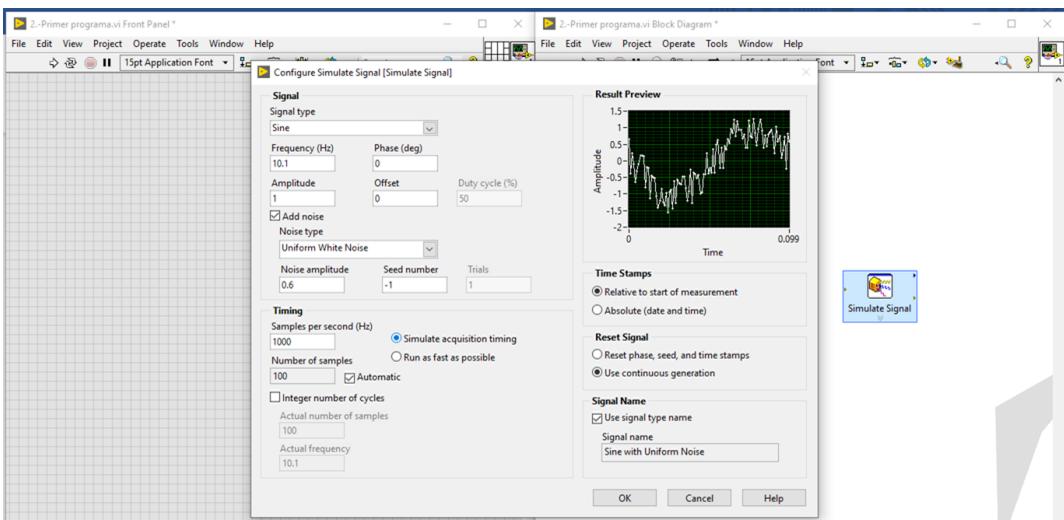




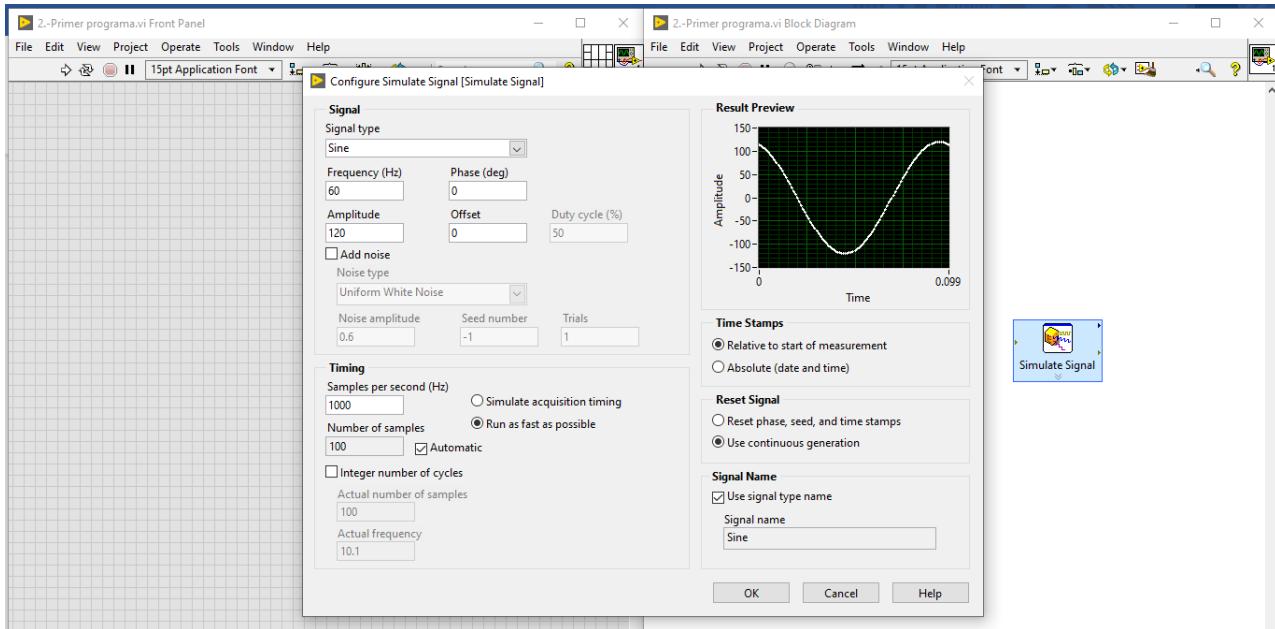
Puedo meter ruido.



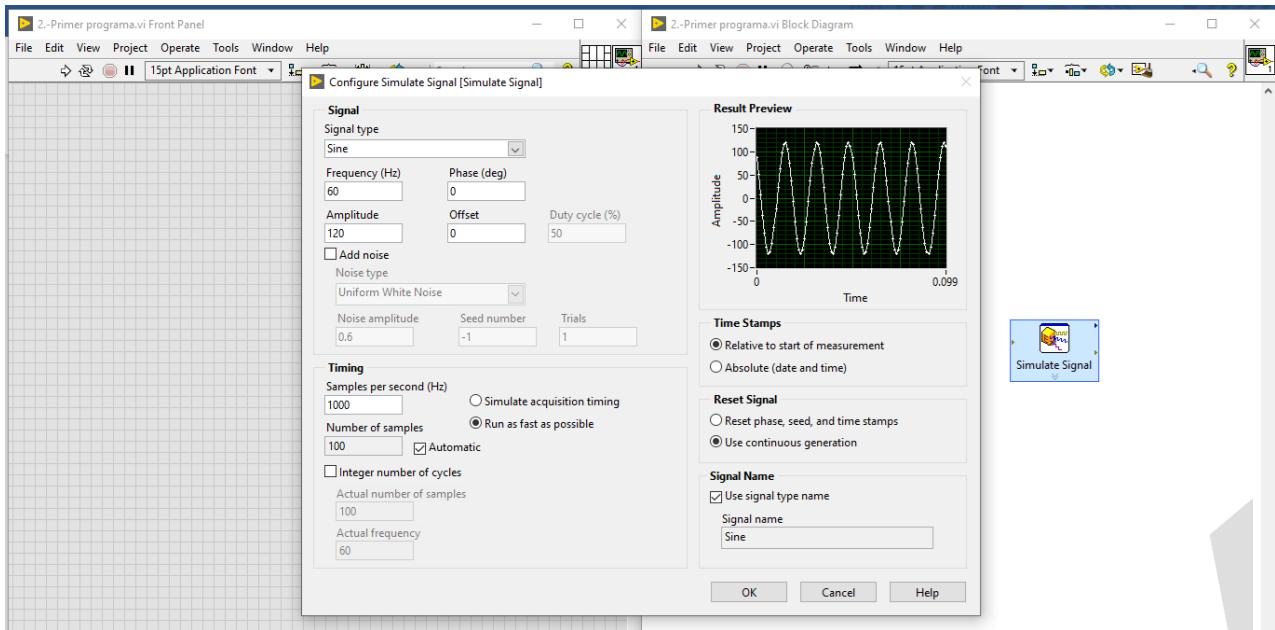
Hacer la señal más lenta, etc.



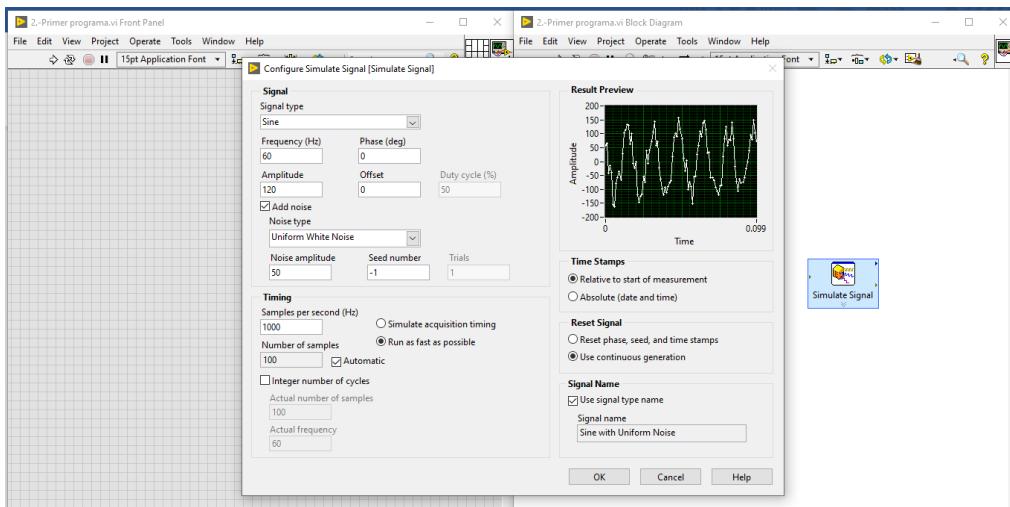
Vamos a crear una señal de 120V de amplitud a 60 Hz como sale en la toma de corriente.



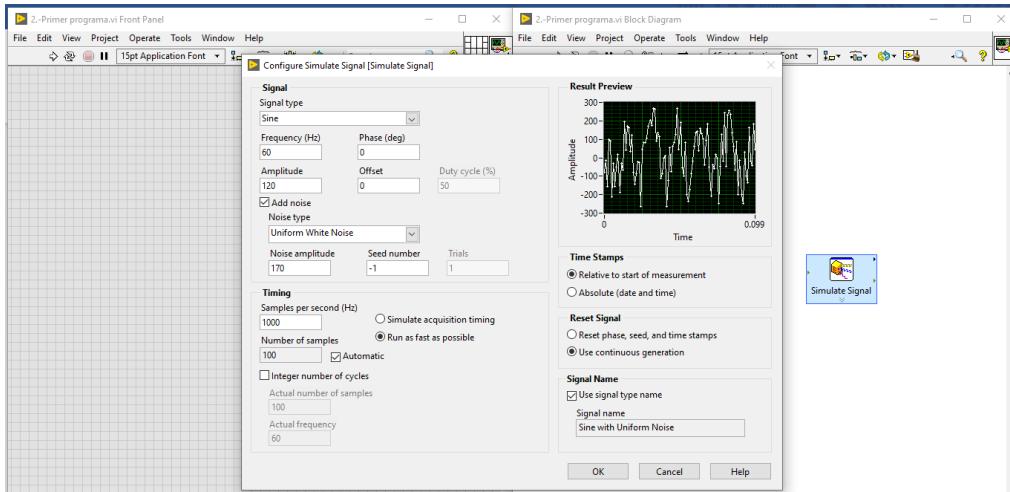
La señal se verá así, aunque en la vida real se ve con más ruido si la medimos con el osciloscopio, por lo que vamos a añadir ruido dando clic en el botón de Add noise y la amplitud del ruido la pondremos de 50V.



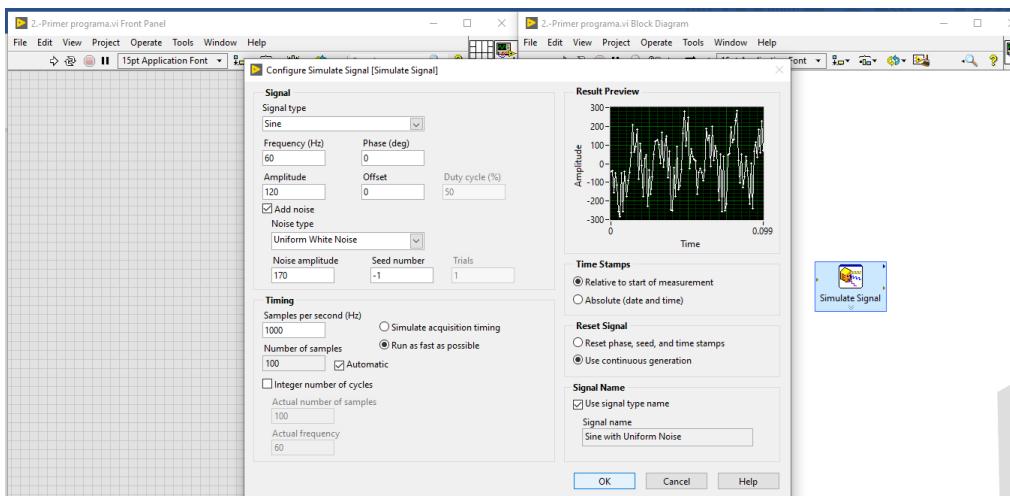
Ya con ruido nuestra señal se verá así.



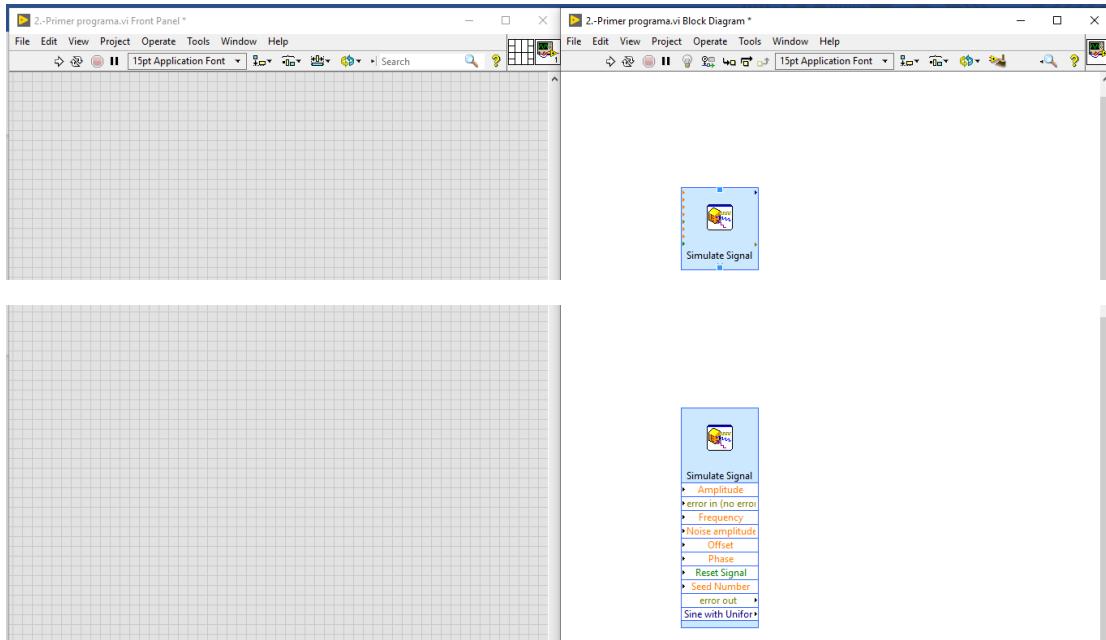
Si aumento el ruido a 170 se verá así.



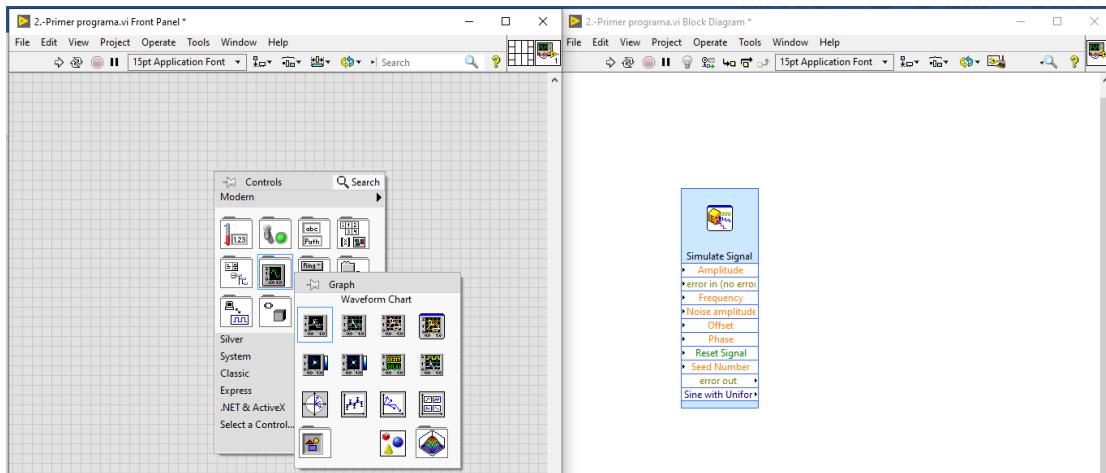
Y le daré clic al botón de Ok.



Para poder configurar y conectar la señal creada, vamos a extender el bloque.

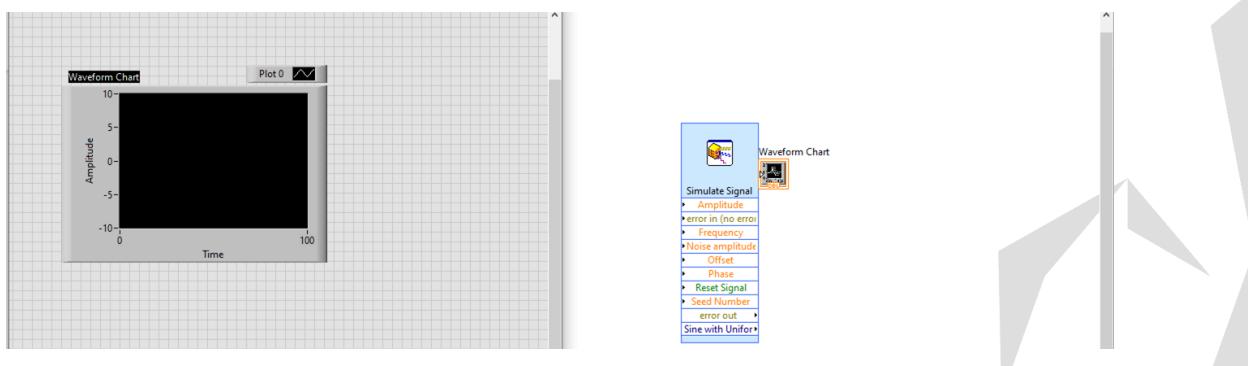


Front Panel: Creación de un Simulador de Funciones

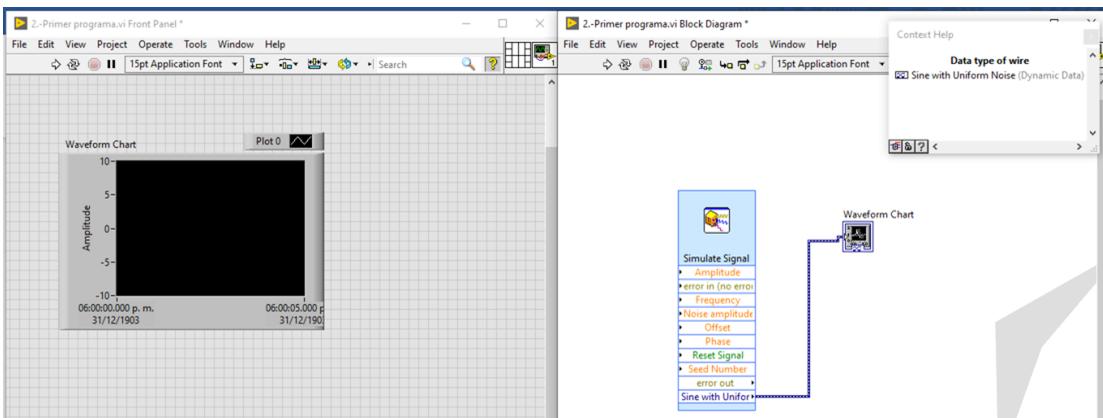
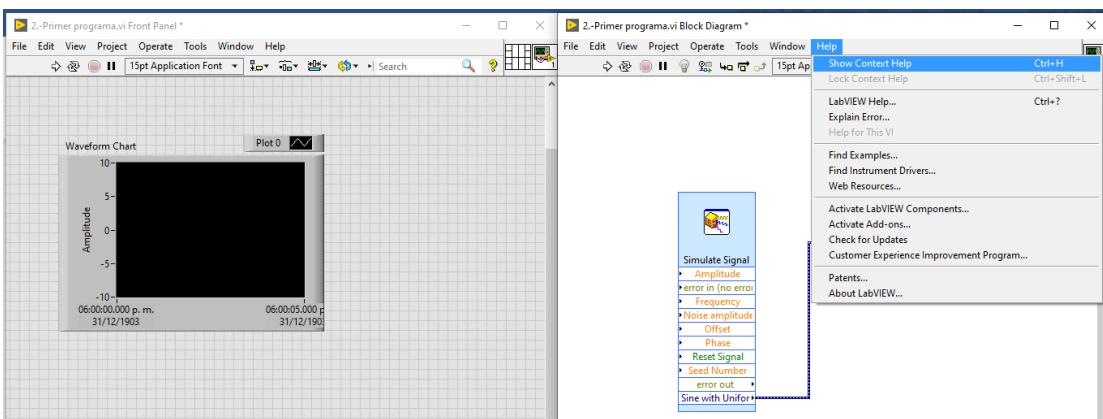
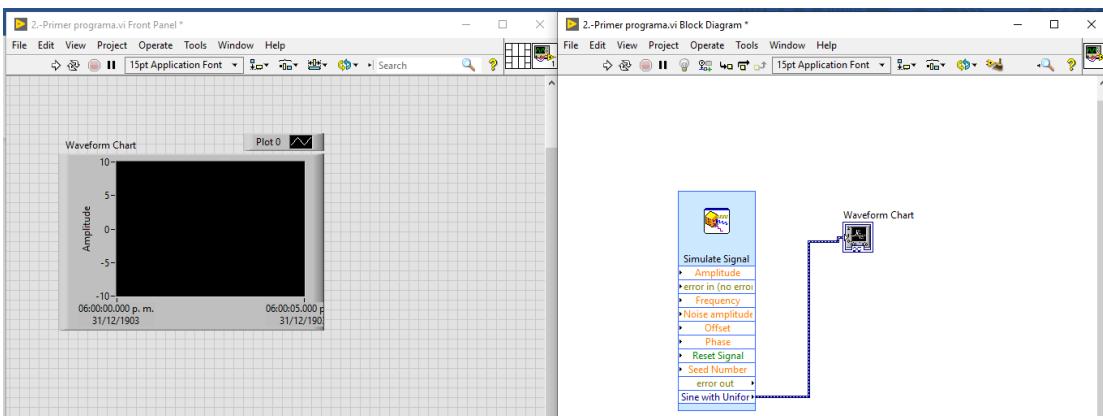
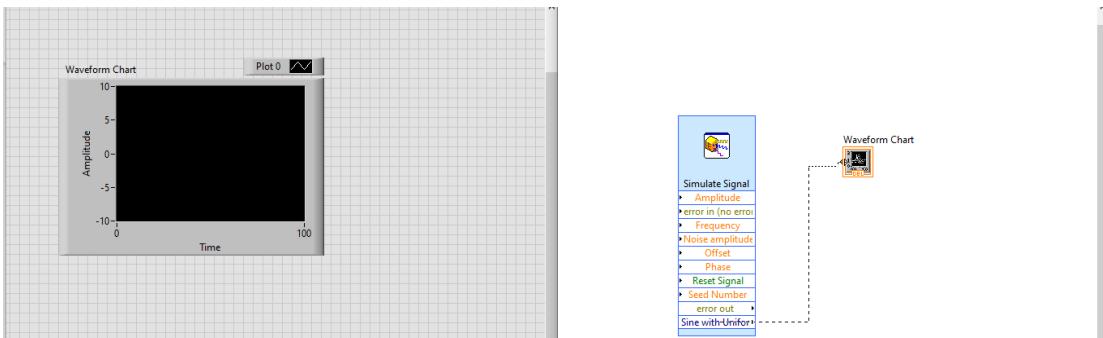


Front Panel - Waveform Chart: Ventana que Muestra una Señal (Dynamic Data)

El Waveform Chart muestra gráficas de tipo Dynamic Data, las cuales son señales virtuales o reales.

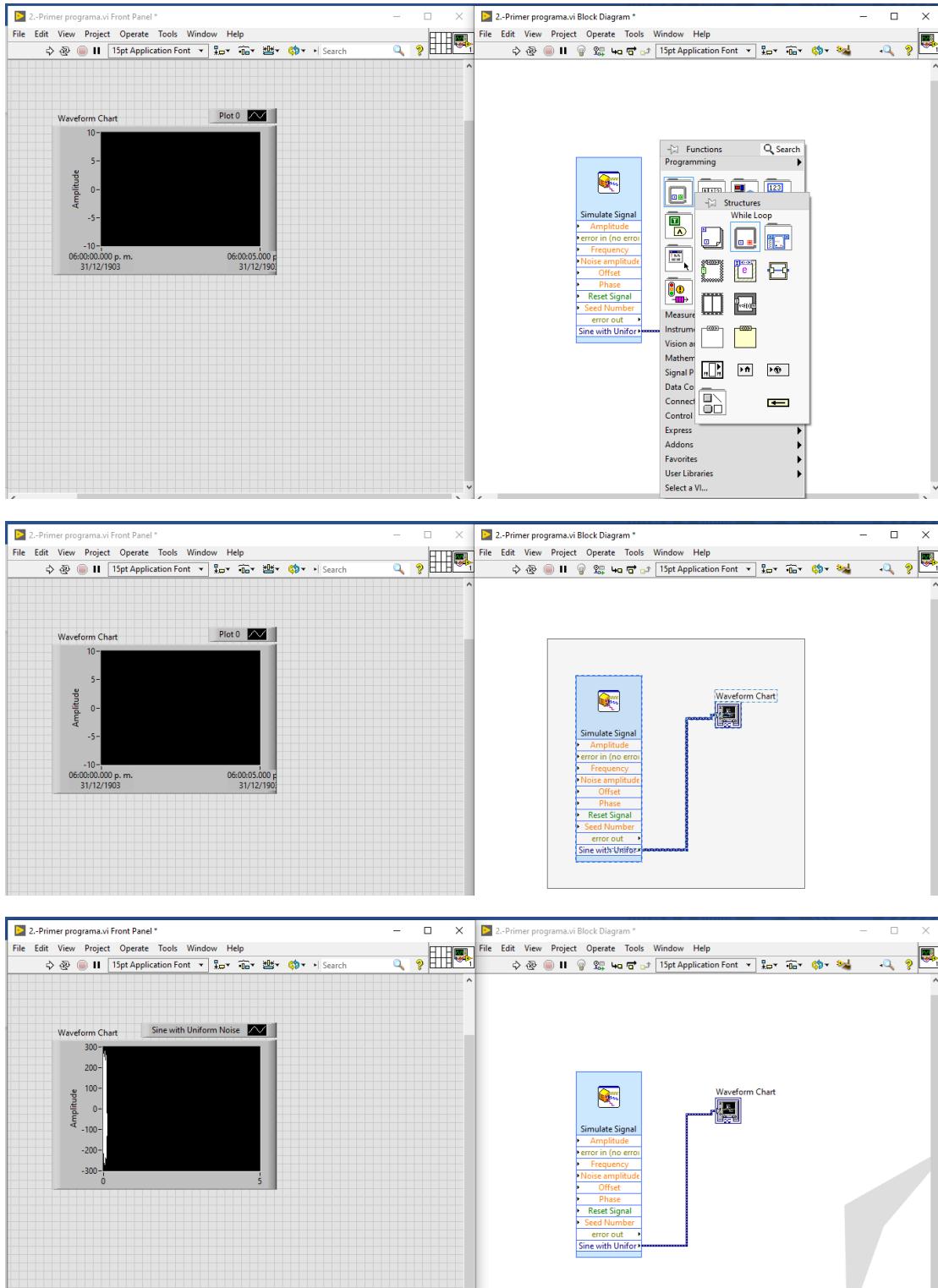


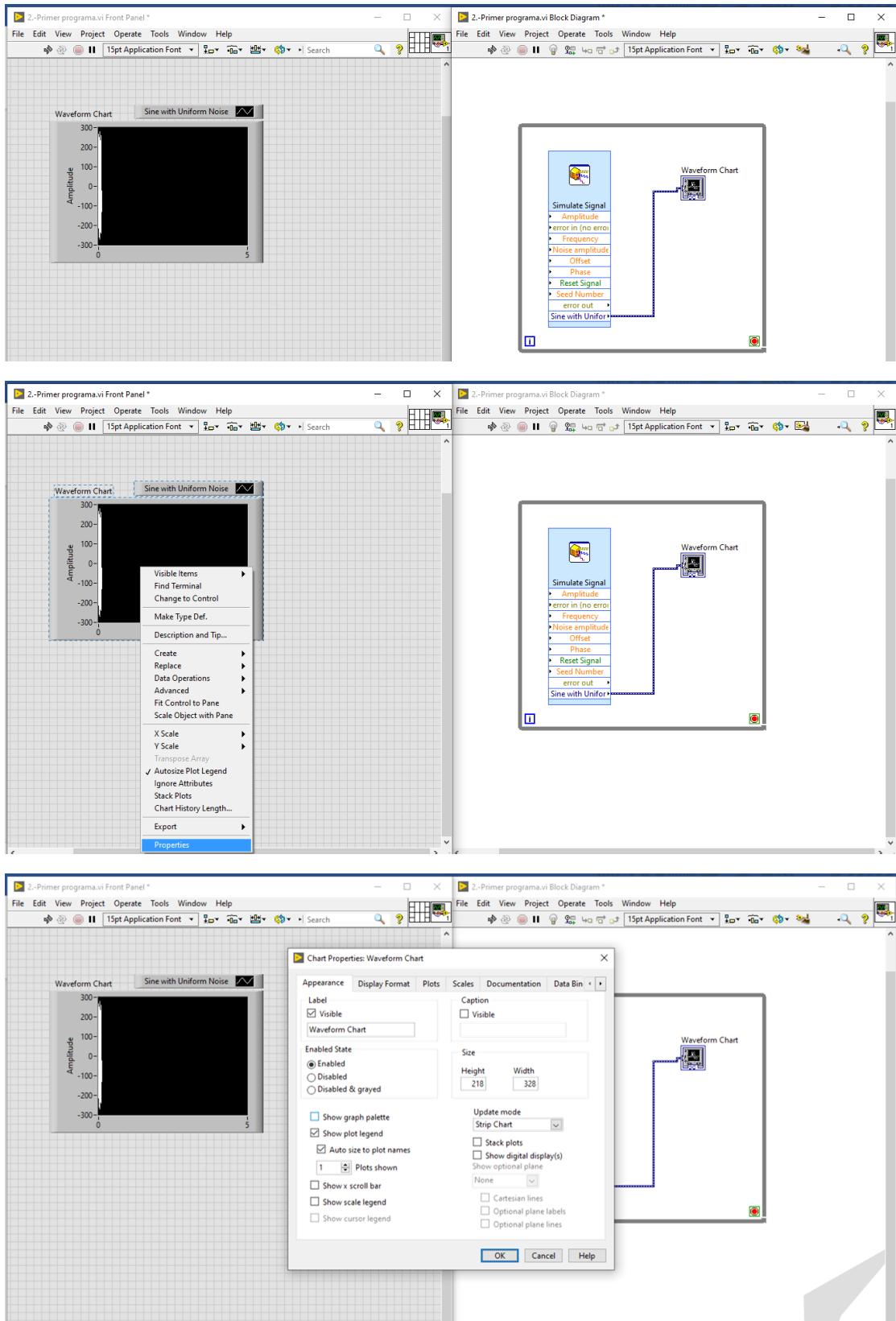
Vamos a unir los bloques.



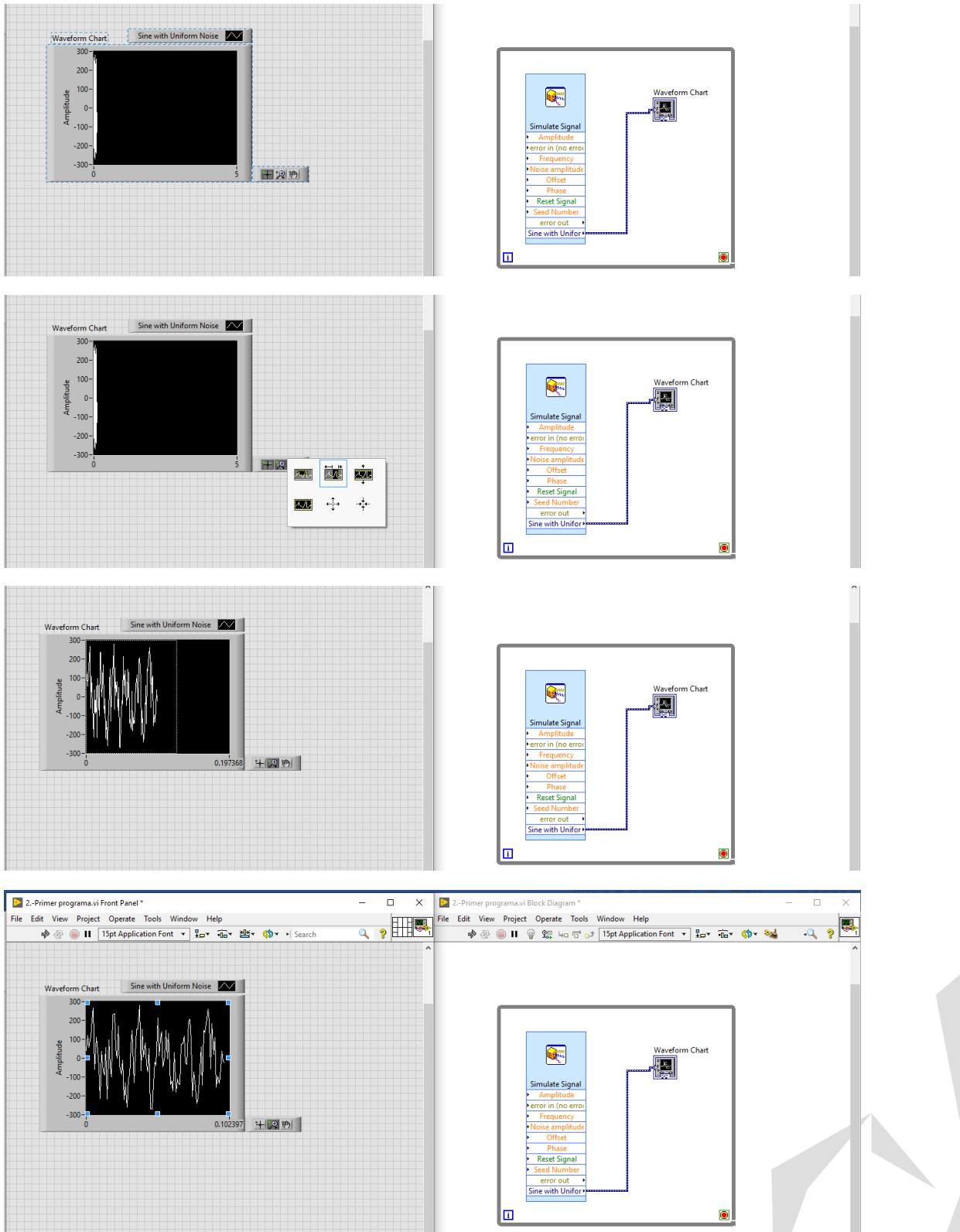
Block Diagram - Bucle While: Generación de Muestras de la Señal

Agregamos un ciclo while para que vaya formado varias muestras de la señal creada con el generador de funciones. Si no agregamos el ciclo while solo va a correr una pequeña parte de la señal, en específico 128 muestras.

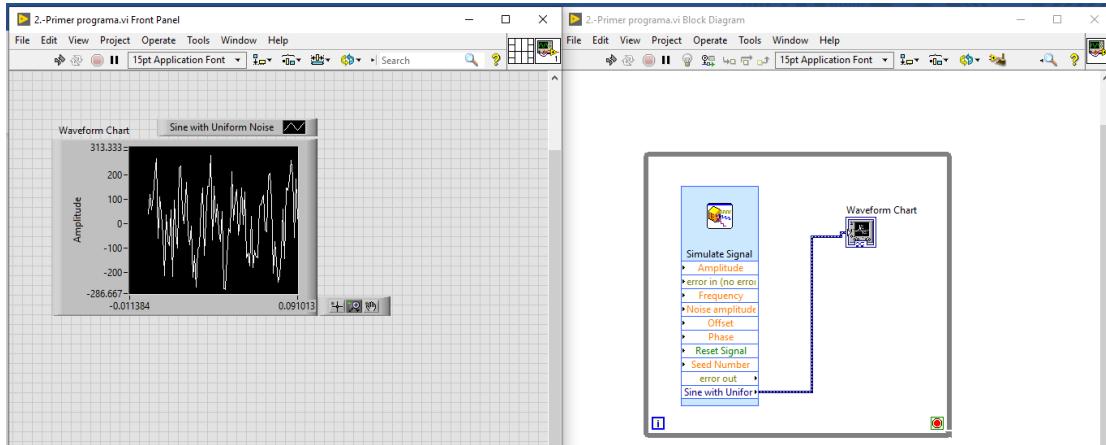




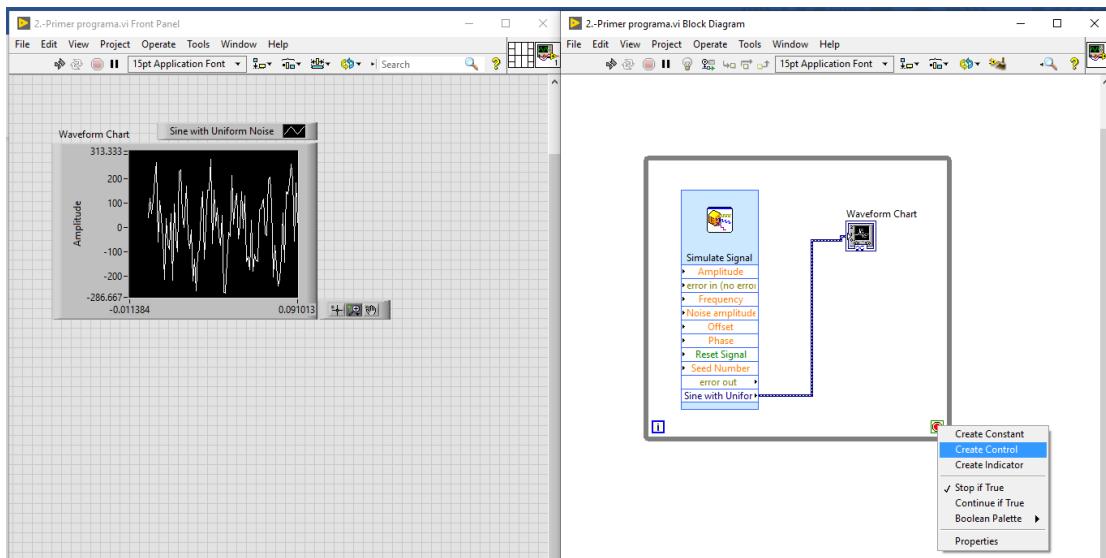
Con la herramienta de la esquina inferior derecha del bloque Waveform Chart llamada **graph palette** puedo hacer zoom al contenido de la ventana que muestra la gráfica.



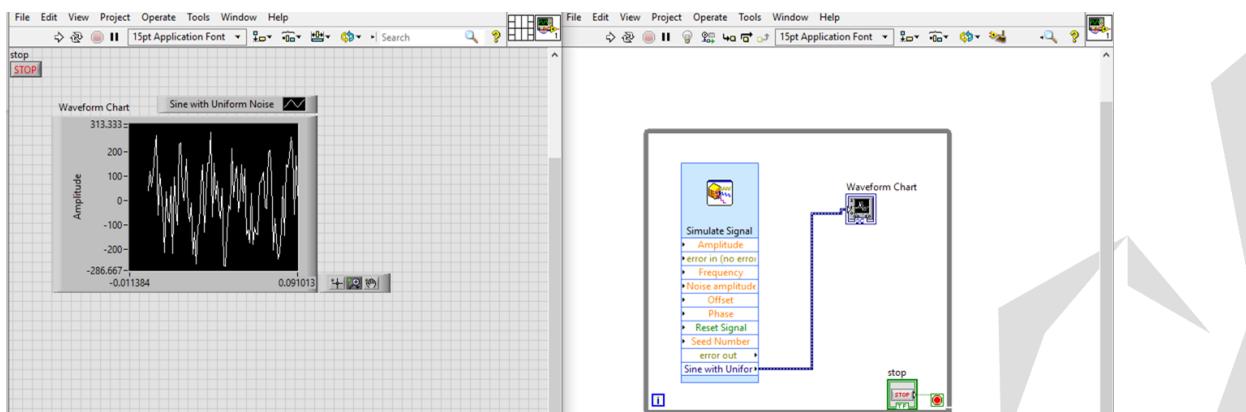
LabView es un lenguaje auto compilado y como no he puesto una condición que pare al programa, me lo indica por medio de una flecha rota en la parte superior de ambas ventanas.



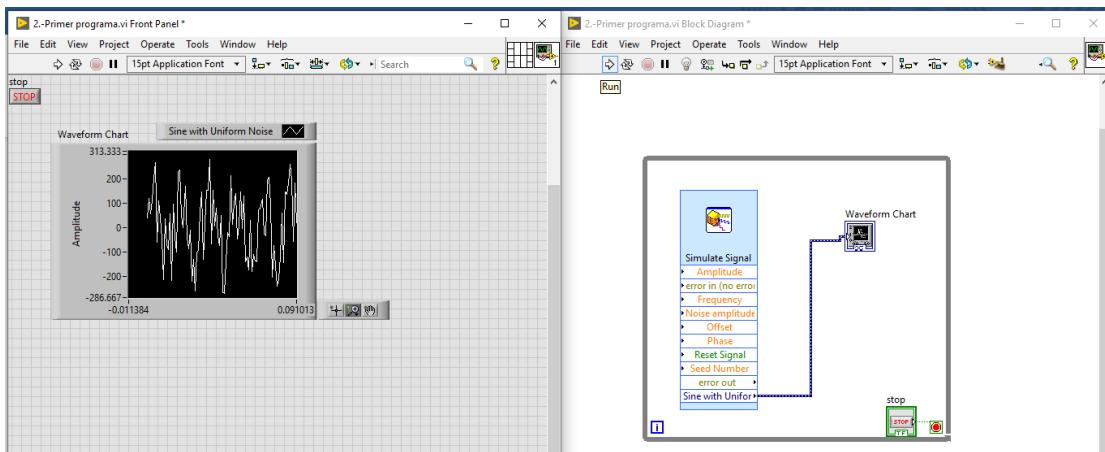
Block Diagram - Bucle While: Botón de Stop Para Detener de forma Manual el Bucle
Para poner el botón de Stop se da clic derecho en el botón de la esquina y se selecciona lo siguiente.



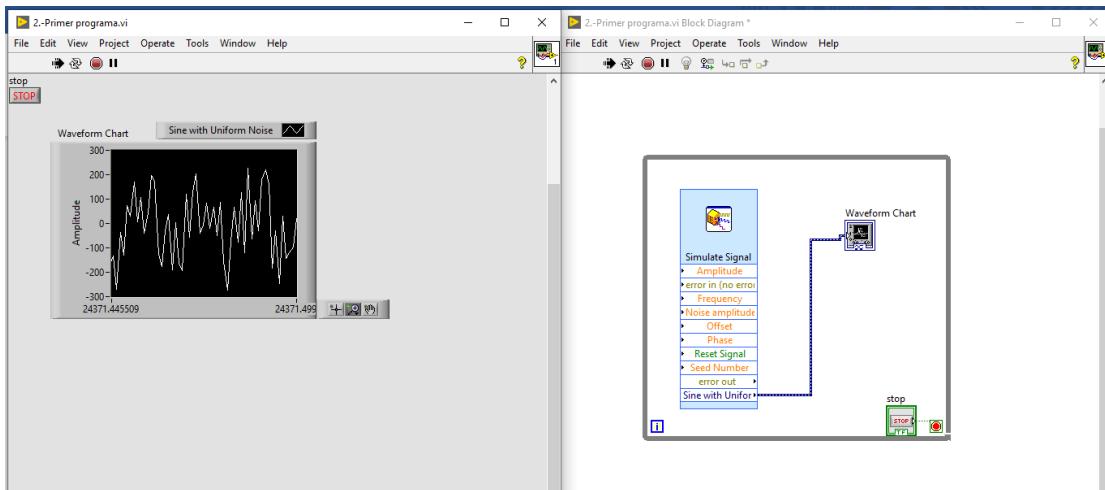
Ya con esto aparecerá el botón de STOP que es la condición del while.



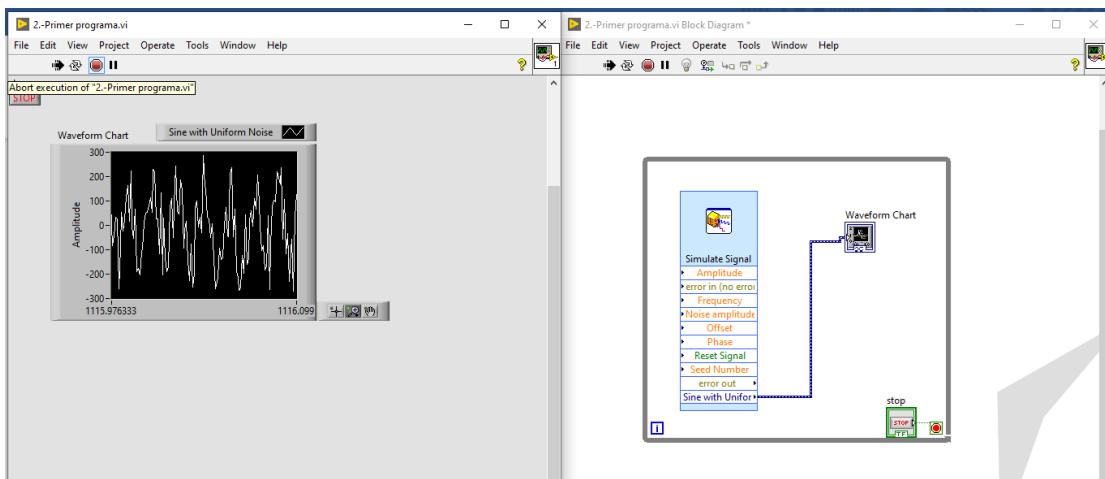
Ya con esto se quita el error y puedo correr mi programa.



Ejecución del Programa: Gráfico de Señal Senoidal con Ruido

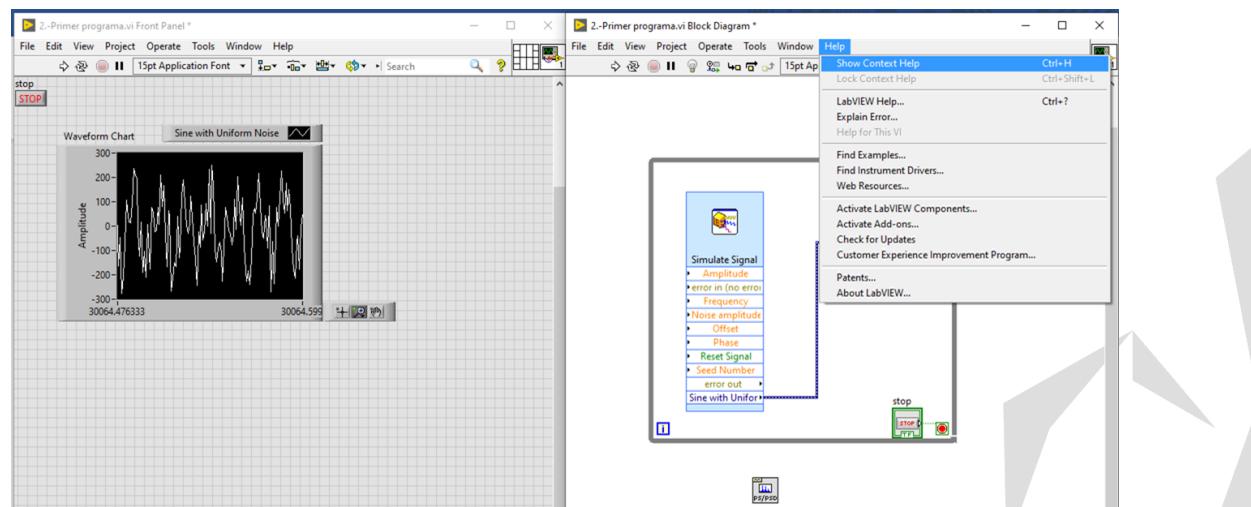
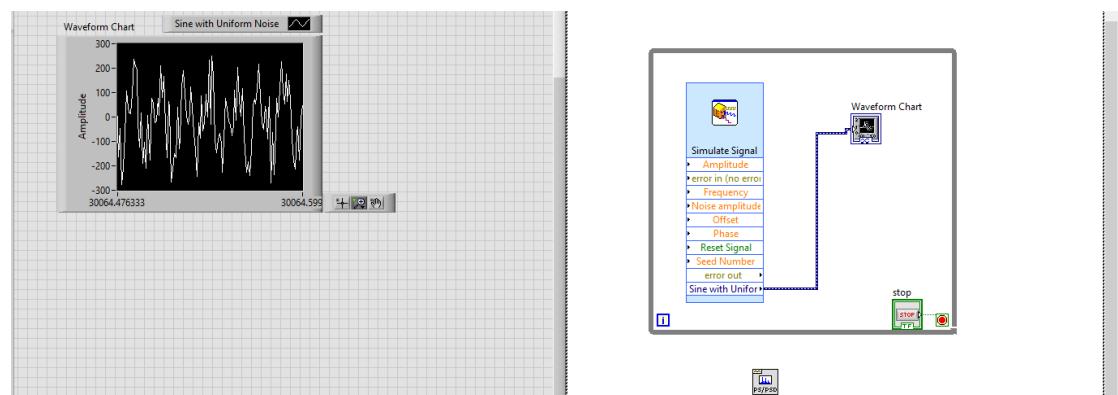
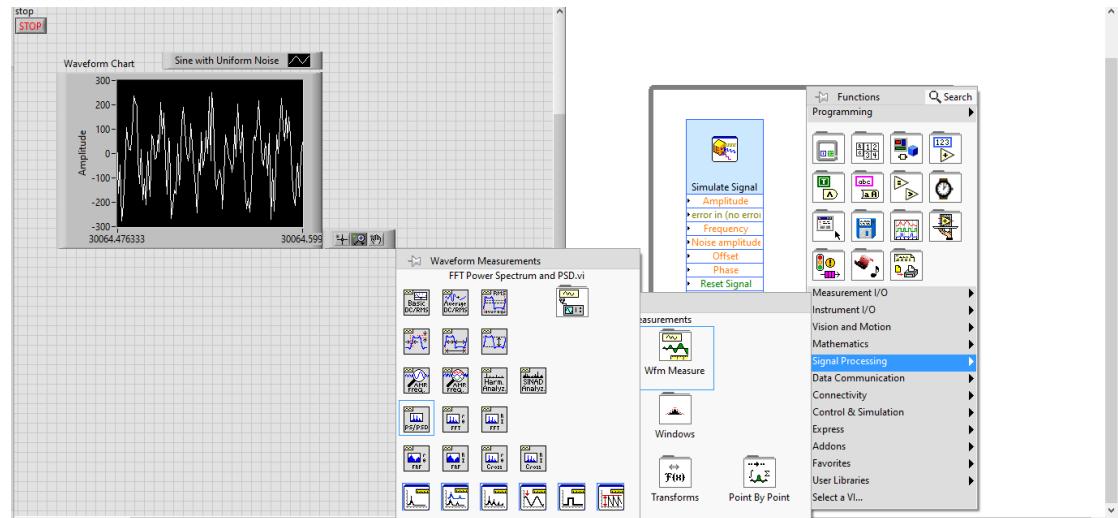


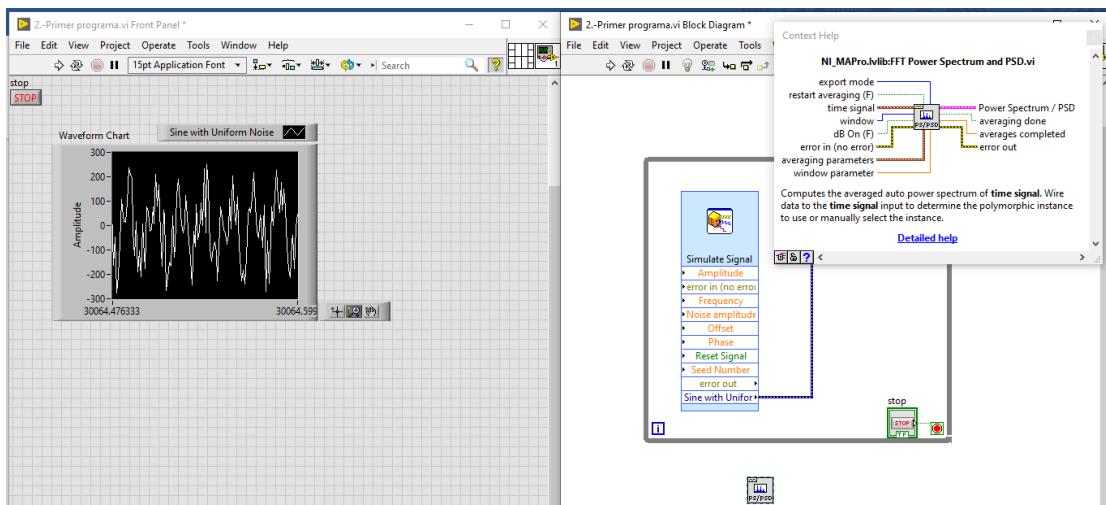
Para que se deje de ejecutar el programa debo dar clic en el botón rojo de stop.



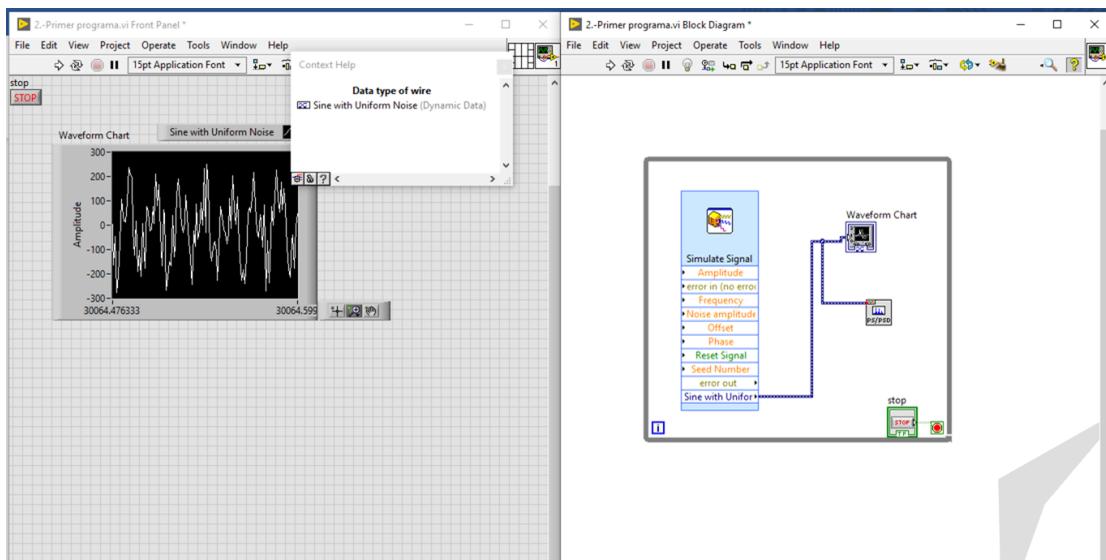
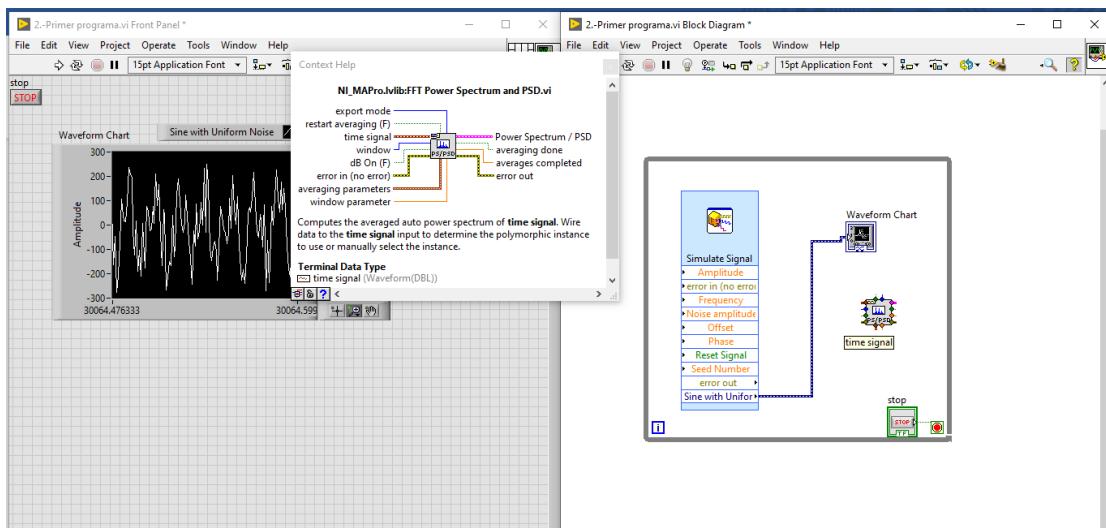
Block Diagram - FFT Power Spectrum and PSD: Medir la Frecuencia de una Señal

Ahora debo quitar el ruido, para ello necesito saber la frecuencia del ruido y así agregar un filtro pasa bajas, pasa altas o pasa bandas. Para saber la frecuencia del ruido puedo usar una transformada de Fourier porque el ruido está sobre la señal misma, para ello necesito pasarlo al dominio de la frecuencia. El chiste de la instrumentación virtual es aplicar el análisis de software por medio de LabView para analizar empíricamente una señal real (proveniente del mundo exterior) o virtual.



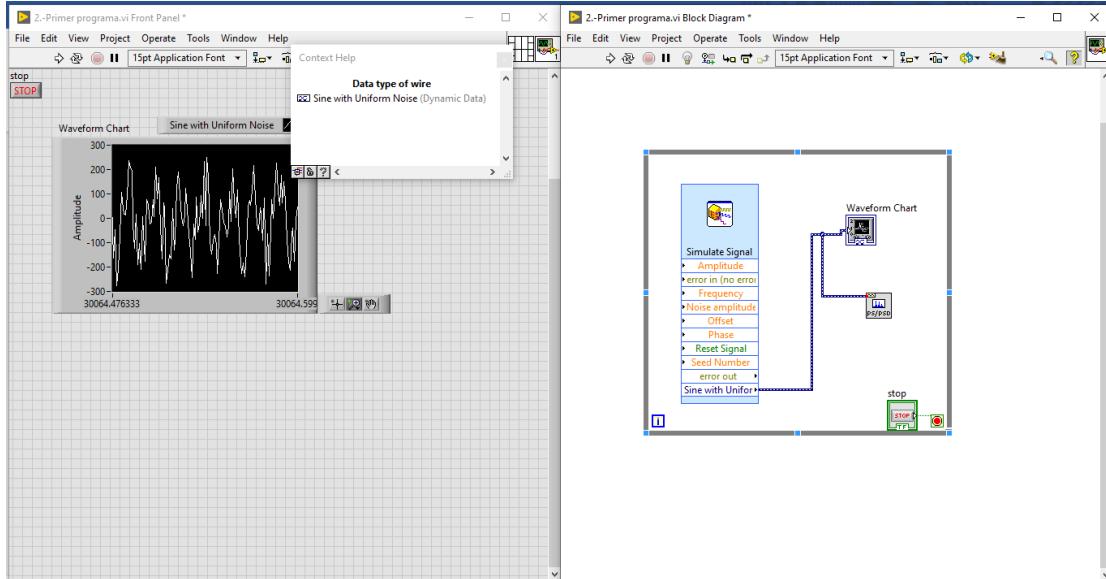


La terminal que se llama time signal la voy a conectar al nodo de la señal.

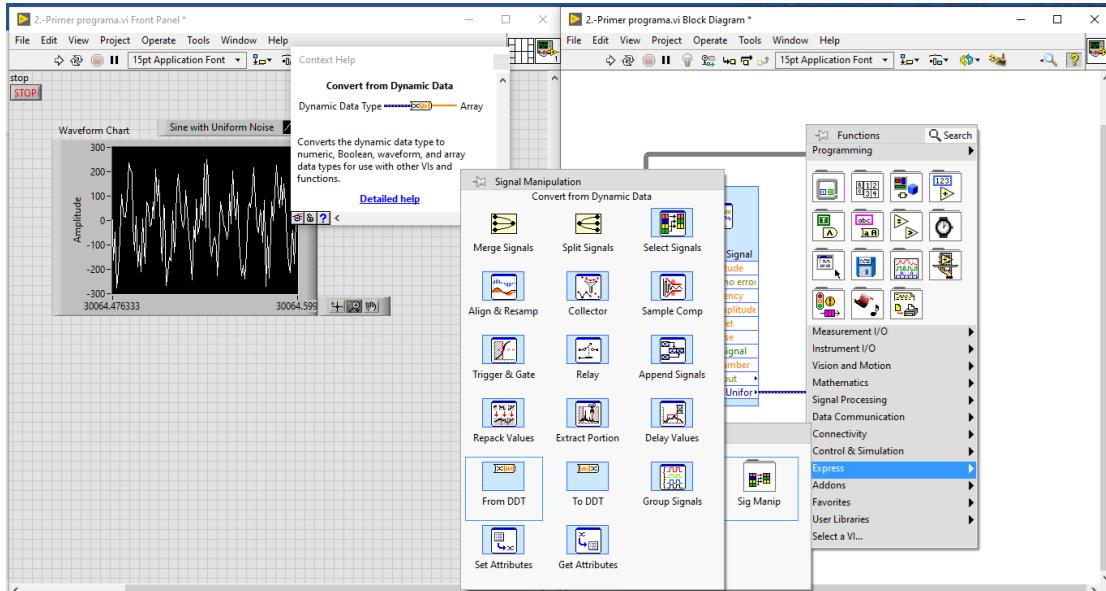


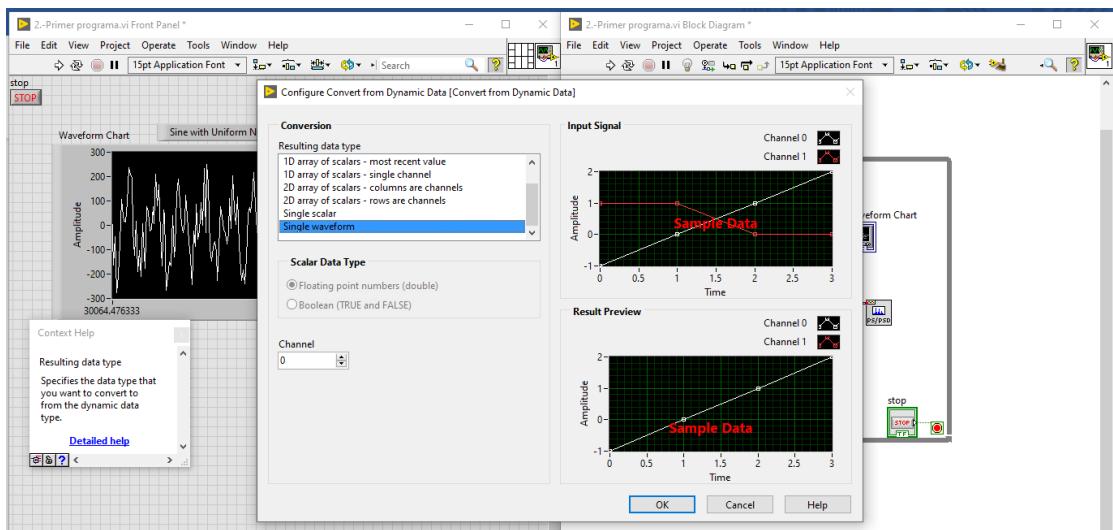
Block Diagram - From DDT: Convierte una Señal a un Array, Evitando un Coersion dot

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.

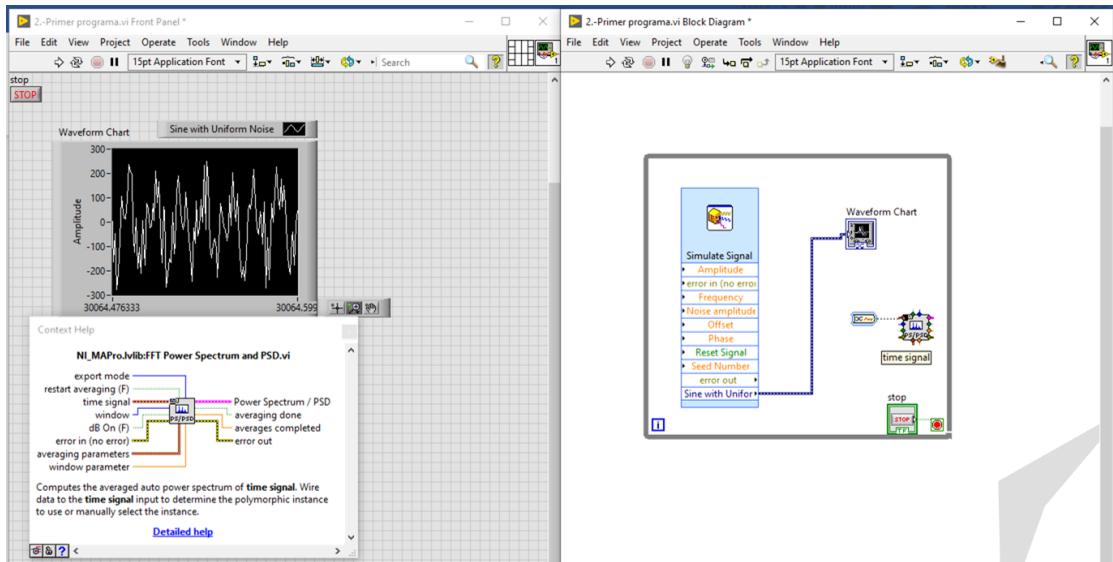
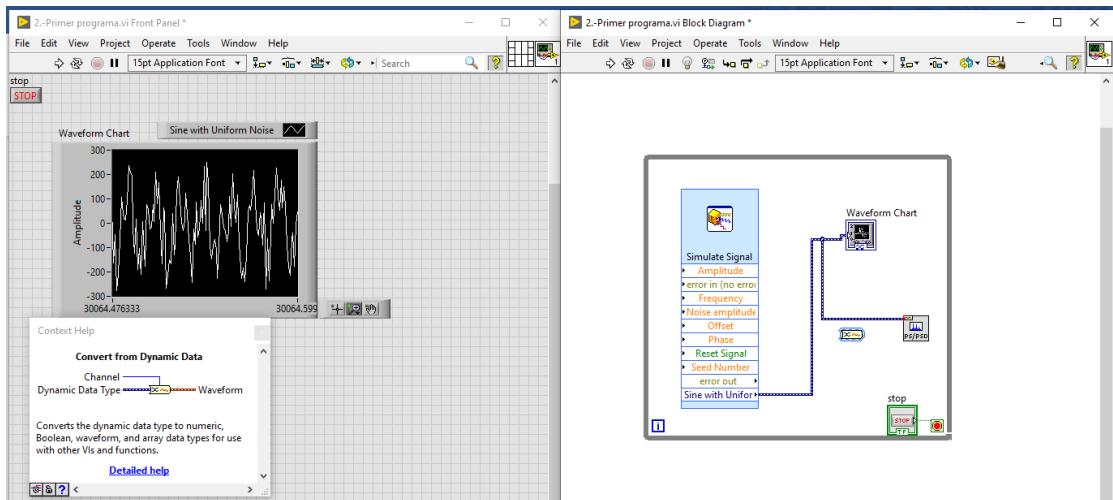


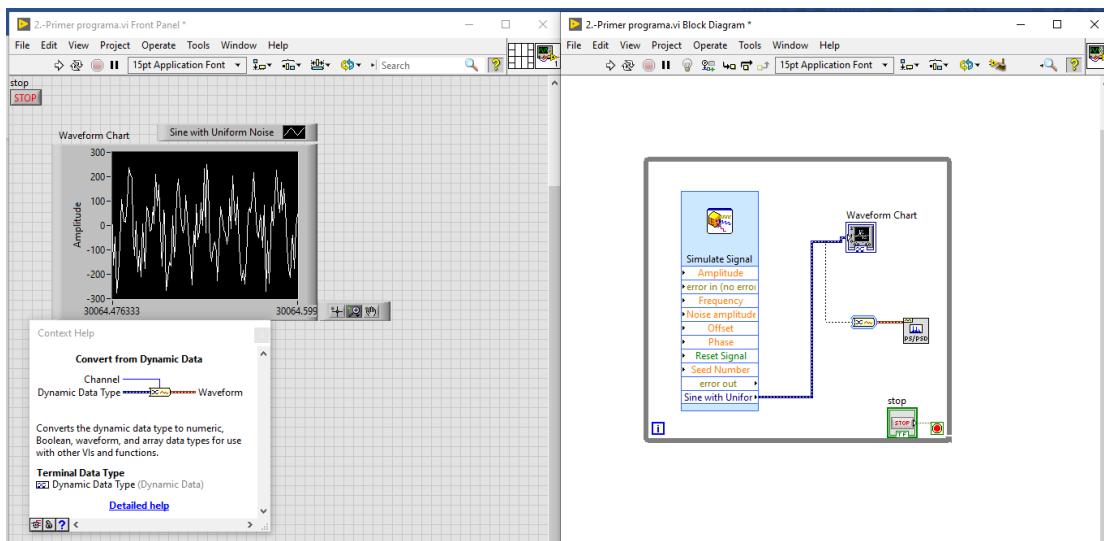
Ahora vamos a hacer la conversión de mi dato de tipo dinámico (la señal) a un arreglo de valores, para que cada uno pueda tener su lugar





Convierto una señal dinámica a una forma de onda.

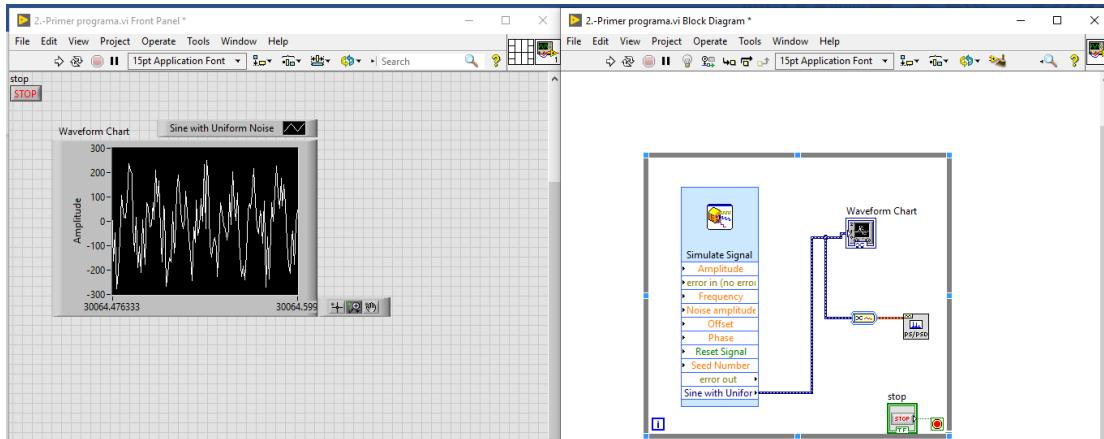




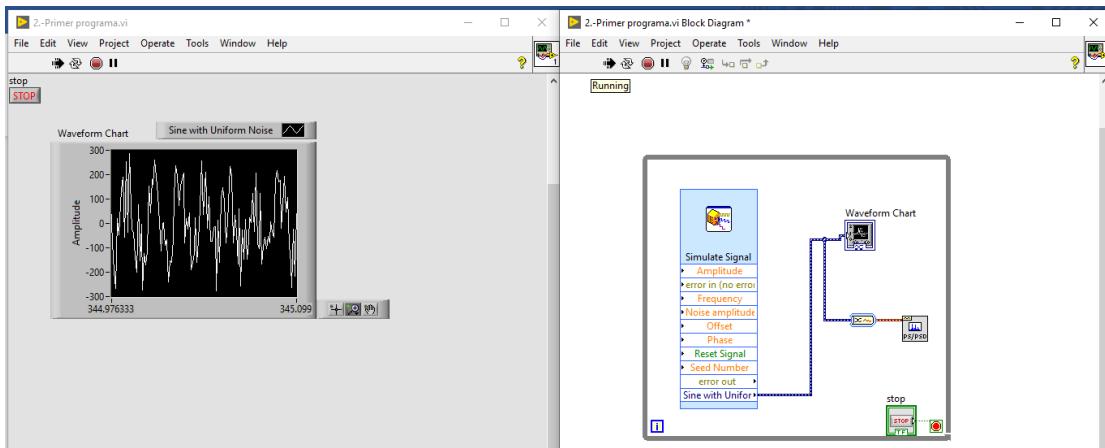
- **Dynamic Data Type:** Es cualquier señal ya sea virtual o real que capte por medio de una tarjeta de desarrollo, ya sea de campo electromagnético, una señal de voltaje (sensor), etc.
- **Waveform Data Type:** Es un tipo de dato que representa una forma de onda que tiene incluida la amplitud, el cambio de la señal respecto al tiempo, etc.

Este tipo de datos son muy específicos de National Instruments en LabView, no existen en otros tipos de programación.

Con manualmente convertir un tipo de dato a otro por medio de un bloque, logramos que se quite el coercion dot, evitando que se pierdan datos.

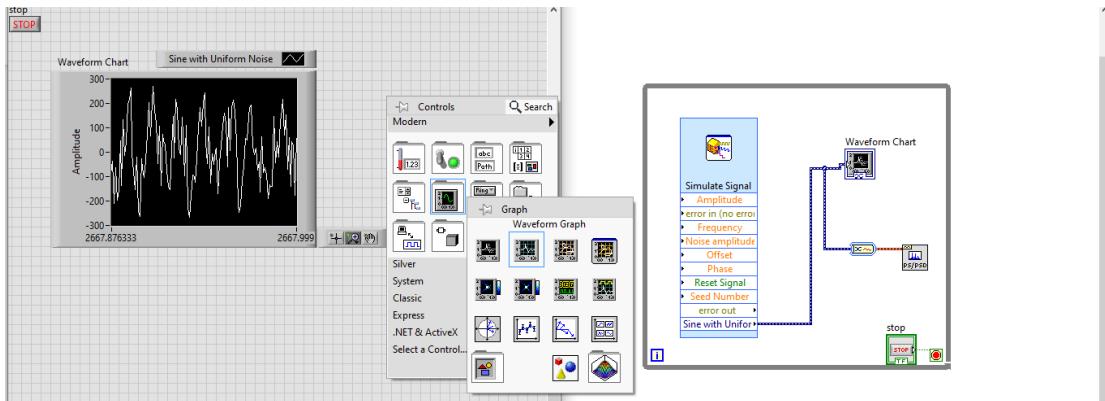


Ejecución del Programa: Conversión de Datos Dynamic a Array Exitosa

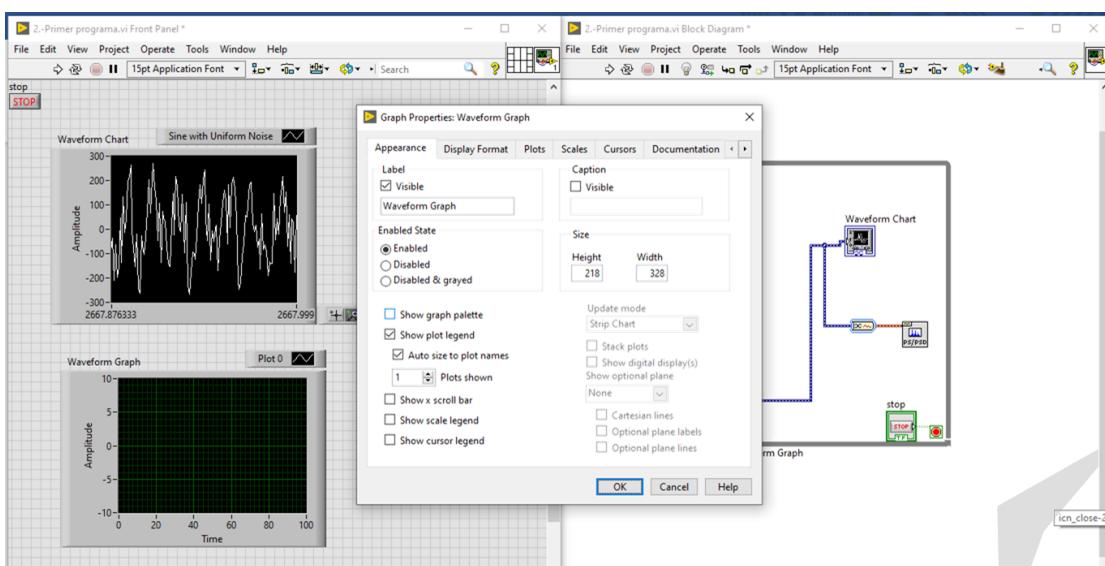


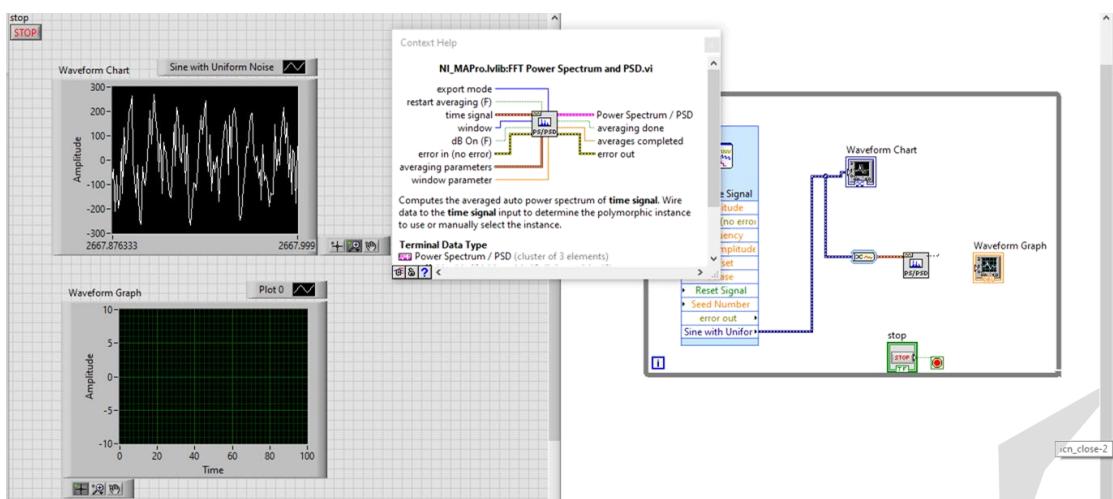
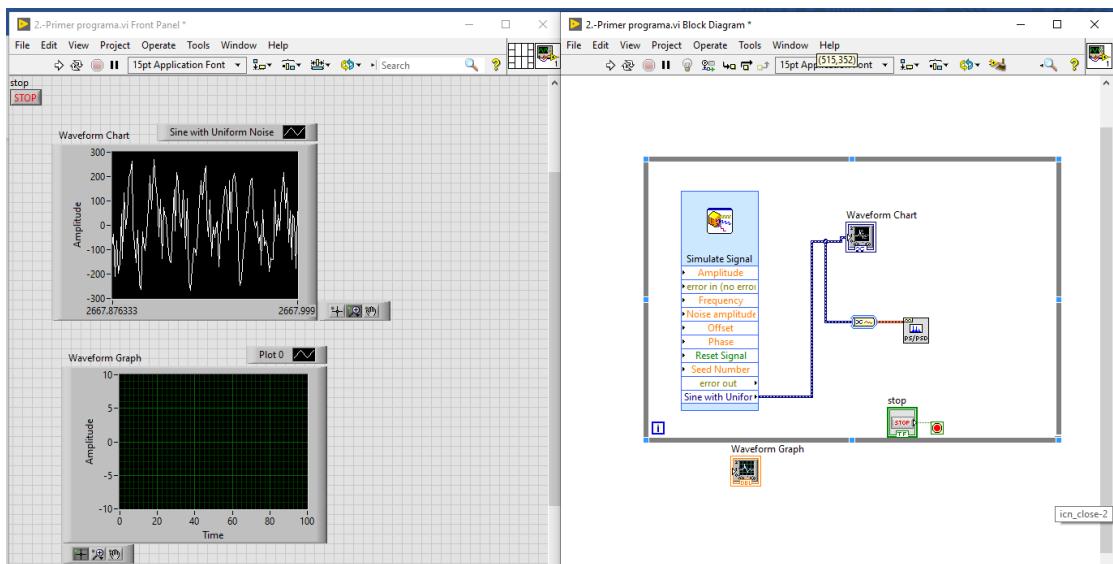
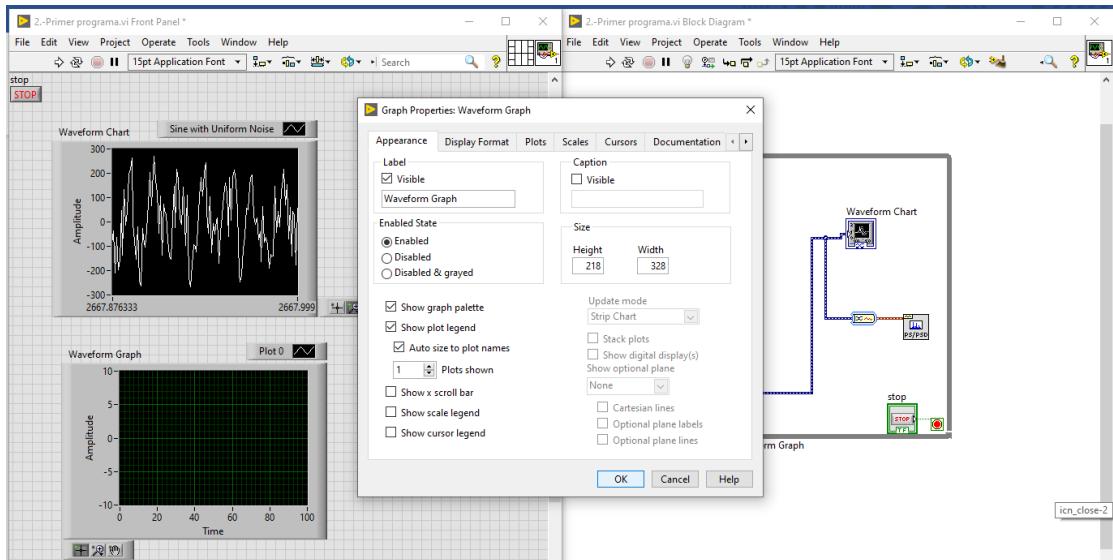
Front Panel - Waveform Graph: Ventana que Muestra una Señal (Array)

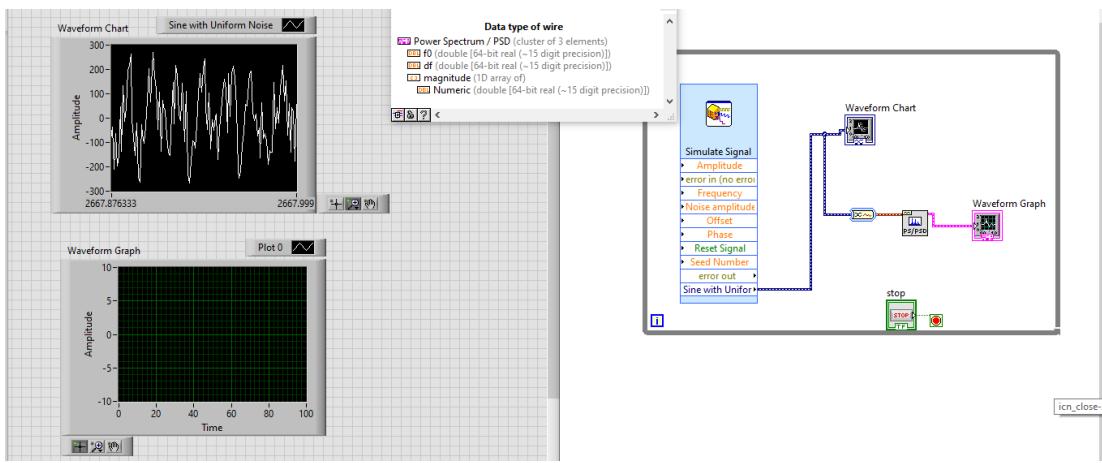
El Waveform Graph muestra gráficas de tipo Array, las cuales son cualquier tipo de grupos de números.



Y voy a agregar un graph palette.

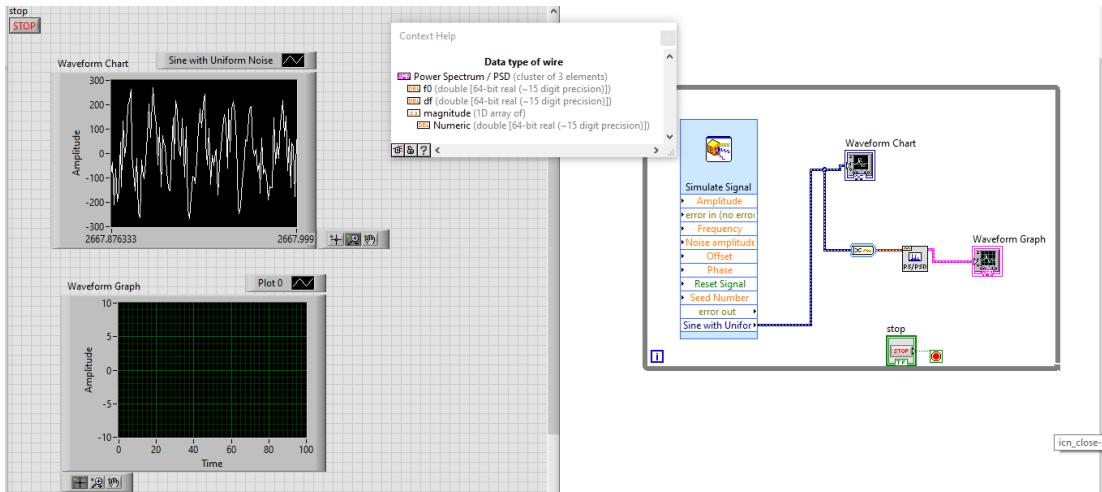




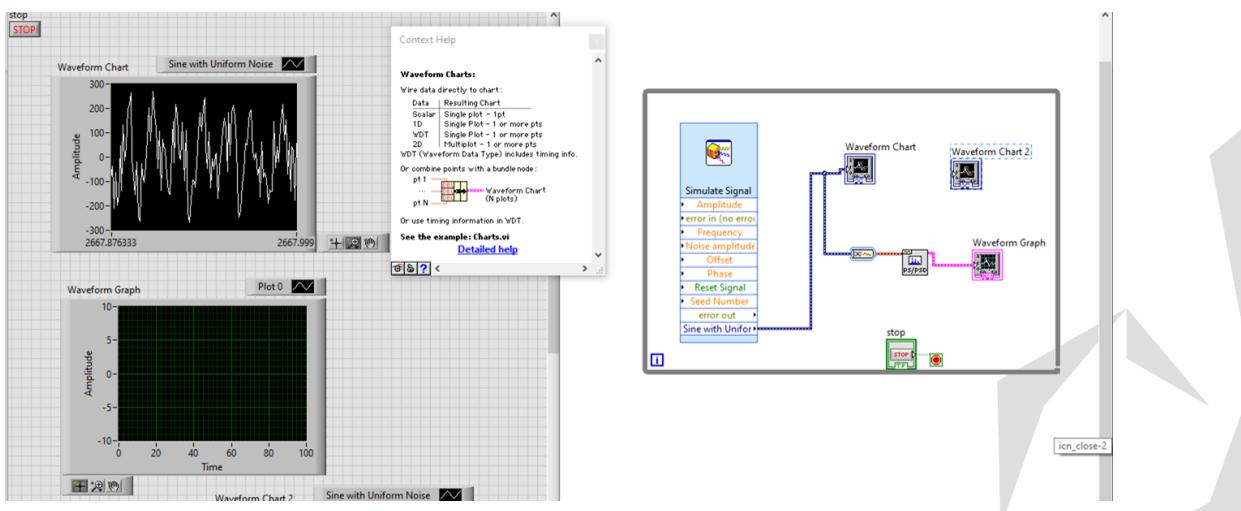


Cluster: Varios Tipos de Datos Agrupados en Uno

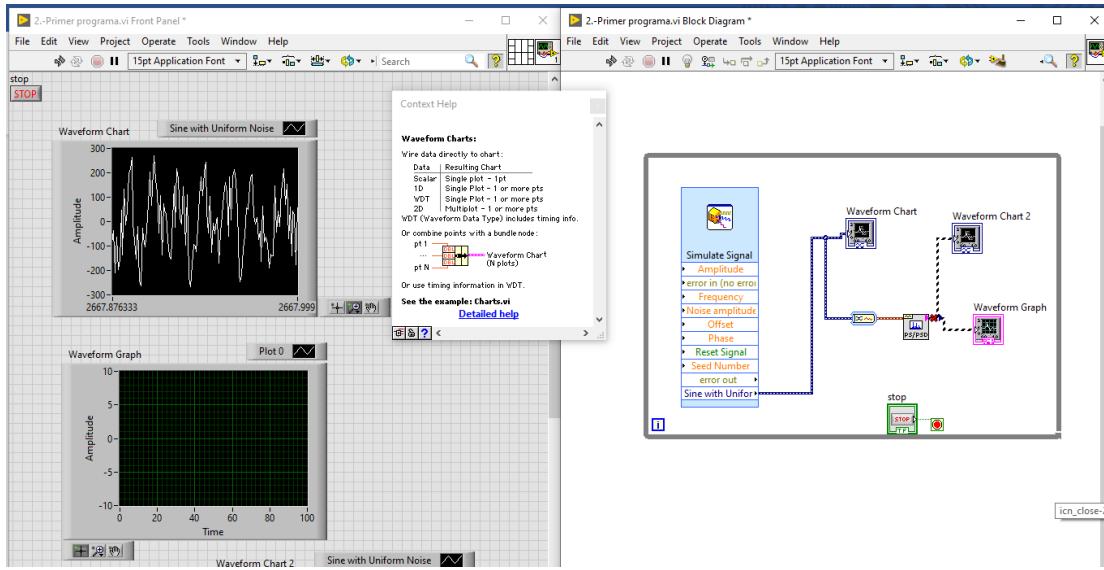
El Waveform Graph recibe en su terminal de entrada un tipo de dato cluster, que es un tipo de dato definido por el usuario que recibe y encapsula varios.



Si presiono la tecla CTRL y arrastro un elemento lo duplico, si intento conectar estos no se va a poder conectar por los tipos de datos que entran y/o salen.



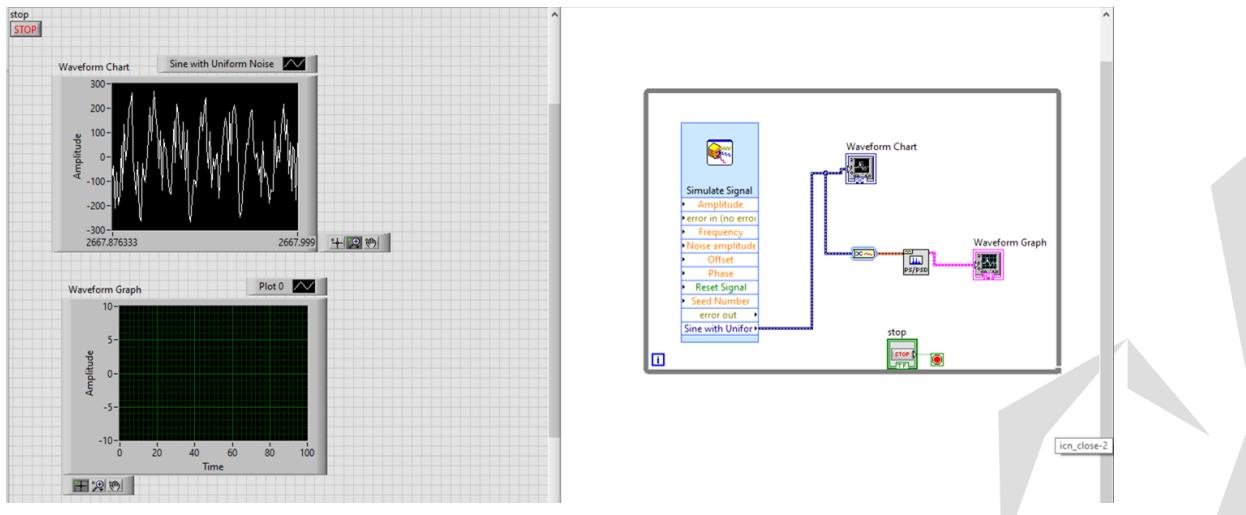
El error causado en el programa se ve por el tache que aparece. Por eso borraré ese bloque, no me sirve.



Block Diagram - FFT Power....: Dominio de Frecuencia de una Señal (Análisis de Fourier)

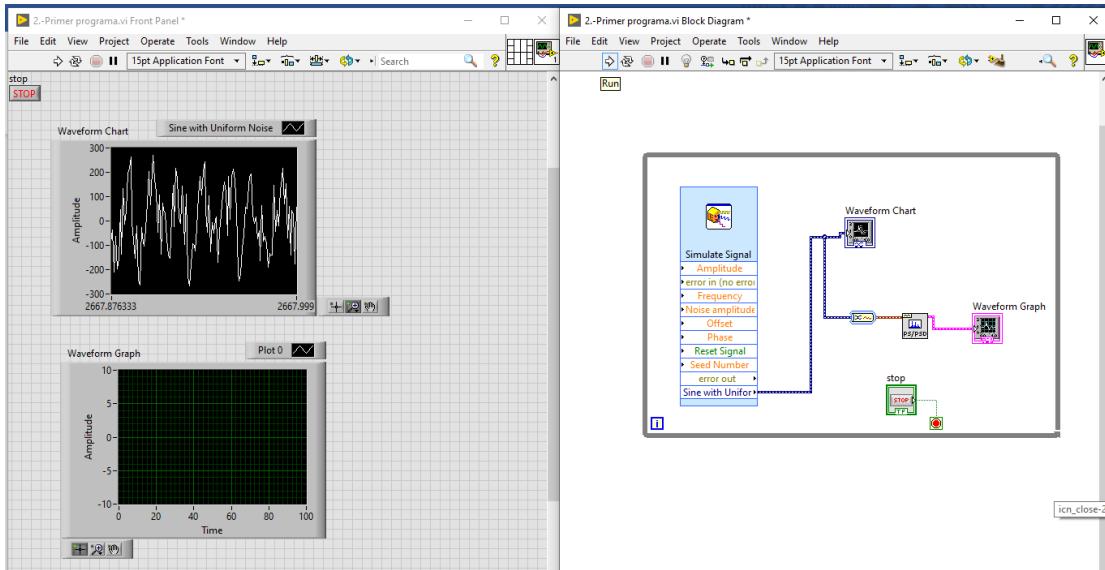
Con el bloque de análisis de Fourier **FFT Power Spectrum and PSD** obtendré las frecuencias del ruido incluido en la señal, esto porque primero con el bloque DDT se realiza la conversión de un tipo de dato Dynamic Data (señal real o virtual) a un Array (grupo de datos numéricos) y de esta manera se puede analizar cuáles son todas las frecuencias que conforman a la señal, las cuales ya son datos discretos calculables debido a que pasaron por el bloque de análisis de Fourier de LabView, con esta información puedo filtrar la señal senoidal y quitarle el ruido con un filtro pasa bandas, ya que la frecuencia que se grifique como mayor en el dominio de la frecuencia es la que corresponde a la señal, lo demás es el ruido. Además, debemos tomar en cuenta porque se utilizaron dos bloques distintos de graficación para observar las dos señales en el Front Panel:

- **Waveform Chart:** Grafica señales de tipo Dynamic Data, las cuales pueden ser señales reales o virtuales.
- **Waveform Graph:** Grafica Grupos de datos numéricos cualquiera, de tipo Array o Cluster.



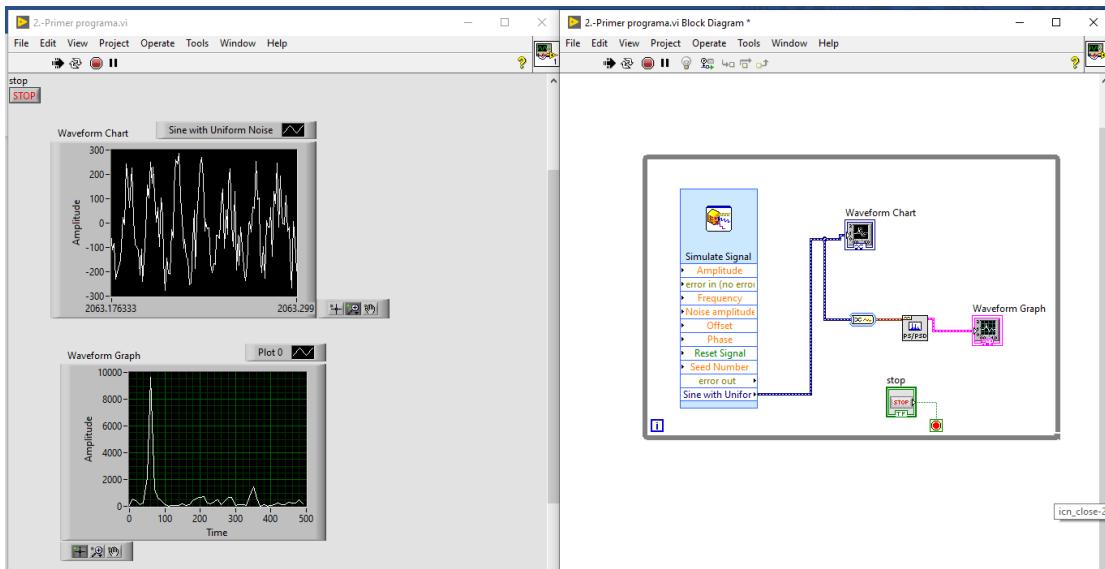
Ejecución del Programa: Gráfica de la Señal y su Dominio de Frecuencia

Al dar clic a run el programa se verá así.

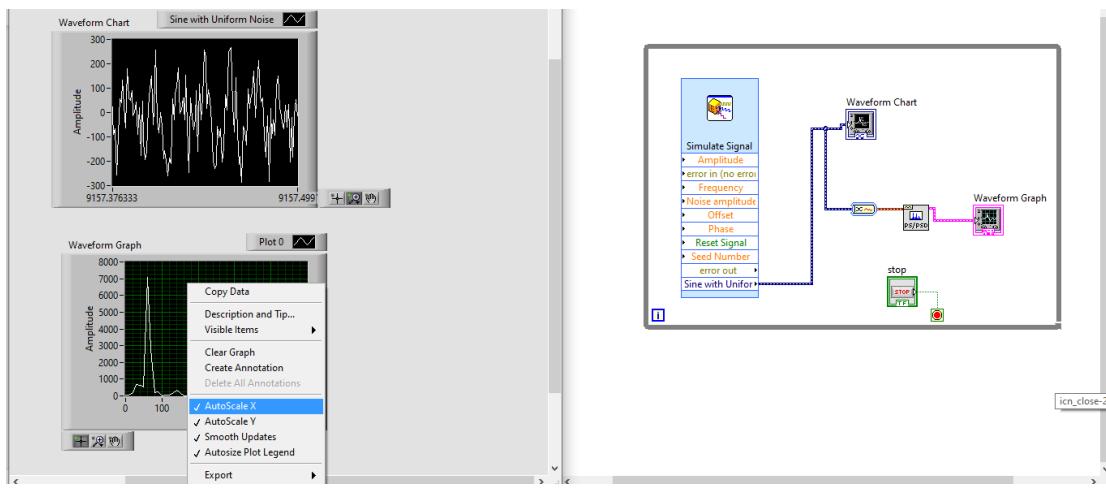


Con eso ya tengo mi señal en el dominio de la frecuencia.

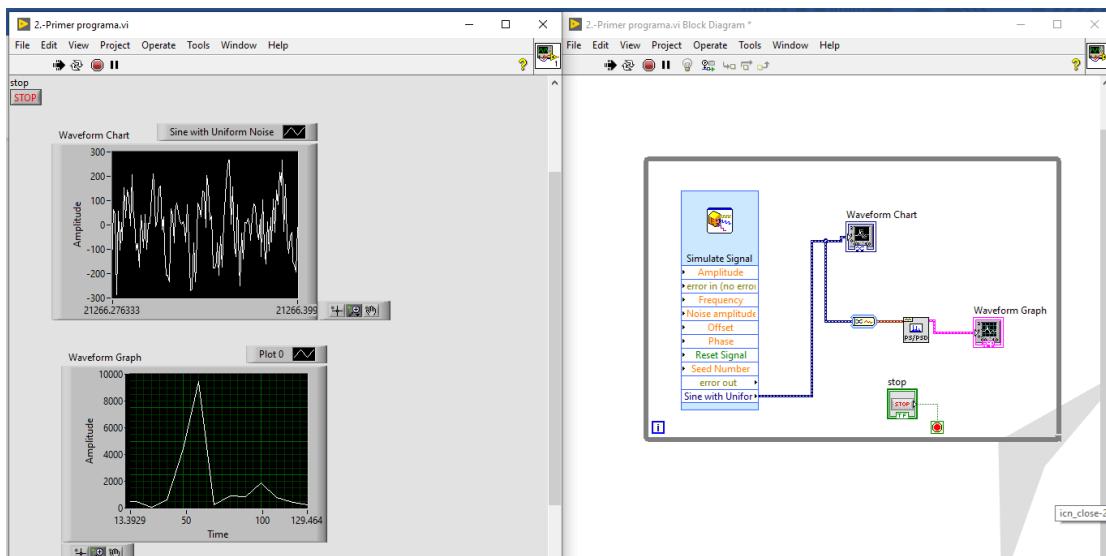
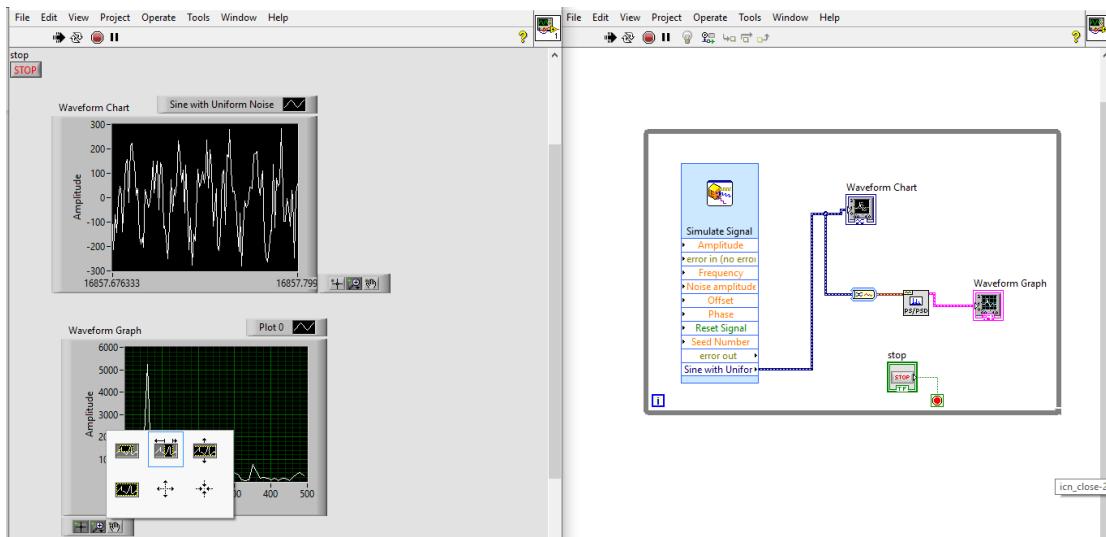
Con esta información puedo quitar la frecuencia del ruido con un filtro pasa bandas (que solo deja pasar un rango de frecuencias después del filtro), ya que la frecuencia que más se ve en el dominio de la frecuencia es la que corresponde a la señal y lo demás que se ve es el ruido.



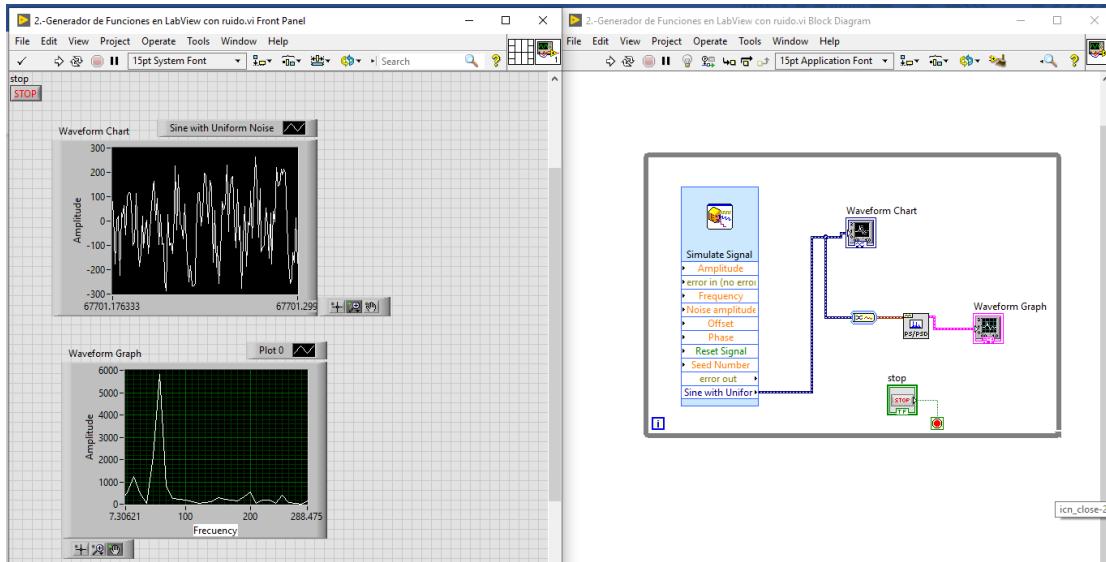
Para poder hacer zoom le voy a quitar la autoescala.



Y aplicaré un ZOOM.

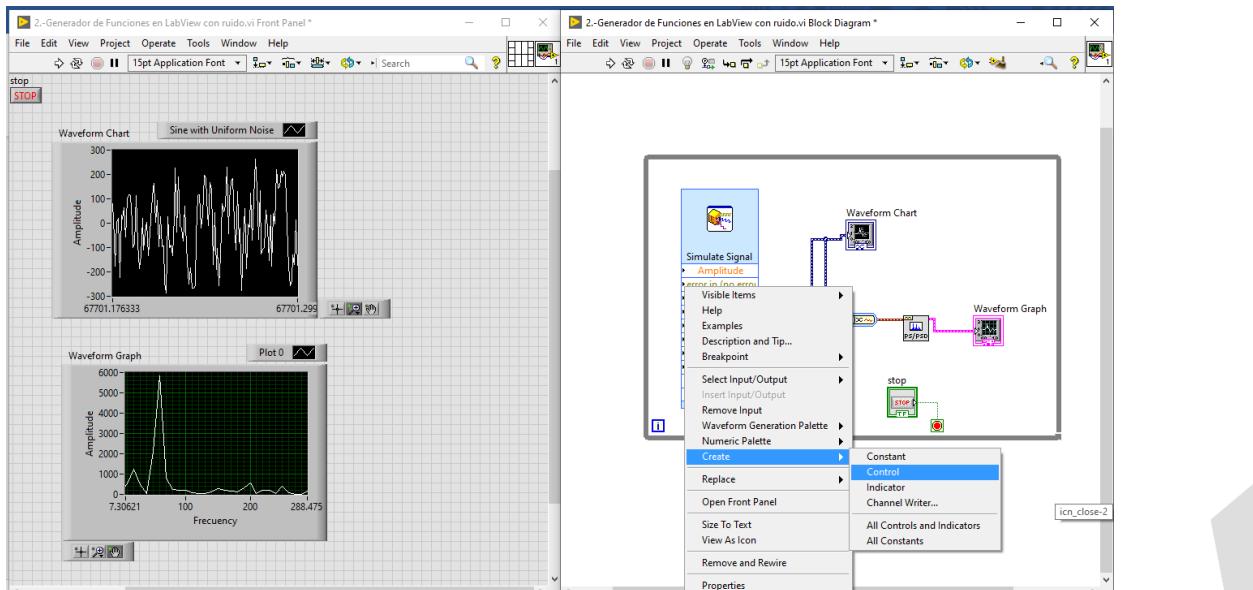


En el eje horizontal va a decir siempre time por default en el waveform chart, pero en sí porque aplicamos una transformada de Fourier, el eje horizontal representa frecuencia, esto se lo puedo cambiar dando clic en el eje horizontal.

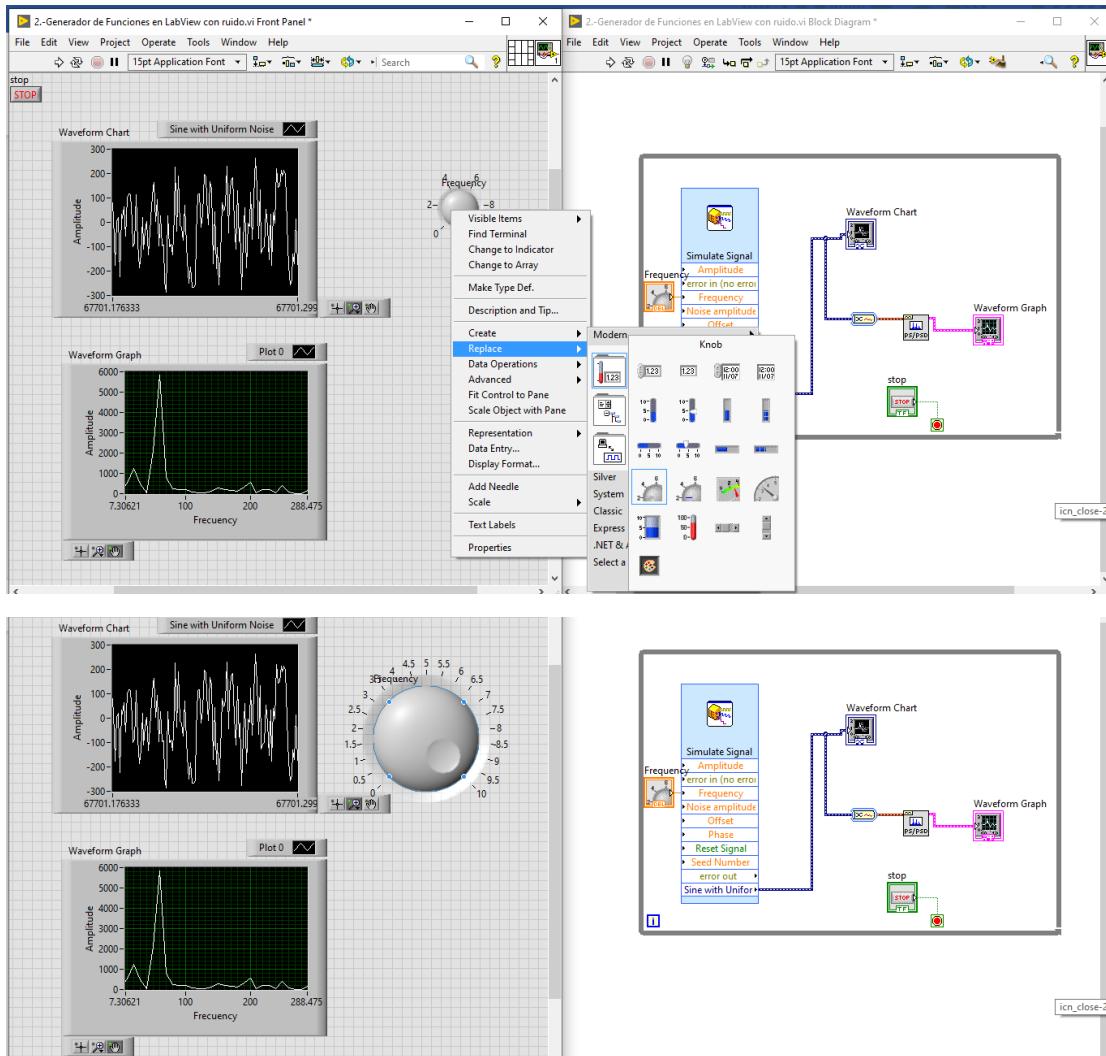


Block Diagram - Simulate Sig: Agregar un Control para la Frecuencia

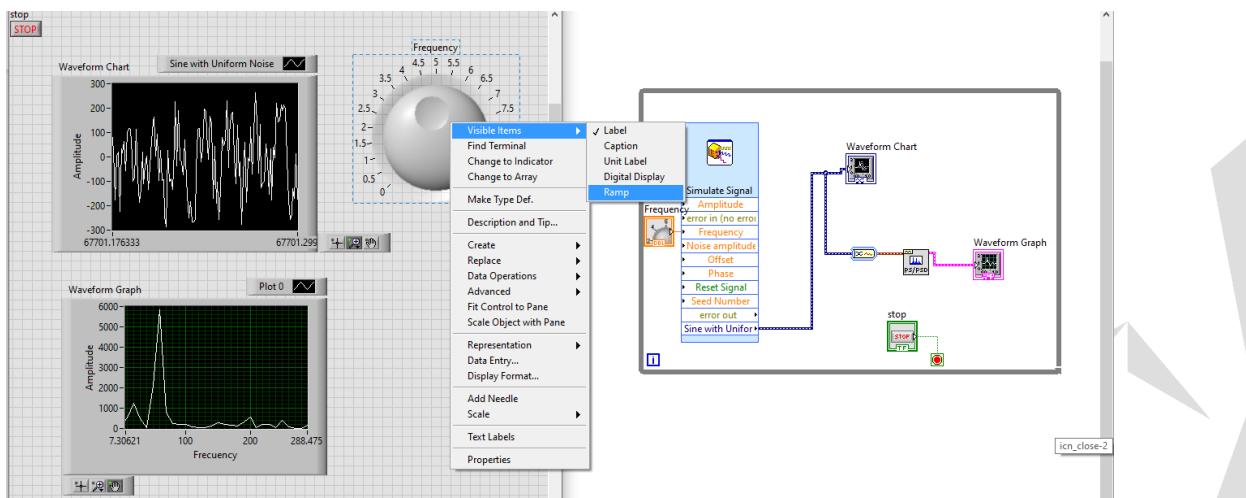
Lo que deberíamos hacer para quitar el ruido es solo dejar el armónico, osea la frecuencia de 60 Hz que es donde está el pico y quitar las demás frecuencias.



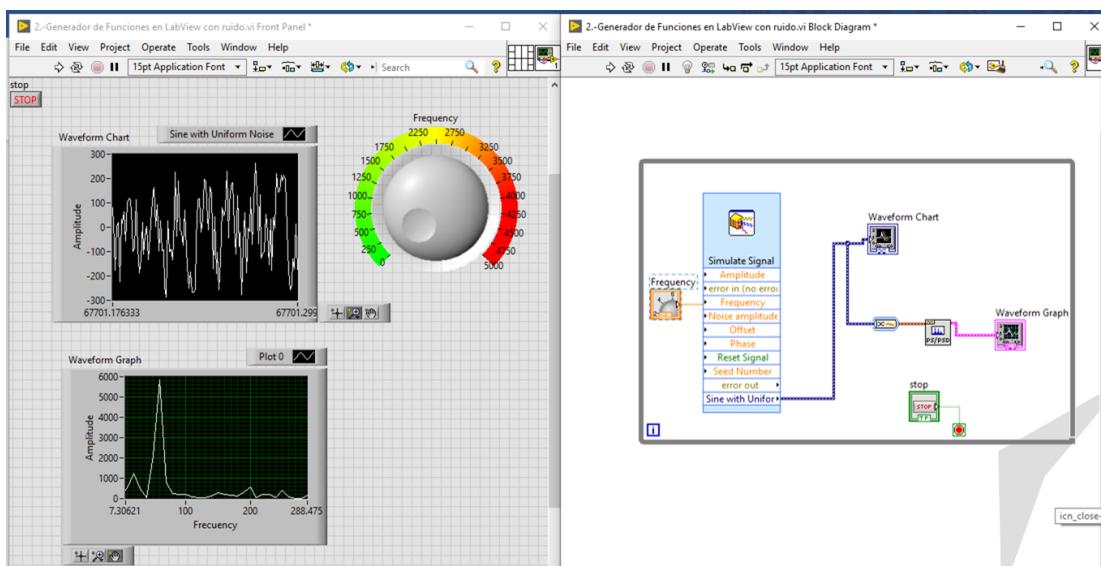
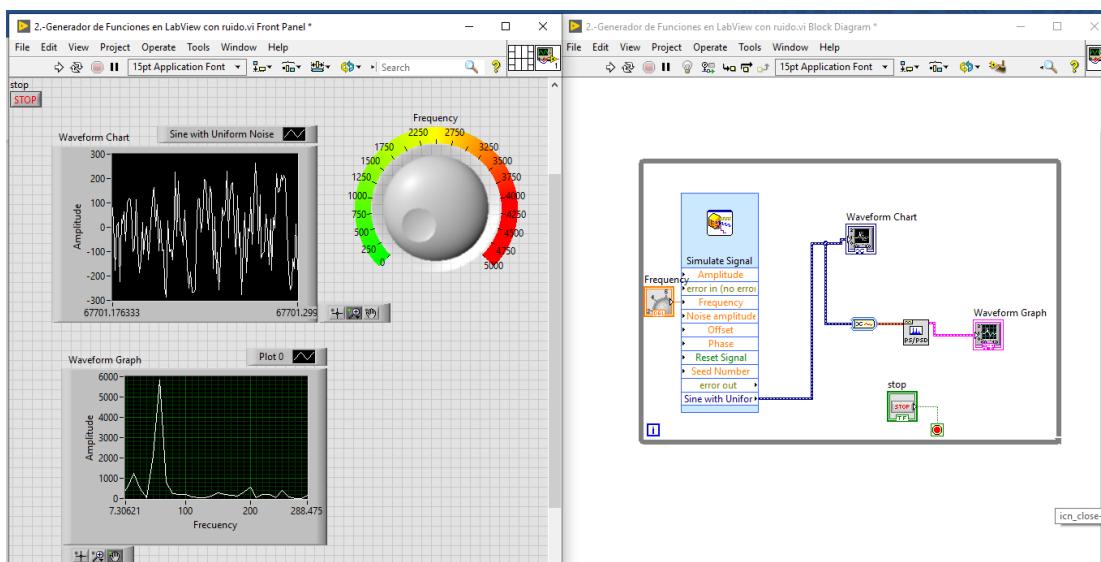
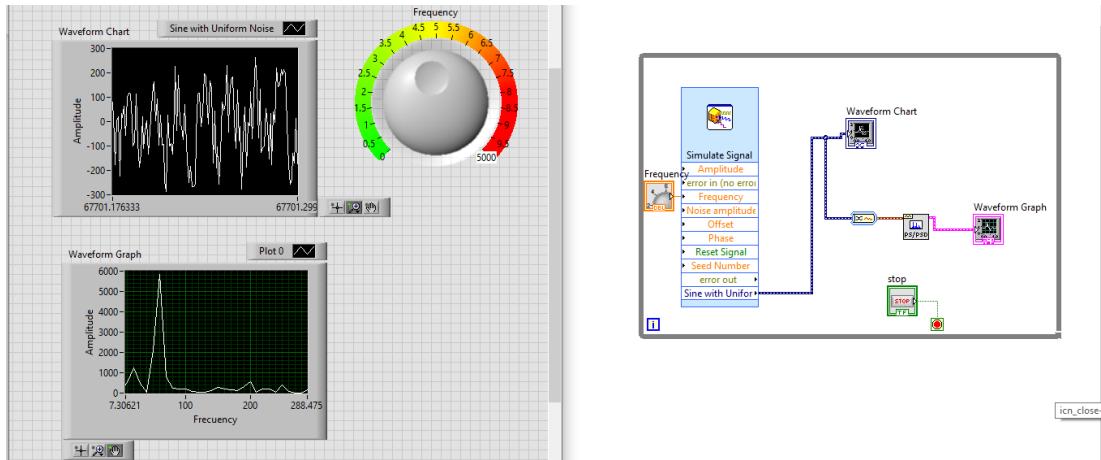
Front Panel - Replace - Knob; Poner una Interfaz de Control en Forma de Perilla



Si doy clic derecho en la interfaz de control y selecciono la opción de Visible Terms → Ramp, los valores de la perilla se pondrán de menor a mayor de color verde, pasando por amarillo y luego de rojo.

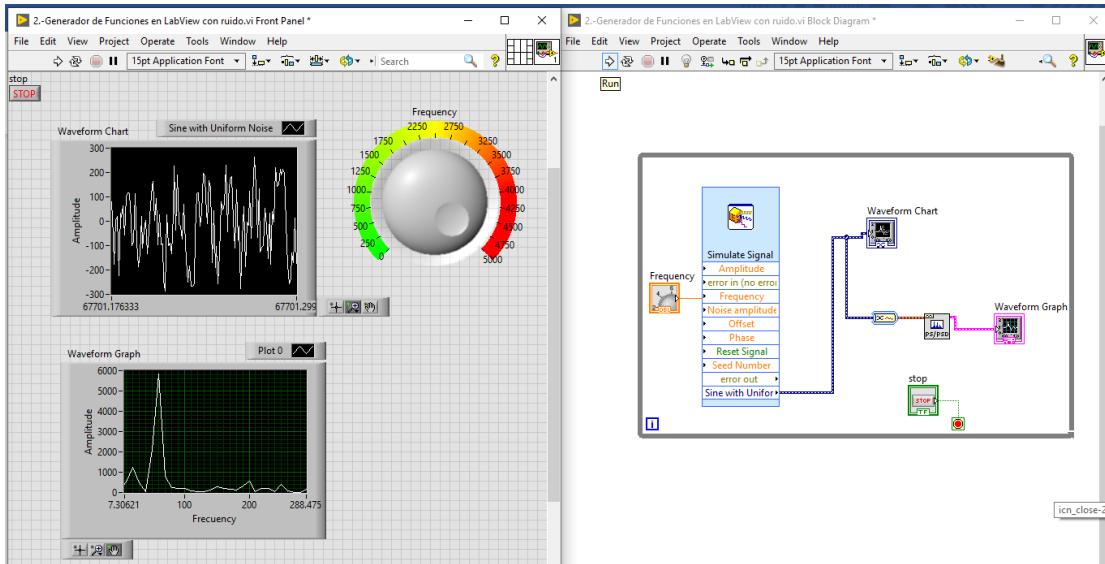


Además, puedo cambiar su valor mínimo y máximo dando doble clic en el valor de su número y colocando el nuevo valor.

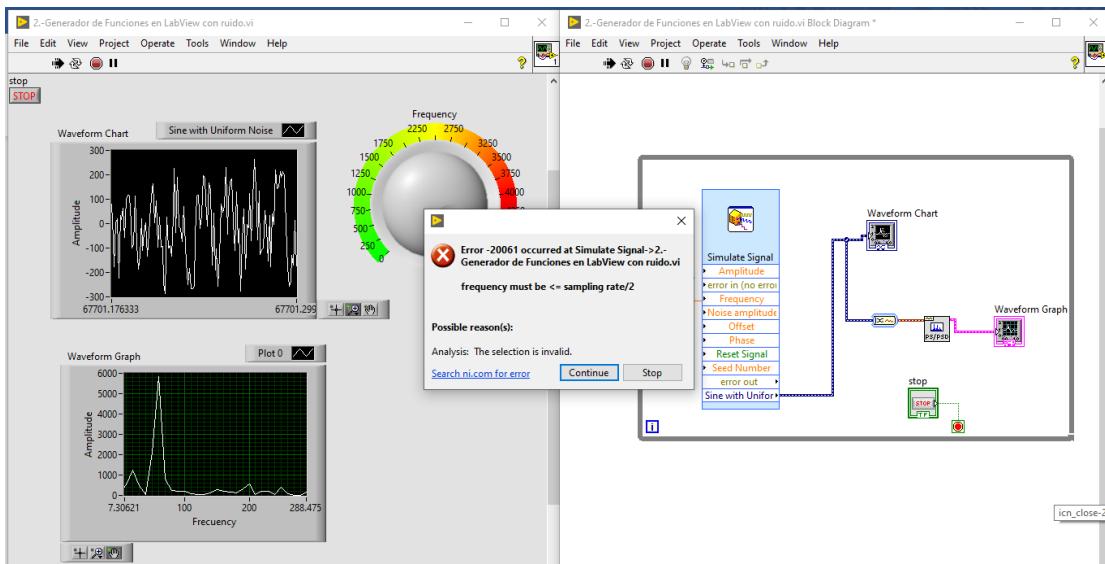


Ejecución del Programa: Error al Subir Mucho la Frecuencia de la Señal

Si subo la frecuencia a 5000 y lo corro sale un error y se para la ejecución del programa.

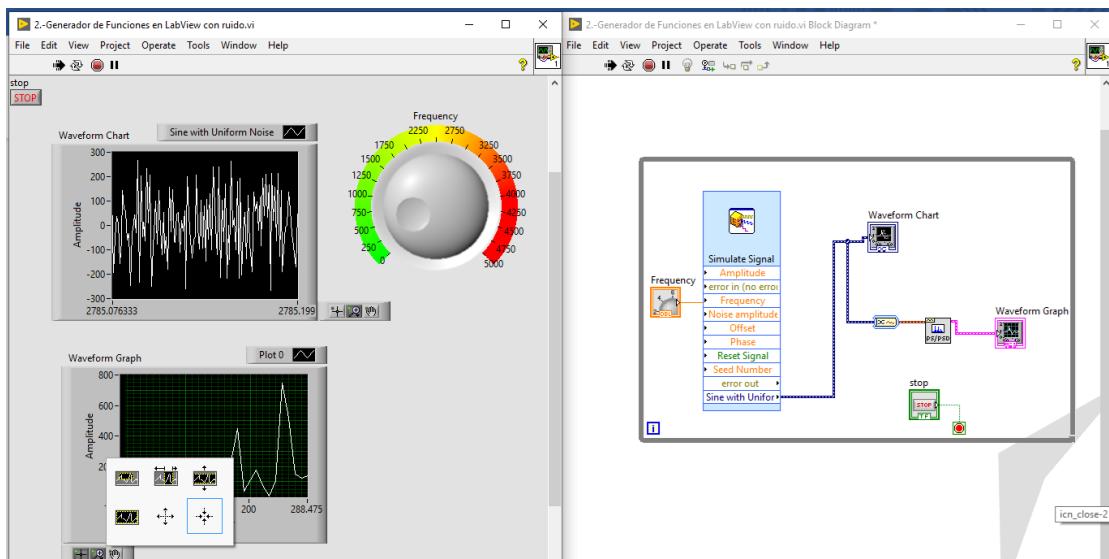
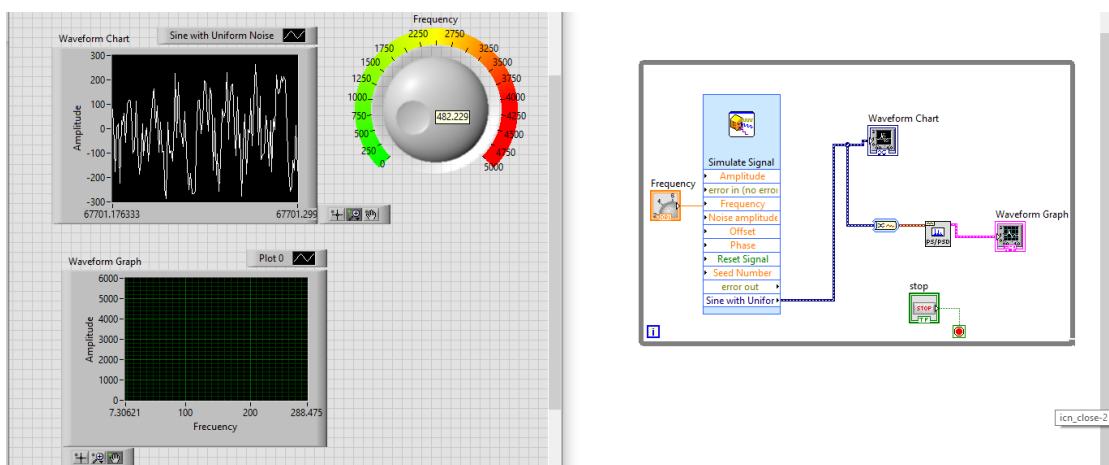
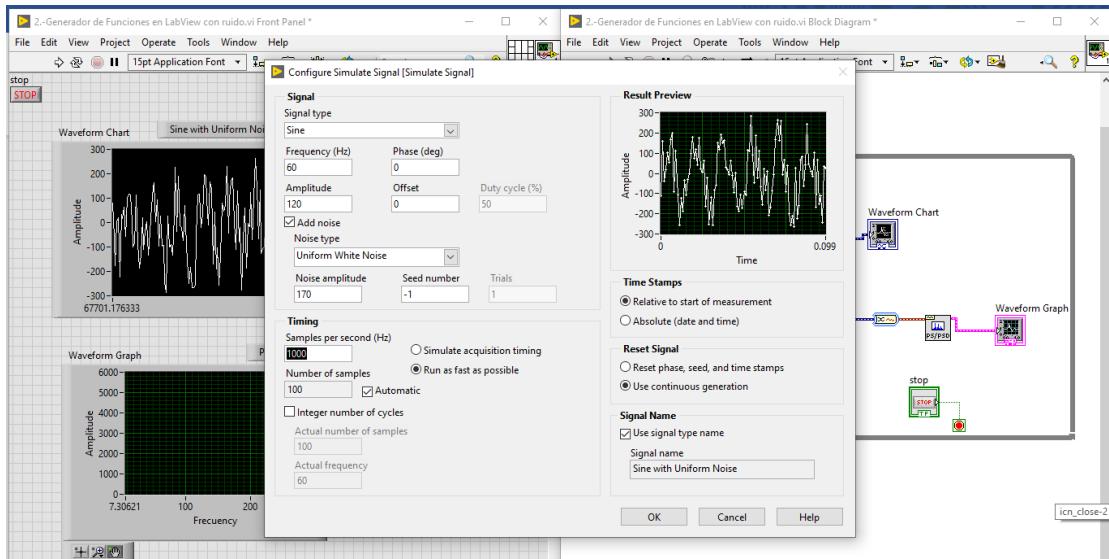


Esto pasa por el teorema de muestreo, en frecuencia de 1000 también marcará error, en 750 igual, en 500 ya corre, pero si pongo frecuencia de 501 Hz marcará error, esto pasa porque la frecuencia es mayor a la tasa de muestreo entre 2.

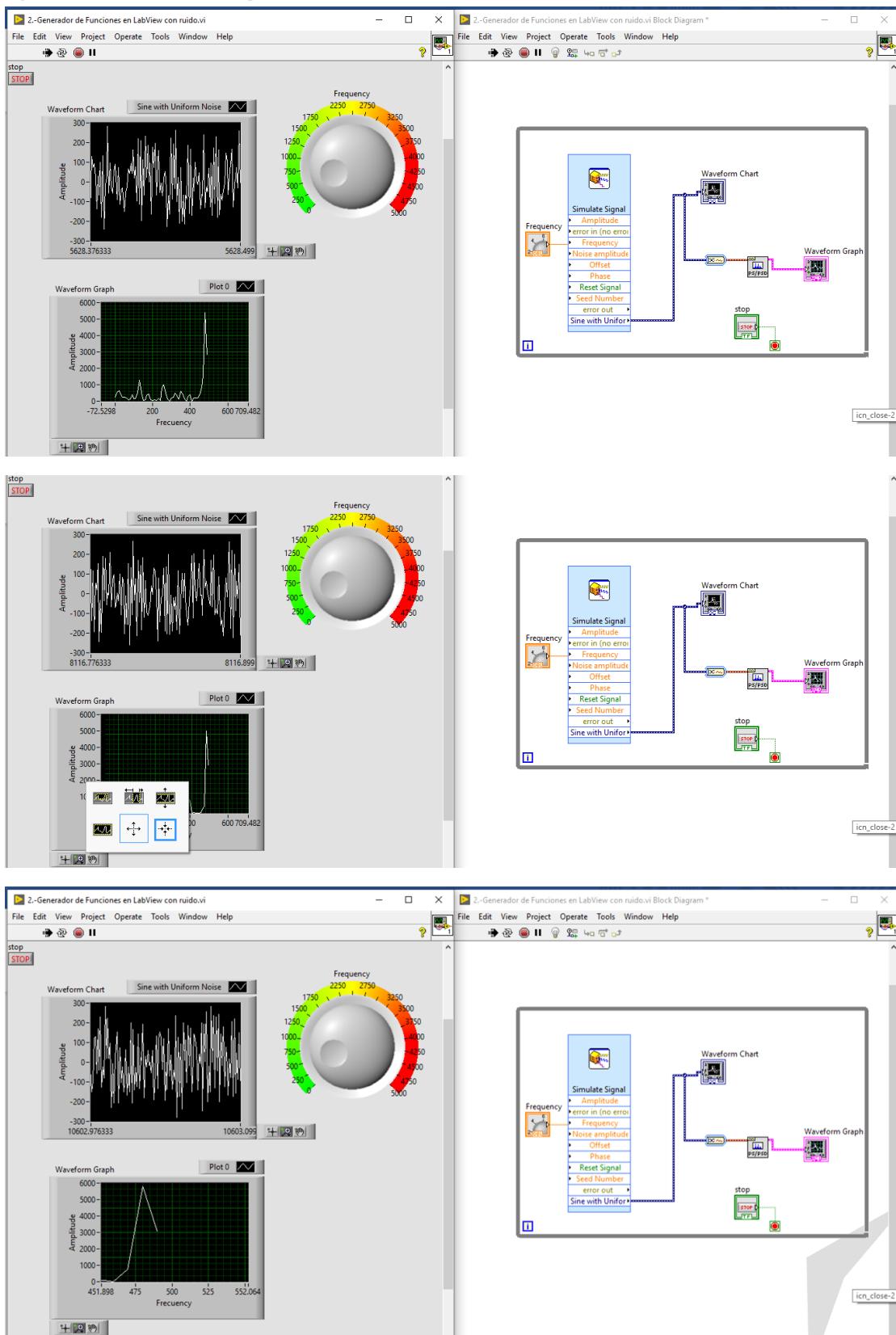


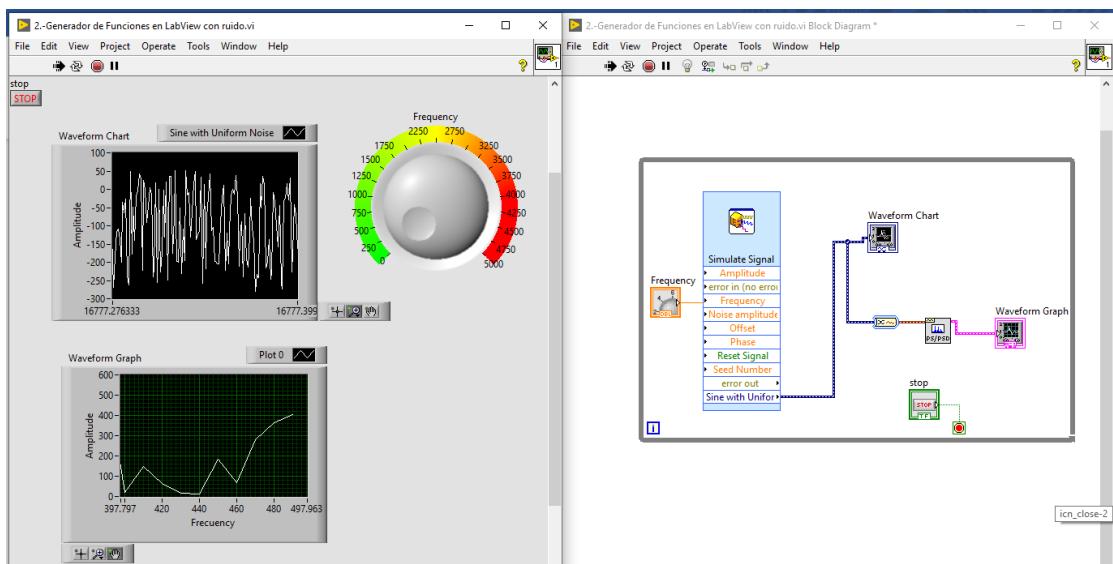
Recordemos por el teorema de Nyquist que para que una señal pueda ser leída correctamente, su frecuencia de muestreo debe ser mínimo 2 veces mayor a la frecuencia de la señal, por eso es que el programa da error cuando se sube demasiado su frecuencia.

Eso de la frecuencia de muestreo entre 2 se ve dando doble clic en el primer bloque que pusimos donde dice Samples per second.



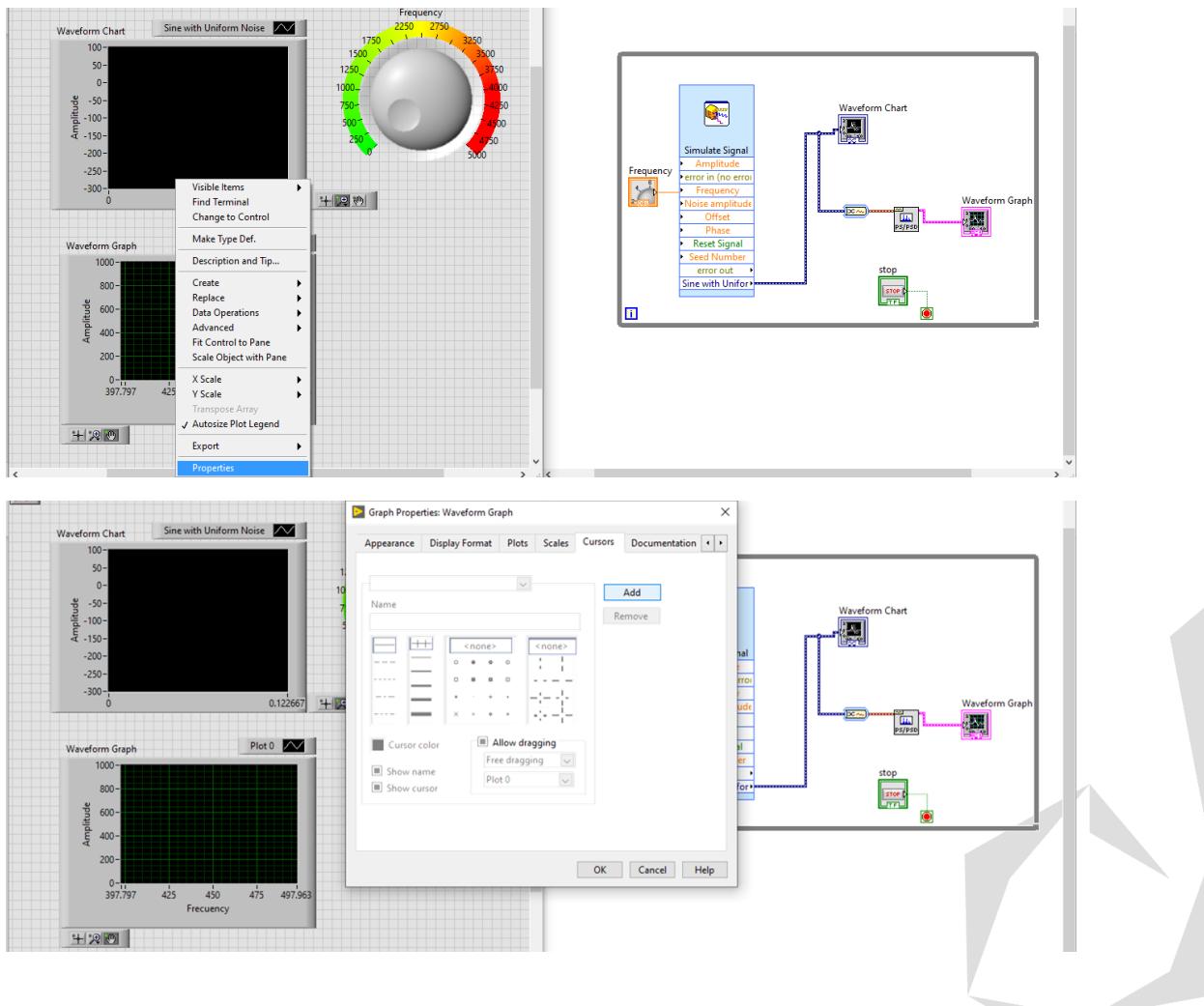
Ejecución del Programa: Sin Error con Frecuencia \leq 500 en el Control



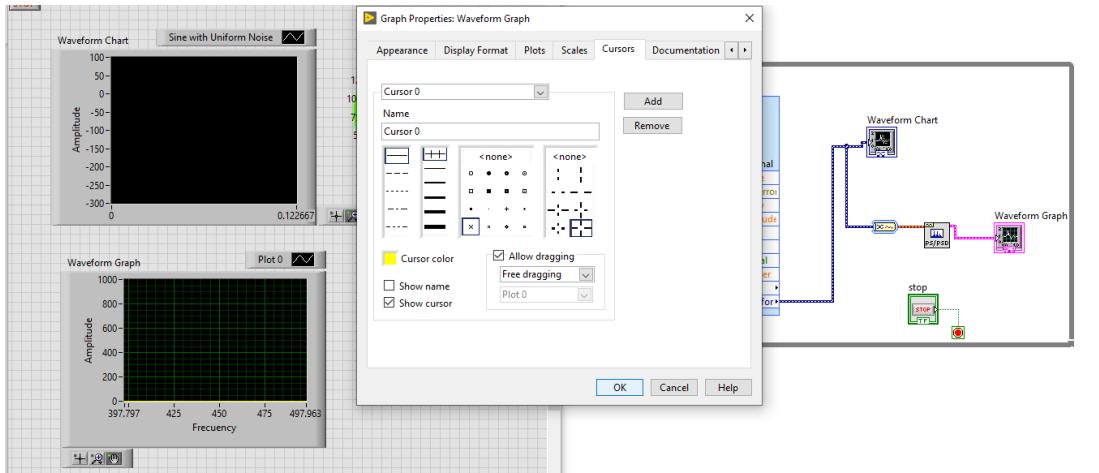


Front Panel - Waveform Chart: Agregar Cursores

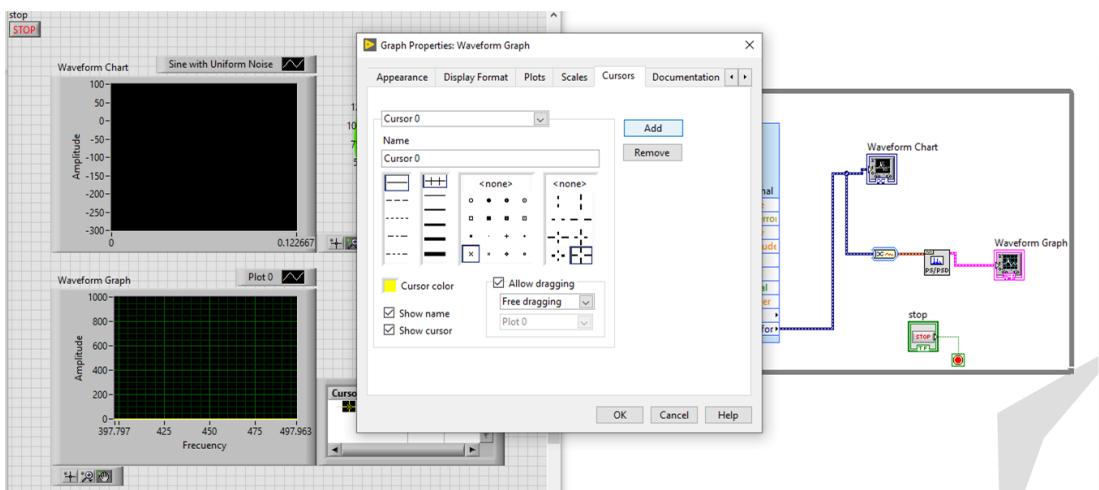
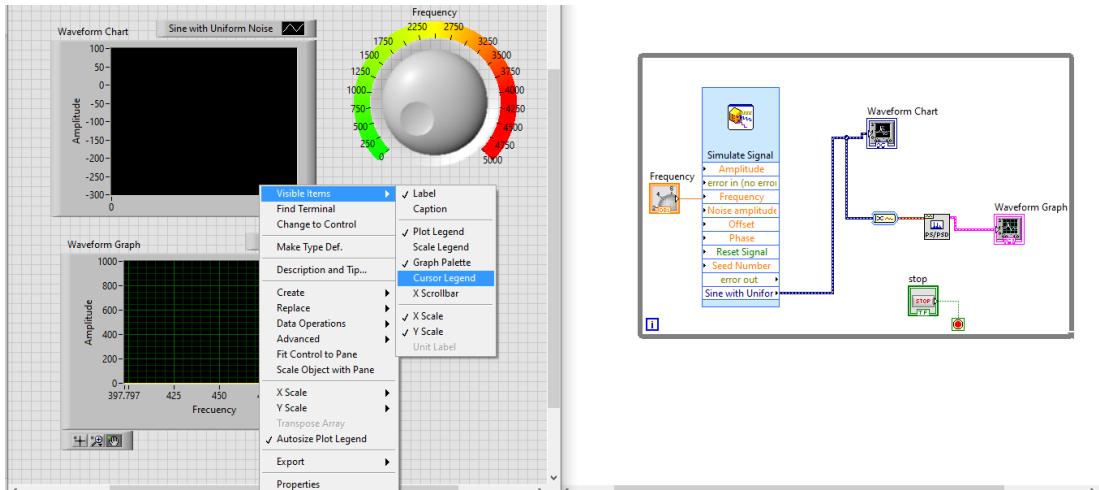
Para agregar cursores (líneas verticales u horizontales de medición) lo que se hace es dar clic derecho sobre el bloque de la interfaz, ingresar a la opción de Properties y realizar lo siguiente:



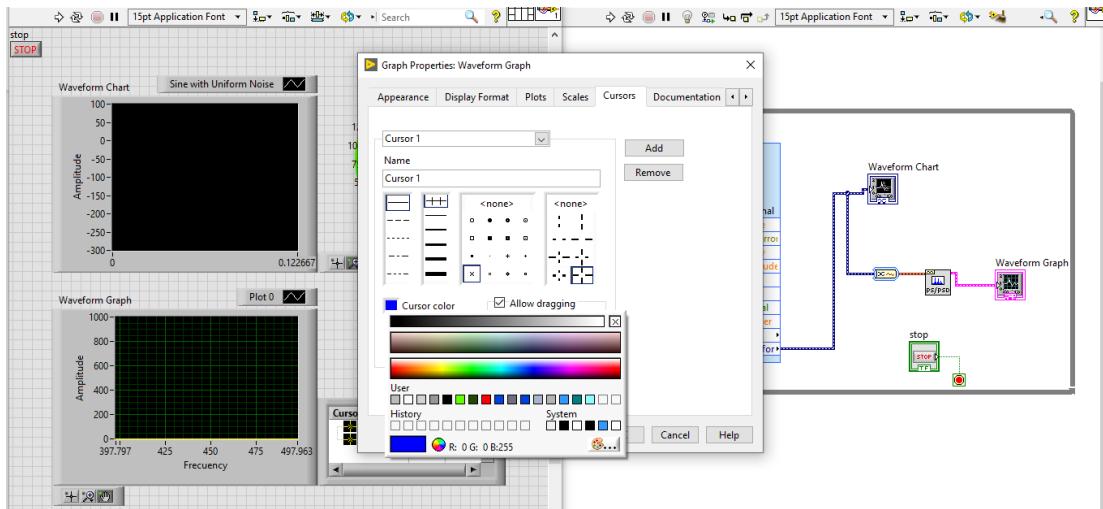
Se debe indicar el nombre, tipo de línea y punto color. También se debe seleccionar su color y seleccionar las checkbox de Allow dragging y Show cursor.



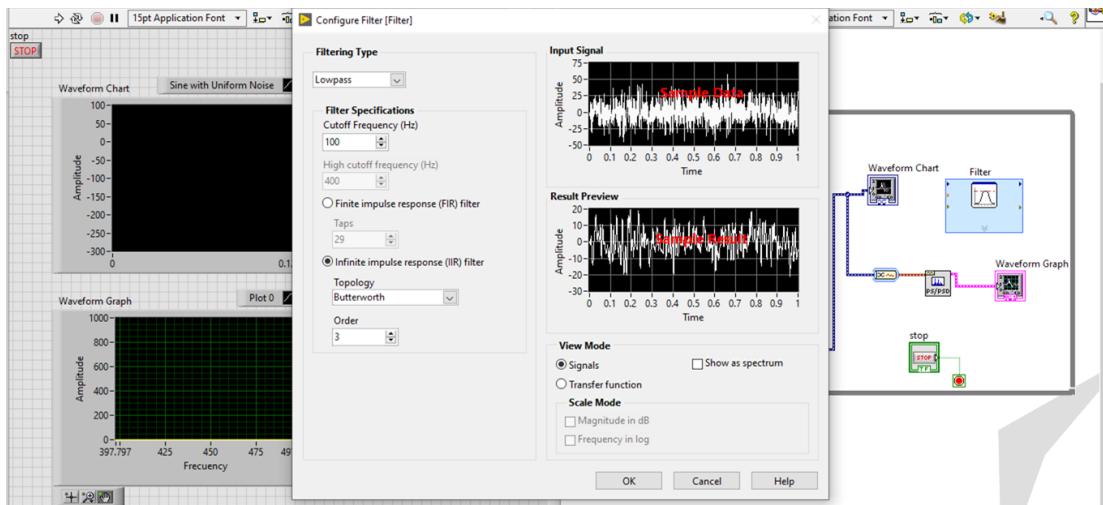
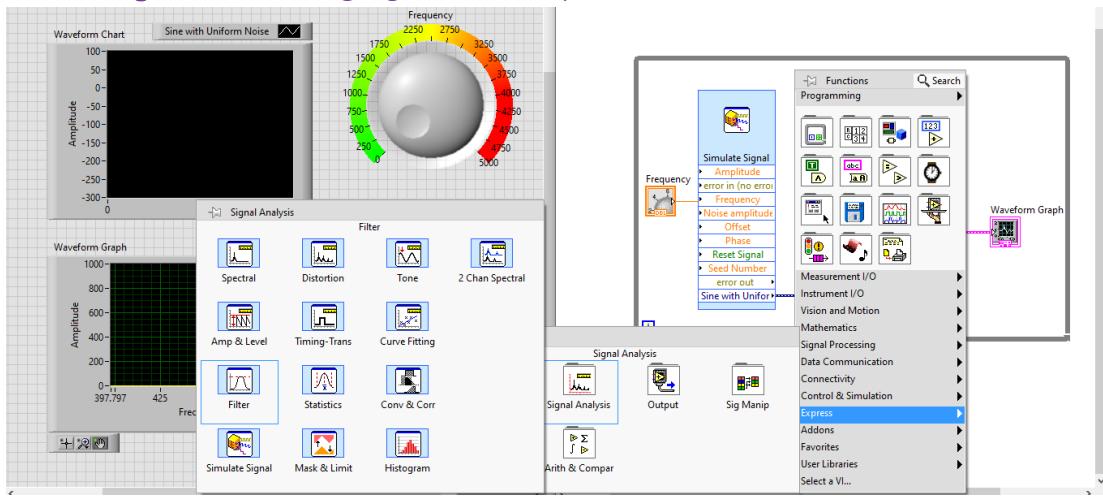
Al dar clic derecho de nuevo en el bloque de Waveform Chart y seleccionar la opción de Cursor Legend, aparecerá el cursor cuando se ejecute el programa.



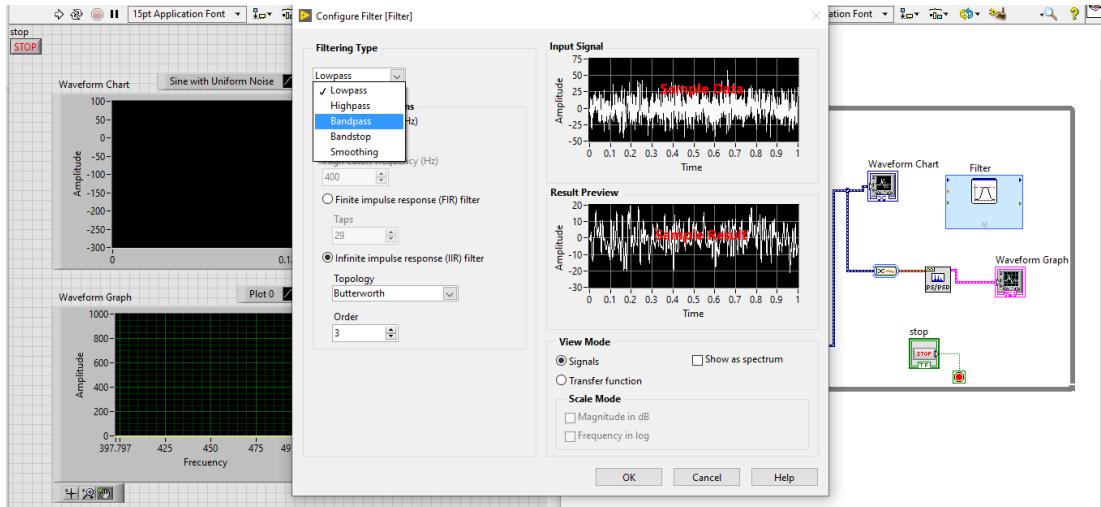
Lo mismo se repite para agregar el otro cursor, ya que mínimo se deben tener 2 para poder realizar mediciones dentro de la gráfica.



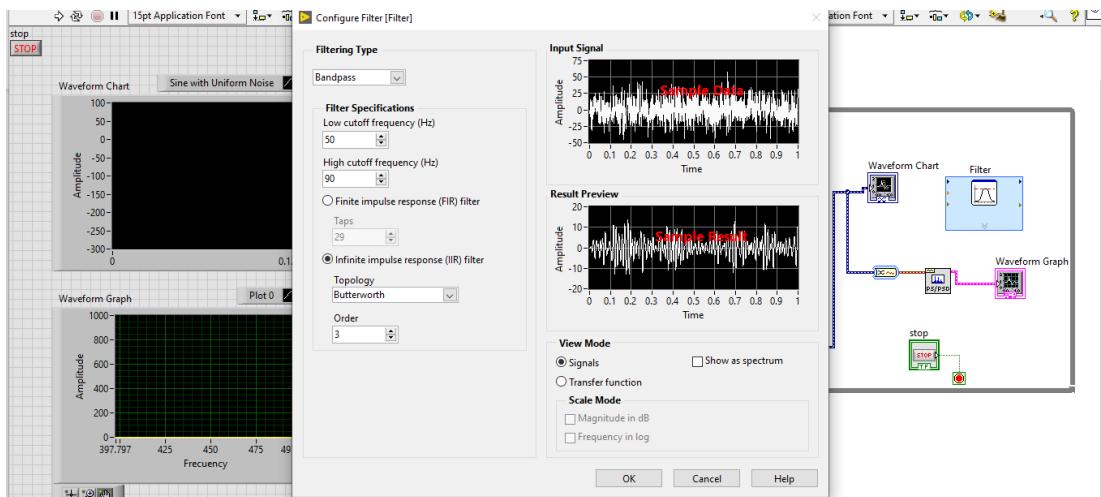
Block Diagram - Filter: Agregar un Filtro para Quitar el Ruido de la Señal



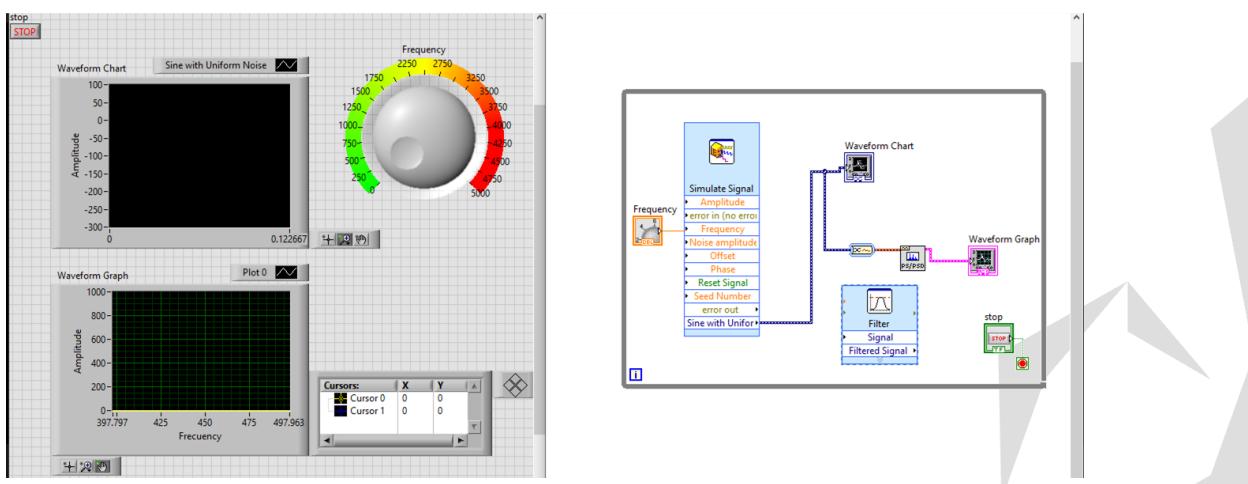
Dentro de la configuración del bloque se debe elegir un tipo de filtro, vamos a agregar un filtro pasa bandas (Bandpass), que solo permite pasar ciertas frecuencias que se encuentren en un rango descrito.

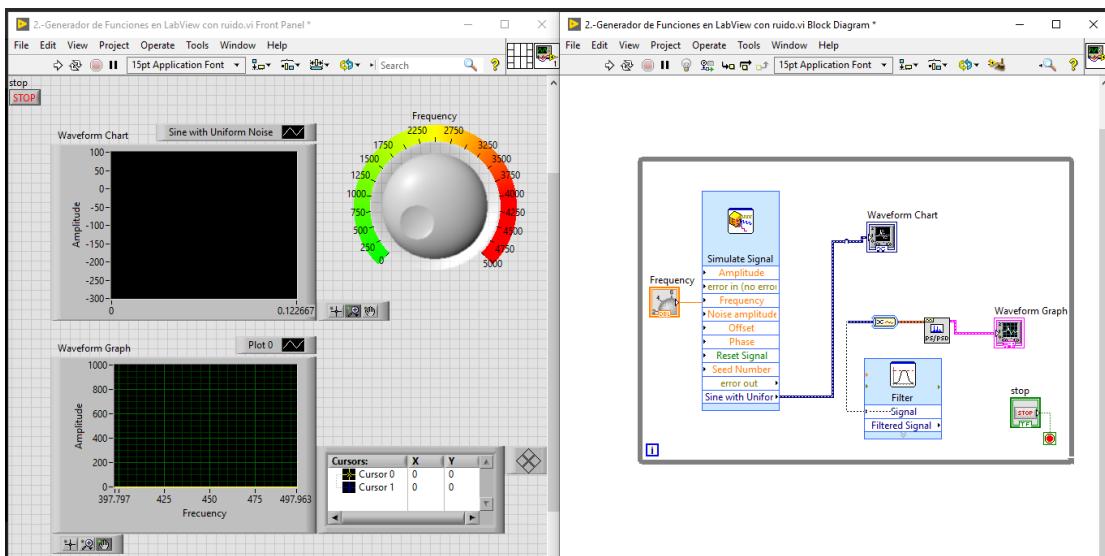


Frecuencias de corte de 50 a 90 Hz, usando un filtro Butterworth de segundo orden pasa bandas.

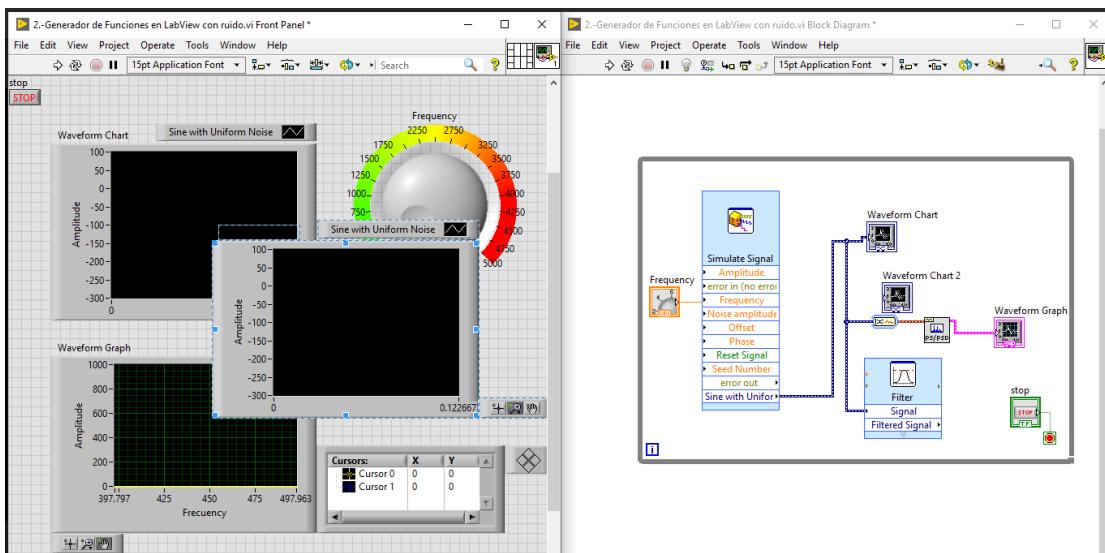


Moveré mis bloques para que quepan dentro del ciclo while





Voy a duplicar el Waveform Chart (Dynamic Data: Señales Reales o Virtuales) para que se vea la señal filtrada.

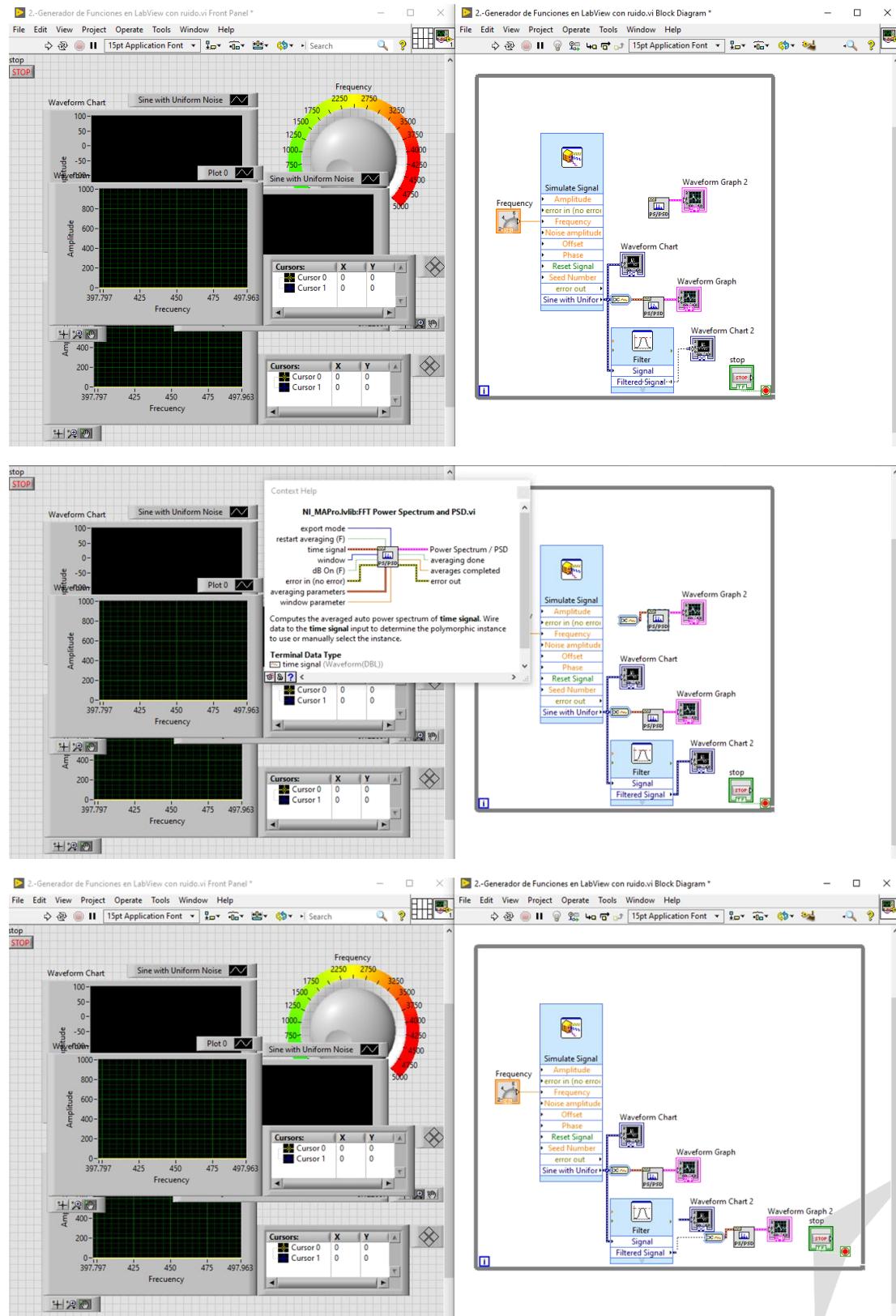


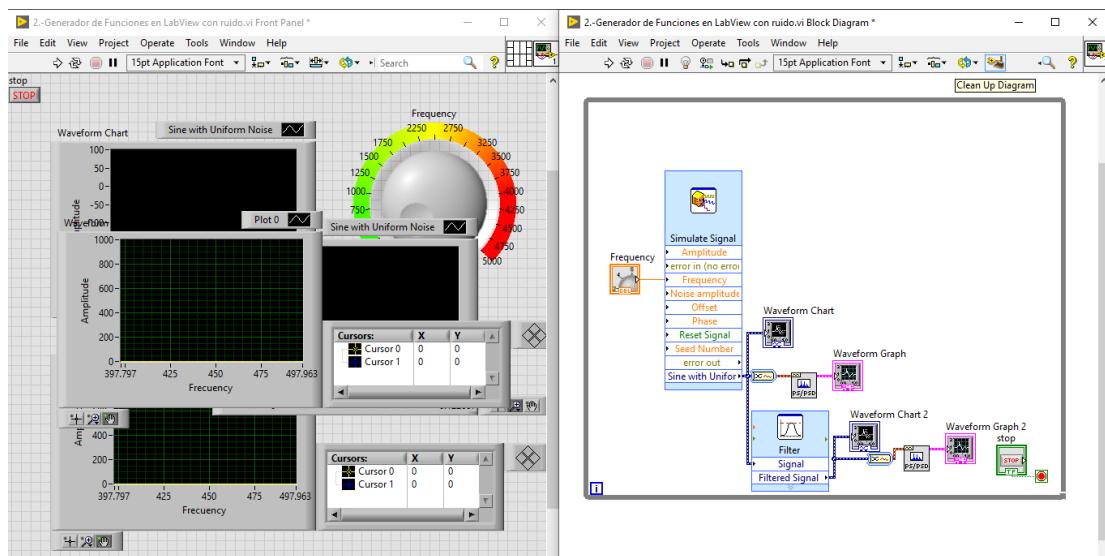
Duplicaré la parte del análisis de Fourier para ver el espectro de frecuencias armónicas después del filtrado en el nuevo diagrama de Waveform Chart.

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.

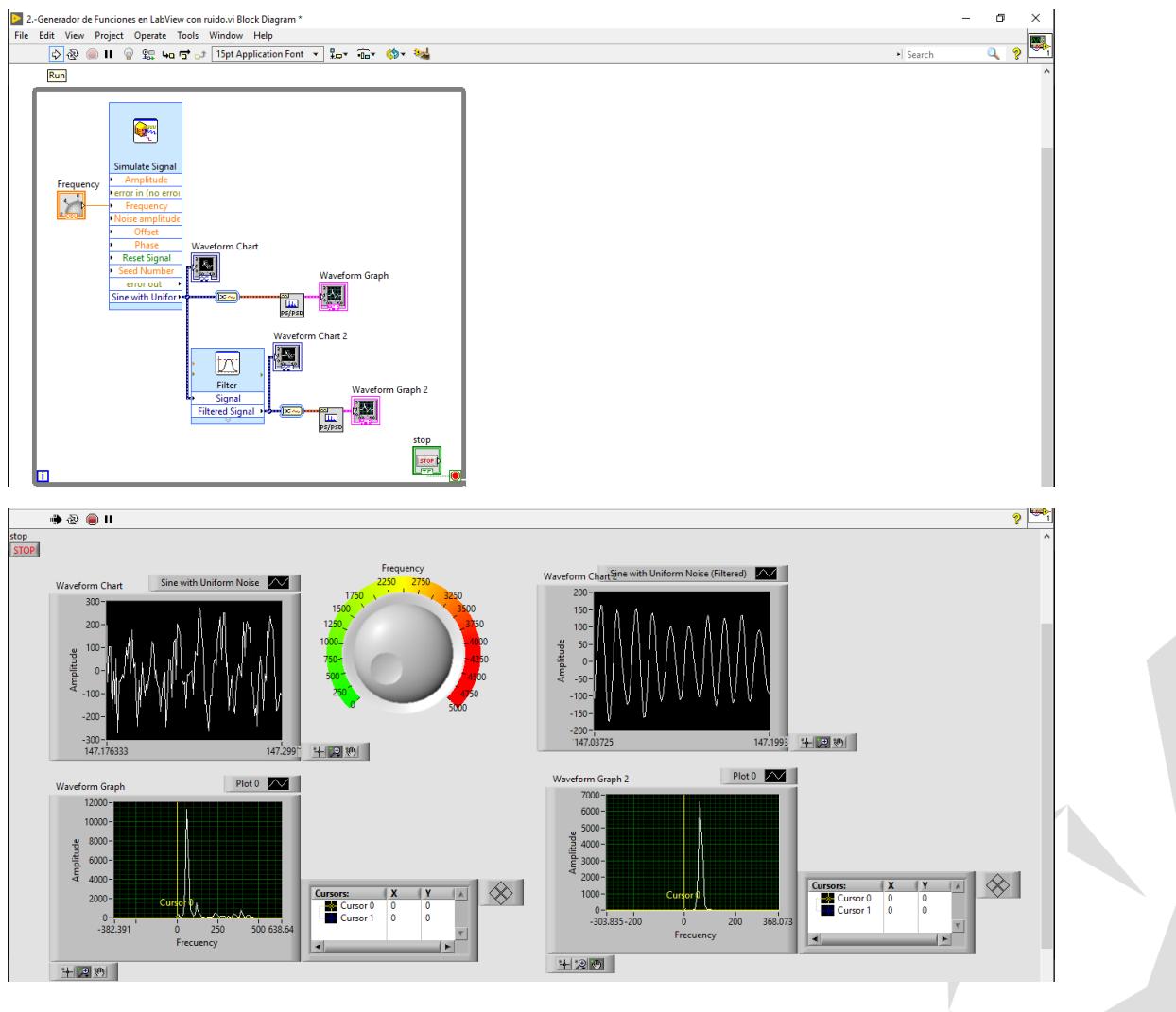
Aunque en teoría esto es lo que debería pasar, no siempre el botón de Clean Up Diagram hace correctamente el trabajo, o mínimo de la forma que el usuario quiere.

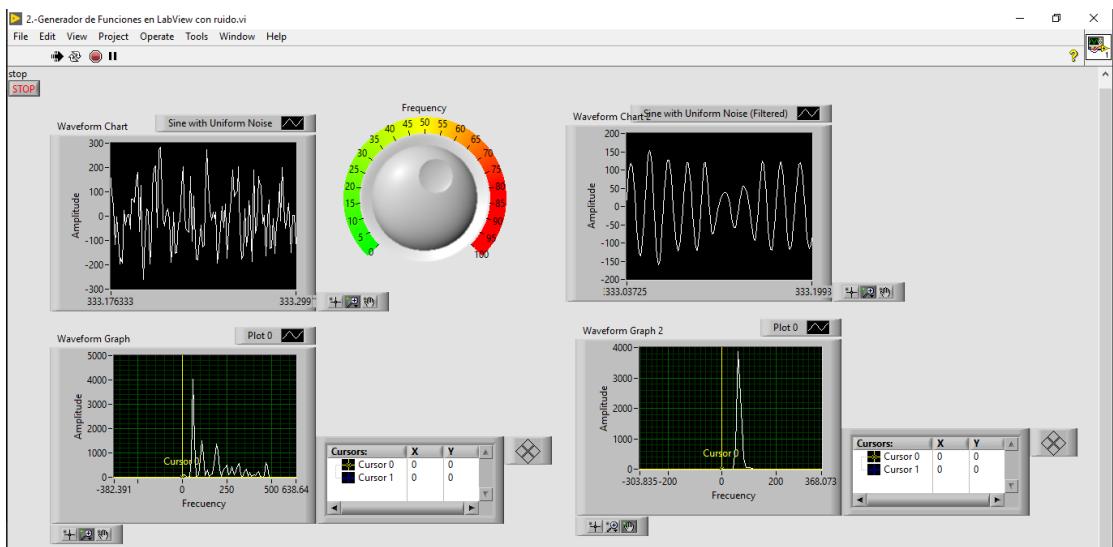




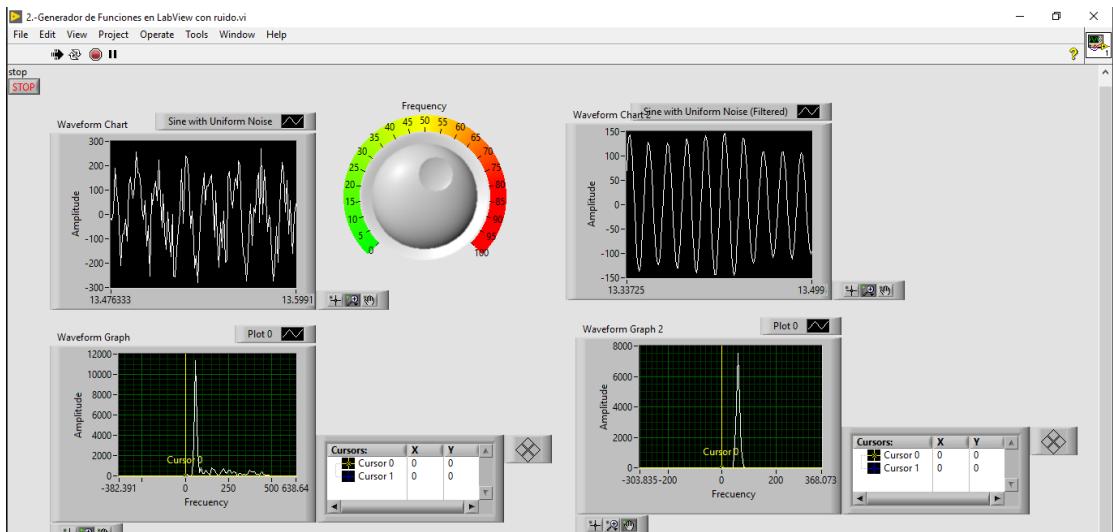
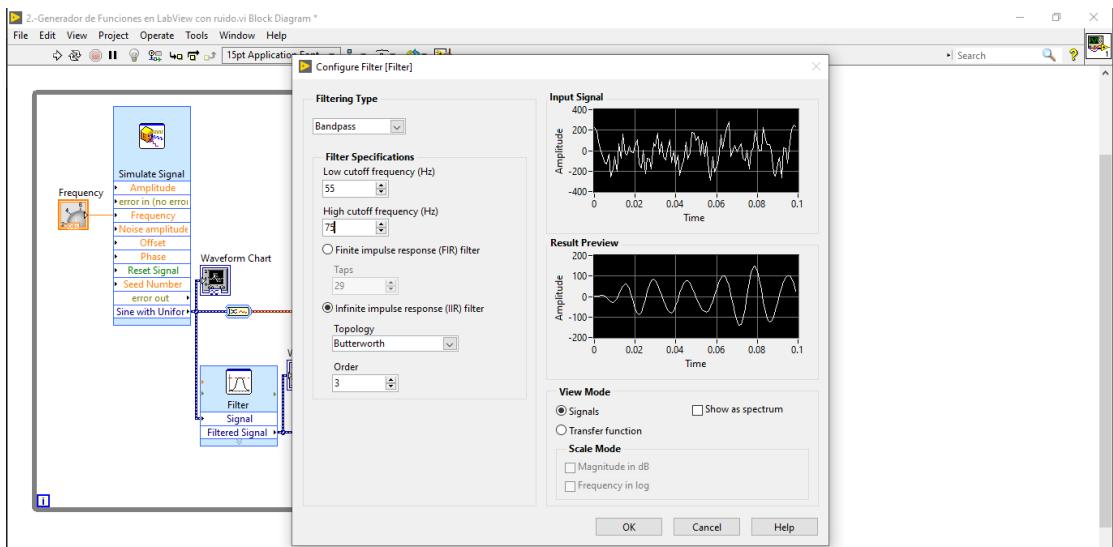
Ejecución del Programa: Gráficas de Frecuencias Con y Sin Ruido Filtrado

Las gráficas muestran la señal con cursores de medición, dominio de frecuencia, con y sin ruido filtrado.

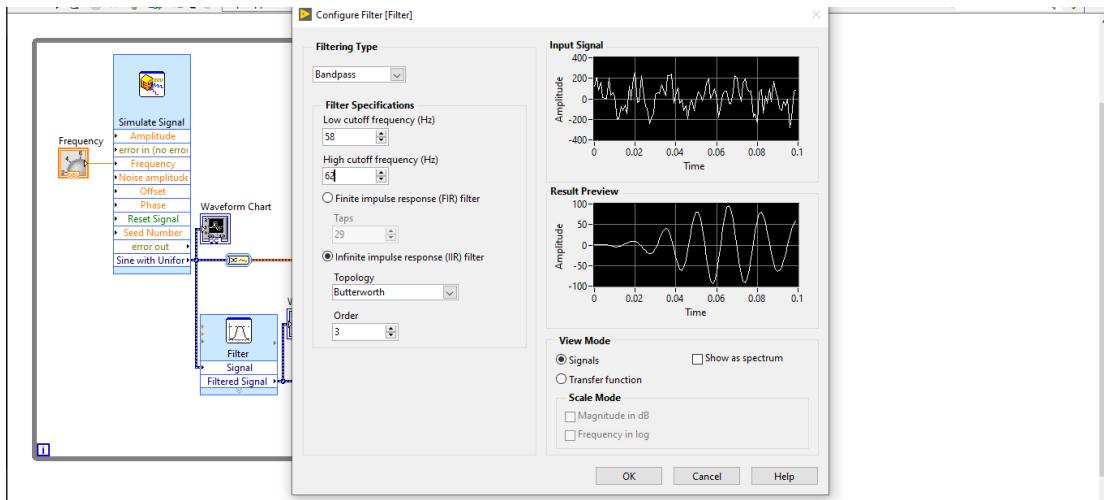




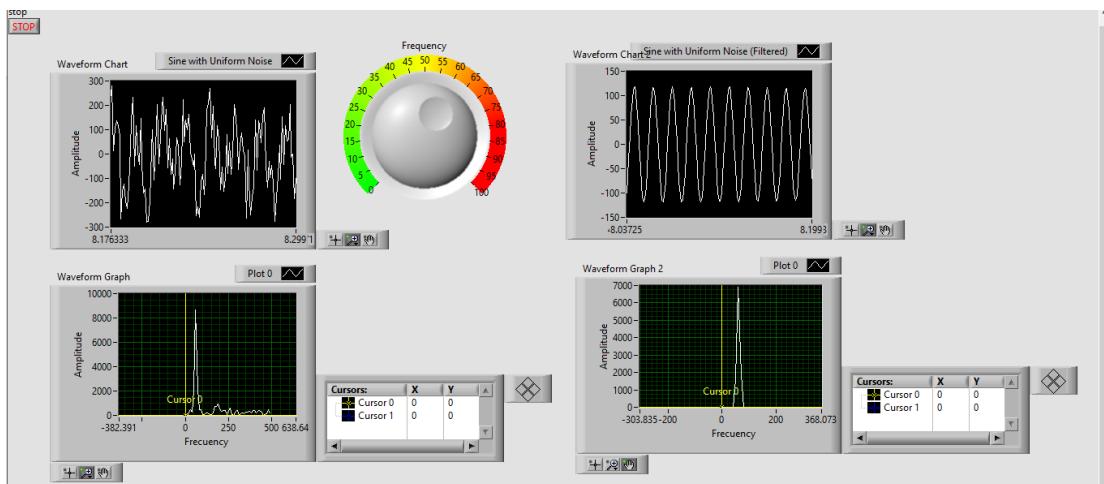
Si se cambian las especificaciones de la señal y/o el ruido, se puede reflejar su resultado en la ejecución.



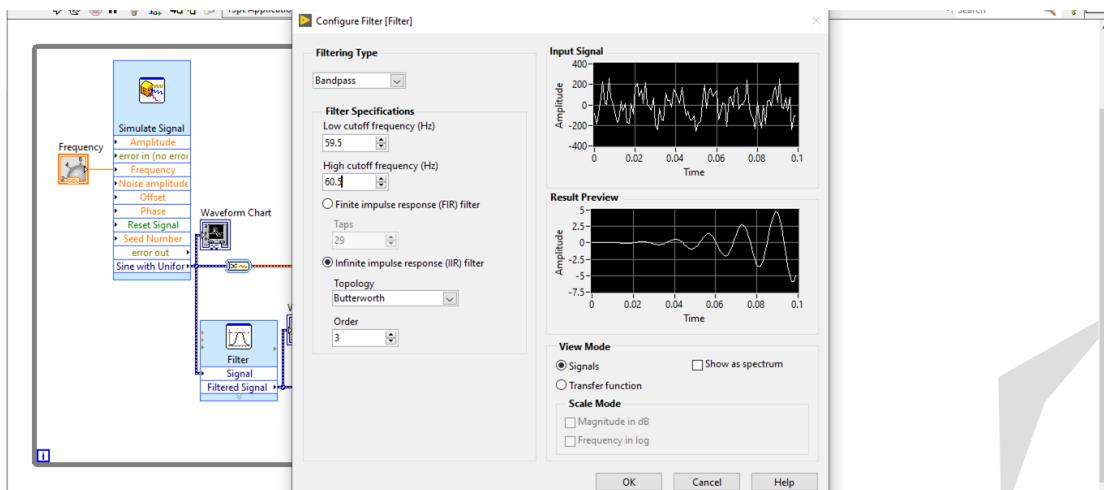
Todavía tengo armónicos que me hacen un poco inestable mi señal, por lo que debo seguir cambiando el filtro hasta que ejecute de forma correcta el filtrado pasa bandas del ruido en la señal.

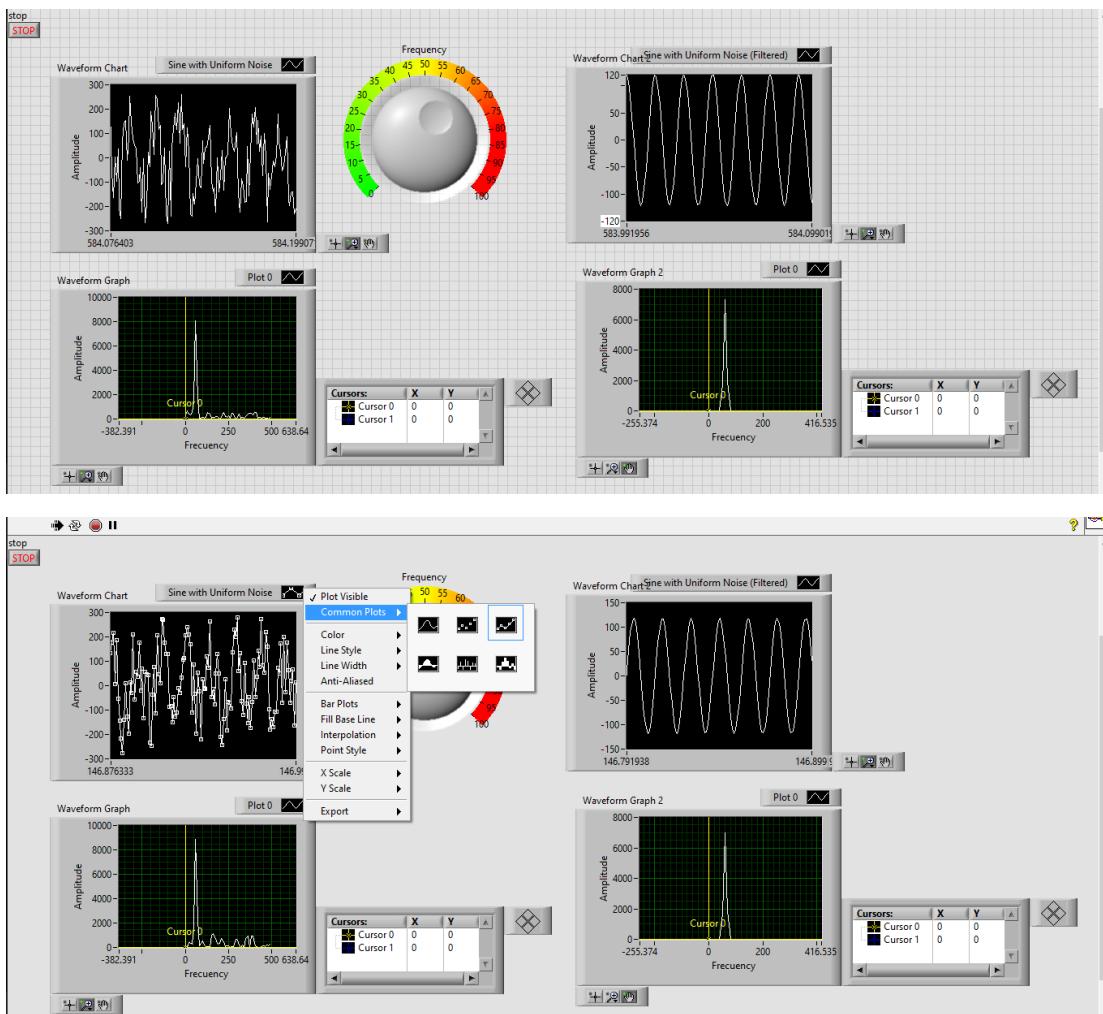


Con este filtro de 58 y 62 Hz deja pasar todavía algunas frecuencias que me están deformando la señal



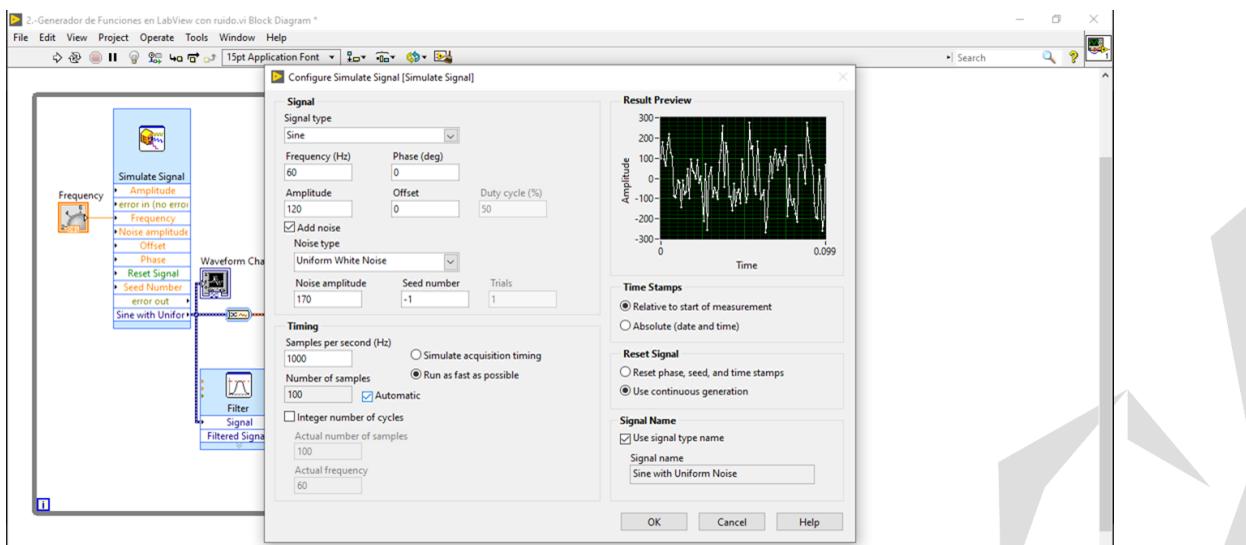
Vamos a volver a cambiar los rangos del filtro pasabandas.

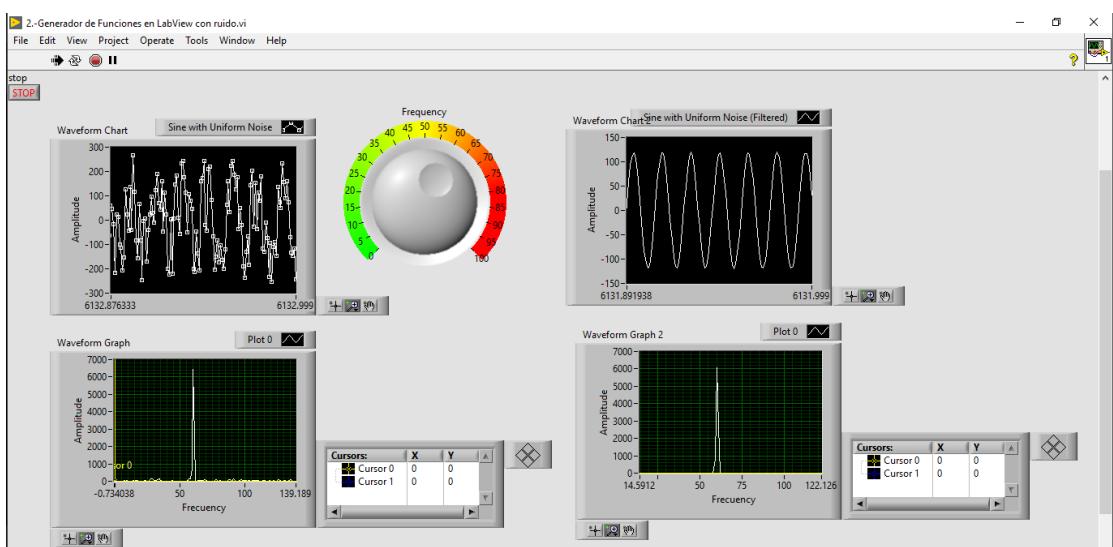
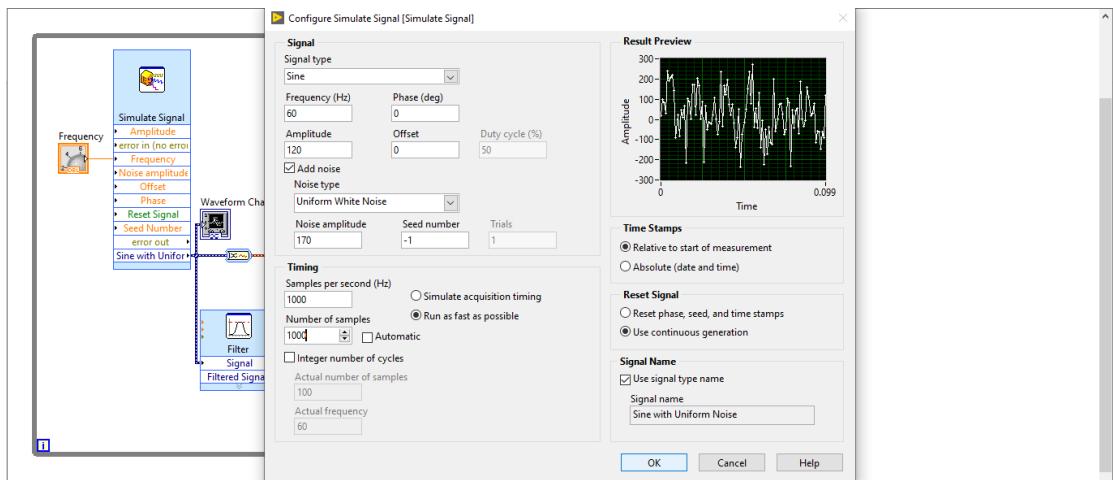




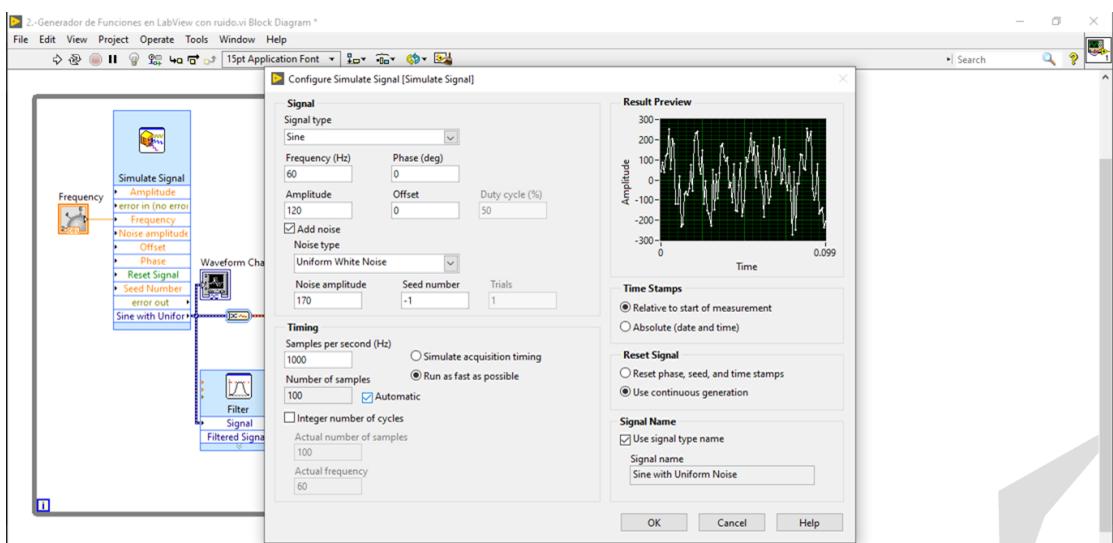
Ejecución del Programa: Gráficas Con y Sin Filtro con Mayor Muestreo

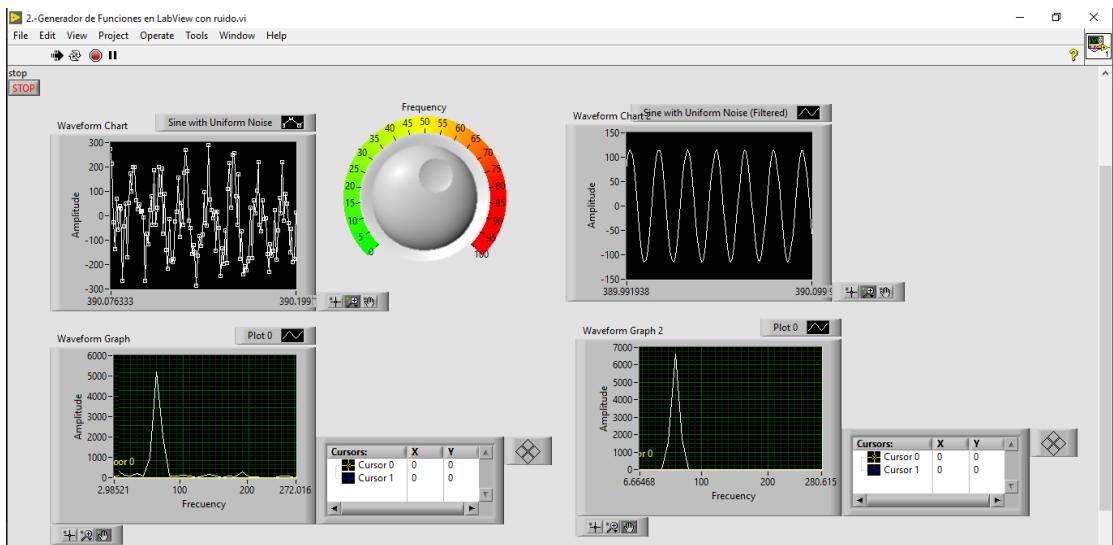
Ahora vamos a subir el número de muestreo, osea a tomar más muestras para la señal original.



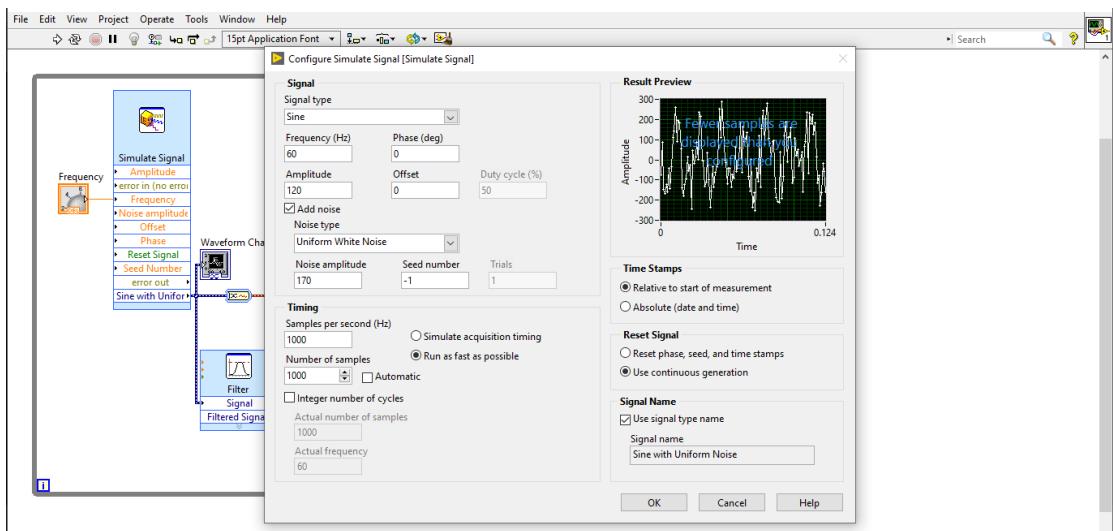


Si vuelvo a cambiar el muestreo a que sea de 100.

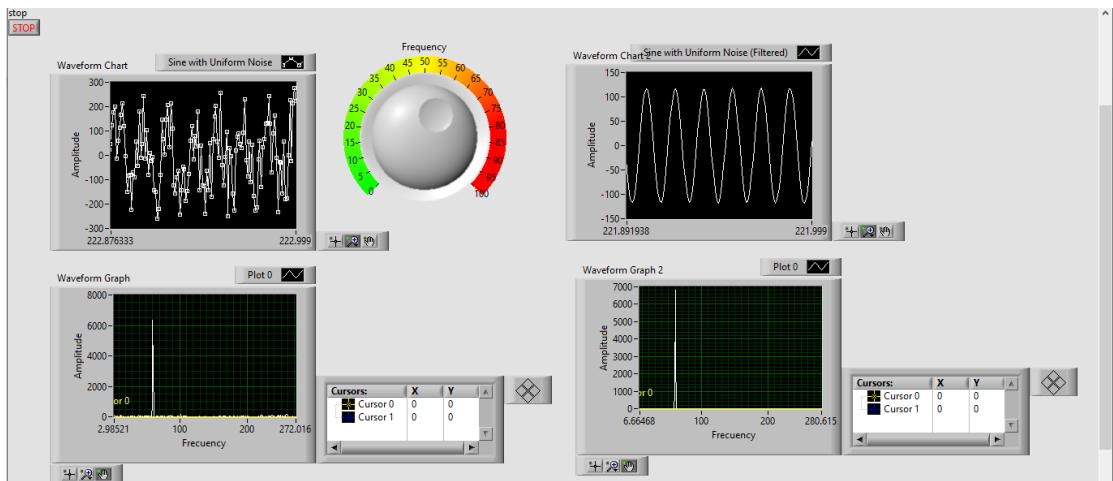


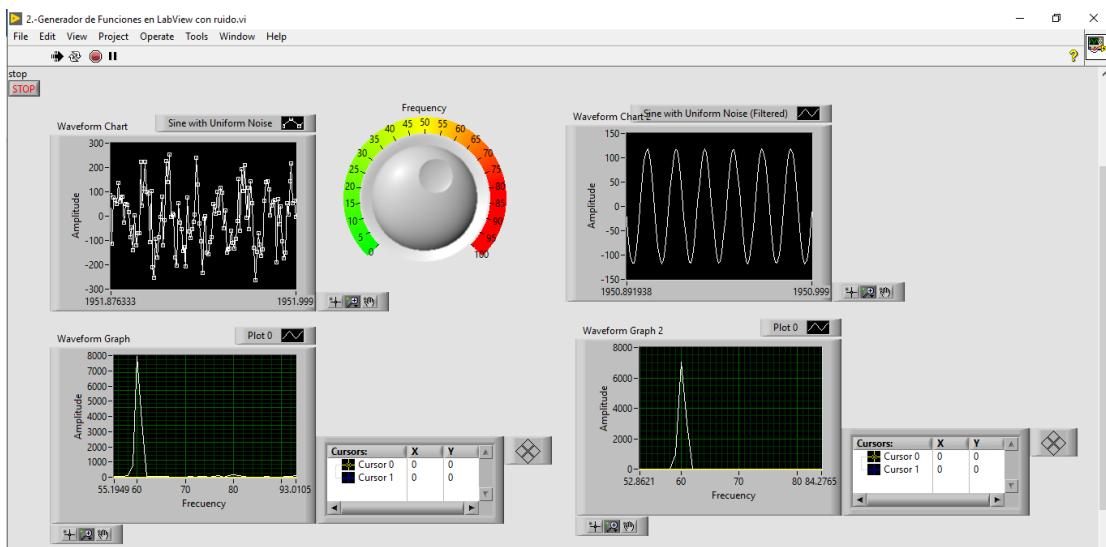


Se ve la diferencia a los 1000 en muestreo.



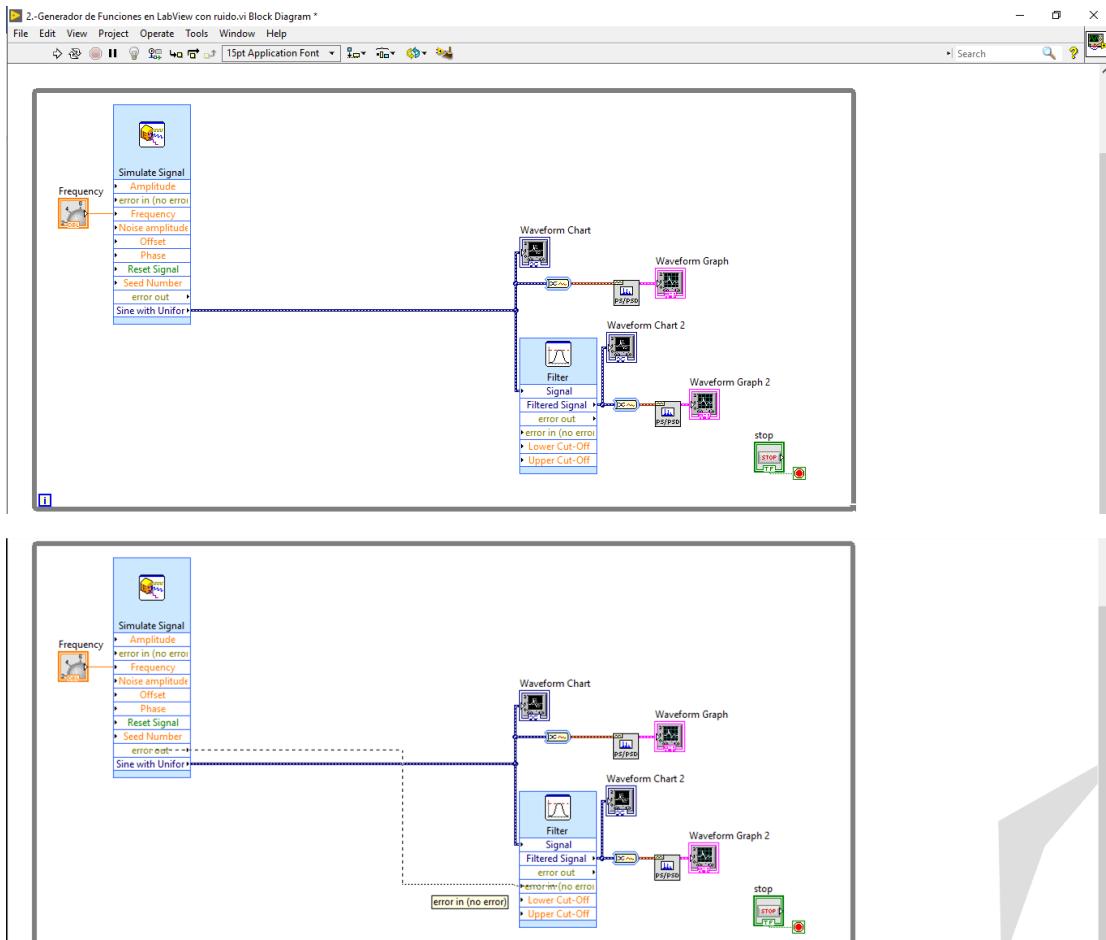
Más que nada esto se puede ver en los Waveform Charts.

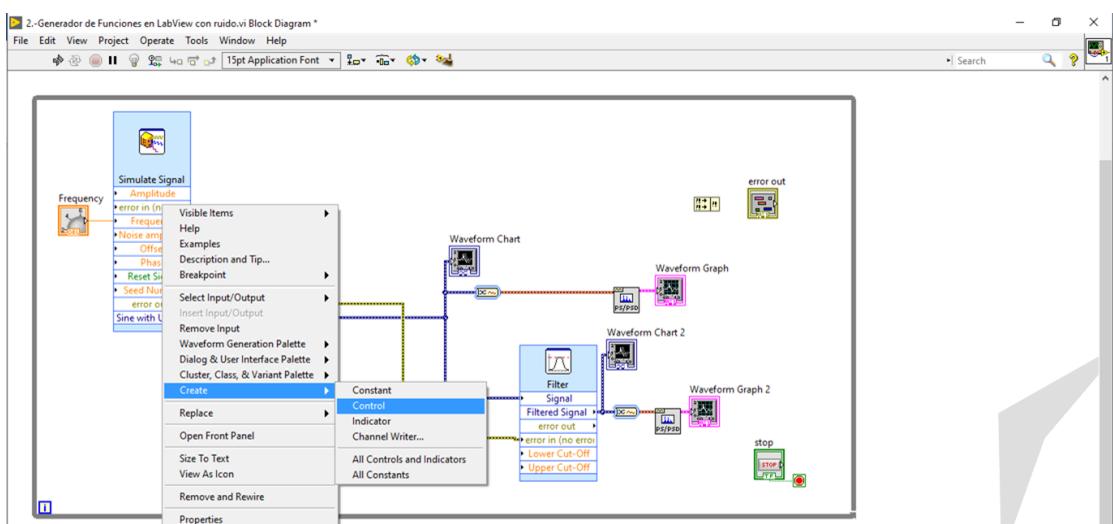
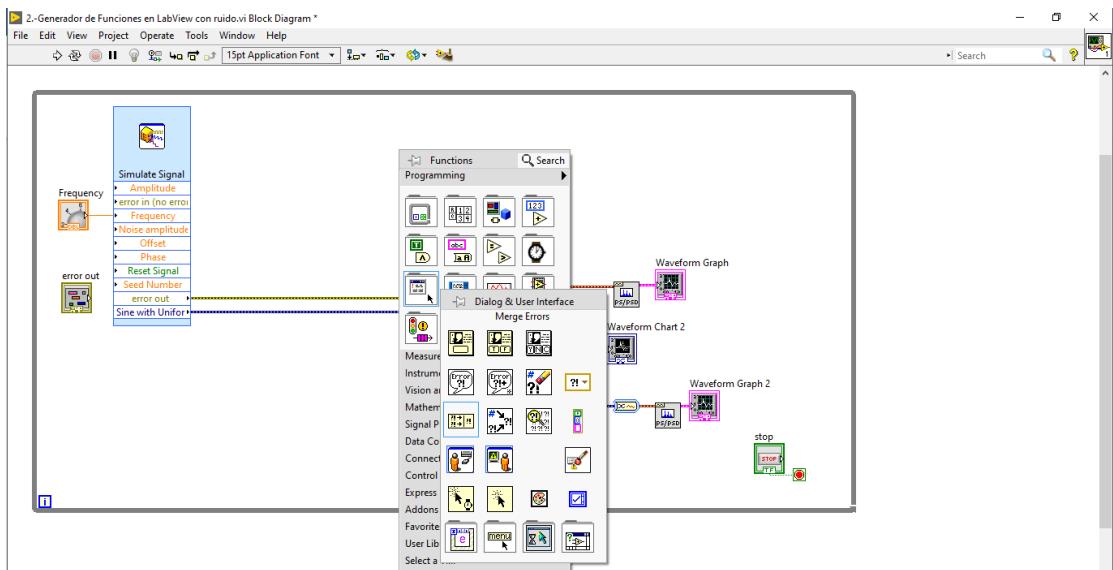
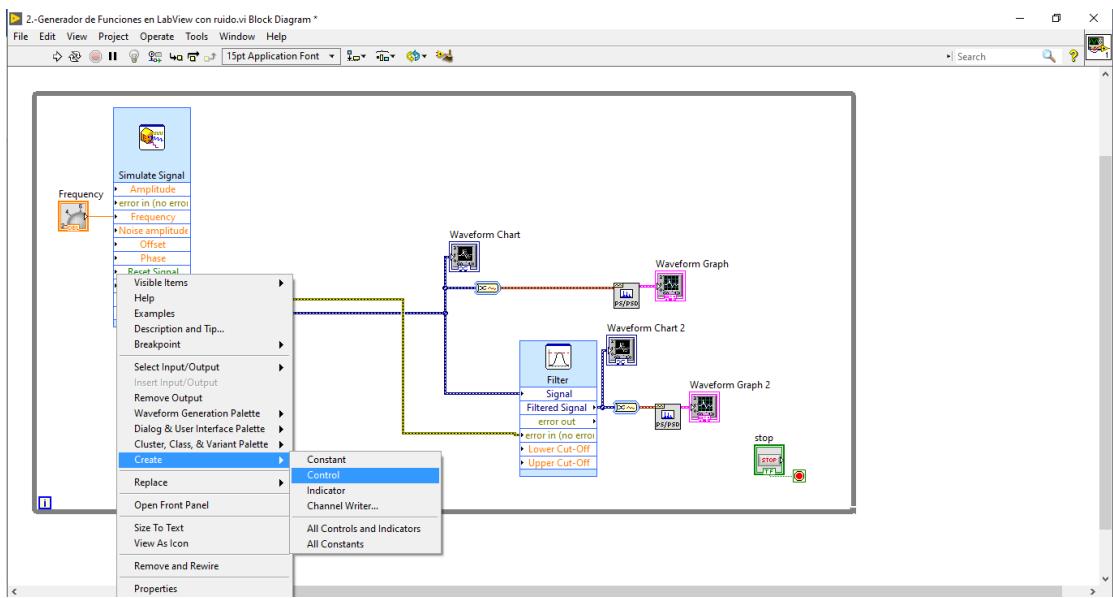


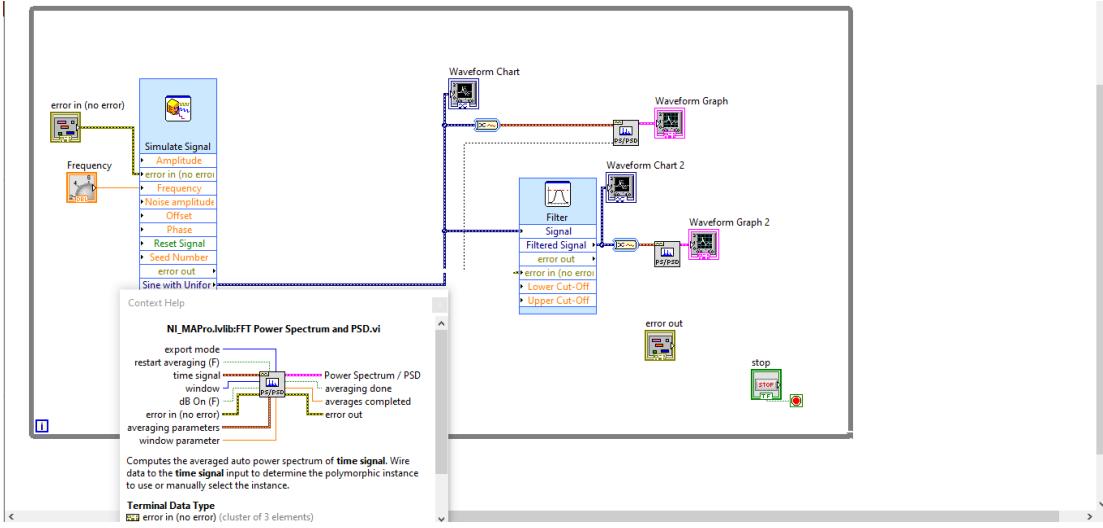


Block Diagram - Simulate Sig: Manejo de Errores Cuando se lee la Señal

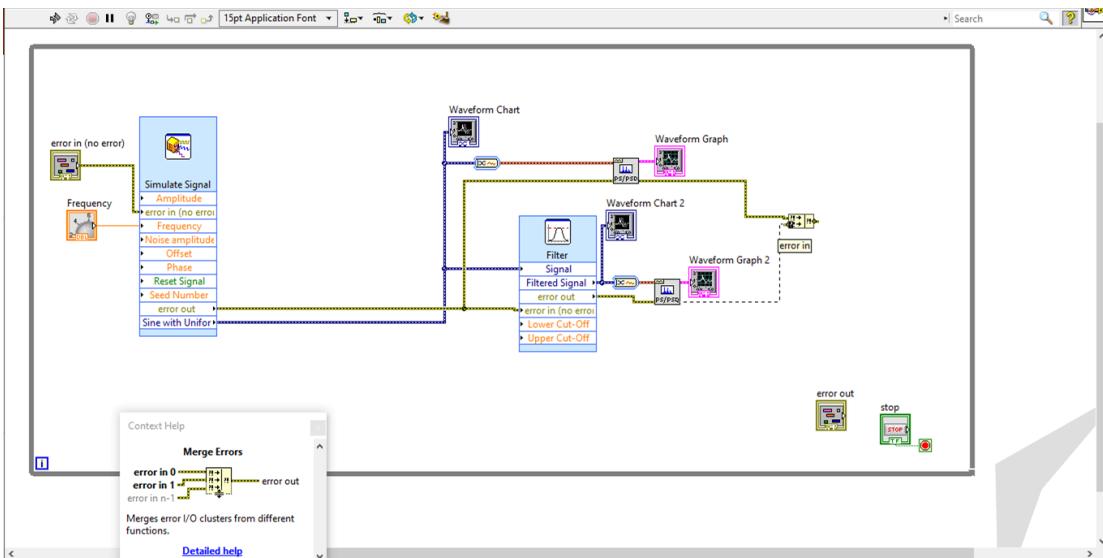
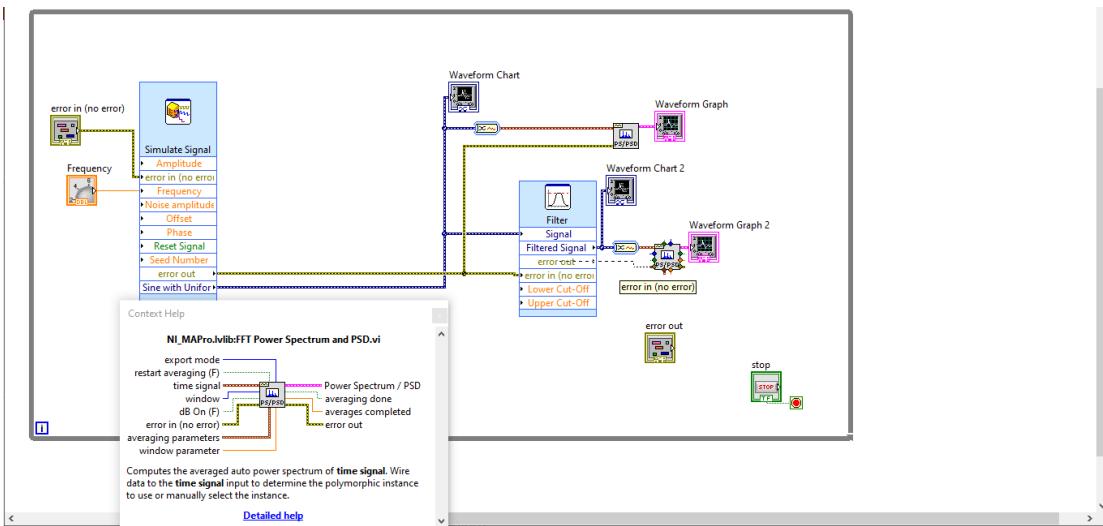
Ahora lo que debemos hacer es arreglar el sistema de paro del while, ya que en este momento la forma en que el programa está hecho es que existe un botón de paro que se ejecutará cuando el usuario se dé cuenta de algún error en el programa, pero esto debe estar programado para que por sí solo se pare el programa cuando detecte si hay algún error.





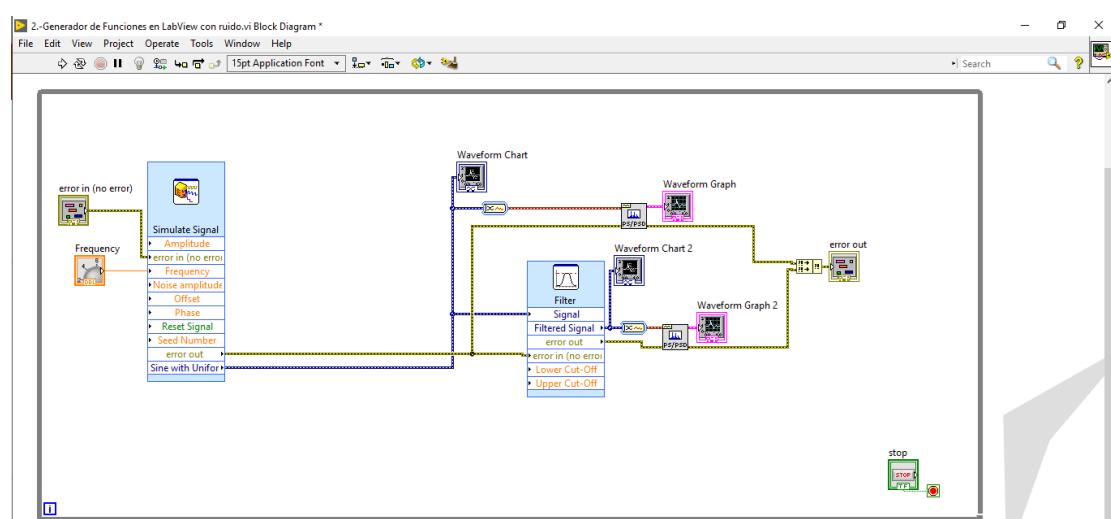
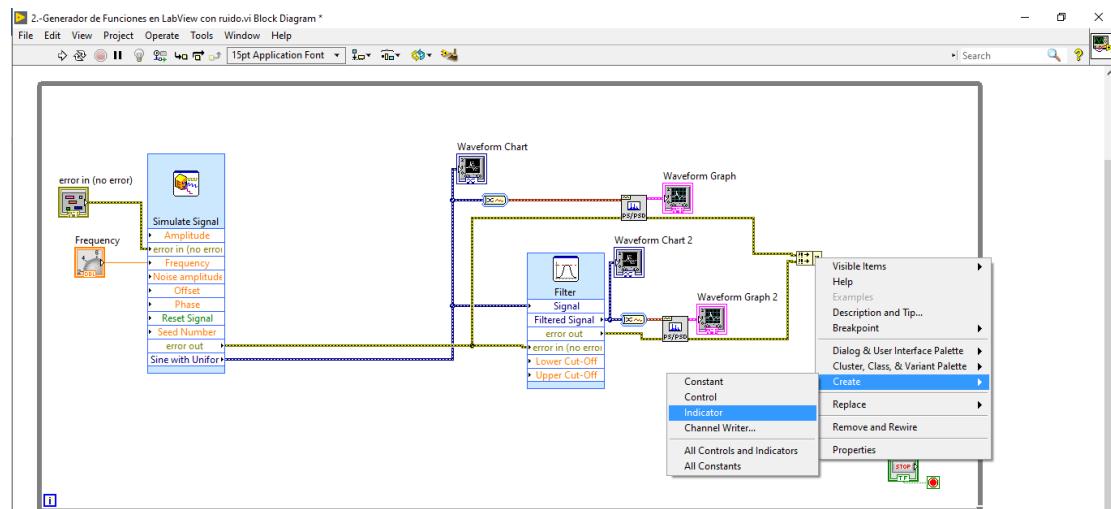
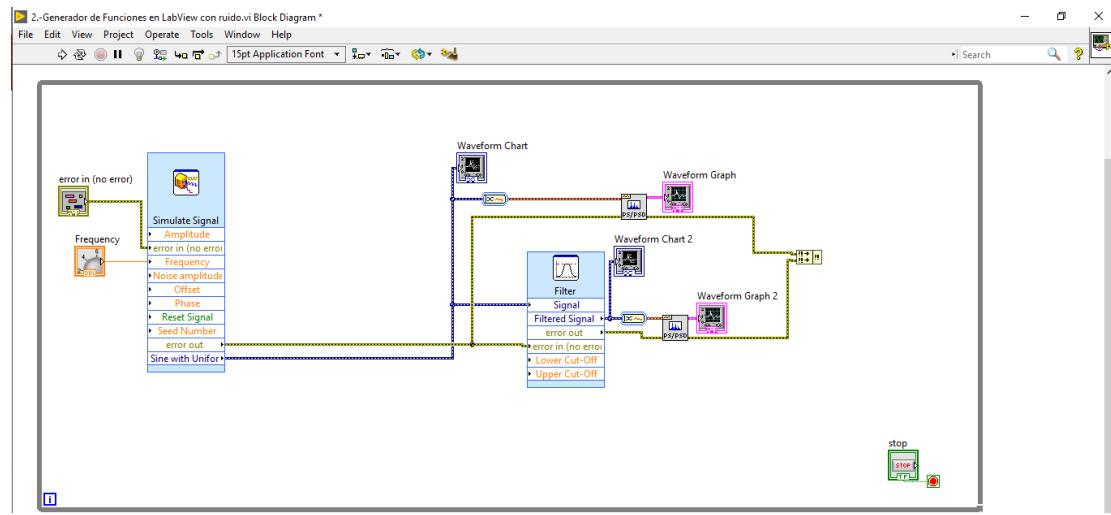


El error de igual manera pasa por el bloque del filtro de la señal para finalmente incluirse en un cluster.

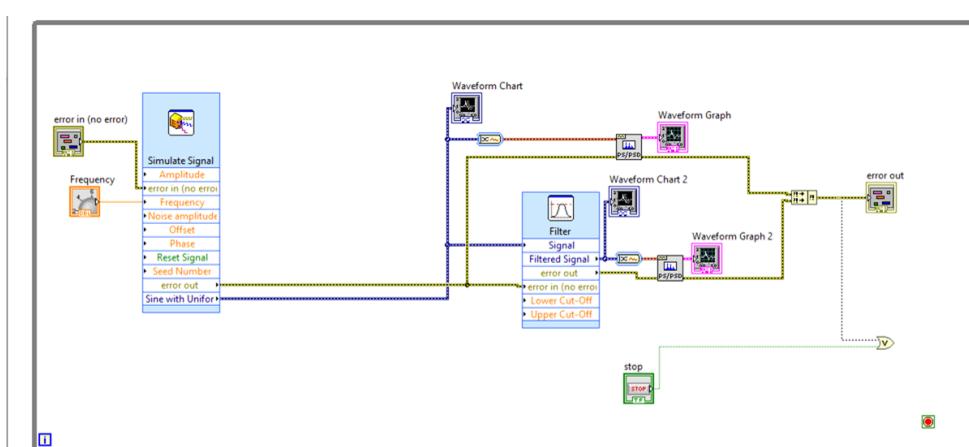
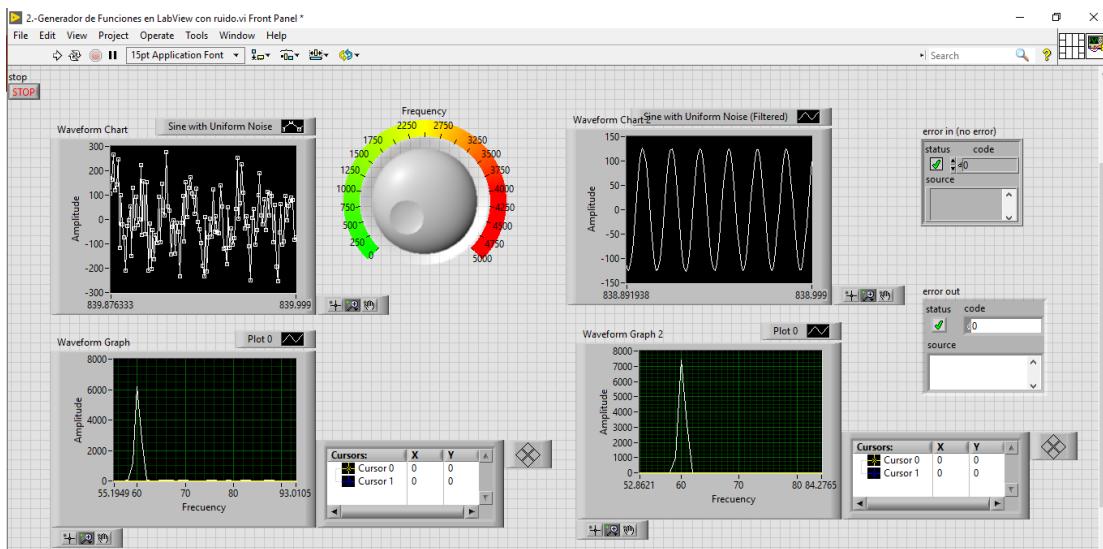
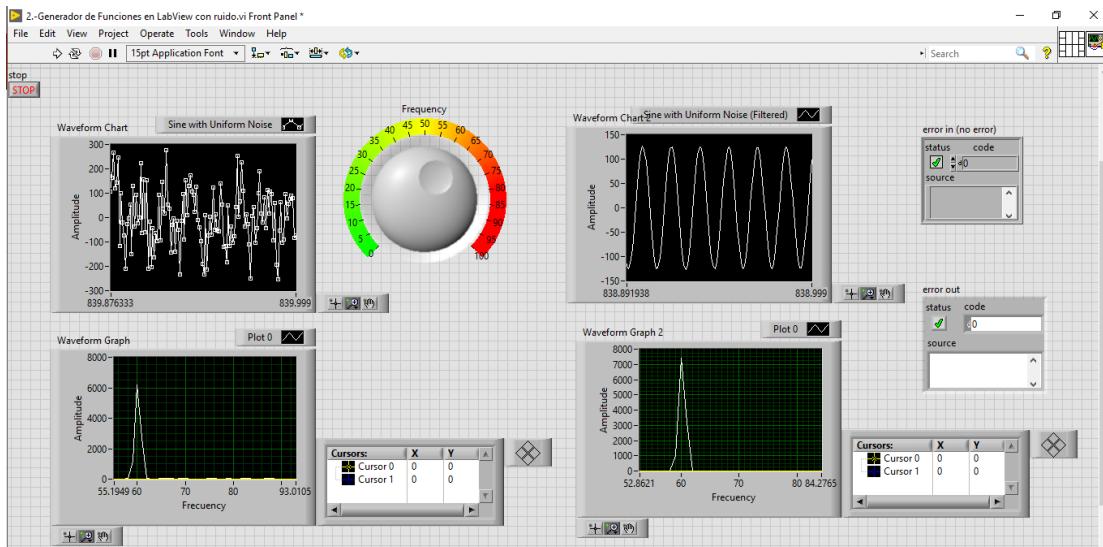


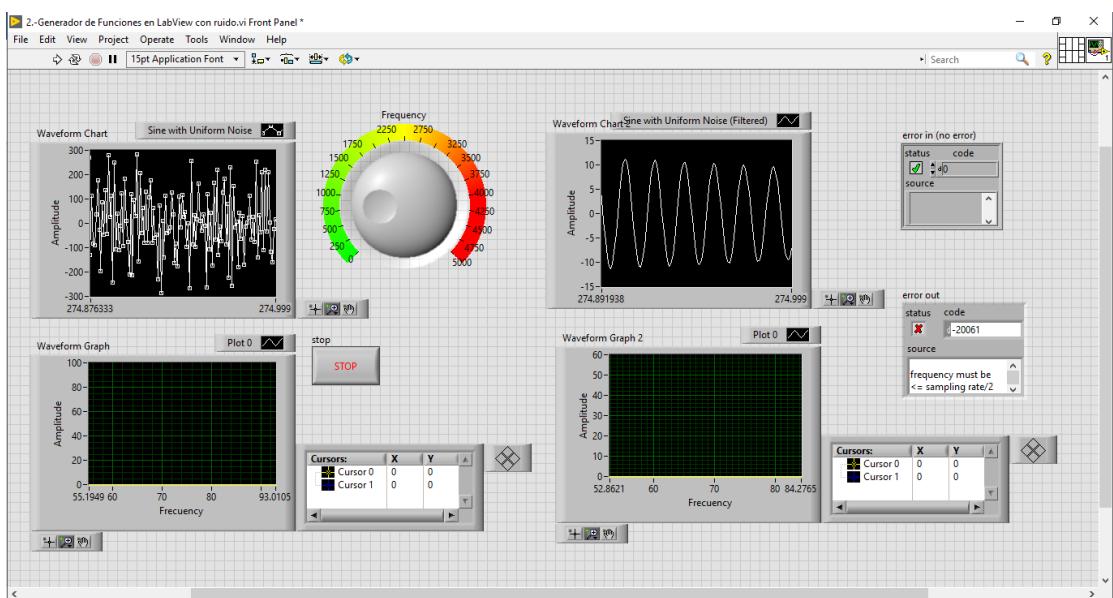
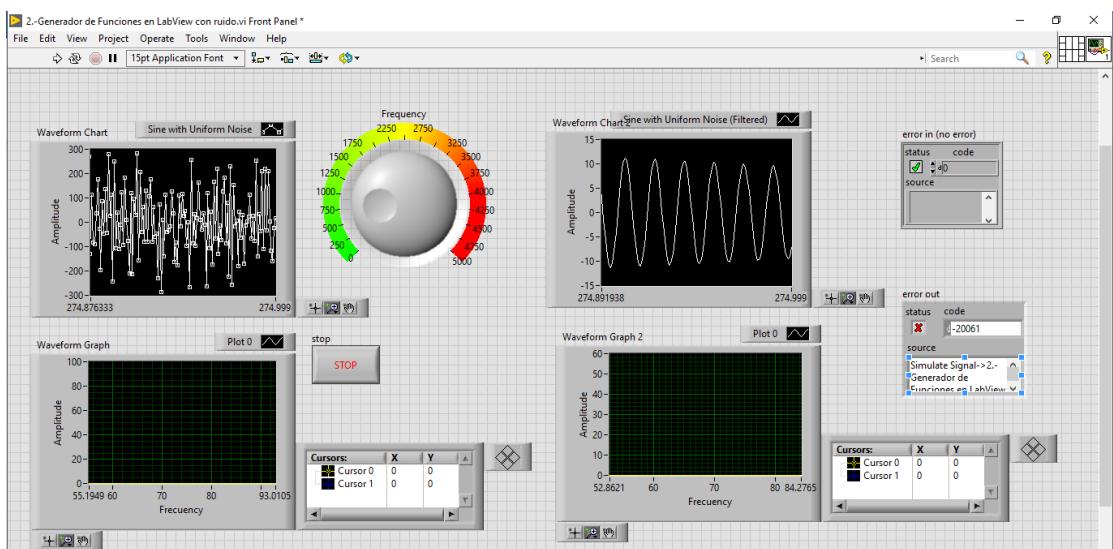
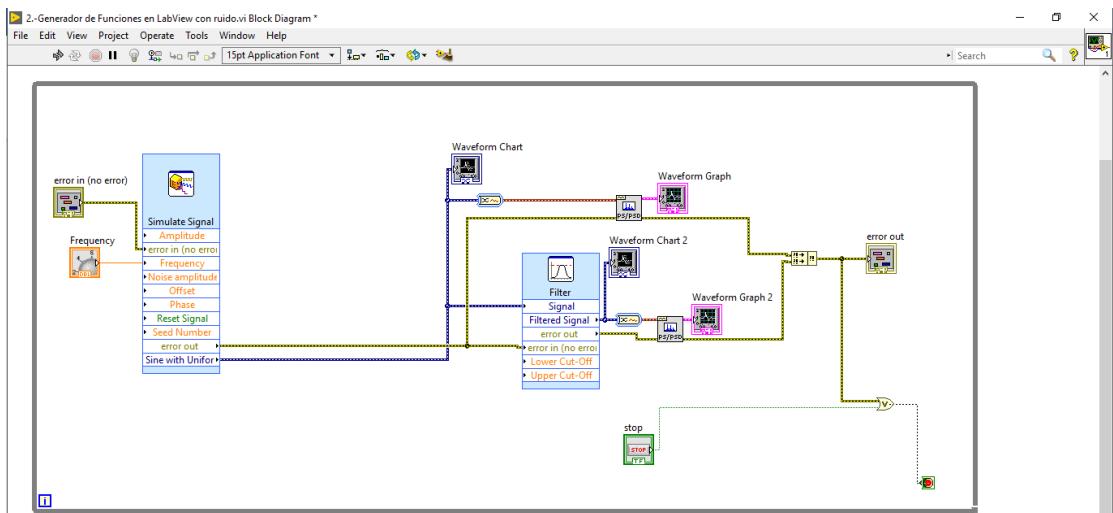
Cluster: Varios Tipos de Datos Agrupados en Uno

Ya que haya llegado al cluster el mensaje de error proveniente del bloque de generación de señal, se debe crear su indicador dando clic derecho sobre él y seleccionando la opción de Create → Indicator.



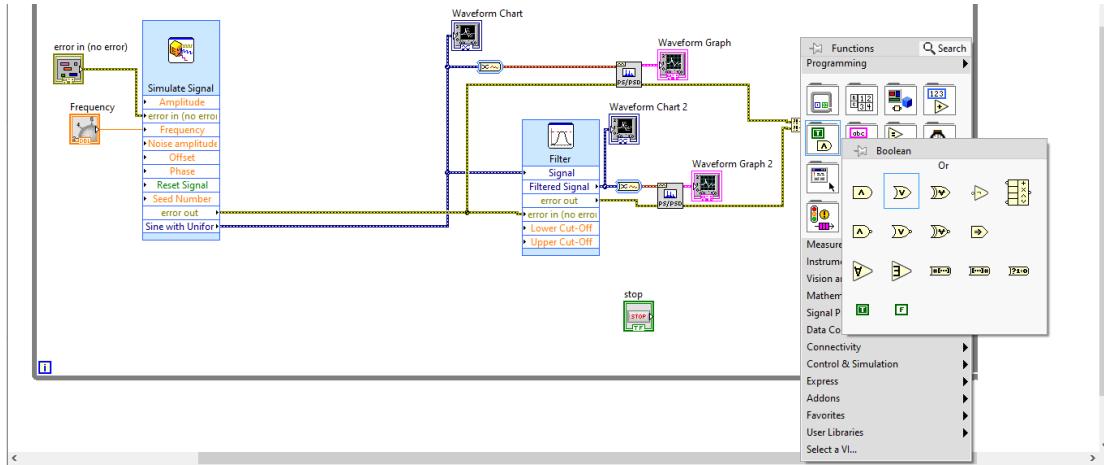
Luego lo que vamos a hacer es forzar un error al aumentar mucho la frecuencia y lo que hará el bloque de error que creamos es solo mandar un mensaje de error en la ventana gris en vez de que el mismo LabVIEW lo mande solo como lo había hecho antes.



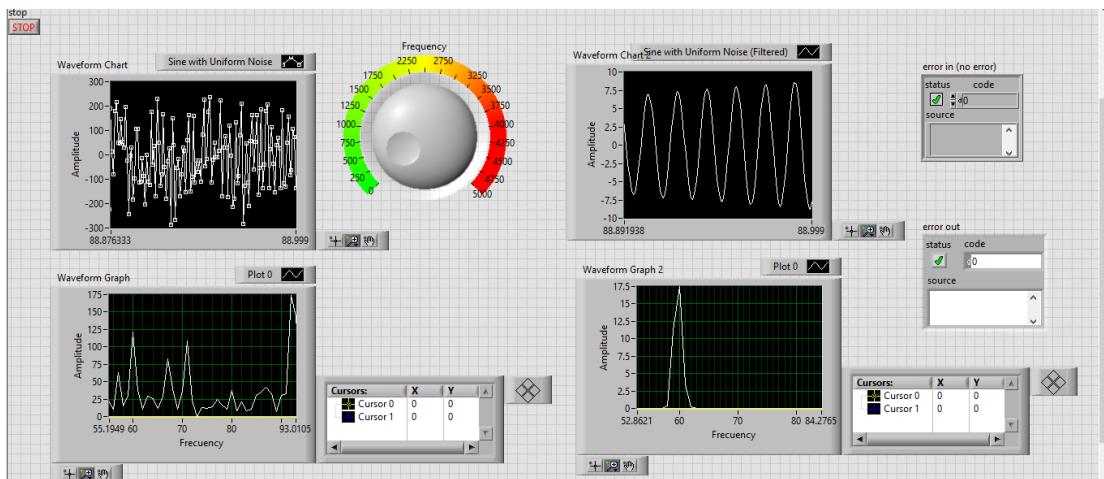
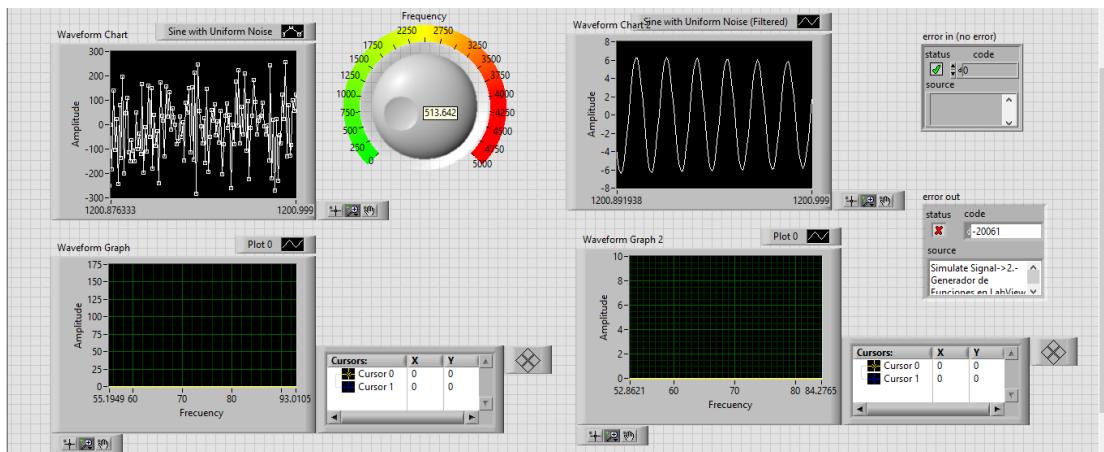


Block Diagram - Compuerta Lógica OR: Detener el Programa por un Error o con Stop

Lo deseado es que cuando aparezca un error el programa se pare o que se pueda hacer de forma manual, para eso vamos a conectar el error conectándolo con el sistema de paro del while y al botón de STOP con una compuerta OR que analice las dos entradas.



El programa entonces se parará cuando la frecuencia sea mayor a 500 Hz o cuando el usuario dé clic en el botón de STOP.



Ejecución del Programa: Gráficas Con Cursores, Con y Sin Filtro y con Manejo de Errores

