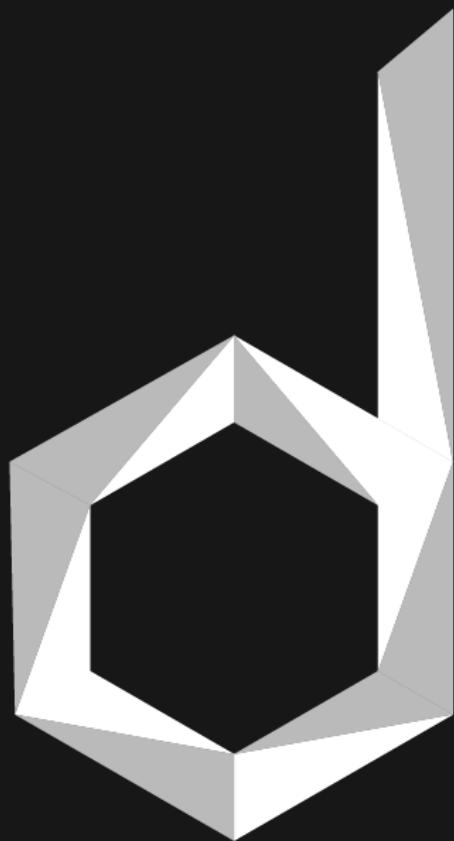


INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

SubVI (Subrutina): Límite Superior,
Led Indicador y Sonido de Gráfica

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: SubVI (Subrutinas)	10
Desarrollo del Programa: SubVI de Señal, Indicador Led y Color de Nivel	10
Block Diagram - Multiply: Multiplicar dos Variables Numéricas Entre Sí.....	10
Block Diagram - Random Number: Creación de Número Aleatorio.....	10
Creación del SubVI: Señal Creada con Números Aleatorios	11
Ingresar a las Ventanas de un SubVI: Doble clic Sobre el Bloque.....	13
Editar Logo de un SubVI: Ícono Superior Derecho	13
Editar la Descripción de un SubVI: Descripción que Aparece con Show Context Help	17
Guardar un SubVI: Como un VI Cualquiera o Para que se Pueda Acceder como Bloque	18
Utilización del SubVI en un VI Nuevo: Medidor de Nivel en Señal	20
Block Diagram - Bucle While: Ejecución Continua del Programa.....	21
Block Diagram - Greater?: Operación Lógica Mayor Que (>)	22
Block Diagram - Bundle: Juntar Varios Tipos de Datos Para Mandarlos a un Cluster.....	23
Front Panel - Waveform Chart: Ventana que Muestra una Señal (Dynamic Data)	24
Ejecución del Programa: Gráfica de Señal de SubVI con Límite Superior.....	26
Block Diagram - Increment: Sumar +1 en cada Ejecución del Bucle	27
Block Diagram - Greater Or Equal?: Operación Lógica Mayor o Igual Que (≥).....	28
Block Diagram - Compuerta Lógica OR: Detener el Programa por un Error o con Stop	30
Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque.....	31
Block Diagram - Wait Until Next ms Multiple: Temporizador en milisegundos.....	32
Ejecución del Programa: Señal con Límite Superior y Led que Parpadea.....	33

Block Diagram - Case Structure: Condicional Switch Case	34
Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque.....	36
Block Diagram - Color Box Constant: Constante de Color.....	39
Ejecución del Programa: Señal con Led y Cambio de Color al Pasar el Límite Superior	40
Block Diagram - Beep: Hacer Sonar un Sonido Durante Cierto Tiempo en Milisegundos	40
Ejecución del Programa: Señal con Led, Cambio de Color al Pasar el Límite Superior y Sonido	41
Block Diagram - Case Structure: Condicional Switch Case	43
Block Diagram - Color Box Constant: Constante de Color.....	45
Ejecución del Programa: Señal con Led, Cambio de Color al Pasar el Límite Superior y Sonido	46
Ejecución del Programa: Señal con Led, Cambio de Color en el Límite Superior y Sonido	47
Block Diagram - Flat Secuence Structure: Ejecución Ordenada Secuencial de Funciones.....	48
Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque.....	49
Block Diagram - Divide: Dividir dos Variables Numéricas Entre Sí	52
Block Diagram - To Unsigned Long Integer: Conversión Número Decimal a Entero	55
Ejecución del Programa: Señal con Led, Color en Límite Superior, Sonido y Control de dx y Puntos....	56
Block Diagram - Get Date/Time String: Obtención de un String con la Fecha y Hora.....	57
Block Diagram - Format Into String: Conversión de Cualquier Tipo de Dato a un String.....	59
Ejecución del Programa: Señal con Led, Color en Límite Superior, Sonido, Controles y Fecha/Hora	62



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.

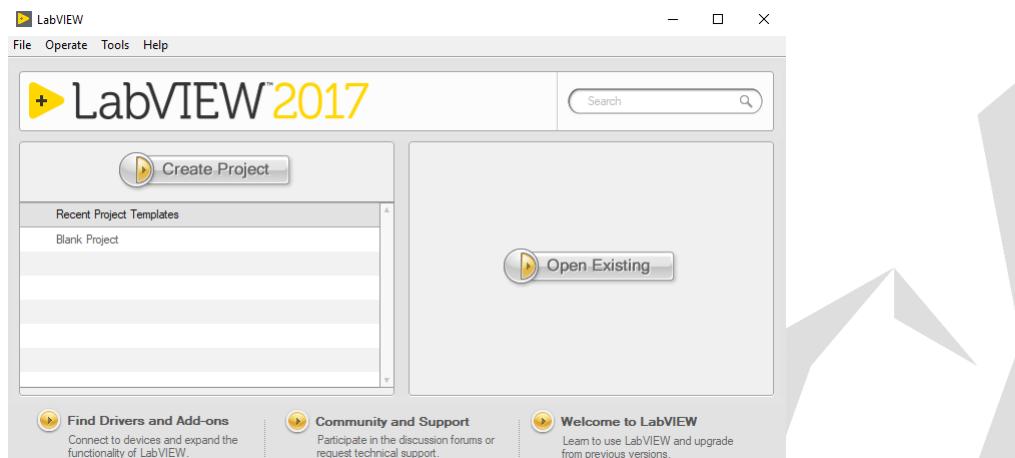
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

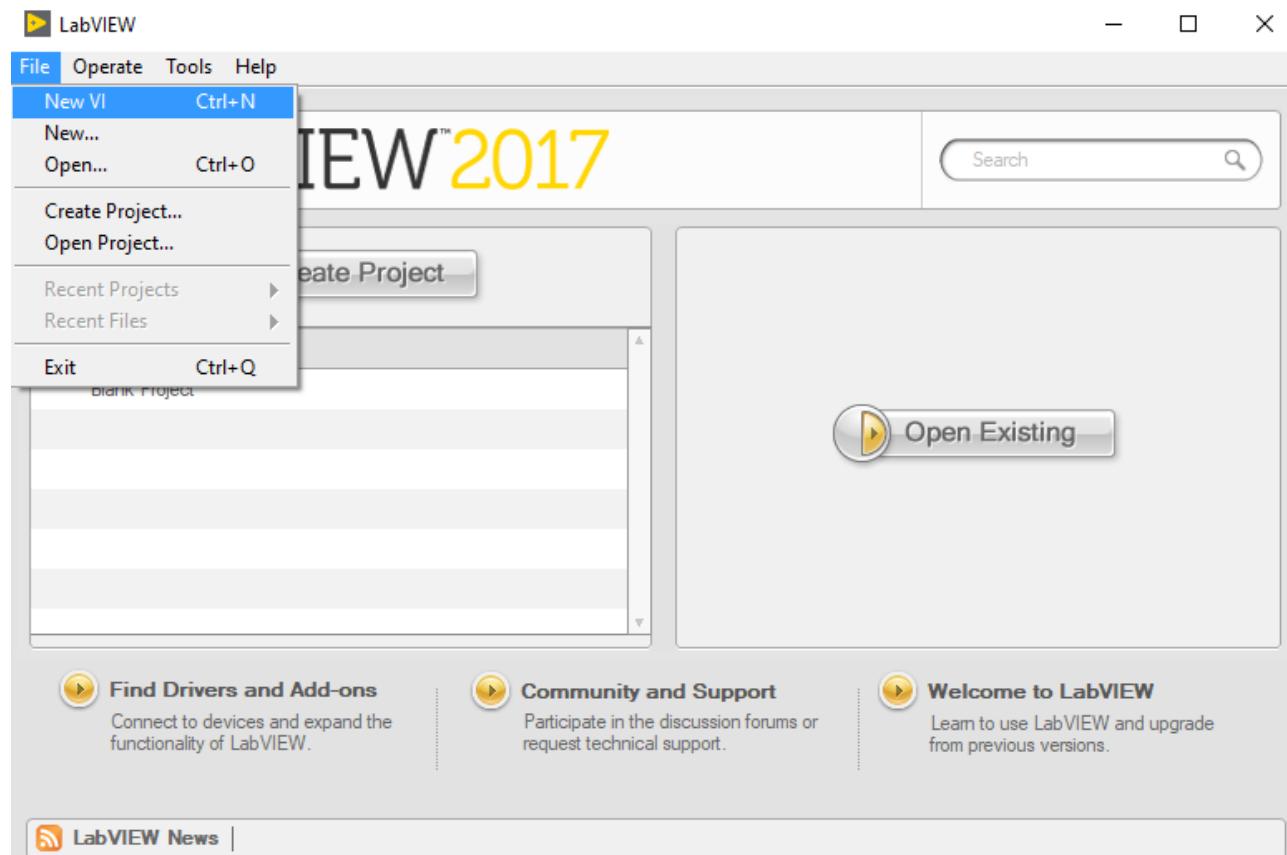


Introducción al Entorno de LabVIEW:

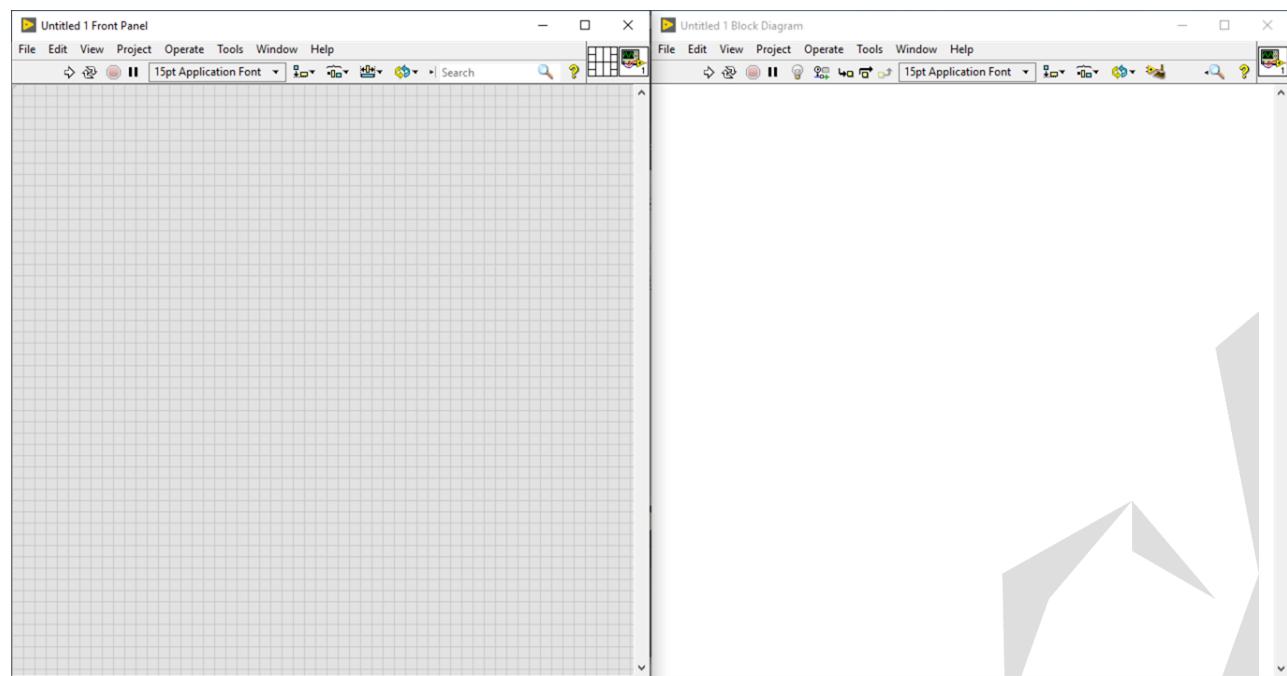
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

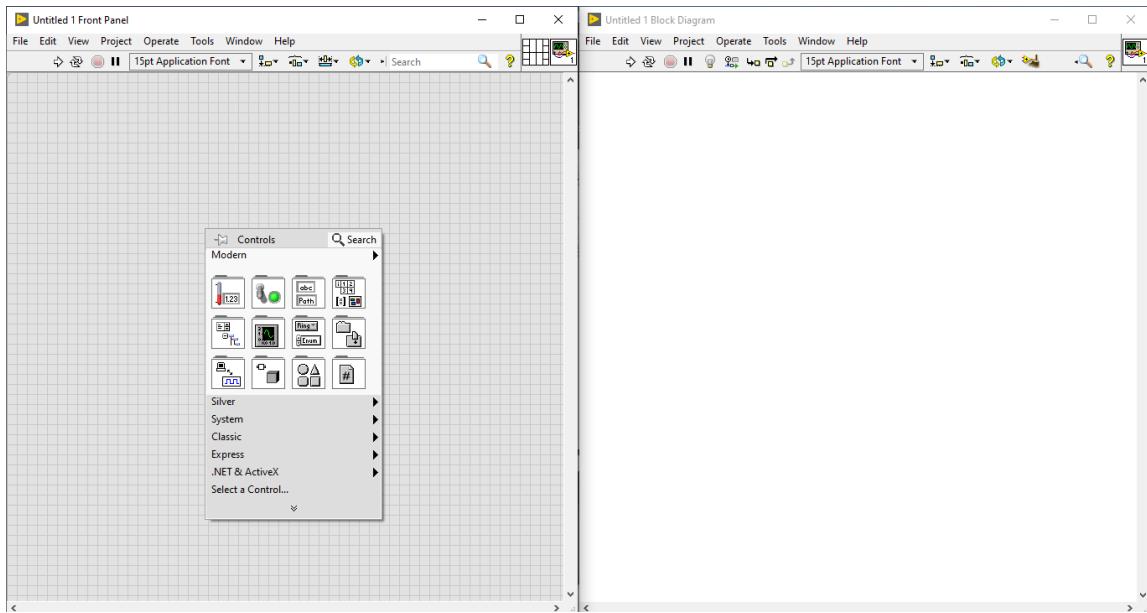


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



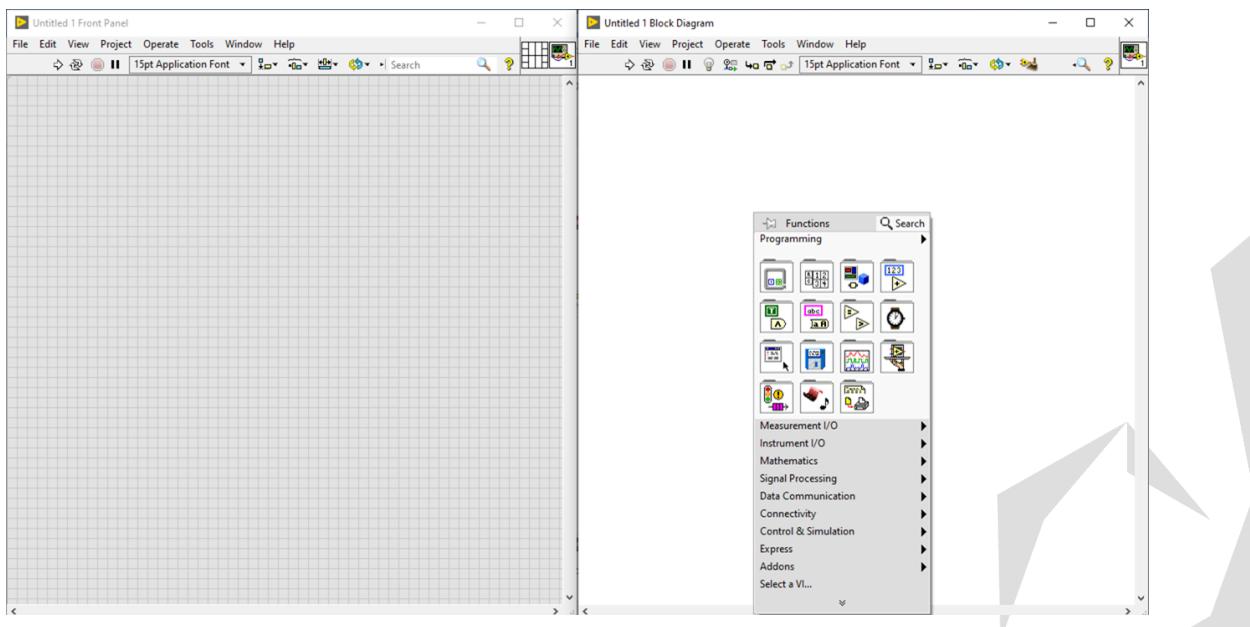
Front Panel: Ventana Gris con la Interfaz del Programa

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.



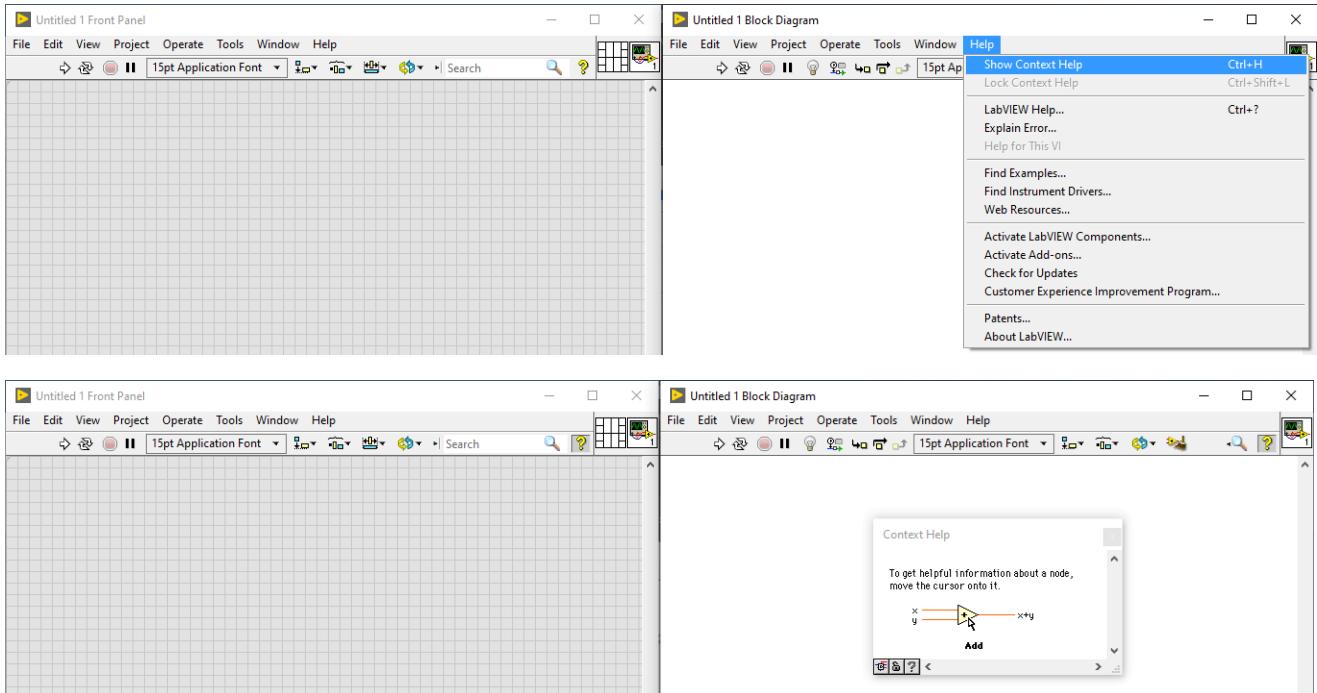
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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

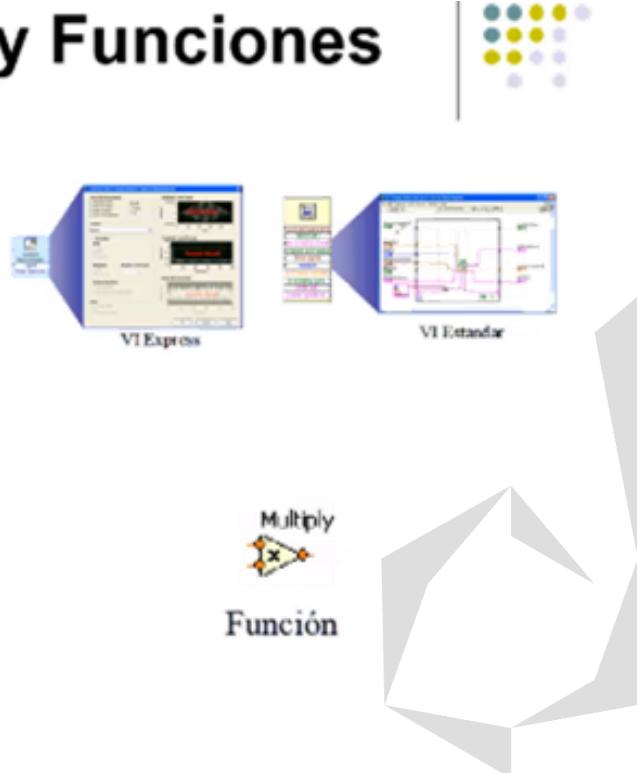
Seleccionando la opción de Help → Show Context Help, aparecerá una ventana emergente que explicará las propiedades de los bloques que se puede seleccionar, mostrando una descripción de su función, imágenes explicativas y significado de sus pines de entrada y salida.



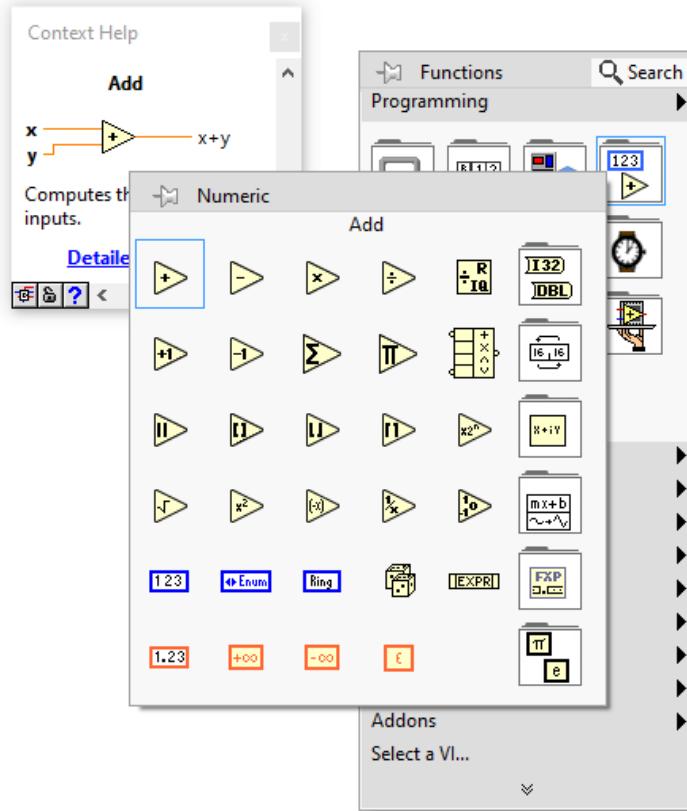
Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:

VIs Express, VIs y Funciones

- **VIs Expreso:** VIs interactivos con pagina de dialogo configurable
- **VIs estándezar:** 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.

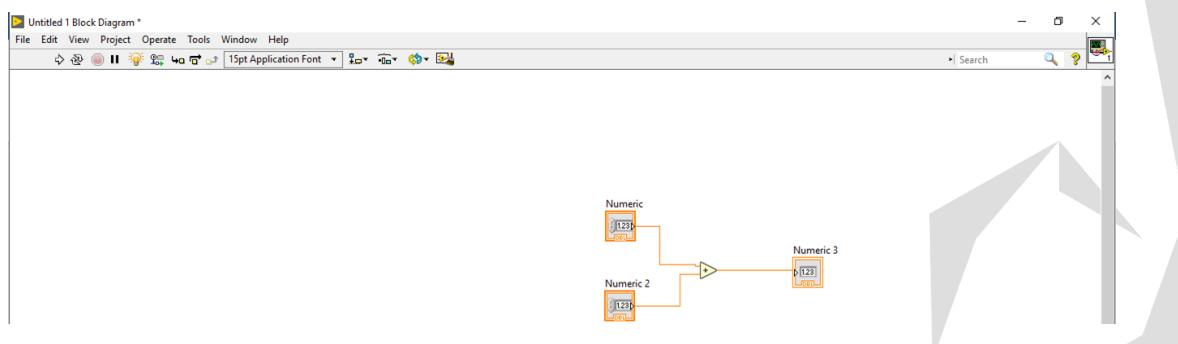


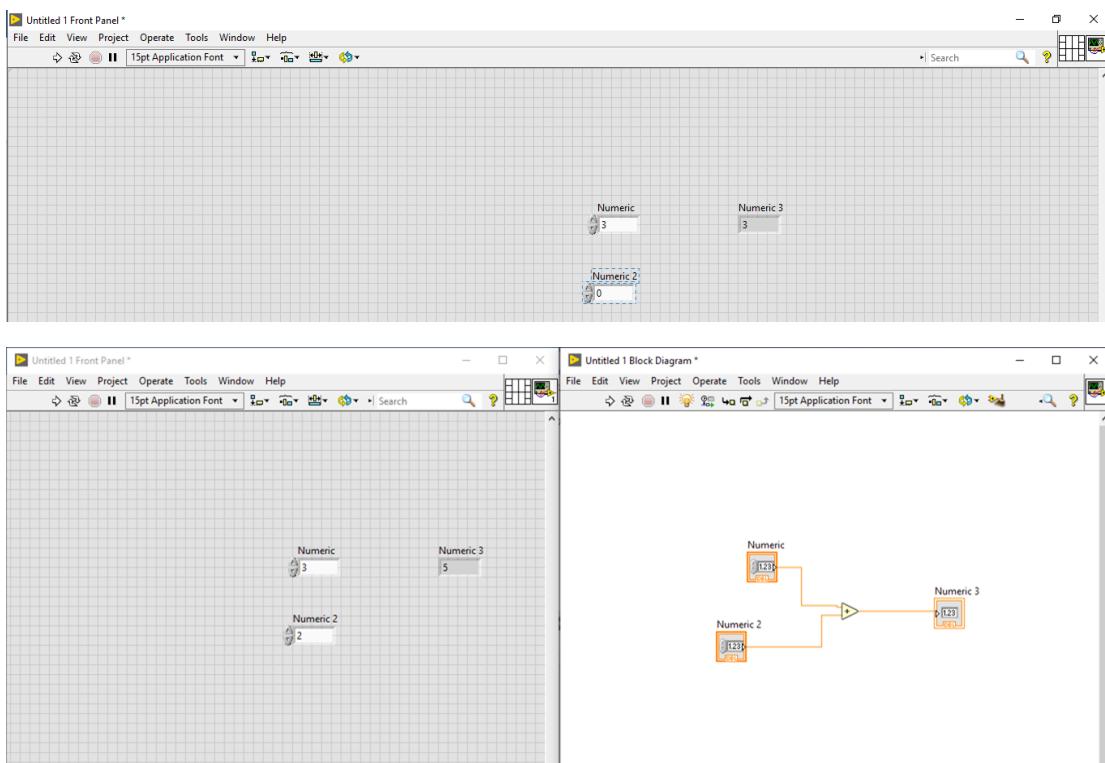
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

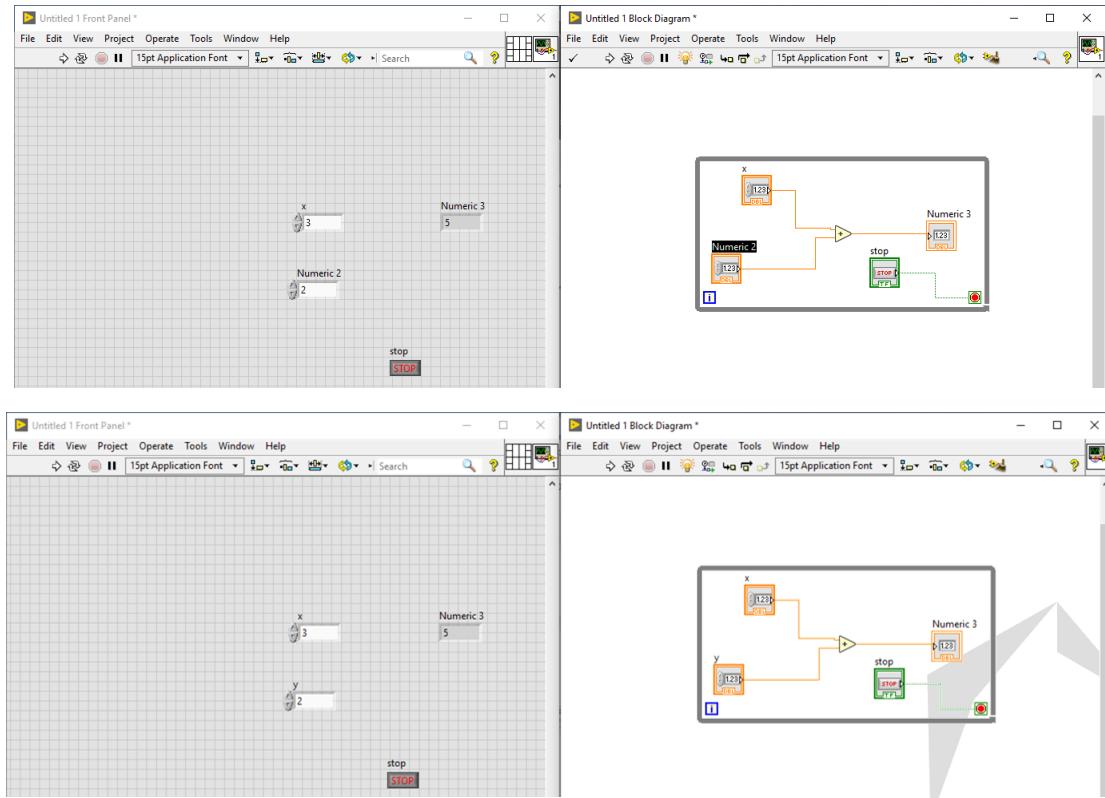
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

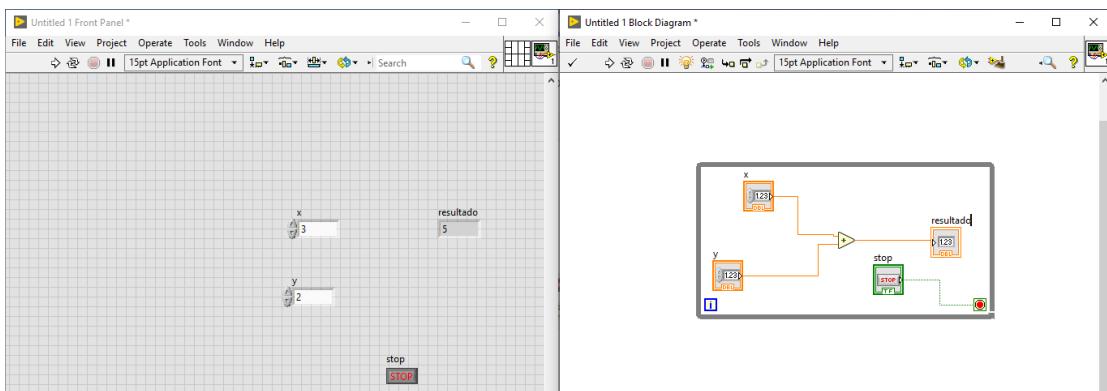




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

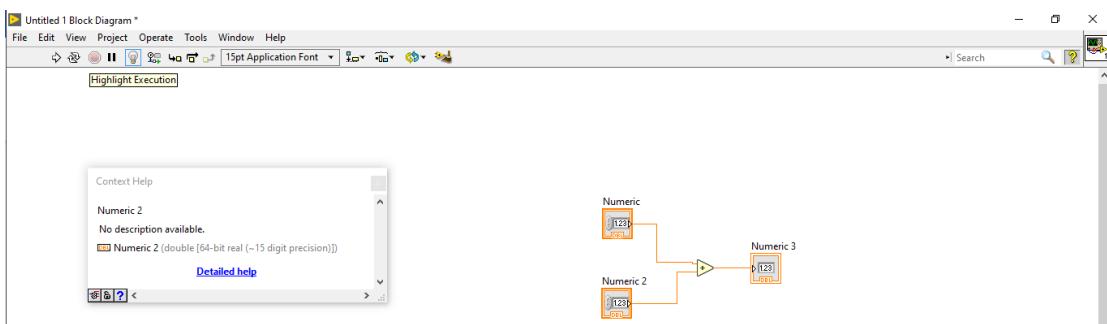
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





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

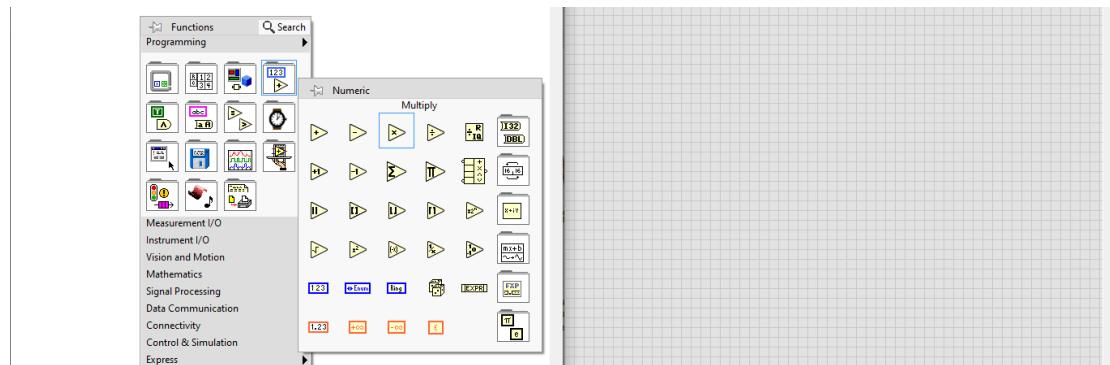
Con el botón de Clean Up Diagram que se encuentra en la parte superior derecha del Block Diagram se organizan mejor y de forma automática mis elementos.

Programa: SubVI (Subrutinas)

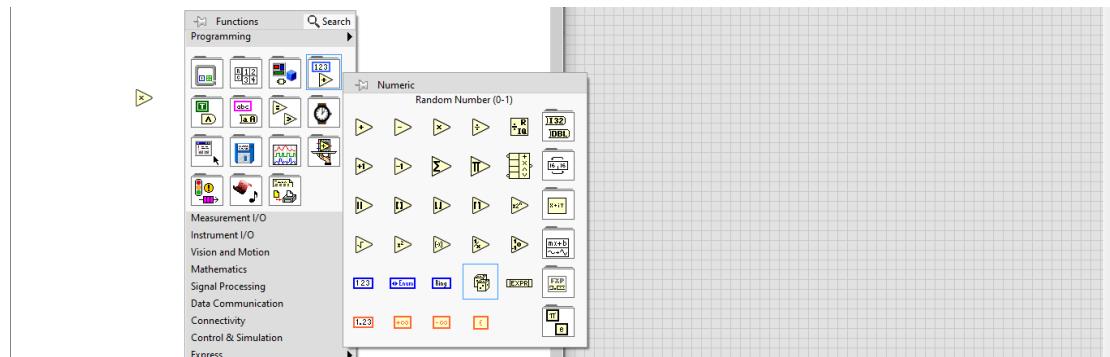
Una subrutina o SubVI (donde VI es un programa de LabVIEW) es un programa de bloques hecho en LabVIEW que se puede guardar para usarse como un bloque dentro de otros programas. SubVI es lo mismo a decir una función o un script en otros lenguajes de programación.

Desarrollo del Programa: SubVI de Señal, Indicador Led y Color de Nivel

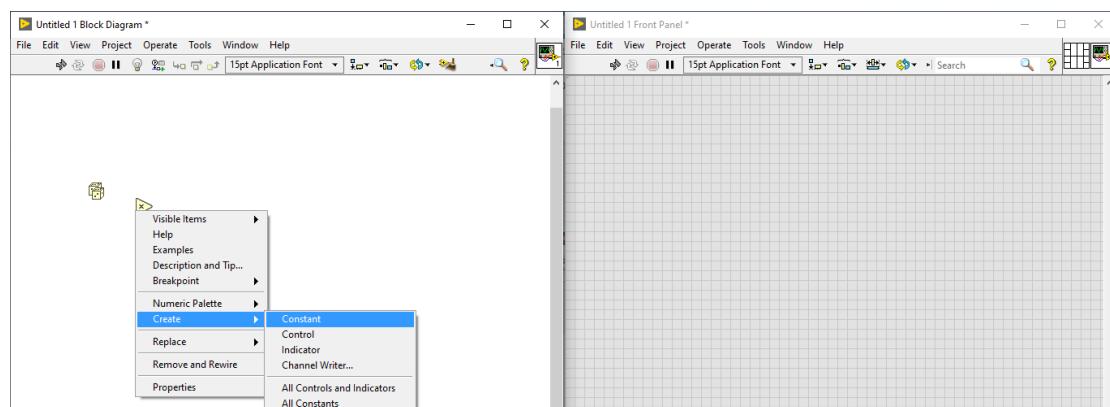
Block Diagram - Multiply: Multiplicar dos Variables Numéricas Entre Sí



Block Diagram - Random Number: Creación de Número Aleatorio

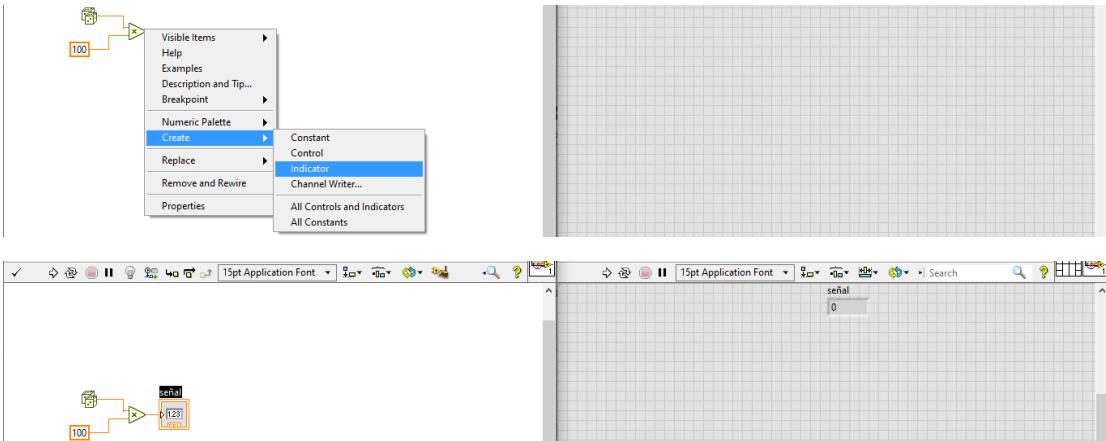


Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant.

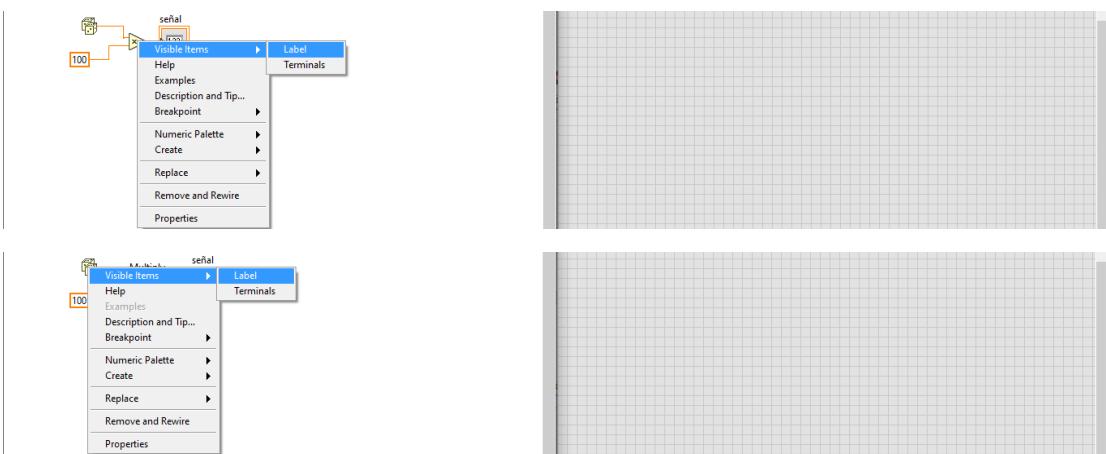




Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator. Con esto se crea un indicador numérico que diga el resultado de la multiplicación en la interfaz, el cual irá mostrando números aleatorios entre 1 y 100.

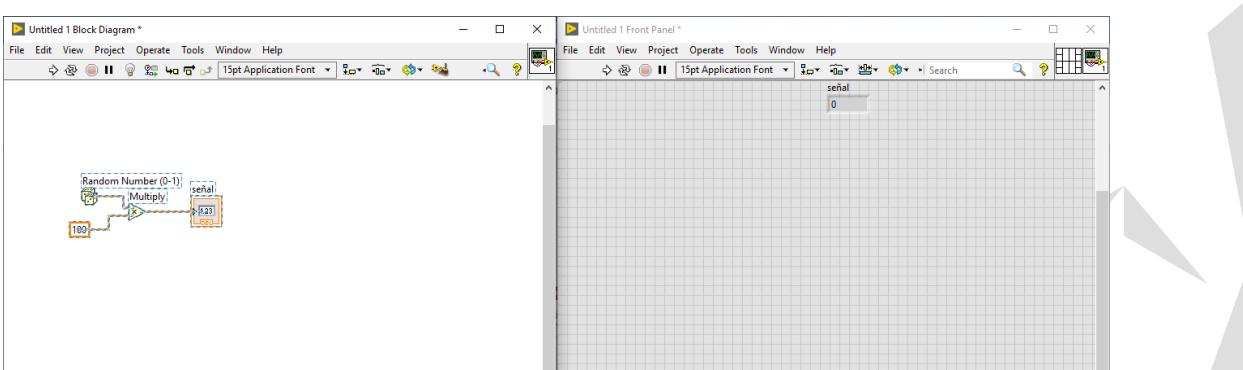


Mostrar nombre del bloque: Clic derecho → Visible Items → Label.

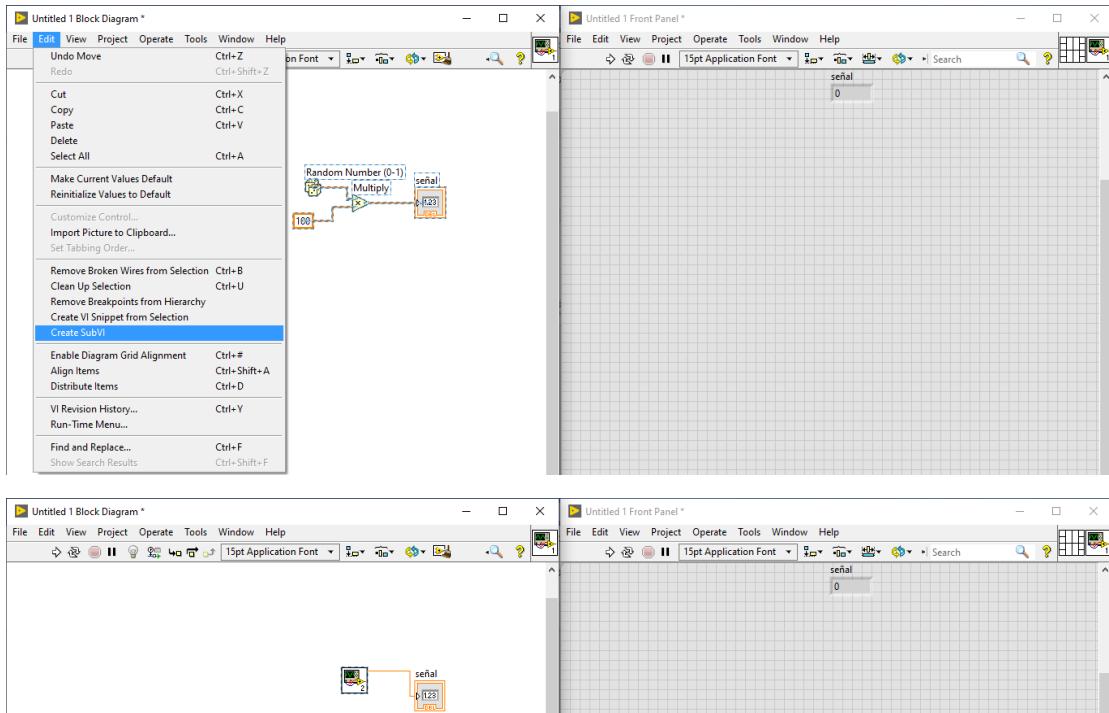


Creación del SubVI: Señal Creada con Números Aleatorios

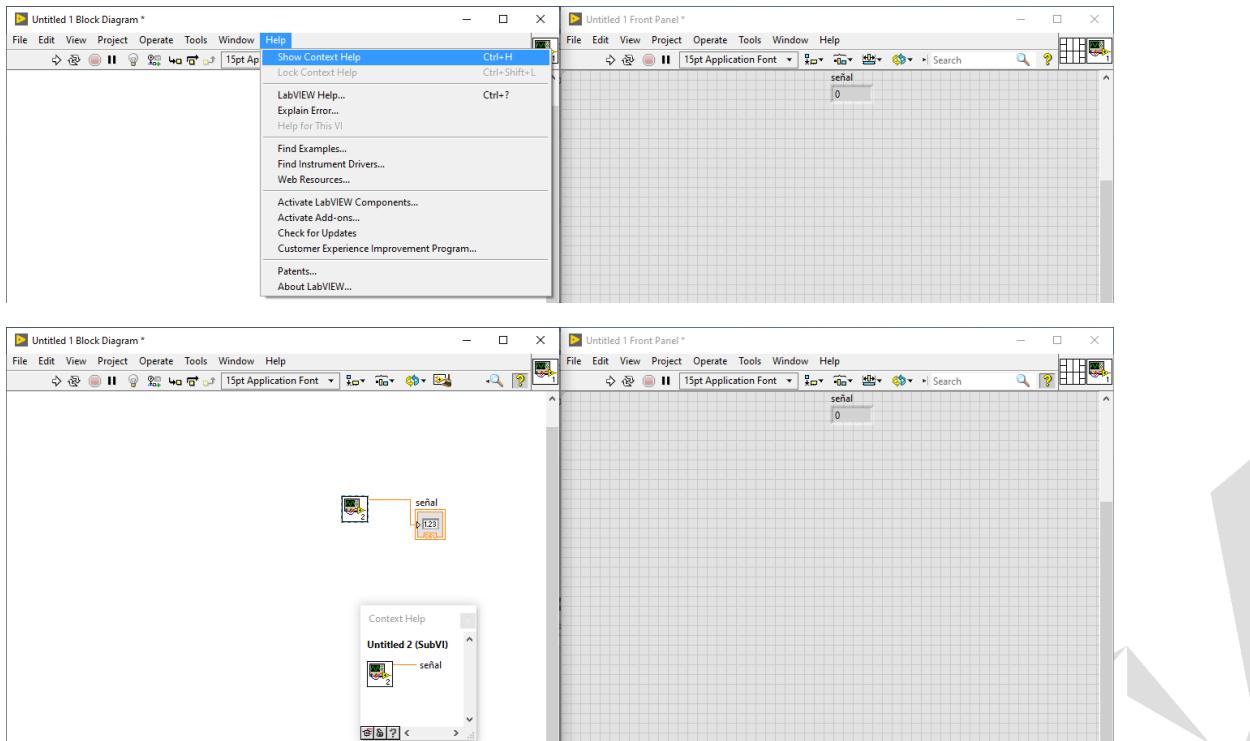
Para crear un SubVI primero se debe seleccionar con el mouse todos los bloques que se quiere incluir.



Después seleccionaremos la opción de Edit → Create SubVI, al hacerlo LabVIEW me va a encapsular todos los bloques de código en un bloque nuevo, el cual tendrá su propia descripción, símbolo, etc.

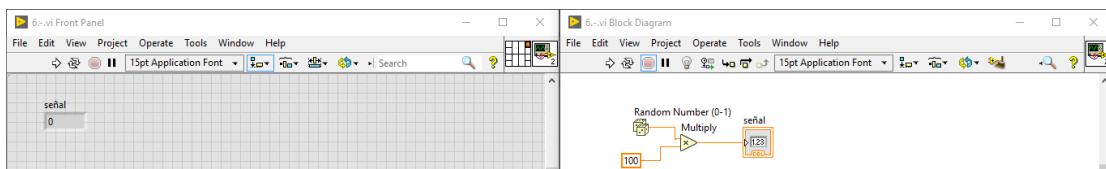
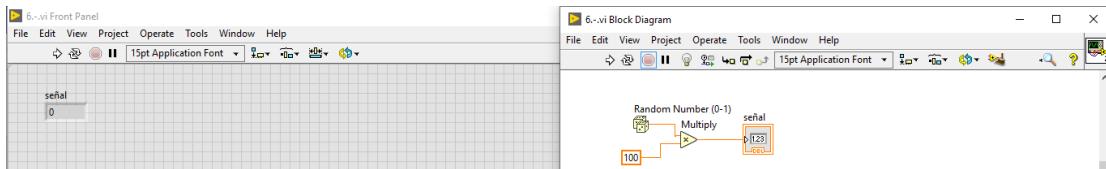
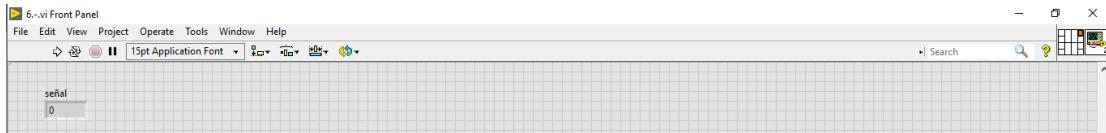


La descripción del SubVI puede ser vista cuando utilice la herramienta de Show Context Help sobre él.



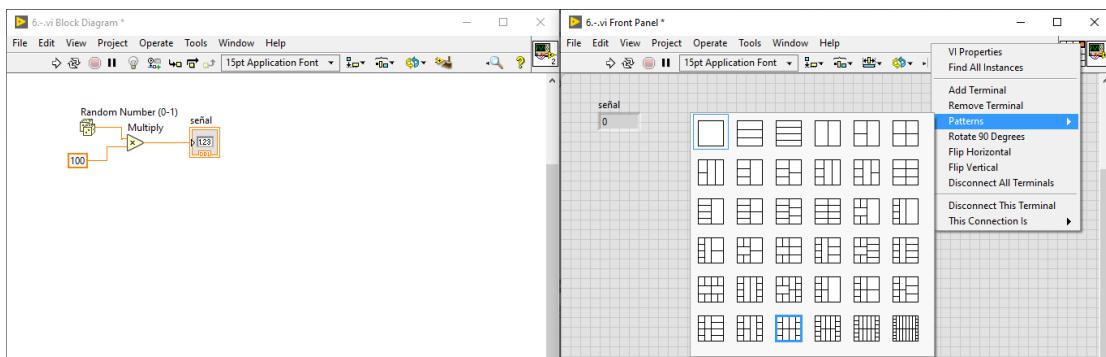
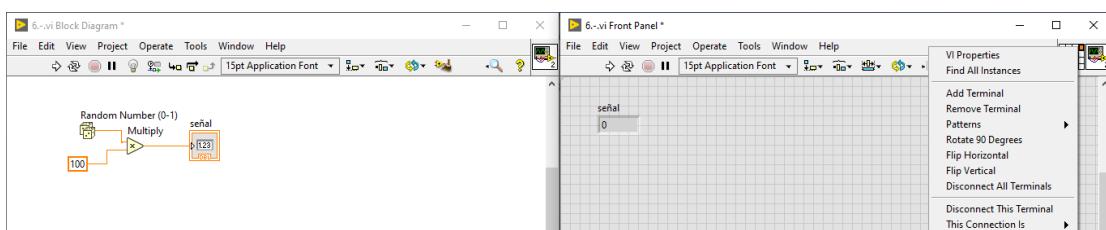
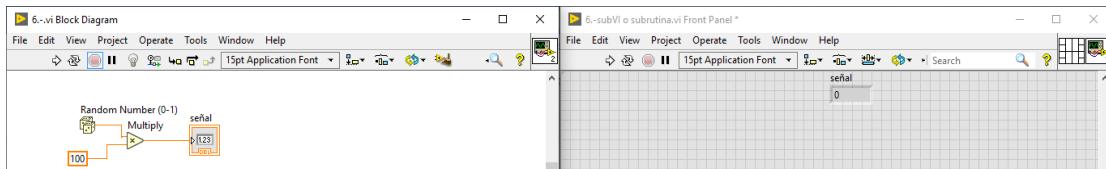
Ingresar a las Ventanas de un SubVI: Doble clic Sobre el Bloque

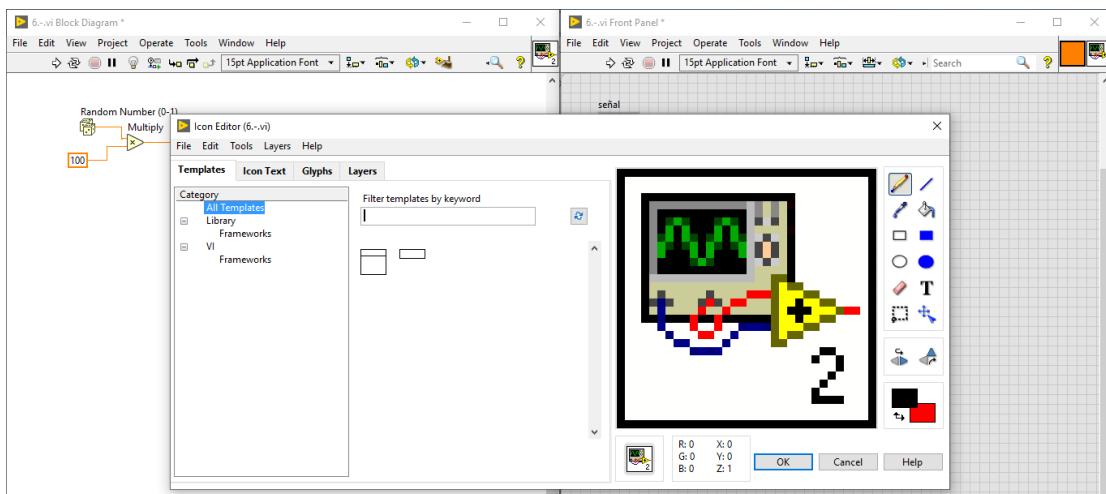
Ahora le daré doble clic sobre el bloque del SubVI se me abrirán sus ventanas de Block Diagram (código de bloques) y/o Front Panel (Interfaz de la SubVI).



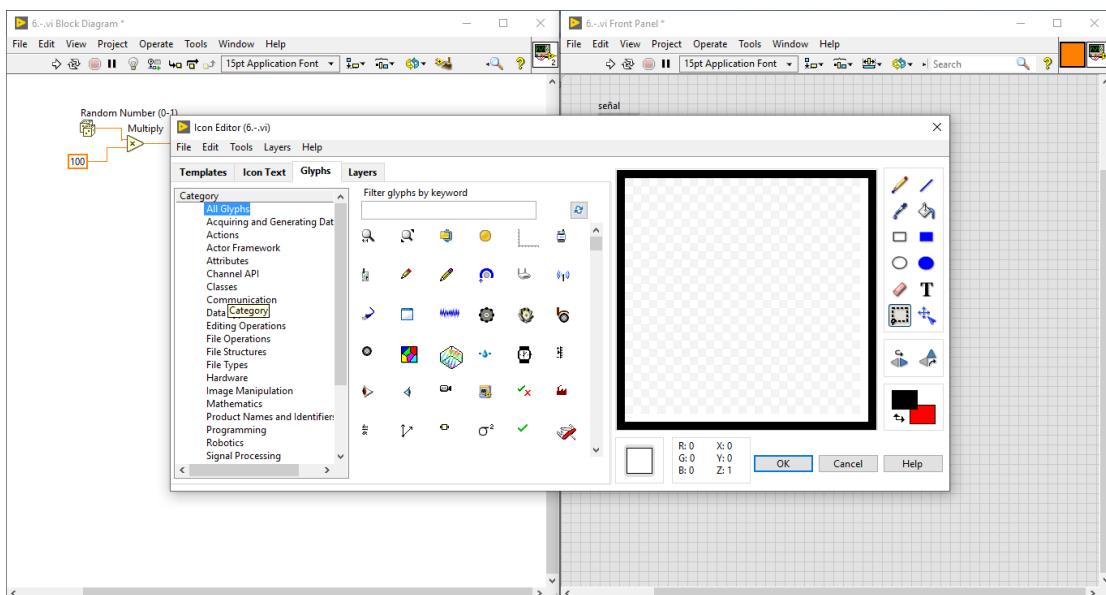
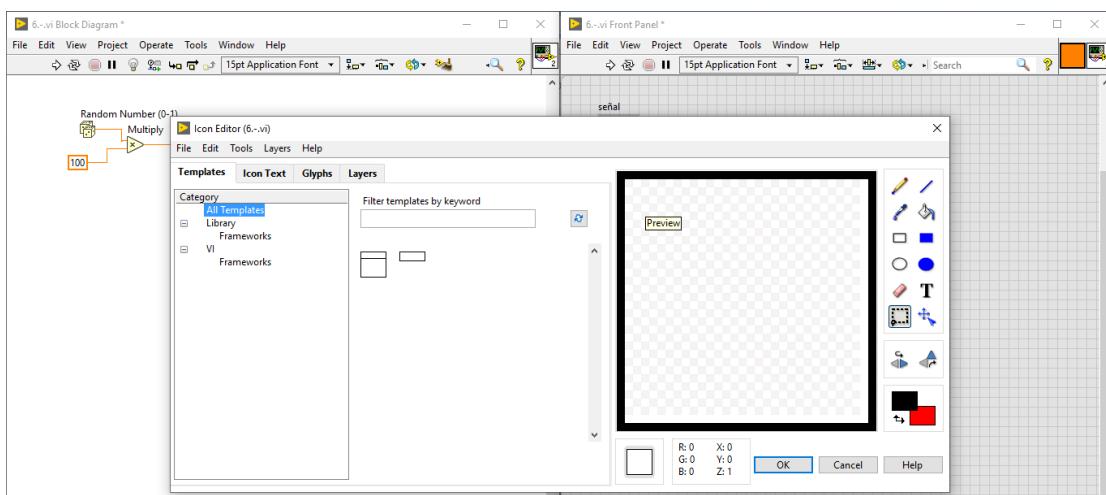
Editar Logo de un SubVI: Ícono Superior Derecho

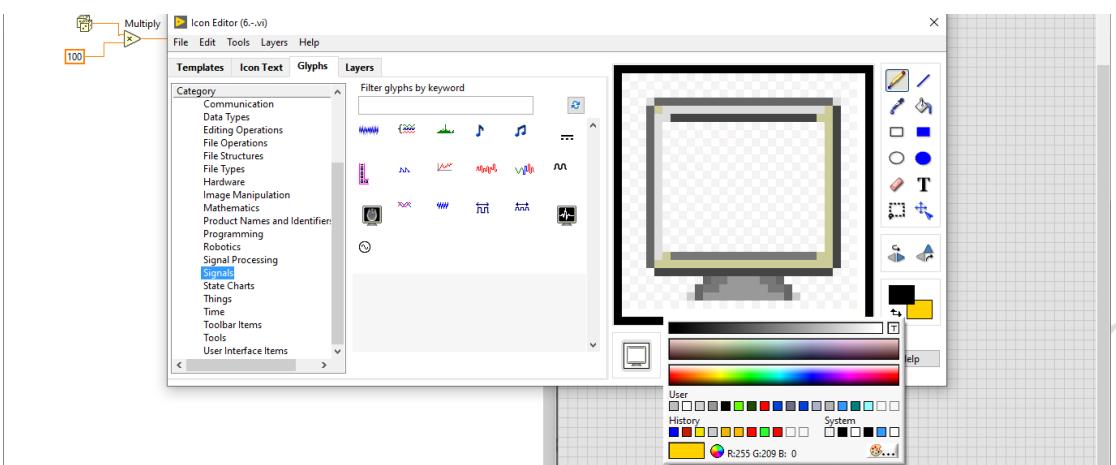
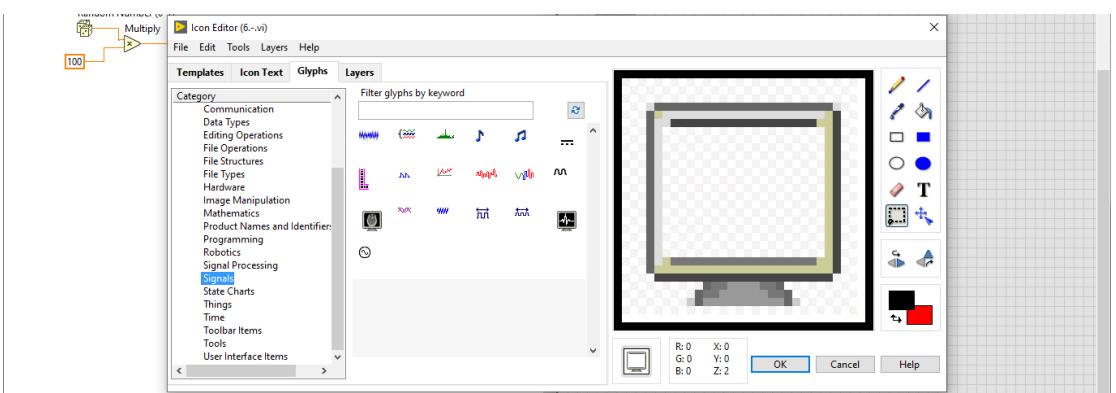
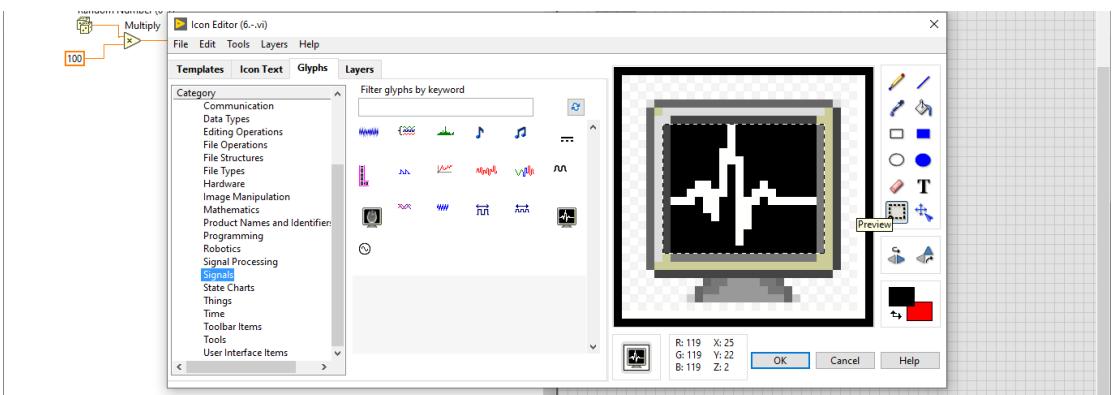
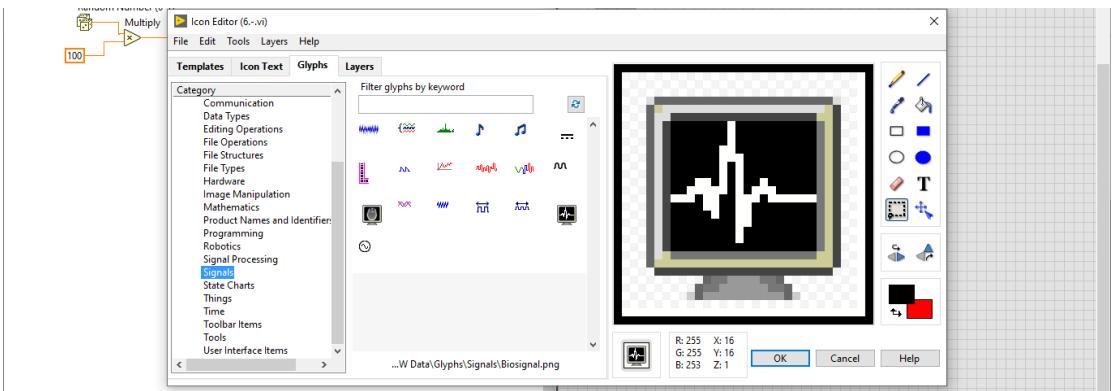
Ahora se va a editar aspectos del subVI como el Logo, para ello podemos editar los Patterns para ver la forma en la que está dividido el logo y luego en la parte de alado daremos doble clic para modificar su ícono.

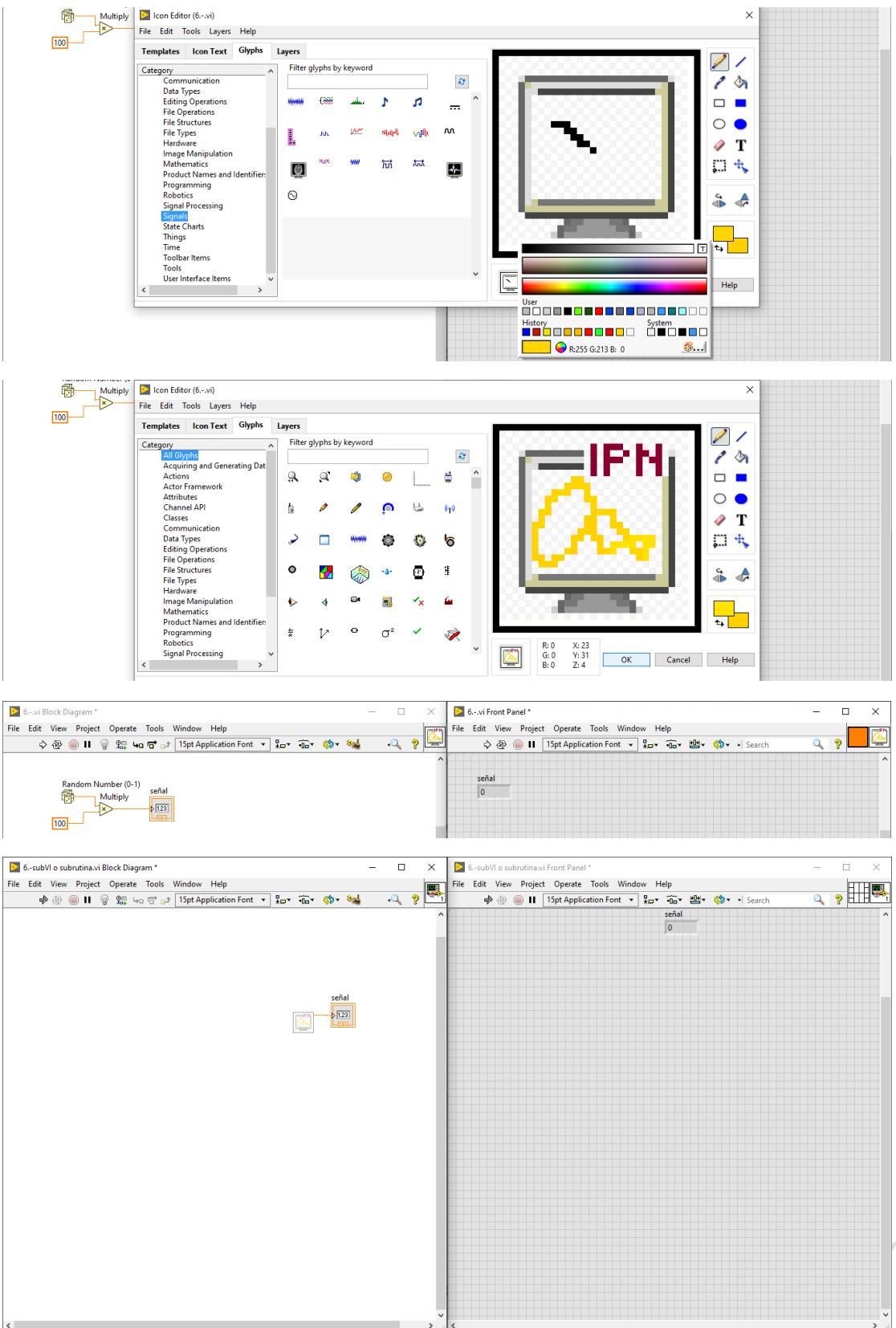




Le doy suprimir y aquí podemos crear un nuevo ícono.

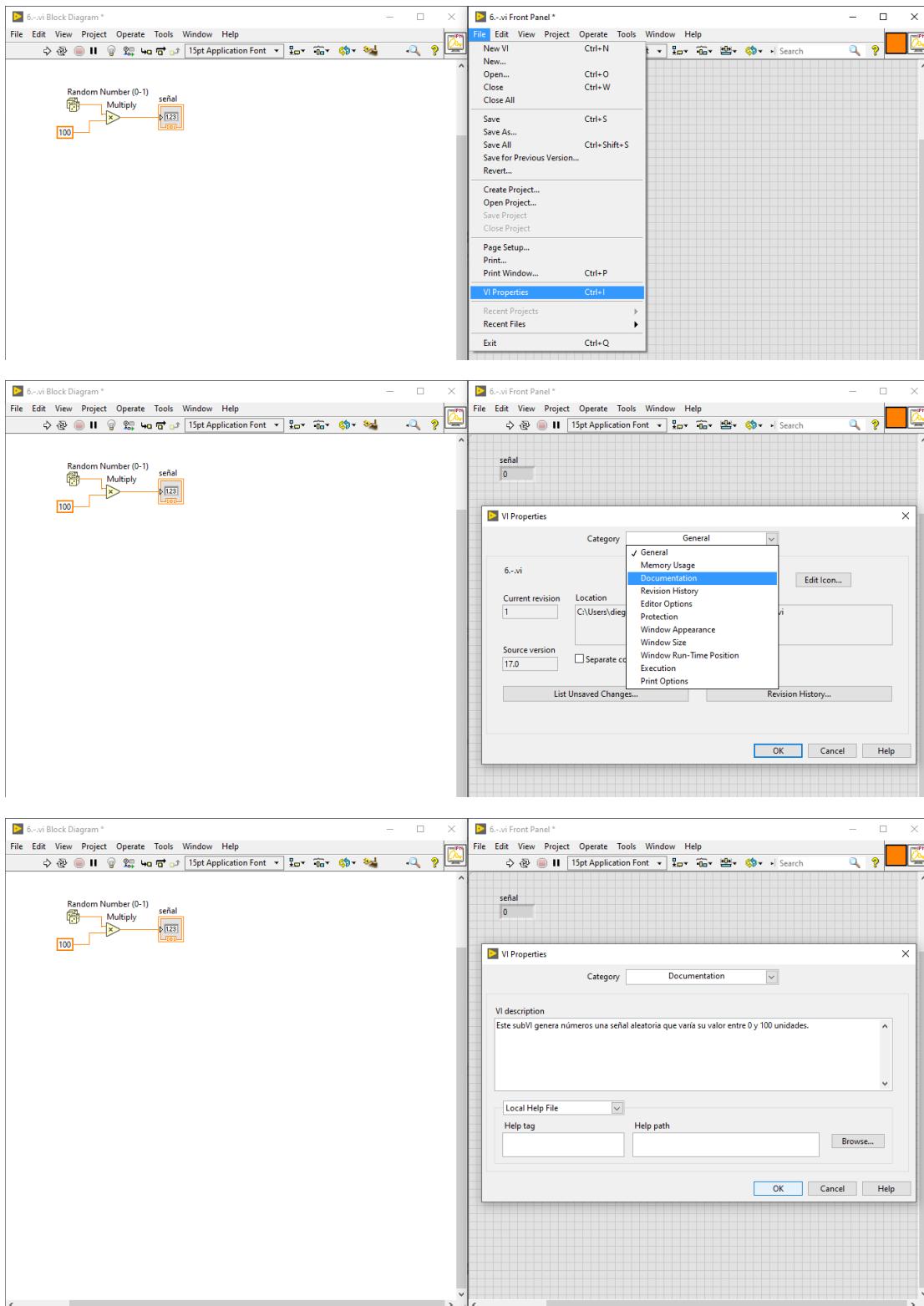


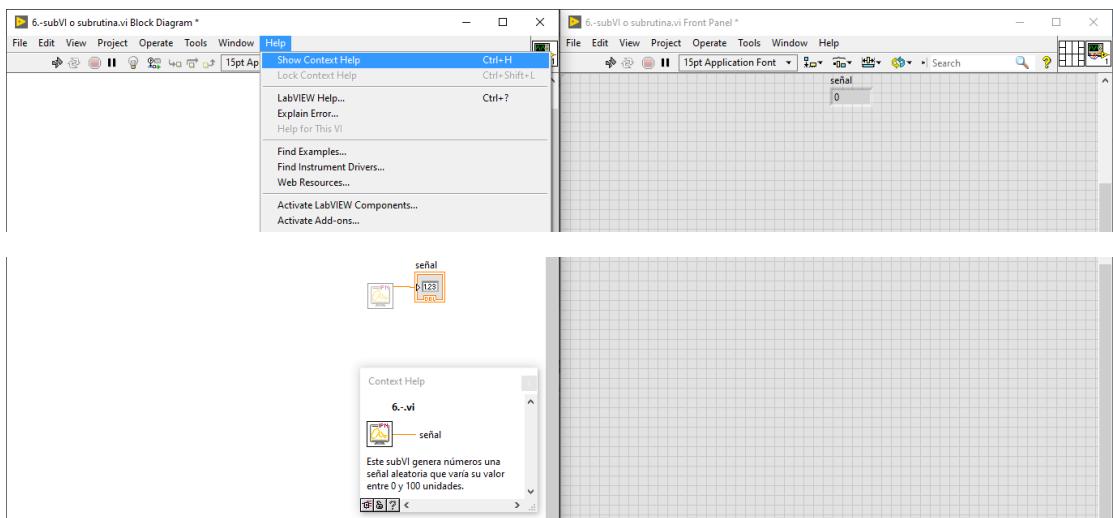




Editar la Descripción de un SubVI: Descripción que Aparece con Show Context Help

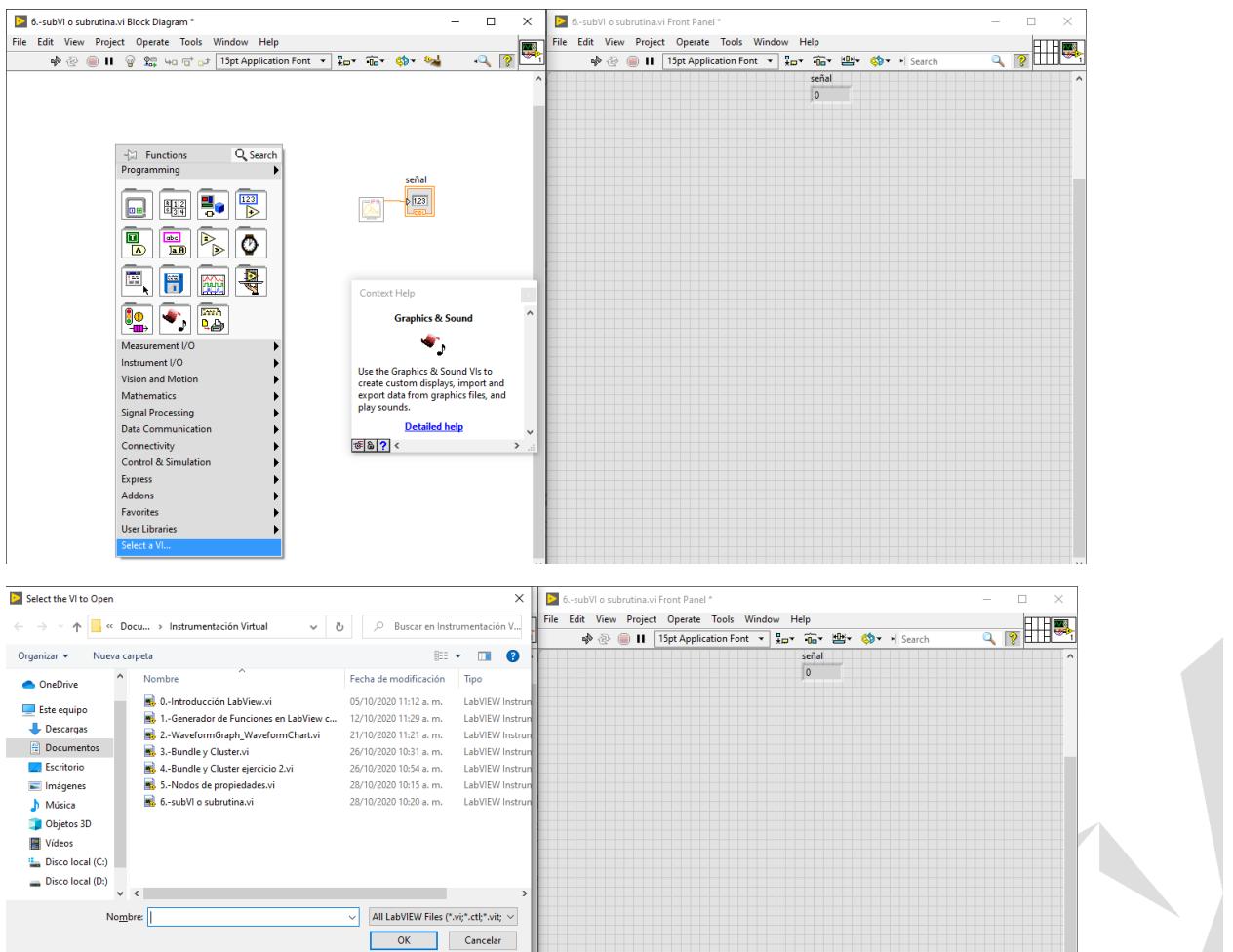
Se le puede asignar una descripción al SubVI, la cual aparecerá cuando se seleccione la herramienta de Help → Show Context Help.



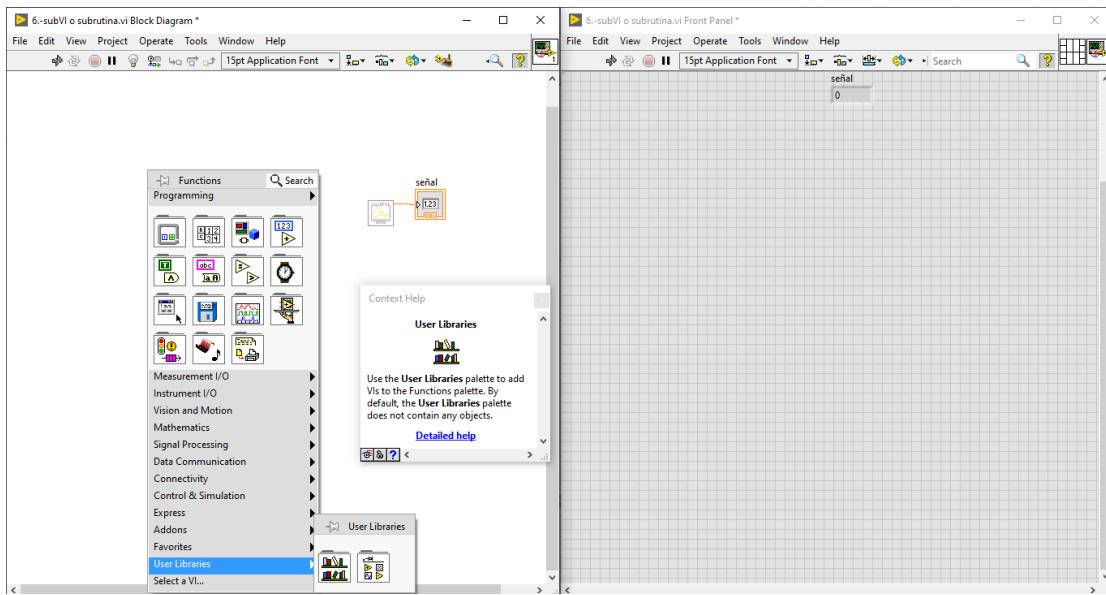


Guardar un SubVI: Como un VI Cualquiera o Para que se Pueda Acceder como Bloque

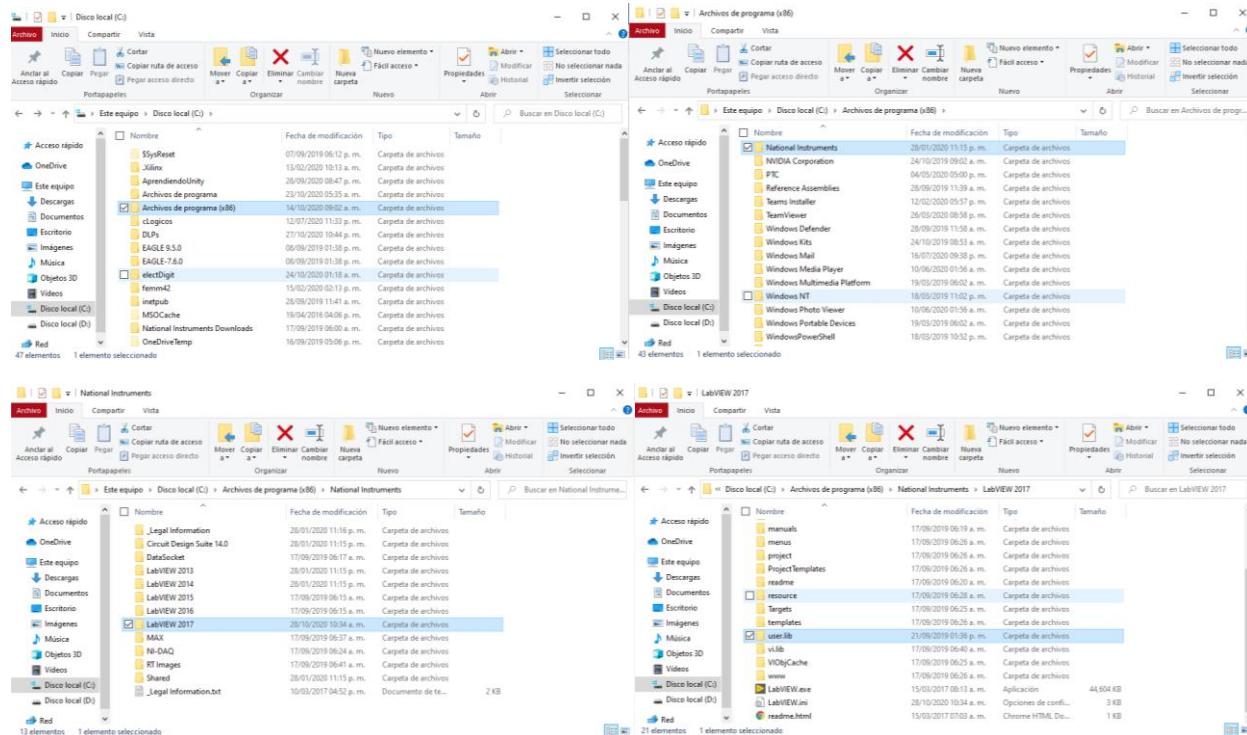
Ahora vamos a guardar el subVI, lo podemos guardar en cualquier dirección de memoria en nuestra computadora para que lo podamos accesar por medio de la opción: Functions → Select a VI...

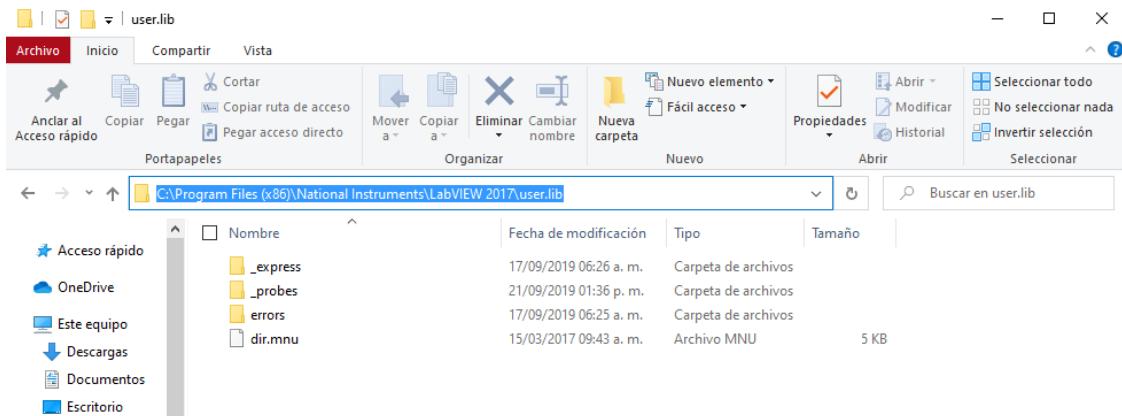


Ahora se puede guardar para que esté incluido en las librerías de LabVIEW y se pueda acceder de la siguiente manera desde el Block Diagram: Functions → User Libraries



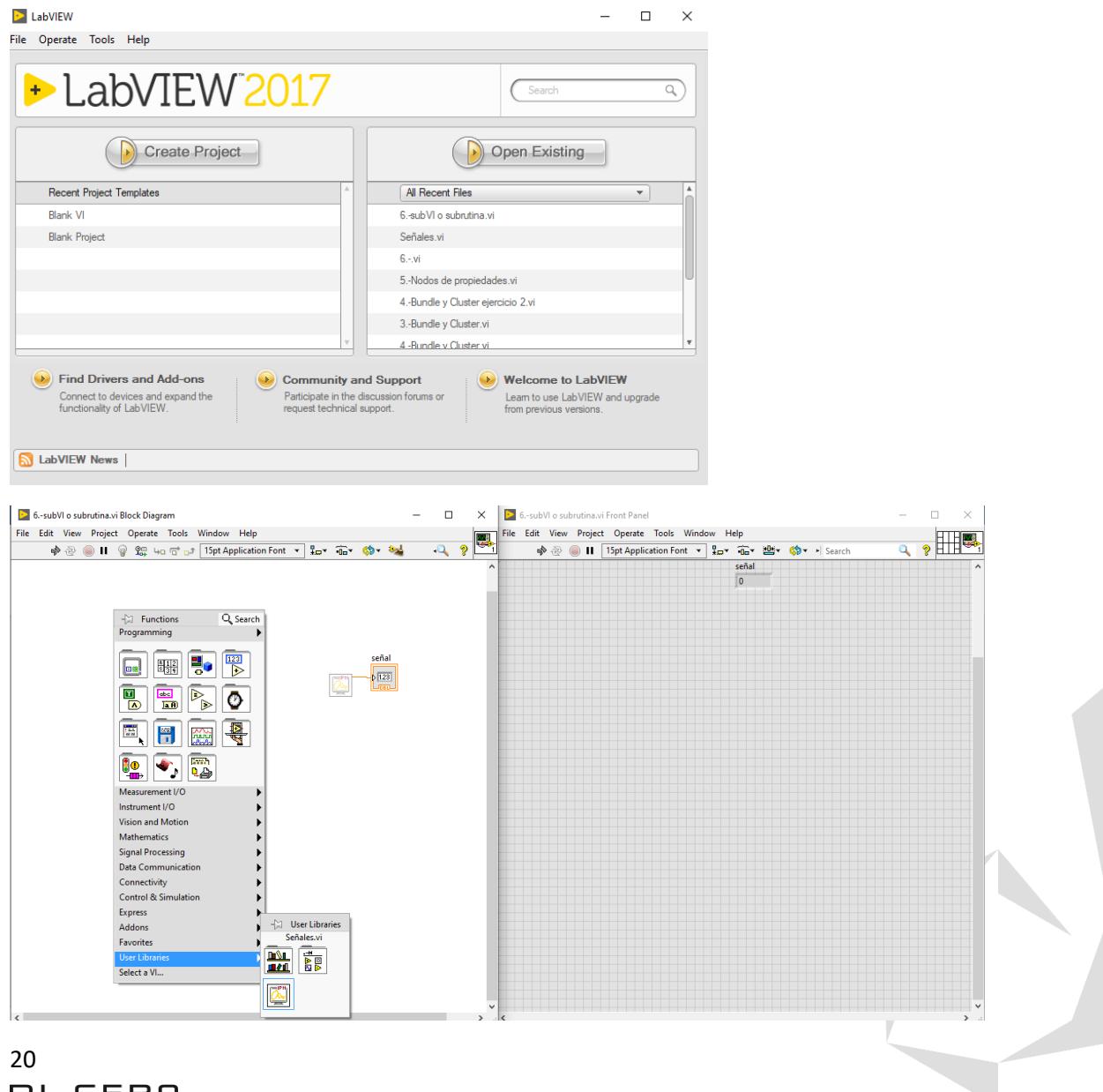
Para poder acceder un SubVI desde las librerías de LabVIEW debemos guardarlo en la siguiente ruta:
D:\Program Files (x86)\National Instruments\LabVIEW 2020\user.lib



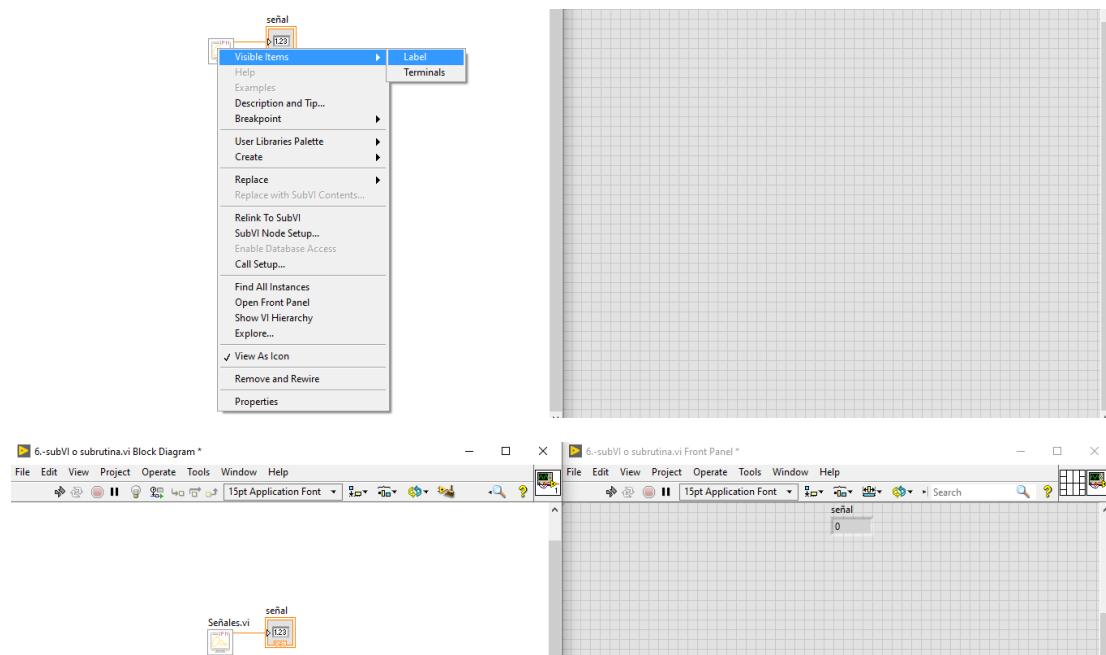


Utilización del SubVI en un VI Nuevo: Medidor de Nivel en Señal

Se va a cargar de la biblioteca de LabVIEW un subVI específico dentro de un VI nuevo.

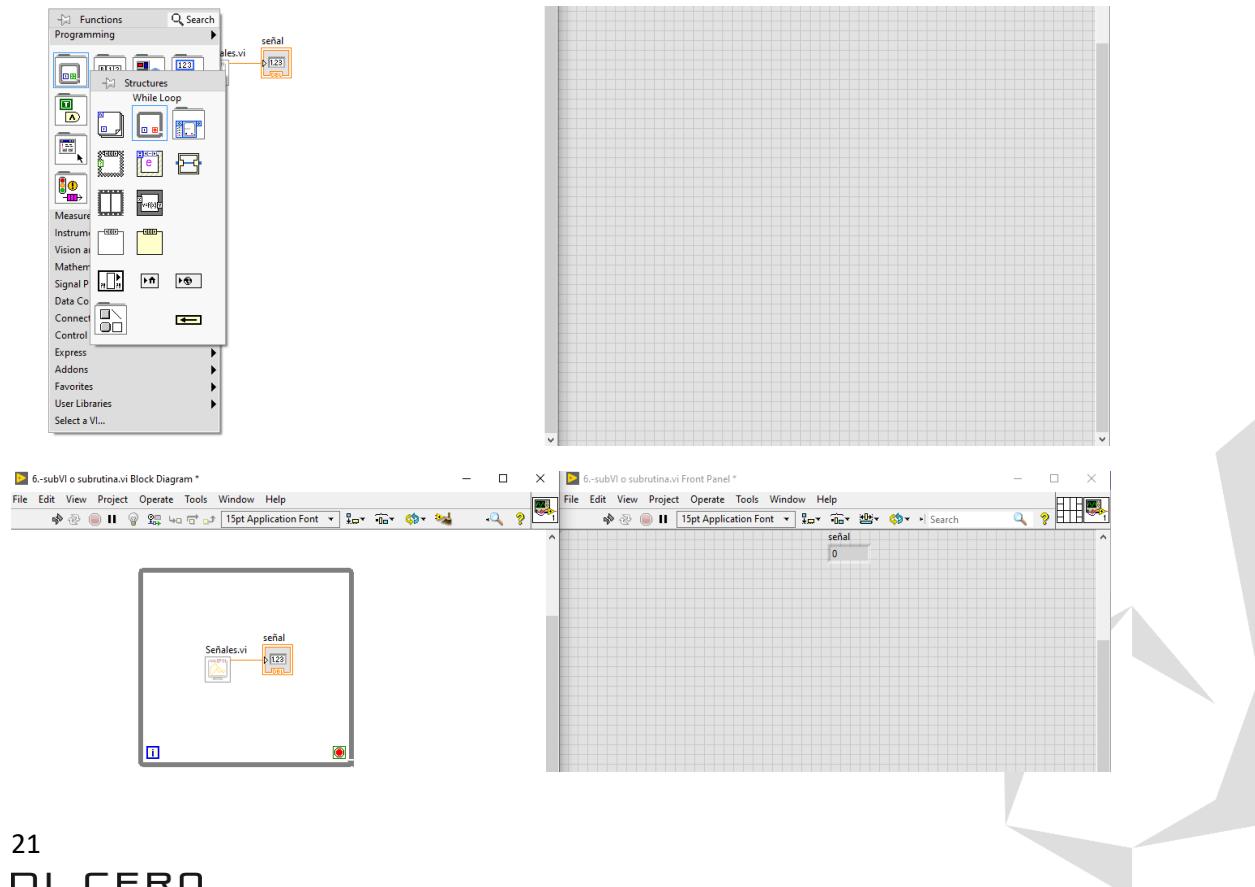


Mostrar nombre del bloque: Clic derecho → Visible Items → Label.

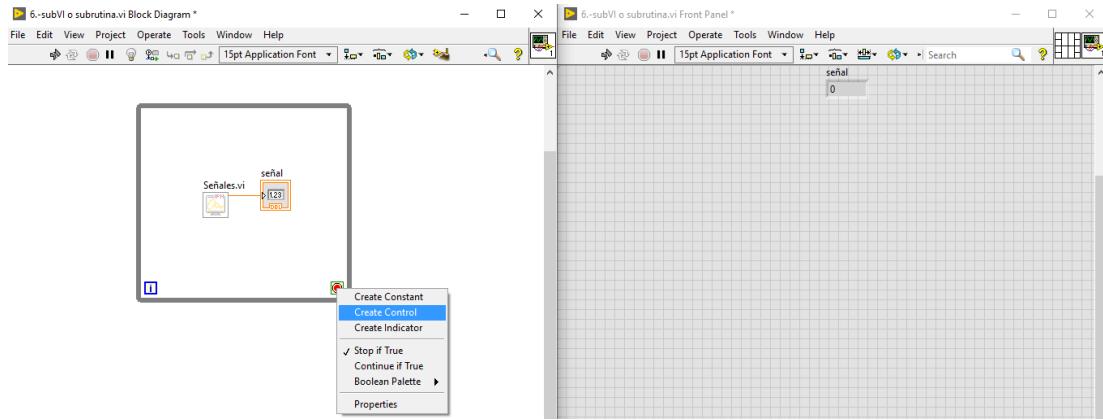


Block Diagram - Bucle While: Ejecución Continua del Programa

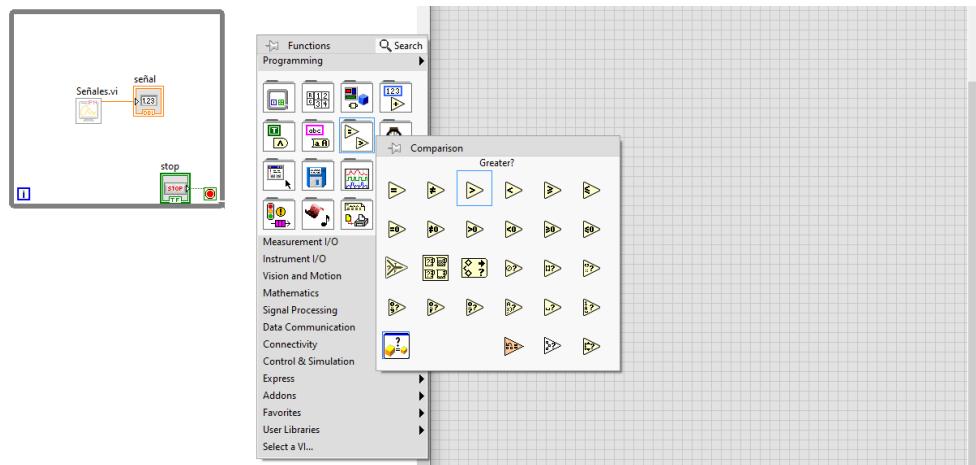
El ciclo while hace que el programa se ejecute hasta que dé clic en el botón de STOP, por eso todo el diagrama de bloques que tengo actualmente lo voy a encerrar en un ciclo while para que esté ejecutando de manera continua.



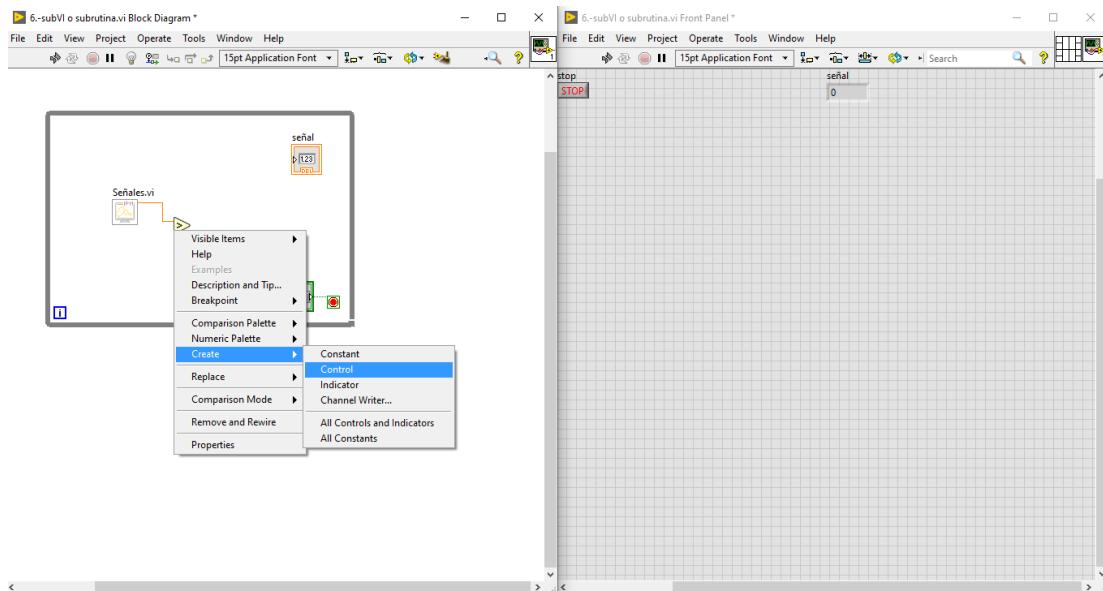
Crear un Control para un Bloque: Clic derecho en el bloque → Create Control.

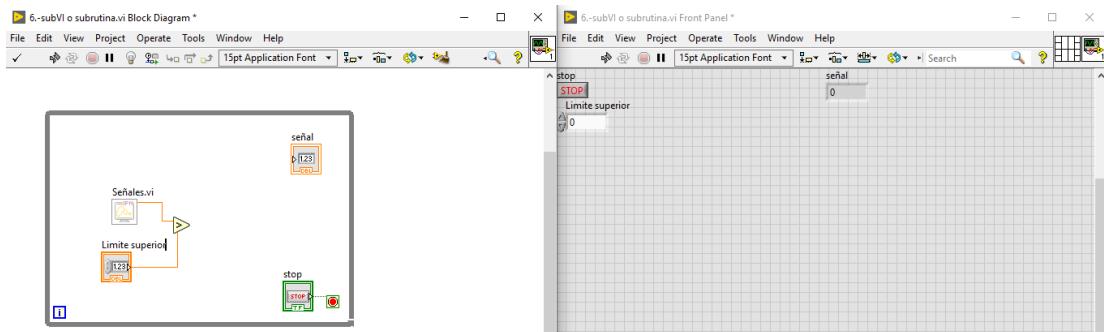


Block Diagram - Greater?: Operación Lógica Mayor Que (>)



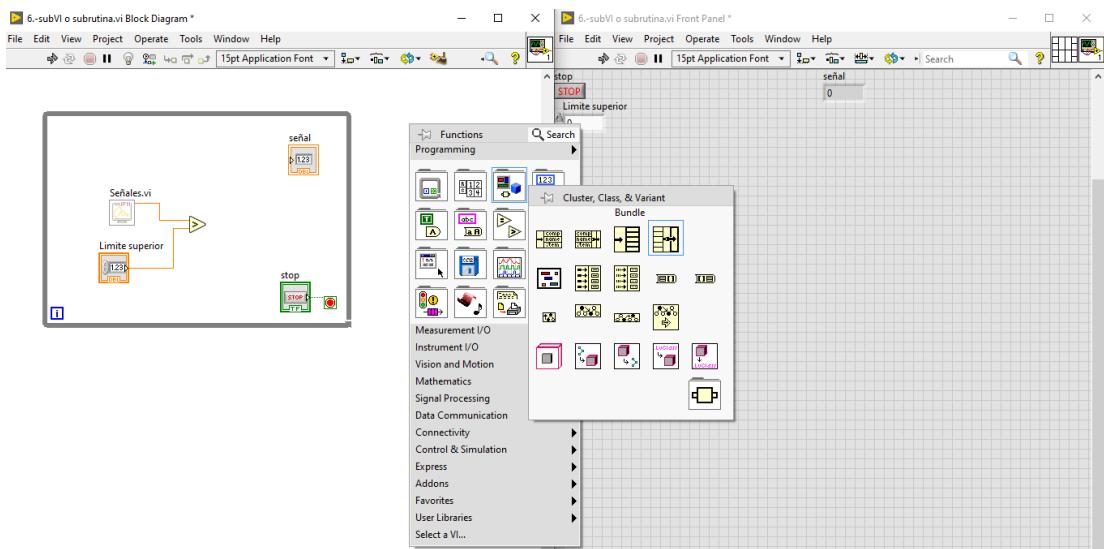
Crear un Control para un Bloque: Clic derecho en el bloque → Create Control.



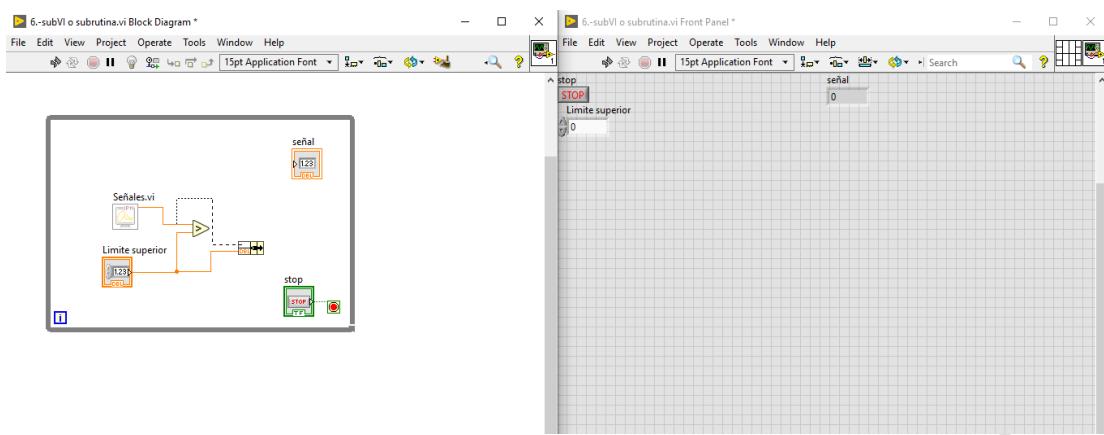


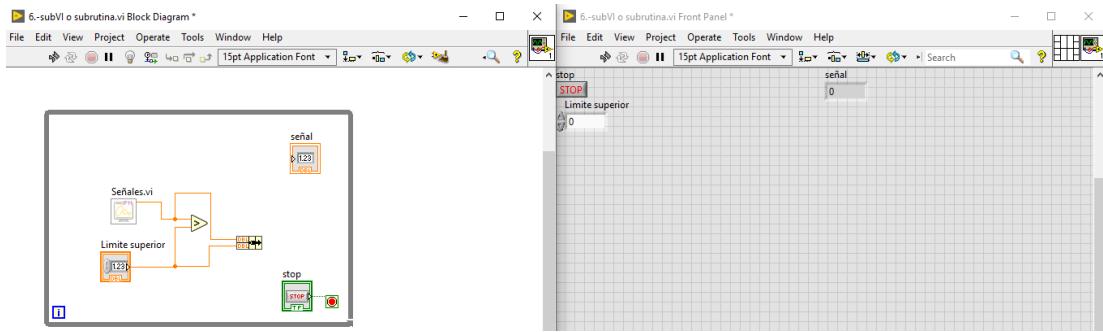
Block Diagram - Bundle: Juntar Varios Tipos de Datos Para Mandarlos a un Cluster

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.

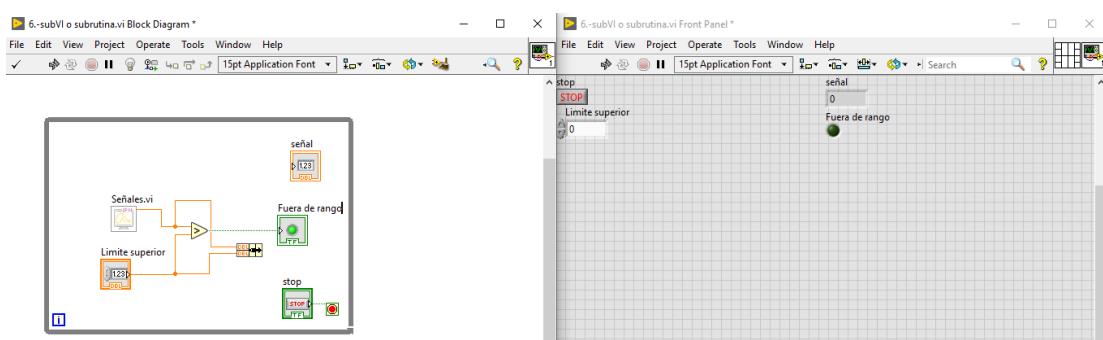
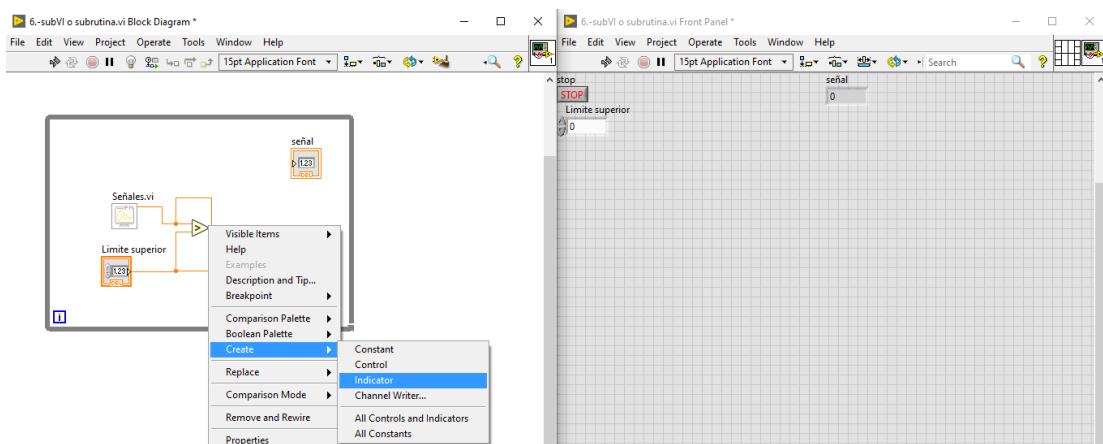


Lo que está pasando en este momento es que se está generando una señal (Dynamic Data) de números aleatorios que cambian al pasar del tiempo y se están comparando con cierto nivel de Límite Superior, donde además al final ambos valores de la señal y el límite se meten en un conjunto Bundle.



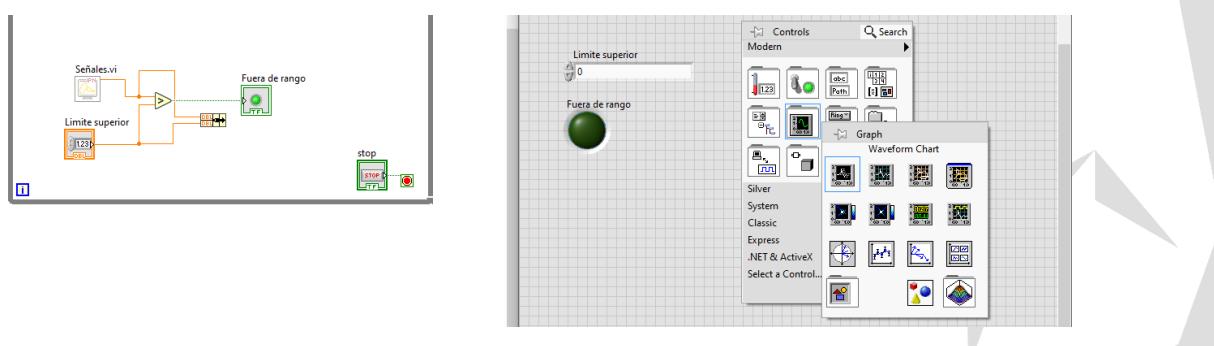


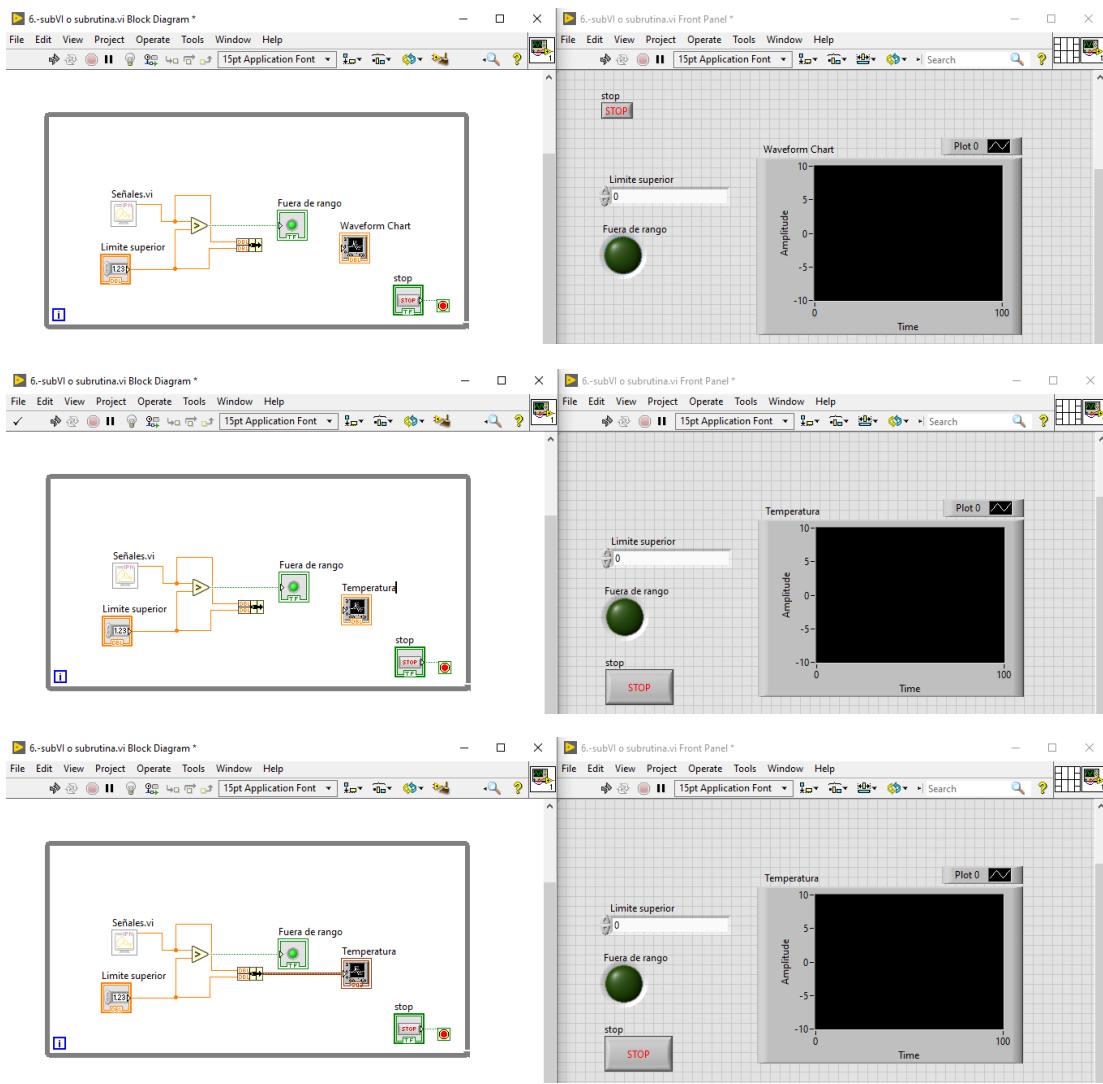
Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator. Con esto se crea un led que indique cuando la señal pase el Límite superior.



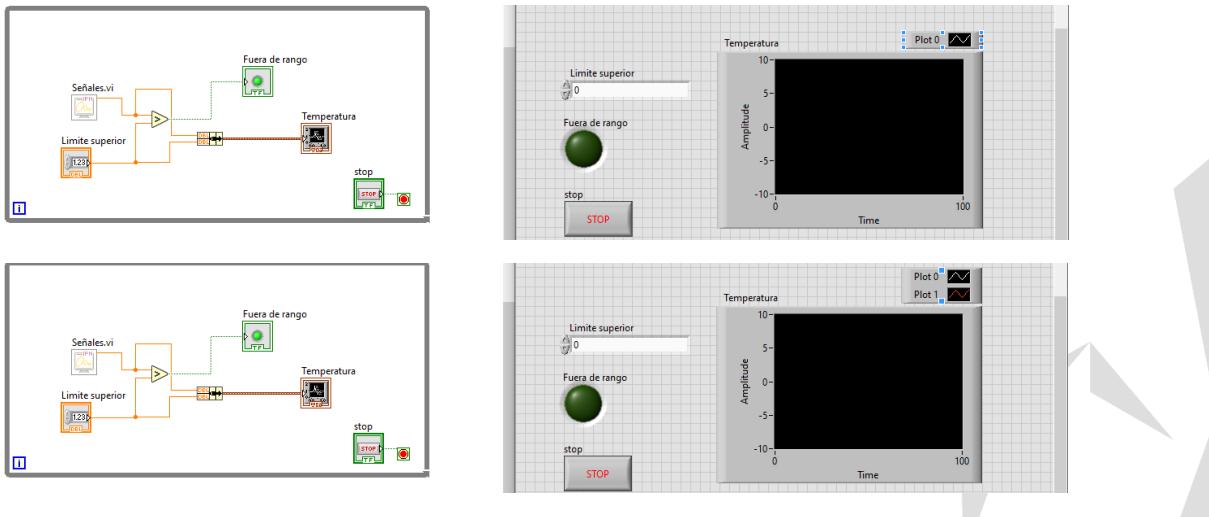
Front Panel - Waveform Chart: Ventana que Muestra una Señal (Dynamic Data)

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

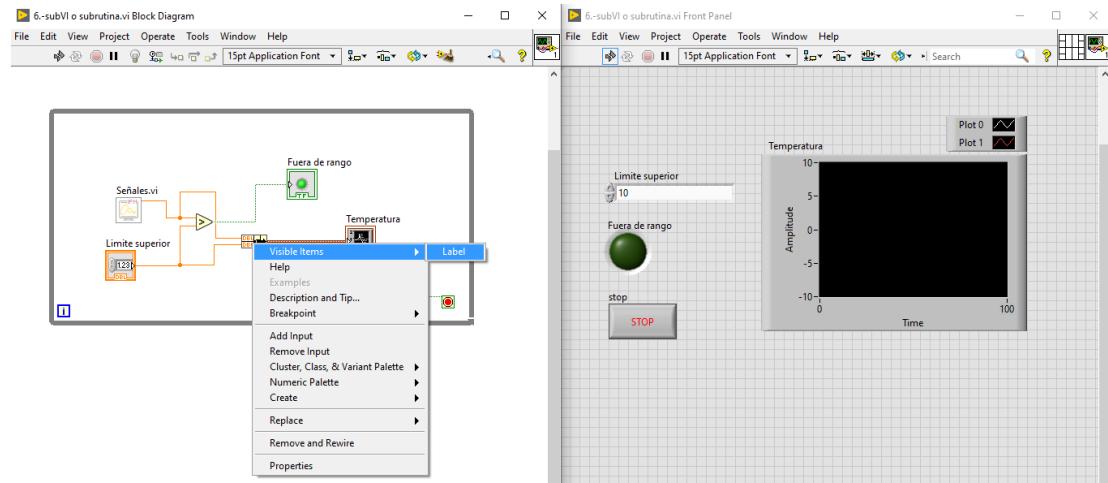




Como se busca tener dos señales dentro del Waveform Chart se debe expandir hacia arriba el menú superior derecho del bloque, donde dice Plot 0.



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



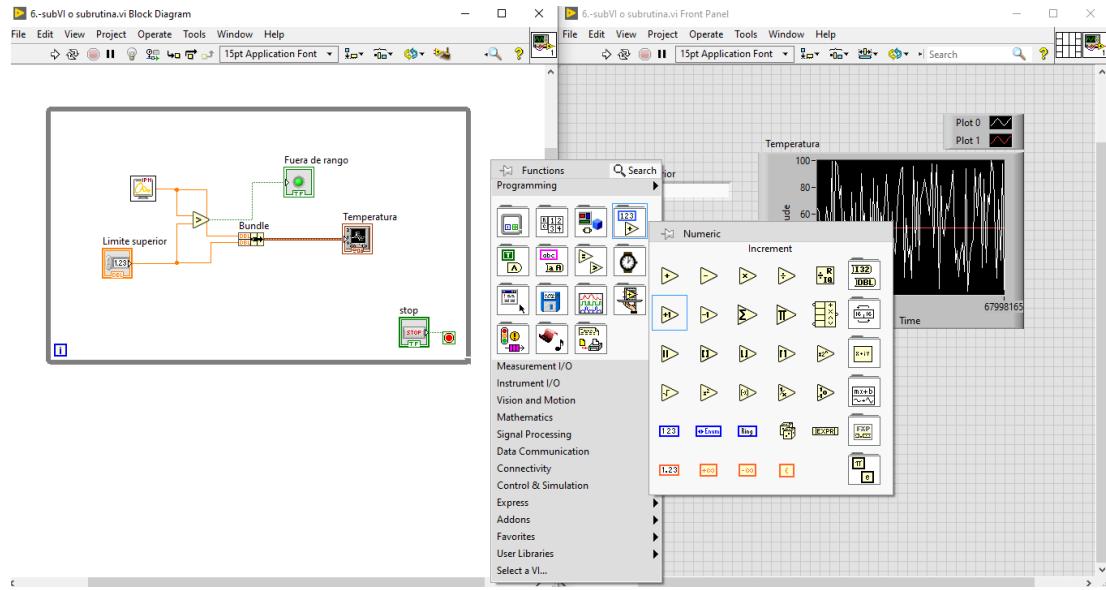
Ejecución del Programa: Gráfica de Señal de SubVI con Límite Superior

El límite superior de la gráfica se debe poner en el control de la interfaz y cada que la gráfica lo supere el led de Fuera de Rango se encenderá.

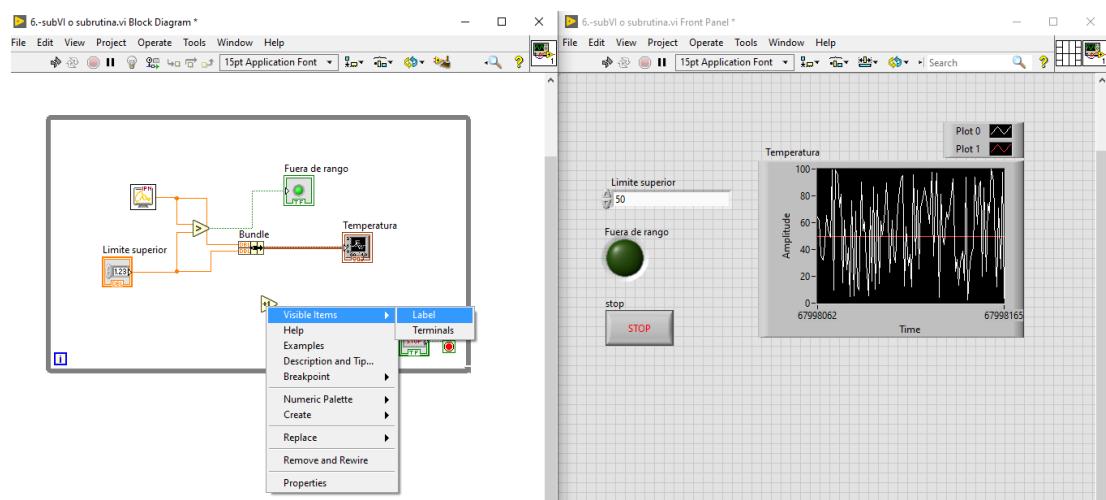
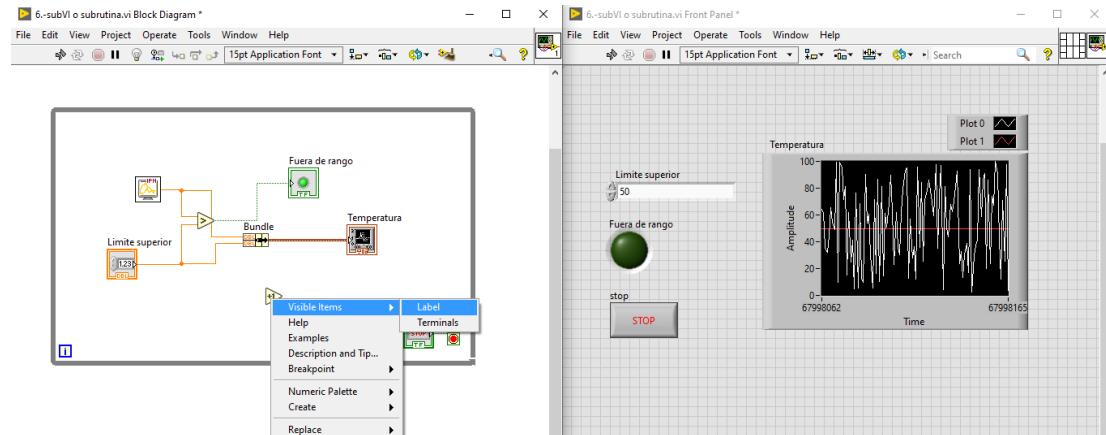


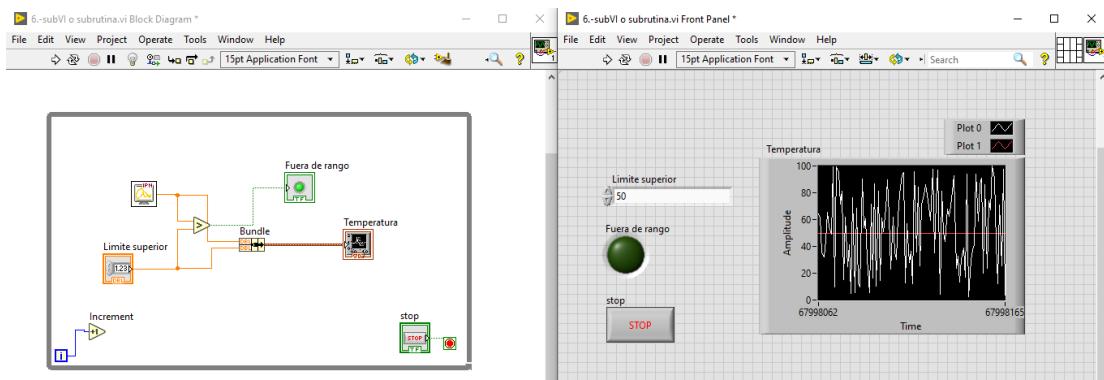
Block Diagram - Increment: Sumar +1 en cada Ejecución del Bucle

Esto lo hacemos para que la primera iteración empiece desde 1 no desde cero como siempre lo hace.

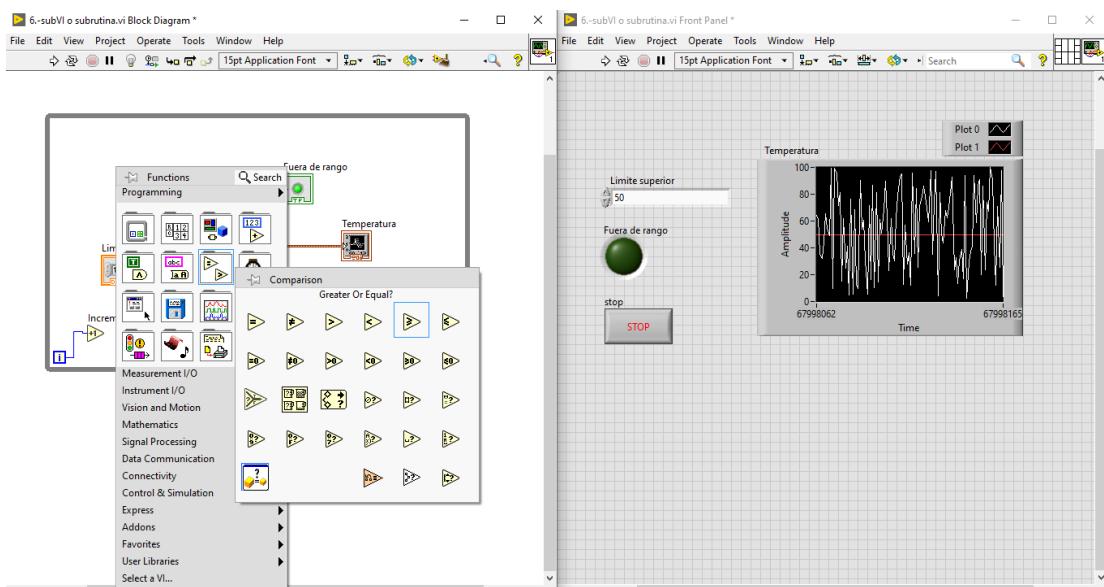


Mostrar nombre del bloque: Clic derecho → Visible Items → Label.

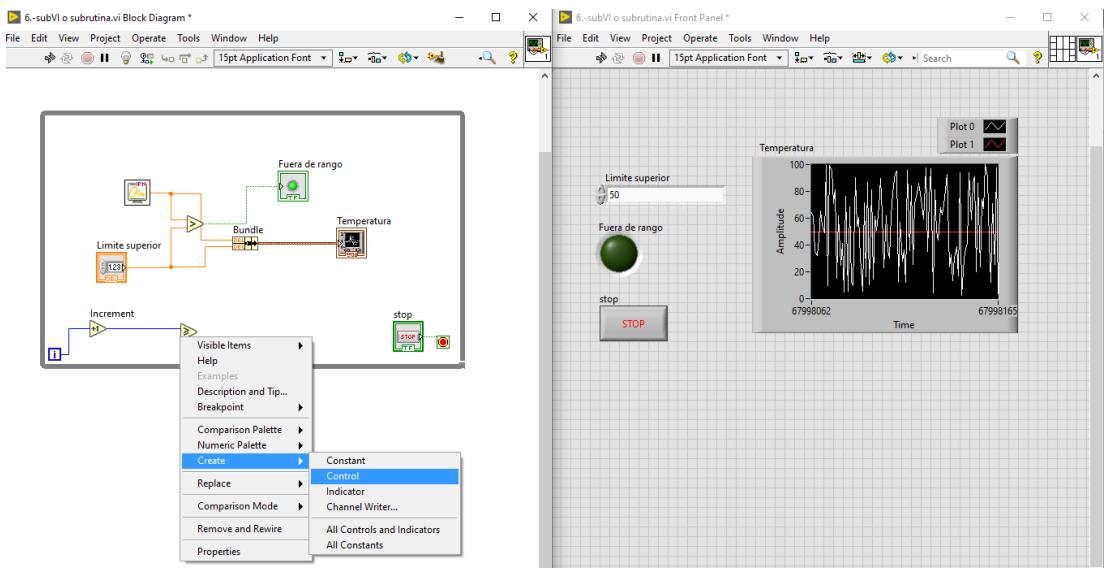


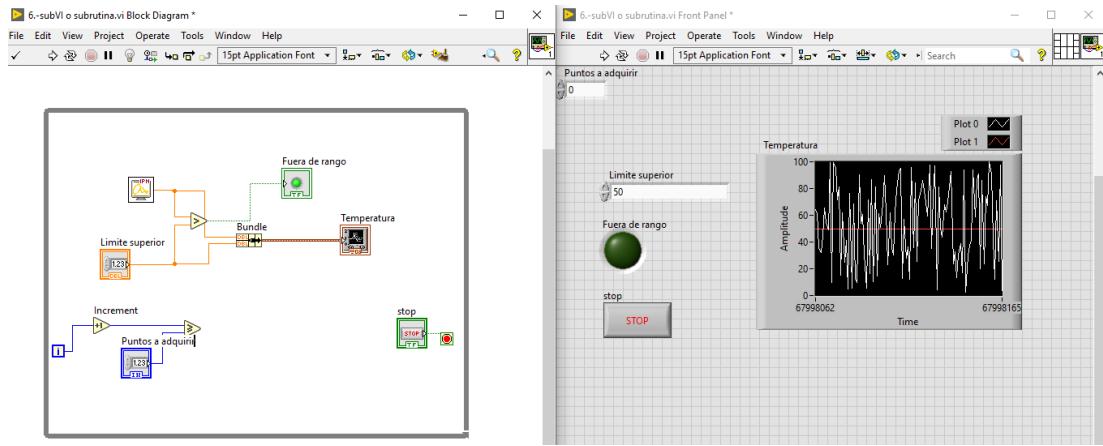


Block Diagram - Greater Or Equal?: Operación Lógica Mayor o Igual Que (\geq)

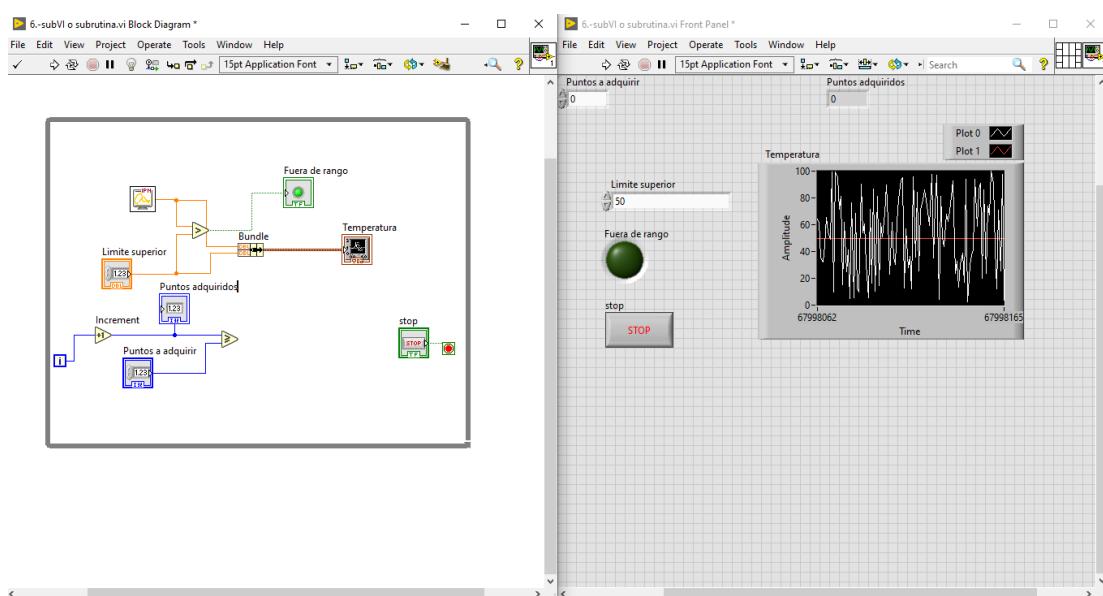
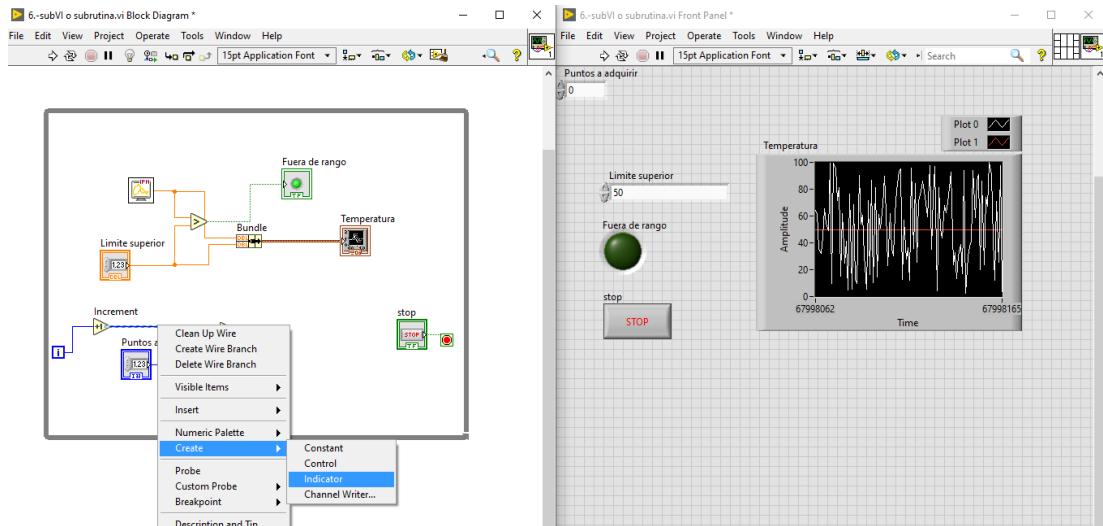


Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant.



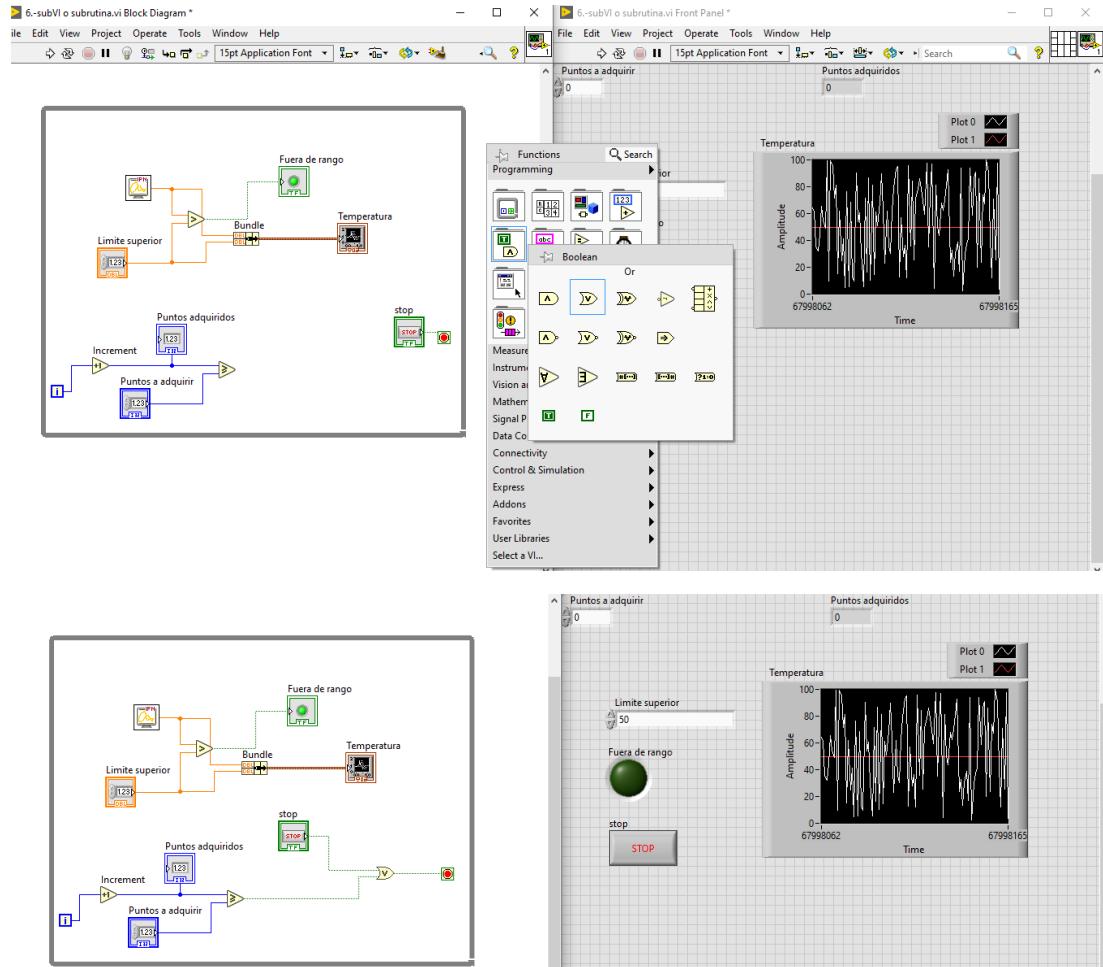


Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator.

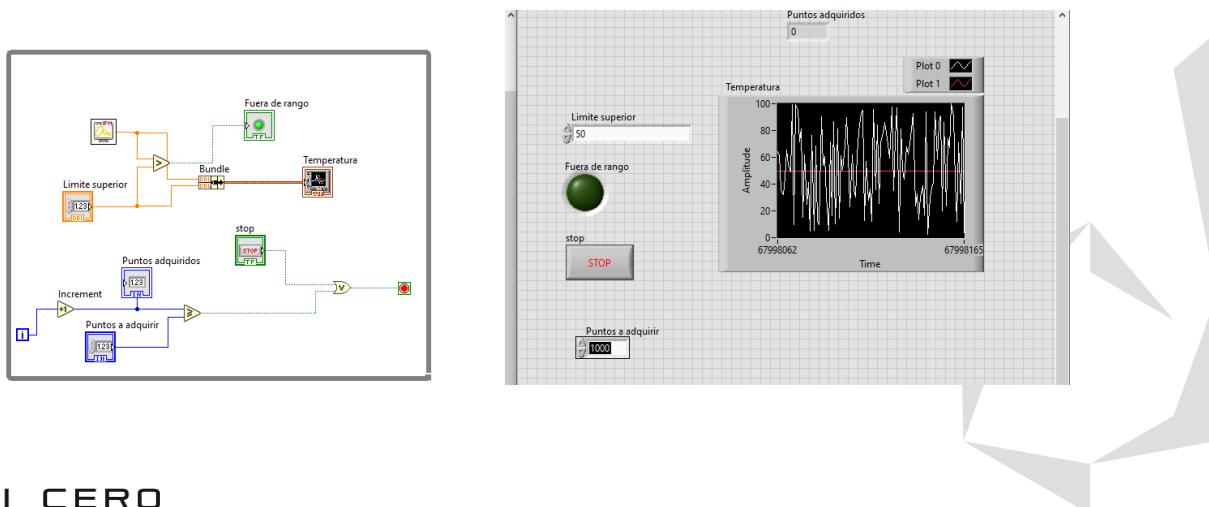


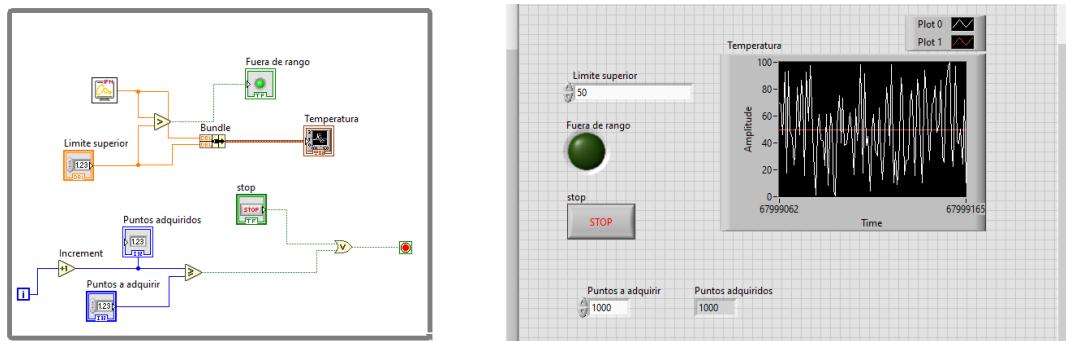
Block Diagram - Compuerta Lógica OR: Detener el Programa por un Error o con Stop

La compuerta OR se utiliza para que al definir los puntos que quiero adquirir de la gráfica (los valores de graficación a través del tiempo) y se me vayan mostrando en el Waveform Chart, de tal manera que cuando el valor de Puntos Adquiridos sea mayor o igual a Puntos a Adquirir, se pare solo el ciclo while y la ejecución del programa, pero que además pueda ser parado manualmente con el botón de STOP.



Así el sistema se puede parar cuando lo diga el usuario o cuando se alcance el límite de puntos a adquirir.

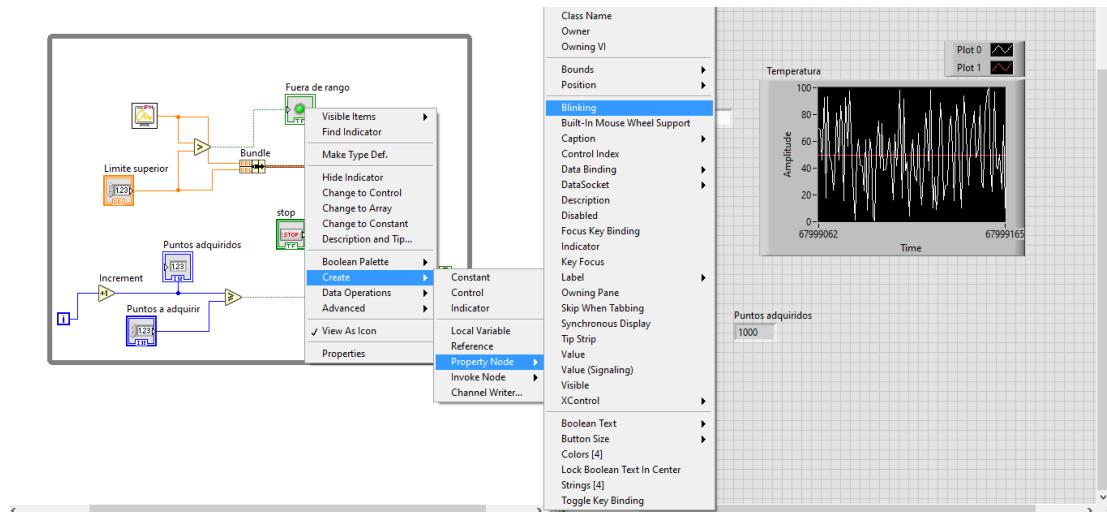




Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque

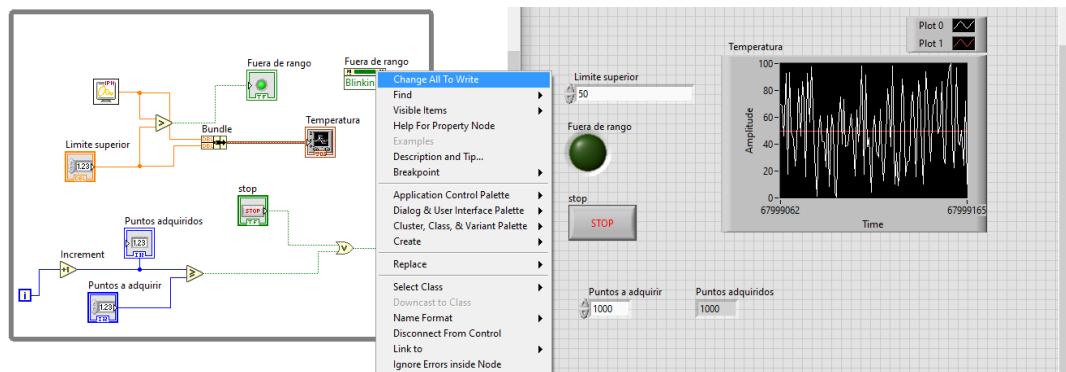
Ahora lo que voy a hacer es acceder a las propiedades del bloque led dando clic derecho en el bloque de la siguiente manera y seleccionando la opción de Create → Property Node → Blinking:

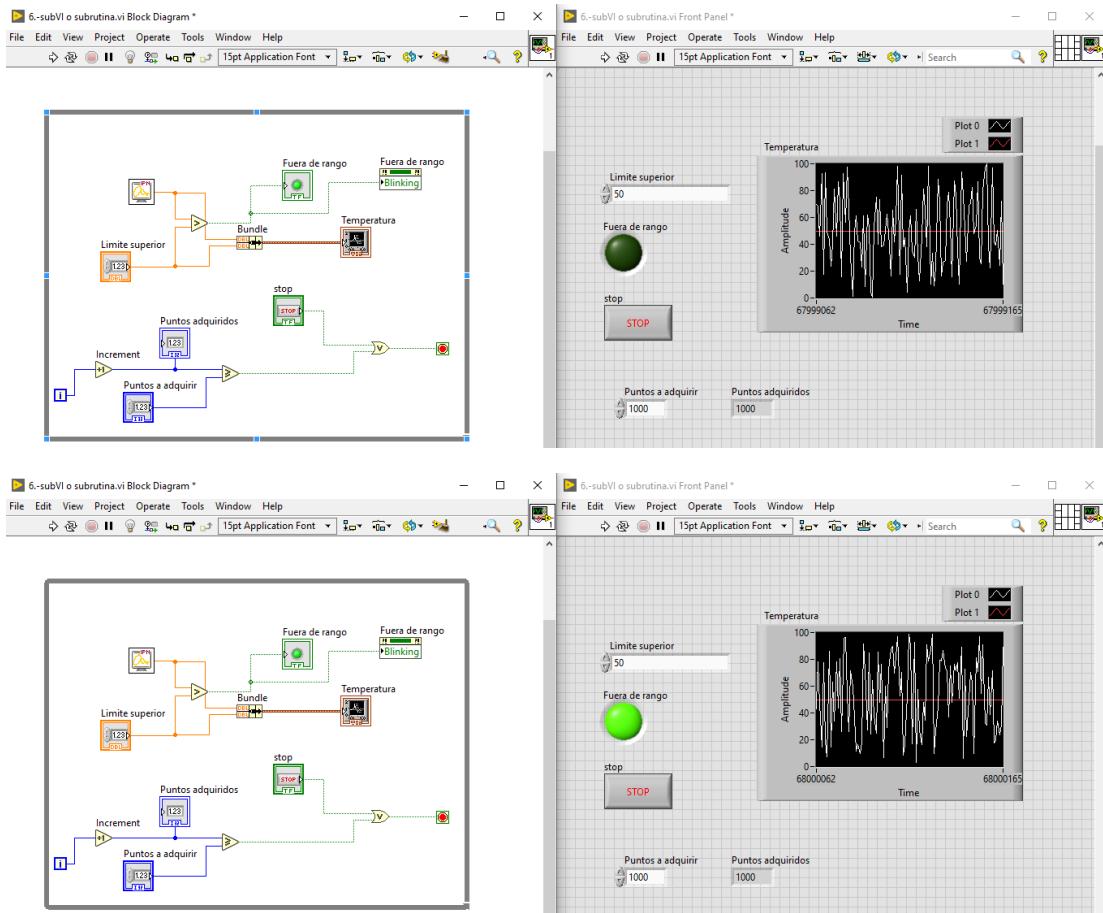
Recordemos que un Property Node es un elemento que accede a cualquier aspecto del bloque indicado, puede ser el historial, color, valor, **si este parpadea o no**, etc.



Este Property Node llamado Blinking lo que hará es lanzar un destello en el indicador, mostrando un color intermedio amarillo entre el apagado y el encendido del led.

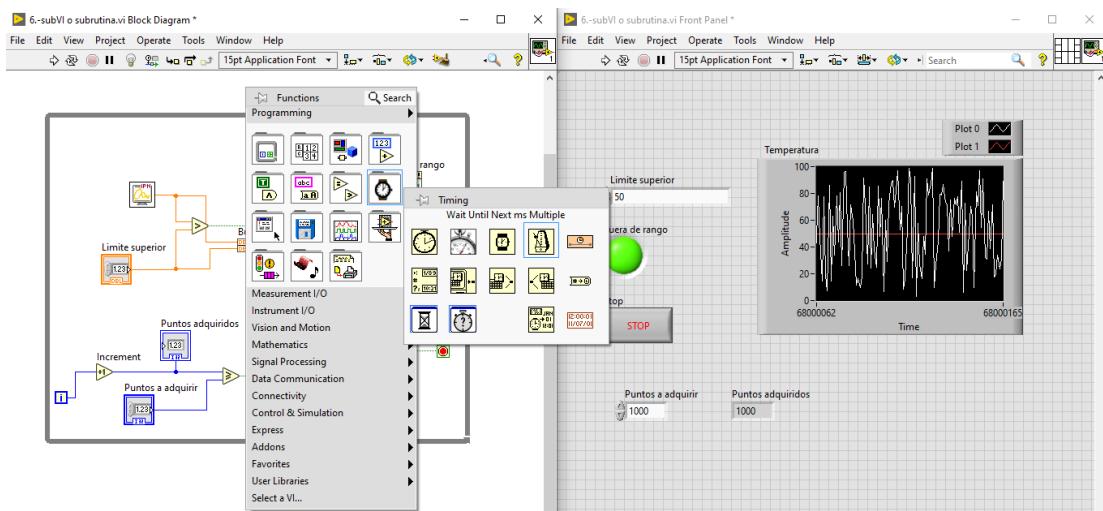
Se cambiará a escritura el Property Node para que pueda afectar el bloque del led con la opción Change All to Write.



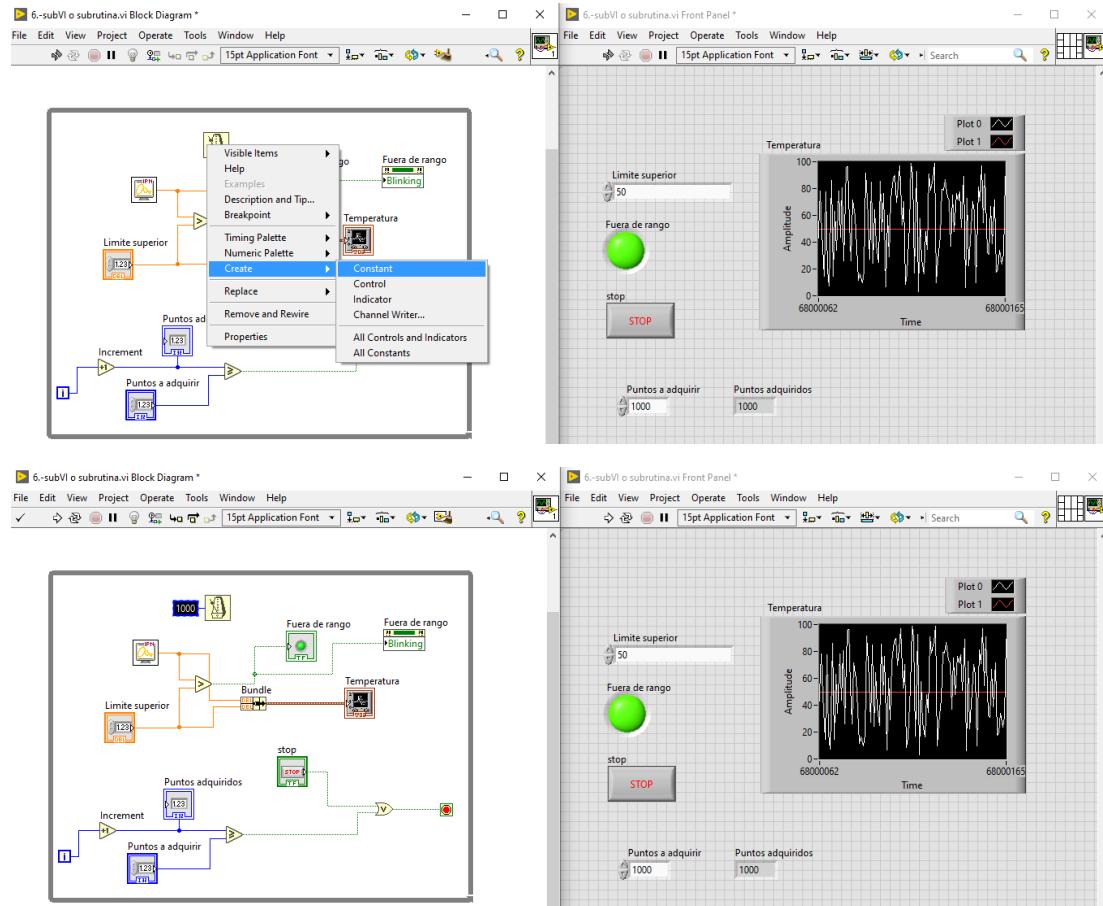


Block Diagram - Wait Until Next ms Multiple: Temporizador en milisegundos

El bloque de wait until se utiliza cuando se debe hacer un retraso de tiempo (delay) por ciertos segundos, para de esta manera parar la ejecución del programa por un cierto tiempo, en específico para que corra este bloque se debe crear una constante dando clic derecho sobre ella y declarando el tiempo en milisegundos.

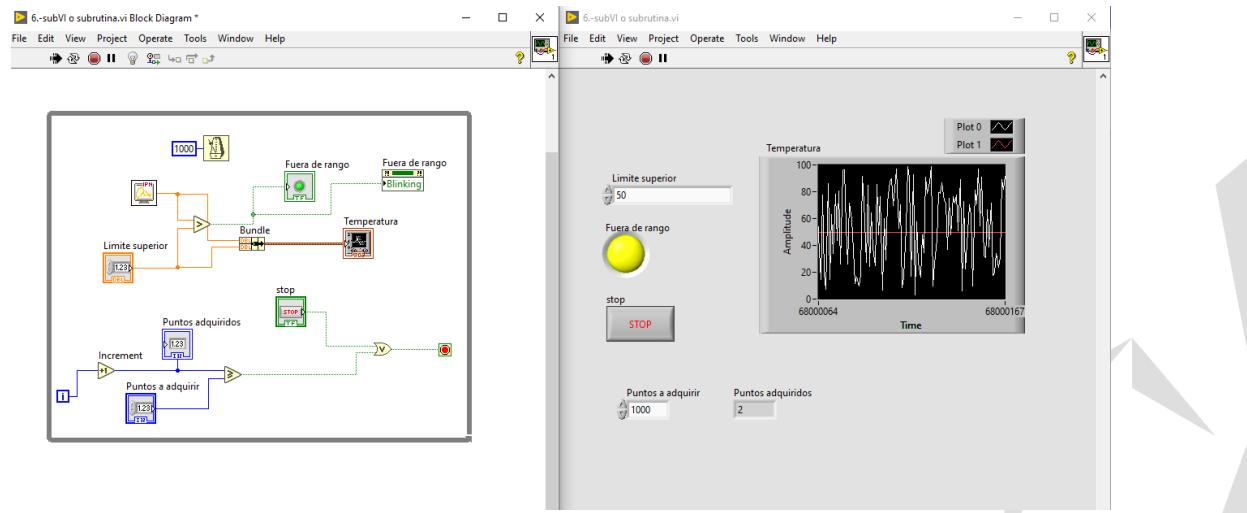


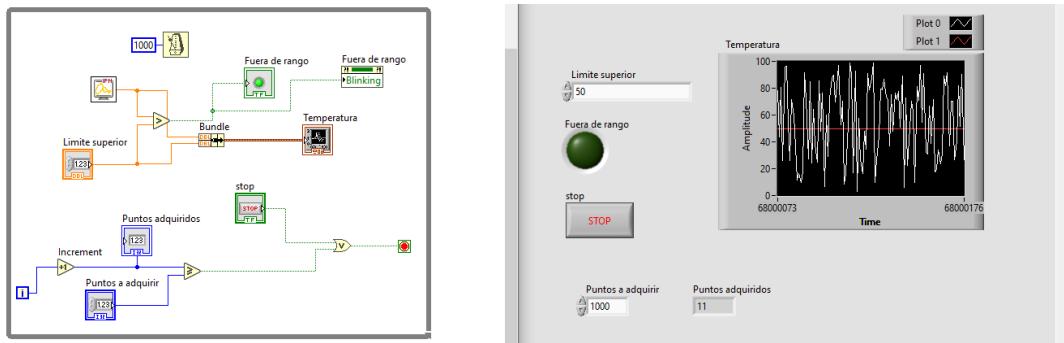
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Tiempo de duración del temporizador en milisegundos.



Ejecución del Programa: Señal con Límite Superior y Led que Parpadea

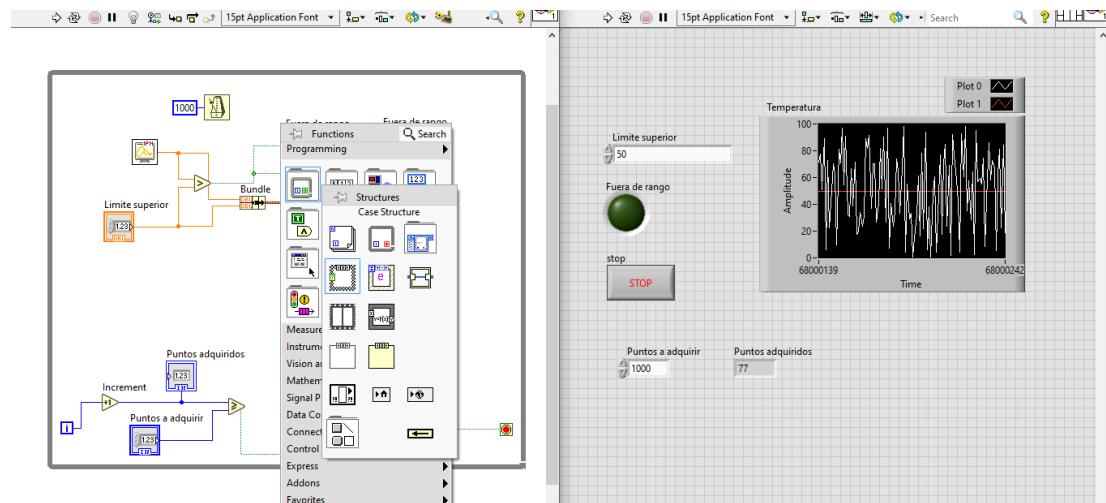
El límite superior de la gráfica se debe poner en el control de la interfaz y cada que la gráfica lo supere el led de Fuera de Rango se encenderá.



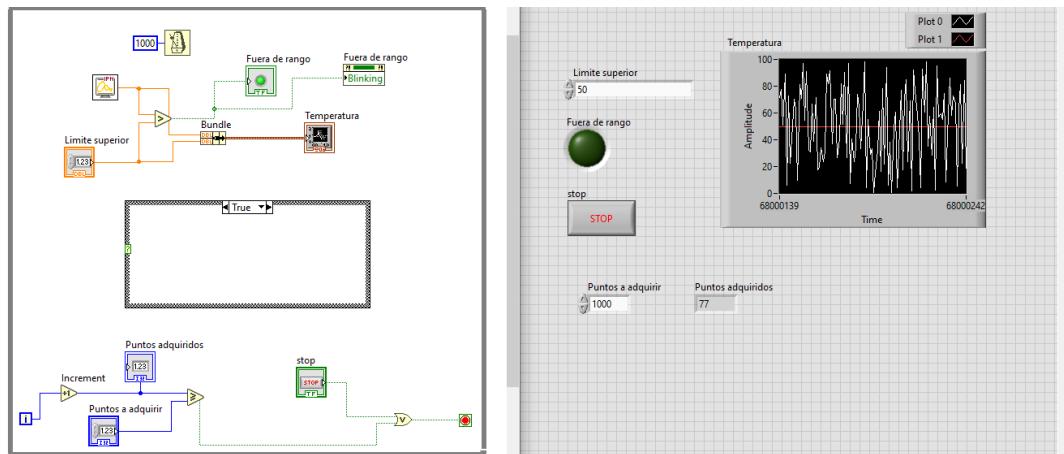


Block Diagram - Case Structure: Condicional Switch Case

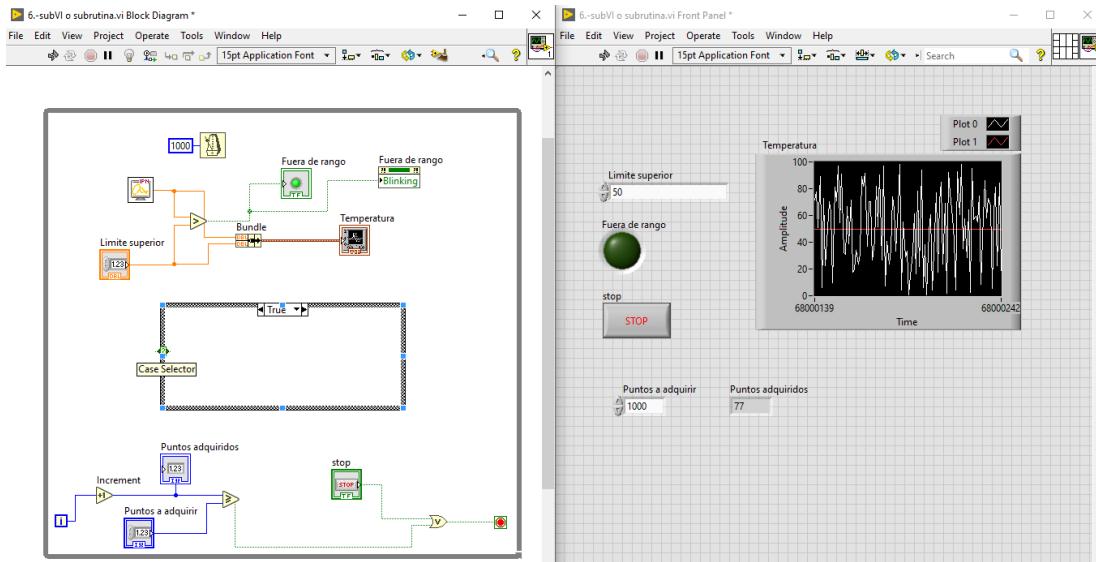
El condicional switch case lo que hace es ejecutarse solamente cuando se cumpla cierta condición específica de opciones enlistadas, esta condición en la Case Structure se analiza a través de un resultado booleano de True o False y al cumplirse se analizan las condiciones que se tenga enlistadas en su parte superior, ejecutando los diferentes contenidos que se incluyan cada caja de su estructura, los cuales pueden ser totalmente distintos entre sí dependiendo del estado específico de la condición.



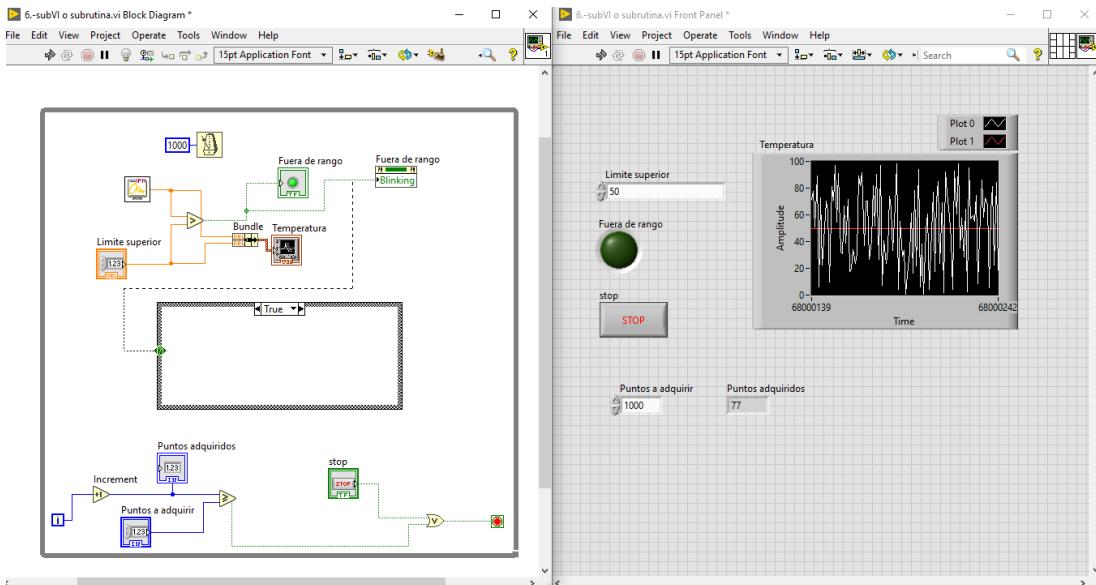
En este caso se agrega un Case Structure muy básico, donde solo se analiza si la condición es falsa o verdadera y dependiendo de ese resultado se ejecuta una cosa u otra completamente distinta.



En la parte superior del Case Structure se puede ver una parte que dice True y cuenta con dos flechas laterales y una de drop down menu, en esta parte se cuenta con todas las posibilidades del condicional switch case, al seleccionar cada una se encontrará una caja vacía donde puedo colocar con otros bloques de operación lo que se busca ejecutar si es que se cumple cada condición. La condición se conecta al símbolo de interrogación (?) verde que se encuentra al costado izquierdo de la estructura.

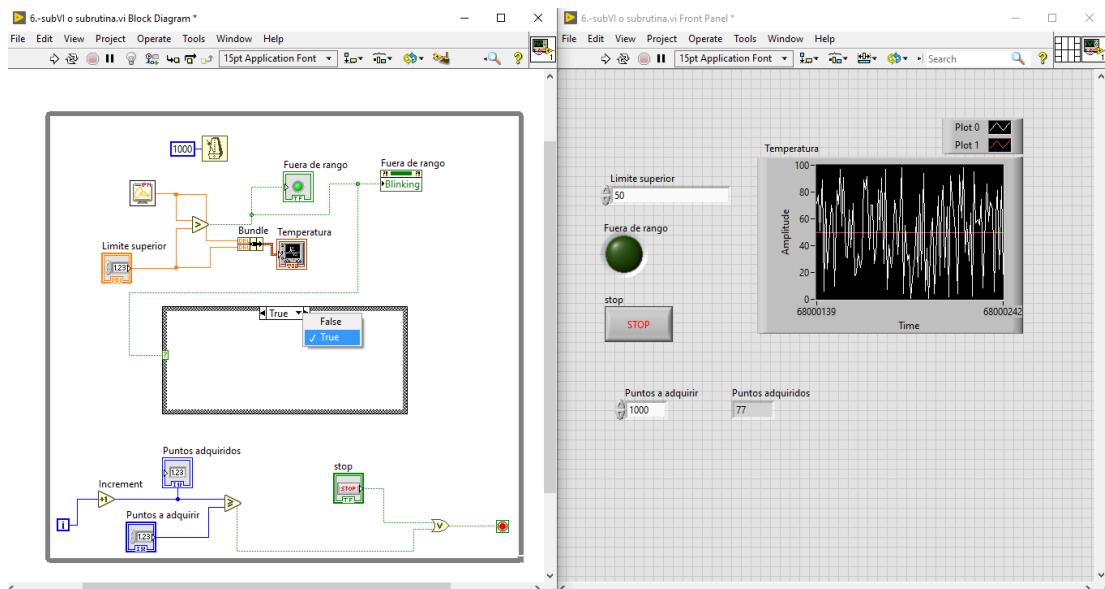


Cada vez que se excede un límite superior, se ejecutará alguna acción dada por el condicional.



Entonces con el condicional switch case (Case Structure) se analizarán dos posibles casos:

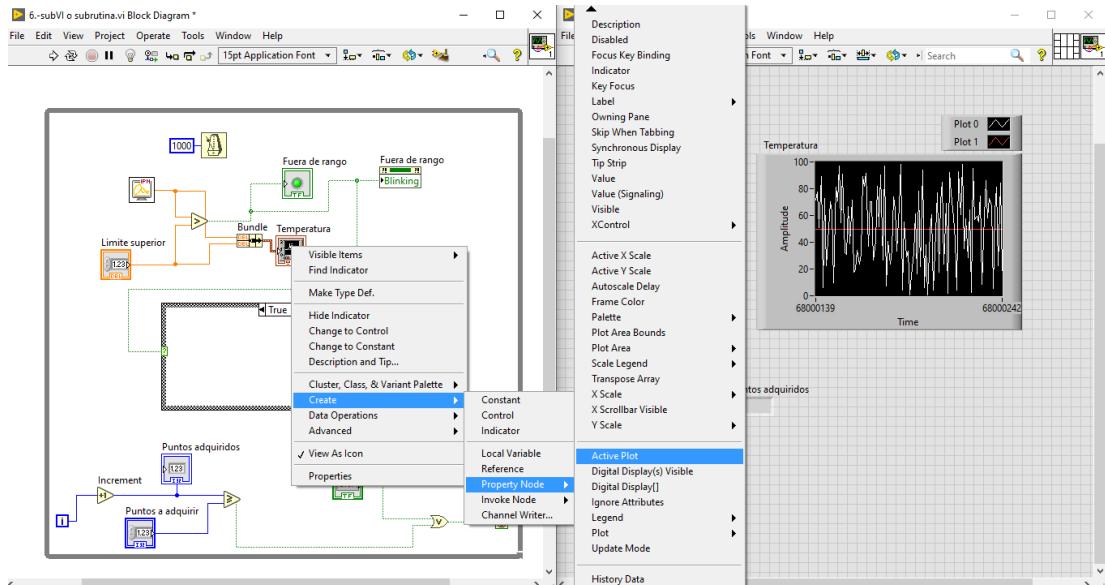
- La señal ha sobrepasado el límite superior (**El led Indicador parpadea, Condición = True**):
 - El color de la señal se **hace rojo** y además suena una alerta sonora.
- La señal NO ha sobrepasado el límite superior (**El led No parpadea, Condición = False**):
 - El color de la señal se queda blanco tal cual está.



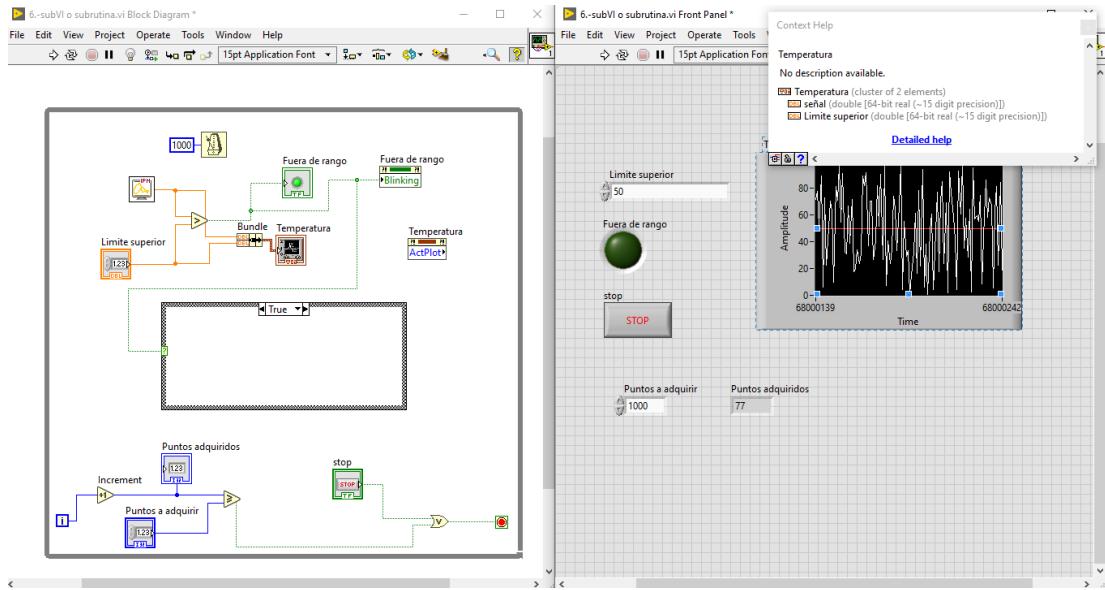
Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque

Ahora lo que voy a hacer es acceder a las propiedades del bloque de graficación Waveform Chart (Dynamic Data) dando clic derecho en el bloque de la siguiente manera y seleccionando la opción de Create → Property Node → Active Plot:

Recordemos que un Property Node es un elemento que accede a cualquier aspecto del bloque indicado, puede ser el historial, color, valor, si este parpadea de otro color o no, etc.

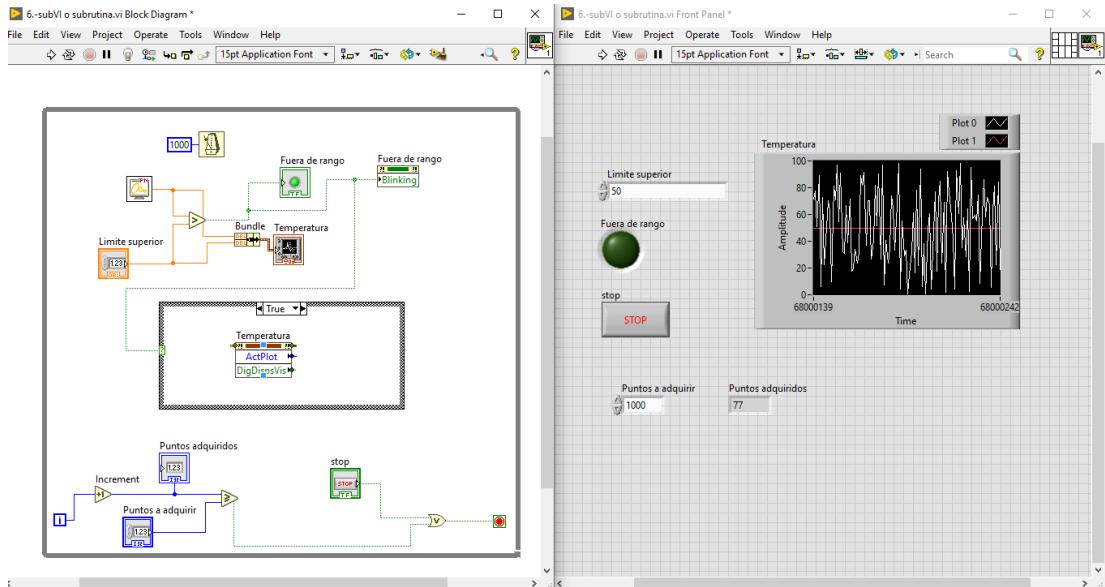


Esta propiedad de Active Plot lo que hace es seleccionar una función o gráfica a la cual queremos modificar una de sus gráficas internas, en este caso es la señal roja y blanca.



El orden en el que entran las gráficas al bloque de Bundle importa cuando tomemos su Property Node y como la señal aleatoria es la primera que entra esa es la 1, la señal constante del límite superior como entra después es la gráfica 2, eso lo puedo ver por el orden que dice en la ventana de Context Help.

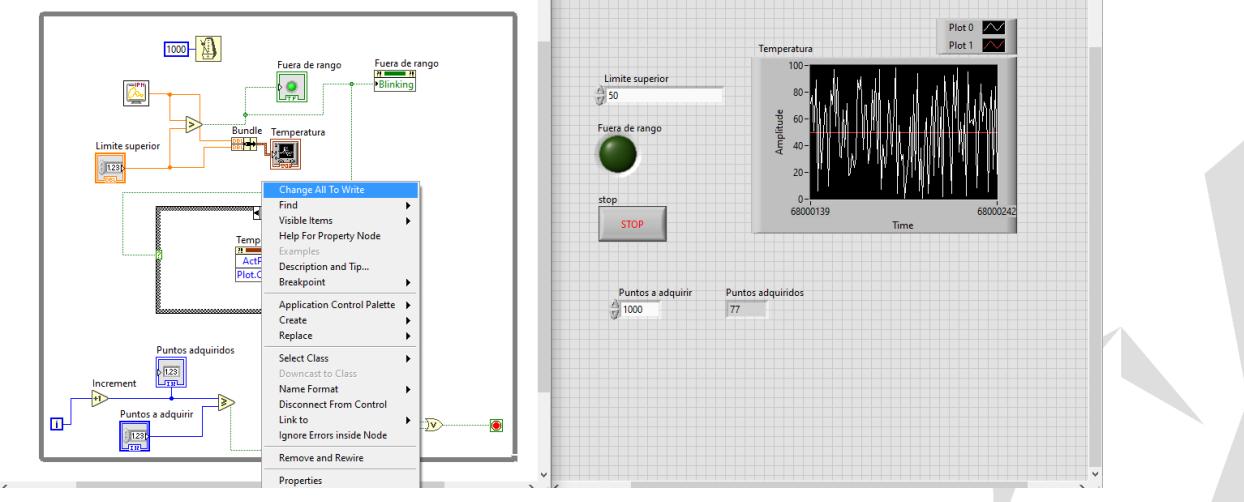
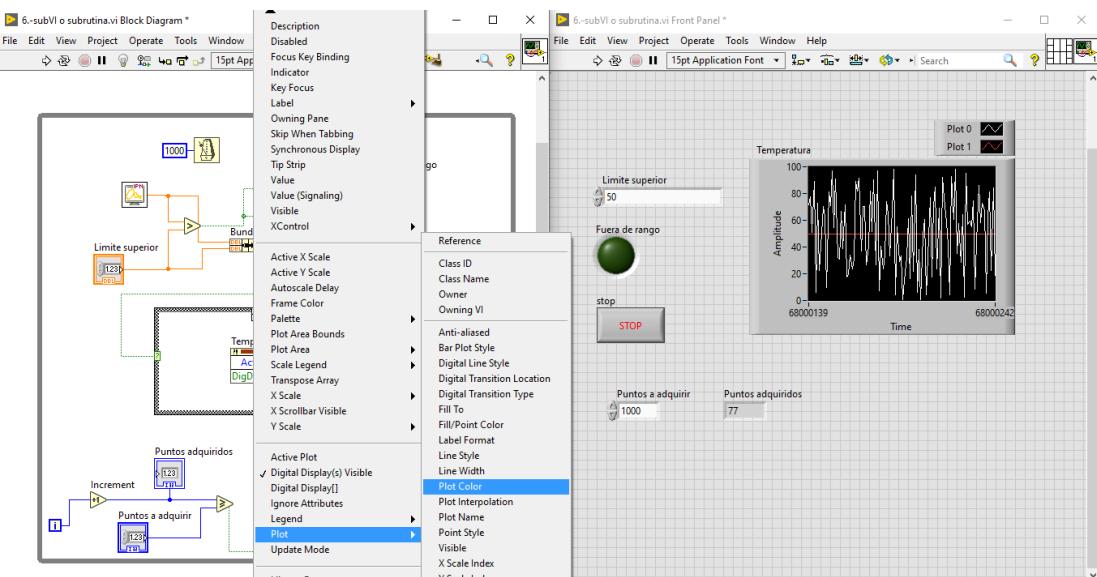
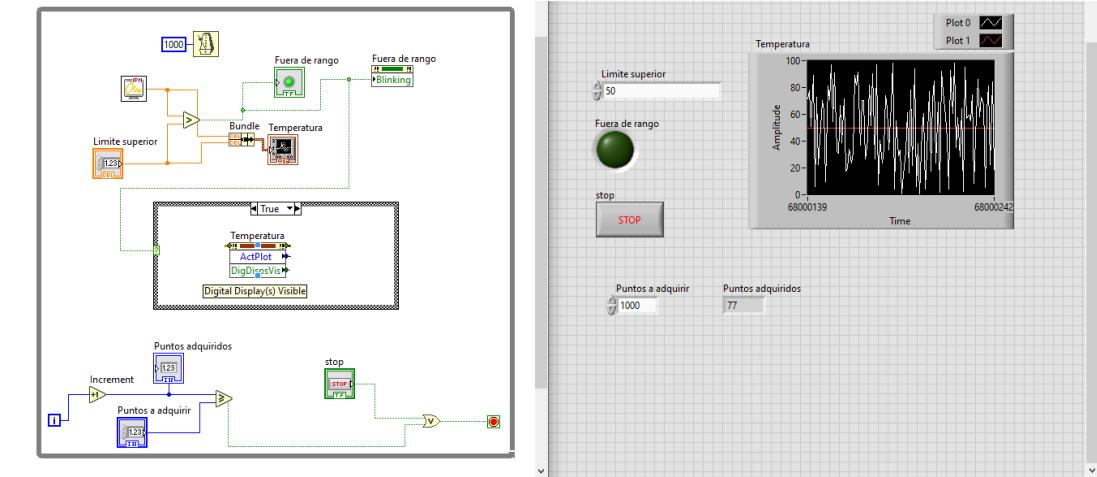
Posteriormente se introducirá el bloque del Property Node en el Case Structure y se extenderá hacia arriba para observar las propiedades que puedo acceder.



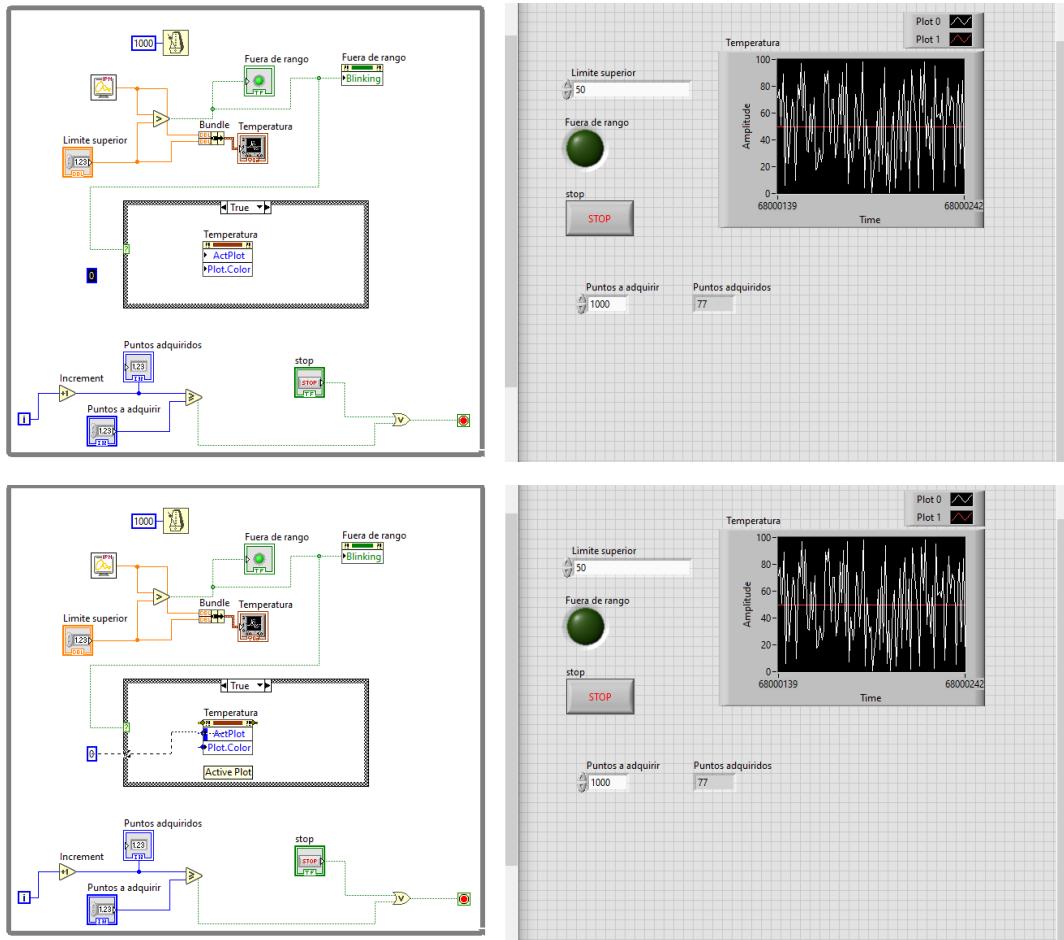
En el Property Node se dará clic derecho en alguna de las propiedades que aparecieron en el bloque y para poder cambiar el color de la gráfica se seleccionará la opción de:

Seleccionar Propiedad de Color: Clic Derecho a Fila en Property Node → Plot → Plot Color.

Cambiar a Escritura la Propiedad de Color: Clic Derecho en Property Node → Change All To Write.

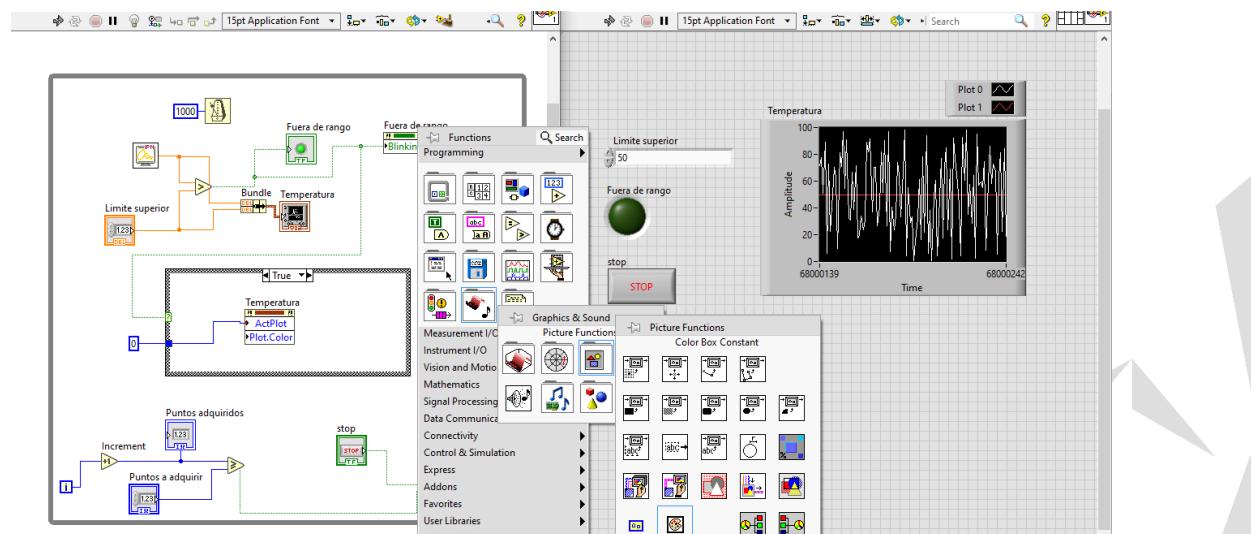


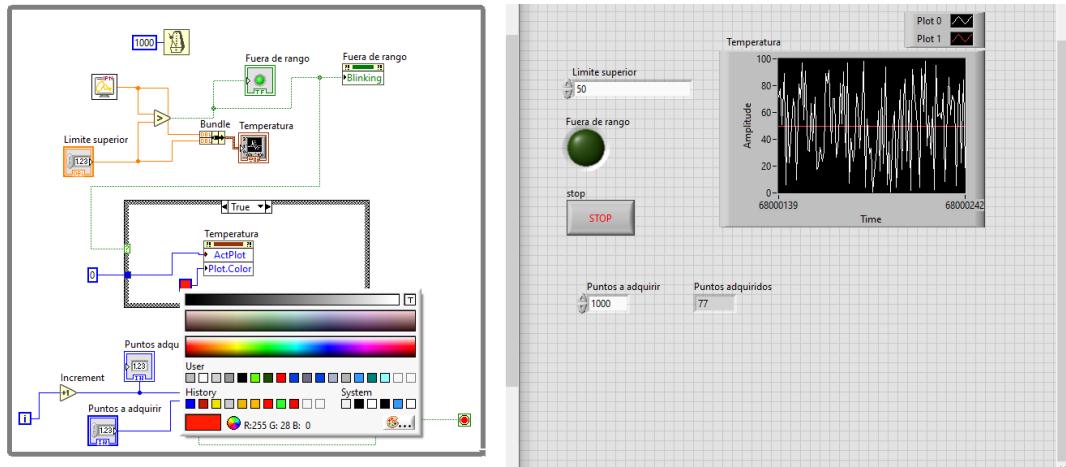
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Constante con valor cero para que los Property Nodes afecten a la gráfica blanca (la primera).



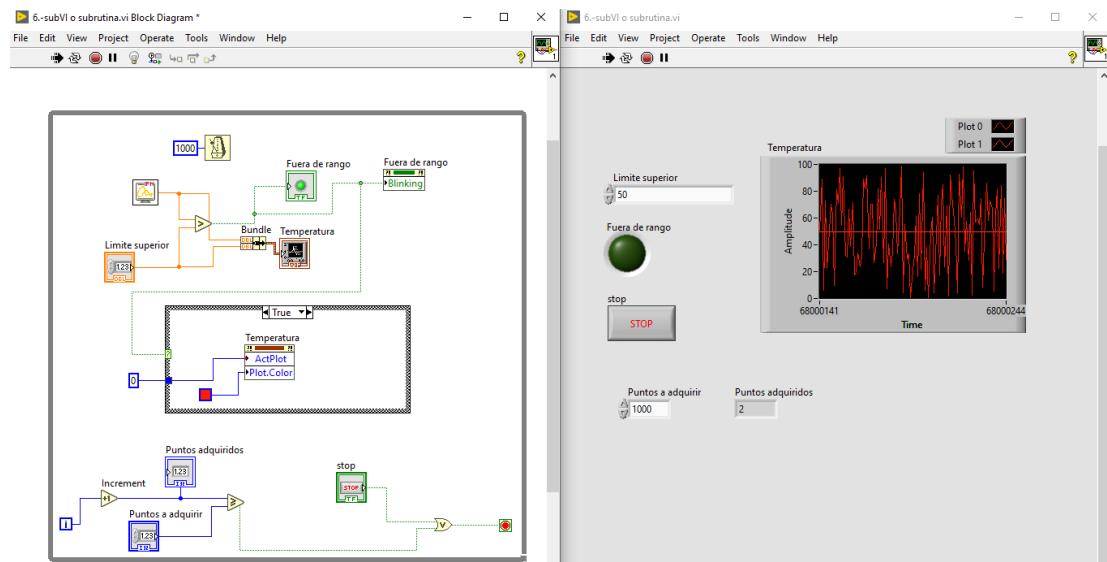
Block Diagram - Color Box Constant: Constante de Color

Esto lo usaré para denotar si el nivel de la gráfica está arriba o abajo del Límite Superior.



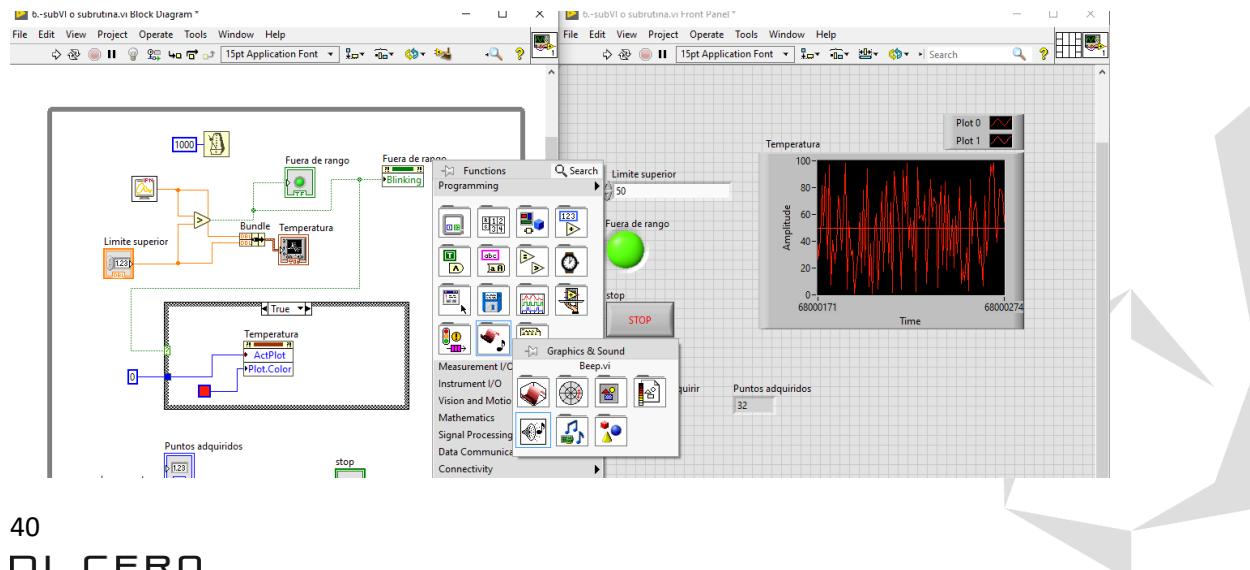


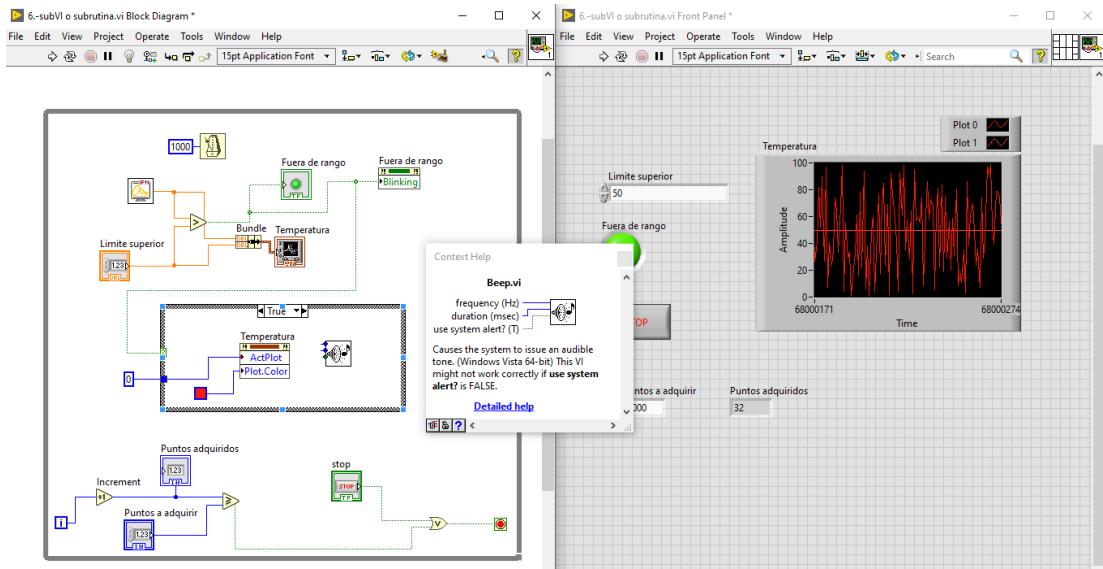
Ejecución del Programa: Señal con Led y Cambio de Color al Pasar el Límite Superior



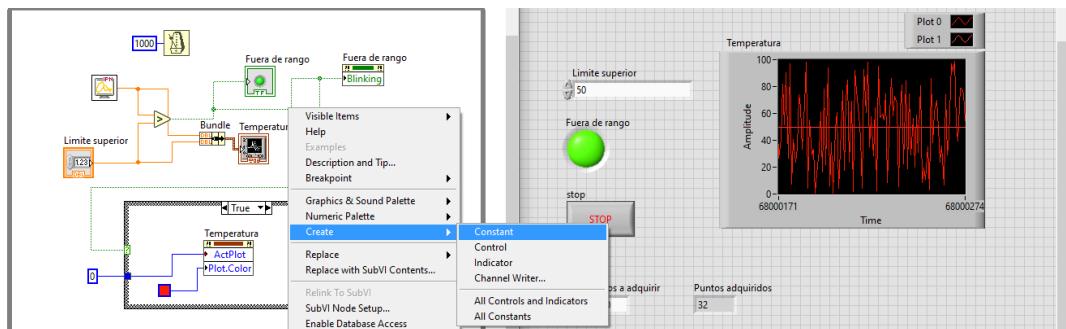
La gráfica actualmente se queda pintada de color rojo porque no tenemos la condición False declarada.

Block Diagram - Beep: Hacer Sonar un Sonido Durante Cierto Tiempo en Milisegundos



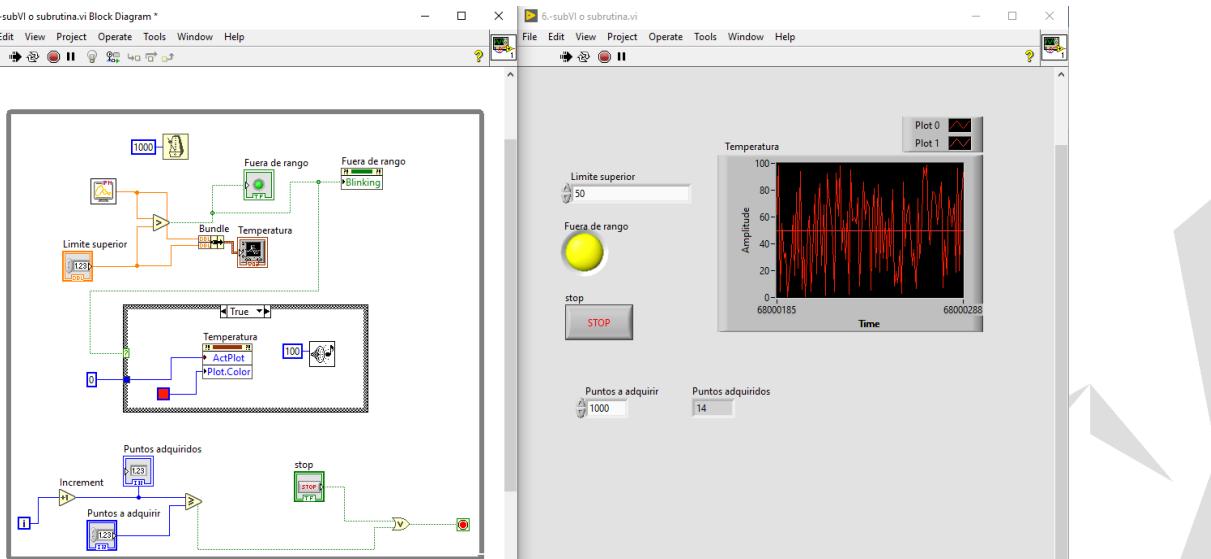


Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Tiempo de duración que suena el sonido, indicado en milisegundos.

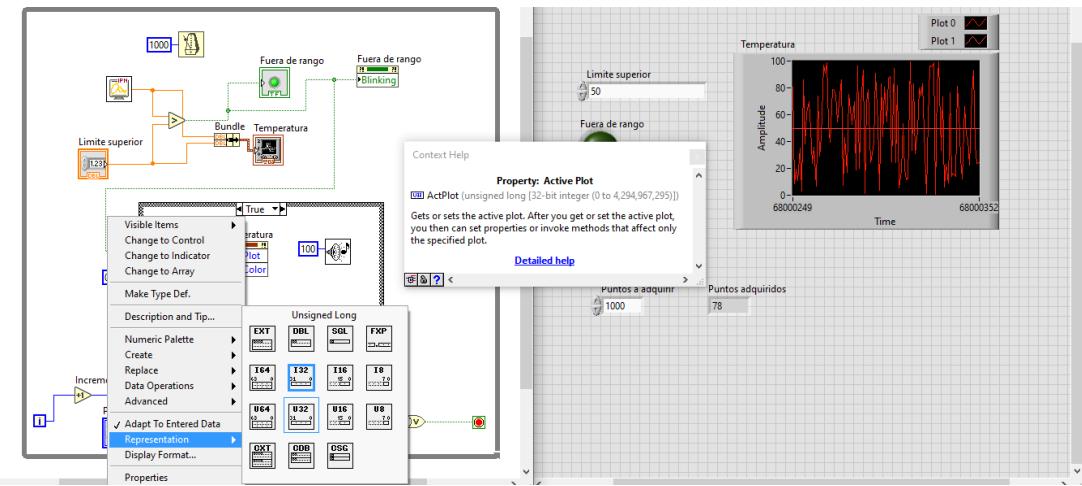


Ejecución del Programa: Señal con Led, Cambio de Color al Pasar el Límite Superior y Sonido

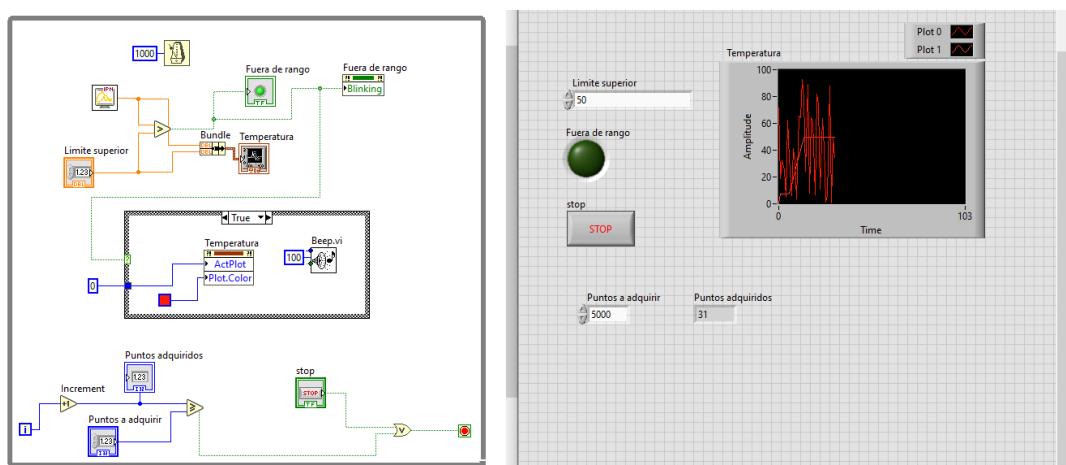
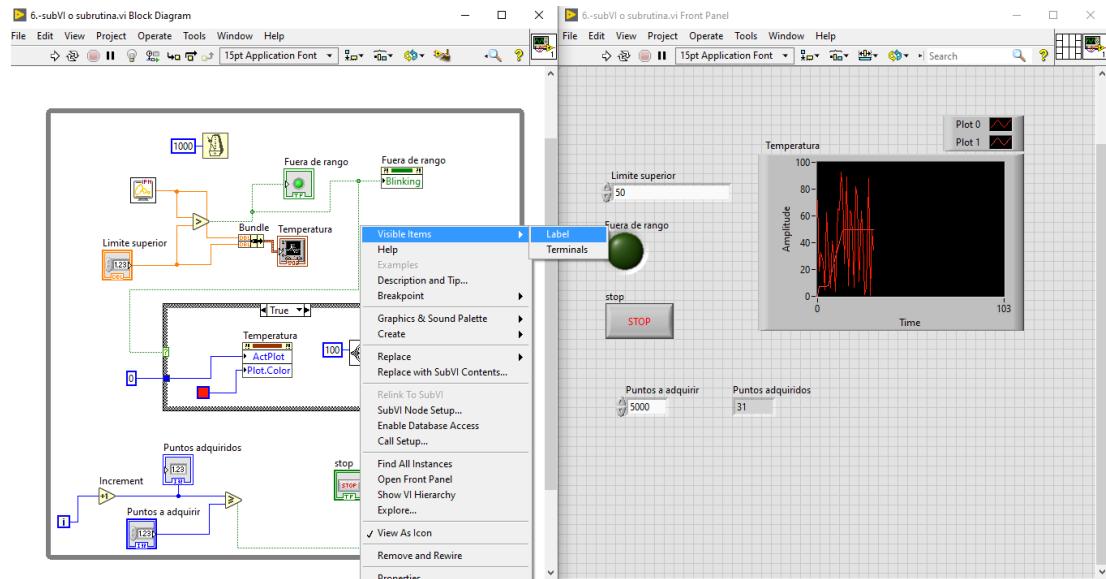
Cada que el led indicador se prenda y parpadee al pasar el límite superior, sonará un sonido también.



Cambiar el tipo de dato de mi elemento: Clic derecho en el bloque → Representation → Tipo de Dato.



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



Block Diagram - Case Structure: Condicional Switch Case

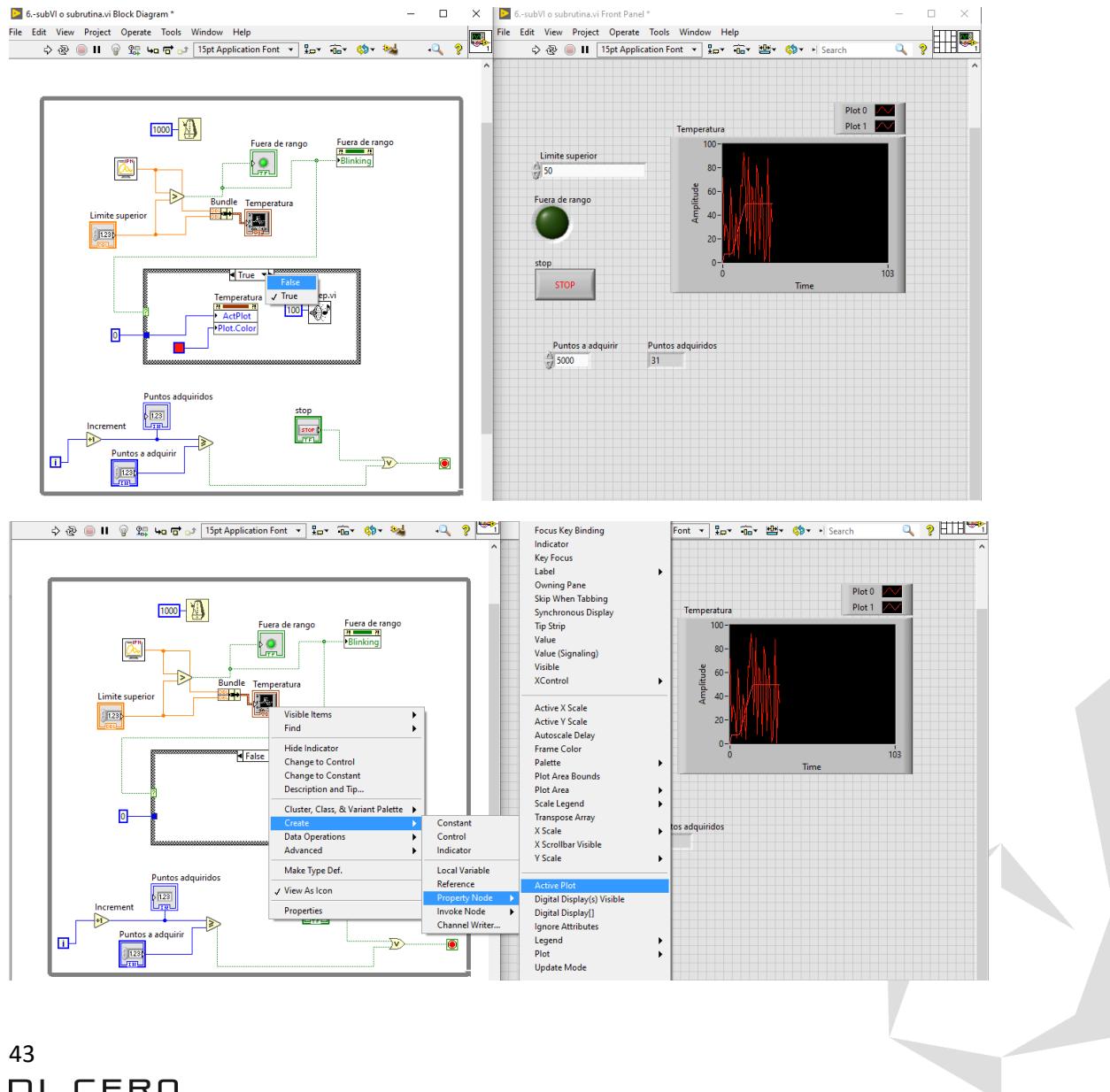
Entonces con el condicional switch case (Case Structure) se analizarán dos posibles casos:

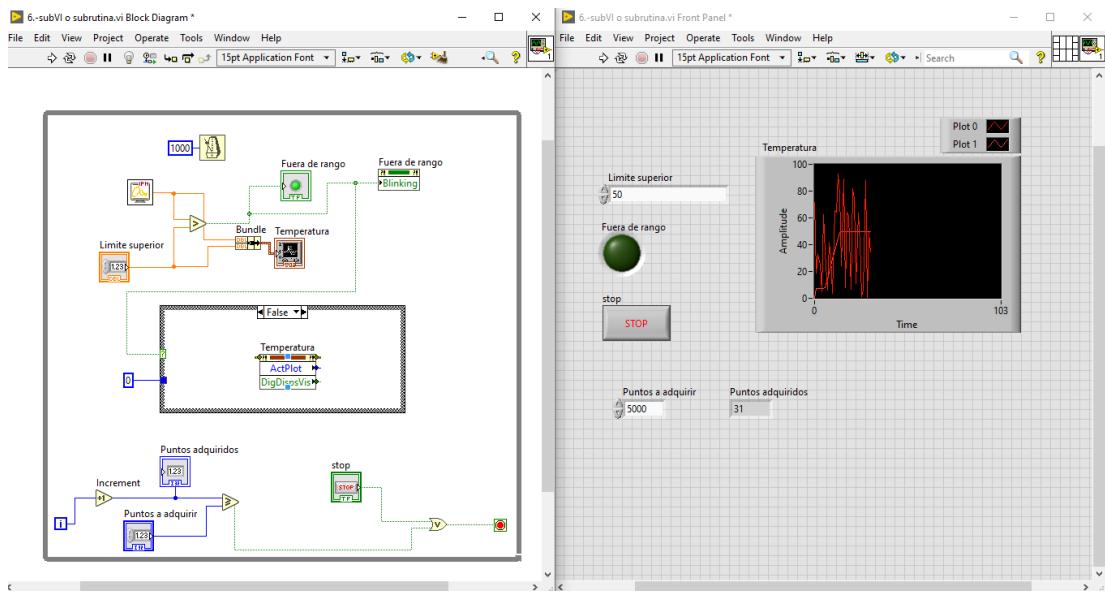
- La señal ha sobrepasado el límite superior (**El led Indicador parpadea, Condición = True**):
 - El color de la señal se **hace rojo** y además suena una alerta sonora.
- La señal NO ha sobrepasado el límite superior (**El led No parpadea, Condición = False**):
 - El color de la señal se queda blanco tal cual está.

Ya hemos declarado el caso de True, ahora vamos a poner el caso de false.

Ahora lo que voy a hacer es acceder a las propiedades del bloque de graficación Waveform Chart (Dynamic Data) dando clic derecho en el bloque de la siguiente manera y seleccionando la opción de Create → Property Node → Active Plot:

Recordemos que un Property Node es un elemento que accede a cualquier aspecto del bloque indicado, puede ser el historial, color, valor, si este parpadea de otro color o no, etc.

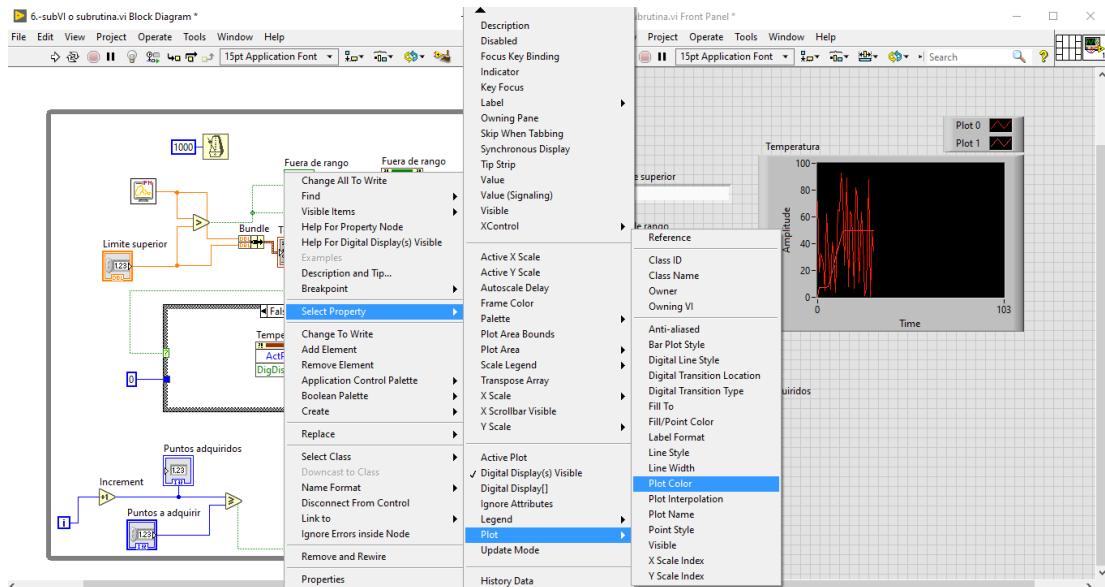




En el Property Node se dará clic derecho en alguna de las propiedades que aparecieron en el bloque y para poder cambiar el color de la gráfica se seleccionará la opción de:

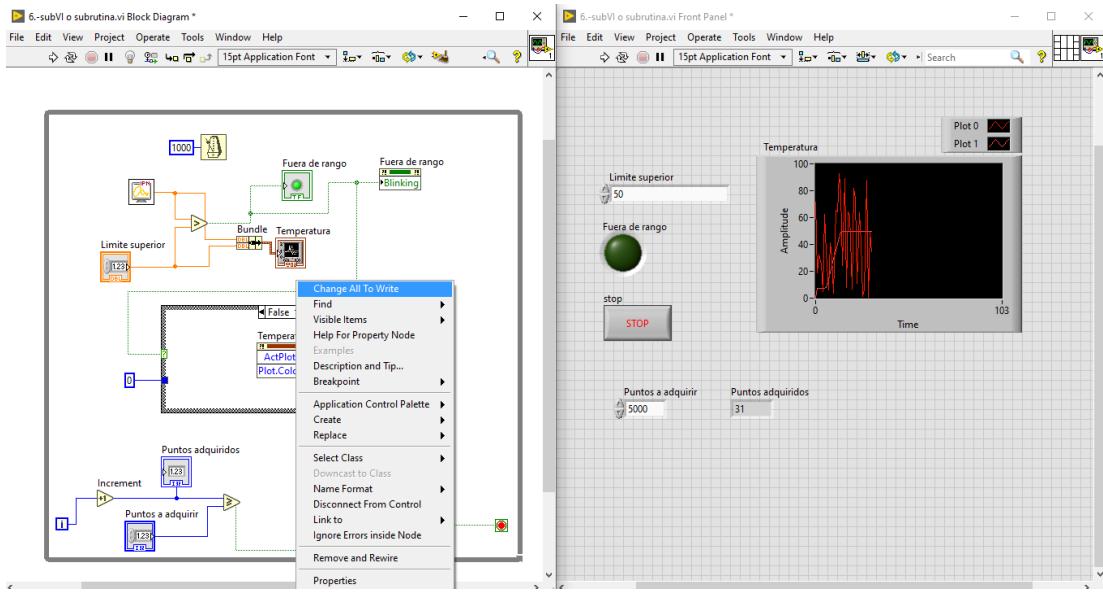
Seleccionar Propiedad de Color: Clic Derecho a Fila en Property Node → Plot → Plot Color.

Cambiar a Escritura la Propiedad de Color: Clic Derecho en Property Node → Change All To Write.



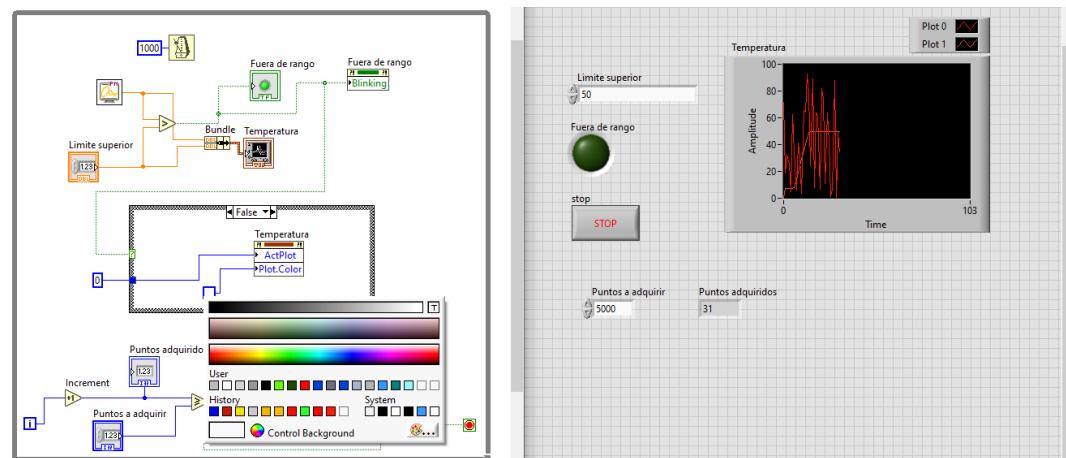
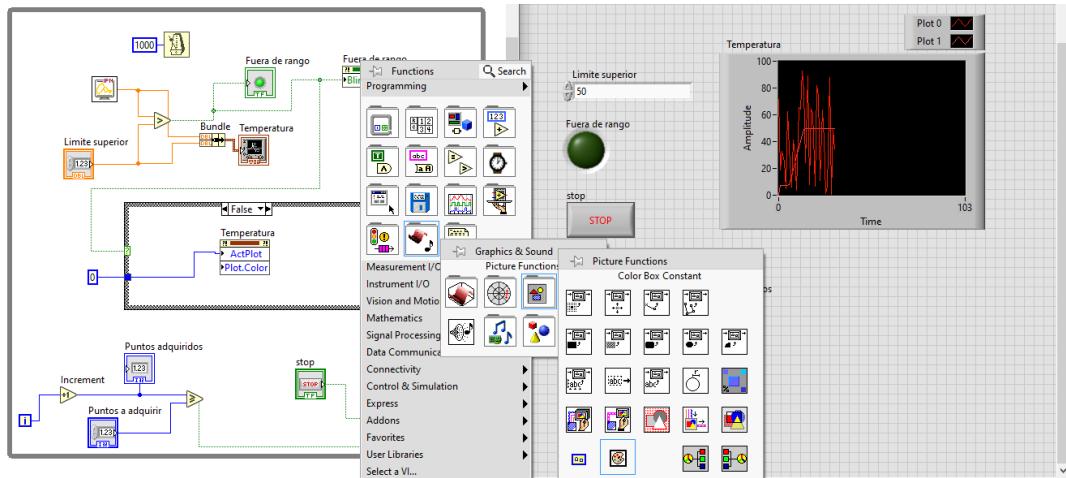
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Constante con valor cero para que los Property Nodes afecten a la gráfica blanca (la primera).

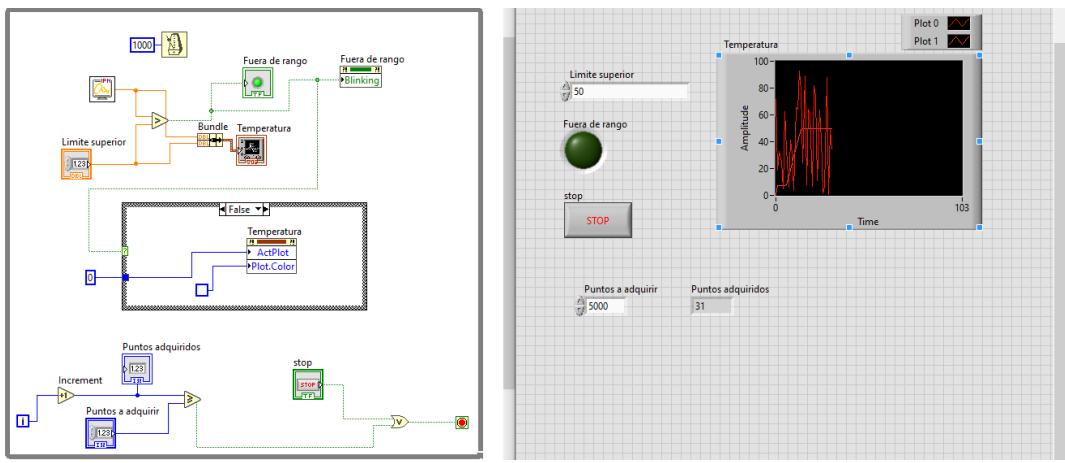
Aunque como esta constante ya había sido creada, solo reutilizo la que se había utilizado con la condición True, metiéndola al condicional switch case por medio de un túnel.



Block Diagram - Color Box Constant: Constante de Color

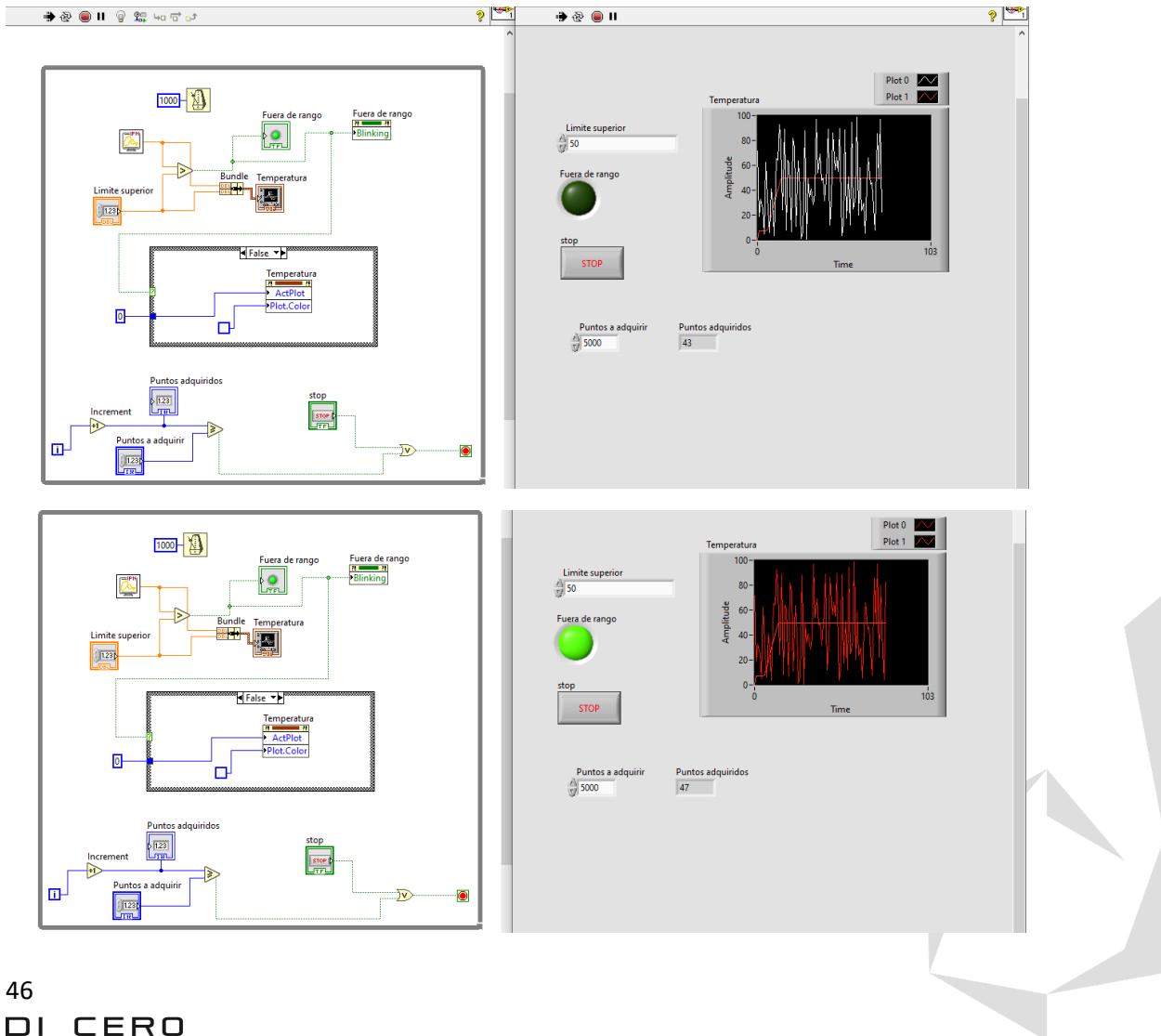
Esto lo usaré para denotar si el nivel de la gráfica está arriba o abajo del Límite Superior.



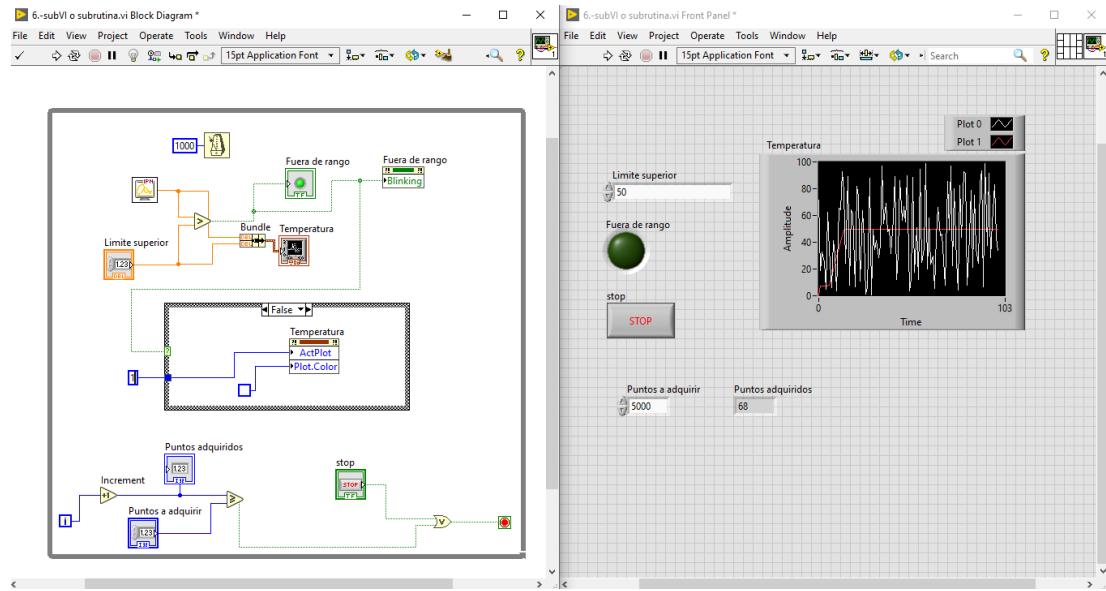


Ejecución del Programa: Señal con Led, Cambio de Color al Pasar el Límite Superior y Sonido

En esta ejecución del programa, cuando el nivel de la gráfica creada por el SubVI que genera números aleatorios de 0 a 100 se encuentre debajo del límite superior, la gráfica se pondrá de color blanco y cuando se encuentre arriba del límite se pondrá en color rojo, además de que sonará un beep y parpadeará un led cada vez que se pase del límite.

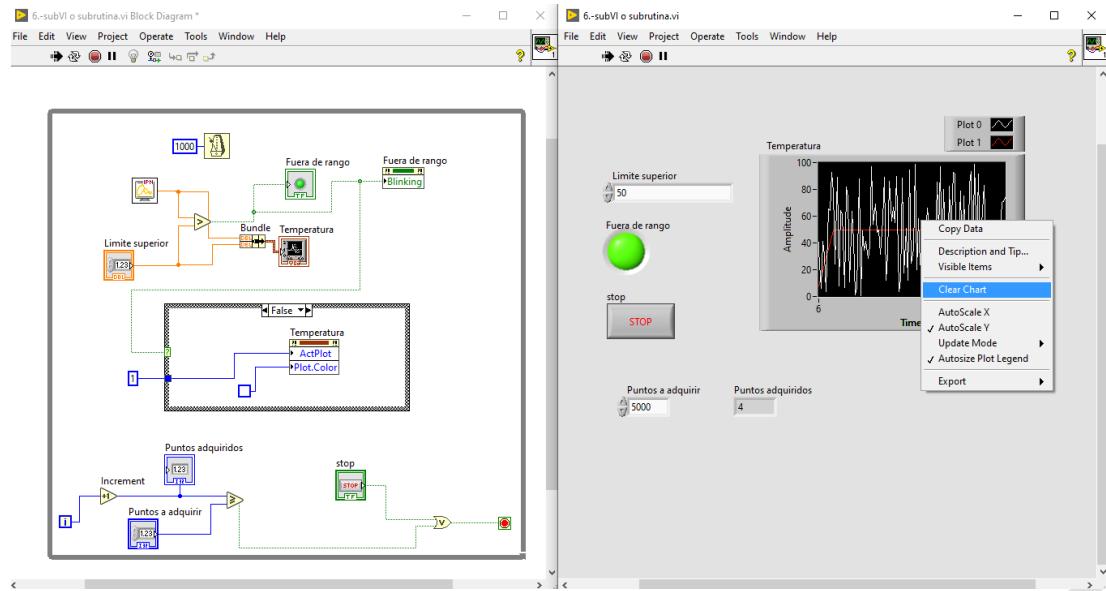


Si en la constante del Property Node que actualmente vale 0, colocamos un 1, se estará accediendo al Active Plot 1.

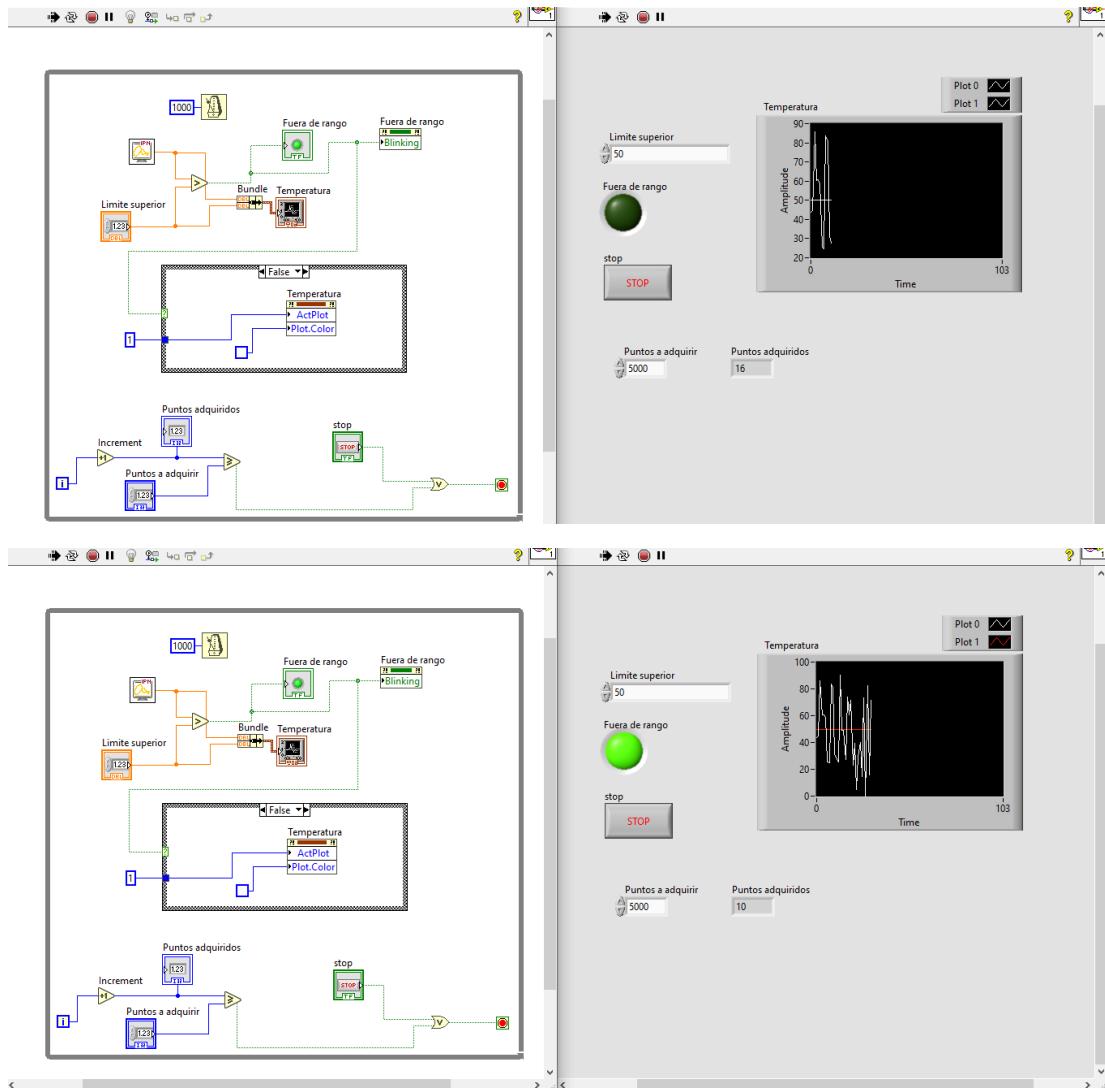


Ejecución del Programa: Señal con Led, Cambio de Color en el Límite Superior y Sonido

El Active Plot 1 (accedido con un 1 en la constante del Property Node que se encuentra dentro de la Case Structure) se refiere a la gráfica del límite superior, lo que ocurrirá si se ejecuta el programa con estas condiciones es que cuando la gráfica de números aleatorios (Active Plot 0) se encuentre por encima del límite superior, esta se pintará de blanco y cuando se encuentre debajo se pintará de rojo, la gráfica del límite superior es la línea constante horizontal.



Además de cambiar de color, cada que la gráfica cruce el nivel del límite superior, sonará una alerta y parpadeará el led.

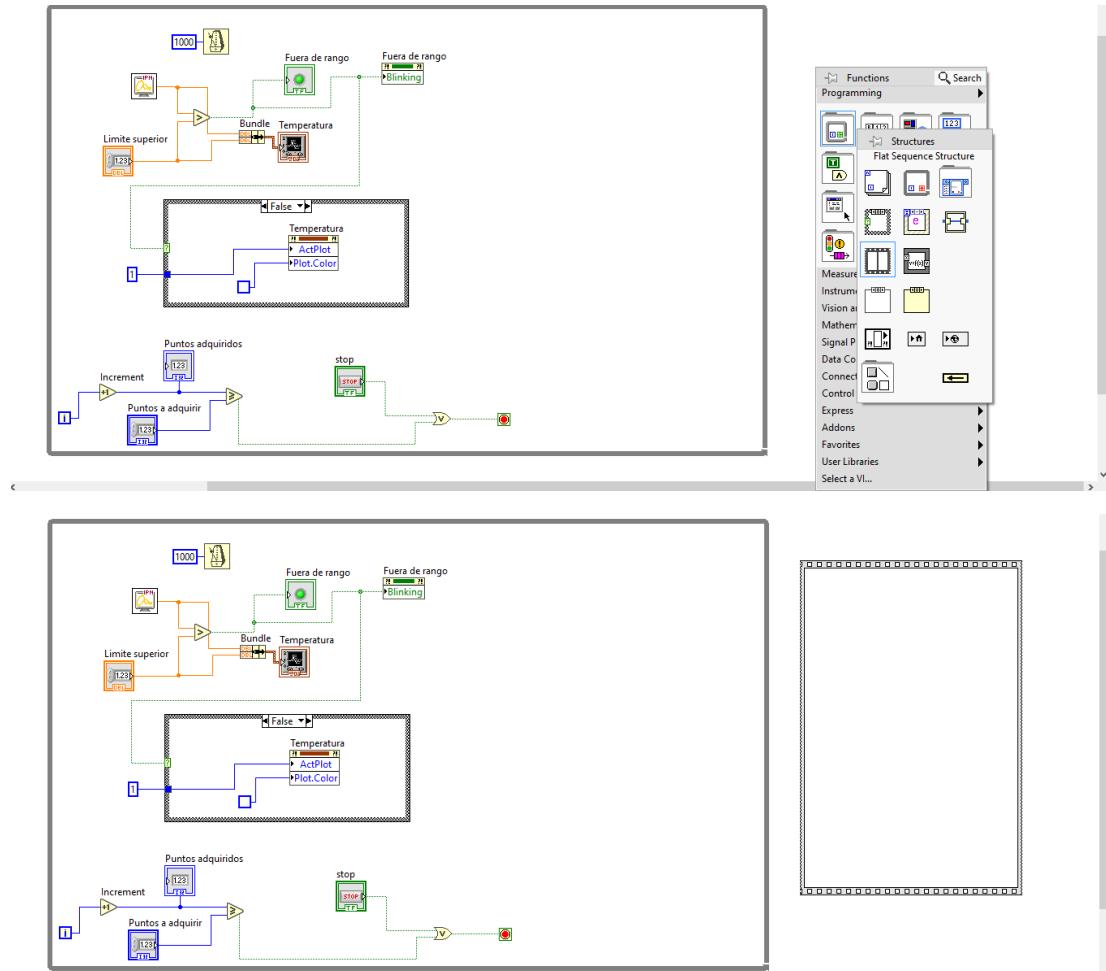


Los Waveform Chart son elementos gráficos que muestran señales reales o virtuales de tipo Dynamic Data (que son simplemente datos que cambian al transcurso del tiempo), cada vez que estos elementos se ejecutan, guardan en su interfaz la gráfica mostrada anteriormente, por lo cual se debe dar clic derecho sobre la interfaz y seleccionar la opción de borrar historial manualmente, por lo cual, si se quiere optimizar el uso de este elemento gráfico, se debe automatizar la limpieza de la gráfica, para que cada vez que el programa se prepare para ejecutar una nueva compilación, la interfaz gráfica del Waveform Chart se limpie por sí sola. Para ello se debe incluir un elemento que cree un orden en la ejecución del código de bloques en LabVIEW, ya que el limpiado del historial debe ocurrir antes que cualquier otra cosa.

Block Diagram - Flat Secuence Structure: Ejecución Ordenada Secuencial de Funciones

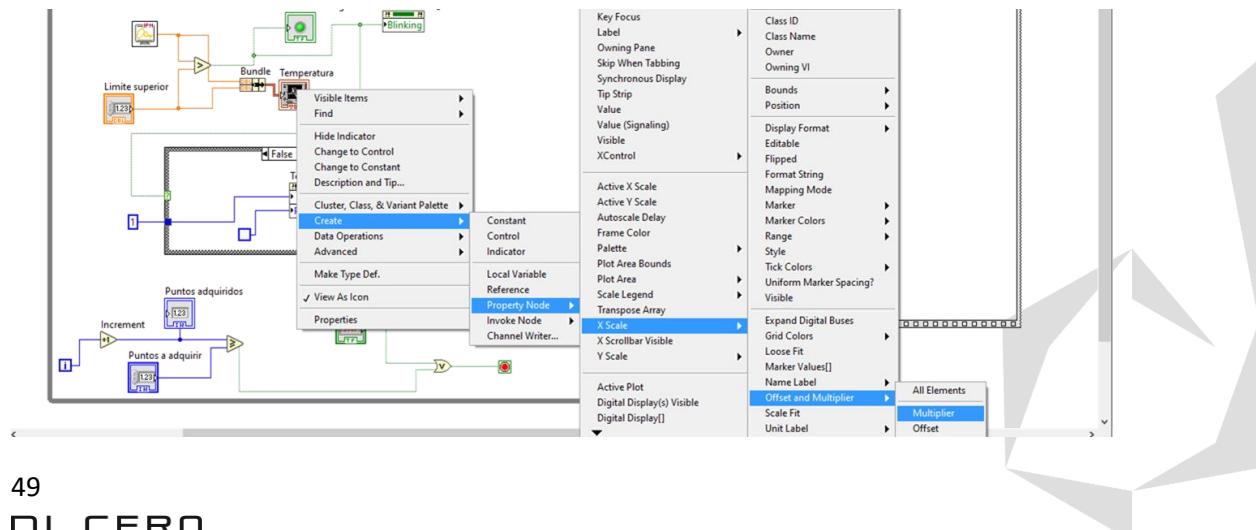
El bloque de Flat Secuence Structure sirve para que cada uno de sus cuadritos ejecute un proceso y hasta que cada uno de sus procesos se ejecuten, se ejecutará el siguiente, siguiendo una programación

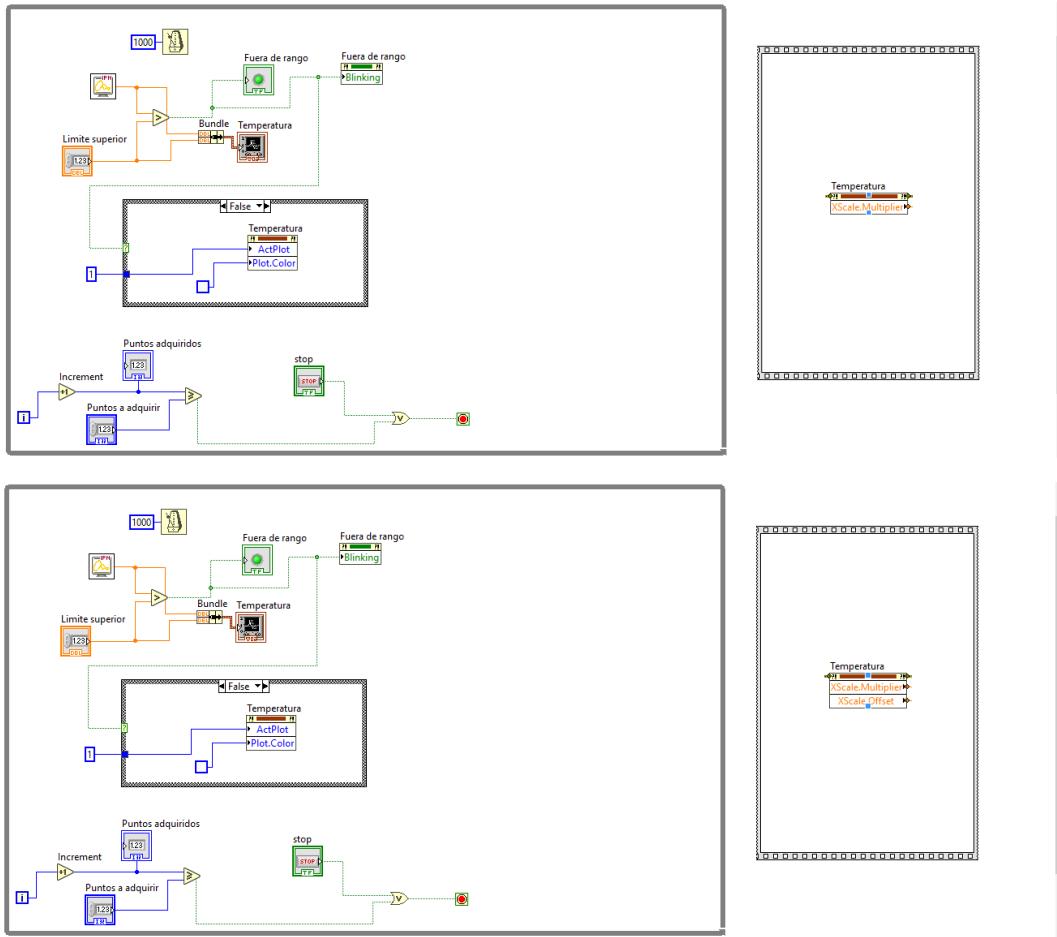
de escalera, esto significa que se ejecutará de forma secuencial, donde hasta que se termine de ejecutar una instrucción de código no se ejecutará la siguiente.



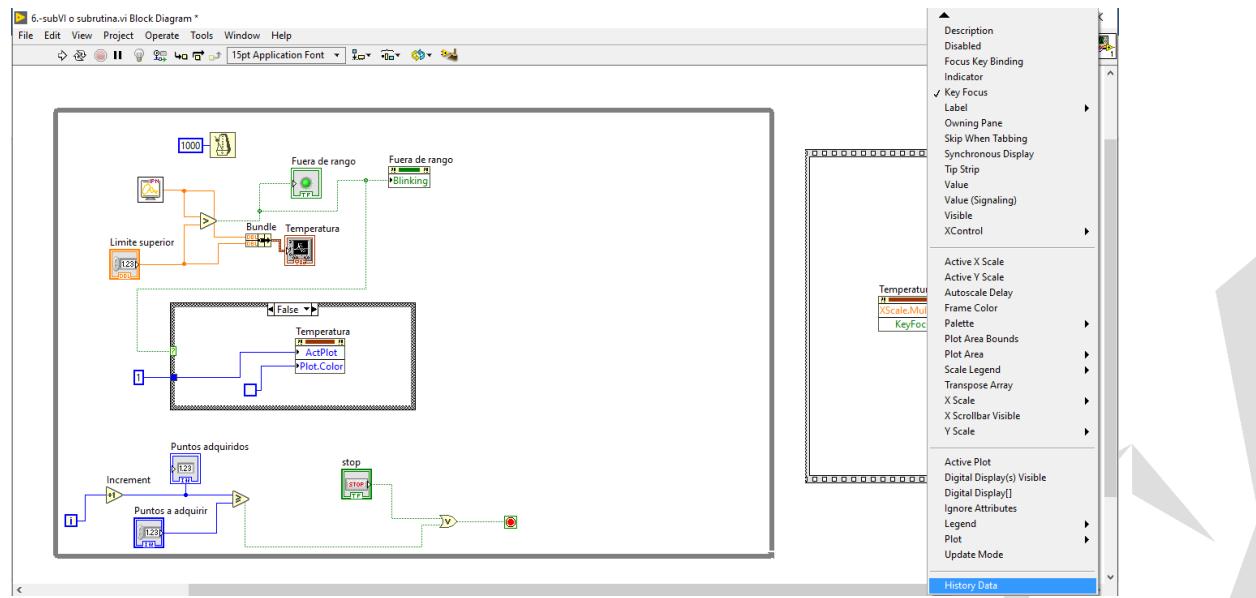
Block Diagram - Property Node: Propiedad Específica de Cualquier Bloque

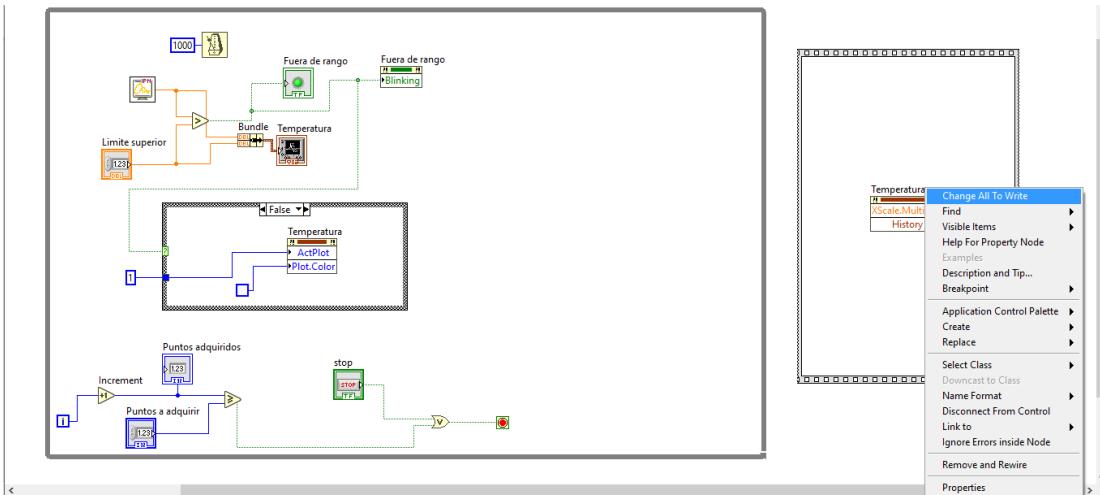
Ahora lo que voy a hacer es acceder a las propiedades de las gráficas dando clic derecho en el bloque de la siguiente manera y seleccionando la opción de Create → Property Node → History Data (Historial) y Create → Property Node → XScale → Multiplier (Paso Entre los Puntos de la Gráfica):



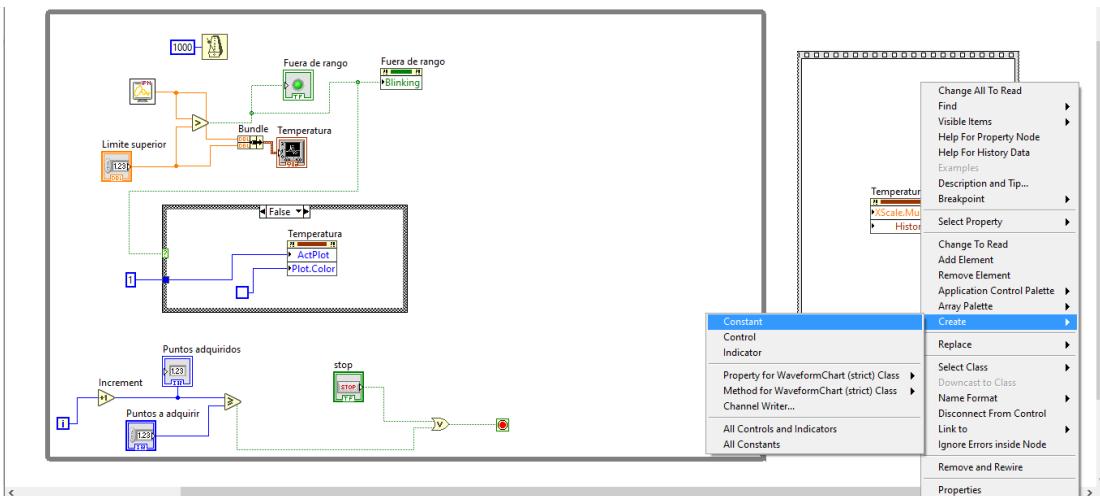


Este nodo (Property Node) de History Data incluye los datos almacenados dentro del Waveform Chart (**Dynamic Data**), pero yo no lo quiero leer, lo quiero escribir o editar, por lo que voy a darle clic derecho y seleccionar la opción de Change All To Write:

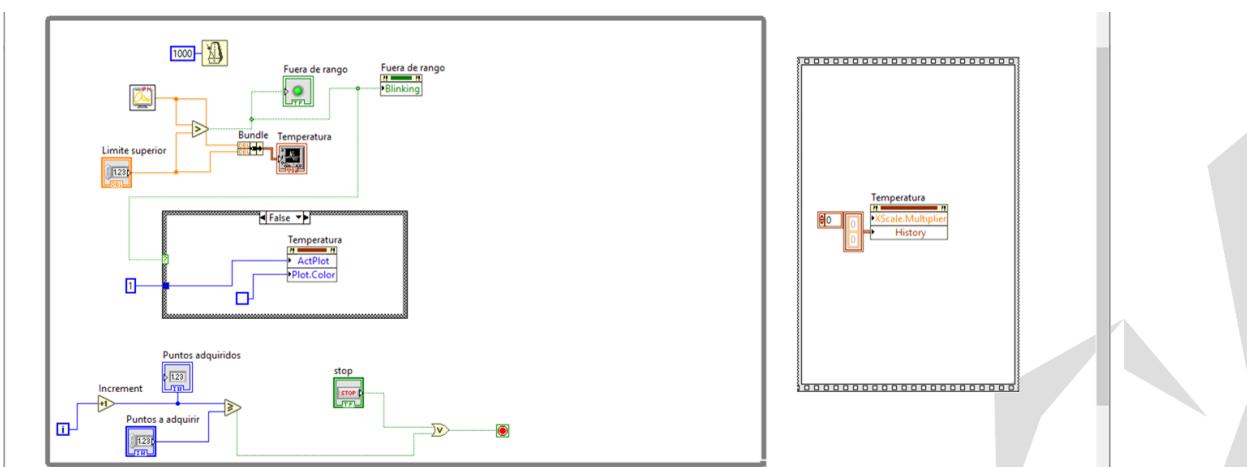




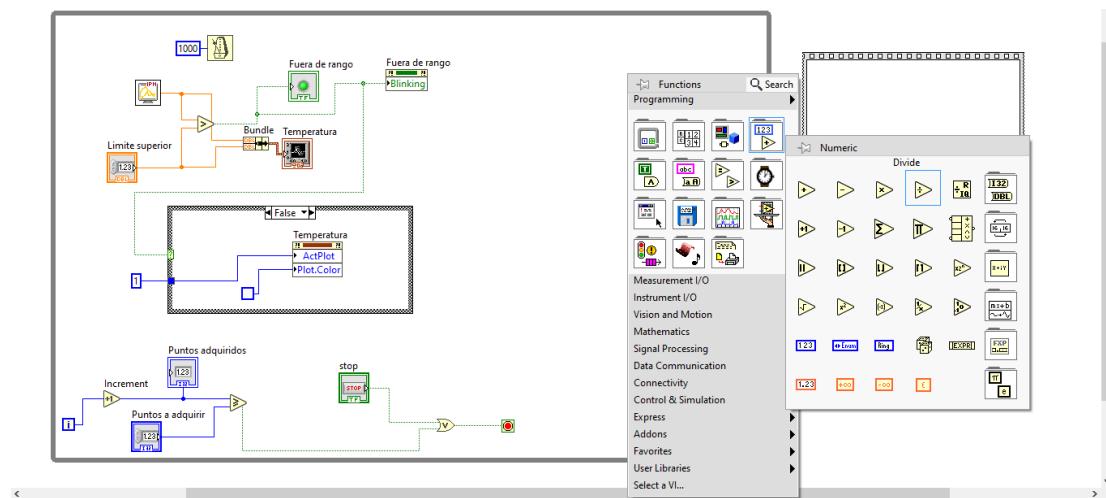
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Constante para indicarle al Waveform Chart que borre sus datos almacenados al ejecutarse.



Con la constante 0 se va a regresar las 2 gráficas a cero, borrando su historial, no debo conectarlo a nada porque es un bloque Property Node del mismo Waveform Chart (**Dynamic Data**).

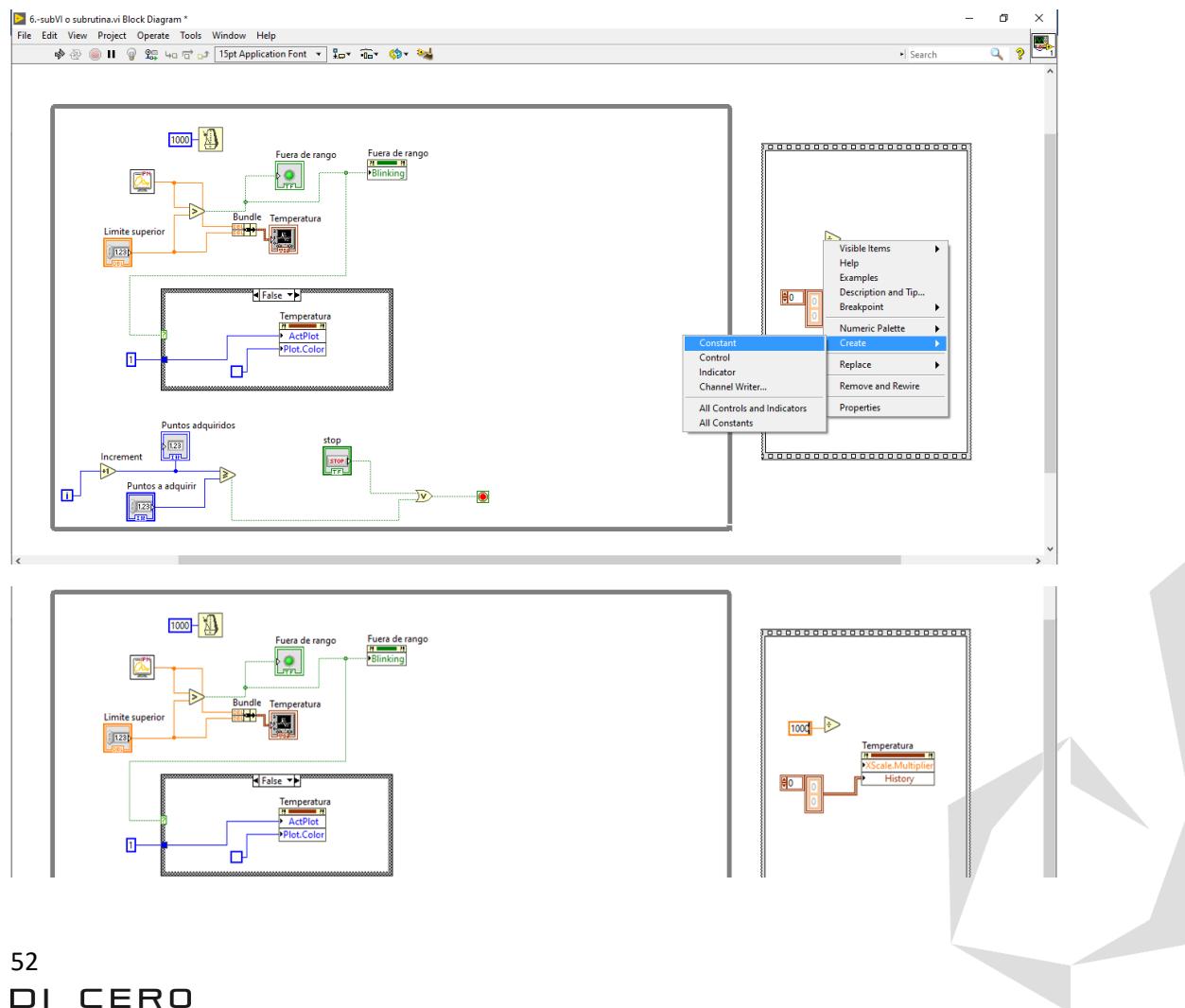


Block Diagram - Divide: Dividir dos Variables Numéricicas Entre Sí

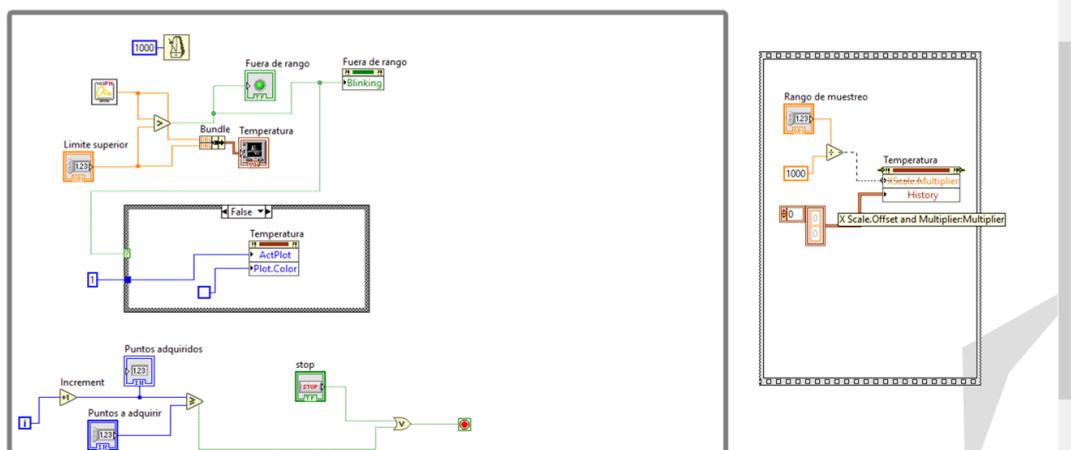
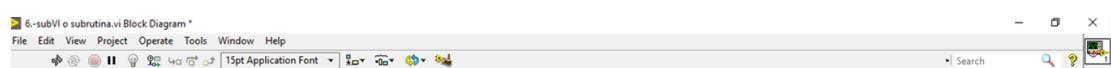
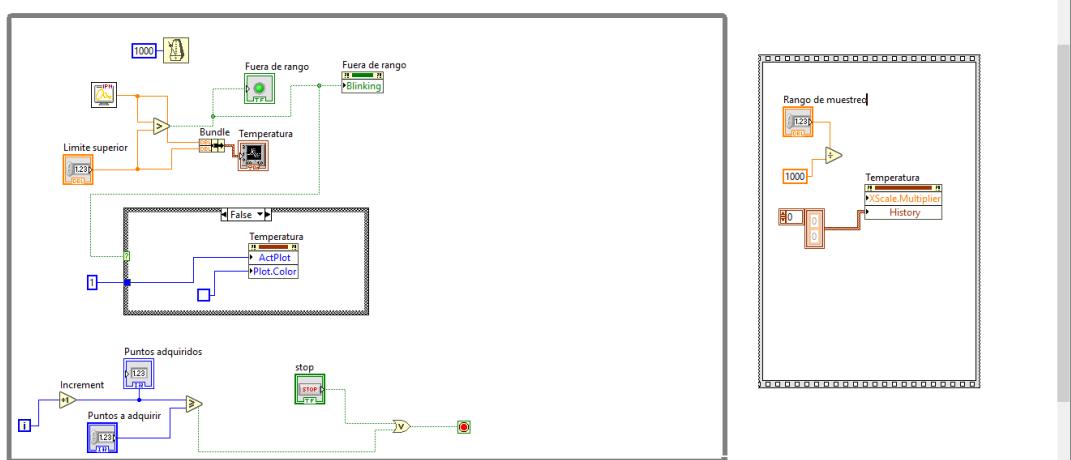
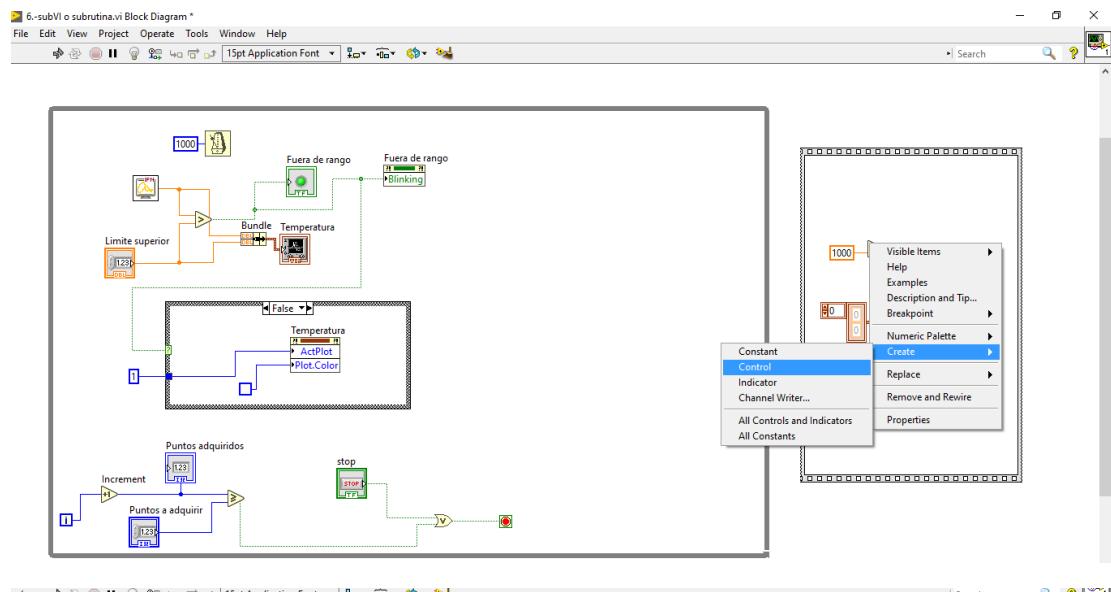


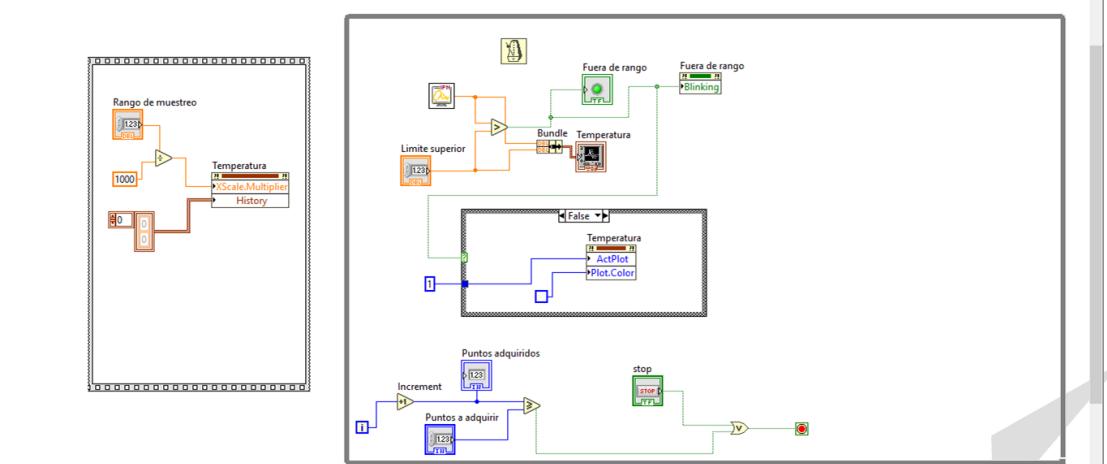
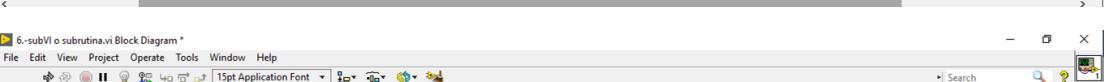
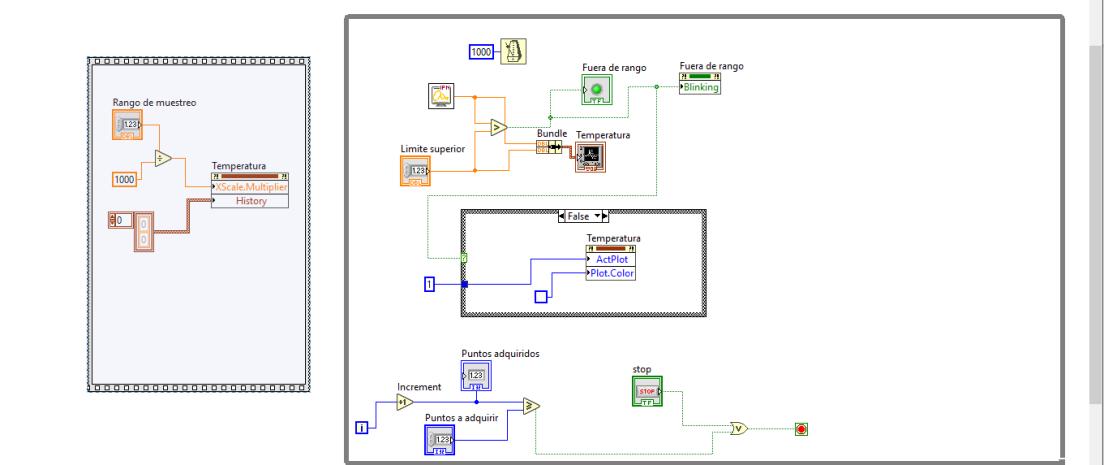
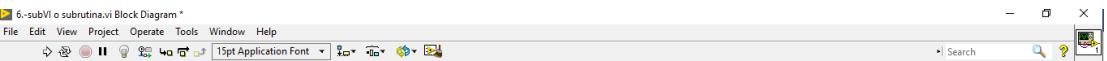
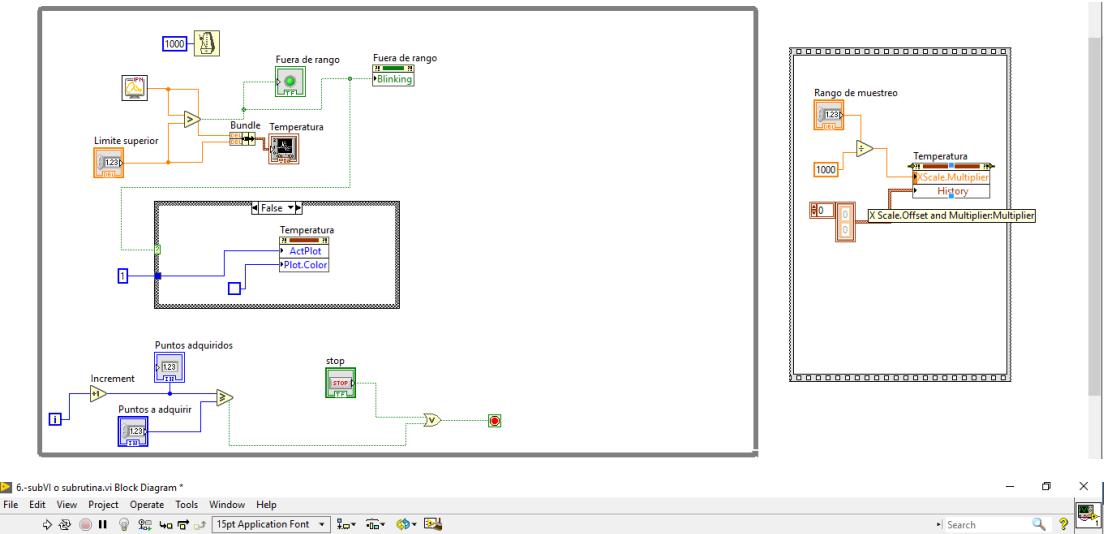
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Constante para declarar el paso (dx) entre los Puntos Adquiridos en la gráfica del Waveform Chart, osea el número decimal que separa en el eje horizontal los puntos de la gráfica.

Para denotar esto se utiliza el Property Node llamado XScale Multiplier.

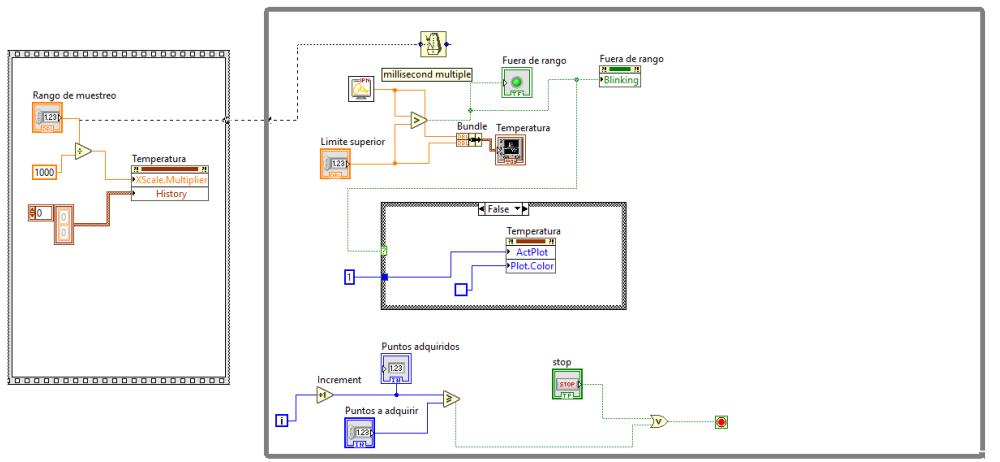


Crear un Control para un Bloque: Clic derecho en el bloque → Create Control.

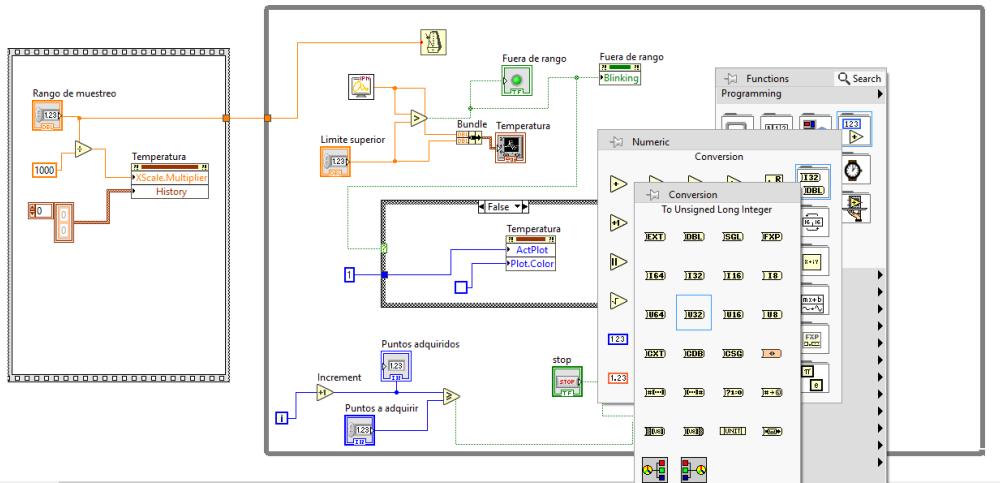




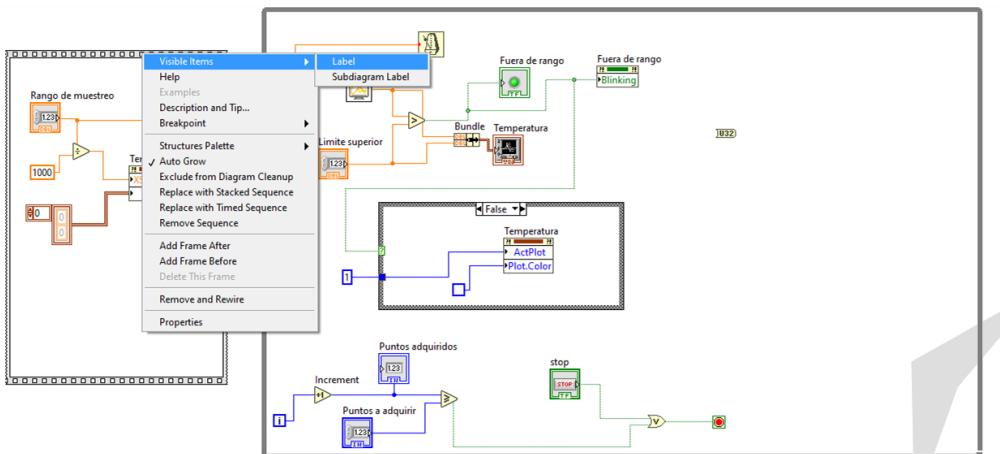
El rango de muestreo se conecta directamente al bloque de Wait Miliseconds que crea el temporizador para que se espere el tiempo indicado en el paso de la gráfica, pero para quitar el coercion dot que se crea se debe hacer una transformación de número decimal a entero.



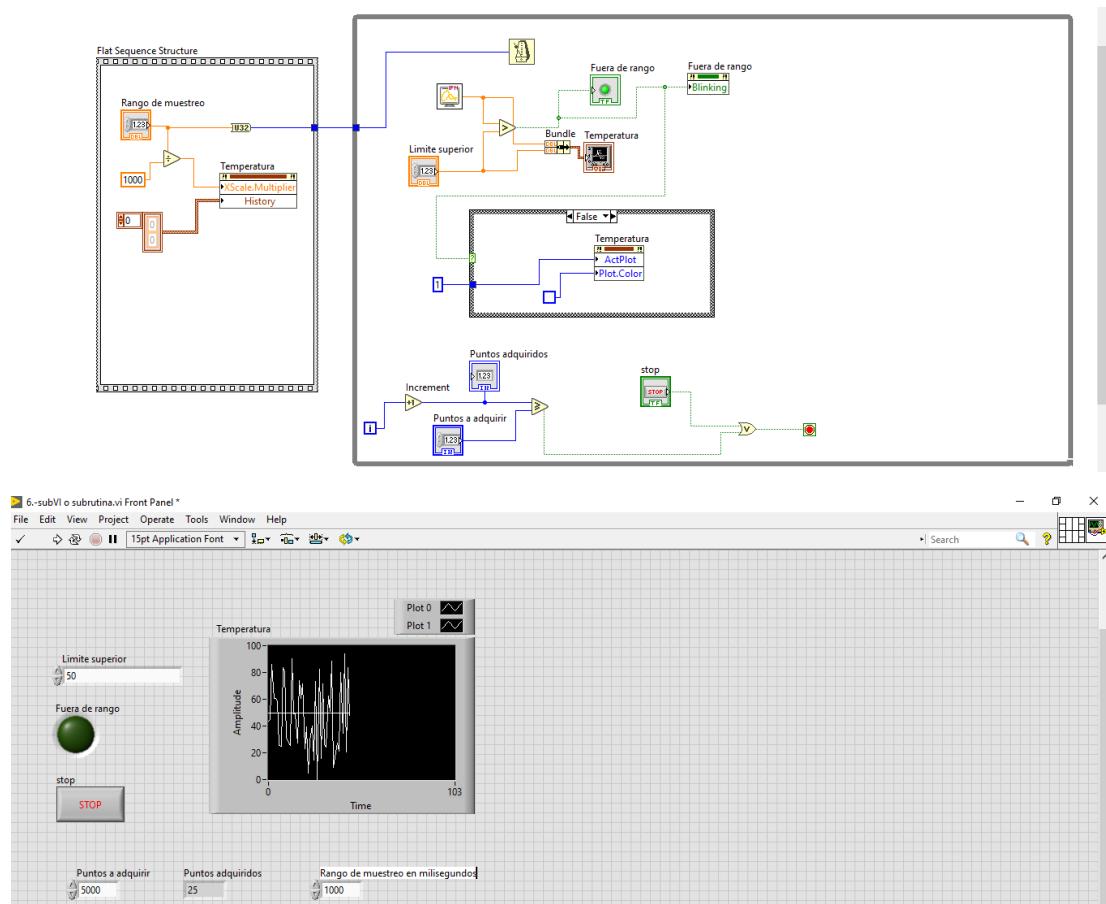
Block Diagram - To Unsigned Long Integer: Conversión Número Decimal a Entero



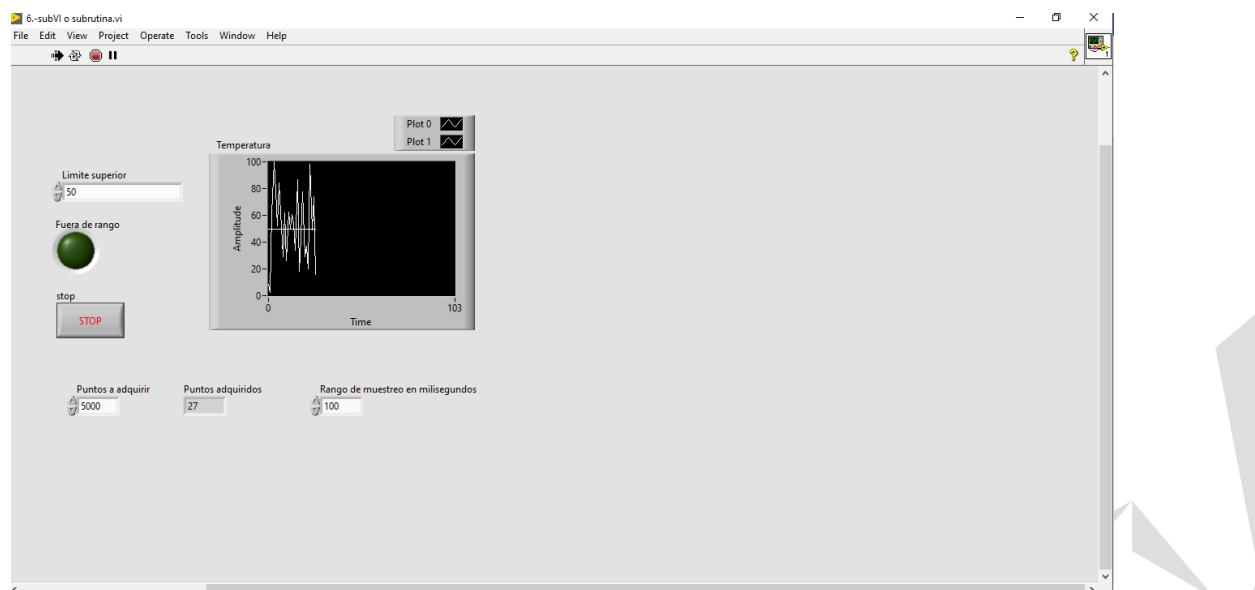
Se debe poner en el Flat Sequence, fuera del ciclo While sino se va a hacer la conversión varias veces.



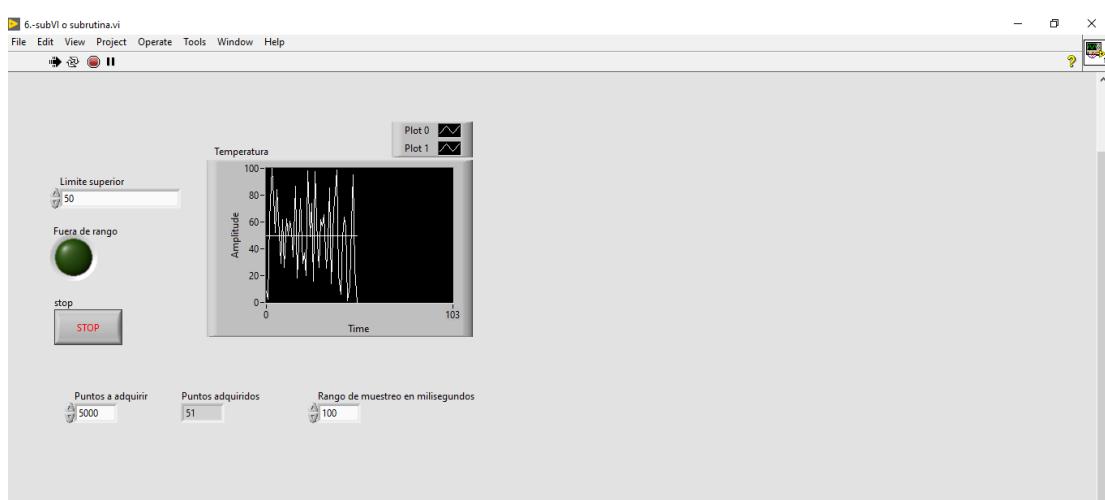
Ya se quitó el coercion dot porque se hizo correctamente la transformación de datos.



Ejecución del Programa: Señal con Led, Color en Límite Superior, Sonido y Control de dx y Puntos

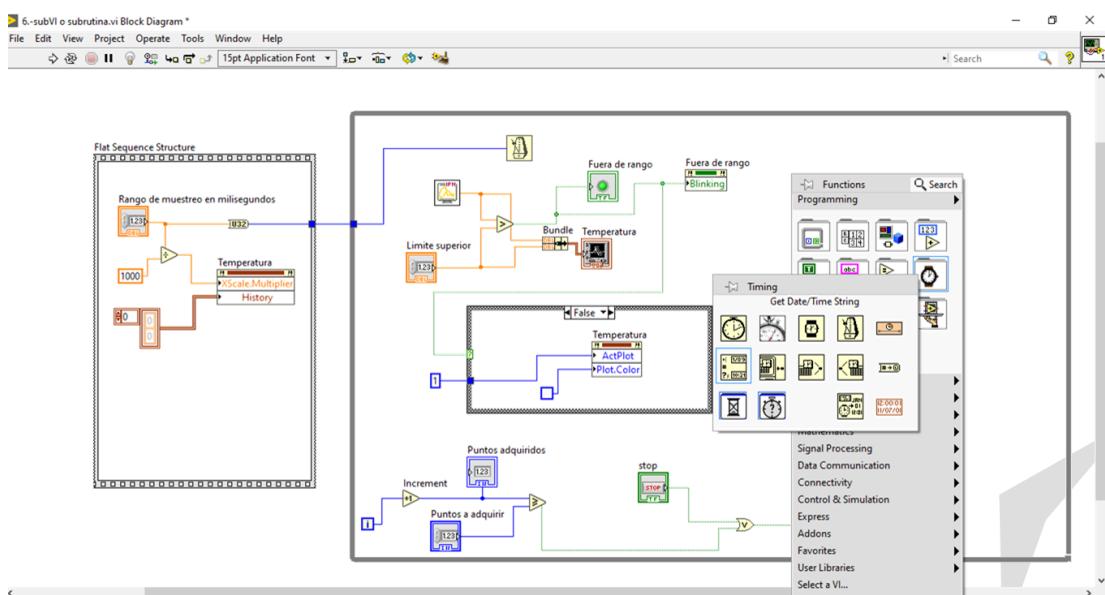


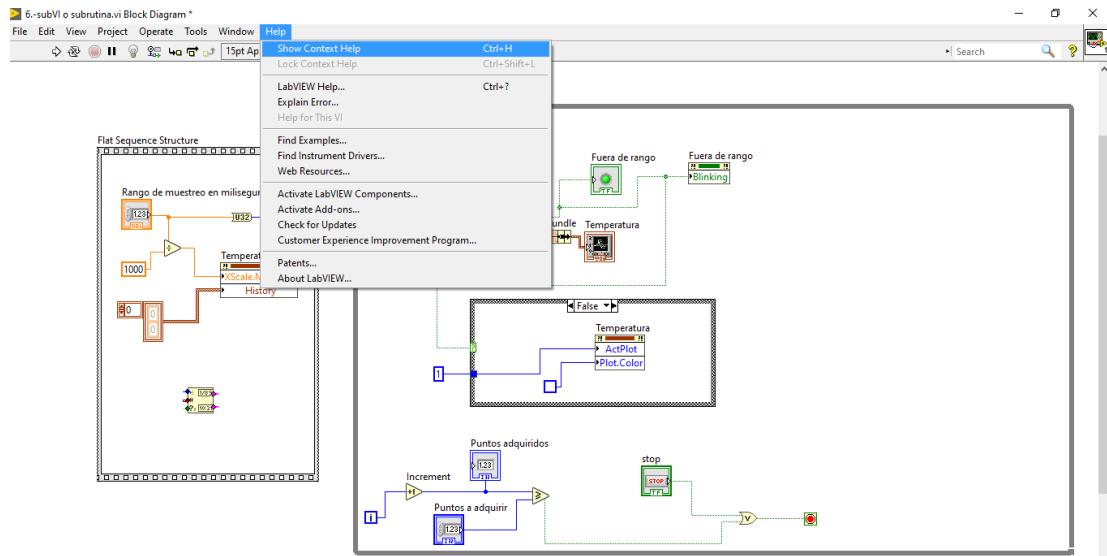
El muestreo se realiza más rápido si bajo el número de Rango de muestreo en milisegundos.



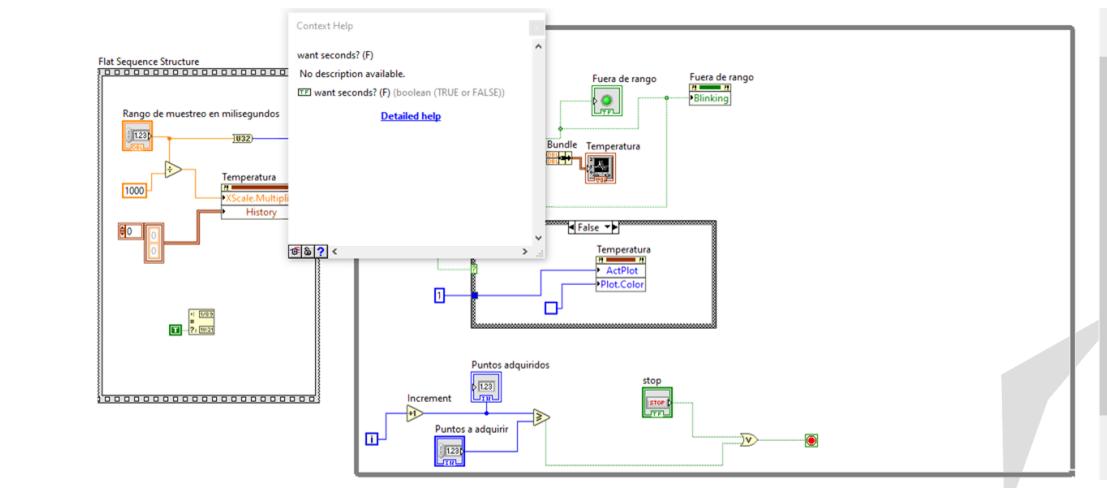
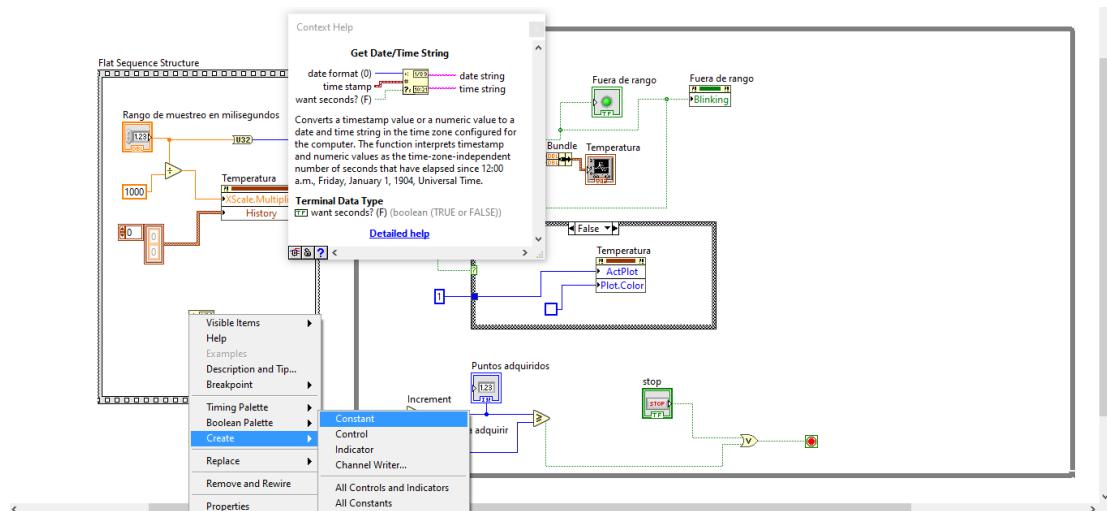
Block Diagram - Get Date/Time String: Obtención de un String con la Fecha y Hora

Vamos a añadir un elemento que me muestre la fecha y hora en la interfaz del Front Panel.



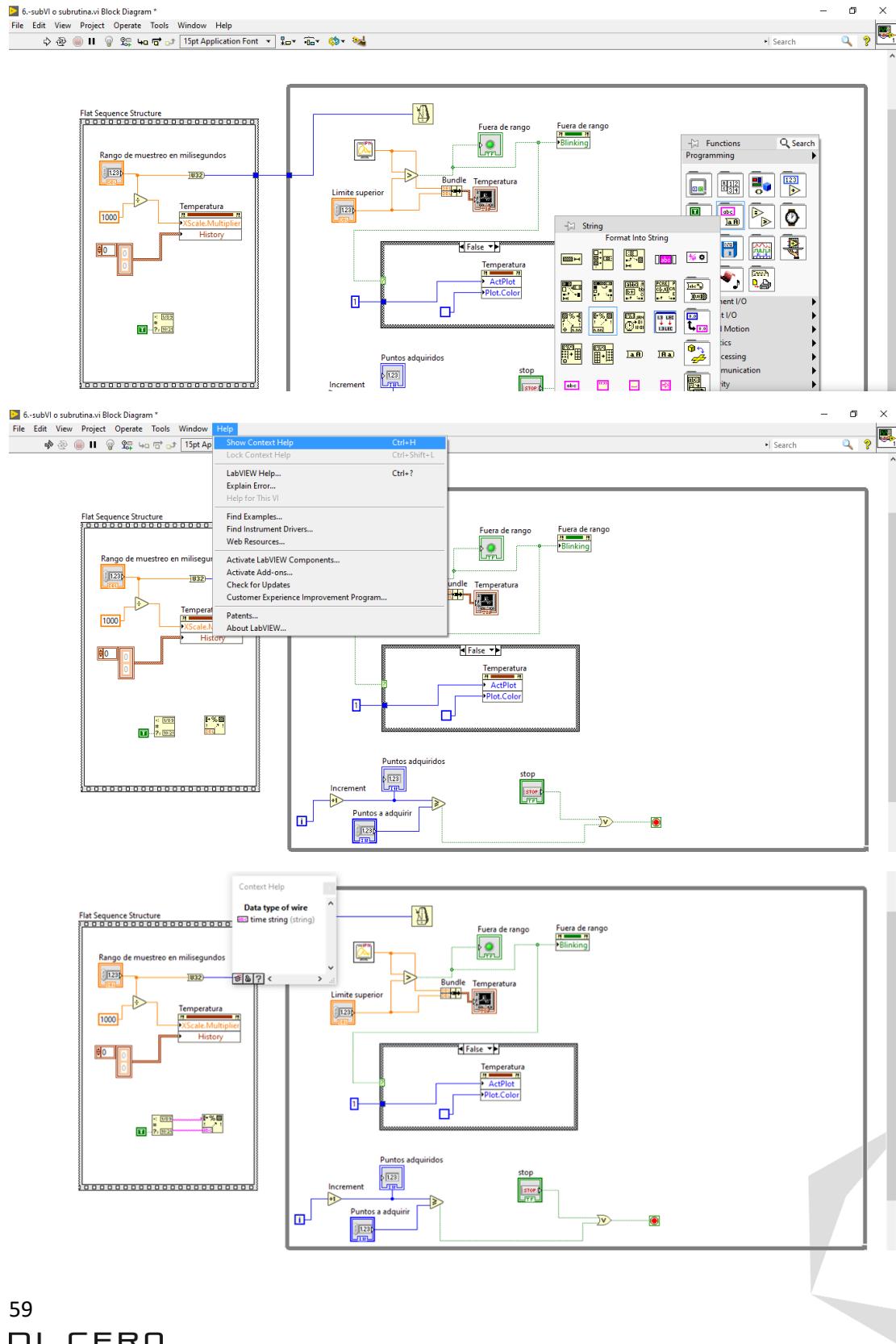


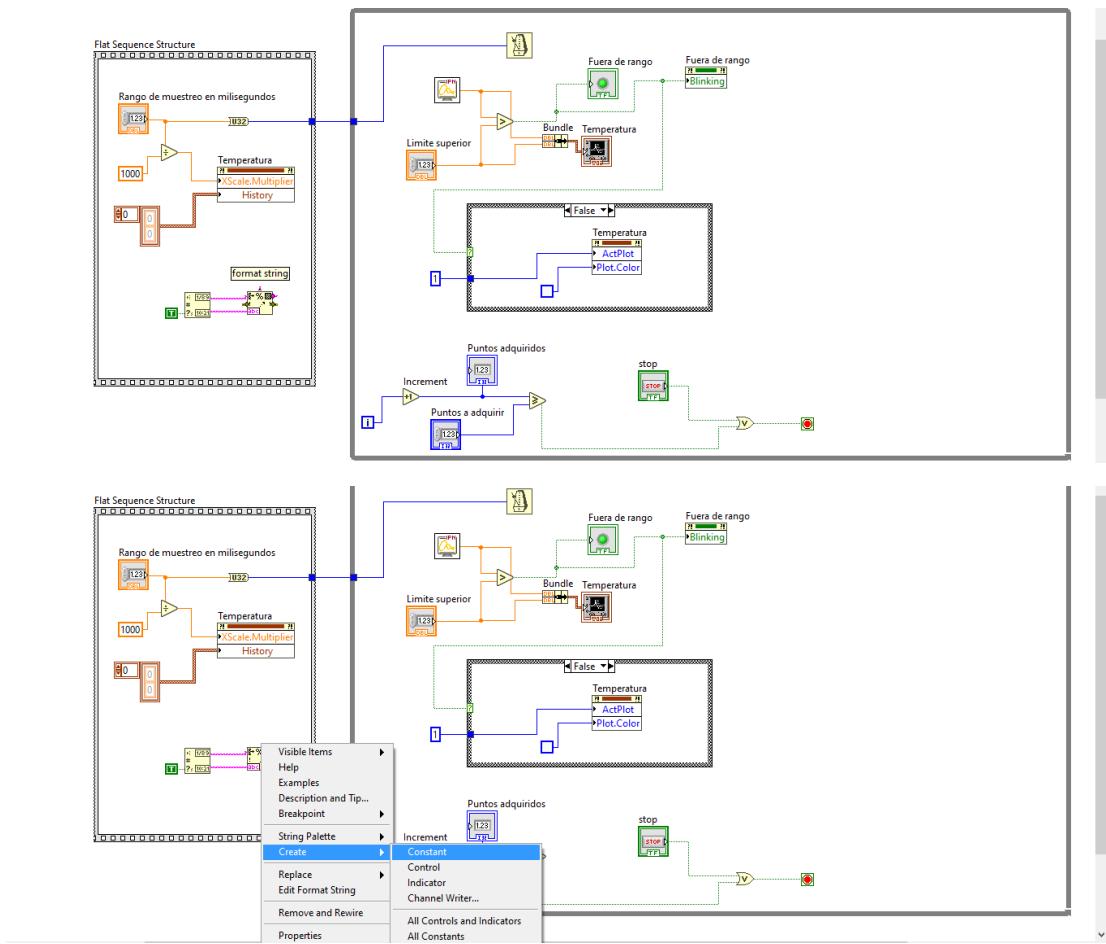
Crear una Constante para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Constant. Si cambio la variable booleana que por default está en False a True, me mostrará los segundos.



Block Diagram - Format Into String: Conversión de Cualquier Tipo de Dato a un String

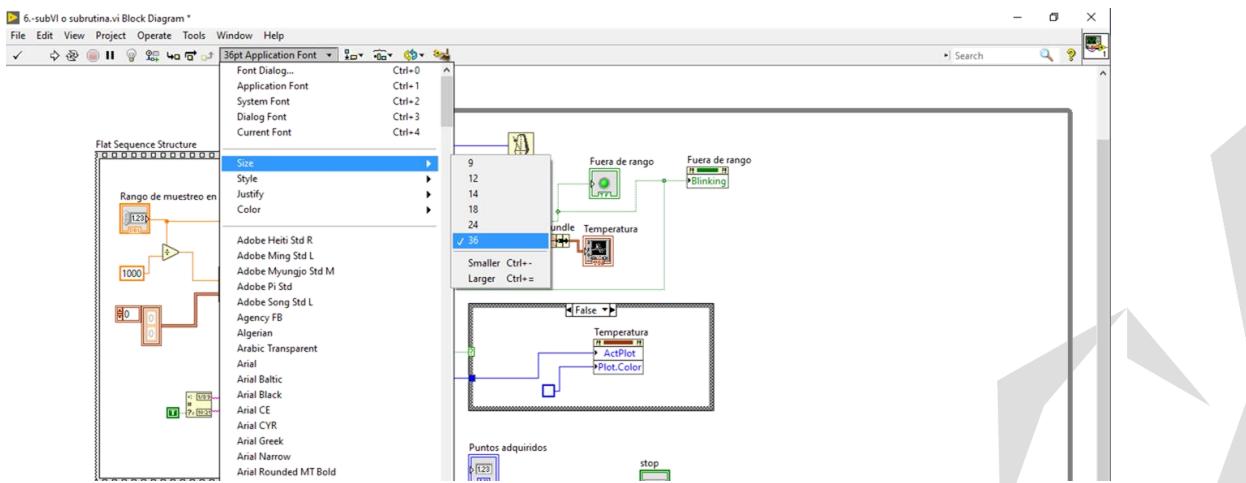
El bloque Format into string sirve para convertir cualquier tipo de dato a un String, ya sea numérico, booleano, dinámico, etc.

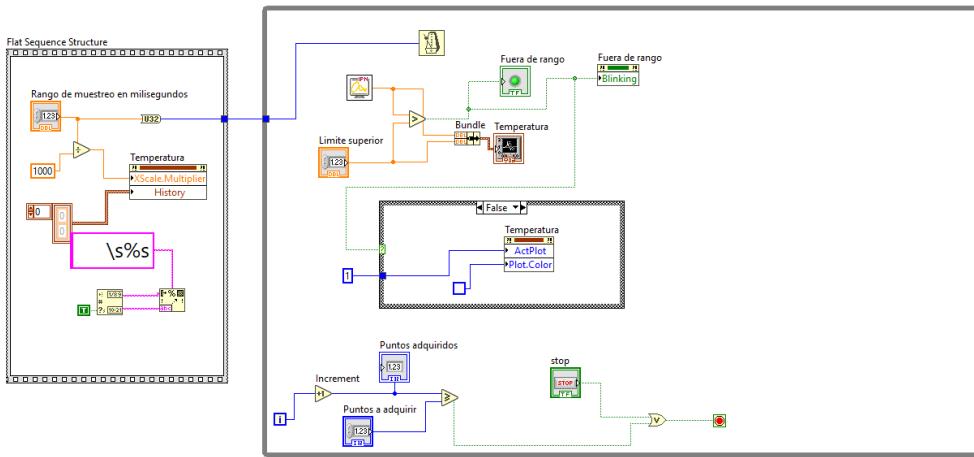




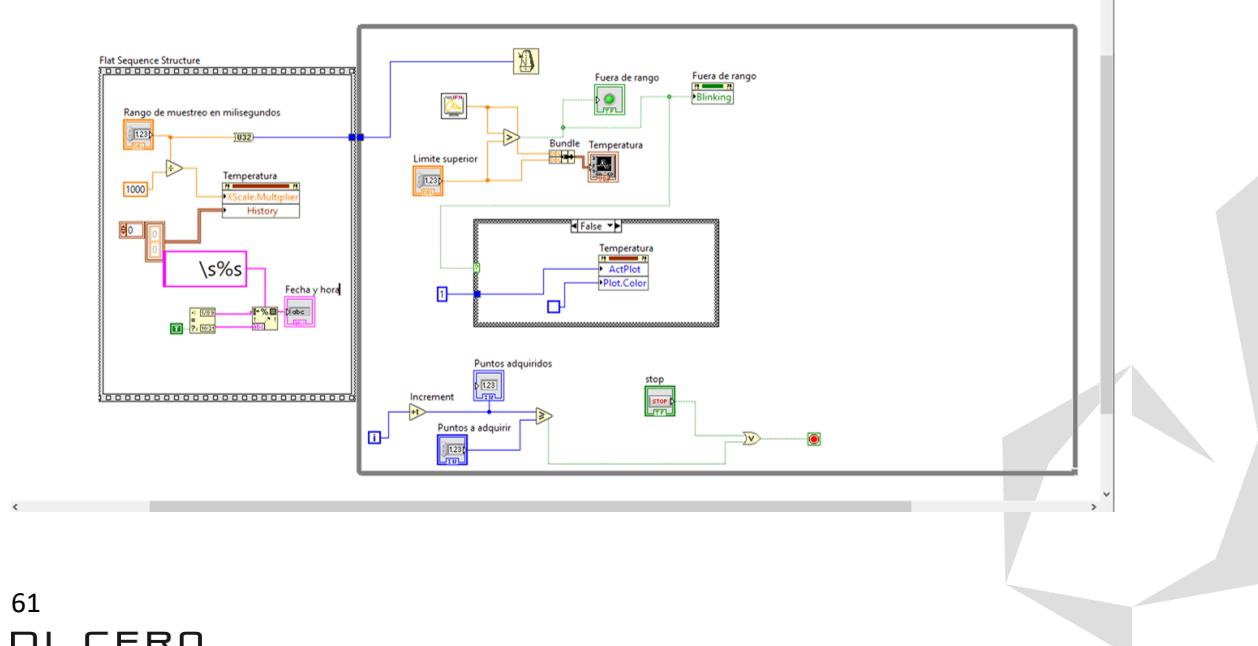
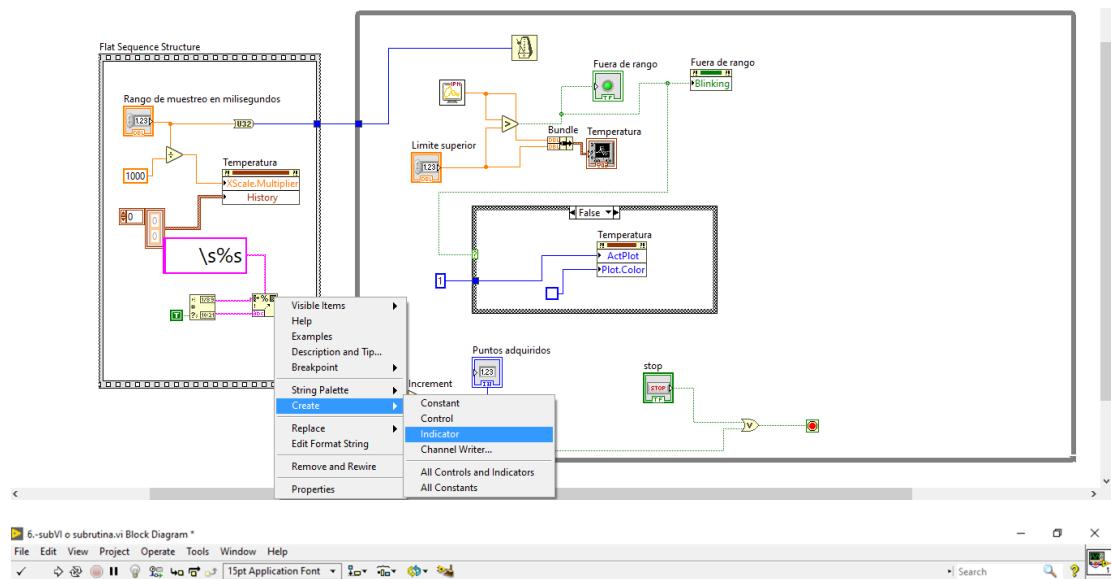
En la terminal de format string, que es la primera debemos indicar al bloque por medio de un código el formato que le va a dar al texto incluido:

- Vamos a poner el código `\s%.4f\s`
 - La parte que dice `\s` es un espacio.
 - **% Indica que se quiere dar la hora con números enteros.**
 - **Después habrá otro espacio \s.**

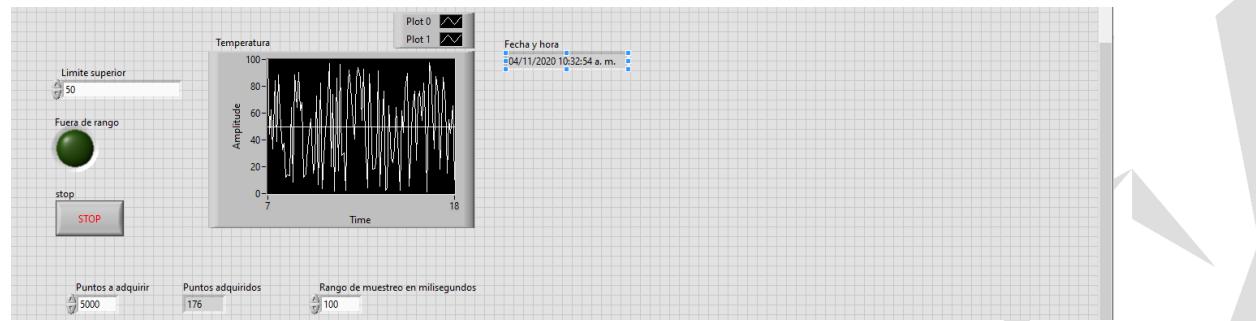
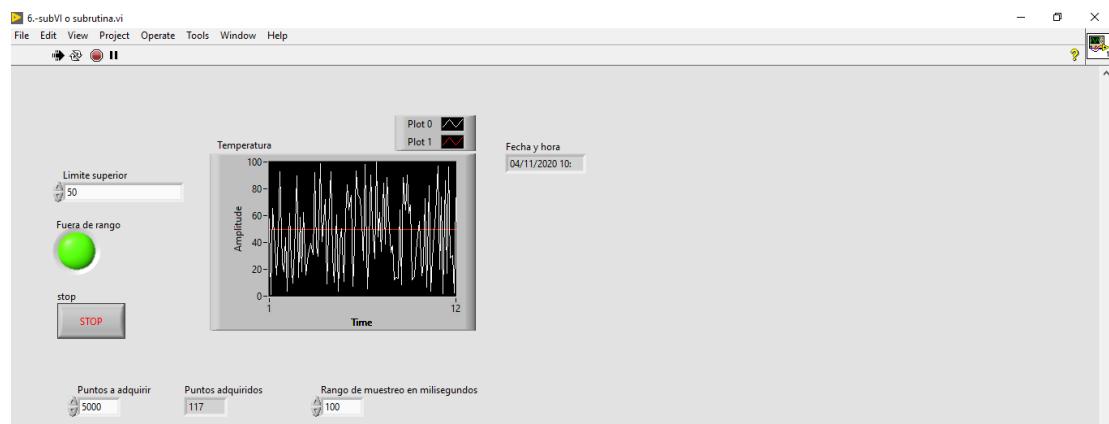
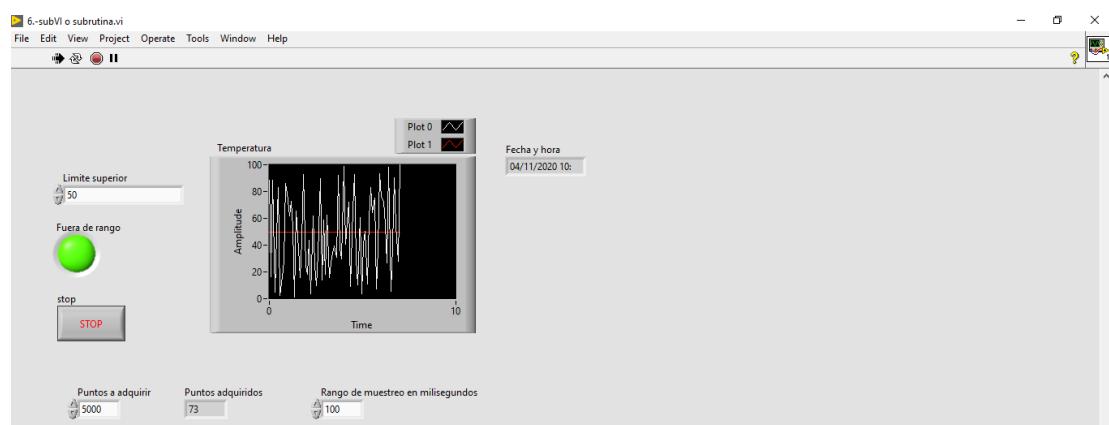




Crear un Indicador para un Bloque: Clic derecho en la terminal del bloque de interés → Create → Indicator. Con esto se crea el indicador que mostrará la hora y fecha en la interfaz del Front Panel.

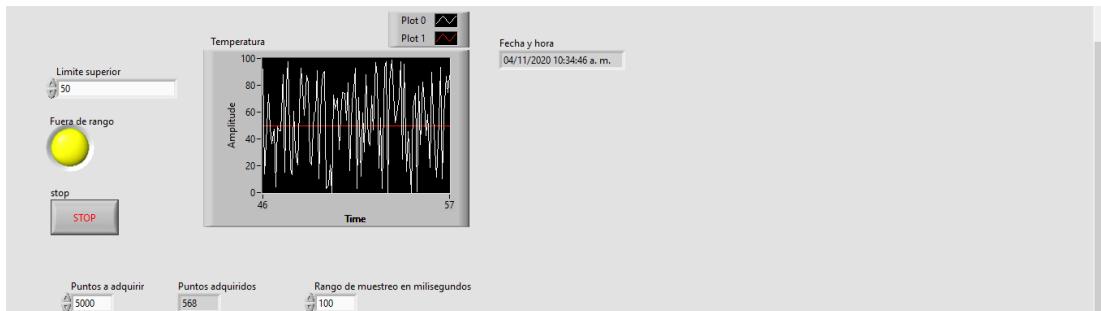


Ejecución del Programa: Señal con Led, Color en Límite Superior, Sonido, Controles y Fecha/Hora

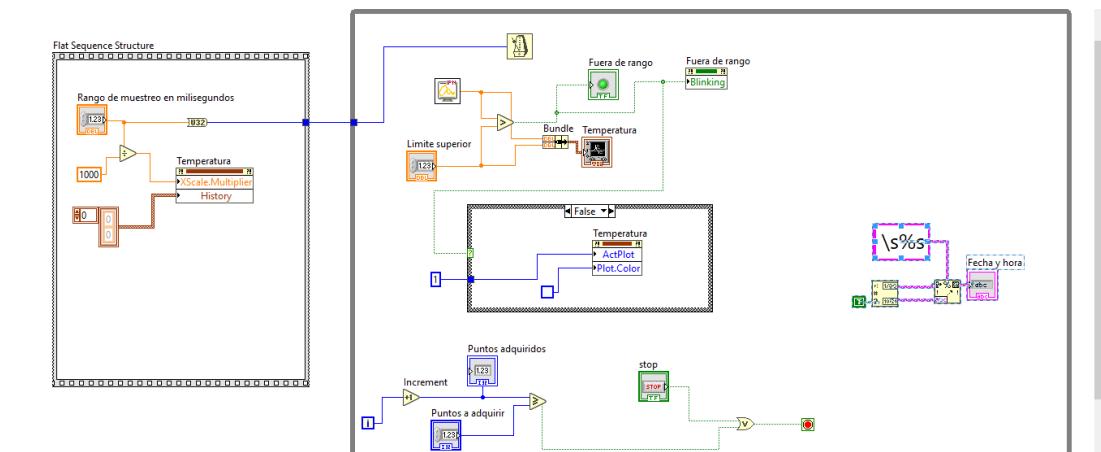




El reloj no avanza porque no está dentro del ciclo while.

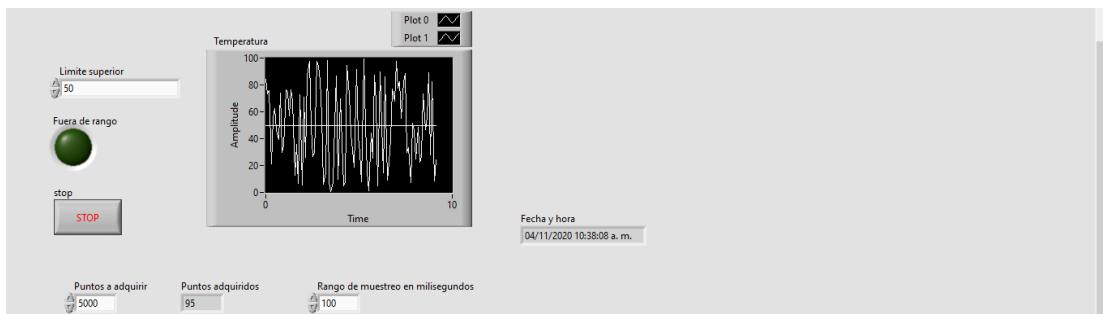
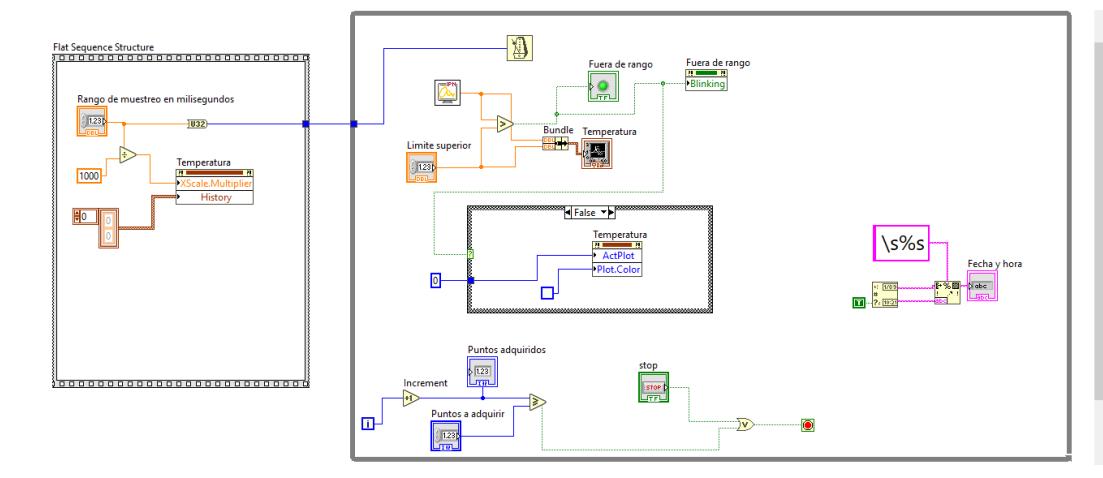


Debemos mover el bloque a estar dentro del bucle while.





Y puedo regresar la constante a valer 0 para que la gráfica cambie de color rojo a blanco como antes.



Con la constante, lo que estoy haciendo es decirle al Property Node qué gráfica dentro del Waveform Chart es la que quiero modificar, la 0 es la gráfica de números aleatorios y la 1 es la del Límite Superior.