

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

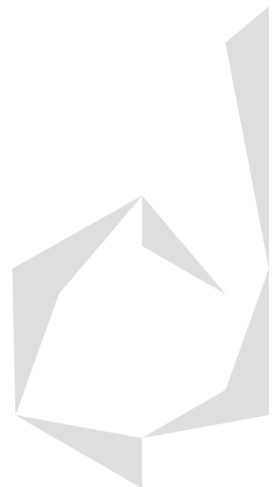
INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Visión Artificial:
Capas HSV

Contenido

Introducción Teórica de LabVIEW:.....	2
Introducción al Entorno de LabVIEW:.....	2
Front Panel: Ventana Gris con la Interfaz del Programa	4
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)	4
Show Context Help: Descripción de Bloques y sus Terminales.....	5
Front Panel y Block Diagram: Navegar de una Ventana a Otra	6
Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel	7
Block Diagram - Highlight Execution: Correr Más Lento el Programa.....	8
Coertion dot: Conversión Automática de Datos por Parte de LabVIEW	8
Block Diagram - Clean Up Diagram: Organizar Automáticamente los Bloques del VI	8
Programa: Capas HSV	9
Introducción Teórica – Capas RGB y HSV.....	9
Programa: Convertir Capas RGB en HSV – Usando Bloque de Property Node	10
Programa: Convertir Capas RGB en HSV – Sin Usar Bloque de Property Node.....	12
Programa: Capas RGB y HSV en Imagen – GUI de Lectura de Imagen	16



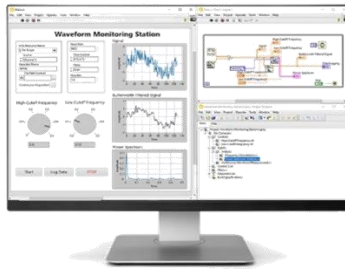
Introducción Teórica de LabVIEW:

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

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

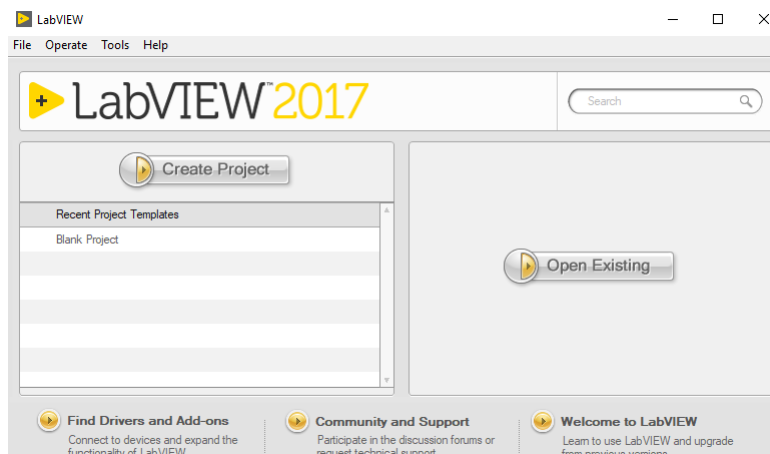
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

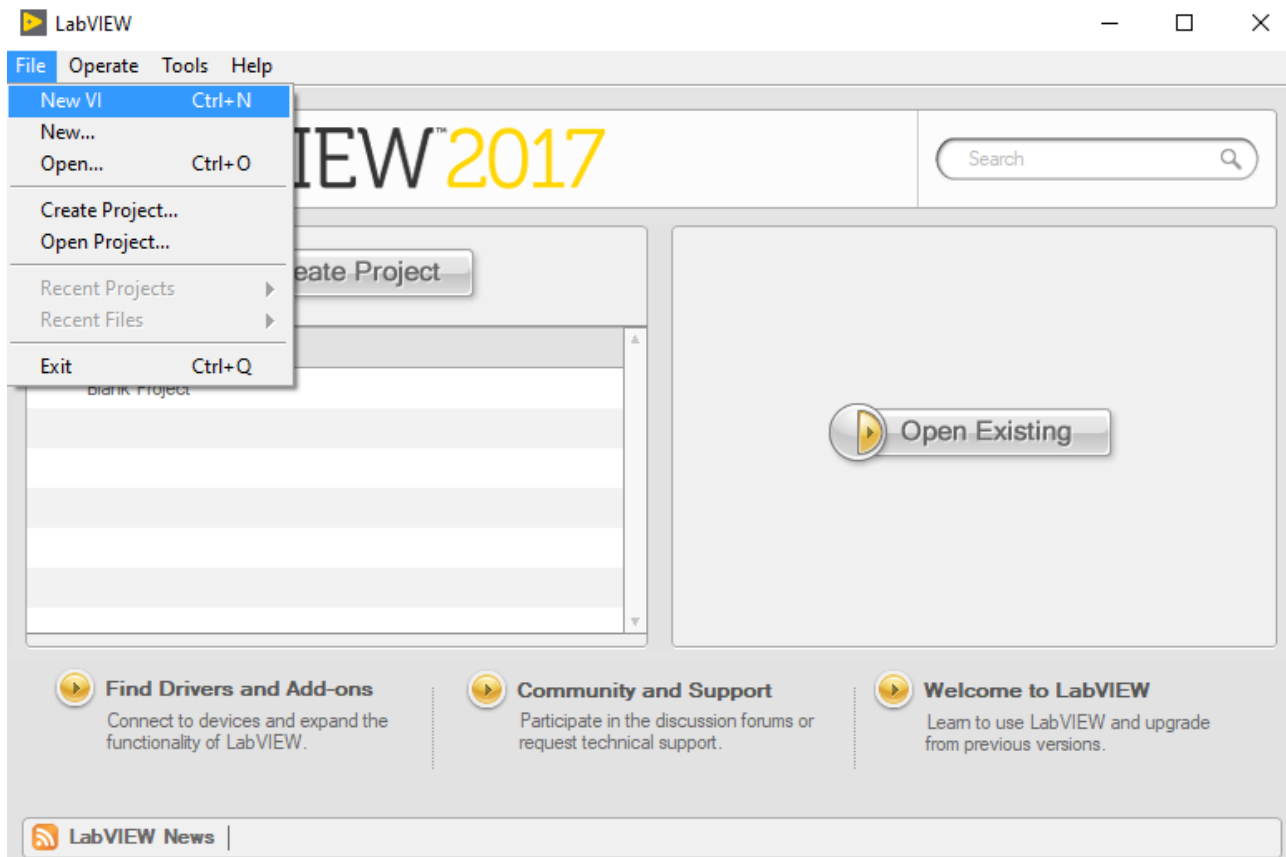


Introducción al Entorno de LabVIEW:

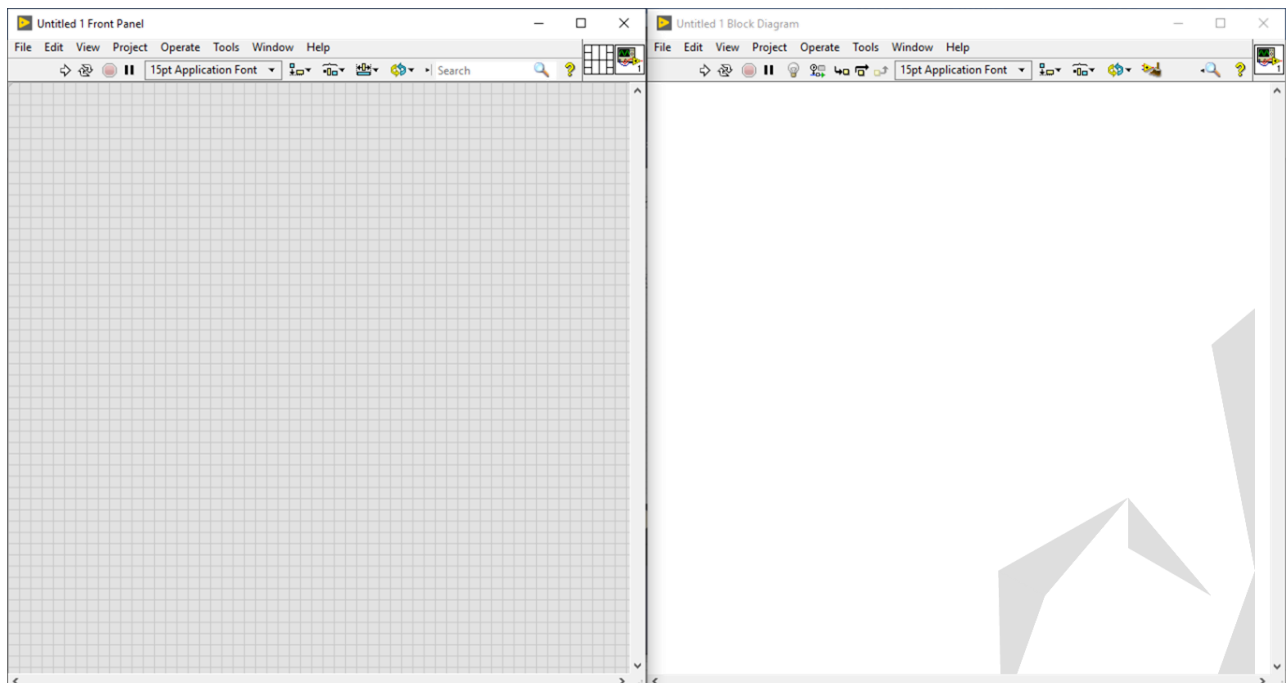
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

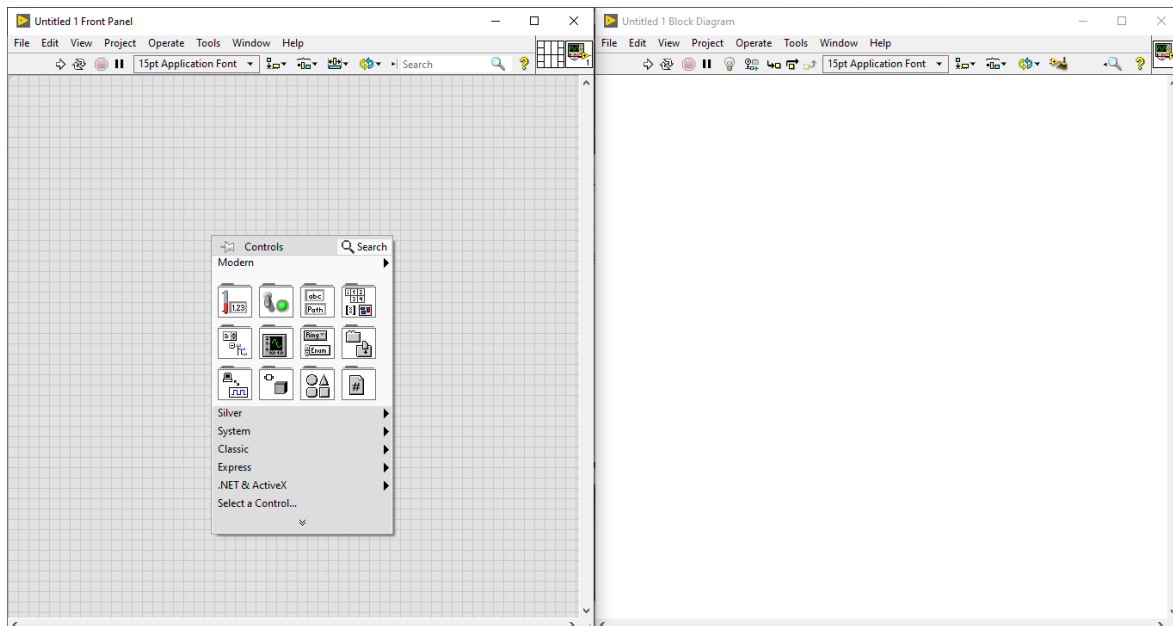


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



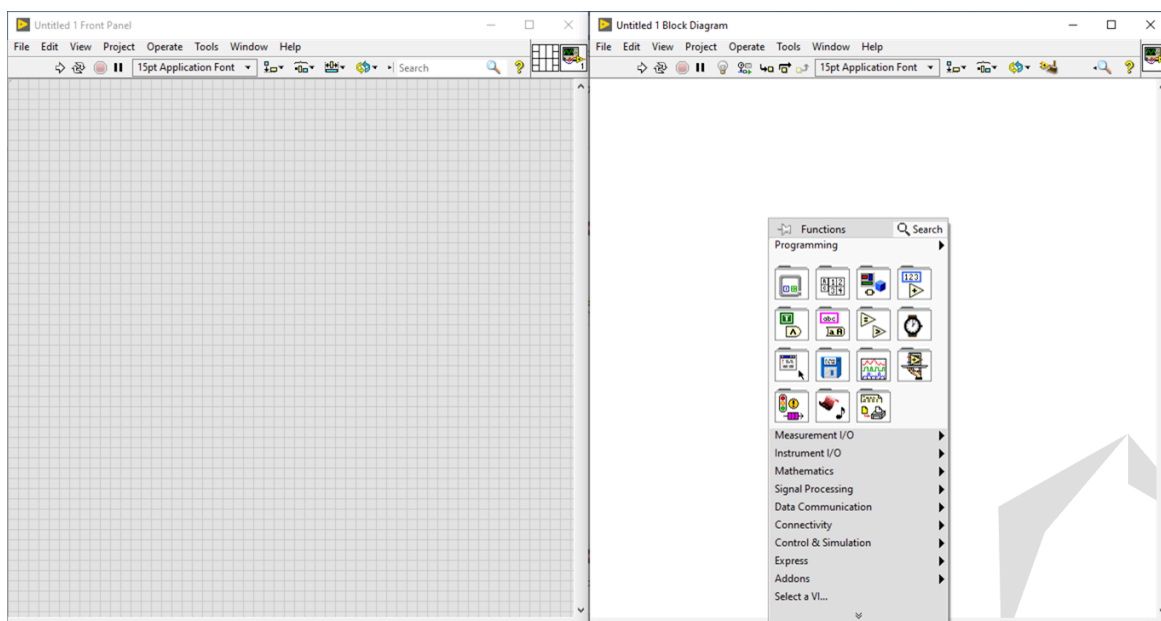
Front Panel: Ventana Gris con la Interfaz del Programa

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



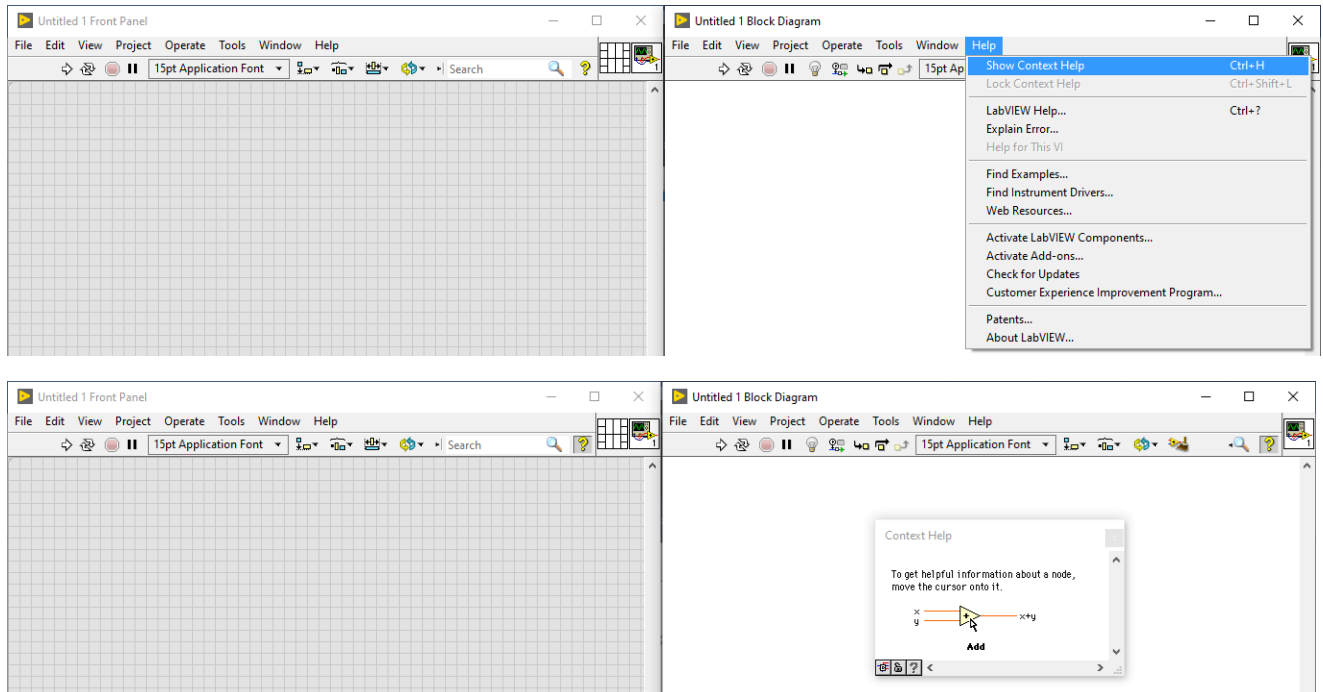
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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



Show Context Help: Descripción de Bloques y sus Terminales

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



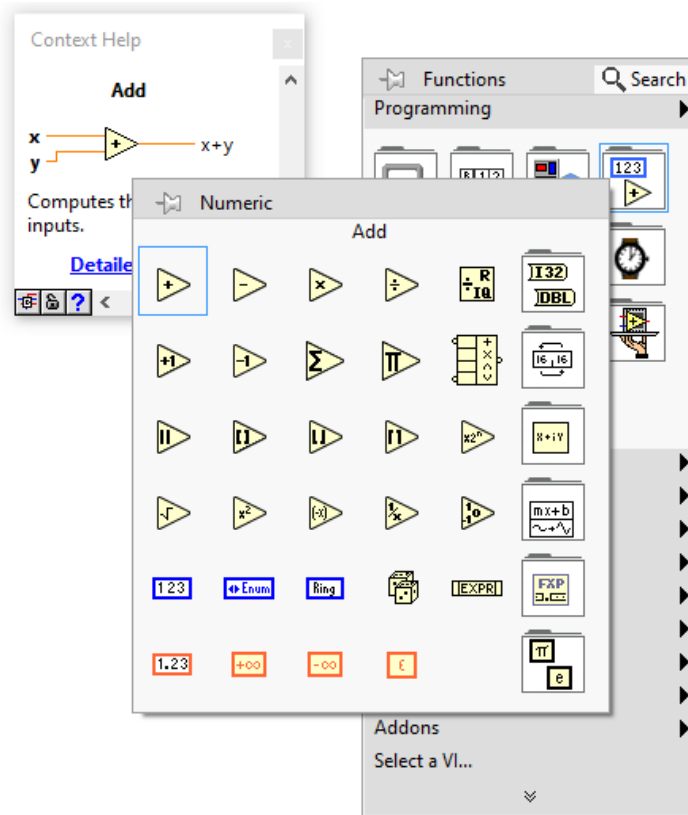
Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:

VIs Express, VIs y Funciones

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



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

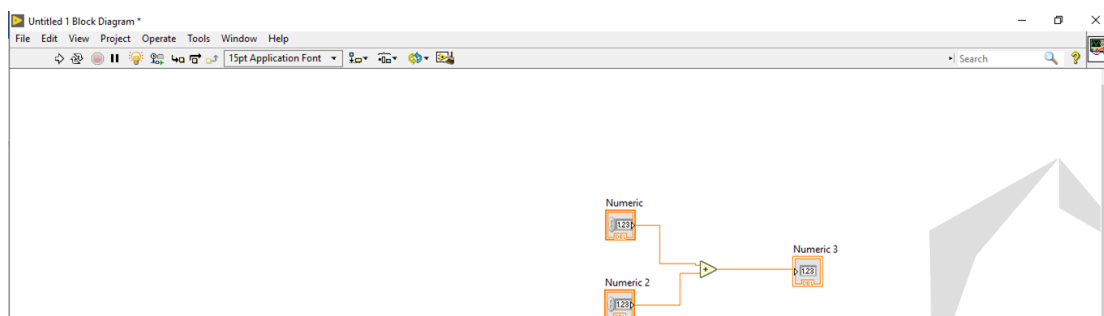


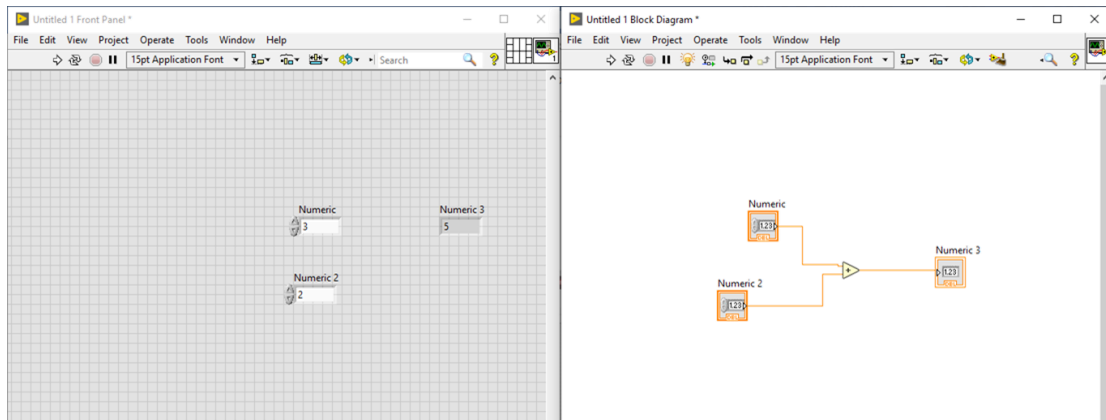
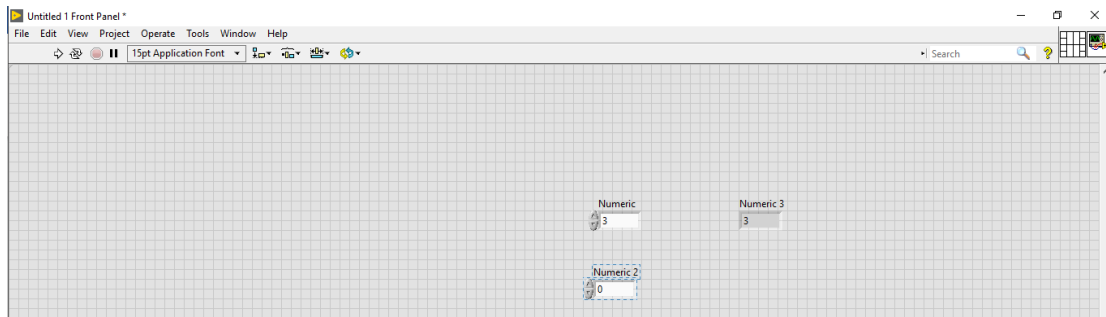
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

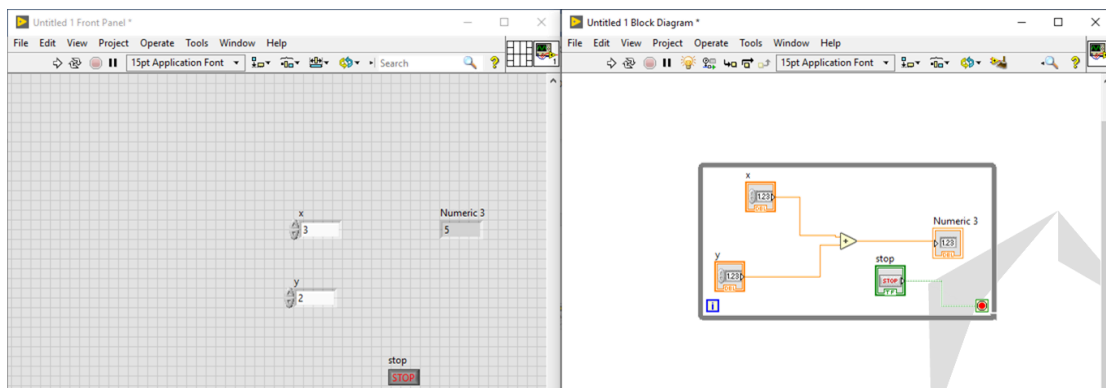
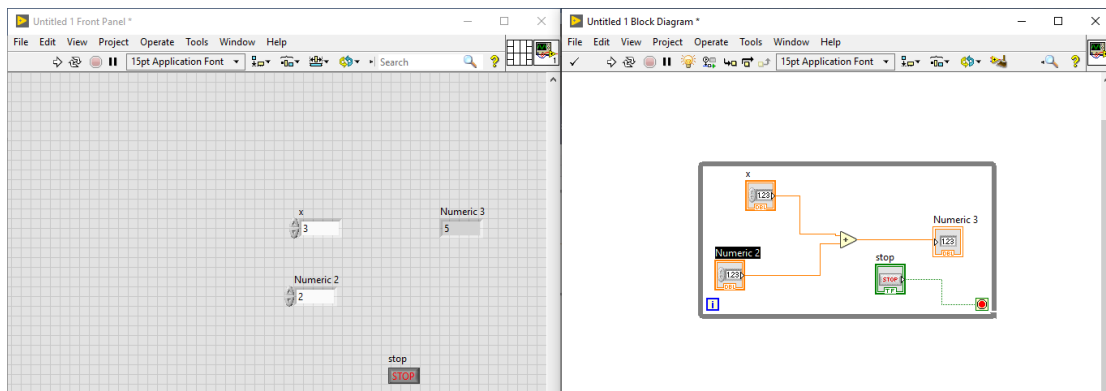
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

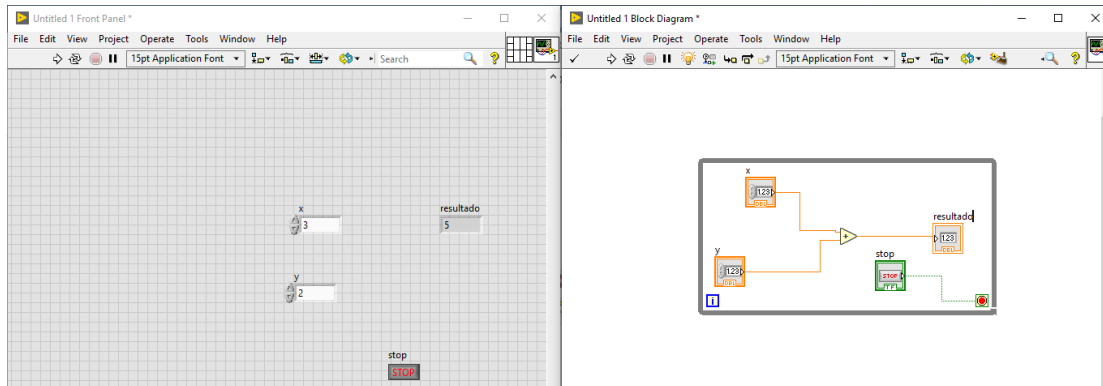




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

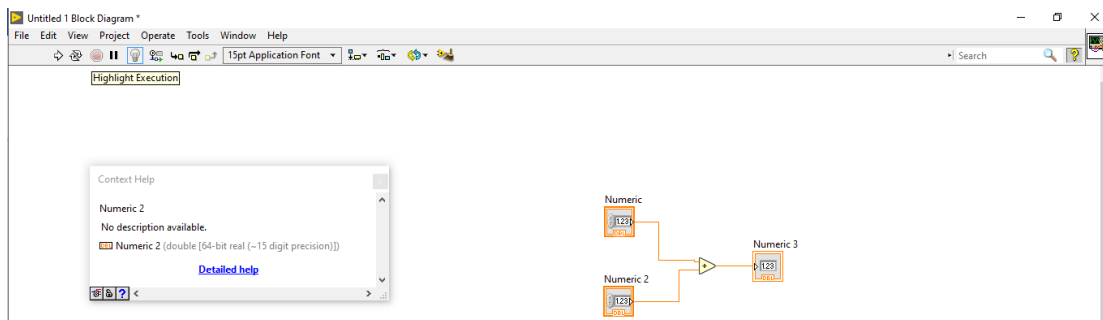
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





Block Diagram - Highlight Execution: Correr Más Lento el Programa

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

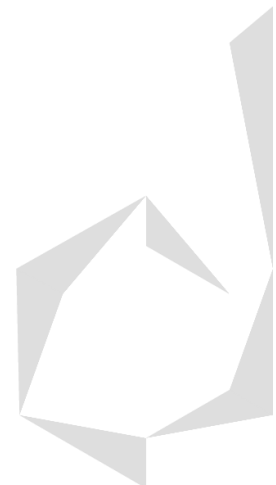


Coertion dot: Conversión Automática de Datos por Parte de LabVIEW

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

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

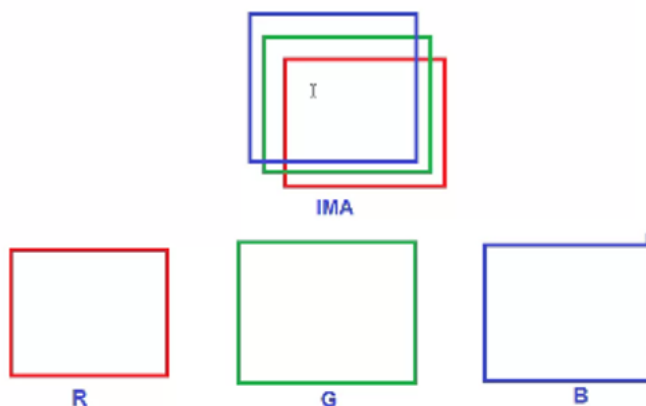
Con el botón de Clean Up Diagram que se encuentra en la parte superior derecha del Block Diagram se organizan mejor y de forma automática mis elementos.



Programa: Capas HSV

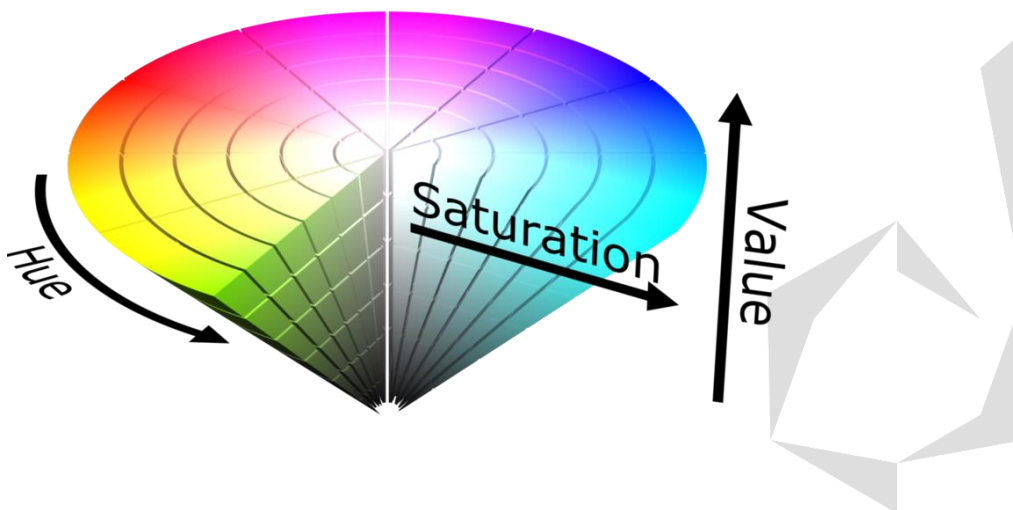
Introducción Teórica – Capas RGB y HSV

Las imágenes digitales son una matriz 3D que contiene valores de 0 a 255 en cada uno de sus píxeles, se compone de 3 capas, cada una representando uno de los tres colores primarios R, G y B. Pero esta no es la única representación de los colores que conforman una imagen, también existe la representación HSV.

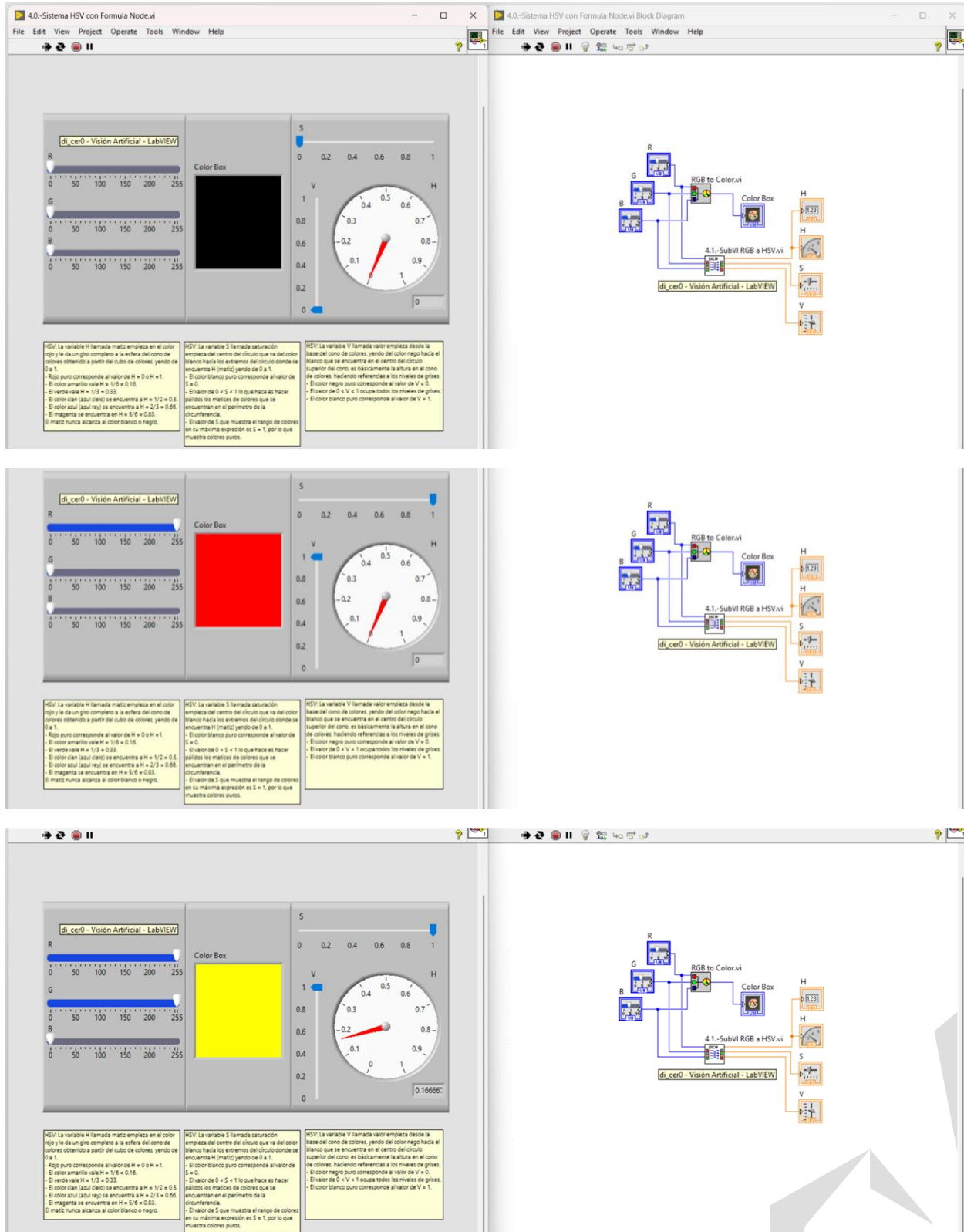


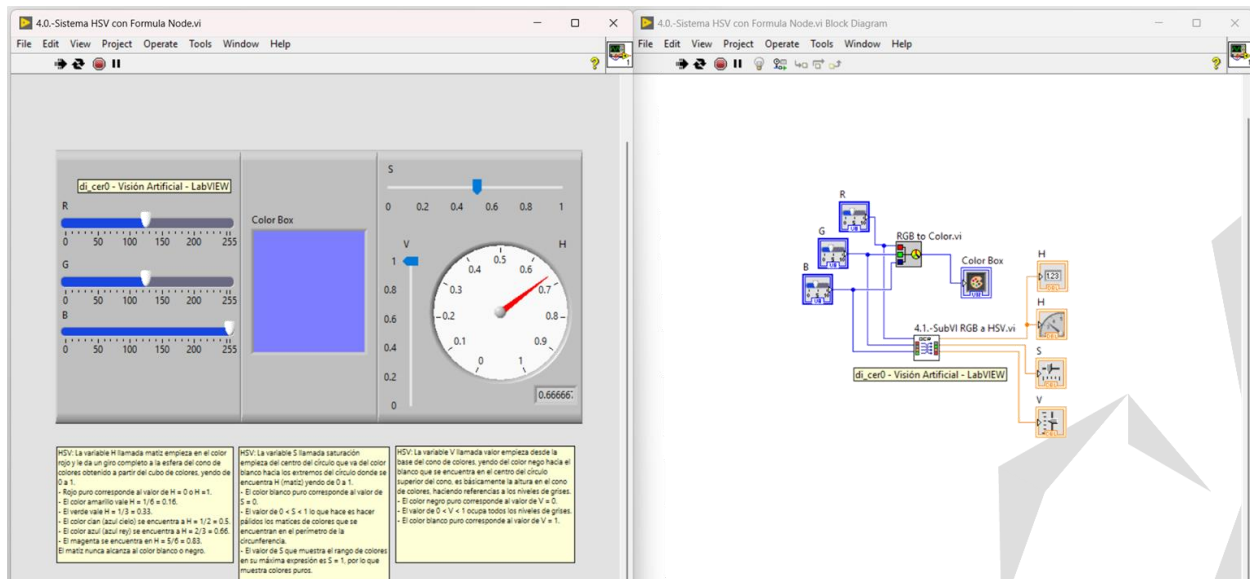
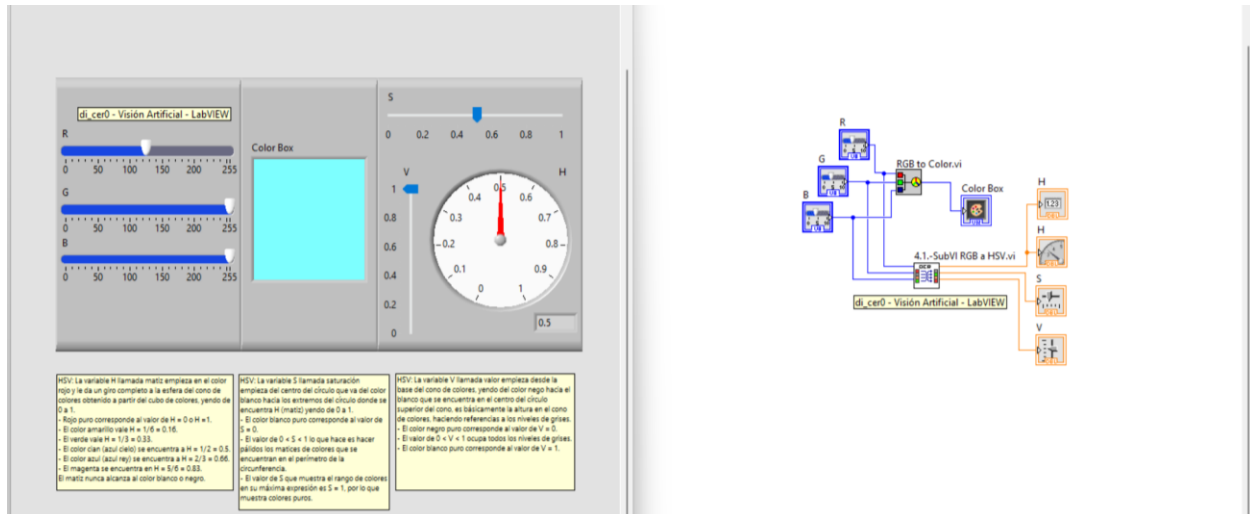
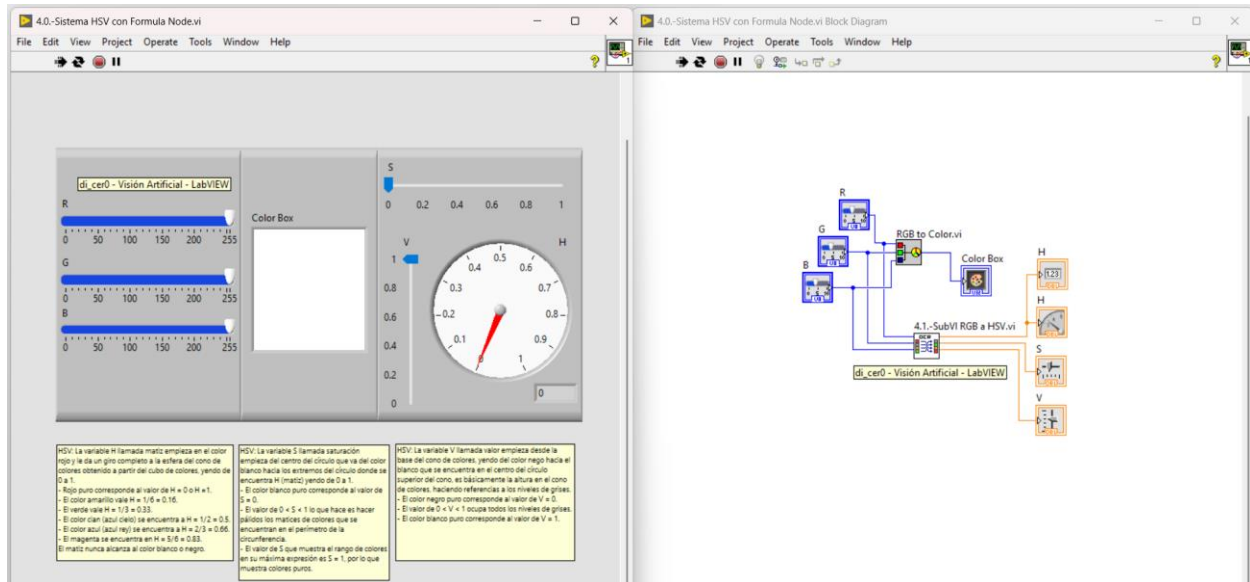
Las capas HSV (Matiz, Saturación y Valor) son una representación de los colores que puede adoptar una imagen digital de una manera más intuitiva y perceptual que las capas RGB.

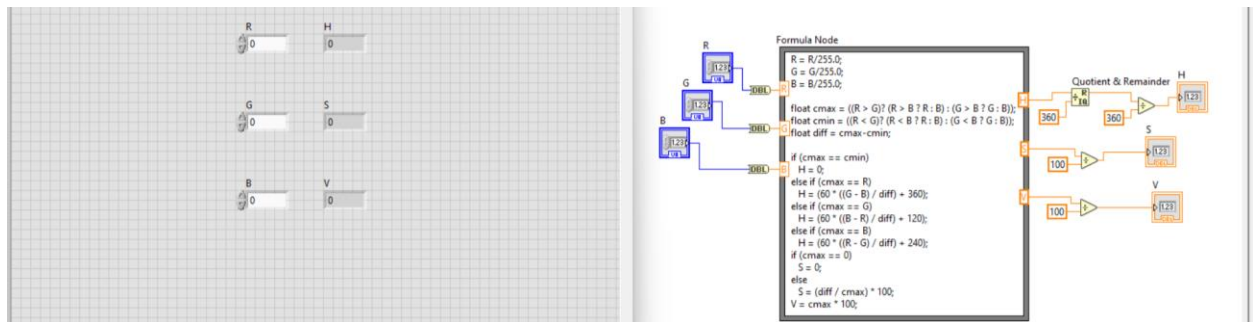
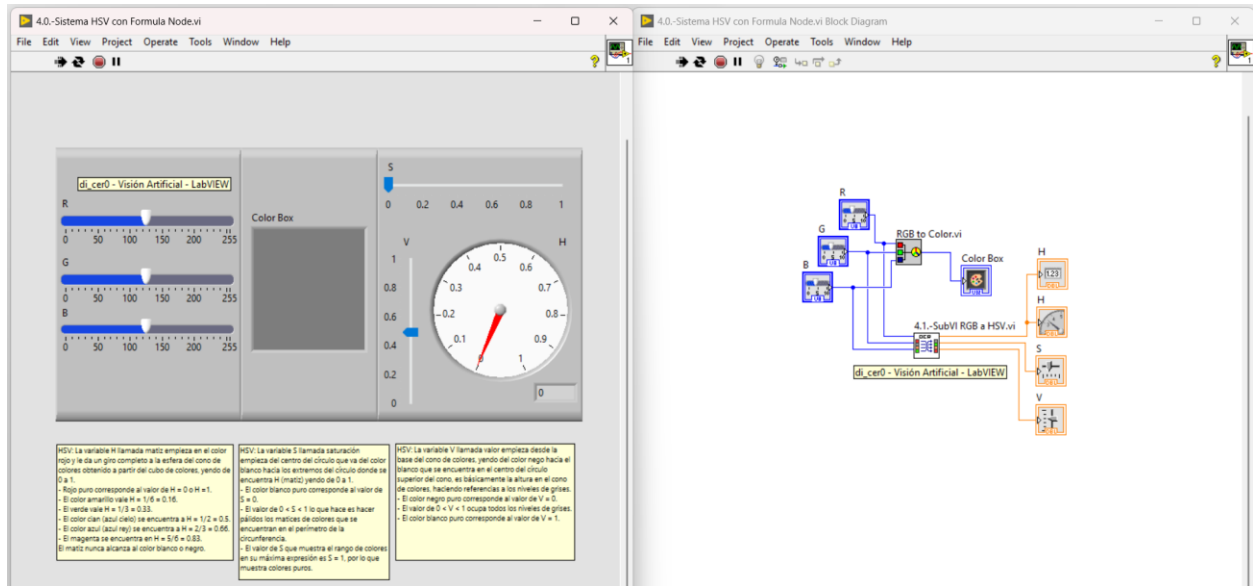
- **Matiz (Hue):** Representa el tono del color dominante. Se expresa en grados y forma un círculo de 360 grados que contienen todos los colores posibles que se pueden crear con una combinación de tonos RGB.
- **Saturación (Saturation):** Indica la pureza o intensidad del color. Un valor de saturación alto significa que el color es más vibrante y puro, mientras que un valor bajo se acerca al gris y reduce la intensidad. La saturación se mide en porcentajes, donde 0% representa una ausencia total de saturación (blanco) y 100% representa la máxima saturación (color intenso).
- **Valor o Brillo (Value/Brightness):** Representa el brillo o la cantidad de luz presente en el color. Un valor alto indica mayor brillo, mientras que un valor bajo indica menor brillo. El valor también se mide en porcentajes, donde 0% representa la ausencia total de luz (negro) y 100% representa la máxima luminosidad (blanco).



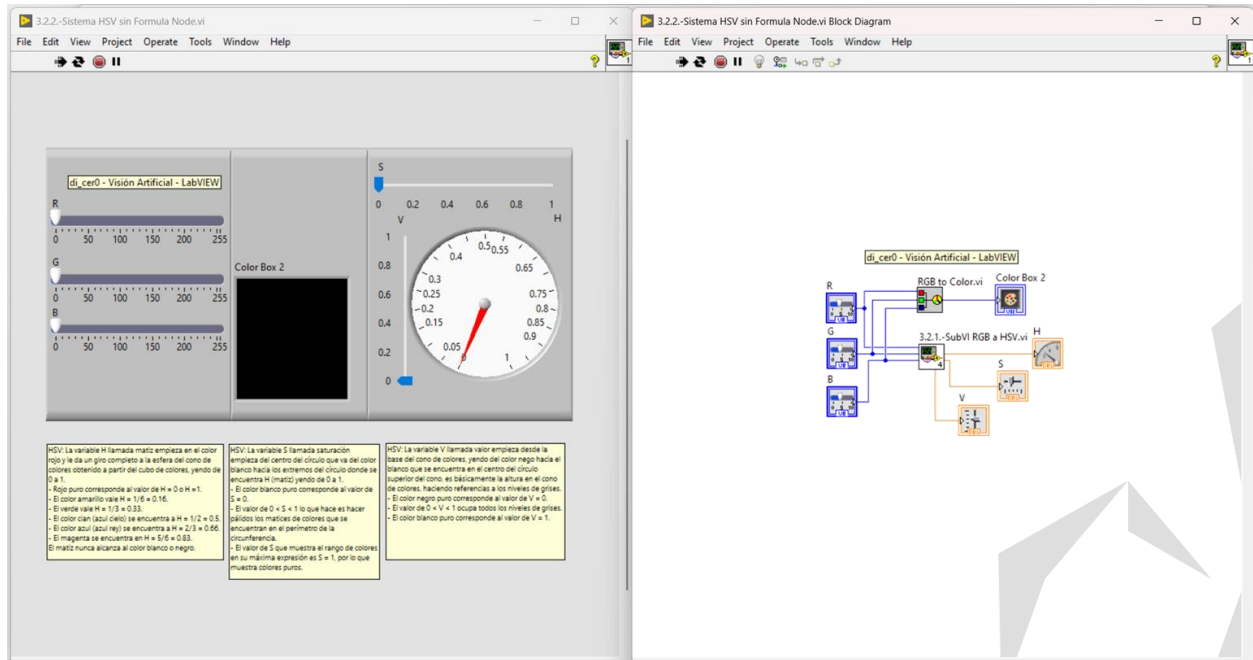
Programa: Convertir Capas RGB en HSV – Usando Bloque de Property Node

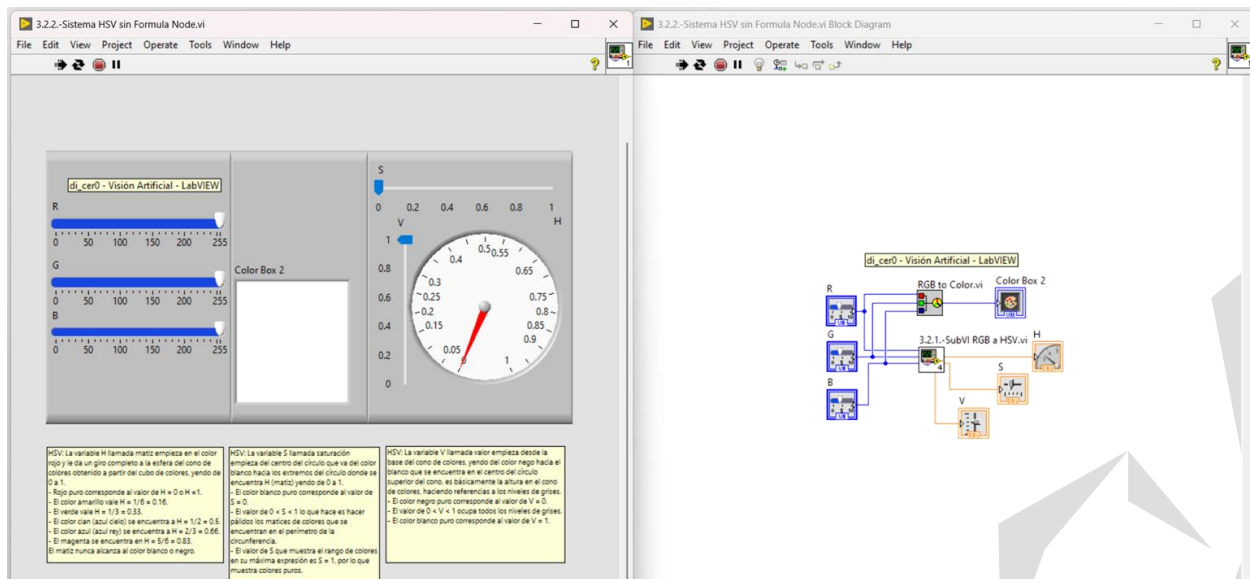
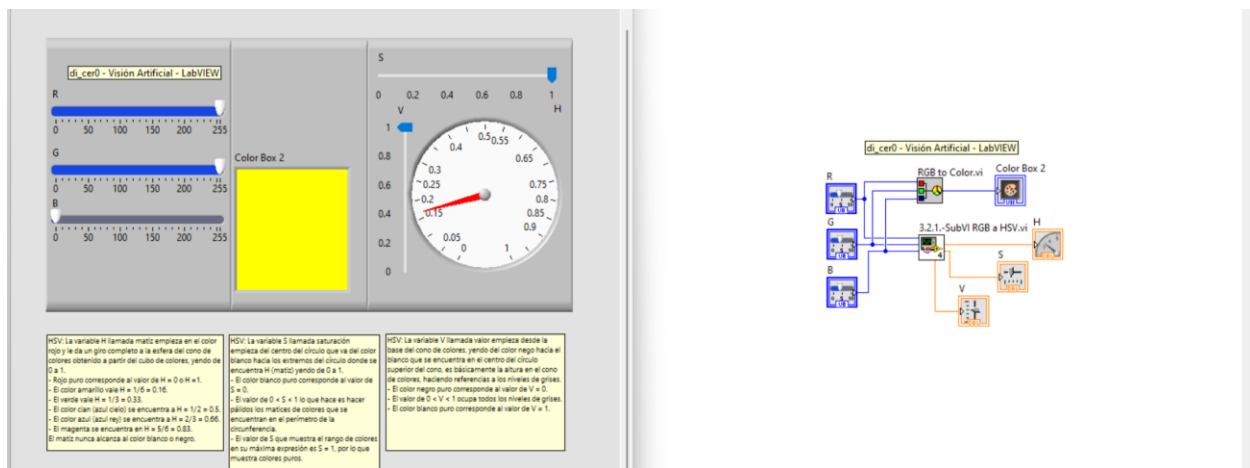
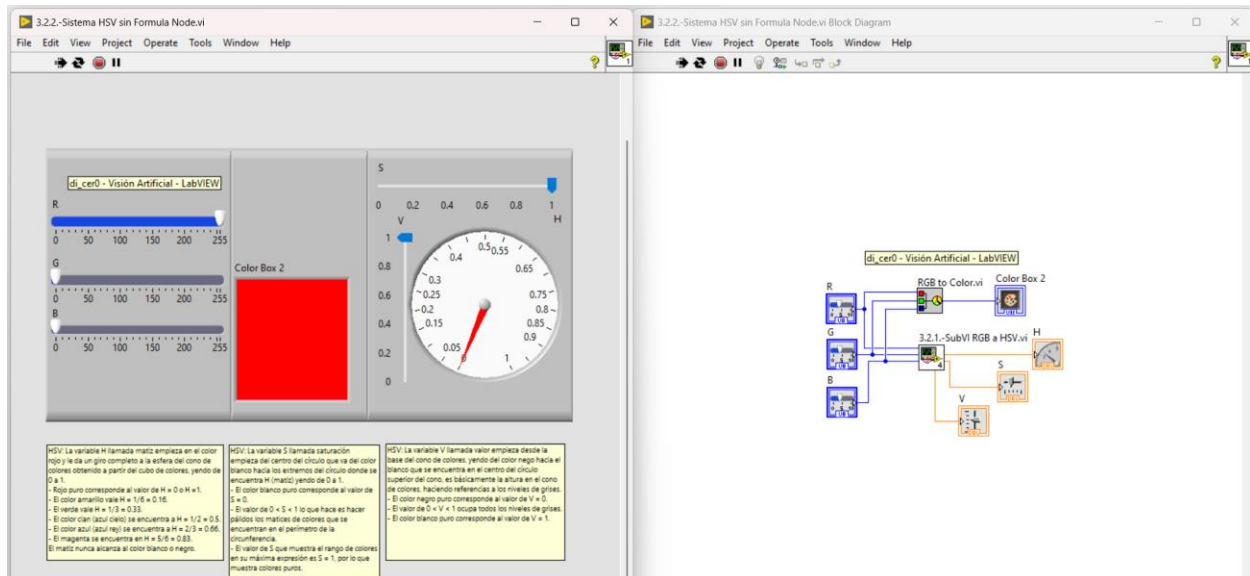


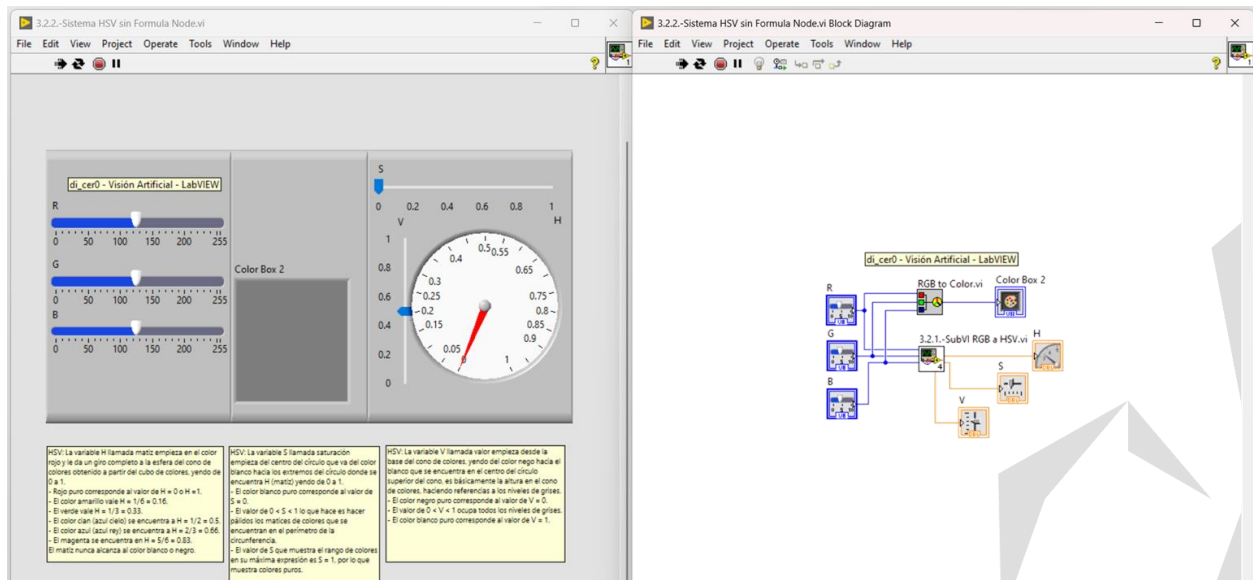
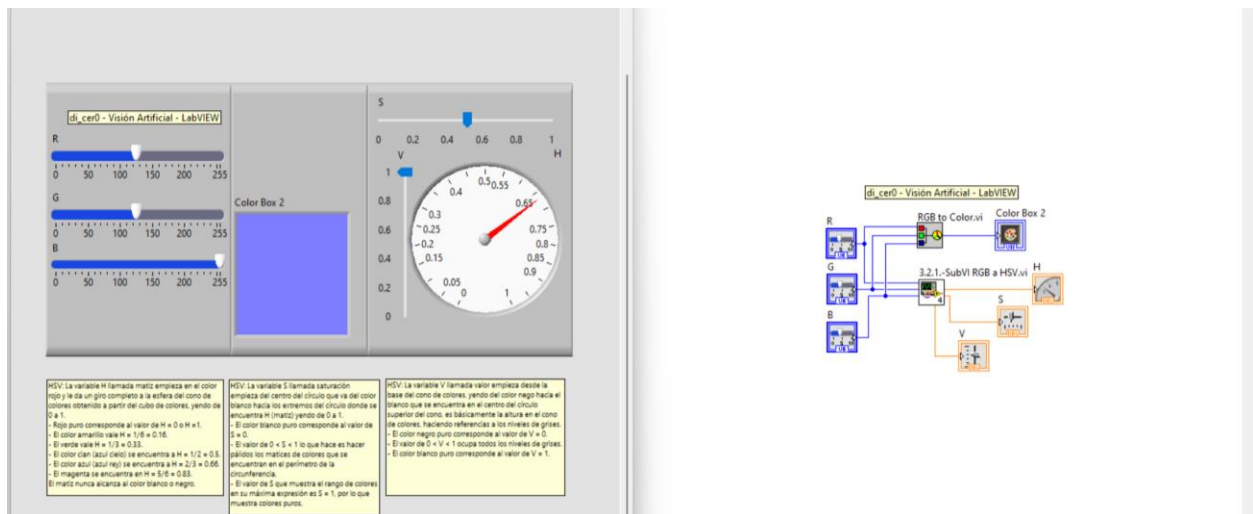
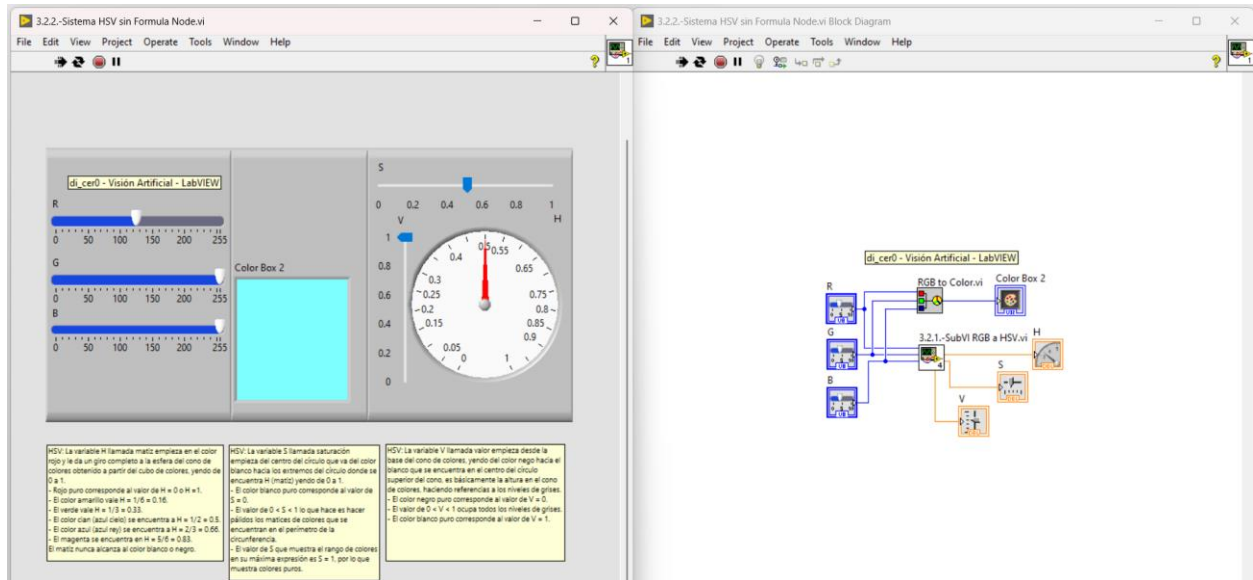


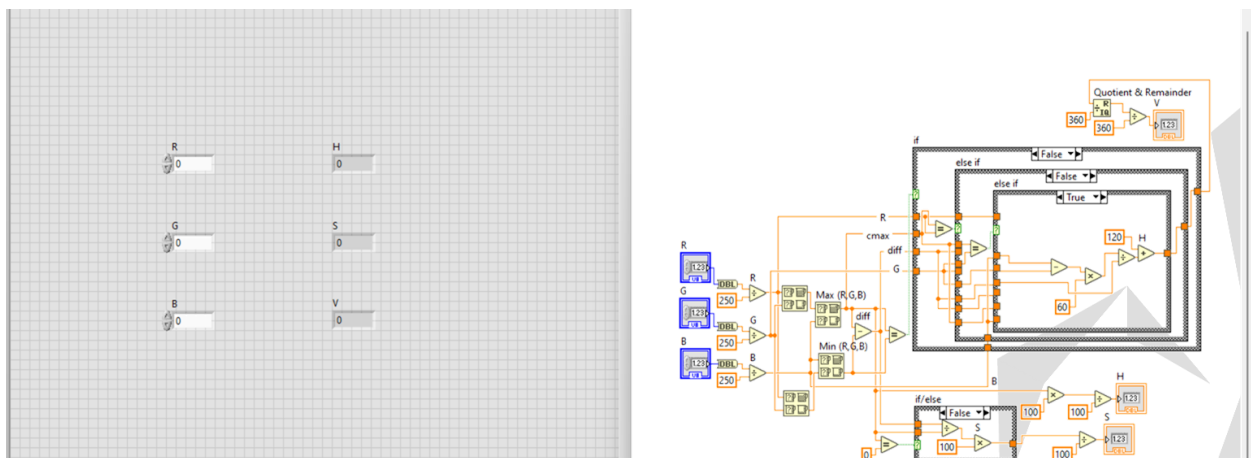
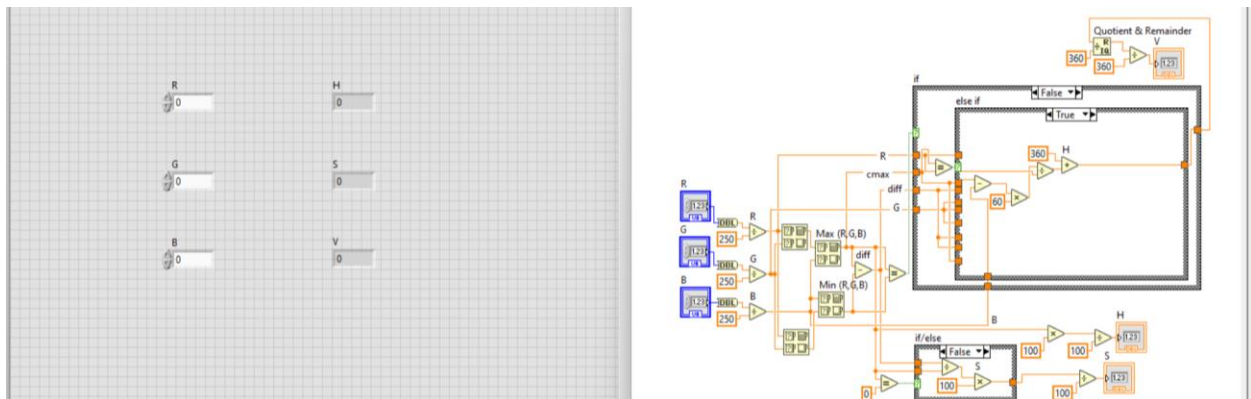
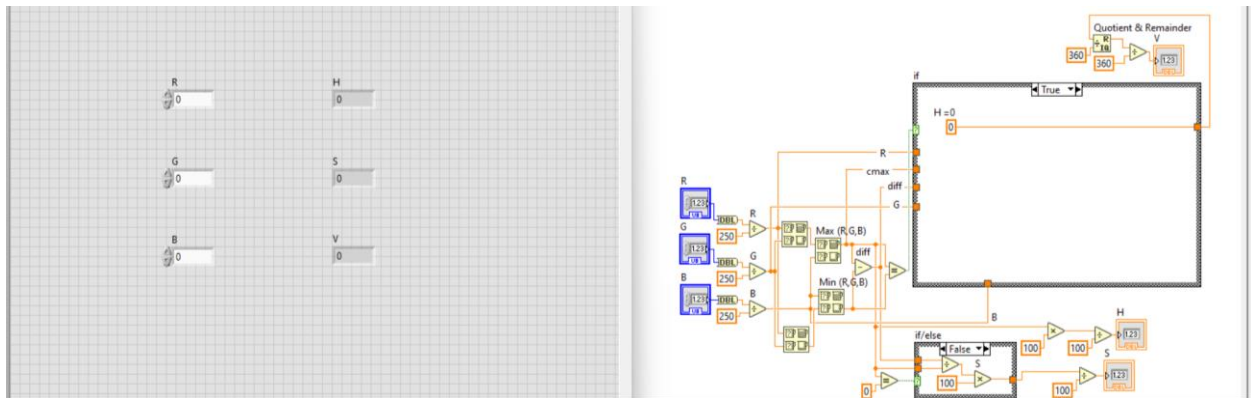
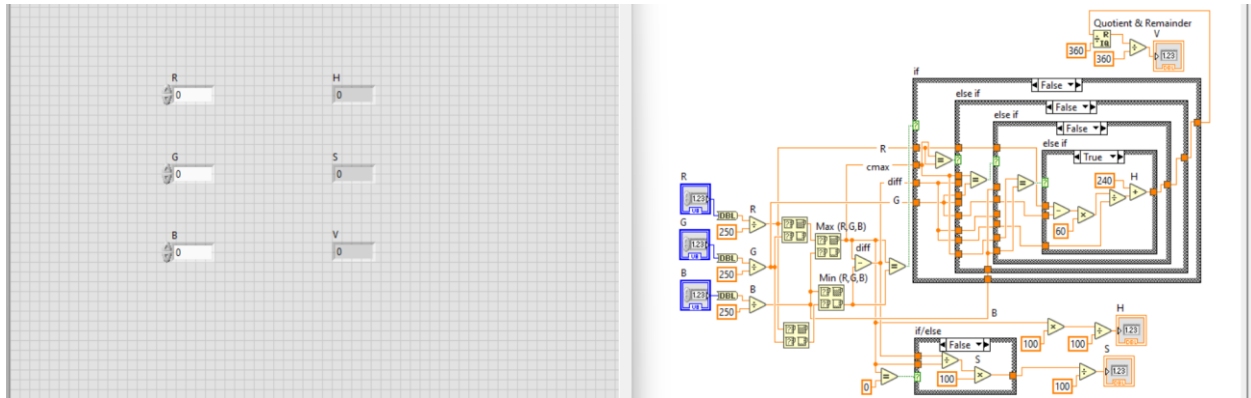


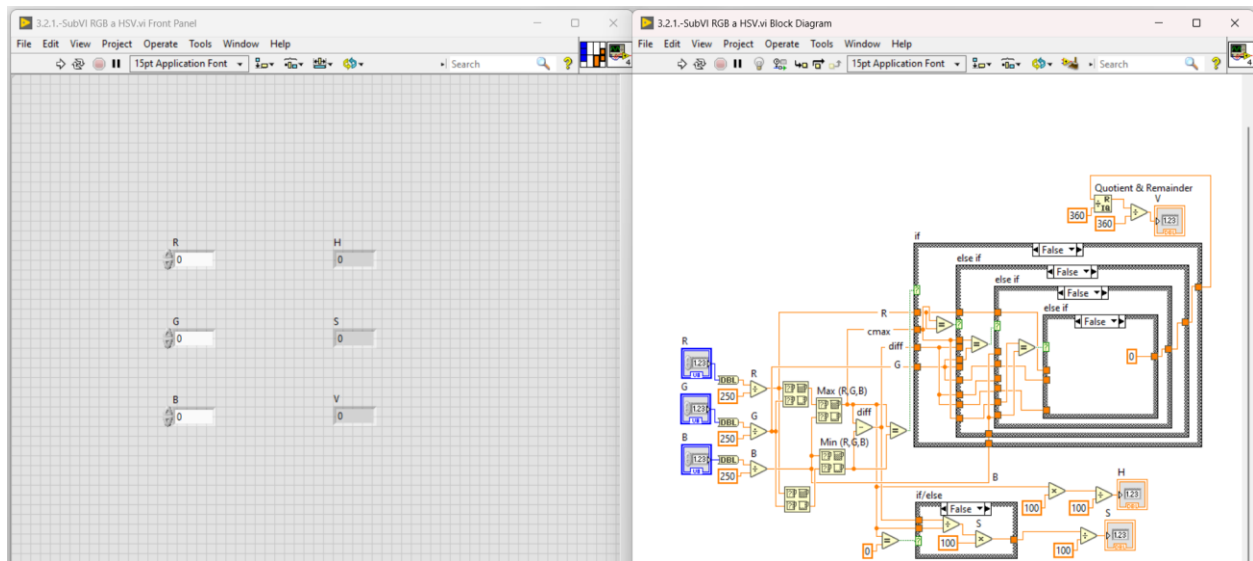
Programa: Convertir Capas RGB en HSV – Sin Usar Bloque de Property Node











Programa: Capas RGB y HSV en Imagen – GUI de Lectura de Imagen

