

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

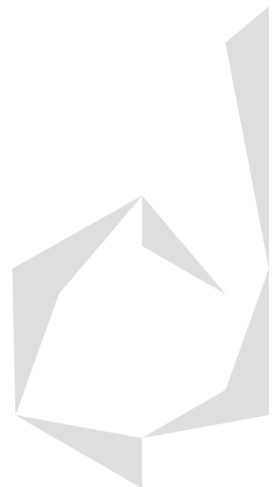
INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Visión Artificial:
Momentos Invariantes de Hu

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: Momentos Invariantes de Hu	9
Introducción Teórica – Momentos Invariantes de Hu	9
Programa: Momento Invariante de Hu Inicial – Mpq.....	9
Programa: Momento Invariante de Hu Central – Upq	9
Programa: Momento Invariante de Hu Normalizado – Npq	10
Programa: Momentos Invariante de Hu – h1, h2, h3, h4, h5, h6 y h7	10
Programa: Momentos Invariantes de Hu – Reconocimiento Patrones.....	11



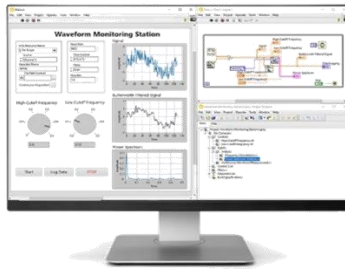
Introducción Teórica de LabVIEW:

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.

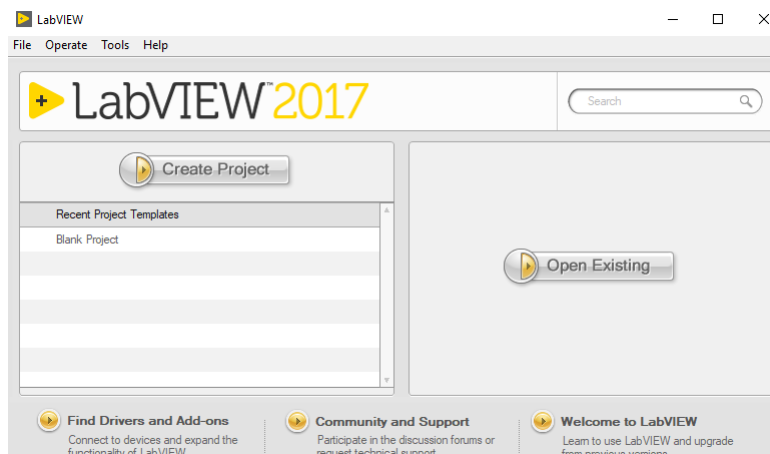
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

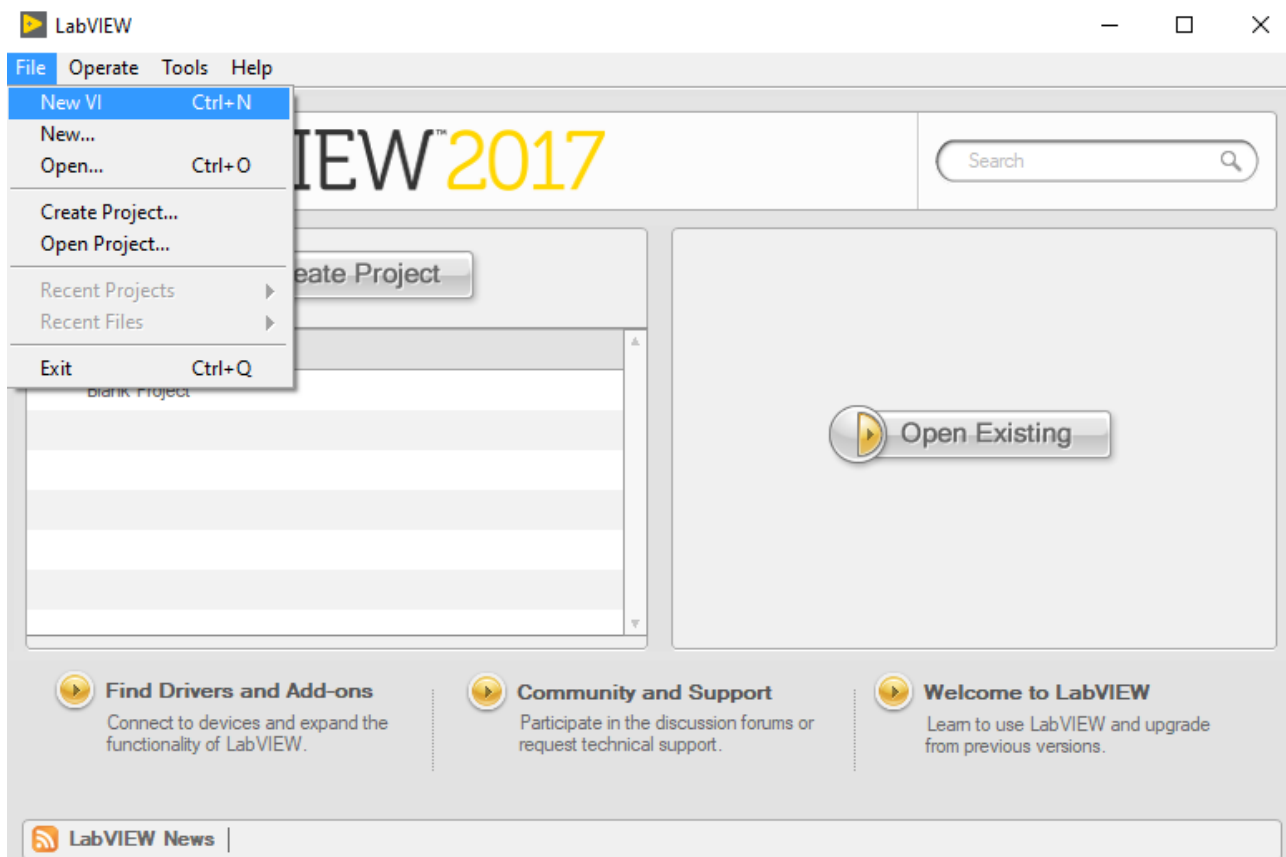


Introducción al Entorno de LabVIEW:

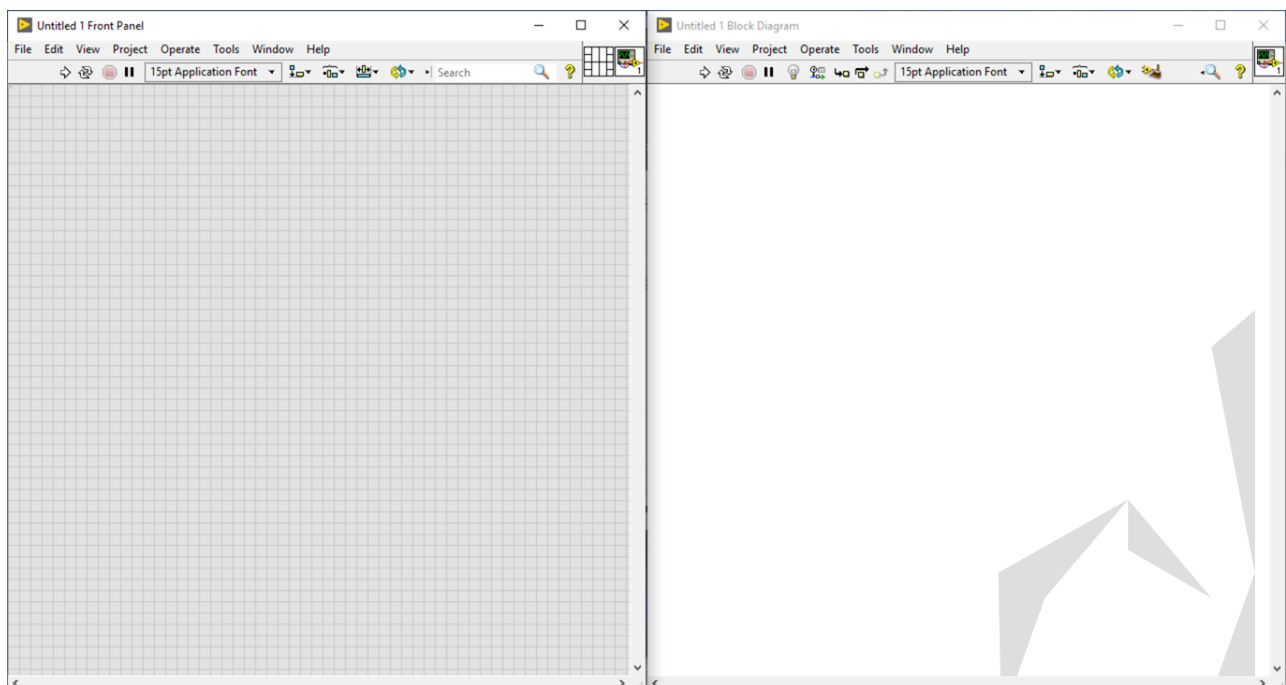
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

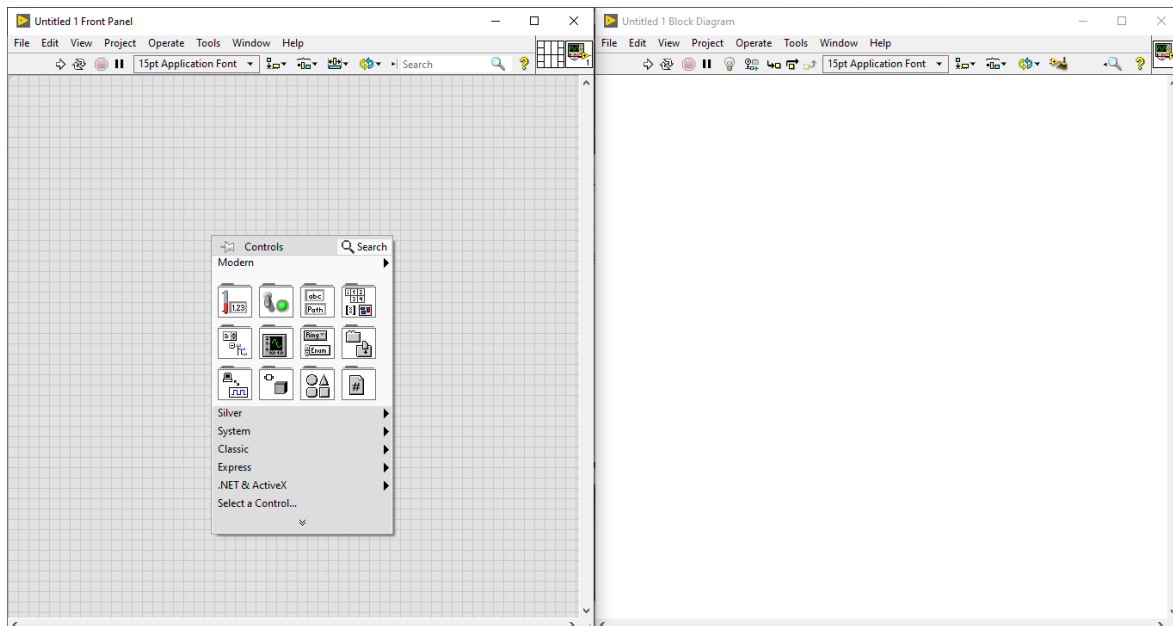


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



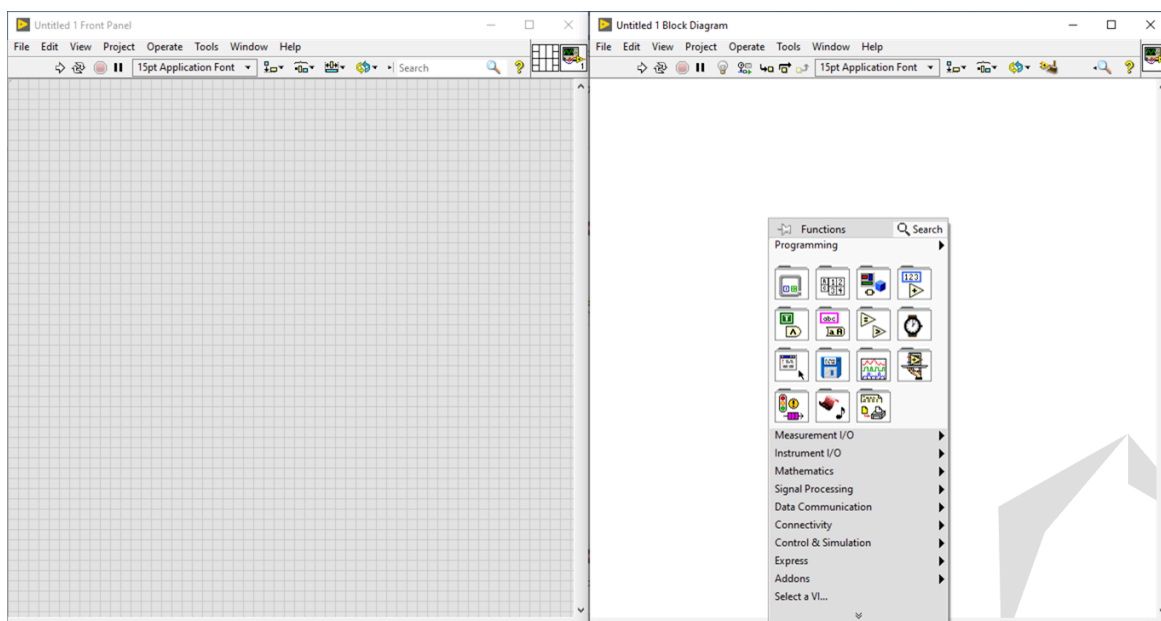
Front Panel: Ventana Gris con la Interfaz del Programa

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.



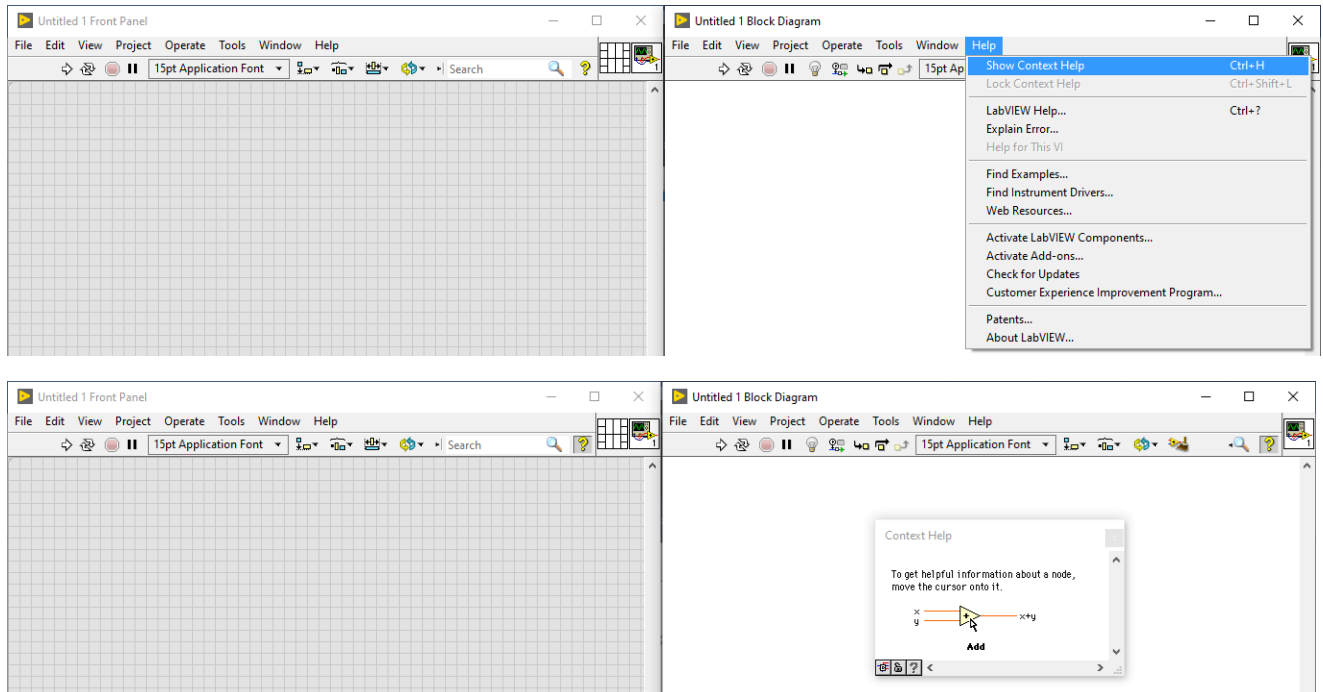
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

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.



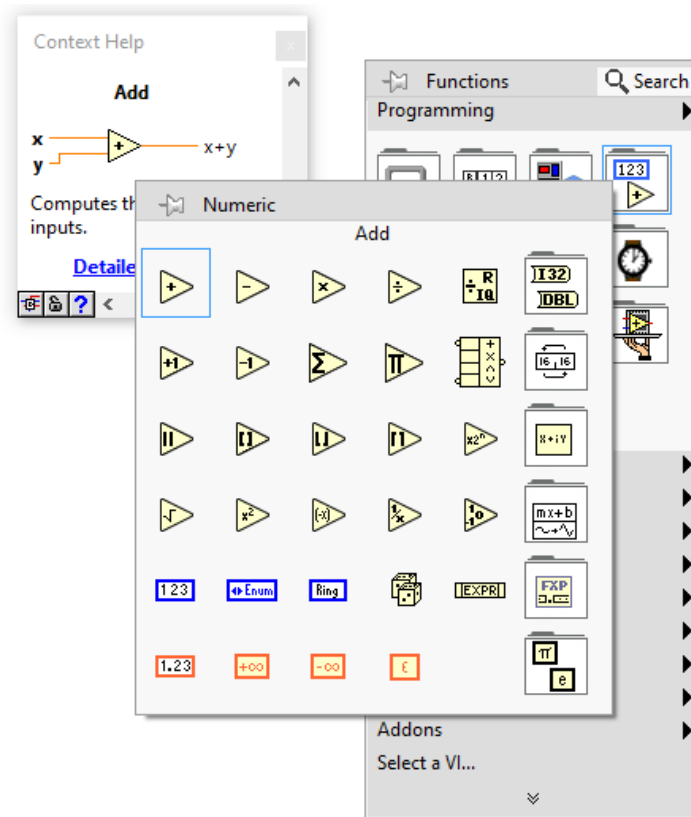
Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:

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.

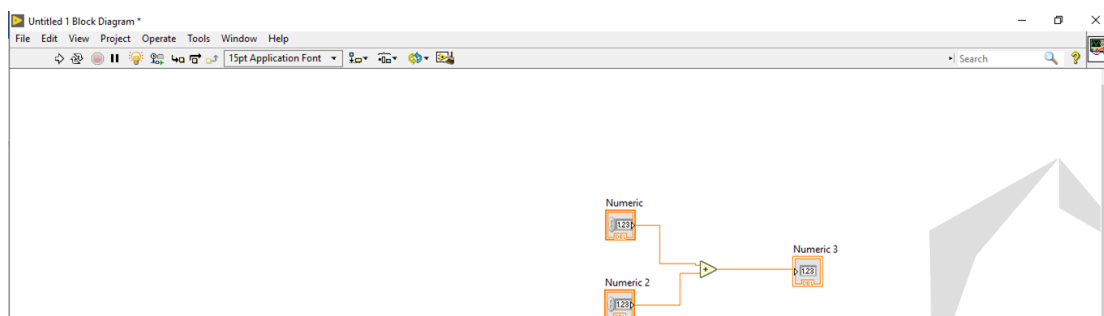


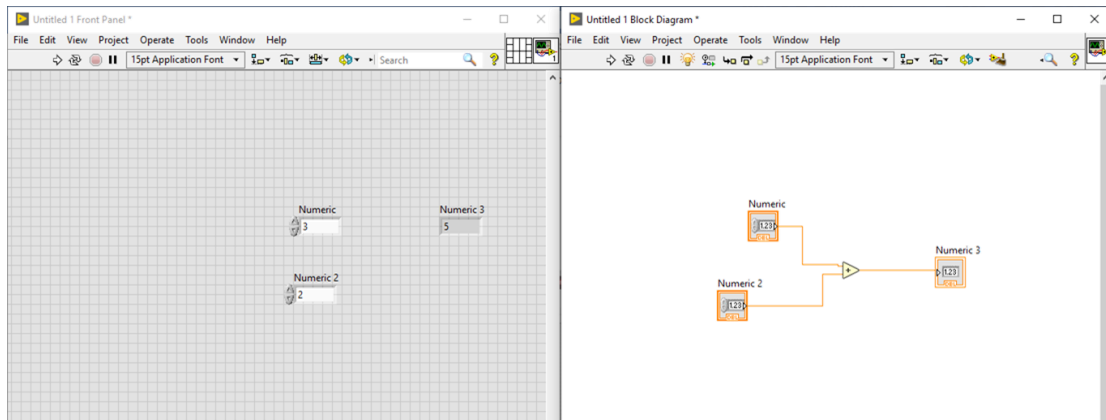
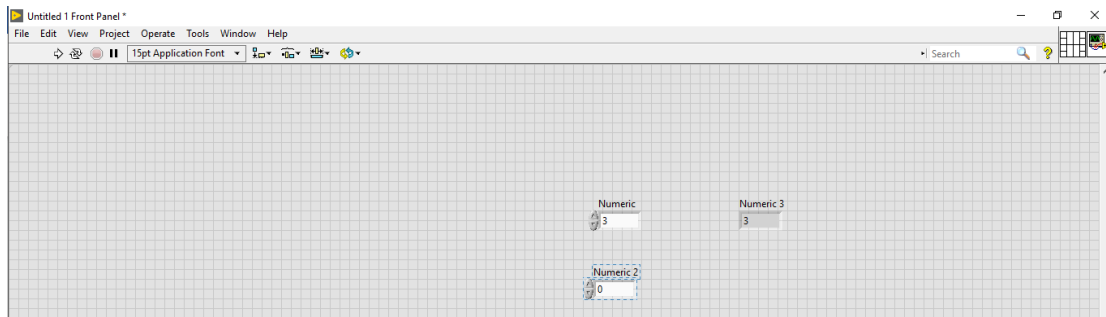
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.



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

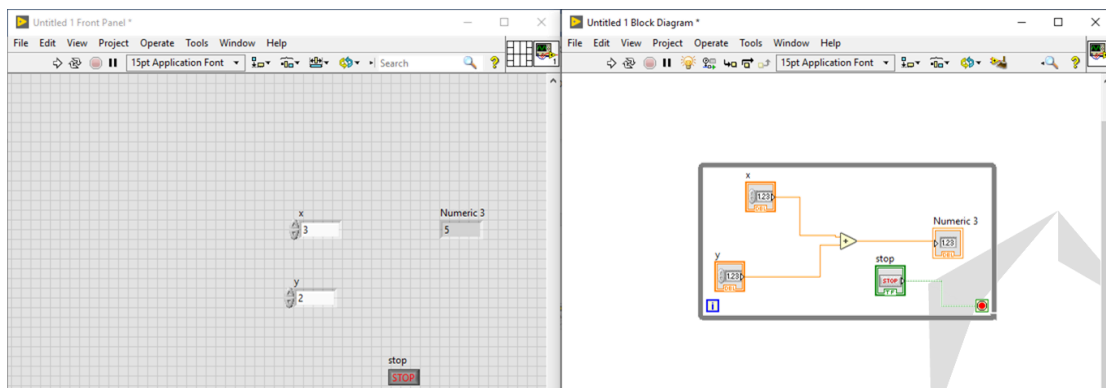
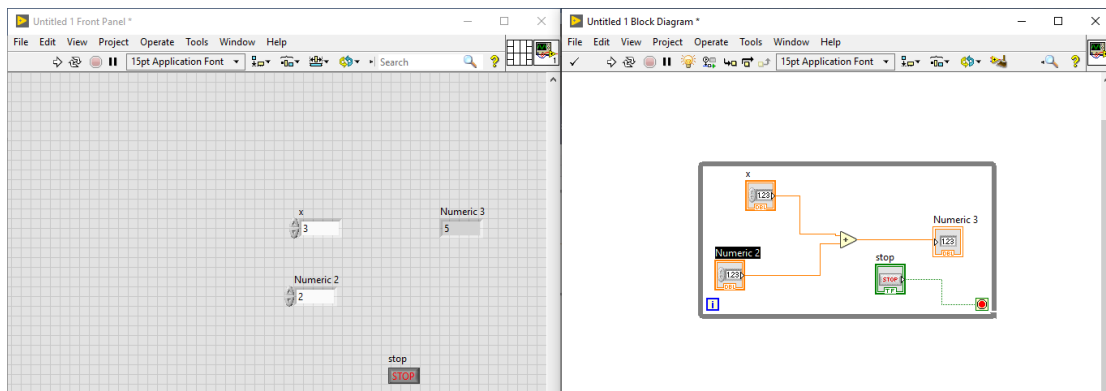
Al dar doble clic en el bloque de la pantalla blanca, me llevará al punto donde se encuentra el mismo bloque, pero en la pantalla gris.

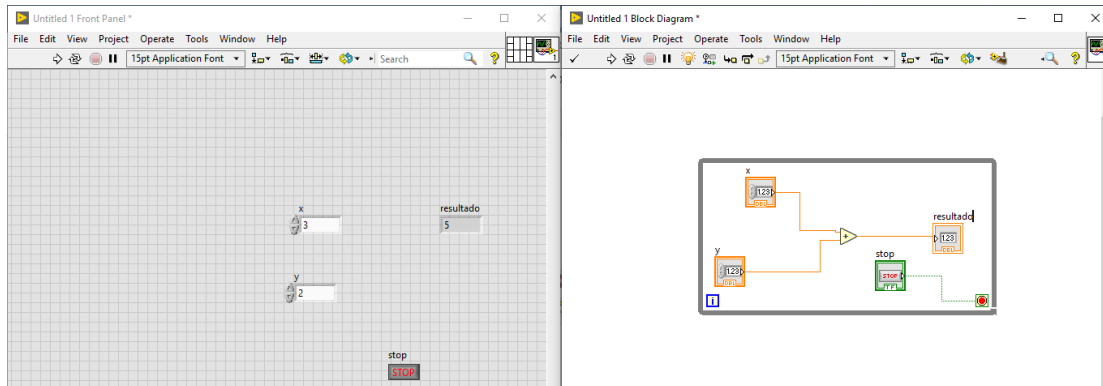




Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

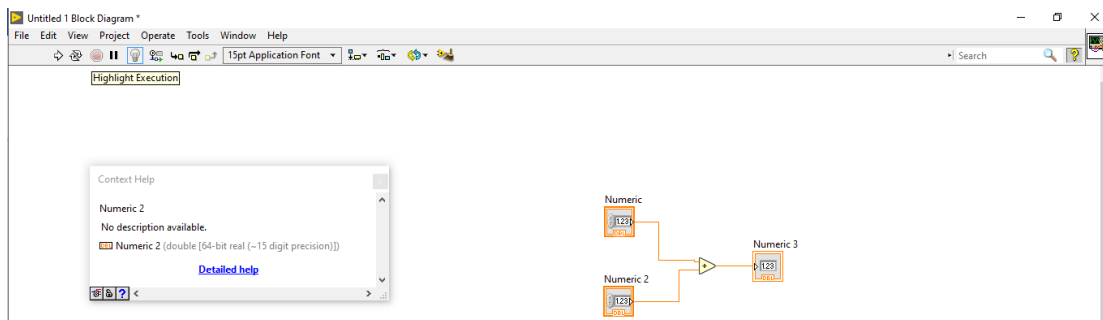
El nombre de los elementos de las interfaces se puede cambiar desde el Block Diagram, cambiándole literal el nombre a los bloques.





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.

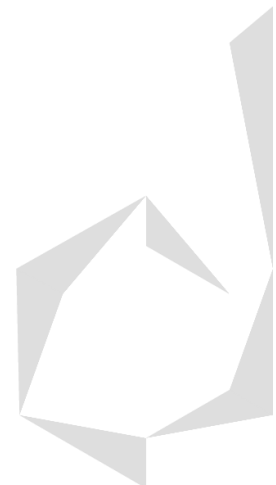


Coertion dot: Conversión Automática de Datos por Parte de LabVIEW

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: Momentos Invariantes de Hu

Introducción Teórica – Momentos Invariantes de Hu

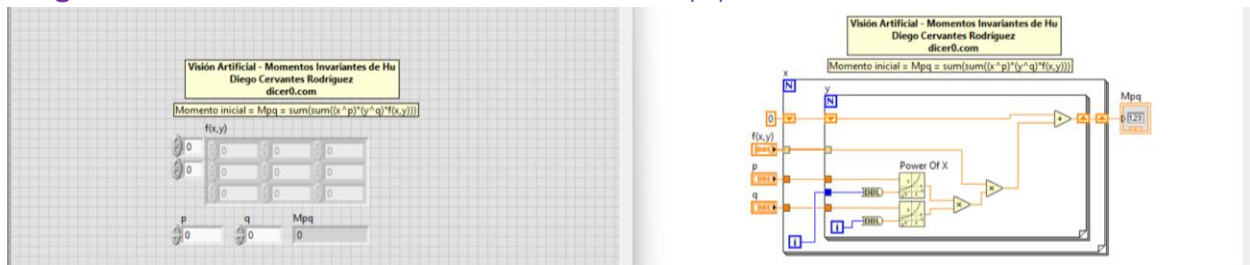
Los momentos invariantes de Hu son un conjunto de ecuaciones numéricas que se usan en el análisis de imágenes para reconocer figuras, son conocidos por su invariancia a ciertas transformaciones geométricas, como rotación, escala y traslación. De esta manera, aunque una misma figura aparezca de forma diferente, será reconocida.

Existen siete momentos invariantes de Hu, etiquetados como h1, h2, h3, h4, h5, h6 y h7. Cada uno de ellos proporciona información sobre una característica particular de la figura y se calculan a partir de los momentos centrales de una imagen y se normalizan para lograr la invariancia, sus ecuaciones son las siguientes:

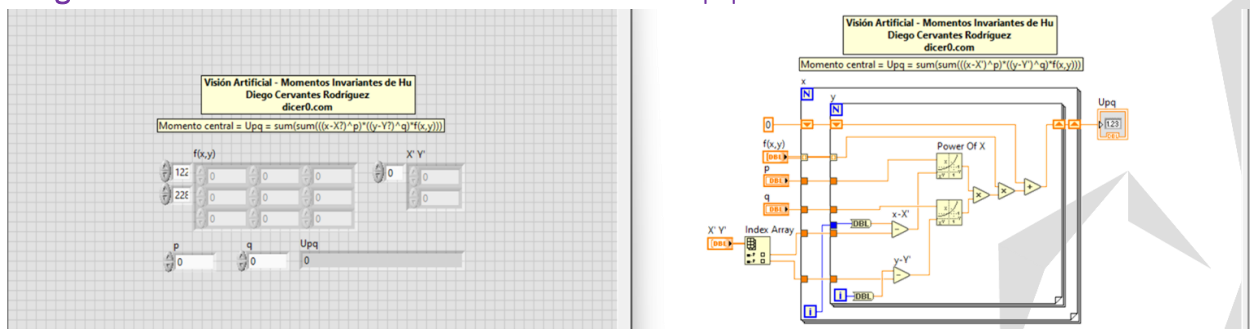
- $h1 = n20 + n02$
- $h2 = (n20 - n02)^2 + (2 * n11)^2$
- $h3 = (n30 - 3 * n12)^2 + (3 * n21 - n03)^2$
- $h4 = (n30 + n12)^2 + (n21 + n03)^2$
- $h5 = ((n30 - 3 * n12) * (n30 + n12) * (n30 + n12)^2 - 3 * (n21 + n03)^2) + ((3 * n21 - n03) * (n21 + n03) * (3 * (n30 + n12)^2 - (n21 + n03)^2))$
- $h6 = ((n20 - n02) * ((n30 + n12)^2 - (n21 + n03)^2) + ((4 * n11) * (n30 + n12) * (n21 + n03)))$
- $h7 = ((3 * n21 - n03) * (n30 + n12) * ((n30 + n12)^2 - 3 * (n21 + n03)^2) + ((n30 - 3 * n12) * (n21 + n03) * (3 * (n30 + n12)^2 - (n21 + n03)^2)))$
- $\text{Momento inicial} = M_{pq} = \sum \sum (x^p * (y^q) * f(x, y))$
- $\text{Momento central} = U_{pq} = \sum \sum (x - X)^p * (y - Y)^q * f(x, y)$
- $\text{Momento normalizado} = N_{pq} = \frac{U_{pq}}{u00^q}$, donde; $d = \frac{(p+q)}{2} + 1$, si $p + q > 1$; $d = 1$, si $p + q \leq 1$

Estos momentos se utilizan ampliamente en reconocimiento de patrones, visión por computadora y aplicaciones de análisis de imágenes. Son especialmente útiles cuando se requiere reconocimiento de formas robusto y se necesitan características que no se vean afectadas por cambios geométricos en la imagen.

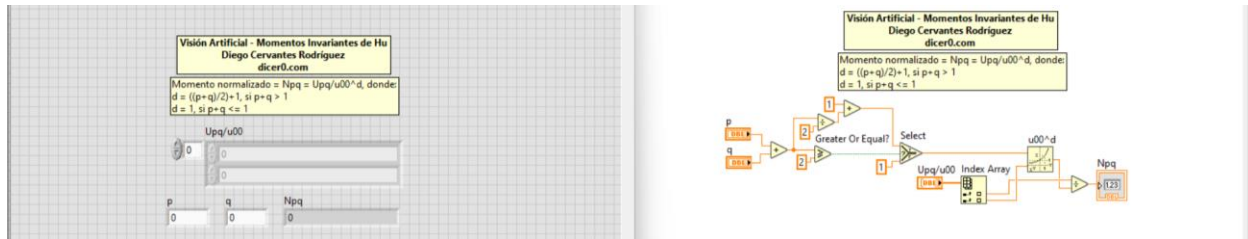
Programa: Momento Invariante de Hu Inicial – M_{pq}



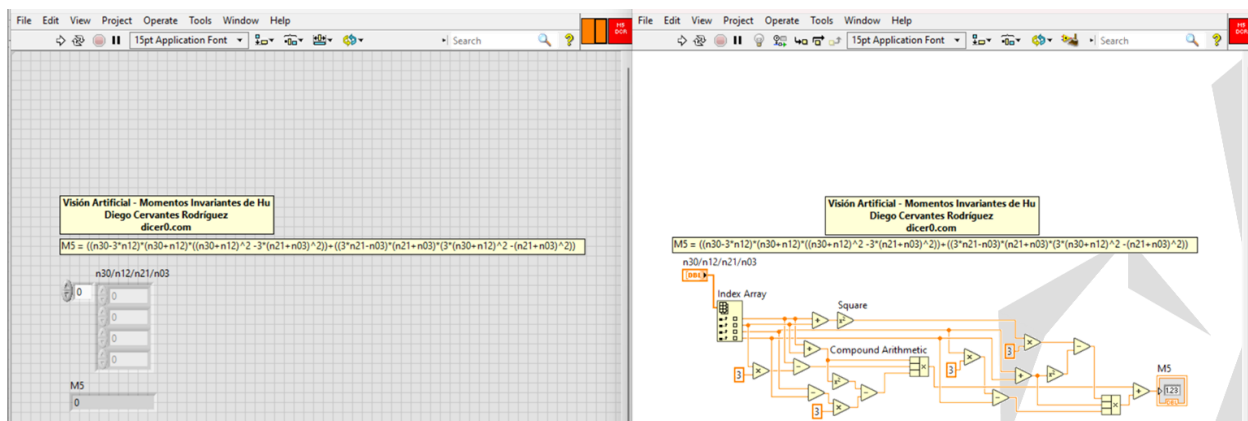
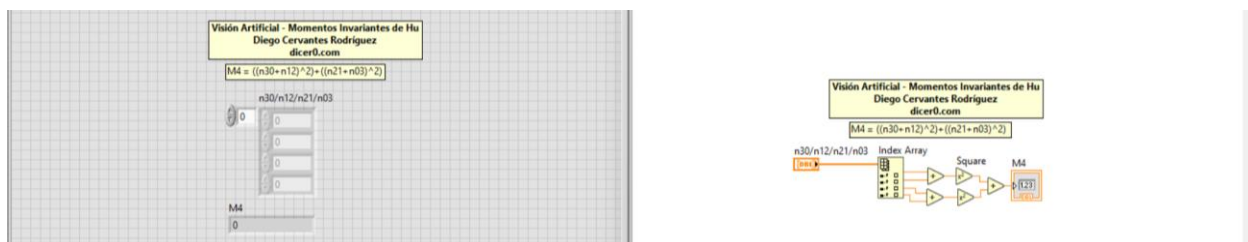
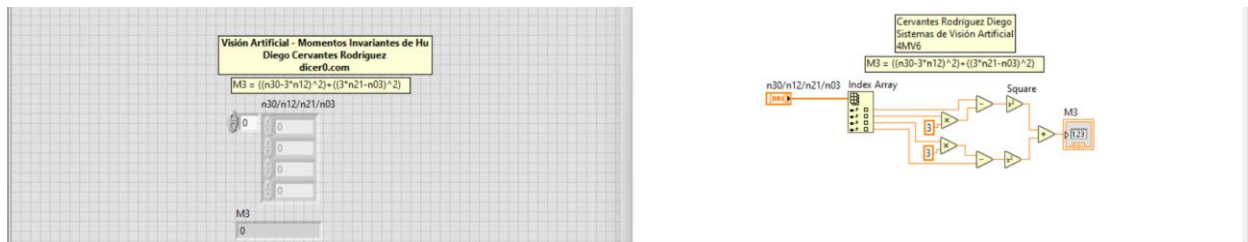
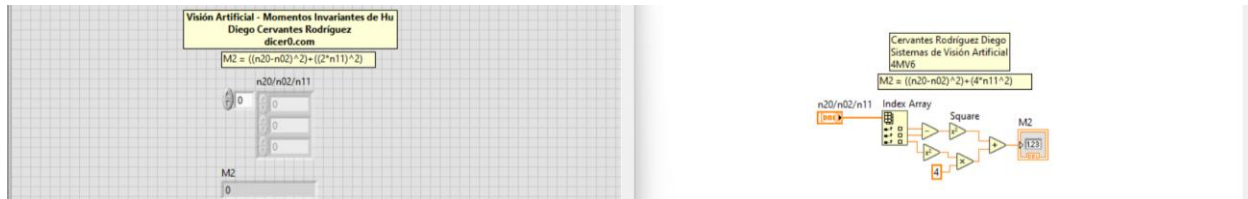
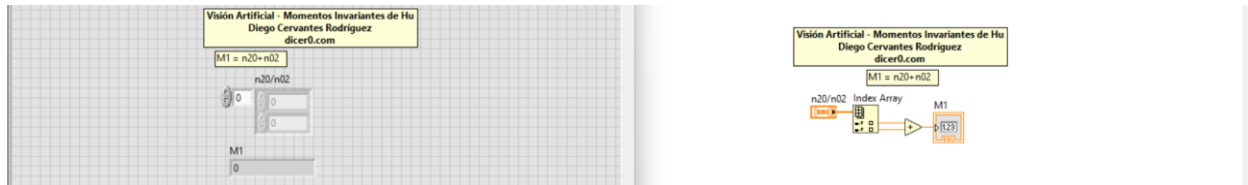
Programa: Momento Invariante de Hu Central – U_{pq}

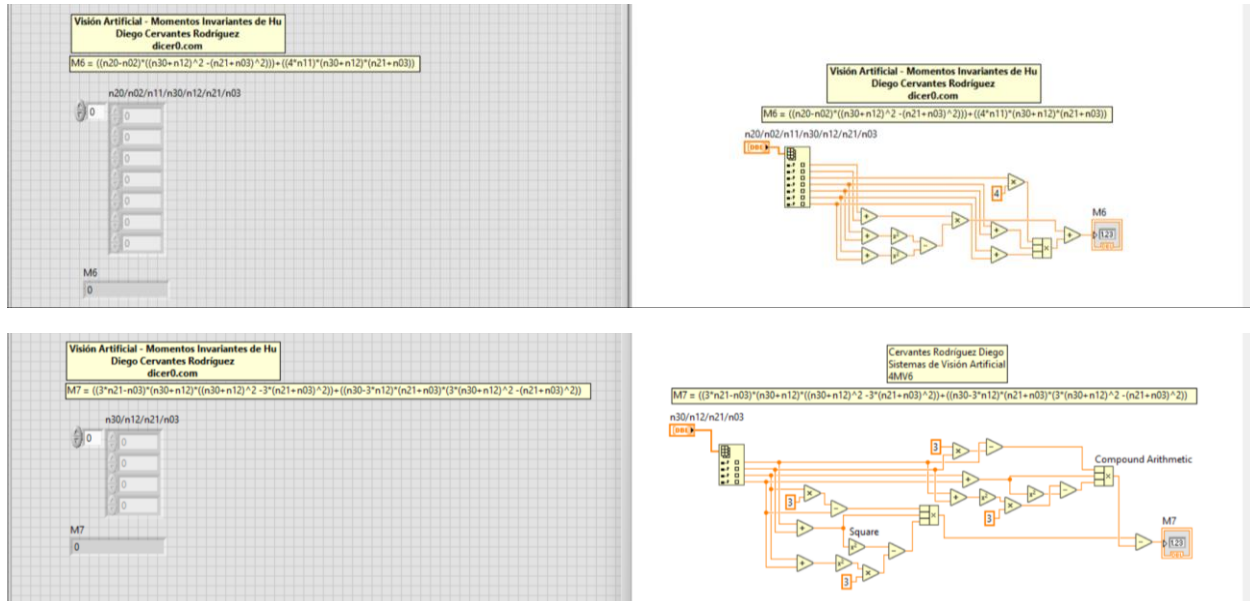


Programa: Momento Invariante de Hu Normalizado – Npq



Programa: Momentos Invariante de Hu – h1, h2, h3, h4, h5, h6 y h7





Programa: Momentos Invariantes de Hu – Reconocimiento Patrones

