INGENIERÍA MECATRÓNICA

DI_CERO

DIEGO CERVANTES RODRÍGUEZ

INSTRUMENTACIÓN VIRTUAL

NI LABVIEW 2020 (32-BIT)

Visión Artificial: Detección de Bordes y Suavizado

Contenido

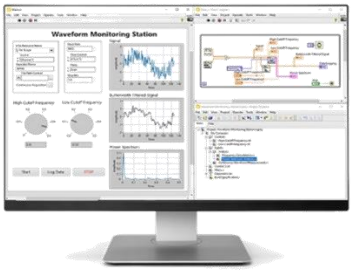
Introducción Teórica de LabVIEW:

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.

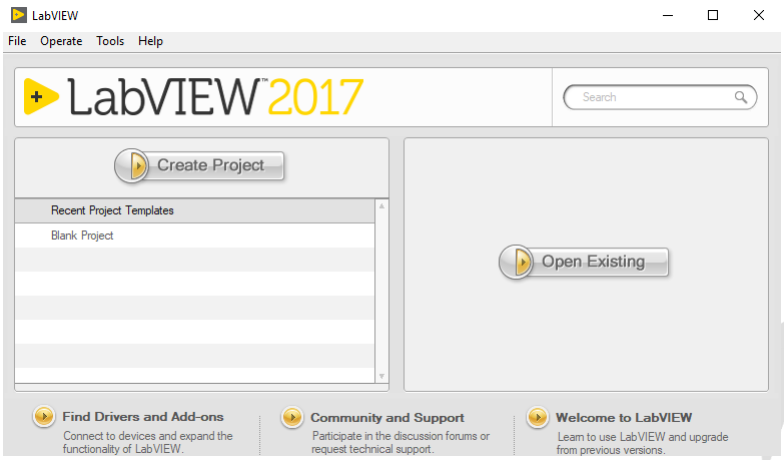
- **Instrumentación Tradicional:** El hardware es más usado, como por ejemplo con los circuitos integrados de un osciloscopio.
- **Instrumentación Virtual:** El software es el más utilizado y sus funciones son modulares, como lo es en una tarjeta de desarrollo de National Instruments.

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

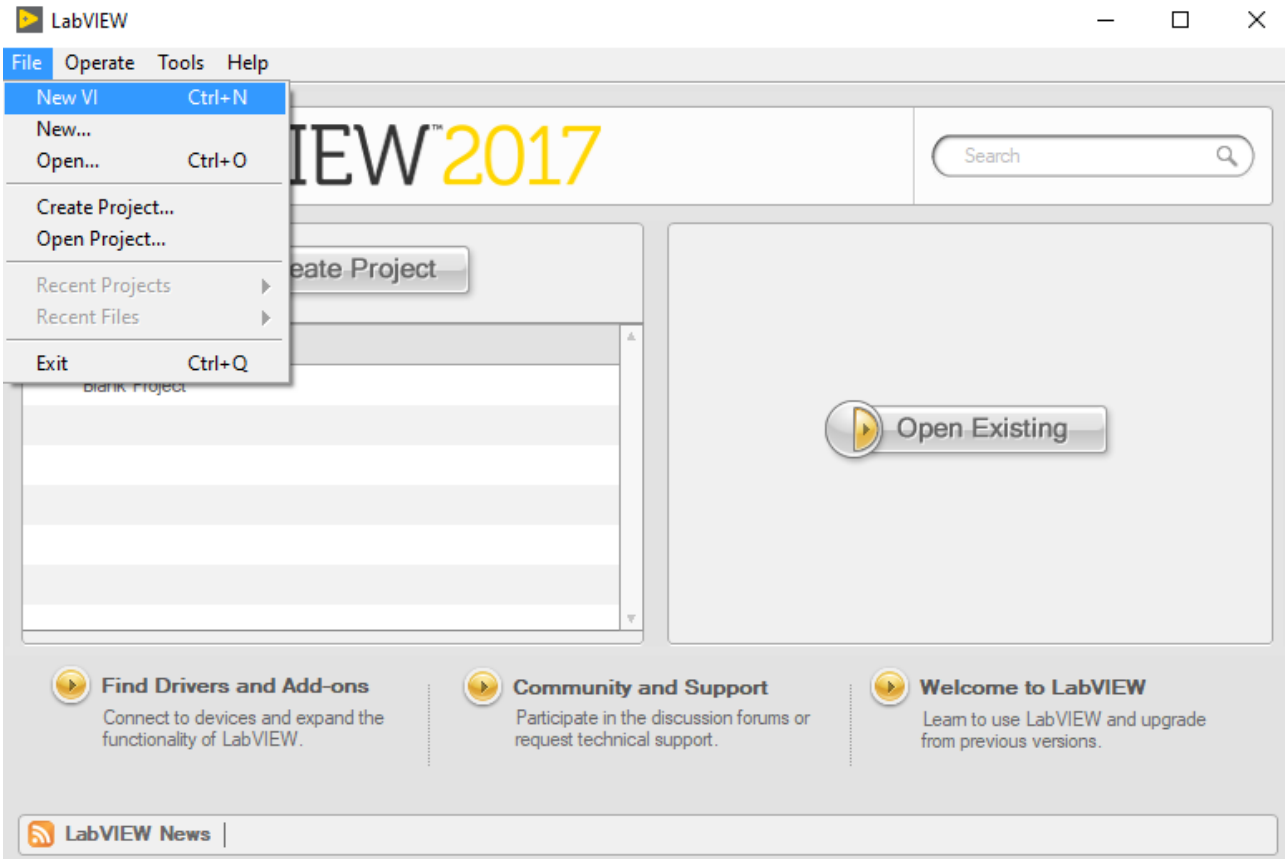


Introducción al Entorno de LabVIEW:

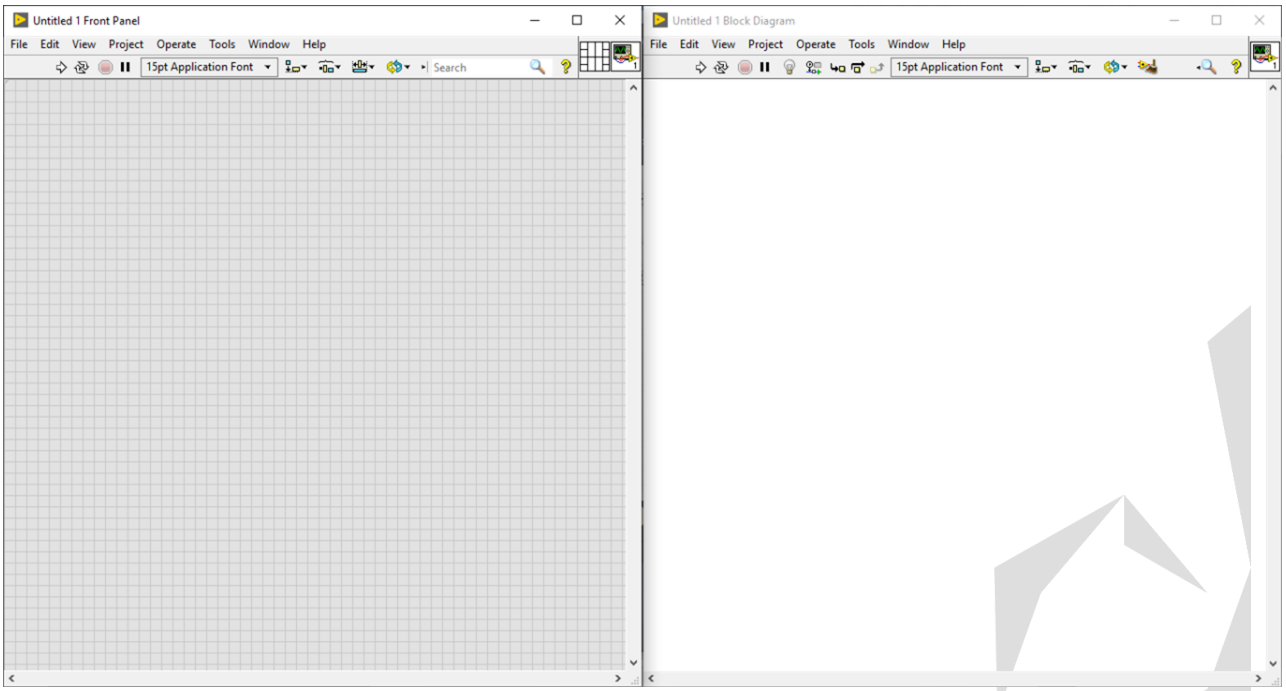
Un nuevo proyecto de LabView se abre por medio del botón de Create project que aparece inmediatamente cuando abra el programa.



VI se refiere a Virtual Instrument.

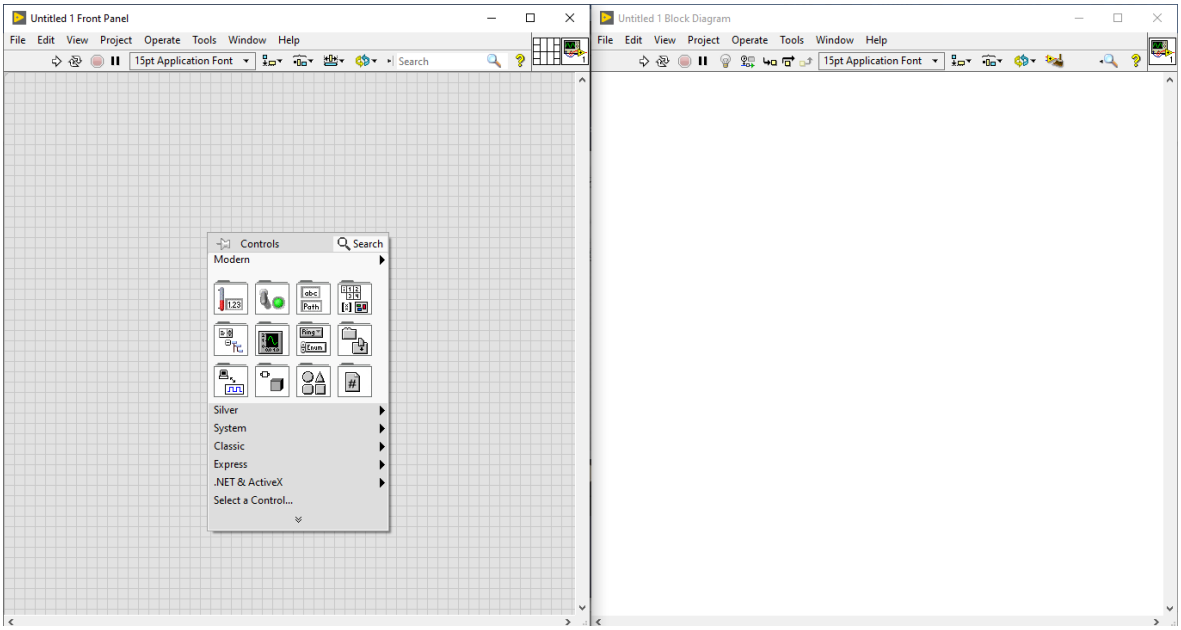


Al hacerlo me abrirá estas dos ventanas, en una de ellas se creará el programa con bloques (Ventana Block Diagram) y en la otra se verá la interfaz (Ventana Front Panel).



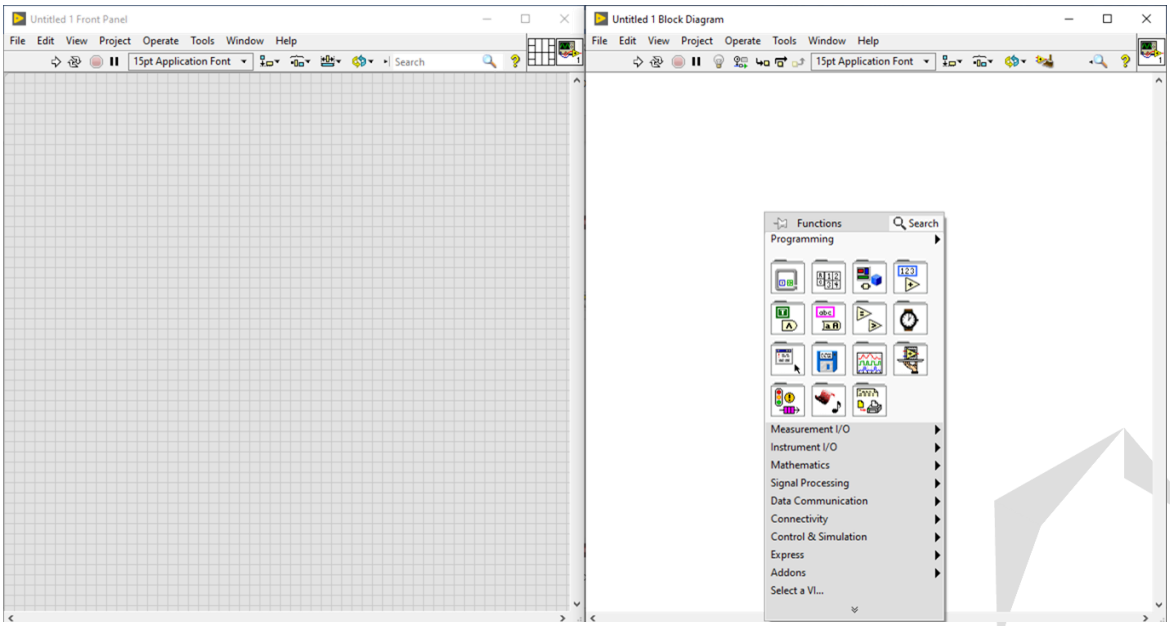
Front Panel: Ventana Gris con la Interfaz del Programa

En la ventana gris llamada *Front Panel*, es donde se observa la interfaz del Programa y se cuenta con el control pallete 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.*



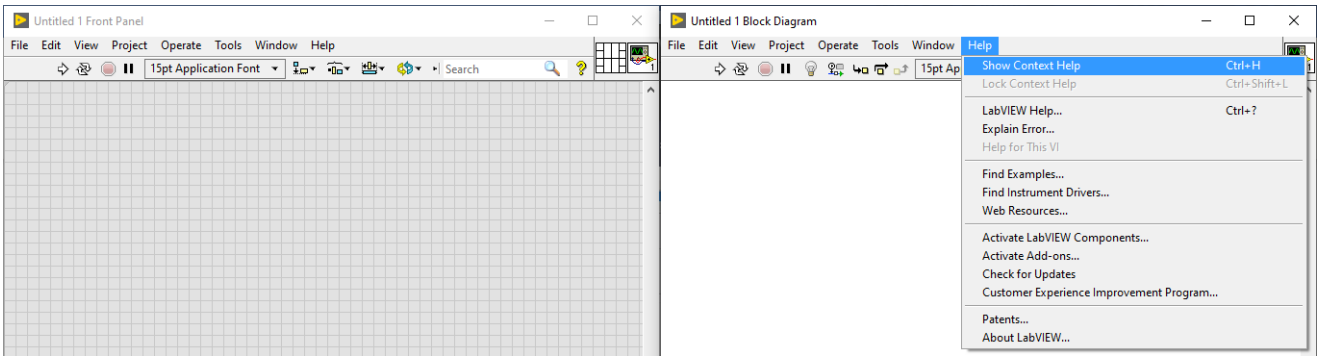
Block Diagram: Ventana Blanca con la Lógica del Programa (Bloques)

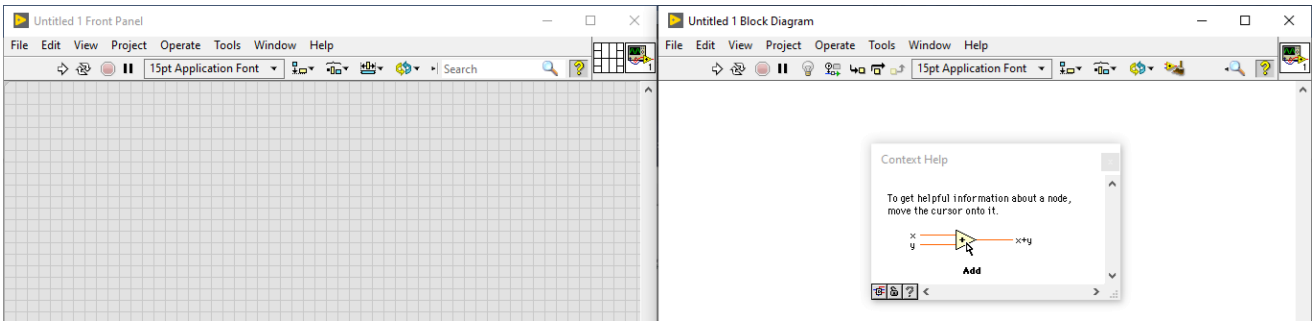
En la ventana blanca llamada *Block Diagram* aparece la paleta de funciones que sirve para introducir los elementos de programación en forma de bloques que se conectarán entre ellos y describirán la función del programa, aparece dando clic derecho en la pantalla gris. Si no aparece la ventana gris se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



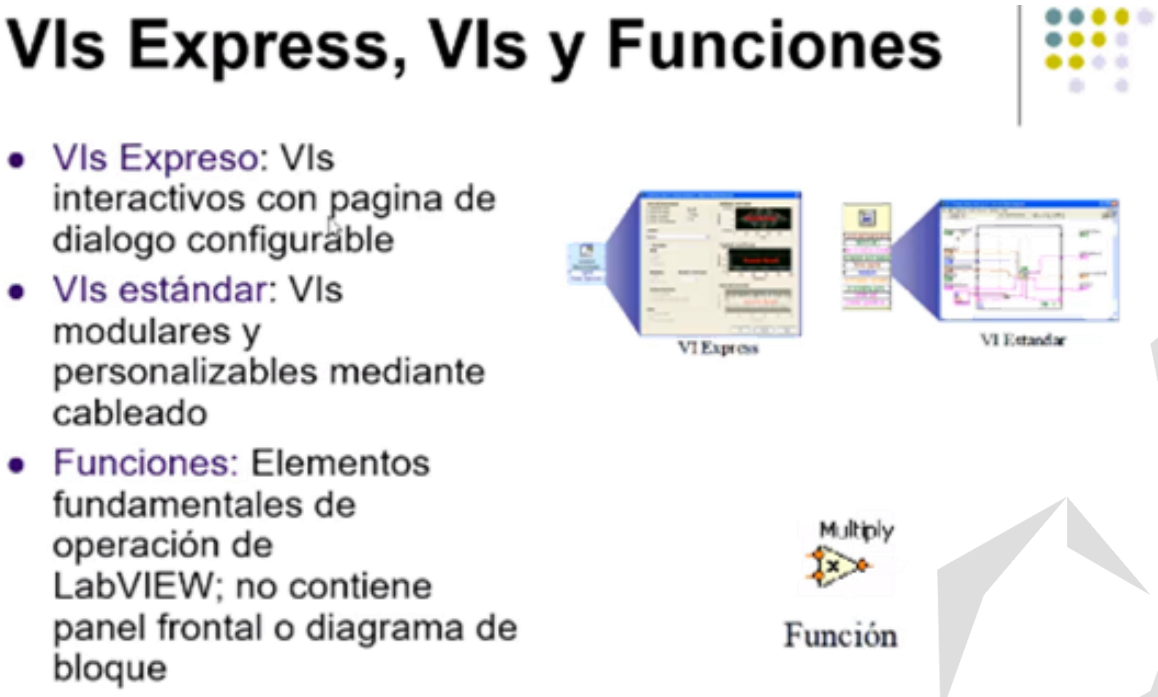
Show Context Help: Descripción de Bloques y sus Terminales

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.

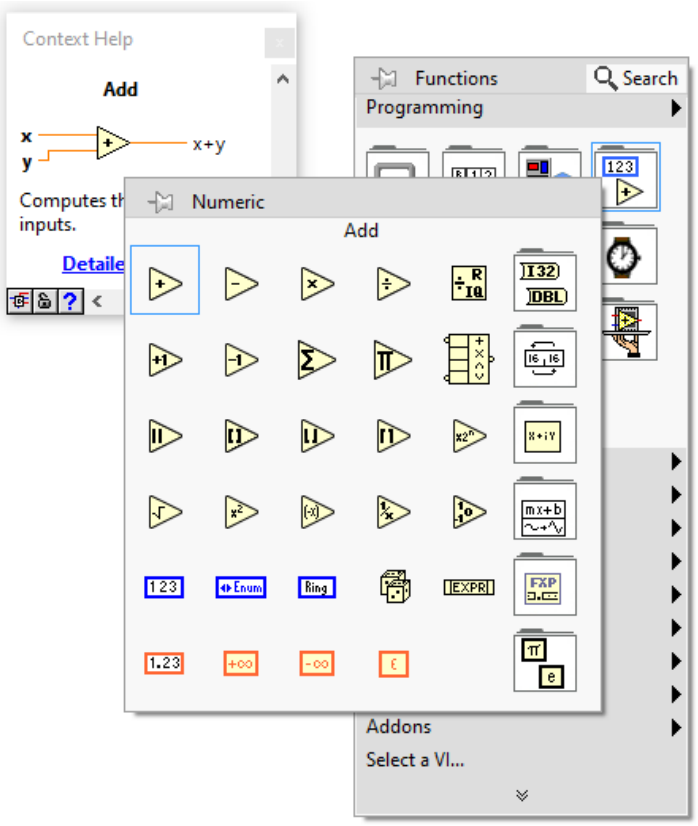




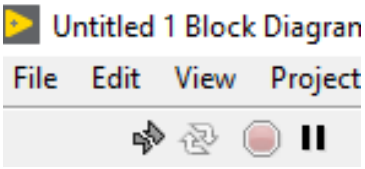
Las funciones o subrutinas son los elementos más básicos que pueden existir en LabView, dentro de ellas existe un código de bloque propio que describe sus funciones, pero además se cuenta con otros elementos:



En un bloque de código, l**as 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.

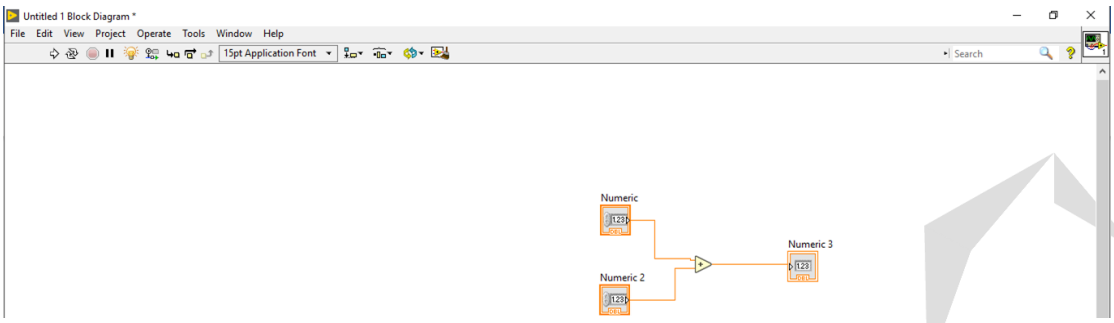


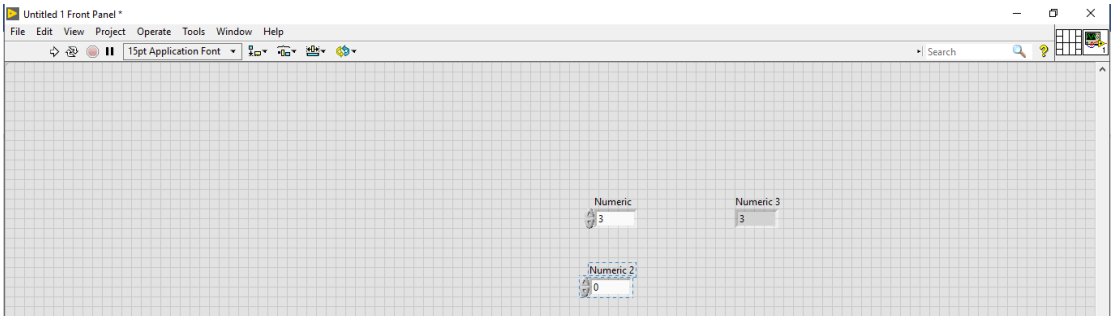
El programa es autocompilable, es decir que se corre por sí solo, por lo que si la flechita aparece rota es porque hay un error en el programa.

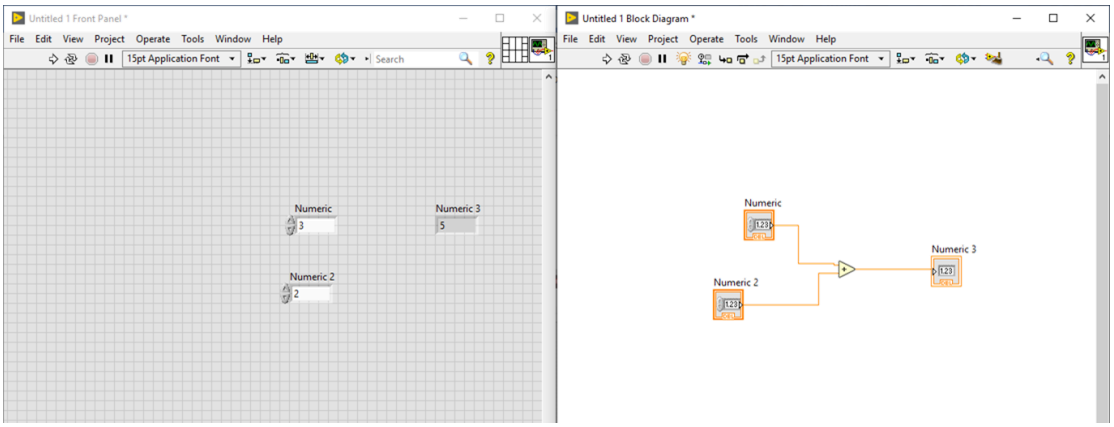


Front Panel y Block Diagram: Navegar de una Ventana a Otra

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.

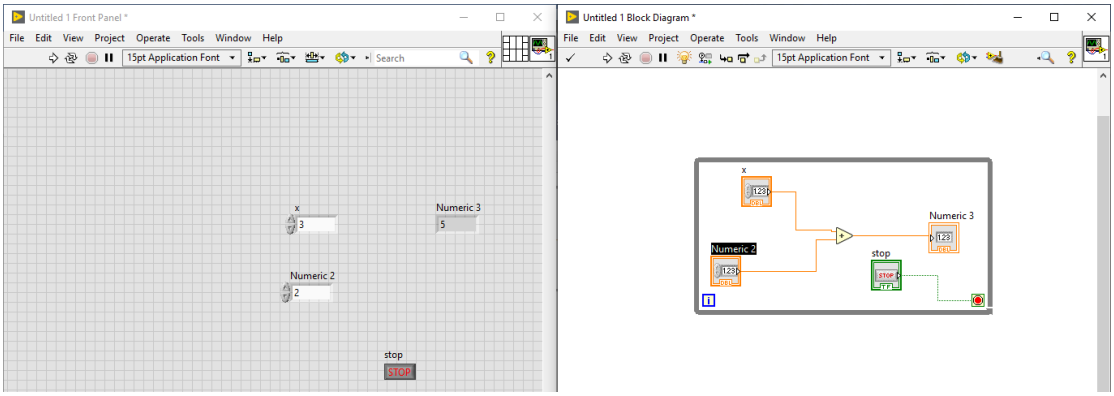


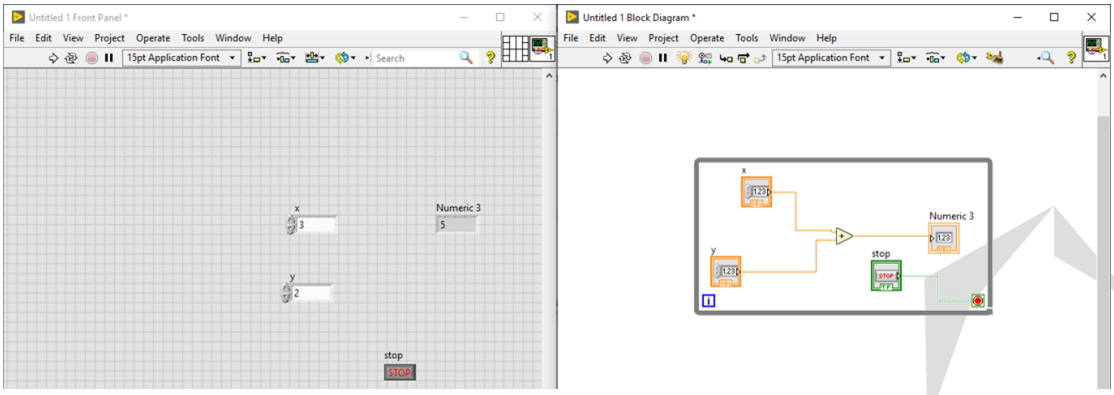


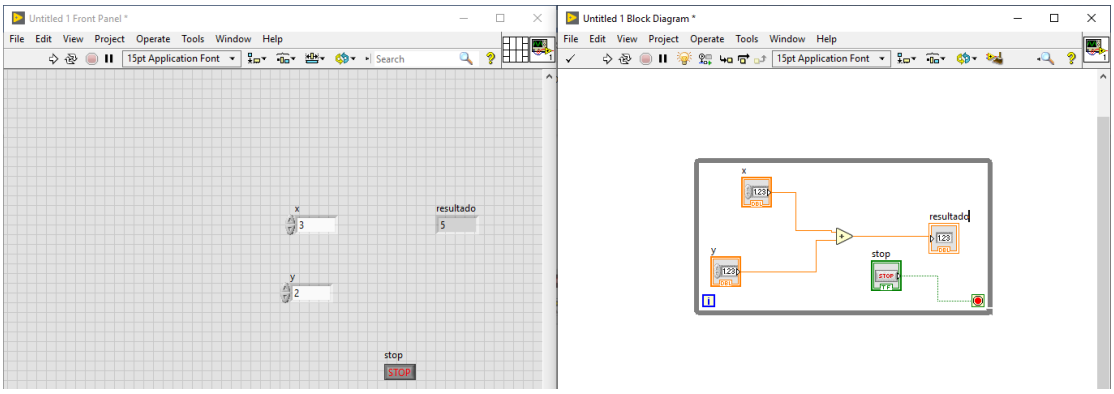


Block Diagram - Cambiar Nombre a los Bloques: Nombre de los elementos en el Front Panel

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

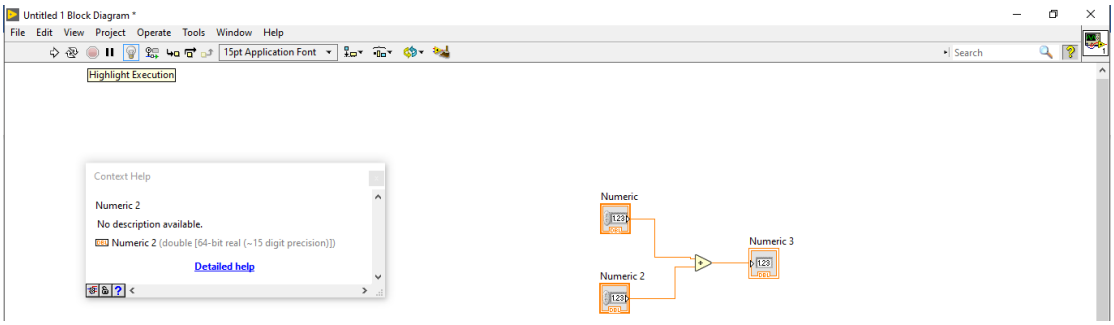






Block Diagram - Highlight Execution: Correr Más Lento el Programa

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 coertion 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 coertion 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: Detección de Bordes y Suavizado

Introducción Teórica – Detección de Bordes con los Teoremas de Sobel y Prewitt

La detección de bordes es un paso fundamental en el procesamiento de imágenes para identificar las transiciones abruptas de intensidad en el color o luz que corresponden a los límites entre objetos en una imagen, esto es muy útil hacerlo ya que de esta manera al binarizar una imagen se puede separar de manera más optima la figura deseada de su contorno. Los teoremas de Sobel y Prewitt son dos técnicas clásicas para la detección de bordes en imágenes.

Los teoremas de Sobel y Prewitt son dos operadores de convolución distintos, utilizados para calcular los gradientes de intensidad en los píxeles en una imagen, en términos más simples, sirve para derivar la matriz de la imagen respecto a la dirección horizontal (Gx), vertical (Gy) o la combinación de ambas (G), obteniendo de esta manera los puntos máximos y mínimos de la matriz de tres dimensiones (capas RGB) que describen la imagen digital. Estos gradientes representan la magnitud y dirección del cambio de intensidad en cada punto de la imagen.

Ambos teoremas utilizan máscaras de convolución que se deslizan sobre los pixeles y vecindades de la imagen para calcular las gradientes Gx, Gy y G. Estas máscaras son matrices pequeñas de 3X3 que determinan cómo se combinan los valores de los píxeles vecinos.

- **Teorema de Sobel:** Sus máscaras están diseñadas para identificar bordes (cambios de intensidad de color o luz) en las dos direcciones principales “x” y “y”.

$$Gx_{Sobel} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Gy_{Sobel} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$Gxy_{Sobel} = \sqrt{Gx_{Sobel}{}^2 + Gy_{Sobel}{}^2}$$

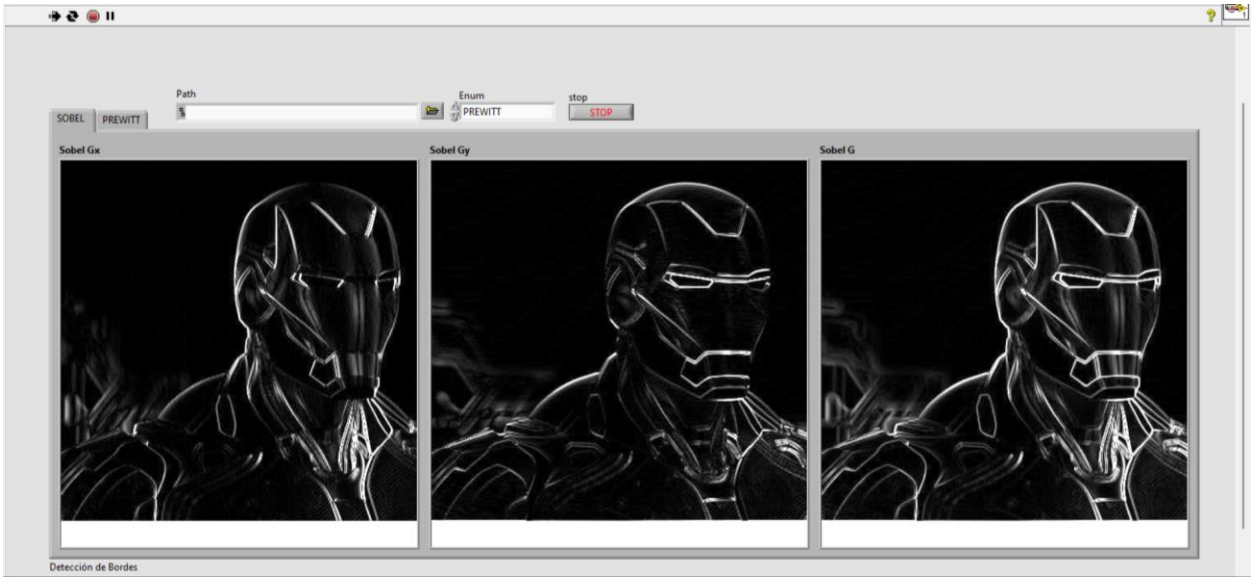
- **Teorema de Prewitt:** Es similar al teorema de Sobel, sin embargo, las máscaras de Prewitt están diseñadas para enfatizar más los cambios de intensidad diagonales además de los cambios horizontales y verticales. Esto permite una detección de bordes más completa en múltiples direcciones.

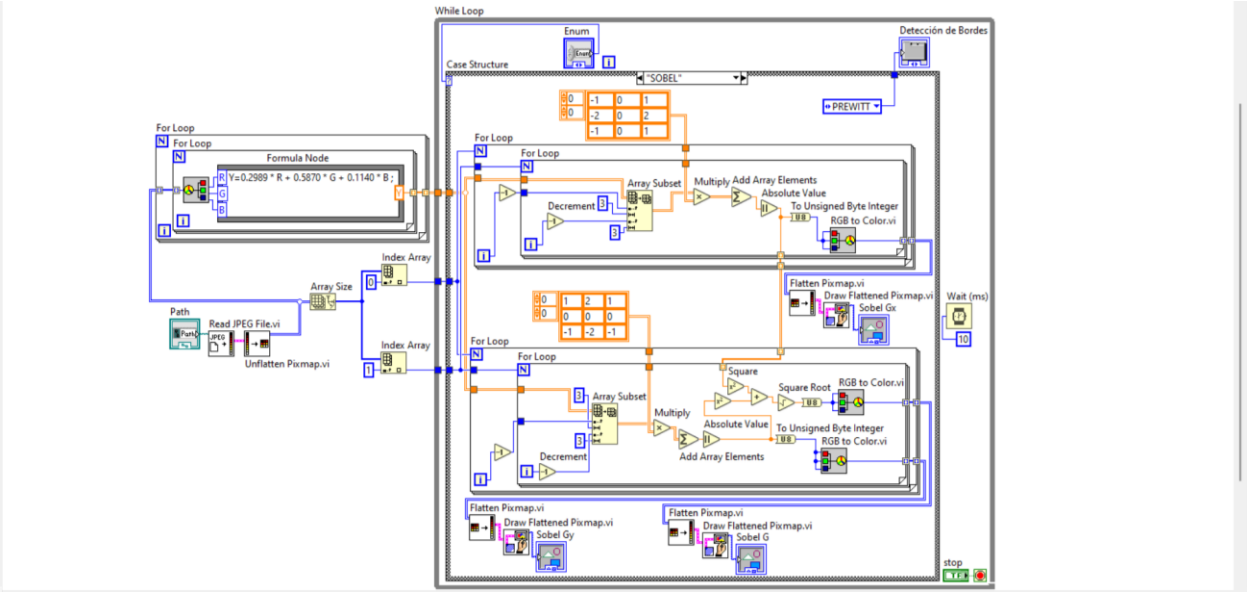
$$Gx_{Prewitt} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

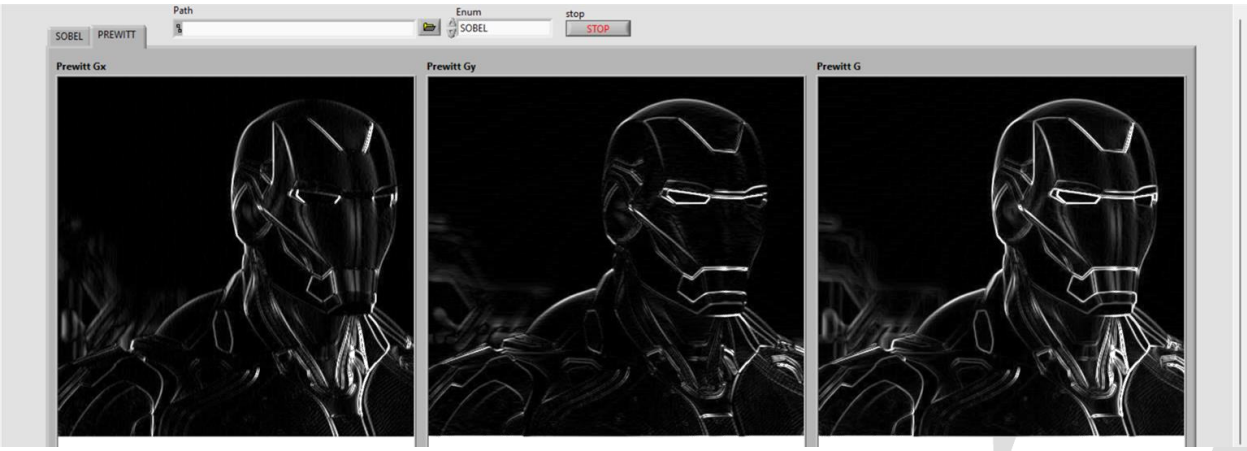
$$Gy_{Prewitt} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

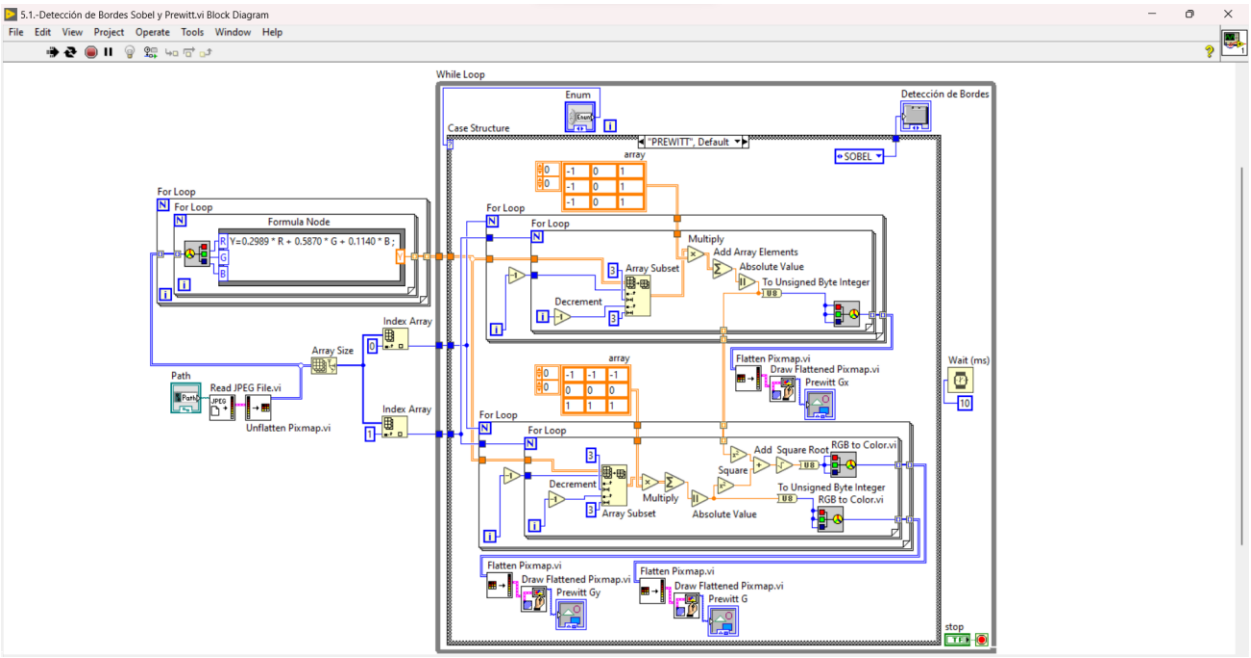
$$Gxy_{Sobel} = \sqrt{Gx_{Prewitt}{}^2 + Gy_{Prewitt}{}^2}$$

Programa: Detección de Bordes – Teoremas de Sobel y Prewitt









Introducción Teórica – Suavizado de Imagen con el Filtro Gaussiano

La campana de Gauss o también llamada distribución normal es **una representación gráfica que muestra una distribución de datos en torno a un valor central llamado media**, con la Campana de Gauss es posible establecer una serie de parámetros que ayudan a predecir y racionalizar lo que aparentemente son resultados aleatorios, obteniendo una versión más clara y visual de la distribución de un conjunto de números. Esta herramienta se utiliza para representar la dispersión de los datos y su tendencia, con el fin de detectar patrones o comportamientos en diferentes situaciones, por lo cual es muy utilizada en estadística, probabilidad, filtros, visión artificial, etc. En su expresión más sencilla:

$$f(x) = (a)e^{\left[\frac{-(x-b)^2}{2c^2}\right]}$$

- a = Representa el valor más alto (amplitud) de la campana de Gauss.
  - Mientras menor sea el valor de σ, mayor será la amplitud de la función.

$$a = \frac{1}{\sigma\sqrt{2\pi}}$$

- b = Es la posición central de la campana.
  - b = μ = m: Esta variable es llamada media.
    - La función es simétrica respecto a la media μ.
    - Su valor máximo se encuentra en la media μ.
- c = Controla el ancho de la campana de Gauss (desviación estándar).
  - $c^2 = \sigma^2 = s^2$: Esta variable es llamada varianza.
    - En las coordenadas de μ − σ y μ + σ se presentan los puntos de inflexión de la curva.

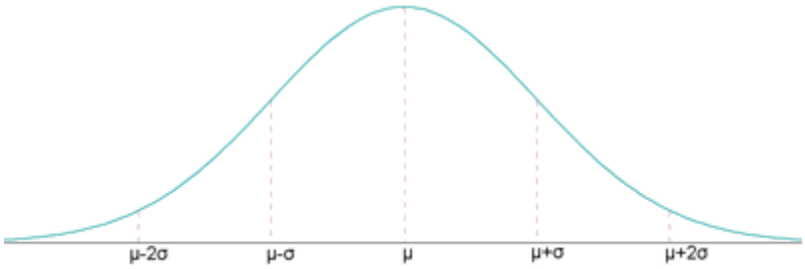
$$f(x) = (a)e^{\left[\frac{-(x-b)^2}{2c^2}\right]} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)e^{\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)e^{\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]}$$

- El 100% de la probabilidad equivale al área encerrada bajo la curva:

$$f(\mu - \sigma < x < \mu + \sigma) = 0.6826 \ = \ 68.26\,\%$$

$$f(\mu - 2\sigma < x < \mu + 2\sigma) = 0.954 \ = \ 95.4\,\%$$

$$f(\mu - 3\sigma < x < \mu + 3\sigma) = 0.997 \ = \ 99.7\,\%$$



La campana Gaussiana en visión artificial se utiliza para suavizar la imagen, mediante algo llamado el filtro gaussiano. **El suavizado de una imagen implica reducir el ruido y las pequeñas variaciones de intensidad en su color o luz**, obteniendo una versión más suave y menos abrupta de la imagen.

El filtro gaussiano se basa en la convolución de la imagen original con un kernel gaussiano (la convolución es una operación matemática que combina dos funciones para producir una tercera función que representa cómo una de las funciones "influye" en la otra), donde el kernel gaussiano tiene la forma misma forma que la campana gaussiana, al aplicar el filtro la media (μ) de la campana se centrará en el pixel central de una vecindad y la varianza (σ) afectará los demás pixeles de la matriz, suavizando de esta manera todos los pixeles de la imagen.

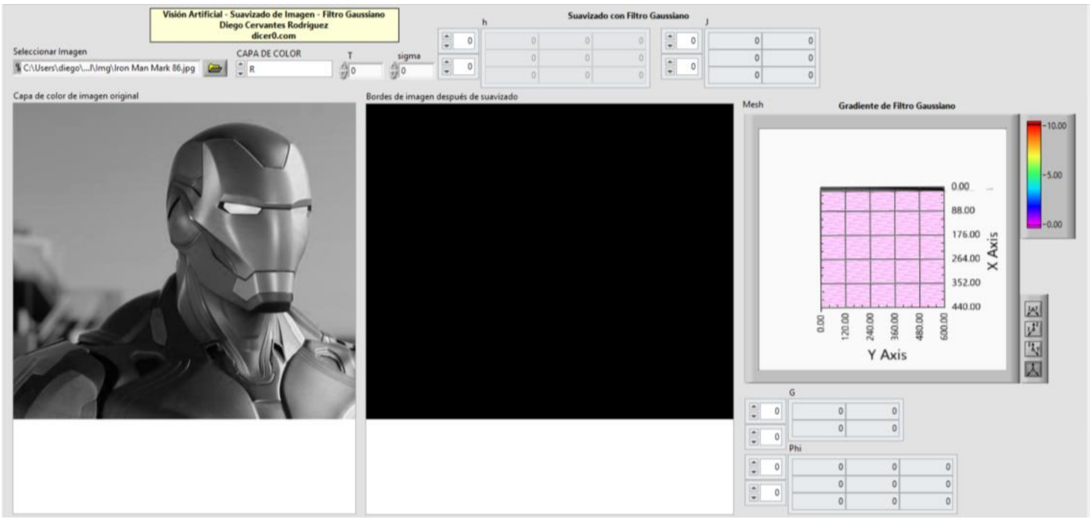
Recordemos que dijimos que el kernel del filtro Gaussiano no es lo mismo a la campana Gaussiana, esto se debe a que este tiene un tamaño matricial T que indica a cuáles pixeles de la vecindad se afectará a la vez, algunos tamaños de kernel comunes utilizados son:

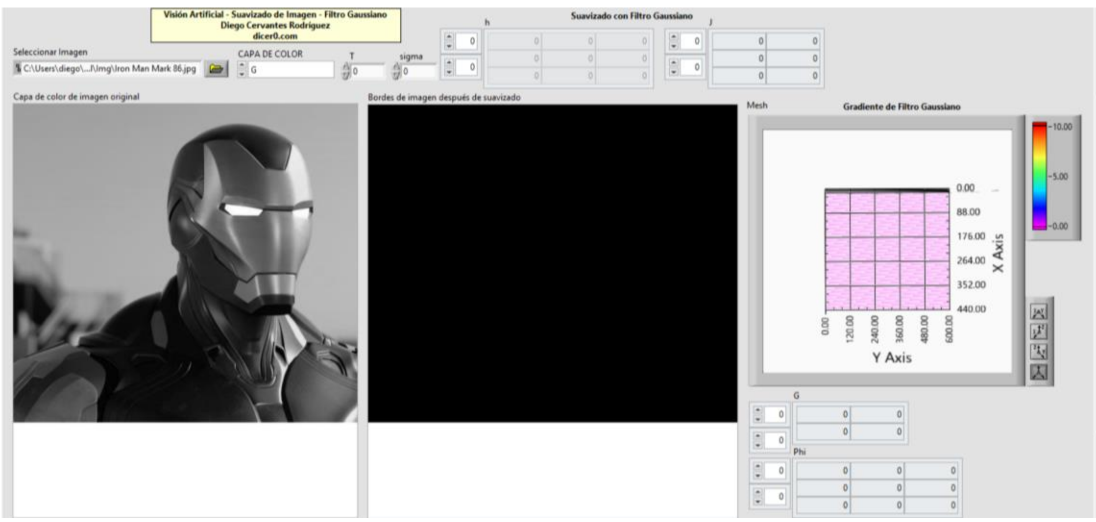
- **1 = 1x1:** En este caso, el kernel consiste en un solo elemento, lo cual representa un suavizado muy ligero y apenas afecta la imagen.
- **3 = 3x3:** Es uno de los tamaños más comunes y ampliamente utilizados. El kernel tiene una dimensión de 3x3 y se aplica una convolución en un vecindario de píxeles adyacentes de 3x3.
- **5 = 5x5:** El kernel tiene una dimensión de 5x5, lo que implica una convolución más amplia en comparación con el tamaño 3x3. Esto puede resultar en un suavizado más pronunciado y una mayor eliminación de detalles finos.
- **7 = 7x7:** El kernel tiene una dimensión de 7x7, lo que implica una convolución aún más amplia y un suavizado más significativo. Este tamaño de kernel se utiliza cuando se desea un suavizado más fuerte y se acepta una mayor pérdida de detalles.
- **9 = 9x9 o mayores:** Se utilizan cuando se necesita un suavizado muy fuerte y una reducción significativa de ruido en la imagen. Estos tamaños de kernel son adecuados para casos en los que se desea eliminar ruido o suavizar áreas grandes sin enfatizar demasiado los detalles finos.

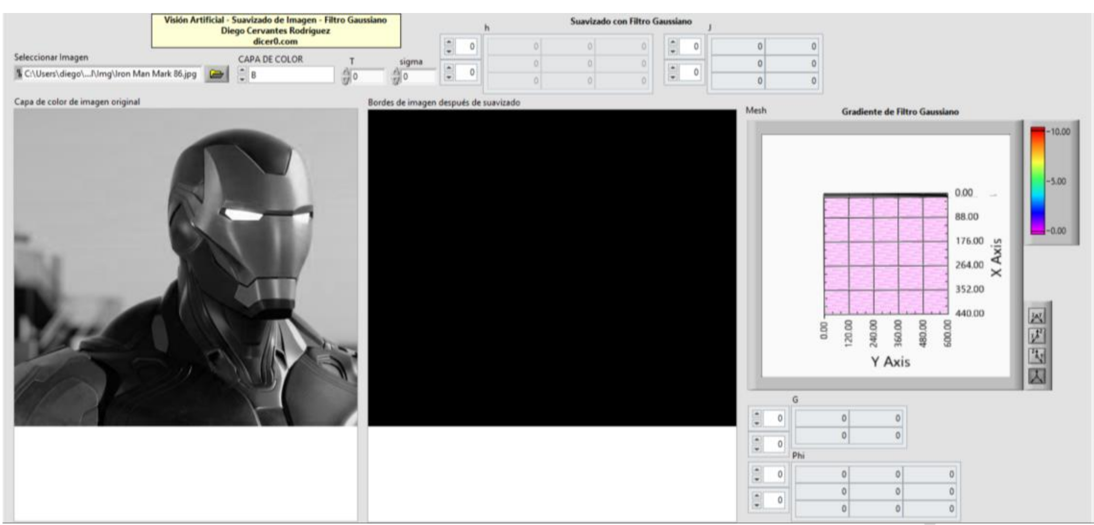
Programa: Suavizado de Imagen – Filtro Gaussiano

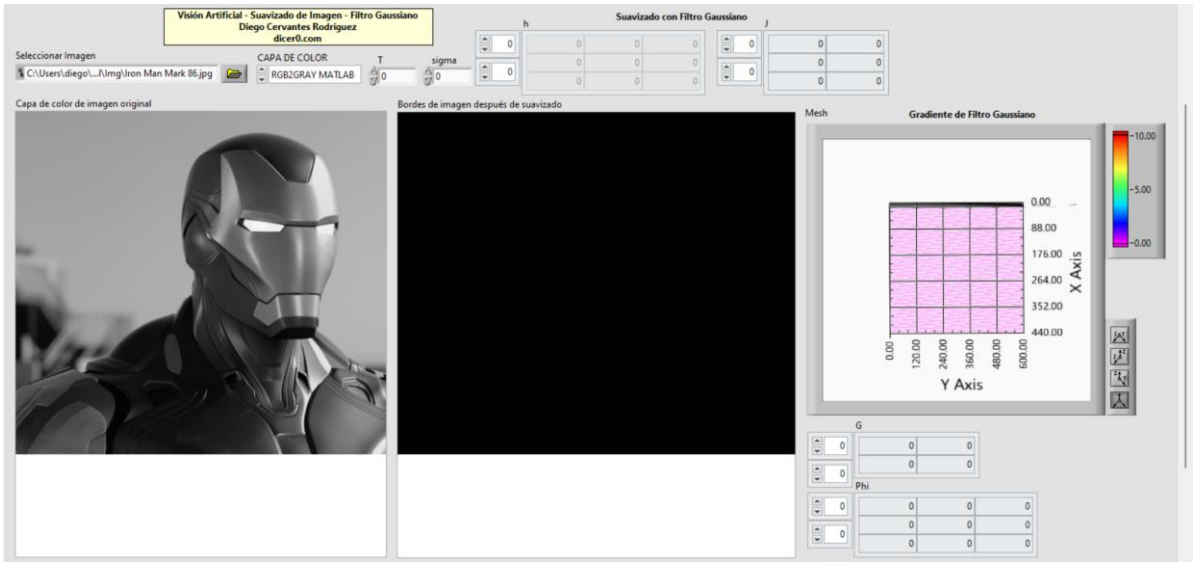
En el siguiente programa se pueden utilizar las capas RGB y HSV para suavizar una imagen, introduciendo el valor del tamaño T del kernel del filtro de Gauss, que indica su ancho y alto, osea la

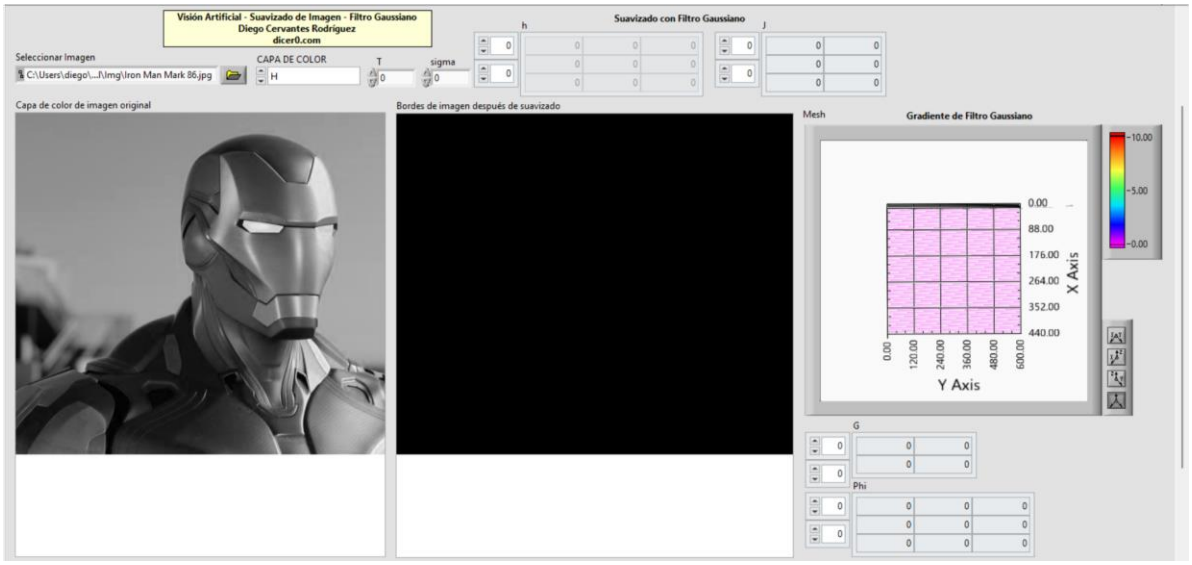
dimensión de la matriz utilizada en la convolución, además de indicar el valor de la variable sigma que representa la varianza o desviación estándar (σ), con esto se podrá ver el resultado del suavizado y su gráfica de Gradiente en la parte derecha de la interfaz gráfica.

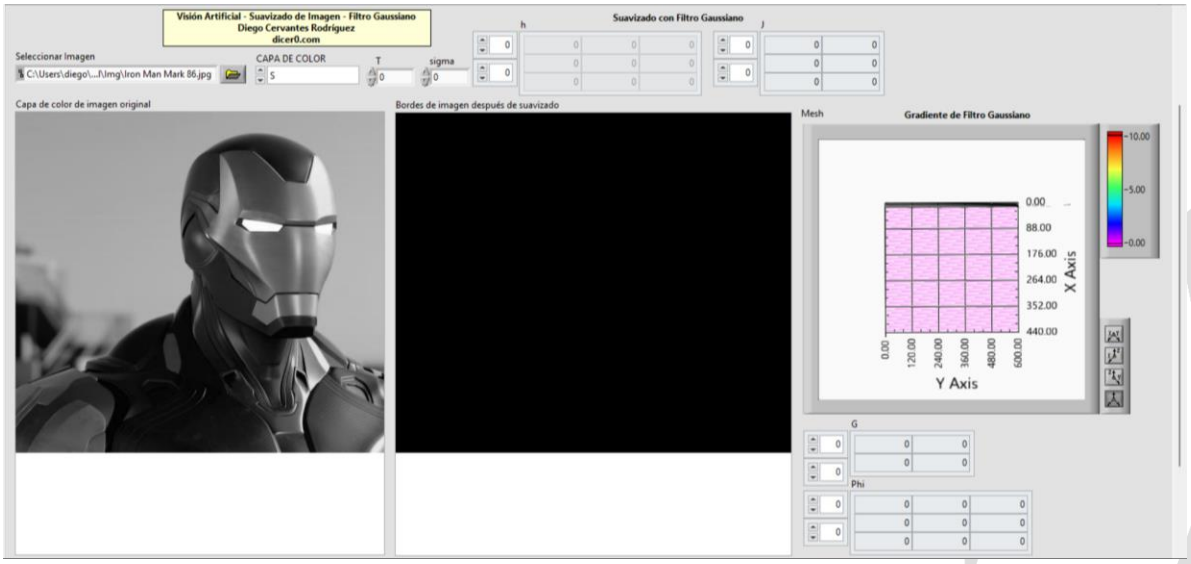


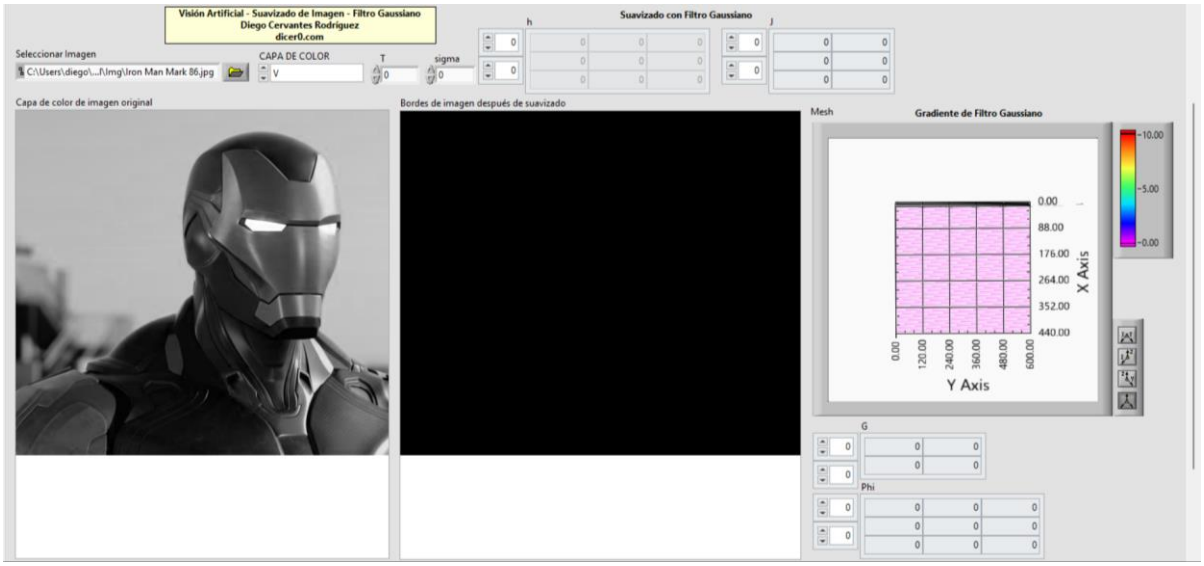


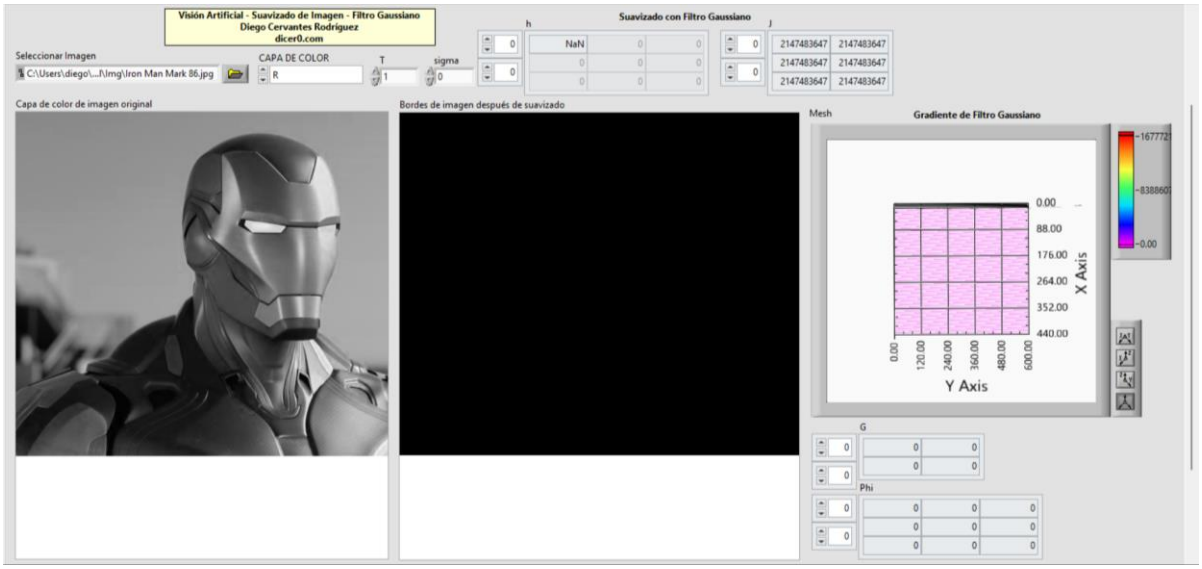


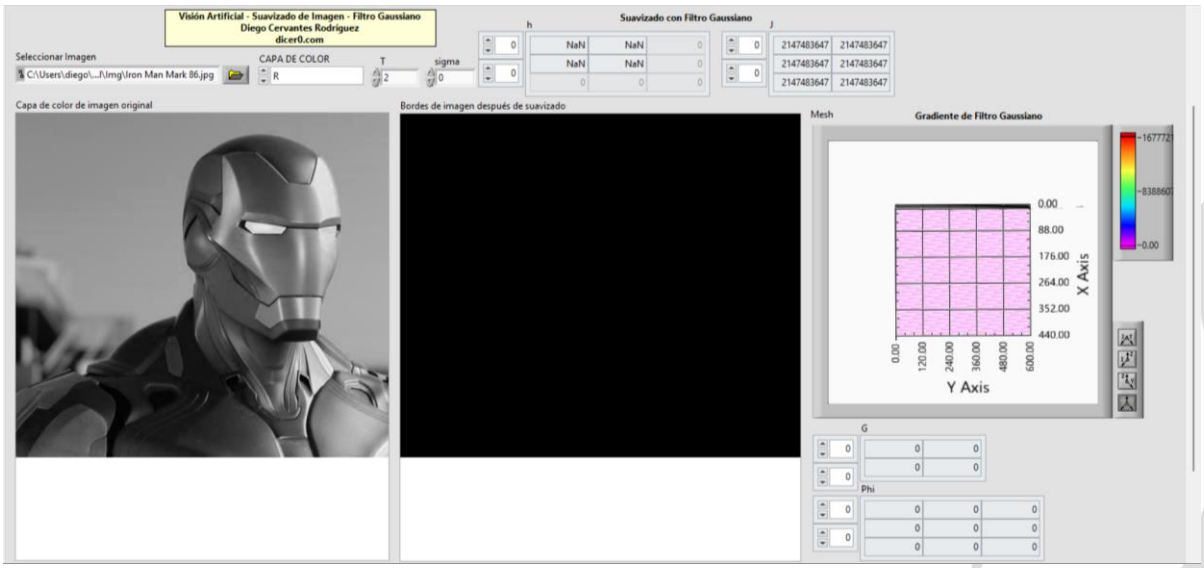

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Suavizado con Filtro Gaussiano
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
RGB2GRAY MATLAB
T
sigma
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

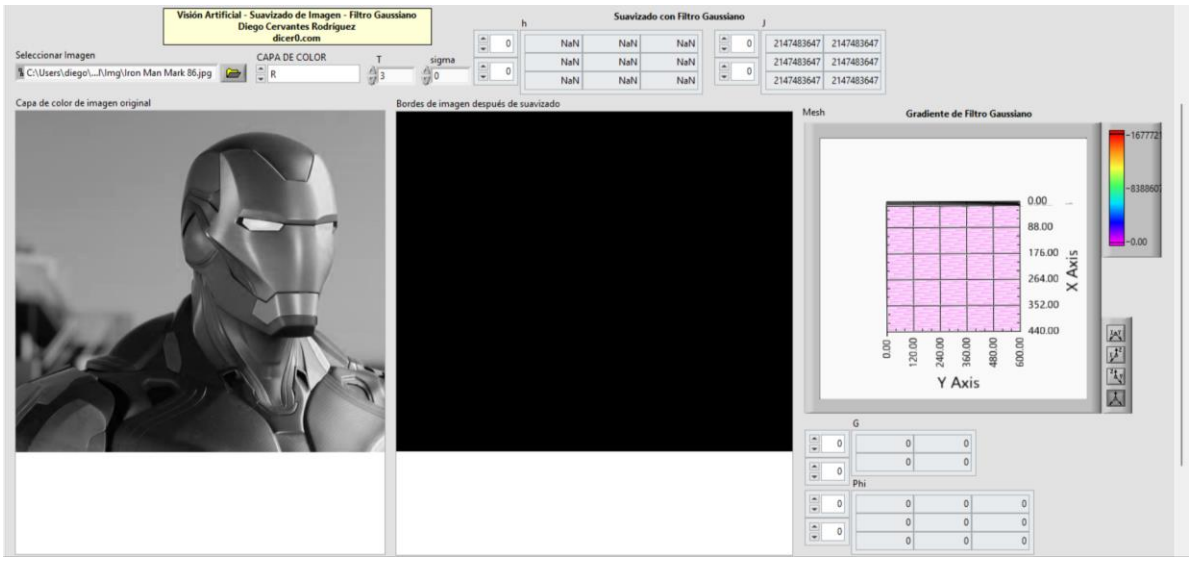

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Suavizado con Filtro Gaussiano
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
H
T
sigma
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

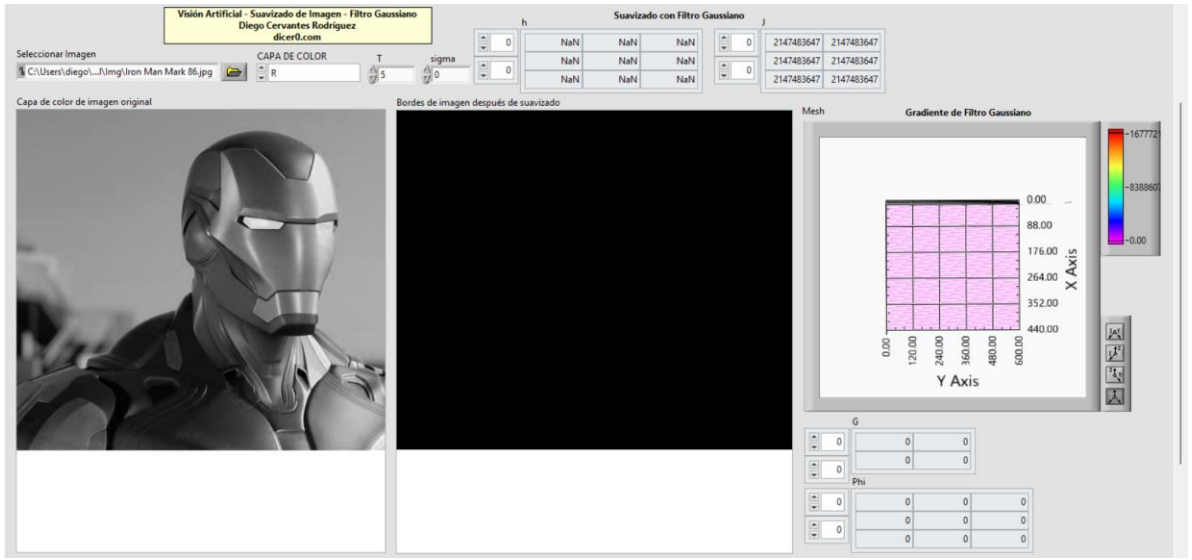

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Suavizado con Filtro Gaussiano
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
5
T
sigma
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

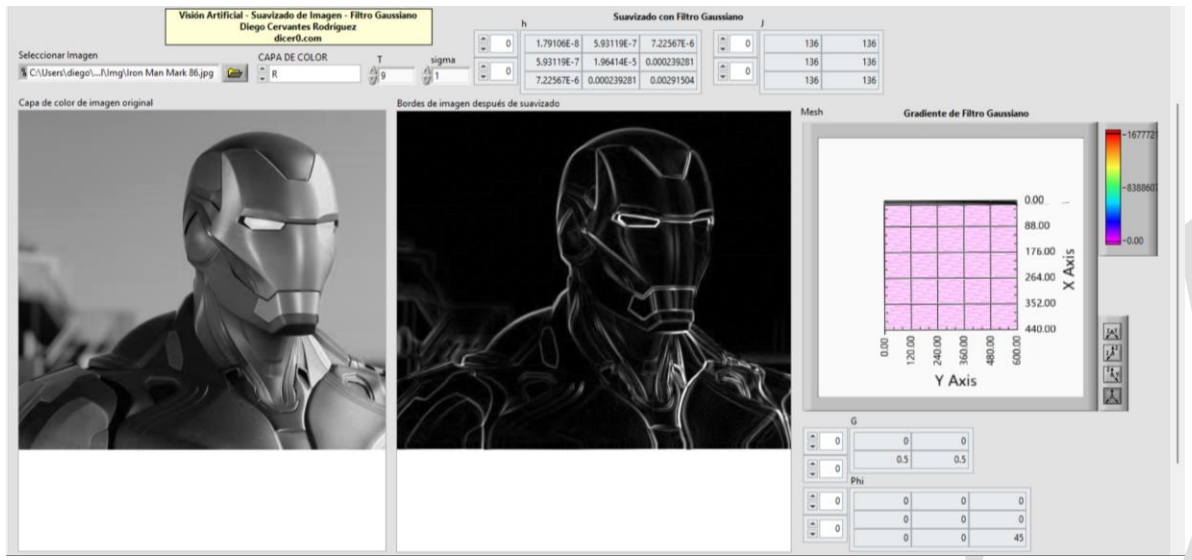

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
V
T
sigma
Suavizado con Filtro Gaussiano
h
J
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

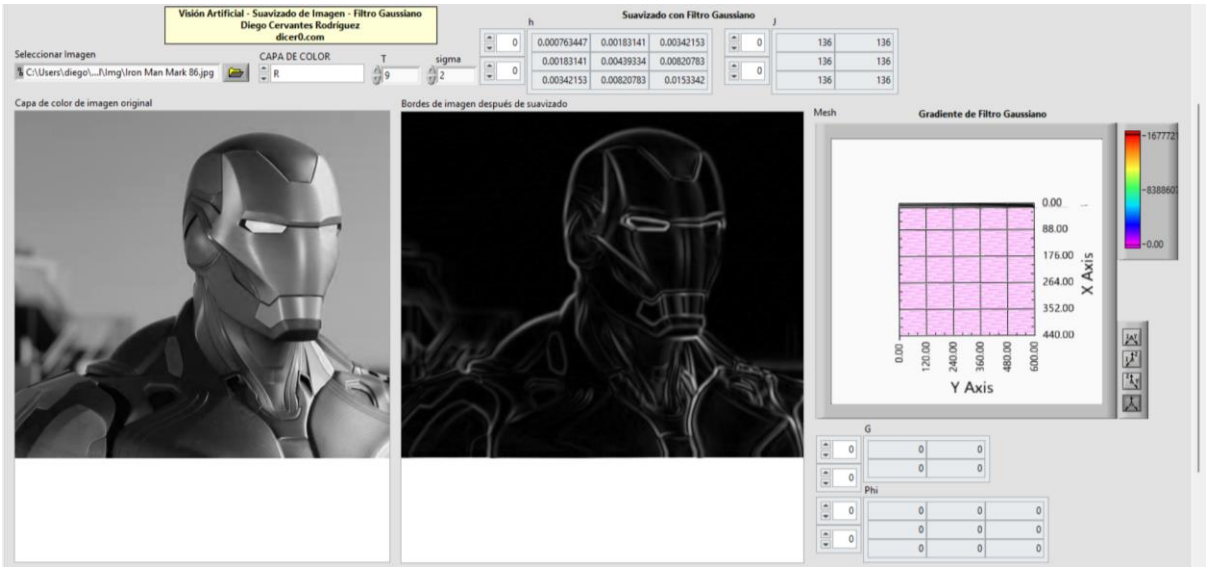

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
sigma
Suavizado con Filtro Gaussiano
h
J
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

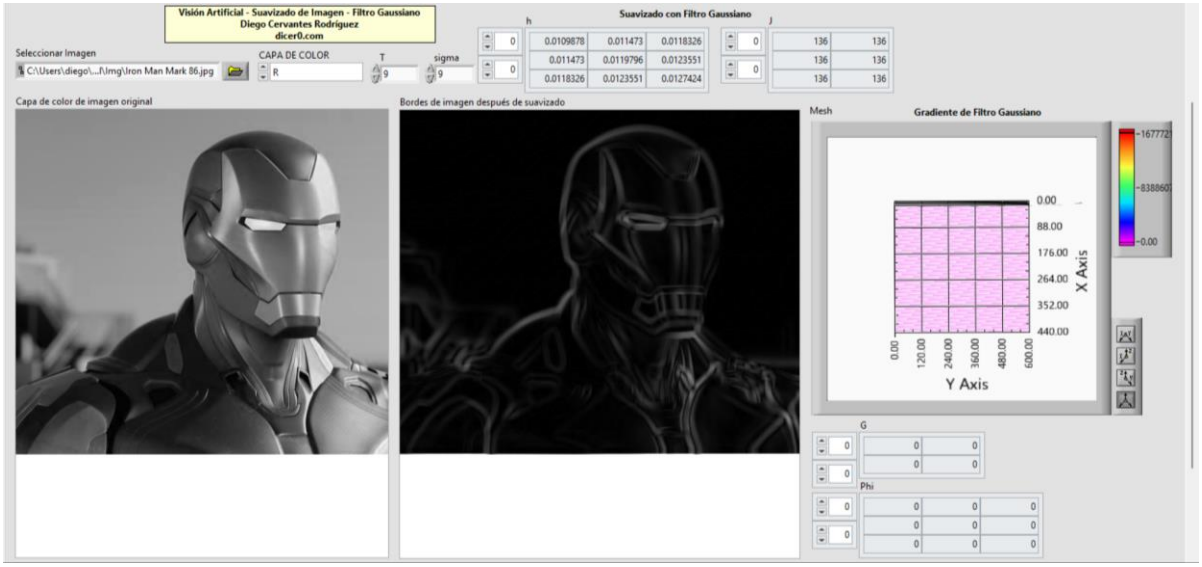

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
sigma
Suavizado con Filtro Gaussiano
h
J
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

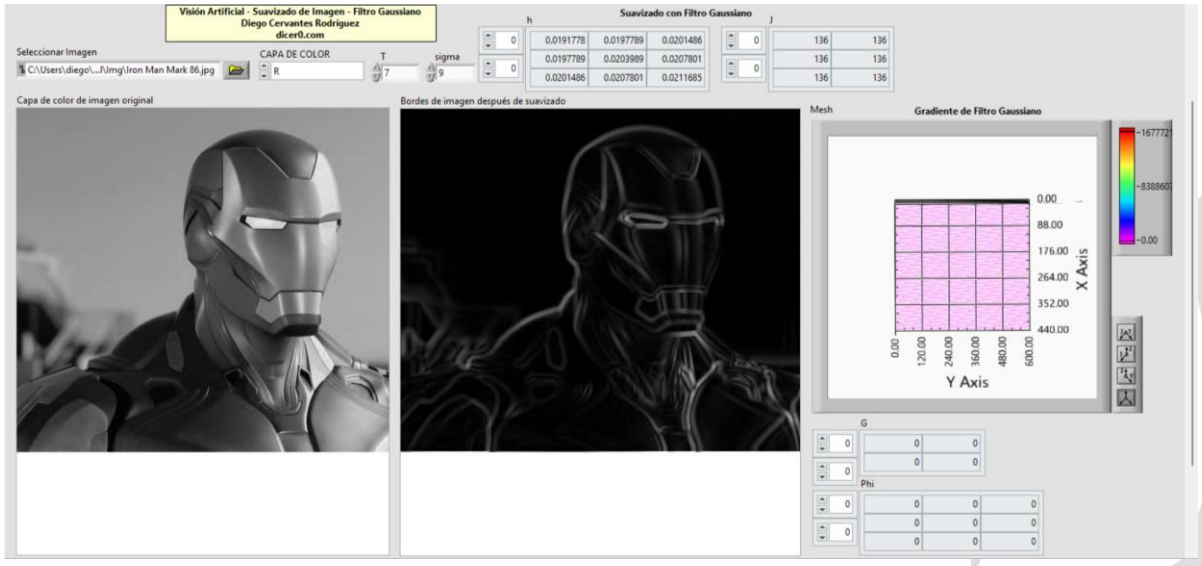

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
3
sigma
0
Suavizado con Filtro Gaussiano
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

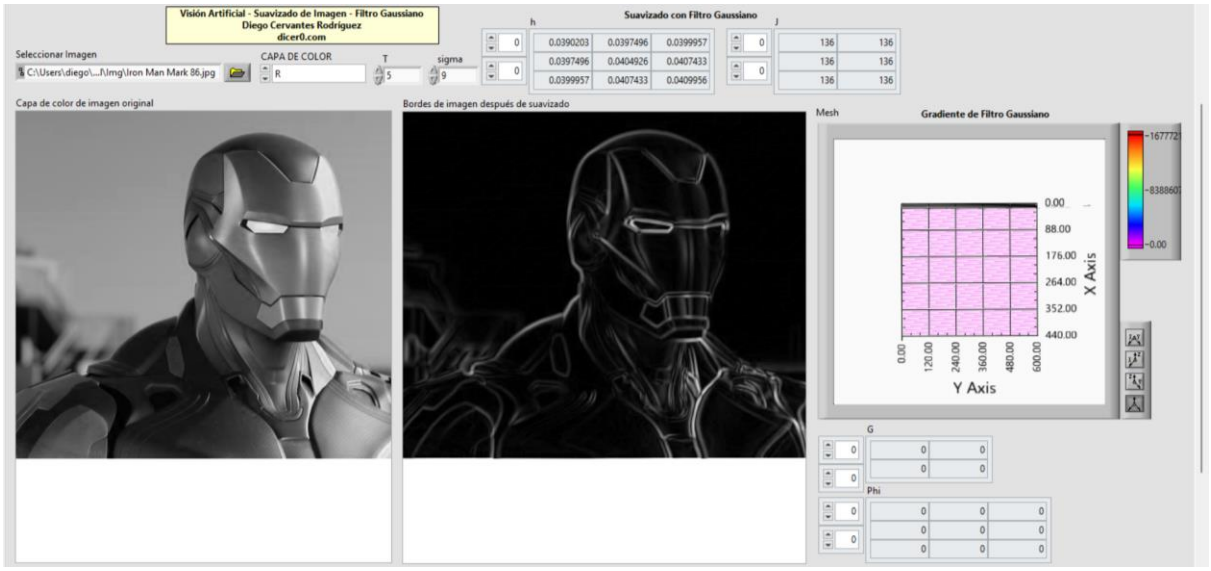

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
5
sigma
0
Suavizado con Filtro Gaussiano
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

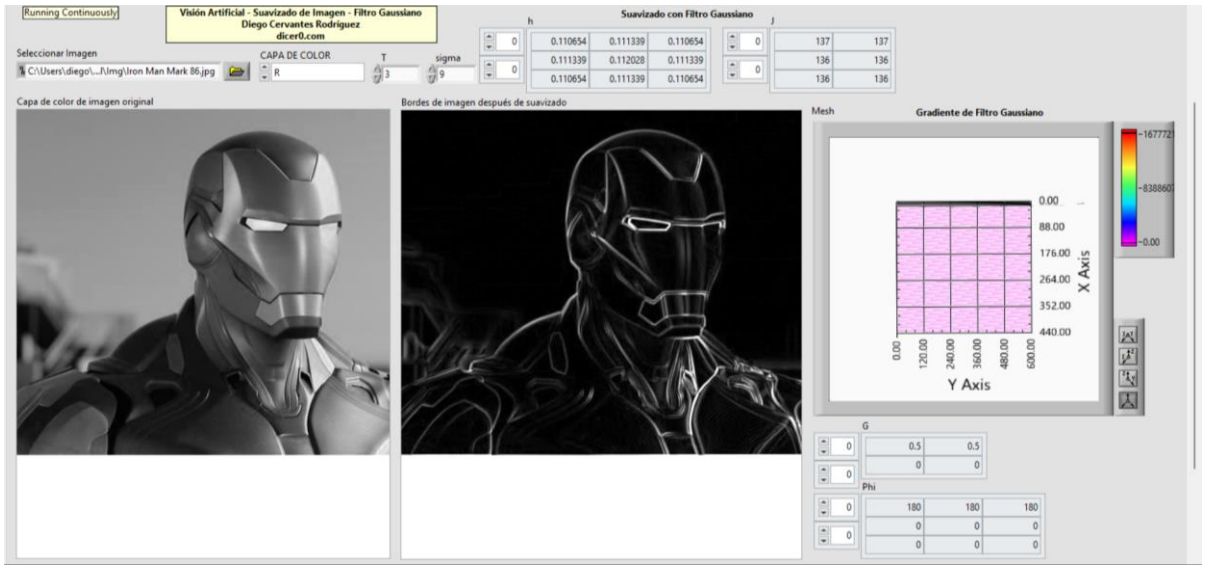

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodríguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
9
sigma
1
Suavizado con Filtro Gaussiano
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi

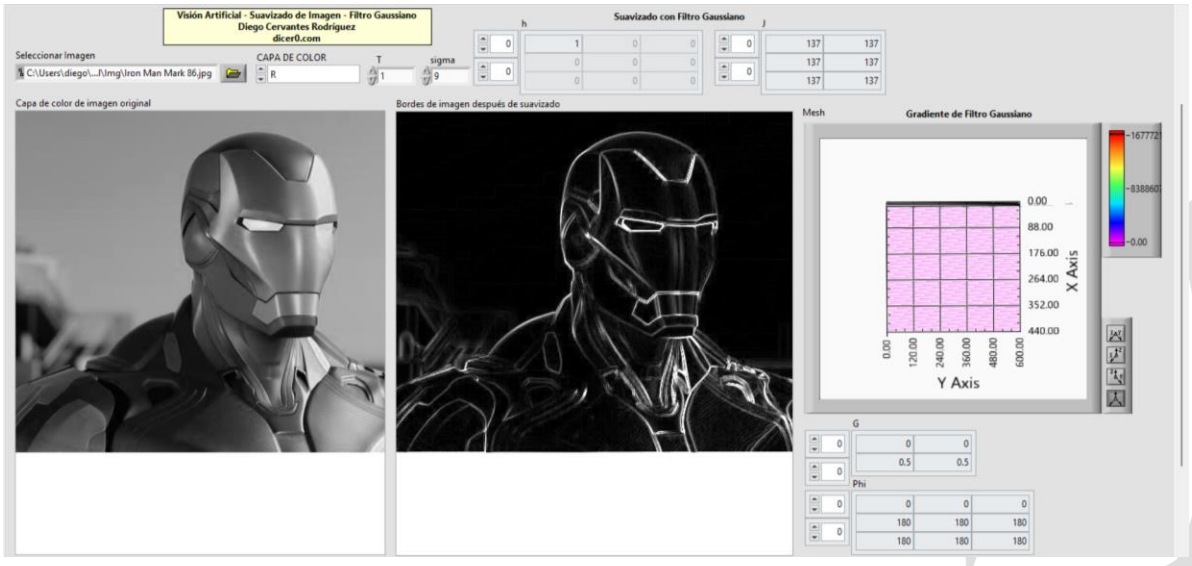

Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
9
sigma
2
Suavizado con Filtro Gaussiano
h
J
0.000763447 0.00183141 0.00342153
0.00183141 0.00439334 0.00820783
0.00342153 0.00820783 0.0153342
136 136
136 136
136 136
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi


Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
9
sigma
9
Suavizado con Filtro Gaussiano
h
J
0.0109878 0.011473 0.0118326
0.011473 0.0119796 0.0123551
0.0118326 0.0123551 0.0127424
136 136
136 136
136 136
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi


Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
7
sigma
9
Suavizado con Filtro Gaussiano
h
J
0.0191778 0.0197789 0.0201486
0.0197789 0.0203989 0.0207801
0.0201486 0.0207801 0.0211685
136 136
136 136
136 136
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi


Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
5
sigma
9
Suavizado con Filtro Gaussiano
h
J
0.0390203 0.0397496 0.0399957
0.0397496 0.0404926 0.0407433
0.0399957 0.0407433 0.0409956
136 136
136 136
136 136
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi


Running Continuously
Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
3
sigma
9
Suavizado con Filtro Gaussiano
h
J
0.110654 0.111339 0.110654
0.111339 0.112028 0.111339
0.110654 0.111339 0.110654
137 137
136 136
136 136
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi


Visión Artificial - Suavizado de Imagen - Filtro Gaussiano
Diego Cervantes Rodriguez
dicer0.com
Seleccionar Imagen
C:\Users\diego\...I\Img\Iron Man Mark 86.jpg
CAPA DE COLOR
R
T
1
sigma
9
Suavizado con Filtro Gaussiano
h
J
1 0 0
0 0 0
0 0 0
137 137
137 137
137 137
Capa de color de imagen original
Bordes de imagen después de suavizado
Mesh
Gradiente de Filtro Gaussiano
X Axis
Y Axis
G
Phi