

Centro de Enseñanza Técnica Industrial.

Visión Artificial.



Practica#2: Operaciones aritméticas Básicas.

Platel: CETI Colomos.

Nombre: Ruiz Macías Luis Enrique – 21310196

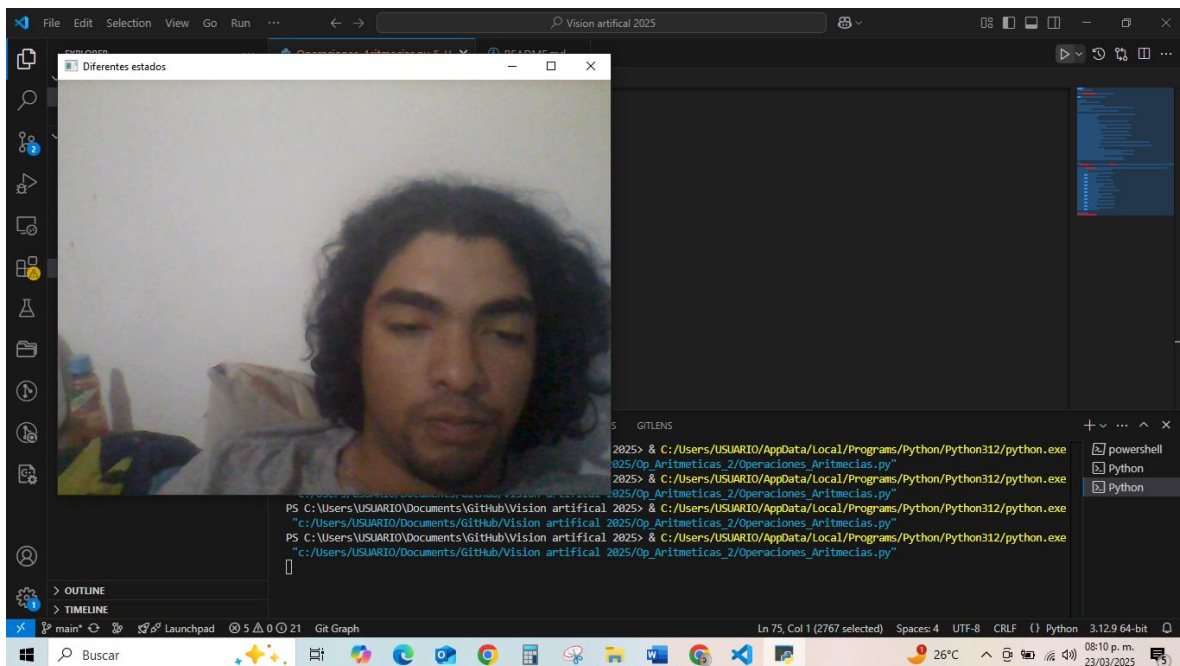
Grado/Grupo: 6°G

Objetivo: Mostrar las operaciones aritméticas básicas en cuanto a la manipulación de imágenes, Siendo estas el aumento/disminución de brillo, el aumento/disminución del contraste, la transpuesta de imagen, aumento del tamaño o reducción, la rotación, o la inversión de colores en tiempo real mediante un la cámara del equipo.

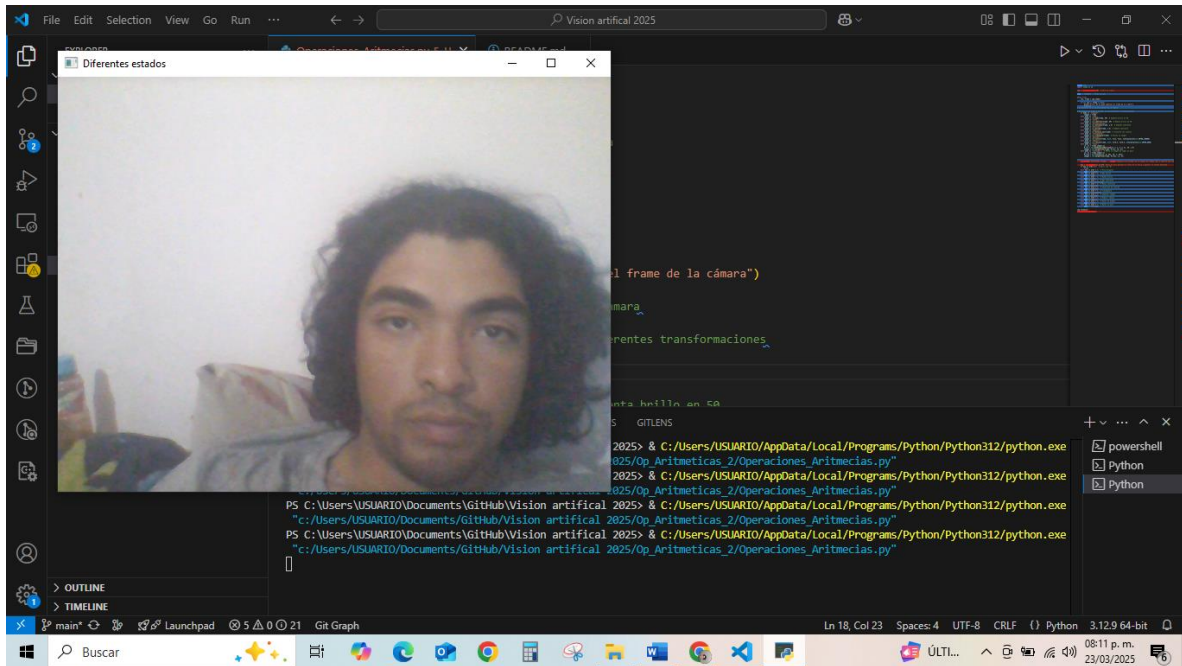
El presente programa se hizo mediante el uso de Python y de Visual Studio Code, en el cual se con la ayuda de Opencv2 para manejar las operaciones básicas, la idea del mismo es la de presentar una imagen de video donde se muestre lo que capta la cámara y a su vez mediante una serie de comandos previamente establecidos cambiar las características de la imagen.

Primero lo que hace el programa es abrir la cámara de la computadora y itera la de forma que crea un bucle en el cual se mantiene recibiendo información de forma constante, y a lo largo del código solo se dedica a la recepción de datos del mismo, mediante el teclado mandamos los valores con las letras previamente establecidas y modificamos la imagen mostrada.

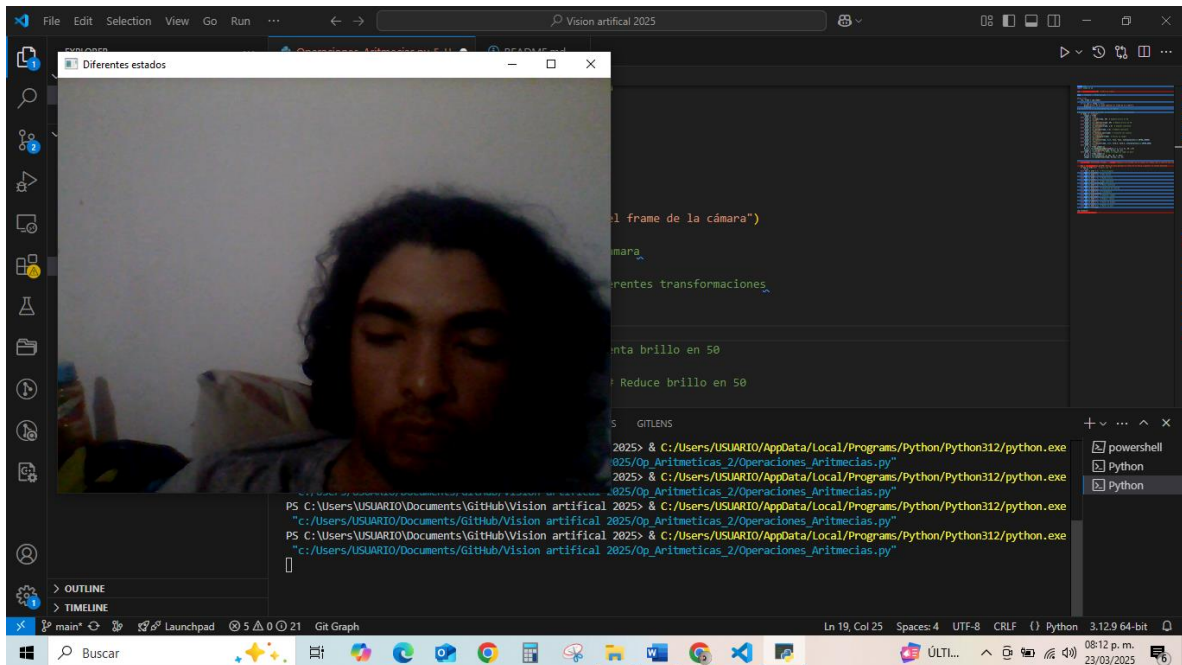
A continuación, se muestran evidencias:



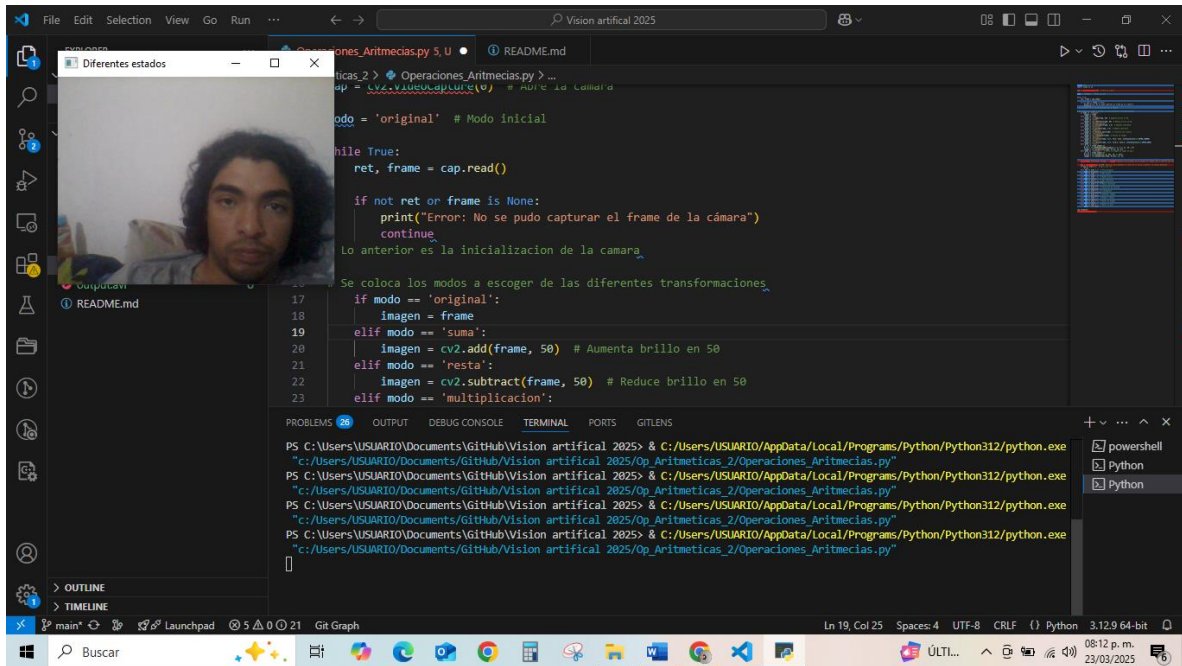
Original.



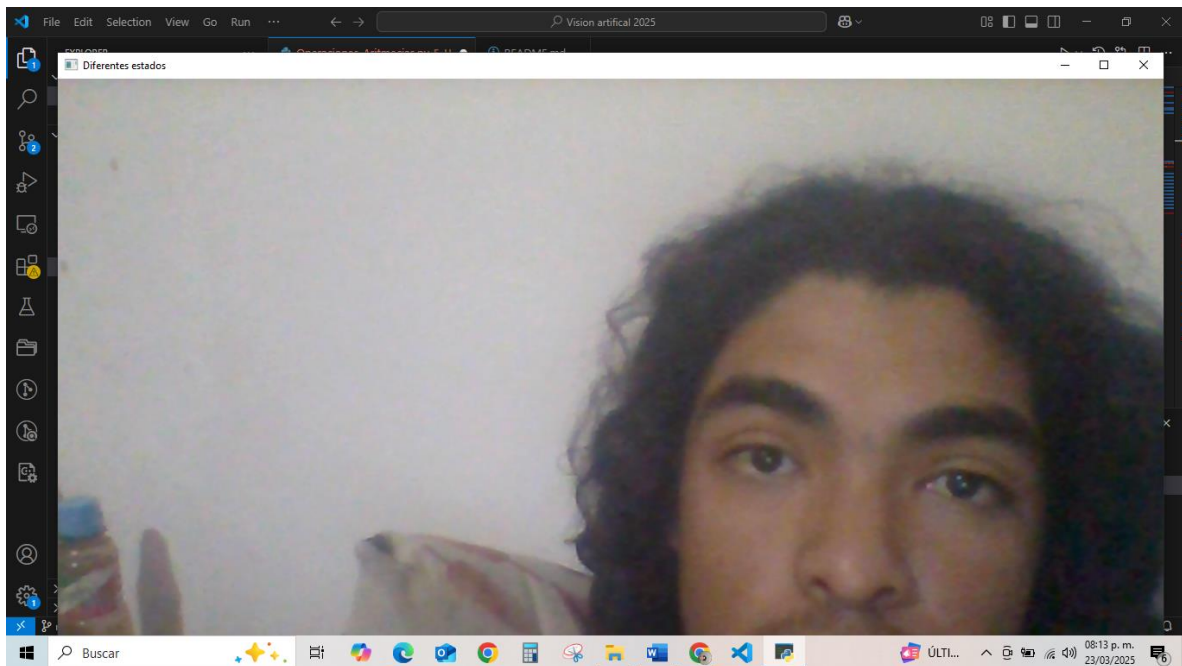
Mas brillo



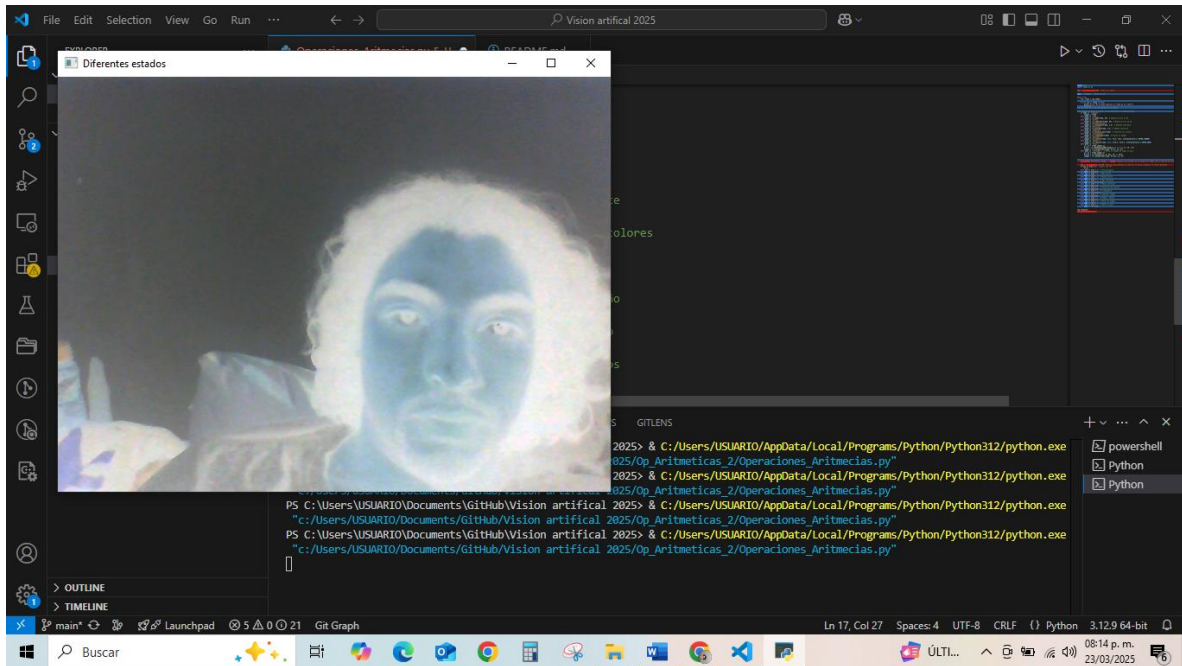
Menos brillo



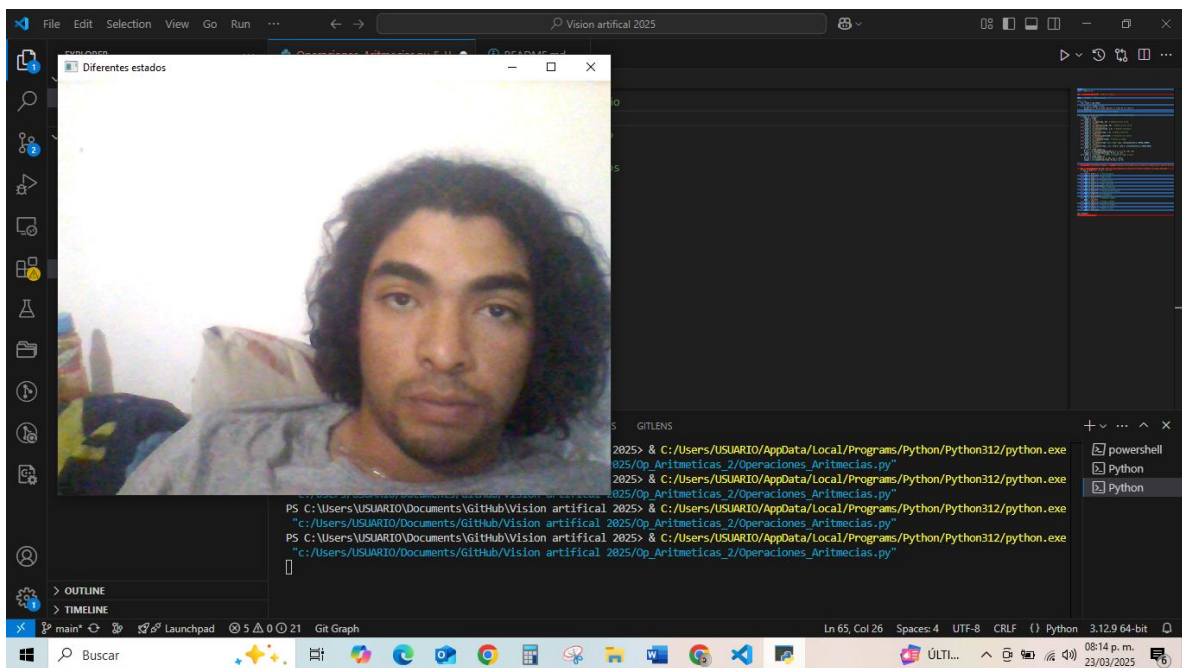
Reducción de tamaño



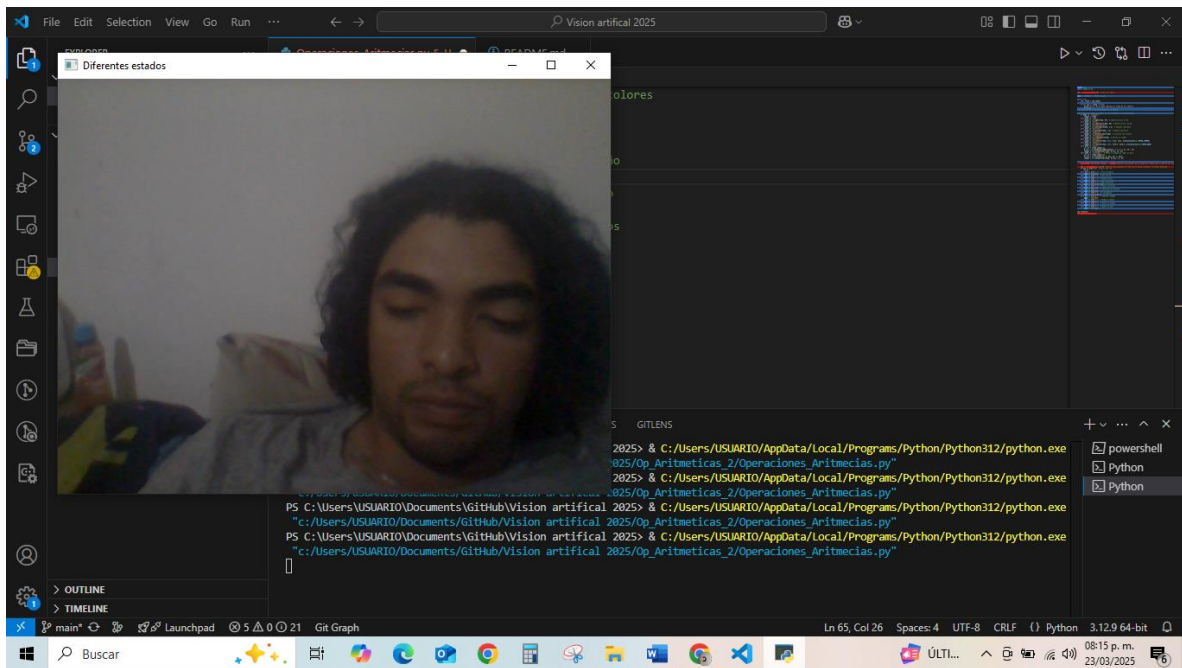
Aumentar tamaño



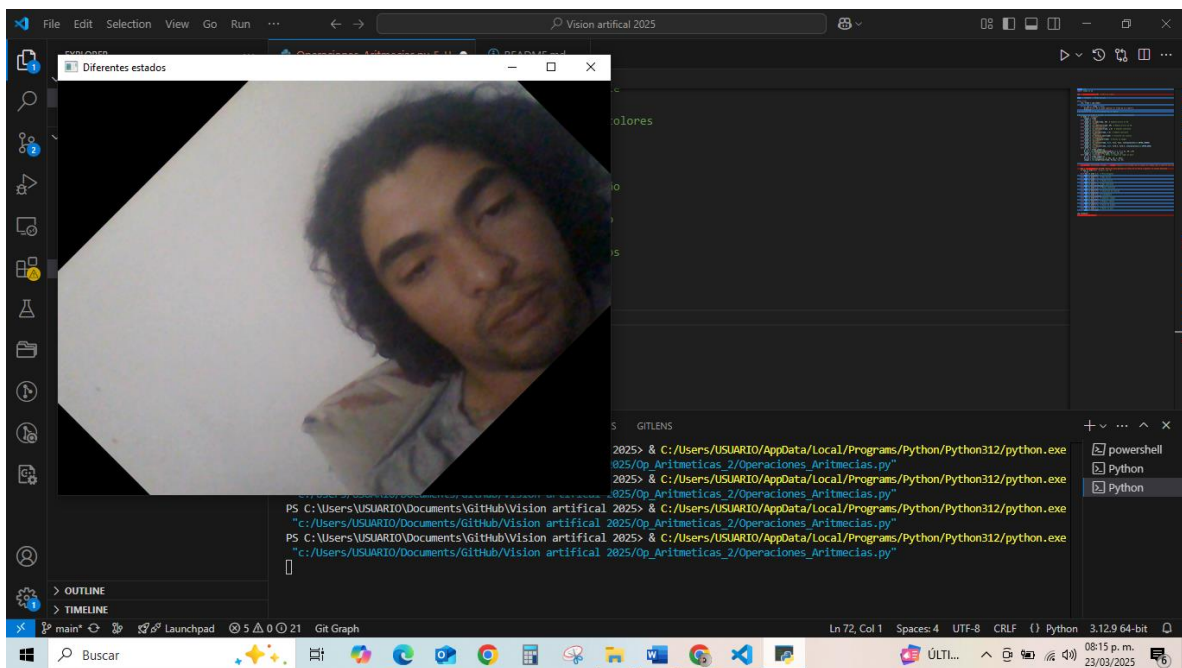
Negativos.



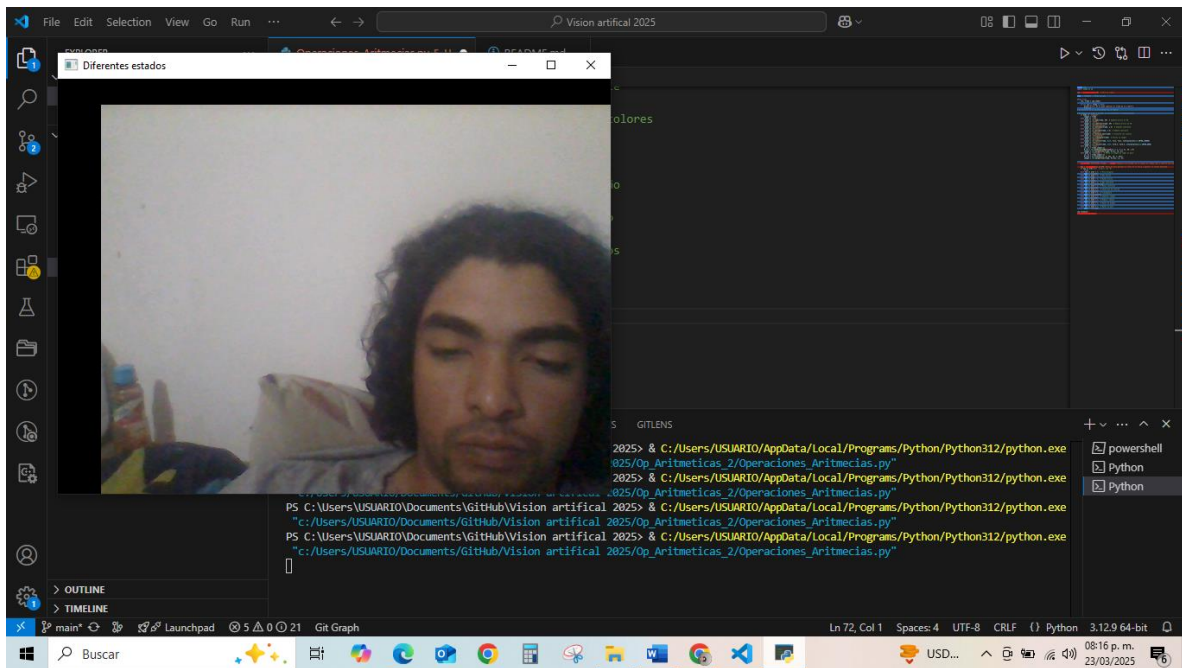
Mas contraste



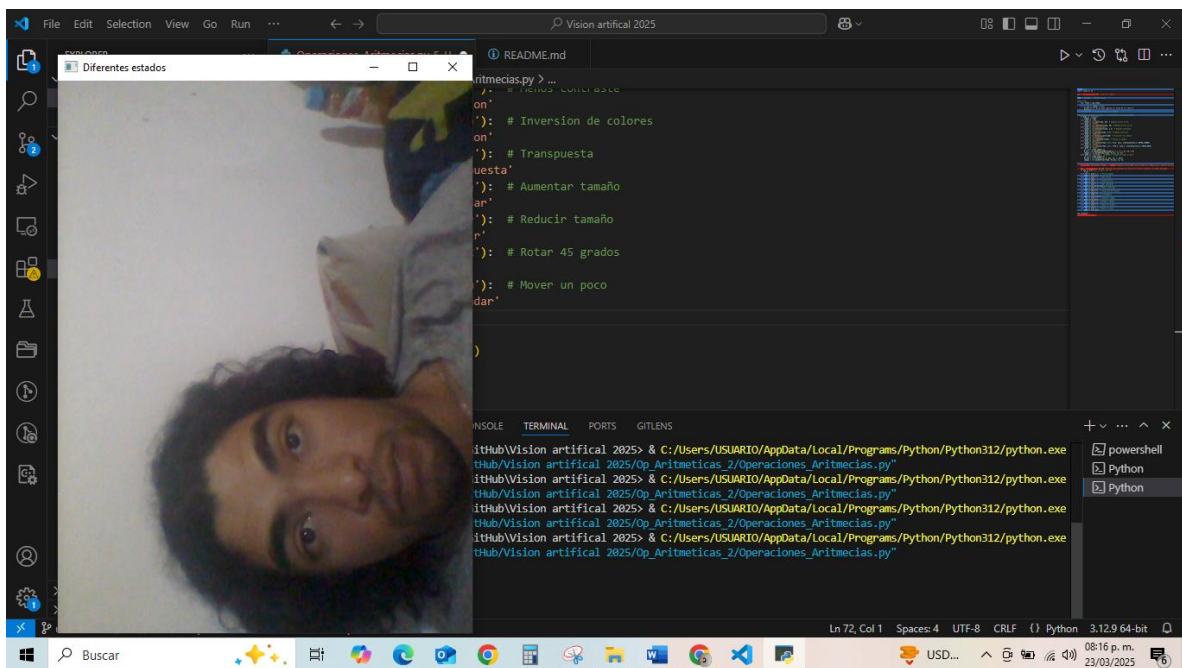
Menos contraste.



Giro a 45°



Movimiento de poscion8(Un poco)



Transpuesta.

Cabe destacar que en un inicio se establecen los datos, el código siguiente solo es para manipular la imagen de video de una forma sencilla y demostrar las operaciones aritméticas en las imágenes.

A continuación, se presentará el código:

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0) # Abre la cámara

modo = 'original' # Modo inicial

while True:
    ret, frame = cap.read()

    if not ret or frame is None:
        print("Error: No se pudo capturar el frame de la cámara")
        continue
    # Lo anterior es la inicializacion de la camara

    # Se coloca los modos a escoger de las diferentes transformaciones
    if modo == 'original':
        imagen = frame
    elif modo == 'suma':
        imagen = cv2.add(frame, 50) # Aumenta brillo en 50
    elif modo == 'resta':
        imagen = cv2.subtract(frame, 50) # Reduce brillo en 50
    elif modo == 'multiplicacion':
        imagen = cv2.multiply(frame, 1.5) # Aumenta contraste
    elif modo == 'division':
        imagen = cv2.divide(frame, 1.5) # Reduce contraste
```



```

elif modo == 'negacion':
    imagen = cv2.bitwise_not(frame) # Invierte los colores
elif modo == 'transpuesta':
    imagen = cv2.transpose(frame) # Voltea la imagen
elif modo == 'agrandar':
    imagen = cv2.resize(frame, None, fx=2, fy=2,
interpolation=cv2.INTER_LINEAR)
elif modo == 'reducir':
    imagen = cv2.resize(frame, None, fx=0.5, fy=0.5,
interpolation=cv2.INTER_AREA)
elif modo == 'rotar':
    (h, w) = frame.shape[:2]
    M_rot = cv2.getRotationMatrix2D((w // 2, h // 2), 45, 1.0)
    imagen = cv2.warpAffine(frame, M_rot, (w, h))
elif modo == 'trasladar': # MUEve la imagen de lugar un poco
    (h, w) = frame.shape[:2]
    M_tras = np.float32([[1, 0, 50], [0, 1, 30]])
    imagen = cv2.warpAffine(frame, M_tras, (w, h))

cv2.imshow('Diferentes estados ', imagen) #Captura los estados de la camara
en timepo real y muestra los cambios

key = cv2.waitKey(1) & 0xFF #Funcion para guardar el valor de la tecla y
generar un estado definido
if key == ord('q'): # Salir con "q"
    break
elif key == ord('o'): # Para Original
    modo = 'original'
elif key == ord('+'): # Mas brillo

```

```
        modo = 'suma'
elif key == ord('-'): # Menos brillo
    modo = 'resta'
elif key == ord('*'): # Mas contraste
    modo = 'multiplicacion'
elif key == ord('/'): # Menos contraste
    modo = 'division'
elif key == ord('n'): # Inversion de colores
    modo = 'negacion'
elif key == ord('t'): # Transpuesta
    modo = 'transpuesta'
elif key == ord('a'): # Aumentar tamaño
    modo = 'agrandar'
elif key == ord('r'): # Reducir tamaño
    modo = 'reducir'
elif key == ord('x'): # Rotar 45 grados
    modo = 'rotar'
elif key == ord('m'): # Mover un poco
    modo = 'trasladar'

cap.release()
cv2.destroyAllWindows()
```