[Practica 3]

[Sistema de Visión Artificial]

Loera Contreras Gildardo Yair – 19110230 – 7E

profesor [Mauricio alejandro cabrera arellano]

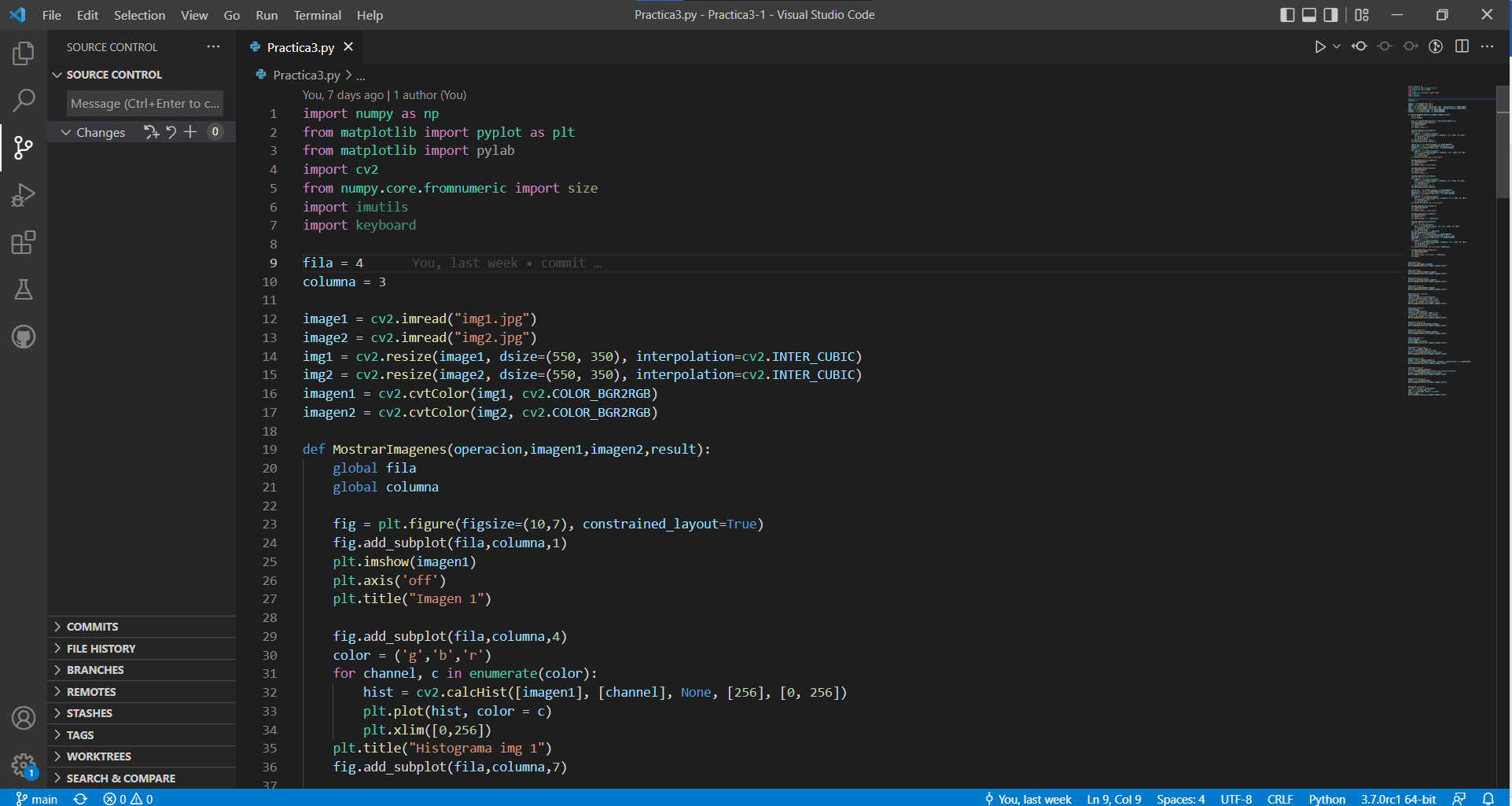


**PROBLEMA**

Ecualizado del Histograma.

De la práctica número 2 ecualizar las imágenes mostrar las operaciones y su ecualización.

**CODIGO**



import numpy as np

from matplotlib import pyplot as plt

from matplotlib import pylab

import cv2

from numpy.core.fromnumeric import size

import imutils

import keyboard

fila = 4

columna = 3

image1 = cv2.imread("img1.jpg")

image2 = cv2.imread("img2.jpg")

img1 = cv2.resize(image1, dsize=(550, 350), interpolation=cv2.INTER\_CUBIC)

img2 = cv2.resize(image2, dsize=(550, 350), interpolation=cv2.INTER\_CUBIC)

imagen1 = cv2.cvtColor(img1, cv2.COLOR\_BGR2RGB)

imagen2 = cv2.cvtColor(img2, cv2.COLOR\_BGR2RGB)

def MostrarImagenes(operacion,imagen1,imagen2,result):

    global fila

    global columna

    fig = plt.figure(figsize=(10,7), constrained\_layout=True)

    fig.add\_subplot(fila,columna,1)

    plt.imshow(imagen1)

    plt.axis('off')

    plt.title("Imagen 1")

    fig.add\_subplot(fila,columna,4)

    color = ('g','b','r')

    for channel, c in enumerate(color):

        hist = cv2.calcHist([imagen1], [channel], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma img 1")

    fig.add\_subplot(fila,columna,7)

    img\_to\_yuv = cv2.cvtColor(imagen1,cv2.COLOR\_RGB2YUV)

    img\_to\_yuv[:,:,0] = cv2.equalizeHist(img\_to\_yuv[:,:,0])

    EcuaHist = cv2.cvtColor(img\_to\_yuv, cv2.COLOR\_YUV2RGB)

    color = ('g','b','r')

    for channel, c in enumerate(color):

        hist = cv2.calcHist([EcuaHist], [channel], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma img 1 Ecualizada")

    fig.add\_subplot(fila,columna,10)

    plt.imshow(EcuaHist)

    plt.axis('off')

    plt.title("Imagen 1 Ecualizada")

    fig.add\_subplot(fila,columna,3)

    plt.imshow(imagen2)

    plt.axis('off')

    plt.title("Imagen 2")

    fig.add\_subplot(fila,columna,6)

    color = ('g','b','r')

    for channel, c in enumerate(color):

        hist = cv2.calcHist([imagen2], [channel], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma img 2")

    fig.add\_subplot(fila,columna,9)

    img\_to\_yuv = cv2.cvtColor(imagen2,cv2.COLOR\_RGB2YUV)

    img\_to\_yuv[:,:,0] = cv2.equalizeHist(img\_to\_yuv[:,:,0])

    EcuaHist2 = cv2.cvtColor(img\_to\_yuv, cv2.COLOR\_YUV2RGB)

    color = ('g','b','r')

    for channel, c in enumerate(color):

        hist = cv2.calcHist([EcuaHist2], [channel], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma img 2 Ecualizada")

    fig.add\_subplot(fila,columna,12)

    plt.imshow(EcuaHist2)

    plt.axis('off')

    plt.title("Imagen 2 Ecualizada")

    fig.add\_subplot(fila,columna,2)

    plt.imshow(result)

    plt.axis('off')

    plt.title("Imagen 2: "+ operacion)

    fig.add\_subplot(fila,columna,5)

    color = ('g','b','r')

    for i, c in enumerate(color):

        hist = cv2.calcHist([result], [i], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma: "+ operacion)

    fig.add\_subplot(fila,columna,8)

    img\_to\_yuv = cv2.cvtColor(result,cv2.COLOR\_RGB2YUV)

    img\_to\_yuv[:,:,0] = cv2.equalizeHist(img\_to\_yuv[:,:,0])

    equaimgOp = cv2.cvtColor(img\_to\_yuv, cv2.COLOR\_YUV2RGB)

    color = ('g','b','r')

    for channel, c in enumerate(color):

        hist = cv2.calcHist([equaimgOp], [channel], None, [256], [0, 256])

        plt.plot(hist, color = c)

        plt.xlim([0,256])

    plt.title("Histograma img Ecualizada:"+operacion)

    fig.add\_subplot(fila,columna,11)

    plt.imshow(equaimgOp)

    plt.axis('off')

    plt.title("Imagen Ecualizada: "+ operacion)

    plt.show()

operacion="Suma"

result=cv2.add(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Resta"

result=cv2.subtract(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Multiplicacion"

result=cv2.multiply(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Division"

result=cv2.divide(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Raiz cuadrada"

result=imagen1

result=cv2.sqrt(np.float32(result))

result=np.asarray(result, dtype = int)

cv2.imwrite("resultadoRaiz.jpg",result)

result = cv2.imread('resultadoRaiz.jpg')

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Potencia"

result=imagen1

result=cv2.pow(result,2)

result=np.asarray(result, dtype = int)

cv2.imwrite("resultadoPot.jpg",result)

result = cv2.imread('resultadoPot.jpg', 1)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Conjuncion"

result=cv2.bitwise\_and(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Disyuncion"

result=cv2.bitwise\_or(imagen1,imagen2)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Negacion"

result=imagen2

result=image= 255-result

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Translacion"

rows,cols = imagen1.shape[:2]

M = np.float32([[1,0,100],[0,1,50]])

result = cv2.warpAffine(img1,M,(cols,rows))

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Escalado"

height, width = imagen1.shape[:2]

result= cv2.resize(imagen1,(2\*width, 2\*height), interpolation = cv2.INTER\_CUBIC)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Rotacion"

rows,cols = imagen1.shape[:2]

M = cv2.getRotationMatrix2D(((cols-1)/2.0,(rows-1)/2.0),90,1)

result = cv2.warpAffine(img1,M,(cols,rows))

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Transpuesta"

result = cv2.transpose(img1)

MostrarImagenes(operacion,imagen1,imagen2,result)

operacion="Logaritmo N"

c = 255 / np.log(1 + np.max(img2))

logn = c \* (np.log(img2 + 1))

logn = np.array(logn, dtype = np.uint8)

result = logn

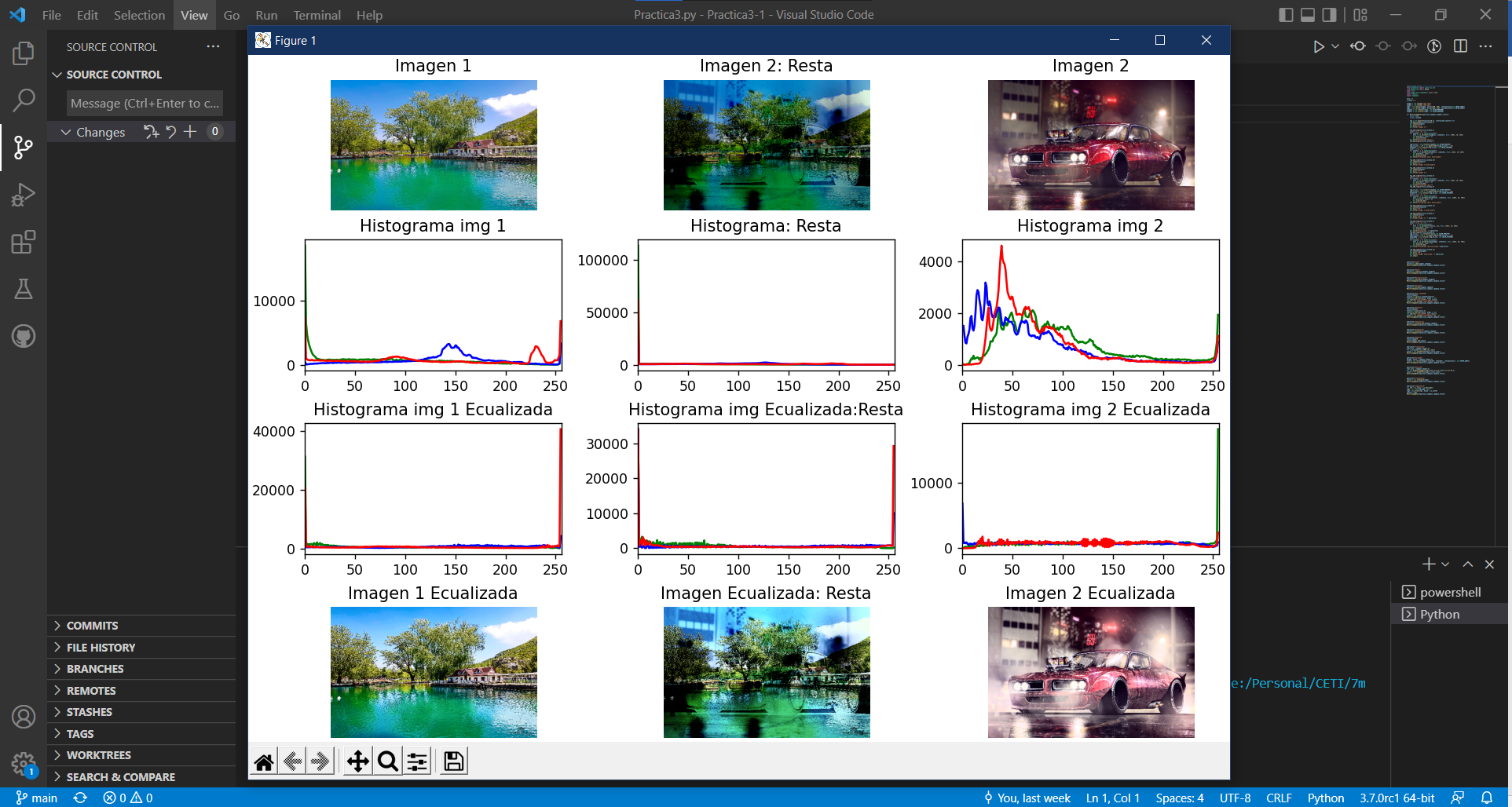
MostrarImagenes(operacion,imagen1,imagen2,result)

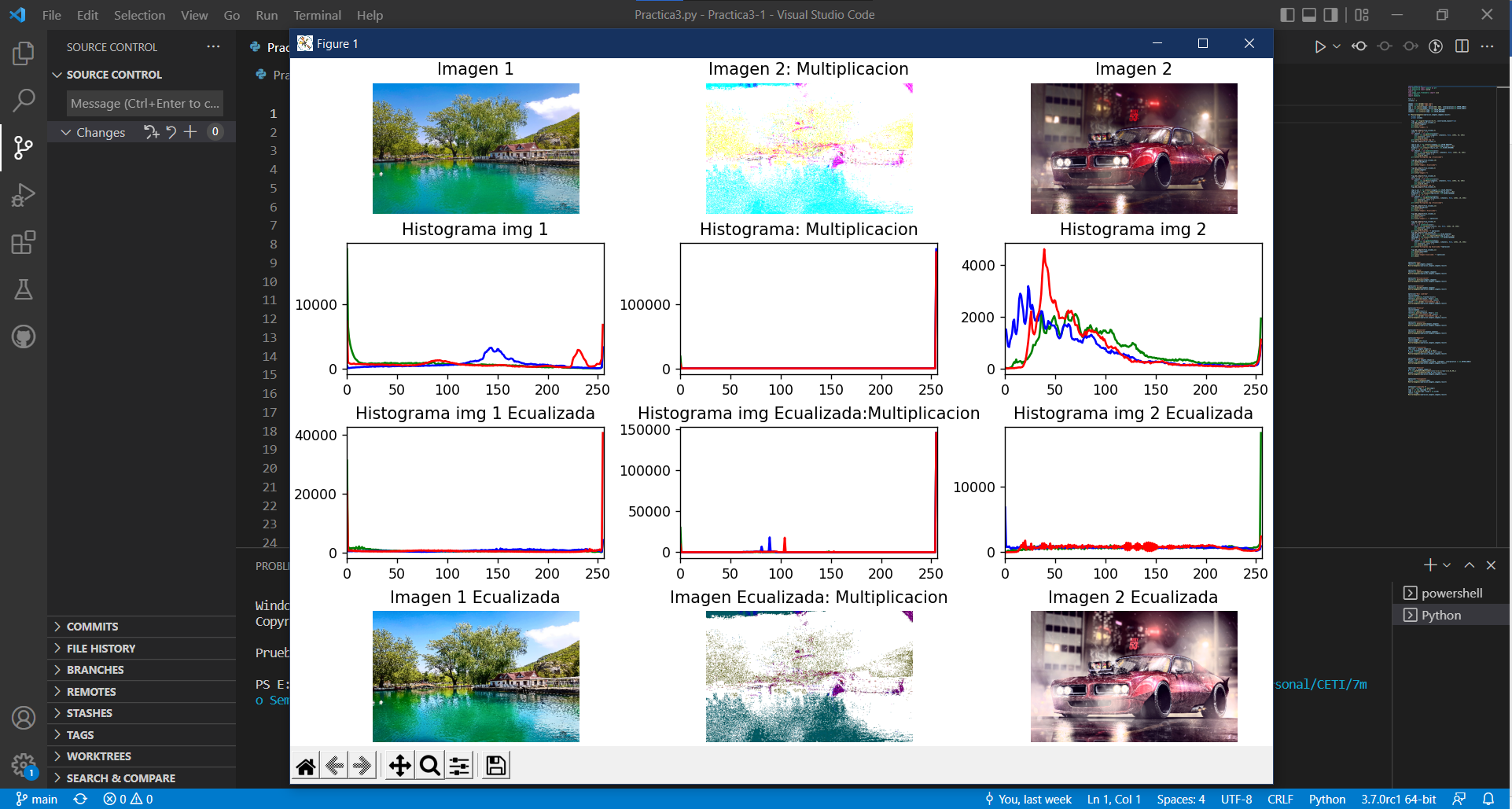
**EVIDENCIAS**

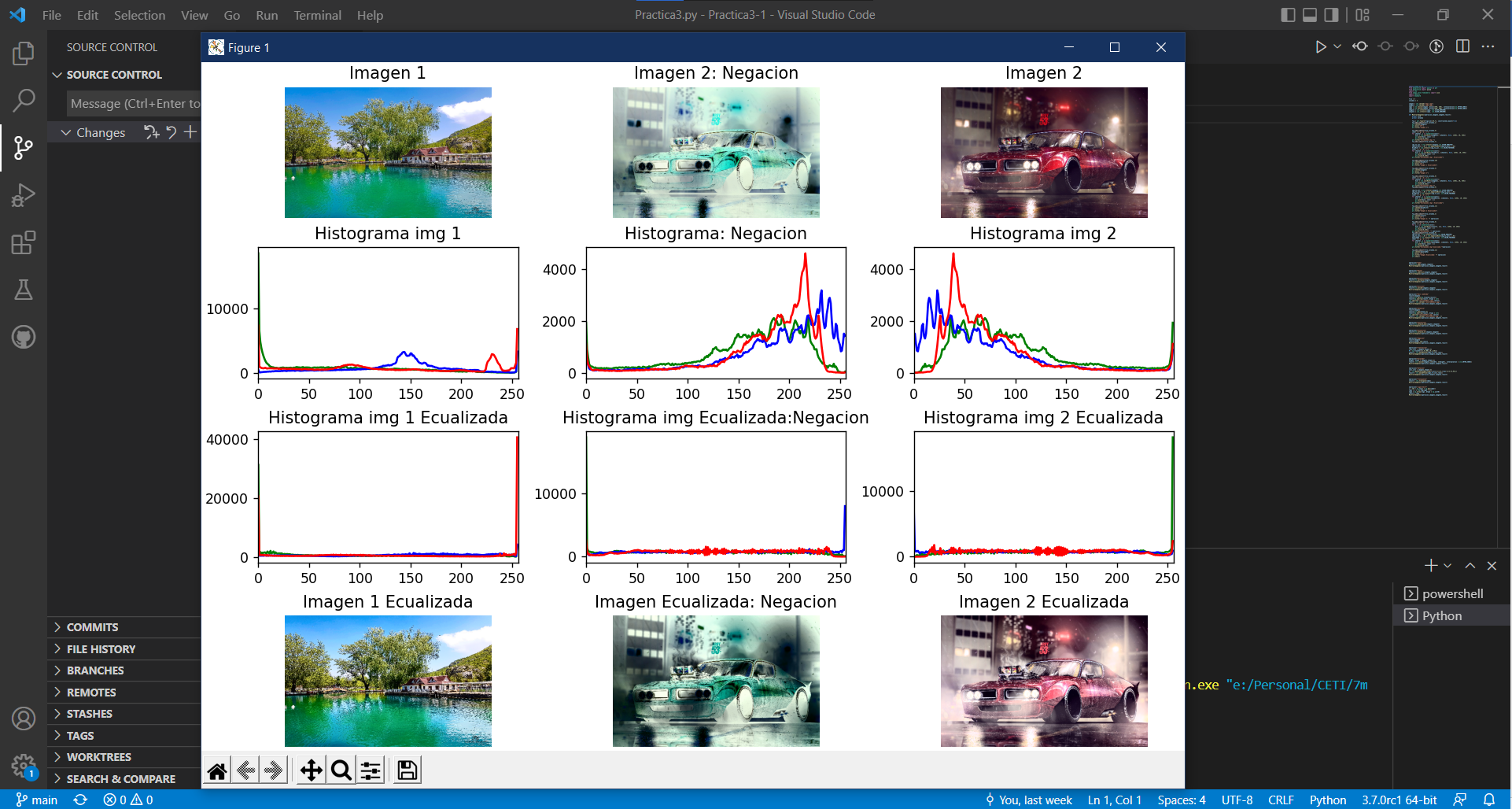
Al correr el programa y nos arroja a la vista las imágenes con las que se trabajaron para su histograma y ecualización de la imagen.

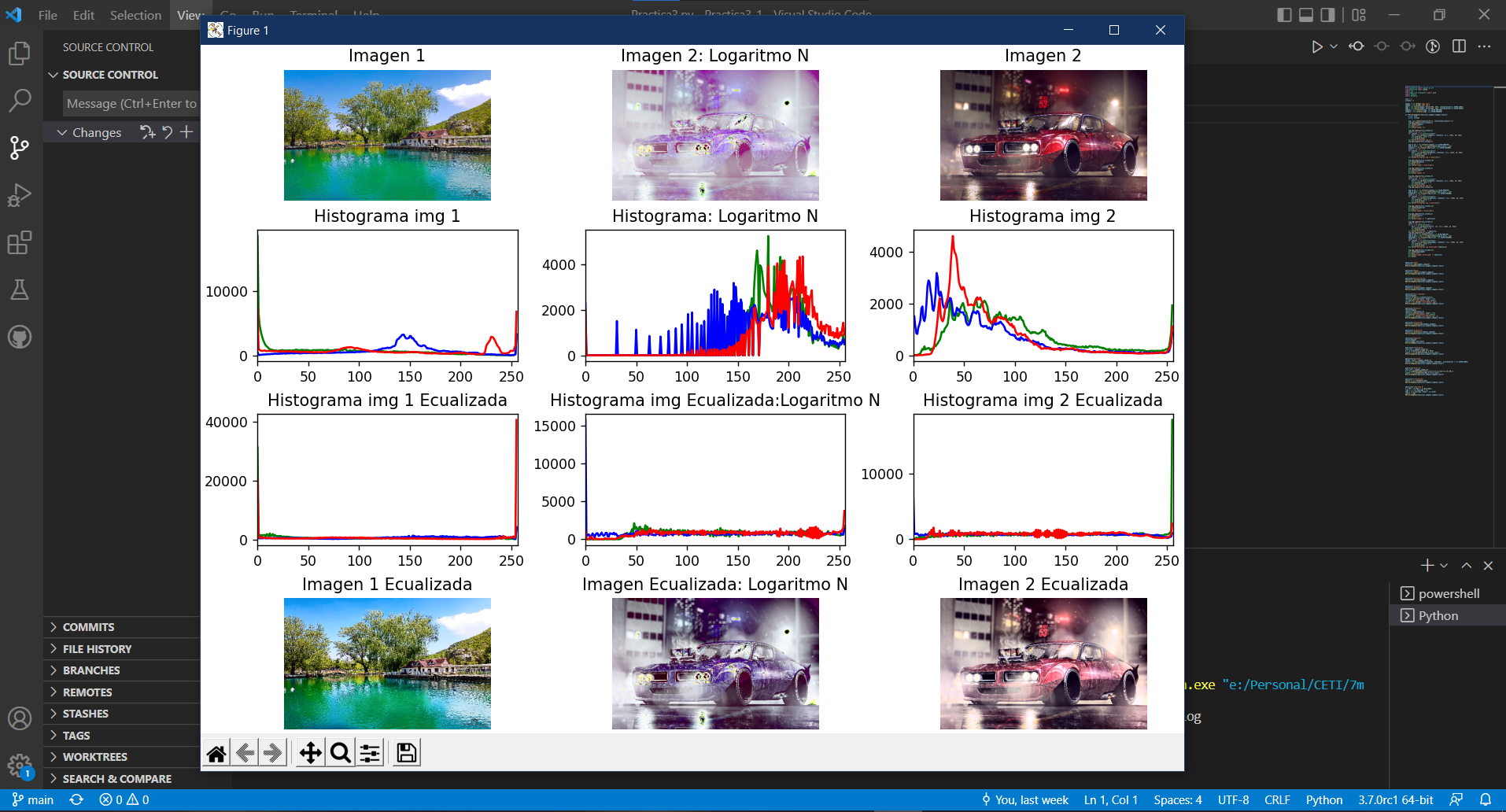
Mostrando así cada una de las operaciones que se hicieron en las imágenes con sus respectivos histogramas y ecualización.











GITHUB:

<https://github.com/Gylc87/Practica3.git>