



06/05/2025

# SEGUNDO PARCIAL

## PRÁCTICA 3

Maestro:

Mauricio Alejandro Cabrera  
Arellano

Alumno:

Orozco Dueñas Roberto Eduardo  
6 - G

Ingeniería Mecatrónica

**OBJETIVO:**

Imprimir en una sola ventana imagen - histograma - imagen ecualizada - histograma ecualizado.

**CÓDIGO:**

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

# Leer la imagen en escala de grises
img = cv2.imread('Img_P2.png', cv2.IMREAD_GRAYSCALE)

# Calcular histograma original
hist_original = cv2.calcHist([img], [0], None, [256], [0, 256])

# Ecualizar la imagen
img_eq = cv2.equalizeHist(img)

# Calcular histograma de la imagen ecualizada
hist_eq = cv2.calcHist([img_eq], [0], None, [256], [0, 256])

# Mostrar todo en una sola ventana
plt.figure(figsize=(12, 6))

plt.subplot(2, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Imagen Original')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.plot(hist_original, color='black')
plt.title('Histograma Original')
plt.xlim([0, 256])

plt.subplot(2, 2, 3)
plt.imshow(img_eq, cmap='gray')
plt.title('Imagen Ecualizada')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.plot(hist_eq, color='black')
plt.title('Histograma Ecualizado')
plt.xlim([0, 256])

plt.tight_layout()
plt.show()
```

## COMENTARIOS:

Esta práctica consiste en usar la imagen que obtuve de la práctica 2, la cuál al principio no pude localizar, por lo que use una que tomé en este momento con el mismo código de la práctica 2, ya después pasado un tiempo pude encontrar la imagen de la práctica 2. Por eso adjunto las dos imágenes.

Consiste en que en una sola ventana se imprima la imagen que le adjuntamos, la imagen ecualizada y los histogramas de ambas imágenes, y como no encontré un video explicando esto, le pedí a chat que me ayudara con el código, pero que me diera una explicación exhaustiva de cada parte del mismo y el por que de cada cosa, aquí pongo la información

```
import numpy as np
```

```
import cv2
```

- ♦ ¿Qué hace esto?

numpy se importa como np: aunque en este código no se está usando directamente, numpy es una biblioteca útil para manejar matrices (imágenes son matrices).

cv2 es la biblioteca OpenCV, que permite trabajar con procesamiento de imágenes y video.

```
cap = cv2.VideoCapture(0)
```

- ♦ ¿Qué hace esto?

Abre la cámara web del sistema.

El parámetro 0 indica que se usará la cámara predeterminada. Si tuvieras varias, puedes probar con 1, 2, etc.

```
while(True):
```

- ♦ ¿Qué hace esto?

Inicia un bucle infinito para capturar continuamente fotogramas de la cámara (hasta que el usuario presione una tecla para salir).

```
    ret, frame = cap.read()
```

- ♦ ¿Qué hace esto?

Captura un fotograma (imagen) de la cámara.

ret es un valor booleano que indica si la captura fue exitosa.

frame es la imagen capturada (en color, por defecto).

```
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

- ♦ ¿Qué hace esto?

Convierte la imagen de color BGR (Blue, Green, Red) a escala de grises.

Esto es útil para simplificar el procesamiento, reducir memoria o aplicar filtros más rápido.

```
cv2.imshow('frame', gray)
```

- ♦ ¿Qué hace esto?

Muestra la imagen en una ventana llamada 'frame'.

En este caso, se muestra en blanco y negro porque usamos gray.

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

- ♦ ¿Qué hace esto?

Espera 1 milisegundo por una tecla.

Si el usuario presiona la tecla 'q', el programa se sale del bucle (break).

0xFF se usa por compatibilidad entre sistemas (evita errores con cv2.waitKey en Windows vs. Linux/Mac).

```
cap.release()
```

```
cv2.destroyAllWindows()
```

- ♦ ¿Qué hace esto?

cap.release() libera la cámara, la desconecta.

cv2.destroyAllWindows() cierra cualquier ventana que se haya abierto con cv2.imshow.

## RESULTADOS:

