

# REPORTE P1

EDUARDO MARTINEZ COLLAZO

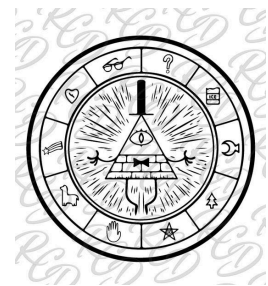
## 1. Introducción

OpenCV es una biblioteca de procesamiento de imágenes y visión artificial ampliamente utilizada en aplicaciones de análisis de imágenes. Explicaremos el funcionamiento de un código el cual carga una imagen en escala de grises y la muestra con diferentes herramientas, incluyendo OpenCV y Matplotlib.

## 2. Librerías Utilizadas

Para ejecutar el código, se requieren las siguientes librerías:

```
import cv2 # Biblioteca para el procesamiento de imágenes
import numpy as np # Biblioteca para operaciones numéricas
import matplotlib.pyplot as plt # Biblioteca para graficación
```



**Explicación:**

- **cv2**: Permite la lectura, manipulación y visualización de imágenes.
- **numpy**: Se usa para operaciones matemáticas y manipular datos de la imagen como matrices.
- **matplotlib.pyplot**: Se emplea para graficar y visualizar imágenes de forma alternativa a OpenCV.

**bill**

## 3. Carga de la Imagen

```
img = cv2.imread('fotos/bill.jpg', cv2.IMREAD_GRAYSCALE)
```

**Explicación:**

- **cv2.imread('fotos/bill.jpg', cv2.IMREAD\_GRAYSCALE)**: Carga la imagen 'bill.jpg' en escala de grises desde la carpeta 'fotos'.
- **cv2.IMREAD\_GRAYSCALE**: Convierte la imagen a escala de grises.

Otros modos de lectura de imágenes:

- **cv2.IMREAD\_COLOR = 1**: Carga la imagen en color.
- **cv2.IMREAD\_UNCHANGED = -1**: Carga la imagen en su formato original, incluyendo transparencia si la tiene.

## 4. Visualización de la Imagen con OpenCV

```
cv2.imshow('hola mundo', img) # Muestra la imagen en una ventana
cv2.waitKey(0) # Espera a que se presione una tecla
cv2.destroyAllWindows() # Cierra todas las ventanas de OpenCV
```

### Explicación:

- `cv2.imshow('hola mundo', img)`: Muestra la imagen en una ventana con el título "hola mundo".
- `cv2.waitKey(0)`: Detiene la ejecución hasta que se presione una tecla.
- `cv2.destroyAllWindows()`: Cierra todas las ventanas abiertas por OpenCV.

## 5. Visualización con Matplotlib

```
plt.imshow(img, cmap='plasma', interpolation='bicubic')
plt.plot([275, 50], [280, 50], 'c', linewidth=5) # Dibuja una línea en la imagen
plt.show()
```

### Explicación:

- `plt.imshow(img, cmap='plasma', interpolation='bicubic')`: Muestra la imagen con la paleta de colores 'plasma' y aplica interpolación bicúbica.
- `plt.plot([275, 50], [280, 50], 'c', linewidth=5)`: Dibuja una línea de color cian ('c') con un grosor de 5.
- `plt.show()`: Muestra la imagen en una ventana de Matplotlib.

## 6. Guardado de la Imagen Modificada

```
cv2.imwrite('fotos/bill_mejorado.png', img)
```

### Explicación:

- `cv2.imwrite('fotos/bill_mejorado.png', img)`: Guarda la imagen procesada en la carpeta 'fotos' con el nombre 'bill\_mejorado.png'.

## 7. Conclusión

Este código permite la carga, visualización y edición básica de una imagen con OpenCV y Matplotlib. OpenCV es una herramienta potente para el procesamiento de imágenes en tiempo real, mientras que Matplotlib ofrece opciones avanzadas de visualización. La combinación de ambas bibliotecas facilita el análisis y manipulación de imágenes para aplicaciones de visión artificial.