

## Evidencia 1 Procesamiento de imágenes

Consigna: *Aplicar al menos 3 filtros a una imagen utilizando OpenCV o PIL.*

A continuación se detalla el paso a paso para utilizar el código:

1. Cargar las librerías e inicializar las variables: Solo se debe ejecutar el código.
2. Cargar la imagen: Al ejecutar el código se habilitará la opción para cargar una imagen desde el dispositivo que se esté ejecutando, elegir la imagen a utilizar y al cargarla se mostrará y se detallará la información de la misma.
3. Menú de modificaciones: Al ejecutar el código se mostrará un menú para elegir la modificación a realizar. Luego de completada la modificación se indicará el nombre de la imagen creada y se podrá realizar otra modificación ejecutando nuevamente el código (en caso de error por no encontrar el archivo "modificaciones.py", solicitar el mismo al creador).
4. Visualización y descarga: Al ejecutar el código se mostrarán las opciones para mostrar las imágenes o descargarlas.

```
#Importamos las librerías e inicializamos variables globales
from google.colab import files
import cv2
import matplotlib.pyplot as plt
from PIL import Image

contador_rotaciones = 1
contador_recortes = 1
contador_blur = 1
contador_bordes = 1
contador_binarias = 1
contador_canny = 1
imagenes_rotadas = []
imagenes_recortadas = []
imagenes_desenfocadas = []
imagenes_bordes = []
imagenes_binarias = []
imagenes_canny = []

#Cargamos la imagen
img_cv2 = None

uploaded = files.upload()
for filename in uploaded.keys():
    ruta = f'/content/{filename}'
    img = cv2.imread(ruta)

    if img is None:
        print("Error: No se pudo cargar la imagen. Verifica la ruta del archivo.")
    else:
        img_cv2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # Mostrar imagen
        plt.imshow(img_cv2)
        plt.title("Imagen Original")
        plt.axis('off')
        plt.show()

        # Mostrar detalles
        print(f"Imagen '{filename}' cargada correctamente.")
        print(f"Dimensiones: {img_cv2.shape}") # (alto, ancho, canales)
        print(f"Tipo de dato: {img_cv2.dtype}")
```



Elegir archivos Tommas.jpeg

- **Tommas.jpeg**(image/jpeg) - 151536 bytes, last modified: 18/4/2025 - 100% done
- Saving Tommas.jpeg to Tommas.jpeg

Imagen Original



Imagen 'Tommas.jpeg' cargada correctamente.  
 Dimensiones: (899, 1599, 3)  
 Tipo de dato: uint8

```
#Menu de modificaciones
import importlib
import modificaciones
importlib.reload(modificaciones)

print("¿Qué desea hacer?:")
print("1. Rotar la imagen")
print("2. Recortar la imagen")
print("3. Aplicar desenfoque gaussiano")
print("4. Detección de bordes")
print("5. Binarización")
print("6. Detección de bordes con Canny")
print("7. Salir")

opcion = int(input("Indique el número de la opción: "))

if opcion == 1:
    modificaciones.rotar_imagen(img_cv2, imagenes_rotadas, contador_rotaciones)
    contador_rotaciones += 1
elif opcion == 2:
    modificaciones.recortar_imagen(img_cv2, imagenes_recortadas, contador_recortes)
    contador_recortes += 1
elif opcion == 3:
    modificaciones.desenfoque_gaussiano(img_cv2, imagenes_desenfocadas, contador_blur)
    contador_blur += 1
elif opcion == 4:
    modificaciones.detectar_bordes(img_cv2, imagenes_bordes, contador_bordes)
    contador_bordes += 1
elif opcion == 5:
    modificaciones.binarizar_imagen(img_cv2, imagenes_binarias, contador_binarias)
    contador_binarias += 1
elif opcion == 6:
    modificaciones.bordes_canny(img_cv2, imagenes_canny, contador_canny)
    contador_canny += 1
elif opcion == 7:
    modificaciones.saludo()
else:
    print("Opción no válida.")

print("\nEjecute nuevamente para realizar otra modificación a la imagen original.")
```



```
¿Qué desea hacer?:
1. Rotar la imagen
2. Recortar la imagen
3. Aplicar desenfoque gaussiano
4. Detección de bordes
5. Binarización
```

```
6. Detección de bordes con Canny
7. Salir
Indique el número de la opción: 7
```

-----

Gracias por utilizar el script! :)

Hasta la próxima

-----

Ejecute nuevamente para realizar otra modificación a la imagen original.

```
#Visualización y descarga
import importlib
import resultados
importlib.reload(resultados)

print("Seleccione la opción deseada:")
print("1. Mostrar las imágenes")
print("2. Descargar las imágenes")
opcion = int(input("Indique el número de la opción: "))

if opcion == 1:
    resultados.mostrar_resultados(img_cv2, contador_rotaciones, contador_recortes, contador_blur, contador_bordes, contador_desenfocadas)
elif opcion == 2:
    zip = input("Ingrese el nombre que desea colocarle al archivo zip: ")
    nombre_zip = zip + '.zip'
    resultados.exportar_zip(nombre_zip, img_cv2, imagenes_rotadas, imagenes_recortadas, imagenes_desenfocadas, imagenes_recortadas)
else:
    print("Opcion no válida")

📄 Seleccione la opción deseada:
1. Mostrar las imágenes
2. Descargar las imágenes
Indique el número de la opción: 2
Ingrese el nombre que desea colocarle al archivo zip: todo

Archivo 'todo.zip' generado y listo para descargar.
```