

# Reporte de Práctica: Detección de Bordes con OpenCV

---

## Practica#8 Detección de Bordes

Registro: 22310172

Alumno: Eduardo Correa Flores

### 1. Introducción

En esta práctica se utiliza la biblioteca OpenCV en Python para capturar imágenes en tiempo real desde la cámara web y aplicar diferentes métodos de detección de bordes: Laplaciano, Sobel y Canny. La detección de bordes es fundamental en el procesamiento de imágenes, ya que permite resaltar los contornos de los objetos presentes en una escena.

### 2. Descripción del Código

El código hace uso de la cámara web para capturar video en tiempo real mediante la función ``cv2.VideoCapture(0)``. Dentro de un bucle infinito, se lee cada cuadro (frame) y se le aplican los siguientes métodos de detección de bordes:

- Laplaciano: Detecta bordes calculando la segunda derivada de la imagen.
- SobelX: Detecta bordes en dirección horizontal usando derivadas en el eje X.
- SobelY: Detecta bordes en dirección vertical usando derivadas en el eje Y.
- Canny: Algoritmo más complejo que incluye reducción de ruido, cálculo de gradientes, supresión de no-máximos y umbralización.

Cada uno de estos resultados se muestra en una ventana separada con la función ``cv2.imshow``. El bucle se detiene cuando se presiona la tecla 'ESC'.

### 3. Resultados Esperados

Al ejecutar el programa, se abrirán varias ventanas que muestran los distintos métodos de detección de bordes aplicados a la imagen capturada por la cámara. Estas ventanas permiten observar cómo cada técnica resalta diferentes características de los bordes presentes en la escena.

### 4. Conclusión

Esta práctica permite comprender y visualizar el funcionamiento de los principales algoritmos de detección de bordes en imágenes. Cada método tiene sus ventajas y aplicaciones específicas. El uso de OpenCV facilita su implementación y análisis en tiempo real.

