

## Documentación del Proyecto

Nos basamos en estas IAs para buscar las APIs

- **ChatGpt**
- **Claude**

Las APIs que se han visto e intentado utilizar son las siguientes:

- **Roboflow**
- **LuxandCloud**

La API que se utilizó fue:

<https://console.groq.com/docs/overview>

Realizamos una aplicación que por medio de la cámara toma fotografías y detecta el objeto o personas que estén en el recuadro

**Odvin realizó investigaciones**

**Cristian Realizó estructura del código**

**Daniel Realizó la interfaz gráfica**

**Jostyn ejecutó y realizó que el programa funcionara**

**Código:**

```
import cv2

import requests

import threading

import tkinter as tk

from ultralytics import YOLO


# --- Configuración API ---

API_URL = "https://api.groq.com/openai/v1/chat/completions"

API_KEY = "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"

model = YOLO('yolov8n.pt') # Asegúrate de tenerlo descargado


# --- Función para enviar descripción a Groq ---

def enviar_a_groq(descripcion):

    payload = {

        "model": "llama3-70b-8192",

        "messages": [{"role": "user", "content": f"Describe la escena: {descripcion}"}],

        "temperature": 0.7

    }

    headers = {

        "Authorization": f"Bearer {API_KEY}",

        "Content-Type": "application/json"

    }

    response = requests.post(API_URL, headers=headers, json=payload)

    if response.ok:

        return response.json()["choices"][0]["message"]["content"]

    else:
```

```
return f"Error: {response.text}"
```

```
# --- Detección con cámara ---
```

```
capturando = False
```

```
def detectar():
```

```
    global capturando
```

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

```
    while capturando:
```

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

```
        if not ret:
```

```
            break
```

```
    results = model(frame)[0]
```

```
    etiquetas = set()
```

```
    for c in results.bboxes.cls:
```

```
        etiquetas.add(model.names[int(c)])
```

```
    if etiquetas:
```

```
        descripcion = ", ".join(etiquetas)
```

```
        respuesta = enviar_a_groq(descripcion)
```

```
        print("Groq:", respuesta)
```

```
    frame_annotado = results.plot()
```

```
    cv2.imshow("Detección", frame_annotado)
```

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

```
        break
```

```
cap.release()

cv2.destroyAllWindows()

# --- Interfaz Tkinter ---

def iniciar_deteccion():

    global capturando

    capturando = True

    hilo = threading.Thread(target=detectar)

    hilo.start()

def detener_deteccion():

    global capturando

    capturando = False

# Crear ventana

ventana = tk.Tk()

ventana.title("Detección con YOLO + Groq")

btn_iniciar = tk.Button(ventana, text="Iniciar detección", command=iniciar_deteccion, bg="green",
fg="white")

btn_iniciar.pack(pady=10)

btn_detener = tk.Button(ventana, text="Detener", command=detener_deteccion, bg="red",
fg="white")

btn_detener.pack(pady=10)

ventana.mainloop()
```