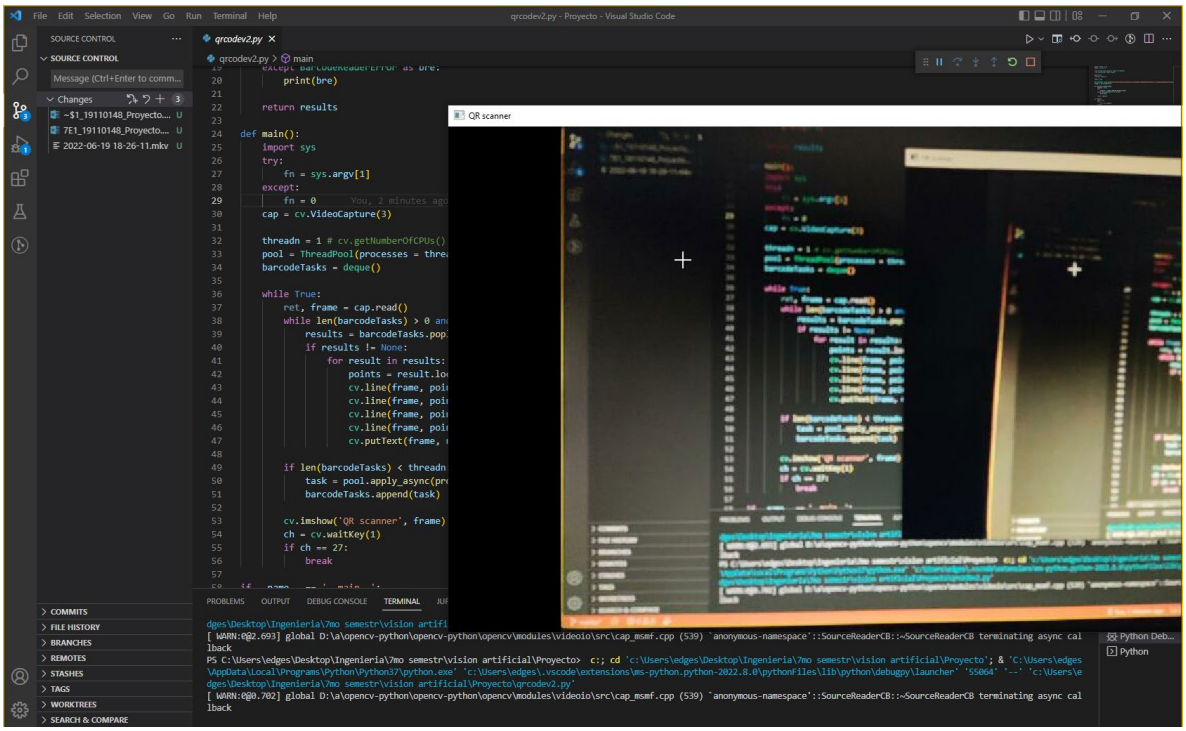
MANUAL DE USUARIO PROYECTO

PEDRO MIGUEL ELGUERA MORA 19110148

CETI COLOMOS VISION ARTIFICIAL 7E1

MANUAL DE USUARIO

EVIDENCIA



Esta es la vista principal de la aplicación.

La aplicación esta a la espera de un código QR para leer, cuando este se le acerque un código QR escribir el mensaje codificado arriba del código QR con letras rojas

Git:

https://github.com/PedroElgueraCeti/Proyecto_VisionArtificial_ReadQRCode.git

Code:

```
#Pedro Miguel Elguera Mora 19110148

import numpy as np
import cv2 as cv

from multiprocessing.pool import ThreadPool
```

```
from collections import deque

import dbr
from dbr import *

import time

BarcodeReader.init_license("DLS2eyJoYW5kc2hha2VDb2RlIjoiMjAwMDAxLTE2NDk4Mjk3
OTI2MzUiLCJvcmdhbml6YXRpb25JRCI6IjIwMDAwMSIsInNlc3Npb25QYXNzd29yZCI6IndTcGR6
Vm05WDJrcEQ5YUoifQ==")
reader = BarcodeReader()

def process_frame(frame):
    results = None
    try:
        results = reader.decode_buffer(frame)
    except BarcodeReaderError as bre:
        print(bre)

    return results

def main():
    import sys
    try:
        fn = sys.argv[1]
    except:
        fn = 0
    cap = cv.VideoCapture(3)

    threadn = 1 # cv.getNumberOfCPUs()
    pool = ThreadPool(processes = threadn)
    barcodeTasks = deque()

    while True:
        ret, frame = cap.read()
        while len(barcodeTasks) > 0 and barcodeTasks[0].ready():
            results = barcodeTasks.popleft().get()
            if results != None:
                for result in results:
                    points = result.localization_result.localization_points
                    cv.line(frame, points[0], points[1], (0,255,0), 2)
                    cv.line(frame, points[1], points[2], (0,255,0), 2)
                    cv.line(frame, points[2], points[3], (0,255,0), 2)
                    cv.line(frame, points[3], points[0], (0,255,0), 2)
```

```
                    cv.putText(frame, result.barcode_text, points[0],
cv.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,255))

        if len(barcodeTasks) < threadn:
            task = pool.apply_async(process_frame, (frame.copy(), ))
            barcodeTasks.append(task)

        cv.imshow('QR scanner', frame)
        ch = cv.waitKey(1)
        if ch == 27:
            break

if __name__ == '__main__':
    main()
    cv.destroyAllWindows()
```