

# Quadro Mágico

João Pedro Araujo

# **1. Concepção da Ideia**

**Desenvolver um aplicativo de desenho em tempo real usando reconhecimento de mãos.**

## 2. Requisitos e Ferramentas

### **Requisitos:**

Câmera para capturar o vídeo em tempo real.

Detecção de mãos e dedos.

Capacidade de desenhar na tela baseado na posição dos dedos.

Biblioteca para processamento de imagem em tempo real.

### **Ferramentas:**

Python

OpenCV para captura e processamento de vídeo.

Biblioteca cvzone para simplificar o uso do módulo de detecção de mãos do  
Mediapipe.

# 3. Estruturação do Projeto

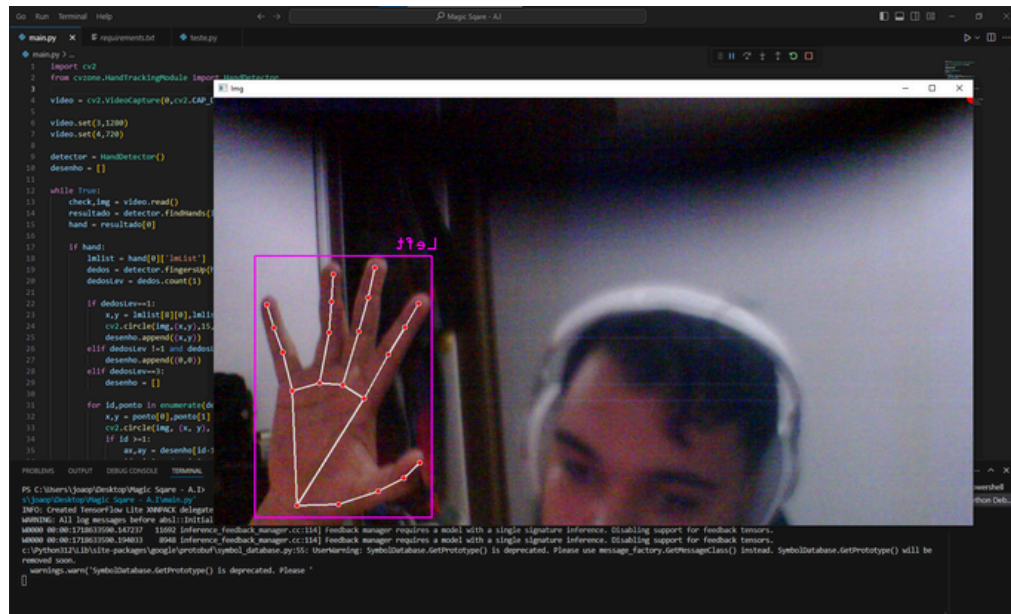
**Passo 1: Inicializar a Captura de Vídeo**

**Passo 2: Detecção de Mão**

**Passo 3: Reconhecimento de Gestos**

**Passo 4: Desenho na Tela**

**Passo 5: Interface Gráfica**



## 4. Desenvolvimento do Código

### Passo 1: Configuração da Captura de Vídeo

```
import cv2
from cvzone.HandTrackingModule import HandDetector

video = cv2.VideoCapture(0, cv2.CAP_DSHOW)
video.set(3, 1280) # Largura
video.set(4, 720) # Altura
```

### Passo 2: Inicialização do Detector de Mãos

```
detector = HandDetector()
desenho = []
```

# Passo 3: Loop Principal para Processamento em Tempo Real

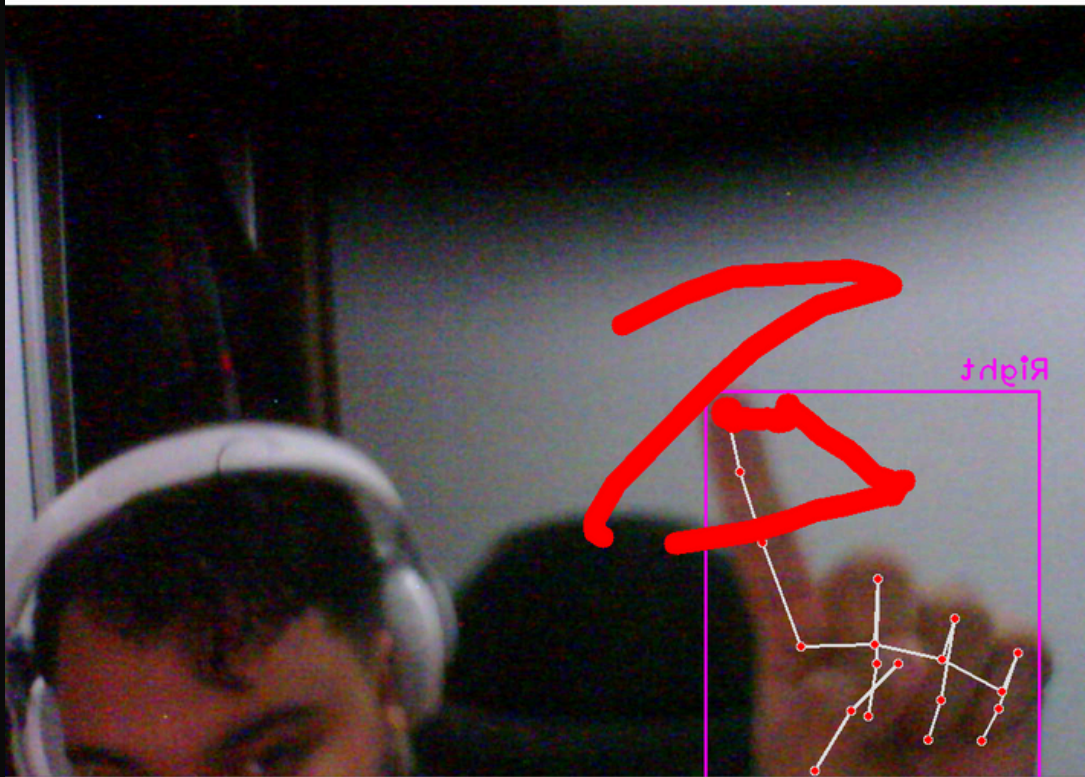
```
while True:
    check, img = video.read()
    resultado = detector.findHands(img, draw=True)
    hand = resultado[0]

    if hand:
        lmlist = hand[0]['lmList']
        dedos = detector.fingersUp(hand[0])
        dedosLev = dedos.count(1)

        if dedosLev == 1:
            x, y = lmlist[8][0], lmlist[8][1] # Coordenadas do dedo indicador
            cv2.circle(img, (x, y), 15, (0, 0, 255), cv2.FILLED)
            desenho.append((x, y))
        elif dedosLev != 1 and dedosLev != 3:
            desenho.append((0, 0))
        elif dedosLev == 3:
            desenho = []

        for id, ponto in enumerate(desenho):
            x, y = ponto[0], ponto[1]
            cv2.circle(img, (x, y), 10, (0, 0, 255), cv2.FILLED)
            if id >= 1:
                ax, ay = desenho[id - 1][0], desenho[id - 1][1]
                if x != 0 and ax != 0:
                    cv2.line(img, (x, y), (ax, ay), (0, 0, 255), 20)

    imgFlip = cv2.flip(img, 1)
    cv2.imshow('Img', imgFlip)
    if cv2.waitKey(1) == 27:
        break
```



**FIM.**