



Examen parcial práctico: IHC

Saul Andersson Rojas Coila



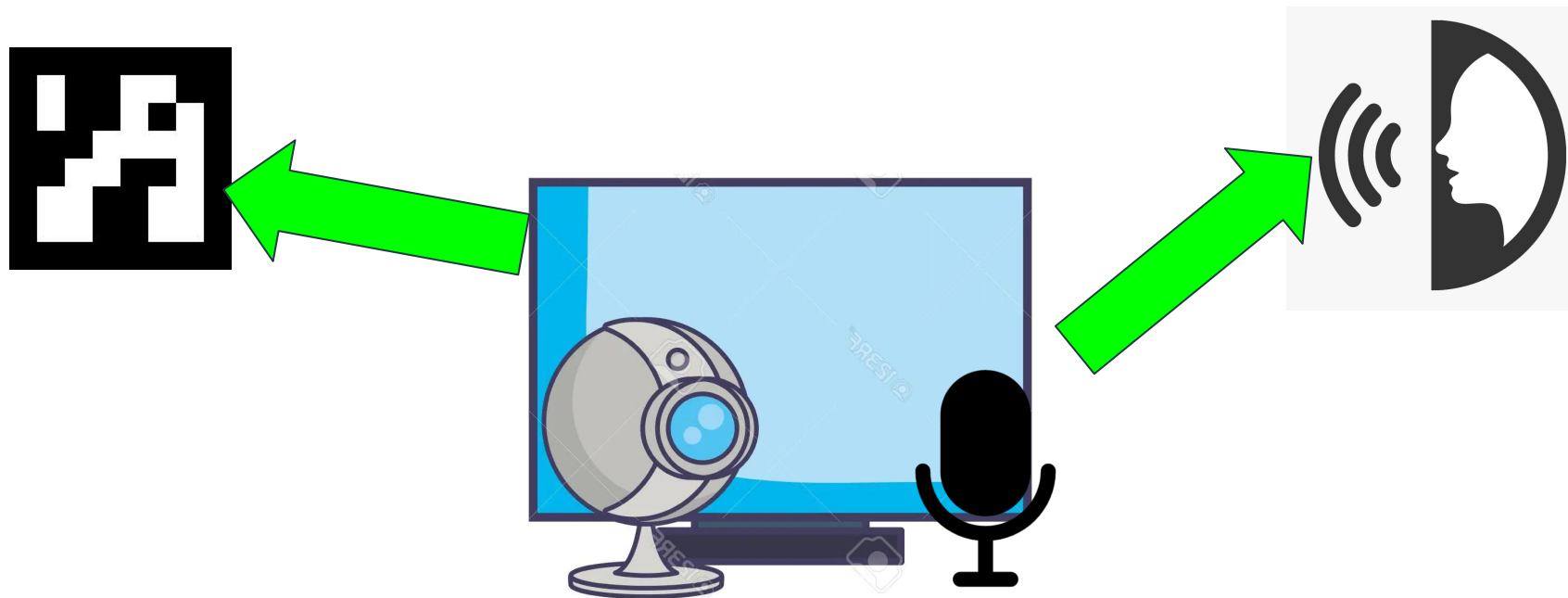
Índice

1. Idea general
 - a. Análisis de especificidad
2. Tecnologías
3. Evolución
 - a. Aruco
 - b. Voz
 - c. GUI
 - d. Juego
4. Conclusión
5. Referencias

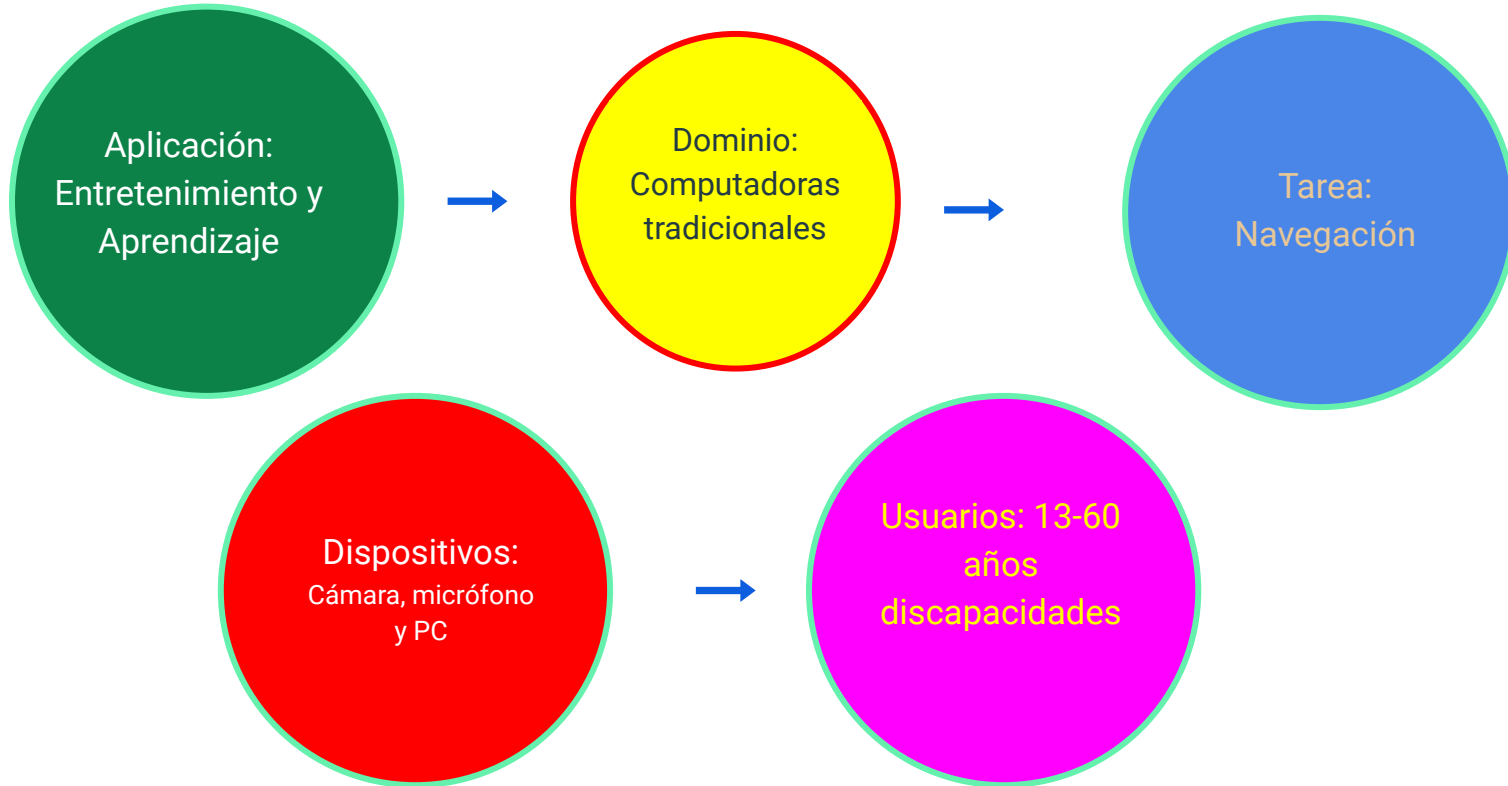


Idea General

Idea general



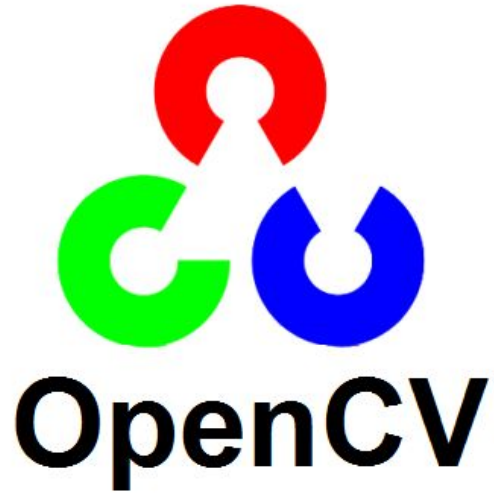
Idea general - Análisis de especificidad





Tecnologías

Tecnologías



Tecnologías - Versionamiento





- Windows 10, x64
- Conda 4.12.0
- Python 3.7.13
- OpenCV 4.5.4
- DeepSpeech 0.9.13



Evolución

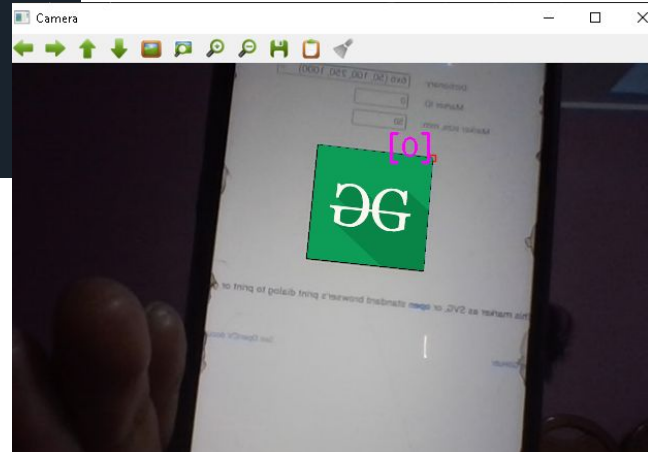
Evolución

Planificación Proyecto IHC ...

 Período	 Tareas	 Estado	 Semana
Primera Semana	- Detección de clibración por cámara con los marcadores ARUCO.	hecho	April 4, 2022 → April 10, 2022
Segunda Semana	- Mas pruebas con los marcadores ARUCO. - Detección de voz en computadora	hecho	April 11, 2022 → April 17, 2022
Tercera Semana	- Integración de voz y marcadores ARUCO,	haciendolo	April 18, 2022 → April 24, 2022
Cuarta Semana	- Combinación de ambas técnicas con un videojuego. - Conexión con windows.	haciendolo	April 25, 2022 → May 1, 2022
Quinta Semana	- Combinación de técnicas con el videojuego (mejora).	haciendolo	May 2, 2022 → May 8, 2022

Evolución - Aruco

```
60 def main():
61     cap = cv2.VideoCapture(0)
62     #loadAugImages('imgs')
63     imgAug = cv2.imread('imgs/geeks14.png')
64     while True:
65         success, frame = cap.read()
66         arucoFound = findArucoMarkers(frame)
67         # loop through all the markers and augment each one
68         if len(arucoFound[0]) != 0:
69             for bbox, id in zip(arucoFound[0], arucoFound[1]):
70                 frame = augmentAruco(bbox, id, frame, imgAug, drawId=True)
71         # flipping the frame to avoid reversed capture on camera
72         frame = cv2.flip(frame, 1)
73         cv2.imshow("Camera", frame)
74
75         if cv2.waitKey(1) == ord('q'):
76             break
77     cap.release()
78     cv2.destroyAllWindows()
79
```



Evolución - Aruco

```
class Calibration():
    def __init__(self):
        self.tablero = (7, 7) # tablero 8x8, se suma +1 para el borde
        self.frameSize = (720, 640)
        # Criterio
        self.criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30,
                        0.001)
        # Preparamos los puntos del tablero
        self.points_obj = np.zeros((self.tablero[0] * self.tablero[1], 3), np.float32)
        self.points_obj[:, :2] = np.mgrid[0:self.tablero[0], 0:self.tablero[1]].T.reshape(-1, 2)
        # Preparamos las listas para almacenar los puntos del mundo real y de la imagen
        self.points_3d = []
        self.points_img = []
```

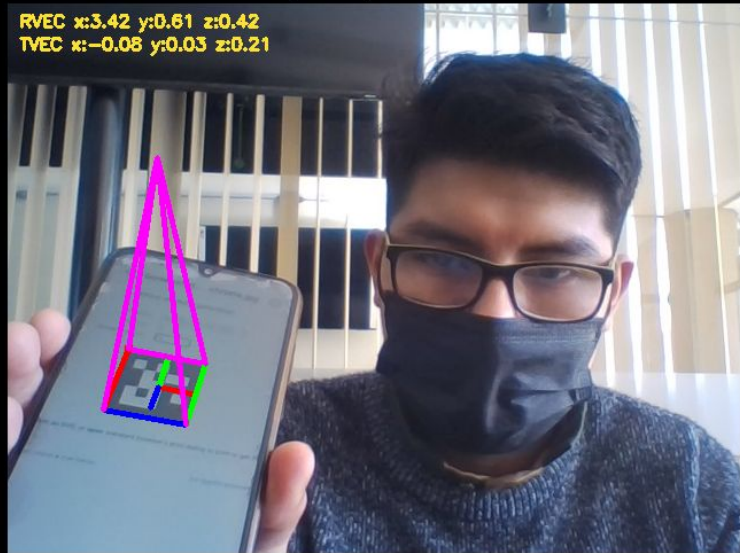
Evolución - Aruco

Calibración de la cámara

Calibrador

Toma 6 fotos del tablero 8x8 en diferentes posiciones para calibrar la cámara.

RVEC $x:3.42$ $y:0.61$ $z:0.42$
TVEC $x:-0.08$ $y:0.03$ $z:0.21$



Record

Stop

Exit

Calibración de la cámara

Calibrador

Toma 6 fotos del tablero 8x8 en diferentes posiciones para calibrar la cámara.



Record

Shot

Stop

Exit

Evolución - Aruco

```
if len(corners) != 0:
    center_aruco = np.sum(corners, axis=0) // 4

    m_width = scr_width / frame_width
    m_height = scr_height / frame_height

    tr_pos = np.array((m_width * center_aruco[0], m_height * center_aruco[1]))
    curr_pos = np.array(mouse.get_position())

    offset = tr_pos - curr_pos

    mouse.move(offset[0], offset[1], absolute=False, duration=0)

img_bytes = cv2.imencode('.png', aruco_detected)[1].tobytes() # ditto
window['image'].update(data=img_bytes)
```

Evolución - Voz

```
54 import speech_recognition as sr
55 def voice_listener(text):
56     recognizer = sr.Recognizer()
57     recognizer.energy_threshold = 4000
58     recognizer.pause_threshold = 0.5
59     microphone = sr.Microphone()
60     # locker = threading.Lock()
61     while True:
62         guess = recognize_speech_from_mic(recognizer, microphone)
63         if guess["transcription"]:
64             text = guess["transcription"].lower()
65             print("You said: ", text)
66             if "click" in text:
67                 print("left click pressed")
68                 mouse.click('left')
69             if not guess["success"]:
70                 # there was an error
71                 print("ERROR: {}".format(guess["error"]))
```

Muy lento, 5 segundos de retraso.

Evolución - Voz

```
def recognize_voice():  
    DEFAULT_SAMPLE_RATE = 16000  
    ARGS_STATIC = {  
        'device': None,  
        'file': None,  
        'model': './deepspeech-0.9.3-models.pbmm',  
        'nosspinner': False,  
        'rate': DEFAULT_SAMPLE_RATE,  
        'savewav': None,  
        'scorer': './deepspeech-0.9.3-models.scorer',  
        'vad_aggressiveness': 3,  
    }  
    config_streaming_args(ARGS_STATIC)
```

DeepSpeech es más rápido, pero no reconoce bien la voz

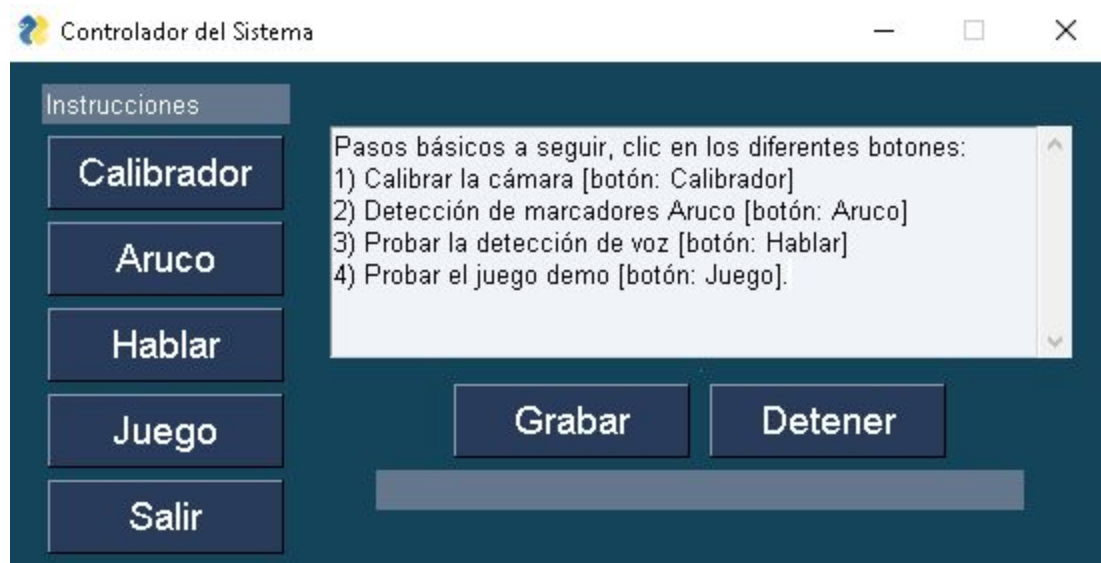
```
[ WARN:0] global ..\modules\videoio\src\cap_gstreamer.cpp (616) cv::GStreamerCapture::isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline ha  
created  
Detected:  
Detected: that  
Detected:
```


Evolución - GUI

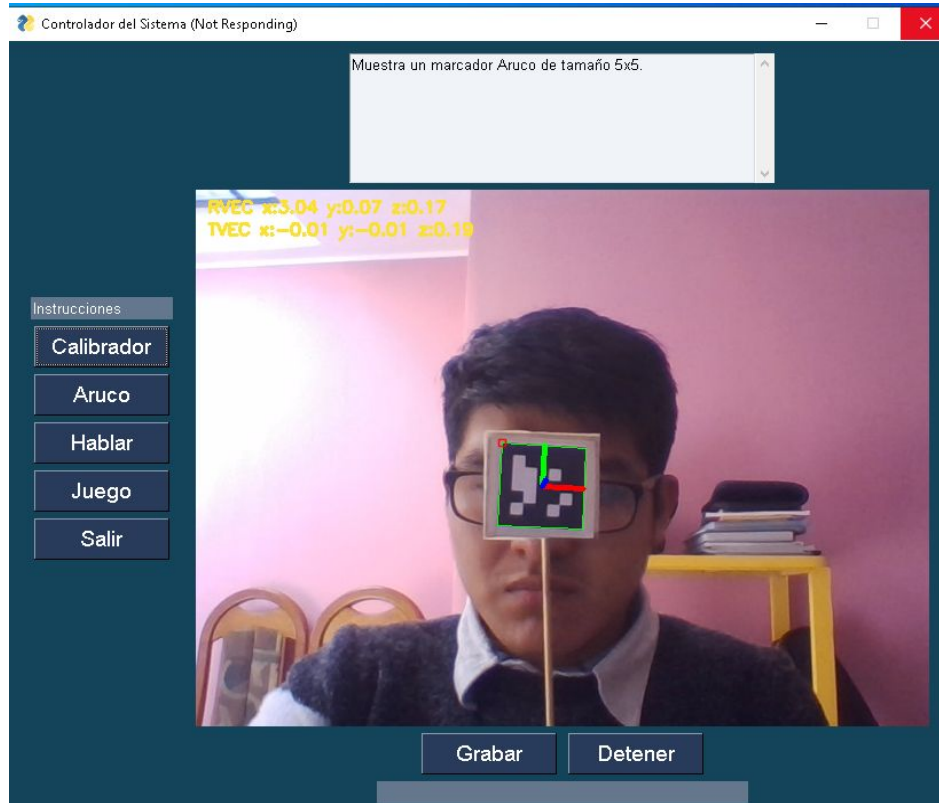
```
col1 = [
    [sg.Text('Instrucciones', size=(15, 1), justification='left', font='Helvetica 10')],
    [sg.Button('Calibrador', size=(10, 1), font='Helvetica 14')],
    [sg.Button('Aruco', size=(10, 1), font='Helvetica 14')],
    [sg.Button('Hablar', size=(10, 1), font='Helvetica 14')],
    [sg.Button('Juego', size=(10, 1), font='Helvetica 14')],
    [sg.Button('Salir', size=(10, 1), font='Helvetica 14')],
]

col2 = [
    [sg.Multiline('Pasos básicos a seguir, clic en los diferentes botones: \n'
        '\t1) Calibrar la cámara [botón: Calibrador]\n'
        '\t2) Detección de marcadores Aruco [botón: Aruco]\n'
        '\t3) Probar la detección de voz [botón: Hablar]\n'
        '\t4) Probar el juego demo [botón: Juego].',
        size=(50, 7), justification='left', font='Helvetica 10', key='simple_instruction')],
    [sg.Image(key='image')],
    [
        sg.Button('Grabar', size=(10, 1), font='Helvetica 14'),
        sg.Button('Detener', size=(10, 1), font='Any 14'),
    ],
    [sg.Text('', size=(40, 1), justification='left', font='Helvetica 10', key='Logs')],
]
```

Evolución - GUI



Evolución - GUI



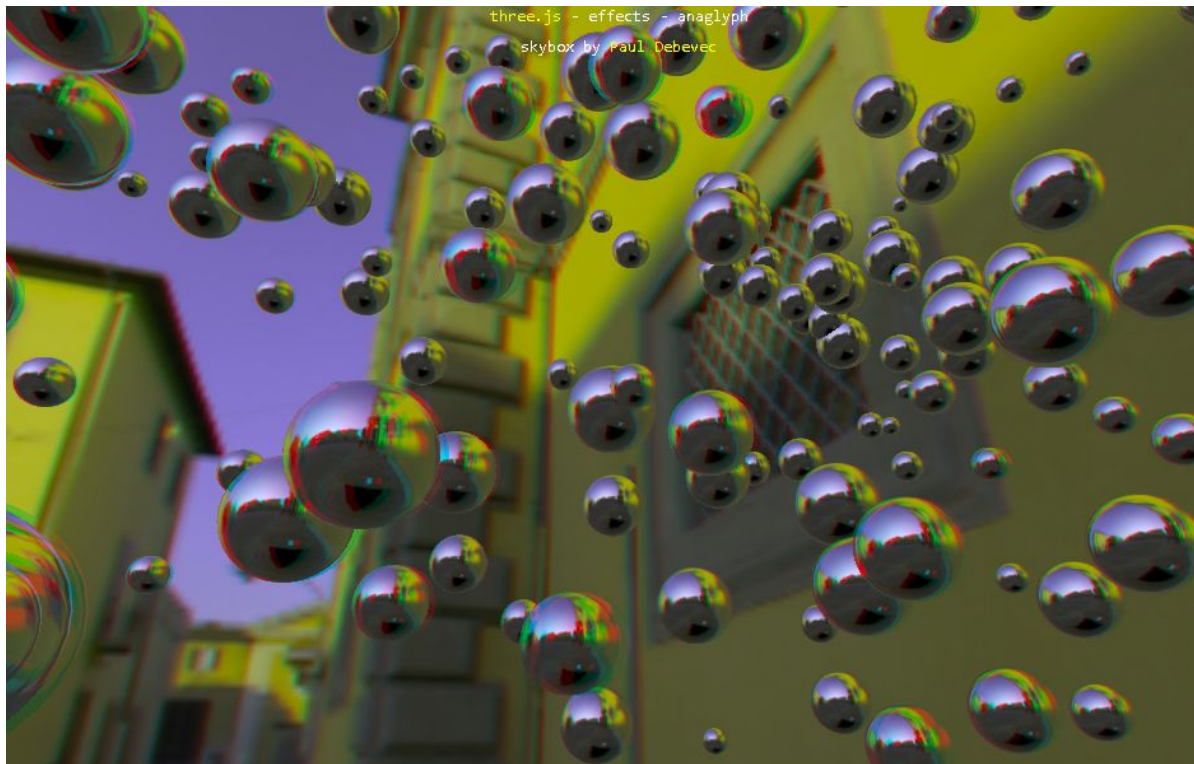
Evolución - GUI



Evolución - Juego



Evolución - Juego





Conclusión

Conclusión

Se ha visto una integración de dos tecnologías: detección de marcadores arUco y detección de voz, aún faltan cosas por mejorar. como la elección de un lenguaje de programación más robusto para estas tareas como C++.

Referencias

https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

<https://github.com/mozilla/DeepSpeech>

https://www.youtube.com/watch?v=pwtiJ5Csval&ab_channel=AprendeIngenia