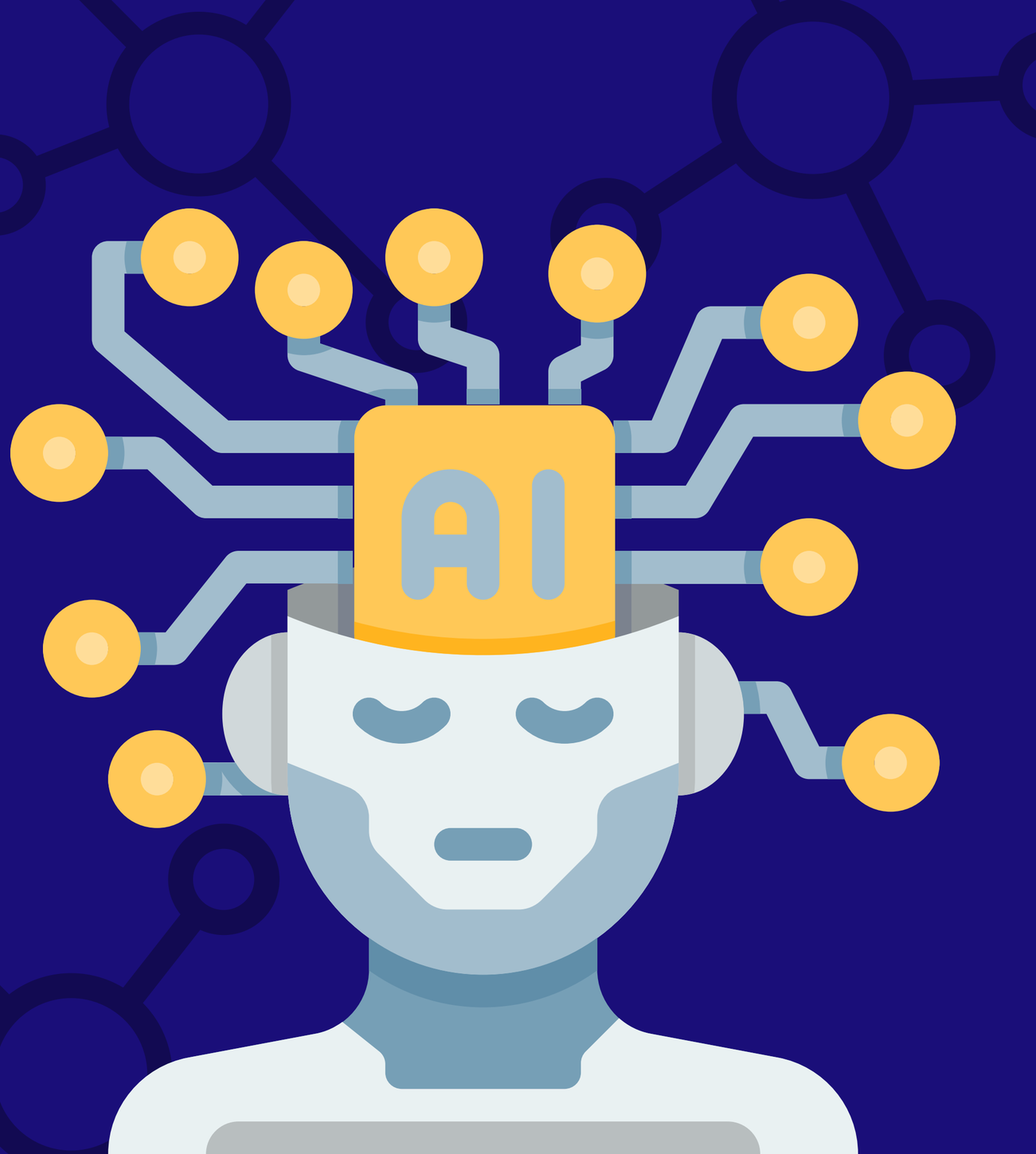
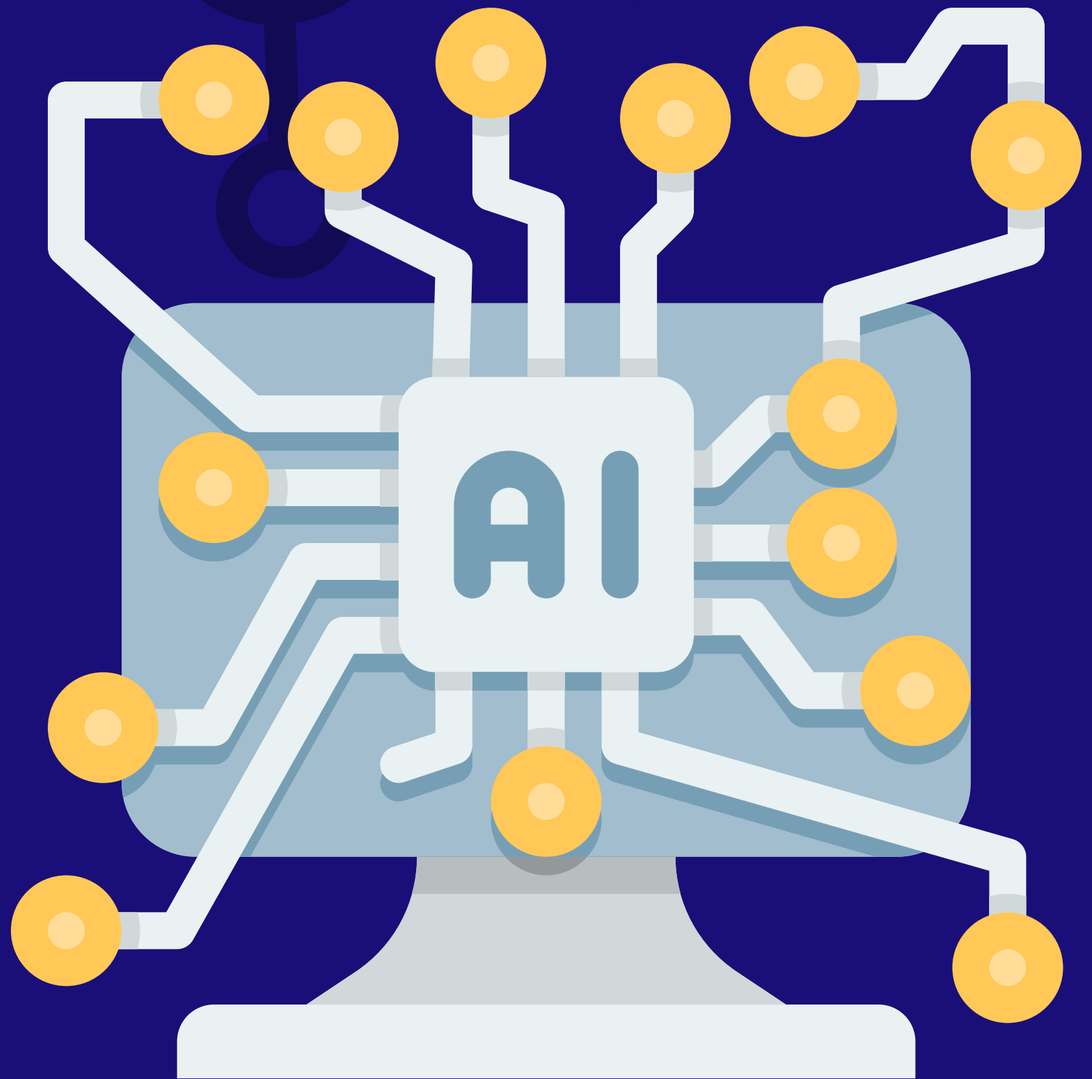
ProyectoIA-Hands

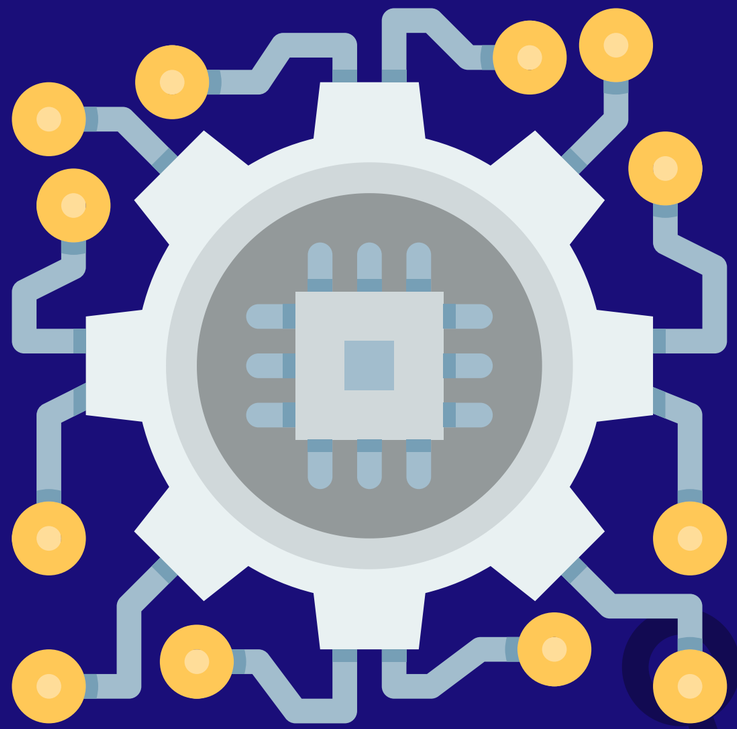
Traductor y lector con voz de lenguaje de señas en tiempo real utilizando visión artificial y Machine Learning para el reconocimiento de cada seña

Gael Navarrete Plancarte - 222791214
Diego Arath Maldonado Cárdenas - 219516075
Aldo García Castañeda - 222790927





Índice



FASE 1 - RECOPILACIÓN

Recopilamos toda la información necesaria para poder desarrollarnos de manera adecuada en el tema de machine learning

FASE 2 - MACHINE LEARNING

Entrenamos a base de imagenes la inteligencia artifical para que pudiera reconocer el abecedario del lenguaje de señas, por lo que reconoce tanto en imagen como en video.

FASE 3 - DESPLIEGUE

Con ayuda de una API logramos subir el código de Python para poder conectarlo con nuestro frontend y mantener nuestra máquina sin tener que volverla a entrenar y manteniendo el código que realizamos en un inicio

REQUERIMIENTOS

Recomiendo altamente ejecutar los scripts en Python 3.9, pues algunas de las librerías parecen dar problemas en versiones anteriores.

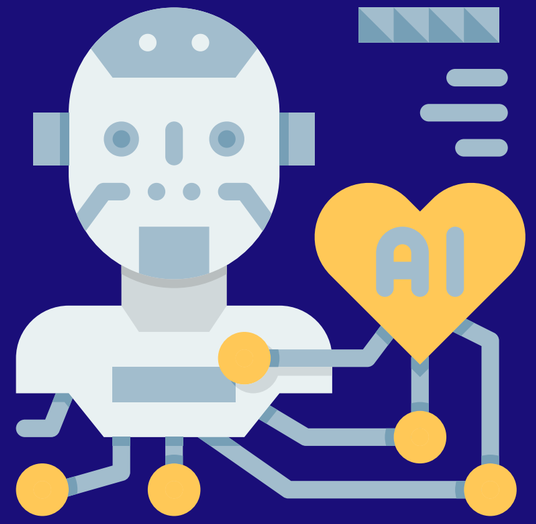
Las librerias a instalar y que fueron utilizadas son:

1. opencv-python,
2. mediapipe,
3. scikit-learn,
4. pyttsx3

Las demás librerías suelen estar precargadas

EXPLICACIÓN DEL CÓDIGO

El repositorio ya incluye un modelo entrenado con las señas incluidas en el ASL ALPHABET, por lo que solo hace falta descargar la carpeta del repositorio, extraerlo y crear una nueva carpeta que contenga el 'model.p' y el 'traductor_main.py' y desde ahí ejecutar dicho script, sin embargo, incluyo las instrucciones del proceso completo
El repositorio incluye 4 archivos indispensables para crear un dataset, entrenar un modelo de IA y posteriormente usar este modelo para hacer predicciones en las señas que se muestran en cámara

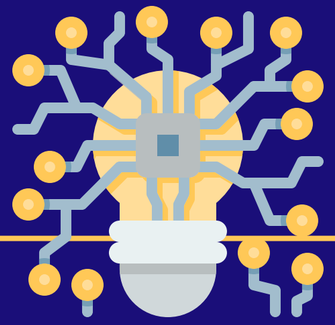


1 recolectar_imagenes.py

2 crear_dataset.py

3 entrenar_modelo.py

4 traductor_main.py



recolectar_imagenes.py

Es el primer archivo a ejecutar, al hacerlo veremos que se abre una ventana que usa la cámara para recolectar las imagenes de cada clase (cada clase siendo una seña en el lenguaje de señas), en el código está señalada que linea hay que modificar para elegir el número de imagenes a guardar, y el número de clases (señas) a registrar. El ciclo de guardado de imagenes empieza presionando `q‘

Una vez terminado el proceso obtendremos una carpeta llamada 'data' con subcarpetas para cada clase

datasets y entrenamiento

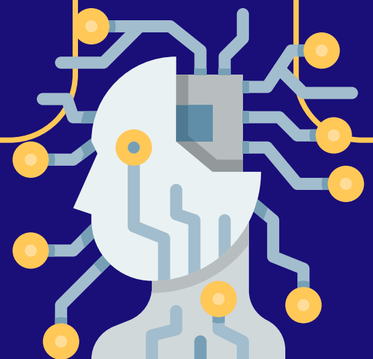
CREAR_DATASET.PY

Una vez creada la carpeta 'data' con las imágenes dentro, no hace falta más que ejecutar este nuevo script, que tomará toda la información de la carpeta para crear un archivo binario llamado 'data.pickle', este conteniene toda la información necesaria para entrenar a nuestro modelo.

Y

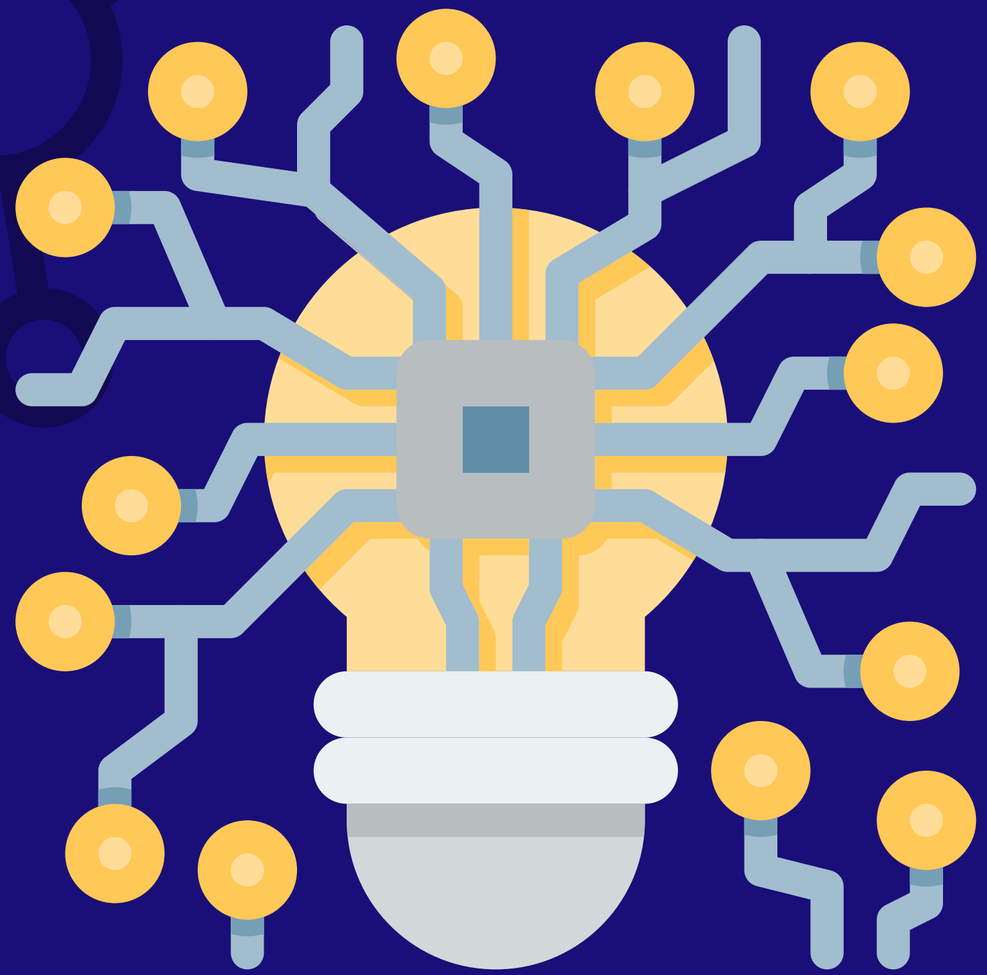
ENTRENAR_MODELO.PY

Ahora contamos con el archivo 'data.pickle', por lo que solo hace falta ejecutar este script para entrenar y testear el modelo encargado de la detección de señas. Al ejecutar el código terminaremos viendo un mensaje en consola que indica el porcentaje de datos detectado correctamente



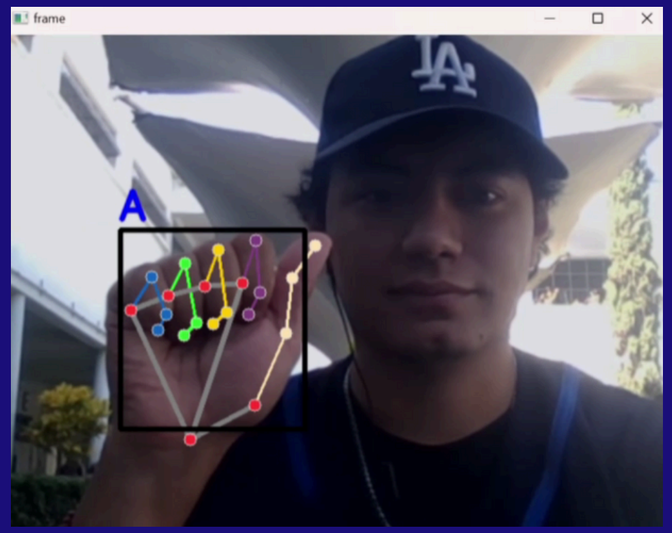
traductor_main.py

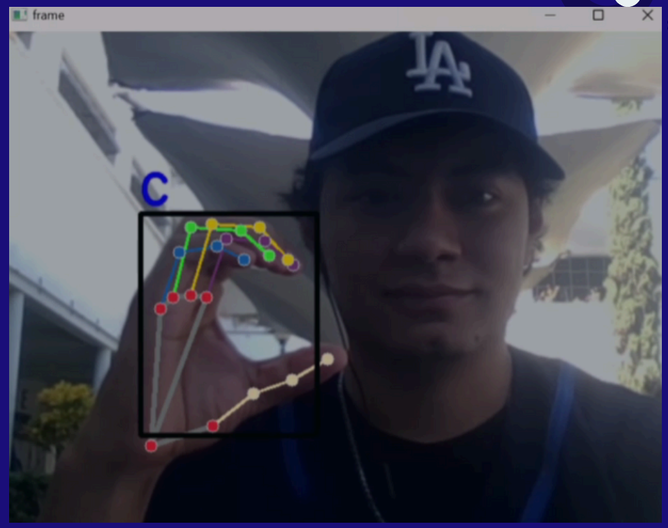
Es el momento de ejecutar el ultimo script!, este archivo ejecutará de nueva cuenta una ventana que activa la cámara integrada de tu computador y está listo para detectar las señas que le enseñaste al modelo incluso reproducidas con voz

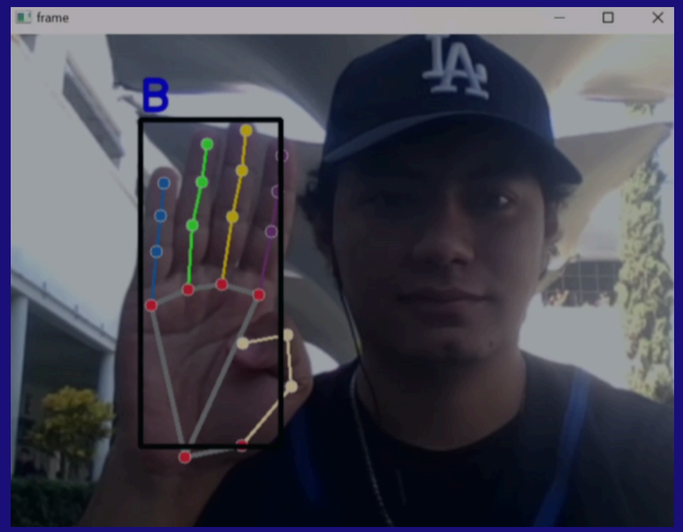


Sí descargas el repositorio y deseas utilizar el traductor que yo creé, sólo es necesario que descargues 'model.p' y traductor_main.py, los mantengas en una sola carpeta y ejecutes el script de python desde ahí!

Muestras del trabajo en tiempo real

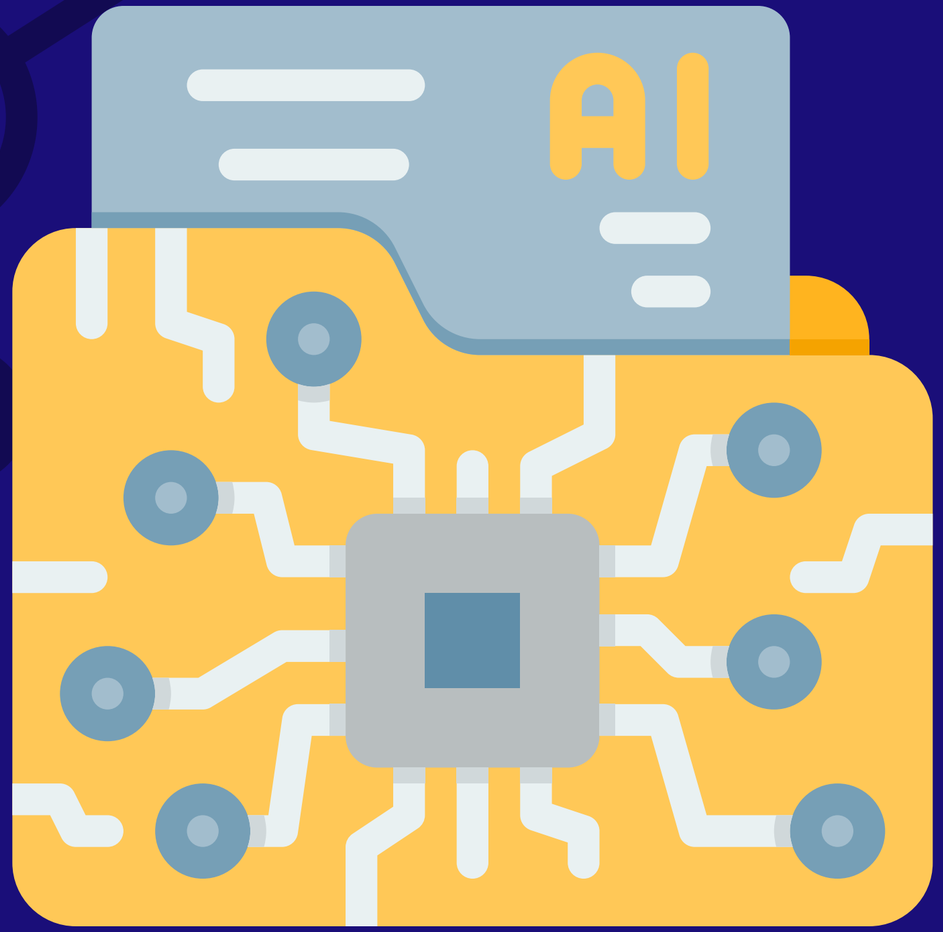




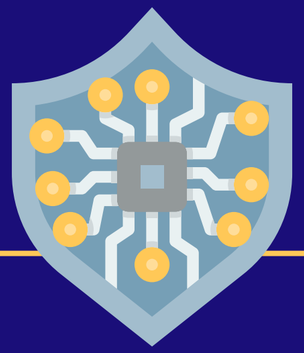


API con FastAPI

Subimos con ayuda FastAPI y de Render nuestro archivo generado en python pra poder conectarlo a nuestro frontend



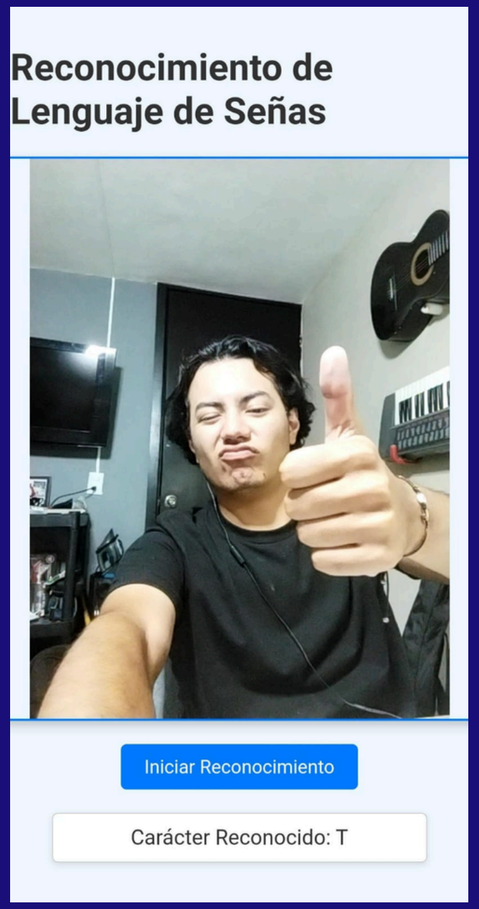
Service name	Status	Type	Runtime	Region	Last deployed ↑
proyecto-manos	✓ Deployed	Web Service	Python 3	Oregon	an hour ago ···

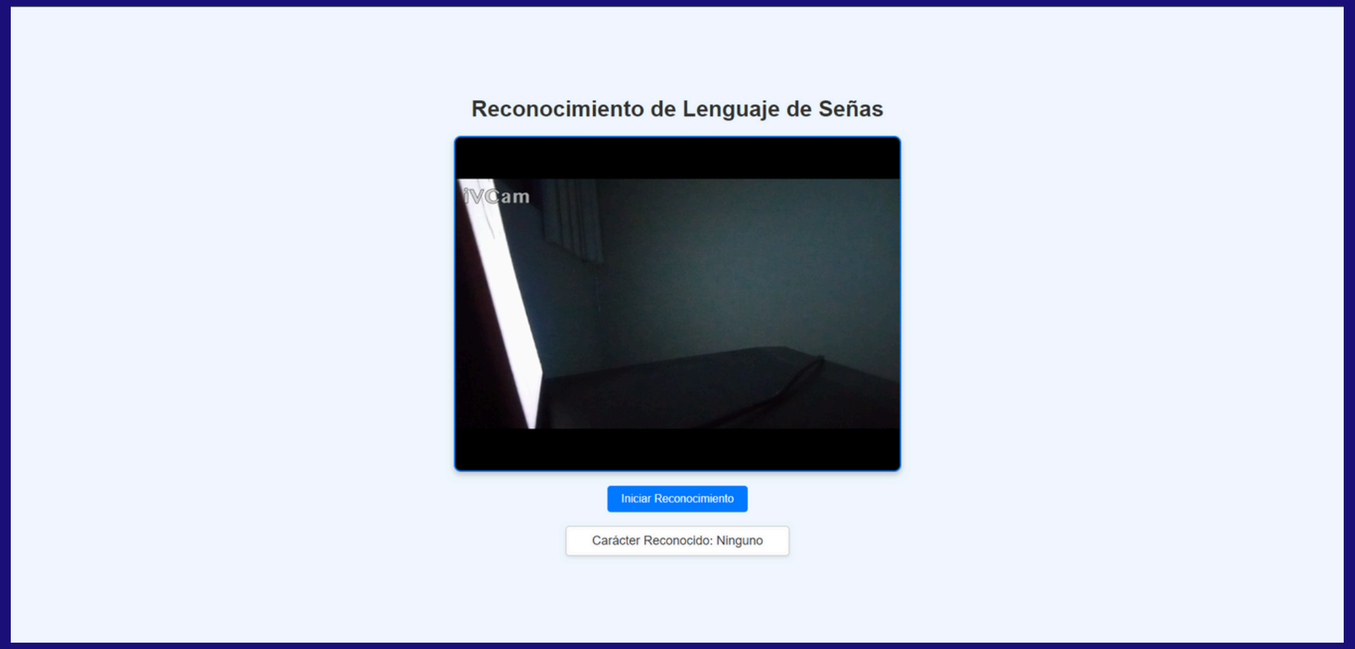


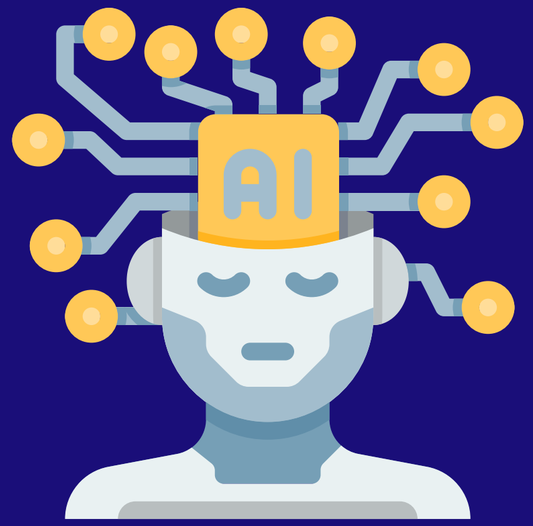
PROBLEMAS

Tuvimos varios problemas al momento de subir la api de render, ya que nos faltaron librerías en el archivo de requerimientos en muchas ocasiones y también tuvimos bastantes problemas con CORS, no nos dejaba acceder directamente a la API usando otra ip.

Muestras de la página ya desplegada







Gracias