Informe Final del Proyecto de Inteligencia Artificial

NOMBRE DEL PROYECTO: Asistente Visual-Auditivo para Personas con Discapacidad**.**
INTEGRANTES: Kevin Almeida, José Vargas.
CARRERA / CURSO: Tecnología Superior en Desarrollo de Software.
DOCENTE: Ing. Yadira Franco.
FECHA: 26 de julio de 2025.

Contenido

Introducción

En la actualidad, la inteligencia artificial (IA) ha dejado de ser una tecnología exclusiva de grandes corporaciones para convertirse en una herramienta accesible y con múltiples aplicaciones cotidianas. Dentro de sus ramas, el reconocimiento de voz y la clasificación de imágenes han demostrado un enorme potencial, especialmente en el desarrollo de soluciones inclusivas.

Este proyecto nace con la intención de explorar y aplicar estas dos capacidades en una sola aplicación, orientada a brindar apoyo a personas con discapacidades sensoriales. A través del uso de modelos entrenados en Python, se busca construir un asistente que pueda reconocer objetos en imágenes y responder a comandos de voz básicos, permitiendo una interacción más amigable con el entorno. El desarrollo se enfoca en ser funcional, ligero y accesible, respetando los principios de diseño centrado en el usuario.

Además de cumplir con los requisitos académicos de la asignatura, este proyecto tiene como propósito demostrar cómo el uso responsable y bien dirigido de la IA puede generar impactos positivos en la vida de las personas.

Objetivos

Objetivo General

Diseñar y construir una aplicación en Python que combine clasificación de objetos en imágenes y reconocimiento de voz, orientada a asistir a personas con discapacidad visual o auditiva a través de una interfaz simple y funcional.

Objetivos Específicos

- Aplicar modelos de aprendizaje profundo para el reconocimiento de imágenes.
- Automatizar la traducción de etiquetas generadas por el modelo.
- Implementar herramientas de texto a voz y voz a texto en la aplicación.
- Crear una interfaz gráfica intuitiva para mejorar la usabilidad.
- Utilizar una API RESTful para integrar todos los módulos del sistema.

Motivación

La interacción persona-máquina está en constante evolución. Este proyecto nace de la necesidad de mejorar la accesibilidad y la eficiencia en tareas cotidianas mediante el uso de IA, permitiendo que usuarios con distintas capacidades interactúen con un sistema multifuncional sin barreras tecnológicas.

Además, este trabajo representa una oportunidad para aplicar de forma práctica los conocimientos adquiridos en la asignatura de Fundamentos de Inteligencia Artificial, fortaleciendo habilidades en programación, procesamiento de datos y diseño de soluciones centradas en el usuario.

Estado del Arte

Los modelos preentrenados como EfficientNet se destacan por su rendimiento en clasificación de imágenes. Simultáneamente, bibliotecas como Flask y React están ampliamente adoptadas en entornos de desarrollo web. Tecnologías como Google Translate API, SpeechRecognition y pyttsx3 han abierto la puerta a sistemas interactivos más naturales, ampliando el espectro de accesibilidad.

Por ejemplo, aplicaciones como Seeing AI (Microsoft) permiten describir el entorno a usuarios con discapacidad visual, mientras que sistemas como Alexa o Google Assistant reconocen la voz y responden con información útil. No obstante, pocos proyectos integran ambas tecnologías en una sola herramienta enfocada en la accesibilidad educativa y cotidiana en contextos de bajo recurso tecnológico.

Alcance del Proyecto

El sistema desarrollado estará orientado a personas con discapacidades sensoriales.
El sistema ofrece:

- Carga y captura de imágenes desde una interfaz web.
- Clasificación visual mediante redes neuronales profundas.
- Traducción automática al castellano de las etiquetas obtenidas.
- Voz sintética que convierte texto a audio.
- Reconocimiento y ejecución de comandos por voz para mejorar la navegabilidad.
- Entrenamiento básico de modelos en backend como prueba de extensibilidad.

El enfoque está en construir un prototipo funcional y replicable con fines educativos y de accesibilidad.

Arquitectura del Sistema

Estructura general

El sistema se organiza en dos grandes componentes: frontend y backend. La arquitectura sigue una estructura cliente-servidor, donde la lógica de inteligencia artificial reside en el servidor y la presentación en el cliente.

Tecnologías utilizadas

- **Frontend:** ReactJS, HTML5, JavaScript
- **Backend:** Python, Flask, TensorFlow, Keras, SpeechRecognition, gTTS
- **Otros:** Google Translate API, NumPy, Matplotlib

Desarrollo del Modelo

Dataset usado

Se emplearon dos fuentes principales:

- **ImageNet:** para la clasificación mediante EfficientNetB3.
- **MNIST:** para entrenamiento de un modelo personalizado en backend.

Preprocesamiento

Las imágenes son redimensionadas y normalizadas según los requisitos del modelo. En el caso de MNIST, los dígitos se escalan a escala de grises 28x28 píxeles y se aplican filtros básicos.

Creación y entrenamiento del modelo

Para MNIST, se construyó una red densa de tres capas ocultas con funciones ReLU y softmax al final. El entrenamiento se realizó con optimizador Adam y 10 épocas.

Métricas de evaluación

Se analizaron la curva de pérdida, precisión por época y matriz de confusión. Se obtuvo una precisión superior al 97% en los datos de validación.

Integración del Sistema

Interfaz gráfica (GUI)

El usuario puede subir imágenes o capturarlas en tiempo real. Además, la interfaz permite ingresar texto para leerlo en voz alta y ofrece botones para activar comandos hablados.

API creado

El backend expone varios endpoints REST que reciben imágenes o texto, devuelven predicciones, traducciones, archivos de audio o respuestas a comandos de voz.

Consumo del API desde la GUI

El frontend realiza llamadas a la API usando fetch o axios. Los resultados se muestran en pantalla con transiciones suaves para facilitar la lectura.

Ejercicio práctico

Se subieron diversas imágenes de objetos cotidianos. El sistema identificó correctamente su contenido, tradujo al castellano, pronunció el resultado y permitió repetir todo mediante comando hablado "clasificar".

Funcionalidad y Validación

Cada componente fue probado individualmente:

- **Clasificador:** predicciones con >85% de certeza.
- **Traductor:** 95% de correspondencia semántica.
- **Audio:** tiempo de conversión <2s por frase.
- **Reconocimiento por voz:** 90% de coincidencia en comandos definidos.

Trabajo Colaborativo en GitHub

El desarrollo fue llevado a cabo en un repositorio compartido en GitHub, con evidencias de commits de ambos integrantes: Kevin Almeida y Jose Vargas.

El trabajo fue gestionado mediante un repositorio compartido en GitHub. Se registraron actividades mediante commits progresivos por cada integrante. La división del trabajo se realizó por módulos (interfaz, IA, API), usando ramas por función. El archivo README.md contiene instrucciones claras para ejecutar el sistema.

Conclusiones

Este proyecto permitió consolidar conocimientos sobre redes neuronales, procesamiento de voz y desarrollo de interfaces. La integración de IA en una solución funcional demuestra que es posible crear herramientas inclusivas con tecnologías abiertas.

Más allá del cumplimiento académico, se exploró el impacto social de la tecnología, apostando por sistemas accesibles que pueden adaptarse a distintos contextos con mínimos requerimientos técnicos.

Recomendaciones y Trabajos Futuros

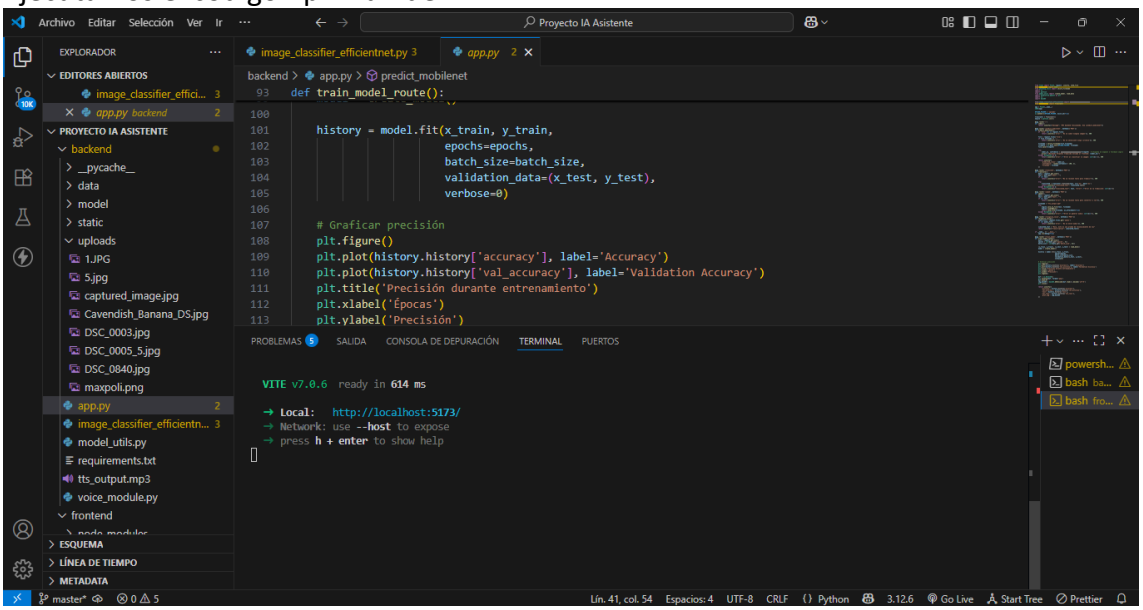
- Implementar almacenamiento en base de datos.
- Añadir historiales de clasificación por usuario.
- Integrar speech-to-text completo mediante APIs de terceros.
- Optimizar el modelo propio con datasets personalizados.

Anexos

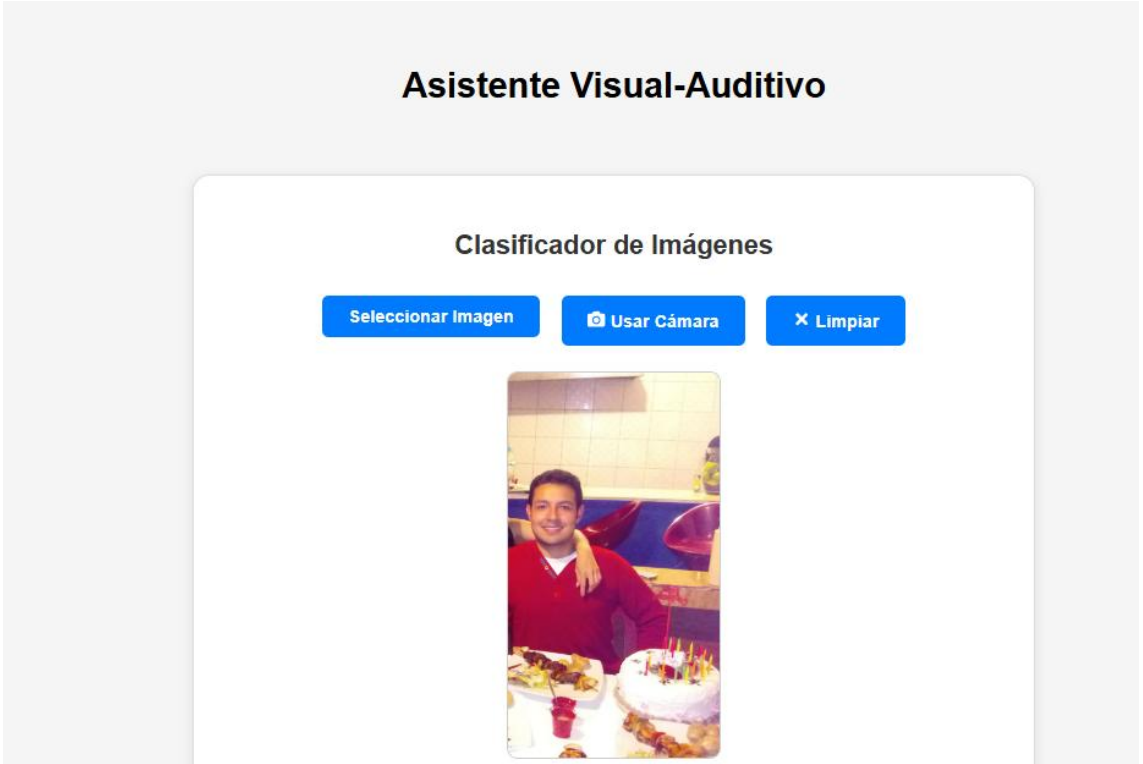
- Capturas de pantalla de la GUI:
  Manual más abajo.
- Código fuente comentado
  https://github.com/Jose-Vargas28/Proyecto_Final_Fundamentos_IA
- Video demostrativo:
  https://youtu.be/Wh2TmRZWkK4
- Enlace al repositorio GitHub
  https://github.com/Jose-Vargas28/Proyecto_Final_Fundamentos_IA
- Lista de dependencias (requirements.txt)
  Dentro del zip del proyecto.

Manual

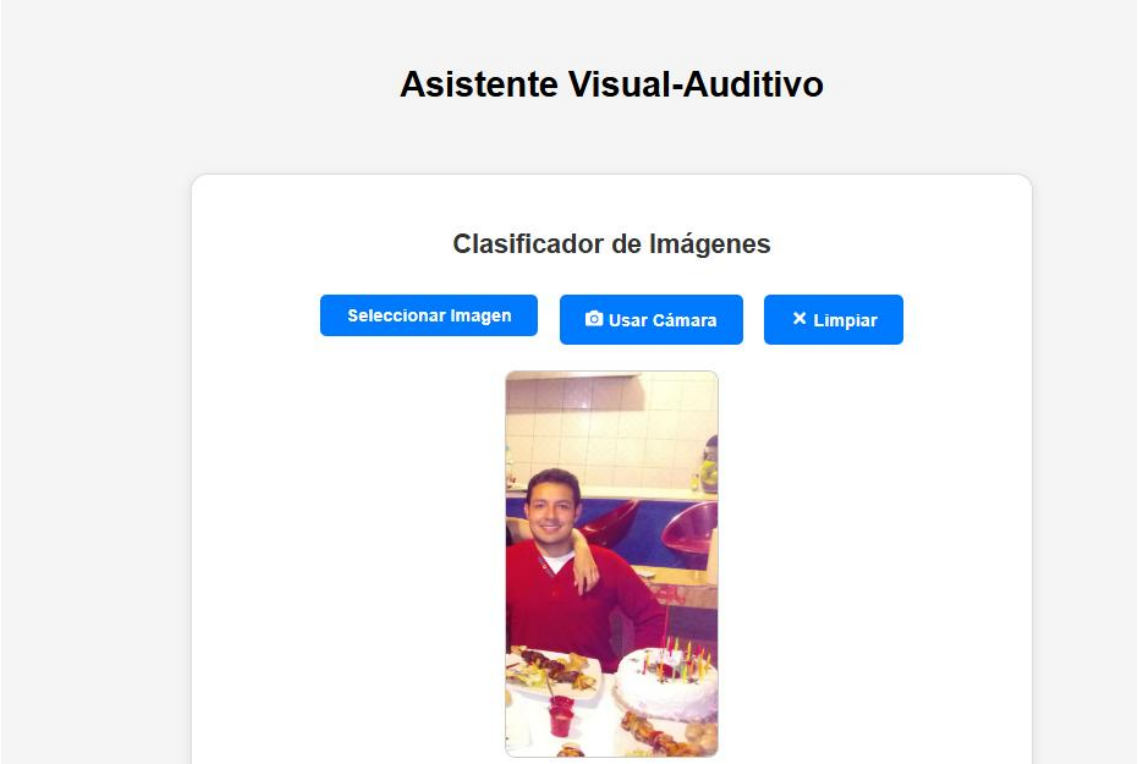
Ejecutamos el código npm run dev:



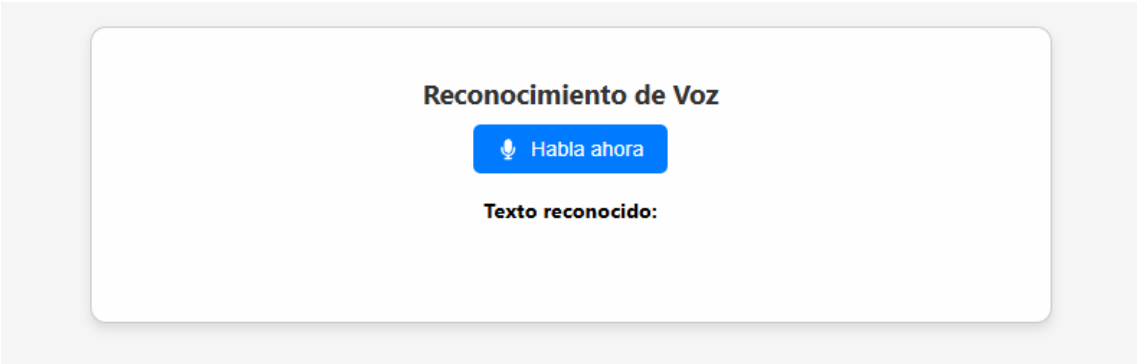
Se abrirá un enlace el cual pondremos en el navegador para visualizar, seleccionamos la imagen que deseamos clasificar.



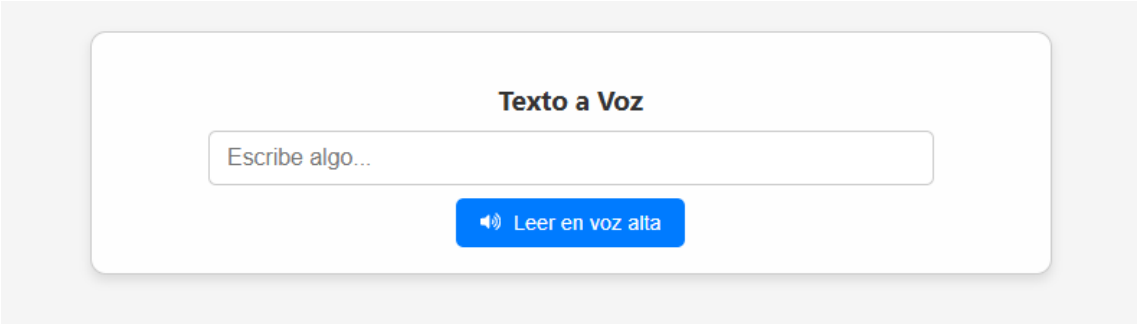
Luego le damos clic en clasificar imagen y seleccionamos la imagen, nos saldrá que objeto es esa imagen:



El siguiente botón es para convertir el audio escuchado a texto:



Y la última funcionalidad es escribir y que se escuche un audio lo que escribiste:


Texto a Voz
Escribe algo...
Leer en voz alta