

# Proyecto Final

Derly Yanneth Rojas Herrera, Juan Felipe Salamanca González

**Abstract**—Este proyecto utiliza Django, un framework web de Python, para desarrollar una aplicación de traducción de objetos a señas. Se implementa un modelo de detección de objetos utilizando la biblioteca ml5.js, que se ejecuta en el navegador del usuario. La detección de objetos se realiza en tiempo real a través de la cámara web del dispositivo del usuario. Los objetos detectados se traducen a señas utilizando una base de datos de imágenes de señas asociadas a cada objeto. La aplicación utiliza tecnologías web como HTML, CSS y JavaScript para crear una interfaz de usuario interactiva y receptiva.

Keywords: Django, ml5.js, detección de objetos, traducción de objetos a señas, tecnologías web.

## I. INTRODUCTION

El proyecto "Traductor de objetos a señas" tiene como objetivo desarrollar una aplicación web que permita a los usuarios traducir objetos del mundo real a señas utilizando la detección de objetos y una base de datos de imágenes de señas asociadas. La aplicación utiliza el framework Django de Python para el desarrollo del backend y la gestión de la base de datos, y ml5.js para la detección de objetos en el frontend.

## II. DESARROLLO

El backend de la aplicación se desarrolla en Django, un framework web de alto nivel basado en Python que facilita la creación rápida y eficiente de aplicaciones web. La estructura del proyecto incluye archivos clave como `asgi.py`, `settings.py`, `urls.py`, y `wsgi.py`, los cuales configuran la aplicación y manejan las solicitudes HTTP entrantes y salientes.

`asgi.py`: Este archivo configura la interfaz de servidor de aplicaciones (ASGI) para la aplicación, exponiendo el callable ASGI como una variable de nivel de módulo llamada `application`. `settings.py`: Aquí se encuentran todas las configuraciones de la aplicación, como la configuración de la base de datos, la configuración de archivos estáticos y multimedia, la configuración de aplicaciones instaladas y otras configuraciones importantes para el funcionamiento del proyecto. `urls.py`: Este archivo define las URL de la aplicación, mapeando las solicitudes entrantes a las vistas correspondientes que manejan esas solicitudes. `wsgi.py`: Configura la interfaz de servidor web de puerta de enlace (WSGI) para la aplicación, exponiendo el callable WSGI como una variable de nivel de módulo llamada `application`. Además, en el archivo `models.py`, se define un modelo de base de datos llamado Señas, el cual representa los objetos del mundo real y las señas asociadas. Este modelo incluye campos como `name` para el nombre del objeto y `image` para la imagen de la señal correspondiente.

En cuanto al frontend de la aplicación, se utiliza una combinación de HTML, CSS y JavaScript para crear una

interfaz de usuario (UI) interactiva y receptiva. La detección de objetos en tiempo real se logra mediante el uso de la biblioteca ml5.js, que permite ejecutar modelos de aprendizaje automático en el navegador del usuario sin necesidad de enviar datos a un servidor externo.

El proceso de detección de objetos comienza capturando el video de la cámara web del dispositivo del usuario, utilizando la API de medios HTML5. Este video se pasa a través del modelo de detección de objetos, el cual identifica y localiza los objetos presentes en la escena en tiempo real. Una vez que se detectan los objetos, se traducen a señas utilizando la base de datos de imágenes de señas asociadas.

## III. RESULTADOS

La aplicación ofrece una experiencia inmersiva y efectiva para los usuarios al permitirles detectar objetos del mundo real a través de su cámara web y traducirlos a señas de manera interactiva. Los resultados de la detección de objetos y las señas asociadas se presentan de forma clara y precisa en la interfaz de usuario en tiempo real, lo que facilita la comprensión y la comunicación para personas con discapacidad auditiva.

La detección de objetos en tiempo real se realiza de manera eficiente gracias a la integración de la biblioteca ml5.js en el frontend de la aplicación. Esta biblioteca ejecuta un modelo de detección de objetos directamente en el navegador del usuario, lo que garantiza una detección rápida y precisa sin la necesidad de enviar datos a un servidor externo. Los objetos detectados se comparan con la base de datos de imágenes de señas asociadas, y las traducciones de señas correspondientes se muestran instantáneamente en la interfaz de usuario.

Los usuarios pueden interactuar con la aplicación de manera intuitiva, simplemente activando la cámara web de su dispositivo y apuntando a los objetos que desean traducir a señas. La interfaz de usuario proporciona retroalimentación visual en tiempo real, lo que permite a los usuarios ver cómo se detectan los objetos y cómo se traducen a señas en el momento.

La precisión y la velocidad de la detección de objetos y la traducción de señas son aspectos clave de los resultados de la aplicación. La aplicación se esfuerza por proporcionar traducciones precisas y oportunas, lo que garantiza una experiencia fluida y satisfactoria para los usuarios. Además, la capacidad de realizar estas operaciones en tiempo real sin retrasos significativos mejora la usabilidad y la accesibilidad de la aplicación.

## IV. CONCLUSIONES

El proyecto demuestra la viabilidad de utilizar tecnologías web como Django y ml5.js para desarrollar aplicaciones

interactivas de traducción de objetos a señas. La detección de objetos en tiempo real y la traducción a señas se logran de manera efectiva, lo que hace que la aplicación sea accesible y útil para personas con discapacidad auditiva. Se pueden explorar futuras mejoras en la precisión de la detección de objetos y la expansión de la base de datos de señas para mejorar la funcionalidad y la usabilidad de la aplicación.

## V. APÉNDICES

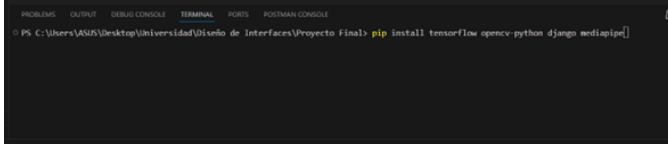


Fig. 1. Instalación de librerías necesarias para el funcionamiento de la aplicación

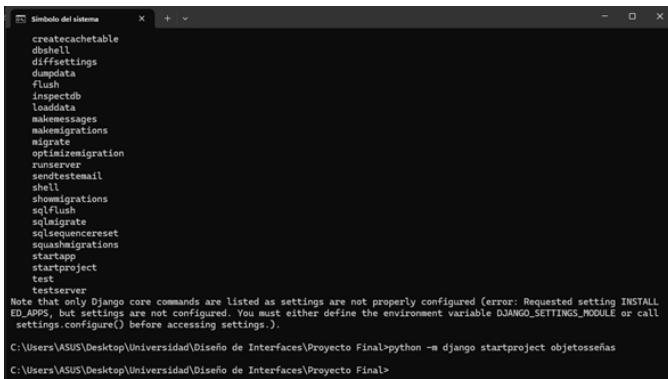


Fig. 2. Creación de la carpeta del proyecto objetoseñas y la inicialización del mismo

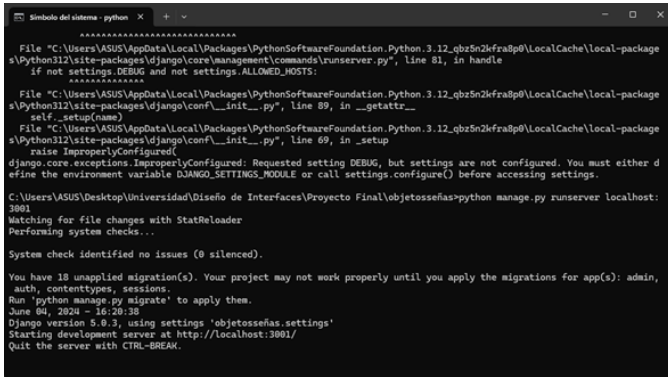


Fig. 3. Detección de la cámara en la aplicación web

## VI. QUERY

( TITLE-ABS-KEY ( object AND detector ) AND TITLE-ABS-KEY ( signals AND language ) ) AND PUBYEAR ; 2018 AND PUBYEAR ; 2024 AND ( LIMIT-TO ( SUB-AREA , “ENGI” ) OR LIMIT-TO ( SUBJAREA , “COMP” ) OR LIMIT-TO ( SUBJAREA , “DECI ‘ ) ) ) AND ( LIMIT-TO ( DOCTYPE , ‘cp’ ) OR LIMIT-TO ( DOCTYPE , “ar” ) ) ) AND ( LIMIT-TO ( LANGUAGE , “English ‘ ) ) ) AND ( LIMIT-TO ( EXACT KEYWORD , ‘Object Detection ‘ ) OR LIMIT-TO ( EXACT KEYWORD , ‘Learning Systems ‘

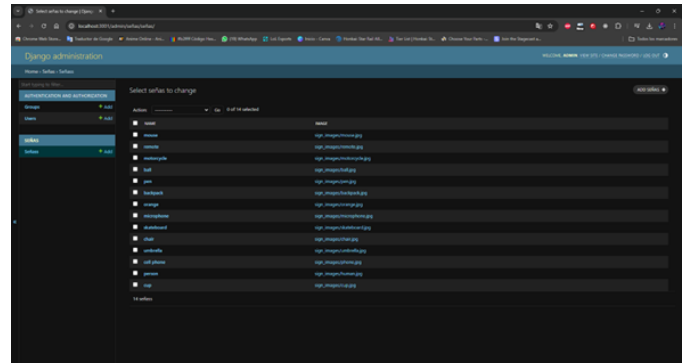


Fig. 4. Creación del super usuario llamado admin y con una contraseña llamada pos123456

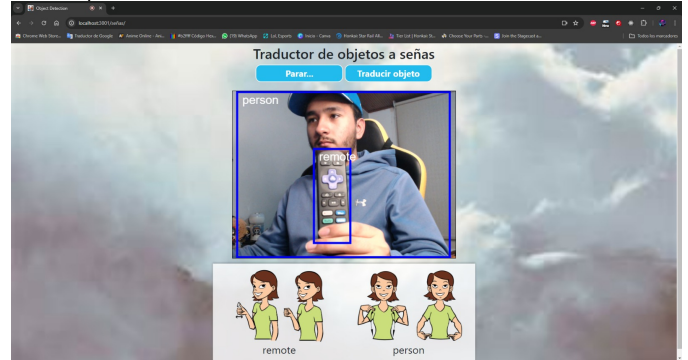


Fig. 5. Visualización de la página

) OR LIMIT-TO ( EXACT KEYWORD , ‘Deep Learning ‘ ) ) AND ( LIMIT-TO ( AFFIL COUNTRY , ‘United States’ ) OR LIMIT-TO ( AFFILCOUNTRY , “India” ) )