

Backend - PDF a Audio

Este backend está construido con Flask y expone un endpoint para convertir archivos PDF a texto y luego a audio usando OCR y edge-tts.

Requisitos

- Python 3.8+
- Tesseract-OCR instalado en el sistema (y accesible desde la variable de entorno `TESSERACT_CMD`)

Instalación y ejecución local

1. Navega a la carpeta `backend`:

```
cd backend
```

2. Crea un entorno virtual:

```
python -m venv env
env\Scripts\activate # En Windows
# o
source env/bin/activate # En Linux/Mac
```

3. Instala las dependencias:

```
pip install -r requirements.txt
```

4. Crea un archivo `.env` con la ruta a tu ejecutable de Tesseract:

```
TESSERACT_CMD="C:\\Ruta\\A\\Tesseract-OCR\\tesseract.exe"
TESSDATA_PREFIX="C:\\Ruta\\A\\Tesseract-OCR\\tessdata"
```

5. Ejecuta el servidor:

```
python app.py
```

El backend estará disponible en <http://localhost:5000>.

Endpoint principal

- **POST** [/procesar](#)
 - **Body:** [form-data](#)
 - [pdf](#) (File, requerido): El archivo PDF a procesar
 - [lang](#) (Text, opcional): Idioma OCR (por defecto: spa)
 - [voice](#) (Text, opcional): Voz para el audio (por defecto: es-ES-ElviraNeural)
 - [out](#) (Text, opcional): Nombre del archivo de texto de salida
 - [audio](#) (Text, opcional): Nombre del archivo de audio de salida
 - **Respuesta:** JSON con los nombres de los archivos generados

Despliegue en Render

- **Servicio:** Web Service
- **Root Directory:** [backend](#)
- **Build Command:** [bash build.sh](#)
- **Start Command:** [gunicorn app:app](#)

Notas

- Los archivos subidos se guardan en la carpeta [input/](#).
- Los resultados se guardan en la carpeta [output/](#).
- El backend acepta conexiones desde cualquier IP ([host=0.0.0.0](#)).