

Manual de Usuario Flask Server

Índice:

[Índice:](#)

[1. Introducción](#)

[2. Requisitos del sistema](#)

[3. Instalación](#)

[3.1 Descargar el programa](#)

[3.2 Requisitos previos](#)

[3.3 Pasos de instalación](#)

[3.4 Desinstalación](#)

[4. Endpoints principales](#)

[POST /ocr](#)

[POST /ocr-file](#)

[POST /aichat](#)

[GET /status](#)

[5. Notas](#)

[6. Solución de problemas comunes](#)

1. Introducción

Nombre del programa: SOCRAI

Versión: 0.0.2 (Beta)

Fecha de lanzamiento: 30/05/25

Autor: Laureano De Sousa Dias

Descripción:

SOCRAI es un servicio REST que combina OCR (Reconocimiento Óptico de Caracteres) con inteligencia artificial para procesar tickets digitales. Permite cargar tickets en formato de imagen o documento y extraer información relevante mediante técnicas avanzadas de OCR y modelos de lenguaje de OpenAI.

2. Requisitos del sistema

Sistema operativo: Windows

Python 3.8 o superior

Tesseract OCR instalado en: C:\Archivos de Programa ([Link](#))

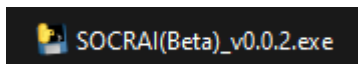
Dependencias Python listadas en requirements.txt (Solo proyecto)

3. Instalación

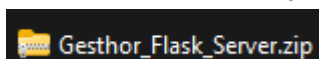
3.1 Descargar el programa

Desde la página de github, en el README del servidor encontrarás Descargas, descarga el archivo desde google Drive.

SOCRAI v0.0.2 .exe (ejecutable listo para usar)



SOCRAI v0.0.2 .zip (ejecutable con manual)



Descarga el repositorio o el directorio del servidor Flask
([Descargar](#))

3.2 Requisitos previos

- Tesseract Instalado con valores default ([Link](#))
- Python 3.8+
- Sí descargas repositorio: `pip install --no-cache-dir -r requirements.txt`

3.3 Pasos de instalación

1. Descargar el ejecutable en el formato deseado.
 - a. .exe muévelo a una carpeta accesible es en donde se ejecutará.
 - b. .zip descomprime , contiene ejecutable e instrucciones
 - c. Código del proyecto ([Github](#))
2. Instalar dependencias (Solo proyecto):
 - a. En la carpeta del proyecto,ejecuta:
`pip install --no-cache-dir -r requirements.txt`
3. Carga las siguientes variable de sistema como mínimo:
 - a. Cargar variables de sistema desde cmd:

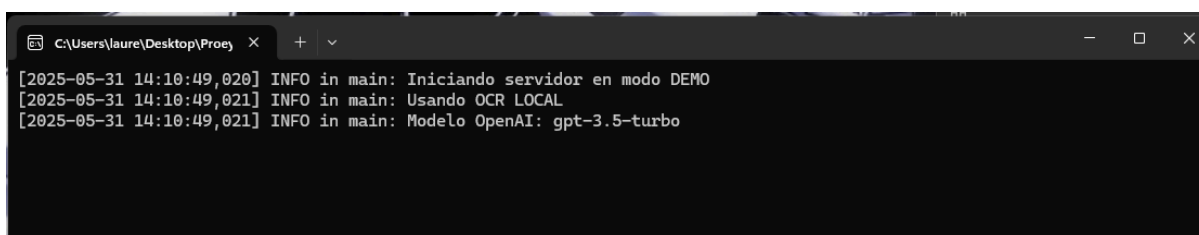
```
setx FLASK_DEMO_MODE false
setx FLASK_OCR_LOCAL true
setx FLASK_MODEL "gpt-3.5-turbo"
setx FLASK_EXTENSIONS "jpg,jpeg,png,pdf,txt,webp"
setx FLASK_MAX_CONTENT_LENGTH_MB 10
setx FLASK_HOST 0.0.0.0
setx FLASK_PORT 5000
setx FLASK_DEBUG_MODE true
setx FLASK_LOCAL_API_KEY "tu_api_key_aqui"
```

```
setx FLASK_OPENAI_API_KEY "tu_openai_api_key_aqui"
```

- b. Cargar variables de sistema desde .env:
 - i. Crea el archivo .env y ponlo al nivel del ejecutable.
 - ii. Crea el archivo .env o editalo en el proyecto.

```
FLASK_DEMO_MODE=false
FLASK_DEBUG_MODE=true
FLASK_OCR_LOCAL=true
FLASK_EXTENSIONS=jpg,jpeg,png,pdf,txt,webp
FLASK_MAX_CONTENT_LENGTH_MB=10
FLASK_MODEL=gpt-3.5-turbo
FLASK_HOST=0.0.0.0
FLASK_PORT=5000
FLASK_LOCAL_API_KEY=tu_api_key_aqui
FLASK_OPENAI_API_KEY=tu_openai_api_key_aqui
```

- 4. Una vez en el directorio deseado, ejecute el servidor, esté instalará las dependencias necesarias y se ejecutará.
 - a. Para .exe bubble click
 - b. Para código, desde cmd:
 - i. Activar entorno virtual (Opcional):
`.\venv\Scripts\activate`
 - ii. Ejecutar el servidor:
`python main.py`



- 5. Modificar variables de entorno:
 - a. Sobrescribe Las mediante cmd paso 3.
 - b. Modifica .env
 - c. Sistema -> Config Avanzada -> Variables de Entorno

3.4 Desinstalación

Las librerías instaladas o usadas son de python, para eliminar el programa borre el directorio en el que se encuentra este, o borre la carpeta logs y el ejecutable.

4. Endpoints principales

POST /ocr

- Procesa una imagen (JPG, PNG, etc.) con OCR local o remoto (OpenAI).
- Parámetros: archivo de imagen (file) en multipart/form-data.
- Cabecera requerida: x-api-key (API key configurada).
- Respuestas:
 - 200 OK: JSON con resultado OCR.
 - 400 Bad Request: archivo faltante o formato no permitido.
 - 503 Service Unavailable: falla conexión.
 - 500 Internal Server Error: error inesperado.
 - En modo demo, devuelve respuesta simulada.

POST /ocr-file

- Procesa archivo PDF o TXT con OCR.
- Parámetros: archivo (file) en multipart/form-data.
- Cabecera requerida: x-api-key.
- Respuestas:
 - 200 OK: JSON con resultado OCR.
 - 400 Bad Request: archivo faltante o formato no permitido.
 - 503 Service Unavailable: falla conexión.
 - 500 Internal Server Error: error inesperado.
 - En modo demo, devuelve respuesta simulada.

POST /aichat

- Envía un mensaje de texto para obtener respuesta generada por IA.
- Parámetros JSON:

```
{"mensaje": "Texto que deseas enviar a la IA"}
```
- Cabecera requerida: x-api-key.
- Respuestas:
 - 200 OK: JSON con texto generado.
 - 400 Bad Request: JSON inválido o falta campo.
 - 503 Service Unavailable: falla conexión IA.
 - 500 Internal Server Error: error interno.
- En modo demo, responde con texto fijo.

GET /status

- Verifica el estado del servidor.
- Respuesta JSON ejemplo:

```
{"statusServer": true, "demo": false, "ocrLocal": true}
```

5. Notas

La carga de tickets puede fallar; intente varias veces.

Tesseract debe estar instalado en C:\Archivos de Programa.

Servidor por defecto: <http://0.0.0.0:5000>

GPT-3.5 no procesa archivos; requiere OCR local.

6. Solución de problemas comunes

- El servidor no arranca:
 - Explicación: Esto suele suceder cuando no encuentra variables del entorno o hay un fallo de configuración.
 - Solución: Activa el modo debug del servidor y revisa los logs en la carpeta logs que se generará.

- El servidor arranca pero no tengo clave:
 - Explicación: Sin clave el servidor solo funciona en modo demo.
 - Solución: Accede a la página de desarrollo de openAI y obtén una clave, si solo quieres probar que funciona, activa el modo demo.