

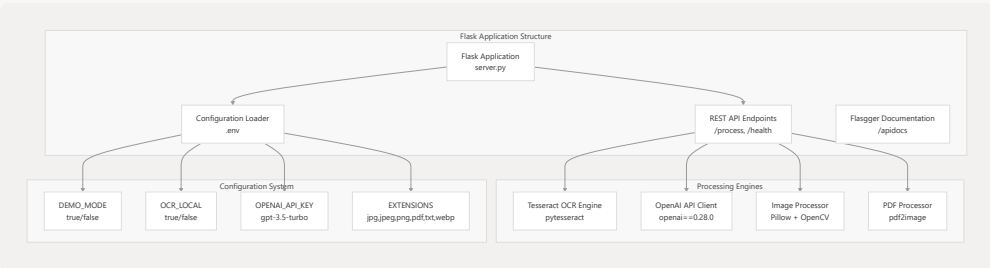
Servidor OCR de Python

Este documento describe el servidor de OCR basado en Flask que procesa imágenes de recibos y tickets para extraer datos estructurados. El servidor proporciona puntos finales de API REST para el procesamiento de imágenes mediante OCR local de Tesseract e integración con OpenAI en la nube.

Para obtener información sobre cómo se integra el backend de Spring Boot con este servicio de OCR, consulte [Integración de OCR con el backend](#) . Para obtener más información sobre la arquitectura general del sistema de procesamiento de OCR, consulte [Sistema de procesamiento de OCR](#) .

Arquitectura y configuración del servidor

El servidor OCR de Python es una aplicación Flask independiente `ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/` que proporciona puntos finales RESTful para el procesamiento de documentos. El servidor admite múltiples modos de procesamiento y formatos de archivo mediante un completo sistema de configuración.



Fuentes: `ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/.env` 1-11

`ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/requirements.txt` 1-21

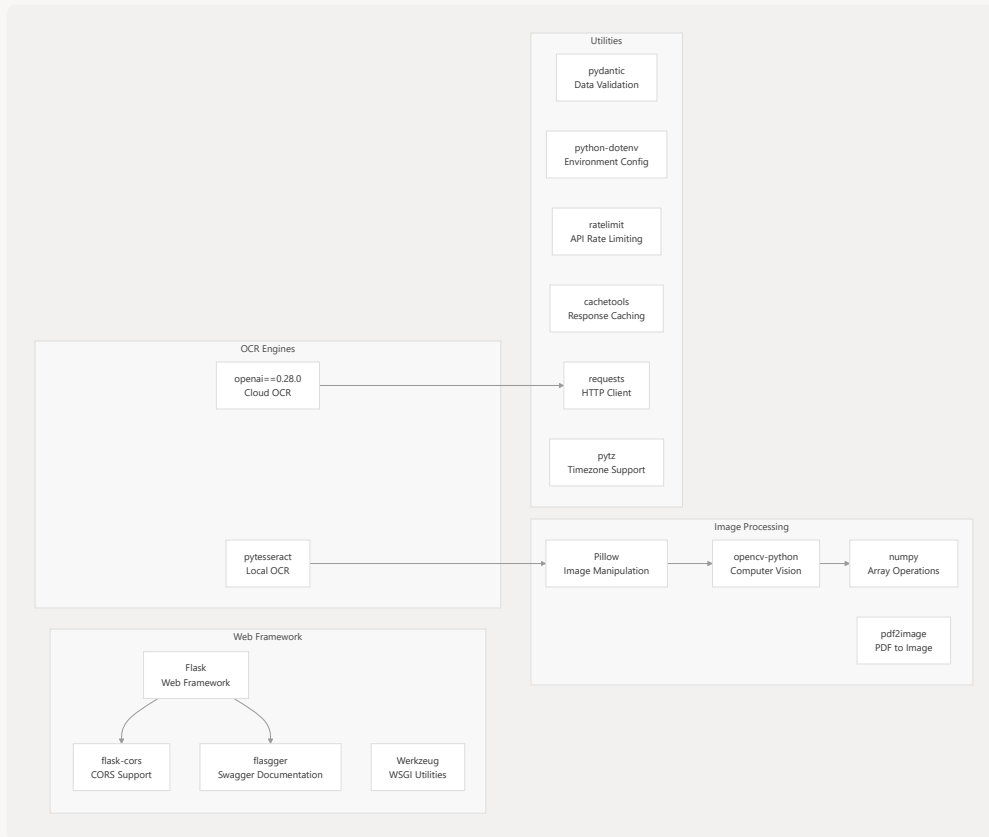
Configuración del entorno

El servidor utiliza variables de entorno definidas en `.env` la configuración de ejecución. Esta configuración admite los modos de desarrollo y producción, con una selección flexible del motor de OCR.

Clave de configuración	Valor predeterminado	Descripción
DEMO_MODE	true	Habilita la funcionalidad de demostración con respuestas simuladas
OCR_LOCAL	true	Utiliza Tesseract local cuando es verdadero, OpenAI cuando es falso
LOCAL_API_KEY	CREAR_API_KEY	Clave de autenticación para el acceso a la API local
MODEL	gpt-3.5-turbo	Modelo OpenAI para extracción de texto
EXTENSIONS	jpg, jpeg, png, pdf, txt, webp	Formatos de archivo compatibles
MAX_CONTENT_LENGTH_MB	10	Tamaño máximo de archivo de carga
HOST	0.0.0.0	Dirección de enlace del servidor
PORT	5000	Puerto del servidor
DEBUG_MODE	true	Modo de depuración de Flask
OPENAI_API_KEY	API-KEY-GPT	Autenticación de API de OpenAI

Dependencias y bibliotecas de Python

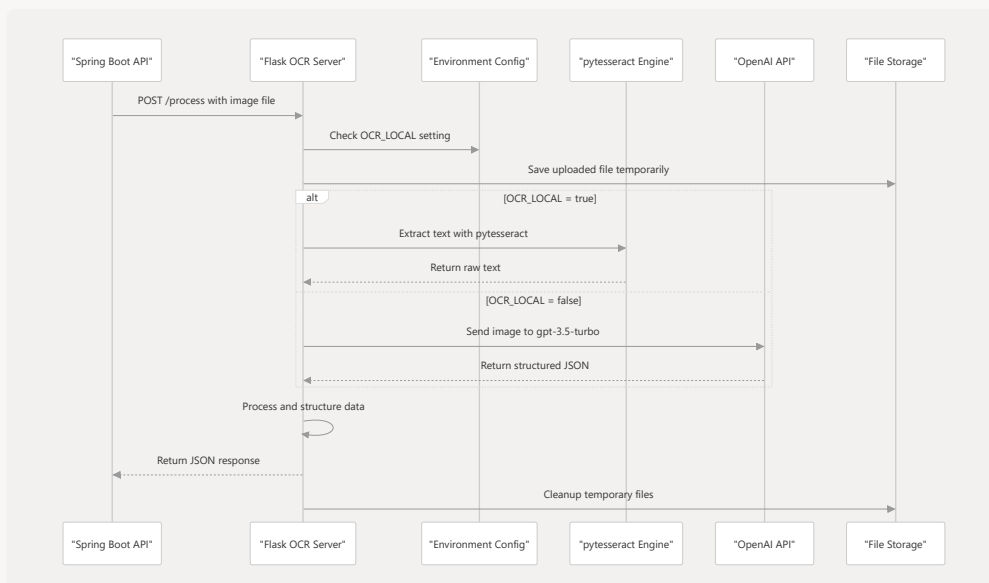
El servidor utiliza un conjunto completo de bibliotecas de Python para la funcionalidad de API REST, procesamiento de imágenes y capacidades de OCR:



Fuentes: [ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/requirements.txt](#) 5-20

Flujo de trabajo de procesamiento y selección de motor

El servidor OCR implementa un flujo de trabajo de procesamiento de doble motor que puede utilizar OCR Tesseract local o procesamiento OpenAI basado en la nube según el `OCR_LOCAL` indicador de configuración.



Fuentes: [ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/.env](#) 3-5

[ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/requirements.txt](#) 9-16

Compatibilidad y validación de formatos de archivo

El servidor acepta múltiples formatos de archivo mediante la `EXTENSIONS` configuración e implementa la validación del tamaño de archivo mediante `MAX_CONTENT_LENGTH_MB`. Los formatos compatibles incluyen tipos de imágenes comunes y documentos PDF.

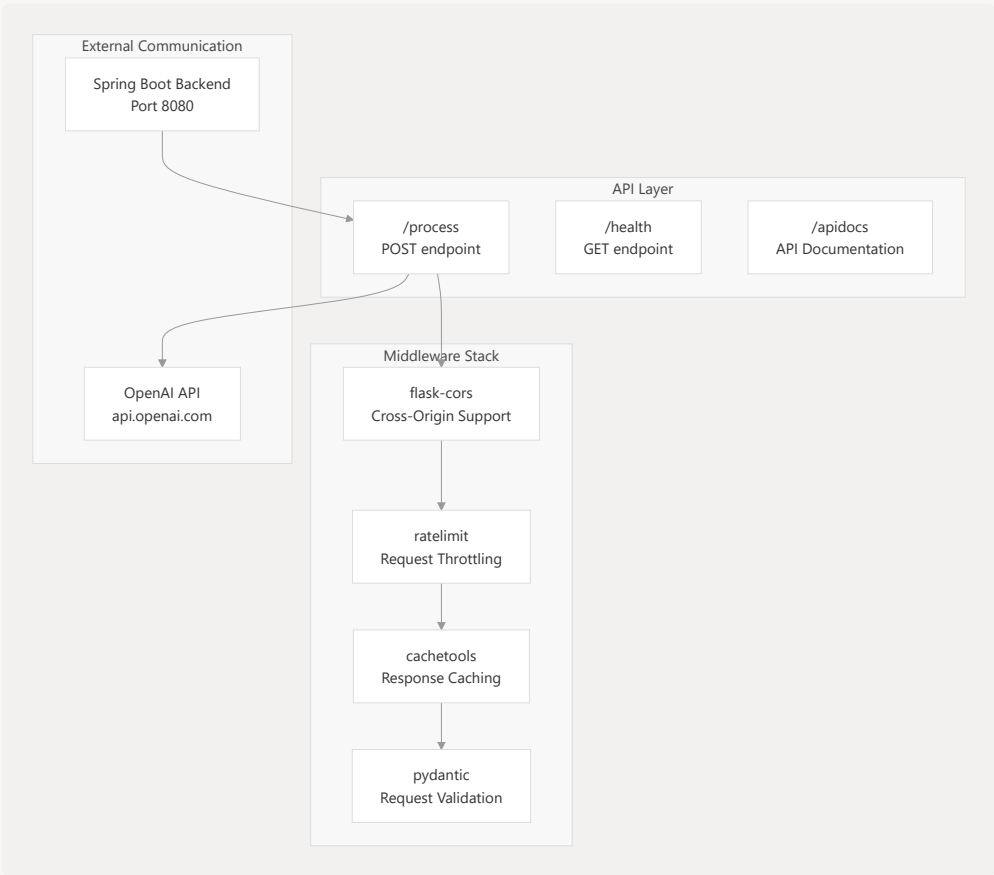
Extensión de archivo	Método de procesamiento	Biblioteca usada
jpg , jpeg	Procesamiento directo de imágenes	Almohada + OpenCV
png	Procesamiento directo de imágenes	Almohada + OpenCV
webp	Procesamiento directo de imágenes	Almohada + OpenCV
pdf	Convertir a imágenes primero	pdf2image → Almohada
txt	Procesamiento directo de texto	Operaciones de archivos integradas

Fuentes: [ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/.env](#) 6-7

[ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/requirements.txt](#) 8-19

Puntos de integración y diseño de API

El servidor Flask proporciona puntos finales REST que se integran con el backend de Spring Boot mediante comunicación HTTP. El servidor incluye limitación de velocidad, almacenamiento en caché y gestión integral de errores para su uso en producción.



Fuentes: [ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/.env](#) 8-9

[ServidorServicioOCR/ServidorPython/server/ApiRestOCR+AI/requirements.txt](#) 12-15