# 3.4 Desarrollo del Backend

El backend del sistema InvoiceMind fue construido utilizando el framework FastAPI, una tecnología moderna y eficiente basada en Python que permite la creación de APIs RESTful con alto rendimiento y soporte nativo para programación asincrónica. El backend cumple el rol central dentro del sistema, coordinando el flujo completo desde la recepción de archivos PDF, el procesamiento y extracción de datos fiscales, hasta la generación del archivo Excel y el almacenamiento del historial de los documentos procesados.

A continuación, se describe en profundidad la arquitectura, estructura modular y lógica funcional del backend, conforme al diseño técnico implementado.

3.4.1 Estructura General del Backend

La estructura del backend fue organizada de forma modular, con separación clara de responsabilidades entre los diferentes componentes. A nivel general, se compone de los siguientes archivos y carpetas:  
  
app/  
│  
├── main.py # Endpoints principales de FastAPI  
├── crud/ # Módulos de persistencia en base de datos  
│ ├── historial.py # Guardado de PDFs procesados  
│ └── historial\_excel.py # Guardado de Excels generados  
├── utils/  
│ ├── procesar\_facturas.py # Procesamiento asincrónico en lotes  
│ ├── pdf\_utils.py # Extracción de texto y tipo de documento  
│ ├── leer\_qr.py # Detección y lectura de códigos QR  
│ ├── parsear\_qr.py # Decodificación de parámetros del QR  
│ ├── extraer\_bs.py # Extracción adicional con Regex + BeautifulSoup  
│ ├── completar\_llm.py # Completado de datos con IA (Gemini)  
│ └── plantilla.py # Plantilla base para estructuración JSON

3.4.2 Punto de Entrada: /procesar-excel

Ubicado en main.py, el endpoint /procesar-excel es el punto de entrada principal para el usuario. Permite subir uno o varios archivos PDF y desencadena el flujo de procesamiento.  
  
Pasos:  
- Se valida el token del usuario y se comprueba si posee procesamientos disponibles.  
- Se verifica que la cantidad de archivos subidos no exceda el límite según su plan.  
- Se llama a la función procesar\_lote\_facturas() pasando los archivos en binario y el ID del usuario.  
- Una vez finalizado el procesamiento, se genera el archivo Excel y se devuelve como respuesta en StreamingResponse.

3.4.3 Procesamiento Asíncrono en Lotes

Función: procesar\_lote\_facturas()  
Archivo: procesar\_facturas.py  
  
Esta función permite procesar múltiples facturas en paralelo mediante la creación de tareas asincrónicas usando asyncio.gather().  
  
Pasos:  
- Por cada archivo PDF se lanza una tarea individual llamando a procesar\_factura\_pdf().  
- Se recolectan los resultados de cada tarea en un listado de JSONs.  
- Se mapean los datos a un formato plano para Excel.  
- Se construye un DataFrame de pandas y se genera el archivo Excel.  
  
Esto permite escalar horizontalmente el sistema y reducir el tiempo de espera del usuario.

3.4.4 Procesamiento Individual de Cada Factura

Función: procesar\_factura\_pdf(pdf\_bytes)  
Ubicación: procesar\_facturas.py  
  
Esta función orquesta todas las etapas de procesamiento para una única factura. El primer paso es determinar si la factura es digital o escaneada.

... (contenido continúa en el documento completo)