**AODA**

**(SAD) Software Architecture Document**

**Versión 1.0**

* **Tabla de Contenidos**

Contenido

[**● Tabla de Contenidos 0**](#_heading=h.gkp737h9219k)

[**● Identificación de Documento 2**](#_heading=h.kvyhq8ffddmh)

[**● Historia de cambios 2**](#_heading=h.42sj4ucaubjt)

[**● Índice de figuras 3**](#_heading=h.3znysh7)

[**● Introducción 4**](#_heading=h.2s8eyo1)

[**● Alcance 4**](#_heading=h.zet794er9cuo)

[**● Referencias 4**](#_heading=h.3rdcrjn)

[**● Arquitectura de Software 4**](#_heading=h.26in1rg)

[**● Objetivos y Restricciones de la Arquitectura 4**](#_heading=h.lnxbz9)

[**● Diagramas 4**](#_heading=h.35nkun2)

[**● Tamaño y desempeño 4**](#_heading=h.xh4j8fds8gvf)

* Identificación de Documento

| **Identificación** | Documento de Arquitectura de Software - Proyecto AODA |
| --- | --- |
| **Proyecto** | AODA (Agente de Orientación y Derivación Asistida) |
| **Versión** | 1.0 |

| **Documento mantenido por** | Ignacio Martinez / Gabriel Fernandez / Benjamin Cortez |
| --- | --- |
| **Fecha de última revisión** | 09 - 10 - 2025 |
| **Fecha de próxima revisión** | 1 - 12 - 2025 |

| **Documento aprobado por** | Jorge Castro (Profesor) |
| --- | --- |
| **Fecha de última aprobación** |  |

* **Historia de cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 09 - 10 - 2025 | 1.0 | Creación documento inicial con la descripción general del sistema y su arquitectura | Ignacio Martinez |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* **Índice de figuras**
* **Introducción**
* **Alcance**
* **Referencias**
* **Arquitectura de Software**
* **Objetivos y Restricciones de la Arquitectura**
* **Diagramas**
* **Tamaño y desempeño**

### **Introducción**

### El presente documento describe la arquitectura de software del sistema **AODA (Agente de Orientación y Derivación Asistida)**, desarrollado para el **Servicio de Salud Metropolitano**. Su propósito es apoyar la gestión de **derivación de víctimas de delitos**, automatizando la identificación del centro de atención más adecuado según variables como comuna, edad, sexo, situación migratoria y tipo de delito.

### AODA se construye sobre una arquitectura **cliente-servidor**, compuesta por un **frontend en React**, un **backend en FastAPI**, y una **base de datos PostgreSQL**, integrando además un **agente de inteligencia artificial (OpenAI)** encargado de procesar la información de los casos y generar recomendaciones.

### El documento detalla los principales componentes del sistema, su estructura lógica, las interacciones entre módulos y las restricciones técnicas que guían su desarrollo, asegurando una solución escalable, segura y alineada con los objetivos institucionales del Servicio de Salud.

### 

### **Alcance**

El sistema **AODA** abarca el diseño, desarrollo e implementación de una plataforma web que permita a los funcionarios del **Servicio de Salud Metropolitano** registrar casos de víctimas de delitos y obtener, mediante un agente de inteligencia artificial, la recomendación del centro de apoyo más adecuado.

El alcance incluye los módulos de **ingreso de información**, **procesamiento y análisis de casos**, **recomendación automática de derivaciones**, **registro histórico de derivaciones** y un **dashboard de indicadores** para el seguimiento institucional.

El proyecto contempla la integración completa entre los componentes **frontend (React)**, **backend (FastAPI)**, **base de datos (PostgreSQL)** y el **agente IA (OpenAI)**, garantizando un flujo de información continuo y trazable.

No se considera dentro del alcance la conexión directa con sistemas externos del Estado ni la gestión de datos personales sensibles, respetando las políticas de privacidad y las limitaciones del entorno institucional.

### **Referencias**

* Acta de Constitución del Proyecto AODA
* Minuta Kick Off del Proyecto
* Especificación de Requerimientos de Software (ERS)
* Documentación de FastAPI, React y PostgreSQL
* Guías de arquitectura REST y diseño de API

### **Arquitectura de Software**

La arquitectura de AODA se basa en una estructura **cliente-servidor** con integración de un **motor de IA**.

**Componentes principales:**

* **Frontend (React):** Interfaz para ingreso de casos, visualización de sugerencias y confirmación de derivaciones.
* **Backend (FastAPI):** Provee servicios REST para procesar solicitudes, comunicar la IA con la base de datos y gestionar usuarios.
* **IA (OpenAI API):** Analiza la información del caso y sugiere el centro de apoyo más pertinente según criterios definidos.
* **Base de Datos (PostgreSQL):** Almacena comunas, delitos, centros, derivaciones e información institucional.
* **Dashboard de Indicadores:** Módulo analítico para visualizar métricas (tiempos, volumen de casos, distribución geográfica).

### **Objetivos y Restricciones de la Arquitectura**

**Objetivos:**

* Diseñar un sistema modular, escalable y seguro.
* Asegurar la interoperabilidad entre los módulos web, IA y base de datos.
* Permitir la trazabilidad total de las derivaciones realizadas.

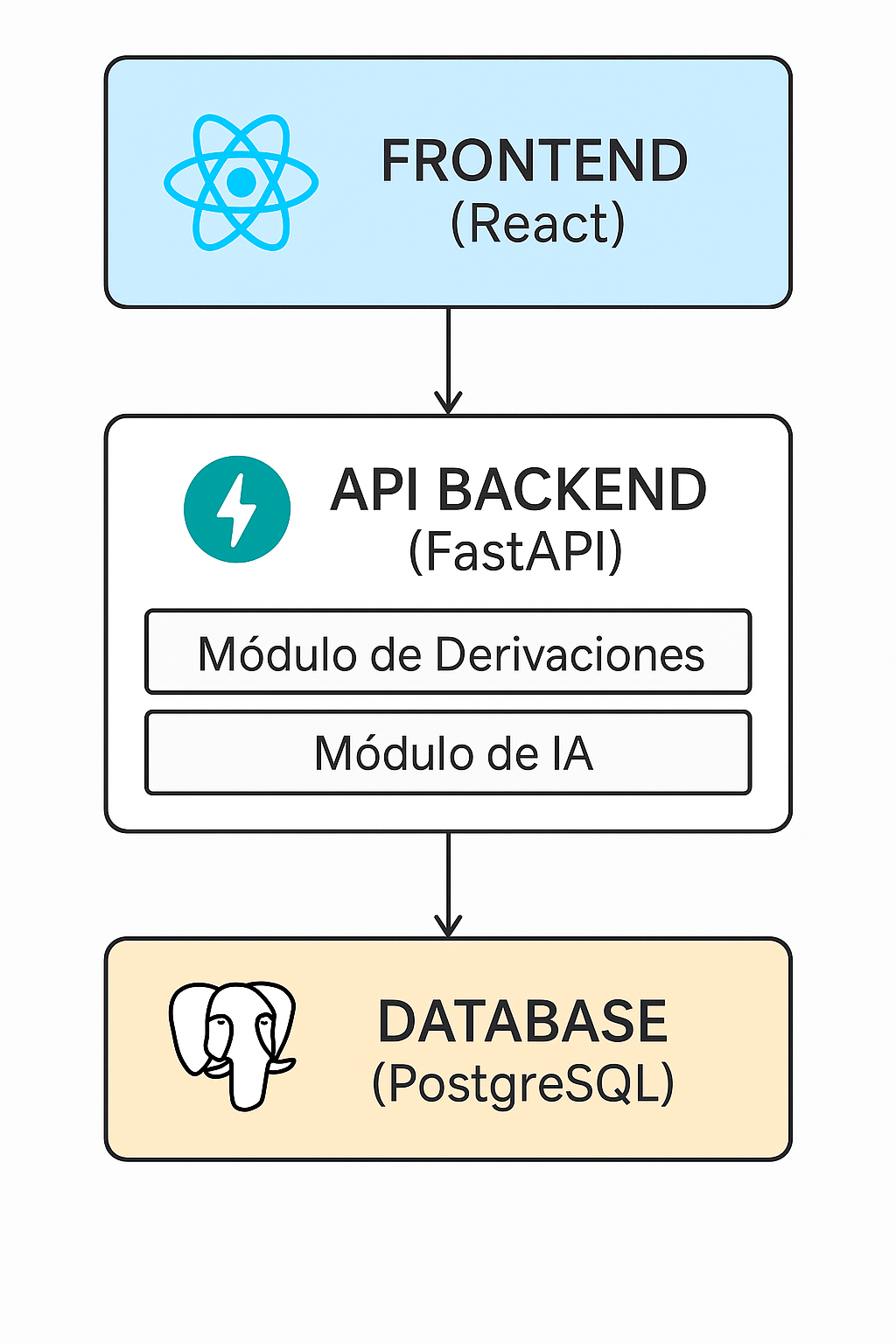
**Restricciones:**

* Limitación de tiempo (10 semanas de desarrollo).
* Presupuesto acotado para infraestructura y licencias.
* Cumplimiento de políticas de privacidad (no almacenar datos personales sensibles).

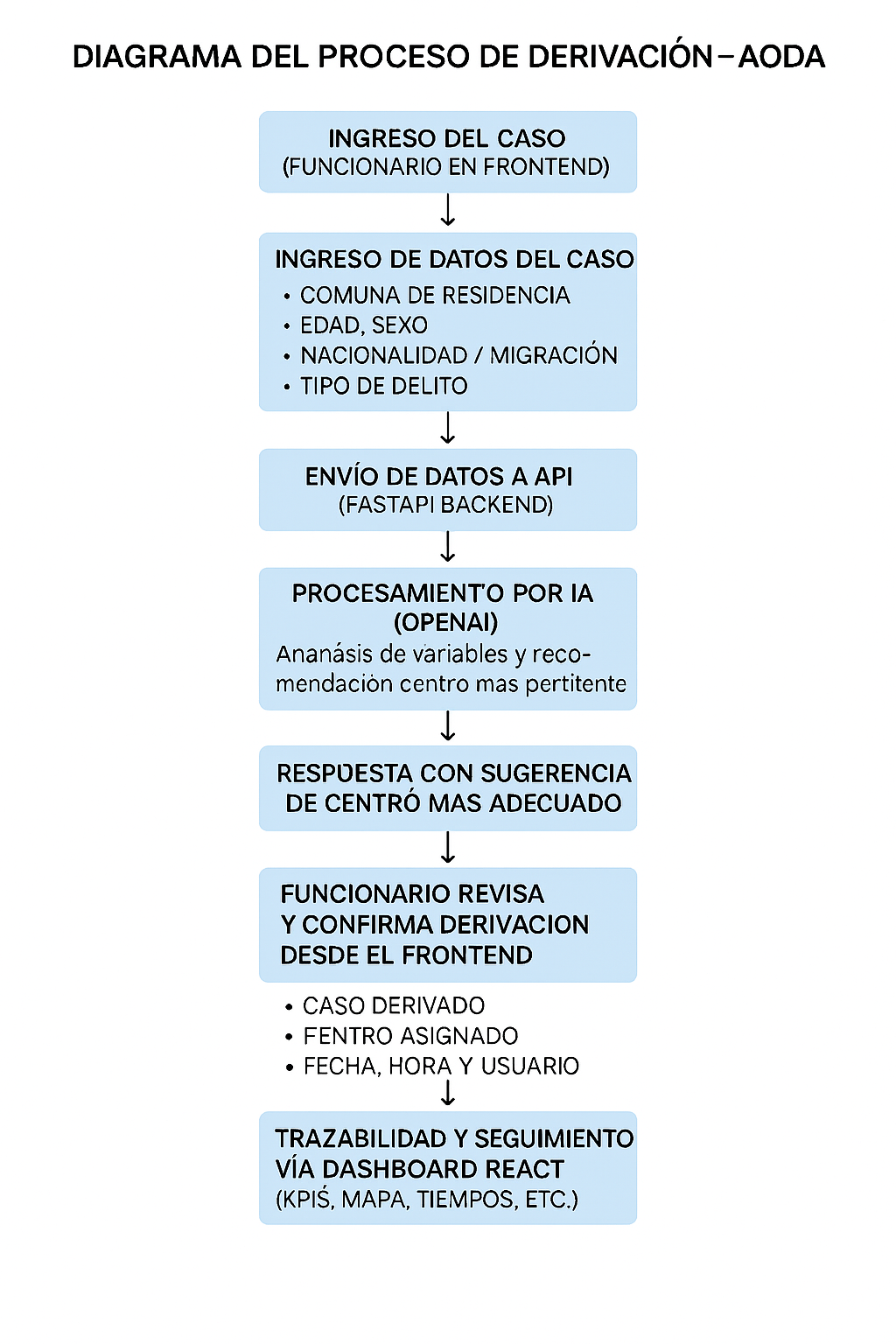
**Diagramas**

Se consideran los siguientes diagramas:

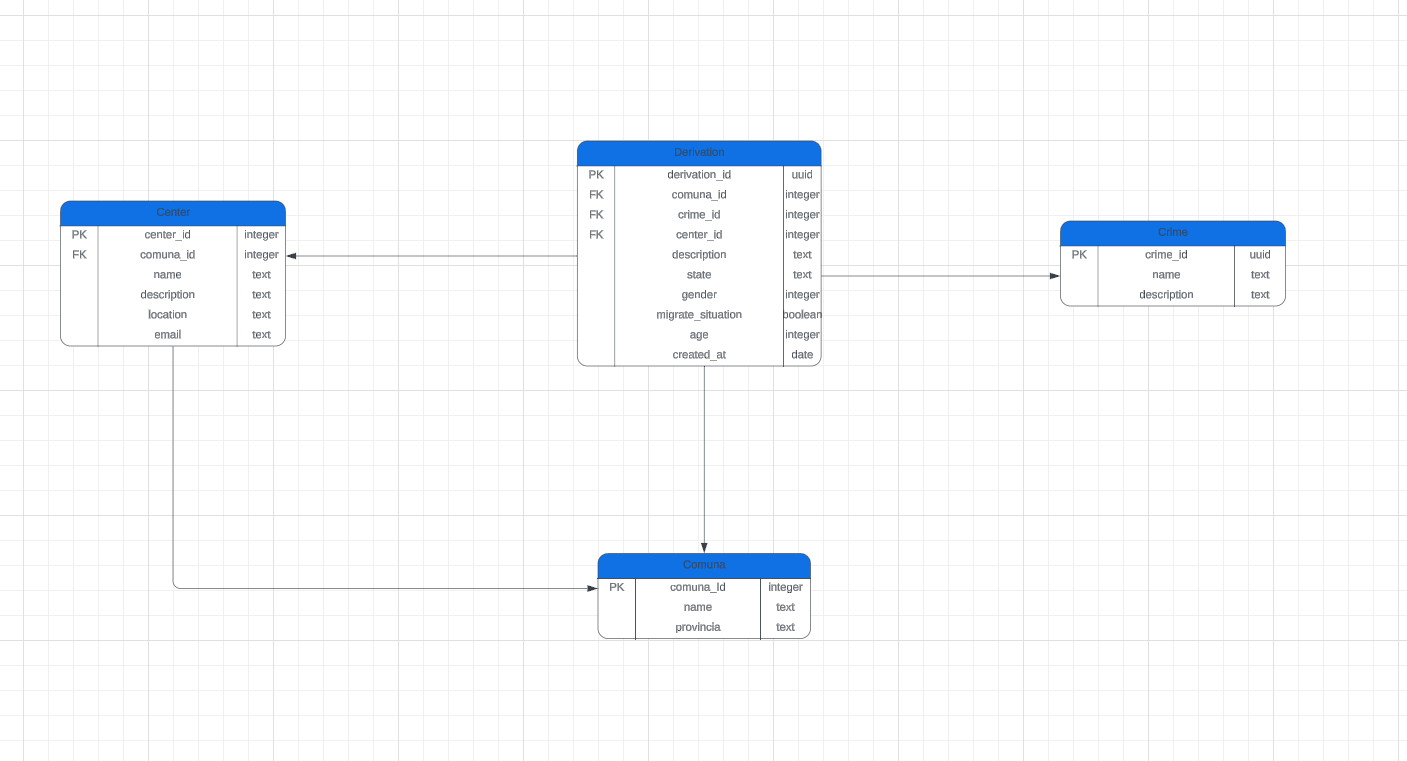
* **Diagrama de Arquitectura General del Sistema:** muestra la comunicación entre frontend, backend, IA y base de datos.

****

* **Diagrama de Flujo de Derivación:** ilustra cómo se procesan los datos de una víctima hasta la recomendación final.



* **Diagrama de Entidades y Relaciones (ER):** representa la estructura de la base de datos PostgreSQL (comunas, delitos, centros, derivaciones).



* **Diagrama de Componentes:** detalla la conexión entre los módulos y servicios del sistema.

### **Tamaño y Desempeño**

El sistema **AODA** está diseñado para operar con un volumen estimado de **100 a 200 derivaciones diarias** durante su fase inicial, con posibilidad de escalar según la demanda institucional. La base de datos PostgreSQL podrá almacenar registros históricos de **hasta 50.000 derivaciones** sin comprometer el rendimiento del sistema.

En condiciones normales, el tiempo promedio de respuesta del agente de inteligencia artificial no deberá superar los **5 segundos** desde el ingreso del caso hasta la entrega de la recomendación. El backend en **FastAPI** y la arquitectura basada en **servicios REST** permiten mantener una comunicación ágil y eficiente entre los módulos.

El sistema está preparado para soportar entre **20 y 30 usuarios concurrentes** en su versión inicial, con opción de aumentar su capacidad mediante escalado horizontal. Asimismo, se implementarán mecanismos de **caché, optimización de consultas SQL y manejo de excepciones**, garantizando la estabilidad y la continuidad operativa del servicio.