| http://www.duoc.cl/sites/default/files/logo_summit_0.png |
| --- |
| Especificación de Requerimientos del Software |
| *Proyecto: AODA* |
|  |
| **Revisión*: 1.0*** |
| **9-10-2025** |

| **ISO/EIC/IEEE 29148 de Ingeniería de Requisitos** |
| --- |

**Tabla de Contenidos**

**Contenido**

[**Ficha del documento 4**](#_heading=h.gjdgxs)

[**1. Introducción 5**](#_heading=h.1ksv4uv)

[1.1. Propósito del Documento 5](#_heading=h.2jxsxqh)

[1.2. Ámbito del Sistema 5](#_heading=h.1y810tw)

[1.3. Definiciones, Acronimos y Abreviaturas 5](#_heading=h.4i7ojhp)

[*1.3.1. Perspectiva del producto 5*](#_heading=h.2xcytpi)

[*1.3.2. Funciones del producto 5*](#_heading=h.1ci93xb)

[*1.3.3. Características del usuario 5*](#_heading=h.3whwml4)

[*1.3.4. Limitaciones 5*](#_heading=h.2bn6wsx)

[**1.4. Referencias 5**](#_heading=h.qsh70q)

[**1.5. Visión General del Documento 5**](#_heading=h.3as4poj)

[**2. Descripción General 6**](#_heading=h.1pxezwc)

[2.1. Perspectiva del Producto 6](#_heading=h.49x2ik5)

[2.2. Funciones del Producto 6](#_heading=h.2p2csry)

[2.3. Características de los usuarios 6](#_heading=h.147n2zr)

[2.4. Restricciones 6](#_heading=h.3o7alnk)

[2.5. Suposiciones y Dependencias 6](#_heading=h.23ckvvd)

[2.6. Requisitos Futuros 7](#_heading=h.ihv636)

[**3. Requisitos Específicos 7**](#_heading=h.32hioqz)

[3.1. Requisitos de las Interfaces 8](#_heading=h.1hmsyys)

[3.1.1. Interfaces de Usuario 8](#_heading=h.41mghml)

[3.1.2. Interfaces de Hardware 8](#_heading=h.2grqrue)

[3.1.3. Interfaces de Software 8](#_heading=h.vx1227)

[*3.1.4. Interfaces de Comunicación 8*](#_heading=h.3fwokq0)

[3.2. Requisitos Funcionales 9](#_heading=h.1v1yuxt)

[3.3. Requisitos no Funcionales 9](#_heading=h.4f1mdlm)

[*3.3.1. Requisitos de Rendimiento 9*](#_heading=h.2u6wntf)

[*3.3.2. Seguridad 9*](#_heading=h.19c6y18)

[*3.3.3. Fiabilidad 9*](#_heading=h.3tbugp1)

[*3.3.4. Disponibilidad 10*](#_heading=h.igv2xye9695p)

[*3.3.5. Mantenibilidad 10*](#_heading=h.jqofmhjxtnqt)

[*3.3.6. Portabilidad 10*](#_heading=h.28h4qwu)

[3.4. Otros requisitos 10](#_heading=h.nmf14n)

[3.5. Requisitos de Base de Datos Lógica. 10](#_heading=h.37m2jsg)

[**4. Propuesta de Planificación 11**](#_heading=h.1mrcu09)

[4.1. Descripción general acerca de la planificación 11](#_heading=h.46r0co2)

[*4.1.1. Definición del Equipo de Trabajo 11*](#_heading=h.2lwamvv)

[*4.1.2. Definición de Actividades principales del Proyecto 11*](#_heading=h.111kx3o)

[*4.1.3. Diagrama EDT 12*](#_heading=h.3l18frh)

[*4.1.4. Carta Gantt 13*](#_heading=h.206ipza)

[*4.1.5. Resumen de Costos del Desarrollo del Proyecto 13*](#_heading=h.4k668n3)

[4.2. Plan de control de Cambio 13](#_heading=h.2zbgiuw)

[**5. Anexos 14**](#_heading=h.1egqt2p)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. **Introducción**
   1. **Propósito del Documento**

El propósito de este documento es especificar de manera formal los requerimientos funcionales y no funcionales del sistema **AODA**, desarrollado para el **Servicio de Salud Metropolitano**.

Este documento está dirigido al equipo de desarrollo, analistas, docentes supervisores y representantes institucionales que participen en el diseño, implementación y validación del sistema.

* 1. **Ámbito del Sistema**

El sistema **AODA (Agente de Orientación y Derivación Asistida)** permitirá automatizar el proceso de derivación de víctimas de delitos hacia centros de apoyo pertinentes, considerando variables como comuna, edad, sexo, situación migratoria y tipo de delito.

El sistema no realizará seguimiento posterior a la atención de la víctima ni almacenará información personal sensible.

* 1. **Definiciones, Acronimos y Abreviaturas**

**AODA:** Agente de Orientación y Derivación Asistida

**IA:** Inteligencia Artificial

**API:** Interfaz de Programación de Aplicaciones

**ERS:** Especificación de Requerimientos de Software

**KPI:** Indicador Clave de Desempeño

* + 1. **Perspectiva del producto**

AODA es un sistema **cliente–servidor** con **frontend en React**, **backend en FastAPI**, **PostgreSQL** como base de datos y un **componente IA (OpenAI)** para recomendación de derivaciones. Se integra internamente entre sus módulos vía **API REST** y opera como capa de apoyo a la gestión institucional del Servicio de Salud, sin integraciones externas obligatorias en la versión inicial.

* + 1. **Funciones del producto**
* **Ingreso de casos:** formulario para comuna, edad, sexo, situación migratoria, tipo de delito y observaciones.

* **Recomendación IA:** procesamiento de criterios y propuesta del centro más pertinente.

* **Confirmación de derivación:** el funcionario acepta/ajusta la sugerencia y registra el envío.

* **Trazabilidad e historial:** registro de cada acción y estado de las derivaciones.

* **Gestión de usuarios y roles:** administrador, supervisor y funcionario.

* **Dashboard/Reportes:** KPIs de tiempos de respuesta, distribución geográfica, frecuencias y capacidad de centros.

* **Catálogo de centros:** mantenimiento de centros, comunas, tipos de servicio y disponibilidad.

* + 1. **Características del usuario**
* **Funcionarios derivadores:** usuarios operativos; manejo básico de plataformas web; foco en rapidez y claridad del flujo.
* **Supervisores/Encargados:** generan reportes, revisan trazabilidad y validan ajustes de criterios.
* **Administradores TI:** gestionan usuarios, catálogos y parámetros técnicos.
* **Perfil técnico general:** alfabetización digital media; no se requieren conocimientos de programación.

* + 1. **Limitaciones**
* **Datos personales:** no se almacenan identificadores sensibles (p. ej., nombre, RUT).
* **Tiempo y alcance:** desarrollo en 10 semanas; sin integraciones gubernamentales externas en la versión inicial.
* **Disponibilidad y conectividad:** depende de Internet y de la disponibilidad del servicio de IA.
* **Recursos:** infraestructura acotada; se prioriza escalado horizontal básico y optimización de consultas.
* **Cumplimiento institucional:** sujeción a políticas de seguridad, privacidad y lineamientos del Servicio de Salud.
* **Cobertura funcional:** no contempla seguimiento clínico posterior ni gestión de atención en el centro receptor (solo derivación y trazabilidad).

* 1. **Referencias**
* Acta de Constitución del Proyecto AODA
* Minuta Kick Off del Proyecto
* Documento de Arquitectura del Sistema (DAS)
* Levantamiento de Requerimientos v1.1
* Normas ISO/IEC/IEEE 29148 de Ingeniería de Requisitos
  1. **Visión General del Documento**

Este documento describe los objetivos, funcionalidades, usuarios, restricciones y requerimientos del sistema AODA, organizados en secciones que detallan la descripción general, los requisitos específicos y la planificación del desarrollo.

1. **Descripción General**
   1. **Perspectiva del Producto**

AODA se basa en una arquitectura cliente-servidor con integración de IA.

Incluye un **frontend en React**, un **backend en FastAPI**, una **base de datos PostgreSQL**, y un **agente IA (OpenAI)** encargado de procesar los datos de entrada y generar recomendaciones de derivación.

* 1. **Funciones del Producto**
* Ingreso de datos de casos por funcionarios.
* Procesamiento de la información mediante IA.
* Recomendación automática del centro de apoyo más adecuado.
* Registro histórico de derivaciones.
* Dashboard de indicadores y métricas.
* Gestión de usuarios y roles.
  1. **Características de los usuarios**

1. **Funcionarios:** usuarios principales, encargados de ingresar casos y realizar derivaciones.
2. **Supervisores:** validan las derivaciones y generan reportes institucionales.
3. **Administradores:** gestionan usuarios, centros y parámetros de derivación.

Nivel técnico: medio. Los usuarios no requieren conocimientos de programación, pero sí manejo básico de plataformas web.

**2.4.Restricciones**

Las siguientes limitaciones se imponen sobre el desarrollo del sistema **AODA (Agente de Orientación y Derivación Asistida)**:

**Políticas de la empresa:** El desarrollo debe alinearse con las políticas del **Servicio de Salud Metropolitano**, cumpliendo con los lineamientos de confidencialidad, trazabilidad institucional y seguridad de la información.

**Limitaciones del hardware:** El sistema está pensado para funcionar en equipos institucionales estándar, con recursos mínimos de **4 GB de RAM y procesador i3 o superior**, por lo que no se contemplan requerimientos de hardware avanzado ni infraestructura dedicada.

**Interfaces con otras aplicaciones:** En la versión inicial no se integrará con sistemas externos. Sin embargo, el diseño de la API REST dejará abierta la posibilidad de conectar el sistema con plataformas públicas futuras (ej. ChileAtiende o bases de datos ministeriales).

**Operaciones paralelas:** El sistema permitirá el uso concurrente de múltiples usuarios, pero la cantidad simultánea estará limitada a aproximadamente **20–30 conexiones activas** en su primera versión.

**Funciones de auditoría:** El sistema incluirá un **registro de auditoría** que documente todas las acciones realizadas por los usuarios (ingresos, modificaciones y derivaciones) con el fin de garantizar trazabilidad y transparencia.

**Funciones de control:** Se implementarán **roles de usuario diferenciados** (administrador, supervisor, funcionario) para asegurar que solo usuarios autorizados puedan realizar operaciones críticas o modificar parámetros del sistema.

**Lenguajes de programación:** El desarrollo se realizará en **Python (FastAPI)** para el backend y **JavaScript (React)** para el frontend, asegurando compatibilidad, escalabilidad y facilidad de mantenimiento.

**Protocolos de comunicación:** La comunicación entre los módulos del sistema y con la base de datos se realizará mediante **API REST** bajo protocolo **HTTPS**, garantizando cifrado de extremo a extremo.

**Requisitos de habilidad:** Los desarrolladores deberán contar con conocimientos en **Python, React, SQL y consumo de APIs REST**, además de experiencia básica en integración con servicios de inteligencia artificial.

**Criticidad de la aplicación:** El sistema es de **criticidad media**, dado que apoya procesos institucionales relevantes, pero no interviene directamente en atención médica o de emergencia.

**Consideraciones acerca de la seguridad:** No se almacenarán datos personales sensibles. Todas las sesiones requerirán autenticación, y se implementarán buenas prácticas de seguridad en el manejo de contraseñas, conexiones y logs.

**2.5.Suposiciones y Dependencias**

**Suposiciones organizacionales:**

Se asume que el **Servicio de Salud Metropolitano** mantendrá su estructura actual de atención a víctimas y los protocolos de derivación vigentes durante el desarrollo y despliegue del sistema.

Se asume la **colaboración continua de los funcionarios institucionales** en la validación de los criterios de derivación y en las pruebas piloto del sistema.

Se presume que el proyecto contará con el **apoyo del área de informática** del Servicio para la implementación técnica y mantenimiento posterior.

**Suposiciones técnicas:**

El sistema funcionará sobre **navegadores modernos** (Chrome, Edge o Firefox) y equipos de escritorio o portátiles con conexión estable a Internet.

Se asume la **disponibilidad y estabilidad de los servicios externos** utilizados, especialmente la API de **OpenAI** para el componente de inteligencia artificial.

Se presupone el uso de una infraestructura de hosting compatible con **FastAPI**, **PostgreSQL** y **React**, con soporte para HTTPS y almacenamiento persistente.

**Dependencias de software:**

Dependencia directa del framework **FastAPI** para la comunicación backend–frontend.

Dependencia de **PostgreSQL** como motor de base de datos relacional.

Dependencia de la **API de IA (OpenAI)**, cuyo rendimiento y costo pueden variar según cambios en políticas del proveedor.

**Dependencias de gestión:**

Dependencia de la **planificación y control de tareas en Jira**, que se usará para coordinar avances y cambios.

Dependencia del **tiempo disponible del equipo técnico y funcional**, considerando que el proyecto tiene una duración máxima de 10 semanas.

Si cualquiera de estas condiciones se modifica —por ejemplo, cambios en la estructura institucional, en la política de acceso a datos o en las herramientas tecnológicas— será necesario **revisar y ajustar los requisitos del sistema** para mantener la coherencia con los objetivos y las capacidades del entorno operativo.

**2.6.Requisitos Futuros**

* Integración con sistemas externos del Estado (ChileAtiende, PISEE).
* Implementación de alertas automáticas ante casos urgentes.
* Módulo de estadísticas históricas avanzadas con filtros personalizados.

**3.Requisitos Específicos**

Esta sección describe los requisitos detallados del sistema **AODA**, que permitirán al equipo de diseño e implementación construir una solución que cumpla con las necesidades del **Servicio de Salud Metropolitano**.

Todos los requisitos definidos en este documento son **corregibles, verificables, trazables y consistentes**, y se han identificado mediante un código único para facilitar su gestión.

**3.1. Requisitos de las Interfaces**

El sistema contará con una **interfaz web responsiva**, accesible desde navegadores modernos (Chrome, Edge y Firefox).  
 La interfaz debe ser **intuitiva, limpia y funcional**, permitiendo al usuario navegar entre módulos sin requerir conocimientos técnicos avanzados.  
 Las principales pantallas del sistema serán:

**3.1.1.Interfaces de Usuario**

* **Pantalla de inicio de sesión:** autenticación mediante usuario y contraseña, con validación de credenciales.
* **Panel de ingreso de casos:** formulario estructurado para ingresar comuna, edad, sexo, situación migratoria, tipo de delito y observaciones adicionales.
* **Pantalla de recomendación IA:** muestra los resultados sugeridos por el agente, con detalles del centro (nombre, comuna, dirección, tipo de servicio y contacto).
* **Módulo de confirmación de derivación:** permite aceptar, modificar o rechazar la recomendación y registrar la derivación final.
* **Historial de casos:** visualización de derivaciones anteriores con filtros por fecha, comuna o funcionario.
* **Dashboard de indicadores:** visualización de métricas clave (número de derivaciones, tiempos promedio, distribución geográfica, disponibilidad de centros).

La interfaz seguirá la paleta institucional del Servicio de Salud (tonos azules y grises) y deberá mantener coherencia visual en todos los módulos.

**3.1.2. Interfaces de Hardware**

.El sistema no requiere hardware especializado.  
 Los requisitos mínimos para los equipos que utilicen el sistema son:

* **Procesador:** Intel i3 o superior.
* **Memoria RAM:** mínimo 4 GB.
* **Almacenamiento:** 1 GB disponible para caché y archivos temporales.
* **Conectividad:** acceso estable a Internet (mínimo 10 Mbps).
* **Resolución recomendada:** 1366x768 píxeles o superior.

El servidor de despliegue deberá contar con las siguientes características:

1. **CPU:** 2 núcleos.
2. **RAM:** 4 GB.
3. **Base de datos PostgreSQL** instalada localmente o en la nube.

**3.1.3. Interfaces de Software**

Indicar si hay que integrar el producto con otros productos de software. Para cada producto de software debe especificarse lo siguiente:

* Descripción del producto software utilizado
* Propósito del interfaz
* Definición del interfaz: contenido y formato

**3.1.4. Interfaces de Comunicación**

El sistema utilizará una arquitectura **cliente–servidor**, basada en **comunicación RESTful sobre protocolo HTTPS**, garantizando la confidencialidad de los datos.

* **Protocolo de transporte:** HTTPS (TLS 1.2 o superior).
* **Formato de intercambio de datos:** JSON UTF-8.
* **Autenticación:** mediante token JWT (JSON Web Token) en cada solicitud.
* **Puertos utilizados:**
  + 443 (producción)
  + 8000 (entorno de desarrollo local).  
     Todas las comunicaciones estarán cifradas y auditadas mediante registros automáticos en el backend.

**3.2. Requisitos Funcionales**

### RF1. Ingreso de datos del caso

El sistema debe permitir al funcionario ingresar los datos relevantes de cada caso, tales como comuna, edad, sexo, situación migratoria, tipo de delito y cualquier otro antecedente necesario para la derivación.

### RF2. Procesamiento con agente IA

El sistema debe aplicar un modelo de inteligencia artificial que analice los datos ingresados y determine el centro de apoyo más adecuado, basándose en los criterios definidos por el Servicio de Salud Metropolitano.

### RF3. Visualización de resultados

El sistema debe mostrar al funcionario la recomendación del centro de apoyo, incluyendo nombre, dirección, teléfono, tipo de servicio y nivel de especialización.

### RF4. Filtro de disponibilidad y pertinencia

El sistema debe verificar que el centro recomendado se encuentre disponible, operativo y que el tipo de atención corresponda al perfil del caso (psicológica, jurídica, social, etc.).

### RF5. Confirmación de derivación

El funcionario debe poder confirmar, modificar o rechazar la sugerencia entregada por el agente IA antes de registrar la derivación final en el sistema.

### RF6. Historial y trazabilidad de casos

El sistema debe registrar cada caso procesado, incluyendo la información del funcionario que realizó la derivación, la fecha y el centro recomendado, permitiendo la trazabilidad y auditoría de los registros.

### RF7. Gestión de usuarios y roles

El sistema debe permitir la creación de cuentas con diferentes niveles de acceso (administrador, funcionario, supervisor), controlando los permisos de visualización y edición.

### RF8. Generación de reportes y dashboard

El sistema debe permitir visualizar y exportar reportes mediante un dashboard que muestre indicadores clave como: número de derivaciones, tiempos de respuesta, distribución geográfica y disponibilidad de los centros.

### RF9. Registro de auditoría

El sistema debe guardar información de cada acción realizada (inicio de sesión, ingreso de caso, derivación, modificación o eliminación) para fines de control y cumplimiento institucional.

*.*

**3.3. Requisitos no Funcionales**

**3.3.1. Requisitos de Rendimiento**

* El tiempo máximo de respuesta del sistema no deberá superar los **5 segundos** para operaciones de derivación.
* El sistema debe soportar **20–30 usuarios concurrentes** en la versión inicial.
* La base de datos debe poder almacenar al menos **50.000 registros históricos** sin degradar el rendimiento.
* El tiempo de carga del dashboard no debe exceder los **3 segundos** bajo condiciones normales.

**3.3.2. Seguridad**

* El sistema implementará **autenticación obligatoria** y **gestión de sesiones seguras**.
* Toda la comunicación se realizará bajo **protocolo HTTPS**.
* Se aplicará **cifrado de contraseñas (bcrypt o similar)** y manejo de tokens JWT.
* No se almacenarán datos personales sensibles (nombre, RUT, dirección exacta).
* Solo los usuarios con rol de administrador podrán modificar parámetros o eliminar registros.
* Se mantendrá un **registro de auditoría** con las acciones realizadas por cada usuario.

**3.3.3. Fiabilidad**

* El sistema deberá garantizar una **tasa de disponibilidad mínima del 95%** durante el horario laboral.
* En caso de falla, el servicio debe poder recuperarse sin pérdida de datos mediante respaldos automáticos diarios.
* Las operaciones críticas (como derivaciones confirmadas) deben confirmarse con doble validación en el backend.

**3.3.4. Disponibilidad**

* El sistema debe estar accesible desde cualquier punto con conexión a Internet.
* Los tiempos de inactividad programados (mantenimiento o actualizaciones) deberán comunicarse con al menos 24 horas de anticipación.
* El sistema deberá permitir **acceso 24/7**, aunque su uso principal será dentro del horario institucional.

**3.3.5. Mantenibilidad**

* El sistema **AODA** debe estar diseñado para permitir la actualización, corrección y mejora de sus componentes sin afectar la operación general.
* El código fuente debe estar **documentado y versionado** en un repositorio controlado (GitHub o similar).
* Las dependencias y configuraciones deberán estar claramente definidas para facilitar el despliegue en nuevos entornos (archivos *requirements.txt* y *package.json*).
* El proceso de mantenimiento será gestionado mediante **procedimientos de control de versiones y pruebas de regresión**, garantizando la estabilidad del sistema tras cada cambio.
* Las actualizaciones podrán realizarse sin interrumpir el servicio, utilizando **integración y despliegue continuo (CI/CD)**.
* Se designará un responsable técnico o equipo de TI institucional encargado del seguimiento y la documentación de cambios.

**Tipo de Mantenimiento:**

* **Correctivo:** solución de errores detectados durante el uso o las pruebas.
* **Adaptativo:** ajustes derivados de cambios en la infraestructura o en las políticas institucionales.
* **Perfectivo:** incorporación de mejoras o ampliaciones funcionales sugeridas por los usuarios.
* **Preventivo:** optimización de rendimiento, actualización de dependencias o eliminación de posibles vulnerabilidades.

**Periodicidad:**

* El sistema deberá contar con **revisiones de mantenimiento mensuales** y reportes técnicos que detallen errores corregidos, actualizaciones aplicadas y propuestas de mejora.

**3.3.6. Portabilidad**

* El sistema debe poder ser migrado entre servidores con sistemas operativos **Windows o Linux** sin necesidad de reescribir el código.
* El uso de **Python (FastAPI)** y **React** garantiza la independencia de plataforma.
* Las dependencias estarán definidas en archivos **requirements.txt** (backend) y **package.json** (frontend).
* El sistema será compatible con contenedores **Docker**, facilitando su implementación en distintos entornos.

**3.4. Otros requisitos**

* La interfaz y documentación deben estar disponibles en idioma español.
* Se incluirá un manual de usuario y un manual técnico para el personal de TI.
* El sistema debe cumplir con las normas básicas de accesibilidad web (WCAG 2.1).
* Las pruebas y validaciones deben quedar documentadas con capturas y reportes técnicos.

**3.5. Requisitos de Base de Datos Lógica.**

| **Entidad** | **Descripción** | **Campos principales** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**4. Propuesta de Planificación**

**4.1. Descripción general acerca de la planificación**

El desarrollo del sistema **AODA (Agente de Orientación y Derivación Asistida)** se abordará mediante una planificación estructurada en **10 semanas de trabajo**, con un enfoque **tradicional en la documentación** y **ágil en la gestión de tareas**, utilizando **Jira** como herramienta de control y seguimiento.

El proyecto contempla una fase inicial de análisis y diseño, seguida de ciclos iterativos de desarrollo y validación. Cada semana se realizarán reuniones de revisión para evaluar los avances, gestionar cambios y asegurar la coherencia con los objetivos institucionales del **Servicio de Salud Metropolitano**.

**Duración total estimada:** 10 semanas (70 días hábiles).  
 **Modalidad de trabajo:** combinada (presencial y virtual).  
 **Revisiones periódicas:** semanales, con retroalimentación de los representantes institucionales.

**Buenas prácticas aplicadas:**

* Uso de control de versiones (GitHub) y documentación continua.
* Validación de entregables en cada iteración.
* Aplicación de pruebas funcionales y revisiones de código cruzadas.
* Mantenimiento de trazabilidad entre requerimientos, desarrollo y pruebas.
* Comunicación continua a través de canales institucionales (Discord, correo y Jira).

**Condiciones necesarias para el éxito del proyecto:**

* Compromiso de los participantes y claridad en la definición de roles.
* Acceso oportuno a la información sobre centros de apoyo y protocolos institucionales.
* Disponibilidad de recursos técnicos mínimos (infraestructura y conectividad).
* Cumplimiento de los plazos establecidos para las entregas parciales y finales.

**4.1.1. Definición del Equipo de Trabajo**

[Describir el equipo de trabajo definido para el Proyecto e insertar Tabla de definición de Roles y funciones]

| **Recurso** | **Rol/Función** | **Departamento/Division** |
| --- | --- | --- |
| **Ignacio Martinez** | **Gestor del Proyecto / Analista Funcional** | | **Coordinación General / Informática** | | --- |  |  | | --- | |
| **Gabriel Fernandez** | |  | | --- |  | **Diseño e implementación de interfaz web en React** | | --- | | **Tecnología / Desarrollo** |
| **Benjamin Cortez** | |  | | --- |  | **Implementación de servicios API, conexión con IA y base de datos** | | --- | | | **Tecnología / Desarrollo** | | --- |  |  | | --- | |
| **Representante Servicio Salud** | **Validación funcional y revisión de criterios de derivación** | **Área Institucional / Gestión de Casos** |

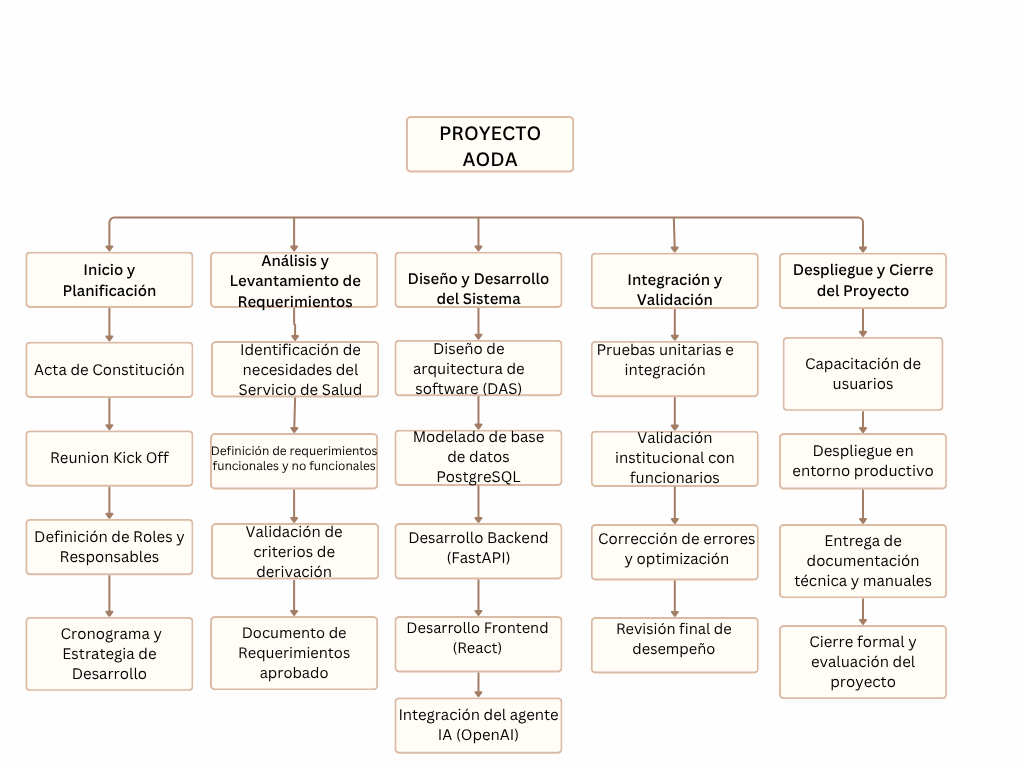
**4.1.2. Definición de Actividades principales del Proyecto**

El desarrollo del sistema **AODA** se estructura en una secuencia de actividades que abarcan desde la planificación inicial hasta el despliegue final del software.

Cada actividad cuenta con objetivos definidos, responsables asignados y productos esperados, garantizando la trazabilidad entre los requerimientos, el desarrollo y las pruebas.

| **Fase / Actividad Principal** | **Descripción** | **Responsable** | **Entregables / Resultados Esperados** |
| --- | --- | --- | --- |
| **1. Análisis y levantamiento de requerimientos** | Recolección, revisión y validación de requerimientos funcionales y no funcionales, junto con los criterios de derivación. | Gestor del Proyecto / Analista Funcional | Documento de levantamiento de requerimientos validado. |
| **2. Diseño de arquitectura del sistema** | Definición de la estructura del sistema, componentes, base de datos y flujos de comunicación. | Desarrollador Backend / Analista Funcional | Documento DAS (Arquitectura de Software) y modelo ER de base de datos. |
| **3. Configuración de entorno y herramientas** | Instalación de dependencias, configuración de repositorios Git y ambientes de desarrollo. | Equipo de Desarrollo | Entorno funcional de desarrollo y control de versiones operativo. |
| **4. Desarrollo del backend (FastAPI)** | Implementación de API REST, conexión con base de datos y modelo de IA. | Desarrollador Backend | API operativa con endpoints documentados y pruebas iniciales. |
| **5. Desarrollo del frontend (React)** | Creación de la interfaz web para ingreso de casos, visualización de recomendaciones y dashboard. | Desarrollador Frontend | Interfaz funcional integrada con la API. |
| **6. Integración del agente IA (OpenAI)** | Conexión del backend con el modelo IA para procesar criterios y generar recomendaciones. | Asesor de IA / Desarrollador Backend | Funcionalidad IA integrada y validada. |
| **7. Pruebas funcionales y de integración** | Validación de módulos individuales y pruebas de interoperabilidad entre backend, frontend e IA. | Equipo QA / Desarrolladores | Informe de pruebas con resultados, errores y correcciones aplicadas. |
| **8. Validación institucional** | Presentación del sistema a representantes del Servicio de Salud para retroalimentación y ajustes. | Gestor del Proyecto / Representante Institucional | Registro de observaciones y mejoras incorporadas. |
| **9. Documentación y capacitación** | Elaboración de manuales técnico y de usuario, y sesiones de capacitación a funcionarios. | Analista Funcional / Equipo QA | Manuales entregados y registro de capacitación. |
| **10. Despliegue y cierre del proyecto** | Instalación del sistema en entorno productivo, verificación final y cierre formal. | Equipo de Desarrollo / Gestor del Proyecto | Sistema operativo, informe de cierre y aceptación institucional. |

**4.1.3. Diagrama EDT**

****

**4.1.4. Carta Gantt**

| **Fase / Actividad** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1. Inicio y Planificación del Proyecto** | **█████** | **█** |  |  |  |  |  |  |  |  |
| **1.1 Acta de Constitución y Kick-Off** | **███** | **█** |  |  |  |  |  |  |  |  |
| **1.2 Definición de roles y cronograma** | **█** | **█** |  |  |  |  |  |  |  |  |
| **2. Análisis y Levantamiento de Requerimientos** |  | **█████** | **█** |  |  |  |  |  |  |  |
| **2.1 Identificación de requerimientos** |  | **███** |  |  |  |  |  |  |  |  |
| **2.2 Validación institucional** |  | **█** | **█** |  |  |  |  |  |  |  |
| **3. Diseño y Desarrollo del Sistema** |  |  | **█████** | **█████** | **███** |  |  |  |  |  |
| **3.1 Diseño de arquitectura y base de datos** |  |  | **███** |  |  |  |  |  |  |  |
| **3.2 Desarrollo Backend (FastAPI)** |  |  | **███** | **███** | **█** |  |  |  |  |  |
| **3.3 Desarrollo Frontend (React)** |  |  |  | **███** | **███** |  |  |  |  |  |
| **3.4 Integración del Agente IA** |  |  |  | **█** | **███** |  |  |  |  |  |
| **4. Integración y Validación** |  |  |  |  | **███** | **███** | **█** |  |  |  |
| **4.1 Pruebas unitarias e integración** |  |  |  |  | **█** | **███** |  |  |  |  |
| **4.2 Validación institucional** |  |  |  |  |  | **█** | **███** |  |  |  |
| **5. Despliegue y Cierre del Proyecto** |  |  |  |  |  |  | **███** | **███** | **█** | **█** |
| **5.1 Capacitación y documentación** |  |  |  |  |  |  | **█** | **███** |  |  |
| **5.2 Despliegue y cierre** |  |  |  |  |  |  |  | **███** | **███** | **█** |

**4.1.5. Resumen de Costos del Desarrollo del Proyecto7**

|  | |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase del Proyecto** | **Rol / Recurso Principal** | **Horas Estimadas** | **Valor Hora (CLP)** | **Costo Parcial (CLP)** | **Descripción de Actividades Principales** |
| **1. Inicio y Planificación** | Gestor del Proyecto / Analista Funcional | 20 | $7.500 | $150.000 | Elaboración del acta, reunión Kick-Off, definición de roles y cronograma. |
| **2. Análisis y Requerimientos** | Gestor del Proyecto / Analista Funcional – Representante Institucional | 30 | $7.500 | $250.000 | Levantamiento y validación de requerimientos funcionales y no funcionales. |
| **3. Diseño y Desarrollo** | Desarrollador Backend – Desarrollador Frontend – Asesor IA | 220 | Promedio $6.500 | $1.200.000 | Desarrollo de backend (FastAPI), frontend (React) e integración con IA. |
| **4. Integración y Validación** | Equipo QA – Desarrolladores – Representante Institucional | 80 | Promedio $5.500 | $300.000 | Pruebas funcionales, control de calidad, validación institucional y corrección de errores. |
| **5. Despliegue y Cierre** | Gestor del Proyecto – QA – Analista Funcional | 50 | Promedio $6.000 | $170.000 | Capacitación, documentación técnica y entrega final del sistema. |
| **—** | **Total General del Proyecto AODA** | **400 h** | — | **$2.070.000 CLP** | — |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

| Costos por fase | |  |  |
| --- | --- | --- | --- |

| **Recurso / Concepto** | **Descripción** | **Costo (CLP)** |
| --- | --- | --- |
| Infraestructura tecnológica | Hosting, base de datos y servicios en la nube | $150.000 |
| Licencias y API IA (OpenAI) | Uso de API para pruebas y consultas | $100.000 |
| Documentación y manuales | Diseño, redacción y edición de entregables | $50.000 |
| **Total costos generales** |  | **$300.000** |

### TOTAL GENERAL DEL PROYECTO

- **$2.370.000 CLP**

Nota: Más especificado en el documento EDT

**4.2. Plan de control de Cambio**

El **Plan de Control de Cambio** tiene como propósito establecer un procedimiento formal para gestionar cualquier modificación en el alcance, requerimientos, diseño o implementación del sistema **AODA (Agente de Orientación y Derivación Asistida)**.  
 Este proceso garantiza que todos los cambios propuestos sean **evaluados, aprobados y documentados**, manteniendo la coherencia del proyecto con los objetivos del **Servicio de Salud Metropolitano**.

El control de cambios es una **actividad paralela y continua** al desarrollo del software, aplicable tanto durante las fases activas del proyecto como en etapas posteriores de mantenimiento o actualización.

**Tipos de cambios y sus alcances**

| **Tipo de cambio** | **Descripción** | **Ejemplo** | **Instancia en que puede aplicarse** | **Aprobación requerida** |
| --- | --- | --- | --- | --- |
| **Funcional** | Modificación, eliminación o incorporación de una funcionalidad existente o nueva dentro del sistema. | Agregar un campo adicional en el formulario de ingreso de casos. | Durante las fases de desarrollo y validación. | Gestor del Proyecto y Representante Institucional. |
| **No funcional** | Ajustes relacionados con rendimiento, usabilidad, seguridad o disponibilidad sin alterar la lógica principal del sistema. | Mejorar tiempos de respuesta del dashboard. | Durante desarrollo o mantenimiento. | Equipo Técnico. |
| **Correctivo** | Cambio derivado de errores o fallas detectadas durante pruebas o en producción. | Corrección de validaciones erróneas en formularios. | Durante pruebas o post-despliegue. | QA y Gestor del Proyecto. |
| **Adaptativo** | Modificación necesaria por cambios en políticas institucionales, infraestructura o normativas externas. | Ajuste en los criterios de derivación según nuevo protocolo institucional. | En cualquier fase, previa revisión de impacto. | Comité de Control de Cambios. |
| **Perfectivo** | Mejora solicitada por usuarios para optimizar el sistema sin alterar el alcance base. | Rediseño visual de la interfaz para mayor claridad. | Post-despliegue o versiones futuras. | Gestor del Proyecto y Representante Institucional. |

### 

### Procedimiento de control de cambios

1. **Identificación:** Cualquier integrante del equipo o usuario institucional puede detectar una necesidad de cambio.
2. **Solicitud formal:** Se registra una **Solicitud de Cambio (SC)** en la herramienta de gestión (**Jira**), indicando tipo, descripción, motivo e impacto.
3. **Evaluación:** El Gestor del Proyecto y el equipo técnico analizan la factibilidad técnica, el costo y el impacto en los plazos.
4. **Aprobación:** El cambio solo podrá ejecutarse tras la validación y firma del **Representante Institucional** o del **Comité de Control de Cambios**.
5. **Implementación:** El equipo técnico aplica la modificación y la documenta en el repositorio y en la bitácora de cambios.
6. **Verificación:** El equipo QA verifica que la modificación no afecte otras funciones ni degrade el desempeño del sistema.
7. **Cierre:** El cambio es marcado como **“implementado”** en la planilla de control y archivado en la documentación del proyecto.

### Aspectos del desarrollo sujetos a cambios

* **Funcionalidades del sistema:** módulos de ingreso de datos, filtros, dashboard y flujos de derivación.
* **Aspectos no funcionales:** rendimiento, escalabilidad, seguridad, diseño visual y accesibilidad.
* **Configuraciones técnicas:** endpoints, estructuras de base de datos, variables de entorno y dependencias.

No se considerarán cambios:

* Que impliquen modificar la finalidad institucional del sistema.
* Que afecten los principios de privacidad y seguridad definidos en el alcance.
* Que requieran un rediseño completo de la arquitectura del sistema sin aprobación formal.

| **N°** | **Fecha** | **Tipo de cambio** | **Descripción breve** | **Impacto estimado** | **Estado** | **Responsable** | **Aprobación** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | — | — | — | — | — | — | — |
| 2 | — | — | — | — | — | — | — |
| 3 | — | — | — | — | — | — | — |

(la tabla será completada a medida que se realicen cambios durante el transcurso del proyecto, de momento no se han realizado cambios drásticos en lo que va de proyecto)

**5. Anexos**