Documento de Especificacion de Requerimientos

Contents

[Documento de Especificacion de Requerimientos 1](#_Toc176720285)

[Diagramas 2](#_Toc176720286)

[Diagramas de Arquitectura de AWS 2](#_Toc176720287)

[Diagrama de Componentes 4](#_Toc176720288)

[Diagrama de componentes en PlantUML 5](#_Toc176720289)

[Diagrama1 de componentes en PlantUML 6](#_Toc176720290)

[Diagrama en formato Mermaid 7](#_Toc176720291)

[Diagrama de arquitectura con DiagramGPT 8](#_Toc176720292)

[Historias de Usuario 9](#_Toc176720293)

[User Story: Procesamiento Automático para detectar Impedimentos 9](#_Toc176720294)

[User Story: Visualización de Impedimentos en el Frontend 10](#_Toc176720295)

[Tickets de Jira para Integración con Read.ai y Frontend de Visualización 10](#_Toc176720296)

[Ticket 1: Integración con Read.ai 10](#_Toc176720297)

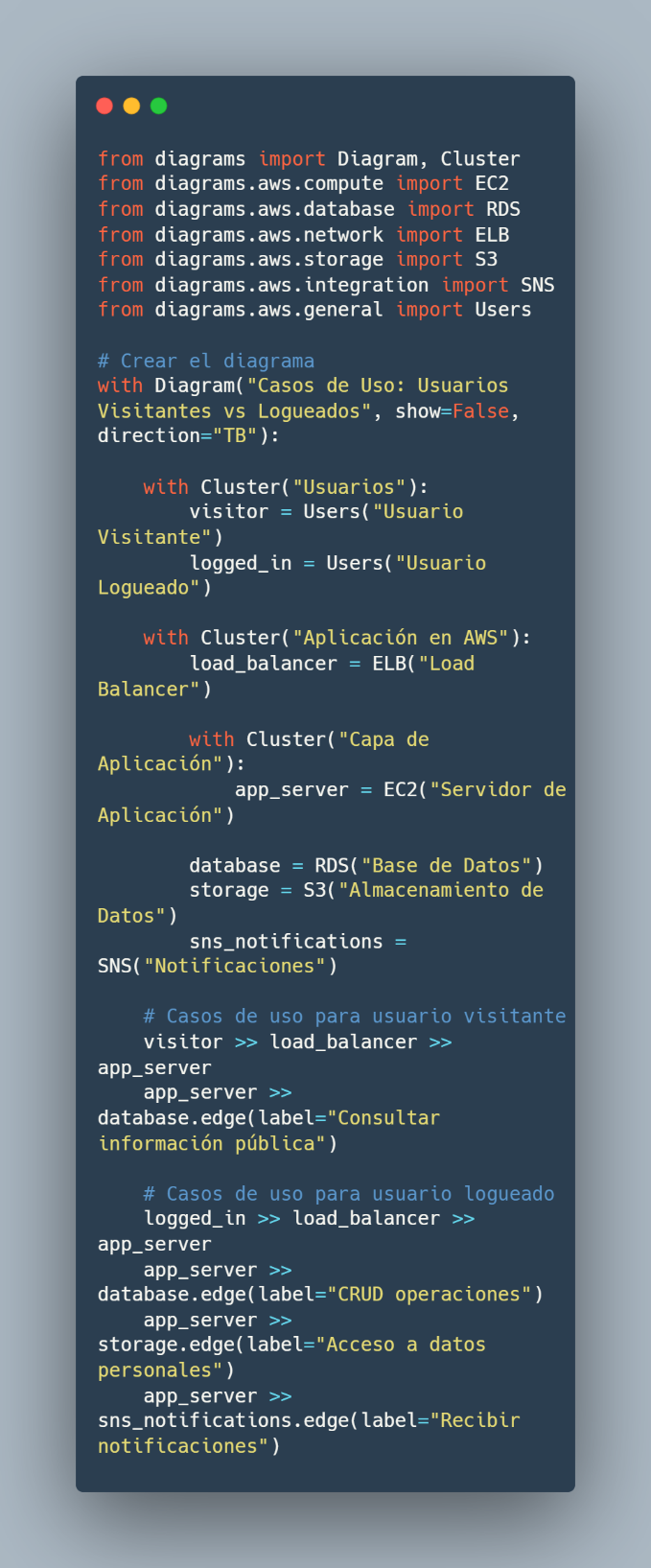
[Ticket 2: Análisis de Transcripciones de Read.ai 11](#_Toc176720298)

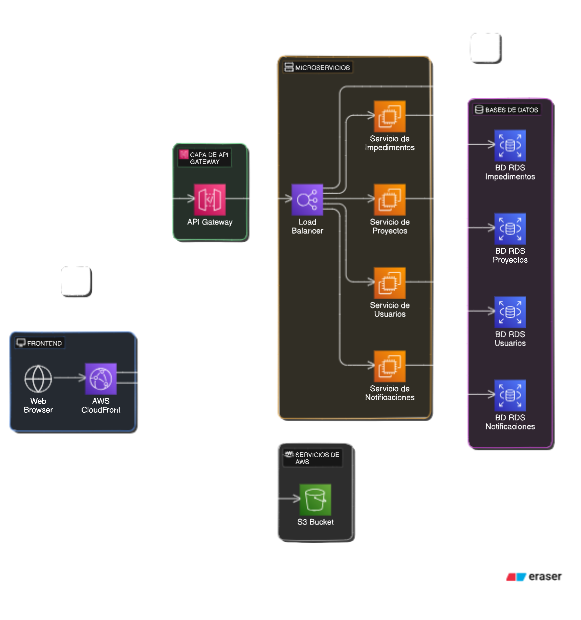
[Ticket 3: Notificaciones de Impedimentos vía Slack 11](#_Toc176720299)

[Ticket 4: Desarrollo del Frontend - Estructura Básica 12](#_Toc176720300)

[Ticket 5: Desarrollo del Frontend - Lista de Impedimentos 13](#_Toc176720301)

Diagramas

Diagramas de Arquitectura de AWS  




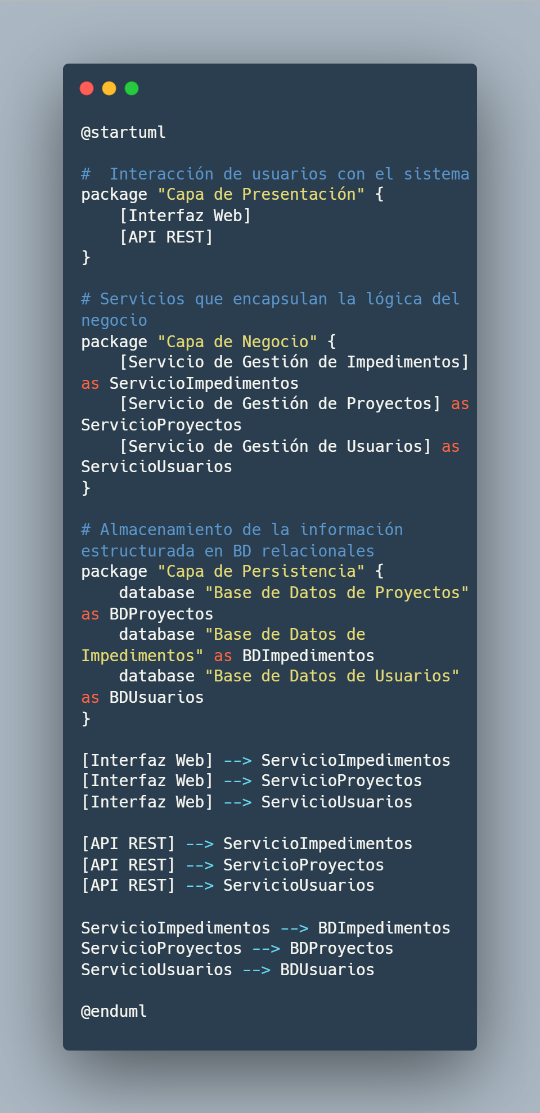
### Diagrama de Componentes

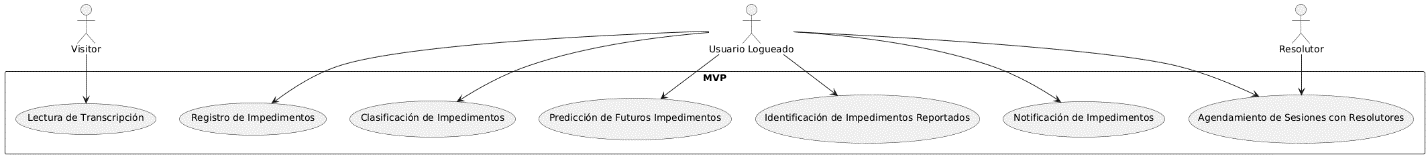


A diagram of a presentation

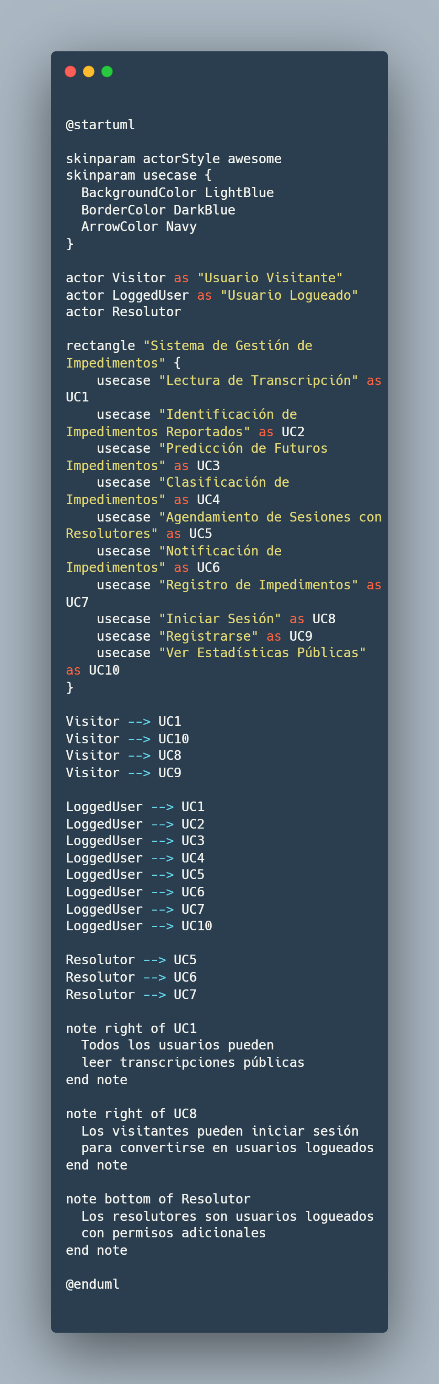
Description automatically generated

### Diagrama de componentes en PlantUML



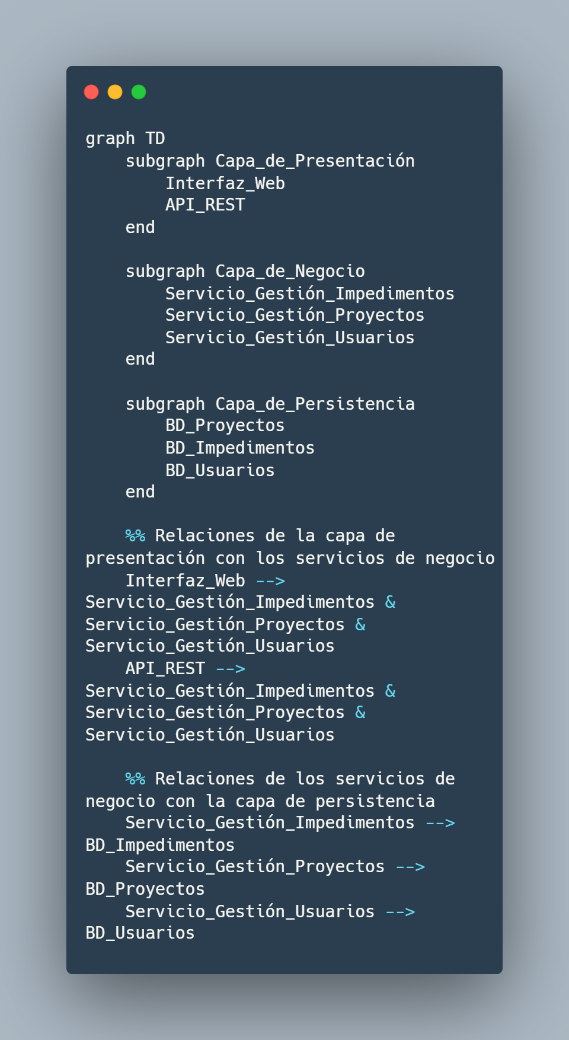


### Diagrama1 de componentes en PlantUML



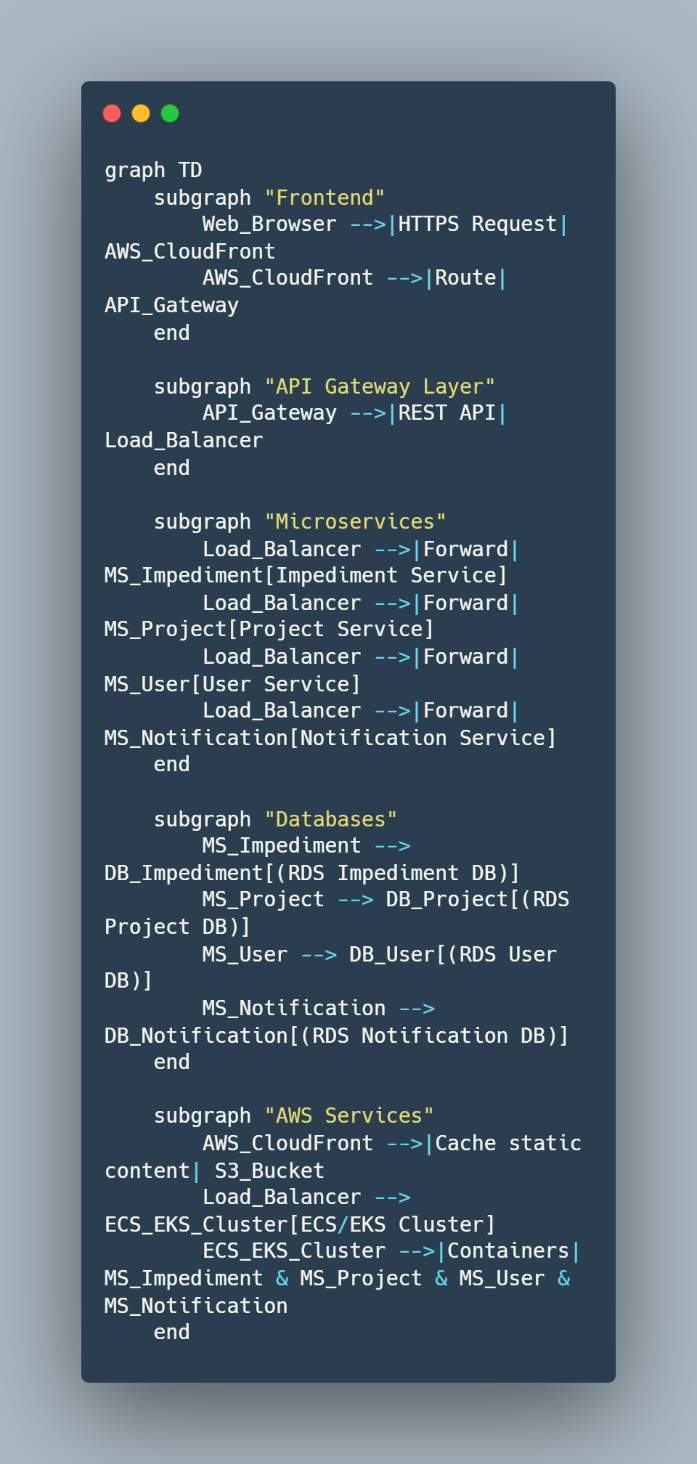


### Diagrama en formato Mermaid

  
  
A diagram of a company

Description automatically generated

### Diagrama de arquitectura con DiagramGPT



A diagram of a company

Description automatically generated

## Historias de Usuario

### User Story: Procesamiento Automático para detectar Impedimentos

**Título**

Procesamiento de Transcripciones de Read.ai para Detectar Impedimentos

**Descripción**

Como miembro del equipo, quiero que la IA analice las transcripciones generadas por Read.ai para identificar impedimentos mencionados o potenciales problemas, de manera que se gestionen de forma anticipada y eficiente.

**Criterios de Aceptación**

1. La IA debe conectarse con Read.ai y acceder a las transcripciones de los dailies.

2. Analizar automáticamente el contenido de las transcripciones para detectar impedimentos reportados o mencionados

3. Los impedimentos identificados deben ser registrados y enviados al Scrum Master y al equipo mediante Slack.

### User Story: Visualización de Impedimentos en el Frontend

**Título**

Visualización de Impedimentos en el Frontend

**Descripción**

Como Scrum Master o Gerente de Proyecto, quiero un frontend web que me permita visualizar los impedimentos detectados, clasificados por proyecto, Project Manager y usuario que reporta el impedimento, para gestionar los problemas de manera eficiente.

**Criterios de Aceptación**

1. El frontend debe mostrar una lista de impedimentos detectados, clasificados por proyecto, Project Manager y usuario que reportó el impedimento.

2. Los impedimentos deben incluir detalles como la descripción del problema, fecha de reporte, estado (pendiente, en progreso, resuelto) y solución sugerida (si aplica).

3. Los usuarios deben poder filtrar los impedimentos por proyecto o estado.

Debe haber una opción para marcar impedimentos como resueltos y actualizar su estado.

4. El diseño debe ser accesible y responsivo en una plataforma web.

## Tickets de Jira para Integración con Read.ai y Frontend de Visualización

### Ticket 1: Integración con Read.ai

I**D del Ticket:** AI-006

**Título del Ticket:** Implementar integración con Read.ai para acceso a transcripciones

**Descripción:** Desarrollar un conector que permita al sistema acceder y recuperar las transcripciones de los dailies generadas por Read.ai.

**Criterios de aceptación:**

  1. El sistema se conecta exitosamente a la API de Read.ai.

  2. Las transcripciones se recuperan automáticamente después de cada daily.

  3. Los errores de conexión o recuperación se manejan adecuadamente y se notifican.

  4. Las transcripciones se almacenan en el sistema para su posterior procesamiento.

**Prioridad:** Alta

**Estimación de esfuerzo:** 20 horas

**Tareas Técnicas:**

  - Investigar y documentar la API de Read.ai

  - Implementar la autenticación con Read.ai

  - Desarrollar el proceso de recuperación de transcripciones

  - Crear un sistema de almacenamiento local para las transcripciones

  - Implementar manejo de errores y logging

**Notas:** Asegurar que se cumplen todos los requisitos de seguridad y privacidad de Read.ai.

### Ticket 2: Análisis de Transcripciones de Read.ai

**ID del Ticket:** AI-007

**Título del Ticket:** Adaptar el modelo de IA para analizar transcripciones de Read.ai

**Descripción:** Modificar el modelo de IA existente para procesar eficazmente las transcripciones generadas por Read.ai y detectar impedimentos.

**Criterios de aceptación:**

  1. El modelo procesa correctamente el formato de transcripción de Read.ai.

  2. Se identifican al menos el 90% de los impedimentos mencionados explícitamente.

  3. El sistema diferencia entre impedimentos reportados y potenciales problemas.

  4. El tiempo de procesamiento no excede los 5 minutos por transcripción.

**Prioridad:** Alta

**Estimación de esfuerzo:** 30 horas

**Tareas Técnicas:**

  - Analizar la estructura de las transcripciones de Read.ai

  - Adaptar el preprocesamiento de datos para el formato de Read.ai

  - Ajustar el modelo de IA para mejorar la detección en este formato

  - Implementar la diferenciación entre impedimentos reportados y potenciales

  - Realizar pruebas de precisión y rendimiento

**Notas:** Considerar las particularidades del formato de Read.ai para optimizar la detección.

### Ticket 3: Notificaciones de Impedimentos vía Slack

**ID del Ticket:** AI-008

**Título del Ticket:** Implementar sistema de notificaciones de impedimentos vía Slack

**Descripción:** Desarrollar un mecanismo para enviar notificaciones automáticas al Scrum Master y al equipo a través de Slack cuando se detecten impedimentos.

**Criterios de aceptación:**

  1. Las notificaciones se envían automáticamente a un canal de Slack designado.

  2. Cada notificación incluye detalles del impedimento (descripción, proyecto, reportador).

  3. Se proporciona un enlace directo al impedimento en el sistema para más detalles.

  4. Las notificaciones se envían dentro de los 10 minutos posteriores a la detección del impedimento.

**Prioridad:** Media

**Estimación de esfuerzo:** 15 horas

**Tareas Técnicas:**

  - Configurar la integración con la API de Slack

  - Diseñar el formato de las notificaciones

  - Implementar la lógica de envío de notificaciones

  - Crear un sistema de cola para manejar las notificaciones

  - Realizar pruebas de integración con Slack

**Notas:** Asegurar que las notificaciones sean configurables por el usuario para evitar sobrecarga de información.

### Ticket 4: Desarrollo del Frontend - Estructura Básica

**ID del Ticket:** FE-001

**Título del Ticket:** Implementar estructura básica del frontend web

**Descripción:** Crear la estructura básica del frontend web para la visualización de impedimentos, incluyendo el esqueleto de la aplicación y la navegación principal.

**Criterios de aceptación:**

  1. Se implementa una estructura de aplicación web usando un framework moderno (por ejemplo, React).

  2. La navegación principal incluye secciones para ver impedimentos por proyecto, Project Manager y usuario.

  3. El diseño es responsivo y se adapta a diferentes tamaños de pantalla.

  4. Se implementa un sistema de autenticación básico.

**Prioridad:** Alta

**Estimación de esfuerzo:** 25 horas

**Tareas Técnicas:**

  - Configurar el proyecto de frontend con el framework elegido

  - Diseñar y implementar la estructura de componentes básica

  - Crear el sistema de enrutamiento para la navegación

  - Implementar un diseño responsivo básico

  - Configurar el sistema de autenticación

**Notas:** Utilizar un sistema de diseño (como Material-UI o Tailwind) para acelerar el desarrollo.

### Ticket 5: Desarrollo del Frontend - Lista de Impedimentos

**ID del Ticket:** FE-002

**Título del Ticket:** Implementar visualización de lista de impedimentos

**Descripción:** Desarrollar la funcionalidad para mostrar una lista de impedimentos con opciones de filtrado y actualización de estado.

**Criterios de aceptación:**

1. La lista muestra impedimentos con detalles como descripción, fecha, estado y solución sugerida.s

  2. Se pueden filtrar impedimentos por proyecto, Project Manager, usuario y estado.

  3. Los usuarios pueden marcar impedimentos como resueltos y actualizar su estado.

  4. La lista se actualiza en tiempo real cuando se realizan cambios.

  5. Se implementa paginación para manejar grandes cantidades de impedimentos.

**Prioridad:** Alta

**Estimación de esfuerzo:** 35 horas

**Tareas Técnicas:**

  - Diseñar e implementar el componente de lista de impedimentos

  - Desarrollar la funcionalidad de filtrado

  - Implementar la actualización de estado de impedimentos

  - Crear la lógica de paginación

  - Integrar con el backend para obtener y actualizar datos en tiempo real

**Notas:** Asegurar que la interfaz sea intuitiva y fácil de usar, considerando la experiencia del usuario.