Universidad Autónoma de Baja California

Ingeniería en computación

Realm Essay - Propuesta de arquitectura

Durán Cárdenas Héctor Jesús 1176461

González Tiang Alán Antonio 1170228

Larrañaga Flores Luis Leonardo 1183087

Núñez López Luis Manuel 1182533

Pérez Solorio Kadir Josafat 1182567

Ingeniería de Software

Prof. José Martín Olguín Espinoza

2024, Mexicali B.C.

Índice

1. Introducción **3**
1.1 Propósito 3
1.2 Alcance 3
1.3 Definiciones, Acrónimos y Abreviaturas 3
2. Representación Arquitectónica **3**
3. Objetivos Arquitectónicos y Restricciones **5**
3.1 Objetivos 5
3.2 Restricciones 5
4. Vista de casos de uso **5**
5. Vista de Componentes **6**
7. Tamaño y desempeño **7**
8. Calidad **7**

1. Introducción

1.1 Propósito

El documento tiene como objetivo definir la estructura del sistema, es decir, establecer cómo están organizados los componentes del software, cómo interactúan entre sí, y cómo se integrarán para formar un sistema cohesivo.

1.2 Alcance

En el documento se especifica que habrá diferentes diagramas los cuales podrán ayudar a entender la arquitectura del sistema, además de que estos diagramas cumpliran con los requerimientos funcionales, y no funcionales lo que ayudará a complementar el documento de requerimientos de software

1.3 Definiciones, Acrónimos y Abreviaturas

UML: El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

API: es una pieza de código que permite a dos aplicaciones comunicarse entre sí para compartir información y funcionalidades. Se usan generalmente en bibliotecas de programación.

2. Representación Arquitectónica

La arquitectura que mejor representa al sistema sería la de microservicio, de las cual se puede obtener escalabilidad,resiliencia,flexibilidad tecnológica,facilidad de Mantenimiento. Este tipo de arquitectura permite tener servicios pequeños e independientes que se comunican entre sí a través de APIs. Cada microservicio es responsable de una funcionalidad específica del sistema y puede ser desarrollado, desplegado, y escalado de manera independiente..

2.1 Vista de Casos de Uso

La vista de casos de uso es una representación de los requisitos funcionales de un sistema, enfocada en cómo los diferentes actores (usuarios o sistemas externos) interactúan con el sistema para lograr sus objetivos. Esta vista se utiliza para capturar y describir las funcionalidades que el sistema debe ofrecer, desde la perspectiva de los usuarios y otras entidades externas que interactúan con él.

2.2 Vista Lógica:

La vista lógica es una de las principales perspectivas en la arquitectura de software que se enfoca en la organización y estructura interna del sistema desde el punto de vista del diseño lógico. Esta vista descompone el sistema en componentes o módulos, describiendo cómo están organizados y cómo interactúan entre sí para cumplir con los requisitos funcionales.

2.3 Vista de Despliegue:

La vista de despliegue es una de las perspectivas clave en la arquitectura de software que se enfoca en cómo los componentes del sistema se distribuyen e instalan en el entorno físico o virtual en el que operan.

2.4 Vista de Datos:

La vista de datos en la arquitectura de software se centra en la organización, estructura, almacenamiento, y acceso a los datos dentro de un sistema. Esta vista es crucial para entender cómo se manejan los datos a lo largo de todo el ciclo de vida de una aplicación, desde su creación y almacenamiento hasta su recuperación y eliminación.

3. Objetivos Arquitectónicos y Restricciones

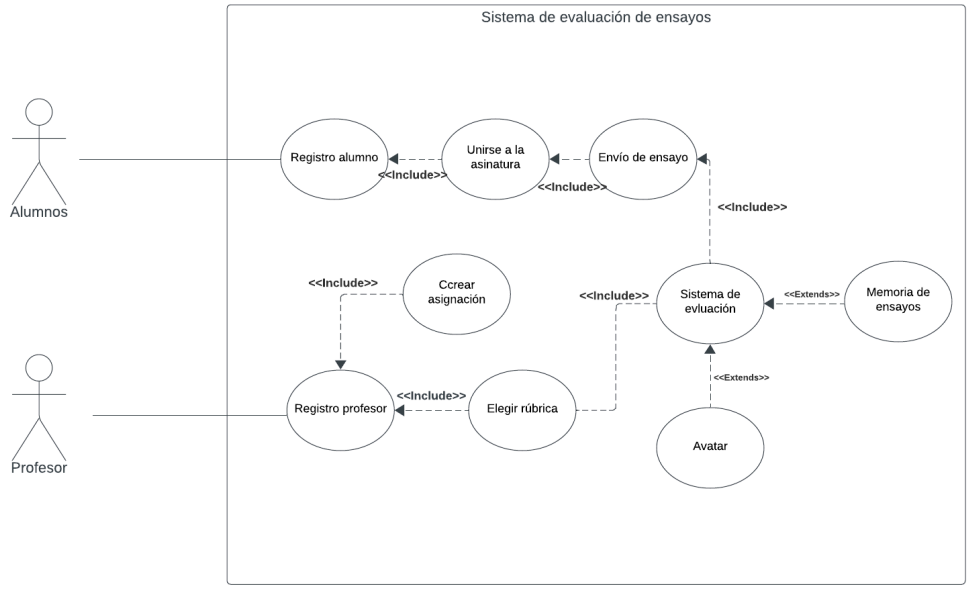
3.1 Objetivos

- Optimizar la velocidad con la que interactúan los componentes.
- Facilitar la corrección de errores.
- Capacidad de cambiar componentes sin problemas.
- implementar autenticación y autorización mediante tokens.

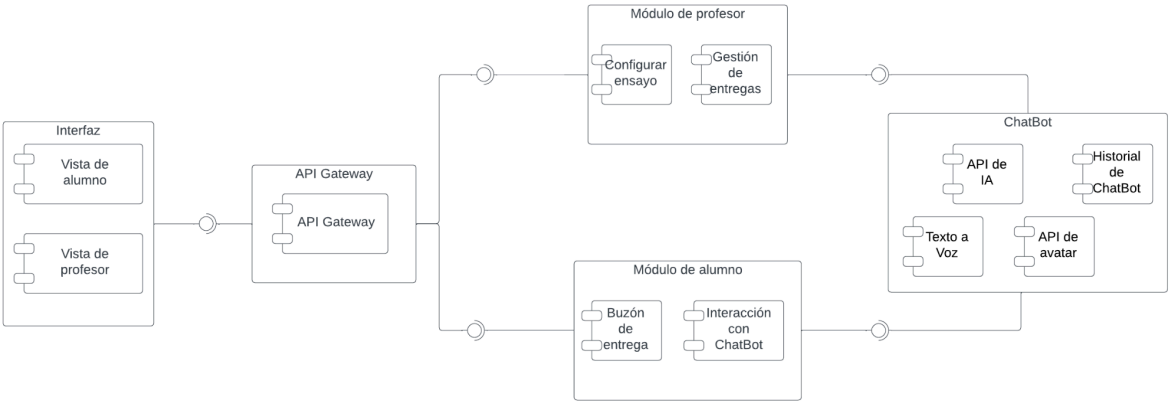
3.2 Restricciones

- No sobrecargar la aplicación
- No dejar datos privados expuestos.

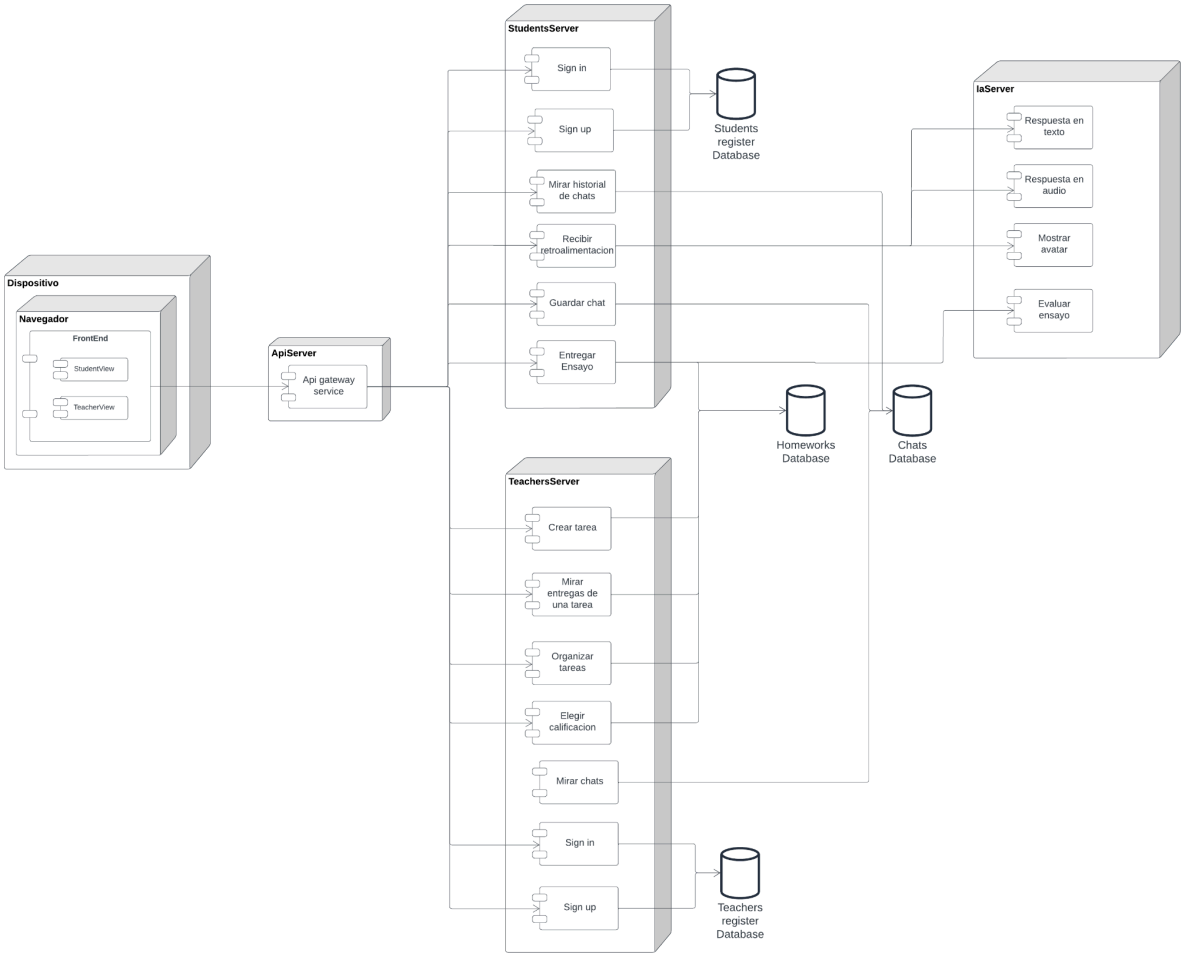
4. Vista de casos de uso



5. Vista de Componentes



6. Vista Despliegue



7. Tamaño y desempeño

Se espera que la aplicación pueda albergar a alumnos de ingeniería en computación, por lo que los usuarios conectados al mismo tiempo que podría aguantar sería alrededor de 200. La idea es que la respuesta del sistema sea rápida, más que nada en la parte del chatbot, donde se espera que pueda calificar el ensayo en menos de 2 minutos.

8. Calidad

- **Seguridad de autentificación:** autenticación: Es necesario que los usuarios se identifiquen antes de usar la aplicación para asegurarse que sólo.
- **Seguridad de datos:** Mantener los ensayos enviados de manera segura con diferentes candados de seguridad para que no sean de fácil acceso.
- **Mantenibilidad:** El sistema debe ser fácil de mantener.