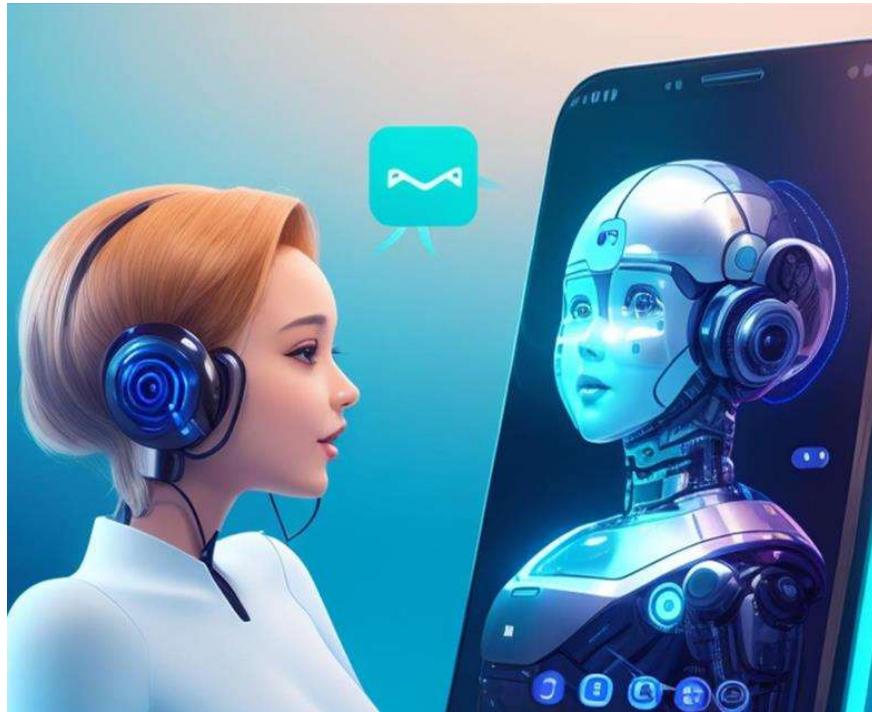


Documento de Propuesta de Diseño de Software I, II y II

TalkBack Julia

Jorge David Hernández Morelo
Camilo Andrés Gómez Bejarano
Adriana Causil García
Tutor: Alexander Toscano Ricardo



Reseña

La propuesta es la integración de un componente "TalkBack" en la plataforma Creavi, con el objetivo de mejorar la accesibilidad para las personas con discapacidad visual en la misma. Este componente permitirá a los usuarios con discapacidad visual navegar por la plataforma utilizando comandos de voz, además de proporcionar lectura audible de los menús principales en pantalla. La idea central es brindar una experiencia de usuario inclusiva y eficiente, lo que facilitará que las personas con discapacidad visual puedan utilizar la plataforma de manera efectiva, accediendo a sus funcionalidades y recursos educativos sin dificultad.

Nivel 1: Diseño de la Aplicación y Análisis de Requisitos

1. Introducción al Diseño de Software
2. Identificación de Necesidades del Usuario
3. Creación de Documentos de Especificación de Requisitos
4. Casos de Uso y Diagramas de Casos de Uso
5. Modelado de Entidad-Relación (E/R)
6. Herramientas para el Diseño de Software (p. ej. Lucidchart, Draw.io)

Nivel 2: Persistencia de Datos con Backend

7. Introducción a la Persistencia de Datos
8. Diseño de la Arquitectura de Backend
9. Elección de una Base de Datos (SQL o NoSQL)
10. Diseño de Tablas y Modelado de Datos
11. Implementación del Backend (p. ej. usando Node.js, Python, Java)
12. Conexión a la Base de Datos y CRUD Operations

Nivel 3: Consumo de Datos y Desarrollo Frontend

13. Introducción al Desarrollo Frontend
14. Creación de Interfaz de Usuario (UI) con HTML y CSS
15. Programación Frontend con JavaScript (JS)
16. Consumo de Datos desde el Backend
17. Interacción Usuario-Interfaz (Eventos, Formularios, Validación)
18. Pruebas y Depuración Frontend
19. Implementación de la Lógica de Negocio en el Frontend

Tabla de contenido

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Reseña..... | 1 |
| Etapa 1 Diseño de la Aplicación y Análisis de Requisitos | 5 |
| 1. Introducción | 5 |
| 1. Descripción General | 6 |
| Usuarios del Sistema..... | 7 |
| Casos de Uso | 8 |
| CASO No. 3 Navegación de la Página | 11 |
| CASO No. 4 Soporte para Múltiples Idiomas | 12 |
| CASO No. 5 Búsqueda por Voz..... | 13 |
| CASO No. 7 Resaltar fragmento de texto por voz | 15 |
| CASO No. 8 Notas por voz | 16 |
| Prioridad de requerimientos..... | 17 |
| Requisitos NO funcionales..... | 18 |
| Requisitos de Desempeño: | 18 |
| 1. Tiempo de Respuesta: | 18 |
| 2. Capacidad de Procesamiento: El sistema debe manejar múltiples solicitudes de voz a la vez sin degradación del rendimiento, soportando al menos 50 usuarios activos en un momento dado. | 18 |
| 3. Consumo de Recursos: | 18 |
| Requisitos de Seguridad: | 18 |
| 4. Acceso Seguro: | 18 |
| 5. Privacidad de Datos: Los datos de los usuarios, incluyendo comandos de voz, deben cifrarse y almacenarse seguramente, cumpliendo con las regulaciones de privacidad vigentes, como GDPR o HIPAA. | 18 |
| 6. Protección contra Ataques: | 18 |
| Requisitos de Usabilidad: | 18 |
| 1. Accesibilidad: | 18 |
| 2. Interfaz Intuitiva: | 18 |
| 3. Personalización: | 18 |
| Requisitos de Escalabilidad: | 19 |

| | |
|-------------------------------------------------------|-----------|
| 1. Adaptabilidad a Crecimiento: | 19 |
| 2. Mantenimiento Sencillo: | 19 |
| 3. Soporte Multiplataforma: | 19 |
| 5. Modelado E/R..... | 20 |
| Etapa 2: Persistencia de Datos con Backend | 23 |
| Etapa 3: Consumo de Datos y Desarrollo Frontend | 32 |

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

1. Introducción

Propósito del proyecto

El presente proyecto se propone integrar un componente de asistencia de pantalla 'TalkBack' a la plataforma educativa Creavi, el objetivo de este componente es optimizar la accesibilidad para individuos con discapacidad visual mediante tecnologías de síntesis voz, que permitirá a los usuarios con discapacidad visual navegar de manera fluida y eficiente por la plataforma a través de comandos de voz personalizados, garantizando una experiencia de usuario enriquecida y accesible. Además, facilitará la navegación, puesto que el componente 'TalkBack' proporcionará una lectura de voz de los elementos de la interfaz de usuario, incluyendo textos sostenidos, menús, opciones de navegación, entre otros, mejorando significativamente la interacción y la comprensión de la información presentada en la plataforma.

Asimismo, la implementación de este componente de asistencia técnica tiene como finalidad primordial promover la igualdad de acceso y el aprendizaje inclusivo para personas con discapacidad visual, permitiéndoles aprovechar plenamente los recursos educativos y las herramientas interactivas ofrecidas por la plataforma Creavi. Esta iniciativa representa un paso significativo hacia la creación de un entorno educativo digital verdaderamente inclusivo y accesible, alineado con los principios de equidad y diversidad en el ámbito de la educación.

Acrónimos

1. **WCAG:** Web Content Accessibility Guidelines (pautas de accesibilidad para el contenido web). Estas son pautas desarrolladas por la W3C para garantizar la accesibilidad de los sitios web, especialmente para personas con discapacidades.
2. **GDPR:** General Data Protection Regulation (Reglamento General de Protección de Datos). Es una regulación de la Unión Europea que se ocupa de la protección de datos personales y la privacidad de los individuos.
3. **HIPAA:** Health Insurance Portability and Accountability Act (Ley de Portabilidad y Responsabilidad del Seguro de Salud). Es una ley de los Estados Unidos que establece estándares para la protección y privacidad de la información de salud.
4. **CPU:** Central Processing Unit (Unidad Central de Procesamiento). Es la parte de una computadora que realiza las instrucciones de un programa.
5. **ID:** Identificación. En el contexto del proyecto, se utiliza para referirse a un identificador único asociado a cada caso de uso.

1. Descripción General

Objetivos del Sistema

- **Facilitar el acceso a la información:** El objetivo principal es garantizar que las personas con discapacidad visual puedan acceder a la información de manera efectiva, independientemente de su discapacidad.
- **Facilitar la navegación:** Garantizar que la página web sea fácil de navegar para las personas con discapacidad visual, con un menú de navegación claro y enlaces internos bien etiquetados.

Funcionalidad General

El componente está diseñado para ayudar a personas con discapacidad visual que proporcionará una voz computarizada o función de lectura de pantalla en voz alta del contenido textual de la página. Esta funcionalidad es esencial para permitir que las personas con discapacidad visual puedan acceder y comprender el contenido web de manera efectiva.

1. **Comandos de voz:** Leer comandos por voz.
2. **Lectura de la pantalla:** La capacidad de leer en voz alta de la pantalla y hacer una descripción de lo que se muestra en pantalla.
3. **Navegación de la página:** La capacidad de navegar por la página web de manera estructurada, permitiendo a los usuarios escuchar una descripción de los elementos y enlaces presentes en la página.
4. **Soporte para múltiples idiomas:** Permitir que los usuarios seleccionen el idioma en el que desean que se les lea el contenido.
5. **Realizar búsquedas por voz:** La capacidad de buscar contenido en la plataforma de manera eficiente por medio de los comandos de voz.
6. **Realizar resumen de la pantalla:** El componente tendrá la capacidad hacer un resumen de la pantalla del usuario.
7. **Resaltar fragmentos por voz:** Mediante comandos se podrán resaltar fragmentos de textos deseados de la pantalla.
8. **Tomar notas por voz:** La capacidad de tomar notas mediante los comandos de voz

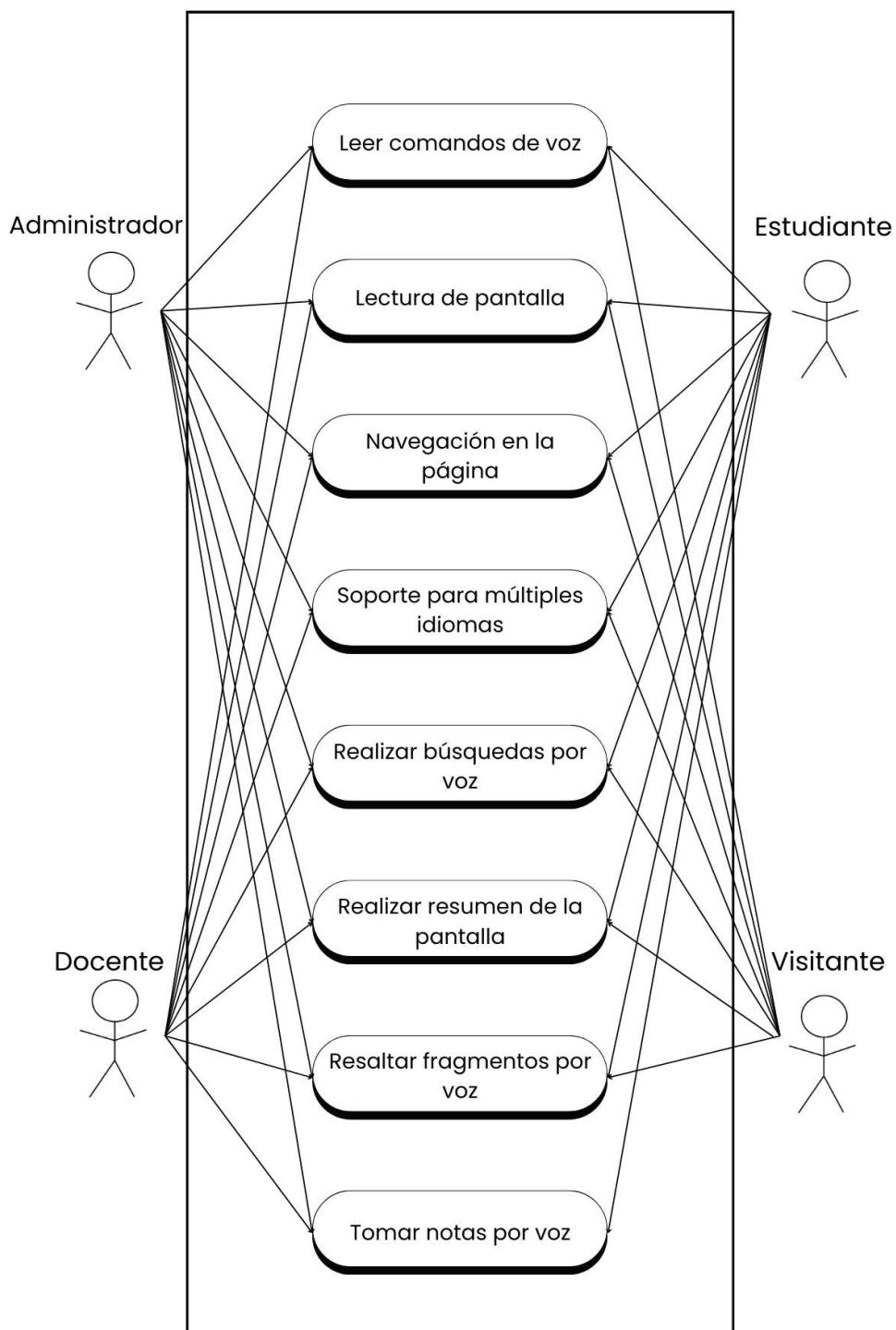
Usuarios del Sistema

Los siguientes usuarios pueden interactuar con la pizarra dependiendo de las funcionalidades.

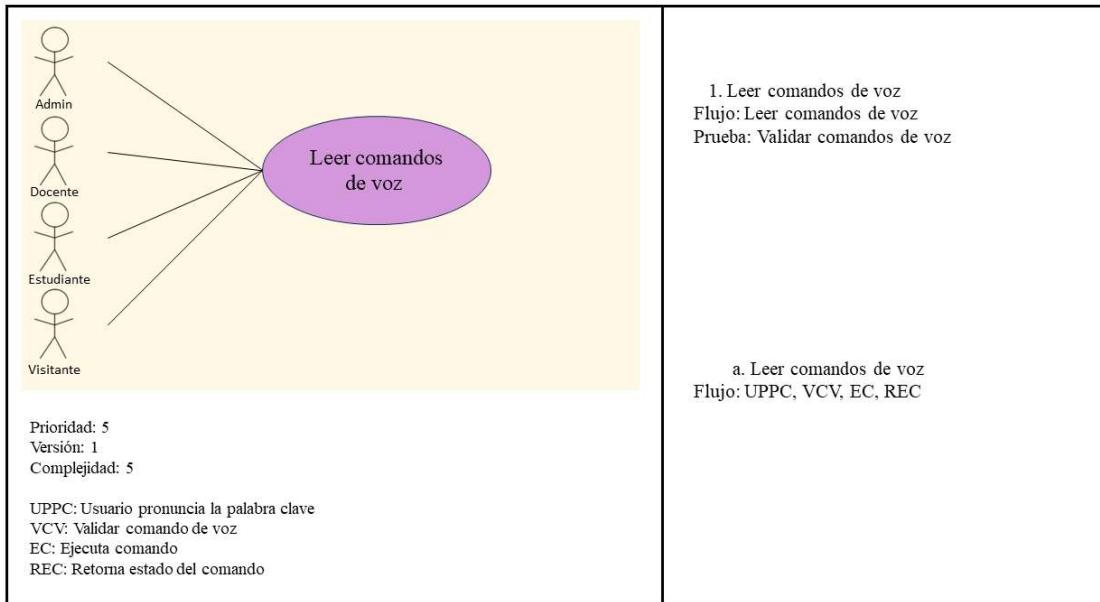
| Funcionalidad | Administradores | Docentes | Estudiantes | Visitante |
|-------------------------------------------|-----------------|----------|-------------|-----------|
| Ler comandos de voz | | | | |
| Lectura de la pantalla | | | | |
| Navegación en la Página | | | | |
| Soporte para múltiples idiomas | | | | |
| Realizar búsquedas por voz | | | | |
| Realizar un resumen de la pantalla | | | | |
| Resaltar fragmentos por voz | | | | |
| Tomar notas por voz | | | | |

Casos de Uso

Diagramas de casos de usos

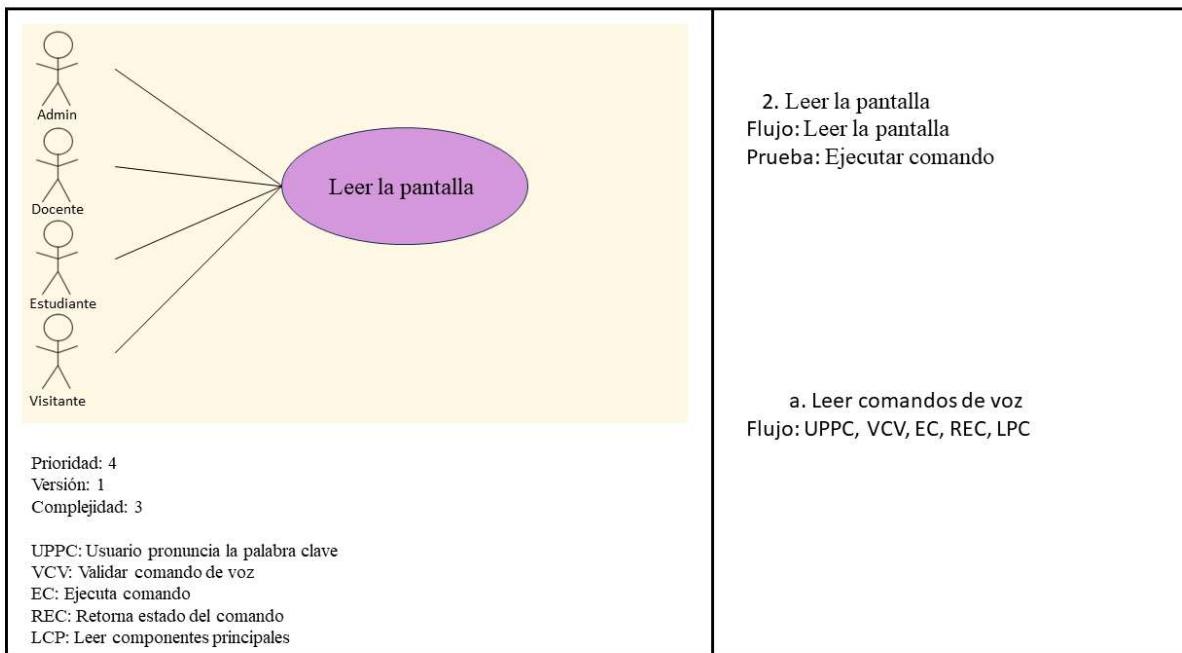


Diagramas de Flujo de Casos de Uso



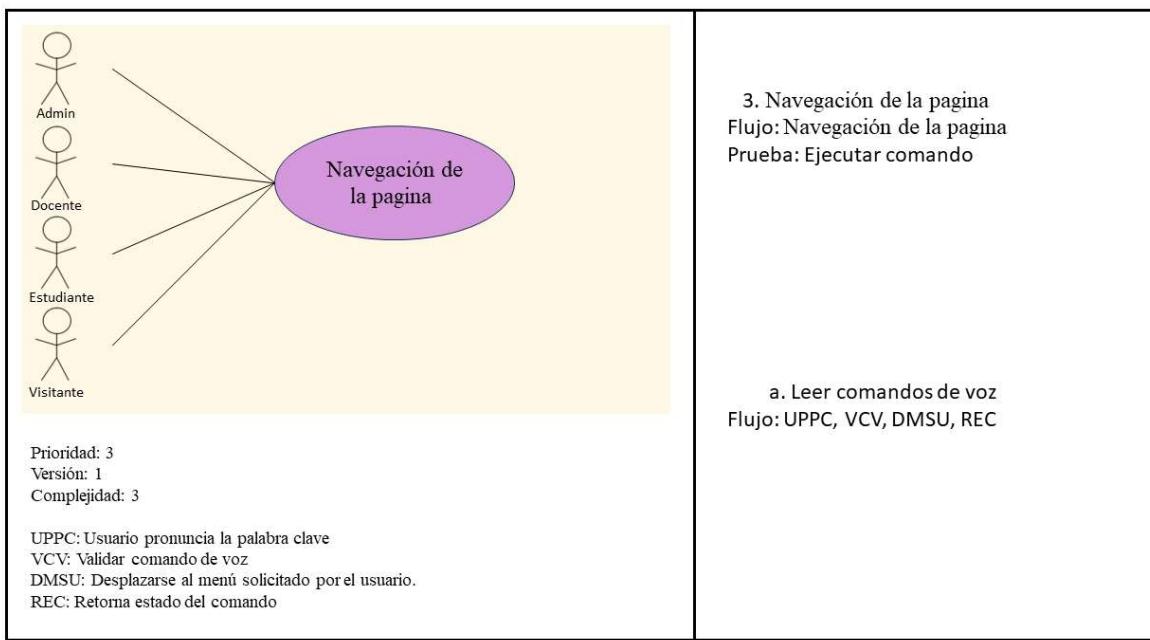
CASO No. 1 Leer Comandos de Voz

| | | |
|---------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| ID: | CU-1 | |
| Nombre | Leer comandos de voz | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe leer comandos de voz | |
| Urgencia | 5 | |
| Esfuerzo | 5 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario (Administrador, docente, estudiante y visitante) | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz de la acción a realizar | |
| | | Validar comando de voz |
| | | Ejecuta comando |
| | | Retorna estado del comando |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "Julia" y el comando de voz de la acción a realizar | |
| | | Validar comando de voz |
| | | No reconoce comando de voz |
| | | Retornar estado del comando |
| | | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" |



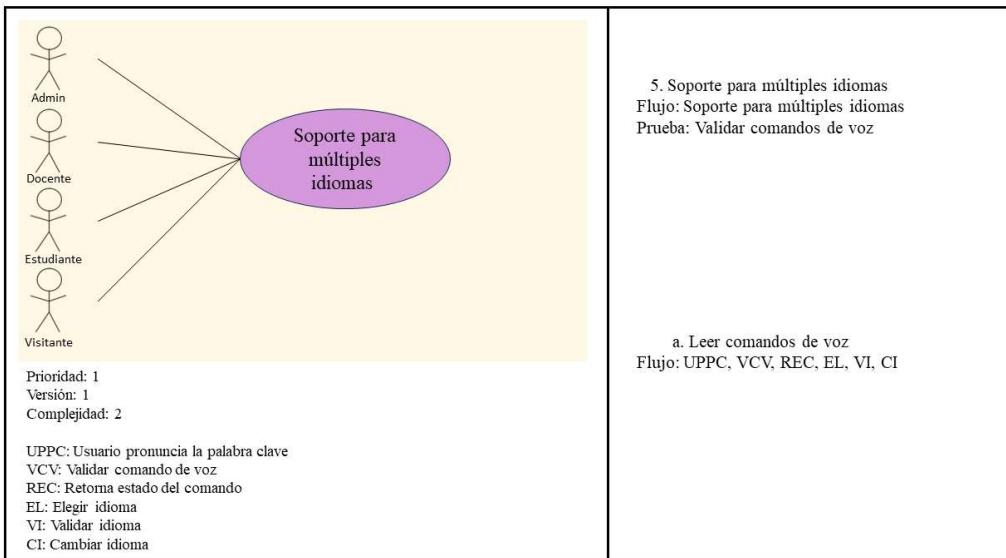
CASO No. 2 Lectura de Pantalla

| | | |
|---------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| ID: | CU-2 | |
| Nombre | Leer la pantalla | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe tener la capacidad de leer la pantalla actual | |
| Urgencia | 4 | |
| Esfuerzo | 3 | |
| Precondiciones | El contenido debe ser accesible. | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz de la "Leer pantalla" | |
| | | Validar comando de voz |
| | | Ejecuta comando |
| | | Retornar estado del comando |
| | | Leer los componentes principales, títulos, menús y si hay recursos multimedia describirlos. |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "Julia" y el comando de voz de la acción a realizar | |
| | | Validar comando de voz |
| | | Responder "No he podido comprender lo que quieras hacer, inténtalo nuevamente" |



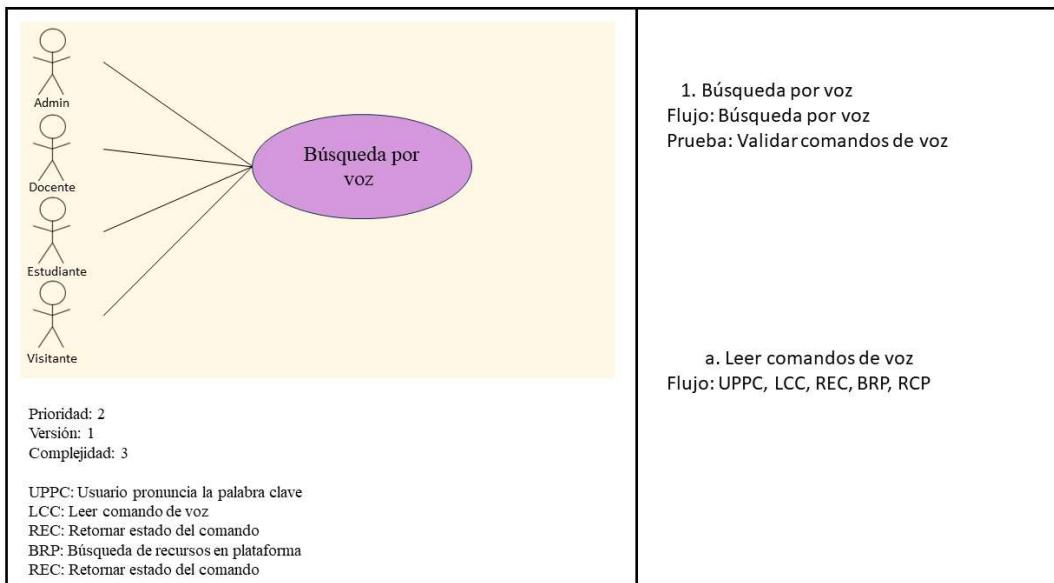
CASO No. 3 Navegación de la Página

| | | |
|---------------------|-------------------------------------------------------------------------------------------------------|---------|
| ID: | CU-3 | |
| Nombre | Navegación de la pagina | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe permitir al usuario la navegación en la pagina | |
| Urgencia | 3 | |
| Esfuerzo | 3 | |
| Precondiciones | Usuario debe conocer la estructura de la página | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz "Llévame a" y el menú u opción al que desea ir | |
| | Validar comando de voz | |
| | Desplazarse al menú u opción solicitada por el usuario. | |
| | Decir "Ahora estás en 'menú u opción descrita por el usuario" | |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "Julia" y el comando de voz "Llévame a" y el menú u opción al que desea ir | |
| | Validar comando de voz | |
| | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" | |



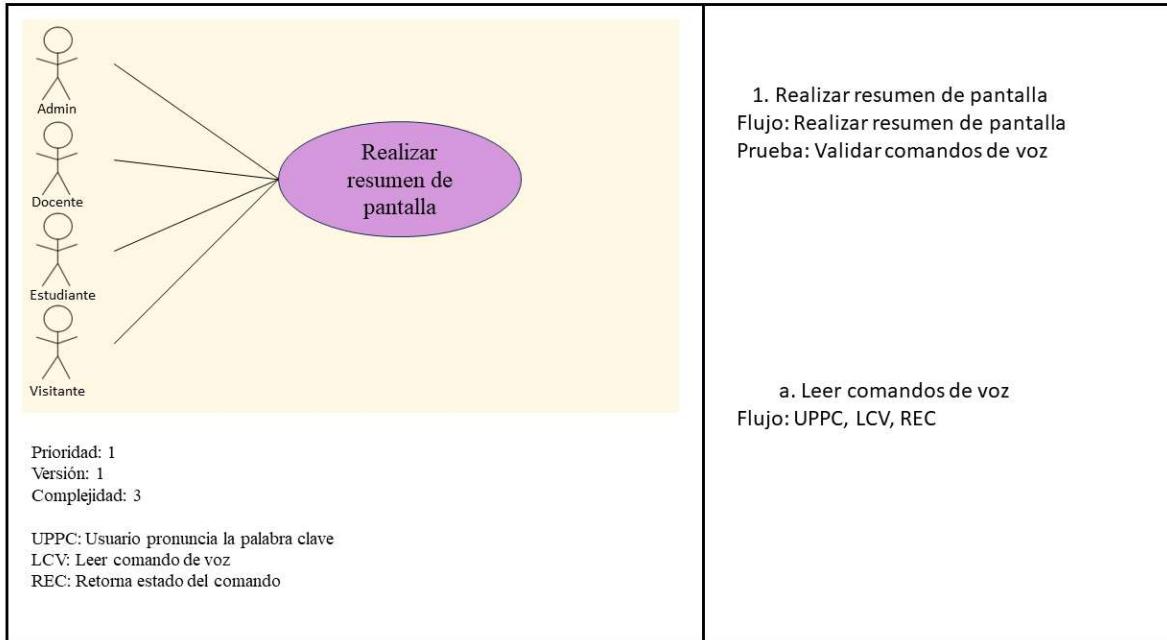
CASO No. 4 Soporte para Múltiples Idiomas

| | | |
|----------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| ID: | CU-5 | |
| Nombre | Soporte para múltiples idiomas | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe tener la capacidad de traducir a múltiples idiomas | |
| Urgencia | 1 | |
| Esfuerzo | 2 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz de cambio de idioma | |
| | | Validar comando de voz |
| | | Retornar estado del comando |
| | | Decir "¿Que idioma prefieres?" |
| | Decir el idioma de preferencia | |
| | | Validar idioma |
| | | Cambiar idioma |
| | | Decir en el idioma solicitado "Está bien, te hablaré en este idioma" |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "Julia" y el comando de voz de la acción a realizar | |
| | | Leer comando de voz |
| | | Retornar estado del comando |
| | | Responder "No he podido comprender lo que quieras hacer, inténtalo nuevamente" |



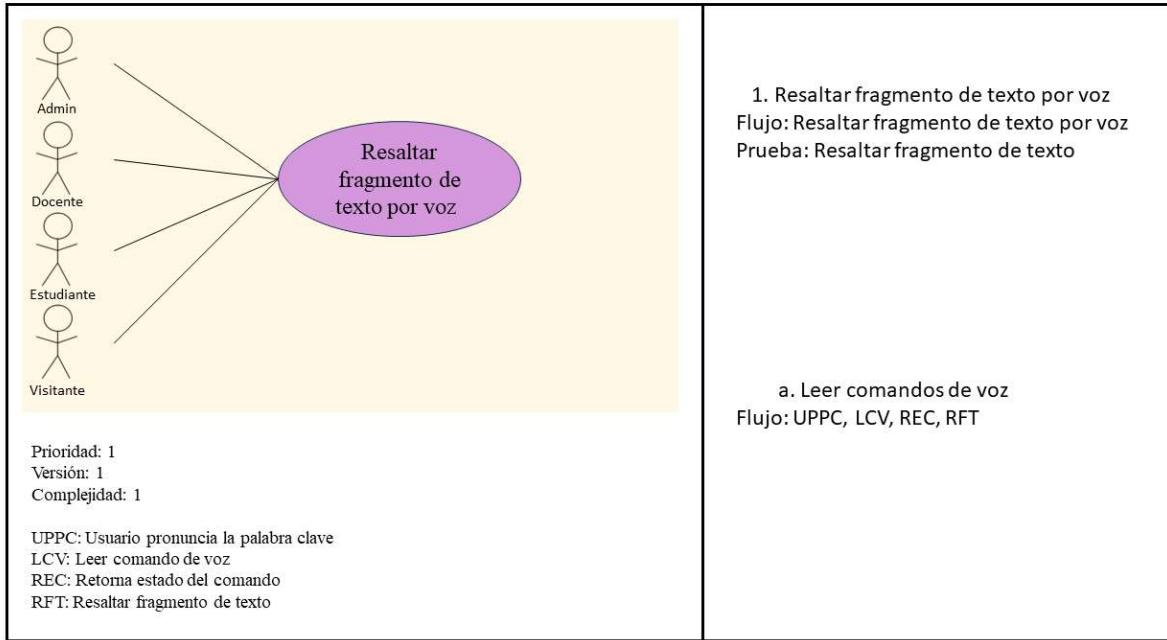
CASO No. 5 Búsqueda por Voz

| | | |
|---------------------|---------------------------------------------------------------------------------------------|---------|
| ID: | CU-6 | |
| Nombre | Búsqueda por voz | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe tener la capacidad de buscar contenido en la plataforma por medio de voz | |
| Urgencia | 2 | |
| Esfuerzo | 3 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia", el comando de voz "busca" y el recurso que desea buscar | |
| | Leer comando de voz | |
| | Retornar estado del comando | |
| | Hacer la búsqueda en la plataforma de los recursos que concuerden con la descripción | |
| | Retornar estado del comando | |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "por definir" y el comando de voz de la acción a realizar | |
| | Leer comando de voz | |
| | Retornar estado del comando | |
| | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" | |



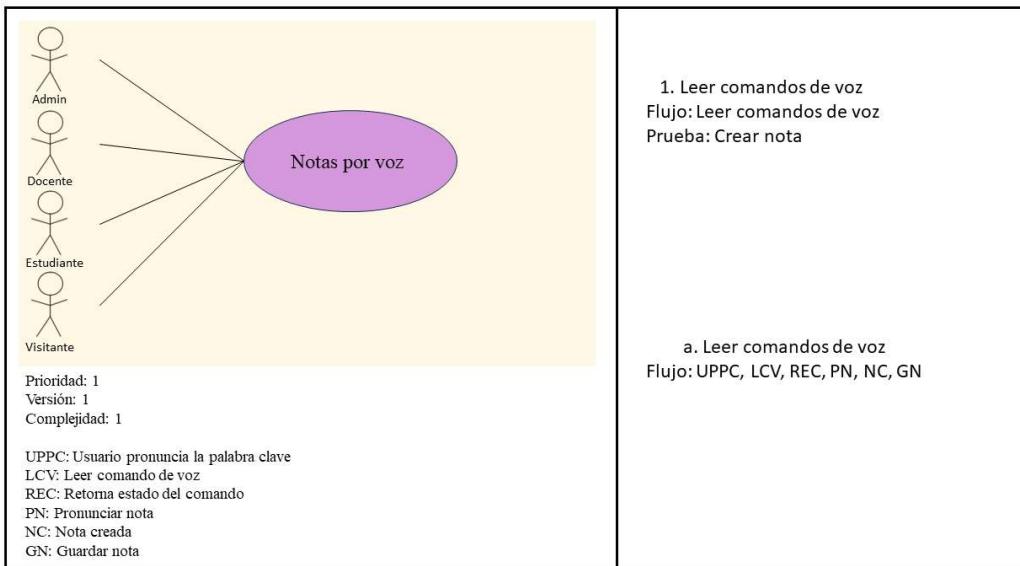
CASO No. 6 Realizar Resumen de Pantalla

| | | |
|---------------------|--------------------------------------------------------------------------------------|---------|
| ID: | CU-7 | |
| Nombre | Realizar resumen de pantalla | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso debe tener la capacidad de realizar un resumen de la pantalla | |
| Urgencia | 1 | |
| Esfuerzo | 3 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz de resumen de pantalla | |
| | Leer comando de voz | |
| | Retornar estado del comando | |
| | Decir "En tu pantalla hay" | |
| | Hacer una descripción breve de los elementos más importantes | |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "por definir" y el comando de voz de la acción a realizar | |
| | Leer comando de voz | |
| | Retornar estado del comando | |
| | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" | |



CASO No. 7 Resaltar fragmento de texto por voz

| | | |
|----------------------------|------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| ID: | CU-8 | |
| Nombre | Resaltar fragmento de texto por voz | |
| Actores | Administrador, docente, estudiante y visitante | |
| Objetivo | Este caso deberá resaltar un fragmento de texto mediante comandos de voz | |
| Urgencia | 1 | |
| Esfuerzo | 1 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia", el comando de voz "resalta el texto" y pronunciar el fragmento de texto a resaltar | |
| | | Leer comando de voz |
| | | Retornar estado del comando |
| | | Resaltar el fragmento de texto pronunciado por el usuario. |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "por definir" y el comando de voz de la acción a realizar | |
| | | Leer comando de voz |
| | | Retornar estado del comando |
| | | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" |



CASO No. 8 Notas por voz

| | | |
|----------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| ID: | CU-9 | |
| Nombre | Notas por voz | |
| Actores | Administrador, docente, estudiante | |
| Objetivo | Este caso debe tener la capacidad de tomar notas por medio de voz y guardarlas | |
| Urgencia | 1 | |
| Esfuerzo | 1 | |
| Precondiciones | Ninguna | |
| Flujo normal | Usuario | Sistema |
| | Pronuncia la palabra clave "Julia" y el comando de voz "Crea una nueva nota" | |
| | | Leer comando de voz |
| | | Retornar estado del comando |
| | | Decir "¿Que deseas que diga la nota?" |
| | Pronunciar la nota | |
| | | Decir "Tu nota ha sido creada" |
| | | Guardar la nota en esa pantalla en el usuario |
| Flujo alternativo 1 | | |
| | Pronuncia la palabra clave "por definir" y el comando de voz de la acción a realizar | |
| | | Leer comando de voz |
| | | Retornar estado del comando |
| | | Responder "No he podido comprender lo que quieres hacer, inténtalo nuevamente" |

Prioridad de requerimientos

| Urgencia | | 1- Baja | 2- Menor | 3- Moderada | 4- Alta | 5- Obligatoria |
|----------|-------------|---------|----------|-------------|---------|----------------|
| Impacto | 5- Muy alto | 5 | 10 | 15 | 20 | 25 |
| | | | | | | CU-1 |
| | 4- Alto | 4 | 8 | 12 | 16 | 20 |
| | | | | | | |
| | 3- Medio | 3 | 6 | 9 | 12 | 15 |
| | | CU-4 | CU-6 | CU-3 | CU-2 | |
| | 2- Bajo | 2 | 4 | 6 | 8 | 10 |
| | | | | | | |
| | 1- Muy bajo | 1 | 2 | 3 | 4 | 5 |
| | | CU-8 | CU-5 | CU-7 | | |

Requisitos NO funcionales.

Requisitos de Desempeño:

1. **Tiempo de Respuesta:** El sistema debe proporcionar retroalimentación a los comandos de voz del usuario en un tiempo máximo de 1 segundo, garantizando una experiencia fluida.
2. **Capacidad de Procesamiento:** El sistema debe manejar múltiples solicitudes de voz a la vez sin degradación del rendimiento, soportando al menos 50 usuarios activos en un momento dado.
3. **Consumo de Recursos:** El componente "TalkBack" debe ser eficiente en el uso de recursos del servidor, minimizando el consumo de CPU y memoria para mantener un rendimiento óptimo.

Requisitos de Seguridad:

4. **Acceso Seguro:** Se debe implementar una autenticación segura para los usuarios, incluyendo medidas como autenticación de dos factores, para proteger la información confidencial.
5. **Privacidad de Datos:** Los datos de los usuarios, incluyendo comandos de voz, deben cifrarse y almacenarse seguramente, cumpliendo con las regulaciones de privacidad vigentes, como GDPR o HIPAA.
6. **Protección contra Ataques:** El sistema debe contar con medidas de seguridad, como firewalls y detección de intrusiones, para prevenir ataques maliciosos y garantizar la integridad de los datos.

Requisitos de Usabilidad:

1. **Accesibilidad:** El componente "TalkBack" debe cumplir con estándares de accesibilidad web (como WCAG) para garantizar que las personas con discapacidad visual puedan utilizarlo de manera efectiva.
2. **Interfaz Intuitiva:** La interfaz de voz debe ser intuitiva y fácil de entender para los usuarios, con instrucciones claras y retroalimentación audible.
3. **Personalización:** Los usuarios deben poder personalizar los comandos de voz y la velocidad de lectura para adaptar la experiencia a sus preferencias individuales.

Requisitos de Escalabilidad:

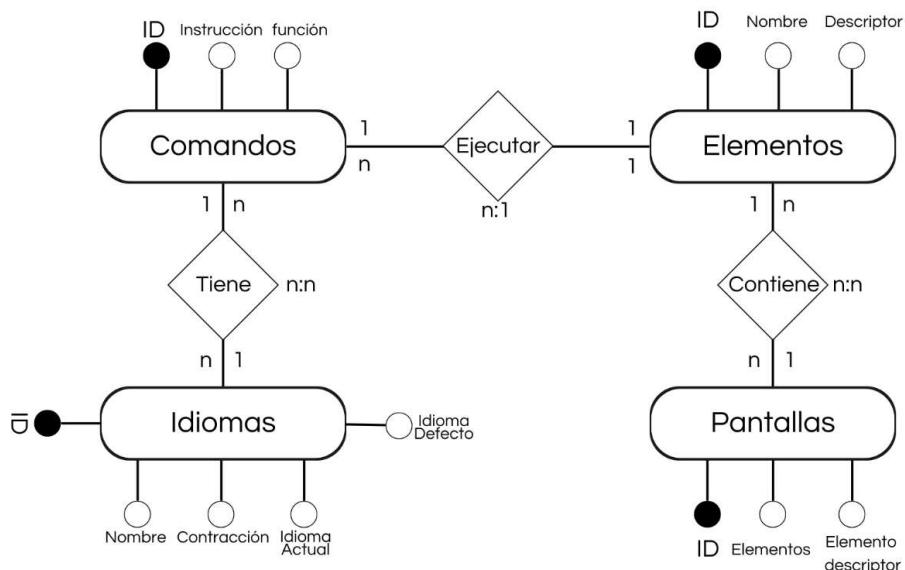
1. **Adaptabilidad a Crecimiento:** El sistema debe ser escalable, de modo que pueda gestionar un aumento de usuarios y contenido sin una degradación significativa del rendimiento.
2. **Mantenimiento Sencillo:** La arquitectura del software debe ser modular y extensible para facilitar futuras actualizaciones y mejoras, sin interrupciones importantes en el servicio.
3. **Soporte Multiplataforma:** El componente "TalkBack" debe ser compatible con una variedad de dispositivos y sistemas operativos, permitiendo a los usuarios acceder desde diferentes plataformas.

5. Modelado E/R

Diagrama de entidad relación

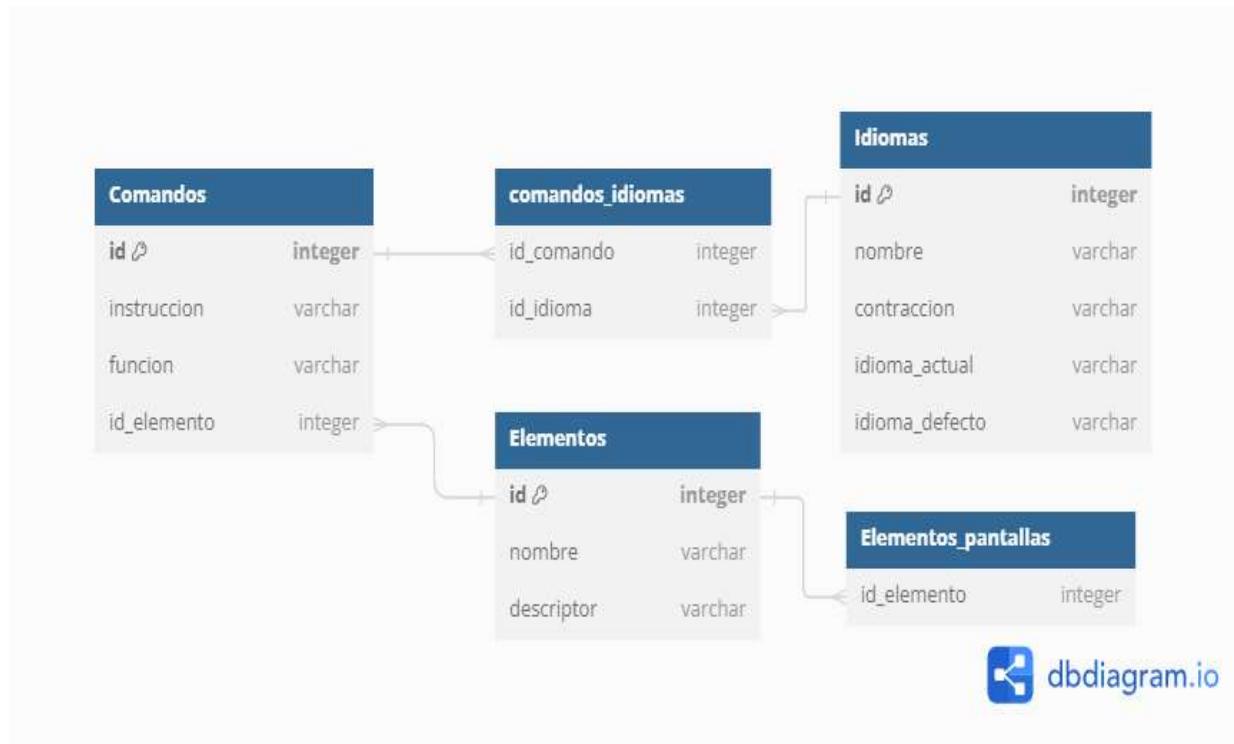
A partir de la extracción de los datos pertenecientes a las funcionalidades descritas, se establece el siguiente modelado de datos.

| | | |
|-----------------------------------------------|-------------------------|--------------------------------------------|
| CU – 1 Leer comandos de voz | Comando activador | Julia |
| | Comando | [“Leer”, “Abrir documento”, “Buscar”, ...] |
| CU – 2 Lectura de pantalla | Comando activador | Julia |
| | Información de pantalla | [“Imagen”, “Titulo”, “tabla” ...] |
| CU- 3 Navegación en la página | Comando activador | Julia |
| | Información de página | [“Página 22”] |
| CU- 4 Soporte para múltiples idiomas | Comando activador | Julia |
| | Idiomas | [“ES”, “US”, “FR”] |
| | Idioma actual | [“ES”] |
| CU- 5 Realizar Búsqueda por voz | Comando activador | Julia |
| | Búsqueda por voz | QUERI |
| | Consulta | “string” |
| CU – 6 Realizar resumen de la pantalla | Comando activador | Julia |
| | Resúmenes | [“Resúmenes”] |
| CU – 7 Resaltar fragmentos por voz | Comando activador | Julia |
| | Resaltable | False |
| CU- 8 Tomar notas Por voz | Comando activador | Julia |
| | Tomar nota | [“el ciclo del agua”] |



Herramientas para el Diseño de Software: dbdiagram.io

Diagrama relacional



Etapa 2: Persistencia de Datos con Backend

1. Introducción

- Propósito de la Etapa
- Alcance de la Etapa
- Definiciones y Acrónimos

2. Diseño de la Arquitectura de Backend

- Descripción de la Arquitectura Propuesta
- Componentes del Backend
- Diagramas de Arquitectura

3. Elección de la Base de Datos

- Evaluación de Opciones (SQL o NoSQL)
- Justificación de la Elección
- Diseño de Esquema de Base de Datos

4. Implementación del Backend

- Elección del Lenguaje de Programación
- Creación de la Lógica de Negocio
- Desarrollo de Endpoints y APIs
- Autenticación y Autorización

5. Conexión a la Base de Datos

- Configuración de la Conexión
- Desarrollo de Operaciones CRUD
- Manejo de Transacciones

6. Pruebas del Backend

- Diseño de Casos de Prueba
- Ejecución de Pruebas Unitarias y de Integración
- Manejo de Errores y Excepciones

Introducción

Propósito de la etapa

En esta etapa se lleva a cabo el proceso de persistencia de datos del proyecto Talkback Julia, donde se crea el diseño del backend de los datos, conexión con bases de datos y demás interacciones que permitirán el funcionamiento correcto del software, para proyectarlo hacia la realización del frontend el semestre siguiente.

Alcance de la etapa

El alcance de esta etapa del proyecto Talkback Julia se centra en establecer una base sólida para el manejo y almacenamiento de datos, esencial para el correcto funcionamiento del software. Esto incluye el diseño y desarrollo del backend, la implementación de la arquitectura de la base de datos, y la creación de conexiones seguras y eficientes entre el backend y la base de datos. Además, se contemplan todas las interacciones necesarias para garantizar la integridad y accesibilidad de los datos. Este trabajo es fundamental para asegurar que, en el próximo semestre, el desarrollo del frontend pueda llevarse a cabo de manera fluida y efectiva, basándose en un sistema de backend robusto y bien estructurado.

Definiciones y acrónimos

CRUD: Acrónimo de Create, Read, Update y Delete. Este concepto se utiliza para describir las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información.

Node.js: Node (o más correctamente: Node. Js) es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript.

Mongo atlas:

Mogno atlas: Es una [base de datos en la nube completamente administrada](#) que maneja toda la complejidad de implementar, administrar y reparar todas tus implementaciones en el proveedor de servicios en la nube de tu elección.

Middleware: El término middleware se refiere al sistema de software que ofrece funciones y servicios de nube comunes para las aplicaciones.

Controller: componentes que procesan solicitudes ODBC y devuelven datos a la aplicación. Si es necesario, los controladores modifican la solicitud de una aplicación en un formulario comprendido por el origen de datos

Service: sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable.

Repository: lugar de almacenamiento del cual pueden ser recuperados e instalados los paquetes de software en un ordenador.

Interceptor: objeto que permite instrumentar la aplicación para que capture datos de interés.

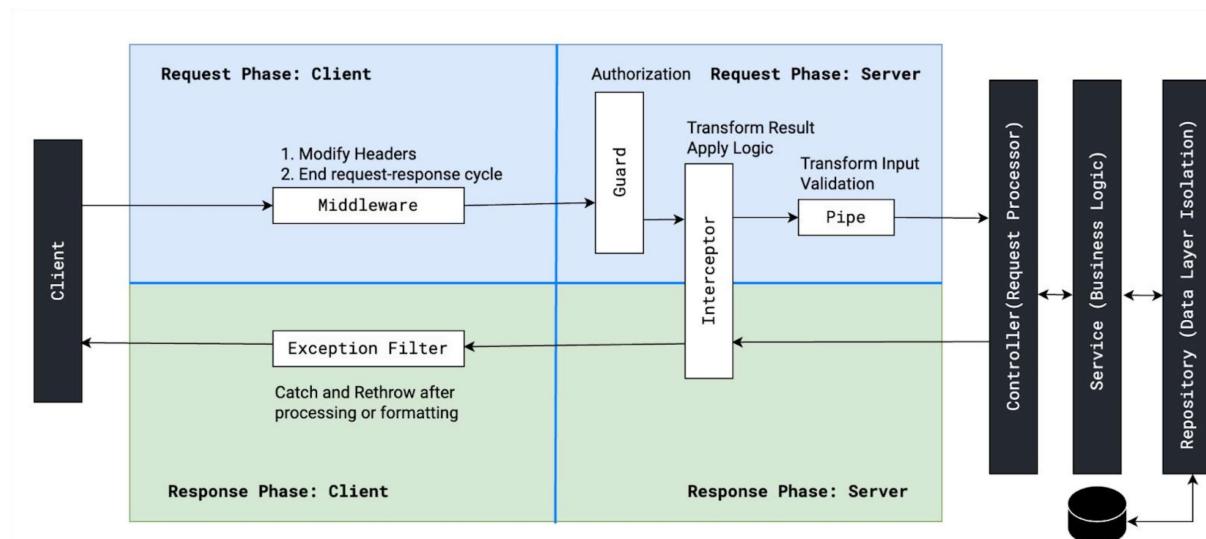
Server: sistema que proporciona recursos, datos, servicios o programas a otros ordenadores, conocidos como clientes, a través de una red.

Diseño de la arquitectura del Backend

Descripción de la arquitectura propuesta

Este proyecto está basado en un sistema de Nest.js que cuenta con múltiples elementos que incluye: el cliente, el interceptor que proporciona una validación al controlador y existe una interacción entre el controlador, el servidor y el repositorio, al igual que con la base de datos.

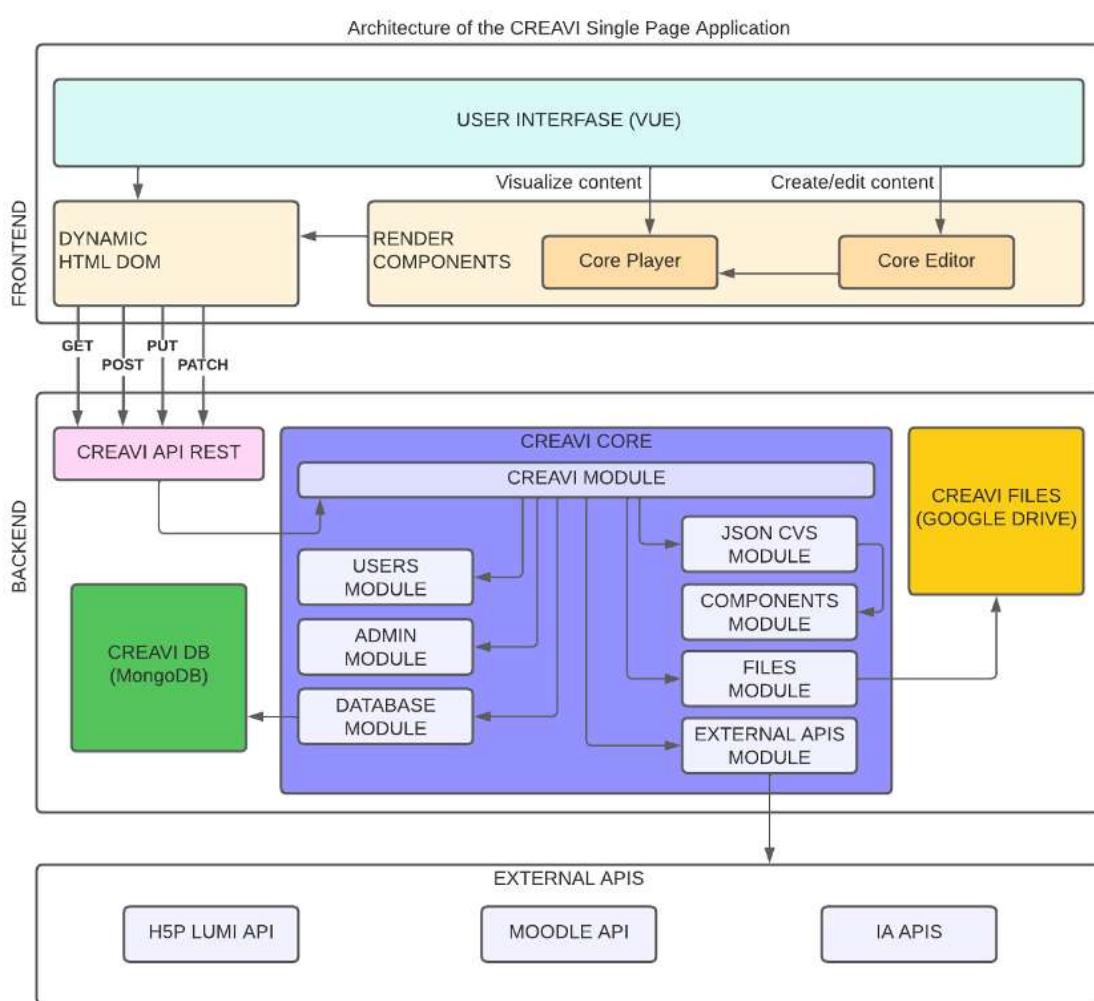
Componentes del backend



En esta etapa del desarrollo backend, avanzamos desde el *Service (Business Logic)* capa donde se maneja la lógica de negocio, donde se procesan y transforman los datos que se reciben del

repositorio antes de enviarlos al controlador. También se pueden gestionar DTOs en esta capa para asegurar que los datos sean transferidos de manera correcta entre el cliente y el servidor. Hasta la *Repository (Data Layer Isolation)* donde se maneja la conexión a la base de datos, las consultas, las operaciones CRUD (Create, Read, Update, Delete) y el uso de esquemas para estructurar los datos. En esta capa se encapsula toda la lógica relacionada con el acceso y la persistencia de datos.

Diagrama de arquitectura



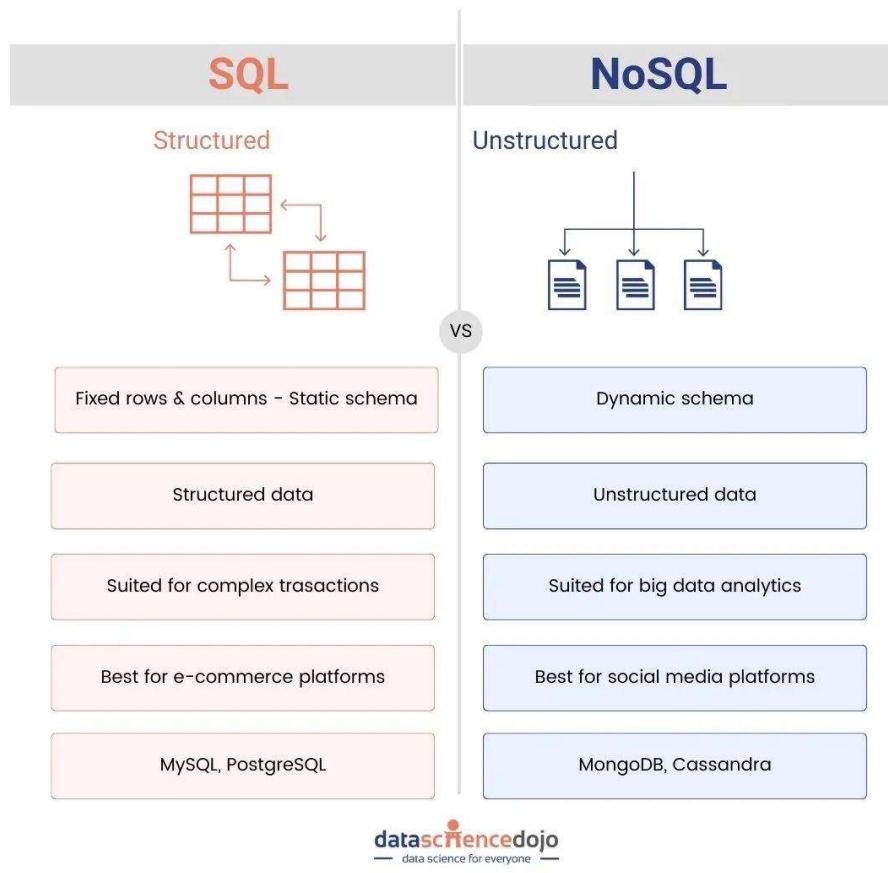
El diagrama anterior muestra la arquitectura de la aplicación de una sola página (SPA) de CREAVI, compuesta por una interfaz de usuario que incluye componentes para la visualización y edición de contenido, un backend que maneja la lógica de la aplicación mediante módulos, y una base de datos MongoDB. En esta arquitectura, se integrará el módulo "Julia", un asistente virtual que facilitará la navegación e interacción de usuarios con limitaciones visuales mediante

comandos de voz. Julia utilizará APIs de reconocimiento de voz para captar comandos, procesarlos y proporcionar retroalimentación auditiva. En el frontend, se incorporarán componentes para la captura de voz y retroalimentación en los módulos de visualización y edición. En el backend, Julia se añadirá como un nuevo módulo dentro del núcleo CREAVID, con endpoints en la API REST para manejar solicitudes de comandos de voz, lógica para interpretar y ejecutar estos comandos, y generación de respuestas auditivas. Esto mejorará significativamente la accesibilidad y usabilidad de la plataforma, permitiendo a los usuarios con limitaciones visuales crear, editar, y navegar contenido mediante comandos de voz.

Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Una de las decisiones más importantes para el desarrollo del backend del proyecto, es la base de datos que se usarán para el software. Entre los tipos de bases de datos más populares están las bases de datos SQL y NoSQL, la elección de una u otra, se basa en las necesidades que presente el software. En la siguiente tabla se muestran las principales diferencias entre SQL y NoSQL.

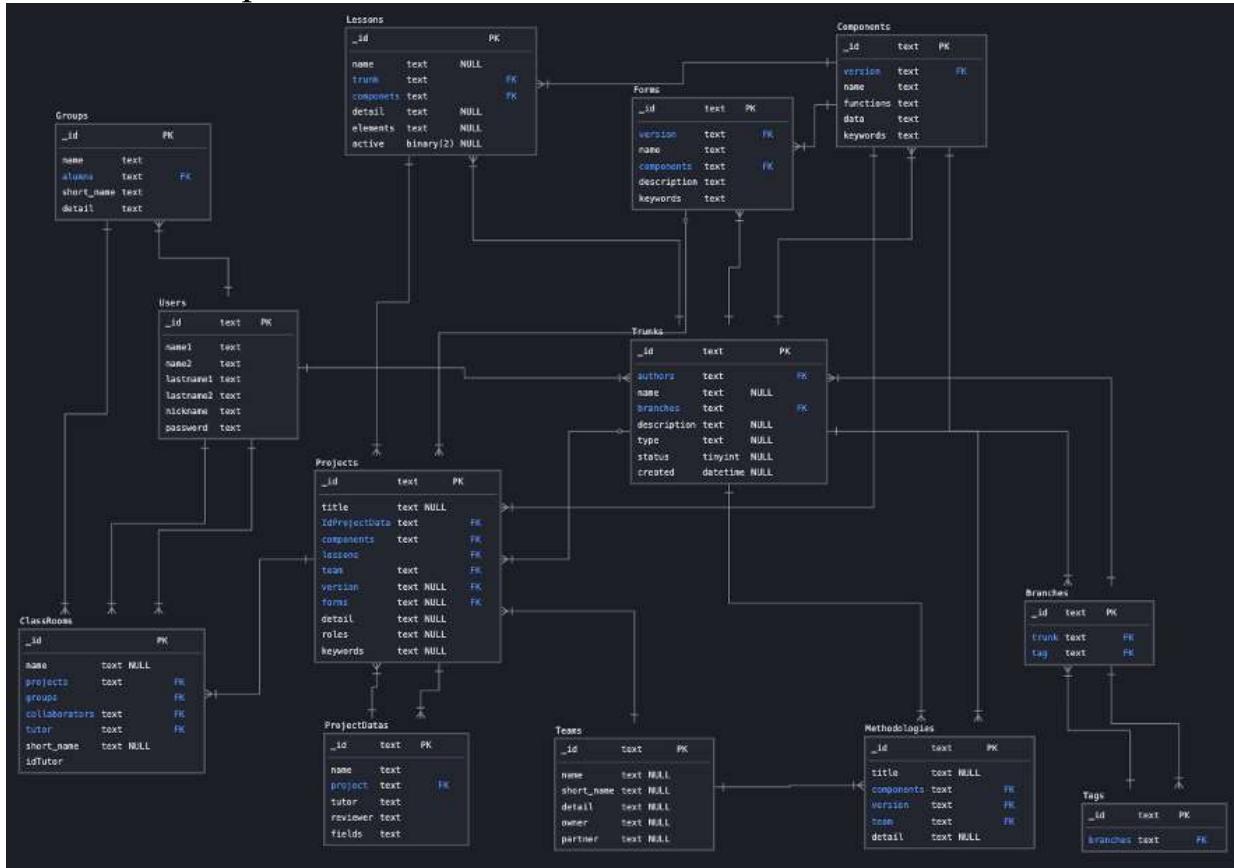


Justificación de la Elección

Para el presente software se usará una base de datos SQL, puesto que se hace necesaria una estructura organizada, que permita la recuperación de datos y ejecución de consultas precisas al momento de realizar comandos por voz al software que proporcionan una respuesta al usuario con discapacidad visual total o parcial que responda a los requerimientos del mismo SQL

NoSQL

Diseño de Esquema de Base de Datos



En el componente Julia, se han definido tres colecciones esenciales: Comandos, Elementos y Lenguajes. La colección Comandos almacena los diferentes comandos de voz que el asistente virtual puede reconocer y ejecutar, cada uno vinculado a una acción específica dentro de la plataforma. La colección Elementos contiene las diversas entidades y componentes de la aplicación que pueden ser manipulados o accedidos mediante comandos de voz, como proyectos, usuarios o archivos. La colección Lenguajes gestiona los diferentes idiomas soportados por Julia, permitiendo que el asistente ofrezca respuestas y reconozca comandos en múltiples idiomas, mejorando la accesibilidad. Estas colecciones están interrelacionadas, donde los Comandos hacen referencia a los Elementos a los que se aplican y a los Lenguajes en los que pueden ser interpretados. Esto garantiza que Julia pueda proporcionar una experiencia de usuario cohesiva y accesible, interactuando con los diversos módulos y funcionalidades del proyecto CREAVI de manera eficiente y multilingüe.

Implementación del Backend

Elección del Lenguaje de Programación

Utilizar Node.js, Nest.js y Express.js en este software tiene varias ventajas clave. Por un lado, Node.js ofrece eficiencia y manejo asíncrono crucial para aplicaciones en tiempo real. Asimismo, Express.js facilita la creación de APIs robustas y seguras, ideal para gestionar solicitudes y respuestas del sistema. Y, Nest.js proporciona una arquitectura modular y escalable, integrando TypeScript para mejorar la mantenibilidad del código. Por tanto, la combinación de una base de datos SQL y MongoDB permite flexibilidad en el almacenamiento y optimización del rendimiento. Esta tecnología conjunta asegura un desarrollo eficiente y accesible para usuarios con necesidades especiales.

Conexión a la Base de Datos

Configuración de la Conexión

La conexión a la base de datos MongoDB Atlas desde el componente Julia se da mediante las credenciales de acceso, la URL de conexión y una configuración específica relacionada con la seguridad del sistema.

Al estar trabajando con Mongo, deben estar instaladas las dependencias de Mongo: `npm install mongodb`.



```
1 imports: [
2   ConfigModule.forRoot({ envFilePath: '.env', isGlobal: true }),
3   ServeStaticModule.forRoot({ rootPath: join(__dirname, '..', 'client') }),
4   MongooseModule.forRoot(process.env.DB_URI),
5   CommandsModule,
6   ElementsModule,
7   LanguagesModule,
8 ],
```

Luego, se realizó la configuración del import en el App Module. Se configuró el módulo importando un archivo `.env`, junto con el `ServeStaticModule` con un `rootPath` utilizando `__dirname` y cliente. Por último, se configuró el `MongooseModule` para que obtenga el enlace a la base de datos desde la variable de entorno `DB_URI`.

```
1 DB_URI = "mongodb+srv://<username>:<password>@<cluster-url>/test?retryWrites=true&w=majority";
```

Por último, en un archivo .env se crea la variable de entorno DB_URI y se le asigna el valor del enlace a la base de datos

Desarrollo de Operaciones CRUD

Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) son fundamentales para el funcionamiento óptimo del componente y para interactuar con las colecciones en MongoDB. Aquí se detalla cómo se implementan estas operaciones para las colecciones de Comando, Elementos y Lenguajes

Commands

```
1 @Injectable()
2 export class CommandsService {
3   constructor(@InjectModel(Commands.name) private CommandsModel: Model<Commands>) {}
4
5   async create(commandsDto: CreateCommandDto): Promise<Commands> {
6     const Commands = new this.CommandsModel(commandsDto);
7     return Commands.save();
8   }
9
10  async findAll(): Promise<Commands[]> {
11    return this.CommandsModel.find().exec();
12  }
13
14  async findOne(id: string): Promise<Commands> {
15    return this.CommandsModel.findById(id).exec();
16  }
17
18  async update(id: string, updateCommandDto: UpdateCommandDto) {
19    const updateCommand = await this.CommandsModel.findByIdAndUpdate(id, updateCommandDto)
20    return updateCommand
21  }
22
23  async remove(id: string) {
24    const deletecommand = await this.CommandsModel.findByIdAndDelete(id)
25    return deletecommand
26  }
27 }
```

Elements

```
● ● ●  
1  @Injectable()  
2  export class ElementsService {  
3    constructor(@InjectModel(Elements.name) private ElementsModel: Model<Elements>) {}  
4  
5    async create(createElementDto: CreateElementDto): Promise<Elements> {  
6      const Elements = new this.ElementsModel(createElementDto);  
7      return Elements.save();  
8    }  
9  
10   async findAll(): Promise<Elements[]> {  
11     return this.ElementsModel.find().exec();  
12   }  
13  
14   async findOne(id: string): Promise<Elements> {  
15     return this.ElementsModel.findById(id).exec();  
16   }  
17  
18   async update(id: string, updateElementDto: UpdateElementDto): Promise<Elements> {  
19     return this.ElementsModel.findByIdAndUpdate(id, updateElementDto).exec();  
20   }  
21  
22   async remove(id: string): Promise<Elements> {  
23     return this.ElementsModel.findByIdAndDelete(id).exec();  
24   }  
25 }
```

Languages

```
● ● ●  
1  @Injectable()  
2  export class LanguagesService {  
3    constructor(@InjectModel(Languages.name) private LanguagesModel: Model<Languages>) {}  
4  
5    async create(createLanguageDto: CreateLanguageDto): Promise<Languages> {  
6      const Languages = new this.LanguagesModel(createLanguageDto);  
7      return Languages.save();  
8    }  
9  
10   async findAll(): Promise<Languages[]> {  
11     return this.LanguagesModel.find().exec();  
12   }  
13  
14   async findOne(id: string): Promise<Languages> {  
15     return this.LanguagesModel.findById(id).exec();  
16   }  
17  
18   async update(id: string, updateLanguageDto: UpdateLanguageDto): Promise<Languages> {  
19     return this.LanguagesModel.findByIdAndUpdate(id, updateLanguageDto).exec();  
20   }  
21  
22   async remove(id: string): Promise<Languages> {  
23     return this.LanguagesModel.findByIdAndDelete(id).exec();  
24   }  
25 }
```

Etapa 3: Consumo de Datos y Desarrollo Frontend

1. Introducción
 - Propósito de la Etapa
 - Alcance de la Etapa
 - Definiciones y Acrónimos
2. Creación de la Interfaz de Usuario (UI)
 - Diseño de la Interfaz de Usuario (UI) con HTML y CSS
 - Consideraciones de Usabilidad
 - Maquetación Responsiva
3. Programación Frontend con JavaScript (JS)
 - Desarrollo de la Lógica del Frontend
 - Manejo de Eventos y Comportamientos Dinámicos
 - Uso de Bibliotecas y Frameworks (si aplicable)
4. Consumo de Datos desde el Backend
 - Configuración de Conexiones al Backend
 - Obtención y Presentación de Datos
 - Actualización en Tiempo Real (si aplicable)
5. Interacción Usuario-Interfaz
 - Manejo de Formularios y Validación de Datos
 - Implementación de Funcionalidades Interactivas
 - Mejoras en la Experiencia del Usuario
6. Pruebas y Depuración del Frontend
 - Diseño de Casos de Prueba de Frontend
 - Pruebas de Usabilidad
 - Depuración de Errores y Optimización del Código
7. Implementación de la Lógica de Negocio en el Frontend
 - Migración de la Lógica de Negocio desde el Backend (si necesario)
 - Validación de Datos y Reglas de Negocio en el Frontend
8. Integración con el Backend
 - Verificación de la Comunicación Efectiva con el Backend
 - Pruebas de Integración Frontend-Backend