

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

Creación Módulo: Diseño de contenido

Autores Software I, II

Elkin Ricardo López Castillo, Michael Steven Barrios Martínez, Valentina Sierra Gloria, Juan Sebastián Tirado Verbel, Jhonatan Salas De la rosa

Github: @atoscano



Descripción del software.

La creación de contenidos para objetos virtuales de aprendizaje es necesaria debido a la personalización del aprendizaje, el acceso a información actualizada, la adaptación a contextos específicos y la integración de metodologías efectivas. Esto permite ofrecer recursos relevantes y útiles que se ajusten a las necesidades individuales de los docentes, brindar información actualizada y pertinente, abordar temas específicos de cada contexto educativo y utilizar estrategias pedagógicas efectivas para promover un aprendizaje significativo.

Tabla de contenido

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos.	6
1. Introducción	6
Propósito del Documento.....	6
Alcance del Proyecto Diseño de Contenido.....	6
Definiciones y Acrónimos	6
2. Descripción General.....	9
Objetivos del Sistema	9
Funcionalidad General	9
Usuarios del Sistema.....	11
Restricciones	11
3. Requisitos Funcionales	12
Casos de Uso.....	14
Diagramas de Flujo de Casos de Uso	15
Descripción detallada de cada caso de uso	15
Prioridad de Requerimientos.....	40
4. Requisitos No Funcionales	42
Requisitos de Desempeño	44
Requisitos de Seguridad.....	45
Requisitos de Usabilidad.....	46
Requisitos de Escalabilidad.	46
5. Modelado E/R	47
Diagrama de Entidad-Relación	47
Diagrama Relacional.....	48
Script de modelo relacional	49
Descripción de Entidades y Relaciones	50
Relaciones:	50
Reglas de Integridad Referencial	51
Colecciones (NoSLQ)	52
6. Anexos	53
Diagramas Adicionales	53
Referencias.....	53
Etapa 2: Persistencia de Datos con Backend	53
7. Introducción.	53
Propósito de la Etapa.	53

Alcance de la Etapa.....	54
8. Diseño de la Arquitectura de Backend.....	54
Descripción de la Arquitectura Propuesta.....	54
Componentes del Backend.....	55
Diagramas de Arquitectura.	55
9. Elección de la Base de Dato.....	56
Evaluación de Opciones (SQL o NoSQL).	56
Justificación de la Elección.....	57
Diseño de Esquema de Base de Datos.	57
10. Implementación del Backend.....	58
Elección del Lenguaje de Programación.....	58
Creación de la Lógica de Negocio	59
Desarrollo de Endpoints y APIs.....	60
11. Conexión a la Base de Datos	60
Configuración de la Conexión	60
Desarrollo de Operaciones CRUD.....	61
Manejo de Transacciones.....	62
12. Pruebas del Backend.....	63
Diseño de Casos de Prueba	63
Ejecución de Pruebas Unitarias y de Integración	63
Manejo de Errores y Excepciones.....	63
Etapa 3: Consumo de Datos y Desarrollo Frontend.....	64
13. Introducción	64
Propósito de la Etapa	65
Alcance de la Etapa.....	65
Definiciones y Acrónimos	65
14. Creación de la Interfaz de Usuario (UI).....	65
Diseño de la Interfaz de Usuario (UI) con HTML y CSS.....	65
Consideraciones de Usabilidad	65
Maquetación Responsiva	65
15. Programación Frontend con JavaScript (JS).....	65
Desarrollo de la Lógica del Frontend.....	65
Manejo de Eventos y Comportamientos Dinámicos.....	65
Uso de Bibliotecas y Frameworks (si aplicable).....	65
16. Consumo de Datos desde el Backend	66
Configuración de Conexiones al Backend	66
Obtención y Presentación de Datos	66

Actualización en Tiempo Real (si aplicable)	66
17. Interacción Usuario-Interfaz	66
Manejo de Formularios y Validación de Datos	66
Implementación de Funcionalidades Interactivas	66
Mejoras en la Experiencia del Usuario	66
18. Pruebas y Depuración del Frontend	66
Diseño de Casos de Prueba de Frontend	66
Pruebas de Usabilidad	66
Depuración de Errores y Optimización del Código	66
19. Implementación de la Lógica de Negocio en el Frontend	66
Migración de la Lógica de Negocio desde el Backend (si necesario)	67
Validación de Datos y Reglas de Negocio en el Frontend	67
20. Integración con el Backend	67
Verificación de la Comunicación Efectiva con el Backend	67
Pruebas de Integración Frontend-Backend	67
ANEXOS	67

Etapas 1 Diseño de la Aplicación y Análisis de Requisitos.

1. Introducción

Propósito del Documento

El propósito principal de este informe es proporcionar una comprensión clara y detallada del proceso de introducción de nuevas funciones a un OVA (Objeto Virtual de Aprendizaje). A través de este, se pretende mostrar de manera explícita cómo se planifica la mejora o adición de características de este. Contribuyendo a la experiencia de los usuarios y la efectividad del recurso educativo.

Alcance del Proyecto Diseño de Contenido

Este Módulo busca mejorar la creación de contenidos para objetos virtuales de aprendizaje, por lo que estos otorgan a los educadores la capacidad de adaptarse a las diversas tecnologías disponibles en el ámbito educativo, este facilita la creación y manejo de contenido en cualquier nivel formativo, combinando las diferentes herramientas de diseño, personalización y seguimiento que permiten la gestión de subida, toma y calificación de actividades. Esto no solo brinda información valiosa para la mejora continua del contenido, sino que también empodera a los educadores para adaptarse a las diferentes tecnologías.

Definiciones y Acrónimos

API: Interfaz de Programación de Aplicaciones (Application Programming Interface).

DBMS: Sistema de Gestión de Bases de Datos (Database Management System).

SQL: Lenguaje de Consulta Estructurada (Structured Query Language).

HTTP: Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol).

REST: Transferencia de Estado Representacional (Representational State Transfer).

JSON: Notación de Objetos de JavaScript (JavaScript Object Notation).

JWT: Token de Web JSON (JSON Web Token).

CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).

ORM: Mapeo Objeto-Relacional (Object-Relational Mapping).

MVC: Modelo-Vista-Controlador (Model-View-Controller).

API RESTful: API que sigue los principios de REST.

CI/CD: Integración Continua / Entrega Continua (Continuous Integration / Continuous Delivery).

SaaS: Software como Servicio (Software as a Service).

SSL/TLS: Capa de sockets seguros/Seguridad de la Capa de Transporte (Secure Sockets Layer/Transport Layer Security).

HTML: Lenguaje de Marcado de Hipertexto (Hypertext Markup Language).

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets).

JS: JavaScript.

DOM: Modelo de Objeto del Documento (Document Object Model).

UI: Interfaz de Usuario (User Interface).

UX: Experiencia del Usuario (User Experience).

SPA: Aplicación de Página Única (Single Page Application).

AJAX: Asíncrono JavaScript y XML (Asynchronous JavaScript and XML).

CMS: Sistema de Gestión de Contenido (Content Management System).

CDN: Red de Distribución de Contenido (Content Delivery Network).

SEO: Optimización de Motores de Búsqueda (Search Engine Optimization).

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment).

CLI: Interfaz de Línea de Comandos (Command Line Interface).

PWA: Aplicación Web Progresiva (Progressive Web App)

2. Descripción General

Objetivos del Sistema

El objetivo central es brindar un conjunto de herramientas innovadoras destinadas a revolucionar el proceso de diseño de contenidos, con el propósito de enriquecer el software educativo. Estas permiten a los usuarios agregar, editar y eliminar acciones de manera eficiente y efectiva, lo que simplifica aún más la creación, visualización y modificación de materiales educativos. Además de optimizar la gestión de contenidos, está diseñado para potenciar la interactividad entre estudiantes y docentes.

De esta manera, proporciona a los usuarios un software intuitivo y versátil, lo que contribuye a la mejora de la comunicación y el aprendizaje. Al facilitar el acceso y el control sobre diferentes temas, se convierte en una herramienta esencial para el enriquecimiento de la experiencia educativa, respaldando así los objetivos de la educación moderna.

Funcionalidad General

- **Creación y Edición Colaborativa:** Permite a los usuarios crear y editar contenido de forma colaborativa en tiempo real. Múltiples usuarios pueden trabajar en el mismo documento simultáneamente.
- **Carga de Imágenes y Multimedia:** Permite a los usuarios cargar imágenes, videos
- **Organización de Contenido:** Ofrece opciones para organizar y estructurar el contenido.

- **Historial de Revisiones:** Registra un historial de revisiones que permite a los usuarios rastrear los cambios realizados y restaurar versiones anteriores si es necesario.
- **Exportación e Impresión:** Ofrece la capacidad de exportar el contenido en varios formatos (PDF, imagen, etc.) y la opción de imprimirlo.
- **Acceso Seguro:** Proporciona medidas de seguridad para garantizar que solo los usuarios autorizados puedan acceder y editar.
- **Acceso Móvil:** Ofrece una experiencia de usuario optimizada en dispositivos móviles, permitiendo el acceso y la colaboración desde smartphones y tabletas.
- **Gestión de Usuarios y Permisos:** Permite a los administradores gestionar usuarios y definir permisos de acceso y edición.
- **Informes y Analíticas:** Proporciona información sobre el uso del software, como quién la ha editado, cuándo se realizaron cambios y estadísticas sobre el contenido.

Usuarios del Sistema

Los siguientes usuarios pueden interactuar con el software dependiendo de las funcionalidades.

Funcionalidad	Administradores	Docente Investigador	Docente Invitado	Alumno	Invitado
Creación y Edición		✓	✓	✓	
Carga de Imágenes y Multimedia		✓	✓	✓	
Organización de Contenido		✓	✓		
Historial de Revisiones	✓	✓	✓		
Exportación e Impresión		✓	✓	✓	
Acceso Seguro	✓	✓	✓	✓	
Acceso Móvil	✓	✓	✓	✓	✓
Gestión de Usuarios y Permisos	✓				
Informes y Analíticas	✓	✓	✓		

Restricciones

Solo usuarios agregados por un anfitrión tendrán acceso a las funcionalidades descritas en la tabla anterior, un anfitrión puede agregar otros anfitriones quienes pueden ser docentes, alumnos o invitados, también se les puede dar el rol de moderador y/o administrador. Las funcionalidades de estos dos roles no se han descrito aún.

3. Requisitos Funcionales

- **Creación y Edición Colaborativa:**

Los usuarios pueden crear y editar contenido en tiempo real.

Los usuarios pueden tener acceso a la edición de contenido.

- **Carga de Imágenes y Multimedia:**

Los usuarios deben poder cargar imágenes, videos y otros medios directamente en la pizarra.

- **Organización de Contenido:**

Ofrece opciones para organizar y estructurar el contenido.

- **Historial de Revisiones:**

La aplicación debe mantener un historial de revisiones que permita a los usuarios rastrear cambios realizados y restaurar versiones anteriores.

- **Exportación e Impresión:**

Los usuarios deben poder exportar el contenido en los diferentes formatos (PDF, imagen, etc.) y tener la opción de imprimirlo.

- **Acceso Seguro:**

Garantiza a los usuarios autorizados acceder y editar.

- **Acceso Móvil:**

Permite el acceso a dispositivos móviles y tabletas.

- **Gestión de Usuarios y Permisos:**

Gestionar usuarios

Definir permisos de acceso y edición.

- **Informes y Analíticas:**

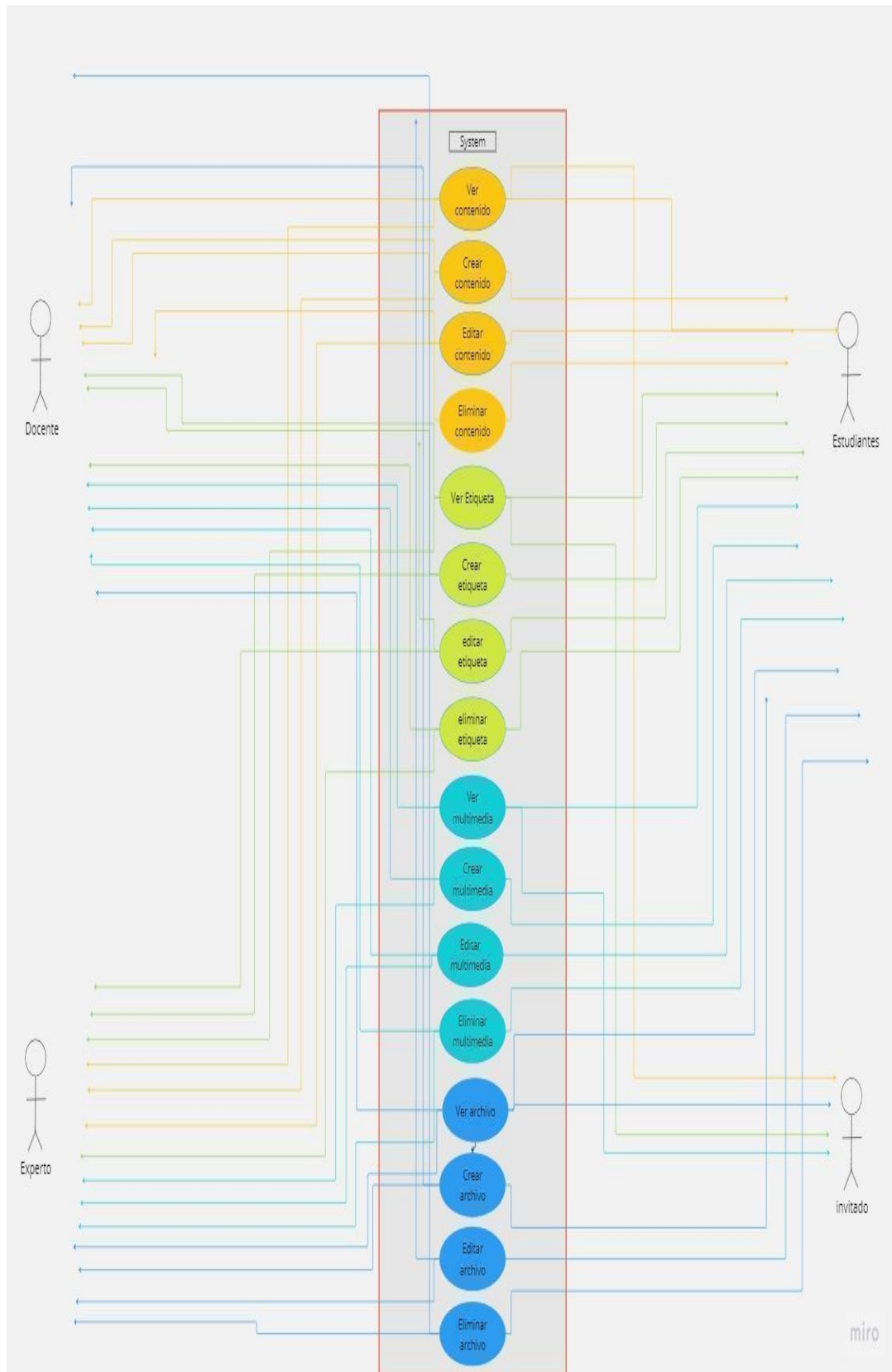
Proporciona información sobre el uso del software.

Como quien la ha editado.

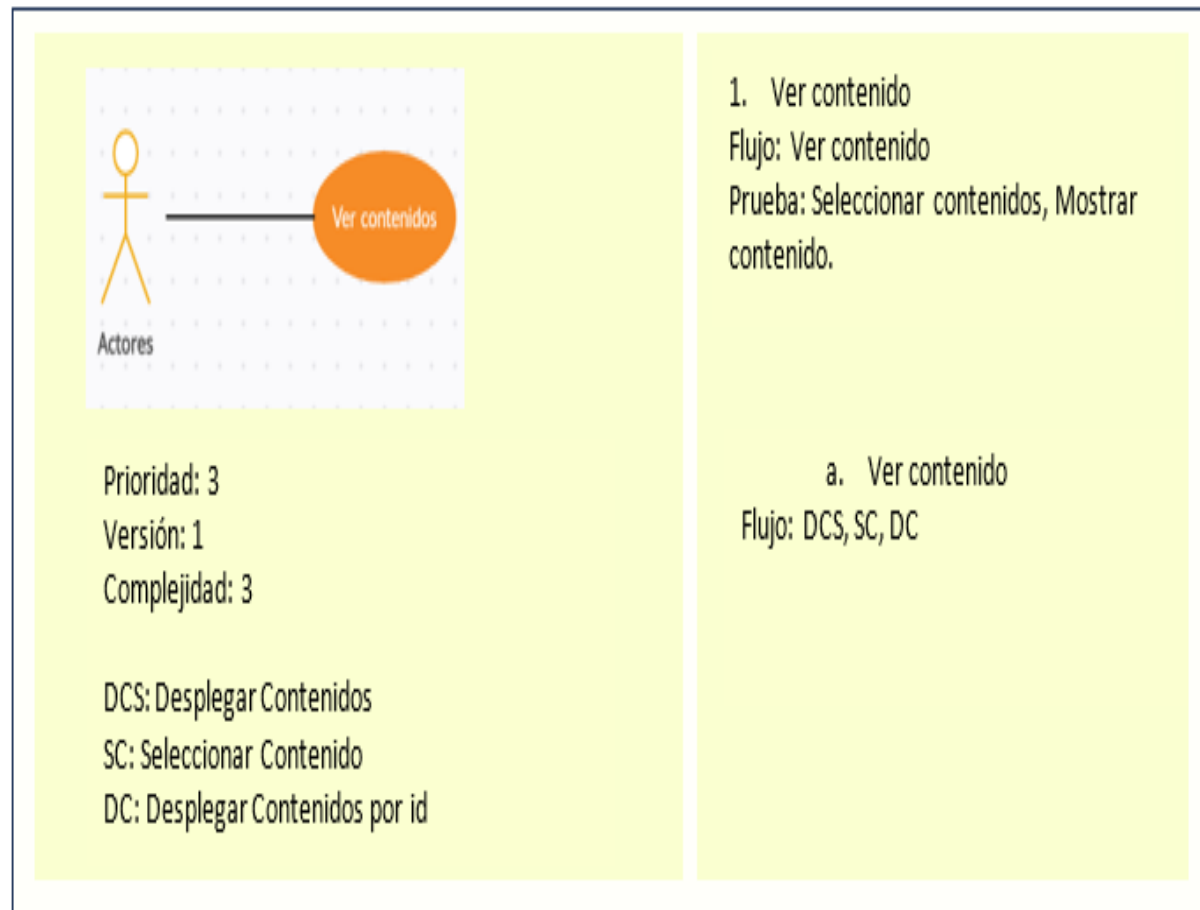
Cuando se realizaron cambios y estadísticas sobre el contenido.

Casos de Uso

Diagrama de caso de uso



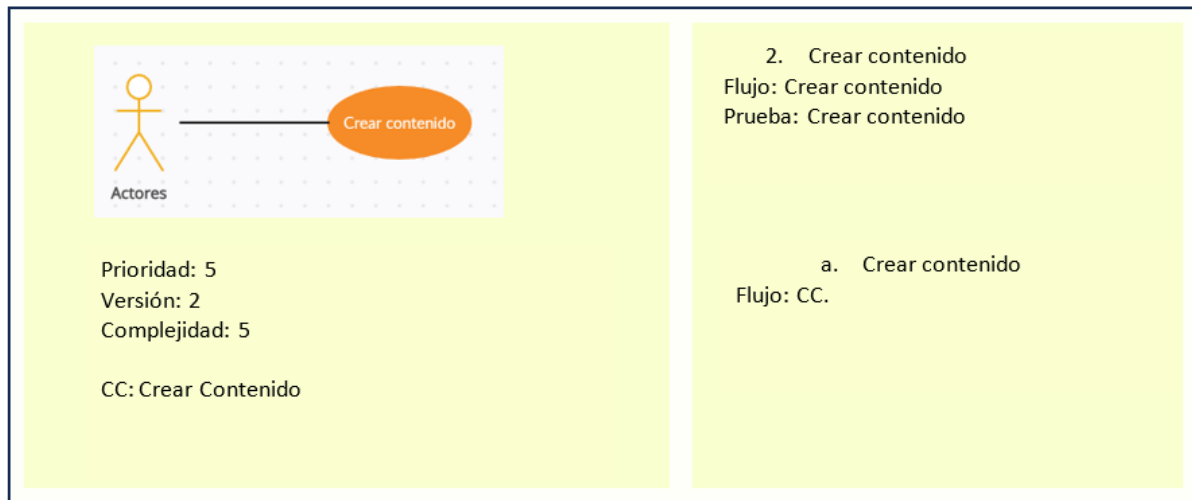
Diagramas de Flujo de Casos de Uso



Descripción detallada de cada caso de uso

CASO No. 1 Ver contenidos

ID:	CU-1	
Nombre	Ver contenidos	
Actores	Docentes, Invitados, Estudiantes, Expertos	
Objetivo	Este caso permite ver un contenido	
Urgencia	3	
Esfuerzo	3	
Precondiciones		
Flujo Normal	Actores	Sistema
		1. Desplegar contenidos
	2. Selecciona Contenido	
		3. Desplegar contenido
	4. Ver contenido	
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

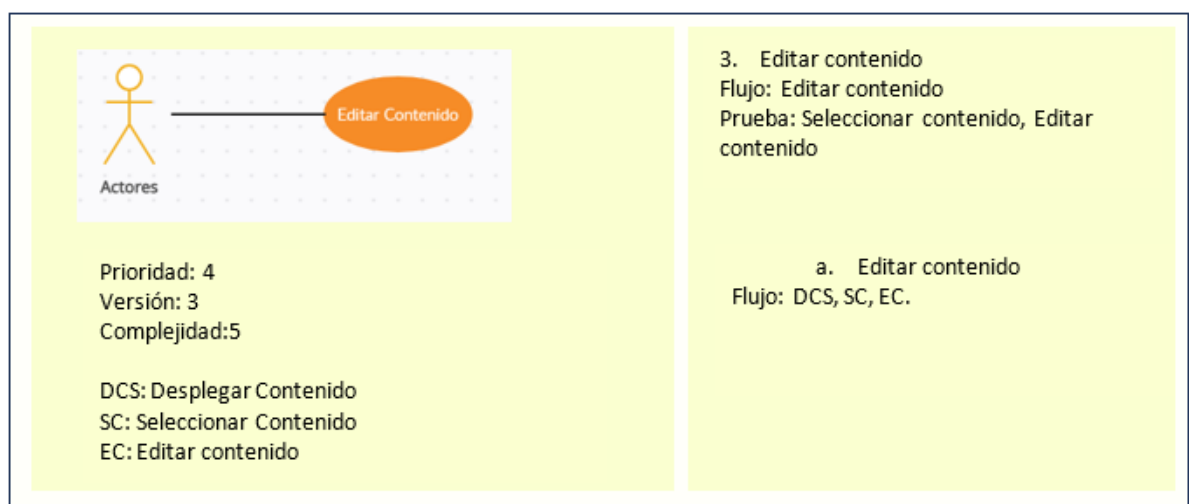


Descripción detallada de cada caso de uso

CASO No. 2 Crear contenido

ID:	CU-2	
Nombre	Ver contenidos	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite crear un contenido	
Urgencia	5	
Esfuerzo	5	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige la opción "Crear"	
		2. Desplegar opciones de tipo de contenido
	3. Crear contenido	
		4. Guardar cambios

Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

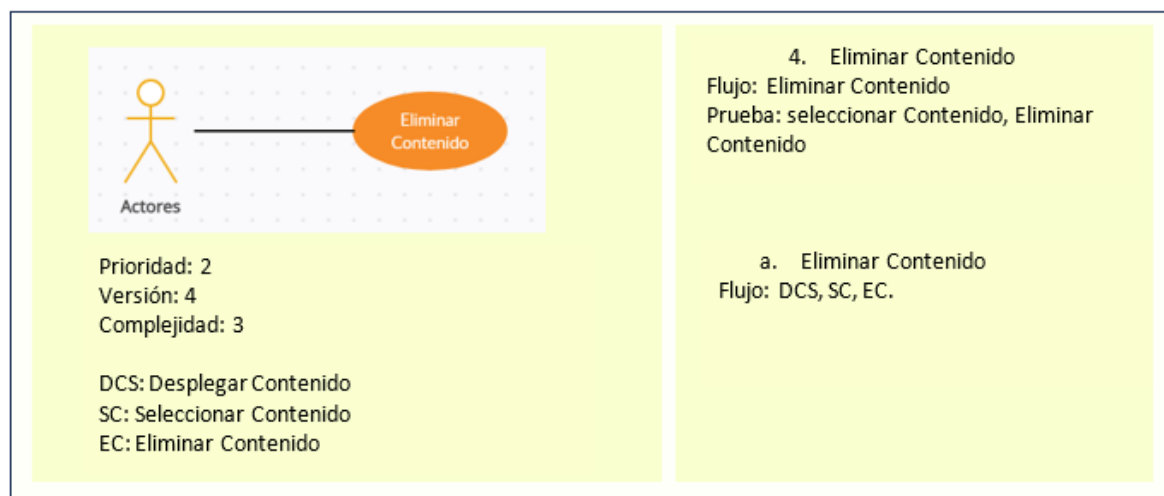


Descripción detallada de cada caso de uso

CASO No. 3 Crear Archivo

ID:	CU-13	
Nombre	Ver contenidos	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite crear un archivo	
Urgencia	5	
Esfuerzo	1	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige la opción “Crear”	
		2. Desplegar formato con herramientas
	3. Crear archivo	
		4. Pedir archivo
	5. subir archivo	
		6. Montar archivo
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		

Excepciones		

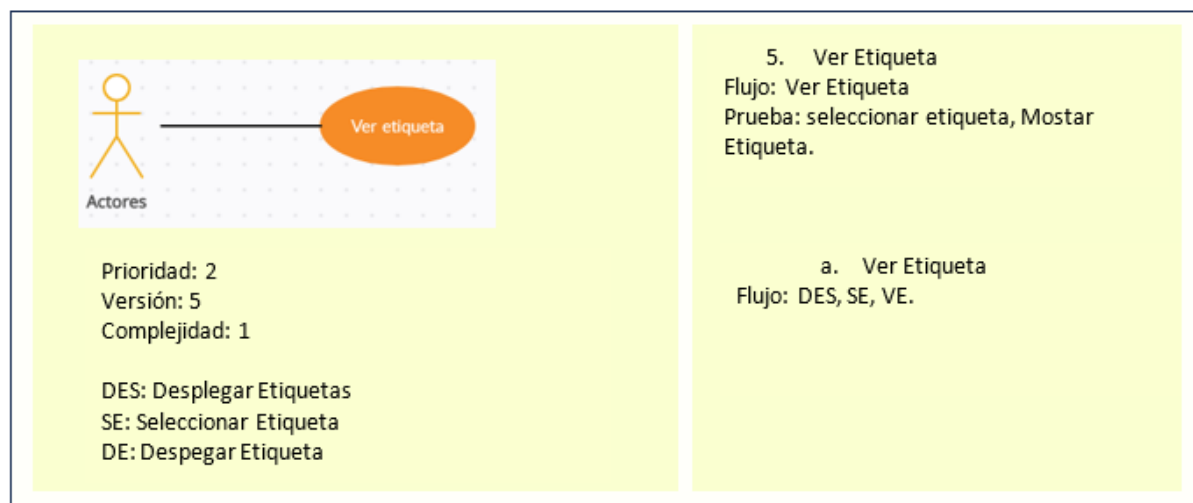


Descripción detallada de cada caso de uso

CASO No. 4 Editar contenido

ID:	CU-3	
Nombre	Editar contenido	
Actores	Docentes, Expertos	
Objetivo	Este caso permite editar contenidos creados	
Urgencia	4	
Esfuerzo	5	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige contenido	
		2. Desplegar contenido
	3. Editar contenido	
		4. Desplegar edición de contenido

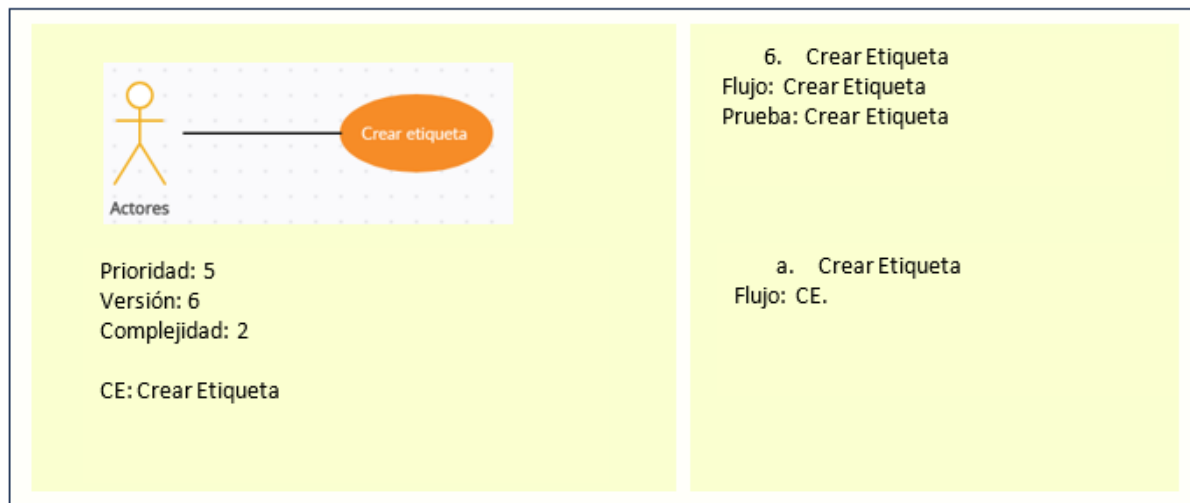
	5. Guardar cambios	
		6. Retornar a contenido actualizado
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 5 Eliminar contenido

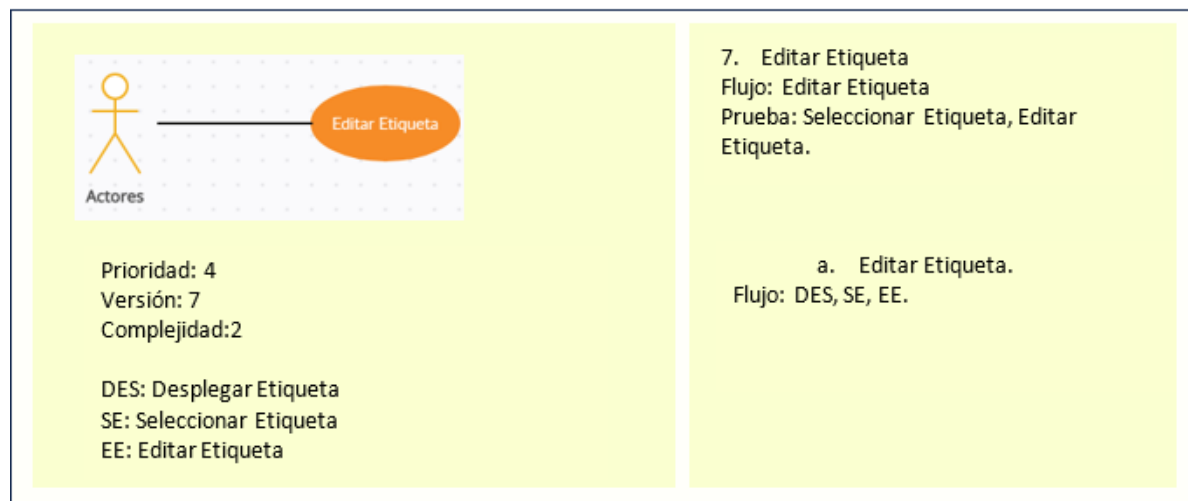
ID:	CU-4	
Nombre	Eliminar contenido	
Actores	Docentes, Estudiantes	
Objetivo	Este caso permite eliminar contenidos creados	
Urgencia	2	
Esfuerzo	3	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige contenido	
		2. Desplegar contenido
	3. Opción eliminar	
		4. Eliminar contenido
		5. Retornar a contenidos
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 6 Ver etiqueta

ID:	CU-5	
Nombre	Ver etiqueta	
Actores	Docentes, Expertos, Estudiantes, Invitados	
Objetivo	Este caso permite ver etiquetas	
Urgencia	2	
Esfuerzo	1	
Precondiciones		
Flujo Normal	Actores	Sistema
		1. Desplegar etiquetas
	2. Seleccionar etiqueta	
		3. Desplegar contenidos de etiqueta
	4. Ver contenido	
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

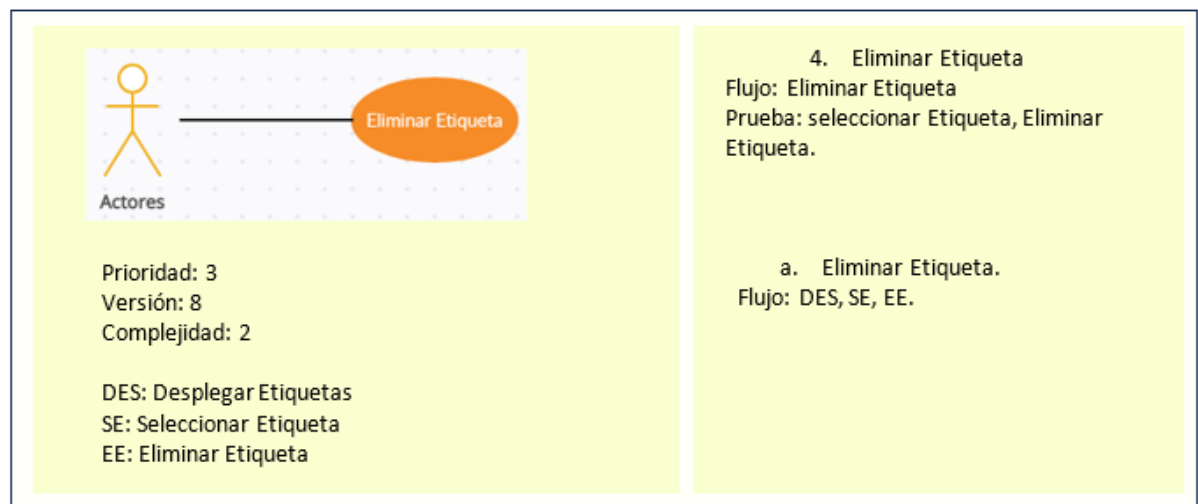


Descripción detallada de cada caso de uso

CASO No. 7 Crear etiqueta

ID:	CU-6	
Nombre	Crear etiqueta	
Actores	Estudiantes, Expertos	
Objetivo	Este caso permite crear una etiqueta	
Urgencia	5	
Esfuerzo	2	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	2. Elige la opción “Crear”	
		2. Desplegar opciones de tipo de contenido
	3. Crear etiqueta	
		4. Guardar cambios

Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 8 Editar Etiqueta

ID:	CU-7	
Nombre	Editar etiqueta	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite editar etiquetas creadas	
Urgencia	4	
Esfuerzo	2	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige etiqueta	
		2. Desplegar contenido de etiqueta
	3. Opción "Editar"	
		4. Desplegar formato para modificar la etiqueta
	5. Guardar cambios	
		6. Actualizar y montar etiqueta
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 9 Eliminar Etiqueta

ID:	CU-4	
Nombre	Eliminar Etiqueta	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite eliminar Etiquetas	
Urgencia	3	
Esfuerzo	2	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige etiqueta	
		2. Desplegar contenido
	3. Opción eliminar	
		4. Eliminar contenido
		5. Retornar a Etiquetas

Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

Actores

Prioridad: 4
Versión: 10
Complejidad: 3

DAS: Desplegar Archivos
SA: Seleccionar Archivo
EA: Eliminar Archivo

10. Eliminar Archivo
Flujo: Eliminar Archivo
Prueba: seleccionar Archivo, Eliminar Archivo.

a. Eliminar Archivo.
Flujo: DAS, SA, EA.

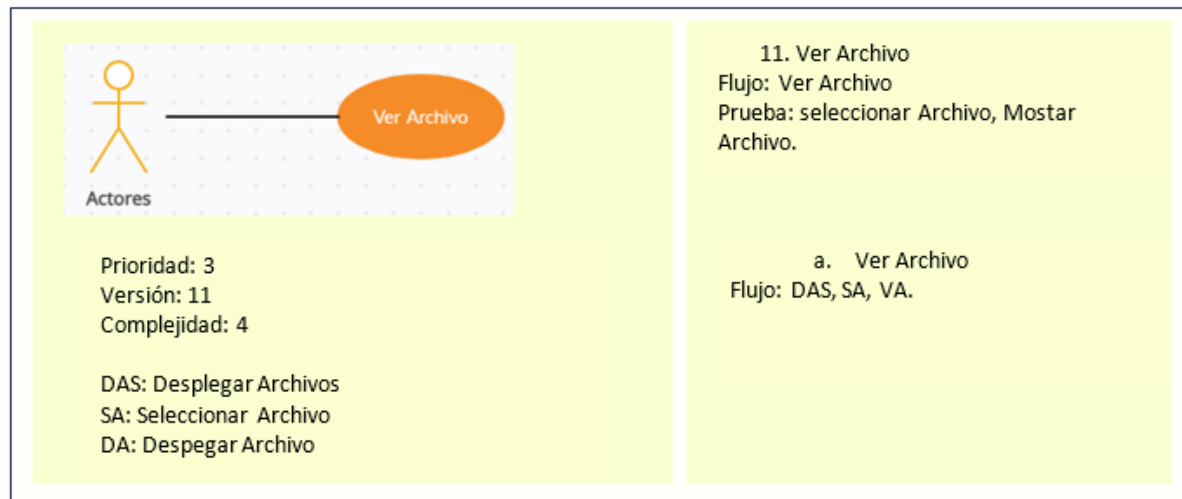
Descripción detallada de cada caso de uso

CASO No. 10 Editar Multimedia

ID:	CU-9
------------	------

Nombre	Editar Multimedia	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite editar multimedia añadida	
Urgencia	3	
Esfuerzo	5	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Selecciona multimedia	
		2. Mostrar viñeta con menú desplegable
	3. Opción “Editar”	
		4. Desplegar formato para modificar multimedia
	5. Guardar cambios	
		6. Actualizar y montar multimedia
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

--	--	--

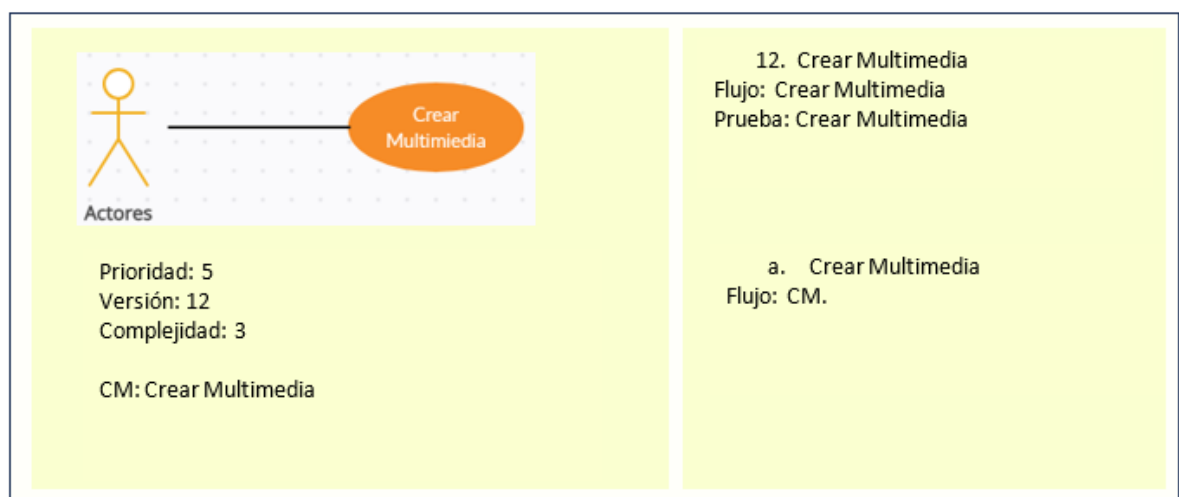


Descripción detallada de cada caso de uso

CASO No. 11 Crear Multimedia

ID:	CU-12	
Nombre	Crear Multimedia	
Actores	Estudiantes, Expertos, Docente	
Objetivo	Este caso permite crear una multimedia	
Urgencia	5	
Esfuerzo	3	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige la opción “Crear”	
		2. Desplegar opciones de tipo de multimedia
	3. Subir archivo multimedia	
		4. Opciones para multimedia

	5. Montar archivo	
		6. Guardar y montar multimedia
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 12 Editar Archivo

ID:	CU-16	
Nombre	Editar Archivo	
Actores	Docentes, Estudiantes, Expertos	
Objetivo	Este caso permite editar archivos	
Urgencia	4	
Esfuerzo	1	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige archivo	
		2. Desplegar menú de opciones
	3. Opción “Editar”	
		4. Desplegar formato para modificar el archivo
	5. Guardar cambios	
		6. Actualizar y montar etiqueta
Flujo alternativo 1		
Flujo alternativo 2		

Post-condiciones		
Excepciones		

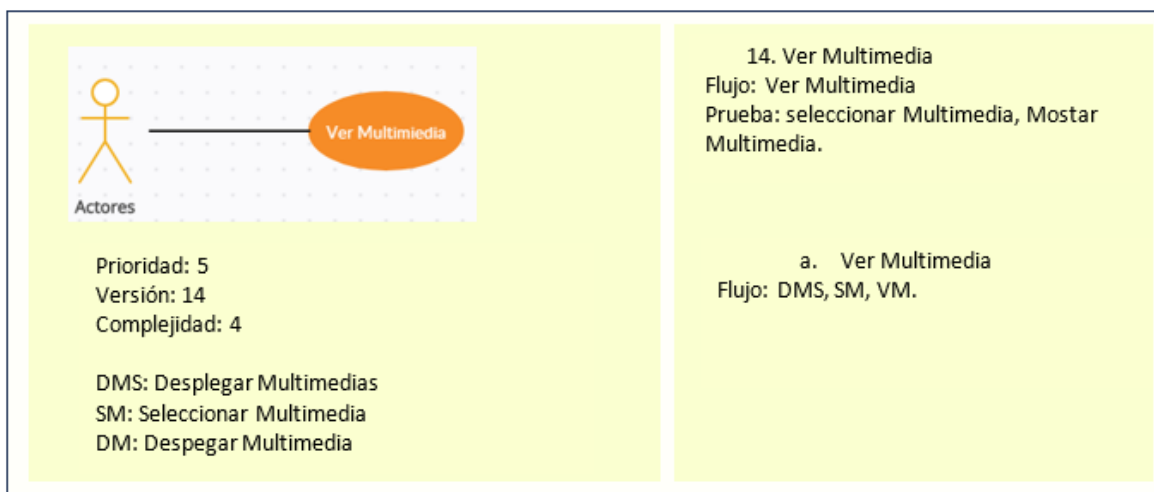


Descripción detallada de cada caso de uso

CASO No. 13 Ver Multimedia

ID:	CU-14
Nombre	Ver Multimedia
Actores	Docentes, Invitados, Estudiantes, Expertos
Objetivo	Este caso permite ver Multimedia
Urgencia	5
Esfuerzo	4
Precondiciones	

Flujo Normal	Actores	Sistema
		1. Desplegar multimedia
	2. Selecciona multimedia	
		3. Desplegar multimedia elegida
	4. Ver multimedia	
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

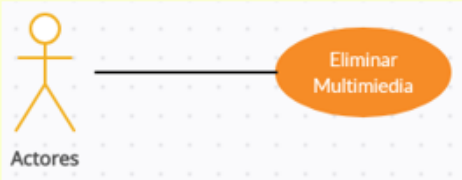


Descripción detallada de cada caso de uso

CASO No. 14 Eliminar Multimedia

ID:	CU-15	
Nombre	Eliminar Multimedia	
Actores	Docentes, Estudiantes	
Objetivo	Este caso permite eliminar Multimedia	
Urgencia	4	
Esfuerzo	4	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige multimedia	
		2. Desplegar menú de opciones
	3. Opción eliminar	
		4. Eliminar contenido
		5. Retornar a contenidos

Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Actores

Prioridad: 4
Versión: 15
Complejidad: 4

DMS: Desplegar Multimedia
SM: Seleccionar Multimedia
EM: Eliminar Multimedia

15. Eliminar Multimedia
Flujo: Eliminar Multimedia
Prueba: seleccionar Multimedia, Eliminar Multimedia.

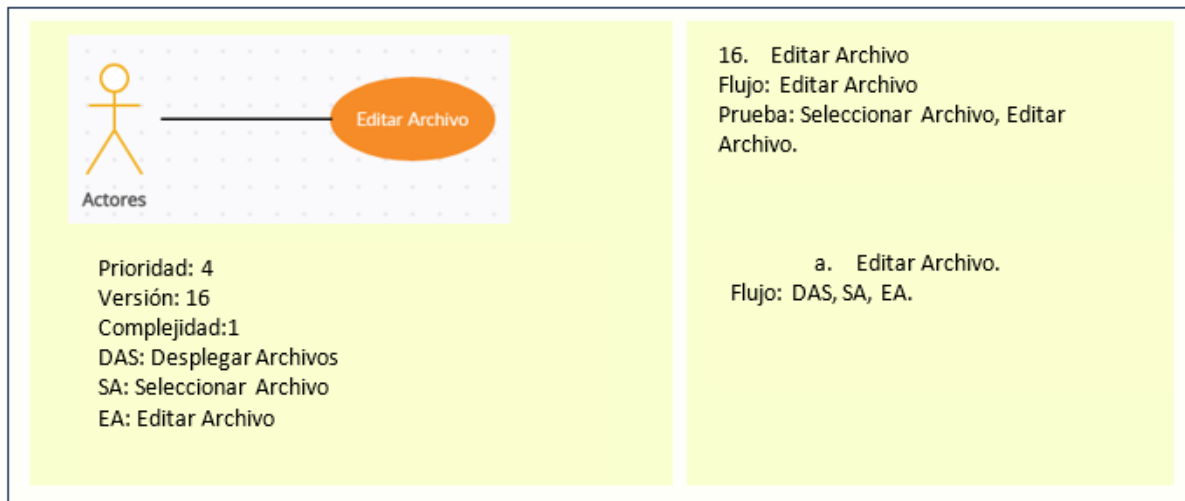
a. Eliminar Multimedia.
Flujo: DMS, SM, EM.

Descripción detallada de cada caso de uso

CASO No. 15 Eliminar Archivo

ID:	CU-10
Nombre	Eliminar Archivo
Actores	Docentes, Estudiantes

Objetivo	Este caso permite eliminar Archivos	
Urgencia	4	
Esfuerzo	3	
Precondiciones	- El actor debe iniciar sesión	
Flujo Normal	Actores	Sistema
	1. Elige multimedia	
		2. Desplegar menú de opciones
	3. Opción eliminar	
		4. Eliminar contenido
		5. Retornar a contenidos
Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		



Descripción detallada de cada caso de uso

CASO No. 16 Ver Archivos

ID:	CU-11	
Nombre	Ver archivos	
Actores	Docentes, Invitados, Estudiantes, Expertos	
Objetivo	Este caso permite ver Archivos	
Urgencia	3	
Esfuerzo	4	
Precondiciones		
Flujo Normal	Actores	Sistema
		1. Desplegar Archivos
	2. Selecciona Archivo	
		3. Desplegar Archivo
	4. Ver Archivo	

Flujo alternativo 1		
Flujo alternativo 2		
Post-condiciones		
Excepciones		

Prioridad de Requerimientos

A partir del análisis de requerimientos, funcionalidades y el proceso de design thinking, se concreta la siguiente matrix de prioridad de requerimientos.

Para la interpretación se tiene en cuenta la siguiente escala con sus valores.

Eje de Urgencia:

- Obligatoria (5)
- Alta (4)
- Moderada (3)
- Menor (2)
- Baja (1)

Eje de Complejidad:

- Muy alto (5)
- Alto (4)
- Medio (3)
- Bajo (2)
- Muy bajo (1)

	Urgencia					
Complejidad		1-Baja	2-Menor	3-Moderada	4-Alta	5-Obligatoria
	5-Muy alto	5	10	15	20	25

		CU-13	CU-6	CU-9 CU-12	CU-3 CU-14	CU-2
4-Alto	4	8	12	16	20	
	CU-16	CU-7	CU-11 CU-10	CU-15		
3-Medio	3	6	9	12	15	
		CU-8	CU-1	CU-10		
2-Bajo	2	4	6	8	10	
	CU-5					
1-Muy bajo	1	2	3	4	5	

<https://asana.com/es/resources/priority-matrix>

4. Requisitos No Funcionales

Seguridad:

Es necesario asegurar la protección de la información y la verificación de identidad de los usuarios mediante técnicas de cifrado.

Rendimiento:

La aplicación debe proporcionar un desempeño excelente, facilitando la colaboración en tiempo real, incluso cuando se trata de un gran número de usuarios.

Escalabilidad:

Es esencial que la plataforma tenga la capacidad de adaptarse al crecimiento en la cantidad de usuarios y al volumen de contenido, siendo escalable para gestionar dichos incrementos.

Disponibilidad:

Es fundamental que la aplicación permanezca accesible y operativa de manera continua, reduciendo al mínimo cualquier periodo de inactividad.

Compatibilidad con Dispositivos:

El software debe ser capaz de funcionar en diversos dispositivos, como computadoras de escritorio, tabletas y teléfonos móviles, garantizando su compatibilidad.

Usabilidad:

La interfaz de usuario debe ser sencilla y fácil de utilizar, adaptándose a usuarios con distintos niveles de habilidad de manera intuitiva.

Accesibilidad:

Debe ser accesible para cualquier usuario.

Cumplimiento Normativo:

Debe cumplir con regulaciones y normativas de privacidad y seguridad de datos.

Tiempo de Respuesta:

La aplicación debe tener tiempos de respuesta rápidos para mantener una experiencia de usuario fluida.

Requisitos de Desempeño

1. **Rendimiento en Tiempo Real:** El software debe proporcionar un rendimiento en tiempo real, lo que significa que los cambios realizados por los usuarios deben reflejarse instantáneamente para todos los colaboradores, incluso cuando múltiples usuarios trabajen simultáneamente.

2. **Tiempo de Carga Rápido:** Se debe cargar de manera eficiente, y los usuarios no deben experimentar tiempos de carga excesivamente largos al acceder o al editar contenido. Se requiere que los componentes estén bien diseñados y acoplados. Por lo general los componentes de la arquitectura sigue el paradigma de la programación orientada a componentes.

3. **Optimización de Recursos:** El sistema debe estar optimizado para utilizar eficientemente los recursos del servidor, minimizando el uso de CPU y memoria. El renderizado de los componentes adecuados garantiza este requisito.

Requisitos de Seguridad

1. **Acceso Seguro:** Se debe implementar una autenticación segura para garantizar que solo usuarios autorizados tengan acceso. Esto puede incluir autenticación de dos factores, inicio de sesión único (SSO), autenticación con JWT o Auth 2.

2. **Protección de Datos:** El software debe garantizar la protección de datos sensibles, como información del usuario y contenido compartido. Se debe cifrar la información en tránsito y en reposo.

3. **Auditoría y Registro de Actividades:** El sistema debe mantener registros de actividades, lo que incluye registros de cambios en la plataforma, acceso de usuarios y eventos relevantes para la seguridad.

4. **Control de versiones:** El sistema debe llevar un registro de los cambios de los datos gestionados, así como también los datos mismos de la estructura del componente.

5. **Variables de entorno:** El sistema debe ser manejado con variables de entorno que garanticen su fácil incorporación con otros módulos y la migración entre plataformas, así como también almacenar los datos iniciales del servidor como lo son las bases de datos y las llaves de autenticación, entre otras.

Requisitos de Usabilidad

1. **Interfaz Intuitiva:** La interfaz de usuario de la plataforma debe ser intuitiva y fácil de usar, permitiendo a los usuarios realizar acciones como dibujar, agregar contenido y colaborar sin dificultad.
2. **Compatibilidad con Dispositivos:** Ser compatible con una variedad de dispositivos, incluyendo computadoras de escritorio, tabletas y dispositivos móviles, y debe adaptarse a diferentes tamaños de pantalla.
3. **Documentación y Ayuda en Línea:** Se debe proporcionar documentación clara y ayuda en línea para los usuarios, incluyendo tutoriales y recursos de soporte.

Requisitos de Escalabilidad.

1. **Manejo de Cargas Elevadas:** El sistema debe ser escalable para manejar un gran número de usuarios simultáneamente, sin degradación significativa del rendimiento.
2. **Balanceo de Carga:** Se debe implementar un mecanismo de balanceo de carga para distribuir las solicitudes de usuarios de manera equitativa entre los servidores para garantizar la escalabilidad.
3. **Arquitectura de Backend Escalable:** La arquitectura del backend debe estar diseñada para escalar horizontalmente, lo que permite agregar más recursos de hardware a medida que aumenta la demanda.

5. Modelado E/R

Diagrama de Entidad-Relación

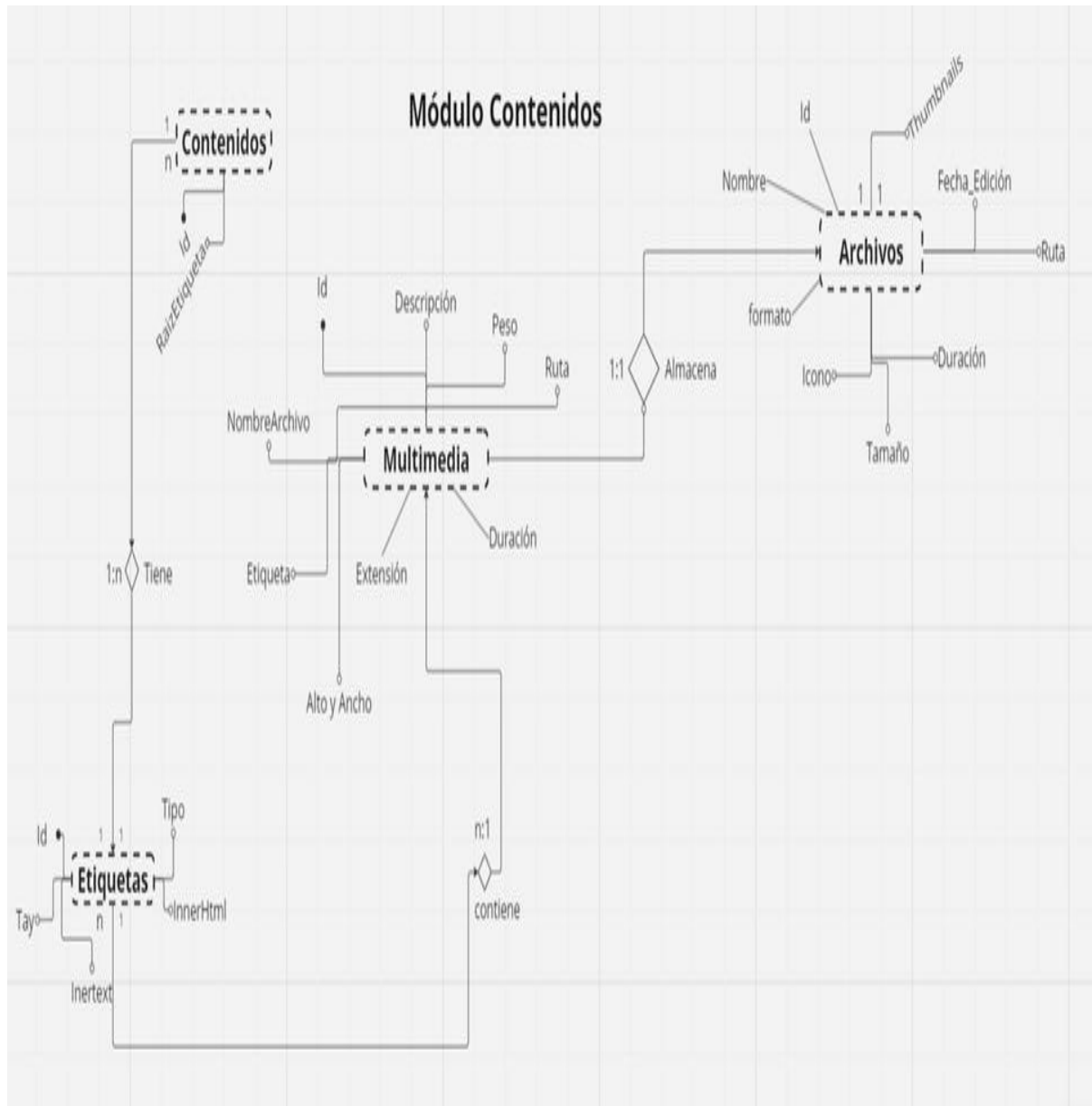
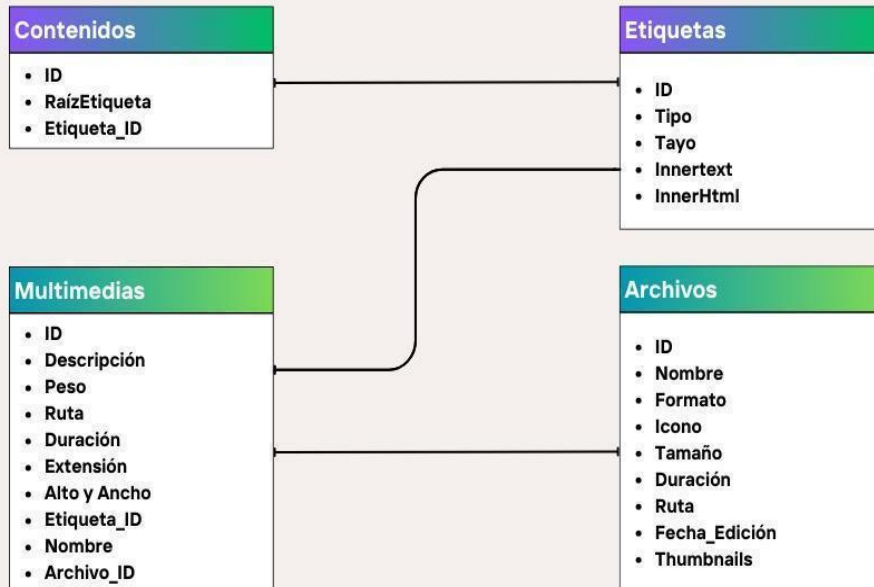


Diagrama Relacional

Modelo relacional

DISEÑO DE CONTENIDOS [MODESEC]



Script de modelo relacional

Entidad: Contenidos (

Id INT PRIMARY KEY,
RaizEtiquetas VARCHAR,

);

Entidad: Etiquetas (

Id INT PRIMARY KEY,
Tay VARCHAR,
InerText VARCHAR,
Tipo VARCHAR,
InnerHtml VARCHAR,
-- Otras columnas específicas de Etiquetas

);

Entidad: Multimedia (

Id INT PRIMARY KEY,
NombreArchivo VARCHAR,
Etiqueta INT, -- Clave foránea referenciando a Etiquetas.Id
AltoYAncho VARCHAR,
Extension VARCHAR,
Duracion INT,
Ruta VARCHAR,
Peso INT,
Descripcion VARCHAR,
FOREIGN KEY (Etiqueta) REFERENCES Etiquetas(Id)

);

Entidad: Archivo (

Id INT PRIMARY KEY,
Nombre VARCHAR,
Formato VARCHAR,
Icono VARCHAR,
Tamaño INT,
Duracion INT,
Ruta VARCHAR,
FechaYEdicion TIMESTAMP,
Thumbnails VARCHAR,

);

Descripción de Entidades y Relaciones

1. Entidad: contenidos.

Atributos: Id, RaizEtiquetas.

2. Entidad: Etiquetas.

Atributos: Id, Tay, InerText, Tipo, InnerHtml.

3. Entidad: Multimedia.

Atributos: Id, NombreArchivo, Etiqueta, AltoYAncho, Extension, Duracion, Ruta, Peso, Descripcion.

4. Entidad: Archivo.

Atributos: Id, Nombre, Formato, Icono, Tamaño, Duracion, Ruta, FechaYEdicion, Thumbnails.

Relaciones:

- "Content" se relaciona con "Labels" para indicar que content esta relacionado directamente a labels, labels esta relacionado con Content.

- "labelst" se relaciona dentro de "Content".

- Labels esta directamente relacionado con Multimedia.

- "Multimedia" se relaciona dentro de "labels".

- "Multimedia" se relaciona directamente con "Files".

"Files" se encuentra en "Multimedia"

Reglas de Integridad Referencial

“Content” puede contener etiquetas.

“Labels” debe estar dentro de contenido.

“Labels” puede tener **“Multimedias”**

“Multimedia” debe estar dentro de **“Labels”**

“Multimedia” puede tener **“Files”**

pero **“Files”** debe estar dentro de **“Multimedia”**

Colecciones (NoSLQ)

Entidad: Contenidos (

Id INT PRIMARY KEY,
RaizEtiquetas VARCHAR,

);

Entidad: Etiquetas (

Id INT PRIMARY KEY,
Tay VARCHAR,
InerText VARCHAR,
Tipo VARCHAR,
InnerHtml VARCHAR,
-- Otras columnas específicas de Etiquetas

);

Entidad: Multimedia (

Id INT PRIMARY KEY,
NombreArchivo VARCHAR,
Etiqueta INT, -- Clave foránea referenciando a Etiquetas.Id
AltoYAncho VARCHAR,
Extension VARCHAR,
Duracion INT,
Ruta VARCHAR,
Peso INT,
Descripcion VARCHAR,
FOREIGN KEY (Etiqueta) REFERENCES Etiquetas(Id)

);

Entidad: Archivo (

Id INT PRIMARY KEY,
Nombre VARCHAR,
Formato VARCHAR,
Icono VARCHAR,
Tamaño INT,
Duracion INT,
Ruta VARCHAR,
FechaYEdicion TIMESTAMP,
Thumbnails VARCHAR,

);

6. Anexos

Diagramas Adicionales

Referencias

Etapas 2: Persistencia de Datos con Backend

7. Introducción.

En esta sección se implementa una efectiva persistencia de datos en el backend contribuyendo a la creación de sistemas que respaldan la gestión y presentación de información relevante para los usuarios, este sienta las bases para un entorno más dinámico al proporcionar una infraestructura sólida para la gestión de información, no solo de almacenamiento de datos, sino también una plataforma que facilita la entrega eficiente de contenidos significativos a nuestros usuarios.

Esta implementación también mejorará la capacidad para adaptarnos de manera ágil a las cambiantes necesidades y expectativas de nuestros usuarios.

Propósito de la Etapa.

El propósito de la etapa de diseño de contenido en el módulo de Persistencia de Datos con Backend es desarrollar un modelo de diseño de contenido práctico que permita la creación efectiva de sistemas que gestionen información de manera duradera y eficiente, contribuyendo al desarrollo de las necesidades específicas de la aplicación y los usuarios.

Alcance de la Etapa.

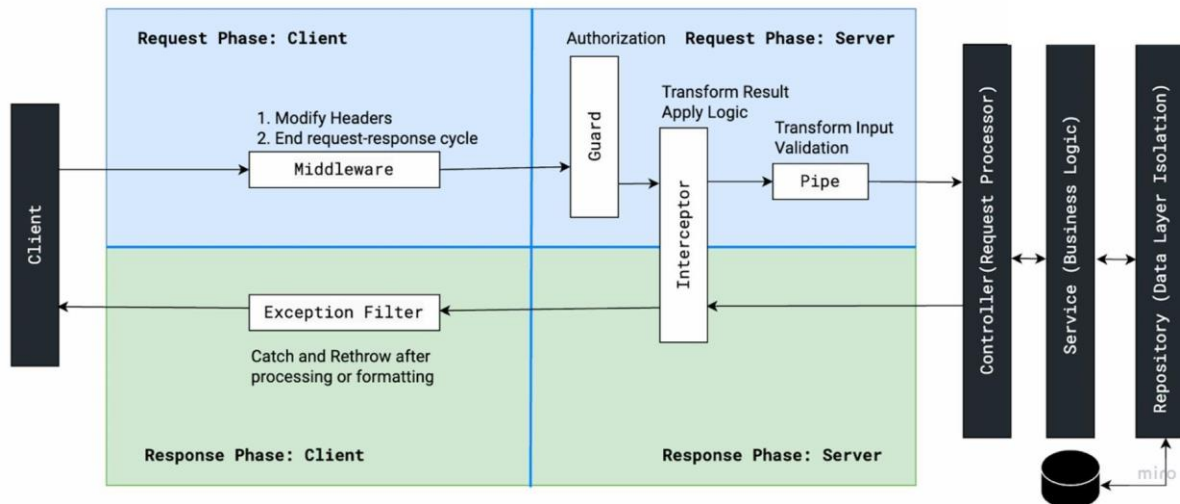
El alcance se centra en la integración efectiva de la persistencia de datos en el diseño de contenido de una aplicación. Esto implica abordar aspectos como el modelado de datos, donde se definirán entidades, relaciones y atributos para respaldar la presentación y gestión del contenido.

8. Diseño de la Arquitectura de Backend.

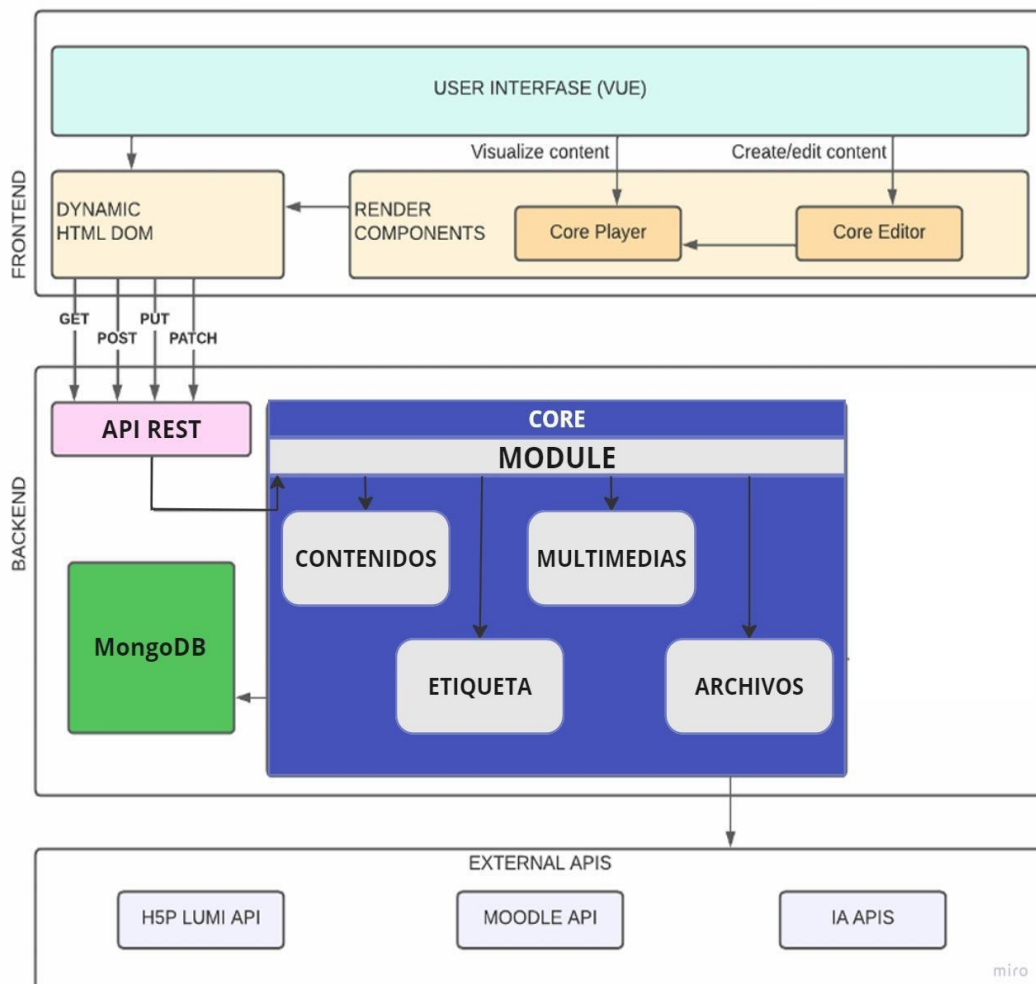
Descripción de la Arquitectura Propuesta.

El módulo pizarra compartida es una herramienta digital que permite a múltiples usuarios colaborar en tiempo real en un espacio virtual común que pueda manejar y mostrar diversos formatos de contenido multimedia, como imágenes, videos, documentos, etc. Esto permitiría a los usuarios incorporar y discutir contenido multimedia directamente en la pizarra, al igual que los elementos creados en la pizarra compartida pueden ser exportados o visualizados dentro de la estructura de la herramienta de creación de contenidos

Componentes del Backend.



Diagramas de Arquitectura.



9. Elección de la Base de Dato.

Evaluación de Opciones (SQL o NoSQL).

Características	Base de Datos Relacional (SQL)	Base de Datos NoSQL
Ventajas		
- Estructura y Consistencia	Adecuada para datos altamente estructurados y relaciones complejas.	Flexibilidad de esquema para datos no estructurados o semiestructurados.
- Integridad Referencial	Mantiene integridad referencial mediante claves foráneas.	-
- Consultas Complejas	Permite realizar consultas complejas y agregaciones de datos mediante SQL.	Puede tener limitaciones en consultas complejas.
8- Escalabilidad Vertical	Escalabilidad vertical mediante aumento de capacidad de hardware.	Escalabilidad horizontal agregando más nodos al clúster.
Desventajas		
- Esquema Rígido	Un esquema rígido puede ser limitante cuando los requisitos de datos cambian.	Mayor flexibilidad para adaptarse a cambios en los requisitos de datos.
- Escalabilidad Limitada	La escalabilidad horizontal puede ser más compleja y costosa.	Ideal para manejar grandes volúmenes de datos y escalar horizontalmente.
Consideraciones Específicas		
- Consistencia	Consistencia inmediata mediante transacciones (SQL).	Consistencia eventual en algunas implementaciones NoSQL.
- Complejidad en Consultas	Puede manejar consultas complejas y agregaciones.	Puede tener limitaciones en consultas complejas.
- Estándares	Estándares bien definidos.	Menos estándares en comparación.

Justificación de la Elección.

Una base de datos NoSQL se fundamenta en criterios estratégicos que buscan maximizar la eficiencia y el rendimiento de la aplicación como plataforma para diseñar modelos de contenidos, permitiendo esquemas dinámicos y adaptativos. Esta elección se respalda por la capacidad de las bases de datos NoSQL para gestionar datos no estructurados de manera eficiente. Además, la rapidez en las operaciones de lectura y escritura contribuye a una experiencia de usuario más ágil y receptiva.

Diseño de Esquema de Base de Datos.

```
TS app.module.ts 2 | TS app.controller.ts 1 | TS content.schema.ts 2 X
src > content > schema > TS content.schema.ts > ...
1  import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
2  import { Document } from 'mongoose';
3
4  export interface content extends Document{
5    title: string;
6    artist: string;
7    duration: number;
8  }
9
10  @Schema({timestamps:true})
11  export class content extends Document{
12    @Prop({ require: true})
13    title: string;
14
15    @Prop({ require: true})
16    artist: string;
17
18    @Prop({ require: true})
19    duration:number;
20  }
21  export const contentschema= SchemaFactory.createClass(content);
```

10. Implementación del Backend

Elección del Lenguaje de Programación

Lenguaje de Programación: Node.js

JavaScript en Backend: Node.js utiliza JavaScript en el servidor, lo que permite una sincronización más fácil entre el frontend y el backend del software. Esto puede facilitar la coherencia en la lógica del lado del cliente y del servidor. además de su gran ecosistema lo que facilita la integración de diversas funcionalidades y bibliotecas para el desarrollo del módulo de contenidos.

Base de Datos: MongoDB (Atlas)

- **Esquema Flexible:** MongoDB es una base de datos NoSQL que utiliza un modelo de documentos BSON (similar a JSON). Esto es beneficioso cuando el esquema de datos es variable o puede cambiar con el tiempo, lo cual es común en aplicaciones de contenido.
- **Escalabilidad Horizontal:** MongoDB es escalable horizontalmente, lo que significa que puedes agregar más servidores a tu conjunto para manejar mayores volúmenes de datos o tráfico. Esto es útil para aplicaciones que pueden experimentar un crecimiento rápido en la cantidad de contenido.
- **Facilidad de Uso:** MongoDB Atlas es una solución gestionada en la nube, lo que significa que te libera de las preocupaciones de mantenimiento y escalabilidad. Puedes concentrarte

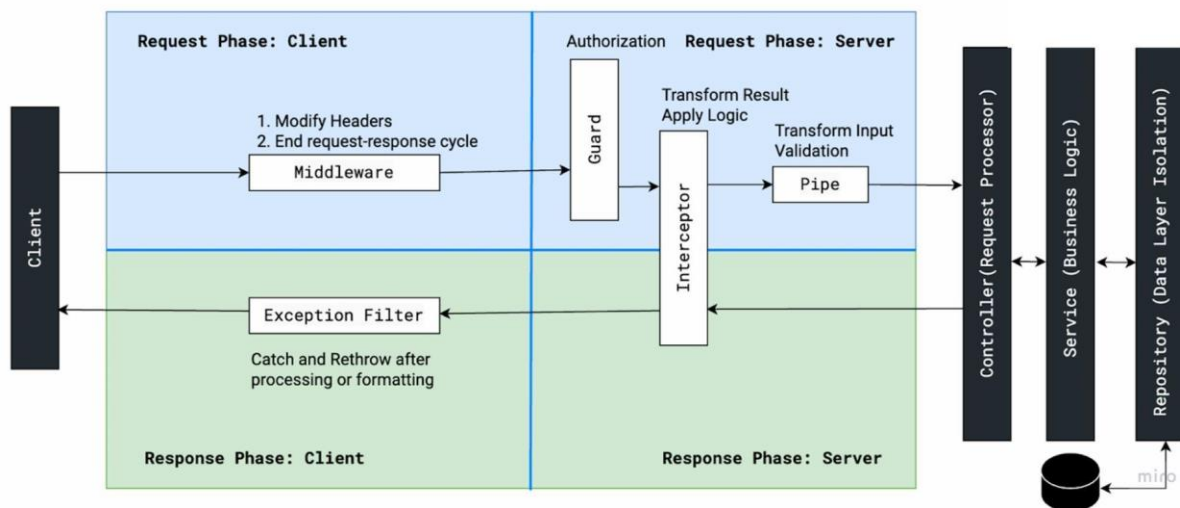
en el desarrollo del módulo de contenidos sin preocuparte demasiado por la administración de la base de datos.

Herramienta de Desarrollo: Postman

Pruebas de API: Postman es una herramienta popular para probar APIs, lo que será útil para garantizar que tu módulo de contenidos se integre correctamente con otras partes del software con sus respectivos CRUD.

Creación de la Lógica de Negocio

es donde aparece todo el backend desde cliente (esquema)



Desarrollo de Endpoints y APIs

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  POSTMAN CONSOLE  COMENTARIOS
jotiv@PCJ-LENOVO MINGW64 /d/Edit
$ cd server

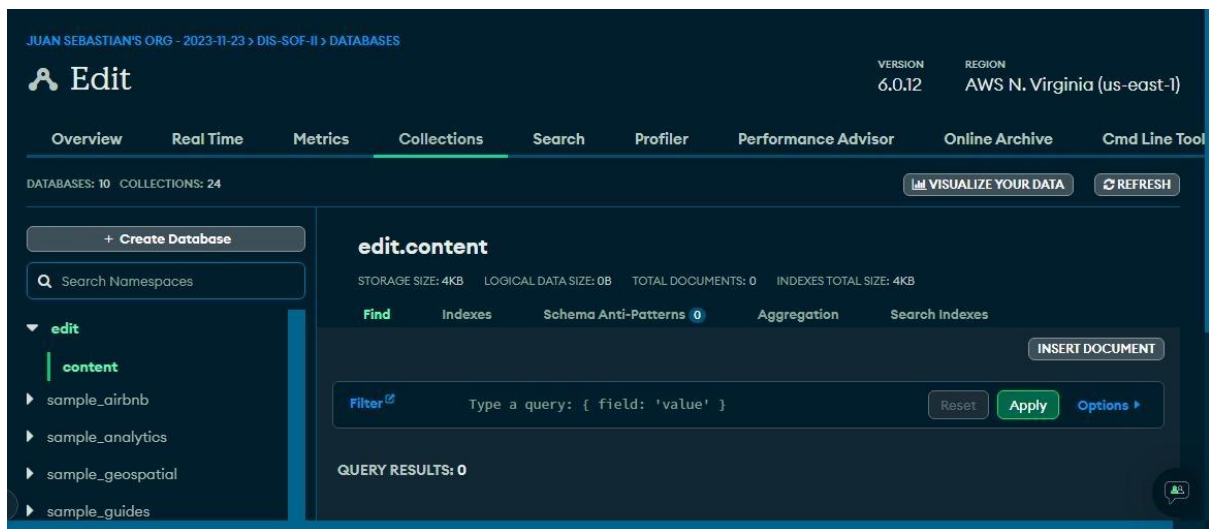
jotiv@PCJ-LENOVO MINGW64 /d/Edit/server (master)
[16:51:01] Starting compilation in watch mode...

[16:51:09] Found 0 errors. Watching for file changes.

[Nest] 14744 - 12/04/2023, 16:51:12 LOG [NestFactory] Starting Nest application...
[Nest] 14744 - 12/04/2023, 16:51:13 LOG [InstanceLoader] MongooseModule dependencies initialized +44ms
[Nest] 14744 - 12/04/2023, 16:51:13 LOG [InstanceLoader] AppModule dependencies initialized +1ms
[Nest] 14744 - 12/04/2023, 16:51:15 LOG [InstanceLoader] MongooseCoreModule dependencies initialized +2108ms
[Nest] 14744 - 12/04/2023, 16:51:15 LOG [RoutesResolver] AppController {/}: +90ms
[Nest] 14744 - 12/04/2023, 16:51:15 LOG [RouterExplorer] Mapped {/, GET} route +5ms
[Nest] 14744 - 12/04/2023, 16:51:15 LOG [NestApplication] Nest application successfully started +4ms
```

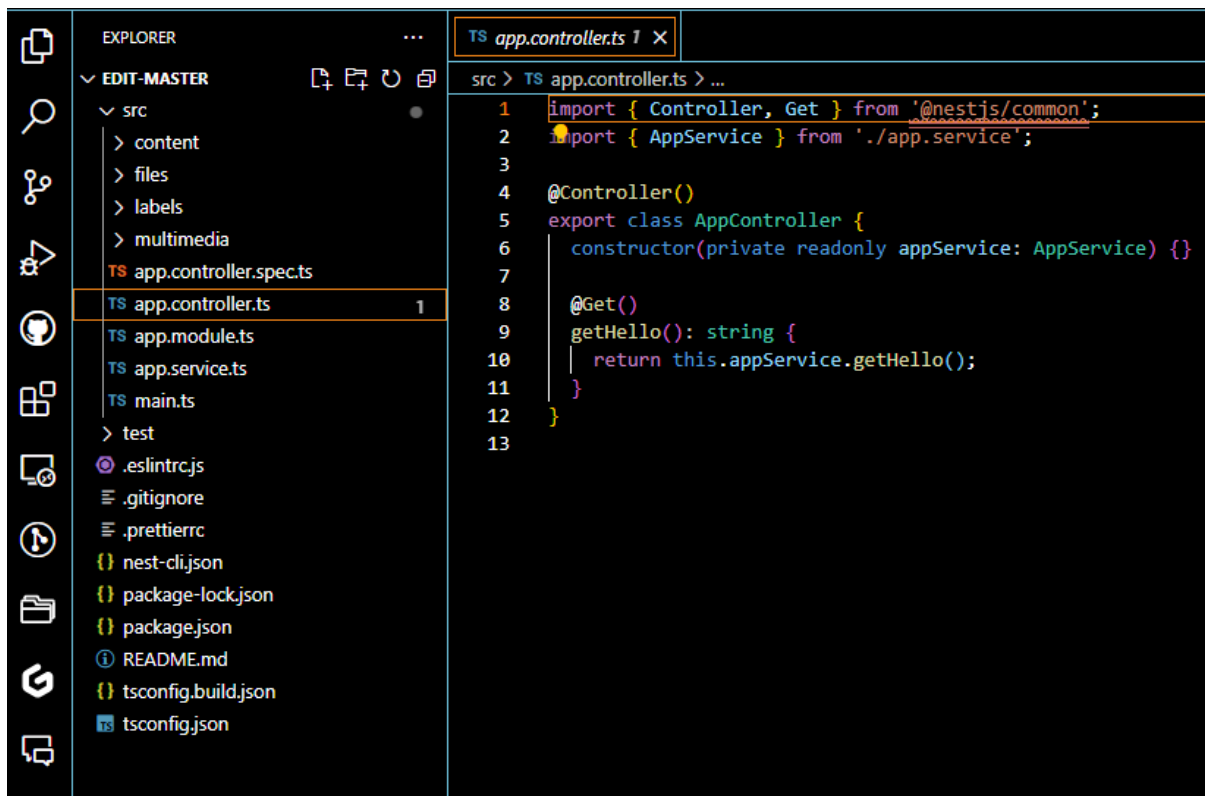
11. Conexión a la Base de Datos

Configuración de la Conexión

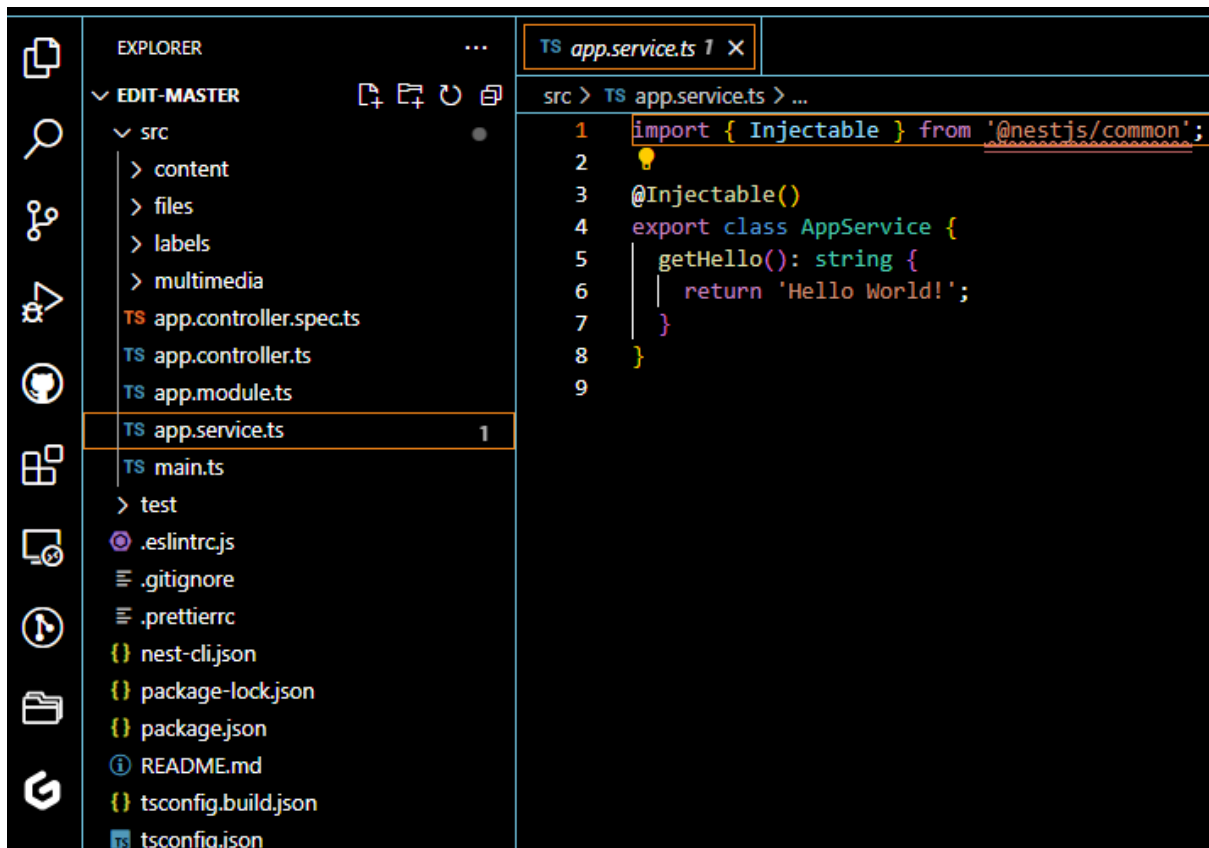


```
TS app.module.ts 2 X
src > TS app.module.ts > ...
1 import { Module } from '@nestjs/common';
2 import { AppController } from '../app.controller';
3 import { AppService } from '../app.service';
4 import { contentModule } from '../content/content.Module'; // Agrega esta línea
5 import { MongooseModule } from '@nestjs/mongoose';
6
7 @Module({
8   imports: [
9     MongooseModule.forRoot('mongodb+srv://jtiradoverbel06:Juan.tirado20@edit.hhfcfay.mongodb.net/?retryWrites=true&appName=nestjs-mongoose'),
10    //contentModule/,
11  ],
12   controllers: [AppController],
13   providers: [AppService],
14 })
15 export class AppModule {}
16
17
18
19
```

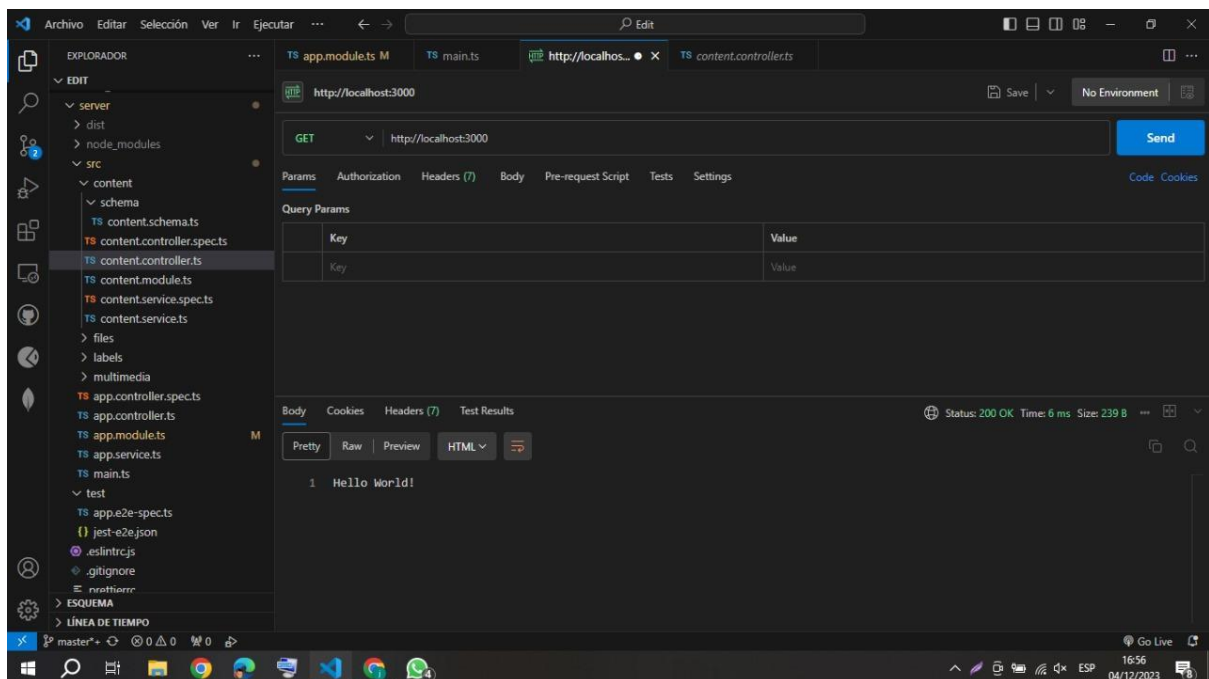
Desarrollo de Operaciones CRUD



```
TS app.controller.ts 1 X
src > TS app.controller.ts > ...
1 import { Controller, Get } from '@nestjs/common';
2 import { AppService } from '../app.service';
3
4 @Controller()
5 export class AppController {
6   constructor(private readonly appService: AppService) {}
7
8   @Get()
9   getHello(): string {
10     return this.appService.getHello();
11   }
12 }
13
```



Manejo de Transacciones



12. Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapas 3: Consumo de Datos y Desarrollo Frontend

Autores

Juan Sebastian Tirado Verbel, Jhonatan Salas De la rosa, Jesús Vargas soto, Andrés Diaz Vergara

13. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

14. Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

15. Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

16. Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

17. Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

18. Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

19. 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

20. Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend

ANEXOS

Diagramas UML

- **Diagrama de Casos de Uso (Use Case Diagram):** Este diagrama muestra las interacciones entre los actores (usuarios) y el sistema. Puede ayudar a identificar las funcionalidades clave y los actores involucrados.
- **Diagrama de Secuencia (Sequence Diagram):** Estos diagramas muestran la interacción entre objetos y actores a lo largo del tiempo. Puedes utilizarlos para representar cómo los usuarios interactúan con la pizarra en un flujo de trabajo específico.
- **Diagrama de Clases (Class Diagram):** Puedes utilizar este diagrama para modelar las clases y estructuras de datos subyacentes en el sistema, como usuarios, pizarras, comentarios, revisiones, etc.
- **Diagrama de Estados (State Diagram):** Este diagrama puede ser útil para modelar el comportamiento de la pizarra en diferentes estados, como "edición", "visualización", "comentario", etc.
- **Diagrama de Despliegue (Deployment Diagram):** Puedes utilizar este diagrama para representar cómo se despliega la aplicación en servidores y cómo interactúa con otros componentes del sistema, como el CMS.
- **Diagrama de Componentes (Component Diagram):** Este diagrama puede ayudar a representar la estructura de componentes del software, como la interfaz de usuario, la lógica de negocio, las bibliotecas y los servicios utilizados.

- **Diagrama de Actividad (Activity Diagram):** Puedes usar este diagrama para modelar flujos de trabajo o procesos específicos, como el flujo de trabajo de creación y edición de contenido en la pizarra.
- **Diagrama de Comunicación (Communication Diagram):** Similar a los diagramas de secuencia, estos diagramas muestran interacciones entre objetos y actores, pero pueden ser más simples y enfocados en la comunicación.
- **Diagrama de Paquetes (Package Diagram):** Este diagrama puede ayudar a organizar y visualizar los paquetes y módulos del software, lo que es útil para el diseño modular.
- **Diagrama de Objetos (Object Diagram):** Puedes utilizar este diagrama para representar instancias de clases y cómo interactúan en un escenario específico.

[\[k1\]](#) Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

[\[k2\]](#) Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.