



**UNIVERSIDAD DE
CÓRDOBA**



LICENCIATURA EN
INFORMÁTICA

Acreditada de Alta Calidad
MEN Res. 10710 25/05/17

Documento
técnico
para
proyectos
de Diseño
de
Software

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

Componente Gestión de Ova – Fase I

Keimer Enrique Muñoz Mora

Luis Carlos Suárez Bravo

José Isaac Chantak Zabaleta

Natalia Sofia Cochero Paternina

Tutor: Alexander Toscano



ENLACE DEL REPOSITORIO: https://github.com/Grupo-Investigacion-Bimadino/paul_ova_manager.git

Breve reseña

En muchos entornos educativos, los recursos educativos están dispersos en diferentes plataformas y formatos, lo que dificulta su acceso y gestión. El componente propuesto es un sistema integral diseñado para la recopilación, gestión y almacenamiento de Objetos Virtuales de Aprendizaje (OVA), abarcando una amplia variedad de formatos de contenidos. Ofrecerá una interfaz de usuario intuitiva y accesible desde múltiples dispositivos, junto con un sistema de búsqueda avanzada para facilitar la localización de recursos relevantes. Además, permitirá la personalización según las necesidades de los usuarios y organizaciones educativas, integrándose de manera fluida con sistemas de gestión del aprendizaje existentes y priorizando la seguridad de los datos mediante medidas como la encriptación y el control de acceso. El componente propuesto proporcionará una solución integral y escalable para la recopilación y gestión de OVA, ofreciendo una experiencia de aprendizaje enriquecida y adaptada a las necesidades individuales y organizativas.

ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS	6
INTRODUCCIÓN	6
PROPÓSITO DEL DOCUMENTO	6
ALCANCE DEL PROYECTO.....	7
DEFINICIONES Y ACRÓNIMOS	7
DESCRIPCIÓN GENERAL.....	7
OBJETIVOS DEL SISTEMA	11
FUNCIONALIDAD GENERAL	12
USUARIOS DEL SISTEMA	12
RESTRICCIONES	12
REQUISITOS FUNCIONALES	13
CASOS DE USO.....	13
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO.....	14
DIAGRAMAS DE FLUJO DE CASOS DE USO	14
PRIORIDAD DE REQUISITOS.....	26
REQUISITOS NO FUNCIONALES	32
REQUISITOS DE DESEMPEÑO.....	32
REQUISITOS DE SEGURIDAD.....	32
REQUISITOS DE USABILIDAD	32
REQUISITOS DE ESCALABILIDAD	32
MODELADO E/R.....	32
DIAGRAMA DE ENTIDAD-RELACIÓN	36
DESCRIPCIÓN DE ENTIDADES Y RELACIONES.....	34
REGLAS DE INTEGRIDAD.....	34
ANEXOS (SI ES NECESARIO).....	36
DIAGRAMAS ADICIONALES	41
REFERENCIAS	41

ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND	42
INTRODUCCIÓN	42
PROPÓSITO DE LA ETAPA	42
ALCANCE DE LA ETAPA	42
DEFINICIONES Y ACRÓNIMOS	42
DISEÑO DE LA ARQUITECTURA DE BACKEND.....	42
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA	42
COMPONENTES DEL BACKEND.....	42
DIAGRAMAS DE ARQUITECTURA	42
ELECCIÓN DE LA BASE DE DATOS	42
EVALUACIÓN DE OPCIONES (SQL o NoSQL).....	42
JUSTIFICACIÓN DE LA ELECCIÓN.....	42
DISEÑO DE ESQUEMA DE BASE DE DATOS	43
IMPLEMENTACIÓN DEL BACKEND	43
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN.....	43
CREACIÓN DE LA LÓGICA DE NEGOCIO	43
DESARROLLO DE ENDPOINTS Y APIS.....	43
AUTENTICACIÓN Y AUTORIZACIÓN	43
CONEXIÓN A LA BASE DE DATOS	43
CONFIGURACIÓN DE LA CONEXIÓN	43
DESARROLLO DE OPERACIONES CRUD.....	43
MANEJO DE TRANSACCIONES	43
PRUEBAS DEL BACKEND	43
DISEÑO DE CASOS DE PRUEBA.....	43
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN.....	44
MANEJO DE ERRORES Y EXCEPCIONES	44

ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND	45
INTRODUCCIÓN	45
PROPÓSITO DE LA ETAPA	45
ALCANCE DE LA ETAPA	45
DEFINICIONES Y ACRÓNIMOS	45
CREACIÓN DE LA INTERFAZ DE USUARIO (UI)	45
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS	45
CONSIDERACIONES DE USABILIDAD	45
MAQUETACIÓN RESPONSIVA	45
PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS).....	45
DESARROLLO DE LA LÓGICA DEL FRONTEND	45
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	45
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE)	46
CONSUMO DE DATOS DESDE EL BACKEND.....	46
CONFIGURACIÓN DE CONEXIONES AL BACKEND	46
OBTENCIÓN Y PRESENTACIÓN DE DATOS	46
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE)	46
INTERACCIÓN USUARIO-INTERFAZ	46
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS	46
IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS	46
MEJORAS EN LA EXPERIENCIA DEL USUARIO	46
PRUEBAS Y DEPURACIÓN DEL FRONTEND	46
DISEÑO DE CASOS DE PRUEBA DE FRONTEND	46
PRUEBAS DE USABILIDAD.....	46
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO.....	47
IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND	47
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO)	47
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND	47
INTEGRACIÓN CON EL BACKEND	47
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND	47
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND	47

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Introducción

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación de software de tipo educativo, comercial, OVA, componente o módulo de aplicaciones. Se divide en tres etapas para facilitar el entendimiento y aplicación a gran escala en la asignatura de diseño de software.

- Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Esta etapa cumple la tarea de recoger todas las competencias desarrolladas en todas las áreas de formación del currículo de la licenciatura en Informática y Medios Audiovisuales y ponerlas a prueba en el diseño y análisis de un producto educativo que se base en las teorías de aprendizaje estudiadas, articule las estrategias de enseñanza con uso de TIC y genere innovaciones en educación con productos interactivos que revelen una verdadera naturaleza educativa. Estos productos deben aprovechar las fortalezas adquiridas en las áreas de tecnología e informática, técnicas y herramientas, medios audiovisuales y programación y sistemas, para generar productos software interactivos que permitan a los usuarios disfrutar de lo que aprenden, a su propio ritmo. Todo esto en el marco de un proceso metodológico (metodologías de desarrollo de software como MODESEC, SEMLI, etc.) que aproveche lo aprendido en la línea de gestión y lo enriquezca con elementos de la Ingeniería de Software.

- Etapa 2: Persistencia de Datos con Backend – Servidor

En la etapa 2 se continua con los lineamientos de la etapa 1, para seguir adicionando elementos de diseño e implementación de software, enfocados en el desarrollo de APIs, servidores o microservicios que permitan soportar aplicaciones cliente del software educativo; en este sentido, el curso presenta los conceptos de los sistemas de bases de datos, su diseño lógico, la organización de los sistemas manejadores de bases de datos, los lenguaje de definición de datos y el lenguaje de manipulación de datos SQL y NoSQL; de tal manera que los estudiantes adquieran las competencias para analizar, diseñar y desarrollar aplicaciones para gestionar y almacenar grandes cantidades de datos, mediante el uso de técnicas adecuadas como el diseño y modelo lógico y físico de base datos, manejo de los sistemas de gestión de bases de datos, algebra relacional, dominio del lenguaje SQL como herramienta de consulta, tecnología cliente / servidor; igualmente, se definirán los elementos necesarios para el acceso a dichas bases de datos, como la creación del servidor API, utilizando tecnologías de vanguardia como node.js, express, Nest.js, Spring entre otros; para, finalmente converger en el despliegue de la API utilizando servicios de hospedaje en la nube, preferiblemente gratuitos. También podrá implementar servidores o API's con inteligencia artificial o en su defecto crear una nueva capa que consuma y transforme los datos obtenidos de la IA. El desarrollo del curso se trabajara por proyectos de trabajo colaborativo que serán evaluados de múltiples maneras, teniendo en cuenta más el proceso que el resultado.

- Etapa 3: Consumo de Datos y Desarrollo Frontend – Cliente

La etapa 3 el estudiante está en capacidad de establecer la mejor elección de herramientas de consumo de datos y técnicas en aras de lograr el mejor producto a nivel de software o hardware acorde a los requerimientos funcionales y no funcionales del problema a solucionar. En este punto el estudiante puede consumir los datos a través de un cliente que puede ser una aplicación de celular, una aplicación de escritorio, una página web, IoT(internet de las cosas) o incluso, artefactos tecnológicos. El diseño gráfico es de los requisitos esenciales en la capa de presentación, por lo tanto, se requieren los cursos de diseño gráfico vistos previamente. Los elementos anteriores nos permiten elegir el paradigma y tecnología para desarrollar nuestras aplicaciones, teniendo en cuenta que podríamos desarrollar aplicaciones de tipo cliente.

Alcance del Proyecto

El componente propuesto es un sistema integral diseñado específicamente para la gestión eficiente de Objetos Virtuales de Aprendizaje (OVA). Este sistema permite administrar los OVA teniendo en cuenta sus características principales, como la página de inicio, los contenidos y la evaluación. Además, incluye una interfaz intuitiva con búsqueda avanzada, segura y se integra fácilmente con los sistemas de gestión del aprendizaje existente. En el siguiente apartado, se detallarán las características de la presente versión y se sugerirán algunas para futuras actualizaciones.

- Subir un OVA.
- Visualizar un OVA.
- Actualizar un OVA.
- Eliminar un OVA.
- Buscar un OVA.
- Exportar OVA.
- Valorar OVA.

Funcionalidades Futuras:

- Interoperabilidad con Herramientas de Autoría
- Importar OVA externo.
- Recomendación Personalizada de Contenidos.
- Recolección de datos XAPI.
- Análisis de datos de Uso con XAPI.

Definiciones y Acrónimos

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

Es un conjunto de reglas y protocolos, permite que diferentes aplicaciones se comuniquen entre sí y compartan datos o funcionalidades.

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

Es un software que facilita la creación, manipulación y administración de bases de datos, permitiendo almacenar, organizar y recuperar información de manera eficiente.

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

Es un lenguaje estándar utilizado para interactuar con bases de datos relacionales. Permite realizar consultas, inserciones, actualizaciones y eliminaciones de datos, así como la creación y modificación de esquemas de base de datos.

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

Es un protocolo de comunicación utilizado para la transferencia de información en la World Wide Web. Define cómo se transmiten los mensajes y cómo navegadores y servidores web interactúan para solicitar y enviar recursos, páginas web, imágenes y otros archivos.

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

Es un estilo arquitectónico para el diseño de sistemas de software distribuido, basado en la representación de recursos a través de URLs y la manipulación de dichos recursos utilizando operaciones estándar de HTTP, como GET, POST, PUT y DELETE.

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

Es un formato de intercambio de datos ligero y legible por humanos, basado en la sintaxis de los objetos de JavaScript. Se utiliza comúnmente para transmitir datos estructurados entre un servidor y un cliente en aplicaciones web y es independiente del lenguaje de programación.

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

Es un estándar abierto (RFC 7519) que define un formato compacto y seguro para la transferencia de información entre dos partes, generalmente utilizado en autenticación y autorización en aplicaciones web. Los JWT están representados como cadenas JSON y son firmados digitalmente para verificar su autenticidad y asegurar la integridad de los datos.

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

Es un conjunto de operaciones básicas utilizadas en sistemas de gestión de bases de datos y aplicaciones web para manipular datos. Estas operaciones son fundamentales para el mantenimiento de la información: crear nuevos registros (Create), leer datos existentes (Read), actualizar registros existentes (Update) y eliminar registros (Delete).

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

Es una técnica de programación que permite convertir datos entre el modelo de objetos utilizado en un lenguaje de programación orientado a objetos y el modelo relacional utilizado en una base de datos relacional.

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

Es un patrón de diseño de software que divide una aplicación en tres componentes principales: el Modelo, que representa los datos y la lógica de la aplicación; la Vista, que se encarga de mostrar la interfaz de usuario y recibir las entradas del usuario; y el Controlador, que actúa como intermediario

entre el Modelo y la Vista, gestionando las interacciones del usuario y actualizando el Modelo en consecuencia.

API RESTful: API que sigue los principios de REST.

API que sigue los principios de REST, utilizando URLs para identificar recursos y métodos HTTP estándar para manipularlos. Es flexible y fácilmente entendida por los desarrolladores.

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

Es un enfoque de desarrollo que automatiza los procesos de construcción, pruebas y despliegue de código para entregar cambios de manera rápida y segura. La Integración Continua se concentra en fusionar y probar código frecuentemente, mientras que la Entrega Continua automatiza el despliegue regular y confiable del código en entornos de producción.

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

Es un modelo de distribución de software donde las aplicaciones son alojadas por un proveedor de servicios y accesibles a través de internet, generalmente mediante suscripciones.

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

Son protocolos de seguridad que cifran las comunicaciones en internet para proteger la privacidad y la integridad de los datos transmitidos entre clientes y servidores.

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

Es el lenguaje estándar utilizado para crear páginas web, permitiendo la estructuración y presentación de contenido en línea.

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

Es un lenguaje utilizado para definir el estilo y la presentación de documentos HTML, permitiendo controlar el diseño, la tipografía, los colores y otros aspectos visuales de una página web.

JS: JavaScript.

Es un lenguaje de programación usado para agregar interactividad y funcionalidad dinámica a páginas web.

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

Es una interfaz de programación que representa la estructura de un documento HTML como un árbol de objetos, permitiendo a los programas acceder y manipular el contenido, la estructura y el estilo de una página web de manera dinámico.

UI: Interfaz de Usuario (User Interface).

Es el medio por el que los usuarios interactúan con un dispositivo, aplicación o sistema informático, permitiéndoles actuar y recibir retroalimentación de forma intuitiva y eficiente.

UX: Experiencia del Usuario (User Experience).

Es la impresión general que tiene un usuario al interactuar con un producto o servicio, incluyendo aspectos como la usabilidad, la accesibilidad y la satisfacción.

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

Es una aplicación web que carga una sola página HTML y actualiza el contenido de manera dinámica utilizando JavaScript, proporcionando una experiencia de usuario fluida y rápida.

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

Es una técnica de desarrollo web que permite realizar solicitudes al servidor de forma asíncrona, sin necesidad de recargar la página completa, lo que mejora la velocidad y la interactividad de las aplicaciones web.

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

Es una plataforma que facilita la creación, edición, gestión y publicación de contenido digital, como páginas web, blogs o tiendas en línea, sin necesidad de conocimientos técnicos avanzados.

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

Es una red de servidores distribuidos geográficamente que almacenan copias de contenido web estático, como imágenes, archivos de estilo y scripts, para entregarlo a los usuarios de manera más rápida y eficiente.

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

Es el proceso de mejorar la visibilidad y la clasificación de un sitio web en los resultados de búsqueda orgánica, mediante técnicas como la optimización del contenido, la estructura del sitio y la construcción de enlaces.

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

Es un software que proporciona herramientas integradas para escribir, depurar y compilar código de programación en un solo lugar, facilitando el desarrollo de software.

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

Es una interfaz de usuario que permite interactuar con un sistema informático mediante comandos de texto, en lugar de utilizar una interfaz gráfica.

PWA: Aplicación Web Progresiva (Progressive Web App).

Es una aplicación web que utiliza tecnologías modernas para proporcionar una experiencia similar a la de una aplicación nativa en dispositivos móviles, incluyendo funciones como el acceso offline, las notificaciones push y la instalación en la pantalla de inicio.

OVA: Objetos Virtuales de Aprendizaje (Virtual Learning Objects)

Un OVA (Objeto Virtual de Aprendizaje), es un recurso educativo interactivo en línea, que facilita el aprendizaje. Está compuesto por elementos como texto, imágenes y videos. Puede distribuirse con documentos HTML para fácil acceso en navegadores web estándar.

XAPI: Experiencia de Aprendizaje Electrónico (Experience API)

Es un estándar de tecnología de aprendizaje electrónico que permite el seguimiento y la recopilación de datos sobre las actividades en línea.

GESTIÓN DEL OVA: (OVA Management)

La gestión del OVA se refiere al conjunto de procesos y estrategias utilizados para administrar y evaluar los recursos educativos digitales conocidos como Objetos Virtuales de Aprendizaje. Esto incluye la creación, distribución, actualización, y seguimiento de los OVAs para garantizar su efectividad en el proceso de enseñanza y aprendizaje.

VALORACIÓN DE OVA: (OVA Assessment)

La valoración de un OVA consiste en asignar una puntuación entre 1 y 5 para evaluar su calidad, efectividad y relevancia en el contexto educativo. Cuanto mayor sea la valoración, mayor será la probabilidad de que el OVA sea recomendado en un entorno de sugerencias.

Descripción General

Objetivos del Sistema

El sistema tiene como objetivos principales facilitar la recopilación, gestión y almacenamiento de Objetos Virtuales de Aprendizaje (OVA) desde diversas fuentes y formatos, así como proporcionar una interfaz de usuario intuitiva y accesible para una fácil navegación y recuperación de recursos educativos. Además, busca permitir la personalización y adaptabilidad de los OVA según las necesidades específicas de los usuarios y las organizaciones educativas, integrarse fluidamente con sistemas de gestión del aprendizaje existentes y priorizar la seguridad y privacidad de los datos del usuario y los contenidos educativos almacenados en el sistema.

Funcionalidad General

Subir un OVA: Permite a los usuarios cargar nuevos Objetos Virtuales de Aprendizaje al sistema. Esta funcionalidad admite múltiples formatos de archivo y ofrece una interfaz amigable que guía al usuario a través del proceso de carga, asegurando que todos los elementos del OVA, como la página de inicio, los contenidos y las evaluaciones, se registren correctamente.

Visualizar un OVA: Facilita la visualización de los OVA dentro del sistema. Los usuarios pueden navegar por los contenidos de manera intuitiva, accediendo fácilmente a todas las secciones del OVA, como la introducción, los módulos de contenido y las evaluaciones. Esta funcionalidad asegura una experiencia de usuario fluida y coherente.

Actualizar un OVA: Permite a los usuarios modificar o actualizar el contenido de un OVA existente. Los cambios pueden incluir la actualización de textos, imágenes, videos y otros recursos multimedia, así como ajustes en las evaluaciones. Esta funcionalidad garantiza que los OVA se mantengan actuales y relevantes.

Eliminar un OVA: Ofrece a los usuarios la capacidad de eliminar OVA que ya no son necesarios o relevantes. Esta funcionalidad incluye medidas de seguridad, como confirmaciones de eliminación y la posibilidad de recuperar OVA eliminados recientemente, para prevenir la pérdida accidental de información.

Buscar un OVA: Proporciona herramientas de búsqueda avanzada que permiten a los usuarios encontrar OVA específicos de manera rápida y eficiente. Los criterios de búsqueda pueden incluir palabras clave, etiquetas, fechas de creación, y otros metadatos asociados con los OVA, asegurando que los usuarios puedan localizar fácilmente el contenido que necesitan.

Exportar OVA: Permite a los usuarios exportar OVA en varios formatos, como PDF, SCORM, y otros estándares compatibles con sistemas de gestión del aprendizaje. Esta funcionalidad facilita la distribución y el uso de los OVA en diferentes plataformas y entornos de aprendizaje.

Valorar OVA: Facilita la evaluación de los OVA por parte de los usuarios. Los usuarios pueden asignar calificaciones y proporcionar comentarios sobre la calidad y la utilidad de los OVA. Esta retroalimentación se utiliza para mejorar continuamente los contenidos y la experiencia de aprendizaje.

Usuarios del Sistema

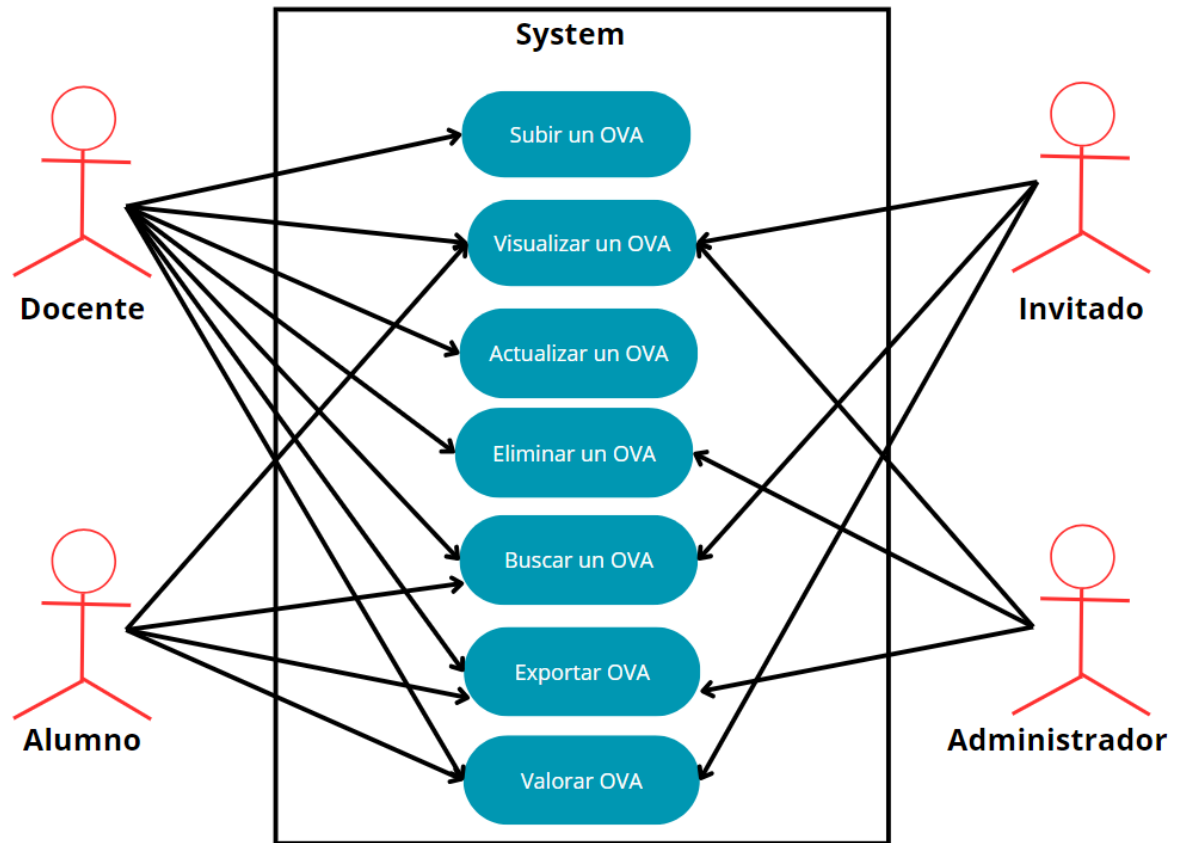
Los siguientes usuarios pueden interactuar con la pizarra dependiendo de las funcionalidades.

Funcionalidad	Administradores	Docente	Alumno	Invitado
Subir OVA		✓		
Visualizar OVA	✓	✓	✓	✓
Actualizar OVA		✓		
Eliminar OVA	✓	✓		
Buscar OVA		✓	✓	✓
Exportar OVA	✓	✓	✓	
Valorar OVA		✓	✓	✓

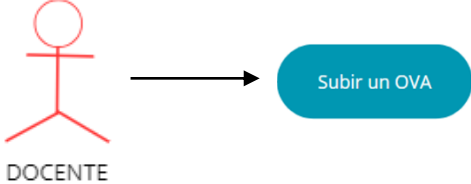
Restricciones

Requisitos Funcionales

Casos de Uso



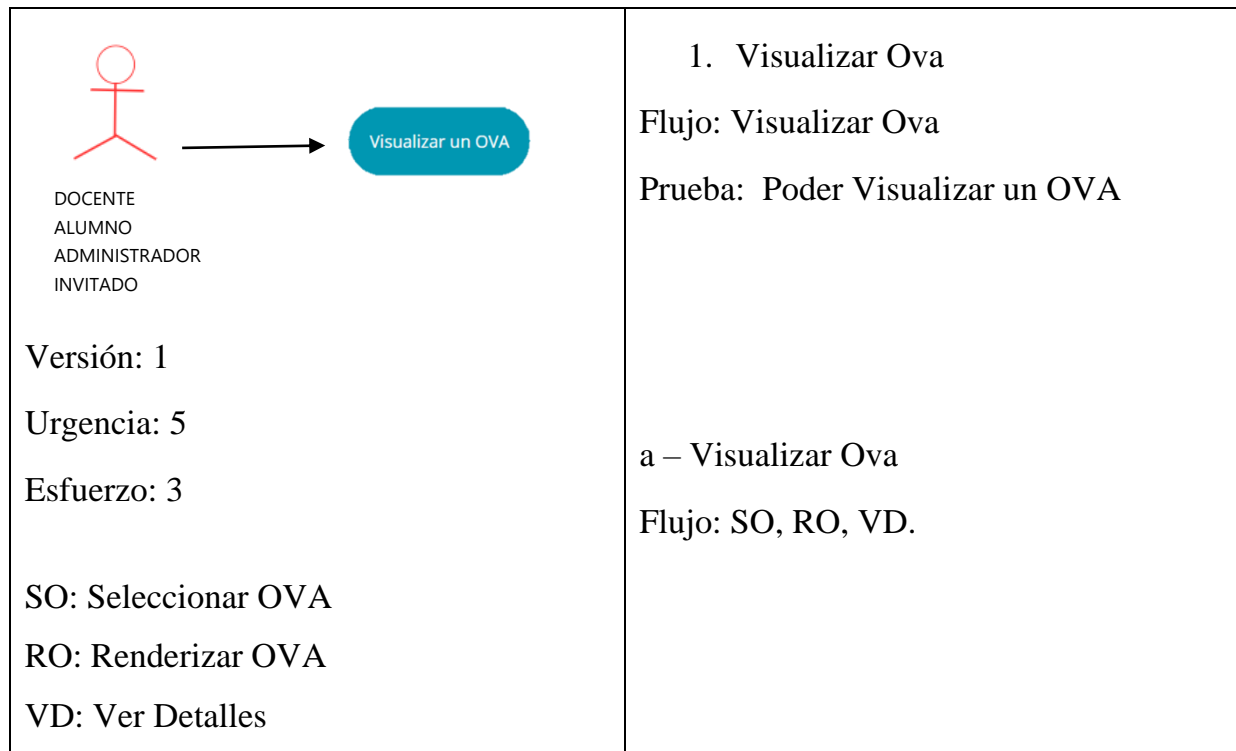
CASO No. 1 Subir Ova

 <p>Versión: 1</p> <p>Urgencia: 5</p> <p>Esfuerzo: 5</p> <p>SO: Subir Ova</p> <p>SA: Seleccionar Archivo</p> <p>AN: Asignar Nombre</p> <p>DO: Descripción Ova</p> <p>CA: Colocar Autor</p> <p>GO: Guardar Ova</p>	<p>1. Subir Ova</p> <p>Flujo: Subir Ova</p> <p>Prueba: Subir Ova, Seleccionar Ova, Cargar Ova, asignar descripción, y guardar Ova.</p> <p>a – Subir Ova</p> <p>Flujo: SO, SA, AN, CO, CA</p>
--	--

ID:	CDU-1
Nombre	Subir Ova
Actores	Docente
Objetivo	Permitir cargar el Ova
Urgencia	5
Esfuerzo	5

Precondiciones	Debe haber opciones para subir el Ova	
Flujo normal	Docente	Sistema
	Selecciona subir un Ova	
		Retorna opciones de subir Ova
	Selecciona Ova a Subir	
		Carga el Ova
	Llena los campos necesarios (descripción) del Ova	
		Registra la subida del Ova
		Mensaje Ova Guardado con Éxito
Flujo alternativo 1	Selecciona subir un Ova	
		Retorna opciones de subir Ova
	Selecciona Ova a Subir	
		Carga el Ova
	Llena los campos necesarios (descripción) del Ova	
		No guarda la actividad por falta de conexión
	.	Retorna un borrador del Ova

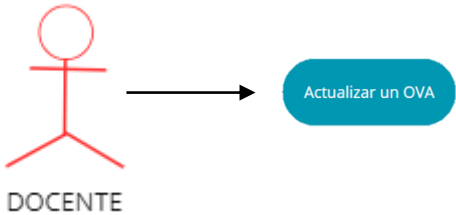
CASO No. 2 Visualizar Ova



ID:	CDU-2	
Nombre	Visualizar Ova	
Actores	Docente, Alumno, Administrador, Invitado.	
Objetivo	Permitir visualizar el Ova	
Urgencia	5	
Esfuerzo	3	
Precondiciones	El debe estar disponible en el sistema	
Flujo normal	Docente	Sistema
	Selecciona visualizar un Ova	
		Muestra lista de Ovas disponibles
	Selecciona el Ova deseado	
		Carga y muestra el contenido del Ova
	Selecciona Ver Detalles	
		Muestra los detalles del OVA

Flujo alternativo 1	Selecciona visualizar Ova	
		Muestra mensaje de error si no hay conexión
	Usuario reintenta la acción	

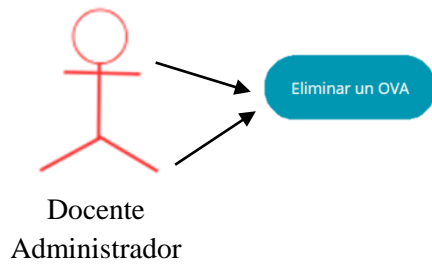
CASO No. 3 Actualizar Ova

 <p>DOCENTE</p> <p>Versión: 1 Urgencia: 4 Esfuerzo: 4</p> <p>AO: Actualizar OVA SO: Seleccionar OVA AC: Actualizar Campos GC: Guardar Cambios</p>	<p>1. Actualizar Ova</p> <p>Flujo: Actualizar OVA</p> <p>Prueba: Cargar Actualización</p> <p>a – Actualizar Ova</p> <p>Flujo: AO, SO, AC, GC.</p>
--	--

ID:	CDU-3	
Nombre	Actualizar OVA	
Actores	Docente	
Objetivo	Permitir actualizar la información del OVA	
Urgencia	4	
Esfuerzo	4	
Precondiciones	El OVA debe existir en el sistema	
Flujo normal	Docente	Sistema
	Actualizar OVA	

		Muestra lista de OVA disponibles
	Selecciona el OVA a actualizar	
		Carga la información actual del OVA
	Modifica la información necesaria	
		Guarda los cambios realizados
		Mensaje de OVA actualizado con Éxito.
Flujo alternativo 1	Actualizar OVA	
		Muestra mensaje de error si no hay conexión
	Docente reintenta la acción	

CASO No. 4 Eliminar Ova



Versión: 1

Urgencia: 3

Esfuerzo: 3

SO: Seleccionar OVA

SE: Seleccionar Eliminar

EO: Eliminar Ova

1. Eliminar Ova

Flujo: Eliminar Ova

Prueba: Borrar Ova de la pantalla de visualización

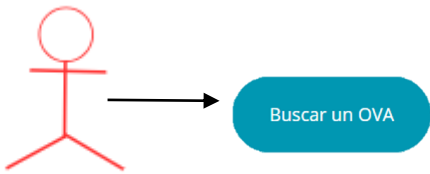
a – Eliminar Ova

Flujo: SO, SE, EO.

ID:	CDU-4	
Nombre	Eliminar Ova	
Actores	Docente, Administrador	
Objetivo	Permitir eliminar un OVA del sistema	
Urgencia	3	
Esfuerzo	3	
Precondiciones	El OVA debe existir en el sistema	
Flujo normal	Docente	Sistema
	Selecciona eliminar OVA	
		Muestra lista de Ovas disponibles
	Selecciona el Ova a eliminar	

		Solicita confirmación de eliminación
	Docente confirma eliminación	
		Confirma la acción
		Mensaje de OVA eliminado correctamente
Flujo alternativo 1	Selecciona eliminar Ova	
		Muestra la lista de Ovas disponible
	Selecciona el Ova a eliminar	
		Muestra mensaje de confirmación
	Docente presiona cancelar	
		Cancela la acción
		Mensaje de OVA no eliminado

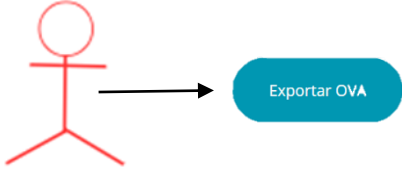
CASO No. 5 Buscar un Ova

 <p>DOCENTE ALUMNO INVITADO</p> <p>Versión: 1 Urgencia: 4 Esfuerzo: 3</p> <p>CB: Criterios de Búsqueda</p>	<p>1. Buscar OVA</p> <p>Flujo: Buscar OVA</p> <p>Prueba: Se debe poder buscar OVA relacionado al Criterio que se introdujo</p> <p>a – Buscar Ova</p> <p>Flujo: CB, BO.</p>
---	--

BO: Buscar OVA	
----------------	--

ID:	CDU-5	
Nombre	Buscar OVA	
Actores	Docente, Estudiante, Invitado	
Objetivo	Permitir buscar OVAs en el sistema	
Urgencia	4	
Esfuerzo	3	
Precondiciones	El usuario debe estar autenticado	
Flujo normal	Usuario	Sistema
	Usuario ingresa criterios de búsqueda	
		Retorna resultados de la búsqueda
Flujo alternativo 1	Usuario ingresa criterios de búsqueda	
		Mensaje de error si no hay conexión

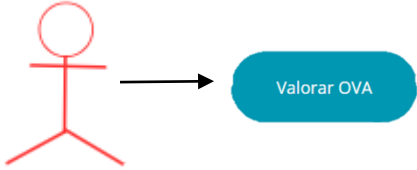
CASO No. 6 Exportar Ova

 <p>DOCENTE ALUMNO ADMINISTRADOR</p> <p>Versión: 1 Urgencia: 3 Esfuerzo: 4</p> <p>EO: Exportar Ova</p>	<p>1. Exportar Ova</p> <p>Flujo: Exportar Ova</p> <p>Prueba: Exporta un OVA a un formato descargable</p> <p>a – Exportar Ova</p> <p>Flujo: EO</p>
---	---

ID:	CDU-6	
Nombre	Exportar Ova	
Actores	Docente	
Objetivo	Permitir exportar un OVA a un formato descargable	
Urgencia	3	
Esfuerzo	4	
Precondiciones	El OVA debe existir en el sistema	
Flujo normal	Docente	Sistema
	Selecciona Exportar OVA	

		Muestra lista de Ovas disponibles
	Selecciona el Ova a exportar	
		Muestra opciones de formato de exportación
	Selecciona el formato deseado	
		Procesa y genera el archivo exportado
		Mensaje OVA exportado correctamente
Flujo alternativo 1	Selecciona Exportar OVA	
		Muestra mensaje de error si no hay conexión
	Docente reintenta la acción	

CASO No. 7 Valorar Ova

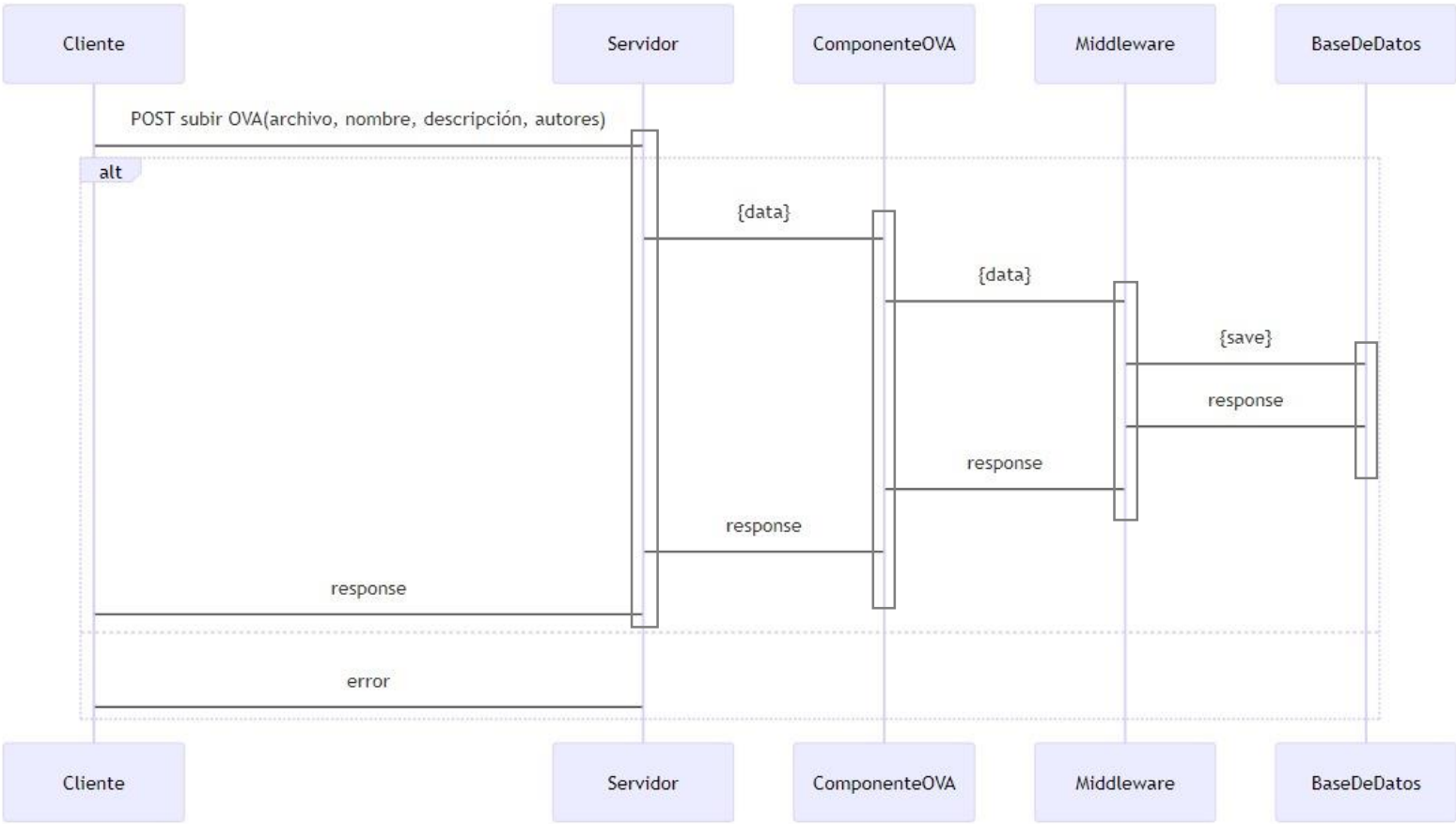
 <p>DOCENTE</p> <p>ALUMNO</p> <p>INVITADO</p> <p>Versión: 1</p> <p>Urgencia: 2</p> <p>Esfuerzo: 2</p> <p>VO: Valorar Ova</p>	<p>1. Valorar Ova</p> <p>Flujo: Valorar Ova</p> <p>Prueba: Puntuar ova de acuerdo con la valoración que se decida.</p> <p>a – Valorar Ova</p> <p>Flujo: VO</p>
---	--

ID:	CDU-7	
Nombre	Valorar Ova	
Actores	Docente, Estudiante, Invitado	
Objetivo	Permitir valorar un OVA dentro del sistema	
Urgencia	2	
Esfuerzo	2	
Precondiciones	El OVA debe estar disponible en el sistema	
Flujo normal	Docente	Sistema
	Usuario selecciona el OVA a valorar	
		Muestra opciones de valoración (estrellas, comentarios, etc.)
	Usuario selecciona una valoración y/o escribe un comentario	
		Guarda la valoración

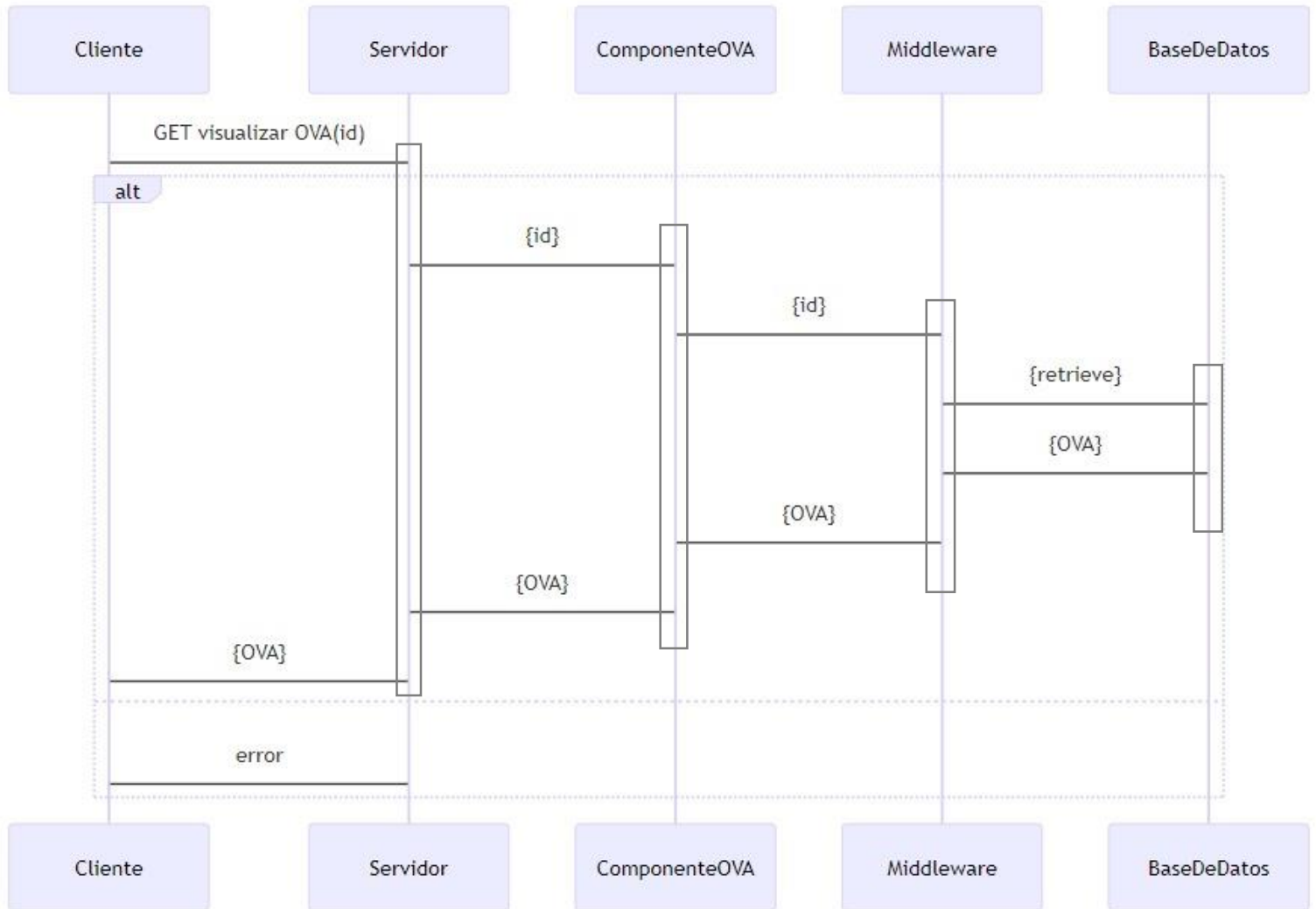
		Mensaje de valoración guardada correctamente
Flujo alternativo 1	Usuario selecciona el OVA a valorar	
		Muestra mensaje de error si no hay conexión
	Usuario reintenta la acción	

Diagramas de Secuencia

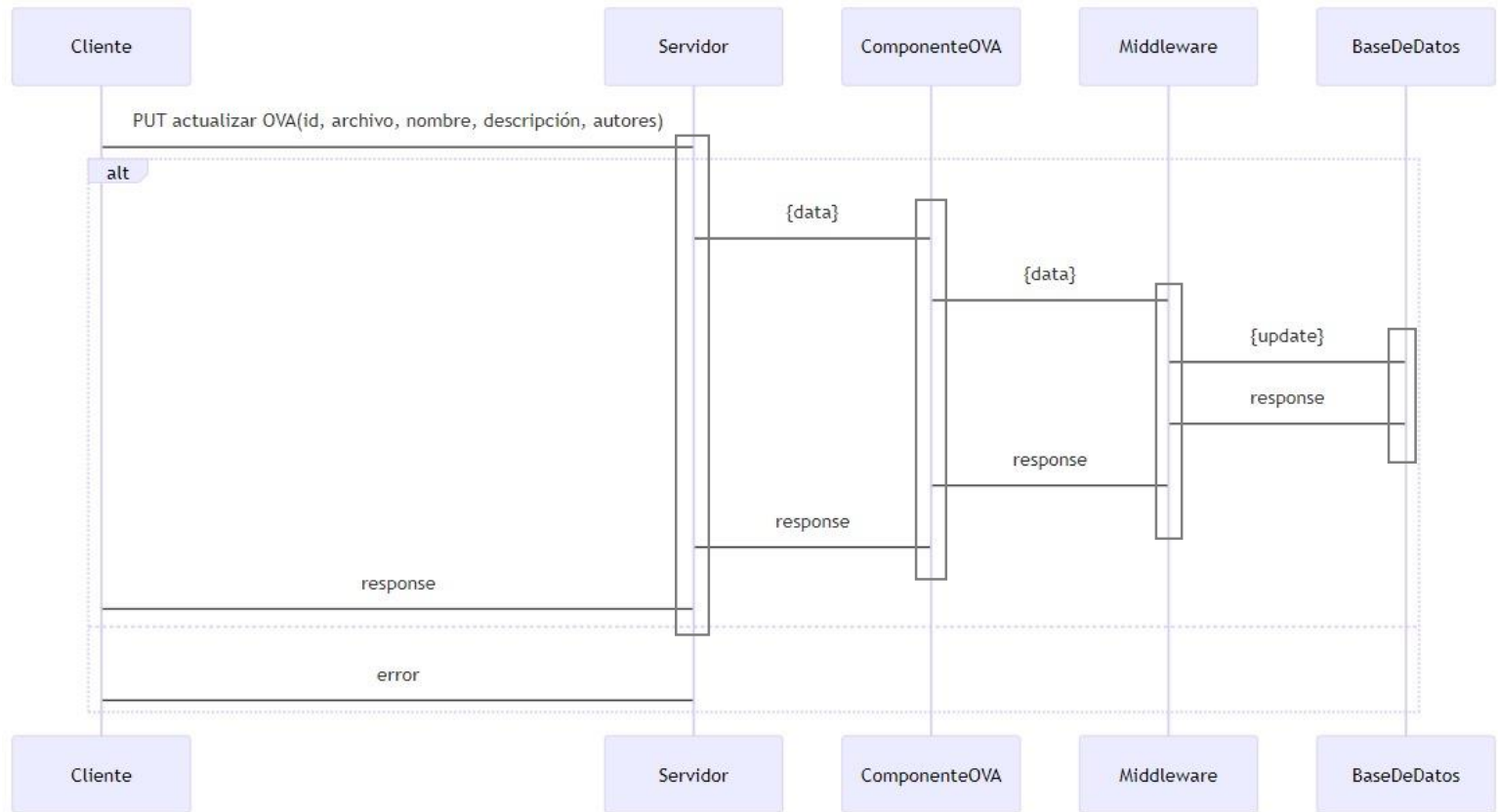
Subir OVA



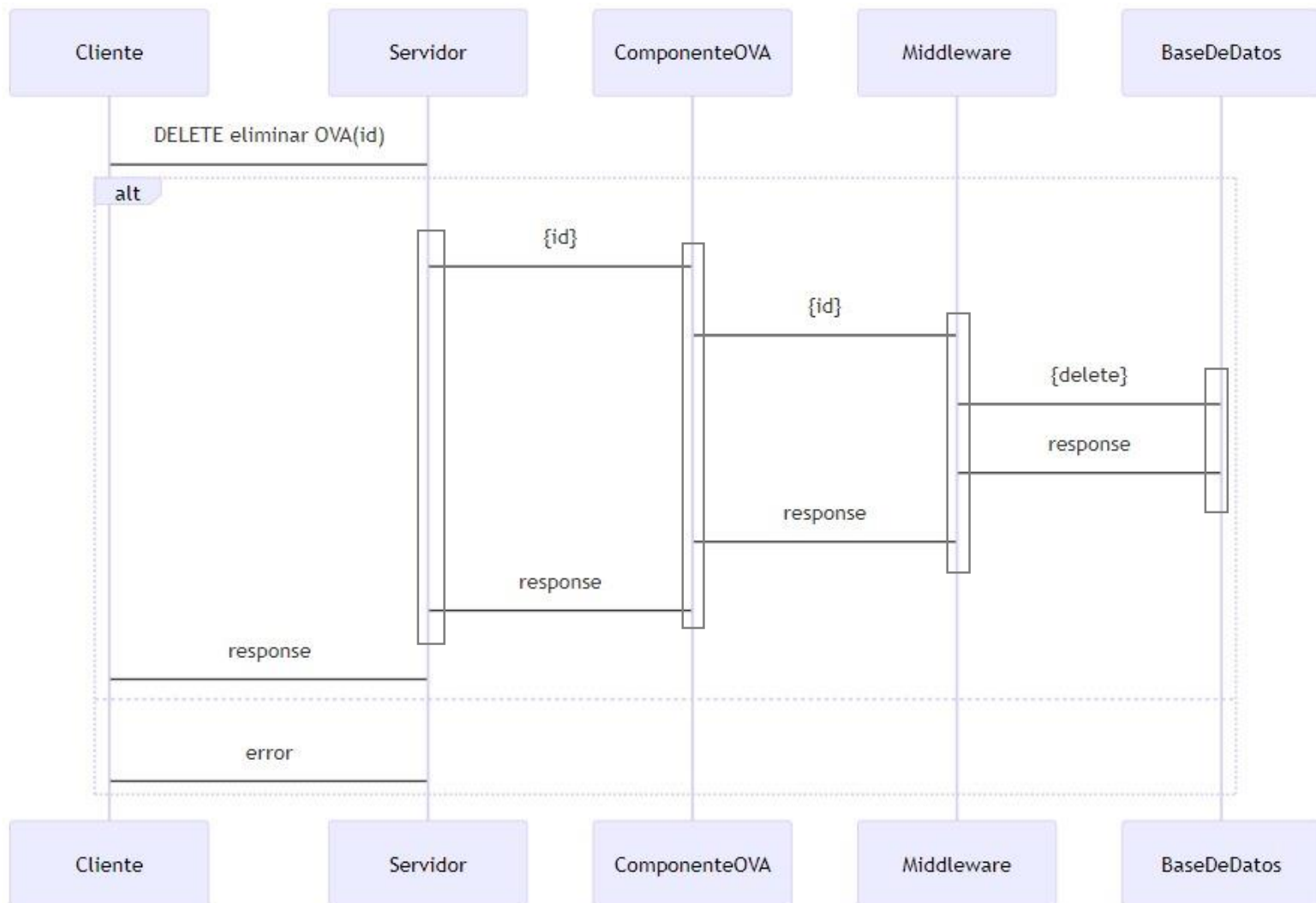
Visualizar OVA



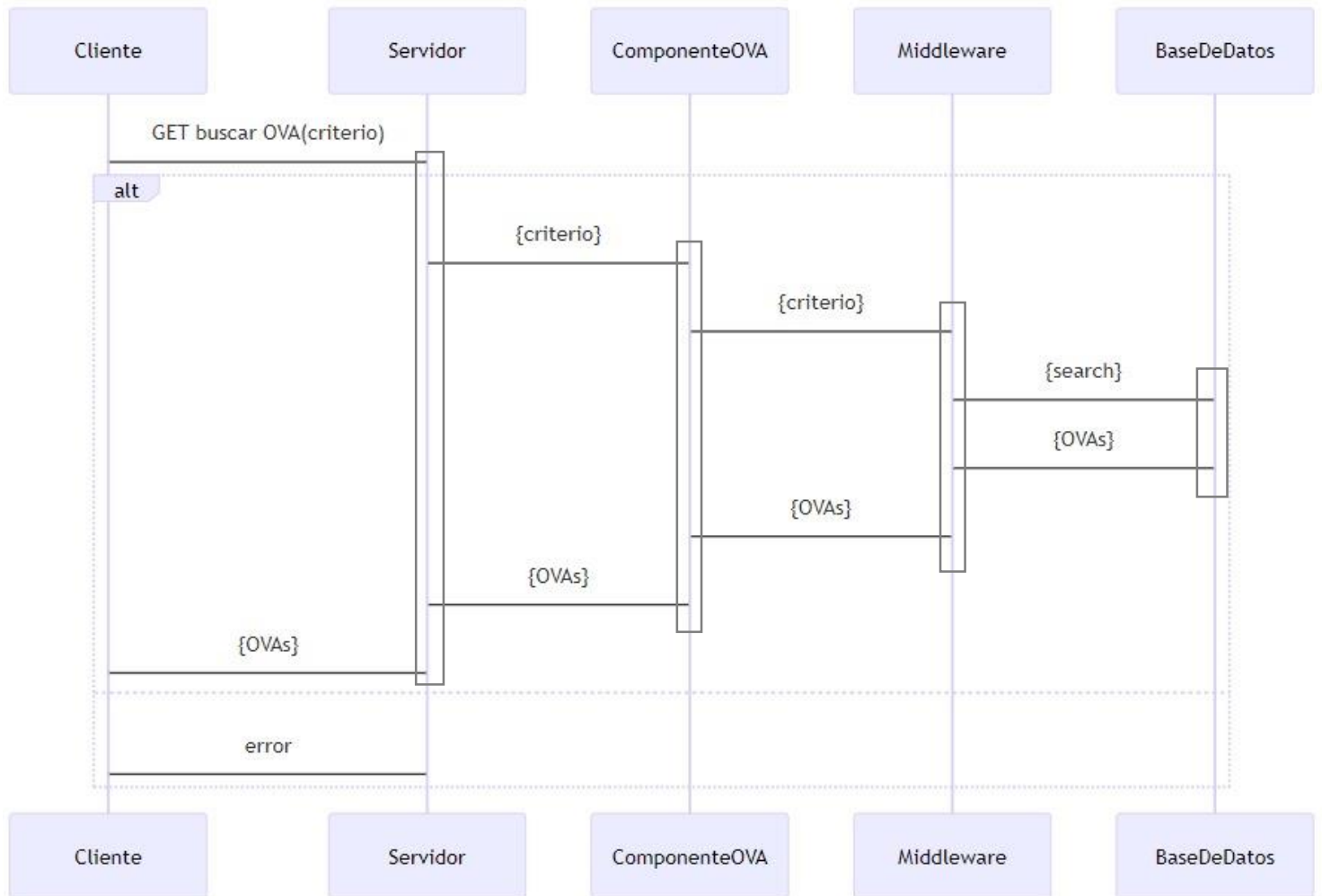
Actualizar OVA



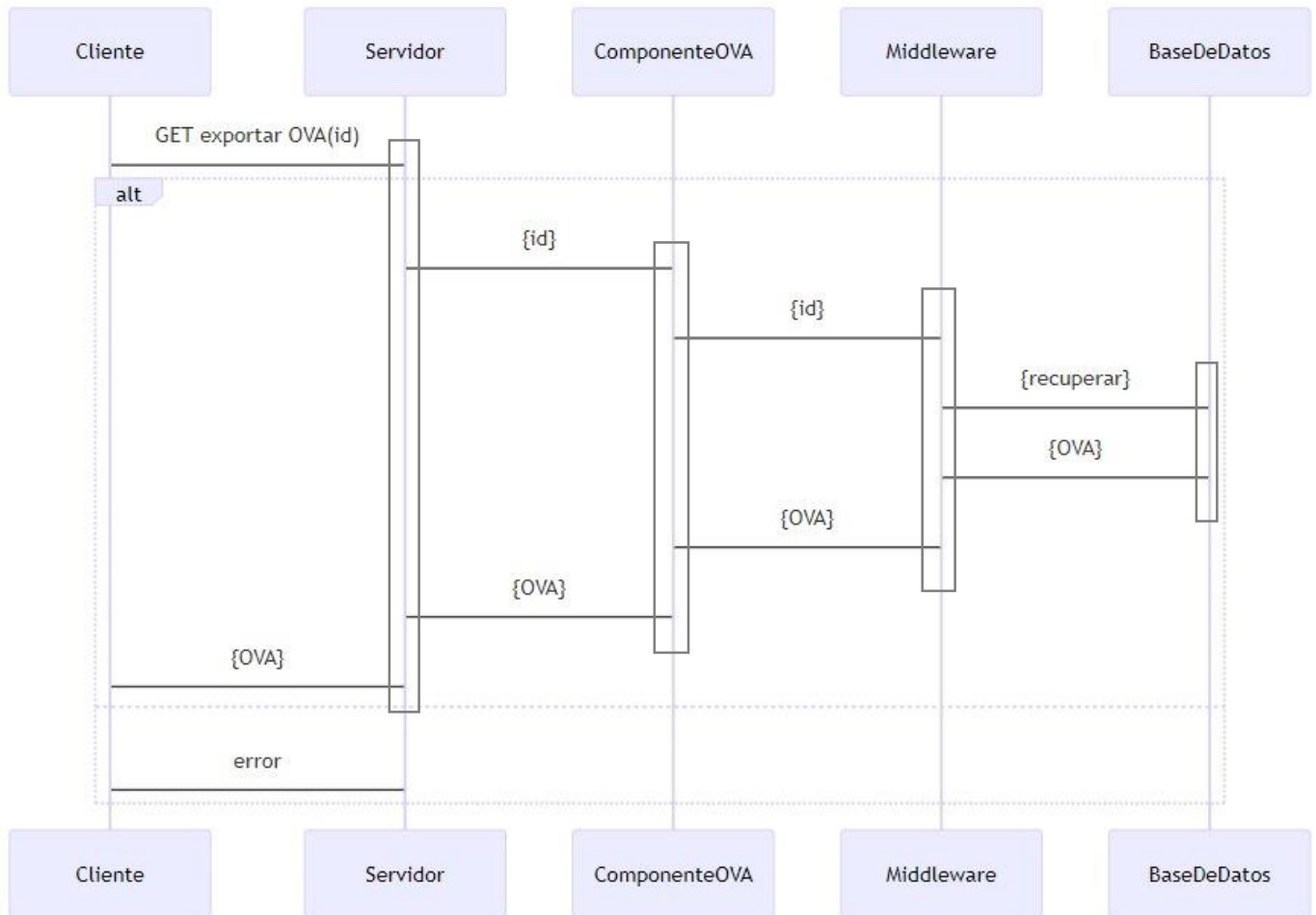
Eliminar OVA



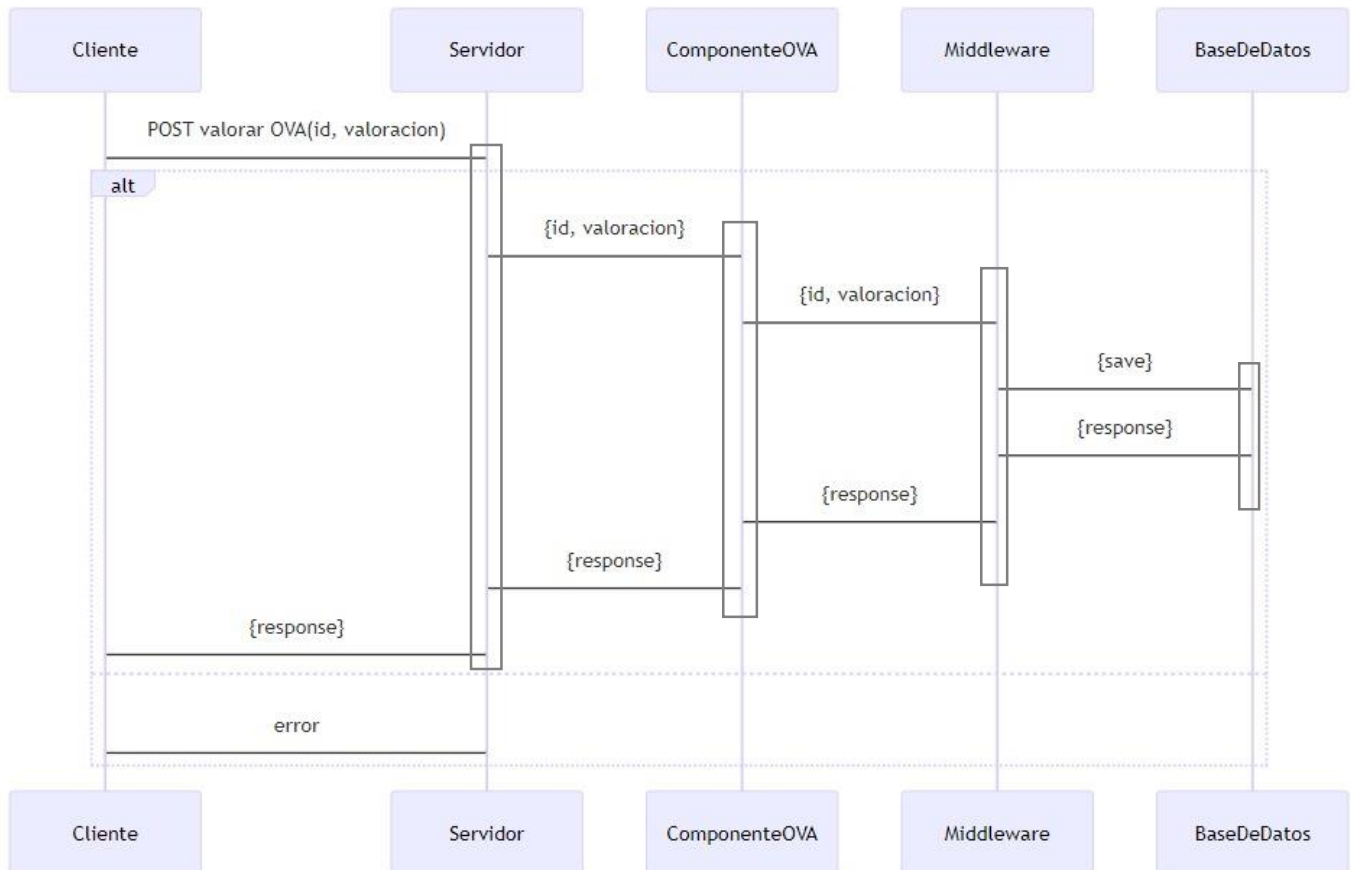
Buscar OVA



Exportar OVA



Valorar OVA



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 Esfuerzo:

- Muy alto (5)

- Alto (4)

- Medio (3)

- Bajo (2)

- Muy bajo (1)

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

Modelado E/R

Descripción de Entidades y Relaciones

1. Subir OVA

- Entidades:

1. OVA

- Atributos: ID archivo, Asignar Nombre, Descripción Ova, Colocar Autor, Asignar fecha y hora de subida.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

2. Visualizar OVA

- Entidades:

1. Visualización de OVA

- Atributos: ID archivo, Nombre del OVA, Descripción del OVA, Autor de subida, Fecha y hora de visualización.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

3. Actualizar OVA

- Entidades:

1. Actualización de OVA

- Atributos: ID archivo, Nombre del OVA, Descripción del OVA, Fecha y hora de actualización.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

4. Eliminar OVA

- Entidades:

1. Elimina OVA

- Atributos: ID archivo, Selecciona OVA a eliminar, Fecha y hora de eliminación.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

5. Buscar OVA

- Entidades:

1. Búsqueda de OVA

- Atributos: ID archivo, Criterio de búsqueda, Fecha y hora de búsqueda.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

6. Exportar OVA

- Entidades:

1. Exportación de OVA

- Atributos: ID archivo, Formato de exportación, Fecha y hora de exportación.

2. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

7. Valorar OVA

- Entidades:

1. Valoración de OVA

- Atributos: ID archivo, Opciones de valoración, Fecha y hora de valoración.

8. Usuario:

- Atributos: Nombre de usuario, Tipo de usuario.

DIAGRAMA ENTIDAD - RELACIÓN

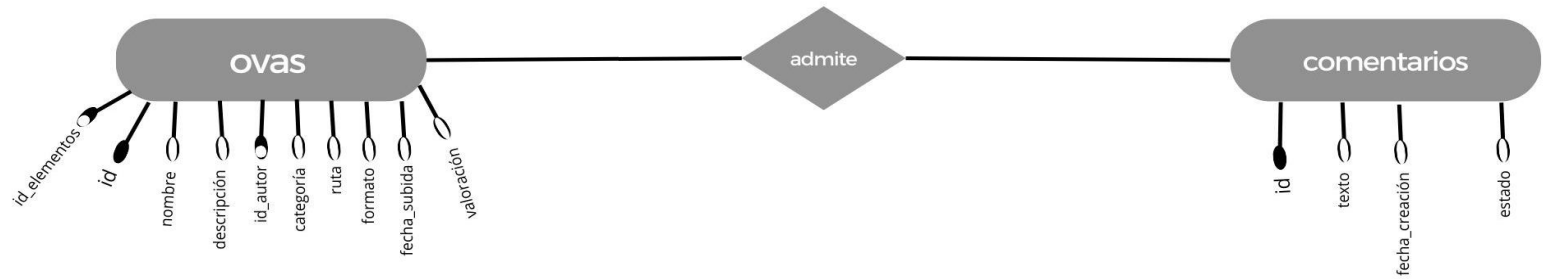


DIAGRAMA CARDINALIDAD

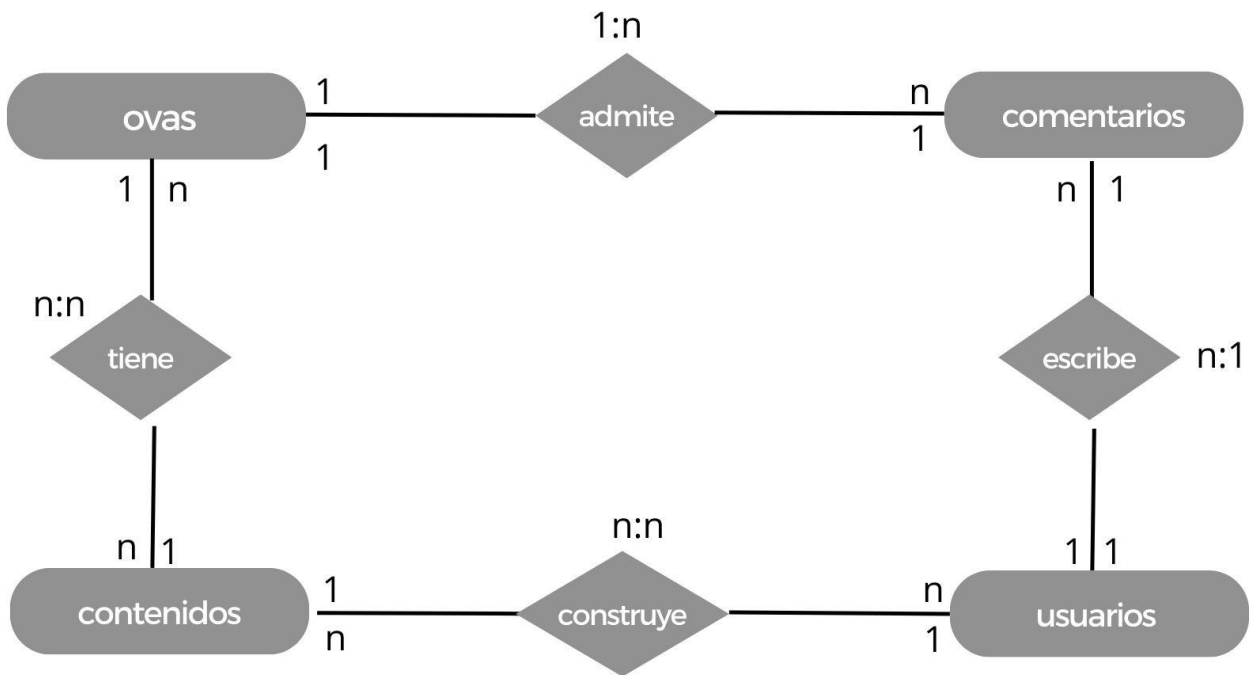
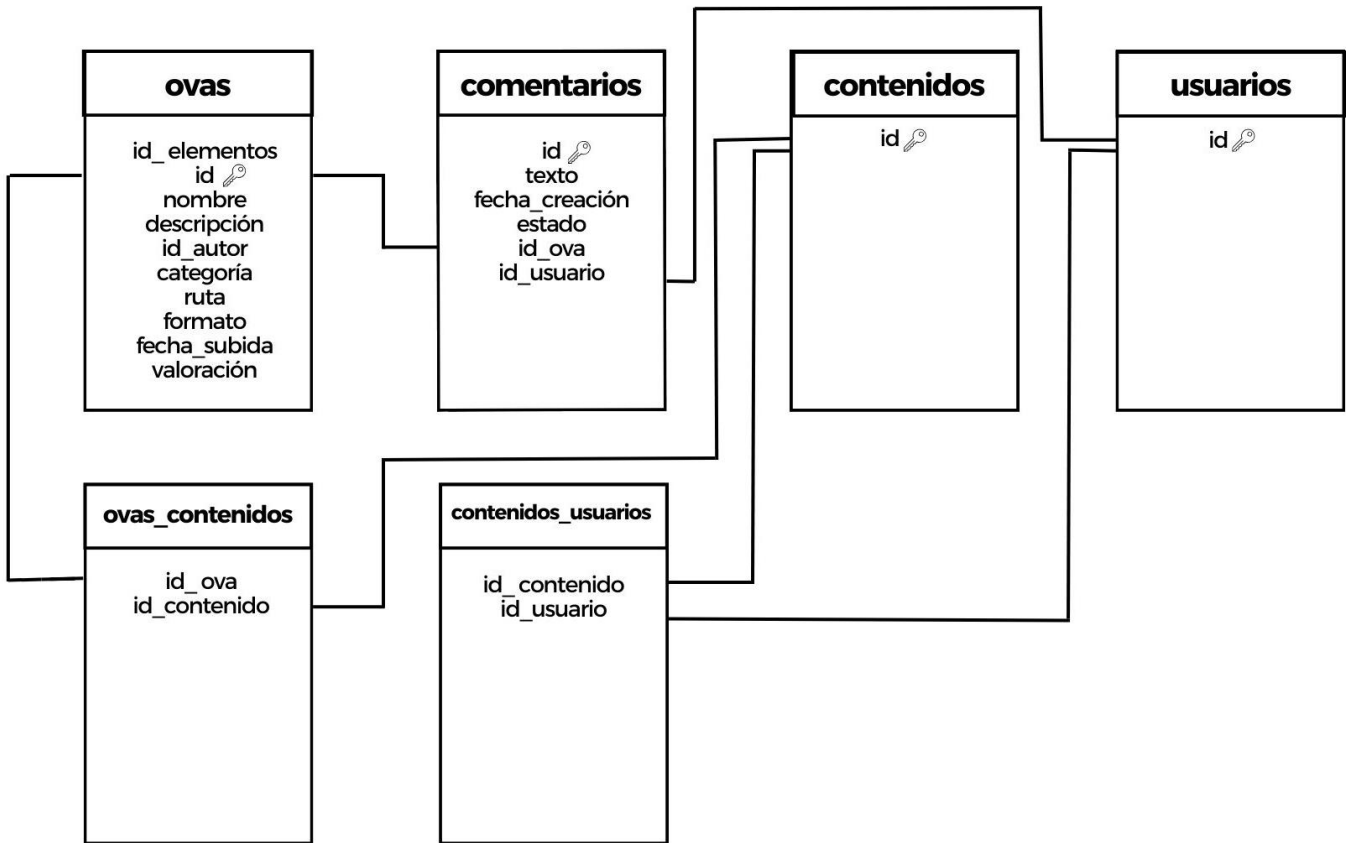


DIAGRAMA MODELO RELACIONAL



NO SQL

```

ovas: {id_elementos:
number,
id: number,
nombre: string,
descripcion: string,
id_autor: number,
categoria: string,
ruta: string,
formato: string,
fecha_subida: timedate,
valoracion: number,
id_contenidos: array,
}
  
```

```

comentarios: {id: number,
texto: string,
fecha_creacion: timedate,
estado: sting,
id_ovas: number,
id_usuarios: number,
}
  
```

```

contenidos: {id: number,
id_ovas: array,
id_usuarios: array,
}
  
```

```

usuarios: {id: number,
id_contenidos: array,
}
  
```

Schema Comentarios

```
import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
import { Document } from 'mongoose';

@Schema({ timestamps: true })
export class Comentarios extends Document {
  @Prop()
  texto: string;

  @Prop()
  senderId: string;

  @Prop({ default: () => new Date() })
  fecha_creacion: Date;

  @Prop()
  estado: string;

  @Prop()
  id_ovas: number;

  @Prop()
  id_usuarios: number;
}

export const ComentariosSchema = SchemaFactory.createForClass(Comentarios);
```

Schema Contenidos

```
import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
import { Document } from 'mongoose';

@Schema({ timestamps: true })
export class Contenidos extends Document {
  @Prop()
  id_ovas: [];

  @Prop()
  id_usuarios: [];
}

export const ContenidosSchema = SchemaFactory.createForClass(Contenidos);
```

Schema Ovas

```
import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
import { Document } from 'mongoose';

@Schema({ timestamps: true })
export class Ovas extends Document {
  @Prop()
  id_elementos: number;

  @Prop()
  nombre: string;

  @Prop()
  descripcion: string;

  @Prop()
  id_autor: number;

  @Prop()
  categoria: string;

  @Prop()
  ruta: string;

  @Prop()
  formato: string;

  @Prop({ default: () => new Date() })
  fecha_subida: Date;

  @Prop()
  valoracion: number;

  @Prop()
  id_contenido: [];
}

export const OvasSchema = SchemaFactory.createForClass(Ovas);
```

Schema Usuarios

```
import { Prop, Schema, SchemaFactory } from '@nestjs/mongoose';
import { Document } from 'mongoose';

@Schema({ timestamps: true })
export class Usuarios extends Document {
  @Prop()
  id_contenidos: [];
}

export const UsuariosSchema = SchemaFactory.createForClass(Usuarios);
```


Anexos (si es necesario)

Diagramas Adicionales

Referencias

Etapla 2: Persistencia de Datos con Backend

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

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

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Creación de la Interfaz de Usuario (UI)

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

Consideraciones de Usabilidad

Maquetación Responsiva

Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

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

Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend