

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

## Creación de modulo de gestion de tareas ( Taskassistant)

Edwin Bertel Negrete

[Ebertelnegrete38@correo.unicordoba.edu.co](mailto:Ebertelnegrete38@correo.unicordoba.edu.co)

Álvaro Pio Villalba

[Avillalbapaez52@correo.unicordoba.edu.co](mailto:Avillalbapaez52@correo.unicordoba.edu.co)

Guillermo del Valle Vitola

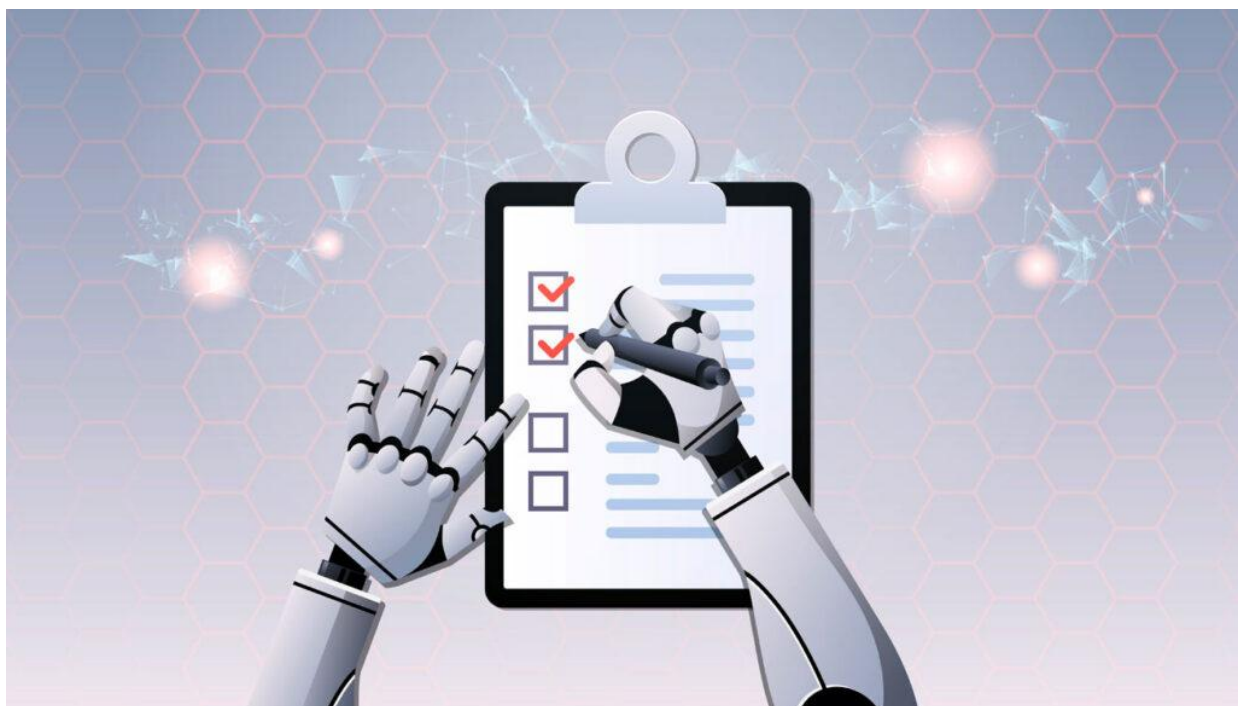
[Gdelvallevitola03@correo.unicordoba.edu.co](mailto:Gdelvallevitola03@correo.unicordoba.edu.co)

Jesi José Correa Galván

[Jcorreagalvan76@correo.unicordoba.edu.co](mailto:Jcorreagalvan76@correo.unicordoba.edu.co)

Bastian Nicolas Berastegui Barrera

[Bberasteguibarrera60@correo.unicordoba.edu.co](mailto:Bberasteguibarrera60@correo.unicordoba.edu.co)



## Descripción del software

El módulo de gestión de tareas está diseñado para simplificar la organización y el seguimiento de las actividades diarias en plataformas de contenido. La solución ofrece una interfaz intuitiva y múltiples funcionalidades que permiten la autogestión y distribución eficiente de tareas, con monitoreo en tiempo real del estado de cada proyecto. Se enfoca en mejorar la comunicación y coordinación dentro de los equipos, optimizando el rendimiento y asegurando una mayor calidad en la gestión de contenidos a través de diferentes plataformas.

<b>ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS .....</b>	<b>6</b>
<b>1. INTRODUCCIÓN .....</b>	<b>6</b>
PROPÓSITO DEL DOCUMENTO .....	6
ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA.....	8
DEFINICIONES Y ACRÓNIMOS.....	9
<b>2. DESCRIPCIÓN GENERAL .....</b>	<b>11</b>
OBJETIVOS DEL SISTEMA.....	11
FUNCIONALIDAD GENERAL.....	11
USUARIOS DEL SISTEMA .....	12
RESTRICCIONES .....	13
<b>3. REQUISITOS FUNCIONALES.....</b>	<b>13</b>
CASOS DE USO.....	14
DIAGRAMAS DE FLUJO DE CASOS DE USO .....	15
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO.....	15
PRIORIDAD DE REQUERIMIENTOS .....	16
<b>4. REQUISITOS NO FUNCIONALES.....</b>	<b>16</b>
REQUISITOS DE DESEMPEÑO .....	16
REQUISITOS DE SEGURIDAD.....	16
REQUISITOS DE USABILIDAD .....	16
REQUISITOS DE ESCALABILIDAD .....	16
<b>5. MODELADO E/R.....</b>	<b>17</b>
DIAGRAMA DE ENTIDAD-RELACIÓN.....	17
DIAGRAMA RELACIONAL.....	17
SCRIPT DE MODELO RELACIONAL.....	18
DESCRIPCIÓN DE ENTIDADES Y RELACIONES .....	18
REGLAS DE INTEGRIDAD REFERENCIAL.....	18
COLECCIONES (NOSLQ).....	18
<b>6. ANEXOS .....</b>	<b>19</b>
DIAGRAMAS ADICIONALES .....	19
REFERENCIAS.....	19
<b>ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND.....</b>	<b>20</b>
<b>7. INTRODUCCIÓN .....</b>	<b>20</b>
PROPÓSITO DE LA ETAPA .....	20
ALCANCE DE LA ETAPA .....	20
DEFINICIONES Y ACRÓNIMOS.....	20
<b>8. DISEÑO DE LA ARQUITECTURA DE BACKEND .....</b>	<b>20</b>
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA.....	20
COMPONENTES DEL BACKEND .....	20

DIAGRAMAS DE ARQUITECTURA .....	20
<b>9. ELECCIÓN DE LA BASE DE DATOS .....</b>	<b>20</b>
EVALUACIÓN DE OPCIONES (SQL O NOSQL) .....	21
JUSTIFICACIÓN DE LA ELECCIÓN .....	21
DISEÑO DE ESQUEMA DE BASE DE DATOS.....	21
<b>10. IMPLEMENTACIÓN DEL BACKEND.....</b>	<b>21</b>
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN .....	21
CREACIÓN DE LA LÓGICA DE NEGOCIO .....	21
DESARROLLO DE ENDPOINTS Y APIS.....	21
AUTENTICACIÓN Y AUTORIZACIÓN .....	21
<b>11. CONEXIÓN A LA BASE DE DATOS .....</b>	<b>21</b>
CONFIGURACIÓN DE LA CONEXIÓN.....	22
DESARROLLO DE OPERACIONES CRUD .....	22
MANEJO DE TRANSACCIONES .....	22
<b>12. PRUEBAS DEL BACKEND .....</b>	<b>22</b>
DISEÑO DE CASOS DE PRUEBA .....	22
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN .....	22
MANEJO DE ERRORES Y EXCEPCIONES .....	22
<b>ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND .....</b>	<b>23</b>
<b>13. INTRODUCCIÓN.....</b>	<b>23</b>
PROPÓSITO DE LA ETAPA .....	23
ALCANCE DE LA ETAPA .....	23
DEFINICIONES Y ACRÓNIMOS.....	23
<b>14. CREACIÓN DE LA INTERFAZ DE USUARIO (UI) .....</b>	<b>23</b>
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS .....	23
CONSIDERACIONES DE USABILIDAD .....	23
MAQUETACIÓN RESPONSIVA.....	23
<b>15. PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS) .....</b>	<b>23</b>
DESARROLLO DE LA LÓGICA DEL FRONTEND .....	24
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS.....	24
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE) .....	24
<b>16. CONSUMO DE DATOS DESDE EL BACKEND .....</b>	<b>24</b>
CONFIGURACIÓN DE CONEXIONES AL BACKEND .....	24
OBTENCIÓN Y PRESENTACIÓN DE DATOS .....	24
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE) .....	24
<b>17. INTERACCIÓN USUARIO-INTERFAZ .....</b>	<b>24</b>
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS.....	25

IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS .....	25
MEJORAS EN LA EXPERIENCIA DEL USUARIO .....	25
<b>18. PRUEBAS Y DEPURACIÓN DEL FRONTEND .....</b>	<b>25</b>
DISEÑO DE CASOS DE PRUEBA DE FRONTEND .....	25
PRUEBAS DE USABILIDAD.....	25
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO .....	25
<b>19. IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND .....</b>	<b>25</b>
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) .....	26
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND .....	26
<b>20. INTEGRACIÓN CON EL BACKEND.....</b>	<b>26</b>
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND .....	26
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND .....	26
ANEXOS .....	26

# Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

## 1. 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, álgebra 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 TaskAssitant

TaskAssitant es una aplicación web diseñada para facilitar la organización, asignación, y seguimiento de tareas en entornos colaborativos. Esta herramienta no solo busca digitalizar la gestión de tareas, sino hacerlo de manera inteligente, intuitiva y adaptativa, mediante el uso de automatización, análisis de productividad y sugerencias personalizadas. Su alcance abarca la gestión integral de tareas, desde su creación hasta su finalización, incluyendo funcionalidades como priorización, distribución automatizada, asignación de recursos, seguimiento en tiempo real, visualización en distintos formatos (lista, kanban, calendario), y colaboración activa entre usuarios mediante comentarios, notificaciones y control de versiones. Además, la plataforma incorpora herramientas de monitoreo del desempeño y generación de reportes, garantizando una experiencia adaptable a distintos perfiles de usuario y entornos de trabajo, con enfoque en la eficiencia, la claridad operativa y la mejora continua del rendimiento individual y grupal.

- Eliminar Tarea
- Recuperar Tarea
- Ajustar Fechas de Entrega
- Agregar comentarios Colaborativos
- Agregar Mención
- Monitorear Progreso
- Distribuir de tarea
- Asignar Recursos
- Asignar tarea
- Resumen general por día.
- Sincronizar tareas con calendarios.
- Etiquetar tareas con colores
- Etiquetar tareas con palabras claves
- Agrupar tareas por proyecto.
- Duplicar tareas
- Guardar cambios de tarea
- Crear metas semanales
- Crear metas mensuales
- Calificar tareas completadas
- Notificar cambios
- Registrar tiempo por tarea
- Exportar tareas
- Exportar en formato PDF.
- Exportar en formato Excel.



- Importar tareas desde archivos externos
- Anclar tareas

#### Funcionalidades futuras

- Personalizar la vista del tablero (modo lista, kanban, calendario).
- Filtrar tareas por estado, etiqueta, fecha de vencimiento o asignado.
- Programar tareas recurrentes (diarias, semanales, mensuales).
- Reasignar tareas rápidamente desde la vista de tablero.
- Evaluar el cumplimiento de objetivos con gráficas de rendimiento.
- Integrar con servicios de almacenamiento en la nube (Drive, Dropbox).
- Predecir fechas de vencimiento óptimas usando aprendizaje automático según la carga de trabajo del usuario.
- Sugerir tareas prioritarias automáticamente con base en comportamiento pasado y plazos cercanos.
- Asistir por voz o chat con IA para crear, editar o consultar tareas con lenguaje natural.
- Recomendar pausas o descansos inteligentes cuando detecte sobrecarga de tareas.
- Detectar conflictos de agenda entre tareas, eventos y reuniones.
- Ofrecer gamificación de productividad, con logros, medallas y niveles.
- Predecir riesgos de retraso y emitir alertas preventivas.
- Aplicar reglas automatizadas personalizadas (Ej: "Si es tarea urgente, enviar notificación por WhatsApp").
- Integrar dashboards con métricas de bienestar digital, como tiempo frente a pantalla o nivel de dispersión.

## 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).

## 2. Descripción General

### Objetivos del Sistema

El objetivo del sistema es proporcionar una pizarra compartida dentro de un Sistema de Gestión de Contenido llamado CREAUI que permita a los usuarios colaborar de manera eficiente y efectiva, facilitando la creación, visualización y edición de contenido visual en tiempo real. Esta pizarra compartida se diseñará con el propósito de mejorar la comunicación y la colaboración en un entorno en línea, ofreciendo a los usuarios una plataforma intuitiva y versátil para crear y compartir ideas, diagramas, esquemas y contenido visual de manera colaborativa, enriqueciendo así la experiencia de usuario y la productividad en el uso del CMS.

### Funcionalidad General

- **Creación y Edición Colaborativa:** Permite a los usuarios crear y editar contenido en la pizarra de forma colaborativa en tiempo real. Múltiples usuarios pueden trabajar en el mismo documento simultáneamente.
- **Herramientas de Dibujo y Anotación:** Proporciona herramientas de dibujo, pinceles, formas y opciones de anotación que permiten a los usuarios plasmar sus ideas y conceptos de manera visual.
- **Carga de Imágenes y Multimedia:** Permite a los usuarios cargar imágenes, videos y otros medios directamente en la pizarra, lo que facilita la ilustración de conceptos.
- **Organización de Contenido:** Ofrece opciones para organizar y estructurar el contenido en la pizarra, como la creación de capas, agrupación de elementos y uso de etiquetas.
- **Historial de Revisiones:** Registra un historial de revisiones que permite a los usuarios rastrear los cambios realizados en la pizarra y restaurar versiones anteriores si es necesario.
- **Compartir y Colaborar:** Permite compartir la pizarra con otros usuarios a través de enlaces o invitaciones, lo que facilita la colaboración con colegas, clientes o amigos.
- **Comentarios y Chat en Tiempo Real:** Los usuarios pueden comentar y discutir sobre el contenido de la pizarra a través de un chat en tiempo real, lo que facilita la comunicación durante la colaboración.
- **Exportación e Impresión:** Ofrece la capacidad de exportar el contenido de la pizarra en varios formatos (PDF, imagen, etc.) y la opción de imprimirlo.

- **Integración con el CMS:** Se integra de manera transparente con el sistema de gestión de contenido (CMS CREAVI), lo que permite incrustar pizarras en los contenidos, metodologías o cualquier otro tipo de componente que permita la pizarra.
- **Personalización y Temas:** Permite a los usuarios personalizar la apariencia de la pizarra y seleccionar temas que se adapten a sus necesidades.
- **Acceso Seguro:** Proporciona medidas de seguridad para garantizar que solo los usuarios autorizados puedan acceder y editar la pizarra.
- **Notificaciones y Actualizaciones en Tiempo Real:** Los usuarios reciben notificaciones sobre cambios en la pizarra y pueden ver actualizaciones en tiempo real mientras otros editan.
- **Acceso Móvil:** Ofrece una experiencia de usuario optimizada en dispositivos móviles, permitiendo el acceso y la colaboración desde smartphones y tabletas.
- **Búsqueda y Filtros:** Facilita la búsqueda de contenido en la pizarra y la aplicación de filtros para organizar y encontrar información específica.
- **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 de la pizarra, como quién la ha editado, cuándo se realizaron cambios y estadísticas sobre el contenido (XAPI).

## Usuarios del Sistema

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

Funcionalidad	Administrador	Docente Principal	Docente Invitado	Estudiante
Crear Tarea	✓	✓		
Eliminar Tarea	✓	✓		
Recuperar tarea	✓	✓		
Ajustar fechas de entrega	✓	✓		
Agregar Comentarios Colaborativos	✓	✓	✓	✓
Agregar mención	✓	✓	✓	✓
Monitorear progreso	✓	✓	✓	✓
Distribuir tareas	✓	✓		
Asignar Recursos	✓	✓		
Asignar Tarea		✓		
Sincronizar Tarea	✓	✓	✓	✓
Etiquetar tareas con colores	✓	✓	✓	✓
Etiquetar tareas con palabras clave	✓	✓	✓	✓
Agrupar Tarea Por Proyecto	✓	✓	✓	✓
Duplicar Tareas	✓	✓	✓	✓

Guardar Cambios De tarea	✓	✓		
Crear Metas Semanales	✓	✓	✓	✓
Crear Metas Mensuales	✓	✓		
Calificar Tareas Completadas	✓	✓		
Notificar Cambios	✓	✓	✓	✓
Registrar Tiempo por tarea	✓	✓	✓	
Exportar Tarea	✓	✓	✓	✓
Exportar en formato PDF	✓	✓	✓	✓
Exportar en formato Excel	✓	✓	✓	✓
Anclar Tareas	✓	✓	✓	✓

## Restricciones

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

## 3. Requisitos Funcionales

### Creación y Edición de Contenido:

- Los usuarios deben poder crear y editar contenido en tiempo real en la pizarra.
- Los usuarios deben tener acceso a herramientas de dibujo, anotación y edición de contenido.

### Compartir y Colaborar:

- Los usuarios deben poder compartir la pizarra con otros usuarios a través de enlaces o invitaciones.
- Múltiples usuarios deben poder colaborar en la misma pizarra simultáneamente.

### Carga de Multimedia:

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

### Historial de Revisiones:

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

**Comentarios y Chat en Tiempo Real:**

- Los usuarios deben poder comentar y discutir sobre el contenido de la pizarra a través de un chat en tiempo real.

**Exportación e Impresión:**

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

**Integración con el CMS:**

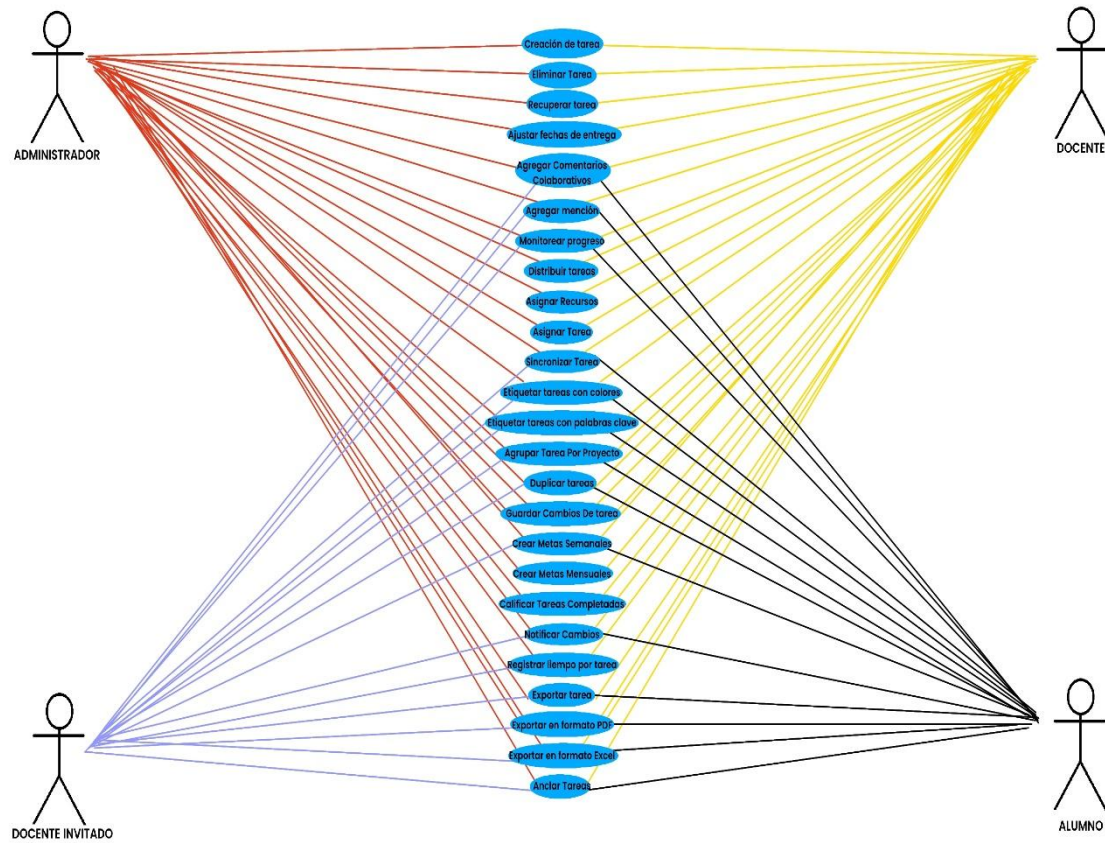
- La pizarra debe integrarse con el sistema de gestión de contenidos (CMS) existente para permitir la incrustación de pizarras en páginas web o artículos.

**Personalización y Temas:**

- Los usuarios deben poder personalizar la apariencia de la pizarra y seleccionar temas que se adapten a sus necesidades.

## Casos de Uso

Diagrama de caso de uso



<https://app.creately.com/d/start/dashboard>

## Diagramas de Flujo de Casos de Uso

Descripción detallada de cada caso de uso

Prioridad de Requerimientos

## 4. Requisitos No Funcionales

Requisitos de Desempeño

Requisitos de Seguridad

Requisitos de Usabilidad

Requisitos de Escalabilidad



## 5. Modelado E/R

Diagrama de Entidad-Relación

Diagrama Relacional

Script de modelo relacional

Descripción de Entidades y Relaciones

Reglas de Integridad Referencial

Colecciones (NoSLQ)

## 6. Anexos

Diagramas Adicionales

Referencias

## Etapa 2: Persistencia de Datos con Backend

### 7. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

### 8. Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

### 9. Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

## 10. Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

## 11. Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

## 12. Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

## Etapa 3: Consumo de Datos y Desarrollo Frontend

### 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.