

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

Creación de módulo de gestión de tareas (Taskassistant)

Edwin José Bertel Negrete

Ebertelnegrete38@correo.unicordoba.edu.co

Álvaro Pio Villalba Páez

Avillalbapaez52@correo.unicordoba.edu.co

Guillermo Andrés del Valle Vitola

Gdelvallevitola03@correo.unicordoba.edu.co

Jesi José Correa Galván

Jcorreagalvan76@correo.unicordoba.edu.co

Bastían Nicolas Berástegui Barrera

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.

ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS	6
1. INTRODUCCIÓN	6
PROPÓSITO DEL DOCUMENTO	6
ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA	8
DEFINICIONES Y ACRÓNIMOS	9
2. DESCRIPCIÓN GENERAL	11
OBJETIVOS DEL SISTEMA.....	11
FUNCIONALIDAD GENERAL	11
USUARIOS DEL SISTEMA	12
RESTRICCIONES	13
3. REQUISITOS FUNCIONALES	13
CASOS DE USO	14
DIAGRAMAS DE FLUJO DE CASOS DE USO	17
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO	17
PRIORIDAD DE REQUERIMIENTOS	63
4. REQUISITOS NO FUNCIONALES	63
REQUISITOS DE DESEMPEÑO	63
REQUISITOS DE SEGURIDAD	66
REQUISITOS DE USABILIDAD	67
REQUISITOS DE ESCALABILIDAD	68
5. MODELADO E/R	69
DIAGRAMA DE ENTIDAD-RELACIÓN	69
DIAGRAMA RELACIONAL.....	69
SCRIPT DE MODELO RELACIONAL.....	70
DESCRIPCIÓN DE ENTIDADES Y RELACIONES	70
REGLAS DE INTEGRIDAD REFERENCIAL	70
COLECCIONES (NoSQL)	70
6. ANEXOS	71
DIAGRAMAS ADICIONALES	71
REFERENCIAS	71
PERSISTENCIA DE DATOS CON BACKEND.....	72
7. INTRODUCCIÓN	72
PROPÓSITO DE LA ETAPA	72
ALCANCE DE LA ETAPA	72
DEFINICIONES Y ACRÓNIMOS	72
8. DISEÑO DE LA ARQUITECTURA DE BACKEND	72
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA	72
COMPONENTES DEL BACKEND	72
DIAGRAMAS DE ARQUITECTURA	72

ETAPA 2:

9.	ELECCIÓN DE LA BASE DE DATOS	72
	EVALUACIÓN DE OPCIONES (SQL O NoSQL)	73
	JUSTIFICACIÓN DE LA ELECCIÓN	73
	DISEÑO DE ESQUEMA DE BASE DE DATOS	73
10.	IMPLEMENTACIÓN DEL BACKEND	73
	ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN	73
	CREACIÓN DE LA LÓGICA DE NEGOCIO	73
	DESARROLLO DE ENDPOINTS Y APIS	73
	AUTENTICACIÓN Y AUTORIZACIÓN	73
11.	CONEXIÓN A LA BASE DE DATOS	73
	CONFIGURACIÓN DE LA CONEXIÓN.....	74
	DESARROLLO DE OPERACIONES CRUD	74
	MANEJO DE TRANSACCIONES	74
12.	PRUEBAS DEL BACKEND	74
	DISEÑO DE CASOS DE PRUEBA	74
	EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN	74
	MANEJO DE ERRORES Y EXCEPCIONES	74
CONSUMO DE DATOS Y DESARROLLO FRONTEND		75
13.	INTRODUCCIÓN.....	75
	PROPÓSITO DE LA ETAPA	75
	ALCANCE DE LA ETAPA	75
	DEFINICIONES Y ACRÓNIMOS	75
14.	CREACIÓN DE LA INTERFAZ DE USUARIO (UI)	75
	DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS	75
	CONSIDERACIONES DE USABILIDAD	75
	MAQUETACIÓN RESPONSIVA	75
15.	PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS)	75
	DESARROLLO DE LA LÓGICA DEL FRONTEND	76
	MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	76
	USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE)	76
16.	CONSUMO DE DATOS DESDE EL BACKEND	76
	CONFIGURACIÓN DE CONEXIONES AL BACKEND	76
	OBTENCIÓN Y PRESENTACIÓN DE DATOS	76
	ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE)	76
17.	INTERACCIÓN USUARIO-INTERFAZ	76
	MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS	77
	IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS	77

MEJORAS EN LA EXPERIENCIA DEL USUARIO	77
18. PRUEBAS Y DEPURACIÓN DEL FRONTEND	77
DISEÑO DE CASOS DE PRUEBA DE FRONTEND	77
PRUEBAS DE USABILIDAD	77
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO	77
19. IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND	77
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO)	78
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND	78
20. INTEGRACIÓN CON EL BACKEND	78
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND	78
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND	78
ANEXOS	78

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, 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 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.

- CRUD 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).

PWA: Aplicación Web Progresiva (Progressive Web App).

2. Descripción General

Objetivos del Sistema

El objetivo del sistema es proporcionar una pizarra compartida dentro de un Sistema de Gestión de Contenido llamado CREAVI 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
CRUD 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:

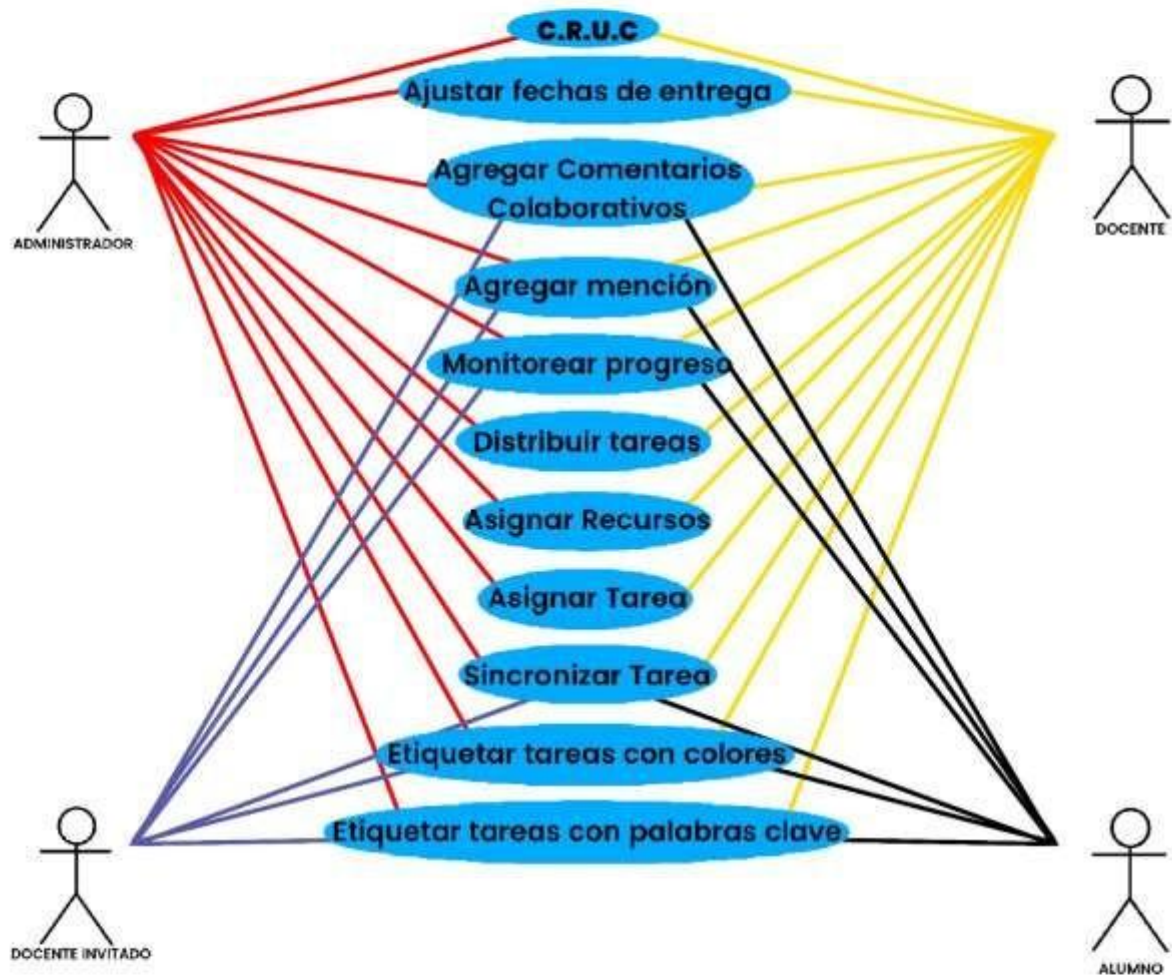
- La pizarra debe integrarse con el sistema de gestión de contenidos (CMS) existente para permitir la incrustación de pizarras en páginas web o artículos.

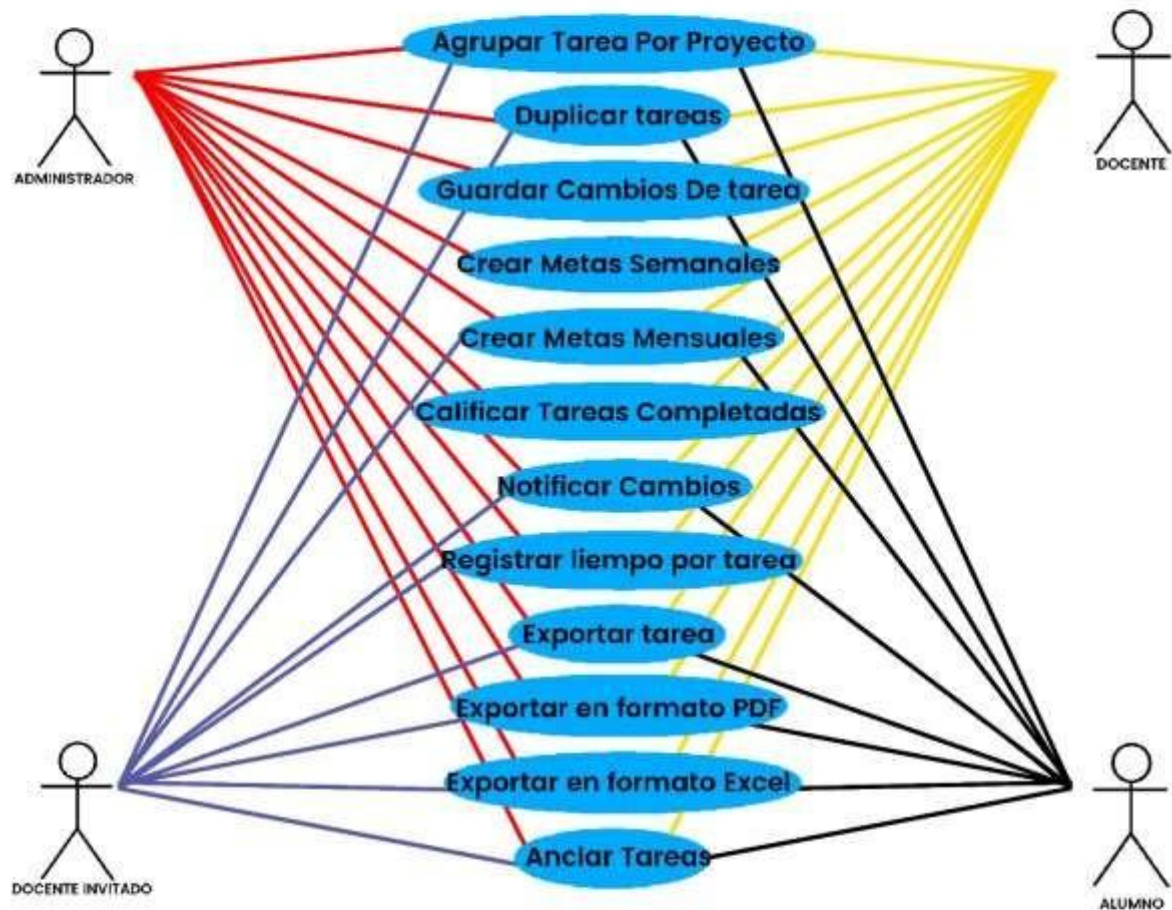
Personalización y Temas:

- 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

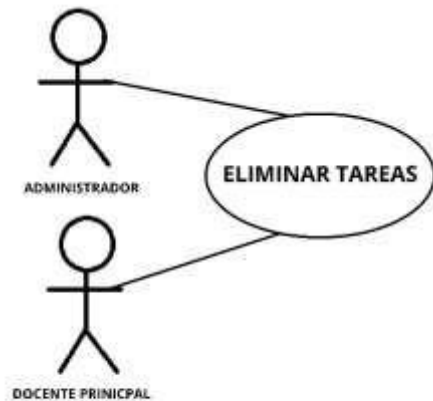


CASO No. 1 Creación de Tareas

ID:	TA-1	
Nombre	Creación de tareas	
Actores	Administrador, Docente Principal	
Objetivo	Permitir al administrador y docente principal crear nuevas tareas.	
Urgencia	5	
Esfuerzo	4	
Pre-condiciones	El administrador y docente principal debe estar autenticado en el sistema.	
	Docente	Sistema
	Opción "Crear Tarea"	

Flujo Normal		Retorna el formulario para crear una tarea
	Rellena el formulario	
		Valida los datos
		guarda los datos de la tarea, retorna un mensaje diciendo la tarea se ha guardado con éxito.
Post-condiciones	La tarea queda registrada en el sistema.	

CASO DE USO 2 - ELIMINAR TAREAS

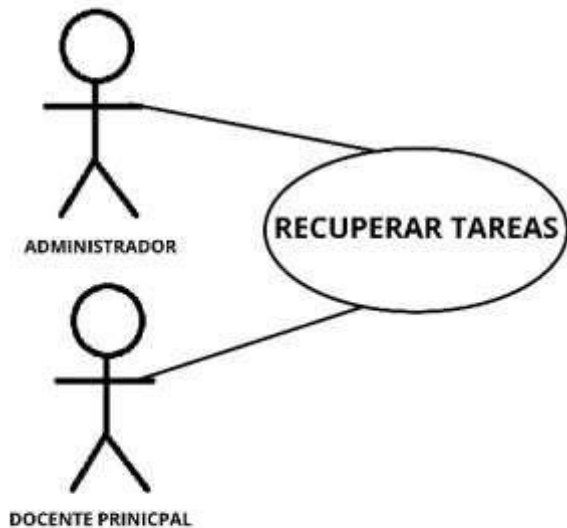


CASO No. 2 Eliminar Tarea

ID:	TA-2
Nombre	Eliminar Tarea
Actores	Administrador, Docente principal

Objetivo	Permitir al administrador y docente principal eliminar tareas obsoletas o incorrectas	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	El administrador debe estar autenticado en el sistema.	
Flujo Normal	Usuario	Sistema
	Selecciona una tarea a eliminar	
		Muestra confirmación de eliminación
	Presiona "Eliminar"	
		El sistema borra la tarea
		El sistema retorna un mensaje diciendo la tarea se ha eliminado con éxito.
Post-condiciones	La tarea eliminada se elimina de las vistas activas	
Exepciones	Error al eliminar tarea por falta de conexión.	

CASO DE USO 3 - RECUPERAR TAREAS



CASO No. 3 Recuperar Tareas Eliminadas

ID:	TA-3
Nombre	Recuperar Tareas Eliminadas
Actores	Administrador, Docente Principal
Objetivo	Permitir al administrador y docente Recuperar Tareas Eliminadas
Urgencia	3
Esfuerzo	2
Pre-condiciones	El administrador y docente debe haber iniciado sesión en el sistema, el sistema debe tener tareas previamente eliminadas

Flujo Normal	Usuario	Sistema
	Accede al módulo de tareas eliminadas	
		Retorna las tareas eliminadas
	Selecciona la tarea a recuperar	
		Retorna la tarea seleccionada
	Visualiza la Tarea	
		Cambiar el estado de eliminada a recuperada
	Cierra el modulo	
Post-condiciones	La tarea es visualizada y descargada por el docente	
Exepciones	El sistema registra la acción como “Tarea recuperada “	
	Error de conexión con el servidor	
	Archivo de tarea dañado	

CASO DE USO 4 - AJUSTAR FECHAS



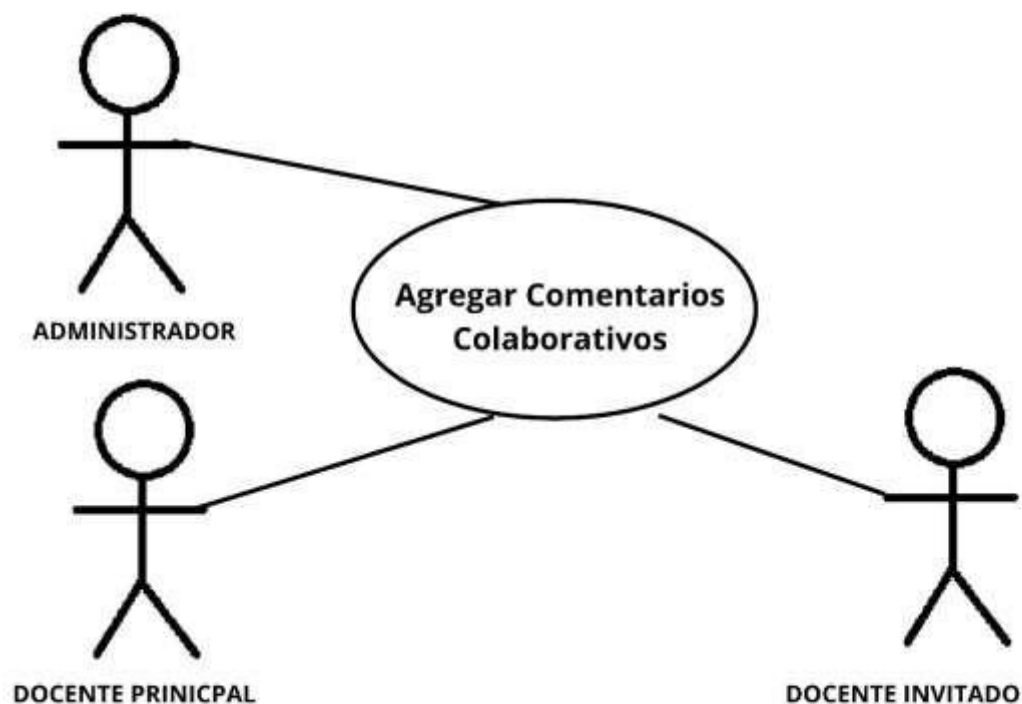
CASO No. 4 Ajustar fechas de entrega

ID:	TA-4	
Nombre	Ajustar fechas de entrega	
Actores	Administrador y Docente	
Objetivo	Permitir al administrador modificar las fechas límite de las tareas.	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	El administrador debe haber iniciado sesión y tener permisos sobre la tarea.	
Flujo Normal	Usuario	Sistema

	El Usuario accede a la tarea	
--	------------------------------	--

		Modifica la fecha y confirma los cambios
	Selecciona la opción “Ajustar fecha de entrega”	
		guarda la nueva fecha y actualiza la información
Post-condiciones	La tarea queda actualizada con la nueva fecha	
Excepciones	Fecha ingresada inválida; error de conexión	

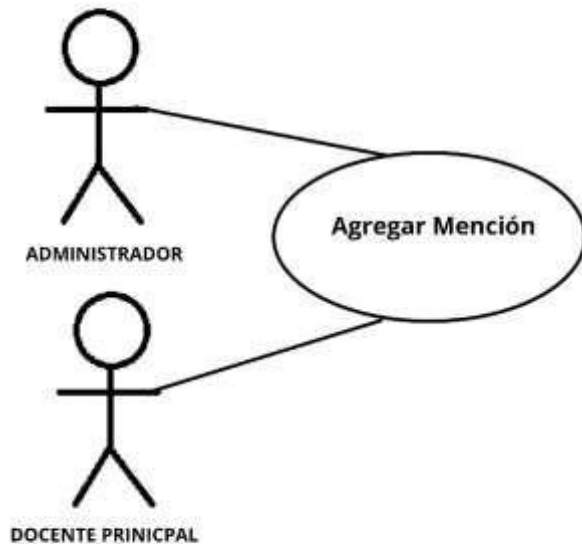
CASO DE USO 5 - AGREGAR COMENTARIOS COLABORATIVOS



CASO No. 5 Agregar Comentarios Colaborativos

ID:	TA-5	
Nombre	Agregar Comentarios Colaborativos	
Actores	Administrador, Docente Principal, Docente Invitado	
Objetivo	Permitir que los usuarios puedan añadir comentarios a las tareas para facilitar la colaboración.	
Urgencia	3	
Esfuerzo	3	
Pre-condiciones	El usuario debe haber iniciado sesión y tener acceso a la tarea.	
Flujo Normal	Docente	Sistema
	selecciona la tarea.	
		El sistema lo registra y lo muestra a todos los colaboradores
	Abre la sección de comentarios.	
	Escribe el comentario y lo envía	
Post-condiciones	El comentario queda vinculado a la tarea	
Excepciones	Error de envío por problemas de conexión.	

CASO DE USO 6 - AGREGAR MENCIÓN

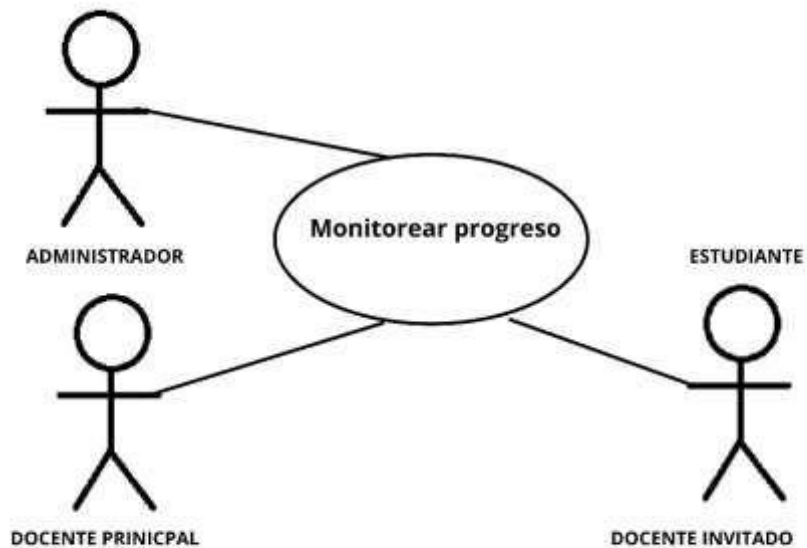


CASO No. 6 Agregar Mención

ID:	TA-6	
Nombre	Agregar Mención	
Actores	Administrador, Docente Principal	
Objetivo	Permitir mencionar a otros usuarios en los comentarios de tareas.	
Urgencia	2	
Esfuerzo	2	
Pre-condiciones	El usuario debe haber iniciado sesión y estar escribiendo un comentario.	
Flujo Normal	Docente	Sistema

	Abre una tarea y accede a la sección de comentarios.	
	Redacta un comentario y escribe "@nombre" para mencionar a alguien	
		Detecta el patrón de mención (@) y valida que el usuario existe
		Resalta la mención en el comentario
		Envía una notificación al usuario mencionado
		El comentario es registrado y el usuario mencionado es notificado
Post-condiciones	La mención queda registrada y el usuario recibe una notificación.	
Excepciones	Usuario no existente o sin acceso; falla en el envío de la notificación	

CASO DE USO 7 - MONITOREAR PROGRESO

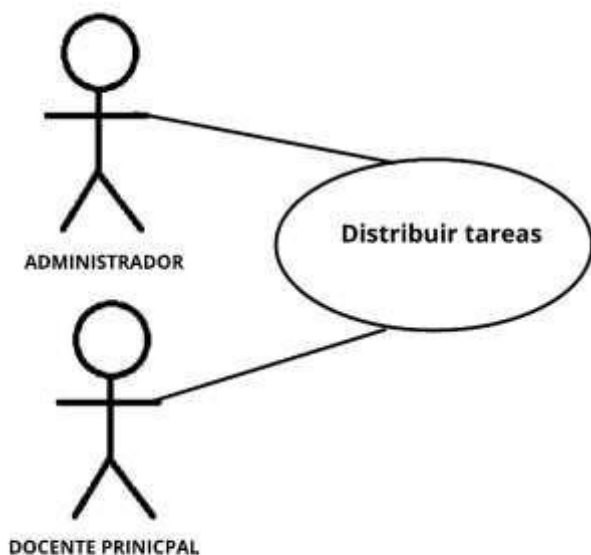


CASO No. 7 Monitorear progreso

ID:	TA-7	
Nombre	Monitorear progreso	
Actores	Administrador, Docente Principal, Docente Invitado	
Objetivo	Ver avance de tareas.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	Autenticado, tareas asignadas.	
Flujo Normal	Docente	Sistema

	Accede al módulo de tareas	
		Muestra lista de tareas asignadas
	Selecciona una tarea específica	
		Muestra barra o porcentaje de avance basado en estado o subtareas
		El docente visualiza el estado actual de progreso de la tarea
Post-condiciones	Visualiza progreso.	

CASO DE USO 8 - DISTRIBUIR TAREAS

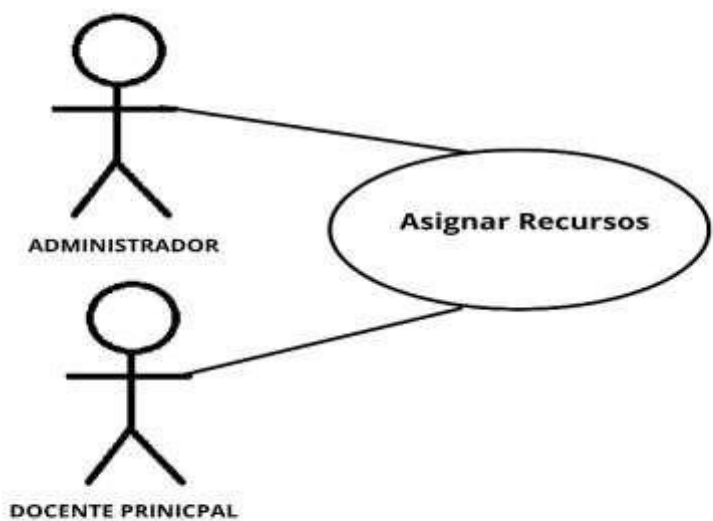


CASO No. 8 Distribuir tareas

ID:	TA-8	
Nombre	Distribuir tareas	
Actores	Administrador, Docente Principal	
Objetivo	Repartir tareas entre usuarios	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	Tener tareas disponibles	
Flujo Normal	Docente	Sistema

	cede al módulo de tareas no asignadas	
		Muestra lista de tareas disponibles
	Selecciona una o varias tareas	
	Selecciona usuarios para asignar cada tarea	
		Registra la asignación y notifica a cada usuario
		Tareas distribuidas correctamente y notificaciones enviadas
Post-condiciones	Tareas distribuidas.	
Excepciones		

CASO DE USO 9 - ASIGNAR RECURSOS

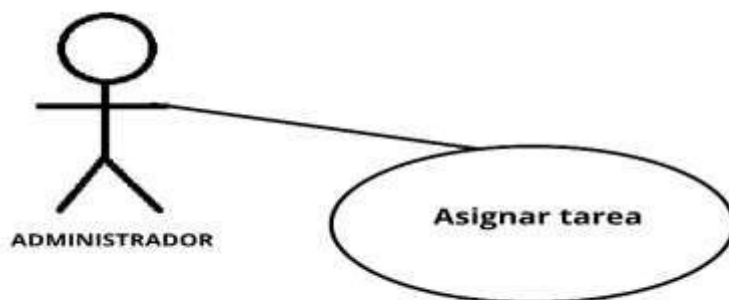


CASO No. 9 Asignar recursos

ID:	TA-9	
Nombre	Asignar recursos	
Actores	Administrador, Docente Principal	
Objetivo	Añadir recursos a tareas	
Urgencia	3	
Esfuerzo	3	
Pre-condiciones	Tener tareas creadas	
Flujo Normal	Docente	Sistema

	Abre una tarea	
	Sube archivo o link (PDF, imagen, video, etc.)	
		Valida el formato y tamaño del archivo
		Almacena y vincula el recurso a la tarea correspondiente
		La tarea tiene recursos accesibles por los participantes
	Sube recurso	
Post-condiciones	Recurso disponible	

CASO DE USO 10 - ASIGNAR TAREA



CASO No. 10 Asignar tareas

ID:	TA-10	
Nombre	Asignar tareas	
Actores	Administrador	
Objetivo	Asignar tarea a usuarios	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	Tener usuario	
Flujo Normal	Docente	Sistema
	Selecciona una tarea	
	Selecciona un usuario o grupo de usuarios a los que se asignará	
		Registra la asignación
		Envía notificación al usuario asignado
		Usuario o grupo informado y tarea asignada
Post-condiciones	Usuario Notificado	

CASO DE USO 11 - SINCRONIZAR TAREA



CASO No. 11 Sincronizar Tareas

ID:	TA-11	
Nombre	Sincronizar tareas	
Actores	Administrador, docente principal, docente invitado	
Objetivo	Sincronizar tareas	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	Tener tarea creada	
Flujo Normal	Docente	Sistema

	Selecciona una tarea para sincronizar con calendario externo	
		Muestra opciones: Google Calendar, Outlook, etc.
	Autoriza acceso e integración	
		Establece enlace y crea evento en calendario
		Evento sincronizado y tarea vinculada en calendario externo
Post-condiciones	Tarea sincronizada	
Excepciones	No tener tarea creada	

CASO DE USO 12 - ETIQUETAR TAREAS CON COLORES

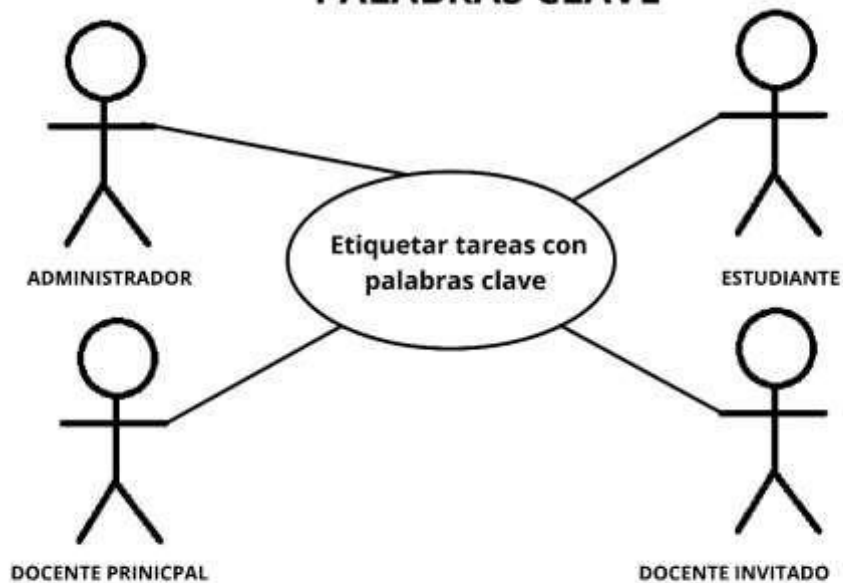


CASO No. 12 Etiquetar tarea con colores

ID:	TA-12	
Nombre	Etiquetar tareas con colores	
Actores	Administrador, docente principal, docente invitado	
Objetivo	Usar colores para asignar tareas	
Urgencia	2	
Esfuerzo	1	
Pre-condiciones	Tener usuario y tarea asignada	
Flujo Normal	Docente	Sistema

	Selecciona una tarea del listado	
	Click al menú de opciones	
	Seleccionar “elegir color de la paleta”	
		Despliega el prima RGB
	Selecciona el color	
	Selecciona “ok”	
		Aplica el color
		retorna un mensaje confirmando que se ha cambiado el color
		Actualiza la visualización
		La tarea tiene un color asignado visible en el tablero
Post-condiciones	Color visible en el panel	
Excepciones	No tener tareas asignadas	

CASO DE USO 13 - ETIQUETAR TAREAS CON PALABRAS CLAVE

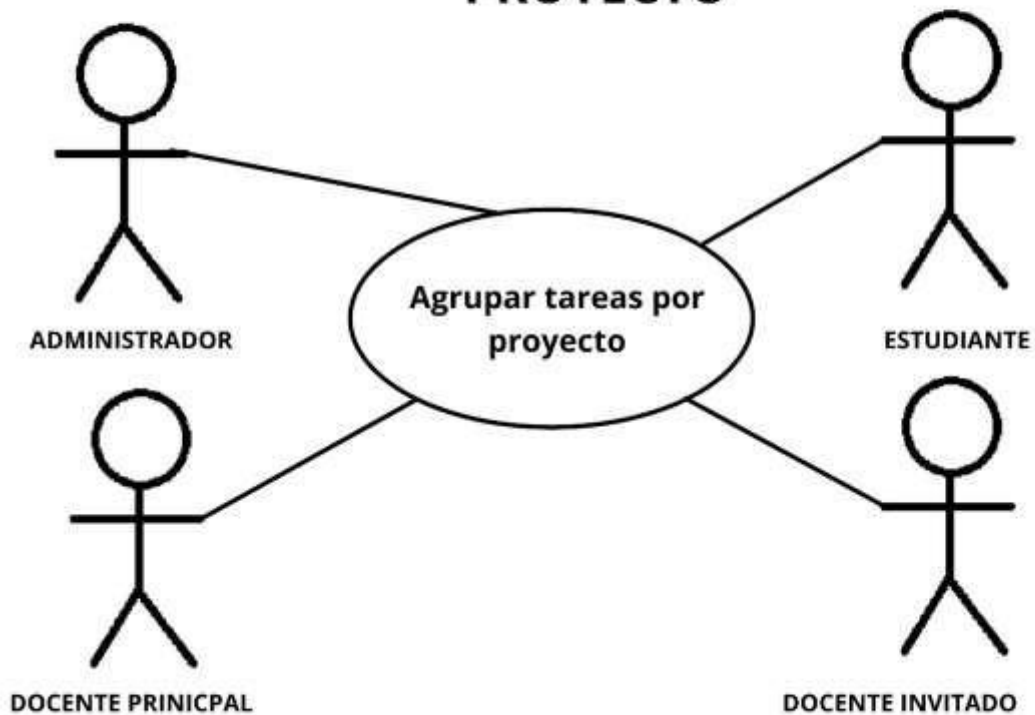


CASO No. 13 Etiquetar tarea con palabras claves

ID:	TA-13
Nombre	Etiquetar tareas con palabras claves
Actores	Administrador, docente principal, docente invitado, estudiante.
Objetivo	Usar colores para asignar tareas
Urgencia	2
Esfuerzo	1

Pre-condiciones	Tener usuario y tarea asignada	
Flujo Normal	Docente	Sistema
	Abre una tarea	
	Selecciona menú contextual	
	Selecciona Agregar Palabras clave	
		Despliega el menú para agregar palabras clave
	Escribe la palabra clave	
	Selecciona ok	
		Aplica el cambio
		Guarda la etiqueta y la muestra junto a la tarea
		Tarea etiquetada y buscable por palabra clave
Post-condiciones	Color visible en el panel	
Excepciones	No tener tareas asignadas	

CASO DE USO 13 - AGRUPAR TAREAS POR PROYECTO

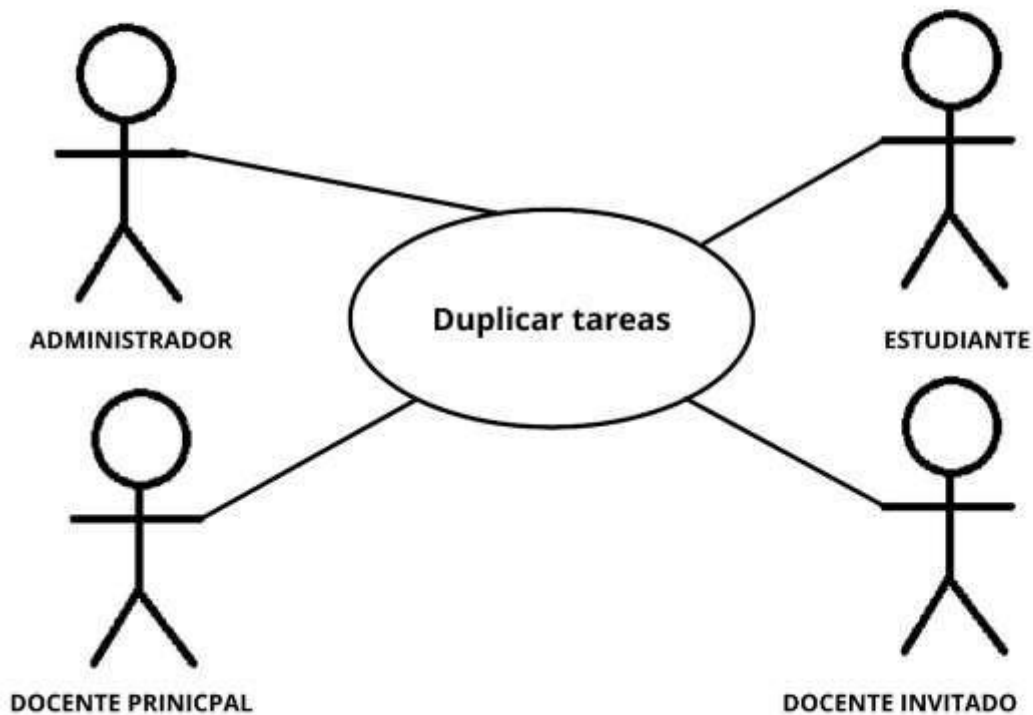


CASO No. 14 Agrupar tareas

ID:	TA-14
Nombre	Agrupar tareas
Actores	Administrador, Docente principal, Docente invitado, Estudiante
Objetivo	Permitir organizar tareas bajo un proyecto específico
Urgencia	3

Esfuerzo	2	
Pre-condiciones	Tener usuario y tarea asignada	
Flujo Normal	Docente	Sistema
	Accede al listado de tareas	
		Muestra todas las tareas disponibles
	Selecciona tareas relacionadas	
		Solicita el nombre del proyecto
	Define nombre del proyecto	
		guarda la relación entre tareas y proyecto
		Retorna que hubo una actualización
		Tareas organizadas bajo el proyecto asignado
Post-condiciones	Las tareas quedan agrupadas bajo el proyecto elegido	
Excepciones	Error al guardar si falla conexión a base de datos	

CASO DE USO 15 - DUPLICAR TAREAS



CASO No. 15 Duplicar Tareas

ID:	TA-15
Nombre	Duplicar Tareas
Actores	Administrador, Docente principal, Docente invitado, Estudiante
Objetivo	Crear una copia exacta de una tarea
Urgencia	2

Esfuerzo	1	
Pre-condiciones	Tarea existente seleccionada	
Flujo Normal	Docente	Sistema
	Selecciona una tarea	
	Click al menú contextual	
	Selecciona la opción “duplicar tareas”	
		Muestra opción de “Duplicar”
	Ajustar campo	
		Guarda la nueva actualización
		Retorna mensaje “ su tarea ha sido duplicada”
		Nueva tarea creada a partir de la anterior
Post-condiciones	Una nueva tarea con datos duplicados es registrada	
Excepciones	Conflicto con ID único, error al guardar	

CASO DE USO 16 - GUARDAR CAMBIOS DE TAREA

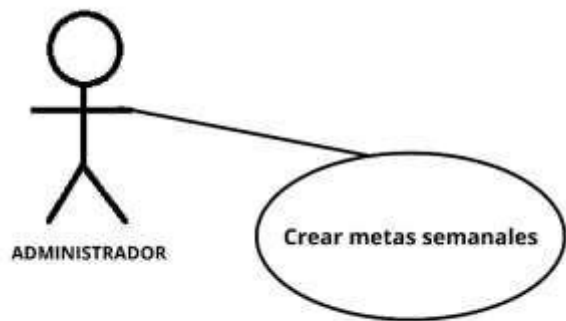


CASO No. 16 Guardar Cambios de Tarea

ID:	TA-16	
Nombre	Guardar Cambios de Tarea	
Actores	Administrador, docente principal	
Objetivo	Almacenar ediciones en una tarea	
Urgencia	3	
Esfuerzo	1	
Pre-condiciones	Tarea existente seleccionada	
Flujo Normal	Docente	Sistema

	Edita los campos de una tarea	
	Presiona “Guardar”	
		guarda en la base de datos
		Retorna mensaje de tarea guardada
		La tarea se actualiza con la nueva información
Post-condiciones	La tarea se actualiza con nueva información	
Excepciones	Validación fallida o error de red	

CASO DE USO 17 - CREAR METAS SEMANALES

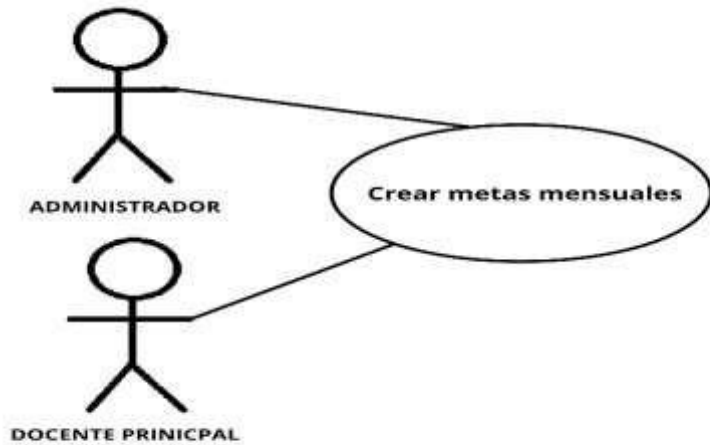


CASO No. 17 Crear Metas Semanales

ID:	TA-17	
Nombre	Crear Metas Semanales	
Actores	Administrador	
Objetivo	Definir objetivos semanales asociados a tareas	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	Se activan Objetivos	
	Docente	Sistema

Flujo Normal	Accede al módulo de metas	
		Muestra opciones de planificación
	Selecciona “Crear meta semanal”	
		Despliega formulario con campos (nombre, descripción, tareas, plazo)
	Llena los campos y selecciona tareas asociadas	
	selecciona tareas asociadas	
		Valida los datos
		registra meta
	Selecciona la opción “guardar meta”	
		Guarda la meta
		Meta semanal visible por proyecto o usuario asignado
Post-condiciones	Meta registrada y visible en el tablero	
Excepciones	Error de validación de fecha o límite de tareas	

CASO DE USO 18 - CREAR METAS MENSUALES



CASO No. 18 Crear Metas Mensuales

ID:	TA-18	
Nombre	Crear Metas Mensuales	
Actores	Administrador, Docente Principal	
Objetivo	Permitir al usuario establecer objetivos mensuales con base en tareas o proy	
Urgencia	3	
Esfuerzo	3	
Pre-condiciones	El usuario debe estar autenticado y tener tareas activas en el sistema.	
Flujo Normal	Docente	Sistema
	Accede al módulo de metas	

		Muestra opciones de metas
	Selecciona “Crear meta mensual”	
		Despliega formulario con campos (nombre, descripción, tareas, plazo)
	Llena los campos	
	selecciona tareas asociadas	
		Valida los datos
		registra meta
	Selecciona la opción “guardar meta”	
		Guarda la meta
		Meta mensual visible por proyecto o usuario asignado
Post-condiciones	La meta mensual queda registrada y asociada al usuario o proyecto correspondiente.	
Excepciones	Campos incompletos: no se permite guardar la meta si faltan datos obligatorios.	

CASO DE USO 19- CALIFICAR TAREAS COMPLETADAS

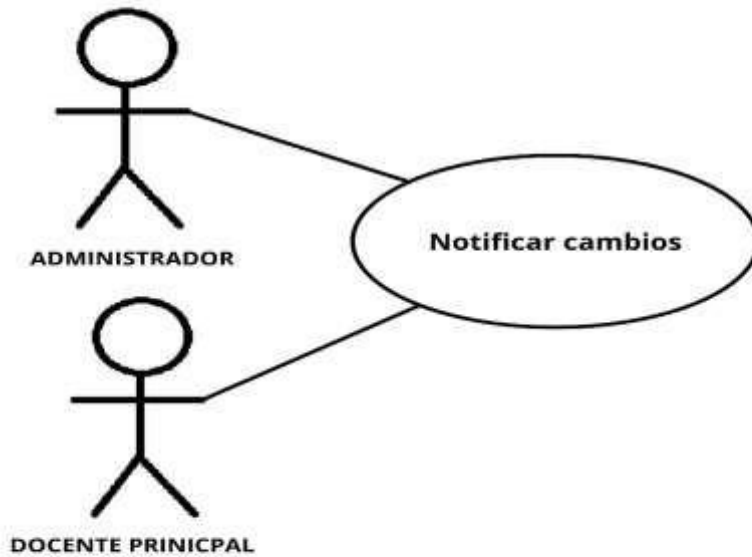


CASO No. 19 Calificar Tareas Completadas.

ID:	TA-19
Nombre	Calificar tareas completadas.
Actores	Docente Principal, Docente Invitado
Objetivo	Permitir a los docentes asignar una calificación a las tareas entregadas.
Urgencia	4
Esfuerzo	3
Pre-condiciones	El docente debe haber iniciado sesión en el sistema. La tarea debe estar en estado de completada.

Flujo Normal	Docente	Sistema
	Accede al módulo de revisión	
		Lista tareas completadas por estudiantes
	Selecciona una tarea para calificar	
		Muestra detalle de entrega
	Ingresar calificación (numérica o cualitativa)	
		Valida la nota
		actualiza la base de datos
		Retorna un mensaje de validación
		Calificación reflejada en el historial del estudiante
Post-condiciones	Las metas mensuales aparecen en vista calendario	
Excepciones	Error de validación: calificación fuera del rango permitido.	

CASO DE USO 20 - NOTIFICAR CAMBIOS



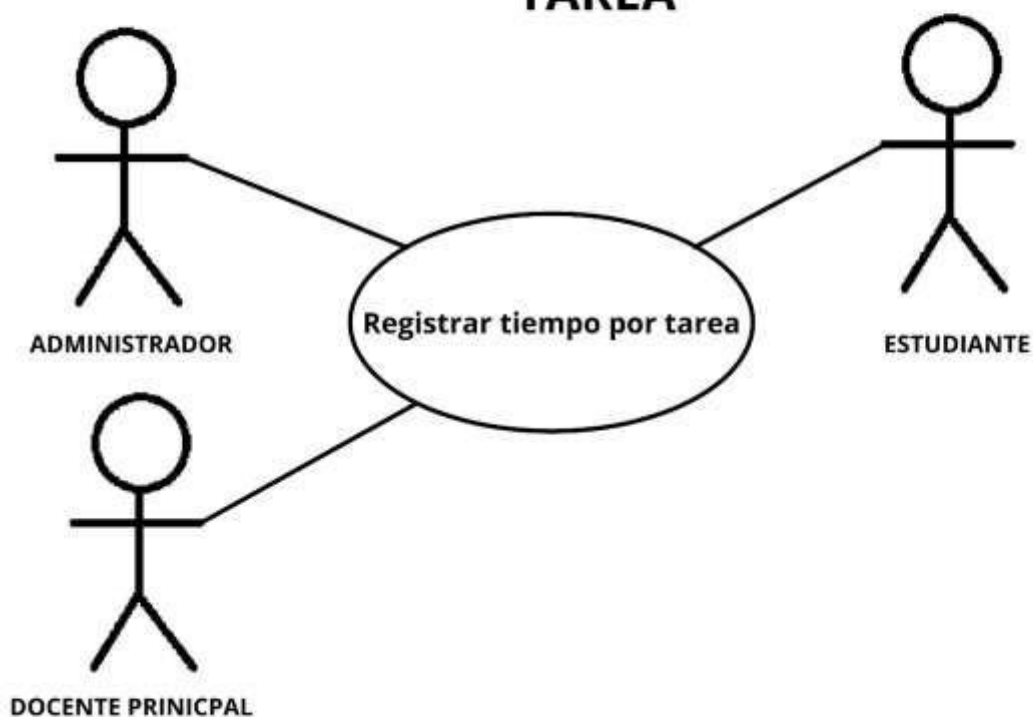
CASO No. 20 Notificar Cambios

ID:	TA-20	
Nombre	Notificar Cambios	
Actores	Administrador, Docente Principal	
Objetivo	Informar a los usuarios involucrados sobre modificaciones	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	El usuario debe estar autenticado y tener permisos para modificar tareas.	
Flujo Normal	Docente	Sistema
	Realiza una modificación en una tarea	
		Detecta el cambio usando triggers
		Guarda nueva versión y genera registro de cambio
		Genera notificación para los usuarios asignados.
Post-condiciones	Los usuarios asignados reciben una notificación clara y registrada del cambio.	

realizadas en una tarea.

Excepciones	Falla en el envío: el servidor de notificaciones no responde
-------------	--

CASO DE USO 21 - REGISTRAR TIEMPO POR TAREA

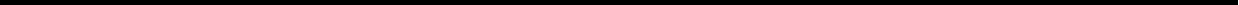
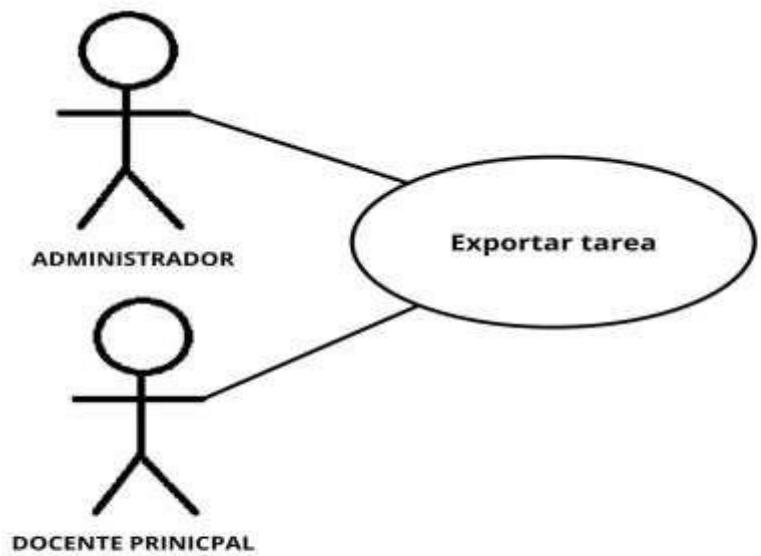


CASO No. 21 Registrar Tiempo por Tarea

ID:	TA-21
Nombre	Registrar Tiempo por Tarea
Actores	Docente Principal, Docente Invitado

Objetivo	Permitir que el docente registre el tiempo invertido en una tarea específica.	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	El usuario debe estar autenticado y haber accedido a una tarea asignada.	
Flujo Normal	Docente	Sistema
	Accede al módulo de tareas asignadas.	
		Muestra la lista de tareas con opción de registrar tiempo.
	Selecciona la tarea correspondiente.	
		Carga la información detallada de la tarea.
	Ingresa el tiempo trabajado (horas, minutos).	
		Valida el formato y guarda el tiempo.
Post-condiciones	El sistema guarda el tiempo registrado y lo asocia a la tarea correspondiente.	
Excepciones	Formato de tiempo inválido: el tiempo ingresado no cumple con el formato esperado.	

CASO DE USO 22 - EXPORTAR TAREA

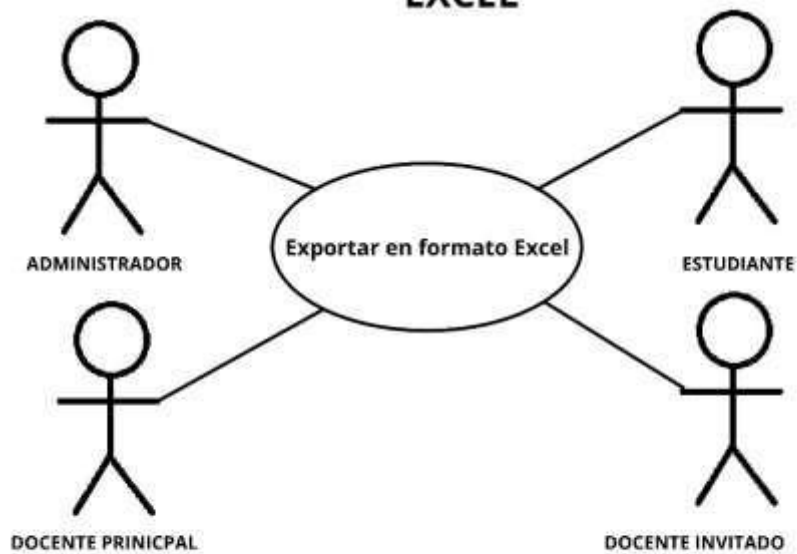


CASO No. 22 Exportar Tarea

ID:	TA-22
Nombre	Exportar tarea
Actores	Administrador, Docente Principal
Objetivo	Permitir que el docente registre el tiempo invertido en una tarea especí
Urgencia	3

Esfuerzo	3	
Pre-condiciones	El usuario debe estar autenticado y tener tareas disponibles para exportar.	
Flujo Normal	Docente	Sistema
	Accede al módulo de tareas. Accede al módulo de tareas.	
		Muestra la lista de tareas disponibles.
	Selecciona las tareas que desea exportar.	
		Habilita opciones de formato de exportación (PDF, Excel, JSON, etc.).
	Escoge el formato de exportación.	
		Genera el archivo en el formato seleccionado.
Post-condiciones	El archivo exportado queda disponible para descarga en el formato seleccionado.	
Excepciones	Error de formato: el sistema no puede generar el archivo debido a una incompatibilidad.	

CASO DE USO 24 - EXPORTAR EN FORMATO EXCEL

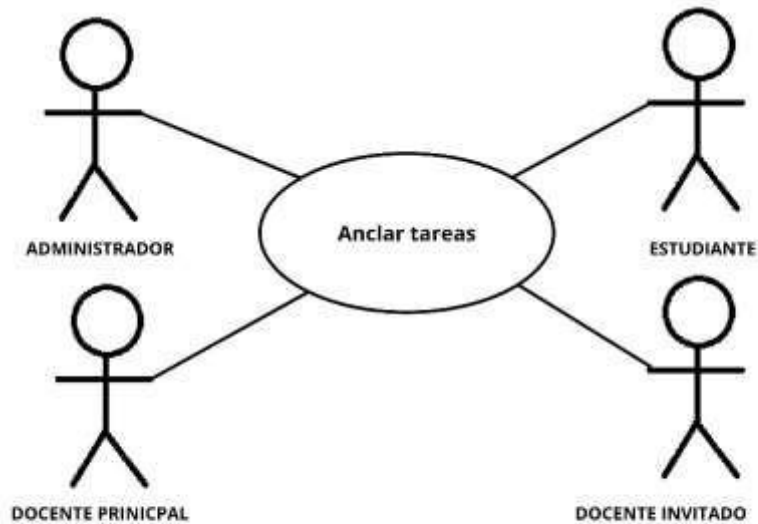


CASO No. 24 Exportar en Formato Excel

ID:	TA-24
Nombre	Exportar en formato Excel
Actores	Administrador, Docente Principal
Objetivo	Permitir exportar las tareas o proyectos a un archivo de hoja de cálculo
Urgencia	3
Esfuerzo	3

Pre-condiciones	El usuario debe estar autenticado y tener tareas disponibles en el sistema.	
Flujo Normal	Docente	Sistema
	Accede al módulo de exportación de tareas.	
		Muestra la lista de tareas con opción para exportar.
	Selecciona "Exportar en formato Excel.	
		Genera el archivo PDF con los datos de la tarea.
	Confirma la exportación.	
		Procesa los datos y genera un archivo.
Post-condiciones	El archivo Excel se genera con éxito y queda disponible para descarga o posterior edición.	
Excepciones	Error en formato de datos: algunos campos no son compatibles con la estructura de hoja de cálculo.	

CASO DE USO 25 - ANCLAR TAREAS



CASO No. 25 Anclar Tareas

ID:	TA-25
Nombre	Anclar tareas
Actores	Administrador, Docente Principal
Objetivo	Permitir al usuario destacar tareas importantes fijándolas en la parte superior del panel de tareas.
Urgencia	2
Esfuerzo	2
Pre-condiciones	

	El usuario debe estar autenticado y tener tareas asignadas en el sistema.	
Flujo Normal	Docente	Sistema
	Accede al panel de tareas.	Muestra las tareas activas.
	Selecciona la tarea que desea anclar.	
		Habilita opción “Anclar tarea”
	Hace clic en Anclar tarea.	La tarea se ancla. La tarea se fija en la parte superior del panel.
Post-condiciones	La tarea aparece anclada en la vista principal hasta que se desancla manualmente.	
Excepciones	Tarea ya anclada: el sistema impide anclar una tarea que ya se encuentra en estado fijo.	

Prioridad de Requerimientos

4. Requisitos No Funcionales

		Urgencia				
Esfuerzo		1-Baja	2-Menor	3-Moderada	4-Alta	5-Obligatoria
	5-Muy alto					
	4-Alto					TA-1
	3-Medio			TA-5 TA-9 TA-18 TA-22 TA-24	TA-8 TA-10 TA-11 TA-19 TA-21	
	2-Bajo		TA-25 TA-6 TA-25	TA-3 TA-4 TA-9 TA-14 TA-17 TA-20 TA-23	TA-2 TA-7	
	1-Muy bajo		TA-12 TA-13 TA-15	TA-16		

Seguridad:

Gestión de roles y permisos: solo usuarios autorizados podrán acceder o modificar tareas según su perfil (Administrador, Docente, Estudiante).

Rendimiento:

Manual o guía interactiva para usuarios nuevos accesible desde el sistema. Accesos rápidos a las funciones más usadas desde el panel principal.

Escalabilidad:

Diseño de base de datos preparado para manejar miles de tareas, usuarios y comentarios sin degradar el rendimiento.

Disponibilidad:

Capacidad de recuperación ante fallos mediante backups automáticos diarios y restauración bajo demanda.

Compatibilidad con Dispositivos:

Compatible con sistemas operativos Windows, macOS, Android y iOS.

Accesibilidad:

Uso de etiquetas semánticas y navegación compatible con lectores de pantalla.

Cumplimiento Normativo: Cumplimiento con normativas de protección de datos personales

Tiempo de Respuesta:

Operaciones CRUD deben tener una latencia menor a 300 ms en condiciones normales.

Requisitos de Desempeño

Tiempo de respuesta:

El sistema deberá procesar y responder a las acciones de los usuarios (crear, editar, eliminar, recuperar tareas) en un tiempo inferior a 2 segundos bajo condiciones normales de operación.

Carga de interfaz:

La carga inicial del tablero principal no deberá superar los 5 segundos, incluso cuando existan más de 100 tareas activas.

Procesamiento de tareas masivas:

Operaciones como exportar tareas a PDF o Excel, importar archivos o sincronizar múltiples tareas deberán completarse en menos de 10 segundos en condiciones óptimas de red y procesamiento.

Requisitos de Seguridad

Autenticación y Autorización:

El acceso al sistema debe estar protegido mediante un sistema de autenticación segura con JSON. El sistema debe implementar control de acceso basado en roles (RBAC), diferenciando permisos según el perfil del usuario

(Administrador, Docente Principal, Docente Invitado, Estudiante).

Cifrado de Datos en Tránsito:

Toda la comunicación entre cliente y servidor debe utilizar el protocolo HTTPS con certificados SSL/TLS válidos.

Protección de Datos Sensibles:

Los datos sensibles (como información personal o historial de comentarios) deben protegerse contra accesos no autorizados.

Gestión de Sesiones:

Las sesiones de usuario deben expirar automáticamente después de un período de inactividad prolongado (ej. 30 minutos).

Auditoría y Registro de Actividades:

El sistema debe mantener un registro (log) de eventos importantes como: creación/eliminación de tareas, calificaciones asignadas, exportaciones de datos, e inicio/cierre de sesión.

Requisitos de Usabilidad

Interfaz Intuitiva y Amigable:

La plataforma debe contar con una interfaz gráfica clara, ordenada y fácil de usar para todos los perfiles de usuario (docentes, estudiantes, administradores).

Curva de Aprendizaje Reducida:

El sistema debe permitir a un nuevo usuario comprender y utilizar las funciones básicas sin requerir capacitación extensa.

Retroalimentación Inmediata:

Cada acción realizada (guardar, eliminar, comentar, exportar, etc.) debe devolver una respuesta visual clara de notificaciones, mensajes de éxito/error, indicadores de carga.

Requisitos de Escalabilidad

Escalabilidad en la Base de Datos:

El sistema debe estar preparado para manejar grandes volúmenes de datos (tareas, comentarios, usuarios, archivos) utilizando bases de datos SQL o NoSQL escalables, y permitir la partición o replicación si el volumen crece significativamente.

Optimización de Recursos en Escenarios de Alta Carga:

En entornos con alta demanda, el sistema debe ser capaz de realizar caché inteligente de tareas consultadas frecuentemente y uso de técnicas de lazy loading para optimizar la carga de contenido.

Escalabilidad Horizontal en Backend:

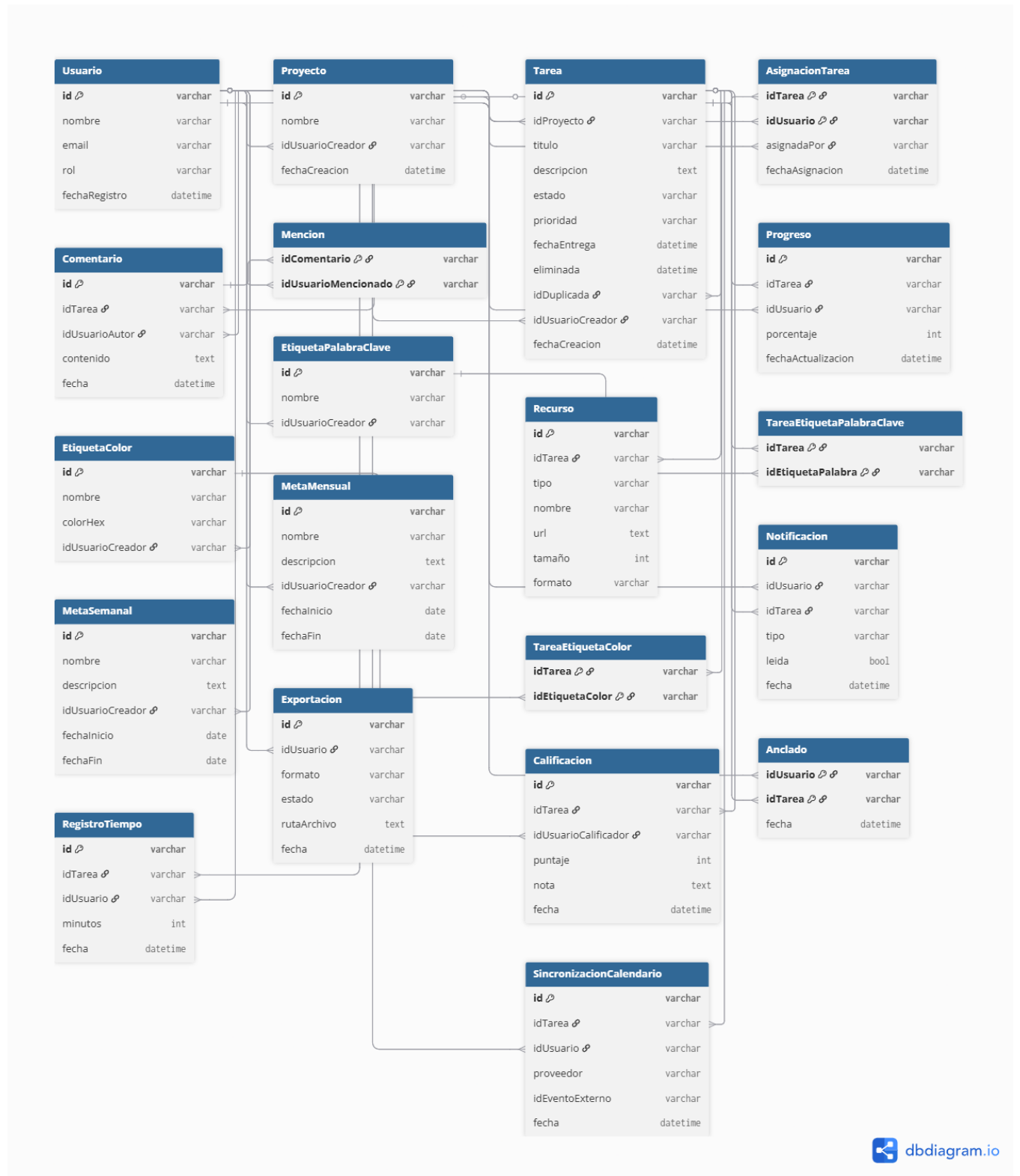
La arquitectura del sistema debe permitir la incorporación de múltiples instancias del servidor backend (Node.js, Express, etc.) para distribuir la carga de usuarios concurrentes mediante balanceadores de carga.

5. Modelado E/R

El sistema TaskAssistant se modela considerando las entidades principales que intervienen en la gestión de tareas colaborativas: usuarios, tareas, proyectos, recursos, comentarios y metas. Las relaciones permiten organizar, asignar y dar seguimiento en tiempo real al trabajo distribuido en equipos.

Diagrama de Entidad-Relación

Diagrama Relacional



Script de modelo relacional

<https://dbdiagram.io/d/68db5a20d2b621e42282895c>

```
Table Usuario {  
  id varchar [pk]  
  nombre varchar  
  email varchar [unique]  
  rol varchar // administrador, docente principal, docente invitado, estudiante  
  fechaRegistro datetime  
}
```

```
Table Proyecto {  
  id varchar [pk]  
  nombre varchar  
  idUsuarioCreador varchar [ref: > Usuario.id]  
  fechaCreacion datetime  
}
```

```
Table Tarea {  
  id varchar [pk]  
  idProyecto varchar [ref: > Proyecto.id]  
  titulo varchar  
  descripcion text  
  estado varchar // nueva, en_progreso, completada, archivada  
  prioridad varchar // baja, media, alta, urgente  
  fechaEntrega datetime  
  eliminada datetime // recuperación  
  idDuplicada varchar [ref: > Tarea.id]  
  idUsuarioCreador varchar [ref: > Usuario.id]  
  fechaCreacion datetime  
}
```

```
Table AsignacionTarea {  
  idTarea varchar [ref: > Tarea.id, pk]  
  idUsuario varchar [ref: > Usuario.id, pk]  
  asignadaPor varchar [ref: > Usuario.id]  
  fechaAsignacion datetime  
}
```

```
Table Comentario {  
  id varchar [pk]  
  idTarea varchar [ref: > Tarea.id]  
  idUsuarioAutor varchar [ref: > Usuario.id]  
  contenido text  
  fecha datetime  
}
```

```
Table Mencion {  
  idComentario varchar [ref: > Comentario.id, pk]
```

```
idUsuarioMencionado varchar [ref: > Usuario.id, pk]
}
```

```
Table Recurso {
  id varchar [pk]
  idTarea varchar [ref: > Tarea.id]
  tipo varchar // pdf, img, video, link, otro
  nombre varchar
  url text
  tamaño int
  formato varchar
}
```

```
Table Progreso {
  id varchar [pk]
  idTarea varchar [ref: > Tarea.id]
  idUsuario varchar [ref: > Usuario.id]
  porcentaje int
  fechaActualizacion datetime
}
```

```
Table EtiquetaColor {
  id varchar [pk]
  nombre varchar
  colorHex varchar
  idUsuarioCreador varchar [ref: > Usuario.id]
}
```

```
Table EtiquetaPalabraClave {
  id varchar [pk]
  nombre varchar
  idUsuarioCreador varchar [ref: > Usuario.id]
}
```

```
Table TareaEtiquetaColor {
  idTarea varchar [ref: > Tarea.id, pk]
  idEtiquetaColor varchar [ref: > EtiquetaColor.id, pk]
}
```

```
Table TareaEtiquetaPalabraClave {
  idTarea varchar [ref: > Tarea.id, pk]
  idEtiquetaPalabra varchar [ref: > EtiquetaPalabraClave.id, pk]
}
```

```
Table MetaSemanal {
  id varchar [pk]
  nombre varchar
  descripcion text
  idUsuarioCreador varchar [ref: > Usuario.id]
```

```
    fechaInicio date
    fechaFin date
}
```

```
Table MetaMensual {
    id varchar [pk]
    nombre varchar
    descripcion text
    idUsuarioCreador varchar [ref: > Usuario.id]
    fechaInicio date
    fechaFin date
}
```

```
Table Calificacion {
    id varchar [pk]
    idTarea varchar [ref: > Tarea.id]
    idUsuarioCalificador varchar [ref: > Usuario.id]
    puntaje int
    nota text
    fecha datetime
}
```

```
Table Notificacion {
    id varchar [pk]
    idUsuario varchar [ref: > Usuario.id]
    idTarea varchar [ref: > Tarea.id]
    tipo varchar // mencion, cambio, asignacion, recordatorio
    leida bool
    fecha datetime
}
```

```
Table RegistroTiempo {
    id varchar [pk]
    idTarea varchar [ref: > Tarea.id]
    idUsuario varchar [ref: > Usuario.id]
    minutos int
    fecha datetime
}
```

```
Table Exportacion {
    id varchar [pk]
    idUsuario varchar [ref: > Usuario.id]
    formato varchar // pdf, excel
    estado varchar // pendiente, completada, fallida
    rutaArchivo text
    fecha datetime
}
```

```
Table SincronizacionCalendario {
```



```

id varchar [pk]
idTarea varchar [ref: > Tarea.id]
idUsuario varchar [ref: > Usuario.id]
proveedor varchar // google, outlook, otro
idEventoExterno varchar
fecha datetime
}

```

```

Table Anclado {
idUsuario varchar [ref: > Usuario.id, pk]
idTarea varchar [ref: > Tarea.id, pk]
fecha datetime
}

```

Descripción de Entidades y Relaciones

1) Usuario

Propósito: personas que usan el sistema (administrador, docente principal, docente invitado, estudiante).

Atributos clave: id (PK), nombre, email (único), rol, fechaRegistro.

Relaciones:

- Crea Proyectos y Tareas. (Usuario 1 — N Proyecto / Tarea)
- Asigna tareas vía AsignacionTarea. (Usuario 1 — N AsignacionTarea)
- Es autor de Comentarios. (Usuario 1 — N Comentario)
- Puede ser mencionado en Mencion. (Usuario 1 — N Mencion)
- Registra Progreso y RegistroTiempo sobre tareas. (Usuario 1 — N Progreso / RegistroTiempo)
- Crea Etiquetas y Metas. (Usuario 1 — N EtiquetaColor / EtiquetaPalabraClave / MetaSemanal / MetaMensual)
- Recibe Notificaciones, solicita Exportaciones, configura SincronizacionCalendario y Anclados. (Usuario 1 — N Notificacion / Exportacion / SincronizacionCalendario / Anclado)

2) Proyecto

Propósito: agrupa tareas por iniciativa o contexto.

Atributos clave: id (PK), nombre, idUsuarioCreador (FK), fechaCreacion.

Relaciones:

- Contiene Tareas. (Proyecto 1 — N Tarea)

3) Tarea

Propósito: ítem de trabajo gestionado en la plataforma.

Atributos clave: id (PK), idProyecto (FK), titulo, descripcion, estado, prioridad, fechaEntrega, eliminada (soft delete), idDuplicada (FK a Tarea), idUsuarioCreador (FK), fechaCreacion.

Relaciones:

- Tiene Comentarios, Recursos, Progreso, RegistroTiempo, Notificaciones, Exportaciones, SincronizacionCalendario, Calificaciones, Anclados. (Tarea 1 — N ...)
- Se etiqueta por color/palabras con tablas puente TareaEtiquetaColor y TareaEtiquetaPalabraClave. (Tarea 1 — N puente)
- Puede pertenecer a Metas (semanales/mensuales). (Meta 1 — N Tarea)
- Puede ser duplicada desde otra tarea (self-FK).

4) AsignacionTarea

Propósito: quién está asignado a una tarea.

Atributos clave: idTarea (PK, FK), idUsuario (PK, FK), asignadaPor (FK), fechaAsignacion.

Relaciones:

- Une Usuario y Tarea (N:M resuelto con PK compuesto).

5) Comentario

Propósito: comunicación colaborativa sobre una tarea.

Atributos clave: id (PK), idTarea (FK), idUsuarioAutor (FK), contenido, fecha.

Relaciones:

- Puede incluir Mencion a usuarios. (Comentario 1 — N Mencion)

6) Mencion

Propósito: referencia directa a un usuario dentro de un comentario.

Atributos clave: idComentario (PK, FK), idUsuarioMencionado (PK, FK).

Relaciones:

- Apunta a Comentario y Usuario (PK compuesta).

7) Recurso

Propósito: archivos o enlaces asociados a una tarea.

Atributos clave: id (PK), idTarea (FK), tipo, nombre, url, tamaño, formato.

Relaciones:

- Cada Recurso pertenece a una Tarea. (Tarea 1 — N Recurso)

8) Progreso

Propósito: avance porcentual por usuario en una tarea.

Atributos clave: id (PK), idTarea (FK), idUsuario (FK), porcentaje, fechaActualizacion.

Relaciones:

- Registrado por Usuario sobre Tarea. (Usuario 1 — N Progreso; Tarea 1 — N Progreso)

9) EtiquetaColor

Propósito: taxonomía de color para clasificar tareas.

Atributos clave: id (PK), nombre, colorHex, idUsuarioCreador (FK).

Relaciones:

- Se usa vía TareaEtiquetaColor. (EtiquetaColor 1 — N TareaEtiquetaColor)

10) EtiquetaPalabraClave

Propósito: keywords para clasificación.

Atributos clave: id (PK), nombre, idUsuarioCreador (FK).

Relaciones:

- Se usa vía TareaEtiquetaPalabraClave. (EtiquetaPalabraClave 1 — N TareaEtiquetaPalabraClave)

11) TareaEtiquetaColor

Propósito: unión N:M entre Tarea y EtiquetaColor.

Atributos clave: idTarea (PK, FK), idEtiquetaColor (PK, FK).

12) TareaEtiquetaPalabraClave

Propósito: unión N:M entre Tarea y EtiquetaPalabraClave.

Atributos clave: idTarea (PK, FK), idEtiquetaPalabra (PK, FK).

13) MetaSemanal

Propósito: objetivos a los que se adscriben tareas.

Atributos clave: id (PK), nombre, descripcion, idUsuarioCreador (FK), fechaInicio, fechaFin.

Relaciones:

- Cada Meta agrupa varias Tareas. (Meta 1 — N Tarea)

14) 14) MetaMensual

Propósito: objetivos a los que se adscriben tareas.

Atributos clave: id (PK), nombre, descripcion, idUsuarioCreador (FK), fechaInicio, fechaFin.

Relaciones:

- Cada Meta agrupa varias Tareas. (Meta 1 — N Tarea)

15) Calificacion

Propósito: valoración de tareas completadas.

Atributos clave: id (PK), idTarea (FK), idUsuarioCalificador (FK), puntaje, nota, fecha.

Relaciones:

- Hecha por Usuario sobre Tarea. (Usuario 1 — N Calificacion; Tarea 1 — N Calificacion)

16) Notificacion

Propósito: avisos a usuarios (mención, cambio, asignación, recordatorio).

Atributos clave: id (PK), idUsuario (FK), idTarea (FK), tipo, leida, fecha.

Relaciones:

- Usuario recibe; Tarea genera. (Usuario 1 — N Notificacion; Tarea 1 — N Notificacion)

17) RegistroTiempo

Propósito: minutos dedicados por usuario a una tarea.

Atributos clave: id (PK), idTarea (FK), idUsuario (FK), minutos, fecha.

Relaciones:

- Usuario registra en Tarea. (Usuario 1 — N RegistroTiempo; Tarea 1 — N RegistroTiempo)

18) Exportacion

Propósito: exportes solicitados (PDF/Excel) y su estado.

Atributos clave: id (PK), idUsuario (FK), formato, estado, rutaArchivo, fecha.

Relaciones:

- Usuario solicita; puede incluir tareas en el archivo. (Usuario 1 — N Exportacion)

19) SincronizacionCalendario

Propósito: vínculo con eventos externos (Google/Outlook).

Atributos clave: id (PK), idTarea (FK), idUsuario (FK), proveedor, idEventoExterno, fecha.

Relaciones:

- Usuario configura; Tarea se sincroniza. (Usuario 1 — N / Tarea 1 — N)

20) Anclado

Propósito: “pineado” de tareas por usuario.

Atributos clave: idUsuario (PK, FK), idTarea (PK, FK), fecha.

Relaciones:

- N:M entre Usuario y Tarea (PK compuesta).

Reglas de Integridad Referencial

1. **Proyecto–Usuario (creador):** Proyecto.idUsuarioCreador debe existir en Usuario.id.
2. **Tarea–Proyecto:** Tarea.idProyecto debe existir en Proyecto.id.
3. **Tarea–Usuario (creador):** Tarea.idUsuarioCreador debe existir en Usuario.id.
4. **Tarea–Tarea (duplicada):** Tarea.idDuplicada (si no es nulo) debe existir en Tarea.id.
5. **AsignacionTarea–Tarea/Usuario:** (idTarea, idUsuario) deben existir en Tarea.id y Usuario.id; asignadaPor debe existir en Usuario.id.
6. **Comentario–Tarea/Usuario:** Comentario.idTarea en Tarea.id y Comentario.idUsuarioAutor en Usuario.id.
7. **Mencion–Comentario/Usuario:** (idComentario, idUsuarioMencionado) deben existir en Comentario.id y Usuario.id.
8. **Recurso–Tarea:** Recurso.idTarea debe existir en Tarea.id.
9. **Progreso–Tarea/Usuario:** Progreso.idTarea en Tarea.id y Progreso.idUsuario en Usuario.id.
10. **EtiquetaColor–Usuario:** EtiquetaColor.idUsuarioCreador debe existir en Usuario.id.
11. **EtiquetaPalabraClave–Usuario:** EtiquetaPalabraClave.idUsuarioCreador debe existir en Usuario.id.
12. **TareaEtiquetaColor–Tarea/EtiquetaColor:** (idTarea, idEtiquetaColor) deben existir en Tarea.id y EtiquetaColor.id.
13. **TareaEtiquetaPalabraClave–Tarea/EtiquetaPalabraClave:** (idTarea, idEtiquetaPalabra) deben existir en Tarea.id y EtiquetaPalabraClave.id.
14. **MetaSemanal–Usuario:** MetaSemanal.idUsuarioCreador debe existir en Usuario.id.
15. **MetaMensual–Usuario:** MetaMensual.idUsuarioCreador debe existir en Usuario.id.
16. **Calificacion–Tarea/Usuario:** Calificacion.idTarea en Tarea.id y Calificacion.idUsuarioCalificador en Usuario.id.
17. **Notificacion–Usuario/Tarea:** Notificacion.idUsuario en Usuario.id y Notificacion.idTarea (si aplica) en Tarea.id.
18. **RegistroTiempo–Tarea/Usuario:** RegistroTiempo.idTarea en Tarea.id y RegistroTiempo.idUsuario en Usuario.id.

19. **Exportacion–Usuario:** Exportacion.idUsuario debe existir en Usuario.id.
20. **SincronizacionCalendario–Usuario/Tarea:** SincronizacionCalendario.idUsuario en Usuario.id y SincronizacionCalendario.idTarea en Tarea.id.
21. **Anclado–Usuario/Tarea:** (Anclado.idUsuario, Anclado.idTarea) deben existir en Usuario.id y Tarea.id.

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.