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

Creación de modulo de gestion de tareas (Task Manager)

Alexander Enrique Toscano Ricardo

Twitter: @kikret

Github: @atoscano

Edwin Bertel Negrette

Guillermo del Valle Vitola

German Rivera Rosario

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

1. Introducción

Propósito del Documento

El propósito del módulo de gestión de tareas descrito es mejorar la organización, seguimiento y eficiencia en la ejecución de actividades diarias en plataformas de contenido. A través de una interfaz intuitiva y herramientas colaborativas, el sistema facilita la gestión eficiente de tareas, distribuye responsabilidades de manera automática y optimiza la coordinación dentro de los equipos. Además, se enfoca en mejorar el bienestar de los usuarios mediante el monitoreo de la carga de trabajo y sugerencias de autocuidado, todo con el objetivo de mejorar el rendimiento y la calidad de la gestión de contenidos

Etapa 1.

Definición de requisitos:

Identifica al usuario: ¿Quién usará el Task Manager? ¿Son individuos, equipos, o departamentos?

Define los objetivos: ¿Para qué se usará el Task Manager? ¿Organización, colaboración, control de plazos, etc.?

Recopila las funcionalidades: ¿Qué funciones específicas necesita el Task Manager? Ej: creación de tareas, asignación, seguimiento del progreso, recordatorios, colaboración, priorización, etc.

Define los flujos de trabajo: ¿Cómo se utilizará el Task Manager en el día a día?

Determina restricciones: ¿Hay limitaciones de hardware, software, tiempo o presupuesto?

Etapa 2

Diseño conceptual:

Crea una estructura de datos: Define las tablas y campos necesarios para almacenar la información de las tareas, usuarios, proyectos, etc.

Diseña la interfaz de usuario: Crea prototipos de la interfaz, considerando la experiencia de usuario (UX), la accesibilidad y la usabilidad.

Escribe la lógica del sistema: Define los procesos que se ejecutarán en el backend para gestionar las tareas, las notificaciones, los permisos, etc.
Planifica la arquitectura del sistema: Decide la tecnología a usar (lenguajes de programación, bases de datos, framework), el modelo de desarrollo (ágil, cascada), etc.

Etapa 3.

Diseño detallado:
Define la arquitectura de software: Elige las tecnologías y componentes específicos a usar.
Diseña la base de datos: Define las tablas, campos, relaciones y tipos de datos.
Crea diagramas de flujo: Define la lógica de los procesos principales del sistema.
Diseña las API: Define las interfaces para la comunicación entre el frontend y el backend.
Diseña las pruebas: Planifica los casos de prueba para asegurar la calidad del software

Alcance del Proyecto Módulo de Pizarra Compartida

Alcance:

Funcionalidades:

- Creación de tareas
- Edición de tareas
- Eliminación de Tareas
- Edición de entrega de tareas
- Agregar comentario en Tareas
- Distribución de tareas
- Priorización de tareas
- Monitoreo de Ta
- Asignación de tareas
- Asignar recursos

Colaboración:

- Comentar tareas y discutirlas con otros usuarios.
- Asignación de roles y permisos a los usuarios.
- Creación de equipos y listas de tareas compartidas.
- Integración con plataformas de comunicación (Slack, Microsoft Teams, etc.).

Reportes:

- Visualización de estadísticas sobre el progreso de las tareas.
- Filtrado y ordenamiento de tareas por distintos criterios.
- Generación de informes personalizados.

Integración:

- Integración con otras aplicaciones (CRM, ERP, etc.).
- Importación y exportación de datos.
- API para permitir la integración con otros sistemas.

Seguridad:

- Gestión de usuarios y permisos.
- Control de acceso a las tareas y datos.

- Encriptación de datos sensibles.

2. Descripción General

Objetivos del Sistema

El objetivo principal del sistema de gestión de tareas es optimizar la organización, eficiencia y colaboración en la ejecución de proyectos y tareas. Busca facilitar la creación, gestión y seguimiento de tareas, automatizar recordatorios y notificaciones, permitir la comunicación y colaboración entre miembros de un equipo, y brindar una visión general del progreso de los proyectos. En resumen, el sistema aspira a mejorar la productividad y el éxito en la realización de proyectos al centralizar la información, agilizar el flujo de trabajo y fomentar el trabajo en equipo

Funcionalidad General

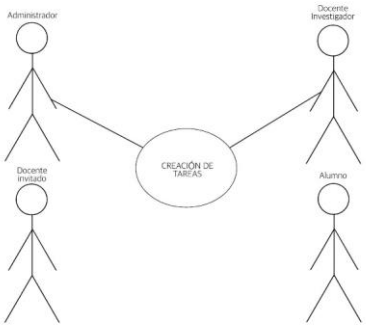
- **Gestión de tareas:** Esta es la funcionalidad principal y engloba todas las acciones relacionadas con el ciclo de vida de una tarea. Incluye la creación de nuevas tareas (especificando título, descripción, fecha de inicio, fecha de vencimiento, prioridad, asignación a usuarios, etc.), la edición de tareas existentes (modificando cualquier parámetro previamente establecido), y la eliminación de tareas completadas o innecesarias. En esencia, abarca todo el proceso desde la concepción de una tarea hasta su finalización.
- **Estado:** Se refiere al estado actual de cada tarea. Esto podría representarse con un campo que indica si la tarea está: *Pendiente*, *En progreso*, *Bloqueada*, *Completada*, *Revisando*, etc. Esta información proporciona una visión rápida del progreso general del proyecto o de las tareas individuales, facilitando la identificación de cuellos de botella o tareas retrasadas. El estado puede cambiar a medida que avanza el trabajo en la tarea.
- **Tareas activas:** Esta funcionalidad mostraría un filtro o vista específica que solo muestra las tareas que actualmente están "en progreso" o que no se han marcado como completadas. Esto facilita la identificación de las tareas que requieren atención inmediata o que están actualmente en ejecución. Se contrapone a una vista que muestre todas las tareas (incluidas las completadas o las pendientes).
- **Límite (o Límite de Tiempo):** Se refiere al plazo o fecha límite de finalización de una tarea. Es un componente crítico para la gestión del tiempo y la prevención de retrasos. El Task Manager debería permitir establecer un límite de tiempo para cada tarea y, opcionalmente, proporcionar alertas o recordatorios cuando se acerca la fecha límite o cuando se supera.
- **Historial:** Esta funcionalidad registraría todas las modificaciones realizadas en una tarea. Podría incluir cambios en el estado, la fecha de vencimiento, la asignación de usuarios, la descripción, etc. El historial permite rastrear la evolución de una tarea, comprender por qué ha habido retrasos o cambios de planes, y facilita la auditoría del proceso de trabajo. Esto es útil para la gestión de proyectos y para la resolución de problemas.

Usuarios del Sistema

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

Funcionalidad	Administradores	Docente Investigador	Docente Invitado	Alumno
Creación de tareas	✓	✓		
Edición de tareas	✓	✓		
Eliminación de Tareas	✓	✓		
Edición de entrega de tareas	✓	✓		✓
Agregar comentario en Tareas	✓	✓	✓	✓
Distribución de tareas	✓	✓		
Priorización de tareas	✓	✓		
Monitoreo de Tarea	✓	✓	✓	✓
Asignación de tareas	✓	✓		
Asignar recursos	✓	✓		
Filtrar tareas	✓	✓	✓	✓


Creación de tareas
Edición de tareas
eliminacion de tarea
Edicion de entrega de tarea
agregar comentarios en tareas
Distribucion de Tareas
Priorizacion de tareas
Monitoreo de tarea
Asignacion de tarea
Asignar recurso
Filtrar tareas
administradores
Docente investigador
alumno
docente invitado



Descripción detallada de cada caso de uso

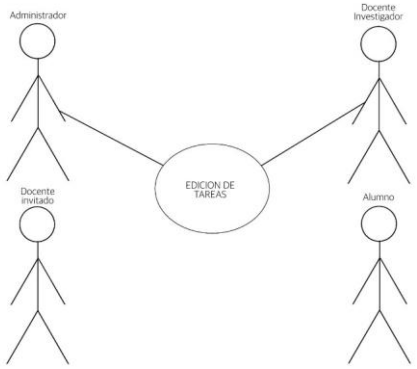
CASO No. 1 Creación de Tareas

ID:	F-ADM-1	
Nombre	Creacion de tareas	
Actores	Administrador, Docente investigador	
Objetivo	Permitir al administrador crear nuevas tareas.	
Urgencia	5	
Esfuerzo	4	
Pre-condiciones	El administrador y docente investigador debe estar autenticado en el sistema.	
Flujo Normal	Docente	Sistema
	Selecciona "Crear Tarea"	Muestra el formulario para crear tarea
	Completa la información de la tarea	Valida los datos y guarda la tarea
	Selecciona tarea en estado borrado	Permite editar y confirmar la tarea
Flujo alternativo 1	Revisa borrador de tarea	Permite editar y confirmar la tarea
Flujo alternativo 2		
Post-condiciones	La tarea queda registrada en el sistema.	

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

Exepciones	Error al guardar tarea por datos inválidos.	

Comentado [k2]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



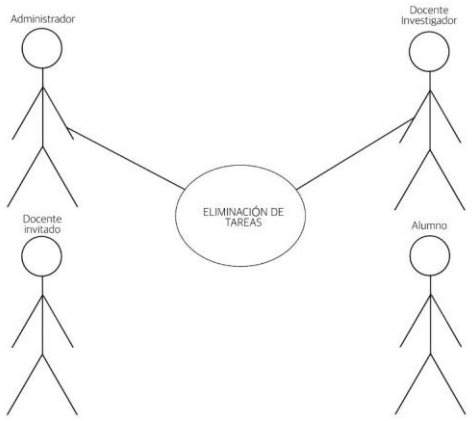
CASO No. 2 Edición de Entrega de Tareas

ID:	F-ADM-2	
Nombre	Edicion de tareas	
Actores	Administrador, Docente investigador	
Objetivo	Permitir al docente modificar la fecha o detalles de la entrega de tareas.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	La tarea debe estar en estado "Pendiente".	
Flujo Normal	Docente	Sistema
	Accede a la tarea que requiere edición	Muestra los detalles actuales de entrega
	Accede a la tarea que requiere edición	Guarda las modificaciones
Flujo alternativo 1	Selecciona un grupo de tareas para editar	Aplica los cambios a todas las tareas seleccionadas

Flujo alternativo 2		
Post-condiciones	La nueva fecha o detalles quedan registrados	
Exepciones	Error al guardar cambios por conflictos de agenda	

Comentado [k3]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k4]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



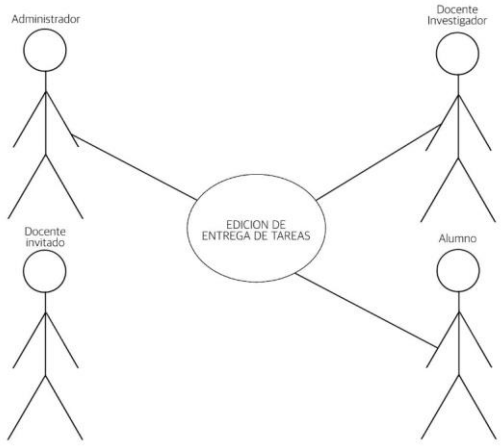
CASO No. 3 Eliminación de Tareas

ID:	F-ADM-3
Nombre	Eliminación de Tareas
Actores	Administrador, Docente investigador
Objetivo	Permitir al administrador eliminar tareas obsoletas o incorrectas

Urgencia	5	
Esfuerzo	3	
Pre-condiciones	El administrador debe estar autenticado en el sistema.	
Flujo Normal	Docente	Sistema
	Selecciona una tarea a eliminar	Muestra confirmación de eliminación
	Presiona "Eliminar"	Borra la tarea del sistema
Flujo alternativo 1	Accede al módulo de tareas eliminadas	Restaura la tarea al estado original
Flujo alternativo 2		
Post-condiciones	La tarea eliminada se elimina de las vistas activas	
Exepciones	Error al guardar comentario por conexión lenta	

Comentado [k5]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k6]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



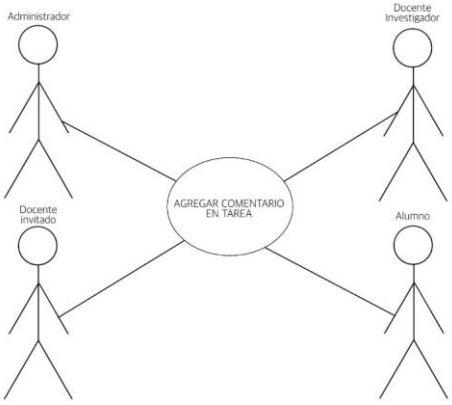
CASO No. 4 Edición de Entrega de Tareas

ID:	F-ADM-4	
Nombre	Edición de Entrega de Tareas	
Actores	Administrador, Docente investigador, alumno	
Objetivo	Permitir al docente modificar la fecha o detalles de la entrega de tareas.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	La tarea debe estar en estado "Pendiente"	
Flujo Normal	Docente	Sistema
	Accede a la tarea que requiere edición	Muestra los detalles actuales de entrega
	Cambia fecha, hora o requisitos	Guarda las modificaciones
Flujo alternativo 1	Selecciona un grupo de tareas para editar	Aplica los cambios a todas las tareas seleccionadas
Flujo alternativo 2		
Post-condiciones	La nueva fecha o detalles quedan registrados	Error al eliminar tarea por conflicto de dependencias.

Comentado [k7]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Exepciones	Error al guardar cambios por conflictos de agenda. |	

Comentado [k8]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



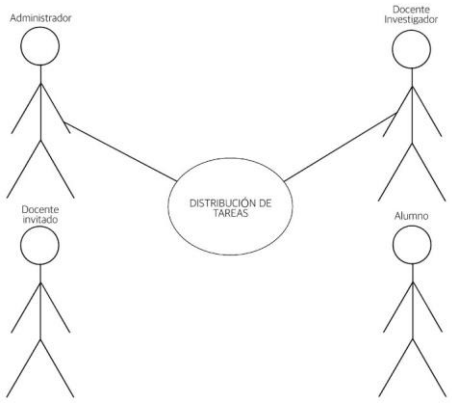
CASO No. 5 Agregar Comentarios en Tareas (Docente Invitado)

ID:	F-ADM-5	
Nombre	Agregar Comentarios en Tareas	
Actores	Administrador, Docente investigador, Docente invitado, alumno	
Objetivo	Permitir al docente invitado agregar comentarios a tareas asignadas.	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	El docente debe tener acceso a las tareas.	
Flujo Normal	Docente	Sistema
	Selecciona una tarea específica	Muestra los detalles de la tarea
	Ingresa el texto del comentario	Guarda el comentario en la tarea

Flujo alternativo 1	Selecciona un comentario existente	Permite modificar y guardar el comentario actualizado
Flujo alternativo 2		
Post-condiciones	El comentario queda registrado en el sistema.	
Exepciones	Error al guardar comentario por conexión lenta	

Comentado [k9]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k10]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



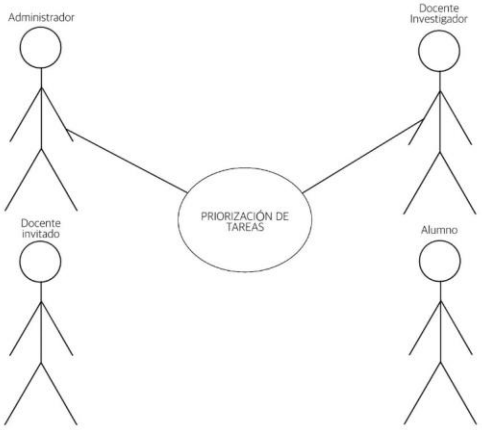
CASO No. 6 Distribución de tareas

ID:	F-ADM-6

Nombre	Distribución de tareas	
Actores	Administrador, Docente investigador,	
Objetivo	Asignar tareas a los diferentes actores del sistema.	
Urgencia	5	
Esfuerzo	3	
Pre-condiciones	Las tareas deben estar creadas previamente.	
Flujo Normal	Docente	Sistema
	Selecciona una tarea para asignar	Muestra la lista de usuarios disponibles
	Selecciona el usuario correspondiente	Registra la asignación en el sistema
Flujo alternativo 1	Reasigna una tarea a un usuario diferente	Actualiza los datos de asignación
Flujo alternativo 2		
Post-condiciones	Las tareas quedan asignadas correctamente	
Exepciones	Usuario no disponible para asignación	

Comentado [k11]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k12]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



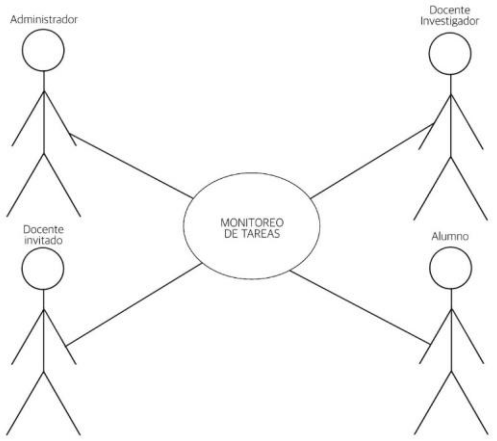
CASO No. 7 Priorización de tareas

ID:	F-ADM-7	
Nombre	Priorización de Tareas	
Actores	Administrador, Docente investigador,	
Objetivo	Permitir al administrador establecer niveles de prioridad en las tareas.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	Las tareas deben estar creadas y visibles en el sistema.	
Flujo Normal	Docente	Sistema
	Marca una tarea con un nivel de prioridad	Actualiza la tarea con el nuevo nivel
Flujo alternativo 1	Selecciona varias tareas para modificar	Aplica la nueva prioridad a todas
Flujo alternativo 2		
Post-condiciones	Las tareas tienen un nivel de prioridad actualizado.	

Comentado [k13]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Exepciones	Error al actualizar prioridad por conflictos en la base de datos.	

Comentado [k14]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



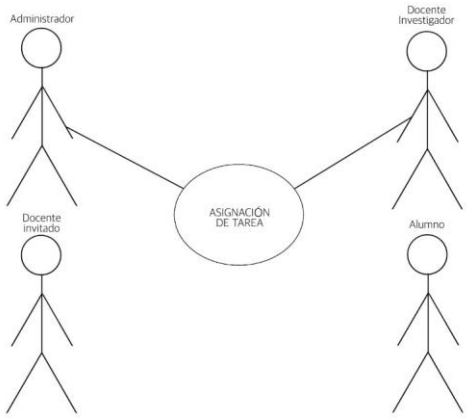
CASO No. 8 Monitoreo de Tareas

ID:	F-ADM-8	
Nombre	Monitoreo de Tareas	
Actores	Administrador, Docente investigador, docente invitado, alumno	
Objetivo	Permitir al docente investigador monitorear el progreso de tareas asignadas.	
Urgencia	5	
Esfuerzo	4	
Pre-condiciones	El docente debe tener tareas asignadas y acceso al sistema.	
Flujo Normal	Docente	Sistema
	Selecciona "Monitorear Tareas"	Muestra las tareas en curso

	Selecciona una tarea específica para consultar detalles	Muestra el progreso de la tarea seleccionada
Flujo alternativo 1	Solicita un reporte detallado de tareas	Genera y descarga el archivo con datos actualizados
Flujo alternativo 2		
Post-condiciones	El docente obtiene información actualizada sobre el estado de las tareas.	
Exepciones	Error al cargar datos por problemas de conexión	

Comentado [k15]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k16]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.

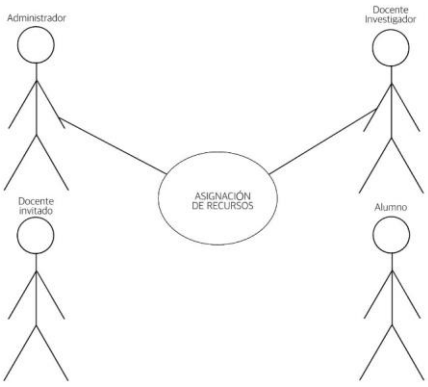


CASO No. 9 Asignación de Tareas (Administrador)

ID:	F-ADM-9	
Nombre	Asignación de Tareas	
Actores	Administrador, Docente investigador	
Objetivo	Permitir al administrador asignar tareas a usuarios específicos (docentes o alumnos)	
Urgencia	5	
Esfuerzo	4	
Pre-condiciones	Las tareas deben estar creadas previamente. - Los usuarios deben estar registrados y disponibles.	
Flujo Normal	Docente	Sistema
	Selecciona "Asignar Tareas"	Muestra la lista de tareas disponibles
	Elige una tarea a asignar	Muestra la lista de usuarios disponibles
Flujo alternativo 1	Selecciona el usuario correspondiente	Genera y descarga el archivo con datos actualizados
	Selecciona el usuario correspondiente	Registra la asignación en el sistema
Flujo alternativo 2		
Post-condiciones	a tarea queda asignada al usuario correspondiente y aparece en su lista de tareas	
Exepciones	El usuario seleccionado no tiene permisos para realizar la tarea. Error al guardar la asignación por conflictos en la base de datos	

Comentado [k17]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k18]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.

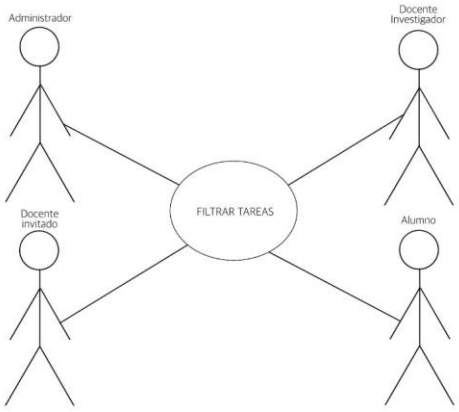


CASO No. 10 Asignación de recursos

ID:	F-ADM-10	
Nombre	Asignación de Recursos	
Actores	Administrador, Docente investigador	
Objetivo	Asignar recursos específicos a tareas o usuarios	
Urgencia	4	
Esfuerzo	3	
Pre-condiciones	Los recursos deben estar registrados en el sistema.	
Flujo Normal	Docente	Sistema
	Selecciona un recurso del catálogo	Muestra el estado del recurso
	Lo asocia a una tarea o usuario	Registra la asignación
Flujo alternativo 1	Cambia el recurso a otra tarea o usuario	Actualiza los registros correspondientes
Flujo alternativo 2		
Post-condiciones	Los recursos quedan correctamente asignados	
Exepciones	Recurso ya asignado a otra tarea	

Comentado [k19]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k20]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.



CASO No. 11 Filtrar Tareas

ID:	F-ADM-11	
Nombre	Filtrar Tareas	
Actores	Administrador, Docente investigador, Docente invitado, alumno	
Objetivo	Permitir al alumno aplicar filtros para organizar sus tareas asignadas.	
Urgencia	2	
Esfuerzo	1	
Pre-condiciones	El alumno debe estar autenticado y tener tareas asignadas.	
Flujo Normal	Docente	Sistema
	Selecciona "Filtrar Tareas"	Muestra las opciones de filtrado
	Define criterios como prioridad o estado	Muestra las tareas según los filtros aplicados
Flujo alternativo 1	Selecciona "Limpiar Filtros"	Muestra todas las tareas sin aplicar filtros

Flujo alternativo 2		
Post-condiciones	El alumno visualiza las tareas organizadas según los filtros. | | **Excepciones:** | - Error al aplicar filtros por parámetros inválidos. |	
Exepciones		

Comentado [k21]: Son los estados finales con los que termina el caso de uso, estos se deben cumplir y cualquier situación que no permita cumplirlos debe quedar consignada como flujo alternativo.

Comentado [k22]: Son excepciones que no dependen del sistema o actores y que se deben resolver en otro flujo.

Prioridad de Requerimientos

A partir del análisis de requerimientos, funcionalidades y el proceso de Task Manager, 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
	4-Alto	4	8	12	16	20 CU-1
	3-Medio	3	6	9	12	15
	2-Bajo	2	4	6	8	10
	1-Muy bajo	1	2	3	4	5

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

Requisitos No Funcionales

Seguridad:
- La pizarra debe garantizar la seguridad de los datos y la autenticación de usuarios. Debe utilizar cifrado para proteger la información.

Rendimiento:
- La aplicación debe ofrecer un rendimiento óptimo, permitiendo la colaboración en tiempo real incluso con un gran número de usuarios.

Escalabilidad:
- La pizarra debe ser escalable para manejar un aumento en el número de usuarios y la cantidad de contenido.

Disponibilidad:
- La aplicación debe estar disponible y funcionando de manera constante, minimizando el tiempo de inactividad.

Compatibilidad con Dispositivos:
- La pizarra debe ser compatible con una variedad de dispositivos, incluyendo computadoras de escritorio, tabletas y teléfonos móviles.

Usabilidad:
- La interfaz de usuario de la pizarra debe ser intuitiva y fácil de usar para usuarios de diferentes niveles de habilidad.

Accesibilidad:
- La aplicación debe ser accesible para personas con discapacidades, cumpliendo con estándares de accesibilidad web.

Cumplimiento Normativo:
- La pizarra debe cumplir con regulaciones y normativas de privacidad y seguridad de datos.

Tiempo de Respuesta:
- La aplicación debe tener tiempos de respuesta rápidos para mantener una experiencia de usuario fluida.

Requisitos de Desempeño

1. **Rendimiento en Tiempo Real**: La pizarra debe proporcionar un rendimiento en tiempo real, lo que significa que los cambios realizados por los usuarios deben reflejarse instantáneamente para todos los colaboradores, incluso cuando múltiples usuarios trabajen simultáneamente en la pizarra. Este aspecto se debe desarrollar con sockets.
2. **Tiempo de Carga Rápido**: La pizarra debe cargar de manera eficiente, y los usuarios no deben experimentar tiempos de carga excesivamente largos al acceder a una pizarra o al editar contenido. Se requiere que los componentes estén bien diseñados y acoplados. Por lo general los componentes de la arquitectura CREAVI la cual sigue el paradigma de la programación orientada a componentes.
3. **Optimización de Recursos**: El sistema debe estar optimizado para utilizar eficientemente los recursos del servidor, minimizando el uso de CPU y memoria. El renderizado de los componentes adecuados garantiza este requisito.

Requisitos de Seguridad

4. **Acceso Seguro**: Se debe implementar una autenticación segura para garantizar que solo usuarios autorizados tengan acceso a las pizarras. Esto puede incluir autenticación de dos factores, inicio de sesión único (SSO), autenticación con JWT o Auth 2.
5. **Protección de Datos**: La pizarra debe garantizar la protección de datos sensibles, como información del usuario y contenido compartido. Se debe cifrar la información en tránsito y en reposo.
6. **Auditoría y Registro de Actividades**: El sistema debe mantener registros de actividades, lo que incluye registros de cambios en la pizarra, acceso de usuarios y eventos relevantes para la seguridad.
7. **Control de versiones**: El sistema debe llevar un registro de los cambios de los datos gestionados en la pizarra así como también los datos mismos de la estructura del componente.
8. **Variables de entorno**: El sistema debe ser manejado con variables de entorno que garanticen su fácil incorporación con otros módulos y la migración entre plataformas, así como tambien almacenar los datos iniciales del servidor como lo son las bases de datos y las llaves de autenticación, entre otras.

Requisitos de Usabilidad

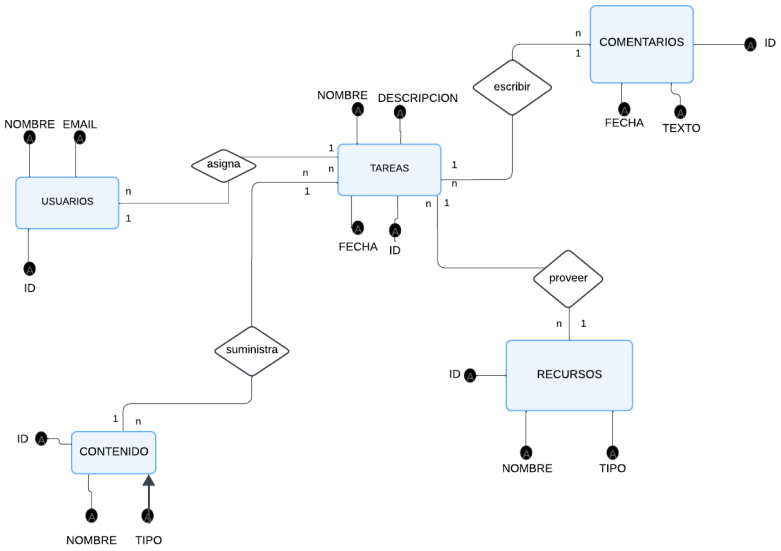
9. **Interfaz Intuitiva:** La interfaz de usuario de la pizarra debe ser intuitiva y fácil de usar, permitiendo a los usuarios realizar acciones como dibujar, agregar contenido y colaborar sin dificultad.
10. **Compatibilidad con Dispositivos:** La pizarra debe ser compatible con una variedad de dispositivos,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

Requisitos de Escalabilidad

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

Modelado E/R

Diagrama de Entidad-Relación



Script de modelo relacional

https://dbdiagram.io/

```
Table Users {
  id integer [primary key]
  username varchar
  role varchar
  created_at timestamp
}

Table User_Content {
  user_id integer [ref: > Users.id]
  content_id integer [ref: > Contents.id]
}

Table Content_Comment {
  content_id integer [ref: > Contents.id]
  comment_id integer [ref: > Comments.id]
}

Table Whiteboards {
  id integer [primary key]
  title varchar
  description varchar
  created_at timestamp
}

Table Whiteboards_WhiteboardMembers {
  whiteboardMembers_id integer [ref: > WhiteboardMembers.id]
  Whiteboards_id integer [ref: > Whiteboards.id]
}

Table Whiteboard_Plugin {
  whiteboard_id integer [ref: > Whiteboards.id]
  plugin_id integer [ref: > Plugins.id]
}

Table Contents {
  id integer [primary key]
  type varchar
  content text [note: 'Content of the content']
  created_at timestamp
}

Table Contents_Whiteboards {
  Contents_id integer [ref: > Contents.id]
  whiteboard_id integer [ref: > Whiteboards.id]
}

Table Content_MultimediaResource {
  content_id integer [ref: > Contents.id]
  MultimediaResources_id integer [ref: > MultimediaResources.id]
}

Table Comments {
  id integer [primary key]
  text text
  created_at timestamp
}

Table RevisionHistorys {
  id integer [primary key]
  date timestamp
  whiteboard_id integer [ref: > Whiteboards.id]
}

Table MultimediaResources {
  id integer [primary key]
  type varchar
  location varchar
  title varchar
  description varchar
  created_at timestamp
}

Table Plugins {
  id integer [primary key]
  name varchar
  description varchar
  author varchar
  version varchar
  configuration varchar
}

Table WhiteboardMembers {
  id integer [primary key]
  user_id integer [ref: > Users.id]
  role varchar
}
```

Descripción de Entidades y Relaciones

Entidades:

1. **User (Usuario):**

   Almacena información sobre los usuarios que pueden acceder a la pizarra.
   Atributos: ID (identificador único), nombre de usuario, rol (como administrador, docente, estudiante), fecha de creación.
   Relaciones: Cada usuario puede estar asociado con varias pizarras a través de la entidad "WhiteboardMember."
2. **Whiteboard (Pizarra):**

   Representa una pizarra en la aplicación de pizarra compartida.
   Atributos: ID (identificador único), título de la pizarra, descripción, fecha de creación.
   Relaciones: Cada pizarra puede contener contenido a través de la relación con la entidad "Content" y puede tener plugins asociados a través de la relación con la entidad "Plugin."
3. **Content (Contenido):**

   Almacena contenido que se puede agregar a las pizarras, como texto, imágenes, videos, documentos, etc.
   Atributos: ID (identificador único), tipo de contenido, contenido en sí, fecha de creación.
   Relaciones: El contenido se asocia con un usuario a través de la relación con la entidad "User" y puede estar relacionado con comentarios.
4. **Comment (Comentario):**

   Almacena comentarios realizados por los usuarios en relación con el contenido de la pizarra.
   Atributos: ID (identificador único), texto del comentario, fecha de creación.
   Relaciones: Cada comentario se relaciona con el contenido específico en la entidad "Content."
5. **RevisionHistory (Historial de Revisiones):**

   Registra el historial de revisiones y cambios realizados en las pizarras.
   Atributos: ID (identificador único), fecha de la revisión.
   Relaciones: Cada entrada de historial se relaciona con una pizarra específica en la entidad "Whiteboard."
6. **MultimediaResource (Recurso Multimedia):**

   Almacena recursos multimedia, como imágenes, videos, documentos, etc.
   Atributos: ID (identificador único), tipo de recurso, ubicación o URL del recurso, título, descripción, fecha de carga.