



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Desarrollo de un gestor web de proyectos colaborativos académicos

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Rubén Arranz Martínez

**Tutor:** Germán Francisco Vidal

2024-2025



# Resumen

---

En el contexto académico universitario, la coordinación de trabajos en grupo presenta desafíos recurrentes, especialmente en titulaciones técnicas como Ingeniería Informática. La distribución de tareas, el seguimiento del progreso individual y la evaluación justa de las contribuciones son aspectos difíciles de gestionar con herramientas generalistas. Este Trabajo de Fin de Grado propone el desarrollo de una aplicación web orientada a la gestión de proyectos colaborativos académicos, adaptada específicamente al entorno y necesidades del estudiante universitario.

La solución planteada permite a los usuarios crear, asignar y gestionar tareas, adjuntar archivos, visualizar el avance del grupo y recibir notificaciones de cambios. Además, se incorporan funcionalidades para la evaluación de las contribuciones individuales y la supervisión por parte del profesorado. El proyecto ha sido desarrollado siguiendo la metodología ágil SCRUM, con una arquitectura basada en React (frontend), FastAPI (backend) y PostgreSQL (base de datos). La memoria detalla tanto el análisis de herramientas existentes como la especificación funcional, el diseño, la implementación y la validación del sistema propuesto.

**Palabras clave:** Desarrollo web, organización de proyectos, aplicaciones colaborativas, checklists, gestión de tareas, entorno académico

# Abstract

---

In the academic context of university studies, coordinating group projects presents recurring challenges, particularly in technical degrees such as Computer Engineering. Task distribution, progress tracking, and fair evaluation of individual contributions are difficult to manage with generic tools. This Final Degree Project proposes the development of a web application specifically designed for academic collaborative project management, tailored to the needs of university students.

The proposed solution allows users to create, assign, and manage tasks, attach files, view group progress, and receive change notifications. It also includes functionalities for tracking individual contributions and providing supervision tools for instructors. The project was developed following the agile SCRUM methodology and is based on an architecture that integrates React (frontend), FastAPI (backend), and PostgreSQL (database). This document details the analysis of existing tools, functional specifications, design, implementation, and validation of the proposed system.

**Keywords :** integer, blandit, pharetra, urna, id.



# Tabla de contenidos

---

1.Introducción.....	7
1.1. Motivación y contexto.....	7
1.2. Objetivos del proyecto.....	7
1.3. Metodología de trabajo.....	8
1.4. Estructura de la memoria.....	12
2.Estado del arte.....	14
2.1 Herramientas existentes.....	14
2.2 Comparativa con tu propuesta.....	15
2.3 Justificación de la solución elegida.....	16
3.Especificación del problema.....	17
3.1. Requisitos funcionales.....	17
3.2. Requisitos no funcionales .....	17
3.3. Usuarios objetivo y escenarios de uso.....	18
3.4 Alcance del proyecto.....	20

## Tabla de figuras

---

Figura 1. User stories Sprint 1 .....	9
Figura 2. User stories Sprint 2 .....	10
Figura 3. User Stories Sprint 3.....	10
Figura 4. Tabla comparativa otras soluciones .....	15
Figura 5. Casos de uso estudiante .....	19
Figura 6. Casos de uso docente.....	20

# 1.Introducción

---

## 1.1. Motivación y contexto

---

A la hora de colaborar en grupo, la coordinación no siempre es fácil, sobre todo en la universidad. En grados más técnicos como Ingeniería Informática, lo habitual es que los estudiantes tengan que colaborar en varios proyectos a lo largo del curso.

Trabajar eficientemente en grupo no siempre es fácil, sobre todo en la universidad. En grados como Ingeniería Informática, lo habitual es que los estudiantes tengan que colaborar en varios proyectos a lo largo del curso. Cuando las tareas no se organizan bien y no hay un buen seguimiento, es muy fácil que el grupo acabe descoordinado. Y eso, al final, afecta negativamente al rendimiento del grupo y a la calidad del producto final.

Aunque existen herramientas como Trello o Notion que permiten gestionar tareas, sin embargo, no están del todo hechas para usarse en clase o en trabajos de la universidad. Pueden servir, sí, pero no cubren todas las necesidades que surgen cuando se trabaja con compañeros en proyectos académicos. Por ejemplo, no permiten hacer seguimiento por roles, gestionar entregas de forma clara o ver con facilidad cómo va cada persona dentro del grupo.

Por este motivo, este Trabajo de Fin de Grado propone crear una aplicación web que se ajuste mejor a estas situaciones. La idea es crear algo simple que ayude a los estudiantes a organizarse mejor, repartir bien el trabajo y seguir el avance de cada parte del proyecto. Incluso los docentes podrían usarla para seguir cómo va cada grupo.

## 1.2. Objetivos del proyecto

---

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una aplicación web para la gestión de proyectos colaborativos en entornos académicos, diseñada para mejorar la organización, coordinación y evaluación del trabajo en grupo entre estudiantes universitarios.

Para lograr este objetivo general, se plantean los siguientes objetivos específicos:

- Optimizar la gestión de tareas y responsabilidades: Facilitar la creación, asignación y seguimiento de tareas, permitiendo a los grupos definir roles y responsabilidades de forma clara y estructurada.
- Mejorar la transparencia y el seguimiento del progreso: Incluir herramientas para visualizar el desempeño individual y grupal, integrando indicadores de avance que faciliten tanto el control interno del grupo como la evaluación externa por parte de los docentes.

- Simplificar la interfaz y la experiencia del usuario: Diseñar una plataforma intuitiva, adaptada a las necesidades del entorno académico, que evite las complejidades innecesarias de las herramientas corporativas.
- Incorporar funcionalidades para la gestión de entregas: Permitir a los grupos gestionar plazos, verificar el cumplimiento de objetivos y facilitar el control de versiones de entregas de proyectos.

Además, se busca que el desarrollo de esta aplicación sirva como oportunidad para:

- Adquirir experiencia en tecnologías web como React, FastAPI y PostgreSQL.
- Aplicar metodologías ágiles como SCRUM para gestionar el proyecto de forma eficiente.
- Desarrollar habilidades en diseño de interfaces de usuario (UI) y experiencia de usuario (UX).

### 1.3. Metodología de trabajo

---

Para el desarrollo de este Trabajo de Fin de Grado, se ha adoptado la metodología ágil **SCRUM** por su capacidad para gestionar proyectos de software de forma iterativa, incremental y flexible. Esta metodología es especialmente adecuada para proyectos en solitario, ya que permite ajustar prioridades y requisitos conforme avanza el desarrollo.

SCRUM divide el trabajo en pequeños bloques llamados *sprints*, que permiten entregar valor de forma continua y detectar problemas de forma temprana.

Ventajas de usar SCRUM en este proyecto

- **Flexibilidad y Adaptabilidad:** permite ajustar el alcance y las prioridades según sea necesario.
- **Entrega Continua de Valor:** los sprints aseguran que siempre se esté avanzando hacia un producto funcional.
- **Transparencia en el Proceso:** facilita el seguimiento del progreso del proyecto y la identificación temprana de problemas.
- **Mejora Continua:** las retrospectivas permiten ajustar el proceso de desarrollo después de cada sprint.

---

### Roles asumidos



Dado que este es un proyecto en solitario, se asumirán todos los roles típicos de un equipo SCRUM:

- **Product Owner:** define los requisitos del proyecto, prioriza las historias de usuario y toma decisiones sobre el alcance.
- **Scrum Master:** gestiona el proceso SCRUM, elimina obstáculos y asegura el cumplimiento de buenas prácticas.
- **Development Team:** implementa las funcionalidades de cada sprint, incluyendo diseño, programación y pruebas.

## Estructura de los sprints

El proyecto se divide en **tres sprints principales** de aproximadamente dos semanas (ajustables según el progreso y los riesgos):

### ➤ Sprint 1: Fundamentos del Proyecto y Backend Básico

- Configurar la base de datos PostgreSQL.
- Crear modelos para usuarios, proyectos y tareas.
- Implementar el sistema de registro y autenticación de usuarios.
- Crear las primeras APIs para gestionar usuarios, proyectos y tareas.

Product Backlog					
As User Story	Description	Priority	Status	Sprint	Story Points
Ver Listado de tareas	Como usuario, quiero poder ver todas las tareas del proyecto en un solo lugar para tener una visión clara del estado del proyecto.	Alta	Backlog	Sprint 1	2
Configurar la Base de Datos	Como desarrollador, quiero configurar la base de datos PostgreSQL para almacenar usuarios, tareas y proyectos de forma estructurada.	Alta	Backlog	Sprint 1	2
Modelo de Proyecto	Como desarrollador, quiero crear el modelo de proyecto para agrupar tareas y asignar usuarios a proyectos específicos.	Alta	Backlog	Sprint 1	2
Crear tareas	Como usuario, quiero poder crear tareas con título, descripción, fecha de vencimiento y prioridad para organizar mi trabajo.	Alta	Backlog	Sprint 1	3
Asignar tareas	Como líder de grupo, quiero poder asignar tareas a diferentes miembros para distribuir el trabajo equitativamente.	Alta	Backlog	Sprint 1	3
Modelo de Usuario	Como desarrollador, quiero crear el modelo de usuario con autenticación básica para que los usuarios puedan registrarse e iniciar sesión.	Alta	Backlog	Sprint 1	3
Modelo de Tarea	Como desarrollador, quiero crear el modelo de tarea para almacenar toda la información relacionada con las tareas del proyecto.	Alta	Backlog	Sprint 1	2
Filtrado de tareas	Como usuario, quiero poder filtrar las tareas por estado, prioridad o fecha de vencimiento para encontrar rápidamente las tareas relevantes.	Baja	Backlog	Sprint 1	2

Figura 1. User stories Sprint 1

### ➤ Sprint 2: Funcionalidades Principales

- la gestión de tareas: creación, asignación, edición y finalización.
- Añadir la funcionalidad de comentarios y archivos en tareas.
- Desarrollar las funcionalidades del dashboard del proyecto.
- Añadir filtros para mejorar la usabilidad de la lista de tareas.

## Desarrollo de un gestor web de proyectos colaborativos académicos

- Configurar la autenticación JWT para proteger las rutas de la API.
- Incorporar la visualización del progreso y las estadísticas de contribuciones por usuario.

Finalización de tareas	Como usuario, quiero poder marcar tareas como completadas para mantener un registro claro del progreso del proyecto.	Alta	Backlog	Sprint 2	1
Editar tareas	Como usuario, quiero poder editar las tareas existentes para actualizar detalles como el título, la descripción, la prioridad y la fecha de vencimiento.	Media	Backlog	Sprint 2	2
Dashboard del proyecto	Como usuario, quiero ver un tablero que me muestre el estado general del proyecto, incluyendo tareas completadas, en progreso y pendientes.	Media	Backlog	Sprint 2	2
Seguimiento de progreso	Como usuario, quiero poder ver el progreso general del proyecto para entender cómo avanza el trabajo en equipo.	Media	Backlog	Sprint 2	4
Gestión de Roles	Como administrador del proyecto, quiero poder asignar roles a los miembros para definir responsabilidades claras.	Media	Backlog	Sprint 2	3
Comentar en tareas	Como miembro del grupo, quiero poder comentar en tareas para discutir detalles específicos con mis compañeros.	Baja	Backlog	Sprint 2	2
API para Crear Usuarios	Como desarrollador, quiero crear una API para registrar nuevos usuarios y almacenar sus datos en la base de datos.	Alta	Backlog	Sprint 2	2
API para Crear Proyectos	Como desarrollador, quiero crear una API para que los usuarios puedan crear proyectos nuevos y asignar miembros.	Alta	Backlog	Sprint 2	3
API para Crear Tareas	Como desarrollador, quiero crear una API para que los usuarios puedan crear tareas dentro de los proyectos.	Alta	Backlog	Sprint 2	3
API para Asignar Tareas	Como desarrollador, quiero crear una API para asignar tareas a usuarios específicos dentro de un proyecto.	Alta	Backlog	Sprint 2	2
Autenticación y Control de Acceso	Como desarrollador, quiero implementar autenticación con JWT para proteger las rutas de la API y asegurar que solo los usuarios autorizados puedan acceder a sus datos.	Alta	Backlog	Sprint 2	4

Figura 2. User stories Sprint 2

### ➤ Sprint 3: Pruebas, Optimización y Despliegue

- Implementar notificaciones de cambios en tareas.
- Optimizar las consultas a la base de datos para mejorar el rendimiento.
- Añadir pruebas unitarias e integración.
- Desplegar el proyecto en un servidor en línea (Vercel o Heroku/Fly.io).

Notificaciones de cambios	Como usuario, quiero recibir notificaciones cuando alguien modifique una tarea asignada a mí para estar siempre actualizado.	Media	Backlog	Sprint 3	3
Adjuntar archivos a tareas	Como usuario, quiero poder adjuntar archivos a las tareas para compartir documentos importantes.	Baja	Backlog	Sprint 3	2
Optimización de Consultas a la Base de Datos	Como desarrollador, quiero optimizar las consultas a la base de datos para mejorar el rendimiento del sistema.	Media	Backlog	Sprint 3	3
API para Comentarios en Tareas	Como desarrollador, quiero crear una API para que los usuarios puedan añadir y ver comentarios en las tareas.	Media	Backlog	Sprint 3	2
API para Notificaciones	Como desarrollador, quiero crear una API para enviar notificaciones a los usuarios cuando se actualizan tareas o se añaden comentarios.	Media	Backlog	Sprint 3	3
Sistema de Logs y Auditoría	Como desarrollador, quiero crear un sistema de logs para registrar todas las acciones importantes del sistema y poder hacer seguimiento de errores.	Media	Backlog	Sprint 3	3
Pruebas Unitarias y de Integración	Como desarrollador, quiero crear pruebas unitarias y de integración para asegurar que las APIs funcionan correctamente.	Alta	Backlog	Sprint 3	5
Despliegue y Configuración del Servidor	Como desarrollador, quiero desplegar la aplicación en un servidor de producción para que los usuarios puedan acceder a ella en línea.	Alta	Backlog	Sprint 3	4
Seguimiento de contribuciones	Como profesor, quiero poder ver cuánto ha contribuido cada miembro del grupo para evaluar el rendimiento individual.	Alta	Backlog	Sprint 3	5
Sistema de evaluación	Como profesor, quiero poder evaluar las contribuciones individuales para dar una calificación justa a cada miembro del grupo.	Alta	Backlog	Sprint 3	5
Gestión de entregas	Como usuario, quiero poder marcar tareas como entregadas y verificar su estado para asegurarme de cumplir con los plazos.	Alta	Backlog	Sprint 3	4

Figura 3. User Stories Sprint 3

## Herramientas utilizadas

- **Gestión de tareas:** Notion para el *Product Backlog* y gestión de sprints.
- **Frontend:** React.js (JavaScript)
- **Backend:** FastAPI (Python)
- **Base de datos:** PostgreSQL
- **Control de versiones:** Git y GitHub
- **Pruebas:** Pytest para unitarias, Insomnia para APIs

- **Despliegue:** Vercel (frontend) y Heroku/Fly.io (backend)
- 

## Definition of Ready (DoR) y Definition of Done (DoD)

Para asegurar la calidad y trazabilidad de las historias de usuario, se siguen los siguientes criterios:

### Definition of Ready (DoR):

- Descripción clara.
- Criterios de aceptación definidos.
- Priorizada y aprobada en el Product Backlog.

### Definition of Done (DoD):

- Implementación completa según los criterios.
  - Código probado y validado.
  - Código limpio y documentado.
    - Cambios registrados en control de versiones.
- 

## Gestión de riesgos y adaptabilidad

Se identifican los siguientes riesgos potenciales:

- **Falta de tiempo:** riesgo alto por carga de trabajo en solitario.
- **Complejidad técnica:** especialmente en la autenticación JWT o la integración de APIs.
- **Cambios en requisitos:** posible necesidad de reorganizar prioridades en función del tiempo disponible.

Para mitigar estos riesgos se priorizarán historias críticas y se mantendrá un enfoque iterativo y flexible.

---

## Conclusión

Esta metodología de trabajo está diseñada para maximizar la productividad, minimizar los riesgos y asegurar la entrega de un producto funcional en el plazo previsto. Se realizarán ajustes iterativos según las necesidades técnicas y los recursos disponibles.

### 1.4. Estructura de la memoria

---

La estructura de la memoria del TFG seguirá el formato recomendado por la ETSINF, asegurando que se incluyan todos los apartados necesarios para cumplir con los requisitos académicos. Esta estructura está diseñada para presentar de forma clara y organizada tanto el desarrollo técnico como los resultados obtenidos durante el proyecto.

#### Capítulos Principales

- **1. Introducción:**  
Incluye el contexto del proyecto, la motivación, los objetivos y la estructura del documento.
- **2. Estado del Arte:**  
Análisis de las soluciones existentes, incluyendo herramientas y metodologías relacionadas con el proyecto. También se justificará el enfoque elegido para desarrollar la aplicación.
- **3. Especificación del Problema:**  
Definición clara del problema que se busca resolver con el proyecto, incluyendo requisitos funcionales y no funcionales.
- **4. Diseño de la Solución:**  
Descripción de la arquitectura del sistema, incluyendo diagramas de flujo, modelos de datos y la estructura del backend y frontend.
- **5. Implementación:**  
Detalles técnicos del desarrollo del sistema, incluyendo código, tecnologías utilizadas y capturas de pantalla de la interfaz.
- **6. Pruebas y Validación:**  
Descripción de las pruebas realizadas para validar las funcionalidades del sistema, incluyendo pruebas unitarias, de integración y de usuario.
- **7. Conclusiones y Trabajo Futuro:**  
Resumen de los logros alcanzados, desafíos enfrentados y posibles mejoras para versiones futuras del proyecto.

- **8. Anexos:**

Documentación adicional, como manuales de usuario, código fuente y cualquier otro material de apoyo relevante para el proyecto.

Esta estructura asegura que todos los aspectos importantes del proyecto sean cubiertos, desde los objetivos iniciales hasta los resultados finales y las conclusiones, cumpliendo con los estándares de calidad exigidos para un Trabajo de Fin de Grado en la ETSINF.

## 2.Estado del arte

---

El análisis del estado del arte es fundamental para entender el contexto en el que se desarrollará este proyecto. Permite identificar las herramientas existentes, sus ventajas y limitaciones, y justificar la creación de una nueva solución adaptada específicamente a las necesidades de los estudiantes universitarios.

### 2.1 Herramientas existentes

---

Existen muchas herramientas populares para la gestión de proyectos colaborativos, cada una con características específicas que las hacen adecuadas para distintos tipos de proyectos. Sin embargo, estas plataformas presentan limitaciones significativas cuando se utilizan en entornos académicos. A continuación, se analizan las más relevantes:

- **Trello:**  
Trello es una herramienta popular basada en tableros Kanban, que permite organizar tareas de forma visual. Es fácil de usar y altamente flexible, pero carece de características específicas para la gestión académica, como la evaluación de contribuciones individuales o la gestión de roles dentro de un grupo.
- **Notion:**  
Notion es una plataforma versátil que combina bases de datos, documentos y gestión de tareas. Aunque ofrece una gran flexibilidad para estructurar la información, no incluye funcionalidades específicas para proyectos colaborativos académicos, como la supervisión de avances individuales o la evaluación de contribuciones.
- **Microsoft Teams:**  
Esta herramienta se enfoca en la comunicación y colaboración en equipo, integrando videollamadas, chat y almacenamiento de archivos. Sin embargo, no ofrece una gestión avanzada de tareas ni una forma clara de supervisar las contribuciones individuales en proyectos académicos.
- **Slack:**  
Slack es una plataforma de mensajería para equipos, diseñada principalmente para la comunicación rápida y eficiente. Aunque permite integrar otras aplicaciones para la gestión de proyectos, no está optimizada para el seguimiento de contribuciones o la gestión de roles en entornos académicos.

- Google Classroom, Moodle y Blackboard:

Estas plataformas son ampliamente utilizadas en el ámbito educativo para la gestión de cursos y la entrega de tareas, pero no están diseñadas para la gestión de proyectos colaborativos. Carecen de herramientas para asignar roles, evaluar contribuciones individuales y hacer un seguimiento detallado del progreso.

## 2.2 Comparativa con tu propuesta

Para entender mejor las limitaciones de las herramientas existentes y justificar la creación de una nueva solución, se presenta la siguiente tabla comparativa, que destaca las funcionalidades clave necesarias para proyectos académicos colaborativos:

Funcionalidad	Trello	Notion	Microsoft Teams	Slack	Mi página web
Gestión de tareas	✓	✓	✓	✗	✓
Seguimiento por roles	✗	✗	✗	✗	✓
Evaluación de contribuciones	✗	✗	✗	✗	✓
Gestión clara de entregas	✗	✗	✓	✗	✓
Supervisión de avances (Individual)	✗	✓	✓	✗	✓
Supervisión de avances (Equipo)	✗	✓	✓	✗	✓
Interfaz adaptada al entorno académico	✗	✗	✗	✗	✓
Simplicidad y enfoque académico	✗	✗	✗	✗	✓
Colaboración en tiempo real	✗	✓	✓	✓	✓
Integración con otras herramientas	✓	✓	✓	✓	✓
Facilidad de uso	✓	✓	✗	✓	✓

Figura 4. Tabla comparativa otras soluciones

Como se observa en la tabla, ninguna de las herramientas comerciales existentes ofrece todas las funcionalidades necesarias para gestionar proyectos académicos de forma eficiente. Esto justifica la creación de una nueva solución adaptada específicamente a las necesidades de los estudiantes universitarios, donde el seguimiento de roles, la evaluación justa de contribuciones y la supervisión de avances son esenciales.

### **2.3 Justificación de la solución elegida**

---

Dado que ninguna de las herramientas actuales cubre completamente las necesidades específicas de la gestión de proyectos colaborativos en el ámbito universitario, se justifica el desarrollo de una nueva aplicación diseñada específicamente para este contexto. Las principales razones para esta decisión incluyen:

- Limitaciones de las herramientas comerciales: Las plataformas populares carecen de funciones críticas para la evaluación justa de contribuciones y el seguimiento de roles.
- Necesidades específicas del entorno académico: Los estudiantes requieren herramientas que permitan distribuir responsabilidades de forma clara, gestionar entregas y evaluar el desempeño individual dentro de un grupo.
- Flexibilidad y personalización: Las herramientas comerciales están diseñadas para entornos empresariales, lo que las hace demasiado complejas para proyectos académicos.
- Impacto positivo en la organización del trabajo en grupo: Una plataforma diseñada específicamente para estudiantes universitarios mejorará significativamente la forma en que los grupos se organizan, colaboran y se autoevalúan, permitiendo una evaluación más justa y precisa del trabajo individual.



## 3. Especificación del problema

---

El desarrollo de proyectos en equipo es una habilidad esencial para los estudiantes universitarios, especialmente en carreras técnicas como Ingeniería Informática. Sin embargo, colaborar de forma eficiente en grupo presenta desafíos significativos, como la distribución de tareas, el seguimiento del progreso individual y la evaluación justa de las contribuciones de cada miembro. Estos problemas son aún más evidentes cuando se trabaja en entornos académicos, donde las herramientas comerciales no siempre se adaptan a las necesidades específicas de los estudiantes.

### 3.1. Requisitos funcionales

---

Los requisitos funcionales describen las capacidades esenciales que debe ofrecer la aplicación para cumplir con los objetivos del proyecto. Estos requisitos se han definido para garantizar que la herramienta sea útil y efectiva para los estudiantes universitarios que trabajan en proyectos colaborativos. Los requisitos funcionales incluyen:

- **Gestión de Tareas:** Crear, editar, asignar y eliminar tareas de forma sencilla. Las tareas deben incluir título, descripción, fecha de vencimiento y prioridad.
- **Asignación de Roles:** Permitir la asignación de roles específicos (líder, miembro, observador) para definir responsabilidades dentro del grupo.
- **Seguimiento de Progreso:** Incluir un tablero de proyecto que muestre el estado general del proyecto, incluyendo tareas completadas, en progreso y pendientes.
- **Gestión de Entregas:** Marcar tareas como entregadas y verificar su estado para asegurar el cumplimiento de los plazos.
- **Notificaciones de Cambios:** Informar a los usuarios cuando se realicen cambios en las tareas asignadas para mantener a todos los miembros actualizados.
- **Evaluación de Contribuciones:** Permitir a los estudiantes y docentes revisar el desempeño individual de cada miembro del grupo.
- **Historial de Cambios:** Registrar cambios en las tareas para tener un registro claro de las modificaciones realizadas.
- **Autenticación y Seguridad:** Implementar un sistema de autenticación seguro para proteger los datos del proyecto y evitar accesos no autorizados.

### 3.2. Requisitos no funcionales

---

Además de los requisitos funcionales, la aplicación debe cumplir con ciertos criterios técnicos para garantizar su rendimiento y facilidad de uso. Estos requisitos incluyen:

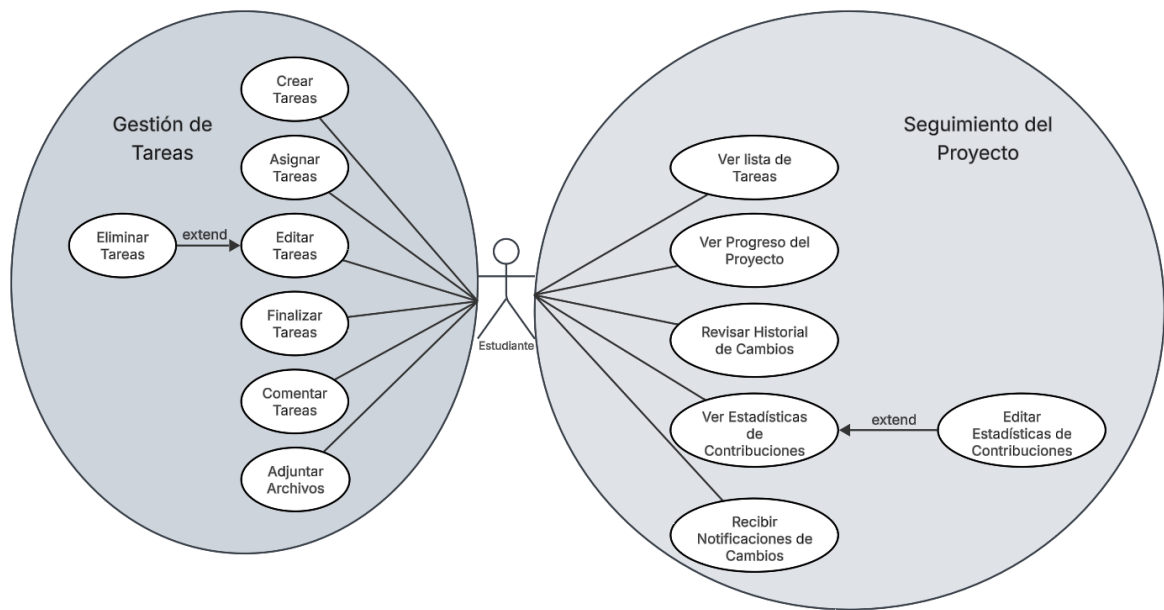
- **Rendimiento:** Responder rápidamente a las solicitudes de los usuarios, incluso con grandes volúmenes de datos.
- **Escalabilidad:** Poder manejar múltiples proyectos y usuarios sin comprometer el rendimiento del sistema.
- **Seguridad:** Proteger los datos mediante autenticación segura y cifrado de la información.
- **Usabilidad:** Tener una interfaz intuitiva y fácil de usar para que los estudiantes puedan enfocarse en sus proyectos sin distracciones.
- **Portabilidad:** Ser accesible desde diferentes dispositivos y plataformas, incluyendo móviles y tabletas.
- **Mantenimiento:** Facilitar futuras actualizaciones y mejoras sin requerir grandes cambios en el código base.

### **3.3. Usuarios objetivo y escenarios de uso**

---

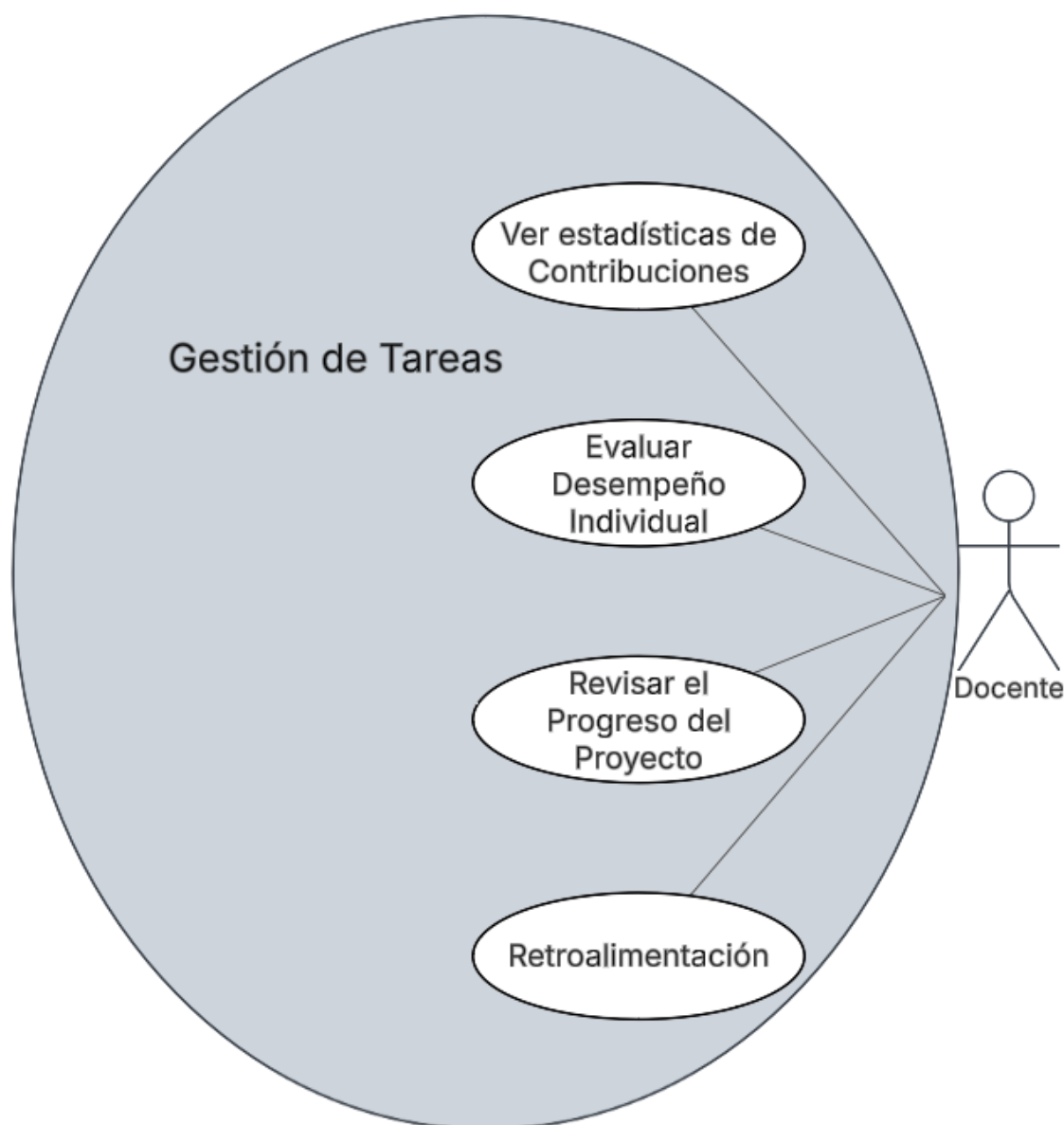
A continuación, se representan los principales casos de uso para los distintos perfiles definidos: Estudiante y Docente.

Estos diagramas muestran las funcionalidades clave que cada usuario puede realizar dentro de la aplicación, organizadas por bloques de acción (como gestión de tareas, seguimiento del progreso y evaluación de contribuciones).



*Figura 5. Casos de uso estudiante*

El estudiante, una vez autenticado, puede crear, asignar, editar, eliminar y finalizar tareas. Además, puede colaborar mediante comentarios y archivos, así como consultar el avance del proyecto, revisar su historial de cambios y ver sus estadísticas de contribución.



*Figura 6. Casos de uso docente*

El docente, por su parte, accede a un entorno de supervisión y evaluación. Tiene la posibilidad de consultar las estadísticas de cada alumno, revisar el progreso global del grupo, evaluar el desempeño individual y proporcionar retroalimentación para mejorar la dinámica de trabajo.

### 3.4 Alcance del proyecto

El alcance del proyecto incluye todas las funcionalidades que se espera que la aplicación ofrezca al momento de la entrega final. Estas funcionalidades han sido seleccionadas para cubrir las necesidades más críticas de los estudiantes universitarios que trabajan en proyectos colaborativos. El proyecto se centrará en desarrollar las siguientes características:

- **Gestión de Tareas:** Crear, asignar, editar y eliminar tareas.

- **Gestión de Roles:** Definir y asignar roles específicos dentro de los grupos de trabajo.
- **Seguimiento del Progreso:** Proporcionar herramientas para evaluar el rendimiento del grupo y el avance del proyecto.
- **Historial de Cambios:** Registrar las modificaciones para un seguimiento claro del progreso.
- **Autenticación y Seguridad:** Proteger los datos mediante autenticación y control de acceso seguro.
- **Evaluación de Contribuciones:** Permitir la evaluación justa del desempeño individual de cada miembro del grupo.
- **Escalabilidad y Flexibilidad:** Diseñar el sistema para que pueda adaptarse a proyectos de diferentes tamaños y complejidades.

En resumen, el alcance del proyecto está diseñado para cubrir las necesidades más importantes de los estudiantes universitarios, asegurando que puedan gestionar sus proyectos de forma eficiente, justa y segura.