



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA DEL SOFTWARE

EL FUTURO DEL APRENDIZAJE COLABORATIVO:  
PLATAFORMA CON IA GENERATIVA PARA COMPARTIR  
Y OPTIMIZAR APUNTES UNIVERSITARIOS

THE FUTURE OF COLLABORATIVE LEARNING:  
GENERATIVE AI-POWERED PLATFORM FOR SHARING  
AND ENHACING UNIVERSITY NOTES

Realizado por  
ANDRÉS RUIZ SÁNCHEZ

Tutorizado por  
EDUARDO GUZMÁN DE LOS RISCOS

Departamento  
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN  
UNIVERSIDAD DE MÁLAGA

MÁLAGA, SEPTIEMBRE DE 2025





UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADUADO EN INGENIERÍA DEL SOFTWARE

**EL FUTURO DEL APRENDIZAJE COLABORATIVO:  
PLATAFORMA CON IA GENERATIVA PARA  
COMPARTIR Y OPTIMIZAR APUNTES  
UNIVERSITARIOS**

**THE FUTURE OF COLLABORATIVE LEARNING:  
GENERATIVE AI-POWERED PLATFORM FOR  
SHARING AND ENHANCING UNIVERSITY NOTES**

Realizado por  
**Andrés Ruiz Sánchez**

Tutorizado por  
**Eduardo Guzmán de los Riscos**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, SEPTIEMBRE DE 2025

Fecha defensa: SEPTIEMBRE DE 2025



# Resumen

Este Trabajo Fin de Grado aborda el diseño e implementación de una plataforma innovadora de aprendizaje colaborativo, potenciada por Inteligencia Artificial Generativa, cuyo objetivo es optimizar la forma en la que los estudiantes universitarios comparten, organizan y enriquecen sus apuntes. La propuesta surge de la necesidad de superar las limitaciones de las plataformas tradicionales de intercambio de materiales académicos, ofreciendo un entorno más adecuado para los usuarios.

El desarrollo se ha llevado a cabo mediante un enfoque full-stack, empleando tecnologías modernas como Node.js, Express, FastAPI y bases de datos híbridas (PostgreSQL, MongoDB y Redis), junto con integración de servicios en la nube como Cloudflare R2. En el plano del frontend, se ha optado por Next.js y React, con diseño basado en componentes y estilos mediante Tailwind CSS, garantizando escalabilidad, accesibilidad y una experiencia de usuario consistente. La incorporación de modelos de IA generativa permite resumir, analizar y enriquecer los apuntes, facilitando así un aprendizaje más eficiente y adaptado a las necesidades individuales de cada estudiante.

El resultado es una plataforma que combina usabilidad, innovación y rigor técnico, con potencial para convertirse en un recurso clave en el ámbito universitario. Además, se discuten las ventajas frente a soluciones existentes y se reflexiona sobre los retos asociados al uso de Inteligencia Artificial en la educación superior.

## **Palabras clave:**

Inteligencia Artificial Generativa, Apuntes universitarios, Express.js, FastAPI, Next.js



# Abstract

This Final Degree Project addresses the design and implementation of an innovative collaborative learning platform powered by Generative Artificial Intelligence, aimed at optimizing the way university students share, organize and enhance their notes. The proposal arises from the need to overcome the limitations of traditional platforms for exchanging academic materials, offering a more suitable environment for users.

The development was carried out through a full-stack approach, using modern technologies such as Node.js, Express, FastAPI, and hybrid databases (PostgreSQL, MongoDB, and Redis), along with the integration of cloud services like Cloudflare R2. On the frontend side, Next.js and React were selected, with a component-based design and styling through Tailwind CSS, ensuring scalability, accessibility, and a consistent user experience. The integration of generative AI models makes it possible to summarize, analyze and enrich notes, thus enabling more efficient learning tailored to the individual needs of each student.

The result is a platform that combines usability, innovation and technical rigor, with the potential to become a key resource in the university environment. Furthermore, the advantages over existing solutions are discussed, and reflections are provided on the challenges associated with the use of Artificial Intelligence in higher education.

## **Keywords:**

Generative Artificial Intelligence, University notes, Express.js, FastAPI, Next.js



# Índice

<b>Resumen</b> .....	<b>1</b>
<b>Abstract</b> .....	<b>1</b>
<b>Índice</b> .....	<b>1</b>
<b>Introducción</b> .....	<b>1</b>
<b>1.1 Motivación</b> .....	<b>1</b>
La problemática de la dispersión de los materiales.....	1
El valor del aprendizaje colaborativo .....	2
La revolución de la Inteligencia Artificial en la educación .....	2
<b>1.2 Objetivos</b> .....	<b>2</b>
Objetivos generales .....	3
Objetivos específicos .....	4
<b>1.3 Estructura de la memoria</b> .....	<b>5</b>
<b>Estado del arte</b> .....	<b>9</b>
<b>2.1 Contexto y evolución del aprendizaje colaborativo</b> .....	<b>9</b>
<b>2.2 Antecedentes</b> .....	<b>11</b>
<b>2.3 Tecnologías utilizadas</b> .....	<b>14</b>
2.3.1. Tecnologías para el Backend.....	14
2.3.2. Tecnologías para el Frontend.....	20
2.3.3. Tecnologías para la API de IA.....	22
2.3.4. Tecnologías para el despliegue .....	24
<b>Desarrollo del proyecto</b> .....	<b>27</b>
<b>3.1 Metodología de trabajo empleada</b> .....	<b>27</b>
<b>3.2 Investigación y análisis</b> .....	<b>28</b>
Exploración de necesidades y primera aproximación.....	29
Investigación de tecnologías y definición del stack .....	29
Evaluación de opciones de inteligencia artificial.....	30
Decisión final sobre la IA.....	31
<b>3.3 Análisis de Requisitos</b> .....	<b>31</b>
3.3.1 Requisitos funcionales .....	31
3.3.2 Requisitos no funcionales .....	35

3.3.3 Casos de uso .....	36
<b>3.4 Diseño de arquitectura .....</b>	<b>39</b>
3.4.1 Componentes principales .....	40
3.4.2 Patrones arquitectónicos .....	41
3.4.3 Interconexiones y Flujo de datos .....	42
<b>3.5 Desarrollo de Backend .....</b>	<b>45</b>
3.5.1 Arquitectura General.....	45
3.5.2 Sistema de Autenticación .....	46
3.5.2 Gestión de datos .....	48
3.5.3 Controladores y lógica de negocio .....	52
3.5.4 Servicios de negocio .....	53
3.5.5 Integración con servicios externos .....	54
<b>3.6 Integración con modelos de IA.....</b>	<b>55</b>
<b>3.7 Desarrollo de Frontend .....</b>	<b>63</b>
<b>3.8 Pruebas de usabilidad y rendimiento .....</b>	<b>67</b>
<b>3.9 Despliegue de la aplicación.....</b>	<b>69</b>
<b>Conclusiones y líneas futuras .....</b>	<b>71</b>
4.1 Conclusiones .....	71
4.2 Líneas futuras .....	72
<b>Bibliografía .....</b>	<b>75</b>
<b>Manual de Despliegue .....</b>	<b>81</b>
Requerimientos .....	81
Configuración de servicios externos .....	82
Variables de Entorno .....	83
Despliegue con Docker (recomendado) .....	84
Despliegue manual .....	85
<b>Manual de Usuario .....</b>	<b>87</b>
Introducción .....	87
Registro y acceso.....	88
Dashboard principal (Inicio) .....	90
Gestión del perfil de usuario .....	91
Configuración .....	94
Información académica.....	96
Gestión de Documentos .....	105
Funcionalidades de Inteligencia Artificial.....	112
Paneles de administración (sólo administradores) .....	116

# 1

# Introducción

## 1.1 Motivación

En los últimos años, la educación universitaria ha experimentado una transformación marcada por la digitalización y la incorporación de nuevas tecnologías en los procesos de enseñanza y aprendizaje. En este contexto, los apuntes constituyen uno de los recursos más importantes para los estudiantes, ya que son una síntesis de las clases, las explicaciones de los docentes y sus propias interpretaciones personales. No obstante, la gestión de estos materiales presenta una serie de problemas.

### **La problemática de la dispersión de los materiales**

Los apuntes suelen encontrarse dispersos en múltiples formatos y ubicaciones: carpetas personales, grupos de mensajería instantánea o plataformas de almacenamiento en la nube. Esa falta de centralización no sólo dificulta el acceso a la información en momentos clave, sino que también conduce a un uso ineficiente del tiempo de estudio y a la pérdida de recursos valiosos. Además, en muchos casos, los estudiantes recurren a documentos incompletos, desactualizados o de fiabilidad dudosa, lo que compromete la calidad de su formación.

## **El valor del aprendizaje colaborativo**

En paralelo, el aprendizaje colaborativo ha cobrado gran relevancia dentro de la educación superior actual. El intercambio de materiales, ideas y experiencias entre estudiantes fomenta un entorno de trabajo en equipo en el que las fortalezas individuales se convierten en un beneficio común. Sin embargo, la ausencia de plataformas centralizadas que faciliten esta colaboración de manera estructurada y segura constituye una limitación evidente para el aprovechamiento colectivo del conocimiento generado en las aulas.

## **La revolución de la Inteligencia Artificial en la educación**

Por otro lado, la irrupción de la Inteligencia Artificial, especialmente de los Modelos Extensos del Lenguaje (LLM), está redefiniendo la forma en que interactuamos con la información. Estas tecnologías permiten procesar grandes volúmenes de texto, generar resúmenes, elaborar esquemas conceptuales, responder preguntas específicas y transformar documentos estáticos en recursos interactivos. Su integración en el ámbito educativo ofrece la posibilidad de modernizar profundamente el ecosistema académico, orientándolo hacia una enseñanza más personalizada, accesible y eficaz.

## **1.2 Objetivos**

Este Trabajo Fin de Grado se enmarca en este escenario y propone el desarrollo de una plataforma web innovadora para la gestión de apuntes universitarios, que centralice, organice y enriquezca los materiales de estudio a través de herramientas basadas en IA. De esta forma, se pretende dar respuesta a una necesidad latente en el ámbito universitario y contribuir a la evolución hacia un modelo de aprendizaje más colaborativo, estructurado y tecnológicamente avanzado.

A partir de esta visión general, se definen los siguientes objetivos generales y específicos que guiarán el desarrollo del proyecto.

### **Objetivos generales**

**1. Facilitar el acceso, organización y compartición de apuntes universitarios.**

Desarrollar un sistema centralizado que permita a los usuarios subir, almacenar y consultar materiales académicos de forma estructurada, organizados por universidades, titulaciones, cursos y asignaturas, garantizando la disponibilidad y fiabilidad de los documentos.

**2. Fomentar el aprendizaje colaborativo mediante una plataforma accesible y segura.**

Promover la interacción y el intercambio de conocimiento a través de funcionalidades que permitan compartir documentos de forma pública o privada, asegurando la privacidad y la autoría de los contenidos.

**3. Automatizar la generación de material de estudio mediante Inteligencia Artificial.**

Integrar modelos de lenguaje avanzados capaces de generar automáticamente resúmenes, esquemas, cuestionarios, clases en formato audio y otros recursos, adaptados a las necesidades individuales de los usuarios.

**4. Garantizar la seguridad, privacidad y protección de los usuarios.**

Implementar mecanismos sólidos de autenticación, autorización y control de accesos que aseguren la confidencialidad de los documentos y respeten los derechos de autor de los estudiantes y docentes.

**5. Ofrecer una experiencia de usuario intuitiva, accesible y escalable.**

Diseñar una aplicación web con una interfaz moderna y adaptable a distintos dispositivos, que facilite la navegación y permita la incorporación de futuras mejoras y funcionalidades.

**Objetivos específicos**

- **Gestión de usuarios y control de acceso.** Desarrollar un sistema de registro, autenticación y permisos diferenciados que regule la visibilidad y la compartición de los apuntes.
- **Subida y almacenamiento de documentos.** Implementar funcionalidades para la subida, almacenamiento y visualización de documentos en formatos comunes (PDF, Word, etc.), asegurando su integridad y disponibilidad.
- **Estructuración académica de los apuntes.** Diseñar una organización jerárquica basada en criterios académicos (universidad, titulación, curso, asignatura) que optimice la búsqueda, filtrado y clasificación de los materiales.
- **Integración de herramientas de IA para la mejora del aprendizaje,** incluyendo:
  - Generación de resúmenes automáticos.
  - Creación de esquemas visuales y mapas conceptuales.
  - Consultas interactivas mediante un asistente conversacional.
  - Producción de clases en formato audio.
  - Generación de preguntas tipo test interactivas.
  - Creación de guías de estudio orientadas al repaso.
  - Creación de ejercicios y problemas adaptados al contenido.
  - Extracción del texto de los documentos.

- **Diseño modular y escalable de la arquitectura.** Implementar una base técnica que permita la evolución del sistema.
- **Validación y pruebas.** Realizar pruebas de usabilidad, rendimiento y seguridad para garantizar una experiencia satisfactoria y confiable para los usuarios.

Estos objetivos permitirán desarrollar una plataforma que no solo responda a las necesidades actuales de los estudiantes universitarios, sino que también contribuya a la modernización del proceso educativo, haciendo el aprendizaje más eficiente, colaborativo y adaptado a las nuevas tecnologías.

### **1.3 Estructura de la memoria**

Esta memoria se encuentra organizada en cuatro capítulos, que incluyen toda la información necesaria para comprender este TFG en su totalidad, desde las fases iniciales, hasta las conclusiones y posibles líneas futuras. A continuación, se hace una breve descripción de cada uno de ellos:

#### **1. Introducción**

- 1.1. Motivación:** En este apartado se exponen las razones que justifican la realización del proyecto. Se describen los problemas actuales en el contexto universitario y el impacto de la IA en la educación.
- 1.2. Objetivos:** Aquí se establecen las metas que persigue el trabajo. Se distinguen los objetivos generales y los específicos.
- 1.3. Estructura de la memoria:** En este subapartado se presenta la organización del documento, explicando brevemente el contenido de cada capítulo.

## 2. Estado del arte

- 2.1. **Contexto y evolución del aprendizaje colaborativo:** Como su nombre indica, hace un análisis de cómo ha evolucionado el aprendizaje colaborativo a lo largo de los años.
- 2.2. **Antecedentes:** Se presentan las principales plataformas existentes que guardan relación con el tema del TFG, para comprender los puntos fuertes y las limitaciones de dichas soluciones.
- 2.3. **Tecnologías utilizadas:** Incluye una revisión de las herramientas y tecnologías que se han utilizado. Se estructura en distintos bloques, diferenciando cada componente de la aplicación.

## 3. Desarrollo del proyecto

- 3.1. **Metodología de trabajo empleada:** Se explican los enfoques de gestión y desarrollo que se han adaptado durante el proyecto.
- 3.2. **Investigación y análisis:** Presenta las necesidades detectadas y la evaluación de las distintas tecnologías, hasta llegar a la selección más adecuada.
- 3.3. **Análisis de requisitos:** Aquí se incluye una especificación de los requisitos de la aplicación, tanto funcionales como no funcionales, así como una descripción de los casos de uso principales.
- 3.4. **Diseño de arquitectura:** Este apartado detalla los componentes, patrones empleados y flujos de datos de la aplicación.
- 3.5. **Desarrollo de Backend:** Se describe la implementación de la lógica de negocio, la gestión de datos, la autenticación y la integración con servicios externos.

- 3.6. Integración con modelos de IA:** Aquí se explica la conexión con los sistemas de IA generativa, que permiten mejorar la experiencia de aprendizaje.
- 3.7. Desarrollo de Frontend:** Se describe el diseño y la implementación de la capa del cliente, con las interfaces de usuario.
- 3.8. Pruebas de usabilidad y rendimiento:** Aquí se recogen las pruebas realizadas sobre la aplicación y los resultados que se han obtenido de ellas.
- 3.9. Despliegue de la aplicación:** Documenta el proceso seguido para poner en producción el sistema.

#### **4. Conclusiones y líneas futuras**

- 4.1. Conclusiones:** Se exponen los resultados alcanzados tras el desarrollo del proyecto, evaluando en qué medida se han cumplido los objetivos.
- 4.2. Líneas futuras:** Se identifican oportunidades de evolución del proyecto, incluyendo mejoras técnicas y ampliaciones de funcionalidades.



# 2

## Estado del arte

### **2.1 Contexto y evolución del aprendizaje colaborativo**

El aprendizaje colaborativo constituye un paradigma en el que múltiples usuarios trabajan conjuntamente para alcanzar objetivos de aprendizaje mediante interacción e intercambio de conocimiento. Desde una perspectiva tecnológica, este enfoque se materializa a través de plataformas digitales que facilitan la compartición, organización y procesamiento conjunto de recursos educativos.

#### **Era Pre-Digital (1970-1990)**

Durante este periodo, la colaboración se limitaba a espacios físicos y herramientas analógicas, presentando limitaciones significativas en términos de escalabilidad y persistencia de la información.

#### **Aparición del CSCL (1990-2000)**

La década de 1990 marcó el surgimiento del Aprendizaje Colaborativo Asistido por Computadora (Computer-Supported Collaborative Learning, CSCL).

Esta transición introdujo las primeras plataformas digitales diseñadas para facilitar la colaboración educativa:

- Foros y sistemas de mensajería que permitían intercambio de información sin restricciones temporales.
- Entornos virtuales para la edición colaborativa de documentos.
- Primeras implementaciones de bases de datos educativas estructuradas.

### **Transformación digital (2000-2020)**

El crecimiento de Internet habilitó nuevas arquitecturas de software para el aprendizaje colaborativo. Las características de esta era incluyen:

- Plataformas LMS (Learning Management Systems) centralizadas para gestión de contenido educativo.
- Herramientas de colaboración en tiempo real.
- Migración hasta servicios en la nube.

### **Capacidades emergentes: Inteligencia Artificial (2020-presente)**

La incorporación de IA generativa representa un salto cualitativo en las capacidades de las plataformas educativas. Las principales innovaciones incluyen:

- Procesamiento inteligente de contenidos.
- Personalización adaptativa.
- Automatización de procesos.

La evolución del aprendizaje colaborativo desde sistemas centralizados tradicionales hacia arquitecturas distribuidas potenciadas por IA representa un cambio paradigmático en el desarrollo de software educativo. Las plataformas modernas deben implementar sistemas que sean tanto tecnológicamente avanzados como accesibles para usuarios no técnicos.

## 2.2 Antecedentes

En la actualidad, existe una gran cantidad de aplicaciones que permiten a los estudiantes compartir apuntes con otros compañeros o utilizar la inteligencia artificial para mejorar su estudio. A continuación se hace un análisis de las más relevantes, mostrando de forma detallada sus funcionalidades y las limitaciones que tienen.

### Wuolah

Wuolah es una plataforma española de intercambio de apuntes, donde los estudiantes:

- Suben y descargan apuntes de diversas asignaturas y niveles (Bachillerato, Universidad y máster) de manera gratuita.
- Obtienen recompensas económicas por la descarga de sus materiales.
- Organizan el contenido por curso y asignatura.

Sin embargo, también presenta una serie de limitaciones:

- Pese a ser una plataforma gratuita, los apuntes incluyen anuncios, lo que puede distraer al estudiante.
- Las interfaces de usuario no son rápidas e intuitivas, lo que impide la rápida búsqueda de los apuntes.
- Ausencia de herramientas de IA y de funcionalidades de personalización del aprendizaje.

### Studocu

Studocu ofrece una biblioteca de millones de documentos compartidos por usuarios y categorizados según su naturaleza. Dispone de:

- Búsqueda avanzada y filtros por asignatura, tipo de documento y fecha.
- Vista previa y descarga en varios formatos.
- Acceso gratuito parcial a los documentos.
- Herramientas de IA: generación de notas, cuestionarios, tarjetas didácticas, etc.

En contraposición, también tiene las siguientes limitaciones:

- Gran parte del contenido relevante está bloqueado tras un muro de pago.
- No hay organización jerárquica por universidades, titulaciones y asignaturas de forma nativa.
- Funciona como un repositorio global, lo que impide encontrar fácilmente lo que estás buscando.

### **NotebookLM**

NotebookLM es una herramienta de Google para facilitar la toma de notas potenciada por LLM. Está orientada a:

- Transformar notas en una base de conocimiento consultable mediante chat.
- Proporcionar asistencia para búsqueda y recuperación de información dentro de los cuadernos.
- Generar contenidos con IA como guías de estudio o podcasts de audio sobre los documentos.

En cuanto a sus limitaciones, encontramos las siguientes:

- No permite compartir documentos con otros usuarios al no tratarse de una plataforma colaborativa.
- Es necesario subir tus propias notas para poder hacer uso de sus múltiples funcionalidades.
- Interfaz no adaptada correctamente al estudio para universitarios.

### **Course Hero**

Course Hero se trata de otra plataforma que ofrece distintas funcionalidades de IA a partir de documentos subidos por el usuario. Entre ellas:

- Soluciones paso a paso para problemas académicos.
- Tarjetas didácticas colaborativas y herramientas de estudio interactivas.
- Chat con IA a partir de los apuntes.
- Buscar apuntes sobre tus estudios subidos por otros compañeros.

Pese a ello, también presenta estos inconvenientes:

- Acceso restringido sin suscripción premium, limitando la obtención del contenido completo.
- Calidad de recursos pobre y difícil búsqueda de documentos, al no estar correctamente organizados por titulaciones.
- Usabilidad no adecuada para el estudio.

## **Quizlet**

Quizlet es una plataforma de aprendizaje que ayuda a los usuarios a estudiar ofreciendo técnicas como:

- Tutor personalizado basado en IA.
- Conversión de apuntes en tarjetas didácticas y cuestionarios automáticos.
- Generación de resúmenes de cualquier texto.

En cuanto a sus limitaciones, se encuentran las siguientes:

- Funciones avanzadas disponibles sólo en la versión de pago, limitando el acceso gratuito.
- No existe la posibilidad de compartir apuntes con otros usuarios de forma jerárquica y organizada.

## **Chatbots con Inteligencia Artificial**

El auge de los chatbots con IA, como ChatGPT o Gemini los ha convertido en una herramienta para facilitar el estudio por parte de los universitarios, con funciones como:

- Capacidades multimodales para comprender entradas en distintos formatos.
- Posibilidad de resumir textos extensos, extraer conceptos clave y generar esquemas, entre otros.
- Interfaz conversacional con un amplio conocimiento y múltiples capacidades.

Aun así, estos chatbots también presentan las siguientes limitaciones:

- No dispone de repositorio propio de apuntes, requiere que el usuario proporcione el contenido.
- La organización y gestión de los documentos depende del usuario, sin una estructura nativa por universidades o asignaturas.
- La interfaz no está adecuada a las necesidades de los estudiantes, siendo un inconveniente para el estudio.

## **2.3 Tecnologías utilizadas**

A continuación, se realizará un análisis exhaustivo de las tecnologías seleccionadas para el desarrollo de este TFG. Las tecnologías implementadas forman un stack tecnológico moderno y robusto, que combina escalabilidad, rendimiento y funcionalidades avanzadas de inteligencia artificial.

### **2.3.1. Tecnologías para el Backend**

Para la capa del servidor, se han escogido las siguientes tecnologías:

#### **Node.js**

Node.js se trata de un entorno de ejecución de JavaScript multiplataforma y de código abierto, que permite ejecutar código JavaScript fuera del navegador web. Está construido sobre el motor V8 de Chrome y se ha consolidado como una de las plataformas más potentes para el desarrollo de backend.

Opera con un modelo de un solo hilo y se basa en un bucle de eventos que maneja operaciones asíncronas y no bloqueantes. Esto permite procesar múltiples solicitudes concurrentes sin la sobrecarga de crear múltiples hilos.

Ventajas principales:

- Rendimiento excepcional: Su arquitectura permite manejar miles de conexiones simultáneas, manteniendo un consumo de recursos mínimo.

- Ecosistema rico: Acceso a más de un millón de paquetes NPM, que ayudan a acelerar el desarrollo.
- Unificación del stack: Permite usar JavaScript tanto en frontend como en backend, reduciendo la complejidad del proyecto.
- Escalabilidad: Diseñado para aplicaciones con alta concurrencia y procesamiento en tiempo real.

## **Express.js**

Express.js es un framework web minimalista, rápido y flexible para Node.js que proporciona características esenciales para la creación de aplicaciones web. Como framework no opinionado, Express permite que los desarrolladores estructuren sus aplicaciones siguiendo sus preferencias específicas.

Destaca por su sistema robusto de enrutamiento, que soporta todos los métodos HTTP (GET, POST, PUT, DELETE, etc.), su arquitectura de middleware que facilita el procesamiento de solicitudes, y su compatibilidad con múltiples motores de plantillas.

Características principales:

- Simplicidad y flexibilidad: Proporciona una pequeña capa de características fundamentales, sin imponer una estructura rígida.
- Middleware robusto: Sistema que facilita la implementación de funcionalidades transversales, como autenticación y logging.
- Comunidad madura: Con más de una década de desarrollo, existe una gran cantidad de documentación.
- Rendimiento: Al ser un framework ligero, no compromete la velocidad de las aplicaciones.

## **TypeScript**

TypeScript es un lenguaje de programación de alto nivel desarrollado por Microsoft. Añade a JavaScript el tipado estático con anotaciones de tipo opcionales. Al tratarse de un superconjunto sintáctico, TypeScript compila a JavaScript puro, manteniendo la compatibilidad con el ecosistema ya existente.

Fue diseñado en 2012 y soporta múltiples paradigmas de programación incluyendo funcional, imperativa, genérica y orientada a objetos. Su sistema de inferencia de tipos y verificación en tiempo de compilación, ayuda a prevenir errores comunes antes de ejecutarse.

Ventajas destacadas:

- Detección temprana de errores: El sistema de tipos estáticos identifica los errores durante el desarrollo.
- Experiencia de desarrollo superior: Autocompletado avanzado, refactorización segura y mejor navegación de código.
- Mantenibilidad: Facilita la comprensión del código y el mantenimiento en equipo.
- Compatibilidad: Debido a su compilación a JavaScript, garantiza compatibilidad universal.

## **PostgreSQL**

PostgreSQL es un sistema de código abierto de gestión de bases de datos relacionales. Presenta transacciones con propiedades ACID (atomicidad, consistencia, aislamiento y durabilidad), vistas actualizables y materializadas, disparadores, claves foráneas y procedimientos almacenados.

Sigue una arquitectura cliente-servidor, donde el proceso principal crea una rama con un nuevo proceso para cada conexión de cliente. Soporta tipos de datos personalizados, lenguajes procedurales, restricciones y disparadores avanzados, e indexación sofisticada.

Características superiores:

- ACID Compliance completo: Garantiza atomicidad, consistencia, aislamiento y durabilidad en todas sus transacciones.
- Extensibilidad avanzada: Permite crear tipos de datos personalizados, funciones y extensiones específicas.
- Rendimiento en consultas complejas: Optimizador avanzado e índices sofisticados para garantizar la eficiencia.
- Soporte JSON: Tiene soporte híbrido para datos estructurados y no estructurados.

## **MongoDB**

MongoDB es una base de datos NoSQL potente, escalable y flexible, que proporciona un alto rendimiento y procesamiento de datos en tiempo real. Al contrario que las bases de datos relacionales tradicionales, MongoDB utiliza un modelo orientado a documentos, que permite almacenar y gestionar grandes volúmenes de datos no estructurados o semiestructurados de forma eficiente.

Almacena datos en colecciones de documentos, utilizando el formato BSON (Binary JSON), una versión extendida de JSON que soporta tipos de datos adicionales, como datos binarios, arrays anidados o fechas. Su estructura jerárquica consiste en bases de datos con colecciones, las cuales contienen documentos relacionados.

Ventajas distintivas:

- Flexibilidad de esquema: Los documentos JSON permiten estructuras variables sin restricciones rígidas.
- Escalabilidad horizontal: Tiene fragmentación automática y balanceo de carga para crecimiento exponencial.
- Optimización para documentos: Ofrece una gran eficiencia para almacenar documentos y metadatos.

- Consultas potentes: Lenguaje de consulta expresivo con soporte para agregaciones complejas.

## **Redis**

Redis (Remote Dictionary Server) es un almacén de datos en memoria de código abierto que puede funcionar como base de datos, caché y bróker de mensajes. Almacena todos los datos en memoria RAM en lugar de usar discos duros o SSDs, lo que proporciona velocidades de acceso excepcionales.

Soporta múltiples estructuras de datos, incluyendo cadenas de caracteres, hashes, listas, conjuntos, conjuntos ordenados, mapas de bits (bitmaps) y flujos de datos (streams). Utiliza un modelo clave-valor donde las claves son siempre cadenas de caracteres, pero los valores pueden ser diferentes tipos de datos.

Beneficios principales:

- Velocidad extrema: Tiempos de respuesta ínfimos que permiten más de 100.000 operaciones por segundo.
- Estructuras de datos avanzadas: Soporte nativo para hashes, listas y conjuntos ordenados.
- Persistencia configurable: Permite opciones desde puramente en memoria hasta persistencia completa.
- Escalabilidad: Clustering automático y replicación maestro-esclavo.

## **Cloudflare R2**

Cloudflare R2 es un servicio de almacenamiento de objetos globalmente distribuido y compatible con la API de Amazon S3. Permite almacenar grandes cantidades de datos no estructurados sin los costes de almacenamiento que ofrecen otras soluciones.

Ventajas distintivas:

- **Compatibilidad con Amazon S3:** Utiliza la API de Amazon S3, que se ha convertido en un estándar en este tipo de servicios.
- **Seguridad robusta:** Implementa cifrado AES-256 en reposo y TLS/SSL durante las transferencias, para garantizar la seguridad de los archivos.
- **Escalabilidad global:** Distribución automática de datos a través de la red de Cloudflare para optimizar el rendimiento.

## **Better Auth**

Better Auth es una librería que representa una solución completa de autenticación, diseñada para aplicaciones web modernas. Gracias a sus funcionalidades implementadas, permite establecer un sistema completo sin necesidad de programarlo desde cero.

Características avanzadas:

- **Framework agnóstico:** Funciona con múltiples tecnologías frontend y backend.
- **Funcionalidades implementadas:** Autenticación multi-factor, gestión de roles, OAuth2.
- **Arquitectura de plugins:** Sistema extensible con complementos para añadir funcionalidades complejas ya implementadas.
- **Seguridad robusta:** Implementa las mejores prácticas de seguridad por defecto, aplicando métodos de cifrado que garantizan la privacidad.

### 2.3.2. Tecnologías para el Frontend

En el lado del cliente, se han utilizado estas tecnologías:

#### Next.js

Next.js es un framework React full-stack de código abierto que proporciona herramientas para crear aplicaciones web modernas. Fue desarrollado por Vercel en 2016 y ha evolucionado hasta convertirse en una solución integral para el desarrollo web frontend y backend.

Ofrece renderizado del lado del servidor (SSR), generación de sitios estáticos (SSG), code splitting automático, optimización de imágenes y capacidades full-stack gracias a las API Routes. Gracias a su arquitectura híbrida, se puede elegir la estrategia de renderizado más apropiada para cada página.

Ventajas clave:

- Renderizado del lado del servidor (SSR): Mejora la velocidad de carga inicial al renderizar en el lado del servidor.
- Generación de sitios estáticos (SSG): Pre-generación de archivos para garantizar cargas instantáneas.
- Optimizaciones automáticas: Code splitting, optimización de imágenes y precarga inteligente.
- Capacidades full-stack: API Routes integradas que permiten desarrollar backend en el mismo proyecto.

#### React

React es una librería JavaScript libre y de código abierto desarrollada por Facebook para construir interfaces de usuarios a partir de componentes. Combinada con TypeScript (.tsx) proporciona una base sólida para el desarrollo frontend.

React se basa en una arquitectura basada en componentes, que divide la interfaz de usuario en componentes pequeños, reutilizables e independientes. Cada

componente tiene su propia lógica, estado y estructura, garantizando así una separación clara de responsabilidades y un fácil mantenimiento del código.

Beneficios:

- Reutilización de código: Los componentes pueden ser reutilizados en múltiples partes de la aplicación sin necesidad de repetir el código.
- Mantenibilidad superior: La arquitectura modular permite que sea más fácil de mantener, actualizar y depurar.
- Escalabilidad: Nuevas funcionalidades pueden añadirse como componentes independientes, sin afectar al sistema existente.

## **Tailwind CSS**

Tailwind CSS es un framework CSS utility-first, que simplifica el desarrollo de interfaces web proporcionando un conjunto de clases utilitarias pre-diseñadas. Al contrario que otros frameworks tradicionales que ofrecen componentes pre-construidos, Tailwind proporciona clases utilitarias de bajo nivel, que pueden combinarse para crear diseños totalmente personalizados.

Sigue un enfoque utility-first donde se proporcionan clases pequeñas y reutilizables que cubren propiedades CSS comunes como colores, márgenes, padding, tipografía o grid, entre otros. Este enfoque permite construir diseños complejos y responsivos sin escribir CSS personalizado.

Ventajas distintivas:

- Desarrollo acelerado: Al no tener que escribir CSS personalizado, la experiencia de desarrollo es más rápida.
- Tamaño del paquete optimizado: Genera sólo el CSS necesario, resultando en archivos de menor tamaño.
- Diseño responsivo: Integra un sistema de breakpoints que permite la adaptación a múltiples dispositivos.

- Customización total: Gracias a su archivo de configuración, permite adaptación completa de la paleta de diseño.

## **Shadcn/ui**

Shadcn/ui es una biblioteca de componentes moderna que complementa a Tailwind CSS con componentes pre-construidos de alta calidad.

Al contrario que otras colecciones de componentes cerradas, Shadcn permite el acceso completo al código, dando al desarrollador la posibilidad de analizar su comportamiento y modificarlo a su medida.

Características destacadas:

- Componentes accesibles: Cumple estándares de accesibilidad web.
- Integración perfecta: Está diseñado específicamente para Tailwind CSS y TypeScript.
- Customización completa: Acceso al código de todos los componentes para permitir su personalización.
- Diseño consistente: Sigue un sistema de diseño coherente y profesional, adaptado a las últimas tendencias del sector.

### **2.3.3. Tecnologías para la API de IA**

En la API REST que ofrecerá las funcionalidades de IA, se utilizan las siguientes tecnologías:

#### **FastAPI**

FastAPI es un framework web moderno de alto rendimiento para crear APIs con Python. Combina la velocidad de Starlette para las partes web con Pydantic para las partes de datos.

Genera automáticamente documentación interactiva OpenAPI/Swagger, valida automáticamente tipos de datos usando Pydantic, y soporta programación

asíncrona nativa usando `async/await`. Su arquitectura está optimizada para ofrecer un rendimiento comparable al de Node.js.

Ventajas:

- Rendimiento excepcional: Gracias a su velocidad, rivaliza con otras alternativas para backend como Node.js.
- Validación automática: Pydantic garantiza la integridad de los datos en peticiones y respuestas.
- Documentación completa: Generación automática de documentación Swagger/OpenAPI interactiva.
- Soporte asíncrono nativo: Arquitectura `async/await` para manejar múltiples peticiones de manera simultánea.

## Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, con una sintaxis clara y legible. Se ha convertido en uno de los lenguajes más populares del mundo en la actualidad.

Es un lenguaje multiparadigma que soporta programación orientada a objetos, procedural y funcional. Su naturaleza interpretada permite la ejecución interactiva y el desarrollo rápido, mientras que su tipado dinámico proporciona flexibilidad.

Ventajas distintivas:

- Simplicidad y legibilidad: Su sintaxis se asemeja al inglés, haciendo que el código sea fácil de leer, escribir y mantener.
- Desarrollo rápido: Permite el desarrollo ágil gracias a su sintaxis concisa y bibliotecas extensas para desarrollo web, análisis de datos, inteligencia artificial, computación científica y más.
- Multiplataforma y portable: Funciona en múltiples sistemas operativos, permitiendo que las aplicaciones sean portables sin modificaciones.

- Gestión automática de memoria: Incluye recolección automática de basura, liberando al programador de la gestión manual de memoria.

## **Google Gemini API**

La API de Google Gemini facilita el uso de modelos de IA generativa con capacidades multimodales avanzadas.

Capacidades distintivas:

- Procesamiento multimodal: Análisis simultáneo de texto, imágenes, audio y documentos.
- Ventana de contexto extensa: Capacidad para procesar hasta 1,5 millones de tokens.
- Funcionalidades avanzadas: Text-to-speech, análisis de imágenes y capacidades de investigación web.
- Optimización de costes: Arquitectura eficiente que optimiza los costes operativos.

### **2.3.4. Tecnologías para el despliegue**

Para desplegar la aplicación, se han utilizado estas tecnologías:

#### **Docker**

Docker es una plataforma para la estructuración a través de contenedores de código abierto que permite automatizar el despliegue de aplicaciones dentro de contenedores ligeros y portables. Proporciona una abstracción que empaqueta aplicaciones con todas sus dependencias.

Utiliza una arquitectura cliente-servidor donde el Docker Daemon gestiona los contenedores, imágenes, redes y volúmenes. Los contenedores Docker comparten el kernel del sistema operativo host, al contrario que las máquinas virtuales, lo que hace que sean significativamente más eficientes.

Ventajas fundamentales:

- Portabilidad universal: Los contenedores funcionan de manera idéntica en cualquier entorno que tenga soporte para Docker.
- Eficiencia de recursos: Tiene un consumo mínimo de recursos comparado con la virtualización tradicional.
- Despliegue rápido: Los tiempos de despliegue para las aplicaciones son casi instantáneos.
- Aislamiento seguro: Cada contenedor opera en un entorno aislado y controlado.
- Escalabilidad horizontal: Facilita la creación y destrucción rápida de instancias.

## **Docker Compose**

Docker Compose es una herramienta para definir y ejecutar aplicaciones Docker multi-contenedor utilizando un archivo YAML que configura los servicios, redes y volúmenes de la aplicación. Simplifica la gestión de aplicaciones complejas que requieren muchos servicios interconectados, permitiendo inicializar toda la infraestructura de la aplicación con un solo comando.

Beneficios:

- Gestión simplificada: Control centralizado de múltiples contenedores con comandos unificados.
- Configuración declarativa: Definición completa del entorno en archivos YAML.
- Escalado dinámico: Capacidad de escalar servicios individuales.
- Networking automático: Creación automática de redes para la comunicación entre los distintos servicios.



# 3

## Desarrollo del proyecto

### 3.1 Metodología de trabajo empleada

El desarrollo del presente Trabajo Fin de Grado se ha llevado a cabo siguiendo una metodología incremental e iterativa basada en los principios ágiles, lo que ha permitido combinar la planificación inicial con la flexibilidad necesaria para introducir mejoras durante el transcurso del proyecto. Se han utilizado ciclos cortos de trabajo (sprints) de aproximadamente dos semanas de duración, en los que se han definido objetivos concretos y alcanzables. Cada iteración ha incluido una fase de planificación, la implementación de funcionalidades, pruebas de validación y una revisión final que ha permitido ajustar prioridades y redefinir tareas de acuerdo a los resultados obtenidos.

Este enfoque ha favorecido una evolución progresiva de la plataforma. De este modo, las funcionalidades esenciales se implementaron en primeras fases, lo que facilitó disponer de un prototipo funcional que sirviera como base para la

validación y la retroalimentación, sobre la cual se han ido añadiendo y refinando nuevas características.

Para la gestión y seguimiento del proyecto se han empleado herramientas de control de versiones y coordinación de tareas (GitHub y Trello), lo que ha permitido mantener una trazabilidad completa de los cambios realizados y las mejoras implementadas. La documentación ha acompañado de manera transversal todo el proceso, recogiendo tanto los requisitos funcionales y no funcionales, como las soluciones adoptadas y la justificación de las tecnologías empleadas.

El trabajo se ha estructurado en fases claramente diferenciadas, comenzando con una etapa de investigación y análisis para determinar las necesidades de los estudiantes y estudiar el estado del arte en plataformas similares. Posteriormente se abordó el diseño de la arquitectura, seguido del desarrollo del backend y la integración con modelos de IA, donde se implementaron funcionalidades clave. A continuación, se trabajó en el frontend, diseñando interfaces intuitivas que facilitasen la interacción con los apuntes y las herramientas de IA. Finalmente, se llevaron a cabo pruebas de usabilidad y rendimiento para garantizar la fiabilidad del sistema y se completó con la elaboración de la documentación, manuales de usuario e instalación y el despliegue final de la aplicación.

En conjunto, esta metodología ha permitido no solo cumplir con los objetivos iniciales del proyecto, sino también garantizar la escalabilidad y calidad del software desarrollado, manteniendo un equilibrio entre la planificación previa y la capacidad de adaptación a los retos que han surgido durante el proceso.

## **3.2 Investigación y análisis**

El proceso inicial de desarrollo de este TFG estuvo marcado por una fase de investigación y análisis en la que se abordaron diferentes frentes de manera paralela: la comprensión del contexto y las necesidades de los usuarios, la exploración de las tecnologías disponibles y la evaluación de las distintas alternativas en cuanto a la

incorporación de modelos de Inteligencia Artificial. Esta etapa constituyó la base sobre la que posteriormente se sustentó el diseño y desarrollo de la plataforma y permitió tomar decisiones fundamentales.

### **Exploración de necesidades y primera aproximación**

En un primer momento, el trabajo se centró en analizar con mayor profundidad la problemática identificada en la fase de anteproyecto: la dispersión de apuntes y materiales de estudio entre los estudiantes universitarios. Esta exploración no se limitó a un análisis documental, sino que incluyó también conversaciones informales con compañeros de distintas titulaciones y revisiones de prácticas habituales en entornos de aprendizaje colaborativo. De este modo, fue posible concretar cuáles eran las expectativas más realistas de los potenciales usuarios de la plataforma, lo que ayudó a definir un conjunto de requisitos iniciales que luego servirían de guía para las decisiones tecnológicas.

### **Investigación de tecnologías y definición del stack**

Con una idea más clara de las necesidades, la atención se desplazó hacia la selección de las tecnologías que debían sustentar la plataforma. La elección del stack tecnológico se realizó considerando factores como la escalabilidad, la comunidad, la madurez de las herramientas y la facilidad de integración con servicios de inteligencia artificial.

En el backend, la combinación de Node.js con Express.js se identificó rápidamente como la opción más adecuada, dado su buen rendimiento en aplicaciones web y la amplia disponibilidad de librerías para manejar autenticación, subida de archivos y comunicación con bases de datos. La decisión de emplear TypeScript respondió a la necesidad de contar con un código más robusto y mantenible.

Para la API de IA, se eligió FastAPI, puesto que utiliza Python, un lenguaje ampliamente utilizado en el ámbito de la Inteligencia Artificial.

En cuanto al frontend, Next.js se impuso como la opción natural por su capacidad de renderizado híbrido, lo que ofrecía ventajas tanto en rendimiento como en optimización. Además, se valoró la familiaridad con el ecosistema React.

Respecto al almacenamiento, se optó por una solución híbrida: PostgreSQL para los datos relacionales y MongoDB para la gestión flexible de documentos, con el apoyo de Redis para tareas de caché. El almacenamiento de archivos se resolvió mediante Cloudflare R2 y el despliegue con contenedores gestionados por Docker.

### **Evaluación de opciones de inteligencia artificial**

Uno de los puntos más críticos de esta fase fue la investigación en torno a las distintas opciones de integración de modelos de lenguaje y, en general, de Inteligencia Artificial Generativa. Desde el principio, la IA era el núcleo diferenciador del proyecto, ya que permitiría transformar la experiencia de los apuntes en algo mucho más interactivo y enriquecedor.

En las primeras exploraciones, se consideró el uso de arquitecturas Retrieval-Augmented-Generation (RAG) mediante LangChain, un framework que destaca por su facilidad para construir aplicaciones que integran recuperación de documentos con generación de lenguaje natural. Esta solución resultaba especialmente atractiva porque respondía bien a la necesidad inicial, permitir consultas directas a los apuntes subidos por los estudiantes. La idea era combinar una base de datos vectorial (vector store) para indexar los documentos con un modelo de lenguaje conectado a través de API, probablemente OpenAI.

Sin embargo, durante el proceso de investigación se produjo un cambio relevante en el panorama tecnológico: la aparición de Gemini 2.5 Flash. Este modelo de Google introducía varias mejoras que lo convertían en una alternativa mucho más atractiva que el esquema inicial basado en RAG. Entre sus características más destacadas, se encontraban una ventana de contexto ampliada, que permitía procesar directamente documentos largos sin necesidad de fragmentarlos; una

capacidad de comprensión multimodal más precisa, especialmente valiosa para trabajar con apuntes en formato PDF; y, por último, un coste de uso menor, que lo hacía una opción muy adecuada para este proyecto.

### **Decisión final sobre la IA**

Tras realizar varias pruebas comparativas, se constató que el enfoque basado en Gemini simplificaba considerablemente la arquitectura del sistema. Mientras que la solución con LangChain requería la implementación y mantenimiento de recuperación documental, embeddings y bases vectoriales, Gemini permitía plantear una integración más directa, enviando los documentos a través de su API y recibiendo respuestas ajustadas sin necesidad de capas intermedias.

Una vez definidas las tecnologías principales y tomada la decisión crítica en torno a la IA, el proceso de investigación y análisis pudo darse por concluido, dando paso a la siguiente fase del proyecto.

## **3.3 Análisis de Requisitos**

A continuación se realizará un análisis de los requisitos del sistema, teniendo en cuenta tanto los requisitos funcionales de la aplicación, como los no funcionales. Además, se hace una descripción de los casos de uso principales.

### **3.3.1 Requisitos funcionales**

RF-1 Gestión de usuarios y autenticación

- RF-1.1 Registro y creación de cuenta:

Los usuarios deben poder registrarse mediante correo electrónico y contraseña o cuentas de Google, con verificación de email obligatoria, validación de datos y nombre de usuario único.

- RF-1.2 Inicio de sesión:  
Los usuarios deben poder iniciar sesión mediante correo electrónico o usuario y contraseña, o cuentas de Google, con sesiones persistentes.
- RF-1.3 Gestión de perfil:  
Los usuarios deben poder ver y editar su información personal, incluyendo configuraciones de privacidad y preferencias.
- RF-1.4 Gestión de roles y permisos:  
El sistema debe gestionar roles de usuario (estudiante y administrador), otorgando a los administradores acceso a funcionalidades avanzadas y permitiendo el baneo temporal o permanente de usuarios.

## RF-2 Gestión de entidades académicas

- RF-2.1 Universidades:  
Los administradores deben poder crear, editar, eliminar y buscar universidades con información detallada.
- RF-2.2 Estudios académicos:  
Los administradores deben poder gestionar los estudios disponibles en cada universidad y los usuarios deben poder inscribirse a ellos.
- RF-2.3 Cursos:  
Los administradores deben poder crear cursos dentro de estudios con su información básica.
- RF-2.4 Asignaturas:  
Los administradores deben poder gestionar asignaturas dentro de cursos con información detallada y los usuarios deben poder marcarlas como favoritas.

## RF-3 Gestión y organización de documentos

- RF-3.1 Subida de documentos:  
Los usuarios deben poder subir archivos asociados a una asignatura, con validación de tipo y tamaño, configuración de privacidad y metadatos.

- RF-3.2 Visualización y búsqueda de documentos:  
Los usuarios deben poder listar, buscar y filtrar documentos con ordenamiento y vista pública para documentos abiertos, incrementando automáticamente el contador de vistas.
- RF-3.3 Gestión personal de documentos:  
Los usuarios deben poder gestionar sus propios documentos, editando metadatos, eliminándolos y consultando estadísticas personales.
- RF-3.4 Moderación de documentos:  
Los administradores deben poder ver, editar o eliminar permanentemente cualquier documento.

#### RF-4 Funcionalidades básicas en Inteligencia Artificial

- RF-4.1 Obtención de resúmenes automáticos:  
Los usuarios deben poder obtener un resumen automático generado por IA a partir del contenido de los documentos.
- RF-4.2 Visualización de esquemas del contenido:  
Los usuarios deben poder ver esquemas estructurados que resuman los puntos clave de sus apuntes, generados por la IA.
- RF-4.3 Consulta interactiva mediante chat:  
Los usuarios deben poder interactuar a través de un chat inteligente que responda preguntas específicas sobre el contenido de los documentos.
- RF-4.4 Conversión a clases en formato audio:  
Los usuarios deben poder transformar el contenido de sus apuntes en clases o resúmenes en formato audio.
- RF-4.5 Generación de preguntas tipo test interactivas:  
Los usuarios deben poder generar preguntas tipo test interactivas para facilitar el proceso de memorización.

- RF-4.6 Creación de guías de estudio:  
Los usuarios deben poder crear guías de estudio completas a partir del contenido de sus documentos.
- RF-4.7 Obtención de exámenes con ejercicios y problemas:  
Los usuarios deben poder obtener exámenes con ejercicios y problemas de práctica adaptados al contenido de sus apuntes mediante IA.
- RF-4.8 Extracción de texto:  
Los usuarios deben poder extraer el contenido textual de los documentos de forma sencilla, incluso cuando se trate de fotografías.
- RF-4.9 Gestión de contenidos generados por IA:  
Los usuarios deben poder visualizar, acceder y eliminar contenidos generados por IA asociados a documentos.

#### RF-5 Sistema de comentarios

- RF-5.1 Comentarios en documentos:  
Los usuarios deben poder crear, ver, editar y eliminar sus comentarios en documentos, con sistema de likes.
- RF-5.2 Moderación de comentarios:  
Los administradores deben poder gestionar y eliminar comentarios inapropiados.

#### RF-6 Sistema de reportes

- RF-6.1 Creación de reportes:  
Los usuarios deben poder reportar documentos seleccionando una categoría y describiendo el motivo.
- RF-6.2 Gestión de reportes:  
Los administradores deben poder visualizar, actualizar estado y eliminar reportes creados por los usuarios.

### 3.3.2 Requisitos no funcionales

#### RNF-1 Rendimiento

- RNF-1.1 Caché:

El sistema debe implementar Redis con caché de universidades, documentos y consultas frecuentes, con invalidación automática al modificar datos.

- RNF-1.2 Optimización de consultas:

El sistema debe usar paginación, índices, lazy loading y compresión de respuestas para mejorar el rendimiento.

#### RNF-2 Seguridad

- RNF-2.1 Autenticación y autorización:

El sistema debe usar Better Auth con verificación de email, protección CSRF, validación y sanitización de datos.

- RNF-2.2 Control de acceso:

El sistema debe implementar middleware de autenticación, verificación de propiedad, roles y permisos para proteger documentos privados.

- RNF-2.3 Seguridad de archivos:

El sistema debe validar tipos MIME y tamaño de archivos, almacenarlos de forma segura y proteger contra archivos maliciosos.

#### RNF-3 Usabilidad

- RNF-3.1 Interfaz intuitiva:

La aplicación debe ofrecer diseño responsivo, navegación clara, mensajes de error informativos, estados de carga y tema claro/oscuro.

- RNF-3.2 Experiencia de usuario:

La aplicación debe ofrecer búsqueda en tiempo real, drag and drop para la subida, previsualización de documentos y notificaciones toast.

#### RNF-4 Escalabilidad

- RNF-4.1 Arquitectura:

El sistema debe estar basado en microservicios, con PostgreSQL, MongoDB y almacenamiento de archivos en la nube.

- RNF-4.2 Optimización:

El sistema debe estar optimizado para garantizar su correcto funcionamiento.

#### RNF-5 Disponibilidad

- RNF-5.1 Robustez:

El sistema debe manejar errores, registrar operaciones y recuperarse de fallos.

- RNF-5.2 Mantenimiento:

El sistema debe incluir documentación técnica y código modular que permitan su correcto mantenimiento.

#### RNF-6 Compatibilidad

- RNF-6.1 Navegadores:

La aplicación debe ser compatible con navegadores modernos y dispositivos móviles.

- RNF-6.2 APIs:

La API debe ser RESTful y contar con limitación de ratio para evitar abuso.

### 3.3.3 Casos de uso

Los casos de uso representan las principales interacciones entre los usuarios y la plataforma. A continuación, se describen los más relevantes, organizados según las funcionalidades clave:

#### CU-1 Registro y autenticación de usuarios

- Actores: Estudiante, Administrador.
- Descripción: El usuario crea una cuenta proporcionando sus datos y verifica el correo electrónico. Posteriormente puede iniciar sesión con credenciales o proveedor externo (Google), obteniendo acceso según su rol.
- Precondiciones: El usuario no debe tener una cuenta previa (en caso de registro).
- Postcondiciones: Sesión activa asociada a un token seguro, acceso a las funcionalidades según permisos.

#### CU-2 Gestión de perfil

- Actores: Estudiante.
- Descripción: El usuario visualiza y edita su información personal (nombre, imagen, contraseña).
- Precondiciones: Sesión iniciada.
- Postcondiciones: Perfil actualizado en base de datos.

#### CU-3 Organización académica

- Actores: Administrador, Estudiante.
- Descripción: El administrador define universidades, titulaciones, cursos y asignaturas. El estudiante consulta esta estructura y se suscribe a las titulaciones o asignaturas que cursa.
- Precondiciones: Datos académicos cargados por un administrador.
- Postcondiciones: Usuario asociado a titulaciones y asignaturas, con acceso rápido a los documentos relacionados.

#### CU-4 Subida y gestión de documentos

- Actores: Estudiante.
- Descripción: El usuario selecciona una asignatura y sube un archivo (PDF, Word, etc.), configurando título, descripción y nivel de privacidad. Posteriormente puede editar metadatos, eliminarlo o consultar estadísticas de visualización.
- Precondiciones: Sesión iniciada, asignatura válida.
- Postcondiciones: Documento almacenado en Cloudflare R2, metadatos registrados en la base de datos.

#### CU-5 Búsqueda y visualización de documentos

- Actores: Estudiante.
- Descripción: El usuario explora o filtra documentos por universidad, titulación, curso, asignatura, nombre o tipo. Puede abrir un documento y visualizarlo directamente desde el navegador.
- Precondiciones: Documentos previamente subidos.
- Postcondiciones: Contador de visualizaciones incrementado.

#### CU-6 Uso de funcionalidades de IA

- Actores: Estudiante.
- Descripción: Desde un documento subido, el usuario solicita generación de resumen, esquema, guía de estudio, preguntas tipo test, conversión a audio o interacción vía chat. La IA procesa el contenido y devuelve el resultado.
- Precondiciones: Documento PDF accesible y procesable por IA.
- Postcondiciones: Contenido disponible y vinculado al documento.

#### CU-7 Sistema de comentarios y valoración

- Actores: Estudiante.
- Descripción: El usuario comenta un documento, responde a otros comentarios y valora positivamente los aportes útiles.
- Precondiciones: Documento público compartido.
- Postcondiciones: Comentario almacenado y visible para otros usuarios.

#### CU-8 Moderación y gestión de reportes

- Actores: Administrador.
- Descripción: El administrador revisa reportes de usuarios sobre documentos o comentarios, y toma acciones.
- Precondiciones: Reporte existente.
- Postcondiciones: Estado del reporte actualizado, acción reflejada en el sistema.

### **3.4 Diseño de arquitectura**

Aquí se describe la arquitectura general de la aplicación, enfocándose en los patrones de diseño arquitectónicos, componentes principales y las interconexiones entre ellos. La aplicación está diseñada como un sistema distribuido compuesto por múltiples servicios, cada uno responsable de una funcionalidad específica, siguiendo principios de modularidad y escalabilidad.

La arquitectura se basa en una aproximación de microservicios, donde cada servicio opera de manera independiente pero se comunica con otros a través de interfaces bien definidas. Esto permite una mayor flexibilidad y mantenibilidad.

### 3.4.1 Componentes principales

#### Frontend (Cliente Web)

- Interfaz de usuario principal de la aplicación, responsable de la presentación e interacción con el usuario final.
- Proporciona una experiencia de usuario rica y responsiva, manejando la navegación, formularios y visualización de datos.
- Arquitectura de componentes basada en React, con enrutamiento del lado del cliente.

#### Backend (API Principal)

- Servicio central que maneja la lógica de negocio, autenticación, autorización y comunicación con bases de datos.
- Actúa como intermediario entre el frontend y los servicios de datos, implementando las reglas de negocio y validaciones.
- Arquitectura en capas (MVC), con separación clara entre rutas, controladores, modelos y servicios.

#### AI-API (Servicio de Inteligencia Artificial)

- Servicio especializado en funcionalidades relacionadas con IA, como generación de contenido y procesamiento de datos.
- Proporciona capacidades de IA para enriquecer la experiencia del usuario a partir de los documentos.
- Servicio independiente con API RESTful, diseñado para ser consumido por otros servicios.

#### Bases de datos

- **Base de datos Relacional (PostgreSQL):** Almacena datos estructurados y relaciones complejas, como usuarios, universidades, titulaciones y asignaturas.

- **Base de datos NoSQL (MongoDB):** Maneja datos no estructurados o semiestructurados, como documentos, chats o contenido generado por IA.
- **Sistema de Caché (Redis):** Proporciona almacenamiento en memoria para datos frecuentemente accedidos, mejorando el rendimiento.

### Servicios de almacenamiento

- **Almacenamiento de archivos (Cloudflare R2):** Servicio externo para el almacenamiento seguro y eficiente de archivos, como documentos académicos o imágenes de perfil.

### 3.4.2 Patrones arquitectónicos

#### Patrón de microservicios

La aplicación sigue el patrón de microservicios, donde cada funcionalidad se implementa como un servicio independiente:

- Escalabilidad independiente, despliegue continuo y aislamiento de fallos.
- Cada servicio se ejecuta en su propio contenedor Docker, con comunicación a través de APIs REST.

#### Patrón MVC (Modelo-Vista-Controlador)

Implementado en el backend para separar responsabilidades:

- **Modelo:** Representa la estructura de datos y lógica de negocio.
- **Vista:** Maneja la presentación de datos (respuestas JSON).
- **Controlador:** Gestiona las solicitudes entrantes y coordina entre modelo y vista.

#### Patrón de Repositorio

Utilizado para abstraer el acceso a datos:

- Prisma ORM para la base de datos relacional y Mongoose para la NoSQL, proporcionando una capa de abstracción sobre las consultas.

- Simplifica el acceso a datos, mejora la mantenibilidad y permite cambios en el almacenamiento sin afectar a la lógica de negocio.

### **Patrón Capa de Servicio**

Separación de la lógica de negocio en servicios dedicados:

- Capa de servicios que encapsula operaciones complejas.
- Reutilización de código y separación de responsabilidades.

### **Patrón Cache-Aside**

Implementado con Redis para optimizar el rendimiento:

- Los datos se almacenan en caché después de ser recuperados de la base de datos.
- Reduce la carga en las bases de datos y mejora los tiempos de respuesta.

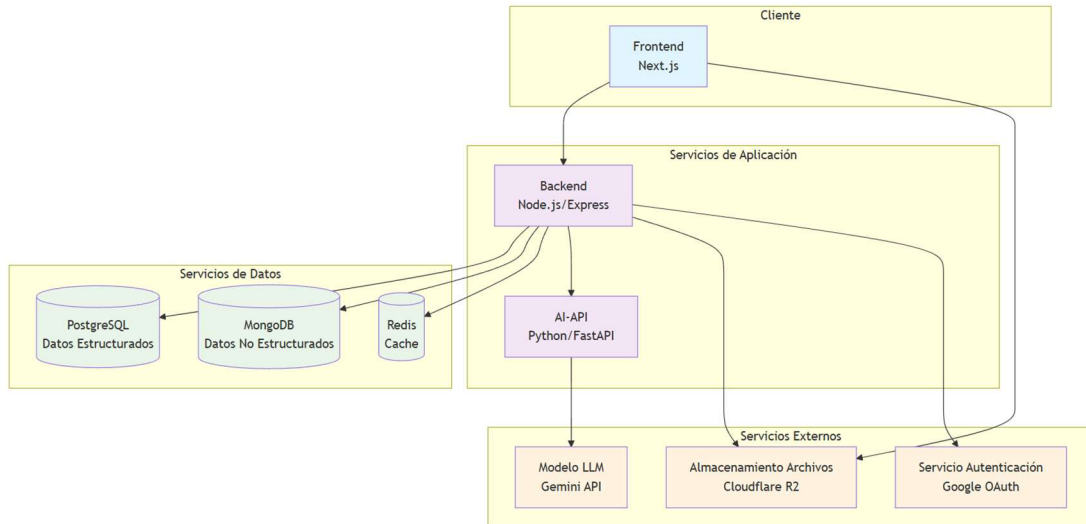
## **3.4.3 Interconexiones y Flujo de datos**

### **Comunicación entre Servicios**

Los diferentes servicios de la aplicación se comunican de la siguiente manera:

- Frontend – Backend: Comunicación HTTP a través de API RESTful.
- Backend – AI-API: Llamadas asíncronas para funcionalidades IA.
- Backend – Bases de datos: Acceso directo a través de ORMs.
- Backend – Redis: Conexión directa para operaciones de caché.
- Backend – Almacenamiento: Integración API externa para subida.

Como se muestra en la **Figura 1**, el sistema se compone de servicios desacoplados que se comunican mediante APIs, reflejando las conexiones descritas anteriormente.



**Figura 1** Diagrama de los servicios que componen la aplicación con sus respectivas conexiones.

### Flujo de autenticación

1. El usuario inicia sesión a través del frontend.
2. El frontend envía credenciales al backend.
3. El backend valida las credenciales contra la base de datos.
4. Se genera un token de sesión que se almacena.
5. El token se devuelve al frontend para futuras solicitudes.

### Flujo de datos académicos

1. El usuario solicita información académica (universidades, titulaciones...).
2. El backend verifica si los datos están en caché.
3. En el caso de que no estén, consulta la base de datos relacional.
4. Los datos se procesan y se devuelven al frontend en formato JSON.

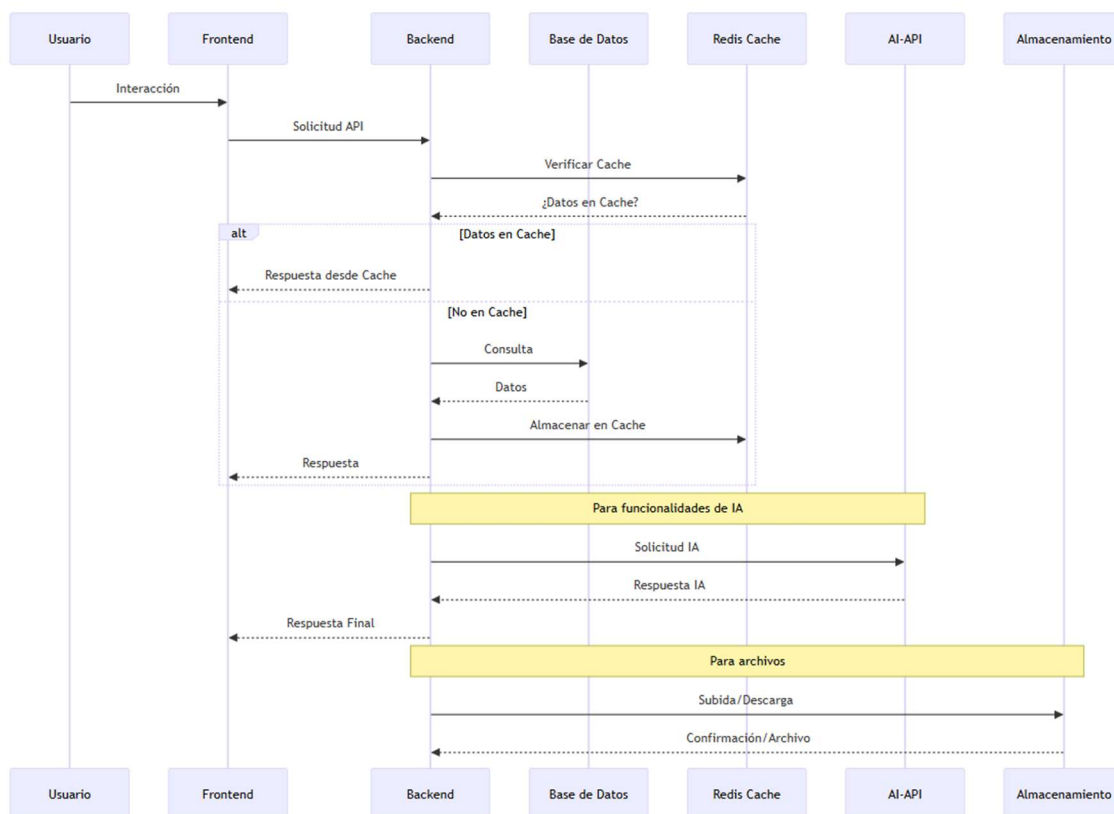
### Flujo de documentos

1. El usuario solicita un documento.
2. El backend verifica si los datos están en caché.
3. En el caso de que no estén, consulta en la base de datos no relacional.
4. Los datos se procesan y se devuelven al frontend en formato JSON.
5. El frontend solicita el documento al servicio de almacenamiento.

## Flujo de IA

1. El usuario solicita una funcionalidad de IA.
2. El frontend envía la solicitud al backend.
3. El backend valida la solicitud y la reenvía a la AI-API.
4. La AI-API procesa la solicitud utilizando servicios externos de IA.
5. El resultado se devuelve al backend, que lo almacena en la base de datos no relacional.
6. El backend devuelve los resultados al frontend.

La **Figura 2** sintetiza el recorrido extremo a extremo de las peticiones entre frontend, backend, almacenamiento, bases de datos y servicios de IA.



**Figura 2** Diagrama de secuencia del flujo de datos de la aplicación.

## 3.5 Desarrollo de Backend

El backend de la aplicación constituye el núcleo central del sistema, encargado de gestionar la lógica de negocio, el acceso a los datos y la comunicación con servicios externos. Está desarrollado bajo una arquitectura modular y escalable, implementando patrones de diseño modernos que garantizan la mantenibilidad y evolución del sistema a lo largo del tiempo.

Aquí se describe de manera detallada la implementación interna del backend, enfocándose en los principios arquitectónicos, las decisiones de diseño tomadas y el funcionamiento de cada componente del sistema.

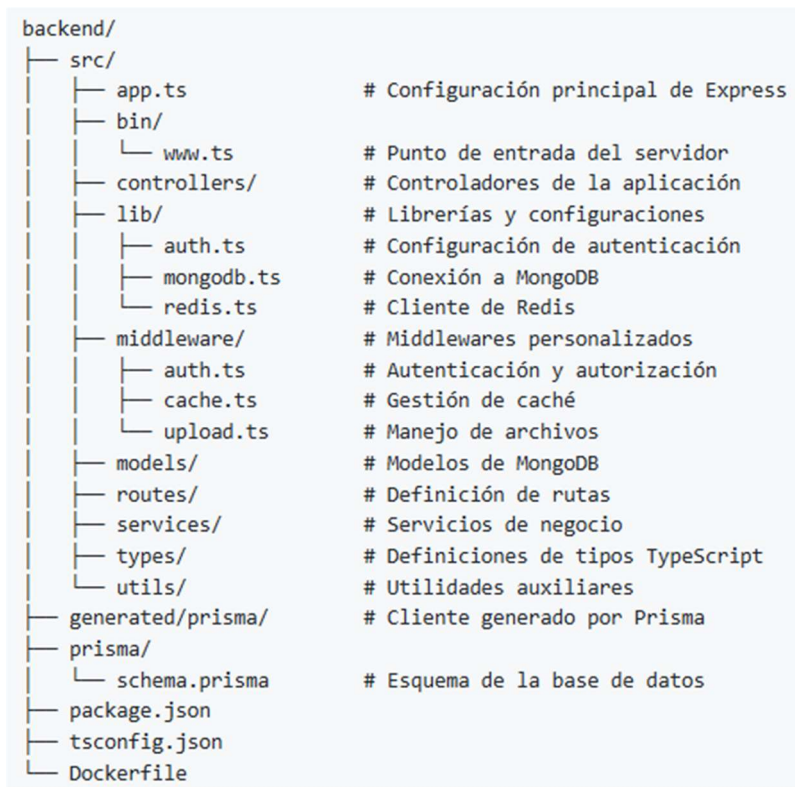
### 3.5.1 Arquitectura General

#### Patrón arquitectónico principal

El backend sigue el patrón arquitectónico Modelo-Vista-Controlador, adaptado al contexto de una API RESTful. Esta separación de responsabilidades permite una organización clara del código donde cada capa tiene responsabilidades bien definidas. El modelo se encarga de la abstracción de los datos y su persistencia, la vista maneja el formato de las respuestas enviadas al cliente, y el controlador coordina entre la entrada del usuario y la lógica de negocio.

#### Estructura organizacional

La organización del código sigue una estructura modular jerárquica que refleja las responsabilidades de cada componente y facilita la navegación y localización de funcionalidades específicas. La **Figura 3** muestra cómo se organiza la estructura de la aplicación.



**Figura 3** Estructura de directorios del backend.

Esta organización divide el código en capas lógicas: configuración principal, capa de presentación con rutas y middlewares, capa de aplicación con controladores y servicios, capa de datos con modelos y utilidades auxiliares.

### 3.5.2 Sistema de Autenticación

#### Arquitectura de autenticación

El sistema de autenticación está construido sobre Better Auth, una librería completa que proporciona una base sólida para la gestión de usuarios y sesiones. Esta implementación incluye múltiples estrategias de autenticación: desde el tradicional email/contraseña hasta la integración con proveedores sociales como Google.

La arquitectura se basa en tokens JWT (JSON Web Tokens) para mantener el estado de autenticación entre solicitudes, combinado con un sistema de sesiones almacenado en base de datos.

## Funcionamiento del proceso de autenticación

Cuando un usuario intenta acceder al sistema, el proceso sigue una secuencia bien definida. Primero se valida la información proporcionada contra la base de datos de usuarios. Una vez confirmada la identidad, se genera un token de sesión que se almacena tanto en el cliente como en el servidor.

Para cada solicitud posterior, el sistema verifica la validez del token incluido en las cabeceras de autenticación. Si el token es válido y no ha expirado, se permite el acceso a los recursos solicitados. En caso contrario, se devuelve un error de autenticación que requiere que el usuario inicie sesión nuevamente.

En la **Figura 4** se representa el intercambio de mensajes entre cliente y servidor durante la autenticación.

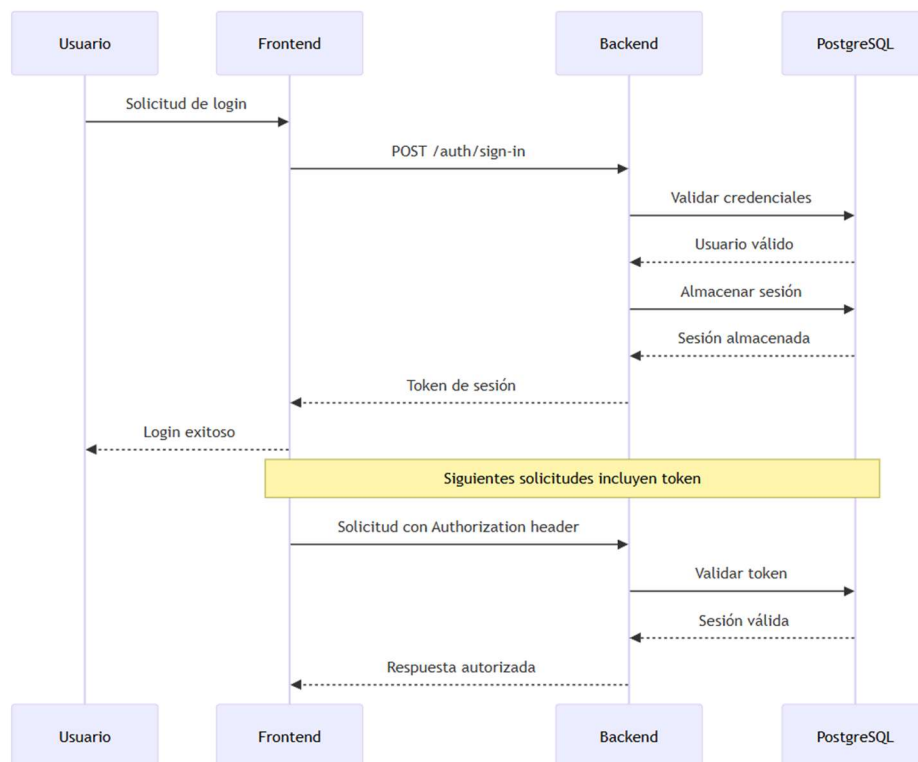


Figura 4 Diagrama de secuencia del proceso de autenticación.

## Gestión de roles y permisos

El sistema implementa una jerarquía de permisos que permite diferenciar entre usuarios regulares y administradores. Los roles se almacenan en la base de datos y se verifican en cada solicitud que requiere autorización especial. Esta

estructura permite un control sobre qué operaciones puede realizar cada tipo de usuario.

## Middleware de seguridad

Se han implementado múltiples capas de middleware para garantizar la seguridad del sistema. El middleware de autenticación verifica la presencia y validez de tokens en cada solicitud protegida. El middleware de autorización controla el acceso basado en roles, mientras que el middleware de validación corrige y valida todos los datos de entrada para prevenir problemas comunes.

### 3.5.2 Gestión de datos

#### Base de datos relacional (PostgreSQL)

PostgreSQL sirve como base de datos principal para almacenar la información estructurada del dominio académico. Utilizando Prisma como ORM (Object-Relational Mapping), el sistema puede interactuar con la base de datos de manera tipada y segura.

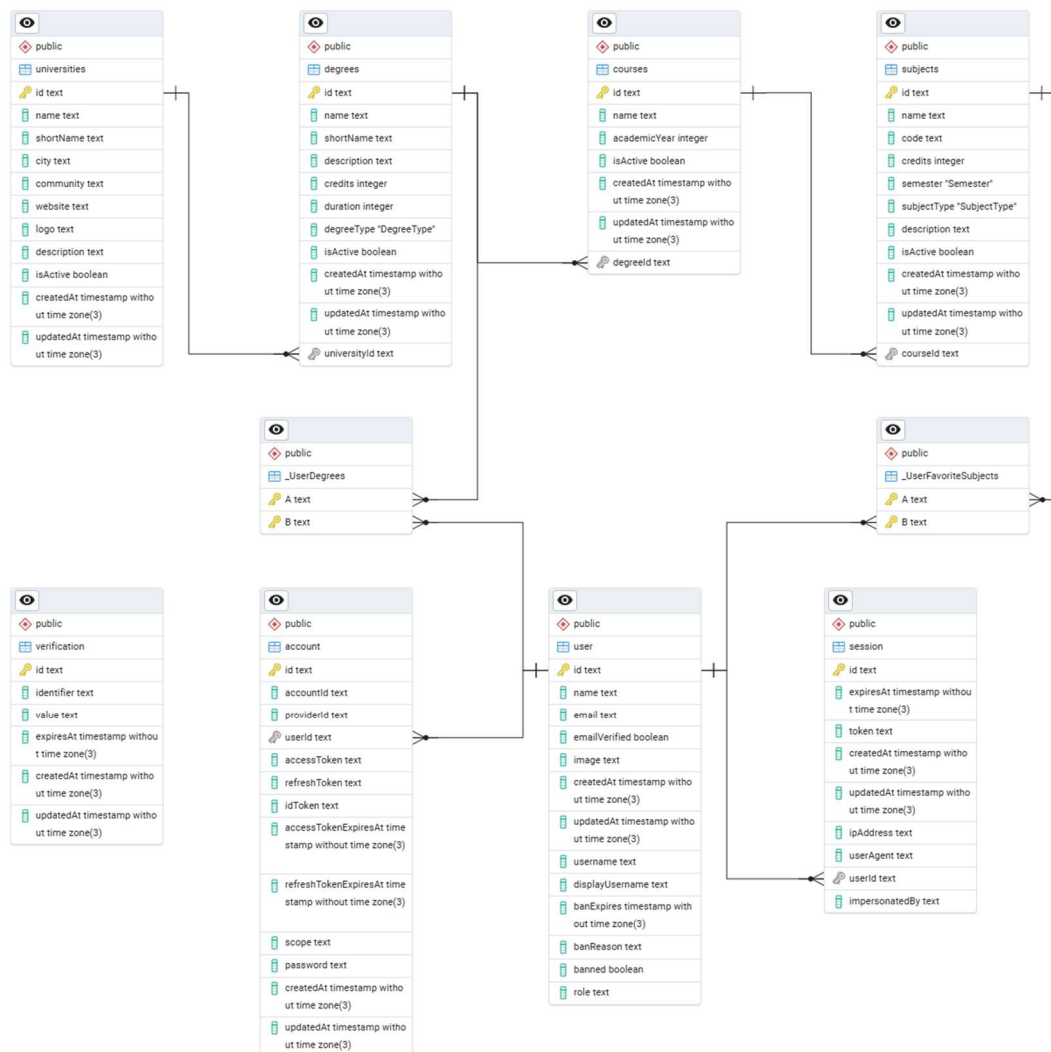
El esquema de datos incluye entidades como usuarios, universidades, titulaciones, cursos y asignaturas, todas relacionadas mediante claves foráneas que mantienen la integridad. Prisma genera automáticamente tipos TypeScript basados en el esquema, lo que reduce errores y simplifica el código.

Estos son los campos de cada entidad:

- **User:** id, name, email, emailVerified, image, createdAt, updatedAt, username, displayUsername, sessions, accounts, role, banned, banReason, banExpires, degrees, favoriteSubjects.
- **Session:** id, expiresAt, token, createdAt, updatedAt, ipAddress, userAgent, userId, User, impersonatedBy.
- **Account:** id, accountId, providerId, userId, user, accessToken, refreshToken, idToken, accessTokenExpiresAt, refreshTokenExpiresAt, scope, password, createdAt, updatedAt.

- **Verification:** id, identifier, value, expiresAt, createdAt, updatedAt.
- **University:** id, name, shortName, city, community, website, logo, description, isActive, createdAt, updatedAt, degrees.
- **Degree:** id, name, shortName, description, credits, duration, degreeType, isActive, createdAt, updatedAt, universityId, university, courses, users.
- **Course:** id, name, academicYear, isActive, createdAt, updatedAt, degreeId, degree, subjects.
- **Subject:** id, name, code, credits, semester, subjectType, description, isActive, createdAt, updatedAt, courseId, course, favoritedByUsers.

De esta manera, el diagrama Entidad/Relación de la base de datos generada es el siguiente (**Figura 5**):



**Figura 5** Diagrama E/R de la base de datos relacional PostgreSQL.

## Base de datos no relacional (MongoDB)

Para el manejo de contenido dinámico y no estructurado, se utiliza MongoDB con Mongoose como capa de abstracción. Esta base de datos almacena documentos académicos, comentario, contenido generado por IA y otros datos que no siguen una estructura rígida.

Mongoose proporciona validación de esquemas y facilita las operaciones CRUD (crear, leer, actualizar y eliminar) sobre los documentos. Los modelos definidos en Mongoose incluyen índices estratégicos que optimizan las consultas más frecuentes.

Los modelos son los siguientes:

- **AIChat:** id, documentId, userId, title, messages, createdAt, updatedAt.
- **AIContent:** id, documentId, userId, type, text, json, raw, audioUrl, audioFormat, classScript, createdAt, updatedAt.
- **Comment:** id, content, documentId, userId, parentId, likes, likedBy, isActive, createdAt, updatedAt.
- **Document:** id, title, description, type, privacy, fileUrl, fileName, fileSize, mimeType, userId, subjectId, views, viewedBy, isActive, createdAt, updatedAt.
- **Report:** id, type, targetId, reason, description, status, priority, reportedBy, reviewedBy, reviewedAt, adminNotes, actionTaken, isActive, createdAt, updatedAt.

La **Figura 6** muestra el modelo de datos implementado en MongoDB, incluyendo las entidades descritas y sus relaciones. Además, se representan los vínculos con la base de datos PostgreSQL (User y Subject), evidenciando la integración entre ambos sistemas de persistencia.

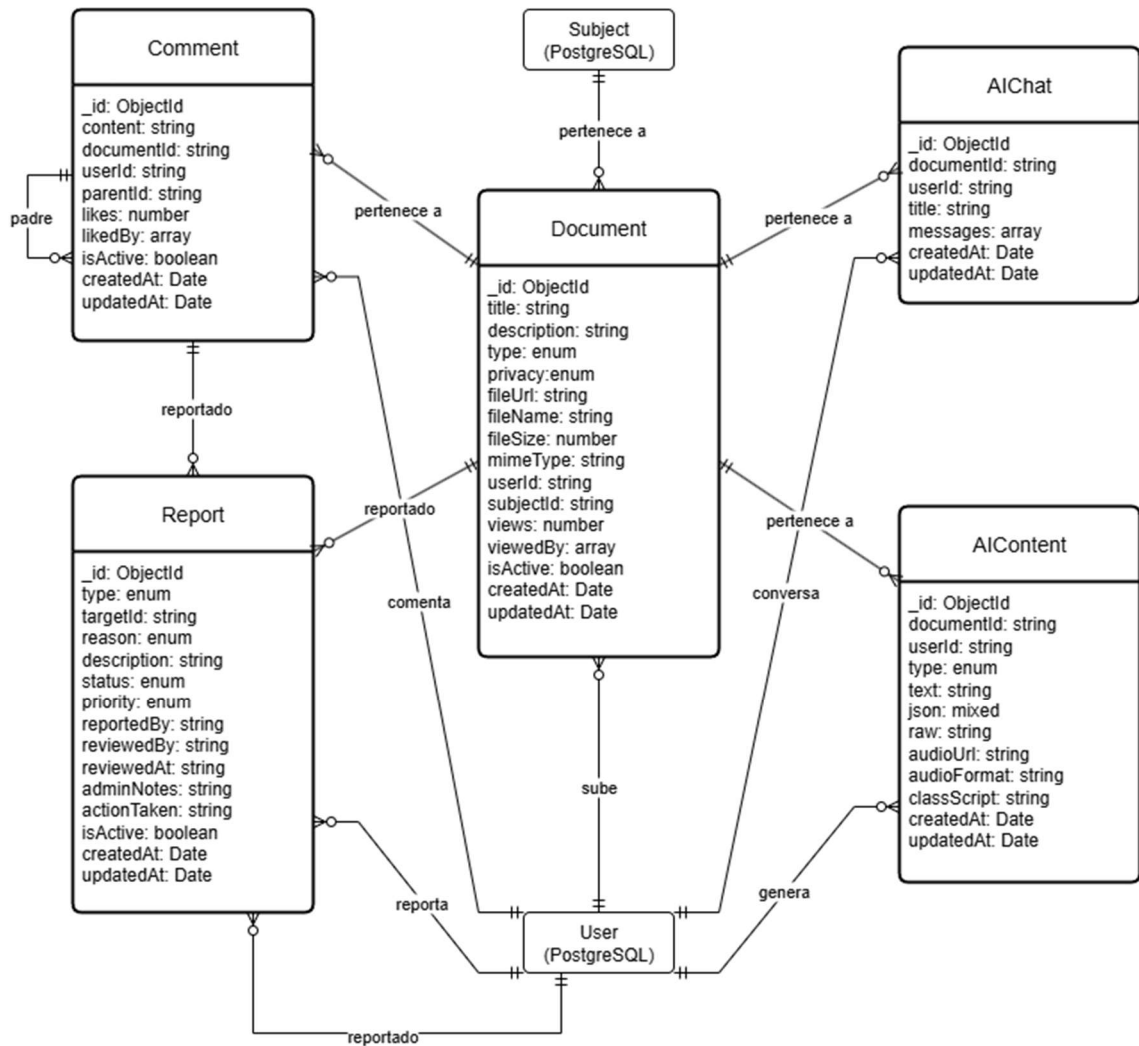


Figura 6 Diagrama de la base de datos no relacional MongoDB.

## Sistema de caché (Redis)

Redis actúa como sistema de caché en memoria, mejorando significativamente el rendimiento del sistema al reducir la carga sobre las bases de datos principales. La implementación sigue el patrón Cache-Aside, donde los datos se cargan en caché después de ser recuperados de la base de datos como se detalla en la **Figura 7**.

El sistema de caché incluye estrategias de invalidación automática que aseguran la consistencia de los datos. Cuando se modifica información en la base de datos, las entradas correspondientes en caché se eliminan o actualizan para reflejar los cambios.

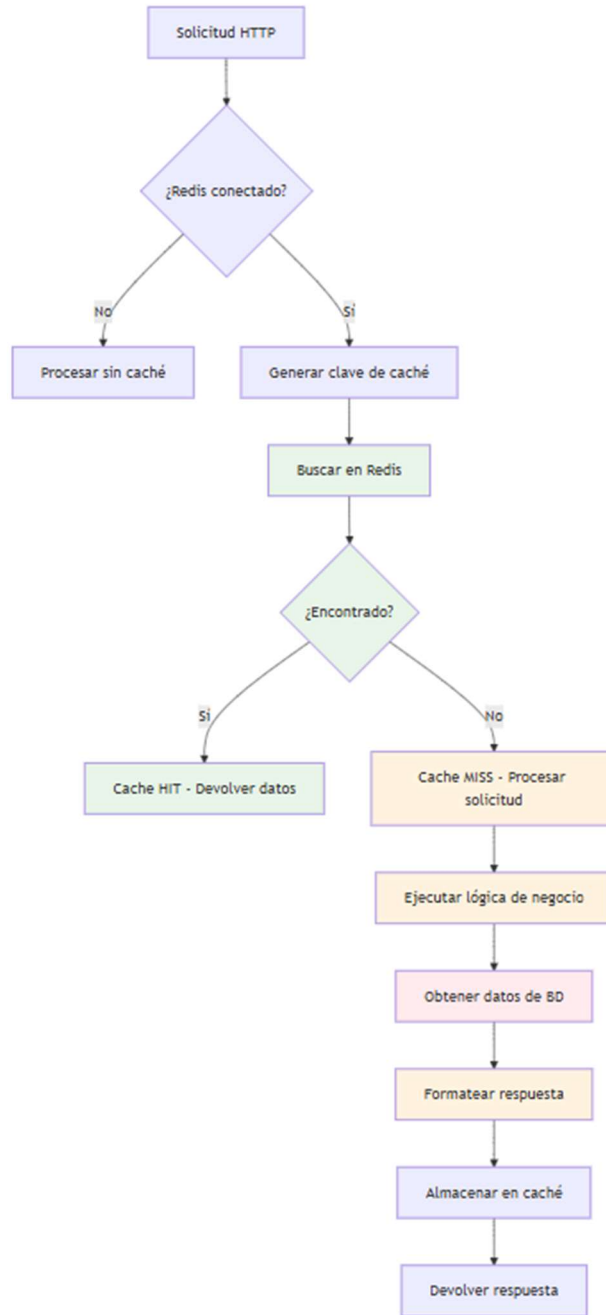


Figura 7 Diagrama de flujo del sistema de caché.

### 3.5.3 Controladores y lógica de negocio

#### Arquitectura de los controladores

Los controladores representan la capa intermedia entre las rutas HTTP y la lógica de negocio. Cada controlador se especializa en un dominio específico del

sistema, como universidades, documentos o usuarios, y coordina las operaciones correspondientes.

El funcionamiento de un controlador sigue un patrón consistente: recibe la solicitud HTTP. Valida los parámetros de entrada, coordina con los servicios de negocio para ejecutar la lógica necesaria, y finalmente formatea la respuesta antes de enviarla al cliente.

### **Validación y manejo de errores**

Cada controlador implementa validación exhaustiva de los datos de entrada para asegurar que solo se procesen solicitudes válidas. Los errores se manejan de manera estructurada, proporcionando mensajes claros y códigos de error específicos que facilitan la depuración y mejoran la experiencia del usuario.

### **Coordinación con servicios**

Los controladores actúan como orquestadores, coordinando llamadas a múltiples servicios según sea necesario. Por ejemplo, al crear un nuevo documento, el controlador podría interactuar con el servicio de almacenamiento de archivos, el servicio de base de datos y el servicio de caché, asegurando que todas las operaciones se ejecuten de manera consistente.

#### **3.5.4 Servicios de negocio**

### **Patrón de servicio**

Los servicios encapsulan la lógica de negocio del sistema, proporcionando una capa de abstracción sobre las operaciones de datos. Cada servicio se enfoca en un dominio específico y contiene toda la lógica necesaria para manejar las operaciones relacionadas con ese dominio.

## Servicios principales

El sistema incluye varios servicios especializados que manejan diferentes aspectos del negocio:

- **Servicio de usuarios:** Gestiona perfiles, preferencias y autenticación.
- **Servicios académicos:** Manejan universidades, titulaciones, cursos y asignaturas.
- **Servicio de documentos:** Controla el ciclo de vida de los apuntes.
- **Servicio de IA:** Coordina las interacciones con el sistema de Inteligencia Artificial.
- **Servicio de Cloudflare:** Gestiona el almacenamiento de documentos en Cloudflare R2.

### 3.5.5 Integración con servicios externos

#### API de Inteligencia Artificial

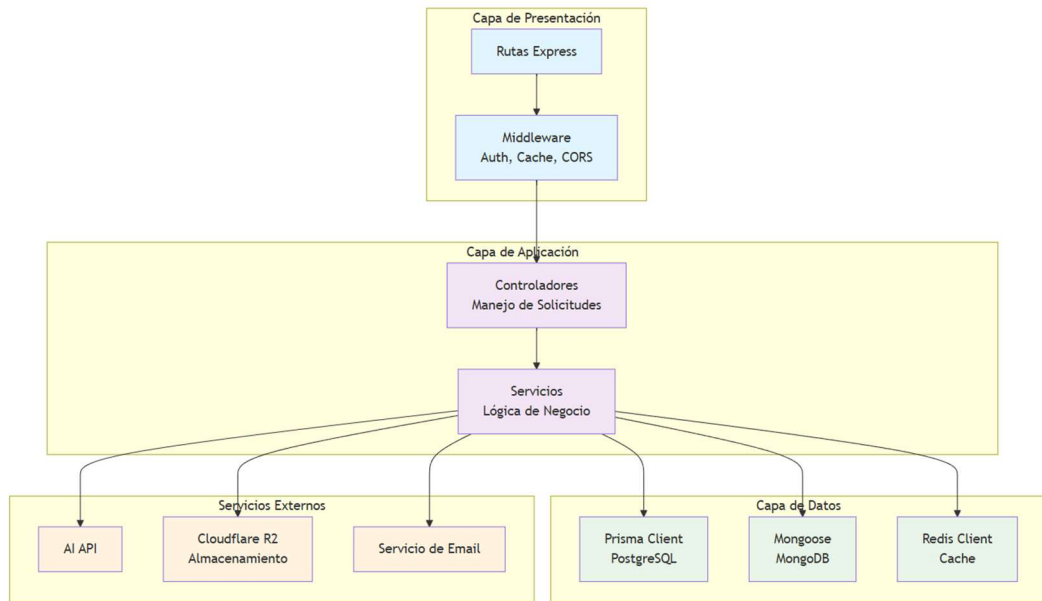
El backend actúa como intermediario, enviando solicitudes con Axios al servicio de IA y procesando las respuestas antes de devolverlas al cliente.

#### Servicio de Almacenamiento

Para el manejo eficiente de archivos, se utiliza un servicio de almacenamiento externo que proporciona escalabilidad y disponibilidad global. Los archivos se suben al servicio externo y se almacenan referencias en la base de datos local, optimizando tanto el rendimiento como los costos de almacenamiento.

#### Servicio de Correo Electrónico

Las comunicaciones por email se gestionan a través de un servicio especializado haciendo uso de la librería NodeMailer. Este servicio incluye funcionalidades como verificación de cuentas o recuperación de contraseñas.



**Figura 8** Diagrama de componentes del backend.

La **Figura 8** presenta los distintos componentes lógicos del backend y su interacción con servicios externos.

### 3.6 Integración con modelos de IA

La API de IA constituye el componente especializado en Inteligencia Artificial de la aplicación, diseñado para procesar documentos académicos y proporcionar funcionalidades avanzadas de análisis de contenido. Desarrollada utilizando FastAPI y el modelo Gemini de Google, esta API actúa como un servicio independiente que se comunica con el backend principal.

El propósito principal de esta API es transformar documentos estáticos en recursos interactivos y pedagógicos, aprovechando las capacidades de procesamiento de lenguaje natural para extraer, analizar y generar contenido educativo de manera automatizada.

#### Diseño de la API

La AI-API sigue una arquitectura de microservicio especializada, centrada en el procesamiento de documentos y la generación de contenido inteligente. Su diseño se basa en principios de simplicidad y eficiencia, manteniendo una interfaz

clara y bien definida con el resto del sistema. Toda la funcionalidad se concentra en un único archivo principal que contiene la lógica completa de procesamiento de IA.

## Modelo de Inteligencia Artificial

Google Gemini constituye el motor de IA principal, integrado a través de la librería oficial de Google:

- **Modelo Gemini 2.5 Flash:** Versión optimizada para velocidad y precisión en tareas de procesamiento de lenguaje.
- **Procesamiento multimodal:** Capacidad para analizar tanto texto como imágenes dentro de documentos PDF.
- **Generación de contenido:** Habilidades para crear texto, diagramas y contenido estructurado.
- **Síntesis de voz:** Funcionalidad integrada para generar audio a partir de texto.

## Modelos de datos

La API define modelos de datos precisos que estructuran todas las interacciones con los clientes:

- **DocumentRequest:** Representa la solicitud básica para procesar un documento PDF.

```
class DocumentRequest(BaseModel):
    pdf_url: str
    output_language: Optional[str] = "es"
    document_title: Optional[str] = None
    subject_name: Optional[str] = None
    degree_name: Optional[str] = None
    university_name: Optional[str] = None
```

- **ChatDocumentRequest:** Modelo específico para interacciones de chat contextual.

```
class ChatDocumentRequest(BaseModel):
    pdf_url: str
    messages: List[ChatMessage]
    output_language: Optional[str] = "es"
    document_title: Optional[str] = None
    subject_name: Optional[str] = None
    degree_name: Optional[str] = None
    university_name: Optional[str] = None
```

## Patrón de llamadas a Gemini

Todas las funcionalidades siguen un patrón consistente de interacción con el modelo de IA:

1. **Preparación del contexto:** Construcción de información contextual académica.
2. **Formulación del prompt:** Creación de instrucciones específicas y detalladas según la funcionalidad buscada. Cada endpoint tiene su propio prompt detallado, que establece de forma clara la respuesta esperada por el modelo y el formato de la misma.
3. **Procesamiento del documento:** Análisis del documento por el modelo de IA.
4. **Generación de respuesta:** Creación del output en el formato solicitado.
5. **Validación y formateo:** Ajuste del resultado para el consumo por el cliente.

En cuanto a la generación de clases en formato audio, se añade un paso intermedio, inmediatamente posterior a la generación de la primera respuesta por parte del modelo. Primero, se solicita el guion que debe ser leído y, este mismo, es enviado de vuelta al modelo con una nueva petición, en este caso para que sea convertido en audio gracias a sus capacidades de **Text-To-Speech (TTS)**. Una vez generado el audio, se procesa y devuelve junto al guion escrito.

## Contexto y prompts

El contexto académico que se incluye es el mismo en todos los casos y se genera con la siguiente función auxiliar:

```
def build_academic_context(document_title: Optional[str], subject_name:
Optional[str], degree_name: Optional[str], university_name:
Optional[str]) -> str:
    context_info = []
    if document_title:
        context_info.append(f"Documento: '{document_title}'")
    if subject_name:
        context_info.append(f"Asignatura: {subject_name}")
    if degree_name:
        context_info.append(f"Grado: {degree_name}")
    if university_name:
        context_info.append(f"Universidad: {university_name}")

    return " | ".join(context_info) if context_info else ""
```

Este contexto es añadido al comienzo de cada prompt y sirve al modelo de IA para comprender de forma concreta dónde se sitúa el documento. Además del contexto, cada prompt incluye la tarea que debe desempeñar, instrucciones concretas, formato de salida, así como otros detalles específicos que puedan ser relevantes según el caso.

Los prompts concretos de cada endpoint son los siguientes:

- Extracción de texto OCR (/ocr):

```
prompt = f"""CONTEXTO: {context}

TAREA: Extraer el contenido completo del documento PDF en formato
markdown.

INSTRUCCIONES:
- Mantén la estructura jerárquica original (títulos, subtítulos,
párrafos)
- Preserva el formato de listas, tablas y elementos estructurados
- Describe imágenes, gráficos y diagramas entre corchetes [Descripción]
- Utiliza formato markdown estándar para todos los elementos
- Conserva la información técnica y académica con precisión
- Omite cualquier marca de agua o texto irrelevante
- Idioma de respuesta: {request.output_language}

FORMATO DE SALIDA: Solo contenido en markdown, sin texto
introdutorio."""
```

- Resumir un documento (/summarize):

```
prompt = f"""CONTEXTO: {context}

TAREA: Crear un resumen académico del documento en formato markdown.

INSTRUCCIONES:
- Identifica y extrae los conceptos clave e ideas principales
- Estructura el resumen con encabezados jerárquicos claros
- Incluye definiciones importantes y teorías fundamentales
- Mantén la precisión académica y terminología técnica
- Organiza la información de manera lógica y coherente
- Extensión: resumen conciso pero completo
- Idioma de respuesta: {request.output_language}

FORMATO DE SALIDA: Solo el resumen en markdown, sin texto
introdutorio."""
```

- Generar preguntas de opción múltiple (/generate\_questions):

```
prompt = f"""CONTEXTO: {context}

TAREA: Generar 10 preguntas tipo test basadas en el documento.

INSTRUCCIONES:
- Crea preguntas que evalúen conceptos clave del documento
- Cada pregunta debe tener exactamente 4 opciones de respuesta
- Solo una opción debe ser correcta
- Nivel de dificultad apropiado para el contexto académico
- Preguntas claras y sin ambigüedades
- Opciones realistas y plausibles
- Idioma: {request.output_language}

FORMATO DE SALIDA REQUERIDO:
Array JSON válido sin texto adicional:
[
  {{
    "question": "Texto de la pregunta",
    "options": ["Opción 1", "Opción 2", "Opción 3", "Opción 4"],
    "correct": 0
  }}
]

IMPORTANTE: Responde únicamente con el array JSON, sin explicaciones ni
texto introdutorio."""
```

- Generar examen (/generate\_exam):

```
prompt = f"""CONTEXTO: {context}

TAREA: Crear un examen completo basado en el documento.

INSTRUCCIONES:
- Incluye 3 tipos de preguntas: opción múltiple, respuesta corta y desarrollo
- Evalúa diferentes niveles: conocimiento, comprensión y aplicación
- Nivel de dificultad progresivo
- Idioma: {request.output_language}
- Determina el número y distribución de preguntas según el contenido del documento
- Estructura el examen de manera lógica y coherente según el tema
- Incluye las respuestas correctas al final

FORMATO DE SALIDA: Examen completo en formato markdown, sin texto introductorio."""
```

- Generar diagrama (/generate\_diagram):

```
prompt = f"""CONTEXTO: {context}

TAREA: Generar un diagrama esquemático en formato Mermaid.

INSTRUCCIONES:
- Identifica los conceptos principales y sus relaciones
- Usa jerarquías cuando sea apropiado (flowchart, mindmap)
- Incluye conexiones lógicas entre elementos
- Mantén claridad visual y legibilidad
- Utiliza sintaxis Mermaid válida
- Idioma de etiquetas: {request.output_language}

TIPOS DE DIAGRAMA SUGERIDOS:
- flowchart TD (para procesos y flujos)
- graph TD (para relaciones conceptuales)
- mindmap (para mapas conceptuales)

FORMATO DE SALIDA: Solo código Mermaid válido, sin explicaciones adicionales."""
```

- Generar guía de estudio (/generate\_study\_guide):

```
prompt = f"""CONTEXTO: {context}

TAREA: Crear una guía de estudio completa y estructurada.

INSTRUCCIONES:
```

- Organiza el contenido en secciones temáticas
- Incluye conceptos clave con definiciones claras
- Añade ejemplos prácticos cuando sea relevante
- Proporciona técnicas de memorización y comprensión
- Sugiere métodos de autoevaluación
- Estructura lógica para facilitar el estudio
- Idioma: `{request.output_language}`
- Determina la estructura y organización según el contenido del documento

FORMATO DE SALIDA: Guía de estudio completa en formato markdown, sin texto introductorio.""

- Generar audio de una clase (`/generate_class_audio`):

```
prompt = f"""CONTEXTO: {context}
```

TAREA: Crear un guión de clase magistral de audio de máximo 10 minutos de duración.

INSTRUCCIONES CRÍTICAS:

- El guión debe estar escrito EXACTAMENTE como debe ser leído en voz alta
- NO incluir comentarios, notas ni instrucciones entre paréntesis
- NO usar frases como "ahora vamos a ver" o "a continuación"
- Estilo natural y conversacional como un profesor experimentado
- Duración: determina el tiempo y extensión según el contenido del documento, máximo 10 minutos (aprox. 1500-1800 palabras)
- Idioma: `{request.output_language}`

ESTRUCTURA RECOMENDADA:

- Introducción
- Desarrollo
- Cierre

ESTILO DE REDACCIÓN:

- Directo y claro, como si hablaras a estudiantes
- Usa transiciones naturales entre ideas
- Incluye pausas naturales con puntos y comas
- Evita jerga innecesaria, explica términos técnicos
- Mantén un tono académico pero accesible

EJEMPLO DE INICIO:

"Buenos días estudiantes. Hoy vamos a explorar..."

FORMATO DE SALIDA: Solo el texto del guión para ser leído directamente, sin títulos ni secciones visibles.""

- Chat sobre un documento (/chat\_document):

```
prompt = f"""ROL: Asistente académico especializado en análisis de documentos.

CONTEXTO: {context}

DIRECTRICES:
- Responde únicamente preguntas relacionadas con el documento proporcionado
- Utiliza información específica del PDF para respaldar tus respuestas
- Mantén precisión académica y terminología apropiada
- Proporciona explicaciones claras y educativas
- Cita secciones relevantes cuando sea necesario
- Idioma de respuesta: {request.output_language}

FORMATO: Respuestas directas sin frases introductorias como "Aquí tienes" o similares.

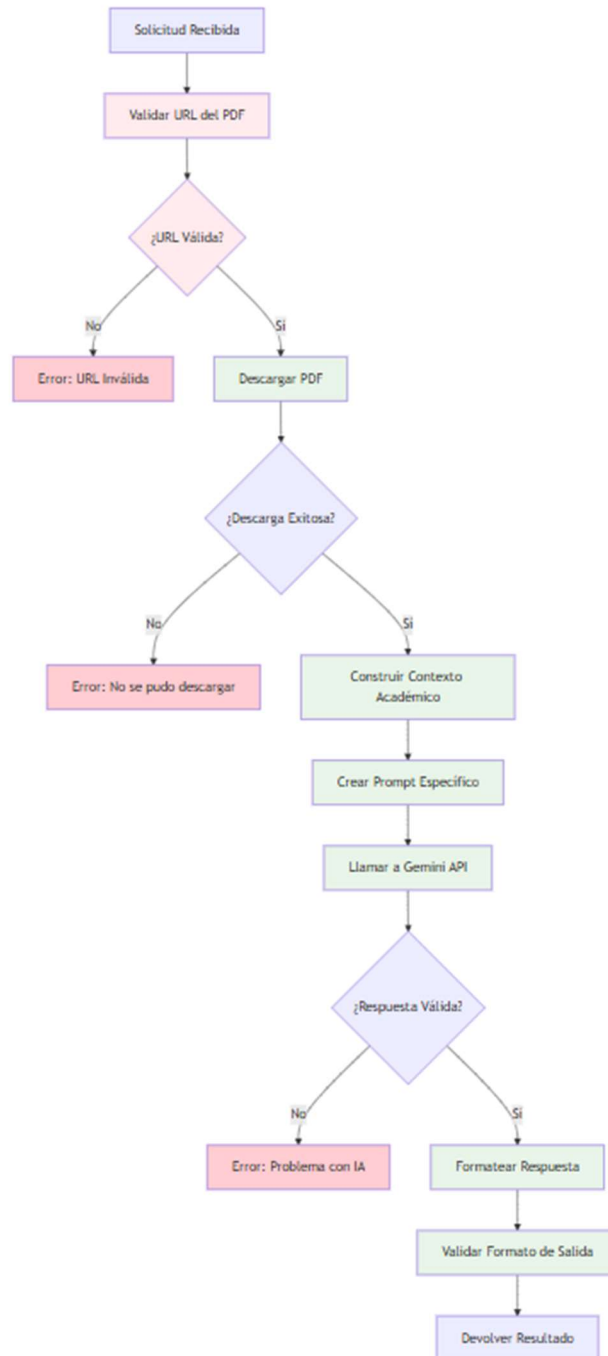
RESTRICCIONES:
- No inventes información que no esté en el documento
- Si no sabes algo, indícalo claramente
- Enfócate en el contenido académico del material"""
```

## Estructura de respuestas

Las respuestas de la API siguen un formato consistente y estructurado que facilita el consumo por parte del backend:

- **Respuestas de Texto:** Contenido generado en formato Markdown.
- **Respuestas JSON:** Datos estructurados para preguntas y evaluaciones.
- **Respuestas multimedia:** Audio codificado en base64.
- **Respuestas de error:** Información detallada sobre problemas encontrados.

El flujo completo desde la petición hasta la respuesta con el resultado se recoge en la **Figura 9**.



**Figura 9** Diagrama de flujo del procesamiento de documentos con IA.

### 3.7 Desarrollo de Frontend

El frontend es la parte más relevante de este TFG, puesto que se trata de la interfaz con la cual los usuarios interactuarán. Por ello, está diseñado específicamente para facilitar el intercambio de apuntes de manera intuitiva, con

funcionalidades avanzadas de inteligencia artificial. De esta manera, se crea una experiencia de estudio óptima para los estudiantes.

La aplicación permite a los estudiantes compartir, buscar y visualizar materiales académicos de manera eficiente y segura, incorporando características de Inteligencia Artificial.

## Arquitectura y tecnologías

La aplicación se ha construido siguiendo una arquitectura sólida que combina las mejores prácticas del desarrollo web moderno. Utiliza Nextjs con su App Router para el enrutamiento del lado del servidor, lo que permite una experiencia de usuario fluida y tiempos de carga rápidos. React 19 es el encargado de manejar la construcción de las interfaces interactivas, mientras que TypeScript con su tipado estático garantiza la robustez del código.

Para el diseño de la interfaz, se utiliza TailwindCSS junto con la colección de componentes shadcn/ui basada en Radix UI, resultando en una apariencia moderna y consistente que se adapta perfectamente a todo tipo de dispositivos.

La autenticación se gestiona mediante Better Auth, utilizando su librería del cliente para conectarse con el backend y gestionar el uso de las sesiones.

## Estructura del proyecto

El proyecto sigue una organización clara y modular que facilita el mantenimiento y la escalabilidad, como se puede observar en la **Figura 10**. La carpeta principal **app/** contiene todas las páginas organizadas según el App Router de Next.js, desde la página de inicio hasta las secciones especializadas como administración, gestión de documentos y funcionalidades de IA.

Los componentes reutilizables se agrupan en la carpeta **components/**, donde se distinguen los componentes de interfaz de shadcn/ui (en **ui/**) de aquellos específicos de la aplicación. Los hooks personalizados, que encapsulan la lógica de

negocio y gestión de estado, se concentran en la carpeta **hooks/**, permitiendo una separación entre la presentación y la lógica.

La carpeta **lib/** tiene las utilidades y configuraciones, incluyendo los servicios que manejan la comunicación con la API, las definiciones de tipos (para manejar los datos devueltos por el backend) y la configuración del cliente de autenticación.

```
frontend/
├── app/                                # Páginas y layouts de Next.js App Router
│   ├── layout.tsx                     # Layout raíz con proveedores globales
│   ├── page.tsx                       # Página principal de la aplicación
│   ├── globals.css                    # Estilos globales de la aplicación
│   ├── favicon.ico                    # Icono de la aplicación
│   ├── admin/                         # Páginas de administración
│   ├── asignaturas/                   # Gestión de asignaturas
│   ├── configuracion/                 # Configuración de usuario
│   ├── documento/                     # Visualización y edición de documentos
│   ├── estudios/                      # Información académica
│   ├── favoritas/                     # Asignaturas favoritas
│   ├── inicio/                        # Página de inicio
│   ├── login/                         # Autenticación de usuarios
│   ├── mi-contenido-ia/               # Contenido generado por IA
│   ├── mis-apuntes/                   # Gestión de apuntes personales
│   ├── mis-chats/                     # Historial de conversaciones IA
│   ├── perfil/                        # Perfil de usuario
│   ├── registro/                      # Registro de nuevos usuarios
│   ├── universidades/                 # Información de universidades
│   └── verificar-email/               # Verificación de email
├── components/                         # Componentes React reutilizables
│   └── ui/                             # Componentes de interfaz básicos de shadcn/ui
├── hooks/                              # Hooks personalizados para lógica reutilizable
├── lib/                                 # Utilidades y configuraciones
│   ├── auth-client.ts                 # Cliente de autenticación
│   ├── utils.ts                       # Funciones auxiliares generales
│   └── services/                       # Servicios de integración
├── public/                             # Archivos estáticos públicos
├── components.json                     # Configuración de componentes UI
├── Dockerfile                          # Configuración de contenedor
├── middleware.ts                       # Middleware de Next.js
├── next-env.d.ts                       # Tipos de Next.js
├── next.config.ts                      # Configuración de Next.js
├── package.json                        # Dependencias y scripts
├── postcss.config.mjs                  # Configuración de PostCSS
└── tsconfig.json                       # Configuración de TypeScript
```

Figura 10 Estructura de directorios del frontend.

## **Componentes y funcionalidades**

El layout raíz establece la estructura base con la configuración de fuentes, el sistema de temas claro/oscuro y las notificaciones globales. El layout protegido asegura que sólo los usuarios autenticados puedan acceder al contenido principal, incorporando una barra lateral de navegación que se adapta al rol del usuario.

La página de inicio para los usuarios no registrados muestra algunas características de la aplicación y guía a los visitantes hacia el registro o inicio de sesión. Una vez autenticados, los usuarios acceden al dashboard principal, que ofrece una visión general de su actividad, documentos recientes y algunos accesos rápidos.

La gestión de documentos es la funcionalidad central, permitiendo a los usuarios subir y visualizar los apuntes. Cada documento se puede clasificar por universidad, estudio, curso y asignatura, facilitando la búsqueda con filtros y paginación. Además, los usuarios pueden marcar documentos como favoritos, controlar su privacidad y acceder a estadísticas de visualización.

## **Sistema de Inteligencia Artificial**

Se ha implementado un chat con los documentos, de forma que los usuarios, a través de una interfaz similar a la de otras aplicaciones de mensajería, puede mantener conversaciones naturales con sus documentos, haciendo preguntas específicas, pidiendo aclaraciones o explorando conceptos.

Además de los chats, también se ofrece la generación automática de diversos tipos de contenido como resúmenes, test interactivos, exámenes, diagramas esquemáticos o guías de estudio. Todos estos, se pueden visualizar de forma sencilla desde la misma página del documento, facilitando el estudio a los usuarios.

También se ha implementado una función de OCR, que permitirá a los estudiantes extraer el texto completo de los apuntes, para casos en los que el documento no tenga texto seleccionable o sean fotografías.

Por último, se ofrece la posibilidad de generar clases en formato audio sobre los documentos, que posteriormente se pueden escuchar con un reproductor de audio integrado en la misma interfaz.

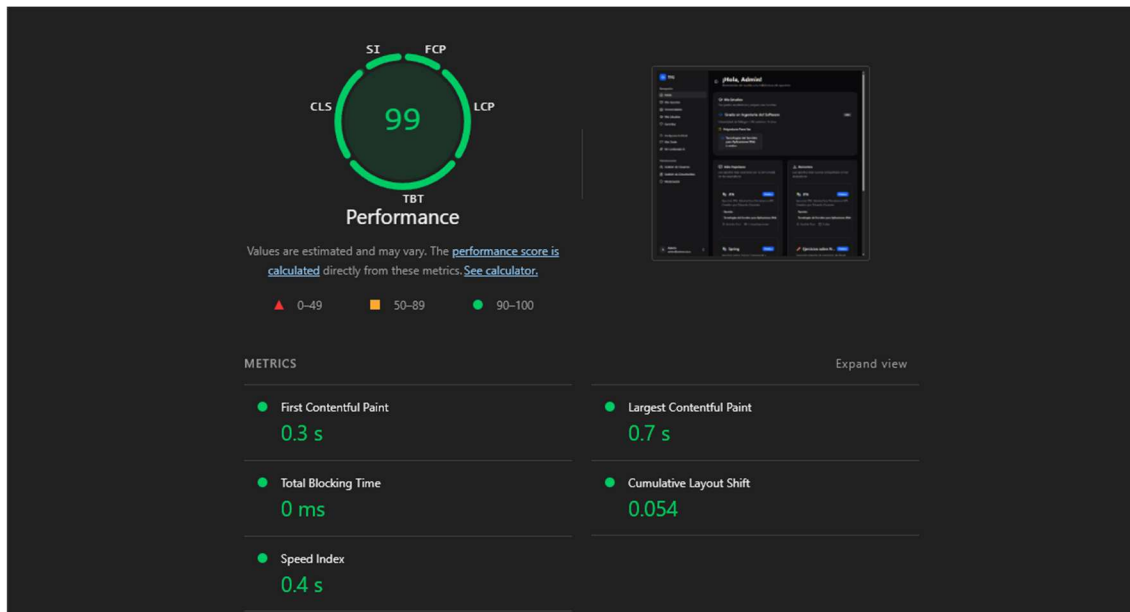
### **3.8 Pruebas de usabilidad y rendimiento**

Una vez finalizada la implementación de la plataforma, resultó fundamental evaluar su funcionamiento mediante diferentes pruebas orientadas a verificar tanto la experiencia de uso como el rendimiento técnico de la aplicación. Este proceso permitió identificar posibles mejoras y garantizar que el sistema cumpliera con los objetivos planteados.

En primer lugar, se realizaron pruebas de usabilidad con un grupo reducido de usuarios pertenecientes al público objetivo: estudiantes universitarios. El objetivo era comprobar si la interfaz resultaba intuitiva, si las funcionalidades principales (subir apuntes, consultar contenidos, generar resúmenes con IA, etc.) podían encontrarse de forma sencilla y si no se presentaban obstáculos. Para ello, se elaboraron pequeñas tareas prácticas (por ejemplo: registrarse en la aplicación, compartir un documento y generar un contenido de IA) y se observó cómo los participantes las llevaban a cabo. Los resultados indicaron que la mayoría pudo completar las tareas sin dificultad, aunque se señalaron algunos aspectos mejorables, como la necesidad de simplificar ciertos menús o resaltar de forma más clara las opciones principales.

En segundo lugar, se llevaron a cabo pruebas de rendimiento, orientadas a evaluar la capacidad de la aplicación para responder con eficiencia bajo distintas condiciones. Estas pruebas se centraron en medir el tiempo de carga de las páginas, la velocidad de respuesta en operaciones críticas (como la búsqueda de apuntes) y la estabilidad general del sistema al incrementarse el número de peticiones simultáneas. Para realizarlas se usó Google Lighthouse, una herramienta de código

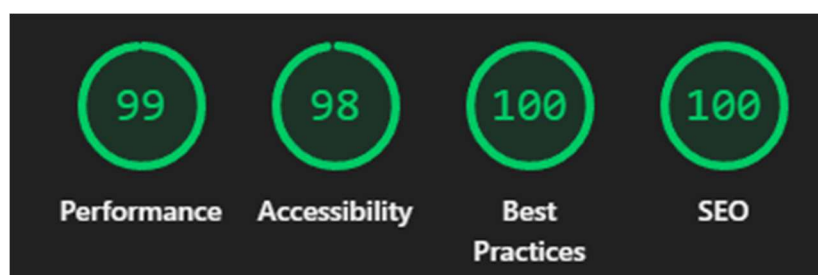
abierto integrada en el navegador que estableció el rendimiento de la página en un 99%, arrojando los siguientes resultados (**Figura 11**):



**Figura 11** Métricas de rendimiento de la plataforma arrojadas por Google Lighthouse.

Los resultados mostraron que la aplicación mantiene un comportamiento estable y tiempos de respuesta adecuados. Sin embargo, se detectó que bajo un uso intensivo el rendimiento puede verse afectado, especialmente en operaciones que dependen de los servicios externos de Inteligencia Artificial. Estas limitaciones no impiden el correcto funcionamiento del prototipo, pero señalan áreas de mejora en futuras versiones.

Con Google Lighthouse también se realizaron otras pruebas que incluían aspectos como accesibilidad, buenas prácticas o SEO. Todas ellas recibieron buenos resultados, como se puede ver en la **Figura 12**, indicando una correcta implementación de la aplicación.



**Figura 12** Resumen de los resultados obtenidos con Google Lighthouse.

En resumen, las pruebas de usabilidad y rendimiento confirmaron que la plataforma es funcional, usable y capaz de ofrecer una experiencia adecuada al usuario final, a la vez que identificaron puntos concretos donde conviene enfocar futuras mejoras.

### **3.9 Despliegue de la aplicación**

Para garantizar un despliegue ordenado y fácilmente mantenible, se ha optado por utilizar Docker como herramienta de estructuración en contenedores. En particular, se ha empleado docker-compose para orquestar de manera conjunta todos los servicios necesarios en un único fichero de configuración. Este enfoque ha permitido encapsular tanto los tres proyectos principales de la aplicación (Express, FastAPI y Next.js), como las tres bases de datos asociadas (PostgreSQL, MongoDB y Redis), garantizando que cada uno de ellos pueda ejecutarse en un entorno aislado y coherente con el resto de componentes del sistema. Gracias a esta infraestructura, resulta posible levantar toda la plataforma con un único comando, reduciendo al mínimo la complejidad y evitando problemas surgidos por diferencias entre entornos de desarrollo y producción.

Además del entorno de ejecución, fue necesario configurar los servicios externos que dan soporte a funcionalidades críticas de la aplicación. En primer lugar, se generaron las claves de OAuth de Google, imprescindibles para habilitar el inicio de sesión mediante cuentas de Gmail. Igualmente, se configuró Gmail como servidor SMTP para Nodemailer, lo que permite el envío automatizado de correos electrónicos de confirmación o validación de credenciales.

Por otra parte, la gestión de los archivos compartidos por los usuarios se ha resuelto mediante la creación de un bucket en Cloudflare R2, un servicio de almacenamiento de objetos compatible con el estándar S3. Para facilitar la integración con todos estos servicios, las claves de acceso y configuraciones sensibles

se almacenaron de forma centralizada en el fichero `.env`, siguiendo buenas prácticas de seguridad y separación de la configuración respecto al código.

Finalmente, la aplicación fue desplegada en un servidor privado virtual (VPS) con Ubuntu Server (Linux) proporcionado por Oracle Cloud, desde el cual se ejecuta en producción y permanece accesible para cualquier usuario que desee visitar la plataforma.

En conjunto, la estrategia de despliegue adoptada combina portabilidad, seguridad y eficiencia operativa, sentando así una base sólida para la evolución futura de la plataforma y su potencial escalabilidad.

# 4

## Conclusiones y líneas futuras

### 4.1 Conclusiones

El Desarrollo de este Trabajo Fin de Grado ha permitido explorar, diseñar e implementar una plataforma orientada al aprendizaje colaborativo, apoyada en modelos de Inteligencia Artificial generativa para optimizar el proceso de creación, compartición y reutilización de apuntes universitarios. A lo largo del proyecto, se han alcanzado los objetivos planteados en la fase inicial, tanto en lo relativo a la construcción de un prototipo funcional como en la validación preliminar de su utilidad en un entorno académico.

Uno de los logros más significativos ha sido demostrar la viabilidad técnica de integrar una arquitectura web moderna (basada en tecnologías como Express, Next.js y FastAPI) con servicios de IA capaces de procesar y transformar contenido textual de forma dinámica. La elección del stack tecnológico ha permitido garantizar escalabilidad, modularidad y facilidad de mantenimiento, aspectos esenciales en un proyecto que aspira a crecer en el future.

Del mismo modo, se ha constatado que el uso de modelos generativos aplicados a los apuntes abre un amplio abanico de posibilidades, desde la coherencia de los materiales, hasta la generación de resúmenes, esquemas o cuestionarios de autoevaluación personalizados. Estas funcionalidades contribuyen a superar la problemática inicial que motivó el trabajo.

En cuanto a la interfaz de usuario, las pruebas realizadas han puesto de manifiesto que la simplicidad en la interfaz y la fluidez en la interacción son factores determinantes para la aceptación de la plataforma. Si bien el prototipo todavía requiere mejoras en aspectos de usabilidad, accesibilidad y rendimiento, los resultados obtenidos son alentadores y evidencian el potencial del enfoque seguido.

Otro aspecto destacable es la aportación de valor académico y social del proyecto. Más allá de su componente tecnológico, la propuesta se alinea con una tendencia creciente en el ámbito universitario: el tránsito hacia ecosistemas de aprendizaje más abiertos, colaborativos y apoyados en tecnologías emergentes.

En resumen, se puede concluir que los objetivos planteados se han cumplido satisfactoriamente, al haber logrado materializar una primera aproximación a una plataforma innovadora, funcional y con potencial de impacto positivo en la comunidad universitaria.

## **4.2 Líneas futuras**

El trabajo aquí presentado constituye un primer paso, pero son numerosas las posibilidades de mejora y extensión que pueden explorarse en futuros desarrollos. Entre ellas, cabe destacar las siguientes:

- Ampliación de funcionalidades basadas en IA:
  - Integración de modelos multimodales capaces de procesar no solo documentos, sino también diagramas, vídeos o grabaciones de voz.

- Implementación de asistentes conversacionales especializados en asignaturas concretas, capaces de responder a preguntas en tiempo real a partir de los apuntes compartidos.
- Generación automática de mapas conceptuales y materiales interactivos que permitan la colaboración a tiempo real con la IA.
- Mejoras en la experiencia de usuario:
  - Optimización del diseño de la interfaz para dispositivos móviles o creación de aplicaciones nativas.
  - Inclusión de mecanismos de accesibilidad avanzados.
  - Desarrollo de sistemas de gamificación que fomenten la participación activa de los estudiantes en la plataforma.
  - Creación de un sistema de notificaciones que mantenga a los usuarios informados de los eventos y novedades.
- Aspectos de escalabilidad y rendimiento:
  - Migración hacia arquitecturas que permitan un despliegue más robusto en entornos de producción.
  - Evaluación del uso de bases de datos distribuidas y servicios en la nube para escalabilidad global.
- Dimensión colaborativa y comunitaria:
  - Establecimiento de un sistema de reputación y validación comunitaria de apuntes que garantice la calidad de los materiales compartidos.
  - Integración con redes sociales académicas o con plataformas institucionales como campus virtuales.
  - Creación de espacios de trabajo en grupo donde los estudiantes puedan editar apuntes de manera conjunta en tiempo real.

- Investigación en aspectos éticos y de seguridad
  - Desarrollo de filtros y mecanismos de verificación que reduzcan la propagación de información incorrecta generada por los modelos de IA.
  - Estudio de técnicas de preservación de la privacidad mediante anonimización y cifrado de datos sensibles.
  - Implementación de sistemas de moderación automáticos potenciados por modelos de lenguaje.
- Validación en entornos reales:
  - Realización de estudios en facultades y grupos de estudiantes para evaluar la adopción de la plataforma en un contexto auténtico.
  - Recogida de métricas sobre el impacto en el rendimiento académico y en la percepción de los estudiantes.
  - Ajuste de las funcionalidades en base al feedback obtenido en los usuarios finales.

En definitiva, este TFG abre un camino de investigación y desarrollo que combina tecnología, pedagogía y colaboración, y que puede evolucionar hacia un producto con un alto grado de impacto en la manera en que los estudiantes universitarios gestionan y comparten su conocimiento

# Bibliografía

Sheldon, R., Denman, J. (2022). *What is Node.js (Node)?*

<https://www.techtarget.com/whatis/definition/Nodejs>

Node.js (s. f.). *Sobre Node.js*®.

<https://nodejs.org/es/about>

Codecademy Team (s. f.). *Express.js Explained: What it is and How to Use it in Your JavaScript Project.*

<https://www.codecademy.com/article/what-is-express-js>

Express (s. f.). *Express. Fast, unopinionated, minimalistic web framework for Node.js.*

<https://expressjs.com/>

Emadamerho-Atori, N. (2024). *The Good and the Bad of Express.js Web Framework.*

<https://www.altexsoft.com/blog/expressjs-pros-and-cons/>

TypeScript (2025), *The TypeScript Handbook.*

<https://www.typescriptlang.org/docs/handbook/intro.html>

Cloudflare (2025). *Cloudflare R2.*

<https://developers.cloudflare.com/r2/>

Ferdous Meem, M. (2025). *Cloudflare R2 Object Storage: What You Need to Know.*

<https://themedev.net/blog/cloudflare-r2-object-storage-what-you-need-to-know>

AWS (s. f.). *What is PostgreSQL?*

<https://aws.amazon.com/es/rds/postgresql/what-is-postgresql/>

GeeksForGeeks (2025). *MongoDB - Working and Features.*

<https://www.geeksforgeeks.org/mongodb/what-is-mongodb-working-and-features/>

IBM (2024). *What is Redis?*

<https://www.ibm.com/think/topics/redis>

Better Auth. (s. f.). *Introduction.*

<https://www.better-auth.com/docs/introduction>

Better Auth (s. f.). *Comparison.*

<https://www.better-auth.com/docs/comparison>

Shah, H. (2025). *Understanding Next.js: Key Features, Benefits, and Use Cases.*

<https://www.prismatic.com/understanding-next-js/>

Next.js (s. f.). *Next.js Docs.*

<https://nextjs.org/docs>

Vamsi Modugula, M. (2024). *A Comprehensive Guide to React.js: The Popular JavaScript Library for Building User Interfaces.*

[https://dev.to/mourya\\_modugula/a-comprehensive-guide-to-reactjs-the-popular-javascript-library-for-building-user-interfaces-d9e](https://dev.to/mourya_modugula/a-comprehensive-guide-to-reactjs-the-popular-javascript-library-for-building-user-interfaces-d9e)

GeeksForGeeks (2025). *What are some advantages of Component Driven Development?*

<https://www.geeksforgeeks.org/reactjs/what-are-some-advantages-of-component-driven-development/>

Tanure, L. (2025). *Tailwind CSS – Utility-First CSS in Action.*

<https://dev.to/letanure/tailwind-css-utility-first-css-in-action-4c0a>

Shadcn/ui (s. f.). *Introduction.*

<https://ui.shadcn.com/docs>

FastAPI (s. f.). *FastAPI framework, high performance, easy to learn, fast to code, ready for production.*

<https://fastapi.tiangolo.com/>

Emanuilov, S. (2024). *What makes FastAPI special — A look at its key features.*

<https://unfoldai.com/fastapi-key-features/>

GeeksForGeeks (2023). *Python Language advantages and applications.*

<https://www.geeksforgeeks.org/python/python-language-advantages-applications/>

Google for Developers (2025). *Gemini API I/O updates.*

<https://developers.googleblog.com/en/gemini-api-io-updates/>

Google AI for Developers (2025). *API de Gemini Developer.*

<https://ai.google.dev/gemini-api/docs?hl=es-419>

Weiss, T. R. (2024). *Exploring Docker for DevOps: What It Is and How It Works.*

<https://www.docker.com/blog/docker-for-devops/>

DockerDocs (s. f.). *Defining and running multi-container applications with Docker Compose.*

<https://docs.docker.com/guides/docker-compose/>

Guminski, L (2015). *Orchestrate Containers for Development with Docker Compose.*

<https://www.cloudbees.com/blog/orchestrate-containers-for-development-with-docker-compose>

Mozilla (s. f.). *MVC.*

<https://developer.mozilla.org/en-US/docs/Glossary/MVC>

Microsoft Learn (s. f.). *Cache-Aside pattern.*

<https://learn.microsoft.com/en-us/azure/architecture/patterns/cache-aside>

StudySmarter (s. f.). *Aplicaciones para estudiar: Wuolah vs StudySmarter.*

<https://www.studysmarter.es/magazine/wuolah-vs-studysmarter/>

CoinCodeCap (2025). *Studocu: The Ultimate Platform for Accessing and Sharing Academics.*

<https://coincodcap.com/studocu-get-all-the-academic-help-in-return-of-them>

Studocu (2025). *The Ultimate Guide to Studocu: Study Notes & AI Tools.*

<https://www.studocu.com/blog/en/the-ultimate-guide-to-studocu-from-study-notes-to-ai-tools>

Ning Shen, Y. (2025). *Limitations of notebooklm every developer should know.*

<https://www.byteplus.com/en/topic/559998?title=limitations-of-notebooklm-every-developer-should-know>

Sarigoz, C. (s. f.). *Course Hero Review: Features, Pricing, Pros & Cons.*

<https://careeradvanceedu.com/course-hero-review/>

Dasgupta, A. (2024). *How to Use Quizlet AI?*

<https://www.primebook.in/blog/how-to-use-quizlet-ai>

Quizlet (2025). *Quizlet Launches New AI-Powered Experience for Back to School.*

<https://www.prnewswire.com/news-releases/quizlet-launches-new-ai-powered-experience-for-back-to-school-302521126.html>

OpenAI (2025). *Presentamos el modo de estudio.*

<https://openai.com/es-ES/index/chatgpt-study-mode/>

London Business School (s. f.). *Collaborative Learning.*

<https://teaching.london.edu/development/teaching-strategies/collaborative-learning/>

The Education Endowment Foundation (2021). *Collaborative learning approaches.*

<https://educationendowmentfoundation.org.uk/education-evidence/teaching-learning-toolkit/collaborative-learning-approaches>

Yang, X. (2023). *A Historical Review of Collaborative Learning and Cooperative Learning*.

<https://pmc.ncbi.nlm.nih.gov/articles/PMC9860218/>

Stahl, G., Koschmann, T., Suthers, D. (2006). *Computer-supported collaborative learning: An historical perspective*.

[https://gerrystahl.net/cscl/CSCL\\_English.pdf](https://gerrystahl.net/cscl/CSCL_English.pdf)

Dilmegani, C. (2025). *12 Digital Transformation Trends & Use Cases in Education*

<https://research.aimultiple.com/digital-transformation-in-education/>

World Economic Forum (2024). *The future of learning: How AI is revolutionizing education 4.0*

<https://www.weforum.org/stories/2024/04/future-learning-ai-revolutionizing-education-4-0/>



# Apéndice A

## Manual de Despliegue

### Requerimientos

#### Para despliegue con Docker:

- Docker: Versión 20.10 o superior.
- Docker compose: Versión 2.0 o superior.

#### Para despliegue manual:

- Node.js: Versión 22.19 o superior.
- Python: Versión 3.11 o superior.
- PostgreSQL: Versión 15 o superior.
- MongoDB: Versión 7 o superior.
- Redis: Versión 6 o superior.
- Npm: Incluido con Node.js.
- Pip: Incluido con Python.

## Configuración de servicios externos

### Creación de claves de Google OAuth:

- Accede a la [Google Cloud Console](#).
- Crea un nuevo proyecto o selecciona uno existente.
- En el menú lateral, ve a APIs y servicios > Pantalla de consentimiento de OAuth y configura los datos básicos.
- En Credenciales > Crear credenciales, genera un nuevo ID de cliente de OAuth.
- Selecciona como tipo de aplicación Aplicación web, dale un nombre y añade las siguientes URIs:
  - Orígenes autorizados de JavaScript:
    - En desarrollo: <http://localhost:3000>
    - En producción: <https://tudominio.com>
  - URIs de redireccionamiento autorizados:
    - En desarrollo: <http://localhost:3000/auth/callback/google>
    - En producción: <https://tudominio.com/auth/callback/google>
- Descarga el archivo con las credenciales y copia los valores GOOGLE\_CLIENT\_ID y GOOGLE\_CLIENT\_SECRET en el fichero .env de la aplicación.

### Configuración de Gmail para Nodemailer:

- Accede a la [cuenta de Gmail](#) que se utilizará para enviar los correos.
- Activa la verificación en dos pasos si no está habilitada.
- Genera una contraseña de aplicación desde la [sección de seguridad](#) de la cuenta.
- Guarda el correo de Gmail (GMAIL\_USER) y la contraseña (GMAIL\_PASSWORD) en el fichero .env.

## Creación de un bucket en Cloudflare R2:

- Accede al [panel de Cloudflare](#).
- En la sección R2 object storage, crea un nuevo bucket indicando el nombre que desees.
- Genera un Access Key ID y un Secret Access Key desde la pestaña de tokens.
- Guarda estos valores en el fichero .env bajo las variables correspondientes:  
R2\_ACCOUNT\_ID, R2\_ENDPOINT, R2\_ACCESS\_KEY\_ID,  
R2\_SECRET\_ACCESS\_KEY, R2\_BUCKET\_NAME y  
R2\_PUBLIC\_URL.

## Variables de Entorno

Crea un archivo .env en la raíz del proyecto con las siguientes variables de entorno, ajustando los valores según tu configuración:

```
BACKEND_PORT=3000
NODE_ENV=development
BETTER_AUTH_SECRET=tu_clave_secreta
BETTER_AUTH_URL=http://localhost:3000
GOOGLE_CLIENT_ID=tu_google_client_id
GOOGLE_CLIENT_SECRET=tu_google_client_secret
DATABASE_URL=postgresql://user:password@postgres:5432/tfg_db
GMAIL_USER=tu_usuario_de_gmail
GMAIL_PASSWORD=tu_contraseña_de_app_de_gmail
FRONTEND_URL=http://localhost:3001
BACKEND_URL=http://localhost:3000
R2_ACCOUNT_ID=tu_r2_account_id
R2_ENDPOINT=https://tu_r2_endpoint
R2_ACCESS_KEY_ID=tu_r2_access_key_id
R2_SECRET_ACCESS_KEY=tu_r2_secret_access_key
R2_BUCKET_NAME=tu_r2_bucket_name
R2_PUBLIC_URL=https://tu_r2_public_url
MONGODB_URI=mongodb://mongo:27017/tfg-documents
REDIS_URL=redis://redis:6379
AI_API_URL=http://ai-api:8000
GOOGLE_API_KEY=tu_google_api_key
NEXT_PUBLIC_API_BASE_URL=http://backend:3000
NEXT_PUBLIC_AUTH_BASE_URL=http://backend:3000/auth
```

## Despliegue con Docker (recomendado)

### Paso 1: Construir y ejecutar los contenedores

- Abre una terminal en el directorio raíz del proyecto, donde se encuentra el archivo `docker-compose.yml`.
- Asegúrate de haber configurado previamente las variables de entorno, como se indica en la sección anterior.
- Ejecuta el comando **`docker-compose up --build`**
- Esto iniciará todos los servicios:
  - PostgreSQL en el puerto 5432
  - MongoDB en el puerto 27017
  - Redis en el puerto 6379
  - AI-API en el puerto 8000
  - Backend en el puerto 3000
  - Frontend en el puerto 3001

### Paso 2: Acceder a la aplicación

- Frontend: <http://localhost:3001>
- Backend API: <http://localhost:3000>
- Usuario/contraseña (perfil administrador): admin/admin

### Paso 3: Detener los contenedores

- Abre una terminal.
- Ejecuta el comando **`docker-compose down`**

## Despliegue manual

### Paso 1: Instalar dependencias del sistema

- Asegúrate de tener instalados PostgreSQL, MongoDB y Redis. Puedes instalarlos y configurarlos localmente o usar Docker para ello:
  - `docker run --name tfg-postgres -e POSTGRES_PASSWORD=password -e POSTGRES_USER=user -e POSTGRES_DB=tfg_db -p 5432:5432 -d postgres:17.5`
  - `docker run -d --name mongo -p 27017:27017 mongo:7`
  - `docker run -d --name redis -p 6379:6379 redis:alpine`

### Paso 2: Configurar la base de datos PostgreSQL

- Asegúrate que existe una base de datos llamada `tfg_db`.
- Ejecuta las migraciones de Prisma (desde el directorio `backend`):
  - `cd backend`
  - `npx prisma migrate dev`
  - `npx prisma generate`

### Paso 3: Desplegar la API de IA

- Abre una terminal y ejecuta los siguientes comandos:
  - `cd ai-api`
  - `python -m venv .venv`
  - En Windows: `.venv\Scripts\Activate.ps1`
  - En Linux/Mac: `source .venv/bin/activate`
  - `pip install -r requirements.txt`
  - `uvicorn main:app --host 0.0.0.0 --port 8000`

#### **Paso 4: Desplegar el Backend**

- Abre una terminal y ejecuta los siguientes comandos:
  - `cd backend`
  - `npm install`
  - `npm run build`
  - `npm run dev`

#### **Paso 5: Desplegar el Frontend**

- Abre una terminal y ejecuta los siguientes comandos:
  - `cd frontend`
  - `npm install`
  - `npm run dev`

#### **Paso 6: Acceder a la aplicación**

- Frontend: <http://localhost:3001>
- Backend API: <http://localhost:3000>
- AI-API: <http://localhost:8000>
- Usuario/contraseña (perfil administrador): admin/admin

# Apéndice B

# Manual de Usuario

## Introducción

Bienvenido a TFG, la plataforma definitiva para compartir y descubrir apuntes universitarios. Esta aplicación web te permite conectar con estudiantes de tu universidad, compartir materiales académicos de calidad y enriquecer tu experiencia de estudio con herramientas de Inteligencia Artificial avanzadas.

Puedes acceder a ella desde cualquier dispositivo a través del siguiente enlace: <https://tfg.andresruiz.es>.

## ¿Qué puedes hacer con TFG?

- **Compartir apuntes:** Sube tus mejores apuntes, resúmenes y materiales de estudio.
- **Descubrir contenido:** Encuentra apuntes de tus asignaturas organizados por universidad, estudios y curso.
- **Conectar con estudiantes:** Forma parte de una comunidad académica activa.
- **Crear contenidos con IA:** Chatea con tus documentos, genera resúmenes automáticos, tests y más.
- **Gestionar tu perfil:** Personaliza tu experiencia académica inscribiéndote en tus estudios y marcando asignaturas favoritas.

Aquí se incluye una guía detallada del funcionamiento de la aplicación, explicando cómo se usa la plataforma en cada situación:

## Registro y acceso

### Creación de cuenta

1. Accede a la página principal. Si no tienes la sesión iniciada, aparecerá una página de bienvenida en la que se muestra información sobre la aplicación.
2. Pulsa en el botón “Comenzar”, “Registrarse” o “Crear cuenta”, todos ellos te llevarán a la página de registro.



**Figura B.1** Página de bienvenida con los botones de Registro resaltados.

3. Elige el método de registro:
  - a. Registro con Google:
    - i. Haz clic en “Registrarse con Google”.
    - ii. Selecciona tu cuenta de Google.
    - iii. Autoriza los permisos solicitados.
  - b. Registro con correo electrónico:
    - i. Completa los campos solicitados con tu información (Nombre, nombre de usuario, email y contraseña).

**Crear cuenta**  
Regístrate con tu cuenta de Google o crea una cuenta nueva

[Regístrate con Google](#)

O regístrate con

**Nombre**  
Tu nombre

**Nombre de usuario**  
usuario123

**Email**  
email@ejemplo.com

**Contraseña**  
Mínimo 8 caracteres

**Confirmar contraseña**  
Repite tu contraseña

[Crear cuenta](#)

¿Ya tienes una cuenta? [Inicia sesión](#)

Al hacer clic en crear cuenta, aceptas nuestros [Términos de Servicio](#) y [Política de Privacidad](#).

**Figura B.2** Formulario de registro (Crear cuenta).

4. Haz clic en “Crear cuenta”.
5. Verifica tu email: Revisa tu bandeja de entrada y haz clic en el enlace de verificación recibido.

## Inicio de sesión

1. Accede a la página de login: Desde la página de bienvenida, haz clic en “Ya tengo cuenta” o “Iniciar Sesión”.



**Figura B.3** Página de bienvenida con los botones de Inicio de sesión resaltados.

2. Elige el método de acceso:
  - a. Autenticación con Google:
    - i. Haz clic en “Iniciar sesión con Google”.
    - ii. Selecciona tu cuenta de Google.
  - b. Autenticación con correo electrónico o usuario:
    - i. Ingresa tu email o nombre de usuario.
    - ii. Ingresa tu contraseña.

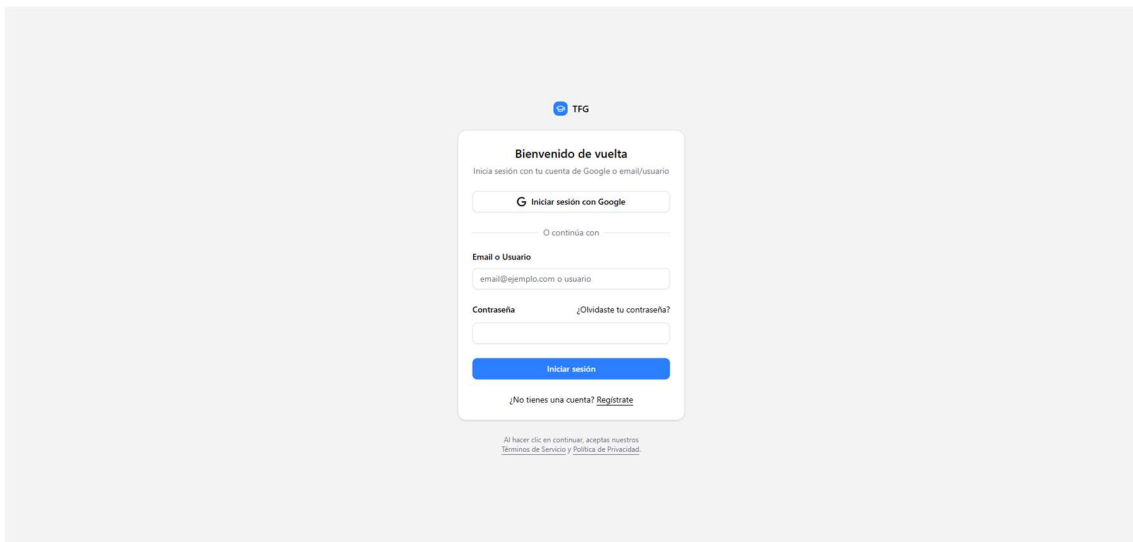


Figura B.4 Formulario de Inicio de sesión.

3. Haz clic en “Iniciar sesión”.
4. Serás redirigido a tu página de inicio.

## Dashboard principal (Inicio)

Después de iniciar sesión, accederás al dashboard principal (/inicio) que muestra:

- Sección “Mis estudios”
  - Muestra los estudios en los que te has suscrito.
  - Lista las asignaturas que has marcado como favoritas.
  - Haz clic en cualquier estudio o asignatura para acceder a su página.

- **Documentos populares**
  - Muestra los apuntes más valorados en tus asignaturas.
  - Haz clic en cualquier documento para verlo.
- **Documentos recientes**
  - Muestra los apuntes más recientes compartidos en tus asignaturas.
  - Haz clic en cualquier documento para verlo.

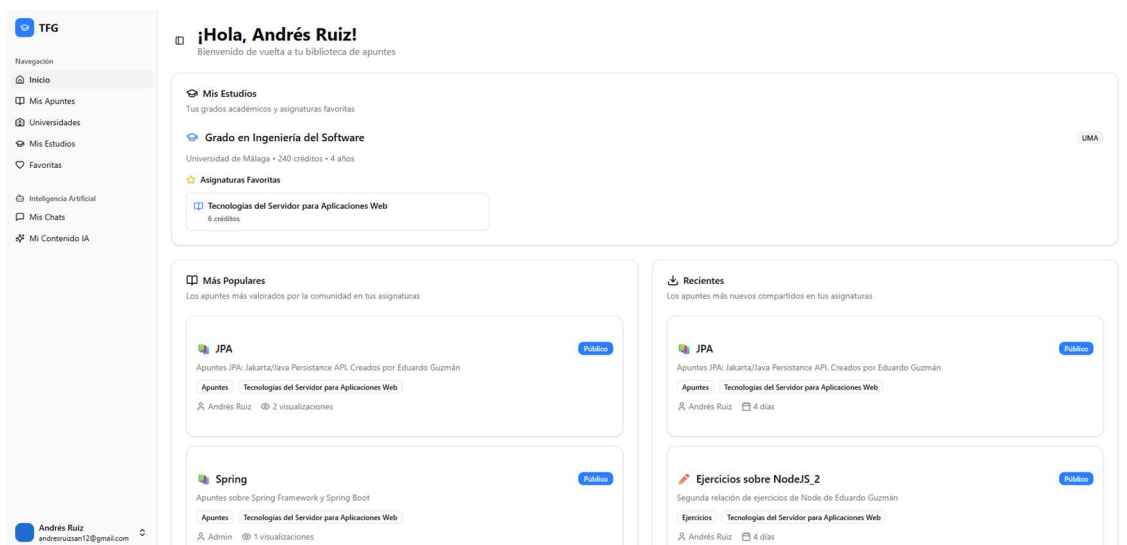


Figura B.5 Dashboard principal (inicio) con las secciones personalizadas.

Si es tu primera vez accediendo a la plataforma, aparecerán accesos directos para ayudarte a personalizar tu experiencia y añadir tus estudios.

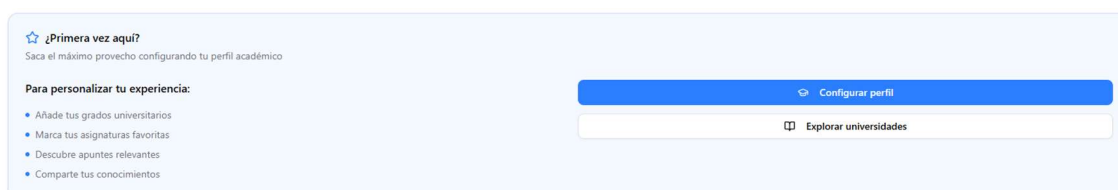


Figura B.6 Consejos y accesos directos para nuevos usuarios.

## Gestión del perfil de usuario

### Acceso al perfil

1. Desde cualquier página, haz clic en tu avatar o nombre en la esquina inferior izquierda de la barra de navegación lateral.
2. Selecciona “Mi perfil” en el menú desplegable.

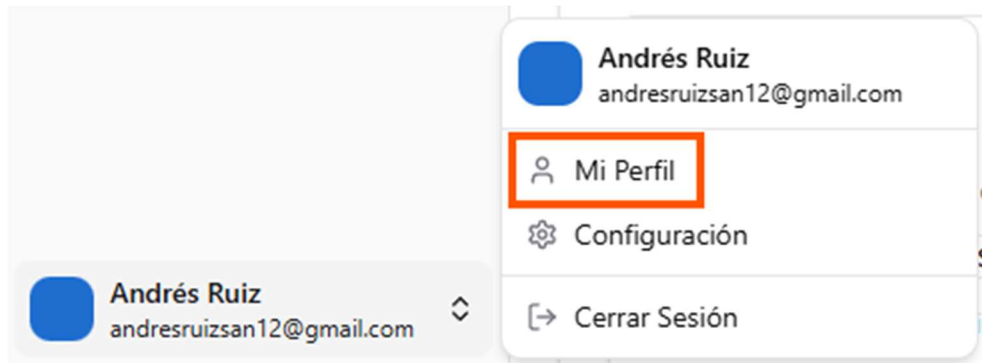


Figura B.7 Menú desplegable con el acceso a “Mi Perfil” resaltado.

3. O ve directamente a /perfil.
4. Podrás ver tu información actual: foto de perfil, nombre, email y usuario.

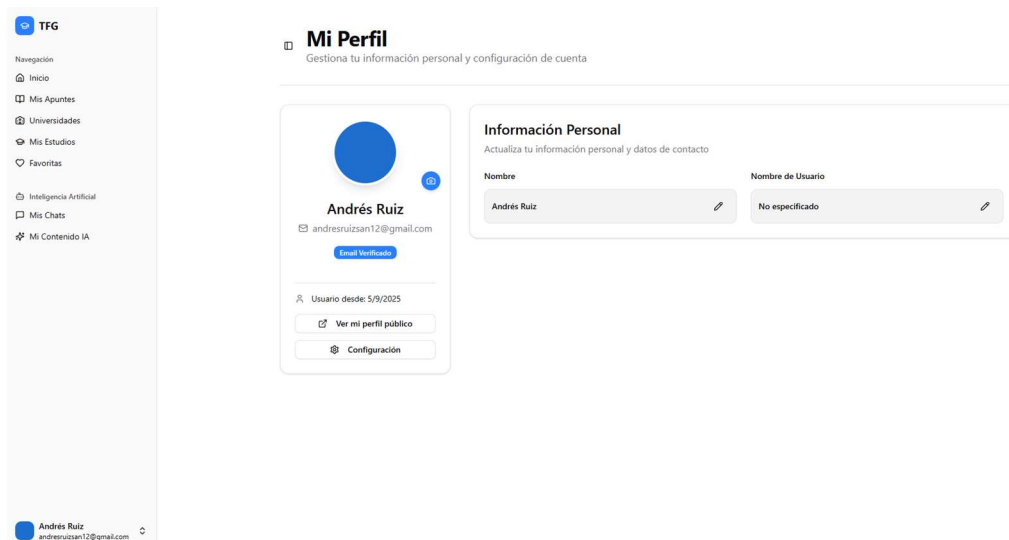


Figura B.8 Página de perfil con la información del usuario.

## Editar información

1. Haz clic en el icono del lápiz situado a la derecha de tu nombre o usuario.

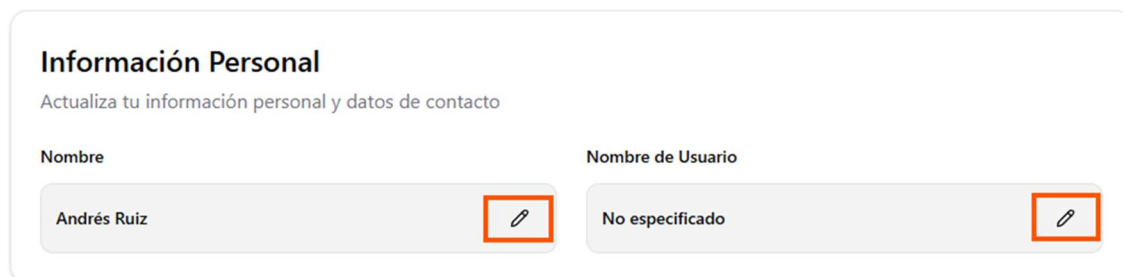


Figura B.9 Botones de edición de la información personal.

2. Modifica el texto e introduce tu nueva información.

**Nombre**

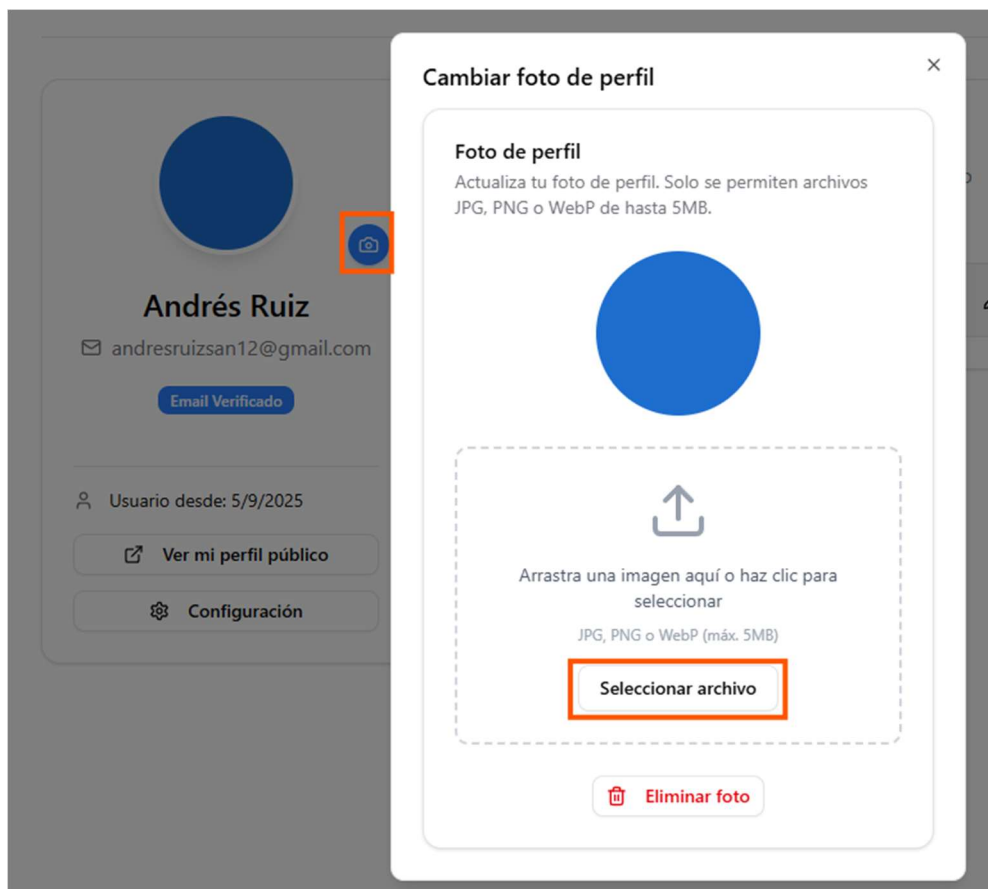
  

**Figura B.10** Edición del Nombre junto a los botones de Confirmar y Cancelar.

3. Haz clic en el icono azul de confirmar para guardar los cambios.
4. O en la X para cancelarlos.
5. Tu información se actualizará automáticamente en la página.

### Modificar foto de perfil

1. Pulsa el icono de la cámara situado a la derecha de tu foto actual.
2. Selecciona una nueva imagen desde tu dispositivo.
3. Si quieres eliminar la foto, pulsa en el botón “Eliminar foto”.



**Figura B.11** Formulario de edición de la imagen de perfil con los botones señalados.

## Ver perfil público

1. Desde tu perfil privado, haz clic en “Ver mi perfil público”.



Figura B.12 Botón “Ver mi perfil público”.

2. Verás cómo aparece tu perfil para otros usuarios.
3. Podrás ver tus estadísticas públicas y documentos compartidos.

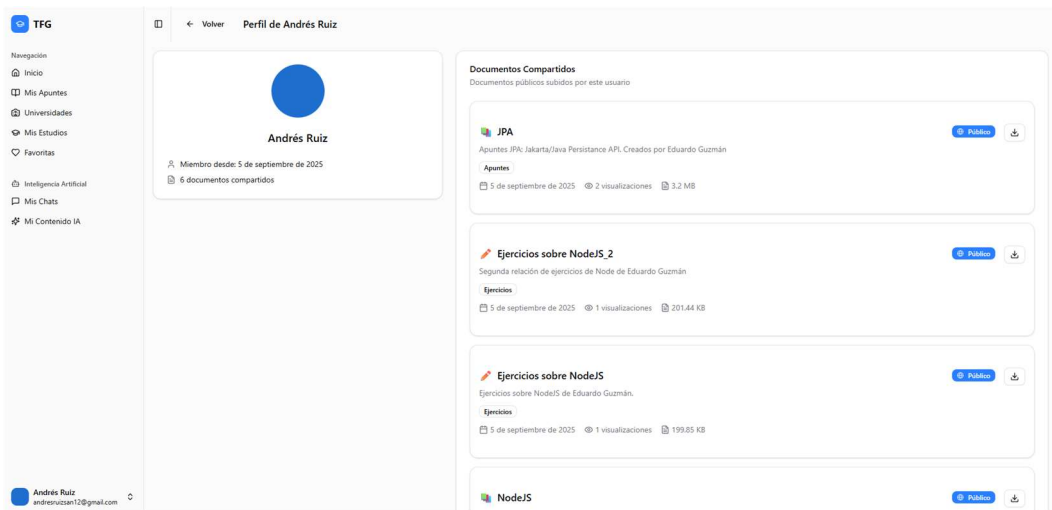


Figura B.13 Perfil público del usuario.

## Configuración

### Acceso a la página de configuración

1. Desde cualquier página, haz clic en tu avatar o nombre en la esquina inferior izquierda de la barra de navegación lateral.
2. Selecciona “Configuración” en el menú desplegable.

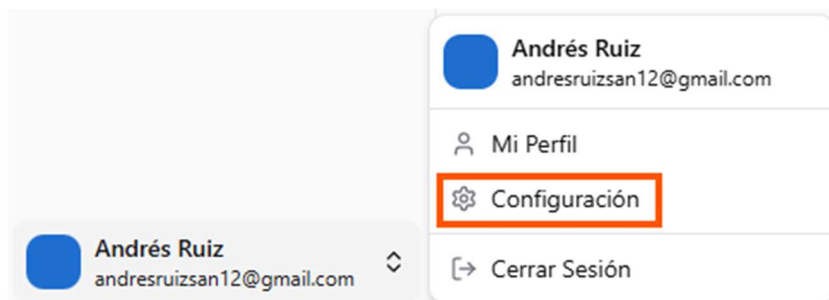
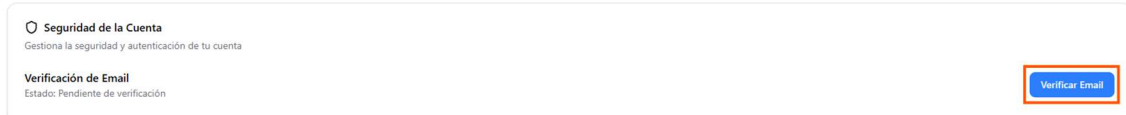


Figura B.14 Menú desplegable con el acceso a “Configuración” resaltado.

3. O ve directamente a /configuración.

## Verificación de email

1. Si aún no has verificado tu email, aparecerá “Pendiente de verificación”.
2. Haz clic en “Verificar email”.

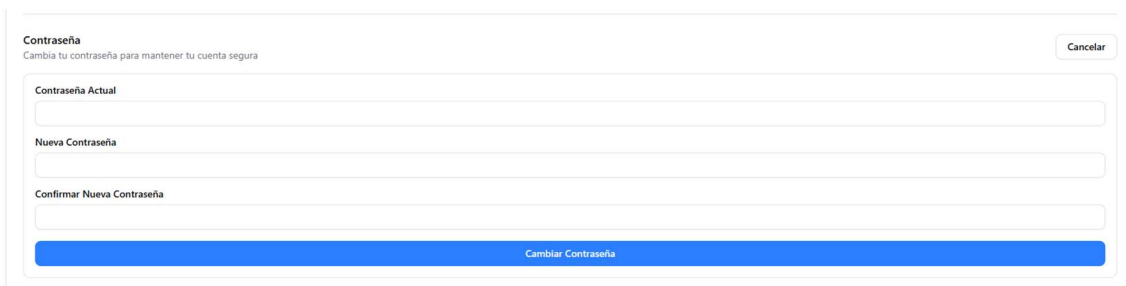


**Figura B.15** Sección de Verificación de email.

3. Revisa tu bandeja de entrada y haz clic en el enlace de verificación recibido.
4. Tu email se verificará automáticamente.

## Cambiar contraseña

1. Pulsa en el botón de “Cambiar contraseña”.
2. Introduce tu contraseña actual.
3. Introduce tu nueva contraseña en dos ocasiones.

The image shows a form for changing a password. At the top left, it says "Contraseña" and "Cambia tu contraseña para mantener tu cuenta segura". At the top right, there is a "Cancelar" button. The form contains three input fields: "Contraseña Actual", "Nueva Contraseña", and "Confirmar Nueva Contraseña". At the bottom of the form, there is a large blue button with the text "Cambiar Contraseña".

**Figura B.16** Formulario de cambio de contraseña.

4. Haz clic en “Cambiar contraseña”.
5. Si has introducido una contraseña válida, se habrá actualizado correctamente.

## Cambiar el tema

1. Para cambiar el tema de color de la aplicación, pulsa en el botón con un sol o una luna presente en la sección de Apariencia.

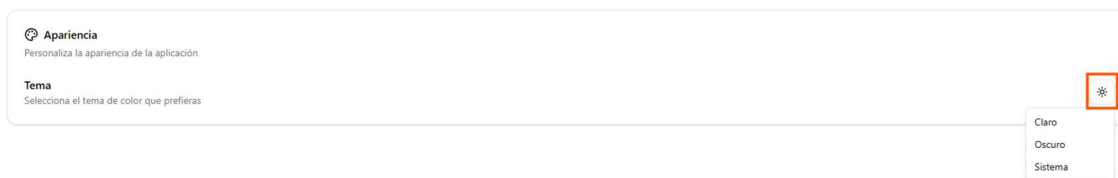


Figura B.17 Botón para cambiar el tema de la aplicación.

2. Selecciona “Claro” u “Oscuro” según tu preferencia.
3. Si quieres utilizar el mismo tema que tu dispositivo, elige “Sistema”.
4. El tema se actualizará de forma automática según tu elección.

## Información académica

### Ver las Universidades disponibles

1. Desde la barra de navegación lateral, haz clic en “Universidades”.
2. O accede directamente a /universidades.
3. Aparecerá la página de Universidades, con un listado de todas las disponibles en la plataforma.

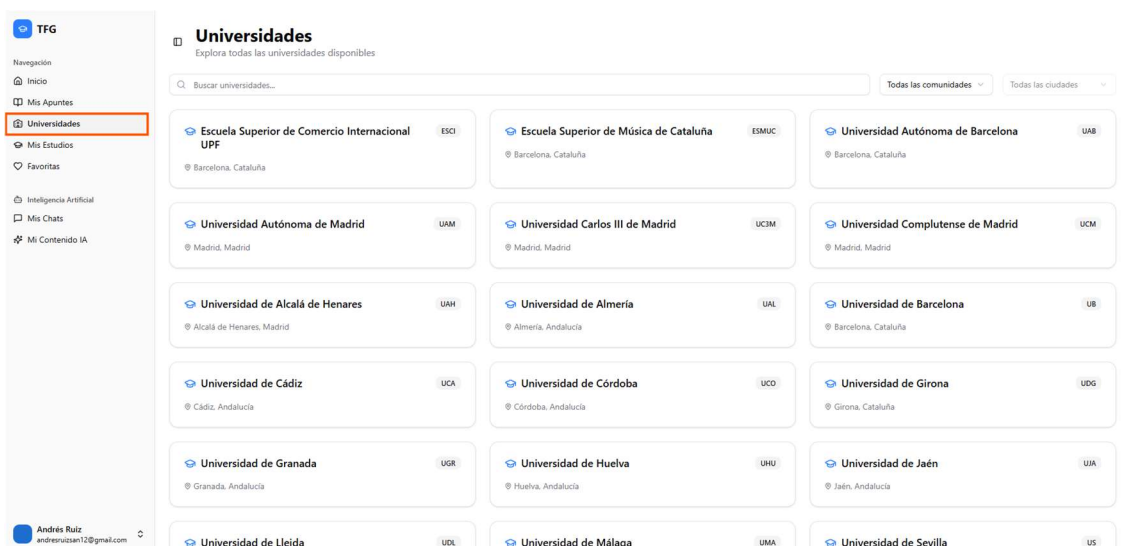


Figura B.18 Página de Universidades con su acceso directo en la barra lateral.

4. Para buscar una universidad, introduce su nombre en el campo de texto disponible en la parte superior.
5. Si quieres filtrar por comunidades autónomas o ciudades, utiliza los seleccionables que se encuentran en la parte superior.

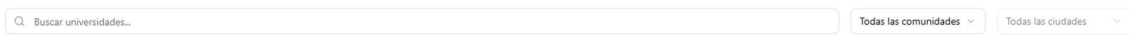


Figura B.19 Sección de búsqueda y filtros en la página de Universidades.

## Ver la información y estudios de una Universidad

1. En la lista de Universidades explicada anteriormente, haz clic en la tarjeta de la Universidad que quieras visualizar.

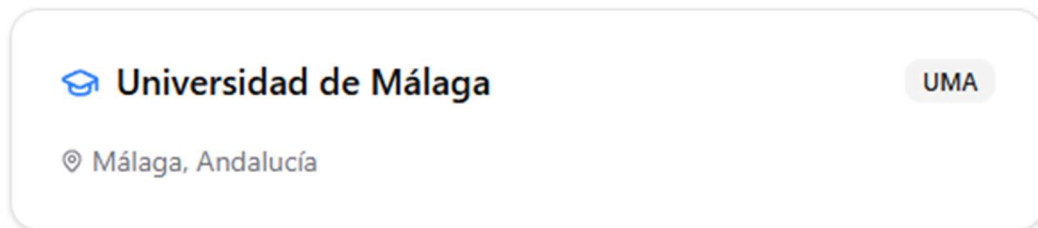


Figura B.20 Tarjeta de la Universidad de Málaga.

2. Se abrirá una página con la información de esa Universidad.
3. Más abajo, podrás encontrar los estudios disponibles en ella.
4. Para buscar un estudio, introduce su nombre en el campo de texto disponible.
5. Si quieres filtrar por tipo, haz uso del campo seleccionable que se encuentra disponible.

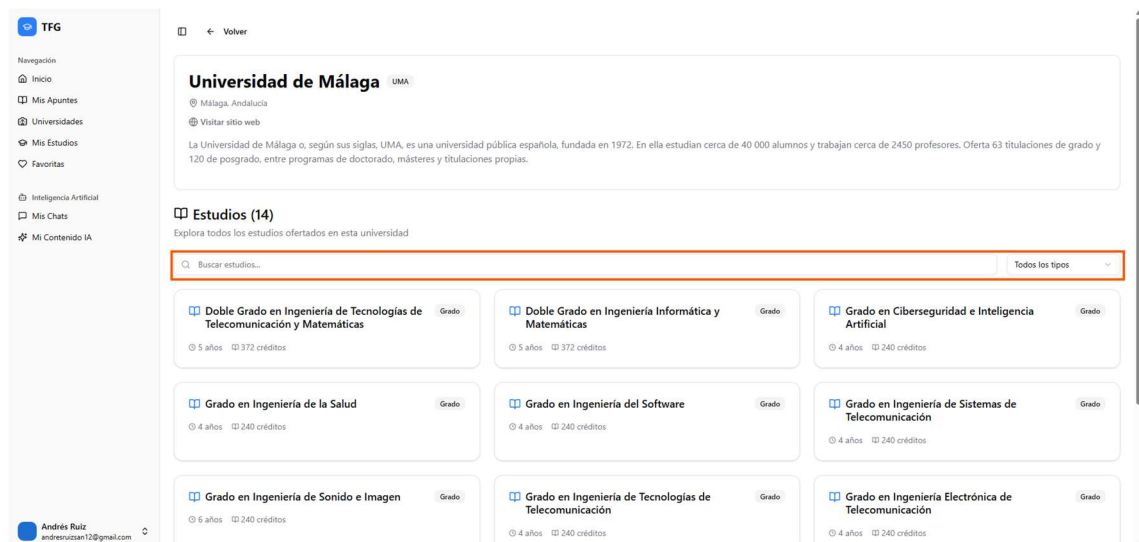


Figura B.21 Página de la Universidad de Málaga con la sección de búsqueda y filtros resaltada.

## Ver la información y asignaturas de unos estudios

1. En la lista de estudios explicada anteriormente, haz clic en la tarjeta de los estudios que quieras visualizar.

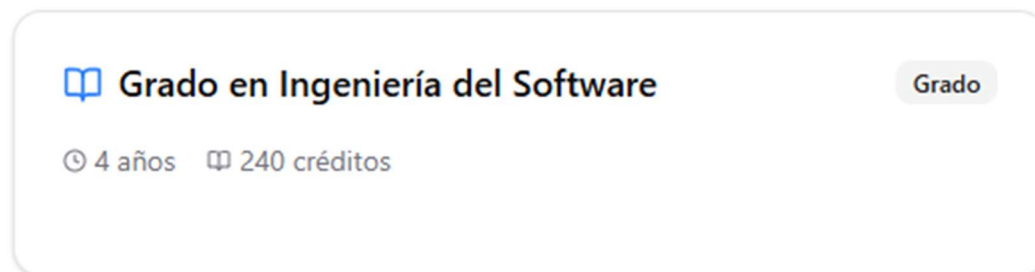


Figura B.22 Tarjeta del Grado en Ingeniería del Software en la página de la UMA.

2. Se abrirá una página con la información de esos estudios.
3. Más abajo, podrás encontrar las asignaturas disponibles, organizadas por curso.
4. Para elegir un curso, pulsa en su nombre y aparecerán todas las asignaturas del mismo.
5. Para buscar una asignatura, introduce su nombre en el campo de texto disponible.
6. Si quieres filtrar por semestre o tipo, utiliza los seleccionables que se encuentran en la parte superior.

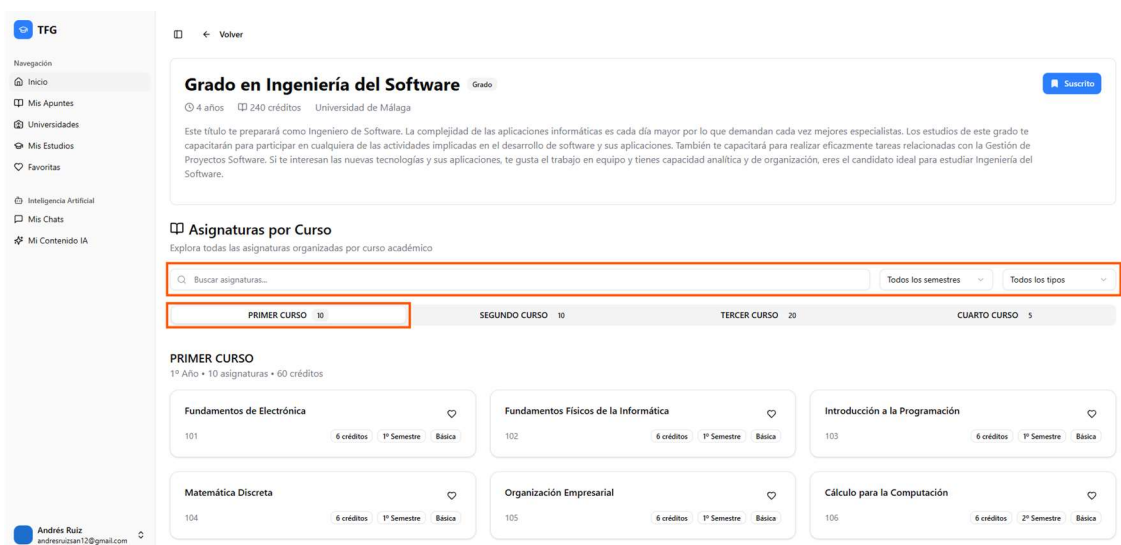
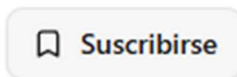


Figura B.23 Página del Grado en Ing. del Software con la sección de búsqueda, filtros y cursos.

## Suscribirte a unos estudios

1. Para suscribirte a unos estudios, haz clic en el botón “Suscribirse” situado en la esquina superior derecha de la página.

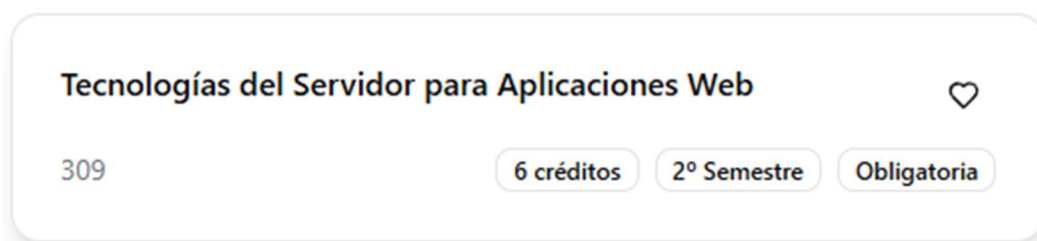


**Figura B.24** Botón “Suscribirse” en la página del estudio.

2. Verás un mensaje diciendo “Te has suscrito al estudio” y el botón pasará a decir “Suscrito”.
3. Una vez suscrito, podrás encontrar un acceso directo a los estudios en la página “Mis estudios”, así como en la página de “Inicio”.
4. Para cancelar la suscripción, vuelve a pulsar el botón.

## Ver la información y documentos de una asignatura

1. En la lista de asignaturas explicada anteriormente, haz clic en la tarjeta de la asignatura que quieras visualizar.



**Figura B.25** Tarjeta de la asignatura “Tecnologías del Servidor para Aplicaciones Web”.

2. Se abrirá una página con la información de esa asignatura.
3. Más abajo, podrás encontrar los documentos disponibles sobre esa asignatura.
4. Para buscar un documento, introduce su nombre en el campo de texto disponible.
5. Si quieres filtrar por tipo o establecer un orden, utiliza los seleccionables que se encuentran a tu disposición.



Figura B.26 Página de una asignatura con la sección de búsqueda y filtros resaltada.

## Marcar una asignatura como favorita

1. Para marcar una asignatura como favorita, haz clic en el corazón situado en la esquina superior derecha de la página.



Figura B.27 Botón de Marcar favorito.

2. Verás un mensaje diciendo “Añadida a favoritos” y el botón cambiará de color.

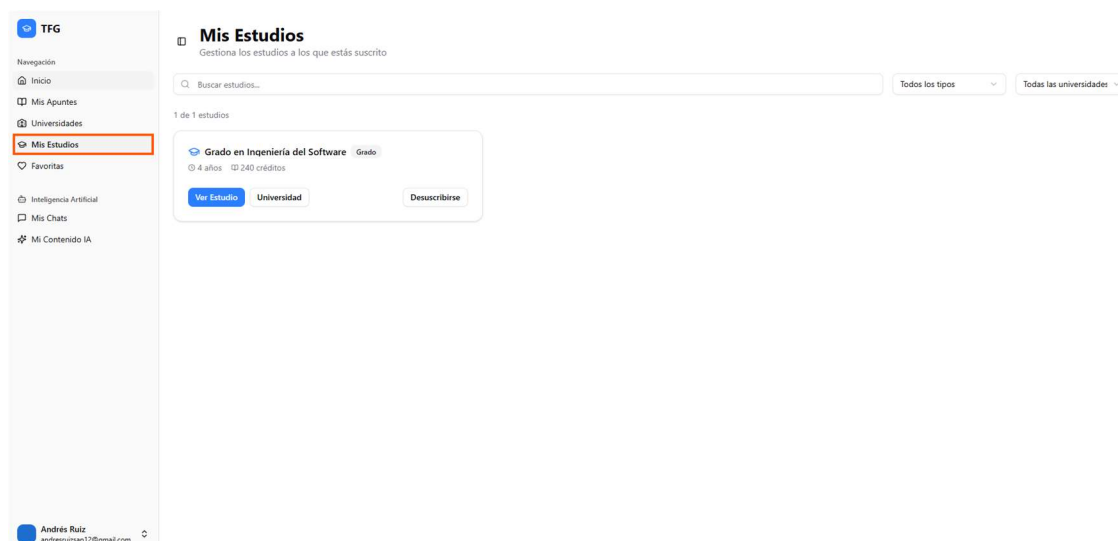


Figura B.28 Asignatura marcada como favorita.

3. Una vez marcada como favorita, podrás encontrar un acceso directo a la asignatura en la página “Favoritas”, así como en la página de inicio.
4. Para quitarla de favoritos, vuelve a pulsar el botón.
5. Adicionalmente, también puedes marcar como favorita una asignatura desde la página de unos estudios, haciendo clic en el corazón que aparece en la tarjeta de esa asignatura.

## Ver tus estudios

1. Desde la barra de navegación lateral, haz clic en “Mis estudios”.
2. O accede directamente a /estudios.
3. Se abrirá una página en la que podrás ver todos tus estudios.



**Figura B.29** Página “Mis Estudios” con su acceso directo en la barra lateral.

4. Para acceder a un estudio, pulsa en “Ver estudio”.
5. Si quieres cancelar la suscripción, haz clic en “Desuscribirse”.
6. Puedes buscar por texto o filtrar haciendo uso de los campos disponibles en la parte superior.

## Ver tus asignaturas favoritas

1. Desde la barra de navegación lateral, haz clic en “Favoritas”.
2. O accede directamente a /favoritas.
3. Se abrirá una página en la que podrás ver todos tus asignaturas favoritas.
4. Para acceder a una asignatura, pulsa sobre su tarjeta.
5. Si quieres quitarla de favoritos, haz clic en el corazón que aparece en la esquina superior derecha de la tarjeta.
6. Puedes buscar por texto o filtrar haciendo uso de los campos disponibles en la parte superior.

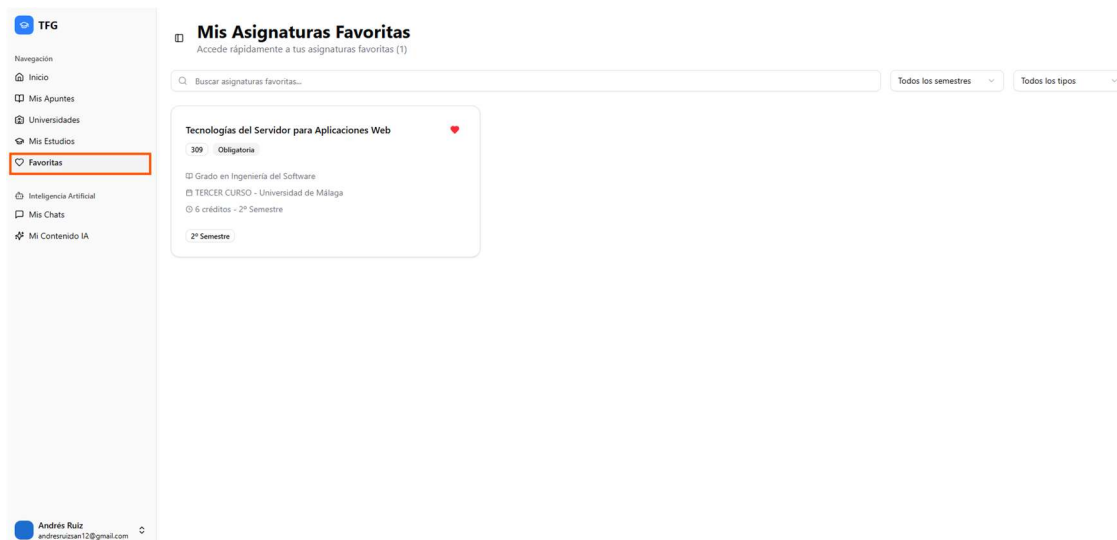


Figura B.30 Página “Mis Asignaturas Favoritas” con su acceso directo en la barra lateral.

### Crear una Universidad (sólo administradores)

1. Desde la barra de navegación lateral, haz clic en “Universidades”.
2. O accede directamente a /universidades.
3. Haz clic en “Nueva Universidad”.
4. Rellena el formulario con los datos de la Universidad que quieres añadir.
5. Pulsa “Crear Universidad”.
6. La Universidad se creará y se mostrará en la lista junto al resto de universidades.

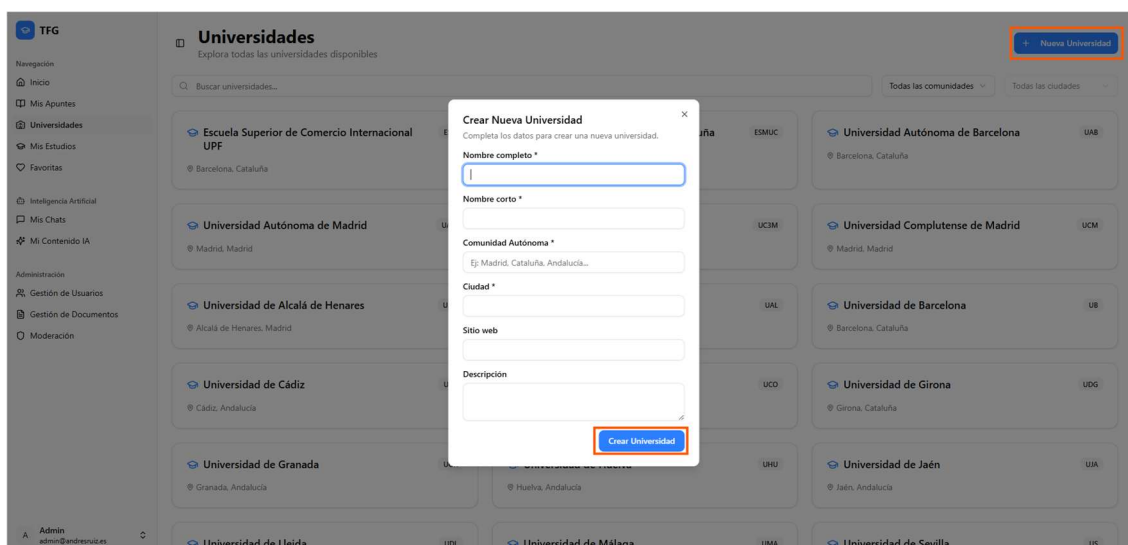


Figura B.31 Formulario de creación de nueva Universidad con los botones resaltados.

## Crear nuevos estudios en una Universidad (sólo administradores)

1. En la lista de Universidades explicada anteriormente, haz clic en la tarjeta de la Universidad que imparte los estudios.
2. Se abrirá una página con la información de esa Universidad.
3. Haz clic en “Nuevo Estudio”.
4. Rellena el formulario con los datos del nuevo estudio.
5. Pulsa en “Crear Estudio”.
6. El estudio se creará y se mostrará en la lista junto al resto de estudios.

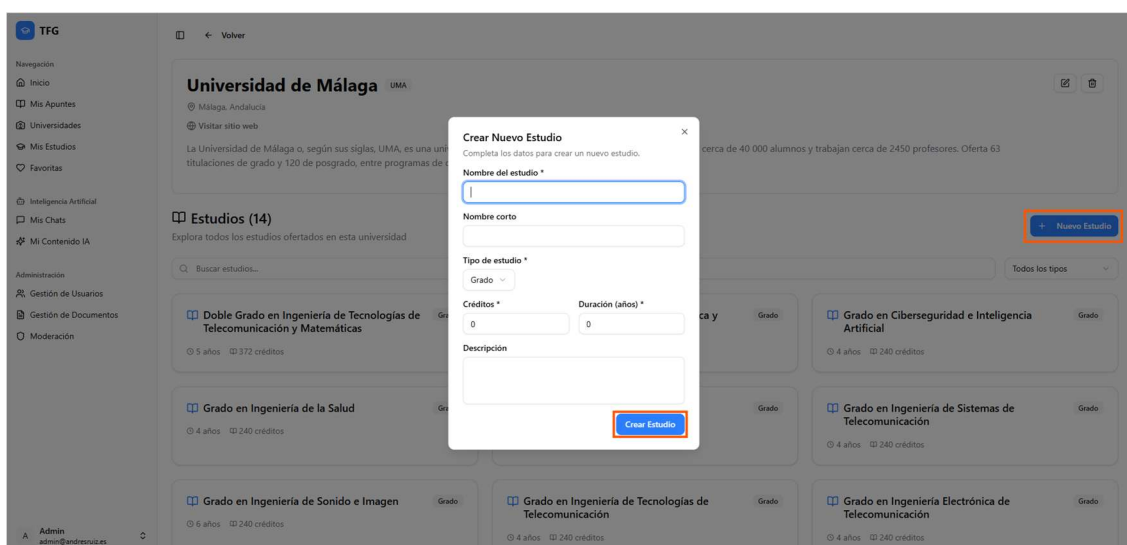


Figura B.32 Formulario de creación de nuevo estudio con los botones resaltados.

## Añadir curso a unos estudios (sólo administradores)

1. En la lista de estudios explicada anteriormente, haz clic en la tarjeta de los estudios que quieras visualizar.
2. Se abrirá una página con la información de esos estudios.
3. Haz clic en “Nuevo Curso”.
4. Rellena el nombre del curso y selecciona el año académico en el que se imparte.
5. Pulsa “Crear Curso”.
6. El curso se creará y se mostrará en la página.

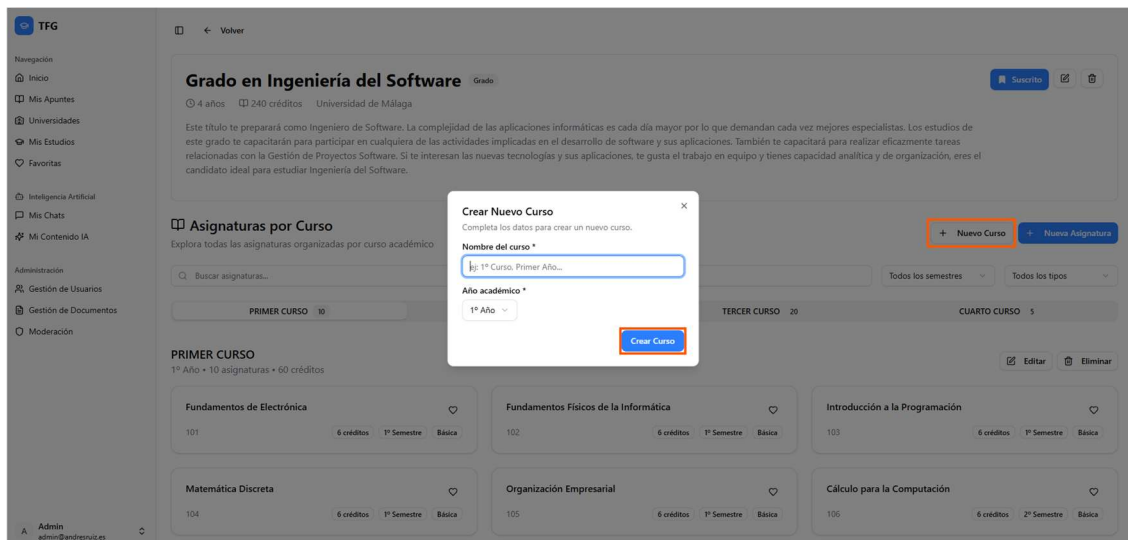


Figura B.33 Formulario de creación de nuevo curso con los botones resaltados.

### Crear nueva asignatura en unos estudios (sólo administradores)

1. En la lista de estudios explicada anteriormente, haz clic en la tarjeta de los estudios que quieras visualizar.
2. Se abrirá una página con la información de esos estudios.
3. Haz clic en “Nueva Asignatura”.
4. Selecciona el curso en el que se imparte y rellena los datos de la asignatura.
5. Pulsa “Crear Asignatura”.
6. La asignatura se creará y se mostrará en la pestaña correspondiente a su curso.

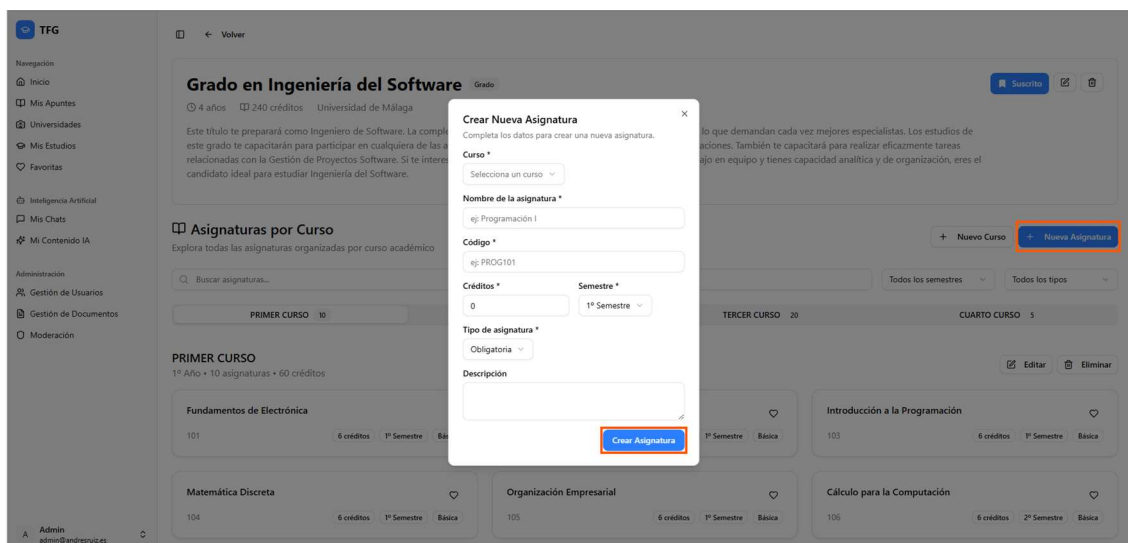


Figura B.34 Formulario de creación de nueva asignatura con los botones resaltados.

# Gestión de Documentos

## Ver un documento

1. Accede a la página del documento pulsando en la tarjeta del mismo desde la página de una asignatura o cualquier otra página en la que se muestren.

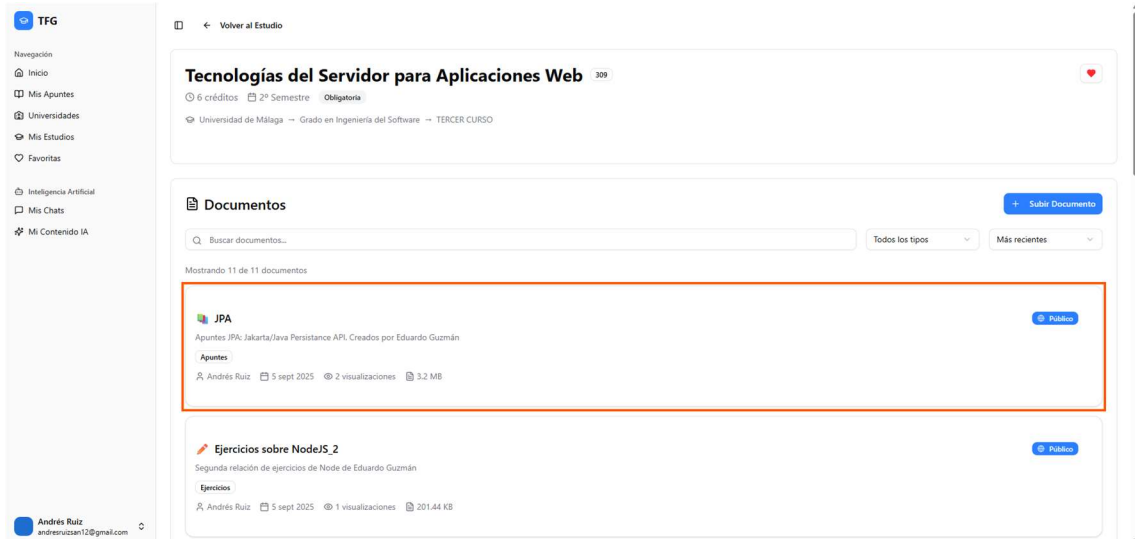


Figura B.35 Página de una asignatura con la tarjeta de un documento seleccionada.

2. Si se trata de un documento PDF, podrás visualizarlo desde la propia aplicación, pudiendo también descargarlo pulsando en el botón del visor mostrado para ello.
3. Si por el contrario se trata de un documento en otro formato, podrás descargarlo pulsando el botón “Descargar archivo”.
4. A la derecha, podrás encontrar la información del documento, así como una sección de comentarios.
5. Si pulsas en la pestaña IA, podrás ver los chats y contenidos de Inteligencia Artificial.



Figura B.36 Página del documento, con el visor de PDF y la sección de información.

### Sección de comentarios

1. En la página de un documento, dentro de la pestaña de Info, podrás encontrar una sección de comentarios en la parte inferior.
2. Sitúate en el campo de texto y escribe el comentario que desees acerca del documento.
3. Haz clic en “Comentar” y el comentario se habrá creado.
4. Para responder a otro comentario, pulsa en “Responder”, situado debajo del comentario al que desees contestar.
5. Escribe el mensaje y pulsa “Responder”, de forma análoga a lo anterior.
6. Para dar “me gusta” a un comentario, pulsa en el corazón situado debajo del mismo.
7. Si quieres editar o eliminar tu comentario, pulsa en los tres puntos (...) situados también debajo del comentario.

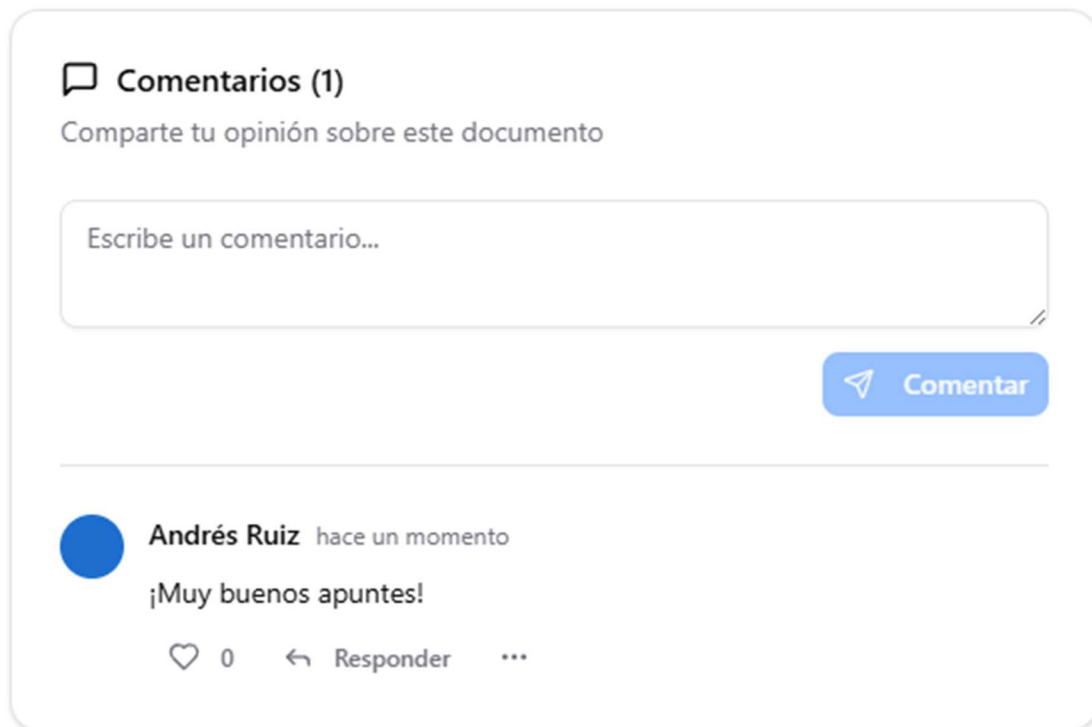


Figura B.37 Sección de comentarios en la página de un documento.

## Reportar un documento

1. En la página de un documento, dentro de la pestaña de Info, encontrarás tres puntos (...) a la derecha del nombre del documento.
2. Pulsa en ellos y haz clic en “Reportar”.

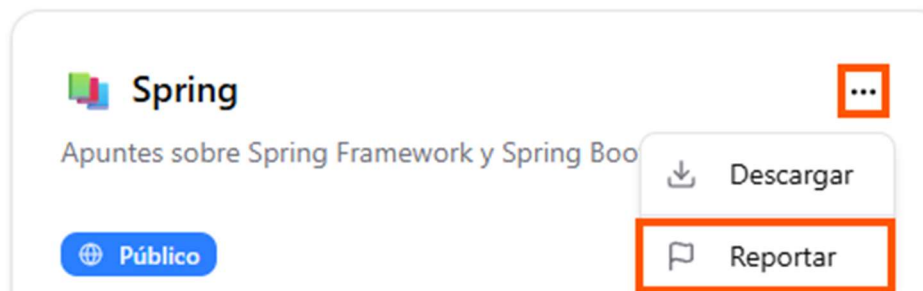



Figura B.38 Botón de “Reportar” en la página de un documento.

3. Se abrirá un diálogo en el que debes elegir la razón del reporte y añadir de forma opcional una descripción del motivo del mismo.
4. Pulsa en el botón “Enviar reporte”.
5. El reporte será recibido por los administradores, que revisarán el problema y tomarán las acciones oportunas.



Reportar documento

Estás reportando: **Spring**  
Tu reporte ayudará a mantener la comunidad segura.

Razón del reporte \*

Selecciona una razón ▾

Descripción adicional (opcional)

Proporciona más detalles sobre el problema...

0/1000 caracteres

Cancelar Enviar reporte

Figura B.39 Formulario de reporte de un documento.

### Subir un documento

1. Accede a la página de la asignatura en la que quieras subir los apuntes.
2. Haz clic en el botón “Subir Documento” situado en la sección de documentos.
3. Se abrirá un diálogo en el que debes seleccionar el archivo de tu dispositivo.
4. Posteriormente, debes rellenar el título, una descripción opcional, elegir el tipo de documento y seleccionar la privacidad.
5. Pulsa en “Subir documento”.
6. El documento se subirá y mostrará una confirmación, apareciendo ya en la lista junto al resto de apuntes.

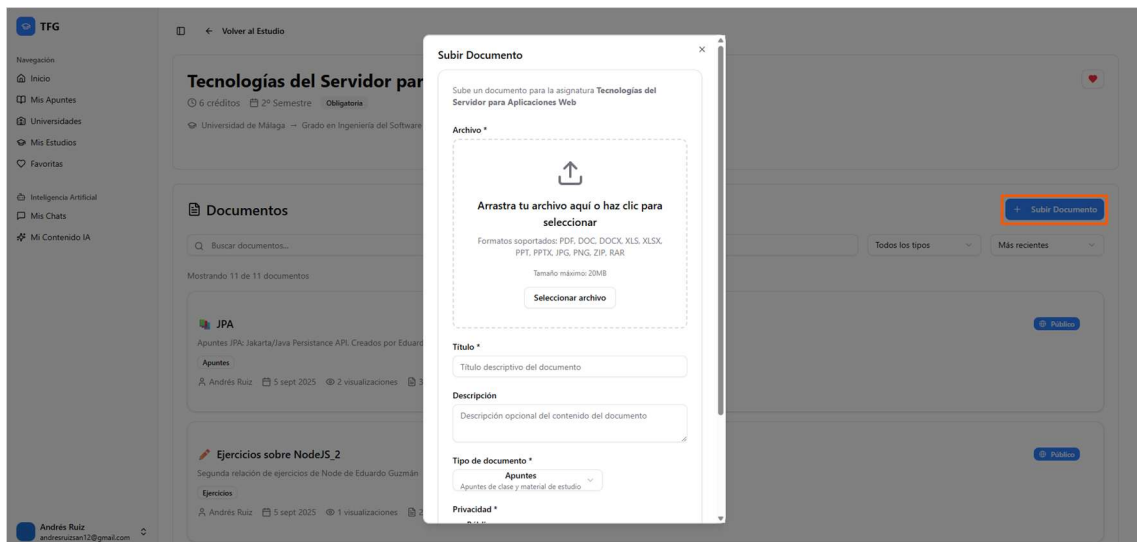


Figura B.40 Formulario de subida de un documento con su botón resaltado.

## Editar un documento

1. Accede a la página del documento pulsando en la tarjeta del mismo.
2. Haz clic en los tres puntos (...) situados a la derecha del título del mismo, en la sección de Info.
3. Pulsa en “Editar”.

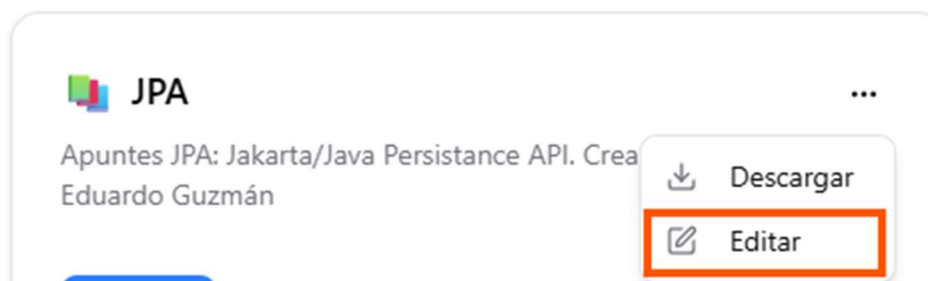


Figura B.41 Botón de “Editar” en la página de un documento.

4. Se abrirá un formulario en el que puedes editar el título, la descripción, el tipo de documento y la privacidad.
5. Cuando hayas editado los campos que necesites cambiar, pulsa en “Actualizar Documento”.
6. La información se actualizará y mostrará un mensaje de confirmación.

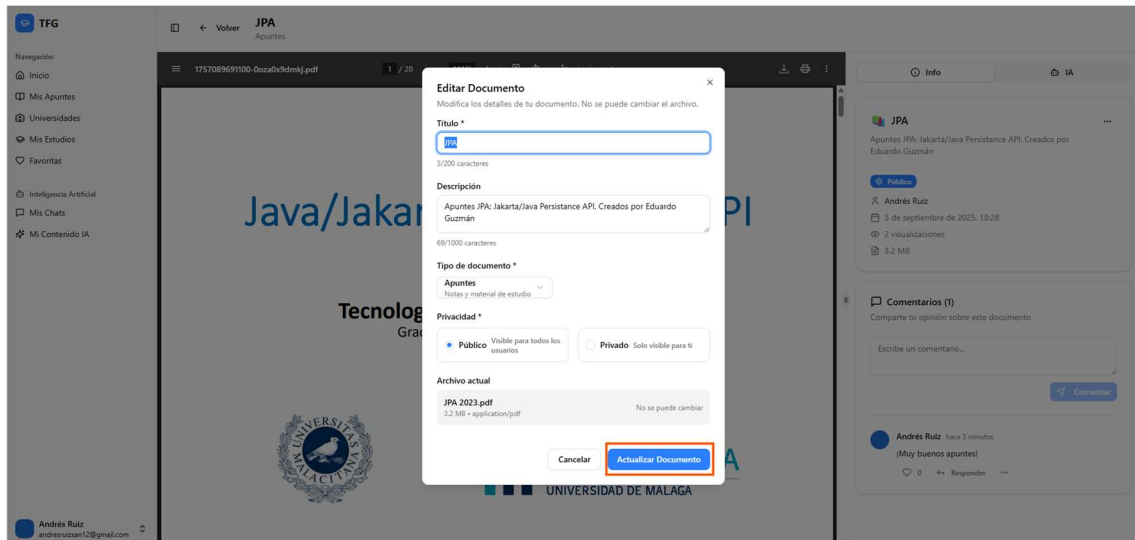


Figura B.42 Formulario de edición de un documento.

## Ver tus apuntes

1. Desde la barra de navegación lateral, haz clic en “Mis apuntes”.
2. O accede directamente a /mis-apuntes.
3. Se abrirá una página en la que podrás ver todos los apuntes que has subido a la plataforma.
4. Puedes buscar por nombre o filtrar haciendo uso de los campos disponibles en la parte superior.

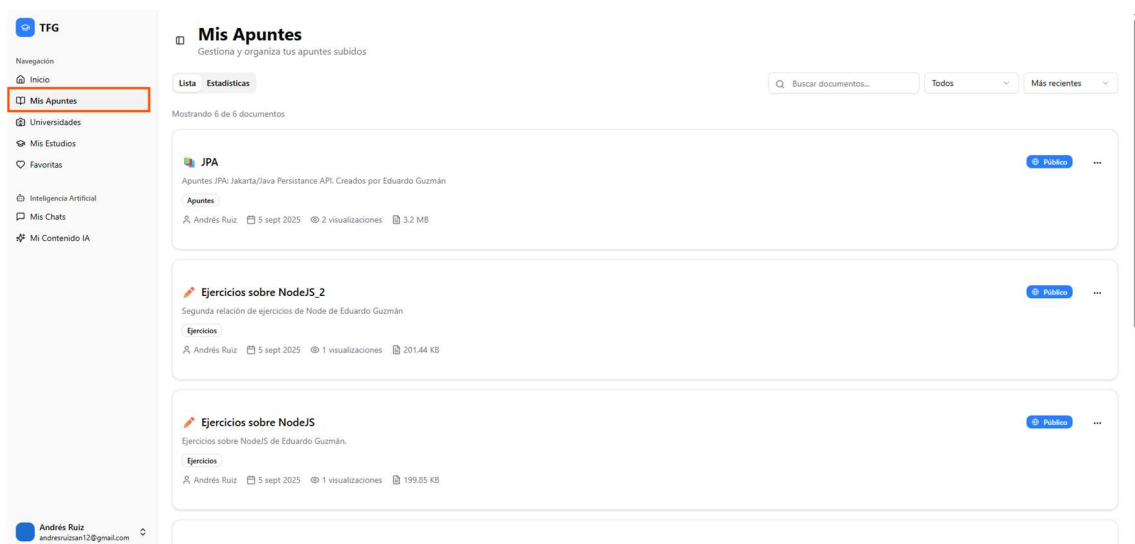


Figura B.43 Página “Mis Apuntes” con su acceso en la barra lateral resaltado.

5. Además, puedes acceder a la pestaña de estadísticas, donde se muestra el rendimiento de tus documentos en la plataforma.

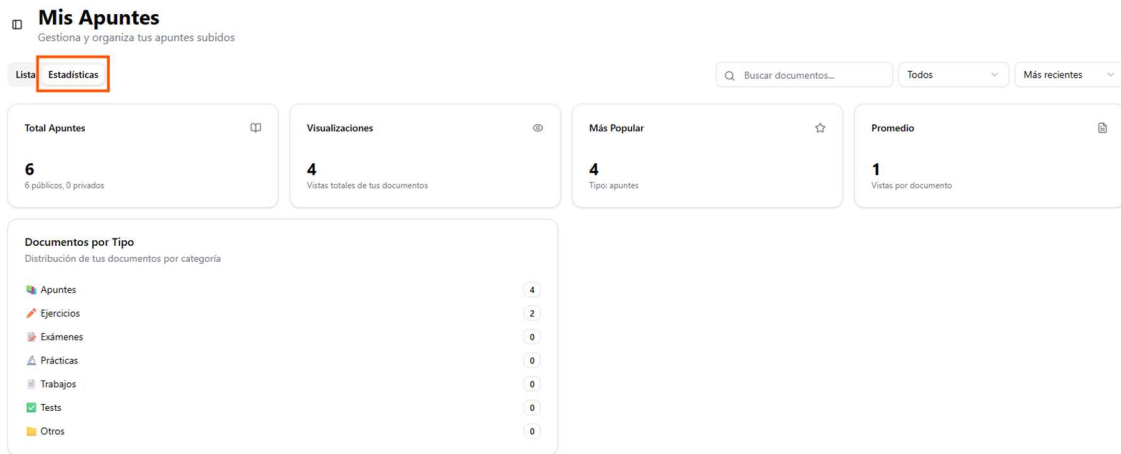


Figura B.44 Sección de Estadísticas en la página “Mis Apuntes”.

## Eliminar un documento

1. En la página de “Mis Apuntes”, busca el documento que desees eliminar.
2. Pulsa sobre los tres puntos (...) que aparecen a la derecha de la tarjeta.
3. Selecciona “Eliminar”.

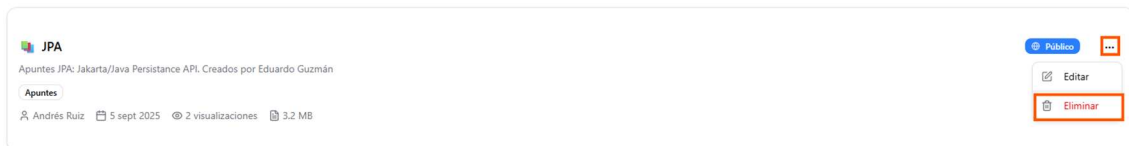


Figura B.45 Botón “Eliminar” en la tarjeta de un documento de la página “Mis Apuntes”.

4. Se mostrará un diálogo para confirmar la acción.
5. Pulsa de nuevo en “Eliminar”.

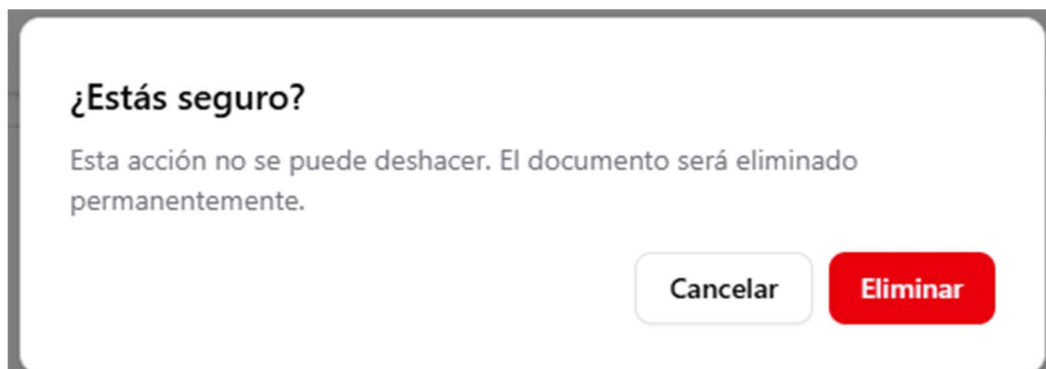


Figura B.46 Confirmación de eliminación.

6. El documento se eliminará de la plataforma y se mostrará un mensaje de confirmación.

# Funcionalidades de Inteligencia Artificial

## Chat con IA sobre Documentos

1. Abre cualquier documento.
2. Ve a la pestaña de “Chats” en la sección de IA.
3. Haz clic en “Nuevo chat”

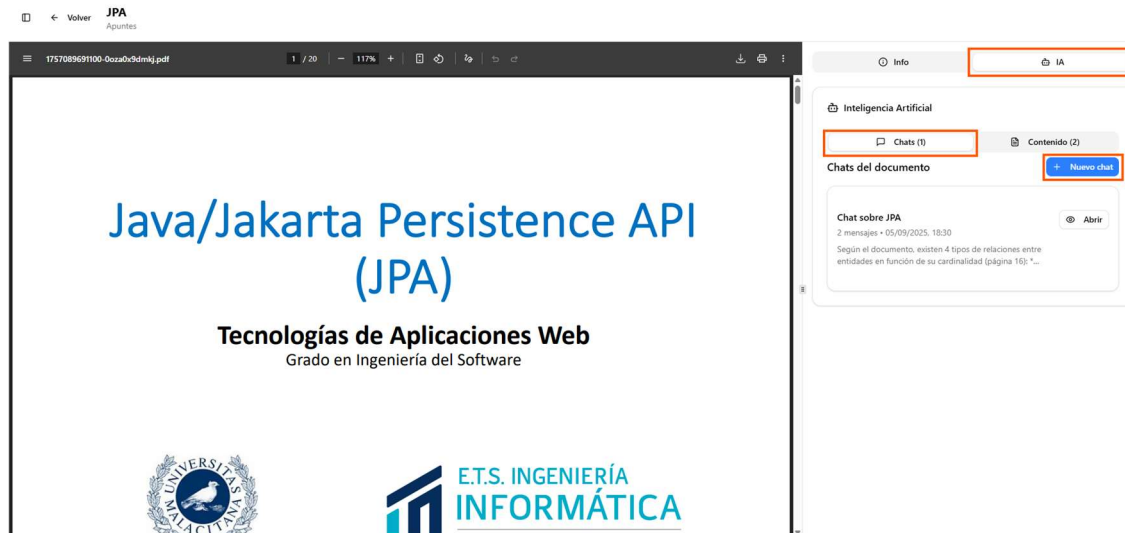


Figura B.47 Botón “Nuevo chat” en la sección de IA de la página de un documento.

4. Comienza a hacer preguntas sobre el contenido escribiendo en el campo de texto y pulsando en el botón de enviar situado a la derecha.
5. Para cambiar el título del chat (para poder distinguirlo de forma rápida), puedes pulsar en el icono del lápiz situado a la derecha del mismo.
6. Si quieres eliminar la conversación, pulsa en el icono de la papelera de la esquina superior derecha.

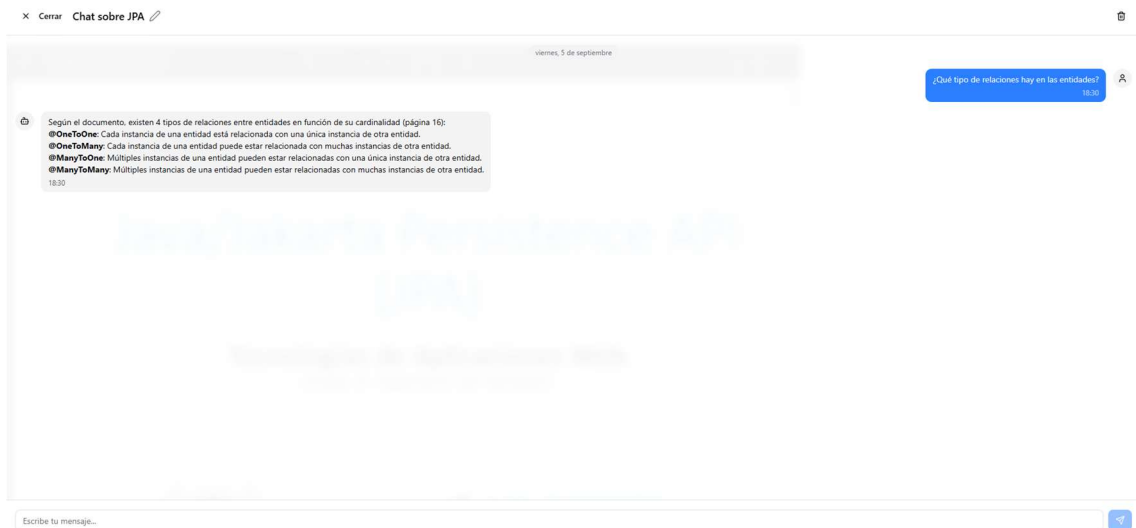


Figura B.48 Chat con la IA sobre un documento.

## Ver tus chats

1. Desde la barra de navegación lateral, haz clic en “Mis Chats”, en la sección de “Inteligencia Artificial”.
2. O accede directamente a /mis-chats.
3. Se abrirá una página en la que podrás ver todas las conversaciones que has creado sobre los distintos documentos.
4. Puedes buscar por nombre haciendo uso del campo de texto disponible en la parte superior.

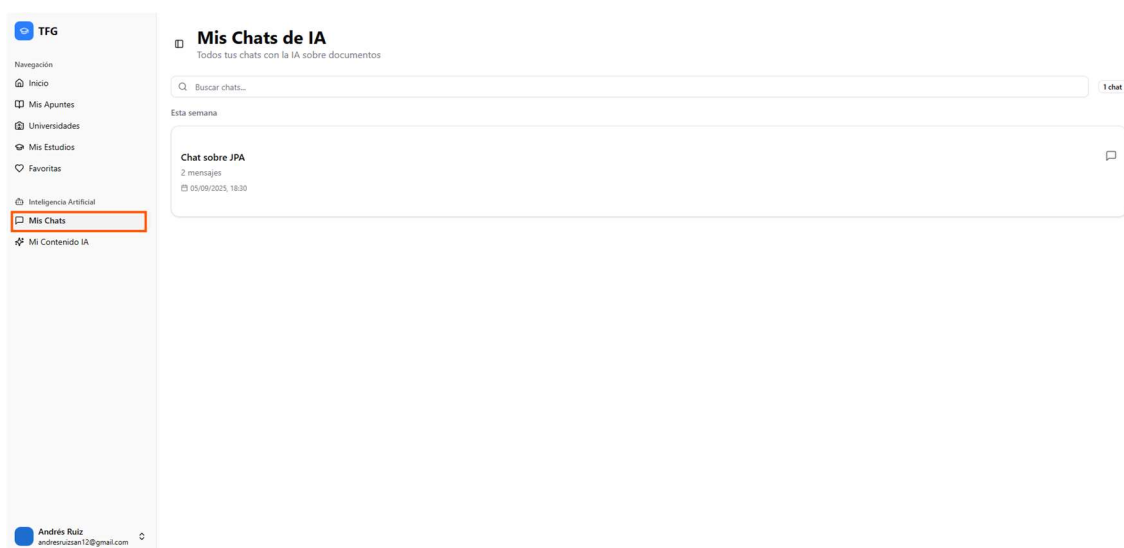


Figura B.49 Página “Mis Chats de IA” con su acceso en la barra lateral resaltado.

5. Pulsa en cualquiera de los chats para abrirlo.

6. Puedes conversar con la IA igual que anteriormente, así como acceder de forma directa a la página del documento haciendo clic en “Ver documento”.

## Generación de contenido con IA

1. Abre cualquier documento.
2. Ve a la pestaña de “Contenido” en la sección de IA.
3. Haz clic en “Generar contenido”.
4. Selecciona el tipo de contenido que quieres generar:
  - a. Resumen.
  - b. Test interactivo.
  - c. Examen.
  - d. Diagrama.
  - e. Guía de estudio.
  - f. Audio de clase.
  - g. Extraer texto.

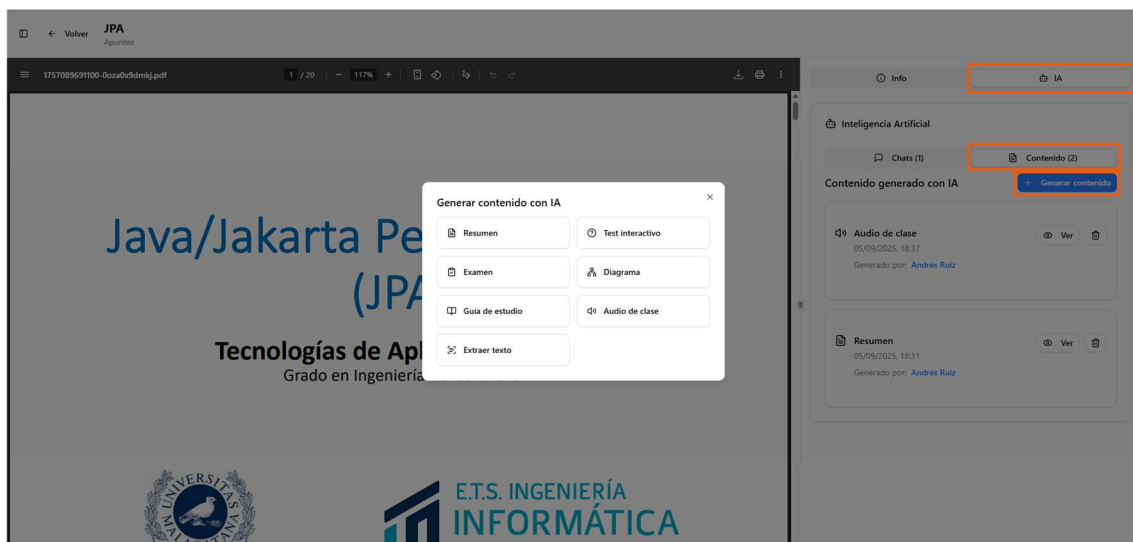


Figura B.50 Generar contenido con IA con su botón resaltado.

5. Espera a que se genere el contenido, puede tardar unos minutos.
6. Una vez generado, se mostrará un mensaje de éxito y el contenido aparecerá en la lista junto al resto de contenido generados para ese documento.

7. Pulsa en “Ver” para visualizarlo.
8. Se mostrará el contenido generado de forma que facilite el estudio.



Figura B.51 Visor de contenidos IA.

## Ver tus contenido de IA

1. Desde la barra de navegación lateral, haz clic en “Mi Contenido IA”, en la sección de “Inteligencia Artificial”.
2. O accede directamente a `/mi-contenido-ia`.
3. Se abrirá una página en la que podrás ver todos los contenidos de IA que has generado sobre los distintos documentos.
4. Puedes buscar por nombre haciendo uso del campo de texto disponible en la parte superior o ver por tipo de contenido haciendo uso de las pestañas.
5. Pulsa en cualquiera de los contenidos para abrirlo.
6. Puedes visualizar el contenido igual que anteriormente, así como acceder de forma directa a la página del documento haciendo clic en “Ver documento”.



Figura B.52 Página “Mi Contenido de IA” junto a su acceso en la barra lateral resaltado.

## Paneles de administración (sólo administradores)

### Gestión de usuarios

1. Si eres administrador, desde la barra de navegación lateral, haz clic en “Gestión de Usuarios”, en la sección de “Administración”.
2. O accede directamente a `/admin/usuarios`.
3. Aparecerá la página de Usuarios, con un listado de todas las personas registradas en la plataforma.
4. Al comienzo de la página, puedes ver una sección con estadísticas del número de usuarios.
5. Puedes buscar por texto o filtrar haciendo uso de los campos disponibles en la parte superior.
6. Para cambiar el rol de un usuario, haz clic en el desplegable y selecciona el nuevo rol.
7. Pulsando en los tres puntos (...) situados a la derecha de cada usuario, puedes realizar las siguientes acciones:
  - a. Cambiar contraseña:
    - i. Introduce la nueva contraseña.
    - ii. Pulsa en “Cambiar contraseña”.

- b. Ver sesiones:
  - i. Se abrirá una pestaña con todas las sesiones del usuario.
  - ii. Puedes revocarlas individualmente pulsando en “Revocar” o cerrarlas todas pulsando en “Revocar Todas”.
- c. Suplantar:
  - i. Accederás a la plataforma con una sesión del usuario al que estás “suplantando”.
  - ii. Verás la aplicación tal como la ve él y podrás realizar acciones en su nombre.
- d. Banear:
  - i. Explica la razón del ban.
  - ii. Selecciona la duración.
  - iii. Pulsa en “Banear usuario” y ya no podrá acceder a la plataforma.
- e. Eliminar:
  - i. Se elimina al usuario definitivamente de la plataforma.

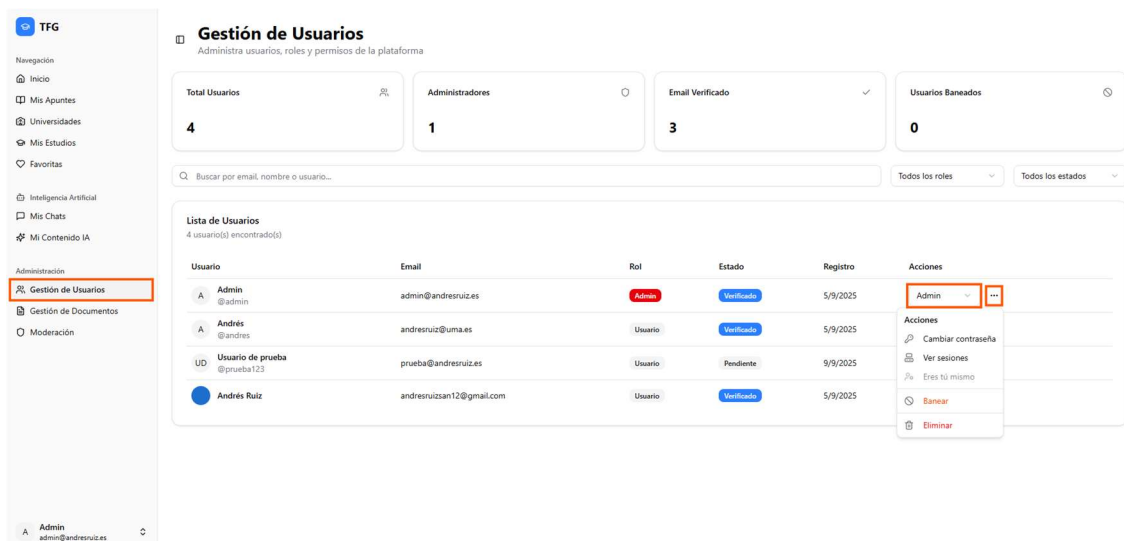


Figura B.53 Página “Gestión de Usuarios”.

## Gestión de documentos

1. Si eres administrador, desde la barra de navegación lateral, haz clic en “Gestión de Documentos”, en la sección de “Administración”.
2. O accede directamente a `/admin/documentos`.
3. Aparecerá la página de Documentos, con un listado de todos los documentos subidos en la plataforma.
4. Al comienzo de la página, así como en la pestaña “Estadísticas”, puedes ver una sección con estadísticas del número de documentos.
5. Puedes buscar por texto o filtrar haciendo uso de los campos disponibles en la parte superior.
6. Puedes acceder a un documento haciendo clic en el mismo.
7. Pulsando en los tres puntos (...) situados a la derecha de cada usuario, puedes realizar las siguientes acciones:
  - a. Editar:
    - i. Se abrirá un formulario en el que puedes editar toda la información del documento.
    - ii. Pulsa en “Actualizar documento” para guardar la nueva información.
  - b. Eliminar:
    - i. Se eliminará el documento de forma definitiva de la plataforma.
    - ii. Pulsa en “Eliminar” para confirmar el borrado.

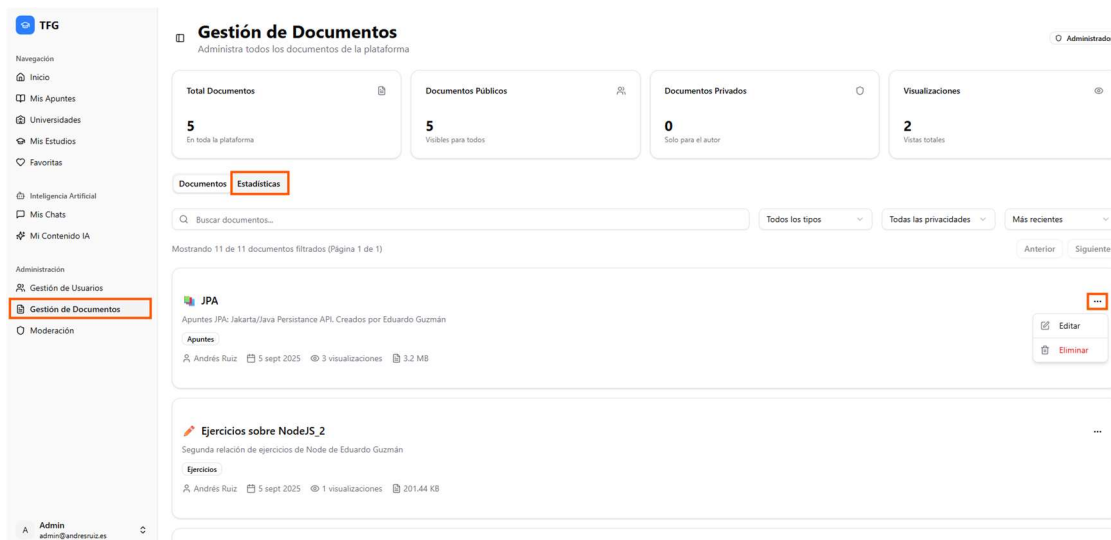


Figura B.54 Página “Gestión de Documentos”.

## Moderación de reportes

1. Si eres administrador, desde la barra de navegación lateral, haz clic en “Moderación”, en la sección de “Administración”.
2. O accede directamente a `/admin/moderacion`.
3. Aparecerá el Centro de Moderación, con un listado de todos los reportes creados en la plataforma.
4. Al comienzo de la página puedes ver una sección con estadísticas del número de reportes por cada estado.
5. Puedes filtrar haciendo uso de los campos disponibles en la parte superior.

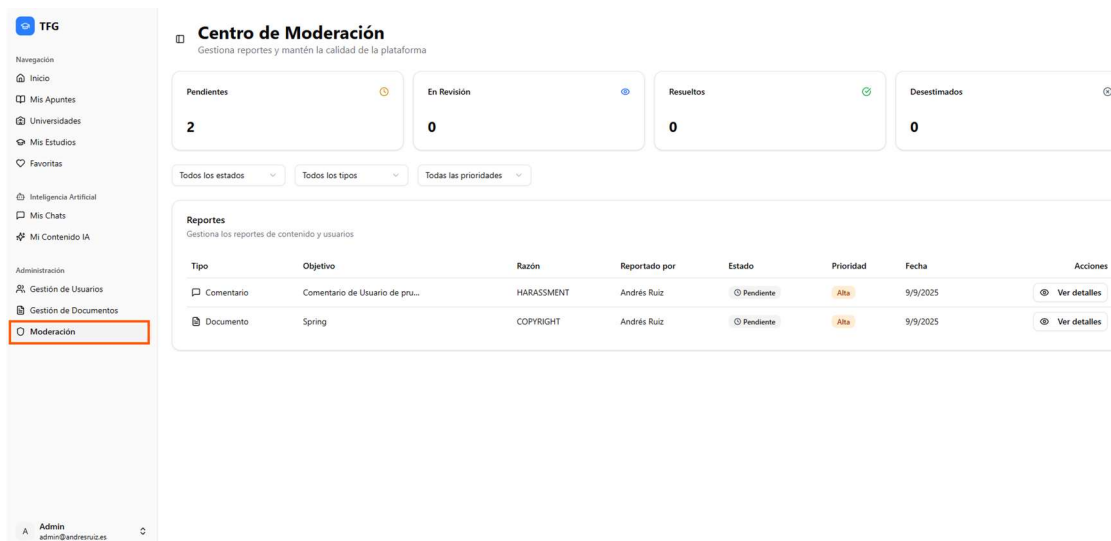
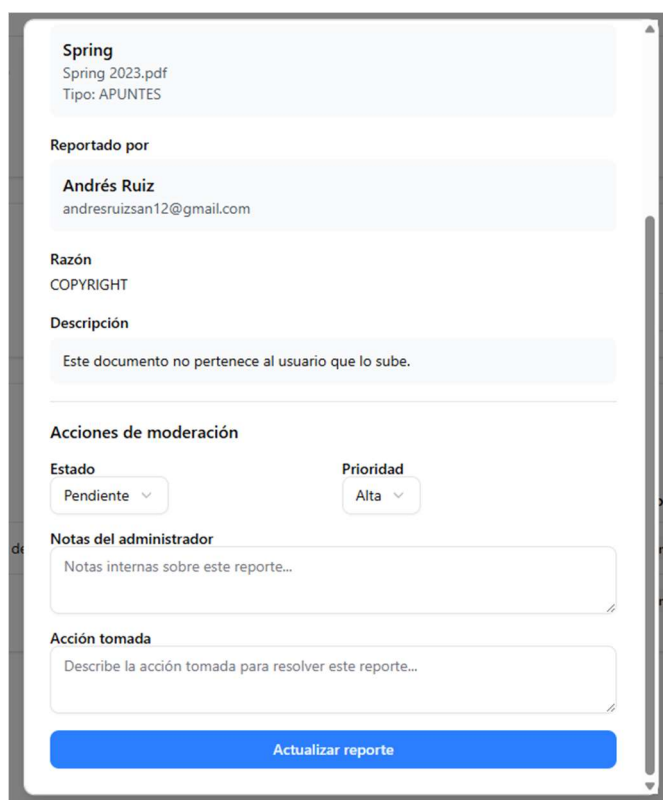


Figura B.55 Página “Centro de Moderación”.

6. Puedes ver los detalles de un reporte pulsando el botón “Ver detalles” y actualizar los siguientes campos:
  - a. Estado: pendiente, en revisión, resuelto o desestimado.
  - b. Prioridad: baja, media, alta o urgente.
  - c. Notas del administrador.
  - d. Acción tomada.
7. Pulsa en “Actualizar reporte” para incluir la nueva información y actualizar los campos.



The image shows a web form for updating a report. At the top, the report title is "Spring" with a subtitle "Spring 2023.pdf" and type "APUNTES". Below this, the reporter is identified as "Andrés Ruiz" with the email "andresruizsan12@gmail.com". The reason for the report is "COPYRIGHT". The description field contains the text "Este documento no pertenece al usuario que lo sube.". Under the "Acciones de moderación" section, there are two dropdown menus: "Estado" set to "Pendiente" and "Prioridad" set to "Alta". Below these are two text input fields: "Notas del administrador" with the placeholder "Notas internas sobre este reporte..." and "Acción tomada" with the placeholder "Describe la acción tomada para resolver este reporte...". At the bottom of the form is a prominent blue button labeled "Actualizar reporte".

Figura B.56 Formulario de actualización de reporte.



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA