



Grado en Ingeniería Informática

Trabajo fin de título



StudyUniv

Asistente digital inteligente para el aprendizaje de asignaturas
basado en modelos de lenguaje avanzados

Autor: Álvaro Cabrera Winter
Tutor: José Miguel Santos Espino

Junio 2025

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D/D^a Álvaro Cabrera Winter, autor/a del Trabajo de Fin de Título Asistente Digital Inteligente para el Aprendizaje de Asignaturas Basado en Modelos de Lenguaje Avanzados, correspondiente a la titulación Grado en Ingeniería Informática, plan 41, en colaboración con la empresa/proyecto (indicar en su caso) _____

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

☒ se autoriza / ☐ no se autoriza, la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

☐ Se ha iniciado o hay intención de iniciarlo (defensa no pública).

☒ No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/la estudiante

Fdo.: _____

A rellenar y firmar **obligatoriamente** por el/la/los/las tutor/a/es/as.

En relación a la presente solicitud, se informa: (firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)

Fdo.: _____

Negativamente (la justificación en caso de informe negativo deberá incluirse en el TFT05)

Fdo.: _____

DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Resumen

La evolución de la inteligencia artificial (IA) ha transformado de manera significativa el ámbito del aprendizaje, proporcionando herramientas y recursos que potencian la adquisición de conocimientos y habilidades, por ello mi aplicación web trata de favorecer el estudio de forma óptima permitiendo para los usuarios registrados crear secciones de trabajo para las materias. Una vez creada y configurada cada asignatura se ofrece resúmenes de temas o tópicos, un chat interactivo para resolver dudas, y un sistema de evaluación, que permite la creación y corrección automatizada de exámenes de preguntas abiertas u opción múltiple. Su propósito es facilitar acceso a información precisa, mejorar la comprensión de contenidos y optimizar el tiempo de estudio, convirtiéndose en una herramienta de apoyo para estudiantes y docentes.

El desarrollo del trabajo se ha realizado con una metodología iterativa incremental, mostrando en cada entrega al tutor funcionalidades nuevas, pero en cada incremento se incluyen mejoras. Se realiza las siguientes fases que componen las tareas: Modelado / Requisitos tratando de analizar y validar los requisitos de la aplicación, diseño / implementación en el que se traducen los requisitos en un plan o diseño de cómo el software funcionará y se estructurará además del código implementado del trabajo, test / despliegue: el software se prueba exhaustivamente para encontrar y corregir errores, documentación / presentación: se prepara la documentación técnica y de usuario, que incluye guías, manuales de usuario.

Abstract

The evolution of artificial intelligence (AI) has significantly transformed the field of learning, providing tools and resources that enhance the acquisition of knowledge and skills, which is why my web application tries to promote study in an optimal way by allowing registered users to create work sections for courses. Once each course is created and configured, summaries of topics or themes are offered, an interactive chat to resolve doubts, and an evaluation system that allows the creation and automated correction of questions for open questions or multiple choice. The purpose is to facilitate access to accurate information, improve understanding content and optimize study time, becoming a support tool for students and professors.

The development of the project has been done with an incremental iterative methodology, showing new features in each delivery to the tutor, but improvements are included in each increment. The following phases that make up the tasks are carried out: Modeling / Requirements trying to analyze and validate the requirements of the application, design / implementation in which the requirements are translated into a plan or design of how the software will work and be structured in addition with implemented code, test / implementation: the software is tested exhaustively to find and correct errors, documentation / presentation: technical and user documentation is prepared, which includes guides, user manuals.

Agradecimientos

Durante la etapa universitaria he tenido voluntad y constancia para seguir y superar todas las situaciones y circunstancias que me he encontrado durante el camino, por ello a continuación quiero agradecer y mencionar a todos los que me han apoyado.

En primer lugar, quiero agradecer a mi tutor José Miguel Santos Espino por el apoyo de guiar y supervisar la evolución de mi Trabajo Fin de Grado.

A mi familia y amigos que me han apoyado dándome educación y buenos valores. También a mis compañeros con los que hemos compartido tiempo de clases y momentos de estudio.

A tod@s los docentes que me han preparado y evaluado para que tenga todos los conocimientos del grado y al personal de administración por el trato mostrado.

Muchas gracias a tod@s

Índice de contenidos

Capítulo 1.	Introducción.....	10
1.1	Antecedentes.....	10
1.2	Objetivos del producto	11
1.3	Objetivos de aprendizaje	11
1.4	Análisis de negocio Lean Canvas	14
Capítulo 2.	Especificaciones del producto	18
2.1	Requisitos funcionales	18
2.2	Requisitos no funcionales.....	22
Capítulo 3.	Normativa y legislación.....	26
3.1	Licencia OpenAI.....	26
3.2	Protección de datos y privacidad.....	26
3.3	Propiedad intelectual y derechos de autor.....	26
3.4	Regulaciones en el uso de inteligencia artificial.....	27
Capítulo 4.	Método y planificación.....	28
4.1	Metodología	28
4.2	Tecnología y soportes empleados	30
4.3	Herramientas del frontend.....	31
4.4	Herramientas del backend.....	32
4.5	Herramientas de soporte al método de trabajo	33
Capítulo 5.	Plan de trabajo ejecutado.....	37
5.1	Investigación y definición.....	37
5.2	Implementación del backend.....	37
5.3	Implementación del frontend.....	37
5.4	Integración de openAI.....	38
5.5	Pruebas y validaciones	38
5.6	Cambios y mejoras.....	38
5.7	Producto actual y propuesta trabajo fin de título (TFT_01)	39
Capítulo 6.	Desarrollo	40
6.1	Análisis del software	40
6.2	Diseño interfaz de usuario	42
6.3	Cursos de formación	60
6.4	Diseño de la base de datos	61
6.5	Configuración del proyecto	68
6.6	Ventajas implementación backend	80
6.7	Configuración frontend.....	81
6.8	Revisión de rendimiento y accesibilidad	91
Capítulo 7.	Conclusiones y trabajos futuros	94
7.1	Conclusiones.....	94
7.2	Trabajos futuros	95
	Referencias.....	97

Índice de ilustraciones

Ilustración 2-1: Replicación de MongoDB.....	23
Ilustración 4-1: Token generado al iniciar sesión	32
Ilustración 4-2: Muestra del entorno Visual Studio Code.....	34
Ilustración 4-3: Muestra de repositorio del proyecto	34
Ilustración 4-4: Ejemplo de método POST para generar exámenes de opción múltiple	35
Ilustración 4-5: Muestra del Trello inicial	36
Ilustración 4-6: Muestra Trello iteraciones completadas	36
Ilustración 6-1: Diagrama de Casos de Uso de las funcionalidades principales del proyecto	40
Ilustración 6-2: Registro de usuario.....	43
Ilustración 6-3: Inicio de sesión	43
Ilustración 6-4: Inicio de sesión con credenciales incorrectas	44
Ilustración 6-5: Creación nueva asignatura	44
Ilustración 6-6: Muestra del Dashboard.....	45
Ilustración 6-7: Contenido de la asignatura	45
Ilustración 6-8: Muestra diseño "Chat para Resolver Dudas"	46
Ilustración 6-9: Dialogo de "Chat para Resolver Dudas"	46
Ilustración 6-10: Muestra diseño "Generador de Resúmenes"	47
Ilustración 6-11: Dialogo de generador de resúmenes.....	47
Ilustración 6-12: Generador de exámenes de opción múltiple	48
Ilustración 6-13: Generador de exámenes de preguntas abiertas	49
Ilustración 6-14: Corrección examen de opción múltiple	50
Ilustración 6-15: Corrección examen de preguntas abiertas.....	51
Ilustración 6-16: Muestra del "Catálogo de Exámenes"	52
Ilustración 6-17: Repetición de examen del "Catálogo de Exámenes"	52
Ilustración 6-18: Secciones del perfil de usuario	53
Ilustración 6-19: Libro de calificaciones	53
Ilustración 6-20: Diagrama de barras	54
Ilustración 6-21: Diagrama de Sector	54
Ilustración 6-22: Gráfico de líneas calificaciones por asignatura a lo largo del tiempo	54
Ilustración 6-23: Gráfico de líneas calificaciones a lo largo del tiempo	55
Ilustración 6-24: Generador de examen en dispositivo móvil.....	55
Ilustración 6-25: Generador de exámenes con múltiples preguntas tipo test en dispositivo móvil.....	56
Ilustración 6-26: Generador de exámenes con múltiples preguntas abiertas en dispositivo móvil.....	57
Ilustración 6-27: Chat para resolver dudas en dispositivo móvil	58
Ilustración 6-28: Generador de resúmenes en dispositivo móvil.....	59
Ilustración 6-29: Muestra de documentos de la colección "users" de la base de datos ...	62
Ilustración 6-30: Muestra de documento de la colección "courses" de la base de datos .	63
Ilustración 6-31: Muestra de documento de la colección "exams" de la base de datos ...	64
Ilustración 6-32: Muestra de documento de la colección "summaries" de la base de datos	65
Ilustración 6-33: Muestra de documento de la colección "scores" de examen de opción múltiple.....	66

Ilustración 6-34: Muestra de documento de la colección “scores” de preguntas abiertas	67
Ilustración 6-35: Muestra de documento de la colección “savedexams” de la base de datos	68
Ilustración 6-36: Código archivo "authMiddleware.js"	71
Ilustración 6-37: Archivo que pertenecen a la carpeta uploads	73
Ilustración 6-38: Código archivo "pdfUtils.js"	73
Ilustración 6-39: Primera parte código archivo "server.js"	74
Ilustración 6-40: Segunda parte código archivo "server.js"	75
Ilustración 6-41: Código primera parte del archivo App.jsx	88
Ilustración 6-42: Código segunda parte del archivo App.jsx	89
Ilustración 6-43: Métricas de rendimiento con baja carga operativa	91
Ilustración 6-44: Métricas de rendimiento con alta carga operativa	91
Ilustración 6-45: Apartados de apariencia que cumple la aplicación web	92
Ilustración 6-46: Apartados de apariencia que carece la aplicación web	93

Índice de tablas

Tabla 1-1: Análisis modelo Lean Canvas.....	16
Tabla 4-1: Fases metodología iterativa	29
Tabla 4-2: Tecnologías empleadas	31
Tabla 5-1: Rutas sobre funcionalidades.....	37
Tabla 6-1: Detalle del caso de uso chatAI	41
Tabla 6-2: Detalle del caso de uso generación de exámenes	41
Tabla 6-3: Detalle del caso de uso generación de resúmenes.....	42
Tabla 6-4: Detalle del caso de uso CRUD de cursos	42
Tabla 6-5: Atributos "users"	62
Tabla 6-6: Atributos "courses".....	63
Tabla 6-7: Atributos "exams".....	64
Tabla 6-8: Atributos "summaries".....	64
Tabla 6-9: Atributos modelo "scores"	65
Tabla 6-10: Atributos "gradingResults".....	66
Tabla 6-11: Atributos modelo "savedExams".....	68
Tabla 6-12: Representación archivos carpeta "models/"	69
Tabla 6-13: Representación archivos carpeta "controllers/"	70
Tabla 6-14: Representación archivos carpeta "routes/"	72
Tabla 6-15: Rutas archivo "coursesRoutes.js"	72
Tabla 6-16: Rutas archivo "userRoutes.js"	73
Tabla 6-17: Ventajas implementación backend	80
Tabla 6-18: Ejemplo de paquetes incluidos de la carpeta node_modules	82
Tabla 6-19: Descripción archivos de la carpeta pages	83
Tabla 6-20: Archivos de subcarpeta components	85
Tabla 6-21: Componentes generales carpeta components	87

Capítulo 1. **Introducción**

En este apartado se explicará el motivo de realización del TFT, los objetivos del trabajo y aprendizaje con las competencias específicas de la titulación, además de un análisis de negocio siguiendo el método Lean Canvas.

1.1 Antecedentes

En la actualidad, el acceso a información de forma rápida y eficiente, impulsado por la digitalización, ha favorecido el aprendizaje que permite a los estudiantes y profesionales que adquieran conocimientos fácilmente. Además, el desarrollo de herramientas tecnológicas ha facilitado la automatización de tareas, reduciendo la carga manual en procesos repetitivos y permitiendo un enfoque más profundo en actividades estratégicas y creativas.

En este contexto, surge la necesidad de integrar tecnologías avanzadas, como la inteligencia artificial (IA) y el procesamiento del lenguaje natural (PLN) [1], en los procesos educativos y profesionales para personalizar la experiencia de aprendizaje, generan contenido educativo de calidad y permiten evaluar el desempeño de manera eficiente.

Este trabajo fin de título se enmarca en este escenario como una propuesta innovadora que aprovecha las capacidades de la IA para optimizar la gestión del conocimiento y la personalización del aprendizaje que al crear una sección de asignaturas permite realizar resúmenes y exámenes, así como resolver dudas.

Al abordar tareas complejas, como la generación de contenido basado en temas específicos y la evaluación personalizada, esta propuesta se convierte en una herramienta idónea para docentes y profesionales, reduciendo la carga operativa y mejorando la calidad del aprendizaje.

1.2 Objetivos del producto

El proyecto tiene como objetivo principal desarrollar una plataforma basada en inteligencia artificial (IA) que automatice procesos educativos clave, como la generación de exámenes, resúmenes y respuestas contextuales a preguntas. El producto se centra en integrar tecnologías avanzadas de procesamiento del lenguaje natural (PLN) que se entiende como una rama de la inteligencia artificial que permite que las computadoras entiendan, interpreten, generen y respondan al lenguaje humano y embeddings semánticos [2] que son representaciones de vectores numéricos que traducen palabras, frases o textos a números que las máquinas pueden procesar manteniendo el significado semántico. Para gestionar de manera eficiente grandes volúmenes de texto, proporcionando soluciones adaptativas y personalizadas en el ámbito académico y profesional.

El sistema estará diseñado para cumplir con las siguientes funcionalidades principales:

- Generación de exámenes: Crear automáticamente evaluaciones, tanto de opción múltiple como de preguntas abiertas, basadas en contenidos específicos proporcionados por el usuario.
- Resúmenes automáticos: Sintetizar información de documentos extensos, como textos en formato PDF, en fragmentos comprensibles y relevantes.
- Dudas específicas: Ofrecer respuestas precisas y adaptadas a preguntas basadas en un contexto definido, como materiales educativos o documentos relevantes.

1.3 Objetivos de aprendizaje

El objetivo de aprendizaje de este proyecto es desarrollar competencias en el diseño e implementación de soluciones tecnológicas innovadoras basadas en Inteligencia Artificial (IA) y Procesamiento del Lenguaje Natural (PLN) aplicadas al ámbito educativo. A través del desarrollo de una plataforma que automatiza la generación de exámenes, resúmenes y respuestas contextuales sobre dudas específicas, adquiriendo los siguientes conocimientos:

- Desarrollo de sistemas inteligentes aplicando técnicas de PLN y embeddings semánticos para el procesamiento y síntesis de información.
- Integración de modelos de IA en entornos educativos, optimizando la gestión del conocimiento y mejorando la personalización del aprendizaje.
- Diseño de soluciones tecnológicas escalables, considerando la experiencia del usuario, la accesibilidad y la interoperabilidad con plataformas existentes.
- Aplicación de metodologías ágiles y de Lean Canvas para la planificación, validación y desarrollo de productos digitales.
- Desarrollar habilidades prácticas en la creación e implementación de herramientas que optimicen el aprendizaje y la gestión del conocimiento.

1.3.1 Competencias específicas

FB4 – “Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación de la ingeniería.”

Se utiliza herramientas como bases de datos para almacenar información (calificaciones, usuarios, contenido del curso) y tecnología MERN [3] (MongoDB, Express, React y Node.js) para interactuar con el backend y frontend.

FB5 – “Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de la programación, y su aplicación para la resolución de problemas propios de la ingeniería.”

La aplicación web se desarrolla con un enfoque en la interconexión entre backend, frontend y bases de datos, utilizando programación en múltiples niveles (algoritmos para resúmenes y calificaciones).

CI1 – “Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.”

La aplicación web abarca el diseño y desarrollo de una aplicación web robusta que integra múltiples funcionalidades, asegurando una experiencia de usuario confiable y segura.

CI6 – “Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.”

Los algoritmos utilizados para dividir textos largos, calcular similitudes semánticas y gestionar la creación de exámenes y resúmenes demuestran el uso de técnicas algorítmicas para resolver problemas complejos.

CI7 – “Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.”

El almacenamiento de fragmentos de texto, usuarios, cursos, y calificaciones implica la elección adecuada de estructuras de datos para gestionar la información.

CI12 – “Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.”

La aplicación web utiliza bases de datos para almacenar información relacionada con los usuarios, cursos, calificaciones y exámenes.

CI13 – “Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web.”

La aplicación web integra herramientas para almacenar contenido (textos de PDF, resúmenes) y sistemas interactivos para generar y visualizar información.

CI16 - “Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del software.”

Se aplica un ciclo de desarrollo típico (planificación, diseño, implementación, pruebas) para crear una aplicación escalable y mantenible.

CI17 – “Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.”

El diseño de la interfaz web para el generador de exámenes, resúmenes y chat de consultas tiene en cuenta la interacción intuitiva y amigable para el usuario.

TI2 – “Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.”

La aplicación web combina tecnologías backend (Node.js, Express), frontend (React) y servicios externos (OpenAI) para ofrecer una solución eficiente y económica.

TI5 - “Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados.”

Permite gestionar información de cursos, calificaciones y consultas de manera centralizada y eficiente.

TI7 – “Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.”

La autenticación de usuarios mediante middleware como authMiddleware.js asegura la protección de los datos y el control de acceso a los recursos.

1.4 Análisis de negocio Lean Canvas

El método Lean Canvas [4] es una herramienta que permite visualizar rápidamente los aspectos clave de un proyecto o producto innovador en una única página. Su objetivo es identificar el problema, la solución, el segmento de clientes y otros elementos críticos, ayudando a validar ideas de forma ágil y efectiva.

Se ha empleado el método Lean Canvas para realizar un análisis de negocio que permite analizar, validar y ajustar la viabilidad del producto destacando prioridades claras, gestión de recursos y reducir significativamente el riesgo de impulsar un nuevo producto. A continuación, se divide con los siguientes apartados [4]:

- **Problema** :Identifica claramente las necesidades, dificultades o desafíos que enfrentan actualmente los clientes que justifica la creación del producto.
- **Segmento de clientes**: Describe quiénes serán los clientes a los que se dirige la solución propuesta, incluyendo organizaciones o instituciones que pueden estar interesadas en resolver el problema definido.
- **Propuesta de valor**: Explica de manera concreta el beneficio diferencial que ofrece el producto destacando cómo satisface las necesidades identificadas y por qué los clientes deberían elegirlo frente a otras alternativas.
- **Solución**: Define las características específicas del producto, detallando cómo resolverá los problemas descritos y cómo se presentará ante el usuario.
- **Canales de Distribución**: Indica las vías a través de los cuales se hará llegar el producto, detallando cómo los usuarios accederán, utilizarán o conocerán la plataforma.
- **Fuentes de Ingreso**: Establece posibles maneras que el producto podrá generar ingresos o se financiará para monetizar o comercializar la solución propuesta.
- **Métricas Clave**: Describe los indicadores o parámetros que se emplean para medir el rendimiento, éxito y valor entregado por el proyecto, permitiendo realizar un seguimiento del progreso.
- **Ventaja Competitiva**: Detalla factores que permite al producto destacar frente a otras opciones del mercado, incluyendo características únicas, diferenciadores tecnológicos y valor agregado para el cliente.

- **Costos Estructurales:** Detalla los principales gastos, recursos o inversiones necesarios para operar y sostener el proyecto, desde tecnología y desarrollo hasta mantenimiento y marketing, proporcionando claridad sobre la estructura financiera que se requiere para su implementación.

A continuación, se presenta el modelo Lean Canvas correspondiente al proyecto, con el objetivo de representar de forma clara y estructurada los elementos esenciales que definen la plataforma web desarrollada.

PROBLEMA	SOLUCIÓN	PROPUESTA DE VALOR ÚNICA	VENTAJA ESPECIAL
	MÉTRICAS CLAVE		SEGMENTO DE CLIENTES
Falta de herramientas automatizadas para la generación de exámenes, resúmenes y respuestas contextuales. Alta carga operativa en docentes para preparar material educativo. Dificultades de los estudiantes para obtener información sintetizada y personalizada.	Generación automática de exámenes (opción múltiple y preguntas abiertas). Resúmenes automáticos de documentos PDF. Asistente virtual para resolver dudas basadas en el material. Interfaz intuitiva y fácil de usar para docentes y estudiantes.	Crear resúmenes automáticos. Generar exámenes personalizados. Resolver dudas específicas. Ofrecer aprendizaje personalizado.	Uso de modelos avanzados de IA y PLN. Automatización de procesos educativos. Personalización del aprendizaje. Escalabilidad y adaptabilidad permitiendo su uso en múltiples entornos.
ESTRUCTURA DE COSTOS		FLUJO DE INGRESOS	
Mantenimiento de la plataforma (OpenAI). Investigación y actualización de modelos IA. Marketing para adquirir clientes.		Licenciamiento institucional. Consultoría y personalización para empresas.	

Tabla 1-1: Análisis modelo Lean Canvas

Respecto a las métricas clave, un estudiante individual puede acceder de forma gratuita a sus funcionalidades, pero si es de servicio empresarial para un conjunto de estudiantes el colegio o la universidad deberá abonar una licencia y registrarse como servicio empresarial con las indicaciones dadas por el administrador.

Como se puede apreciar en la tabla representada, desde una perspectiva de negocio, el modelo permite prever vías de ingreso factibles y canales de distribución, por lo que existe un marco inicial sólido para la sostenibilidad económica del proyecto.

En cuanto a costes, el modelo plantea un equilibrio razonable entre inversión tecnológica y retorno potencial, aunque dependerá del uso eficiente de recursos como las APIs de IA. Este análisis también ha ayudado a detectar posibles riesgos, como la dependencia de terceros en este caso OpenAI.

La presencia de canales como una plataforma web accesible y estrategias de marketing digital facilita la llegada al público objetivo sin requerir una infraestructura comercial compleja.

En conclusión, se permite anticipar su viabilidad tanto funcional como comercial. Su aplicación facilita la toma de decisiones estratégicas informadas, al tiempo que identifica con claridad los aspectos que requieren validación en etapas posteriores, como la aceptación por parte de los usuarios y la sostenibilidad económica a largo plazo.

Capítulo 2. Especificaciones del producto

En este apartado se especificará los requisitos funcionales [5] , no funcionales [6] y la normativa y legislación sobre las características de la aplicación web.

2.1 Requisitos funcionales

Se detalla una lista de todos los requisitos funcionales del producto:

1. El usuario debe registrarse con un correo electrónico, nombre y contraseña y opcionalmente una foto de perfil.
2. El sistema debe permitir que solo los usuarios previamente registrados inicien sesión
3. El sistema debe validar que las credenciales proporcionadas sean correctas.
4. El sistema debe proteger las rutas sensibles con middleware de autenticación.
5. El usuario debe crear nuevos cursos proporcionando un nombre, descripción y documentación.
6. El sistema debe extraer automáticamente el texto de la documentación subida.
7. Los usuarios desde el dashboard deben visualizar una lista de todos los cursos disponibles.
8. Los usuarios deben poder seleccionar un curso para ver su contenido, incluyendo el material del curso subido previamente.
9. El usuario debe poder crear el material asociado a un curso.
10. El usuario debe poder editar el material asociado a un curso.
11. El usuario debe poder eliminar el material asociado a un curso.
12. El usuario debe poder descargar el material que ha subido previamente.
13. Los usuarios deben poder enviar preguntas al sistema relacionadas con un curso o tema específico.
14. El sistema debe generar respuestas utilizando la IA basada en el contenido de la documentación subida de un curso.
15. Si el usuario sube documentación el sistema debe usar su contenido como contexto para responder preguntas.
16. El sistema debe seleccionar fragmentos relevantes del texto para construir una respuesta precisa.
17. El usuario debe poder generar un resumen sobre un curso.
18. El usuario debe poder asignar un límite de palabras para generar el resumen.
19. El usuario debe poder detallar un tema para generar el resumen.
20. El usuario debe poder crear un examen sobre un curso.
21. El usuario debe especificar el número de preguntas para generar un examen.

22. El usuario debe especificar el tipo de examen si es opción múltiple o preguntas abiertas para generar un examen.
23. El sistema debe poder generar respuestas automáticamente basándose en el contenido del curso.
24. El sistema debe permitir guardar un examen generado para poder realizarlo posteriormente.
25. El sistema debe poder calificar automáticamente las preguntas de opción múltiple basándose en las respuestas correctas.
26. El sistema debe proporcionar retroalimentación para preguntas abiertas basándose en análisis semántico.
27. El sistema debe permitir desde el “Catálogo de Exámenes” visualizar los exámenes que hayan sido guardado previamente.
28. El sistema debe permitir repetir un examen redirigiendo al “Generador de exámenes” para realizar el examen con las mismas preguntas y calificarlo.
29. El usuario debe poder visualizar las calificaciones de todos los exámenes desde el libro de calificaciones.
30. El usuario debe poder visualizar de nuevo la corrección del examen desde el libro de calificaciones.
31. El usuario debe poder visualizar desde la sección “Estadísticas” el número de exámenes realizados.
32. El usuario debe poder visualizar desde la sección “Estadísticas” un gráfico de barras sobre el promedio de calificaciones obtenidas en los exámenes por asignatura.
33. El usuario debe poder visualizar en la sección "Estadísticas" un diagrama de sectores sobre el número de exámenes realizados de opción múltiple y preguntas abiertas
34. El usuario debe poder visualizar en la sección "Estadísticas" un gráfico lineal sobre las calificaciones obtenidas por asignatura lo largo del tiempo.
35. El usuario debe poder visualizar en la sección "Estadísticas" un gráfico lineal sobre las calificaciones obtenidas en los exámenes a lo largo del tiempo.

2.1.1 Gestión de usuarios

- El usuario debe registrarse con un correo electrónico, nombre y contraseña y opcionalmente una foto de perfil.
- El sistema debe permitir que solo los usuarios previamente registrados inicien sesión
- El sistema debe validar que las credenciales proporcionadas sean correctas.
- El sistema debe proteger las rutas sensibles con middleware de autenticación.

2.1.2 Gestión de cursos

- El usuario debe crear nuevos cursos proporcionando un nombre, descripción y documentación.
- El sistema debe extraer automáticamente el texto de la documentación subida.

2.1.3 Visualización de cursos

- Los usuarios desde el dashboard deben visualizar una lista de todos los cursos disponibles.
- Los usuarios deben poder seleccionar un curso para ver su contenido, incluyendo el material del curso subido previamente.

2.1.4 Gestión material del curso

- El usuario debe poder crear el material asociado a un curso.
- El usuario debe poder editar el material asociado a un curso.
- El usuario debe poder eliminar el material asociado a un curso.
- El usuario debe poder descargar el material que ha subido previamente.

2.1.5 Resolución de dudas (Chat AI)

- Los usuarios deben poder enviar preguntas al sistema relacionadas con un curso o tema específico.
- El sistema debe generar respuestas utilizando la IA basada en el contenido de la documentación subida de un curso.

2.1.6 Generación de dudas específicas

- Si el usuario sube documentación el sistema debe usar su contenido como contexto para responder preguntas.
- El sistema debe seleccionar fragmentos relevantes del texto para construir una respuesta precisa.

2.1.7 Generación de resúmenes

- El usuario debe poder generar un resumen sobre un curso.
- El usuario debe poder asignar un límite de palabras para generar el resumen.
- El usuario debe poder detallar un tema para generar el resumen.

2.1.8 Generador de exámenes

- El usuario debe poder crear un examen sobre un curso.
- El usuario debe especificar el número de preguntas para generar un examen.
- El usuario debe especificar el tipo de examen si es opción múltiple o preguntas abiertas para generar un examen.

2.1.9 Generación automática

- El sistema debe poder generar respuestas automáticamente basándose en el contenido del curso.

2.1.10 Guardar examen generado

- El sistema debe permitir guardar un examen generado para poder realizarlo posteriormente.

2.1.11 Corrección y calificación

- El sistema debe poder calificar automáticamente las preguntas de opción múltiple basándose en las respuestas correctas.
- El sistema debe proporcionar retroalimentación para preguntas abiertas basándose en análisis semántico.

2.1.12 Catálogo de exámenes

- El sistema debe permitir desde el “Catálogo de Exámenes” visualizar los exámenes que hayan sido guardado previamente.
- El sistema debe permitir repetir un examen redirigiendo al “Generador de exámenes” para realizar el examen con las mismas preguntas y calificarlo.
- El usuario debe poder visualizar las calificaciones de todos los exámenes desde el libro de calificaciones.
- El usuario debe poder visualizar de nuevo la corrección del examen desde el libro de calificaciones.

2.1.13 Estadísticas

- El usuario debe poder visualizar desde la sección “Estadísticas” el número de exámenes realizados.
- El usuario debe poder visualizar desde la sección “Estadísticas” un gráfico de barras sobre el promedio de calificaciones obtenidas en los exámenes por asignatura.
- El usuario debe poder visualizar en la sección "Estadísticas" un diagrama de sectores sobre el número de exámenes realizados de opción múltiple y preguntas abiertas.

- El usuario debe poder visualizar en la sección "Estadísticas" un gráfico lineal sobre las calificaciones obtenidas por asignatura a lo largo del tiempo.
- El usuario debe poder visualizar en la sección "Estadísticas" un gráfico lineal sobre las calificaciones obtenidas en los exámenes a lo largo del tiempo.

2.2 Requisitos no funcionales

Se detalla una lista de todos los requisitos no funcionales del producto:

1. El sistema debe procesar y responder a solicitudes comunes (como autenticación, envío de datos) en un tiempo inferior a 500 ms.
2. Operaciones más complejas, como la generación de resúmenes o exámenes, deben completarse en menos de 5 segundos en condiciones normales.
3. Las tareas que requieren interacción con la API de OpenAI
4. deben ejecutarse de manera asíncrona para no bloquear la interfaz del usuario.
5. División de textos largos en fragmentos manejables para evitar exceder los límites de tiempo o tokens de OpenAI.
6. MongoDB debe configurarse para manejar grandes volúmenes de datos mediante replicación de tipo "Replica Set – 3 nodes"
7. Se utiliza JWT (JSON Web Tokens) para garantizar sesiones seguras de usuario.
8. Todos los datos sensibles (contraseñas, tokens) se cifran antes de almacenarse en la base de datos.
9. Compatible con navegadores modernos en sistemas Windows, macOS y Linux.
10. El frontend está optimizado para uso en dispositivos móviles y de escritorio.
11. El backend interactúa directamente con OpenAI API para generar resúmenes, exámenes y respuestas en el chat de dudas.
12. Soporte para archivos PDF en el módulo de resúmenes y procesamiento de documentación.
13. Diseño responsivo con Bootstrap
14. para una experiencia fluida en dispositivos móviles y de escritorio.
15. Navegación intuitiva con menús claros y formularios validados en tiempo real.
16. El código está organizado en componentes reutilizables (frontend) y controladores separados (backend) para facilitar el mantenimiento.
17. Uso de Git para control de versiones y seguimiento del desarrollo del proyecto.
18. Implementación de control de versiones con Git para rastrear cambios, colaborar y mantener historial del código.
19. Gestión del historial de cambios del proyecto mediante Git.
20. Implementación futura de pruebas automatizadas para verificar la funcionalidad crítica.

21. El sistema debe manejar errores comunes (como tiempo de espera en la API de OpenAI) mediante mensajes de error claros para el usuario.
22. Se espera una disponibilidad completa con el sistema en condiciones normales.

2.2.1 Requisitos de rendimiento

Tiempo de respuesta

- El sistema debe procesar y responder a solicitudes comunes (como autenticación, envío de datos) en un tiempo inferior a 500 ms.
- Operaciones más complejas, como la generación de resúmenes o exámenes, deben completarse en menos de 5 segundos en condiciones normales.

Procesamiento asíncrono

Las tareas que requieren interacción con la API de OpenAI [7] deben ejecutarse de manera asíncrona para no bloquear la interfaz del usuario.

División de textos largos en fragmentos manejables para evitar exceder los límites de tiempo o tokens de OpenAI.

2.2.2 Escalabilidad

Base de datos

VERSION	REGION	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SQL	ATLAS SEARCH
8.0.8	AWS / Frankfurt (eu-central-1)	Replica Set - 3 nodes	Inactive	None Linked	Connect	Create Index
+ Add Tag						

Ilustración 2-1: Replicación de MongoDB

- MongoDB debe configurarse para manejar grandes volúmenes de datos mediante replicación de tipo “Replica Set – 3 nodes” [8].

2.2.3 Seguridad

Autenticación y autorización

- Se utiliza JWT (JSON Web Tokens) para garantizar sesiones seguras de usuario.

Protección de datos

- Todos los datos sensibles (contraseñas, tokens) se cifran antes de almacenarse en la base de datos.

2.2.4 Compatibilidad e interoperabilidad

Sistemas operativos

Compatible con navegadores modernos en sistemas Windows, macOS y Linux.

El frontend está optimizado para uso en dispositivos móviles y de escritorio.

Integración con OpenAI

El backend interactúa directamente con OpenAI API para generar resúmenes, exámenes y respuestas en el chat de dudas.

Compatibilidad de archivos

Soporte para archivos PDF en el módulo de resúmenes y procesamiento de documentación.

2.2.5 Usabilidad

Interfaz de usuario

Diseño responsivo con Bootstrap [9] para una experiencia fluida en dispositivos móviles y de escritorio.

Navegación intuitiva con menús claros y formularios validados en tiempo real.

2.2.6 Mantenibilidad

Código modular

El código está organizado en componentes reutilizables (frontend) y controladores separados (backend) para facilitar el mantenimiento.

Control de versiones

Uso de Git para control de versiones y seguimiento del desarrollo del proyecto.

Implementación de control de versiones con Git para rastrear cambios, colaborar y mantener historial del código.

Gestión del historial de cambios del proyecto mediante Git.

Pruebas automatizadas

Implementación futura de pruebas automatizadas para verificar la funcionalidad crítica.

2.2.7 Requisitos de fiabilidad

Tolerancia a fallos

El sistema debe manejar errores comunes (como tiempo de espera en la API de OpenAI) mediante mensajes de error claros para el usuario.

Disponibilidad

Se espera una disponibilidad completa con el sistema en condiciones normales.

Capítulo 3. Normativa y legislación

El desarrollo se mantiene en un entorno local y se utilizan datos generados por el propio usuario, no hay riesgo de infracción legal, ya que no se está compartiendo ni comercializando contenido protegido.

Si en un futuro se desea entrar en producción hay que tener en cuenta todas las normativas que se describen en los siguientes apartados.

3.1 Licencia OpenAI

Al utilizar la API de OpenAI, StudyUniv está sujeto a los términos y condiciones establecidos por OpenAI, lo que implica:

- Uso autorizado: La API solo puede utilizarse para los fines permitidos por OpenAI.
- Restricciones: No se puede usar para generar contenido malicioso, engañoso o ilegal.
- Seguridad: Se debe proteger la clave API y evitar su exposición pública.

3.1.1 Acción recomendada

- No compartir públicamente la clave de la API en repositorios de GitHub o archivos accesibles.
- Revisar periódicamente los términos de OpenAI para asegurarse de que el uso de la API sigue cumpliendo con sus directrices.

3.2 Protección de datos y privacidad

Debe manejar la información de usuarios, así como sus datos personales y sus contenidos educativos, deben seguir las siguientes normativas:

- Reglamento General de Protección de Datos (GDPR - UE 2016/679) [10]: Para los usuarios de la Unión Europea, impone requisitos sobre la recopilación, almacenamiento, uso de datos personales y que la aplicación web contenga políticas de privacidad y permisos del usuario.
- Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD - España) [11]: Complementa el GDPR en España para mejorar la privacidad y datos personales.

3.3 Propiedad intelectual y derechos de autor

Emplea código abierto por ejemplo con las tecnologías Express.js, React.js, MongoDB, por lo tanto, debe cumplir con las licencias de cada componente como MIT [12], Apache 2.0 [13] o GNU GPL [14].

3.4 Regulaciones en el uso de inteligencia artificial

Al emplear IA en la aplicación web StudyUniv para generar contenido y respuestas automáticas debe ajustarse a Principios de IA Responsable de la Organización para la Cooperación y el Desarrollo Económico (OCDE) [15] que respete los derechos humanos y privacidad.

Capítulo 4. Método y planificación

En este apartado se explicará con detalle la metodología utilizada, tecnologías de desarrollo, plan de trabajo ejecutado y estimación de costes.

4.1 Metodología

Se ha desarrollado la aplicación web con una metodología iterativa incremental, descrito en el TFT_01:

Fases	Duración Estimada (horas)	Tareas <i>(nombre y descripción, obligatorio al menos una por fase)</i>
Modelado Requisitos	/ 60	Tarea 1.1: Nombre: Se configuran las tecnologías principales del proyecto: Frontend con React, backend utilizando Node.js con Express, y base de datos MongoDB. Esto incluye la instalación de dependencias, configuración inicial de los servidores y estructura básica del proyecto.
		Tarea 1.2: Nombre: Recolección y análisis de requisitos funcionales Descripción: Identificación de funcionalidades clave: carga de documentos, generación de resúmenes, creación de exámenes personalizados, gestión de perfiles de usuario.
		Tarea 1.3: Nombre: Diseño del modelo de datos. Descripción: Se define el esquema principal para MongoDB, incluyendo las colecciones para usuarios, actividades, documentos subidos y resultados de los exámenes.
	140	Tarea 2.1:

Diseño / Implementación		Nombre: Desarrollo del frontend. Descripción: Creación y gestión del perfil de usuario, creación de las vistas principales con React, incluyendo interfaces para la carga de documentos, visualización de resúmenes, generación de exámenes.
		Tarea 2.2: Nombre: Implementación del backend. Descripción: Desarrollo de las rutas de la API con Express para manejar solicitudes del frontend, autenticación de usuarios, integración con la API de OpenAI y gestión de datos en MongoDB. Tarea 2.3: Nombre: Configuración de comunicación entre frontend y backend. Descripción: Integración de las solicitudes de API del frontend al backend.
Test / Despliegue	70	Tarea 3.1: Nombre: Pruebas funcionales Descripción: Verificación de cada funcionalidad principal, como carga de documentos, resúmenes generados, personalización de exámenes y retroalimentación de resultados.
Documentación / Presentación	30	Tarea 4.1: Nombre: Documentación del proyecto Descripción: Crear un documento que explique todas las funcionalidades de la web.

Tabla 4-1: Fases metodología iterativa

4.2 Tecnología y soportes empleados

En este apartado se describen las herramientas, tecnologías y recursos utilizados durante el desarrollo de la aplicación web.

Se muestra a continuación una tabla con todas las tecnologías empleadas y soporte al método de trabajo:

Área	Herramienta	Función principal
Frontend	React.js [16]	Interfaz de usuario y gestión del estado.
	Bootstrap	Estilos CSS y diseño responsive.
	react-pdfotext [17]	Lectura de contenido PDF en el navegador.
	React Router DOM [18]	Gestión de rutas en aplicaciones SPA.
	React Spinners [19]	Indicadores visuales de carga.
	Recharts [20]	Visualización de gráficos estadísticos.
	SweetAlert2 [21]	Alertas modales personalizadas.
	JWT Decode [22]	Decodificación de tokens JWT en el cliente.
	Axios [23]	Solicitudes HTTP entre frontend y backend.
Comunicación	Node.js [24] + Express.js [25]	Servidor, rutas y lógica de negocio.
	Multer [26]	Subida y procesamiento de archivos (PDFs).
	express-fileupload [27]	Alternativa para manejo de archivos subidos.
	MongoDB	Base de datos para usuarios, contenido y resúmenes.
	OpenAI API	Generación de resúmenes, preguntas y respuestas.
	pdf-parse [28]	Extracción de texto desde archivos PDF.
	bcryptjs [29]	Encriptación de contraseñas.
	jsonwebtoken [30]	Generación y validación de tokens JWT.
	cors [31]	Permitir solicitudes entre dominios distintos.
	dotenv [32]	Gestión de variables de entorno.

Soporte al método de trabajo	Visual Studio Code [33]	Editor principal para escribir, depurar y mantener el código.
	Git [34] + GitHub [35]	Control de versiones y alojamiento de repositorios.
	Postman [36]	Pruebas y verificación del funcionamiento de la API.
	Trello [37]	Gestión de tareas e iteraciones del proyecto.

Tabla 4-2: Tecnologías empleadas

4.3 Herramientas del frontend

En el área de frontend, la tabla recoge una selección de tecnologías con el objetivo de construir una interfaz moderna, funcional y visualmente atractiva.

React.js constituye la base del desarrollo, permitiendo gestionar el estado de la aplicación y estructurar la interfaz en componentes reutilizables.

Complementariamente, se han utilizado **Bootstrap** para garantizar un diseño responsive y consistente, mientras que bibliotecas como **React Spinners** y **SweetAlert2** se han utilizado para enriquecer la experiencia visual del usuario: los spinners aparecen durante los procesos de carga, como la generación de resúmenes o la subida de archivos, indicando que el sistema está procesando la solicitud; por su parte, SweetAlert2 permite mostrar mensajes emergentes de confirmación, advertencia o error de forma clara y atractiva, como cuando se completa una operación, se detecta un fallo o se requiere una validación previa.

Para el enrutamiento interno de la aplicación se ha empleado **React Router DOM**. Además, se ha integrado la biblioteca **Recharts** para estadísticas de resultados y gráficos. Por su parte, **react-pdfotext** permite extraer automáticamente el contenido de los documentos PDF cargados por el usuario, para ser procesado por la inteligencia artificial.

Finalmente, **JWT Decode JWT** es una tecnología estándar para la autenticación basada en tokens, que garantiza que solo los usuarios autenticados puedan acceder a funcionalidades sensibles de la aplicación.

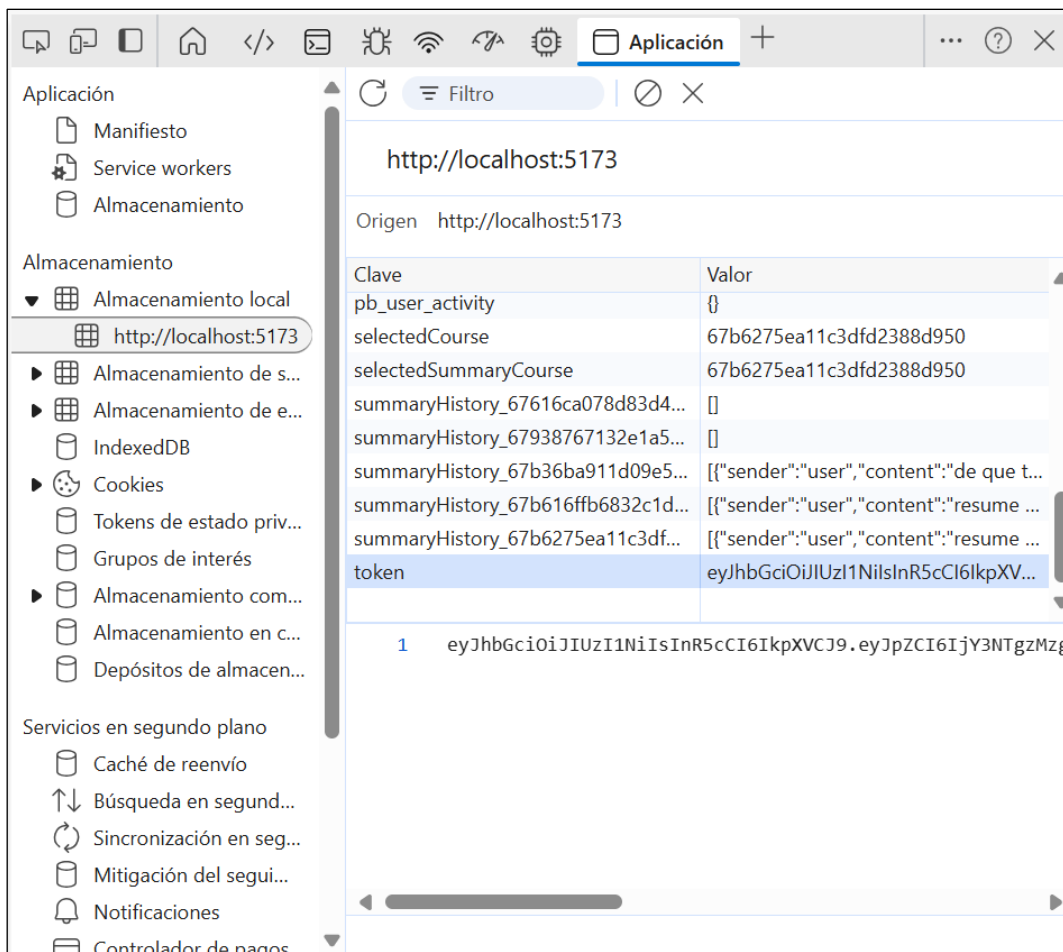


Ilustración 4-1: Token generado al iniciar sesión

4.4 Herramientas del backend

En la parte backend del proyecto se han utilizado diversas herramientas orientadas a garantizar una estructura robusta, segura y eficiente para el procesamiento de datos, la lógica del servidor y la integración con servicios externos. La base de esta capa la componen **Node.js** y **Express.js**, tecnologías que permiten ejecutar código en el servidor, definir rutas y manejar **peticiones HTTP** de forma flexible. Estas herramientas gestionan toda la lógica de la aplicación, desde la validación de datos hasta la interacción con la base de datos y la comunicación con la **API de OpenAI**.

Para el almacenamiento de información se ha empleado **MongoDB**, una base de datos NoSQL que permite guardar de forma eficiente datos estructurados y

no estructurados, como usuarios, cursos, resúmenes generados o resultados de evaluación. La conexión con MongoDB se ha realizado mediante Mongoose.

Una parte central del backend es la integración con la API de OpenAI, encargada de generar automáticamente contenido como resúmenes, preguntas de evaluación y respuestas basadas en el texto cargado por el usuario.

En cuanto al manejo de archivos, se han empleado dos herramientas: **Multer** y **express-fileupload**, que permiten la carga y gestión de documentos PDF enviados por los usuarios. Estos archivos son procesados para extraer su contenido textual mediante la librería **pdf-parse**, que transforma el texto en datos que pueden ser interpretados por la IA.

Para asegurar el acceso a funcionalidades protegidas, se ha implementado un sistema de autenticación basado en **JSON Web Tokens (JWT)**, utilizando las librerías **jsonwebtoken**, **bcrypt** y **bcryptjs** para la generación de tokens y el cifrado de contraseñas. A su vez, la librería **cors** ha sido integrada para permitir peticiones entre dominios diferentes (frontend y backend), y **dotenv** se ha utilizado para gestionar de forma segura las variables de entorno sensibles, como claves de API o credenciales de conexión.

4.5 Herramientas de soporte al método de trabajo

Se han empleado diversas herramientas de soporte que han facilitado la organización, el control de versiones, la prueba de funcionalidades y el mantenimiento del código a lo largo del proyecto.

En primer lugar, se ha utilizado Visual Studio Code como entorno de desarrollo principal. Este editor de código ligero y altamente personalizable ha permitido trabajar de manera eficiente con múltiples tecnologías, integrando funcionalidades como depuración, resaltado de sintaxis, extensiones específicas para JavaScript y control de versiones integrado. Su uso ha sido imprescindible para implementar la aplicación web.

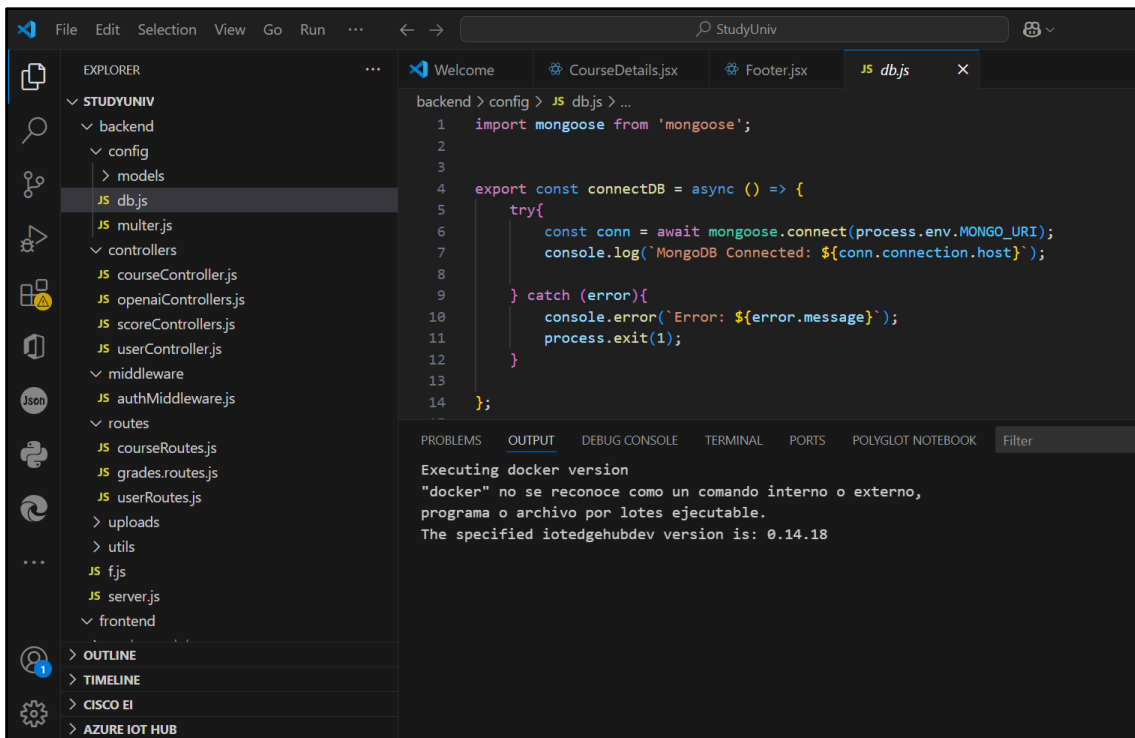


Ilustración 4-2: Muestra del entorno Visual Studio Code

Para el control de versiones se ha utilizado Git, junto con la plataforma remota GitHub. Estas herramientas han facilitado el seguimiento de cambios y la gestión de ramas, permitiendo además mantener un historial estructurado del desarrollo del proyecto. GitHub ha sido también el lugar donde se ha alojado el repositorio, disponible para consulta o despliegue desde cualquier dispositivo.

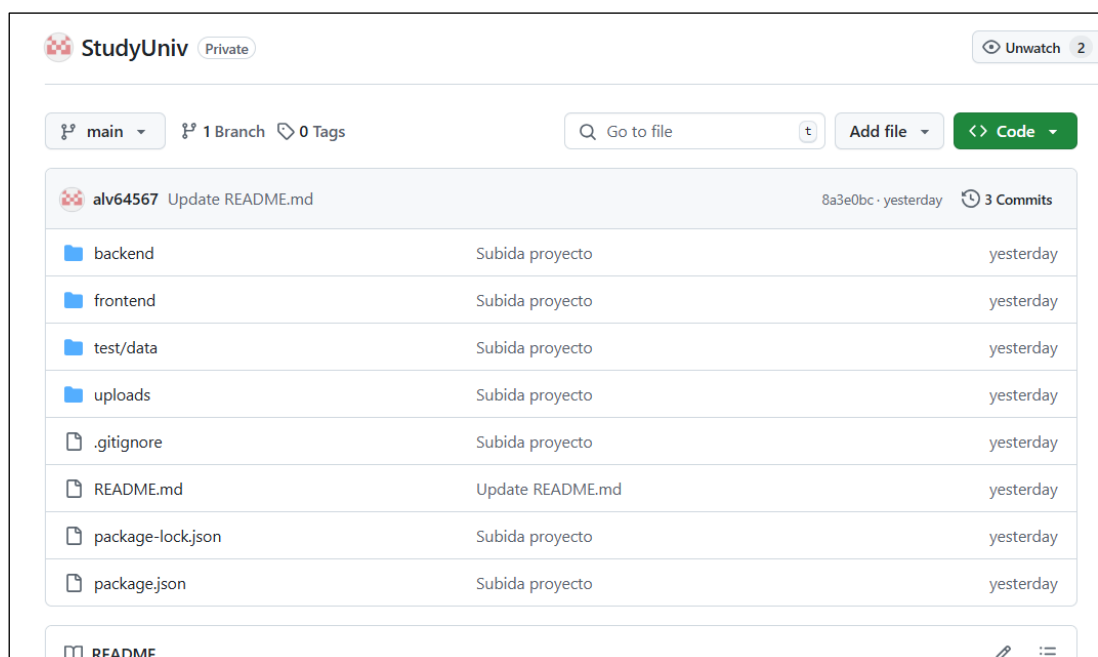


Ilustración 4-3: Muestra de repositorio del proyecto

Durante la fase de pruebas del backend, se ha recurrido a Postman, una herramienta ampliamente utilizada para verificar y documentar APIs REST. Postman ha permitido enviar peticiones GET y POST a los distintos endpoints de la API desarrollada, comprobando el comportamiento esperado del sistema, especialmente en funcionalidades clave como la generación de exámenes o la autenticación de usuarios.

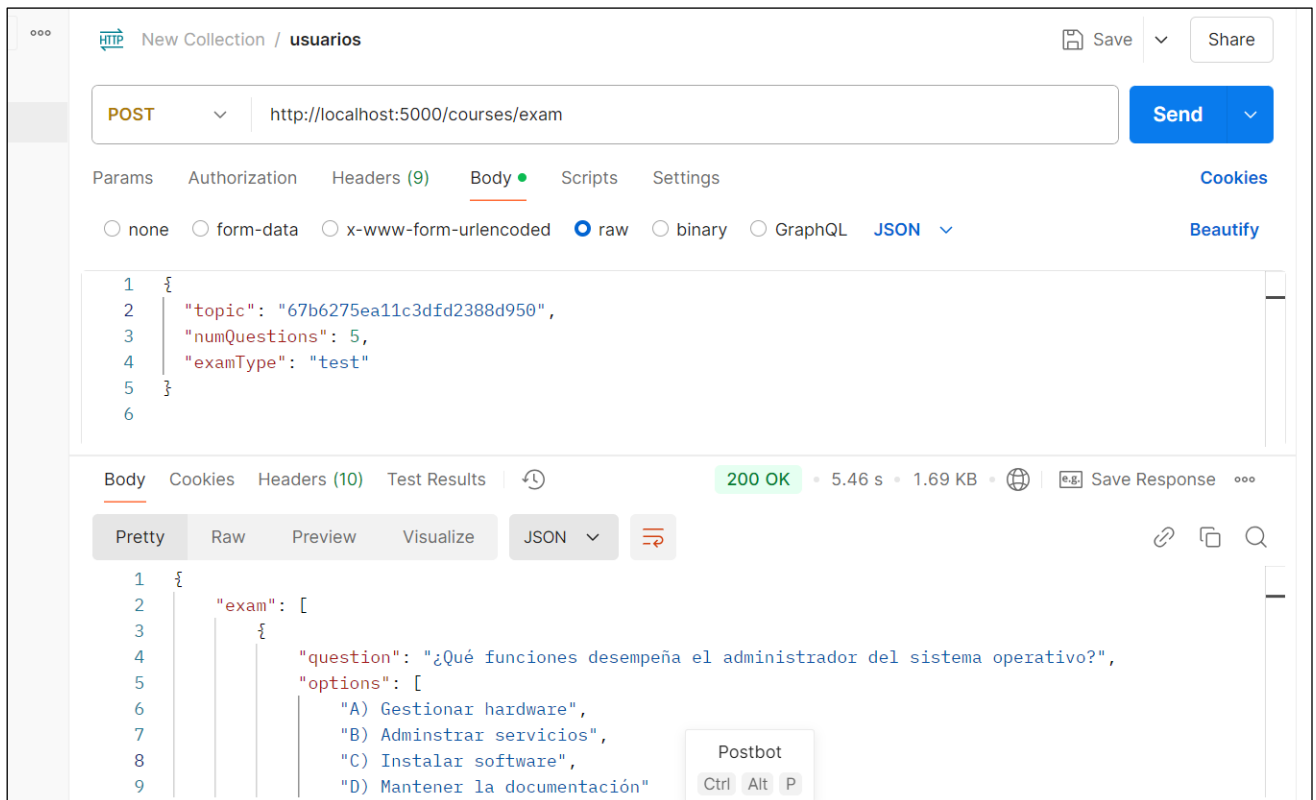


Ilustración 4-4: Ejemplo de método POST para generar exámenes de opción múltiple

Por último, para la organización del flujo de trabajo, se ha empleado Trello, una herramienta de gestión de tareas basada en tableros. Esta aplicación ha sido útil para planificar y visualizar el avance del proyecto, organizado por historias de usuario compuesto por las siguientes fases “Backlog”, “En progreso”, “En revisión”, “Completado iteración 0”, “Completado iteración 1”, “Completado iteración 2” “Completado iteración 3”.

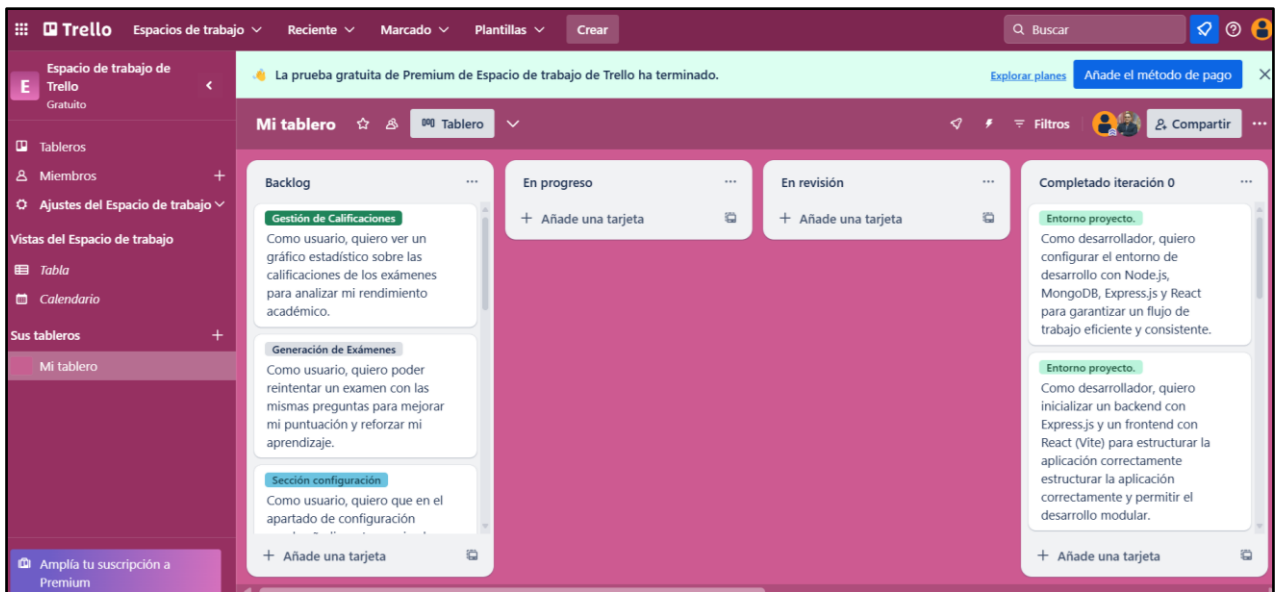


Ilustración 4-5: Muestra del Trello inicial



Ilustración 4-6: Muestra Trello iteraciones completadas

Capítulo 5. Plan de trabajo ejecutado

El desarrollo de la web fue llevado a cabo en varias etapas siguiendo un enfoque iterativo. A continuación, se detalla el trabajo realizado en cada fase, destacando los logros, las dificultades enfrentadas y las decisiones tomadas durante la implementación:

5.1 Investigación y definición

- Revisión de los Objetivos: Se revisaron los objetivos definidos en la planificación previa para asegurarse de que todas las funcionalidades principales estuvieran alineadas con las necesidades de los usuarios.
- Validación de Requisitos: Se analizaron nuevamente los requisitos funcionales y no funcionales para priorizar las tareas clave.
- Adopción de Nuevas Herramientas: Durante la implementación, se decidió incorporar herramientas adicionales como Mongoose para la gestión de la base de datos MongoDB y Vite para mejorar el rendimiento del frontend.

5.2 Implementación del backend

Se creó el servidor backend utilizando Node.js y Express, se configuraron middlewares esenciales como cors y express.json, para manejar solicitudes de clientes y JSON.

Se implementaron rutas organizadas para manejar diferentes aspectos de la funcionalidad:

Rutas	Funcionalidad
/login	Iniciar sesión.
/register	Registrar usuario.
/courses	Operaciones CRUD relacionadas con cursos y materiales.
/summary	Generación de resúmenes utilizando OpenAI.
/exam	Generación y evaluación de exámenes.
/chat	Interacción con Chat AI.

Tabla 5-1: Rutas sobre funcionalidades

Respecto a la base de datos, se diseñaron esquemas en Mongoose para usuarios, cursos, exámenes, calificaciones y materiales, asegurando una estructura clara y flexible.

Por último, se utilizó Postman para probar cada endpoint, identificando errores y optimizando respuestas.

5.3 Implementación del frontend

Se utilizó React para construir componentes reutilizables, como Navbar, Footer y módulos funcionales (ChatAI, ExamGenerator, SummaryGenerator).

Además, se configuraron rutas utilizando React Router, con soporte para rutas protegidas mediante un componente ProtectedRoute.

Se diseñaron pantallas amigables para los usuarios, en la página principal esta las opciones de iniciar sesión o registrarse.

Una vez que el registrado e iniciada la sesión, en el panel de control (Dashboard) para acceder a las funcionalidades clave.

Se emplearon archivos CSS individuales para cada componente, garantizando modularidad y fácil mantenimiento.

5.4 Integración de openAI

- Conexión con la API de OpenAI: Se integró OpenAI en el backend para proporcionar las siguientes funcionalidades:
- Generación de resúmenes: En base a documentos cargados o texto proporcionado.
- Creación de preguntas: Puede ser opción múltiple o preguntas abiertas.
- Respuestas a preguntas: En el módulo de Chat AI.
- Optimización de Solicitudes: Se implementaron divisores de texto para manejar largos documentos de manera eficiente.
- Pruebas de Respuestas: Se realizaron pruebas exhaustivas de las respuestas generadas por OpenAI, asegurando precisión y relevancia.

5.5 Pruebas y validaciones

- Pruebas Funcionales: Cada módulo fue probado individualmente para verificar que cumpliera con sus objetivos.
- Pruebas de Integración: Se aseguraron interacciones fluidas entre el frontend y el backend, verificando flujos como autenticación y generación de contenido.
- Pruebas de Usabilidad: Usuarios simulados probaron la web, proporcionando retroalimentación sobre la experiencia general.
- Pruebas de Carga: Se evaluó la capacidad de respuesta del sistema con múltiples solicitudes concurrentes, ajustando configuraciones del servidor cuando fue necesario.

5.6 Cambios y mejoras

En el módulo de generación de resúmenes, se añadió una funcionalidad para limitar la longitud del texto generado. Se realizaron ajustes en las interfaces para mejorar la usabilidad y accesibilidad. Sobre la gestión de errores se implementaron mensajes de error descriptivos para guiar al usuario cuando ocurren problemas.

5.7 Producto actual y propuesta trabajo fin de título (TFT_01)

Las funciones de la aplicación web cumple con todos los requisitos descritos en el TFT_01, ya que como objetivo se propuso que se pueda obtener resúmenes de asignaturas y de temas o tópicos basados en la materia. Además de un chat para resolver dudas y realizar exámenes y obtener sus resultados. Cabe destacar que cumple con todos los objetivos y además se realizó las siguientes ampliaciones: “Catálogo de exámenes”, “Libro de calificaciones “, “Estadísticas”. En la propuesta también está descrito que se podrá “ofrecer otras funcionalidades accesorias a la gestión de cuenta del usuario”, implementado en el apartado de “Configuración”.

Se cumple con todas las tecnologías principales propuestas, ya que se emplea Frontend con React y backend Node.js con Express, su base de datos MongoDB y api de OpenAI para sus funcionalidades principales.

Respecto a la metodología descrita se emplea en el producto actual realizando la metodología iterativa incremental realizado en el Trello y descrito en el capítulo anterior.

Capítulo 6. Desarrollo

En este apartado nos enfocaremos en el diseño e implementación de las funcionalidades de la aplicación web. Se mostrarán diagramas de casos de uso de las funcionalidades principales y un análisis de las historias de usuarios cumplidas.

6.1 Análisis del software

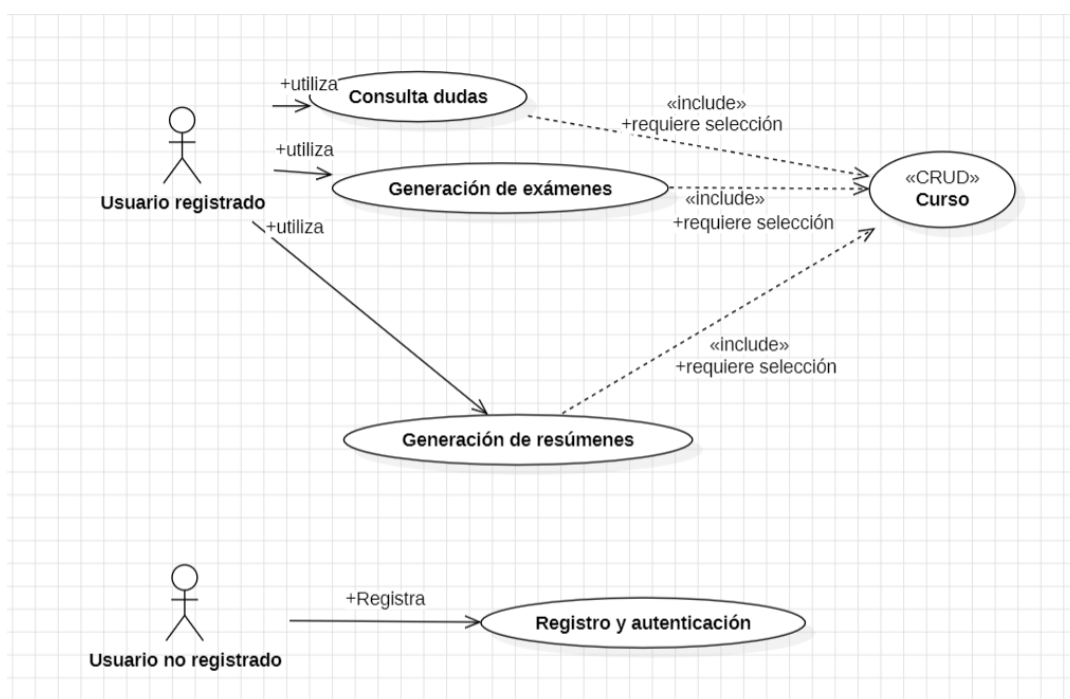


Ilustración 6-1: Diagrama de Casos de Uso de las funcionalidades principales del proyecto

Este diagrama de casos de uso representa las principales interacciones de los usuarios con el sistema web, organizado en torno a sus funcionalidades clave.

Los actores principales y sus relaciones con los casos de uso son los siguientes:

- Usuario no registrado: No tiene acceso a ninguna funcionalidad, para ello debe estar registrado previamente.
- Usuario registrado: Es el actor principal que interactúa con todas las funcionalidades del sistema. Puede consultar dudas, generar exámenes y resúmenes, y gestionar cursos a través de operaciones de CRUD.

A continuación, se muestra una serie de tablas que describen los principales casos de uso del sistema. Cada tabla detalla las funcionalidades disponibles para los usuarios registrados y no registrados, así como su relación con el módulo de gestión de cursos.

Elemento	Descripción
Descripción	El usuario registrado puede interactuar con el asistente de inteligencia artificial para resolver dudas relacionadas con un curso o tema específico.
Relación con Cursos	Incluye la selección de un curso específico para proporcionar contexto a las consultas.
Relaciones	<<include>> con el caso de uso CRUD de Curso, ya que las consultas requieren información sobre cursos previamente gestionados.

Tabla 6-1: Detalle del caso de uso chatAI

Elemento	Descripción
Descripción	El usuario registrado puede generar exámenes en base a un curso específico.
Relación con Cursos	<<include>> con CRUD de Curso, ya que los exámenes están vinculados a los contenidos del curso seleccionado.
Detalles Adicionales	El usuario debe seleccionar un curso antes de generar un examen.

Tabla 6-2: Detalle del caso de uso generación de exámenes

Elemento	Descripción
Descripción	Permite al usuario registrado generar resúmenes basados en el material proporcionado para un curso específico.

Relación con Cursos	<<include>> con CRUD de Curso, ya que requiere información sobre un curso seleccionado.
Extensión	Se extiende para integrar materiales adicionales como documentos o textos asociados al curso.

Tabla 6-3: Detalle del caso de uso generación de resúmenes

Elemento	Descripción
Descripción	Permite al usuario registrado realizar operaciones de Crear, Leer, Actualizar y Eliminar cursos. Es la funcionalidad base del sistema.
Relación con Otros Casos de Uso	<<include>> desde Consulta de Dudas, Generación de Exámenes y Generación de Resúmenes, ya que todos requieren acceso a la información del curso.

Tabla 6-4: Detalle del caso de uso CRUD de cursos

6.2 Diseño interfaz de usuario

El diseño de la interfaz del sistema se ha basado en principios fundamentales que buscan mejorar la experiencia del usuario. Se ha priorizado la simplicidad y el minimalismo mediante un diseño limpio, con colores neutros y sin elementos visuales innecesarios que puedan distraer. La consistencia está presente en todos los elementos de la interfaz, como botones, menús y estilos de texto, permitiendo una navegación fluida y sin confusiones. Además, se ha prestado especial atención en la accesibilidad, utilizando textos claros, tamaños de fuente ajustables y etiquetas en todos los campos para facilitar el uso del sistema a todos los usuarios.

Registrarse

Nombre de Usuario

Foto de Perfil

No se ha sele...ingún archivo

Email

Contraseña

Registrarse

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

Ilustración 6-2: Registro de usuario

Iniciar Sesión

Email

Contraseña

Entrar

[¿No tienes cuenta? Regístrate aquí](#)

Ilustración 6-3: Inicio de sesión

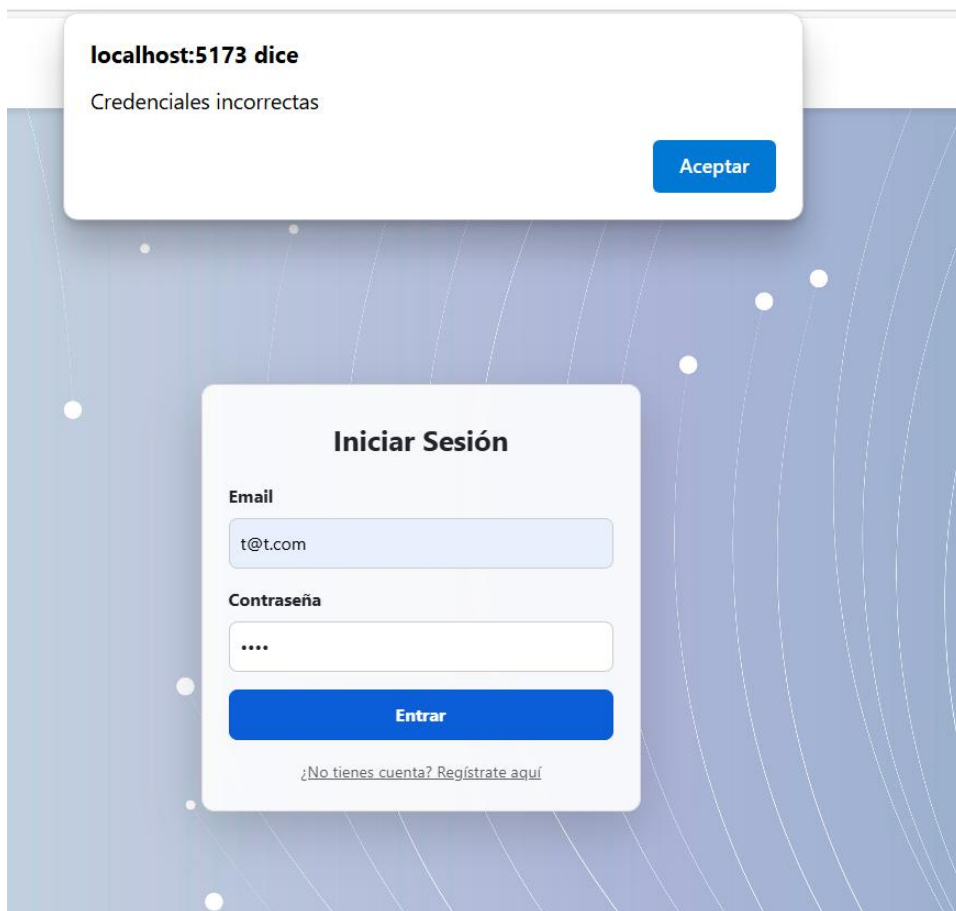


Ilustración 6-4: Inicio de sesión con credenciales incorrectas

Crear Nueva Asignatura

Nombre de la Asignatura

Descripción

Subir Material

No se ha seleccionado ningún archivo

© 2025 Asistente Universitario. Todos los derechos reservados.

Ilustración 6-5: Creación nueva asignatura

Bienvenido a tu Dashboard

Gestiona tus asignaturas y maximiza tu aprendizaje con IA.

Crear Nueva Asignatura

Añade un nombre y una descripción para empezar.

[Crear Asignatura](#)

Cómo Empezar

1. Crea una asignatura en ["Crear Asignatura"](#).
2. Sube material como apuntes o PDFs.
3. Genera resúmenes, chatea con IA o crea exámenes.

Tus Asignaturas

Gestiona tus asignaturas desde aquí:

Sistemas Operativos 1	Ver detalles
Economía Básica	Ver detalles
Economía 2	Ver detalles
economía7	Ver detalles
economía9	Ver detalles

Ilustración 6-6: Muestra del Dashboard

Detalles de la Asignatura

ASO

apuntes

Material Subido

1739990876722-ASO-01-03 La figura del administrador del sistema.pdf	Descargar	Eliminar
1739990876722-ASO-01-04 FHS, organización de directorios en Linux.pdf	Descargar	Eliminar
1739990876729-ASO-01-05 Filesystem Hierarchy Standard v3.0.pdf	Descargar	Eliminar
1739990876732-ASO-01-06 Manejo de la configuración del sistema, servicio systemd.pdf	Descargar	Eliminar

[Editar](#)[Eliminar](#)[Volver](#)

Ilustración 6-7: Contenido de la asignatura

Chat para Resolver Dudas

Selecciona una asignatura

Escribe tu pregunta...

Enviar

© 2025 Asistente Universitario. Todos los derechos reservados.

Ilustración 6-8: Muestra diseño "Chat para Resolver Dudas"

¿El directorio home en Linux es obligatorio?

No, el directorio /home en Linux no es obligatorio. Según el material proporcionado, /home es un directorio donde se encuentran las carpetas de inicio de los distintos usuarios y es opcional.

Dime un directorio opcional en Linux

El directorio opcional en Linux mencionado en el material proporcionado es `/home`.

Dime un directorio obligatorio en Linux

Un directorio obligatorio en Linux es `/bin`.

Di 3 directorios de Linux

1. `/bin`: Directorio que contiene binarios esenciales para el funcionamiento del sistema.
2. `/etc`: Contiene los archivos de configuración locales del sistema.
3. `/usr`: Directorio para órdenes y software no esenciales.

Di un directorio en Linux que no sea `/bin`, `/etc`, `/usr`

Un directorio en Linux que no sea `/bin`, `/etc`, `/usr` es `/var`.

Escribe tu pregunta...

Enviar

© 2025 Asistente Universitario. Todos los derechos reservados.

Ilustración 6-9: Dialogo de "Chat para Resolver Dudas"

Generador de Resúmenes

Selecciona una asignatura

▼

Escribe un tema...

Palabras: 100

●

Generar

Resume sobre usuario administrador

El administrador del sistema operativo es una figura clave en la gestión y el mantenimiento de los sistemas informáticos. Sus principales responsabilidades incluyen la administración de usuarios, hardware y software, así como asegurar el funcionamiento eficiente y seguro del sistema. Este rol exige una planificación metódica de todas las acciones y la capacidad de revertir cambios si es necesario. Además, el administrador debe interactuar efectivamente con los usuarios, manteniendo una comunicación clara y respetuosa, e informando sobre cualquier incidencia o mantenimiento programado. Los privilegios del administrador incluyen establecer normas que garanticen la confidencialidad, integridad y disponibilidad del sistema, facilitando su uso incluso para personas con discapacidades.

Resume sobre comando systemctl

El comando 'systemctl' es una herramienta esencial para la gestión de servicios en sistemas operativos que utilizan 'systemd' como su sistema de inicio. Permite realizar diversas acciones sobre los servicios, como iniciar, detener, reiniciar, recargar y mostrar su estado. Además, 'systemctl' permite habilitar o deshabilitar servicios para que se inicien automáticamente en el arranque del sistema. También se utiliza para gestionar "targets", que son grupos de unidades que configuran un modo específico de operación del sistema. Por ejemplo, se puede cambiar al modo gráfico usando 'systemctl isolate graphical.target'. Este comando es fundamental para la administración y configuración eficiente del sistema, facilitando la gestión de los servicios y la observación de su comportamiento en tiempo real.

Escribe un tema...

Palabras: 100

Generar

© 2025 Asistente Universitario. Todos los derechos reservados.

 **Generador de Exámenes**

Asignatura:
ASO

Tipo de Examen:
☒ Opción Múltiple ☐ Preguntas Abiertas

Nº de Preguntas:
4

Generar Examen

1. ¿Cuáles son las responsabilidades clave de un administrador de sistema operativo para asegurar un sistema eficiente y seguro?

☐ A) Optimizar los juegos en línea y las aplicaciones multimedia

☐ B) Asegurar la confidencialidad, integridad y disponibilidad del sistema

☐ C) El desarrollo de software y pruebas de usuario

☐ D) Promocionar los productos de software

2. ¿Cuál de las siguientes es una tarea básica del administrador del sistema operativo?

☐ A) Escribir código para nuevas aplicaciones

☐ B) Diseñar interfaz gráfica para usuarios

☐ C) Administrar el hardware del sistema

☐ D) Gestionar campañas de marketing digital

3. ¿Qué principio de actuación es crucial en la relación entre el administrador y los usuarios del sistema?

☐ A) Competencia

☐ B) Ambición personal

☐ C) Espíritu de servicio y cooperación al usuario

☐ D) Independencia

4. Según el FHS, ¿cuál de los siguientes directorios es compartible y variable?

☐ A) /usr

☐ B) /var/mail

☐ C) /etc

☐ D) /boot

Corregir Examen

Guardar Examen

Ilustración 6-12: Generador de exámenes de opción múltiple

 **Generador de Exámenes**

Asignatura:
ASO

Tipo de Examen:
☐ Opción Múltiple ☒ Preguntas Abiertas

Nº de Preguntas:
4

Generar Examen

1. ¿Cuáles son los tres aspectos clave, identificados como 'CID', que el administrador del sistema debe asegurar en el sistema informático?

Escribe tu respuesta aquí...

2. Describe una tarea fundamental del administrador del sistema relacionada con el software.

Escribe tu respuesta aquí...

3. ¿Qué acción debe tomar un administrador de sistema según los principios de la disciplina de trabajo antes de realizar cambios?

Escribe tu respuesta aquí...

4. ¿Qué importancia tiene el directorio '/var' en el contexto del FHS?

Escribe tu respuesta aquí...

Corregir Examen

Guardar Examen

Ilustración 6-13: Generador de exámenes de preguntas abiertas

Generador de Exámenes

Asignatura:

ASO
▼

Tipo de Examen:
☒ Opción Múltiple ☐ Preguntas Abiertas

Nº de Preguntas:

4

Generar Examen

1. ¿Cuáles son las responsabilidades clave de un administrador de sistema operativo para asegurar un sistema eficiente y seguro?

☐ A) Optimizar los juegos en línea y las aplicaciones multimedia
☒ B) Asegurar la confidencialidad, integridad y disponibilidad del sistema
☐ C) El desarrollo de software y pruebas de usuario
☐ D) Promocionar los productos de software

✔ **Puntuación:** 100/100
💡 **Comentario:** ✔ Correcto.

2. ¿Cuál de las siguientes es una tarea básica del administrador del sistema operativo?

☐ A) Escribir código para nuevas aplicaciones
☐ B) Diseñar interfaz gráfica para usuarios
☒ C) Administrar el hardware del sistema
☐ D) Gestionar campañas de marketing digital

✔ **Puntuación:** 100/100
💡 **Comentario:** ✔ Correcto.

3. ¿Qué principio de actuación es crucial en la relación entre el administrador y los usuarios del sistema?

☐ A) Competencia
☐ B) Ambición personal
☐ C) Espíritu de servicio y cooperación al usuario
☒ D) Independencia

✔ **Puntuación:** 0/100
💡 **Comentario:** ✖ Incorrecto. La respuesta correcta es: "C) Espíritu de servicio y cooperación al usuario"

4. Según el FHS, ¿cuál de los siguientes directorios es compartible y variable?

☒ A) /usr
☐ B) /var/mail
☐ C) /etc
☐ D) /boot

✔ **Puntuación:** 0/100
💡 **Comentario:** ✖ Incorrecto. La respuesta correcta es: "B) /var/mail"

Calificación Final: 50.00/100

Ilustración 6-14: Corrección examen de opción múltiple

Generador de Exámenes

Asignatura:

ASO

Tipo de Examen:

Opción Múltiple

Preguntas Abiertas

Nº de Preguntas:

4

Generar Examen

1. ¿Cuáles son los tres aspectos clave, identificados como 'CID', que el administrador del sistema debe asegurar en el sistema informático?

Confidencialidad, Integridad y Disponibilidad.

✓

Puntuación: 100/100

Comentario:

Tu respuesta es precisa y correcta, ya que has identificado correctamente los tres aspectos clave 'CID' que son Confidencialidad, Integridad y Disponibilidad. Estos son fundamentales para la seguridad de los sistemas informáticos y es importante que los comprendas bien. ¡Buen trabajo en destacar estos elementos esenciales!

2. Describe una tarea fundamental del administrador del sistema relacionada con el software.

Los cambios se deben realizar de forma incremental.

✓

Puntuación: 40/100

Comentario:

Tu respuesta toca un aspecto importante sobre cómo se deben gestionar los cambios en el software, lo cual es relevante. Sin embargo, sería ideal que especificaras más sobre qué implica la administración del software, como la instalación, actualización, configuración y mantenimiento de los programas. Intenta ampliar tu respuesta para abarcar más claramente las responsabilidades del administrador del sistema en relación con el software.

3. ¿Qué acción debe tomar un administrador de sistema según los principios de la disciplina de trabajo antes de realizar cambios?

Probar, probar y probar antes de actuar.

✓

Puntuación: 40/100

Comentario:

Tu respuesta destaca la importancia de probar antes de implementar cambios, lo cual es crucial en la administración de sistemas para evitar problemas inesperados. Sin embargo, es importante recordar que la planificación previa es un paso fundamental que incluye no solo pruebas, sino también la evaluación de impactos, la preparación de procedimientos de reversión y la comunicación con los usuarios afectados. Te animo a considerar todos estos aspectos en futuras respuestas para abarcar completamente los principios de la disciplina de trabajo en la administración de sistemas.

4. ¿Qué importancia tiene el directorio 'var' en el contexto del FHS?

Contiene subdirectorios y archivos de datos que utilizan aplicaciones.

✓

Puntuación: 50/100

Comentario:

Tu respuesta identifica correctamente que 'var' contiene datos utilizados por aplicaciones, lo cual es un buen comienzo. Sin embargo, sería ideal profundizar un poco más explicando que estos datos son variables, es decir, que cambian o se actualizan con frecuencia, como las colas de impresión y los logs del sistema. Intenta incluir estos detalles específicos en futuras respuestas para proporcionar una comprensión más completa del tema.

Calificación Final: 57.50/100

Ilustración 6-15: Corrección examen de preguntas abiertas

Desarrollo

51

Asignatura	Tipo	Preguntas	Estado	Acción
ASO	test	1	En progreso	Repetir Examen
ASO	test	2	En progreso	Repetir Examen
ASO	test	1	En progreso	Repetir Examen
ASO	test	1	En progreso	Repetir Examen
Matemáticas básicas	test	1	En progreso	Repetir Examen
sistema operativo 3	test	1	En progreso	Repetir Examen
ASO	test	3	En progreso	Repetir Examen
ASO	test	1	En progreso	Repetir Examen
ASO	test	1	En progreso	Repetir Examen
ASO	test	4	En progreso	Repetir Examen
ASO	open	1	En progreso	Repetir Examen
ASO	test	1	En progreso	Repetir Examen
ASO	test	2	En progreso	Repetir Examen

Ilustración 6-16: Muestra del "Catálogo de Exámenes"

Generador de Exámenes

Asignatura:

ASO

Tipo de Examen:

Opción Múltiple

Preguntas Abiertas

Nº de Preguntas:

5

Generar Examen

1. ¿Cuáles son los tres aspectos clave que un administrador del sistema debe garantizar en el sistema operativo según el contenido del curso?

A) Eficiencia, Accesibilidad, Usabilidad

B) Confidencialidad, Integridad, Disponibilidad

C) Seguridad, Flexibilidad, Rendimiento

D) Control, Monitorización, Regulación

2. ¿Qué principio NO se menciona como parte del protocolo de comunicación entre el administrador del sistema y los usuarios?

A) Informar de cambios planificados

B) Asegurar la transparencia total de los procesos

C) Responder siempre a las consultas de los usuarios

D) Mantener un estilo de comunicación respetuoso y adaptado a la cultura

Corregir Examen

Ilustración 6-17: Repetición de examen del "Catálogo de Exámenes"

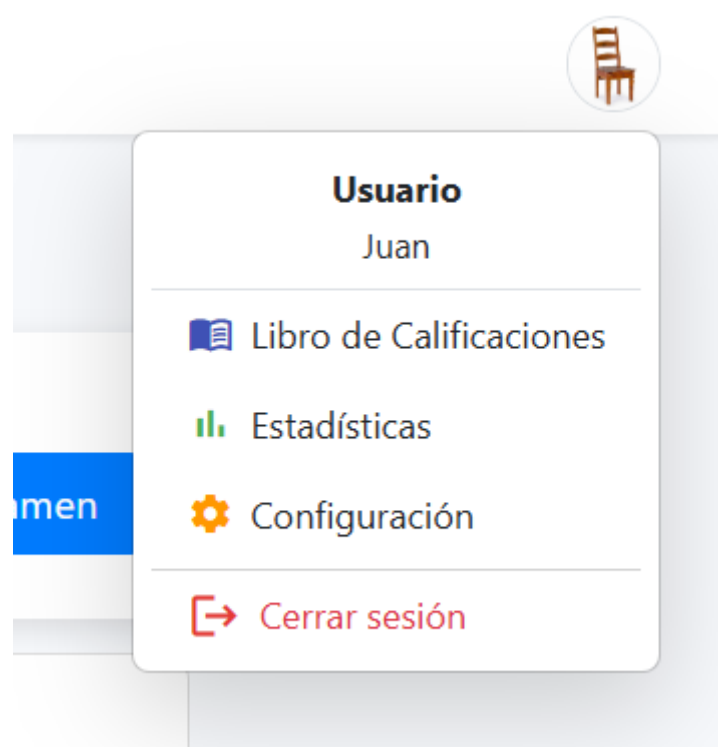


Ilustración 6-18: Secciones del perfil de usuario

StudyUniv			
Chat AI			
Generador de Exámenes			
Catálogo de Exámenes			
Generador de Resúmenes			
Libro de Calificaciones			
Asignatura	Tipo de Examen	Calificación	Acción
sistema operativo 3	test	66.67/100	Ver Corrección
	open	30/100	Ver Corrección
	open	31.67/100	Ver Corrección
	open	25/100	Ver Corrección
	open	28.33/100	Ver Corrección
	open	28.33/100	Ver Corrección
	open	28.33/100	Ver Corrección
ASO	open	3.33/100	Ver Corrección
	open	6.67/100	Ver Corrección
	open	3.33/100	Ver Corrección
	test	0/100	Ver Corrección

Ilustración 6-19: Libro de calificaciones

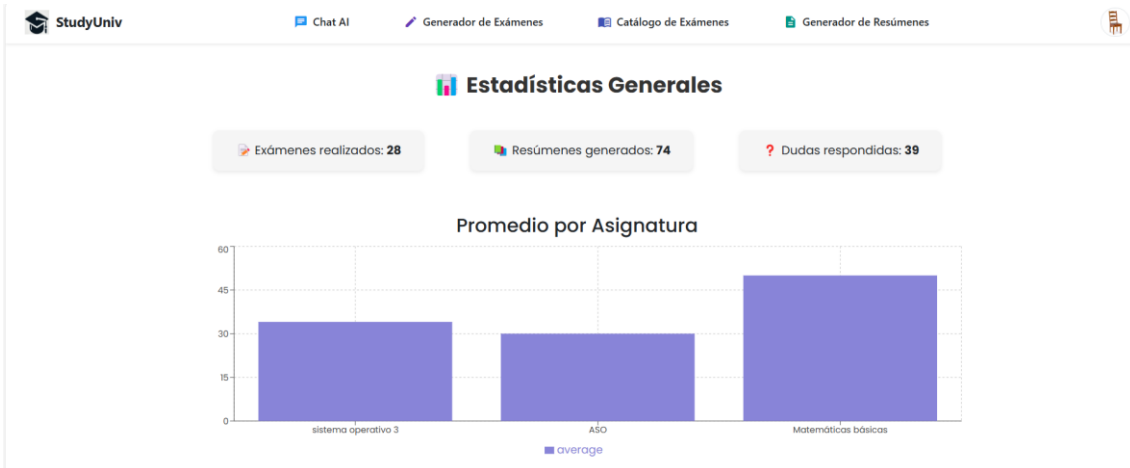


Ilustración 6-20: Diagrama de barras

Distribución de Tipos de Examen

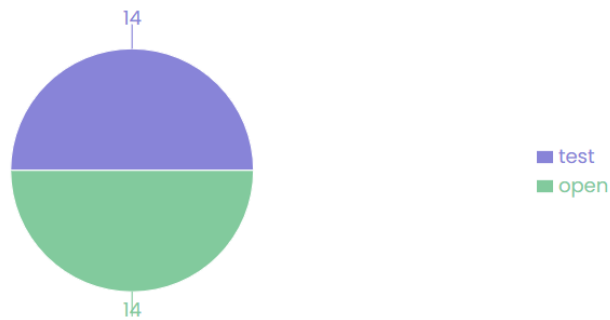


Ilustración 6-21: Diagrama de Sector

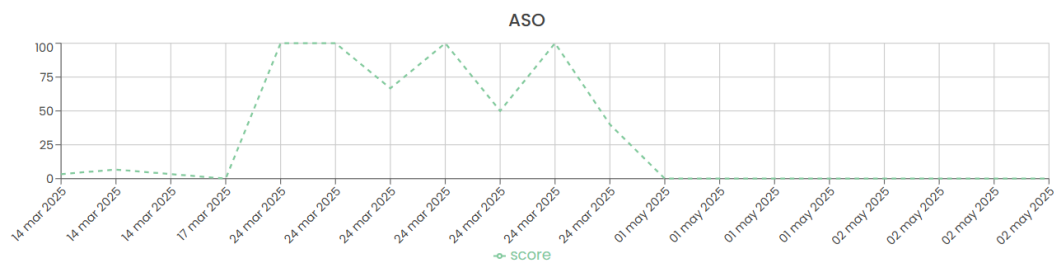


Ilustración 6-22: Gráfico de líneas calificaciones por asignatura a lo largo del tiempo



Ilustración 6-23: Gráfico de líneas calificaciones a lo largo del tiempo

6.2.1 Diseño adaptativo

El diseño adaptativo permite que las aplicaciones web se ajusten correctamente a diferentes tamaños de pantalla, mejorando la experiencia del usuario tanto en dispositivos de escritorio como móviles. La aplicación web se adapta a escritorios, tabletas y móviles.

El módulo de exámenes reorganiza los controles de generación en una pila, con cada selector ocupando toda la anchura disponible. Esta disposición permite una interacción más cómoda en pantallas táctiles. Además, el contenido del examen se presenta en tarjetas verticales que mantienen la legibilidad y navegabilidad.

Generador de Exámenes

Asignatura:
Selecciona una asignatura

Tipo de Examen:
☒ Opción Múltiple
 ☐ Preguntas Abiertas

Nº de Preguntas:
5

Generar Examen

No hay preguntas generadas aún.

Tipo de Examen:
☐ Opción Múltiple
 ☒ Preguntas Abiertas

Nº de Preguntas:
1

Generar Examen

1. ¿Cuál de las siguientes es una responsabilidad principal del administrador del sistema operativo para asegurar que el sistema informático sea eficiente y seguro?

Escribe tu respuesta aquí...

Corregir Examen

© 2025 Asistente Universitario. Todos los derechos reservados

Ilustración 6-24: Generador de examen en dispositivo móvil

Como se puede apreciar en la “Ilustración 6-24” la disposición vertical facilita la experiencia de usuario. Los elementos del formulario, como la selección de asignatura, el tipo de examen (opción múltiple o preguntas abiertas) y el número de preguntas, se encuentran apilados uno debajo del otro, maximizando el uso del espacio disponible. Esta distribución es fundamental en entornos táctiles, donde cada elemento debe estar lo suficientemente separado para evitar errores en la interacción.

Una vez generado el examen, la pregunta se muestra en un bloque definido, el botón "Corregir Examen" aparece claramente visible debajo en otro bloque, conservando un tamaño óptimo para la interacción táctil, con suficiente espacio alrededor para evitar toques accidentales.

The image shows a mobile application interface for an exam. It consists of two vertical blocks of questions, each with a title and four radio button options. Below the questions is a green button labeled 'Corregir Examen'. At the bottom right, there is a small copyright notice: '© 2025 Asistente Universitario. Todos los derechos reservados.'

1. ¿Cuál de las siguientes es una responsabilidad del administrador del sistema operativo?

- ☐ A) Crear contenido web
- ☐ B) Establecer procedimientos y normas para la eficiencia y seguridad del sistema
- ☐ C) Diseñar gráficos para interfaces de usuario
- ☐ D) Manejar campañas de marketing digital

2. Una de las tareas básicas del administrador del sistema es:

- ☐ A) Planear actividades recreativas para usuarios
- ☐ B) Administrar los derechos de autor de software
- ☐ C) Comprobación del buen funcionamiento del sistema
- ☐ D) Diseñar el logotipo de la empresa

3. ¿Qué significa 'CID' en el contexto de las responsabilidades del administrador del sistema operativo?

- ☐ A) Creatividad, Innovación, Determinación
- ☐ B) Confidencialidad, Integridad, Disponibilidad
- ☐ C) Calculo, Implementación, Desarrollo
- ☐ D) Comunicación, Instrucción, Documentación

4. ¿Qué indica la jerarquía FHS en un sistema Unix?

- ☐ A) Fracciones de Hora Solar
- ☐ B) Funciones de Hardware Secundario
- ☐ C) Formato de Hardware Simplificado
- ☐ D) Filesystem Hierarchy Standard

Corregir Examen

© 2025 Asistente Universitario. Todos los derechos reservados.

Ilustración 6-25: Generador de exámenes con múltiples preguntas tipo test en dispositivo móvil

Todas las preguntas se encuentran de forma vertical. Cada bloque de pregunta incluye su enunciado destacado con una tipografía clara y jerárquica, seguido de las opciones de respuesta con un espaciado adecuado que evita la saturación visual.

Las opciones tipo radio se presentan con un tamaño de selección suficientemente grande para garantizar la precisión al tocar, elemento crucial en la experiencia móvil. El diseño asegura también que el usuario se enfoque en una pregunta a la vez, pero pueda desplazarse con fluidez por el resto del examen. El botón para “Corregir Examen” se mantiene al final de la página, siempre accesible y con una presencia destacada para facilitar la acción final del usuario.

De la misma forma para exámenes de preguntas abiertas, como se muestra a continuación:

The image shows a mobile application interface for an exam. It features five questions, each with a text input field for the answer. The questions are:

- 1. ¿Cuáles son los tres aspectos principales que el administrador del sistema operativo debe asegurar en términos de 'CID'?**
Escribe tu respuesta aquí...
- 2. ¿Qué tareas básicas del administrador del sistema fueron destacadas?**
Escribe tu respuesta aquí...
- 3. ¿Cuál es el principio clave en la relación entre el administrador del sistema y los usuarios según el contenido del curso?**
Escribe tu respuesta aquí...
- 4. En la estructura del sistema de ficheros de Unix bajo FHS, ¿qué directorios se mencionan como ejemplos de 'Compartibles'?**
Escribe tu respuesta aquí...
- 5. Según el FHS, ¿qué tipo de directorios son considerados como 'Estáticos'?**
Escribe tu respuesta aquí...

At the bottom of the screen, there is a green button with a white icon of a document and the text "Corregir Examen". Below the button, there is a small copyright notice: "© 2025 Asistente Universitario. Todos los derechos reservados".

Ilustración 6-26: Generador de exámenes con múltiples preguntas abiertas en dispositivo móvil

En el caso de las preguntas abiertas, el diseño responsivo organiza todos los elementos de manera vertical, optimizando la lectura y la interacción en pantallas pequeñas. Cada bloque de pregunta comienza con su enunciado claramente destacado, utilizando una tipografía jerarquizada que facilita la comprensión inmediata del contenido. Esta estructura permite que se centre en una única

pregunta a la vez, lo cual es especialmente útil en dispositivos móviles donde el espacio visual es limitado.

Debajo del enunciado, se presenta un área de texto ampliada para que el estudiante escriba su respuesta. Esta área cuenta con un tamaño adecuado y márgenes bien definidos, lo que no solo mejora la legibilidad del texto ingresado, sino que también ofrece suficiente espacio táctil para facilitar la escritura. El campo de respuesta se adapta automáticamente al tamaño de la pantalla, evitando cualquier necesidad de desplazamiento horizontal.

Además, el botón “Corregir Examen” se encuentra al final de la interfaz, claramente visible y con un tamaño accesible, lo cual garantiza una navegación intuitiva hasta la acción final. Su ubicación asegura que el usuario pueda finalizar el proceso sin necesidad de buscar controles ocultos o de interactuar con elementos superpuestos.

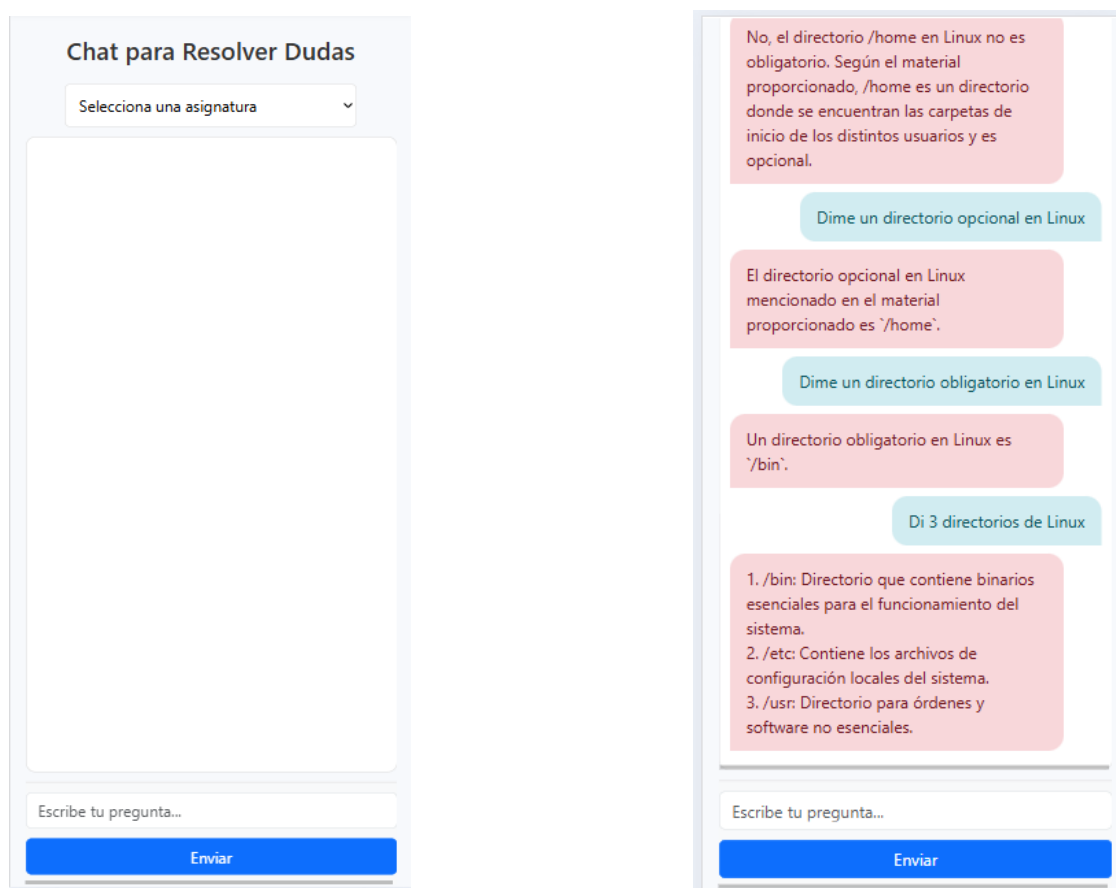


Ilustración 6-27: Chat para resolver dudas en dispositivo móvil

La interfaz se adapta perfectamente a una disposición vertical que favorece la experiencia de conversación. Cada mensaje se muestra en un contenedor claramente delimitado, con colores diferenciados según el emisor: preguntas del

usuario en azul claro y respuestas en color rosa. Esta distinción visual permite seguir fácilmente el flujo del diálogo sin confusión.

Los textos están justificados y cuentan con un tamaño adecuado para su lectura en pantallas pequeñas. Además, el espaciado entre los bloques de mensajes evita la sensación de saturación visual, permitiendo al usuario enfocarse en una respuesta a la vez. El campo para escribir nuevas preguntas permanece fijo en La parte inferior, asegurando su disponibilidad en todo momento y facilitando la interacción continua.

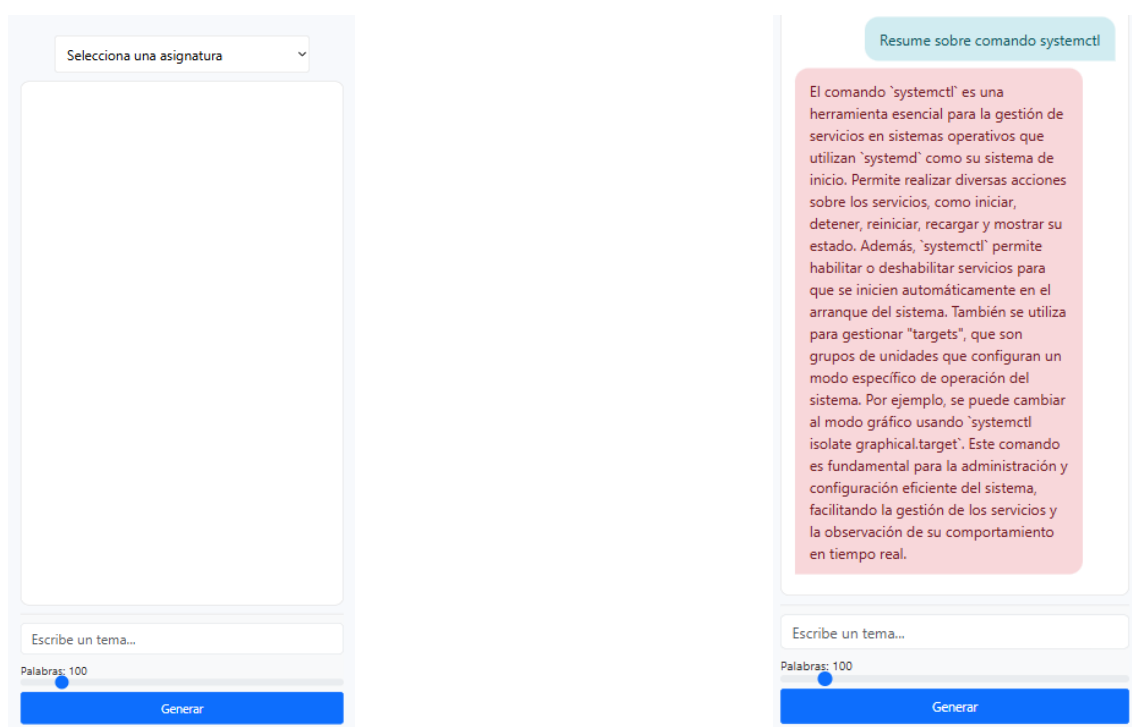


Ilustración 6-28: Generador de resúmenes en dispositivo móvil

El diseño del Generador de Resúmenes en dispositivos móviles ofrece una disposición limpia y funcional. La entrada del tema a resumir se encuentra en la parte inferior de la vista, seguida del control deslizante para seleccionar la longitud del resumen en palabras, y finalmente, el botón para generar el contenido. Esta secuencia vertical facilita una navegación natural y progresiva desde la entrada hasta la acción.

Una vez generado, el resumen aparece en la parte superior del área de contenido, en un bloque de texto con buen contraste y justificación que favorece la legibilidad. El diseño prioriza un enfoque de lectura fluido y sin distracciones,

permitiendo al usuario revisar la información con claridad. Además, el botón "Generar" se mantiene visible y accesible al final de la interfaz, incentivando la reutilización inmediata de la herramienta sin recargar la pantalla.

6.3 Cursos de formación

Antes de comenzar con el desarrollo del proyecto, dediqué tiempo a formarme en las tecnologías necesarias, ya que muchas de ellas no las había utilizado en profundidad anteriormente. Me enfoqué principalmente en el stack MERN (MongoDB, Express, React, Node.js) y en la integración de servicios de inteligencia artificial, con el objetivo de adquirir las competencias técnicas necesarias para llevar a cabo una aplicación funcional, escalable y moderna.

Una de las principales fuentes de aprendizaje fue el tutorial "Build and Deploy a Full Stack MERN AI Image Generation App | Midjourney & DALL-E Clone" [38], disponible en YouTube. Este recurso me permitió comprender cómo estructurar una aplicación completa con MERN, y fue clave para familiarizarme con la lógica de conexión entre frontend y backend, así como la integración de modelos de IA en tiempo real.

En el desarrollo del frontend, aprendí a utilizar React.js de forma práctica. Algunos conceptos que apliqué directamente en el proyecto incluyen:

- Componentización: Organicé la interfaz dividiendo las funcionalidades en componentes reutilizables, lo que facilitó la escalabilidad del código y su mantenimiento.
- Gestión de estados: A través de `useState` y `useEffect`, gestioné dinámicamente la interacción con el usuario y la actualización de los contenidos, como las secciones de resúmenes, exámenes o dudas.
- Estilización avanzada: Aunque el tutorial proporcionaba una base de diseño, lo complementé con el uso de Bootstrap, mejorando así la presentación visual y la experiencia de usuario.

Además, seguí el curso gratuito "React - The Complete Guide" de Academind (Maximilian Schwarzmüller) [39] a través de YouTube. Este curso fue especialmente útil para entender con más profundidad el ciclo de vida de los componentes, la estructura de carpetas, la creación de rutas privadas y públicas, y el manejo de props entre componentes.

Para el backend, aprendí a desarrollar un servidor funcional utilizando Node.js y Express, siguiendo una arquitectura modular inspirada en los ejemplos del tutorial ya mencionado anteriormente sobre Full Stack MERN. Entre los conceptos aplicados destacan:

- Definición de rutas RESTful: Implementé rutas como `/summary`, `/exam` y `/chat`, siguiendo el modelo de diseño orientado a recursos.
- Middlewares personalizados:

- Middleware de validación de datos: Asegura que las solicitudes entrantes cumplen con los formatos esperados.
- Middleware de autenticación con JWT: Restringe el acceso a funcionalidades sensibles únicamente a usuarios autenticados.
- Gestión de errores: Incorporé control de errores globales para asegurar respuestas claras en caso de fallos.

Para asegurar la protección de datos sensibles, aprendí a gestionar variables de entorno mediante archivos `.env`, evitando así la exposición de claves privadas (como la API Key de OpenAI). Este enfoque lo aprendí tanto del tutorial como de la documentación oficial de Express y dotenv, que consulté durante el proceso.

Complementariamente, para el uso de MongoDB, consulté el manual oficial de MongoDB University [40] para aprender a estructurar consultas, gestionar datos con eficiencia y aplicar buenas prácticas en la integración de bases de datos dentro de aplicaciones MERN.

6.4 Diseño de la base de datos

El diseño de la base de datos para este proyecto se ha realizado con un enfoque relacional y estructurado, utilizando MongoDB como sistema de gestión de base de datos no relacional (NoSQL). A continuación, se detalla su justificación:

MongoDB permite trabajar con modelos de documentos que son similares a las tablas relacionales, pero sin la necesidad de un esquema rígido lo que facilita iterar rápidamente durante el desarrollo. Además, MongoDB ofrece una integración nativa con entornos que emplean JavaScript, como Node.js. La biblioteca Mongoose facilita la creación de un mapeo directo entre los objetos definidos y los documentos almacenados en la base de datos, que se denomina Object Data Modeling (ODM) [41].

Por lo tanto, este diseño busca garantizar eficiencia, escalabilidad y flexibilidad al almacenar y recuperar los datos necesarios para las funcionalidades de la web. A continuación, se describen los principales aspectos del diseño de la base de datos.

6.4.1 Objetivos del diseño

- Escalabilidad: Asegurar que la base de datos soporte el crecimiento en el número de usuarios y los datos asociados (asignaturas, materiales, resúmenes, etc.).
- Eficiencia: Diseñar las relaciones entre colecciones para minimizar redundancias y optimizar las consultas.
- Integridad de datos: Garantizar que los datos almacenados sean consistentes y válidos.
- Flexibilidad: Permitir la integración de nuevas funcionalidades sin reestructurar completamente la base de datos.

6.4.2 Entidades principales

El sistema está compuesto por varias entidades clave que representan los elementos funcionales principales:

- Usuarios (users)

Propósito: Almacenar información de los usuarios registrados y no registrados.

Atributos principales:

Atributos	Significado
_id	Identificador único generado automáticamente por MongoDB.
username	Nombre del usuario.
email	Dirección de correo electrónico.
password	Contraseña encriptada con bcrypt. No es legible, pero se puede comparar para validar login.
profilePicture	URL de la imagen almacenada.
__v	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-5: Atributos "users"

users	<pre>_id: ObjectId('681610c278ba528ab00f4bf0') username: "kiko" email: "pedros@pedros.com" password: "\$2b\$10\$25Ud6v04KtddMRQAYtZ0QuQv8wDKMrgrsWHPFqX/ow9YYpvnJwJ2" profilePicture: "data:image/jpeg;base64,/9j/4QAYRXhpZgAASUkqAAgAAAAAAAAAAAAAAP/sABFEdWN..." __v: 0</pre>
-------	--

Ilustración 6-29: Muestra de documentos de la colección "users" de la base de datos

Relaciones: Asociado con asignaturas, resúmenes y exámenes.

- Asignaturas (courses)

Propósito: Representar las asignaturas y cursos disponibles en el sistema.

Atributos principales:

Atributos	Significado
id	Identificador único generado automáticamente por MongoDB.
name	Referencia al _id del usuario que creó el curso. Permite saber a quién pertenece.
description	Descripción general del curso.
materials	Array de strings con los nombres de los archivos PDF subidos.

extractedText	Texto extraído automáticamente del contenido de los archivos PDF, para todas las funcionalidades de la web(resúmenes,exámenes,dudas).
createdAt	Fecha y hora en que se creó el curso. (formato ISO 8601)
updatedAt	Fecha de la última vez que se actualizó el curso. (formato ISO 8601)
__v	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-6: Atributos "courses"

courses	Filter 	Type a query: { field: 'value' }	Reset	Apply	Options 
exams					
scores					
summaries					
users					
	<pre> _id: ObjectId('67d9cdec64558290b0def5db') user : ObjectId('67d6dda88f99d67e33072cef') name : "Administración Sistemas Operativos" description : "Conocimientos fundamentales." ▼ materials : Array (5) 0: "1742327274952-ASO-01-00 Objetivos de aprendizaje del tema 1.pdf" 1: "1742327274968-ASO-01-01 Trabajar con GNU-Linux.pdf" 2: "1742327274994-ASO-01-02 Chuleta de Ãrdenes de UNIX-Linux.pdf" 3: "1742327274994-ASO-01-03 La figura del administrador del sistema.pdf" 4: "1742327274996-ASO-01-04 FHS, organizaciÃn de directorios en Linux.pdf" extractedText : "Administraci3n de Sistemas Operativos Tema 1: objetivos de aprendizaje." createdAt : 2025-03-18T19:47:56.607+00:00 updatedAt : 2025-03-24T11:56:55.778+00:00 __v : 0 </pre>				

Ilustraci3n 6-30: Muestra de documento de la colecci3n "courses" de la base de datos

Relaciones: Asociado con materiales, resúmenes y exámenes.

- Exámenes (exams)

Prop3sito: Almacenar informaci3n sobre los exámenes generados.

Atributos principales:

Atributos	Significado
id	Identificador 3nico generado autom3ticamente por MongoDB.
userId	ID del usuario que gener3 el examen.
courseld	ID del curso al que pertenece el examen.

createdAt	Fecha y hora en que se creó el curso. (formato ISO 8601)
updatedAt	Fecha de la última vez que se actualizó el curso. (formato ISO 8601)
__v	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-7: Atributos "exams"

```
_id: ObjectId('67e1853b5dc89a6a90507cb9')
userId : ObjectId('6758338039879aadb9c5ce30')
courseId : ObjectId('67b6275ea11c3dfd2388d950')
createdAt : 2025-03-24T16:15:56.022+00:00
updatedAt : 2025-03-24T16:15:56.022+00:00
__v : 0
```

Ilustración 6-31: Muestra de documento de la colección “exams” de la base de datos

Relaciones: Relacionado con preguntas y cursos.

- Resúmenes (summaries)

Propósito: Almacenar los resúmenes generados para los cursos.

Atributos principales:

Atributos	Significado
_id	Identificador único generado automáticamente por MongoDB.
userId	ID del usuario que generó el resumen.
courseId	ID del curso al que pertenece el resumen.
topic	Tema del resumen escrito por el usuario.
summary	Texto del resumen generado.
createdAt	Fecha de generación del resumen.
__v:	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-8: Atributos "summaries"



Ilustración 6-32: Muestra de documento de la colección “summaries” de la base de datos

Relaciones: Asociado con cursos.

- Puntuaciones obtenidas de los exámenes (scores)

Propósito: Almacenar las calificaciones obtenidas sobre los exámenes corregidos.

Atributos	Significado
_id	Identificador único generado automáticamente por MongoDB.
userId	ID del usuario que se califica el examen.
topic	Curso o asignatura al que pertenece el examen evaluado.
score	Puntuación final obtenida del examen.
examType	Tipo de examen (Opción múltiple o preguntas abiertas)
questions	Preguntas generadas para el examen
options	Opciones del examen en caso de examen de opción múltiple.
correctAnswers	Opción correcta.
createdAt	Fecha de generación de preguntas del examen.
__v	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-9: Atributos modelo "scores"

gradingResults: Se encuentra como principales apartados:

Atributos	Significado
question	Pregunta del examen
userAnswers	Respuesta del usuario.
correctAnswers	Opción correcta
score	Puntuación obtenida de cada pregunta.
feedback	Indicador si es correcto o incorrecto la respuesta proporcionada.

Tabla 6-10: Atributos "gradingResults"

exams

scores

summaries

users

type a query: { field: value }

```
_id: ObjectId('67e18466914938e97ce3b19b')
userId: ObjectId('6758338039879aadb9c5ce30')
topic: ObjectId('67b6275e11c3dfd2388d950')
score: 100
examType: "test"
questions: Array (1)
responses: Array (1)
gradingResults: Array (1)
createdAt: 2025-03-24T16:12:22.294+00:00
__v: 0
```

```
_id: ObjectId('67e1854c5dc89a6a90507cbd')
userId: ObjectId('6758338039879aadb9c5ce30')
topic: ObjectId('67b6275e11c3dfd2388d950')
score: 100
examType: "test"
questions: Array (1)
  0: Object
    question: "¿Cuál de las siguientes tareas NO es comúnmente asociada a las respons..."
    options: Array (4)
      0: "A) Administrar usuarios"
      1: "B) Mantener la documentación del sistema"
      2: "C) Diseñar el logo de la empresa"
      3: "D) Seguridad del sistema"
    correctAnswer: "C) Diseñar el logo de la empresa"
    _id: ObjectId('67e1854c5dc89a6a90507cbe')
  responses: Array (1)
    0: "C) Diseñar el logo de la empresa"
  gradingResults: Array (1)
    0: Object
      question: "¿Cuál de las siguientes tareas NO es comúnmente asociada a las respons..."
      userAnswer: "C) Diseñar el logo de la empresa"
      correctAnswer: "C) Diseñar el logo de la empresa"
      score: 100
      feedback: "✅ Correcto."
      _id: ObjectId('67e1854c5dc89a6a90507cbf')
    createdAt: 2025-03-24T16:16:12.211+00:00
    __v: 0
```

PREVIOUS

Ilustración 6-33: Muestra de documento de la colección “scores” de examen de opción múltiple

```

_id: ObjectId('67e18fc25dc89a6a90507d03')
userId: ObjectId('6758338039879aadb9c5ce30')
topic: ObjectId('67b6275ea11c3dfd2388d950')
score: 40
examType: "open"
▶ questions: Array (1)
▼ responses: Array (1)
  0: "Un sistema operativo es el software más importante que se ejecuta en t..."
▼ gradingResults: Array (1)
  0: Object
    question: "Explique la importancia de la figura del administrador del sistema ope..."
    userAnswer: "Un sistema operativo es el software más importante que se ejecuta en t..."
    correctAnswer: "(C) Tiene la responsabilidad de establecer procedimientos para mantener..."
    score: 40
    feedback: "La respuesta proporcionada describe de manera general las funciones de..."
    _id: ObjectId('67e18fc25dc89a6a90507d05')
  createdAt: 2025-03-24T17:00:50.792+00:00
  __v: 0

```

Ilustración 6-34: Muestra de documento de la colección “scores” de preguntas abiertas

Aunque en los exámenes de tipo abierto el usuario escribe una respuesta libre, internamente el sistema mantiene la respuesta esperada (correctAnswer) generada por OpenAI.

Para que el modelo de IA tenga un criterio al momento de comparar la respuesta del usuario con una referencia correcta, y así asignar un score de 0 a 100 y dar un comentario explicativo.

- savedExams

Propósito: Almacenar los exámenes generados por los usuarios sin incluir respuestas ni calificaciones. Estos exámenes se pueden recuperar para ser reutilizados o corregidos posteriormente.

Atributos	Significado
_id	Identificador único generado automáticamente por MongoDB.
userId	ID del usuario que generó y guardó el examen.
courseld	Curso o asignatura al que pertenece el examen.
examType	Tipo de examen (Opción múltiple o preguntas abiertas).
exam	Preguntas generadas para el examen.
question	Texto de la pregunta.
options	Opciones de respuesta si el examen es de tipo test.
correctAnswer	Respuesta correcta esperada para cada pregunta.
examLocked	Indica si el examen fue bloqueado para edición (false por defecto).
finalScore	No se almacena aquí, campo nulo por defecto.
gradingResults	No se utiliza en este contexto. Se deja nulo.

createdAt	Fecha en la que se guardó el examen en la base de datos.
__v	Propiedad de Mongoose usada para manejar versiones.

Tabla 6-11: Atributos modelo "savedExams"

savedexams scores summaries users	<pre> _id: ObjectId('68173e258c08952f5db81f1b') userId: ObjectId('6758338039879aadb9c5ce30') courseId: ObjectId('67b6275ea11c3dfd2388d950') examType: "test" exam: Array (2) 0: Object question: "¿Cuáles son los aspectos clave que el administrador del sistema debe a..." options: Array (4) 0: "A) Compatibilidad, Innovación, Durabilidad" 1: "B) Confidencialidad, Integridad, Disponibilidad" 2: "C) Conectividad, Integración, Desarrollo" 3: "D) Comunicación, Identificación, Documentación" correctAnswer: "B) Confidencialidad, Integridad, Disponibilidad" _id: ObjectId('68173e258c08952f5db81f1c') 1: Object question: "¿Cuál es la función principal de la estructura de directorios /var seg..." options: Array (4) correctAnswer: "C) Alojamiento de datos variables como colas de impresión y correos electrónicos..." _id: ObjectId('68173e258c08952f5db81f1d') createdAt: 2025-05-04T10:15:01.545+00:00 __v: 0 </pre>
---	--

Ilustración 6-35: Muestra de documento de la colección “savedexams” de la base de datos

6.5 Configuración del proyecto

6.5.1 Configuración backend

Se crea el proyecto en Visual Studio Code instalando las bibliotecas y dependencias necesarias, se instala Node.js para manejar el backend npm para gestionar las dependencias del proyecto.

Para el backend se crea una estructura organizada:

- config
 - models
 - chat.model.js
 - course.model.js
 - exam.model.js
 - material.model.js
 - questionModel.js
 - score.model.js
 - user.mode.js
 - db.js
 - multer.js
-
- controllers
 - courseController.js
 - openaiController.js
 - scoreController.js
 - userController.js
- middleware

- authMiddleware.js
- routes
 - courseRoutes.js
 - userRoutes.js
- uploads
- utils
 - pdfUtils.js
- server.js

6.5.2 Carpeta config

Contiene la configuración esencial del proyecto, incluyendo conexiones a la base de datos y herramientas necesarias.

Se encuentra dentro de la carpeta el archivo “db.js” que se encarga de configurar y establecer la conexión con MongoDB utilizando Mongoose. Como ventaja facilita la conexión centralizada en la base de datos y proporciona manejo de errores en caso de problemas de conexión.

Además, se encuentra en la misma carpeta “multer.js” que su función es la configuración para manejar la subida de archivos al servidor. Define el almacenamiento y los nombres de archivos subidos y facilita la gestión de materiales de curso como PDFs.

En la carpeta “models/” se define los esquemas y modelos de datos utilizados en MongoDB. Cada archivo representa una entidad en la base de datos:

Archivos carpeta “models/”	Representación en la base de datos
chat.model.js	Representa las interacciones de chat, almacenando mensajes, preguntas y respuestas.
course.model.js	Define los datos relacionados con los cursos, como el nombre, material asociado y descripciones.
exam.model.js	Modela los datos de los exámenes, incluyendo preguntas, respuestas correctas y detalles del examen.
score.model.js	Maneja las calificaciones de los usuarios, incluyendo el puntaje, tema y tipo de examen.
summary.model.js	Almacena los resúmenes creados para cada curso y usuario.
user.model.js	Define la estructura para almacenar datos de los usuarios, como nombre, correo electrónico, contraseña y roles.

Tabla 6-12: Representación archivos carpeta “models/”

6.5.3 Carpeta controllers

Archivos carpeta "controllers/"	Función
courseController.js	Maneja operaciones relacionadas con los cursos, como crear, actualizar, eliminar y obtener información. Además, proporciona soporte para asociar materiales a los cursos.
openaiControllers.js	Interactúa con la API de OpenAI para funcionalidades de generación de texto. Implementa la lógica para chat, resúmenes y generación de exámenes.
scoreController.js	Gestiona el almacenamiento y recuperación de las calificaciones de los usuarios. Proporciona funcionalidades para guardar y listar los puntajes obtenidos en exámenes.
statisticsController.js	Gestiona la recopilación y presentación de estadísticas del usuario. Proporciona datos sobre calificaciones, resúmenes, tipos de examen y rendimiento por curso.
userController.js	Contiene la lógica para el registro, inicio de sesión y manejo de usuarios. Maneja la creación de tokens JWT para autenticación.

Tabla 6-13: Representación archivos carpeta "controllers/"

6.5.4 Carpeta middleware

Incluye funcionalidades que se ejecutan entre la solicitud del usuario y la respuesta del servidor.

Se encuentra en dicha carpeta el archivo "authMiddleware.js" que se encarga de verificar si las solicitudes están autenticadas mediante tokens JWT. Protege rutas específicas para que solo los usuarios autorizados puedan acceder.

```
import jwt from 'jsonwebtoken';
import User from '../config/models/user.model.js';

//garantizar solo usuarios autenticados

const protect = async (req, res, next) => {
  const authHeader = req.headers.authorization;

  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    return res.status(401).json({ message: 'No autorizado, falta el token' });
  }

  const token = authHeader.split(' ')[1];

  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET);
    req.user = await User.findById(decoded.id).select('-password');
    if (!req.user) {
      return res.status(401).json({ message: 'Usuario no encontrado' });
    }
    next();
  } catch (error) {
    console.error('Error al validar el token:', error.message);
    res.status(401).json({ message: 'Token inválido o expirado' });
  }
};

export default protect;
```

Ilustración 6-36: Código archivo "authMiddleware.js"

Implementa un middleware llamado protect, cuya función es garantizar que solo los usuarios autenticados puedan acceder a determinadas rutas protegidas del backend, valida la identidad del usuario a través de un token JWT (JSON Web Token).

Cuando un usuario realiza una solicitud a una ruta protegida, el middleware verifica primero que exista un encabezado de autorización ("Authorization") en la petición. Este encabezado debe seguir el formato estándar "Bearer <token>". Si el encabezado está ausente o mal formado, el middleware detiene la ejecución y devuelve una respuesta de error 401 indicando que el token es obligatorio.

Si el token está presente, se extrae de la cabecera y se valida utilizando la clave secreta definida en las variables de entorno del servidor (JWT_SECRET). Este proceso permite decodificar el token y obtener el identificador del usuario. A continuación, se realiza una búsqueda en la base de datos para confirmar que el usuario realmente existe, excluyendo el campo de contraseña por seguridad. Si el usuario no es encontrado, se retorna un mensaje de error indicando que no está autorizado.

En caso de que la validación del token falle por estar vencido, manipulado o ser inválido, el middleware captura la excepción y responde con un error adecuado, informando que el token no es válido o ha expirado.

Finalmente, si todas las comprobaciones son exitosas, el objeto req.user se completa con la información del usuario autenticado, permitiendo que las siguientes funciones accedan a sus datos de manera segura. Esta estructura permite implementar un sistema de autenticación robusto en cualquier endpoint del backend que lo requiera.

6.5.5 Carpeta routes

Define las rutas del backend, que actúan como puntos de entrada para las solicitudes HTTP.

Archivos carpeta "routes/"	Función
courseRoutes.js	Contiene todas las rutas relacionadas con los cursos, como creación, eliminación, subida de material, interacción con inteligencia artificial (IA) y generación de exámenes o resúmenes.
grades.routes.js	Contiene rutas para consultar las calificaciones agrupadas por curso.
userRoutes.js	Contiene rutas relacionadas con los usuarios, como registro, inicio de sesión, perfil, autenticación, calificaciones y estadísticas.

Tabla 6-14: Representación archivos carpeta "routes/"

Las rutas que contiene el archivo "coursesRoutes.js", se muestra a continuación en la siguiente tabla:

Ejemplo de rutas	Función
POST /	Crear un curso.
GET /	Obtener todos los cursos.
GET /:id	Obtener un curso por ID.
PUT /:id	Actualizar un curso.
DELETE /:id	Eliminar un curso.
DELETE /:id/material	Eliminar material del curso.
POST /summary	Generar resumen del curso.
POST /chat	Interactuar con el asistente IA.
POST /exam	Generar examen.
POST /exam/grade	Calificar examen.
GET /:id/has-material	Verificar si hay material.
POST /chat/context	Preguntar con contexto.
GET /chat/history/:courseId	Obtener historial de chat del curso.
GET /:id/material/:filename	Descargar archivo de material.
POST /analyze	Endpoint de prueba para análisis.

Tabla 6-15: Rutas archivo "coursesRoutes.js"

Las rutas que contiene el archivo "grades.routes.js" es "GET /grades" para obtener un listado estructurado de calificaciones por curso.

Las rutas que contiene el archivo "userRoutes.js", se muestra a continuación en la siguiente tabla:

Ejemplo de rutas	Función
POST /register	Registrar un nuevo usuario
POST /login	Autenticar un usuario
GET /protected	Ruta protegida de prueba
GET /profile	Obtener perfil del usuario
GET /grades	Obtener calificaciones

GET /exam/correction/:examId	Obtener corrección de examen
GET /statistics	Obtener estadísticas del usuario

Tabla 6-16: Rutas archivo "userRoutes.js"

6.5.6 Carpeta uploads

Almacena los archivos subidos por los usuarios, en este caso los documentos PDF.

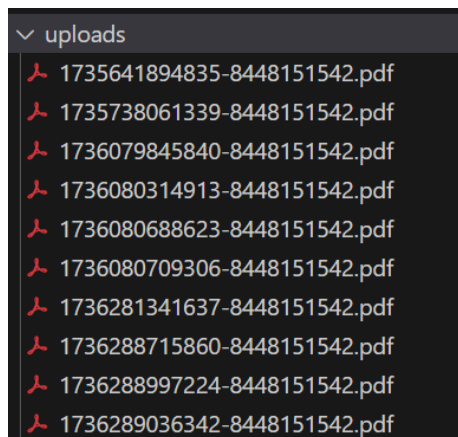


Ilustración 6-37: Archivo que pertenecen a la carpeta uploads

6.5.7 Carpeta utils

En la carpeta "utils" contiene el archivo "pdfUtils.js" se encarga de procesar documentos PDF subidos por los usuarios, específicamente para extraer el contenido textual que contienen. Esta funcionalidad es esencial en el sistema para tareas como la generación automática de resúmenes, análisis semántico del contenido y preparación de material para exámenes.

```
import fs from 'fs';
import pdf from 'pdf-parse';

export const extractTextFromPDF = async (filePath) => {
  try {
    const dataBuffer = fs.readFileSync(filePath);
    const data = await pdf(dataBuffer);

    // Convertir texto en string
    const text = data.text.replace(/\s+/g, ' ').trim(); // Eliminar espacios extra
    return text || ''; // Devuelve string vacío si no hay texto
  } catch (error) {
    console.error('Error al extraer texto del PDF:', error);
    throw new Error('No se pudo extraer el texto del PDF');
  }
};
```

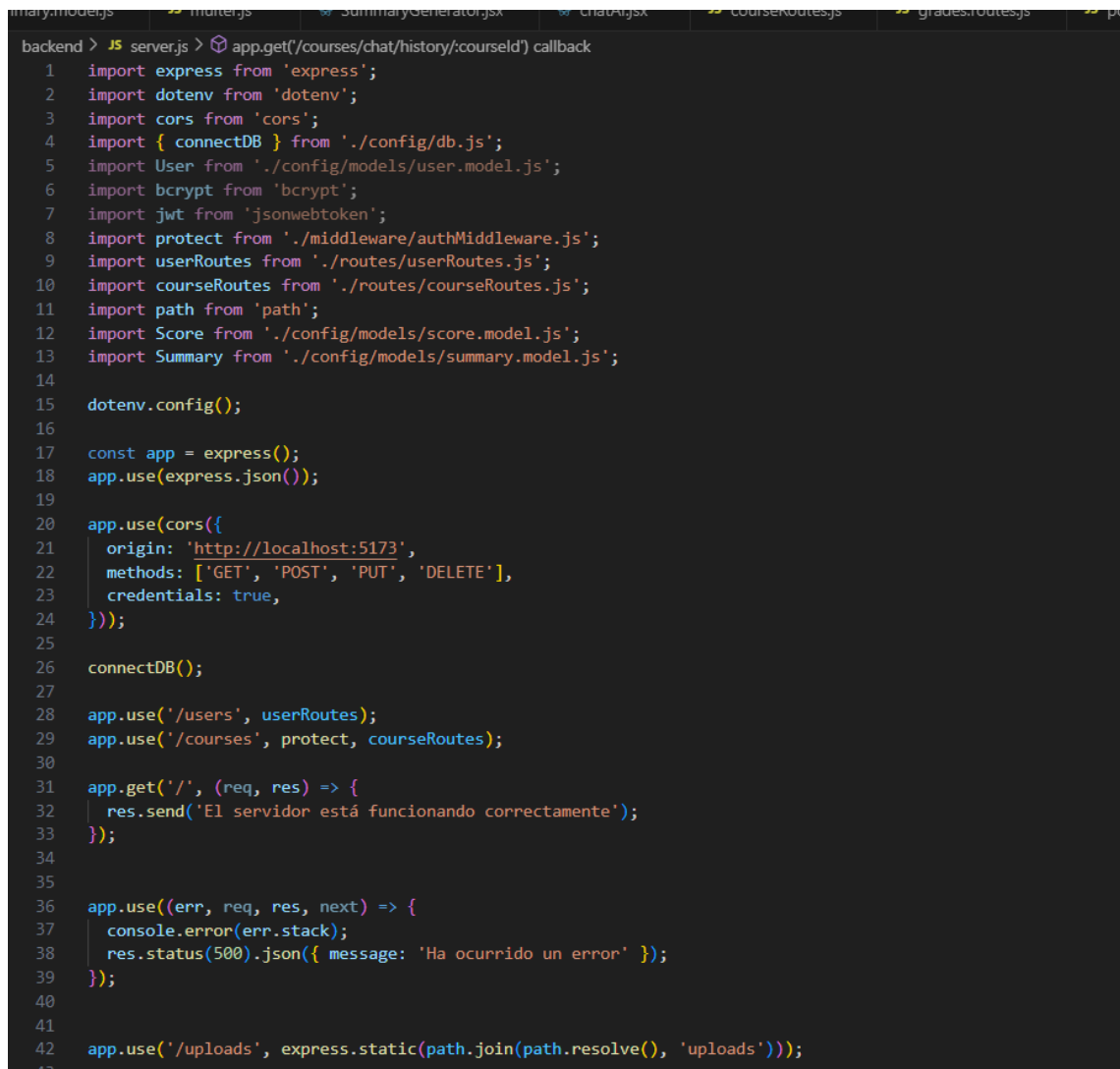
Ilustración 6-38: Código archivo "pdfUtils.js"

La función `extractTextFromPDF(filePath)`, toma como entrada la ruta de un archivo PDF almacenado en el servidor. Utiliza la biblioteca "pdf-parse" para interpretar el contenido del archivo, que es leído en formato binario mediante `fs.readFileSync`. Luego, el texto extraído se limpia aplicando una expresión

regular que elimina saltos de línea, múltiples espacios y tabulaciones, dejando una cadena de texto plano compacta y coherente.

6.5.8 Archivo server.js

El punto de entrada principal para iniciar el servidor. Configura el entorno global y registra todas las rutas y middlewares.



```
backend > JS server.js > app.get('/courses/chat/history/:courseId') callback
1  import express from 'express';
2  import dotenv from 'dotenv';
3  import cors from 'cors';
4  import { connectDB } from './config/db.js';
5  import User from './config/models/user.model.js';
6  import bcrypt from 'bcrypt';
7  import jwt from 'jsonwebtoken';
8  import protect from './middleware/authMiddleware.js';
9  import userRoutes from './routes/userRoutes.js';
10 import courseRoutes from './routes/courseRoutes.js';
11 import path from 'path';
12 import Score from './config/models/score.model.js';
13 import Summary from './config/models/summary.model.js';
14
15 dotenv.config();
16
17 const app = express();
18 app.use(express.json());
19
20 app.use(cors({
21   origin: 'http://localhost:5173',
22   methods: ['GET', 'POST', 'PUT', 'DELETE'],
23   credentials: true,
24 }));
25
26 connectDB();
27
28 app.use('/users', userRoutes);
29 app.use('/courses', protect, courseRoutes);
30
31 app.get('/', (req, res) => {
32   res.send('El servidor está funcionando correctamente');
33 });
34
35
36 app.use((err, req, res, next) => {
37   console.error(err.stack);
38   res.status(500).json({ message: 'Ha ocurrido un error' });
39 });
40
41
42 app.use('/uploads', express.static(path.join(path.resolve(), 'uploads')));
43
```

Ilustración 6-39: Primera parte código archivo "server.js"

```

app.post('/scores', protect, async (req, res) => {
  try {
    const { topic, score, examType } = req.body;
    const userId = req.user.id;
    const newScore = new Score({ userId, topic, score, examType });
    await newScore.save();
    res.status(201).json({ message: 'Calificación guardada.' });
  } catch (error) {
    res.status(500).json({ message: 'Error al guardar la calificación.' });
  }
});

app.get('/courses/chat/history/:courseId', protect, async (req, res) => {
  const { courseId } = req.params;
  const userId = req.user.id;

  try {
    const chatHistory = await Chat.find({ courseId, userId });
    res.json({ history: chatHistory });
  } catch (error) {
    res.status(500).json({ message: 'Error al obtener el historial del chat.' });
  }
});

app.get('/courses/summary/history/:courseId', protect, async (req, res) => {
  const { courseId } = req.params;
  const userId = req.user.id;

  try {
    const history = await Summary.find({ courseId, userId });
    res.json({ history });
  } catch (error) {
    res.status(500).json({ message: 'Error al obtener el historial de resúmenes.' });
  }
});

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
  console.log(`Servidor corriendo en http://localhost:${PORT}`);
});

```

Ilustración 6-40: Segunda parte código archivo "server.js"

Carga de variables de entorno

```
dotenv.config();
```

Se utiliza dotenv para cargar variables de entorno desde un archivo .env. Esto permite mantener configuraciones sensibles, como claves API y configuraciones de base de datos, fuera del código fuente.

Inicialización de express

```
const app = express();
```

```
app.use(express.json());
```

Se inicializa el servidor Express y se habilita el middleware para analizar cuerpos de solicitudes en formato JSON.

Configuración de CORS

```
app.use(cors({  
  origin: 'http://localhost:5173',  
  methods: ['GET', 'POST', 'PUT', 'DELETE'],  
  credentials: true,  
}));
```

Esto permite que el frontend, alojado en `http://localhost:5173`, se comuniquen con este servidor backend. También habilita métodos HTTP específicos y el envío de cookies/credenciales.

Conexión a la base de datos

```
connectDB();
```

Llama a la función `connectDB` para establecer una conexión con una base de datos MongoDB. Esta función está definida en el backend ubicado en `./config/db.js`.

Definición de rutas

Rutas de usuarios

```
app.use('/users', userRoutes);
```

Gestiona las rutas relacionadas con los usuarios, como registro, inicio de sesión y autenticación. Estas rutas están definidas en `userRoutes.js`.

Rutas de cursos (protegidas)

```
app.use('/courses', protect, courseRoutes);
```

Todas las rutas relacionadas con cursos están protegidas por el middleware `protect`, que verifica la autenticación del usuario mediante un token JWT. Las rutas específicas están definidas en `courseRoutes.js`.

Rutas y funcionalidades adicionales

Ruta base

```
app.get('/', (req, res) => {  
  
  res.send('El servidor está funcionando correctamente');  
});
```

```
});
```

Una ruta simple que indica que el servidor está operativo.

Rutas de subida de archivos

```
app.use('/uploads', express.static(path.join(path.resolve(),  
'uploads')));
```

Permite servir archivos estáticos desde la carpeta uploads, como documentos subidos.

Guardado de calificaciones

```
app.post('/scores', protect, async (req, res) => {  
  const { topic, score, examType } = req.body;  
  
  const userId = req.user.id;  
  
  const newScore = new Score({ userId, topic, score, examType });  
  
  await newScore.save();  
  
  res.status(201).json({ message: 'Calificación guardada.' });  
});
```

Esta ruta permite a usuarios autenticados guardar calificaciones en la base de datos. Los datos incluyen el tema del examen, el puntaje y el tipo de examen.

Historial de chats del curso

```
app.get('/courses/chat/history/:courseId', protect, async (req,  
res) => {  
  
  const { courseId } = req.params;  
  
  const userId = req.user.id;  
  
  try {  
  
    const chatHistory = await Chat.find({ courseId, userId });  
  
    res.json({ history: chatHistory });  
  
  } catch (error) {
```

```

        res.status(500).json({ message: 'Error al obtener el
historial del chat.' });

    }

});

```

Este endpoint permite obtener el historial completo de interacciones (preguntas y respuestas) entre un usuario y el sistema dentro de un curso específico.

- El parámetro :courseId identifica el curso del cual se desea recuperar el historial.
- El middleware protect garantiza que solo el usuario autenticado pueda acceder a su propio historial.
- Se realiza una búsqueda en la colección Chat filtrando por el courseId y el userId del usuario autenticado.
- La respuesta contiene un arreglo con todas las conversaciones previas en ese curso.

Historial de resúmenes del curso

```

app.get('/courses/summary/history/:courseId', protect, async
(req, res) => {

    const { courseId } = req.params;

    const userId = req.user.id;

    try {

        const history = await Summary.find({ courseId, userId });

        res.json({ history });

    } catch (error) {

        res.status(500).json({ message: 'Error al obtener el historial
de resúmenes.' });

    }

});

```

Este endpoint permite recuperar todos los resúmenes generados mediante IA por un usuario para un curso específico.

- El parámetro :courseId indica el curso del cual se desean obtener los resúmenes.
- El middleware protect asegura que únicamente el usuario autenticado pueda acceder a sus propios resúmenes.
- Se consultan los registros en la colección Summary, filtrando por courseId y userId.
- La respuesta contiene un historial completo de temas y resúmenes generados por el usuario en ese curso.

Middleware de manejo de errores

```
app.use((err, req, res, next) => {  
  console.error(err.stack);  
  res.status(500).json({ message: 'Ha ocurrido un error' });  
});
```

Cualquier error que ocurra en el servidor será capturado por este middleware y se enviará una respuesta estándar al cliente con un mensaje de error

Inicio del servidor

```
const PORT = process.env.PORT || 5000;  
  
app.listen(PORT, () => {  
  console.log(`Servidor corriendo en http://localhost:${PORT}`);  
});
```

El servidor escucha en un puerto definido en las variables de entorno o por defecto en el puerto 5000.

6.6 Ventajas implementación backend

Características	Descripción
Modularidad	La funcionalidad está distribuida en archivos separados como <code>userRoutes.js</code> , <code>courseRoutes.js</code> y <code>authMiddleware.js</code> , lo que facilita la escalabilidad y el mantenimiento del código.
Seguridad	El middleware de autenticación 'protect' protege rutas sensibles mediante tokens JWT, asegurando que solo usuarios autenticados accedan a funcionalidades privadas.
Manejo de Errores	El middleware global captura errores y los maneja de forma centralizada, mejorando la estabilidad del sistema.
Flexibilidad	Soporte para CORS e integración con diferentes dominios. Maneja múltiples métodos HTTP (GET, POST, PUT, DELETE) para diversas operaciones.
Gestión de Archivos	La carpeta 'uploads' permite subir y almacenar documentos, especialmente PDFs, facilitando la interacción del usuario con el sistema.
Conexión con Base de Datos	Utiliza Mongoose para conectarse a MongoDB, permitiendo definir esquemas y modelos estructurados para los datos.
Configuración Centralizada	Uso de variables de entorno con <code>dotenv</code> para una configuración segura y adaptable a distintos entornos (desarrollo, producción, etc.).
Escalabilidad	El diseño modular del backend permite agregar nuevas funcionalidades sin afectar el código existente.
Compatibilidad	El uso de JSON como formato de intercambio de datos garantiza compatibilidad con diversos frontends y plataformas.
Gestión de Calificaciones	Permite registrar calificaciones de usuarios vinculadas a exámenes, ideal para aplicaciones educativas o de evaluación.

Tabla 6-17: Ventajas implementación backend

6.7 Configuración frontend

- node_modules
- pages
 - course.jsx
 - CourseDetails.jsx
 - CreatePage.jsx
 - EditCourse.jsx
 - Home.jsx
 - login.jsx
 - Register.jsx
- src
 - assets
 - components
 - ChatAI
 - ChatAI.css
 - ChatAI.jsx
 - ExamGenerator
 - ExamGenerator.css
 - ExamGenerator.jsx
 - SummaryGenerator
 - SummaryGenerator.css
 - SummaryGenerator.jsx
 - Dashboard.jsx
 - Footer.jsx
 - Navbar.jsx
 - ProtectedRoute.jsx
 - App.css
 - App.jsx
 - index.css
 - main.jsx
- index.html
- package.json

6.7.1 Carpeta node_modules

Carpeta generada automáticamente al ejecutar npm install. Contiene todas las dependencias y paquetes necesarios para que la aplicación funcione correctamente, con los paquetes incluidos, por ejemplo: React, React DOM, Vite (herramienta de construcción), entre otros.

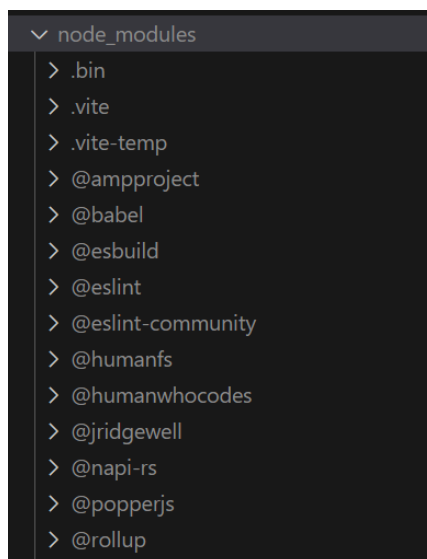


Tabla 6-18: Ejemplo de paquetes incluidos de la carpeta node_modules

6.7.2 Carpeta pages

Carpeta donde se encuentran las páginas principales de la aplicación, definidas como componentes React. Representan las vistas que los usuarios interactúan directamente.

Componente	Descripción
course.jsx	Página para listar y gestionar cursos. Muestra la lista de cursos disponibles y permite acceder a sus detalles.
CourseDetails.jsx	Muestra los detalles específicos de un curso seleccionado, incluyendo nombre, descripción y material relacionado.
CreatePage.jsx	Página para crear nuevos cursos. Incluye formularios para añadir información como título, descripción y material.
EditCourse.jsx	Permite editar un curso existente con información precargada del curso a modificar.
Home.jsx	Página de inicio de la aplicación. Suele contener una introducción o resumen de las funcionalidades de la plataforma.
login.jsx	Página de inicio de sesión. Contiene un formulario para que los usuarios ingresen sus credenciales (email y contraseña).
Register.jsx	Página para registrar nuevos usuarios. Permite introducir nombre,

	email y contraseña en un formulario de registro.
--	--

Tabla 6-19: Descripción archivos de la carpeta pages

6.7.3 Carpeta src

Carpeta principal donde se encuentra la lógica y el código fuente del frontend.

De subcarpetas que almacenan archivos se encuentran:

- assets/: Almacena recursos multimedia como imágenes, íconos o archivos relacionados con el diseño.
- components/: Contiene componentes reutilizables de React que se utilizan en varias páginas o secciones de la aplicación web.

6.7.4 Carpeta components

En esta tabla se muestra los estilos y componentes de las subcarpetas sobre las funcionalidades principales de la aplicación web.

Subcarpeta	Estilos	Componente	Responsabilidades principales	Características
chatAI	Llamado el archivo 'chatAI.css'. Contiene los estilos específicos para el componente de ChatAI. Asegura que el diseño sea consistente. Ejemplo: estiliza el cuadro de chat, mensajes, campo de entrada y botón 'Enviar'.	Llamado el archivo 'chatAI.jsx'. Componente React para el chat interactivo con la IA.	Gestionar el estado de las preguntas y respuestas. Manejar las solicitudes a la API de OpenAI. Renderizar la interfaz del chat.	Casilla de entrada para preguntas. Muestra respuestas de la IA en tiempo real. Manejo de errores (sin conexión, etc.).
ExamGenerator	Llamado el archivo 'ExamGenerator.css'. Estilos específicos para la generación de exámenes. Ejemplo: contenedor del examen, etiquetas, formularios, botones como 'Generar', 'Corregir', 'Reintentar', área de resultados.	Llamado el archivo 'ExamGenerator.jsx'. Componente React para generar exámenes según tipo y número de preguntas.	Permite realizar exámenes y recibir calificación en tiempo real. Recibe parámetros de entrada. Solicita generación al backend de preguntas basadas en el contenido. Renderiza preguntas generadas en la interfaz.	Opciones de preguntas (abiertas o múltiple). Muestra preguntas y respuestas. Manejo de calificaciones (envío y cálculo de resultados).

SummaryGenerator	Llamado el archivo 'SummaryGenerator.css'. Estilos específicos para el generador de resúmenes. Ejemplo: cuadro de chat, campo de entrada y botón 'Enviar'.	Llamado el archivo 'SummaryGenerator.jsx'. Componente React para generar resúmenes.	Permite ingresar texto o documentos para resumir. Envía solicitudes al backend. Muestra el resumen generado.	Opción para limitar el número de palabras en el resumen, compatibilidad con material del curso.
------------------	--	---	--	---

Tabla 6-20: Archivos de subcarpeta components

6.7.5 Componentes generales en la carpeta Components

Dashboard

Componente React que actúa como la página de inicio para usuarios autenticados. Su función principal es ofrecer una experiencia guiada para comenzar a usar la aplicación, gestionar asignaturas y acceder a funcionalidades clave como creación de cursos, carga de materiales, generación de resúmenes, chat con IA y creación de exámenes.

Footer

Componente que representa el pie de página de la aplicación. Incluye información de derechos de autor.

Navbar

Componente que representa la barra de navegación principal. Permite al usuario desplazarse por las diferentes secciones de la aplicación y gestiona el estado de autenticación del usuario.

6.7.1 ProtectedRoute.jsx

Componente de enrutamiento protegido que verifica si el usuario está autenticado antes de permitir el acceso a ciertas rutas. En caso de no estar autenticado, redirige al usuario a la página de inicio de sesión.

6.7.2 Representación componentes generales de la carpeta components

Componente	Responsabilidades	Ejemplo de Contenido	Descripción General
Dashboard.jsx	Obtener asignaturas del usuario desde el backend y mostrarlas. Ofrecer accesos rápidos para crear asignaturas y guiar al usuario en los primeros pasos.	Lista de asignaturas, botones para crear nuevas, sección “Cómo empezar” con pasos iniciales, spinner de carga.	Componente que actúa como pantalla de inicio tras iniciar sesión. Presenta las asignaturas existentes y fundamental para comenzar con funcionalidades como subir material, generar resúmenes, chatear o crear exámenes.
Footer.jsx	Mostrar información general, como derechos de autor o enlaces importantes.	© 2025 Nombre de la Aplicación. Todos los derechos reservados.	Componente que representa el pie de página de la aplicación. Incluye información como derechos de autor o enlaces de utilidad para el usuario.
Navbar.jsx	Proporcionar enlaces para navegar entre las páginas (inicio, chat, exámenes, resúmenes). Gestionar el estado de inicio/cierre de sesión del usuario.	Opciones de navegación para usuarios autenticados y no autenticados. Botón de cierre de sesión.	Componente que representa la barra de navegación principal. Permite al usuario desplazarse por las diferentes secciones de la aplicación y gestiona el estado de autenticación del usuario.
ProtectedRoute.jsx	Verificar si el usuario está autenticado antes de permitir el acceso. Redirigir a la página de inicio de sesión si no está autenticado.	Implementa lógica para proteger rutas como la creación de cursos o generación de exámenes.	Componente de enrutamiento protegido que verifica si el usuario está autenticado antes de permitir el acceso a ciertas rutas. En caso de no estar autenticado, redirige al usuario a la página de inicio de sesión.

Tabla 6-21: Componentes generales carpeta components

6.7.3 Archivo App.jsx

```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Navbar from '../components/Navbar';
import Footer from '../components/Footer';
import Home from '../pages/Home';
import Login from '../pages/login';
import Register from '../pages/Register';
import Dashboard from '../components/dashboard';
import ProtectedRoute from '../components/ProtectedRoute';
import Courses from '../pages/course';
import CourseDetails from '../pages/CourseDetails';
import EditCourse from '../pages/EditCourse';
import ChatAI from '../components/ChatAI/chatAI';
import ExamGenerator from '../components/ExamGenerator/ExamGenerator';
import SummaryGenerator from '../components/SummaryGenerator/SummaryGenerator';

const App = () => {
  return (
    <Router>
      <Navbar />
      <div className="container-fluid p-0">
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/login" element={<Login />} />
          <Route path="/register" element={<Register />} />
          <Route
            path="/dashboard"
            element={
              <ProtectedRoute>
                <Dashboard />
              </ProtectedRoute>
            }
          />
          <Route
            path="/courses"
            element={
              <ProtectedRoute>
                <Courses />
              </ProtectedRoute>
            }
          />
          <Route
            path="/courses/:id"
            element={
```

Ilustración 6-41: Código primera parte del archivo App.jsx

En la “Ilustración 6-41” se puede apreciar la estructura de rutas implementada con React Router en una aplicación React. El componente principal está envuelto por BrowserRouter, renombrado como Router, permite que la aplicación utilice la API del historial del navegador para cambiar de página sin recargar la web, creando una experiencia más fluida para el usuario.

Dentro del componente Router, se encuentra el bloque Routes, que actúa como contenedor para todas las rutas que componen la aplicación. Cada ruta se define utilizando el componente Route, el cual recibe una propiedad path, que indica la URL que activará esa ruta, y una propiedad element, que especifica el componente que debe renderizarse cuando se accede a dicha ruta.

Por ejemplo, se define una ruta inicial ("/") que renderiza el componente Home, y también rutas como "/login" y "/register" que muestran los componentes Login y Register respectivamente. Estas son rutas públicas, es decir, cualquier usuario puede acceder a ellas sin restricciones.

Por otro lado, se observa el uso de rutas protegidas como "/dashboard" y "/courses", las cuales están envueltas en un componente ProtectedRoute. Este componente actúa como una capa de seguridad adicional, permitiendo el acceso únicamente a usuarios autenticados.

```

    />
    <Route
      path="/courses/:id"
      element={
        <ProtectedRoute>
          <CourseDetails />
        </ProtectedRoute>
      }
    />
    <Route
      path="/courses/edit/:id"
      element={
        <ProtectedRoute>
          <EditCourse />
        </ProtectedRoute>
      }
    />
    <Route
      path="/chat-ai"
      element={
        <ProtectedRoute>
          <ChatAI />
        </ProtectedRoute>
      }
    />
    <Route
      path="/exam-generator"
      element={
        <ProtectedRoute>
          <ExamGenerator />
        </ProtectedRoute>
      }
    />
    <Route
      path="/summary-generator"
      element={
        <ProtectedRoute>
          <SummaryGenerator />
        </ProtectedRoute>
      }
    />
  </Routes>
</div>
<Footer />
</Router>
);
};

export default App;
```

Ilustración 6-42: Código segunda parte del archivo App.jsx

Además de las rutas públicas y principales, en la segunda parte del código se pueden observar varias rutas protegidas que ofrecen funcionalidades específicas de la aplicación, todas envueltas en el componente `ProtectedRoute`. Este patrón asegura que dichas páginas solo puedan ser accedidas por usuarios que hayan iniciado sesión correctamente, reforzando la seguridad y la experiencia personalizada dentro de la plataforma.

Una de estas rutas es `"/courses/:id"`, la cual renderiza el componente `CourseDetails`. Su propósito es mostrar información detallada sobre un curso en particular, y el parámetro: `id` permite que la ruta sea dinámica, adaptándose a cada curso individual.

También se encuentra la ruta `"/courses/edit/:id"`, que utiliza el componente `EditCourse`. Esta ruta está pensada para permitir la edición de un curso existente. Al igual que la anterior, es dinámica y requiere autenticación para ser accedida, asegurando que solo los usuarios autorizados puedan modificar la información del curso.

Otra ruta destacada es `"/chat-ai"`, la cual carga el componente `ChatAI`. Esta funcionalidad permite que los usuarios interactúen con una inteligencia artificial para realizar consultas académicas o relacionadas al contenido del curso, siendo una herramienta clave dentro de la plataforma. Su acceso también está protegido mediante `ProtectedRoute`.

Asimismo, se define la ruta `"/exam-generator"`, que renderiza el componente `ExamGenerator`. A través de esta sección, el usuario puede generar exámenes personalizados basados en el material del curso. Al ser una herramienta avanzada que trabaja con datos personales y de aprendizaje, se mantiene protegida para garantizar su correcto uso.

Por último, se incluye la ruta `"/summary-generator"`, vinculada al componente `SummaryGenerator`. Esta funcionalidad permite crear resúmenes automáticos de un tema o contenido específico, utilizando inteligencia artificial. Como ocurre con las demás herramientas especializadas, su acceso está restringido a usuarios autenticados mediante `ProtectedRoute`.

6.8 Revisión de rendimiento y accesibilidad

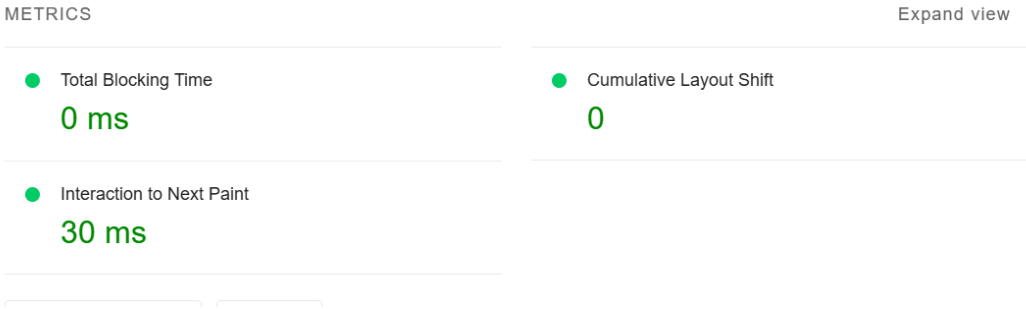


Ilustración 6-43: Métricas de rendimiento con baja carga operativa

Se ha empleado la herramienta “Lighthouse” [42] del navegador Microsoft Edge y se ha probado para situaciones normales como navegar de una sección a otra en este caso de “ChatAI” a “Generador de exámenes” sin seleccionar una asignatura y entrar en apartados de poca carga operativa como el de “Configuración” se obtiene de rendimiento los siguientes valores:

- Total Blocking Time (TBT) [43]: Se muestra con un valor de 0 ms quiere decir que no se bloquea en absoluto la interacción del usuario.
- Interaction to Next Paint (INP) [44]: Se muestra con un valor de 30 ms refleja una respuesta inmediata tras la interacción del usuario.
- Cumulative Layout Shift (CLS) [45]: Muestra que el diseño de la página permanece completamente estable, sin cambios inesperados en la posición de los elementos durante la carga.

A continuación, se muestra accediendo al “Dashboard” con la carga de todas las asignaturas y seleccionando una asignatura en una sección en este caso con “ChatAI” y mostrando sus conversaciones anteriores, se obtuvo los siguientes valores:

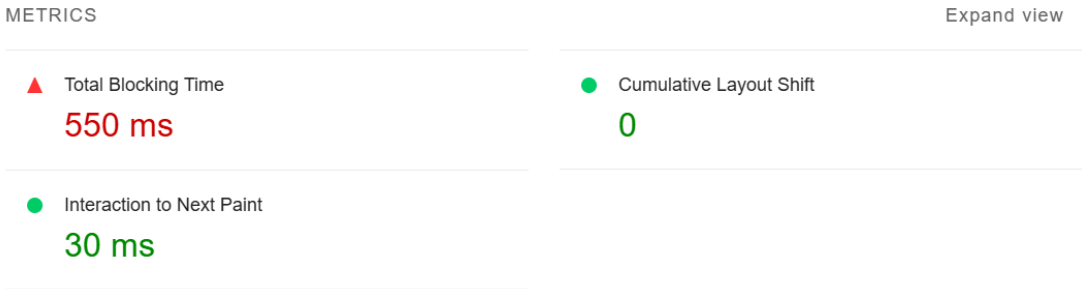


Ilustración 6-44: Métricas de rendimiento con alta carga operativa

Total Blocking Time (TBT): Se muestra con un valor de 550 ms, Este aumento significativo del TBT ocurre porque la aplicación necesita procesar y renderizar numerosos documentos PDF asociados a la asignatura seleccionada en “ChatAI”, generando una carga adicional en el navegador. Este proceso incluye operaciones costosas como la lectura de contenido,

extracción de texto, procesamiento mediante IA para generación de resúmenes y respuestas contextuales, por lo tanto, incrementa notablemente el tiempo de bloqueo de JavaScript en la interfaz.

Interaction to Next Paint (INP): Se muestra con un valor de 30 ms como en la comprobación anterior de poca carga operativa, a pesar del alto valor del TBT, la interacción sigue siendo ágil una vez cargados los contenidos iniciales, ya que la respuesta es inmediata después de la interacción del usuario.

Cumulative Layout Shift (CLS): Se sigue mostrando estable sin cambios inesperados con el mismo valor que en la comprobación anterior.

Sobre la apariencia se realiza un análisis general y verifica una lista de ítems que cumple la aplicación web, a continuación, se muestra algunos ejemplos:

PASSED AUDITS (11)		Hide
●	<code>[aria-hidden="true"]</code> is not present on the document <code><body></code>	▼
●	Image elements have <code>[alt]</code> attributes	▼
●	<code>[user-scalable="no"]</code> is not used in the <code><meta name="viewport"></code> element and the <code>[maximum-scale]</code> attribute is not less than 5.	▼
●	Document has a <code><title></code> element	▼
●	<code><html></code> element has a <code>[lang]</code> attribute	▼
●	<code><html></code> element has a valid value for its <code>[lang]</code> attribute	▼
●	Links are distinguishable without relying on color.	▼
●	Links have a discernible name	▼
●	List items (<code></code>) are contained within <code></code> , <code></code> or <code><menu></code> parent elements	▼

Ilustración 6-45: Apartados de apariencia que cumple la aplicación web

La aplicación web cumple con atributos “[alt]” en imágenes, atributos válidos en etiquetas “<html>”, etiquetas “<title>” y enlaces identificables.

Muestra que carece correctamente de restricciones como “[user-scalable=“no”]” y “[aria-hidden=“true”]”. Este aspecto es bastante positivo, porque si se hubiesen empleado estas restricciones, podría ocasionar problemas graves de accesibilidad, impidiendo el ajuste del contenido (zoom) para usuarios con dificultades visuales y ocultando su contenido.

Sobre la estructura, no presenta problemas en listas o visibilidad de enlaces sin depender del color.

En general, cumple con la mayoría de los puntos de accesibilidad mostrados.

Algunos ejemplos que muestra lo no se ha empleado:

- ☐ The page contains a heading, skip link, or landmark region
- ☐ `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements.
- ☐ Definition list items are wrapped in `<dl>` elements

Ilustración 6-46: Apartados de apariencia que carece la aplicación web

Carece de elementos de accesibilidad como encabezados claros, enlaces rápidos (skip links) o regiones de referencia, que ayudan a usuarios a navegar de forma eficiente, no se emplea ninguna etiqueta `<dl>` que se recomienda para mejorar la presentación de contenido de asignaturas.

Capítulo 7. Conclusiones y trabajos futuros

7.1 Conclusiones

El presente Trabajo Fin de Titulación (TFT) me ha aportado nuevos conocimientos y habilidades en tecnologías modernas, como puede ser integración de una api externa en este caso OpenAI para incorporar las funcionalidades basadas en inteligencia artificial. También he mejorado mis habilidades en creación de interfaces dinámicas con React, manejo de estados, e implementar un diseño adaptable a diferentes dispositivos. He comprendido el funcionamiento de middlewares, y gestión de tokens JWT para la autenticación. Encriptar contraseñas para que no sea visible la contraseña ni en la base de datos, aplicado en este proyecto.

He reforzado mis habilidades en la planificar un proyecto mediante Trello, Git y GitHub, fundamental para organizar tareas y mantener control de versiones.

Para su realización de forma satisfactoria han sido de utilidad los cursos de formación y manuales oficiales sobre las tecnologías implementadas.

Para la organización del trabajo la metodología iterativa e incremental ha sido crucial, implementando en cada iteración funcionalidades nuevas añadidas además de mejoras.

El producto resultante lo considero innovador ya que se centra en la educación principalmente en estudiantes, para que puedan estudiar de forma autónoma, evaluar sus conocimientos y obtener un seguimiento exhaustivo.

Valoro positivamente que las tecnologías desarrolladas son actuales, hay que tener en cuenta openai lanza versiones actuales, para las funcionalidades de esta aplicación web se ha empleado el “GPT – 4 Turbo” que fue anunciado el 6 de noviembre de 2023, los embeddings fueron entrenados hasta septiembre 2021, lo cual es fundamental para representar el contenido textual en un formato numérico comprensible por los modelos de inteligencia artificial. La aplicación web se basa en la tecnología MERN (MongoDB, Express.js, React.js, Node.js) que comenzó a utilizarse el conjunto de estas cuatro tecnologías en su totalidad en 2015.

Respecto a la interfaz de usuario se adapta a dispositivos móviles, debido al elevado uso de smartphones y tablets en la actualidad para acceder a plataformas digitales. Esta adaptabilidad garantiza una experiencia de usuario consistente, accesible y funcional sin importar el tamaño de pantalla, lo que mejora la usabilidad y experiencia para el usuario. Además, se aplicaron principios de diseño responsivo utilizando tecnologías como React y Bootstrap para asegurar una visualización fluida en distintos dispositivos.

En conclusión, cumple con lo que se planea previamente, contiene una planificación adecuada empleando historias de usuario con Trello que permite mostrar durante el desarrollo del trabajo en qué fase se encuentra,

funcionalidades principales y accesorias implementadas, uso de tecnologías actuales y un diseño adaptable (responsive) adaptado a ordenadores, smartphones y tablets.

7.2 Trabajos futuros

El Trabajo Fin de Titulación (TFT) se realiza en un tiempo limitado, ya que se realiza en el intervalo de un curso académico y se ha realizado más de 300 horas, siempre pudiendo añadir nuevas funcionalidades, en este caso si en un futuro se desea que el producto sea comercial se destacan las siguientes áreas de mejora que han sido identificadas durante el desarrollo del proyecto que no se han podido realizar por limitaciones de tiempo y superan el alcance planteado previamente:

- 1) **Análisis en busca de virus o amenaza sobre la documentación:** Al subir la documentación de la materia incorporar un botón llamado “Analizar”, que haga un escaneo antivirus de forma asíncrona, y mostrará una barra de progreso, al terminar actualizará el mismo visor con un indicador verde “Limpio” o rojo “Virus o amenaza encontrada”, y se podrá desglosar los detalles de esta.
- 2) **Soporte de múltiples idiomas en su interfaz y funcionalidades:** Una barra selectora en la parte superior de la web, en la que aparece de forma abreviada cada idioma y al seleccionar el idioma se actualiza automáticamente el idioma para toda la interfaz de la aplicación web.
- 3) **Configuración de exámenes con temporizador, formato de las preguntas de forma individual o secuencial y nivel de dificultad:** En la sección de “Configuración” añadir un campo llamado “Exámenes” que se podrá seleccionar en el selector el tiempo máximo de realización del examen, cuando el tiempo se termina automáticamente se corregirá el examen, además contendrá un selector que se podrá seleccionar si se desea las preguntas de forma individual o secuencial y otro selector para el nivel de dificultad.
- 4) **Configuración de resúmenes seleccionando si la redacción sea formal o informal:** En la sección de “Configuración” añadir un campo llamado “Resúmenes” y en el selector se podrá seleccionar si se desea la redacción formal o informal.
- 5) **Despliegue y requisitos legales:** Cuando se haga público la aplicación web se asegurará el cumplimiento de RGPD, normativa de privacidad, propiedad intelectual y derechos de autor, e incorporando términos y condiciones, y políticas de cookies.

- 6) **Monetización y gestión financiera:** Tendrá dos accesos para los usuarios uno de acceso gratuito llamado “Freemium” y otro de pago denominado “Premium”. El acceso gratuito el almacenamiento de documentos para las materias será limitado también para generación de resúmenes, exámenes y dudas. El acceso de pago se podrá subir documentos de asignaturas sin restricción, y generación ilimitada de resúmenes, exámenes, dudas, además la generación de todas ellas será con mayor precisión. Para los usuarios “Premium” se incorporará una pasarela de pago que sea intuitiva y segura y que permita en distintas plataformas digitales como PayPal, Stripe, Redsys.

Referencias

- [1] Wikipedia, «Procesamiento de lenguajes naturales,» Fundación Wikimedia, 20 Julio 2024. [En línea]. Available: https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales. [Último acceso: 28 Enero 2025].
- [2] Wikipedia, «Word embedding,» Fundación Wikimedia, 4 Mayo 2024. [En línea]. Available: https://es.wikipedia.org/wiki/Word_embedding. [Último acceso: 15 Enero 2025].
- [3] MongoDB, «MERN Stack Tutorial,» MongoDB Inc, [En línea]. Available: <https://www.mongodb.com/es/resources/languages/mern-stack-tutorial>. [Último acceso: 13 Noviembre 2024].
- [4] Finamik, «Lean Canvas: Crea un modelo de negocio en 2024,» 30 Mayo 2023. [En línea]. Available: <https://www.finaimik.com/blog/lean-canvas-modelo-negocio-exitoso/>. [Último acceso: 31 Marzo 2025].
- [5] A. Ken, «Requisitos funcionales: ¿Qué son y cómo funcionan?,» Gluo, 3 Diciembre 2024. [En línea]. Available: <https://www.gluo.mx/blog/requisitos-funcionales-que-son-y-como-funcionan>. [Último acceso: 29 Enero 2025].
- [6] A. Ken, «Requisitos no funcionales: ¿Por qué son importantes?,» Gluo, 17 Agosto 2023. [En línea]. Available: <https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes>. [Último acceso: 30 Enero 2025].
- [7] OpenAI, «OpenAI API,» OpenAI, [En línea]. Available: <https://openai.com/index/openai-api/>. [Último acceso: 30 Octubre 2024].
- [8] I. MongoDB, «Three Member Replica Sets,» MongoDB, [En línea]. Available: <https://www.mongodb.com/docs/manual/core/replica-set-architecture-three-members/>. [Último acceso: 4 Febrero 2025].
- [9] B. Team, «Build fast, responsive sites with Bootstrap,» Bootstrap Team, [En línea]. Available: <https://getbootstrap.com/>. [Último acceso: 15 Enero 2025].
- [10] EL PARLAMENTO EUROPEO Y EL CONSEJO DE LA UNIÓN EUROPEA, «Boe,» 27 Abril 2016. [En línea]. Available: <https://www.boe.es/doue/2016/119/L00001-00088.pdf>. [Último acceso: 28 Febrero 2025].
- [11] Jefatura del Estado, «Boe,» 5 Diciembre 2018. [En línea]. Available: <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>. [Último acceso: 28 Febrero 2025].
- [12] MIT License, «MIT License,» [En línea]. Available: <https://mit-license.org/>. [Último acceso: 28 Febrero 2025].

- [13] Apache Software Foundation, «Apache License, Version 2.0,» [En línea]. Available: <https://www.apache.org/licenses/LICENSE-2.0>. [Último acceso: 28 Febrero 2025].
- [14] Free Software Foundation, «GNU General Public License,» [En línea]. Available: <https://www.gnu.org/licenses/gpl-3.0.html>. [Último acceso: 28 Febrero 2025].
- [15] OECD, «OECD AI Principles overview,» [En línea]. Available: <https://oecd.ai/en/ai-principles>. [Último acceso: 28 Febrero 2025].
- [16] I. Meta Platforms, «React,» Meta Platforms, [En línea]. Available: <https://react.dev/>. [Último acceso: 29 Noviembre 2024].
- [17] C. Nwachukwu, «react-pdftotext,» npm, [En línea]. Available: <https://www.npmjs.com/package/react-pdftotext>. [Último acceso: 17 Diciembre 2024].
- [18] I. Shopify, «React Router,» Shopify, Inc., [En línea]. Available: <https://reactrouter.com/>. [Último acceso: 14 Noviembre 2024].
- [19] «react-spinners,» npm, [En línea]. Available: <https://www.npmjs.com/package/react-spinners>. [Último acceso: 20 Enero 2025].
- [20] R. Group, «Recharts,» Recharts Group, [En línea]. Available: <https://recharts.org/en-US>. [Último acceso: 25 Marzo 2025].
- [21] SweetAlert2, «SweetAlert2,» SweetAlert2, [En línea]. Available: <https://sweetalert2.github.io/>. [Último acceso: 5 Febrero 2025].
- [22] Okta, «jwt-decode,» npm, 2023. [En línea]. Available: <https://www.npmjs.com/package/jwt-decode>. [Último acceso: 10 Enero 2025].
- [23] Z. Matt, «Axios,» Axios, [En línea]. Available: <https://axios-http.com/>. [Último acceso: 20 Diciembre 2024].
- [24] O. Foundation, «Node.js,» OpenJS Foundation, [En línea]. Available: <https://nodejs.org/es>. [Último acceso: 4 Diciembre 2024].
- [25] O. Foundation, «Express,» OpenJS Foundation, [En línea]. Available: <https://expressjs.com/>. [Último acceso: 4 Diciembre 2024].
- [26] Multer, «multer,» npm, [En línea]. Available: <https://www.npmjs.com/package/multer>. [Último acceso: 9 Enero 2024].
- [27] Express, «express-fileupload,» npm, [En línea]. Available: <https://www.npmjs.com/package/express-fileupload>. [Último acceso: 4 Diciembre 2024].
- [28] npm, «pdf-parse,» npm, [En línea]. Available: <https://www.npmjs.com/package/pdf-parse>. [Último acceso: 6 Diciembre 2024].

- [29] S. Girish, «bcryptjs,» npm, [En línea]. Available: <https://www.npmjs.com/package/bcryptjs>. [Último acceso: 9 Diciembre 2024].
- [30] Auth0, «jsonwebtoken,» npm, [En línea]. Available: <https://www.npmjs.com/package/jsonwebtoken>. [Último acceso: 10 Diciembre 2024].
- [31] T. Goode, «cors,» npm, [En línea]. Available: <https://www.npmjs.com/package/cors>. [Último acceso: 15 Diciembre 2024].
- [32] dotenv, «dotenv,» npm, [En línea]. Available: <https://www.npmjs.com/package/dotenv/v/16.4.7>. [Último acceso: 1 Febrero 2024].
- [33] Microsoft Corporation, «Visual Studio Code,» Microsoft Corporation, [En línea]. Available: <https://code.visualstudio.com/>. [Último acceso: 15 Octubre 2024].
- [34] Software Freedom Conservancy, «Git,» Software Freedom Conservancy, [En línea]. Available: <https://git-scm.com/>. [Último acceso: 15 Enero 2025].
- [35] GitHub, Inc, «GitHub,» GitHub, Inc, [En línea]. Available: <https://github.com/>. [Último acceso: 15 Enero 2025].
- [36] Postman, Inc, «Postman,» Postman, Inc, [En línea]. Available: <https://www.postman.com/>. [Último acceso: 15 Enero 2025].
- [37] Atlassian Corporation Plc, «Trello,» Atlassian Corporation Plc, [En línea]. Available: <https://trello.com/>. [Último acceso: 20 Febrero 2025].
- [38] JavaScript Mastery, «Build and Deploy a Full Stack MERN AI Image Generation App,» JavaScript Mastery, 20 Enero 2023. [En línea]. Available: <https://www.youtube.com/watch?v=EylvuigqDoA>. [Último acceso: 4 Diciembre 2024].
- [39] Daniel | Tech & Data, «React the Complete Guide Course Review - 2025 | Maximilian Schwarzmüller's React Course Review,» [En línea]. Available: <https://www.youtube.com/watch?v=RceMRzibELU>. [Último acceso: 6 Diciembre 2024].
- [40] MongoDB, Inc, «Learn MongoDB and Advance Your Career,» MongoDB, Inc, [En línea]. Available: <https://learn.mongodb.com/>. [Último acceso: 30 Noviembre 2024].
- [41] DCT Academy, «DCT Academy,» [En línea]. Available: <https://www.dctacademy.com/blog/what-is-object-document-mapper-odm>. [Último acceso: 28 Mayo 2025].
- [42] Wikipedia, «Google Lighthouse,» [En línea]. Available: https://es.wikipedia.org/wiki/Google_Lighthouse. [Último acceso: 28 Mayo 2025].
- [43] B. P. Philip Walton, «Tiempo de bloqueo total (TBT),» [En línea]. Available: <https://web.dev/articles/tbt?hl=es-419>. [Último acceso: 28 Mayo 2025].

- [44] J. Wagner, «Interaction to Next Paint (INP),» [En línea]. Available: <https://web.dev/articles/inp?hl=es-419>. [Último acceso: 28 Mayo 2025].
- [45] P. W. Milica Mihajlija, «Cumulative Layout Shift (CLS),» [En línea]. Available: <https://web.dev/articles/cls?hl=es-419>. [Último acceso: 28 Mayo 2025].
- [46] B. P. Philip Walton, «web.dev,» [En línea]. Available: <https://web.dev/articles/tbt?hl=es-419>. [Último acceso: 28 Mayo 2025].