

**FACULTAD DE CIENCIAS**  
**GRADO EN INGENIERÍA INFORMÁTICA**



**VNiVERSIDAD  
D SALAMANCA**

**TRABAJO DE FIN DE GRADO**

**Sistema de generación de diagramas de  
clases en UML basado en Inteligencia  
Artificial**

**Class diagram generation system in UML  
based on Artificial Intelligence**

Autor:  
Cristian Alejandro Castillo Salguero

Tutores:  
Francisco José García Peñalvo  
Andrea Vázquez Ingelmo

Curso 2023/24

## Resumen

En el ámbito de la Ingeniería de Software, el aprendizaje del lenguaje *Unified Model Language* (UML) constituye un elemento fundamental para el desarrollo de sistemas de software. Uno de los desafíos a los que enfrentan los estudiantes, es la dificultad para generar y resolver nuevos problemas a partir de modelos de dominio UML. Ante este escenario, se ha desarrollado una aplicación web que integra tanto modelos de generación de texto como de imágenes, permitiendo a los usuarios obtener enunciados de problemas y su solución, así como adjuntar imágenes de diagramas de clases para poder obtener una explicación de mismo y poder asimilar correctamente los conceptos. Como enfoque a la hora de llevar a cabo el proyecto se ha utilizado una metodología ágil, con el objetivo de maximizar la eficiencia y adaptabilidad. Para la realización del entorno web se ha usado el *framework* web Django y para la generación de texto e imágenes han utilizado los modelos ChatGPT 3.5 Turbo y ChatGPT-4 usando las interfaces de servicios que provee OpenAI para dichos modelos de lenguaje. Se han cumplido todos los objetivos propuestos, además de otros aspectos que han surgido durante el desarrollo y que han permitido identificar nuevas funcionalidades. Como resultado se tiene una aplicación operativa que puede ofrecerse al estudiantado y profesorado para ayudar en el proceso de enseñanza aprendizaje.

## Abstract

Learning the Unified Modeling Language (UML) is a fundamental element of software system development in software engineering. One of the challenges students face is the difficulty in generating and solving new problems based on UML domain models. In response to this scenario, a web application has been developed that integrates both text and image generation models, enabling users to obtain problem statements and their solutions and attach class diagram images to receive explanations and correctly assimilate the concepts. An agile methodology maximizes efficiency and adaptability during the project execution. The Django web framework was used to create the web environment, while ChatGPT 3.5 Turbo and ChatGPT-4 models, utilizing OpenAI's service interfaces, were employed for text and image generation. All proposed objectives were met, along with additional aspects that emerged during development, leading to the identification of new functionalities. As a result, an operational application is available for students and faculty to aid in the teaching and learning process.

## Glosario

1. API (*Application Programming Interface*): conjunto de reglas y especificaciones que permiten que diferentes programas de software se comuniquen entre sí.
2. BD (Base de datos): sistema organizado de almacenamiento de datos que permite la inserción, almacenamiento, consulta, actualización y administración de datos de manera eficiente y segura.
3. DALL-E: Modelo de inteligencia artificial desarrollado por OpenAI que genera imágenes a partir de descripciones textuales.
4. DOM (*Document Object Model*): interfaz de programación para documentos web. Proporciona una representación estructurada del documento HTML o XML y define la manera en que la estructura puede ser accedida y manipulada por programas.
5. *Feedback*: respuestas o información proporcionada por los usuarios sobre su experiencia con un sistema o servicio.
6. *Framework*: conjunto estructurado y reutilizable de herramientas, librerías y prácticas recomendadas que proporcionan una base para desarrollar aplicaciones de software.
7. Gamificación: aplicación de elementos y principios del diseño de juegos en contextos no lúdicos, como la educación o la empresa, para mejorar la participación y la motivación de los usuarios.
8. GRIAL (Grupo de Investigación en Interacción y eLearning): grupo de investigación especializado en tecnologías educativas y e-learning, basado en la Universidad de Salamanca.
9. IA (Inteligencia Artificial): tecnología que permite que las máquinas imiten las capacidades cognitivas humanas, como aprender y resolver problemas.
10. IDE (*Integrated Development Environment*): entorno de desarrollo integrado que proporciona a los programadores un conjunto completo de herramientas de software para escribir, probar y depurar su código en una sola interfaz. Combina características como editor de texto, compilador, intérprete y herramientas de depuración, facilitando el desarrollo de aplicaciones.
11. LLM (*Large Language Model*): modelo de lenguaje de gran escala diseñado para entender y generar texto humano de manera coherente y relevante.
12. *Middleware*: software que actúa como un puente entre las aplicaciones y la red o el sistema operativo.

13. MVC (Modelo-Vista-Controlador): patrón de diseño de software que separa la información, la interfaz de usuario y la lógica de control en tres componentes interconectados.
14. MVT (*Model-View-Template*): variante del patrón MVC, el cual sustituye la vista por la plantilla (*template*), el modelo sigue siendo el mismo y el controlador ahora es la vista.
15. PlantUML: herramienta que permite a los usuarios crear diagramas a partir de un lenguaje descriptivo en texto plano.
16. PostgreSQL: sistema de gestión de bases de datos relacional, conocido por su robustez y conformidad con los estándares SQL.
17. Python: lenguaje de programación de alto nivel, dinámico y ampliamente utilizado para desarrollo web, análisis de datos, inteligencia artificial y más.
18. SAAS (*Software as a Service*): modelo de distribución de software donde las aplicaciones son alojadas por un proveedor de servicios y puestas a disposición de los clientes a través de internet.
19. SCRUM: marco de trabajo para desarrollo ágil de software que organiza el trabajo en ciclos cortos e incrementales llamados *sprints*.
20. *Sprints*: fases de trabajo en la metodología SCRUM, durante las cuales los equipos completan ciertas partes de un proyecto de software.
21. Stakeholders: en el marco de trabajo SCRUM, los *stakeholders* se refieren a todas las personas involucradas en el proyecto que tienen un interés en el resultado final, pero que no son parte del equipo SCRUM principal.
22. SQL (*Structured Query Language*): lenguaje de programación diseñado para gestionar y manipular bases de datos relacionales. Se utiliza para consultar, actualizar, e insertar datos en una base de datos, así como para la administración y definición de estructuras de datos.
23. *Stakeholders*: en el marco de trabajo SCRUM, se refieren a todas las personas involucradas en el proyecto que tienen un interés en el resultado final, pero que no son parte del equipo SCRUM principal.
24. TIC (Tecnologías de la Información y la Comunicación): conjunto de recursos, herramientas, equipos, programas informáticos, aplicaciones, redes y medios que permiten la compilación, procesamiento, almacenamiento y transmisión de información como voz, datos, texto, vídeo e imágenes.
25. UML (*Unified Modeling Language*): lenguaje de modelado estándar que se utiliza para visualizar, especificar, construir y documentar los artefactos de sistemas de software.

## Índice de capítulos

<b>Introducción</b> .....	<b>12</b>
<b>Objeto</b> .....	<b>14</b>
Objetivos funcionales .....	14
Objetivos personales .....	14
<b>Antecedentes</b> .....	<b>15</b>
Aspectos teóricos.....	15
Grupo de Investigación GRIAL.....	16
<b>Descripción de la situación actual</b> .....	<b>17</b>
Análisis de la competencia.....	17
<b>Normas y referencias</b> .....	<b>19</b>
<b>Métodos</b> .....	<b>19</b>
<b>Herramientas</b> .....	<b>20</b>
Herramientas de implementación.....	20
Herramientas de comunicación y almacenamiento.....	20
Herramientas para la elaboración del TFG .....	21
<b>Modelos</b> .....	<b>21</b>
<b>Prototipos</b> .....	<b>23</b>
Wireframes.....	23
Mockups o prototipos digitales .....	24
<b>Requisitos iniciales</b> .....	<b>25</b>
Diagramas de casos de uso: .....	25
Historias de usuario .....	27
<b>Hipótesis y restricciones / Alcance</b> .....	<b>28</b>
Requisitos no funcionales .....	28
Restricciones del proyecto.....	28
Restricciones tecnológicas.....	28
Restricciones de tiempo y presupuesto .....	29
Impacto esperado .....	30
Impacto en los usuarios .....	30
Impacto en el sector educativo.....	30
<b>Estudio de alternativas y viabilidad</b> .....	<b>31</b>
Análisis de herramientas y tecnologías .....	31
Viabilidad y formas de monetización.....	34
<b>Descripción de la solución propuesta</b> .....	<b>35</b>
<b>Análisis de riesgos</b> .....	<b>44</b>
Análisis DAFO.....	44
<b>Organización y gestión del proyecto</b> .....	<b>45</b>
<b>Organización</b> .....	<b>45</b>
Arquitectura del sistema .....	45
Componentes .....	48

Desarrollo de la solución propuesta .....	50
Gestión del proyecto .....	52
Recursos utilizados .....	52
Planificación temporal y diagrama de Gantt .....	52
Conclusiones y trabajo futuro .....	54
Referencias.....	56
Anexos .....	60

## Índice de anexos

<b>Anexo 1: Especificaciones del sistema.....</b>	<b>60</b>
Requisitos funcionales del sistema .....	60
Requisitos no funcionales .....	81
Requisitos de información .....	87
Diagramas de casos de uso.....	90
Jerarquía de actores .....	92
Historias de usuario.....	92
<b>Anexo 2: Análisis y Diseño del sistema .....</b>	<b>96</b>
Modelo de dominio .....	96
Vista de arquitectura.....	96
Realización casos de uso-análisis .....	97
Arquitectura de capas .....	101
Diagrama de clases real .....	102
Casos de uso-diseño .....	103
Modelo de despliegue.....	107
Esquema de la base de datos .....	107
<b>Anexo 3: Estimación del tamaño y esfuerzo.....</b>	<b>112</b>
Factores de complejidad técnica.....	112
Factores de complejidad del entorno .....	113
<b>Anexo 4: Plan de seguridad.....</b>	<b>115</b>

## Índice de figuras

Introducción .....	12
Objeto.....	14
Antecedentes .....	15
Descripción de la situación actual .....	17
Figura 1. Interfaz de la aplicación web Lucidchart.....	17
Figura 2. Ejemplo de un diagrama de clases generado por ChatGPT-4.....	18
Normas y referencias .....	19
Figura 3. Tipos de modelos de IA y sus características. Fuente: <a href="https://openai.com/">https://openai.com/</a> .....	22
Figura 4. Características de los modelos que ofrece OpenAI. Fuente: <a href="https://openai.com/">https://openai.com/</a> .....	22

Figura 5. Tiempo de ChatGPT en llegar al millón de usuarios. Fuente: <a href="https://www.digitalinformationworld.com/">https://www.digitalinformationworld.com/</a> .....	22
Figura 6. Wireframes de la aplicación web .....	23
Figura 7. Prototipo digital de la aplicación web .....	24
Figura 8. Componentes del prototipo digital .....	24
<b>Requisitos iniciales .....</b>	<b>25</b>
Figura 9. Diagrama de casos de uso de las conversaciones .....	26
Figura 10. Diagrama de casos de uso de la gestión de usuarios .....	26
Figura 11. Jerarquía de actores en los diagramas de casos de uso .....	27
<b>Hipótesis y restricciones / Alcance.....</b>	<b>28</b>
Figura 12. Ejemplo de uso de la herramienta “Tokenizer” Fuente: <a href="https://platform.openai.com/tokenizer">https://platform.openai.com/tokenizer</a> .....	29
Figura 13. Gráfica del porcentaje de docentes a favor de la IA. Fuente: (Florencia Melo, 2023) .....	30
<b>Estudio de alternativas y viabilidad .....</b>	<b>31</b>
Figura 14. Ejemplo código PlantUML y su diagrama de clases .....	33
<b>Descripción de la solución propuesta .....</b>	<b>35</b>
Figura 15. Página de inicio de la aplicación.....	35
Figura 16. Formulario de registro .....	35
Figura 17. Contenido del correo electrónico de verificación .....	36
Figura 18. Formulario de inicio de sesión.....	36
Figura 19. Eliminación de la cuenta de un usuario .....	37
Figura 20. Cambio del rol de un usuario .....	37
Figura 21. Comparación del menú entre usuarios con distintos roles .....	38
Figura 22. Solicitud de un enunciado al modelo .....	39
Figura 23. Solicitud de un diagrama de clases al modelo .....	39
Figura 24. Solicitud de una explicación al modelo.....	40
Figura 25. Elección de una imagen para enviar al chat .....	40
Figura 26. Solicitud del análisis de una imagen .....	41
Figura 27. Actualización de la barra de mensajes al solicitar realimentación..	41
Figura 28. Actualización de la lista de chats al solicitar feedback .....	42
Figura 29. Actualización de la lista de chats al mandar feedback .....	42
Figura 30. Ventana de ajustes .....	42
Figura 31. Confirmación del cierre de sesión .....	42
Figura 32. Esquema BPMN de una interacción normal entre el usuario y la IA	43
Figura 33. Esquema BPMN cuando el usuario sube una imagen a la aplicación .....	43
La aplicación se ha desplegado en un servidor del equipo de investigación GRIAL ( <a href="https://d-ia-gramas.grial.eu/">https://d-ia-gramas.grial.eu/</a> ) y se ha hecho un vídeo demostrativo en el	

que se explican y se pueden ver todas las funcionalidades ( <a href="https://youtu.be/gpEW9jzUqRw">https://youtu.be/gpEW9jzUqRw</a> ).....	43
<b>Análisis de riesgos .....</b>	<b>44</b>
Figura 34. Matriz del análisis DAFO de la aplicación .....	45
<b>Organización y gestión del proyecto .....</b>	<b>45</b>
Figura 35. Comparación MVC y MVT. Fuente: <a href="https://sahiljaggarwal.hashnode.dev/mvc-vs-mvt-architecture">https://sahiljaggarwal.hashnode.dev/mvc-vs-mvt-architecture</a> .....	46
Figura 36. Arquitectura de Django. Fuente: <a href="https://dev.to/sm0ke/django-tutorial-mvt-architecture-custom-commands-19nb">https://dev.to/sm0ke/django-tutorial-mvt-architecture-custom-commands-19nb</a> .....	46
Figura 37. Esquema de la base de datos del proyecto generada por el IDE ....	48
Figura 38. Comparación de HTTP y HTTPS. Fuente: (Hostilica, 2024) .....	49
Figura 39. Esquema del funcionamiento de ChatGPT. Fuente: (Cuofano, 2024) .....	50
Figura 40. Primera parte del diagrama de la planificación temporal y el diagrama de Gantt .....	53
Figura 41. Segunda parte del diagrama de la planificación temporal y el diagrama de Gantt .....	53
<b>Conclusiones y trabajo futuro .....</b>	<b>54</b>
<b>Referencias.....</b>	<b>56</b>
<b>Anexos .....</b>	<b>60</b>
Figura 42. Diagrama de casos de uso del paquete “Gestión de usuarios” .....	91
Figura 43. Diagrama de casos de uso del paquete “Conversaciones” .....	91
Figura 44. Jerarquía de actores del sistema .....	92
Figura 45. Diagrama de clases del sistema.....	96
Figura 46. Vista de arquitectura del modelo de análisis.....	96
Figura 47. Caso de uso-análisis “Registrarse” .....	97
Figura 48. Caso de uso-análisis “Iniciar sesión” .....	97
Figura 49. Caso de uso-análisis “Cerrar sesión” .....	98
Figura 50. Caso de uso-análisis “Cambiar rol” .....	98
Figura 51. Caso de uso-análisis “Eliminar cuenta” .....	99
Figura 52. Caso de uso-análisis “Modificar datos” .....	99
Figura 53. Caso de uso-análisis “Iniciar conversación” .....	100
Figura 54. Caso de uso-análisis “Eliminar conversación” .....	100
Figura 55. Caso de uso-análisis “Ver conversaciones” .....	101
Figura 56. Arquitectura de capas del modelo de diseño .....	101
Figura 57. Diagrama de clases real “Gestión de usuarios” .....	102
Figura 58. Diagrama de clases real “Gestión de conversaciones” .....	102
Figura 59. Caso de uso-diseño “Registrarse” .....	103
Figura 60. Caso de uso-diseño “Iniciar sesión” .....	103



Figura 61. Caso de uso-diseño “Cerrar sesión” .....	104
Figura 62. Caso de uso-diseño “Recuperar contraseña” .....	104
Figura 63. Caso de uso-diseño “Recuperar contraseña” .....	105
Figura 64. Caso de uso-diseño “Modificar datos” .....	105
Figura 65. Caso de uso-diseño “Ver perfil” .....	106
Figura 66. Caso de uso-diseño “Iniciar conversación” .....	106
Figura 67. Caso de uso-diseño “Eliminar conversación” .....	106
Figura 68. Caso de uso-diseño “Enviar chat para feedback” .....	107
Figura 69. Diagrama de despliegue de la aplicación.....	107
Figura 70. Esquema de la base de datos de la aplicación.....	111
Figura 71. Estimación de esfuerzo en el programa EZEstimate .....	114
Figura 72. Contraseña codificada .....	116
Figura 73. Certificado válido de una conexión HTTPS. ....	116

## Índice de tablas

<b>Anexo 1: Especificaciones del sistema.....</b>	<b>60</b>
<b>Requisitos funcionales del sistema .....</b>	<b>60</b>
Tabla 1. Caso de uso “Registrarse” .....	61
Tabla 2. Caso de uso “Iniciar sesión” .....	62
Tabla 3. Caso de uso “Ver perfil” .....	63
Tabla 4. Caso de uso “Cerrar sesión” .....	64
Tabla 5. Caso de uso “Asignar rol de profesor” .....	65
Tabla 6. Caso de uso “Dar de baja un usuario” .....	66
Tabla 7. Caso de uso “Recuperar contraseña” .....	67
Tabla 8. Caso de uso “Enviar email” .....	68
Tabla 9. Caso de uso “Modificar datos” .....	69
Tabla 10. Caso de uso “Mostrar información” .....	70
Tabla 11. Caso de uso “Iniciar conversación” .....	71
Tabla 12. Caso de uso “Eliminar conversación” .....	73
Tabla 13. Caso de uso “Guardar conversación” .....	73
Tabla 14. Caso de uso “Generar respuesta” .....	74
Tabla 15. Caso de uso “Generar enunciado” .....	75
Tabla 16. Caso de uso “Generar diagrama” .....	76
Tabla 17. Caso de uso “Enviar chat para <i>feedback</i> ” .....	78
Tabla 18. Caso de uso “Ver conversaciones” .....	78
Tabla 19. Caso de uso “Revisar chat y dar feedback” .....	79
Tabla 20. Caso de uso “Ver feedback” .....	80
<b>Requisitos no funcionales .....</b>	<b>81</b>
Tabla 21. Requisito no funcional “Usabilidad” .....	81
Tabla 22. Requisito no funcional “Rendimiento” .....	82
Tabla 23. Requisito no funcional “Escalabilidad” .....	83
Tabla 24. Requisito no funcional “Seguridad” .....	84
Tabla 25. Requisito no funcional “Compatibilidad” .....	85
Tabla 26. Requisito no funcional “Portabilidad” .....	86
<b>Requisitos de información .....</b>	<b>87</b>
Tabla 27. Requisito de información “Información de los usuarios” .....	87
Tabla 28. Requisito de información “Información de los chats” .....	88
Tabla 29. Requisito de información “Información de los mensajes” .....	89
<b>Diagramas de casos de uso .....</b>	<b>90</b>

<b>Jerarquía de actores .....</b>	<b>92</b>
<b>Historias de usuario.....</b>	<b>92</b>
Tabla 30. Historia de usuario “Registro en la aplicación” .....	93
Tabla 31. Historia de usuario “Inicio de sesión” .....	93
Tabla 32. Historia de usuario “Cerrar sesión” .....	93
Tabla 33. Historia de usuario “Iniciar una conversación” .....	94
Tabla 34. Historia de usuario “Ver las conversaciones” .....	94
Tabla 35. Historia de usuario “Dar de alta a un profesor” .....	94
Tabla 36. Historia de usuario “Dar <i>feedback</i> a la IA.....	95
Tabla 37. Historia de usuario “Eliminar una cuenta” .....	95
<b>Anexo 2: Análisis y Diseño del sistema .....</b>	<b>96</b>
<b>Modelo de dominio .....</b>	<b>96</b>
<b>Vista de arquitectura.....</b>	<b>96</b>
<b>Realización casos de uso-análisis .....</b>	<b>97</b>
<b>Arquitectura de capas .....</b>	<b>101</b>
<b>Diagrama de clases real .....</b>	<b>102</b>
<b>Casos de uso-diseño .....</b>	<b>103</b>
<b>Modelo de despliegue.....</b>	<b>107</b>
<b>Esquema de la base de datos .....</b>	<b>107</b>
Tabla 38. Tabla del usuario en la BD .....	108
Tabla 39. Tabla del mensaje en la BD.....	109
Tabla 40. Tabla del chat en la BD .....	110
<b>Anexo 3: Estimación del tamaño y esfuerzo.....</b>	<b>112</b>
<b>Factores de complejidad técnica.....</b>	<b>112</b>
<b>Factores de complejidad del entorno .....</b>	<b>113</b>
<b>Anexo 4: Plan de seguridad.....</b>	<b>115</b>

## Introducción

A la hora de adquirir conocimientos en el área de la Ingeniería de Software, uno de los conceptos clave por aprender es el modelo del dominio. El modelo de dominio es una forma de representación de objetos del mundo real a través de clases conceptuales (García-Peñalvo et al., 2024), en el proceso de diseño de un sistema software es de gran importancia el correcto entendimiento de este concepto y la abstracción de la información del dominio del problema para poder plantear una solución óptima. Una de las maneras de representar dicha solución es mediante UML (*Unified Modeling Language*), el cual es un lenguaje de modelado estándar que permite crear diagramas que describen la estructura y el comportamiento de los sistemas a través de clases que tienen atributos y/o operaciones y las respectivas relaciones entre cada clase (Booch et al., 2005). Esto permite tener una visión global de sistema y ver el comportamiento dinámico de los objetos que lo componen, para así poder identificar de manera visual posibles problemas que pueda llegar a tener dicho sistema y solucionarlos de manera efectiva.

Una de las principales dificultades a las que se enfrenta un docente en el momento de enseñar los principios del modelado de dominio en asignaturas como Ingeniería de Software, es la generación de enunciados lo suficientemente completos para que el estudiante ponga en práctica los conocimientos que ha adquirido y los resuelva mediante un diagrama de clases. La solución a dicho enunciado y dependiendo de la dificultad de este, debería de incluir tanto los distintos tipos de clase, como los distintos tipos de relaciones que hay entre cada clase y resulta complicado tener una visión amplia de un problema que requiere una solución de tales características. En el caso de los estudiantes, muchas veces pueden llegar a entender de manera general lo que se ha explicado, pero en el momento en el que se origina una duda específica puede resultar complicado encontrar enunciados que hagan hincapié en la misma.

Debido al auge y la creciente popularidad de la Inteligencia Artificial (IA) que ha experimentado la sociedad en estos últimos años, una de las principales herramientas con las que cuentan los estudiantes son los modelos de generación de texto como ChatGPT, que cuenta con una gran capacidad para proporcionar cualquier tipo de respuesta que se le solicite. Dependiendo del modelo de IA que se utilice, se le pueden hacer distintos tipos de peticiones, la más básica es la generación de texto, la cual incluye desde texto plano a código en un lenguaje específico entre muchas otras. También hay algunos que permiten el análisis de imágenes y facilitar una explicación de las mismas, por lo que, aplicando estas características como ayuda a la hora de enseñar los diagramas de clase UML se puede conseguir una herramienta que genere enunciados con las características que se le indique, proporcione respuestas concretas sobre dudas que tenga el usuario, generar código en un formato determinado para poder generar los diagramas de clase que servirán a los usuarios a la hora de comprender correctamente conceptos generales como las clases conceptuales u otros más específicos.

Actualmente hay aplicaciones web que permiten generar diagramas de clase a través de código en un formato determinado, otras que procesan el lenguaje natural y devuelven respuestas acordes con lo que se le solicite y otras que tienen la capacidad de crear imágenes interpretando una frase que proporciona el usuario. El principal problema que tienen las últimas es la poca exactitud con la que se ajustan a las peticiones que se le indican, ya que cuentan con cierto grado de imaginación (técnicamente reciben el nombre alucinaciones) que es problemático a la hora de generar diagramas de clase, los cuales tienen que ser exactos y muy precisos. Otro de los problemas que tienen las aplicaciones de diagramas de clase, es que no cuentan con ningún tipo de ayuda que permita al usuario saber la razón por la que el diagrama es de una manera u otra.

El siguiente documento se estructura en dos partes principales, la primera es la memoria la cual contiene una serie de capítulos en los que se explican aspectos como los antecedentes del proyecto, la descripción de la situación actual en la que se explican las competencias del trabajo, las normas y referencias que se han seguido para hacer el proyecto, los requisitos que han guiado las funcionalidades que tiene la aplicación. También hay apartados en los que se hace un estudio de las características del proyecto como la metodología que se han seguido, las herramientas necesarias para poder crear la parte de la aplicación web y otros capítulos orientados a la gestión y organización del proyecto.

La segunda parte cuenta con una serie de anexos en los que se explican las especificaciones, donde se detalla la documentación de los requisitos y las historias de usuario entre otras. En el segundo de los anexos se explica el análisis y diseño del sistema, en el que se incluyen los diagramas y arquitectura de trabajo. El tercer anexo cuenta con los diagramas UML que han sido necesarios al inicio para hacer la arquitectura del proyecto y el diagrama de la base de datos. Por último, en el cuarto anexo se explica el plan de seguridad, en el cual se incluyen aspectos del ámbito de la seguridad del proyecto como las leyes de protección de datos, la ética informática o la seguridad que tiene la propia aplicación web.

# Objeto

## Objetivos funcionales

El objetivo principal del proyecto es poder ayudar a los docentes a la hora de enseñar conceptos clave como los diagramas de clase UML en asignaturas como Ingeniería de Software a través de una aplicación web. Dentro del objetivo principal del sistema podemos encontrar una serie de sub-objetivos que se han tenido que alcanzar para poder desarrollar la aplicación por completo.

El primero de ellos es la capacidad por parte de los usuarios de poder generar enunciados de problemas que se resuelvan mediante un diagrama de clases UML. Para llevar a cabo esta tarea, ha sido necesario integrar un modelo de generación de texto como ChatGPT para, de esta manera, permitir a los usuarios interactuar con dicho modelo y obtener los enunciados. Otro de los sub-objetivos que se ha tenido que alcanzar ha sido el desarrollar una interfaz que permita al estudiantado proporcionar un diagrama de clase al modelo y obtener su análisis por parte de la IA.

El siguiente de los objetivos es el de generación de la imagen de los diagramas de clase UML por parte de la IA. Este punto ha sido clave en el desarrollo del proyecto ya que de por sí, el modelo de generación de imágenes con el que cuenta ChatGPT-4 interpreta de manera abstracta la petición del usuario a la hora de generar las imágenes. Estas herramientas cuentan con un grado de imaginación que actúa como un inconveniente a la hora de presentar los diagramas de clase, los cuales tienen que ser detallados y ceñirse al código.

Por último, ha sido necesario diseñar un sistema de realimentación que permita al estudiantado enviar cualquiera de los chats que tienen a un profesor para recibir comentarios sobre un determinado aspecto o para comprobar si la manera en la que está entendiendo los conceptos a la hora de resolver los enunciados es la correcta.

## Objetivos personales

Los objetivos personales que se buscan alcanzar mediante la realización de este trabajo, es el de desarrollar las capacidades de poder hacer una aplicación web completa que sirva de punto de partida para los proyectos futuros que surjan en el ámbito laboral y el de comprender todas las fases que puede llegar a tener el desarrollo de un proyecto. El hecho de que para este trabajo se haya hecho uso de la IA para hacer determinadas tareas, también implica un proceso de entendimiento del funcionamiento de los modelos de generación de texto y como se pueden aprovechar sus características para producir productos o aplicaciones innovadores que no existen todavía. También ha sido necesario adquirir conocimientos sobre cómo los modelos de IA interpretan las peticiones del usuario para obtener los resultados deseados.

Teniendo en cuenta que la aplicación servirá de ayuda a los docentes, otro de los objetivos es el de poder contribuir a la mejora del aprendizaje en asignaturas como Ingeniería del Software.

# Antecedentes

## Aspectos teóricos

Uno de los aspectos teóricos del trabajo es la didáctica de la Ingeniería de Software, en la cual se puede destacar la enseñanza de UML a los estudiantes del sector de la Ingeniería Informática. A la hora de formar a los alumnos, UML es una herramienta esencial que les permite modelar la arquitectura, funcionalidades y documentar un sistema software de manera visual. La manera de enseñar los conceptos que envuelven al lenguaje UML consta de una explicación inicial de los mismos y posteriormente, la realización de ejercicios prácticos para una correcta asimilación tanto de la teoría como de la práctica. Muchas veces los estudiantes no llegan a entender algunos términos o tipos de conceptos que pueden llegar a resultar complejos, lo que desemboca en una falta de base sólida de conocimiento para poder usar el lenguaje UML. Este tipo de problemas se podría resolver mediante una explicación detallada de los aspectos teóricos y a través de la realización de ejercicios que hagan hincapié en las dificultades que tenga el alumno. Para un docente puede llegar a ser complicado el diseño de enunciados que ayuden a cada uno de los alumnos que no entienda correctamente algo en concreto y a su vez tendría que disponer de una gran cantidad de tiempo para explicar a cada alumno que se presente con una duda teórica y asegurarse de que lo entiende.

La mayoría de las herramientas educativas que se utilizan para enseñar UML se centran principalmente, en proporcionar ejemplos muy generales de problemas, que no se adaptan a las necesidades individuales de los estudiantes, ni ofrecen retroalimentación personalizada. Esto resulta en una experiencia de aprendizaje que puede ser demasiado genérica y no suficientemente enfocada en las áreas donde los estudiantes experimentan mayores dificultades. Ante esta situación, las tecnologías emergentes en el campo de la IA presentan una oportunidad única para revolucionar la forma en que se enseña el lenguaje UML. Los sistemas basados en IA, como el que se propone en este trabajo, pueden diseñar ejercicios dinámicos y adaptativos que respondan en tiempo real a las interacciones del estudiante, identificando sus errores y ofreciendo explicaciones y correcciones personalizadas. Esto no solo ayuda a los estudiantes a comprender mejor los conceptos complejos, sino que también permite a los profesores optimizar su tiempo y recursos.

En relación con la utilización de la inteligencia artificial para la enseñanza y aprendizaje, es de gran importancia explorar cómo las tecnologías emergentes están siendo integradas en la educación, especialmente en disciplinas técnicas como la Ingeniería de Software y se está dejando a un lado la posible concepción de que la IA es un inconveniente en vez de una ventaja a la hora de que los estudiantes puedan sacar provecho de ella de la manera correcta.

## Grupo de Investigación GRIAL

Este proyecto se ha realizado en colaboración con el Grupo de Investigación GRIAL (<https://grial.usal.es/about>), el cual es un grupo de investigación con sede en Salamanca, España. Centrado principalmente en la interacción y eLearning, en el que se desarrollan soluciones innovadoras para la educación y la formación, el cual está formado por multitud de docentes pertenecientes a la Universidad de Salamanca entre mucho otros (García-Peñalvo et al., 2019). A continuación, se explican las líneas de investigación del equipo que más se ajustan al marco del trabajo.

La primera es la Calidad y Evaluación en Educación, la cual se enfoca en el desarrollo y la implementación de estándares y procedimientos para asegurar y mejorar la calidad de la educación, además de métodos para evaluar de manera efectiva los procesos y resultados educativos.

La siguiente es la Ingeniería Web y Arquitecturas software, en la que se estudian técnicas y metodologías para el desarrollo de aplicaciones web y la estructuración de software para asegurar la funcionalidad, mantenibilidad y escalabilidad.

Otra de las líneas de investigación son las Metodologías eLearning, donde se investiga sobre los métodos de enseñanza y aprendizaje a través de medio electrónicos, centrándose en optimizar el diseño pedagógico y la tecnología educativa para el aprendizaje en línea. Los Sistemas de Aprendizaje Interactivos se centran en el diseño y evaluación de sistemas que permiten interacción dinámica entre el estudiante y el contenido educativo, mejorando la experiencia de aprendizaje mediante la participación activa.

Por último, las Tecnologías de la Información y la Comunicación (TIC) e Innovación Educativa, en las que se explora la forma en la que las TIC y la comunicación pueden ser utilizadas para innovar en educación, transformando las prácticas pedagógicas y los entornos de aprendizaje.

En lo que respecta a los proyectos desarrollados por GRIAL, estos se pueden dividir en las siguientes escalas o ámbitos geográficos: Internacionales, europeos, Nacionales y Regionales.

Uno de los proyectos europeos desarrollados por el equipo de investigación es ENCORE (<https://project-encore.eu/>), el cual es un proyecto europeo destinado a implementar una fase cuantitativa/basada en datos y una fase cualitativa/impulsada por expertos para reutilizar Recursos Educativos Abiertos (REA) en habilidades Digitales, Verdes y Empresariales (GDE), y apoyar a los educadores en el diseño de enseñanza y aprendizaje (Bucchiarone et al., 2023). Otro de los proyectos a escala europea es GAMIGRATION (<https://gamigration.eu/>), un proyecto que ha explorado y demostrado cómo la educación gamificada puede ser una herramienta eficaz para facilitar la integración de personas inmigrantes en el sistema escolar (García-Holgado et al., 2022).

# Descripción de la situación actual

## Análisis de la competencia

Actualmente hay sitios web como Visual Paradigm Online o LucidChart (ver Figura 1) que permiten la creación de cualquier tipo de diagramas desde cero. La principal diferencia que hay entre estas aplicaciones y la que se ha desarrollado para este trabajo es la forma en la que se generan los diagramas, ya que se sirven de una interfaz diseñada específicamente para que el usuario pueda interactuar con una serie de formas que pueden editar y unir mediante las relaciones que ellos deseen. El gran inconveniente de esta forma de crear los diagramas radica en la necesidad de que el usuario tenga cierto grado de conocimiento de los conceptos teóricos necesarios para la parte práctica de UML, ya que no recibirá ningún tipo de comentario y/o ayuda personalizada para poder corregir el o los posibles errores que haya tenido a la hora de hacer el diagrama. El hecho de no haber entendido bien la parte teórica y la acción repetitiva de aplicar ese error en la parte práctica puede dificultar la corrección y la asimilación de lo que sí que es correcto por parte de estudiante en el futuro. Aunque cometer errores puede favorecer el desarrollo del estudiante a la hora de solucionar determinados problemas, se necesita una corrección lo más pronta posible para que las ideas mal concebidas se cambien por las que sí que son correctas (Giraldez, 2024).

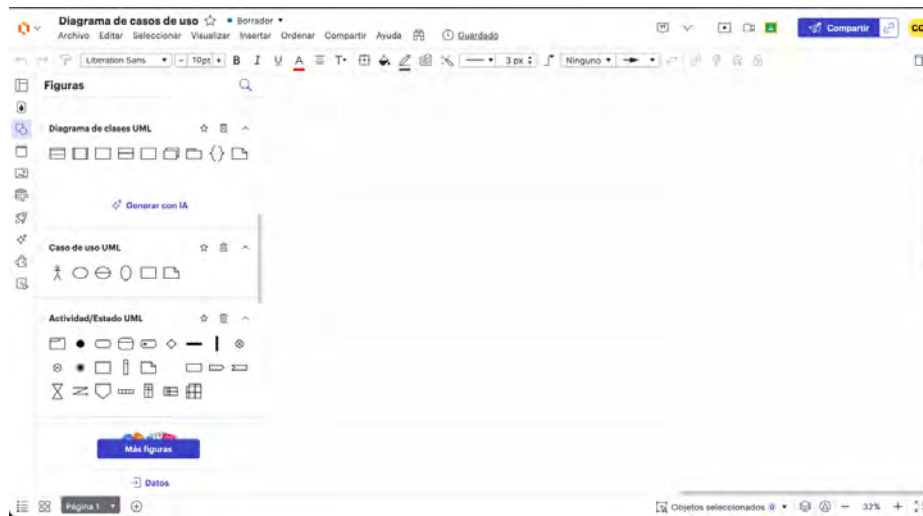


Figura 1. Interfaz de la aplicación web Lucidchart

La generación de imágenes por medio de texto es una de las características que pueden hacer determinados modelos de IA, como por ejemplo ChatGPT-4 con la ayuda de DALL-E (<https://platform.openai.com/docs/models/dall-e>), el cual es un sistema de inteligencia artificial que puede crear imágenes y arte realistas a partir de una descripción en lenguaje natural. Aunque tiene una gran capacidad de abstracción a la hora de producir las imágenes, también cuenta con un parámetro de imaginación o temperatura (<https://platform.openai.com/docs/guides/text-generation/how-should-i-set-the-temperature-parameter>) que actúa como un inconveniente en el caso de la aplicación que se ha desarrollado. En la Figura 2 se puede ver un ejemplo de un



diagrama de clases generado por ChatGPT 4 y que ilustra perfectamente la problemática explicada.

Teniendo en cuenta la desventaja que se ha explicado anteriormente, se ha optado por combinar la capacidad de generación de texto de ChatGPT 3.5 y la librería de Python llamada PlantUML, la cual es una herramienta de código abierto diseñada para permitir a los usuarios crear diagramas a partir de un lenguaje de descripción textual simple. Se utiliza para representar diagramas UML como diagramas de clases, de secuencia, de casos de uso, de actividades y de componente entre otros. La herramienta convierte las descripciones escritas en un formato específico en imágenes visuales de diagramas, facilitando así la documentación y el diseño de sistemas de software (PlantUML, 2024). De esta manera se ha conseguido que los diagramas que se generan sean precisos y libres del parámetro de temperatura anteriormente explicado.



Figura 2. Ejemplo de un diagrama de clases generado por ChatGPT-4

El hecho de disponer de una plataforma en la que se pueda solicitar realimentación por parte del profesor es una manera rápida y efectiva de solventar las posibles dudas y problemas de comprensión de determinados conceptos que pueda llegar a tener un estudiante. Una de las aplicaciones que se usa para este fin es Moodle, ya que aparte de ser la plataforma principal que utiliza la Universidad de Salamanca (USAL) para gestionar a los alumnos, los cursos y las diferentes titulaciones que hay disponibles, también cuenta con un apartado de “foros” en los que cada estudiante puede preguntar cualquier tipo de duda dependiendo de cada asignatura. El problema con este tipo de foros es que el alumnado tiene cierta dificultad a la hora de plantear sus dudas y muchas veces prefieren enviar correos electrónicos en lugar de preguntar a través de los medios que se les proporciona. Esto se puede deber a la comodidad y familiaridad que tienen con el uso de correos electrónicos o la posible falta de conciencia sobre la función de los foros en el entorno de aprendizaje (Trivedi, 2023). Para poder solventar este posible problema se ha desarrollado un sistema de realimentación entre profesores y estudiantes para poder recibir comentarios sobre los chats que se deseen de una manera más individual que los foros.

# Normas y referencias

## Métodos

Para la gestión del proyecto, se ha adoptado la metodología SCRUM. Este tipo de marco de trabajo ágil ha sido ampliamente utilizado en el desarrollo de software, aunque también se aplica en otros campos donde los proyectos requieren flexibilidad y respuestas rápidas a cambios frecuentes. Se centra en la colaboración, la adaptabilidad y el ciclo iterativo de mejoras a través de *sprints* o iteraciones cortas (Schwaber, 1997).

A continuación, se explican de manera breve las características principales de esta metodología:

- **Iterativa e Incremental:** SCRUM permite que los equipos trabajen en ciclos cortos y controlados llamados *sprints*. Al final de cada sprint, el equipo produce un incremento del producto que es entregable y está en condiciones de poder usarse.
- **Roles Definidos:** hay 3 roles principales en SCRUM
  - **Product Owner (PO):** Representa los intereses de los *stakeholders* o partes interesadas y es responsable de definir las características del producto y priorizar las tareas pendientes de mayor importancia.
  - **Scrum Máster:** Actúa como un intermediario entre el equipo de desarrollo y el PO, asegurándose de que el proceso SCRUM se siga correctamente, y ayudando a resolver los obstáculos que puedan surgir.
  - **Equipo de Desarrollo:** Un grupo de profesionales que realizan el trabajo de entregar el producto.

Como cualquier metodología, SCRUM cuenta con una serie de fases que se hacen a lo largo del ciclo de desarrollo del producto:

- **Planificación del *sprint* (*sprint planning*):**
  - En esta fase, el equipo se reúne y selecciona las tareas que serán completadas durante el próximo sprint. Estas tareas se desglosan y planifican en detalle.
- **Desarrollo del *sprint*:**
  - Durante cada sprint, el equipo trabaja en las tareas que hay que desarrollar en ese momento. El progreso es revisado diariamente en reuniones breves llamadas *Daily Scrums*, donde cada miembro del equipo informa sobre lo que hizo el día anterior, lo que hará hoy y cualquier problema que pueda perjudicar su progreso.
- **Revisión del *sprint* (*sprint review*):**
  - Al final del sprint, el equipo presenta el trabajo realizado. Esta es una demostración del progreso y sirve para recibir retroalimentación que puede influir en el próximo ciclo.
- **Retrospectiva del *sprint* (*sprint retrospective*):**
  - Después de la revisión y antes de la siguiente planificación del sprint, el equipo se reúne para discutir qué ha ido bien, qué podría mejorarse y cómo implementar esas mejoras en los próximos *sprints*.

(García-Peñalvo et al., 2021).

## Herramientas

### Herramientas de implementación

Para la implementación de este trabajo se han utilizado diferentes herramientas como por ejemplo PyCharm, el cual es un Entorno de Desarrollo Integrado (IDE) que pertenece a la empresa JetBrains y que está diseñado específicamente para el lenguaje de programación Python (<https://www.jetbrains.com/es-es/pycharm/>), en el que se ha desarrollado toda la parte *backend* de la aplicación web.

Visual Studio Code ha sido otro de los IDE que se ha utilizado para desarrollar la parte *frontend* del trabajo, ya que cuenta con un *marketplace* de extensiones que añade multitud de funcionalidades innovadoras que otros editores de código no tienen (<https://code.visualstudio.com/>).

Como se ha explicado anteriormente, el *framework* que se ha utilizado para la aplicación ha sido Django, aparte de ser gratuito y de código abierto, fomenta un desarrollo rápido y un diseño limpio y pragmático. Algunas de las principales características son su seguridad, rapidez y escalabilidad (<https://www.djangoproject.com/>).

En lo que respecta a las herramientas de control de versiones se ha hecho uso de Github (<https://github.com/>), siendo esta una plataforma de desarrollo colaborativa que permite alojar proyectos de cualquier tipo en una serie de directorios virtuales o repositorios, que utiliza el sistema de control de versiones Git. Github permite clonar un directorio local a uno remoto para acceder a él desde cualquier ordenador pudiendo visualizar y descargar su contenido. Por otra parte, Git es el que gestiona las versiones de un proyecto a través de copias y actualizaciones del mismo (<https://www.git-scm.com/>).

Para el diseño de la interfaz se ha utilizado el software Figma (<https://www.figma.com/es-es/>), una herramienta de diseño en la que se puede colaborar con varias personas del mismo equipo. Permite diseñar componentes de la interfaz de la aplicación web, añadir efectos en las transiciones y visualizar de manera fácil y rápida, cómo sería la interacción del usuario con la aplicación final a través de las relaciones entre componentes y pantallas que se pueden hacer. Con el modo “Dev” que ha incorporado recientemente Figma, se pueden obtener los atributos de cada elemento que se haya diseñado en lenguaje CSS para pasarlos directamente al proyecto.

### Herramientas de comunicación y almacenamiento

Como herramienta de comunicación y control del progreso del proyecto, tanto los tutores como el alumno se han comunicado a través de emails o a través de la aplicación Slack, utilizada por un gran número de empresas y que permite el envío de mensajes, reuniendo a las personas que trabajan del mismo equipo en un solo sitio (<https://slack.com/intl/es-es>).

Otra plataforma que se ha utilizado en este sector ha sido Google Drive (<https://www.google.com/intl/es-es/drive/>), almacenando los archivos y diagramas de relevancia que necesitaban una verificación por parte de los tutores en una sola carpeta compartida.

## Herramientas para la elaboración del TFG

Para la redacción de este trabajo se ha utilizado principalmente Microsoft Word (<https://www.microsoft.com/es-es/microsoft-365/word>), el cual permite redactar documentos, insertar imágenes, generar índices de forma automática y aplicar diferentes estilos al texto entre muchas otras cosas más. Para almacenar las páginas referenciadas y para hacer la bibliografía se ha utilizado Zotero (<https://www.zotero.org/>), una aplicación que cuenta con una extensión para navegadores como Google Chrome que permite de manera fácil y rápida guardar y organizar las páginas visitadas incluyendo información relevante como el título, el tipo de elemento, la URL y la fecha en la que se ha accedido a ella.

## Modelos

Como modelo de inteligencia artificial se ha recurrido a los de la empresa OpenAI como ChatGPT-3.5 Turbo y ChatGPT-4 Turbo. Entre los modelos que han desarrollado se encuentran los nombrados anteriormente y, el más reciente ChatGPT-4o, el cual incorpora las funcionalidades del más avanzado y grande (ChatGPT-4 Turbo), pero siendo más rápido y accesible. Cabe destacar la rapidez con la que la empresa ha conseguido llegar al millón de usuarios (ver Figura 5) debido a la popularidad y el carácter innovador que suponen este tipo de herramientas, superando a las principales plataformas y aplicaciones que ya llevan un periodo de tiempo considerable operando.

Cada petición que se les haga consume una cierta cantidad de *tokens*, los cuales equivalen a una cierta cantidad de dinero real dependiendo de qué modelo se esté usando y la hora de que generen la respuesta también consumen un número de *tokens*. A continuación, se detallan los precios de cada modelo para cada millón de *tokens* (<https://openai.com/api/pricing/>):

- **ChatGPT-3.5 Turbo:**
  - Input: 0.50\$
  - Output: 1.50\$
  
- **ChatGPT-4 Turbo:**
  - Input: 10\$
  - Output: 30\$
  
- **ChatGPT-4o:**
  - Input: 5\$
  - Output: 15\$

El primero de ellos ha servido para hacer la generación de código PlantUML y las respuestas en formato de texto al usuario, mientras que ChatGPT-4 se ha utilizado para el análisis de las imágenes que puede subir el usuario a través de la interfaz. Este tipo de modelos pueden resultar algo difíciles de utilizar a través de código, por lo que se ha tenido que recurrir a la documentación de la API (*Application Programming Interface*) que hay disponible en su página web.

En la Figura 3 se muestran capturas de la página oficial de OpenAI (<https://openai.com/>) en las que se puede observar las características de cada modelo, así como el precio de los tokens de cada uno. En la Figura 4 se

presentan los servicios que pueden brindar los modelos como, por ejemplo: la creación de imágenes, la posibilidad de transformar el texto a voz (*Text-to-Speech*), el análisis de imágenes y la transcripción de audio a texto.

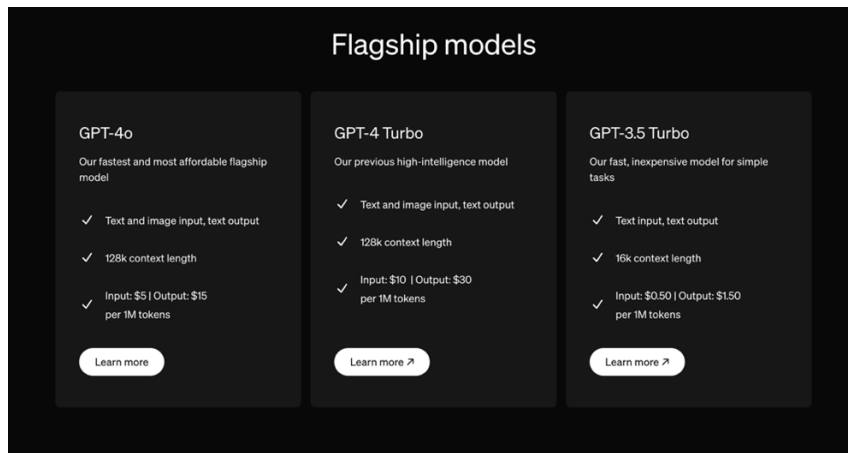


Figura 3. Tipos de modelos de IA y sus características. Fuente: <https://openai.com/>

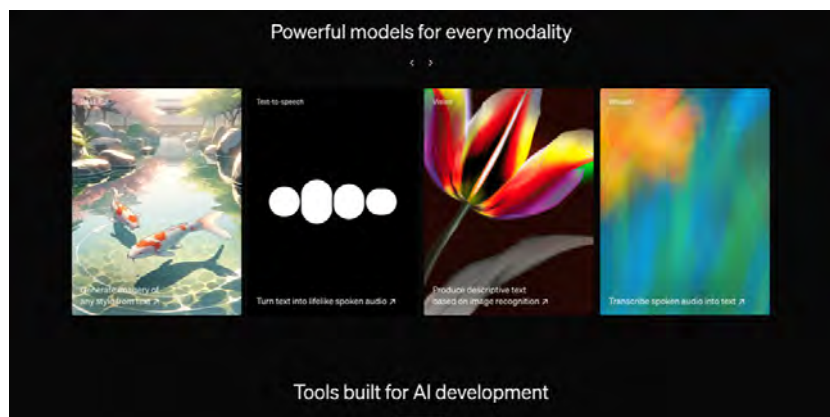


Figura 4. Características de los modelos que ofrece OpenAI. Fuente: <https://openai.com/>

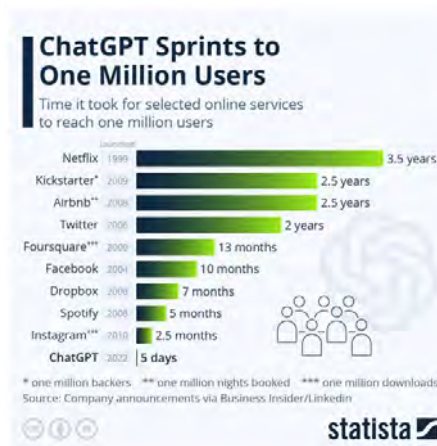


Figura 5. Tiempo de ChatGPT en llegar al millón de usuarios. Fuente: <https://www.digitalinformationworld.com/>

## Prototipos

### Wireframes

Los *wireframes* son diseños de la aplicación que se hacen al inicio para tener una idea global de cómo va a ser y el flujo de interacción que tendrá. No tienen ningún tipo de color o imágenes.

La Figura 6 pertenece a los *wireframes* de las distintas páginas que tiene la aplicación (<https://www.figma.com/design/h2b6re36WWRgMbz6Uab4re/TFG-Wireframe-2?node-id=0-1&t=89lCaNsm4HZpK5j3-1>). Entre ellas se incluye la página de inicio, la página de registro e inicio de sesión, la página principal, la página que contiene la lista de chats del usuario y un diseño inicial del sistema de realimentación que iba a tener en un principio. Al ser un diseño inicial y previo al desarrollo del código y la implementación, ha sufrido cambios significativos que se verán a lo largo de la memoria:

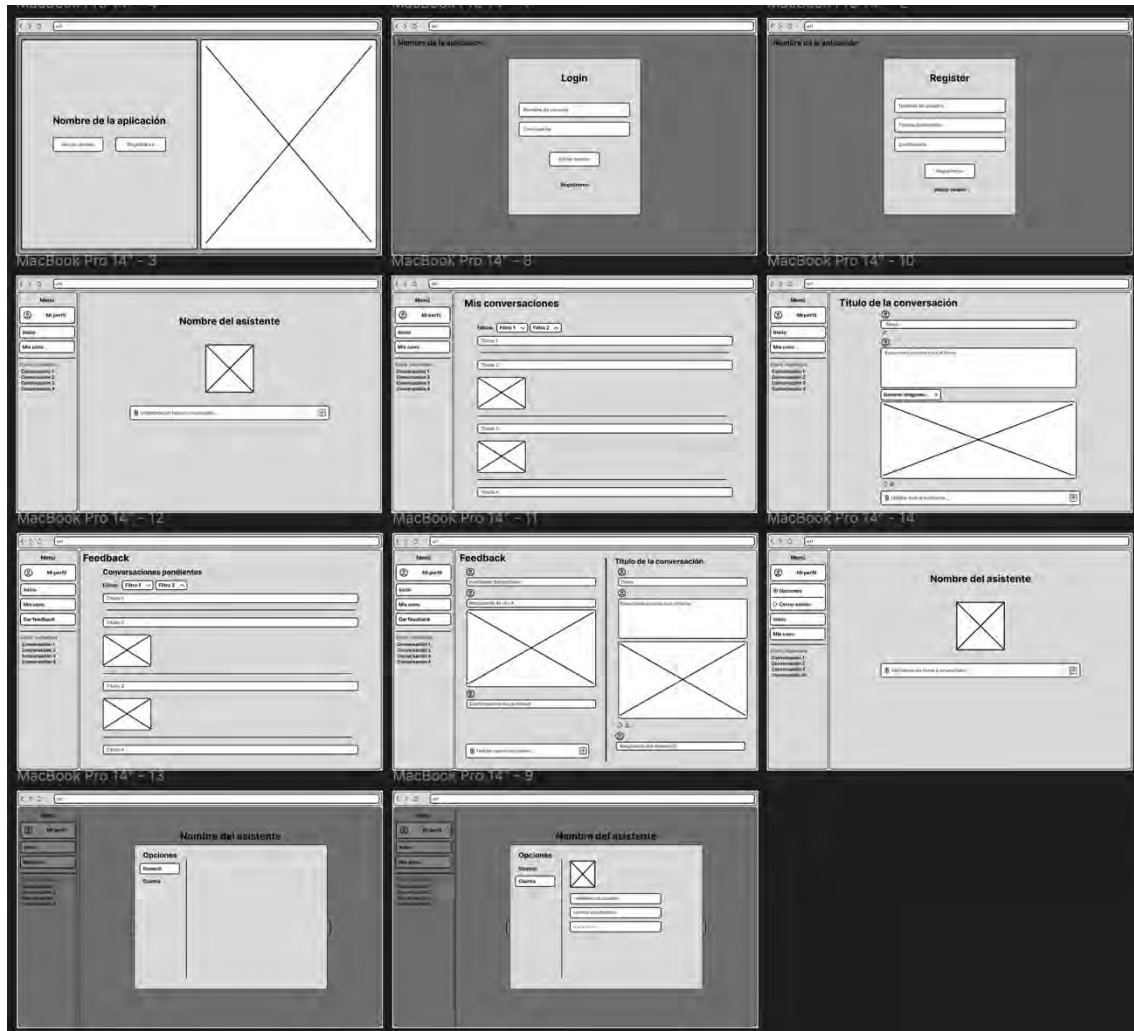


Figura 6. Wireframes de la aplicación web

## Mockups o prototipos digitales

En fases más avanzadas en el desarrollo de la aplicación se han tenido que hacer una serie de *mockups* o prototipos digitales en Figma de cada una de las páginas para poder ver el flujo de interacción y tener una idea del diseño de cada una (<https://www.figma.com/design/nVTd4nLNDIK6JLRaZ5Xzp8/Prototipo-digital?node-id=0-1&t=KWNxHhRAaMewxeSK-1>). En este tipo de prototipos ya se puede incluir colores, animaciones o imágenes en el caso de que sea necesario. Para poder ver los diferentes estados de la aplicación se ha tenido que hacer un conjunto de componentes como el menú lateral o las transiciones de los botones (ver Figura 7 y Figura 8).

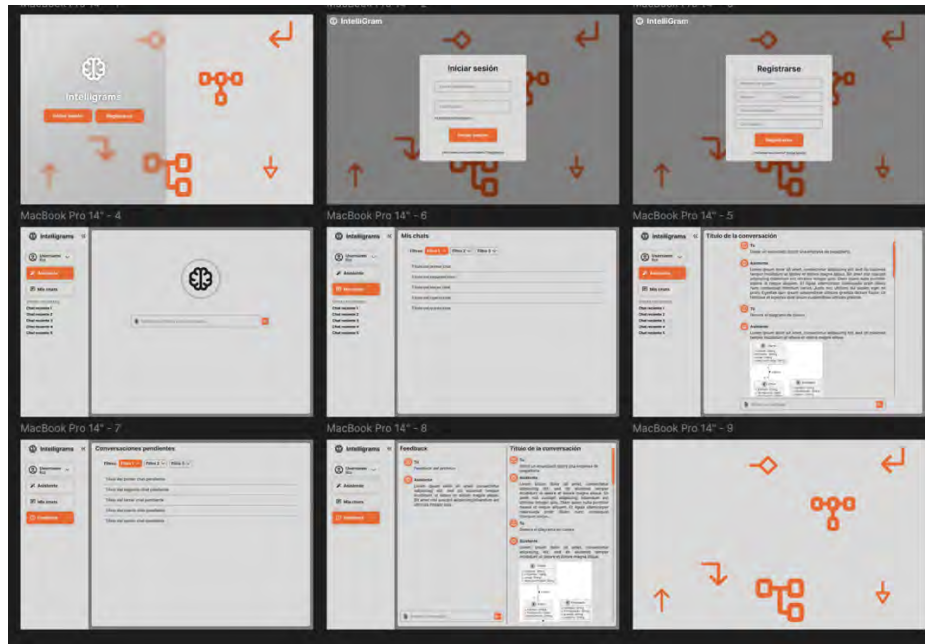


Figura 7. Prototipo digital de la aplicación web

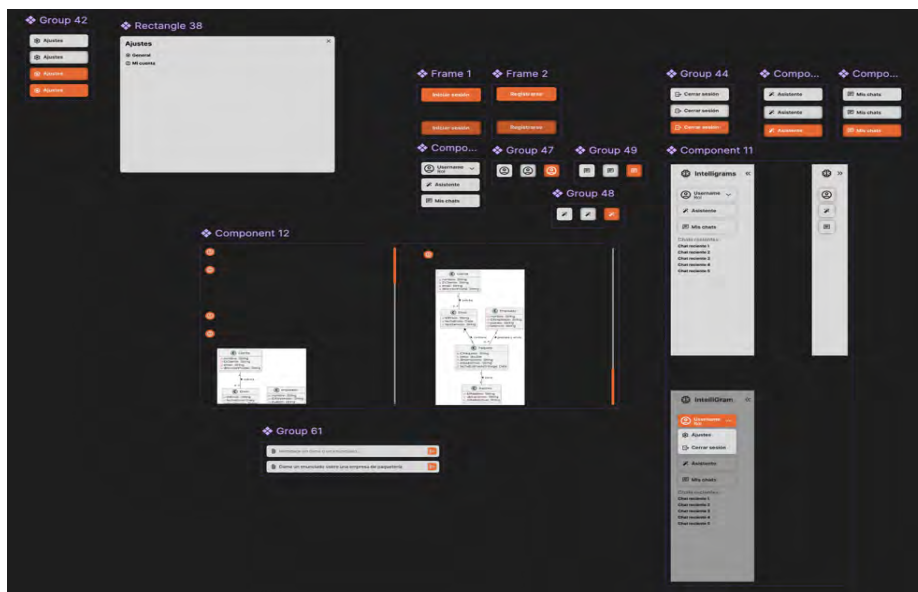


Figura 8. Componentes del prototipo digital

## Requisitos iniciales

A la hora de definir la propuesta se han tenido que presentar una serie de objetivos funcionales o requisitos principales que iba a tener la aplicación y que se exponen brevemente a continuación:

- **Generación de problemas de modelado de dominio mediante IA:**
  - Este requisito hace referencia a la necesidad de que el modelo de IA que se utilice pueda generar enunciados de problemas cuya solución sea un diagrama de clases UML.
- **Desarrollar una interfaz que permite al estudiantado plantear diagramas de clases en UML para solucionar problemas de modelado de dominio generados por IA:**
  - En este requisito se expresa el hecho de que los alumnos podrán proporcionar los diagramas de clase que hagan al modelo, para que este los analice y proporcione una respuesta.
- **Generación de diagramas de clases en UML para resolver los problemas previamente generados:**
  - Posibilidad por parte de la IA, de generar las imágenes de los diagramas de clase que sirven de solución a problemas planteados previamente.
- **Adaptación del sistema basado en IA mediante la realimentación del profesorado para mejorar el modelado de diagramas de clases en UML:**
  - Este requisito indica que el modelo aprenderá de la realimentación del profesorado y modificará su comportamiento para mejorar determinados aspectos a la hora de generar los diagramas de clase UML.

En el Anexo 1 se pueden encontrar las tablas (ver Tablas 1 - 20) pertenecientes a los requisitos funcionales que se han identificado a partir de los requisitos iniciales de la propuesta y los cuales se ha usado para hacer los diagramas de casos de uso (ver Figuras 42 y 43)

### Diagramas de casos de uso:

Los diagramas de caso de uso UML muestran el comportamiento que se espera de un sistema o software en un caso de uso concreto el cual se puede relacionar con los distintos actores del sistema o con otros casos de uso. Las siguientes imágenes (Figura 9 y 10) pertenecen a los diagramas de casos de uso que se han hecho en base a los requisitos iniciales y la jerarquía de actores (ver Figura 11), los cuales se han organizado en 2 paquetes principales (Conversaciones y Gestión de usuarios) en los que se puede observar las interacciones de los actores del sistema con los requisitos descritos anteriormente y otros que han surgido como necesidad para algunos aspectos esenciales de la funcionalidad del sistema.

A medida que se ha ido desarrollando la aplicación el único requisito que ha cambiado con respecto a los descritos al inicio de este apartado, ha sido el relativo a la modificación del comportamiento de modelo a través de la retroalimentación de los docentes. Se ha abandonado la idea de cambiar el comportamiento de modelo por parte de los docentes y se ha optado por un



sistema de realimentación entre profesores y alumnos para ayudar en las posibles dudas puedan llegar a tener a la hora de estudiar esta serie de conceptos teórico-prácticos de una forma más directa y rápida.

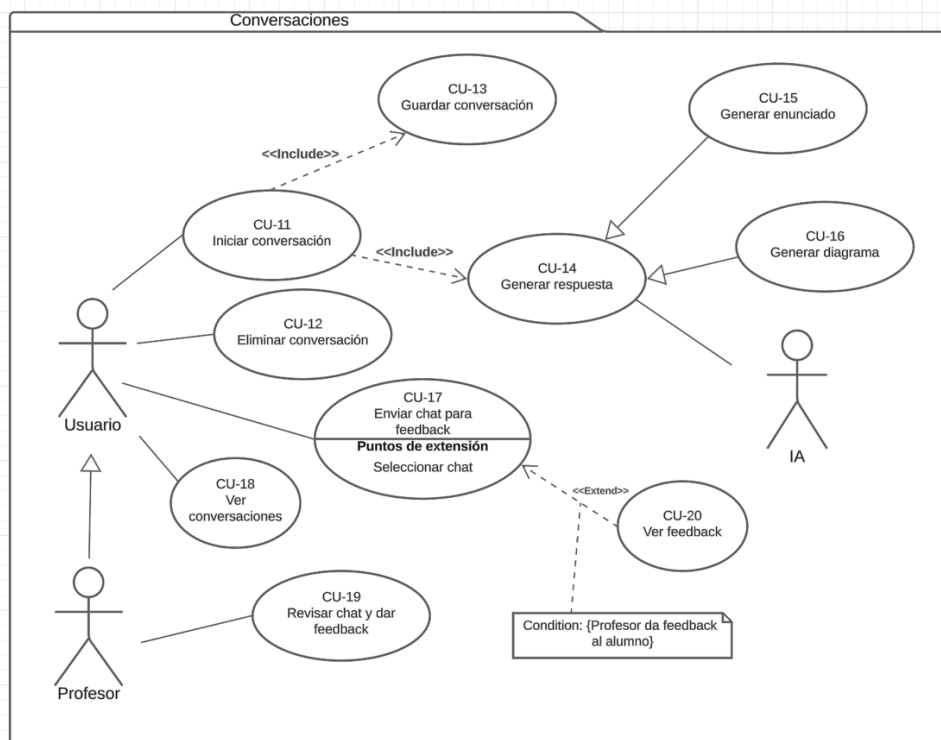


Figura 9. Diagrama de casos de uso de las conversaciones

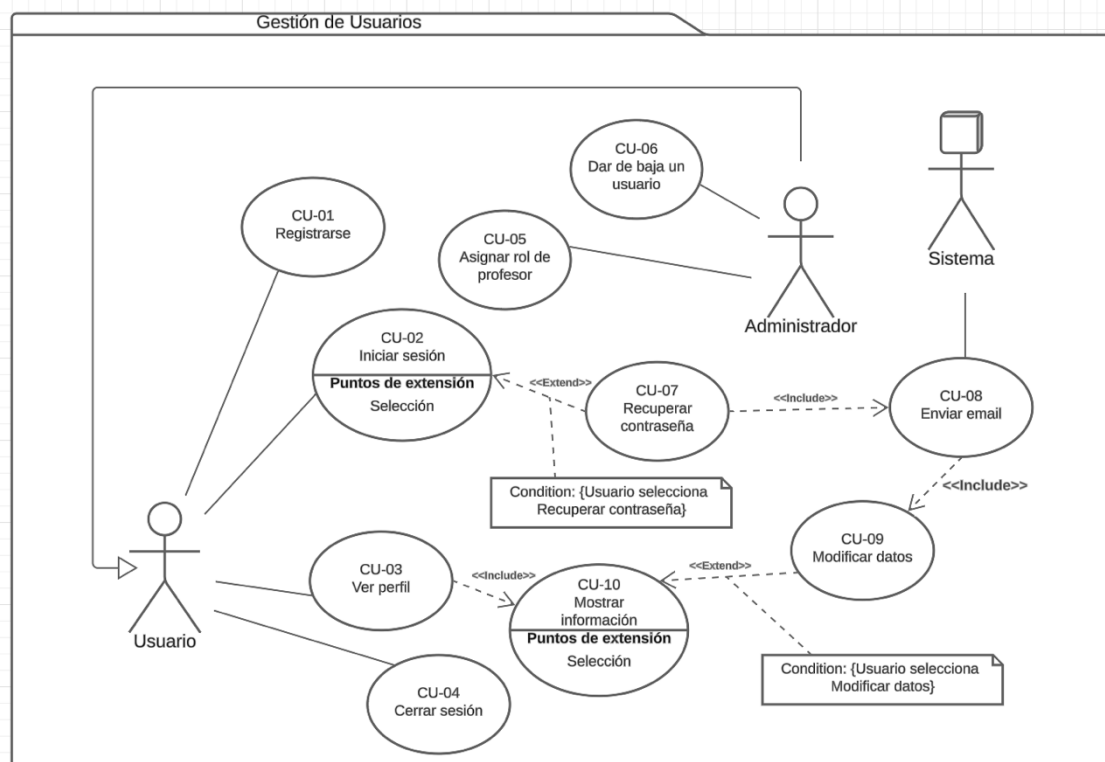


Figura 10. Diagrama de casos de uso de la gestión de usuarios

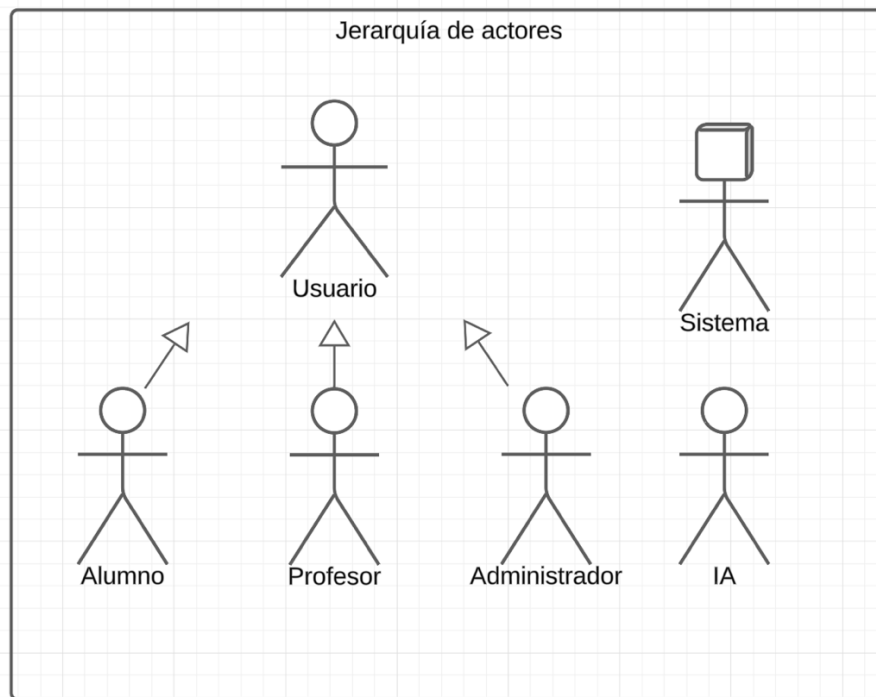


Figura 11. Jerarquía de actores en los diagramas de casos de uso

## Historias de usuario

En la fase inicial del diseño de la interfaz se han utilizado las historias de usuario, las cuales son una técnica utilizada en el desarrollo de software ágil para capturar requisitos de manera concisa y comprensible. A continuación, se describirán brevemente las historias de usuario que se ha empleado en el trabajo:

- **Registro en la aplicación:** descripción de la secuencia de acciones que tendría que hacer el usuario para poder registrarse en la aplicación.
- **Inicio de sesión:** descripción de los pasos a seguir por el usuario para poder iniciar sesión en la aplicación, habiéndose registrado previamente.
- **Cerrar sesión:** secuencia de acciones por hacer por parte del usuario para cerrar la sesión que tiene activa.
- **Iniciar una conversación:** acciones a realizar por el usuario para poder iniciar una conversación con la IA partiendo desde la página de “Inicio”.
- **Ver las conversaciones:** descripción de la secuencia de acciones que tiene que hacer el usuario para poder ver la lista de conversaciones que tiene asignadas a su usuario.
- **Dar de alta a un profesor:** secuencia de pasos a realizar por el administrador para cambiar el rol de usuario que es un profesor
- **Dar realimentación:** pasos a seguir por el profesor a la hora de dar realimentación a un alumno.
- **Eliminar una cuenta:** descripción de las acciones que tiene que hacer el administrador para eliminar una cuenta del sistema.

En el Anexo 1 se pueden encontrar las tablas (ver Tablas 30 - 37) pertenecientes a las historias de usuario realizadas para identificar los requisitos de la aplicación.

## Hipótesis y restricciones / Alcance

### Requisitos no funcionales

Además de las funcionalidades que definen las principales capacidades que tiene sistema, también hay una serie de requisitos no funcionales los cuales abarcan aspectos que no están directamente ligados a las acciones específicas que realiza el software, pero son esenciales para proporcionar una buena experiencia de usuario y asegurar la calidad del sistema.

Los requisitos no funcionales que se han identificado para este trabajo han sido los siguientes:

- **Usabilidad:** el sistema debe contar con una interfaz de usuario clara y fácil de usar, permitiendo a los usuarios (estudiantes y profesores) interactuar sin necesidad de entrenamiento extensivo.
- **Rendimiento:** el sistema debe generar respuestas (enunciados, diagramas y respuestas generales) en un tiempo razonable, idealmente no más de unos pocos segundos para mantener la fluidez de la interacción. Dependiendo de la complejidad de la petición este tiempo puede variar.
- **Escalabilidad:** el sistema debe estar diseñado para facilitar la incorporación de nuevas funcionalidades o el aumento de capacidad de usuarios sin grandes modificaciones estructurales ni problemas de rendimiento.
- **Seguridad:** implementar medidas de seguridad adecuadas para proteger los datos personales de los usuarios y la información generada durante las sesiones. También debe incluir sistemas robustos de autenticación y autorización para asegurar que sólo usuarios autorizados puedan acceder a ciertas funcionalidades.
- **Compatibilidad:** el sistema debe ser compatible e integrarse eficientemente con otras herramientas y plataformas educativas utilizadas por la institución como, por ejemplo: la incorporación de la aplicación a la plataforma de enseñanza que usen los docentes sin que haya problemas.
- **Portabilidad:** El software debe ser capaz de operar en diferentes sistemas operativos y dispositivos sin necesidad de modificaciones significativas. Al estar desarrollado en un entorno web esto no es de gran dificultad.

En el Anexo 1 se encuentran las tablas (Tablas 21 - 26) que describen las características de cada uno de los requisitos no funcionales de la aplicación.

### Restricciones del proyecto

#### Restricciones tecnológicas

Debido a que la aplicación se sirve de modelos de IA a la hora de generar tanto las respuestas generales, como diagramas de clase y enunciados, hay una gran dependencia en este aspecto. La empresa OpenAI es la encargada de mantener operativos los modelos, en el caso de que en un momento determinado decidan no continuar con alguno de los modelos de IA que se usan para el trabajo o cambiar la manera en la que se usa su API, los desarrolladores se verían en la tesitura de tener que modificar las posibles funcionalidades que se hayan visto afectadas por los cambios hechos por la compañía.

Por otra parte, y haciendo referencia a los cambios y actualizaciones que hacen las empresas a sus productos, en el caso de que cualquier parte del software (*framework*, lenguajes, etc.) cambie su modo de uso o actualicen la estructura de proyecto en el caso del *framework*, también podría llegar a afectar de manera negativa al proyecto que se ha desarrollado.

### Restricciones de tiempo y presupuesto

Al tratarse de un proyecto académico de fin de grado el cual se realiza a la par que se estudian otras asignaturas, el tiempo ha sido un factor determinante a la hora de desarrollar la aplicación. Como se ha explicado en apartados anteriores se ha utilizado la metodología SCRUM, este enfoque incremental no solo ha permitido una gestión efectiva del tiempo, sino que también facilitó la adaptación continua del proyecto a los cambios y la realimentación proporcionada por parte de los tutores, evitando así el desarrollo de funcionalidades que no agregan valor significativo al resultado final y posibles errores de comprensión de los requisitos.

Otra de las restricciones ha sido el presupuesto. La forma en la que se OpenAI saca beneficio de los usuarios que usan sus modelos por medio de su API es a través del cobro de una cierta cantidad de dinero a la hora de hacer una petición y obtener una respuesta a la IA. Los modelos más avanzados y los cuales pueden proporcionar respuestas de mayor calidad y extensión (ChatGPT-4 Turbo o ChatGPT-4o) son también en los que el millón de tokens pueden llegar a valer hasta diez o veinte veces más que en el modelo más básico (ChatGPT-3.5 Turbo). Para controlar la cantidad de *tokens* que se iban a consumir a la hora de hacer las pruebas pertinentes se ha utilizado la herramienta “Tokenizer” que pertenece a la misma empresa (ver Figura 12). Dicha herramienta permite introducir el texto de la solicitud que se le va a hacer al modelo y poder visualizar los tokens que se van a usar y la forma en la que el algoritmo detecta las secuencias de caracteres que componen un *token* separado por colores (OpenAI, 2024).

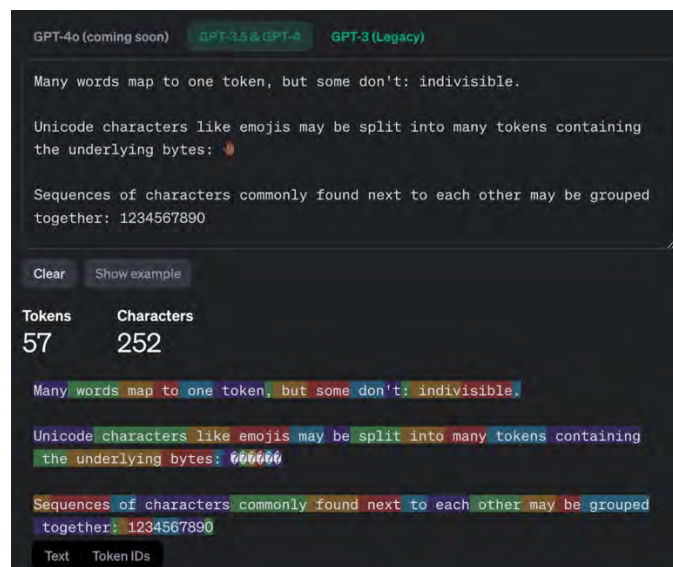


Figura 12. Ejemplo de uso de la herramienta “Tokenizer” Fuente: <https://platform.openai.com/tokenizer>

## Impacto esperado

El desarrollo de este trabajo se ha visto guiado por una serie de objetivos claros destinados a mejorar la enseñanza y el aprendizaje en el ámbito de la Ingeniería de Software. En los siguientes apartados se explican los tres sectores en los que se espera que la aplicación vaya a tener un impacto significativo.

### Impacto en los usuarios

El sistema diseñado en este proyecto de trabajo de fin de grado tiene como uno de sus principales objetivos e impactos la mejora significativa en el aprendizaje de UML por parte de los alumnos. A través de la implementación de una plataforma interactiva y personalizada, se espera que los estudiantes no solo adquieran conocimientos de manera más eficiente, sino que también logren una mejor retención a largo plazo de conceptos que son clave a la hora de desarrollar sistemas software de calidad. Esta mejora se logra mediante la integración de la inteligencia artificial en la aplicación, que adapta los ejercicios y el contenido a las necesidades específicas de cada usuario. Así, los estudiantes pueden avanzar a su propio ritmo, enfocándose en áreas donde necesiten más práctica y reforzamiento, lo que reduce la frustración y mejora la experiencia de aprendizaje global. Al facilitar un entorno más atractivo y receptivo, el sistema también busca aumentar la motivación de los estudiantes.

### Impacto en el sector educativo

El uso de la IA en el sector educativo es un tema que ha generado gran polémica, ya que por un lado hay personas que están totalmente en contra de ella, pero cada vez y más que están a favor. Según encuestas realizadas en distintos países, los docentes creen que el uso de la IA por parte de los estudiantes puede tener resultados positivos, aunque también hay quienes dicen que debido a ella los estudiantes tienden a desarrollar una dependencia que les perjudica en vez de favorecerles (García-Peñalvo et al., 2024). Con la aplicación que se ha desarrollado se busca reducir todavía más el porcentaje (ver Figura 13) que está en contra y hacer ver que puede ser de gran utilidad, tanto para los alumnos como para los profesores.

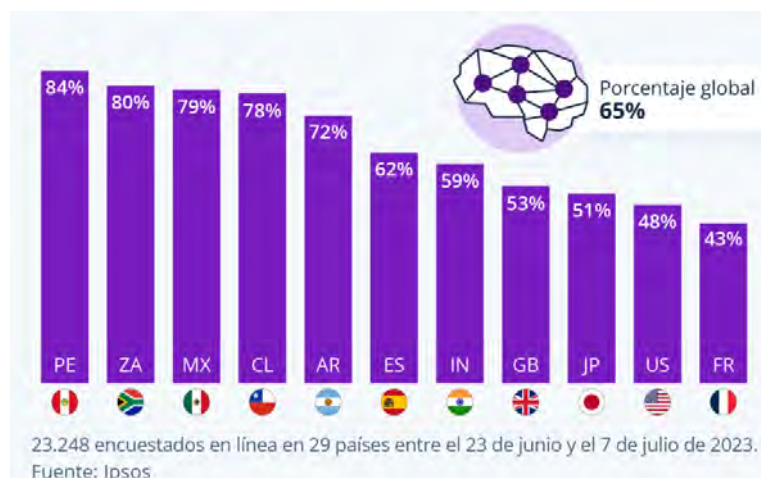


Figura 13. Gráfica del porcentaje de docentes a favor de la IA. Fuente: (Florencia Melo, 2023)

# Estudio de alternativas y viabilidad

## Análisis de herramientas y tecnologías

Para poder implementar la aplicación se ha tenido que elegir entre la gran multitud de opciones que existe hoy en día.

Como lenguaje de programación principal se ha elegido Python debido a que es un lenguaje fácil y rápido de aprender. A parte de esto, al ser uno de los lenguajes de programación más populares actualmente, a la hora de solucionar los problemas que han ido surgiendo se ha podido recurrir multitud de recursos web con documentación y foros de la comunidad en los que resolver y plantear las dudas surgidas.

Para la parte del *backend* de la aplicación se ha optado por PyCharm, el cual es un IDE desarrollado específicamente para Python y que permite la creación de proyectos de Django creando de manera automática la estructura de ficheros que se necesita para que la aplicación funcione correctamente. Ha sido de utilidad a la hora de programar la aplicación ya que cuenta con un sistema de predicción que permite avanzar en el código mucho más rápido y de forma intuitiva.

Como se ha explicado anteriormente se ha utilizado Django sobre otros *frameworks* de desarrollo de aplicaciones web como Flask o Pyramid, ya que aparte de ser gratuito y de código abierto, fomenta un desarrollo rápido y un diseño limpio y pragmático. Entre sus principales características se destaca su seguridad, rapidez y escalabilidad, describiéndose como un framework “*con baterías incluidas*”, ya que proporciona una amplia gama de funcionalidades ya integradas. Esto incluye autenticación, paneles de administración, manejo de formularios, operaciones de base de datos, entre muchas otras.

Por el contrario, Flask es un *micro-framework* que adopta un enfoque minimalista y proporciona las herramientas básicas para poner en marcha una aplicación web, dejando muchas decisiones y extensiones (como formularios o autenticación) al desarrollador. Esto puede ser una ventaja para proyectos que necesitan gran flexibilidad y personalización, pero puede requerir más tiempo para alcanzar la funcionalidad completa.

Pyramid se sitúa entre los dos anteriores. Es más flexible que Django y más estructurado que Flask, permitiendo a los desarrolladores elegir entre minimalismo o extensión según lo requiera el proyecto. No tiene tantas soluciones preempaquetadas como Django, lo que conlleva una necesidad configuración mayor y menor rapidez en el desarrollo inicial.

Para la parte *frontend* se ha utilizado Visual Studio Code, un editor de código con el que ya se había trabajado antes y que incorpora un *marketplace* de extensiones desarrolladas por la comunidad y por empresas las cuales permiten añadir una variedad de características con las que no cuentan otros IDE. Una de ellas es Live Server, una extensión que permite visualizar los cambios de diseño que se hagan en la página web de forma dinámica y en tiempo real, lo que ha

sido de gran ayuda para resolver los errores de la interfaz de forma rápida y eficaz.

Para estructurar el contenido de las distintas páginas con las que cuenta la aplicación se ha utilizado HTML y para darles un estilo a cada elemento se ha utilizado CSS. Ambos son los dos lenguajes de programación estándar que se usan para hacer páginas web. Para dar alguna que otra funcionalidad extra también se ha hecho uso de algunas bibliotecas de Javascript como, por ejemplo, jQuery o Ajax. jQuery ha sido de gran ayuda ya que permite manipular el *Document Object Model* (DOM), manejar eventos y añadir animaciones entre muchas otras cosas más (Mozilla, 2024).

Por otra parte, Ajax se ha utilizado para manejar eventos asíncronos directamente en la página web y sin tener que recurrir al *backend*. Se han elegido dichas bibliotecas debido a que se pueden incorporar mediante un elemento “script” a la plantilla de HTML y su sintaxis es fácil de aprender, ya que están creadas en Javascript, un lenguaje con el que se había trabajado antes en otros proyectos individuales.

Django permite a través de conjunto de etiquetas, añadir código ejecutable directamente en la plantilla de HTML de cada página de la aplicación, esto se ha usado por ejemplo para incrustar el menú lateral o para mostrar la lista de chats que tiene el usuario.

Como base de datos (BD) se ha utilizado PostgreSQL (<https://www.postgresql.org/>), la cual es una base de datos relacional, conocida por su robustez, escalabilidad y cumplimiento de estándares. Aunque existen muchas bases de datos populares, como MySQL, Oracle, SQL Server, MongoDB o SQLite3 (base de datos que incluyen los proyectos de Django por defecto), se ha optado por esta debido a que soporta una amplia gama de tipos de datos y permite la creación de otros nuevos definidos por el usuario. También soporta *arrays*, JSON y XML entre otros. Para poder hacer uso de esta BD se ha tenido que instalar un middleware que actúe de intermediario entre PostgreSQL y el proyecto de Django llamado *psycopy2-binary*”.

Alguno de los paquetes que se ha tenido que instalar de forma manual a la hora de desarrollar la aplicación es Pillow, el cual ha sido necesario para crear los atributos pertenecientes a las imágenes de los mensajes que se almacenan en la base de datos.

El otro paquete imprescindible para el proyecto ha sido PlantUML (<https://pypi.org/project/plantuml/>), un paquete de Python encargado de crear las imágenes de los diagramas de clase después de que ChatGPT genere el código. En el momento en el que se empezó a incorporar esa funcionalidad al proyecto, no había ninguna otra librería de Python que generase imágenes proporcionándole un fichero. La sintaxis de los ficheros PlantUML para generar los diagramas de clase es la siguiente (<https://plantuml.com/es/class-diagram>):

- **Inicio y fin del diagrama:** se delimita el código del diagrama de clases con las etiquetas de inicio y fin, “@startuml” y “@enduml” respectivamente.

- **Definición de clases:** para poder definir una clase se tendrá que utilizar la etiqueta “class” seguido del nombre de la clase y los atributos y métodos entre los caracteres “{}”.
- **Visibilidad de atributos y métodos:** para poder indicar la visibilidad que tendrán en el diagrama los distintos atributos y métodos de cada clase se pueden añadir los siguientes caracteres:
  - Privado: -
  - Público: +
  - Protegido: #
  - Paquete: ~
- **Relaciones entre clases:** después de haber definido las clases que tendrá el diagrama se podrán asignar relaciones entre ellas de la siguiente manera:
  - **Asociación:** se representa con una línea simple.
    - Ejemplo: Clase1 --> Clase2.
  - **Herencia:** se representa con una línea que termina en un triángulo apuntando a la superclase.
    - Ejemplo: ClaseHija --|> SuperClase
  - **Agregación:** Se representa con una línea que termina en un diamante vacío.
    - Ejemplo: ClaseContenedor <>-- ClaseContenida
  - **Composición:** se representa con una línea que termina en un diamante lleno.
    - Ejemplo: ClaseContendor ++-- ClaseContenida
- **Comentarios:** para hacer un comentario se deberá iniciar la línea con el carácter apóstrofo ('):
  - Ejemplo: ' Comentario en PlantUML

En la Figura 14 se puede observar un ejemplo del código PlantUML y su diagrama de clases correspondiente.

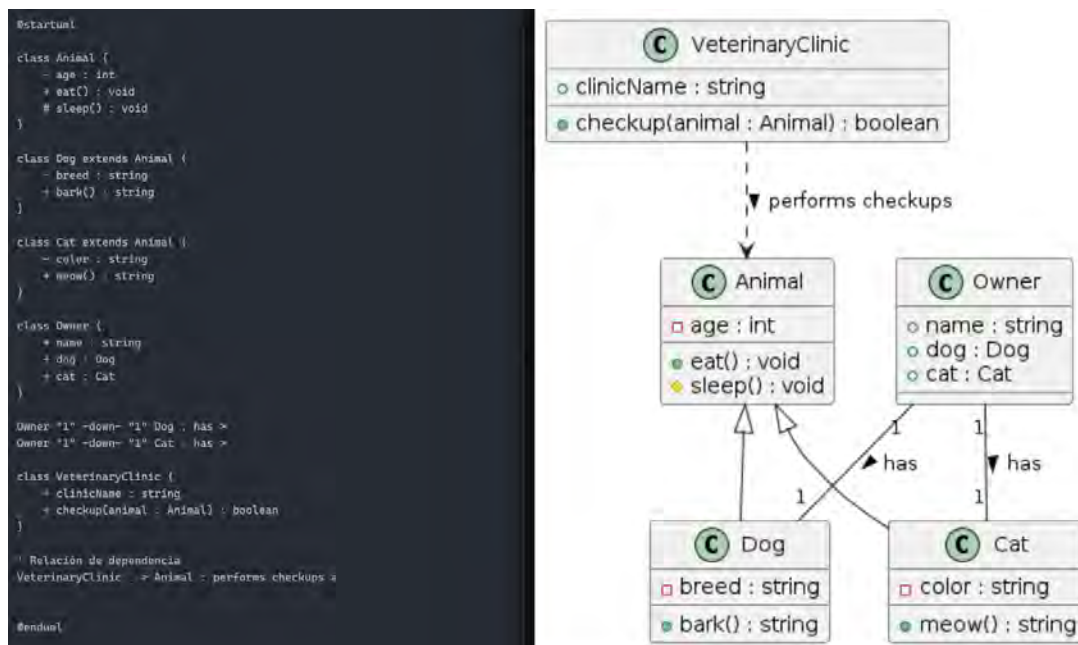


Figura 14. Ejemplo código PlantUML y su diagrama de clases



## Viabilidad y formas de monetización

El sistema que se ha desarrollado se puede implementar y mantener sin ningún tipo de problema, los gastos que conllevaría en la parte de mantenimiento serían, el posible sueldo del desarrollador encargado de solucionar cualquier tipo de problema que surja en el uso diario de la aplicación. También habría que tener en cuenta el coste de tener alojado dicho sistema en un servidor remoto y que funcione constantemente para que los estudiantes pudiesen acceder sin ningún problema a cualquier hora. En lo que respecta a los costes de haber desarrollado esta aplicación se puede tener en cuenta de que, en el caso de una empresa, esta tiene que pagar un sueldo mensual al desarrollador, así como mantener la plataforma en la que se vaya a incluir o vender. En este caso se ha podido terminar el proyecto en un intervalo de unos cuatro meses desde que se empezó con las fases iniciales.

A continuación, se detallan algunas de las posibles formas de sacar un beneficio con dicha aplicación teniendo en cuenta los costes previos:

- **Tarifa única:** esta es la forma más común y sencilla de sacar beneficios de cualquier aplicación que se desarrolle. El problema es que muchas veces a las personas les produce una sensación de rechazo el hecho de pagar por el software, ya que este no es tangible, aunque sea un producto que solucione sus problemas y tienden a gastar mucho más tiempo en buscar una posible solución gratuita.
- **Suscripción mensual/anual:** este es un modo de obtener beneficios que se ha puesto muy de moda en el sector de los *Software as a Service* (SAAS), ya que permite a los desarrolladores ganar una cierta cantidad de dinero de manera constante y a cambio, el cliente obtiene el producto y una serie de “extras” como, por ejemplo, una asistencia personalizada en un intervalo de tiempo determinado o poder probar y dar realimentación de las funcionalidades que todavía no se ha implementado para todo el público. En el caso de la suscripción anual, el usuario se compromete durante un año entero a pagar de manera mensual una cantidad de dinero, suele tener alguna característica más que el otro plan de suscripción y tiene una rebaja en el precio en comparación de si se pagase de forma mensual.
- **Acceso limitado a las funciones de manera gratuita:** en este modelo de negocio se permite al usuario acceder a la mayoría de las funcionalidades que tiene la aplicación de manera gratuita, pero hay un grupo de características a las que sólo se puede acceder a través de un pago, ya sea por suscripción o una tarifa única.

Estos modelos de negocios son algunas de las formas en las que se podría obtener un beneficio de la aplicación que se ha desarrollado. El tiempo en el que se verían cubiertos los gastos totales dependería del número de clientes y la cantidad a la que se venda el producto.

## Descripción de la solución propuesta

La aplicación que se ha desarrollado cuenta con varias funcionalidades, algunas que son comunes en prácticamente todas las aplicaciones ya existentes y otras que son exclusivas de esta. La primera página (ver Figura 15) a la que acceden los usuarios es la de “inicio”, la cual está dividida en dos partes. En la parte de la izquierda se pueden encontrar los botones de “Iniciar sesión” y “Registrarse” sobre un fondo que difumina la imagen de fondo de la página y en la parte de la derecha deja ver el fondo con claridad.

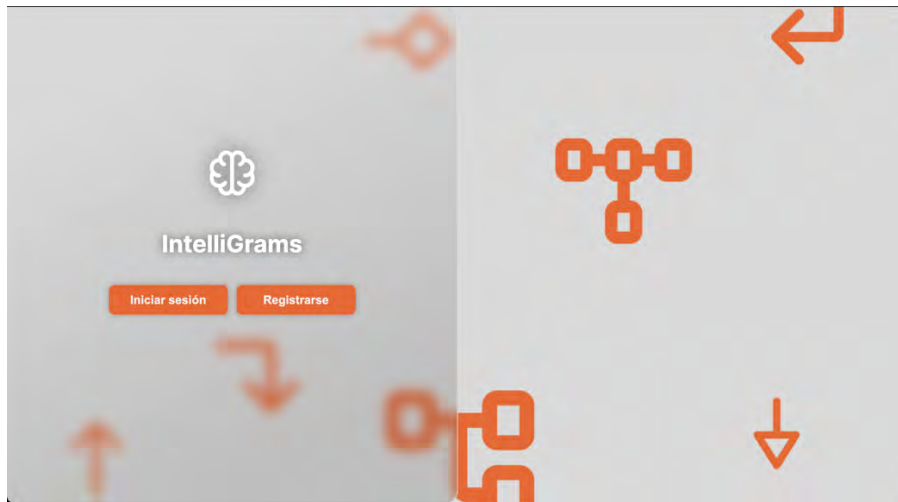


Figura 15. Página de inicio de la aplicación

Entre las funcionalidades comunes se puede encontrar las relativas a la gestión de los usuarios como, por ejemplo, el registro del usuario (ver Figura 16) en la aplicación el cual se puede hacer a través de un formulario a rellenar con los datos del usuario (email, nombre, apellidos y contraseña). Los elementos únicos que diferencian a los usuarios son sus direcciones de correo electrónico, por lo que no puede haber más de una cuenta con el mismo email. En el caso de que el usuario ya tenga una cuenta puede hacer clic en el enlace de la parte inferior del formulario.

Figura 16. Formulario de registro

Para que el sistema verifique que el correo que ha introducido el usuario es válido y tiene acceso al mismo, se enviará un email de manera automática al usuario (ver Figura 17) cuando se pulse el botón de “Registrarse” en el formulario de la Figura 16. En dicho email se podrá acceder a un enlace que redirigirá al usuario a la página de inicio de sesión. Es importante destacar que este mecanismo de seguridad cumple una función muy importante, ya que evita que usuarios creen cuentas con correos que no les pertenezca. Hasta que el usuario no haga clic en el enlace que se le manda, el estado interno de la cuenta en el sistema no será válido y no podrá acceder a la aplicación.

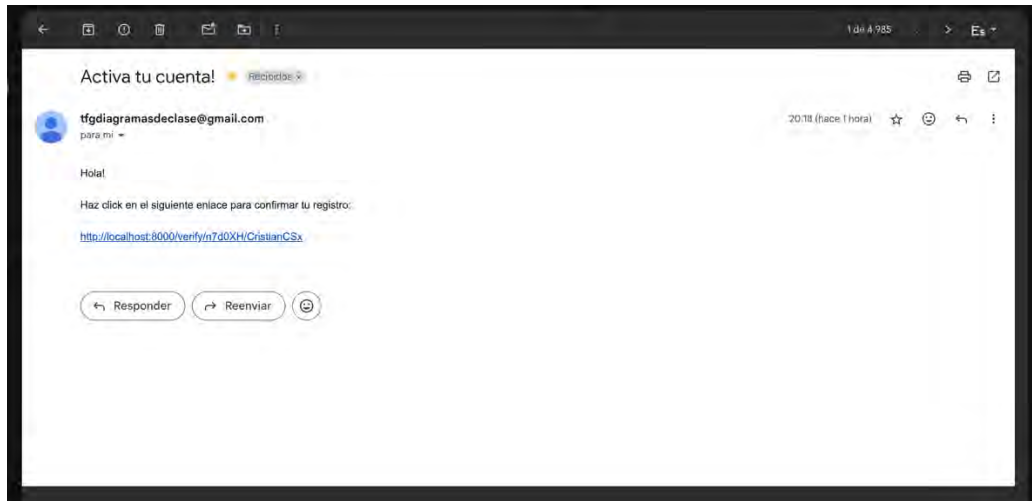


Figura 17. Contenido del correo electrónico de verificación

Una vez el usuario se ha registrado y ha verificado su dirección de correo electrónico, puede iniciar sesión en la aplicación. Para ello se tendrá que rellenar un formulario con su correo electrónico y su contraseña (ver Figura 18).

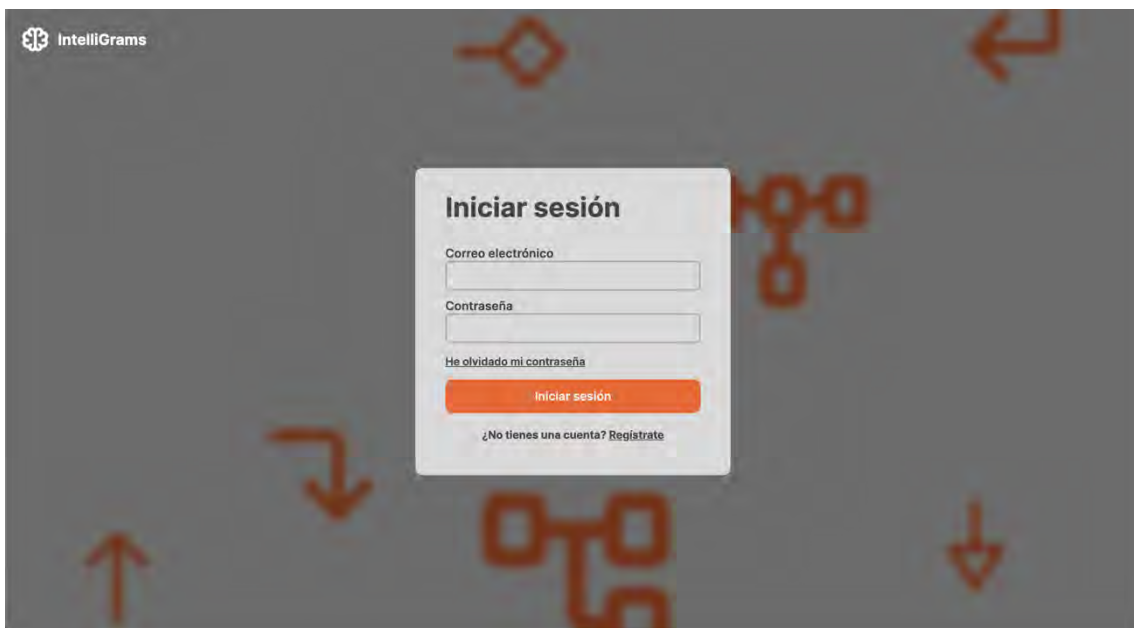


Figura 18. Formulario de inicio de sesión

Por otra parte, el proceso de eliminación de cuenta lo tendrá que hacer el administrador de manera manual desde el panel de administración que proporciona Django de forma automática al crear el proyecto. Debido al diseño de la base de datos, cuando se elimina un usuario del sistema también se eliminan todos los mensajes y todos los chats que haya creado para evitar posibles problemas (ver Figura 19).

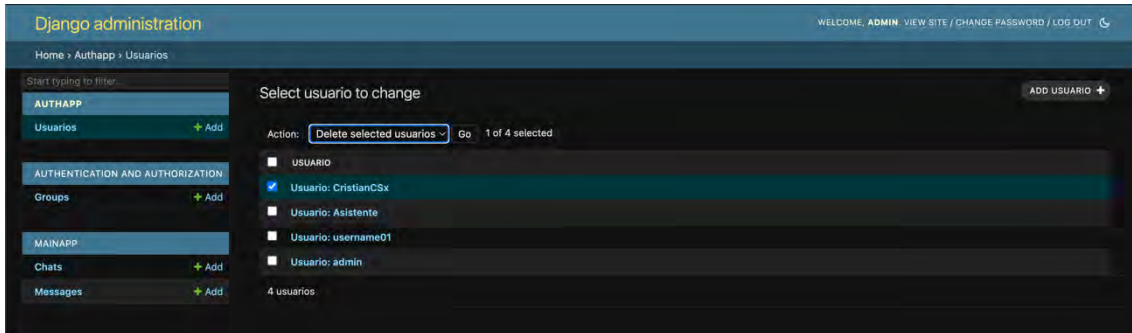


Figura 19. Eliminación de la cuenta de un usuario

En el momento en que un usuario se registra en la aplicación también se le asigna el rol de “Alumno” por defecto para que pueda utilizar la aplicación instantáneamente (ver Figura 20 y Figura 21). En el caso de que un usuario sea un docente, el administrador tendrá que asignarle el rol de “Profesor” de manera manual. Solo los usuarios que tienen el rol de “Profesor” pueden ver la sección de “Feedback” en el menú lateral.

Figura 20. Cambio del rol de un usuario

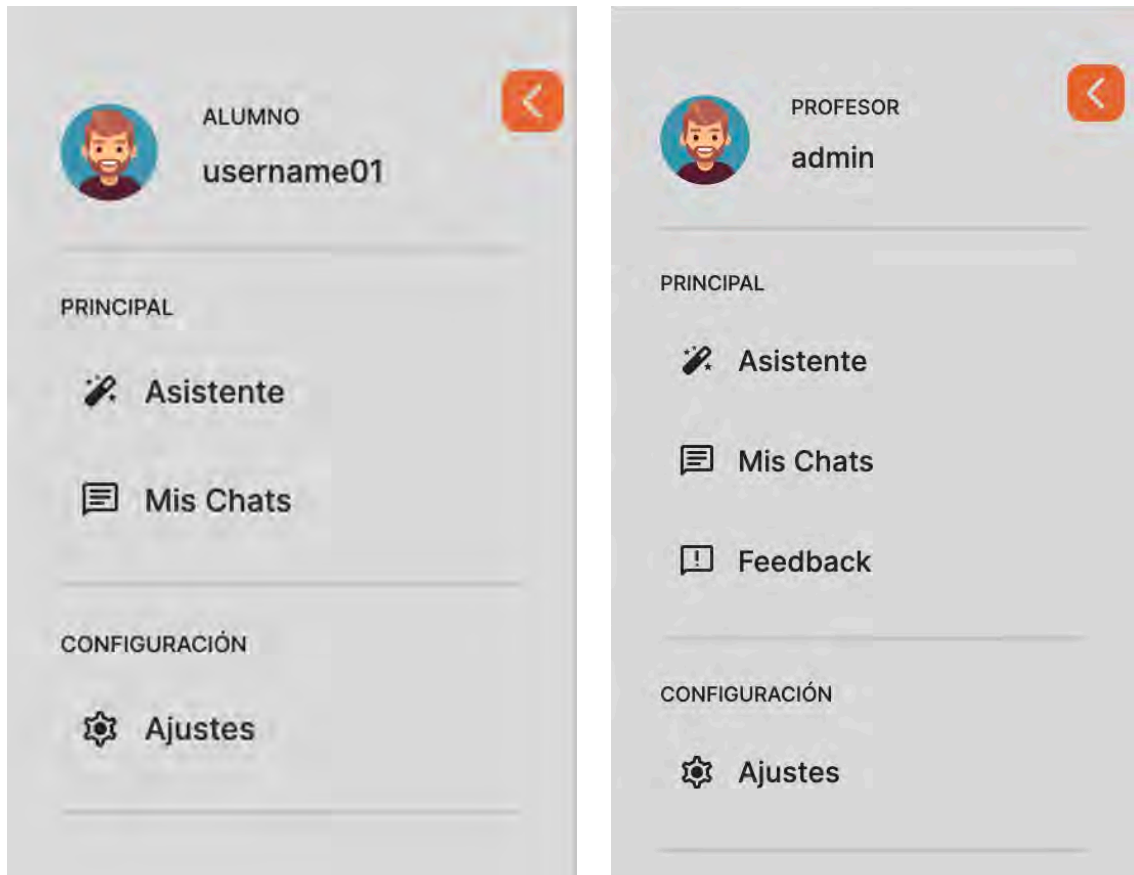


Figura 21. Comparación del menú entre usuarios con distintos roles

Cuando un usuario introduzca una pregunta o solicitud a través de la aplicación, el mensaje se mandará a la API de ChatGPT y se mostrará la respuesta que devuelva el modelo. Dependiendo de la solicitud que haga el usuario, la IA puede proporcionar una explicación, un enunciado que se resuelva con un diagrama de clases UML o la solución ha dicho enunciado en formato de imagen.

Para implementar la funcionalidad de que la IA cree los diagramas de clase ha sido necesario usar una biblioteca de Python llamada PlantUML, la cual genera imágenes en formato “.png” de un diagrama de clases a partir de un fichero de texto que se pase como argumento. La manera en la que se usa esta biblioteca es a través de comandos por medio de un terminal, por lo que para poder implementarla de manera automática se ha tenido que recurrir al módulo “subprocess” que ofrece Python, el cual permite ejecutar comandos de terminal desde el código. Para identificar las imágenes de los diagramas de clase de manera única, se les asigna un nombre en función de la fecha y hora en la que se cree el mensaje.

Como se explicado en apartados anteriores, la principal funcionalidad de la aplicación es que el usuario pueda interactuar con un modelo de IA y este le responda. Entre las respuestas que puede solicitar el usuario se encuentran las siguientes:

- **Enunciado:** el usuario solicitará un enunciado de un problema que se resuelva a través de un diagrama de clases UML (ver Figura 22).

- **Diagrama de clases:** el usuario solicitará que se genere el diagrama de clases que sirve de solución a un problema propuesto previamente por la parte de la IA (ver Figura 23).
- **Análisis de una imagen:** el usuario solicitará a la IA que analice y devuelva una explicación de la imagen que ha proporcionado el usuario (ver Figura 24). Para ello podrá subir una imagen a la aplicación a través de un botón disponible en el chat (ver Figura 25). Esta imagen se corresponderá con un diagrama de clases que haya hecho el alumno, por ejemplo (ver Figura 26).

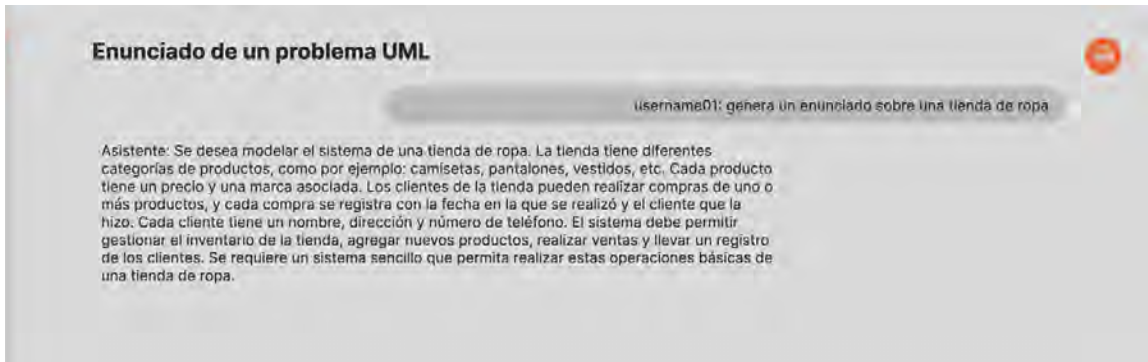


Figura 22. Solicitud de un enunciado al modelo



Figura 23. Solicitud de un diagrama de clases al modelo

**Enunciado de un problema UML**

```

classDiagram
    class Telefono {
        String telefono
    }
    class Compra {
        Date fecha
    }
    class Tienda {
        String nombre
    }
    class Producto {
        Float precio
    }
    class Categoria {
        String nombre
    }
    class Marca {
        String nombre
    }
    Telefono "1" -- "n" Compra
    Compra "1" -- "1" Tienda
    Compra "1" -- "1" Producto
    Tienda "1" -- "n" Categoria
    Producto "1" -- "n" Marca
    Producto "1" -- "1" Categoria : pertenece a
    
```

username01: Explicame qué es una clase de un diagrama de clases UML

Asistente: Una clase en un diagrama de clases UML representa un conjunto de objetos con propiedades, métodos y relaciones comunes. Es la plantilla para crear objetos que comparten características similares. En la clase se especifican los atributos (variables), operaciones (métodos) y relaciones con otras clases. Las clases se representan con un rectángulo dividido en tres secciones: la primera sección contiene el nombre de la clase, la segunda sección lista los atributos de la clase y la tercera sección muestra los métodos de la clase.

Escribe un mensaje

Figura 24. Solicitud de una explicación al modelo

**Creando un enunciado juntos**

Asistente: ##Enunciado. En una tienda de ropa se venden diferentes tipos de prendas, como camisetas, pantalones, y zapatos. Cada prenda tiene un precio y una marca. Los clientes pueden realizar compras de una o más prendas en una sola visita a la tienda. Además, la tienda ofrece

genera el diagrama de clases

CristianCS: esta bien así

Nombre	Tamaño	Clase	Fecha de inclusión
OpenAI	--	Carpeta	22 abr 2024, 17:24
diagrama.png	23 KB	Imagen PNG	15 abr 2024, 12:00
diagrama.txt	685 bytes	Texto	15 abr 2024, 11:59
script.py	1 KB	Script Python	11 abr 2024, 12:35
View_Pruebas	--	Carpeta	11 abr 2024, 11:34
PruebaTFG	--	Carpeta	9 abr 2024, 0:33
Chatgpt	--	Carpeta	4 abr 2024, 12:55
TallerChatGPT	--	Carpeta	4 abr 2024, 9:58
discordBot	--	Carpeta	15 mar 2024, 0:16
proyectoWebDjango	--	Carpeta	6 mar 2024, 16:19
TutorialDjango	--	Carpeta	3 mar 2024, 20:17
TallerPython	--	Carpeta	23 feb 2024, 11:11
Pruebas.py	5 KB	Script Python	10 feb 2024, 19:07
Ej1.py	1 KB	Script Python	10 feb 2024, 18:26

Escribe un mensaje

Figura 25. Elección de una imagen para enviar al chat



Figura 26. Solicitud del análisis de una imagen

Cuando un alumno desee enviar uno de sus chats a un profesor para que este le proporcione realimentación, lo puede hacer pulsando el botón que se encuentra en la esquina superior derecha del chat. En el momento en que el usuario presione dicho botón, el sistema seleccionará de manera aleatoria a cualquiera de los usuarios que tenga el rol de “Profesor” y el chat aparecerá en el apartado “Feedback” que solo pueden ver los docentes en el menú lateral.

La vista del alumno se actualizará y se deshabilitará la parte en la que los usuarios pueden escribir y mandar sus mensajes para que, de este modo, el siguiente mensaje que aparezca en el chat sea el comentario de realimentación que haga el profesor (ver Figura 27).

Se ha descartado la opción de bloquear el acceso al chat cuando un alumno manda el chat a revisión y se ha optado por deshabilitar la posibilidad de mandar mensajes para que el alumno pueda seguir teniendo acceso al chat y ver los mensajes que ha mandado en el caso de que lo necesite. Cuando un chat esté pendiente de que un docente proporcione su realimentación, este aparecerá en la lista de los chats del alumno con un indicador a la derecha que lo diferenciará del resto (ver Figura 28). Cuando el profesor haya mandado el mensaje correspondiente, la vista se actualizará resaltando el chat con un borde naranja (ver Figura 29).

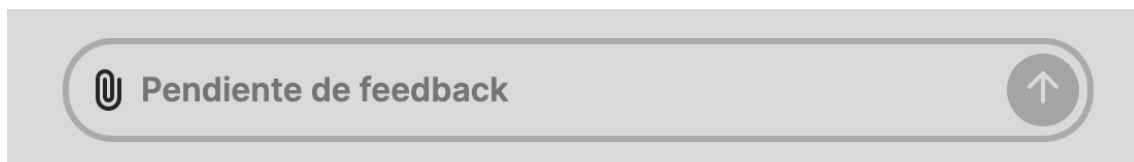


Figura 27. Actualización de la barra de mensajes al solicitar realimentación



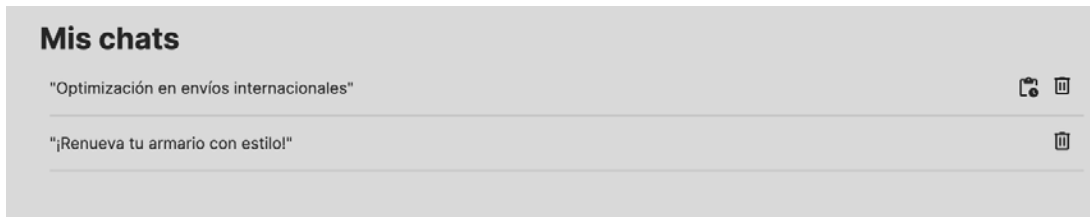


Figura 28. Actualización de la lista de chats al solicitar feedback



Figura 29. Actualización de la lista de chats al mandar feedback

Alguna de las funcionalidades extra con las que cuenta la aplicación sería el apartado de "Ajustes que se encuentra en el menú lateral, al pulsarlo se abrirá una ventana modal en la que el usuario podrá cambiar algunos datos de su cuenta como, por ejemplo, el nombre, el nombre de usuario y los apellidos (ver Figura 30). En esta ventana también se mostrará el rol que tiene en el sistema y el correo electrónico asociado a la cuenta, pero no los podrá modificar ya que ese tipo de parámetros son críticos a la hora de acceder a funcionalidades que se limitan solo a los profesores como el apartado de "Feedback" y a la hora de hacer iniciar sesión en la aplicación. Otra de las ventanas modales es la de confirmación del cierre de sesión (ver Figura 31).

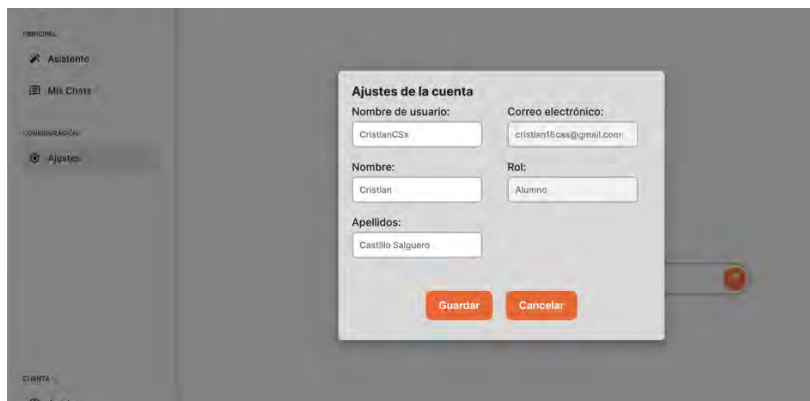


Figura 30. Ventana de ajustes

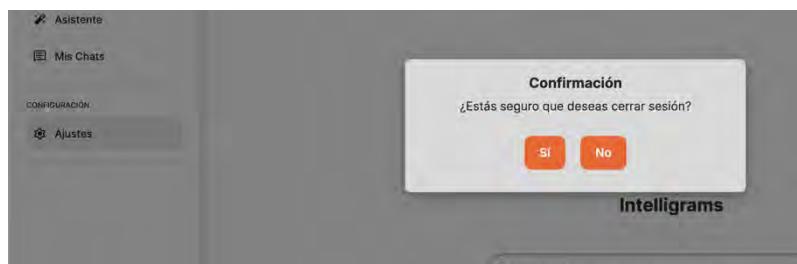


Figura 31. Confirmación del cierre de sesión

Para un mejor entendimiento sobre el flujo de conversación entre el sistema y el usuario se ha realizado los esquemas BPMN (<https://bpmn.io/>) de los dos escenarios de conversación posibles en la aplicación (ver Figuras 32 y 33).

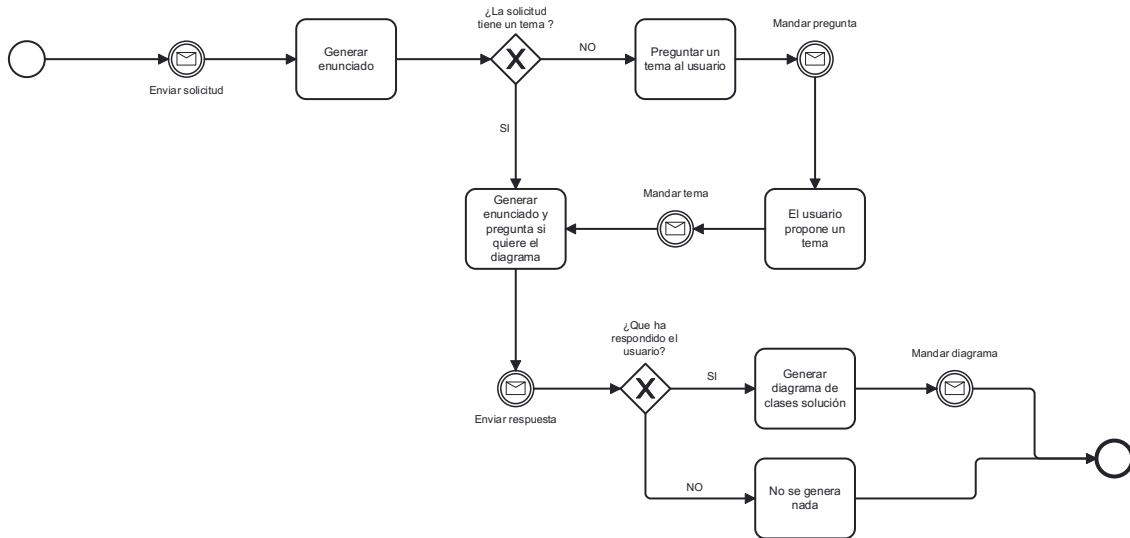


Figura 32. Esquema BPMN de una interacción normal entre el usuario y la IA

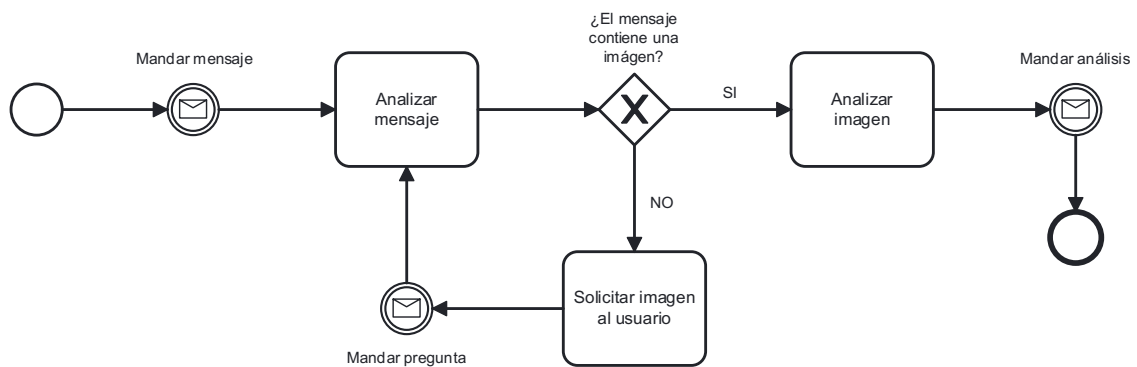


Figura 33. Esquema BPMN cuando el usuario sube una imagen a la aplicación

La aplicación se ha desplegado en un servidor del equipo de investigación GRIAL (<https://d-ia-gramas.grial.eu/>) y se ha hecho un vídeo demostrativo en el que se explican y se pueden ver todas las funcionalidades (<https://youtu.be/gpEW9jzUqRw>).

# Análisis de riesgos

## Análisis DAFO

Para el análisis de riesgos se ha utilizado el análisis DAFO (ver Figura 34), una herramienta estratégica que ayuda a evaluar un proyecto desde diferentes perspectivas, permitiendo planificar estrategias efectivas para enfrentar los desafíos y aprovechar las ventajas. Se divide principalmente en dos tipos de análisis, el análisis interno en el que se encuentran las debilidades y las fortalezas y el análisis externo en el que se evalúan las amenazas y oportunidades. Estos dos grupos cuentan con el análisis de aspectos positivos y negativos que afectan al proyecto.

- **Análisis interno:**

- **Debilidades:** algunas de las debilidades que se han identificado en el análisis son la curva de aprendizaje por parte del usuario. Al principio puede resultar algo complicado entender el flujo de conversación que hay tener con la IA para obtener los resultados deseados, pero después de una serie de pruebas se vuelve algo muy intuitivo y sencillo. Otra de las debilidades que se han identificado son las limitaciones tecnológicas del proyecto debido a la dependencia que se tiene de los algoritmos de IA.
- **Fortalezas:** el hecho de haber incorporado un modelo de IA al proyecto ha favorecido la automatización de tareas de gran importancia para el uso de la aplicación. El proyecto introduce una herramienta educativa innovadora que puede transformar la manera en que los estudiantes aprenden y comprenden la ingeniería de software y el diseño de sistemas. Otra de las fortalezas ha sido el desarrollo de una interfaz de usuario amigable y fácil de usar que puede facilitar la adopción por parte de los usuarios finales, como estudiantes y profesores.

- **Análisis externo**

- **Amenazas:** entre las principales amenazas identificadas se puede encontrar la posible competencia que pueda surgir debido a la popularidad creciente de los modelos de IA entre los usuarios. Otra de las amenazas que se ha encontrado es la posible resistencia por parte de los educadores o instituciones educativas a adoptar nuevas tecnologías por preferencia a métodos tradicionales. La rapidez con la que evoluciona la tecnología podría hacer que el proyecto se vuelva obsoleto si no se actualiza constantemente.
- **Oportunidades:** una de las oportunidades de la aplicación es el potencial para implementar la solución en diversas instituciones educativas, como las universidades, ya que conlleva beneficios significativos tanto para los alumnos como para los profesores. Y, por último, la oportunidad de formar alianzas con empresas de tecnología y educativas que buscan fomentar la innovación en la enseñanza de la ingeniería de software.

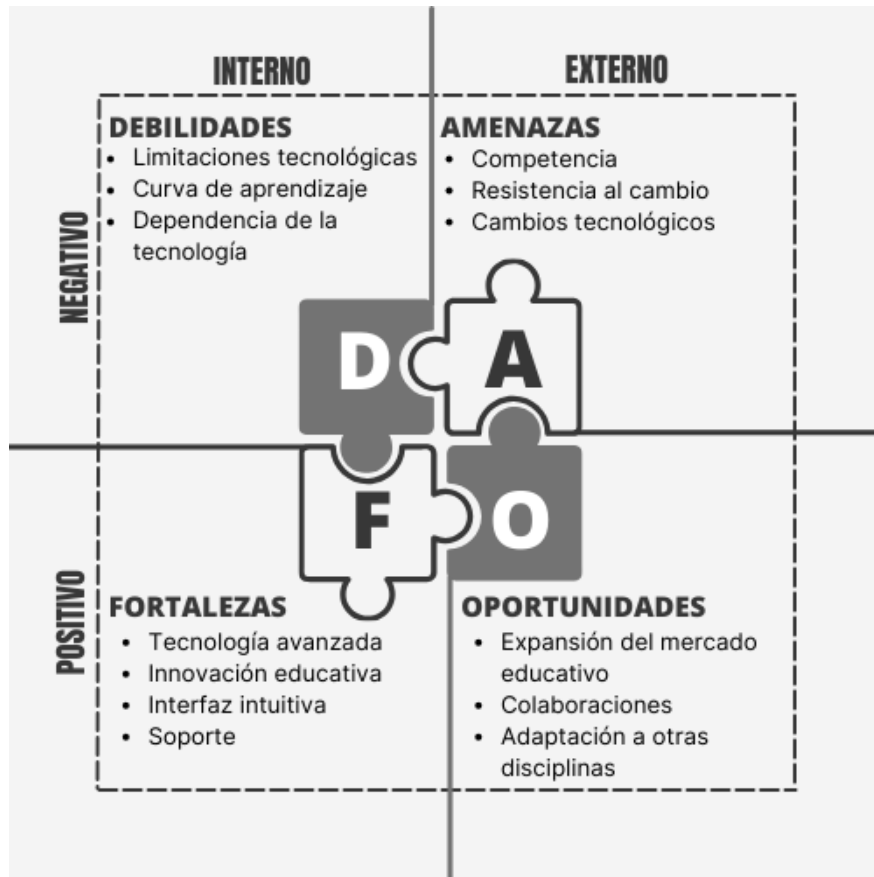


Figura 34. Matriz del análisis DAFO de la aplicación

## Organización y gestión del proyecto

### Organización

#### Arquitectura del sistema

Como ya se ha explicado en apartados anteriores, este proyecto se ha desarrollado con el *framework* web Django de Python. Para entender cómo funciona y cuál es la arquitectura de los proyectos creados con Django se puede comparar con el patrón Modelo-Vista-Controlador (MVC), que es un patrón de diseño software ampliamente utilizado para desarrollar interfaces de usuario dividiendo una aplicación en tres componentes interconectados (Buschmann et al., 1996). Permite separar los posibles errores que pueda tener cada parte, facilitando la gestión y el mantenimiento del código. En el modelo se puede encontrar lo relativo al tratamiento de datos y lógica que se haya implementado como algoritmos y las funcionalidades correspondientes a la aplicación. El componente de la vista se encarga de mostrar la información al usuario de manera que sea entendible y que puede estar formada por texto, entrada de datos y botones entre otros.

Por último, el controlador actúa de intermediario entre el modelo y la vista recibiendo las entradas o datos que proporcione el usuario a través de la vista, envía esos datos al modelo, el cual se encarga de procesarlos y finalmente devuelve los datos a la vista para ser presentados.

Django utiliza el patrón Modelo-Plantilla-Vista (del inglés, *Model-Template-View* – MVT), (ver Figura 35 y Figura 36) siendo una adaptación del patrón descrito anteriormente. En este caso el modelo sigue siendo y comportándose de la misma manera que en el MVC, pero se sustituye lo que antes era la vista por la plantilla (*template*) y el controlador por la vista (Infante Montero, 2012). En el modelo se definen los datos almacenados, los cuales se encuentran en forma de clases de Python que representarán objetos del sistema y tendrán una serie de atributos y métodos. La plantilla será un archivo HTML en el que se recibirán y presentarán los datos utilizando una serie de etiquetas que incorpora Django para acceder a datos que tengamos almacenados directamente desde el código de la plantilla de HTML y también permite incluir estructuras de datos como estructuras condicionales *if* o bucles *for* y *while*. Una de las principales ventajas que brinda este *framework* es el de poder incrustar plantillas dentro de otras de una manera muy simple, permitiendo al usuario reutilizar código y ahorrar tiempo en el proceso de desarrollo de componentes.

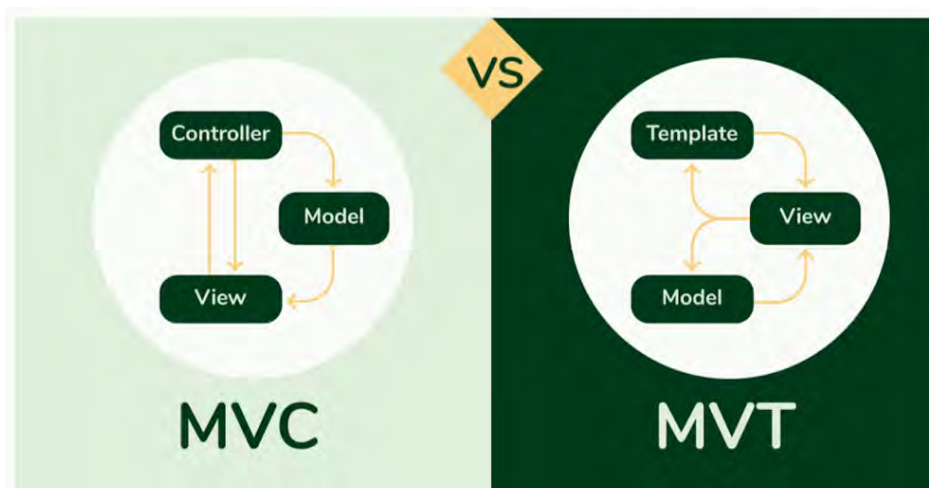


Figura 35. Comparación MVC y MVT. Fuente: <https://sahiljaggarwal.hashnode.dev/mvc-vs-mvt-architecture>

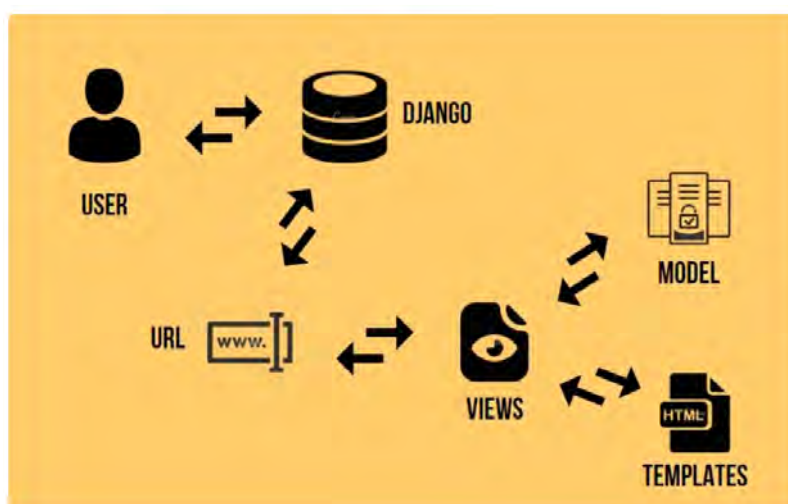


Figura 36. Arquitectura de Django. Fuente: <https://dev.to/sm0ke/django-tutorial-mvt-architecture-custom-commands-19nb>

Siguiendo la estructura de los proyectos de Django, se ha dividido en dos aplicaciones principales, la primera de ellas encargándose de la gestión de los usuarios (inicio de sesión, registro de usuarios, verificación de emails, etc.) y la segunda con el resto de las funcionalidades de la aplicación (chats, ajustes, etc.).

El esquema de implementación de una funcionalidad sería el siguiente:

1. **Definir la funcionalidad en el archivo views.py:** se creará una función de Python con un nombre característico en la que se implementará lo que desee el usuario que suceda en un determinado momento, ya sea que el usuario presione un botón o acceda a una página en específico. Será también en este fichero donde se determine la plantilla que verá el usuario al acceder a cada página.
2. **Incluir la URL en el archivo urls.py:** la dirección a la que tendrá que acceder el usuario y que podrá ver en la parte superior del navegador, la cual se asociará con la función previamente creada y se le asignará un nombre.
3. **Asociar un elemento de la vista con la URL:** en la plantilla de HTML se vinculará el nombre de la URL que se ha añadido anteriormente con un elemento de la plantilla HTML como, por ejemplo, un botón. De este modo cuando se pulse dicho botón, se llamará a la función asociada con la URL y se procederá a tratar los datos que correspondan.

Las clases que se definen en el archivo models.py y sus respectivos atributos se corresponden con las tablas que albergará la base de datos del proyecto. Todos los cambios sobre los datos que se realicen se guardarán si así se desea, para ello se podrán definir una serie de métodos en las clases con los que se obtendrán y modificarán los valores almacenados. A la hora de crear un proyecto de Django se incluye y se crea una base de datos de SQLite3 por defecto, la cual permite estructurar los datos que se vayan a almacenar en el sistema siempre y cuando sean de tipos básicos como, por ejemplo: cadenas de caracteres, fechas, números, etc.

En el caso de que el desarrollador quiera incluir atributos de cierta complejidad como un array o tributos en formato JSON o XML, se tendrá que optar por usar otra base de datos como PostgreSQL, la cual permite definir estructuras de datos algo más complejas en los atributos de los objetos del sistema. En los archivos de configuración del proyecto se debe de indicar el nombre del middleware que actuará de intermediario y comunicará el proyecto de Django con la base de datos.

Para poder trasladar los cambios que se hagan en el proyecto a la DB, hay que ejecutar una serie de comandos:

- **python manage.py makemigrations:** Este comando es responsable de crear nuevas migraciones basadas en los cambios realizados en los modelos de Django. Una migración en Django es un archivo que describe cómo adaptar la base de datos para que coincida con los cambios realizados en el código.
- **python manage.py migrate:** es el comando que realmente aplica las migraciones generadas por “makemigrations” a la base de datos. Este es

el proceso que modifica la estructura de la base de datos para que coincida con los modelos definidos en el código.

## Componentes

### Base de datos

Para poder ver el esquema de la base de datos (ver Figura 37) se ha hecho uso de un plugin del IDE PyCharm, el cual permite generar un esquema con todas las relaciones y tablas que haya en la base de datos que se utilice en el proyecto. En futuros apartados (ver Anexo 3) se hace una explicación en profundidad.



Figura 37. Esquema de la base de datos del proyecto generada por el IDE

### Protocolos

Uno de los protocolos básicos que se utilizan en un proyecto de Django y en las aplicaciones web es el protocolo HTTP/HTTPS. HTTP funciona como un protocolo de solicitud-respuesta entre un cliente y un servidor. El cliente, el que generalmente es un navegador web, envía una solicitud al servidor; el servidor, a su vez, procesa los datos y responde con los recursos solicitados y un mensaje de estado. HTTPS (HTTP Secure) es la versión segura de HTTP, el cual utiliza un cifrado SSL/TLS para proteger la integridad y la privacidad de los datos intercambiados. HTTPS encripta la sesión con un certificado digital, asegurando que todos los datos transferidos sean seguros y a prueba de interceptaciones. Aunque este último protocolo cuenta con mecanismos de seguridad también cuenta con algunas limitaciones, ya que cifra los datos durante la transmisión, por lo que, si un atacante accediera a las páginas almacenadas en caché del navegador de la víctima, aún podría robar datos (Hostilica, 2024). Una comparativa entre estos dos protocolos se tiene en la Figura 38.

# HTTP vs HTTPS



Figura 38. Comparación de HTTP y HTTPS. Fuente: (Hostilica, 2024)

## Algoritmos

El algoritmo empleado por los modelos desarrollados por la empresa OpenAI a la hora de generar las respuestas se basa en la arquitectura Transformer (Vaswani et al., 2023). Estos modelos se entrenan con un conjunto de datos de texto muy grande, el cual le permite aprender patrones de lenguaje. El proceso de generación de respuestas se basa en la predicción de palabras sucesivas, donde el modelo selecciona la siguiente palabra más probable basándose en el contexto proporcionado por las palabras anteriores. La capacidad del modelo para comprender y generar texto de manera coherente se refina a través de técnicas de aprendizaje profundo (*deep learning*) y ajuste fino (*fine-tuning*) con datos específicos de la tarea.

Estos patrones de aprendizaje son bastante complejos de entender para una persona que no tenga conocimientos previos en el sector de la IA y de los modelos de lenguaje de gran tamaño (del inglés, *Large Language Model* – LLM), por lo que a la hora de usar ChatGPT la manera más fácil es simulando un chat en el que el usuario hace una serie de preguntas y el modelo la analiza para, posteriormente, proporcionar una respuesta válida.

En el esquema de la Figura 39 se puede ver de manera simplificada y separada por partes, la forma en la que trabajan modelos como ChatGPT (Cuofano, 2024).



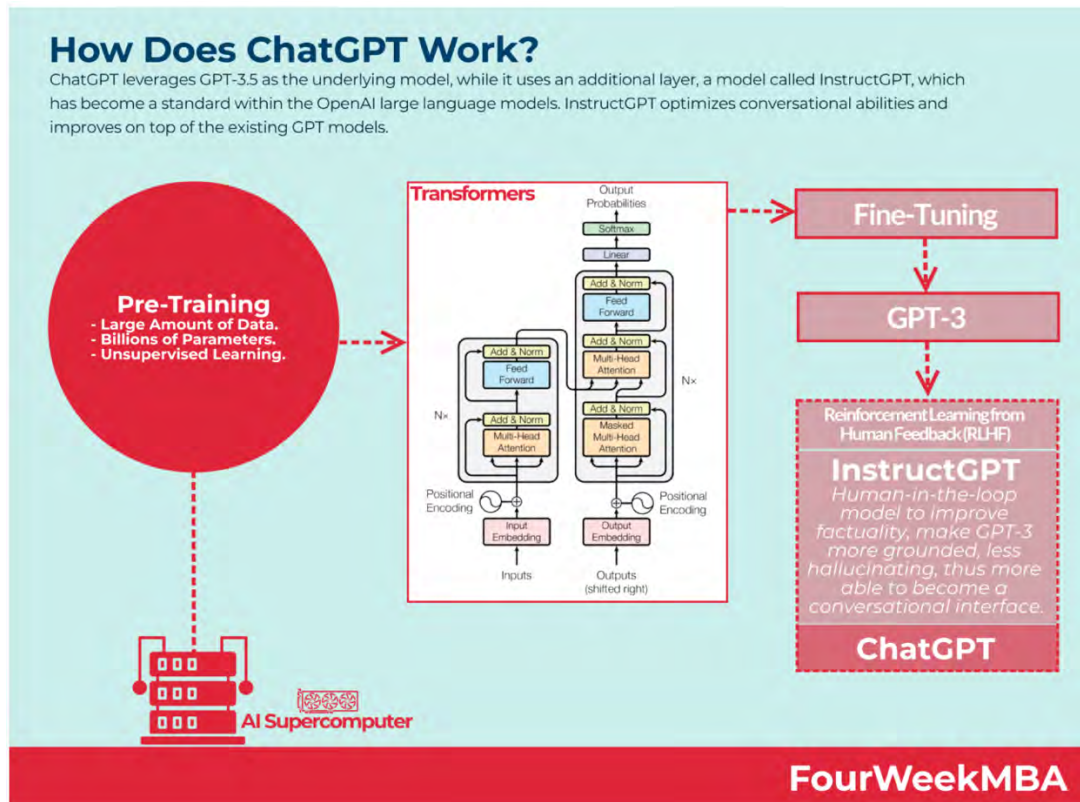


Figura 39. Esquema del funcionamiento de ChatGPT. Fuente: (Cuofano, 2024)

## Desarrollo de la solución propuesta

Como se ha detallado en puntos anteriores, la metodología empleada para desarrollar este trabajo ha sido la metodología SCRUM, la que se caracteriza por dividir en *sprints* (en proyectos grandes y con varias personas en el equipo de desarrollo son entre 2 y 4 semanas) el desarrollo incremental de un proyecto. Cada sprint con templa un periodo corto de tiempo en el que se pedirá al desarrollador que se implementen funcionalidades que se verificarán en el próximo. Dado que el equipo estaba formado por una sola persona (el estudiante), estos periodos han variado entre 1,2 y 3 semanas para implementar cada funcionalidad dependiendo del nivel de complejidad y el incremento solicitado.

Después de haber estudiado, analizado y documentado lo que iba a ser el trabajo, se modeló la estructura base de proyecto de Django y se creó una aplicación inicial que es la que se encarga de la gestión de los usuarios (AuthApp). Dentro de esta aplicación se desarrollaron las plantillas de inicio, registro e inicio de sesión y se implementó la funcionalidad para cada página además de la verificación del correo electrónico del usuario.

Después de haber adquirido suficientes conocimientos sobre los modelos de lenguaje, se creó la aplicación en la que se incluye la funcionalidad relativa a los chats (MainApp). Se diseñó el menú lateral que se puede observar en todas las páginas a las que accede el usuario una vez en la aplicación y se crearon las plantillas incrustando el menú en cada una para reutilizarlo y no tener que

duplicar código. Estas plantillas solo tenían la estructura base de los archivos HTML y el menú, no contenían datos adicionales.

Posteriormente se especificaron las características de las funcionalidades relativas al *feedback*. En un primer momento y como se detalla en la propuesta, los docentes iban a tener la posibilidad de interactuar directamente con el modelo para aportar observaciones y correcciones sobre la forma en la que la IA genera el código PlantUML para crear los diagramas de clase, pero después de debatirlo se optó por implementar un sistema que permita a los usuarios enviar las conversaciones que ellos deseen, esperando comentarios/mensajes de *feedback* por parte de un profesor que se puedan ver en el propio chat pero con otro estilo para diferenciarlos del resto de mensajes. También se contempló el formato en el que la IA iba a generar el código para crear los diagramas y cómo se iban a generar los diagramas, siendo XML una de las primeras opciones propuestas por los tutores, el estudiante tendría que copiar y pegar manualmente el código en una aplicación web externa para poder visualizar los diagramas de clases propuestos por la IA. Como resultado de la búsqueda información sobre los mecanismos disponibles para este cometido que fuesen compatibles con Python se llegó a la librería PlantUML, la cual permite implementar un mecanismo de generación de imágenes y poder generar los diagramas de clase directamente en el código.

Posteriormente, se desarrollaron por completo las funcionalidades relacionadas con la gestión de los usuarios y unos prototipos básicos de chats en los que se prefijaba una respuesta por defecto en el código. Esto se hizo principalmente para ver los posibles flujos de interacción entre el alumno, la IA y el docente.

Una vez se tuvieron claros dichos flujos, se procedió a implementar la funcionalidad usando la API de ChatGPT para generar respuestas desde cero proporcionándole el comportamiento que debería tener, algunos ejemplos de respuestas a determinadas peticiones y alguna serie de restricciones. Debido a que cada petición a la API de OpenAI cuesta cierta cantidad de tokens, se creó un GPT desde la página web de ChatGPT y se configuró en él su comportamiento. Esto fue de gran ayuda ya que poco a poco se pudo refinar la manera en que la IA interactuaba con el usuario y el formato de sus respuestas sin tener que hacer multitud de peticiones costosas. Una de las funcionalidades extra añadidas que ha sido posible gracias a la IA, es la de generar un título para cada conversación de manera automática y que esté relacionado con la primera pregunta que haga el usuario al crear un chat.

Una vez ya se habían implementado los chats entre los alumnos, la IA y los profesores, se debatió sobre la funcionalidad de que los usuarios puedan subir imágenes a la aplicación para que el modelo las analice y proporcione una explicación sobre ellas. Esta funcionalidad presenta un inconveniente de relativa importancia, ya que se tendría que usar un modelo que permita este cometido y actualmente las peticiones a ChatGPT-4 Turbo suponen un gasto importante de dinero (hasta 900 veces más que ChatGPT- 3.5 por cada millón de tokens), por este motivo se ha implementado de manera superficial, pero haría falta refinar ciertos aspectos tanto en la vista de usuario como en el código del proyecto.

## Gestión del proyecto

### Recursos utilizados

Entre los recursos que se han empleado para desarrollar el trabajo se encuentran los recursos humanos, los cuales estarían formados por el personal involucrado en el proyecto, definiendo sus roles y responsabilidades. Aplicando lo que se ha descrito antes a este proyecto, el equipo de desarrollo estaría formado por una única persona (el estudiante), su rol sería el de desarrollador principal y único de la aplicación y tendría a su cargo todas las responsabilidades de creación del proyecto.

En el caso de los recursos materiales hardware que han sido necesarios para crear la aplicación se puede incluir el ordenador portátil del estudiante, en el que se ha tenido que instalar una serie de recursos software que se explicarán a continuación. Como recurso material software se pueden incluir los distintos programas como el IDE, navegador web y programas de gestión de proyectos como Excel, el cual se ha utilizado para planificar y estimar el tiempo de todas las tareas.

Otro de los tipos de recursos involucrados en el proyecto han sido los financieros, debido a que se necesita tener cierta cantidad de dinero disponible en la cuenta de OpenAI desde la que se hagan las peticiones a la API de ChatGPT. Este recurso delimita en gran medida la escalabilidad que tenga el proyecto, ya que si se cuenta con una gran cantidad de dinero se podrán desarrollar funcionalidades como la del análisis de imágenes, análisis de audio y la posibilidad de implementar el *Text-to-Speech* para convertir el texto en audio.

### Planificación temporal y diagrama de Gantt

Debido a que la metodología que se ha usado para desarrollar el trabajo ha sido SCRUM, durante los primeros días se hicieron actividades relacionadas con la identificación y documentación de los requisitos. Posteriormente, se empezó con la fase de análisis, en la que se hizo el modelado del dominio (diagramas de clase), la identificación de los recursos y el análisis de los riesgos. La siguiente fase fue la del diseño, donde durante un periodo de una semana se hicieron los *wireframes* y *mockups* para tener una idea de cómo iban a ser los flujos de interacción dentro de la aplicación y el aspecto que tendría la interfaz. Por último, la fase en la que se ha invertido más tiempo es en la de implementación, donde aparte de definir cómo iba a ser la arquitectura del sistema, hacer los diagramas de flujo y modelar los datos, también se han desarrollado la parte del *backend* y *frontend* del proyecto de manera paralela. Después de tener implementadas ciertas funcionalidades también se han invertido varios días para solucionar ciertos errores que se iban presentando. Debido a que el equipo constaba de una sola persona, las últimas tres tareas han sido paralelas. A continuación (ver Figuras 40 y 41), se pueden ver los detalles de la planificación temporal del proyecto y el diagrama de Gantt correspondiente.

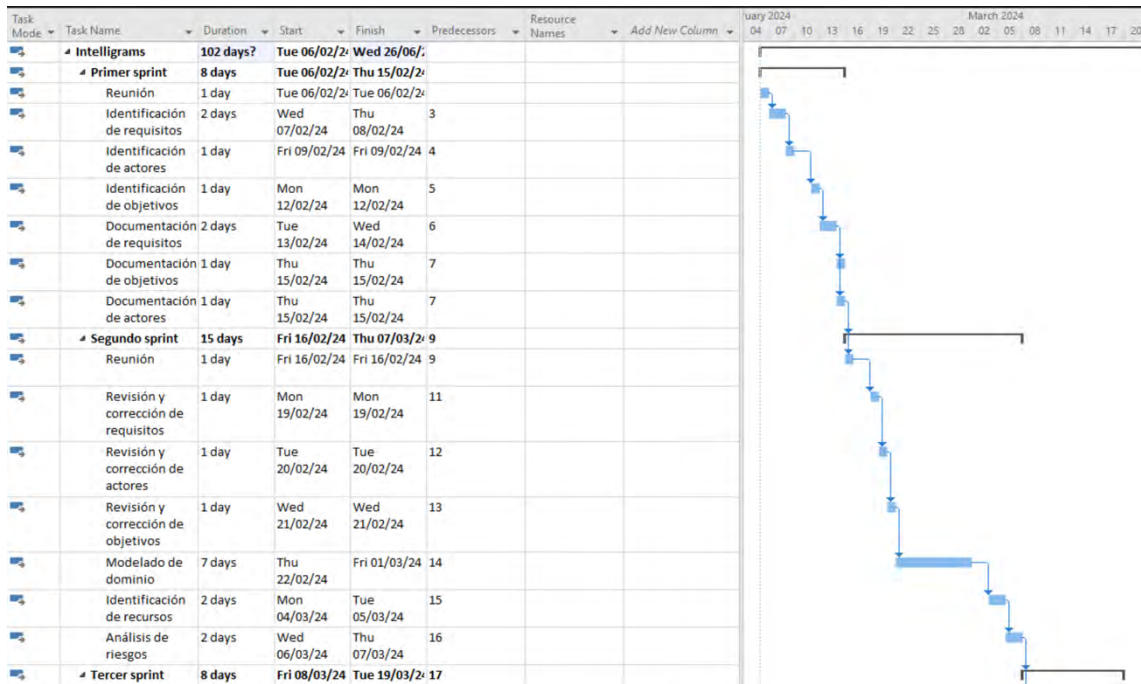


Figura 40. Primera parte del diagrama de la planificación temporal y el diagrama de Gantt

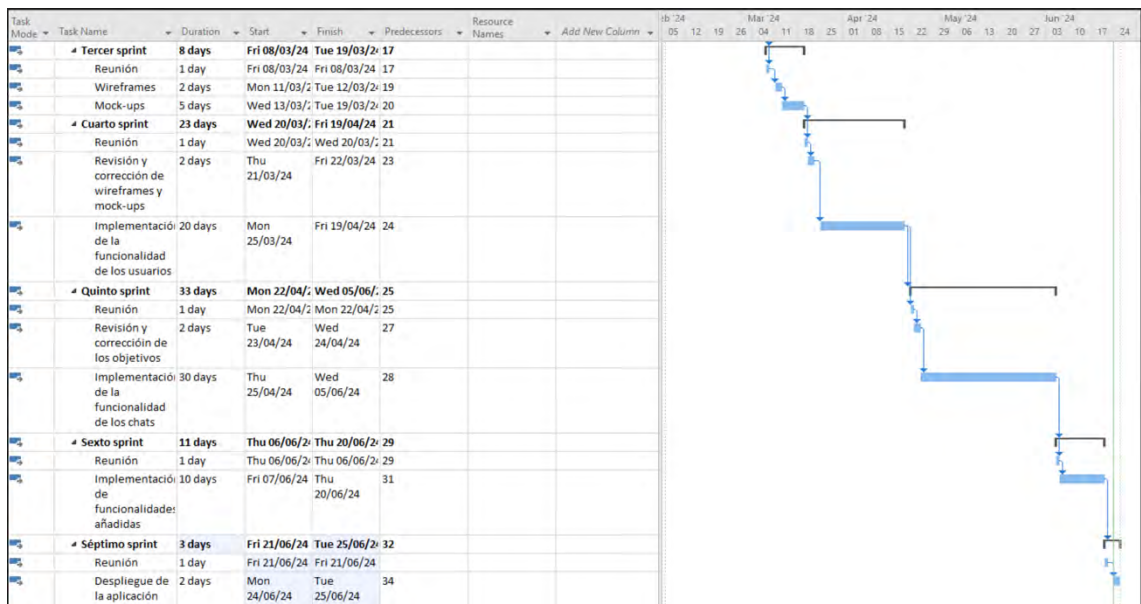


Figura 41. Segunda parte del diagrama de la planificación temporal y el diagrama de Gantt

## Conclusiones y trabajo futuro

Después de haber concluido con el proyecto y haber realizado su correspondiente documentación, se puede observar que se han podido cumplir los requisitos planteados, tanto los que se pueden encontrar en la propuesta inicial de este trabajo, como los que se han ido identificando a lo largo del ciclo de desarrollo.

El primero de los requisitos u objetivos funcionales que se ha logrado es el de la generación de problemas de modelado de dominio mediante IA, el cual ha sido posible haciendo uso de modelos de generación de texto como ChatGPT. Si bien ha sido relativamente fácil de implementar, se han tenido que adquirir una cierta cantidad de conocimientos suficientes sobre este tipo de modelos como, por ejemplo, el estudio sobre cómo hacer las peticiones correctas a la IA para obtener los resultados esperados (*prompting*) o el hecho de poder transformar el texto que se le proporcione en estructuras de datos como los vectores matemáticos con un número “n” de dimensiones para que el modelo pueda devolver lo que más se adecúe con la solicitud (*embeddings*).

Como en un primer momento solo iba a haber comunicación entre el usuario y la IA, se decidió hacer un sistema de peticiones síncrono que respondiese de una determinada manera cuando el usuario hiciese una petición. Más tarde, el concepto de realimentación se cambió, debido al factor del tiempo se tuvo que adaptar la estructura de la funcionalidad ya existente. Este es el motivo por el que no se usa el protocolo de *websockets* a la hora de acceder y escribir en un chat.

El siguiente de los objetivos que se ha desarrollado con éxito es el de una interfaz que permita al alumnado plantear diagramas de clases en UML para solucionar los problemas planteados previamente. Aunque es una funcionalidad que se puede ver afectada por los factores económicos, el usuario tiene la capacidad de adjuntar imágenes en los mensajes que manda al chat con el fin de que la IA las analice y proporcione una respuesta.

El tercer objetivo es el relativo a la generación de los diagramas de clase, en el que se ha tenido que optar por no usar el sistema de generación de imágenes que incluyen los modelos de OpenAI y se ha tenido que usar una librería de Python externa. De esta manera las imágenes de los diagramas son exactas y se ciñen a los parámetros que contiene el código en formato PlantUML.

Debido a que este proyecto implicaba el desarrollo integral de una aplicación web, ha sido necesario adquirir una serie de conocimientos que han servido de gran ayuda. Desde la creación del sistema de gestión de los usuarios a lograr comunicar la parte del *frontend* con la del *backend* para implementar todas las funcionalidades que componen la aplicación. Este trabajo ha supuesto un punto de partida desde donde iniciar para poder crear futuros proyectos haciendo uso de tecnologías innovadoras como la IA, la cual permite la creación de herramientas que pueden usarse en cualquier tipo de sector.

Una de las posibles funcionalidades que podría implementar en la aplicación es la de una interfaz de usuario por voz (VUI) que permita una comunicación más natural entre el modelo y el usuario, esto sería posible gracias a los nuevos modelos que está desarrollando la empresa OpenAI como ChatGPT-4o. Otra de las funcionalidades que podría implementar es la del análisis en tiempo real de la pantalla del usuario por parte de la IA para que sin necesidad de tener que escribir y solo mostrándole la imagen de un diagrama de clases, se analice y proporcione una explicación de cada uno de los elementos que lo componen. En lo que respecta a los problemas que se me han presentado a la hora de desarrollar las funcionalidades que involucren a la IA, puedo destacar el análisis de las imágenes que sube el usuario a la aplicación.

Como he explicado en puntos anteriores, las solicitudes a modelos que permiten hacer este tipo de acciones son muy costosas y pueden suponer un gran inconveniente en el caso que no se tengan los suficientes fondos para hacer frente a todas las peticiones que hagan los usuarios.

Otra de las mejoras a futuro que podría añadirse a la aplicación, sería cambiar la lógica con la que se reciben las preguntas del usuario, cómo responde la IA a esas peticiones y la unión de 2 usuarios al chat simultáneamente para que no se tenga que deshabilitar la escritura de mensajes del usuario que manda el chat para su revisión hasta que el profesor proporcione el comentario de realimentación. Haciendo uso de funciones asíncronas también se podría actualizar la vista del usuario en tiempo real para notificar de los posibles eventos en los chats.

## Referencias

*Beneficios de la IA en la educación* -. (s. f.). Recuperado 13 de junio de 2024, de

<https://www.smileandlearn.com/beneficios-de-la-ia-en-la-educacion/>

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd, Ed.). Addison-Wesley.

Bucchiarone, A., Vázquez-Ingelmo, A., Schiavo, G., Barandoni, S., García-Holgado, A., García-Peñalvo, F. J., Mosser, S., Pierantonio, A., Zschaler, S., & Barnett, W. (2023). Towards Personalized Learning Paths to Empower Competence Development in Model Driven Engineering Through the ENCORE Platform. In *Proceedings 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C 2023)* (Västerås, Sweden, 1-6 October 2023) (pp. 122-129). IEEE Computer Society. <https://doi.org/10.1109/MODELS-C59198.2023.00036>

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-oriented software architecture. A system of patterns. Volume 1.* John Wiley & Sons.

Cuofano, G. (2024, 22 de enero). ¿Cómo funciona ChatGPT? FourWeekMBA. <https://d66z.short.gy/mcZLX0>

European Commission. (2020). *Communication from the Commission to the European Parliament and the Council: Data protection as a pillar of citizens' empowerment and the EU's approach to the digital transition - two years of application of the General Data Protection Regulation.* <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX:52020DC0264>

Florencia Melo, M. (2023, 26 de septiembre). Inteligencia artificial: ¿importante en la enseñanza? *statista*. <https://d66z.short.gy/rdt9rx>

García-Holgado, L., Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., Yiğit, Ö., Ramón-Navarro, C., Pasin, D. G., Aktug, C., & Greilich, A. (2023). Gamified Values Education for Fostering Migrant Integration at Schools. In *Proceedings TEEM 2022: Tenth International Conference on Technological Ecosystems for Enhancing Multiculturality*. Salamanca, Spain, October 19–21, 2022 (pp. 1117–1123). Springer Nature. [https://doi.org/10.1007/978-981-99-0942-1\\_118](https://doi.org/10.1007/978-981-99-0942-1_118)

*García-Peñalvo, F. J., Rodríguez-Conde, M. J., Therón, R., García-Holgado, A., Martínez-Abad, F., & Benito-Santos, A. (2019). Grupo GRIAL. IE Comunicaciones. Revista Iberoamericana de Informática Educativa(30), 33-48. <https://bit.ly/35lIQh9>*

García-Peñalvo, F. J., García-Holgado, A., & Vázquez-Ingelmo, A. (2021). Scrum. In F. J. García-Peñalvo, A. García-Holgado, & A. Vázquez-Ingelmo (Eds.), *Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2020-2021*. Grupo GRIAL, Universidad de Salamanca. <https://doi.org/10.5281/zenodo.5781344>

García-Peñalvo, F. J., García-Holgado, A., & Vázquez-Ingelmo, A. (2024). Modelo de Dominio. In F. J. García-Peñalvo, A. García-Holgado, & A. Vázquez-Ingelmo (Eds.), *Recursos docentes de la asignatura Ingeniería de Software I. Grado en Ingeniería Informática. Curso 2023-2024*. Grupo GRIAL. <https://doi.org/10.5281/zenodo.10963986>

García-Peñalvo, F. J., Llorens-Largo, F., & Vidal, J. (2024). The new reality of education in the face of advances in generative artificial intelligence. *RIED*:



Cristian Alejandro Castillo Salguero

revista iberoamericana de educación a distancia, 27(1), 9–39.

<https://doi.org/10.5944/ried.27.1.37716>

Giraldez, A. (2024, 22 de junio). El error como oportunidad de aprendizaje. ¿Y si

dejamos de castigar los errores? Educación 3.0.

<https://d66z.short.gy/sj2nu0>

*Hostilica*. (2024). *HTTP vs HTTPS*. *Hostilica*. <https://d66z.short.gy/8GJniF>

Moreno García, M. N. (2024). Estimación del esfuerzo. Recursos docentes de la asignatura Gestión de Proyectos. Grado en Ingeniería Informática. Curso 2023-2024.

[https://studium23.usal.es/pluginfile.php/195085/mod\\_resource/content/1/GP-GII\\_Primer\\_practica.pdf](https://studium23.usal.es/pluginfile.php/195085/mod_resource/content/1/GP-GII_Primer_practica.pdf)

Infante Montero, S. (2012, 30 de abril). Curso Django: Entendiendo como trabaja

Django. Maestros del Web. <https://d66z.short.gy/rOOTjJ>

Mozilla. (2024). Aprende desarrollo web. <https://d66z.short.gy/xF3cIE>

OpenAI. (2023). *DALL·E3 System Card*. OpenAI. <https://bit.ly/3vsU5fv>

PlantUML. (2024). FAQ. PlantUML. <https://plantuml.com/es-dark/faq>

Schwaber, K. (1997). SCRUM Development Process. In J. Sutherland, C.

Casanave, J. Miller, P. Patel, & G. Hollowell (Eds.), *Business Object*

*Design and Implementation*. OOPSLA '95 Workshop Proceedings 16

October 1995, Austin, Texas (pp. 117-134). Springer London.

[https://doi.org/10.1007/978-1-4471-0947-1\\_11](https://doi.org/10.1007/978-1-4471-0947-1_11)

Trivedi, S. (2023, 6 de abril). Cómo fomentar la participación de los estudiantes

en las aulas en línea. Moodle. <https://d66z.short.gy/v3ukHn>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need. arXiv, Article arXiv:1706.03762v7. <https://doi.org/10.48550/arXiv.1706.03762>

## Anexos

### Anexo 1: Especificaciones del sistema

En las primeras fases del proyecto se han tenido que definir los requisitos del sistema en base a los objetivos que se plantearon en un primer momento en la propuesta del trabajo. A continuación, se van a incluir las tablas de cada uno de los requisitos identificados:

#### Requisitos funcionales del sistema

<b>CU-01</b>	<b>Registrarse</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-02 Iniciar sesión CU-05 Asignar rol de profesor	
<b>Descripción</b>	El sistema solicita un conjunto de datos necesarios (nombre de usuario, nombre, apellidos, correo electrónico y contraseña) para registrar y crear la cuenta del usuario en el sistema. Posteriormente se iniciará el proceso de verificación del correo electrónico.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.- 4.-	El usuario inicia el proceso de registro pulsando el botón de “Registrarse” de la página de inicio. El sistema redirige a la persona a la página en la que se encuentra el formulario de registro. La persona introduce los datos en el formulario (nombre de usuario, nombre, apellidos y contraseña) y lo envía. El sistema comprueba que los datos introducidos por el usuario (correo electrónico y nombre de usuario) no se corresponden con una cuenta existente en la BD.
<b>Postcondición</b>	El sistema crea la cuenta del nuevo usuario, se le asocia el rol de “Alumno” por defecto y se inicia el proceso de verificación de la dirección de correo electrónico.	

	<b>Paso</b>	<b>Acción</b>
<b>Excepciones</b>	2.-	El sistema determina que alguno/s de los datos introducidos por el usuario son incorrectos y se lo comunica el usuario mediante un mensaje en la página.
	4.-	El sistema determina que los datos introducidos por el usuario (correo electrónico y nombre de usuario) coinciden con los de una cuenta ya existente en la BD y se lo comunica al usuario mediante un mensaje en la página.
<b>Frecuencia esperada</b>	Una sola vez.	

Tabla 1. Caso de uso "Registrarse"

<b>CU-02</b>	<b>Iniciar sesión</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-03 Ver perfil CU-04 Cerrar sesión CU-07 Recuperar contraseña	
<b>Descripción</b>	El sistema solicitará al usuario el correo electrónico y la contraseña para iniciar sesión en la aplicación. Posteriormente comprobará si los datos introducidos son correctos, verificará que ha completado con éxito el proceso de verificación de la dirección de correo electrónico y permitirá el acceso a la aplicación en el caso de que lo sean. Si son erróneos lo comunicará mediante un mensaje en la vista del usuario.	
<b>Precondición</b>	El usuario debe haberse registrado en el sistema previamente y haber completado la verificación del correo electrónico con éxito.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario comienza el proceso de inicio de sesión haciendo clic en el botón de "Iniciar sesión" de la página de inicio.
	2.-	El sistema redirige al usuario a la página a la página en la que se encuentra el formulario de inicio de sesión.

	3.- 4.-	El usuario introduce los datos solicitados en el formulario (correo electrónico y contraseña). El sistema verifica que los datos introducidos por el usuario son correctos y actualiza la vista redirigiendo al usuario a la página principal de “Asistente” dentro de la aplicación.
<b>Postcondición</b>	Si los datos son correctos dirige al usuario a la página de “Asistente” de la aplicación. En caso contrario, dirige al usuario a la página de inicio de sesión en la que estaba para que vuelva a introducir los datos y muestra un mensaje.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4.-	El sistema determina que alguno/s de los datos introducidos son incorrectos redirigiendo al usuario a la página de inicio de sesión y comunicándole mediante un mensaje en la vista.
<b>Frecuencia esperada</b>	Una vez al día.	

Tabla 2. Caso de uso “Iniciar sesión”

<b>CU-03</b>	<b>Ver perfil</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-10 Mostrar información
<b>Descripción</b>	El usuario seleccionará la opción que permite “Ajustes” en el menú lateral de la aplicación y el sistema actualiza la vista mostrando una ventana con los datos correspondientes con la cuenta del usuario que lo ha solicitado como, por ejemplo, el nombre de usuario, nombre, apellidos, rol y dirección de correo electrónico.
<b>Precondición</b>	El usuario ha tenido que iniciar sesión con éxito en la aplicación.

<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.-	El usuario hace clic en el botón de “Ajustes” del menú lateral. El sistema recupera de la BD los datos del usuario (nombre de usuario, nombre, apellidos, correo electrónico y rol) y muestra una ventana modal que contiene los datos recuperados.
<b>Postcondición</b>	El sistema permitirá al usuario modificar su nombre de usuario, su nombre y sus apellidos. Debido a que la dirección de correo electrónico y el rol son parámetros críticos, no se permitirá su modificación, pero si se mostrarán en la ventana.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Una vez a la semana.	

Tabla 3. Caso de uso “Ver perfil”

<b>CU-04</b>	<b>Cerrar sesión</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión
<b>Descripción</b>	El usuario selecciona la opción que le permite cerrar sesión en el menú lateral de la aplicación y el sistema mostrará una ventana en la que se solicitará una confirmación por parte del usuario para cerrar la sesión que tienen activa. En el caso de que el usuario quiera seguir con el proceso se cerrará la sesión y se le redirigirá a la página de inicio. Si decide cancelar el proceso, el sistema ocultará la ventana de confirmación.

<b>Precondición</b>	El usuario ha tenido que iniciar sesión.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario solicita cerrar la sesión que tiene activa
	2.-	El sistema solicita confirmación al usuario mostrando una ventana modal.
	3.-	El usuario confirma su decisión
	4.-	El sistema cierra la sesión del usuario y le redirige a la página de inicio.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3.-	El usuario decide cancelar la operación sin llegar al paso número 4 haciendo clic en el botón de "Cancelar" de la ventana de confirmación.
<b>Frecuencia esperada</b>	Una vez al mes.	

Tabla 4. Caso de uso "Cerrar sesión"

<b>CU-05</b>	<b>Asignar rol de profesor</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios
<b>Requisitos asociados</b>	<p>IRQ-01 Información de los usuarios                  IRQ-02 Información de los chats                  IRQ-03 Información de los mensajes                  NFR-01 Usabilidad                  NFR-02 Rendimiento                  NFR-03 Escalabilidad                  NFR-04 Seguridad                  NFR-05 Compatibilidad                  NFR-06 Portabilidad                  CU-01 Registrarse                  CU-02 Iniciar sesión</p>
<b>Descripción</b>	El administrador accederá al panel de administración de la aplicación iniciando sesión con su nombre de usuario y su contraseña. Posteriormente, accederá a la sección de "Usuarios" y buscará la cuenta del usuario deseada. Seleccionará la cuenta y hará clic en la opción de "Rol" y cambiará el rol de "Alumno" a "Profesor".

<b>Precondición</b>	El usuario ha tenido que registrarse en la aplicación con éxito.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El administrador accede al panel de administración de la aplicación con sus datos (nombre de usuario y contraseña).
	2.-	El sistema actualiza la vista y muestra los servicios de administración disponibles.
	3.-	El administrador accede al apartado de “Usuarios” del panel de administración.
	4.-	El sistema recupera de la BD la lista de usuarios existentes en el sistema.
	5.-	El administrador busca y selecciona la cuenta deseada.
	6.-	El administrador hace clic en la opción de “Rol”, cambia el rol de “Alumno” a “Profesor” y guarda los cambios haciendo clic en la opción de “Guardar”.
	7.-	El sistema guarda los cambios en la BD.
<b>Postcondición</b>	Se actualiza el rol de la cuenta del usuario y se actualiza la vista en la aplicación.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Una sola vez.	

Tabla 5. Caso de uso “Asignar rol de profesor”

<b>CU-06</b>	<b>Dar de baja un usuario</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión



<b>Descripción</b>	El administrador accederá al panel de administración de la aplicación iniciando sesión con su nombre de usuario y su contraseña. Posteriormente, accederá a la sección de “Usuarios” y buscará la cuenta del usuario deseada. Seleccionará la cuenta y hará clic en la opción de “Delete”.	
<b>Precondición</b>	El usuario ha tenido que registrarse en la aplicación.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El administrador accede al panel de administración de la aplicación con sus datos (nombre de usuario y contraseña).
	2.-	El sistema actualiza la vista y muestra los servicios de administración disponibles.
	3.-	El administrador accede al apartado de “Usuarios” del panel de administración.
	4.-	El sistema recupera de la BD la lista de usuarios existentes en el sistema.
	5.-	El administrador busca y hace clic en la cuenta que desea eliminar.
	6.-	El administrador hace clic en la opción “Delete” y confirma su decisión.
	7.-	El sistema elimina la cuenta de la BD.
<b>Postcondición</b>	El sistema elimina la cuenta del usuario.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	6.-	El administrador decide cancelar su decisión y no se elimina la cuenta del usuario.
<b>Frecuencia esperada</b>	Sin especificar.	

Tabla 6. Caso de uso “Dar de baja un usuario”

<b>CU-07</b>	<b>Recuperar contraseña</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios

<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-08 Enviar email	
<b>Descripción</b>	El usuario decide iniciar el proceso de recuperación de su contraseña haciendo clic en el enlace de “He olvidado mi contraseña” de la página de inicio de la aplicación. Posteriormente, se le solicitará el correo electrónico asociado su cuenta y se le mandará un email. Dicho email contendrá un enlace que redirigirá al usuario a la página de Django para introducir su nueva contraseña dos veces. Por último, se le indicará mediante un mensaje en pantalla que su contraseña se ha modificado correctamente y que puede iniciar sesión.	
<b>Precondición</b>	El usuario ha tenido que seleccionar la opción de “He olvidado mi contraseña” en el formulario de inicio de sesión.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario selecciona la opción de “He olvidado mi contraseña” en el formulario de inicio de sesión.
	2.-	El sistema redirige al usuario a la página en la que se le solicita el correo electrónico de la cuenta.
	3.-	El usuario introduce la dirección de correo electrónico.
	4.-	El sistema manda un correo electrónico al usuario con un enlace.
	5.-	El usuario hace clic en el enlace y se le redirige a la página en la que puede introducir su nueva contraseña dos veces para confirmar el cambio.
	6.-	El usuario modifica su contraseña.
	7.-	El sistema actualiza la contraseña asociada a la cuenta en la BD y comunica al usuario que puede iniciar sesión con la nueva contraseña.
<b>Postcondición</b>	El usuario accede a la aplicación con su nueva contraseña.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Sin especificar.	

Tabla 7. Caso de uso “Recuperar contraseña”

<b>CU-08</b>	<b>Enviar email</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-07 Recuperar contraseña	
<b>Descripción</b>	El sistema puede enviar un email al usuario en dos ocasiones, para confirmar su dirección de correo electrónico a la hora de registrarse en la aplicación o para reiniciar su contraseña. Dichos emails contendrán un enlace en el que el usuario tendrá que hacer clic y se le redirigirá a la página que corresponda en cada caso.	
<b>Precondición</b>	El usuario ha tenido que iniciar el proceso de registro en la aplicación o de recuperación de contraseña.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario inicia el proceso de registro o de recuperación de contraseña.
	2.-	El usuario introduce los datos necesarios (correo electrónico) para que el sistema le mande un email.
	3.-	El sistema envía un email a la dirección de correo electrónico que ha proporcionado el usuario.
<b>Postcondición</b>	El usuario accede al email que se ha enviado y hace clic en el enlace que contiene.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Sin especificar.	

Tabla 8. Caso de uso "Enviar email"

<b>CU-09</b>	<b>Modificar datos</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-08 Enviar email CU-10 Mostrar información	
<b>Descripción</b>	El usuario decide modificar alguno de los datos de su cuenta haciendo clic en la opción de “Ajustes” del menú lateral o iniciando el proceso de recuperación de contraseña. Para la primera opción el sistema muestra una ventana con los datos de la cuenta y el usuario modifica los que estén disponibles. Para confirmar su decisión el usuario hace clic en el botón de “Guardar”. Para la segunda opción el usuario inicia el proceso de recuperación de contraseña haciendo clic en el enlace de “He olvidado mi contraseña” en la página de inicio de sesión.	
<b>Precondición</b>	El usuario ha tenido que seleccionar la opción de “He olvidado mi contraseña” de la página de inicio de sesión o “Ajustes” en el menú lateral de la aplicación.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.- 4.-	El sistema muestra una vista al usuario que permite modificar alguno/s de sus datos. El usuario introduce los datos que desee modificar. El usuario confirma su decisión. El sistema actualiza los nuevos datos en la BD y actualiza la vista.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Sin especificar.	

Tabla 9. Caso de uso “Modificar datos”

<b>CU-10</b>	<b>Mostrar información</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-09 Modificar datos	
<b>Descripción</b>	El sistema mostrará información al usuario después de que este solicite ver los datos de su cuenta o después de seleccionar la opción de “Guardar” en el apartado de “Ajustes” del menú lateral de la aplicación.	
<b>Precondición</b>	El usuario ha tenido que seleccionar la opción de “Guardar” en la ventana de ajustes o ha solicitado al sistema ver los datos de su cuenta.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.-	El usuario selecciona alguna de las opciones que implica visualizar los datos. El sistema recupera de la BD los datos del usuario que lo ha solicitado. El sistema actualiza la vista y muestra los datos recuperados de la cuenta del usuario.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Sin especificar.	

Tabla 10. Caso de uso “Mostrar información”

<b>CU-11</b>	<b>Iniciar conversación</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-13 Guardar conversación CU-14 Generar respuesta	
<b>Descripción</b>	El usuario accede al apartado de “Asistente” de la aplicación y escribe la pregunta o solicitud que quiera hacer a la IA. Posteriormente, la envía y se le redirige a la vista de la conversación que acaba de iniciar.	
<b>Precondición</b>	El usuario ha tenido que iniciar sesión en la aplicación para poder acceder a esta funcionalidad.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario accede al apartado “Asistente” desde el menú lateral de la aplicación.
	2.-	El usuario introduce una pregunta o solicitud y la envía.
	3.-	El sistema crea un chat y lo asocia con el primer mensaje que ha mandado el usuario.
	4.-	El sistema redirige al usuario a la vista del chat que acaba de crear.
<b>Postcondición</b>	La IA analiza la solicitud o pregunta del usuario, genera una respuesta acorde con ella y la envía al chat.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2.-	Hay un error y el mensaje no se envía correctamente.
	3.-	El usuario vuelve a mandar el mensaje.
<b>Frecuencia esperada</b>	Varias veces al día	

Tabla 11. Caso de uso “Iniciar conversación”

<b>CU-12</b>	<b>Eliminar conversación</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación	
<b>Descripción</b>	El usuario decide eliminar uno de los chats que tiene asociados a su cuenta. Accede al apartado de “Mis chats” del menú lateral de la aplicación y hace clic en el icono de borrar que se encuentra a la derecha de cada chat. Se solicita al usuario una confirmación de que desea eliminar el chat mediante una ventana modal. En caso afirmativo, se elimina el chat y se actualiza la vista y en caso negativo, la ventana de confirmación desaparece.	
<b>Precondición</b>	Para poder eliminar una conversación el usuario ha tenido que iniciar como mínimo una previamente.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario accede al apartado de “Mis chats” del menú lateral de la aplicación.
	2.-	El sistema recupera de la BD las conversaciones asociadas a la cuenta del usuario y las muestra en forma de lista.
	3.-	El usuario hace clic en el icono de borrar que se encuentra a la derecha de cada chat.
	4.-	El sistema solicita una confirmación al usuario mediante una ventana modal.
	5.-	El usuario confirma su decisión.
	6.-	El sistema elimina la conversación de la BD y actualiza la vista.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4.-	El usuario decide no eliminar el chat y la ventana desaparece.
<b>Frecuencia esperada</b>	Varias veces a la semana.	

Tabla 12. Caso de uso “Eliminar conversación”

<b>CU-13</b>	<b>Guardar conversación</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación	
<b>Descripción</b>	Una vez el usuario ha mandado un mensaje desde el apartado de “Asistente”, el sistema crea y almacena una conversación entre el usuario y la IA. El sistema genera un título de forma automática con ayuda de la IA y guarda tanto la solicitud del usuario como la respuesta de la IA.	
<b>Precondición</b>	El usuario ha tenido que iniciar sesión dentro de la aplicación y haber escrito una pregunta o solicitud desde el apartado de “Asistente”.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario manda el primer mensaje de la conversación.
	2.-	El sistema crea y almacena la conversación en la BD.
	3.-	El sistema genera un título para la conversación con ayuda de la IA.
	4.-	El sistema guarda el primer mensaje del usuario y la respuesta de la IA.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1.-	Hay un error y el mensaje del usuario no se envía correctamente.
	2.-	El usuario vuelve a mandar el mensaje.
<b>Frecuencia esperada</b>	Varias veces al día.	

Tabla 13. Caso de uso “Guardar conversación”



<b>CU-14</b>	<b>Generar respuesta</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-02 Iniciar sesión CU-11 Iniciar conversación CU-15 Generar enunciado CU-16 Generar diagrama	
<b>Descripción</b>	Después de que el usuario haya mandado el primer mensaje del chat desde el apartado de “Asistente” o mensajes posteriores desde la vista del chat, se mandará dicho mensaje a la IA para que genere una respuesta. Dependiendo del tipo de solicitud, la IA podrá generar una respuesta, un enunciado de un problema o un diagrama de clases.	
<b>Precondición</b>	El usuario ha tenido que mandar el primer mensaje de la conversación desde el apartado de “Asistente” o un mensaje posterior desde la vista del chat.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El sistema manda a la IA el mensaje que acaba de escribir el usuario en el chat.
	2.-	La IA genera una respuesta acorde con la solicitud del usuario.
	3.-	El sistema crea un mensaje con la respuesta de la IA y se asocia con el chat.
<b>Postcondición</b>	Se actualiza la vista del chat mostrando la respuesta de la IA.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1.-	Hay un error y la respuesta no se genera correctamente.
	2.-	El usuario vuelve a mandar el mensaje
	3.-	La IA genera la respuesta otra vez
<b>Frecuencia esperada</b>	Varias veces al día.	

Tabla 14. Caso de uso “Generar respuesta”

<b>CU-15</b>	<b>Generar enunciado</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-14 Generar respuesta	
<b>Descripción</b>	El usuario ha mandado un mensaje al chat previamente solicitando el enunciado de un problema que se resuelva con un diagrama de clases UML. La IA analiza la petición del usuario y genera el enunciado del problema.	
<b>Precondición</b>	El usuario ha tenido que mandar un mensaje al chat solicitando el enunciado de un problema.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.- 4.- 5.-	El sistema manda a la IA el mensaje que acaba de escribir el usuario en el chat. La IA analiza la solicitud del usuario. La IA genera el enunciado acorde con el tema que ha proporcionado el usuario. La IA manda el enunciado al chat en forma de mensaje. El sistema actualiza la vista del chat mostrando el nuevo mensaje de la IA.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2.- 3.-	La IA determina que el usuario no ha proporcionado un tema sobre el que generar el enunciado. La IA solicita un tema al usuario.
<b>Frecuencia esperada</b>	Varias veces al día.	

Tabla 15. Caso de uso “Generar enunciado”

<b>CU-16</b>	<b>Generar diagrama</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación CU-14 Generar respuesta	
<b>Descripción</b>	El usuario manda un mensaje al chat solicitando el diagrama de clases UML que sirve de solución a un enunciado generado anteriormente. La IA analiza la petición del usuario y genera el diagrama de clases.	
<b>Precondición</b>	El usuario ha tenido que mandar un mensaje al chat solicitando el diagrama de clases de un enunciado previo.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	La IA analiza la solicitud del usuario y genera el código PlantUML del diagrama de clases.
	2.-	El sistema utiliza el código generado por la IA para generar la imagen del diagrama de clases.
	3.-	El sistema crea un mensaje que contiene la imagen del diagrama de clases.
	5.-	El sistema asocia el mensaje al chat del usuario.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2.-	Hay un problema a la hora de generar la imagen del diagrama de clases.
	3.-	El usuario vuelve a solicitar el diagrama de clases
<b>Frecuencia esperada</b>	Varias veces al día.	

Tabla 16. Caso de uso “Generar diagrama”

<b>CU-17</b>	<b>Enviar chat para feedback</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación CU-19 Revisar chat y dar <i>feedback</i>	
<b>Descripción</b>	El usuario accede a uno de los chats que tiene asociados a su cuenta desde el apartado de “Mis chats”. Posteriormente, hará clic en el botón de solicitar <i>feedback</i> que se encuentra en la parte superior derecha de la vista. El sistema seleccionará a uno de los usuarios que tengan el rol de “Profesor” de forma aleatoria y se le asignará el chat. La vista del usuario se actualizará y se deshabilitará la parte de la interfaz donde el usuario puede escribir los mensajes.	
<b>Precondición</b>	El usuario ha tenido que iniciar una conversación con la IA previamente.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.- 4.- 5.- 6.-	El usuario accede al apartado de “Mis chats” del menú lateral de la aplicación. El sistema recupera de la BD los chats asociados a la cuenta del usuario y los muestra en forma de lista. El usuario accede a uno de los chats disponibles para solicitar <i>feedback</i> por parte del profesor. El usuario pulsa el botón que se encuentra en la parte superior derecha de la vista del chat. El sistema selecciona y asigna el chat a uno de los usuarios con el rol de “Profesor” de manera aleatoria. Se actualiza la vista del usuario y se deshabilita el apartado de la interfaz que permite mandar mensajes.
<b>Postcondición</b>	El profesor al que se le asigne el chat podrá acceder al mismo desde el apartado de “Feedback” del menú lateral de la aplicación.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>Frecuencia esperada</b>	Una vez al día.
----------------------------	-----------------

Tabla 17. Caso de uso “Enviar chat para *feedback*”

<b>CU-18</b>	<b>Ver conversaciones</b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación	
<b>Descripción</b>	El usuario accede al apartado de “Mis chats” disponible en el menú lateral de la aplicación. La vista se actualiza y muestra los chats del usuario.	
<b>Precondición</b>	El usuario ha tenido que iniciar una conversación como mínimo para ver algún chat. En caso contrario se indicará con un mensaje en la misma vista.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario selecciona el apartado de “Mis Chats” en el menú lateral de la aplicación.
	2.-	El sistema recupera de la BD los chats asociados a la cuenta del usuario.
	3.-	La vista del usuario se actualiza y muestra los chats que tenga asociados a su cuenta.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El usuario no tiene ningún chat asociado a su cuenta.
	2.-	El sistema indica al usuario que no tiene ningún chat asociado a su cuenta mediante un mensaje en la vista.
<b>Frecuencia esperada</b>	Varias veces al día.	

Tabla 18. Caso de uso “Ver conversaciones”

<b>CU-19</b>	<b>Revisar chat y dar <i>feedback</i></b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación CU-17 Enviar chat para <i>feedback</i>	
<b>Descripción</b>	El profesor accede al apartado de “Feedback” disponible en el menú lateral de la aplicación para aquellos usuarios con el rol de “Profesor”. Se actualiza la vista mostrando todas las conversaciones que se le hayan asignado de manera aleatoria para proporcionar <i>feedback</i> . El profesor accede a uno de los chats y manda un mensaje. Por último, cuando el usuario se salga de la vista del chat, este desaparecerá de la lista de chats pendientes de revisar.	
<b>Precondición</b>	El usuario ha tenido que solicitar <i>feedback</i> sobre un chat previamente.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.-	El profesor accede al apartado de “Feedback” del menú lateral de la aplicación.
	2.-	El sistema recupera de la BD los chats de los alumnos que se han asignado a la cuenta del profesor para dar <i>feedback</i> y los muestra en forma de lista.
	3.-	El profesor selecciona el chat en la lista de chats pendientes de revisar.
	4.-	El sistema muestra los mensajes del chat.
	5.-	El profesor manda un mensaje de <i>feedback</i> .
	6.-	El sistema modifica el estado del chat pasando de “Pendiente” a “Tiene <i>feedback</i> ”
<b>Postcondición</b>	El sistema indica al usuario que el profesor ha mandado un mensaje resaltando el chat en la lista de chats del apartado “Mis chats”.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Varias veces a la semana.	

Tabla 19. Caso de uso “Revisar chat y dar *feedback*”

<b>CU-20</b>	<b>Ver <i>feedback</i></b>	
<b>Objetivos asociados</b>	OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-11 Iniciar conversación CU-17 Enviar chat para <i>feedback</i>	
<b>Descripción</b>	El usuario accede al apartado de “Mis chats” disponible en el menú lateral de la aplicación. En el caso de que un profesor haya mandado un mensaje de <i>feedback</i> en el chat se le indicará resaltando el chat sobre los demás. Por último, el usuario accede a la conversación para ver el mensaje y el indicador del chat desaparece.	
<b>Precondición</b>	El usuario ha tenido que solicitar <i>feedback</i> sobre uno de sus chats.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1.- 2.- 3.- 4.-	El usuario accede al apartado de “Mis chats”. El sistema recupera de la BD los chats asociados a la cuenta del usuario y los muestra en forma de lista. El usuario selecciona el chat que tiene <i>feedback</i> por parte del profesor. El sistema actualiza la vista y muestra los mensajes del chat.
<b>Postcondición</b>	El sistema actualiza la lista de chats eliminando el indicador de <i>feedback</i> del chat.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Frecuencia esperada</b>	Varias veces a la semana.	

Tabla 20. Caso de uso “Ver feedback”

## Requisitos no funcionales

<b>NFR-01</b>	<b>Usabilidad</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-04 Cerrar sesión CU-05 Asignar rol de profesor CU-06 Dar de baja un usuario CU-07 Recuperar contraseña CU-08 Enviar email CU-09 Modificar datos CU-10 Mostrar información CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>
<b>Descripción</b>	La aplicación contará con una interfaz intuitiva y familiar para el usuario que no requiera un largo periodo de aprendizaje y adaptación.

Tabla 21. Requisito no funcional “Usabilidad”



<b>NFR-02</b>	<b>Rendimiento</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>
<b>Descripción</b>	El modelo de IA que se ha integrado debe de generar las respuestas en un corto periodo de tiempo para poder mantener una interacción fluida con el usuario.

Tabla 22. Requisito no funcional “Rendimiento”

<b>NFR-03</b>	<b>Escalabilidad</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-04 Cerrar sesión CU-05 Asignar rol de profesor CU-06 Dar de baja un usuario CU-07 Recuperar contraseña CU-08 Enviar email CU-09 Modificar datos CU-10 Mostrar información CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>
<b>Descripción</b>	El sistema permitirá la incorporación de nuevas funcionalidades/servicios y número de usuarios sin que sin que afecten a su rendimiento.

Tabla 23. Requisito no funcional “Escalabilidad”

<b>NFR-04</b>	<b>Seguridad</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-04 Cerrar sesión CU-05 Asignar rol de profesor CU-06 Dar de baja un usuario CU-07 Recuperar contraseña CU-08 Enviar email CU-09 Modificar datos CU-10 Mostrar información CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>
<b>Descripción</b>	Se implementarán medidas de seguridad para asegurar la privacidad de los datos que se manejan en la aplicación.

Tabla 24. Requisito no funcional “Seguridad”

<b>NFR-05</b>	<b>Compatibilidad</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	<p>IRQ-01 Información de los usuarios          IRQ-02 Información de los chats          IRQ-03 Información de los mensajes          NFR-01 Usabilidad          NFR-02 Rendimiento          NFR-03 Escalabilidad          NFR-04 Seguridad          NFR-05 Compatibilidad          NFR-06 Portabilidad          CU-01 Registrarse          CU-02 Iniciar sesión          CU-03 Ver perfil          CU-04 Cerrar sesión          CU-05 Asignar rol de profesor          CU-06 Dar de baja un usuario          CU-07 Recuperar contraseña          CU-08 Enviar email          CU-09 Modificar datos          CU-10 Mostrar información          CU-11 Iniciar conversación          CU-12 Eliminar conversación          CU-13 Guardar conversación          CU-14 Generar respuesta          CU-15 Generar enunciado          CU-16 Generar diagrama          CU-17 Enviar chat para <i>feedback</i>          CU-18 Ver conversaciones          CU-19 Revisar chat y dar <i>feedback</i>          CU-20 Ver <i>feedback</i></p>
<b>Descripción</b>	El sistema debe de poder integrarse en las plataformas correspondientes que permitan el acceso de usuarios sin ningún tipo de problema.

Tabla 25. Requisito no funcional “Compatibilidad”

<b>NFR-06</b>	<b>Portabilidad</b>
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-04 Cerrar sesión CU-05 Asignar rol de profesor CU-06 Dar de baja un usuario CU-07 Recuperar contraseña CU-08 Enviar email CU-09 Modificar datos CU-10 Mostrar información CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>
<b>Descripción</b>	Al ser una aplicación web, el sistema podrá utilizarse desde cualquier tipo de sistema operativo y dispositivo sin necesidad de una configuración previa de estos.

Tabla 26. Requisito no funcional “Portabilidad”

## Requisitos de información

<b>IRQ-01</b>	<b>Información de los usuarios</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-02 Información de los chats IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-01 Registrarse CU-02 Iniciar sesión CU-03 Ver perfil CU-04 Cerrar sesión CU-05 Asignar rol de profesor CU-06 Dar de baja un usuario CU-07 Recuperar contraseña CU-08 Enviar email CU-09 Modificar datos CU-10 Mostrar información	
<b>Descripción</b>	El sistema almacenará los datos de los usuarios que se registren en la aplicación.	
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Nombre de usuario</li> <li>• Correo electrónico</li> <li>• Contraseña</li> <li>• Rol</li> </ul>	
<b>Tiempo de vida</b>	<b>Medio</b>	<b>Máximo</b>
	4 años	6 años
<b>Ocurrencias simultáneas</b>	<b>Medio</b>	<b>Máximo</b>
	5000 tuplas	10000 tuplas
<b>Importancia</b>	10	
<b>Urgencia</b>	10	

Tabla 27. Requisito de información “Información de los usuarios”

<b>IRQ-02</b>	<b>Información de los chats</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-03 Información de los mensajes NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>	
<b>Descripción</b>	El sistema almacenará los datos de los chats que creen los usuarios.	
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Propietario</li> <li>• Propietario2</li> <li>• Nombre</li> <li>• ID</li> <li>• Pendiente</li> <li>• <i>Feedback</i></li> <li>• Creado</li> </ul>	
<b>Tiempo de vida</b>	<b>Medio</b>	<b>Máximo</b>
	4 años	6 años
<b>Ocurrencias simultáneas</b>	<b>Medio</b>	<b>Máximo</b>
	5000 tuplas	10000 tuplas
<b>Importancia</b>	10	
<b>Urgencia</b>	10	

Tabla 28. Requisito de información “Información de los chats”

<b>IRQ-03</b>	<b>Información de los mensajes</b>	
<b>Objetivos asociados</b>	OBJ-01 Gestión de los usuarios OBJ-02 Gestión de conversaciones	
<b>Requisitos asociados</b>	IRQ-01 Información de los usuarios IRQ-02 Información de los chats NFR-01 Usabilidad NFR-02 Rendimiento NFR-03 Escalabilidad NFR-04 Seguridad NFR-05 Compatibilidad NFR-06 Portabilidad CU-11 Iniciar conversación CU-12 Eliminar conversación CU-13 Guardar conversación CU-14 Generar respuesta CU-15 Generar enunciado CU-16 Generar diagrama CU-17 Enviar chat para <i>feedback</i> CU-18 Ver conversaciones CU-19 Revisar chat y dar <i>feedback</i> CU-20 Ver <i>feedback</i>	
<b>Descripción</b>	El sistema almacenará los datos de los mensajes que manden los usuarios.	
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Chat</li> <li>• Emisor</li> <li>• Contenido</li> <li>• Imagen</li> <li>• <i>Timestamp</i></li> </ul>	
<b>Tiempo de vida</b>	<b>Medio</b>	<b>Máximo</b>
	4 años	6 años
<b>Ocurrencias simultáneas</b>	<b>Medio</b>	<b>Máximo</b>
	5000 tuplas	10000 tuplas
<b>Importancia</b>	10	
<b>Urgencia</b>	10	

Tabla 29. Requisito de información “Información de los mensajes”



## Diagramas de casos de uso

A la hora de representar y modelar el sistema en las fases iniciales se han tenido que realizar los correspondientes diagramas de casos de uso, formando dos paquetes principales (Gestión de usuarios y Conversaciones) en los que se pueden observar las interacciones que van a tener cada uno de los casos de uso descritos anteriormente en cada tabla (Tabla 1 - 20), con otros casos de uso o con los actores que representan a las personas/entidades que hay en el sistema. En el paquete “Conversaciones” (ver Figura 43) se puede encontrar los casos de uso involucrados a la hora de establecer una comunicación entre un usuario de la aplicación (profesor o alumno) y la IA. A continuación, se explican brevemente:

- **Iniciar conversación:** el usuario iniciar una conversación con la IA
- **Guardar conversación:** cuando el usuario inicia una conversación, el sistema automáticamente la almacena para poder acceder a ella en un futuro.
- **Eliminar conversación:** el usuario puede eliminar una conversación en el caso de que lo desee.
- **Ver conversaciones:** el usuario puede ver un listado de todas las conversaciones que ha tenido con la IA.
- **Enviar chat para realimentación:** el usuario (alumno) puede enviar cualquiera de sus chats para que el profesor le proporcione comentarios de realimentación.
- **Revisar chat y dar realimentación:** el profesor puede enviar comentarios de realimentación en el caso de que un alumno le haya enviado una de sus conversaciones.
- **Ver realimentación:** una vez el profesor haya mandado la realimentación, el alumno podrá ver dichos comentarios en el mismo chat.
- **Generar respuesta:** el modelo genera un tipo respuesta dependiendo del tipo de petición que le haya hecho el usuario.

En el paquete “Gestión de usuarios” (ver Figura 42) se encuentran los casos de uso relativos a los usuarios. Tanto las posibles acciones sobre su cuenta como la asignación de roles que hay dentro de la aplicación, como el sistema de recuperación de contraseña entre otros. Se explican con brevedad a continuación:

- **Registrarse:** un alumno o profesor se registra en la aplicación.
- **Iniciar sesión:** un usuario (registrado previamente en la aplicación) inicia sesión
- **Ver perfil:** un usuario puede ver los datos de su perfil y modificarlos.
- **Cerrar sesión:** un usuario con una sesión activa puede cerrar sesión en la aplicación.
- **Recuperar contraseña:** un usuario solicita la recuperación de su contraseña en el caso de que no se acuerde de ella.
- **Asignar rol de profesor:** el administrador asigna de manera manual el rol de “Profesor” a los usuarios que corresponda.
- **Dar de baja un usuario:** el administrador puede dar de baja la cuenta de cualquier usuario.
- **Enviar email:** el sistema enviará un email para que el usuario pueda modificar su contraseña.
- **Modificar datos:** el sistema modificará los datos del usuario en la BD.

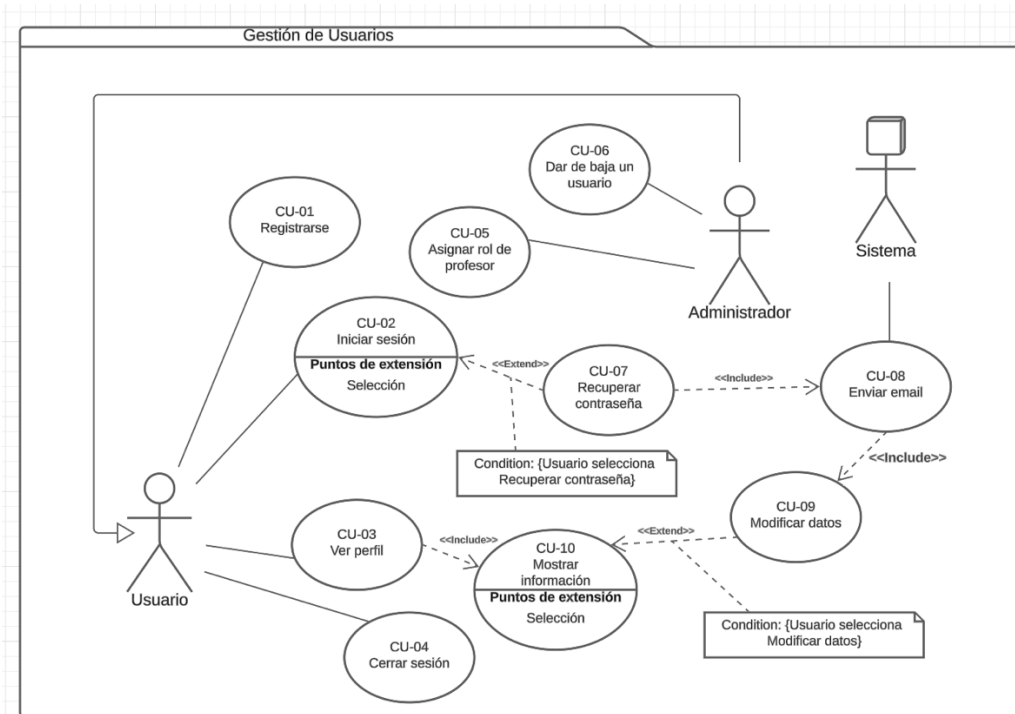


Figura 42. Diagrama de casos de uso del paquete “Gestión de usuarios”

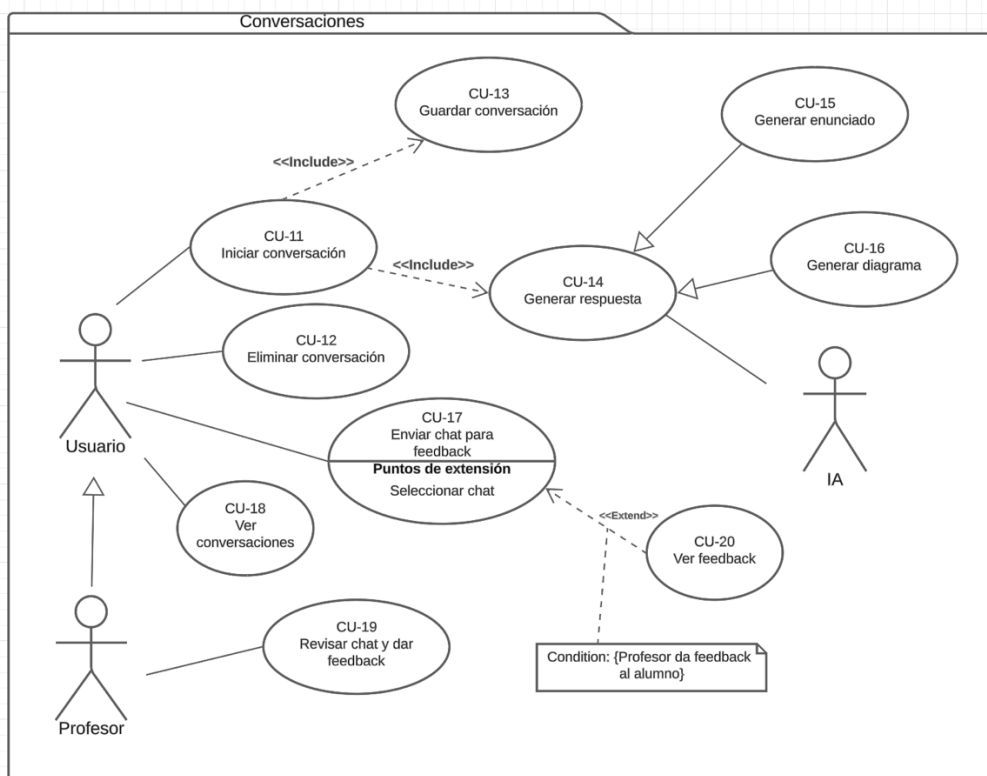


Figura 43. Diagrama de casos de uso del paquete “Conversaciones”

### Jerarquía de actores

Para poder identificar cuáles son los actores que intervienen a la hora de hacer cada caso de uso se ha realizado una representación de la jerarquía de actores del sistema (ver Figura 44). En ella se puede observar que los actores Alumno, Profesor y Administrador se podrán representar en un solo actor Usuario. Por otra parte, se ha incluido un actor Sistema, el cual sirve para representar determinadas acciones que se realizan en la aplicación como, por ejemplo, enviar un email a la hora de iniciar el proceso de recuperar contraseña.

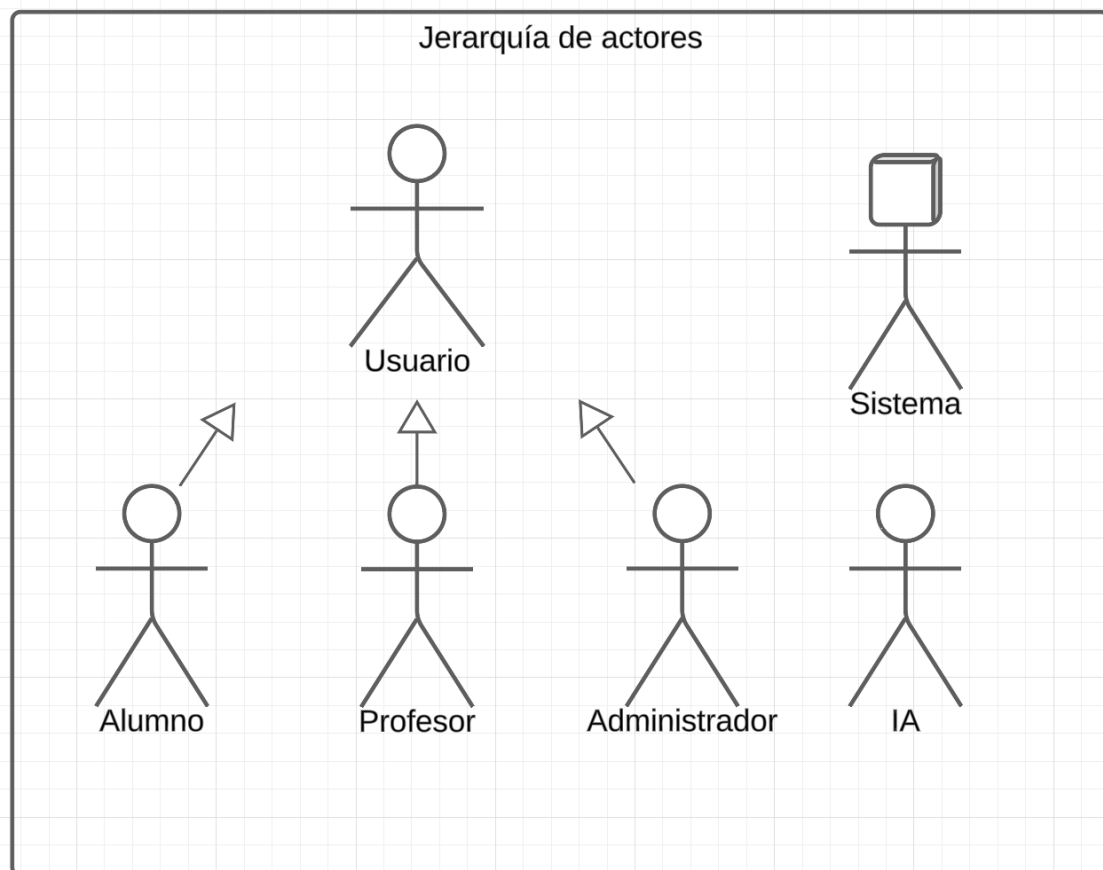


Figura 44. Jerarquía de actores del sistema

### Historias de usuario

Otra de los métodos utilizados para identificar los requisitos del sistema son las historias de usuario, las cuales son descripciones cortas y simples de las acciones que tendría que realizar una persona para lograr un determinado objetivo (ver Tablas 30 - 37). Las descripciones se relatan en primera persona poniéndose en la perspectiva del usuario y los criterios de aceptación en tercera persona desde el punto de vista del desarrollador.

Título	Registro en la aplicación
Descripción	Yo como usuario, quiero poder registrarme en la aplicación para poder acceder a todas sus funcionalidades.

<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El usuario debe de poder acceder al formulario de registro desde la página principal</li> <li>• El formulario de registro debe de solicitar al usuario un nombre de usuario, una contraseña y un correo electrónico</li> <li>• El correo electrónico no debe de existir previamente en el sistema</li> <li>• El correo electrónico y la contraseña deben de tener un formato válido</li> <li>• Se debe de informar al usuario si alguno de los datos es erróneo</li> <li>• Cuando envíe el formulario, el sistema enviará un correo electrónico de confirmación al usuario</li> <li>• Cuando el usuario confirme su correo electrónico se iniciará sesión en la aplicación y se asignará el rol de alumno por defecto</li> </ul>
--------------------------------	---

Tabla 30. Historia de usuario “Registro en la aplicación”

<b>Título</b>	<b>Inicio de sesión</b>
<b>Descripción</b>	Yo como usuario, quiero poder iniciar sesión en la aplicación.
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El usuario debe de poder acceder al formulario de inicio de sesión desde la página principal</li> <li>• El formulario de inicio de sesión debe de solicitar al usuario su nombre de usuario y su contraseña</li> <li>• Se debe de informar al usuario si alguno de los datos es erróneo</li> <li>• Cuando el usuario envíe el formulario se iniciará sesión en la aplicación</li> </ul>

Tabla 31. Historia de usuario “Inicio de sesión”

<b>Título</b>	<b>Cerrar sesión</b>
<b>Descripción</b>	Yo como usuario, quiero poder cerrar sesión en la aplicación.
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El usuario debe de elegir la opción de cerrar sesión entre las disponibles en el menú lateral de la aplicación</li> <li>• El sistema solicitará una confirmación por parte del usuario</li> <li>• El usuario confirmará su elección</li> </ul>

Tabla 32. Historia de usuario “Cerrar sesión”

<b>Título</b>	<b>Iniciar una conversación</b>
<b>Descripción</b>	Yo como usuario, quiero poder iniciar una nueva conversación con la I.A.
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El usuario debe haber iniciado sesión previamente en la aplicación</li> <li>• El sistema debe de mostrar la página principal</li> <li>• El usuario deberá de introducir un tema o un enunciado para iniciar una conversación</li> <li>• El sistema guardará la conversación en el registro de conversaciones del usuario</li> <li>• El asistente responderá la petición del usuario</li> </ul>

Tabla 33. Historia de usuario “Iniciar una conversación”

<b>Título</b>	<b>Ver las conversaciones</b>
<b>Descripción</b>	Yo como usuario, quiero poder ver las conversaciones que he tenido previamente con la IA.
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El usuario debe seleccionar la opción de “Mis chats” en el menú lateral</li> <li>• El sistema debe de mostrar la interfaz con las conversaciones que ha tenido el usuario</li> </ul>

Tabla 34. Historia de usuario “Ver las conversaciones”

<b>Título</b>	<b>Dar de alta a un profesor</b>
<b>Descripción</b>	Yo como administrador, quiero poder dar de alta a profesores que se han registrado previamente.
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El administrador debe de entrar en el apartado de usuarios desde el panel de administración</li> <li>• El administrador debe de poder buscar a los usuarios por el nombre</li> <li>• El usuario debe de poder cambiar el rol de “Alumno” por el de “Profesor”</li> </ul>

Tabla 35. Historia de usuario “Dar de alta a un profesor”

<b>Título</b>	<b>Dar <i>feedback</i> a la IA</b>
<b>Descripción</b>	Yo como profesor, quiero poder dar el <i>feedback</i> correspondiente sobre una conversación a un alumno que lo ha solicitado
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El profesor debe de poder seleccionar la opción de “Dar <i>feedback</i>” en el menú lateral</li> <li>• El sistema mostrará al profesor las conversaciones que estén pendientes de dar <i>feedback</i></li> <li>• El profesor debe seleccionar una de las conversaciones pendientes y poner los comentarios que crea oportunos</li> </ul>

Tabla 36. Historia de usuario “Dar *feedback* a la IA

<b>Título</b>	<b>Eliminar una cuenta</b>
<b>Descripción</b>	Yo como administrador, quiero poder eliminar la cuenta de un usuario del sistema
<b>Criterios de aceptación</b>	<ul style="list-style-type: none"> <li>• El administrador debe de entrar en el apartado de usuarios desde el panel de administración</li> <li>• El administrador debe de poder buscar a los usuarios por el nombre</li> <li>• El administrador debe de poder eliminar la cuenta de dicho usuario</li> </ul>

Tabla 37. Historia de usuario “Eliminar una cuenta”

## Anexo 2: Análisis y Diseño del sistema

### Modelo de dominio

Después de identificar y documentar todos los tipos de requisitos se ha realizado el diagrama de clases (ver Figura 45) para poder tener una representación visual y global de los objetos y relaciones del sistema.

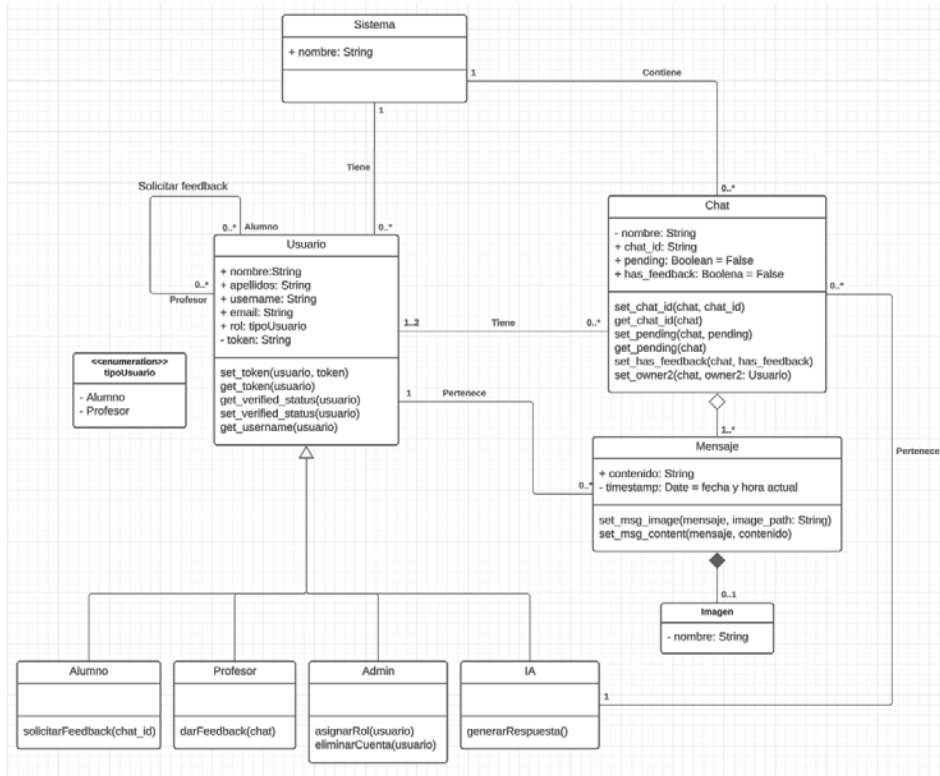


Figura 45. Diagrama de clases del sistema

### Vista de arquitectura

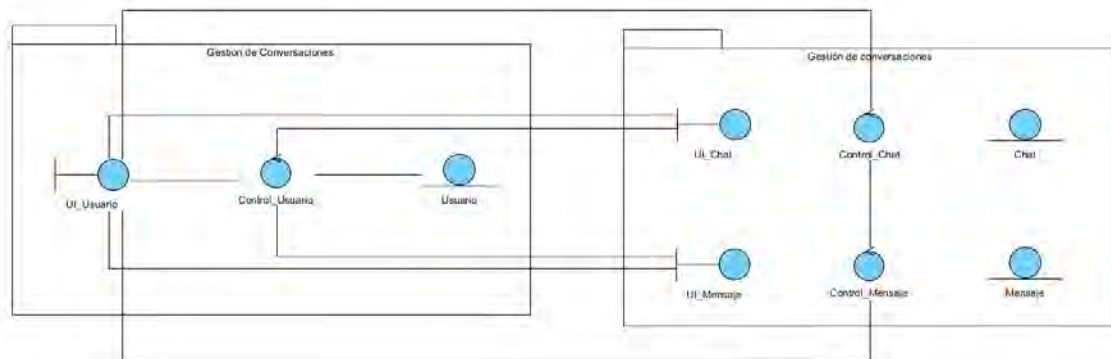


Figura 46. Vista de arquitectura del modelo de análisis

Realización casos de uso-análisis

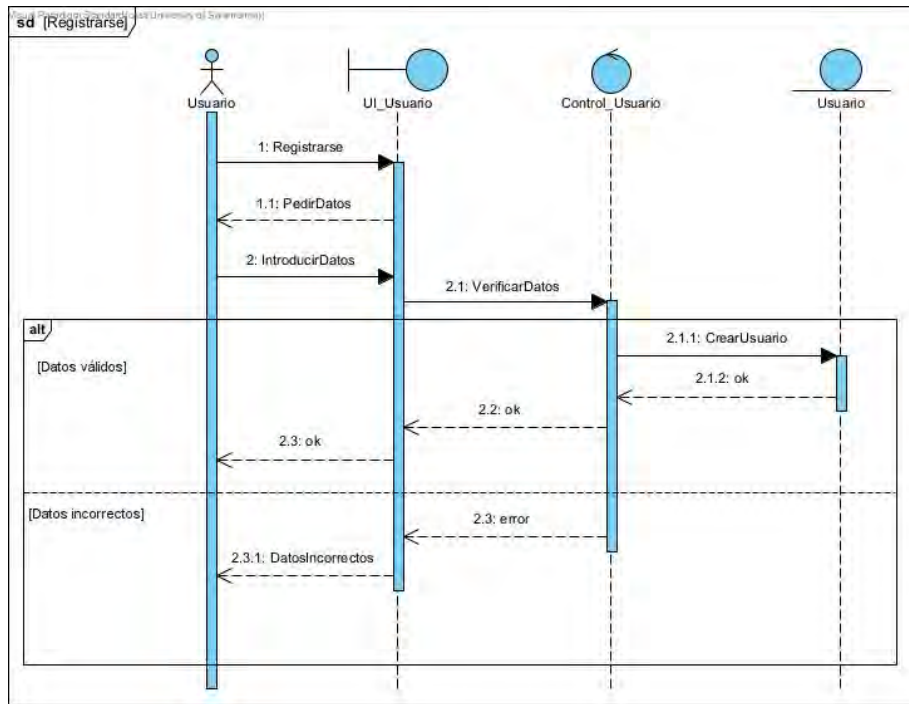


Figura 47. Caso de uso-análisis “Registrarse”

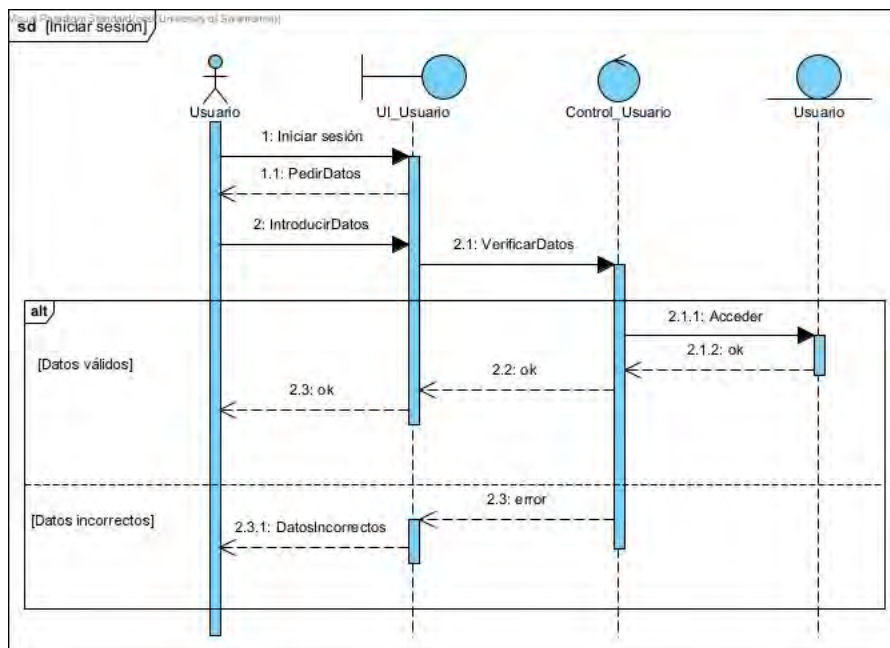


Figura 48. Caso de uso-análisis “Iniciar sesión”



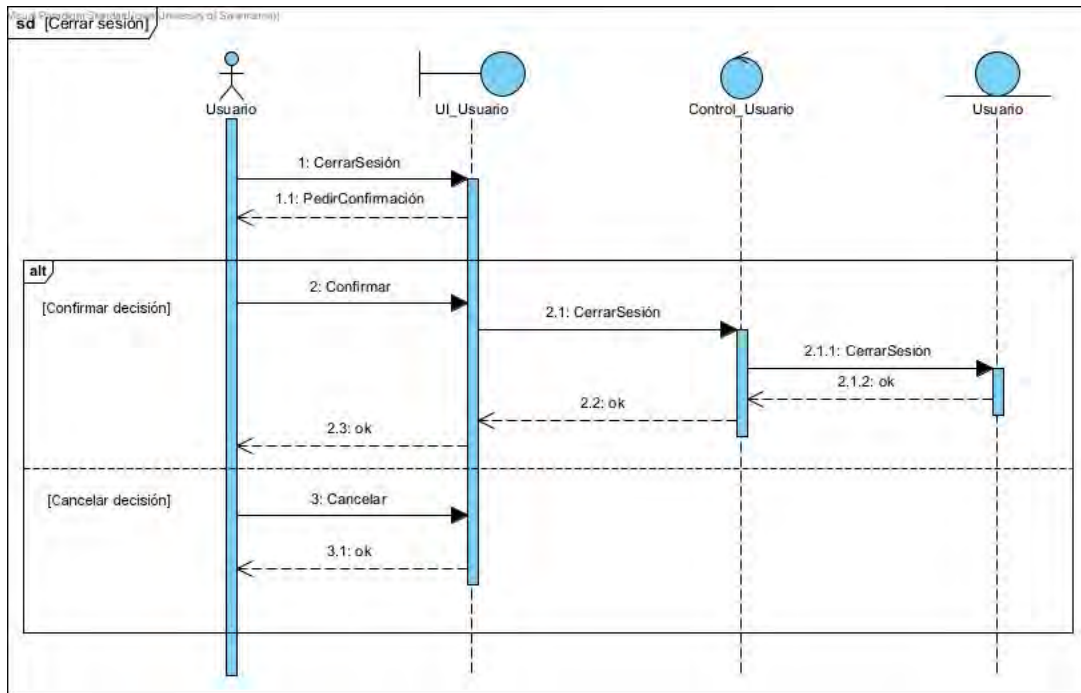


Figura 49. Caso de uso-análisis “Cerrar sesión”

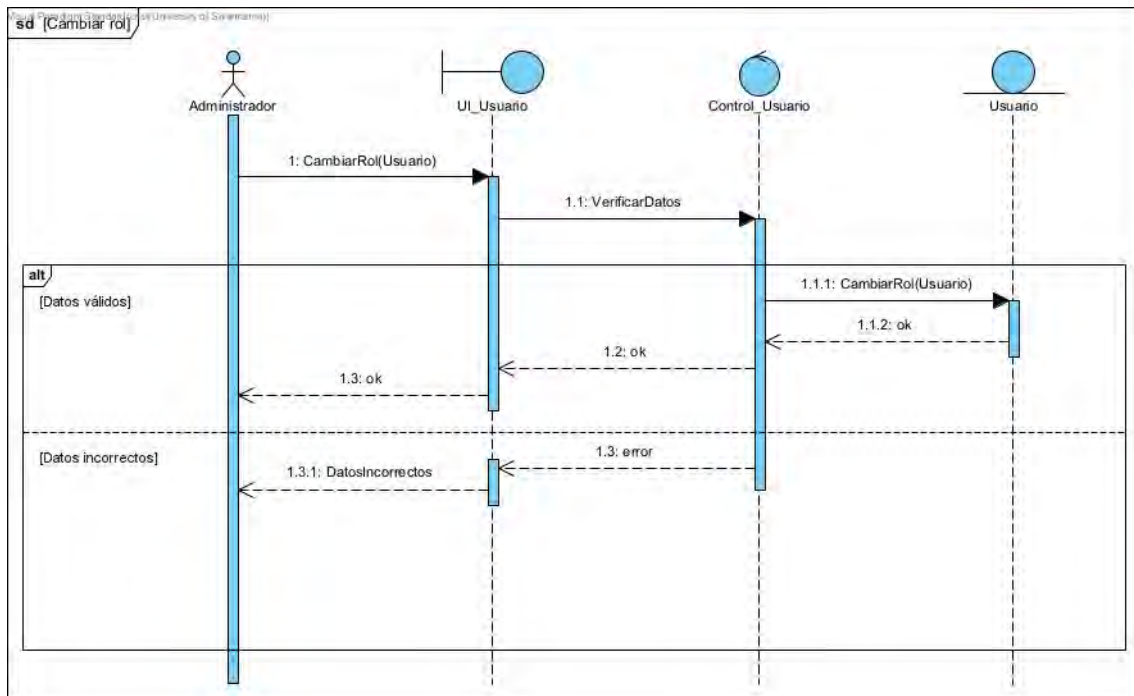


Figura 50. Caso de uso-análisis “Cambiar rol”

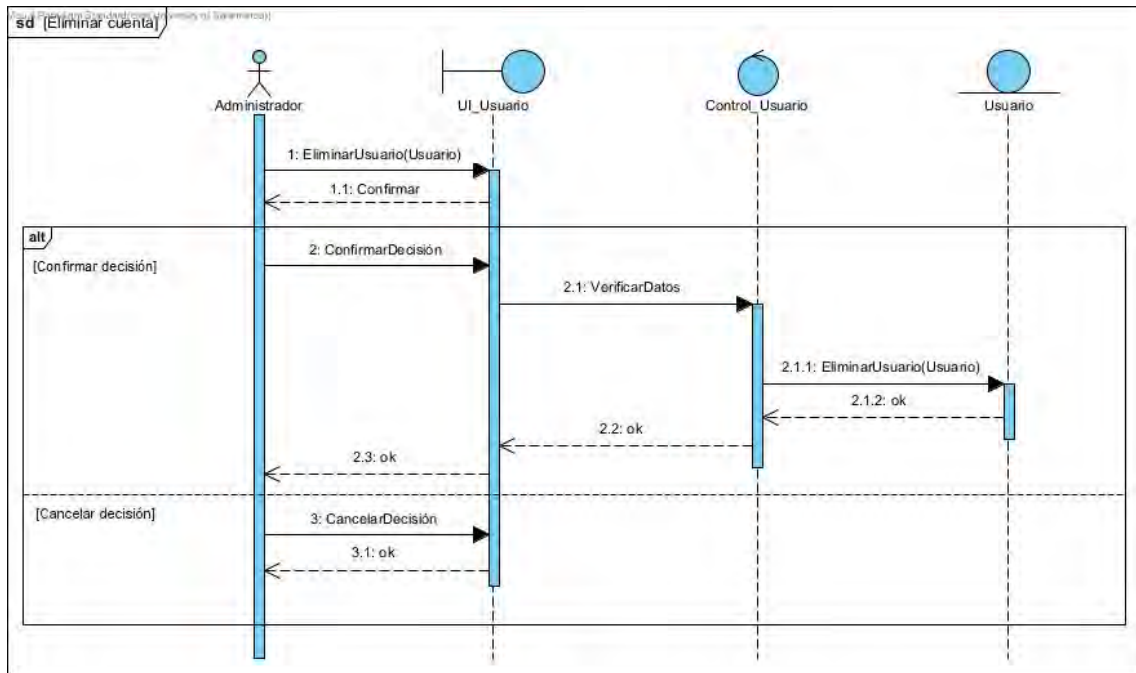


Figura 51. Caso de uso-análisis “Eliminar cuenta”

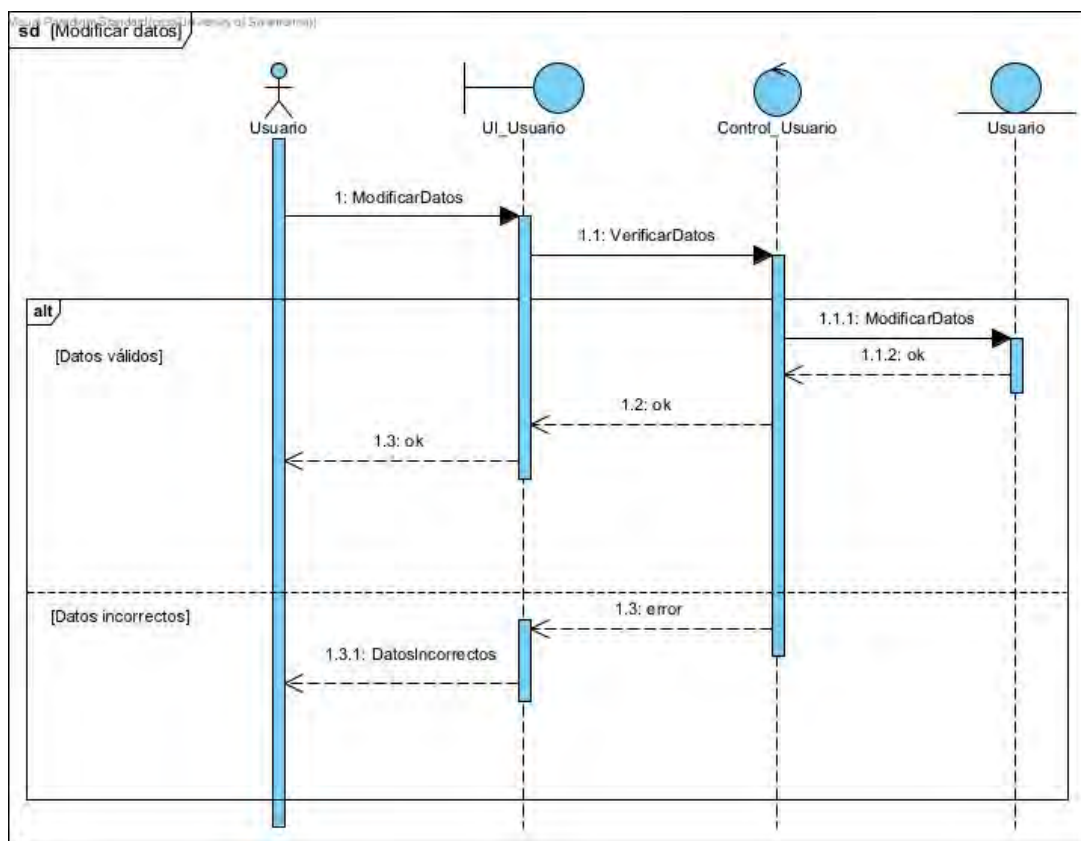


Figura 52. Caso de uso-análisis “Modificar datos”

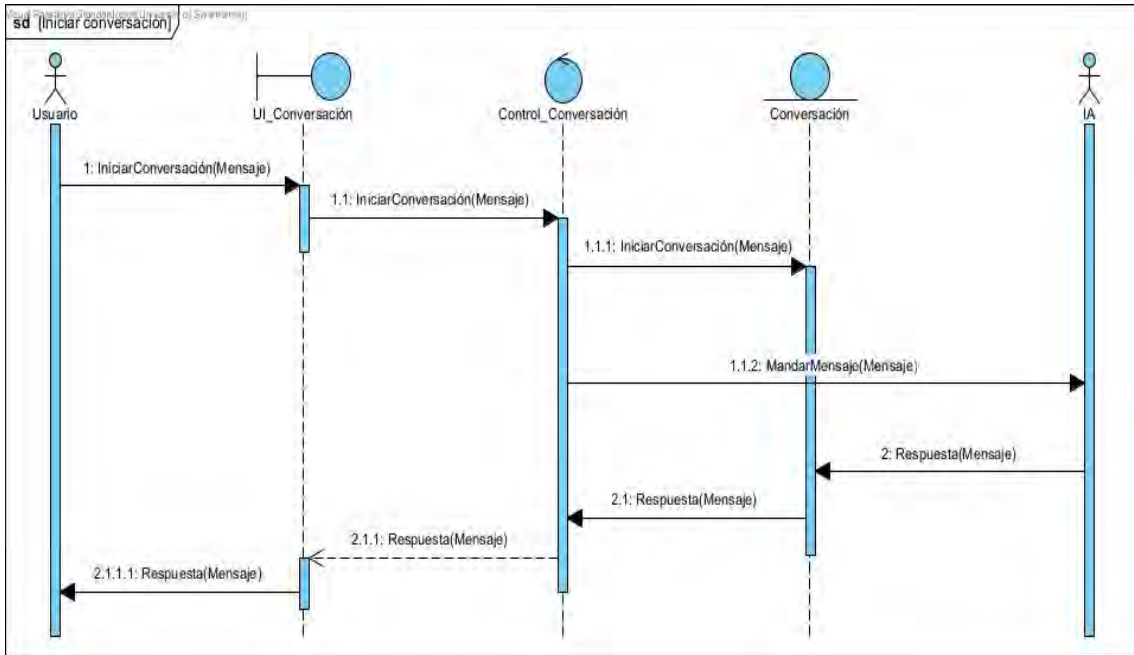


Figura 53. Caso de uso-análisis “Iniciar conversación”

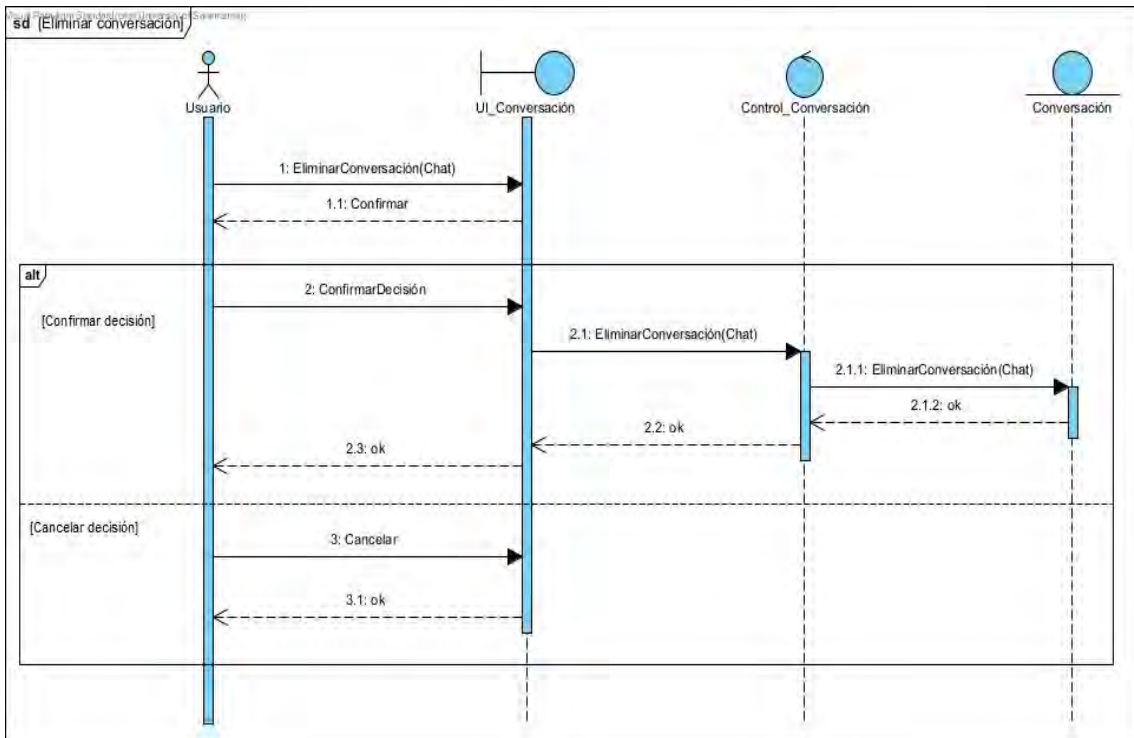


Figura 54. Caso de uso-análisis “Eliminar conversación”

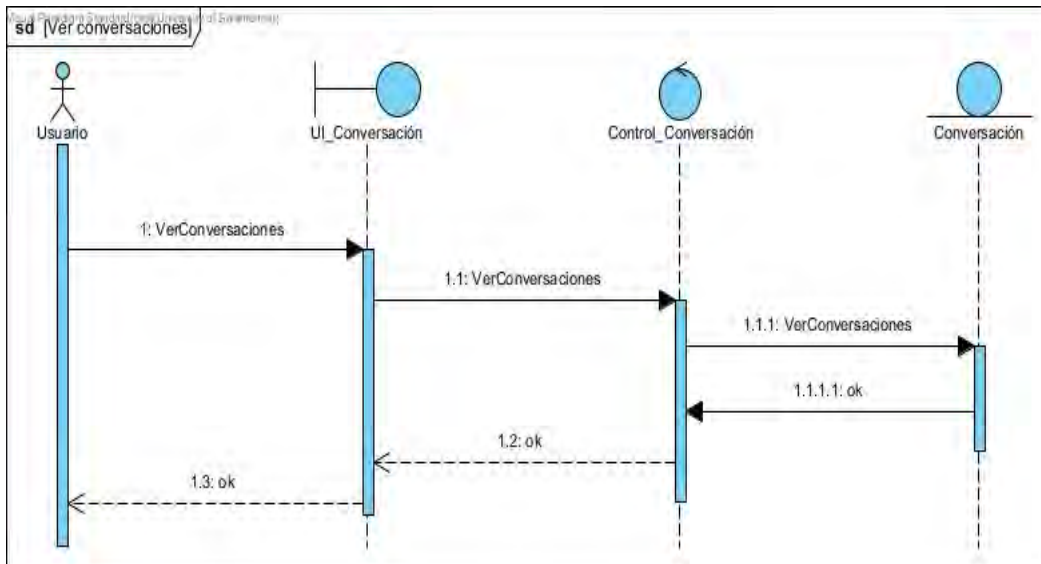


Figura 55. Caso de uso-análisis “Ver conversaciones”

### Arquitectura de capas

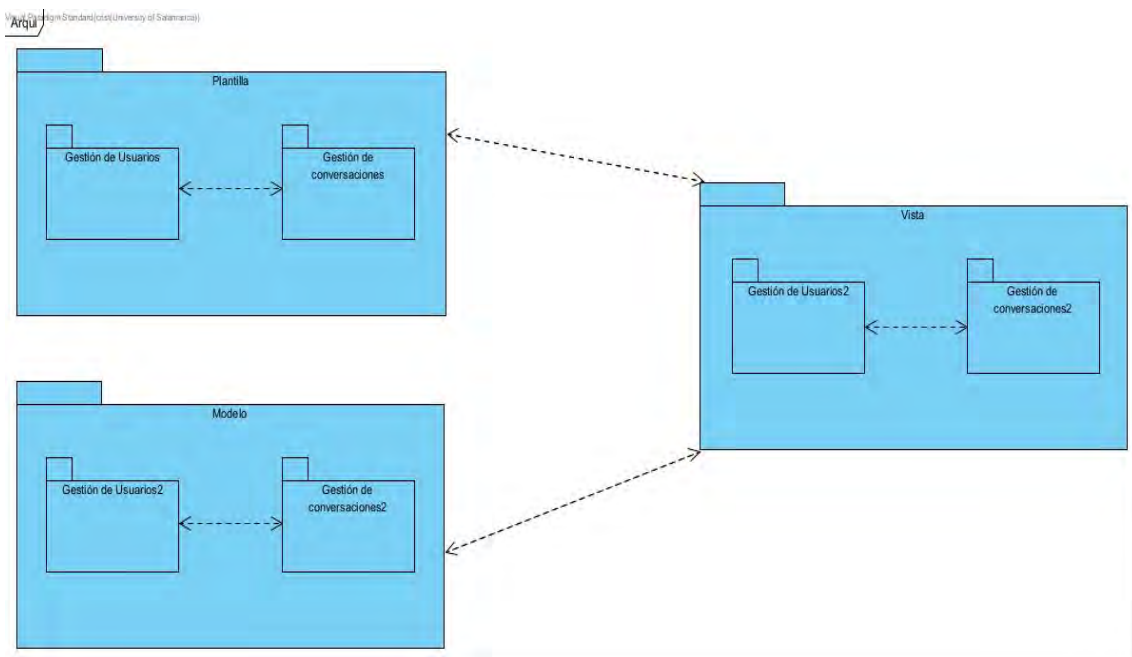


Figura 56. Arquitectura de capas del modelo de diseño

## Diagrama de clases real

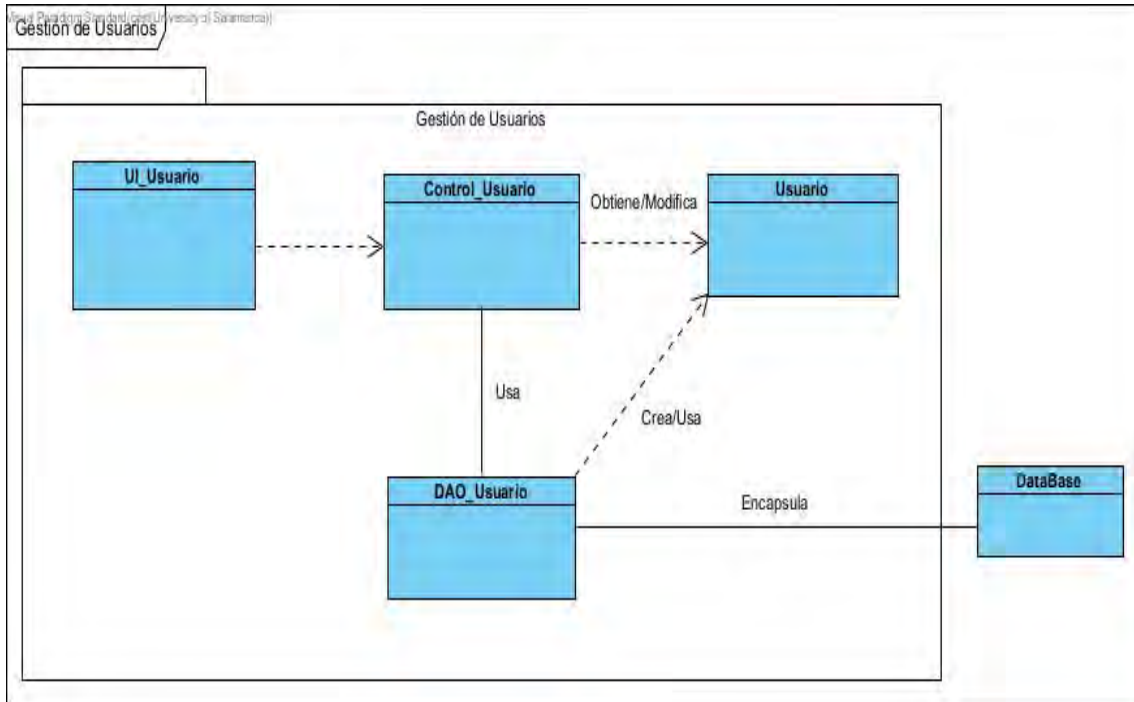


Figura 57. Diagrama de clases real "Gestión de usuarios"

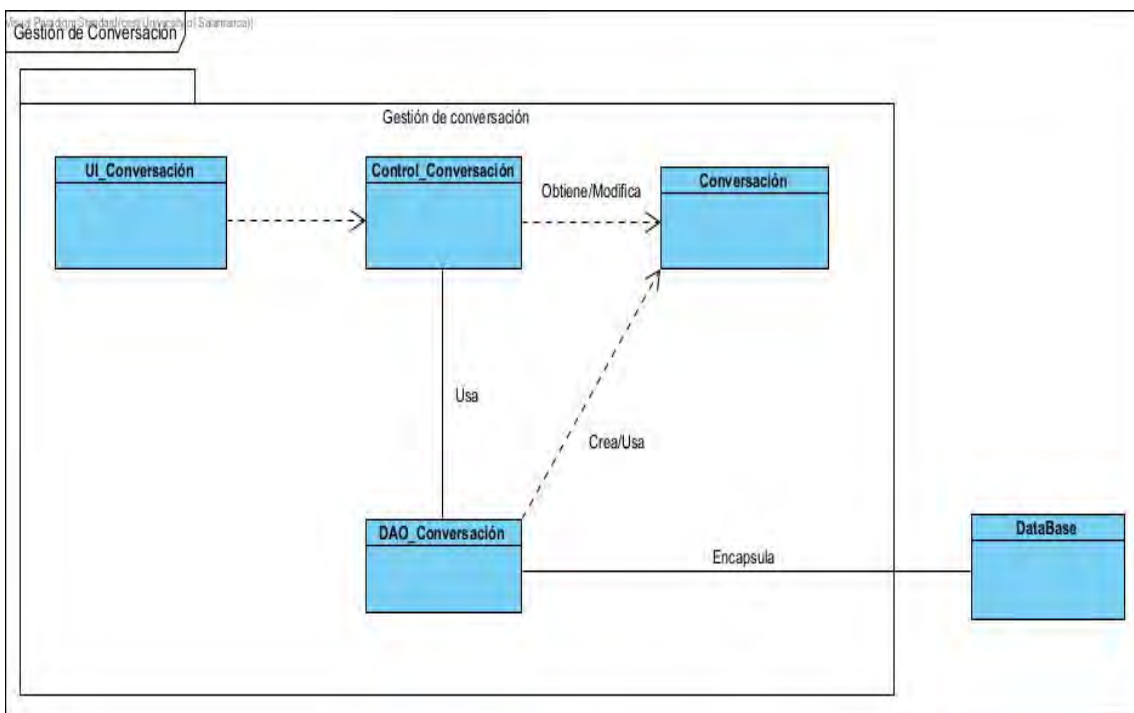


Figura 58. Diagrama de clases real "Gestión de conversaciones"

Casos de uso-diseño

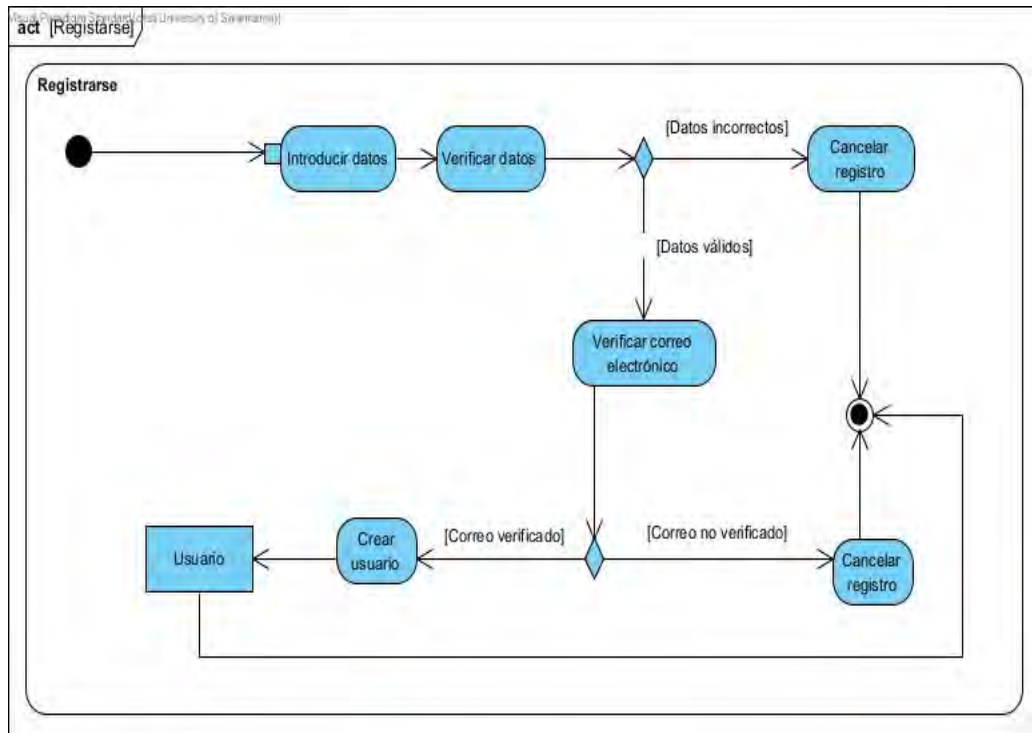


Figura 59. Caso de uso-diseño "Registrarse"

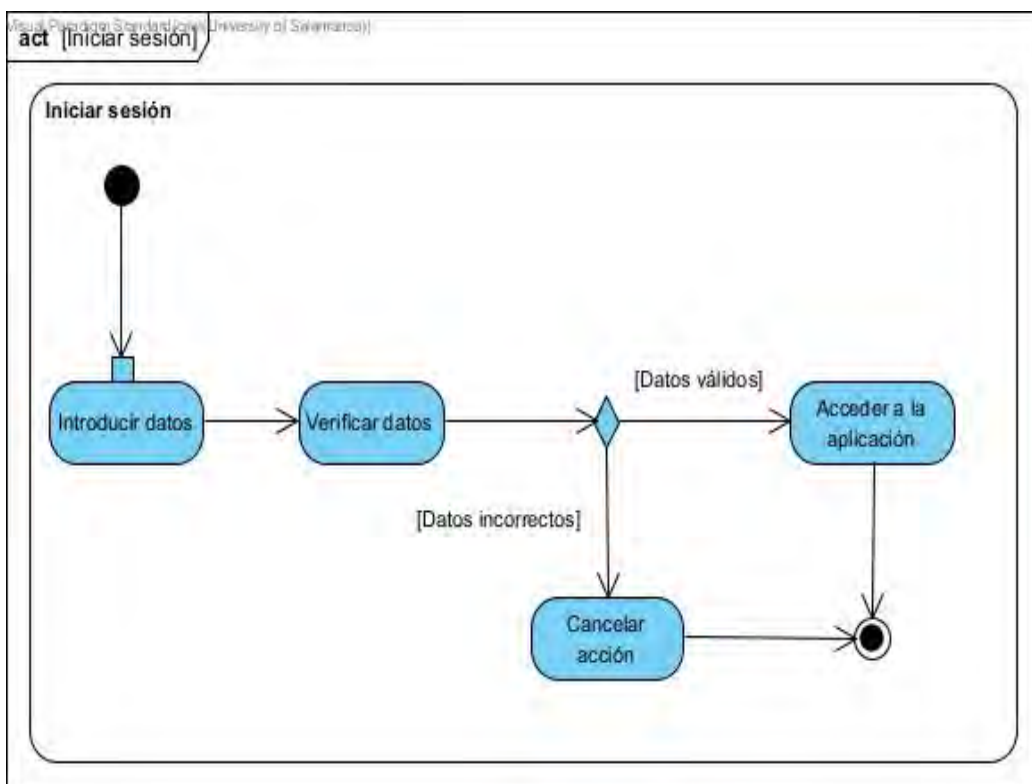


Figura 60. Caso de uso-diseño "Iniciar sesión"

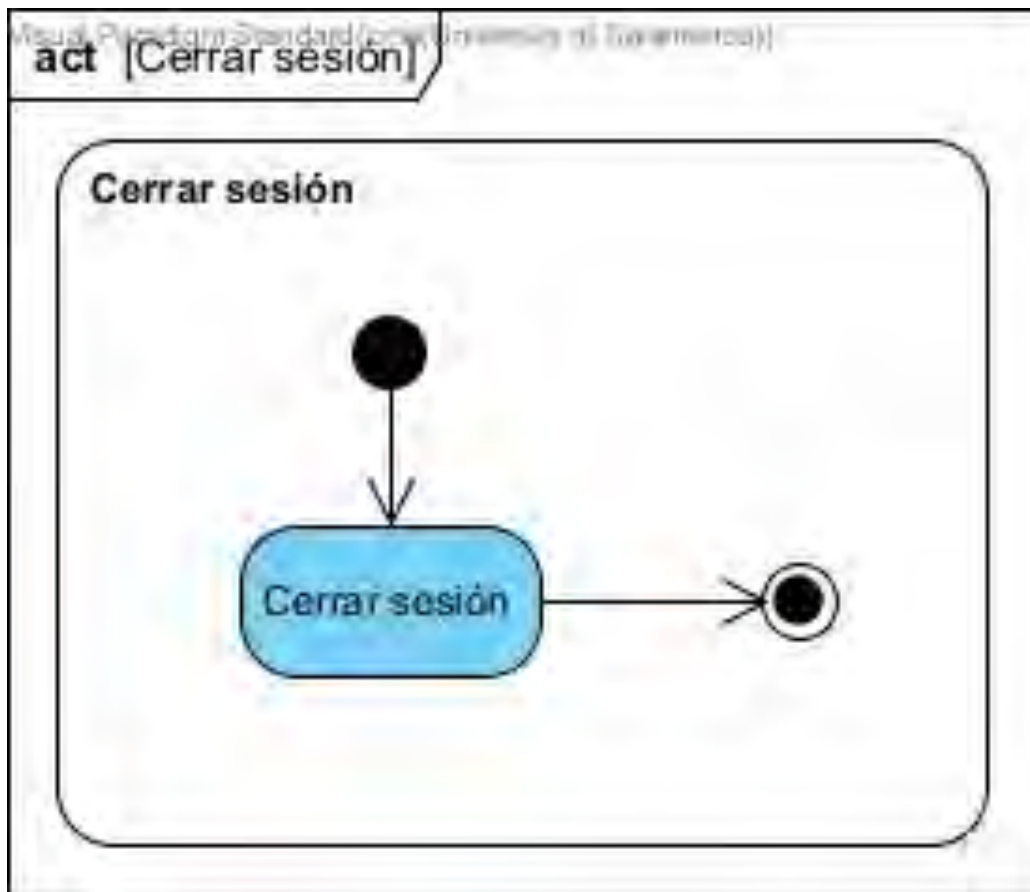


Figura 61. Caso de uso-diseño "Cerrar sesión"

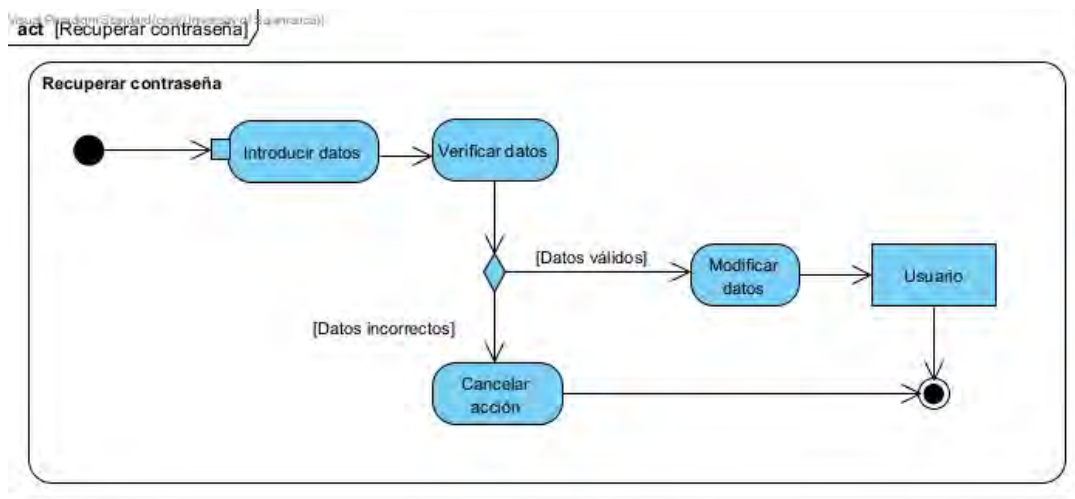


Figura 62. Caso de uso-diseño "Recuperar contraseña"

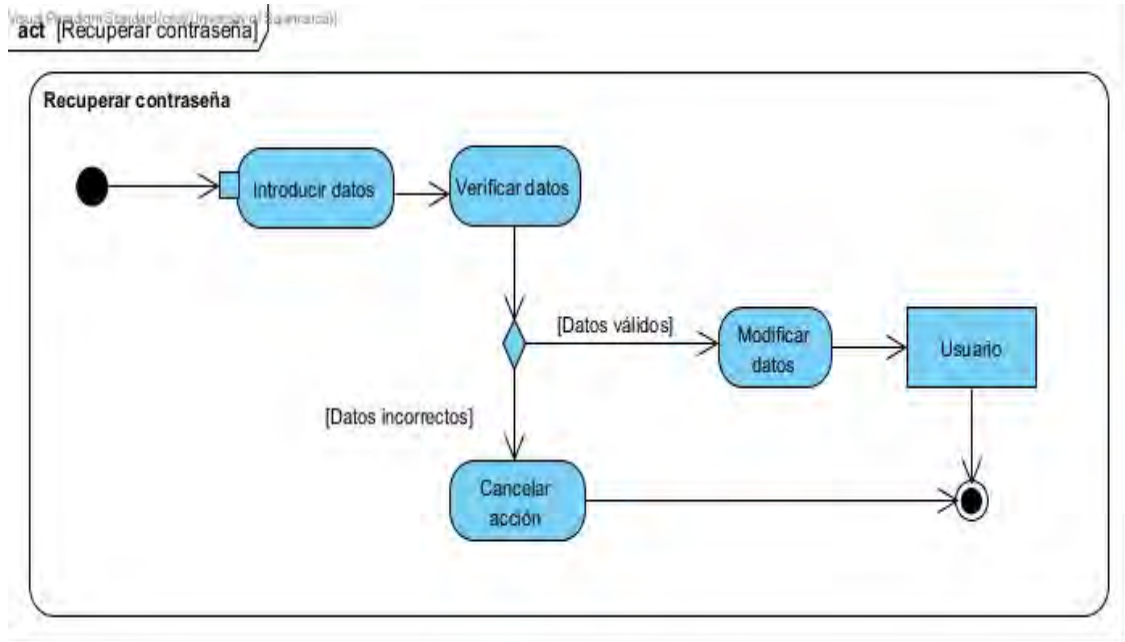


Figura 63. Caso de uso-diseño “Recuperar contraseña”

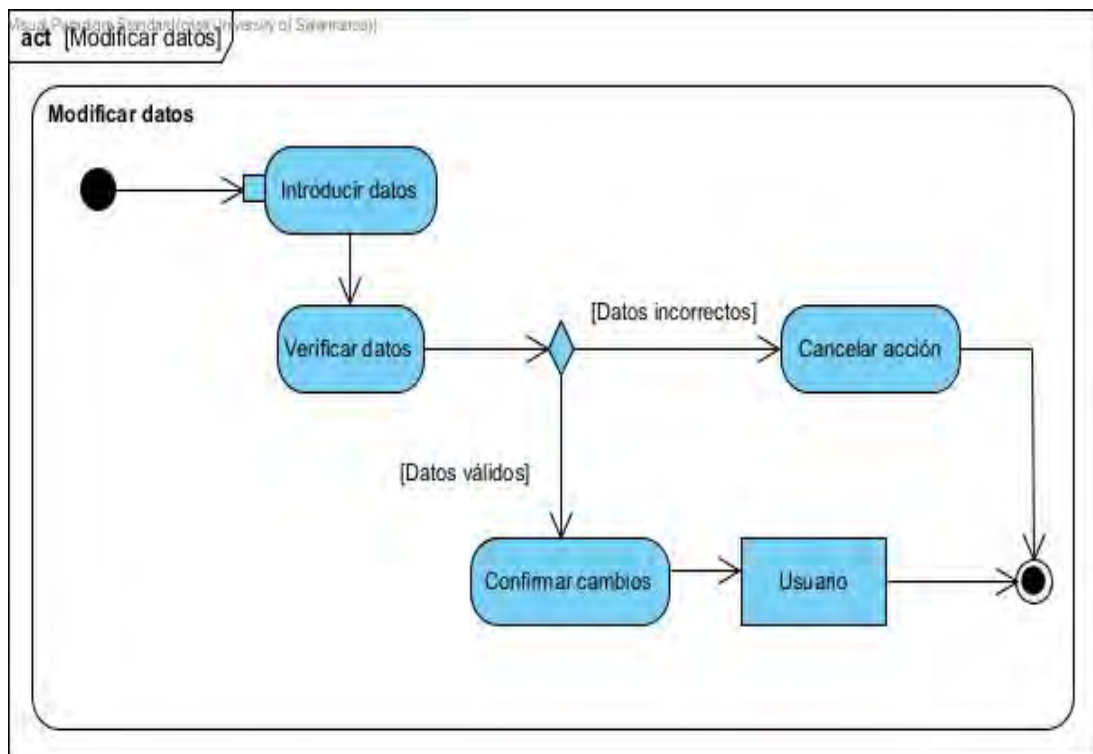


Figura 64. Caso de uso-diseño “Modificar datos”



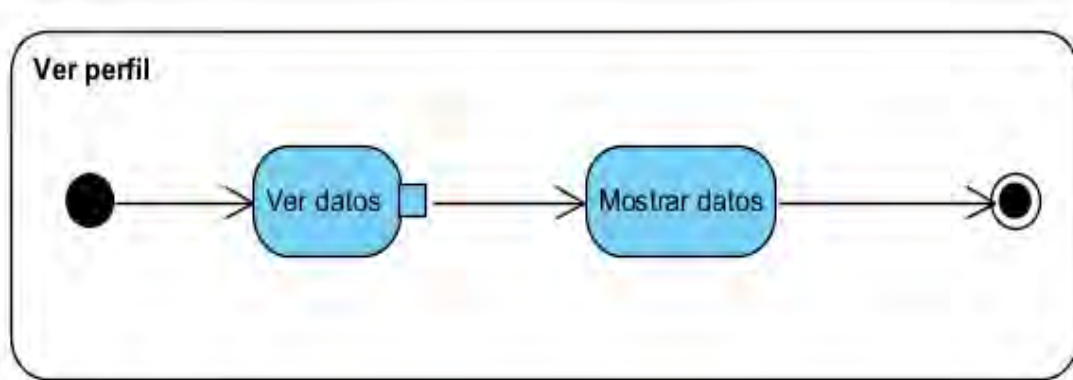


Figura 65. Caso de uso-diseño "Ver perfil"

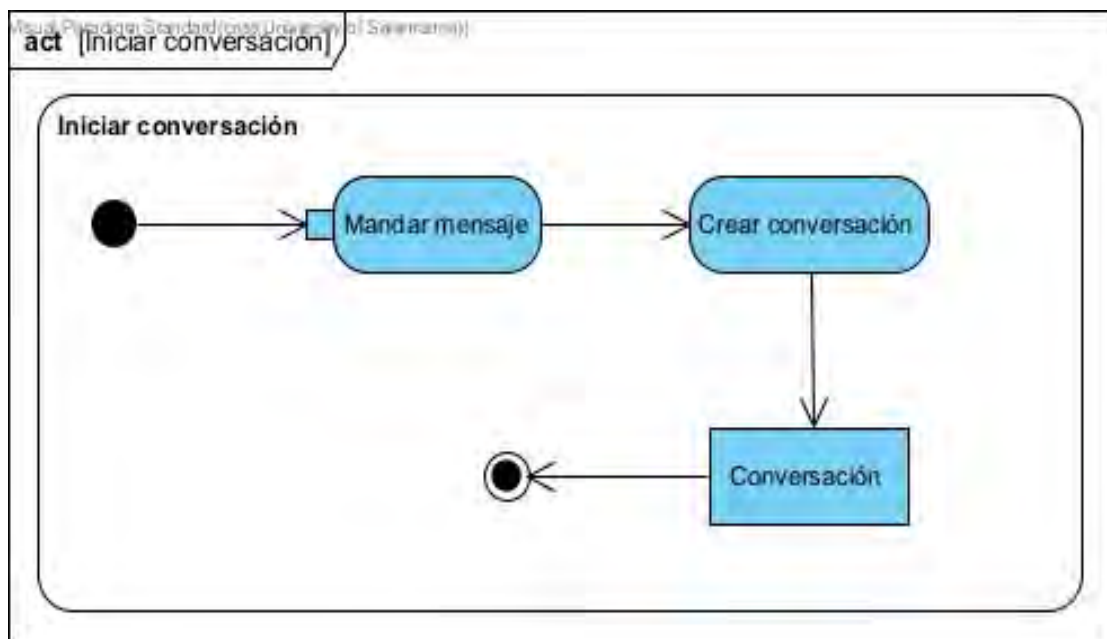


Figura 66. Caso de uso-diseño "Iniciar conversación"

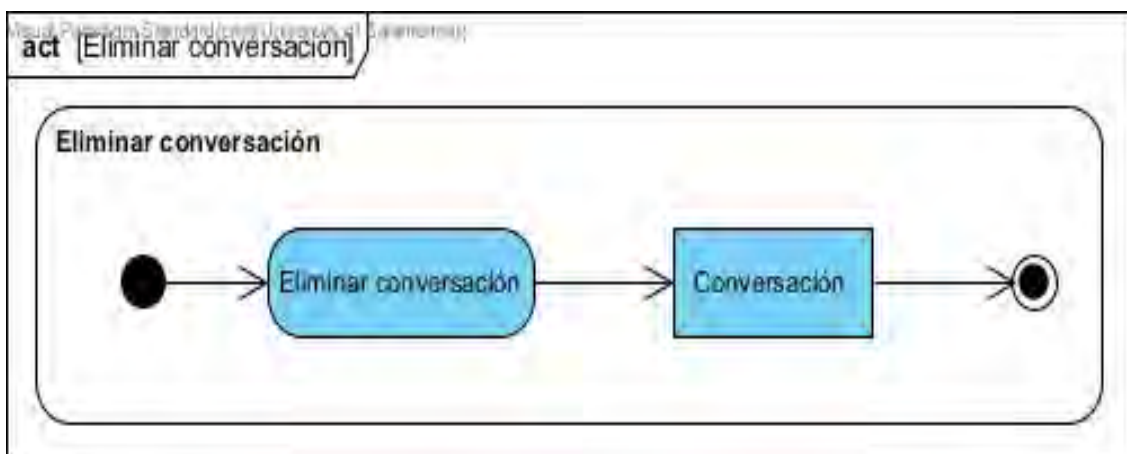


Figura 67. Caso de uso-diseño "Eliminar conversación"

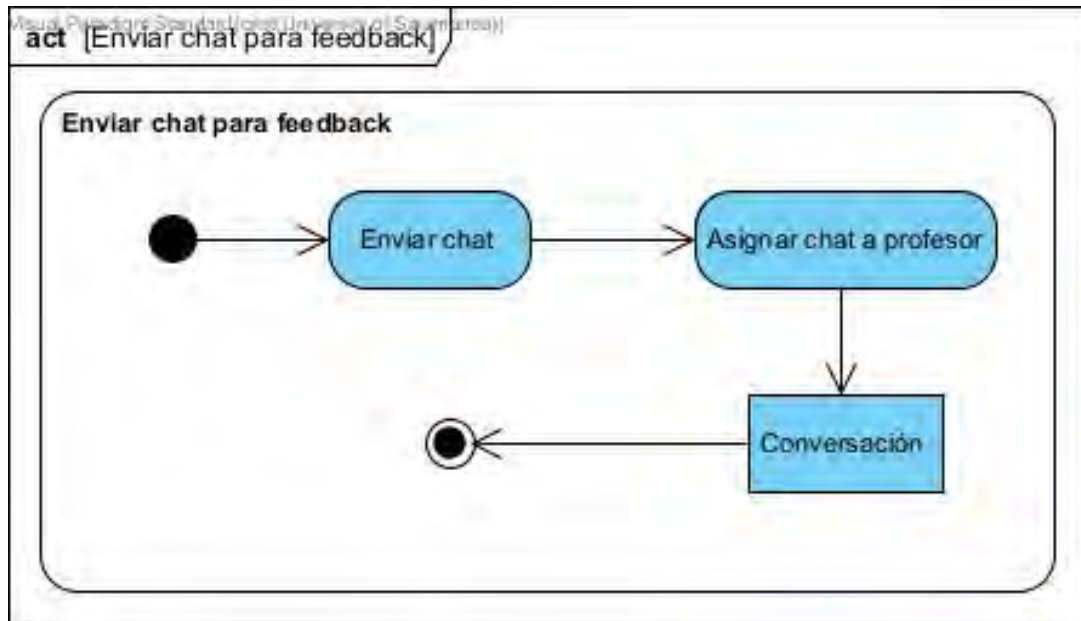


Figura 68. Caso de uso-diseño “Enviar chat para feedback”

### Modelo de despliegue

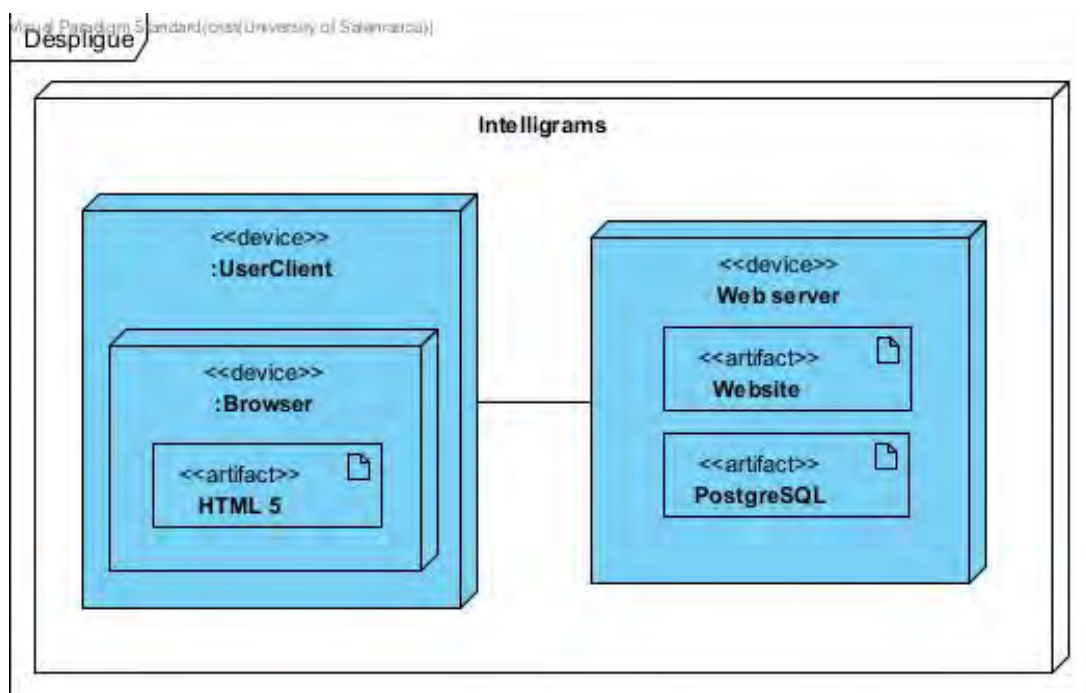


Figura 69. Diagrama de despliegue de la aplicación

### Esquema de la base de datos

La estructura de la base de datos del proyecto se puede dividir en tres tablas principales (Tabla 38, Tabla 39 y Tabla 40) y otras adicionales que crea Django para la gestión de los permisos de los usuarios, las sesiones y lo relativo al administrador de la aplicación. A continuación, se va a describir cada tabla.

AuthApp_usuario	
<b>Descripción general</b>	Esta tabla se corresponde con la clase Usuario que se ha definido en el archivo “models.py” del proyecto Django. En ella se incluyen atributos necesarios para la gestión de los usuarios (inicio de sesión, registro, verificación del correo electrónico) y otros que se han usado a la hora de gestionar el acceso de un determinado tipo de usuario a una característica de la aplicación como, por ejemplo, el apartado de <i>feedback</i> .
<b>Atributo</b>	<b>Descripción</b>
<b>id</b>	Atributo que asigna de forma automática Django para identificar a cada elemento de forma única en la BD.
<b>password</b>	Se corresponde con la contraseña del usuario en el sistema la cual estará encriptada, por lo que nadie que no sea el usuario podrá acceder a su cuenta.
<b>last_login</b>	Atributo de tipo “timestamp with time zone” que indica la hora del último inicio de sesión por parte del usuario en el sistema.
<b>username</b>	Nombre de usuario que tendrá el usuario y que se elige a la hora de registrarse en el sistema.
<b>email</b>	Atributo necesario para iniciar sesión en la aplicación junto con la contraseña. Cada cuenta tendrá una dirección de correo electrónico única.
<b>nombre</b>	Nombre real del usuario propietario de la cuenta.
<b>apellidos</b>	Apellidos del usuario propietario de la cuenta.
<b>imagen</b>	Imagen de perfil que tendrá la cuenta del usuario.
<b>is_admin</b>	Atributo booleano que indica si un usuario es administrador.
<b>is_verified</b>	Atributo que indica si un usuario ha verificado su dirección de correo electrónico.
<b>rol</b>	Tipo de usuario dentro del sistema. Puede ser Alumno o Profesor.
<b>token</b>	Atributo que almacena el código aleatorio que se genera para verificar la dirección de correo electrónico del usuario.

Tabla 38. Tabla del usuario en la BD

MainApp_message	
<b>Descripción general</b>	Tabla que se corresponde con la clase Message que se ha definido en el archivo “models.py” del proyecto Django. En ella se pueden encontrar los atributos propios de cada mensaje que se mande por la aplicación por parte de los usuarios.
<b>Atributo</b>	<b>Descripción</b>
<b>id</b>	Atributo que asigna de forma automática Django para identificar a cada elemento de forma única en la BD.
<b>content</b>	En este atributo se almacena el propio contenido del mensaje en formato de texto.
<b>timestamp</b>	Atributo de tipo “timestamp with time zone” que indica la hora en la que se ha enviado el mensaje por la aplicación. Este atributo es de gran ayuda a la hora de ordenar los mensajes en la vista del usuario.
<b>chat_id</b>	Este atributo se corresponde con el identificador del chat (id) al que está asociado el mensaje.
<b>sender_id</b>	Atributo que indica el usuario que ha mandado el mensaje y que se usa para aplicar un estilo diferente a cada mensaje en la vista del usuario.
<b>image</b>	Atributo en el que se almacena la imagen que el usuario puede adjuntar con cada mensaje para que la analice el modelo de IA.

Tabla 39. Tabla del mensaje en la BD

MainApp_chat	
<b>Descripción general</b>	Tabla que se corresponde con la clase Chat que se ha definido en el archivo “models.py” del proyecto Django. En ella se pueden encontrar los atributos propios de cada chat que se cree en la aplicación cuando se hace una pregunta al modelo de IA por primera vez desde el apartado de “Asistente”.
<b>Atributo</b>	<b>Descripción</b>
<b>id</b>	Atributo que asigna de forma automática Django para identificar a cada elemento de forma única en la BD.

<b>owner_id</b>	Este atributo se corresponde con el usuario que ha creado el chat en el sistema. Gracias al parámetro "on_delete=models.CASCADE" que se ha incluido en el código, cuando se elimine un usuario del sistema también se eliminarán todos los chats asociados a dicho usuario. También se usa a la hora de determinar la lógica del <i>feedback</i> en el sistema y para determinar quién ha mandado cada mensaje en la vista del usuario.
<b>name</b>	En este atributo se almacena el nombre que se le asigna al chat cuando se crea. Para automatizar esta tarea se hace uso de la IA para generar un título en base a la pregunta o solicitud del usuario.
<b>chat_id</b>	Cadena de caracteres que se genera aleatoriamente mediante una función en el archivo "views.py" y que se asigna de manera automática para identificar y acceder a cada chat en una página distinta.
<b>pending</b>	Este atributo booleano indica si el chat está en estado pendiente de recibir <i>feedback</i> por parte del profesor.
<b>owner2_id</b>	Atributo que identifica el segundo usuario al que pertenece el chat cuando un alumno manda el chat a revisión, dicho usuario se elige de manera aleatoria entre todos los que tengan el rol de "Profesor".
<b>has_feedback</b>	Parámetro utilizado a la hora de comprobar si un chat ha recibido <i>feedback</i> por parte del profesor para indicarlo en la vista del usuario y volver a permitir escribir en el chat.
<b>created</b>	Es un tributo de tipo "timestamp with time zone" el cual se asigna de forma automática cuando se crea el chat, para de este modo mostrar la lista de los chats ordenada, estando los chats más antiguos en la parte inferior y los recién creados en la superior.

Tabla 40. Tabla del chat en la BD

Para poder comunicar las tablas descritas anteriormente, se han tenido que relacionar a través de atributos que los identifiquen de manera única. Dos de las relaciones que se puede observar en el esquema de la base de datos (ver Figura 70) son las que unen el atributo "id" del usuario con los atributos "owner\_id" y "owner2\_id" del chat, de esta forma a cada usuario del sistema le pueden pertenecer 0 o más chats y cada chat puede tener 1 o 2 propietarios dependiendo si se ha enviado el chat para revisión o no. La relación con owner2\_id no tiene ningún tipo de restricción, por lo que se podrá eliminar dicho usuario sin que se presente ningún problema en la BD.

Otra de las relaciones que hay en el esquema es la que une el atributo "chat\_id" del mensaje con el atributo "id" del chat, de esta manera a cada chat le pueden pertenecer 1 o más mensajes (en el caso de eliminar el chat también se

eliminarán los mensajes asociados a este). Por último, hay una relación que une el atributo “sender\_id” del mensaje con el atributo “id” del usuario, para de esta manera, asociar los mensajes al usuario que los mande.

El resto de las tablas y relaciones las genera de forma automática Django a la hora de crear el proyecto para las funcionalidades que tiene el administrador del sistema, gestión de los permisos y sesiones activas en la aplicación.

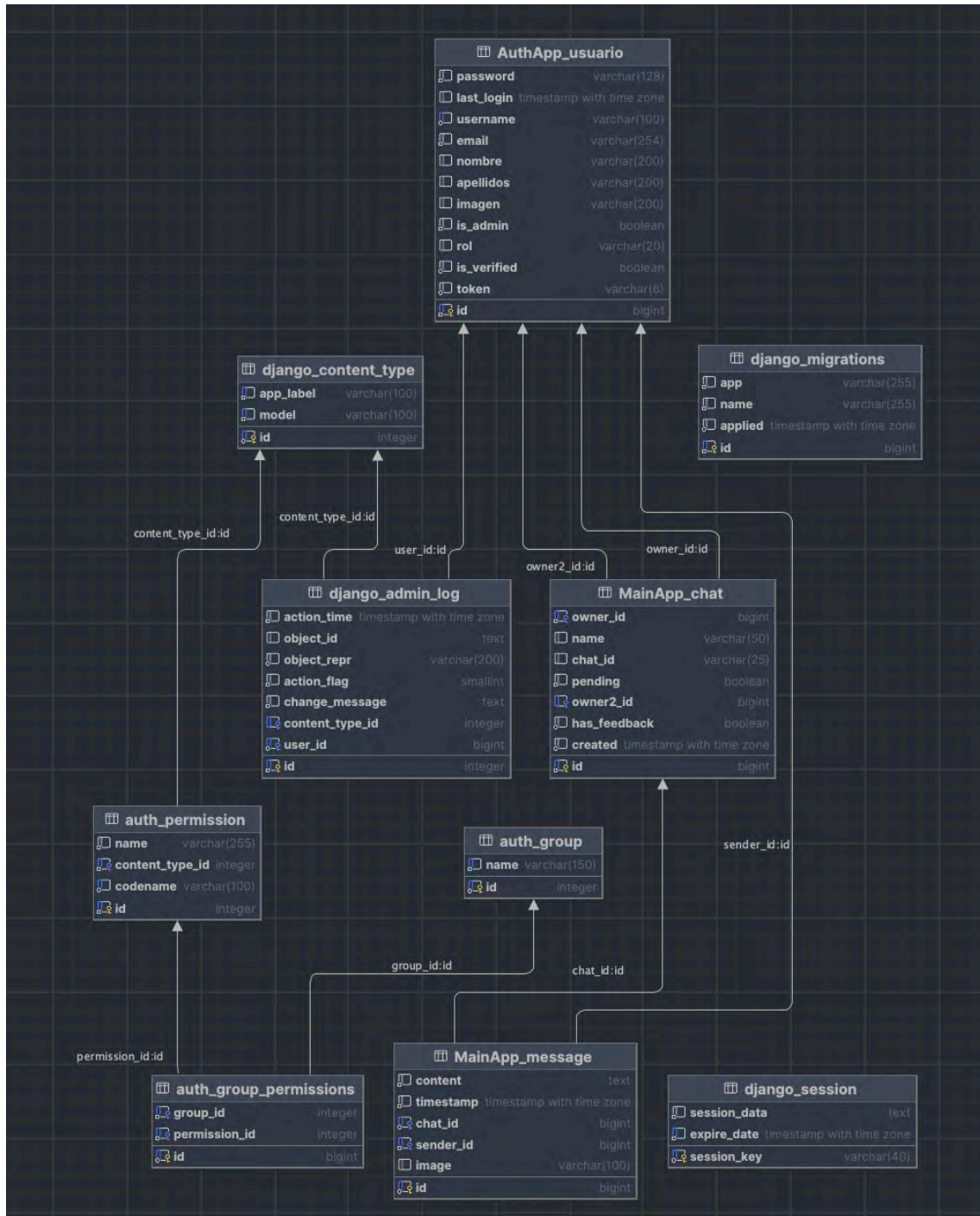


Figura 70. Esquema de la base de datos de la aplicación

## Anexo 3: Estimación del tamaño y esfuerzo

En este apartado, se detalla la estimación del tamaño y el esfuerzo requerido para el desarrollo y la implementación del proyecto. La precisión en la estimación es crucial, ya que proporciona la base para un presupuesto y asegura una gestión de recursos eficaz a lo largo del ciclo de vida del proyecto. El objetivo principal es aplicar métricas específicas que permitan una valoración objetiva del trabajo y los recursos necesarios.

Esta estimación no solo facilita la planificación financiera y de tiempo, sino que también sirve como un indicador crítico para la asignación de tareas y la gestión del progreso del proyecto. Las dos métricas que se han empleado para hacer la estimación han sido los factores de complejidad técnica y los factores de entorno (Moreno-García, 2024).

### Factores de complejidad técnica

- **Sistema distribuido:** este factor evalúa si el software debe operar en un entorno distribuido, aumentando la complejidad por la necesidad de sincronización y comunicación entre componentes. Debido a que la aplicación se comporta principalmente de manera síncrona se le ha asignado un valor de 1.
- **Rendimiento:** en este factor considera si el sistema requiere altos niveles de rendimiento, como tiempos de respuesta rápidos o alto procesamiento de transacciones. Debido a que el factor de rendimiento recae sobre los tiempos en los que se genera la respuesta y es un factor importante para mantener una conversación fluida pero que a su vez no depende de desarrollador, se le ha asignado un valor de 3.
- **Eficiencia de usuario final:** mide cómo el software mejora la eficiencia del usuario final, simplificando tareas o reduciendo tiempo de operaciones. Debido a que se han realizado un número de pruebas muy reducido a medida que se desarrollaba la aplicación y todavía no se tiene una estimación clara de cómo va a afectar la aplicación a la hora de las tareas didácticas, se le ha asignado un valor de 2.
- **Procesamiento interno complejo:** evalúa la complejidad de los procesos que el software necesita manejar, incluyendo la gestión de múltiples tareas simultáneamente. Debido a que se han empleado distintos métodos un tanto complejos para generar las respuestas, como para generar los diagramas de clase y el análisis de imágenes, se le ha asignado un valor de 4.
- **Reusabilidad:** capacidad de reutilizar el código desarrollado en otros proyectos o componentes. Teniendo en cuenta que gran parte del código que gestiona el apartado de usuario y el que gestiona la generación de respuestas por parte de ChatGPT se puede reutilizar en otro sistema que integre IA, se le ha asignado un valor de 1.
- **Facilidad de instalación:** determina si el proceso de instalación del software es sencillo o complejo, lo cual puede afectar la adopción del sistema. Debido a que el entorno en el que se ha desarrollado en un entorno web que no requiere de configuración por parte del usuario para poder utilizar la aplicación, se le ha asignado un valor de 0.
- **Portabilidad:** considera si el software necesita ser operable en diferentes plataformas o sistemas operativos. Teniendo en cuenta que se necesita una

gran capacidad de portabilidad para que los alumnos puedan utilizar la aplicación en cualquier entorno, se le ha asignado un valor de 4.

- **Facilidad de cambio:** mide cuán fácil es modificar o actualizar el software después de su implementación inicial. Debido a que se depende en cierta medida de la forma en la que se utiliza y se generan las respuestas por parte de la API de OpenAI, se le ha asignado un valor de 2.
- **Concurrencia:** evalúa si el sistema puede manejar múltiples usuarios o procesos accediendo al mismo tiempo sin degradación del servicio. Teniendo en cuenta que la funcionalidad relativa al *feedback* implica una cierta concurrencia de usuarios a la hora de mandar mensajes, se le ha asignado un valor de 3.
- **Características especiales de seguridad:** valora la importancia y complejidad de las características de seguridad implementadas en el software. A este factor se le ha asignado un valor de 4 debido a que se utilizan características de seguridad en varios puntos de la aplicación como, por ejemplo, la conexión a la web segura (HTTPS) y la verificación de correo electrónico para evitar problemas a la hora de registrar a los usuarios.
- **Acceso directo a terceras partes:** determina si el sistema necesita integrarse o comunicarse con sistemas externos o de terceros. Debido a que el plan es integrar la aplicación en plataformas que utilicen los docentes y que internamente la aplicación se comunica con recursos externos, se le ha asignado un valor de 2.
- **Se requiere entrenamiento especial del usuario:** estima la cantidad de entrenamiento que necesitarán los usuarios para operar el software eficientemente. A este factor de complejidad se le ha asignado un valor de 2 ya que, aunque la interfaz se intuitiva y fácil de manejar, puede que al principio cueste que la IA genere lo que el usuario desee exactamente.

### Factores de complejidad del entorno

- **Familiaridad con UML:** evalúa cuánto conocimiento y experiencia tienen los miembros del equipo en utilizar el Lenguaje Unificado de Modelado. Debido a que se ha tenido que volver a leer y revisar gran parte de la documentación UML disponible para poder realizar este trabajo, se le ha asignado un valor de 2.
- **Trabajadores a tiempo parcial:** considera el impacto de tener miembros del equipo que no trabajan a tiempo completo, lo que puede afectar la continuidad y el ritmo del proyecto. Debido a que el proyecto se ha desarrollado a la vez que se cursaban otras asignaturas, se le ha asignado un valor de 3.
- **Capacidad de los analistas:** mide la habilidad y competencia de los analistas en entender y especificar los requisitos del software correctamente. Debido a que no ha sido de gran dificultad la identificación de los distintos tipos de requisitos del sistema, se le ha asignado un valor de 1.
- **Experiencia en la aplicación:** hace referencia a la cantidad de experiencia que tiene el equipo con aplicaciones similares a la que están desarrollando. Debido a que se habían realizado distintos tipos de aplicaciones tanto en el ámbito académico, como en el personal, se le ha asignado un valor de 4.



- **Experiencia en orientación a objetos:** valora cuánto conocimiento tiene el equipo en programación orientada a objetos, un paradigma crucial para muchos proyectos de software. A este factor de complejidad del entorno se le ha asignado un valor de 3 debido a que se tiene una base sólida, aunque no muy extensa de programación orientada a objetos.
- **Motivación:** evalúa el nivel de motivación del equipo, que puede afectar directamente la productividad y la calidad del trabajo. Teniendo en cuenta que no se había realizado un trabajo de estas características para el ámbito académico por parte del equipo y se quiere demostrar los conocimientos obtenidos, se le ha asignado un valor de 5.
- **Dificultad del lenguaje de programación:** considera cómo la complejidad del lenguaje de programación utilizado puede influir en el desarrollo del proyecto. Se le ha asignado un valor de 2 debido a que, el lenguaje de programación y la manera de usarlo ha sido de gran facilidad y sencillez.
- **Estabilidad de los requisitos:** mide la frecuencia con que cambian los requisitos del proyecto, lo cual puede complicar el desarrollo y la planificación. Aunque la gran parte de los requisitos se ha mantenido fiel a los que se especificaban en la propuesta, ha habido otros que han variado y que han añadido cierta complejidad a la hora de desarrollar la aplicación. Es por esto, por lo que se le ha asignado un valor de 2 a este factor.



Figura 71. Estimación de esfuerzo en el programa EZEstimate

Como se puede ver en la Figura 71, después de haber introducido los casos de uso especificados en apartados anteriores (ver Anexo 1) y asignarles una complejidad, el programa muestra el valor total de las horas por punto de caso de uso (Hrs/UCP) que harán falta para realizar el proyecto. Originalmente el programa asigna un valor de 20 a la variable "Total Effort", pero dicho valor se ha tenido que reducir a un valor más realista para poder obtener un número de horas por punto de caso de uso más razonables (820,04). Para poder obtener la cantidad de tiempo en meses que llevaría desarrollar la aplicación se puede dividir la anterior cantidad entre 150, ya que este valor se corresponde con las horas de trabajo mensuales por persona. Después de hacer ese cálculo, se obtiene un valor de 5,46 meses, lo que se asemeja en gran medida al tiempo real empleado en desarrollar por completo la aplicación.

## Anexo 4: Plan de seguridad

El Plan de seguridad es un componente esencial en la gestión de riesgos, diseñado para proteger todos los elementos críticos asociados con el proyecto. Este apartado define las metodologías y herramientas que se ha implementado para asegurar la integridad, confidencialidad y disponibilidad de la información.

En primer lugar, se pueden destacar las leyes relevantes de protección de datos, como por ejemplo el Reglamento General de Protección de Datos (GDPR) europeo. En el cual se incluyen puntos de gran importancia como los siguientes:

- **Leyes de Protección de Datos y Marco Legal:**
  - El GDPR se enfoca en proteger los datos personales y la privacidad de los individuos dentro de la Unión Europea (UE) y extiende sus requisitos a entidades fuera de la UE que procesan datos de personas dentro de la UE. El reglamento busca dar más control a los individuos sobre sus datos personales.
  - **Derechos de los individuos bajo el GDPR:** Incluye el derecho al acceso, rectificación y borrado de sus datos personales, así como el derecho a la portabilidad de los datos.
- **Seguridad Física y Lógica:**
  - **Seguridad física:** Implementación de medidas para proteger los datos y sistemas físicos de accesos no autorizados o alteraciones.
  - **Seguridad lógica:** Uso de técnicas de cifrado para proteger los datos durante su procesamiento y almacenamiento.
- **Dependencias de Terceros:**
  - **Contratos y acuerdos con terceros:** Asegura que todos los terceros que procesan datos cumplan con el GDPR, utilizando cláusulas contractuales estándar o garantías adecuadas.
- **Implementaciones de Seguridad en la Aplicación:**
  - **HTTPS y Certificados Digitales:** Implementación de HTTPS para asegurar las comunicaciones entre los usuarios y el servidor. Uso certificados digitales para autenticar la identidad del servidor.
  - **Cifrado y Copias de Seguridad:** Asegurar que los datos sensibles estén cifrados en tránsito y en reposo. Implementando políticas regulares de copias de seguridad para proteger los datos contra pérdidas o corrupción.

- **Ética:** Asegurar que el proyecto maneje los datos de manera ética, respetando la privacidad y los derechos de los individuos, y considerando las implicaciones éticas del uso de datos personales.

(European Commission, 2020)

En lo que respecta a la aplicación, Django utiliza el algoritmo de hash seguro SHA1 (ver Figura 72), el cual permite codificar las contraseñas de las cuentas de los usuarios para que, de ese modo solo el usuario tenga acceso a su cuenta.

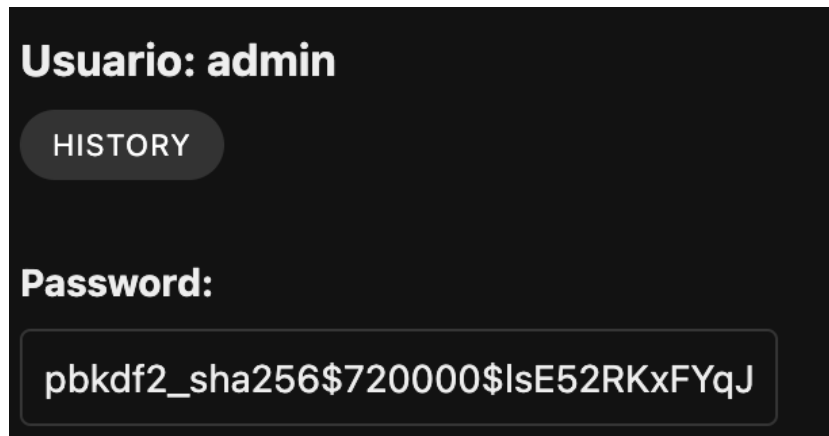


Figura 72. Contraseña codificada

Debido a que la aplicación se ha desplegado en un servidor perteneciente a la Red USAL, las medidas de seguridad con las que cuenta la red de la universidad también se aplican al servidor. Para poder asegurar que la comunicación entre los usuarios y la aplicación es segura, se utiliza el protocolo HTTPS (ver Figura 73). De este modo se cifran los datos intercambiados entre el navegador del usuario y el servidor web mediante certificados SSL/TLS, lo que ayuda a proteger la integridad y la confidencialidad de los datos contra ataques como la interceptación y el *man-in-the-middle*. Este tipo de certificados incluyen información clave sobre la identidad del propietario del certificado, como el nombre de la organización, el nombre del servidor, la ubicación, y están firmados digitalmente por una Autoridad de Certificación (CA) para verificar su autenticidad.

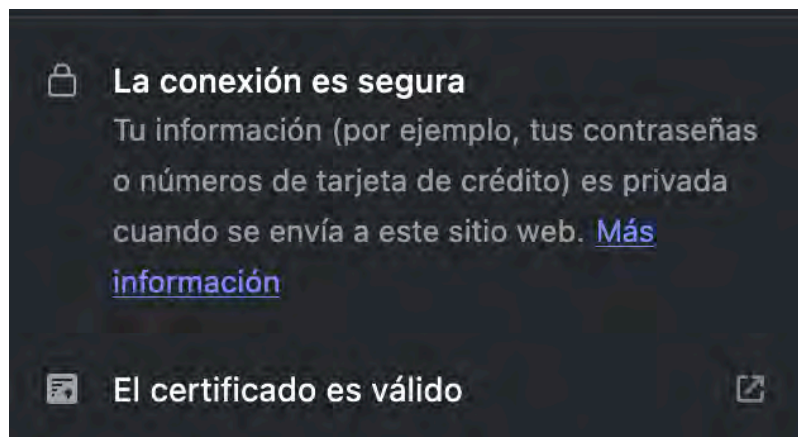


Figura 73. Certificado válido de una conexión HTTPS.