



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**Diseño e implementación de un asistente
de aprendizaje basado en RAG**

Alumno: Celia Ruiz Valero

Tutores: Miguel Ángel Martínez Prieto
Aníbal Bregón Bregón

Fecha: 26 de junio de 2025

Diseño e implementación de un asistente de aprendizaje basado en RAG

Celia Ruiz Valero

26 de junio de 2025

A mi familia y seres queridos.

*El mayor peligro para la mayoría de nosotros
no es que nuestra meta sea demasiado alta y no la alcancemos,
sino que sea demasiado baja y la logremos.*

— Michelangelo Buonarroti

*Haz lo mejor que puedas hasta que sepas más.
Entonces cuando sepas más, hazlo mejor.*

— Maya Angelou

Haz lo que puedas, con lo que tienes, donde estás.

— Theodore Roosevelt

Agradecimientos

En primer lugar, me gustaría agradecer a los profesores D. Aníbal Bregón Bregón y D. Miguel Ángel Martínez Prieto por tutorizar este proyecto y también por aportarme sus diferentes consejos, indicaciones, ayuda constante; por su accesibilidad y también por la confianza depositada en mí a lo largo del mismo. Además, quiero mostrar mi gratitud al resto de profesores de la Escuela de Ingeniería de Informática de Segovia por el trato recibido a lo largo de todos estos años, cercanía, actitud, dedicación y esfuerzo a la hora de trasmitir sus conocimientos. Gracias a todos por acompañarme en esta etapa.

También quiero agradecer a mis amigos y a toda mi familia por el apoyo incondicional a lo largo de todos estos años, en especial a mi padre Ángel, a mi madre Milagros, a mi tía Montse, a José, a Víctor y a mi hermana María. Gracias por vuestra paciencia, apoyo y cariño a lo largo de toda mi vida.

Resumen

Los *Large Language Models* (LLMs) son modelos de inteligencia artificial diseñados para comprender, generar y manipular lenguaje natural. A través de arquitecturas complejas, como las redes neuronales transformadoras (*transformers*), estos modelos se han entrenado con una gran cantidad de corpus de texto, permitiéndoles de esta manera capturar una amplia gama de conocimientos lingüísticos y contextuales. La generalización en el uso de LLMs ha traído consigo la creación de una gran variedad de herramientas asistenciales de diferentes tipos, como por ejemplo los asistentes de aprendizaje. Un asistente de aprendizaje es una herramienta diseñada para apoyar y mejorar el proceso educativo de los estudiantes, utilizando inteligencia artificial y técnicas avanzadas de procesamiento del lenguaje natural. Estos asistentes permiten interactuar con los alumnos de una manera personalizada: respondiendo a sus preguntas, proporcionando explicaciones detalladas, guiándolos a través de conceptos complejos, etc.

El presente Trabajo de Fin de Grado tiene como objetivo desarrollar un asistente de aprendizaje basado en *Large Language Models* (LLMs) que apoye a los estudiantes en la comprensión y el dominio de conceptos técnicos en un marco de trabajo basado en eXtreme Learning y para una asignatura determinada. Este asistente proporcionará una experiencia de aprendizaje interactiva y personalizada, facilitando el acceso a información relevante, generando ejercicios prácticos y ofreciendo respuestas a preguntas específicas de la materia. Además, ayudará a los estudiantes a seguir su progreso y a identificar áreas de mejora continua. La construcción del asistente se llevará a cabo siguiendo el enfoque RAG (*Retrieval-Augmented Generation*), con el objetivo de que el asistente disponga de información relevante sobre la asignatura (objetivos de aprendizaje, recursos educativos...) y pueda utilizarla para proporcionar respuestas precisas y personalizadas para los estudiantes. Finalmente, se desarrollará una interfaz de usuario a modo de prueba de concepto que de soporte a todo el procedimiento asistencial del proceso de enseñanza-aprendizaje y se evaluará el resultado final del proyecto sobre una asignatura real, perteneciente al Grado en Ingeniería Informática de Servicios y Aplicaciones.

Palabras claves: Large Language Models (LLMs), Asistentes de aprendizaje, Procesamiento del Lenguaje Natural, Retrieval-Augmented Generation (RAG), Educación Personalizada, eXtreme Learning.

Abstract

Large Language Models (LLMs) are artificial intelligence models designed to understand, generate, and manipulate natural language. Through complex architectures such as transformer neural networks, these models have been trained on vast amounts of textual data, enabling them to capture a wide range of linguistic and contextual knowledge. The widespread use of LLMs has led to the creation of a broad variety of assistive tools, including learning assistants. A learning assistant is a tool designed to support and enhance students' educational processes, using artificial intelligence and advanced natural language processing techniques. These assistants enable personalised interaction with students by answering their questions, providing detailed explanations, guiding them through complex concepts, and more.

This project aims to develop a learning assistant based on Large Language Models (LLMs), to support students in understanding and mastering technical concepts within a framework based on eXtreme Learning, and tailored to a specific subject. The assistant will offer an interactive and personalised learning experience, facilitating access to relevant information, generating practical exercises, and providing answers to specific questions related to the subject. Moreover, it will help students monitor their progress and identify areas for continuous improvement.

The assistant will be built following the Retrieval-Augmented Generation (RAG) approach, with the objective of equipping the assistant with relevant information about the subject (learning objectives, educational resources, etc.), which it can then use to provide accurate and personalised responses to students. Finally, a user interface will be developed as a proof of concept, supporting the entire educational assistance process. The final outcome of the project will be evaluated using a real subject from the Bachelor's Degree in Computer Engineering for Services and Applications.

Keywords: Large Language Models (LLMs), Learning Assistants, Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG), Personalized Education, eXtreme Learning.

Índice general

Lista de figuras	v
Lista de tablas	ix
I Descripción del proyecto	1
1. Introducción	3
1.1. Planteamiento del problema	7
1.2. Objetivos	8
1.3. Condicionantes	10
1.4. Alcance del proyecto	10
1.5. Estructura de la memoria	12
2. Gestión del Proyecto	15
2.1. Metodología de trabajo	15
2.1.1. Roles	16
2.1.2. Eventos	17
2.1.3. Artefactos	19
2.1.4. Entorno de Trabajo	19
2.2. Interacción	19
2.3. Planificación temporal	20
2.3.1. Sprint #1	23
2.3.2. Sprint #2	26
2.3.3. Sprint #3	28
2.3.4. Sprint #4	30
2.4. Presupuestos	32
2.4.1. Recursos <i>Hardware</i>	32
2.4.2. Recursos <i>Software</i>	33
2.4.3. Recursos Humanos	34
2.4.4. Costes Totales	36
2.5. Gestión de Riesgos	37
2.5.1. Identificación de Factores de Riesgo	37
2.5.2. Estimación de Riesgos	37
2.5.3. Matriz Probabilidad - Impacto	38
2.5.4. Plan de Contingencia	41

Índice general

2.6.	Balance temporal y económico	42
2.6.1.	Balance temporal	42
2.6.2.	Balance económico	50
3.	Antecedentes	53
3.1.	Tecnología Educativa	55
3.1.1.	Contexto de Mercado	56
3.1.2.	Stakeholders	57
3.1.3.	Desafíos y Normativas	58
3.2.	Estado del arte	59
3.2.1.	Descripción de trabajos relacionados	60
3.2.2.	Discusión	65
3.3.	eXtreme Learning	67
3.3.1.	Características del Producto de Aprendizaje	67
4.	Inteligencia Artificial Generativa	71
4.1.	Fundamentos	73
4.1.1.	Machine Learning	73
4.1.2.	Redes Neuronales	79
4.1.3.	DeepLearning	83
4.2.	Arquitectura	86
4.2.1.	Redes Neuronales Recurrentes - RNN	86
4.2.2.	Redes de Memoria Larga a Corto Plazo - LSTM	86
4.2.3.	Autoencoders	87
4.2.4.	Encoder-Decoder	88
4.2.5.	Transformers	89
4.3.	IA Generativa	96
4.3.1.	PNL	97
4.3.2.	LLM	98
4.3.3.	Tipos de LLM	98
4.4.	Ingeniería de Prompting	99
4.4.1.	Técnicas Básicas	100
4.4.2.	Técnicas Avanzadas	100
4.5.	Retrieval Augmented Generation - RAG	101
4.5.1.	Funcionamiento	103
4.5.2.	Arquitectura	107
4.5.3.	Ventajas/Beneficios	108
4.5.4.	Limitaciones RAG	109
4.5.5.	Asistente de Aprendizaje	110
4.5.6.	Fundamentos Técnicos	110
4.5.7.	Comentarios	113
II	Desarrollo de la solución	115
5.	Análisis	117
5.1.	Actores	117

5.2. Requisitos de usuario	117
5.2.1. Casos de Uso	118
5.3. Requisitos de información	124
5.3.1. Modelo Entidad Relación	124
5.3.2. Diccionario de Datos	125
5.4. Requisitos no funcionales	130
5.5. Limitaciones y Restricciones	132
5.5.1. Limitaciones del Proyecto	132
5.5.2. Restricciones de Interfaz Externa	132
6. Diseño	133
6.1. Arquitectura Lógica Alto Nivel	133
6.2. Arquitectura Física	134
6.3. Arquitectura Lógica Bajo Nivel	134
6.3.1. Componentes	134
6.3.2. Comportamiento	138
6.4. Diagramas	140
6.4.1. Diagrama de Despliegue	140
6.4.2. Diagramas de Componentes	141
6.4.3. Diagramas de Secuencia	142
6.4.4. Diagramas de Flujo	146
6.4.5. Diagramas de Clases	149
6.5. Datos	149
6.5.1. Base de Datos	151
6.5.2. Respuesta de Asistente	153
6.5.3. Progreso Alumno	153
6.6. Interfaz de usuario	155
7. Implementación	163
7.1. Herramientas	163
7.2. Desarrollo	165
7.2.1. Estructura	165
7.2.2. Procesamiento de Datos y Creación de la Base de Datos: <code>process.py</code> . . .	165
7.2.3. Configuración <code>config.py</code>	167
7.2.4. RAG <code>backend.py</code>	168
7.2.5. Prompts <code>prompts.py</code>	175
7.2.6. Interfaz <code>main.py</code>	183
7.2.7. Directorio <code>paginas</code>	183
7.3. Dificultades	188
7.4. Aspectos Clave	191
8. Pruebas	193
8.1. Casos de Prueba	193
8.2. Análisis: Defectos y Mejoras	197

III Resultados	201
9. Aceptación	203
9.1. Diseño experimental	203
9.2. Métricas de Éxito	203
9.2.1. <i>Retrieval-Augmented Generation Assessment Scores - RAGAS</i>	203
9.2.2. Tiempo de Respuesta	207
9.2.3. Tiempo de Carga	207
9.2.4. Usabilidad Visual de la Interfaz	207
9.2.5. Satisfacción del Profesor	207
9.2.6. Otras herramientas y métricas de evaluación	208
9.2.7. Resumen Métrica de Éxito	209
9.2.8. Recursos	209
9.2.9. Datos	209
9.3. Plan de aceptación	210
9.3.1. Fase 1: Preparación	211
9.3.2. Fase 2: Ejecución de Pruebas	211
9.3.3. BLOQUE 4: Ver Progreso	214
9.3.4. Fase 3: Presentación de Resultados	215
9.3.5. Análisis Resultados	218
9.3.6. Entorno de evaluación	219
10. Conclusiones y trabajo futuro	221
10.1. Conclusiones	221
10.1.1. Perspectiva del proyecto	221
10.1.2. Perspectiva personal	223
10.2. Trabajo futuro	224
Bibliografía	227
IV Apéndices	253
A. Manual de Instalación	255
B. Manual de Usuario	257
B.0.1. Prerrequisitos	257
B.0.2. Funcionamiento Producto Final	257

Índice de figuras

1.1.	Impacto de la IA en la Educación	6
1.2.	Alcance	11
2.1.	Cuaderno De Trabajo	20
2.2.	Estimación Sprint 1	24
2.3.	Cronograma Sprint 1	25
2.4.	Estimación Sprint 2	27
2.5.	Cronograma Sprint 2	27
2.6.	Cronograma Sprint 3	28
2.7.	Estimación Sprint 3	29
2.8.	Estimación Sprint 4	31
2.9.	Cronograma Sprint 4	31
2.10.	Autoevaluación Sprint 1	42
2.11.	Carga de Trabajo Sprint 1	43
2.12.	Cronograma Sprint 1	43
2.13.	Cronograma Sprint 2	44
2.14.	Carga Trabajo Sprint 2	45
2.15.	Autoevaluación Sprint 2	45
2.16.	Carga Trabajo Sprint 3	46
2.17.	Autoevaluación Sprint 3	46
2.18.	Cronograma Sprint 3	47
2.19.	Carga Trabajo Sprint 4	48
2.20.	Autoevaluación Sprint 4	48
2.21.	Cronograma Sprint 4	49
2.22.	Sprint 2 Tokens API	50
2.23.	Gasto Sprint 2	50
2.24.	Sprint 3 Tokens API	50
2.25.	Gasto Sprint 3	50
2.26.	Sprint 4 Tokens API	51
2.27.	Gasto Sprint 4	51
2.28.	Coste API Total	51
3.1.	Inicio Knewton	60
3.2.	Support Knewton	60
3.3.	IBM Watson Tutor	61
3.4.	Jill Watson	61

Índice de figuras

3.5. Khanmigo	62
3.6. Socratic	62
3.7. Perplexity AI	63
3.8. Mathew	64
3.9. Objetivos del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos	68
3.10. Historias de Aprendizaje del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos	68
3.11. Ejemplo Criterio de Aceptación del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos	69
4.1. Etapas Proceso ML	74
4.2. Estructura de una Neurona	80
4.3. Funciones de Activación	82
4.4. Estructura de una Neurona Artificial	82
4.5. Capas de una Red Neuronal Artificial	83
4.6. Estructura RNN	86
4.7. Estructura de un LSTM	87
4.8. Estructura <i>Autoencoder</i>	87
4.9. Estructura <i>Encoder - Decoder</i>	89
4.10. Estructura <i>Transformer</i>	90
4.11. Arquitectura de una Red Neuronal <i>Feedforward</i>	91
4.12. Estructura <i>Transformer</i> Funcionamiento	94
4.13. Funcionamiento <i>Transformer</i>	95
4.14. Ejemplo de Flujo de Optimización de LLM	102
4.15. <i>Retrivial-Augmented-Generation</i>	103
4.16. Proceso General RAG	104
4.17. Arquitectura RAG	108
4.18. Flujo RAG	108
5.1. Casos de Uso	118
5.2. Modelo Entidad Relación	124
5.3. Diccionario de Datos - CUESTIÓN	125
5.4. Diccionario de Datos - OBJETIVO	125
5.5. Diccionario de Datos - ENUNCIADO	125
5.6. Diccionario de Datos - HISTORIA	126
5.7. Diccionario de Datos - TEST	126
5.8. Diccionario de Datos - CRITERIO	126
5.9. Diccionario de Datos - PREGUNTA	127
5.10. Diccionario de Datos - RESPUESTA	127
5.11. Diccionario de Datos - PRESENTACIÓN	127
5.12. Diccionario de Datos - FRAGMENTO	128
5.13. Diccionario de Datos - ABARCAR	128
5.14. Diccionario de Datos - REFLEJAR	128
5.15. Diccionario de Datos - DESCRIBIR	128
5.16. Diccionario de Datos - CONTENER	129
5.17. Diccionario de Datos - DETALLAR	129

5.18. Diccionario de Datos - TENER	129
5.19. Diccionario de Datos - POSEER	129
5.20. Diccionario de Datos - DIVIDIR	129
5.21. Diccionario de Datos - HABER	130
5.22. Diccionario de Datos - RELACIONAR	130
5.23. Ejemplo <i>Status</i> Erróneo de OpenAI a día 10 de Junio	132
6.1. Arquitectura Lógica Alto Nivel	133
6.2. Arquitectura Física	134
6.3. Diagrama de Despliegue	140
6.4. Diagrama de Componentes	141
6.5. Diagrama Secuencia: Caso de Uso Consulta	142
6.6. Diagrama Secuencia: Test	143
6.7. Diagrama Secuencia: Problemas	144
6.8. Diagrama Secuencia: Ver Progreso	145
6.9. Consulta Flujo	146
6.10. Ver Progreso Flujo	146
6.11. Problema Flujo	147
6.12. Test Flujo	148
6.13. Diagrama de Clases	149
6.14. Inicio	155
6.15. Consulta	156
6.16. Tipo Test 1	157
6.17. Tipo Test 2	158
6.18. Tipo Test 3	159
6.19. Problemas 1	160
6.20. Problemas 2	161
6.21. Ver Progreso	162
7.1. Estructura Proyecto	165
7.2. Prompt Consulta	176
7.3. Prompt Test	177
7.4. Prompt Problemas: Enunciados Modelos ER	178
7.5. Prompt Problemas: Funcionalidades SQL	179
7.6. Prompt Problemas: Consulta SQL	180
7.7. Prompt Problemas: Soluciones Modelo ER	181
7.8. Prompt Problemas: Soluciones Funcionalidades SQL	182
8.1. Casos de Prueba Caja Negra	194
8.2. Casos de Prueba Caja Blanca	197
9.1. Preparación Datos	212
9.2. Ejecución	212
9.3. Gráfica Evaluación Métricas RAGAS	217
9.4. Gráfica Evaluación Métricas Tiempos	217
9.5. Gráfica Evaluación Métrica Satisfacción del Profesional	218
9.6. Gráfica Evaluación Métrica Usabilidad de la Interfaz	218

Índice de figuras

B.1. Realizar Preguntas Abiertas (1/4)	258
B.2. Realizar Preguntas Abiertas (2/4)	258
B.3. Realizar Preguntas Abiertas (3/4)	259
B.4. Realizar Preguntas Abiertas (4/4)	259
B.5. Solicitar Preguntas Tipo Test (1/7)	260
B.6. Solicitar Preguntas Tipo Test (2/7)	260
B.7. Solicitar Preguntas Tipo Test (3/7)	261
B.8. Solicitar Preguntas Tipo Test (4/7)	261
B.9. Solicitar Preguntas Tipo Test (5/7)	262
B.10. Solicitar Preguntas Tipo Test (6/7)	262
B.11. Solicitar Preguntas Tipo Test (7/7)	263
B.12. Solicitar Supuestos Prácticos (1/3)	264
B.13. Solicitar Supuestos Prácticos (2/3)	264
B.14. Solicitar Supuestos Prácticos (3/3)	265
B.15. Visualizar Progreso de Aprendizaje (1/3)	265
B.16. Visualizar Progreso de Aprendizaje (2/3)	266
B.17. Visualizar Progreso de Aprendizaje (3/3)	266

Índice de tablas

2.1. Resumen Eventos ASAP	18
2.2. Distribución de Paquetes de Trabajo por Sprint	21
2.3. Tabla Planificación Sprint 1	24
2.4. Tabla Planificación Sprint 2	26
2.5. Tabla Planificación Sprint 3	29
2.6. Tabla Planificación Sprint 4	30
2.7. Presupuestos <i>Hardware</i>	33
2.8. Presupuesto Recursos <i>Software</i>	34
2.9. Presupuesto <i>Recursos Humanos</i>	35
2.10. Resumen de Presupuesto	36
2.11. Listado de Riesgos	37
2.12. Riesgos priorizados según impacto y ocurrencia	38
2.13. Tabla de Riesgos, Ocurrencia y Justificación	39
2.14. Tabla de Riesgos, Impacto y Justificación	40
2.15. Tabla de Plan de Contingencia para Riesgos	41
2.16. Costes <i>Recursos Hardware</i>	50
2.17. Costes <i>Recursos Software</i>	51
2.18. Costes <i>Recursos Humanos</i>	52
2.19. Resumen de Costes	52
3.1. Comparativa de tecnologías educativas basadas en IA (1/2)	65
3.2. Comparativa de tecnologías educativas basadas en IA (2/2)	66
4.1. Medidas de desempeño en Clasificación	78
4.2. Medidas de desempeño en predicción	79
4.3. Tabla diferentes modelos LLM	99
5.1. Tabla de Actores y Descripciones	117
5.2. Requisitos de Usuario (RU)	118
5.3. CU-1 Solicitar Pregunta Abierta	119
5.4. CU-2 Solicitar Pregunta Tipo Test	120
5.5. CU-3 Solicitar Supuesto Práctico	121
5.6. CU-4 Visualizar Progreso de Aprendizaje	122
5.7. CU-5 Visualizar Historial de Interacción	123
5.8. Requisitos No Funcionales (RNF)	131

Índice de tablas

6.1.	Distribución de Documentos por Objetivo e Historia	151
6.2.	Distribución de documentos relacionados con los problemas prácticos	152
6.3.	Diccionario de datos CSV	154
6.4.	Interfaz Solicitar Pregunta Abierta: Consulta	155
6.5.	Interfaz Solicitar Pregunta Tipo Test: Tipo Test (1/3)	157
6.6.	Interfaz Solicitar Pregunta Tipo Test: Tipo Test (2/3)	158
6.7.	Interfaz Solicitar Pregunta Tipo Test: Tipo Test (3/3)	159
6.8.	Interfaz Solicitar Supuesto Práctico: Problemas (1/2)	160
6.9.	Interfaz Solicitar Supuesto Práctico: Problemas (2/2)	161
6.10.	Interfaz Visualizar Progreso de Aprendizaje: Ver Progreso	162
8.1.	Caso de Prueba CP-1: Consulta General	194
8.2.	Caso de Prueba CP-2: Generación de Problema	195
8.3.	Caso de Prueba CP-3: Generación de Test	195
8.4.	Caso de Prueba CP-4: Recuperación RAG	196
8.5.	Caso de Prueba CP-7: Fallo en la API del LLM	196
9.1.	Métricas de Evaluación del Sistema	209
9.2.	Evaluación del Sistema en el Bloque 1	215
9.3.	Evaluación del Sistema en el Bloque 2	215
9.4.	Evaluación del Sistema en el Bloque 3: ER	216
9.5.	Evaluación del Sistema en el Bloque 3: SQL	216
9.6.	Evaluación del Sistema en el Bloque 3: SQL (2)	217
9.7.	Evaluación del Sistema en el Bloque 4	217
A.1.	Versión de Paquetes para el entorno: Producto	255
A.2.	Versión de Paquetes para el entorno: Evaluación	256

Parte I

Descripción del proyecto

Capítulo 1

Introducción

Nos encontramos en un mundo de cambio constante. Múltiples tecnologías y herramientas nacen de un día para otro, y es nuestra decisión estar al corriente de esas nuevas vertientes de información o, por el contrario, dejar pasar la oportunidad. No obstante, como dijo Darwin: “No es el más fuerte de las especies el que sobrevive, tampoco es el más inteligente, sino aquel que mejor se adapta al cambio”.

La ciencia nos acompaña durante ese camino evolutivo. Gracias a ella se ha podido progresar en sociedad, y es nuestro deber moral como ciudadanos de este planeta poder emplearla para un bien común. Desde esta perspectiva, el presente Trabajo de Fin de Grado (TFG) se plantea como un proyecto de investigación cuyo objetivo es desarrollar una solución accesible y equitativa dentro del ámbito educativo. A pesar de los avances tecnológicos, muchas de las soluciones propuestas para mejorar la enseñanza no son accesibles para todos debido a barreras tecnológicas, sociales o económicas [321]

Bien es cierto que el sistema educativo tradicional español garantiza cierto acceso al material didáctico y a recursos como dispositivos y redes en muchos centros escolares [111], pero sigue presentando limitaciones. La principal deficiencia radica en la falta de herramientas tecnológicas que permitan personalizar la enseñanza, ya que el modelo actual parte de la suposición de que todos los estudiantes poseen la misma capacidad, situación y motivación para aprender. Esta uniformidad en el enfoque educativo no tiene en cuenta la diversidad de perfiles estudiantiles y sus necesidades individuales, lo que puede generar desigualdades en el aprendizaje. [233]

Esta inflexibilidad hacia el alumno no contempla la diversidad de los estudiantes ni sus distintas formas, ritmos y necesidades de aprendizaje. Como consecuencia, muchos alumnos enfrentan dificultades que pueden afectar negativamente su desarrollo académico y emocional, especialmente aquellos con necesidades específicas, como estudiantes con Trastorno por Déficit de Atención e Hiperactividad (TDAH), dislexia o dificultades auditivas. [325].

Sin embargo, este problema no se limita solo a quienes tienen diagnósticos médicos, sino que también afecta a aquellos que requieren de más tiempo para asimilar ciertos conceptos, a los que necesitan reforzar el contenido para ganar seguridad en su aprendizaje o, simplemente, a quienes no pudieron asistir un día a clase y necesitan “un último repaso” con un apoyo adicional.

Además, toda consecuencia negativa se magnifica cuando se trata de la enseñanza durante las épocas más tempranas, que es cuando más se aprende, pero también cuando más se influye directamente en la vida del estudiante. Es ahí cuando se forjan las bases de su futuro, seguridad personal y confianza en sí mismo, por lo que una experiencia negativa en los primeros años de educación no solo impacta en el conocimiento que van a adquirir, sino también en la percepción de

Capítulo 1. Introducción

ellos mismos y su capacidad para resolver futuros desafíos. [133]. Cabe destacar que la UNESCO enfatiza que una educación temprana de calidad no solo impacta en el conocimiento adquirido, sino también en la autoestima y la capacidad de los niños para afrontar desafíos futuros.[324]

La educación es un derecho fundamental reconocido a nivel universal, así como la libertad, acceso e igualdad en la enseñanza. En cambio, no puede haber igualdad cuando se discriminan indirectamente a aquellos que trabajando son incapaces de poder adaptarse a los ritmos impuestos por el sistema por, simplemente, ser como son. Por ser diferentes a lo establecido [268].

Esta falta de personalización no solo dificulta su aprendizaje, sino que también puede afectar su autoestima y percepción de sus propias capacidades [38]. En consecuencia, el sistema educativo actual, al no contemplar estrategias flexibles e inclusivas, perpetúa desigualdades y deja atrás a quienes no encajan en su estructura tradicional [325]

No se trata de igualdad, si no de equidad. Igualdad implica ofrecer los mismos recursos a todos, sin importar sus diferencias individuales. Equidad trata de enfocar la enseñanza en función de las capacidades de los alumnos personificando sus necesidades. Por tanto, no se trata de dar un trato preferente, sino de implicarse verdaderamente con las personas y asegurarse de que aprendan. No se trata solo de dar los recursos, si no de empatizar, de respetar el proceso único de aprendizaje y reconocer la dignidad de cada alumno, promoviendo siempre un entorno inclusivo y justo para todos.

Además, es necesario que todo este proceso se lleve a cabo en un entorno seguro. Hoy en día cada vez son más las noticias que abren telediarios hablando sobre la salud mental y, lamentablemente, también en el ámbito de la educación. Son muchas las personas que se encuentran perdidas en su futuro, sin ambición o que se sienten incompetentes debido a sus resultados académicos. Estrés, cansancio y ansiedad son palabras que desafortunadamente ya se incorporan a nuestro lenguaje diario, agravado cada vez más por la incapacidad de lograr alcanzar el ritmo de la clase, no encontrar el apoyo necesario o verse desmotivados y frustrados por los estudios.

Según Patiño Domínguez [145] “el bienestar psicosocial es fundamental para el bienestar general de los estudiantes. Un entorno escolar y familiar que promueva la salud mental y emocional puede contribuir a que los jóvenes se sientan seguros, apoyados y capaces de enfrentar los desafíos de la vida cotidiana”. Asimismo, la UNESCO ha manifestado su compromiso con la salud mental en el ámbito educativo, destacando la necesidad de integrar el bienestar psicosocial como parte fundamental del proceso de aprendizaje para garantizar una educación inclusiva y sostenible [325]

Por lo tanto, ha surgido un nuevo desafío vinculado a la comprensión, flexibilidad y adaptación de los contenidos de aprendizaje, donde la inclusividad, interactividad y claridad sean claves para garantizar una grata experiencia en la enseñanza individualizada del alumno. Todo ello de la mano de las nuevas tecnologías que facilitan nuestro día a día y sirven para crear un futuro común mejor.

La cuestión que surge es por qué no aprovechar la inteligencia artificial (IA) en el ámbito educativo para hacerla accesible a todos. La respuesta parece sencilla, pero en la práctica, su implementación presenta diversos desafíos.

Últimamente, se ha hecho plausible el uso exponencial de la IA en todos los sectores. Sin embargo, las soluciones que puede ofrecer esta tecnología no son igualmente accesibles para todos debido a diversas circunstancias, como limitaciones económicas, sociales o tecnológicas. Esta disparidad plantea un reto en términos de equidad en el acceso a las innovaciones tecnológicas, especialmente en el ámbito educativo. En este contexto, la cuestión que surge es por qué no utilizar la IA de manera que sea accesible para todos, brindando así nuevas oportunidades de

aprendizaje que complementen y enriquezcan el sistema educativo tradicional. [48]

La clave se encuentra en relacionar la personalización del sistema educativo con la IA, de tal manera que exista algún tipo de medio que acapare tanto la visión del alumno como del profesor. Básicamente, en encontrar la forma de poder utilizarla.

Afortunadamente, la IA ofrece esa oportunidad de transformar la enseñanza proporcionando, por ejemplo, asistentes de aprendizaje. Un asistente de aprendizaje es una herramienta diseñada para apoyar y mejorar el proceso educativo de los estudiantes, utilizando inteligencia artificial y técnicas avanzadas de procesamiento del lenguaje natural (PLN). Estos asistentes permiten interactuar con los alumnos de una manera personalizada: respondiendo a sus preguntas, proporcionando explicaciones detalladas, guiándolos a través de conceptos complejos... [186]

De esta manera, por tanto, se podría conseguir esa comprensión, retroalimentación y flexibilidad centradas en el desarrollo individual que tanto necesitaría el alumno; además de fomentar ese aprendizaje inclusivo, respetuoso y digno. La pandemia de COVID-19 aceleró la adopción de tecnologías digitales en la educación, poniendo en evidencia las limitaciones del sistema tradicional y abriendo la puerta a soluciones innovadoras basadas en IA.[320] Este último año es evidente que ha sido una época clave para la emergencia, popularidad y despliegue de la IA, con avances tecnológicos significativos que han transformado sectores completos, incluido el educativo. Además, las herramientas tecnológicas que utilizan IA permiten personalizar el aprendizaje de manera más eficiente, lo que facilita la inclusión de estudiantes con necesidades diversas. [205] [28]

Entonces, ahora es el momento de aprovechar ese oportunidad e impulso para crear algo innovador pero necesario. Además, hay distintos estudios que afirman que el impacto de la IA en la educación ha ayudado gratamente a los alumnos, así como en reducir su estrés, cuidar su salud mental y especialmente, en obtener unos mejores resultados. [325] [205] [56]

Un ejemplo de ello sería el reciente estudio realizado en China donde se corroboró que el uso de *chatbots* basados en IA ayudó a reducir la ansiedad de los estudiantes al leer, lo que mejoró sus resultados en pruebas de comprensión [343]. Por otro lado, según un artículo de Statista, un alto porcentaje de encuestados considera esencial que los maestros se capaciten para usar la IA en sus métodos de enseñanza [298]. Además, el Monitor Global de Educación de Ipsos refuerza esta idea, señalando la importancia de la formación en IA para los educadores [174]. El estudio de Cobos-Gutiérrez también indica que la integración de la IA en el proceso educativo contribuye a mejorar el rendimiento académico de los estudiantes de secundaria, destacando la relación positiva entre el uso de tecnologías basadas en IA y los resultados académicos [66]. Otro último ejemplo sería la encuesta de investigación a 800 alumnos de Pearson [230], donde se consiguieron conclusiones similares y distintos hallazgos, que se ilustran en la Figura: 1.1

- El 51 % de los estudiantes del semestre de primavera afirmó que la IA generativa les ayudó a obtener mejores calificaciones, un aumento de 4 puntos porcentuales respecto al otoño de 2023.
- El 56 % de los estudiantes dijo que la IA les ayudó a ser más eficientes, con un incremento de 7 puntos porcentuales respecto al semestre anterior.
- Entre los estudiantes de carreras STEM, el 51 % busca que la IA generativa les proporcione guía para resolver problemas.
- Casi el 60 % de los estudiantes muestra interés en probar nuevas herramientas de IA generativa.

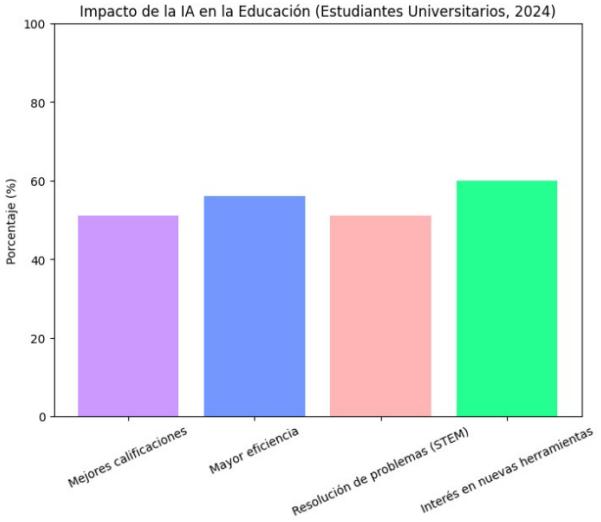


Figura 1.1: Impacto de la IA en la Educación

La IA ha venido para quedarse, y es nuestra elección emplearla para un propósito global, bienestar general y equidad social. Como dijo en su día Bill Gates: “*las grandes oportunidades nacen de haber sabido aprovechar las pequeñas*”. Y es hora de hacerlo.

El año 2023 marcó el inicio del auge de los “*chatbots* avanzados”, o asistentes virtuales basados en inteligencia artificial (IA), con herramientas como ChatGPT, Gemini o DeepSeek ganando rápidamente notoriedad. En la actualidad, estas plataformas son ampliamente conocidas y utilizadas, especialmente por estudiantes que las emplean como recursos para apoyarse en su proceso de aprendizaje, demostrando ser, en muchas ocasiones, una valiosa herramienta.

No obstante, surge un desafío importante. Al intentar obtener respuestas sobre conceptos específicos de una asignatura, los estudiantes frecuentemente se enfrentan a dificultades para recibir respuestas precisas, bien estructuradas y adaptadas a su nivel de comprensión. A pesar de ser de una gran ayuda debido a la inmensa cantidad de información que procesan, los asistentes no siempre son capaces de focalizar, unificar ni aclarar adecuadamente los conceptos que realmente desean los usuarios. La falta de personalización de los asistentes es tangible a medida que se profundiza en su uso. Aunque están diseñados para ser asistentes de aprendizaje, su incapacidad para adaptarse al perfil y las necesidades del usuario genera limitaciones. Además, en ocasiones, los *chatbots* pueden generar respuestas incorrectas o incluso inventar información, lo que agrava aún más la falta de fiabilidad en su uso educativo.

Este es el punto de partida del presente TFG, en el que se abordará la construcción de un asistente de aprendizaje. Para ello se empleará la potencia, tecnología y avances proporcionados por herramientas basadas en inteligencia artificial (IA), como los Modelos de Lenguaje de Gran Escala (LLM), en inglés *Large Language Models*. Estos modelos utilizan técnicas avanzadas de procesamiento del lenguaje natural (PLN) para interactuar con los usuarios, permitiendo que los estudiantes reciban apoyo educativo personalizado a través de respuestas generadas por IA.

Los asistentes de aprendizaje actuales se basan principalmente en LLM, que son modelos preentrenados con grandes cantidades de datos disponibles en Internet, lo que les permite tener una amplia comprensión del lenguaje humano[328]. Como bien mencionaba antes, entre los ejemplos más conocidos de estas herramientas se encuentran ChatGPT o Gemini, pero existen

otros asistentes virtuales que aprovechan esta tecnología para proporcionar interacciones más naturales con los usuarios [45]

No obstante, este enfoque presenta un problema importante si el objetivo conseguir un aprendizaje verdaderamente personalizado, individualizado y adaptado a las necesidades de cada estudiante. Si bien los LLM pueden procesar grandes cantidades de información y generar respuestas rápidas, su falta de personalización es una de las limitaciones más significativas. Esto se debe a que los modelos están entrenados de forma general y no están específicamente diseñados para adaptarse a las peculiaridades del estilo de aprendizaje de cada alumno, lo que puede generar respuestas que no están alineadas con el nivel de comprensión o el contexto educativo particular del estudiante [35]. Esta falta de personalización podría dificultar el proceso de aprendizaje y limitar la efectividad del asistente como herramienta educativa.

Por tanto, la solución principal se va a sustentar en proporcionar material específico y relevante relacionado con una asignatura en el caso de este proyecto. Este material puede incluir recursos como libros, artículos académicos, notas de clase, ejercicios prácticos y cualquier otra fuente que aporte datos directamente aplicables al proceso de aprendizaje. De esta manera, el asistente de aprendizaje podrá basarse en esos datos para ofrecer resultados más precisos, eficientes y eficaces. Para ello, no será necesario construir modelos desde cero, ya que existen varios modelos preentrenados con un gran potencial, como GPT-3 [45] o BERT [93]. Lo que se hará es adaptar y personalizar esos modelos existentes mediante la técnica RAG (Retrieval-Augmented Generation), que combina la generación de lenguaje con la recuperación de información. [197] Este enfoque permite adaptar la experiencia a un dominio específico, lo que facilita la obtención de respuestas más precisas y detalladas, mejorando así el rendimiento y la calidad. Además, ofrece la oportunidad de proporcionar una enseñanza más personalizada y orientada a los objetivos particulares de cada estudiante. [344]

En definitiva, este Trabajo de Fin de Grado propone el desarrollo de un asistente de aprendizaje que mejore la precisión y utilidad de las respuestas ofrecidas al estudiante. Su funcionamiento básico radica en que el estudiante (usuario) introduzca una consulta o solicite una acción, y mediante la técnica RAG el sistema localice el contenido más relevante relacionado con esa consulta dentro de un conjunto de materiales específicos. A continuación, ese contexto recuperado es utilizado por un LLM generar una respuesta clara, coherente y adaptada al ámbito educativo en cuestión; y devuelva la respuesta. Cabe resaltar que la aplicación se realizará como un prototipo de prueba de concepto (PoC) dejando al margen aspectos propios de un despliegue más avanzado, como la gestión de usuarios, una interfaz web avanzada o la concurrencia.

1.1. Planteamiento del problema

A continuación, se expone el planteamiento del problema presentando su estructura: ideal, realidad, consecuencias y propuesta.

- IDEAL: la educación es un pilar esencial para poder sembrar nuestro futuro. Por ello, la situación ideal sería que los alumnos cuenten con un asistente virtual disponible en todo momento que les apoye en la comprensión y el dominio de conceptos técnicos para una asignatura determinada, así como resolver sus dudas y generar y corregir ejercicios en función a los recursos de aprendizaje. Esto podría derivar en un gran avance para conseguir una enseñanza equitativa.

- REALIDAD: sin embargo, actualmente no existe un asistente que aborde de forma integrada las necesidades específicas de cada estudiante o que personalice su experiencia, y tampoco modelos de lenguajes específicos, abiertos y enfocados a la docencia. Los asistentes disponibles suelen estar generalizados, lo que impide que sus respuestas se alineen del todo con los contenidos, objetivos y terminología específica de cada asignatura.
- CONSECUENCIAS: esta situación limita la personalización de los procesos de aprendizaje. Como consecuencia, impide que las respuestas de los asistentes actuales no se alineen en su totalidad con lo impartido en la asignatura, pudiendo llegar a mezclar conceptos. Esto puede dar lugar a explicaciones inexactas, confusión entre significados o impedimentos para alcanzar los estándares y objetivos planteados en la materia, afectando negativamente al rendimiento y experiencia del estudiante.
- PROPUESTA: este proyecto tiene como meta investigar y desarrollar un asistente de aprendizaje que, basándose en los objetivos específicos de la asignatura, permita ayudar al alumno en su proceso formativo. Para ello, se aprovechará LLMs y técnicas del PNL ya existentes, implementando una arquitectura RAG que dé soporte a los casos de uso habituales de una asignatura. El propósito es garantizar respuestas precisas, contextualizadas y personalizadas, ofreciendo así una experiencia de aprendizaje más alineada con las necesidades individuales del alumnado y facilitando una participación efectiva y satisfactoria en la materia.

1.2. Objetivos

Como se indicaba anteriormente, el presente Trabajo de Fin de Grado tiene como objetivo desarrollar un asistente de aprendizaje con que apoye a los estudiantes en la comprensión y el dominio de conceptos técnicos para una asignatura determinada. Este asistente proporcionará una experiencia de aprendizaje interactiva y personalizada, facilitando el acceso a información relevante, generando ejercicios prácticos y ofreciendo respuestas a preguntas específicas de la materia. Además, ayudará a los estudiantes a seguir su progreso y a identificar áreas de mejora continua. Todo ello, da lugar a los siguientes objetivos.

La construcción del asistente se llevará a cabo siguiendo el enfoque RAG (*Retrieval-Augmented Generation*), con el fin de que el asistente disponga de información relevante sobre la asignatura (objetivos de aprendizaje, recursos educativos...) y pueda utilizarla para proporcionar respuestas precisas y personalizadas para los estudiantes. Finalmente, se desarrollará una interfaz de usuario que de soporte a todo el proceso asistencial del proceso de enseñanza-aprendizaje y se evaluará el resultado final del proyecto sobre una asignatura real, perteneciente al Grado en Ingeniería Informática de Servicios y Aplicaciones.

Por tanto, de este modo los objetivos identificados para el desarrollo del proyecto son los siguientes:

OBJ 1: Investigación del uso de la IA Generativa en el ámbito de la educación

- SubObj 1.1:** Realizar el estado del arte sobre LLMs, RAG y su uso en entornos educativos.
- SubObj 1.2:** Analizar funcionamiento, técnicas existentes de RAG y su aplicabilidad en la recuperación de información contextualizada.

OBJ 2: Construir la Base de Conocimiento sobre los materiales de la asignatura estructurando todos sus contenidos.

SubObj 2.1: Analizar y estudiar el material académico proporcionado (*raw data*) relevante necesario para crear la base de información.

SubObj 2.2: Transformar y tratar los datos en un formato estructurado (*smart data*) que permita una fácil recuperación mediante técnicas de RAG, incluyendo además las historias, criterios de aceptación y objetivos de la asignatura.

SubObj 2.3: Emplear técnicas de representación semántica del contenido para mejorar la comprensión y búsqueda de información relevante a partir de las consultas del estudiante.

SubObj 2.4: Integrar un sistema de recuperación de información eficiente que permita extraer el contexto adecuado de la base de conocimiento cuando se presente una consulta.

OBJ 3: Permitir la resolución de dudas de la asignatura a cerca de todos sus contenidos teóricos (Consultas Generales)

SubObj 3.1: Diseñar e implementar un módulo de consultas que permita a los estudiantes introducir dudas generales y recibir respuestas claras, generadas por un LLM, a partir del contexto recuperado por el sistema RAG.

SubObj 3.2: Optimizar las respuestas generadas para asegurarse de que sean comprensibles y se ajusten al nivel del estudiante, adaptando la respuesta del LLM según las necesidades del usuario.

OBJ 4: Generar cuestiones de elección múltiple sobre todos los contenidos teóricos de la asignatura

SubObj 4.1: Desarrollar una funcionalidad que genere preguntas tipo test, a partir del material académico estructurado, utilizando el LLM para generar preguntas en función de los criterios y nivel del estudiante.

SubObj 4.2: Implementar un sistema de evaluación automática que permita al estudiante autoevaluarse después de responder el test, proporcionando retroalimentación basada en las respuestas generadas.

OBJ 5: Generación de problemas de la asignatura

SubObj 5.1: Desarrollar una funcionalidad que permita la generación de un problema práctico a la vez con LLM, en relación con los objetivos y criterios de la asignatura.

OBJ 6: Implementar dashboards analíticos para la personalización y seguimiento del estudiante en la asignatura.

SubObj 6.1: Mostrar el progreso del estudiante mediante una serie de elementos analíticos de su seguimiento o dashboards que muestren al menos 4 indicadores clave de su

desempeño y participación, como aciertos, errores, totales o tiempo medio, además de diversas gráficas que resulten de interés (como filtrado por categoría, totales...) a partir de los datos recogidos de las cuestiones, de tal manera que se asegure la accesibilidad a ellos.

1.3. Condicionantes

No obstante, nos encontramos con una serie de restricciones que pueden llegar a afectar al correcto desarrollo del proyecto en términos de negocio y técnicos, como son:

- REST 1: El proyecto ha de tener una carga de 12 ETCS, lo que corresponde a un rango temporal entre 300 y 360 horas.
- REST 2: El asistente se construirá utilizando modelos de lenguaje disponibles a través de la API de OpenAI.
- REST 3: Las respuestas se basarán en el material educativo proporcionado de la asignatura de Sistemas de Bases de Datos

1.4. Alcance del proyecto

El presente Trabajo de Fin de Grado tiene como objetivo el desarrollo de un asistente de aprendizaje basado en *Large Language Models* (LLMs) para apoyar a los estudiantes en la comprensión y dominio de conceptos técnicos de la asignatura de Sistemas de Bases de Datos, impartida en el 2º curso del Grado en Ingeniería Informática de Servicios y Aplicaciones de la Universidad de Valladolid. Aunque el asistente se evaluará tomando como referencia los objetivos de aprendizaje de la asignatura, cabe destacar que la especificación y desarrollo de estos objetivos se lleva a cabo de acuerdo con el concepto de producto de aprendizaje descrito en eXtreme Learning [213], que será implementado dentro del propio asistente para facilitar la calidad de sus respuestas.

El asistente proporcionará una experiencia de aprendizaje interactiva y personalizada. La Figura 1.2 ilustra el alcance del asistente, que incluye tres características principales:

- Respuesta a preguntas abiertas sobre cualquier aspecto de una asignatura universitaria.
- Generación de preguntas tipos test para reforzar la comprensión de los fundamentos teórico-prácticos de una asignatura universitaria.
- Generación de supuestos prácticos para apoyar el desarrollo de competencias aplicadas en el ámbito de una asignatura universitaria.

El sistema se implementará con un enfoque RAG (*Retrieval-Augmented Generation*), integrando bases de datos de conocimiento con modelos de lenguaje para mejorar la precisión de las respuestas. El fin se sustenta en que el asistente disponga de información relevante sobre la asignatura (objetivos de aprendizaje, recursos educativos...) y pueda utilizarla para proporcionar respuestas precisas y personalizadas para los estudiantes. Finalmente, se desarrollará una interfaz de usuario que de soporte a todo el proceso asistencial del proceso de enseñanza-aprendizaje y se evaluará el resultado final del proyecto sobre la asignatura real ya mencionada.

El propósito principal se centrará en un backend funcional, en el cual también se implementará la opción de seguimiento de progreso del estudiante. No obstante, se realizará de manera paralela esa interfaz de usuario para asegurar una experiencia accesible y sencilla para los alumnos.

Por tanto, el asistente de aprendizaje ofrecerá un entorno interactivo donde los alumnos podrán realizar consultas sobre temas de la asignatura, responder preguntas de test, proporcionar problemas prácticos y visualizar su progreso académico.

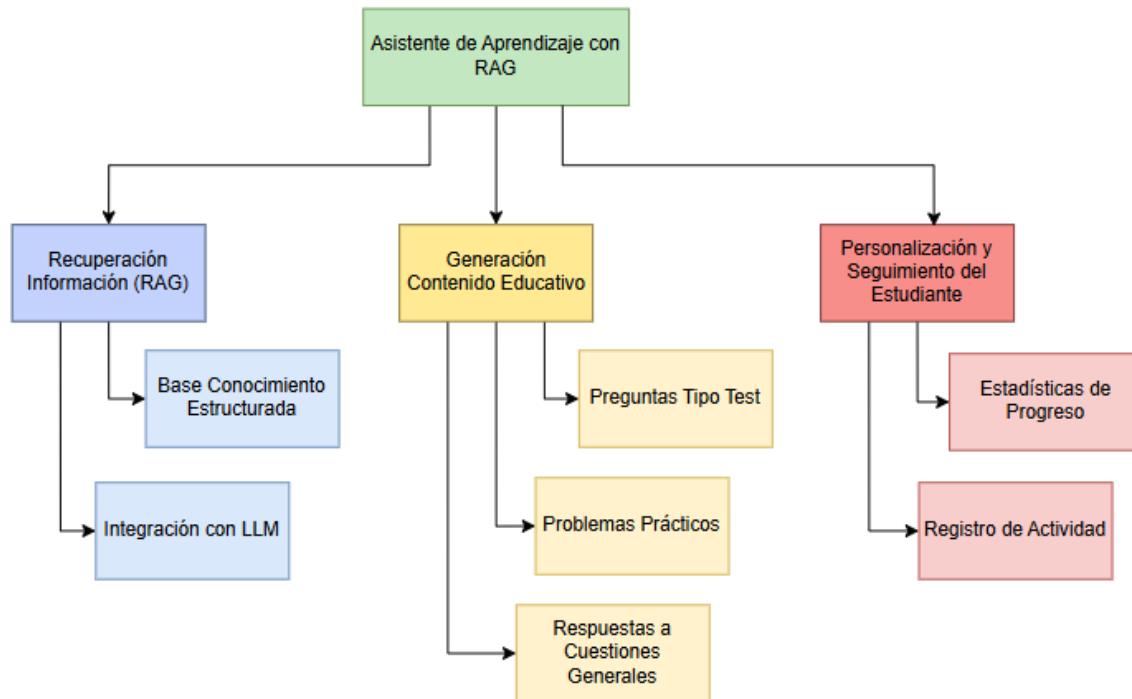


Figura 1.2: Alcance

1.5. Estructura de la memoria

La memoria del presente proyecto se recoge a lo largo de los siguientes bloques y capítulos.

Bloque I: Descripción del proyecto

En este bloque se presenta la relación con el tema a tratar, es decir, el entorno, los desafíos actuales y los conocimientos previos necesarios para desarrollar el producto. Además, se aborda la gestión del proyecto.

Capítulo 1 – Introducción: se expone el desafío de la falta de personalización en el aprendizaje, así como la motivación del proyecto, sus objetivos, alcance y condicionantes.

Capítulo 2 – Gestión del proyecto: se recoge la metodología a emplear para el desarrollo del proyecto, así como la planificación, presupuestos estimados, riesgos, interacción y balance temporal y económico.

Capítulo 3 – Antecedentes: presenta el entorno de negocio y el uso de la inteligencia artificial en el ámbito educativo como medio para solventar el desafío planteado en el capítulo 1. Además, se abordan aspectos clave como los *stakeholders*, los principales desafíos, los trabajos relacionados y el *framework* en el que se apoya el proyecto.

Capítulo 4 – Inteligencia Artificial Generativa: se explica la importancia y funcionamiento de la IA Generativa en el proyecto, además de la técnica RAG. Por tanto, reúne los conocimientos básicos teóricos y técnicos necesarios para poder entender el transcurso del proyecto.

Bloque II: Desarrollo de la solución

Este bloque se centra en todo lo relacionado con el producto, desde la comprensión y análisis del problema hasta la implementación y prueba de la solución final.

Capítulo 5 – Análisis: contiene el análisis y modelado del proyecto desde un punto de vista de requisitos y actores del sistema.

Capítulo 6 – Diseño: se detallan las decisiones importantes de diseño, los diagramas del *software*, el análisis de datos, y un prototipo de la interfaz de usuario.

Capítulo 7 – Implementación: a lo largo de este apartado se detalla el proceso de implementación del producto, especificado de una manera más técnica y abordando también los problemas encontrados y cómo se solventaron.

Capítulo 8 – Pruebas: se presentan diferentes casos de pruebas aplicados al producto para evaluar su comportamiento y verificar que se cumple con todos los requisitos planteados o la calidad necesaria. Además, se realiza un análisis donde se detectan defectos o se proponen distintas áreas de mejoras.

Bloque III: Resultados

Este último bloque aborda el diseño y ejecución de un plan que permita evaluar el cumplimiento de los objetivos establecidos del proyecto, además de presentar las conclusiones finales.

Capítulo 9 – Aceptación: para evaluar el rendimiento del sistema, se propone un diseño experimental que abarque distintas métricas y un plan detallado para su validación.

Capítulo 10 – Conclusiones y trabajo futuro: finalmente, se presentan las principales conclusiones extraídas del proyecto y se proponen futuras líneas de investigación o trabajo.

Anexos

Apéndice A – Manual de instalación

Apéndice B – Manual de usuario

Capítulo 2

Gestión del Proyecto

En este capítulo se detalla la gestión llevada a cabo durante el desarrollo del TFG. Se describe la metodología seguida y la planificación aplicada para el transcurso del proyecto, el presupuesto estimado y finalmente el análisis de las desviaciones observadas. Todo ello con el objetivo de garantizar un enfoque estructurado, eficiente y adaptable.

2.1. Metodología de trabajo

La planificación y desarrollo de este proyecto se llevarán a cabo utilizando la metodología ASAP, (Agile Student Academic Projects), dado su enfoque estructurado y sistemático y su capacidad para fortalecer competencias técnicas y transversales de una manera efectiva y profesional [212].

ASAP se basa en los distintos principios y valores ágiles [32] con el fin de proporcionar un ritmo regular de trabajo y comunicación durante la realización del TFG. Asimismo, define una evolución y estructura clara, colaborativa y sencilla gracias también a un *feedback* constante que contribuye a alcanzar los objetivos del proyecto. Esto genera una gran satisfacción tanto para el alumno como para el profesor, garantizando la consecución de resultados exitosos y valiosos, siempre dentro del plazo, alcance, presupuesto y calidad previamente establecidos.

Por otro lado, es importante destacar que ASAP se estructura en torno a tres componentes fundamentales: roles, eventos y artefactos, los cuales se basan en los mismos elementos utilizados en Scrum [287]. Además, su desarrollo se lleva a cabo dentro de un entorno de trabajo optimizado, con herramientas específicamente seleccionadas para mejorar la productividad, la organización y el rendimiento a lo largo de todo el proyecto.

El objetivo principal de ASAP es plantear una dinámica de trabajo incremental e iterativa sustentada en la interacción regular de las partes y generación frecuente de retroalimentación. Este planteamiento permite adaptar las necesidades de trabajo y avances del proyecto, por lo que trata una capacidad crítica, reflexiva y resolutiva; fomentando una autonomía y mejora de las habilidades de comunicación. ASAP aborda el desarrollo del proyecto en torno a cinco paquetes de trabajo, en los que se enfocan los objetivos de aprendizaje principales de un TFG en Ingeniería Informática:

- **Proyecto:** este paquete de trabajo enfoca el TFG como un proyecto que da respuesta a desafíos realistas y, por tanto, debe abordarse desde la misma perspectiva. Para ello, se propone desarrollar en TFG en un proceso iterativo e incremental, con una carga constante

a lo largo de su ciclo de vida y facilitando el seguimiento y revisión con distintos *feedbacks*. Asimismo, se ha de evitar retrasos, tanto durante la planificación como en el desarrollo y presentación del proyecto final.

- **Antecedentes:** este paquete de trabajo se centra en que el estudiante adquiera los conocimientos previos y necesarios para poder comprender la envergadura del proyecto. A tal efecto, es recomendable considerar distintos artículos y libros para entender el impacto potencial del trabajo a realizar, su entorno de negocio y las soluciones parecidas (teóricas y tecnológicas) ya existentes. Por tanto, se fomenta una labor investigadora para comprender su posible alcance.
- **Desarrollo:** este paquete de trabajo contempla el proceso de construcción del producto con el fin del cumplimiento de los objetivos establecidos en el proyecto y alineados con la metodología seleccionada.
- **Aceptación:** en este paquete de trabajo se aborda la evaluación de los distintos objetivos del proyecto, aunque también el análisis y discusión de los resultados finales obtenidos.
- **Comunicación:** este último paquete tiene como finalidad trabajar las competencias de comunicación oral y escrita, a través de la memoria técnica del proyecto y su presentación, durante el acto de defensa.

2.1.1. Roles

Una vez establecidos los paquetes de trabajo que se abordarán en el TFG, se procede a presentar los componentes de ASAP. El TFG se plantea como un trabajo individual que se elabora de forma autónoma por parte de un estudiante y bajo la supervisión de un tutor. Este será presentado ante un tribunal, que se encargará de su evaluación final.

De acuerdo con este enfoque, ASAP trata de favorecer la interacción de los diferentes roles participantes en el TFG para mejorar el proceso de aprendizaje del estudiante. Para ello, caracteriza de forma específica cada uno de los roles anteriores e introduce un cuarto (“comunidad”) para afianzar la calidad de la interacción entre todos ellos:

- **Estudiante:** es el rol fundamental en ASAP, ya que cuenta con un papel activo en todas las etapas del proyecto. Asume responsabilidades relacionadas con la identificación de tareas, su planificación y ejecución de las mismas, donde se ha de incluir también la memoria técnica y el propio desarrollo del proyecto. Asimismo, es el encargado de la mejora continua del producto y del proceso, la adaptación de forma precisa de los distintos *feedbacks* recibidos y el aseguramiento de una transparencia y comunicación efectiva durante toda la vida del proyecto.
- **Tutor:** el tutor, o tutores, debe asumir la tutela del estudiante y comprometerse a colaborar con él durante el planteamiento del problema y definición de los objetivos. Por otra parte, es importante que también lo guíe en el material bibliográfico y supervise continuamente su avance proporcionando una retroalimentación clara que le sirva de sustento, inspiración y orientación.
- **Comunidad:** está compuesta por aquellas personas que puedan proporcionar un valor adicional al proyecto mediante su punto de vista, investigación o conocimiento. De manera

general, suele concordar con alumnos que se encuentran en la misma tesitura de elaboración del TFG con ideas interconectadas, tutores y profesores o expertos sobre el tema a tratar. Con esto, se logra conseguir distintas perspectivas del proyecto, un apoyo adicional y mayor perfección del producto a entregar.

- **Tribunal:** este rol está compuesto por los profesores que evaluarán el TFG en consecuencia de la calidad de la documentación aportada, el grado de satisfacción de la consecución de los objetivos establecidos y la exposición del estudiante entorno al desarrollo y producto final del proyecto.

2.1.2. Eventos

Los eventos en ASAP son la base en la que se fundamenta la comunicación y trabajo de esta metodología. Al igual que Scrum, se divide en distintos tipos de reuniones que contribuyen a una interacción continua entre los diferentes roles y facilitan un ritmo constante de trabajo. Se destacan cinco eventos:

- **Sprint:** espacio temporal de aproximadamente 1 mes de duración (4-5 semanas) en el que se divide el TFG de forma periódica. En cada sprint, se definen distintos objetivos en conjunto por el alumno y tutor con el fin de desarrollar de forma iterativa e incremental el producto mínimo viable asociado a estos propósitos. Al inicio de cada uno, se establece el objetivo del sprint con distintas historias¹ de usuario que se abordarán y finalizarán en el marco temporal establecido por el sprint. Una vez completadas, se habrán logrado los objetivos del sprint. Asimismo, es importante resaltar que a su finalización, además de presentar la documentación técnica y realizar una presentación de los avances consolidados durante el sprint, se recibe por parte del tutor un *feedback* con mejoras de lo elaborado hasta el momento. Finalmente, resaltar que las propuestas mencionadas en la retroalimentación se considerarán para el siguiente sprint.
- **Reunión de Inicio:** primera reunión celebrada al comienzo de cada sprint. En ella, el tutor define el alcance del siguiente sprint y es el estudiante quien establece las distintas tareas y su planificación temporal que se deberán llevar a cabo para el cumplimiento de esos nuevos objetivos. Destacar que no debe exceder los 30 minutos.
- **Reunión de Sincronización:** reunión semanal programada el mismo día de la semana, a la misma hora y de máxima duración de 15 minutos. En ella interviene el alumno para presentar su avance, dudas y sus puntos de bloqueo al tutor. Por otro lado, este último intentará dar resolución a esos inconvenientes, además de su punto de vista, consejos y opinión.
- **Comunicación de Progresos:** evento que tiene lugar el último día del sprint. En él el alumno ha de exponer al resto de la comunidad y profesores su avance del proyecto hasta dicho momento, emulando como si fuera el acto de defensa final. Tras la exposición, la

¹Una historia de usuario está contenida dentro de un objetivo (compuesto por distintas historias de usuario). Para alcanzar dicho objetivo, es necesario completar con éxito todas sus historias, ya que contribuyen a su progreso. A su vez, cada una de ellas está caracterizada por varios criterios de aceptación que se encuentran relacionados con los requisitos y aportan valor al proyecto mediante su implementación. Igualmente, una historia se completa cuando se han efectuado todos sus resultados.

comunidad evaluará a modo de retroalimentación el progreso descrito, de tal manera que le sirvan al alumno las recomendaciones propuestas.

- **Retrospectiva:** último evento del sprint de duración de 30 minutos donde se recoge la retroalimentación del evento anterior. En ella, se ha de considerar la calidad de lo mencionado en la exposición y recoger de forma anónima los distintos aspectos positivos indicados, aunque también posibles mejoras para la situación del TFG dado.

A forma de resumen, se exhibe una tabla resumen a cerca de los eventos en ASAP:

Evento	Roles Intervienen	Duración	Objetivo
Sprint	Estudiante	4 - 5 semanas	Desarrollar de forma iterativa e incremental el proyecto para obtener un MVP
Reunión Inicio	- Estudiante - Tutor	- 30 min - 1 vez por sprint	Definir objetivo sprint
Reunión Sincronización	- Estudiante - Tutor	- 15 min - 1 vez por semana	- Revisar progreso - Analizar bloqueos - Presentar plan de trabajo para la siguiente semana
Comunicación Progresos	- Estudiante - Tutor - Comunidad	- 1 hora por alumno - Último día del Sprint	Simular acto de defensa Recibir feedback
Retrospectiva	Estudiante Tutor Comunidad	- 30 min - Final de Comunicación de Progresos	Plantear acciones de mejora en la dinámica de trabajo

Tabla 2.1: Resumen Eventos ASAP

2.1.3. Artefactos

El desarrollo del proyecto se materializa sobre dos artefactos:

- **Incremento:** este artefacto se define de forma comparable a como se hace en Scrum, puesto que representa el estado en el que se encuentra el proyecto, recogiendo hasta ese preciso instante las historias de usuario completadas. Por ende, reúne a su vez resultados ya consolidados hasta la fecha. Resaltar de igual manera que dichas tareas a tratar han de figurar en los distintos entornos de trabajo para así que se forje constancia del avance del proyecto.
- **Retroalimentación:** este elemento abarca el *feedback* proporcionado tanto por el tutor en el sprint, como en la comunicación de progresos por la comunidad. Su objetivo radica principalmente en ofrecer una serie de aspectos a mejorar, realzar o profundizar para así asegurar el cumplimiento de los objetivos delimitados en el alcance del proyecto.

2.1.4. Entorno de Trabajo

La pujante dinámica de trabajo en ASAP requiere del despliegue de distintos entornos tecnológicos que faciliten el avance, supervisión, consecución y planificación del proyecto, en particular:

- **Espacio de trabajo compartido:** es un espacio de comunicación privada entre el tutor y el estudiante donde se comparten los recursos y el trabajo colaborativo. A modo de repositorio, en él se almacenan los distintos artefactos producidos en cada sprint, además del objetivo del sprint, acuerdos de trabajo y cuaderno de trabajo. Contiene principalmente 4 secciones: documentación, entregas, referencias y workspace, siendo responsabilidad del alumno mantener este espacio limpio, ordenado y organizado. Resaltar por último que este sería un espacio de trabajo privado, pero existe otro público para la comunidad en el que se comentan dudas de carácter general sobre la dinámica, publicaciones de interés o cualquier otro aspecto de índole común. Todo ello se coordina gracias a la plataforma de Microsoft Teams.
- **Cuaderno de Trabajo:** Es una herramienta que permite registrar de manera sistemática todas las tareas realizadas durante un proyecto. Su objetivo es medir el progreso y plasmar todo el trabajo llevado a cabo, desde la planificación hasta la ejecución. Tal como se muestra en la figura 2.1 se organiza en columnas clave como fecha, historia de usuario (planteamiento del problema, objetivos, condicionantes, metodología, planificación, balance, interacción, entorno de negocio, estado del arte, fundamentos teóricos, fundamentos técnicos, análisis del problema, diseño de la solución, construcción del producto, documentación técnica, presentación), tarea, tiempo invertido y comentarios, facilitando tanto la organización personal como la revisión por parte de tutores o supervisores.

2.2. Interacción

Cabe destacar la existencia de la historia de Interacción, la cual se extiende a lo largo de todo el sprint y tiene como objetivo garantizar una comunicación fluida, efectiva y constante durante el desarrollo del proyecto. Esto incluye diversas reuniones periódicas con el tutor, las cuales pueden ser tanto presenciales como realizadas por Teams. En primer lugar, se lleva a

Capítulo 2. Gestión del Proyecto

Fecha	Historia	Tarea	Tiempo (minuto)	Consideraciones adicionales
26/02/2025	Planificación	Planificar Sprint	30	Planificación del Sprint
26/02/2025	Planteamiento del problema	Plantear Statement	20	Planteamiento del Problema
26/02/2025	Documentación técnica	Introducción Planteamiento	30	Aun sin finalizar
27/02/2025	Planteamiento del problema	Planificar Sprint	5	
27/02/2025	Documentación técnica	Introducción Planteamiento	20	
27/02/2025	Objetivos	Plantear Objetivos	20	
27/02/2025	Documentación técnica	Objetivos y Planteamiento	180	Documentar los objetivos y planteamiento
27/02/2025	Metodología	Inicio	10	
28/02/2025	Metodología	Reunión	15	
24/02/2025	Metodología	Reunión	60	
28/02/2025	Condicionantes	Restricciones	20	Restricciones posibles
28/02/2025	Documentación técnica	Restricciones	5	
28/02/2025	Entorno de negocio	Antecedentes	20	Esquematizar los puntos claves

Figura 2.1: Cuaderno De Trabajo

cabo una reunión de inicio la primera semana de comenzar cada sprint. Después, se establecen reuniones de sincronización todos los viernes de 13:30 a 13:45, con el propósito de evaluar el progreso semanal, resolver bloqueos y definir el plan de trabajo hasta la siguiente reunión. Estas sesiones tendrán una duración máxima de 15 minutos y se desarrollarán apoyándose en el tablero de aprendizaje. Además, el último viernes de cada sprint, se organiza una sesión de comunicación de progresos de 2 horas y 30 minutos donde se obtendrá el *feedback* de la comunidad. Respecto a la generación del *feedback* por parte de los tutores, se recibiría el último día del sprint, que resulta ser domingo.

En general, concluir que este enfoque de interacción favorece la transparencia, colaboración y facilita la retroalimentación, contribuyendo así a una mejora del proyecto.

2.3. Planificación temporal

La planificación, alcance y tiempo de este proyecto se encuentran limitados por la cantidad de créditos del TFG, 12 ECTS, y sus horas relacionadas: 300 horas. No obstante, puesto que este proyecto se está llevando de manera conjunta con la Beca de Colaboración, es necesario incrementar el número de horas a 450 al contemplarse un alcance mayor.

Para ello, se ha dividido en 4 sprints, intentando superar las 100h por sprint:

- Sprint #1: 24 febrero - 21 marzo
- Sprint #2: 24 marzo - 25 abril
- Sprint #3: 28 abril - 23 mayo
- Sprint #4: 26 mayo - 20 junio

De manera general, los objetivos a abordar son: Proyecto, Antecedentes, Desarrollo, Aceptación y Comunicación. La tabla 2.2 muestra la distribución de los paquetes de trabajo por sprint. Para cada uno de los paquetes de trabajo, se contemplan las siguientes acciones:

Paquete	Historias	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Proyecto	Planteamiento del Problema				
	Objetivos				
	Condicionantes				
	Metodología				
	Planificación				
	Balance				
	Interacción				
Antecedentes	Entorno de Negocio				
	Estado del Arte				
	Fundamentos Teóricos				
	Fundamentos Técnicos				
Desarrollo	Análisis del Problema				
	Diseño de la Solución				
	Construcción del Producto				
	Evaluación del Producto				
	Puesta en Producción				
Aceptación	Métricas de Éxito				
	Plan de Aceptación				
	Ánalisis de Resultados				
Comunicación	Documentación Técnica				
	Presentación				

Tabla 2.2: Distribución de Paquetes de Trabajo por Sprint

1. **Proyecto:** Como se indicaba anteriormente, este paquete de trabajo, se enfoca en la gestión de proyectos en Ingeniería Informática, destacando la identificación precisa del problema, los objetivos y la metodología adecuada. Implica planificar, organizar y gestionar recursos humanos, financieros y temporales, así como la comunicación entre los participantes. Este paquete de trabajo se descomponen en las siguientes 7 tareas.

- *Planteamiento del Problema:* construir el *problem statement* del proyecto para asegurar la identificación y comunicación precisa del problema a resolver.
- *Objetivos:* establecer los objetivos del proyecto para guiar el trabajo y evaluar el éxito de sus resultados de manera efectiva.
- *Condicionantes:* establecer las condiciones que pueden afectar al desarrollo del proyecto en términos de negocio y técnicos.
- *Metodología:* seleccionar y aplicar la metodología adecuada para asegurar un enfoque estructurado y sistemático en el desarrollo del proyecto.

- *Planificación*: desarrollar un plan detallado de tareas y actividades que optimice el uso eficiente de los recursos disponibles en el proyecto y cumplir con los plazos establecidos.
 - *Balance*: hacer el balance entre los recursos utilizados y los resultados obtenidos, analizando las desviaciones que se han producido respecto a la planificación original, para asegurar la viabilidad, la eficiencia y el aprendizaje continuo.
 - *Interacción*: mantener una interacción efectiva y continua durante todo el proyecto para asegurar la transparencia, la colaboración y la mejora continua.
2. **Antecedentes**: Se centra en la recopilación y análisis de información relevante para contextualizar el proyecto, incluyendo el entorno de negocio, el estado del arte y los fundamentos teóricos y técnicos. A través de estos antecedentes, se identifican necesidades, oportunidades y desafíos, proporcionando un conocimiento sólido del dominio y los aspectos técnicos que sustentan el desarrollo del TFG. Este paquete de trabajo se descomponen en las siguientes tareas.
- *Entorno de Negocio*: conocer el entorno de negocio en el que se desarrolla el proyecto para construir un producto que se alinee con las necesidades y expectativas de sus stakeholders.
 - *Estado del Arte*: completar una revisión exhaustiva del estado del arte relacionado con el proyecto para identificar soluciones y enfoques existentes y, en base a ello, determinar las oportunidades de mejora y diferenciación del producto a desarrollar en el TFG.
 - *Fundamentos Teóricos*: comprender los fundamentos teóricos del proyecto para poder elegir las teorías o principios más adecuados para abordar el diseño de la solución.
 - *Fundamentos Técnicos*: comprender los fundamentos técnicos del proyecto para poder elegir objetivamente las herramientas necesarias para construir el producto.
3. **Desarrollo**: Este paquete se enfoca en las competencias necesarias para llevar a cabo el proceso de desarrollo del proyecto, desde el análisis del problema hasta la implementación y prueba del producto final. A diferencia de otros paquetes de trabajo, este se centra en las características específicas del producto técnico a desarrollar en el TFG. Este paquete de trabajo se descomponen en las siguientes tareas.
- *Análisis del Problema*: analizar el problema para determinar los diferentes tipos de requisitos (y los stakeholders relacionados) que describen el producto a construir.
 - *Diseño de la Solución*: establecer el diseño de la solución para poder construir un producto que alcance los objetivos del proyecto.
 - *Construcción del Producto*: construir un producto acorde al diseño planteado para alcanzar los objetivos del proyecto.
 - *Evaluación del Producto*: probar el producto para asegurar que cumple con todos sus requisitos.
 - *Puesta en Producción*: caracterizar el proceso de puesta en producción para que los stakeholders puedan utilizar el producto construido de acuerdo con sus necesidades.

4. **Aceptación:** El paquete “Aceptación” se centra en diseñar y ejecutar un plan para evaluar el cumplimiento de los objetivos del proyecto. Se requiere un análisis detallado de las causas y un plan de acción para corregirlo. Incluye la definición de métricas, umbrales de éxito y un plan para la evaluación y análisis de los resultados. Este paquete de trabajo se descomponen en las siguientes tareas.

- *Métricas de Éxito:* definir las métricas de éxito para evaluar el cumplimiento de los objetivos del proyecto.
- *Plan de Aceptación:* diseñar el plan de aceptación para asegurar una evaluación estructurada y rigurosa del producto.
- *Análisis de Resultados:* seleccionar y aplicar la metodología adecuada para asegurar un enfoque estructurado y sistemático en el desarrollo del proyecto.

5. **Comunicación:** Este paquete final se enfoca en la transmisión clara y estructurada del trabajo realizado y sus resultados, para garantizar su comprensión y valoración por diversas audiencias. Incluye la elaboración de una memoria técnica detallada y organizada, así como una presentación oral en la que se debe defender el trabajo y responder a preguntas de forma precisa. Este paquete de trabajo se descomponen en las siguientes tareas.

- *Documentación Técnica:* asegurar que la memoria técnica sea completa, original, clara y bien estructurada, con contenidos precisos y organizados lógicamente. Asegurar también la correcta integración de elementos visuales, el uso adecuado de citas y el cumplimiento de requisitos de estilo y legibilidad.
- *Presentación:* asegurar que el contenido presentado ofrece una visión completa, precisa y consistente del trabajo realizado, además de una correcta calidad de la exposición.

A continuación se presenta una planificación detallada de cada uno de los sprints, en la que se reflejan las acciones necesarias para alcanzar el objetivo de cada uno. Asimismo, se indica la complejidad estimada de cada acción según su tamaño. La planificación del proyecto se ha realizado utilizando el Diagrama de Gantt, que permite visualizar el progreso temporal de las tareas. Las historias y tareas han sido estimadas según tamaño y dificultad como XS, S, M, L o XL, lo que ha facilitado la asignación de recursos y la distribución del trabajo a lo largo de los sprints.

2.3.1. Sprint #1

En este primer sprint se contempla la realización de las acciones enumeradas en la tabla 2.3, con sus distintos tamaños de las historias 2.2. Asimismo, su planificación temporal se aprecia en detalle a través de la figura 2.3. Este primer sprint se enfoca en investigar el marco teórico y tecnológico del proyecto, explorando el uso de IA en educación y analizando soluciones existentes al problema planteado.

ID	Historia	Objetivo	Estimación
Proyecto-1	Planteamiento del problema	Proyecto	S
Proyecto-2	Objetivos	Proyecto	M
Proyecto-3	Condicionantes	Proyecto	M
Proyecto-4	Metodología	Proyecto	L
Proyecto-5	Planificación	Proyecto	M
Proyecto-6	Balance	Proyecto	XS
Proyecto-7	Interacción	Proyecto	S
Antecedentes-1	Entorno de negocio	Antecedentes	M
Antecedentes-2	Estado del arte	Antecedentes	XL
Antecedentes-3	Fundamentos teóricos	Antecedentes	XL
Antecedentes-4	Fundamentos técnicos	Antecedentes	M
Desarrollo-1	Análisis del problema	Desarrollo	M
Desarrollo-2	Diseño de la solución	Desarrollo	M
Desarrollo-3	Construcción de producto	Desarrollo	L
Comunicación-1	Documentación técnica	Comunicación	XL
Comunicación-2	Presentación	Comunicación	M

Tabla 2.3: Tabla Planificación Sprint 1

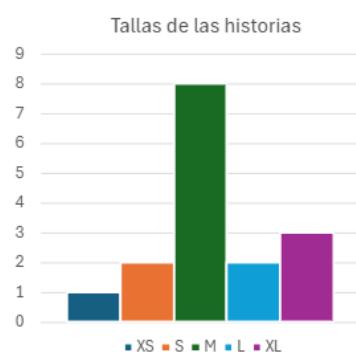


Figura 2.2: Estimación Sprint 1

2.3. Planificación temporal

	Total Horas	Semana #1				Semana #2				Semana #3				Semana #4															
Proyecto	0,00	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Planteamiento del problema	0,00																												
Objetivos	0,00																												
Condicionantes	0,00																												
Metodología	0,00																												
Planificación	0,00																												
Balance	0,00																												
Interacción	0,00																												
Antecedentes	0,00	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Entorno de negocio	0,00																												
Estado del arte	0,00																												
Fundamentos teóricos	0,00																												
Fundamentos técnicos	0,00																												
Desarrollo	0,00	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Análisis del problema	0,00																												
Diseño de la solución	0,00																												
Construcción de producto	0,00																												
Comunicación	0,00	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Documentación técnica	0,00																												
Presentación	0,00																												
	0,00																												

Figura 2.3: Cronograma Sprint 1

2.3.2. Sprint #2

Para el segundo sprint, se realizan historias y tareas similares al anterior, aunque se incluye mayoritariamente la parte del desarrollo y aceptación del producto final. Por tanto, este sprint se centrará más en la parte de construcción del proyecto, así como su evaluación. Los puntos tratados se encuentran en la siguiente tabla 2.4 con sus distinta estimación en la figura 2.4. En ambas se puede apreciar que la carga es mayor al anterior porque se aborda más recorrido de cada historia, así como las correcciones del incremento anterior y nuevas tareas del proyecto. Finalmente, resaltar que en los primeros días se tratará lo más relevante al sprint anterior debido a las correcciones propuestas, en la parte central lo verdaderamente significativo en este sprint, y al final, lo más relativo a la metodología, balance y primera toma de contacto del paquete aceptación. Todo ello se puede observar en el cronograma de 2.5.

Por último, es importante comentar que para las acciones que se repiten en sprints siguientes (o como es este sprint), resalta que su complejidad se va reduciendo paulatinamente, hasta que se finalizan después de incorporar todo el *feedback* del tutor.

ID	Historia	Objetivo	Estimación
Proyecto-1	Planteamiento del problema	Proyecto	S
Proyecto-2	Objetivos	Proyecto	S
Proyecto-3	Condicionantes	Proyecto	XS
Proyecto-4	Metodología	Proyecto	M
Proyecto-5	Planificación	Proyecto	M
Proyecto-6	Balance	Proyecto	XS
Proyecto-7	Interacción	Proyecto	S
Antecedentes-1	Entorno de negocio	Antecedentes	M
Antecedentes-2	Estado del arte	Antecedentes	XL
Antecedentes-3	Fundamentos teóricos	Antecedentes	XL
Antecedentes-4	Fundamentos técnicos	Antecedentes	XL
Desarrollo-1	Análisis del problema	Desarrollo	M
Desarrollo-2	Diseño de la solución	Desarrollo	M
Desarrollo-3	Construcción de producto	Desarrollo	M
Desarrollo-4	Evaluación del producto	Aceptación	L
Aceptación-1	Métricas de éxito	Aceptación	S
Aceptación-2	Plan de aceptación	Aceptación	S
Comunicación-1	Documentación técnica	Comunicación	XL

Tabla 2.4: Tabla Planificación Sprint 2

2.3. Planificación temporal

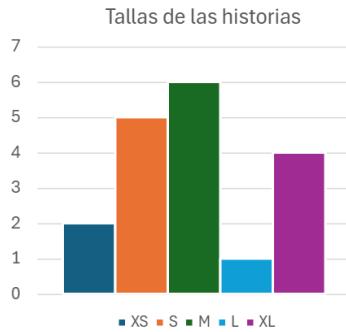


Figura 2.4: Estimación Sprint 2

Proyecto	Total Horas	Semana #1				Semana #2				Semana #3				Semana #4				Semana #5																									
		24/02	25/02	26/02	27/02	28/02	29/02	30/02	31/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03	24/03	25/03									
Planteamiento del problema	0,00																																										
Objetivos	0,00																																										
Condicionantes	0,00																																										
Metodología	0,00																																										
Planificación	0,00	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████						
Balance	0,00																																										
Interacción	0,00																																										
Antecedentes	0,00	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████				
Entorno de negocio	0,00																																										
Estado del arte	0,00																																										
Fundamentos teóricos	0,00																																										
Fundamentos técnicos	0,00																																										
Desarrollo	0,00	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████			
Análisis del problema	0,00																																										
Diseño de la solución	0,00																																										
Construcción del producto	0,00																																										
Evaluación del producto	0,00																																										
Aceptación	0,00	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████			
Métricas de éxito	0,00																																										
Plan de aceptación	0,00																																										
Comunicación	0,00	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████		
Documentación técnica	0,00																																										

Figura 2.5: Cronograma Sprint 2

2.3.3. Sprint #3

En relación con el tercer sprint, se abordan casi todas las historias que conforman el producto de aprendizaje, tal como se detalla en la tabla correspondiente 2.5, (a excepción de la puesta en producción y conclusiones finales). Cabe destacar que se profundiza especialmente en la integración de las historias de aprendizaje dentro del desarrollo del producto, así como en la sección de antecedentes, con el objetivo de completarlas. Esto implica una estimación de esfuerzo mayor, como se refleja en la figura de estimación 2.7.

Es importante señalar que la carga de trabajo es considerada, ya que se tratarán todos los aspectos del proyecto, incluidos la historia de Presentación. No obstante, aumentarla, aunque de manera parcial, no supondrá un problema, ya que durante este sprint cuento con algo más de tiempo. Por otra parte, en el cronograma 2.6, se aprecia la planificación inicial. Al estar más centrado en la etapa de construcción, se asigna mayor peso temporal a estas tareas. La primera parte del plazo se dedicará principalmente a corregir elementos identificados en las retrospectivas anteriores, asegurando así una base sólida para avanzar en esta iteración. Además, se destina tiempo a fortalecer los fundamentos, así como a actividades nuevas como la evaluación y el análisis de resultados.

Total Horas	Semana #1				Semana #2				Semana #3				Semana #4														
	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05	
Proyecto	0,00																										
Planteamiento del problema	0,00																										
Objetivos	0,00																										
Planificación	0,00																										
Balance	0,00																										
Interacción	0,00																										
Antecedentes	0,00	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05
Entorno de negocio	0,00																										
Estado del arte	0,00																										
Fundamentos teóricos	0,00																										
Fundamentos técnicos	0,00																										
Desarrollo	0,00	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05
Análisis del problema	0,00																										
Diseño de la solución	0,00																										
Construcción del producto	0,00																										
Evaluación del producto	0,00																										
Aceptación	0,00	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05
Métricas de éxito	0,00																										
Plan de aceptación	0,00																										
Ánalisis de resultados	0,00																										
Comunicación	0,00	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05
Documentación técnica	0,00																										
Presentación	0,00																										

Figura 2.6: Cronograma Sprint 3

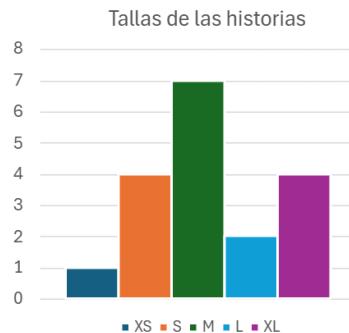


Figura 2.7: Estimación Sprint 3

ID	Historia	Objetivo	Estimación
Proyecto-1	Planteamiento del problema	Proyecto	S
Proyecto-2	Objetivos	Proyecto	S
Proyecto-3	Condicionantes	Proyecto	S
Proyecto-4	Metodología	Proyecto	S
Proyecto-5	Planificación	Proyecto	S
Proyecto-6	Balance	Proyecto	XS
Proyecto-7	Interacción	Proyecto	S
Antecedentes-1	Entorno de negocio	Antecedentes	L
Antecedentes-2	Estado del arte	Antecedentes	L
Antecedentes-3	Fundamentos teóricos	Antecedentes	XL
Antecedentes-4	Fundamentos técnicos	Antecedentes	XL
Desarrollo-1	Análisis del problema	Desarrollo	S
Desarrollo-2	Diseño de la solución	Desarrollo	S
Desarrollo-3	Construcción de producto	Desarrollo	XL
Desarrollo-4	Evaluación del producto	Aceptación	M
Aceptación-1	Métricas de éxito	Aceptación	M
Aceptación-2	Plan de aceptación	Aceptación	M
Aceptación-3	Análisis de resultados	Aceptación	M
Comunicación-1	Documentación técnica	Comunicación	XL
Comunicación-2	Presentación	Comunicación	M

Tabla 2.5: Tabla Planificación Sprint 3

2.3.4. Sprint #4

Durante este último sprint se abordan aquellas historias que no fueron finalizadas en el sprint anterior o que requieren una última iteración para su consolidación, tales como planificación, iteración o balance, entre otras. Todas ellas se detallan en la tabla correspondiente 2.6. A simple vista, se puede observar que las estimaciones asociadas a secciones como antecedentes o planificación son considerablemente menores en comparación con iteraciones previas. Esto se debe a que dichos apartados ya fueron profundizados anteriormente.

Por otra parte, en esta fase se requiere de un esfuerzo significativo en tareas relacionadas con el desarrollo y la aceptación del producto, las cuales implican actividades como terminar el desarrollo de manera precisa, realizar ajustes, y refinar aspectos clave del proceso de aceptación. Por tanto, el enfoque de este sprint es consolidar primero las partes menos laboriosas y también finalizar el desarrollo en las primeras semanas, para posteriormente centrarse en la evaluación y los últimos detalles necesarios. Todo ello se refleja en el cronograma 2.9, donde se observa cómo esta etapa se dedica principalmente a concluir el trabajo iniciado y asegurar la calidad del resultado final.

Respecto la carga de trabajo en relación a otros sprints, se aprecia que hay objetivos clave a trabajar, además de diversos puntos más. Aun así, es parecida y asequible considerando que es el último sprint. Esto se refleja en la figura 2.8. Por tanto, este sprint se va a basar en finalizar todo, tanto memoria (que supondrá bastante dedicación), desarrollo, aceptación y conclusiones importantes para el proyecto.

En resumen, este sprint está orientado a cerrar todos los aspectos del proyecto: finalizar el desarrollo, completar la documentación y llevar a cabo las evaluaciones y conclusiones finales, fundamentales para una adecuada calidad del producto.

ID	Historia	Objetivo	Estimación
Proyecto-5	Planificación	Proyecto	S
Proyecto-6	Balance	Proyecto	S
Proyecto-7	Interacción	Proyecto	S
Antecedentes-3	Fundamentos teóricos	Antecedentes	M
Desarrollo-1	Análisis del problema	Desarrollo	M
Desarrollo-2	Diseño de la solución	Desarrollo	M
Desarrollo-3	Construcción de producto	Desarrollo	L
Desarrollo-4	Evaluación del producto	Aceptación	XL
Desarrollo-5	Puesta en producción	Aceptación	XL
Aceptación-1	Métricas de éxito	Aceptación	M
Aceptación-2	Plan de aceptación	Aceptación	L
Aceptación-3	Ánalisis de resultados	Aceptación	XL
Comunicación-1	Documentación técnica	Comunicación	XL
Comunicación-2	Presentación	Comunicación	L

Tabla 2.6: Tabla Planificación Sprint 4

2.3. Planificación temporal

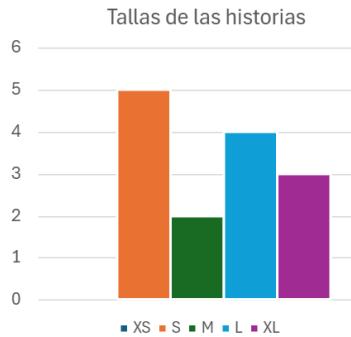


Figura 2.8: Estimación Sprint 4

Proyecto	Total Horas	Semana #1				Semana #2				Semana #3				Semana #4															
		26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Planteamiento del problema	0,00																												
Objetivos	0,00																												
Condicionantes	0,00																												
Metodología	0,00																												
Planificación	0,00																												
Balance	0,00																												
Interacción	0,00																												
Antecedentes	0,00	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Entorno de negocio	0,00																												
Estado del arte	0,00																												
Fundamentos teóricos	0,00																												
Fundamentos técnicos	0,00																												
Desarrollo	0,00	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Análisis del problema	0,00																												
Diseño de la solución	0,00																												
Construcción del producto	0,00																												
Evaluación del producto	0,00																												
Puesta en producción	0,00																												
Aceptación	0,00	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Métricas de éxito	0,00																												
Plan de aceptación	0,00																												
Ánalysis de resultados	0,00																												
Comunicación	0,00	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Documentación técnica	0,00																												
Presentación	0,00																												

Figura 2.9: Cronograma Sprint 4

2.4. Presupuestos

Tras realizar la planificación temporal del proyecto, resulta necesario concretar la estimación del presupuesto que supondrá el proyecto para lograr un mayor control en cuanto a la gestión económica, así como una mejor toma de decisiones. El coste total estimado será la suma de los presupuestos destinadas a lo recursos *hardware*, *software* y recursos humanos. A continuación, se detalla cada uno de ellos.

2.4.1. Recursos *Hardware*

Dentro de los recursos *hardware*, además del coste de adquisición de los distintos equipos y dispositivos físicos empleados, será preciso considerar su vida útil, el porcentaje real de uso de cada uno para el proyecto, los posibles gastos asociados a su mantenimiento y la duración de su utilización. Principalmente para llevar a cabo el proyecto se ha hecho uso de un ordenador personal, en concreto de un ASUS VivoBook 14. Este dispositivo de gama media-alta posee un procesador Intel Core i5-1135G7 de 11^a generación, con una frecuencia base de 2,40 GHz y turbo de hasta 2,42 GHz; una memoria RAM de 16,0 GB (15,7 GB utilizables); un almacenamiento principal correspondiente a un disco SSD NVMe de 477 GB (Micron-2210-MTFDHBA512QFD); tarjeta gráfica Intel Iris Xe Graphics integrada, con 128 MB de memoria reservada y finalmente, un sistema operativo de Windows de 64 bits compatible con su arquitectura x64. En su momento costó 499€, con una vida útil aproximada de 5 años, equivalente a 43800 horas:

$$5 \text{ años} \times 365 \frac{\text{días}}{\text{año}} \times 24 \frac{\text{horas}}{\text{día}} = 5 \times 365 \times 24 \text{ horas} = 5 \times 8,760 \text{ horas} = 43800 \text{ horas}$$

Sin embargo, tal y como se estima en la planificación, la duración del TFG equivale a 450 horas, lo que implica el 1,03% tiempo de uso del ordenador respecto al total. En cuestiones pecuniarias, resulta en 5,14€:

$$1,03\% \text{ de } 499 \text{ €} = 499 \times \frac{1,03}{100} = 499 \times 0,0103 = 5,14 \text{ €}$$

Por otra parte, es imprescindible contar con una conexión a Internet para poder realizar la labor investigadora y de documentación del proyecto, como la búsqueda de múltiples artículos, libros o páginas, aunque empleándola también para distintas herramientas en línea como Microsoft Office (Teams, Excel...), Overleaf, Trello y especialmente para la API de OpenAI.

Por lo tanto, durante los 5 meses de duración del proyecto, se hará uso de una tarifa de 450MB de fibra óptica simétrica con un coste de 25€/mes. No obstante, la conexión se comparte con otra persona al convivir en la misma vivienda, lo que sería el 50% correspondiente. Asimismo, su uso total no será destinado al TFG, si no que el porcentaje de utilización relacionado se estima en el 60%. Con ello, el coste asignado al mes sería de 7,5 €.

$$\begin{aligned} 25 \times 0,5 &= 12,5 \text{ €} \\ 12,5 \times 0,60 &= 7,5 \text{ €} \end{aligned}$$

Como el proyecto perdura 5 meses, sería 37,5 €:

$$7,5 \times 5 = 37,5 \text{ €}$$

Respecto al resto de gastos, como la electricidad consumida para cargar el ordenador, no es posible conocer su porcentaje de uso con exactitud, por lo que quedan excluidos en los presupuestos.

Finalmente, el total sería 42,94€ ($37,5 + 5,14 = 42,94$ €), cuyo desglose se detalla en la tabla 2.7:

Dispositivo	Coste	Vida Útil	Tiempo de Uso	Porcentaje de Uso	Coste Estimado
Ordenador Personal	499	5 años 43.800 horas	5 meses 450 horas	1.03 %	5,14 €
Wifi	11,5	5 meses	5 meses	60 %	37,5 €
TOTAL					42,94 €

Tabla 2.7: Presupuestos *Hardware*

2.4.2. Recursos *Software*

Los componentes *software* con los que operar son, en su mayoría, programas o plataformas en línea y gratuitas, o bien recursos que cuentan con licencias de acceso libre. Se presentan las siguientes herramientas para el desarrollo del proyecto (incluidas las previamente mencionadas):

- Anaconda: distribución *software* de código abierto que contiene diversas librerías, paquete y entornos en Python. Entre ellos se integra el *framework* LangChain, junto con herramientas como Chroma, todas ellas de uso libre.
- Hugging Face: plataforma que contiene diversos modelos IA. En este caso, se recurre a la versión de libre acceso para la realización de los *embeddings*.
- Streamlit: *framework* de código abierto empleado para el despliegue del sistema mediante una interfaz web local. Permite ejecutar la aplicación de forma interactiva, y es de uso completamente gratuito.
- Trello: a efectos de tablero de proyecto.
- Visual Studio Code: como editor de código.
- API OpenAI: para acceder a los distintos modelos de OpenAI e integrarlos en el desarrollo de respuestas generadas mediante el LLM seleccionado. Además, se utiliza para la evaluación del sistema mediante RAGAS, que puede requerir llamadas adicionales a modelos de lenguaje alojados en OpenAI.
- Microsoft Office: especialmente Teams y Excel, que facilitan la comunicación y el despliegue del entorno de trabajo. Cabe destacar que entra dentro de las licencias académicas gratuitas de la UVa, por lo que sería un coste cero.
- Overleaf: para la redacción de la memoria en L^AT_EX.

- StarUML: necesario para la realización de distintos diagramas. Asimismo, también entraría dentro de las licencias académicas.
- GitHub: plataforma basada en control de versiones que también se incluye como licencia.
- Windows 10: sistema operativo empleado, facilitado mediante las licencias académicas gratuitas de la UVa.

Todas ellas son gratuitas, a excepción de la API y Overleaf. Con el descuento de la licencia académica UVa la licencia de Overleaf es de 60€/año, lo que equivale a 5€/mes. Como la duración del TFG es de 5 meses el total sería 25€. Para la API de OpenAI se asignó un crédito de 50€ como presupuesto máximo para el desarrollo del proyecto. Aunque se prevé un gasto menor, ese sería el presupuesto total para la API y, por tanto, el coste de los recursos *software* asciende a 75€. En resumen, se presenta la tabla simplificada:

Herramienta	Precio Licencia (€)	Coste Estimado (€)
Anaconda	0	0
API OpenAI	Pago por uso	50
GitHub	0 (Versión Gratuita)	0
Hugging Face	0 (Versión Gratuita)	0
Microsoft Office	85	0
Overleaf	60	25
StarUML	80	0
Streamlit	0	0
Trello	0 (Versión Gratuita)	0
Visual Studio Code	0	0
Windows 10	20	0
TOTAL		75 €

Tabla 2.8: Presupuesto Recursos *Software*

2.4.3. Recursos Humanos

Finalmente, se presenta el presupuesto de recursos humanos. Aunque el proyecto se ha realizado de manera individual por una única persona, se ha de contemplar los múltiples roles o distintos perfiles profesionales (*seniors*) que he abarcado, y que serían requeridos para efectuar un proyecto de estas características. A continuación, se enumeran los roles necesarios, junto con sus funciones:

1. **Gestor de Proyecto (Project Manager)**: responsable de definir el alcance del proyecto, elaborar los presupuestos, distribuir recursos, realizar la planificación, gestionar riesgos, coordinar tareas, definir cronogramas, supervisar el estado del proyecto, resolver sucesos y liderar al equipo.

2. **Analista de Requisitos:** vital para comprender las necesidades de los usuarios y los distintos requisitos del proyecto, y traducirlos en especificaciones técnicas claras y viables.
3. **Ingeniero de Datos (Data Engineer):** dentro del flujo de datos ETL se centra en la recopilación, limpieza, preparación, vectorización y almacenamiento en la base de datos.
4. **Analista de Datos (Data Analyst):** encargado de limpieza y análisis de datos y de resultados, además de la elaboración de visualizaciones de esos mismos resultados obtenidos del LLM mediante dashboards.
5. **Científico de Datos (Data Scientist):** responsable de seleccionar e integrar modelos LLMs, configurar parámetros como temperatura o top-p, diseñar estrategias de prompting y evaluar la calidad de las respuestas generadas (por ejemplo, con RAGAS).
6. **Desarrollador de Software / Web:** implementa la aplicación completa, integrando la lógica de negocio y la despliega mediante interfaces interactivas y dinámicas con Streamlit.
7. **Probador (Tester):** Realiza pruebas de funcionalidad y usabilidad, evaluando la calidad y el rendimiento del sistema.

Cabe destacar que además de las responsabilidades específicas indicadas, cada rol será responsable de documentar sus respectivas partes del proyecto.

Las horas previstas para el desarrollo del proyecto son 450, por lo que en la tabla se estiman las distintas horas que dedicaría cada profesional, así como su coste total. Para ello, me he basado en el salario anual bruto [262, 120], lo que equivale a 1800 horas anuales. La suma de los distintos salarios en función de las horas dedicadas al proyecto asciende a 10.152,95€: 2.9

Rol	Horas Estimadas	Salario Bruto Anual (€)	Salario Hora Bruto (€/h)	Coste Total Estimado (€)
Gestor de Proyecto	90	45.000	25	2.250,00
Analista de Requisitos	45	35.000	19,44	874,80
Ingeniero de Datos	80	45.000	25,00	2.000,00
Analista de Datos	60	40.000	22,22	1.333,20
Científico de Datos	70	50.000	27,78	1.944,60
Desarrollador	90	30.000	16,67	1.500,30
Probador	15	30.000	16,67	250,05
TOTAL	450 h	-	-	10.152,95 €

Tabla 2.9: Presupuesto *Recursos Humanos*

No obstante, el resultado de 10.152,95€ no es el coste final, ya que es preciso añadir también el coste que supone dar de alta en la Seguridad Social a los trabajadores. Esta cuota oscila entre el 28,3 % y 30,3 % [228] del salario bruto, por lo que se supone un 29,3 %. Luego el coste final asociado a los recursos humanos sería 13.127,76€:

$$\text{Coste Final Recursos Humanos} = 10.152,95 \text{ €} + 10.152,95 \text{ €} \times 0,293 = 13.127,76 \text{ €}$$

2.4.4. Costes Totales

La suma de los diversos presupuestos para cada recurso será el coste total: 13.245,70€, el cual se indica también en la tabla:

Presupuesto	Coste Estimado Total (€)
<i>Hardware</i>	42,94
<i>Software</i>	75,00
Humanos	13.127,76
TOTAL	13.245,70 €

Tabla 2.10: Resumen de Presupuesto

2.5. Gestión de Riesgos

En un proyecto pueden subsistir distintos eventos inciertos que pueden afectar al desarrollo del mismo, tanto de manera positiva (oportunidades) como negativa (amenazas). Por tanto, es importante tenerlos en cuenta para poder lograr los objetivos. Según el PMBOK [250], la gestión de riesgos consiste en identificar los riesgos, analizar su probabilidad de ocurrencia e impacto mediante un análisis cualitativo, realizar un análisis cuantitativo, y desarrollar planes de contingencia para mitigar amenazas. Además, implica un seguimiento y control continuo, permitiendo reducir la incertidumbre y aumentar la probabilidad de éxito del proyecto, garantizando una ejecución más controlada y eficiente.

2.5.1. Identificación de Factores de Riesgo

De manera inicial, en la tabla 2.11 se recoge el listado de riesgos:

ID	Riesgo
R1	Problemas con la integración de los modelos de IA (LLMs)
R2	Baja precisión en las respuestas generadas por el asistente
R3	Retrasos en la implementación del asistente debido a problemas técnicos o falta de recursos
R4	Inadecuada personalización del asistente para cada estudiante
R5	Problemas con la integración de datos reales de la asignatura
R6	El proyecto se retrasa respecto a la planificación inicial
R7	Poca experiencia con IA, Machine Learning (ML), RAG, OpenAI API y Streamlit para la interfaz
R8	Dificultades inherentes a la naturaleza investigativa del proyecto

Tabla 2.11: Listado de Riesgos

2.5.2. Estimación de Riesgos

Seguidamente, es necesario estimar el valor o probabilidad de ocurrencia de cada riesgo, así como su impacto en caso de suceder dentro del proyecto. Para ello, se establecen dos tipos de métricas:

- Probabilidad de ocurrencia: sobre 100 %.
- Valor: valor de ocurrencia que hace referencia a una escala del 1 al 5 normalizada de la probabilidad.
- Impacto: valorado en una escala del 1 al 5 donde:
 1. Impacto muy bajo
 2. Impacto bajo

3. Impacto medio
4. Impacto alto
5. Impacto crítico

Con base en estos criterios, se elabora la siguiente tabla de riesgos, que incluye una justificación para cada caso: Tabla de Riesgos, Ocurrencia y Justificación 2.13 y la de su impacto 2.14:

2.5.3. Matriz Probabilidad - Impacto

A continuación, es necesario realizar el producto Valor Probabilidad×Valor Impacto = Riesgo Total para poder clasificar los riesgos en función de su prioridad:

- Riesgo Total <5 = Baja
- 6 <Riesgo Total <10 = Media
- Riesgo Total >10 = Alta

Por tanto, a continuación se muestra la matriz de probabilidad - impacto 2.12. Como se puede apreciar, en ella se muestra que los riesgos **R2**, **R3**, **R4**, **R7**, son los que necesitan más atención además de acciones preventivas.

ID	Impacto	Ocurrencia	Riesgo Total	Prioridad de Riesgo
R1	5	2	10	Media
R2	4	4	16	Alta
R3	4	3	12	Alta
R4	3	4	12	Alta
R5	4	2	8	Media
R6	2	2	4	Baja
R7	3	5	15	Alta
R8	3	3	9	Media

Tabla 2.12: Riesgos priorizados según impacto y ocurrencia

ID	% Ocurrencia	Valor Cualitativo	Valor	Justificación
R1	40 %	Alto	2	Los modelos de LLM de IA deben integrarse correctamente con el RAG
R2	85 %	Medio	4	Es probable que el modelo o prompt necesite ajustes y mejoras continuas para mantener la precisión
R3	50 %	Alto	3	El desarrollo del sistema puede verse afectado por problemas técnicos o falta de recursos
R4	70 %	Medio	4	El nivel de personalización dependerá de cómo se implementen los módulos e historial de adaptación
R5	25 %	Alto	2	Utilizar datos reales podría presentar desafíos en cuanto a la calidad, formato, estructura y accesibilidad de los mismos
R6	20 %	Medio	2	Los plazos establecidos pueden no cumplirse debido a complicaciones imprevistas durante el desarrollo
R7	90 %	Alto	5	La falta de experiencia con el uso de los <i>frameworks</i> , tecnologías y herramientas específicas (OpenAI API, Streamlit y RAG), así como las técnicas de IA, ML y RAG, puede generar retrasos importantes, particularmente durante su implementación y adaptación a las necesidades del proyecto
R8	45 %	Medio	3	Al ser un proyecto investigativo, los resultados pueden no ser tan predecibles, lo que podría generar incertidumbre

Tabla 2.13: Tabla de Riesgos, Ocurrencia y Justificación

ID	Impacto Estimado	Valor	Justificación
R1	Crítico	5	Una fallo en la integración con el LLM impide la generación de respuestas a las cuestiones de los estudiantes
R2	Alto	4	Una baja precisión en las respuestas afectaría directamente la utilidad del asistente para los estudiantes
R3	Alto	4	Los retrasos por problemas técnicos llevan retrasos en la construcción del producto y viabilidad y planificación del proyecto
R4	Medio	3	Una personalización deficiente reduciría la efectividad del sistema, pero no lo haría inutilizable
R5	Alto	4	La calidad de los datos impacta el rendimiento, además de retrasar en el desarrollo
R6	Bajo	2	Los retrasos son manejables si se cuenta con márgenes adecuados o ajustes en el cronograma
R7	Medio	3	La inexperiencia puede paralizar completamente el desarrollo o generar errores críticos o una no implementación óptima de la solución. Pero, lleva a un mejor conocimiento de las herramientas
R8	Medio	3	La incertidumbre investigativa puede influir en los resultados, pero también permitir descubrimientos valiosos

Tabla 2.14: Tabla de Riesgos, Impacto y Justificación

2.5.4. Plan de Contingencia

Por último, se muestra el plan de contingencia de cada uno de los riesgos 2.15:

ID	Plan de Contingencia
R1	Leer la documentación oficial e investigar a cerca de cómo integrarla
R2	Implementar un sistema de retroalimentación para ajustar las respuestas basadas en el rendimiento de los estudiantes, además de evaluar una serie de casos de prueba y luego refinar
R3	Establecer un calendario con hitos claros y revisar semanalmente el progreso para identificar retrasos y ajustar el trabajo
R4	Realizar pruebas de usuario frecuentes para ajustar la personalización y mejorar la experiencia de aprendizaje
R5	Colaborar estrechamente con el tutor y responsable de la asignatura para garantizar la calidad y el formato adecuado de los datos
R6	Reajustar el calendario de acuerdo con los hitos alcanzados y los retrasos, con flexibilidad para las fases críticas, y al inicio del sprint en la planificación
R7	Invertir tiempo en capacitación y estudios sobre IA, ML y RAG. Dedicar tiempo inicial para formarse en los <i>frameworks</i> , tecnologías y técnicas requeridas. Documentarse en OpenAI API y Streamlit y realizar proyectos piloto para familiarizarse con las herramientas antes de la implementación final
R8	Asegurar que se realicen investigaciones paralelas y pruebas de concepto, con objetivos claros y una metodología flexible para adaptarse a los resultados

Tabla 2.15: Tabla de Plan de Contingencia para Riesgos

2.6. Balance temporal y económico

En esta última sección se llevará a cabo un análisis de las desviaciones del proyecto respecto a su planificación, tanto en términos de tiempo como de costes. Con ello, se obtiene una visión más precisa de cómo ha sido el desarrollo del proyecto y su grado de alineación con la planificación inicial.

2.6.1. Balance temporal

Anteriormente, se ha documentado la planificación mediante el uso de Diagramas de Gantt. En cuanto al balance temporal, se ha vuelto a emplear esta herramienta para reflejar el transcurso real del proyecto, complementándose con una gráfica con una representación del esfuerzo invertido realmente en cada una de las actividades planificadas. En el diagrama, los objetivos cumplidos en plazo se indican en verde, los que han excedido la planificación original en rojo, y los estimados inicialmente pero no iniciados en negro. Además, es importante resaltar la autoevaluación de las historias, que se realiza al final de cada sprint y permite analizar el rendimiento y verificar qué tan bien se ha cumplido con los objetivos establecidos.

Sprint #1

Este sprint se ha centrado en sentar las bases del proyecto a través de la investigación del estado del arte, el análisis del entorno, los antecedentes y los fundamentos teóricos. Además, se han explorado diversas técnicas y el desarrollo de RAG, así como su aplicación en el ámbito educativo.

Al examinar las gráficas 2.12 2.11 , se puede observar una carga de trabajo constante, con la mayoría de las historias completadas dentro del plazo previsto. Destacar que el esfuerzo realizado durante este sprint ha sido 103,5 horas y, tal y como se muestra la autoevaluación 2.10, completadas de una manera satisfactoria . En términos generales, ha sido un sprint de crecimiento, en el que se ha puesto especial énfasis en comprender el entorno, definir el alcance y la metodología, y comenzar con el desarrollo del proyecto.

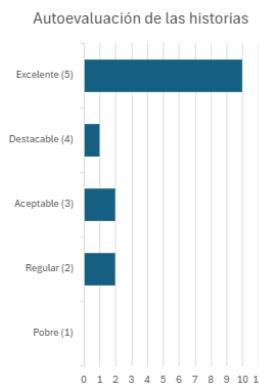


Figura 2.10: Autoevaluación Sprint 1

2.6. Balance temporal y económico



Figura 2.11: Carga de Trabajo Sprint 1

Proyecto	Total Horas	Semana #1				Semana #2				Semana #3				Semana #4															
		24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Planteamiento del problema	4,08																												
Objetivos	1,00																												
Condicionantes	0,33																												
Metodología	2,83																												
Planificación	3,67																												
Balance	0,75																												
Interacción	2,17																												
Antecedentes	22,17	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Entorno de negocio	1,33																												
Estado del arte	2,67																												
Fundamentos teóricos	6,25																												
Fundamentos técnicos	11,92																												
Desarrollo	14,67	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Análisis del problema	2,67																												
Diseño de la solución	7,33																												
Construcción de producto	4,67																												
Comunicación	51,83	24/02	25/02	26/02	27/02	28/02	01/03	02/03	03/03	04/03	05/03	06/03	07/03	08/03	09/03	10/03	11/03	12/03	13/03	14/03	15/03	16/03	17/03	18/03	19/03	20/03	21/03	22/03	23/03
Documentación técnica	46,58																												
Presentación	5,25																												
	103,50																												

Figura 2.12: Cronograma Sprint 1

Sprint #2

En este segundo sprint, el foco principal ha estado en el desarrollo del producto, así como en su evaluación inicial. También se han realizado correcciones puntuales basadas en el *feedback* recibido sobre el sprint anterior, incluyendo aspectos relacionados con los antecedentes y la metodología planificados para más adelante.

Al analizar las gráficas 2.13 y 2.14, se puede observar que durante los primeros días se intentaron seguir las correcciones propuestas, especialmente en lo referente a los elementos finales de los antecedentes y la metodología, aunque estos no llegaron a cumplirse completamente. En cuanto al desarrollo, este avanzó según lo previsto al inicio del sprint, aunque surgieron complicaciones durante la fase de construcción que afectaron al calendario previsto para la evaluación y la parte de aceptación, lo cual también queda reflejado el retraso en las gráficas. Finalmente, al surgir dichos contratiempos, no se pudo avanzar apenas en la aceptación.

A modo de opinión personal, aunque se han producido ciertos retrasos respecto al plan inicial, los objetivos principales del sprint han sido abordados de manera satisfactoria, cumpliendo con lo esencial y sentando una base sólida para continuar el desarrollo. Hubiera sido grato poder finalizar ciertas historias de usuario, pero al no poder (de ahí los valores de “pobre”) se deja como objetivo para el siguiente sprint 3. Esta autoevaluación se aprecia en la figura 2.15.

Asimismo, resaltar que la constante documentación e interacción con el tutor ha facilitado mantener una visión clara del progreso e incluir futuras mejoras.

Además, destacar que aunque se han realizado 97 horas de trabajo, algo menor a 100, sigue permitiendo mantener el plan inicial de una carga constante y equilibrada en cada sprint. En particular, en la figura 2.14, se mostraría las horas trabajadas a lo largo de estas 4-5 semanas.

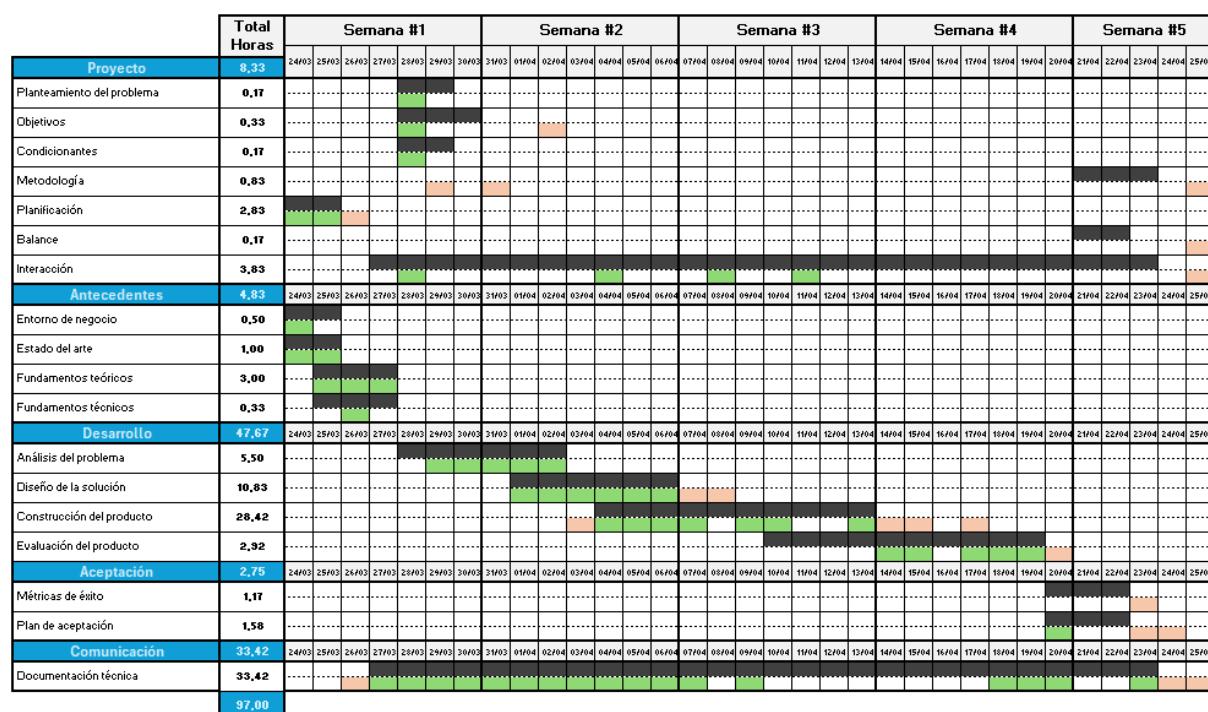


Figura 2.13: Cronograma Sprint 2



Figura 2.14: Carga Trabajo Sprint 2

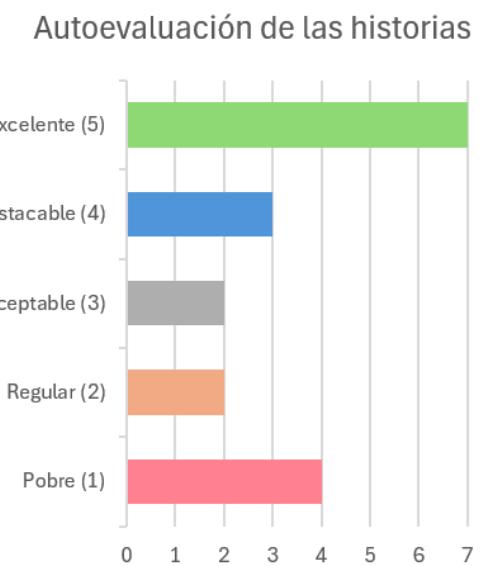


Figura 2.15: Autoevaluación Sprint 2

Sprint #3

En este tercer sprint, el núcleo ha sido el desarrollo del producto, en especial la parte de construcción. No obstante, bien es cierto que durante los primeros días me centré en realizar los ajustes necesarios debido a la retrospectiva del sprint anterior.

En general, el trabajo ha seguido de acuerdo a la planificación, aunque se puede apreciar un parón en el cronograma 2.18 en la construcción, motivada por la espera de la validación del profesor respecto a los datos que habían sido reetiquetados. Esto fue necesario porque el tutor solicitó que se incluyeran las historias de aprendizaje, y surgieron algunas dudas, ya que no era una cuestión del todo clara o evidente. Durante ese tiempo, se aprovechó para avanzar en la parte de documentación y en los antecedentes generales que tendría que abarcar los próximos días.

Finalmente, ha sido en los últimos días del sprint cuando se ha podido retomar el desarrollo del producto. Sin embargo, el avance en su construcción no ha sido tan significativo como esperaba, debido al tiempo dedicado previamente a la validación de los datos y a la documentación. Como resultado, no fue posible abordar la fase de aceptación ni la evaluación con los nuevos datos incorporados dentro del plazo previsto. Por tanto, esos puntos serán los primeros a tratar en el próximo sprint. Aun así, el adelanto del trabajo de otros apartados ha servido para casi dar por terminado esas historias del producto 2.17.

Por último, destacar el esfuerzo dedicado, que exceden de las 110 horas (111), lo cual implica una mayor constancia 2.16 y dedicación al disponer de tiempo para ello; todo con el objetivo de obtener un producto con calidad.



Figura 2.16: Carga Trabajo Sprint 3

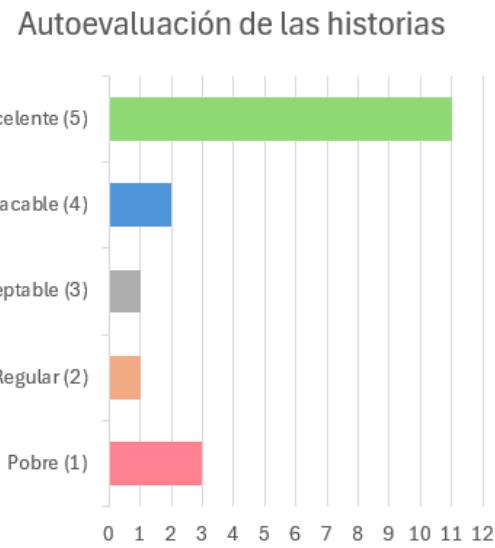


Figura 2.17: Autoevaluación Sprint 3

2.6. Balance temporal y económico

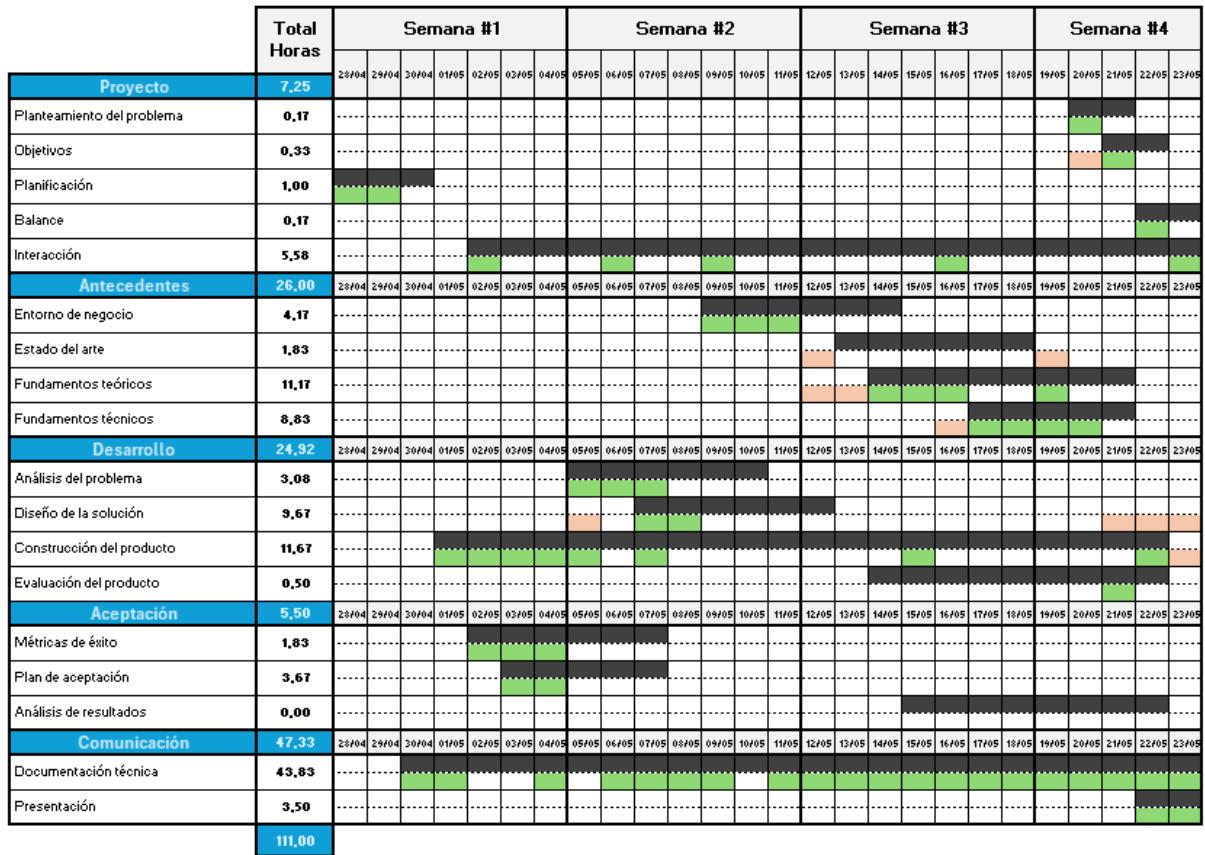


Figura 2.18: Cronograma Sprint 3

Sprint #4

En este cuarto y último sprint, se ha tratado de finalizar el proyecto de una manera exitosa. Tal y como se aprecia en el cronograma 2.21, la ejecución del incremento ha seguido mayoritariamente según lo previsto, e incluso se han finalizado algunas historias antes del tiempo asignado en la planificación del sprint. Por tanto, en la autoevaluación se considera todos los objetivos alcanzados y de manera gratificante 2.20.

No obstante, es necesario destacar que en este sprint la elaboración de la memoria ha adquirido un mayor protagonismo, ya que fue necesario revisar y corregir aspectos pendientes de sprints anteriores. En cuanto al desarrollo, gracias a la sólida base construida en fases previas, las tareas pudieron completarse de forma eficaz y en un menor tiempo del esperado. Asimismo, la fase de aceptación también requirió menos tiempo del previsto, ya que aunque se modificara la distribución de los datos; las métricas y el plan de aceptación finalmente pudieron mantenerse, lo cual simplificó el proceso. Además, la ejecución de las pruebas no implicó un esfuerzo considerable, ya que dentro del flujo de actividades incluyó ciertas funciones específicas para registrar y guardar toda la información relevante. Aunque en un primer momento los resultados obtenidos no eran del todo satisfactorios, tras la optimización del asistente se lograron mejoras significativas al trabajar con los nuevos datos, superando los resultados obtenidos en iteraciones anteriores.

Por todo ello, durante este sprint dediqué una parte sustancial del tiempo a completar y detallar la memoria del proyecto, haciendo especial hincapié en los aspectos que requerían revisión o no estaban completamente definidos.

Respecto la carga de trabajo 2.19, esta ha sido considerablemente mayor a los incrementos anteriores, alcanzando las 141,5 horas. Esto se debe a que es el último sprint y era necesario un último esfuerzo para finalizar el proyecto y con una calidad considerable.

Por último, el total de horas dedicadas al proyecto ha sido de 452 horas, que se asemejan a las 450 horas estimadas al inicio, lo que refleja una buena planificación global inicial.



Figura 2.19: Carga Trabajo Sprint 4

Autoevaluación de las historias

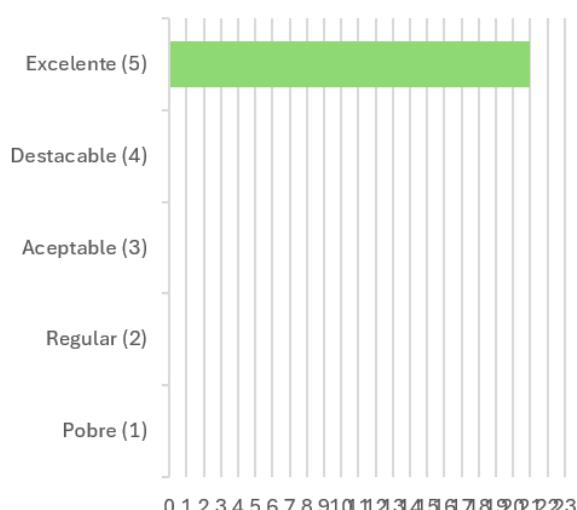


Figura 2.20: Autoevaluación Sprint 4

2.6. Balance temporal y económico

	Horas	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06	
Proyecto	10,33																													
Planteamiento del problema	0,00																													
Objetivos	0,00																													
Condicionantes	0,00																													
Metodología	0,00																													
Planificación	5,00	██████	██████	██████	██████	██████	██████	██████																						
Balance	3,33																									██████	██████	██████		
Interacción	2,00																									██████	██████	██████		
Antecedentes	2,50	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06	
Entorno de negocio	0,00																													
Estado del arte	0,00																													
Fundamentos teóricos	2,50								██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████	██████
Fundamentos técnicos	0,00																													
Desarrollo	27,17	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06	
Ánalisis del problema	0,50																													
Diseño de la solución	9,33																													
Construcción del producto	12,00																													
Evaluación del producto	3,33																													
Puesta en producción	2,00																													
Aceptación	11,76	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06	
Métricas de éxito	1,00																													
Plan de aceptación	5,57																													
Analisis de resultados	5,08																													
Comunicación	89,75	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06	
Documentación técnica	83,75																													
Presentación	6,00																													
	141,50																													

Figura 2.21: Cronograma Sprint 4

2.6.2. Balance económico

El balance económico se ha elaborado una vez finalizado el proceso, para así poder obtener mejores conclusiones al comparar de forma precisa los costes estimados frente a los reales.

Como el proyecto ha durado 2 horas más de las planeadas, habrá cambios respecto con la estimación presupuestaria inicial.

En los recursos hardware, el gasto ha sido bastante similar al anterior, aunque empleando las mismas formulas propuestas anteriormente se obtiene que el coste final sería de 42,95€, detallado en la tabla 2.16:

Dispositivo	Coste	Vida Útil	Tiempo de Uso Estimado	Tiempo de Uso Real	Porcentaje de Uso Estimado	Porcentaje de Uso Real	Coste Estimado	Coste Real
Ordenador Personal	499 €	5 años 43800 horas	5 meses 450 horas	5 meses	1,03 %	1,031 %	5,14 €	5,15 €
Wifi	11,5 €	5 meses	5 meses	5 meses	60 %	60 %	37,5 €	37,5 €
TOTAL							42,94 €	42,95 €

Tabla 2.16: Costes Recursos Hardware

Respecto a los recursos *software*, centrados en el uso de la API, el coste varía según el modelo utilizado. En este proyecto se empleó el modelo gpt-3.5-turbo, cuyo precio depende tanto del número de tokens de entrada como de salida.

A continuación, se incluyen diagramas 2.22, 2.23, 2.24, 2.25, 2.26, 2.27 y 2.28 que ilustran el uso del modelo a lo largo del proyecto, así como su gasto. Destacar que en ellos también se incluye el uso de todo lo relacionado a la API del apartado de Aceptación.

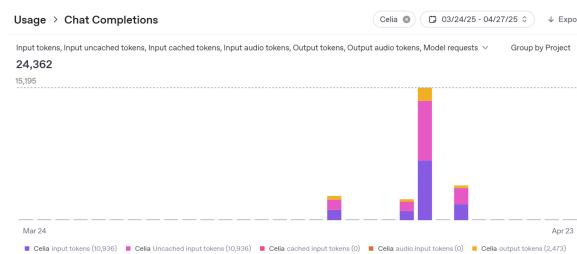


Figura 2.22: Sprint 2 Tokens API

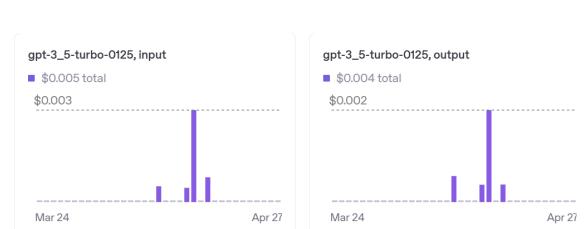


Figura 2.23: Gasto Sprint 2

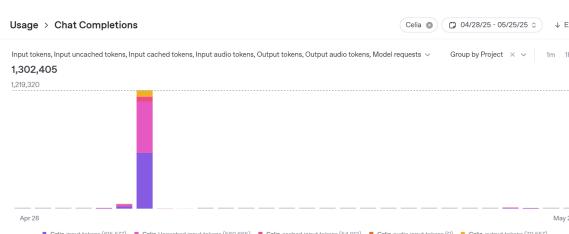


Figura 2.24: Sprint 3 Tokens API

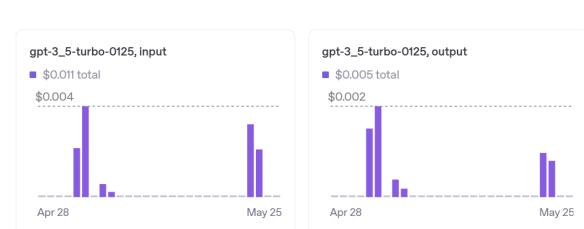


Figura 2.25: Gasto Sprint 3

2.6. Balance temporal y económico



Figura 2.26: Sprint 4 Tokens API

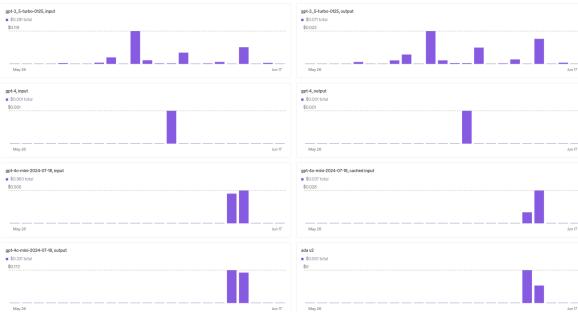


Figura 2.27: Gasto Sprint 4

June budget
\$1.68 / \$80.00

Figura 2.28: Coste API Total

En conjunto el coste final ha sido bastante inferior al inicialmente estimado, en concreto, tan solo se ha gastado 1,68 € 2.28. Esto se debe en parte a lo ya previsto durante la planificación, donde se contempló un margen amplio para el uso de la API, el cual no fue necesario consumir en su totalidad. Con ello, se permite mantener el presupuesto restante de la API para otros proyectos de eXtreme Learning. Por tanto, la tabla resumen de los costes de recursos *software* sería la siguiente:

Herramienta	Precio Licencia (€)	Coste Estimado (€)	Coste Real (€)
Anaconda	0	0	0
API OpenAI	Pago por uso	50	1,68
GitHub	0 (Versión Gratuita)	0	0
Hugging Face	0 (Versión Gratuita)	0	0
Microsoft Office	85	0	0
Overleaf	60	25	25
StarUML	80	0	0
Streamlit	0	0	0
Trello	0 (Versión Gratuita)	0	0
Visual Studio Code	0	0	0
Windows 10	20	0	0
TOTAL		75 €	26,68 €

Tabla 2.17: Costes Recursos Software

Por último, en lo relativo a los recursos humanos, se han contabilizado un total de 452 horas de trabajo, siendo de Desarrollador quien ha realizado 2 horas de más. Por tanto, esto se traduce en un coste de aproximadamente 13.171,75 €, presentado en la tabla 2.18 y la fórmula 2.1:

Rol	Horas Estimadas	Horas Reales	Salario Bruto Anual (€)	Salario Hora Bruto (€/h)	Coste Total Estimado (€)	Coste Total Real (€)
Gestor de Proyecto	90	90	45.000	25	2.250,00	2.250,00
Analista de Requisitos	45	45	35.000	19,44	874,80	874,80
Ingeniero de Datos	80	80	45.000	25,00	2.000,00	2.000,00
Analista de Datos	60	60	40.000	22,22	1.333,20	1.333,20
Científico de Datos	70	70	50.000	27,78	1.944,60	1.944,60
Desarrollador	90	92	30.000	16,67	1.500,30	1.533,64
Probador	15	15	30.000	16,67	250,05	250,05
TOTAL	450 h	452 h	-	-	10.152,95 €	10.186,29 €

Tabla 2.18: Costes *Recursos Humanos*

$$\text{Coste Final Recursos Humanos} = 10.186,29 \text{ €} + 10.186,29 \text{ €} \times 0,293 = 13.171,75 \text{ €} \quad (2.1)$$

A modo de resumen, en la siguiente tabla 2.19 se presenta el presupuesto inicial estimado y el real para cada tipo de recurso:

Presupuesto	Coste Estimado (€)	Coste Real (€)
<i>Hardware</i>	42,94	42,95
<i>Software</i>	75,00	26,68
Humanos	13.127,76	13.171,75
TOTAL	13.245,70 €	13.244,38 €

Tabla 2.19: Resumen de Costes

Como se puede apreciar, el presupuesto total real 13.244,38 €, es muy similar al estimado inicialmente, 13.245,70 €. Aunque se produjo un ligero sobrecoste en el apartado de recursos humanos, este fue compensado por un menor gasto en el área de *software*.

Capítulo 3

Antecedentes

“Hace falta personalizar el aprendizaje”. Este es el título del reportaje [70] que recoge las conclusiones del proyecto *PERSONAE*, liderado por César Coll, catedrático emérito de Psicología Evolutiva y de la Educación de la Universidad de Barcelona. En él se señala, de forma clara, la necesidad urgente de avanzar hacia un sistema educativo capaz de satisfacer las necesidades de aprendizaje reales del alumnado. Este informe, elaborado junto a un equipo de expertos en pedagogía y publicado por la Fundació Bofill, además presenta los principales indicadores sobre el estado del sistema educativo catalán y propone respuestas a los desafíos estructurales que impiden garantizar una educación de calidad y equitativa para todos [40].

La personalización del aprendizaje es una estrategia educativa que busca adaptar la enseñanza a las particularidades de cada estudiante, teniendo en cuenta sus intereses, necesidades, ritmos y estilos de aprendizaje [53]. En cambio, va más allá de modificar contenidos o metodologías [105], ya que promueve que el alumno participe activamente en la construcción de su propio proceso educativo, decidiendo objetivos, actividades y modos de evaluación [296].

Sin embargo, pese a los distintos avances dentro de este área y su presencia en discursos educativos contemporáneos [39], la realidad actual es que siguen existiendo carencias en materia de la personalización, agravándose y haciéndose especialmente visibles en aquellos estudiantes que presentan necesidades específicas de apoyo educativo o tiene diferentes ritmos de aprendizaje.

El artículo *Barreras para el Aprendizaje y la Participación en la Escuela del Alumnado con Dislexia: Voces de Familias* [114] profundiza en esta falta de individualización y además en la exclusión social que aún persisten en nuestras aulas, centrándose en este caso en estudiantes con dislexia. A través de diversos testimonios directos se muestran las consecuencias emocionales y académicas que sufren estos alumnos ante la ausencia de respuestas individualizadas en el aula. Además, recoge cómo, en muchos casos, las respuestas del sistema educativo no son solo inadecuadas, sino que conllevan dinámicas de exclusión que afectan gravemente al bienestar emocional y académico de los alumnos.

Dicha investigación subraya que recurrir a ciertas medidas estándar como la repetición de curso no resuelve las causas reales del problema. Más bien al contrario, pone de manifiesto la falta de personalización, individualización, contextualización del aprendizaje, acompañamiento real, apoyos adecuados y estrategias de atención a la diversidad. Todo ello evidencia las limitaciones del modelo educativo tradicional, que tiende a ignorar las características, necesidades y ritmos del alumnado. Por ello, es necesario una transformación profunda del sistema, una que reconozca las diferencias no como obstáculos, sino como una fuente de riqueza educativa y social.

Por tanto, hay que afrontar la enseñanza con una actitud abierta al cambio, ya que es un

factor imprescindible para lograr que los espacios educativos sea un espacio inclusivo y que confíe en las capacidades de todos sus estudiantes. Sobre todo en aquellos más vulnerables o necesarios de un apoyo extra, cuyas diferencias se convierten muchas veces en obstáculos por la falta de adaptaciones reales y además agravan su inseguridad. Es necesario un enfoque que deje atrás las etiquetas de “tonto” o “vago”, interiorizadas por tantos niños, y abrace la diversidad como un valor educativo y social.

Este artículo [114] insiste en que alcanzar una educación de calidad implica que esta sea verdaderamente inclusiva y equitativa, reconociendo el valor de la diversidad. Tal como señala Echeita [100], identificar y reducir las barreras al aprendizaje y la participación es una tarea fundamental en todos los niveles del sistema, si se quiere asegurar que ningún estudiante quede en desventaja.

En esta misma línea se encuentra, el Informe Delphi: *Personalizar, un modelo para una educación de calidad en el siglo XXI* [226], el cual reafirma la idea que la personalización del aprendizaje no es una tendencia, sino una condición indispensable para lograr una educación de calidad en el siglo XXI. En sus conclusiones, se enfatiza en la necesidad inminente de evaluación centrada en el progreso, la implicación del alumno en su propio aprendizaje, la corresponsabilidad de expertos y el uso de la tecnología como herramienta clave para el acompañamiento individual. Es decir, no basta con tener acceso a plataformas educativas, sino que estas deben permitir un seguimiento significativo, adaptado y contextualizado.

Centrándonos más en la educación superior, concretamente en el ámbito universitario que es donde se aplica este TFG, también se aprecian estos retos pendientes de personalización, adecuación y apoyo al alumno. Existen diversas investigaciones que coinciden en señalar la falta de adaptación del sistema universitario a las trayectorias individuales de los estudiantes. Por ejemplo, según una revisión publicada en la revista EDUTEC [206], aunque el interés por prácticas personalizadas mediadas por tecnologías digitales ha crecido en los últimos años, su aplicación en la universidad sigue siendo limitada debido a barreras estructurales, metodológicas y formativas que impiden un despliegue efectivo de estas estrategias.

A esta misma preocupación se suma un informe de PsicoSmart [253], el cual destaca que más del 50 % de las instituciones de educación superior no cuentan con los recursos adecuados para capacitar al profesorado en metodologías centradas en el estudiante, lo que dificulta la implementación real de modelos personalizados. En la misma línea, un análisis realizado por Virtual Educa [102] revela que más de la mitad del alumnado universitario percibe que su educación no está alineada con sus intereses personales ni con sus aspiraciones profesionales, lo que evidencia una brecha entre la oferta educativa y las necesidades reales del estudiantado. Además, expertos del eLearning Innovation Center de la UOC [168] han señalado que avanzar hacia una personalización auténtica requiere transformar la universidad en una organización más flexible, capaz de acoger la diversidad y adaptarse a las múltiples formas de aprender.

Todos estos estudios convergen en la necesidad de que el sistema educativo evolucione hacia modelos más flexibles, personalizados e inclusivos. Esto no es solo una cuestión técnica, sino también ética: se trata de ofrecer no solo igualdad de oportunidades, sino equidad en el aprendizaje, ajustándose verdaderamente a cada alumno y respetando la dignidad de sus necesidades y proceso educativo [51].

Para poder contar con esta transformación de una manera real, efectiva y eficaz, es importante señalar el uso de herramientas que permitan una atención más individualizada, continua y significativa. Aquí es donde el papel de la tecnología, y en concreto de la IA, cobra una especial relevancia. Si bien la personalización es una necesidad pedagógica de fondo, su implementación

masiva y eficaz en contextos educativos diversos requiere apoyarse en innovaciones tecnológicas que faciliten el seguimiento, la adaptación y la respuesta ante las distintas trayectorias de aprendizaje.

3.1. Tecnología Educativa

El sector de la tecnología educativa (EdTech) ha trascendido en los últimos años de una manera significativa. No solo por su expansión en diversas plataformas como Moodle, Microsoft Teams o Google Colab, hecho que se vio intensificado por una creciente digitalización en todos los niveles educativos a raíz de la pandemia del Covid-19 en el año 2020 [232]; sino también por los avances más recientes en IA, que han transformado las posibilidades de enseñanza y aprendizaje [138].

Dentro de estos avances destaca la reciente irrupción de la utilización de los grandes modelos de lenguaje (*Large Language Models: LLM*) [16] [303] y a raíz de ello, el tema principal de este TFG en cuanto a la posible incorporación del enfoque RAG en la educación. Este enfoque tiene como objetivo especificar con mayor precisión el dominio del conocimiento sobre el que se realizan preguntas o se genera contenido educativo, además de mejorar la experiencia del aprendizaje, especialmente en términos de accesibilidad, adaptabilidad y eficiencia en la entrega de contenidos relevantes [341].

La tecnología educativa se encuentra en una etapa de madurez creciente, donde no tiene como objetivo ya solo digitalizar el contenido, sino también ofrecer esas experiencias educativas individualizadas, accesibles e inclusivas que tanto ansiamos [323]. Sin embargo, pese al progreso en automatización y acceso, el EdTech muestra una deficiencia clara en cuanto a personalizar realmente el aprendizaje. Aunque muchas plataformas ofrecen cierto grado de adaptabilidad, no logran completamente una personalización real y profunda que tenga en cuenta el contexto individual del alumno, su nivel previo de conocimiento, sus intereses, capacidades o su ritmo de progreso [57]. Estas situaciones se ven intensificadas en aquellos estudiantes con necesidades educativas especiales, que requieran un refuerzo fuera del aula, una seguridad extra para estudiar en casa, o que simplemente buscan alternativas complementarias y eficaces al refuerzo escolar tradicional.

La personalización del aprendizaje no solo es una tendencia global, sino también una necesidad urgente y aún no resuelta [204]. Frente a esta realidad, la combinación de LLMs y RAG ofrece una solución de alto potencial y ventaja competitiva clara. Permitir la interacción con cada alumno al acceder al contenido educativo disponible de una manera dinámica, adaptando las respuestas no solo a su nivel y sino también a un contexto concreto, presenta una ventaja significativa frente a los modelos generalistas, ofreciendo una experiencia más fiable y útil en entornos educativos reales. Además, su valor reside en que responder a una necesidad educativa aún no resuelta.

Cabe destacar que el uso de la IA en educación está ya respaldado por diversas investigaciones. El estudio “*Técnicas y aplicaciones del Machine Learning e Inteligencia Artificial en educación: una revisión sistemática*” [113], recoge múltiples aplicaciones de la IA en contextos educativos en niveles diferentes de educación (primaria, secundaria y superior) en 38 países, además de su utilización. Se debe destacar las siguientes aplicaciones, consecuencias y conclusiones recogidas:

- Detectar el rendimiento académico de los estudiantes de forma temprana.
- Mejorar las habilidades educativas de los docentes.

- Facilitar el aprendizaje de los estudiantes con trastornos del espectro autista (TEA).
- Predecir la deserción escolar y tomar decisiones al respecto.
- Mejorar y generar contenido educativo.
- Cerrar brechas educativas. Implementar la enseñanza de la IA en todos los niveles educativos.
- Fortalecer la seguridad de la información de la comunidad educativa. Motivar el aprendizaje a través de dispositivos móviles.
- Fortalecer el campo de la robótica.
- Mejorar la orientación académica y profesional de los estudiantes.
- Prevenir la difusión de noticias falsas en redes sociales.
- Entender y reflexionar sobre la relación entre humanos y máquinas.
- Desarrollar el pensamiento crítico a partir del pensamiento computacional.

Esto demuestra que la IA ya está siendo utilizada para tareas como el análisis del desempeño, tutoría automática, generación de materiales, o retroalimentación personalizada. [340] Asimismo, Guettala [127] revela estos numerosos beneficios tangibles derivados de la integración de la IA generativa, y otros artículos [288, 207] ofrece un análisis crítico sobre el papel de la IA en la personalización del aprendizaje (especialmente en el entorno universitario). De la misma manera, revisa siete estudios recientes que destacan avances como el aprendizaje adaptativo, el uso de *chatbots*, la evaluación automatizada y la creación de itinerarios personalizados. En él se señalan beneficios, pero también advierte sobre la necesidad de una integración ética y pedagógica. Además, subraya que, aunque la IA ofrece potencial innovador, muchas promesas son recicladas de intentos anteriores.

No obstante, es el momento de beneficiarse de todas esas conclusiones recabadas hasta ahora y aprovechar ese impulso tecnológico para avanzar hacia una educación más inclusiva, interactiva y comprensiva, donde la personalización no sea solo una opción, sino un derecho y la base del proceso formativo.

Finalmente destacar que este nuevo paradigma educativo exige que dejemos atrás el enfoque de igualdad para abrazar el principio de equidad, que implica adaptar los recursos y estrategias a las necesidades de cada estudiante. No se trata de dar un trato preferente, sino de reconocer la dignidad y el proceso único de aprendizaje de cada persona. La verdadera innovación pedagógica pasa por empatizar, personalizar y acompañar, promoviendo un entorno justo, respetuoso e inclusivo [323].

3.1.1. Contexto de Mercado

El 30 de noviembre de 2022, OpenAI lanzó ChatGPT al público. Ese momento marcó un antes y un después en la historia de la IA. No porque se creara una IA generativa que beneficiara a la humanidad, ya que este propósito llevaba vigente desde mucho antes, sino porque fue por primera vez accesible a todo tipo de usuarios, logrando a su vez más de un millón de suscripciones en 5 días [248].

Este aumento en la popularidad ha servido como punto de inflexión para seguir progresando en este tipo de tecnologías, y desde entonces, ha supuesto que el desarrollo de asistentes IA entre también en un mercado conversacional y con inserción en diversos sectores [300]. OpenAI fue el principal impulsor en traspasar la IA al público general, aunque no el primero en desarrollar modelos avanzados. Previamente, empresas como Apple con Siri o Google con Google Assistant ya había introducido asistentes y reconocimiento de voz basados en IA, pero su disponibilidad estaba limitado a un público que consumiese sus productos.

Actualmente, el panorama ha progresado con la aparición de Gemini, de Google o DeepSeek, entre otras, consolidando una competencia cada vez más fuerte en el mercado. No obstante, más allá de las distintas opciones disponibles, lo verdaderamente importante son los grandes avances que se están logrando en el PLN con LLMs.

Los LLMs han transformado la manera en que se realiza la interacción humano-máquina, ya que son capaces de generar lenguaje humano de forma coherente, estructurada y contextualizada [42]. Estos nuevos modelos han alcanzado un nivel inimaginable y sin precedentes para procesar y producir textos semánticamente correctos que son genéricos, ajustados y orientados a cualquier tipo de público. Sin embargo, esta evolución ha sido posible gracias al PLN, el cual permite a los sistemas interpretar no solo el significado literal de las palabras, sino también su contexto, intención y matices, facilitando por tanto que las interacciones con la IA sean más naturales, útiles y cercanas.

Desde una perspectiva de entorno de negocio, estas mejoras han impactado de manera significativa en sectores como la educación, el comercio y la salud, redefiniendo la relación entre la tecnología y sus usuarios [224]. La democratización de estas herramientas ha permitido que estudiantes y profesores accedan a soluciones avanzadas, brindando asistencia personalizada y accesible.

No obstante, la fuerte competencia entre las empresas dedicadas a ello ha acelerado la investigación, desarrollo y progreso de la IA, pero también han impulsado la comercialización de estos modelos generando preocupaciones en cuanto al libre uso, regulación, sostenibilidad y accesibilidad equitativa de la tecnología [258].

3.1.2. Stakeholders

El auge de una IA generativa libre es esencial para la transformación del entorno educativo, permitiendo que tanto estudiantes como profesores accedan a herramientas avanzadas otorgando una ayuda de forma más accesible e individualizada. Esta democratización de la IA no solo facilita el acceso al conocimiento, sino que también optimiza el proceso de enseñanza y aprendizaje, adaptándose a las necesidades específicas de cada usuario [335].

Por una parte, los **estudiantes** tienen la oportunidad de profundizar en aquello que les resulta más difícil de comprender o simplemente, dudas básicas de respuesta rápida. Además, la propuesta de múltiples ejercicios, cuestiones, explicaciones distintas a bajo nivel y una retroalimentación personalizada hacen posible que los alumnos puedan reforzar sus conocimientos a su propio ritmo y con mayor autonomía [249]. Todo ello, con el fin de que sea más fácil poder acceder y profundizar en aquello que conlleva más esfuerzo, problemas, o un menor entendimiento.

Por otro lado, para los **profesores** supone una gran optimización de tiempo y recursos. No solo por la parte de automatizar distintas correcciones o respuestas a los alumnos, sino también por la capacidad para generar distintos materiales más dinámicos, nuevas actividades innovadoras y otras estrategias pedagógicas activas para llegar mejor al alumnado y transmitir

verdaderamente pasión por el temario a impartir.

3.1.3. Desafíos y Normativas

Desarrollar un asistente de aprendizaje basado en el material de un profesor específico para su grupo de estudiantes permitirá ofrecer una enseñanza más justa, cercana y adaptada a las necesidades individuales. Al mismo tiempo, se fomentará una educación innovadora, actualizada y creativa que facilite el acceso al conocimiento para todo tipo de personas.

No obstante, no todo son ventajas y surgen nuevos retos más allá de la calidad de las respuestas y la personalización ya comentada anteriormente. Uno de los principales obstáculos es garantizar que la información proporcionada por el asistente sea precisa, clara y coherente con los materiales de la asignatura, evitando errores o confusiones que puedan afectar el aprendizaje [13]. Asimismo, la capacidad de adaptación a cada estudiante es fundamental, por lo que el asistente debe ofrecer explicaciones rigurosas, entendibles y adecuadas al nivel de conocimiento y estilo de aprendizaje de cada estudiante [347].

Otro aspecto preocupante es el riesgo de dependencia a la IA, donde un alumno que no tuviera iniciativa propia podría no aprender a replantearse ni contrastar la información proporcionada, afectando directamente a su capacidad de autocritica, reflexión y madurez [225]. Además, la integración práctica de la IA en el aula plantea interrogantes sobre cómo implementarla sin reemplazar la labor del profesor ni el incentivo del pensamiento autónomo de los estudiantes [71].

Finalmente, otro elemento a destacar sería la privacidad y ética en el uso de los datos, determinando hasta qué punto es conveniente proporcionar una información u otra, y asegurando que el asistente respete al alumno y cumpla con las reglas de la enseñanza [347].

Relacionado con esto último, otro matiz que ha de estar presente es el cumplimiento riguroso de las distintas leyes, normativas y regulaciones sobre IA y Protección de Datos, como el Reglamento General de Protección de Datos [306]. Además, debe prestarse especial atención también a la propiedad intelectual, asegurando que los contenidos generados o utilizados por el asistente respeten los derechos de autor. En este sentido, el Convenio de Berna para la Protección de las Obras Literarias y Artísticas protege los derechos de autor de materiales educativos, evitando la reproducción no autorizada de contenidos [125]. Por último, el Principio de Uso Justo (*fair use*) y las Excepciones Educativas permiten el uso de ciertos materiales protegidos en contextos académicos bajo condiciones específicas, siempre que no se infrinjan los derechos de los autores originales [30]. Entonces, algunas de las normas a cumplir más relevantes serán:

1. **Ley de Inteligencia Artificial de la Unión Europea (AI Act):** donde se establecen directrices para el desarrollo y uso seguro de la IA, clasificando los riesgos y exigiendo transparencia en sistemas automatizados [316].
2. **Reglamento General de Protección de Datos (RGPD):** regula el tratamiento de datos personales, garantizando la privacidad de los usuarios y estableciendo medidas de seguridad para su protección [315].
3. **Convenio de Berna para la Protección de las Obras Literarias y Artísticas:** protege los derechos de autor de materiales educativos, evitando la reproducción no autorizada de contenidos [237].

4. **Principio de Uso Justo (*fair use*) y Excepciones Educativas:** permite el uso de ciertos materiales protegidos en contextos académicos bajo condiciones específicas, siempre que no se infrinjan los derechos de los autores originales [234].

Por tanto, el presente proyecto no solo busca desarrollar un asistente de aprendizaje, sino que también tiene como desafío hacer frente a este tipo de retos, mejorando la calidad y personalización de las respuestas para que sean correctas, contextualizadas y acordes a lo definido en los materiales de la asignatura en el ámbito universitario. De esta manera, se pretende ofrecer una herramienta que realmente ayude y potencie el aprendizaje y la autonomía de los estudiantes, sin generar dependencia ni afectar en el rol del profesor durante el proceso educativo.

3.2. Estado del arte

A pesar de la novedad del tema, existen diferentes asistentes del aprendizaje que cubren, en mayor o menor medida, los objetivos planteados en este proyecto. Por ello, es necesario conocer sus fortalezas y debilidades, como base para abordar nuestros objetivos y diferenciar nuestra propuesta respecto al estado del arte. Para ello, estudiaremos las metodologías empleadas en estas soluciones, sus características principales y los *frameworks* en los que se basa su desarrollo (LLMs, PLN o RAGs).

Es importante destacar el uso de la técnica RAG, ya que se empleará para proporcionar los resultados generados a partir de la información relativa a la asignatura. Este enfoque mejorará significativamente la exactitud, precisión y contextualización de las respuestas, lo cual implicará un aumento en la calidad de las mismas.

De esta manera, un análisis crítico proporcionará las propuestas y tendencias actuales, así como conocer el impacto real en el ámbito de la educación y la importancia de realizar una solución innovadora, personalizada y accesible. A pesar de los avances actuales, no existe por ahora una integración completa donde un profesor aporte previamente su material docente y el asistente resuelva las dudas de manera estructurada dentro de un aula virtual o entorno educativo.

Bien es cierto que existen herramientas que permiten generar sus preguntas respecto a un temario proporcionado, como ChatGPT o el nuevo Asistente de IA de Adobe Acrobat Reader, que también emplean búsqueda semántica con *embeddings* y empleando la técnica RAG. En cambio, no hay todavía una integración efectiva clara que cumpla con el paradigma RAG en un entorno educativo ni satisface todas las necesidades pedagógicas, como pueden ser modelos que:

- Generen ejercicios adaptativos en función de la dificultad y dudas del estudiante.
- Evalúen en tiempo real el progreso del alumno, proporcionando métricas útiles tanto para estudiantes como para profesores.
- Formen respuestas precisas, totalmente correctas y ajustadas al contexto educativo sin necesidad de una supervisión humana constante.
- Sean accesible y de manera gratuita facilitando el acceso a todos los estudiantes, sin límites económicos.

Estas consideraciones previas se cumplirían en el proyecto, representando por tanto ese valor añadido de la solución propuesta al mantener un enfoque realista y centrado en desarrollar un asistente práctico, útil y valioso para un contexto educativo.

3.2.1. Descripción de trabajos relacionados

A continuación, se presentan algunas de las herramientas más destacadas que, aunque sea de manera parcial, ofrecen funcionalidades similares y permiten resolver dudas basándose en un temario proporcionado:

Knewton

Knewton [17] realmente es una empresa de aprendizaje adaptativo que ha desarrollado una plataforma basada en los datos de los materiales de sus cursos para que el sistema pueda personalizar el progreso del usuario otorgando ejercicios preexistentes en su propia base de datos. De esta manera, ofrece un análisis del rendimiento del alumno que le permite adaptar continuamente su aprendizaje, pero también fundamentado en las tendencias existentes [333].

Esto lo hace mediante distintos algoritmos de ML para predecir las necesidades de cada alumno y con modelos estadísticos para analizar su comportamiento. Por medio de un análisis minucioso de los datos recopilados en los usuarios a raíz de sus interacciones, diálogos y consultas, permite crear una experiencia individual.

La similitud con el proyecto a realizar se centra principalmente en el estudio del progreso de cada alumno, su evaluación continua y en la orientación del aprendizaje de unos materiales específicos. Asimismo, emplea técnicas de IA y PLN y a través de una interfaz amigable. Sin embargo, existen diferencias claras, como que el sistema no implementa la generación de ejercicios dinámicos personalizados basados en documentos específicos de asignaturas impartida por el profesor. Además, ofrece una plena accesibilidad, pero no es completamente gratuito.

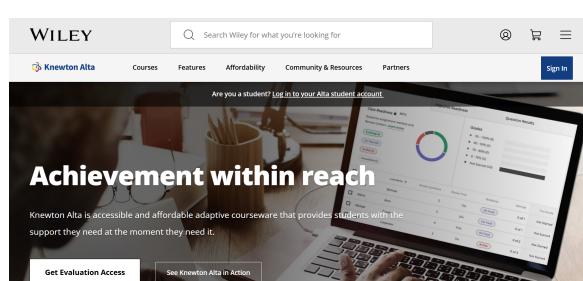


Figura 3.1: Inicio Knewton

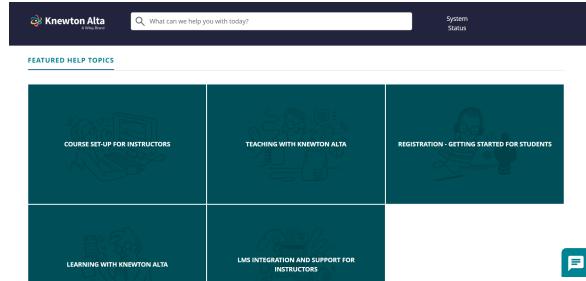


Figura 3.2: Support Knewton

IBM Watson Tutor – Jill Watson

IBM Watson Tutor [148] es otra plataforma educativa que también proporciona soporte a los alumnos, aunque esta vez recurriendo a tutorías personalizadas. Se enfoca en la generación de contenido adaptativo, ajustando el material a las capacidades, nivel y necesidades del estudiante. Además, realiza una posterior evaluación de su rendimiento progreso a lo largo del tiempo, promoviendo un aprendizaje dinámico, pero único [148].

Para ello, emplea IBM Watson AI, una herramienta avanzada con IA generativa y ML donde también aprovecha un análisis cognitivo para interpretar y proporcionar las respuestas y preguntas.

Al igual que en el caso anterior, la similitud se aprecia en la capacidad para generar ejercicios adaptativos, una interfaz amigable, el análisis del progreso y retroalimentación, además del uso de una IA cognitiva, ML y PLN.

En este contexto, cabe destacar que otro caso interesante relacionado con el uso de IBM Watson y a tener en cuenta sería Jill Watson. Se trata de un *chatbot* educativo desarrollado por el Instituto de Tecnología de Georgia que emplea esa herramienta para el PLN y junto a la IA conversacional logra responder preguntas de sus cursos online emulando a un tutor humano de manera interactiva, lógica, fluida y clara. Igualmente, tiene capacidad de aprender y mejorar en base a retrospectivas otorgadas [72], [308], [144].

Sin embargo, estos enfoques están más centrados en brindar tutorías automatizadas dentro de entornos concretos que en servir como herramientas de aprendizaje completamente autosuficientes. En particular, Jill Watson no implementa la técnica RAG ni permite el uso de documentos personalizados como base para generar contenido educativo dinámico, como sí hacen otras soluciones más recientes como Watson. No obstante, aunque Watson integra parcialmente capacidades avanzadas de RAG y otras funciones de IA generativa, se trata de una plataforma empresarial orientada a desarrolladores y no de una solución educativa diseñada específicamente para estudiantes. Por tanto, no incluye de forma nativa funcionalidades como la evaluación continua, la generación dinámica de ejercicios ni el seguimiento personalizado del aprendizaje en base al progreso del alumno. Asimismo, no son soluciones completamente personalizadas, accesibles ni gratuitos para todos los usuarios, limitando por tanto su disposición y uso.

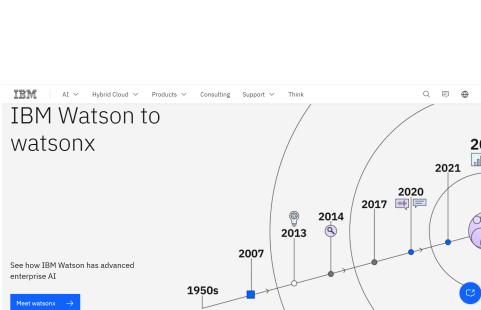


Figura 3.3: IBM Watson Tutor



Figura 3.4: Jill Watson

Khanmigo

Khanmigo [4] presenta una gran cercanía con el sistema a desarrollar en este proyecto. Este asistente emplea el modelo de PLN GPT-4 de OpenAI para interactuar con los estudiantes y usa como contexto los materiales educativos previos existentes en su plataforma Khan Academy [242], [73], [101].

Permite ofrecer una experiencia personalizada y proporcionar respuestas precisas, explicadas y relevantes. Además, también guía, orienta y se ajusta a las necesidades de cada estudiante en función de su nivel de aprendizaje y por medio de diálogos completamente naturales.

No obstante, la diferencia radica en que no se focaliza en el progreso ni evaluación del estudiante, no emplea la técnica RAG y se produce en un entorno donde no se especifica el contenido de la materia sobre el que generar las respuestas o ejercicios. Finalmente, aunque Khan Academy es gratuita y accesible independientemente de la situación económica del usuario, el acceso a Khanmigo está restringido a determinados usuarios.



Figura 3.5: Khanmigo

Socratic

Por otro lado, existe la aplicación educativa de Socrative [122], creada por Google, y con el mismo fin de orientar a los estudiantes. Para ello, genera una serie de contenido educativo ofreciendo recursos adicionales y realizando un diagnóstico de preguntas y resolución de posibles problemas en función de múltiples contenidos [331], [41]. Para tal efecto emplean sus propios modelos con Google AI, PNL y distintos algoritmos de ML para analizar las dudas de los usuarios, aunque también para las posibles imágenes que pueden adjuntar en la aplicación con sus propios ejercicios. Añadido a esto último, recurren a OCR (Reconocimiento Óptimo de Caracteres) para examinar posible texto en dichas imágenes.

En este caso, bien es cierto que alberga las funcionalidades de generación de ejercicios dinámica, su evaluación y la capacidad de anexar distintos documentos. En cambio, solo acoge la opción de agregar imágenes, ningún otro tipo más de documento previo. A causa de esto, tampoco se dispone la opción de basar y fundamentar al asistente en otros recursos anexados anteriormente por el profesorado y, por tanto, tampoco el uso de la técnica RAG. Otro aspecto a tener en cuenta sería la falta de revisión del progreso del alumno, ya que se centra más en la contestación inmediata a cerca de dudas específicas que en un acompañamiento continuo.

Sin embargo, el punto a favor de Socrative es que es completamente gratuito y accesible para cualquier usuario de cualquier tipo de *hardware*, tanto iOS como Android.



Figura 3.6: Socrative

Perplexity AI

Perplexity AI [10] es otra herramienta fuertemente relacionada con el sistema a desarrollar en nuestro proyecto, ya que es un modelo de búsqueda conversacional (análogo a un “motor de consultas”) que emplea LLMs con capacidades de PLN, otros modelos de ML, como GPT-4, Claude 3, Mistral Largue, y su propio modelo experimental [9], [11], [295]. Con ello, se obtienen respuestas rápidas, concisas y precisas, pero adicionalmente semántica y sintácticamente correctas. No obstante, lo que se destaca de este modelo, aparte de su búsqueda avanzada y generación de soluciones lógicas y adecuadas, es el uso de la técnica RAG para acceder a una base de datos

externa y desarrollar esas respuestas en tiempo real en función de una información detallada, contextualizada y focalizada. En consecuencia, consigue un alto rendimiento, experiencia específica del dominio, relevancia y privacidad.

Pese a ello, carece de varias funcionalidades educativas, en concreto, la creación dinámica y sistemática de ejercicios, la evaluación constante, actualizada e individualizada del estudiante y su continuo análisis del aprendizaje y personalización del contenido educativo. Aunque los modelos de lenguaje que utiliza pueden generar ejercicios en tiempo real durante una interacción, estas capacidades no están integradas ni optimizadas dentro de la plataforma para un seguimiento educativo completo. Por ello, estas funcionalidades representan una oportunidad para desarrollar un enfoque educativo novedoso y diferencial en este proyecto. Además, del mismo modo que en casos anteriores, consta de una versión gratuita accesible, pero para mayor servicio es imprescindible depender de la de pago. En compensación, estas carencias son significativas para poder incorporar a este proyecto un enfoque novedoso, inexistente y revolucionario.



Figura 3.7: Perplexity AI

Mathew

Por otro lado Mathew [5], un asistente educativo desarrollado por la startup española Adaptical, con el objetivo de facilitar tanto a docentes como a estudiantes la creación de contenido, el acompañamiento personalizado y la evaluación del aprendizaje. Esta herramienta está pensada para adaptarse a las necesidades específicas de cada centro educativo, permitiendo generar situaciones de aprendizaje personalizadas, actividades con distintos niveles de dificultad, así como evaluaciones automáticas alineadas con los criterios curriculares [6].

Mathew se apoya en técnicas avanzadas de IA y PLN para ofrecer un tutor virtual capaz de resolver preguntas teóricas y hacer seguimiento del progreso de cada estudiante. Además, cumple con el RGPD y mantiene sus servidores en Europa, garantizando así la seguridad y privacidad de los datos [5].

Aunque presenta similitudes con este TFG en cuanto a la personalización del aprendizaje y el uso de IA para ofrecer retroalimentación adaptativa, no se ha identificado que implemente explícitamente la técnica RAG ni que permita trabajar directamente con documentos personalizados aportados por el profesorado. De igual forma, su orientación está más centrada en el currículo oficial que en la generación dinámica de contenido a partir de materiales propios. En consecuencia, este proyecto se diferencia de Mathew por poner el foco en la integración de RAG para generar contenido educativo dinámico a partir de documentos concretos proporcionados por el profesorado. Este enfoque permite una solución más flexible, abierta y extensible, centrada en la personalización del aprendizaje a partir de materiales reales y contextualizados.



Figura 3.8: Mathew

EdGPT

Cabe destacar EdGPT, un término reciente que se está utilizando en el mundo académico para describir modelos LLMs adaptados al ámbito educativo. Realmente es un enfoque o concepto que parte de modelos como GPT-3, GPT-4, LLaMA entrenados con material docente (CV, libros de texto, actividades pedagógicas...) enfocados en tareas educativas (generar planes de clase, resolver ejercicios, adaptar el nivel del contenido, ayudar en tutoría). Sin embargo, aun se encuentra en una fase inicial. Existen prototipos (como EduChat o MathGPT), pero no hay una única implementación oficial ni API pública de EdGPT [322]. Aunque tiene un enfoque totalmente centrado en la educación, no implementa RAG explícitamente y se encuentra aun en desarrollo.

Otros

Por otro lado, también es importante resaltar que existen otros asistentes de aprendizaje como Duolingo Max, MATHia, Quizlet, Brainly, Gradescope, entre otros. Estos asistentes, aunque relevantes, tienen algunas diferencias clave con el enfoque propuesto en este TFG, que se basa en el uso de RAG. A continuación, se detallan brevemente:

- **Duolingo Max:** utiliza IA generativa para personalizar las lecciones, pero no emplea RAG para acceder a información externa ni generar respuestas personalizadas en tiempo real [99].
- **MATHia:** personaliza la dificultad de los ejercicios matemáticos según el alumno, pero no usa RAG ni consulta fuentes externas durante el aprendizaje [215].
- **Quizlet:** se basa en tarjetas de estudio adaptativas, pero no genera contenido dinámico en función de documentos proporcionados ni usa RAG para enriquecer las respuestas [255].
- **Brainly:** aunque ofrece respuestas generadas por IA, no emplea RAG para acceder a grandes bases de datos externas ni personaliza las respuestas en función del contexto de los estudiantes [44].
- **Gradescope:** se centra en la corrección automatizada de tareas y la retroalimentación, pero no incorpora RAG ni personalización dinámica en tiempo real [126].

Sin embargo, se ha optado por resaltar los anteriores por el mero hecho de su importancia, su popularidad, su relevancia en el ámbito de la enseñanza educativa, su avance tecnológico y su considerable impacto en el análisis, personalización y accesibilidad del aprendizaje del alumno en cuestión.

3.2.2. Discusión

Ahora discutimos con las tablas resumen 3.1 y 3.2

Proyecto	¿Usa RAG?	Tecnología/Modelo	Funcionalidad Principal
Knewton	No	IA, ML, PLN	Personalización de contenido educativo y evaluación adaptativa
IBM Watson Tutor	Parcial	IBM Watson IA (IA Cognitiva, ML, PLN)	Tutor personalizable y análisis de progreso de estudiantes
Jill Watson	No	IBM Watson IA	Respuestas automáticas en foros de discusión educativa
Khanmigo	No	PLN (GPT-4), IA	Tutor virtual para guiar a los estudiantes y generar respuestas personalizadas
Socratic by Google	No	Google AI: ML, PLN, OCR	Respuestas y explicaciones a preguntas en diversas materias
Perplexity AI	Sí	GPT-4, RAG	Generación de respuestas precisas basadas en datos online con arquitectura RAG
MATHEW	No	IA, PLN	Asistente educativo personalizable para centros escolares; ofrece tutoría digital y apoyo adaptado a las necesidades del alumnado
TFG	Sí	PLN, OpenAI(GPT-3.5)	Generación de ejercicios adaptativos, respuestas precisas, y seguimiento de progreso

Tabla 3.1: Comparativa de tecnologías educativas basadas en IA (1/2)

Capítulo 3. Antecedentes

Proyecto	Generación de Ejercicios	Evaluación en Tiempo Real y Métricas	Respuestas Precisas y Contextualizadas	Accesibilidad y Gratuito
Knewton	Sí, adapta ejercicios según el nivel estudiante	Sí, seguimiento continuo del progreso estudiante	Sí, ajustadas a la dificultad y contexto	Freemium: Accesible, pero no completamente gratuito
IBM Watson Tutor	Parcial, según el progreso del estudiante	Sí, métricas detalladas	Sí, basadas en la comprensión de los datos educativos	Freemium: Accesible, pero no completamente gratuito
Jill Watson	No genera ejercicios, pero responde preguntas en foros y asiste en la discusión	No evalúa el progreso de los estudiantes	Sí, genera respuestas precisas y relevantes en los foros	Gratis en entornos educativos
Khanmigo	No, pero ayuda a los alumnos con problemas	No, métricas limitadas en tiempo real	Sí, respuestas precisas y contextuales	Gratis en Khan Academy
Socratic by Google	No, pero ayuda a resolver problemas y preguntas	No evalúa el progreso, pero proporciona respuestas detalladas	Sí, respuestas precisas y adaptadas al contexto de la pregunta	Gratis
Perplexity AI	No, orientado a consulta de información	No evalúa el progreso del estudiante ni proporciona métricas en tiempo real	Sí, respuestas precisas vía RAG	Gratis para uso básico
Mathew	No genera ejercicios directamente, pero apoya en tareas mediante <i>feedback</i> automatizado	No incluye métricas detalladas ni seguimiento del progreso en tiempo real	Sí, contextualizada por preguntas y materias educativas	Licencia institucional No es de libre acceso generalizado; depende de convenios con centros educativos
TFG	Sí, genera ejercicios adaptativos	Sí, evalúa el progreso del estudiante	Sí, proporciona respuestas precisas	Gratis

Tabla 3.2: Comparativa de tecnologías educativas basadas en IA (2/2)

3.3. eXtreme Learning

La personalización efectiva del aprendizaje trasciende al panorama tecnológico, revisado previamente. Este desafío demanda metodologías específicas que permitan integrar estas herramientas de forma que favorezcan los procesos de enseñanza-aprendizaje. En este proyecto, usamos eXtreme Learning (χLe) [213] como marco de referencia y orientamos el desarrollo de nuestro asistente en torno al concepto de producto de aprendizaje que se define en ella. Para ello, desplegaremos nuestro asistente en el entorno de Sistemas de Bases de Datos, una asignatura universitaria organizada y desarrollada con eXtreme Learning en el Grado de Ingeniería Informática de Servicios y Aplicaciones de la Universidad de Valladolid.

Cabe resaltar que este modelo delega en el propio estudiante la responsabilidad principal sobre su proceso de aprendizaje, por lo que es el punto de partida ideal para conseguir que el alumno participe en el propio proceso educativo. Además, según Delgado Martínez [90] el aprendizaje es de mayor calidad cuando el estudiante “experimenta la construcción de un producto con sentido”. Este *framework* asienta las bases del aprendizaje y sus recursos, tecnologías y procesos de una forma clara, tangible y estructurada, por lo que su uso va a facilitar lograr esa ansiada personalización en el entorno educativo.

Precisamente, uno de los objetivos fundamentales de este TFG es avanzar hacia un modelo de enseñanza más personalizado, accesible y centrado en el estudiante. Esto implica no solo adaptar los contenidos, sino también ofrecer apoyo contextualizado, facilitar el progreso individual, y permitir que cada alumno pueda profundizar en aquellos aspectos que le resultan más significativos o necesarios. En este sentido, χLe proporciona un marco metodológico especialmente adecuado gracias a la existencia de una estructura clara basada en objetivos de aprendizaje, historias de aprendizaje y criterios de aceptación (inspirados en metodologías ágiles). Todo ello configura el *Producto de Aprendizaje*. Este planteamiento no solo permite una planificación más concreta y alineada con los intereses del alumnado, sino que facilita al estudiante definir y profundizar en aquello que realmente necesita o desea aprender.

3.3.1. Características del Producto de Aprendizaje

El producto de aprendizaje representa el resultado final que un estudiante debe alcanzar para completar con éxito un proyecto de aprendizaje. Se define partiendo de las competencias del plan de estudios y las demandas de la sociedad al determinar qué conocimientos y habilidades deben adquirir los titulados universitarios.

Se definen cuatro niveles de granularidad propuestos para la especificación del producto que ayudan a satisfacer las necesidades de los implicados:

1. **Visión:** ofrece una descripción general y breve del objetivo global de la asignatura desde una perspectiva social y profesional. Su finalidad es ayudar al estudiante a entender el propósito y valor del aprendizaje en un contexto laboral real.
2. **Objetivos de aprendizaje:** refinan la visión y definen los hitos principales en bloques temáticos del proyecto educativo. Funcionan como metas principales dentro del producto de aprendizaje, y agrupan diversas historias de aprendizaje. Cada uno aclara su utilidad y relación con el aprendizaje global. Las historias de aprendizaje propuestas en la asignatura Sistemas de Bases de Datos se muestran en la figura 3.9.

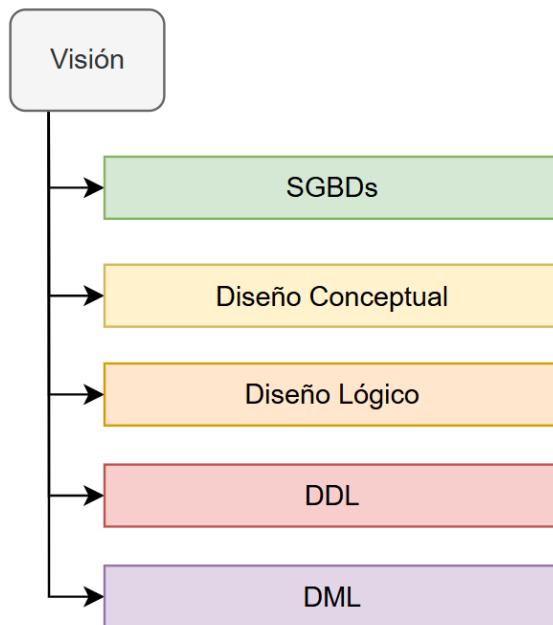


Figura 3.9: Objetivos del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos

3. Historias de aprendizaje (subobjetivos): se establecen subjetivos más específicos dentro de cada bloque. Se enfocan en lo que el estudiante va a aprender y por qué es importante. Se inspiran en las “características funcionales” del producto en metodologías ágiles, y cada historia tiene valor educativo en sí misma. Las historias de aprendizajes para cada objetivo se muestran en la figura 3.10:

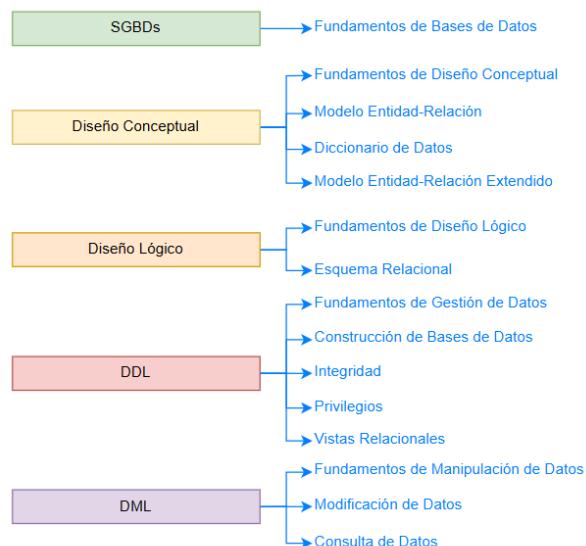


Figura 3.10: Historias de Aprendizaje del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos

4. **Criterios de aceptación:** son los elementos más concretos y medibles del producto de aprendizaje. No tienen valor funcional por sí solos, pero son esenciales para completar una historia. Sirven como base para la evaluación y el *feedback*, y son comunicados a los estudiantes durante el sprint correspondiente. Ejemplo de ello se muestra en la figura 3.11, que se encuentra dentro del Objetivo Diseño Lógico e Historia de Aprendizaje Modelo Relacional (Conocer las características del modelo relacional para ser capaz de obtener el diseño lógico relacional de cualquier base de datos).

- **LÓGICO-2.1.** Soy capaz de transformar cualquier entidad existente en un diseño conceptual (con sus atributos correspondientes) en su representación relacional normalizada (3FN).
- **LÓGICO-2.2.** Soy capaz de transformar cualquier relación existente en un diseño conceptual (con sus atributos correspondientes) en su representación relacional normalizada (3FN).
- **LÓGICO-2.3.** Soy capaz de establecer el dominio de cualquier columna a partir de la información disponible en el diseño conceptual.
- **LÓGICO-2.4.** Soy capaz de establecer restricciones de integridad de entidad a partir de la información disponible en el diseño conceptual.
- **LÓGICO-2.5.** Soy capaz de establecer restricciones de integridad referencial a partir de la información disponible en el diseño conceptual.

Figura 3.11: Ejemplo Criterio de Aceptación del Producto de Aprendizaje de la asignatura Sistemas de Bases de Datos

En el marco de este TFG, se propone aplicar la estructura de eXtreme Learning para favorecer una experiencia personalizada de aprendizaje. En concreto, se ofrecerá al estudiante la posibilidad de seleccionar, desde el inicio del proceso, qué objetivos de aprendizaje desea abordar y, dentro de ellos, qué historias de aprendizaje quiere trabajar en profundidad. Esta estrategia permite adaptar los contenidos a sus intereses, nivel, ritmo y necesidades formativas, promoviendo así un enfoque más activo, autónomo y significativo del aprendizaje. Además, se busca fomentar el compromiso del alumno al darle voz en la planificación de su trayectoria educativa, alineándose plenamente con los principios de personalización que inspiran este TFG. Por tanto, la generación de los recursos se apoyará completamente en la jerarquía de especificación de los objetivos de aprendizaje que plantea χLe .

Asimismo, es necesario resaltar que χLe no solo define de una manera clara el Producto de Aprendizaje, si no que también se recaba conceptos como el *Proceso de Aprendizaje, Entorno de Aprendizaje y Resultados*, donde se recoge el aprendizaje, metodología, trabajo, revisión y entorno de los proyectos que se enmarcan dentro del χLe . No obstante, no son tan relevantes para la consecución del fin del presente TFG, el cual es el desarrollo de un asistente de aprendizaje que permita la personalización en el entorno educativo y sirva como apoyo a los estudiantes.

Finalmente, resaltar que χLe tiene como un reciente objetivo abordar el uso de Inteligencia Artificial Generativa (IAG), de tal manera que se demuestre el potencial de la IAG para automatizar la creación de recursos de aprendizaje y la evaluación de tareas. Así, el propósito principal de ese proyecto es consolidar una guía de referencia que le sirva al profesorado universitario para automatizar los procesos referidos, con independencia del contexto en el que se requiera. Por tanto, el presente TFG, recoge las características ideales para enmarcarse como un proyecto dentro de χLe . Es más, este trabajo se sitúa como uno de los primeros desarrollos encaminados a materializar esa intención, por lo que podría servir como referencia útil para seguir avanzando en esa dirección de lograr incorporar y analizar la IAG en el aprendizaje.

Capítulo 4

Inteligencia Artificial Generativa

Desde tiempos inmemoriables el ser humano ha sido por naturaleza un ser curioso. Esa necesidad de saber es lo que lo ha impulsado hacia un futuro tecnológico mejor: con una constante evolución y transformación del mundo. Poco a poco ha ido creciendo dentro de diferentes áreas de conocimiento, facilitando además labores de la vida diaria. Sin embargo, no solo ha revolucionado nuestra cotidianidad, sino que también ha logrado grandes resultados en sectores como la medicina, industria y ciencia. Gracias a esta curiosidad, la humanidad ha podido encontrar soluciones para grandes problemas, incluidos aquellos derivados de sus propios conflictos, como las guerras.

Sin embargo, la adopción de nuevas tecnologías no ha estado exenta de recelos y temores. Un claro ejemplo de ello sería planteado en la obra de teatro R.U.R de Karel Čapek estrenada en 1921. En ella se introduce un nuevo concepto de robots como máquinas dotadas de sentimientos que eventualmente se rebelaban contra sus creadores humanos. Fue considerada una obra de gran influencia en la cultura popular y científica de la época, planteando desde entonces la inquietud sobre el posible dominio de las máquinas inteligentes sobre los humanos [348].

Afortunadamente, este tipo de relatos también sirvió de contrapartida para la comunidad investigadora, puesto que a partir de esa fecha han sido numerosos los hallazgos e hitos importantes dentro de este campo, y en especial, en el de la automatización e IA [273], [98].

El caso más conocido de la época es el de la Segunda Guerra Mundial, donde la tecnología jugó un papel esencial en la estrategia militar y victoria de los Aliados. Alan Turing diseñó Bombe, una máquina que permitía descifrar los mensajes de los nazis más rápido que cualquier humano lo haría manualmente. Debido a ello, los Aliados lograron anticiparse a los movimientos alemanes, permitiendo salvar millones de vidas, ya que se estima que ayudó a reducir la guerra en dos años [137], [74].

A partir de ese momento, el mundo entró en una era de avances tecnológicos, desde la creación del primer ordenador, pasando por el artículo del Test de Turing en 1950 [314], hasta llegar al nacimiento oficial de la inteligencia artificial en la Conferencia de Dartmouth de 1956, donde John McCarthy acuñó el término “inteligencia artificial” para describir “la ciencia y la ingeniería de hacer máquinas inteligentes”[219]. La Inteligencia Artificial (IA) es “un campo de la informática que se enfoca en crear sistemas que puedan realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, el razonamiento y la percepción”[273]. Una vez definido este fundamento, se desarrollaron desde entonces los primeros algoritmos de aprendizaje automático, y posteriormente, redes neuronales, *deep learning* y así hasta llegar a nuestros días con el gran auge del *Big Data*, *Machine Learning* e IA Generativa. Todo ello con el objetivo de

imitar la inteligencia humana, resolver problemas complejos, reconocer de patrones o tomar de decisiones de forma autónoma y coherente.

No obstante, a tal efecto, es imprescindible contar con una información clara, precisa y representante en la que basarnos para poder aprender sobre esa inteligencia humana: los datos. Sin datos no hay inteligencia; son la base fundamental para que las máquinas aprendan a “pensar” y desarrollen esas habilidades cognitivas que nos diferencian respecto al resto de las especies. Por tanto, la calidad y cantidad de datos es crucial para el desarrollo, progreso y éxito de la inteligencia artificial [211] [49].

Hoy en día, la palabra datos se utiliza con demasiada normalidad y frecuencia. Su popularidad ha aumentado drásticamente en los últimos años y su importancia, todavía más. Dato tiene su origen en el latín “Datum”, que significa “lo dado” y se define según la RAE como “información sobre algo concreto que permite su conocimiento exacto o sirve para deducir las consecuencias derivadas de un hecho” [261]. Sin embargo, un conjunto de datos proporciona un conocimiento general sobre los hechos e información valiosa debido a diferentes patrones, relaciones y tendencias que se repiten continuamente. Por tanto, no es de extrañar que, en la actualidad, en un mundo de optimización para obtener el máximo beneficio, sea esencial analizar esos datos para poder sacar el mayor rendimiento de ellos y tomar decisiones más informadas.

Además, dado el volumen masivo de datos generados continuamente, su estudio se ha convertido en un área de interés creciente. En cambio, para poder procesar ese gran volumen de datos o *Big Data*, es imprescindible utilizar distintas herramientas como el *Machine Learning*, minería de datos o análisis predictivos [218].

El auge del *Big Data* y los avances en el aprendizaje automático han permitido el desarrollo de sistemas capaces no solo de analizar información, sino de crear contenido nuevo y original. En este contexto surge la inteligencia artificial generativa (IAG), una rama de la IA que se centra en la generación automática de texto, imágenes, audio u otros tipos de datos a partir de patrones aprendidos durante su entrenamiento [273, 98].

A diferencia de los sistemas tradicionales, que se limitan a clasificar o predecir, los modelos generativos pueden producir información innovadora, inédita y coherente, emulando en cierta medida la creatividad humana. El desarrollo de arquitecturas como *deep learnign* y los LLM ha impulsado la IAG a nuevos niveles, permitiendo aplicaciones novedosas en educación, arte y comunicación. En el ámbito educativo, estas tecnologías facilitan la personalización del aprendizaje, generando recursos, explicaciones y ejercicios adaptados a las necesidades y ritmo de cada estudiante; así como su posible medida de progreso en tiempo real.

Por tanto, la IAG no solo amplía el alcance de la automatización, sino que también plantea nuevos retos y oportunidades en términos de creatividad, ética y personalización del aprendizaje.

La Inteligencia Artificial Generativa (IAG) es un tipo de IA que tiene como fin crear contenido original como textos, imágenes, música, videos y códigos a partir de patrones y datos previamente aprendidos [273, 98, 170]. Por tanto, hay cierta similitud con la creatividad humana. A través de técnicas avanzadas de *Machine Learning* y *Deep Learning* [281], la IAG es capaz de comprender y reproducir estructuras complejas de información, generando resultados que no solo son coherentes, sino también creativos y funcionales.

Aunque a diferencia de los enfoques tradicionales de IA, que se centran en analizar o clasificar datos, la IAG es capaz de producir nuevas ideas, explicaciones, resúmenes y recursos educativos de manera autónoma, ampliando así las posibilidades de personalización y creatividad en el ámbito educativo [130, 138, 339].

El uso de la IAG está revolucionando la enseñanza y el aprendizaje mediante la facilita-

ción de generación de material didáctico, adaptación de contenidos al ritmo y necesidades de cada estudiante y la automatización de tareas docentes. El uso de herramientas basadas en ella permitirá la personalización del aprendizaje, como ofrecer explicaciones, crear ejercicios adaptativos y generar respuestas contextualizadas, lo que incrementa la eficiencia docente y mejora el rendimiento académico [170, 130, 138, 299].

En este sentido, el asistente de aprendizaje propuesto en este trabajo se fundamenta en técnicas de IAG y específicamente en la arquitectura *Retrieval-Augmented Generation* (RAG), que combina la generación de lenguaje natural con la recuperación de información relevante para ofrecer respuestas más precisas y actualizadas. Por ello, comprender los fundamentos y el impacto de la IAG es esencial para el desarrollo y la justificación del sistema presentado en este TFG. En este capítulo, se mostrarán esos fundamentos necesarios y más tarde, se entrará en materia con el RAG.

4.1. Fundamentos

Para comprender en profundidad el funcionamiento y el potencial de la inteligencia artificial generativa en el ámbito educativo, es fundamental conocer primero los principios y tecnologías que la hacen posible. A continuación, se presentan los conceptos esenciales del *Machine Learning*, así como redes neuronales y el *Deep Learning* para fundamentar la base de los modelos generativos más avanzados.

4.1.1. Machine Learning

El *Machine Learning* (ML) es el área dentro de la inteligencia artificial que se centra en desarrollar técnicas que permitan que las computadoras aprendan. O visto de otro modo, que permitan mejorar el desempeño de tareas específicas a partir de la experiencia, sin ser programadas para ello [289, 332, 280]. Por tanto, se basa en la búsqueda de patrones para luego hacer predicciones, reconocer, agrupar y clasificar instancias y tomar decisiones en relación con la identificación y conclusiones de esas respuestas. Esto deriva en que el ML tiene como resultado un modelo para resolver una tarea dada [84, 297, 193].

Para ello, es necesario una serie de datos de entrada, que pueden ser tanto iniciales como la salida esperada para ellos. Una vez que se tiene ese conjunto de datos suficiente, preciso y representativo, se procedería a crear o emplear los diferentes algoritmos de aprendizajes o modelos junto al distinto poder de cómputo, herramientas y bibliotecas para así obtener los diversos resultados deseados. Finalmente, una vez entrenados dichos modelos, habría que evaluar su precisión, ajustarlos y volver a analizarlos para mejorar su desempeño y rendimiento final [89, 275, 110].

Sin embargo, son muchos los obstáculos que se encuentran durante el desarrollo de esta disciplina, como puede ser la capacidad de cómputo, ya que la mayoría de los problemas a los que se enfrenta son de tipo *NP-hard* [177]. Encontrar una solución factible es realmente una auténtica aventura.

Aun así, tiene aplicaciones en una gran variedad de sectores: desde la propia informática hasta campos como medicina, ganadería, cualquier tipo de entretenimiento o en los que no existe la experiencia humana previa (como la navegación de Marte). El ML incluye funciones relacionadas con motores de búsquedas, reconocimiento del habla, del lenguaje o robótica; predicción de fallos en las máquinas de las industrias, detección de rostro, genética, *forecast*, análisis de imágenes, de

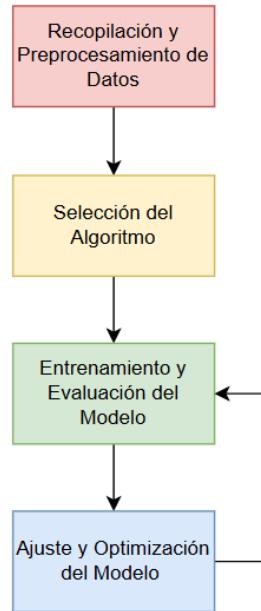


Figura 4.1: Etapas Proceso ML

comportamientos y productividad, vehículos autónomos y robots, resumen de textos, diagnósticos médicos, minería de datos, *Big Data*, previsión de series temporales... [77], [273, 231, 37, 60, 129].

Por otra parte, es importante conocer, aunque sea de forma general, el proceso de desarrollo o construcción de un modelo ML [273, 231, 37, 60, 129], el cual implica una serie de etapas que son esenciales llevar a cabo, como las mostradas en la figura 4.1:

1. **Recopilación y preprocesamiento de los datos:** en esta primera etapa se recogen los datos necesarios para entrenar el modelo. Luego, estos deben ser procesados para obtener conjuntos con calidad, precisión y representativos del problema a resolver. Incluye diferentes pasos esenciales:

- Limpieza de datos: identificar y corregir datos erróneos, como duplicados, nulos o incorrectos para mejorar la fiabilidad del conjunto.
- Transformación de datos: conversión de los datos en un formato adecuado para el modelo.
- Integración de datos: integrar datos que provengan de diferentes fuentes.
- Normalización de datos: expresar los datos en las mismas unidades de medida, escala o rango.
- Imputación de datos perdidos: completar o estimar valores faltantes debido al proceso de recopilación.
- Identificación de ruido: encontrar y eliminar valores que no aportan información relevante, a través de, por ejemplo, detectar errores aleatorios o varianzas en las variables medidas.

2. **Selección del algoritmo:** seleccionar el algoritmo de ML que mejor se adapte a la naturaleza del problema a resolver, teniendo en cuenta por tanto el tipo de aprendizaje que se va a tratar (se explicará en los apartados sobre aprendizaje en 4.1.1).
3. **Entrenamiento y evaluación del modelo:** en una primera parte de esta etapa, se entrena el modelo con los datos recopilados. No obstante, para ello se divide primero el conjunto en tres subconjuntos:
 - Conjunto de Entrenamiento: son los datos empleados para entrenar el modelo. Corresponde entre el 60 - 80 % de los datos del conjunto total.
 - Conjunto de Validación: se aplica para validar y ajustar los hiperparámetros del modelo.
 - Conjunto de prueba: es el conjunto que se utilizará a la hora de evaluar la precisión del modelo final después de haber sido validado y entrenado. Alberga entre el 10-20 % del total.

Para seleccionar qué parte de ejemplos usar para el entrenamiento y qué para la validación, existen dos técnicas distintas:

- **Hold-out:** se divide aleatoriamente el conjunto de datos en dos subconjuntos: uno para entrenamiento y otro para prueba (por ejemplo, 80 % - 20 %). El modelo se entrena con el primer conjunto y luego se evalúa con el segundo. Es rápida y sencilla, pero su precisión depende de la forma en la que se realice la división de los datos.
- **K -fold Cross Validation:** en este caso los datos se dividen en k subconjuntos de igual tamaño. A raíz de ello, se hacen k aprendizajes, utilizando en la iteración i el subconjunto i como prueba y el resto de entrenamiento. Para acabar, se toma la media aritmética de las k iteraciones realizadas para obtener la medida de evaluación final. Es más robusta, pero cuenta con mayor costo computacional.

Finalmente, en la segunda etapa se procederá a la evaluación del modelo para poder analizar y extraer conclusiones de ello.

4. **Ajuste y optimización del modelo:** por último, una vez evaluado de manera inicial, se procederá a un reajuste (en caso de ser necesario) del modelo en cuestión para mejorar su desempeño, lo que conllevan repetir las etapas 2 y 3, pero incluir a su vez una optimización de los parámetros o considerar otras características. Destacar que el conjunto de entrenamiento-validación-prueba (*train-validation-test*) ha de ser el mismo.

Una vez visto el proceso de construcción del modelo de aprendizaje automático, es necesario emplear distintos algoritmos en función del problema que buscamos resolver. Existen tres tipos principales en los que se clasifica dependiendo de la forma en la que aprende el modelo según los datos:

Aprendizaje Supervisado

Recibe como entrada datos etiquetados, es decir, una salida conocida. Por tanto, sus modelos se basan en clasificar si la salida generada coincide con la esperada, siendo por ende un procesamiento entre cada par de entrada/salida para ver si la clase inferida coincide en ambos y en función de ello, variar el modelo inicial [146, 155, 190].

El esquema básico de este aprendizaje es aprender cierto concepto objetivo a partir de ejemplos (los cuales están definidos por unos atributos de los que se conocen los resultados). Dentro de este punto, se encuentran otros dos tipos de aprendizaje:

- **Regresión:** predice un valor numérico continuo para las instancias en cuestión de los parámetros de los atributos. Un ejemplo de ello sería predecir el precio de una vivienda en función de características como su antigüedad, vecindario, metros cuadrados, ubicación...
- **Clasificación:** se centra en identificar a qué clase pertenece una determinada muestra del problema, por lo que se asigna etiquetas o categorías a las instancias de entrada. Un caso típico sería diferenciar entre correos de spam – no spam.

En cambio, se encuentran ciertos desafíos a tratar, como sería que el modelo clasifique bien los ejemplos dados, pero también generalice los casos nuevos. Asimismo, se ha de evitar un sobreajuste e identificar el ruido y distintos outliers posibles dentro del conjunto de datos inicial.

Finalmente, ejemplos de algoritmos de aprendizaje supervisado destacables serían tales como Naive Bayes, SVM, Regresión Logística, Regresión Lineal, Árboles de Decisión, Ensembles, K-NN y Redes Neuronales.

Aprendizaje No Supervisado:

En este caso recibe datos sin etiquetar, por lo que sus modelos tratarán de buscar patrones ocultos dentro de los datos o realizar distintas predicciones, generalmente regresiones matemáticas. Por tanto, el objetivo a seguir es visualizar o preprocessar subgrupos sin ningún tipo de etiquetado en los datos. Para ello, existen otros dos tipos principales de aprendizaje no supervisado:

- **Clustering:** es una técnica de agrupación de datos sin etiquetar dependiendo de sus similitudes o diferencias. Los algoritmos principales incluyen el k-medias, k-medianas, Cobweb o SOM.
- **Reducción dimensionalidad:** consiste en reducir la dimensionalidad de los datos sin perder información, por lo que se suele emplear antes de utilizar el aprendizaje supervisado para visualizar o preprocessar los datos. Un ejemplo de este algoritmo sería el análisis de componentes principales (PCA).

Aprendizaje Por Refuerzo:

El algoritmo en cuestión aprende interactuando con el entorno y obteniendo una recompensa positiva o negativa en función del resultado dado y del deseado. Cabe destacar que es aplicable en áreas de datos que pueden simularse o en grandes volúmenes de datos, y siempre en base al aprendizaje resultante intentarán obtener recompensas positivas.

Otros tipos de aprendizaje

Adicionalmente, existen otros tipos de aprendizaje, especialmente a resaltar:

- **Aprendizaje semi-supervisado:** cuando se dispone tanto de datos etiquetados como sin etiquetar.

- **Aprendizaje no balanceado:** que se suele dar cuando hay problemas de distribución de los datos entre las clases de interés.
- **Clasificación multietiqueta:** donde cada instancia está asociada a un subconjunto de clases en lugar de una sola.
- **Aprendizaje por transferencia:** se utilizan patrones o modelos previamente entrenados para resolver problemas específicos distintos. Suele ser el caso de las redes profundas o como se apreciará más adelante, el presente proyecto con el uso de los LLMs, al emplear modelos ya entrenados en el ámbito del PLN y para desarrollar el asistente de aprendizaje.

Métricas

Una vez entrenado el modelo, es necesario evaluar su rendimiento utilizando datos que no hayan sido empleados durante el proceso de aprendizaje. Para ello, se aplican diferentes métricas que permiten cuantificar la calidad del modelo según el tipo de tarea, ya sea clasificación, regresión u otros enfoques.

En problemas de clasificación, las métricas de evaluación más habituales (tabla 4.1) se basan en dos tipos de errores: los *falsos positivos*¹ y los *falsos negativos*². Estos conceptos permiten calcular medidas como la precisión, la sensibilidad o la especificidad.

En cambio, para problemas de regresión, las métricas más comunes (tabla 4.2) son el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), el error absoluto medio (MAE) o el coeficiente de determinación R^2 [76]:

¹Un falso positivo ocurre cuando el modelo predice erróneamente una clase positiva.

²Un falso negativo ocurre cuando el modelo no detecta una clase positiva que sí está presente.

Problema	Medidas	Descripción	Fórmula	Intervalo	Ideal
Clasificación ³	Accuracy	Proporción de clasificados correctamente sobre el total	$\frac{TP+TN}{TP+TN+FP+FN}$	[0, 1]	1
	False Positive Rate	Proporción de clasificados erróneamente como positivos sobre todos los negativos	$\frac{FP}{FP+TN}$	[0, 1]	0
	Precision	Proporción de clasificados correctamente como positivos sobre todos los clasificados como positivos	$\frac{TP}{TP+FP}$	[0, 1]	1
	Recall	Proporción de clasificados correctamente como positivos sobre todos los positivos	$\frac{TP}{TP+FN}$	[0, 1]	1
	F1	Media armónica de la precisión y el recall	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	[0, 1]	1

³ TP = True Positives; TN = True Negatives; FP = False Positives; FN = False Negatives.

Tabla 4.1: Medidas de desempeño en Clasificación

Problema	Medidas	Descripción	Fórmula	Intervalo	Ideal
Predicción	MSE	Promedio de los errores al cuadrado entre las predicciones y los valores reales	$\frac{1}{n} \sum_{i=1}^n (y_{\text{pred}}^{(i)} - y_{\text{real}}^{(i)})^2$	$[0, \infty)$	0
	RMSE	Raíz cuadrada del MSE. Mide la magnitud promedio de los errores	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_{\text{pred}}^{(i)} - y_{\text{real}}^{(i)})^2}$	$[0, \infty)$	0
	MAE	Promedio de los errores absolutos entre las predicciones y los valores reales	$\frac{1}{n} \sum_{i=1}^n y_{\text{pred}}^{(i)} - y_{\text{real}}^{(i)} $	$[0, \infty)$	0
	R^2	Proporción de la variación en los datos que es explicada por el modelo. Indica la bondad del ajuste	$1 - \frac{\sum_{i=1}^n (y_{\text{real}}^{(i)} - y_{\text{pred}}^{(i)})^2}{\sum_{i=1}^n (y_{\text{real}}^{(i)} - \bar{y})^2}$	$(-\infty, 1]$	1

Tabla 4.2: Medidas de desempeño en predicción

Por tanto, en función del resultado de estas métricas, será necesario optimizar o no el modelo para tener un alto *accuracy* principalmente, pero que no sobreajuste, y que además minimice la pérdida entre datos clasificados y datos reales. No obstante, destacar que el *accuracy* no es suficiente para evaluar el resultado, hace falta también un *recall* alto que indique que los datos son balanceados.

4.1.2. Redes Neuronales

Las redes neuronales artificiales son sistemas computacionales inspirados en las redes biológicas del cerebro humano, conformando un modelo formal, y diseñadas para reconocer patrones y aprender de grandes volúmenes de datos. Para ello emulan la manera en que los seres humanos procesan la información. La neurona es la unidad de procesamiento básica que compone la red neuronal y del mismo modo, se basa en la neurona biológica. Por tanto, para entender el funcionamiento de la red neuronal, es imprescindible conocer el comportamiento de una neurona y el proceso que siguen las redes neuronales biológicas [147, 54, 265, 65, 272, 20, 161].

Una neurona es una célula del sistema nervioso que se encarga de la transmisión de señales eléctricas y químicas entre distintas regiones del cuerpo. Es la responsable de recibir información del mundo exterior y de hacer llegar las señales de la respuesta a los diferentes órganos. Por tanto, permite la comunicación interna entre el cerebro, la médula espinal y el resto del organismo, posibilitando el movimiento humano y el almacenamiento de información. Esta célula nerviosa

consta de cuatro partes fundamentales, cuyo esquema se muestra en la figura 4.2:

- **Cuerpo celular (o soma):** es donde suceden todos los procesos metabólicos de la neurona, o, dicho de otra manera, donde ocurre el procesamiento de la información. En su interior se localiza el núcleo, el cual contiene el material genético (ADN) y, además, es la base central de la neurona, donde se regulan todos los procesos.
- **Axón:** es un tubo que se origina en el soma de la neurona que conduce la señal eléctrica desde el cuerpo de la neurona hasta las otras neuronas.
- **Dendrita:** son prolongaciones surgidas del soma con forma de ramas que cubren el centro neuronal y captan los neurotransmisores producidos por la neurona más cercana, así como de enviar la información química al cuerpo de la neurona, que hace que este se active eléctricamente. Por tanto, captan la información en forma de señales químicas y estimulan al cuerpo para comunicar que la anterior neurona de la red está intentando enviar un impulso.
- **Sinapsis:** ramificaciones ubicadas en la parte terminal del axón que es el punto de comunicación con las dendritas de otras neuronas. Para ello, libera al medio externo los neurotransmisores (señal química) el impulso eléctrico recibido.

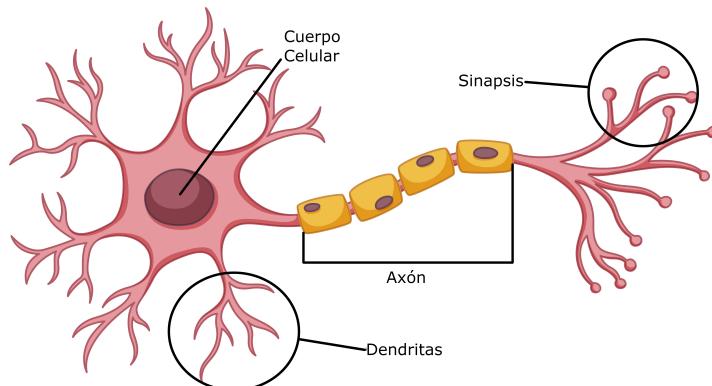


Figura 4.2: Estructura de una Neurona

El proceso de transmisión de la información se inicia en las dendritas, dónde se reciben los distintos impulsos o señales químicas, que se albergan en el soma. Cuando se reúne una cantidad de señales suficientemente fuertes para superar cierto umbral, se genera un impulso eléctrico. Este se propaga por el axón hasta llegar a la sinapsis, la cual convierte la energía en una señal química y se comunica con las dendritas de otra neurona. Así sucesivamente, por lo que al ampliar el enfoque y difundirse esa señal con otras neuronas, nos encontramos ante una red neuronal.

En el ámbito de la informática, las redes neuronales artificiales se inspiran en este principio, donde cada neurona artificial recibe señales de múltiples entradas (equivalentes a las dendritas), las procesa (similar al soma) y, si se supera un cierto umbral, transmite la señal a través de las conexiones (sinapsis) a otras neuronas. Consecuentemente, las neuronas artificiales constan de las siguientes partes [222, 147, 54, 265, 65, 272, 20, 161]:

- **Entradas:** equivalente a las dendritas, representa los valores del conjunto inicial que recibe la neurona artificial. Se encuentran representadas por la nomenclatura x_1, x_2, \dots, x_m .
- **Pesos:** los pesos se encuentran asociados a las entradas (una entrada – un peso) y determinan la fuerza de conexión entre dos nodos, en este caso, entre la entrada y la neurona. Primordialmente trata de controlar el grado de influencia de cada entrada en el resultado final del modelo mediante una ponderación. Finalmente, destacar que se representan por: w_1, w_2, \dots, w_m .
- **Función Sumatoria:** es el análogo al soma y es una función matemática que calcula la suma de las distintas entradas multiplicadas por sus respectivos pesos y un cierto sesgo. Viene dada por la siguiente fórmula:

$$v = \sum_{i=1}^m x_i w_i + b$$

Aunque es susceptible a cambios, no es algo que suceda con frecuencia. Cabe destacar que en el caso de la regresión, $f(X)$ nos da el resultado predicho, dado el conjunto o vector de entrada $X = \{x_1, x_2, \dots, x_m\}$. En el caso de la clasificación, si hubiese solo dos categorías, la clase predicha está dada por:

$$\text{clasificación} = \begin{cases} 1 & \text{si } f(X) > 0 \\ 0 & \text{si } f(X) \leq 0 \end{cases}$$

- **Función de activación:** representada por $f(\cdot)$, es una función no lineal que se aplica al resultado de la función sumatoria para generar la salida final de la neurona. En la tabla 4.3 se recogen las funciones de activación más comunes.
- **Sesgo:** o *bias* (b), controla qué tan predispuesta está la neurona a disparar un valor independientemente de los pesos. En otras palabras, es un parámetro interno dentro de la función sumatoria anterior que influye en la tendencia de la neurona a producir una salida específica. Esto permite que la neurona pueda activarse incluso cuando las entradas sean cero, lo cual es crucial para que la red pueda representar patrones más complejos. Resaltar que cuanto mayor sea el sesgo, más predispuesta estará la neurona a generar una salida positiva y activarse con mayor facilidad.
- **Salida:** finalmente, la salida representada por y , es el resultado final obtenido a través de este método, expresándose de la siguiente forma:

$$y = f \left(\sum_{i=1}^m x_i w_i + b \right) = f(v)$$

Este proceso se puede apreciar en la figura 4.4, donde se muestra el esquema básico de una neurona artificial con estos elementos descritos.

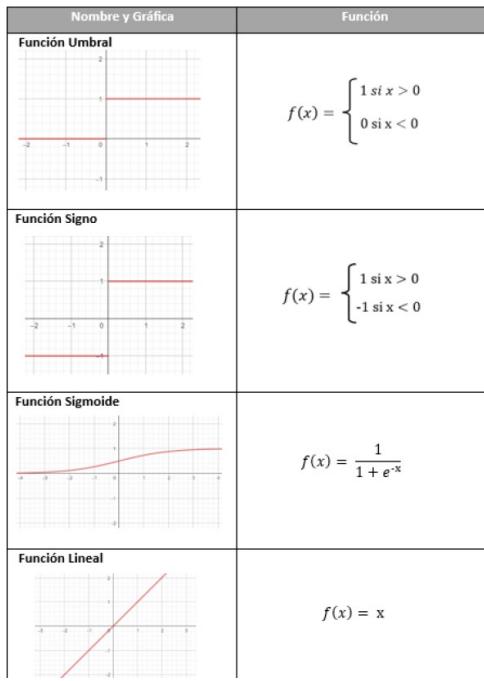


Figura 4.3: Funciones de Activación

Una vez expuesto el funcionamiento de una neurona artificial, se puede proceder a explicar el funcionamiento de la red neuronal. Esta se compone de numerosas neuronas artificiales, las cuales se disponen en distintas capas interconectadas como se muestran en la figura 4.5. Existen por tanto tres tipos diferentes, cada una ejerciendo una función clave para el desempeño global de la red:

- **Capa de entrada:** es la capa inicial de la red, formada por neuronas cuyo objetivo es recibir el conjunto de entrada. En ella los nodos de entrada procesan los datos, los analizan, ponderan y los pasan hacia próximas capas.
- **Capas ocultas:** son capas intermedias que adoptan como datos de entrada el conjunto salida de la capa anterior, que puede ser tanto de capa de entrada o de otras capas ocultas.

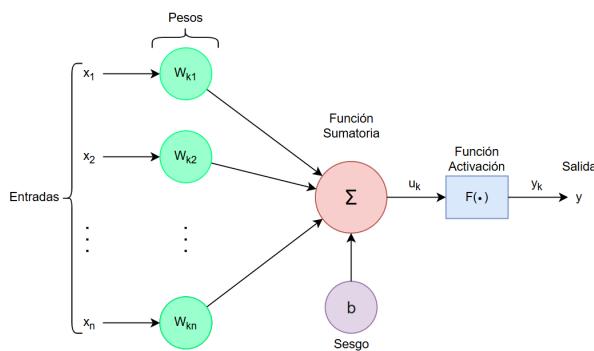


Figura 4.4: Estructura de una Neurona Artificial

Cada una de ellas contiene una serie de neuronas que procesan la información. Resaltar que no interactúa directamente con el exterior, por lo que el número de neuronas y la cantidad de capas intermedias son parámetros a concretar en el diseño y entrenamiento de la red neuronal.

- **Capa de salida:** es la capa final que proporciona el resultado de todo el procesamiento de datos que realiza la red neuronal. Se debe destacar que la solución depende del tipo de problema: en cuestiones de clasificación, la capa tendrá un nodo que devolverá 1 o 0 si es binaria, pero si es multiclase, puede que esté compuesta por más de uno. Es decir, será necesario tantas neuronas como categorías. Al mismo tiempo, si se trata de un asunto de regresión, se manifiesta una única neurona que contendrá el valor numérico final.

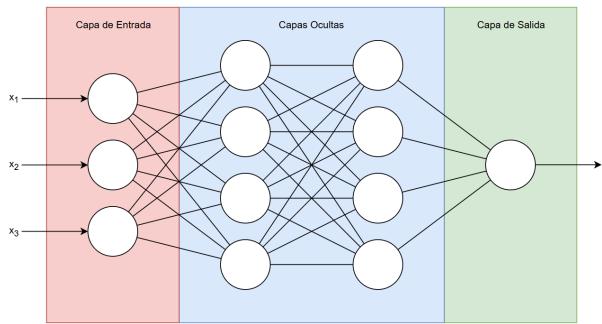


Figura 4.5: Capas de una Red Neuronal Artificial

El entrenamiento de una red neuronal consiste en estimular los distintos conjuntos de pesos y bias de las diferentes entradas que presente cada neurona incluida en la red, y que determinan su salida hacia la siguiente capa de la red. Asimismo, es importante resaltar que la función de activación se acciona cuando se supera un límite de umbral ajustable durante el entrenamiento. Este proceso se centra en minimizar una función de pérdida del dataset, para lo cual existen múltiples técnicas, como el *Hill climbing*, Descenso por Gradiente, Algoritmo de Retropropagación³... Esta función de pérdida cuantifica la diferencia entre las predicciones de la red y los valores reales esperados, donde las principales serían: *MSE*, *Cosine similarity* y *Cross-Entropy*.

Destacar igualmente que durante el entrenamiento se debe evitar el sobreajuste (*overfitting*), ya que esto podría hacer que la red aprenda patrones específicos del conjunto de entrenamiento sin generalizar correctamente a nuevos datos. Para evitarlo, se utilizan técnicas como la regularización (L1-L2), el *Dropout* o el *Early Stopping*.

4.1.3. DeepLearning

Las primeras ideas que implantaron los principios de las redes neuronales surgieron a mediados del siglo XX [220, 269]. No obstante, conforme pasaban los años, aumentaba la cantidad de datos disponibles y la capacidad computacional, lo que impulsó investigaciones en modelos más

³Algoritmo de Retropropagación: utilizado en las RRNN supervisadas para ajustar los pesos de las conexiones entre neuronas con el fin de reducir el error de predicción. Su funcionamiento consiste en calcular el error de salida, propagarlo hacia atrás a través de la red y ajustar los pesos mediante el descenso por gradiente de para minimizar la pérdida

complejos [194]. Fue entonces cuando surgió la necesidad de redes neuronales más profundas, capaces de aprender representaciones jerárquicas y abstractas de los datos.

Este es el núcleo del *Deep Learning*, una rama avanzada del *Machine Learning* que utiliza redes neuronales con múltiples capas ocultas, conocidas como Redes Neuronales Profundas (*DNNs*) [121].

Estas redes profundas son capaces de resolver problemas bastante más complejos, optimizando resultados en ámbitos como el reconocimiento de patrones en imágenes, el procesamiento de texto o la predicción de series temporales [36]. La clave en las técnicas del *Deep Learning* es que son capaces de crear modelos flexibles que explotan la información enterrada en conjuntos de datos masivos de una manera mucho más eficiente que las técnicas clásicas de *Machine Learning* con extracción manual y aprendizaje de características de alto nivel de forma autónoma [286].

Entrenamiento

El entrenamiento de modelos de *Deep Learning* se realiza utilizando grandes conjuntos de datos etiquetados, de forma similar a las redes neuronales tradicionales. El objetivo es ajustar los parámetros (pesos) de la red neuronal para que el modelo pueda hacer predicciones precisas sobre datos nuevos. Este proceso puede llevarse a cabo de dos formas [217]:

1. Desde cero: donde el modelo es entrenado completamente con los datos disponibles.
2. Usando un modelo preentrenado: en este caso, se utiliza un modelo previamente entrenado y luego se adapta o ajusta a una tarea específica mediante técnicas de *fine-tuning*.

En ambos casos, el proceso de entrenamiento es similar, aunque en el caso de usar un modelo preentrenado, no es necesario diseñar desde cero la arquitectura ni decidir el número de capas, neuronas o funciones de activación, ya que esta estructura ya ha sido definida previamente [121].

Pasos Entrenamiento

A continuación, se muestra un proceso de entrenamiento posible: [31],[158], [184, 285, 78, 22, 217, 290]

1. **Inicialización de los parámetros.** Los pesos y sesgos de la red neuronal se inicializan, normalmente con valores aleatorios. Así como la distribución de los conjuntos *test-train-validation*.
2. **Propagación hacia adelante (Forward Propagation).** Los datos de entrada se introducen en la red y pasan capa por capa, aplicando las funciones de activación correspondientes en cada capa para obtener las predicciones finales del modelo.
3. **Cálculo de la función de pérdida (Loss Function).** Una vez obtenida la predicción, se calcula el error mediante una función de pérdida, que mide la diferencia entre la salida predicha y la salida real.
4. **Retropropagación.** El error calculado se retropropaga a través de la red para ajustar los pesos, utilizando distintos algoritmos, como el de descenso por gradiente. Este proceso ayuda a minimizar la función de pérdida [31].

5. **Optimización y Regularización.** Durante el proceso de entrenamiento, se aplican técnicas de regularización para evitar el sobreajuste (*overfitting*), como [158]:

- **Early Stopping:** detiene el entrenamiento cuando el modelo empieza a perder rendimiento en el conjunto de validación.
- **Dropout:** desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para evitar que el modelo dependa de características específicas.
- **Regularización L2:** penaliza valores de pesos excesivamente grandes para evitar que el modelo se vuelva demasiado complejo.

6. **Ajuste de Hiperparámetros.** También se ajustan parámetros clave durante el entrenamiento:

- **Factor de Aprendizaje (Learning Rate):** controla el tamaño de los pasos durante la actualización de los pesos, o en otras palabras, regula cuánto se ajustan los pesos en respuesta al error en cada iteración (por medio del descenso de gradiente). Controla la magnitud de cambio en respuesta al error calculado durante la retropropagación [171].
- **Número de Épocas (Epochs):** es el número de veces que el modelo pasa por todo el conjunto de entrenamiento.
- **Tamaño del Lote (Batch Size):** determina el número de parámetros que entran a un modelo durante el entrenamiento.
- **Ground truth:** se refiere a las etiquetas con las que se califica a los datos de entrada a un modelo, así como las salidas verdaderas.

7. **Evaluación del Modelo.** Una vez finalizado el entrenamiento, el modelo se evalúa utilizando el conjunto de prueba. Las métricas para ello se encuentran ya definidas, pero las más comunes para evaluar el rendimiento del modelo son *Accuracy*, *Recall* y *F1-Score*.

Este proceso se repite durante varias iteraciones (épocas) definidas, ajustando los parámetros hasta que el modelo alcanza un rendimiento adecuado o el *accuracy* o valor de *loss function* que veamos suficiente.

Existen numerosas arquitecturas de DNNs que varían en función de su complejidad y aplicación. Algunas de ellas son más adecuadas para tareas de visión computacional, clasificación de imágenes o análisis de volúmenes de datos masivos. En cambio, para el desarrollo de *chatbots* y asistentes virtuales, como el que se propone en este TFG, existen arquitecturas más específicas que son fundamentales para el procesamiento del lenguaje natural (PLN) y la generación de respuestas, dentro de la IAG. Entre ellas destacan las RNNs y los modelos basados en *transformers*, como GPT y BERT, que han demostrado ser altamente efectivos en tareas de comprensión y generación de texto, permitiendo la creación de asistentes educativos interactivos, capaces de comprender preguntas, generar respuestas coherentes y mantener conversaciones fluidas.

4.2. Arquitectura

A continuación, se presentan algunas de las arquitecturas de *deep learning* empleadas en IAG, con mayor foco en aquellas que permitan la generación de texto, que es en lo que se sustenta este TFG [318, 24, 128, 152, 311, 329, 338, 209, 139, 135, 291, 328].

4.2.1. Redes Neuronales Recurrentes - RNN

Las RNNs son conocidas por su capacidad de procesar y obtener información de datos secuenciales, como texto, audio o series temporales. Su gran diferencia radica en que comparte parámetros al incluir ciclos que conectan nodos adyacentes, lo que permite capturar las dependencias secuenciales y temporales. El impacto notorio de esta red se observaría en el PLN, ya que permite crear una interpretación contextual del lenguaje, entendiendo cómo las palabras previas afectan al significado de las posteriores, algo esencial para tareas como la traducción automática o la generación de texto. La configuración estándar se muestra a continuación:

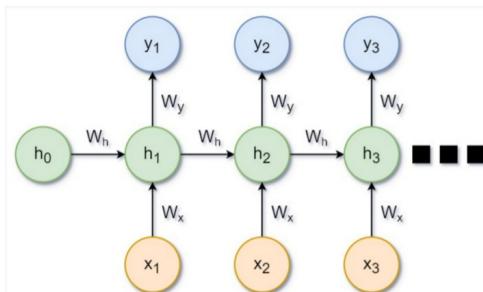


Figura 4.6: Estructura RNN

Sin embargo, se encuentra limitada principalmente por el desvanecimiento del gradiente, dificultando el aprendizaje de dependencias a largo plazo debido a que el gradiente de error se vuelve muy pequeño.

4.2.2. Redes de Memoria Larga a Corto Plazo - LSTM

Las LSTM son un tipo particular de RNNs que solucionan el problema de memoria anterior. Para ello, introduce celdas de memoria que permiten que la red retenga información durante períodos de tiempo más largos, lo que les permite aprender dependencias a largo plazo de manera más efectiva. Su funcionamiento radica en tres puertas principales que regulan el flujo de

información dentro de la celda de memoria: la puerta de olvido (qué eliminar), la de entrada (qué almacenar); y la de salida (qué se utilizará en la salida de la red).

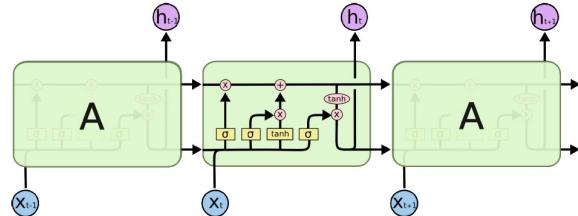


Figura 4.7: Estructura de un LSTM

4.2.3. Autoencoders

Es un tipo de arquitectura de red neuronal no supervisada (no son necesarios datos de entrada etiquetados) que se entrena para replicar sus datos de entrada. Está diseñada para comprimir (codificar) eficazmente los datos de entrada hasta reducirlos a sus características esenciales y, a continuación, reconstruir (decodificar) la entrada original a partir de esta representación comprimida. Por otro lado, se componen de ciertos elementos clave: [159, 81, 216, 279]

- **Codificador:** reduce la dimensionalidad de los datos de entrada a través de capas con menos nodos. Su objetivo es comprimir la información de forma progresiva.
- **Cuello de botella (o código):** es la representación más comprimida de los datos. Sirve de puente entre codificador y decodificador para descubrir las características importantes de los datos de entrada.
- **Decodificador:** reconstruye los datos desde el código comprimido usando capas con un número considerablemente mayor de nodos. Su salida se compara con la entrada original para evaluar el error de reconstrucción y medir la eficacia del *autoencoder*.

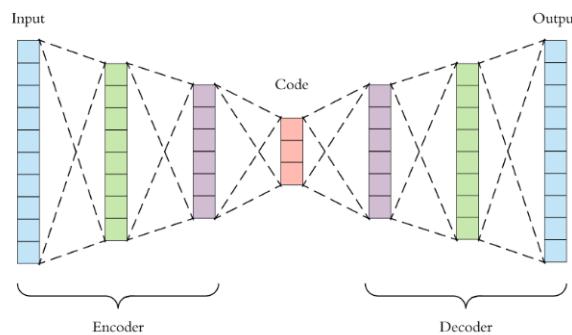


Figura 4.8: Estructura *Autoencoder*

Destacar que estas redes corresponden al tipo *feedforward*, ya que la información se encuentra dirigida en una sola dirección. Además, para optimizar el *autoencoder* existen una serie de hiperparámetros más comunes, como: el número de capas de las redes neuronales codificador y

decodificador, su número de nodos, la función de pérdida que se utilizará en el proceso de optimización (MSE) o el tamaño del espacio latente (cuanto menor sea, mayor será la compresión).

Finalmente, es importante resaltar que a lo largo de los años se han desarrollado diferentes tipos de *autoencoders* principalmente según su aplicación. No obstante, de manera general, su utilización deriva en comprensión o denotación de imágenes y su eliminación de posible ruido, detección de anomalías, predicciones, generación de datos, imágenes o frases de texto nuevo a partir de texto existente y comunicaciones digitales.

4.2.4. Encoder-Decoder

Esta arquitectura es fundamental y muy poderosa en el campo del Procesamiento del Lenguaje Natural (PLN). Se utiliza especialmente en tareas de secuencia a secuencia, como la traducción automática, la generación de resúmenes y la respuesta a preguntas. Su fortaleza radica en su capacidad para procesar y generar datos de forma secuencial.

La arquitectura *Encoder-Decoder* está compuesta de dos componentes principales: el codificador y el decodificador, ambos formados por redes neuronales (RNN, LSTM, GRU, CNN o *Transformers*) que trabajan de manera conjunta pero independiente [124, 182, 160, 196].

1. **Codificador.** Toma la secuencia de entrada (palabras, caracteres u otros) y la procesa para capturar su información esencial, características o significado semántico. A partir de entonces, la transforma en una representación numérica conocida como vector de contexto. Este proceso se realiza mediante:
 - a) **Embeddings** para representar el significado de los *tokens* (estas incrustaciones suelen estar preentrenadas mediante técnicas como Word2Vec o GloVe).
 - b) **Capas recurrentes**, como LSTM o GRU, o **mecanismos de autoatención Self-Attention**, que capturan las relaciones contextuales entre palabras y por tanto, permite encapsular la información de las partes anteriores de la secuencia. Para ello, calcula un estado oculto basado en la entrada actual y el estado oculto anterior.
 - c) **Codificación personal**, que añade información sobre el orden de los *tokens*.
 - d) **Red de propagación hacia adelante**, que refina las representaciones para ser procesadas por las capas siguientes.
2. **Vector de Contexto.** Es una representación comprimida y contextualizada de toda la secuencia de entrada, donde captura los aspectos cruciales de forma condensada. Resume tanto el contenido como el orden de los *tokens* para alimentar al decodificador.
3. **Decodificador** Genera la secuencia de salida (traducción, resumen, etc.) a partir del vector de contexto. Para ello utiliza:
 - a) **Autoatención enmascarada**, que impide el uso de *tokens* futuros para predecir el siguiente.
 - b) **Atención cruzada**, que enfoca el modelo en partes relevantes del vector de contexto al generar cada *token*.
 - c) **Codificación posicional y red de propagación hacia adelante**, que procesan las representaciones en cada paso.

Cada palabra generada depende solo de las anteriores, imitando cómo los humanos entendemos el lenguaje, por lo que funciona de una manera autorregresiva.

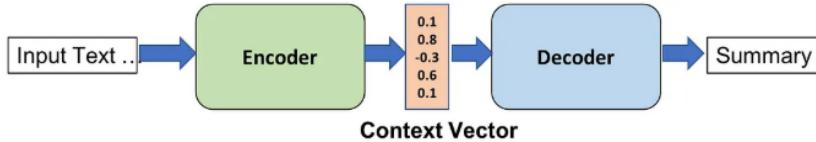


Figura 4.9: Estructura *Encoder - Decoder*

De manera resumida, se puede ver que su funcionamiento se basa en tomar una secuencia de entrada y devolver una secuencia de salida. La cantidad de elementos en la entrada y en la salida puede ser diferente, y los *tokens* posibles en cada lado también pueden variar. Además, esta arquitectura suele utilizarse como base dentro de modelos más complejos, especialmente en tareas de PLN, donde se combina con mecanismos como la atención y redes profundas tipo Transformer.

Mecanismos de Atención de Bahdanau

Dentro de la arquitectura Encoder-Decoder, es importante señalar el Mecanismo de Atención de Bahdanau. Este es una técnica que se emplea en traducción automática, generación de texto, resumen automático o *chatbots*. Su objetivo principal es permitir que el modelo se enfoque dinámicamente en las partes más relevantes de la secuencia de entrada al generar cada elemento de la secuencia de salida. Todo ello para mejorar la comunicación entre el codificador y el decodificador y eliminar ese cuello de botella de información [150, 330, 27].

4.2.5. Transformers

Los *transformers* son producto de la evolución de las arquitecturas anteriores y destacan por su capacidad para modelar dependencias a largo plazo e interacciones complejas en datos secuenciales.

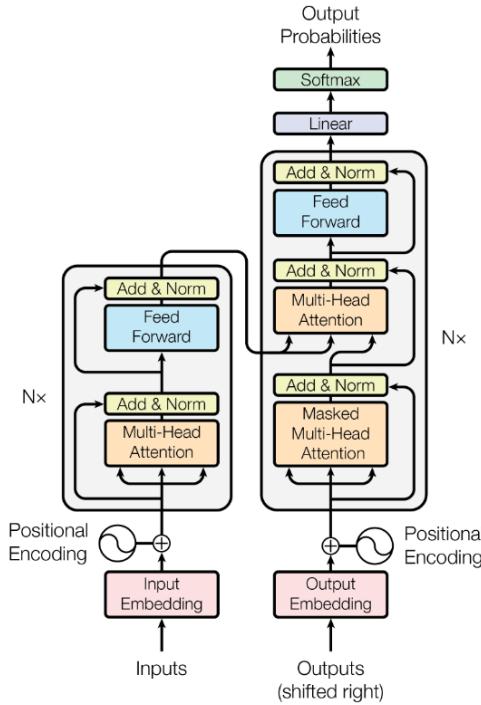
Este tipo de sistema permite alinear ciertas palabras de una secuencia con otras, calculando una representación de dicha secuencia mucho más precisa y eficiente que en anteriores modelos.

Para ello, emplea un mecanismo de atención, donde se asigna un peso a cada palabra de la secuencia, permitiendo que el modelo se enfoque en las partes más relevantes para una tarea específica, sin la necesidad de procesar la secuencia de manera ordenada. Esto permite que los *transformers* manejen contextos más amplios y complejos de manera más efectiva que las RNNs o LSTMs [169, 14, 8, 328, 338, 21, 23, 337, 326, 47, 185, 140, 112, 141, 34, 246].

La arquitectura de los *transformers* se organiza en dos bloques: codificador y decodificador (arquitectura *Encoder-Decoder* vista con anterioridad). No obstante, antes de poder entender su funcionamiento completo, es necesario precisar de unos componentes clave:

Embeddings

Son vectores en un espacio de alta dimensión que representan palabras o fragmentos de texto (*tokens*) permitiéndoles captar el significado tanto semántico como relacional entre las distintas


 Figura 4.10: Estructura *Transformer*

palabras de las frases. Los *embeddings* se generan a través de una matriz de proyección, transformando las palabras en representaciones numéricas que preservan sus características semánticas y contextuales. Entonces, dada una secuencia de entrada $X = \{x_1, x_2, \dots, x_n\}$, cada palabra x_i se mapea a un vector de *embedding* $e_i \in \mathbb{R}^d$, donde d es la dimensión del espacio de *embedding*. Una técnica habitual aparte de los *transformers* para generar *embeddings* es el uso de *Word2Vec*, donde dependiendo de su enfoque o bien predice una palabra basada en su contexto, o viceversa.

Atenciones

Dentro del mecanismo de atención, existen dos tipos principales:

- **Atención Propia Self-Attention:** permite al modelo analizar dentro de una oración la importancia de cada palabra con respecto a otras. En decir, permite a un modelo de red neuronal detectar y ponderar las relaciones entre todos los elementos de una secuencia, sin importar la distancia entre ellos. En general, trata de evaluar el contexto de la sentencia; algo crucial para capturar dependencias a largo plazo. Para alcanzar dicho objetivo, se calculan 3 vectores Consultas (Q), Claves (K) y Valores (V) a partir de los *embeddings* de entrada. Las fórmulas de estas representaciones son:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

Donde X son los *embeddings* de entrada y W corresponden a la matriz de Q,K,V.

Posteriormente, se calcula la atención mediante el producto entre las consultas y las claves,

seguido de la aplicación de la función *Softmax* para obtener una distribución de probabilidad. La fórmula de atención es la siguiente:

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Donde:

- QK^T es el producto punto entre las consultas y las claves.
 - d_k es la dimensión de las claves (y consultas).
 - *Softmax* garantiza que los pesos de atención sean no negativos y sumen 1.
- **Atención Cruzada Cross-Attention:** este caso es más usado para modelos de traducción, donde se alinea palabras de una secuencia de entrada con otras de secuencias de salida. Aquí también el decodificador usa los anteriores valores Q, K, V provenientes del codificador para obtener una distribución de probabilidad aplicando una función *Softmax*:

$$\text{Attention}(Q, K, V) = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Red Neuronal FeedForward (FFNN)

Es un tipo de red neuronal que consta de capas densas donde la información fluye en una sola dirección, sin ciclos ni retroalimentaciones. Por tanto, la salida de cada capa será la entrada de la siguiente. Cabe resaltar que es usada para entrenar algoritmos de aprendizaje supervisado mediante el ajuste de pesos en las conexiones entre neuronas. Esto aumenta considerable la precisión, y por tanto, en el caso de los *transformers*, ayuda a refinar las características extraídas por el mecanismo de atención gracias al procesamiento de los *embeddings* para producir una nueva representación. Básicamente, perfeccionan la representación de cada palabra procesada.

Para ello, después de calcular la atención, la salida pasa por la FFNN constando de dos capas lineales separadas por la función de activación no lineal (usualmente ReLU). La fórmula general para FFNN es:

$$\text{FFNN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Donde W_1 y W_2 son matrices de pesos y b_1 , b_2 son los sesgos de las capas.

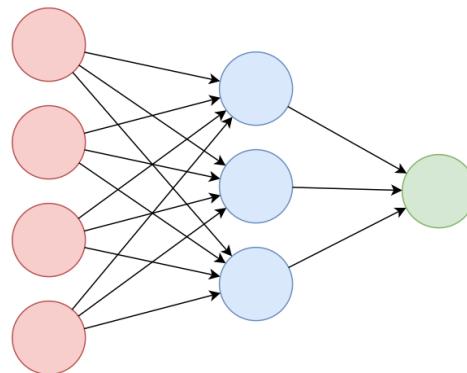


Figura 4.11: Arquitectura de una Red Neuronal *Feedforward*

Mecanismos de normalización y *residuals*

La mejora, estabilidad y aceleración del entrenamiento se consigue mediante distintos mecanismos de normalización, como por lotes o *batch normalization*, y conexiones residuales. Estas facilitan el flujo de la información entre capas, evitando futuros problemas como el de la degradación del gradiente. Además, incorporan estrategias para preservar el orden de las palabras, asegurando que la estructura de la secuencia se mantenga durante el procesamiento del modelo. Normalmente, se utiliza mecanismos como LayerNorm:

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sigma} \cdot \gamma + \beta$$

Donde:

- μ es la media y σ es la desviación estandar de los valores en el vector x .
- γ, β son parámetros aprendidos.

Para facilitar el flujo del gradiente y evitar su degradación en redes profundas, se introducen las conexiones residuales, que permiten sumar las entradas originales a las salidas de ciertas capas. Estas conexiones se basan en la siguiente fórmula:

$$\text{Residual}(x) = x + \text{LayerNorm}(x)$$

Multi-Head Attention

Finalmente este componente es una matriz de atención en paralelo dentro del mismo bloque que permite al modelo aprender y capturar diferentes relaciones semánticas entre palabras. Esto mejora su capacidad de entendimiento y de asimilar dependencias complejas, texto y sentido de la oración.

Para calcular varias cabezas de atención en paralelo por medio de este componente, se describe la operación:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

Aquí, head_i representa cada una de las cabezas de atención calculadas de forma independiente, y la salida final es una concatenación de estas cabezas pasadas por una capa lineal W^O .

Funcionamiento

La arquitectura de los *transformers* se basa en *encoders* y *decoders*, que trabajan de manera conjunta para procesar y generar secuencias de texto. El flujo de procesamiento sería el siguiente 4.13:

En una primera instancia, el codificador recibe la sentencia de entrada y la divide en distintos *tokens*, para luego convertir cada uno de ellos en *embeddings* (que pueden ser preentrenados o aleatorios). A continuación, se aplica junto a la posición de entrada *Self-Attention*, para que el modelo evalúe la significación de cada palabra en relación con las demás. Los vectores resultantes pasan por una red *feedforward* (FFNN) para ajustar sus interpretaciones y mejorar la representación de cada *token*.

Después, ya en el bloque del *decoder*, el decodificador recibe la representación de cada palabra generada por el codificador y comienza a crear una salida secuencial. En este paso, se utiliza

Masked Self-Attention, lo que implica que el modelo solo puede ver las palabras generadas hasta el momento. Al no poder ver las futuras, se evitan “fugas de información” durante el proceso de generación, algo crucial para que el modelo genera la secuencia palabra por palabra sin conocer lo sucesivo.

Más adelante, se aplica *Cross-Attention* para alinear y hacer coincidir las palabras de la entrada con las de la salida. Esto permite determinar las partes más relevantes de la secuencia de entrada al momento de originar su salida. Por último, el decodificador volvería a pasar la información procesada por una red FFNN y generaría la siguiente palabra de la salida. El proceso se repite hasta completar la secuencia final.

Al terminar el proceso de generación en los *transformers*, la salida pasa por una capa lineal, que proyecta las dimensiones de las representaciones a un espacio del tamaño del vocabulario. Luego, haría falta convertirla en una palabra mediante la función *Softmax*:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^V e^{z_j}}$$

Donde:

- z_i es la entrada a la función *Softmax* para la clase i (*logit*).
- El numerador e^{z_i} eleva la entrada z_i a la exponencial.
- El denominador suma las exponenciales de todas las entradas z_j , normalizando los valores para que representen probabilidades.

En los *transformers*, dado que la salida del decodificador es un vector de características h , se puede modelar z_i como:

$$z_i = W_i \cdot h + b_i$$

Ya que la capa final del modelo es una transformación lineal que proyecta h en un espacio de dimensión V , donde cada z_i representa el *logit* asociado a la clase i .

Por tanto, se puede expresar como:

$$\text{Softmax}(z_i) = \frac{e^{W_i \cdot h + b_i}}{\sum_{j=1}^V e^{W_j \cdot h + b_j}}$$

Donde:

- z_i es el logit para la clase (o palabra) i .
- W_i es el vector de pesos asociado con la clase i .
- h es la representación de la secuencia generada hasta el momento (la salida del decodificador).
- b_i es el sesgo para la clase i ,
- V es el tamaño del vocabulario (el número total de clases).

De esta manera, la función *Softmax* convierte los *logits* en una distribución de probabilidad sobre el vocabulario, permitiendo seleccionar la palabra más probable en cada paso del proceso de generación.

La función *Softmax* se aplica a las activaciones de la capa lineal para convertirlas en una distribución de probabilidades, lo que permite seleccionar la siguiente palabra en función de la probabilidad más alta.

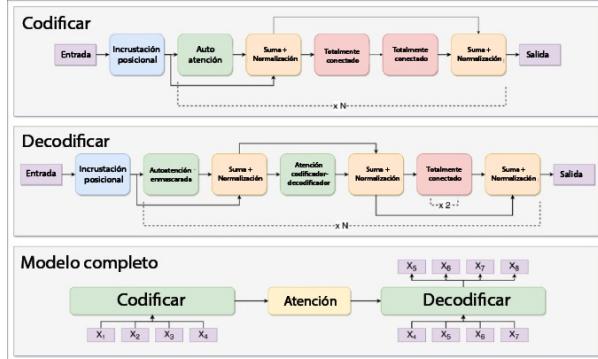


Figura 4.12: Estructura *Transformer* Funcionamiento

Ejemplo

Una explicación visual e interactiva del funcionamiento de los transformadores puede encontrarse en [64], que detalla a la perfección la lógica de su funcionamiento de una manera más clara e intuitiva. Para obtener una mayor comprensión o entendimiento, se recomienda consultar [192], así como el recurso ilustrado de Alammar [15], o directamente el artículo que da pie a su creación: “*Attention is All You Need*” [328].

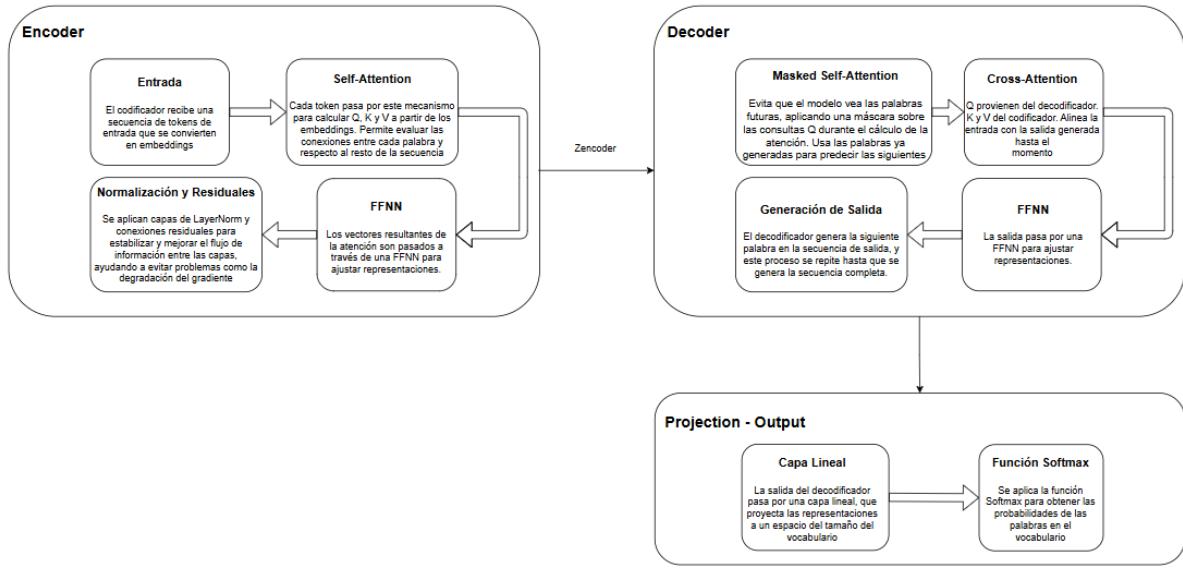
Entrenamiento

El entrenamiento de los *transformers* sigue una lógica similar a su funcionamiento, pero con un enfoque específico en la optimización. Durante el entrenamiento, los *embeddings* de entrada pasan por el codificador y decodificador. La salida generada se compara con la secuencia real a través de una función de pérdida, como *cross-entropy*, que mide la diferencia entre las predicciones y las respuestas correctas. Finalmente, mediante retropropagación, los pesos del modelo se actualizan para minimizar esta pérdida, repitiéndose este ciclo de entrenamiento en múltiples iteraciones para mejorar la precisión y rendimiento del modelo.

Los *transformers*, a diferencia de los modelos tradicionales como RNNs o LSTMs, permiten procesar secuencias de texto en paralelo, mejorando la eficiencia y la capacidad de aprendizaje de dependencias a largo plazo. Asimismo, el modelo puede recibir múltiples entradas, como preguntas y contexto, para alinear la respuesta generada.

Las RNNs y LSTMs han sido fundamentales en el inicio del procesamiento de secuencias y tareas clásicas de traducción automática, generación de texto, reconocimiento de voz y análisis de sentimientos.

Sin embargo, los *transformers* han supuesto un avance considerable, superando muchas de sus limitaciones y revolucionando especialmente el área de procesamiento del lenguaje natural. Alguna de sus aplicaciones abarca tales como clasificación de texto, respuestas a preguntas y


 Figura 4.13: Funcionamiento *Transformer*

chatbots o asistentes virtuales, por lo que son esenciales para el desarrollo de asistentes educativos que requieren comprender y generar respuestas de forma coherente y relevante, como el que se plantea en este Trabajo de Fin de Grado.

4.3. IA Generativa

La IA generativa es un tipo de IA que permite crear nuevo contenido original en forma de texto, imágenes, música, audio o vídeos [61]. Para ello se basa en distintos modelos de ML y DNN, donde aprenden patrones y generan nuevas instancias en función de los datos aprendidos [282, 157, 131]. Sus aplicaciones son muy extensas, pero debido a la naturaleza de este TFG, se resalta la generación de texto mediante el uso de LLMs y por tanto, de *transformers*.

En este sentido, entender cómo estos modelos generan texto requiere comprender las fases que componen su funcionamiento, desde el entrenamiento hasta la producción de contenido personalizado. El funcionamiento general de estos modelos sigue un enfoque común al *deep learning*, que se puede concretar en tres fases principales [157]:

- **Entrenamiento:** Una vez elegido el modelo LLM, este aprende a partir de los datos de entrada. Para ello, se ajustan los hiperparámetros y pesos para minimizar la función de pérdida. Este proceso se realiza de forma iterativa hasta alcanzar un nivel de error aceptable.
- **Ajuste:** Despues del entrenamiento inicial, se realiza el ajuste (o *fine-tuning*) para adaptar el modelo a tareas específicas y mejorar su rendimiento [80, 154]. En esta fase se optimizan nuevamente los hiperparámetros para aumentar la precisión y la calidad de la generación en contextos particulares y que así sea más efecicaz y relevante la respuesta generada.
- **Generación:** Una vez terminados los anteriores, el modelo genera contenido nuevo a partir de una instrucción o prompt que proporciona el usuario. La salida se basa en lo aprendido durante el entrenamiento y ajuste. Por tanto, la calidad de la generación depende tanto del modelo y su precisión como de la claridad del prompt [58]. Bien es cierto que si los resultados no son los esperados, tras una evaluación, se podría proceder a un reajuste de los hiperparámetros. Cabe destacar que dentro de la generación, existe la técnica del *Retrieval-Augmented Generation* (RAG) que permite complementar la generación con información externa al modelo, garantizando respuestas más actuales y específicas. Esta es la técnica en la que se base el presente TFG, la cual se explicará con más detalle en un apartado posterior.

Dentro del contexto del aprendizaje en la educación, es inminente la efectividad e impacto de la IAG como asistente estratégico en el EdTech. Los resultados de diversos estudios han demostrado que el uso de la IAG mejora significativamente el proceso de aprendizaje individual, lo que refuerza la necesidad de asentar esta tecnología desarrollando un asistente de aprendizaje innovador y no antes visto [75, 176, 176]. Además, distintas investigaciones de prestigiosas universidades como Harvard o Standford han concluido que la integración de la IAG en la educación incrementa notoriamente la personalización, mejora la satisfacción estudiantil y optimiza la eficiencia en el estudio, asentando por tanto su potencial en la educación [75, 180].

Gracias a la IAG, se logra una mejor personalización del aprendizaje, adaptando contenidos y metodologías a las necesidades específicas de cada estudiante. Actualmente, ya se utiliza para automatizar la evaluación, facilitar la educación inclusiva, promover la educación continua y apoyar el seguimiento del progreso del alumnado, sirviendo como una ayuda integral para docentes y estudiantes [319, 119, 166, 164].

Por ello, en este proyecto se propone el desarrollo de un asistente de aprendizaje basado en IAG (el cual empleará distintos modelos de DNN y LLMs) debido a su capacidad para comprender

el lenguaje natural y generar texto coherente, útil y relevante [1]. Todo eso para lograr ofrecer una experiencia de aprendizaje personalizada, adaptada al nivel, ritmo y necesidades específicas de cada estudiante, a partir de un contexto educativo concreto y estructurado.

Su uso permitirá crear una herramienta educativa innovadora centrada en mejorar y reforzar el aprendizaje de forma adaptativa, accesible y eficaz. De esta manera, se contribuirá a desarrollar una aplicación práctica con impacto directo en el ámbito educativo.

4.3.1. PNL

Dado que gran parte de la IA generativa de texto se basa en la comprensión y producción del lenguaje humano, es necesario referenciar el Procesamiento del Lenguaje Natural (PLN), disciplina clave que hace posible esta capacidad.

El Procesamiento de Lenguaje Natural (PLN) es un campo de la inteligencia artificial y la lingüística computacional que estudia la interacción entre las computadoras y el lenguaje humano. Su objetivo es lograr que las máquinas comprendan, interpreten y generen texto de manera natural, permitiendo tareas como traducción automática, generación de texto, preguntas y respectivas respuestas, resúmenes y *chatbots*. Esta tecnología no solo está revolucionando la forma en que nos comunicamos con las máquinas, sino también cómo se analizan grandes volúmenes de texto para obtener *insights* valiosos [34, 136, 195, 302].

Dichos objetivos han podido ser abordados gracias a la evolución significativa del PLN en los últimos años, con el desarrollo del ML y DNNs. Estos campos son fundamentales para la construcción de *chatbots* y por tanto, del asistente de aprendizaje de este TFG, ya que permiten que un modelo genere texto de contenido educativo relevante sin necesidad de programar reglas lingüísticas manualmente.

Inicialmente, el PLN utilizaba enfoques basados en reglas y métodos estadísticos, donde se analizaba la probabilidad de aparición de palabras en función de datos previos. Sin embargo, estos procedimientos no captaban el significado profundo del lenguaje ni las relaciones contextuales en los textos largos.

Posteriormente, el uso de RNNs y LSTMs mejoraron el procesamiento de secuencias de texto, facilitando tareas como la traducción automática y el análisis de sentimientos. No obstante, se presentaban problemas relacionados con entender secuencias largas con múltiples palabras, dificultando la comprensión [312, 3, 200, 266, 304, 69, 210, 263, 106].

Finalmente, no fue hasta 2017 cuando se produjo el avance más significativo con la utilización de la arquitectura *transformer* [328]. A raíz del uso del mecanismo de atención, se permite capturar relaciones clave dentro de un texto sin necesidad de procesarlo secuencialmente, logrando una mayor eficiencia y precisión en tareas de comprensión y generación de lenguaje.

El desarrollo de algoritmos de PLN desde cero es un proceso altamente costoso tanto en los recursos de cómputo como en la necesidad de recopilar una cantidad masiva de datos lingüísticos, sus respectivas reglas y su posterior entrenamiento [301, 43]. Por tanto, este proyecto no plantea crear un nuevo modelo, sino aprovechar modelos existentes basados en *transformers*, y explotarlo a su totalidad para optimizar su uso en el ámbito educativo [256, 167].

Así, tras esta evolución tecnológica del PLN, surgen los *Large Language Models* (LLMs) como una de sus expresiones más sofisticadas, y pieza central para el desarrollo de sistemas conversacionales educativos como el propuesto en este trabajo.

4.3.2. LLM

Los *Large Language Models* (LLMs) son modelos específicos dentro del PLN, entrenados con inmensas cantidades de datos para predecir y generar texto de manera coherente. Por tanto, son modelos de inteligencia artificial basados en los *transformers* y diseñados para comprender, generar y manipular lenguaje natural. Debido a su entrenamiento con un extenso *corpus* de texto, son capaces de capturar una amplia gama de conocimientos lingüísticos y contextuales [267, 25, 162].

Los LLMs representan una evolución significativa dentro del PLN. No obstante, a diferencia de los modelos tradicionales, poseen una mayor capacidad de generalización y adaptación a diversas tareas lingüísticas sin requerir un entrenamiento específico para cada una. Además, no se limitan únicamente a la generación de texto, sino que también pueden realizar tareas como el análisis de sentimientos, la traducción automática y la síntesis de información con mayor precisión y coherencia.

Finalmente, gracias a su vasto *corpus* de datos, los LLMs tienen una considerable capacidad para aprender patrones y estructuras complejas del lenguaje, lo que les permite generar respuestas más naturales, fluidas y alineadas con la comprensión humana. La generalización en el uso de LLMs ha impulsado el desarrollo de una elevada variedad de herramientas asistenciales de distintos tipos, como los asistentes de aprendizaje, que son el tema central de este proyecto. Debido a sus características, resultan ideales para la creación de sistemas educativos interactivos que mejoran la experiencia de aprendizaje del usuario [19].

Como bien se ha comentado, los LLMs funcionan gracias a redes neuronales de tipo *transformers*, que procesan texto a través de atención posibilitando al modelo sopesar la importancia de diferentes palabras en una oración, independientemente de su distancia posicional entre sí. Aun así, también están compuestos de distintos elementos fundamentales propios de ellos [59]:

- **Embeddings**
- **Capas de atención:** como bien se definían con anterioridad, son diferentes capas de transformadores que componen los modelos y en donde se asignan distintos pesos a cada palabra en función de su importancia en la frase. Los mecanismos de atención se utilizarían por tanto para comprender el contexto y las relaciones de palabras, todo ello en dentro de un aprendizaje dinámico y flexible.
- **Pre-entrenamiento y Fine-Tuning:** las enormes cantidades de texto que emplean los LLMs, también durante un proceso de pre-entrenamiento, sirven para que aprendan la estructura, matices, contexto, lenguaje y patrones de manera generalizada. Posteriormente, para tareas específicas el modelo puede ser ajustado, como el caso de los asistentes, lo que permite especializar la experiencia sin necesidad de entrenarlo todo otra vez desde cero. A través de un proceso iterativo de exposición y ajuste, los LLM extraen patrones, estructuras y significados del lenguaje natural.

4.3.3. Tipos de LLM

Respecto los tipos que se pueden encontrar, a continuación, se presenta una tabla con alguno de los principales modelos más utilizados actualmente, incluyendo sus características y aplicaciones clave [266, 3, 26, 52]:

(A fecha del 16 de Junio de 2025.)

Modelo	Desarrollador	Entrenamiento	Aplicaciones clave
GPT-4	OpenAI	Pre-entrenado con datos masivos de texto	Generación de texto, traducción automática, chatbots, asistencia en tareas complejas
GPT-3	OpenAI	Pre-entrenado con texto variado	Redacción de textos: resúmenes, respuestas a preguntas
BERT	Google	Pre-entrenado con texto no estructurado	Comprensión y clasificación de texto, análisis de sentimientos
LaMDA	Google	Entrenamiento específico con datos conversacionales	Chatbots, asistentes conversacionales, generación de diálogo
Gemini	Google	Pre-entrenado con grandes volúmenes de texto	Generación de texto, chatbots, respuestas a preguntas, análisis de texto
Codex	OpenAI	Pre-entrenado con datos de código fuente	Generación de código, autocompletado de código, asistencia en programación
Ollama	Ollama	Entrenado con texto variado y datos técnicos	Generación de texto, chatbots, generación de código, asistencia conversacional

Tabla 4.3: Tabla diferentes modelos LLM

4.4. Ingeniería de Prompting

Anteriormente se ha deducido que la calidad de las respuestas generadas dependen directamente de la precisión, detalle, claridad, exactitud y rigor del prompt [156]. Por tanto, antes de exponer la técnica RAG es importante presentar el concepto de ingeniería de prompting.

La ingeniería de prompting es la práctica de diseñar y refinar prompts (preguntas o instrucciones) para obtener respuestas específicas de los modelos de IA [86]. Esta práctica no solo permite mejorar la calidad de las respuestas, sino que también ayuda a comprender mejor las capacidades, limitaciones y el comportamiento de los modelos, como información que puede servir para posibles refinamientos, ajustes o mejoras posteriores [251].

Un prompt bien diseñado garantiza que el modelo comprenda el contexto, los matices y la intención detrás de la consulta, lo cual es esencial para generar respuestas relevantes y de calidad [86]. Además, permite maximizar el rendimiento del modelo, optimizar resultados, reducir errores,

mejorar la eficacia, y ofrecer un mayor control sobre las respuestas generadas [173].

Aunque no existe un prompt perfecto, si es posible perfeccionarlo por medio de la experimentación, creatividad y refinamiento progresivo. Asimismo, existen diferentes elementos clave, pautas y técnicas comunes que permiten construir mejores instrucciones y obtener un buen prompt: [86, 96, 173]

- **Contexto:** facilitar información adicional o de un contexto más amplio puede beneficiar a la IA para contemplar un marco de referencia mayor e incluso mejorar significativamente las respuestas. Sin contexto, las respuestas pueden ser poco precisas y genéricas.
- **Instrucción Clara:** indicar de forma precisa, concisa y clara qué debe de hacer la IA. Para ello, hay que evitar la ambigüedad e imprecisión e intentar ser limitado, específico y dirigido.
- **Datos de Entrada:** contenidos concretos que procesará el modelo.
- **Indicador de Salida:** define el formato o tipo de respuesta deseado, como puede ser en una lista, tabla, texto normal, para ayudar a ajustar el estilo. Resaltar que al proporcionar la salida, es importante siempre revisarla.
- **Ejemplos:** a veces puede servir de ayuda proporcionar ejemplos base que guíen la respuesta, en especial cuando el modelo imite un patrón específico de respuesta.
- **Tono y Estilo:** se define el formato, estilo, tono, grado de profesionalidad, extensión o longitud de la respuesta.

4.4.1. Técnicas Básicas

- **Role Prompting:** consiste en asignar un rol al modelo, lo que le permite adaptar el lenguaje, tono y dominio de la respuesta.
- **Prompting Iterativo:** por medio de ajustes, refinamientos, especificaciones o *feedback*, reformular o aclarar el prompt inicial puede hacer que sea más directo, preciso y específico, lo que mejorará el rendimiento de la respuesta. Es decir, en base a las respuestas del modelo refinarlo gradualmente y de manera iterativa.
- **Prompt Reframing:** en función de los resultados generados por el modelo, realizar distintos ajustes o dictar más especificaciones para crear los siguientes prompts. Esta interacción dinámica garantiza que las respuestas del modelo se ajusten más a las expectativas del usuario con el tiempo.
- **Prompt como plantilla:** crear *templates* o estructuras fijas con vacíos o variables que se modifican conforme lo necesario. Esto permite generar múltiples resultados según un esquema.

4.4.2. Técnicas Avanzadas

En esta sección, se muestran distintas técnicas que se encuentran más relacionadas con el conocimiento interno y con una comprensión más profunda del comportamiento del modelo LLM empleado.

- **Zero-shot prompting:** proporcionar al modelo una tarea o instrucción sin ningún ejemplo o contexto previo, de tal manera que se basa únicamente en su conocimiento y proceso de entrenamiento. Es de utilidad para poner a prueba la capacidad del modelo para generalizar y producir resultados relevantes sin depender de ejemplos anteriores.
- **One-shot y Few-shot prompting:** en función de ejemplos o contexto previos, “aprende de ellos” para guiar la respuesta que genere. Cuantos más ejemplos adecuados se incluyan, mayor precisión puede lograrse.
- **Chain-Of-Thought Prompting:** esta técnica consiste en guiar al modelo a través de una serie de pasos de razonamiento. Al dividir la tarea en pasos intermedios, el modelo puede lograr una mejor comprensión del lenguaje y unos resultados más exactos. Básicamente es solicitar que el modelo muestre su razonamiento paso a paso.

4.5. Retrieval Augmented Generation - RAG

Los LLMs comprenden diversos desafíos: en ocasiones pueden ser imprecisos, inconscientes, inconsistentes, ofrecer respuestas incorrectas, genéricas; incluso inventadas (“alucinan”), poco precisas o no entender bien el contexto. Esto se debe a que están basados principalmente en arquitecturas de tipo *transformer*, lo que implica que no comprenden realmente el significado de las palabras, sino que identifican patrones estadísticos entre ellas. Además, como han sido entrenados con grandes volúmenes de texto estático (por lo que tienen un conocimiento limitado), es posible que no incluyan información específica, correcta o actualizada, o que interpreten ciertos términos de manera errónea según el contexto [85].

Por ejemplo, la palabra “piso” puede significar suelo en Colombia, pero vivienda en España. O “llama” puede referirse tanto a un animal como al fuego. Son casos de homógrafos o palabras polisémicas que el modelo podría interpretar mal si no tiene acceso a un contexto adecuado [307, 201, 294, 94].

Otro ejemplo claro se da en el ámbito de la educación, donde distintas instituciones pueden enseñar los mismos conceptos utilizando terminologías, nomenclaturas o enfoque diferentes. En la UCM, por ejemplo, se enseña QBE (*Query By Example*), mientras que en la UVA se prefiere SQL. Si un estudiante hace una pregunta sobre bases de datos, un modelo sin información contextual podría responder usando QBE, aunque el estudiante esperaba una respuesta en SQL. Lo mismo ocurre con las distintas notaciones del modelo entidad-relación (ER), o en los conceptos utilizados en diferentes asignaturas o facultades. RAG permite superar estas limitaciones.

Retrieval Augmented Generation o Generación Aumentada por Recuperación (RAG), es un enfoque cuyo objetivo es mejorar la calidad de las respuestas generadas por los modelos de lenguaje de gran tamaño (LLMs). Para ello, se basa en recuperar datos desde una fuente de conocimiento externa al entrenamiento del modelo, con el fin de fundamentar las respuestas con información más precisa y actualizada [163, 88, 223, 18, 252, 334, 172]. En lugar de depender solo de los datos con los que fue entrenado el modelo, se conecta a fuentes adicionales, como bases de datos, documentos institucionales, biografías, noticias...etc, para recuperar información pertinente y actualizada. Esto es especialmente útil en contextos donde la información cambia constantemente, como el deporte o la educación. En la figura 4.14, se muestra cómo RAG se puede considerar como una forma de optimizar un LLM [240].

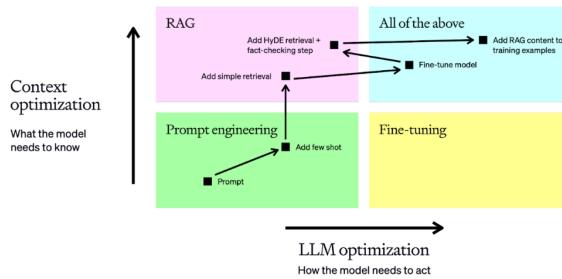


Figura 4.14: Ejemplo de Flujo de Optimización de LLM

Existe el ejemplo de la liga deportiva que quiere implementar un asistente para que sus aficionados puedan consultar datos o dudas sobre los jugadores, los equipos, la historia y las reglas del deporte, así como las estadísticas y clasificaciones actuales. Un LLM general podría responder preguntas acerca de las reglas, historia o distintos partidos; pero en cambio en un partido que se haya jugado el mismo día no podría ofrecer información actualizada sobre el tránscurso de este debido a que no se ha entrenado al modelo con dichos datos. Además, los recursos computacionales para entrenar cada día el modelo pueden estar limitados y entrenarlo cada vez que haya una novedad sería inviable.

En cambio, con RAG, se podría acceder a distintas fuentes, como noticias, artículos o *feeds* que describan ese partido del día. De esta manera, se accede a una información novedosa, actualizada y nueva en la que basar en LLM para permitir una respuesta precisa, relevante, correcta, acertada y apropiada al interés del usuario. Todo ello además, sin tener que reentrenar el modelo.

En el caso del campo de la enseñanza, bien es cierto que cada día que se imparte una asignatura no hay una noticia donde se describa, comente o interprete la situación. En cambio, se pueden emplear los distintos materiales educativos de ese temario a modo de repositorio de conocimiento para cada institución, asignatura o incluso docente. De este manera los estudiantes podrán interactuar con un asistente basado en LLM que responda con el lenguaje, los conceptos y los contenidos específicos que han aprendido en clase y por tanto, que se centren en el temario impartido por el profesor en esa materia. Los resultados generados serán más relevantes, precisas, oportunos, confiables y útiles, lo que permitirá una enseñanza más personalizada, alineada y adecuada con el contexto real del usuario.

A modo de conclusión, RAG amplía las capacidades de los LLMs, permitiéndoles generar respuestas más notables, precisas y verificables, sin necesidad de reentrenar el modelo. Al utilizar recursos externos, se reduce la probabilidad de errores o “alucinaciones”, y se evita el riesgo de que el modelo genere o filtre información sensible. Además, precisa los resultados generados así como su relevancia y ayuda en la personalización del aprendizaje brindando la oportunidad de incidir en los aspectos vistos en clase ese mismo día.

Como afirma Lastras: “Es la diferencia entre un examen a libro abierto y uno a libro cerrado. En un sistema RAG, se le pide al modelo que responda a una pregunta hojeando el contenido de un libro, en lugar de intentar recordar datos de memoria” [191].

En definitiva, RAG no busca “reinventar la rueda” ni trata de volver a entrenar modelos desde cero, sino más bien aprovechar lo ya existente para mejorar la precisión, profundidad y pertinencia de las respuestas generadas. Al proporcionar contexto confiable y específico, RAG mejora significativamente la calidad de la interacción entre el usuario y la IA.

4.5.1. Funcionamiento

Una vez introducido el concepto e importancia del uso de la técnica RAG, a continuación se detalla su funcionamiento en torno a dos fases o aspectos clave que se combinan:

- Recuperación. Esta fase se centra en la recuperación de información de las distintas fuentes externas en las que se base el contexto de la respuesta para un pregunta determinada introducida por el usuario. Para ello busca y selecciona los datos relevantes en la base de conocimiento existente.
- Cabe destacar que la fase de Recuperación hace referencial al ***Retrievial*** del nombre, y Generación al ***Generation***. La palabra faltante, ***Augmented*** 4.15, sería parte del proceso intermedio necesario para pasar de una fase a otra, puesto que sería la ampliación de la indicación o el paso en el que se inserta la información recuperada en el contexto de la pregunta original utilizando además técnicas de ingeniería de prompting para crear una pregunta ampliada. Esto permite al LLM generar una respuesta más precisa e informativa.
- Generación, que es la generación de la respuesta por medio de un LLM basada en la información previamente recuperada en la otra fase.



Figura 4.15: *Retrievial-Augmented-Generation*

A continuación, se muestra el funcionamiento del RAG, además de enfatizar en los distintos componentes claves y necesarios que forman parte del proceso.

Proceso y Componentes

La figura 4.16 ilustra los pasos que se siguen en un RAG:

1. Petición Pregunta. El usuario introduce una pregunta o consulta que se envía al sistema.
2. Recuperación documentos. La pregunta se procesa y se identifican y buscan los documentos relevantes en las bases de conocimiento.
3. Generación de Respuesta. Los documentos filtrados o contexto se concatenan con la pregunta del usuario y se introducen en el modelo de generación.
4. Mostrar Resultado. Finalmente, se generaría y mostraría al usuario la respuesta resultante a su pregunta inicial.

Sin embargo, en todo este proceso existen una serie de matices clave para su funcionamiento óptimo. Por tanto, se destacan los siguientes puntos:



Figura 4.16: Proceso General RAG

Crear datos externos - Incrustación en bases de datos vectoriales

Antes que nada, es fundamental disponer de datos externos que sirvan como base para las respuestas generadas por el sistema RAG. Como bien se ha mencionado previamente, este conjunto de datos ha de ser nuevo con respecto a los del entrenamiento original del modelo de LLM y contener la información relevante y específica en la que fundamentar las respuestas, es decir, el dominio en cuestión. Dichos datos pueden provenir de distintos orígenes, como API, bases de datos o repositorios de documentos. En nuestro contexto, derivarán de los materiales de la asignatura (se profundizará en ello más adelante). Por tanto, para simplificar el enfoque, nos centraremos en que estos son documentos en formato PDF y, por tanto, texto.

Una vez recopilados los distintos documentos fuente, es esencial estructurarlos de manera uniforme para que el sistema pueda realizar búsquedas eficientes y relevantes. Una estrategia común es almacenarlos en una base de datos. Sin embargo, en lugar de guardar el texto en bruto tal cual o simplemente las palabras, se transforma en una representación semántica mediante *embeddings*. De esta manera, se almacenarían estas representaciones vectoriales, las cuales permiten realizar búsquedas por similitud semántica, lo que mejora notablemente la precisión de la recuperación de información.

Para esto, se utilizan bases de datos vectoriales [178, 62, 63, 82], como FAISS, Weaviate o ChromaDB, que permiten indexar y recuperar rápidamente los fragmentos de texto más relevantes. Estas bases construyen un índice vectorial a partir de la fragmentación previa de los documentos (por ejemplo, por párrafos, frases o secciones), generando *embeddings* para cada fragmento (*chunk*) con modelos entrenados para ellos y almacenando dicho resultado junto con metadatos que faciliten la recuperación posterior.

El proceso de fragmentación (*chunking*) es una etapa crítica, ya que influye directamente en la calidad del resultado en la fase de recuperación. Los *chunks* son las unidades básicas de texto que se transforman en *embeddings* y estas representaciones vectoriales son las que se almacenan en la base de datos vectorial junto al texto original y posibles metadatos.

Cabe destacar que el contenido inicial se ha de fragmentar de manera coherente, estructurada y manteniendo el contexto necesario en cada unidad semántica, para así recuperar mejor la información [134]. Esto se refleja en diversos trabajos [97, 118, 179], donde se concluye que las distintas estrategias de *chunking* elegidas, así como las unidades de información indexadas en la base de datos vectorial, son un factor clave en la recuperación y rendimiento del RAG: si los *chunks* son demasiado grandes, pueden contener información irrelevante o redundante; si son demasiado pequeños, pueden perder contexto y coherencia. [134, 292, 293]

Una fragmentación adecuada debe lograr un equilibrio entre los siguientes factores [276]:

1. División de conceptos: cada *chunk* debe centrarse en un concepto, para que la información sea relevante y evite el ruido existente.
2. Preservación del contexto: los *chunks* deben conservar suficiente información contextual para que el modelo LLM pueda comprender el contenido recuperado.

3. Eficiencia en la recuperación: una segmentación adecuada y coherente conlleva búsquedas más rápidas y óptimas.

Paralelamente, existen diversas estrategias de fragmentación, donde las más empleadas serían [12, 179, 55]:

- Tamaño Fijo (*Fixed-size chunking*): divide el texto en segmentos uniformes de bloques de una longitud determinada, ya sea por número de *tokens* o palabras. Aunque sea simple, su problema deriva en que puede dividir conceptos.
- Delimitadores (*Delimiter-based chunking*): se basa en el uso de distintos separadores, como saltos de línea, párrafos, puntos, comas o seccionares como lugar de corte.
- Recursiva (*Recursive chunking*): a partir del caso anterior, subdivide iterativamente el texto en función de separadores hasta que el bloque generado se encuentra en un límite deseado. De esta manera, se adapta mejor al texto, aportando flexibilidad y coherencia semántica.
- Contextual o Semántico (*Semantic chunking*): a raíz de modelos, técnicas o reglas del PNL segmenta el texto en función de frases de su contenido temático, buscando mantener la integridad del mensaje.
- Híbrida: finalmente, existiría un enfoque híbrido que combinaría los distintos enfoques.

Existen otras perspectivas, como la de proposiciones introducido por Chen, pero que no es relevante para este TFG [55].

Finalmente, mencionar que adicionalmente a la fragmentación, existen aspectos importantes que también son significativos para una óptima recuperación de la información. Es es el caso de [118]:

1. Metadatos: para cada *chunk*, se enriquece el fragmento con información propia del documento como el número de página, nombre, categoría, autor... Para luego poder filtrar por estas secciones.
2. Metadato artificial: además de extracción de metadatos del propio documento, es posible añadir información adicional como un resumen, preguntas adicionales o en el caso del *framework* de este TFG, las historias de aprendizaje que abarca cada *chunk*.
3. Índice estructurado: se refiere a una organización jerárquica o lógica de los *chunks* y sus *embeddings*. No todos los sistemas de recuperación vectorial utilizan un índice plano; algunos *frameworks* modernos permiten construir estructuras más avanzadas (como árboles de decisión o índices jerárquicos), lo que mejora la precisión al seleccionar subconjuntos de documentos relevantes antes de realizar una búsqueda más profunda. Esta organización puede facilitar tanto la eficiencia computacional como una recuperación más precisa al contextualizar mejor la información.

Existen distintas herramientas y bibliotecas como Hugging Face, LangChain o LlamaIndex que permiten realizar estas labores de fragmentación, asignación de metadatos e indexado estructurado de forma eficiente y escalable. Estas librerías automatizan tareas como la fragmentación recursiva basada en separadores, la detección de límites semánticos o el enriquecimiento con metainformación adicional [214].

En resumen, la combinación de una buena estrategia de segmentación, la conversión a *embeddings* y el uso de una base de datos vectorial bien indexada, permite a los sistemas RAG (en el *retrieve*) recuperar la información más relevante de forma rápida y precisa, sentando así una base sólida para la generación de respuestas fundamentadas.

Recuperar información relevante - Consultar la información pertinente (*Retrieve*)

Una vez que se dispone de la base de datos vectorial correctamente construida, el siguiente paso es recuperar la información pertinente y clave para la pregunta del usuario. En otras palabras, se presenta la fase *Retrieve*.

Para ello, la pregunta del usuario se convierte también en un *embedding* o representación vectorial, utilizando el mismo modelo empleado previamente para indexarlos. De esta manera, se garantiza que ambos comparten el mismo espacio semántico, lo que permite comparar semánticamente todos fragmentos indexados en la base vectorial con la pregunta del usuario [198]. Para ello, emplea una función de similitud, como la del coseno, producto punto, distancia Euclídea o normal L2 [165, 7, 259]. Como resultado, se obtienen los k fragmentos más similares y por tanto, relevantes para responder la pregunta.

Filtrado y Clasificación

Cabe destacar que, además de esta búsqueda por similitud semántica, los sistemas RAG pueden también incorporar distintas técnicas de filtrado para reducir el cuerpo de búsqueda, como el filtrado de metadatos o búsqueda por palabras clave. En el caso de filtrado por metadatos, se podría acotar la búsqueda a fragmentos pertenecientes a cierto tema, historia de aprendizaje u objetivo [123].

Asimismo, existe la posibilidad de aplicar sobre los fragmentos inicialmente recuperados una segunda etapa de reordenamiento *re-ranking*. Esta etapa principalmente opcional, aunque efectiva, trata de reevaluar los distintos fragmentos candidatos por medio de un modelo más profundo o un *cross-encoder*, que analiza la relación entre la pregunta y cada fragmento a nivel textual completo, y no solo mediante *embeddings*. A raíz de ello, se mejora la precisión y relevancia del resultado final [310, 115].

Este proceso a la hora de implementarlo se suele gestionar a través de un módulo *retriever* conectado al *vectorstore* (base de datos vectorial). En el caso específico de bibliotecas como LangChain, existe la función `get-relevant-documents()` que automatiza internamente esta lógica de consulta semántica, retorno top-k y aplicación de filtros si se configuraran [270].

Cabe destacar que en esta etapa únicamente se buscan y recuperan los fragmentos de texto más relevantes, sin generar todavía ninguna respuesta. Se centran en encontrar el contenido adecuado sobre el que el modelo generativo se apoyará más adelante para construir la respuesta final. Es decir, el contexto. Esta separación entre recuperación (*retrieve*) y generación (*generate*) es una de las claves para un funcionamiento óptimo de RAG.

Ampliación del prompt (*Augment*) Aumentar la solicitud de LLM

Para mejorar la calidad y precisión de las respuestas generadas, el modelo RAG amplía la entrada (el prompt) del usuario agregando los fragmentos de texto relevantes previamente recuperados (el contexto) en el *Retrieve*. Esta técnica permite que el modelo de lenguaje disponga

de la información necesaria para generar respuestas fundamentadas y contextualmente coherentes [198, 175].

Asimismo, existen más enfoques donde se aplican métodos de ingeniería de prompting para guiar al modelo y mejorar las respuestas. Entre ellos se incluyen tales como la estructuración del mensaje (introducción, contexto, pregunta), roles, instrucciones explícitas o formato de la salida [203, 202], además de las ya vistas en el apartado 4.4.

Por tanto, el prompt final que recibiría el LLM es una combinación de la pregunta del usuario, el contexto recuperado y elementos adicionales de la ingeniería del prompting para maximizar la calidad y utilidad de la respuesta final.

Generación de respuestas (*Generate*)

Finalmente, el modelo de LLM generaría la respuesta final empleando el prompt Enriquecido construido en la fase anterior.

Es necesario destacar que el comportamiento, calidad y precisión de la respuesta final dependerá en función del modelo utilizado (como GPT-4, Mistral, Claude...). Además, otros parámetros también influyen en el estilo y creatividad del texto generado, como la temperatura, top-k o longitud máxima de respuesta [46, 238]. Asimismo, en implementaciones más avanzadas, pueden aplicarse técnicas de post-procesamiento, como el filtrado de respuestas, la detección de alucinaciones o la verificación de citas [117].

Con la etapa *Generate* culmina el ciclo del sistema RAG, proporcionando al usuario una respuesta Enriquecida, generada a partir de conocimiento específico recuperado y adaptado por el modelo de lenguaje.

Consideraciones: Actualizar datos externos

Por último, cuando se dice que RAG realiza “recuperación dinámica”, se refiere a que no depende únicamente del conocimiento entrenado del LLM, sino que consulta una fuente externa actualizable en el momento de la consulta [198]. Esta fuente puede ser dinámica (por ejemplo, APIs o repositorios online), o estar ya construida (por ejemplo, un índice vectorial basado en PDFs), y la recuperación no necesita reconstruirse en tiempo real.

Aunque en este caso no se realiza una recuperación dinámica en tiempo real desde APIs o repositorios externos, el sistema sigue un enfoque RAG, ya que el modelo generativo se alimenta de un índice vectorial previamente procesado. Esto permite mantener el conocimiento actualizado sin necesidad de reentrenar el modelo base y su recuperación contextualizada permite fundamentar la generación con información precisa y adaptada [175].

4.5.2. Arquitectura

A partir de los componentes descritos, se obtiene la arquitectura representada en las figuras 4.17 4.18. En ella se observa el funcionamiento general del sistema: el usuario introduce una consulta, y a partir de esta se recuperan los fragmentos más relevantes de los documentos existentes en la base de conocimiento. Estos fragmentos se utilizan como contexto, que se incorpora junto a la pregunta original para que el modelo LLM pueda generar la respuesta final.

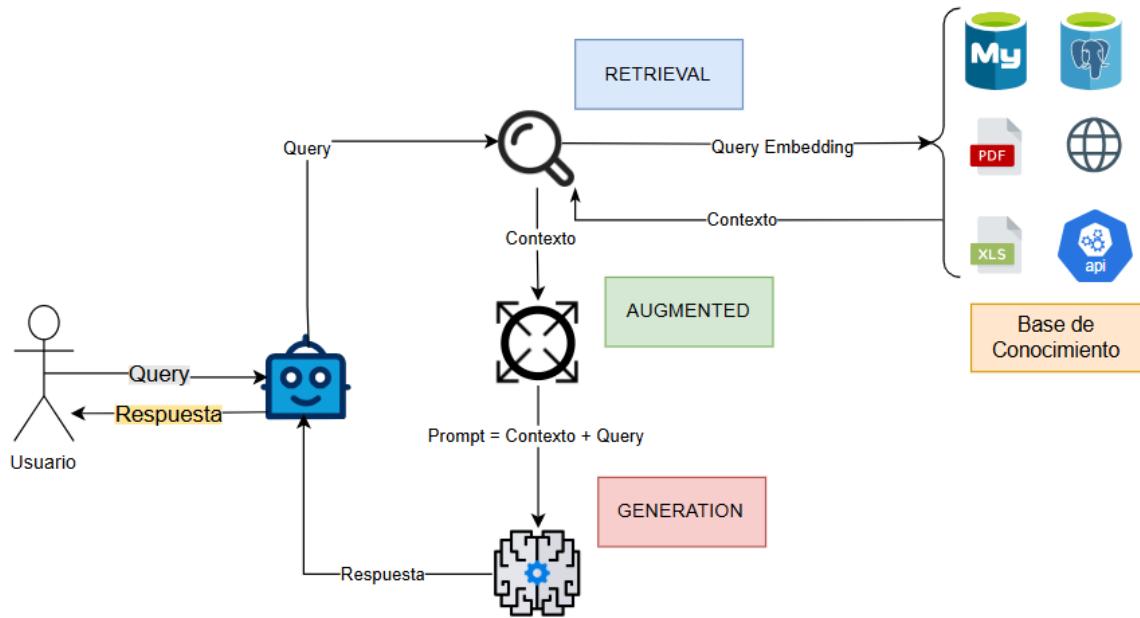


Figura 4.17: Arquitectura RAG

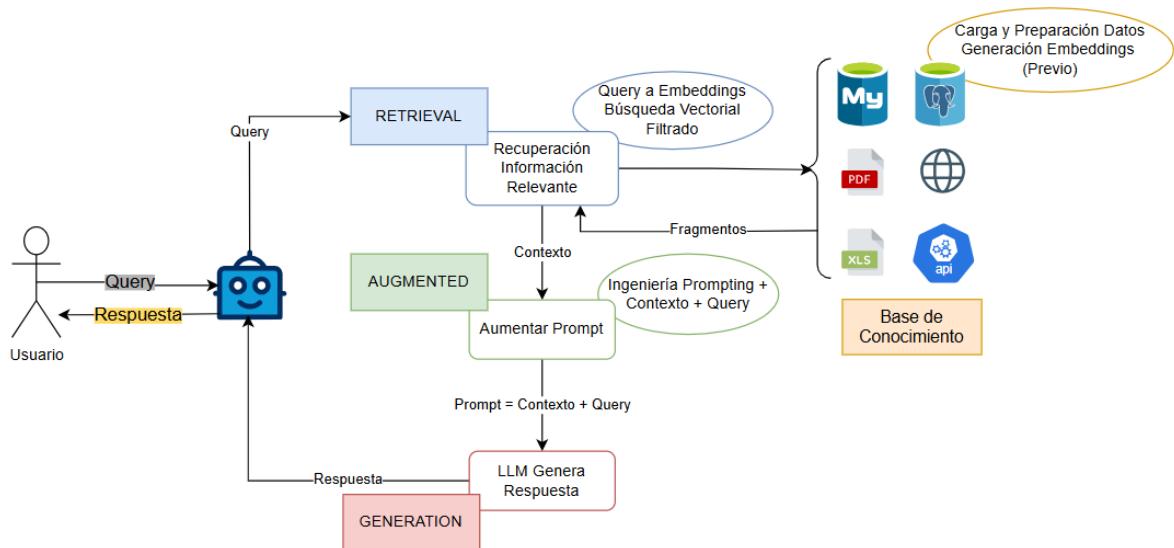


Figura 4.18: Flujo RAG

4.5.3. Ventajas/Beneficios

RAG aporta diversos beneficios dentro de la IAG, así como distintas aplicaciones:

Ventajas

Sus ventajas principales son [103, 198, 223, 18, 88, 163, 244, 85]:

- **Acceso a información actualizada:** permite complementar el conocimiento entrenado

del LLM con datos recientes o específicos del dominio, evitando así respuestas desactualizadas. Por tanto, proporciona la información más reciente y de manera dinámica.

- **Implementación rentable:** al no requerir el reentrenamiento del modelo con nueva información más reciente, se reduce el coste computacional, se facilita su mantenimiento y es posible generar una respuesta más actual y contextualizada. Además, requiere menos cómputo y espacio de almacenamiento, ya que no es necesario tener un modelo LLM propio ni gastar tiempo, recursos y dinero en entrenarlo.
- **Mayor control para los desarrolladores:** es posible definir las fuentes de conocimiento utilizadas, restringir el acceso a documentos sensibles o cambiar dinámicamente los contenidos según el caso de uso.
- **Mayor confianza para los usuarios:** el sistema puede citar los fragmentos utilizados en la respuesta, proporcionando transparencia y trazabilidad. Esto refuerza la fiabilidad del sistema al permitir verificar el origen de la información. Asimismo, como se conoce el origen de los datos, se puede corregir o suprimir la información incorrecta de la RAG y responder con mayor precisión a consultas ambiguas.
- **Respuestas más personalizadas y relevantes:** la recuperación semántica permite adaptar el contenido generado a las necesidades contextuales y específicas del usuario.
- **Mitigación de errores y alucinaciones:** al fundamentar las respuestas en textos concretos, se reduce la probabilidad de que el modelo genere respuestas incorrectas o inventadas. El sesgo puede ser un problema en cualquier herramienta creada por el ser humano. Al confiar en fuentes externas verificadas, la RAG puede ayudar a reducir el sesgo en sus respuestas.

4.5.4. Limitaciones RAG

Por el contrario, son varias las limitaciones y retos que presentan: [85, 244],[307, 201, 294, 94, 118, 104, 327, 50]

- Ambigüedad semántica: los sistemas RAG pueden confundir términos polisémicos.
- Riesgos de privacidad: el uso de datos sensibles en las bases de conocimiento puede exponer información privada si no se implementan medidas de seguridad adecuadas.
- Complejidad en la integración: modelar y combinar datos estructurados y no estructurados en una base de datos vectorial requiere un diseño cuidadoso y puede ser técnicamente desafiante.
- Escalabilidad y rendimiento: a medida que aumenta el volumen de datos y usuarios concurrentes, mantener tiempos de respuesta bajos y una alta disponibilidad puede ser complicado.
- Costes operativos: aunque RAG evita el reentrenamiento completo de los modelos, su infraestructura puede incrementar los costes. Además, la actualización dinámica y automática de fuentes (como nuevas versiones de documentos) sigue siendo un reto técnico y de diseño para sistemas que requieren información constantemente renovada.

- Evaluación del rendimiento: medir la efectividad de un sistema RAG puede ser complejo, ya que las métricas tradicionales no siempre capturan la interacción entre la recuperación y la generación.
- Persistencia de alucinaciones: aunque RAG puede reducir las respuestas inexactas de los modelos de lenguaje, no las elimina por completo, y los modelos pueden seguir generando información errónea si interpretan mal los datos recuperados.
- Frente a Fine Tuning: RAG puede no ser suficiente en tareas muy especializadas o técnicas, donde un *fine-tuning* directo del modelo puede ofrecer mejores resultados en términos de coherencia y estilo. Sin embargo, este último requiere mayores recursos computacionales y datos etiquetados.

4.5.5. Asistente de Aprendizaje

Los modelos de RAG son versátiles y se pueden aplicar a una amplia gama de tareas de PNL. No obstante, no solo comprenden el procesamiento de texto, si no que también es útil en imágenes (por ejemplo, preguntas sobre imágenes de radiografías donde primero se extraen textualmente y luego sigue con el procedimiento de RAG). Entre sus numerosas aplicaciones destaca por [223, 85]: resumen de textos, recomendaciones personalizadas, inteligencia empresarial, búsqueda y recomendación de información, análisis de datos y *chatbots* y asistentes virtuales.

De las numerosas aplicaciones de RAG, una de ellas es en *chatbots* o asistentes virtuales. Un asistente (de IA) es una aplicación de *software* impulsada por inteligencia artificial que entiende el lenguaje natural, procesa comandos y realiza tareas específicas para ayudar a sus usuarios [278].

En el ámbito educativo, al inicio de este TFG se veía una notable falta de la personalización en la enseñanza. Por ello, este TFG propone desarrollar un asistente de aprendizaje basado en RAG, que permita ayudar a los usuarios, en este caso estudiantes, ofreciendo respuestas adaptadas a sus necesidades, capacidades o ritmo y fundamentadas en materiales específicos.

Por tanto, el uso de este asistente de aprendizaje con RAG permite proporcionar una experiencia formativa más personalizada y precisa para el estudiante que permita solventar ese problema inicial.

4.5.6. Fundamentos Técnicos

A continuación, se presentarán las distintas herramientas para integrar de manera coherente los diferentes componentes del RAG. En este proyecto, el objetivo es construir un asistente basado en esta técnica que funcione de forma precisa, modular y accesible.

Lenguajes de Programación

Actualmente existen varias opciones para el desarrollo de asistentes basado en RAG, pero Python es el lenguaje predominante debido principalmente a su ecosistema, paquetes y librerías especializado en IA, PNL y procesamiento de datos. Además, es popular, versátil y compatible con distintos *framework* y herramientas del flujo de RAG.

Otros lenguajes como JavaScript tienen menos soporte para estos casos, aunque pueden usarse para interfaces [243, 68, 313, 305].

Modelos LLM

Por otro lado, es necesario un modelo LLM que genere las respuestas. Como se apreciaba en el apartado 4.3.2 hay una gran variedad de tipos. Bien es cierto que se destacan debido a su uso los GPT de OpenAI y Ollama [88, 221, 241, 241, 142]:

- OpenAI: mediante API, proporciona modelos como GPT-4, GPT-3.5, GPT-4...
- Ollama: plataforma emergente donde sus modelos OLLAMA se pueden acceder tanto por API como de forma local.
- Otros: destaca la plataforma Hugging Face, que alberga una gran cantidad de modelos de distintos tamaños y capacidades, *open source* para descargar, de uso local, y también por API. Resaltan Mistral y BERT.

Asimismo, resaltar la aleatoriedad o temperatura a emplear, donde una baja limita el modelo a las palabras y frases con mayor probabilidad, lo que resulta en respuestas más precisas y seguras, pero menos creativas. Y por el contrario una alta permite que el modelo explore opciones menos probables, lo que puede generar respuestas más originales y variadas, pero también puede introducir errores o incoherencias.

Tecnologías para bases de conocimiento y almacenamiento e indexación y búsqueda

Es necesario unas bases de conocimiento donde almacenar los datos externo que empleará el RAG. Las opciones más comunes para nuestro contexto (interno y documentos en pdf) son [83, 67]:

- Base de datos relacionales, como SQL o PostgreSQL, utilizadas para almacenar metadatos o documentos estructurados.
- Base de datos no relacionales NoSQL, como MongoDB, que permiten almacenar documentos no o semi-estructurados y facilitan la escalabilidad horizontal.
- Base de datos vectoriales: diseñadas para almacenar, indexar y gestionar representaciones vectoriales, lo que permiten una búsqueda óptima de los *embeddings*. Ejemplos de ella son FAISS, Milvus, Pinecone o ChromaDB [79, 247].

Cabe resaltar que todas ellas permiten guardar tanto el contenido textual como su *embedding* y su referencia. No obstante, para desarrollar el asistente, se va a optar por una base de datos vectorial, ya que es la opción más común. Esto se debe a que permite realizar búsquedas por similitud, lo cual es fundamental cuando el usuario formula preguntas, ya que mejora significativamente la recuperación de información relevante [153].

Asimismo, dentro del almacenamiento vectorial es relevante comentar las diversas soluciones en relación a la indexación y búsqueda vectorial [264].

1. FAISS: es una biblioteca *open source* desarrollada por Facebook AI Research para búsqueda y eficiente en espacios vectoriales de cualquier tamaño. Resaltar que es de gran usabilidad, especialmente para búsquedas a gran escala.

2. Pinecone: es un servicio gestionado por nube que ofrece indexación y almacenamiento. Destaca por sus rápidas búsquedas y permitir la simplificación de la infraestructura.
3. ChromaDB, de código abierto, ofrece persistencia local con integración sencilla con *frameworks* de LangChain, manejo eficiente de índices sencillo y óptimo almacenamiento. Sobresale por su escalabilidad, fácil utilización y gran soporte.

Por último, resaltar que no es compatible emplear dos soluciones distintas, a no ser que se combine manualmente. Por ejemplo, si se utiliza ChromaDB con LangChain, no se emplea FAISS a menos que se configure explícitamente un vectorstore con FAISS y se integre manualmente.

Tecnologías para generación *embeddings*

Finalmente, se encuentran diversos medios por el cual generar la fragmentación y *embeddings* a partir del texto de los documentos.

- Fragmentación [87]
 - pdfplumber o PyMuPDF: Para la extracción de texto de documentos PDF.
 - NLTK: conjunto de bibliotecas para el PLN que permiten el procesamiento y tokenización de los documentos.
 - SpaCy: librería de código abierto para el PLN que también permiten la tokenización, procesamiento y tratamiento de los documentos, entre otras funcionalidades que conlleva.
 - A esto hay que sumarle la lógica anterior descrita de aplicar las diversas técnicas posibles de fragmentación (para las cuales proporciona Python distintos métodos o incluso, los propios SpaCy y NLTK).
- *Embeddings* [317, 33, 239, 336]:
 - OpenAI: dentro de OpenAI se encuentran modelos que permiten generar *embeddings* de alta calidad también por API, como puede ser text-embedding-3-small o text-embedding-ada-002. Al encontrarse alojados en la nube ofrecen escalabilidad y acceso a modelos constantemente actualizados, pero depende de conexión a Internet y son costosos.
 - Hugging Face Transformers: como se mencionaba anteriormente en esta plataforma albergan distintos modelos, aunque múltiples de ellos sirven para generar *embeddings* y se pueden ejecutar de manera local. Destacar la familia de Sentence Transformers (un módulo de Python para acceder, usar y entrenar modelos dedicados a embeddings y *reranking*). Entre ellos all-MiniLM-L6-v2. O de igual manera, el de bge-m3 de BAAI.
 - Otros: existen otros modelos de *open source* como fastText, GloVe o componentes de spaCy, o también dentro de Hugging Face, que no son tan conocidos pero que pueden ser útiles en aplicaciones con menores requerimientos o cuando se busca evitar el uso de servicios externos.

Frameworks

Por otro lado, hace falta coordinar óptimamente el flujo del RAG. Para ello, existen diversos *frameworks* que ayudan a la construcción de sus distintas fases (carga y procesamiento de datos, generación de *embeddings*, almacenamiento vectorial, recuperación y consulta a modelos LLM) [151, 149, 229, 187, 132, 236]:

- LlamaIndex: facilita construir pipelines RAG de forma eficaz y es eficiente para grandes volúmenes de datos. Integra componentes como VectorStoreIndex para almacenamiento vectorial (además de bases vectoriales como Chroma o FAISS) y es compatible con modelos de *embeddings* de Hugging Face (como `bge-small-en-v1.5`). Además, destacar que puede emplearse junto a plataformas como Watsonx para integrar LLMs.
- LangChain: *framework* muy extendido, popular y compatible con múltiples LLMs (tanto *open source* de Hugging Face como los de OpenAI) y almacenes vectoriales como FAISS, ChromaDB o Pinecone. Además, tiene una estructura modular que permite enlazar las distintas fases del RAG fácilmente, una comunidad activa y ofrece distintas herramientas personalizadas que lo convierten en una opción muy flexible y extensible.
- Haystack : orientado a sistemas de pregunta-respuesta, soporta *embeddings* como Sentence-BERT y almacenamiento en FAISS, Elasticsearch o Milvus. Aporta buenas herramientas de evaluación de resultados, despliegue de producción y una arquitectura sólida y extensible.

Interfaz y Despliegue

Para concluir, para el desarrollo del asistente de aprendizaje se ha optado por una interfaz sencilla, prototípica y fácil de desplegar. Entre las opciones disponibles están Django, Streamlit, Gradio y React [143, 227].

Django es robusto y completo, ideal para proyectos con gestión de usuarios, pero más complejo. Gradio y Streamlit facilitan demos rápidas para ML, con buena integración y facilidad de uso. React es más tradicional y potente para interfaces complejas, pero su desarrollo es más costoso y excede el alcance actual. Albergaría otro TFG entero.

Por tanto, finalmente se elige Streamlit por su rapidez, facilidad para manejar dashboards y datos con NumPy, y un despliegue ágil, ajustándose a la necesidad de un prototipo visual y ligero.

4.5.7. Comentarios

Cabe destacar que, en entornos de producción o proyectos a gran escala, es habitual desplegar asistentes basados en RAG mediante arquitecturas en la nube como AWS o Azure, las cuales además son más robustas y complejas [18]. No obstante, para este proyecto se prioriza el funcionamiento del RAG, y debido a la naturaleza del proyecto al ser una prueba de concepto (PoC), se opta por una solución ligera, con interfaz asequible y en local.

Parte II

Desarrollo de la solución

Capítulo 5

Análisis

En este apartado se analizará el problema abordado en el proyecto para identificar los distintos requisitos existentes, así como los actores que interactúan con el producto a construir.

5.1. Actores

En la sección de Entorno de Negocio se habían fijado dos stakeholders clave: los estudiantes y los profesores. El estudiante es el usuario central que empleará el asistente de aprendizaje mediante la interfaz. En cambio, es importante la entidad del profesor puesto que deberá evaluar la calidad de las respuestas del asistente. Aunque no interactúe directamente con el sistema, sería importante considerarlo. En la tabla 5.1, se describen las características de cada actor:

ID	Actor	Descripción
A-1	Alumno	El alumno es el actor principal, puesto que va a ser quien directamente use el asistente de aprendizaje para: obtener y responder preguntas tipo test, problemas y preguntas generales sobre la asignatura
A-2	Profesor	El profesor evaluará a través de un formulario la calidad de las respuestas dadas por el asistente para un mejor resultado futuro

Tabla 5.1: Tabla de Actores y Descripciones

5.2. Requisitos de usuario

Anteriormente, ya se había mencionando el alcance del proyecto, el cual tiene como finalidad desarrollar un asistente de aprendizaje para mejorar el dominio y comprensión de una asignatura específica. Para ello, se plantean diversas funcionalidades como:

- Generación de preguntas tipo test adaptadas a las historias seleccionadas por el estudiante.
- Creación de problemas prácticos relacionados con el contenido de la asignatura.
- Respuestas a preguntas generales, abiertas o dudas, obteniendo información relevante desde los materiales educativos proporcionados.

- Registro y análisis del progreso del alumno, ofreciendo estadísticas detalladas sobre su desempeño.

Gracias a esto, el asistente podrá ofrecer una experiencia de aprendizaje adaptativa, permitiendo que cada estudiante refuerce sus conocimientos de manera personalizada. Todo ello mediante la selección de las historias de aprendizaje que facilitará que los estudiantes profundicen en aquellos temas que deseen, además de más indicadores para que el asistente se adapte al ritmo y las necesidades del usuario.

En consecuencia, en esta sección se recogen en la tabla 5.2 los requisitos de usuario que debe cumplir el asistente de aprendizaje a desarrollar:

ID	Requisito de Usuario (RU)
RU - 1	El alumno podrá realizar una pregunta abierta
RU - 2	El alumno podrá solicitar preguntas tipo test
RU - 3	El alumno podrá solicitar supuesto práctico
RU - 4	El alumno podrá visualizar el progreso de su aprendizaje
RU - 5	El alumno podrá visualizar el historial de interacción

Tabla 5.2: Requisitos de Usuario (RU)

5.2.1. Casos de Uso

A continuación, en la figura 5.1 se muestra el diagrama de casos de uso del sistema.

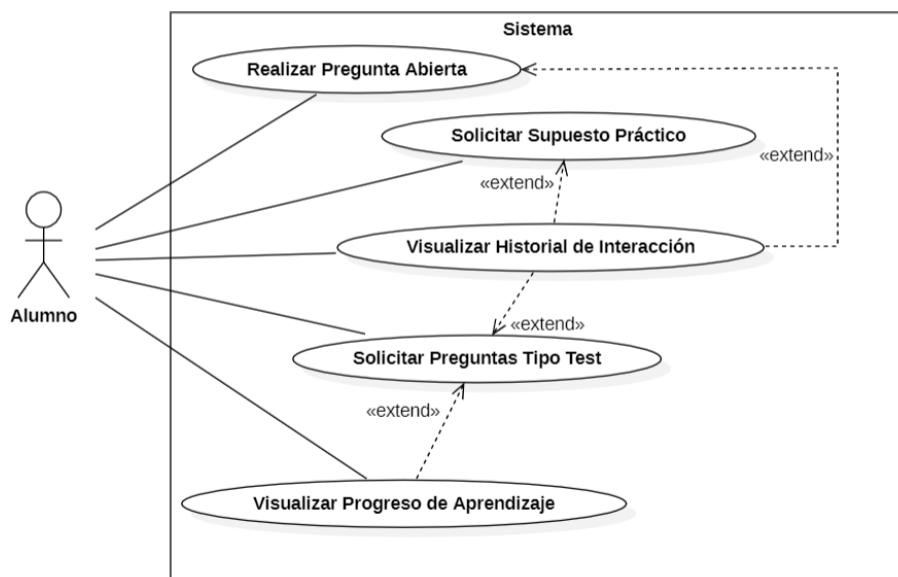


Figura 5.1: Casos de Uso

Especificación de Casos de Uso

Seguidamente, se presenta la especificación de los casos de usos de la interacción entre el usuario y el asistente de aprendizaje 5.3, 5.4, 5.5, 5.6, 5.7.

CU-1	Solicitar Pregunta Abierta
Versión	1.0
Dependencias	
Precondición	Seleccionar como tipo de actividad Consulta
Descripción	El usuario podrá ser capaz de realizar una pregunta, duda o consulta al asistente, la cual se ha de relacionar con el objetivo que especifique
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona Consulta en el menú 2. El sistema redirige al usuario a la página de consultas y habilita la posibilidad de que el usuario seleccione el objetivo e introduzca la respuesta 3. El usuario especifica el objetivo y realiza su consulta 4. El sistema procesa la consulta, muestra la respuesta generada y actualiza el historial de conversación
Postcondición	La respuesta se refleja por pantalla, así como el historial de la conversación actualizado
Excepciones	<p>E1: No se permite la entrada vacía Mensaje Advertencia - Si el usuario incluye una entrada vacía el sistema muestra un mensaje de advertencia y no genera la respuesta.</p> <p>E2: Es necesario presionar el botón, no la tecla Enter Notificación - Si el usuario presiona la tecla Enter en vez de hacer click en el botón para generar la respuesta, el sistema muestra una notificación informativa</p>
Frecuencia	Cada vez que se inicie la aplicación. Cada vez que se quiera cambiar de actividad
Importancia	Muy Alta
Prioridad	Muy Alta
Estado	Finalizado

Tabla 5.3: CU-1 Solicitar Pregunta Abierta

CU-2	Solicitar Pregunta Tipo Test
Versión	1.0
Dependencias	Visualizar Progreso de Aprendizaje
Precondición	Seleccionar como tipo de actividad Tipo Test
Descripción	El usuario podrá ser capaz de responder un cuestionario, en base a la configuración inicial y personalizada que desee
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona Tipo Test en el menú 2. El sistema redirige al usuario a la página de test y muestra el formulario para inicializar la configuración 3. El usuario especifica el número de preguntas 4. El usuario indica el objetivo 5. El sistema muestra la descripción del objetivo seleccionado 6. El usuario selecciona la historia relacionada 7. El sistema muestra la descripción de las historias 8. El usuario añade los criterios de aprendizaje 9. El sistema muestra la descripción de los criterios y los criterios seleccionados 10. El usuario señala el nivel de dificultad 11. El sistema genera el test con las preguntas y opciones 12. El usuario contesta al cuestionario 13. El sistema verifica que todas han sido contestadas, y en ese caso, muestra la reorientación con el desempeño y el resultado
Postcondición	El feedback se muestra por pantalla, además de las preguntas y opciones
Excepciones	<p>E1: Se admiten criterios vacíos Mensaje Advertencia - Por defecto, si el usuario no añade ningún criterio, se seleccionan todos los criterios.</p> <p>E2: Es necesario contestar todas las preguntas: Mensaje Advertencia - Si el usuario no contesta todas las preguntas, el sistema muestra un mensaje de advertencia</p>
Frecuencia	Cada vez que se inicie la aplicación. Cada vez que se quiera cambiar de actividad
Importancia	Muy Alta
Prioridad	Muy Alta
Estado	Finalizado

Tabla 5.4: CU-2 Solicitar Pregunta Tipo Test

CU-3	Solicitar Supuesto Práctico
Versión	1.0
Dependencias	
Precondición	Seleccionar como tipo de actividad Problemas
Descripción	El usuario podrá ver los enunciados generados en función del tipo de problema y supuesto o temática
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona Problemas en el menú 2. El sistema redirige al usuario a la página problemas y muestra el formulario para seleccionar cada tipo de problema 3. El usuario especifica el tipo de problema <ol style="list-style-type: none"> 3.1 Si el tipo de problema es Enunciado Modelo ER: <ol style="list-style-type: none"> 3.1.1. El sistema muestra diferentes temáticas y habilita para Otro una opción de texto 3.1.2. El usuario indica la temática 3.2. Si el tipo de problema es SQL: <ol style="list-style-type: none"> 3.2.1. El sistema presenta los supuestos 3.2.2. El usuario señala el supuesto 4. El sistema genera el enunciado del problema en función de los parámetros seleccionados 5. El sistema muestra el enunciado generado y actualiza el historial de conversación
Postcondición	La respuesta se refleja por pantalla, así como el historial de la conversación actualizado
Excepciones	
Frecuencia	Cada vez que se inicie la aplicación. Cada vez que se quiera cambiar de actividad
Importancia	Muy Alta
Prioridad	Muy Alta
Estado	Finalizado

Tabla 5.5: CU-3 Solicitar Supuesto Práctico

CU-4	Visualizar Progreso de Aprendizaje
Versión	1.0
Dependencias	Solicitar Preguntas Tipo Test
Precondición	Seleccionar como tipo de actividad ver progreso
Descripción	Se mostrará mediante un análisis el progreso del usuario en relación al número de cuestiones que durante la sesión ha respondido: total, correctas y erróneas; y también respecto a las distintas secciones/actividades proporcionadas por el asistente
Secuencia Normal	1. El usuario selecciona Ver Progreso en el menú 2. El sistema presenta los diferentes dashboards
Postcondición	Acceso a ver progreso
Excepciones	E1: Si resulta que no hay datos de tests contestados, el sistema mostrará un mensaje de aviso
Frecuencia	Cada vez que el usuario desee ver su progreso
Importancia	Muy Alta
Prioridad	Muy Alta
Estado	Finalizado

Tabla 5.6: CU-4 Visualizar Progreso de Aprendizaje

CU-5	Visualizar Historial de Interacción
Versión	1.0
Dependencias	Solicitar Supuesto Práctico, Solicitar Preguntas Tipo Test, Realizar Pregunta Abierta
Precondición	Es necesario acceder a alguno de los requisitos de usuario previos, excepto el de visualizar progreso
Descripción	El usuario podrá ser capaz de visualizar el historial de interacción en cada una de las actividades. En consultas o problemas será el historial de sesión mientras que en el tipo test las preguntas generadas para ese momento
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario selecciona una actividad 2. El sistema redirige a la página de esa actividad 3. Si es la primera vez que se accede dentro de la sesión actual: 3.1 El usuario debe realizar alguna interacción 3.2. El sistema no muestra historial, ya que no hay 4. Una vez que el usuario realiza alguna acción, el sistema actualiza el historial y lo presenta por pantalla
Postcondición	Se muestra el historial de interacción para esa actividad
Excepciones	E1: Si no hay interacciones previas, el sistema no muestra nada dentro historial
Frecuencia	Cada vez que se accede a una actividad
Importancia	Muy Alta
Prioridad	Muy Alta
Estado	Finalizado

Tabla 5.7: CU-5 Visualizar Historial de Interacción

5.3. Requisitos de información

Por otro lado, es destacable la existencia de los requisitos de información para soportar los requisitos de usuario. A través del diseño conceptual que se presenta a continuación, se recogería la información necesaria. Para ello, se muestra el modelo entidad relación del sistema y su diccionario de datos relacionado.

5.3.1. Modelo Entidad Relación

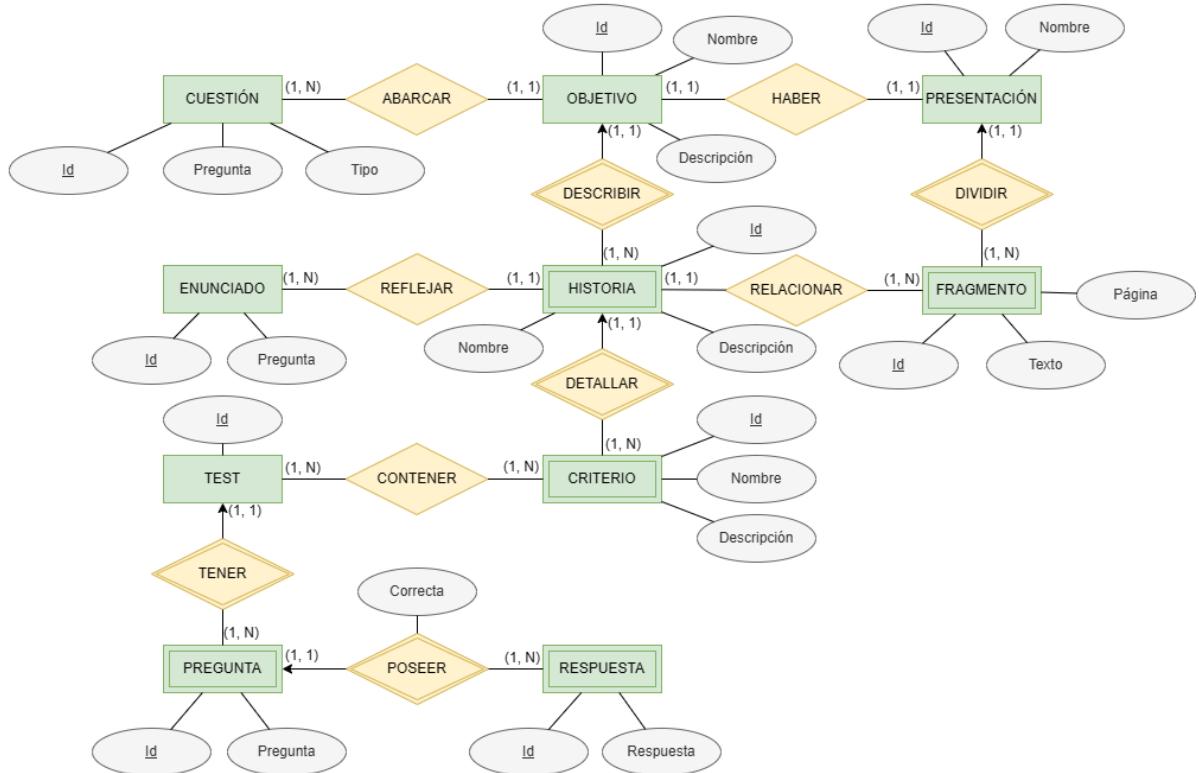


Figura 5.2: Modelo Entidad Relación

5.3. Requisitos de información

5.3.2. Diccionario de Datos

Entidades

ID	E-01		Nombre	CUESTIÓN						
Definición: Representa la cuestión que introduce el usuario en base a un objetivo										
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricciones o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A01.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A01.02	Pregunta	Pregunta, duda o cuestión que introduce el usuario	TEXT	No vacío	No	No	No	No	No
Identificadores:										
Identificadores:		ID	Nombre Atributo							
		A01.01	Id							

Figura 5.3: Diccionario de Datos - CUESTIÓN

ID	E-02		Nombre	OBJETIVO						
Definición: Representa los diferentes objetivos dentro de la visión										
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricciones o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A02.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A02.02	Nombre	Nombre del objetivo a tratar	ENUMARATE	SGBDs, Diseño Lógico, Diseño Conceptual, DDL y DML	No	No	Sí	No	No
	A02.03	Descripción	Descripción detallada sobre el objetivo	TEXT		No	No	Sí	No	Sí
Identificadores:										
Identificadores:		ID	Nombre Atributo							
		A02.01	Id							

Figura 5.4: Diccionario de Datos - OBJETIVO

ID	E-03		Nombre	ENUNCIADO						
Definición: Representa el enunciado que genera el asistente en base a las historias										
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restriccione s o condicionamien tos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A03.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A03.02	Enunciado	Enunciado generado por el asistente	TEXT	No vacío	No	No	No	No	No
Identificadores:										
Identificadores:		ID	Nombre Atributo							
		A03.01	Id							

Figura 5.5: Diccionario de Datos - ENUNCIADO

Capítulo 5. Análisis

ID	E-04		Nombre	HISTORIA						
Definición:	Representa las diferentes historias que contiene un objetivo, por tanto, HISTORIA es DÉBIL de OBJETIVO.									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
Atributos:	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restriccione s o condiciona_mientos que afecten al atributo	¿Es multivaluad o? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A04.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A04.02	Nombre	Nombre de la historia a tratar	ENUMARATE	Valores definidos en ENUMERATE y dependen del objetivo	No	No	Sí	No	No
	A04.03	Descripción	Descripción detallada sobre la historia	TEXT		No	No	Sí	No	Sí
Identificadores:	ID	Nombre Atributo		Id						

Figura 5.6: Diccionario de Datos - HISTORIA

ID	E-05		Nombre	TEST						
Definición:	Representa el test que genera el asistente en base a los criterios									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
Atributos:	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restriccione s o condiciona_mientos que afecten al atributo	¿Es multivaluad o? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A05.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
Identificadores:	ID	Nombre Atributo		Id						

Figura 5.7: Diccionario de Datos - TEST

ID	E-06		Nombre	CRITERIO						
Definición:	Representa los diferentes criterios que contiene una historia, por tanto, CRITERIO es DÉBIL de HISTORIA									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
Atributos:	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restriccione s o condiciona_mientos que afecten al atributo	¿Es multivaluad o? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A06.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A06.02	Nombre	Nombre del criterio a tratar	ENUMARATE	Valores definidos en ENUMERATE y dependen de la historia	No	No	Sí	No	No
	A06.03	Descripción	Descripción detallada sobre el criterio	TEXT		No	No	Sí	No	Sí
Identificadores:	ID	Nombre Atributo		Id						

Figura 5.8: Diccionario de Datos - CRITERIO

5.3. Requisitos de información

ID	E-07		Nombre	PREGUNTA						
Definición:	Representa la pregunta del test que genera el asistente, por lo que PREGUNTA es DÉBIL de TEST									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricciones o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A07.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A07.02	Pregunta	Pregunta generada por el asistente	TEXT		No	No	No	No	No
Identificadores:	ID	Nombre Atributo								Id

Figura 5.9: Diccionario de Datos - PREGUNTA

ID	E-08		Nombre	RESPUESTA						
Definición:	Representa la respuesta a la pregunta del test que genera el asistente, por tanto, RESPUESTA es DÉBIL de PREGUNTA									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricciones o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A08.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A08.02	Respuesta	Respuesta a la pregunta generada por el asistente	TEXT		No	No	No	No	No
Identificadores:	ID	Nombre Atributo								Id

Figura 5.10: Diccionario de Datos - RESPUESTA

ID	E-09		Nombre	PRESENTACIÓN						
Definición:	Representa el material docente impartido en la asignatura, el cual se encuentra en diferentes presentaciones									
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluad	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricciones o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí/No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A09.01	Id	Identificador	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A09.02	Nombre	Nombre de la presentación	TEXT		No	No	Sí	No	No
Identificadores:	ID	Nombre Atributo								Id

Figura 5.11: Diccionario de Datos - PRESENTACIÓN

Capítulo 5. Análisis

ID	E-10		Nombre	FRAGMENTO						
Definición: Representa los diferentes fragmentos en los que se divide la presentación, por lo que FRAGMENTO es DÉBIL de PRESENTACIÓN										
Atributos:	ID	Nombre	Definición	Tipo	Reglas	Multivaluado	Compuesto	UNIQUE	DEFAULT	NULL
	ID atributo	Nombre Atributo	Descripción del atributo	Tipo de datos del atributo	Restricción es o condicionamientos que afecten al atributo	¿Es multivaluado? (Sí / No)	¿Es compuesto? (Sí / No)	¿El valor tiene que ser diferente para todas las instancias? (Sí / No)	¿El atributo tiene algún valor inicial?	¿Es posible que el valor del atributo sea NULL? (Sí / No)
	A10.01	Id	Identificador del fragmento	AUTO INCREMENTAL (INT)		No	No	Sí	No	No
	A10.02	Texto	Contenido del fragmento	TEXT		No	No	No	No	No
Identificadores:	ID	Nombre Atributo		Id						
	A10.01									

Figura 5.12: Diccionario de Datos - FRAGMENTO

Relaciones

ID	R-01		Nombre	ABARCAR	
Definición: La relación modela que en una cuestión se abarca un objetivo					
Entidades	ID	Nombre Entidad	Participación	Cardinalidad	
	ID entidad	NOMBRE_ENTIDAD	Opcional (0) / Obligatoria (1)	Uno (1) / Muchos (N)	
	E-01	CUESTIÓN	1	N	
	E-02	OBJETIVO	1	1	

Figura 5.13: Diccionario de Datos - ABARCAR

ID	R-02		Nombre	REFLEJAR
Definición: La relación modela que un enunciado refleja diferentes historias				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	ID entidad	NOMBRE_ENTIDAD	Opcional (0) / Obligatoria (1)	Uno (1) / Muchos (N)
	E-03	ENUNCIADO	1	N
	E-04	HISTORIA	1	1

Figura 5.14: Diccionario de Datos - REFLEJAR

ID	R-03		Nombre	DESCRIBIR
Definición: La relación modela que un objetivo contiene diferentes historias, por lo que es DÉBIL				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	ID entidad	NOMBRE_ENTIDAD	Opcional (0) / Obligatoria (1)	Uno (1) / Muchos (N)
	E-02	OBJETIVO	1	1
	E-04	HISTORIA	1	N

Figura 5.15: Diccionario de Datos - DESCRIBIR

5.3. Requisitos de información

ID	R-04	Nombre	CONTENER
Definición:			
Entidades	ID ID entidad	Nombre Entidad NOMBRE_ENTIDAD	Participación Opcional (0) / Obligatoria (1)
	E-05	TEST	Cardinalidad Uno (1) / Muchos (N)
	E-06	CRITERIO	1 N

Figura 5.16: Diccionario de Datos - CONTENER

ID	R-05	Nombre	DETALLAR
Definición:			
Entidades	ID ID entidad	Nombre Entidad NOMBRE_ENTIDAD	Participación Opcional (0) / Obligatoria (1)
	E-04	HISTORIA	Cardinalidad Uno (1) / Muchos (N)
	E-05	CRITERIO	1 N

Figura 5.17: Diccionario de Datos - DETALLAR

ID	R-06	Nombre	TENER
Definición:			
Entidades	ID ID entidad	Nombre Entidad NOMBRE_ENTIDAD	Participación Opcional (0) / Obligatoria (1)
	E-05	TEST	Cardinalidad Uno (1) / Muchos (N)
	E-07	PREGUNTA	1 N

Figura 5.18: Diccionario de Datos - TENER

ID	R-07	Nombre	POSEER
Definición:			
Entidades	ID ID entidad	Nombre Entidad NOMBRE_ENTIDAD	Participación Opcional (0) / Obligatoria (1)
	E-07	PREGUNTA	Cardinalidad Uno (1) / Muchos (N)
	E-08	RESPUESTA	1 N
Atributos:	ID atributo	Nombre Atributo	Descripción del atributo (con toda la independencia posible de la relación que lo contiene).
	R07.01	Correcta	Indica si la respuesta generada es la opción correcta para la pregunta con
			Tipo de datos del atributo BOOLEAN
			Reglas Restricciones o condicionamientos que afecten al atributo
			Multivaluado ¿Es multivalorado?
			Compuesto ¿Es compuesto?
			UNIQUE ¿El valor tiene que ser diferente para todas las instancias? (Sí / No)
			DEFAULT ¿El atributo tiene algún valor inicial?
			NULL ¿Es posible que el valor del atributo sea NULL? (Sí / No)

Figura 5.19: Diccionario de Datos - POSEER

ID	R-08	Nombre	DIVIDIR
Definición:			
Entidades	ID ID entidad	Nombre Entidad NOMBRE_ENTIDAD	Participación Opcional (0) / Obligatoria (1)
	E-09	PRESENTACIÓN	Cardinalidad Uno (1) / Muchos (N)
	E-10	FRAGMENTO	1 N

Figura 5.20: Diccionario de Datos - DIVIDIR

ID	R-09		Nombre	HABER
Definición: La relación modela que una objetivo se encuentra dentro de una presentación				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	ID entidad	NOMBRE_ENTIDAD	Opcional (0) / Obligatoria (1)	Uno (1) / Muchos (N)
	E-02	OBJETIVO	1	1
	E-09	PRESENTACIÓN	1	1

Figura 5.21: Diccionario de Datos - HABER

ID	R-09		Nombre	RELACIONAR
Definición: La relación modela que una historia se basa o relaciona en diferentes fragmentos				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	ID entidad	NOMBRE_ENTIDAD	Opcional (0) / Obligatoria (1)	Uno (1) / Muchos (N)
	E-04	HISTORIA	1	1
	E-10	FRAGMENTO	1	N

Figura 5.22: Diccionario de Datos - RELACIONAR

5.4. Requisitos no funcionales

Finalmente, se identifican y especifican en la tabla 5.8 los requisitos no funcionales que determinan la calidad del producto final:

ID	Requisito No Funcional (RNF)
RNF - 1	El almacenamiento de datos debe realizarse de forma segura y cumplir con normativas de protección de datos, como el RGPD
RNF - 2	Se debe garantizar un tiempo de disponibilidad del sistema del 99.5 %
RNF - 3	El tiempo de espera de la respuesta proporcionada por el asistente no debe exceder de 1 minuto
RNF - 4	La interfaz será amigable, intuitiva y accesible a todo el mundo, donde el 90 % de usuarios sin experiencia previa complete cada tarea en menos de 5 minutos y la puntuación otorgada por estos en encuestas de facilidad de uso se encuentre como mínimo de 4 sobre 5
RNF - 5	El sistema debe ser ético y respetuoso, garantizando que las respuestas del asistente no sean discriminatorias, sesgadas ni ofensivas hacia ningún grupo o individuo siguiendo principios de igualdad y justicia, garantizando que el 99 % cumpla con estas condiciones mediante una validación con la Constitución. Además, la puntuación mínima otorgada por el usuario será de 4 sobre 5 en encuestas
RNF - 6	El historial de interacción del usuario debe ser almacenado en formato JSON y CSV para facilitar el acceso y análisis, permitiendo la personalización y mejora continua del sistema
RNF - 7	El sistema debe ser eficiente en términos de recursos, utilizando técnicas de procesamiento de lenguaje natural (PLN) como Langchain y garantizando que las consultas y respuestas se procesen de manera rápida y precisa, sin afectar el rendimiento general del sistema. Para ello, manteniendo un tiempo de respuesta menor a 1 minuto en el 99 % de los casos
RNF - 8	El sistema debe permitir el registro de logs de errores y eventos críticos, almacenándolo con una duración mínima de 90 días
RNF - 9	El sistema debe ser fácilmente modificable, permitiendo la incorporación de nuevas funcionalidades en menos de 1 mes
RNF - 10	En el almacenamiento de datos se debe garantizar el cifrado de datos privados, además de realizar backends semanales

Tabla 5.8: Requisitos No Funcionales (RNF)

5.5. Limitaciones y Restricciones

5.5.1. Limitaciones del Proyecto

- **L1:** El asistente está diseñado para apoyar una forma estructurada de organización del aprendizaje (como es el producto de aprendizaje), aplicable en distintos contextos. No obstante, se evaluará con contenidos de una única asignatura y como prueba de concepto.
- **L2:** La base de datos de conocimiento se limitará a los recursos proporcionados por la asignatura.
- **L3:** Se centrará en consultas técnicas y conceptuales, sin incluir preguntas de opinión o ejercicios de extensos.
- **L4.** La interfaz se implementará como una aplicación web ligera, accesible desde navegadores de escritorio. No se desarrollarán versiones específicas para dispositivos móviles.

5.5.2. Restricciones de Interfaz Externa

- **R1 - Dependencia de la API:** El sistema depende del modelo de la API de OpenAI para la generación de preguntas, lo que impone limitaciones en latencia, rendimiento, accesibilidad, costos y privacidad de datos. Asimismo, se trata de un servicio externo, por lo que no se pueden modificar los pesos del LLM, se requiere de conexión a internet para interactuar con él y de su actual *status* (5.23). Por tanto, esta dependencia afectará al flujo de los casos de uso cuando el usuario interactúe con la interfaz, por lo que destacar que se tendrá en cuenta internamente en el diseño de cada flujo y en las posibles excepciones de los requisitos de usuario. Por último, resaltar que está sujeto a límites de cuota y tarifas, lo que podría afectar la frecuencia de uso y el costo a largo plazo.

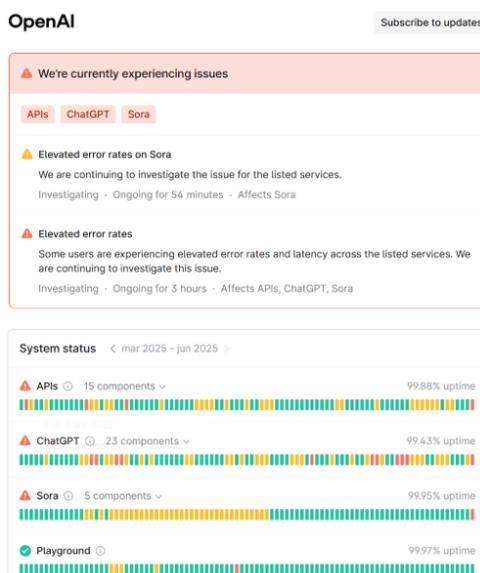


Figura 5.23: Ejemplo *Status* Erróneo de OpenAI a día 10 de Junio

Capítulo 6

Diseño

En esta sección, se establece el diseño de la solución para poder construir un producto que alcance los objetivos del proyecto. Este diseño abarca la definición de la estructura y los componentes del asistente de aprendizaje, asegurando su alineación con los requisitos del proyecto. Además, se detallarán los aspectos clave de la arquitectura, la funcionalidad, el almacenamiento de datos y la interfaz de usuario. También se presentará el modelado visual de las decisiones de diseño y su documentación integrada, garantizando una visión completa y estructurada del sistema.

6.1. Arquitectura Lógica Alto Nivel

La arquitectura lógica planteada, aunque se fundamenta en una arquitectura RAG por componentes, se puede organizar en distintas capas *software* como se muestra en la Figura 6.1. Aun así es importante destacar que dentro de estas capas, especialmente la capa de negocio, se pueden abstraer distintos módulos, los cuales hacen referencia a los componentes que conforman un RAG. Además, destacar que en la sección 6.3 se presenta la arquitectura lógica de bajo nivel.

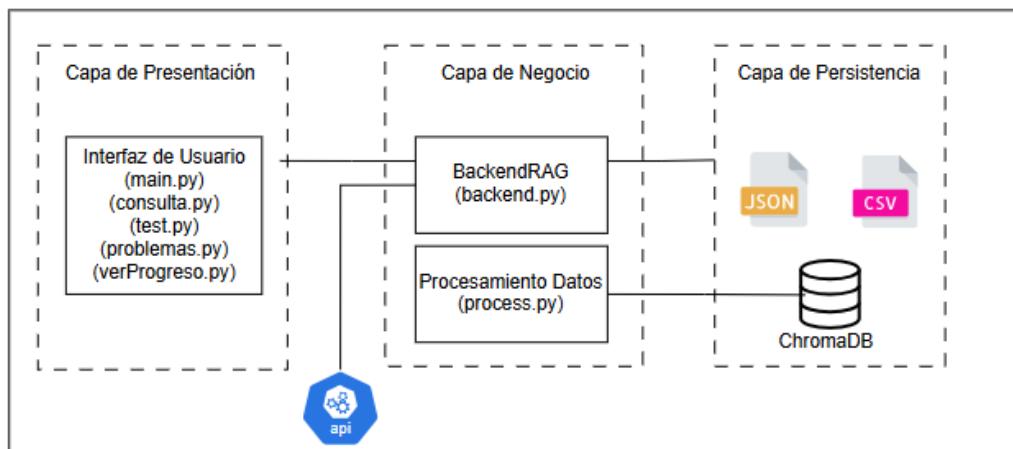


Figura 6.1: Arquitectura Lógica Alto Nivel

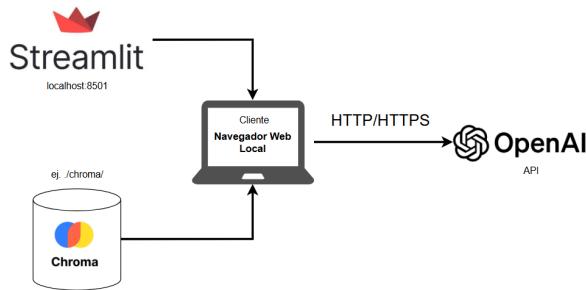


Figura 6.2: Arquitectura Física

6.2. Arquitectura Física

La arquitectura física del sistema 6.2 está diseñada para ejecutarse completamente en un entorno local, dentro de una única máquina, como es el ordenador propio. El asistente se ejecuta mediante Streamlit, el cual inicia un servidor web local accesible desde el navegador en la dirección localhost:8501. En este caso, el navegador web local actúa como cliente, mientras que la aplicación actúa como servidor local, ambos ejecutándose en el mismo entorno local. Por otra parte, la gestión de los datos se procesa mediante la base vectorial ChromaDB, que se almacena en el disco local como archivos (.sqlite3, metadatos y otros formatos internos). Aunque estos archivos son visibles en el sistema de archivos, su contenido no es directamente legible. Cabe resaltar que la única conexión externa es la de la API de OpenAI, que requiere acceso a Internet y se comunica mediante el protocolo HTTPS. Esta API se usa para funcionalidades como generación de texto.

Este diseño permite un despliegue rápido, funcional y autónomo, lo que resulta ideal para probar el asistente sin depender de entornos *cloud*, servidores remotos ni infraestructura adicional. En definitiva, el sistema completo (cliente, servidor, lógica y base de datos) reside y se ejecuta en su mayoría localmente, excepto por la dependencia puntual del servicio de OpenAI.

6.3. Arquitectura Lógica Bajo Nivel

A continuación, se presenta los elementos clave que componen cada capa, más tarde se detallará el flujo final incorporando diversos diagramas.

6.3.1. Componentes

Interfaz de usuario:

Gestiona la interacción entre el usuario (alumno) y el asistente, por lo que se alberga en la capa de presentación. Inicialmente permite la selección entre distintas actividades: consulta general, tipo test, problemas y ver progreso. Dependiendo de la opción, se mostrará y realizarán diferentes lógicas de negocio. Para ello, el archivo `main.py` actúa como punto de entrada y controlador de la navegación. Asimismo, se utiliza el paquete `páginas` que agrupa los módulos correspondientes a cada funcionalidad. Cada archivo dentro de `páginas` contendrá los componentes dinámicos necesarios para la interacción con el usuario y, paralelamente, invocar la lógica de negocio correspondiente.

- En primer lugar, si el usuario selecciona consulta general, se carga `consultas.py`, y se muestra un recuadro en el que el alumno indica su duda, pregunta abierta o explicación más general a cerca de los contenidos de la asignatura. Así, sucesivamente hasta que decide salir o cambiar de actividad.
- En el caso de tipo test, `test.py`, el usuario debe indicar sobre qué parte del contenido, más bien, objetivos, (SGBDs, Diseño Conceptual, Diseño Lógico, DDL y DML), historias y criterios de aprendizaje se generarán las cuestiones de opción múltiple. De esta manera, se consigue personalizar los aspectos a tratar por el alumno. Además, incluye otra opción donde podrá perfilar el nivel de dificultad y el número de preguntas que desee. Una vez que inicie el test, se mostrará por pantalla las diversas preguntas, las cuales tendrá que ir contestando, hasta que las finalice. En ese caso, se evaluaría y enseñaría el *feedback* correspondiente en tiempo real.
- Respecto a los problemas, `problemas.py`, es similar al test, salvo que el asistente generaría ejercicios para o bien enunciados de Modelos Entidad Relación, o funcionalidades de SQL. Resaltar que en el caso de los problemas SQL, seleccionaría entre varios supuestos previos vistos en clase y para los enunciados, una temática en la que ambientar el problema.
- Finalmente, existe la opción de ver progreso, `progreso.py`, donde se mostraría un análisis sobre el comportamiento y progreso del alumno en los test mediante gráficos y tablas. Tal estudio estará relacionado con la creación y visualización de distintos dashboards en relación con: el número de preguntas contestadas de manera correcta y errónea, tanto de manera global como por sección; tiempo transcurrido y puntuación total. En él, se predispondrán los siguientes elementos:
 - KPIs clave, como la suma total de acierto, fallos, tiempo total y tiempo medio por pregunta.
 - Gráfica de barras apiladas (*stacked bar chart*) para visualizar los aciertos y fallos por objetivo.
 - Gráfica de barras (*bar chart*) del total de preguntas por objetivo.
 - Gráfica circular (*pie chart*), que muestra la proporción global de aciertos y fallos.
 - Gráfica de barras con el tiempo total invertido por objetivo.
 - Tabla final con los datos.

Como se mencionó anteriormente, toda la gestión e interacción de la interfaz con el usuario se centraliza en el archivo `main.py`. Este archivo actúa como controlador principal, delegando la ejecución a los distintos módulos contenidos en el paquete páginas. Esta organización permite una separación clara de responsabilidades, facilitando la mantenibilidad y desarrollo del código al distribuir la funcionalidad por componentes específicos.

Procesamiento de datos

Este módulo se encarga de procesar y preparar los documentos proporcionados para la asignatura en la que se basará el asistente, encontrándose dentro de la capa de negocio mediante el archivo `process.py`. Este proceso se realizaría antes del inicio del asistente, de tal forma que se crea previamente la base de conocimiento en la que fundamentar el RAG. Para ello, se extrae

el texto de cada pdf proporcionado y se segmenta en fragmentos (*chunks*) luego se transforman a *embeddings*, y se crean los distintos índices y vectores de búsqueda, así como las bases. Estos vectores se almacenan en la capa de persistencia en la base de datos vectorial, lo que optimiza las búsquedas posteriores al proporcionar los *embeddings* necesarios y útiles para las consultas que realice el usuario. Además, a la hora de la creación y almacenamiento de los *embeddings*, se añaden metadatos adicionales para facilitar el filtrado y recuperación eficiente de la información. En resumen, este proceso sería la fase de *Retrieval* dentro del RAG.

Backend RAG

En este punto, reside la lógica central del sistema, concretamente en el archivo `backend.py`, el cual se sitúa en la capa de negocio. Este implementa los distintos componentes principales del RAG, gestionando la interacción con la base vectorial, la recuperación de información relevante, el procesamiento del contexto y la generación de respuestas que más tarde se mostrarán por la interfaz, proporcionando soluciones personalizadas y contextualizadas.

Su funcionamiento radica en que `backend.py` recibe la entrada del usuario, la analiza y ejecuta un flujo en función de la actividad seleccionada. Asimismo, los parámetros también varían:

- Cuando se trata de la consulta, se pasa la pregunta del usuario junto con el objetivo y el historial de la sesión (recopilado por medio de la interfaz).
- En el tipo test, la pregunta sería el conjunto de criterios concatenados que ha seleccionado, además de que también se adjunta el historial de tests anteriores así como el objetivo, historia de aprendizaje, número de preguntas y nivel de dificultad en el que basar el test.
- Para los problemas, se especifica el tipo de problema (enunciado ER o SQL). En el caso de los problemas SQL, se añade el supuesto a resolver, y en el ER, la temática; además de adjuntar el historial de problemas.

Por tanto, la estructura del archivo se encuentra modularizada para adaptarse a las diferentes actividades. En particular, se divide en tres módulos principales:

1. RAG General: para responder a consultas generales sobre la materia.
2. RAG Test: para la generación de preguntas tipo test y evaluar el rendimiento del usuario en tiempo real.
3. RAG Problemas: para generar problemas prácticos que el alumno debe resolver.

A pesar de sus diferencias, estos comparten una misma lógica subyacente basada en el enfoque RAG. Todos ellos reutilizan un planteamiento común que define la lógica del RAG. En otras palabras, el proceso incluye la carga, búsqueda y recuperación de fragmentos relevantes. Posteriormente, a partir de cierto historial de sesión, contexto y query, el aumento y generación de respuestas con LLM seguiría un enfoque similar. La diferencia radica en los parámetros de entrada y por tanto, en la construcción de los prompts. Además, el proceso difiere en que no se emplean los mismos documentos base para generar un enunciado que para responder a una consulta conceptual o realizar el test. Asimismo, también es distinto el trato con un filtrado por objetivos e historias que con una recuperación únicamente semántica. Por tanto, durante la recuperación varía el criterio de similitud, ya que en las consultas o test se prioriza la recuperación

semántica, mientras que para la generación de problemas se prefiere un filtrado por metadatos que recupere un conjunto más amplio y directo.

De la misma manera, la manipulación de las respuestas obtenidas por el LLM será diferente. Por ejemplo, en la consulta o problemas se proporcionará directamente la respuesta, pero en los test será necesario un tratamiento previo para separar la opción correcta de cada pregunta para poder presentarlo de manera correcta en la interfaz y personalizar la experiencia del aprendizaje.

Finalmente respecto a la llamada a la API, esta se centraliza en cuanto a la elección del modelo, *key*, temperatura y su construcción. No obstante, debido a que varían los prompts, los parámetros a añadir al LLM también son distintos, principalmente por la necesidad de personalizar el contenido generado según el objetivo, la actividad o el tipo de problema tratado.

Para gestionar esta diversidad, aspectos en común o mantener una estructura clara y modular, se ha elaborado distintos archivos con responsabilidades y utilidades específicas:

- **prompt.py**, contiene los múltiples prompts bases que empleará el LLM (que serán enviados a través de la API) para cada tipo de actividad. Con esta organización, se favorece la mantenibilidad y reutilización, ya que los prompts están parametrizados mediante variables que se adaptan en función de la información recuperada o de las opciones seleccionadas por el usuario en la interfaz. Por ejemplo, pueden variar según el objetivo trabajado (como SGBDs o Diseño Lógico), el tipo de tarea o el contexto. Por esta razón, no se recurre a un prompt *template*.
- **definiciones.py**, aunque se encuentre dentro del paquete **páginas**, contiene las definiciones de los objetivos, historias de aprendizaje y criterios, permitiendo construir dinámicamente el contenido contextual del prompt. Su ubicación vinculada a la interfaz responde a la necesidad de mantener actualizada automáticamente la interfaz si se modifica algún objetivo o criterio.
- **config.py**, finalmente, las variables globales de configuración se recopilan en este archivo, como el modelo por defecto, la clave de la API o diversas rutas de archivos a emplear (como los que se comentan a continuación).

Con esta organización modular, se consigue mantener un sistema más limpio, ordenado y fácilmente escalable, asegurando que cada componente esté bien delimitado y especializado en su función concreta.

Por otro lado, dentro del *backend* de RAG, es importante destacar que también se gestiona el almacenamiento de las respuestas proporcionadas por el modelo en un archivo JSON (junto a otras características). Este archivo se empleará y detallará más adelante, pero a grandes rasgos comentar que será necesario para crear un archivo que se enviará a los profesores expertos en la asignatura para poder evaluar el rendimiento y calidad del asistente, y también para permitir su evaluación por medio de distintas herramientas destinadas para ello. Sin embargo, esto se puntualizará más adelante en el apartado de datos y el capítulo Aceptación 9.

Análogamente, se realiza para el progreso del alumno, de tal manera que se guarda en un CSV las distintas interacciones que haya tenido en las cuestiones de opción múltiple. En consecuencia, los datos relacionados con las respuestas, progreso e interacciones persisten de manera local para ser procesados y analizados.

Por último destacar que aunque se procesen en esta capa dichos datos, realmente es en la interfaz donde se generan los múltiples dashboards informativos, ya que es menos laborioso gracias

a los recursos que proporciona la herramienta empleada para su creación: Streamlit. Asimismo, los archivos generados, se encontrarían en la capa de persistencia.

En definitiva, en esta parte se encuentra la lógica principal del sistema en cuanto a las fases principales del RAG, organizada en módulos según la actividad seleccionada por el usuario. Este es un diseño ágil, eficaz y eficiente que permite una mayor reutilización de componentes, optimizando el rendimiento mediante flujos diferenciados por actividad. Además, la separación clara de responsabilidades facilita la escalabilidad y el mantenimiento del sistema.

6.3.2. Comportamiento

A continuación, se detalla el flujo final. El proceso comienza con la interacción del alumno en la interfaz, donde selecciona una actividad (consulta, test, problemas o ver progreso). No obstante, es importante volver a destacar que previamente se ha llevado a cabo el procesamiento de los documentos, generando y guardando los distintos *embeddings* y metadatos para optimizar las búsquedas semánticas y filtrado de la información (o la fase *Retriever*) en la base de datos vectorial local (ChromaDB). Esta carga previa (junto al diseño modular) permite reducir la carga computacional durante la ejecución y mejora significativamente la velocidad de respuesta. Al mantenerse en local, se elimina la dependencia de servicios externos para la recuperación de información, lo que garantiza mayor autonomía y eficiencia.

Dependiendo del tipo de actividad, se dan diversos sucesos: (antes de nada, destacar que la implementación interna, se detalla más adelante en la implementación, puesto que ahora solo se centra el apartado de diseño)

- Consulta: el usuario selecciona el objetivo a tratar e introduce una pregunta. Estos se adjuntan como parámetro junto al historial de preguntas de la sesión en una función que se encontrará en el *backend*. A continuación, se carga la base de datos y se realizaría la recuperación de la información, mediante la optimización, filtrado búsqueda de fragmentos más relevantes. Finalmente, se adjuntaría tanto la pregunta, como el historial y el contexto recuperado en un prompt, el cual se enviaría por medio de la API al LLM para generar la respuesta. Por último, si no ha habido ningún error de la API, esta se guardaría y se mostraría por pantalla al usuario.
- Test: el usuario selecciona los objetivos, historias, criterios, dificultad y número de preguntas sobre el que realizar el cuestionario. A continuación, se carga la base de datos y se realizaría la recuperación de la información, mediante la optimización y búsqueda de fragmentos más relevantes. En este caso, se filtraría por objetivo, historia y semánticamente por criterios. Más adelante, se añadirían en un prompt todos los parámetros iniciales que ha elegido el estudiante y, análogamente al anterior, se enviaría por medio de la API al LLM para generar la respuesta y guardarla. Puntualizar que en esta situación, se manipularía la respuesta devuelta para poder disponer por la interfaz cada pregunta con sus opciones relacionadas al usuario. Una vez que el usuario interactúa con la interfaz y marca las distintas opciones, se procesa sus elecciones, se muestra las opciones correctas, se comparan, visualiza la puntuación, tiempo, preguntas correctas o erróneas, y todo ello, se guardaría en un CSV.
- Problemas: de igual manera, el usuario ha de elegir entre el tipo de problema que le interesa (enunciado o SQL). En el caso de los problemas SQL, deberá decidir el supuesto. En el

enunciado, la temática en la que ambientar el ejercicio. Más adelante, se recuperaría la información relacionada con el tipo de enunciado, en este caso mediante únicamente un filtrado por metadatos. A continuación, se añadirían en un prompt el tipo de problema, historial, ejemplos de ejercicios similares recuperados y se enviaría por medio de la API al LLM para generar la respuesta y también, guardarla. Finalmente, se mostraría por pantalla.

- Ver progreso: para dicha actividad, se muestra por pantalla las diversas estadísticas y análisis de su progreso, actuación e interacción en los test mediante múltiples dashboards realizados a partir de la información correspondiente almacenada.

Es importante destacar la fase de recuperación de fragmentos relevantes, ya que es esencial para garantizar la calidad de las respuestas del asistente. En el caso de una consulta o una pregunta tipo test, el proceso comienza con un filtrado inicial por objetivo/historia, delimitando así el subconjunto de documentos que poseen contenido relevante. A continuación, se realizarían una comparación semántica entre los *embeddings* de la base de datos vectorial con la entrada convertida a *embedding*, y se recuperaría los 10 fragmentos más similares a la pregunta del usuario. Resaltar que en el caso de que la recuperación de los fragmentos fuera vacía, se procedería a realizar la búsqueda de manera global, sin filtrar por objetivo y así evitar una solución nula.

Tras recuperar los fragmentos más similares, se aplica un proceso de *reranking* para poder identificar aquellos 3 más relevantes. Para ello, se comparan los fragmentos recuperados con la pregunta original, seleccionando aquellos más coherentes y afines a la entrada. Este proceso se lleva a cabo mediante un modelo LLM especializado en clasificación, distinto al usado posteriormente para generar las respuestas finales. Sin embargo, los fragmentos finalmente recopilados por este LLM, se utilizan como contexto en el modelo generativo, lo que permite mejorar tanto la precisión como la adecuación del contenido proporcionado por el asistente.

Una vez presentado el flujo, cabe destacar que la personalización del aprendizaje se encuentra integrada en distintos niveles del sistema. El asistente adapta su comportamiento en función de los objetivos, historias y criterios seleccionados por el alumno, permitiendo ajustar la experiencia según su progreso, necesidades y dificultad. Asimismo, los prompts generados se ajustan dinámicamente según la actividad, contexto e historial, mejorando la relevancia de las respuestas generadas por el modelo. Además, el sistema proporciona retroalimentación inmediata, evalúa el rendimiento en tiempo real y genera dashboards interactivos donde el alumno puede consultar métricas personalizadas como precisión, tiempo de respuesta y evolución en los distintos tipos de tareas.

En conclusión, el sistema ofrece una experiencia personalizada y optimizada, asegurando una interacción eficiente y escalable. En parte, también gracias a la estructuración de proyectos dentro de eXtreme Learning.

6.4. Diagramas

En esta sección, se adjuntan a su vez diversos diagramas para el diseño que facilitan el entendimiento del comportamiento del asistente.

6.4.1. Diagrama de Despliegue

En primer lugar, se presenta el diagrama de despliegue en la Figura 6.3, el cual permite modelar la disposición física de los artefactos en nodos.

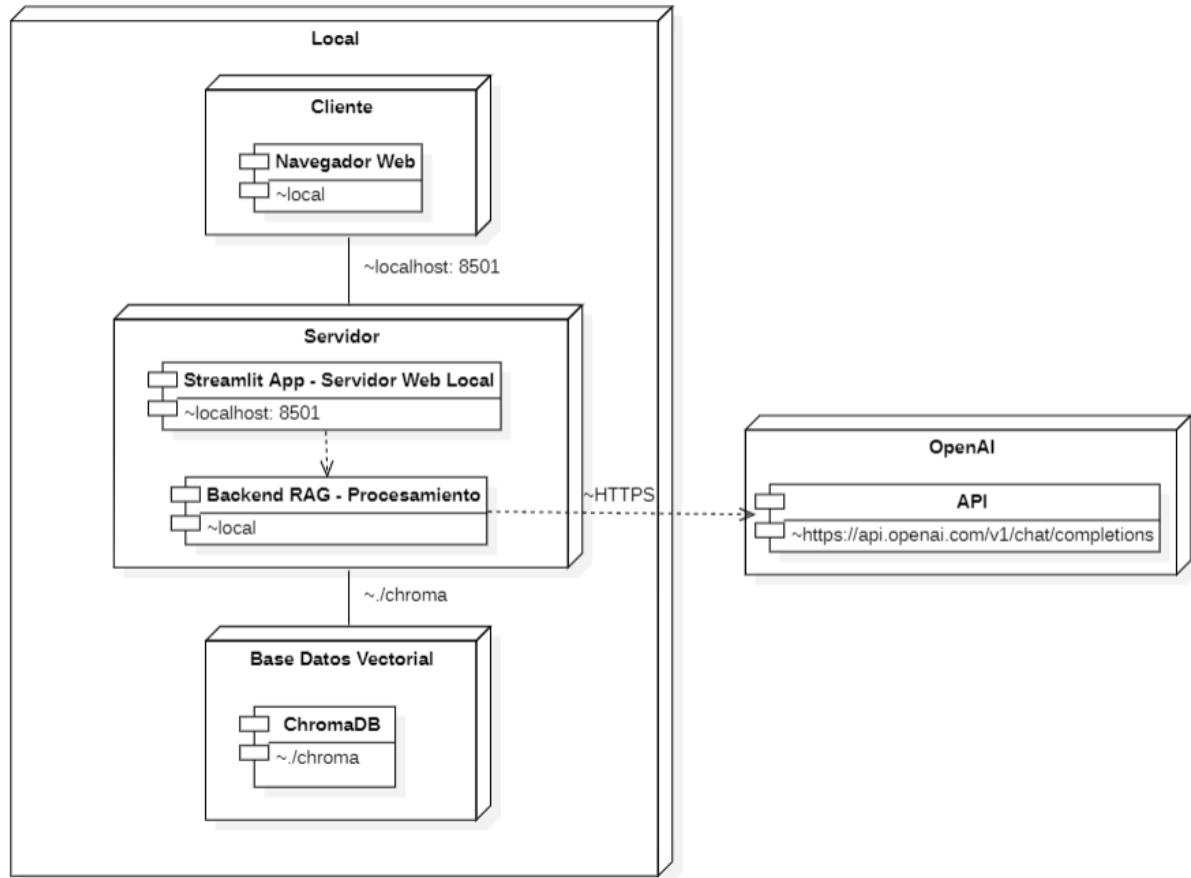


Figura 6.3: Diagrama de Despliegue

6.4.2. Diagramas de Componentes

A continuación, se muestra mediante el diagrama de componentes de la Figura 6.4 los distintos componentes presentados a lo largo de la anterior sección.

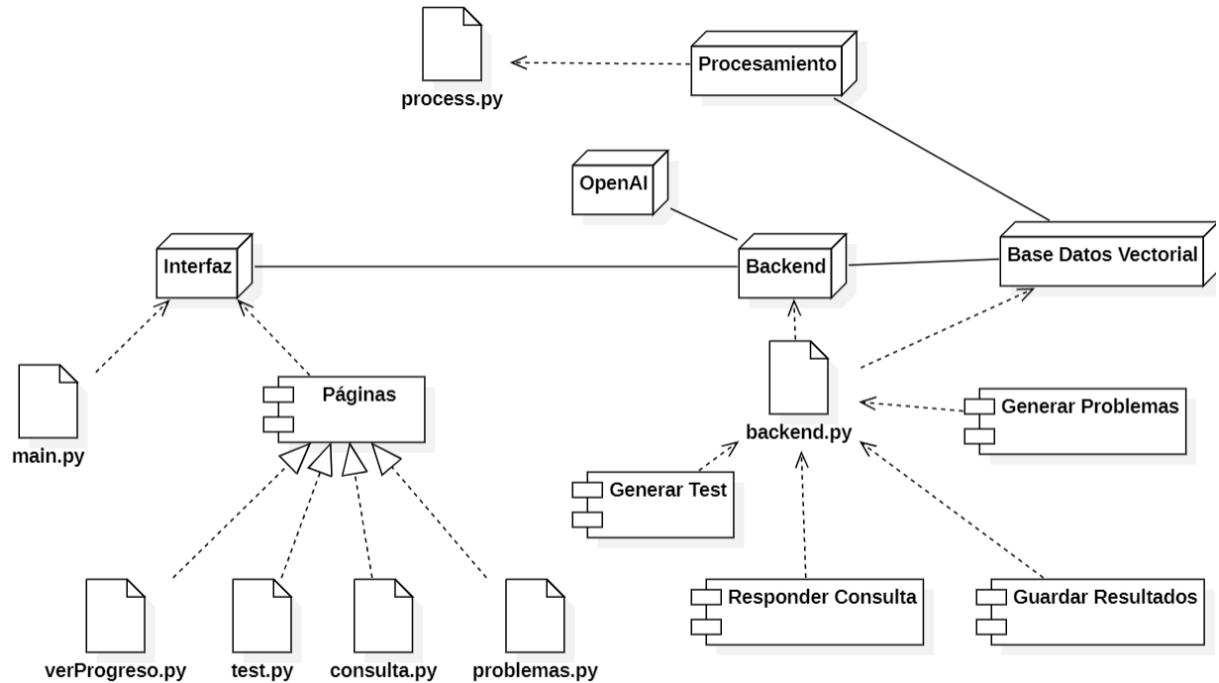


Figura 6.4: Diagrama de Componentes

6.4.3. Diagramas de Secuencia

Prosiguiendo con los diagramas de secuencia, estos se emplean para permitir modelar y representar la interacción, flujo y comportamiento del sistema. En consecuencia, se muestran los siguientes diagramas.

Consulta

En este apartado se detalla el comportamiento del caso de uso de Solicitar Pregunta Abierta en la Figura 6.5.

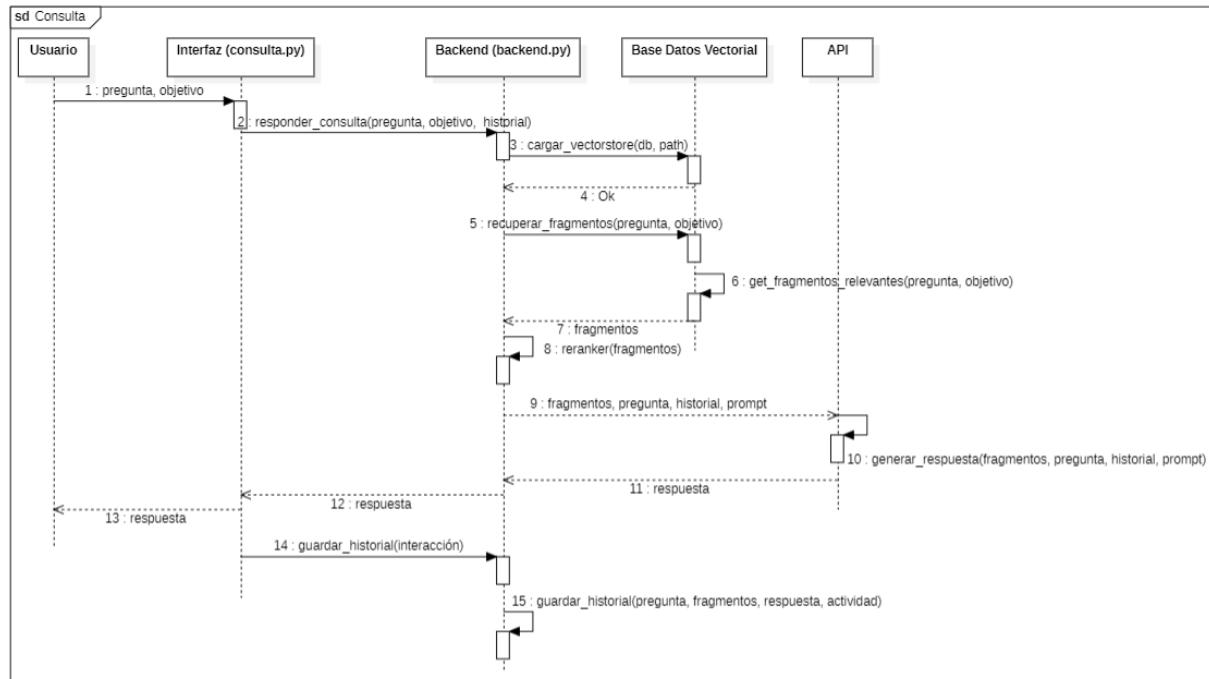


Figura 6.5: Diagrama Secuencia: Caso de Uso Consulta

Test

A continuación, se expone el comportamiento del caso de uso de Solicitar Pregunta Tipo Test en la Figura 6.6.

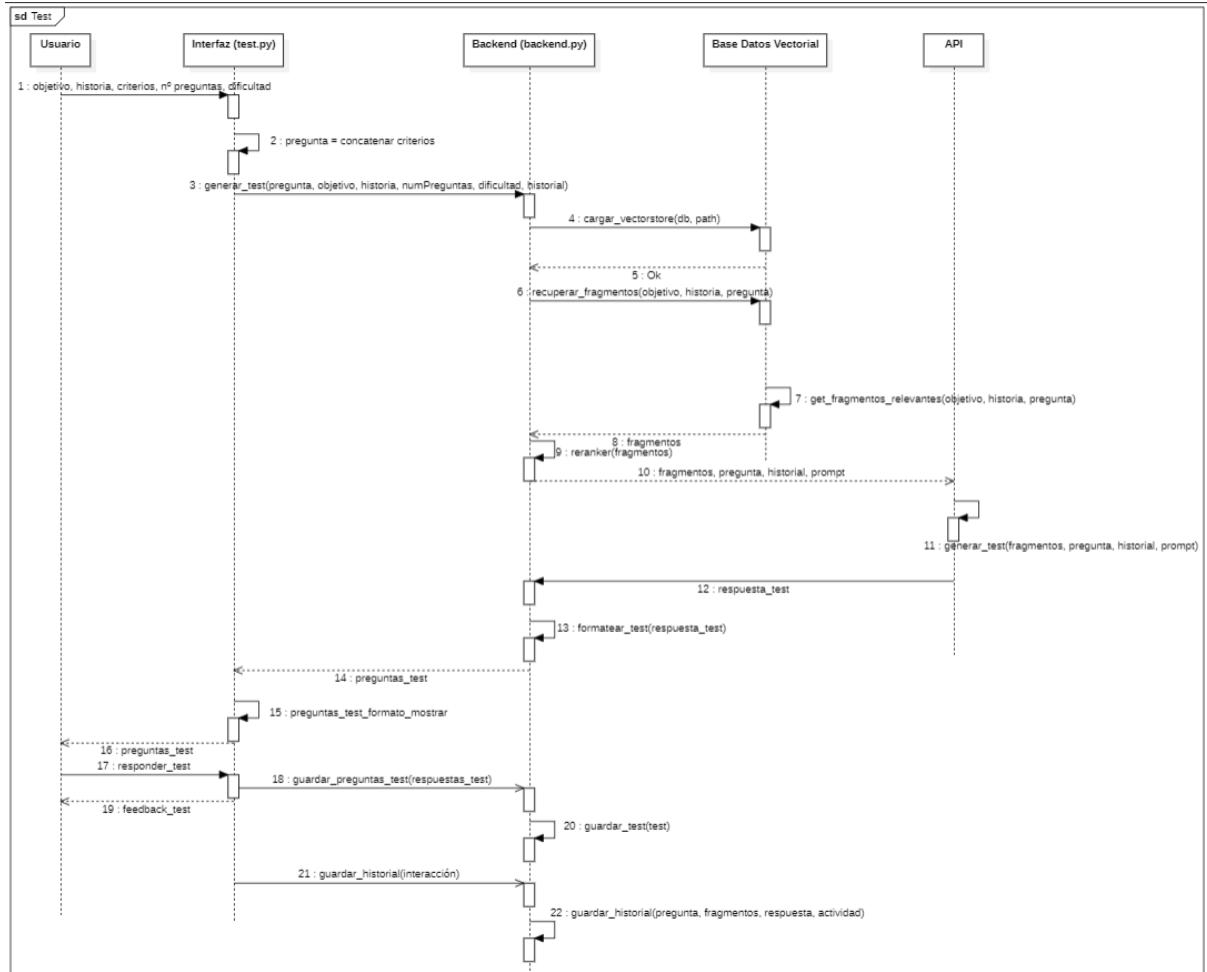


Figura 6.6: Diagrama Secuencia: Test

Problemas

En este apartado se especifica el comportamiento del caso de uso de Solicitar Supuesto Práctico en la Figura 6.7.

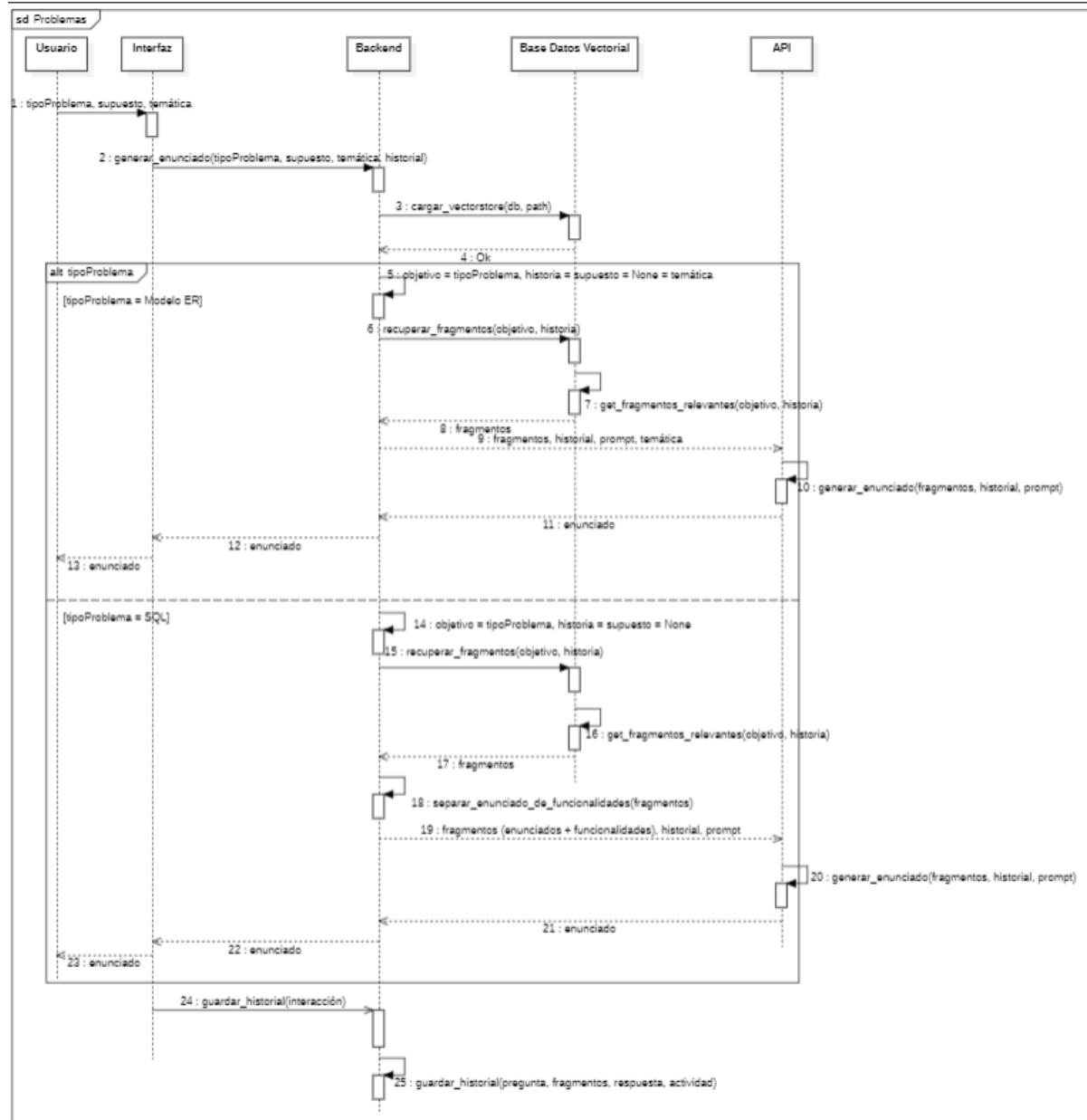


Figura 6.7: Diagrama Secuencia: Problemas

Ver Progreso

Finalmente, se muestra el comportamiento del caso de uso de Visualizar Progreso de Aprendizaje en la Figura 6.8.

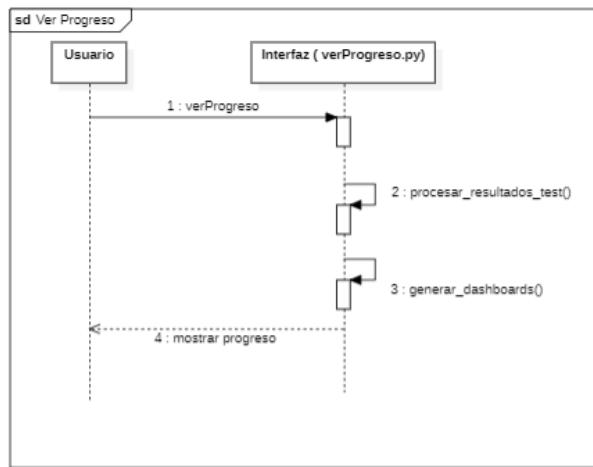


Figura 6.8: Diagrama Secuencia: Ver Progreso

6.4.4. Diagramas de Flujo

A continuación, se muestra un workflow desde el punto de vista del usuario. Para ello, se presenta el flujo de las actividades principales que permite realizar el asistente.

Inicialmente, la Figura 6.9 representa el proceso de una Consulta, donde se aprecia que el usuario en primer lugar introduce el objetivo y luego su pregunta. Después visualiza la respuesta, y si no continúa preguntando, finalizaría el flujo.

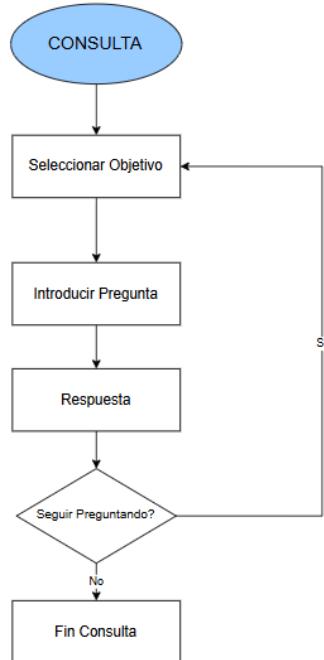


Figura 6.9: Consulta Flujo

La Figura 6.10 presentaría el flujo de Ver Progreso, el cual solo sería acceder a dicha actividad y ver los distintos dashboards presentados: no interactuaría más el usuario.



Figura 6.10: Ver Progreso Flujo

La actividad de Problemas expuesta en la Figura 6.11 recoge el flujo a seguir. De manera inicial, el usuario selecciona el tipo de problema. En función de su elección, indicaría el supuesto o temática, y finalmente visualizaría el problema generado. Si no quisiera solicitar otro problema, terminaría el flujo.

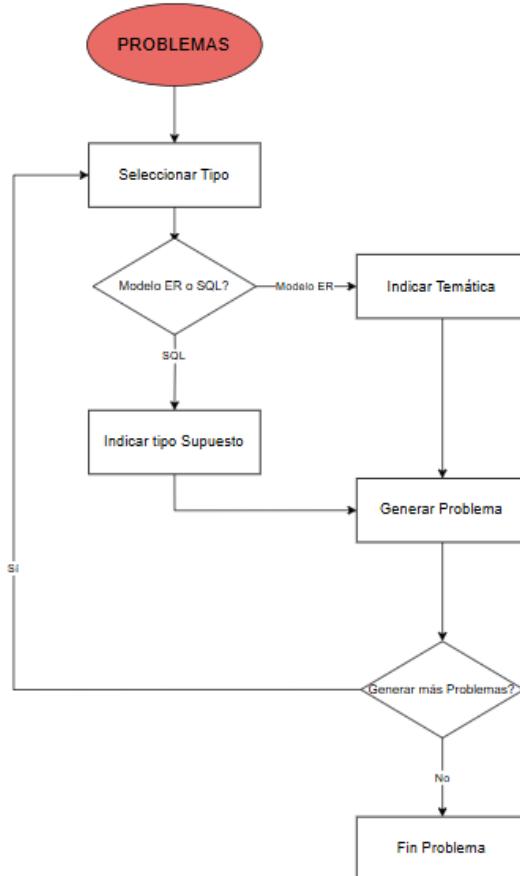


Figura 6.11: Problema Flujo

Por último, el flujo de la actividad Tipo Test representada en la Figura 6.12 se inicia con la configuración de una serie de parámetros, como el número de preguntas, objetivo, historias y criterios. Una vez generado test, el usuario visualizaría una por una las diferentes preguntas proporcionadas, las cuales también respondería de manera individual. Al finalizar el test, apreciaría el resultado de su desempeño obtenido, y si no quisiera continuar, terminaría la actividad.

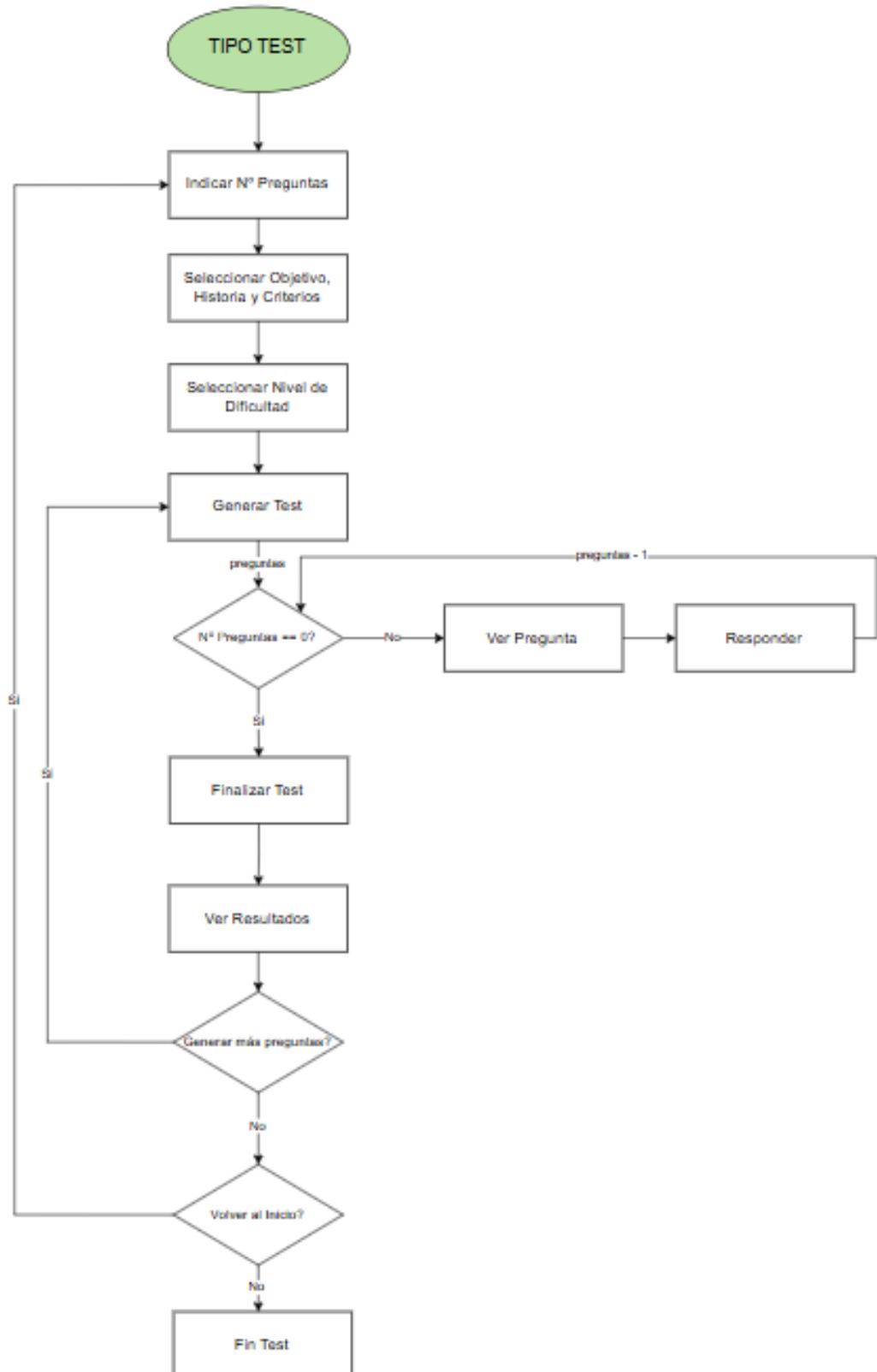


Figura 6.12: Test Flujo

6.4.5. Diagramas de Clases

Por otra parte, en la Figura 6.13 se presenta el diagrama de clases para describir la estructura del sistema mediante posibles relaciones, clases, atributos y métodos existentes.

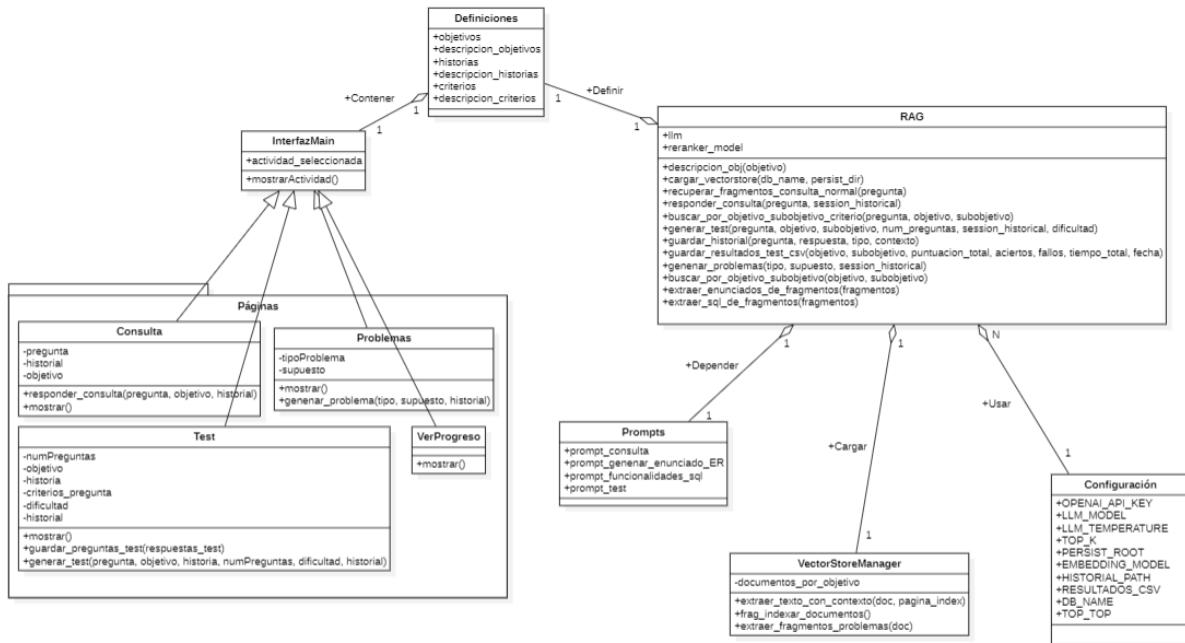


Figura 6.13: Diagrama de Clases

6.5. Datos

Desde el inicio, la construcción del asistente de aprendizaje se asienta sobre una asignatura perteneciente al Grado en Ingeniería Informática de Servicios y Aplicaciones: Sistemas de Bases de Datos. Por tanto, los documentos empleados como base de conocimiento corresponden al material didáctico oficial de la asignatura, fundamentalmente los PDFs del temario impartido en clase, así como una colección de enunciados y resoluciones de ejercicios prácticos.

La asignatura se divide en cinco grandes bloques temáticos: los objetivos generales, cada uno con sus propios documentos asociados, subdivididos conceptualmente en historias de aprendizaje, y a su vez, en criterios de aprendizaje. Esta estructura, consecuencia del marco eXtreme Learning, permite una organización clara y precisa del contenido, facilitando la personalización del aprendizaje y el seguimiento del progreso del estudiante.

Por una parte, los materiales teóricos se dividen de manera más clara en objetivos, historias de aprendizaje y criterios definidos en el enfoque de eXtreme Learning. Esto facilita significativamente la personalización del aprendizaje, ya que permite asistir a las necesidades específicas del estudiante en función de lo que quiera trabajar en cada momento. Además, gracias a esta estructura, es posible filtrar y localizar con precisión los fragmentos de contenido relevantes, mejorando la calidad y pertinencia de las consultas y evaluaciones.

Por otro lado, se presentan diversos materiales asociados con la problemática y parte práctica de la asignatura, los cuales se modelan de manera distinta a la teoría. Es más difuso vincular

únicamente un solo objetivo o historia en cada problema, ya que cada ejercicio práctico suele abordar múltiples conceptos de forma simultánea. Por este motivo, se propone un enfoque distinto para esta segunda parte, de tal manera que se presenta una colección de ejemplos de modelos de ER por un lado, y por el otro, un conjunto de ejercicios centrados en varios supuestos y funcionalidades de SQL. Esta separación permite una categorización más flexible, adaptada a los contenidos prácticos y ofreciendo por parte del asistente un apoyo específico según el tipo de problema planteado.

No obstante, en ambos es indispensable determinar con precisión en qué documentos debe buscar el asistente en cada momento, ya que de ello depende la calidad de la respuesta generada. En consecuencia, es fundamental poder relacionar cada material con las actividades que realice el asistente.

Para resolver esta necesidad, el sistema incorpora mecanismos de selección a través de la interfaz, lo que permite al usuario especificar el ámbito de la actividad. En el caso de las consultas y test (materiales teóricos), puede puntuizar en qué dominio, objetivos e historias de aprendizaje el alumno desea profundizar. Por su parte, en los materiales prácticos, el sistema permite seleccionar el tipo de problema o funcionalidad, como por ejemplo modelado ER o consultas SQL. Esta organización se ve reflejada en la arquitectura del *backend* mediante la existencia de tres módulos independientes, cada uno especializado en una actividad distinta. Con esto se consigue optimizar el rendimiento y la relevancia de las respuestas generadas.

Dicho esto, los documentos proporcionados según cada objetivo e historia de aprendizaje son los siguientes de la tabla 6.1. Destacar que estos serán los empleados para las consultas, tipo test y la visualización del progreso.

La estructura interna de cada pdf teórico, es una serie de diapositivas (una por página) con el contenido de la asignatura, cada una con un leve encabezado que actúa como índice de los criterios abarcados en cada una. Cabe destacar que estos datos no requieren de ninguna transformación adicional, ya que dicho índice resulta útil para recuperar la información. Todo ello se va a almacenar en una colección dentro de la base llamada `chroma_objetivos`, y se fragmentará en bloques por diapositiva. Es decir, se creará un fragmento por cada diapositiva, donde además se incluirá los párrafos de sus adyacentes para mejorar el contexto conceptual.

Por otro lado, resaltar la parte de los problemas, que se van a sumar en otra colección llamada `chroma_problemas`, con el objetivo de simplificar la lógica y la organización de metadatos. Esta base contiene un conjunto de ejemplos de enunciados de modelos Entidad - Relación y por otra parte, supuestos y funcionalidades SQL. Asimismo, comentar que en este caso sí que se ha tenido que hacer un proceso de transformación más elaborado de los datos. Inicialmente había únicamente dos pdfs de problemas: (uno de ER er.pdf y otro de SQL, sql.pdf), pero para mejorar la recuperación, se ha decidido:

- En el caso de los enunciados ER, adaptar cada ejemplo para que quepa en una única hoja.
- En el caso de los supuestos SQL, dividir cada caso en un PDF independiente (QuiereZon, ElNorte+, Flying your Dreams).

Además, dentro de estos supuestos de SQL, la parte que describe el contexto del problema y modelo lógico sobre la creación de la base de datos de clase se ha separado en páginas diferentes respecto a las funcionalidades de ejemplo. Esto permitirá facilitar en fases posteriores el proceso de enriquecimiento y aumento de las respuestas. Por tanto, la organización sería la siguiente:

Objetivo	Nombre del Objetivo	Documento	Historia
1	SGBDs	20240912_1.1_intro.pdf	Fundamentos de Bases de Datos
1	SGBDs	20240913_1.2_ciclo.pdf	Fundamentos de Bases de Datos
1	SGBDs	20241003_2.3_metodologia.pdf	Fundamentos de Bases de Datos
2	Conceptual	20240918_2.1_conceptual.pdf	Fundamentos de Diseño Conceptual
2	Conceptual	20241003_2.3_metodologia.pdf	Fundamentos de Diseño Conceptual
2	Conceptual	20241010_2.4_is_a.pdf	Fundamentos de Diseño Conceptual
3	Lógico	20240925_3.1_relacional.pdf	Fundamentos de Diseño Lógico
3	Lógico	20241003_2.3_metodologia.pdf	Fundamentos de Diseño Lógico
3	Lógico	20241010_2.4_is_a.pdf	Fundamentos de Diseño Lógico
3	Lógico	20241106_3.3_normalizacion.pdf	Fundamentos de Diseño Lógico
4	DDL	20241107_4.2_integridad.pdf	Integridad
4	DDL	20241128_4.3_privilegios.pdf	Privilegios
4	DDL	20241212_4.4_vistas.pdf	Vistas Relacionales
4	DDL	20241002_4.1_mysql_1.pdf	Fundamentos de Gestión de Datos
4	DDL	20241003_2.3_metodologia.pdf	Fundamentos de Gestión de Datos
4	DDL	20241002_4.1_mysql_1_2.pdf	Fundamentos de Gestión de Datos
4	DDL	20241107_4.2_integridad.pdf	Construcción de Bases de Datos
5	DML	20241017-24_5.1_dml_i.pdf	Fundamentos de Manipulación de Datos
5	DML	20241120_5.4_dml_ii.pdf	Fundamentos de Manipulación de Datos
5	DML	20241121_5.5_dml_iii.pdf	Fundamentos de Manipulación de Datos

Tabla 6.1: Distribución de Documentos por Objetivo e Historia

6.5.1. Base de Datos

Los materiales utilizados en el sistema se procesan mediante el archivo `process.py`. Este archivo se encarga de extraer el texto de los archivos PDF, dividirlos en fragmentos por páginas y realizar el preprocesamiento necesario para su almacenamiento en la base de datos. El sistema utiliza una base vectorial *ChromaDB* como base de datos para almacenar y gestionar los documentos procesados. Antes de la inicialización de la sesión de usuario, los documentos se cargan una sola vez, se procesan y se almacenan en *ChromaDB* en local, generando las representaciones vectoriales o *embeddings* a partir de los fragmentos de texto.

Paralelamente, en el caso de la teoría se añaden los distintos metadatos relacionados como: su objetivo, la historia de aprendizaje relacionada, la página del pdf. En el caso de la práctica, se incluirían: problemática y la página del pdf.

El archivo maneja este flujo de trabajo de manera eficiente al realizar el siguiente proceso:

1. **Extracción de texto:** se extrae el contenido de los documentos en formato PDF.
2. **División en fragmentos:** el texto extraído se divide en fragmentos de tamaño variado conforme cada diapositiva/página para facilitar la recuperación y el análisis.

Sección	Documento	Problemática	Descripción
Problemas	er.pdf	Modelo ER	Colección de enunciados de ejercicios de modelado entidad-relación. Cada enunciado se presenta en una página independiente.
Problemas	sql_QuiereZon.pdf	QuiereZon	Supuesto práctico de SQL. Página 1: contexto y modelo lógico. Página 2: ejemplos de funcionalidades
Problemas	sql_ElNorte+.pdf	ElNorte+	Supuesto práctico de SQL. Página 1: contexto y modelo lógico. Página 2: ejemplos de funcionalidades
Problemas	sql_Flying_your_Dreams.pdf	Flying your Dreams	Supuesto práctico de SQL. Páginas 1–2: contexto y modelo lógico. Página 3: ejemplos de funcionalidades

Tabla 6.2: Distribución de documentos relacionados con los problemas prácticos

3. **Generación de embeddings:** los fragmentos de texto se transforman en *embeddings* utilizando un modelo preentrenado de HuggingFaceEmbedings. Cabe resaltar que para mejorar el contexto, se incluyen también las dos últimas y primeras líneas de sus documentos adyacentes (excepto para el procesamiento de los documentos de problemas).

4. **Indexación y almacenamiento:** los *embeddings* generados se almacenan junto con los índices correspondientes en la base de datos. Asimismo, se añade la información de los metadatos.

El esquema de la base de datos es simple y consta de dos colecciones, una que alberga los contenidos teóricos y otra los prácticos. Destacar que, de manera indiferente, los campos en ambas son todos de tipo texto, a excepción del índice y la página:

1. **chroma_objetivos:**

- **Índice:** Identificador único de cada fragmento.
- **Objetivo:** El objetivo principal al que pertenece el fragmento (SGBDs, Diseño Conceptual, Diseño Lógico, DDL, DML).
- **Subobjetivo:** La historia de aprendizaje relacionada con el fragmento.
- **Página:** La página del pdf de donde se extrae, lo cual permite saber el orden.
- **Embeddings:** Los *embeddings* generados a partir de los fragmentos de texto.

2. **chroma_problemas:**

- **Índice:** Identificador único de cada fragmento.

- **Sección:** Problemas.
- **Problemática:** Tipo de problema a tratar. *Modelo ER* indica que se creará un enunciado relacionado con el modelado Entidad-Relación. *QuiereZon*, *ElNorte+* o *Flying your Dreams* hacen referencia a supuestos prácticos sobre los que se generará diversas funcionalidades para SQL específicas.
- **Página:** La página del pdf de donde se extrae, lo cual permite saber el orden.
- **Embeddings:** Las representaciones vectoriales generadas a partir de los fragmentos de texto.

Con este planteamiento, los documentos y los *embeddings* se mantienen en la base de datos local y no es necesario crearla ni reiniciarla en cada sesión, lo que optimiza el rendimiento y facilita la rapidez en las consultas del sistema.

6.5.2. Respuesta de Asistente

Otro aspecto a resaltar sería la necesidad de persistencia de las respuestas del asistente para facilitar una mejora continua del sistema. La conservación de estas respuestas es clave para el proceso de retroalimentación, ya que permite identificar qué respuestas son adecuadas y cuáles necesitan ser mejoradas, y por tanto, un paso esencial para la evaluación del rendimiento del asistente.

En concreto, todas las respuestas generadas por el asistente se almacenan en un archivo JSON, con una estructura del tipo:

```
{Tipo: "...", Pregunta: "...", Respuesta: "...", Contexto: "...”}
```

Donde como se puede apreciar, se guarda también la actividad (Tipo = {test, consulta, problema}) desde la cual ha sido generada la respuesta del asistente, así como el contexto recuperado para dicha contestación.

Este almacenamiento no solo sirve para mantener un historial de interacciones, sino que también permite la evaluación posterior de distintas herramientas y de los profesores de la asignatura mediante el envío de las respuestas en un archivo. En este archivo, el docente también podrá evaluar cada par pregunta–respuesta la calidad de la respuesta generada. Este proceso de evaluación es esencial para realizar ajustes y mejoras en las respuestas del asistente, asegurando que el sistema proporcione respuestas cada vez más precisas y relevantes. No obstante, esta parte se detallará con mayor profundidad en el Plan de Aceptación 9.3, donde se establecerán los criterios, herramientas y mecanismos específicos para la valoración y retroalimentación de las respuestas generadas.

6.5.3. Progreso Alumno

Finalmente, podría existir una lógica parecida al apartado anterior aunque en este caso con un enfoque sobre el comportamiento del alumno. Mediante un CSV, se guardaría las preguntas que realizarían el alumno en un día determinado, así como el número de cuestiones contestadas correctamente, de manera errónea, por categoría, puntuaje total y tiempo dedicado al asistente. Con ello, se podrían realizar los distintos dashboards presentados en la interfaz. En el caso de que hubiese gestión de alumnos, se llevaría acabo por medio de la base de datos. Sin embargo, como

no se ha implementado, se simula mediante la inserción de la fecha como dato. Las columnas del CSV serían 6.3:

Fecha, Objetivo, Subobjetivo, Puntuación Total, Aciertos, Fallos, Total Preguntas, Tiempo Total

Columna	Formato	Ejemplo
Fecha	Fecha (YYYY-MM-DD)	2025-06-04
Objetivo	Texto	Objetivo 1
Subobjetivo	Texto	Fundamentos de Bases de Datos
Puntuación Total	Entero	85
Aciertos	Entero	17
Fallos	Entero	3
Total Preg.	Entero	20
Tiempo Total	Time (segundos)	150 s

Tabla 6.3: Diccionario de datos CSV

6.6. Interfaz de usuario

A continuación, se presentan las distintas descripciones (Tablas 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10), y mockups (Figuras 6.14, 6.15, 6.16, 6.17, 6.18, 6.19, 6.20, 6.21) de la interfaz. No obstante, resaltar que de manera general el planteamiento radica en que en la parte izquierda se mostrará el menú con las distintas opciones, y en la derecha se cargará dinámicamente el contenido según cada tipo de actividad correspondiente, tal y como se refleja en la Figura 6.14.



Figura 6.14: Inicio

Consulta	
Descripción	El usuario podrá ser capaz de realizar una pregunta, duda o consulta al asistente, la cual se ha de relacionar con el objetivo que especifique
Activación	Al pulsar en “Consulta” en el menú de selección de actividad
Boceto	Figura 6.15
Eventos	<ol style="list-style-type: none"> 1. Especifica Objetivo 2. Realiza consulta 3. Pulsa botón Responder Consulta 4. Si resulta que es correcta la entrada, se genera la respuesta y la ve, junto con el historial actualizado. Si no, se muestra un aviso de advertencia y se vuelve al paso 2

Tabla 6.4: Interfaz Solicitar Pregunta Abierta: Consulta

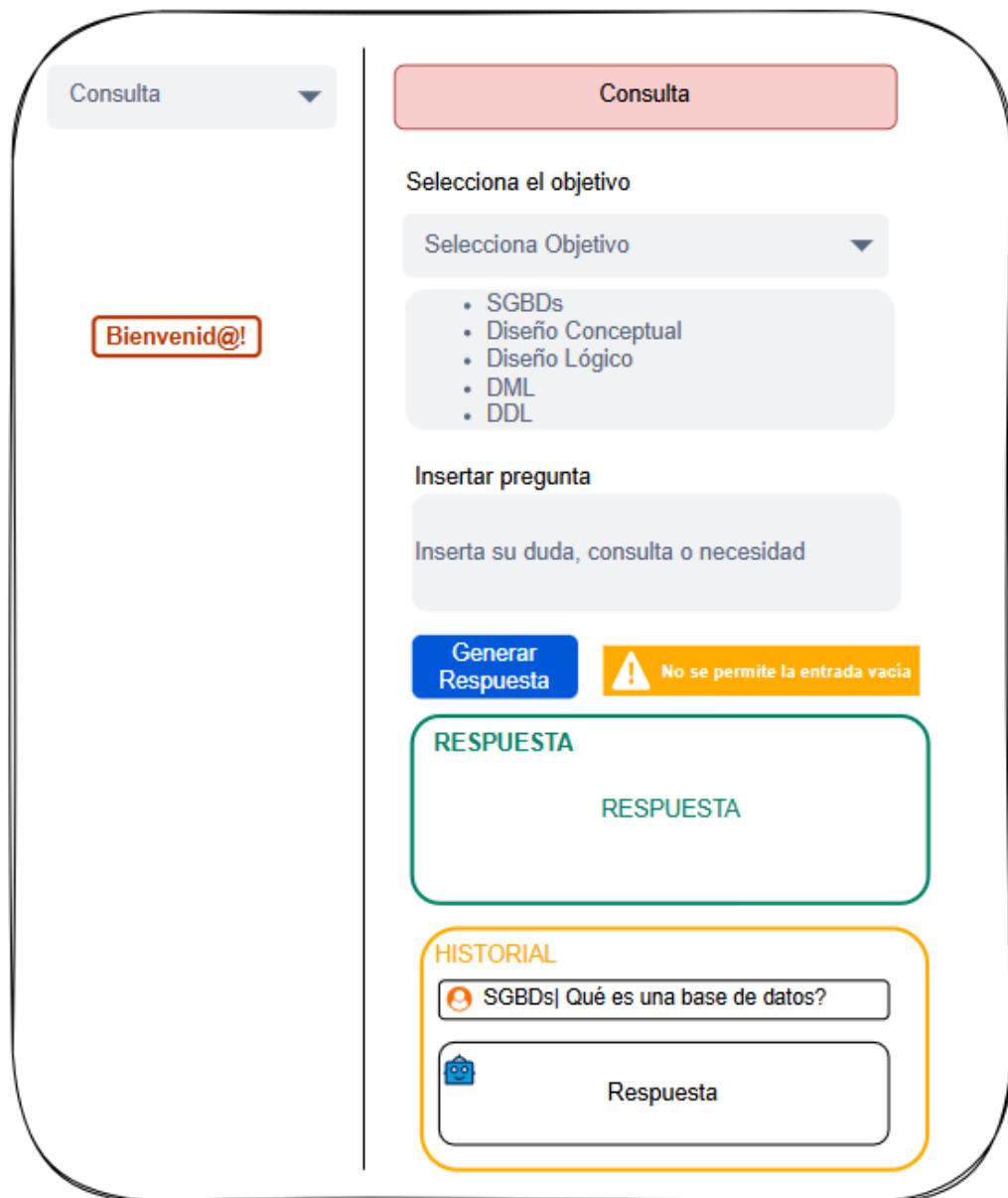


Figura 6.15: Consulta

Tipo Test	
Descripción	El usuario podrá ser capaz de reconfigurar los parámetros sobre los que se generará el test
Activación	Al pulsar en “Tipo Test” en el menú de selección de actividad
Boceto	Figura 6.16
Eventos	<ol style="list-style-type: none"> 1. Especifica Número de Preguntas 2. Indica Objetivo (2.1 Leer Descripción) 3. Selecciona Historia (3.1 Leer Descripción) 4. Añade Criterios (4.1 Leer Descripción – 4.2 Leer advertencia: si la entrada es vacía, se añadirán todos por defecto) 5. Señala Nivel de Dificultad 6. Pulsa botón Generar Test 7. Redirige a la interfaz 6.17 (solo si el <i>status</i> de la API es satisfactorio)

Tabla 6.5: Interfaz Solicitar Pregunta Tipo Test: Tipo Test (1/3)

The figure displays a wireframe of a user interface for generating a test. On the left, there's a sidebar with a dropdown menu labeled "Tipo Test". The main area is titled "Tipo Test" and contains the following fields:

- "Nº Preguntas": A slider set to 3, ranging from 1 to 10.
- "Selecciona el objetivo": A dropdown menu set to "Diseño Lógico".
- "Selecciona la historia": Two checkboxes: "Fundamentos de Diseño Lógico" and "Esquema Relacional".
- "Selecciona los criterios": A dropdown menu set to "Fundamentos de Diseño Lógico" and three buttons: "Columna", "Tabla", and "PK".
- A warning message: "Si no has seleccionado ninguno, se usarán todos por defecto" (If you have not selected any, all will be used by default).
- A blue "Generar Test" button at the bottom.

Figura 6.16: Tipo Test 1

Responder Test	
Descripción	El usuario podrá ser capaz de responder un cuestionario, en base a la configuración inicial, personalizada, que ha deseado
Activación	Al pulsar en "Generar Test"(y no error en la API)
Boceto	Figura 6.17
Eventos	<ol style="list-style-type: none"> 1. Ve Preguntas y Opciones generadas 2. Contesta Cuestionario 3. Presiona botón Finalizar Test. Si ha contestado a todas las preguntas, redirige a la interfaz 6.7. Si no, muestra advertencia sobre las cuestiones no contestadas, y se volvería al paso 2.

Tabla 6.6: Interfaz Solicitar Pregunta Tipo Test: Tipo Test (2/3)

The wireframe illustrates the 'Tipo Test' interface. It features a header with a dropdown menu labeled 'Tipo Test'. A red-bordered button labeled 'Bienvenid@!' is positioned on the left. The main content area contains two rounded rectangular boxes, each representing a question with a list of four options. Question 1 asks 'Pregunta 1: ¿Pregunta 1?' with options 'Opción 1', 'Opción 2', 'Opción 3', and 'Opción 4'. Question 2 asks 'Pregunta 2: ¿Pregunta 2?' with options 'Opción 1', 'Opción 2', 'Opción 3' (which has a green checkmark), and 'Opción 4'. Below the second question is a yellow warning bar containing a warning icon, the text 'Debes responder todas las preguntas.', and 'Pregunta que no has contestado: 1'. At the bottom right is a blue button labeled 'Finalizar Test'.

Figura 6.17: Tipo Test 2

Retroalimentación Test	
Descripción	El usuario podrá observar un feedback de su rendimiento en el test realizado en tiempo real
Activación	Al pulsar en "Finalizar Test", y con todas las preguntas contestadas
Boceto	Figura 6.18
Eventos	1. Ve retroalimentación del desempeño

Tabla 6.7: Interfaz Solicitar Pregunta Tipo Test: Tipo Test (3/3)

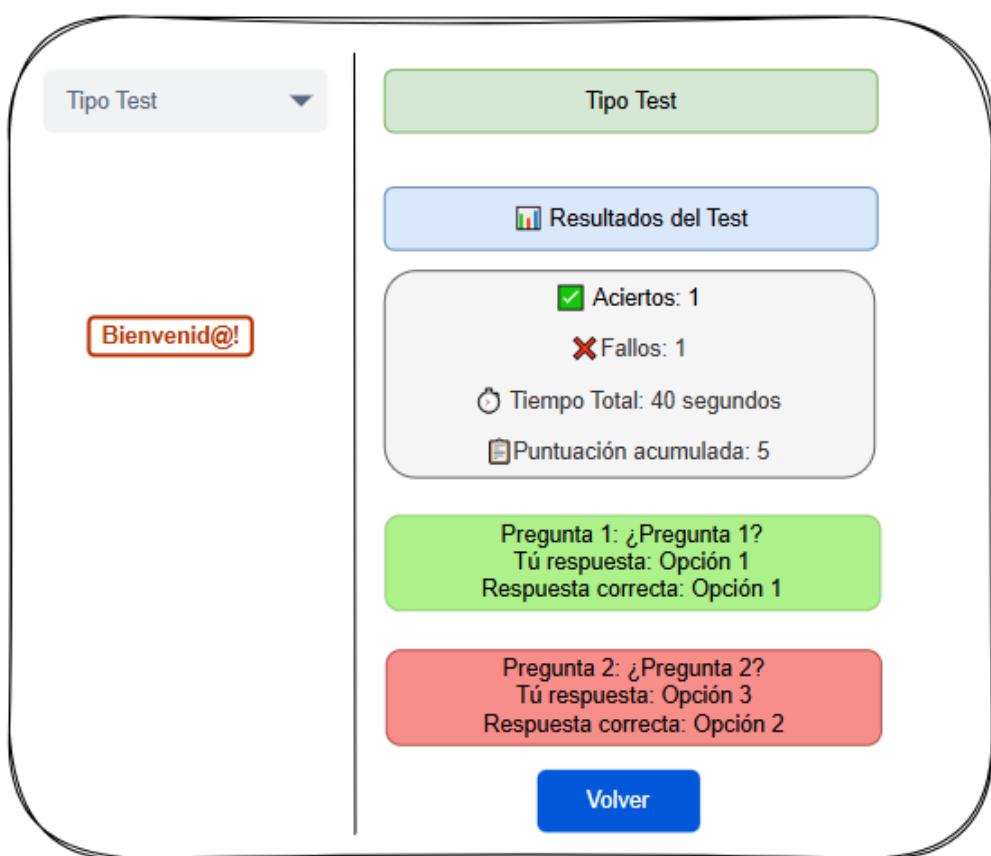


Figura 6.18: Tipo Test 3

Problemas - Modelo ER	
Descripción	El usuario podrá ver los enunciados generados en función del tipo de problema y supuesto o temática. En este caso, se tiene como tipo de problema: Enunciado Modelo ER
Activación	Al pulsar en “Problemas” en el menú de selección de actividad
Boceto	Figura 6.19
Eventos	<ol style="list-style-type: none"> 1. Especificar Tipo de Problema (Enunciado Modelo ER) 2. Indicar temática. (2.1. Si temática es Otro, entonces posibilidad de añadirla manualmente) 3. Pulsa botón Generar Problema 4. Ve Problema Enunciado Generado (solo si el <i>status</i> de la API es satisfactorio, es decir, no ERROR)

Tabla 6.8: Interfaz Solicitar Supuesto Práctico: Problemas (1/2)



Figura 6.19: Problemas 1

Problemas - SQL	
Descripción	El usuario podrá ver los enunciados generados en función del tipo de problema y supuesto o temática. En este caso, se tiene como tipo de problema: SQL
Activación	Al pulsar en “Problemas” en el menú de selección de actividad
Boceto	Figura 6.20
Eventos	<ol style="list-style-type: none"> 1. Especificar Tipo de Problema (SQL) 2. Indicar supuesto 3. Pulsar botón Generar Problema 4. Ver Problema Generado (más bien, funcionalidades y si no hay error en la API, si no ve mensaje de ERROR)

Tabla 6.9: Interfaz Solicitar Supuesto Práctico: Problemas (2/2)

The screenshot shows a user interface for generating practice problems. At the top, there's a navigation bar with a dropdown menu set to 'Problemas'. Below it, a main area has a title 'Problemas' and a sub-instruction 'Selecciona el tipo de problema' followed by a dropdown menu showing 'Enunciado Modelo ER'. Underneath, there's a section for 'Selecciona la temática' with checkboxes for 'Deportes', 'Comercio', 'Educación', and 'Otro' (with a placeholder 'Especifica la temática'). A large blue button labeled 'Generar Problema' is centered. To the right, there's a green-bordered box labeled 'RESPUESTA' containing the word 'RESPUESTA'. At the bottom, there's a yellow-bordered box labeled 'HISTORIAL' containing entries for 'Modelo ER | Deporte' and 'Respuesta'.

Figura 6.20: Problemas 2

Ver Progreso	
Descripción	Se mostrará mediante un análisis el progreso del usuario en relación al número de cuestiones que durante la sesión ha respondido: total, correctas y erróneas; y también respecto a las distintas secciones/actividades proporcionadas por el asistente (en caso de que haya una interacción previa)
Activación	Al pulsar en “Ver Progreso” en el menú de selección de actividad
Boceto	Figura 6.21
Eventos	1. Ve KPIs, tabla y diagramas de análisis del desempeño de los test.

Tabla 6.10: Interfaz Visualizar Progreso de Aprendizaje: Ver Progreso

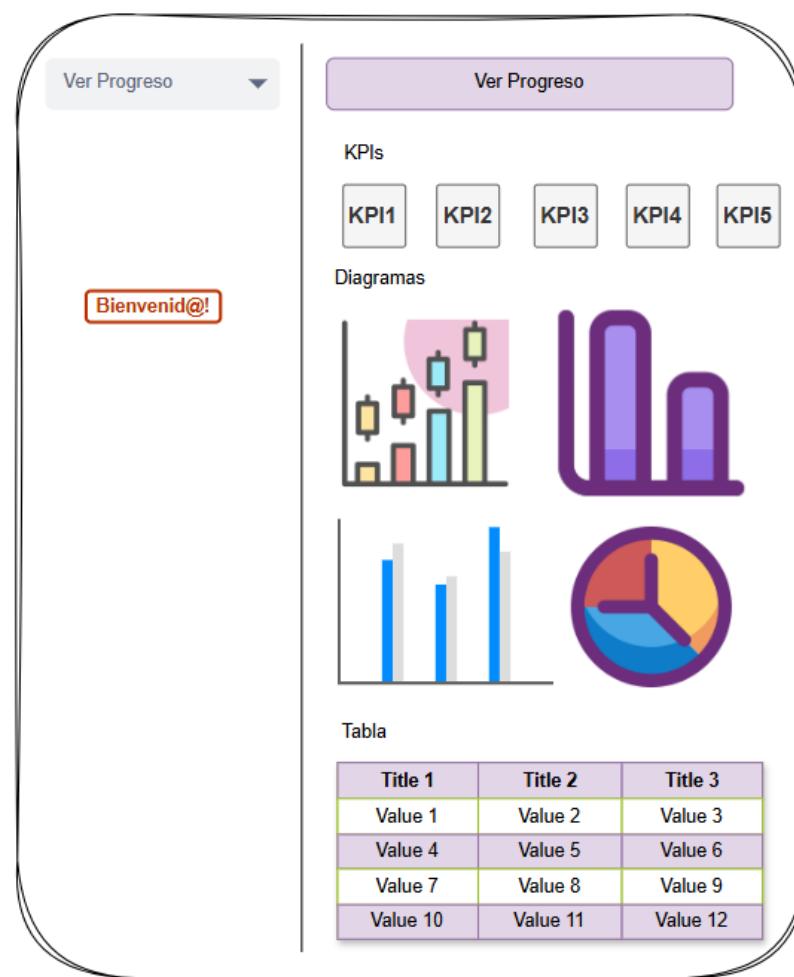


Figura 6.21: Ver Progreso

Capítulo 7

Implementación

En este apartado se recoge el proceso de construcción del producto, incluyendo decisiones tomadas, problemas encontrados y soluciones aplicadas. Por ello, se detalla cómo se ha llevado a cabo la implementación del asistente de aprendizaje conforme al diseño previamente establecido. A lo largo de este proceso, se ha procurado garantizar que el sistema funcione de manera eficiente, cumpliendo con los objetivos planteados en el proyecto y asegurando una experiencia de usuario satisfactoria.

7.1. Herramientas

Anteriormente, en el apartado 4.5.6, se presentaron una serie de herramientas que permitían integrar los múltiples componentes del RAG, y por tanto, imprescindibles para la construcción del asistente de aprendizaje. En base a ello, en este apartado se seleccionarán las tecnologías específicas que finalmente se emplearán, atendiendo además a criterios como la adecuación a los objetivos del proyecto, la compatibilidad entre componentes y la facilidad de integración. Asimismo, se ha priorizado el rendimiento, la escalabilidad y el uso de herramientas ampliamente adoptadas en el desarrollo de soluciones educativas basadas en IA. Por tanto las herramientas principales a utilizar para el desarrollo del asistente son:

- **Python:** lenguaje principal de desarrollo, seleccionado por su versatilidad, facilidad de uso y amplio ecosistema en el ámbito de la IA y procesamiento de datos.
- **LangChain:** se ha decidido emplear este *framework* debido a que facilita la creación de sistemas RAG, además de ser muy extendido, popular y compatible con diversos LLMs y almacenamientos vectoriales. Permite estructurar cadenas de recuperación y generación de forma modular, integrando herramientas como ChromaDB, HuggingFaceEmbeddings y OpenAI. Su uso simplifica la lógica de negocio y mejora la mantenibilidad del proyecto. LangChain facilita la conexión y orquestación entre las herramientas propuestas, haciendo que la integración y el uso de modelos de lenguaje y RAG sean más accesibles y eficientes.
- **Streamlit:** *framework* recurrido para el desarrollo de la interfaz gráfica del asistente. Su enfoque minimalista y declarativo facilita la creación rápida de aplicaciones web o demos. Además, permite la gestión de estado de sesión sin necesidad de un sistema de autenticación completo, útil para funciones como el seguimiento del progreso. A diferencia de las

aplicaciones web tradicionales basadas en HTML, en Streamlit no se definen explícitamente etiquetas HTML para construir la interfaz. En su lugar, la estructura de la página se genera dinámicamente mediante funciones de Python, lo que permite una actualización reactiva del contenido ante cualquier interacción del usuario. Finalmente, también se ha seleccionado por su sencillez, buena integración y manejo de datos en dashboards.

- **ChromaDB:** respecto al almacenamiento, se ha decidido fundamentarse en la base de datos vectorial local ChromaDB, ya que permite almacenar, indexar y consultar *embeddings* de forma eficiente. Esto se debe a que ofrece la posibilidad de guardar directamente las representaciones vectoriales, junto a distintos metadatos, proporcionando una persistencia local. Además, es *open source*, con documentación accesible, soporte activo, fácil de usar y con buena escalabilidad. ChromaDB también tiene una fácil integración con LangChain, lo que permite la realización de búsquedas rápidas y precisas sobre los documentos procesados. Su integración se realiza mediante la instrucción: `from langchain.vectorstores import Chroma`.
- **OpenAI:** utilizado para acceder a modelos LLM a través de su API en la nube, permitiendo aprovechar modelos avanzados como GPT-3.5 Turbo o GPT-4 sin necesidad de requerir una infraestructura local. La decisión de seleccionar OpenAI se debe a que se dispone de un presupuesto destinado a recursos de *software*, lo cual facilita el acceso a estas tecnologías de manera rápida y escalable. Sin embargo, comentar que en caso contrario, podría haberse considerado opciones alternativas como Ollama, que permite ejecutar modelos localmente. Por otro lado, el modelo a emplear será GPT-3.5 Turbo, ya que ofrece una mejor relación coste-calidad en las respuestas gracias a su mayor capacidad para la generación de texto coherente, creativo y congruente. Además, también se integra con LangChain a través de la instrucción: `from langchain.llms import OpenAI`. Esta solución facilita una integración rápida y escalable, lo que permite desarrollar un asistente de gran precisión, utilidad y con acceso a los últimos avances en IA y LLM.
- **Hugging Face Transformers:** para la generación de *embeddings* a partir de texto, se ha decidido emplear esta plataforma debido a que alberga modelos de lenguaje ya entrenados para ese fin, (también para *reranking*), y además, sin costes adicionales. En concreto, se utilizará `all-MiniLM-L6-v2` para la creación de *embeddings* y `ms-marco-MiniLM-L-6-v2` para el *reranking*, debido a su alto rendimiento, capacidades y popularidad [336, 277]. Se integra a través de LangChain mediante la instrucción: `from langchain.embeddings import HuggingFaceEmbeddings`.
- **Fitz:** biblioteca de Python utilizada para el procesamiento de documentos PDF, que permite extraer el contenido textual del temario y de los ejercicios de la asignatura. Se importa mediante: `import fitz`. Cabe destacar que también es conocida como PyMuPDF.
- **JSON:** librería utilizada para guardar las respuestas generadas por el asistente en un archivo JSON. Su estructura clave–valor lo hace adecuado para registrar pares pregunta–respuesta de forma sencilla y legible, permitiendo su posterior evaluación o reutilización.
- **CSV:** biblioteca incorporada en Python que facilita la lectura y escritura de archivos CSV. Su uso resultará en que luego será más fácil crear los dashboards.

- **Otras herramientas:** se prevé la necesidad de otras librerías, como NumPy o Pandas, para el procesamiento y análisis de datos y Plotly para la creación de visualizaciones interactivas, como dashboards, que serán incorporados en el proyecto. Además, también se empleará los módulos time y datetime para registrar la duración de los tests y la fecha en que se realizan. Por último, el módulo os se requerirá para acceder y gestionar archivos y directorios del sistema, como los archivos JSON, CSV o la propia base de datos de Chroma.

7.2. Desarrollo

En esta sección se expondrá la implementación del asistente de aprendizaje. Inicialmente, se detallará el desarrollo según la estructura del proyecto, más tarde, se determinará el funcionamiento del sistema y finalmente, se presentarán una serie de desafíos y adversidades de su implementación. En conclusión, a continuación se presenta tanto la construcción del producto como los distintos problemas encontrados.

7.2.1. Estructura

La estructura del proyecto se organiza en base a una jerarquía de archivos y directorios, la cual se muestra en la Figura 7.1

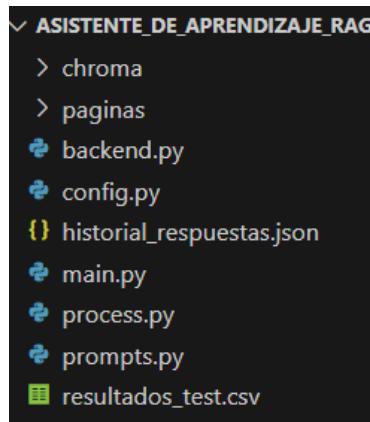


Figura 7.1: Estructura Proyecto

A continuación, se describe el propósito de cada archivo y directorio, así como sus responsabilidades y funciones importantes para el desarrollo del asistente.

7.2.2. Procesamiento de Datos y Creación de la Base de Datos: process.py

En este archivo, como bien se indica en el título, se procesan los datos (que son los materiales proporcionados) y se procede a la creación y carga de la base de datos vectorial Chroma. Para ello, se lleva a cabo la siguiente implementación y planteamiento.

En primer lugar, se define una variable `documentos_por_objetivo` que organiza los archivos PDF utilizados por el sistema según su objetivo e historia y, en el caso de la parte práctica, se clasifican por su problemática.

A partir de entonces, se procedería a la fragmentación e indexación (y con ello la conversión a *embeddings*) de los documentos. Para ello, se ejecuta la función `frag_indexar_documentos()` descrita en el código 7.1. Inicialmente, crea la base de Chroma con un *vectorstore* y mediante el objeto `Chroma()` proporcionado por Langchain, pero diferenciando las dos colecciones para almacenar los datos de forma organizada. En cada una resaltar que se incluye el directorio, el nombre de la colección y la función que se usará para la construcción de las representaciones vectoriales.

```
# Modelo para generar los embeddings
embeddings = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

def frag_indexar_documentos():
    # Directorio donde se almacena
    persist_dir = './chroma'

    # Crear Chroma

    # Colección general -> teórico
    vectordb_general = Chroma(
        collection_name='chroma_objetivos',
        embedding_function=embeddings,
        persist_directory=persist_dir
    )

    # Colección problemas -> práctica
    vectordb_problemas = Chroma(
        collection_name='chroma_problemas',
        embedding_function=embeddings,
        persist_directory=persist_dir
    )
```

Fragmento de Código 7.1: Creación Colecciones

Una vez inicializadas ambas colecciones, la función iterá sobre la variable definida anteriormente. Para los archivos con objetivos del 1 al 5, se extrae como fragmento cada diapositiva y se concatena con dos líneas de la páginas anterior y siguiente, para mejorar la calidad de los *embeddings* (`Extraer texto con contexto = 2LíneasPágAnterior + FragmentosEsaPágina + 2LíneasPágPosterior`). En cambio, para los problemas, no se añadiría este contexto adicional.

Cada representación de cada fragmento de texto se almacena en la colección correspondiente junto con los metadatos que describen el contenido (vistos en la sección 6.5). Estos metadatos permitirán filtrar la información para una mejor búsqueda en el *vectorstore* y todo se guarda en la colección mediante: `vectordb_problemas.add_texts([texto], metadatas=[metadatos])` y `vectordb_general.add_texts([texto_con_contexto], metadatas=[metadatos])`

Finalmente, se incluiría dentro de la función los métodos `vectordb_general.persist()` y `vectordb_problemas.persist()` para que persista la información almacenada y se asegure que se encuentre disponible en futuras consultas o acceso a Chroma.

7.2.3. Configuración config.py

Este archivo contiene las variables globales que se emplearán en `backend.py`, y son tales como: el modelo LLM de OpenAI, la temperatura de generación, número de *embeddings* a recuperar, el modelo de *embeddings* para ello, el modelo del *reranking*, el número de *embeddings* finales, las rutas a archivos y la clave a OpenAI:

- Clave API:
 - `OPENAI_API_KEY`: clave utilizada para acceder al modelo LLM de OpenAI.
- Parámetros del modelo de lenguaje (LLM):
 - `LLM_MODEL = "GPT-3.5-turbo"`: modelo de lenguaje utilizado para la generación de respuestas.
 - `LLM_TEMPERATURE = 0.3`: grado de aleatoriedad en la generación de texto. Con este valor, se asegura que la respuesta se ciña al contexto recuperado.
- Recuperación de información:
 - `TOP_K = 10`: número de documentos recuperados inicialmente por *embeddings*.
 - `TOP_TOP = 3`: número de documentos seleccionados tras el *reranking*.
 - `PERSIST_ROOT = "./chroma"`: ruta donde se almacenan la base de datos vectorial.
 - `DB_NAME_OBJETIVOS = "chroma_objetivos"` y `DB_NAME_PROBLEMAS = "chroma_problemas"`: nombres de las colecciones de Chroma diferenciadas para teoría y práctica.
- Modelo de embeddings:
 - `EMBEDDING_MODEL = "sentence-transformers/all-MiniLM-L6-v2"`: modelo utilizado para realizar los *embeddings* (transformar a representaciones vectoriales texto).
 - `RERANKING_MODEL = CrossEncoder("cross-encoder/ms-marco-MiniLM-L-6-v2")`: reordena los resultados con mayor precisión¹.
- Archivos de entrada y salida:
 - `HISTORIAL_PATH`: archivo donde se guarda el historial de interacciones.
 - `RESULTADOS_JSON` y `RESULTADOS_CSV`: rutas donde se almacenan los resultados de evaluación del asistente.

Con esta estructura, se permite modificar cualquier valor global desde un único punto, consintiendo el sistema más mantenible y flexible, ya que evita tener que actualizar manualmente cada instancia donde se empleen dichos parámetros. Resaltar que para poder diferenciarlas del resto, se definirán en mayúsculas.

¹Un *Cross-Encoder* evalúa el par (consulta, documento) conjuntamente para estimar la relevancia.

7.2.4. RAG backend.py

Este archivo alberga tanto las fases como los componentes principales de la técnica RAG. No obstante, se detallará en función a los módulos vistos anteriormente.

Antes de nada, en primer lugar, es necesario la importación del archivo `config.py` con las configuraciones y parámetros globales, así como `paginas.definiciones` con las descripciones de cada objetivo; `prompts.py`, con los prompts base, y las distintas librerías necesarias para la realización del RAG. A continuación, se crearía una instancia del modelo LLM de OpenAI de manera general 7.2:

```
llm = ChatOpenAI(
    model_name=LLM_MODEL,
    temperature=LLM_TEMPERATURE,
    openai_api_key=OPENAI_API_KEY
)
```

Fragmento de Código 7.2: Inicializar LLM de OpenAI y API

Prosiguiendo ahora con cada módulo, se comienza con el supuesto de la consulta.

Consulta

La función `responder_consulta_con_objetivo(pregunta, objetivo, session_historical)` es el núcleo para la implementación cuando el usuario selecciona la actividad consulta. En ese momento se invocaría y devolvería la respuesta a la pregunta. Su funcionamiento sería el siguiente:

1. Preparación: se recibe el historial de la sesión de las consultas y se transforma a texto plano para incluirlo como contexto (`session_text`).
2. Fase *Retrieval* - Recuperación de fragmentos: se invoca `buscar_por_objetivo(pregunta, objetivo)` que:
 - Carga el *vectorstore* con `cargar_vectorstore(DB_NAME_OBJETIVOS)` (pasándole como parámetro la colección de los objetivos). Esta función carga una colección de Chroma desde disco 7.3:

```
def cargar_vectorstore(db_name):
    persist_dir=PERSIST_ROOT
    embeddings =
        HuggingFaceEmbeddings(model_name=EMBEDDING_MODEL)

    vector_store = Chroma(
        collection_name=db_name,
        embedding_function=embeddings,
        persist_directory=persist_dir
    )
    return vector_store
```

Fragmento de Código 7.3: Cargar Vectorstore

- Crea el filtro por objetivo.

- Realiza búsqueda semántica con similitud coseno:
`resultados = vectorstore.similarity_search(query=pregunta, k=TOP_K, filter=filtro)`. Con este método, primero se recuperan los metadatos en función del objetivo establecido, a continuación, se realiza la búsqueda semántica en ese subconjunto y por último, se recuperan TOP_K fragmentos (variable definida en config.py)
- Además, en todo momento se manejan posibles errores o excepciones. Por ejemplo, que `vectorstore` no cargue bien o que no encuentre ningún `embedding` para ese objetivo, en este caso, se realizaría la búsqueda respecto a todos los `embeddings` de la base de datos, sin el filtro.

3. Reranking:

- Con los `embeddings` recuperados en la fase anterior, se crea una lista de pares, donde cada elemento es (pregunta, contenido_fragmanto).
- Se calcula la relevancia del contenido del fragmento respecto a la pregunta de cada par obtenido mediante el modelo de `reranking`.
- Se ordenan, y seleccionan los 3 más relevantes. Este será el contexto recuperado final incluido en la variable: `context`.

4. Preparación del prompt (*Augmented*) y llamada al modelo (*Generation*) 7.4:

```
# Definicion Prompt
prompt = PromptTemplate(
    template=prompt_consulta,
    input_variables=["context", "session", "query"]
)

#Construcion de la cadena
chain = (
    RunnableMap({
        "context": lambda _: context,
        "session": lambda _: session_text,
        "query": lambda _: pregunta
    })
    | prompt
    | llm
)

# Ejecucion
respuesta = chain.invoke({}))
```

Fragmento de Código 7.4: Prompt - Modelo Consulta

`PromptTemplate`, crea una plantilla prompt donde se incluirán tres variables: `context`, `session` y `query`. Cabe destacar que al usar un `template`, dichas variables del `input` serán obligatoriamente requeridas.

`prompt_consulta`, es el prompt para la consulta, el cual se encuentran en `prompts.py`.

`chain`, es una cadena de procesamiento, o pipeline, que permite concatenar con “|” distintas operaciones para generar una respuesta. Esta consta de:

`RunnableMap()`, que permite mapear de forma explícita las entradas del prompt, actuando como un diccionario dinámico de variables. En este caso, toma las variables creadas `context`, `session_text` y `pregunta` y las asigna respectivamente a las claves `context`, `session` y `query` que el prompt espera recibir. `lambda _`, es la función que permite devolver el valor actual de la variable, sin importar el argumento recibido. Todo ello sería la parte del *Augmented*, ya que se aumenta, refina y mejora el prompt que se enviará al LLM. Resaltar que se ha optado por esta herramienta debido a que dependiendo de la actividad (e incluso dentro de la misma interacción) varía tanto el contenido como el número de las variables de entrada al prompt, entonces resulta en la forma más eficaz de construirlo. Por ejemplo, para una consulta se necesitan las variables `context`, `session`, `query`, pero en otros casos, como en la parte del test, se requerirán de más. En definitiva, `RunnableMap()` ofrece la flexibilidad para construir dinámicamente la cadena de procesamiento adecuada para cada situación.

`prompt`, inserta los valores anteriores en la plantilla del prompt.

`llm`, pasa el prompt al `llm` para generar la respuesta.

Finalmente, se invoca a `invoke()`, el cual lanza el `chain`, ejecutándolo en cada paso y devuelve la respuesta generada por el modelo. Es decir, envolvería la fase *Generation* de RAG. Cabe resaltar que se valida dicha llamada a la API, encapsulándola en un `try-except` para capturar posibles excepciones y controlar los mensajes de error que se devolvería en la salida de la interfaz.

5. Guardar del historial. Se guarda la pregunta, la respuesta generada por el asistente, el tipo de actividad y los fragmentos recuperados o contexto utilizado mediante la función 7.5:

```
def guardar_historial(pregunta, respuesta, tipo,
                      contextos=None)
    # Se crea entrada:
    entrada = {
        "tipo": tipo,
        "pregunta": pregunta,
        "respuesta": respuesta,
        "contextos": contextos if contextos else []
    }
    #Y se guardaria esta entrada en el json con:
    historial.append(entrada)
```

Fragmento de Código 7.5: Guardar historial

Por medio de la llamada: `guardar_historial(pregunta, respuesta.content, “consulta”, contextos=context)`, que se realiza tras la ejecución de la llamada a la API del punto anterior.

Destacar que también se validaría si el archivo existe o no, si la escritura ha sido correcta o no y que solo se invoca una vez que la respuesta de la API ha sido correcta, es decir, no ha habido una excepción.

6. Finalmente, en el `return` se incluye el contenido de la respuesta generada por el LLM: `respuesta.content` y, de manera opcional, también se podrían devolver los fragmentos de los documentos recuperados.

Test

Para generar el test, se sigue un proceso similar al anterior, salvo que difiere en las funciones que emplea y en que introduce un nuevo flujo en el proceso. El curso de esta parte comienza con la función `generar_test(pregunta, objetivo, subobjetivo, num_preguntas, session_historical, dificultad)`.

1. Al igual que en la consulta, se prepara el historial para el test, que contiene las preguntas que ha generado y todas sus opciones múltiples.
2. Fase *Retrieval* - Recuperación de fragmentos: se invoca `buscar_por_objetivo_subobjetivo_criterio (pregunta, objetivo, subobjetivo)` que:
 - De la misma manera que antes, carga el *vectorstore* con `cargar_vectorstore(DB_NAME_OBJETIVOS)`
 - Crea el filtro por objetivo y subobjetivo (o historia), y se concatena con '\$and'
 - Realiza búsqueda semántica con similitud coseno:
`resultados = vectorstore.similarity_search (query=pregunta, k=TOP_K, filter=filtro)`
 - Además, en todo momento se manejan posibles errores o excepciones
3. *Reranking* (con la misma lógica que en el caso anterior).
4. Preparación del prompt (Augmented) y llamada al modelo (Generation): 7.6

```
# Definicion Prompt
prompt = PromptTemplate(
    template = prompt_test,
    input_variables=["context", "session",
                    "num_preguntas", "dificultad", "objetivo",
                    "subobjetivo", "objetivoDescripcion",
                    "subobjetivoDescripcion", "query"]
)

#Construccion de la cadena
chain = (
    RunnableMap({
        "context": lambda _: context,
        "session": lambda _: session_text,
        "num_preguntas": lambda _: num_preguntas,
        "dificultad": lambda _: dificultad,
        "objetivo": lambda _: objetivo,
        "subobjetivo": lambda _: subobjetivo,
        "objetivoDescripcion": lambda _:
```

```

        objetivo_descripcion ,
        "subobjetivoDescripcion": lambda _:
            subobjetivo_descripcion ,
            "query": lambda _: pregunta ,
        })
        | prompt
        | llm
    )

# Ejecucion
respuesta = chain.invoke({})

```

Fragmento de Código 7.6: Prompt - Modelo Test

Como se puede apreciar, en esta situación, se añaden como variables el contexto, el historial de la sesión, la pregunta, el número de preguntas a generar, la dificultad y los objetivos e historias junto a sus descripciones (a las cuales se acceden a raíz del archivo `definiciones.py`).

5. Al igual que antes, se guarda el historial, esta vez con `guardar_historial(pregunta, respuesta.content, "test", contextos=context)` y de la misma manera, se invoca cuando no ha habido una excepción previa en la llamada a la API.

Hasta este punto, el procedimiento seguido es semejante al descrito previamente. No obstante, en el caso del test, es necesario incorporar una serie de pasos adicionales para el correcto procesamiento del resultado.

6. Parsear las preguntas: la respuesta generada por el modelo sigue una estructura compuesta, en la que se incluye, para cada pregunta, el enunciado, las distintas opciones de respuesta, y finalmente, la opción considerada correcta. Debido a ello, se construye una lista llamada `preguntas_final`, que permite separar y almacenar cada una de estas partes de forma individual, facilitando su posterior tratamiento:

```

preguntas_final.append({
    "pregunta": enunciado,
    "opciones": opciones,
    "correcta": letra_correcta
})

```

Fragmento de Código 7.7: Estructura de cada pregunta parseada

Además, se efectúa un preprocesamiento adicional del texto (escapado) para poder identificar correctamente la opción marcada como correcta, resultando en una complejidad extra.

7. Finalmente, en el `return` se incluiría esta variable `preguntas_final`.
8. Por último, es importante señalar la existencia de la función `guardar_resultados_test_csv(objetivo, subobjetivo, puntuacion_total, aciertos,`

`fallos, tiempo_total, fecha)`, que aunque no se invoca en este flujo desde el backend, se llama desde la interfaz durante esta actividad y está directamente relacionada con el test. En ella, básicamente se crea o abre el archivo csv para escribir en él los datos del desempeño del alumno en el cuestionario.

Generar Problemas

Prosiguiendo por los problemas, en esta sección se contempla la generación de enunciados prácticos, diferenciando la implementación respecto al tipo de problema seleccionado por el usuario. El procedimiento presenta diferencias importantes respecto a la generación de respuestas para consultas, tanto por el contenido como por la estructura del prompt utilizado.

1. Preparar el historial. A diferencia del caso de las consultas, en este escenario no existe una pregunta explícita del usuario. Por ello, el historial se construye únicamente a partir del tipo de problema y la temática o el supuesto.
2. En función de la alternativa designada por el usuario, se determinan dos parámetros clave: la `seccion` (Enunciado Modelo ER o SQL) y la `problematica` o supuesto específico sobre el cual generar el problema. Si se selecciona el enunciado, la problemática equivaldrá a Modelo ER. Además, también se incluiría en `entorno` o temática.
3. Fase *Retrieval - Recuperación*: Se utiliza la función `buscar_por_objetivo_subobjetivo(seccion, problematica)`, la cual, a diferencia de otros casos, no requiere una pregunta como entrada. Su funcionamiento se resumiría en:

- Carga el *vectorstore*.
- Recuperar los documentos en función de los metadatos (sección, problemática). En este caso, no se hace uso del método `similarity_search`, ya que no hay una pregunta como entrada. Por tanto, la implementación para recuperar los ejemplos de los enunciados, o cada supuesto es distinta. En primer lugar, se invoca a `resultados = vectorstore.get(include=['metadatas', 'documents'])`, que recupera todos los fragmentos. El filtro, dada a la ausencia de una pregunta predefinida, requiere de un enfoque manual 7.8.

```
#Al no haber pregunta, el filtrado ha de ser manual
documentos_filtrados = []
metadatas_filtrados = []

# Por cada par documets (fragmento) - metadato, se itera
for doc, meta in zip(resultados_raw["documents"],
                      resultados["metadatas"]):
    # Se almacena aquellos fragmentos donde el parametro coincida
    if ((not seccion or meta.get("seccion") == "Problemas")
        and
        (not problematica or meta.get("problematica")
         == subobjetivo)):
        documentos_filtrados.append(doc)
```

```
metadatas_filtrados.append(meta)
```

Fragmento de Código 7.8: Filtro manual

- Finalmente, se retornaría las variables que contienen los datos.
4. Preparación del prompt (*Augmented*) y llamada al modelo (*Generation*): En este punto, se aplica una lógica distinta dependiendo del tipo de problema. Sin embargo, señalar que para generalizar y reutilizar código y la estructura de llamada al modelo, ver 7.9, se mantienen constantes las variables del prompt (`context`, `query`, `session` y `entorno`). Aunque las variables tengan el nombre igual, su contenido varía según el caso: por ejemplo, `query` puede ser una descripción o una lista de funcionalidades, según el tipo de problema. También se adapta el PromptTemplate mediante una variable `prompt_en_problemas`, que se define dinámicamente en función del supuesto. Esto permite usar una única `chain` sin duplicar código. Dicho esto, se prosigue con la lógica dependiendo del tipo de problema.

```
# Definicion Prompt
prompt = PromptTemplate(
    template=prompt_consulta,
    input_variables=["context", "session", "query", "entorno"]
)

#Construcion de la cadena
chain = (
    RunnableMap({
        "context": lambda _: context,
        "session": lambda _: session_text,
        "query": lambda _: pregunta,
        "entorno": lambda _: entorno
    })
    | prompt
    | llm
)

# Ejecucion
respuesta = chain.invoke({}))
```

Fragmento de Código 7.9: Prompt - Modelo Problemas

a) Tipo “Enunciado Modelo ER”

- El contexto `context` se forma extrayendo los fragmentos de los ejemplos de los enunciados del pdf por medio de la función `extraer_enunciados_de_fragmentos(fragmentos)`. Este método, toma como entrada los fragmentos recuperados por la función anterior, y únicamente los combina en un bloque de texto, incluyendo separaciones de estilo (limitadores ‘—’) y encabezados del tipo Ejemplo N, para que el modelo entienda claramente los múltiples ejemplos proporcionados.

- La pregunta describiría el objetivo del prompt (`prompt_generar_enunciado_ER`) para la generación del enunciado.
- Finalmente, se introduce una nueva variable denominada `entorno`, que en este caso, ambientaría la temática del enunciado en el prompt.

b) Tipo “SQL”

- Para esta circunstancia, se recurre a la función `extraer_sql_de_fragmentos(fragmentos)`, que devuelve en el contexto `contexto` tanto el enunciado como el modelo lógico del supuesto por un lado. Por el otro, `pregunta` se asignaría a las funcionalidades SQL de ejemplo del supuesto indicado.
- Además, en la variable `entorno` anterior definida, se le atribuiría el nombre del supuesto, para contextualizar mejor el caso del que generar.

5. Guardar el historial con `guardar_historial(supuesto, respuesta.content, ‘problemas’, contextos=context)` y se invoca cuando no ha habido una excepción previa en la llamada a la API.
6. Finalmente, se retornaría la respuesta del LLM con el ejercicio generado con `respuesta.content`.

7.2.5. Prompts prompts.py

Este archivo contiene los distintos plantillas de los prompts para que el asistente pueda generar las respuestas y ejercicios. Cada uno está diseñado para una finalidad específica en función del tipo de actividad. Antes de presentar cada uno, destacar que los campos de cada prompt (indicados entre { }), se completan dinámicamente según la construcción del `chain` y, por tanto, de la situación y necesidades del estudiante. A continuación, se exponen los prompts empleados como entrada del LLM, los cuales siguen las técnicas de ingeniería de prompting ya vistas para así aumentar la calidad de las respuestas a recibir. Se destacan los siguientes: `prompt_consulta` en la Figura 7.2, `prompt_test` en 7.3, `prompt_genenar_enunciado_ER` en 7.4 y `prompt_funcionalidades_sql` en 7.5.

Destacar el prompt 7.6 que permite añadir una funcionalidad extra, donde se podría incluir también la situación inversa: a partir de cierta consulta en SQL, el estudiante podría interpretar qué devuelve o qué realiza esa consulta.

Cada prompt está diseñado para ser flexible, reutilizable y adaptado al historial de interacción del estudiante, mejorando así la personalización de la experiencia.

Capítulo 7. Implementación

`prompt_consulta`

```
prompt_consulta = """
Eres un asistente de aprendizaje que actúa como un profesor experto en Sistemas de Bases de Datos.
Tu tarea es ayudar a un estudiante a responder su consulta de forma clara, precisa y didáctica.

A continuación, te proporcionaré un contexto que debes usar como base para generar la respuesta a la pregunta del estudiante.

-----
Contexto (usa esta información para fundamentar tu respuesta):
{context}

-----
Histórico de la sesión (interacciones previas con el estudiante, para mantener coherencia):
{session}

-----
Pregunta del estudiante:
{query}

-----
Instrucciones para responder:

1. Si el contexto es suficiente o relevante para responder la pregunta,
genera una respuesta clara, precisa, específica y basada en la información proporcionada en el contexto, siempre que esta sea clara y
suficiente, aunque no esté organizada explícitamente.
No muestres ningún aviso en este caso.

2. Si el contexto es insuficiente, irrelevante o no contiene la información necesaria para responder efectivamente,
empieza tu respuesta con el siguiente aviso:
"AVISO: contexto insuficiente para generar la respuesta únicamente a partir de los apuntes proporcionados."

Después del aviso, sugiere que el estudiante revise el material, reformule o especifique mejor su pregunta.
Solo en este caso, puedes intentar responder basándote en tus conocimientos generales,
pero siempre dejando claro que la respuesta puede no estar completamente fundamentada en el contexto.

Recuerda que tu objetivo principal es ayudar al estudiante utilizando el contexto proporcionado como base para la respuesta.

Comienza generando directamente la respuesta al estudiante.
"""

"""

```

Figura 7.2: Prompt Consulta

prompt_test

```
prompt_test = """
Eres un asistente de aprendizaje que actúa como un profesor experto en Sistemas de Bases de Datos.
Tu tarea es generar preguntas tipo test para que un estudiante practique los conceptos clave del curso.

Instrucciones
- Usa el contexto proporcionado como base para generar las preguntas.
- Si el contexto es insuficiente, completa con tus conocimientos generales del dominio.
- Genera exactamente {num_preguntas} preguntas tipo test centradas en el tema "{objetivo}", con un nivel de dificultad "{dificultad}".
- Asegúrate de que las preguntas estén enfocadas en el siguiente subobjetivo:
| "{subobjetivo}"
- Las preguntas deben ayudar activamente al estudiante a comprender y trabajar tanto el objetivo como el subobjetivo proporcionados.

Requisitos de formato (imprescindible)
- Cada pregunta debe comenzar con "Pregunta N:", seguida del enunciado claro.
- Proporciona exactamente cuatro opciones: A), B), C), D)
- Indica al final la opción correcta con el formato: CORRECTA: B (reemplaza "B" por la letra correcta), y que no sean siempre la misma letra

-----
Contexto del tema
{context}

-----
Descripción del objetivo
{objetivoDescripcion}

-----
Descripción del subobjetivo
{subobjetivoDescripcion}

-----
Historial de interacción con el estudiante
Ten en cuenta estas interacciones previas para evitar repeticiones o ajustar el estilo:
{session}

-----
Aspectos adicionales introducidos que el estudiante quiere trabajar:
{query}

-----
Comienza generando directamente las preguntas siguiendo todas las instrucciones anteriores.
"""

"""

```

Figura 7.3: Prompt Test

Capítulo 7. Implementación

prompt_genenar_enunciado_ER

```
prompt_genenar_enunciado_ER = """
Eres un asistente de aprendizaje que actúa como un profesor experto en bases de datos.
Tu tarea es generar un nuevo enunciado de un problema de modelado Entidad-Relación (ER) para que el estudiante practique el diseño de modelos Entidad-Relación, orientado a estudiantes universitarios de Sistemas de Bases de Datos.

Toma como inspiración los ejemplos que te proporciono a continuación. Estos ejemplos sirven como contexto, estructura y estilo, pero NO deben copiarse literalmente.

El enunciado debe ser completamente original, coherente y con un contexto realista y aplicable.

Dominio o entorno temático:
El nuevo enunciado debe estar ambientado en el la temática "{entorno}".
Este contexto debe reflejarse en las entidades, relaciones y desarrollo del enunciado generado, el cual ha de ser creativo y contener elementos de la temática indicada.

Asegúrate de que el nuevo enunciado incluya, siempre que sea posible, al menos los siguientes elementos, pero no de manera explícita:
- Al menos una **entidad débil** y su **entidad fuerte asociada**
- Relaciones con o sin jerarquías **IS-A**
- Atributos **multivaluados** y **derivados**
- Cardinalidades expresadas usando la notación de **Chen**: `(0:1)`, `(0:N)`, `(1:N)`, `(N:N)`
- Relaciones de distintas entidades
- **Restricciones** relevantes (por ejemplo: exclusividad, participación total) y **supuestos realistas**
- Mínimo 6 entidades distintas
- La longitud del enunciado debe de ser de una hoja aproximadamente (25 líneas)

Importante:
- No reutilices frases completas, nombres exactos ni esquemas de los ejemplos.
- El texto debe estar redactado con claridad y tono formal, como si fuera parte de un examen o práctica académica.
- Incluye para cada enunciado algo creativo y distintivo, no siempre la misma estructura o patrón.
- El enunciado NO tiene que ser obvio NI dar pistas. Es decir, NO pueden aparecer explícitamente los elementos que incluyes.
Ejemplo: NO ponga directamente que existe una IS-A o que existe una entidad llamada X. Si no que, gracias al enunciado, el alumno tendrá que averiguarlo.
Fíjate en los ejemplos de cómo hacerlo.
- El enunciado no ha de ser simple, si no que sea un poco tedioso, de tal manera que tengas que darle una segunda vuelta.

-----
Ejemplos para inspirarte:
{context}

-----
Para generar el ejercicio ten en cuenta:
{query}

-----
Ten en cuenta estas interacciones previas y enunciados que ya has generado para evitar repeticiones o ajustar el estilo a uno más novedoso:
{session}

-----
Al final, devuelve únicamente el nuevo enunciado generado, pero ten en cuenta la temática ambientada.
Por ejemplo, si la temática es Salud, una entidad que sea Hospital, y NO un perfil de una aplicación. Pon especial cuidado en ello.
***
```

Figura 7.4: Prompt Problemas: Enunciados Modelos ER

`prompt_funcionalidades_sql`

```

prompt_funcionalidades_sql = """
Eres un asistente de aprendizaje que actúa como un profesor experto en Bases de Datos.
Tu tarea es ayudar a los estudiantes a practicar la escritura y comprensión de sentencias SQL a partir de un caso práctico.

A continuación se proporciona un texto que incluye tanto el supuesto o enunciado del problema como el modelo lógico (tablas, atributos y relaciones entre ellas). También se incluyen algunos **ejemplos de funcionalidades ya redactadas** que pueden servirte de inspiración.

-----
Texto del enunciado y modelo lógico, que será la base para la Base de Datos en MySQL:
{context}

-----
Ejemplos de funcionalidades son:
{query}

-----
Tu tarea consiste en:
1. Redactar 5 funcionalidades SQL nuevas que podrían implementarse en ese sistema.
Estas funcionalidades deben ser completas, estar expresadas en lenguaje natural, como indicaciones claras que podría recibir un estudiante. Ejemplos:
- "Obtener los usuarios que han reservado más de una vez en el último mes."
- "Listar los productos cuyo precio es mayor al promedio de su categoría."
- "Mostrar para cada curso el número total de alumnos inscritos, ordenados de mayor a menor."
| Fíjate en las funcionalidades de ejemplo que te he proporcionado antes.

-----
Indicaciones clave:
- Asegurate de que las funcionalidades sean variadas y aprovechen relaciones entre tablas.
- Utiliza operaciones SQL relevantes: 'JOIN', 'WHERE', 'GROUP BY', 'ORDER BY', 'DISTINCT', funciones agregadas ('COUNT', 'SUM', 'AVG', etc.).
- La redacción debe ser formal y clara, como si se tratara de ejercicios de una práctica o examen.
- Evita copiar literalmente ningún fragmento del modelo lógico, pero utiliza sus nombres y estructura como base.
- No repitas funcionalidades parecidas.
- La consulta inversa debe ser coherente y realista.

-----
Ten en cuenta estas interacciones previas para evitar repeticiones o ajustar el estilo (si las hay):
{session}

-----
Devuelve únicamente las 5 funcionalidades propuestas para el supuesto {entorno}
"""

```

Figura 7.5: Prompt Problemas: Funcionalidades SQL

Capítulo 7. Implementación

```
prompt_funcionalidades_sql = """
Eres un asistente de aprendizaje que actúa como un profesor experto en Bases de Datos.
Tu tarea es ayudar a los estudiantes a practicar la escritura y comprensión de sentencias SQL a partir de un caso práctico.

A continuación se proporciona un texto que incluye tanto el supuesto o enunciado del problema como el modelo lógico (tablas, atributos y relaciones entre ellas).
También se incluyen algunos **ejemplos de funcionalidades ya redactadas** que pueden servirte de inspiración.

-----
Texto del enunciado y modelo lógico, que será la base para la Base de Datos en MySQL:
{context}

-----
Ejemplos de funcionalidades son:
{query}

-----
Tu tarea consiste en:
1. Redactar 5 funcionalidades SQL nuevas que podrían implementarse en ese sistema.
Estas funcionalidades deben ser completas, estar expresadas en lenguaje natural, como indicaciones claras que podría recibir un estudiante. Ejemplos:
- "Obtener los usuarios que han reservado más de una vez en el Último mes."
- "Listar los productos cuyo precio es mayor al promedio de su categoría."
- "Mostrar para cada curso el número total de alumnos inscritos, ordenados de mayor a menor."
- Fijate en las funcionalidades de ejemplo que te he proporcionado antes.
2. Formular una pregunta inversa: proporciona una sentencia SQL inspirada en el caso, y plantea una pregunta al estudiante que lo lleve a interpretar qué devuelve o qué hace esa consulta.

-----
Indicaciones clave:
- Asegúrate de que las funcionalidades sean variadas y aprovechen relaciones entre tablas.
- Utiliza operaciones SQL relevantes: `JOIN`, `WHERE`, `GROUP BY`, `ORDER BY`, `DISTINCT`, funciones agregadas (`COUNT`, `SUM`, `AVG`, etc.).
- La redacción debe ser formal y clara, como si se tratara de ejercicios de una práctica o examen.
- Evita copiar literalmente ningún fragmento del modelo lógico, pero utiliza sus nombres y estructura como base.
- No repitas funcionalidades parecidas.
- La consulta inversa debe ser coherente y realista.

-----
Ten en cuenta estas interacciones previas para evitar repeticiones o ajustar el estilo (si las hay):
{session}

-----
Devuelve únicamente:
- Las 5 funcionalidades propuestas para el supuesto {entorno}
- La pregunta inversa con su sentencia SQL
"""
"""


```

Figura 7.6: Prompt Problemas: Consulta SQL

Soluciones a los problemas

En esta subsección, se muestran como variarían los prompts iniciales de los problemas, en la situación de que sea necesario querer obtener también la solución para cada tipo de problema. Ejemplo de ello son los prompts 7.7 y 7.8.

```
Al final, devuelve primero el enunciado generado, redactado de forma clara y formal como se indicó anteriormente.

Después del enunciado, incluye una sección titulada SOLUCIÓN PROPUESTA que describa de forma estructurada los elementos del modelo Entidad-Relación:

- Entidades: nombra todas las entidades principales identificadas.
- Atributos: indica atributos clave, multivaluados y derivados para cada entidad.
- Relaciones: describe las relaciones entre entidades, incluyendo nombres y cardinalidades (usando notación de Chen como '(0:1)', '(1:N)', etc.).
- Entidad débil: especifica cuál es la entidad débil y su entidad fuerte asociada.
- Jerarquías IS-A: si existen, indica las jerarquías o especializaciones.
- Restricciones: señala restricciones como participación total, exclusividad, etc.
- Supuestos: añade si es necesario algún supuesto que ayude a entender o justificar el modelo.

El objetivo es que otro profesor o estudiante pueda entender claramente si el modelo planteado es coherente, sin necesidad de interpretar el enunciado por sí mismo.

Formato de salida esperado:

*** ENUNCIADO ***
[Texto del enunciado]

*** SOLUCIÓN PROPUESTA ***
Entidades: ...
Atributos: ...
Relaciones: ...
Entidad débil: ...
Jerarquías IS-A: ...
Restricciones: ...
Supuestos: ...
***
```

Figura 7.7: Prompt Problemas: Soluciones Modelo ER

Capítulo 7. Implementación

```
1. Redactar 5 funcionalidades SQL nuevas que podrían implementarse en ese sistema.  
Cada funcionalidad debe presentarse en el siguiente formato:  
  
**Funcionalidad X:**  
_Descripción en lenguaje natural de la funcionalidad_  
**Sentencia SQL:**  
```sql  
[aquí la sentencia SQL correspondiente]

2. Formular una pregunta inversa, que consiste en:
-Proporcionar una sentencia SQL escrita por ti.
- Plantear una pregunta al estudiante para que interprete qué devuelve esa consulta.
Formato:

Pregunta inversa:
[Pregunta en lenguaje natural]

Sentencia SQL:
[consulta SQL correspondiente]

Solución de lo que devolvería la consulta en lenguaje natural

Indicaciones clave:
- Asegúrate de que las funcionalidades sean variadas y aprovechen relaciones entre tablas.
- Utiliza operaciones SQL relevantes: 'JOIN', 'WHERE', 'GROUP BY', 'ORDER BY', 'DISTINCT', funciones agregadas ('COUNT', 'SUM', 'AVG', etc.).
- La redacción debe ser formal y clara, como si se tratara de ejercicios de una práctica o examen.
- Evita copiar literalmente ningún fragmento del modelo lógico, pero utiliza sus nombres y estructura como base.
- No repitas funcionalidades parecidas.
- La consulta inversa debe ser coherente y realista.

Ten en cuenta estas interacciones previas para evitar repeticiones o ajustar el estilo (si las hay):
{session}

Devuelve únicamente:
1º, las funcionalidades, con el siguiente formato, adaptado al supuesto {entorno}:

*** FUNCIONALIDADES PROPUESTAS PARA EL SUPUESTO {entorno} ***

Funcionalidad 1:
[Descripción]

Sentencia SQL:
[Consulta SQL]
//////////

Funcionalidad 2:
[Descripción]

Sentencia SQL:
[Consulta SQL]
//////////
(.... y así hasta la 5:)

Funcionalidad 5:
[Descripción]

Sentencia SQL:
[Consulta SQL]

2º La pregunta inversa y su solución
"""
```

Figura 7.8: Prompt Problemas: Soluciones Funcionalidades SQL

### 7.2.6. Interfaz main.py

Esta archivo es el punto de entrada para acceder al asistente de aprendizaje. Define la estructura general de la interfaz mediante un menú lateral y la cabecera. Además, renderiza la vista en función de la opción seleccionada en el menú, por lo que permite la navegación entre las distintas actividades que ofrece el asistente.

Dependiendo de la opción seleccionada por el usuario en el menú, se invoca dinámicamente la función `mostrar()` correspondiente a cada página, la cual se cargaría dinámicamente. Alternativamente, se podría utilizar `st.switch_page`, aunque en este caso no se emplea.

En resumen, centraliza la navegación y habilita la carga de componentes definidos en otros archivos (los contenidos en `paginas`).

### 7.2.7. Directorio paginas

Este directorio contiene principalmente una serie de archivos que contendrán las interfaces para cada tipo de actividad. Dependiendo de la elección del usuario, se carga dinámicamente el contenido de la página asociada, sin necesidad de recargar toda la aplicación.

Cabe destacar que todos los archivos de `interfaz.py` dentro de este directorio comparten el mismo esquema de funcionamiento, ya que se utiliza la biblioteca `Streamlit`. Esto permite una carga dinámica y reactiva del contenido, actualizándose automáticamente en función de las acciones del usuario.

#### Página Consultas consultas.py

Este fichero alberga el flujo de las consultas. En primer lugar, el usuario selecciona el objetivo específico sobre el cual desea formular su duda. A continuación, introduciría textualmente su consulta, inquietud o necesidad. Uno de los elementos clave es el uso de un `selectbox`, donde se muestran las opciones derivadas de los ítems principales definidos en la variable `OBJETIVOS`:

```
objetivo = st.selectbox("Elige un objetivo principal",
options=list(OBJETIVOS.keys()), index=0)
```

Fragmento de Código 7.10: *Selectbox*

El asistente generaría la respuesta una vez que el usuario pulsa el botón “Generar Respuesta” (`if st.button("Generar respuesta")`). Al hacer click, el sistema comprueba que el texto no esté vacío e invoca a la función `resultado = responder_consulta(pregunta, objetivo, st.session_state.historial)`, que devuelve la respuesta generada en función del objetivo y el historial de la sesión (si el `status` de la API es correcto). Este historial se construye a partir de la variable de sesión `st.session_state.historial`, que permite mantener el contexto dentro de la sesión de Streamlit añadiendo la interacción previa. En este caso, la pregunta y la respuesta, apreciado en el código 7.11.

```
st.session_state.historial.append({
 "pregunta": pregunta,
 "respuesta": resultado["respuesta"]
})
```

Fragmento de Código 7.11: Variable Sesión

Finalmente, la respuesta se muestra en pantalla junto con el historial de la conversación, si existe, simulando una interacción tipo chat (`if st.session_state.historial:`).

### Página Test test.py

Este archivo haría referencia a las cuestiones de opción múltiple o tipo test. Inicialmente, el usuario configura una serie de valores clave para la generación del test:

1. Número de preguntas, mediante un `slider`
2. Especificar el objetivo, de manera similar al caso anterior
3. Seleccionar uno o varios criterios de evaluación a través de un `multiselect`
4. Definir el nivel de dificultad con otro `radio`

Resaltar que tanto para la historia como para los criterios se actualiza la información mediante una variable global. Además, el sistema incorpora controles robustos para manejar errores comunes, como la omisión de criterios o apartados sin responder, e incluso mecanismos para exhibir información adicional de las descripciones a modo de ayuda. Cada selección se almacena en una variable de sesión (`st.session_state`), lo que permitirá mantener el estado entre interacciones.

En este punto, el usuario, pulsa el botón Empezar Test, y se generan las preguntas mediante la función `generar_test(pregunta, objetivo, historia, num_preguntas, st.session_state.historial_test, dificultad)`, donde se le pasan los parámetros de la configuración (siendo la pregunta `pregunta = ', '.join(st.session_state.criterios_seleccionados)`, y solo si el `status` de la API es correcto).

Estas cuestiones se almacenan en `st.session_state.test_preguntas`, y se inicializan marcadoreyes como el tiempo de inicio del test y un flag que indica que el cuestionario se ha iniciado.

Si el test está iniciado (`st.session_state.quiz_iniciado`), se carga dinámicamente el contenido y se presenta un formulario interactivo que recorre cada pregunta, mostrando sus opciones de respuesta con botones tipo `radio` para evaluar el conocimiento del usuario. Por defecto, se incluye una opción vacía (Selecciona una opción) para forzar al usuario a tomar una decisión explícita.

Una vez que el usuario responde todas las preguntas y pulsa “Finalizar Test”, el sistema valida que todas hayan sido contestadas. En caso contrario, se notifica una advertencia con el número de preguntas sin responder.

Tras la validación, se calcula el número de aciertos y fallos, así como el tiempo total invertido y se muestra un resumen de dichos resultados: opción escogida, si es correcta, tiempo transcurrido y puntuación global de la sesión.

Estos campos se acumulan en la puntuación global `st.session_state.puntuacion_total`, que junto al resto de variables de sesión, se pasan como parámetros a la función `guardar_resultados_test_csv()` para registrar los resultados en el fichero CSV 7.12.

```
fecha = date.today()
guardar_resultados_test_csv(
 st.session_state.objetivo,
 st.session_state.historia,
```

```
st.session_state.puntuacion_total ,
aciertos ,
fallos ,
tiempo_total ,
fecha
)
```

Fragmento de Código 7.12: CSV

Finalmente, destacar `st.session_state.historial_test`, la variable de sesión donde se recabarán las preguntas, respuestas y opciones correctas del test. Así como el botón Volver, que permite reiniciar la actividad del test y borrar el estado de las sesiones (a excepción de la que contiene el historial de preguntas, que evitaría la repetición).

#### Página Problemas problemas.py

Este fichero contiene el flujo de los problemas. En primer lugar, el usuario indica el tipo de problema sobre el cual desea practicar con `selectbox`.

Si resulta que el estudiante quiere practicar modelos de entidad relación, es decir, el tipo de problema es Modelos ER, se incluye con `radio` ejemplos de distintas temáticas en el que basar dicho ejercicio, aunque siempre puede incluir otro tipo escribiéndole textualmente y de manera manual.

En cambio, en el supuesto de que desee profundizar en el aprendizaje de funcionalidades de SQL, es necesario que mediante otro `radio` seleccione el supuesto de ejemplos preexistentes sobre el que el asistente generará las nuevas funcionalidades.

Una vez que el usuario pulsa el botón “Generar Problema” (`if st.button("Generar Problema")`), se invoca a la función

```
generar_problemas(tipo, supuesto, entorno, session_historical_problems),
```

que devuelve la respuesta generada en función del problema e historial de sesión (si el `status` de la API es correcto). Este historial se construye a partir de la variable de sesión `st.session_state.historial` 7.13.

```
st.session_state.historial_problemas.append({
 "tipo": tipo ,
 "supuesto": supuesto ,
 "respuesta": resultado["respuesta"]
})
```

Fragmento de Código 7.13: Variable Sesión

Finalmente, la respuesta se muestra en pantalla junto con el historial de la conversación, si existe, simulando una interacción tipo chat (`if st.session_state.historial:`).

#### Página Progreso verProgreso.py

Por último, se dispone de este archivo para estudiar, visualizar y analizar el rendimiento del alumno a lo largo de los diversos test que este ha efectuado. Mediante la carga de los datos predispuestos en el csv, se posibilita mostrar el comportamiento del usuario de manera perceptible, lo que le ayudará a entender aquellos objetivos sobre los que poder trabajar a mayor profundidad y por tanto, le servirá para conocer su estado de aprendizaje.

Para ello, se usará `plotly`, una librería de python que permite la modelización visual de un *dataframe*, facilitando por tanto la creación de los distintos gráficos, dashboards y tablas que se incluye en la vista. Destacar `pandas` para el tratamiento de los datos. A continuación, se muestra la implementación en función de los diagramas definidos en el diseño.

En primer lugar, se cargan los datos como *dataframe* (mediante `pd.read_csv()`). Seguidamente, se inicializan las variable a utilizar. Este paso abarca la creación de la variable `resumen` que almacena para cada objetivo, la suma de aciertos, fallos y tiempo total. Es decir, se agrupan los datos por la columna objetivo 7.14:

```
resumen = df.groupby("objetivo",
 as_index=False).agg({
 "aciertos": "sum",
 "fallos": "sum",
 "tiempo_total": "sum"
 })
resumen["total"] = resumen["aciertos"] +
resumen["fallos"]
```

Fragmento de Código 7.14: Resumen por Objetivo

Prosiguiendo, se procedería a la construcción de los distintos elementos visuales descritos en el capítulo anterior:

- KPIs: para cada columna del *dataframe*, calcular la suma de los aciertos, fallos, preguntas y tiempo total (`df['columna'].sum()`, además de la media del tiempo (`df['columna'].median()`). A la hora de mostrar estos KPI por la interfaz, se muestran en una fila de cinco columnas, utilizando `st.columns()` y el método `.metric()` para visualizar los valores de forma ordenada, legible y clara 7.15:

```
k1, k2, k3, k4, k5 = st.columns(5)
k1.metric("Total_Preguntas", total_preguntas)
k2.metric("Aciertos_Totales", total_aciertos)
k3.metric("Fallos_Totales", total_fallos)
k4.metric("Tiempo_Total_(s)", tiempo_total)
k5.metric("Tiempo_Medio_por_Pregunta_(s)", tiempo_total_avg)
```

Fragmento de Código 7.15: KPIs

- Diagramas: respecto a las visualizaciones de las gráficas, se han implementado los distintos diagramas detallados en el capítulo 6. Los gráficos de barras apiladas, que muestran aciertos y fallos por objetivo, se crean con `plotly.graph_objects`. No obstante, para el resto se hará uso de `plotly.express`: el total de preguntas por objetivo se representa mediante un gráfico de barras simple con `px.bar`, así como para el tiempo total invertido por objetivo y además, se incorpora un gráfico circular con `px.pie`. Todos estos gráficos se integran en la interfaz usando `st.plotly_chart()` con el parámetro `use_container_width=True` para adaptar el tamaño del diagrama al contenedor creado.
- Tabla: finalmente, se ofrece al usuario una tabla interactiva con el historial completo de los tests realizados, implementado con `st.dataframe(df)`, lo que facilita la revisión de sus respuestas, tiempo empleado, puntuación y desempeño de forma detallada.

En conjunto, este archivo transforma los datos obtenidos en métricas y visualizaciones comprensibles y efectivas, favoreciendo una autoevaluación clara y orientada del progreso académico del alumno.

#### **Página Definiciones definiciones.py**

Finalmente, se presenta este archivo que define el diccionario de objetivos, historias y criterios, incluyendo sus relaciones y la descripción de cada uno. Con su implementación, se permite centralizar estas especificaciones, definiciones y conceptos y facilitar su uso y mantenimiento en otras partes del sistema.

### 7.3. Dificultades

Durante esta fase de preparación e integración de herramientas, se identificaron diversas dificultades y retos técnicos. En primer lugar, la gestión de dependencias y versiones incompatibles, especialmente entre bibliotecas que requieren versiones específicas de Python o de otros paquetes auxiliares. La instalación de versiones de librerías compatibles supuso más tiempo del esperado, pero mediante una labor investigadora, se solventó el problema.

Por otro lado, y en cuanto a la base vectorial, el uso de ChromaDB presentó una curva de aprendizaje considerable. Aunque se exploraron alternativas como MySQL o MyScale y se intentó su integración mediante contenedores Docker. Sin embargo, fue sin éxito, ya que aumentaba exponencialmente la complejidad del proyecto, por lo que se decidió profundizar más en su documentación y cómo utilizarlo junto a Langchain.

Respecto a este *framework*, también se identificaron limitaciones relacionadas con su empleo. La falta de experiencia previa requirió una investigación intensiva y continua, especialmente para comprender cómo construir y encadenar correctamente los distintos componentes. Lo mismo ocurrió con Streamlit para el desarrollo de la interfaz de usuario, cuyo aprendizaje fue realizado desde cero y de manera paralela al desarrollo del proyecto.

Uno de los principales retos estuvo vinculado a la decisión sobre qué modelo LLM implementaría el asistente de aprendizaje basado en RAG. Desde el inicio del proyecto se contaba con la iniciativa de contemplar algún modelo de OpenAI, ya que el departamento disponía de los recursos necesarios para cubrir ese gasto. No obstante, en las primeras etapas aún no se contaba con la clave de acceso a la API, y era necesario comenzar el desarrollo sin más dilación. Por este motivo, se dedicó tiempo a investigar dentro de la naturaleza del proyecto diferentes modelos que resultaran los más óptimos posibles. Finalmente, debido a su capacidad generativa de texto con calidad, se recurrió temporalmente al uso de Ollama, una alternativa gratuita que permitía ejecutar modelos LLM directamente en el equipo personal.

Esta solución inicial permitió avanzar en el diseño y pruebas del sistema, pero también presentó limitaciones importantes. Ollama tenía problemas de compatibilidad con otras herramientas del ecosistema, y no ofrecía soporte integrado para embeddings, lo que obligaba a realizar implementaciones manuales adicionales. Además, al ejecutarse localmente, el rendimiento se vio comprometido por la capacidad limitada del *hardware*, especialmente de la GPU, provocando tiempos de respuesta elevados.

Una vez obtenida la clave de acceso a OpenAI, se procedió a la migración del sistema, lo que implicó adaptar parte de la arquitectura y los *chains*, debido a diferencias en la forma de interacción con los modelos y construcción del flujo. Con esta modificación se pudo observar a su vez una mejora notable en estabilidad, velocidad y calidad de respuesta, consolidando la decisión inicial de utilizar un servicio externo más robusto. No obstante, también fue necesario realizar una profunda labor de investigación para identificar y analizar aquellos modelos que ofrecieran las mejores respuestas, equilibrando calidad y tiempo de contestación sin exceder el presupuesto disponible.

A pesar de sus limitaciones, destacar que el uso provisional de Ollama fue útil para conocer y explorar los modelos alternativos alojados en HuggingFace para la generación de *embeddings*, y para comprender mejor el funcionamiento interno de los LLMs y su integración con *pipelines RAG*. Además, también ayudó a tomar decisiones más informadas sobre las herramientas y servicios a utilizar, así como en una ampliación del conocimiento sobre la variedad de opciones disponibles y posibles.

Un aspecto que también supuso modificaciones a posteriori fue el relativo a la fragmentación de documentos para su indexación. Al inicio del proyecto, debido al desconocimiento sobre cuál era la mejor estrategia, se optó por utilizar técnicas de `RecursiveCharacterTextSplitter` para los *chunks*. Sin embargo, este enfoque resultó poco eficaz, ya que en el material proporcionado, mayoritariamente cada diapositiva, estaban organizados de forma muy clara por objetivos, historias y criterios. Por tanto, tras realizar un profundo análisis de rendimiento de las diversas técnicas existentes, se optó por el actual, el cual aunque acabó siendo la opción más simple, era el que mejores resultados de eficiencia obtuvo. Asimismo, como consecuencia final, destacar que implicó rehacer la lógica de creación de la base de datos vectorial.

Respecto a la fase de recuperación, o búsqueda de los fragmentos más relevantes dentro de la base de datos vectorial, un importante desafío fue la gestión y filtrado de *embeddings*. Tal y como concluyen diversos estudios [189, 254, 183] una búsqueda más dirigida o acotada a un subconjunto mejora significativamente la calidad de los documentos recuperados. Esta idea llevó a la urgencia de plantear un diseño eficaz en cuanto a la lógica de los metadatos. Para lograrlo, fue necesario dedicar cierto tiempo en su diseño, optimización y estructuración, para así poder recuperar segmentos más acotados, relevantes y precisos. Por otro lado, a pesar de disponer fragmentos más determinantes a raíz de lo recientemente comentando, en ocasiones la información que se adjuntaba como contexto en el prompt (los tres resultados con más similitud) no eran los más coherentes con la intención de la pregunta. Es otras palabras, la similitud semántica difiere del sentido o coherencia contextual, es decir, una no garantiza la otra. Tras apreciar esta discrepancia, se procedió a la integración de la lógica del *reranking*, cuya implementación permitió finalmente seleccionar los fragmentos más relevantes no solo desde un punto de vista semántico, sino también en términos de coherencia con la voluntad de la pregunta. Por último, cabe destacar que tanto en esta fase como en la anterior, el análisis no fue trivial. Fue necesario un proceso iterativo de observación, exploración, ajustes y pruebas sucesivas para alcanzar un equilibrio adecuado entre precisión, calidad de la información y pertinencia del contexto proporcionado.

Por otro lado, una consideración interesante para el desarrollo del asistente es la validación de las entradas, pregunta y filtros aplicados a la búsqueda. Aunque ChromaDB no es vulnerable a inyecciones SQL, puede existir el riesgo de que el modelo LLM se haya entrenado con prompts maliciosos que recuperen o borren información durante la transformación de los fragmentos. Asimismo, destacar que en el caso de que la aplicación de los múltiples filtros devuelva un conjunto de fragmentos vacíos, se ha decidido realizar la comparación de similitud semántica para toda la colección.

También se presentaron dificultades en el diseño de prompts. Las respuestas generadas por el modelo variaban significativamente con pequeños cambios en la redacción, lo cual obligaba a refinar e iterar sobre la formulación del prompt para obtener resultados coherentes y útiles. Por ejemplo, cambiar la frase “responde exactamente con el contexto” por “responde en base al contexto” alteraba radicalmente la calidad y texto de la respuesta. A esto se sumaban los problemas con los formatos devueltos por el modelo, que en ocasiones no se trataban de un modo asequible. No obstante, se solucionó redactando mejor cada entrada, estructurándola con instrucciones precisas de formato y texto, y procesando la respuesta obtenida previamente antes de adjuntarla a la interfaz.

En relación al *pipeline* completo de RAG se intentó inicialmente emplear `RetrievalQA` por su abundante documentación y sencillez. No obstante, este es adecuado cuando no se necesita personalizar demasiado la lógica interna. Sin embargo, en el ámbito de este proyecto que se requiere de una mayor personalización del flujo de entrada y salida de datos, resultó ser un

impedimento. Por tanto, el problema se solventó con `chain`, que permitió una implementación más flexible y mayor control sobre las variables e instrucciones que se pasan al modelo.

En cuanto a la arquitectura del código, se detectaron problemas derivados de decisiones iniciales. Fue necesario reestructurar gran parte del sistema, ya que, en una primera versión, se evidenciaba la falta de una buena práctica de programación o sólo se contemplaba una única colección de documentos. Posteriormente, se amplió el enfoque para trabajar con múltiples colecciones, lo que también implicó ajustar los filtros de búsqueda y modificar los prompts utilizados. También se rediseñó el sistema de tests: en un principio, el usuario proporcionaba libremente lo que quería evaluar, pero más adelante se introdujeron criterios estructurados, lo que supuso una reorganización del código y de los flujos de interacción.

A nivel conceptual surgieron dudas durante el desarrollo, ya que no estaba clara la parte de incluir las historias u objetivos como núcleos principales para la personalización en los problemas. Finalmente, se inclinó por prescindir de ello, y optar por ejercicios y supuestos preexistentes vistos en clase que permitieran sustentar el estudio del alumno.

Otro reto fue la interacción con el usuario. Las respuestas generadas eran inicialmente muy independientes entre sí, lo cual generaba una conversación poco fluida. Para solucionarlo, se decidió implementar un historial de conversación que permitiera mantener el contexto de la sesión y mejorar la naturalidad del diálogo y experiencia del usuario.

Por último, la personalización del aprendizaje se presentó como un desafío complejo. El asistente debía ser capaz de adaptar sus respuestas no solo al contenido, sino también al nivel del estudiante, sus objetivos, sus criterios de evaluación y su rendimiento. Para ello, se diseñaron componentes que permitieran definir el nivel de dificultad, los criterios e historias asociados a cada objetivo, y un seguimiento del progreso individual. Esta capacidad de personalización es una de las aportaciones clave del asistente como herramienta de apoyo al estudio individual.

## Personalización de la Enseñanza

La personalización es el principal reto que encontramos en la realización de este proyecto. El desafío predominante al iniciar el proyecto podría haber sido que la personalización del contenido en función del rendimiento del usuario no se lleva a cabo de manera completamente efectiva. Específicamente, el sistema no adaptara adecuadamente el nivel de dificultad de las actividades, no ofreciera contenido adicional basado en las necesidades específicas del alumno, ni proporcionara un apoyo real para el aprendizaje. En otras palabras, no se lograra una verdadera personalización ni se ajusta el contenido al rendimiento real del estudiante, lo que impide que el sistema sea realmente útil para evaluar y mostrar el progreso.

Para abordar estos problemas, se conectó los dashboards con la generación de contenido adaptado, lo que permitiría ajustar el contenido y las actividades según el rendimiento del alumno. Además, debería evaluarse el progreso del estudiante en los test, permitiendo así una visualización clara de su avance. También es importante que se incluyan las consideraciones específicas del usuario a la hora de generar test y problemas. Todo ello para asegurar que las actividades sean relevantes y ajustadas a las necesidades de cada estudiante.

Estos son las maneras en que se ha abordado el desafío de la personalización en la enseñanza a lo largo del proyecto. Sin embargo, comentar que se podrían plantear o considerar más formas de continuar con la personalización de la enseñanza y en este proyecto. No obstante, en ello se profundiza en el capítulo 10, en la sección trabajos a futuro.

## 7.4. Aspectos Clave

Durante el desarrollo del sistema, se han abordado distintos aspectos clave que garantizan la funcionalidad, estabilidad y eficiencia del asistente. A continuación, se detalla el proceso seguido:

- Se ha creado un entorno virtual en Anaconda para gestionar de forma organizada las dependencias del proyecto y evitar conflictos entre versiones de librerías (ver apéndices).
- ChromaDB se inicia antes del comienzo de la aplicación y permanece disponible durante toda la sesión. Esto permite reutilizar la base de datos entre consultas, evitando recargas innecesarias y optimizando el rendimiento.
- LangChain ha sido empleado para construir todo el flujo del pipeline RAG: desde la creación de embeddings, la indexación en ChromaDB, la gestión de la recuperación de fragmentos, la construcción dinámica del prompt y la llamada final al modelo LLM.
- Streamlit se ha utilizado como marco principal para la construcción de la interfaz y la lógica de control de flujo. A través de su sistema de `session-state`, se ha podido gestionar correctamente la actividad del usuario y mantener el estado entre interacciones. Por ello, no se ha empleado un patrón arquitectónico MVC tradicional, ya que Streamlit se basa en una estructura orientada a componentes al igual que RAG.
- La interfaz gráfica se comunica con el backend mediante funciones que encapsulan la lógica necesaria para cada actividad. Esta separación de responsabilidades permite una estructura modular y mantenible.
- Se ha prestado especial atención a la calidad técnica del código: modularidad, funciones bien definidas, reutilizables y con un único propósito, así como una documentación interna clara. La estructura de carpetas y archivos sigue una lógica coherente: `main.py`, `process.py`, `backend.py`, entre otros.
- El sistema se basa en una arquitectura RAG, donde cada tipo de interacción del usuario se canaliza por un pipeline específico. Esto permite adaptar los documentos recuperados, los prompts generados y el modelo de respuesta en función del contexto de la actividad.
- Se dispone de funciones comunes reutilizables que permiten cargar la base de datos vectorial (ChromaDB) y guardar las preguntas del usuario, respuestas del asistente y los datos generados en archivos persistentes. Estas funciones son clave para mantener el flujo del sistema sin duplicar lógica y facilitando la escalabilidad del proyecto.
- Durante el desarrollo se tomaron decisiones estratégicas: se evitó el uso de soluciones en la nube por privacidad (utilizando Ollama y ChromaDB en local), se diseñaron tres *pipelines RAG* independientes en lugar de uno generalista, y se optó por una gestión del progreso del usuario, almacenando resultados en archivos CSV para permitir mejoras futuras.
- Se enfrentaron múltiples retos técnicos: problemas de compatibilidad de versiones, complejidad en la instalación y uso de ChromaDB (inicialmente se intentó usar MyScale vía Docker), desconocimiento inicial de LangChain y Streamlit, limitaciones de Ollama y OpenAI, configuración de *embeddings* manuales cuando no se usaba OpenAI, entre otros y análisis de los resultados para optimizar rendimiento.

- Finalmente, se estableció el uso de OpenAI por razones de calidad, presupuesto, fiabilidad y mejores capacidades en generación de lenguaje, aún considerando el coste de uso como una limitación a tener en cuenta.

# Capítulo 8

## Pruebas

En esta sección se presentan diferentes casos de pruebas aplicados al producto, abarcando tanto de caja negra como de caja blanca. Con ello, se permitirá evaluar del comportamiento del producto, verificar que el producto cumple con todos sus requisitos y detectar defectos o proponer distintas áreas de mejoras.

A continuación, se evaluará si el producto desarrollado cumple con los distintos requisitos especificados y si funciona correctamente en las condiciones previstas. Para ello, primero se definen una serie de casos de prueba, después se ejecutarán dichas pruebas, más adelante se analizarán los resultados y, por último, se propondrán una serie de mejoras.

### 8.1. Casos de Prueba

La cantidad de casos de prueba depende directamente de los requisitos especificados para el asistente, previamente definidos en el capítulo de Análisis 5. No obstante, solo se tratarán en detalle aquellos que simbolizan los tipos de uso más representativos, así como posibles fallos críticos. A continuación, se detallan los casos de prueba definidos, incluyendo su nombre, descripción, entrada, resultados esperados y criterios de aceptación necesarios, diferenciando entre pruebas de caja negra y blanca.

#### Caja Negra

En esta sección se detallan las pruebas de caja negra, las cuales se centran en evaluar la funcionalidad del sistema sin necesidad de conocer su estructura interna, por tanto, examinan la funcionalidad del *software*. En consecuencia, facilitan comprobar si el sistema cumple con los requisitos establecidos a partir de sus entradas y salidas esperadas. La Figura 8.1 recoge estas pruebas.

A un nivel más específico, en las Tablas 8.1, 8.2, 8.3, 8.4 y 8.5 se detallan más a fondo algunos de dichos casos de prueba.

## Capítulo 8. Pruebas

---

ID	Nombre	Descripción	Entrada	Resultado Esperado	Criterio de Aceptación
CP-1	Consulta General	Responder una pregunta abierta sobre la asignatura en base a un objetivo	Objetivo: SGBDs "¿Qué es una base de datos?"	Definición precisa de base de datos	Respuesta coherente, correcta y en línea con el contexto proporcionado
CP-2	Generación Problema	Generación de problema práctico	Tipo Problema Supuesto o Temática	Problema coherente con el tema	Problema coherente, resoluble y relacionado con la
CP-3	Generación Test	Crear preguntas tipo test	Número de Preguntas:X Objetivo Historia Criterios Dificultad Opciones Test	X preguntas de opción múltiple, en base al objetivo, historia, criterios y dificultad seleccionada. Al finalizar el test, un feedback en tiempo real del rendimiento	Preguntas relevantes y bien formuladas
CP-4	Recuperación RAG	Recuperación relevante de información desde ChromaDB, en función de la similitud fragmentos a la pregunta o filtros	Entrada del CP-1	Fragments recuperados por metadato SGBDs y relacionados con "qué es una base de datos"	Fragments recuperados relevantes al contexto de la pregunta, y que ayudan a generar una respuesta precisa
CP-5	Consulta Ambigua	Manejo de preguntas ambiguas, poco claras o no relacionadas con el objetivo	"Expícame la asignatura" ó Con objetivo DML, "Dime los tipos de agregaciones"	Respuesta pidiendo aclaración, aunque intenta responder con el contexto recuperado. Si no es suficiente, también lo comenta	Sistema capaz de identificar consultas ambiguas, intentar proporcionar una respuesta clara y solicitar detalles
CP-6	Entrada Vacía	Manejo de inputs vacíos	"" (cadena vacía)	Mensaje de error solicitando input	Sistema no procesa entradas vacías en las preguntas de la consulta
CP-7	Fallo API LLM	Error en conexión al modelo de lenguaje	Cualquier llamada a la API - Status API	Mensaje: "Servicio no disponible. Inténtalo más tarde" + log error	Manejo de excepciones o fallo de conexión
CP-8	Fallo ChromaDB	Error de acceso o carga	Pregunta - filtros	Mensaje de error de el vectorstore no se ha cargado correctamente	Manejo de excepciones o fallo de conexión
CP-9	Ver Progreso sin datos	Acceder a "Ver Progreso" sin haber realizado ningún test	Usuario sin tests	Mensaje indicando que no hay datos de progreso disponibles	Manejo de excepciones mediante notificación

Figura 8.1: Casos de Prueba Caja Negra

<b>CP-1</b>	Consulta General
<b>Propósito</b>	Permitir a los usuarios realizar una consulta
<b>Prerrequisitos</b>	El usuario debe estar en la actividad Consulta
<b>Datos de entrada</b>	Objetivo, Duda
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceder a la sección Consulta</li> <li>2. Seleccionar Objetivo</li> <li>3. Plantear Duda</li> </ol>
<b>Resultado Esperado</b>	El usuario ve la respuesta coherente a la duda, donde en ella además se indica si la pregunta planteada se puede responder a partir del objetivo elegido. También observa el historial de la conversación actualizado
<b>Resultado Obtenido</b>	Resultado Esperado

Tabla 8.1: Caso de Prueba CP-1: Consulta General

<b>CP-2</b>	Generación de Problema
<b>Propósito</b>	Permitir a los usuarios generar un problema
<b>Prerrequisitos</b>	El usuario debe estar en la actividad Consulta
<b>Datos de entrada</b>	Tipo de Problema, Supuesto o Temática
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceder a la sección Problemas</li> <li>2. Seleccionar Tipo de Problema</li> <li>3. Indicar Supuesto o Temática</li> </ol>
<b>Resultado Esperado</b>	El usuario ve el problema generado, el cual es coherente con la configuración señalada. Además del historial de la conversación actualizado
<b>Resultado Obtenido</b>	Resultado Esperado

Tabla 8.2: Caso de Prueba CP-2: Generación de Problema

<b>CP-3</b>	Generación de Test
<b>Propósito</b>	Permitir a los usuarios responder un cuestionario
<b>Prerrequisitos</b>	El usuario debe estar en la actividad Tipo Test
<b>Datos de entrada</b>	Número de Preguntas, Objetivo, Historia, Criterios, Dificultad, Opciones Test
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceder a la sección Tipo Test</li> <li>2. Indicar el Número de Preguntas</li> <li>3. Seleccionar Objetivo</li> <li>4. Elegir Historia</li> <li>5. Marcar Criterios</li> <li>6. Especificar Dificultad</li> <li>7. Pulsar botón Generar Test</li> <li>8. Señalar Respuestas</li> </ol>
<b>Resultado Esperado</b>	El usuario ve las diferentes preguntas planteadas, las cuales concuerdan con la configuración y además, son coherentes y selecciona aquellas que considera correctas. Al finalizar el test, se muestra un feedback de su rendimiento
<b>Resultado Obtenido</b>	Resultado Esperado

Tabla 8.3: Caso de Prueba CP-3: Generación de Test

<b>CP-4</b>	Recuperación RAG
<b>Propósito</b>	Recuperación relevante de información desde ChromaDB, en función de la similitud fragmentos a la pregunta o filtros
<b>Prerrequisitos</b>	La base de datos debe de existir, así como las distintas funciones que permiten el flujo de RAG
<b>Datos de entrada</b>	Objetivo y Duda
<b>Procedimiento</b>	Una vez que se selecciona el objetivo e incluye la consulta correctamente: 1. Filtrar por metadatos objetivo 2. Convertir pregunta a embedding 3. Recuperar 10 fragmentos más similares 4. Aplicar reranking para obtener los 3 más coherentes con la pregunta 5. Construir LLM, donde en el prompt se indique si la calidad del contexto recuperado es significativa 6. Mostrar respuesta
<b>Resultado Esperado</b>	El usuario ve el problema generado, el cual es coherente con la configuración señalada. Además del historial de la conversación actualizado
<b>Resultado Obtenido</b>	Resultado Esperado

Tabla 8.4: Caso de Prueba CP-4: Recuperación RAG

<b>CP-7</b>	Fallo API LLM
<b>Propósito</b>	Manejar errores de conexión al LLM
<b>Prerrequisitos</b>	La implementación de la construcción del LLM y llamda a la API debe existir, así como las funciones necesarias para permitir el flujo de RAG y generación de respuesta
<b>Datos de entrada</b>	Cualquier llamada a la API, por ejemplo, pasarle al LLM el prompt, contexto y pregunta de caso de prueba consulta
<b>Procedimiento</b>	Una vez que se realiza la petición, validar si ocurre una excepción o no. En caso afirmativo, informar al usuario de que no se ha podido generar la respuesta requerida
<b>Resultado Esperado</b>	Mostrar mensaje de error informativo
<b>Resultado Obtenido</b>	Resultado Esperado

Tabla 8.5: Caso de Prueba CP-7: Fallo en la API del LLM

### Caja Blanca

Por el contrario, las pruebas de caja blanca se enfocan en evaluar la estructura interna del código, así como coberturas y flujos. En entornos reales, son claves para la calidad del *software*.

La Figura 8.2 resume las pruebas de caja blanca llevadas a cabo.

ID	Nombre	Descripción	Componente Evaluado	Criterio de Aceptación
CP-10	Progreso	Dashboards del usuario	Ver Progreso	Las métricas deben calcularse correctamente, mostrar el rendimiento, la correcta actualización del progreso, precisión en los datos reflejados en los dashboards
CP-11	Cambio de Actividades	Integración entre módulos	Sistema: flujo	Transiciones sin errores ni inconsistencias
CP-12	Calidad de Respuestas LLM	Evaluación de respuestas generadas	Integración LLM	Respuestas deben usar el contexto relevante y ser coherentes
CP-13	Filtrado	Manejo de fallos en el sistema de filtrado	Recuperación	El sistema debe continuar sin aplicar el filtro si este falla, sin detener la ejecución
CP-14	Rendimiento	Tiempos de respuesta	Sistema completo	Respuestas en menos de 60 segundos para consultas típicas
CP-15	Procesamiento de Test	Conversión datos test a CSV	Ver Progreso/Test	El 100% de las preguntas deben almacenarse
CP-16	Historial	Persistencia histórica de interacción durante la sesión y actividad	Personalización/RAG/LLM	Coherencia en los diálogos y persistencia adecuada del historial. Uso de <code>st.session_state</code> y personalizar experiencia
CP-17	Almacenar Respuestas	Conversión a JSON de los datos (actividad - pregunta - respuesta LLM - contexto)	Evaluación	Evaluación respuestas para la aceptación
CP-18	Cobertura	Verificación de una serie de coberturas típicas clave	Sistema completo	Debe lograrse cobertura de: - Instrucciones - Decisiones - Múltiples - Caminos

Figura 8.2: Casos de Prueba Caja Blanca

## 8.2. Análisis: Defectos y Mejoras

Tras la ejecución manual de los casos de prueba definidos, se procede a analizar los resultados obtenidos con el fin de identificar posibles defectos, fallos de funcionamiento o áreas susceptibles de mejora dentro del sistema desarrollado y respecto a los requisitos establecidos.

De manera general, se puede distinguir una serie de defectos y áreas de mejora comunes en los distintos casos de prueba implementados. En función de los resultados obtenidos, a continuación se detalla su análisis:

### Consulta General o Generación de Problemas y Test

- En los casos donde se generaba una respuesta a una consulta general, un problema práctico o un tipo de test, se identificaron varios defectos, tales como respuestas inexactas, incompletas, erróneas, irrelevantes, mal formuladas o basadas en información incorrecta, además de posibles alucinaciones del modelo. En otras palabras, las respuestas no se alineaban con las necesidades del usuario, ni en términos de contenido ni de nivel de dificultad.

Para abordar dichos problemas en la generación de problemas y tests, se implementó la selección de los diferentes objetivos, historias o supuestos en los que el usuario podría fundamentarse e incluir consideraciones o especificaciones adicionales sobre el tema, el

nivel de dificultad o cualquier otro detalle relevante para que el sistema pudiera generar una pregunta más precisa. Asimismo, se incorporó el historial de interacciones para mejorar la coherencia y calidad de las respuestas. Por último, también se refinaron los prompts.

- No obstante, la mejora propuesta a futuro para estos tres casos sería la siguiente: antes de enviar la respuesta generada por el LLM, sería recomendable evaluar su calidad utilizando *Retrieval-Augmented Generation Assessment Scores* (RAGAS). Además, para optimizar la precisión y relevancia de las respuestas, una opción podría ser aumentar el número de fragmentos recuperados para el contexto. En el caso específico de los problemas y test, también se propone mejorar los metadatos, los filtros temáticos, afinar las etiquetas y reforzar el control del contexto recuperado. Sin embargo, esto sería un trabajo a mayores, y sería necesario contar con unos datos más estructurados y una capacidad de cómputo mayor.

### Resolución de Test

- En la resolución de los tests, el principal error dado fue la incapacidad de identificar la respuesta correcta y, con ello, la presentación incorrecta de los datos en el *feedback*. Durante el desarrollo, se produjo este problema, ya que el sistema no procesaba correctamente los saltos de línea y causaba un mal formato de la respuesta. Esto resultó en que la solución correcta y los datos se mostraban de manera errónea.
- Como solución final, se implementó un proceso para escapar los caracteres especiales y representarlos en formato ASCII, lo que permitió que los saltos de línea y otros caracteres especiales se manejaran correctamente. Por lo tanto, se recomienda mejorar la validación del *input* y asegurarse de que el texto de la respuesta esté correctamente escapado, para obtener resultados verídicos y precisos. Además, es importante comprobar que todos los cálculos se sumen correctamente y que la información se muestre de manera adecuada en la interfaz de usuario.

### Visualización del Progreso

- En esta parte, se puede producir que los datos mostrados no concuerden con la realidad. Esto se puede deber a un mal proceso de almacenamiento de los datos, lo que generaba inconsistencias en las estadísticas presentadas al usuario.
- La mejora propuesta radica en garantizar que los datos se almacenen y actualicen correctamente y de forma continua. Para ello, tras la finalización de cada test, se guardan los resultados de forma permanente en el archivo CSV, asegurando así que la información visualizada sea precisa y refleje correctamente el rendimiento del usuario a lo largo del tiempo.

### Recuperación de Información y Manejo de Consultas Ambiguas

- Un aspecto importante a destacar es que los fragmentos recuperados de ChromaDB pueden ser irrelevantes o repetidos, lo que influye directamente en el contexto que se pasa al modelo LLM, y como consecuencia, disminuye la calidad de la respuesta generada. Además, las consultas ambiguas pueden llevar a respuestas erróneas si el sistema no solicita más

información, especificación o aclaración al usuario respecto a su pregunta anterior. También, en casos de baja similitud entre la pregunta y los fragmentos, el sistema no siempre obtiene el contexto adecuado.

- Para solucionar estos problemas, se sugiere ajustar el umbral de similitud entre los fragmentos recuperados y la pregunta, e incluir un *reranker* para mejorar la relevancia de los fragmentos seleccionados. Además, sería imprescindible incluir una lógica para detectar cuando la similitud es baja y, en esos casos, solicitar al usuario más detalles para refinar la respuesta e incluir el historial previo de interacciones. Asimismo, si la pregunta no tiene relación con el objetivo (a partir del contexto recuperado no se puede responder la pregunta), incluir en la respuesta un aviso. También sería recomendable aumentar el número de fragmentos recuperados y realizar una evaluación con RAGAS para asegurar que el contexto proporcionado sea el más relevante y adecuado, aunque esto último a futuro.

## Otros

Por otro lado, se identificaron otros casos de prueba que, aunque críticos, son menos importantes que los problemas principales mencionados anteriormente. Son tales como el cambio entre tipos de actividad, la entrada vacía u otros errores y el historial de interacciones.

- Al cambiar de actividad, puede haber errores o inconsistencias de estado, por lo que se plantea resetear los estados y añadir un *feedback* visual del cambio para mejorar la experiencia del usuario.
- En cuanto a la entrada vacía al recuperar los fragmentos o generar la respuesta con el LLM, puede producir incoherencias o tiempos de espera innecesarios. Por lo tanto, una validación previa y la visualización de un mensaje de error o advertencia serían suficientes para evitar el envío de una entrada vacía. Asimismo, es necesario validar el estado de la API, así como el acceso, carga y recuperación correcta de la base de datos vectorial e incluir un flujo adicional de posibles excepciones.
- Finalmente, existe el riesgo de que el historial de interacciones previas no se mantenga coherente o recoja todas las actividades, cuando solo debería registrar la relevante. Para solucionar esto, sería adecuado utilizar una variable de sesión y gestionarlo por actividad, asegurando que solo se registre lo pertinente para cada una.

A modo de resumen comentar que a lo largo de la evaluación de los casos de prueba, se han distinguido varios defectos comunes que afectan la efectividad del sistema. Entre ellos destacan las alucinaciones generadas por el modelo LLM, (la posible falta de personalización efectiva), la recuperación de fragmentos irrelevantes y problemas con la validación de entradas. Para resolver estos problemas, las soluciones comunes incluyen mejorar las etiquetas y metadatos utilizados en la recuperación de fragmentos, así como su segmentación inicial, aumentar el número de fragmentos recuperados (mediante el parámetro top-k), utilizar herramientas como RAGAS para refinar y evaluar el contexto y conectar el sistema con el dashboard para proporcionar un *feedback* correcto y adaptado al rendimiento del estudiante.

El sistema ha demostrado ser funcional en muchos aspectos, pero también se ha encontrado con varios retos que requieren ajustes. Mejorar la recuperación de información, la personalización del contenido, y la validación de respuestas son áreas clave para incrementar la precisión y

la efectividad del sistema. Finalmente, implementando múltiples de las propuestas de mejora sugeridas, se han superado todos los casos y se ha conseguido un sistema apto, más robusto y adaptado a las necesidades individuales de los usuarios.

Por otro lado, resaltar que habría otras mejoras más relevantes y sustanciales, pero que no se incluyen en el alcance actual del proyecto. Por tanto, se profundizan sobre dichas implementaciones adicionales en el apartado 10, donde se especificará el trabajo a futuro, y por tanto, se profundizará en múltiples mejoras.

# Parte III

# Resultados



# Capítulo 9

## Aceptación

El propósito de esta sección aborda el diseño y ejecución de un plan que permita evaluar el cumplimiento de los objetivos establecidos en el proyecto. Para ello, en primer lugar se definen las métricas de éxito, más adelante se detalla un plan de aceptación para asegurar una evaluación estructurada y rigurosa del producto, y finalmente, se interpretan los resultados obtenidos.

### 9.1. Diseño experimental

El diseño experimental expone de forma detallada el proceso que se va a llevar a cabo para realizar la aceptación del producto. Esto implica varios pasos:

1. Especificar el conjunto de métricas que se utilizarán para valorar la calidad del producto.
2. Describir los conjuntos de datos que se utilizarán en el proceso de aceptación.
3. Detallar el plan de pruebas que se va a seguir, determinando cada una de las pruebas que se van a llevar a cabo e indicando posibles interdependencias entre ellas.
4. Destacar el entorno en el que se va a llevar a cabo la evaluación, tanto de manera *hardware* como *software*.

### 9.2. Métricas de Éxito

A continuación, se presentan una serie de métricas de éxito usadas para evaluar el cumplimiento de los objetivos del proyecto así como su calidad.

#### 9.2.1. *Retrieval-Augmented Generation Assessment Scores - RAGAS*

Retrieval-Augmented Generation Assessment Scores o RAGAS, es una herramienta de medida que permite evaluar sistemáticamente la calidad de las respuestas producidas por sistemas de generación aumentada por recuperación; es decir, que empleen la técnica RAG [107]. Debido a la naturaleza del RAG, se considera tanto la precisión, relevancia e información actualizada de los fragmentos recuperados, como la coherencia, adecuación y calidad de la respuesta final obtenida a partir del modelo LLM. [274]. Por tanto, para este proyecto, es de especial interés su empleo.

RAGAS evalúa múltiples aspectos clave de las respuestas del LLM teniendo en cuenta tanto la pregunta del usuario como los fragmentos recuperados; es decir, el contexto. Para ello, proporciona un marco para medir objetivamente la eficacia de los diversos componentes dentro de un proceso RAG. Entre sus métricas principales se encuentran: [208] [107] [2] [271]

- *Faithfulness* (Fidelidad): mide la fidelidad de la respuesta, considerándose como fiel si todas sus afirmaciones pueden respaldarse en el contexto recuperado. Por tanto, mide la consistencia factual de la respuesta generada con respecto al contexto dado. Es un factor esencial para detectar posibles alucinaciones del modelo, lo cual se obtiene a partir de la siguiente fórmula:

$$\text{Faithfulness Score} = \frac{\text{Nº de afirmaciones en la respuestas respaldadas por el contexto recuperado}}{\text{Nº total de afirmaciones en la respuestas}}$$

Siendo el valor máximo 1, si la afirmación es respaldada, o 0 si no lo es.

- *Answer Relevance* (Relevancia de la Respuesta): como su nombre indica, mide la relevancia de la respuesta en función de la pregunta del usuario. Realmente, evalúa la similitud semántica entre los *embeddings* mediante la similitud del coseno para cuantificar la relevancia de la respuesta con respecto a la entrada:

$$\text{Answer Relevance} = \frac{1}{N} \sum_{i=1}^N \text{cosine similarity}(E_{g_i}, E_o)$$

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^N \frac{E_{g_i} \cdot E_o}{\|E_{g_i}\| \|E_o\|}$$

**Donde:**

- $E_{g_i}$ : *Embedding* de la  $i$ -ésima pregunta generada.
- $E_o$ : *Embedding* de la entrada del usuario.
- $N$ : Número de preguntas generadas (por defecto es 3).

Un valor próximo a 1 indica que la respuesta respuesta es relevante al abordar directa y adecuadamente la pregunta original.

- *Context Recall* (Recuerdo del Contexto): evalúa la relación semántica entre la información recuperada y la respuesta final generada, como si fuera esta la verdad fundamental. Entonces, mide el grado en el que los fragmentos contienen información relevante con respecto a la pregunta del usuario. En otras palabras, indica qué tan relevante es el contenido recuperado respecto la pregunta propuesta por el usuario. Se basa en la similitud semántica promedio entre los fragmentos recuperados y la pregunta:

$$\text{Context Recall} = \frac{\text{Nº de afirmaciones en la respuestas respaldadas por el contexto recuperado}}{\text{Nº total de afirmaciones en la respuestas}}$$

Un valor alto (cercano a 1) indica que se han recuperado fragmentos útiles y coherentes con la temática preguntada.

- *Context Precision* (Precisión del Contexto): mide la proporción de fragmentos del contexto que han sido utilizados o referenciados en la respuesta final generada. Se expresa mediante la fórmula, donde valores altos reflejan un uso más eficiente y específico del contexto en la generación.

$$\text{Context Precision}@K = \frac{\sum_{k=1}^K (\text{Precision}@k \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}}$$

$$\text{Precision}@k = \frac{\text{true positives}@k}{\text{true positives}@k + \text{false positives}@k}$$

- *Context Entity Recall* (Recuerdo de Entidades del Contexto): proporciona la medida del recuerdo del contexto recuperado, cuántas entidades claves han sido realmente usadas en la respuesta. Se entiende como entidad clave a cualquier término (persona, lugar, organización, concepto, etc.) que sea esencial para comprender o verificar el contenido de la referencia. Para calcular esta métrica, se utilizan dos conjuntos:

- $E_r$ : El conjunto de entidades en la Referencia (RE).
- $E_c$ : El conjunto de entidades en el Contexto Recuperado (RCE).

Se calcula el número de entidades comunes a ambos conjuntos ( $E_r \cap E_c$ ) y se divide por el número total de entidades en la referencia ( $|E_r|$ ). La fórmula es:

$$\text{Context Entity Recall} = \frac{|E_r \cap E_c|}{|E_r|}$$

$$\text{Context Entity Recall} = \frac{\text{Nº de entidades comunes entre RCE y RE}}{\text{Nº total de entidades en RE}}$$

Este valor varía entre 0 y 1, donde 1 indica que todas las entidades relevantes del contexto han sido utilizadas en la respuesta.

- *Answer Semantic Similarity*: evalúa la similitud semántica entre la respuesta generada y una respuesta de referencia (*ground truth answer*), calculada mediante similitud del coseno entre los *embeddings*. Un valor cercano a 1 indica una alta similitud.
- *Answer Correctness*: implica medir la precisión de la respuesta generada en comparación con la realidad. Mide tanto la exactitud factual como la similitud semántica entre la respuesta generada y la respuesta ideal o *ground truth*.

Con la combinación de estas métricas, se permite conocer si una respuesta generada está correctamente fundamentada para poder así mejorar el proceso de recuperación y diseño del prompt enviado al LLM, con su contexto y pregunta. Por tanto, RAGAS facilita la posibilidad de una evaluación objetiva y automática, minimizando la dependencia de una revisión manual [107] [95] [108] [109]. Resaltar que este proceso de evaluación se realiza mediante un LLM adicional que implementa la propia herramienta. Para su uso, es necesario disponer de una clave API proporcionada por algún distribuidor de LLM, aunque sin que exista limitaciones estrictas de los modelos o proveedores empleados. En este caso, como se cuenta con presupuesto y con una

clave API de OpenAI para el funcionamiento del asistente de aprendizaje, se ha decidido también utilizarla para la evaluación del sistema.

Para satisfacer las necesidades del proyecto, se considera el empleo de las múltiples métricas que contiene RAGAS. No obstante, no se utilizarán todas, si no aquellas que se consideran más importantes para el proyecto, como son:

- Faithfulness (F)
- Answer Relevance (AR)
- Context Recall (CRC)
- Context Precision (CP)

La razón de esta selección radica en que estas métricas permiten evaluar de manera integral la relevancia y fidelidad de la respuesta en función a la calidad y uso del contexto recuperado de la asignatura. Para satisfacer correctamente los criterios de aceptación, el umbral de éxito para cada una será el mostrado a continuación. En cambio, destacar que esta elección se basan en prácticas comunes en sistemas RAG existentes [283, 260, 108]:

- **Faithfulness (F) > 0,7:** garantiza respuestas fieles al contexto, reduciendo alucinaciones.
- **Answer Relevance (AR) > 0,7:** asegura que las respuestas sean relevantes y respondan a la intención y necesidad del usuario.
- **Context Recall (CRC) > 0,5:** intenta asegurar que al menos la mitad del contexto sea relevante para generar la respuesta.
- **Context Precision (CP) > 0,5:** controla que la mayoría del contenido incluido sea relevante, evitando ruido.

Por otro lado, se ha decidido no utilizar otras métricas, como *Answer Semantic Similarity* o *Answer Correctness*, debido a que requieren de una respuesta predefinida o *ground truth* para su evaluación. En consecuencia, dado que los sistemas de IA Generativa basados en RAG no necesariamente producen respuestas exactas a un conjunto fijo de respuestas correctas, estas métricas no son apropiadas en este contexto. En el caso de *Context Entity Recall*, no se empleará debido a que englobaría el concepto sistema de bases de datos como uno solo, y no los diferentes criterios de aprendizaje especificados en la asignatura, obteniendo por tanto valores muy bajos. Por otro lado, resaltar que *Context Precision* dependerá directamente de la calidad de los documentos proporcionados. En definitiva, las métricas seleccionadas son más adecuadas para evaluar la calidad contextual y semántica de las respuestas sin la necesidad de una respuesta ideal predefinida.

Finalmente, es necesario destacar la importancia de la evaluación en RAG para poder garantizar la precisión de las respuestas, medir el impacto de los posibles cambios efectuados, detectar errores, inconsistencias o situaciones no óptimas y continuar con una mejora constante para asegurar un mantenimiento alineado con los objetivos y necesidades del usuario.

### 9.2.2. Tiempo de Respuesta

Otro aspecto importante a considerar es el tiempo de respuesta. Este será el tiempo transcurrido entre que el usuario envía su consulta o petición hasta que recibe la respuesta del modelo o asistente. Para lograrlo de una manera satisfactoria, este tiempo deberá ser inferior a 1 minuto. Cabe destacar que esta métrica puede variar en función del entorno de evaluación, ya que influyen factores como el dispositivo *hardware* (limitaciones de recursos como GPU, CPU, RAM...), calidad de conexión a la red y la concurrencia o simultaneidad de procesos en ejecución en local. En este proyecto, el único recurso externo involucrado es el acceso a la API, lo cual también puede afectar el tiempo de respuesta, retrasándola, dependiendo de la disponibilidad y latencia de OpenAI.

### 9.2.3. Tiempo de Carga

De manera similar al anterior, se considera el tiempo de carga que tarda en mostrarse los distintos elementos de la interfaz. En este caso, el umbral de éxito estará por debajo de 10 segundos. Al igual que en el tiempo de respuesta, el tiempo de carga también depende del entorno de evaluación. En este caso por aspectos como el *hardware*, concurrencia o simultaneidad de procesos en ejecución en local.

### 9.2.4. Usabilidad Visual de la Interfaz

Relacionado con la interfaz, la experiencia de usuario es un factor clave para garantizar la eficacia y aceptación del asistente por parte del usuario final. En este sentido, se define esta métrica de usabilidad visual de la interfaz que se centra en la percepción subjetiva de claridad, organización y facilidad de uso del asistente. Esta evaluación se enfocará en aspectos como:

- Uso de colores adecuados (evitando combinaciones agresivas o poco accesibles).
- Estructura coherente y organizada de los elementos.
- Coherencia visual en los cambios de estado o navegación (cambios visibles).
- Claridad en la disposición de botones, gráficos e información textual.

Para valorar esta métrica, se usarán valores binarios de *Si/No*, dependiendo de si se cumple o no con los aspectos anteriores y aplicando los distintos principios básicos de usabilidad y diseño visual. Para ello, sería el usuario quien valorara este aspecto. Sin embargo, aprovechando que el profesor o experto debe también evaluar al asistente, será este también que mida la usabilidad visual de la interfaz.

### 9.2.5. Satisfacción del Profesor

Por último, como parte de la evaluación final, las respuestas generadas por el asistente serán evaluadas por profesores expertos en la materia. Para ello, se enviará un archivo Excel donde se incluirán las preguntas realizadas por los usuarios, así como configuración del tipo cuestiones, test y problemas, junto con las respuestas generadas por el modelo y su contexto. Los profesores deberán calificar la calidad de las respuestas en una escala del 1 al 10, teniendo en cuenta criterios como claridad, precisión, coherencia y relevancia. Esta métrica aborda una valoración

más subjetiva y centrada en tres aspectos esenciales: la utilidad de la respuesta generada, su alineación con los objetivos académicos y alineación con las necesidades del estudiante y contexto general. Por tanto, el experto referente valorará cuestiones como:

- Si la respuesta tiene sentido en la relación al contexto proporcionado.
- Si la respuesta resulta relevante dentro de un enfoque pedagógico o el *framework* eXtreme Learning para profundizar y conseguir los distintos criterios de aprendizaje.
- Si la respuesta se adapta a la situación, nivel de dificultad y perfil del usuario.
- Si la respuesta incluye todos los elementos claves a valorar, permitiendo que sea el alumno quien desarrolle el ejercicio sin que le proporcione ningún tipo de solución directa.
- Si la respuesta no presenta incoherencias o si no existen inconsistencias.

El umbral de éxito para esta métrica se establece en una puntuación media igual o superior a 7. Esta métrica resulta fundamental para obtener una evaluación cualitativa del sistema desde la perspectiva de los expertos, permitiendo identificar áreas de mejora y asegurar que las respuestas cumplen con los estándares educativos establecidos para la materia de Sistema de Bases de Datos, lo que contribuirá a la mejora continua del sistema en este proyecto académico.

#### 9.2.6. Otras herramientas y métricas de evaluación

En el mundo actual, existen muchas otras métricas adicionales como BLEU, ROUGE o METEOR, que se encuentran más orientadas a la comparación entre el texto generado y un texto de referencia previamente definido. Estas métricas son consideradas como verdad de campo o *ground truth*, ya que necesitan de una respuesta ideal o referencia para compararlas con el resultado generado [346] y obtener el valor. No obstante, no se han tenido en cuenta para este proyecto por diversas razones.

Para empezar, no existe siempre esa respuesta idílica predefinida que permita aplicar este tipo de métricas de forma objetiva. Por otro lado, estas métricas no permiten evaluar todo el proceso completo implicado en la generación aumentada por recuperación, donde intervienen tanto la calidad de los fragmentos recuperados como la coherencia y adecuación de la respuesta generada [245] [199] [29].

Por otra parte, una línea actual de investigación presenta el enfoque *ReGReT* (*Relevance-Grounded Beam Search for RAG*), también conocido como RGB-RAG [345]. Este método propone mejorar la fase de recuperación, mediante una búsqueda centrada en la relevancia, estructura o ciertos criterios de recuperación. Sin embargo, se fundamenta también en verdades y se encuentra desarrollada en inglés y chino, sin implementación actual del idioma español.

Respecto otras herramientas, cabe mencionar LangSmith [188], desarrollada por los creadores de LangChain. Esta infraestructura facilita la trazabilidad, análisis o evaluación del asistente, complementando la aceptación del proyecto. No obstante, no se ha utilizado debido a la necesidad de servicios externos que dependen de la creación de una cuenta en la plataforma y con recursos limitados.

Por otro lado, RAGAS ha sido específicamente diseñada para evaluar sistemas de generación aumentada por recuperación, de forma integral y habilitada en español. Además trata métricas sin verdad de campo, pero también basada en las respuestas del LLM, lo que implica que no

depende de una respuesta predefinida para poder evaluar el *output* resultante. Aunque, como bien se ha mencionado, puede incluir métricas para evaluar un hipotético conjunto de respuestas predefinidas. Esta herramienta permite una valoración más completa y precisa tanto del proceso de recuperación de la información como de la generación final de respuestas. Por ello, la convierte en la opción ideal para seleccionarla junto a las otras métricas ya descritas.

### 9.2.7. Resumen Métrica de Éxito

A modo de resumen, la siguiente tabla 9.1 presenta las métricas que se emplearán.

Sigla	Métrica
F	Faithfulness (Fidelidad)
AR	Answer Relevance (Relevancia de la Respuesta)
CRC	Context Recall (Recuerdo del Contexto)
CP	Context Precision (Precisión del Contexto)
TR	Tiempo de Respuesta
TC	Tiempo de Carga
UI	Usabilidad de la Interfaz
SP	Satisfacción del Profesional

Tabla 9.1: Métricas de Evaluación del Sistema

Cada métrica está diseñada para evaluar una dimensión crítica del sistema: rendimiento de las respuestas, coherencia con el contexto recuperado, eficiencia técnica, y la percepción tanto del usuario final como de los expertos.

### 9.2.8. Recursos

Para poder realizar el análisis posterior de los resultados es necesario el uso de distintas librerías, en particular rargas, así como la creación del Excel de los profesores y un archivo que contenga las distintas preguntas-respuesta-contexto, tiempos y otro con el script de evaluación. Del mismo modo, se ha de realizar de manera posterior a la creación final del asistente.

### 9.2.9. Datos

Para llevar a cabo la evaluación de las distintas métricas mencionadas, es necesario contar con un conjunto de datos que incluya tanto la respuesta generada por el asistente, como la pregunta del usuario y el contexto recuperado. Esta información se almacena en un archivo JSON de historial, con el fin de facilitar su análisis posterior. Para ello, se emplea la función `guardar_historial()`, ya explicada en el apartado de implementación.

La creación de este conjunto de datos se realiza de forma manual, introduciendo dos tipos de preguntas:

1. **Consultas:** centradas en aspectos descriptivos y relevantes de la asignatura, como conceptos clave o temas fundamentales. Todo ello en base a un objetivo, es decir, sin contar

aquellas cuestiones no relacionadas con este.

2. **Preguntas tipo test o problemas:** en estos casos se proporcionan indicaciones específicas o consideraciones a tener en cuenta para que el asistente genere preguntas de opción múltiple o enunciados de ejercicios relacionados con el contenido de la materia:

- Tipo test: se incluiría la pregunta (los criterios) en los que fundamentar el test.
- Problemas: contendría el tipo de problema, así como su supuesto o temática.

Para asegurar que los datos sean lo suficientemente representativos y variados, se describirá en el apartado siguiente ejemplos de diseño pruebas dentro de cada tipo de actividad.

Por último, destacar que igualmente será necesario obtener los datos de tiempo de respuesta total y carga, todo ello en segundos, y almacenado en un CSV.

### 9.3. Plan de aceptación

La selección de métricas se ha realizado con el objetivo de cubrir distintas dimensiones clave del sistema: el rendimiento de respuesta, la calidad del contenido generado en relación con el contexto recuperado, la eficiencia técnica y la percepción del usuario final.

El presente plan de aceptación tiene como finalidad asegurar una evaluación estructurada y rigurosa del producto desarrollado. Para ello, se establecerán una serie de etapas que permitan validar el cumplimiento de los criterios definidos, en función de las métricas seleccionadas.

Cada etapa del plan describe los pasos a seguir, los recursos necesarios, los resultados esperados y el modo en que se recopilarán y analizarán los datos obtenidos durante el proceso de validación del proyecto. Por tanto, a partir de las métricas seleccionadas en este apartado se llevará a cabo la evaluación del proyecto.

Por otro lado, es importante destacar que, en los casos de generación de test y problemas, algunas métricas evaluadas mediante RAGAS pueden presentar valores más bajos, especialmente en lo relativo a fidelidad y precisión del contexto. Esto se debe a que en estas respuestas donde se genera contenido principalmente nuevo, aunque se base en un contexto proporcionado, tienden a recibir una puntuación inferior en comparación con tareas de tipo pregunta-respuesta, donde la solución se basa en un contexto delimitado [108].

Esa valoración no implica necesariamente una menor calidad en los resultados, sino que refleja las limitaciones del marco de evaluación en la actualidad cuando se aplica a apartados más creativos que no se restringen a respuestas textualmente cerradas, las cuales se evalúan en función de la similitud semántica.

Por tanto, para resolver el problema que presentan las métricas en cuanto a una evaluación de generación de texto creativa, se considerará más relevante la opinión de los expertos a la hora de elaborar las conclusiones oportunas. Esta decisión radica en que no se dispone, en el marco de este TFG, de una herramienta adecuada para evaluar contenido creativo mediante IA Generativa. Bien es cierto que la mayoría de las herramientas existentes actuales requieren una *ground truth*, pero no se cuenta con ello para este proyecto. No obstante, se podría plantear aquellas basadas en modelos LLM, conocidas como *LLM-as-a-Judge*, pero que aun presentan varios desafíos [116, 342]. Como consecuencia de todo ello, se reconoce la dificultad de aplicar métricas automáticas a tareas creativas, por lo que se concluye en que se valorará más la opinión de los expertos a la hora de realizar las conclusiones.

A continuación, se prosigue con el desarrollo del apartado.

### 9.3.1. Fase 1: Preparación

En primer lugar, es necesario inicializar el asistente de aprendizaje. Para ello, hace falta crear un entorno de ejecución que incluya con una serie de importaciones y configuraciones previas, ya que se ejecuta sobre Streamlit y con el *framework* Langchain, entre otras dependencias. Todo ello se recoge en la parte de los apéndices.

Una vez iniciado el asistente, se da pie a diversas actividades y, en consecuencia, a la fase de ejecución y evaluación de las pruebas. No obstante, antes de describir cada prueba, es imprescindible almacenar la información de las distintas interacciones entre el usuario - asistente. Para ello, como bien se ha especificado en el apartado de Datos, se hará uso de la función definida en la implementación: `guardar_historial()`, que permite guardar el historial de la conversación entre el asistente y el usuario (petición-respuesta, por tipo de actividad), además del contexto recuperado para cada respuesta generada en cada interacción. Gracias a la utilización de dicha función, se permitirá abarcar la fase de análisis y conseguir una evaluación objetiva del sistema.

Asimismo, para medir los tiempos de respuesta y carga, se implementa `guardar_tiempo_respuesta()` en el *backend*, donde toma el tiempo total, de búsqueda, *reranking* y modelo LLM; y `guardar_tiempo_carga()` para medir el tiempo de carga de cada actividad en el *frontend*. Todo ello, se almacena en un CSV con la información correspondiente.

Finalmente, resaltar que el primer paso para la ejecución de las pruebas y su evaluación será la creación del nuevo entorno, el cual se especifica más adelante en el apartado anexos correspondiente.

### 9.3.2. Fase 2: Ejecución de Pruebas

La separación de la evaluación del proyecto por bloques facilita una organización más clara y permite realizar un análisis más enfocado y detallado de cada parte del sistema. De esta manera, cada sección del asistente de aprendizaje se puede evaluar de forma independiente, lo que también simplifica la identificación de posibles problemas o áreas de mejora futuras.

En esta fase, se llevarán a cabo las siguientes actividades para cada bloque:

- Creación de pruebas: se diseñarán pruebas representativas donde se recoja el flujo de respuestas - preguntas por objetivo y escenarios representativos de funcionalidades clave del sistema. Estas interacciones se introducirán manualmente a través de la interfaz desarrollada para el asistente. Se almacenarán en archivos para su posterior análisis y evaluación, incluyendo tanto las preguntas planteadas como las respuestas generadas, los fragmentos de contexto recuperado y otra información procesada relevante.
- Evaluación de métricas: a partir de los archivos generados con las preguntas y respuestas, se empleará la librería RAGAS para evaluar las métricas del sistema de manera automatizada. Aunque la entrada de los datos se realiza de forma manual, su recopilación (por medio de `guardar_historial()`) y generación del archivo JSON será automática.
- Ejecución de las pruebas planteadas. Finalmente, se llevaría a cabo el proceso de ejecución de las pruebas para así obtener los resultados. Para ello, se ejecutarán los siguientes scripts de python sobre Jupyter Notebook. Destacar que en una primera instancia se preparan los datos 9.1, y después es cuando se realiza la ejecución de las pruebas 9.2.

```

from datasets import Dataset
from ragas.metrics import faithfulness, answer_relevancy, context_recall, context_precision, context_entity_recall
from ragas.evaluation import evaluate
import pandas as pd
from datasets import Dataset
import os

Inicializar la API KEY
os.environ["OPENAI_API_KEY"] = "API_KEY"

Abrir el fichero con los datos almacenados
DATA_PATH = "./historial_respuestas.json"
df = pd.read_json(DATA_PATH)

Convertir el campo 'contextos' de string a lista de strings
df["contextos"] = df["contextos"].apply(lambda x: x.split("\n") if isinstance(x, str) else [])

Incluir el parámetro reference (no se usa, es solo para que se pueda ejecutar la librería)
df[["reference"]] = df[["respuesta"]]

Crear un nuevo dataframe por bloque y redefiniendo las columnas
df_ragas = df.rename(columns={
 'pregunta': 'user_input',
 'respuesta': 'response',
 'contextos': 'retrieved_contexts'
})[['tipo', 'user_input', 'response', 'retrieved_contexts', 'reference']]

df_consulta = df_ragas[df_ragas['tipo'] == 'consulta']
df_test = df_ragas[df_ragas['tipo'] == 'test']
df_problemas = df_ragas[df_ragas['tipo'] == 'problemas']

```

Figura 9.1: Preparación Datos

```

Ahora, por cada nuevo dataframe-bloque, se ejecutaría el siguiente código
dataset = Dataset.from_pandas(df_ragas_tipo)

metrics = [
 faithfulness,
 answer_relevancy,
 context_recall,
 context_precision,
 context_entity_recall
]

results = evaluate(dataset, metrics=metrics)
print("---RESULTADOS DE EVALUACIÓN RAGAS ---")
print(results)

```

Figura 9.2: Ejecución

A continuación, se detallan los bloques de evaluación que se implementarán:

### BLOQUE 1: Consulta General

- Dataset:** Este bloque incluirá una serie de 10 preguntas diversas relacionadas con Sistemas de Bases de Datos. Las preguntas serán variadas para cubrir diferentes aspectos de cada objetivo de la asignatura (por lo que no se tendrá en cuenta aquellas cuestiones no relacionadas con el objetivo) y poner a prueba la capacidad del asistente para generar respuestas coherentes y útiles 9.1.

```

{
 "tipo": "consulta",
 "pregunta": "Que es una entidad?",
 "respuesta": "Respuesta por el asistente",
 "contextos": "Fragmentos recuperados relevantes que sean objetivo = Disenio Logico"
}

```

Fragmento de Código 9.1: Ejemplo Prueba Consulta General

- **Evaluación:** Se evaluará la fidelidad, relevancia, recuerdo y precisión de los contextos y de las respuestas generadas. También se medirá el tiempo medio de respuesta y se verificará la calidad de la interfaz y las respuestas de los expertos.
- **Métricas:** Fidelidad, Relevancia, Precisión, Usabilidad, Tiempos medios, Satisfacción Profesor. Es decir, todas. Los resultados de cada pregunta se agruparán más adelante en una tabla para analizar las métricas por cada pregunta. Destacar que los expertos son 3 personas, por tanto, se presentará la media resultante de ellos y por el conjunto global.

## BLOQUE 2: Generación Tipo Test

- **Dataset:** Este bloque se centrará en la generación de 10 preguntas tipo test sobre Sistemas de Bases de Datos. Al igual que en el bloque anterior, la entrada deberá ser variada (en función de cada objetivo, historia y criterios) y se pondrá a prueba la capacidad del asistente para generar cuestiones de opción múltiple respecto a un contexto 9.2.

```
{
 "tipo": "test",
 "pregunta": "Actores , Rol BD, Funciones SGBD",
 "respuesta": "Respuesta por el asistente , que
generaria el test",
 "contextos": "Fragmentos recuperados relevantes
que sean objetivo = SGBDs"
}
```

Fragmento de Código 9.2: Ejemplo Prueba Tipo Test

- **Evaluación:** Se evaluará la claridad de las preguntas generadas, la corrección de las soluciones y la adecuación del contexto usado para generar las respuestas. Además, se analizará el tiempo medio de respuesta y la interacción con la interfaz.
- **Métricas:** Mismas que en el apartado anterior.

## BLOQUE 3: Generación Problemas

- **Dataset:** Este bloque se centrará en la generación 10 de problemas; 5 destinadas a enunciados de modelos ER, y otros 5 a funcionalidades SQL, por lo que conjunto de pruebas en este caso dependerá del tipo de problema y de supuesto o temática. Aun así, se pondrá a prueba la capacidad del asistente de generar casos hipotéticos nuevos, pero relacionados con la asignatura 9.3.

```
{
 "tipo": "problemas",
 "pregunta": "Genera funcionalidades SQL para el
supuesto Flying your Dreams",
 "respuesta": "Respuesta por el asistente , donde
generar a el problema",
 "contextos": "Fragmentos recuperados relevantes
que sean problematica = Flying your Dreams"
```

```
#Es decir , toda la informacion de ese supuesto
},
{
 "tipo": "problemas",
 "pregunta": "Genera un Enunciado ER con
tem tica deportes",
 "respuesta": "Respuesta por el asistente , donde
generar a el problema",
 "contextos": "Fragmentos recuperados relevantes
que sean problematica = Modelo ER"
 #Es decir , ejemplos de otros enunciados
},
```

Fragmento de Código 9.3: Ejemplo Prueba Tipo Test

- **Evaluación:** Se evaluará la claridad del enunciado o funcionalidades generados, la corrección de las soluciones, el contexto recuperado y la adecuación del contexto usado para generar las respuestas. Además, se analizará el tiempo de respuesta y la interacción con la interfaz. Cabe destacar que primero se evaluarán cada tipo de problema por separado, para ver su calidad, y finalmente, ambos de manera conjunta para así ver la diferencia obtenida.
- **Métricas:** Mismas que en el primer apartado.

### 9.3.3. BLOQUE 4: Ver Progreso

- **Dataset:** Este bloque no necesita un dataset predefinido, ya que se encargará en evaluar la funcionalidad de visualización del progreso del estudiante.
- **Evaluación:** Se verificará los tiempos de respuesta y de carga, así como la amigabilidad de la interfaz.
- **Métricas:** Al ser únicamente el flujo del trabajo seleccionar Ver Progreso y comprobar si los datos reflejados el dashboard son correctos; se mide la Usabilidad de la Interfaz y Tiempo de Carga.

### 9.3.4. Fase 3: Presentación de Resultados

Los resultados obtenidos en cada bloque, con respecto a su umbral de éxito, y si este ha sido alcanzado, se muestra en las siguientes tablas:

#### BLOQUE 1

Sigla	Métrica	Resultado	Umbral de Éxito	¿Éxito?
F	<i>Faithfulness</i>	0.8520	>0.7	Sí
AR	<i>Answer Relevance</i>	0.8804	>0.7	Sí
CRC	<i>Context Recall</i>	0.9217	>0.5	Sí
CP	<i>Context Precision</i>	0.8310	>0.5	Sí
TR	Tiempo de Respuesta	5.49s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí
SP	Satisfacción del Profesional	8 <sup>1</sup>	>7	Sí

<sup>1</sup> Experto 1 = 8,7; Experto 2 = 7,6; Experto 3 = 7,7

Tabla 9.2: Evaluación del Sistema en el Bloque 1

#### BLOQUE 2

Sigla	Métrica	Resultado	Umbral de Éxito	¿Éxito?
F	<i>Faithfulness</i>	0.6056	>0.7	No
AR	<i>Answer Relevance</i>	0.7201	>0.7	Sí
CRC	<i>Context Recall</i>	0.6197	>0.5	Sí
CP	<i>Context Precision</i>	0.5290	>0.5	Sí
TR	Tiempo de Respuesta	5.63s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí
SP	Satisfacción del Profesional	8,03 <sup>2</sup>	>7	Sí

<sup>2</sup> Experto 1 = 9,0; Experto 2 = 7,9; Experto 3 = 7,2

Tabla 9.3: Evaluación del Sistema en el Bloque 2

### BLOQUE 3

Enunciados Modelos ER:

Sigla	Métrica	Resultado	Umbral de Éxito	¿Éxito?
F	<i>Faithfulness</i>	0.0000	>0.7	No
AR	<i>Answer Relevance</i>	0.7575	>0.7	Sí
CRC	<i>Context Recall</i>	0.4667	>0.5	No
CP	<i>Context Precision</i>	0.0000	>0.5	No
TR	Tiempo de Respuesta	7.1879s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí
SP	Satisfacción del Profesional	7,73 <sup>3</sup>	>7	Sí

<sup>3</sup> Experto 1 = 8,6; Experto 2 = 7,0; Experto 3 = 7,6

Tabla 9.4: Evaluación del Sistema en el Bloque 3: ER

Funcionalidades SQL:

Sigla	Métrica	Resultado	Umbral de Éxito	¿Éxito?
F	<i>Faithfulness</i>	0.0367	>0.7	No
AR	<i>Answer Relevance</i>	0.0000	>0.7	No
CRC	<i>Context Recall</i>	0.7714	>0.5	Sí
CP	<i>Context Precision</i>	0.2464	>0.5	No
TR	Tiempo de Respuesta	7.4497s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí
SP	Satisfacción del Profesional	9,93 <sup>4</sup>	>7	Sí

<sup>4</sup> Experto 1 = 9,8; Experto 2 = 10,0; Experto 3 = 10,0

Tabla 9.5: Evaluación del Sistema en el Bloque 3: SQL

Bloque 3 en Conjunto:

Sigla	Métrica	Resultado	Umbra de Éxito	¿Éxito?
F	<i>Faithfulness</i>	0.0779	>0.7	No
AR	<i>Answer Relevance</i>	0.3797	>0.7	No
CRC	<i>Context Recall</i>	0.6065	>0.5	Sí
CP	<i>Context Precision</i>	0.1501	>0.5	No
TR	Tiempo de Respuesta	7.7458s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí
SP	Satisfacción del Profesional	8,83 <sup>5</sup>	>7	Sí

<sup>5</sup> Experto 1 = 9,2; Experto 2 = 8,5; Experto 3 = 8,8

Tabla 9.6: Evaluación del Sistema en el Bloque 3: SQL (2)

#### BLOQUE 4

Sigla	Métrica	Resultado	Umbra de Éxito	¿Éxito?
TR	Tiempo de Respuesta	<1s	<60s	Sí
TC	Tiempo de Carga	<1s	<10s	Sí
UI	Usabilidad de la Interfaz	Sí	= Sí	Sí

Tabla 9.7: Evaluación del Sistema en el Bloque 4

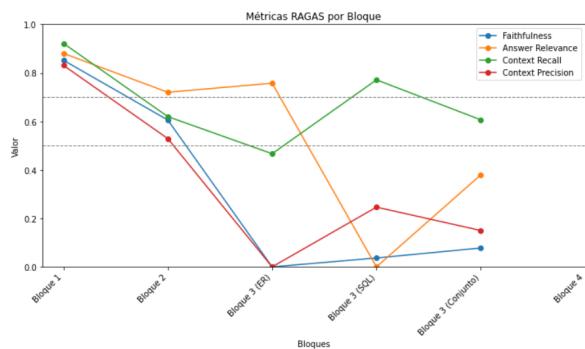


Figura 9.3: Gráfica Evaluación Métricas RA-GAS

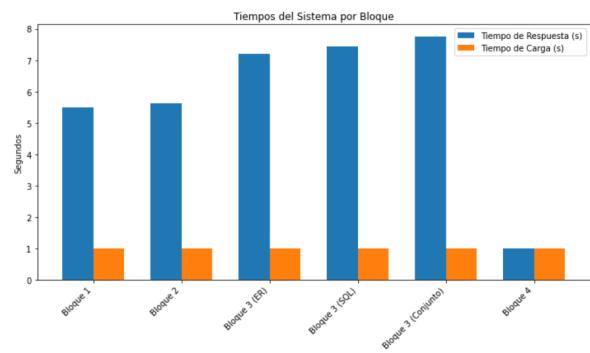


Figura 9.4: Gráfica Evaluación Métricas Tiempos

Satisfacción del Profesional

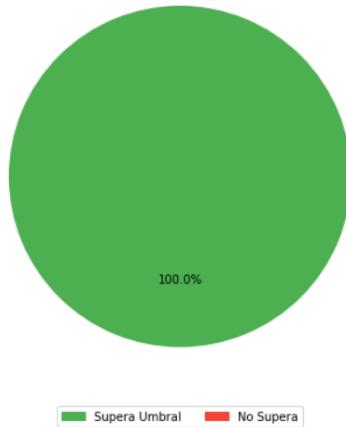


Figura 9.5: Gráfica Evaluación Métrica Satisfacción del Profesional

Usabilidad de la Interfaz

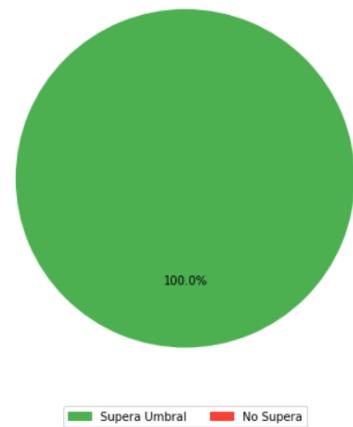


Figura 9.6: Gráfica Evaluación Métrica Usabilidad de la Interfaz

### 9.3.5. Análisis Resultados

A partir de los resultados obtenidos, a continuación se realiza su análisis.

En primer lugar, se aprecia que en el Bloque 1 todos los resultados han sido mayores al umbral de éxito, y en varios de ellos, destacar que con bastante margen. Esto indica una buena calidad tanto de la respuesta generada como del contexto empleado. Asimismo, los profesionales consideran también que el asistente proporciona resultados correctos, por lo que se podría concluir que este bloque ha sido un caso de éxito.

Prosiguiendo con el Bloque 2, únicamente la métrica de *faithfulness* se encuentra por debajo del umbral de éxito, aunque sea por apenas una décima. Por otro lado, las métricas evaluadas con RAGAS presentan valores más pequeños respecto al Bloque 1. En cuanto al resto de medidas, se mantienen de manera similar.

Una fidelidad por debajo del umbral implica que la respuesta no ha sido completamente óptima respecto a los fragmentos recuperados. Sin embargo, esto no es inesperado, ya que las preguntas generadas se fundamentan en el contexto, pero no van a ser iguales a este. Además, las opciones disponibles incluirán a su vez alternativas erróneas. Debido a que la evaluación se basa principalmente en una similitud semántica entre la pregunta generada por el asistente y el contexto, es normal que se obtengan valores por debajo del umbral, así como que presenten valores menores de calidad respecto al anterior bloque.

Dado que la mayoría de las métricas han alcanzado el éxito y únicamente la medida comentada no ha superado el umbral, por poco y por las razones comentadas, se concluye que el caso puede considerarse igualmente aceptado.

Para analizar mejor los resultados, al inicio del plan de aceptación se decidió dividir el Bloque 3 en dos subbloques en función de cada problemática, ya que, al tratarse casos diferentes, podría ocurrir que no se evaluara correctamente. Comenzando por la parte de modelos ER, se aprecia que las métricas de fidelidad, recuperación de contexto y precisión no alcanzan el umbral establecido. Esto se debe en parte también a lo comentado recientemente. La generación de enunciados de modelos de ER se basa en una serie de ejemplos que permiten describir y buscar patrones para

generar la respuesta, pero esta es totalmente creativa. Por ello, aunque se emplea el contexto recuperado como referencia, el contenido no se fundamenta completamente en él, ya que, de ser así, no tendría sentido un asistente que permita crear enunciados para que los estudiantes los practicaran, puesto que siempre se obtendría la misma solución, lo que reduciría su utilidad.

Respecto a las funcionalidades, se aprecia también resultados deficientes en la evaluación directa mediante RAGAS. Sin embargo, en este caso, aunque la respuesta no es relevante, si supera el umbral el contexto. Esto se explica porque el asistente recibe un supuesto claro y debe crear nuevas funcionalidades basándose en dicho contexto. Al ser un proceso creativo, el asistente debe utilizar ese contexto de inspiración y no replicarlo. Por ello, los resultados son bajos en las métricas basadas en similitud semántica, dado que estas comparan textos de forma literal, cuando en realidad el asistente genera contenido nuevo inspirado en el contexto.

Al evaluar de manera conjunta ambos bloques se observa estos patrones mencionados, aunque resaltar que únicamente el contexto supera el umbral debido a los supuestos planteados.

Por lo tanto, podría interpretarse que este caso no ha sido superado. Sin embargo, dado que se intenta evaluar aspectos creativos con RAGAS y orientados a una similitud textual, se ha optado por dar mayor peso a la satisfacción del profesional experto, quien puede valorar de manera más subjetiva la calidad global del resultado.

Finalmente, como los tiempos de respuesta y carga son óptimos, al igual que la usabilidad, y que el valor del experto, se concluye que los resultados generados por el asistente son satisfactorios y por ello, que este caso ha sido superado con éxito.

Por último, en el Bloque 4 todos los casos han sido superados con éxito, ya que únicamente se valoraba el tiempo de carga, el tiempo de respuesta y la usabilidad de la interfaz en la opción de seguimiento del progreso del alumno.

Las desviaciones encontradas respecto a los umbrales establecidos son diversas, pero se atribuyen a la incapacidad e inexistencia de métricas en la actualidad que permitan evaluar de forma efectiva la generación de texto creativo. Esto ya se mencionaba previamente al inicio del apartado, por lo que la solución factible consiste en considerar la satisfacción del profesional experto. Asimismo, es importante destacar que a mayor número de expertos, se obtiene una mejor variedad de validaciones y opiniones de mejora, por lo que de cara al futuro sería interesante incluir la participación de expertos en usabilidad.

De manera general, para mejorar el asistente de aprendizaje, una idea básica sería recuperar un mayor número de fragmentos en el contexto, aumentando por ejemplo de 10 a 20 en número de *embeddings*, y luego de 3 a 10 en el *reranking*. Por otro lado, podría ser necesario continuar reestructurando los datos dentro del marco de eXtreme Learning, con el fin de optimizar el filtrado.

Además, la calidad del material proporcionado influye notablemente en los resultados. Por tanto, para mejorar aún más las respuestas, podría ser aconsejable emplear material que no esté basado únicamente en diapositivas PDF, sino más bien en un formato similar a un libro redactado, tanto en la teoría como en la práctica. Esto es fundamental, ya que la recuperación se realiza mediante similitud semántica.

De esta manera, se podrán obtener respuestas de mayor calidad, adecuación y relevancia.

### 9.3.6. Entorno de evaluación

Finalmente, volver a resaltar que para la ejecución de las pruebas y su evaluación, será necesario un nuevo entorno. La creación de un entorno se especifica en el apéndice.



# Capítulo 10

## Conclusiones y trabajo futuro

Por último, en este capítulo se presentan las conclusiones finales. En primer lugar, se especificarán desde una perspectiva del proyecto, más adelante desde la perspectiva personal y finalmente, se describirán posibles líneas de trabajo futuro e investigación.

### 10.1. Conclusiones

A continuación, se detallan las conclusiones desde un punto de vista del proyecto y personal.

#### 10.1.1. Perspectiva del proyecto

Desde de la perspectiva del proyecto, las conclusiones se pueden estructurar en torno a tres ejes centrales: la consecución de los objetivos dentro del alcance definido, la utilidad del asistente de aprendizaje en el futuro y finalmente, el desarrollo del proyecto dentro de la metodología de trabajo seleccionada.

- **Objetivos específicos:** al inicio del proyecto se planteaban seis objetivos y se puede afirmar que todos ellos se han cumplido de manera satisfactoria. Los capítulos 3 y 4 abarcan a la perfección la investigación sobre el uso de la IA Generativa en el ámbito de la educación para personalizar la enseñanza, tanto sus diversas aplicaciones, proyectos existentes y su posible auge, estructuración y soluciones. Respecto al resto de objetivos, estos se van recogiendo a lo largo del desarrollo del producto. El **OBJ-2** consistía en construir la base de conocimiento de los materiales de la asignatura y estructurar su conocimiento para cimentar el asistente. Esto se detalla a lo largo de los capítulos 5 y 6, gracias a un diseño conceptual y al análisis exhaustivo realizado de los diferentes datos existentes, además de su tratamiento, forma de organización y posible utilización. El **OBJ-3** permitía la resolución de dudas de la asignatura a cerca de todos sus contenidos. La implementación descrita en el capítulo 7 demuestra que se ha logrado con éxito al existir un flujo que responda a la pregunta del usuario a partir de la base de conocimiento anterior y mediante RAG. No obstante, han sido clave los capítulos 5 y 6, donde se define y analiza el caso de negocio, la optimización del flujo de búsqueda y la calidad de las respuestas. El **OBJ-4** consistía en generar cuestiones de elección múltiple basadas en el contenido teórico. Al igual que en el caso anterior, también se ha cumplido satisfactoriamente gracias a un diseño adecuado que

permite filtrar y generar preguntas relevantes de forma optimizada y con una implementación que favorece la personalización del aprendizaje. El **OBJ-5** se centraba en generar problemas prácticos de la asignatura. En esta parte el diseño ha sido fundamental, ya que se abordó mediante una colección diferente de datos, por lo que implicó una lógica distinta en el diseño e implementación. Aun así, se consiguió una solución funcional, adaptable a las necesidades del estudiante y con técnicas de programación reutilizables. Finalmente, para abordar el **OBJ-6** se implementaron dashboards analíticos orientados a la personalización y seguimiento del progreso del estudiante en la asignatura. En esta funcionalidad, el diseño ha sido un aspecto clave, ya que permite definir un enfoque adecuado para el análisis de los aspectos más relevantes del aprendizaje. Gracias a ello, es posible plasmar de forma clara el seguimiento del progreso del alumno y ofrecer una experiencia de aprendizaje más individualizada y eficiente.

En resumen, se concluye que todos los objetivos han sido logrados con éxito. Asimismo, destacar que, tal y como se refleja en el capítulo de aceptación, se puede evidenciar que la consecución de todos ellos se ha obtenido también con una generación de soluciones de calidad.

- **Utilidad para el futuro:** desde el principio, el tema propuesto para este TFG busca ser de utilidad social y ayudar al alumnado en el ámbito de la enseñanza. Este asistente de aprendizaje es una propuesta innovadora, que aunque se haya limitado a la asignatura de Sistemas de Bases de Datos, puede extenderse a cualquier tipo de ámbito, nivel y docencia, tanto en universidades como en institutos, y por tanto, expandiéndose a más público y tipos de estudiantes. Para ello, la asignatura en cuestión debería tener una serie de objetivos, historias y criterios claros, y entonces, solo habría que actualizar el material proporcionado en el que basar el asistente. Sin embargo, no tiene porqué destinarse únicamente a los alumnos, ya que puede servir a los profesores para poder evaluar si el material impartido es suficiente para que los estudiantes alcancen los conocimientos requeridos o en cambio, necesitan de una mejora; o para ahorrar tiempo a la hora de crear nuevos ejercicios o respondiendo dudas de los estudiantes. Por último, comentar que podría integrarse en diferentes plataformas educativas, evolucionar hacia un tutor virtual totalmente personalizado que además de recursos proporcione métodos de aprendizaje o dar un soporte multilingüe. Además, podría derivar también en una detección temprana de dificultades de aprendizaje. En lo relativo a la IA, se podría incluir un módulo de *feedback* por parte del profesorado para recoger datos relacionados y métodos de valoración, para así aumentar la precisión, relevancia y calidad de las respuestas del asistente. Esto podría desembocar en la creación de algún modelo o métrica a nivel global que permitiera la evaluación de las respuestas obtenidas por IA Generativa, como la forma en la que medir la adecuación, coherencia o creatividad, es decir, aquellos valores más subjetivos y humanos.
- **Desarrollo del proyecto/metodología de trabajo:** en cuanto al uso de la metodología, ASAP ha sido fundamental para el desarrollo del proyecto. A raíz a su organización, ha sido posible especificar de manera precisa, detallada y limpia los diferentes paquetes a abordar en cada sprint, además de mantener un ritmo de trabajo constante. Además, gracias a una comunicación también permanente, ha sido posible mantener el estado del proyecto siempre operativo y en una línea investigadora.

### 10.1.2. Perspectiva personal

Desde una perspectiva más personal, son varios los puntos que me gustaría tratar.

En primer lugar destacar que desarrollar el TFG junto a la beca de colaboración, me ha permitido conocer cómo sería un proyecto de investigación de una mejor manera.

La labor de investigación me ha resultado más compleja de lo esperado, aunque a su vez fascinante. Las primeras ideas a tratar en el TFG fueron propuestas en julio del año pasado, aunque no fue hasta febrero cuando empezó mi trabajo efectivo. Al principio tenía mucha confusión, ya que apenas había información relacionada con ejemplos de RAG y mucho menos un asistente orientado a la educación universitaria. No obstante, conforme avanzaba e investigaba en profundidad, aprendía cómo realmente se encuentra la información relacionada y desde un punto de vista científico. Es decir, iba adquiriendo la habilidad para buscar los distintos artículos publicados por la comunidad científica de manera óptima. Sin embargo, me encontré con muchos casos donde inicialmente la información proporcionada parecía relacionarse, contribuir y ser útil al proyecto, pero finalmente no concretaba lo que necesitaba. Aunque podría interpretarse como una pérdida de tiempo esa fase de investigación, no fue en absoluto así, ya que me permitió aprender sobre otros aspectos valiosos.

Por otra parte, desde que se propuso el tema hasta su culminación, en tan solo un período de un año, se ha publicado una gran cantidad de material, incluyendo artículos, ejemplos y otros recursos, algunos relevantes y otros no tanto. Esto derivó en varios obstáculos durante el desarrollo del proyecto, ya que lo que había investigado hasta el momento no siempre respaldaba mis ideas iniciales; o al contrario, la información nueva encontrada era más adecuada, sencilla y relevante para el proyecto, por lo que a veces implicó realizar una reimplementación del proyecto. Por tanto, esa incertidumbre respecto a si los avances realizados hasta el momento eran los más óptimos o los definitivos fue algo desesperante, frustrante y agotador, ya que impedía dar por concluido cualquier parte del trabajo.

Sin embargo, esta situación contribuyó a que comprendiera mejor los límites, alcance y tecnologías del proyecto, además de a tomar decisiones claras.

Asimismo, el TFG también me ha ayudado a conocer en profundidad tanto la técnica RAG como Streamlit y diversos métodos de optimización, formas de realizar análisis de datos y otro conocimiento surgido durante los diversos artículos visualizados.

Han sido numerosos los desafíos, retos y problemas los que he afrontado durante el desarrollo, tanto externos como internos, que en ocasiones me obligaron a modificar o rehacer trabajo, incluso cuestiones que no dependían del proyecto. Sin embargo, todos ello se han ido solventando, esquivando y resolviendo a medida que se progresaba en el proyecto.

En definitiva, aunque el proceso tuvo momentos difíciles en los que he podido sufrir, finalmente ha sido muy gratificante apreciar que todo ha evolucionado de manera satisfactoria y se ha obtenido un producto de calidad y exitoso.

Por otra parte, resaltar que este TFG ha exigido conocer diversos aspectos. Además, he visto reflejado todo lo aprendido a lo largo de la carrera, especialmente, la parte de análisis, diseño y programación, ya que he podido asentar y mejorar los conocimientos adquiridos previos. Por otro lado, la metodología empleada ASAP me ha ayudado a saber qué hacer y cómo plantear el proyecto desde un inicio de forma efectiva, facilitando la labor de estructuración y planificación y además, con una comunicación activa.

Comentar que desde el principio tenía claro que quería trabajar en algo innovador, con impacto social, que ayudara a las personas en su día y que además les fuera útil. Para ello, me

inspiré en ChatGPT, ya que veía que muchas personas lo usaba diariamente, pero noté que este no permitía la subida de documentos para limitar el conjunto de datos de la respuesta. Por tanto, vi una oportunidad para aportar algo novedoso orientado a la educación y sus desafíos existentes como la falta de personalización. Cabe destacar que, aunque ChatGPT luego (una vez iniciado el proyecto) incorporó la subida de documentos, su funcionamiento no se basa en RAG, sino que simplemente añade la información a los prompts de forma masiva, por lo que este TFG sigue siendo innovador y necesario.

Resaltar que lo largo de proyecto y a medida que pasaba el tiempo, he podido asimilar que paralelamente pueden existir personas trabajando en ideas similares a las tuyas [284], y más en un campo como este. Por ello, el mundo de la investigación en informática es más exigente de lo que pensaba, ya que lo que se plantea o sugiere puede que se esté desarrollando en el mismo momento en otros lugares. Esto me hizo valorar aún más el esfuerzo de la comunidad científica y de que sus diferentes contribuciones no sean en vano.

Concluyendo sobre el TFG, considero que ha sido alcanzando con éxito, donde además prioricé un resultado final con calidad, tanto de memoria como del producto, que las horas dedicadas. No obstante, comentar que he tenido que limitar el alcance, pero creo que es un buen punto de partida para futuros trabajos con bastante potencial. Sin embargo, conforme avanzaba en su desarrollo conocía más acerca el tema propuesto y, una vez terminado el proyecto he de asumir que me entristece no haber tenido los conocimientos obtenidos sobre RAG desde el comienzo del proceso, ya que podría haber derivado en un proyecto con mucha más amplitud y relevancia científica. Aun así, destacar que estoy contenta con los resultados obtenidos y espero que pueda ser de utilidad en el futuro.

Finalmente, me gustaría terminar resaltando que este proyecto no es el final, si no es el principio de muchas más investigaciones resultantes. Como diría el filósofo Séneca: "*No es un final, es simplemente un nuevo comienzo*".

## 10.2. Trabajo futuro

En este proyecto se ha desarrollado un asistente de aprendizaje basado en RAG y en relación a una asignatura dentro del marco eXtreme Learning. Aunque se haya alcanzados los distintos objetivos planteados, aun son múltiples las líneas de investigación futuras que puede tener este proyecto. A continuación, se presentan diferentes opciones:

1. Añadir complejidad a la actividad de problemas o test, definiendo la complejidad en función de los criterios de aceptación. En general, problemas o test aún más detallados, donde permita al usuario introducir a bajo nivel lo que desea trabajar.
2. En cada problema, insertar la resolución al enunciado.
3. Incluir un módulo que permita la evaluación o corrección de los problemas y posteriormente, también reflejarlo para realizar el seguimiento del progreso.
4. Añadir sugerencias en la actividad de ver progreso en función del rendimiento del alumno (también con IA Generativa, y por ejemplo, en función de los datos obtenidos, indicar aquello en lo que necesite profundizar más).
5. Gestionar los usuarios, así como sus respuestas y permitir al profesorado ver las distintas preguntas y problemas generadas por el asistente.

6. Incluir un módulo de *feedback* para el profesorado para recoger datos relacionados y métodos de valoración, para así aumentar la precisión, relevancia y calidad de las respuestas del asistente.
7. En base al punto anterior, crear un modelo de IA o métricas para la evaluación de respuestas obtenidas a partir de modelos de IA Generativa, para medir la adecuación, coherencia o creatividad, es decir, aquellos valores más subjetivos y humanos.
8. Crear una aplicación para Sistemas de Base de Datos donde se incluya el asistente, además de otros TFG relacionados, con mayor despliegue y mejor interfaz.
9. Permitir una carga dinámica del material en el que fundamentar el asistente y, por tanto, el RAG. Para conseguir una personalización, será clave el planteamiento de los materiales en objetivos, historias y criterios de aprendizaje.
10. Cambiar el formato de salida del test a XMLMoodle para guardar las preguntas y más tarde, incorporarlas a la plataforma educativa Moodle.
11. Integrar el asistente con plataformas educativas, como puede ser Moodle, Teams o Google Classroom, para dar soporte directo a los estudiantes dentro de su entorno habitual de estudio.
12. Crear un tutor virtual totalmente personalizado que además de recursos proporcione métodos de aprendizaje, de un soporte multilingüe y cree un mecanismo de recompensa o de juegos para motivar al alumno.
13. Implementar un flujo ETL más amplio además de técnicas de recuperación más avanzadas para seguir perfeccionando tanto la fragmentación como la búsqueda semántica y calidad de la información recuperada.
14. Utilizar TruLens junto a RAGAS para medir la calidad del LLM.
15. Tratar las vulnerabilidades y fuga de información debido al uso de LLMs externos y el historial de session. Sobre todo, tener en cuenta los datos sensibles.
16. Tratar vulnerabilidad de ataque de inyección de prompts.
17. Ampliar el sistema en otros niveles educativos, instituciones o ámbitos y luego evaluar el impacto real del asistente en el rendimiento académico a través de estudios de muestras.
18. Analizar las respuestas del alumnado para obtener una detección temprana de dificultades de aprendizaje.
19. Incorporar capacidades de aprendizaje adaptativo que ajusten dinámicamente el contenido o la dificultad en función del progreso y estilo de aprendizaje del estudiante.
20. Adaptar el asistente para entornos accesibles, por ejemplo, dando soporte a estudiantes con discapacidad visual, auditiva o cognitiva.

*"Afortunadamente, la imaginación no está limitada y el ser humano tiende a inventar lo que puede imaginar".*



# Bibliografía

- [1] 4Geeks Academy. *IA Generativa vs LLM: Entendiendo las Diferencias y Sinergias*. Accedido el 14 de mayo de 2025. 2024. URL: <https://4geeks.com/es/lesson/ia-generativa-vs-llm>.
- [2] Chaymae El Aattabi. *Evaluación de Pipelines RAG. Potenciando la IA con la integración de conocimiento externo*. Accedido el 19 de abril de 2025. 2024. URL: <https://www.hiberus.com/crecemos-contigo/evaluacion-de-pipelines-rag/>.
- [3] Taye Abodunrin, Oguntade Ayo y George Christopher. «Natural Language Processing: An Overview of the State-of-the-Art Techniques». En: *ResearchGate* (2025). Accedido el 29 de mayo de 2025. URL: [https://www.researchgate.net/publication/391441211\\_Natural\\_Language\\_Processing\\_An\\_Overview\\_of\\_the\\_State-of-the-Art\\_Techniques](https://www.researchgate.net/publication/391441211_Natural_Language_Processing_An_Overview_of_the_State-of-the-Art_Techniques).
- [4] Khan Academy. *Khanmigo AI assistant*. 2023. URL: <https://www.khanmigo.ai/>.
- [5] Adaptical. *Mathew - Asistente educativo con IA*. Accedido el 12 de mayo de 2025. 2025. URL: <https://mathew.ai/>.
- [6] Adaptical. *¿Para quién es Mathew?* Accedido el 12 de mayo de 2025. 2025. URL: <https://mathew.ai/#para-quien-es>.
- [7] Jesús Aguado. *Retrieval Augmented Generation: ¿Qué es el RAG?* Accedido el 19 de mayo de 2025. 2024. URL: <https://blog.damavis.com/retrieval-augmented-generation-que-es-el-rag/>.
- [8] Delatorre AI. *Los transformers en la inteligencia artificial: una explicación sencilla*. Accedido el 29 de mayo de 2025. 2025. URL: <https://delatorre.ai/los-transformers-en-la-inteligencia-artificial-una-explicacion-sencilla/>.
- [9] Perplexity AI. *How Does Perplexity Work?* 2025. URL: <https://www.perplexity.ai/es/hub/faq/how-does-perplexity-work>.
- [10] Perplexity AI. *Perplexity AI*. 2024. URL: <https://www.perplexity.ai/>.
- [11] Phospho AI. *How AI-powered search works*. 2024. URL: <https://blog.phospho.ai/how-does-ai-powered-search-work-explaining-rag-llm-and-perplexity/>.
- [12] Airbyte. *5 Chunking Strategies For RAG Applications*. Accedido el 16 de mayo de 2025. 2024. URL: <https://airbyte.com/data-engineering-resources/chunk-text-for-rag>.
- [13] C. Akgun S. Greenhow. «Artificial intelligence in education: Addressing ethical challenges in K-12 classrooms». En: *PMC* (2022). URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8455229/>.

- [14] J. Alammar. *The Illustrated Transformer*. Accedido el 29 de mayo de 2025. 2025. URL: <http://jalammar.github.io/illustrated-transformer/>.
- [15] Jay Alammar. *The Illustrated Transformer*. Accedido el 29 de mayo de 2025. 2018. URL: <http://jalammar.github.io/illustrated-transformer/>.
- [16] Marc Alier, María José Casañ y David Aguilar Filvà. «Aplicaciones de aprendizaje inteligente: Aprovechamiento de los LLM para una tecnología educativa contextualizada y ética». En: *Actas de TEEM 2023. TEEM 2023. Apuntes de las clases sobre tecnología educativa*. Ed. por Jorge A. de C. Gonçalves, João Luís S. de M. Lima, João Pedro Coelho, Francisco José García-Peña y Alicia García-Holgado. Singapur: Springer, 2024. DOI: 10.1007/978-981-97-1814-6\_18. URL: [https://doi.org/10.1007/978-981-97-1814-6\\_18](https://doi.org/10.1007/978-981-97-1814-6_18).
- [17] Knewton Alta. *Knewton Support Center*. Accedido el 12 de mayo de 2025. 2024. URL: <https://support.knewton.com/s/>.
- [18] Amazon Web Services. *¿Qué es Retrieval-Augmented Generation (RAG)?* Accedido el 15 de mayo de 2025. 2024. URL: <https://aws.amazon.com/es/what-is/retrieval-augmented-generation/>.
- [19] Amazon Web Services. *¿Qué es un LLM (modelo de lenguaje de gran tamaño)?* 2025. URL: <https://aws.amazon.com/es/what-is/large-language-model/>.
- [20] Amazon Web Services. *¿Qué es una red neuronal?* Accedido el 28 de mayo de 2025. 2023. URL: <https://aws.amazon.com/es/what-is/neural-network/>.
- [21] Amazon Web Services. *¿Qué son los transformadores en la inteligencia artificial?* Accedido el 29 de mayo de 2025. 2025. URL: <https://aws.amazon.com/es/what-is/transformers-in-artificial-intelligence/>.
- [22] Aprende Machine Learning. *Aprendizaje profundo: una guía rápida.* 2019. URL: <https://www.aprendemachinelearning.com/aprendizaje-profundo-una-guia-rapida/>.
- [23] Aprende Machine Learning. *¿Cómo funcionan los Transformers? en Español.* Accedido el 29 de mayo de 2025. 2022. URL: <https://www.aprendemachinelearning.com/como-funcionan-los-transformers-espanol-nlp-gpt-bert/>.
- [24] Carlos Arana. *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales.* Serie Documentos de Trabajo 797. Accedido el 28 de mayo de 2025. Universidad del CEMA, 2021. URL: <https://www.econstor.eu/bitstream/10419/238422/1/797.pdf>.
- [25] Aïcha. *¿Cómo puede mejorar sus modelos de PNL utilizando servicios de anotación de textos?* 2024. URL: <https://es.innovatiana.com/post/improve-nlp-with-ai-services>.
- [26] Diego B. *Los mejores modelos de lenguaje grandes (LLM) para 2025 y cómo elegir el adecuado para tu sitio.* Accedido el 29 de mayo de 2025. 2025. URL: <https://www.hostinger.com/es/tutoriales/modelos-grandes-de-lenguaje-llm>.
- [27] Dzmitry Bahdanau, Kyunghyun Cho y Yoshua Bengio. «Neural Machine Translation by Jointly Learning to Align and Translate». En: *arXiv preprint arXiv:1409.0473* (2014). URL: <https://arxiv.org/abs/1409.0473>.

- 
- [28] R. S. Baker y P. S. Inventado. «Educational data mining and learning analytics». En: *Learning Analytics: From Research to Practice*. Springer, 2014, págs. 61-75.
  - [29] Satanjeev Banerjee y Alon Lavie. «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments». En: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005, págs. 65-72. URL: <https://www.aclweb.org/anthology/W05-0909.pdf>.
  - [30] Patent Attorney Bao Tran. «DMCA and Fair Use in Education: What Instructors Must Understand». En: *PatentPC* (2025). URL: <https://patentpc.com/blog/dmca-and-fair-use-in-education-what-instructors-must-understand>.
  - [31] Luis Rodrigo Barba Guaman. «Uso de técnicas deep learning para reconocimiento de objetos en imágenes». Tesis de mtría. Universidad Politécnica de Madrid, 2021. URL: [https://oa.upm.es/67969/1/LUIS\\_RODRIGO\\_BARBA\\_GUAMAN\\_01.pdf](https://oa.upm.es/67969/1/LUIS_RODRIGO_BARBA_GUAMAN_01.pdf).
  - [32] Kent Beck et al. *Manifiesto por el Desarrollo Ágil de Software*. 2001. URL: <http://agilemanifesto.org/iso/es/manifiesto.html>.
  - [33] Beijing Academy of Artificial Intelligence. *BAAI/bge-m3: Multi-Functionality, Multi-Linguality, and Multi-Granularity Embedding Model*. Accedido el 2020 de mayo de 2025. 2024. URL: <https://huggingface.co/BAAI/bge-m3>.
  - [34] Néstor Camilo Beltrán Beltrán y Edda Camila Rodríguez Mojica. «Procesamiento del lenguaje natural (PLN) - GPT-3.: Aplicación en la Ingeniería de Software». En: *Tecnología Investigación y Academia* 8.1 (2021), 18–37. URL: <https://revistas.udistrital.edu.co/index.php/tia/article/view/17323>.
  - [35] E. M. Bender, T. Gebru, A. McMillan-Major y S. Shmitchell. «On the dangers of stochastic parrots: Can language models be too big?» En: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, págs. 610-623. DOI: 10.1145/3442188.3445922.
  - [36] Yoshua Bengio, Aaron Courville y Pascal Vincent. «Representation Learning: A Review and New Perspectives». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013). Accedido el 28 de mayo de 2025, págs. 1798-1828. DOI: 10.1109/TPAMI.2013.50. URL: <https://arxiv.org/abs/1206.5538>.
  - [37] Hans-Dieter Billhardt, Antonio Fernández-Gil y Sascha Ossowski. *Inteligencia artificial: Ejercicios resueltos*. Accedido el 10 de junio de 2025. Madrid: Editorial Universitaria Ramón Areces, 2015.
  - [38] D. F. Bjorklund y K. B. Causey. *Children's Thinking: Cognitive Development and Individual Differences*. Cengage Learning, 2017.
  - [39] Jordi SOLÉ BLANCH. «El cambio educativo ante la innovación tecnológica, la pedagogía de las competencias y el discurso de la educación emocional: una mirada crítica». En: *Torrossa* (2021). Consultado el 8 de mayo de 2025. URL: <https://www.torrossa.com/en/resources/an/4608257>.
  - [40] BLE - Bridging Learning Experiences. *Personae*. Consultado el 8 de mayo de 2025. 2020. URL: <http://psyed.edu.es/ble2/es/personae/>.

- [41] Perkins School for the Blind. *Socratic App Accessibility Review*. 2023. URL: <https://www.perkins.org/resource/socratic-app-accessibility-review/>.
- [42] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli y et al. «On the Opportunities and Risks of Foundation Models». En: *arXiv preprint arXiv:2108.07258* (2021). URL: <https://arxiv.org/abs/2108.07258>.
- [43] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli y et al. «On the Opportunities and Risks of Foundation Models». En: *arXiv preprint arXiv:2108.07258* (2021). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/2108.07258>.
- [44] Brainly. *Brainly*. 2024. URL: <https://brainly.com/>.
- [45] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal y D. Amodei. «Language models are few-shot learners». En: *Proceedings of NeurIPS 2020*. 2020. URL: <https://arxiv.org/abs/2005.14165>.
- [46] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al. «Language models are few-shot learners». En: *Advances in Neural Information Processing Systems* 33 (2020). Accedido el 19 de mayo de 2025, págs. 1877-1901. URL: <https://arxiv.org/abs/2005.14165>.
- [47] Tom B. Brown et al. «Language Models are Few-Shot Learners». En: *arXiv preprint arXiv:2005.14165* (2020). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/2005.14165>.
- [48] E. Brynjolfsson y A. McAfee. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W.W. Norton & Company, 2014.
- [49] Erik Brynjolfsson y Andrew McAfee. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W.W. Norton & Company, 2014.
- [50] Gülsüm Budakoglu y Hakan Emekci. «Unveiling the Power of Large Language Models: A Comparative Study of Retrieval-Augmented Generation, Fine-Tuning and Their Synergistic Fusion for Enhanced Performance». En: *IEEE Access PP* (ene. de 2025), págs. 1-1. DOI: [10.1109/ACCESS.2025.3542334](https://doi.org/10.1109/ACCESS.2025.3542334).
- [51] José Bernardo Carrasco y José F. Javaloyes Juan J. y Calderero. *Cómo personalizar la educación: Una solución de futuro*. Primera. Vol. 108. Educación Hoy Estudios. Accedido el 18 de junio de 2025. Madrid, España: Narcea Ediciones, 2007, pág. 256. ISBN: 9788427715592. URL: <https://books.google.es/books?hl=es&lr=&id=9mlCEQAAQBAJ&oi=fnd&pg=PA95>.
- [52] Marcos Casado Herrero. «OSINT Infohound – Síntesis de datos de fuentes abiertas por medio de modelos de lenguaje de gran tamaño (LLM)». Accedido el 29 de mayo de 2025. Trabajo de Fin de Máster. Universitat Oberta de Catalunya, 2024. URL: <https://openaccess.uoc.edu/handle/10609/149544>.
- [53] Yone Castro. *La personalización del aprendizaje escolar*. Consultado el 8 de mayo de 2025. 2024. URL: <https://prodis360.org/la-personalizacion-del-aprendizaje-escolar/>.
- [54] CEI. *Las Redes Neuronales en Inteligencia Artificial*. Accedido el 28 de mayo de 2025. 2021. URL: <https://cei.es/redes-neuronales/>.

- [55] Hao Chen, Hao Fang, Xiaodong Liu et al. «Dense X Retrieval: What Retrieval Granularity Should We Use?». En: *arXiv preprint arXiv:2312.06648* (2023). URL: <https://arxiv.org/abs/2312.06648>.
- [56] L. Chen, H. Xie y Z. Cheng. «AI in Education: A Review». En: *Journal of Educational Technology & Society* 24.1 (2021), págs. 88-103.
- [57] Xieling Chen, Haoran Xie, Di Zou y Gwo-Jen Hwang. «Application and theory gaps during the rise of Artificial Intelligence in Education». En: *Computers and Education: Artificial Intelligence* 1 (ene. de 2020), pág. 100002. DOI: 10.1016/j.caeari.2020.100002.
- [58] Juan Diego Chiluisa Gallardo y Gustavo Rodríguez Bárcenas. «Optimización de la planificación del silabo en la Universidad Técnica de Cotopaxi mediante Inteligencia Artificial Generativa: Un enfoque personalizado basado en LLAMA 2». En: *Tesla Revista Científica* 4.2 (2024). Accedido el 18 de junio de 2025, e464. DOI: 10.55204/trc.v4i2.e464. URL: <https://tesla.puertomaderoeditorial.com.ar/index.php/tesla/article/view/464/467>.
- [59] Merwan Chinta. *Large Language Models (LLMs) — Top View*. 2024. URL: <https://medium.com/codenx/large-language-models-llms-top-view-153581d2b9f3>.
- [60] François Chollet. *Deep Learning with Python*. Accedido el 10 de junio de 2025. Shelter Island, NY: Manning Publications, 2017.
- [61] Google Cloud. *Casos de uso de la IA generativa*. Accedido el 14 de mayo de 2025. n.d. URL: <https://cloud.google.com/use-cases/generative-ai?hl=es>.
- [62] Cloudflare. ¿Qué es una base de datos vectorial?. Accedido el 16 de mayo de 2025. 2024. URL: <https://www.cloudflare.com/es-es/learning/ai/what-is-vector-database/>.
- [63] Cloudflare. ¿Qué son las integraciones?. Accedido el 16 de mayo de 2025. 2024. URL: <https://www.cloudflare.com/es-es/learning/ai/what-are-embeddings/>.
- [64] Georgia Tech Polo Club. *The Transformer Visualizer*. Accedido el 29 de mayo de 2025. 2023. URL: <https://poloclub.github.io/transformer-explainer/>.
- [65] Clínica Universidad de Navarra. *Neurona*. Accedido el 28 de mayo de 2025. 2025. URL: <https://www.cun.es/diccionario-medico/terminos/neurona>.
- [66] C. E. Cobos-Gutiérrez. «Impacto de la Inteligencia Artificial en el Rendimiento Académico de Estudiantes de Secundaria: Un Estudio Correlacional». En: *Puriq* 6 (2024), e740. DOI: 10.37073/puriq.6.740. URL: <https://doi.org/10.37073/puriq.6.740>.
- [67] CodeAwake. *PostgreSQL como base de datos vectorial*. Accedido el 20 de mayo de 2025. 2024. URL: <https://codeawake.com/es/blog/postgresql-vector-database>.
- [68] CoderFaculty. *What is Retrieval-Augmented Generation (RAG)?* Accedido el 20 de mayo de 2025. 2024. URL: <https://www.coderfaculty.com/blog.php?id=rag>.
- [69] Colegio Oficial de Ingenieros de Valencia. *Guía Básica de Inteligencia Artificial*. Accedido el 29 de mayo de 2025. 2025. URL: <https://multimedia2.coev.com/pdfs/Guia-Basica-IA.pdf>.
- [70] César Coll y Bernat Albaigés, eds. *L'estat de l'educació a Catalunya. Anuari 2020*. Barcelona: Fundació Jaume Bofill, 2021. ISBN: 978-84-121371-3-2. URL: <https://fundaciobofill.cat/publicacions/anuari-2020>.

- [71] College of Education Office of Communications. «AI in Schools: Pros and Cons». En: *University of Illinois College of Education* (2024). URL: <https://education.illinois.edu/about/news-events/news/article/2024/10/24/ai-in-schools--pros-and-cons>.
- [72] GTPE Communications. *Meet Jill Watson: Georgia Tech's First AI Teaching Assistant*. 2021. URL: <https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant>.
- [73] Khan Academy Community. *Khanmigo y GPT-4*. 2023. URL: <https://support.khanacademy.org/hc/en-us/community/posts/15789901226509>.
- [74] B. Jack Copeland. *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle to Build the Modern Computer*. Oxford: Oxford University Press, 2004.
- [75] Alberto Cortés Hernández, Christian Antonio Hernández Hernández, Alejandra Berenice García Torres y Mauricio Mata Quezadas. «La Inteligencia Artificial Generativa como un Asistente Estratégico en la Era del Aprendizaje Digital». En: *Ciencia Latina Revista Científica Multidisciplinaria* 8.4 (2024), págs. 2159-2178. DOI: 10.37811/cl\_rcm.v8i4.12456. URL: <https://ciencialatina.org/index.php/cienciala/article/view/12456>.
- [76] Nicolás Arrioja Landa Cosio. *Métricas en regresión*. Accedido el 28 de mayo de 2025. Dic. de 2021. URL: <https://medium.com/@nicolasarrioja/m%C3%A9tricas-en-regresi%C3%B3n-5e5d4259430b>.
- [77] Coursera Staff. *3 tipos de machine learning que debes conocer*. Accedido el 13 de mayo de 2025. 2023. URL: <https://www.coursera.org/mx/articles/types-of-machine-learning>.
- [78] Alberto Céspedes. *Deep Learning - Extracting information from Imagery*. 2022. URL: <https://www.idee.es/resources/presentaciones/JIIDE22/Presentaciones/T06.pdf>.
- [79] DataCamp. *ChromaDB Tutorial: Step-by-Step Guide*. Accedido el 20 de mayo de 2025. 2024. URL: <https://www.datacamp.com/es/tutorial/chromadb-tutorial-step-by-step-guide>.
- [80] DataCamp. *Guía introductoria para el ajuste preciso de los LLM*. Accedido el 15 de mayo de 2025. 2024. URL: <https://www.datacamp.com/es/tutorial/fine-tuning-large-language-models>.
- [81] DataCamp. *Introducción a los autocodificadores: desde lo básico hasta aplicaciones avanzadas en PyTorch*. 2024. URL: <https://www.datacamp.com/es/tutorial/introduction-to-autoencoders>.
- [82] DataCamp. *Las 5 mejores bases de datos vectoriales: una lista con ejemplos*. Accedido el 16 de mayo de 2025. 2023. URL: <https://www.datacamp.com/es/blog/the-top-5-vector-databases>.
- [83] DataCamp. *Tipos de bases de datos: resumen general*. Accedido el 20 de mayo de 2025. 2024. URL: <https://www.datacamp.com/es/blog/types-of-databases-overview>.
- [84] DataCamp. *¿Qué es el aprendizaje automático? Una guía para principiantes*. 2024. URL: <https://www.datacamp.com/es/blog/what-is-machine-learning>.

- [85] DataCamp. *¿Qué es la Generación Aumentada de Recuperación (GRA)?* Accedido el 15 de mayo de 2025. 2024. URL: <https://www.datacamp.com/es/blog/what-is-retrieval-augmented-generation-rag>.
- [86] DataCamp. *¿Qué es la ingeniería de prompts? Una guía detallada para 2024.* Accedido el 15 de mayo de 2025. 2024. URL: <https://www.datacamp.com/es/blog/what-is-prompt-engineering-the-future-of-ai-communication>.
- [87] Datos.gob.es. *10 librerías populares de procesamiento del lenguaje natural.* Accedido el 2020 de mayo de 2025. 2022. URL: <https://datos.gob.es/es/blog/10-librerias-populares-de-procesamiento-del-lenguaje-natural>.
- [88] datos.gob.es. *RAG - Retrieval Augmented Generation: La llave que abre la puerta de la precisión a los modelos del lenguaje.* Accedido el 15 de mayo de 2025. 2024. URL: <https://datos.gob.es/es/blog/rag-retrieval-augmented-generation-la-llave-que-abre-la-puerta-de-la-precision-los-modelos-del>.
- [89] datos.gob.es. *¿Qué es el aprendizaje automático o Machine Learning?* 2024. URL: <https://datos.gob.es/es/blog/que-es-el-aprendizaje-automatico-o-machine-learning>.
- [90] Luis María Delgado Martínez. «Aprendizaje centrado en el estudiante, hacia un nuevo arquetipo docente». En: *Enseñanza y Teaching: Revista Interuniversitaria de Didáctica* 37.1 (2019), 139–154. DOI: 10.14201/et2019371139154. URL: <https://revistas.usal.es/tres/index.php/0212-5374/article/view/et2019371139154>.
- [91] Alberto Tomás Delso Vicente, Marisol Carvajal Camperos y Daniel Ángel Corral De La Mata. «La evolución del procesamiento del lenguaje natural y su influencia en la inteligencia artificial: Una revisión y líneas de investigación futura». En: *European Public and Social Innovation Review* 10 (2024), 1–23. DOI: 10.31637/epsir-2025-782. URL: <https://epsir.net/index.php/epsir/article/view/782>.
- [92] Chang M. W. Lee K. & Toutanova-K. Devlin J. *BERT: Pre-training of deep bidirectional transformers for language understanding.* 2018. URL: <https://arxiv.org/abs/1810.04805>.
- [93] J. Devlin, M. W. Chang, K. Lee y K. Toutanova. «BERT: Pre-training of deep bidirectional transformers for language understanding». En: *Proceedings of NAACL 2019.* 2019. URL: <https://arxiv.org/abs/1810.04805>.
- [94] Dezzai. *Entender la ambigüedad lingüística y el impacto de Chat GPT.* Accedido el 16 de mayo de 2025. 2023. URL: <https://dezzai.com/es/blog/impacto-chat-gpt-ambiguedad-lingistica/>.
- [95] Karthikeyan Dhanakotti. *RAGAS for RAG in LLMs: A Comprehensive Guide to Evaluation Metrics.* Accedido el 19 de abril de 2025. 2024. URL: <https://dkaarthick.medium.com/ragas-for-rag-in-llms-a-comprehensive-guide-to-evaluation-metrics-3aca142d6e38>.
- [96] Diego B. *Prompt Engineering: Qué es y cómo usarlo.* 2025. URL: <https://www.hostinger.com/es/tutoriales/prompt-engineering>.
- [97] Yuchen Ding, Yuxuan Wang, Zhen Li y Wei Zhang. «A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models». En: *arXiv preprint arXiv:2405.06211* (2024). URL: <https://arxiv.org/abs/2405.06211>.

- [98] Pedro Domingos. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, 2015.
- [99] Duolingo. *Duolingo Max*. 2024. URL: <https://www.duolingo.com/max>.
- [100] Gerardo Echeita Sarriónandia y Ana Belén Domínguez Gutiérrez. «Educación inclusiva. Argumento, caminos y encrucijadas». En: *Aula* 17 (2011), 23–35. DOI: 10.14201/8393. URL: <https://revistas.usal.es/tres/index.php/0214-3402/article/view/8393>.
- [101] Project ECHO. *Transforming Classroom Education*. 2023. URL: <https://projectecho.unm.edu/story/khammigo-echo-transforming-classroom-education-by-training-teachers-to-use-ai/>.
- [102] Virtual Educa. *Nuevas tendencias en educación superior: aprendizaje personalizado, flexible y a lo largo de la vida*. 2024. URL: <https://virtualeduca.org/mediacenter/nuevas-tendencias-en-educacion-superior-aprendizaje-personalizado-flexible-y-a-lo-largo-de-la-vida/>.
- [103] Elastic. ¿Qué es la generación aumentada por recuperación (RAG)? Accedido el 19 de mayo de 2025. 2024. URL: <https://www.elastic.co/es/what-is/retrieval-augmented-generation>.
- [104] Simeon Emanuilov. *Retrieval-Augmented Generation (RAG): Limitations*. Accedido el 19 de mayo de 2025. 2024. URL: <https://medium.com/@simeon.emanuilov/retrieval-augmented-generation-rag-limitations-d0c641d8b627>.
- [105] Anna Engel Rocamora y Cristina Coll Salvador. *Entornos híbridos y personalización del aprendizaje*. Blog post en \*Aretio\* (Hypotheses). Accedido el 18 de junio de 2025. 2022. URL: <https://aretio.hypotheses.org/9974>.
- [106] Caroline Eppright. *What is Natural Language Processing (NLP)*? Accedido el 29 de mayo de 2025. 2021. URL: <https://www.oracle.com/hk/artificial-intelligence/what-is-natural-language-processing/>.
- [107] Equipo de Documentación de Ragas. *List of Available Metrics*. Accedido el 19 de abril de 2025. 2025. URL: [https://docs.ragas.io/en/stable/concepts/metrics/available\\_metrics/](https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/).
- [108] Shahul Es, Jithin James, Luis Espinosa-Anke y Steven Schockaert. «RAGAS: Automated Evaluation of Retrieval Augmented Generation». En: *arXiv preprint arXiv:2309.15217* (2023). Accedido el 19 de abril de 2025. URL: <https://arxiv.org/abs/2309.15217>.
- [109] Shahul Es, Jithin James, Luis Espinosa-Anke y Steven Schockaert. «RAGAS: Automated Evaluation of Retrieval Augmented Generation». En: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Accedido el 19 de abril de 2025. St. Julians, Malta: Association for Computational Linguistics, 2024, págs. 150-158. URL: <https://aclanthology.org/2024.eacl-demo.16/>.
- [110] Computing España. ¿Qué es el Machine Learning y cómo funciona? 2024. URL: <https://wwwcomputing.es/analytics/aprendizaje-automatico-machine-learning-que-es-como-funciona/>.
- [111] Ministerio de Educación y Formación Profesional de España. *Informe sobre Digitalización en la Educación*. 2022. URL: <https://www.educacionyfp.gob.es>.

- 
- [112] Hugging Face. *How to Generate Text: Text Generation with Transformers*. Accedido el 29 de mayo de 2025. 2021. URL: <https://huggingface.co/blog/how-to-generate>.
  - [113] Wiston Forero-Corba y Francisca Negre Bennasar. «Técnicas y aplicaciones del Machine Learning e inteligencia artificial en educación: una revisión sistemática». En: *RIED-Revista Iberoamericana de Educación a Distancia* 27.1 (2024), 209–253. DOI: 10.5944/ried.27.1.37491. URL: <https://revistas.uned.es/index.php/ried/article/view/37491>.
  - [114] Dolors Forteza, Laura Fuster y Francisca Moreno-Tallón. «Barreras para el Aprendizaje y la Participación en la Escuela del Alumnado con Dislexia: Voces de Familias». En: *Revista Internacional de Educación para la Justicia Social* 8.2 (2019), 113–130. DOI: 10.15366/riejs2019.8.2.006. URL: <https://revistas.uam.es/riejs/article/view/riejs2019.8.2.006>.
  - [115] Luyu Gao y Jamie Callan. «ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction». En: *arXiv preprint arXiv:2112.01488* (2021). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2112.01488>.
  - [116] Tianyu Gao, Shuyuan Lin, Deming Ye, Daniel Khashabi, Chitta Baral y Dan Roth. «LLM-as-a-Judge: Are We Ready for Automated Assessments in Open-Ended Tasks?» En: *Transactions of the Association for Computational Linguistics* (2024). Accedido el 11 de junio de 2025. URL: <https://arxiv.org/html/2411.15594v4>.
  - [117] Tianyu Gao, Caiming Xiong y Jamie Callan. «RAGRank: Augmenting Retrieval-Augmented Generation with Cross-Encoder Ranking». En: *arXiv preprint arXiv:2407.02485* (2023). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2407.02485>.
  - [118] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang y Haofen Wang. «Retrieval-Augmented Generation for Large Language Models: A Survey». En: *arXiv preprint arXiv:2312.10997* (2023). URL: <https://arxiv.org/abs/2312.10997>.
  - [119] Carlos Gil López. *Formación en IA generativa: llega la educación tecno-pedagógica y realmente personalizada*. Accedido el 14 de mayo de 2025. 2025. URL: <https://www.udima.es/formacion-ia-generativa-llega-educacion-tecnopedagogica-realmente-personalizada>.
  - [120] Glassdoor. *Sueldo: Gestor De Proyectos en España 2025*. Accedido el 28 de mayo de 2025. Feb. de 2025. URL: [https://www.glassdoor.es/Sueldos/gestor-de-proyectos-sueldo-SRCH\\_K00%2C19.htm](https://www.glassdoor.es/Sueldos/gestor-de-proyectos-sueldo-SRCH_K00%2C19.htm).
  - [121] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: <https://www.deeplearningbook.org>.
  - [122] Google. *Socratic*. 2023. URL: <https://socratic.org/>.
  - [123] Google Cloud. *¿Qué es la generación aumentada de recuperación (RAG)?* Accedido el 19 de mayo de 2025. 2024. URL: <https://cloud.google.com/use-cases/retrieval-augmented-generation?hl=es>.
  - [124] Google Cloud Skills Boost. *Arquitectura Encoder-Decoder*. 2024. URL: [https://www.cloudskillsboost.google/course\\_templates/543?locale=es](https://www.cloudskillsboost.google/course_templates/543?locale=es).

- [125] GOV.UK. «Copyright and Artificial Intelligence». En: *GOV.UK* (2025). URL: <https://www.gov.uk/government/consultations/copyright-and-artificial-intelligence/copyright-and-artificial-intelligence>.
- [126] Gradescope. *Gradescope*. 2024. URL: <https://www.gradescope.com/>.
- [127] M. Guettala, S. Bourekkache, O. Kazar y S. Harous. «Generative artificial intelligence in education: Advancing adaptive and personalized learning». En: *Acta Informatica Pragensia* 13.3 (2024). Accedido el 16 de junio de 2025, págs. 460-489. DOI: 10.18267/j.aip.235. URL: <https://doi.org/10.18267/j.aip.235>.
- [128] Y. Guo, T. Zhu, Z. Li y C. Ni. «Auto-Modal: Air-Quality Index Forecasting with Modal Decomposition Attention». En: *Sensors* 22.18 (2022). Accedido el 29 de mayo de 2025. DOI: 10.3390/s22186953. URL: <https://doi.org/10.3390/s22186953>.
- [129] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. Accedido el 10 de junio de 2025. Sebastopol, CA: O'Reilly Media, 2017.
- [130] J. Gómez, J. Rodríguez e I. García. «La inteligencia artificial generativa y su impacto en la docencia universitaria». En: *Journal of Technology and Science Education* 13.2 (2023), pág. 2680. URL: <https://www.jotse.org/index.php/jotse/article/view/2680>.
- [131] Red Hat. *¿Qué es la IA generativa?* Accedido el 14 de mayo de 2025. n.d. URL: <https://www.redhat.com/es/topics/ai/what-is-generative-ai>.
- [132] Haystack. *Tutorial: Creating Your First QA Pipeline with Retrieval-Augmentation*. Accedido el 20 de mayo de 2025. 2025. URL: [https://haystack.deepset.ai/tutorials/27\\_first\\_rag\\_pipeline](https://haystack.deepset.ai/tutorials/27_first_rag_pipeline).
- [133] J. J. Heckman. «Skill formation and the economics of investing in disadvantaged children». En: *Science* 312.5782 (2006), págs. 1900-1902.
- [134] Raúl Hermosa Cruz. «Aplicación de técnicas de aprendizaje automático para la clasificación de imágenes médicas». Accedido el 16 de mayo de 2025. Trabajo Fin de Máster. Universidad Politécnica de Madrid, 2023. URL: [https://oa.upm.es/83450/1/TFM\\_RAUL\\_HERMOSA\\_CRUZ.pdf](https://oa.upm.es/83450/1/TFM_RAUL_HERMOSA_CRUZ.pdf).
- [135] S. Hesaraki. *Long Short-Term Memory (LSTM)*. Accedido el 29 de mayo de 2025. 2023. URL: <https://medium.com/@saba99/long-short-term-memory-lstm-ffffc5eaebfdc>.
- [136] Julia Hirschberg y Christopher D. Manning. «Advances in Natural Language Processing». En: *Science* 349.6245 (2015). Accedido el 29 de mayo de 2025, págs. 261-266. DOI: 10.1126/science.aaa8685. URL: <https://www.science.org/doi/10.1126/science.aaa8685>.
- [137] Andrew Hodges. *Alan Turing: The Enigma*. Princeton University Press, 2014.
- [138] Wayne Holmes, Maya Bialik y Charles Fadel. *Artificial Intelligence in Education. Promise and Implications for Teaching and Learning*. Mar. de 2019. ISBN: 978-1794293700.

- 
- [139] Emam Hossain, Mohd Arafath Uddin Shariff, Mohammad Shahadat Hossain y Karl Andersson. «A Novel Deep Learning Approach to Predict Air Quality Index». En: *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. Ed. por M. Shamim Kaiser, Anirban Bandyopadhyay, Mufti Mahmud y Kanad Ray. Accedido el 29 de mayo de 2025. Singapore: Springer Singapore, 2021, págs. 367-381. ISBN: 978-981-33-4673-4. DOI: 10.1007/978-981-33-4673-4\_29. URL: [https://doi.org/10.1007/978-981-33-4673-4\\_29](https://doi.org/10.1007/978-981-33-4673-4_29).
  - [140] Pablo Huet. *Embeddings: Qué son y cómo transforman datos en información*. Accedido el 29 de mayo de 2025. 2024. URL: <https://openwebinars.net/blog/embeddings/>.
  - [141] Pablo Huet. *Qué son las redes neuronales y sus aplicaciones*. Accedido el 29 de mayo de 2025. 2023. URL: <https://openwebinars.net/blog/que-son-las-redes-neuronales-y-sus-aplicaciones/>.
  - [142] Hugging Face. *Modelos de Hugging Face*. Accedido el 20 de mayo de 2025. 2024. URL: <https://huggingface.co/models>.
  - [143] HulkApps. *Django vs React: Selección del mejor marco de trabajo para sus necesidades de desarrollo web*. Accedido el 19 de mayo de 2025. 2024. URL: <https://www.hulkapps.com/es/blogs/ecommerce-hub/django-vs-react-seleccion-del-mejor-marco-de-trabajo-para-sus-necesidades-de-desarrollo-web>.
  - [144] IAIED. *Jill Watson Showcase*. 2025. URL: <https://iaied.org/showcase/6hRMxzuPi8sYwCw6YiU8YS>.
  - [145] Ibero. *Problemas de salud mental impactan rendimiento académico de estudiantes*. 2023. URL: <https://ibero.mx/prensa/problemas-de-salud-mental-impactan-rendimiento-academico-de-estudiantes>.
  - [146] Ibertech. *¿Qué es Machine Learning y qué aplicaciones tiene en nuestro día a día?* Accedido el 28 de mayo de 2025. Mar. de 2018. URL: <https://www.ibertech.org/que-es-machine-learning-y-que-aplicaciones-tiene-en-nuestro-dia-a-dia-2>.
  - [147] IBM. *El modelo de redes neuronales*. Accedido el 28 de mayo de 2025. 2023. URL: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>.
  - [148] IBM. *IBM Watson Tutor Overview*. 2020. URL: <https://www.ibm.com/es-es/products/watsonx-ai>.
  - [149] IBM. *LlamaIndex RAG Tutorial*. Accedido el 2020 de mayo de 2025. 2025. URL: <https://www.ibm.com/think/tutorials/llamaindex-rag>.
  - [150] IBM. *Mecanismo de atención*. URL: <https://www.ibm.com/es-es/think/topics/attention-mechanism>.
  - [151] IBM. *Orquestación en soluciones de Generación Aumentada por Recuperación (RAG)*. Accedido el 2020 de mayo de 2025. 2024. URL: <https://www.ibm.com/architectures/papers/rag-cookbook/orchestration>.
  - [152] IBM. *Redes neuronales recurrentes (RNN)*. Accedido el 29 de mayo de 2025. 2025. URL: <https://www.ibm.com/es-es/think/topics/recurrent-neural-networks>.
  - [153] IBM. *What is a vector database?* Accedido el 5 de junio de 2025. 2024. URL: <https://www.ibm.com/think/topics/vector-database>.
  - [154] IBM. *¿Qué es el ajuste de modelos?* Accedido el 15 de mayo de 2025. 2025. URL: <https://www.ibm.com/es-es/think/topics/model-tuning>.

- [155] IBM. *¿Qué es el aprendizaje supervisado?* Accedido el 28 de mayo de 2025. 2023. URL: <https://www.ibm.com/es-es/topics/supervised-learning>.
- [156] IBM. *¿Qué es el prompt engineering?* Accedido el 15 de mayo de 2025. 2024. URL: <https://www.ibm.com/es-es/think/topics/prompt-engineering>.
- [157] IBM. *¿Qué es la inteligencia artificial generativa?* Accedido el 14 de mayo de 2025. URL: <https://www.ibm.com/es-es/think/topics/generative-ai>.
- [158] IBM. *¿Qué es la regularización?* 2023. URL: <https://www.ibm.com/es-es/think/topics/regularization>.
- [159] IBM. *¿Qué es un autocodificador?* 2023. URL: <https://www.ibm.com/es-es/think/topics/autoencoder>.
- [160] IBM. *¿Qué es un modelo encoder-decoder?* 2024. URL: <https://www.ibm.com/think/topics/encoder-decoder-model>.
- [161] IBM. *¿Qué son las redes neuronales?* Accedido el 28 de mayo de 2025. 2023. URL: <https://www.ibm.com/es-es/think/topics/neural-networks>.
- [162] IBM. *¿Qué son los grandes modelos de lenguaje (LLM)?* 2023. URL: <https://www.ibm.com/es-es/think/topics/large-language-models>.
- [163] IBM Research. *What is retrieval-augmented generation (RAG)?* Accedido el 15 de mayo de 2025. 2023. URL: <https://research.ibm.com/blog/retrieval-augmented-generation-RAG>.
- [164] IGNITE Serious Play. *Aplicaciones de la inteligencia artificial generativa en la educación.* Accedido el 14 de mayo de 2025. 2023. URL: <https://igniteseriousplay.com/aplicaciones-de-la-inteligencia-artificial-generativa-en-la-educacion/>.
- [165] Pinecone Systems Inc. *What is Vector Similarity?* Accedido el 19 de mayo de 2025. 2023. URL: <https://www.pinecone.io/learn/vector-similarity/>.
- [166] Innovaitors.ai. *Aplicaciones de IA generativa en educación.* Accedido el 14 de mayo de 2025. n.d. URL: <https://www.innovaitors.ai/aplicaciones-ia-generativa-educacion/>.
- [167] Innovatiana. *Mejora el procesamiento de lenguaje natural con servicios de IA.* Accedido el 29 de mayo de 2025. 2025. URL: <https://es.innovatiana.com/post/improve-nlp-with-ai-services>.
- [168] eLearning Innovation Center (UOC). *Personalización para una universidad más abierta a la diversidad.* 2023. URL: <https://blogs.uoc.edu/elearning-innovation-center/es/personalizacion-para-una-universidad-mas-abierta-a-la-diversidad/>.
- [169] Instituto de Ingeniería del Conocimiento. *Transformers en Procesamiento del Lenguaje Natural.* Accedido el 29 de mayo de 2025. 2025. URL: <https://www.iic.uam.es/innovacion/transformers-en-procesamiento-del-lenguaje-natural/>.
- [170] INTEF. *El desafío de la IA generativa en educación.* 2024. URL: <https://code.intef.es/noticias/desafio-de-la-ia-generativa-en-educacion/>.
- [171] Inteligencia Artificial 360. *Tasa de aprendizaje.* <https://inteligenciaartificial360.com/glosario/tasa-de-aprendizaje/>. Accedido el 14 de mayo de 2025. n.d.

- 
- [172] InterSystems. *Retrieval Augmented Generation (RAG): qué es y cómo evita los errores de la IA*. Accedido el 18 de junio de 2025. 2024. URL: <https://www.intersystems.com/es/recursos/retrieval-augmented-generation-rag/>.
  - [173] IONOS. *Prompt engineering: definición, ejemplos y buenas prácticas*. Accedido el 15 de mayo de 2025. 2025. URL: <https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/prompt-engineering/>.
  - [174] Ipsos. *Monitor Global de Educación*. Accedido: 01 de abril de 2025. 2024.
  - [175] Gautier Izacard y Edouard Grave. «Leveraging passage retrieval with generative models for open domain question answering». En: *arXiv preprint arXiv:2007.01282* (2021). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2007.01282>.
  - [176] Pablo Santoro Domingo-Laura Martínez Álvaro David Carabantes-Alarcón J. Ignacio Pi-chardo Oriol Borrás-Gené. *Documento académico sobre IA (UCM)*. <https://docta.ucm.es/rest/api/core/bitstreams/319d0a95-a184-4891-801a-cfca6009cb6e/content>. Accedido el 14 de mayo de 2025.
  - [177] Jarroba. *¿Qué es el Machine Learning?* Accedido el 13 de mayo de 2025. s.f. URL: <https://jarroba.com/que-es-el-machine-learning/>.
  - [178] Yifeng Jiang. *Vector Database and Storage. Is it true generative AI and RAG increase data storage by up to 10x?* Accedido el 18 de junio de 2025. 2024. URL: <https://uprush.medium.com/vector-database-and-storage-8fa8d07e77e4>.
  - [179] Antonio Jimeno-Yepes, Yao You, Jan Milczek, Sebastian Laverde y Renyu Li. «Financial Report Chunking for Effective Retrieval Augmented Generation». En: *arXiv preprint arXiv:2402.05131* (2024). URL: <https://arxiv.org/abs/2402.05131>.
  - [180] E. Jurado-Enríquez, K. Vargas-Prado, W. Melgarejo-Ángeles, Ú. Aniceto-Norabuena y T. Villacorta-Granados. «Integración de la Inteligencia Artificial Generativa en la docencia universitaria [Título en inglés]». En: *European Public & Social Innovation Review* 10 (2025), págs. 01-15. DOI: 10.31637/epsir-2025-1612. URL: <https://doi.org/10.31637/epsir-2025-1612>.
  - [181] E. Jurado-Enríquez, K. Vargas-Prado, W. Melgarejo-Ángeles, Ú. Aniceto-Norabuena y T. Villacorta-Granados. «Integración de la Inteligencia Artificial Generativa en la docencia universitaria [Título en inglés]». En: *European Public & Social Innovation Review* 10 (2025), págs. 01-15. DOI: 10.31637/epsir-2025-1612. URL: <https://doi.org/10.31637/epsir-2025-1612>.
  - [182] Kainat. *Arquitectura Encoder-Decoder*. 2023. URL: <https://medium.com/@kitkat73275/encoder-decoder-architecture-70232089c98e>.
  - [183] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen y Wen-tau Yih. «Dense Passage Retrieval for Open-Domain Question Answering». En: *arXiv preprint arXiv:2004.04906* (2020). Accedido el 12 de junio de 2025. URL: <https://arxiv.org/pdf/2004.04906.pdf>.
  - [184] KeepCoding. *Entrenamiento de una red neuronal en Deep Learning*. 2024. URL: <https://keepcoding.io/blog/entrenamiento-red-neuronal-deep-learning/>.

- [185] Diederik P. Kingma y Max Welling. «Auto-Encoding Variational Bayes». En: *arXiv preprint arXiv:1312.6114* (2014). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/1312.6114>.
- [186] Knewton. *Adaptive Learning Technology*. Recuperado de: <https://www.knewton.com/about/>. 2020.
- [187] LangChain. *LangChain Documentation*. Accedido el 20 de mayo de 2025. 2025. URL: <https://docs.langchain.com>.
- [188] LangChain. *LangSmith: Observability for LLM Applications*. Accedido el 11 de junio de 2025. 2024. URL: <https://smith.langchain.com>.
- [189] LangChain. *Retrievers - LangChain Documentation*. Accedido el 12 de junio de 2025. 2023. URL: <https://python.langchain.com/docs/concepts/retrievers/>.
- [190] Camila Larrosa. *Aprendizaje supervisado / Algoritmos, ejemplos*. Accedido el 28 de mayo de 2025. Feb. de 2023. URL: <https://datarmony.com/aprendizaje-supervisado-algoritmos-ejemplos/>.
- [191] Lastras. Explicación sobre sistemas RAG en Revista Cloud Computing. Accedido el 16 de junio de 2025. 2023. URL: <https://www.revistacloudcomputing.com/2023/08/que-es-la-generacion-aumentada-de-recuperacion/>.
- [192] Aprende Machine Learning. *¿Cómo funcionan los Transformers? Explicación en español con ejemplos de NLP, GPT y BERT*. Accedido el 29 de mayo de 2025. 2023. URL: <https://www.aprendemachinelearning.com/como-funcionan-los-transformers-espanol-nlp-gpt-bert/>.
- [193] Aprende Machine Learning. *¿Qué es el Machine Learning y cómo funciona?* 2024. URL: <https://aprendemachinelearning.com/que-es-machine-learning/>.
- [194] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. «Deep Learning». En: *Nature* 521.7553 (2015). Accedido el 28 de mayo de 2025, págs. 436-444. DOI: 10.1038/nature14539. URL: <https://www.nature.com/articles/nature14539>.
- [195] Isidoro Gil Leiva y José Vicente Rodríguez Muñoz. «El procesamiento del lenguaje natural aplicado al análisis del contenido de los documentos». En: *Revista General de Información y Documentación* 6.1 (1996). Accedido el 29 de mayo de 2025, págs. 123-138. URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=169971>.
- [196] Grupo de Procesamiento de Lenguaje Natural. *Redes Neuronales para Lenguaje Natural: Clase 9 - Encoder-Decoder y Atención*. 2024. URL: [https://eva.fing.edu.uy/pluginfile.php/491493/mod\\_resource/content/4/RNLN%20-%20Clase%209%20-%20Encoder-Decoder%20y%20Atenci%C3%B3n.pdf](https://eva.fing.edu.uy/pluginfile.php/491493/mod_resource/content/4/RNLN%20-%20Clase%209%20-%20Encoder-Decoder%20y%20Atenci%C3%B3n.pdf).
- [197] P. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed y R. Leblond. «Retrieval-augmented generation for knowledge-intensive NLP tasks». En: *Proceedings of NeurIPS 2020*. 2020. URL: <https://arxiv.org/abs/2005.11401>.
- [198] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». En: *Advances in Neural Information Processing Systems (NeurIPS)* (2020). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2005.11401>.

- [199] Chin-Yew Lin. «ROUGE: A Package for Automatic Evaluation of Summaries». En: *Proceedings of the Workshop on Text Summarization Branches Out*. 2004, págs. 74-81. URL: <https://www.aclweb.org/anthology/W04-1013.pdf>.
- [200] Zachary C. Lipton, John Berkowitz y Charles Elkan. «A Critical Review of Recurrent Neural Networks for Sequence Learning». En: *arXiv preprint arXiv:1506.00019* (2015). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/1506.00019>.
- [201] Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A. Smith y Yejin Choi. «We're Afraid Language Models Aren't Modeling Ambiguity». En: *arXiv preprint arXiv:2304.14399* (2023). URL: <https://arxiv.org/abs/2304.14399>.
- [202] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zihan Jiang, Hiroaki Hayashi y Graham Neubig. «Prompt Engineering: A Survey». En: *arXiv preprint arXiv:2301.13688* (2023). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2301.13688>.
- [203] Pengfei Liu et al. «Pre-train Prompt Tuning: Towards Better Few-shot Learning with Language Models». En: *arXiv preprint arXiv:2109.04332* (2023). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2109.04332>.
- [204] R. Luckin. «Towards Artificial Intelligence-based Assessment Systems». En: *Nature Human Behaviour* 1 (2017), pág. 0028. DOI: 10.1038/s41562-016-0028.
- [205] R. Luckin, W. Holmes, M. Griffiths y L. B. Forcier. *Intelligence Unleashed: An argument for AI in education*. Pearson, 2016.
- [206] Rodríguez E. & Hernández A. & Martínez M. «Prácticas de personalización del aprendizaje mediadas por las tecnologías digitales: una revisión sistemática». En: *EDUTEC. Revista Electrónica de Tecnología Educativa* 83 (2023). URL: <https://www.edutec.es/revista/index.php/edutec-e/article/view/3019/1197>.
- [207] Management Solutions. *El auge de los large language models: de los fundamentos a la aplicación*. Technical Report. Accedido el 18 de junio de 2025. Management Solutions, 2024. URL: <https://www.managementsolutions.com/sites/default/files/minisite/static/72b0015f-39c9-4a52-ba63-872c115bfbd0/11m/pdf/auge-de-los-11m.pdf>.
- [208] Subhadeep Mandal. *A Beginner's Guide to Evaluating RAG Pipelines Using RAGAS*. Accedido el 19 de abril de 2025. 2024. URL: <https://www.analyticsvidhya.com/blog/2024/05/a-beginners-guide-to-evaluating-rag-pipelines-using-ragas/>.
- [209] Y. Mansar. *How to Use Transformer Networks to Build a Forecasting Model*. Accedido el 29 de mayo de 2025. 2021. URL: <https://towardsdatascience.com/machine-learning-for-store-demand-forecasting-and-inventory-optimization-part-1-xgboost-vs-9952d8303b48/>.
- [210] Francisca e PEREZ-GARCIAS Adolfina MARIN-JUARROS Victoria I.; NEGRE-BENNASAR. «Entornos y redes personales de aprendizaje (PLE-PLN) para el aprendizaje colaborativo. Comunicar». En: *Educación* 21.42 (2014). Accedido el 29 de mayo de 2025. URL: [http://educa.fcc.org.br/scielo.php?pid=S1988-32932014000100005&script=sci\\_abstract&tlang=en](http://educa.fcc.org.br/scielo.php?pid=S1988-32932014000100005&script=sci_abstract&tlang=en).
- [211] Bernard Marr. *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Wiley, 2016.

- [212] Miguel Ángel Martínez Prieto, Jorge Silvestre, Aníbal Bregón, Patricia Baz Domínguez, Clara Gándara González, Paula Mielgo Martín e Irene Peñas Pérez. «Una metodología basada en prácticas ágiles para la realización de Trabajos Fin de Grado». En: *Actas de las JENUI - Vol. 8 (2023)*. 2023, págs. 367 -374.
- [213] Miguel Á. Martínez Prieto, Jorge Silvestre, Aníbal Bregón y Diego García Álvarez. «Proyectos de aprendizaje (ÁGILES)». En: *Innovación docente en Educación Superior: interacción, participación y colaboración*. Valladolid: Universidad de Valladolid, 2023, pág. 287. ISBN: 978-84-18583-04-9.
- [214] Adnan Masood. *Optimizing Chunking, Embedding, and Vectorization for Retrieval-Augmented Generation*. Accedido el 16 de mayo de 2025. 2024. URL: <https://medium.com/@adnanmasood/optimizing-chunking-embedding-and-vectorization-for-retrieval-augmented-generation-ea3b083b68f7>.
- [215] MATHia. *MATHia*. 2024. URL: <https://www.carnegielearning.com/mathia/>.
- [216] MathWorks. *Autoencoders*. 2024. URL: <https://es.mathworks.com/discovery/autoencoder.html>.
- [217] MathWorks. *Deep Learning*. 2025. URL: <https://es.mathworks.com/discovery/deep-learning.html>.
- [218] Viktor Mayer-Schönberger y Kenneth Cukier. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013.
- [219] John McCarthy, Marvin Minsky, Claude Shannon y John von Neumann. «A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence». En: *Dartmouth College*. 1956.
- [220] Warren S. McCulloch y Walter Pitts. «A Logical Calculus of the Ideas Immanent in Nervous Activity». En: *The Bulletin of Mathematical Biophysics* 5.4 (1943). Accedido el 28 de mayo de 2025, págs. 115-133. DOI: 10.1007/BF02478259. URL: <https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>.
- [221] MindsDB. *Navigating the LLM Landscape: A Comparative Analysis of Leading Large Language Models*. Accedido el 20 de mayo de 2025. 2024. URL: <https://mindsdb.com/blog/navigating-the-llm-landscape-a-comparative-analysis-of-leading-large-language-models>.
- [222] ML4A. *Redes Neuronales*. Accedido el 28 de mayo de 2025. URL: [https://ml4a.github.io/ml4a/es/neural\\_networks/](https://ml4a.github.io/ml4a/es/neural_networks/).
- [223] Víctor Mollá. *¿Qué es RAG? Una gran arquitectura en LLMs*. Accedido el 15 de mayo de 2025. 2024. URL: <https://www.victormolla.com/que-es-rag>.
- [224] András Molnár. «How generative AI is reshaping industries». En: *McKinsey Quarterly* (2023). URL: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-generative-ai>.
- [225] S. Moquin. «Ethical Considerations For AI Use In Education». En: *Enrollify* (2024). URL: <https://www.enrollify.org/blog/ethical-considerations-for-ai-use-in-education>.

- 
- [226] Ana Moreno. *Informe Delphi: Personalizar, un modelo para una educación de calidad en el siglo XXI*. Inf. téc. Consultado el 8 de mayo de 2025. Impuls Educació, 2020. URL: [https://impulseducacio.org/wp-content/uploads/2020/04/INFORME-DELPHI\\_CAST\\_DEF.pdf](https://impulseducacio.org/wp-content/uploads/2020/04/INFORME-DELPHI_CAST_DEF.pdf).
  - [227] MyScale. *Streamlit vs Gradio: The Ultimate Showdown for Python Dashboards*. Accedido el 19 de mayo de 2025. 2024. URL: <https://myscale.com/blog/es/streamlit-vs-gradio-ultimate-showdown-python-dashboards/>.
  - [228] MZG Asesores. *¿Cuánto cuesta dar de alta a un trabajador un día?* Accedido el 28 de mayo de 2025. Oct. de 2024. URL: <https://www.mzgasesores.es/blog/cuanto-cuesta-dar-de-alta-a-un-trabajador-un-dia>.
  - [229] Netra Prasad Neupane. *Retrieval Augmented Generation (RAG) using LlamaIndex and Mistral—7B*. Accedido el 20 de mayo de 2025. 2024. URL: <https://netraneupane.medium.com/retrieval-augmented-generation-rag-using-llamaindex-and-mistral-7b-228f93ba670f>.
  - [230] eCampus News. *College Students Using AI to Improve Grades*. Accedido: 01 abril 2025. 2024. URL: <https://www.ecampusnews.com/teaching-learning/2024/07/12/college-students-ai-better-grades/#:~:text=Fifty%20Done%20percent%20of%20spring,point%20increase%20from%20Fall%20'23>.
  - [231] Nils J. Nilsson. *Inteligencia Artificial: Una Nueva Síntesis*. Accedido el 10 de junio de 2025. Madrid: McGraw-Hill, 2000.
  - [232] OECD. *The State of School Education: One Year into the COVID Pandemic*. 2021. URL: <https://doi.org/10.1787/201dde84-en>.
  - [233] OECD. *Trends Shaping Education 2019*. Accedido el 7 de marzo de 2025. 2019. URL: [https://doi.org/10.1787/trends\\_edu-2019-en](https://doi.org/10.1787/trends_edu-2019-en).
  - [234] U.S. Copyright Office. *¿Puedo usar la obra de otra persona? ¿Alguien más puede usar la mía?* 2025. URL: <https://www.copyright.gov/espanol/faq/uso-justo.html>.
  - [235] Ollama. *Ollama Official Website*. Accedido el 20 de mayo de 2025. 2024. URL: <https://ollama.com/>.
  - [236] Shittu Olumide. *Getting Started with Building RAG Systems Using Haystack*. Accedido el 20 de mayo de 2025. 2025. URL: <https://www.kdnuggets.com/getting-started-building-rag-systems-haystack>.
  - [237] Órganos de la OMPI. *Convenio de Berna para la Protección de las Obras Literarias y Artísticas*. 1886. URL: <https://www.wipo.int/treaties/es/ip/berne/>.
  - [238] OpenAI. «GPT-4 Technical Report». En: *arXiv preprint arXiv:2303.08774* (2023). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2303.08774>.
  - [239] OpenAI. *Guía de modelos de embedding de OpenAI*. Accedido el 2020 de mayo de 2025. 2024. URL: <https://platform.openai.com/docs/guides/embeddings#embedding-models>.
  - [240] OpenAI. *LLM Optimization: Context*. Accedido el 16 de mayo de 2025. 2024. URL: <https://platform.openai.com/docs/guides/optimizing-llm-accuracy/llm-optimization-context>.

- [241] OpenAI. *OpenAI Platform Documentation Overview*. Accedido el 20 de mayo de 2025. 2024. URL: <https://platform.openai.com/docs/overview>.
- [242] OpenAI. *OpenAI y Khan Academy*. 2023. URL: <https://openai.com/index/khan-academy/>.
- [243] OpenWebinars. *Python vs JavaScript*. Accedido el 20 de mayo de 2025. 2024. URL: <https://openwebinars.net/blog/python-vs-javascript/>.
- [244] Oracle. *¿Qué es la generación aumentada de recuperación (RAG)?* Accedido el 15 de mayo de 2025. 2023. URL: <https://www.oracle.com/es/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/>.
- [245] Kishore Papineni, Salim Roukos, Todd Ward y Wei Zhu. «BLEU: A Method for Automatic Evaluation of Machine Translation». En: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2002, págs. 311-318. DOI: 10.3115/1073083.1073135.
- [246] Rajvardhan Patil y Venkat Gudivada. «A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs)». En: *Applied Sciences* 14.5 (2024). Accedido el 18 de junio de 2025, pág. 2074. DOI: 10.3390/app14052074. URL: <https://www.mdpi.com/2076-3417/14/5/2074>.
- [247] Pinecone. *Retrieval-Augmented Generation (RAG)*. Accedido el 20 de mayo de 2025. 2024. URL: <https://www.pinecone.io/learn/retrieval-augmented-generation/>.
- [248] PrimeWeb. *¿Cuántos usuarios tiene ChatGPT? Estadísticas 2023*. 2023. URL: <https://www.primeweb.com.mx/chatgpt-usuarios-estadisticas>.
- [249] ProFuturo. *IA generativa en la educación: ¿cómo lo hacemos?* 2023. URL: <https://profuturo.education/observatorio/competencias-xxi/ia-generativa-en-la-educacion-como-lo-hacemos/>.
- [250] Project Management Institute. *Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Sixth. Project Management Institute, 2017. ISBN: 9781628254518.
- [251] Prompting Guide. *Guía de Ingeniería de Prompt*. Accedido el 15 de mayo de 2025. 2024. URL: <https://www.promptingguide.ai/es>.
- [252] Prompting Guide. *Retrieval Augmented Generation (RAG) for LLMs*. Accedido el 18 de junio de 2025. 2025. URL: <https://www.promptingguide.ai/research/rag>.
- [253] PsicoSmart. *La personalización del aprendizaje y su impacto en la eficacia de los sistemas de gestión del aprendizaje*. 2023. URL: <https://psicosmart.net/es/articulos/articulo-la-personalizacion-del-aprendizaje-y-su-impacto-en-la-eficacia-de-los-sistemas-de-gestion-del-aprendizaje-189860>.
- [254] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu y Haifeng Wang. «RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering». En: *arXiv preprint arXiv:2010.08191* (2020). Accedido el 12 de junio de 2025. URL: <https://arxiv.org/pdf/2010.08191.pdf>.
- [255] Quizlet. *Quizlet*. 2024. URL: <https://quizlet.com/>.

- 
- [256] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li y Peter J. Liu. «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer». En: *Journal of Machine Learning Research* 21.140 (2020). Accedido el 29 de mayo de 2025, págs. 1-67. URL: <https://arxiv.org/abs/1910.10683>.
  - [257] Equipo de Ragas. *Ragas: Supercharge Your LLM Application Evaluations*. <https://github.com/explodinggradients/ragas>. Accedido el 19 de abril de 2025.
  - [258] Lee Rainie, Janna Anderson y Emily A. Vogels. *Experts Doubt Ethical AI Design Will Be Broadly Adopted as the Norm Within the Next Decade*. Pew Research Center, 2021. URL: <https://www.pewresearch.org/internet/2021/06/16/experts-doubt-ethical-ai-design-will-be-broadly-adopted-as-the-norm-within-the-next-decade/>.
  - [259] Luis Angel Pérez Ramos. *Modelos LLM y aplicaciones RAG paso a paso - Parte I - Introducción*. Accedido el 18 de junio de 2025. Oct. de 2024. URL: <https://es.community.intersystems.com/post/modelos-llm-y-aplicaciones-rag-paso-paso-parte-i-introducci%C3%B3n>.
  - [260] Rajveer Rathod. *Evaluating LLM Responses with DeepEval Library: A Comprehensive Practical Guide*. Accedido el 18 de junio de 2025. 2024. URL: <https://medium.com/@rajveer.rathod1301/evaluating-llm-responses-with-deepeval-library-a-comprehensive-practical-guide-e55ef1f9eeab>.
  - [261] Real Academia Española. *Diccionario de la Real Academia Española*. 23.<sup>a</sup> ed. RAE, 2023. URL: <https://rae.es/>.
  - [262] Redacción RRHH Digital. *Salarios de hasta 85.000€ en el sector tecnológico español: un análisis de las tendencias salariales de 2025*. Accedido el 28 de mayo de 2025. Dic. de 2024. URL: <https://www.rrhhdigital.com/secciones/actualidad/762796/salarios-de-hasta-85-000e-en-el-sector-tecnologico-espanol-un-analisis-de-las-tendencias-salariales-de-2025/>.
  - [263] Repsol. *Procesamiento del lenguaje natural*. Accedido el 29 de mayo de 2025. 2025. URL: <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/procesamiento-lenguaje-natural/index.cshtml>.
  - [264] RisingWave. *Chroma DB vs. Pinecone vs. FAISS: Vector Database Showdown*. Accedido el 20 de mayo de 2025. 2024. URL: <https://risingwave.com/blog/chroma-db-vs-pinecone-vs-faiss-vector-database-showdown/>.
  - [265] Joaquín Amat Rodrigo. *Redes neuronales con Python*. Accedido el 28 de mayo de 2025. Mayo de 2021. URL: <https://cienciadedatos.net/documentos/py35-redes-neuronales-python>.
  - [266] «Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part II». En: *Lecture Notes in Computer Science*. Ed. por Ido Roll, Danielle McNamara, Sergey Sosnovsky, Rose Luckin y Vania Dimitrova. Vol. 12749. Accedido el 29 de mayo de 2025. Springer, 2021. URL: <https://link.springer.com/content/pdf/10.1007/978-3-030-78270-2.pdf>.

- [267] Ido Roll, Danielle McNamara, Sergey Sosnovsky, Rose Luckin y Vania Dimitrova. «Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part II». En: *Lecture Notes in Computer Science*. Ed. por Ido Roll, Danielle McNamara, Sergey Sosnovsky, Rose Luckin y Vania Dimitrova. Vol. 12749. Springer, 2021. DOI: 10.1007/978-3-030-78270-2. URL: <https://link.springer.com/book/10.1007/978-3-030-78270-2>.
- [268] D. H. Rose y A. Meyer. *Teaching Every Student in the Digital Age: Universal Design for Learning*. Association for Supervision y Curriculum Development (ASCD), 2002.
- [269] Frank Rosenblatt. «The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain». En: *Psychological Review* 65.6 (1958). Accedido el 28 de mayo de 2025, págs. 386-408. DOI: 10.1037/h0042519. URL: <https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf>.
- [270] Aymeric Roucher. *Advanced RAG on Hugging Face documentation using LangChain*. Accedido el 19 de mayo de 2025. 2024. URL: [https://huggingface.co/learn/cookbook/advanced\\_rag](https://huggingface.co/learn/cookbook/advanced_rag).
- [271] Sujoy Roychowdhury, Sumit Soman, H G Ranjani, Neeraj Gunda, Vansh Chhabra y Sai Krishna Bala. «Evaluation of RAG Metrics for Question Answering in the Telecom Domain». En: *Proceedings of the ICML 2024 Workshop on Foundation Models in the Wild*. Accedido el 19 de abril de 2025. 2024. URL: <https://arxiv.org/abs/2407.12873>.
- [272] Laura Ruiz. *Neurona: qué es y cuáles son sus partes*. Accedido el 28 de mayo de 2025. Ago. de 2021. URL: <https://www.psyciencia.com/neurona-que-es-y-cuales-son-sus-partes/>.
- [273] Stuart Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Pearson, 2021.
- [274] Buko Sabino. *Framework de evaluación para el patrón RAG implementado en Justicio*. Accedido el 19 de abril de 2025. 2024. URL: <https://medium.com/@bukosabino/framework-de-evaluaci%C3%B3n-para-el-patr%C3%B3n-rag-implementado-en-justicio-d8ca06bee17f>.
- [275] Universidad Alfonso X el Sabio. *¿Qué es el Machine Learning y para qué sirve?* 2024. URL: <https://www.uax.com/blog/ingenieria/que-es-el-machine-learning>.
- [276] Tahir Saeed. *Chunking and Embedding Strategies in RAG: A Guide to Optimizing Retrieval-Augmented Generation*. Accedido el 16 de mayo de 2025. 2024. URL: [https://medium.com/@tahir.saeed\\_46137/chunking-and-embedding-strategies-in-rag-a-guide-to-optimizing-retrieval-augmented-generation-7c95432423b1](https://medium.com/@tahir.saeed_46137/chunking-and-embedding-strategies-in-rag-a-guide-to-optimizing-retrieval-augmented-generation-7c95432423b1).
- [277] Ramteja Sajja, Yusuf Sermet e Ibrahim Demir. «An Open-Source Dual-Loss Embedding Model for Semantic Retrieval in Higher Education». En: *arXiv preprint arXiv:2505.04916* (2025). Accedido el 18 de junio de 2025. URL: <https://arxiv.org/abs/2505.04916>.
- [278] Salesforce. *¿Qué es un asistente de IA?* Accedido el 19 de mayo de 2025. 2024. URL: <https://www.salesforce.com/es/agentforce/ai-assistant/#ai-assistant-defined>.
- [279] Francisco Sancho Caparrini. *Autoencoders Variacionales (VAE)*. Accedido el 16 de junio de 2025. URL: <https://www.cs.us.es/~fsancho/Blog/posts/VAE.md>.

- 
- [280] SAP. *¿Qué es el aprendizaje automático (machine learning)?* 2024. URL: <https://www.sap.com/spain/products/artificial-intelligence/what-is-machine-learning.html>.
- [281] SAP. *¿Qué es la IA generativa? / Ejemplos y casos de uso.* Visitado el 13 de mayo de 2025. 2025. URL: <https://www.sap.com/spain/products/artificial-intelligence/what-is-generative-ai.html>.
- [282] SAP. *¿Qué es la inteligencia artificial generativa?* Accedido el 14 de mayo de 2025. URL: <https://www.sap.com/spain/products/artificial-intelligence/what-is-generative-ai.html>.
- [283] Bhaskarjit Sarmah, Mingshu Li, Jingrao Lyu, Sebastian Frank, Nathalia Castellanos, Stefano Pasquali y Dhagash Mehta. *How to Choose a Threshold for an Evaluation Metric for Large Language Models.* Accedido el 18 de junio de 2025. 2024. arXiv: 2412.12148 [stat.ML]. URL: <https://arxiv.org/abs/2412.12148>.
- [284] Andrés Felipe Guerrero Sarmiento. *Integración de RAG en asistentes inteligentes para mejorar los procesos de enseñanza y aprendizaje en la Universidad de los Andes.* Accedido el 18 de junio de 2025. Bogotá, Colombia, 2025. URL: <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/b96eca78-a6eb-46ef-93b3-05c5d567409a/content>.
- [285] SAS Institute. *Deep Learning.* 2025. URL: [https://www.sas.com/es\\_es/insights/analytics/deep-learning.html](https://www.sas.com/es_es/insights/analytics/deep-learning.html).
- [286] Jürgen Schmidhuber. «Deep Learning in Neural Networks: An Overview». En: *Neural Networks* 61 (2015). Accedido el 28 de mayo de 2025, págs. 85-117. DOI: 10.1016/j.neunet.2014.09.003. URL: <https://arxiv.org/abs/1404.7828>.
- [287] Ken Schwaber y Jeff Sutherland. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.* 2020. URL: <https://scrumguides.org/scrum-guide.html>.
- [288] J. L. Serrano y J. Moreno-García. «Inteligencia artificial y personalización del aprendizaje: ¿innovación educativa o promesas recicladas?» En: *Edutec, Revista Electrónica de Tecnología Educativa* 89 (2024). Accedido el 16 de junio de 2025, págs. 1-17. DOI: 10.21556/edutec.2024.89.3577. URL: <https://doi.org/10.21556/edutec.2024.89.3577>.
- [289] Amazon Web Services. *¿Qué es el aprendizaje automático?* 2024. URL: <https://aws.amazon.com/es/what-is/machine-learning/>.
- [290] Servicios de Apoyo a la Investigación - Hospital Nacional de Parapléjicos. *3 ejemplos de entrenamientos de redes neuronales mediante Deep Learning para el análisis de imágenes mediante la Inteligencia Artificial (IA).* 2024. URL: <https://saishnp.com/2024/01/18/3-ejemplos-de-entrenamientos-de-redes-neuronales-mediante-deep-learning-para-el-analisis-de-imagenes-mediante-la-inteligencia-artificial-ia/>.
- [291] A. Sherstinsky. «Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network». En: *Physica D: Nonlinear Phenomena* 404 (2020). Accedido el 29 de mayo de 2025, pág. 132306. DOI: 10.1016/j.physd.2019.132306. URL: <https://arxiv.org/abs/1808.03314>.
- [292] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer y Wen-tau Yih. «Enhancing RAG with Reward Driven Supervision». En: *arXiv preprint arXiv:2410.03780* (2024). URL: <https://arxiv.org/abs/2410.03780>.

- [293] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer y Wen-tau Yih. «REPLUG: Retrieval-Augmented Black-Box Language Models». En: *arXiv preprint arXiv:2301.12652* (2023). URL: <https://arxiv.org/abs/2301.12652>.
- [294] Avi Shmidman, Cheyn Shmuel Shmidman, Dan Bareket, Moshe Koppel y Reut Tsarfaty. «Do Pretrained Contextual Language Models Distinguish between Hebrew Homograph Analyses?» En: *arXiv preprint arXiv:2405.07099* (2024). URL: <https://arxiv.org/abs/2405.07099>.
- [295] Sarah Smith. *¿Qué convierte a Perplexity AI en un motor de búsqueda revolucionario?* Consultado el 13 de mayo de 2025. 2025. URL: <https://www.pageon.ai/es/blog/perplexity-ai-search-engine-08u7f>.
- [296] SMOWL. *Personalización del aprendizaje: estrategias y herramientas a usar*. Consultado el 8 de mayo de 2025. 2022. URL: <https://smowl.net/es/blog/personalizacion-del-aprendizaje/>.
- [297] Management Solutions. *Machine Learning: definición y aplicaciones*. 2024. URL: <https://www.managementsolutions.com/sites/default/files/publicaciones/esp/machine-learning.pdf>.
- [298] Statista. *Encuestados que consideran esencial que los maestros se capaciten para usar la IA en la enseñanza*. Accedido: 01 de abril de 2025. 2024. URL: <https://es.statista.com/grafico/30894/encuestados-que-consideran-esencial-que-maestros-se-capaciten-para-usar-ia-en-la-ensenanza/>.
- [299] A. Stojanov. «Learning with ChatGPT 3.5 as a more knowledgeable other: An autoethnographic study». En: *International Journal of Educational Technology in Higher Education* 20.1 (2023), pág. 54. DOI: 10.1186/s41239-023-00404-7.
- [300] Chris Stokel-Walker. «ChatGPT listed as author on research papers: many scientists disapprove». En: *Nature* 613 (2023), págs. 620-621. DOI: 10.1038/d41586-023-00107-z.
- [301] Emma Strubell, Ananya Ganesh y Andrew McCallum. «Energy and Policy Considerations for Deep Learning in NLP». En: *arXiv preprint arXiv:1906.02243* (2019). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/1906.02243>.
- [302] Cole Stryker y Jim Holdsworth. *¿Qué es el procesamiento de lenguaje natural (PLN)?* Accedido el 29 de mayo de 2025. 2024. URL: <https://www.ibm.com/mx-es/think/topics/natural-language-processing>.
- [303] Jérémie Sublime e Ilaria Renna. «IA Generativa en la Educación: Una Nueva Frontera». En: *SciSimple* (2025). Consultado el 13 de mayo de 2025. URL: <https://scisimple.com/es/articles/2025-01-30-ia-generativa-en-la-educacion-una-nueva-frontera--a3jeerv>.
- [304] Ilya Sutskever, Oriol Vinyals y Quoc V. Le. «Sequence to Sequence Learning with Neural Networks». En: *arXiv preprint arXiv:1409.3215* (2014). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/1409.3215>.
- [305] Syntonize. *Programación de Inteligencia Artificial*. Accedido el 20 de mayo de 2025. 2024. URL: <https://www.syntonize.com/programacion-de-inteligencia-artificial/>.

- 
- [306] Muhammad Tahir, Ishaq Amanat y Sher Awan. «Ethical Considerations and Challenges in the Integration of Artificial Intelligence in Education: A Systematic Review». En: *Journal of Excellence in Management Sciences* 3 (sep. de 2024), págs. 35-50. DOI: 10.69565/jems.v3i4.314.
  - [307] Eshaan Tanwar, Gayatri Oke y Tanmoy Chakraborty. «Multilingual LLMs Struggle to Link Orthography and Semantics in Bilingual Word Processing». En: *arXiv preprint arXiv:2501.09127* (2025). URL: <https://arxiv.org/abs/2501.09127>.
  - [308] Georgia Tech. *Jill Watson - AI Teaching Assistant*. 2025. URL: <https://dilab.gatech.edu/jill-watson/>.
  - [309] Autor del TFG. *TFG: Estudio sobre modelos de lenguaje*. 2025. URL: <https://uvadoc.uva.es/bitstream/handle/10324/71277/TFG-G6876.pdf?sequence=1&isAllowed=y>.
  - [310] Nandan Thakur, Nils Reimers, Abhishek Sanni e Iryna Gurevych. «BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models». En: *arXiv preprint arXiv:2104.08663* (2021). Accedido el 19 de mayo de 2025. URL: <https://arxiv.org/abs/2104.08663>.
  - [311] A. F. Tipan. *TFG sobre Inteligencia Artificial en la UMH*. Accedido el 29 de mayo de 2025. 2025. URL: <https://dspace.umh.es/bitstream/11000/30273/1/TFG-Nasimba%20Tipan%2c%20Alexis%20Fabian.pdf>.
  - [312] Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavaf y Edward A. Fox. «Natural Language Processing Advancements By Deep Learning: A Survey». En: *arXiv preprint arXiv:2003.01200* (2020). Accedido el 29 de mayo de 2025. URL: <https://arxiv.org/abs/2003.01200>.
  - [313] Turing. *Python vs JavaScript: Complete Introduction*. Accedido el 20 de mayo de 2025. 2024. URL: <https://www.turing.com/kb/python-vs-javascript-complete-introduction>.
  - [314] Alan M. Turing. «Computing Machinery and Intelligence». En: *Mind* 59.236 (1950), págs. 433-460. DOI: 10.1093/mind/LIX.236.433.
  - [315] UE. *Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos*. 2016. URL: <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679>.
  - [316] UE. *Reglamento (UE) 2024/1689 del Parlamento Europeo y del Consejo de 13 de junio de 2024 por el que se establecen normas armonizadas sobre inteligencia artificial*. 2024. URL: <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>.
  - [317] UKPLab. *SentenceTransformers Documentation*. Accedido el 2020 de mayo de 2025. 2025. URL: <https://sbert.net/>.
  - [318] «Una evaluación integral de las técnicas de IA para predecir el índice de calidad del aire». En: *Dialnet* (2025). Accedido el 28 de mayo de 2025. URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=9984828>.
  - [319] UNED. *Usos de la inteligencia artificial en el aprendizaje*. Accedido el 14 de mayo de 2025. 2024. URL: <https://blogs.uned.es/bibliounedabierta/2024/11/26/usos-de-la-inteligencia-artificial-en-el-aprendizaje/>.

- [320] UNESCO. *Education: From disruption to recovery*. Recuperado de: <https://en.unesco.org/covid19/educationresponse>. 2020.
- [321] Unesco. *Global Education Monitoring Report: Inclusion and education – All means all*. 2020. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000373718>.
- [322] UNESCO. *Guidance for Generative AI in Education and Research*. Accedido el 19 de mayo de 2025. 2023. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000386693>.
- [323] UNESCO. *Reimagining our futures together: A new social contract for education*. 2022. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000379707>.
- [324] UNESCO. *Right from the Start: Investing in Early Childhood Education for a Sustainable Future*. La UNESCO enfatiza que una educación temprana de calidad no solo impacta en el conocimiento adquirido, sino también en la autoestima y la capacidad de los niños para afrontar desafíos futuros. Disponible en: <https://www.unesco.org>. 2021.
- [325] UNESCO. *The State of the Global Education Crisis: A Path to Recovery*. Disponible en: <https://www.unesco.org>. 2021.
- [326] Jakob Uszkoreit. *Transformer: A Novel Neural Network Architecture for Language Understanding*. Accedido el 29 de mayo de 2025. 2017. URL: <https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/>.
- [327] Valprovia. *Top 7 Challenges with Retrieval-Augmented Generation*. Accedido el 19 de mayo de 2025. 2024. URL: <https://www.valprovia.com/en/blog/top-7-challenges-with-retrieval-augmented-generation>.
- [328] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser e Illia Polosukhin. «Attention is All you Need». En: *Advances in Neural Information Processing Systems*. Ed. por I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan y R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: <https://arxiv.org/abs/1706.03762>.
- [329] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan y L. Sun. «Transformers in Time Series: A Survey». En: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI '23)*. Accedido el 29 de mayo de 2025. Macao, P.R. China, 2023. DOI: 10.24963/ijcai.2023/759. URL: <https://doi.org/10.24963/ijcai.2023/759>.
- [330] Wikipedia. *Atención (aprendizaje automático)*. n.d. URL: [https://es.wikipedia.org/wiki/Atenci%C3%B3n\\_\(aprendizaje\\_autom%C3%A1tico\)](https://es.wikipedia.org/wiki/Atenci%C3%B3n_(aprendizaje_autom%C3%A1tico)).
- [331] Wikipedia. *Socratic (Google) - Wikipedia*. 2023. URL: [https://en.wikipedia.org/wiki/Socratic\\_%28Google%29](https://en.wikipedia.org/wiki/Socratic_%28Google%29).
- [332] Wikipedia contributors. *Aprendizaje automático – Wikipedia, La enciclopedia libre*. 2024. URL: [https://es.wikipedia.org/wiki/Aprendizaje\\_autom%C3%A1tico](https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico).
- [333] Wiley. *Knewton adaptive learning*. 2023. URL: <https://www.wiley.com/en-us/education/alta>.
- [334] Cameron R. Wolfe. *A Practitioner's Guide to Retrieval Augmented Generation (RAG)*. Accedido el 18 de junio de 2025. 2024. URL: <https://cameronrwolfe.substack.com/p/a-practitioners-guide-to-retrieval>.

- [335] World Economic Forum. *Con la IA generativa podemos reimaginar la educación —y el cielo es el límite.* 2024. URL: <https://es.weforum.org/stories/2024/02/con-la-ia-generativa-podemos-reimaginar-la-educacion-y-el-cielo-es-el-limite/>.
- [336] WWWWhat's New. ¿Qué son los Modelos de Embeddings? ¿Cuál es mejor usar en mi RAG? Accedido el 2020 de mayo de 2025. 2025. URL: <https://wwwwhatsnew.com/2025/04/17/que-son-los-modelos-de-embeddings-cual-es-mejor-usar-en-mi-rag/>.
- [337] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang y Tie-Yan Liu. «On Layer Normalization in the Transformer Architecture». En: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 108. Accedido el 29 de mayo de 2025. PMLR, 2020, págs. 10524-10533. DOI: 10.48550/arXiv.2002.04745. URL: <https://arxiv.org/abs/2002.04745>.
- [338] D. C. Yadav y S. Pal. «15 - Measure the Superior Functionality of Machine Intelligence in Brain Tumor Disease Prediction». En: *Artificial Intelligence-Based Brain-Computer Interface*. Ed. por V. Bajaj y G. Sinha. Accedido el 29 de mayo de 2025. Academic Press, 2022, págs. 353-368. DOI: 10.1016/B978-0-323-91197-9.00005-9. URL: <https://doi.org/10.1016/B978-0-323-91197-9.00005-9>.
- [339] A. Yusuf, N. Pervin y M. Román-González. «Generative AI and the future of higher education: A threat to academic integrity or reformation? Evidence from multicultural perspectives». En: *International Journal of Educational Technology in Higher Education* 21.1 (2024), pág. 21. DOI: 10.1186/s41239-024-00314-x.
- [340] Erick Daniel Mero Alcívar Esther Verónica Ordóñez Valencia Paola Jomaira Villalba Poveda Vilma Esther Intriago Zambrano. «Implementación de la inteligencia artificial y el aprendizaje autónomo en la Educación para personalizar la enseñanza». En: *Revista Imaginario Social* (2024). Consultado el 8 de mayo de 2025. URL: <https://www.revista-imaginariosocial.com/index.php/es/article/view/209/365>.
- [341] O. Zawacki-Richter, V. I. Marín, M. Bond y F. Gouverneur. «Systematic review of research on artificial intelligence applications in higher education – where are the educators?» En: *International Journal of Educational Technology in Higher Education* 16.1 (2019), pág. 39. DOI: 10.1186/s41239-019-0171-0.
- [342] Jiayu Zheng, Deming Ye, Yixuan Xie, Maosong Sun, Tianyu Gao y Zhiyuan Liu. «Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena». En: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Accedido el 11 de junio de 2025. 2023. URL: <https://arxiv.org/pdf/2306.05685>.
- [343] S. Zheng. «The effects of chatbot use on foreign language reading anxiety and reading performance among Chinese secondary school students». En: *Computers and Education Artificial Intelligence* 7 (2024), pág. 100271. DOI: 10.1016/j.caeai.2024.100271. URL: <https://doi.org/10.1016/j.caeai.2024.100271>.
- [344] H. Zhou, X. Du y Z. Li. «Personalized learning in intelligent tutoring systems: Approaches and challenges». En: *Proceedings of the International Conference on Learning and Knowledge Analytics*. 2020. DOI: 10.1109/ICLKA48703.2020.9312373.
- [345] Linyong Zhou, Chao Chen, Xiang Ren et al. *ReGReT: Relevance-Grounded Beam Search for RAG*. Accedido el 11 de junio de 2025. 2023. URL: <https://arxiv.org/abs/2305.18405>.

## Bibliografía

---

- [346] Zilliz. *Optimizing RAG Applications: A Guide to Methodologies, Metrics, and Evaluation Tools for Enhanced Reliability*. Accedido el 19 de abril de 2025. 2024. URL: [https://medium.com/@zilliz\\_learn/optimizing-rag-applications-a-guide-to-methodologies-metrics-and-evaluation-tools-for-enhanced-a9ae3d9c7149](https://medium.com/@zilliz_learn/optimizing-rag-applications-a-guide-to-methodologies-metrics-and-evaluation-tools-for-enhanced-a9ae3d9c7149).
- [347] Azman Ö y Tümkaya S. «Navigating the ethical landscape of AI integration in education: Balancing innovation and responsibility [version 1; peer review: 1 not approved]». En: *F1000Research* 14.299 (2025). DOI: 10.12688/f1000research.160011.1.
- [348] Karel Čapek. *R.U.R. (Rossum's Universal Robots)*. Prague: Vydavatelské sdružení, 1921.

# Parte IV

## Apéndices



## Apéndice A

# Manual de Instalación

### Instalación del Producto

Una vez especificadas las herramientas a emplear, se ha decidido crear un entorno de desarrollo en Anaconda para facilitar la gestión de dependencias y garantizar la compatibilidad entre las bibliotecas utilizadas. Este entorno virtual permite manejar las herramientas de forma aislada, evitando posibles conflictos entre versiones de paquetes y asegurando un desarrollo más organizado y eficiente. Dado que el proyecto requiere el uso de versiones específicas de ciertas bibliotecas para garantizar su estabilidad y funcionamiento, a continuación se detallan las versiones de las principales herramientas y librerías utilizadas en el desarrollo del asistente A.1:

Paquete	Versión
Python	3.10.16
Langchain	0.3.18
Streamlit	1.41.1
Pandas	1.4.4
Numpy	1.26.4
Plotly	6.0.1
Pymupdf	1.23.26
Openai	1.61.1

Tabla A.1: Versiones de Paquetes para el entorno: Producto

Por tanto, en primer lugar se crearía el entorno y a continuación, se instalarían las versiones. Para ello, se efectuarían los siguientes pasos:

- Instalar Anaconda y abrir Anaconda Prompt
- Crear el entorno virtual: `conda create -name tfg`
- Activarlo: `conda activate tfg`
- Instalar las librerías necesarias: `pip install librería1 librería2 librería3...`

### Instalación del Entorno de Aceptación

Para la evaluación del producto se creará un entorno aparte para evitar conflictos entre librerías. Por tanto, los pasos a seguir serán los mismos que en el caso anterior, exceptuando las librerías necesarias, que son las que se muestran en la siguiente tabla A.2:

Paquete	Versión
Python	3.9.21
Datasets	3.5.1
Ragas	0.2.15
Pandas	2.2.3
Jupyter	1.1.1

Tabla A.2: Versiones de Paquetes para el entorno: Evaluación

# Apéndice B

## Manual de Usuario

### B.0.1. Prerrequisitos

#### Entorno

Se requiere de un entorno. Para ello, ver el anexo anterior, la instalación del Producto.

#### Creación del producto

Antes de poner en funcionamiento el asistente, es imprescindible disponer del material en el que se fundamenta el asistente de aprendizaje basado en RAG. En otras palabras, es necesario procesar los distintos documentos de la asignatura para tener el temario en el que basar el asistente. De esta forma, se crea en ChromaDB los distintos *embeddings* en función del archivo al que pertenezca y el lugar del que luego se extraerá la información. No obstante, otra alternativa sería poseer directamente la base de datos vectorial con los *embeddings* previamente cargados.

#### Conexión a Internet

Se necesita conexión a internet para poder conectarse con la API.

### B.0.2. Funcionamiento Producto Final

En primer lugar, se ha de abrir el entorno de Anconda Prompt creado: `conda activate`. A continuación, moverse a la ruta donde se ubicarían los archivos: `cd ruta_contiene_archivos`. Seguidamente, si es la primera vez que se accede a la aplicación, ejecutar `python process.py` para crear la base de datos vectorial. En caso contrario, no sería necesario y se pasaría finalmente al último paso de iniciar el asistente, mediante el comando: `streamlit run main.py`.

Al acceder a la aplicación, por defecto se muestra las consultas, pero mediante el menú de la izquierda se puede cambiar de actividad.

#### Consulta

De manera principal, permite al usuario seleccionar un objetivo, y en base a este, introducir la consulta en el campo de texto habilitado para ello. Destacar que si la entrada que se envía es vacía, se genera un mensaje informando de su obligatoriedad. Por otro lado, si resulta que

## Apéndice B. Manual de Usuario

---

presiona la tecla Enter para emitir su duda, se muestra una notificación indicando que es necesario pulsar el botón Generar respuesta para lanzar la pregunta e iniciar el proceso.

Una vez enviada la consulta correctamente, se presenta un *spinner* que señala que el sistema está procesando la solicitud, y no en un estado ocioso. Finalmente, la respuesta generada se muestra en un recuadro verde debajo del área de texto de la consulta. En cambio, si resulta que ha habido un error, de la API o en la base de datos, se mostraría en rojo la razón.

Resaltar al mismo tiempo que en la parte inferior, debajo de la solución ofrecida, se muestra la conversación entre el asistente y el usuario sobre peticiones y respuestas previas. En otras palabras, el usuario podrá ver el historial de la sesión. Asimismo, destacar la capacidad de respuesta del asistente para preguntas no relacionadas con el temario o con el objetivo.

Por último, una vez generada la respuesta, el usuario podrá cambiar la configuración y realizar otra cuestión sobre el objetivo que desee.

Selecciona una actividad:  
Consultas

Bienvenido a RAGile

RAGile  
Un revolucionador Asistente de Aprendizaje que podrás usar para que te ayude sobre los documentos reales de la asignatura

Consultas

Aquí puedes escribir cualquier consulta que tengas sobre el contenido de la asignatura. Puedes hacer preguntas generales, abiertas o específicas sobre algún tema. ¡Espero ayudarte!

Elege el objetivo principal en el que enfocar tu consulta  
SGBDs

Escribe tu consulta:

Generar respuesta

⚠️ No puedes enviar una consulta vacía

Figura B.1: Realizar Preguntas Abiertas (1/4)

Selecciona una actividad:  
Consultas

Bienvenido a RAGile

RAGile  
Un revolucionador Asistente de Aprendizaje que podrás usar para que te ayude sobre los documentos reales de la asignatura

Consultas

Aquí puedes escribir cualquier consulta que tengas sobre el contenido de la asignatura. Puedes hacer preguntas generales, abiertas o específicas sobre algún tema. ¡Espero ayudarte!

Elege el objetivo principal en el que enfocar tu consulta  
Lógico

Escribe tu consulta:  
Qué es normalizar?

Generar respuesta

Haz clic en 'Generar respuesta' para lanzar la consulta.

Figura B.2: Realizar Preguntas Abiertas (2/4)

## 💡 Respuesta:

La normalización en el contexto de bases de datos se refiere al proceso de organizar los datos de una manera que minimice la redundancia y evite las anomalías, garantizando así la integridad y consistencia de la información. Esto se logra identificando las dependencias funcionales entre los atributos de una tabla y dividiendo la información en tablas más pequeñas y relacionadas entre sí. El objetivo final de la normalización es obtener tablas que cumplan con ciertas propiedades deseables, como contener el número mínimo de atributos necesarios, reducir la redundancia de datos y facilitar el acceso y mantenimiento de la información en la base de datos.

 Ver fragmentos fuente usados

## 💬 Conversación de esta sesión:

 Tú: Qué es normalizar?

 RAGILE: La normalización en el contexto de bases de datos se refiere al proceso de organizar los datos de una manera que minimice la redundancia y evite las anomalías, garantizando así la integridad y consistencia de la información. Esto se logra identificando las dependencias funcionales entre los atributos de una tabla y dividiendo la información en tablas más pequeñas y relacionadas entre sí. El objetivo final de la normalización es obtener tablas que cumplan con ciertas propiedades deseables, como contener el número mínimo de atributos necesarios, reducir la redundancia de datos y facilitar el acceso y mantenimiento de la información en la base de datos.

Figura B.3: Realizar Preguntas Abiertas (3/4)

Eige el objetivo principal en el que enfocar tu consulta

Lógico

 Escribe tu consulta:

Qué es una patata?

 Generar respuesta

## 💡 Respuesta:

AVISO: contexto insuficiente para generar la respuesta únicamente a partir de los apuntes proporcionados. Te recomendaría revisar el material proporcionado o reformular tu pregunta para poder brindarte una respuesta más precisa y fundamentada. ¿Hay algo más en lo que pueda ayudarte?

Figura B.4: Realizar Preguntas Abiertas (4/4)

## Test

Al seleccionar esta actividad, el usuario ha de configurar previamente cómo desea su cuestionario: en base a qué objetivos, historias, criterios, dificultad y número de preguntas. Para ello, se presentan los distintos elementos ya vistos, aunque comentar aun así varios aspectos.

En primer lugar, el número de preguntas puede ajustarse con un *slider*, con un rango del 1 al 10, siendo 3 el valor por defecto. A la hora de seleccionar un objetivo, se ofrece la posibilidad de consultar su descripción, así como la de la historia de aprendizaje y criterios relacionados. Por tanto, facilita que el alumno comprenda qué aspectos se evaluarán en el test y también que pueda profundizar en ellos.

## Apéndice B. Manual de Usuario

En cuanto a los criterios, si no se selecciona ninguno, el sistema asume por defecto que se aplican todos (habilitando un mensaje informativo para ello). Además, se permite seleccionar todos los criterios de forma independiente mediante una opción capacitada para ello. Destacar que tanto las historias como los criterios se actualizan dinámicamente en función del objetivo seleccionado. Finalmente, se indicaría el nivel de dificultad.

Una vez configurados todos los parámetros, al pulsar el botón Generar Test, se muestra un *spinner* y un resumen de la configuración elegida. A continuación, se inicia el cuestionario (si no ha habido ningún error previo), donde se carga el contenido dinámico y se presentan las preguntas con sus respectivas opciones; ninguna seleccionada por defecto. Si se intenta avanzar sin seleccionar una respuesta, el sistema emite una advertencia.

Al finalizar el test, se muestra un resumen del rendimiento y desempeño del usuario y se ofrece la opción de volver al inicio.

The screenshot shows the RAGile platform's configuration interface for a 'Test Type' question. On the left, there's a sidebar with a dropdown menu for selecting an activity, currently set to 'Tipo Test'. Below it is a welcome message: 'Bienvenido a RAGile'. The main content area has a header 'Configura tu Test' with two sections: '1. Elige el número de preguntas:' and '2. Selecciona un objetivo:'. In section 1, a slider is set to 3. In section 2, the objective is selected as 'SGBDs'. A detailed description of the objective is provided: 'Estudia los sistemas que permiten almacenar, gestionar y recuperar datos estructurados de forma eficiente.'.

Figura B.5: Solicitar Preguntas Tipo Test (1/7)

This screenshot continues the configuration of a 'Test Type' question. It shows the '3. Historia de usuario:' section where the user selects 'Fundamentos de Bases de Datos' as the related user history. A detailed description of this history is provided: 'Conocer los fundamentos conceptuales y tecnológicos de las bases de datos para poder diseñar y construir soluciones que satisfagan los requisitos de información de los sistemas software a desarrollar en el ámbito profesional.' Below this is the '4. Criterios de evaluación:' section, which lists various evaluation criteria: 'Rol BD', 'Limitaciones ficheros', 'SGBD / BD', 'Funciones SGBD', 'Componentes SGBD', 'Ventajas y desventajas SGBD', 'Arquitectura SGBD', and 'Independencia lógica/física'. Each criterion has a brief description next to it.

Figura B.6: Solicitar Preguntas Tipo Test (2/7)

---

Selecciona los criterios específicos a evaluar

Choose an option ▾

⚠ No has seleccionado ningún criterio. Se usarán todos los disponibles por defecto.

**Nivel de Dificultad:**

Selecciona la dificultad del Test:

● Fácil    ● Fácil    ● Medio    ● Difícil

---

 Generar Test

Figura B.7: Solicitar Preguntas Tipo Test (3/7)

Rol BD ×

✓ Criterios seleccionados: 1

1. Rol BD

**Nivel de Dificultad:**

Selecciona la dificultad del Test:

● Fácil    ● Fácil    ● Medio    ● Difícil

---

 Generar Test

○ Generando preguntas tipo test...

Configuración del test:

- Objetivo: SGBDs
- Historia: Fundamentos de Bases de Datos
- Criterios: Rol BD
- Número de preguntas: 3
- Dificultad: ● Fácil

Figura B.8: Solicitar Preguntas Tipo Test (4/7)

## Apéndice B. Manual de Usuario

1. Pregunta 1: ¿Qué clase de usuarios no son conscientes de la existencia de la base de datos y utilizan programas de aplicación diseñados para interactuar con ella de forma sencilla?

— Selecciona una opción —

- A) Usuarios avanzados
- B) Administradores
- C) Diseñadores
- D) Usuarios ingenuos

2. Pregunta 2: ¿Qué componente del gestor de la base de datos se encarga de determinar la estrategia óptima para la ejecución de las instrucciones necesarias para resolver una operación solicitada?

— Selecciona una opción —

- A) Control de autorización
- B) Procesador de comandos
- C) Verificador de integridad
- D) Optimizador de consultas

3. Pregunta 3: ¿Qué ventaja se menciona en el contexto como resultado de utilizar un Sistema de Gestión de Bases de Datos?

— Selecciona una opción —

- A) Mayor complejidad en la seguridad
- B) Reducción de costes al integrar todos los datos de la organización
- C) Dificultad para resolver conflictos entre departamentos
- D) Limitación en el acceso a los datos

Finalizar Test

 Debes responder todas las preguntas antes de finalizar el test. Preguntas que no has contestado: 1, 2

Figura B.9: Solicitar Preguntas Tipo Test (5/7)

## Resultados del Test

 Aciertos: 1

 Fallos: 2

 Tiempo total: 47.96 segundos

 Puntuación acumulada: 1

Pregunta 1: ¿Qué clase de usuarios no son conscientes de la existencia de la base de datos y utilizan programas de aplicación diseñados para interactuar con ella de forma sencilla?

- Tu respuesta: C) Diseñadores
- Respuesta correcta: D) Usuarios ingenuos

 Incorrecto

Figura B.10: Solicitar Preguntas Tipo Test (6/7)

---

Pregunta 2: ¿Qué componente del gestor de la base de datos se encarga de determinar la estrategia óptima para la ejecución de las instrucciones necesarias para resolver una operación solicitada?

- Tu respuesta: A) Control de autorización
- Respuesta correcta: D) Optimizador de consultas

Incorrecto

Pregunta 3: ¿Qué ventaja se menciona en el contexto como resultado de utilizar un Sistema de Gestión de Bases de Datos?

- Tu respuesta: B) Reducción de costes al integrar todos los datos de la organización
- Respuesta correcta: B) Reducción de costes al integrar todos los datos de la organización

Correcto

Volver al inicio

Figura B.11: Solicitar Preguntas Tipo Test (7/7)

## Problemas

Al acceder a esta parte, el usuario selecciona el tipo de problema que desea practicar. Si resulta que se trata de un enunciado, se habilita un selector para elegir la temática del ejercicio. En caso de señalar la opción Otro, se permite introducir una temática personalizada.

Si el tipo de problema es SQL, se presentan distintos supuestos sobre los que se sustentará las funcionalidades del ejercicio para que el usuario indique uno.

Una vez pulsado el botón de Generar Problema, el *spinner* indica que está procesando la solicitud, y no en un estado ocioso. Finalmente, la respuesta generada se muestra en un recuadro verde debajo del área de texto de la consulta. En caso de algún error, se mostraría en rojo la razón.

Finalmente, resaltar que al mismo tiempo, en la parte inferior debajo de la solución ofrecida, se muestra la conversación entre el asistente y el usuario: el tipo de peticiones de los ejercicios y respuestas previas. En otras palabras, el usuario podrá ver el historial de la sesión.

## Apéndice B. Manual de Usuario

The screenshot shows the RAGile platform interface. At the top, there's a header with icons for user profile, brain, and the text "RAGile". Below it, a sub-header says "Un revolucionador Asistente de Aprendizaje que podrás usar para que te ayude sobre los documentos reales de la asignatura". A dropdown menu on the left says "Selecciona una actividad:" with "Problemas" selected. The main content area has a title "Problemas" with a yellow triangle icon. A sub-section titled "1. Tipo de problema:" asks "Selecciona el tipo de problema:" with "Enunciado Modelo ER" selected. Another section titled "2. Temática:" asks "Selecciona la temática sobre la que basar el enunciado del Modelo ER:" with several options listed, including "Música y arte" which is checked.

Figura B.12: Solicitar Supuestos Prácticos (1/3)

### 4. Generar problema:

Generar problema

#### ⭐ Problema generado:

La empresa "ArtSound" se dedica a la organización de conciertos exclusivos en los que se fusiona la música con el arte visual, ofreciendo experiencias únicas para los amantes de la cultura. Con el objetivo de gestionar de manera eficiente sus eventos, han decidido desarrollar un sistema informático denominado "ArtConcerts". Este sistema permitirá a los usuarios registrarse como artistas o asistentes, brindando así la posibilidad de participar activamente en los conciertos organizados por la empresa.

Cada artista registrado en "ArtConcerts" contará con un código único de identificación, su nombre artístico, género musical, país de origen y una breve descripción de su trayectoria. Además, se podrá especificar si el artista es solista o forma parte de una banda. Por otro lado, los asistentes tendrán la posibilidad de crear un perfil con su nombre, dirección de correo electrónico, ciudad de residencia y preferencias musicales.

Los conciertos organizados por "ArtSound" serán gestionados a través de "ArtConcerts", donde cada evento estará asociado a un artista principal y podrá contar con artistas invitados. Cada concierto tendrá un título, fecha, hora y lugar de celebración, así como un precio de entrada y un límite de asistentes. Además, se podrá indicar si el concierto es presencial o virtual, proporcionando la URL de acceso en caso de ser online.

Para enriquecer la experiencia de los asistentes, "ArtConcerts" permitirá la compra de obras de arte exclusivas relacionadas con los conciertos, las cuales estarán disponibles para su adquisición durante el evento. Cada obra de arte estará identificada por un código único, título, descripción, precio y artista creador. Asimismo, se registrarán las ventas realizadas durante cada concierto, incluyendo la fecha de compra, el comprador y la obra adquirida.

Además, "ArtConcerts" ofrecerá la posibilidad de que los asistentes puedan dejar reseñas y valoraciones sobre los conciertos y las obras de arte adquiridas, contribuyendo así a la retroalimentación y mejora continua de los eventos organizados por "ArtSound". Cada valoración contendrá una puntuación del 1 al 5 y comentarios adicionales para expresar la experiencia vivida en el concierto o con la obra de arte.

#### 📋 Historial de problemas generados:

Tipo: Enunciado Modelo ER | Supuesto: None

La empresa "ArtSound" se dedica a la organización de conciertos exclusivos en los que se fusiona la música con el arte visual, ofreciendo experiencias únicas para los amantes de la cultura. Con el objetivo de gestionar de manera eficiente sus eventos, han decidido desarrollar un sistema informático denominado "ArtConcerts". Este sistema permitirá a los usuarios registrarse como artistas o asistentes,

Figura B.13: Solicitar Supuestos Prácticos (2/3)

## ⚠ Problemas

Aquí podrás resolver problemas prácticos de la asignatura con soporte del asistente.

### 1. Tipo de problema:

Selecciona el tipo de problema:

SQL

### 2. Supuesto para SQL:

Selecciona el supuesto sobre el que quieras practicar:

- ElNorte+
- QuiereZon
- Flying your Dreams

### 4. Generar problema:

 Generar problema

## ⭐ Problema generado:

### Funcionalidades SQL:

F1. Obtener el nombre y la categoría de los hoteles situados en ciudades donde se oferten vuelos con fecha de ida entre el 1 y el 15 de marzo de 2025, ordenados alfabéticamente por el nombre del hotel.

F2. Listar el número de vuelos programados para el 10 de febrero de 2025 que tengan como destino un aeropuerto ubicado en un país diferente al de su origen, mostrando el código del vuelo y la aerolínea correspondiente

Figura B.14: Solicitar Supuestos Prácticos (3/3)

## Ver Progreso

Por último, al acceder a esta actividad, se presenta un dashboard con la información relacionada sobre el desempeño del usuario. En consecuencia, a raíz de los datos visualizados, es posible extraer conclusiones sobre el progreso alcanzado.

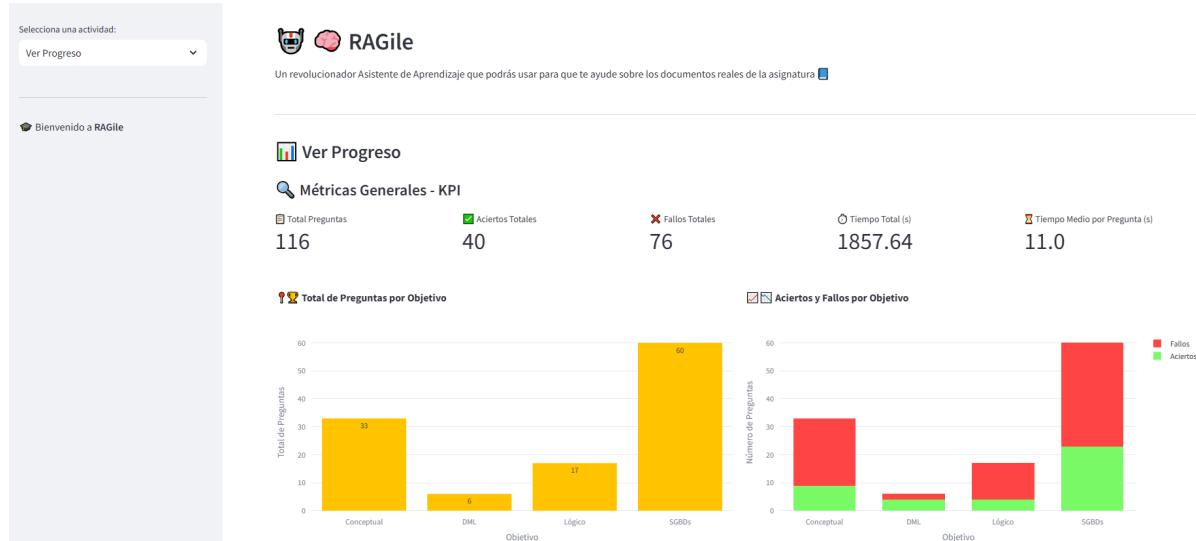


Figura B.15: Visualizar Progreso de Aprendizaje (1/3)

## Apéndice B. Manual de Usuario

---

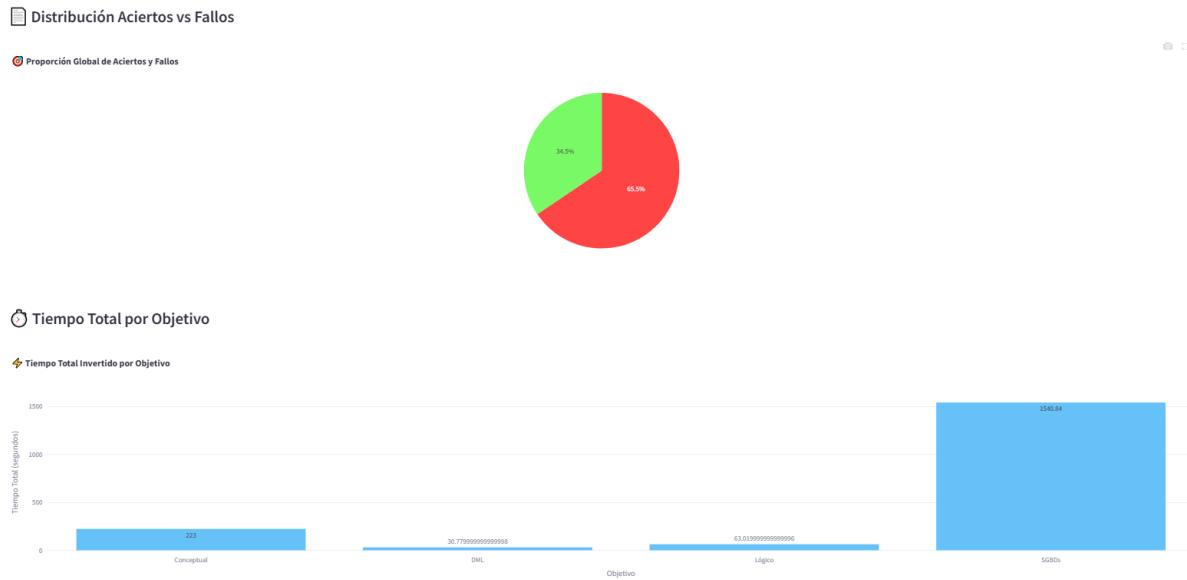


Figura B.16: Visualizar Progreso de Aprendizaje (2/3)

**Datos: Historial Completo de Tests**

	fecha	objetivo	subobjetivo	puntuacion_total	aciertos	fallos	tiempo_total	total_preguntas
0	2025-05-22	SGBDs	Modelo Relacional	1	1	1	3.86	2
1	2025-05-23	Conceptual	Fundamentos	1	1	2	12.2	3
2	2025-05-23	Conceptual	Fundamentos	1	0	3	11.39	3
3	2025-05-23	Conceptual	Diccionario de Datos	1	1	1	2.45	2
4	2025-05-23	SGBDs	Fundamentos de Bases de Datos	1	1	1	2.41	2
5	2025-05-23	SGBDs	Fundamentos de Bases de Datos	2	1	1	1.09	2
6	2025-05-23	SGBDs	Fundamentos de Bases de Datos	3	1	1	2.44	2
7	2025-05-23	SGBDs	Fundamentos de Bases de Datos	4	1	1	1.24	2
8	2025-05-23	SGBDs	Fundamentos de Bases de Datos	5	1	1	1.35	2
9	2025-05-23	SGBDs	Fundamentos de Bases de Datos	0	0	2	10.17	2

Figura B.17: Visualizar Progreso de Aprendizaje (3/3)

En este caso concreto, se puede apreciar que el usuario ha respondido un total de 116 preguntas, siendo 40 de ellas aciertos y 76 de errores. El tiempo total transcurrido ha sido de 1857,64 segundos, lo que implica de una media de 11 segundos por pregunta.

Las gráficas adjuntas respaldan estos resultados, y permiten una representación visual y más clara de la información. Por ejemplo, se evidencia un mayor número de preguntas realizadas sobre el tema de SGBDs, lo que también se traduce en un mayor tiempo dedicado a este apartado. Sin embargo, más de la mitad de las respuestas en esta categoría han sido incorrectas, lo cual podría implicar que sería recomendable que el alumno continúe practicando este objetivo para reforzar los conocimientos.

Finalmente, se puede consultar también la información en formato de tabla para apreciar su análisis detallado.

