

MEMORIA DEL PROYECTO.

*Chatbot aplicado a la resolución de consultas en asignaturas
de Bases de Datos*

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Septiembre de 2023

Autor:

Manuel Santa Isabel Mayo

Tutoras:

Ana Belén Gil González

Ana De Luis Reboredo

Certificado del/los tutor/es TFG

D./Dña. Ana Belén Gil González, profesor/a
del Departamento de Informática y Automática de la Universidad de Salamanca,

D./Dña. Ana de Luis Reboredo, profesor/a
del Departamento de Informática y Automática de la Universidad de Salamanca,

HACE/N CONSTAR:

Que el trabajo
titulado “Chatbot aplicado a la resolución de consultas en asignaturas de Bases de Datos_”,
que se presenta, ha sido realizado por Manuel Santa Isabel Mayo, con DNI 70263552R
y
constituye la memoria del trabajo realizado para la superación de la asignatura
Trabajo de Fin de Grado en Ingeniería Informática_ en esta Universidad.

Salamanca, 6 de septiembre 2023

Fdo.: _____

GIL GONZALEZ
ANA BELEN -
07973126S

Firmado digitalmente
por GIL GONZALEZ ANA
BELEN - 07973126S
Fecha: 2023.09.06
09:34:39 +02'00'

DE LUIS
REBOREDO,
ANA (FIRMA)

Firmado digitalmente
por DE LUIS REBOREDO,
ANA (FIRMA)
Fecha: 2023.09.06
20:19:48 +02'00'

RESUMEN

El aprendizaje de las bases de datos es uno de los ámbitos fundamentales de la informática. Es una materia que, por su amplia aplicación y complejidad, tiene una gran demanda de formación.

Con la llegada de la Inteligencia Artificial, el aprendizaje de cualquier materia, en especial en informática, ha supuesto un nuevo modelo de enseñanza y aprendizaje. Se basa en un modelo de autoaprendizaje adaptativo a las necesidades de cada usuario.

El presente Trabajo de Fin de Grado investiga y desarrolla el aprendizaje de las bases de datos mediante la integración de un Chatbot basado en Inteligencia Artificial, compuesto por una red neuronal, creada por el autor de este proyecto, y el modelo de Inteligencia Artificial de ChatGPT. Con el componente de Chatbot desarrollado, se pretende resolver los problemas que tengan los usuarios en esta área y adaptarse a las necesidades de los mismos.

A lo largo de este proyecto examinaremos cómo estas tecnologías innovadoras pueden mejorar el proceso de aprendizaje de las bases de datos, facilitando la comprensión de conceptos clave y ayudando en la aplicación práctica de los mismos.

Se ha realizado una investigación para entender cómo funcionan las Inteligencias Artificiales y dar respuesta y comprender el lenguaje natural de las personas. Siendo este objetivo, uno de los mayores retos que se plantean dentro del ámbito de las Inteligencias Artificiales, el Procesamiento del Lenguaje Natural (NLP). Vemos esta evolución del problema y cómo solucionarlo dentro de nuestro proyecto con las dos tecnologías que componen nuestra herramienta de Chatbot: ChatGPT (un modelo de Inteligencia Artificial muy avanzado e innovador) y una red neuronal secuencial (un modelo mucho más básico que pueda resolver el entendimiento del lenguaje natural).

Para implementar el modelo se ha diseñado una web denominada "DBCapy" que permite a los usuarios aprender bases de datos tanto a nivel teórico como práctico con ayuda de nuestro Chatbot.

En resumen, este Trabajo Fin de Grado se adentra en el aprendizaje de bases de datos con el propósito de demostrar cómo la inteligencia artificial aplicada a un Chatbot puede mejorar la manera en que los estudiantes adquieren y aplican conocimientos en este ámbito de la informática.

Palabras clave: *Inteligencia Artificial, aprendizaje Bases de Datos, Procesamiento de Lenguaje Natural, Chatbot, ChatGPT.*

ABSTRACT

Learning about databases is one of the fundamental domains within the field of computer science. It is a subject that, due to its extensive applicability and complexity, is in high demand for education.

With the advent of Artificial Intelligence, the learning of any subject, particularly in the field of computer science, has ushered in a new model of teaching and learning. This model is based on adaptive self-learning tailored to the unique needs of each user.

This bachelor's Final Project investigates and advances the learning of databases by integrating an Artificial Intelligence-based Chatbot. This Chatbot is composed of a neural network, created by the author of this project, and the Artificial Intelligence model of ChatGPT. With the developed Chatbot component, the aim is to address users' challenges in this domain and adapt it to their specific requirements.

Throughout this project, we will examine how these innovative technologies can enhance the database learning process, simplifying the comprehension of key concepts, and aiding in their practical application.

Extensive research has been conducted to comprehend the functioning of Artificial Intelligences and to provide solutions and understanding of natural language from individuals, Natural Language Processing (NLP). Achieving this objective stands as one of the most significant challenges within the realm of Artificial Intelligence. We observe the evolution of this problem and how to tackle it within our project through the integration of two technologies that comprise our Chatbot tool: ChatGPT (an advanced and innovative Artificial Intelligence model) and a sequential neural network (a more basic model capable of natural language understanding).

To implement the model, a website named "DBCapy" has been designed, enabling users to learn about databases both theoretically and practically with the assistance of our Chatbot.

In summary, this bachelor's Final Project delves into the realm of database learning with the aim of demonstrating how artificial intelligence applied to a Chatbot can enhance how students acquire and apply knowledge in the field of computer science.

Keywords: *Artificial Intelligence, Database Learning, Natural Language Processing, Chatbot, ChatGPT.*

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
2. ANTECEDENTES	3
2.1. Primera toma de contacto con el proyecto	3
2.2. Proyectos similares	5
2.3. Primeras decisiones visuales de la web	9
2.4. Elección de la herramienta de Chatbot	11
3. OBJETIVOS DEL PROYECTO	13
3.1. Objetivos técnicos	13
3.2. Objetivos personales	14
4. CONCEPTOS TEÓRICOS	15
4.1. Redes neuronales secuenciales	15
4.2. Procesamiento del Lenguaje Natural (NLP) y modelo Bag of Words	17
4.3. Embedding y la evolución del modelo Transformer	19
4.4. Arquitectura Cliente-Servidor	21
5. METODOLOGÍA, TÉCNICAS Y HERRAMIENTAS	22
5.1. Metodología	22
5.2. Técnicas	33
5.2.1. Backend - Frontend	33
5.2.2. Método de Durán y Bernárdez	34
5.3. Herramientas	35
5.3.1. Herramientas usadas para crear el Chatbot	35
5.3.1.1. TensorFlow.....	35
5.3.1.1.1. NLTK WordNetLemmatizer	36
5.3.1.1.2. NLTK Word_tokenize	36
5.3.1.1.3. Pickle	36
5.3.1.2. ChatGPT	37
5.3.1.2.1. OpenAI API.....	37
5.3.1.2.2. TikToken.....	37
5.3.1.2.3. Ingeniería de prompts	38
5.3.1.2.4. Fine Tuning.....	39
5.3.2. Herramientas usadas para crear el entorno de la web	41
5.3.2.1. ReactJS.....	41
5.3.2.2. Node.JS.....	41
5.3.2.3 Django.....	41

5.3.2.3.1. Django Rest Framework.....	41
5.3.2.4 PostgreSQL.....	41
5.3.2.5 Djoser.....	42
5.3.2.6 Axios.....	42
5.3.3. Herramientas complementarias.....	43
5.3.3.1 DBDiagram.io.....	43
5.3.3.2 Draw.io.....	43
5.3.3.3 Microsoft Project.....	43
5.3.3.4 EZEstimate.....	43
5.3.3.5 Adobe Color.....	43
5.3.3.6 Google Forms.....	43
6. ASPECTOS RELEVANTES AL DESARROLLO.....	44
6.1. Marco de trabajo.....	44
6.1.1. Modelo de negocio.....	44
6.1.2. Requisitos.....	45
6.1.3. Análisis.....	51
6.1.4. Diseño.....	55
6.1.5. Implementación.....	57
6.1.5.1 Implementación de la Red Neuronal.....	58
6.1.5.2 Implementación de ChatGPT.....	62
6.1.5.3 Modelo de despliegue iconográfico.....	63
6.1.6. Pruebas.....	64
6.1.6.1. Pruebas chatbot.....	64
6.1.6.2. Pruebas de fiabilidad de respuestas.....	64
6.1.6.3 Pruebas de límites de tokens.....	65
6.1.6.4. Valoraciones de usuarios.....	65
6.2. Vista.....	66
6.2.1. Vista de usuario.....	66
6.2.2. Vista de teoría.....	67
6.2.3. Vista de práctica.....	68
7. RESULTADOS Y CONCLUSIONES.....	69
7.1. Resultados.....	69
7.2. Conclusiones.....	73
7.2.1. Conclusiones técnicas.....	73
7.2.2. Conclusiones de usuarios.....	73
7.3. Líneas de trabajo futuras.....	74

8. BIBLIOGRAFÍA	75
------------------------------	-----------

ÍNDICE DE FIGURAS

Figura 1. Sistema de BDDD MySQL en W3School.....	5
Figura 2. Ejercicios de MySQL en W3School.....	6
Figura 3. Extracto de la aplicación Socratic de Google.....	6
Figura 4. Extracto del proyecto "Desarrollo de un agente conversacional para el aprendizaje de SQL.....	7
Figura 5. Mascota de DBCapy.....	9
Figura 6. Paleta de colores de DBCapy.....	10
Figura 7. Logo de DBCapy.....	10
Figura 8. Uso de ChatGPT en una consulta de BBDD.....	12
Figura 9. Representación de una propagación hacia delante en una RNA.....	16
Figura 10. Ejemplo de lematización de la palabra "Change".....	17
Figura 11. Representación gráfica del modelo Bag of Words.....	18
Figura 12. Vector bag of words.....	18
Figura 13. Relación en contexto semántico de la palabra "perro" con otras palabras.....	19
Figura 14. Representación de un modelo Transformer.....	20
Figura 15. Ejemplo arquitectura Cliente-Servidor.....	21
Figura 16. Diagrama de Gantt 1.....	24
Figura 17. Diagrama de Gantt 2.....	25
Figura 18. Diagrama de Gantt 3.....	26
Figura 19. Diagrama de Gantt 4.....	27
Figura 20. Diagrama de Gantt 5.....	28
Figura 21. Diagrama de Gantt 6.....	29
Figura 22. Diagrama de Gantt 7.....	30
Figura 23. Diagrama de Gantt 8.....	31
Figura 24. Diagrama de Gantt 9.....	32
Figura 25. Ejemplo de Backend-Frontend con Django y React.....	33
Figura 26. Diferentes etapas del método de Durán y Bernárdez.....	34
Figura 27. Fichero con datos de entrenamiento para TensorFlow.....	35
Figura 28. Tokenización con ChatGPT y TikToken.....	38
Figura 29. Prompt en DBCapy.....	39
Figura 30. Set de datos para Fine Tuning.....	40
Figura 31. Entrenamiento de Davincii-003.....	40
Figura 32. Extracto del diagrama de Casos de Uso (Gestión chatbot desde teoría).....	45
Figura 33. Extracto del diagrama de Casos de Uso (Gestión del chatbot desde prácticxa).....	46
Figura 34. Diagrama de clases del sistema.....	51
Figura 35. UC-09 Diagrama de secuencia en el análisis.....	53
Figura 36. Subsistema del diseño.....	55
Figura 37. Subpaquete Chatbot Teoría.....	56
Figura 38. Modelo de despliegue.....	57
Figura 39. Endpoints del sistema.....	58
Figura 40. Procesamiento de frases en la RNA.....	60
Figura 41. Claves personales de OpenAI.....	62
Figura 42. Modelo de despliegue iconográfico o mockup.....	63
Figura 43. Coste llamadas a ChatGPT 3.5.....	65
Figura 44. Vista Logging.....	66
Figura 45. Vista teoría en su totalidad.....	67

Figura 46. Vista de teoría - Ejemplos de llamadas	67
Figura 47. Vista práctica.....	68
Figura 48. Vista práctica - Respuesta chatbot.....	68
Figura 49. Ejemplo de consistencia en el chatbot	69
Figura 50. Demográfica de usuarios que han probado la web	70
Figura 51. Encuesta de satisfacción - Preguntas de comodidad con ChatGPT	71
Figura 52. Encuesta de satisfacción – Chatbot	72

ÍNDICE DE TABLAS

Tabla 1. Catálogo de requisitos de información.....	46
Tabla 2. Catálogo de requisitos funcionales.....	47
Tabla 3. Catálogo de requisitos no funcionales.....	50

1. INTRODUCCIÓN

En la era digital actual, la tecnología ha revolucionado la forma en la que adquirimos conocimiento. Hace muy pocos años, cuando un estudiante tenía una duda, se pedían clases a profesores particulares para poder obtener una enseñanza de una forma más particular y adaptada a las necesidades de un alumno concreto. Con el paso de los años y las nuevas tecnologías, se ha podido estandarizar un amplio catálogo de ayudas como blogs, videos tutoriales y recientemente, inteligencias artificiales como ChatGPT, que nos sirven como apoyo para la enseñanza. Dentro del ámbito de la informática, este cambio de modelo de enseñanza lleva existiendo desde la aparición de estas nuevas tecnologías.

Teniendo en cuenta este contexto, se presenta DBCapy, una plataforma web de aprendizaje de la asignatura de Bases de Datos ofrecida por la USAL. Dentro de nuestra plataforma se aprovechará la potencia de la inteligencia artificial para ofrecer una experiencia educativa eficaz, enriquecedora y personal a las necesidades de cada alumno.

La enseñanza de la materia de Bases de Datos es fundamental, donde la gestión y el análisis de datos son habilidades esenciales para cualquier programador. Crearemos la web DBCapy, para poder formar a los usuarios en la materia de Bases de Datos. Además, implementaremos nuestro propio Chatbot para mejorar la experiencia de aprendizaje. Dicho bot ayudará tanto para aprender un temario estrictamente teórico de la materia, como con una ayuda para resolver ejercicios prácticos sobre Bases de Datos.

Nuestro Chatbot se compondrá de dos herramientas de inteligencias artificiales, una red neuronal Bag of Words (la cual solventará dudas relacionadas con el funcionamiento de la web Dbcapy y también conversaciones sencillas que no sean del ámbito de bases de datos) y ChatGPT (la cual nos permitirá un alto alcance para solventar las dudas de los alumnos). Este Chatbot tendrá múltiples funcionalidades:

- **Asistencia general:** El Chatbot cubrirá tanto dudas sobre la web (red neuronal) como dudas sobre la materia de Bases de Datos (ChatGPT).
- **Tutor Virtual:** Los usuarios pueden hacer preguntas específicas sobre conceptos, consultas o problemas relacionados con bases de datos. El chatbot aprovecha su capacidad de procesamiento de lenguaje natural para proporcionar explicaciones claras y ejemplos relevantes.
- **Corrección de ejercicios:** Nuestro Chatbot permitirá no solo corregir diferentes ejercicios sobre bases de datos, sino mostrar una explicación de cómo se ha corregido dicho ejercicio.

La clave fundamental de todas estas funciones es la de adaptarse a las necesidades de cada usuario de forma individual. Da igual la forma en la que un usuario formulé una pregunta, se le responderá y se le ayudará.

En cuanto a la estructura de esta memoria, en primer lugar, se hablará sobre los antecedentes de nuestro proyecto, analizando de esta forma diferentes webs que se usen para la enseñanza de bases de datos y puedan integrar inteligencias artificiales conversacionales. Después, se enumerarán los distintos objetivos de nuestro sistema, luego hablaremos de los

distintos conceptos teóricos que debemos conocer para entender bien las tecnologías usadas (en especial hablaremos sobre las inteligencias artificiales para el tratamiento del lenguaje natural). A continuación, se desarrollarán las metodologías, técnicas y herramientas usadas. Se presentarán también los aspectos relevantes del desarrollo del proyecto y finalizaremos con una sección de los resultados, conclusiones y líneas de trabajo a futuro.

Esta memoria se complementa con 6 archivos, que explican de forma más técnica los aspectos del desarrollo de nuestro proyecto:

- - **Anexo I: Plan del proyecto:** Estimación y planificación temporal del proyecto junto a un diagrama de Gantt para visualizar toda la planificación.
- - **Anexo II: Especificación de requisitos software:** Elicitación de requisitos del sistema junto a los objetivos y la división de dichos requisitos en subsistemas.
- - **Anexo III: Análisis de requisitos:** Diagrama de clases del sistema, análisis de los casos de uso representados de forma abstracta con diagramas de secuencia.
- - **Anexo IV: Diseño del sistema software:** análisis de los casos de uso de forma mucho menos abstracta y definición de interfaces.
- - **Anexo V: Documentación técnica:** Documentación técnica junto a explicación de uso de la herramienta de ChatGPT
- - **Anexo VI: Manual de usuario:** Explicación de las vistas de la aplicación y descripción de las funcionalidades que ofrecen

2. ANTECEDENTES

En este apartado vamos a reflejar las ideas con las que partimos para crear nuestro proyecto web. Empezaremos desde el primer enunciado que se nos entrega para crear la web, a continuación, veremos otras páginas o proyectos que tengan un propósito similar. Después, explicaremos todo lo relacionado con el aspecto visual que queremos que tenga nuestra web, justificando las decisiones estéticas que tomamos. Por último y más importante, valoraremos y exploraremos diferentes herramientas de ChatBot que sean útiles para nuestro proyecto. Al final de este punto justificaremos cual puede ser la mejor herramienta de Inteligencia Artificial para usar dentro de la aplicación.

Debemos llegar a unas conclusiones que se apliquen y se reflejen en el resto del desarrollo del proyecto. Las herramientas escogidas al final de este punto se explicarán de forma mucho más detallada en el apartado de memoria de Herramientas.

2.1. Primera toma de contacto con el proyecto

Este proyecto de fin de carrera fue propuesto con el siguiente enunciado: “...*La gran mayoría de chatbots se destinan a plataformas comerciales para asistencia al cliente o captación de estos. Esta propuesta plantea llevar a cabo el diseño y la implementación de un chatbot vinculado a la docencia en las asignaturas de bases de datos.*” Partiendo de esta premisa se ha planteado la creación de una herramienta de aprendizaje sobre la materia de Bases de datos junto a la ayuda de un Chatbot.

No se nos especifica cuáles pueden ser las funcionalidades que el Chatbot puede traer a la hora de crear el proyecto, así que debemos siempre validar y pensar diferentes opciones que puedan beneficiar a los estudiantes que vayan a aprender sobre la materia de Bases de Datos.

El desarrollador de este proyecto ya cursó la asignatura de BBDD, por tanto, ya partimos con una serie de ideas para poder ayudar en el aprendizaje de la materia.

Para comenzar, la asignatura dentro de la USAL se plantea en dos asignaturas distintas: Diseño de Bases de Datos y Sistemas de Bases de Datos. Ambas materias tienen una gran carga de temario teórico y práctico. Una buena forma en la que podemos comenzar a plantear la funcionalidad del Chatbot es en la ayuda de este temario teórico-práctico.

En ocasiones, las explicaciones teóricas pueden llegar a ser complicadas para un estudiante, ya que tratan términos y conceptos complejos de informática. Partiendo de este contexto, nuestro Chatbot puede resolver estos “cuellos de botella” teóricos. Por ejemplo, nuestro bot podría responder a dudas más particulares del temario y conseguir que un estudiante pueda comprender mejor un concepto en concreto.

Otro posible escenario para mejorar la calidad de estudio de la asignatura, es poder plantear ejemplos prácticos para el contenido teórico. Muchas veces, el ver conceptos de una forma más ejemplificada, puede ayudar en gran medida a aprender conceptos complejos. El uso del Chatbot en este supuesto, puede ser el de dar un ejemplo a los estudiantes de cualquier tema que se esté mostrando dentro de la web.

Un último escenario que se puede plantear para el contenido teórico de la web puede ser el de ampliar temarios que solo se mencionen de forma breve dentro de la web. A continuación, se redacta un ejemplo de este supuesto: Un usuario de la web se interesa con el concepto de “Inyección de SQL”, el cual es un tema que mezcla la ciberseguridad y los Sistemas de Bases de Datos. Este tema se menciona muy brevemente dentro de nuestra web. En esta

ocasión, el usuario podría usar la herramienta de Chatbot para poder ampliar sus conocimientos en esta materia de su interés.

Dejando de lado los posibles usos que podríamos plantear en nuestro sistema desde un contexto de enseñanza teórica, vamos ahora a plantear cómo se podría usar el Chatbot para el aprendizaje de BBDD con ejercicios prácticos.

A la hora de estudiar la asignatura de BBDD con ejercicios prácticos, se suelen pedir mayoritariamente ejercicios del lenguaje SQL. Estos ejercicios pueden ser algo complejos y difíciles de resolver cuanto más avanzamos dentro de la asignatura. En ocasiones, podemos llegar a bloquearnos y no conseguir resolver dichos ejercicios.

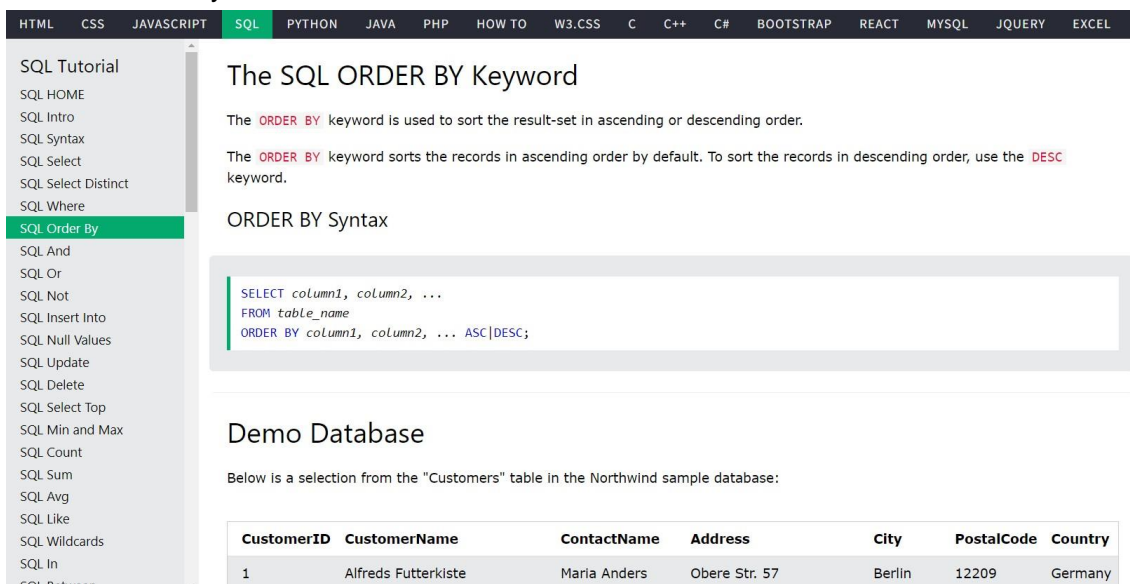
Una implementación de nuestro Chatbot dentro de la web es el de resolver los distintos ejercicios. Dando a su vez alguna explicación de cómo ha llegado a dicha conclusión. Otro posible enfoque es que no resuelva el ejercicio, solo buscamos dar diferentes pistas o guías para que un usuario aprenda para resolver el ejercicio.

Con todos estos conceptos e ideas, vamos a explorar diferentes proyectos o sistemas que se centren en la enseñanza de la materia de Bases de Datos y buscar ejemplos del uso de ChatBots para la enseñanza.

2.2. Proyectos similares

Las preguntas que debemos plantearnos para buscar proyectos similares al nuestro son las siguientes: ¿Es una web que enseña la materia de Bases de Datos?, ¿cómo implementa la enseñanza de dicha materia?, ¿se utiliza una herramienta de Chatbot para enseñar?, ¿qué funciones tiene dicho bot y cómo benefician a los estudiantes? Debemos buscar webs de enseñanza de la materia de Bases de Datos y webs que integren un Chatbot para la enseñanza.

Las primeras páginas webs que se pensaron al plantear estos objetivos fueron todas las webs dedicadas a la enseñanza de lenguajes de programación. Un ejemplo muy conocido es la web de W3Schools, la cual proporciona un gran catálogo de lenguajes y herramientas para la enseñanza de informática. En el ámbito que nos interesa, W3School tiene varias áreas dedicadas a la enseñanza de los lenguajes de SQL y MONGODB además del sistema de Bases de Datos MySQL.



The screenshot shows the W3Schools website interface. At the top, there is a navigation bar with various programming topics: HTML, CSS, JAVASCRIPT, SQL (highlighted), PYTHON, JAVA, PHP, HOW TO, W3.CSS, C, C++, C#, BOOTSTRAP, REACT, MYSQL, JQUERY, EXCEL. On the left, a sidebar menu lists SQL topics, with 'SQL Order By' selected. The main content area is titled 'The SQL ORDER BY Keyword' and explains that the ORDER BY keyword is used to sort the result-set in ascending or descending order. It notes that ORDER BY sorts records in ascending order by default, and to sort in descending order, the DESC keyword is used. Below the text is a code block containing a SQL query:

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

 Underneath the code block is a 'Demo Database' section. It states: 'Below is a selection from the "Customers" table in the Northwind sample database:'. Below this text is a table with the following data:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany

Figura 1. Sistema de BDDD MySQL en W3School

En la imagen adjuntada, podemos ver cómo se plantean distintos conceptos del lenguaje de programación SQL. A la vez que se plantean los conceptos se muestran diferentes ejemplos con Bases de Datos que los usuarios pueden probar.

Esta web nos sirve como una buena referencia de cómo podemos plantear nuestro contenido web, dividido en un índice con los distintos temas y siempre adjuntando con la explicación un ejemplo práctico.

Si seguimos observando esta web, comprobaremos cómo también existe un apartado dedicado a ejercicios para que los usuarios resuelvan.

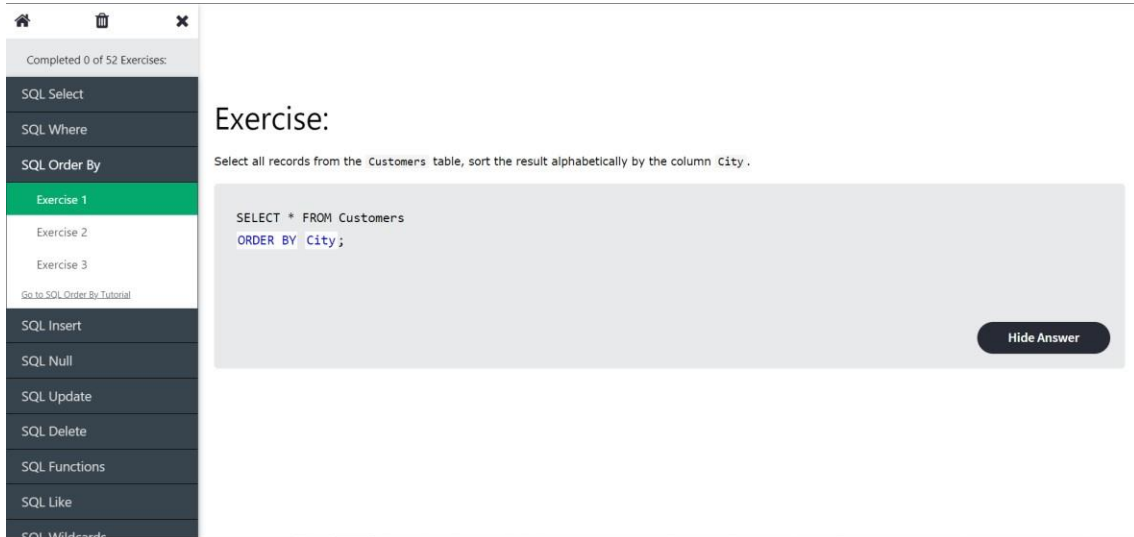


Figura 2. Ejercicios de MySQL en W3School

En el caso que se muestra en la imagen adjunta, se puede observar cómo un ejercicio puede mostrar su respuesta si nos quedamos bloqueados.

W3School es una muy buena referencia para plantear cómo podemos organizar nuestro proyecto en diferentes apartados y cómo la implementación de ejercicio a resolver para el usuario es de gran beneficio para los estudiantes de la materia.

No obstante, no podemos sacar ninguna referencia en esta web sobre el uso de Chatbot para la enseñanza. Vamos a investigar aplicaciones o proyectos que utilicen Chatbots para la enseñanza.

Investigando nuevas tecnologías, descubrí una herramienta creada por Google llamada **Socratic**, la cual ofrece a los estudiantes ayuda personal con diferentes materias de estudio. El gran atractivo de dicha aplicación es el uso de inteligencias artificiales para poder responder de forma personal a los usuarios.



Figura 3. Extracto de la aplicación Socratic de Google

Aunque no tengamos un componente Chatbot como tal, podemos destacar cómo el uso de inteligencias artificiales puede ser muy beneficioso a la hora de poder resolver cualquier tipo de planteamiento y saber identificar la asignatura con la que estamos trabajando. Esta libertad de comprensión y de entendimiento de distintas materias, nos permite plantearnos un sistema que detecte el tema de la pregunta de un usuario. En nuestro caso, podríamos solo aceptar las preguntas que tengan una relación con la materia de Bases de Datos, el resto de las preguntas que son incoherentes con la asignatura las podríamos descartar.

El último proyecto que se ha buscado como referencia, es un proyecto creado por la universidad de Cádiz con un planteamiento y objetivos muy parecidos a los que se nos plantea para la creación de nuestro proyecto. El proyecto tiene de el nombre de: "[Desarrollo de un agente conversacional para el aprendizaje de SQL](#)". En este proyecto, se plantea un Chatbot para ayudar a los estudiantes con el aprendizaje de la programación de SQL. La idea del proyecto es crear una plataforma web en la que los profesores puedan plantear diferentes ejercicios sobre BBDD y los alumnos puedan resolverlos con diferentes pistas obtenidas de un Chatbot. Es un gran planteamiento para crear un Chatbot que enseñe de forma progresiva y adaptativa para cada usuario.

En vez de solo mostrar la solución, se planten diferentes pistas para poder ayudar a los usuarios.

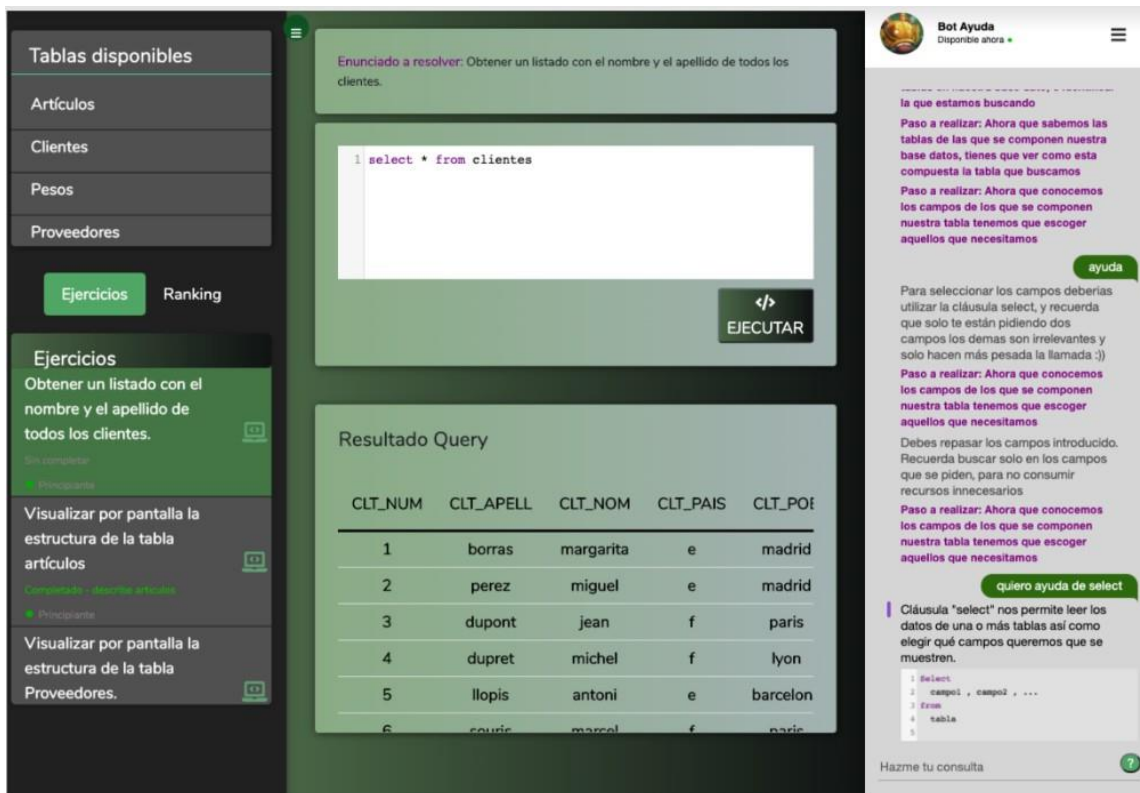


Figura 4. Extracto del proyecto "Desarrollo de un agente conversacional para el aprendizaje de SQL"

Podemos ver que se van dando pequeñas pistas dependiendo del tipo de duda que tenga un estudiante. Cada vez que el estudiante falle un ejercicio se mostrará una ayuda en el ChatBot. Es un planteamiento muy bueno de cómo poder aplicar un Chatbot dentro de nuestra aplicación de enseñanza.

No obstante, creo que el modo de resolver ejercicios con pequeñas pistas aún puede causar un problema para algunos estudiantes, los cuales aun con muchas pistas no entienden cómo

resolver un ejercicio. Una posible alternativa es que el mismo chatbot pueda ofrecer tanto estas pequeñas pistas como la resolución y explicación de un ejercicio práctico. De esta forma los estudiantes pueden progresar sin tener problemas de bloquearse en ciertos ejercicios.

Después de analizar cada una de estas plataformas, podemos avanzar en nuestros antecedentes creando una lista de decisiones visuales basadas en diferentes ejemplos que hayamos observado.

2.3. Primeras decisiones visuales de la web

Basándonos en los distintos ejemplos que hemos observado en el apartado anterior, vamos a tomar unas ciertas decisiones estéticas y de diseño del proyecto que vamos a crear.

Lo primero que analizaremos será cómo vamos a distribuir nuestro contenido de la web, luego, hablaremos del estilo y la marca que queremos usar para nuestro proyecto.

Lo primero de todo, como hemos visto en la web de **W3Schools**, una buena forma de distribuir la asignatura de Bases de datos es dividiendo la web en dos partes. Una parte de temario donde se expliquen diferentes conceptos sobre la materia, (incluyendo algún ejemplo práctico) y una segunda parte en la que los estudiantes puedan resolver diferentes ejercicios. Dichas partes deben de estar entrelazadas de alguna forma, para que podamos navegar desde el apartado de teoría al de práctica de forma intuitiva. Es muy importante que cada ejercicio a resolver tenga que ver con un apartado de la teoría. Con la práctica, los estudiantes van a poder entender mucho mejor la materia de Bases de Datos.

Respecto al tema de la estética de nuestro proyecto, debemos observar cuáles son los estilos (tipografía, colores, etc.) que atraigan a los estudiantes, en concreto a los estudiantes de informática. Buscamos un estilo que sea tranquilo pero que nos recuerde que estamos dentro de una aplicación para aprender una materia informática. Otro factor que se ha tenido en cuenta es el uso de un logo atractivo.

A la hora de abordar este proyecto, se valoró la idea de usar un animal como símbolo de nuestra aplicación. Otros proyectos enfocados hacia el aprendizaje (o producto informático) tienen diferentes mascotas que reflejen los objetivos de su producto. Por ejemplo, la enseñanza con el búho de **Duolingo** (y también de la anteriormente mencionada **Socratic**), o la seguridad y firmeza del gestor de bases de datos **PostgreSQL** representada con un elefante.

En el caso de este proyecto, se ha decidido que un animal que puede ser representativo es el capibara. Este animal tranquilo ha conseguido gran popularidad en estos últimos años. Debido a que vamos a utilizar herramientas novedosas como el chatbot y que deseamos que los estudiantes al usar nuestra web se sientan tranquilos, el animal que puede representar más fielmente a nuestra aplicación es el capibara.

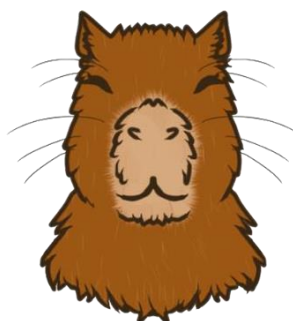


Figura 5. Mascota de DBCapy

A la hora de decidir el estilo visual de la web, se decide usar colores azulados, los cuales son complementarios con el color de la mascota del logo.

Para elegir el color azul, nos basamos en diferentes webs de enseñanza que usan los tonos azulados, como por ejemplo la web de **CodeCademy** o la anteriormente mencionada **Socratic**.

Usando la aplicación de Adobe Color se elige una paleta que utilizaremos dentro de nuestro proyecto.



Figura 6. Paleta de colores de DBCapy

El último punto que queremos explorar dentro de este apartado es el de la tipografía. Ya que queremos que sea un producto destinado a los informáticos, vamos a escoger varios estilos que sean parecidos a los que nos encontraremos en una terminal. Para un uso general de la web vamos a usar la tipografía *Roboto-Mono*; para apartados que queremos destacar, vamos a usar otra fuente como *Soomch Sans*. Por último, para el nombre de la web que se complementará con el logo, vamos a usar un estilo que sea píxel y atractivo para informáticos. Usaremos el estilo *Upheaval* de la web **Dafont**. El nombre de la web, DBCapy, es un juego de palabras de Data Base Capybara.



Figura 7. Logo de DBCapy

2.4. Elección de la herramienta de Chatbot

El último apartado que vamos a estudiar dentro de nuestros antecedentes se trata del propio componente de chatbot. Hemos visto como referencias el proyecto "[Desarrollo de un agente conversacional para el aprendizaje de SQL](#)" y la aplicación Socratic.

Ambos usan los chatbot y las inteligencias artificiales de formas distintas para cumplir diferentes funcionalidades. Socratic usa las herramientas de inteligencia artificial para poder identificar el contexto de una pregunta de un alumno. Por otro lado, el proyecto de la Universidad de Cádiz utiliza el chatbot para poder comunicarse con los usuarios cuando tengan distintas dudas sobre los ejercicios de SQL.

Vamos a buscar diferentes herramientas disponibles que puedan ser útiles para construir nuestro componente de Chatbot.

La primera referencia que tenemos gracias al proyecto de aprendizaje de SQL es **IBM Watson**. Una herramienta de inteligencia artificial que se utiliza para crear diferentes flujos de diálogos entre usuarios y el chatbot. Al ser nosotros los que vamos a definir las diferentes rutas conversacionales, podemos definir las posibles respuestas.

Muchas otras herramientas de creación de Chatbot funcionan de la misma manera. Por ejemplo, **Chatfuel** y **Botsify** son dos herramientas similares a IBM Watson.

Aunque son herramientas muy buenas, no destacan en el apartado de poder detectar cualquier tipo de mensajes y saber si se tratan de la materia de Bases de Datos. Este es el principal inconveniente para usar IBM Watson (u otras herramientas de chatbot similares) para nuestro proyecto. Nosotros buscamos al igual que en Socratic detectar cualquier pregunta que nos haga el usuario y saber si esa pregunta tiene que ver con la materia de Bases de Datos.

Buscando referencias, encontré varios ejemplos de cómo la nueva tecnología de inteligencia artificial creada por la compañía Open AI, ChatGPT, había sido utilizada en diversos proyectos académicos. Un ejemplo que llamó especialmente mi atención se trata de un [estudio de la universidad de Cornell de una emulación de un pueblo usando la herramienta de ChatGPT](#).

El uso de la web de ChatGPT es gratuito, por tanto, podríamos comprobar la eficacia de esta herramienta con un caso práctico que puede ocurrir dentro de nuestra web.

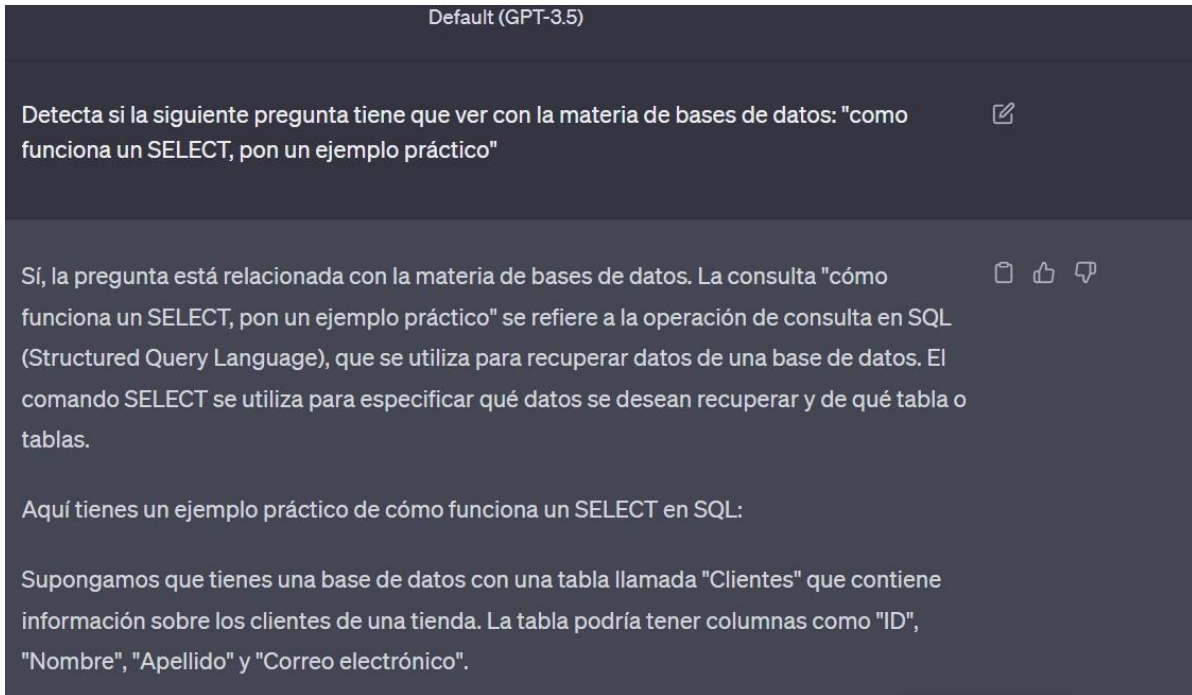


Figura 8. Uso de ChatGPT en una consulta de BBDD

Con el ejemplo que se muestra en la *Figura 8*, Aparte de poder detectar si una pregunta es del temario de Bases de Datos, responde de forma única a la pregunta del usuario. Por estas características de flexibilidad y entendimiento, vamos a usar ChatGPT como nuestra herramienta principal para crear el Chatbot.

El único inconveniente que tiene ChatGPT es que no es completamente gratuito. Para poder usar ChatGPT dentro de una web, se deben hacer solicitudes a la API de dicha web. Cada llamada tiene un coste asociado a la longitud del mensaje y la respuesta que se envía. Para poder resolver este inconveniente y no depender solamente de la herramienta de Open AI.

Una forma de poder gestionar este problema es tener un subsistema que detecte cuando un usuario quiera hacer una llamada a ChatGPT y le permita realizar la pregunta relacionada con la materia de bases de datos. Una buena solución es la creación de una **Red Neuronal** que detecte si un usuario necesita ayuda. Se decide que es preferible la creación de una red neuronal propia en vez de usar herramientas como IBM Watson, aunque son mucho más potentes y fiables. Se toma esta decisión ya que se quiere profundizar en la materia de creación y entendimiento sobre redes neuronales. De esta forma se podrá ver una evolución entre un modelo más clásico y un modelo actual.

Además, un añadido que puede facilitar no tener grandes costes con ChatGPT, es poner diferentes límites a los mensajes que se realizan a dicha API.

3. OBJETIVOS DEL PROYECTO

3.1. Objetivos técnicos

Los principales objetivos del proyecto se han ido planteando de forma abstracta a la hora de ver los antecedentes del proyecto. En este apartado, vamos a fijarlos y verlos de manera mucho más concreta.

- **Implementar ChatBot:** Este proyecto se centrará en la forma de integrar un asistente chatbot dentro del marco del aprendizaje de la asignatura de bases de datos. El asistente se compone de un modelo de red neuronal secuencial y el modelo de ChatGPT. Ya que este es el principal objetivo de nuestro proyecto, se va a dividir en cinco objetivos asociados.
 - **Gestionar llamadas a la API de ChatGPT:** ChatGPT será el principal motor con el que trabajaremos para la función de asistente en el proyecto. Debemos gestionar cada llamada que se hace a la API.
 - **Implementar red neuronal:** Se trabajará con una red neuronal de entendimiento de lenguaje natural que responda órdenes simples de los usuarios. Con la red neuronal se accederá a las llamadas de ChatGPT
 - **Gestionar respuestas:** Se debe gestionar las diferentes respuestas que nos devuelva tanto la Red Neuronal como ChatGPT. Además, mostraremos y guardaremos las interacciones que un usuario tenga con nuestro ChatBot. De esta forma los usuarios pueden tener una experiencia más personal con la web.
 - **Gestionar Prompts para ChatGPT:** Un prompt es la orden que se manda a la API de ChatGPT. Nos serviremos de la Ingeniería de Prompts para diseñar y gestionar sentencias para que la API responda de manera eficiente y correcta. Este concepto se explorará de forma más detallada dentro del siguiente apartado de conceptos teóricos.
 - **Gestionar limitaciones para los usuarios:** Se deberán manejar los diferentes límites que un usuario tiene a la hora de llamar a ChatGPT. Estos límites incluirán el número de llamadas diarias y la longitud de los mensajes. Por último, se crearán diferentes límites dependiendo del estatus de un usuario. Esta es la segunda incorporación que se hace para poder bajar el coste de las llamadas a la API de OpenAI.
- **Incorporar materias de aprendizaje para la asignatura de bases de datos:** Se deberán mostrar tanto material teórico como práctico, los cuales deben estar relacionados entre sí. Como habíamos comentado en el apartado anterior, diferenciar en estos dos apartados nos dará más flexibilidad de separar la web en funcionalidades. Una más centrada en materia teórica y la otra enfocada al planteamiento de ejercicios que puedan resolver los usuarios.
- **Incorporar un sistema de estadísticas:** Al tratar con un tema tan complejo como la enseñanza de una asignatura de informática, debemos incorporar un sistema que nos muestre el feedback de nuestra web. Nos centraremos en buscar opiniones sobre todo de cómo hemos implementado el Chatbot.
- **Gestionar usuarios:** Por último, planteamos un objetivo relacionado con el usuario de la web. El sistema deberá gestionar el registro, la validación y cierre de sesión de un usuario. Además, el registro incluirá la elección de los distintos tipos de usuarios que pueden acceder a la web, esto tendrá efecto dentro del anteriormente mencionado límite de llamadas.

3.2. Objetivos personales

Después de enumerar los distintos objetivos técnicos de nuestro proyecto, vamos a explicar los objetivos puramente personales para el desarrollo del proyecto.

El principal objetivo personal es la motivación de aprender sobre la materia de inteligencias artificiales. Es un campo de la informática de gran interés para el desarrollador. La elección de construir una red neuronal y usar ChatGPT en vez de usar solo usar diagramas de dialogo con la herramienta de IBM Watson se debe completamente a la motivación del programador para profundizar mucho más en este tema tan interesante.

Otro objetivo personal del proyecto es el aprendizaje de nuevos lenguajes de programación y de frameworks. Este planteamiento autodidacta tiene como fin tener una mejor experiencia para un futuro laboral.

4. CONCEPTOS TEÓRICOS

En esta sección vamos a hablar sobre los diferentes conceptos teóricos que están fuertemente fijados dentro de nuestro proyecto. Este apartado tiene mucho que ver con el siguiente apartado de Herramientas usadas en el proyecto. La principal diferencia es que aquí explicaremos conceptos y protocolos que serán luego usados en las Herramientas.

Nos vamos a centrar sobre todo en el entendimiento de las redes neuronales, en concreto, explicaremos la diferencia entre dos modelos enfocados al procesamiento del lenguaje natural. Primero de todo, haremos una explicación de cómo funcionan las redes neuronales, sobre todo enfocados a la red neuronal que vamos a usar dentro de nuestro ChatBot. Luego hablaremos sobre el procesamiento del lenguaje natural, un concepto fundamental para comprender cómo evolucionan los dos modelos que se explican. Y, por último, dentro de la materia de Inteligencias artificiales, hablaremos sobre la novedosa arquitectura Transformer, que es la usada por la inteligencia artificial ChatGPT.

Para concluir, haremos una breve descripción del concepto arquitectónico en el que se basará la web.

4.1. Redes neuronales secuenciales

Una red neuronal secuencial es un tipo de arquitectura de Red Neuronal en el que encontramos una o varias capas ocultas entre la entrada y salida. Todas las capas estarán interconectadas entre sí. El objetivo es poder ingresar datos en la entrada y obtener una respuesta en la salida.

Para poder entender cómo con solo una entrada, una salida y un conjunto de capas intermedias se obtiene resultados como por ejemplo una clasificación, debemos entender los siguientes conceptos:

- **Ponderación y suma ponderada:** Cada neurona toma un valor de entrada, este dato es multiplicado por un valor conocido como el Peso. Después, realizará una suma con todos los datos de entrada ponderados (ya multiplicados por el Peso).
- **Función de Activación:** Con el valor obtenido de la suma ponderada, vamos a ingresar dicho dato en una función de activación. Esta función no es única, y dependiendo de la función de activación seleccionada se obtendrán unos resultados u otros. La función de activación agrega una mayor complejidad a las redes neuronales y a los problemas que pueden resolver. Vamos a explicar las funciones de activación que se usaron dentro de nuestra red neuronal:
 - o **Relu:** La función Relu es de la más sencillas y de las más usadas a la hora de crear una red neuronal. Consiste en anular todos los valores que sean negativos, y el dejar los valores positivos con su valor tal como entran.
 - o **SoftMax:** Es una función con un mayor coste computacional que Relu, por tanto, se reserva sobre todo a las últimas capas de la red neuronal. Su objetivo es pasar una serie de valores numéricos en una distribución de probabilidades. Por tanto, es muy usada sobre todo en las tareas como clasificación (lo cual nos viene perfecto para decidir el tipo de mensaje que se manda de un usuario).

- **Propagación hacia delante:** Los resultados de las funciones de activación se mandan a las neuronas de las capas siguientes.

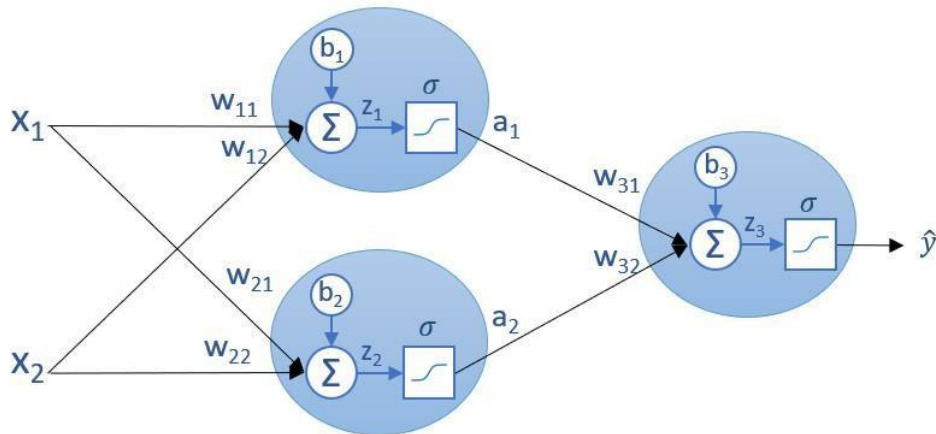


Figura 9. Representación de una propagación hacia delante en una RNA

- **Capa de salida:** La capa de salida tiene como objetivo hacer predicción con todos los valores que hemos obtenido de las capas interiores ocultas. Por ejemplo, como se explicó en el apartado de SoftMax, el resultado de nuestra red neuronal será el de clasificar el tema de un mensaje que ingrese un usuario.
- **Ajuste de pesos:** Durante el entrenamiento de la red neuronal se comparan los resultados obtenidos con las respuestas reales. De esta forma se calcula una media de error, la cual se aplica a ajustar los pesos de las neuronas para poder bajar dicho error. Aunque el concepto parece sencillo, involucra una gran complejidad matemática y de algoritmos que se pueden aplicar. Centraré en explicar solo los dos algoritmos se han utilizado para compilar la red neuronal.
 - **Función de pérdida Categorical Crossentropy:** Como hemos mencionado anteriormente, la pérdida es la función que muestra como de reales son las predicciones que se hacen. Con la función *Categorical Crossentropy* nos centramos en el problema de clasificación múltiple. Se centra en la diferencia entre las probabilidades creadas por la red neuronal y las soluciones con las que estamos entrenando dicha red.
 - **SGD (Gradiente Descendente Estocástico):** Para ajustar los pesos de la red neuronal, se centra en ajustar dichos pesos en dirección opuesta al gradiente de la función de pérdida (buscamos el punto mínimo dentro de nuestra función de pérdida para poder hacer un cambio en los pesos).

Ya con esta pequeña explicación podemos comprender cómo funciona una red neuronal secuencial, y también comprendemos cual es el objetivo de usar una red neuronal dentro de nuestro Chatbot. Lo siguiente es centrarnos en un aspecto fundamental para las inteligencias artificiales usadas en el proyecto, el procesamiento del lenguaje natural.

4.2. Procesamiento del Lenguaje Natural (NLP) y modelo Bag of Words

El siguiente punto para tratar es uno de los campos más importantes para los que han sido creadas las redes neuronales. La comprensión y la creación de un lenguaje natural.

Este campo es muy amplio y complejo, además han surgido grandes avances en la última década. En estos últimos años ha habido una explosión de herramientas de NLP. Pero nosotros nos centraremos en los conceptos más básicos teóricos para poder comprender NLP.

Para comenzar, nuestra red neuronal necesita pasar de alguna forma una frase de un usuario a un dato numérico. Para empezar, lo primero que debemos hacer con esta frase es dividirla en partes. Cada una de estas partes será conocida como **TOKEN**. Un token puede ser tanto una frase, una palabra, unos pocos caracteres, etc. La división de tokens depende siempre del modelo que estemos utilizando.

Lo siguiente es poder reducir nuestras palabras en estructuras más sencillas. Este es un aspecto puramente lingüístico, donde entran conceptos como buscar las raíces de una palabra o su **lematización** (buscar los lemas de las palabras, ósea sus formas básicas). La lematización es un aspecto muy importante en NLP dentro de nuestra red neuronal, ya que al simplificar las palabras podemos poner al mismo nivel un conjunto de palabras que tienen un significado parecido. Un ejemplo de lematización es por ejemplo de “niños” su palabra lematizada “niño”.

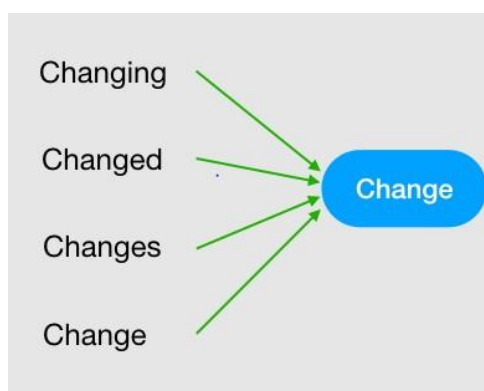


Figura 10. Ejemplo de lematización de la palabra "Change"

Ya en este punto en el que tenemos una frase tokenizada y posteriormente lematizada podemos tomar diferentes alternativas para que nuestra red neuronal la comprenda. Primero de todo vamos a explicar la forma en la que nuestra red neuronal funciona, para poder compararlo luego con un modelo mucho más sofisticado como es el que utiliza ChatGPT.

El primer enfoque es el de entender que un tema tiene una serie de palabras asociadas. Nuestro objetivo es clasificar una frase dentro de un tema teniendo en cuenta dichas palabras claves y las veces que se repitan. Este modelo es conocido como **Bag of Words**, y es el modelo que se ha aplicado dentro de nuestra red neuronal.

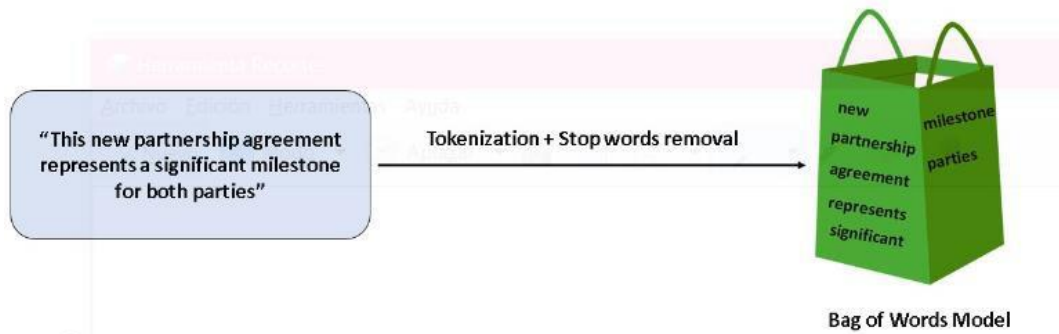


Figura 11. Representación gráfica del modelo Bag of Words

La idea clave de este modelo es crear un vector que mida las veces que se repita una palabra dentro de una frase. De esta forma creamos las palabras claves asociadas a un tema. Cada frase se comparará con el vector conocido como Bag of Words(vector con todas las palabras de nuestro vocabulario).

De esta forma, podemos entrenar a nuestra red neuronal para que entienda como por ejemplo las frases "Hello", "Hello, how you doing", "Hello, nice to meet you" y asociar la palabra Hello a el tema de "Saludos".

	she	loves	pizza	is	delicious	a	good	person	people	are	the	best
She loves pizza, pizza is delicious	1	1	2	1	1	0	0	0	0	0	0	0
She is a good person	1	0	0	1	0	1	1	1	0	0	0	0
good people are the best	0	0	0	0	0	0	1	0	1	1	1	1

Figura 12. Vector bag of words

Estos conjuntos de datos son conocidos como **Corpus**, los cuales engloban grandes cantidades de frases y temas para poder de esta forma entrenar nuestras redes neuronales de forma más amplia.

Aunque este modelo puede ser bueno para clasificar distintos temas, tiene un gran problema, no se fija en el significado de las frases. Solo nos estamos enfocando en la probabilidad de que una palabra puede aparecer dentro de un tema en concreto. Además, nosotros somos los que fijamos los temas, convirtiéndolo en un sistema muy estático y difícil de escalar. Cada vez que ampliamos nuestro vocabulario, más grande se convierte el Bag of Words y más difícil de computar.

Este modelo es muy poco sofisticado, sobre todo si necesitamos tareas mucho más complejas como el entendimiento y la generación de textos. Para poder solucionar este problema, vamos a usar un modelo mucho más evolucionado conocido como los **Transformers**. Dicho modelo evoluciona el modelo anterior teniendo en cuenta el significado de cada palabra y asociando un peso a la posición en la que se encuentra la palabra dentro de una frase.

4.3. Embedding y la evolución del modelo Transformer

Ya hemos hablado de NLP y como dependiendo del modelo vamos a tratar las frases de una forma u otra. Después de ver el modelo de Bag of Words, vamos a explicar el concepto de Embedding y el modelo arquitectónico de los Transformers.

Con el concepto de **Embedding** vamos a evolucionar la idea de asociar una palabra a una representación numérica. Anteriormente asociábamos una palabra dentro de un vector con todo el vocabulario. No nos fijábamos en el propio significado de cada palabra. Ahora con el Embedding vamos a crear relaciones entre las diferentes palabras, sobre todo intentaremos asociar palabras dentro de diferentes campos semánticos. Lo que queremos es comprimir nuestro vocabulario creando un mapa en el que podamos relacionar una palabra dentro de un contexto. Un Embedding se crea al crear relaciones vectoriales entre distintas palabras y formadas por un entrenamiento con una red neuronal.

Dicho de otro modo, buscamos que una palabra tenga un valor numérico que pueda mandarse a nuestra red neuronal. Además, queremos que este valor numérico esté asociado a un contexto semántico.

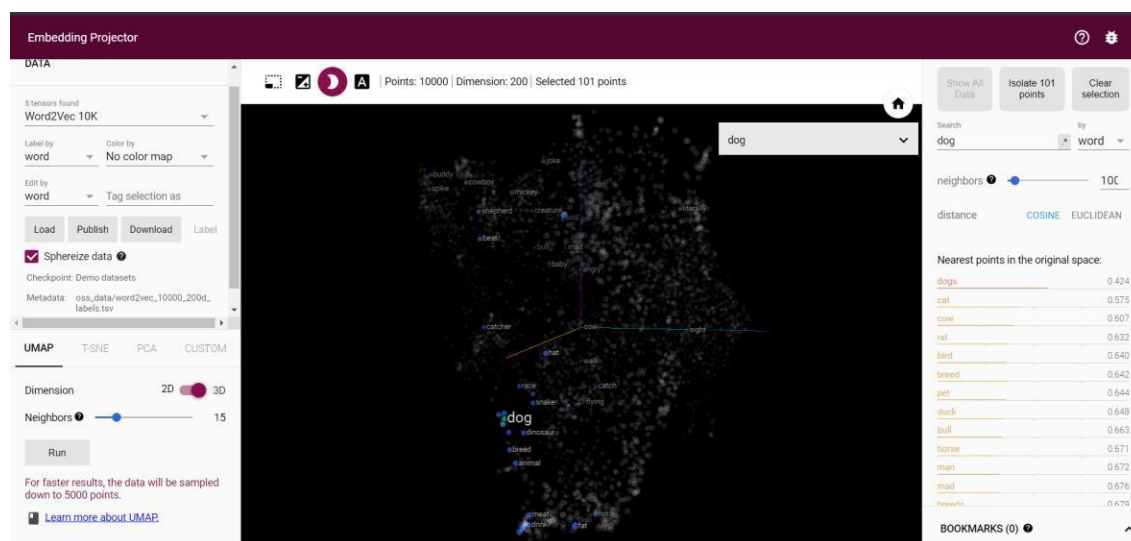


Figura 13. Relación en contexto semántico de la palabra "perro" con otras palabras

En esta imagen podemos observar como la palabra perro está relacionada con otras palabras de un mismo contexto semántico. Para poder agruparlas se mide su distancia dentro de este espacio vectorial que hemos creado.

Ya entendiendo como las palabras pueden ser representadas con su contexto semántico, vamos a enfrentarnos a otro problema para poder comprender el significado de las palabras.

Una palabra dentro de una frase tiene además de su significado propio, un significado en la posición en la que se encuentra. Aplicando este concepto de dar un valor a la posición de una palabra, los modelos arquitectónicos de redes neuronales **Transformers**, van a poder no solo entender el significado de las palabras, sino también el significado y contexto de una frase.

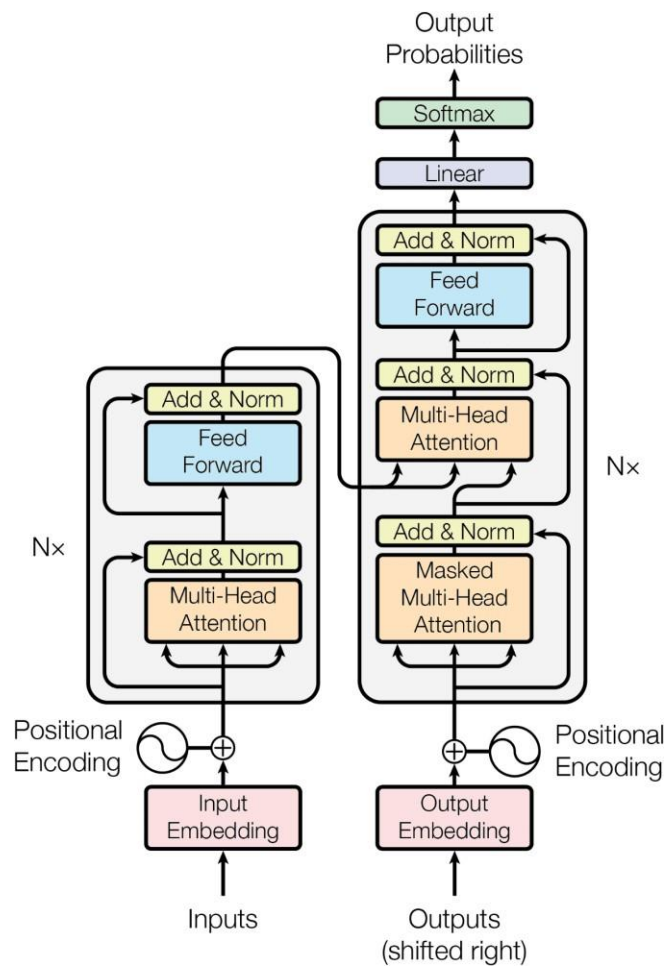


Figura 14. Representación de un modelo Transformer

Un modelo Transformer sigue siendo un modelo de red neuronal, solo que se han aplicado todos los conceptos de NLP anteriores como el Embedding, Positional Encoding (valor de posicionamiento a cada palabra), Tokenización de las palabras, etc.

Además, vamos a destacar un elemento fundamental para que este modelo sea tan eficiente y sea usado en aplicaciones como ChatGPT. El elemento **Multi-Head Attention**. Este elemento va a fijarse en las palabras y en la posición en la que se encuentran, creando así patrones y mapas en los que las palabras se relacionan entre sí. Esta función se produce entre varias palabras de forma paralela y en diferentes posiciones dentro de la arquitectura, por eso el nombre de Multi-Head.

Con este modelo de Red Neuronal, no solo podemos procesar textos, sino entrenarlos de forma que puedan comprender el contexto y la forma en la que el lenguaje natural funciona.

4.4. Arquitectura Cliente-Servidor

Este estilo de arquitectura de Software es uno de los más usados hoy en día. Esta arquitectura hace una separación en dos componentes dentro del sistema: Clientes (los cuales hacen peticiones) y Servidor (el que responde a dichas peticiones) y es de gran utilidad para sistemas distribuidos.

Estas dos partes trabajan juntas para proporcionar servicios, recursos e información a los usuarios.

Primero tenemos los Clientes, los cuales son las aplicaciones que interactúan con nuestros usuarios y hacen las peticiones a los Servicios.

En el caso de nuestra aplicación web, tenemos Clientes que solicitan al Servicio de ChatGPT con distintos mensajes. Nuestro servicio web (Cliente) interpretará todos los mensajes que nos responda el Servidor de ChatGPT para luego mostrarlos al usuario.

Por otra parte, tenemos al Servidor, que tiene como objetivo resolver las diferentes peticiones que se realizan por parte del Cliente.

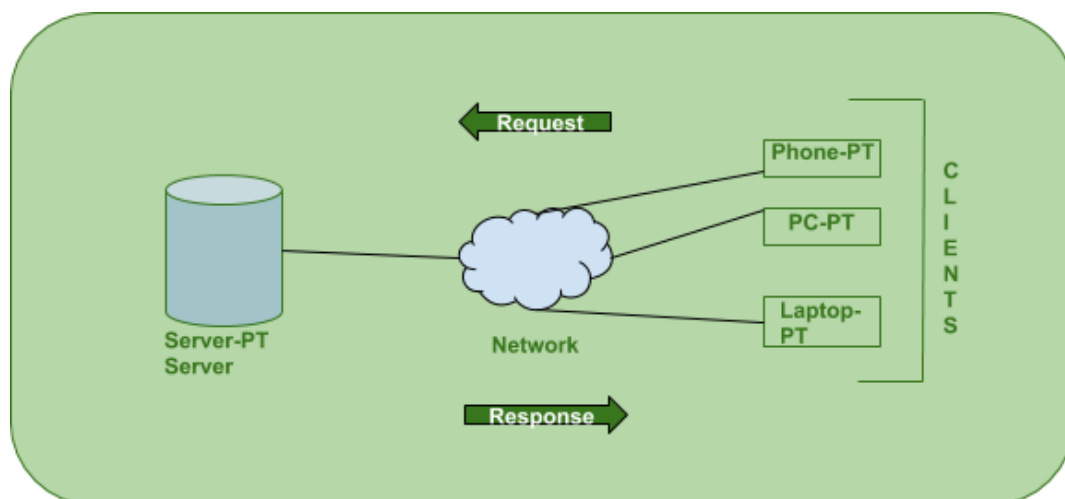


Figura 15. Ejemplo arquitectura Cliente-Servidor

5. METODOLOGÍA, TÉCNICAS Y HERRAMIENTAS

En este apartado vamos a hablar de los siguientes temas: Primero sobre la metodología usada en este proyecto, que se trata del proceso unificado. Segundo las diferentes técnicas usadas como patrones y los estilos arquitectónicos seguidos. Por último, se enumerarán y explicarán las distintas herramientas utilizadas. Dentro de este apartado, toma especial importancia como hemos usado la API de ChatGPT y se hará una explicación de la Ingeniería de Prompts.

5.1. Metodología

La metodología con la que trabajaremos será el modelo de Proceso Unificado. Esta metodología permite identificar, catalogar y planificar los distintos casos de uso dentro de nuestro sistema. Gracias a esta metodología, podremos seguir una estructura y un mapa que nos ayude con el desarrollo.

Debemos tener en cuenta que el Proceso Unificado se basa en cuatro etapas que se corresponden con la vida y desarrollo de un proyecto: Inicio, Elaboración, Construcción y Transición. Cada una de estas se encuentra dividida en diferentes iteraciones. Una iteración está compuesta por tareas. La división de tareas se organizará en seis clases distintas: Modelo de negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas. De esta forma podremos diferenciar tareas más enfocadas a la investigación de tareas como la integración de una nueva parte a nuestro proyecto. Dependiendo de la etapa del proyecto, habrá más o menos peso en las distintas clases de tareas.

En el caso concreto del sistema web que estamos desarrollando, dividiremos todo el proyecto en 5 iteraciones, las cuales van a reflejar las cuatro etapas de desarrollo de un proyecto con el Proceso Unificado.

- **Iteración 1:** La primera etapa se hace una gran inversión en la investigación y en decidir hacia qué rumbo va a ir el proyecto. Al tratarse de una etapa muy corta, prácticamente solo nos centraremos en la investigación, apenas se dedicará tiempo hacia la Implementación, Pruebas y Diseño.

- **Iteración 2:** En esta segunda iteración nos centramos sobre todo en los primeros pasos para desarrollar la idea de nuestro chatbot. Todavía es una etapa en la que no hay tanto peso en cuanto a Implementación y Pruebas. Tiene más peso tareas relacionadas con los Requisitos y el Análisis.

- **Iteración 3:** En esta tercera iteración ya habrá un mayor peso a la hora de Implementar y Probar nuestro bot. Nos centramos solo en las funciones del apartado de teoría, ya que son muy similares a las del apartado de práctica. Sobre todo, queremos un chatbot mucho más refinado y a lo largo de esta etapa vemos cómo evolucionan los dos subsistemas que lo involucran (la red neuronal y las llamadas a ChatGPT). También en esta iteración, vamos a dedicarnos a las primeras etapas de Requisitos y Análisis de los demás sistemas que componen la web.

- **Iteración 4:** En esta cuarta iteración se centrará por completo en funcionalidades dentro de la Implementación y Pruebas. El resto de las funciones no tienen tanto protagonismo. Vamos a trabajar en el resto de los subsistemas de nuestro proyecto para integrarlo junto al Chatbot ya funcional.

- **Iteración 5:** Nuestra última iteración es la etapa final del proyecto. Estudiaremos los problemas que nuestro producto final tiene. También incluiremos los sistemas de estadísticas para que los usuarios puedan darnos opiniones sobre el sistema final. En conclusión, vamos a tener una etapa corta que se centrará casi por completo en los riesgos existentes del producto final.

Después de cada iteración, tendremos los hitos, los cuales nos permiten marcar los finales de cada iteración y poder hacer una buena planificación de nuestro proyecto.

Ya con esta explicación de cómo hemos aplicado la metodología, podremos mostrar todas las tareas que componen el desarrollo de nuestro proyecto. Para representar estas tareas y sus relaciones, se usa una gráfica de Gantt.

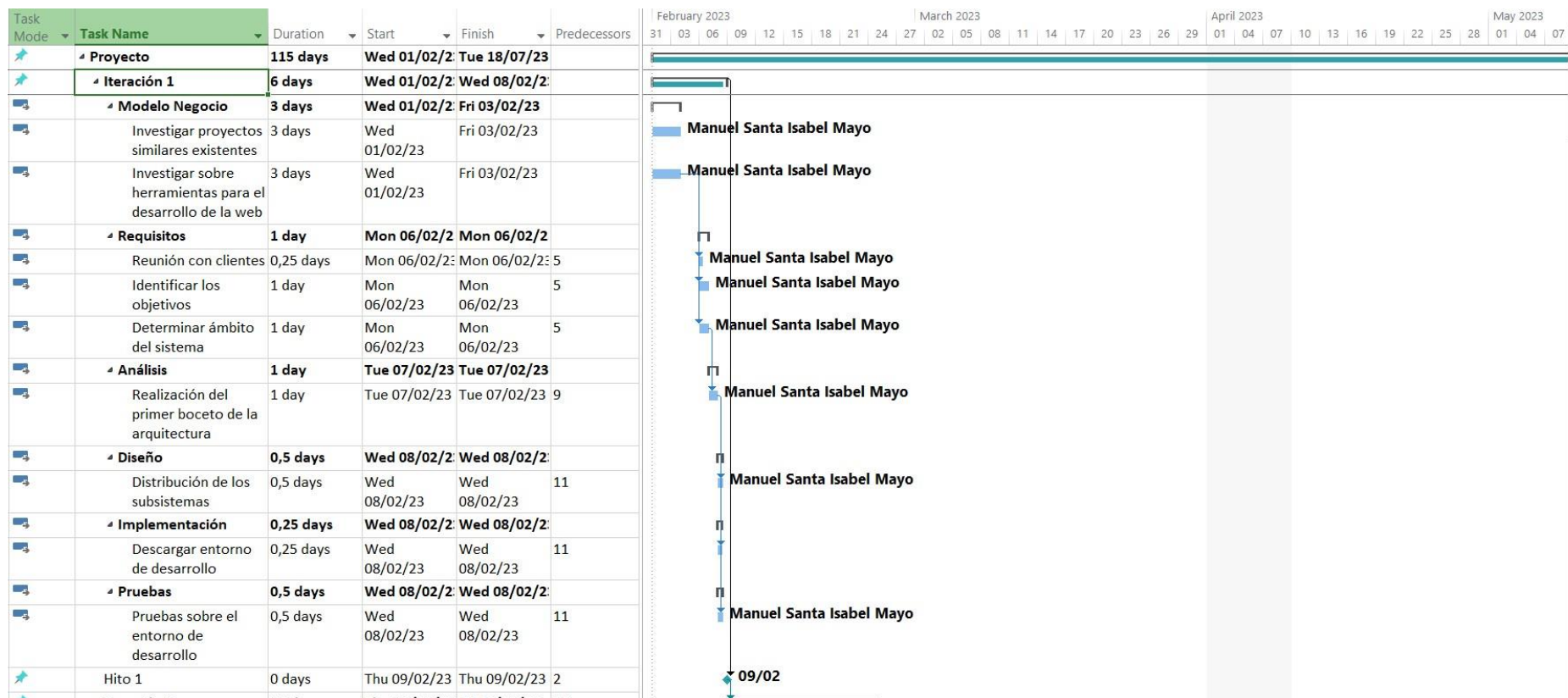


Figura 16. Diagrama de Gantt 1

Manuel Santa Isabel Mayo

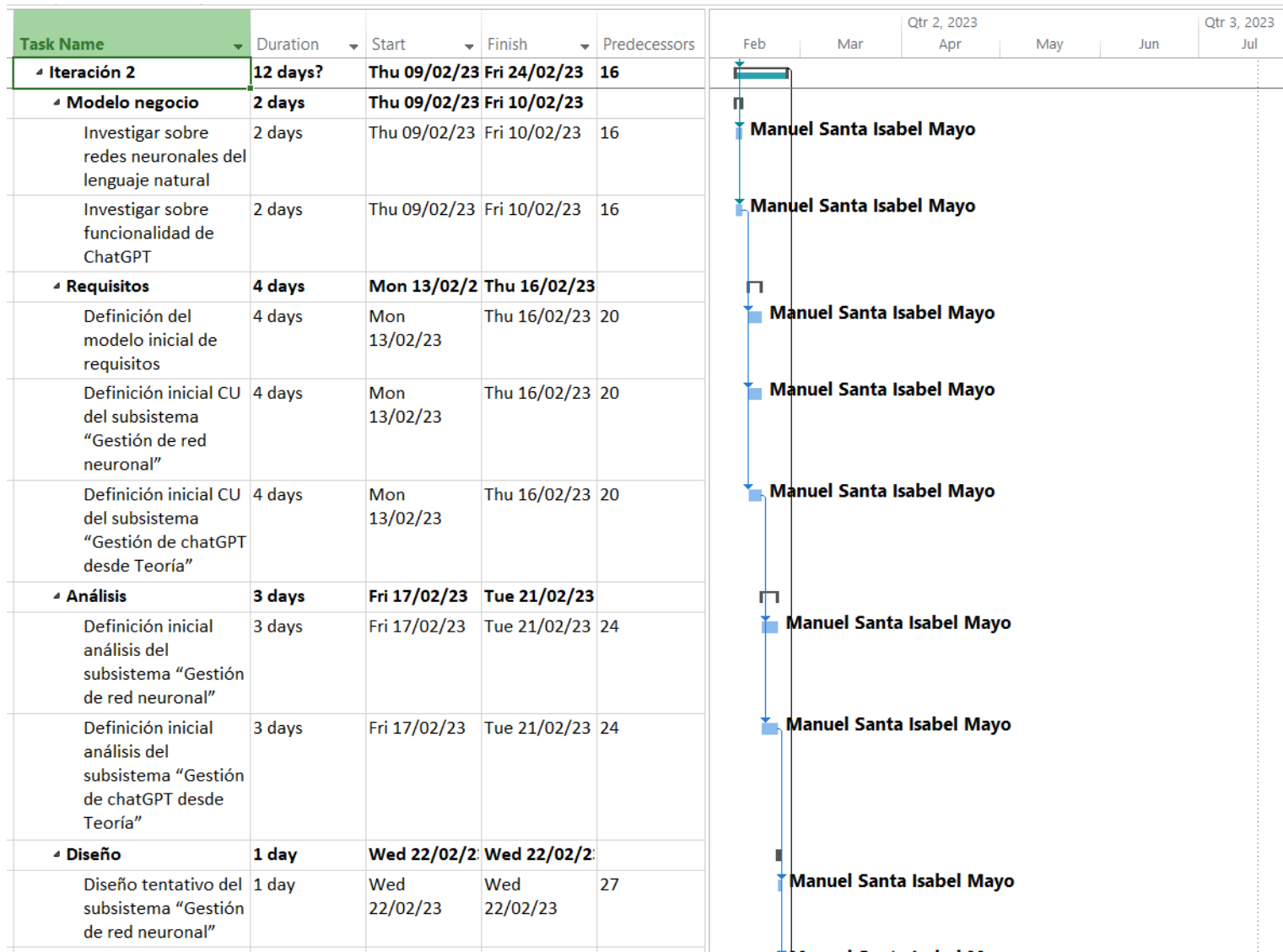


Figura 17. Diagrama de Gantt 2

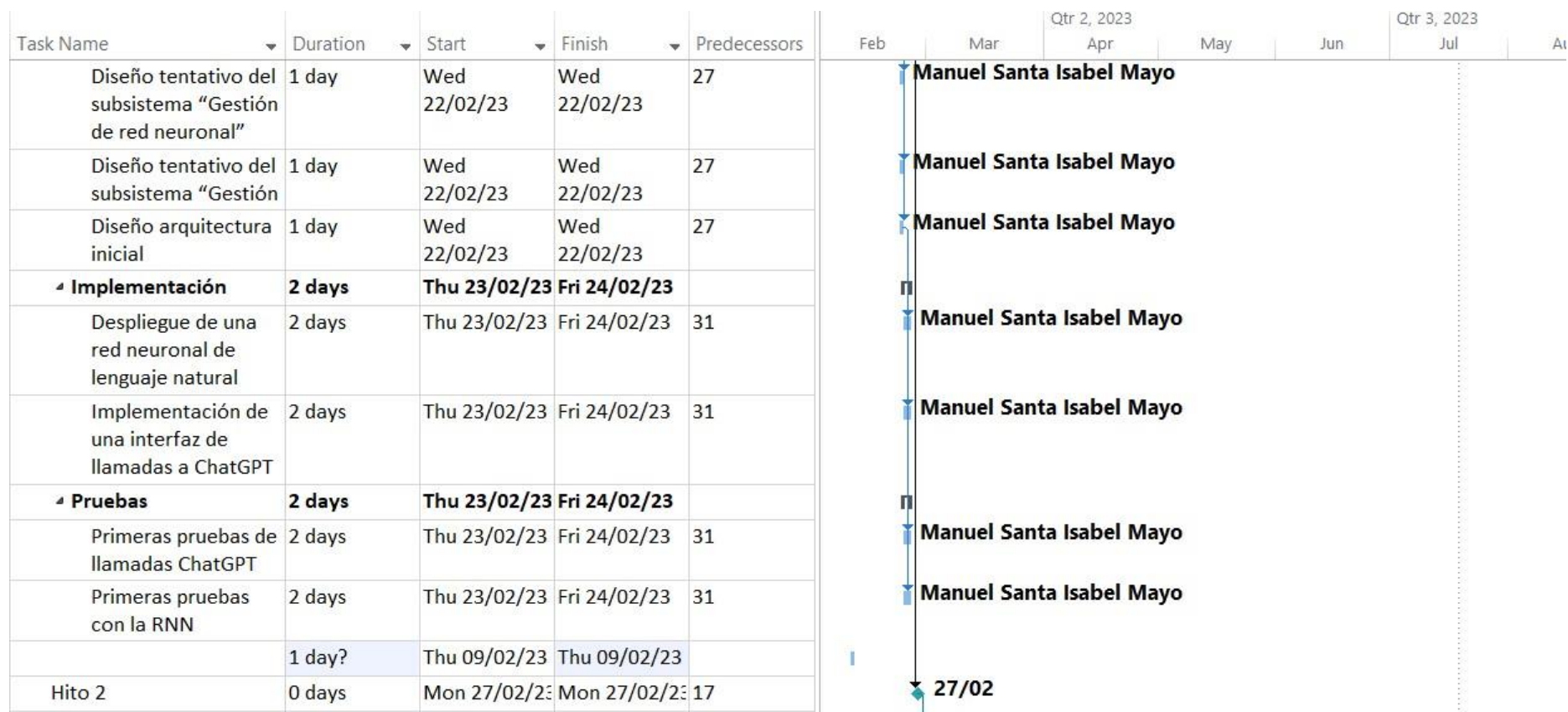


Figura 18. Diagrama de Gantt 3

Manuel Santa Isabel Mayo

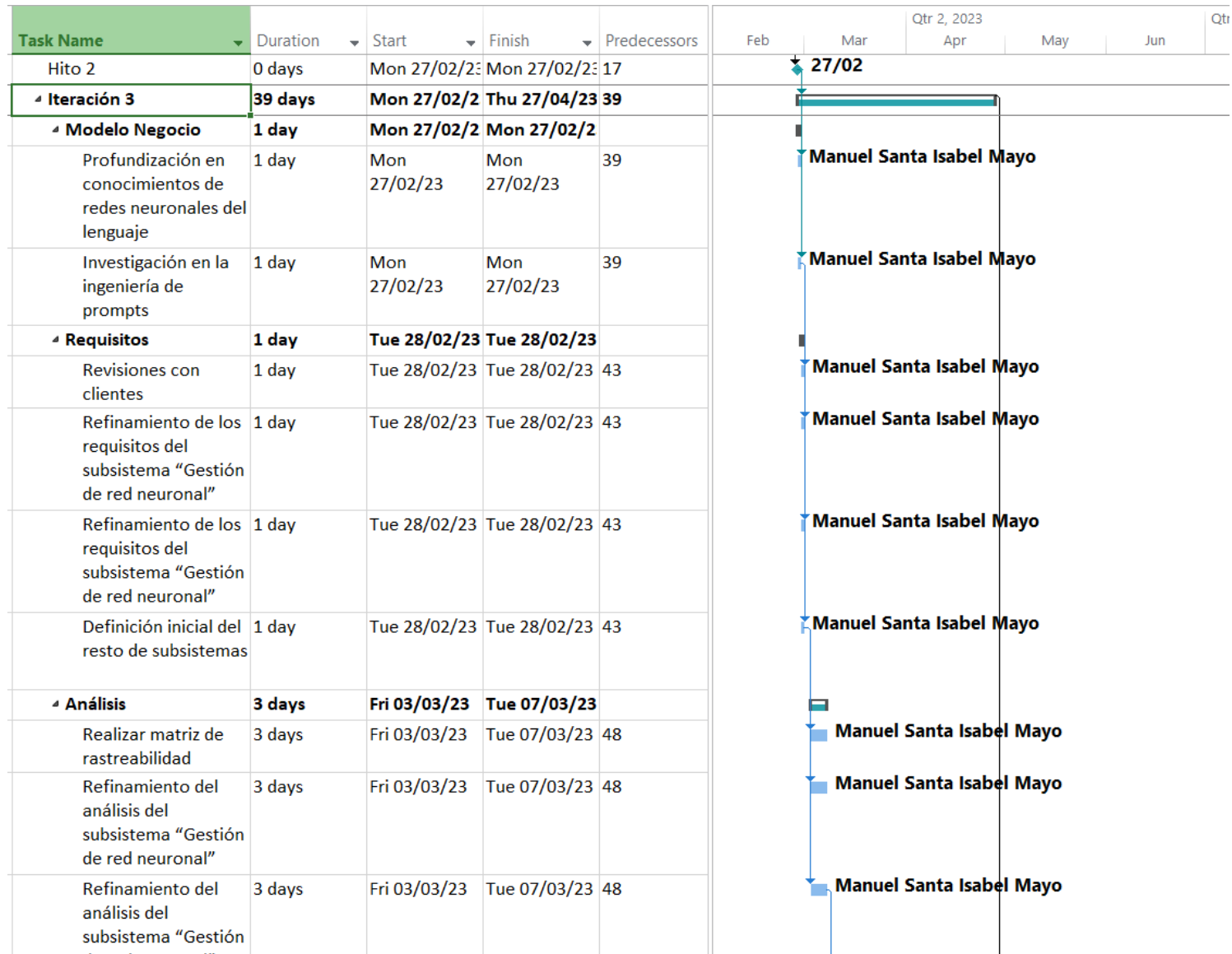


Figura 19. Diagrama de Gantt 4

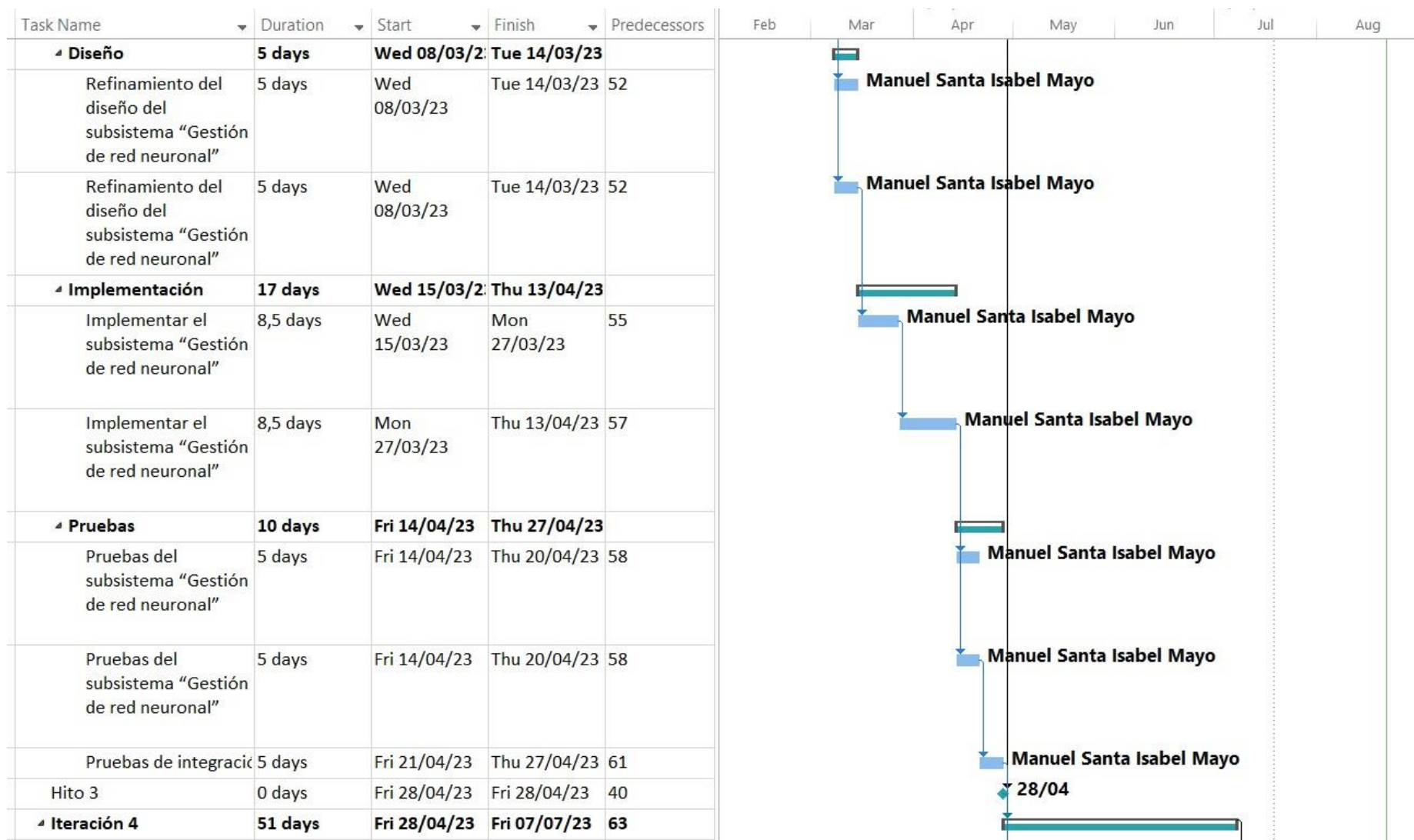


Figura 20. Diagrama de Gantt 5

Manuel Santa Isabel Mayo

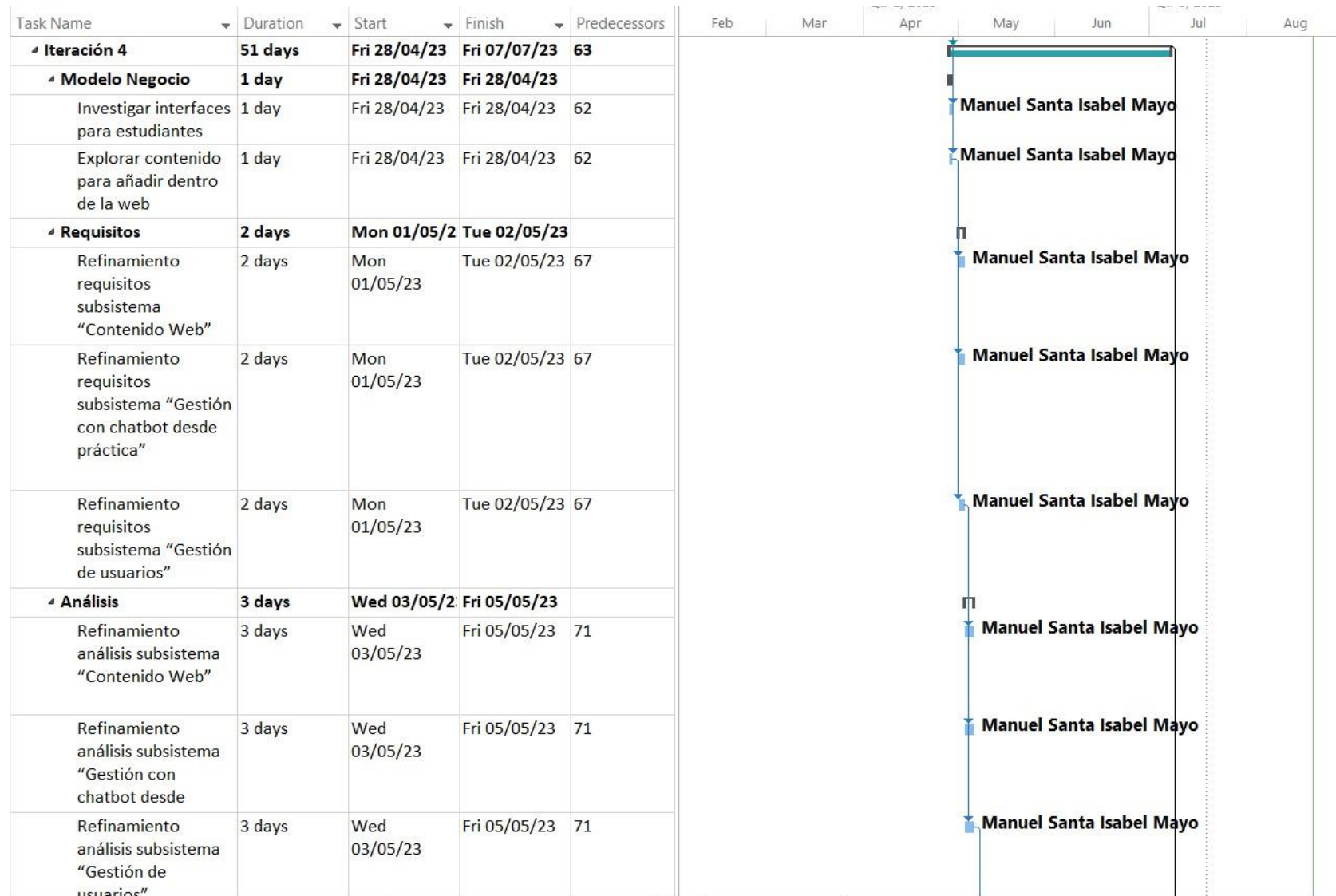


Figura 21. Diagrama de Gantt 6

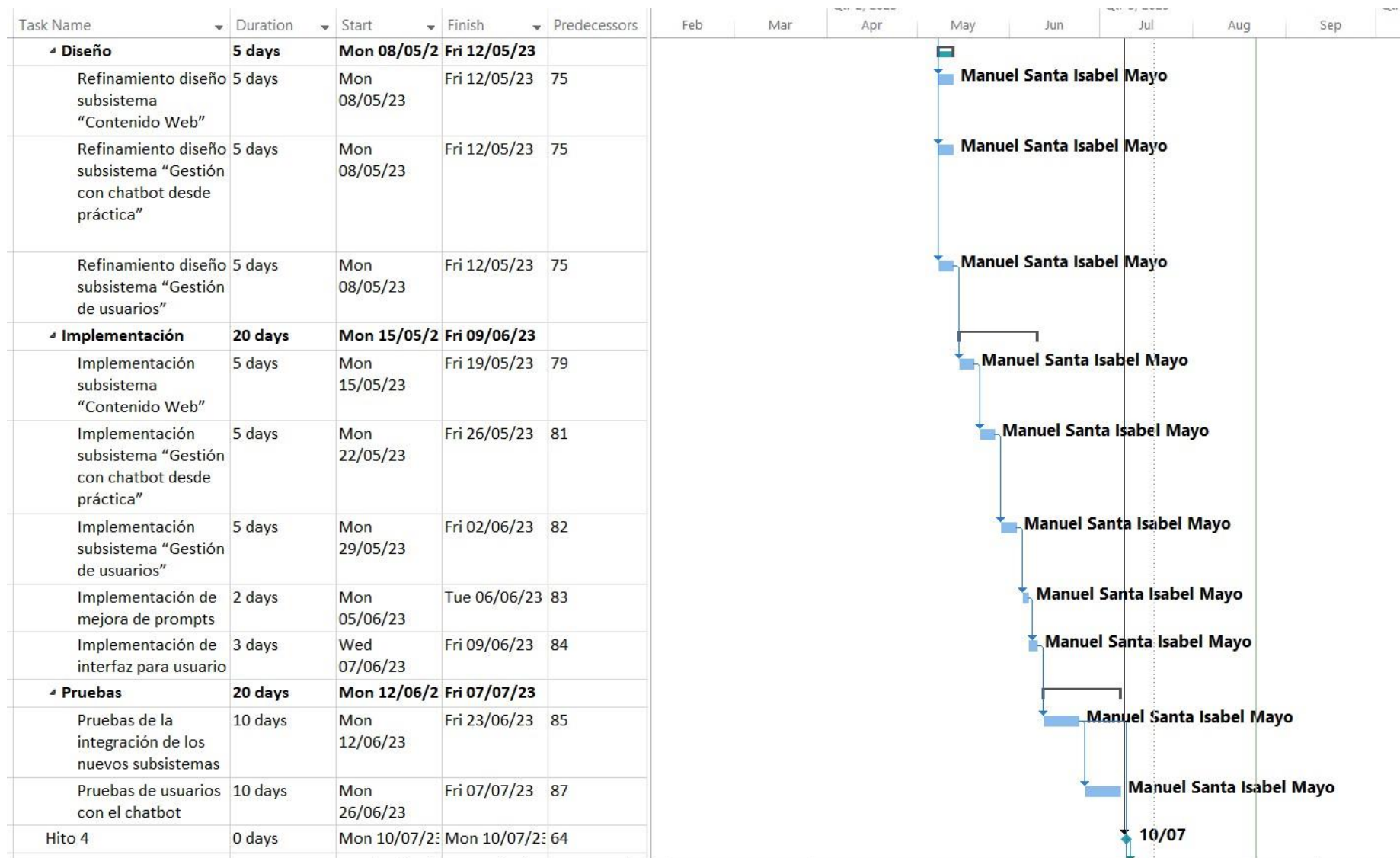


Figura 22. Diagrama de Gantt 7

Manuel Santa Isabel Mayo

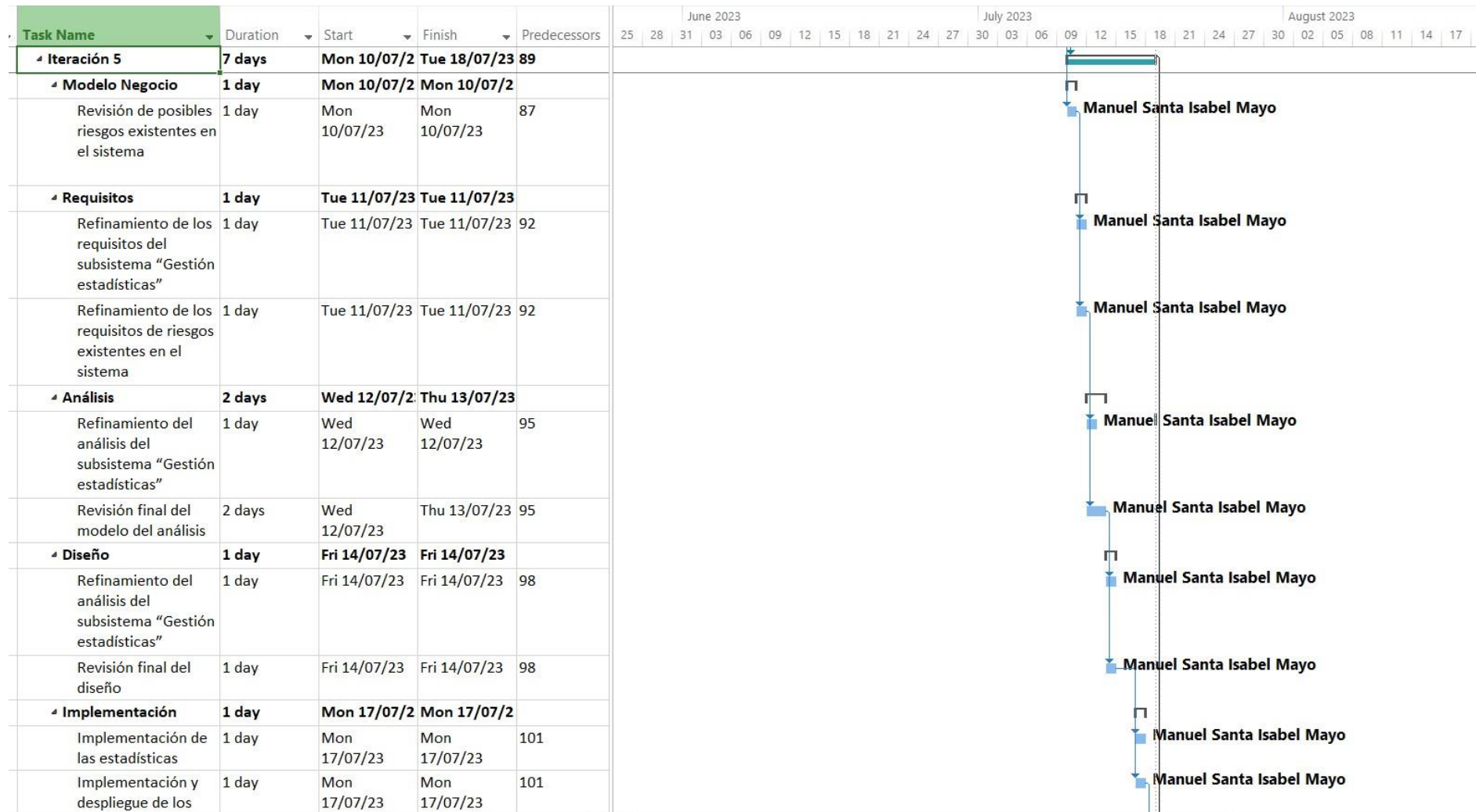


Figura 23. Diagrama de Gantt 8

Implementación	1 day	Mon 17/07/23	Mon 17/07/23	
Implementación de las estadísticas	1 day	Mon 17/07/23	Mon 17/07/23	101
Implementación y despliegue de los cambios	1 day	Mon 17/07/23	Mon 17/07/23	101
Pruebas	1 day	Tue 18/07/23	Tue 18/07/23	
Ultimas pruebas sobr	1 day	Tue 18/07/23	Tue 18/07/23	104
Hito Final	0 days	Tue 18/07/23	Tue 18/07/23	90

Figura 24. Diagrama de Gantt 9



5.2. Técnicas

5.2.1. Backend - Frontend

La principal técnica que se usó para crear nuestro servicio es el de Backend Frontend. Su estructura es muy similar a la de Cliente Servidor, pero nos servirá para explicar mejor las herramientas usadas dentro de nuestra Web.

Para empezar, tendremos una estructura que interactúa directamente con nuestro cliente, el Frontend. Dicha estructura está construida con la librería React JS, la cual interactúa con nuestro usuario mostrando una interfaz para la web, enseñando el ChatBot, y administrando diversos datos internos (como por ejemplo para comprobar la autenticación de los usuarios).

Podríamos definirlo como una gran estructura en la cual se juntan tantos componentes de Vista Modelo y Controlador. Esto se debe a que se fusionan tanto código HTML, CSS y JavaScript (con diferentes modificaciones) Otra de las funciones que cumple el Frontend son las llamadas a ChatGPT.

Lo siguiente que tendremos será el Backend, que está construido con la herramienta de Django. El Backend funcionará como un servidor Rest (gracias al framework Django Rest), el cual solo acepta peticiones de HTTP para comunicarse.

Nuestro Backend de Django también se comunicará con una base de datos, en la que guardaremos datos como los usuarios, los mensajes que escriben y las distintas peticiones que realizan a ChatGPT.

Para comunicarse el Backend y el Frontend, se realizarán llamadas HTTP con la herramienta AXIOS de ReactJS.

Esta estructura es muy similar a la arquitectura Cliente Servidor, no obstante, nuestro sistema no está distribuido en diferentes servidores. En este caso el Frontend se alojará dentro del mismo servidor que el Backend. Para ello simplemente se crea una versión Build del Frontend para almacenar la dentro del servidor de Django.

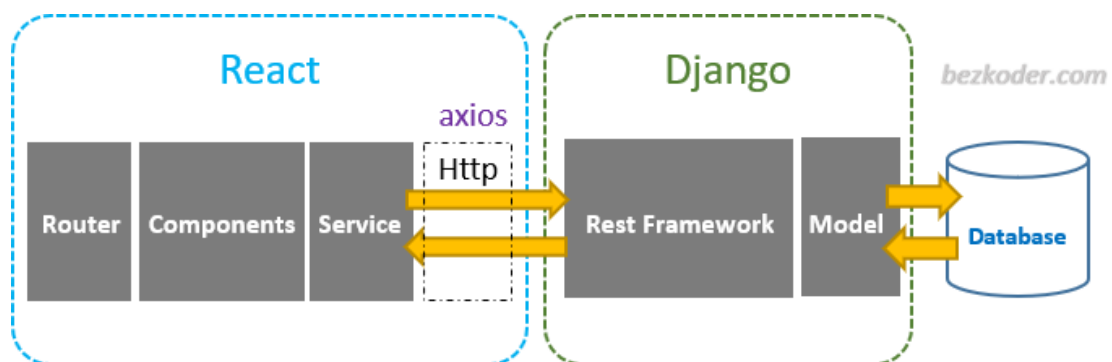


Figura 25. Ejemplo de Backend-Frontend con Django y React

5.2.2. Método de Durán y Bernárdez

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de requisitos de la fase de ingeniería de requisitos del desarrollo del software.

Como se ve en la imagen adjuntada a continuación, se puede observar las distintas partes de esta metodología

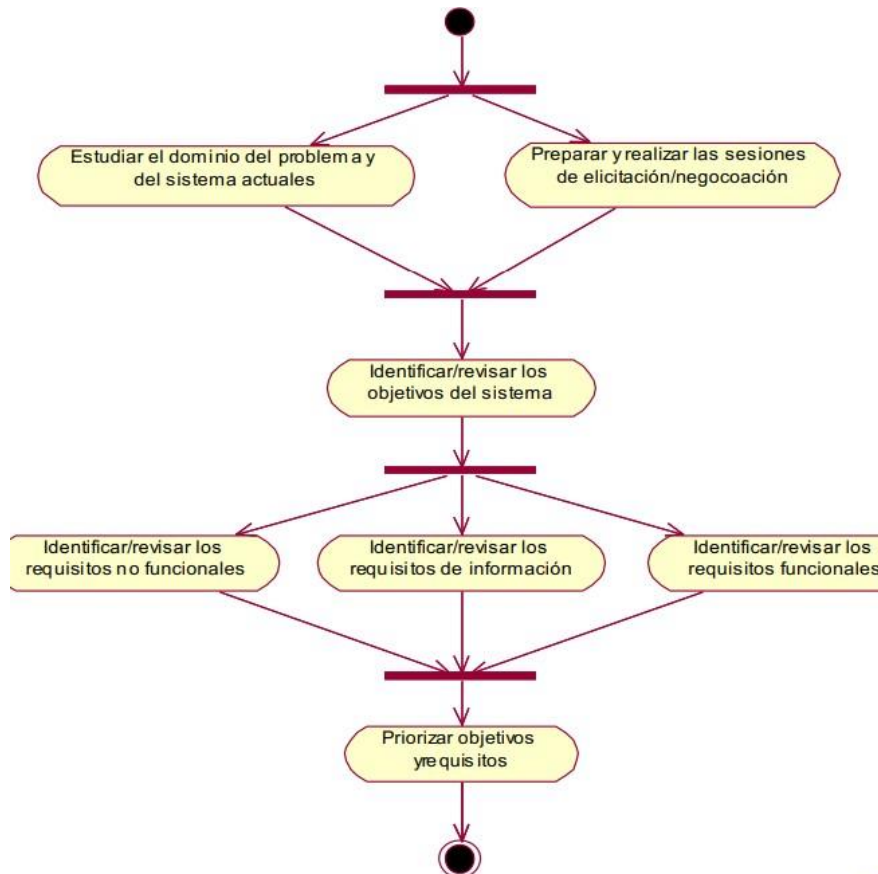


Figura 26. Diferentes etapas del método de Durán y Bernárdez

5.3. Herramientas

Ya habiendo visto todas las metodologías y técnicas que vamos a usar para desarrollar en nuestro proyecto, vamos a ver las herramientas que usaremos para poder construir la web. Vamos a centrarnos en tres apartados muy importantes: Herramientas usadas para crear el ChatBot, Herramientas para crear el entorno de la web y Herramientas secundarias usadas durante el proyecto.

5.3.1. Herramientas usadas para crear el Chatbot

Este primer grupo de herramientas está compuesto principalmente por dos grandes programas: TensorFlow y ChatGPT.

5.3.1.1. TensorFlow

TensorFlow es una biblioteca de código abierto creada por Google para el desarrollo y creación de modelos de redes neuronales. Para usar esta herramienta programaremos en el lenguaje de programación Python. En el caso de nuestro proyecto, usaremos TensorFlow para crear nuestro sistema que entrene una red neuronal con el modelo de Bag of Words.

Para poder crear dicho modelo, primero debemos crear un fichero con los datos con los que nuestra red neuronal pueda entrenarse. Este fichero estará compuesto por diferentes frases asociadas a un tema. También incluiremos las distintas respuestas que queremos que la red neuronal devuelva dependiendo del tema que hayamos determinado.

```
{
  "tag": "commands",
  "patterns": ["What commands can I use with this virtual agent?",
    "What are the basic voice commands that I can use with this chatbot?",
    "Can you guide me on how to interact with this virtual agent using commands?",
    "What are the most commonly used commands for this chatbot?",
    "Can you tell me the list of voice commands available for this virtual agent?",
    "How can I activate the voice recognition feature to use commands with this agent?",
    "What are the different ways I can use commands to communicate with this virtual assistant?",
    "What specific actions can I perform with the commands on this chatbot?",
    "How can I learn more about the commands and their functionalities on this virtual agent?",
    "Can you explain the syntax and structure of the commands that I can use with this chatbot?",
    "What are the limitations of using voice commands with this virtual assistant?",
    "How can I customize the commands to suit my needs with this chatbot?",
    "What kind of feedback or responses can I expect when using commands with this virtual agent?",
    "Can I use natural language commands with this chatbot?",
    "What are the security measures in place for using commands with this virtual assistant?",
    "Can I use specific keywords or phrases to trigger different actions with this chatbot?",
    "How can I see a list of all the available commands for this virtual agent?",
    "What kind of user data is collected when using commands with this chatbot?",
    "Can I use commands to navigate to different sections or pages on this virtual agent?",
    "What are the advanced commands that I can use with this chatbot?"
  ],
  "responses": [
    "Claro, los comandos más útiles que puedes usar son los siguientes: help | hello | what's your name? | what time is it? | good bye | who is your creator? | how to use"
  ]
},
{
  "tag": "bot",
  "patterns": [
    "What are chat bots used for on this website?"
  ]
}
```

Figura 27. Fichero con datos de entrenamiento para TensorFlow

Aunque TensorFlow incluye un gran número de herramientas disponibles para poder crear la red neuronal, usaremos más herramientas para poder desarrollar el procesamiento del lenguaje natural.

5.3.1.1.1. NLTK WordNetLemmatizer

Herramienta dentro de la biblioteca de NLTK (Natural Language ToolKit) que nos sirve para poder Lematizar palabras. Esta herramienta usa el recurso léxico WordNet, que es una base de datos léxicos del inglés.

Es importante recalcar que se ha utilizado una herramienta que solo procesa vocabulario en inglés. Esto se debe a que la herramienta está mucho más desarrollada y funciona mucho mejor que otras alternativas con el idioma castellano. A la hora de desarrollar la red neuronal se probaron alternativas que pudieran lematizar palabras con el idioma castellano como FreeLing, pero debido al gran número de fallos para integrar dicha herramienta dentro del proyecoto se decidió solo usar una lematización del inglés.

5.3.1.1.2. NLTK Word_Tokenize

Herramienta incluida en la misma biblioteca de NTLK que nos permite Tokenizar frases en formatos más sencillos. Podemos tokenizar las palabras ya sean en caracteres, palabras o frases.

5.3.1.1.3. Pickle

La biblioteca Pickle en Python es un módulo que se utiliza para la serialización y deserialización de objetos Python. La serialización es el proceso de convertir un objeto en memoria en una secuencia de bytes que puede ser almacenada en un archivo o transmitida a través de una red.

5.3.1.2. ChatGPT

ChatGPT es una gran herramienta de inteligencia artificial generativa creada por la empresa OpenAI. Esta gran herramienta está basada en un modelo de arquitectura GPT (Generative Pre-trained Transformer), la cual lleva los modelos Transformers hacia una tarea de comprensión y generación de texto.

Al trabajar con esta gran herramienta, debemos tener en claro que no solo vamos a enviar a través de su API preguntas de los usuarios y esperar una respuesta. Vamos a detallar cuales han sido las herramientas y técnicas usadas para trabajar con ChatGPT, además exploraremos una alternativa de crear nuestro propio modelo.

5.3.1.2.1. OpenAI API

Primero empezaremos explicando la herramienta con la que hacemos las llamadas a la API de ChatGPT, OpenAIApi. Esta biblioteca creada por OpenAI está disponible para hacer llamadas en muchos tipos de Frameworks y lenguajes de programación. En el caso de la web, esta herramienta se usa dentro de la librería de ReactJS junto a NodeJS.

Para poder hacer una llamada, debemos primero tener una cuenta registrada dentro de la web de OpenAI, ahí podremos crear una clave única para hacer las llamadas a la API. Para hacer la llamada debemos poner en la configuración del código el nombre de la clave, el modelo de ChatGPT que queremos utilizar, el prompt (solicitud) que queremos resolver, el máximo número de Tokens que queremos en nuestra llamada y la temperatura con la que queremos que se nos responda.

ChatGPT tiene diferentes modelos disponibles, siendo el mejor Davincii-003 a la hora de comprender y completar tareas. Dependiendo del modelo, tendrá un coste u otro. El coste viene determinado por el tamaño de tokens de la llamada (la suma de la pregunta que mandamos y la solución de ChatGPT). Es por esto que debemos especificar el tamaño máximo de tokens que queremos como respuesta.

La temperatura de respuesta hace referencia a como de creativa queremos que sea nuestra respuesta.

5.3.1.2.2. TikToken

TikToken es una biblioteca de Python que usaremos para poder medir la longitud de un mensaje medido en Tokens. Esta función es muy útil, ya que dependiendo del modelo de ChatGPT que usemos, se realizará una tokenización distinta de las palabras.

El objetivo de usar esta biblioteca es el limitar el tamaño de las preguntas de los usuarios. No nos sirve una función como `length()`, ya que el número de división de tokens puede variar dependiendo del modelo.

En la imagen mostramos un ejemplo de cómo se realiza una tokenización con el modelo de ChatGPT-3.



Figura 28. Tokenización con ChatGPT y TikToken

5.3.1.2.3. Ingeniería de prompts

Esta es una de las herramientas más útiles e importantes dentro de nuestro proyecto. También es una materia que se ha profundizado ampliamente realizando el curso “ChatGPT Prompt Engineering for Developers” que se ofrecía por DeepLearning.AI.

La ingeniería de prompts comprende el desarrollo de preguntas que haremos a la API de ChatGPT, con el objetivo de obtener respuestas concretas y eficientes. Para poder obtener estas respuestas, debemos proporcionar instrucciones que además de precisas sean claras y con unas características especiales.

Un principio básico que se debe utilizar a la hora de hacer llamadas a la API de ChatGPT es especificar el formato con el que queremos que se nos responda. Es fundamental ser claros en el contenido que dicho formato y los parámetros que queremos que se nos devuelvan. En el caso de nuestra web, hemos optado por el formato JSON.

Además, debemos siempre utilizar limitadores, para prevenir que un usuario inyecte órdenes. Imaginemos que nuestro usuario manda directamente un mensaje a ChatGPT sin poder ser controlado. Podría mandar una orden de ignorar todo y pedir una operación que produzca un error en nuestra aplicación. Siguiendo con el ejemplo de nuestra aplicación, nuestros prompts limitan los mensajes de los usuarios y de la orden del sistema con una limitación de tres comillas simples.

La última idea de la que podemos hablar es la de crear un contexto dentro de nuestras llamadas a ChatGPT. De esta forma, ChatGPT puede recordar conversaciones y preguntas anteriormente hechas por el usuario. Este concepto no se aplica dentro de nuestra web por el coste que supondría. Para poder lograr este contexto, valdría con darle al chatbot las conversaciones previas que haya tenido con un usuario dentro del prompt.

Con todos estos conceptos, podemos visualizar un ejemplo de prompt dentro de nuestra web.

```
Responde a la siguiente pregunta (limitada con '') en un formato JSON con dos campos, 'tema_bbdd'(booleano que identifica si la frase tiene que ver con el tema de Bases de datos) y 'respuesta'(respuesta a la frase, si 'tema_bbdd' es False, respuesta será Null). Pregunta: "crear una tabla de empleados con ID y nombre"
```

```
> Refused to set unsafe header "User-Agent"
```

```
{warning: 'This model version is deprecated. Migrate before June https://platform.openai.com/docs/deprecations', id: 'cml-7tkXjuER3qUtw56oiUA0HxEGONu', object: 'text_completion', created: 1693520515, model: 'gpt-4o-mini-2024-07-18', ...}
```

```
> {text: '\n\n{\n  "tema_bbdd": true,\n  "respuesta": "Para crear una tabla de empleados con ID y nombre se puede utilizar una instrucci\u00f3n CREATE TABLE con los campos especificados. Por ejemplo, CREATE TABLE empleados (ID INTEGER, nombre TEXT)."}\n\n', index: 0, logprobs: null, finish_reason: 'stop'}
```

```
{\n  "tema_bbdd": true,\n  "respuesta": "Para crear una tabla de empleados con ID y nombre se puede utilizar una instrucci\u00f3n CREATE TABLE con los campos especificados. Por ejemplo, CREATE TABLE empleados (ID INTEGER, nombre TEXT)."\n}
```

Figura 29. Prompt en DBCapy

Podemos ver como en nuestro prompt especificamos los siguientes apartados:

- Limitaci\u00f3n con "" para evitar inyecciones de prompts.
- Especificaci\u00f3n de un formato JSON (formado por tema_bbdd y respuesta).
- Especificaci\u00f3n de cada tarea a resolver de forma clara y concisa.

Ya con estas tres herramientas, podemos hacer un uso correcto de llamadas a ChatGPT.

5.3.1.2.4. Fine Tuning

El \u00faltimo punto que veremos dentro de ChatGPT ser\u00e1 el de entrenar el modelo pre-entrenado de ChatGPT para que se adapte a nuestras necesidades de una forma m\u00e1s espec\u00edfica. En otras palabras, vamos a crear nuestro propio modelo con un set de datos espec\u00edfico a partir del modelo de ChatGPT.

Se explica este concepto, ya que se dise\u00f1\u00f3 un modelo Fine Tuning para poder utilizarlo dentro de nuestra web, pero al tratarse de una herramienta tan novedosa y que exige una gran cantidad de datos para poder ser entrenada, se dej\u00f3 como un producto en formato "Beta" para poder ser mejorada en un futuro. Aun as\u00ed, se cree que es una herramienta muy potente igual que la Ingenier\u00eda de Prompts y se debe destacar dentro de este punto.

Se trata de una utilidad mucho m\u00e1s avanzada dentro de las que ofrece OpenAI, ya que nos permite poder crear funciones muy espec\u00edficas para problemas concretos que tengamos.

Para poder crear este modelo fino de ChatGPT primero debemos crear nuestro set de datos. Este set de datos debe estar formado por diferentes situaciones que pueden darse entre conversaciones de usuarios y el bot. Por ejemplo, preguntas sobre el temario de bases de datos o pedir resolver ejercicios. Las respuestas del bot siempre deben de ser en un formato JSON (como hab\u00edamos especificado anteriormente dentro los prompts). Tambi\u00e9n incluiremos diferentes preguntas que no tengan que ver con la materia para que nuestro modelo Fine Tuning entienda que debe solo responder preguntas que tengan que ver con la materia de bases de datos.

	A	B
B3		{ "tema_bbdd": "True", "respuesta": "La diferencia entre una base de datos relacional y una base de datos no relacional radica en su estructura y forma de almacenar
7	Cual es la diferencia entre SQL y MongoDB?	{ "tema_bbdd": "True", "respuesta": "SQL es un lenguaje de consulta estructurado utilizado para gestionar bases de datos relacionales, donde los datos se organizan en ta
8	Quien es el presidente de España?	{ "tema_bbdd": "False", "respuesta": "Null" }###
9	¿Qué es el modelo entidad-relación y cómo se u	{ "tema_bbdd": "True", "respuesta": "El modelo entidad-relación es un enfoque para diseñar bases de datos que representa visualmente las entidades, relaciones y atribur
10	¿Cuáles son las ventajas de utilizar un sistema di	{ "tema_bbdd": "True", "respuesta": "El uso de un SGBD en lugar de archivos planos brinda ventajas como una organización estructurada de los datos, garantía de integrid
11	¿Qué es la normalización de una base de datos?	{ "tema_bbdd": "True", "respuesta": "La normalización de una base de datos es el proceso de diseñar su estructura eficientemente. Es importante por las siguientes razos
12	Explica las diferencias entre las tres primeras fo	{ "tema_bbdd": "True", "respuesta": "1NF: Cada atributo de una tabla debe contener un solo valor y no puede ser multivaluado. No debe haber duplicación de filas.
13	quien es el creador de esta aplicación?	{ "tema_bbdd": "False", "respuesta": "Null" }###
14	¿Qué es un capibara?	{ "tema_bbdd": "False", "respuesta": "Null" }###
15	Cual es la capital de Perú?	{ "tema_bbdd": "False", "respuesta": "Null" }###
16	¿Qué son los índices en una base de datos y cuántos	{ "tema_bbdd": "True", "respuesta": "Los índices en una base de datos son estructuras que mejoran la velocidad y eficiencia de las operaciones de búsqueda y recuperaci
17	Puedes poner un ejemplo de una operación CO	{ "tema_bbdd": "True", "respuesta": "Supongamos que tenemos una tabla llamada 'usuarios' con los siguientes campos: 'id' (clave primaria) y 'nombre'. Queremos contar
18	¿Qué es una CPU?	{ "tema_bbdd": "False", "respuesta": "Null" }###
19	Cual es la diferencia entre el lenguaje de progr	{ "tema_bbdd": "False", "respuesta": "Null" }###
20	¿Cómo puedo utilizar Python para gestionar bas	{ "tema_bbdd": "True", "respuesta": "Aquí tienes una explicación de cómo utilizar Python para gestionar bases de datos:
21	Explica el concepto de transacción en el contex	{ "tema_bbdd": "True", "respuesta": "Una transacción en una base de datos es una secuencia de operaciones que se ejecutan como una unidad indivisible. Se basa en las
22	Puedo utilizar funciones dentro de SQL?	{ "tema_bbdd": "True", "respuesta": "Sí, se pueden utilizar funciones en SQL para realizar operaciones y manipulaciones de datos. Las funciones integradas en SQL permite
23	¿Quién es Linus Torvalds?	{ "tema_bbdd": "False", "respuesta": "Null" }###
24	Puedes darme una receta de un pastel?	{ "tema_bbdd": "False", "respuesta": "Null" }###
25	Puedes poner un ejemplo de MongoDB? Puedes	{ "tema_bbdd": "True", "respuesta": "Aquí tienes una explicación de cómo utilizar MongoDB:
26	Como creo una nueva tabla de bases de datos c	{ "tema_bbdd": "True", "respuesta": "No puedo darte la solución con código directamente, pero si puedo corregir códigos y darte otro tipo de ayudas." }###
27	Cual es el mejor Sistema Operativo que existe?	{ "tema_bbdd": "False", "respuesta": "Null" }###

Figura 30. Set de datos para Fine Tuning

El problema con nuestro Fine Tuning fue que nuestro set de datos solo tenía 60 ejemplos, y trataba de cubrir una funcionalidad muy compleja. Para que este modelo fuese funcional, deberíamos crear un set de datos con cientos de ejemplos distintos.

Teniendo ya el set de datos, solo nos hace falta entrenar un modelo ya pre-entrenado, como por ejemplo el anteriormente mencionado Davincii-003 para crear el modelo Fine Tuning.

```
(openaienv) C:\Users\msant\OneDrive\Escritorio\modelFine-Tuning>openai api fine_tunes.follow -i ft-gx4PGhZZu7jMNz5tT8mnl9iY
[2023-06-25 21:36:14] Created fine-tune: ft-gx4PGhZZu7jMNz5tT8mnl9iY
[2023-06-25 22:21:06] Fine-tune costs $1.30
[2023-06-25 22:21:06] Fine-tune enqueued. Queue number: 0
[2023-06-25 22:21:07] Fine-tune started
[2023-06-25 22:23:55] Completed epoch 1/4
[2023-06-25 22:24:17] Completed epoch 2/4
[2023-06-25 22:24:40] Completed epoch 3/4
[2023-06-25 22:25:02] Completed epoch 4/4
[2023-06-25 22:25:44] Uploaded model: davinci:ft-usal-2023-06-25-20-25-43
[2023-06-25 22:25:45] Uploaded result file: file-DPzr5VwKPzPeMSaanyRJ0xLi
[2023-06-25 22:25:45] Fine-tune succeeded

Job complete! Status: succeeded 0
Try out your fine-tuned model:

openai api completions.create -m davinci:ft-usal-2023-06-25-20-25-43 -p <YOUR_PROMPT>
```

Figura 31. Entrenamiento de Davincii-003

5.3.2. Herramientas usadas para crear el entorno de la web

En este apartado vamos a centrarnos en todas las herramientas que se usaron para poder crear nuestra web. Se dará una breve explicación de cada una de ellas y un pequeño contexto de cómo se usaron.

5.3.2.1. ReactJS

ReactJS es una biblioteca de JavaScript que nos servirá como interfaz interactiva para los usuarios. Cada parte de la interfaz estará comprendida dentro de un Container el cual puede estar compuesto por diferentes componentes.

React utiliza un enfoque declarativo, que significa que la interfaz puede cambiar en función del estado actual y de los datos. React se encargará de actualizar de forma automática la interfaz cada vez que se cambien los datos.

Hemos optado por utilizar ReactJS dentro de nuestro proyecto ya que es uno de los entornos más conocidos y utilizados para la programación web.

5.3.2.2. Node.JS

Node.JS es un entorno de ejecución de JavaScript. Permite ejecutar códigos JavaScript fuera de un navegador. Para poder crear nuestra aplicación de React, debe correr siempre dentro de Node.JS.

También Node será imprescindible dentro de nuestro proyecto ya que nos facilita muchas bibliotecas para poder desarrollar nuestro servidor.

5.3.2.3 Django

Django es un framework de desarrollo de web de código abierto basado en Python. Usaremos Django para crear un Backend para nuestra web. Dicho Backend lo utilizaremos sobre todo para responder las distintas llamadas de nuestro Frontend. Además, dentro de Django almacenaremos los modelos que se guardarán dentro de nuestra base de datos.

5.3.2.3.1. Django Rest Framework

Django Rest Framework es una potente biblioteca de Django que se utiliza para la creación de API's RESTFull dentro de nuestro entorno de Django.

En nuestro caso, usaremos esta biblioteca para poder crear una API capaz de procesar todos los mensajes que nos lleguen desde el Frontend de React y vayan dirigidos a Django.

5.3.2.4 PostgreSQL

PostgreSQL será nuestro sistema gestor de bases de datos relacional. Se caracteriza por ser eficiente, escalable y capaz de realizar muchas consultas.

Usaremos PostgreSQL para poder guardar tanto los datos personales de nuestros usuarios, como los mensajes y las peticiones que se van a ir realizando a ChatGPT.

5.3.2.5 Djoser

Djoser es una biblioteca de Python añadida dentro de Django para poder simplificar la creación de los usuarios y su validación con un sistema de Tokens. Usaremos además en Djoser para especificar un usuario customizado con los estatus y las características que necesitamos.

5.3.2.6 Axios

Axios es una biblioteca de JavaScript para realizar solicitudes de HTTP desde nuestro entorno Node.JS. Usaremos Axios para comunicar nuestro Frontend de ReactJS con el Backend de Django.

5.3.3. Herramientas complementarias

Por último, haremos las últimas explicaciones de las herramientas complementarias que hemos usado dentro de nuestro proyecto

5.3.3.1 DBDiagram.io

DBDiagram.io es una herramienta gráfica para poder visualizar las bases de datos relacionales. Estos diagramas se usarán para visualizar los ejercicios prácticos dentro de nuestra web.

5.3.3.2 Draw.io

Draw.io es otra aplicación web usada para poder visualizar los diferentes esquemas gráficos usados para explicar el proceso unificado.

5.3.3.3 Microsoft Project

Microsoft Project es una aplicación de administración de proyectos creado por Microsoft. Se ha usado Project para poder crear nuestro diagrama de Gantt.

5.3.3.4 EZEstimate

Se trata de una herramienta CASE en forma de aplicación de escritorio de Windows que permite calcular la estimación temporal y de costes de un sistema a partir de los requisitos del sistema.

5.3.3.5 Adobe Color

Se trata de una herramienta en línea de Adobe que se utiliza para poder crear paletas de colores. Se usó Adobe Color para poder crear un patrón de colores que sea complementario y coherente dentro de nuestra web.

5.3.3.6 Google Forms

Google Forms es una aplicación en línea creada por Google que permite a los usuarios hacer formularios. En el caso de nuestro proyecto, vamos a usar Forms para crear un formulario en el que los usuarios de nuestra aplicación puedan dar sus opiniones de los diferentes recursos que la web ofrece. Sobre todo, nos vamos a centrar en hacer preguntas a los usuarios relacionadas con el Chatbot, ya que es el principal aporte que tiene nuestra web.

6. ASPECTOS RELEVANTES AL DESARROLLO

En este apartado vamos a ver los diferentes puntos relevantes en el desarrollo de nuestro proyecto. Nos centraremos por completo en el ciclo de vida del desarrollo del sistema dentro de la vista del proceso unificado.

Como se vió anteriormente, dividiremos el desarrollo en cinco iteraciones las cuales marcarán las distintas etapas de creación de la web. Se explicarán en mayor detalle cada una de las etapas junto a las tareas de dichas etapas.

Por último, se mostrarán a modo de resumen global, la vista de la web mostrando las distintas funcionalidades que se ofrecen.

6.1. Marco de trabajo

Anteriormente se explicó que el modelo para desarrollar nuestro proyecto se basa en el Proceso Unificado. También se destacaron cada una de las etapas y sus características.

En este punto vamos a ir explicando cada una de las iteraciones, pero dividido en las diferentes clases de trabajo dentro del Proceso Unificado (Modelo de Negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas).

6.1.1. Modelo de negocio

El modelo de negocio junto todas las tareas que tengan una relación referente con el estudio y con la recolección de datos para luego ser aplicados al resto de las fases.

Las características de nuestro modelo de Negocio en cada etapa fueron las siguientes:

- **Iteración 1:** En esta primera etapa del proyecto se investigaron sobre todo lo referido a los antecedentes del proyecto. Véase por ejemplo las diferentes webs que implementaban una enseñanza de bases de datos, diferentes proyectos que usaban Chatbot para la enseñanza. También se empezó a investigar sobre que plataformas podríamos crear la web.
- **Iteración 2:** En esta segunda iteración se hace una gran inversión investigar sobre como implementar una red neuronal y ChatGPT dentro de nuestro proyecto. Ya hemos decidido que estas serán las herramientas para implementar el bot dentro de nuestra web, por tanto, vamos a valorar diferentes formas en las cuales puedan ser incorporadas.
- **Iteración 3:** En esta etapa se centra en refinar el ChatBot. Aquí haremos una investigación en técnicas y herramientas más avanzadas para poder mejorar nuestro Chatbot. Dentro de esta iteración se hará la investigación sobre la ingeniería de Prompts y un profundiza miento en cómo se procesa el lenguaje natural. Ya en esta etapa el modelo de negocio pierde algo de importancia.
- **Iteración 4:** En esta etapa el modelo de negocio pierde gran importancia. Solo nos centramos en la investigación del contenido que queremos que contenga la web y la estética que queremos conseguir dentro de la web.
- **Iteración 5:** En esta iteración nos centramos en una sola tarea, revisión de los posibles riesgos existentes en el sistema. La descripción de esta tarea es la siguiente:

“Visionado final de posibles riesgos que pueda sufrir nuestro proyecto ya terminado. Por ejemplo, contemplar posibles limites en el uso de ChatGPT que debemos tener en cuenta si decidimos escalar más el sistema.”.

6.1.2. Requisitos

En la primera etapa de nuestro proyecto, además de hacer una reunión con los clientes, que este caso serán las tutoras. Creando así la primera propuesta de objetivos del sistema. Estos objetivos, aunque son iniciales, vamos a intentar a que sean lo más estables a lo largo de nuestro proyecto.

Aunque a lo largo del proyecto vamos a ir refinando los requisitos en subsistemas diferentes, es importante mencionar como en la segunda iteración se crea el primer modelo inicial de requisitos, este viene dado por los Casos de Uso, ya teniendo en cuenta todos los objetivos y con la idea de cómo se forma el chatbot. A pesar de que poseen su propio apartado en su respectivo Anexo, se incluyen a continuación algunos de los UC más relevantes del sistema.

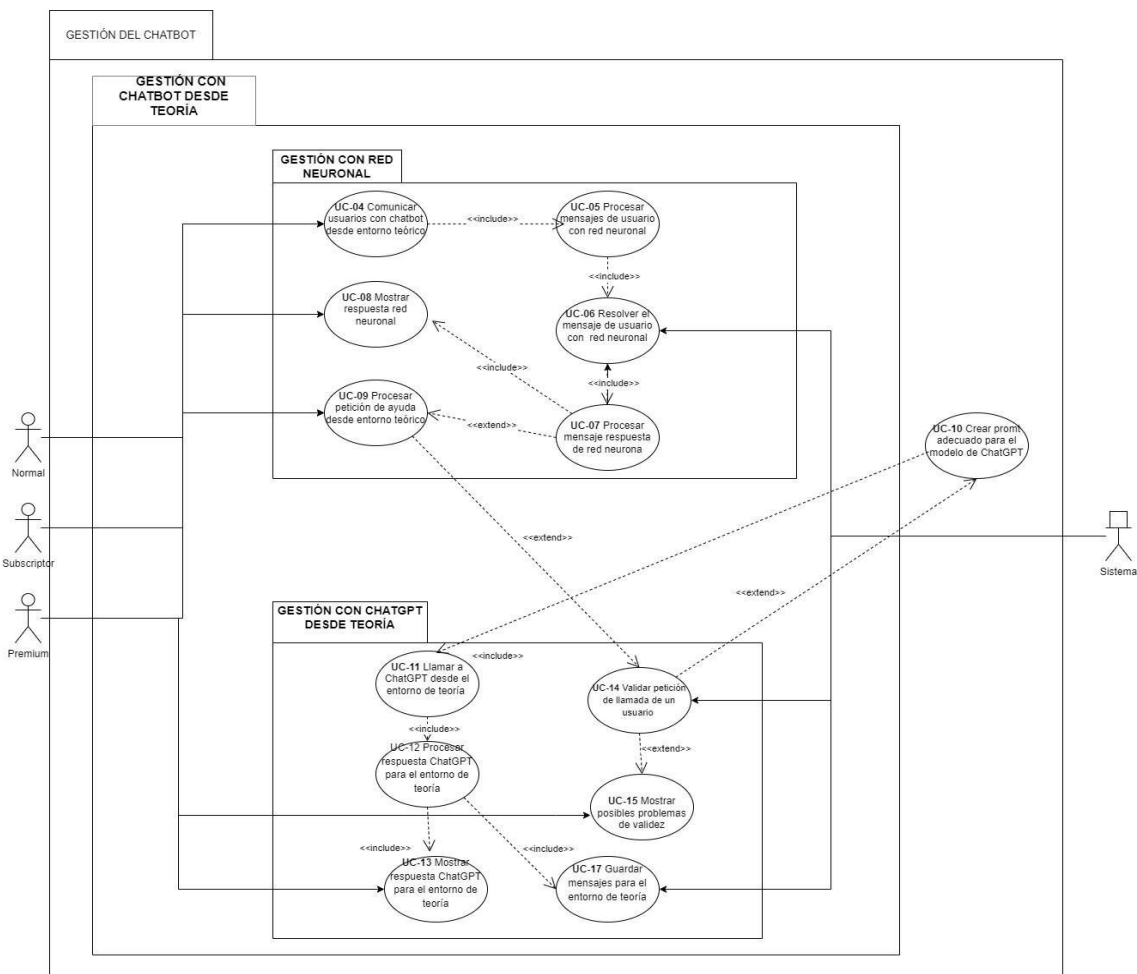


Figura 32. Extracto del diagrama de Casos de Uso (Gestión chatbot desde teoría)

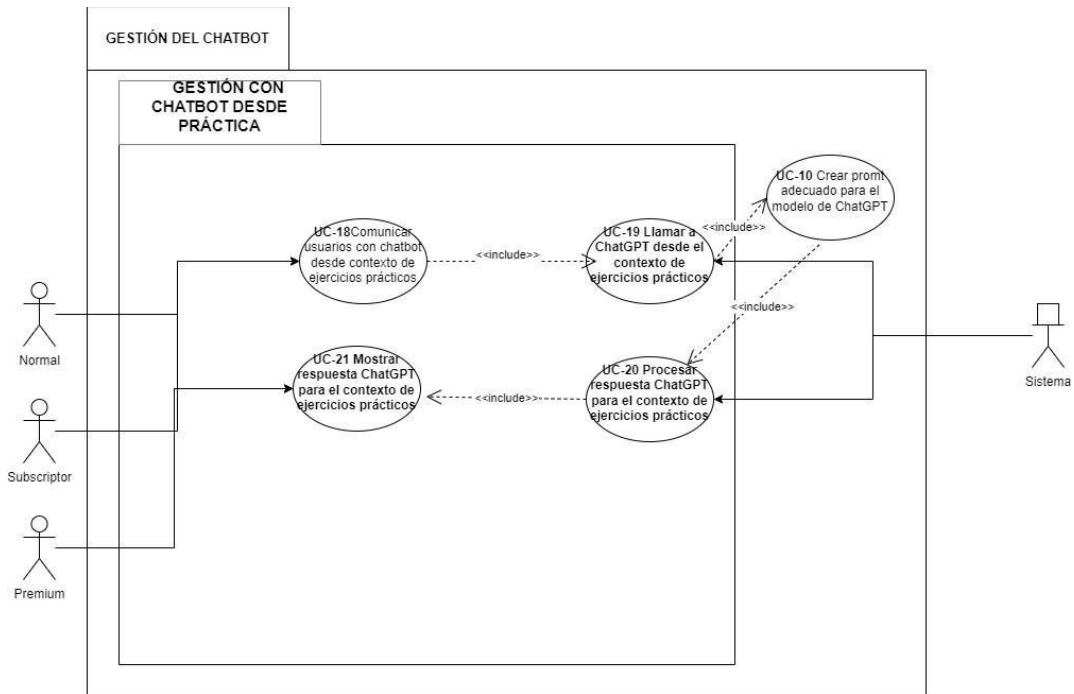


Figura 33. Extracto del diagrama de Casos de Uso (Gestión del chatbot desde prácticxa)

Ya formados los requisitos funcionales gracias a los UC, estos se irán refinando a lo largo de nuestro proyecto, pero intentando siempre mantener una estructura parecida. Al final lo que estamos intentando con nuestros requisitos es cumplir todos los objetivos establecidos.

Lo siguiente que mostraremos son las distintas tablas reflejan el resto de los requisitos de nuestro sistema. Solo se mostrarán sus identificaciones y su descripción, para poder ver más detalles de cada uno de los requisitos se recomienda ir al “Anexo II: Especificación de requisitos software”.

Tabla 1. Catálogo de requisitos de información

ID	Nombre	Descripción
IRQ-01	Información sobre los usuarios	El sistema deberá almacenar la información correspondiente a los usuarios para el registro y demás funcionalidades.
IRQ-02	Información sobre peticiones a ChatGPT	El sistema deberá almacenar la información correspondiente al registro de mensajes que se hacen a ChatGPT.
IRQ-03	Información sobre mensajes de usuarios	El sistema deberá almacenar la información correspondiente a los distintos mensajes e interacciones entre el usuario y el chatbot.
IRQ-04	Información sobre modelo de red neuronal	El sistema deberá almacenar la información correspondiente a los distintos datos para entrenar a la red

		neuronal.
CRQ-01	Limitación en las llamadas a la API	La información almacenada por el sistema deberá satisfacer la siguiente restricción: Dependiendo del tipo de usuario, se limitará el número y la longitud de preguntas a la API que un usuario puede hacer.

Tabla 2. Catálogo de requisitos funcionales

ID	Nombre	Descripción
UC-01	Mostrar material teórico de la asignatura bases de datos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario entre dentro del servicio para encontrar material teórico para la formación en la asignatura de Bases de Datos.
UC-02	Relacionar información teórica con ejercicios prácticos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario esté estudiando la materia teórica de la asignatura y quiera realizar ejercicios prácticos.
UC-03	Mostrar ejercicios prácticos de la asignatura de bases de datos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera realizar enunciados prácticos a resolver.
UC-04	Comunicar usuarios con chatbot desde entorno teórico	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee comunicarse con el agente virtual desde el entorno de teoría.
UC-05	Procesar mensajes de usuario con red neuronal	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema reciba un mensaje del usuario dirigido hacia la red neuronal.
UC-06	Resolver el mensaje de usuario con red neuronal	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema deba resolver el mensaje del usuario.

UC-07	Procesar mensaje respuesta de red neuronal	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema resuelva el mensaje y deba procesar la respuesta.
UC-08	Mostrar respuesta red neuronal	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema vaya a mostrar la respuesta.
UC-09	Procesar petición de ayuda desde entorno teórico	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicite ayuda a través del agente en el contexto teórico.
UC-10	Crear prompt adecuado para el modelo de ChatGPT	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario realice una llamada a ChatGPT y se deba crear un prompt para dicha llamada.
UC-11	Llamar a ChatGPT desde el entorno de teoría	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema realice una llamada a ChatGPT desde el entorno de teoría.
UC-12	Procesar respuesta ChatGPT para el entorno de teoría	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema reciba la respuesta de ChatGPT desde el entorno de teoría.
UC-13	Mostrar respuesta ChatGPT para el entorno de teoría	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema debe mostrar la respuesta al usuario desde el entorno de teoría.
UC-14	Validar petición de llamada de un usuario	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario vaya a realizar una llamada a ChatGPT desde el entorno de teoría.
UC-15	Mostrar posibles problemas de validez	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema encuentre un error en la validación de llamada de un usuario.
UC-16	Registrar petición	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema valide de forma positiva una petición.

UC-17	Guardar mensajes para el entorno de teoría	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema detecte un nuevo mensaje entre el usuario y el agente inteligente desde el entorno de teoría.
UC-18	Comunicar usuarios con chatbot desde contexto de ejercicios prácticos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee comunicarse con el agente virtual desde el contexto de los ejercicios prácticos.
UC-19	Llamar a ChatGPT desde el contexto de ejercicios prácticos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema realice una llamada a ChatGPT desde el contexto de los ejercicios prácticos.
UC-20	Procesar respuesta ChatGPT para el contexto de ejercicios prácticos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema reciba la respuesta de ChatGPT desde el contexto de los ejercicios prácticos.
UC-21	Mostrar respuesta ChatGPT para el contexto de ejercicios prácticos	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el sistema debe mostrar la respuesta al usuario desde el contexto de los ejercicios prácticos.
UC-22	Crear estadísticas para los usuarios	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera dar su opinión sobre el funcionamiento del servicio.
UC-23	Iniciar sesión	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera entrar en el servicio web.
UC-24	Registro	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera registrarse dentro del servicio web.
UC-25	Recuperar contraseña	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera recuperar la contraseña.
UC-26	Validar usuario	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera terminar de registrarse.
UC-27	Salir sesión	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando

		el usuario quiera salir del servicio web.
--	--	---

Tabla 3. Catálogo de requisitos no funcionales

ID	Nombre	Descripción
RNF-01	Escalabilidad	El sistema deberá afrontar un aumento de escala de usuarios que quieran usar el servicio.
rdRNF-02	Interfaz	El sistema deberá tener una interfaz que guíe a los usuarios para el aprendizaje de la materia de bases de datos.
RNF-03	Usabilidad	El sistema deberá ser fácil de entender para los usuarios.
RNF-04	Mantenibilidad	El sistema deberá ser fácil de mantener y de cambiar si se desea cambiar alguna de las dos herramientas del chatBot.
RNF-05	Seguridad	El sistema deberá garantizar una seguridad para todos los usuarios al depositar información sensible.
RNF-06	Eficiencia	El sistema deberá ser capaz de proporcionar un rendimiento bueno con los recursos disponibles.
RNF-07	Manual de usuario	El sistema deberá tener un manual de usuario para que cualquier estudiante pueda entender todas las funciones que se pueden realizar dentro de la web.
RNF-08	Manual de programador	El sistema deberá tener un manual de programador donde se detalle las tecnologías usadas. En especial el uso de la nueva herramienta de ChatGPT.

6.1.3. Análisis

Anteriormente hemos definido todos los requisitos de nuestra aplicación. No obstante, todas las especificaciones tienen un nivel de abstracción alto.

Buscamos en este apartado poder bajar el nivel de abstracción y crear una visión mucho más realista de cómo será nuestro proyecto. El análisis de requisitos es una etapa de vital importancia ya que ayuda en las siguientes etapas de diseño y de desarrollo del sistema. Queremos aproximarnos a este concepto partiendo de todos los requisitos anteriormente redactados.

En la primera iteración se realizará el primer boceto de la arquitectura (modelo de dominio), definiendo los elementos que se almacenan dentro de nuestro sistema y ver cómo se relacionan. Debido a que nos encontramos en la primera iteración, debemos considerar el nivel de abstracción de ésta y por tanto se ha realizado este modelado mediante un diagrama de clases.

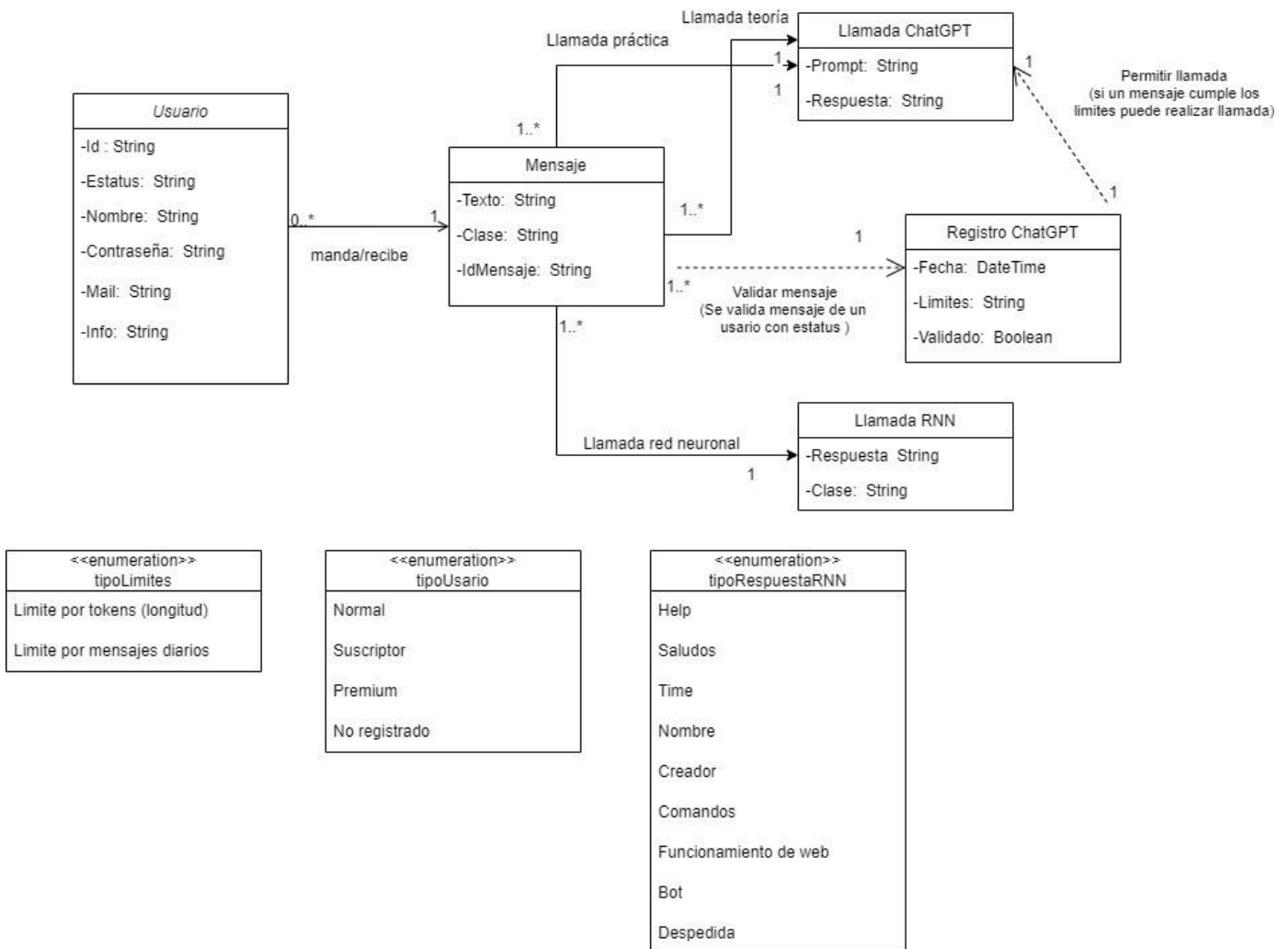


Figura 34. Diagrama de clases del sistema

Usuario: El usuario es la clase principal de nuestro sistema, ya que cada usuario es el que genera los distintos mensajes y llamadas al chatbot. Cada usuario tiene su identificador, sus valores (nombre, contraseña, etc.) y un estatus. Se listan los distintos estatus en la enumeration tipoUsuario. Dependiendo del estatus, un usuario puede tener unos límites distintos a la hora de hacer llamadas a la API de ChatGPT.

- **Mensaje:** Los mensajes es una clase que engloba tanto mensajes que un usuario pueda hacer como las respuestas que ChatGPT o la Red Neuronal puede tener.

- **Llamada ChatGPT:** Las llamadas de ChatGPT se compondrán de un Prompt (la petición que se manda a la API de ChatGPT) y una respuesta que obtendremos.

- **Llamada RNN:** Una llamada a la Red Neuronal también tiene una estructura parecida a la Llamada ChatGPT, ya que obtenemos una respuesta además de una clase. La Clase hace referencia a los distintos tipos de respuesta que se le puede dar a un usuario. Todas estas clases están enumeradas dentro del enumeration tipoRespuestaRNN.

- **Registro ChatGPT:** el último dominio que vamos a analizar es el de Registro ChatGPT. Este será el dominio encargado de guardar las peticiones que realizan los usuarios para poder llamar a ChatGPT. Se usa dentro del dominio un límite para validar el mensaje de un usuario. Hay dos tipos de límites que se muestran dentro del enumeration tipoLimites: límite del tamaño de la pregunta (Limite por tokens) y límite de peticiones de un usuario diarias (Limite por mensaje).

En el resto de las iteraciones se siguió refinando la definición del sistema en este dominio valiéndose de la definición de los casos de uso mediante diagramas de secuencia. En otras palabras, vamos a bajar el nivel de abstracción de los casos de uso anteriormente definidos para visualizar su flujo de información

Se destaca el caso de uso “UC-09 Procesar petición de ayuda desde entorno teórico” ya que se trata de uno de los casos de uso representativos de nuestro sistema.

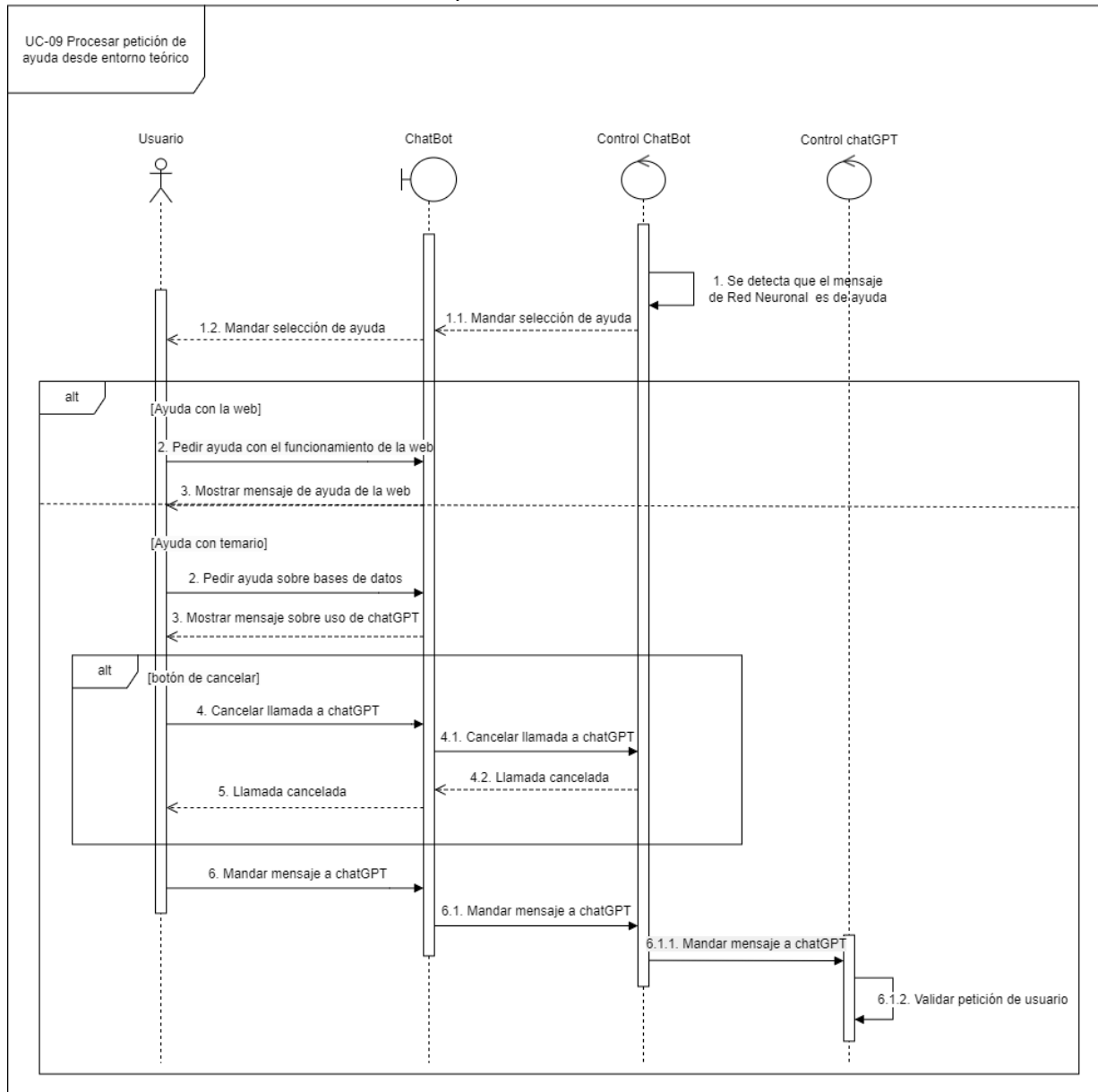


Figura 35. UC-09 Diagrama de secuencia en el análisis

A pesar de que la definición completa de estos diagramas se encuentra en su respectivo Anexo, se destaca el funcionamiento del mismo de nuevo aquí por su importancia.

Nos encontramos con los siguientes elementos: Actor usuario, la interfaz del Chatbot, el control del Chatbot y el control de ChatGPT. Destacamos como por un lado separamos nuestro ChatBot de sus herramientas (ChatGPT y la red neuronal), cada uno tendrá su propio control.

Vemos también como el Chatbot es quien procesa los mensajes que recibe tanto de ChatGPT y de la red Neuronal para poder comunicárselo a la interfaz y posteriormente al usuario.

6.1.4. Diseño

En esta etapa del Proceso Unificado, nos centramos en un punto de vista ya muy refinado y poco abstracto de nuestro sistema final. Tendremos en cuenta todos los diseños y conceptos de las anteriores etapas para tener un nivel de abstracción muy bajo.

Las etapas en las que más destaca el diseño son a partir de la tercera, ya que anteriormente debemos tener un sistema con todas las definiciones de requisitos y análisis.

Teniendo en cuenta la estructura de Frontend y Backend, podremos crear nuestro Subsistema de diseño para poder organizar nuestro proyecto.

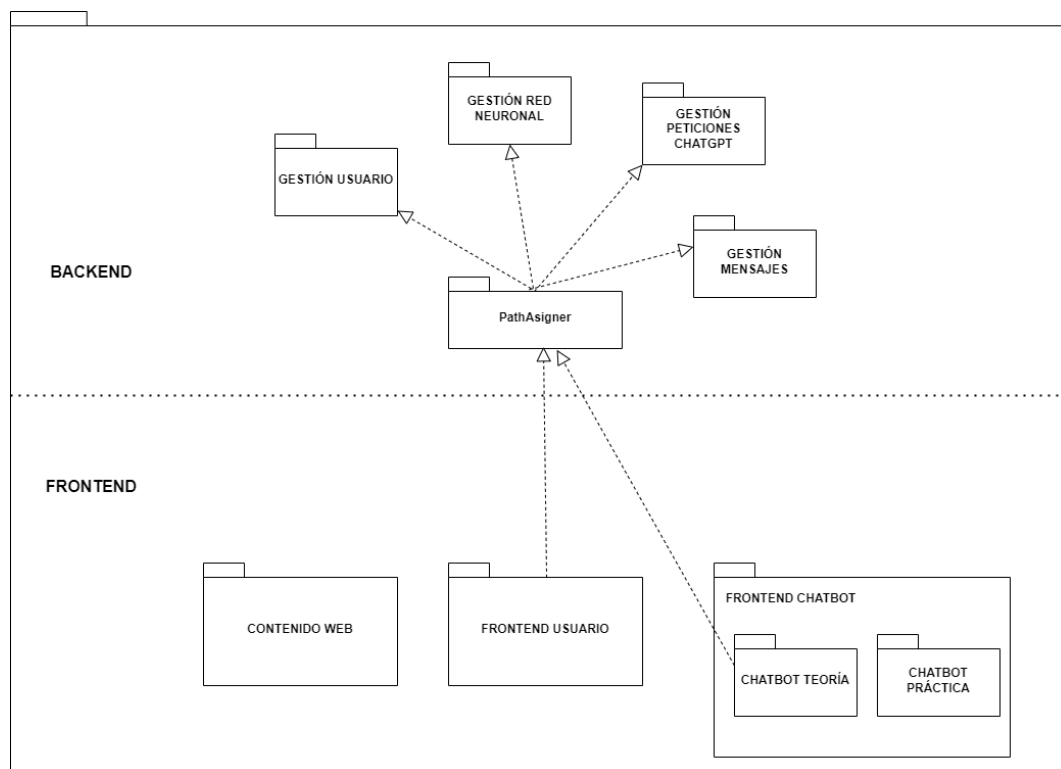


Figura 36. Subsistema del diseño

Este subsistema de diseño se divide en varios paquetes tal y como se puede observar en la figura anterior. Estos paquetes se componen de las Clases de Análisis, que se tratan de la unidad fundamental para la definición del modelo del dominio actual.

Se destaca a continuación la clase de análisis más importante para comprender las pautas seguidas a la hora de realizar este modelado.

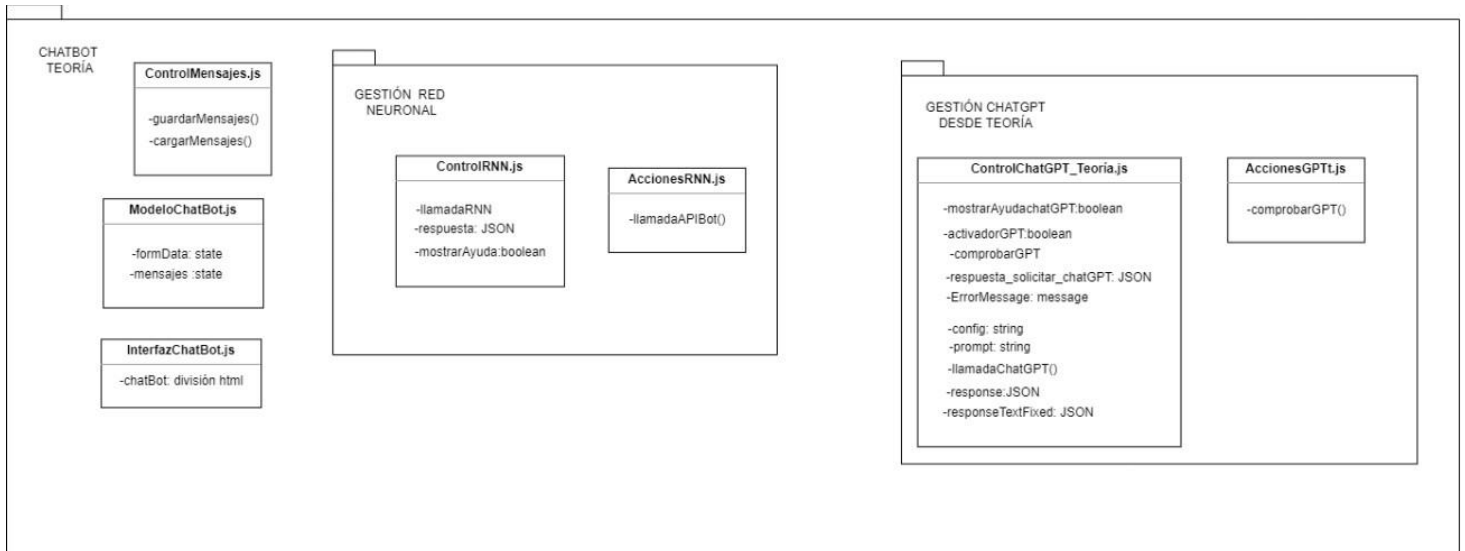


Figura 37. Subpaquete Chatbot Teoría

Este paquete se encuentra dentro del Frontend de nuestro sistema, el cual está compuesto por distintos subpaquetes que engloban tanto funciones como distintos controles y modelos dentro de nuestro sistema gracias a las clases de análisis.

Podemos observar tres tipos de clases de análisis diferentes.

- Interfaz: las cuales contienen la parte visual para un usuario. En este ejemplo encontramos la interfaz de Chatbot, que se encuentra dentro del archivo `InterfazChatBot.js`.
- Control: las cuales contienen las distintas funciones y comprobadores para los datos. Por ejemplo, podemos destacar `ControlRNN.js` la cual está compuesta por las diferentes funciones y datos que se utilizan para poder gestionar los mensajes entre un usuario y la red neuronal.
- Modelo: los encargados de poder guardar y cargar datos. Destacamos el modelo de `InterfazChatbot.js`, el cual es el encargado de guardar los mensajes entre un usuario y el chatbot.

Cabe destacar que en React, muchos de los archivos pueden tener una estructura que mezcle el modelo VMC (Vista-Modelo-Controlador). Para poder comprender mejor estos archivos y el flujo de información que hay en el diseño, se ha decidido renombrar los paquetes para poder entender mejor los próximos diagramas de flujo. En otras palabras, un mismo archivo puede aparecer en múltiples dominios, pero con el nombre cambiado para diferenciarlo en secciones.

6.1.5. Implementación

La implementación de nuestro proyecto toma una gran importancia sobre todo en la tercera y cuarta etapa. Ya que se hará un gran esfuerzo para poder desarrollar primero el componente de Chabot y posteriormente el resto de las funciones de nuestro sistema.

Para poder entender cómo se ha implementado nuestro sistema, se incluye a continuación el modelo de despliegue realizado mediante un diagrama de componentes.

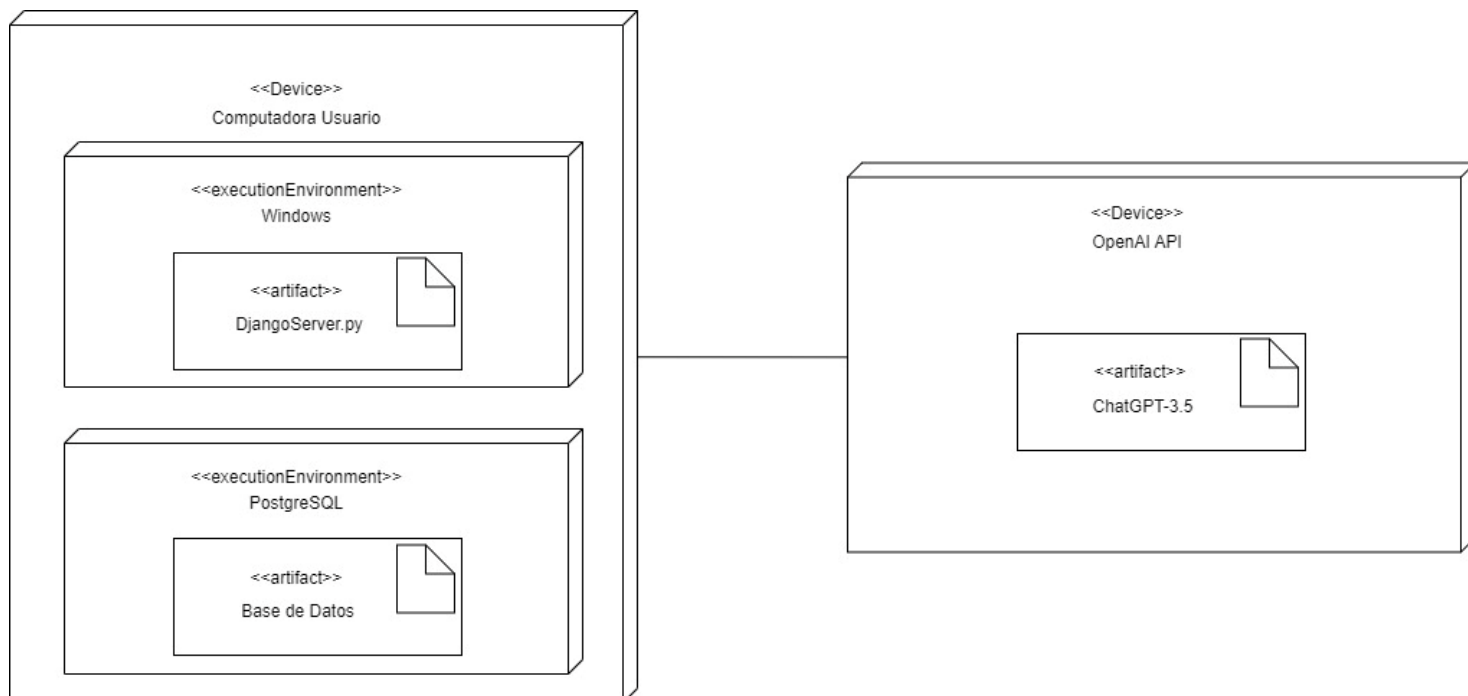


Figura 38. Modelo de despliegue

Nuestro sistema está compuesto solo por un computador de un Usuario, el cual contiene tanto nuestro servidor de Django y nuestra Base de Datos de PostgreSQL. Dicho computador tiene una relación con el servicio que ofrece OpenAI para el uso de la API de ChatGPT.

Este proyecto web solo se ha planteado de forma local, por tanto, solo vamos a preocuparnos de tener un servicio el cual contenga todas nuestras herramientas, sin contar por supuesto con el servidor de OpenAI.

Cabe destacar que solo tenemos un servidor dentro de nuestra computadora de usuario, DjangoServer.py. Esto se debe que, aunque podemos crear un puerto para un servidor de React.js con Node, para que el sistema funcione en conjunto debemos hacer un montaje dentro de Django. En otras palabras, React se comprimirá dentro un fichero que está incluido dentro de nuestro servidor de Django. El código de React se seguirá comunicando con Django, aunque estén en el mismo puerto y servidor con llamadas HTTP. Toda esta comunicación es posible con las bibliotecas de Axios (dentro de React) y la biblioteca Django Rest (que permite crear nuestro servicio Django en una API la cual procesa los mensajes).

Podemos ver las distintas llamadas que se realizan desde el Frontend hacia el Backend con el siguiente diagrama.

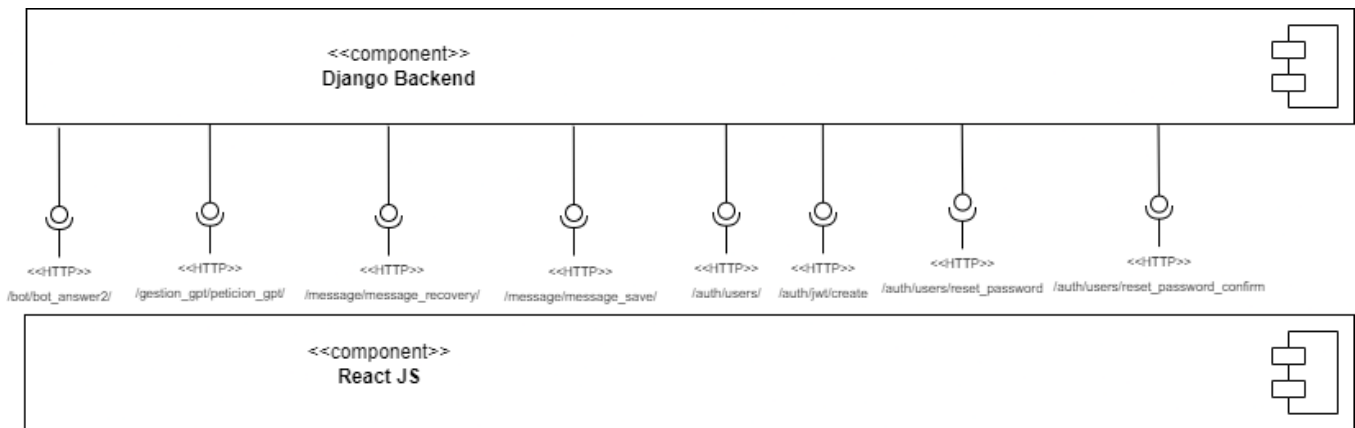


Figura 39. Endpoints del sistema

Otro punto clave que debemos tratar dentro de este apartado es la implementación de nuestro Chatbot. A continuación, explicaremos como se creó y entreno nuestra red neuronal y posteriormente explicaremos en más detalle la integración y el uso de la API que ofrece OpenAI.

6.1.5.1 Implementación de la Red Neuronal

Nuestra red neuronal secuencial está basada en el modelo de *Bag of Word*, modelo previamente explicado dentro de nuestro apartado de conceptos teóricos. El objetivo del modelo es poder clasificar mensajes de los usuarios y relacionarlos con un tema en concreto. Conociendo el tema podremos responder a nuestro usuario con una respuesta ya predefinida. Todo el proceso que se explicará a continuación se ve reflejado dentro del código entrega en el TFG dentro del fichero `back\DJANGO_BACKEND\mysite\bot_architectre\RN_NL.py`.

Para poder comenzar con este modelo, lo primero que debemos hacer es crear nuestra **bolsa de palabras**, es decir, un array que contenga todo el vocabulario que nosotros vamos a permitir que nuestra red neuronal entienda. Para hacer esta función, debemos procesar nuestro set de datos para sacar el vocabulario que manejará la red neuronal. El objetivo de este modelo de Procesado de Lenguaje es determinar el número de repeticiones que una frase tiene respecto a nuestra bolsa de palabras. Conociendo este número de repeticiones, podremos entrenar nuestra red neuronal.

Manuel Santa Isabel Mayo

Hay que señalar que se ha diseñado este modelo en lengua inglesa. El uso de la lengua castellana nos producía continuos errores, que se solucionaban con herramientas que procesaban la lengua inglesa. Además, el inglés es el idioma más extendido en el ámbito de la informática.

Nuestro set de datos se encuentra en un fichero JSON, *Intents_en.json*, dicho JSON contiene una lista de variables *intents*, compuesta por las siguientes variables: *tag* (la clase con la que relacionamos una palabra), *patterns* (conjunto todas las frases que se relacionan con un tema y compondrán nuestro vocabulario) y por último, *responses* (las respuestas predefinidas para cada uno de los temas).

Para poder crear nuestro array de palabras, simplemente debemos recorrer nuestra lista de *intents* y recogiendo cada palabra de la variable *patterns*. Para poder dividir todas las palabras de las frases en *patterns*, haremos uso de la herramienta de lematización *WordNetLemmatizer*. Por último, debemos simplificar nuestra lista de palabras quitando las palabras duplicadas y eliminando también carácter de puntuación, los cuales no nos interesan dentro de nuestro vocabulario.

De esta forma ya tenemos creada nuestra lista con todas las palabras que nuestra red neuronal procesará. A su vez, mientras procesábamos toda la lista de palabras, hemos creado una lista con todos los tags y también una lista llamada *documents*, la cual almacenaremos las frases y el tag de cada una de ellas. Esta última lista es muy importante y la usaremos más adelante.

El siguiente paso es mucho más complejo, ya que ahora lo que debemos realizar es la conversión de nuestro vector de palabras en un conjunto de datos que puedan ser interpretados por la red neuronal. Primero de todo, crearemos dos listas: *training*, una lista que empezará vacía e irá almacenando todos los datos del entrenamiento, y luego *outputEmpty*, otra lista que guardará las posibles respuestas que puede determinar nuestra red neuronal (en este caso, el tamaño de la lista son el número de *Tags*). Siempre debemos recordar que lo que intentamos con nuestra red neuronal es entrenar dicho modelo para que procese una palabra y determine de que tema se trata (a que *Tag* pertenece).

Lo siguiente que realizamos es el procesado de nuestra lista *documents*, dividiremos esta lista para quedarnos solo las frases. Para cada frase crearemos una lista llamada ***bag***, dicha lista será completada por 1 o 0 dependiendo si una palabra dentro de nuestra bolsa de palabras se encuentra o no dentro de dicha frase que estamos analizando.

El resultado serán tantas listas *bags* de tamaño igual al vocabulario como el número de frases de nuestro set de datos.

Para cada lista *bag*, se le asociará una lista *outputRow* (copiada de la anteriormente creada *outputEmpty*) asociando a cada *bag*, el *tag* correspondiente al que pertenece. De esta forma, ya tenemos tanto los datos de entrada y los datos de salida que nuestra red neuronal va a procesar.

Se puede ver de forma gráfica todo este proceso con la siguiente ilustración:

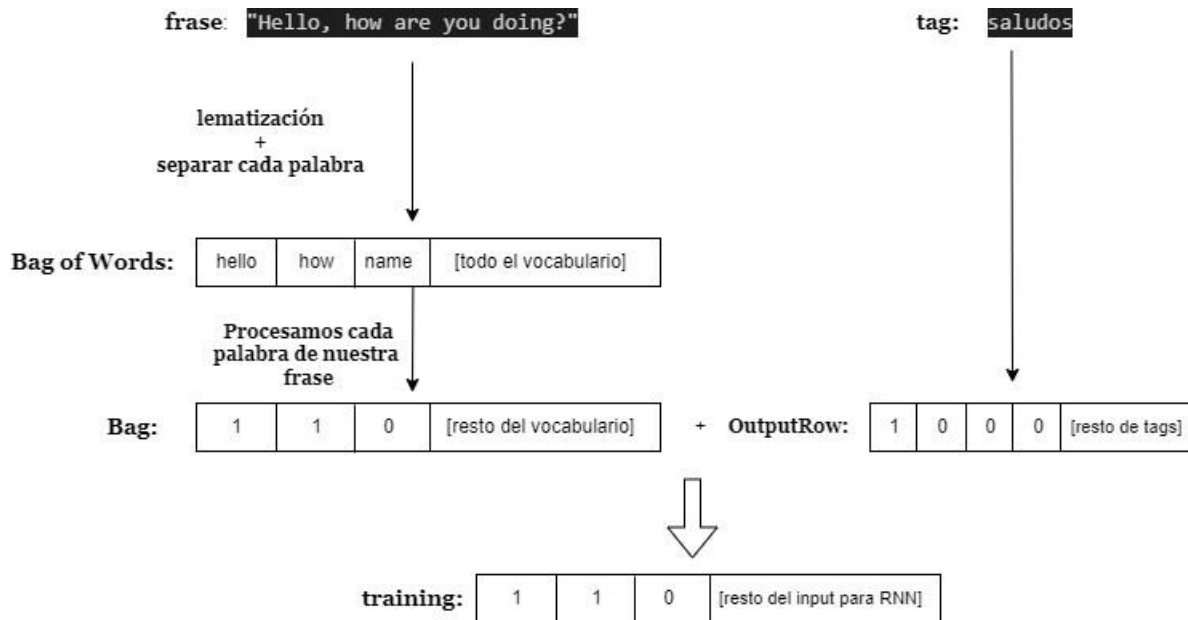


Figura 40. Procesamiento de frases en la RNA

Ya teniendo claro cómo vamos a procesar los datos nuestra red neuronal, es momento de hablar de la arquitectura y finalmente del entrenamiento de nuestra red neuronal.

Nuestra red neuronal secuencial se construye con la biblioteca de Tensorflow y está compuesta por 5 capas (dos de ellas son capas de Dropout, que sirven para no saturar a nuestra red neuronal). Las capas tienen las siguientes características:

- Capa 1: Capa densa de 128 neuronas y función de activación ReLu que usaremos como capa de entrada, por tanto, su tamaño será igual al tamaño de una lista bag.
- Capa 2: Primera capa de Dropout que usaremos para poder bajar la densidad de todas las respuestas generadas. Al estar creando un modelo que procese un vector tan grande, debemos incluir estas capas anti-saturaciones.
- Capa 3: Segunda capa densa de 64 neuronas y con una función de activación ReLu.
- Capa 4: Segunda capa de Dropout.
- Capa 5: Última capa densa con un total de neuronas igual al número de tags. En esta capa se utilizará la función de activación de Softmax, ya que buscamos una predicción para saber la clase a la que pertenece una frase.

La función de pérdida se establece en *'categorical_crossentropy'* ya que se trata de un problema de clasificación de múltiples clases.

Lo último que tendremos que hacer es entrenar a nuestra red neuronal con nuestros datos. Para ello, utilizaremos la función de *fit* de TensorFlow. Dentro de esta función, especificaremos los datos de entrada, los datos de salida, el número de repeticiones (denominadas épocas) que queremos que tenga nuestro aprendizaje el lote de épocas que queremos tener antes de un ajuste de pesos (esto se debe a que si aplicamos un ajuste de pesos a cada una de las épocas tendremos además de un mayor coste computacional, una mayor dificultad para poder tener ajustes más finos).

Ya entrenado nuestro modelo de red neuronal, estamos preparada para resolver inputs que nos manden los usuarios. Solo debemos integrar nuestro modelo creado dentro de nuestro





Manuel Santa Isabel Mayo

servidor de Django. Usaremos TensorFlow para poder cargar y utilizar este modelo entrenado dentro del servidor.

6.1.5.2 Implementación de ChatGPT

La implementación de ChatGPT es mucho más sencilla que la de nuestra red neural. Primero de todo, debemos instalar dentro de React JS la biblioteca de OpenAI, esta biblioteca será llamada y configurada cada vez que hagamos una llamada a ChatGPT.

Es muy importante recalcar que para poder realizar cualquier llamada a ChatGPT debemos primero declarar nuestra llave personal de OpenAI. Esta llave funciona como Token de validación para poder llamar a la API. En nuestro caso, esta será la única configuración que tendremos para las llamadas.

NAME	KEY	CREATED	LAST USED ⓘ	
dbcapy	sk-...vbzw	24 ago 2023	Never	 
dbcapyNew	sk-...JNU6	24 ago 2023	4 sept 2023	 

[+ Create new secret key](#)

Figura 41. Claves personales de OpenAI

Para hacer una llamada de ChatGPT solo debemos indicar nuestra configuración, el prompt que queremos enviar, el modelo con el que queremos que nos resuelvan nuestro prompt, el máximo de tokens que esperamos de vuelta en una llamada y por último el nivel de creatividad que queremos en nuestras respuestas.

6.1.5.3 Modelo de despliegue iconográfico

Ya entendiendo toda nuestra implementación podremos representar nuestro sistema completo de una forma gráfica. Para ello, veremos todas las herramientas utilizadas y explicadas dentro de un modelo de despliegue iconográfico.

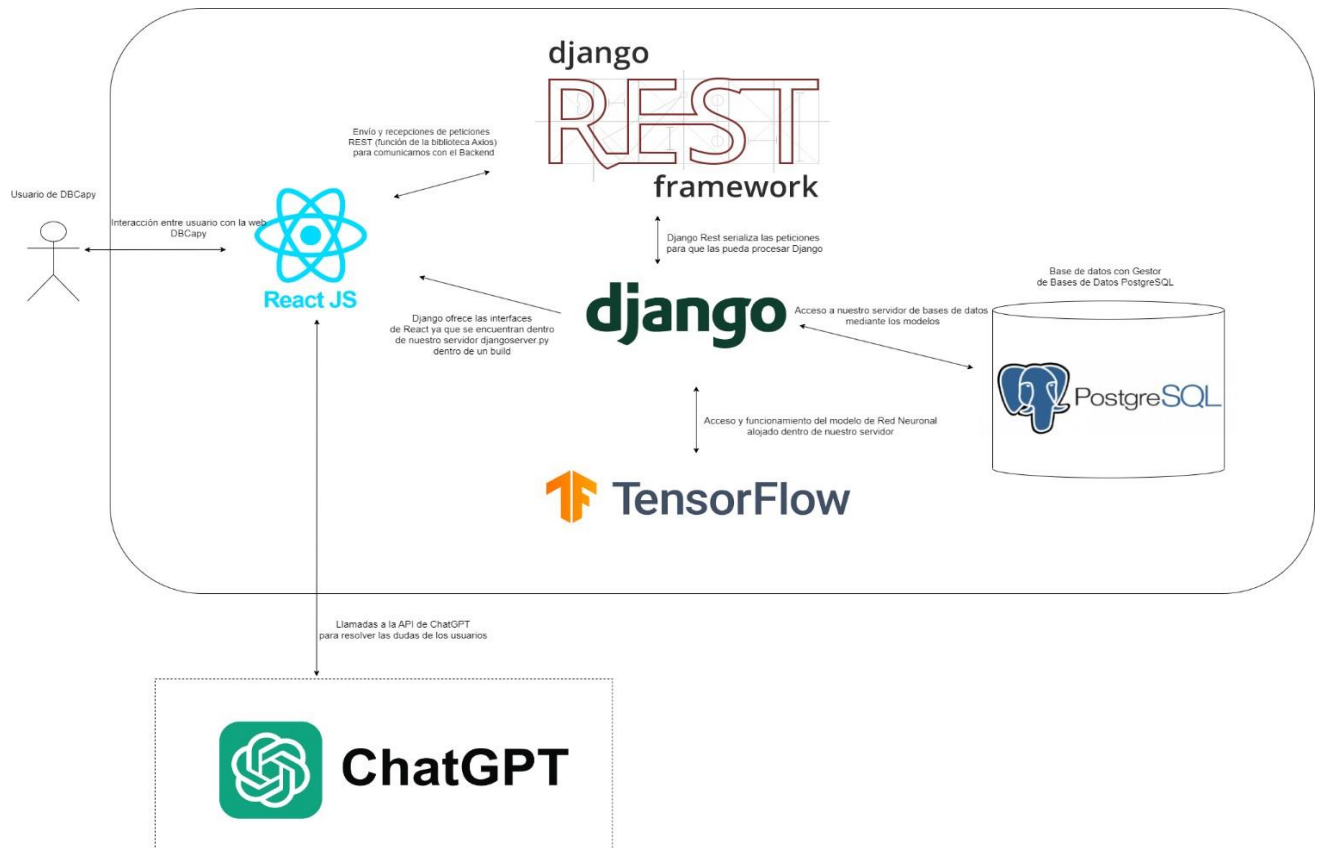


Figura 42. Modelo de despliegue iconográfico o mockup

6.1.6. Pruebas

Las pruebas tienen un ciclo de vida similar a las implementaciones, ya que en las dos primeras iteraciones invertiremos muy poco tiempo en las tareas relacionadas con las pruebas. Mientras el proyecto avanza, la importancia de las pruebas tiene un peso mayor.

A la hora de explicar estas pruebas que se han realizado sobre el proyecto, lo dividiremos en cuatro tipos de pruebas.

6.1.6.1. Pruebas chatbot

Son las pruebas que se realizan desde la iteración segunda a la cuarta, en las que vamos a pasar de pruebas tentativas de este componente, a una refinación del componente integrado al resto de los elementos de nuestra web.

Sobre todo, nos centramos al testear nuestro Chatbot los factores de eficacia, usabilidad y eficiencia.

Dentro de la red neuronal nos enfocamos en la consistencia de las respuestas. Buscamos que las probabilidades de acertar en un mensaje sea la más alta posible. Además, nos enfocamos que la forma de integrar nuestra red neuronal sea fácil de usar y que pueda interconectarse de forma sencilla con las llamadas a ChatGPT.

Las pruebas de ChatGPT se centran en gran parte en la creación de prompts que sean lo más eficientes y correctos posibles. Como se detalló anteriormente, el entendimiento de la ingeniería de prompts es clave para poder solucionar las consultas del usuario de la manera más precisa posible.

6.1.6.2. Pruebas de fiabilidad de respuestas

Entrando en más detalle sobre lo descrito en apartado anterior, vamos a hablar de las diferentes técnicas aplicadas para conseguir una respuesta de ChatGPT que se pueda procesar dentro de nuestros controladores.

Gracias a las diferentes pruebas para crear un prompt eficiente, se determina usar el formato JSON con dos parámetros. Un booleano que se determine dependiendo si una pregunta tiene que ver con bases de datos, y un segundo parámetro que sea la respuesta de la respuesta solo si el booleano es verdadero.

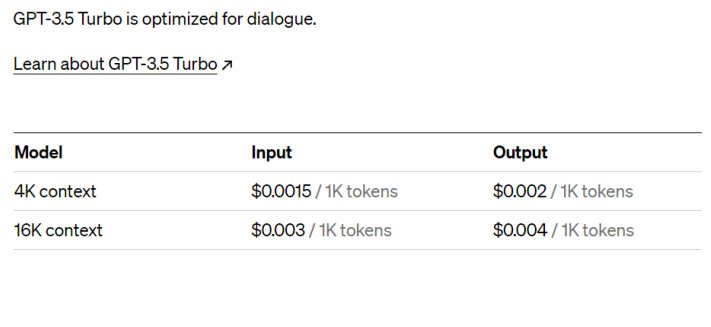
Al realizar estas pruebas de fiabilidad, vamos testear como de certera es la Inteligencia Artificial con detectar diferentes temas de la materia de bases de datos. Nuestro objetivo con estas pruebas que nuestra web siempre pueda procesar los mensajes de forma certera.

6.1.6.3 Pruebas de límites de tokens

A la hora de implementar nuestro componente de Chatbot junto con el resto de sistemas que componen el proyecto, debemos hacer diferentes pruebas relacionadas con dicha integración.

Una de las principales pruebas es la de determinar cómo validar las peticiones que un usuario hace a la API de ChatGPT. Dichas peticiones van a ser validadas dependiendo del tipo de usuario.

Como se ha mencionado anteriormente, el origen de estos límites se debe al coste que supone una llamada a ChatGPT.



GPT-3.5 Turbo is optimized for dialogue.
[Learn about GPT-3.5 Turbo ↗](#)

Model	Input	Output
4K context	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
16K context	\$0.003 / 1K tokens	\$0.004 / 1K tokens

Figura 43. Coste llamadas a ChatGPT 3.5

Probaremos cómo aplicar una limitación tanto en el número de llamadas diarias y al tamaño del mensaje que se realiza.

Para poder mostrar estas limitaciones a los usuarios, se probará con mensajes emergentes dentro de la web.

6.1.6.4. Valoraciones de usuarios

La última tarea que haremos relacionada con las pruebas tiene que ver con la valoración del producto final por parte de los usuarios.

Crearemos una encuesta en Google Forms para poder realizar estadísticas del nivel de satisfacción de nuestro sistema. Para crear nuestra encuesta nos centraremos sobre todo en el componente de Chatbot de la web, ya que es nuestra herramienta más potente dentro del proyecto.

Se realizarán un conjunto de pruebas para elaborar una encuesta que sea clara, fácil de realizar por los usuarios, y que nos dé una valoración objetiva de nuestro sistema. Buscamos no tener preguntas ambiguas dentro del cuestionario.

A la hora de realizar las distintas pruebas para mostrar las funcionalidades de la web, se hace primero una breve introducción de nuestro proyecto, para después mostrarles la página inicial y que puedan interactuar con ella. Nos interesa comprobar la facilidad que tiene nuestro Chatbot de ser usado. También buscamos ver si las respuestas que ofrecemos a través del Chatbot son buenas y satisfacen al usuario. Por último, en nuestra encuesta se pide a los usuarios que puedan dar una opinión más extensa de los aciertos y los fallos que encontraron mientras probaban el sistema.

6.2. Vista

En este último apartado veremos las distintas vistas de nuestro sistema a forma de resumen y para poder ver de forma global todas las decisiones que fueron tomadas. Podremos ver como las ideas vistas en los antecedentes se materializan en el producto final.

Se hablará de los puntos más importantes a la hora de crear las vistas, para ver en más detalle cada una de las vistas y el funcionamiento de ellas se recomienda leer el “Anexo VI Manual de Usuario”.

6.2.1. Vista de usuario

Comenzaremos explicando la primera vista que un cliente que no ha hecho login obtiene al entrar dentro de la web

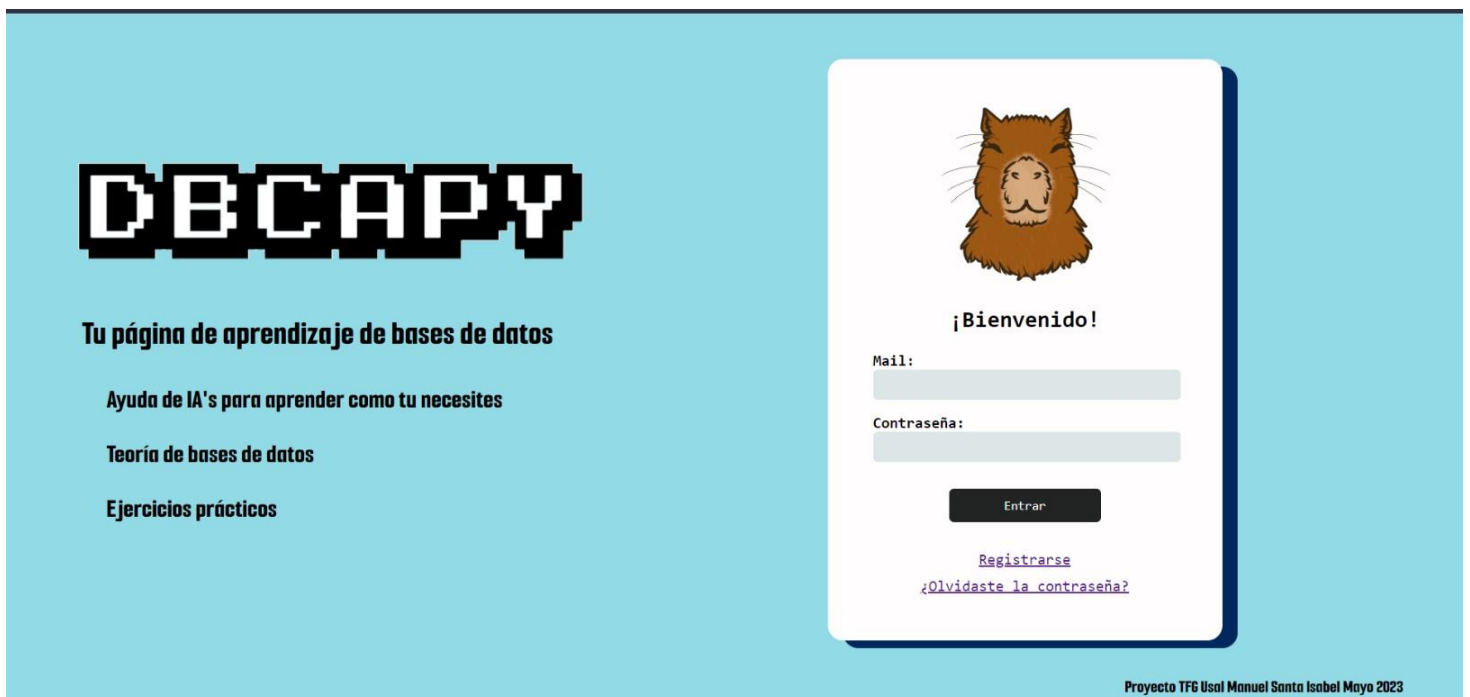


Figura 44. Vista Logging

Lo primero que vamos a destacar es que en la página de inicio ya se hace un breve resumen de lo que nos encontraremos dentro de la web. La frase “Ayuda de IAs para aprender como tú necesitas” recalca como la intención del Chatbot es otorgar una experiencia personalizada a las necesidades de cada estudiante.

6.2.2. Vista de teoría

Se muestra a continuación la vista de teoría, destacándose tanto un índice de los distintos temas de la web, el propio contenido y el componente de Chatbot. Ya que este último componente posee la mayor importancia.

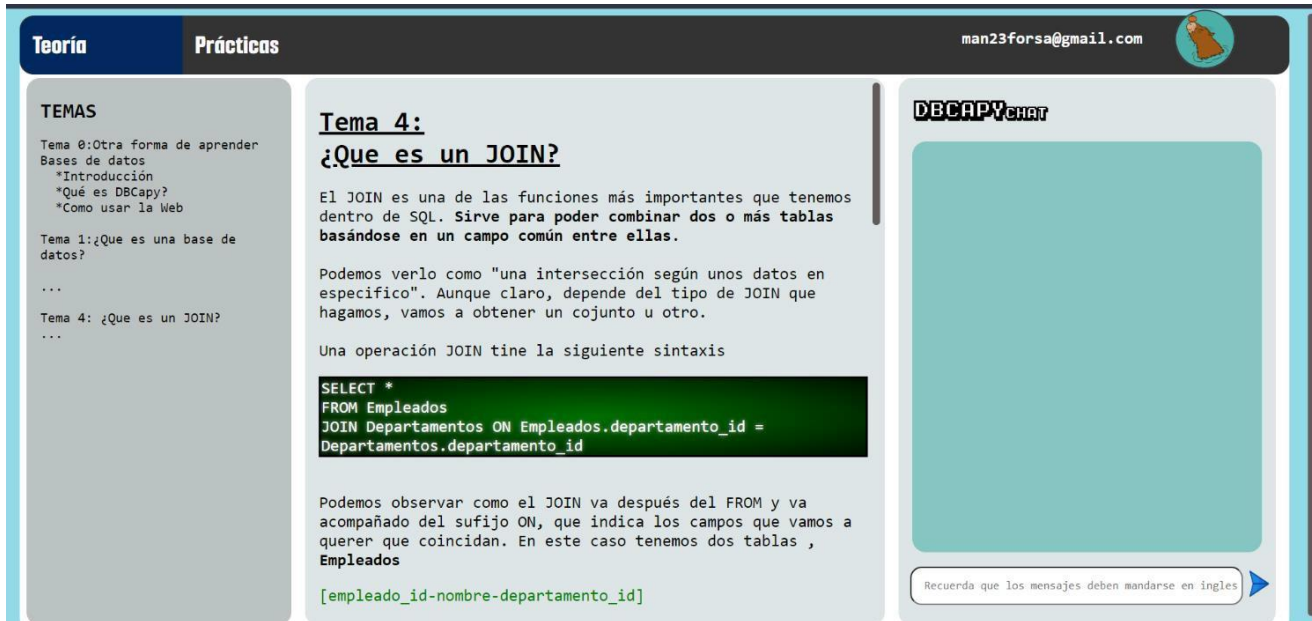


Figura 45. Vista teoría en su totalidad

En las siguientes imágenes de la vista, podemos ver dos ejemplos de una conversación que un usuario puede tener con nuestro Chatbot. Destacamos sobre todo cómo se detecta cuándo una pregunta tiene que ver con el tema de la materia de Bases de datos. Además, es importante ver cómo en la propia caja de texto se aclara que se debe utilizar el idioma en inglés para comunicarse con la Red Neuronal.

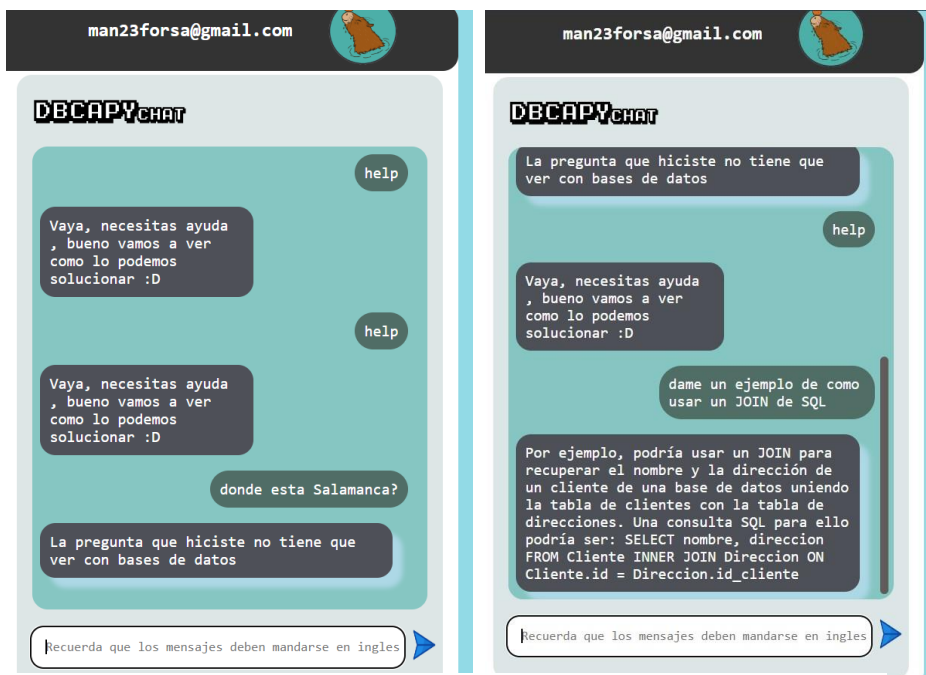


Figura 46. Vista de teoría - Ejemplos de llamadas

6.2.3. Vista de práctica

Dentro de la última vista, podemos observar cómo se ha incluido un diagrama para ayudar a los usuarios con la visualización de los ejercicios prácticos. En este apartado, el Chatbot tiene la funcionalidad de poder corregir las respuestas de los usuarios y de explicar dicha corrección.

Figura 47. Vista práctica

Y, por último, destacamos cómo se vería la respuesta que recibimos por parte del Chatbot integrado en los ejercicios de práctica. En este ejemplo, no solo se nos dice que una respuesta es correcta, sino que también nos explica las razones por las cuales la respuesta es correcta.

Ejercicio 1

Escribe la consulta de SQL que sirva para poder sacar el nombre de cada cliente, el nombre del producto que ha comprado y la fecha de compra. (ten en cuenta que puedes declarar más de un JOIN en una sola declaración, y que hay varias formas de resolver este ejercicio).

INSERT SQL: `SELECT C.nombre AS nombre_cliente, P.nombre AS nombre_p`

Se realiza una consulta con varios JOINS, seleccionando de la tabla Compras (Co) la fila fecha asociada a, en el primer JOIN, la tabla Clientes (C) seleccionando la columna nombre, así como de la tabla Productos (P) seleccionando su columna nombre

Figura 48. Vista práctica - Respuesta chatbot

7. RESULTADOS Y CONCLUSIONES

En este apartado final de la memoria se hablarán sobre los resultados obtenidos a lo largo del proyecto. Destacando tanto los puntos más importantes dentro de nuestro sistema, como el funcionamiento final del Chatbot. Después se mostrarán las conclusiones obtenidas después de realizar todo el proyecto y, por último, se tratarán las líneas de trabajo futuras.

7.1. Resultados

En cuanto al apartado de resultados, vamos a hablar de los efectos después de las tareas de prueba de nuestro sistema.

Primero se han podido observar cómo todas las llamadas que se realizan a la red neuronal (creada con el modelo Bag of Words) siempre se responden de forma determinista, ya que el ámbito de la respuesta se encuentra controlado. Cuando se produzca una petición por parte del usuario que nuestra red neuronal simple no sepa responder, esta red siempre contestará con un mensaje de bienvenida, ya que es el mensaje por defecto.

Hablando ahora sobre nuestra segunda herramienta para el Chatbot, la cual se sirve de las llamadas a ChatGPT, también se han logrado cumplir los objetivos asociados a ella.

Los resultados de las llamadas a ChatGPT desde el apartado de Teoría son muy buenas y capturan perfectamente multitud de temas relacionados con Bases de Datos. Incluso haciendo las pruebas, se han llegado a preguntar dudas muy ambiguas del temario, como por ejemplo “¿qué es una vista?”, respondiendo con la definición de este concepto relativo con bases de datos.

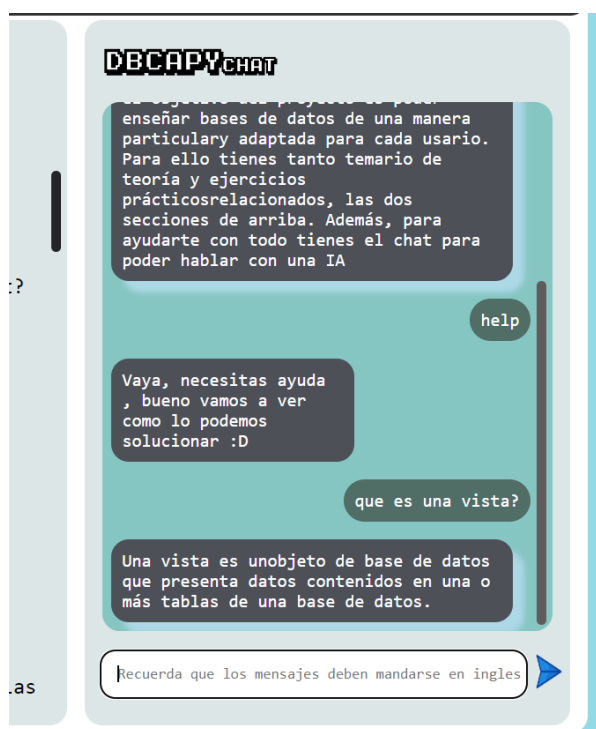


Figura 49. Ejemplo de consistencia en el chatbot

Lo siguiente que veremos son los resultados obtenidos después de realizar las pruebas de validación a distintos usuarios. Analizaremos las repuestas de los usuarios y veremos cuales son las opiniones de nuestra web.

Primero hablaremos del tipo de usuario que ha utilizado nuestra web, en este caso todos los testers han tenido o tienen una relación con la materia de Bases de Datos. Además, destacamos que uno de los usuarios que han probado nuestra web es un profesor de la asignatura.

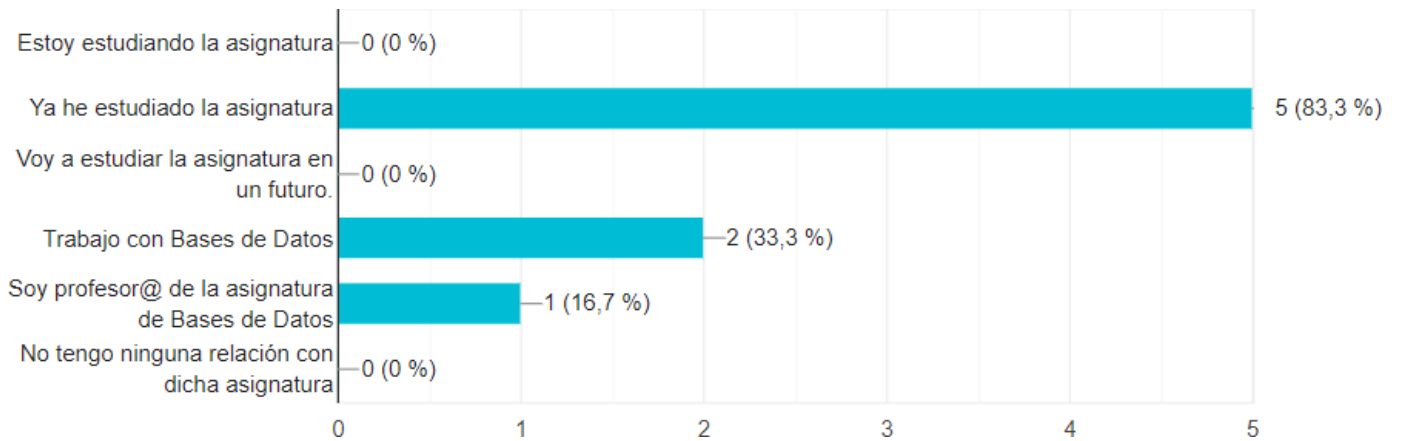


Figura 50. Demográfica de usuarios que han probado la web

La segunda pregunta que se les hacía tiene la intención de poder plantear el contexto de nuestra web y de cómo usar una inteligencia artificial dentro del ámbito de la enseñanza y aprendizaje. La mayoría respondió que eran herramientas muy potentes y que se podían llevar al ámbito de la enseñanza, pero siempre como una herramienta de soporte. Se destaca la respuesta de un usuario sobre este tema:

“Por supuesto, sin embargo, su fiabilidad está por demostrar. Es una herramienta útil para resolver dudas de una manera eficiente, pero sin un 100% de efectividad sí que es cierto que la enseñanza se puede ver comprometida. En informática es un caso particularmente importante demostrar la efectividad de algoritmos y códigos, ya que mínimos errores llevan a dificultades en el aprendizaje, pero sí creo que puede llegar a ser capaz de enseñar las bases de un tema relacionado con la materia de forma apropiada. Sin embargo, no creo que se pueda sustituir por ahora a los profesores físicos, pero es una ayuda innegable que a futuro puede ser más útil.

Creo que la inteligencia artificial tiene un potencial significativo para mejorar la enseñanza en el campo de la informática. Las IA pueden ofrecer un aprendizaje personalizado, retroalimentación instantánea y accesibilidad continua, todo mientras se mantienen al día con las tendencias tecnológicas más recientes. Sin embargo, no deben ser vistas como un reemplazo para la instrucción humana, ya que aún hay aspectos emocionales y éticos que solo un educador humano puede abordar adecuadamente.”

A continuación, se les preguntó sobre la satisfacción de uso de ChatGPT en este proyecto, podemos ver una de esas preguntas en la siguiente figura

4.En este proyecto, se ha usado ChatGPT y una red neuronal para poder solucionar dudas sobre la materia de Bases de Datos. Se ha planteado resol...xplica como usar estas herramientas es correcta?

6 respuestas

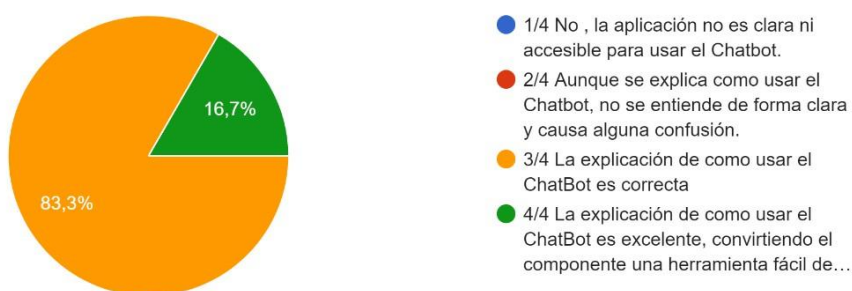


Figura 51. Encuesta de satisfacción - Preguntas de comodidad con ChatGPT

También en otras preguntas relacionadas con la integración de dicho bot dentro de la web, demuestran cómo este componente es muy útil tanto en el apartado Teórico como en el apartado Práctico. Alguna de las opiniones de los usuarios han sido las siguientes:

- “Sí. Pienso que es intuitivo y que responde correctamente a mis cuestiones. Incluso me respondió a una pregunta muy poco específica correctamente.”
- “Me gusta que el chatbot aparezca en la parte derecha de la pantalla, ya que permite visualizar el contenido de diferentes conceptos teóricos y la vez hacer consultas al chatbot.”

- “Sí, da una explicación que va al grano y en caso de pedírselo te da un ejemplo claro y sencillo de las cláusulas sobre las que le preguntes.”

Y también se ve la satisfacción de cómo responde nuestro bot dentro del apartado práctico.

7. Ahora hablaremos de la parte Práctica de la web, se ha optado en vez de usar un Chatbot como en el apartado anterior, solo implementar el uso de...ien integrado esta funcionalidad dentro de la web?

6 respuestas

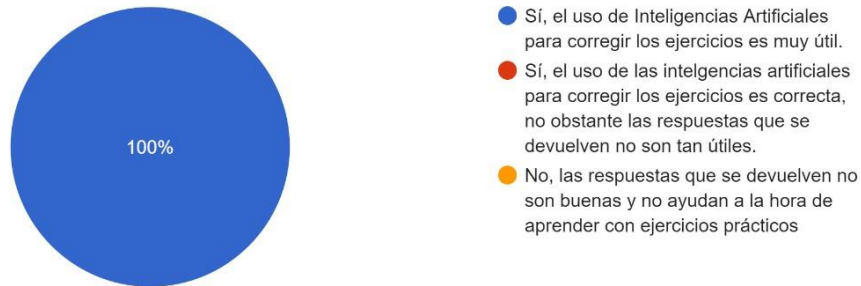


Figura 52. Encuesta de satisfacción – Chatbot

Por último, se pidió a los usuarios que redactasen los puntos fuertes y débiles que encontraban dentro del servicio. Encontramos con una sólida mayoría de opiniones positivas y algún que otro apunte para mejorar en un futuro incluido ideas de funcionalidades que se podrían aplicar a líneas futuras. Una muy buena respuesta que engloba todo lo anterior es la siguiente realizada por un usuario:

“Creo que DBCapy ofrece una solución muy necesaria a una brecha en la educación en línea, especialmente en el campo de la informática y las bases de datos. Una de las características más destacadas del proyecto es su interfaz, que es sencillamente un 10/10 en términos de diseño visual y facilidad de uso. Este enfoque centrado en el usuario hace que el proceso de aprendizaje sea mucho más accesible y atractivo.

Dicho esto, el bot parece estar limitado en ciertos aspectos. Por ejemplo, la necesidad de usar comandos en inglés podría ser una barrera para aquellos que no hablan el idioma con fluidez, aunque las preguntas pueden hacerse en español. Es algo confuso hacer una pregunta en un idioma y que te la devuelvan en otra. Además, el bot está diseñado para ayudar, pero no para proporcionar soluciones directas, lo cual podría ser tanto una ventaja como una desventaja, dependiendo del estilo de aprendizaje del usuario. Me parece que debería especificarse mejor qué clase de preguntas se pueden y no se pueden hacer a la hora de preguntar dudas, utilizando ejemplos más allá del help. Sin embargo, las preguntas sencillas son respondidas de forma muy clara y correcta, y pienso que sí podría utilizarse para el aprendizaje.

En general, el proyecto demuestra cómo la inteligencia artificial puede contribuir a la educación en línea, especialmente en campos técnicos como este. Las limitaciones son puntos menores comparados con las posibilidades que puede llegar a ofrecer.”

7.2. Conclusiones

7.2.1. Conclusiones técnicas

Lo primero que debemos marcar de las conclusiones del proyecto es que se consiguieron satisfacer todos los objetivos de nuestro sistema. En concreto el objetivo más ambicioso del proyecto que es la creación de un ChatBot fue cumplida de forma satisfactoria.

Algunas de las conclusiones a destacar se citan a continuación:

- Integración correcta del Chatbot dentro de nuestro sistema, de esta forma vamos a garantizar que la experiencia de aprendizaje de cada usuario sea personal y se adapte a sus necesidades.
- Incorporación de una herramienta muy novedosa como es ChatGPT dentro del terreno de la enseñanza del temario de Bases de Datos.

7.2.2. Conclusiones de usuarios

Lo siguiente que destacamos dentro de este apartado son las conclusiones obtenidas a partir de las estadísticas de satisfacción de los usuarios, ya expuestas en el apartado de resultados.

Por lo general, podemos ver como nuestro bot está bien integrado dentro del sistema desde el punto de vista de un usuario. Encontramos que la amplia mayoría ha obtenido siempre respuestas por parte del bot que les han ayudado. Por tanto, concluiremos que la integración de nuestro Chatbot ha sido exitosa.

7.3. Líneas de trabajo futuras

Este proyecto tiene un amplio abanico de posibilidades de mejora que se redactarán a continuación:

- Conseguir una red neuronal más potente y que resuelva peticiones en castellano. Este es el primer punto de mejora ya que al final si queremos plantear nuestra web para usuarios de habla hispana, nos interesa que la totalidad de ella pueda usar el chatbot en castellano. Además, nuestra red neuronal puede mejorar creando más y más ejemplos de los temas que queremos tratar para así crear esquemas de diálogos mejores.
- Desplegar la plataforma DBCapy en un servidor externo que aloje nuestro servicio. Esto puede facilitar a otras compañías o a otras herramientas el uso de nuestra plataforma como un software de terceros.
- Ampliar el temario sobre bases de datos. En nuestro proyecto final solo incluíamos dos temas y un tema de ejercicios, por tanto, podría interesarnos ampliamente añadir el temario de la asignatura de Bases de Datos entera.
- Permitir a los usuarios añadir sus propias claves de ChatGPT. Este es un gran añadido que se podría hacer dentro de nuestra aplicación para permitir a los usuarios hacer llamadas a ChatGPT desde nuestra web, pero sin necesidad de tener un límite de llamadas. De este modo los usuarios podrían utilizar nuestra plataforma con su configuración y suscripción a ChatGPT favorita.
- Estar abiertos a la implementación de nuevos modelos de inteligencias artificiales. En caso de que en un futuro decidamos que no queremos usar más la herramienta de ChatGPT, siempre podremos valorar la integración de una IA alternativa.

8. BIBLIOGRAFÍA

- Balderas, A. (5 de 2022). *Chatbot-Based Learning Platform for SQL Training*. Obtenido de <https://reunir.unir.net/handle/123456789/14311>
- Brownlee, J. (7 de 8 de 2019). *Introducción al modelo Bag of Words*. Obtenido de <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
- ChatFuel (2023), *Bot ChatFuel*, Obtenido de <https://chatfuel.com/>
- Chaudray, M. (28 de 8 de 2020). *Entendimiento de funciones de activación*. Obtenido de <https://medium.com/@cmukesh8688/activation-functions-sigmoid-tanh-relu-leaky-relu-softmax-50d3778dcea5#:~:text=Generally%20%2C%20we%20use%20ReLU%20in,use%20in%20last%20output%20layer%20.>
- Código Fuente. (25 de 4 de 2019). *RNA profundas - Tipos y características*. Obtenido de <https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/>
- CSV, D. (30 de 6 de 2020). *RNA orientadas al lenguaje natural (NLP)*. Obtenido de <https://youtu.be/Tg1MjMIVArc?si=L8wNMG48ySI3JEK2>
- Deeplearning.AI. (2023). *Entendimiento de la ingeniería de prompts*. Obtenido de <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>
- IBM Watson (2023), *IBM Watson*, Obtenido de <https://www.ibm.com/es-es/watson>
- InteractiveChaos. (s.f.). *Imagen arquitectura red neuronal*. Obtenido de <https://interactivechaos.com/es/manual/tutorial-de-deep-learning/derivada-parcial-con-respecto-un-peso>
- NeuralNine. (4 de 12 de 2020). *Intelligent AI Chatbot in Python*. Obtenido de (Arquitectura utilizada en el TFG): https://www.youtube.com/watch?v=1lwddP0KUEg&ab_channel=NeuralNine
- NVIDIA. (2023). *Entendimiento del modelo Transformer*. Obtenido de <https://la.blogs.nvidia.com/2022/04/19/que-es-un-modelo-transformer/>
- Socratic. (2023). *Socratic*. Obtenido de <https://socratic.org/>
- OpenAI (2023). *Obtención de las claves y el uso de ChatGPT*. Obtenida de <https://platform.openai.com/>
- Sung, J. (4 de 2023). *Generative Agents: Interactive Simulacra of Human Behavior*. Obtenido de <https://arxiv.org/abs/2304.03442>
- TensorFlow. (2023). *Visualizador de Bedding*. Obtenido de <https://projector.tensorflow.org/>
- W3Schools. (2023). *W3Schools*. Obtenido de <https://www.w3schools.com/>