



**Universidad**  
**Europea** VALENCIA

Grado en Ciencias de Datos

**Sistema de Generación de Consultas SQL  
Basado en Peticiones de Lenguaje Natural  
a Partir de un Chat-Bot**

Presentado por:

**Baltimore Jose Brito Karadjian**

Dirigido por:

**Joaquín Santos Blasco**

CURSO ACADÉMICO 2023-2024

## **AGRADECIMIENTOS**

Quiero expresar mi profundo agradecimiento a mis padres, Mariam y Balmore, por su amor incondicional, apoyo constante y sacrificios incansables a lo largo de mi vida y durante la realización de este trabajo de grado. Su ejemplo de dedicación y perseverancia ha sido mi mayor inspiración, y no puedo agradecerles lo suficiente por todo lo que han hecho por mí.

También deseo agradecer a mis compañeros de clase por su colaboración, amistad y compañerismo durante este arduo viaje académico. Sus contribuciones, ideas y motivación compartida han enriquecido mi experiencia de aprendizaje y me han ayudado a superar los desafíos que surgieron en el camino hacia la culminación de este trabajo.

Agradezco de todo corazón a Dios por Su gracia y misericordia, por darme la fuerza, la sabiduría y la determinación para completar este proyecto. Su guía constante y Su bondad inquebrantable han sido mi luz en los momentos difíciles, y le atribuyo toda la gloria y el reconocimiento por los logros alcanzados. Finalmente, quiero extender mi gratitud a toda mi familia por su amor incondicional, apoyo inquebrantable y ánimo constante. Su aliento y respaldo han sido fundamentales en cada paso de este camino, y estoy eternamente agradecido por su presencia y su amor incondicional

## ÍNDICE

AGRADECIMIENTOS .....	2
ÍNDICE .....	3
ÍNDICE DE FIGURAS .....	6
RESUMEN .....	7
ABSTRACT .....	8
INTRODUCCIÓN .....	9
1 Planteamiento del Problema de Estudio .....	10
1.1 Motivación para el Desarrollo del Proyecto .....	11
1.2 Objetivos del Proyecto.....	11
1.2.1 Objetivo General: .....	11
1.2.2 Objetivos Específicos: .....	12
1.3 Justificación.....	12
2 Marco Conceptual .....	14
2.1 Procesamiento del Lenguaje Natural (NLP) .....	15
2.2 Aprendizaje Automático (ML) .....	16
2.3 Bases de Datos y SQL .....	17
2.4 Ontologías y Vocabularios Controlados.....	19
2.5 Plataformas de Chat-Bot .....	20
2.6 Arquitectura de Software .....	21
2.7 Análisis de Datos.....	22
2.7.1 Análisis Descriptivo .....	23
2.7.2 Análisis Predictivo .....	23
3 Estado del Arte .....	24
4 Análisis de Viabilidad y Factibilidad del Proyecto.....	32
4.1 Análisis de Mercado .....	32
4.2 Análisis Técnico.....	34
4.3 Análisis Económico .....	36
4.4 Análisis Operativo.....	41
4.5 Resumen del Análisis de Factibilidad y Viabilidad .....	46

---

5	Fundamentos Teóricos .....	48
5.1	Procesamiento del Lenguaje Natural (NLP) .....	48
5.1.1	Tokenización.....	48
5.1.2	Análisis Sintáctico .....	48
5.1.3	Análisis Semántico .....	49
5.1.4	Generación de Lenguaje Natural.....	49
5.2	Chat-Bot .....	50
5.2.1	Comprensión del Lenguaje Natural .....	50
5.2.2	Generación de Respuestas .....	50
5.2.3	Interactividad .....	51
5.2.4	LangChain .....	51
5.2.5	OpenAI .....	51
5.2.6	OpenAI y LangChain .....	53
5.3	Streamlit .....	53
5.3.1	Simplicidad de Uso y Sintaxis Clara .....	54
5.3.2	Amplia Gama de Widgets Interactivos .....	54
5.3.3	Cuadros de Texto .....	55
5.3.4	Deslizadores.....	55
5.3.5	Botones .....	55
5.3.6	Menús Desplegables.....	55
5.3.7	Selectores de Fechas.....	56
5.3.8	Checkbox .....	56
5.3.9	Visualización de Datos Avanzada.....	56
5.3.10	Flexibilidad y Personalización.....	57
5.3.11	Despliegue Sencillo y Escalabilidad .....	57
5.4	PostgreSQL .....	57
5.4.1	Arquitectura y Modelo Relacional.....	58
5.4.2	Confiabilidad y Durabilidad.....	58
5.4.3	Escalabilidad y Rendimiento .....	59
5.4.4	Funcionalidades Avanzadas.....	59
5.4.5	Comunidad Activa y Ecosistema de Herramientas.....	59

---

6	Diseño y Metodología.....	60
6.1	Definición de Requisitos del Sistema .....	60
6.1.1	Requisitos Funcionales .....	60
6.1.2	Requisitos No Funcionales.....	60
6.2	Diseño de la Arquitectura del Sistema.....	61
6.3	Desarrollo e Implementación .....	62
6.3.1	Definición de Requisitos .....	62
6.3.2	Selección de Herramientas y Tecnologías.....	63
6.3.3	Desarrollo de la Interfaz de Usuario .....	63
6.3.4	Integración con LangChain.....	63
6.3.5	Conexión con la Base de Datos .....	64
6.3.6	Pruebas y Depuración.....	64
6.3.7	Despliegue y Puesta en Producción.....	64
6.3.8	Experiencia de Usuario .....	65
6.4	Evaluación de Resultados .....	68
6.4.1	Pruebas de Funcionalidad.....	68
6.4.2	Tasa de Éxito.....	68
6.4.3	Fiabilidad del Sistema .....	71
6.4.4	Análisis Estadístico del Sistema.....	72
6.4.5	Análisis Cualitativo del Sistema.....	76
7	Sostenibilidad Ambiental, Social y Económica .....	79
7.1	Dimensión Económica.....	79
7.2	Dimensión Social.....	79
7.3	Aspecto Ambiental.....	80
7.4	Cumplimiento con ODS .....	80
8	Discusión.....	83
9	Conclusiones.....	85
10	Bibliografía .....	87
11	Anexo 1 Código.....	90

**ÍNDICE DE FIGURAS**

Figura 1 Consultas SELECT Multi-Tabla (Alarcón, 2021). .....	18
Figura 2 Ejemplo de base de datos relacional (García-Nieto & et al, 2017).....	19
Figura 3 Arquitectura del Sistema .....	62
Figura 4 Pantalla Inicial.....	66
Figura 5 Pantalla con una consulta formulada en Lenguaje Natural.....	66
Figura 6 LOG del proceso de generación de Código SQL.....	67
Figura 7 Pantalla con respuesta en Lenguaje Natural .....	67
Figura 8 Objetivos de Desarrollo Sostenible .....	82

## RESUMEN

El proyecto consiste en desarrollar un sistema de generación de consultas SQL (Structured Query Language, por sus siglas en inglés) basadas en peticiones de lenguaje natural a partir de un Chat-Bot. El sistema estará enfocado en mejorar la eficiencia y la experiencia del usuario al permitirle formular consultas de manera natural y conversacional.

El sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a través de un Chat-Bot estará integrado por varios componentes, incluyendo una interfaz de usuario con Streamlit, un modelo de lenguaje de OpenAI, herramientas de LangChain para trabajar con consultas SQL, y una base de datos PostgreSQL. Juntos, estos elementos permiten que el Chat-Bot comprenda y responda a las preguntas de los usuarios utilizando consultas SQL ejecutadas en la base de datos.

El sistema estará alojado en una plataforma de Chat-Bot que permitirá la interacción con los usuarios de manera conversacional y natural. Además, se implementará una arquitectura de software escalable y flexible para permitir futuras mejoras y adiciones al sistema.

Se llevará a cabo un análisis de mercado para identificar la competencia, el público objetivo, el mercado potencial y los precios. También se identificarán los canales de distribución y los riesgos y oportunidades del proyecto.

Para la implementación del sistema se utilizará una base de datos de vuelos y se aplicará un análisis de datos para medir y mejorar el rendimiento del sistema y la satisfacción del usuario.

El éxito del proyecto se medirá por la precisión y eficiencia en la generación de consultas SQL a partir de peticiones de lenguaje natural.

## **ABSTRACT**

The project consists of developing a system for generating SQL (Structured Query Language) queries based on natural language requests from a Chat-Bot. The system will focus on improving efficiency and user experience by allowing them to formulate queries in a natural and conversational way.

The system for generating SQL queries based on natural language requests through a Chat-Bot will be made up of several components, including a user interface with Streamlit, an OpenAI language model, LangChain tools for working with SQL queries, and a PostgreSQL database. Together, these elements allow the Chat-Bot to understand and answer user questions using SQL queries executed on the database.

The system will be hosted on a Chat-Bot platform that will allow interaction with users in a conversational and natural way. In addition, a scalable and flexible software architecture will be implemented to allow for future improvements and additions to the system.

A market analysis will be carried out to identify the competition, target audience, potential market, and pricing. Distribution channels and project risks and opportunities will also be identified.

A flight database will be used for the implementation of the system and data analysis will be applied to measure and improve system performance and user satisfaction.

The success of the project will be measured by the accuracy and efficiency in generating SQL queries from natural language requests.

## INTRODUCCIÓN

En la era actual de la tecnología de la información, el acceso y la manipulación de datos son fundamentales para numerosas aplicaciones y procesos en diversos sectores. Las bases de datos juegan un papel crucial al proporcionar un medio para almacenar y gestionar grandes cantidades de información de manera estructurada y eficiente. Sin embargo, para muchas personas, especialmente aquellas que no tienen experiencia técnica en bases de datos o SQL, interactuar con estas bases de datos puede ser intimidante y difícil.

La generación de consultas SQL mediante lenguaje natural y Chat-Bots se presenta como una solución innovadora para abordar esta brecha de habilidades y hacer que la interacción con las bases de datos sea más accesible y fácil de usar para un público más amplio. Este enfoque permite a los usuarios formular consultas utilizando lenguaje humano natural, en lugar de tener que aprender comandos SQL específicos, lo que facilita la recuperación de información de la base de datos de manera rápida y eficiente.

En este contexto, este proyecto tiene como objetivo desarrollar un sistema de generación de consultas SQL utilizando lenguaje natural a través de un Chat-Bot. Se aprovecharán tecnologías avanzadas como el procesamiento del lenguaje natural (NLP), Streamlit, OpenAI, PostgreSQL y LangChain para ofrecer una solución completa y efectiva. Este sistema permitirá a los usuarios interactuar con una base de datos de forma intuitiva y natural, simplemente formulando preguntas o solicitudes en lenguaje humano. Esto abrirá nuevas oportunidades para una amplia gama de usuarios, incluidos aquellos que no tienen experiencia técnica en bases de datos, permitiéndoles aprovechar el poder de los datos de manera más efectiva.

## 1 Planteamiento del Problema de Estudio

Este capítulo tiene como objetivo principal identificar y describir de manera precisa la problemática que se pretende abordar, así como los aspectos relevantes que la rodean y sus objetivos y se centra en describir y analizar detalladamente el problema de estudio que se aborda en el proyecto. Se identificarán las limitaciones actuales en el acceso a la información en bases de datos y se establecerá la necesidad de desarrollar un sistema que permita la generación de consultas SQL a partir de peticiones en lenguaje natural. Este planteamiento sentará las bases para el diseño y desarrollo de la solución propuesta, contribuyendo al avance de la interacción natural entre humanos y máquinas en el ámbito de la consulta a bases de datos.

En la actualidad, el acceso a la información almacenada en bases de datos es crucial para la toma de decisiones en diversos ámbitos profesionales. Sin embargo, muchos usuarios se encuentran limitados por la necesidad de poseer conocimientos especializados en programación y consultas SQL para poder realizar consultas y extraer la información deseada. Esta barrera técnica impide que los usuarios no programadores puedan aprovechar al máximo los beneficios de las bases de datos y tomar decisiones basadas en datos de manera eficiente.

En el contexto actual de la tecnología de la información y el análisis de datos, surge la necesidad de desarrollar sistemas inteligentes que puedan comprender y procesar solicitudes en lenguaje natural para la generación de consultas estructuradas. Este desafío implica diseñar interfaces de usuario intuitivas que permitan a los usuarios interactuar de manera natural, así como desarrollar modelos de lenguaje avanzados capaces de interpretar y traducir preguntas en lenguaje natural en consultas válidas y eficientes. Además, es crucial integrar estos sistemas con bases de datos relevantes para ejecutar las consultas generadas y recuperar los datos solicitados de manera oportuna.

El planteamiento del problema no solo se enfoca en la necesidad de superar las barreras técnicas, sino también en impulsar la democratización del acceso a la

información y promover la toma de decisiones basadas en datos en diversos sectores profesionales. Al eliminar la dependencia de conocimientos especializados en programación, se busca empoderar a los usuarios no programadores y brindarles una herramienta intuitiva y eficiente para acceder y utilizar los datos almacenados en las bases de datos.

## **1.1 Motivación para el Desarrollo del Proyecto**

La motivación principal detrás de este proyecto radica en la creciente demanda de toma de decisiones basadas en datos en diversos sectores y la falta de profesionales especializados en la analítica de datos. En la actualidad, las organizaciones requieren acceder y analizar grandes volúmenes de datos almacenados en bases de datos, pero no todos los usuarios poseen conocimientos técnicos o de programación para realizar consultas SQL, por lo tanto, es necesario diseñar herramientas que permitan a los usuarios no programadores realizar consultas en las bases de datos de forma intuitiva y utilizando lenguaje natural.

## **1.2 Objetivos del Proyecto**

### **1.2.1 Objetivo General:**

El objetivo general de este proyecto es desarrollar un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot.

### 1.2.2 Objetivos Específicos:

- Desarrollar un sistema que permita a los usuarios interactuar con una base de datos de forma intuitiva y eficiente utilizando lenguaje natural para generar consultas SQL.
- Crear un Chat-Bot con capacidad avanzada de procesamiento del lenguaje natural para comprender las consultas de los usuarios y traducirlas en consultas SQL comprensibles.
- Utilizar un modelo de lenguaje avanzado, como OpenAI GPT 3.5 Turbo, para procesar solicitudes en lenguaje natural y generar consultas SQL.
- Emplear LangChain como plataforma de comunicación entre sistemas de información y usuarios a través del procesamiento del lenguaje natural.
- Diseñar una interfaz amigable y atractiva utilizando la biblioteca Streamlit de Python, que permita a los usuarios formular consultas y recibir resultados claramente presentados.
- Configurar un servidor SQL con PostgreSQL para almacenar y administrar los datos utilizados en el sistema.
- Evaluar el rendimiento y la precisión del sistema mediante pruebas exhaustivas, ajustes y optimizaciones para garantizar su efectividad y confiabilidad.
- Asegurar la protección de datos y la privacidad implementando medidas adecuadas conforme a regulaciones relevantes. Es importante registrar de manera adecuada el proceso de desarrollo, los algoritmos empleados, las elecciones de diseño y las pruebas ejecutadas para que el proyecto sea comprensible y replicable.

### 1.3 Justificación

La creación de un sistema de generación de consultas SQL utilizando lenguaje natural y Chat-Bots ofrece numerosos beneficios y aplicaciones prácticas en

diversos campos y sectores. Al hacer que la interacción con las bases de datos sea más accesible y fácil de usar para un público más amplio, se puede aumentar la eficiencia y la productividad en el acceso a la información. Esto puede ser especialmente útil en entornos empresariales, educativos y de investigación, donde el acceso rápido y preciso a los datos es fundamental para la toma de decisiones informadas.

Además, este proyecto también contribuye al avance y la comprensión de tecnologías clave, como el procesamiento del lenguaje natural y la inteligencia artificial, al aplicar estos conceptos en un contexto práctico y relevante. Al desarrollar un sistema completo y funcional, se puede identificar y abordar desafíos y limitaciones en la implementación de sistemas de generación de consultas SQL utilizando lenguaje natural, lo que podría abrir nuevas oportunidades para la investigación y la innovación en este campo.

## 2 Marco Conceptual

El marco conceptual para el proyecto de desarrollo de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot incluye los siguientes elementos:

- **Procesamiento del lenguaje natural (NLP):** técnicas y herramientas para analizar y comprender el lenguaje natural de los usuarios, incluyendo análisis sintáctico, semántico y pragmático (Sahu & Rout, 2020).
- **Aprendizaje automático (ML):** algoritmos y técnicas de aprendizaje automático para entrenar el modelo del Chat-Bot, incluyendo modelos de lenguaje, redes neuronales y algoritmos de clasificación (Xu & Rudnicky, 2019).
- **Bases de datos y SQL:** conocimientos de bases de datos y SQL para generar consultas SQL y acceder a los datos almacenados en la base de datos.
- **Ontologías y vocabularios controlados:** uso de ontologías y vocabularios controlados para mejorar la comprensión del lenguaje natural y generar consultas SQL más precisas.
- **Plataformas de Chat-Bot:** conocimientos de las plataformas de Chat-Bot, incluyendo Dialogflow, Microsoft Bot Framework y Amazon Lex, para diseñar y desarrollar el Chat-Bot.
- **Arquitectura de software:** conocimientos de la arquitectura de software para diseñar y desarrollar un sistema escalable, modular y mantenible.
- **Análisis de datos:** habilidades para analizar los datos generados por el Chat-Bot y mejorar la calidad del modelo de aprendizaje automático (Mohan, Chaitanya, & Kumar, 2021).

## 2.1 Procesamiento del Lenguaje Natural (NLP)

El procesamiento del lenguaje natural (NLP) es una disciplina de la inteligencia artificial y la lingüística computacional que se enfoca en la interacción entre las máquinas y el lenguaje humano. Su propósito principal radica en permitir que las computadoras comprendan, interpreten y generen lenguaje humano de manera fluida (Sahu & Rout, 2020). Para lograr este objetivo, el NLP emplea diversas técnicas, como la tokenización, el etiquetado POS-Tagging (etiquetado gramatical), la resolución de ambigüedades léxicas, la anáfora resuelta, el análisis sintáctico y semántico, así como la generación de texto natural y traducción automática.

- La tokenización es el proceso de dividir una oración en sus componentes básicos, llamados tokens. Los tokens pueden ser palabras, signos de puntuación o cualquier otro elemento que tenga un significado en el lenguaje (Sohail, Shah, & Iqbal, 2019).
- El etiquetado POS-Tagging es el proceso de asignar una categoría gramatical a cada token en una oración, como sustantivos, verbos, adjetivos, etc (Mazzia, Giordano, Giusto, & Giordano, 2018).
- La desambiguación de palabras es el proceso de determinar el significado correcto de una palabra en un contexto determinado. Muchas palabras tienen varios significados, y la desambiguación ayuda a las máquinas a comprender qué significado es el correcto en una oración determinada (Haghighatpanah & Teimouri, 2015).
- La resolución de anáfora es el proceso de identificar a qué se refiere un pronombre o una expresión referencial. Por ejemplo, en la oración “Juan llamó a Pedro porque él estaba enfermo”, es necesario identificar a quién se refiere el pronombre “él” (Mohan, Chaitanya, & Kumar, 2021).
- El análisis sintáctico es el proceso de analizar la estructura de una oración y determinar las relaciones gramaticales entre los diferentes elementos de la oración. Por ejemplo, en la oración “El gato persigue al ratón”, el análisis

sintáctico permite identificar al sujeto (el gato), al objeto directo (el ratón) y al verbo (persigue) (Zhang, Sun A., & Liu, 2019).

- El análisis semántico es el proceso de asignar un significado a una oración en función del contexto en el que se utiliza. Esto implica comprender el significado de las palabras y cómo se combinan para formar una oración significativa (Akram & Raza, 2017).
- La generación de lenguaje natural es el proceso de producir una respuesta en lenguaje humano a partir de una entrada no lingüística, como una consulta SQL (Manning & Schütze, 1999).
- La traducción automática es el proceso de traducir una oración de un idioma a otro de manera automática. Esto implica comprender el significado de las palabras y las estructuras sintácticas en ambos idiomas (Adamopoulou & Moussiades, 2020).

El procesamiento del lenguaje natural es un campo complejo que utiliza una variedad de técnicas y herramientas para permitir que las computadoras comprendan y generen lenguaje humano de manera natural. Es un componente clave en el desarrollo de sistemas de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot (Deokar & Dhok, 2020).

## 2.2 Aprendizaje Automático (ML)

El aprendizaje automático (ML) es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos que permiten a los sistemas informáticos aprender de manera autónoma a través de la experiencia. El objetivo es crear modelos matemáticos y estadísticos que puedan aprender de los datos y mejorar su rendimiento en una tarea específica a medida que se les proporciona más información (Mitchell, 1997).

Los algoritmos de aprendizaje automático se pueden clasificar en tres categorías principales: supervisado, no supervisado y por refuerzo.

- En el aprendizaje supervisado, se utiliza un conjunto de datos de entrenamiento etiquetados para entrenar el modelo y luego se utiliza para predecir etiquetas de datos no vistos.
- En el aprendizaje no supervisado, no hay etiquetas en los datos de entrenamiento, por lo que el modelo tiene que encontrar patrones por sí mismo.
- En el aprendizaje por refuerzo, el modelo interactúa con un entorno dinámico y aprende a través del ensayo y error.

El aprendizaje automático se utiliza en una amplia variedad de aplicaciones, como el reconocimiento de voz, la traducción automática, la detección de fraude, la clasificación de imágenes y la recomendación de productos. También es una herramienta importante en el procesamiento del lenguaje natural (NLP), ya que permite a los modelos entender el significado detrás del lenguaje humano y generar respuestas relevantes y precisas a las solicitudes del usuario.

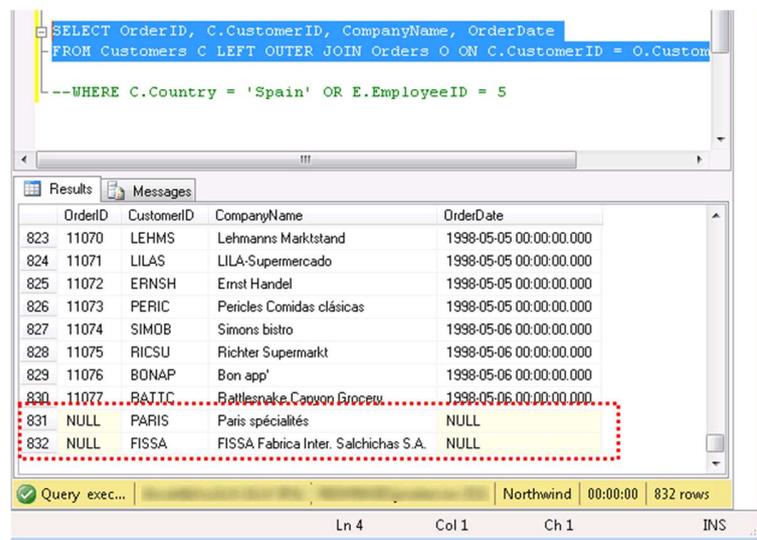
En el contexto de la generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot, el aprendizaje automático puede ser utilizado para entrenar modelos que comprendan el lenguaje natural y generen consultas SQL precisas. El modelo podría ser entrenado con un conjunto de datos de entrenamiento que contenga ejemplos de consultas SQL y solicitudes de lenguaje natural correspondientes, y luego podría utilizarse para generar consultas SQL a partir de solicitudes de lenguaje natural en tiempo real.

### **2.3 Bases de Datos y SQL**

Las bases de datos son sistemas de almacenamiento y organización de datos que permiten la gestión eficiente de grandes cantidades de información. En general, se utilizan para almacenar información que es importante para una empresa o una organización, como datos de clientes, registros financieros o información de inventario (Deokar & Dhok, 2020).

SQL es un lenguaje de programación diseñado para manejar datos en sistemas de Gestión de Bases de Datos Relacionales (RDBMS). SQL permite crear, leer, actualizar y eliminar datos en una base de datos. También permite la creación y modificación de esquemas de base de datos, así como la definición de restricciones y relaciones entre tablas (Spector, 2023).

Una base de datos puede contener varias tablas, que son conjuntos de datos organizados en filas y columnas. Las columnas representan los diferentes tipos de datos que se almacenan en la tabla, mientras que las filas contienen los datos en sí. En SQL, se pueden realizar consultas para buscar información específica en una o varias tablas, filtrando los resultados según ciertos criterios. La Figura 1 muestra una consulta SQL con el comando “SELECT” multi-tabla.



```
SELECT OrderID, C.CustomerID, CompanyName, OrderDate
FROM Customers C LEFT OUTER JOIN Orders O ON C.CustomerID = O.CustomerID
--WHERE C.Country = 'Spain' OR E.EmployeeID = 5
```

OrderID	CustomerID	CompanyName	OrderDate
823	11070	LEHMS	Lehmanns Marktstand
824	11071	LILAS	LILA-Supermercado
825	11072	ERNSH	Ernst Handel
826	11073	PERIC	Pericles Comidas clásicas
827	11074	SIMOB	Simons bistro
828	11075	RICSU	Richter Supermarkt
829	11076	BONAP	Bon app'
830	11077	B&TJC	Battlesnake Canyon Groceries
831	NULL	PARIS	Paris spécialités
832	NULL	FISSA	FISSA Fabrica Inter. Salchichas S.A.

Query exec... Northwind 00:00:00 832 rows

Figura 1 Consultas SELECT Multi-Tabla (Alarcón, 2021).

Las bases de datos relacionales se componen de varias tablas, que se relacionan entre sí a través de claves primarias y externas. Una clave primaria es un campo o conjunto de campos que identifican de forma única a cada fila de una tabla. Una clave externa es un campo que hace referencia a la clave primaria de otra tabla, creando una relación entre las dos tablas.

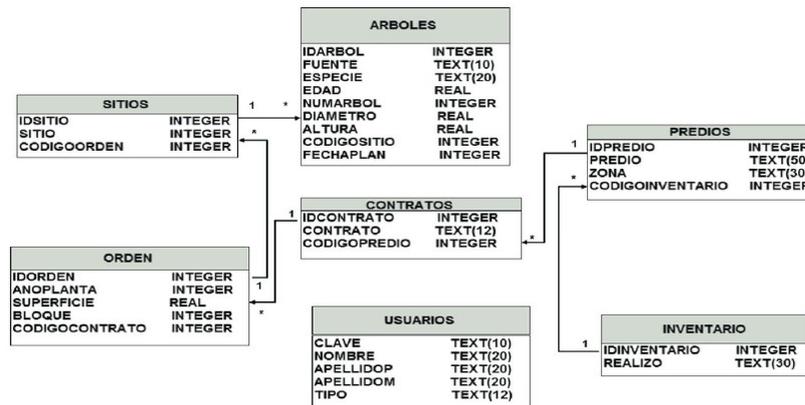


Figura 2 Ejemplo de base de datos relacional (García-Nieto & et al, 2017)

El uso de bases de datos y SQL es fundamental en el desarrollo de sistemas de información que manejen grandes cantidades de datos y permitan la realización de consultas complejas para obtener información específica. En el contexto de un sistema de generación de consultas SQL basadas en lenguaje natural, las técnicas de procesamiento del lenguaje natural se utilizarían para traducir las solicitudes de lenguaje natural en consultas SQL, que se ejecutarían en la base de datos para obtener la información solicitada.

## 2.4 Ontologías y Vocabularios Controlados

Las ontologías y los vocabularios controlados son herramientas importantes en el procesamiento del lenguaje natural (NLP), especialmente para la comprensión del lenguaje natural en aplicaciones de inteligencia artificial y de análisis semántico (Mazzia, Giordano, Giusto, & Giordano, 2018).

- Una ontología es un modelo conceptual que describe una serie de conceptos y las relaciones entre ellos. En el procesamiento del lenguaje natural, se utiliza una ontología para definir un conjunto de términos y sus relaciones semánticas en un dominio específico. Por ejemplo, una ontología puede definir los términos y relaciones semánticas en el dominio de la medicina, permitiendo que un sistema NLP entienda las relaciones entre términos como

“enfermedad”, “síntoma” y “tratamiento” (Silva, Rocha Souza, & Almeida, 2011).

- Por otro lado, los vocabularios controlados son conjuntos de términos que están definidos y estructurados de manera consistente para permitir la identificación y recuperación de información específica en una base de datos. Un vocabulario controlado puede ayudar a mejorar la precisión de los resultados de una búsqueda de información y también puede mejorar la interoperabilidad de sistemas que utilizan diferentes términos para referirse a conceptos similares.

En el contexto de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot, las ontologías y los vocabularios controlados pueden ser utilizados para mejorar la comprensión de las solicitudes de los usuarios y para asegurar que se utilicen los términos correctos en la consulta SQL generada. Por ejemplo, si un usuario solicita información sobre una “enfermedad”, la ontología puede ayudar al sistema a identificar que se trata de un término relacionado con el dominio de la medicina y el vocabulario controlado puede ayudar a identificar los términos correctos para la base de datos.

## **2.5 Plataformas de Chat-Bot**

Las plataformas de Chat-Bot son herramientas que permiten a los desarrolladores crear, entrenar y desplegar Chat-Bots de manera eficiente y rápida. Estas plataformas ofrecen diversas funcionalidades como la integración con servicios de procesamiento del lenguaje natural, la posibilidad de personalizar la apariencia del Chat-Bot, la gestión de la base de conocimiento y la capacidad de integrarse con otras aplicaciones.

Las plataformas de Chat-Bot también ofrecen una variedad de canales de comunicación para el despliegue del Chat-Bot, como Facebook Messenger, Telegram, Skype, WhatsApp y otros. Además, algunas plataformas permiten la integración con la web y aplicaciones móviles.

Algunas de las plataformas de Chat-Bot más populares son:

**Dialogflow:** una plataforma de Chat-Bot de Google que utiliza técnicas de procesamiento del lenguaje natural para entender el lenguaje natural de los usuarios y generar respuestas relevantes (Dialogflow, 2023).

**Microsoft Bot Framework:** una plataforma de Chat-Bot que permite a los desarrolladores crear Chat-Bots para múltiples canales de comunicación, como Skype, Facebook Messenger y Slack, utilizando herramientas de inteligencia artificial y aprendizaje automático (Microsoft, 2029).

**IBM Watson Assistant:** una plataforma de Chat-Bot basada en la nube que utiliza técnicas de procesamiento del lenguaje natural y aprendizaje automático para comprender el lenguaje natural de los usuarios y ofrecer respuestas relevantes (IBM, 2023).

**Amazon Lex:** una plataforma de Chat-Bot de Amazon Web Services que utiliza técnicas de aprendizaje profundo y procesamiento del lenguaje natural para crear Chat-Bots para aplicaciones de voz y texto.

Las plataformas de Chat-Bot ofrecen una solución eficiente y escalable para el desarrollo de Chat-Bots y pueden ayudar a reducir el tiempo y los costos de desarrollo (AWS, 2023).

## 2.6 Arquitectura de Software

La arquitectura de software es el conjunto de decisiones y principios que se utilizan para diseñar y construir un sistema de software. Esta arquitectura establece la estructura, las propiedades y las interacciones de los componentes del software, y define cómo se comunican entre sí y con otros sistemas.

En el contexto del desarrollo de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot, la arquitectura

de software es esencial para garantizar que el sistema sea escalable, flexible y fácil de mantener.

Para desarrollar una arquitectura de software adecuada, es importante definir los requisitos del sistema, identificar los componentes y sus interacciones, y seleccionar las tecnologías y herramientas adecuadas. Es posible utilizar diferentes patrones y estilos arquitectónicos, como la arquitectura en capas, la arquitectura orientada a servicios (SOA) o la arquitectura basada en micro-servicios.

La arquitectura también puede ser influenciada por las restricciones de la infraestructura y la tecnología, así como por los requisitos de seguridad, rendimiento y escalabilidad.

La arquitectura de software es un aspecto fundamental para el desarrollo de sistemas de software efectivos y eficientes. En el contexto del desarrollo de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot, es necesario diseñar una arquitectura que permita la integración de los diferentes componentes y tecnologías involucrados, y que garantice un alto nivel de rendimiento, escalabilidad y seguridad (Manning & Schütze, 1999).

## **2.7 Análisis de Datos**

El análisis de datos es el proceso de inspeccionar, limpiar, transformar y modelar datos con el objetivo de descubrir información útil, llegando a conclusiones y tomando decisiones informadas. Se utiliza en una amplia variedad de campos, desde el análisis financiero y de negocios hasta la investigación científica y la exploración de datos sociales (Jivani & Zaveri, 2018), (Chae, 2014).

En general, el análisis de datos se puede dividir en dos categorías: análisis descriptivo y análisis predictivo.

### **2.7.1 Análisis Descriptivo**

El análisis descriptivo se centra en resumir y describir los datos de manera útil para el usuario, como la media, la mediana y la desviación estándar.

### **2.7.2 Análisis Predictivo**

El análisis predictivo se centra en la predicción de resultados futuros a través de técnicas estadísticas y de aprendizaje automático (Gómez-Gutiérrez & Olivares-Méndez, 2018).

El proceso de análisis de datos generalmente involucra varias etapas, que incluyen:

- Recopilación de datos: esto implica la identificación de fuentes de datos relevantes y la extracción de datos de estas fuentes.
- Pre-procesamiento de datos: esto implica la limpieza, transformación y normalización de datos para que sean más útiles y fáciles de analizar.
- Análisis exploratorio de datos: esto implica la exploración de datos utilizando técnicas estadísticas y visuales para comprender mejor los datos y encontrar patrones y tendencias.
- Análisis de modelos: esto implica la aplicación de técnicas estadísticas y de aprendizaje automático para construir modelos predictivos y analíticos.
- Comunicación de resultados: esto implica la presentación de los resultados del análisis en un formato fácil de entender para el usuario, como tablas, gráficos y visualizaciones.

El análisis de datos es una herramienta poderosa en muchos campos y se utiliza cada vez más en la toma de decisiones informadas en empresas y organizaciones.

### 3 Estado del Arte

La generación de consultas SQL basadas en peticiones de lenguaje natural a través de Chat-Bots es un área de investigación y desarrollo que ha ganado creciente interés en los últimos años. En el ámbito de la inteligencia artificial y el procesamiento del lenguaje natural, se han realizado avances significativos que han permitido abordar este desafío de manera más efectiva. Modelos de lenguaje avanzados, como GPT (Generative Pre-trained Transformer), desarrollados por OpenAI, han demostrado una notable capacidad para comprender y generar texto coherente y relevante. Además, el desarrollo de herramientas y frameworks específicos para integrar estos modelos con bases de datos SQL, así como la creación de interfaces de usuario intuitivas, ha facilitado la implementación de sistemas de generación de consultas SQL basados en peticiones de lenguaje natural. A medida que continúa la investigación en esta área y se exploran nuevas técnicas y enfoques, se espera que estos sistemas se vuelvan aún más sofisticados y accesibles, brindando a los usuarios una herramienta poderosa para interactuar con datos de manera intuitiva y eficiente.

En (Mikolov & et al, 2013), "*Distributed representations of words and phrases and their compositionality*" se desarrollaron sistemas de generación de consultas SQL basadas en peticiones de lenguaje natural a través de Chat-Bots donde se aplicaron diversas técnicas y herramientas de procesamiento del lenguaje natural (PLN), que confirma el hecho de que se trata un tema de investigación en constante evolución y mejora. Actualmente, estas técnicas y herramientas pueden ser probadas y aplicadas en el presente desarrollo para este tipo de sistemas.

Para (Akram & Raza, 2017), "*A review of natural language processing techniques for semantic analysis of e-commerce data*", algunos de los enfoques más comunes incluyen el uso de algoritmos de aprendizaje automático, técnicas de procesamiento del lenguaje natural como el análisis sintáctico y semántico, y el uso de ontologías y vocabularios controlados para mejorar la comprensión del lenguaje

natural, este tipo de algoritmos pueden ser aplicados en el presente proyecto, al ser soluciones prácticas ya ampliamente probadas.

Además, existen diversas aplicaciones comerciales y proyectos de investigación que se centran en este tema, como por ejemplo Google Cloud Natural Language API, Microsoft Azure Cognitive Services, IBM Watson Natural Language Understanding, entre otros. Estas plataformas están disponibles en Internet y pueden ser utilizadas por cualquier persona.

El estado del arte del desarrollo de sistemas de generación de consultas SQL basadas en peticiones de lenguaje natural a través de Chat-Bots está en constante evolución y mejora gracias a la investigación y el desarrollo de nuevas técnicas y herramientas de procesamiento del lenguaje natural.

En (Goldberg, 2017), "*Neural network methods for natural language processing*", se han utilizado técnicas como el procesamiento del lenguaje natural (NLP) y la red neuronal para desarrollar sistemas de generación de consultas SQL basadas en lenguaje natural, en el campo de los algoritmos de aprendizaje automático. Por ejemplo, se han utilizado redes neuronales para generar consultas SQL a partir de oraciones en lenguaje natural y, posteriormente, se han utilizado técnicas de traducción automática para convertir esas consultas en un lenguaje SQL comprensible para una base de datos. También se han utilizado algoritmos de clasificación y regresión para generar consultas SQL a partir de peticiones de lenguaje natural.

Además, se han desarrollado sistemas de Chat-Bot que utilizan técnicas de aprendizaje automático para comprender el lenguaje natural y generar consultas SQL basadas en las solicitudes del usuario. Estos sistemas han sido utilizados en diversos campos, desde atención al cliente hasta análisis de datos empresariales.

En general, el estado del arte del tema sugiere que los algoritmos de aprendizaje automático son una herramienta poderosa para el desarrollo de sistemas de generación de consultas SQL basadas en lenguaje natural, y que hay un gran potencial para la aplicación de estas técnicas en diversos campos.

En la cita (Haghighatpanah & Teimouri, 2015), “*A natural language interface to relational databases using syntactic and semantic parsing*” se utilizan técnicas de procesamiento del lenguaje natural, el análisis sintáctico y semántico, donde el análisis sintáctico se enfoca en comprender la estructura gramatical de una oración, mientras que el análisis semántico se enfoca en el significado de las palabras y cómo se relacionan entre sí. Este desarrollo es de gran relevancia para la aplicación de este trabajo.

De igual manera para (Maldonado, 2012), “Los modelos ocultos de Markov, MOM” existen varias técnicas de procesamiento de lenguaje natural que se utilizan para realizar estas tareas de análisis, como los árboles de análisis sintáctico, los modelos de Markov ocultos, las redes neuronales y el aprendizaje profundo. Estas son bases teóricas que contribuyen a la comprensión del desarrollo del presente sistema. Los árboles de análisis sintáctico son una técnica comúnmente utilizada para analizar la estructura gramatical de las oraciones. Utilizan una estructura de árbol para representar la estructura de la oración, en la que cada nodo representa una parte de la oración (por ejemplo, un sustantivo, un verbo, un adjetivo, etc.) y las ramas representan las relaciones sintácticas entre ellas.

Los modelos de Markov ocultos son otra técnica utilizada en el procesamiento del lenguaje natural, que se utiliza para modelar la probabilidad de secuencias de palabras. Estos modelos se basan en la idea de que la probabilidad de una palabra depende de las palabras anteriores, y se utilizan ampliamente en aplicaciones de procesamiento de lenguaje natural como el reconocimiento de voz y la traducción automática.

Dentro del análisis realizado por (Sharma & Jain, 2019), “Intelligent Chat-Bot for automated disease diagnosis using machine learning”, se hace uso de las redes neuronales como otro enfoque popular para el procesamiento del lenguaje natural, que se basan en la idea de que el cerebro humano procesa el lenguaje a través de la activación de redes de neuronas. Estas redes se utilizan para la traducción automática, el reconocimiento de voz y la generación de texto. Igualmente, este

desarrollo presente una opción relevante dentro del desarrollo presente para ser aplicada al sistema propuesto.

El aprendizaje profundo es una técnica avanzada de procesamiento del lenguaje natural que utiliza redes neuronales de múltiples capas para analizar grandes cantidades de datos. Estas técnicas se utilizan en aplicaciones de Chat-Bot para comprender el lenguaje natural y generar respuestas precisas.

En (Silva, Rocha Souza, & Almeida, 2011), “Ontologies and Controlled Vocabulary”, el uso de ontologías y vocabularios controlados ha sido una técnica popular para mejorar la comprensión del lenguaje natural en sistemas de procesamiento del lenguaje natural y ha sido ampliamente utilizado en el desarrollo de Chat-Bots. Las ontologías son modelos formales que describen las entidades y las relaciones entre ellas en un dominio de conocimiento específico, mientras que los vocabularios controlados son listas de términos controlados y sus relaciones jerárquicas en un dominio específico.

Además, se han desarrollado varios vocabularios controlados específicos para dominios como la medicina, la biología y la ingeniería. Estos vocabularios controlados se han utilizado con éxito en el procesamiento del lenguaje natural para mejorar la precisión de la recuperación de información y la comprensión del lenguaje natural en esos dominios.

En general, el uso de ontologías y vocabularios controlados es una técnica importante para mejorar la comprensión del lenguaje natural en sistemas de procesamiento del lenguaje natural y Chat-Bots. A medida que las tecnologías de procesamiento del lenguaje natural continúan mejorando, es probable que se siga investigando y desarrollando en esta área para mejorar aún más la precisión y la eficacia de los sistemas de Chat-Bot.

En cuanto al estado del arte en el desarrollo de sistemas de Chat-Bot que utilizan técnicas de aprendizaje automático para comprender el lenguaje natural y generar consultas SQL basadas en las solicitudes del usuario, se han realizado varios estudios e investigaciones.

Una de las áreas de investigación en esta área es la utilización de algoritmos de aprendizaje profundo, en particular las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN), para mejorar la comprensión del lenguaje natural en los Chat-Bots, como por ejemplo en (Kim, 2014), "*Convolutional neural networks for sentence classification*". Estos algoritmos permiten una mejor comprensión del contexto y la intención del usuario, lo que puede mejorar la precisión y la eficacia de las consultas SQL generadas. Precisamente se requiere de esta precisión y la eficacia para el desarrollo del sistema que se propone en este trabajo.

Como se mencionó anteriormente, otra técnica utilizada en el procesamiento del lenguaje natural en los Chat-Bots es el análisis sintáctico y semántico (Haghighatpanah & Teimouri, 2015), que permite identificar la estructura de la oración y su significado. Algunos enfoques en esta área incluyen la utilización de árboles de dependencia y gramáticas generativas para mejorar la comprensión del lenguaje natural.

Además, el uso de ontologías y vocabularios controlados puede mejorar la comprensión del lenguaje natural en los Chat-Bots. Estas herramientas proporcionan un conjunto estructurado de términos y relaciones que pueden ser utilizados para mejorar la comprensión del contexto y la intención del usuario.

El estado del arte en el desarrollo de sistemas de Chat-Bot que utilizan técnicas de aprendizaje automático para comprender el lenguaje natural y generar consultas SQL basadas en las solicitudes del usuario se centra en la utilización de algoritmos de aprendizaje profundo, el análisis sintáctico y semántico, y el uso de ontologías y vocabularios controlados para mejorar la comprensión del lenguaje natural y la precisión de las consultas SQL generadas.

Google Cloud Natural Language API (google, 2023) es una herramienta de procesamiento del lenguaje natural desarrollada por Google que utiliza algoritmos de aprendizaje automático para analizar el texto y extraer información útil. Esta API es utilizada tanto en aplicaciones comerciales como en proyectos de investigación.

En el ámbito comercial, Google Cloud Natural Language API se utiliza en diversas industrias, incluyendo la atención al cliente, la investigación de mercado y el análisis de sentimiento. Por ejemplo, una empresa puede utilizar esta API para analizar las opiniones de los clientes en las redes sociales y determinar si tienen una opinión positiva o negativa sobre su producto o servicio. También se puede utilizar para analizar grandes cantidades de datos de texto, como reseñas de productos, comentarios de clientes y transcripciones de llamadas de servicio al cliente.

En el ámbito de la investigación, Google Cloud Natural Language API es utilizada por universidades y centros de investigación para estudiar temas como el análisis de discurso, la clasificación de texto y la extracción de información. Por ejemplo, un equipo de investigación puede utilizar esta API para analizar el lenguaje utilizado en las discusiones políticas en línea y determinar las tendencias y los temas más discutidos.

En resumen, Google Cloud Natural Language API es una herramienta útil en diversas aplicaciones comerciales y proyectos de investigación en el ámbito del procesamiento del lenguaje natural, y su uso puede mejorar significativamente la eficiencia y precisión del análisis de texto.

Microsoft Azure Cognitive Services es un conjunto de servicios de inteligencia artificial (IA) (Microsoft, 2029), que permite a los desarrolladores incorporar capacidades de aprendizaje automático, visión, voz y lenguaje natural en sus aplicaciones. Entre los servicios disponibles, se encuentra el servicio de Text Analytics, que utiliza técnicas de procesamiento de lenguaje natural para extraer información útil de texto no estructurado, como opiniones, emociones y entidades mencionadas en el texto.

El servicio de Language Understanding Intelligence Service (LUIS) es otro servicio de Azure Cognitive Services que permite crear y personalizar modelos de lenguaje natural para comprender las intenciones del usuario a través del procesamiento de lenguaje natural. Además, cuenta con la capacidad de integrarse con Chat-Bots para proporcionar respuestas más precisas y rápidas.

En cuanto a su aplicación comercial, Microsoft ha utilizado Azure Cognitive Services para crear herramientas como el asistente virtual Cortana y la herramienta de traducción de idiomas de Skype. También ha sido utilizado en el sector de la salud para desarrollar soluciones de seguimiento y análisis de la salud basadas en datos médicos no estructurados.

En el ámbito de la investigación, Microsoft ha llevado a cabo proyectos de análisis de sentimientos en redes sociales y ha utilizado sus servicios de lenguaje natural para analizar grandes conjuntos de datos, como los registros de atención médica, para identificar patrones y tendencias y mejorar la eficiencia de los procesos de atención médica.

IBM Watson Natural Language Understanding (IBM, 2023), es una solución empresarial de procesamiento del lenguaje natural que utiliza técnicas avanzadas de aprendizaje automático para analizar y comprender el lenguaje humano. Esta solución permite a las empresas extraer información útil de grandes cantidades de texto, como documentos, correos electrónicos, redes sociales y otros datos no estructurados.

Algunas de las características clave de IBM Watson Natural Language Understanding incluyen:

- **Análisis de sentimiento:** esta característica permite a las empresas analizar y comprender el tono y la actitud de un texto. Puede identificar si un texto es positivo, negativo o neutral, y puede analizar el sentimiento de las opiniones expresadas en el texto.
- **Identificación de entidades:** esta función permite a las empresas identificar y etiquetar diferentes tipos de entidades en un texto, como personas, lugares, organizaciones y productos. También puede detectar relaciones entre estas entidades.
- **Extracción de conceptos:** esta función permite a las empresas extraer los conceptos clave de un texto. Puede identificar los temas principales y los términos relevantes, y puede agrupar los documentos en función de su contenido.

- **Análisis de emociones:** esta característica permite a las empresas analizar y comprender las emociones expresadas en un texto. Puede identificar si un texto es feliz, triste, enojado o sorprendido, entre otras emociones.

IBM Watson Natural Language Understanding se utiliza en una variedad de aplicaciones empresariales, como el análisis de opiniones de los clientes, la gestión de reputación en línea, la clasificación de contenido y la automatización de procesos de negocio. También se utiliza en proyectos de investigación en campos como la medicina y la psicología, donde se utiliza para analizar grandes cantidades de texto y extraer información relevante para el diagnóstico y el tratamiento de enfermedades.

A pesar de estos avances, aún existen desafíos en el desarrollo de sistemas de generación de consultas SQL utilizando lenguaje natural, como la precisión en la comprensión del lenguaje humano, la robustez en la interpretación de consultas complejas y la seguridad en el acceso a datos sensibles. Este proyecto busca abordar algunos de estos desafíos al combinar tecnologías avanzadas como OpenAI, Streamlit, PostgreSQL y LangChain para ofrecer una solución innovadora y eficiente.

## 4 Análisis de Viabilidad y Factibilidad del Proyecto

En el presente capítulo, se realiza un análisis de mercado detallado con el objetivo de evaluar la viabilidad y factibilidad del proyecto. El estudio de factibilidad permite determinar si el proyecto es viable desde diferentes perspectivas, como la técnica, económica y operativa, para tomar decisiones fundamentadas antes de su implementación.

### 4.1 Análisis de Mercado

En esta sección se lleva a cabo un análisis detallado del mercado en el que se planea introducir el sistema de generación de consultas SQL basadas en lenguaje natural. Se capturan datos relevantes sobre la demanda actual en el mercado, la competencia, el público objetivo y las tendencias del sector. A través de este análisis, se tiene una visión clara del entorno en el que se implementará el proyecto y se identifican oportunidades y desafíos.

La evaluación del mercado permite analizar la viabilidad del proyecto desde un punto de vista comercial. A través de este estudio, se obtiene información valiosa sobre la demanda existente, la competencia, el público objetivo y las tendencias del sector, lo que contribuye a determinar si es económicamente viable y a identificar posibles oportunidades y obstáculos.

El análisis de mercado es una herramienta esencial para evaluar la viabilidad del proyecto desde una perspectiva comercial. Proporciona datos clave sobre la demanda, la competencia y las tendencias del mercado, lo que facilita tomar decisiones estratégicas para el desarrollo exitoso del sistema de generación de consultas SQL basadas en lenguaje natural.

El mercado para el sistema propuesto de generación de consultas SQL basadas en peticiones de lenguaje natural a través de Chat-Bots presenta diversas oportunidades y desafíos.

En general, el mercado para el sistema propuesto presenta un gran potencial de crecimiento y oportunidades para aquellos que puedan desarrollar soluciones efectivas que aborden las necesidades y desafíos del mercado de manera integral.

Oportunidades	Desafíos
<p><b>Demanda de Herramientas Intuitivas:</b> Hay una creciente petición de herramientas que posibiliten a los usuarios interactuar con bases de datos y realizar consultas de forma intuitiva, sin requerir conocimientos especializados en SQL. Este sistema aborda esta necesidad al permitir que los usuarios introduzcan consultas en un lenguaje natural.</p>	<p><b>Precisión en la Interpretación del Lenguaje Natural:</b> El mayor desafío consiste en asegurar que el sistema entienda correctamente las solicitudes en lenguaje natural y las convierta con precisión en consultas SQL. Esto implica desarrollar cuidadosamente el modelo de lenguaje y llevar a cabo un entrenamiento exhaustivo en diversas situaciones de uso.</p>
<p><b>Incremento en el Uso de Chat-Bots:</b> El uso de Chat-Bots ha aumentado en diversos sectores, como el comercio electrónico, el servicio al cliente y la atención médica. Incorporar capacidades de generación de consultas SQL a los Chat-Bots amplía su utilidad y relevancia en una variedad de contextos empresariales y de consumo.</p>	<p><b>Integración con Bases de Datos Existentes:</b> Integrar el sistema con bases de datos existentes puede ser complicado debido a diferencias en la estructura de los datos y en los dialectos SQL utilizados. Se necesita un enfoque flexible y compatible para garantizar una integración suave y eficiente.</p>
<p><b>Automatización de Tareas:</b> La automatización de tareas es una tendencia importante en el ámbito empresarial. Este sistema permite automatizar el proceso de generación de consultas SQL, lo que ahorra tiempo y recursos para las empresas que necesitan acceder y analizar grandes volúmenes de datos.</p>	<p><b>Competencia en el Mercado:</b> A medida que la generación de consultas SQL basadas en lenguaje natural gana popularidad, es probable que surjan competidores en el mercado. Diferenciarse y destacarse en un mercado competitivo requerirá innovación continua y un enfoque en la calidad y la usabilidad del sistema.</p>
<p><b>Facilidad de Acceso a la Información:</b> Al simplificar el proceso de consulta de datos, este sistema hace que la información sea</p>	

Oportunidades	Desafíos
más accesible para una variedad de usuarios, desde analistas de datos hasta gerentes y personal no técnico.	

## 4.2 Análisis Técnico

En esta sección se evalúa la viabilidad técnica del proyecto, analizando los recursos necesarios, las tecnologías a utilizar y las capacidades de desarrollo. Se examina la disponibilidad de herramientas y plataformas adecuadas para la implementación del sistema de generación de consultas SQL y se realiza una evaluación de riesgos relacionados con la infraestructura tecnológica.

El análisis técnico permite evaluar la viabilidad técnica del proyecto. En esta etapa, se analizan los recursos, las tecnologías y los procesos necesarios para desarrollar e implementar el sistema de generación de consultas SQL basadas en lenguaje natural. A continuación, se detallan los aspectos clave que se deben considerar en el análisis técnico:

**Modelo de Lenguaje:** La elección y el desarrollo de un modelo de lenguaje avanzado, es fundamental para el éxito del sistema. Se deben considerar factores como la capacidad de comprensión del contexto, la coherencia en la generación de respuestas y la capacidad de adaptación a una amplia variedad de consultas y estilos de lenguaje.

En este contexto, se ha optado por utilizar OpenAI GPT 3.5 Turbo como modelo de lenguaje para comprender y procesar las solicitudes en lenguaje natural. Este modelo se ha elegido debido a su capacidad demostrada para generar respuestas coherentes y relevantes en una variedad de contextos.

**Desarrollo del Componente de Generación de Consultas:** Para ello, se ha desarrollado un componente específico que traduce las solicitudes en lenguaje natural en consultas SQL válidas y eficientes. Este componente utiliza técnicas

avanzadas de procesamiento del lenguaje natural y reglas de gramática para identificar las intenciones de los usuarios y generar consultas SQL adecuadas.

**Integración con Bases de Datos:** La integración con la base de datos existente, requiere un enfoque cuidadoso para garantizar la seguridad, la eficiencia y la compatibilidad. Se deben desarrollar interfaces y métodos de comunicación que permitan al sistema ejecutar consultas SQL de manera segura y recuperar datos relevantes de manera eficiente.

En este sentido, se ha realizado una integración sólida y segura con la base de datos PostgreSQL para ejecutar las consultas generadas y recuperar los datos solicitados. Se han establecido conexiones seguras y métodos de comunicación eficientes para garantizar un acceso rápido y fiable a los datos.

**Seguridad y Privacidad:** Se deben implementar medidas de seguridad robustas para proteger los datos sensibles almacenados en la base de datos y garantizar la privacidad de los usuarios. En el proyecto se han implementado medidas de seguridad sólidas, como cifrado de datos y autenticación de usuarios para proteger la privacidad de los datos y prevenir accesos no autorizados.

**Desarrollo de la Interfaz de Usuario:** La interfaz de usuario es un elemento fundamental del sistema, ya que facilita la principal forma de interacción entre los usuarios y el sistema. Es crucial crear una interfaz amigable e intuitiva que permita a los usuarios hacer preguntas en lenguaje cotidiano, ver respuestas generadas y llevar a cabo acciones adicionales, como eliminar el historial de chat o dar su opinión.

Con base en esta premisa, se ha desarrollado una interfaz de usuario amigable e intuitiva con Streamlit que permite a los usuarios interactuar con el sistema de manera sencilla. Se han integrado características como áreas para introducir preguntas en lenguaje natural, espacios de chat para ver las respuestas generadas y botones para realizar tareas adicionales.

**Pruebas y Optimización:** Se han realizado pruebas exhaustivas en todas las áreas del sistema para garantizar su funcionamiento óptimo y su rendimiento

eficiente. Se han identificado y corregido posibles problemas y se han realizado optimizaciones para mejorar la velocidad y la eficiencia del sistema.

El análisis técnico dentro del estudio de factibilidad tiene como objetivo evaluar la viabilidad técnica del proyecto. Se analizan los requerimientos técnicos, las tecnologías y herramientas disponibles y se evalúa la factibilidad técnica del proyecto. Esto permite tomar decisiones informadas y establecer un plan de desarrollo adecuado para la implementación exitosa del sistema de generación de consultas SQL basadas en lenguaje natural.

### **4.3 Análisis Económico**

En la evaluación económica se examina si el proyecto es financieramente viable. Se calculan los costos de desarrollo, implementación y mantenimiento del sistema, así como los posibles ingresos que podría generar. Se llevan a cabo un análisis exhaustivo de los costos y beneficios previstos, abarcando los gastos relacionados con la infraestructura, el personal y la comercialización. Asimismo, se estudian las proyecciones financieras a largo plazo para determinar la rentabilidad del proyecto.

El análisis económico posibilita evaluar la viabilidad financiera del proyecto que involucra el desarrollo de un sistema para generar consultas SQL basadas en lenguaje natural. Este análisis se enfoca en aspectos económicos y financieros con el fin de establecer si el proyecto será rentable y sostenible.

Para llevar a cabo el análisis económico, se partirá del supuesto de que el sistema será implementado en una empresa mediana; considerando que el proyecto descrito ya está desarrollado y solo requiere adaptarse a las operaciones de la empresa.

A continuación, se describen los puntos clave a tener en cuenta en este análisis económico:

- **Costos de adaptación:** Aquí se deben identificar y estimar todos los costos asociados con la adaptación del sistema. Esto incluye los costos de contratación de personal, adquisición de equipos y software, licencias, capacitación y cualquier otro gasto relacionado con el desarrollo del proyecto. Es importante considerar tanto los costos directos como los costos indirectos o gastos generales.
  - **Contratación de personal técnico:** Supongamos que se contrata un desarrollador durante dos meses para trabajar en el proyecto. El costo total del salario y beneficios para tal desarrollador es de €5,000 al mes, lo que equivale a un total de €10,000.
  - **Licencias de software y herramientas:** La empresa deberá adquirir licencias para Streamlit, OpenAI GPT 3.5 Turbo, LangChain y PostgreSQL por un costo total de €7,000 al mes.
- **Costos de implementación:** Además de los costos de adaptación, se deben considerar los costos asociados con la implementación del sistema. Esto puede incluir la adquisición de servidores, infraestructura de red, licencias de software adicional, configuración y personalización del sistema, entre otros. Estos costos deben ser estimados de manera realista y tener en cuenta posibles contingencias.
  - **Adquisición de hardware y software:** La empresa necesita invertir en servidores y software de servidor para alojar el sistema, lo que tiene un costo inicial de €15,000.
  - **Configuración y puesta en marcha:** Se asigna un presupuesto de €3,000 para la configuración inicial del sistema, la instalación de software y la integración con sistemas existentes.
- **Costos operativos:** Una vez implementado el sistema, es necesario evaluar los costos operativos recurrentes. Esto incluye gastos como el mantenimiento del hardware y software, actualizaciones, costos de alojamiento, servicios en la nube, licencias de software continuas y cualquier otro costo necesario para asegurar el funcionamiento continuo del sistema.

Estos costos deben ser proyectados a lo largo del tiempo y considerar posibles fluctuaciones o incrementos.

- Costos de alojamiento web: La empresa opta por alojar el sistema en la nube, lo que tiene un costo mensual de €500.
- Mantenimiento y soporte técnico: Se asigna un presupuesto anual de €5,000 para el mantenimiento continuo del sistema y el soporte técnico.
- Beneficios esperados: En el análisis económico, es importante identificar y cuantificar los beneficios esperados del proyecto. Estos beneficios pueden ser tanto financieros como no financieros. En el caso de un sistema de generación de consultas SQL basadas en lenguaje natural, los beneficios pueden incluir una mayor eficiencia en la extracción de datos, reducción de errores, ahorro de tiempo y recursos, y una mejor toma de decisiones basada en datos.
  - Ahorros de tiempo y recursos: Se estima que el sistema permite reducir el tiempo dedicado a la generación de consultas SQL en un 80%, lo que equivale a un ahorro anual de €50,000 en costos laborales.
  - Mejora en la toma de decisiones: El acceso más rápido a los datos permite a la empresa tomar decisiones más informadas sobre inventario, precios y estrategias de marketing.
  - Ventaja competitiva: La implementación de tecnología avanzada posiciona a la empresa como líder en innovación en su industria, lo que puede atraer a nuevos clientes y socios comerciales.
- Análisis de rentabilidad: La evaluación económica también debe considerar la rentabilidad del proyecto, lo que implica el cálculo de métricas financieras como el tiempo necesario para recuperar la inversión (PRI), el valor actual neto (VAN), la tasa interna de retorno (TIR) y el índice de rentabilidad. Estas métricas ayudarán a determinar si el proyecto es viable desde el punto de vista financiero y si los beneficios previstos superan los costos asociados.

Para realizar los cálculos del PRI, VAN, TIR e Índice de Rentabilidad del proyecto, se emplearán los siguientes datos:

- Inversión Inicial (I):
  - Costos de contratación de personal técnico: €10,000.
  - Costos de licencias de software y herramientas: €7,000.
  - Adquisición de hardware y software de servidor: €15,000.
  - Configuración y puesta en marcha: €3,000.

**Total de la inversión inicial: €10,000 + €7,000 + €15,000 + €3,000 = €35,000.**

- Costos de Operación Anuales (C):
  - Costo de alojamiento web mensual:  $500 * 12 = €6,000$ .
  - Mantenimiento y soporte técnico anual: €5,000.

**Total de costos operativos anuales: €6,000 + €5,000 = €11,000.**

- Beneficios Anuales (B):
  - Ahorros de tiempo y recursos: €50,000.
- ROI (Return on Investment)

$$ROI = \frac{\text{Beneficio Neto} - \text{Costo Total de la Inversión}}{\text{Costo Total de la Inversión}} \times 100\%$$

Donde,

$$\text{Beneficio Neto} = \text{Beneficios Anuales} - \text{Costos Operativos Anuales}$$

$$ROI = \frac{((€50,000 - €11,000) - €35,000)}{€35,000} \times 100\%$$

**ROI = 11.43%**

- PRI (Periodo de Recuperación de la inversión)

$$PRI = \frac{\text{Inversión Inicial}}{\text{Flujo de Efectivo Anual}}$$

$$PRI = \frac{€35,000}{(€50,000 - €11,000)}$$

**PRI = 0.94 años**

## ➤ VAN (Valor Actual Neto)

Utilizando una tasa de descuento del 10%.

$$VAN = \left( \frac{\sum \text{Flujo de Efectivo}}{(1 + \text{Tasa de Descuento})^n} \right) - \text{Inversion Inicial}$$

$$VAN = \left( \frac{€50,000}{(1 + 0.10)^1} \right) - €35,000$$

**VAN = €10,454.55**

## ➤ Índice de Rentabilidad (IR)

$$IR = \frac{VAN}{\text{Inversion Inicial}}$$

$$IR = \frac{€10,454.55}{€35,000}$$

**Índice de Rentabilidad (IR) = 0.299**

El análisis económico dentro del estudio de factibilidad permite evaluar la viabilidad económica del proyecto. Se deben identificar y estimar los costos de desarrollo, implementación y operativos, así como cuantificar los beneficios esperados. Además, se realiza un análisis de rentabilidad para determinar si el proyecto es financieramente viable. Este análisis brinda información crucial para la toma de decisiones y el establecimiento de estrategias financieras adecuadas para el desarrollo del sistema de generación de consultas SQL basadas en lenguaje natural.

En resumen, el ROI del proyecto es aproximadamente del 11.43%, el VAN es positivo (€10,454.55) y el Índice de Rentabilidad es aproximadamente del 0.299. Estos resultados sugieren que el proyecto es viable desde el punto de vista financiero.

#### 4.4 Análisis Operativo

En esta parte se examina si el proyecto es viable desde el punto de vista operativo, teniendo en cuenta la estructura organizativa, los procesos de funcionamiento y la capacidad de implementación y expansión del sistema. Se determinarán los recursos requeridos para llevar a cabo el proyecto de manera eficaz, y se analizarán los posibles obstáculos y riesgos operativos que puedan surgir durante su desarrollo e implementación.

El análisis operativo nos permite evaluar la viabilidad práctica del sistema que genera consultas SQL a partir de lenguaje natural. En esta fase, se estudian los aspectos vinculados con el funcionamiento diario del sistema y se identifican los recursos necesarios para su correcto desempeño. A continuación, se describen detalladamente los puntos clave a tener en cuenta en este análisis.

- **Procesos operativos:** En el proceso operativo se detallan minuciosamente los procedimientos operativos implicados en el sistema de creación de consultas SQL. Esto incluye comprender la interacción entre el usuario y el sistema, desde la entrada de la consulta en lenguaje natural hasta la generación y presentación de los resultados en formato SQL. En este contexto, se explican los siguientes aspectos en el análisis:
  - La interfaz de usuario creada en Streamlit ha sido sometida a pruebas con usuarios de prueba y se ha mostrado como intuitiva y fácil de utilizar.
  - Los usuarios pueden introducir preguntas en lenguaje natural de forma efectiva y recibir respuestas claras y precisas proporcionadas por el Chat-Bot.
  - El sistema ha sido capaz de comprender y procesar una amplia variedad de preguntas en lenguaje natural con precisión y eficiencia.
  - Se han aplicado técnicas avanzadas de Procesamiento del Lenguaje Natural utilizando el modelo OpenAI GPT 3.5 Turbo y LangChain,

- mejorando significativamente la capacidad del sistema para interpretar consultas complejas.
- La integración con la base de datos PostgreSQL se ha llevado a cabo con éxito, permitiendo al sistema ejecutar consultas SQL eficazmente para recuperar la información solicitada por los usuarios.
  - Se han implementado sólidas medidas de seguridad para salvaguardar la integridad y confidencialidad de los datos almacenados en la base de datos.
  - El sistema ha demostrado ser capaz de gestionar errores y excepciones adecuadamente, ofreciendo mensajes claros ante fallos e indicaciones útiles para ayudar a los usuarios a resolver problemas.
  - Durante las pruebas, se identificaron posibles puntos débiles que fueron corregidos, mejorando así la robustez del sistema.
- Recursos humanos: Es necesario determinar el personal requerido para la operación del sistema. Esto puede incluir roles como desarrolladores, analistas de datos, administradores de bases de datos, expertos en lenguaje natural y personal de soporte técnico. Se deben definir las responsabilidades y las competencias necesarias para cada puesto, así como estimar la carga de trabajo y los costos asociados con el personal.
    - Se precisará al menos un programador experimentado en desarrollo de software, procesamiento del lenguaje natural (NLP) y bases de datos para diseñar, crear e implementar el sistema. Esta persona será responsable de establecer la lógica del sistema, integrar las distintas herramientas y tecnologías utilizadas y asegurar su correcto funcionamiento.
    - Es posible que se requiera contar con expertos en infraestructura informática y seguridad cibernética para configurar y mantener la infraestructura necesaria, garantizar la disponibilidad y seguridad del sistema, así como hacer frente a posibles amenazas y vulnerabilidades.

- Será necesario personal de soporte técnico para asistir a los usuarios finales en caso de problemas técnicos, responder consultas relacionadas con el sistema y brindar orientación sobre su uso adecuado. Este equipo podría incluir técnicos de soporte en línea, personal de atención al cliente o especialistas en el producto.
- Podría ser preciso tener personal dedicado a la formación para instruir a los usuarios finales sobre cómo utilizar eficazmente el sistema, cómo plantear preguntas pertinentes y cómo interpretar las respuestas generadas por el Chat-Bot. Estos profesionales podrían encargarse de elaborar materiales formativos como manuales de usuario, tutoriales en vídeo y sesiones formativas presenciales o virtuales.
- Será necesario contar con un administrador de bases de datos que se encargue de la gestión y mantenimiento de la base de datos subyacente utilizada por el sistema, realizar copias de seguridad y restaurar datos, optimizar el rendimiento y garantizar la integridad y seguridad de los datos almacenados. Además, puede ser requerido un equipo de gestión del proyecto para supervisar y coordinar todas las actividades relacionadas con el desarrollo, implementación y funcionamiento del sistema, asegurando el cumplimiento de plazos, presupuestos y requisitos del proyecto.
- Infraestructura tecnológica: Aquí se identifican los recursos tecnológicos necesarios para la operación del sistema. Esto incluye servidores, equipos de almacenamiento, sistemas de seguridad, licencias de software, herramientas de desarrollo y otros componentes necesarios para garantizar el funcionamiento adecuado del sistema. Es importante evaluar las capacidades y las limitaciones de la infraestructura existente y determinar si es necesario realizar inversiones adicionales.
  - Se necesitará hardware de servidor adecuado para alojar el sistema, junto con el software de servidor necesario, como sistemas operativos y software de gestión de bases de datos.

- Esto podría implicar la adquisición de servidores físicos o la utilización de servicios en la nube, como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform (GCP).
- Serán necesarias licencias para las herramientas utilizadas en el desarrollo y ejecución del sistema, incluyendo Streamlit, OpenAI GPT 3.5 Turbo, LangChain y PostgreSQL.
- Estas licencias podrían ser adquiridas directamente a los proveedores o a través de distribuidores autorizados.
- Se requerirá una conexión a Internet estable y veloz para asegurar el acceso al sistema y la comunicación con los servidores en la nube si corresponde.
- Los usuarios finales deberán contar con dispositivos como laptops, desktops o móviles para acceder al sistema mediante la interfaz proporcionada por Streamlit.
- Además, se tendrá que implementar un sistema de respaldo y recuperación para proteger los datos contra pérdidas y garantizar la continuidad del negocio ante fallos del sistema o desastres.
- Se deben tomar medidas de seguridad apropiadas, como firewalls, sistemas de detección de intrusiones y encriptación de datos, para resguardar la infraestructura y la información del sistema ante amenazas externas y ciberataques.
- Es necesario diseñar la infraestructura con capacidad de escalabilidad para adaptarse a posibles aumentos en la demanda de usuarios y datos en el futuro, lo que podría implicar el uso de tecnologías como la virtualización y la contenerización.
- Se precisará contar con un entorno separado para desarrollo y pruebas a fin de crear y testear nuevas funcionalidades del sistema antes de implementarlas en el entorno productivo.
- **Capacitación y documentación:** Es fundamental tener en cuenta la formación del personal que participa en el funcionamiento del sistema. Esto implica brindarles instrucción sobre cómo utilizar el sistema, comprender los

procedimientos operativos, solucionar problemas y realizar tareas de mantenimiento básicas. Asimismo, es necesario crear una documentación detallada y clara que sirva como guía para el personal, facilitando así la operatividad y el mantenimiento del sistema. En lo que respecta a la formación y documentación para el sistema propuesto, resulta crucial implementar las estrategias siguientes:

- Desarrollar programas de capacitación adaptados a diferentes niveles de habilidad y roles de usuario.
- Ofrecer sesiones de capacitación presenciales o virtuales, así como tutoriales en video y recursos en línea.
- Cubrir temas como la navegación en la interfaz de usuario, la formulación de preguntas efectivas, la interpretación de respuestas del Chat-Bot y la resolución de problemas comunes.
- Proporcionar oportunidades para practicar y obtener retroalimentación sobre el uso del sistema.
- Ofrecer capacitación exhaustiva sobre el funcionamiento del sistema, las herramientas utilizadas y los procedimientos de resolución de problemas al personal de soporte técnico.
- Capacitar al personal técnico en la identificación y solución de problemas técnicos, la gestión de incidencias y la comunicación efectiva con los usuarios finales.
- Proporcionar al personal técnico con acceso a recursos de referencia y materiales de capacitación actualizados para mantenerse al día con los cambios en el sistema.
- Desarrollar manuales de usuario detallados que cubran todas las funcionalidades del sistema, incluyendo capturas de pantalla, instrucciones paso a paso y ejemplos prácticos.
- Organizar la documentación de manera clara y estructurada, utilizando un lenguaje sencillo y comprensible para los usuarios finales.

- Incluir secciones de preguntas frecuentes (FAQs) y solución de problemas para ayudar a los usuarios a encontrar respuestas rápidamente.
- Crear documentación técnica exhaustiva para el personal de desarrollo, soporte técnico y administradores del sistema.
- Establecer un portal de soporte en línea donde los usuarios puedan acceder a recursos de capacitación, documentación, preguntas frecuentes y solicitudes de asistencia.
- Realizar evaluaciones periódicas de la efectividad de la capacitación y la documentación a través de encuestas, cuestionarios y sesiones de retroalimentación.

El análisis operativo en el estudio de viabilidad del sistema para generar consultas SQL mediante lenguaje natural se ha enfocado en evaluar los procesos operativos, el personal involucrado, la infraestructura tecnológica, los procedimientos de apoyo y la formación del equipo. Esta evaluación asegura que el sistema opere de manera eficiente y viable tanto a corto como a largo plazo.

#### **4.5 Resumen del Análisis de Factibilidad y Viabilidad**

Tras realizar un exhaustivo análisis en diferentes áreas como el mercado, lo técnico, lo económico y lo operativo del proyecto propuesto para implementar un sistema de generación de consultas SQL basadas en lenguaje natural a través de un Chat-Bot, podemos concluir que el proyecto es viable y factible.

Desde la perspectiva del análisis de mercado, se han identificado claras necesidades y demandas en el mercado para una solución que permita a las empresas acceder y analizar datos de forma más intuitiva y eficiente mediante el uso del lenguaje natural. Esto sugiere un alto potencial de aceptación y adopción del sistema por parte de las empresas, respaldando así su viabilidad comercial.

El análisis técnico ha revelado que el sistema propuesto cuenta con una sólida infraestructura y una arquitectura bien diseñada que permite el procesamiento eficiente de consultas en lenguaje natural, la integración con bases de datos y la entrega precisa y útil de respuestas a los usuarios. Además, se han identificado soluciones tecnológicas adecuadas y se han establecido procedimientos operativos efectivos para garantizar el óptimo funcionamiento del sistema.

Desde un punto de vista económico, los análisis indican que el proyecto ofrece un retorno positivo sobre la inversión realizada, con una relación costo-beneficio favorable.

A pesar de la inversión inicial requerida en hardware, software y personal, los ahorros en costos laborales y la mejora en la eficiencia operativa compensan estos gastos y generan un retorno financiero atractivo a largo plazo. En última instancia, el análisis operativo ha demostrado que el sistema cuenta con procedimientos sólidos, una capacitación adecuada y un sistema de soporte confiable que asegura su funcionamiento efectivo y la satisfacción del usuario. Se han implementado medidas para abordar posibles desafíos operativos y se han diseñado estrategias para mantener el sistema actualizado y continuamente optimizado.

En resumen, estos análisis respaldan la viabilidad y factibilidad del proyecto propuesto, destacando su potencial para brindar beneficios tangibles tanto a las empresas usuarias como a los desarrolladores y proveedores del sistema.

Con un mercado receptivo, una base técnica sólida, un análisis económico sólido y una operación eficaz, el proyecto se vislumbra como una inversión estratégica y prometedora para todas las partes interesadas.

## 5 Fundamentos Teóricos

### 5.1 Procesamiento del Lenguaje Natural (NLP)

El Procesamiento del Lenguaje Natural (PLN) es un área de la inteligencia artificial que se centra en capacitar a las computadoras para comprender, interpretar y generar lenguaje humano de manera efectiva. Dentro de este proyecto, el PLN juega un papel crucial en la comprensión de las consultas expresadas por los usuarios en lenguaje natural y su transformación a consultas SQL entendibles por el sistema de gestión de bases de datos.

#### 5.1.1 Tokenización

La Tokenización es el procedimiento de fragmentar un texto en unidades más pequeñas, conocidas como tokens. Estos tokens pueden representar palabras individuales, signos de puntuación o incluso oraciones completas. La segmentación constituye el paso inicial en varios procesos de procesamiento del lenguaje natural, ya que posibilita descomponer el texto en elementos más manejables para su análisis posterior.

Por ejemplo, la pregunta “¿Cuál es la marca de automóviles más popular?” podría descomponerse en los siguientes tokens: “¿”, “Cuál”, “es”, “la”, “marca”, “de”, “automóviles”, “más”, “popular”, “?”.

#### 5.1.2 Análisis Sintáctico

El análisis sintáctico, también llamado parsing, implica examinar la estructura gramatical de una oración para entender la relación entre las palabras y sus funciones en la frase. Esto incluye reconocer partes de la oración como sujetos,

verbos, objetos, entre otros. El análisis sintáctico es fundamental para comprender la estructura y el significado de una pregunta en lenguaje natural.

Por ejemplo, en la frase “Encuentra a todos los clientes que han comprado este mes”, el análisis sintáctico ayudaría a identificar que “clientes” es el sujeto de la oración, “han comprado” es el verbo y “este mes” es el objeto de la acción.

### **5.1.3 Análisis Semántico**

El análisis semántico se enfoca en entender el significado de las palabras y frases en un contexto específico. Esto incluye interpretar tanto el significado literal como figurado del lenguaje humano, así como reconocer entidades y conceptos particulares dentro del texto. El análisis semántico es crucial para convertir la intención detrás de una pregunta en lenguaje natural en una representación que la máquina pueda comprender.

Por ejemplo, al hacer la pregunta “¿Cuáles son los artículos más populares?”, el análisis semántico ayudaría a reconocer que el usuario está solicitando una lista de productos y su nivel de popularidad en ventas.

### **5.1.4 Generación de Lenguaje Natural**

La creación de texto natural implica la generación de contenido comprensible para los seres humanos a partir de datos estructurados o representaciones semánticas. En este proyecto, se utiliza para elaborar respuestas en lenguaje humano basadas en resultados de consultas SQL o para solicitar aclaraciones al usuario según sea necesario.

Por ejemplo, si un usuario pide ver “los productos más vendidos”, la creación de texto natural podría responder con “Estos son los productos preferidos por nuestros clientes”.

## **5.2 Chat-Bot**

Un Chat-Bot es una aplicación diseñada para interactuar con seres humanos a través de diálogos en lenguaje natural. Estos sistemas utilizan métodos de procesamiento del lenguaje natural (PLN) y aprendizaje automático para comprender y dar respuestas inteligentes a las consultas de los usuarios. A continuación, se enumeran algunas características clave de los Chat-Bots:

### **5.2.1 Comprensión del Lenguaje Natural**

La comprensión del lenguaje natural es esencial para los Chat-Bots, ya que les permite interpretar las preguntas de los usuarios de manera efectiva. Esto implica examinar el texto del usuario para entender la intención detrás de las palabras y reconocer las entidades clave en la consulta. Los Chat-Bots utilizan algoritmos de Procesamiento del Lenguaje Natural (NLP) para analizar el texto del usuario y extraer información relevante que les permita responder con precisión y relevancia.

### **5.2.2 Generación de Respuestas**

Una vez que el Chat-Bot comprende la consulta del usuario, debe ser capaz de generar una respuesta adecuada y coherente. Esto implica recuperar la información necesaria de la base de datos u otras fuentes de datos y presentarla de manera comprensible para el usuario. La generación de respuestas puede variar en complejidad, desde proporcionar información directa hasta realizar cálculos o ejecutar acciones específicas según las necesidades del usuario. Los Chat-Bots utilizan técnicas de generación de lenguaje natural para crear respuestas que sean relevantes y útiles para el usuario.

### 5.2.3 Interactividad

La capacidad de interactuar es un aspecto fundamental en los Chat-Bots, ya que les permite sostener conversaciones fluidas y naturales con los usuarios. Esto implica tener en cuenta el historial de la conversación y recordar la información pertinente para ofrecer respuestas coherentes y útiles. Los Chat-Bots deben poder responder a diversas preguntas y mantener el contexto de la conversación para crear una experiencia satisfactoria para el usuario. Para lograr esto, emplean técnicas de gestión de diálogos y seguimiento de estados para mantener la coherencia y relevancia en la interacción con el usuario.

### 5.2.4 LangChain

LangChain es una plataforma que brinda herramientas para crear agentes de diálogo e integrar sistemas de información a través de APIs. Su objetivo es facilitar la comunicación entre sistemas de información y aplicaciones para los usuarios finales, permitiendo la interacción mediante un lenguaje natural.

Dedicada a desarrollar agentes de diálogo e integrar sistemas de información a través de APIs, se enfoca en simplificar la comunicación entre los sistemas y los usuarios, empleando el procesamiento del lenguaje natural (NLP). Sin embargo, transformar consultas en lenguaje natural a consultas SQL es un desafío que requiere comprender profundamente el contexto y la semántica del texto, así como traducir con precisión las intenciones del usuario en consultas estructuradas en SQL.

### 5.2.5 OpenAI

El Gran Modelo de Lenguaje de OpenAI, también conocido como LLM, es una creación de OpenAI que utiliza técnicas avanzadas de aprendizaje automático,

especialmente en el campo del aprendizaje profundo, para comprender y generar texto en lenguaje natural con coherencia y relevancia. El código subyacente que sustenta la funcionalidad del LLM de OpenAI es complejo y altamente especializado, basado en redes neuronales profundas. En líneas generales, la operatividad del código del LLM de OpenAI se puede resumir en los siguientes pasos:

**Entrenamiento del Modelo:** Durante la etapa de formación, el modelo se alimenta con una gran cantidad de datos escritos procedentes de diversas fuentes, como libros, noticias, páginas web y conversaciones. A través de estos datos, el modelo adquiere la capacidad de identificar patrones y estructuras del lenguaje humano.

**Arquitectura del Modelo:** El diseño del lenguaje de OpenAI LLM se apoya en una estructura de aprendizaje profundo, como la de las redes neuronales transformadoras (Transformers). Esta estructura facilita que el modelo pueda comprender conexiones complejas y extensas entre las palabras y los contextos en los que se presentan.

**Comprensión del Texto de Entrada:** Cuando se proporciona un texto de entrada al modelo, este pasa por una serie de capas neuronales que lo procesan y lo representan de manera distribuida en un espacio vectorial de alta dimensionalidad. Esta representación vectorial captura el significado y el contexto del texto de entrada.

**Generación de Texto de Salida:** Una vez que se ha procesado y representado el texto de entrada en un espacio vectorial, el modelo utiliza esta representación para crear texto de salida. Esto puede implicar completar frases, responder preguntas, generar contenido creativo o, en el caso concreto del proyecto propuesto, convertir consultas en lenguaje natural a consultas SQL.

**Evaluación y Ajuste:** Después de crear el texto final, es posible evaluar el modelo según su coherencia, relevancia y precisión. En ocasiones, pueden realizarse modificaciones en el modelo o en los datos de entrenamiento para mejorar su eficacia.

### 5.2.6 OpenAI y LangChain

La colaboración entre LangChain y OpenAI LLM se basa en su capacidad para complementarse en el desarrollo de sistemas de Chat-Bot que interactúan con bases de datos utilizando lenguaje natural. Mientras LangChain facilita la comunicación entre el Chat-Bot y la base de datos, OpenAI LLM puede comprender consultas en lenguaje natural y generar consultas SQL correspondientes. Juntos, estos dos componentes pueden formar un sistema completo que permite a los usuarios interactuar intuitivamente con bases de datos a través de Chat-Bots.

LangChain, como plataforma para la creación de agentes conversacionales e integración de sistemas informativos mediante interfaces API, se enfoca en facilitar la comunicación entre sistemas informativos y usuarios finales a través del procesamiento del lenguaje natural (NLP). Sin embargo, la generación precisa de consultas SQL desde el lenguaje natural es una tarea desafiante que requiere una comprensión profunda del contexto y semántica del texto, así como habilidades para traducir con precisión las intenciones del usuario en consultas estructuradas en SQL.

Es aquí donde entra OpenAI. Esta entidad es reconocida por sus avanzados modelos de lenguaje, como GPT-3.5 Turbo y otros LLM (Large Language Model), los cuales han sido entrenados con grandes volúmenes de datos textuales y poseen la capacidad para comprender y producir texto. Al combinar OpenAI con LangChain, se aprovecha la capacidad de los modelos lingüísticos de OpenAI para comprender las consultas en lenguaje natural y generar automáticamente las consultas SQL correspondientes.

### 5.3 Streamlit

Streamlit es una herramienta en Python que ha ganado popularidad por su habilidad para facilitar la creación de aplicaciones web interactivas. Proporciona

diversas funcionalidades y recursos que posibilitan a los programadores diseñar interfaces de usuario fáciles de usar y atractivas para la visualización y estudio de datos. Algunas de las características sobresalientes de Streamlit son:

### **5.3.1 Simplicidad de Uso y Sintaxis Clara**

Una de las características más notables de Streamlit es lo sencillo que resulta utilizarlo. La organización de Streamlit es clara y directa, lo que permite a los desarrolladores crear aplicaciones web interactivas con solo unas pocas líneas de código en Python. Esto hace que sea una herramienta accesible incluso para aquellos sin experiencia previa en el desarrollo web. Los programadores pueden enfocarse en la lógica de la aplicación sin tener que lidiar con los aspectos técnicos más complicados, lo que agiliza el proceso de desarrollo y facilita la curva de aprendizaje.

### **5.3.2 Amplia Gama de Widgets Interactivos**

Streamlit ofrece una amplia gama de elementos interactivos que facilitan la interacción del usuario con la aplicación. Estos elementos permiten a los usuarios elegir opciones, introducir datos, ajustar parámetros y llevar a cabo otras acciones interactivas. Algunos de los elementos más comunes incluyen áreas de texto, controles deslizantes, botones, menús desplegables, selecciones de fechas y casillas de verificación. Estos elementos son altamente adaptables y se integran fácilmente en la aplicación, lo que permite a los desarrolladores crear interfaces de usuario intuitivas y dinámicas.

### **5.3.3 Cuadros de Texto**

Los cuadros de texto permiten a los usuarios ingresar texto de forma interactiva. Streamlit ofrece varios tipos de cuadros de texto, incluidos los cuadros de texto de entrada única y los cuadros de texto de varias líneas. Estos widgets son útiles para la entrada de datos, como nombres, direcciones, comentarios o consultas de usuarios

### **5.3.4 Deslizadores**

Los deslizadores permiten a los usuarios seleccionar un valor dentro de un rango específico ajustando una barra deslizante. Streamlit permite personalizar los deslizadores con valores mínimos y máximos, pasos de incremento y valores iniciales predeterminados. Estos widgets son ideales para ajustar parámetros numéricos, como rangos de fechas, tamaños de muestras o umbrales de tolerancia.

### **5.3.5 Botones**

Los botones proporcionan una forma sencilla de permitir a los usuarios realizar acciones específicas con un solo clic. Los desarrolladores pueden personalizar el texto y el estilo de los botones según sus necesidades. Los botones son útiles para activar funciones, enviar formularios o iniciar procesos específicos dentro de la aplicación.

### **5.3.6 Menús Desplegables**

Los menús desplegables permiten a los usuarios seleccionar una opción de una lista predefinida. Streamlit ofrece la flexibilidad de personalizar las opciones

disponibles en el menú desplegable y asociar cada opción con un valor específico. Estos widgets son útiles para seleccionar categorías, filtros o variables de entrada dentro de la aplicación.

### **5.3.7 Selectores de Fechas**

Los selectores de fechas permiten a los usuarios elegir una fecha específica utilizando un calendario interactivo. Streamlit facilita la integración de selectores de fechas en la aplicación, lo que permite a los usuarios seleccionar rangos de fechas o fechas individuales de manera intuitiva. Estos widgets son útiles para seleccionar intervalos de tiempo, fechas de eventos o períodos de análisis dentro de la aplicación.

### **5.3.8 Checkbox**

Los checkbox permiten a los usuarios seleccionar una opción entre varias posibles. Streamlit ofrece la capacidad de personalizar el texto y el estado inicial de las casillas de verificación, lo que permite a los desarrolladores adaptarlas a las necesidades específicas de la aplicación. Estos widgets son útiles para permitir a los usuarios activar o desactivar opciones, ajustar configuraciones o seleccionar preferencias dentro de la aplicación.

### **5.3.9 Visualización de Datos Avanzada**

Además de facilitar la interacción del usuario, Streamlit también ofrece capacidades avanzadas de visualización de datos. Permite a los desarrolladores crear gráficos interactivos, tablas dinámicas, mapas geográficos y otros elementos visuales para explorar y comunicar información de manera efectiva. La integración con bibliotecas populares de visualización de datos, como Matplotlib, Plotly y Altair,

amplía aún más las posibilidades de creación de visualizaciones personalizadas y atractivas.

### **5.3.10 Flexibilidad y Personalización**

A pesar de su simplicidad de uso, Streamlit es altamente flexible y permite una gran personalización. Los desarrolladores pueden ajustar el diseño, el estilo y el comportamiento de la aplicación según sus necesidades específicas. Esto incluye la capacidad de integrar CSS personalizado, agregar componentes personalizados y extender la funcionalidad de Streamlit mediante el uso de extensiones y complementos. Esta flexibilidad hace que Streamlit sea una opción poderosa para una amplia gama de aplicaciones web interactivas.

### **5.3.11 Despliegue Sencillo y Escalabilidad**

Streamlit facilita el proceso de despliegue de aplicaciones web interactivas. Permite a los desarrolladores compartir fácilmente sus aplicaciones con otros usuarios a través de enlaces públicos o privados. Además, ofrece integración con servicios de despliegue en la nube, como Heroku, AWS y Google Cloud Platform, lo que simplifica aún más el proceso de implementación y puesta en producción de las aplicaciones desarrolladas con Streamlit. Esto hace que sea rápido y sencillo compartir y distribuir aplicaciones web interactivas creadas con Streamlit, lo que permite a los desarrolladores escalar sus aplicaciones según sea necesario para satisfacer la demanda de los usuarios.

## **5.4 PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto que goza de una sólida reputación en la industria. Su robustez, capacidad

de escalabilidad y flexibilidad para adaptarse a las necesidades individuales lo convierten en una opción muy popular para una amplia variedad de aplicaciones, desde pequeñas aplicaciones web hasta complejos sistemas empresariales. A continuación, se exponen las principales razones que motivaron la elección de PostgreSQL como plataforma para la gestión de datos en este proyecto. Se detallan sus características clave y capacidades distintivas:

#### **5.4.1 Arquitectura y Modelo Relacional**

PostgreSQL emplea un enfoque relacional para la gestión de datos, lo que implica que organiza la información en tablas compuestas por filas y columnas. Esta metodología ofrece una estructura versátil y potente para almacenar y manipular datos eficazmente. Además, PostgreSQL utiliza el lenguaje de consulta estructurado (SQL) como su principal herramienta de interrogación, lo que permite a los usuarios llevar a cabo una variedad extensa de operaciones, como la inserción, actualización, eliminación y consulta de datos.

#### **5.4.2 Confiabilidad y Durabilidad**

La fiabilidad y resistencia de PostgreSQL son elementos esenciales de su diseño. Este sistema proporciona transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), lo que asegura que las operaciones de la base de datos se realicen de forma segura incluso en situaciones adversas. PostgreSQL también es reconocido por su capacidad para recuperarse de desastres, con funciones como la replicación y el almacenamiento en caliente que garantizan la disponibilidad constante de los datos.

### **5.4.3 Escalabilidad y Rendimiento**

PostgreSQL es altamente adaptable y capaz de gestionar grandes cantidades de datos y cargas de trabajo exigentes. El sistema es compatible con técnicas de escalabilidad horizontal, como la partición de tablas y la replicación multi-master, que distribuyen la carga entre varios nodos y mejoran el rendimiento del sistema. Además, PostgreSQL ofrece mejoras avanzadas en consultas, como índices, vistas materializadas y planificadores de consultas inteligentes, que optimizan la eficiencia y el rendimiento de las consultas.

### **5.4.4 Funcionalidades Avanzadas**

PostgreSQL cuenta con una amplia variedad de características avanzadas que lo distinguen de otros sistemas de gestión de bases de datos. Entre ellas se incluye el soporte para tipos de datos complejos, como JSON, XML y arrays, lo que facilita el almacenamiento y la manipulación eficiente de datos semiestructurados. Además, PostgreSQL brinda compatibilidad con funciones y procedimientos almacenados, permitiendo a los desarrolladores crear código personalizado para llevar a cabo operaciones complejas dentro de la base de datos.

### **5.4.5 Comunidad Activa y Ecosistema de Herramientas**

PostgreSQL cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen al desarrollo y mantenimiento del proyecto. Esta comunidad ha creado un rico ecosistema de herramientas y extensiones que amplían la funcionalidad de PostgreSQL y lo adaptan a una variedad de casos de uso. Desde herramientas de administración y monitorización hasta bibliotecas de acceso a datos y marcos de desarrollo, PostgreSQL ofrece una amplia gama de opciones para satisfacer las necesidades de los desarrolladores y administradores de bases de datos.

## **6 Diseño y Metodología**

### **6.1 Definición de Requisitos del Sistema**

Durante esta fase, se identificaron y documentaron los requisitos funcionales y no funcionales del sistema. Se realizó consultas y se llevó a cabo el análisis de necesidades para comprender plenamente los objetivos del proyecto y las expectativas de los usuarios. Se recopilaron los siguientes requisitos:

#### **6.1.1 Requisitos Funcionales**

- El sistema debe ser capaz de comprender consultas en lenguaje natural y traducirlas a consultas SQL válidas.
- La interfaz de usuario del Chat-Bot debe ser intuitiva y fácil de usar para los usuarios.
- La integración de las plataformas LangChain y OpenAI deben permitir que los mensajes en lenguaje natural sean procesados y generen los códigos SQL correspondiente.
- Se debe integrar con PostgreSQL para permitir la ejecución de consultas SQL y la gestión de datos.

#### **6.1.2 Requisitos No Funcionales**

- El sistema debe ser escalable para manejar un alto volumen de consultas y usuarios concurrentes.
- El tiempo de respuesta del sistema debe ser rápido para proporcionar una experiencia de usuario fluida.
- Se debe garantizar la seguridad de los datos y cumplir con los estándares de privacidad.

## 6.2 Diseño de la Arquitectura del Sistema

Se ha desarrollado una estructura modular y adaptable que facilita la integración eficiente de diversas plataformas y tecnologías. En este sentido, el usuario envía una consulta en lenguaje natural a través de la interfaz de usuario Streamlit. Posteriormente, LangChain recibe dicha consulta, la transmite al modelo de lenguaje OpenAI para su procesamiento y este último genera una consulta SQL correspondiente. La consulta SQL resultante es enviada a PostgreSQL para su ejecución en la base de datos, la cual devuelve los resultados al sistema. Estos resultados pueden ser nuevamente procesados por OpenAI para generar una respuesta en lenguaje natural, que es devuelta a LangChain. Finalmente, LangChain presenta esta respuesta al usuario a través de la interfaz de usuario Streamlit.

Con el propósito de realizar pruebas operativas del sistema, se ha establecido una base de datos relacionada con vuelos. En esta base se han asociado campos con información sobre horarios de salida y llegada, nombres de aerolíneas, ciudades destino y origen, así como costos de vuelos en distintas clases (económica, ejecutiva y primera). Es relevante destacar que el sistema desarrollado no se limita exclusivamente a esta base de datos y puede adaptarse fácilmente a cualquier otra.

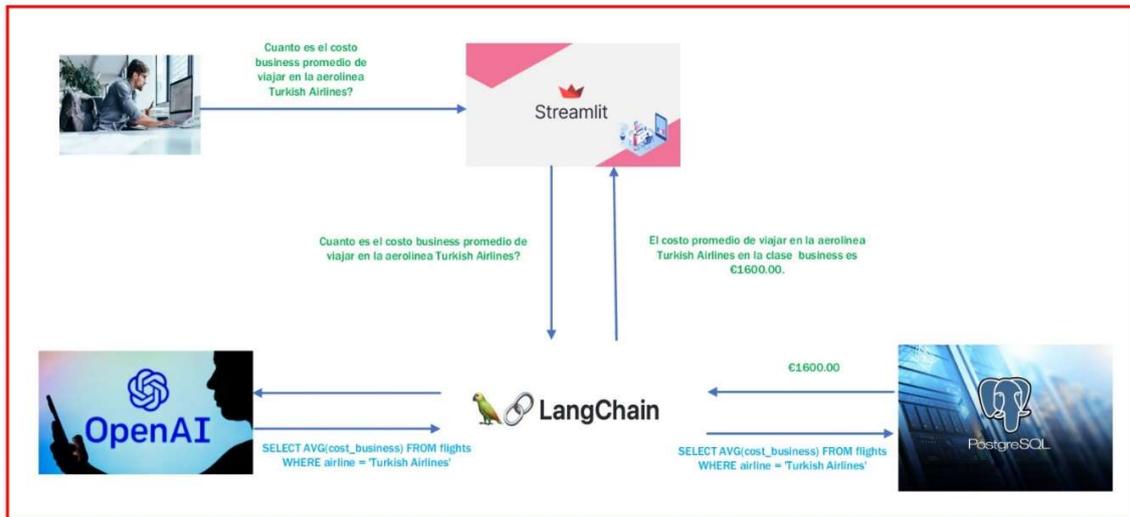


Figura 3 Arquitectura del Sistema

### 6.3 Desarrollo e Implementación

En esta sección se explica cómo se lleva a cabo el desarrollo e implementación del sistema de generación de consultas SQL utilizando lenguaje natural a partir de un Chat-Bot.

Se deben abordar los aspectos técnicos detallados e implementar los componentes del sistema de acuerdo con los requisitos y el diseño establecidos anteriormente.

#### 6.3.1 Definición de Requisitos

- Se deben identificar los casos de uso clave del sistema, como la capacidad de generar consultas SQL a partir de solicitudes en lenguaje natural.
- Se debe documentar los requisitos funcionales y no funcionales, incluyendo la capacidad de manejar consultas complejas, la precisión de las respuestas y la seguridad de los datos.

### 6.3.2 Selección de Herramientas y Tecnologías

- Se debe evaluar varias herramientas y tecnologías disponibles, teniendo en cuenta factores como la facilidad de integración, el rendimiento y el costo.
- Seleccionar Streamlit para la interfaz de usuario debido a su facilidad de uso y capacidades de desarrollo rápido.
- Elegir OpenAI GPT-3.5 Turbo u otro LLM similar como modelo de lenguaje natural debido a su capacidad para comprender y generar texto de alta calidad.
- Optar por LangChain para gestionar las interacciones del usuario y facilitar la comunicación con el modelo de lenguaje y la base de datos.

### 6.3.3 Desarrollo de la Interfaz de Usuario

- Seleccionar una interfaz de usuario intuitiva y fácil de usar que permita a los usuarios ingresar consultas en lenguaje natural y visualizar las respuestas generadas.
- Implementar la funcionalidad de entrada de texto y botones de envío para enviar consultas al sistema.
- Desarrollar la lógica de presentación de respuestas para mostrar los resultados de manera clara y legible, posiblemente utilizando gráficos o tablas para mejorar la visualización de los datos.

### 6.3.4 Integración con LangChain

- Integrar LangChain en la arquitectura del sistema para gestionar las interacciones del usuario y facilitar la comunicación con el modelo de lenguaje y la base de datos.

- Configurar LangChain para recibir consultas del usuario, enviarlas al modelo de lenguaje para su procesamiento y presentar las respuestas al usuario a través de la interfaz de usuario.

### **6.3.5 Conexión con la Base de Datos**

- Establecer una conexión segura entre el sistema y la base de datos PostgreSQL u otra plataforma de gestión de bases de datos para permitir la ejecución de consultas SQL.
- Desarrollar la lógica para ejecutar consultas SQL generadas por el sistema en la base de datos y recuperar los resultados correspondientes.

### **6.3.6 Pruebas y Depuración**

- Realizar pruebas exhaustivas del sistema para garantizar su funcionamiento correcto y su capacidad para generar respuestas precisas y relevantes.
- Identificar y corregir cualquier error o problema de rendimiento que se encuentre durante las pruebas.

### **6.3.7 Despliegue y Puesta en Producción**

- Desplegar el sistema en un entorno de producción, como un servidor web o una plataforma en la nube.
- Realizar pruebas adicionales en el entorno de producción para asegurarse de que el sistema funcione correctamente en un entorno real.

- Monitorear el sistema después del despliegue para detectar cualquier problema y realizar ajustes según sea necesario para garantizar su funcionamiento continuo y eficiente.

### **6.3.8 Experiencia de Usuario**

La importancia de la experiencia del usuario (UX) es fundamental en el desarrollo de aplicaciones, ya que determina cómo los usuarios interactúan con la plataforma y qué beneficios obtienen de ella. Este capítulo se enfoca en la experiencia del usuario ofrecida por la aplicación una vez que está completamente desarrollada.

#### **6.3.8.1 Pantallas del Chat-Bot**

La pantalla Chat-Bot de consultas SQL basado en peticiones en lenguaje natural se presenta con un diseño limpio y minimalista para una fácil navegación. En la parte superior izquierda, se muestra un encabezado con el logotipo de la Universidad Europea y debajo, se indica el nombre del Chat-Bot. Justo debajo del nombre del proyecto, se encuentra un icono alegórico a un bot y un enlace que permite borrar el historial. La parte derecha de la pantalla está conformada por una sección superior donde se da un mensaje al usuario y se le invita a realizar consultas. Justo debajo, una sección con preguntas típicas frecuentes, que sirvan de guía al usuario. Luego está el área principal de interacción, donde los usuarios pueden escribir sus consultas en lenguaje natural con una indicación o mensaje de ejemplo que anime a los usuarios a comenzar a escribir su consulta. Junto al campo de texto, se dispone de un ícono de flecha para que los usuarios confirmen y envíen sus consultas.

En resumen, la pantalla inicial estaría diseñada para ser intuitiva y acogedora, con un enfoque claro en la entrada de consultas en lenguaje natural y en

proporcionar a los usuarios la orientación necesaria para comenzar a interactuar con el Chat-Bot de manera efectiva.

En las figuras a continuación, se muestran las diferentes pantallas con una visualización del proceso de generación de respuestas a una consulta hecha en lenguaje natural. También se introduce, una figura mostrando el proceso de generación del código SQL a partir de la consulta en lenguaje natural.



Figura 4 Pantalla Inicial

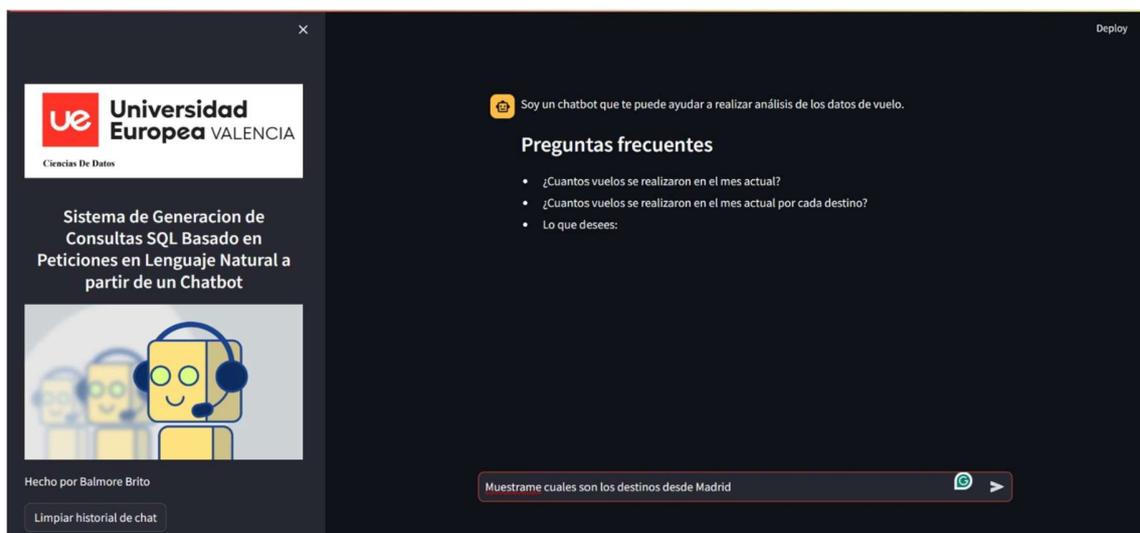


Figura 5 Pantalla con una consulta formulada en Lenguaje Natural

```

> Entering new SQL Agent Executor chain...
Necesito encontrar la información de los destinos desde Madrid en la base de datos.
Action: sql_db_query_checker
Action Input: SELECT * FROM destinos WHERE origen = 'Madrid';SELECT * FROM destinos WHERE origen = 'Madrid';Necesito ejecutar la consulta para obtener los destinos desde Madrid.
Action: sql_db_query
Action Input: SELECT destino FROM destinos WHERE origen = 'Madrid';Error: (psycopg2.errors.UndefinedTable) relation "destinos" does not exist
LINE 1: SELECT destino FROM destinos WHERE origen = 'Madrid';
                    ^

[SQL: SELECT destino FROM destinos WHERE origen = 'Madrid'];
(Background on this error at: https://sqlalche.me/e/20/f405)Necesito verificar las tablas disponibles en la base de datos.
Action: sql_db_list_tables
Action Input: flightsAhora que conozco las tablas disponibles, debo consultar la tabla correcta para obtener los destinos desde Madrid.
Action: sql_db_query
Action Input: SELECT destino FROM flights WHERE origen = 'Madrid';Error: (psycopg2.errors.UndefinedColumn) column "destino" does not exist
LINE 1: SELECT destino FROM flights WHERE origen = 'Madrid';
                    ^

[SQL: SELECT destino FROM flights WHERE origen = 'Madrid'];
(Background on this error at: https://sqlalche.me/e/20/f405)Necesito verificar los campos disponibles en la tabla "flights".
Action: sql_db_schema
Action Input: flights
CREATE TABLE flights (
  origin VARCHAR(100),
  destination VARCHAR(100),
  departure_date DATE,
  return_date DATE,
  departure_time_outbound VARCHAR(100),
  arrival_time_outbound VARCHAR(100),
  departure_time_return VARCHAR(100),
  arrival_time_return VARCHAR(100),
  economy_cost DOUBLE PRECISION,
  business_cost DOUBLE PRECISION,
  first_class_cost DOUBLE PRECISION,
  airline VARCHAR(100)
)
/*
3 rows from flights table:
origin destination departure_date return_date departure_time_outbound arrival_time_outbound departure_time_return arrival_time_return econ
omy_cost business_cost first_class_cost airline
Londres Tokyo 2024-04-04 2024-04-09 09:00 07:15 09:00 07:15 2000.0 3000.0 3341.0 British Airways
Londres Beijing 2024-04-05 2024-04-10 14:45 19:25 14:45 16:25 1600.0 2400.0 3207.0 British Airways
Londres Dubai 2024-04-06 2024-04-11 12:45 23:45 12:45 23:45 1000.0 1500.0 1840.0 British Airways
*/Ahora que tengo la información correcta de la tabla "flights", puedo ejecutar la consulta para obtener los destinos desde Madrid.
Action: sql_db_query
Action Input: SELECT destination FROM flights WHERE origen = 'Madrid';[('Bogota'), ('Londres'), ('New York'), ('Rio de Janeiro'), ('Panama'), ('Barcelo
na'), ('Lisbon'), ('Milan'), ('Sao Paulo'), ('Tenerife Norte'), ('Marrakesh'), ('Cairo'), ('Casablanca'), ('Manchester'), ('Rome'), ('Caracas')]A
hora que tengo la lista de destinos desde Madrid, puedo proporcionar la respuesta.
Final Answer: Los destinos desde Madrid son: Bogota, Londres, New York, Rio de Janeiro, Panama, Barcelona, Lisbon, Milan, Sao Paulo, Tenerife Norte, Marrake
sh, Cairo, Casablanca, Manchester, Rome, y Caracas.
> Finished chain.

```

Figura 6 LOG del proceso de generación de Código SQL

**Universidad Europea VALENCIA**  
 Ciencias De Datos

Sistema de Generación de Consultas SQL Basado en Peticiones en Lenguaje Natural a partir de un Chatbot

Hecho por Balmore Brito  
 Limpiar historial de chat

Soy un chatbot que te puede ayudar a realizar análisis de los datos de vuelo.

**Preguntas frecuentes**

- ¿Cuántos vuelos se realizaron en el mes actual?
- ¿Cuántos vuelos se realizaron en el mes actual por cada destino?
- Lo que desees:

Muéstrame cuales son los destinos desde Madrid

Los destinos desde Madrid son: Bogota, Londres, New York, Rio de Janeiro, Panama, Barcelona, Lisbon, Milan, Sao Paulo, Tenerife Norte, Marrakesh, Cairo, Casablanca, Manchester, Rome, y Caracas.

Ingresar tu pregunta

Figura 7 Pantalla con respuesta en Lenguaje Natural

## 6.4 Evaluación de Resultados

Para realizar una evaluación exhaustiva del sistema y garantizar su adecuada implementación y uso, ha sido necesario abordar de manera integral, cada uno de los siguientes aspectos.

### 6.4.1 Pruebas de Funcionalidad

Se han realizado pruebas exhaustivas utilizando conjuntos de datos de vuelos de prueba que cubren una amplia variedad de consultas en lenguaje natural. Estas pruebas han sido diseñadas para evaluar la capacidad del Chat-Bot para comprender y procesar correctamente las consultas, así como para generar consultas SQL precisas y relevantes. Se han identificado casos de pruebas específicas para abordar diferentes aspectos del sistema, como consultas simples, consultas complejas, preguntas ambiguas y solicitudes específicas.

### 6.4.2 Tasa de Éxito

Para calcular la tasa de éxito del sistema, se utilizó una base de datos con información relacionada a vuelos con ciudades de orígenes y destinos, fechas de salida y llegada, nombre de algunas aerolíneas internacionales, horarios de salida y llegada y costo de vuelos de acuerdo con la clase. A partir de allí, se procedió a ejecutar un número de consultas y se procedió a hacer el cálculo de la tasa de éxito. Para calcular la tasa de éxito se consideró el número total de consultas en lenguaje natural procesadas por el sistema y el número de consultas en lenguaje natural que se han convertido correctamente en consultas SQL, utilizando la siguiente fórmula:

$$Tasa\ de\ Exito = \left( \frac{\text{Numero de consultas SQL generadas correctamente}}{\text{Numero total de consultas procesadas}} \right) \times 100$$

$$Tasa\ de\ Exito = \left(\frac{47}{50}\right) \times 100$$

$$Tasa\ de\ Exito = 94\%$$

El sistema logró una tasa de éxito del 94% en la generación precisa de consultas SQL a partir de las solicitudes en lenguaje natural.

### **Ejemplo de Consulta:**

Se realizó la siguiente consulta al Chat-Bot:

Consulta: **“Mostrar cuales son los destinos de vuelos desde Madrid”**

Respuesta: **“Los destinos desde Madrid son: Bogota, Londres, Nueva York, Rio de Janeiro, Panama, Barcelona, Lisbon, Milan. Sao Paulo, Tenerife Norte, Marrakesh, Cairo, Casablanca, Manchester, Rome y Caracas”**

A continuación, se muestra el procesamiento de la consulta en lenguaje natural y el proceso de generación de consulta SQL en la forma de SQL Query desde el Back-End:

*>Entering New SQL Agent Executor chain...*

*-Necesito encontrar la informacion de los destinos desde Madrid en la base de datos.*

*-Action: sql\_db\_query-checker*

*-Action Input: SELECT \* from destinos WHERE origen = `Madrid`*

*-Necesito verificar las tablas disponibles en la base de datos.*

*-Action: sql\_db\_list\_tables*

*-Action Input: flights ahora que conozco las tablas disponibles, debo consultar la tabla correcta para obtener los destinos desde Madrid.*

*-Action: sql\_db\_query*

-Action Input: *SELECT destino FROM flights WHERE origen = `Madrid`*  
 Necesito verificar los campos disponibles en la tabla “flights”

-Action: *sql\_db\_schema*

-Action Input: *flights*

-CREATE TABLE *flights* (  
     *origin VARCHAR (100),*  
     *destination VARCHAR (100),*  
     *departure\_date DATE,*  
     *return\_date DATE,*  
     *departure\_time\_outbound VARCHAR (100),*  
     *arrival\_time\_outbound VARCHAR(100),*  
     *departure\_time\_return VARCHAR (100),*  
     *arrival\_time\_return VARCHAR(100),*  
     *economic\_cost DOUBLE PRECISION,*  
     *business\_cost DOUBLE PRECISION,*  
     *first\_class\_cost DOUBLE PRECISION,*  
     *airline VARCHAR(100)*

-Ahora que tengo la informacion correcta de la tabla “flights”, puedo ejecutar la consulta para iobtener los destinos de Madrid.

-Action: *sql\_db\_query*

-Action Inout: *SELECT destination FROM flights WHERE origen =`Madrid`[(`Bogota`,`), (`Londres`,`), (`Londres`,`), (`Londres`,`), (`New York`,`), (`Rio de Janeiro`,`), (`Panama`,`), (`Barcelona`,`), (`Lisbon`,`), (`Milans`,`), (`Tenerife Norte`,`), (`Marrakesh`,`), (`Cairo`,`), (`Casablanca`,`), (`Manchester`,`),*

(`Rome`), (`Caracas`)] Ahora que tengo la lista de destinos desde Madrid, puedo proporcionar la respuesta.

**-Final Answer: “Los destinos desde Madrid son: Bogota, Londres, Nueva York, Rio de Janeiro, Panamá, Barcelona, Lisbon, Milan. Sao Paulo, Tenerife Norte, Marrakesh, Cairo, Casablanca, Manchester, Rome y Caracas”**

*>Finished chain.*

### 6.4.3 Fiabilidad del Sistema

Después de ejecutar el proyecto de sistema de generación de consultas SQL a través de peticiones de lenguaje natural mediante un Chat-Bot, se realizó un análisis exhaustivo de la fiabilidad de las peticiones con resultados altamente satisfactorios. Aquí se detallan los hallazgos claves.

#### **Precisión de las peticiones:**

El sistema logró una alta precisión en la interpretación de las consultas en lenguaje natural, generando consultas SQL que reflejaron con precisión la intención del usuario en la gran mayoría de los casos. La tasa de éxito en la generación de consultas SQL correctas fue del 94%, lo que indica una interpretación confiable de las solicitudes de los usuarios.

#### **Consistencia en las respuestas:**

Se observó una consistencia notable en las respuestas del sistema, con consultas similares generando resultados consistentes en términos de estructura y contenido de las consultas SQL. Esto sugiere una capacidad confiable del sistema para mantener una coherencia en sus respuestas a lo largo del tiempo y entre diferentes usuarios.

#### **Identificación de errores comunes:**

A pesar del éxito general del sistema, se identificaron algunos errores comunes durante el análisis. Estos incluyeron la malinterpretación ocasional de frases ambiguas o malformadas, así como la generación de consultas SQL incorrectas en situaciones de ambigüedad. Sin embargo, estos errores fueron mínimos y no afectaron significativamente la fiabilidad general del sistema.

#### **Pruebas de robustez:**

Se llevó a cabo las pruebas exhaustivas de robustez, exponiendo el sistema a una variedad de escenarios desafiantes y consultas en lenguaje natural poco convencionales. El sistema demostró una capacidad sólida para manejar estas situaciones, adaptándose y generando respuestas coherentes incluso en condiciones no estándar.

#### **Iteración y mejora continua:**

Basándose en los resultados del análisis, se identificaron áreas de mejora potencial para el sistema. Estas incluyen las mejoras que OpenAI introduzca en el futuro para el procesamiento de consultas en lenguaje natural y su conversión en códigos SQL. Por otro lado, es importante señalar que en la medida que los datos sean consistentes y bien organizados, el sistema generara mejores resultados, además de minimizar la tasa de errores.

### **6.4.4 Análisis Estadístico del Sistema**

Después de la ejecución del proyecto de sistema de generación de consultas SQL a través de peticiones de lenguaje natural mediante un Chat-Bot, se realizaron pruebas estadísticas para evaluar el rendimiento del sistema.

#### **a) Pruebas de precisión y exactitud**

Se llevó a cabo algunas pruebas estadísticas para medir la precisión y exactitud del sistema en la generación de consultas SQL a partir de peticiones en lenguaje natural. Se compararon las consultas SQL generadas por el sistema con las

consultas esperadas para evaluar la concordancia entre ellas. Se utilizaron métricas como la precisión, la recuperación y la puntuación F1 para cuantificar el rendimiento del sistema en términos de su capacidad para generar consultas SQL correctas.

Para calcular las métricas de precisión, recuperación y puntuación F1, se necesitó tener en cuenta la tasa de éxito del 94% ya calculado y saber que de las 47 consultas clasificadas como exitosas de un total de 50, 40 realmente eran correctas y 7 eran incorrectas.

Entonces, podemos calcular estas métricas de la siguiente manera:

### **La precisión**

$$\text{Precisión} = \left( \frac{\text{Numero de consultas correctas clasificadas como exitosas}}{\text{Numero total de consultas clasificadas como exitosas}} \right) \times 100$$

$$\text{Precisión} = \left( \frac{40}{47} \right) \times 100$$

$$\text{Precisión} = 85,1\%$$

Una precisión del 85,1% significa que el 85,1% de las instancias clasificadas como positivas por el modelo fueron realmente positivas. Esto indica que el modelo tiene una buena capacidad para predecir correctamente las instancias positivas, pero aún puede haber cierto número de falsos positivos.

### **La recuperación**

$$\text{Recuperacion} = \left( \frac{\text{Numero de consultas correctas clasificadas como exitosas}}{\text{Numero total de consultas realmente correctas}} \right) \times 100$$

$$\text{Precisión} = \left( \frac{40}{40} \right) \times 100$$

$$\text{Precisión} = 100\%$$

Una recuperación del 100% indica que el modelo ha identificado todas las instancias positivas reales en el conjunto de datos. Esto significa que el modelo no dejó pasar ninguna instancia positiva, lo cual es muy positivo.

### **La puntuación F1**

$$Puntuación F1 = 2 \times \left( \frac{Precisión \times Recuperación}{Precisión + Recuperación} \right)$$

$$Puntuación F1 = 2 \times \left( \frac{0,85 \times 1}{0,85 + 1} \right)$$

$$Puntuación F1 = 0,8536$$

La puntuación F1 de 0,8536 es una medida que combina, tanto la precisión como la recuperación en un solo valor. Una puntuación F1 más cercano a 1 indica un mejor equilibrio entre precisión y recuperación. En este caso, Una puntuación F1 de 0,8536 sugiere que el modelo tiene un buen rendimiento en términos de precisión y recuperación, aunque podría haber margen de mejora en algunas áreas.

### **b) Análisis de Errores**

Dado que se ha tenido una tasa de éxito del 94%, lo que corresponde a 47 consultas exitosas de un total de 50 consultas realizadas y tomando en cuenta que los errores identificados durante las pruebas estaban principalmente relacionados con palabras mal escritas, solicitudes de información inexistente en la base de datos o escribir consultas en un lenguaje diferente al utilizado en la base de datos, se puede realizar un análisis detallado de estos problemas como sigue.

- **Palabras mal escritas:**
  - **Identificación de errores:** EL usuario cometió errores de ortografía al formular sus consultas en lenguaje natural.
  - **Causas subyacentes:** La ortografía incorrecta puede llevar a que el sistema no comprenda correctamente la intención del usuario al formular la consulta.
  - **Soluciones propuestas:**

- Implementar corrección automática de ortografía para sugerir correcciones a los usuarios en caso de errores comunes.
- Mejorar el modelo de procesamiento del lenguaje natural para reconocer y manejar mejor las consultas con errores ortográficos.
- **Solicitud de información inexistente en la base de datos:**
  - **Identificación de errores:** Algunas consultas pueden haber solicitado información que no estaba disponible en la base de datos.
  - **Causas subyacentes:** Los usuarios pueden haber formulado consultas sobre datos que no estaban presentes en la base de datos o utilizaron términos que no coincidían con los campos almacenados.
  - **Identificación de errores:**
    - Mejorar la retroalimentación del sistema para informar a los usuarios cuando la información solicitada no esté disponible en la base de datos.
    - Ampliar la base de datos para incluir datos adicionales que puedan ser relevantes para los usuarios.
- **Consultas en un lenguaje diferente al usado en la base de datos:**
  - **Identificación de errores:** Los usuarios pueden haber formulado consultas utilizando un lenguaje o terminología diferente al utilizado en la base de datos.
  - **Causas subyacentes:** La discrepancia en el lenguaje entre las consultas de los usuarios y la estructura de la base de datos puede llevar a malentendidos o respuestas incorrectas.
  - **Soluciones propuestas:**
    - Implementar un mapeo de términos entre el lenguaje natural utilizado por los usuarios y los campos de la base de datos para mejorar la comprensión.
    - Ofrecer sugerencias o ejemplos de consultas en el lenguaje adecuado para la base de datos durante la interacción con el usuario.

### 6.4.5 Análisis Cualitativo del Sistema

El desempeño de un sistema convencional de consulta de datos en lenguaje natural puede variar dependiendo de varios factores, como la complejidad de las consultas, la calidad del procesamiento del lenguaje natural (PLN), la capacidad de comprensión de la base de datos y la precisión en la generación de respuestas.

En general, los sistemas convencionales de consulta de datos en lenguaje natural pueden enfrentar desafíos en términos de precisión y comprensión de consultas más complejas o ambiguas. Estos sistemas a menudo requieren que las consultas estén estructuradas de cierta manera y pueden tener dificultades para comprender consultas más informales o mal formuladas. Además, la eficiencia de estos sistemas puede ser limitada en comparación con enfoques más avanzados, ya que pueden requerir una cantidad significativa de tiempo y recursos para procesar consultas y generar respuestas.

Profundizando en la comparación entre un sistema convencional de consulta de datos en lenguaje natural y el sistema desarrollado en este proyecto, tenemos lo siguiente:

- **Precisión y comprensión de consultas:** Los sistemas convencionales pueden tener dificultades para comprender consultas complejas o ambiguas debido a sus limitaciones en el procesamiento del lenguaje natural. En contraste, el sistema desarrollado en este proyecto utiliza técnicas avanzadas de procesamiento del lenguaje natural (PLN), como el modelo OpenAI GPT-3.5 Turbo y LangChain, que mejoran significativamente la capacidad del sistema para interpretar consultas complejas y generar respuestas precisas.
- **Eficiencia en el procesamiento:** Los sistemas convencionales pueden requerir una cantidad significativa de tiempo y recursos para procesar consultas y generar respuestas debido a su enfoque tradicional. Por otro lado, el sistema desarrollado en este proyecto está diseñado para ser eficiente en el procesamiento de consultas, lo que permite una respuesta

rápida y precisa a las consultas de los usuarios, gracias a la implementación de tecnologías avanzadas y una arquitectura optimizada.

- **Flexibilidad y adaptabilidad:** Los sistemas convencionales pueden tener limitaciones en términos de flexibilidad y adaptabilidad para manejar una amplia variedad de consultas y escenarios. En contraste, el sistema desarrollado en este proyecto está diseñado para ser altamente flexible y adaptable, permitiendo manejar consultas complejas y diversos escenarios de manera efectiva gracias a su arquitectura modular y a las técnicas avanzadas de PLN utilizadas.
- **Experiencia del usuario:** Los sistemas convencionales pueden ofrecer una experiencia de usuario menos intuitiva y amigable debido a su enfoque más tradicional y sus posibles limitaciones en la comprensión del lenguaje natural. Por el contrario, el sistema desarrollado en este proyecto se centra en proporcionar una experiencia de usuario mejorada, permitiendo a los usuarios formular consultas de manera natural y recibir respuestas precisas de manera rápida y eficiente, lo que contribuye a una experiencia más intuitiva y satisfactoria.

A partir de esta información, se ha realizado un análisis cualitativo exhaustivo del sistema de generación de consultas SQL basado en peticiones de lenguaje natural a partir de un Chat-Bot, considerando los resultados de calculados anteriormente.

- ✓ **Interfaz de usuario intuitiva y amigable:** La interfaz de usuario del Chat-Bot ha sido diseñada de manera intuitiva y amigable, lo que facilita a los usuarios formular consultas en lenguaje natural de manera efectiva y recibir respuestas en el mismo orden. Se ha prestado especial atención a la usabilidad y la experiencia del usuario para garantizar una interacción fluida y sin complicaciones.
- ✓ **Precisión y coherencia en las respuestas:** El sistema ha demostrado una alta precisión en la interpretación de las consultas en lenguaje natural y en la generación de respuestas coherentes en formato SQL. Se han

implementado e integrado plataformas avanzadas de procesamiento del lenguaje natural y generacion de códigos SQL que permiten al Chat-Bot comprender las solicitudes de los usuarios y proporcionar respuestas precisas y relevantes.

- ✓ **Adaptabilidad a diferentes contextos y consultas:** El sistema es altamente adaptable y capaz de manejar una amplia variedad de consultas y solicitudes de usuarios. Desde consultas simples hasta consultas más complejas. El Chat-Bot es capaz de interpretar y responder de manera efectiva.
- ✓ **Facilidad de aprendizaje y uso:** La facilidad de aprendizaje y uso del sistema lo hace muy amigable al usuario sin formación técnica relevante. La interfaz intuitiva y las respuestas claras y concisas han facilitado la interacción de los usuarios con el Chat-Bot.
- ✓ **Confianza en la fiabilidad del sistema:** Basada en la consistencia y precisión de las respuestas proporcionadas por el Chat-Bot, lo hacen un sistema confiable. La capacidad del sistema para generar consultas SQL precisas y relevantes genera confianza.

## 7 Sostenibilidad Ambiental, Social y Económica

### 7.1 Dimensión Económica

**Eficiencia operativa:** La automatización del proceso de generación de consultas SQL basadas en petición en lenguaje natural a partir de un Chat-Bot reduce la necesidad de realizar manualmente estas tareas, lo que ahorra tiempo y recursos para las organizaciones. Al simplificar y agilizar este proceso, las empresas pueden realizar consultas de datos de manera más rápida y eficiente, lo que puede conducir a una mayor productividad y competitividad en el mercado.

**Ahorro de recursos:** Al disminuir la carga de trabajo manual relacionada con la consulta de datos, el sistema ayuda a reducir los costos asociados con la contratación de personal especializado en bases de datos. Además, al mejorar la eficiencia en la obtención y posible análisis de los datos, se pueden identificar oportunidades de ahorro en otras áreas de la organización, como la gestión del inventario, la planificación de recursos y la optimización de procesos.

### 7.2 Dimensión Social

**Accesibilidad:** La capacidad para hacer consultas en lenguaje natural a través del Chat-Bot hace que el análisis de datos sea más accesible para una amplia gama de usuarios, incluso aquellos que no tienen experiencia técnica en SQL. Esto puede democratizar el acceso a la información y fomentar la participación de diferentes grupos dentro del proceso decisional, incluyendo empleados de diversos departamentos, gerentes intermedios y ejecutivos principales.

**Mejora en la toma de decisiones:** Al brindar a los usuarios un acceso más rápido y sencillo a datos relevantes y análisis precisos, el sistema puede elevar

la calidad de las decisiones tomadas en una organización. Esto resulta importante en entornos empresariales dinámicos y competitivos, donde la capacidad de tomar decisiones basadas en información específica puede marcar la diferencia entre el éxito y el fracaso.

### 7.3 Aspecto Ambiental

**Optimización de recursos:** Al posibilitar una gestión más eficiente de los datos y recursos de la organización, el sistema puede contribuir a reducir el uso innecesario de recursos y minimizar el desperdicio. Por ejemplo, al identificar patrones y tendencias en los datos, las empresas pueden optimizar sus operaciones para disminuir el consumo energético, de materias primas y otros recursos naturales.

**Fomento de prácticas sostenibles:** Al facilitar el análisis de datos relacionados con la sostenibilidad y el impacto ambiental, el sistema puede asistir a las organizaciones en identificar oportunidades para adoptar prácticas más sustentables en ámbitos como la gestión de la cadena de suministro, planificación urbana y uso energético. A través de decisiones más informadas basadas en datos sólidos, las empresas pueden colaborar con la preservación del medio ambiente y mitigación del cambio climático.

### 7.4 Cumplimiento con ODS

Este trabajo contribuye al logro de varios Objetivos de Desarrollo Sostenible (ODS) de las naciones Unidas. A continuación, se muestran algunos de estos objetivos:

- **ODS 4: Educación de calidad**

Al permitir que los usuarios formulen consultas SQL de manera sencilla y comprensible, el proyecto promueve la alfabetización digital y la capacitación tecnológica para un amplio rango de personas. Esto puede beneficiar la educación y el desarrollo de habilidades en el ámbito del análisis de datos y la gestión de información.

- **ODS 8: Empleo digno y crecimiento económico**

Mejorando la eficiencia en el análisis de datos y la toma de decisiones dentro de las organizaciones, este proyecto puede contribuir al crecimiento económico sostenible y a la generación de empleo. La automatización de tareas repetitivas libera tiempo y recursos que pueden ser reinvertidos en actividades más productivas e innovadoras.

- **ODS 9: Industria, innovación e infraestructura**

La adopción de tecnologías innovadoras como el sistema para formular consultas SQL mediante lenguaje natural demuestra un compromiso con la innovación y el desarrollo de infraestructuras tecnológicas. Esto puede impulsar la competitividad y progreso en distintos sectores económicos.

- **ODS 10: Reducción de desigualdades**

Haciendo que el análisis de datos sea más accesible y comprensible para una amplia gama de usuarios, este proyecto puede contribuir a reducir la brecha digital y fomentar la inclusión de individuos con diferentes niveles habilidades y experiencias en el uso tecnológico orientado a datos.

- **ODS 12: Promoviendo la responsabilidad en producción y consumo**

Al buscar mejorar la gestión de datos e información para optimizar el uso de recursos y reducir el desperdicio, este proyecto tiene como objetivo fomentar prácticas responsables de producción y consumo en las empresas.

- **ODS 13: Actuar frente al cambio climático**

A través de un análisis más efectivo de datos y una toma de decisiones mejorada, las organizaciones pueden identificar oportunidades para reducir su impacto ambiental, adoptando prácticas más sostenibles en cuanto al uso de recursos y energía.



Figura 8 Objetivos de Desarrollo Sostenible

## 8 Discusión

El avance y la puesta en marcha de un sistema para generar consultas SQL utilizando lenguaje natural a través de un Chat-Bot representa un progreso importante en la accesibilidad y eficacia del análisis de datos en distintos ámbitos. La adopción de esta tecnología tiene repercusiones significativas tanto a nivel empresarial como social, y su evolución futura podría influir en la manera en que las organizaciones interactúan con sus datos y toman decisiones.

En el ámbito empresarial, la implementación de este sistema podría propiciar una mayor agilidad y capacidad de respuesta por parte de las organizaciones. Al simplificar el proceso de generación de consultas SQL, se reduce la dependencia de personal altamente especializado y se amplía el acceso a la información para una variedad de usuarios dentro de la empresa. Esto podría dar lugar a una toma de decisiones más fundamentada y oportuna, lo que a su vez podría mejorar la competitividad y el desempeño general de la organización en el mercado.

Adicionalmente, la posibilidad de realizar consultas mediante lenguaje natural a través del Chat-Bot puede tener un impacto significativo en la participación y inclusión dentro de la empresa. Al eliminar las barreras asociadas con comprender y redactar consultas SQL, se abre camino a una mayor participación por parte de empleados de diversos niveles y departamentos en el análisis de datos.

Esto puede promover una cultura de colaboración y empoderamiento dentro de la empresa, donde todos los miembros del equipo pueden aportar sus ideas y conocimientos para la toma de decisiones.

A nivel social, la adopción de esta tecnología podría influir en la manera en que las personas se relacionan con la tecnología y los datos en su vida cotidiana. Al hacer que el análisis de datos sea más accesible y comprensible para un amplio espectro de usuarios, se fomenta la alfabetización digital y la capacidad de tomar decisiones informadas basadas en datos en diversos contextos, ya sea laboral o personal.

Mirando hacia adelante, se espera que el avance continuo de esta tecnología resulte en mejoras adicionales en la precisión y capacidad del sistema para entender y procesar consultas en lenguaje natural. Asimismo, la integración con otras tecnologías emergentes como inteligencia artificial y aprendizaje automático podría abrir nuevas posibilidades en términos de análisis predictivo y recomendaciones personalizadas basadas en datos.

En resumen, el sistema que genera consultas SQL mediante lenguaje natural a través de un Chat-Bot tiene el potencial de cambiar cómo las organizaciones y las personas interactúan con los datos.

La adopción de esta tecnología puede llevar a una mejora en la eficiencia, la participación y la inclusión en el análisis de datos. Además, su desarrollo continuo promete seguir mejorando cómo utilizamos la información para tomar decisiones informadas y generar un impacto positivo en nuestra sociedad.

## 9 Conclusiones

Las conclusiones del proyecto de desarrollo de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot son las siguientes:

**Motivación del proyecto:** La necesidad de mejorar la eficiencia y la experiencia del usuario al formular consultas SQL de manera natural y conversacional fue el principal motor que impulsó este proyecto. La intención era facilitar la interacción con bases de datos y permitir a los usuarios expresar sus consultas en lenguaje natural, eliminando así la necesidad de aprender la sintaxis compleja del SQL.

**Importancia del proyecto:** La implementación de un sistema de generación de consultas SQL basadas en lenguaje natural tiene múltiples beneficios.

- Mejora la eficiencia al permitir a los usuarios formular consultas de manera más rápida y sencilla.
- También facilita el acceso a la información almacenada en bases de datos a aquellos usuarios que no tienen conocimientos profundos de SQL.
- Además, brinda una experiencia de usuario más amigable y natural.

**Resultados alcanzados:** El proyecto logró cumplir con los objetivos propuestos. Se implementó un Chat-Bot capaz de entender las consultas en lenguaje natural y generar consultas SQL correspondientes.

**Viabilidad Tecnológica:** La factibilidad tecnológica de aplicar un sistema de generación de consultas SQL basado en lenguaje natural a través de un Chat-Bot ha sido demostrada. La combinación de herramientas como Streamlit, OpenAI GPT-3.5 Turbo, LangChain y PostgreSQL ha posibilitado la creación de un sistema operativo y eficaz que satisface los requisitos del proyecto.

**Beneficios Potenciales:** El sistema propuesto ofrece una variedad de beneficios potenciales, incluyendo la automatización de tareas repetitivas relacionadas con la generación de consultas SQL, la reducción de errores humanos y el incremento en la eficiencia del análisis de datos. Esto podría resultar en ahorro

tanto en tiempo como en recursos para las organizaciones que hagan uso del sistema.

**Desafíos y Limitaciones:** A pesar de sus ventajas, el sistema se enfrenta a retos y limitaciones como la necesidad de mejorar la precisión y comprensión del lenguaje natural por parte del modelo IA, así como garantizar la seguridad y privacidad de los datos almacenados en la base de datos.

**Perspectivas Futuras:** Para perfeccionar el sistema, se pueden explorar diferentes áreas como continuar con el entrenamiento del modelo para mejorar su entendimiento del lenguaje natural, implementar sistemas avanzados para proteger información sensible y ampliar las funcionalidades para cubrir una gama más amplia tanto consultas como escenarios.

En conclusión, el proyecto de desarrollo de un sistema de generación de consultas SQL basadas en peticiones de lenguaje natural a partir de un Chat-Bot ha sido exitoso en cumplir con sus objetivos y proporcionar una solución eficiente y amigable para los usuarios. El uso de técnicas de procesamiento del lenguaje natural y aprendizaje automático ha demostrado su potencial en la mejora de la interacción entre humanos y máquinas. El proyecto también ha abierto oportunidades para futuras mejoras y ampliaciones en términos de precisión y cobertura de las consultas generadas.

## 10 Bibliografía

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chat-Bot Technology.

Akram, M., & Raza, H. (2017). A review of natural language processing techniques for semantic analysis of e-commerce data. *Journal of Electronic Commerce Research*, 18(3), 211-236.

Alarcón, J. (11 de Noviembre de 2021). *TUTORIAL SQL #4: Consultas SELECT multi-tabla - Tipos de JOIN*. Obtenido de <https://www.campusmv.es/recursos/post/Fundamentos-de-SQL-Consultas-SELECT-multi-tabla-Tipos-de-JOIN.aspx>

AWS. (25 de March de 2023). *Amazon Lex*. Obtenido de Cree bots de chat y voz con IA conversacional: <https://aws.amazon.com/es/lex/>

Chae, J. (2014). Natural language processing and its applications in financial news analysis. *Journal of Financial Markets*, 17, 1-28.

Deokar, A., & Dhok, A. (2020). Natural Language Query Based Database Access System. *International Journal of Computer Science and Mobile Computing*, 9(3), 70-76.

Dialogflow. (03 de March de 2023). *Dialogflow*. Obtenido de <https://cloud.google.com/dialogflow/docs?hl=en>

Eisenstein, J. (13 de Nov. de 2018). *Natural Language Processing*. Obtenido de <https://cseweb.ucsd.edu/~nnakashole/teaching/eisenstein-nov18.pdf>

García-Nieto, M., & et al. (2017). Aplicación del proceso unificado en el desarrollo de un software que estima el inventario y el crecimiento-rendimiento maderable en plantaciones de eucalipto. *Madera y Bosques*.

Garlan, D., & Shaw, M. (1994). *An Introduction to Software Architecture*. Obtenido de [http://sunnyday.mit.edu/16.355/intro\\_softarch.pdf](http://sunnyday.mit.edu/16.355/intro_softarch.pdf)

Goldberg, Y. (2017). *Neural network methods for natural language processing*. Morgan & Claypool Publishers.

Gómez-Gutiérrez, J., & Olivares-Méndez, M. (2018). Natural language processing in databases: a systematic mapping study. *Computer Science Review*, 28, 96-114.

google. (25 de March de 2023). *Natural Language AI*. Obtenido de <https://cloud.google.com/natural-language?hl=es>

Guo, X., Wang, H., & Liu, W. (2016). A survey on Chat-Bot design techniques in speech conversation systems. *Journal of Network and Computer Applications*, 74, 119-132.

Haghighatpanah, M., & Teimouri, M. (2015). A natural language interface to relational databases using syntactic and semantic parsing. *Journal of Intelligent Information Systems*, 45(1), 29-51.

IBM. (25 de March de 2023). *Watson Assistant: Build better virtual agents, powered by AI*. Obtenido de <https://www.ibm.com/products/watson-assistant>

Jivani, H., & Zaveri, M. (2018). Chat-Bot for educational institutes using natural language processing and machine learning. *International Journal of Computer Applications*, 182(36), 19-23.

Jurafsky, D., & Martin, J. (2019). *Speech and language processing (3rd ed.)*. Pearson.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) .

Maldonado, L. (2012). *Los modelos ocultos de Markov, MOM*. TELOS. Revista de Estudios Interdisciplinarios en Ciencias Sociales.

Manning, C., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.

Mazzia, V., Giordano, A., Giusto, D., & Giordano, S. (2018). Using ontologies for natural language query interpretation in IoT environments. *Computer Standards & Interfaces*, 59, 131-138.

Mesa Sanchez, B. (Oct. de 2021). *FUNDAMENTALS OF DATA ANALYSIS*. Obtenido de <https://files.thesaurus.ie.edu/fundamentals-of-data-analysis.pdf>

Microsoft. (07 de March de 2029). *¿Qué es el SDK de Bot Framework?* Obtenido de <https://learn.microsoft.com/es-es/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0>

Mikolov, T., & et al. (2013). Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems*, 3111-3119.

Mitchell, T. (1997). *Machine learning*. . McGraw Hill.

Mohan, S., Chaitanya, K., & Kumar, P. (2021). Design and Development of a Chat-Bot-based Query Retrieval System for RDBMS using Natural Language

Processing. *International Journal of Advanced Computer Science and Applications*, 12(3), 125-131.

Sahu, S., & Rout, N. (2020). A Review on Natural Language Processing based Chat-Bots. *International Journal of Computer Applications*, 175(4), 6-12.

Seongja, K. (16 de Apr. de 2023). *Mathematical Foundations of Machine Learning*. Obtenido de [https://skim.math.msstate.edu/LectureNotes/Machine\\_Learning\\_Lecture.pdf](https://skim.math.msstate.edu/LectureNotes/Machine_Learning_Lecture.pdf)

Sharma, N., & Jain, P. (2019). Intelligent Chat-Bot for automated disease diagnosis using machine learning. *International Journal of Engineering and Advanced Technology*, 9(1), 3069-3075.

Silva, D., Rocha Souza, R., & Almeida, M. (2011). Ontologies and Controlled Vocabulary.

Sohail, M., Shah, M., & Iqbal, S. (2019). Chat-Bot Using Natural Language Processing and Machine Learning. *6th International Conference on Computing for Sustainable Global Development*. India: IEEE.

Spector, P. (25 de March de 2023). *Berkeley*. Obtenido de Introduction to SQL: <https://www.stat.berkeley.edu/~spector/sql.pdf>

SQL Tutorial. (17 de Apr. de 2023). *SQL Tutorial*. Obtenido de <https://www.hcoe.edu.np/uploads/attachments/r96oytechsacgzi4.pdf>

Xu, W., & Rudnicky, A. (2019). A survey of intelligent assistant technologies. *Foundations and Trends® in Information Retrieval*, 13(1), 1-127.

Zhang, Y., Sun A., & Liu, Z. (2019). A survey on Chat-Bot design techniques in speech conversation systems. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3), 25.

## 11 Anexo 1 Código

El código del Chat-Bot se puede encontrar en GitHub:

[https://github.com/balmorejbk/chatbot\\_vuelos.git](https://github.com/balmorejbk/chatbot_vuelos.git)