UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO INGENIERIA COMERCIAL
SANTIAGO - CHILE

RIESGOS ASOCIADOS A LA CREACIÓN Y USO DE APLICACIONES UTILIZANDO MODELOS GRANDES DE LENGUAJE (LLM)

SANTIAGO JESÚS VASCONCELLO ACUÑA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO COMERCIAL

PROFESOR GUÍA	:	SR. PABLO ISLA
PROFESOR CORREFERENTE	:	SR. THIERRY DE SAINT PIERRE.

Diciembre 2023

A quienes forman parte de mi vida . . .

(AGRADECIMIENTOS) [Título es opcional]

RESUMEN EJECUTIVO

Este trabajo investiga los riesgos inherentes a la creación y uso de aplicaciones basadas en Modelos Grandes de Lenguaje (LLM) en la industria. Su foco es en Procesamiento del Lenguaje Natural (NLP) específicamente la generación de texto, excluyendo otras formas de inteligencia artificial generativa. El proyecto se basa en la experiencia de desarrollar una aplicación utilizando LLM y analiza los riesgos que pueden afectar tanto al equipo de desarrollo como a los resultados obtenidos. Se determinan los factores de riesgo en la creación y uso de estas aplicaciones, utilizando como caso de estudio un proyecto de búsqueda de jurisprudencia en tribunales ambientales. La metodología empleada incluye la creación del proyecto, un ejemplo práctico de uso y la evaluación de riesgos en cada etapa del proceso. El objetivo es proporcionar una estructura para aplicaciones que usen LLM, analizar los problemas y riesgos asociados con el uso de información para alimentar estos modelos, incluyendo un proceso completo de ETL (Extract, Transform, Load).

Palabras Clave. Modelos Grandes de Lenguaje (LLM), Generación de Texto, Riesgo, Inteligencia Artificial, Proceso de ETL, Jurisprudencia Ambiental

ABSTRACT

This work investigates the inherent risks in the creation and use of applications based on Large Language Models (LLM) in the industry. Its focus is on Natural Language Processing (NLP), specifically text generation, excluding other forms of generative artificial intelligence. The project is based on the experience of developing an application using LLM and analyzes the risks that can affect both the development team and the obtained results. The risk factors in the creation and use of these applications are determined, using as a case study a jurisprudence search project in environmental courts. The methodology employed includes the creation of the project, a practical example of use, and the evaluation of risks at each stage of the process. The goal is to provide a structure for applications that use LLM, analyze the problems and risks associated with the use of information to feed these models, including a complete ETL (Extract, Transform, Load) process.

Keywords. Large Language Models (LLM), Text Generation, Risk, Artificial Intelligence, ETL Process, Environmental Jurisprudence

Índice de Contenidos

Índice de Figuras

1 | Introducción

La inteligencia artificial, también conocida como IA, ha experimentado un notable auge en la industria en los últimos tiempos de la mano de la llamada Industria 4.0 [8], especialmente en el ámbito en areas algo reacias como la administración y finanzas[7]. Este incremento no se debe necesariamente a un aumento en la capacidad de cómputo, ya que esta ha ido creciendo gradualmente a lo largo del tiempo (buscar respaldo). Anteriormente, aunque importante, no generaba tanto interés como en la actualidad. No fue sino hasta que OpenAI lanzó ChatGPT el 30 de noviemnbre de 2022 que el público en general pudo experimentar, probar y comprender de manera más completa la gran revolución llamada inteligencia artificial generativa [27].

De acuerdo con Google, "La inteligencia artificial generativa se refiere al uso de la IA para crear contenido, como texto, imágenes, música, audio y videos"[1]. Gracias a su interfaz amigable, resultó sencillo para personas de diversas industrias descubrir que existía una herramienta capaz de generar texto y responder preguntas de manera comprensible, incluso para aquellos que no eran expertos en tecnología.

La génesis de esta tesis se basa en la experiencia de llevar a cabo un proyecto utilizando estas tecnologías y los riesgos asociados a ellas. En este contexto, entendemos el riesgo como cualquier aspecto que pueda afectar tanto al equipo involucrado en la creación del proyecto como a los resultados obtenidos. El proyecto se centró en el uso de un modelo de lenguaje de gran envergadura, conocido como LLM por sus siglas en inglés, limitándose a la generación de texto. Por lo tanto, no profundizaremos en otros tipos de inteligencia artificial generativa, como la generación de imágenes o audio. El enfoque principal de este trabajo se concentra solo en el area del procesamiento del lenguaje natural aplicados a LLM.

1.1. Obejetivos

1.1.1. Objetivo General

Determinar los factores de riesgo que pueden llegar a influir, tanto de la creación como del uso de aplicaciones que utilicen modelos grandes de lenguaje (LLM) aplicados a la industria, usando de base el proyecto de búsqueda de jurisprudencia de los tribunales ambientales.

1.1.2. Objetivo Específico

1. Determinar una posible estructura de una aplicación usando LLM
2. Desarrollar los estados del Arte del uso de LLM y de los modelos generativo en si
3. Desarrollar los problemas que conlleva el uso de información para alimentar dichos problemas
4. Analizar un proceso de ETL de principio a fin para observar sus posibles riesgos

1.2. Metodologia

La metodología empleada en esta tesis se estructura en torno a tres componentes esenciales: la creación del proyecto, un ejemplo de uso concreto y la evaluación de los riesgos asociados a cada etapa del proceso, tanto en la fase de desarrollo del proyecto como en su aplicación práctica.

1.2.1. Creación del Proyecto

Esta fase inicial comprende el desarrollo del proyecto basado en IA generativa. Incluye los siguientes pasos:

- **Definición de Objetivos y Alcance:** Establecimiento claro de los propósitos y límites del proyecto, identificando las metas a alcanzar.
- **Selección de Tecnologías y Herramientas:** Evaluación y elección de las tecnologías y herramientas apropiadas para la implementación del proyecto.
- **Diseño de la Arquitectura:** Desarrollo de la estructura y componentes del proyecto, considerando aspectos de escalabilidad y rendimiento.
- **Implementación y Desarrollo:** Construcción efectiva del proyecto, incluyendo la programación y configuración de la inteligencia artificial generativa.

1.2.2. Ejemplo de Uso del Proyecto

Esta fase implica la aplicación práctica del proyecto en un contexto específico, demostrando su funcionalidad y utilidad. En este caso nuestro interés mas que en el output que genere la aplicación, es como funciona internamente el proceso cosa que para la siguiente etapa sea más fácil

1.2.3. Evaluación de Riesgos

Se trabajará la identificación y análisis de los riesgos potenciales en cada etapa del proceso, así como los riesgos derivados del caso de uso. Incluye:

- **Riesgos en la Creación del Proyecto:** Identificación de posibles obstáculos y contratiempos durante la etapa de concepción y desarrollo.
- **Riesgos en el Uso de la Aplicación:** Consideración de los riesgos asociados a la implementación práctica del proyecto en el contexto definido.

Esta metodología proporciona un enfoque integral para la creación y aplicación de un proyecto utilizando LLM, permitiendo una evaluación de los riesgos en cada etapa del proceso y en el caso de uso específico. Esto facilita la toma de decisiones informadas y la formulación de estrategias para mitigar posibles contratiempos.

2 | Estado del Arte

Actualmente, el desarrollo y uso de la inteligencia artificial está en boca de todos, pero ¿qué es esta famosa inteligencia artificial y por qué ha adquirido tanta relevancia recientemente? A pesar de que ha sido desafiante definir este concepto durante mucho tiempo, podemos decir que la inteligencia artificial, también conocida como inteligencia de máquina (Machine Learning en inglés), es el uso de la inteligencia demostrada por la tecnología y maquinas [19]. En general, la inteligencia artificial, abreviada como AI (del inglés Artificial Intelligence) o IA (de la palabra en español), engloba técnicas como el aprendizaje automático, el aprendizaje profundo y otros aspectos de la inteligencia artificial [19]. Estos temas no son nuevos y han sido objeto de estudio durante muchos años. Por ejemplo, en el caso del aprendizaje profundo (Deep Learning en inglés), este se basa en el perceptrón, descubierto en 1958 [40]. No fue hasta tiempos recientes, cuando el poder de cómputo y las interfaces han sido democratizadas para los usuarios, que hemos podido experimentar y entender realmente lo que la inteligencia artificial puede realizar.

Luego de entender lo que es la inteligencia artificial ¿Qué es lo diferente que no puede ofrecer actualmente? El 30 de noviembre de 2022 es abierto al publico la aplicación ChatGPT por la empresa OpenAI [34], deslumbrando a todos con la capacidad de responder las preguntas que le entregaban y con ello elevando aún mas el interés por esta empresa. La base de esta herramienta nace de una rama especifica de la inteligencia artificial llamada procesamiento del lenguaje natural, abreviado NLP, siendo este un subcampo de la Inteligencia Artificial y lingüístico, dedicado a hacer que las computadoras comprendan declaraciones o palabras escritas en lenguajes humanos [10].

El verdaero imparto que provocó ChatGPT en el mundo, fue el conocimiento popular de lo que hoy llamamos inteligencia artificial generativa, que puede ser definida como una técnica de inteligencia artificial que genera artefactos sintéticos analizando ejemplos de entrenamiento; aprendiendo sus patrones y distribución; y luego creando facsímiles realistas. La inteligencia artificial generativa (GAI) utiliza la modelización generativa y los avances en el aprendizaje profundo (DL) para producir contenido diverso a gran escala utilizando medios existentes como texto, gráficos, audio y video [21]. Por lo que, la población general pudo entender que existían herramientas que podían crear y con ello el boom entre la población fue cada vez mas grande.

Cuando hablamos de inteligencia artificial y sus aplicaciones, usualmente nos referimos a la generación de modelos. Sin embargo, en el contexto de la inteligencia artificial generativa, como las herramientas que producen texto o asisten en problemas de Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés), también estamos hablando de un modelo de inteligencia artificial. La diferencia radica en que estos son considerablemente más grandes en términos del volumen de datos que manejan. Dado que están enfocados en temas de lenguaje, comúnmente los denominamos modelos grandes de lenguaje o LLM, por sus siglas en inglés de “Large Language Models”, siendo estos formalmente definidos como herramientas de inteligencia artificial (AI) basadas en redes neuronales recurrentes multicapa que son entrenadas con vastas cantidades de datos para generar texto similar al humano [2].

ChatGPT funciona mediante una arquitectura base llamada Transformer [42], arquitectura creada por Google, que ha generado la gran revolución en la inteligencia artificial como la conocemos hasta la fecha debido a que no necesariamente se centra en NLP, sino en inteligencia artificial generativa en general. Si somos aún más específicos, GPT viene de Transformer generativo pre entrenado, "Generative Pre-trained Transformer" en inglés, siendo esta una arquitectura con habilidad de comprender el lenguaje de mejor manera usando Transformers [39]. Aunque no fue hasta que este modelo creció en la cantidad de parámetros que pudo mostrar sus capacidades en la gran amalgama de tareas de procesamiento natural, incluyendo la generación de texto [5].

Finalmente, es importante entender el funcionamiento de estos modelos ¿Cómo es posible que logren entender lo que escribo? Los modelos GTP funcionan en base a redes neuronales las cuales no entienden ni de letra y palabras, por lo que este texto tiene que pasar por una función de Embedding. Embedding es el proceso en el que representamos un texto, párrafo o documento de manera numérica, siendo esta representación en vector de múltiples dimensiones [32], estos vectores se pueden "grafica" en un espacio multi dimensional y con ello es posible ver la cercanía de cada uno de estos vectores entre ellos, por lo que este vector sirve como punto de entrada para el funcionamiento de los modelos GTP [31]. Ademas, se suelen usar para tareas como busqueda, agrupación, recomendaciones, busqueda de anomalias, etc [38].

3 | Creación del Proyecto

Los Tribunales Ambientales son órganos jurisdiccionales especiales, sujetos a la superintendencia directiva, correccional y económica de la Corte Suprema, cuya función es resolver las controversias medioambientales de su competencia y ocuparse de los demás asuntos que la ley somete a su conocimiento [25]. Estos tribunales generan una cantidad de jurisprudencia que puede ser encontrada en su portal de consulta llamado buscador ambiental [6].

El proyecto consiste en la generación de un chatbot en donde se pueda preguntar sobre la jurisprudencia de estos tribunales, aunque por razones de capacidad el chatbot se vea acotado solamente a las reclamaciones recibidas por el tribunal, con una estructura representada a Figura 3.1.

Por lo que, este proyecto consiste en un proceso de extracción de datos desde el buscador ambiental, transformación de estos datos para su utilización, generación de vectores de estos datos para que puedan interaccionar con la aplicación, carga de estos en una base de datos, para que después la aplicación pueda interactuar con ellos y mandando esa información a el LLM, siendo en este caso gpt-4 perteneciente a OpenAI.

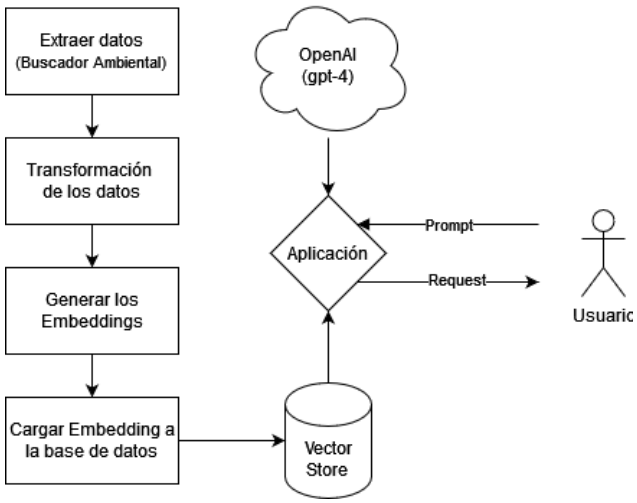

Figura 3.1: Estructura basica de la aplicación
(Fuente: Elaboración propia)

A partir de esta estructura mientras se avance en el desarrollo, se explicará parte por parte el proceso y con ello los riesgos de cada uno de ellos.

3.1. ETL

Para realizar el proyecto fue necesario realizar un proceso de ETL. El término ETL se refiere a las técnicas de .Extracción, Transformación y Carga"(Extract, Transform, Load), que constituyen un proceso clave para los datos necesarios para el proyecto. Este proceso implica la extracción de datos de fuentes heterogéneas, su transformación para ajustarse a las necesidades del negocio y su posterior carga en un destino que, por lo general, es un almacén de datos diseñado para el análisis y la generación de informes[3]. Siendo en este proyecto un estrutura como la Figura 3.2.

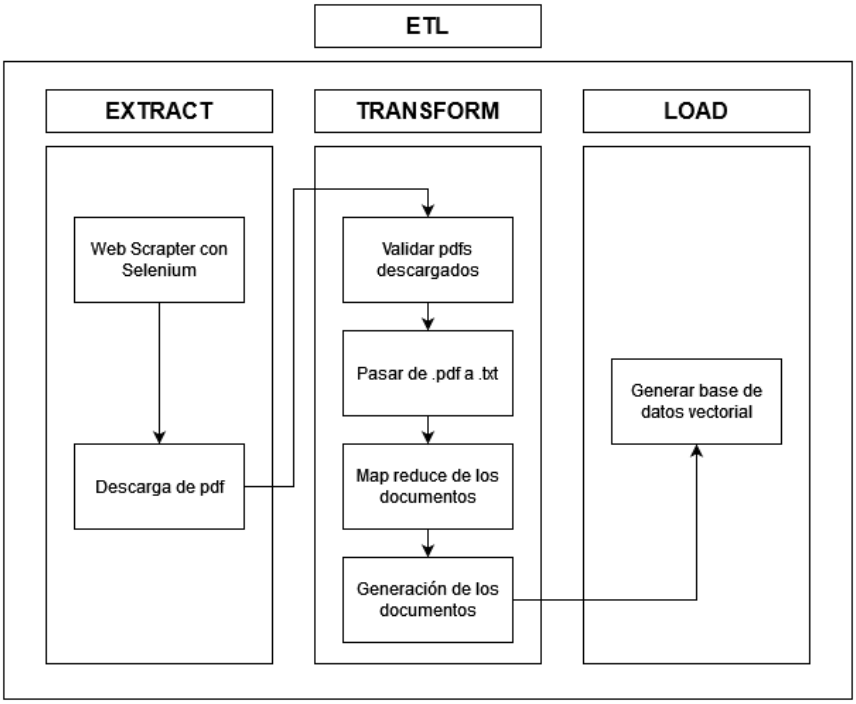

Figura 3.2: Estructura del proceso de ETL para el Buscador Ambiental
(Fuente: Elaboración propia)

La fase de extracción implica la recolección de datos de múltiples fuentes, que pueden variar desde bases de datos estructuradas hasta información no estructurada en la web. La transformación se refiere al proceso de limpieza, conversión, y consolidación de estos datos en un formato adecuado para el análisis. Finalmente, la carga es el proceso de transferir los datos transformados al sistema de destino, donde se pueden almacenar y utilizar para la toma de decisiones estratégicas [3].

3.1.1. Extract

La información requerida para el desarrollo del Chatbot se obtuvo del "Buscador ambiental"del Tribunal de Protección Ambiental de Chile a través de su sitio web[6]. Este portal aloja todos los documentos públicos disponibles para su consulta en cualquiera de los tres tribunales ambientales. Para acceder a la base de datos necesaria, se llevó a cabo la creación de un bot capaz de recopilar las entradas de este buscador de manera análoga a un usuario convencional.

Para esta tarea, se empleó Selenium, una herramienta originalmente diseñada para generar pruebas, pero que,

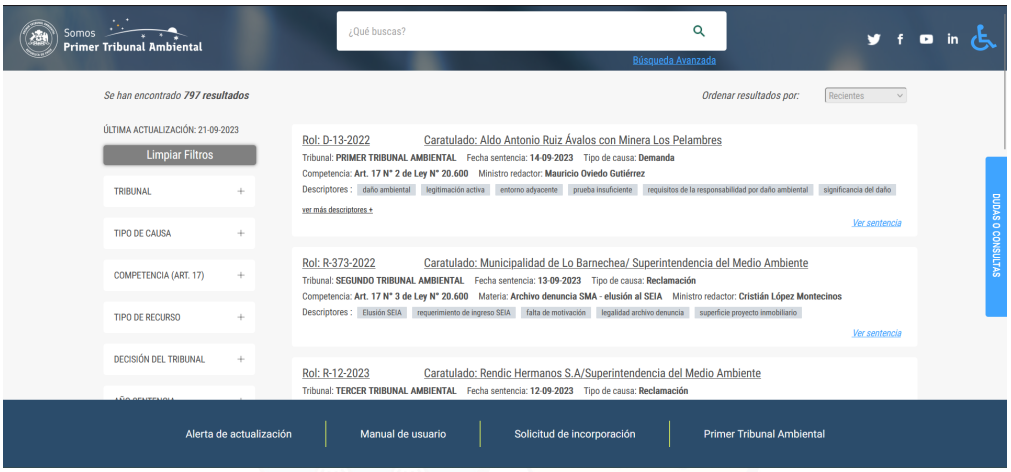

Figura 3.3: Buscador de la pagina del Primer Tribunal Ambiental
(Fuente: Pagina del Primer Tribunal Ambiental)

debido a la naturaleza reactiva y dinámica de los sitios web, así como a la detección de bots por parte de algunas páginas, resultó ser la elección más apropiada. Este bot, después de explorar todas las páginas del buscador ambiental, como se ilustra en la Figura 3.3, logró recuperar cada uno de los enlaces individuales que conducen a las páginas específicas de cada caso, tal como se muestra en la Figura 3.4.

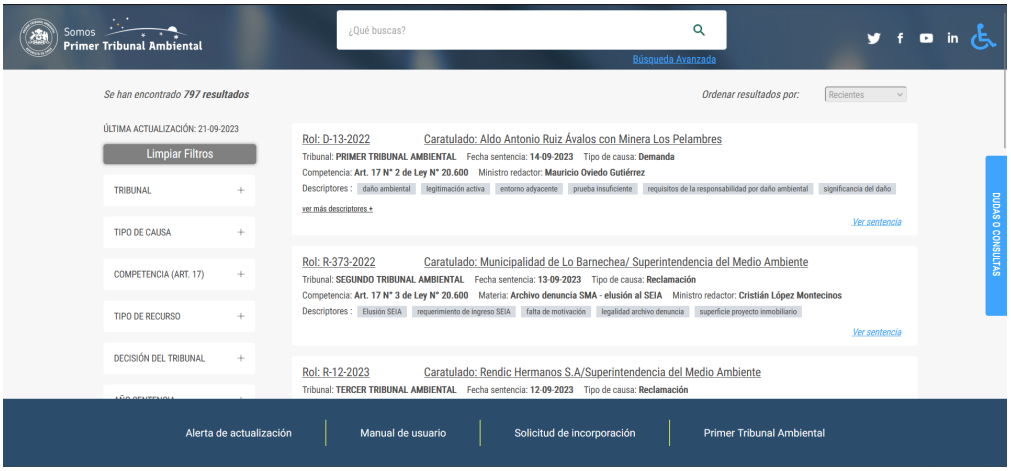

Figura 3.4: Screenshot de una sola reclamación en la pagina web del buscador ambiental
(Fuente: Pagina del Primer Tribunal Ambiental)

Posteriormente, se contemplaba la posibilidad de obtener tanto los enlaces a cada documento en formato PDF como la información detallada de cada uno de estos documentos mediante la creación de un nuevo bot. Sin embargo, durante el proceso de desarrollo de este bot, se logró acceder a la API que permitía obtener directamente todos los datos mencionados anteriormente. Esto suprimió la necesidad de crear otro tipo de bot utilizando Selenium, ya que bastaba con realizar una solicitud a la mencionada API.

Para completar la fase de extracción de datos (Extract), una vez que se había obtenido toda la información mediante las solicitudes a la API, el último paso consistió en generar nuevas solicitudes con el objetivo de descargar todos los archivos PDF de cada una de las entradas. Estos archivos ahora están descargados y listos para la próxima etapa del proyecto, que implica la transformación de los datos con el fin de obtener la información necesaria para construir la

base de datos a partir de los documentos.

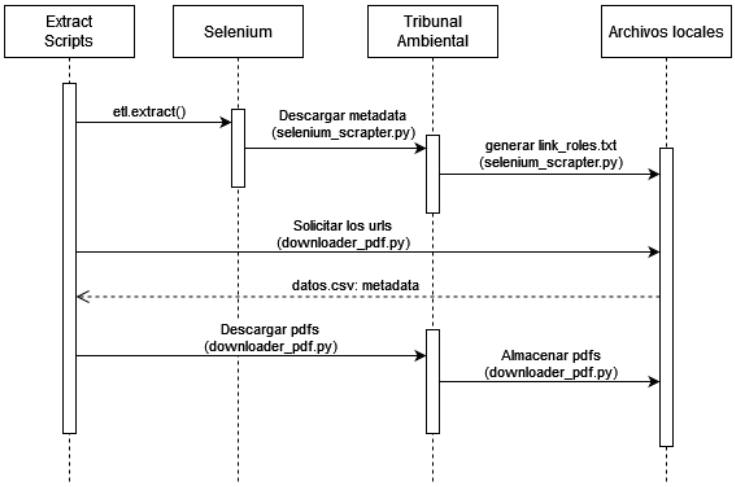

Figura 3.5:
(Fuente: Elavoración propia)

3.1.2. Transform

En la continuación en el proceso de ETL (Extracción, Transformación y Carga), los PDFs que previamente han sido descargados requieren ser sometidos a modificaciones con el objetivo de convertir la información que inicialmente se presenta en un estado "sucio" en datos "limpios" que puedan ser adecuadamente utilizados en el proyecto. Este proceso se denomina "transformación", o "transform," en inglés.

Entre los datos descargados, nos encontramos con un extenso número de PDFs que presentan dificultades significativas para su manipulación. Esto se debe a que el Tribunal Ambiental no sigue un formato estándar en la estructura de las reclamaciones presentadas. En consecuencia, cada uno de los textos posee un formato propio, lo que complica en gran medida la extracción eficiente de las diversas secciones contenidas en dichos textos. Sin embargo, gracias al funcionamiento del proceso de semejanza semántica, esta diversidad de formatos no representa un problema insuperable para el proyecto.

No obstante, surgen dificultades adicionales cuando se trata de las reclamaciones que son presentadas a los tribunales ambientales en formato digital o, en su defecto, en forma de fotocopias. Esto implica que no todos los documentos están habilitados para su procesamiento. En consecuencia, el primer paso en el proceso de transformación involucra la discriminación de qué PDFs son susceptibles de ser procesados y cuáles no. Para llevar a cabo esta tarea, se ha desarrollado un script capaz de detectar texto dentro de un archivo PDF. Si el texto es legible, se almacena; de lo contrario, se elimina.

Una vez separados los PDFs legibles y adecuados para el trabajo posterior, se procede con la transformación de estos documentos al formato TXT (texto plano). Esta etapa se lleva a cabo considerando la conveniencia de trabajar con archivos en formato de texto en comparación con los archivos en formato PDF puro, dado que el próximo método de transformación, que implica el uso de map-reduce en Langchain, requiere que los datos estén en formato de texto.

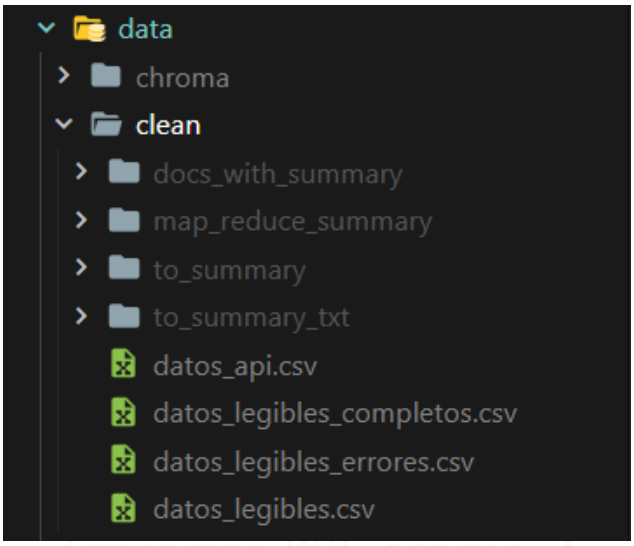

Figura 3.6:
(Fuente: Elavoración propia)

3.1.2.1. Map-Reduce

El proceso de Map-Reduce es un modelo de programación diseñado para procesar grandes cantidades de datos de manera eficiente, escalable y distribuida a través de clústeres de servidores. En el contexto de un archivo PDF muy grande, por ejemplo, si se quisiera resumir el contenido o analizar la frecuencia de ciertas palabras, Map-Reduce podría ser utilizado para dividir la tarea en partes más pequeñas y manejables. Primero, la función de map tomaría el texto del PDF y lo dividiría en elementos más pequeños, como párrafos o líneas, asignando a cada uno un resumen intermedio [30]. Luego, la función de reduce recogería todos los resumenes intermedios asociados con el documento y los combinaría para producir un resultado agregado, con un resumen de todo el documento.

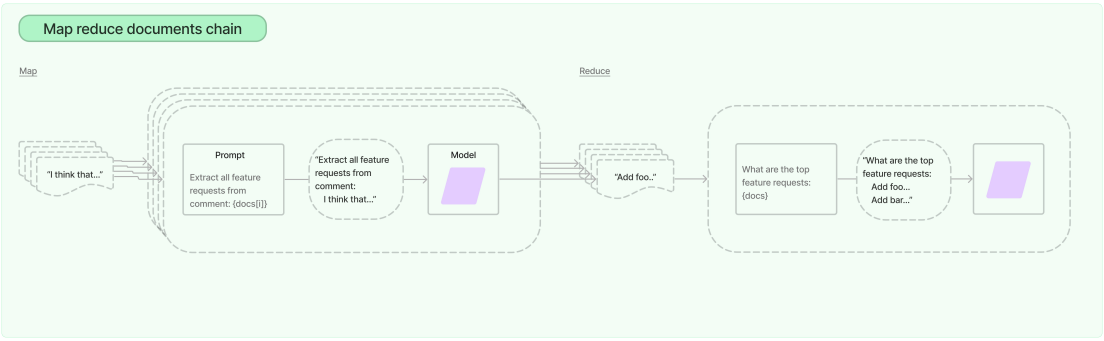

Figura 3.7:
(Fuente: Elavoración propia)

Sin embargo, es importante destacar que un archivo .txt puede contener un número de tokens demasiado elevado como para ser reducido de manera inmediata. En situaciones de este tipo, es necesario recurrir a un proceso de subdivisión que fragmente los textos en segmentos con un número de tokens inferior al límite impuesto por la API de OpenAI. Cada archivo .txt puede ser dividido, resumido y exportado a un nuevo archivo .txt una vez que ha sido fragmentado previamente en segmentos.

Los documentos procesados son combinados utilizando otro proceso de Langchain para obtener un resultado final consolidado. Para concluir el proceso de transformación, los resúmenes generados después de haber pasado por el procedimiento de map-reduce se someten a un último paso antes de ser incorporados en la base de datos. Este paso implica la fusión de los resúmenes con la información obtenida a través de la información Extraida por Selenium previamente, presentada en formato de texto. Este proceso resulta en la creación de un único documento que engloba toda la información, al cual nos referiremos como "documentos finales". Con esto, se concluye la fase de transformación y se procede al último procedimiento, conocido como "carga" (Load), que consiste en el almacenamiento estos documentos finales en la base de datos.

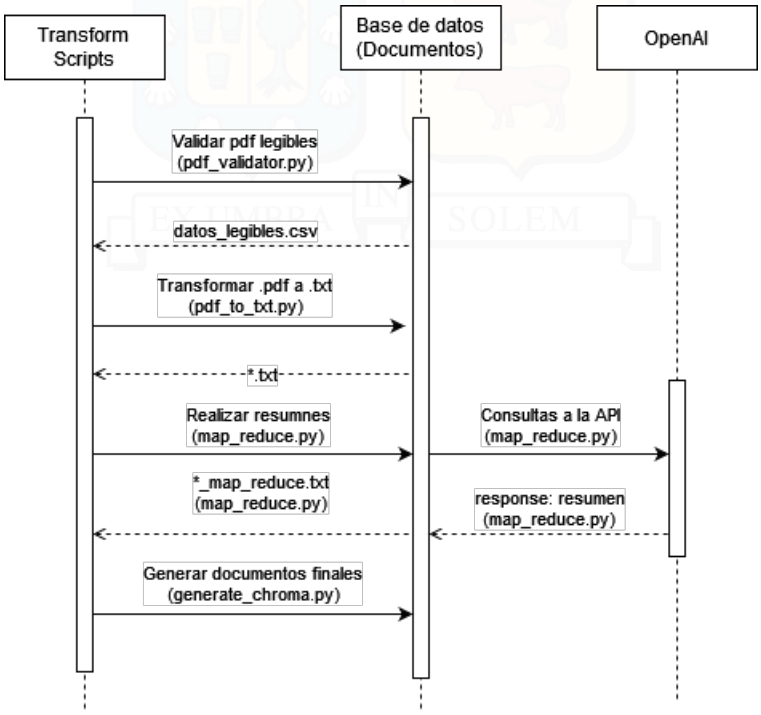

Figura 3.8: Diagrama de secuencia para el proceso de transformación (Trasform) de datos
(Fuente: Elavoración propia)

3.1.3. Load

Al culminar el proceso de Extracción, Transformación y Carga (ETL), resulta fundamental llevar a cabo la fase de carga, también conocida como "load" en inglés, en la cual se incorporan todos los documentos previamente descargados y transformados en una base de datos. Para este proyecto, en el cual se utiliza LangChain, resulta de vital importancia fragmentar los documentos en secciones más pequeñas, por lo que se deben dividir en chunks todos los documentos.

Esta necesidad surge debido a que los documentos deben ser sometidos a un proceso de Embeddings antes de ser introducidos en la base de datos. Esto se debe principalmente a que las funciones de Embeddings tienen un límite en la extensión de grupos de caracteres, conocidos como "tokens", que pueden ser procesados. En el contexto del modelo de Embeddings "text-embedding-ada-002", este límite se establece en 8191 tokens [33], lo que constituye la longitud máxima de los fragmentos.

3.1.3.1. Embeddings

Por lo tanto, cuando se trabaja con documentos extensos, es imperativo dividirlos en fragmentos más pequeños antes de proceder con su incorporación. Según la información proporcionada en el Blog de OpenAI, los embeddings son "representaciones numéricas de conceptos convertidos en secuencias numéricas, lo que facilita que las computadoras comprendan las relaciones entre los conceptos"[33]. En términos sencillos, los embeddings son representaciones vectoriales de texto que permiten su comprensión por parte del Modelo de Lenguaje de Gran Tamaño (LLM). Dado que los LLM son redes neuronales, el proceso de Embedding resulta esencial para traducir el texto en números, que es el formato comprensible para esta red neuronal.

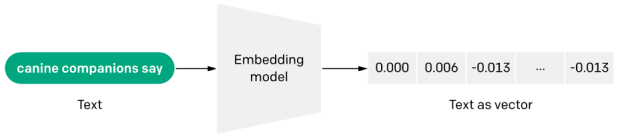

Figura 3.9:
(Fuente: Elavoración propia)

Los embeddings resultantes se almacenan posteriormente en una base de datos vectorial denominada ChromaDB. Esta base de datos ha sido diseñada para ser compacta, escalable y eficiente, con el propósito de almacenar y recuperar vectores de manera efectiva. ChromaDB genera índices que permiten una recuperación rápida y eficiente de los embeddings en función de las consultas realizadas por los usuarios[41].

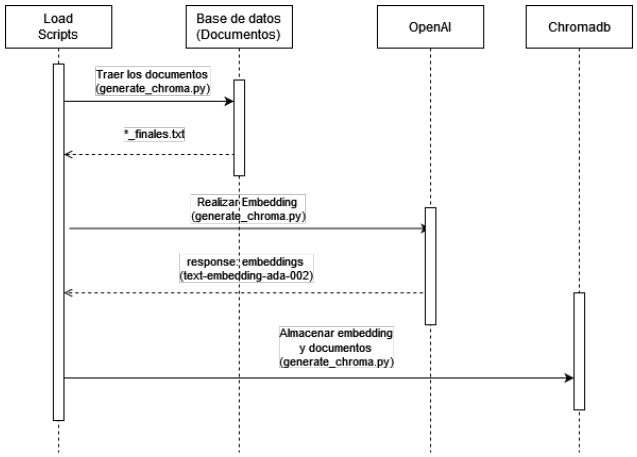

Figura 3.10:
(Fuente: Elavoración propia)

3.2. Chatbot

La estructura del chatbot se desarrolló en su totalidad utilizando el lenguaje de programación Python. Esta elección se debió a la experiencia del equipo en Python, lo que facilitó tanto la creación del frontend como del backend de la aplicación.

Para la parte de la interacción del usuario (frontend), se empleó Python junto con el framework Flask para la presentación de contenido en pantalla, incluyendo tanto la estructura de HTML como las hojas de estilo CSS. Además, se aprovechó la potencia del framework Bootstrap para agilizar el proceso de maquetación.

En cuanto al backend, se implementó una API con el objetivo de facilitar la interacción entre el frontend y el backend. Para este propósito, se utilizó FastAPI, un framework que permite la creación rápida de APIs y que ofrece la ventaja de contar con Swagger para la prueba y generación automática de documentación.

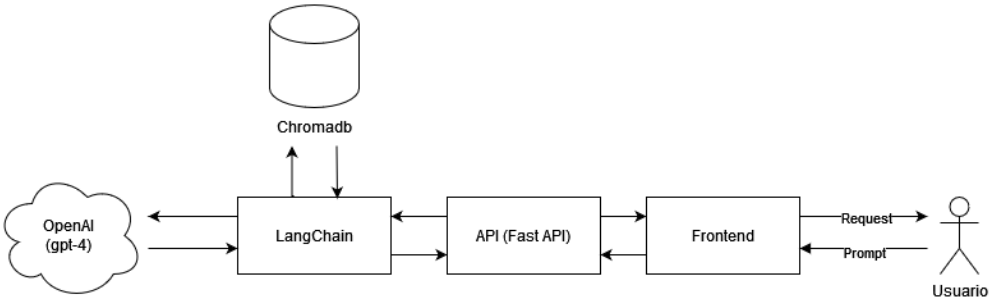

Figura 3.11: Diagrama de funcionamiento del Chatbot
(Fuente: Elavoración propia)

El funcionamiento interno de la aplicación se basa en el framework Langchain para la interacción con el Modelo de Lenguaje Grande (LLM). LangChain es un framework poderoso que simplifica el desarrollo de aplicaciones utilizando modelos de lenguaje grandes (LLM). Proporciona una interfaz única y personalizable capaz de gestionar diferentes LLM, incluida la gestión rápida, el procesamiento, el aumento de datos, la orquestación de agentes, el almacenamiento y la evaluación. Este marco versátil permite a los desarrolladores integrar perfectamente los LLM con sus flujos de trabajo del mundo real y datos con el mínimo esfuerzo [41].

Para la base de datos se utilizó ChromaDB. ChromaDB es una base de datos vectorial sin esquemas diseñada específicamente para su uso en aplicaciones de inteligencia artificial. Es liviano y muy potente, lo que permite el almacenamiento, la recuperación y la gestión eficiente de datos vectoriales (Embeddings), lo cual es esencial para las aplicaciones de chat de documentos basadas en LangChain y OpenAI [41]. Para que finalmente estas trabajaran en conjunto con los modelos de OpenAI, específicamente el modelo gpt-4 que actualmente es el más potente del mercado.

La función principal del chatbot es generar respuestas utilizando la técnica Retriever-Augmented Generation"(RAG), que implica proporcionar contexto adicional en el prompt enviado a OpenAI. Mientras que una generación simple suele constar de instrucciones y una consulta, en RAG se agrega contexto dentro del prompt con el propósito de reducir la probabilidad de alucinaciones y mejorar la calidad de las respuestas.

En cambio, mediante “Retriever-Augmented Generation” consiste al igual que el proceso anterior en entregar las instrucciones y una consulta, pero junto ello se agrega un contexto de este, cosa de que el modelo largo de lenguaje

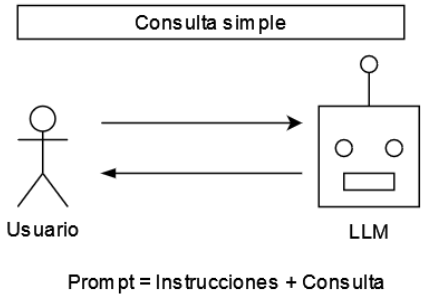

Figura 3.12:
(Fuente: Elavoración propia)

tenga menor capacidad de alucinar y generar una mejor respuesta. Podemos decir que “Retriever-Augmented Generatio” (RAG) se refiere a un modelo de generación de lenguaje que se mejora por la capacidad de recuperar información de una base de datos, como un índice de vectores que representan los documentos del buscador ambiental, además de la memoria que ya posee. Este enfoque se utiliza para mejorar la generación de respuestas en tareas de procesamiento de lenguaje natural (NLP) que requieren conocimiento intensivo [24].

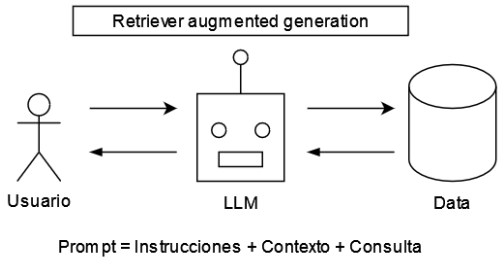

Figura 3.13:
(Fuente: Elavoración propia)

Finalmente, este chatbot envía el prompt con las instrucciones y el contexto obtenido mediante el RAG realizado con Langchain a el LLM, siendo en este caso gpt-4 de OpenAI, luego de pasar por una función de Embedding, cosa de que pueda ser leído por el modelo. Por lo que se realiza un request con la información a la API de OpenAI, para que se obtenga el output con la respuesta.

4 | Ejemplo de Uso del Proyecto

Esta tesis se centra tanto en el riesgo como en el desarrollo de un Chatbot, este chatbot funciona en base a la una arquitectura de RAG (Retriever-Augmented Generation) por lo que esta consiste en la recuperación de contexto el cual es enviado junto con el prompt al LLM. El proceso comienza con la obtención de un prompt específico del usuario, tal como “Dame un resumen del documento Dominga” dentro de la barra de búsqueda del frontend. Este prompt actúa como entrada inicial para el sistema de recuperación de información.

El prompt se procesa mediante una función de Embedding, empleando para ello OpenAI, siendo esta la función ’text-embedding-ada-002’. Esta función puede manejar hasta un máximo de 8191 tokens y produce un vector de 1536 dimensiones en forma de lista [33]. Para determinar la similitud entre el vector del prompt y los vectores correspondientes a los documentos almacenados, se utiliza la función de similitud coseno presente en la Ecuación 4.1.Esta mide el coseno del ángulo entre dos vectores, Siendo estos A y B respectivamente, y este proporciona un valor que refleja su proximidad semántica entre el vector del prompt y los vectores de todos los documentos almacenados en la base de datos ChromaDB.

$$\text{similitud\_coseno}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}||||\mathbf{B}||} \tag{4.1}$$

El proceso de embedding como se menciono antes lo que realiza en otras palabras es la conversión de un texto a un vector, esto debido a que los modelos de LLM al ser basados en redes neuronales necesitan un presentación grafica de estos texto a un formato el cual pueda ser legible por el modelo, por lo que se convierten en números como muestra la Figura 4.1 a continuación.

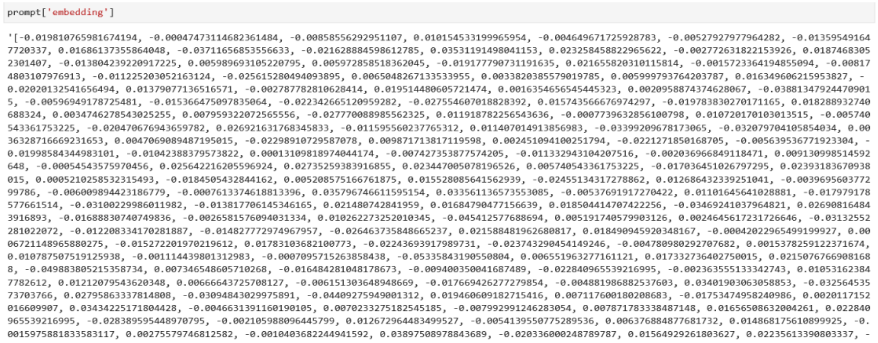

Figura 4.1: Representación de un prompt luego de pasar por la funcion de embedding ’text-embedding-ada-002’
(Fuente: Elavoración propia)

Una vez calculada la similitud coseno entre el vector asociado al prompt y los vectores que presentan los documentos que se encontraban en la base de datos, se procede a elaborar un ranking de los documentos más relevantes, según la similaridad obtenida del cálculo de similitud de cosenos. Esto se realiza seleccionando los 'n' documentos con los valores de similitud más altos, siendo en el ejemplo proporcionado un total de 3 presente en la Figura 4.2.

```
df.sort_values("similarity", ascending=False).head(3)
```

	rol	nombreProyecto	page	page_content	embedding	prompt	color	similarity
49	R-1-2017	Dominga	0.0	El caso Dominga\n\nRol: "R-1-2017", Rol: "R-1-...	[-0.003615596564486623, -0.0018946133786812425...	R-1-2017	1	0.791294
40	R-2-2014	Loteo Riberas de la Dehesa	0.0	El caso Predio Tres Bocas Valdivia\n\nRol: "R-...	[0.0069757443852722645, -0.005744528956711292,...		0	0.772962
36	R-283-2021	PTAS Gomero	0.0	El caso PTAS Gomero\n\nRol: "R-283-2021", Rol:...	[0.008394405245780945, -0.0023615628015249968,...		0	0.772533

Figura 4.2:
(Fuente: Elavoración propia)

Todo este proceso funciona internamente usando Langchain que consulta estos 'n' documentos, siendo en este caso 3, a la base de datos ChormaDB y son enviados como contexto dentro del prompt a OpenAI mediante el uso de su API, junto a la key de autorización, para obtener el resultado del modelo.

Finalmente, se obtiene el resultado por parte del modelo y es procesado por el frontend de la forma que se observa en la Figura 4.3, lo que da fin al proceso que realiza el Chatbot de principio a fin.

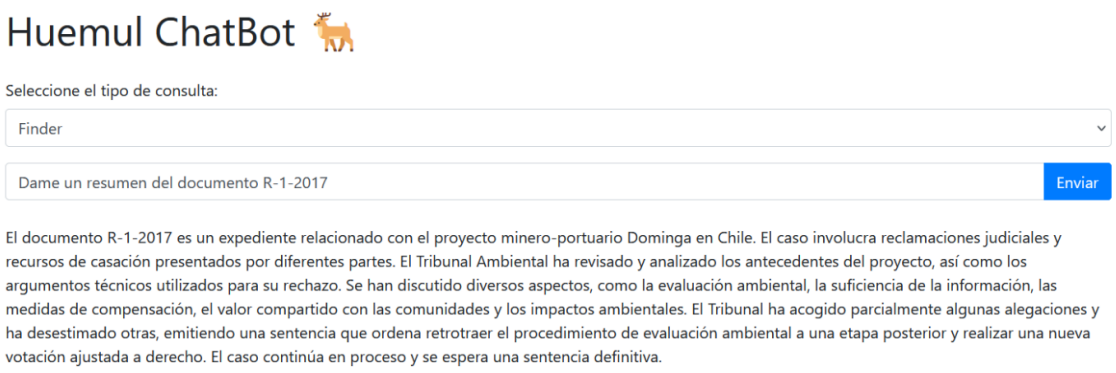

Figura 4.3:
(Fuente: Elavoración propia)

5 | Evaluación de Riesgos

5.1. Creación del Proyecto

5.1.1. No tener un análisis previo de que se busca lograr

La realización de un análisis previo resulta crucial para el inicio de cualquier proyecto basado en datos además del contexto de este. La mayoría de las personas que tienen acceso a herramientas analíticas realmente no entienden lo que sucede dentro de esas herramientas [17], dado ello es crucial que al empezar un proyecto se entienda a la perfección donde se quiere llegar y que realmente necesita.

Además, suele pasar que, en un conjunto de datos muy grandes, cualquier efecto que desee probar aparecerá como significativo [17], de ahí la importancia de un análisis previo. Esto es tan importante porque existen casos en donde un problema que puede resultar complejo en primera instancia puede sugerir el uso de un modelo complejo. Sin embargo, usando un modelo mas simple se llegan a resultados mejores que con el uso de complejo [15], siendo este un ejemplo claro de que un análisis previo adecuado puede mejorar tanto los resultados como la experiencia de trabajo.

5.1.2. Calidad de los datos

Para cualquier tipo de trabajo, aplicación o estudio la calidad de los datos es en extremo importante esto debido a que, la calidad de los datos es crítica para un sistema de Machine Learning, porque los datos deficientes podrían causar problemas graves como predicciones erróneas o baja precisión de clasificación [11]. Ahora si hablamos de LLM que entra en el terreno del Deep learning, simplificamos los atributos de calidad de datos en los tres más importantes para el Deep Learning: la fidelidad, la variedad y la veracidad de un conjunto de datos [11].

Como ejemplo dentro usando el proyecto que acompaña a este análisis de riesgo, se tuvo problemas en ciertas reclamaciones tratadas en el proceso de ETL, debido principalmente a que no existe un estándar para subir las reclamaciones en el buscador ambiental, existían pdf que eran legibles y otros que eran solo fotocopias, lo que imposibilitaba la extracción de información. Incluso dentro de la calidad de los datos, se puede extrapolar a la calidad de la metadata que era anexa en la base de datos.

5.1.3. Sesgos

Los Modelos de Lenguaje Grande (LLM), al ser entrenados con una masiva cantidad de datos, pueden manifestar sesgos debido a la procedencia de los datos utilizados en su entrenamiento. Estos sesgos pueden dar lugar a desafíos cuando se aplican en contextos distintos, ya que las respuestas generadas por el modelo pueden no ser adecuadas ni ajustarse a la realidad de esos nuevos escenarios.

Los modelos preentrenados con corupus generados por humanos contien sesgos sociales hacia ciertos grupos demograficos, estos sesgos son preocupantes, debido a que pueden ser propagados o incluso amplificados en las tareas que estos modelos realizan [16].

Como ejemplo podemos citar el dicho por Bill Gates en su entrevista “Can AI Save the World? Expert Insights with Bill Gates” en donde menciono que: “Los sesgos en los modelos de IA pueden llevar a diagnósticos incorrectos, como se vio en el ejemplo donde Chat GPT diagnosticó erróneamente la tuberculosis como gripe debido a las bajas tasas de tuberculosis en los EE. UU.” [29]. Con lo que podemos dimensional el efecto real de estos sesgos en lo correcto que puede llegar a ser una respuesta por parte de estos modelos.

5.1.4. Elección correcta del modelo

Actualmente la oferta de grandes modelos de lenguaje es muy amplia, desde los privados como: ChatGPT, PaLM, Bloom, etc. Como también modelos de código abierto como: Llama 2, OpenLLaMA, Falcon, Dolly, etc. [28] Con sus respectivas variantes, debido a que existen variables del modelo como por ejemplo Llama 2 que se puede encontrar en versión de 7, 13 y 70 billones de parámetros [22].

Elegir un modelo para trabajar es sumamente importante debido a que: la diversidad y calidad de los datos de preentrenamiento influyen sustancialmente en la capacidad del modelo de lenguaje para comprender y proporcionar respuestas precisas, que el tamaño puede tener una gran influencia en el rendimiento, que el soporte lingüistico podría ser crucial dependiendo la necesidad [26].

5.1.5. Costos Monetarios

Los costos relacionados con la creación o el uso de un modelo LLM pueden aumentar de manera exponencial. Por lo tanto, es fundamental tener en cuenta los costos asociados al utilizar un servidor externo, así como el costo de operar un servidor local, incluyendo el consumo de energía eléctrica. Por ejemplo, hay estudios en donde realizando un ajuste al modelo, fine tunning, es consumo de energía es comparable al de pequeñas ciudades y el dióxido de carbono emitido es equivalente a 500 veces la de un vuelo de ida y vuelta entre Nueva York y San Francisco [18].

5.1.6. Funciones de Embedding

Las funciones de Embedding son específicas para cada modelo de lenguaje y no son intercambiables entre modelos. Esto se debe a que los embeddings son representaciones de alto nivel provenientes de los pesos y parámetros de cada modelo, por lo que están diseñadas para captar y almacenar las relaciones semánticas específicas de cada modelo

[23]. Por lo que si quieres usar un modelo es necesario contar con su función de Embedding adema de la apacidad de contar con la posibilidad de usar dicha función.

5.1.7. Conocimiento de Framework

Cuando se desarrolla una aplicación, es esencial comprender el funcionamiento interno de los frameworks o herramientas que se están utilizando. Esta comprensión no solo es valiosa para comprender el proceso en su conjunto, sino que también es necesaria para tener un control los costos asociados al proyecto en caso de llevarse a producción.

Por ejemplo, en el contexto del proyecto, es importante saber que Langchain realiza múltiples llamadas a los modelos [9]. Si no se cuantifican de manera adecuada la cantidad de llamadas y la extensión de ellos, esto puede dar lugar a problemas de cuantificación de costos. Por lo tanto, la capacidad de comprender y medir con precisión el uso de recursos, como las llamadas a los modelos, es esencial para gestionar eficazmente los costos y asegurar el éxito del proyecto si es que quise ser llevado a producción.

5.1.8. Volatilidad del Mercado

Hasta la fecha actual, el 06 de noviembre de 2023, la creación de Chatbots utilizando el método RAG se perfilaba como una de las tendencias más destacadas en el mercado, siendo posiblemente una de las aplicaciones más prometedoras de los LLM. No obstante, en este mismo día, durante la OpenAI DevDay Keynote, se anunciaron novedades significativas, como la entrada en escena de los GPTs, que permite personalizar versiones de ChatGPT con instrucciones, conocimiento extra y cualquier otra combinación de habilidades [36]. Además, se introdujeron otros modelos como gpt-4 turbo, un playground de desarrollo para la herramienta, text-to-speech (TTS), entre otros [37].

En este contexto, comprometerse con cualquier tecnología conlleva riesgos, especialmente en este período caracterizado por una volatilidad extrema y una inversión extremadamente agresiva en inteligencia artificial. La Inteligencia Artificial Generativa continúa evolucionando de manera acelerada, lo que la hace cada vez más disruptiva y más eficiente. Por lo tanto, la investigación y la implementación de soluciones de inteligencia artificial centradas en asistentes o chatbots representan un compromiso de alto riesgo en este entorno en constante cambio.

5.2. Uso de la Aplicación

5.2.1. Entrega de contexto adecuado

Los LLM a menudo presentan alucinaciones, por lo que es esencial reducir la frecuencia de este fenómeno. Para lograrlo, la provisión de contexto dentro del prompt no es simplemente precisa, sino que resulta absolutamente indispensable. De hecho, la entrega de contexto adecuado dentro del prompt ha demostrado ser una medida altamente efectiva para reducir las alucinaciones, logrando una disminución de hasta un 99.88 porciento [12].

Por consiguiente, la correcta entrega de contexto dentro del prompt desempeña un papel fundamental en la generación de respuestas precisas a las consultas. Esto se debe a que, ya sea que el contexto proporcionado sea correcto, incorrecto o incluso irrelevante, el modelo de lenguaje lo utilizará como base para generar sus respuestas.

En el contexto del proyecto, la generación de respuestas se basa por completo en la entrega de contexto dentro del prompt, lo que a veces puede dar lugar a la transmisión de más información de la necesaria debido al funcionamiento del framework de Langchain. Esto puede llevar a situaciones en las que el modelo, influenciado por la información incorrecta o adicional proporcionada, genere respuestas que no reflejan un output con una respuesta en su totalidad correcta.

5.2.2. Limitaciones de la similitud de cosenos

La similitud del coseno, siendo este método más usado en modelos RAG para la extracción de contexto en la base de datos, como medida de similitud semántica en los embeddings, particularmente para palabras de alta frecuencia en tareas de procesamiento de lenguaje natural (NLP) como preguntas y respuestas (QA), recuperación de información (IR) y traducción automática (MT) presenta limitaciones en su uso, esto principalmente sucede debido a que la frecuencia de las palabras en los datos de entrenamiento afecta la geometría representacional de los embeddings contextualizados, siendo las palabras de baja frecuencia más concentradas geométricamente [46].

Por lo tanto, este problema se extrapola a que, en el momento de querer recuperar contexto pertinente de la base de datos, cuando se realiza el proceso de semejanza semántica entre el prompt y los vectores de la base de datos, este pueda recibir información no relacionada con el prompt, por lo que se enviar como contexto y puede dar oportunidad a alucinaciones.

5.2.3. Uso de información privada

Actualmente las empresas que entregan servicios de LLM son muy herméticos con la manera en que entrenan sus modelos, por lo tanto, no sabemos con qué información han sido entrenados la cual no necesariamente es solamente publica, además toda la información que preguntamos por ejemplo a OpenAI va a los servidores y sirve para reentrenar a los modelos.

Se ha probado que los ataques de reconstrucción de datos son posibles, se ha propuesto un ataque de reconstrucción dirigido de caja negra donde el adversario conoce parte de un ejemplo de entrenamiento (es decir, un indicio de texto) e infiere el resto (por ejemplo, un número de tarjeta de crédito), con lo que la posibilidad de extraer datos de entrenamiento de los modelos puede representar un riesgo serio para la privacidad [4].

Actualmente OpenAI esta siendo demandada tanto por violar los derechos de autor [43] como por robo sistemático [20], debido a que quienes demandas alegan que sus obras han sido usadas para entrenar a sus modelos de LLM, por lo que hasta que no tengamos total transparencia del proceso, a pesar de que existen servicios donde tus inputs supuestamente no son usados para reentrenar el modelo [35], es preferible ser cautelosos con la información que se manda a los LLM si estos no están corriendo de manera local.

5.2.4. Alucinaciones

El termino alucinación se refiere a la generación de textos o respuestas que exhiben corrección gramatical, fluidez y autenticidad, pero se desvían de las entradas de fuente proporcionadas (fidelidad) o no se alinean con la precisión factual (factualidad) [45]. Siendo en palabras más simples la entrega de información invetada por el modelo.

Dicho lo anterior, podemos decir que este es un gran factor de riesgo para el uso de una aplicación, porque a pesar de estar usando un sistema RAG, que dificulta la posibilidad de generar alucinaciones, sigue estando la posibilidad de que estas sucedan lo que puede entregar un output con información errónea y si es que no se revisa con criterio, se podría a llevar a cometer graves errores debido al uso de información que es directamente falsa.

5.2.5. Aprendizaje por Refuerzo con Retroalimentación Humana (RLHF)

Los modelos grandes de lenguaje suelen dar respuestas que a veces tanto política como moralmente no son correctas, por lo que las empresas tienen por objetivo alinear los valores humanos con los sistemas de aprendizaje automático y dirigir los algoritmos de aprendizaje hacia los objetivos e intereses de los humanos [47]. A esto se le llama Aprendizaje por Refuerzo con Retroalimentación Humana (RLHF), esto puede llegar a ser un problema si es que se busca realizar una aplicación en un usuario con una cultura diferente al proveedor del modelo o que si el usuario al no expresarse bien el modelo se confunda y no entregue una respuesta satisfactoria.

6 | Conclusiones

Tal como dijo Arthur C. Clarke: “Cualquier tecnología suficientemente avanzada es indistinguible de la magia”. Nos encontramos en un periodo donde la Inteligencia Artificial esta alcanzado capacidades que cada vez nos sorprenden y nos asustan por igual, donde posiblemente nos estamos ilusionando al no lograr lo que nos imaginamos en primera instancia, pero nos maravillamos viendo como logra cosas que ni siquiera nos imaginamos en un inicio.

El uso comercial de la inteligencia artificial en espacial de lo LLM, como fue la cobertura de esta tesis, presenta una cantidad muy alta de riesgos, tanto de los que son fáciles de prevenir como pueden ser los cuantitativos como prevenir costos excesivos en servidores como los difíciles de prevenir como las alucinaciones. Es importante para que estas aplicaciones tengan un buen futuro entender tanto como funcionan como de qué manera fueron creadas, los sesgos que presentan estos modelos no dejan de ser un reflejo de lo que somos y que le dimos de alimento a estos modelos para ser entrenados, siendo un gran reflejo de incluso como somos nosotros y la manera en la que actuamos, quedando claro que somos lo que consumimos.

Queda propuesto para quien quiera continuar con esta tesis el entregar resultados más empíricos que pruebas la viabilidad de usar en producción este tipo de chatbot, también queda propuesto el solucionar el problema de traer al contexto información que no era necesaria propia de la similitud de cosenos.

La industria de la Inteligencia Artificial deja expuesto a absolutamente todos los trabajos desde ahora en adelante en mayor o en menor medida, por lo que hay que tener cautela en las decisiones que se toman pues el riesgo es sumamente grande. La volatilidad del mercado posiblemente es y será el riesgo más grande por considerar para cualquier tipo de proyecto en el área, no fue hace mucho que los llamados Prompt engineer serían los profesionales más cotizados incluso mencionados así por el CEO en Nvidia [14]. Sin embargo, estos fueron ya rápidamente reemplazados por los mismos LLM que se supone tenían que domar, debido a la optimización [44] o el auto mejoramiento mediante generación de prompt producidos el mismo LLM [13], dejando de esa manera obsoleto un rol que hace menos de un mes tres meses a la fecha de publicación de esta tesis seria uno de los roles más importantes a futuro.

Finalmente, para cualquier tipo de proyecto sobre o con uso de Inteligencia artificial siempre lo más importante serán los datos y el criterio de científico de datos detrás de ellos, porque existirán datos y herramientas, pero sin un conocimiento de mercado al que se apunta, realizar cualquier tipo de acción es trabajar en la oscuridad porque sin criterio, trabajar con datos es un trabajo en vano y sin sentido.

Bibliografía

[1] *¿Qué es la IA generativa y cuáles son sus aplicaciones? | Google Cloud*. URL: https://cloud.google.com/use-cases/generative-ai?hl=es.

[2] Ian L Alberts y col. "Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be?" En: *European journal of nuclear medicine and molecular imaging* 50.6 (2023), págs. 1549-1552.

[3] Syed Muhammad Fawad Ali y Robert Wrembel. "From conceptual design to performance optimization of ETL workflows: current state of research and open problems". En: *VLDB Journal* 26 (6 dic. de 2017), págs. 777-801. ISSN: 0949877X. DOI: 10.1007/S00778-017-0477-2/FIGURES/18. URL: https://link.springer.com/article/10.1007/s00778-017-0477-2.

[4] Borja Balle, Giovanni Cherubin y Jamie Hayes. "Reconstructing Training Data with Informed Adversaries". En: *Proceedings - IEEE Symposium on Security and Privacy* 2022-May (2022), págs. 1138-1156. ISSN: 10816011. DOI: 10.1109/SP46214.2022.9833677.

[5] Tom Brown y col. "Language models are few-shot learners". En: *Advances in neural information processing systems* 33 (2020), págs. 1877-1901.

[6] *Buscador Ambiental*. URL: https://www.buscadorambiental.cl/buscador/#/.

[7] Yi Cao y Jia Zhai. "Bridging the gap–the impact of ChatGPT on financial research". En: *Journal of Chinese Economic and Business Studies* 21 (2 2023), págs. 177-191. ISSN: 14765292. DOI: 10.1080/14765284.2023.2212434. URL: https://www.tandfonline.com/action/journalInformation?journalCode=rcea20.

[8] Armanda Cetrulo y Alessandro Nuvolari. "Industry 4.0: revolution or hype? Reassessing recent technological trends and their impact on labour". En: *Journal of Industrial and Business Economics* 46 (3 sep. de 2019), págs. 391-402. ISSN: 19724977. DOI: 10.1007/S40812-019-00132-Y/TABLES/1. URL: https://link.springer.com/article/10.1007/s40812-019-00132-y.

[9] Harrison Chase. *Welcome to LangChain*. https://langchain-doc.readthedocs.io/en/latest/index.html. Accessed: 2023-11-06. 2022.

[10] Abhimanyu Chopra, Abhinav Prashar y Chandresh Sain. "Natural language processing". En: *International journal of technology enhancements and emerging engineering research* 1.4 (2013), págs. 131-134.

[11] Junhua Ding y col. "A case study of the augmentation and evaluation of training data for deep learning". En: *Journal of Data and Information Quality* 11 (4 ago. de 2019). ISSN: 19361963. DOI: 10.1145/3317573. URL: https://doi.org/10.1145/3317573.

[12] Philip Feldman, James R. Foulds y Shimei Pan. "Trapping LLM Hallucinations Using Tagged Context Prompts". En: (jun. de 2023). URL: https://arxiv.org/abs/2306.06085v1.

[13] Chrisantha Fernando y col. *Promptbreeder: Self-Referential Self-Improvement via Prompt Evolution*. 2023.

[14] Yahoo Finance. *Nvidia's CEO just gave a graduation speech about the future of work and said that A.I. won't steal jobs but 'someone who's an expert with A.I. will'*. 2023. URL: https://finance.yahoo.com/news/nvidia-ceo-just-gave-graduation-183507133.html.

[15] Edward J. Gregr y col. "Why less complexity produces better forecasts: an independent data evaluation of kelp habitat models". En: *Ecography* 42 (3 mar. de 2019), págs. 428-443. ISSN: 16000587. DOI: 10.1111/ECOG.03470.

[16] Yue Guo, Yi Yang y Ahmed Abbasi. "Auto-Debias: Debiasing Masked Language Models with Automated Biased Prompts". En: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. por Smaranda Muresan, Preslav Nakov y Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, mayo de 2022, págs. 1012-1023. DOI: 10.18653/v1/2022.acl-long.72. URL: https://aclanthology.org/2022.acl-long.72.

[17] D HAND. "Intelligent data analysis: Issues and opportunities". En: *Intelligent Data Analysis* 2 (1-4 ene. de 1998), págs. 67-79. ISSN: 1088-467X. DOI: 10.1016/S1088-467X(99)80001-8.

[18] Kai Huang y col. "Towards Green AI in Fine-tuning Large Language Models via Adaptive Backpropagation". En: *arXiv preprint arXiv:2309.13192* (2023).

[19] Whitney Hunt, Kendal Marshall y Ryan Perry. *Artificial Intelligence's Role in Finance and How Financial Companies are Leveraging the Technology to Their Advantage*. Disponible en SSRN: https://ssrn.com/abstract=3707908. 2020.

[20] Infobae. *Escritor de Juego de Tronos demanda a OpenAI por "robo sistemático"*. URL: https://www.infobae.com/tecno/2023/09/23/escritor-de-juego-de-tronos-demanda-a-openai-por-robo-sistematico/.

[21] Mladan Jovanovic y Mark Campbell. "Generative artificial intelligence: Trends and prospects". En: *Computer* 55.10 (2022), págs. 107-112.

[22] Lakera. *The List of 11 Most Popular Open Source LLMs of 2023 | Lakera – Protecting AI teams that disrupt the world.* URL: https://www.lakera.ai/blog/open-source-llms.

[23] Microsoft Learn. *LLM AI Embeddings*. URL: https://learn.microsoft.com/en-us/semantic-kernel/memories/embeddings.

[24] Patrick Lewis y col. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 9459-9474. URL: https://github.com/huggingface/transformers/blob/master/.

[25] *Ley Chile - Ley 20600 - Biblioteca del Congreso Nacional*. URL: https://www.bcn.cl/leychile/navegar?idNorma=1041361&idParte=9269911.

[26] Shreekant Mandvikar. "Factors to Consider When Selecting a Large Language Model: A Comparative Analysis". En: *International Journal of Intelligent Automation and Computing* 6 (3 ago. de 2023), págs. 37-40. URL: https://research.tensorgate.org/index.php/IJIAC/article/view/53.

[27] Puranjay Savar Mattas. "ChatGPT: A Study of AI Language Processing and its Implications". En: *International Journal of Research Publication and Reviews* 04 (02 2023), págs. 435-440. DOI: 10.55248/GENGPI.2023.4218.

[28] Meta. *Llama 2 - Meta AI*. URL: https://ai.meta.com/llama/.

[29] Mrwhosetheboss. *Can AI really save the World? ft. Bill Gates - YouTube*. URL: https://www.youtube.com/watch?v=l9m3IKG8i88&t=16s.

[30] N. K. Nagwani. "Summarizing large text collection using topic modeling and clustering based on MapReduce framework". En: *Journal of Big Data* 2 (1 dic. de 2015), págs. 1-18. ISSN: 21961115. DOI: 10.1186/S40537-015-0020-5/FIGURES/15. URL: https://link.springer.com/articles/10.1186/s40537-015-0020-5%20https://link.springer.com/article/10.1186/s40537-015-0020-5.

[31] Arvind Neelakantan y Lilian Weng. *Introducing text and code embeddings*. https://openai.com/blog/introducing-text-and-code-embeddings. Accessed: 2023-11-16. Ene. de 2022.

[32] Arvind Neelakantan y col. "Text and Code Embeddings by Contrastive Pre-Training". En: *arXiv preprint arXiv:2201.10005* arXiv:2201.10005 (ene. de 2022). URL: https://ar5iv.org/abs/2201.10005.

[33] OpenAI. *Embeddings - OpenAI API*. URL: https://platform.openai.com/docs/guides/embeddings/what-are-embeddings.

[34] OpenAI. *Introducing ChatGPT*. https://openai.com/blog/chatgpt. Accessed: 2023-11-16. Nov. de 2022.

[35] OpenAI. *Introducing ChatGPT Enterprise*. URL: https://openai.com/blog/introducing-chatgpt-enterprise.

[36] OpenAI. *Introducing GPTs*. URL: https://openai.com/blog/introducing-gpts.

[37] OpenAI. *New models and developer products announced at DevDay*. URL: https://openai.com/blog/new-models-and-developer-products-announced-at-devday.

[38] OpenAI. *What are Embeddings*. https://platform.openai.com/docs/guides/embeddings/what-are-embeddings. Accessed: 2023-11-16. Nov. de 2023.

[39] Alec Radford y col. "Improving language understanding by generative pre-training". En: (2018).

[40] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." En: *Psychological review* 65.6 (1958), pág. 386.

[41] Mirela Șorecău y Emil Șorecău. "AN ALTERNATIVE APPLICATION TO CHATGPT THAT USES RELIABLE SOURCES TO ENHANCE THE LEARNING PROCESS". En: XXIX (2023), pág. 2023. DOI: 10.2478/kbo-2023-0084.

[42] Ashish Vaswani y col. *Attention Is All You Need*. 2023.

[43] WIRED. *Comediante Sarah Silverman demanda a OpenAI y Meta por infringir derechos de autor*. URL: https://es.wired.com/articulos/sarah-silverman-demanda-a-openai-y-meta-por-infringir-derechos-de-autor.

[44] Chengrun Yang y col. *Large Language Models as Optimizers*. 2023.

[45] Hongbin Ye y col. "Cognitive Mirage: A Review of Hallucinations in Large Language Models". En: (sep. de 2023). URL: https://arxiv.org/abs/2309.06794v1.

[46] Kaitlyn Zhou y col. "Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words". En: *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 2 (mayo de 2022), págs. 401-423. ISSN: 0736587X. DOI: 10.18653/v1/2022.acl-short.45. URL: https://arxiv.org/abs/2205.05092v1.

[47] Banghua Zhu, Jiantao Jiao y Michael I. Jordan. "Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons". En: (ene. de 2023). ISSN: 26403498. URL: https://arxiv.org/abs/2301.11270v4.